Draft NISTIR 7682

Information System Security Best Practices for UOCAVA-Supporting Systems

Geoff Beier
Santosh Chokhani
Nelson Hastings
Jim Knoke
Andrew Regenscheid
Scott Shorter

National Institute of Standards and Technology
U.S. Department of Commerce

[This page intentionally left blank.]

Draft NISTIR 7682

Information System Security Best Practices for UOCAVA-Supporting Systems

Geoff Beier
Santosh Chokhani
Nelson Hastings
Jim Knoke
Andrew Regenscheid
Scott Shorter

April 2010

U.S. Department of Commerce
Gary Locke, Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Director

[This page intentionally left blank.]

Acknowledgements

The authors, Andrew Regenscheid and Nelson Hastings of NIST, and Geoff Beier, Santosh Chokhani, Jim Knoke, and Scott Shorter of CygnaCom, wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content. In particular, the authors would like to acknowledge Shirley Radack, Ray Perlner, Erika McCallister, Murugiah Souppaya, Karen Scarfone, and John Wack of NIST, Matt Masterson, and James Long of the Election Assistance Commission, and Carol Paquette, Mark Skall and Tom Caddy for their feedback on drafts of this document.

Comments on this publication may be submitted to:

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Electronic mail: uocava-voting@nist.gov

Table of Contents

Executive Summary

The *Uniformed and Overseas Citizens Absentee Voting Act* (UOCAVA) protects the absentee voting rights for U.S. Citizens, including active members of the uniformed services and the merchant marines, and their spouses and dependents who are away from their place of legal voting residence. It also protects the voting rights of U.S. civilians living overseas. Federal, state and local election administrators are charged with ensuring that each UOCAVA voter can exercise the right to vote. In order to meet this responsibility, election officials must provide assorted mechanisms that enable overseas voters to obtain information about voter registration and voting procedure descriptions, and to receive and return their ballots. UOCAVA also establishes requirements for reporting statistics on the effectiveness these mechanisms to the Election Assistance Commission.

In order to streamline the process of absentee voting and to ensure that these voters are not adversely impacted by the transit delays involved due to the difficulty of mail delivery around the world, Information Technology (IT) systems can be used to facilitate overseas absentee voting in several ways. They can:

- Distribute information about the process of applying for absentee ballots, including eligibility requirements and application forms.
- Distribute information about the facts relating to specific elections, including dates, offices involved and the text of ballot questions.
- Collect completed voter registration applications.
- Inform voters of their registration status.
- Provide ballot tracking information.
- Distribute blank ballots.
- Collect voted ballots.
- Maintain statistics used to prepare the UOCAVA-mandated reports.
- Maintain absentee voter registration information used to distribute ballots.

IT systems used to provide these functions face a variety of threats. If IT systems are not selected, configured and managed using security practices commensurate with the importance of the services they provide and the sensitivity of the data they handle, a security compromise could carry severe consequences for the integrity of the election, or the confidentiality of sensitive voter information. Failure to adequately address threats to these systems could prevent voters from casting ballots, expose individuals to identity fraud, or even compromise the results of an election. This document offers procedural and technical guidance, along with references to additional resources, to assist jurisdictions with the secure deployment of these systems. The guidance found in this document focuses on IT systems used to support overseas remote voting but does not define a specific architecture or configuration.

Component and system selection guidance

The technical controls outlined in this document rely on features that are frequently, but not always, found in commercially available IT products. In some cases, a product may appear to offer a feature but fail to support the options required for secure operation. Many of the practices required for secure operation are relevant to both IT systems as a whole and to the individual discrete components that may be used to build these systems. As a result, it is important that organizations or individuals responsible for selecting the IT products that will be deployed understand these controls and the features required to implement them both in the case of purchasing a turn-key system or selecting components to assemble into a system.

Care should be taken to ensure that IT products selected offer sufficient capabilities to be integrated and deployed as part of a UOCAVA voting system with the controls described in this document. The functionality and adequacy of these capabilities should be evaluated by a neutral third party or by the agency acquiring the products.

Component and system configuration guidance

In most cases, the IT products used to support overseas absentee voting will be general-purpose commercial products suitable for a wide variety of applications with widely differing security requirements. As such, these products will be

highly configurable. Many of the options offered by these products are not appropriate for every application, and could result in a security posture that is insufficient for a critical system or for one that contains sensitive data.

The guidelines in this document aim to assist system designers and administrators in two ways. First, as systems and components are configured for operation, this document lists sets of controls and configuration options that are critical to system security. When creating configuration checklists for systems which will support voting, every type of control should be addressed for every component where it can be applied. Second, this document details options for security controls which jurisdictions can use to help meet their security objectives for voting applications. The configuration practices found in this document aim to ensure that selections appropriate to the criticality and sensitivity of the systems are made, and address all security-critical facets of configuration. Depending on the architecture or implementation of the overseas remote voting system, jurisdictions will have customized their configurations.

Operational Guidance

Finally, both technical and procedural controls are critical to securing these systems in operation. Organizations operating IT systems in support of UOCAVA voting should have comprehensively-documented, detailed security procedures for bringing the systems to a secure operating state, maintaining that secure state during operation, and securely terminating operations.

The guidance in this publication will assist election officials in collaborating with system designers and administrators to define roles and establish processes that ensure the ongoing secure operation of the systems. It should also be consulted by system designers when documenting system operations and by administrators when assigning individuals to fulfill roles defined by the system design.

1 Introduction

To support State and local election officials in carrying out their responsibilities under the Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA), the Election Assistance Commission (EAC) requested that the National Institute of Standards and Technology (NIST) research electronic technologies that could facilitate the UOCAVA voting process. A number of state and local jurisdictions have begun to use information technology (IT) systems and the Internet to facilitate UOCAVA voting. These systems have been, and are being, used to distribute election information to voters, to send and collect voter registration and ballot request forms, to deliver blank ballots, and to receive voted ballots. This document is intended to provide jurisdictions with a set of computer security best practices that can be used as a baseline set of controls for securing their IT systems, and the supporting infrastructure. It examines the large collection of cyber security resources, including standards, guidelines, tools, and metrics, that NIST has developed to help federal agencies under the Federal Information Security Management Act (FISMA) of 2002 and summarizes them for those designing, deploying, or using information technology systems that support UOCAVA voting.

In December 2008, NIST released NISTIR 7551, *A Threat Analysis on UOCAVA Voting Systems* [NISTIR7551], which documents the threats to UOCAVA voting systems using electronic technologies for all aspects of the overseas voting process. NISTIR 7551 identified a number of threats to using electronic technologies to obtain voter registration materials, deliver blank ballots, or return cast ballots, emphasizing the need for implementing strong and comprehensive security controls to mitigate the identified threats. While NISTIR 7551 discussed high-level security controls capable of mitigating threats, the focus of that report was identifying technologies and associated risks. This document complements NIST 7551 by providing detailed security best practices to help jurisdictions obtain, deploy, manage and use UOCAVA voting systems based on security practices used in other IT applications.

At the time of the release of this draft, the EAC has posted a draft of their *UOCAVA Pilot Program Testing Requirements* document [PILOTREQ]. The *UOCAVA Pilot Program Testing Requirements* document defines conformance requirements for remote electronic voting systems using a manned-kiosk architecture that is intended for use in a UOCAVA pilot program. Nothing in this document should be construed to supersede any requirements provided in the EAC's *UOCAVA Pilot Program Testing Requirements* document. The scope of this document is much broader than the UOCAVA pilot program thus some of the best practices described in this document may not be suitable for the specific pilot architecture.

1.1 Purpose and Scope

This document provides best practices for the secure operation of information systems that support overseas voting in accordance with the requirements of the Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA) [HAVA, UOCAVA]. These best practices are based on existing NIST standards and guidelines used to secure non-national security information systems. This document summarizes the standards and guidelines that were deemed most applicable for jurisdictions using IT systems to support UOCAVA voting. For more detailed standards and guidelines, readers should consult the original NIST publications on a particular subject matter.

IT systems may be used to support UOCAVA voting in a variety of ways including managing or obtaining voter registration material, tracking requests for absentee ballots, providing or delivering blank ballots, or deploying remote electronic absentee voting systems. How information systems are specifically used to support UOCAVA voting will vary across jurisdictions, as different state and local jurisdictions have different procedures and systems for dealing with overseas voters. The appropriate security controls for these systems will be highly dependent on the type of systems that are deployed and how they are used. Since there are many potential ways to use IT systems to support UOCAVA voting, it is infeasible to provide detailed best practices for every possible architecture application, and configuration. Instead, this document provides a set of minimum security controls that should be applicable to any type of IT system used to support UOCAVA voting, including best practices for technical, physical personnel and procedural security of such systems.

The best practices in this document are intended to be broadly applicable to all voting systems supporting UOCAVA that leverage IT systems, but they do not cover all requirements for all UOCAVA voting systems. The baseline best practices provided must be augmented with additional safeguards depending on a jurisdiction's particular circumstances. After implementing the best practices described in this document, jurisdictions should carefully consider the type of UOCAVA voting system deployed, and its context of use to determine what additional security measures are required. It may not be possible to protect system-specific threats, such as those that would be unique to ballot delivery or return systems, using only the best practices described in this document. As described in NISTIR 7551 *A Threat Analysis on UOCAVA Voting Systems*, some types of UOCAVA voting systems face threats that are very difficult to mitigate with current technology, such as remote voting from personal computers. Jurisdictions must consider the potential threats to a UOCAVA voting system, along with the totality of security controls and measures implemented in the system, when determining whether the system is within an acceptable level of risk.

1.2 Intended Audience

This document contains detailed discussions of technical, procedural and managerial controls for information systems used to support UOCAVA voting. This document is directed toward readers who have a high degree of technical literacy of computer and network components, as well as computer security technologies. The primary audience for this document is technical personnel charged with implementing, deploying or maintaining UOCAVA voting systems. This includes technical support staff at state or local jurisdictions, vendors of products aimed at supporting UOCAVA voting, and service providers that host UOCAVA voting systems. It is important for jurisdictions to direct the information found in this document to the appropriate department or organization. In some cases, the individuals charged with supporting information technology equipment may not realize the equipment is used to support UOCAVA voting. For instance, technical staff may provide support for all county information systems, including those used by election officials and administrators for UOCAVA voting.

This document refers to system designers, implementers, operators, auditors and administrators. These roles are defined relative to the IT system used to support UOCAVA voting. They may not directly correspond to job titles within the organization(s) assembling, procuring, deploying or maintaining these systems. For example, an individual who holds the title “System Administrator” in an organization’s IT department may be charged with designing and deploying a system that sends blank ballots via email.

In addition, contracting officers, IT support staff, and other technical staff charged with making technical recommendations to policymakers may find this document useful as informative background material. Contracting officers may be able to identify specific security functionality that should be present in UOCAVA voting systems when evaluating products. Technical staff making technical recommendations to policymakers can use the background material in this document when weighing the advantages and disadvantages of different technical solutions to security issues. In addition, this document can be a useful guide for ensuring a jurisdiction employs a minimum baseline of security controls to protect UOCAVA voting systems and associated data.

2 General Overview

This section identifies the components that may be used to support UOCAVA voting and lists the technical security, operational and assurance controls that apply to the secure deployment, management and operation of the system.

IT systems facilitating UOCAVA voting can be used to support the following activities:

- Information delivery.
- Voter registration.
- Electronic blank ballot delivery.
- Remote electronic voting from controlled environments.
- Remote electronic voting from personally-owned systems.

The remaining sections of the document describe the controls in detail and offers guidelines for how these controls can be used to design, deploy and operate an overseas voting system. Because the roles of administering an election are different than the roles of administering an IT system, individuals are identified by their role relative to the system being deployed. This may not be the same as their role within the organization deploying the system. For example, a system administration team in a jurisdiction's IT department may be tasked with selecting, assembling, deploying and managing components used in a web application where voters can download blank ballots. Even though members of this team might be considered system administrators within the organization, relative to the voting system they are both designers and administrators.

Different sections of this document will be of more or less interest to the reader based on their role relative to the deployed UOCAVA voting system. Section 3 is primarily intended for designers of systems used to support remote absentee voting. The specific guidance in sections 4, 5 and 6 are intended for system administrators and other technical staff who will be charged with deploying the systems. These sections additionally provide important background material of interest to system designers. Section 7 is intended for systems administrators and technical staff who will be charged with the secure operation of these systems. Section 7 provides guidance for designers and other personnel tasked with selecting components which will be integrated into the voting system, along with informative background material for system administrators.

2.1 Overseas Voting Systems Components

The following identifies information technology components that may be found in IT systems deployed in support of overseas and military voters and explains the security objectives they can achieve. These components could exist as separate devices or multiple components may be located on a single device. For example, a firewall could be a hardware appliance on the network, a software process operating on each computer system, or both.

The components of an Internet-connected IT system supporting UOCAVA voting can be quite different than those used in a more traditional polling place voting systems. Polling place systems are often closed systems, where the voting system components, and any supporting infrastructure, are used only for conducting elections. An IT system that supports UOCAVA voting, particularly one that is Internet-connected, will almost certainly be a more open system. These systems may reuse a jurisdiction's existing communications infrastructure that is also used for important functions other than voting and elections. However, the IT systems that are directly used by election officials and voters rely on that infrastructure for important security protections. As such, this document contains best practices for IT components that may not be traditionally viewed as a component of a voting system, such as a hardware firewall appliance, or an intrusion detection system.

In this document, the term *server* is used to describe a computer system that primarily stores and/or manages data for various users and applications, and/or executes voting applications. The term *workstation* is used to describe a computer system that is used by a single user or limited number of users to perform individual tasks on the system itself or to access the servers.

An IT system facilitating UOCAVA voting may contain some or all of the following components:

- **Election Administration Components**
    - **Voter Registration Database:** Contains applicable information for registered voters.
    - **Administrative Console:** Used by the system administrators to manage the voting system, such as updating system software and monitoring event logs.

- **Election Official Workstation:** Used by the election officials to perform election related functions, such as creating ballot definitions and corresponding with voters via email.

- **Communications Components**
  - **Web Server:** Used to provide a browser-based interface and workflow for the users of the voting system.
  - **E-Mail System:** Used to send and receive e-mails from the voters, such as inquiries from voters, and attachments of blank ballots, and voter registration forms.
  - **Fax System:** Used to send blank ballots to the voters and to receive filled out ballots from the voters.
- **Security Components**
  - **Firewall:** Used to protect internal systems and network from unauthorized access and unauthorized communication traffic, and to block attack attempts from external systems and users.
  - **Intrusion Detection System (IDS) and Intrusion Prevention System (IPS):** Used to prevent and detect attacks attempted against the system and network, and to notify administrators.
  - **Authentication System:** Used for voters, election officials, and administrators to identify and authenticate themselves in order to perform their authorized functions.
  - **Public Key Infrastructure (PKI) Certification Authority (CA):** Used to issue public key certificates to web servers and users for use in Transport Layer Security (TLS) and other forms of authentication.
  - **Event Logging System:** Used to capture security and voting-related events in logs for accountability and forensic purposes.

This document covers only computer systems under the control of their respective election jurisdictions, or other parties designated by jurisdiction with the responsibility of operating those systems. As such, the security of voters' personal computers is not addressed in this document. However, voters may use jurisdiction-administered systems to interact with the voting system, as would be the case with kiosk-based systems. In these instances, jurisdiction-administered kiosks should be protected using similar controls to those used on election official workstations.

Remote overseas voting systems require information to be exchanged between the different components. How the information is exchange between components can take different forms. Information can be exchange between components by a connected set of computer systems such as a local or wide area network (LAN/WAN) or the Internet. Alternatively, physically moving storage media such as a disk or thumb drive between components can be used to exchange information. However information is exchanged between components, it needs to take steps to secure the exchange.

Not every overseas voting system will contain all of these components. For example, a system that merely delivers information to the voting public need not be connected to a voter registration database. It may also not need an e-mail system or a fax system. In systems that don't make heavy use of public key infrastructure, designers may opt to obtain and import certificates and revocation data from an external certification authority service rather than operate one as part of the voting system. However, most of the best practices described in this document will be applicable to any internet-connected system that is important to the election process. The implementation of these practices will often involve configuring and deploying security components, such as firewalls and intrusion detection systems.

2.2 Technical Controls

Technical security controls need to be established in the following areas in order to achieve the jurisdiction's security objectives for their UOCAVA systems:

1. ***Identification and Authentication (I&A)*** controls are used to establish the identity of a user and convey that identity to the system and applications running on the system.
2. ***Access Control*** uses the result of the I&A mechanisms to make a determination, either at the system or application level, whether a user is authorized to access data or perform operations on that data within the system.
3. ***Personally Identifiable Information (PII) Protection*** controls deal specifically with identifying and restricting the exposure of data that could be used to identify individuals while enabling sufficient access to this information that the system can function as intended.
4. ***Confidentiality*** controls detail mechanisms that ensure that potentially sensitive information about indviduals and about the system are is protected both in transit and at rest.

5. ***Integrity*** controls ensure that information critical to the proper functionality of the system cannot be undetectably altered in transit or at rest.
6. ***Availability*** controls are intended both to prevent situations which would render the system inoperable at critical times and to enable swift restoration of important functionality if these situations should arise.
7. ***Cryptographic Security*** controls support I&A, confidentiality and integrity protection using FIPS-standardized cryptographic mechanisms.
8. ***Communication Systems*** controls focus on maintaining the security and availability of the channels used to transmit data between the voting system and external systems and users.

Section 3 describes each category of control in detail and outlines specific options that may be available in various systems to support these. Section 4 expands on specific network-level protections required to enforce these controls. Section 5 discusses host-level protections used to implement these controls.

2.3 Operational Controls

Operational controls need to be established in the following areas in order to achieve the jurisdiction's security objectives for their UOCAVA systems

1. ***Facility Controls*** address physical security requirements for the equipment and wiring used to support the system.
2. ***Media Storage Controls*** establish physical and logical mechanisms for restricting the distribution of and access to media that contain sensitive information.
3. ***Personnel Security Controls*** are used to establish roles, duties and qualifications for those individuals tasked with operating the system.
4. ***Event Log Processing*** procedures are aimed at ensuring that system logs both constitute a complete record of system activity and are reviewed frequently enough to offer assurance that the system is operating as intended.
5. ***Backup and Archive*** procedures are intended to ensure both that a system can be audited in the future and that data sufficient to implement the Disaster Recovery controls is maintained.
6. ***Configuration Management*** controls ensure that a system is deployed and maintained in accordance with its functional and security objectives over its entire lifecycle.
7. ***Disaster Recovery*** controls are intended to ensure that an appropriate plan is established to enable restoration of system functionality in the event of unanticipated catastrophic failures.
8. ***Ongoing Testing*** is used to establish confidence that a system continues to meet its design goals.
9. ***Incident Handling*** processes establish a mechanism for reporting and remediation of security failures.
10. ***Removal from Service*** controls ensure both that the ability to audit events is preserved when systems are removed from service and that sensitive information is not exposed by systems that are no longer in service.

Section 6 describes these operational controls in detail and discusses their application to UOCAVA systems.

2.4 Assurance Controls

Assurance controls are subtly different from security controls. Where security controls are used to protect the data and functionality of a system in accordance with its design objectives, assurance controls serve two related purposes. First, they offer evidence that the security controls are in fact sufficient to meet these objectives. Secondly, they are used to establish confidence that these security controls are deployed and maintained. The assurance controls which are most important to UOCAVA systems fall into the following categories:

1. ***Documentation Requirements*** address both the design documents required to assure implementers that a given design meets the system's security objectives as well as documentation of those procedures necessary to install, configure and maintain the system in accordance with its design goals.
2. ***Vulnerability Analysis*** documentation offers evidence that potential vulnerabilities were considered and addressed during the design and deployment of a system.
3. ***Testing Requirements*** detail the test documentation used to establish that the above areas have been properly evaluated.

In short, assurance controls govern the documentation and testing required to demonstrate that the security best practices found in this document are followed for a particular system. The assurance controls take the form of design documentation to demonstrate how the system was designed to meet the IT security best practices, a vulnerability analysis explaining how common exploits for such systems and well-known security holes in system components are mitigated, and administrative guidance that instructs administrators in the secure operation of the system. Testing includes functional and penetration testing of the system performed as part of the development process. Assurance controls are described in detail in Section 7.

3 Security Controls

3.1 Identification and Authentication (I&A)

Authentication is the process of establishing confidence in the claimed identity of a user or system. Establishing the identity of a user is critical to the security of the system since the authenticated identity forms the basis for what actions the user can perform on the system and what information the user may access. Any IT system used to support UOCAVA voting will likely have several classes of users, each with their own set of rights and privileges on the system. The strength of authentication necessary depends on the consequences of an authentication error. As such, users with more privileged levels of access should, in general, be authenticated with a higher level of assurance. For example, three likely classes of users on an IT system supporting UOCAVA voting are system administrator, election officials, and voters.

This section summarizes guidelines from NIST Special Publication (SP) 800-63, *Electronic Authentication Guideline*, [SP800-63] and explains how these apply to UOCAVA systems in general. The primary audience for this section is system designers. Other readers should refer to this section and to [SP800-63] as needed.

In this section, we first offer general background information on the identification and authentication systems and then provide the best practices that are applicable and feasible for the various types of information technology systems described in Section 2.1. The remainder of this section is divided into the following subsections:

1. Threats to Credential Issuance Methods and Mitigations
2. Credential Issuance Methods
3. Threats to Authentication Mechanisms and Mitigations
4. Threats to Authentication Protocols and Mitigations
5. Types of Authentication Mechanisms
6. Best Practices for voting systems

3.1.1 Threats to Credential Issuance Methods and Mitigations

The issuance process is used by the users to establish trusted relationships with the authentication system and to obtain their authentication tokens[1]. The following subsections are examples of issuance mechanisms. Any gathered registration information (e.g., driver's license number, passport number, financial account information) should be protected as Personally Identifiable Information (PII) while in transit and while stored in the systems. The decision to store or delete this PII needs to be made based on the need to balance the protection of PII and the requirement to provide a basis for the legitimacy of voter registration records. For a more detailed discussion of PII protection, see section 3.3.

The following table provides a summary of threats to the credential issuance process and approaches to mitigate those threats.

Table 1: Threats to Credential Issuance Mechanisms and Mitigations

Threat/Attack	Threat Mitigation Mechanisms
Impersonation of claimed identity	In-person identity proofing by trusted party and the user providing Government issued photo IDs such as driver's licenses and passports to prove his identity. Additional assurance can be achieved by the user supplying a current document (e.g., last month's gas bill) with their name and address on it.
Repudiation of issuance	Have the individual sign a form acknowledging issuance of the token.

[1] The term *issuance* in this document includes some elements, such as verification of an applicant's identity, which are often referred to as *registration*. However, to avoid confusion between the voter registration process and the registration process for issuing credentials, only the term *issuance* is used in this document.

Threat/Attack	Threat Mitigation Mechanisms
Disclosure of Token	Issue token in person, or by physically mailing it in a sealed envelope to a secure location, or through the use of a communication protocol that protects the confidentiality of the session data.
Physical Theft of Token	Issue token in person or by physically mailing it in a sealed envelope to a secure location or via continuously tracked mail (e.g., registered mail, Federal Express, etc.)
Voluntary Disclosure of Token	A user may disclose their token in order to sell their vote. There is little protection against this threat.
Tampering of Token	Issue credentials in person, by physically mailing storage media in a sealed envelope, or through the use of a communication protocol that protects the integrity of the session data. Establish a procedure that allows the user to authenticate the source of token (e.g., digital signature on electronic transmission)
Unauthorized issuance	Establish procedures to ensure that the individual who receives the token is the same individual who participated in the registration procedure. For example, issue token in person, or physically mail it in a sealed envelope to the address of record of the user.

3.1.2 Credential Issuance Methods

Jurisdictions may establish a trusted relationship with a user and issue authentication tokens in-person, remotely, or using a combination of methods. For example:

a) ***In-person Issuance-*** Under this approach, the user appears before a trusted party. The trusted party authenticates the user on the basis of antecedent relationship or photo identification cards (e.g., drivers' license, passport). The user is issued a credential on the basis of this identity proofing in-person, online, or out of band.

b) ***On-line Issuance-*** Under this approach, the user accesses the authentication system online and provides information unique to the user that is not widely-known (e.g., bank account number, credit card number, account balances, passport number, etc.) The authentication system validates the information from authoritative databases and issues a credential online.

c) ***Out-of-Band Issuance-*** Under this approach, the user accesses the authentication system online and provides information unique to the user that is not widely-known (e.g., bank account number, credit card number, account balances, passport number, etc.) The authentication system validates the information from authoritative databases and issues a credential to the user to their address of record.

For voters, authentication credentials can be issued in association with voter identification or some other individually unique data set. Or jurisdictions could rely on credentials issued by some other trusted authority, such as the Department of Defense Common Access Card.

3.1.3 Threats to Authentication Mechanisms and Mitigations

Once credentials have been issued, authentication mechanisms allow users to provide another party with some level of assurance that they are who they claim to be. The follow table identifies high-level threats to authentication mechanisms and strategies for mitigating these threats.

Table 2: Threats to Authentication Mechanisms and Mitigations

Token Threat/Attack	Threat Mitigation Mechanisms
Theft	Use a password, PIN or biometric authentication to the token itself. The token locks up after a number of consecutive failed activation attempts.
Duplication	Use tokens that are difficult to duplicate, such as hardware cryptographic tokens.

Token Threat/Attack	Threat Mitigation Mechanisms
Discovery	Use authentication protocols in which the token cannot be discovered. Examples include supplying the token information over a Transport Layer Security (TLS) tunnel or using protocols such as Secure Shell (SSH) or Simple Authentication and Security Layer (SASL) with approved cryptographic algorithms.
Eavesdropping	Use authentication protocols in which the token cannot be captured by eavesdroppers. Examples include supplying the token information over TLS or using SSH and SASL-type protocols with approved cryptographic algorithms. Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator. OTP and cryptographic protocols (e.g., client authenticated TLS) are examples of this.
Offline cracking	Use a token with a high entropy token secret. Long, randomly generated passwords and cryptographic keys with a security strength of 112 bits or higher are good examples.
Phishing or pharming	Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator. OTP and cryptographic tokens are good examples. Use tokens that generate authenticators based on randomly generated input or challenge from authentication system. Cryptographic protocols such as TLS, SSH, and SASL, when used with approved cryptographic algorithms, are good examples.
Social engineering	Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator. OTP and cryptographic tokens, when used with approved cryptographic algorithms, are good examples. Use tokens that generate authenticators based on randomly generated input or challenge from authentication system. Cryptographic protocols such as TLS and SSH are good examples.
Online guessing	Use a token with a high entropy token secret. Long, randomly generated passwords and cryptographic keys with a security strength of 112 bits or higher are good examples. Use a token that locks after a number of repeated failed activation attempts.

3.1.4 Threats to Authentication Protocols and Mitigations

Some of the threats such as eavesdropping, phishing, pharming, and online guessing have been discussed above. The following table provides additional threats that arise for authentication protocols and how to mitigate those threats.

Table 3: Threats to Authentication Protocols and Mitigations

Authentication Protocol Threat/Attack	Threat Mitigation Mechanisms
Replay	Cryptographic protocols that use nonces, sequence numbers, or challenges. TLS is an example of such a protocol.
Session Hijacking	Cryptographic key derived from the authentication process is used to authenticate all session data (e.g., individual packets). TLS is an example of such a protocol. Note: Application-level concerns arising from session hijacking are mitigated by layering this authentication and following the practices outlined in section 5.10.

Authentication Protocol Threat/Attack	Threat Mitigation Mechanisms
Man-in-the-middle	Cryptographic protocols that protect the user from revealing information (e.g., authentication secret) to an attacker masquerading as the authentication system. Client authenticated TLS is an example of such a protocol; due to the mechanisms in the protocol, a masquerading party cannot make the user sign the appropriate secret to complete the man-in-the-middle attack. In the case of websites served over HTTPS, server side-only TLS is also protected from this threat so long as the user is not deceived into using an attacker's Uniform Resource Locator (URL). Commercial products will warn users of this deception so long as no certification authority trusted by the user acts improperly by issuing a certificate to an attacker attempting to pose as the legitimate authentication system.

3.1.5 Types of Authentication Mechanisms

Authentication mechanisms are broken down in two broad categories: token based and biometric. Material developed in this section is based on [SP800-63]. That document, particularly Sections 7 and 8, may be consulted for additional background and technical information.

3.1.5.1 Token Based Authentication

Token based authentication relies on the user demonstrating possession and control of something that can be used to establish identity. This can incorporate one or more of three factors: something the user has, something the user knows, or something the user is. The system uses an authentication protocol to validate the user's possession and control of the token. There are various types of tokens that may be used depending on the capabilities and assurance requirements of the system authenticating the user. These are described in detail below.

a) ***Memorized Secret Tokens-*** Using memorized secret tokens, users prove their identities by providing a secret known to them and verifiable by the authentication system. Passwords and Personal Identification Numbers (PINs) are good examples of memorized secret tokens. This secret needs to be established during the user registration process. User Identifier (ID) and password for a computer account, or a PIN for unlocking a cryptographic token are examples of memorized secret tokens. The advantages of the memorized secret tokens are ease of use and wide availability in commercial products. Disadvantages of this approach and their corresponding mitigations, where possible, are listed below:

   i) The token can be revealed to unauthorized parties during token issuance. This threat can be mitigated by issuing the token using a protected channel such as in-person hand-off or sending the token to an address of record via continuously tracked mail; or protecting the electronic communication channel used for token issuance.

   ii) The token can be revealed via "shoulder surfing" while being presented (entered or typed in) for authentication. This threat can be mitigated by not echoing the token when it is entered.

   iii) The token is written down and hence can be accessed by unauthorized parties. This threat can be mitigated by memorizing the token or by protecting the written down value.

   iv) The token can be obtained by eavesdropping during the authentication process. This threat can be mitigated by cryptographically protecting the authentication channel or by using authentication protocols that prove the possession of the token without revealing it.

   v) An unauthorized party can use manual or automated means to authenticate by providing values for the token until authentication succeeds (e.g., performing an online dictionary attack). This threat can be mitigated by locking the account after a small number of unsuccessful authentication attempts, or by introducing a delay between unsuccessful authentication attempts that increases after each failure.

   vi) An attacker can mount an offline dictionary attack by eavesdropping on the protected protocol. This threat can be mitigated by avoiding protocols that are susceptible to offline dictionary attacks. Users are generally incapable of generating or remembering passwords that are strong enough to prevent an offline dictionary

attack (17 randomly chosen characters,) and may compromise the security of passwords by writing them down.

vii) The legitimate token owner can provide the token to someone else in a vote buying scheme or can be tricked into sending the password to a party impersonating the legitimate voting system. There is no easy mitigation to this threat.

viii) Malware on a user's computer can capture the token as it is entered by the user, and pass it on to an unauthorized party. Up-to-date and activated antimalware software can mitigate this threat on administered systems.

b) Pre-registered Knowledge Token- Under this authentication approach, a user establishes a set of questions and answers during the user registration process with the authentication system. In order to be effective, questions and answers should be easy for the user to recall from memory, and difficult for others to obtain or guess. Authentication is based on the accuracy of the responses provided by the user. An example of a Pre-registered Knowledge Token would be a question such as "What was the first car you ever owned?" and requiring the answer to contain the year, make, model and color. Based on the accuracy of the responses supplied by the user, the authentication system determines if the attempt is successful or not. Another example is asking the user to select an image or set of images that the user memorizes during the registration phase; the user then has to identify the correct images from a set(s) of similar images. Note that pre-registration is different from Knowledge Based Authentication (KBA); in KBA the answers are verified by querying a database containing information about the user. The advantages of the pre-registered knowledge tokens are ease of recall, ease of use and wide availability of commercial implementations. Disadvantages of this approach and their corresponding mitigations, where possible, are listed below:

i) The token can be revealed to unauthorized parties during token registration. This threat can be mitigated by using a protected communication channel during the token registration.

ii) The token can be revealed via "shoulder surfing" while being registered or presented (entered or typed in) for authentication. This threat can be mitigated by not echoing the knowledge as it is input. In such a case, during input, the knowledge may need to be entered twice to protect against typing errors.

iii) The token can be obtained by eavesdropping during the authentication process. This threat can be mitigated by cryptographically protecting the authentication channel or by using authentication protocols that prove the knowledge of the token without revealing it.

iv) An unauthorized party can use manual or automated means to authenticate by providing values for the token until authentication succeeds (e.g., performing an online dictionary attack). This threat can be mitigated by locking the account after a small number of unsuccessful authentication attempts, or by introducing a delay between unsuccessful authentication attempts that increases after each failure.

v) An attacker can mount an offline dictionary attack by eavesdropping on the protected protocol. This threat can be mitigated by avoiding protocols that are susceptible to offline dictionary attacks.

vi) The knowledge which is prompted for could be discoverable by searching public records or social networking sites. Mitigation of this threat is difficult. That is why this mechanism is generally used as an added secondary authentication mechanism.

vii) The legitimate token owner can provide the token to someone else in a vote buying scheme, or can be tricked into sending the token to a party impersonating the legitimate voting system. There is no easy mitigation to this threat

viii) Malware on a user's computer can capture the token as it is entered by the user, and pass it on to an unauthorized party. Up-to-date and activated antimalware software can mitigate this threat on administered systems.

c) Look-Up Secret Token- Under this authentication approach, the user and the authentication system share one or more secrets that are held in a physical or electronic medium by the user. The user uses the token to look up the appropriate secret(s) that are needed to respond during authentication. For example, a user may be asked by the authentication system to provide a specific subset of the numeric or character strings printed on a card in table format. If the user is able to provide the correct response, the user is successfully authenticated. The shared secret(s) needs to be established during the user registration process. The advantages of look-up secret tokens are

that they are less susceptible to eavesdropping and to online and offline dictionary attacks. Disadvantages of this approach and their corresponding mitigations, where possible, are listed below:

i) The implementation of these tokens requires additional software and possibly hardware on the authentication server side, resulting in increased cost.

ii) If the token is hardware based, this further increases the overall cost.

iii) The tokens are not as easy to use as a static secret token.

iv) The tokens cannot be memorized and hence must be stored in hardware, software, or printed form.

v) The token can become unusable due to malfunction or availability. For example, hardware tokens can stop functioning, software list of secrets can get corrupted or become otherwise un-accessible, tokens can be misplaced or can become unreadable (e.g., due to fading or smudging).

vi) The token can be revealed to unauthorized parties during token issuance. This threat can be mitigated by issuing the token using a protected channel such as in-person hand-off, or sending the token to address of record via continuously tracked mail; or cryptographically protecting the electronic communication channel used for token issuance.

vii) The token can be obtained by eavesdropping during the authentication process, if the token secret space is limited (e.g., grids). This threat can be mitigated by cryptographically protecting the authentication channel.

viii) An unauthorized party can use manual or automated means to authenticate by providing values for the token until authentication succeeds (e.g., perform an online dictionary attack). This threat can be mitigated by locking the account after a small number of unsuccessful authentication attempts.

ix) The legitimate token owner can provide the token to someone else in a vote buying scheme, or can be tricked into sending the token to a party impersonating the legitimate voting system. There is no easy mitigation to this threat.

x) Malware on a user's computer can capture the token as it is entered by the user, and pass it on to an unauthorized party. Up-to-date and activated antimalware software can mitigate this threat on administered systems.

d) Out of Band Token- Under this authentication approach, a secret authenticator is transmitted from the authentication system to a physical device or system controlled by user. The communication channel for this transmission must be separate from the communication channel used for user authentication. The secret authenticator transmitted is valid for one time use and expires within minutes. An example of out of band token is as follows: a user attempts to log into a website and receives a password or PIN on his or her cellular phone, PDA, pager, or land line which the user must enter in the web session in order to be authenticated. Note that the user cellular phone, PDA, pager, or land line number is registered during the user registration process. The advantages of the out of band tokens are that they mitigate the threat of eavesdropping (attacker is less likely to succeed in eavesdropping two channels, particularly with the second one existing only for a very short duration), and thus, also protecting against successful online or offline dictionary attacks against the authentication secret. The disadvantages of this approach and their corresponding mitigations, where possible, are listed below:

i) The destination of the token (e.g., specific phone number) could be specified by the attacker during issuance. This threat can be mitigated by using a protected channel for token channel registration such as in-person hand-off or cryptographically protecting the electronic communication channel used for token channel registration.

ii) Most commercial products require enhancement or additional commercial products to implement the out of band tokens, resulting in higher costs.

iii) The user being authenticated requires the second channel. Not all voters may have access to a second.

iv) The legitimate token owner can provide the token to someone else in a vote buying scheme, or can be tricked into sending the token to a party impersonating the legitimate voting system. There is no easy mitigation to this threat, but using the second channel requires the voter to register another party's channel (resulting in possible detection during auditing) or to be present to cast a ballot.

v) Malware on a user's computer can capture the token as it is entered by the user, and pass it on to an unauthorized party. The one-time nature of these tokens requires a more sophisticated attack whereby the

malware must pass the captured token for immediate use by an attacker.

e) One Time Password (OTP) Device- this authentication approach, the user holds a hardware device that supports the spontaneous generation of one time passwords. The authentication system is synchronized with the hardware device. Authentication is accomplished by providing an acceptable one time password from the device. These devices themselves may or may not require biometric or password/PIN authentication in order to generate the one time password. The synchronization of the hardware device with the authentication system needs to be established during the user registration process. The advantages of OTP tokens are that they are not susceptible to online or offline dictionary attacks, and are not susceptible to eavesdropping. Disadvantages of this approach and their corresponding mitigations, where possible, are listed below:

i) The implementation of these tokens requires additional software and possibly hardware on the authentication server side, resulting in increased cost.

ii) The token is generally hardware based, adding to the cost.

iii) The token can be provided to unauthorized parties during token issuance. This threat can be mitigated by issuing the token using a protected channel, such as in-person hand-off or sending the token to address of record via continuously tracked mail.

iv) The token can be stolen. This can be mitigated by user vigilance, by adding a secret PIN or password to the OTP, and/or by using the device with a biometric. The biometric, secret PIN or password could be used in a variety of ways depending on the OTP implementation. It could be used to unlock the token, could be input to create the OTP, or could be simply appended to the OTP.

v) The token can become unusable due to malfunction.

vi) The token may be deemed difficult to use. If the token and the authentication server are out of synchronization, the protocol may automatically synchronize or may require the user to perform additional actions until the token is brought back in synchronization with the authentication server.

vii) The legitimate token owner can provide the token to someone else in a vote buying scheme, or can be tricked into sending the token to a party impersonating the legitimate voting system. There is no easy mitigation to this threat.

viii) Malware on a user's computer can capture the token as it is entered by the user, and pass it on to an unauthorized party. The one-time nature of these tokens requires a more sophisticated attack whereby the malware must pass the captured token for immediate use by an attacker.

f) Cryptographic Token- Under this authentication approach, a cryptographic key token is held by the user. The token could be hardware based (e.g., a smart card or Universal Serial Bus (USB) form factor cryptographic module) or could be software based (e.g., CD or USB storage device). Furthermore, the token could perform functions with or without local authentication. Local authentication could be biometric or password/PIN based. Authentication is accomplished by proving possession of the cryptographic key by performing a cryptographic key based operation during an authentication protocol (e.g., challenge – response). For example, a public, private key token is held by the user and the user performs a digital signature on a random challenge from the authentication server. User authentication via client-authenticated TLS is an example of such protocol. The association of cryptographic key with the user needs to be established during the user registration process or using other means such as Public Key Infrastructure (PKI). The advantages of the cryptographic tokens are that they are not susceptible to online or offline dictionary attacks, and are not susceptible to eavesdropping. Disadvantages of this approach and their corresponding mitigations, where possible, are listed below:

i) The implementation of these tokens requires additional software on the authentication server side, but this is not a significant disadvantage since the software is part of commercial products and comes bundled in resulting in no added cost except for requiring some additional time to configure the system.

ii) If the token is hardware based, it adds to the cost.

iii) The token can be provided to unauthorized parties during token issuance. This threat can be mitigated by issuing the token using a protected channel such as in-person hand-off, sending the token to address of record via continuously track mail, or providing the token in a protected communication channel.

iv) The token can be stolen. This can be mitigated by user vigilance, by adding a secret PIN to the token, and/or by using the token with a biometric. The secret PIN or biometric can be used to unlock and use the token.

v) The token can become unusable due to malfunction.

vi) The legitimate token owner can provide the token to someone else in a vote buying scheme. There is no easy mitigation to this threat.

vii) Malware on a user's computer can capture the cryptographic token, and any tokens entered by the user to unlock the cryptographic token, and pass it on to an unauthorized party. Hardware-based cryptographic tokens can significantly mitigate this threat.

3.1.5.2 Biometric Authentication

Under the biometric authentication approach, the user is authenticated based on one or more intrinsic biological traits such as fingerprint, iris, face, voice, palm, or other characteristics that cannot be forged. Such systems do not provide perfect authentication since there are always false positives in which another person's biometric information is deemed to match that of the user, or false negatives in which a legitimate user's information is rejected due to an error in scanning the biometric data. In addition, physical handicaps can prevent an individual from using a biometric authentication mechanism, for example, an amputee may not have fingerprints. Biometric mechanisms are also vulnerable to capture and replay attacks unless compensating means such as cryptographic and "liveness" properties (such as a nonce or a challenge) are included to mitigate the capture and replay threat. These mechanisms are generally used only as a second factor (e.g., to unlock one-time password devices or cryptographic tokens). Furthermore, these mechanisms are generally used locally or to locally authenticate someone in the presence of a trusted individual (e.g., fingerprint scan in the presence of a guard while entering or exiting a secure facility)).

3.1.6 Best Practices for Voting Systems

The authentication mechanisms discussed above offer differing levels of assurance about the user's identity and carry differing associated costs. Furthermore, not all authentication mechanisms are feasible for all products. The security criticality of the various functions should be weighed against the cost inherent in and assurance provided by the available I&A options. [OMB0404] offers guidelines for considering the potential impact of authentication failures and the likelihood of that impact should a failure occur. Section 2.2 of [OMB0404] provides guidance on making the identified risks to the appropriate authentication assurance level. [SP800-63] offers technical guidance for mapping authentication mechanisms to the results of this assessment.

In assessing the risks associated with authentication failure in a UOCAVA system, it is helpful to consider three broad classes of users: administrative personnel, election officials and voters.

Administrative personnel require access to the system in order to install, configure and operate the software. These personnel are critical to the security of the system; should an unauthorized entity gain administrative control of the system, the integrity of the UOCAVA voting system could be compromised. This constitutes high harm to agency programs and public interests. One or more compromised administrative accounts could also lead to release of personal voter information to unauthorized parties on a large scale. As a result, administrative personnel should be authenticated in accordance with assurance level 4 in order to perform their duties, as [OMB0404] describes assurance level 4 as being "*appropriate for transactions needing very high confidence in the asserted identity's accuracy.*" Thus, in accordance with the guidance published in [SP800-63], in-person identity proofing should be required to register administrative personnel and a hardware cryptographic token over a secure channel should be used for authentication.

Election officials require access to the system in order to configure the voting application, conduct the election, and audit the results. These personnel are likewise critical to the security of the system; should an unauthorized entity improperly access the system and assume the role of an election official, integrity of the UOCAVA voting system could be compromised. This constitutes high harm to agency programs and public interests. As a result, election officials should also be authenticated in accordance with assurance level 4 in order to perform their duties. The same level of identity proofing and authentication control should apply to election officials as to administrative personnel.

Voters require much more limited access to the UOCAVA system. In a properly controlled system, compromise of a single voter account would lead to, at most, the compromise of a single vote. The limited impact of a compromised identity in this case suggests that authenticating voters should require at least assurance level 2 as described in [OMB0404]: "*Level 2 credentials are appropriate for a wide range of business with the public where agencies require an initial identity assertion (the details of which are verified independently prior to any Federal action).*" According to [SP800-63], level 2 credentials include a password sent over a secure channel. However, a higher assurance level would be needed to mitigate phishing, man-in-the-middle, and certain malware attacks.

In all three cases, the secure channel employed should be TLS with a cipher suite that provides 112-bit security or greater and where the X.509 certificates are validated according to the algorithm in [RFC5280]. TLS should perform mutual authentication for administrative access, and perform at least server-side authentication for voters connecting to the system.

3.2 Access Control

Access control technology deals with providing access to the stored information such as files, directories and functions to authorized users and denying that access to others. I&A and Access Control go hand in hand. I&A is performed in order to gain assurance of the user's identity. Once the identity of the user is established, an access control decision based on this authenticated identity appropriately enforces the system access control policy. Thus, first performing I&A, and then performing access control based on authenticated identity are required to enforce the security of an Information Technology (IT) system. The primary audience for this section is voting system designers.

Access control mitigates the threat of unauthorized actions such as access to or modification of the data or attempting to perform unauthorized functions. If administrative actions are not properly controlled, the security controls of the entire voting system can be defeated by the person who can bypass administrative access controls. The compromise may include the unauthorized person determining the outcome of the election. If voter actions are not properly controlled, any of the following can be compromised: voter personally identifiable information, voter election choices, and unauthorized vote casting.

Protection of information in transit is dealt with using technologies such as cryptography and protected communication links and is discussed elsewhere in this document.

The remainder of this section is divided into the following subsections:

1. Types of Access Control Mechanisms
2. Threats to Access Control Mechanisms
3. Best Practices for Voting Systems

3.2.1 Types of Access Control

The following are examples of access control mechanisms:

1. Discretionary Access Control (DAC)
2. Role Based Access Control (RBAC)
3. Privilege/Attribute Based Access Control (PBAC/ABAC)
4. Mandatory Access Control (MAC)
5. Type Enforcement
6. Capability Based Access Control (CBAC)

3.2.1.1 Discretionary Access Control (DAC)

DAC is the mechanism where the owner or the creator of the information determines who can have what type of access to the information.

The type of access is also termed "access mode" and refers to the types of operations that can be performed on the information or the object containing the information. Examples of types of operations that may be protected with DAC include: read, write, execute (for program files), search (for directories/folders), list (for directories/folders), etc.

DAC is widely implemented in today's commercially available operating systems such as Unix, Linux, and Windows.

Unix protection bits are an example of DAC. Each file or directory has 3 sets of bits, each set containing 3 bits for a total of nine bits. One set of bits represents permissions for the individual owner of the object (read, write, or execute[2]). The

[2] Read permission for a directory is interpreted as the ability to list the contents of a directory. Write permission to a directory is interpreted as ability to create files and subdirectories underneath the directory. Execute permission for directory is interpreted as the ability to search the directory.

second set of bits represents permissions for the owning group of the object[3]. The third and final set of bits represents permissions for all other users and groups.

Another example of DAC in wide use in operating systems is Access Control List (ACL). Conceptually an ACL is a collection of Access Control Entries (ACEs). Each ACE contains a user, group, or role name and access mode. In some implementations even delegation is supported by either including access modes for delegation in ACE or by having special ACE entries for delegation.

Fine-grained DAC is supported using similar concepts at application level data in many commercial products. For example, Relational Data Base Management System (RDBMS) can support ACL for rows, columns, tables, views, stored procedures, etc.

3.2.1.2 Role Based Access Control (RBAC)

RBAC is similar to DAC except that an individual's role dictates what information or functions that individual can access. The RBAC is defined by roles and the permitted operations for a role on a given object. Thus, conceptually, RBAC can be viewed as (and can be implemented using) ACL and ACE, where subject of the entries is a role rather than named users or groups. For example, in a system that implements RBAC, only users assigned the role of "auditor" might be permitted to read audit log entries, and only users assigned the role of "registration authority" would be permitted to create new users.

RBAC can be made hierarchical by adding relations for supporting role hierarchies where a role has all the authorizations of all the subordinate roles.

RBAC is implemented using DAC in commercial operating systems and in RDBMS.

RBAC can be viewed as a DAC mechanism if the object owner determines to share the object based on role.

RBAC can be viewed as MAC if the system makes the determination to share the object based on role instead of the object's owner/creator.

3.2.1.3 Privilege/Attribute Based Access Control (PBAC/ABAC)

PBAC is akin to RBAC except that privileges are atomic rights. A role can be viewed as collection of privileges. Access control for data and functions is implemented using PBAC in commercial operating systems and in RDBMS.

3.2.1.4 Mandatory Access Control (MAC)

MAC is also called label-based access control. It is termed mandatory because the inputs for the access control policy are system determined and are not at the discretion of the object's owner/creator. Objects and user sessions are assigned security labels by the system, and access decisions are enforced based on the compatibility of these labels. Not many commercial products offer MAC. For a more detailed explanation of MAC, see [TCSEC].

3.2.1.5 Type Enforcement

Type enforcement is another form of mandatory policy. The policy is enforced based on "domain definition" table. A "domain definition" table consists of rows representing domains of execution and types representing object type and cells consisting of "access mode". In order for a process to perform an operation on an object, the cell representing the execution domain of the process and object type is examined to determine if the "access mode" representing the operation is permitted.

There are not many commercial products offering type enforcement.

3.2.1.6 Capability Based Access Control (CBAC)

CBAC consists of the object owner obtaining the object capability (e.g., a handle or random number) when the object is created. The object owner can pass this capability to others. Thus having the object access information is an implicit right to access the object.

There are not many commercial products offering CBAC.

[3] Owning group is defined as the group the user session was invoked with when the object was created.

3.2.2 Threats to Access Control Mechanisms

The following table provides a summary of threats to the access control mechanisms and approaches to mitigate those threats.

Table 4: Threats to Access Control Mechanisms

Threat/Attack	Threat Mitigation Mechanisms
Access control modified by the user	Access control is implemented in a protected operating system
Access control bypassed by the user	All object access is mediated by the operating system so that the operating system can enforce the access control policy
Fine-grained application-based access exploited by the user to gain greater access	Application control fine-grained objects implemented using operating system object. These objects are under the control of the operating system and owning application only.
One application accessing another application's objects	Application control fine-grained objects implemented using operating system object. These objects are under the control of the operating system and owning application only.
Resource exhaustion covert channels against MAC	Use trusted application so that the channels cannot be exploited Audit the channels Eliminate the channel by sound design and by reducing resource sharing
Other storage channel attacks against MAC	Use trusted application so that the channels cannot be exploited Audit the channels Eliminate the channel by sound design and by reducing resource sharing
Timing channel attacks against MAC	Use trusted application so that the channels cannot be exploited Audit the channels Eliminate or reduce the channel capacity by using fixed time slices where possible.

3.2.3 Best Practices for Voting Systems

In general, the voting system designer should use an operating system and commercial applications that provide DAC. The voting system application should implement RBAC using these DAC facilities. Functions associated with the configuration, use and maintenance of the voting system application should be assigned to named roles, and these roles should be assigned to users or groups of users. Users should only be permitted to perform the functions associated with their roles when the role is active and the user has authenticated. So, for example, the role associated with the "register new voters" function might only be activated at certain times. The system should ensure that the role is both assigned to the user and active for the authenticated session.

3.3 Personally Identifiable Information (PII) Protection

The Government Accountability Office defines personally identifiable information (PII) as "any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual's identity, such as name, social security number, date and place of birth, mother's maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information."[GAO08536] Election authorities should consult relevant state and local laws to determine if there are governing definitions for PII in their jurisdiction.

Examples of PII include, but are not limited to:

- Name, such as full name, maiden name, or mother's maiden name.
- Personal identification number, such as social security number (SSN), passport number, driver's license number, or financial account number.

- Contact information, such as street address or email address.
- Personal characteristics, including photographic image, handwritten signatures, or biometric data.

Not all PII must be protected equally. Section 3 of NIST SP 800-122, *Guide to Protecting the Confidentiality of Personally Identifiable Information,* identifies six factors that organization should consider when determining the appropriate level of protection. Organizations should consider the following:

- How easily the PII can be tied to specific individuals.
- The number of individuals whose PII is stored in the system.
- The sensitivity of the data.
- The context of how the data will be used, stored, collected, or disclosed.
- Legal obligations to protect the data
- The location of the data, and level of authorized access to the data.

Further guidance on what constitutes PII, factors that influence PII sensitivity, and how PII should be handled from collection to destruction is provided in NIST SP 800-122, *A Guide to Protecting the Confidentiality of Personally Identifiable Information* [SP800-122]. The guidance in this section primarily applies to voting system designers and technical staff charged with protecting sensitive information on voting system equipment. The best practices outlined in this section should be used by election officials as a baseline for determining the appropriate controls to protect any PII stored by the jurisdiction. Based on the factors identified above, an organization may decide that additional protection is needed, or that some of the practices can be relaxed.

For the purpose of a voting system, PII identified in Section 3.3.1 is considered linked and highly sensitive. The rest of the guidance is formed on that basis.

The following subsections discuss the protection of PII:

1. Information Identified as PII
2. Threats to PII
3. Mechanisms for PII Protection while in Transit
4. Mechanisms for PII Protection while in Storage

For additional discussion of safeguards to protect the confidentiality of PII, see Sections 4 and 5 of [SP800-122].

3.3.1 Personally Identifiable Information (PII)

The following are examples of PII that may be found in a voting system:

1. Information in the voter registration database:
    a) Voter Name
    b) Voter Address
    c) Voter Contact information (e.g., phone number(s), e-mail address, etc.)
    d) Voter Political affiliation
2. Information used to verify voter identity during voter registration. Examples include one or more of the following:
    a) Driver's License Number
    b) Passport Number
    c) Bank Account Number
    d) Credit Card Number

3.3.2 Threats to PII

The following table details threats to PII along with possible mitigation mechanisms.

Table 5: Theats to PII

Threat to PII	Threat Mitigation Mechanisms
Unauthorized disclosure during transit	Encrypt the PII with FIPS validated encryption algorithm using appropriate key size so that only the authorized recipient can successfully decrypt the PII. Physically carry the PII or send it via physically protected paper mail.
Unauthorized modification during transit	Cryptographically protect the PII using FIPS validated algorithm using appropriate key size so that the recipient can verify the integrity of PII. Examples of cryptographic integrity protection are digital signatures, HMAC, or Cipher-based Message Authentication Code (CMAC) Physically carry the PII or send it via physically protected paper mail.
PII can be obtained by an attacker who gains access to a computer system where it is stored	Use a mix of computer security controls, firewalls, and IDS/IPS to deny attackers access to information including PII. Only store the PII that is required to be maintained. Only store the PII for the duration it is required. PII in storage can be cryptographically protected using FIPS validated algorithm, using a key that is stored off the system, or that must be unlocked with something stored off the system.
PII can be modified by an attacker who gains access to a computer system where it is stored	Use a mix of computer security controls, firewalls, and IDS/IPS to deny attackers access to information including PII. Store PII on non-rewritable media (e.g., Write-Once Read Many (WORM))
PII can be obtained by an unauthorized user of a computer system where it is stored	Use a mix of computer security controls provided by a secure operating system that requires identification and authentication. Use the access control mechanisms of the secure operating system to provide access to the PII.
PII can be modified by an unauthorized user of a computer system where it is stored	Use a mix of computer security controls provided by a secure operating system that requires identification and authentication. Use the access control mechanisms of the secure operating system to provide access to the PII. Configure the access control on PII to prohibit modification. Store PII on non-rewritable media (e.g., WORM)
Stored PII can be inappropriately accessed (viewed) by authorized personnel	Use a mix of computer security controls provided by a secure operating system that requires identification and authentication, and provides access control mechanisms. Use the access control mechanisms to restrict PII access to administrators. Require multi-person control for access to administrative accounts using a combination of technical and procedural controls. Examine event logs regularly to determine if PII is being accessed for unauthorized purposes by authorized users. Encrypt the PII and provide access to the decryption key to someone other than the person having access to PII.

Threat to PII	Threat Mitigation Mechanisms
Stored PII can be inappropriately modified by authorized personnel	Use a mix of computer security controls provided by a secure operating system that requires identification and authentication, and provides access control mechanisms. Use the access control mechanisms to restrict PII modification to administrators. Require multi-person control for access to administrative accounts using a combination of technical and procedural controls. Store PII on non-rewritable media (e.g., WORM) in an encrypted format.

3.3.3 Best Practices for Protection of PII in Transit

The voter PII in transit electronically should be secured using FIPS 140-2 validated cryptography, using FIPS algorithms, 112 bit security, and standardized Internet protocols. Examples of such mechanisms include:

1. TLS that is based on 2048 bit Rivest, Shamir, Adelman (RSA) certificates, using 3 key Triple Data Encryption Standard (TDES) and SHA-1[4] or SHA-2.
2. Internet Protocol Security (IPSec) that is based on 3 key TDES or Advanced Encryption Standard (AES) encryption and based on a 2048 bit Diffie Hellman (DH) Group for key exchange and 2048 bit RSA for end point authentication.

3.3.4 Best Practices for Protection of PII in Storage

Personally identifiable information should be provided only to the authenticated voter and other authorized individuals.

Personally identifiable information should be protected from unauthorized access and disclosure while it is stored in the voting system. At a minimum, the native operating system access control enforcement mechanism should be used to protect the voter PII storage container (e.g., file or database). These protection mechanisms should permit only authorized voting system applications access to the voter database. Additional application level DAC should be implemented so that only authorized users whose identity has been properly authenticated can access the voter PII. An example is a database with a DBMS that offers fine grained DAC based on tables, rows, columns, and views. The user authentication can be obtained from the underlying operating system or the DBMS can perform its own authentication. The user role is derived from the authenticated identity.

Given the Commercial-Off-The-Shelf (COTS) capabilities, administrators are likely to have access to the PII discussed above. One method mitigating the threat of abuse by administrators is to enforce separation of administrative duties. This could be accomplished with multi-person physical control to the system and administrative functions while prohibiting remote access. Note that multi-person administrative control can also be achieved by strictly limiting remote access to a workstation that is under the same multi-person physical control and has the following additional security controls:

1. The remote workstation has the same computer security controls as the voting system
2. The remote workstation is connected to no other networks but the voting system and uses FIPS validated, 112 bit security FIPS algorithms, Internet approved protocols (e.g., TLS, IPSec, etc.) to secure the communication channel between the remote workstation and the voting system.
3. The communication protocol used provides for mutual authentication, integrity and confidentiality.

[4] SHA-1-based HMAC is considered to offer security commensurate with the key size as opposed to 80 bits.

3.4 Confidentiality

If the confidentiality of information is not protected, it can lead to the compromise of PII (leading to identity theft, blackmail, embarrassment, etc.) or to a masquerading party obtaining information that can be used to authenticate as an administrator, election official, or voter. The masquerading in turn can lead to threats listed in Section 3.1. The primary audience of this section is system designers.

Table 6: Threats to Confidentiality

Threat to Confidentiality	Threat Mitigation Mechanisms
Information can be obtained during transit	Encrypt the information with FIPS validated encryption algorithm using appropriate key size so that only the authorized recipient can successful decrypt PII. Physically carry the information or send it via physically protected paper mail.
Information can be obtained by an attacker from a computer system where it is stored	Use a mix of computer security controls, firewalls, and IDS/IPS to deny attackers access to information.
Information can be obtained by an unauthorized user of a computer system where it is stored	Use a mix of computer security controls provided by a secure operating system that requires identification and authentication. Use the access control mechanisms of the secure operating system to provide access to the information.
Stored Information can be inappropriately accessed (viewed) by authorized personnel	Use a mix of computer security controls provided by a secure operating system that requires identification and authentication, and provides access control mechanisms. Use the access control mechanisms of the secure operating system to restrict access to the information. Encrypt the information and restrict access to the decryption key to someone other than the person having access to the information, effectively providing two person control. Regularly review event log for access events and rely on event log monitoring as a deterrent.

The following subsections discuss the confidentiality of information:

1. Information Requiring Confidentiality Protection
2. Confidentiality Mechanisms for Information in Transit
3. Confidentiality Mechanisms for Information in Storage

3.4.1 Information Requiring Confidentiality Protection

Voter PII protection has been addressed in Section 3.3. This section addresses confidentiality of other voter information.

The following are examples of information that require confidentiality protection:

1. Cast ballots should not be accessible to system administrators.
2. Event logs (both the operating system and application) should be accessible only by the administrators.
3. Passwords and private and secret keys should be protected from unauthorized access or use.

3.4.2 Best Practices for Confidentiality Protection of Information in Transit

Information requiring confidentiality protection which is electronically transmitted should be secured using FIPS 140-2 validated cryptography, using FIPS algorithms, 112 bit security, and standardized Internet protocols. Examples of such mechanisms include:

1. TLS that is based on 2048 bit RSA certificates, using 3 key TDES and SHA-1 or SHA-2. In TLS, each packet is encrypted using TDES or AES algorithm using a secret key that is securely established during the TLS connection formation.
2. IPSec that is based on 3 key TDES or AES for encryption, 2048-bit DH Group for key exchange, and 2048 bit RSA for end-point authentication. Authentication is based on either TDES Cipher Block Chaining (CBC) mode, SHA-1 based HMAC, or AES Counter with CBC Message Authentication Code (CCM) mode. DH is used to negotiate shared session key. The shared session key in turn is used for TDES or AES encryption of data.

3.4.3 Best Practices for Confidentiality Protection of Information in Storage

Information requiring confidentiality protection should be provided only to the authorized individuals.

Information requiring confidentiality protection should be protected from unauthorized access while it is stored in the voting system.

When the ballot information is no longer required, it should be erased. Such information may require retention to support audit, and federal law requires the retention of election-related data for 22 months. It is recommended that upon the close of the election such information should be archived, with archival access maintained under strict two person control, and the information deleted from the online system. Depending upon whether the system supports residual information protection and at what level of granularity, simple deletion may not be sufficient; erasure using commercial or custom products may be required.

Much of this information can also be protected using cryptographic mechanisms such as encryption. Such protection is of limited value in scenarios where decryption keys are stored with, or under the same controls as the information in question. These mechanisms are most effective when the information is stored on or transported to other media, and the decryption keys or the materials required to activate those keys are retained in a separate and secure place.

Given current COTS capabilities, system administrators are likely to have access to all of the information discussed above. One method mitigating the threat of administrative abuse is to provide for multi-person physical control to the system and administrative functions. Multi-person administrative control can be achieved either by permitting administrative functions from the system console or from a workstation that is under the same multi-person physical control and has the following additional security controls:

1. The remote workstation has the same computer security controls as the voting system
2. The remote workstation is connected to no other networks but the voting system and uses FIPS validated, 112 bit security FIPS algorithms, Internet approved protocols (e.g., TLS, IPSec, etc.) to secure the communication channel between the remote workstation and the voting system.
3. The communication protocol used provides for mutual authentication, integrity and confidentiality.

At a minimum, the event logs should be protected using the operating system DAC facilities.

Where applicable and feasible, the event logs should be protected using the application DAC[5].

Passwords need not be stored in the clear. Passwords should be stored in one-way encrypted form (e.g., fixed value encrypted with the password or hashed password) so that they cannot be deciphered even by the administrators. However, even if the passwords are stored in non-decipherable form, they must be protected using the operating system DAC so that no one can read the password. Users should be able to modify their own passwords using the operating system or application facilities. In addition, when applicable and feasible, application passwords should be protected using application DAC capabilities.

Secret and private keys should be protected in the FIPS 140-2 validated cryptographic modules. When the module is software based, the keys should be protected by the underlying operating system DAC. In addition, when applicable and feasible, application keys should be protected using application DAC capabilities.

3.5 Integrity

If integrity of information is not protected, it can lead to compromise of voting system. For example, unauthorized modification of stored PII can lead to an unauthorized person casting a vote. Modification to the event log can aid an

[5] For example, the operating system generated event log and RDBMS generated event log are protected by the operating system DAC. In addition, the RDBMS event log is protected by the RDBMS DAC.

attacker in covering his tracks. Unauthorized modification to system files or data can lead to the compromise of PII as well as the entire election; an attacker could undermine the election outcome. Unauthorized modification to passwords or keys can lead to the compromise of the authentication mechanism which in turn can lead to threats listed in Section 3.1.

The guidance in this section is primarily intended for voting system designers.

Table 7: Threats to Integrity

Threat to Integrity	Threat Mitigation Mechanisms
Information can be modified during transit	Cryptographically protect the information using FIPS validated algorithm using appropriate key size so that the receiving end can verify the integrity of information. Examples of cryptographic integrity protection are digital signatures, HMAC, or CMAC Physically carry the information or send it via physically protected paper mail.
Information can be modified by an attacker from a computer system where it is stored	Use a mix of computer security controls, firewalls, and IDS/IPS to deny attackers access to the information. Store the information on non-rewritable media (e.g., WORM)
Information can be modified by an unauthorized user of a computer system where it is stored	Use a mix of computer security controls provided by a secure operating system that requires identification and authentication, and provides access control mechanisms. Use the access control mechanisms of the secure operating system to restrict access to the information. Set the access controls on the information to prohibit modification. Store the information on non-rewritable media (e.g., WORM)
Stored information can be inappropriately modified by authorized personnel	Use a mix of computer security controls provided by a secure operating system that requires identification and authentication, and provides access control mechanisms. Use the access control mechanisms of the secure operating system to restrict access to the information. Use the access control mechanisms to restrict PII modification to administrators. Require multi-person control for access to administrative accounts. Store the information on non-rewritable media (e.g., WORM)

The following subsections discuss integrity-related topics:

1. Information Requiring Integrity Protection
2. Integrity Mechanisms for Information in Transit
3. Integrity Mechanisms for Information in Storage

3.5.1 Information Requiring Integrity Protection

The following are examples of information that require integrity protection:

1) PII as discussed in Section 3.3.1.
2) Ballot tracking information
3) Flags indicating whether an individual has voted or not.
4) Cast vote records.
5) Ballot/Election definition files.
6) Unmarked ballots.

7) Event logs (event logs may contain information that can be used to make inferences about voter activities).
8) All executable files.
9) All system data.
10) Passwords.
11) All cryptographic keys (private, secret and public keys).

3.5.2 Best Practices for Integrity Protection of Information in Transit

Information requiring integrity protection which is electronically transmitted should be secured using FIPS 140-2 validated cryptography, using FIPS algorithms, 112 bit security, and standardized Internet protocols. Examples of such mechanisms include:

1. TLS that is based on 2048 bit RSA certificates, using 3 key TDES and SHA-1 based HMAC[6]. SHA-1 based HMAC is applied to each packet, providing integrity to the packet and hence the data stream. HMAC secret key is securely established during the TLS.
2. IPSec that is based on 3 key TDES or AES encryption, 2048 DH Group for key exchange, and 2048 bit RSA for end-point authentication. Authentication Header (AH) is associated with each packet providing integrity. AH is calculated using either TDES CBC mode CMAC, SHA-1 based HMAC or AES CCM mode CMAC. DH is used to negotiate shared session key. The shared session key in turn is used CMAC.

The information listed in Section 3.5.1 in transit physically should be secured using continuously tracked mail; regular mail does not offer sufficient assurance of integrity.

3.5.3 Best Practices for Integrity Protection of Information in Storage

Information requiring integrity protection should be only provided to the voter and other authorized individuals.

Information requiring integrity protection should be protected from unauthorized modification while it is stored in the voting system. At a minimum, the native operating system DAC mechanism should be used to protect the voter information storage container (e.g., file or database). These protection mechanisms should only permit authorized voting system applications modify access to the voter database. Additional application level DAC should be implemented so that only authorized users whose identity has been properly authenticated can modify the voter information as described below. An example is the implementation of an RDBMS that offers fine grained DAC based on tables, rows, columns, and views. The user authentication can be obtained from the underlying operating system or the RDBMS can perform its own authentication. The user role is derived from the authenticated identity.

1. Certain records should only be modifiable by the user that owns them or an authorized authority acting on that user's behalf.
2. Other records should only be viewable by privileged roles, and then only at certain times.
3. Some records should not be modifiable under any circumstances.
4. Event logs (both the operating system and application) should not be modified by anyone except the operating system and application logging software.
    a) At a minimum, the event log integrity should be protected using the operating system DAC.
    b) Where applicable and feasible, the event log integrity should be protected using the application DAC[7].

Much of this information can also be protected using cryptographic mechanisms such as digital signatures, Message Authentication Code (MAC), HMAC, and hash. However, none of these mechanisms alone can protect the integrity of information while it is stored on the system since the adversary who can access the stored information can also access the keys to recalculate and update the integrity check. However, these mechanisms are useful when the information is stored or transported to other media and the integrity check parameters (e.g., public key, MAC or HMAC secret, or hash) are retained in a secure place.

[6] SHA-1-based HMAC is considered to offer security commensurate with the key size as opposed to 80 bits.

[7] For example, the operating system generated event log and RDBMS generated event log are protected by the operating system DAC. In addition, the RDBMS event log is protected by the RDBMS DAC.

Given the current capabilities of COTS, system administrators are likely to have access to all the information discussed above. The cost-effective way to ensure that the systems are implemented using commercial technology and protected from administrative abuse is to provide for multi-person physical control to the system and administrative functions. Note that multi-person administrative control can be achieved either by permitting administrative functions from the system console or from a workstation that is under the same multi-person physical control and has the following additional security controls:

1. The remote workstation has the same computer security controls as the voting system
2. The remote workstation is connected to no other networks but the voting system and uses FIPS validated, 112 bit security FIPS algorithms, standardized Internet protocols (e.g., TLS, IPSec, etc.) to secure the communication channel between the remote workstation and the voting system.
3. The communication protocol used provides for mutual authentication, integrity and confidentiality.

3.6 Availability

Successful denial of service attacks can prevent certain voters from being able to cast their ballots, which in turn can unduly impact the outcome of the election.

Table 8: Threats to Availability

Threat to Availability	Threat Mitigation Mechanisms
Natural Disaster such as fire, flood, earthquake	Use multiple sites. Fireproof site and computer room. Build site in area which is not in earthquake or flood zone. Computer room on upper floors. Install computers on raised floor and in racks.
Power Outage	Use multiple sites Use backup power (e.g., oil or gas operated generator)
Network Outage	Use multiple sites Procure redundant communication service from different service providers
Excessive Workload	Use multiple systems in load balanced configuration
Hardware Failure	Use multiple systems in load balanced configuration
Software or Data Loss	Perform frequent system backups
Denial of Service Attack	Use packet filters, firewalls and IDS/IPS to thwart attacks. Use capabilities of firewall and IDS/IPS to detect and anticipate denial of service attacks.

The guidance in this section is primarily intended for voting system designers.

The voting system data and functions will require high availability during the voting period. Denial of service attacks can compromise the voting functions. Two approaches are taken to ensure availability:

1. System Data Backup: Under this approach system data and files are backed up so that the system can be restored from data or file corruption; and/or
2. System Redundancy: Under this approach a hot, warm, or cold backup is available to take over if and when the system goes down.

These approaches are further described in the sections below.

3.6.1 System Data Backup

System data should be routinely backed up so that in case of system failure or data corruption, the backup can be used to restore the system. It is a good practice to perform incremental backup daily and full backup weekly.

In order to protect the integrity, confidentiality and availability of the data on the system, the backup media should be under the same multi-person system administrator control as most sensitive components of the voting system itself. See Section 6.2 for a description of additional operational controls which apply to the backup media.

Backups may be performed using one of the following mechanisms:

1. Local storage media such as tapes, Digital Video Disc (DVD), and Compact Disc (CD)
2. Backup to a central system over the communication line
3. Storage Area Network (SAN)

When backup data is sent over a communication line (e.g., for central backup or SAN synchronization) outside the secure Local Area Network (LAN), the following should be ensured to protect the data in transit:

1. FIPS validated cryptographic modules should be used
2. FIPS algorithms should be used
3. All cryptographic modules should use at least 112-bit security algorithms
4. Both ends of the communication should authenticate each other
5. Information should have confidentiality protection
6. Information should have integrity protection
7. Information should have anti-replay protection
8. Cryptographic protocol should be Internet standard

Client authenticated TLS with 2048 bit RSA certificates, 3 key TDES and SHA-1 is an example of the protocol that meets the above requirements.

Media that stores backup data should be maintained using operational controls equivalent to those used for media that stores live data, as described in section 6.

3.6.2 System Redundancy

The IT infrastructure used to support UOCAVA voting may contain redundant systems. If one system fails, the other system can take over. The redundant system can be any one of the following:

1. **Hot:** In this case, one or more systems share the operational workload with the primary system. In the case of the primary system failure, other system(s) take over. Generally, work is distributed across systems using load balancing hardware.
2. **Warm Standby:** In this case, a standby system is running and kept synchronized with the primary system. When the primary system goes down, the standby system takes over using automated detection or manual configuration.
3. **Cold Standby:** In this case, a standby system is powered down and requires manual configuration including loading the system backup tapes to bring up and operate the standby system.

In addition to redundancy within the design of the system itself, redundancy of hosting can provide additional robustness for functions that require continuous availability. For such systems, if the primary site can have a long term site failure due to natural disaster, power outage or communication failure, diverse sites should be used. A site is considered geographically diverse if the same incident will not cause failure at the secondary site when the primary site is hot with a failure (e.g., the two sites are not on the same weather pattern, on the same fault line, and same flood plain).

For any site, communication diversity should be achieved by procuring different communications lines from different communications service providers. The communications service providers must not share any of the following:

1. Facility
2. Communication trunks
3. Communication tail circuits
4. Communications service providers should either have backup power or should not share the same power utility provider.

When hot backup is used at geographically diverse sites, global load balancing hardware should be used to distribute the traffic among the diverse sites.

Note that all communications among the geographically diverse sites must be protected as listed in Section 3.6.1.

3.6.3 Best Practices for Availability of Functions

Some voting system functions are only used during an election cycle; high availability of the IT components that support them is only required during the election cycle. The election cycle is defined to begin with the pre-election time required to prepare the system for election and is defined to end with post election when the ballots have been counted.

During the election period, however, functions critical to the conduct of the election should be highly available.

The best practices for a voting system to provide a high degree of availability include all of the following:

1. Make sure that all software and firmware components (e.g., operating system, database, web server, applications, malware detectors) are running with the latest vendor patches.
2. Make sure that the malware detection software updates its signature database on a frequent basis (at least weekly).
3. Make sure that the malware detection software is executed on a regular basis (at least daily).
4. Make sure that all media introduced to the voting system (e.g., CD, USB, etc.) are scanned for malware.
5. Ensure that the firewalls only permit those services required to conduct the election, and any temporary ports opened for testing or other reasons are closed.
6. Ensure that the IDS/IPS execute with the latest signatures.
7. Conduct regular port scans on the system to identify open ports and available services.
8. Put an incident handling process in place as described in Section 6.9.
9. Store ballot information on Redundant Array of Inexpensive Disks (RAID) drives.
10. Use IDS/IPS to:
    a) Terminate offending sessions.
    b) Throttle bandwidth usage.
11. Use Network Behavior Analysis (NBA) IDS/IPS to identify threats that generate unusual traffic flows, such as Distributed Denial of Service (DDoS) attacks.

In addition to the above, the importance of the functionality provided by some IT systems will dictate additional redundancy to ensure continuous availability. Such systems should be hosted at facilities which provide for one or both of the following:

1. Use two or more sites for the systems. If more than one site is live, distribute traffic among the sites using geographical load balancers. Otherwise use automated or manual means to enable rapid failover from the primary site to a backup site in the event of an outage.
2. Use two or more voting systems at each site. If more than one system is live, distribute traffic between these using local load balancers. Otherwise use automated or manual means to enable rapid failover from the primary system to the backup system in the event of an outage.

3.7 Cryptographic Security

In this section we discuss the Public Key Infrastructure (PKI) Certification Authority (CA) requirements and requirements for cryptography and key management. The primary audience for this section is voting system designers.

3.7.1 Certification Authority (CA) Requirements

A PKI CA issues X.509 certificates to systems and personnel. These certificates serve to bind an asymmetric key pair to either a device or a user identity.

Although a dedicated CA could be deployed in conjunction with a voting system, it is not necessary or desirable to do so in most cases. The initial and ongoing costs associated with operating a dedicated CA are significant, both in terms of equipment and procedural overhead; these costs will not generally be offset unless the system being deployed requires an unusually large number of certificates. As long as the requirements specified in this section are met and all certificates along

with fresh revocation status information are accessible to the voting system, there is no security advantage to deploying a dedicated CA.

Most commonly, an existing enterprise or third party CA will be used to issue certificates that will be used by servers and personnel associated with the voting system.

Certificates issued to the voting system web servers and personnel should be issued by a CA that meets the following requirements:

1. The CA should perform identity proofing of the certificate applicant.
2. The CA should revoke a certificate if and only if an authorized party requests the certificate revocation.
3. Upon a certificate revocation, the CA should publish a Certificate Revocation List (CRL) in a timely fashion.
4. The CA should operate under personnel, physical, and procedural controls that are commensurate with those specified for the voting system in "Section 6 Operational Controls".
5. The CA should operate with computer security and network security controls that are commensurate with those specified for the voting system in Sections 4 and 5.
6. The CA should use FIPS 140-2 Level 3 or higher hardware cryptographic module for protection of the certificate and CRL signing private key.
7. The CA cryptographic module should be under two person control.
8. The CA should use the same private key to sign certificates and CRLs.

If a CA external to the voting system is used, its Certification Policy (CP) and Certification Practice Statement (CPS) should be examined in conjunction with the results of an independent audit to ensure that these requirements are met.

3.7.2 Certificate Checking

The voting system should perform TLS client authentication using certification path validation in full compliance with [RFC5280], including revocation checking.

The voting system should match the presented client certificate with the certificate registered for the claimant. The match should consist of the full certificate match.

The user should be advised to use a browser that performs certification path validation in compliance with [RFC5280], including revocation checking. The client browser should be configured for revocation checking. The following are examples of configuring revocation checking for two of the commonly used browsers:

- For Microsoft Internet Explorer (IE) use Tools → Internet Options → Advanced. Scroll down to "security" and check both "Check for publisher's certificate revocation" and "Check For server certificate revocation".
- For Mozilla Firefox use Tools → Options → Advanced. In the encryption tab, click "Validation" and check "Use the Online Certificate Status Protocol (OCSP)." If the voting system's PKI does not provide OCSP, administrators can click "Revocation Lists," import CRLs and check "Enable Automatic Update."

3.7.3 Cryptographic Algorithms

All cryptographic algorithms used should be FIPS approved. The algorithms and key sizes should be selected to provide 112 bit equivalent or greater security. All cryptographic modes of operations and schemes should be FIPS approved. All cryptographic algorithm implementations should undergo National Institute of Standards and Technology (NIST) Cryptographic Algorithm Validation Program (CAVP) and should receive a CAVP certificate. This ensures that the vendor's implementation conforms to the FIPS-approved security parameters and that this implementation will interoperate with others that have also been certified.

3.7.4 Cryptographic Module Engineering

All cryptographic modules should be validated to FIPS 140-2 Level 1 or higher. When cryptographic tokens are used by individuals, these should be hardware cryptographic modules and should be validated to FIPS 140-2 Level 2 or higher. CAs should use hardware cryptographic modules validated to FIPS 140-2 Level 3 or higher. All validated modules will receive a certificate from the NIST Cryptographic Module Validation Program (CMVP) and will have a published security policy. When operated in accordance with that security policy, validated modules meet the mandatory standards for the protection of sensitive data on Federal systems. Modules that have not been validated are considered to provide no protection to this data.

3.7.5 Best Practices for Managing Cryptographic Keys

The following guidelines should be used in managing long-term, static cryptographic keys. Ephemeral keys are managed in accordance with the cryptographic protocol that uses them.

1. The keys should be generated in FIPS validated cryptographic modules using FIPS approved method for the cryptographic algorithm(s) for which the key is intended.
2. The keys should be generated in the cryptographic module that is intended to use them, whenever possible and feasible. If this is not feasible, the keys should be transferred to the cryptographic module using FIPS approved methods, using FIPS approved algorithms, and using transport key sizes commensurate with the key being transported. The transfer mechanism should ensure integrity of the keys and confidentiality of the secret and private keys.
3. Cryptographic modules holding the keys should be protected at all times. Note that the cryptographic modules holding public keys also require protection to protect against substitution threat.
4. The keys should be changed every election cycle or every 3 years, whichever comes first.
5. The secret and private encryption keys used to protect stored data (as opposed to data in transit) and public key certificate, CRL, and Online Certificate Status Protocol (OCSP) signing keys should be backed up. The backup cryptographic module should meet all the security requirements of the operational cryptographic module.
6. Public keys should be archived based on the requirements to retain election information. This requirement applies to the extent that information is retained when digital signature need to be verified.
7. Private keys should be archived based on the requirements to retain election information. This requirement applies to the extent that information is retained in encrypted form and the private key is required for decryption.
8. Secret keys should be archived based on the requirements to retain election information. This requirement applies for the following:
    a) The information is retained in encrypted form and the secret key is required for decryption; or
    b) The information is retained and its integrity needs to be verified and the integrity is dependent on the secret key (e.g., HMAC or CMAC).
9. Secret and private keys should be reset to zero when no longer needed.

3.8 Communication Systems

This section is intended to provide guidance for securing external communications channels. These guidelines are intended for system administrators and system designers.

3.8.1 Email

This section was developed using NIST SP 800-45 *Guidelines on Electronic Mail Security*, [SP800-45], which should be consulted for background and additional details.

One or more dedicated platforms should be used for mail servers. The mail servers should have the following security controls:

1. The mail server should operate on a hardware platform dedicated to performing e-mail server and associated logging functions only.
2. The mail server should operate in a protected execution environment to protect itself from interference and tampering by other applications.
3. The platform should not permit any network based user login.
4. The platform should contain the minimum number of administrative accounts required for the mail server administration.
5. If the platform requires user accounts for mail access, the user accounts should not have any privileges. These should also apply to the administrators as mail recipients.
6. Administrative personnel should have separate user accounts as administrators and as mail recipients.
7. The appropriate and latest security template or hardening script should be applied to the server.

8. SMTP, Post Office Protocol (POP), and Interactive Mail Access Protocol (IMAP) service banners (and others as required) should be reconfigured so as not to report mail server and operating system type and version.
9. All dangerous or unnecessary mail commands (e.g., VRFY and EXPN) should be disabled.
10. The platform should be configured to execute the mail server application with a user account with the least privilege required.
11. The mail server application should limit user access to information that the user is authorized to access.
12. The mail server application should only write to the files and directories in areas dedicated for the mail server operational data. These areas should not include system files and mail server application files.
13. All users should be properly identified and authenticated.
14. Administrative accounts should require logon as described in Section 3.1.6. In addition, remote administration is strongly discouraged.
15. The platform should have only the mail server and associated logging applications installed.
16. Only those network services that are required for operation of the mail service should be installed and active. All other network services should be either not installed or disabled.
17. The mail server log should be protected from unauthorized examination and modification. The mail server log should be treated like the operating system log discussed in Section 5.4 to ensure that the mail server log cannot be used to compromise PII.
18. If inbound mail is required:
    a) Server-based malware scanning should be deployed.
    b) All attachments should be removed prior to delivery. If attachments absolutely must be allowed, all of the following should be done:
        i. Attachments that are known to be executable once decoded such as .exe .msi .com .mde, .cer, etc. should be deleted or quarantined.
        ii. Other attachments should be scanned for virus and harmful macros.
        iii. The maximum allowable attachment size should be determined; attachments above a certain size should be rejected.
    c) Server-based content filtering should be deployed.
    d) Appropriate bounce or non-delivery notice should be provided for rejected mail, unknown recipients, removed attachments, etc.
19. The mail server should reject mail from known blacklisted mail servers.
20. The mail server should relay mail from only known internal voting system IP addresses.
21. The mail server should relay mail from only authenticated users.
22. The mail server should abide by the following network architecture principles:
    a) The mail server should not be placed on the protected voting system sub-network unless it is further protected by a mail gateway. The mail gateway in turn should be in a DMZ protected from the Internet by a firewall.
    b) The mail server may be placed in a DMZ protected from the Internet by a firewall.

3.8.2 Fax and Telephone PBX

The guidelines listed below were developed using NIST SP-800-24, *PBX Vulnerability Analysis: Finding Holes in Your PBX Before Someone Else Does*, [SP800-24], which should be consulted for background and additional details.

The PBX should use the following security features:

1. Remote maintenance access should be normally blocked unless unattended access is required.
2. Local personnel involvement should be required to open remote maintenance ports when remote access is required for troubleshooting. Thus, remote maintenance cannot be enabled from remote location.

3. Two-factor strong authentication should be used on remote maintenance ports. For example, one factor can be a smart card or one-time password, and the other factor can be traditional password.
4. Maintenance ports should be physically protected from unauthorized access.
5. Password for Private Branch Exchange (PBX) accounts:
    a) Should be automatically generated
    b) Should be randomly generated
    c) Should have entropy of 64 bits
6. If fax line goes through PBX, it should use a dedicated line.

4 Voting System Network Protections

For the voting system, network protection should use a multi-layered approach by incorporating a firewall to prevent remote network-based attacks along with IPS/IDS for attack attempts that are not stopped at the firewall.

In architectures where the workstations used by election officials and administrators are on a separate network from the servers, the workstation network should use the same controls as the voting system network. In most cases, the controls for the workstation network can be more restrictive than those for the network that contains the servers, as the workstations will not generally require that external systems be permitted to access them over the network in order to provide voting application functionality.

The following sections describe the network security technologies.

4.1 Firewall

This section was developed using NIST SP-800-41 Revision 1, *Guidelines on Firewalls and Firewall Policy* [SP800-41], which should be consulted for background and additional details.

4.1.1 Firewall Types

Firewalls are devices or programs that control the flow of network traffic between networks or hosts that employ differing security postures. There are several types of firewalls, each with varying capabilities to analyze network traffic and allow or block specific instances by comparing traffic characteristics to existing policies. These types are listed below:

1. Packet Filtering Firewall
2. Stateful Inspection Firewall
3. Application-Proxy Gateway
4. Circuit-Level Gateway
5. Dedicated Proxy Server

The following subsections describe each of these firewall types.

4.1.1.1 Packet Filtering Firewall

The most basic feature of a firewall is to filter the incoming and outgoing traffic based on one or more of the following:

1. Source Internet Protocol (IP) Address
2. Destination IP Address
3. Port Number
4. Direction (Inbound or Outbound)
5. Network Protocol (e.g.., Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP))

Unlike more advanced filters, packet filters do not protect the content of packets. Their access control functionality is governed by a set of directives referred to as a ruleset defined in terms of the 5-tuple listed above. Packet filtering capabilities are built into most operating systems and devices capable of routing. Firewalls that are only packet filters and provide no advanced features have two main strengths—speed and flexibility. Since packet filters seldom examine data above the network layer (with the possible exception of limited transport layer information), they can operate very quickly. And because most modern network protocols can be accommodated via the network layer and below, packet filters can be used to provide some security for nearly any type of network communication or protocol. The boundary router in the diagram below can be configured as a packet filtering firewall.

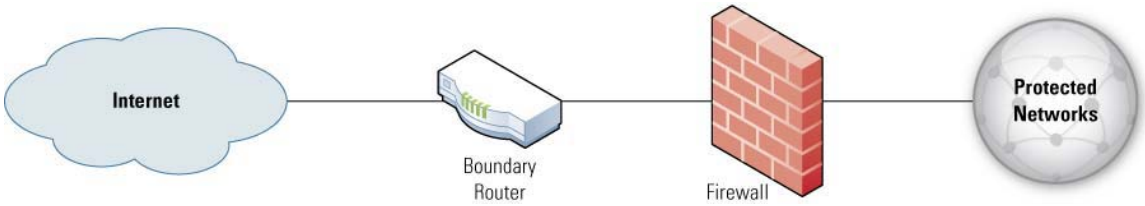

Figure 1. Boundary Router and Firewall

4.1.1.2 Stateful Inspection Firewall

Stateful inspection improves on the functions of packet filters by tracking the state of connections and blocking packets that deviate from the expected state. This is accomplished by incorporating greater awareness of the transport layer. As with packet filtering, stateful inspection intercepts packets at the network layer and inspects them to see if they are permitted by an existing firewall rule, but unlike packet filtering, stateful inspection keeps track of each connection in a state table. While the details of state table entries vary by firewall product, they typically include source IP address, destination IP address, port numbers, and connection state information. Each new packet is compared by the firewall to the firewall's state table to determine if the packet's state contradicts its expected state. For example, an attacker could generate a packet with a header indicating it is part of an established connection, in order to pass through a firewall. If the firewall uses stateful inspection, it will first verify that the packet is part of an established connection listed in the state table. A deeper inspection of the packet may also be conducted. The packet can be analyzed at the network, transport, and application protocol layers to compare firewall-configured profiles of benign protocol activity against observed events to identify deviations. This enables the identification of unexpected sequences of packets, such as issuing the same command repeatedly or issuing a command that was not preceded by another command on which it is dependent. These suspicious commands often originate from buffer overflow attacks, Denial of Service (DoS) attacks, malware, and other forms of attack carried out within. Another common feature is reasonableness checks for individual commands, such as minimum and maximum lengths for arguments. For example, a username argument with a length of 1000 characters is suspicious—even more so if it contains binary data.

4.1.1.3 Application-Proxy Gateways

An application-proxy gateway combines lower layer access control with upper layer functionality. These firewalls contain a proxy agent that acts as an intermediary between two hosts that attempt to establish communications with each other, and never allows a direct connection between the two hosts. Each successful connection attempt actually results in the creation of two separate connections—one between the client and the proxy server, and another between the proxy server and the true destination (shown in Figure 2). The proxy is transparent to the two hosts, and a direct connection seems to have been established. Because external hosts only communicate with the proxy agent, internal IP addresses are not made known to the outside world. The proxy agent interfaces directly with the firewall ruleset to determine whether a given piece of network traffic should be allowed to transit the firewall. In addition to the ruleset, each proxy agent has the ability to require authentication of each individual network user. This user authentication can take many forms, including user ID and password, hardware or software token, source address, and biometrics.

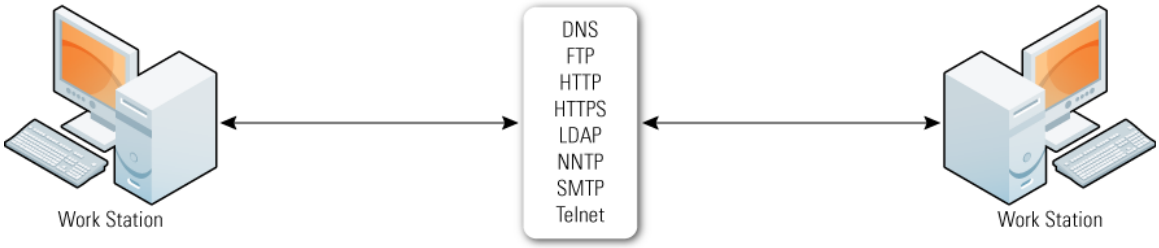

Figure 2. Proxy Gateways

The proxy gateway operates at the application layer and can inspect the actual content of the traffic. Unlike stateful protocol analysis, which mainly verifies that traffic is consistent with protocol definitions, application-proxy gateways break down the data and more thoroughly examine packet content, distinguishing between normal traffic for a specific protocol and traffic that could contain exploits for known flaws. The proxy gateways also perform the TCP handshake with the source system and are able to protect against exploitations at each step of a communication. In addition, proxy gateways can make decisions to permit or deny traffic based on information in the application protocol headers or payloads.

4.1.1.4 Circuit-Level Gateways

A circuit-level gateway is another type of proxy, and is sometimes referred to as a circuit-level proxy. In addition to their proxy capabilities, which shield internal systems from the outside world, circuit-level gateways validate each connection before it is established in a manner similar to that of stateful inspection. When a connection request is received, the circuit-level gateway checks its ruleset to determine if the connection should be allowed. In addition to the 5-tuple discussed in Section 4.1.1.1, some circuit-level gateways can also base their rulesets on user authentication or time restrictions.

Once a connection is permitted, an entry is placed in a virtual circuit table that also contains state information. Packets listed in the table are allowed to pass through the firewall without further validation. When the connection has been terminated or has been inactive for a pre-determined period of time, the entry is removed from the table. A circuit-level proxy provides many of the same features as a firewall that has stateful inspection, with the added functionality of a proxy to prevent direct connections between hosts on opposite sides of the firewall. Circuit-level gateways are usually faster than application-proxy gateway firewalls because they perform fewer evaluations on the data; they do not examine the content of the application packets.

4.1.1.5 Dedicated Proxy Servers

Dedicated proxy servers differ from application-proxy and circuit-level gateways; while they retain proxy control of traffic for one or more applications, they do not have firewalling capabilities. Although dedicated proxy servers are not firewalls, they work closely with application-proxy gateway firewalls and circuit-level gateway firewalls. Because these servers do not have firewall capabilities, they are typically deployed behind traditional firewall platforms. Typically, a main firewall could accept inbound traffic, determine which application is being targeted, and hand off traffic to the appropriate proxy server (e.g., email proxy). The dedicated proxy server would perform filtering or logging operations on the traffic, and then forward the traffic to internal systems. A proxy server could also accept outbound traffic directly from internal systems, filter or log the traffic, and pass it to the firewall for outbound delivery. An example of this is an HTTP proxy deployed behind the firewall; users would need to connect to this proxy en route to connecting to external Web servers. Dedicated proxy servers are generally used to decrease firewall workload and conduct specialized filtering and logging that might be difficult to perform on the firewall itself.

The inbound proxy servers are not used because these proxy servers must mimic the capabilities of the real server that they are protecting, an activity which becomes nearly impossible when protecting a server with many features. Using a proxy server with fewer capabilities than the server it is protecting renders the non-matched capabilities unusable. Additionally, the essential features that inbound proxy servers should have (logging, access control, and so on) are usually built into the real servers. Most proxy servers now in use are outbound proxy servers, with the most common being HTTP proxies. The figure below illustrates a typical network architecture where a DMZ is protected from the Internet using a filtering router, the Demilitarized Zone (DMZ) contains dedicated proxy servers for HTTP and SMTP and the Intranet is further protected using a stateful inspection firewall.

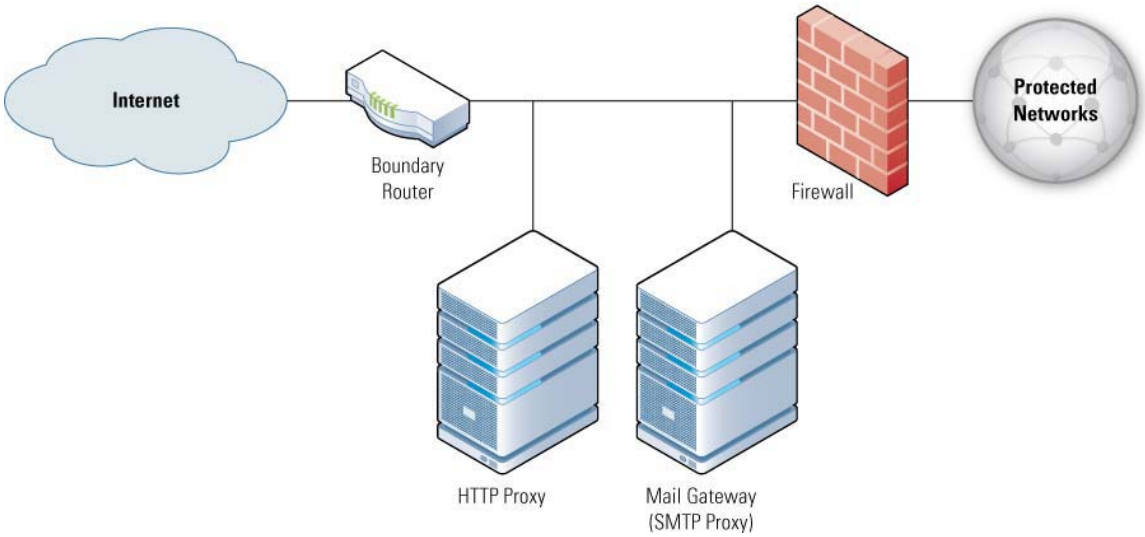

Figure 3. Dedicated Proxy Servers

4.1.2 Best Practices for Voting Systems

One or more dedicated firewall platforms (in addition to the host-based software firewall discussed in "Section 5.5 Host-Based Firewall") should be used. The firewall should have the following security controls:

1. The firewall should operate on a hardware appliance dedicated to performing firewall and associated logging functions only.
2. The firewall should operate in a protected execution environment to protect itself from interference and tampering by other applications.
3. The platform should not permit any network based user login.
4. The platform should contain the minimum number of administrative accounts required for the firewall administration. This can, and should, include separate administrative accounts for each individual administering the firewall. The platform should not contain any other user accounts.
5. The platform should have only the firewall and associated logging applications installed.
6. The platform should only have the network services installed and active that are required for handling the ports and protocols permitted through the firewall. All other network services should be either not installed or disabled.
7. The firewall log should be protected from unauthorized examination and modification.

The firewall may permit outbound Domain Name Service requests, and their corresponding replies, to registered, authorized and trust Domain Name Server servers. The firewall may optionally permit Network Time Protocol (NTP) outbound to a registered, authorized, and trusted time server if and only if time synchronization is done automatically.

The firewall may permit outbound SMTP from the mail server.

All other protocols should not be permitted in or out, except any other protocols required to perform election-related functions.

A recommended notional network architecture for the voting systems and workstations is described in Section 4.4.

4.2 Intrusion Detection System

This section was developed using NIST SP 800-94, *Guide to Intrusion Detection and Prevention Systems* [SP800-94], which may be consulted for background and additional details.

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. Intrusion prevention is the process of performing intrusion detection

and attempting to stop detected possible incidents. Intrusion detection and prevention systems (IDS/IPS) are used for the following purposes:

1. Identifying possible security incidents
2. Logging information about security incidents
3. Attempting to stop security incidents
4. Reporting security incidents to security administrators.
5. Identifying problems with security policies
    a) Violations of the security policies
    b) Need to change security policies
    c) Deter individuals from violating security policies
6. Documenting current threats

An IDS/IPS cannot provide completely accurate detection; it generates false positives (incorrectly identifying benign activity as malicious) and false negatives (failing to identify malicious activity). Thus an IDS/IPS must be tuned so that false negatives are decreased. This may lead to increase in the false positives, which necessitates additional analysis resources to differentiate false positives from true malicious events.

The following topics are of interest for IDS/IPS:

1. IDS/IPS Detection Methods
2. IDS/IPS Technologies
3. Components of IDS/IPS
4. IDS/IPS Functions
5. Securing IDS/IPS
6. Best Practices for IDS/IPS Voting Systems

The following subsections discuss each of these topics.

4.2.1 IDS/IPS Detection Methods

An IDS/IPS uses one or more of the following detection methodologies:

1. Signature-based Detection
2. Anomaly-based Detection
3. Stateful Protocol Analysis

The following subsections describe each of these techniques.

4.2.1.1 Signature-based Detection

Signature-based detection compares known threat signatures to observed events to identify incidents. This is very effective at detecting known threats but largely ineffective at detecting unknown threats and many variants on known threats. Signature-based detection cannot track and understand the state of complex communications, so it cannot detect most attacks that comprise multiple events.

4.2.1.2 Anomaly-based Detection

Anomaly-based detection compares definitions of what activity is considered normal against observed events to identify significant deviations. This method uses profiles that are developed by monitoring the characteristics of typical activity over a period of time. The IDS/IPS then compares the characteristics of current activity to thresholds related to the profile. Anomaly-based detection methods can be very effective at detecting previously unknown threats. Common problems with anomaly-based detection are inadvertently including malicious activity within a profile, establishing profiles that are not sufficiently complex to reflect real-world computing activity, and generating many false positives.

4.2.1.3 Stateful Protocol Analysis

Stateful protocol analysis compares predetermined profiles of generally accepted definitions of benign protocol activity for each protocol state against observed events to identify deviations. Unlike anomaly-based detection, which uses host or network-specific profiles, stateful protocol analysis relies on profiles that specify how particular protocols should and should not be used. Stateful protocol analysis monitors and tracks the state of protocols that have a notion of state, resulting in the detection of many attacks that other methods overlook. Problems with stateful protocol analysis include: it is often very difficult or impossible to develop completely accurate models of protocols, it is very resource-intensive, and it cannot detect attacks that do not violate the characteristics of generally acceptable protocol behavior.

4.2.2 IDS/IPS Technologies

The following are primary types of IDS/IPS technologies of interest in a voting system:

1. Network-based
2. Network Behavior Analysis (NBA)
3. Host-based

A combination of network-based and host-based IDS/IPS is needed for an effective IDS/IPS solution for voting systems. NBA technologies can also be deployed to counter DDoS attacks, worms, and other threats that NBAs are particularly good at detecting.

The following subsections describe each of these technologies.

4.2.2.1 Network-based

The network-based IPDS monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity.

Network-based IDS/IPSs cannot detect attacks within encrypted network traffic; therefore, either they should be deployed where they can monitor traffic before encryption or after decryption, or host-based IDS/IPSs should be used on endpoints to monitor unencrypted activity. Network-based IDS/IPSs are often unable to perform full analysis under high loads. Organizations with high-traffic loads should select sensors that can recognize high load conditions and either pass certain types of traffic without performing full analysis or drop low-priority traffic to reduce load, depending on the level of risk to the systems behind the firewall. Network-based IDS/IPSs are susceptible to various types of attacks, most involving large volumes of traffic. Organizations should select products that offer features designed to make them resistant to failure due to attack.

4.2.2.2 Network Behavior Analysis (NBA)

NBA IDS/IPS examines network traffic to identify threats that generate unusual traffic flows, such as DDoS attacks, scanning, and certain forms of malware.

NBA technologies are delayed in detecting attacks because of their data sources, especially when they rely on flow data from routers and other network devices. This data is often transferred to the NBA in batches from every minute to a few times an hour. Attacks that occur quickly may not be detected until they have already disrupted or damaged systems. This delay can be avoided by using sensors that do their own packet captures and analysis; however, this is much more resource-intensive than analyzing flow data. Also, a single NBA aggregator can analyze flow data from many networks, while a single sensor can generally directly monitor only a few networks at once. Therefore organizations that opt to avoid this delay by performing analysis on the sensors rather than on an aggregator might have to purchase more powerful sensors and/or more sensors.

4.2.2.3 Host-based IDS/IPS

A host-based IDS/IPS monitors the characteristics of a single host and the events occurring within that host for suspicious activity.

In host-based IDS/IPS, some detection techniques are performed only periodically, such as hourly or a few times a day, to identify events that have already happened, causing significant delay in identifying certain events. Also, many host-based IDS/IPSs forward their alert data to management servers in batches a few times an hour, which can cause delays in initiating response actions. Because host-based IDS/IPSs run agents on the hosts being monitored, they can impact host performance because of the resources the agents consume. Installing an agent can also cause conflicts with existing host security

controls, such as personal firewalls and VPN clients. Agent upgrades and some configuration changes can also necessitate rebooting the monitored hosts.

4.2.3 Components of IDS/IPS

The following are components of an IDS/IPS solution:

1. Sensors (also known as agents): Sensors monitor and analyze activity; sensors are used to monitor networks and hosts.
2. Management Servers: Management servers receive information from sensors and manage the sensors and the information received from the sensors.
3. Database Servers: Database servers are repositories for event information recorded by the sensors or agents and management servers
4. Consoles: Consoles are programs that provide interfaces for IDS/IPS users and administrators

These components can be connected to each other through an organization's standard networks or through a separate network strictly designed for security software management known as a management network. A management network helps to protect the IDS/IPS from attack and to ensure it has adequate bandwidth under adverse conditions. A virtual management network can be created using a virtual local area network (VLAN); this provides protection for IDS/IPS communications, but not as much protection as a physically separate management network could provide since the network infrastructure would be shared.

4.2.4 IDS/IPS Functions

Most IDS/IPSs can provide a wide variety of security capabilities. Some products offer information gathering capabilities, such as collecting information on hosts or networks from observed activity. IDS/IPSs also typically perform extensive logging of data related to detected events. This data can be used to confirm the validity of alerts, investigate incidents, and correlate events between the IDS/IPS and other logging sources. Generally, logs should be stored both locally and centrally to support the integrity and availability of the data.

IDS/IPSs typically offer extensive, broad detection capabilities. The types of events detected and the typical accuracy of detection vary greatly depending on the type of IDS/IPS technology. Most IDS/IPSs require at least some tuning and customization to improve their detection accuracy, usability, and effectiveness. Typically, the more powerful a product's tuning and customization capabilities are, the more its detection accuracy can be improved from the default configuration. Administrators should review tuning and customizations periodically to ensure that they are still accurate. Administrators should also ensure that any products collecting baselines for anomaly-based detection have those baselines rebuilt periodically as needed to support accurate detection.

Most IDS/IPSs offer multiple prevention capabilities; the specific capabilities vary by IDS/IPS technology type. IDS/IPSs usually allow administrators to specify the prevention capability configuration for each type of alert. This includes enabling or disabling prevention, as well as specifying which type of prevention capability should be used.

4.2.5 Securing IDS/IPS

In addition to hardening software-based IDS/IPS components and ensuring that all IDS/IPS components are fully up-to-date, administrators should perform additional actions to ensure that the IDS/IPS components themselves are secured appropriately. Examples include creating separate accounts for each IDS/IPS user and administrator, restricting network access to IDS/IPS components, and ensuring that IDS/IPS management communications are protected appropriately. All encryption used for protection should be performed using FIPS-approved encryption algorithms.

Administrators should maintain IDS/IPSs on an ongoing basis. This should include monitoring the IDS/IPS components for operational and security issues, performing regular vulnerability assessments, responding appropriately to vulnerabilities in the IDS/IPS components, and testing and deploying IDS/IPS software and signature updates. Administrators should verify the integrity of updates before applying them, because updates could have been inadvertently or intentionally altered or replaced. Administrators should test software and signature updates before applying them, except for emergency situations. Administrators should also back up configuration settings periodically and before applying software or signature updates to ensure that existing settings are not inadvertently lost.

4.2.6 Best Practices for IDS/IPS for Voting Systems

One or more dedicated platforms (also called appliances) should be used for intrusion detection and prevention. The IDS/IPS should have the following security controls:

1. The IDS/IPS should operate on a hardware appliance dedicated to performing IDS/IPS and associated logging functions only.
2. The IDS/IPS should operate in a protected execution environment to protect itself from interference and tampering by other applications.
3. The platform should not permit any network based user login.
4. The platform should contain the minimum number of administrative accounts necessary for the IDS/IPS administration. This can, and should, include separate administrative accounts for each individual administering the IDS/IPS. The platform should not contain any other user accounts.
5. The platform should have only the IDS/IPS and associated logging applications installed.
6. The platform should only have the network services installed and active required for operation of IDS/IPS. All other network services should be either not installed or disabled.
7. The IDS/IPS log should be protected from unauthorized examination and modification. The IDS/IPS log should be treated like the operating system log discussed in Section 5.4 to ensure that the IDS/IPS log cannot be used to compromise voter PII.

At a minimum network-based IDS/IPS should be used with the following capabilities:

1. Information gathering
2. Detection
3. Blacklisting
4. Passive prevention

Network architecture for IDS and IPS could be any one of the following, however, the IDS/IPS data must be managed so as to not reveal voter choices. Further discussion of these architectural choices is provided in NIST SP800-94.

1. Inline
2. Passive
3. Tap
4. Load Balance

Network IDS and IPS should be able to at a minimum terminate an offending TCP session. Other actions maybe also be used: firewalling (i.e., drop or reject suspicious network activity); throttling bandwidth usage; and sanitizing packets to remove malicious content.

Network IDS and IPS may optionally perform NBA, which examines network traffic to identify threats that generate unusual traffic flows, such as DDoS attacks, certain forms of malware (e.g., worms, backdoors), and policy violations (e.g., a client system providing network services to other systems).

4.3 Virtual Private Network (VPN)

A VPN encrypts traffic and provides user authentication and integrity checking and thus providing secure network links across untrusted networks. VPN technology is widely used to extend the protected network of a multi-site organization across the Internet. VPN technology is also used to provide secure remote user access to internal organizational networks via the Internet.

The following circumstances are examples of when VPN technology is used:

1. The organization wishes to secure communication between two sites without going through the cost and inconvenience of providing cryptographic capability for each user and/or machine.
2. The organization wishes to technically enforce the security policy to protect information between two or more sites
3. The organization is concerned that users may accidentally or intentionally not encrypt data sent between two or more sites.
4. Remote users and offices are connected with the location where IT systems and applications reside.

VPNs allow the firewall administrator to decide which users have access to which network resources. This access control is normally on a per-user basis; that is, the VPN policy outlines which users are authorized to access which resources. VPNs

generally rely on authentication protocols such as Remote Authentication Dial In User Service (RADIUS) [RFC2865]. RADIUS uses several different types of authentication credentials, with the most common examples being username and password, digital signatures, and hardware tokens.

Two common choices for secure VPNs are:

1. IPsec based VPN
2. TLS based tunnel VPN. TLS based tunnel VPNs can be invoked using one of the three methods
    a) Preinstalled client: This approach is most secure and recommended
    b) Downloadable client from the VPN Server: While the downloaded code is digitally signed and can be verified, the number of trust anchors in a typical workstation environment and effort required to determine true identity of signer and validity of signature make this option less attractive. Additionally, the user must have sufficient privileges to install the downloaded client.
    c) Java applet download: While the downloaded code is digitally signed and can be verified, the number of trust anchors in a typical workstation environment and effort required to determine true identity of signer and validity of signature make this option less attractive.

The three most common VPN architectures are:

1. Gateway-to-Gateway
2. Host-to-Gateway
3. Host-to-Host

The following subsections describe each of the architectures:

4.3.1 Gateway-to-Gateway

A gateway-to-gateway VPN connects multiple fixed sites over unsecured network (e.g., the Internet) through the use of a VPN gateway. This architecture is used to connect geographically dispersed offices of an organization. A VPN gateway is usually part of another network device such as a firewall or router. When a VPN connection is established between the two gateways, users at the two locations are unaware of the connection and do not require any special settings on their computers.

The advantage of this approach is that it is cost-effective and enforces the security policy for protection of data in transit. However, this approach does not protect users within the protected Enterprise network from each other or protect sensitive hosts and servers from internal users.

4.3.2 Host-to-Gateway

A host-to-gateway VPN provides a secure connection to the network for individual remote users, who are located outside the physical network. In this situation, a client on the user machine negotiates the secure connection with the VPN gateway. The gateway side of the Host-to-gateway VPN is part of the firewall.

The advantage of this approach is that is very useful for telecommuters and travelers to electronically connect to the office and access all the resources. The disadvantage of this approach is that is requires each remote user to install the VPN client. The host to gateway VPN client can also provide an attack path if the remote machine is connected on an unsecured network. An attacker can compromise the machine over the unsecured network and then use the machine to attack the organization's network.

4.3.3 Host-to-Host VPN

Host-to-host VPN is rarely used. This setup typically enables remote administration of a single server.

The advantage of this approach is that two highly sensitive hosts located in different locations can securely communicate with each other.

4.4 Log Management Infrastructure

Because of the sensitivity of the information likely to be contained within UOCAVA system and network logs, UOCAVA systems should not share an organization-wide centralized log management infrastructure. Because the log entries themselves are potentially sensitive, any centralized log repository receiving data from the system should be protected using the same controls as the information on the most sensitive hosts from which it receives log data.

In determining whether or not a centralized log management infrastructure is required for a UOCAVA system, the size and purpose of the deployment should be taken into consideration. Copying logs to a centralized, distinct system provides valuable assurance that the logs constitute an accurate record of system activities. It also streamlines log review during operation. Because of the verbosity of the log entries on systems configured according to the guidelines in this document, when centralized log management is implemented, a dedicated logging network may be required in order to prevent the increase in network traffic from interfering with the operation of the system.

The size and scope of many installations will be sufficiently limited that the processes and policies outlined elsewhere for log management and processing can be followed for each host and component of the system without imposing prohibitive personnel overhead, and other controls may provide assurance that logs are accurate.

If the scale or function of a particular deployment requires centralized log management, designers and administrators should consult NIST SP 800-92, *Guide to Computer Security Log Management* [SP800-92], for detailed guidance on the design and deployment of a secure, dedicated log management infrastructure specific to the voting system. The controls in this publication should be applied to any such system.

4.5 Best Practices for Voting System: Network Architecture

The figure below depicts a network architecture which follows the best practices described in this document for an IT system used to support UOCAVA voting.

The architecture has the following salient features:

1. The voting system and administrative consoles are within a physically secure environment.
2. The administrative consoles are directly connected to the voting systems.
3. The voting system is under two person physical control.
4. The voting system is protected by a stateful inspection firewall.
5. The DMZ is protected by filtering router.
6. The DMZ contains outbound proxy for SMTP and HTTP.
7. Network-based and host-based IDS/IPS are installed on the voting system network and servers respectively.
8. The election official workstations are within a physically secure environment.
9. The election official workstations are connected by Host-Gateway VPN to the voting system.
10. The election official workstations are protected by Enterprise firewall.
11. Host-based IDS/IPSs are installed on the election official workstations.

In the diagram, it is assumed that system administration is conducted within the application hosting facility, while election officials configure the application to support a particular election from a separate location, designated an “Election Management Facility” on the diagram.

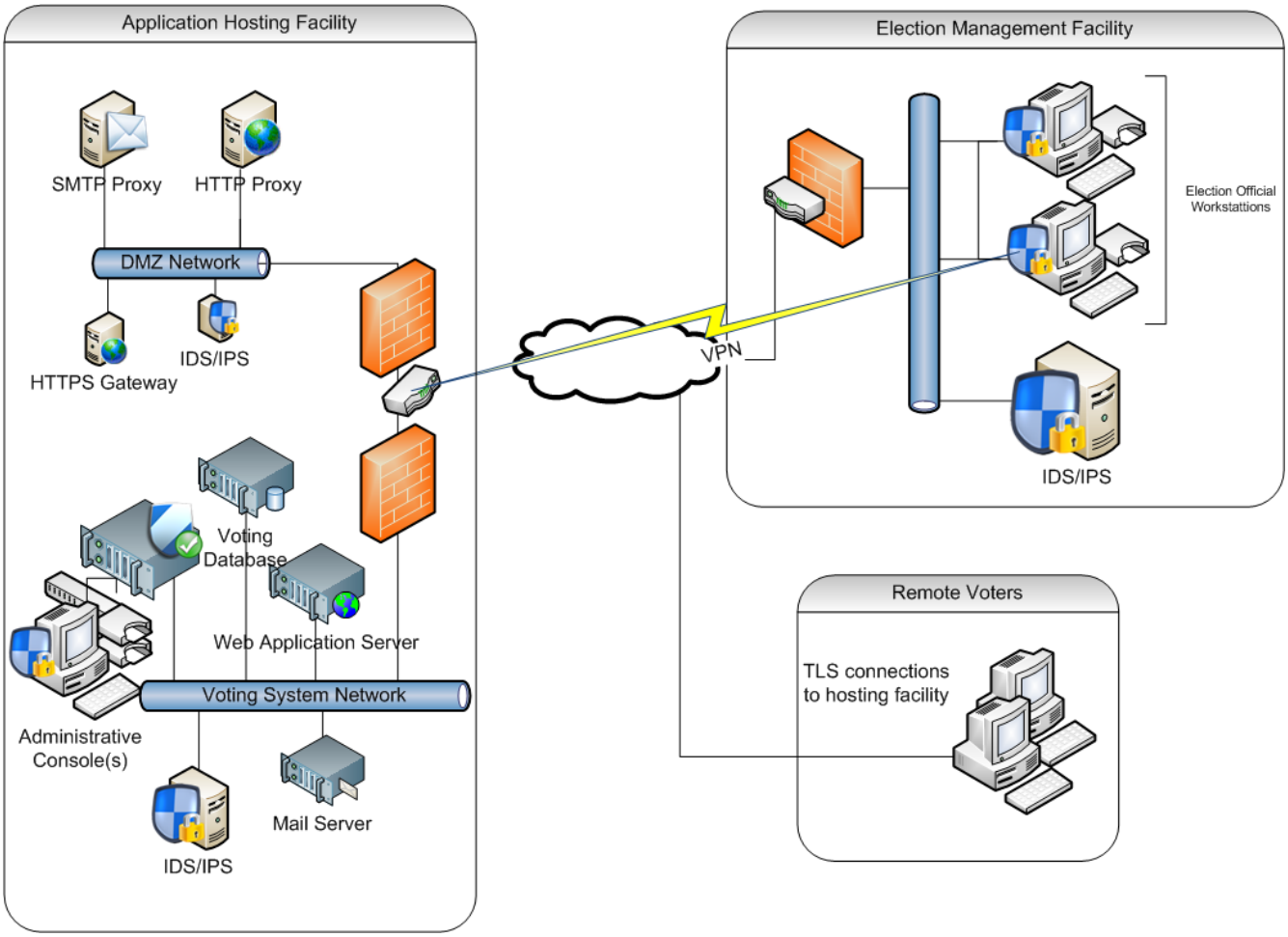

Figure 4. Voting System Network Architecture

5 Host Protection

All servers and workstations that provide IT services in support of an overseas voting system should be protected using appropriate system and application security controls. The specific mechanisms and settings involved will differ according to the purpose of the server or workstation in question. This section outlines controls and practices that should be configured on every system. For detailed configuration steps and additional, application-specific considerations, consult the National Checklist Program (http://checklists.nist.gov) as well as application-specific NIST guidelines published on http://csrc.nist.gov/.

Although much of the information in this section will be useful to system designers, the guidelines here are primarily intended to be used by voting system administrators to deploy overseas voting systems securely.

For each server and workstation, the protections specified in the following subsections should be used to establish a secure baseline configuration. The function of each system should be clearly defined, and the most restrictive protections which will permit fulfillment of that function should be selected in each area.

Once the systems are configured and brought into operation, the configuration management guidance outlined in Section 6.6 should be followed to ensure their continued secure operation

This baseline should be documented prior to the deployment of each server or workstation and kept up-to-date as changes are made. The controls described in section 6.6 should be used to ensure that the secure baseline configuration is continuously updated.

5.1 Operating System Identification & Authentication (I&A)

The operating system I&A is used to authenticate individuals who are required to use the operating system. There is no need for accounts other than administrative users for the operating system accounts. Election officials and voters obtain services via voting application and thus do not require operating system accounts.

Identification and authentication of administrative personnel to the operating system should be user ID and password or certificate based as discussed in Section 3.1.6.

5.2 Operating System Discretionary Access Control

The Operating System DAC should be used to protect all the system and voting application files. Only users and processes that require access to system and voting application files should be granted access to those files. Additionally, only the required level of access permissions (e.g., read, write, execute) should be granted. All other users and processes should not have access to those files.

5.3 Account Management

Server and workstation operating systems should be configured such that only the authorized administrators can create accounts on the system. On servers, aside from the authorized administrators, voting applications may need accounts in order to execute with application account privileges as opposed to administrative privileges. This approach helps enforce the principle of least privilege.

Servers should only contain the accounts that are required for the operation of the system. Furthermore, the accounts that do not require operating system logon should be configured to prohibit logon.

On workstations, aside from the system administrator, the workstation should only contain the accounts for the voting system administrators or the election officials who use the workstation to access the voting system.

See http://checklists.nist.gov/ for detailed guidance on configuration of specific systems.

5.4 Event Log

The operating system event logs should be protected from unauthorized examination and modification using operating system DAC as described earlier.

System clocks should be synchronized with an authoritative time source using NTP. System clock synchronization against a time source is required to ensure that the analysis of event ordering and timing is accurate.

The following list of events should be logged by the operating system:

1. System startup
2. System shutdown
3. Login and logout
4. Execution of applications and services
5. Administrative actions
6. Changes to system configuration
7. Change in authentication values (e.g., password, certificate)
8. Event log
    a) Change to list of events to be logged
    b) Event log deletion
    c) Overwrite of event log
    d) Backup of event log
    e) Change in event log space allocation (e.g. log roll threshold, maximum log size)
    f) Change to system clock
9. Modification to system files
10. Addition and deletion of files
11. Backup
12. Restore
13. Unsuccessful attempts to access any file
14. Any attempt to access system files
15. Account Management
    a) Creation
    b) Deletion
    c) Modification
    d) Changes to privileges
16. Malware protection software events
    a) Software update
    b) Signature update
    c) Execution
17. Cryptographic key generation and destruction: This event may be generated manually, by the operating system or cryptographic module.

5.5 Host-Based Firewall

Voting system servers and workstations should be configured with a host-based firewall.

The firewall should be configured to allow only the minimum set of inbound and outbound connections required for the operation of the voting application. These connections should be limited to protocols and IP addresses designated as narrowly as possible.

Consult http://checklists.nist.gov/ for specific guidance on host-based firewall configuration.

5.6 Minimize Services

Servers and workstations should be configured such that the network services and other computing services software that are not required for the operation of the voting application are removed from the system altogether. If they cannot be

removed, execution of these services should be disabled. Auto-run and auto-play upon introduction of media and files in the system should be disabled.

Note that a locked down and secured voting system will have many otherwise routine network and computing services removed and disabled. Unless necessary for a system to perform its duties, the following services should be disabled or removed on both servers and workstations:

1. File and printer sharing services (e.g., Windows Network Basic Input/Output System (NetBIOS) file and printer sharing, Network File System (NFS)], File Transfer Protocol (FTP))
2. Wireless networking services
3. Remote control and remote access programs
4. Directory services (e.g., Lightweight Directory Access Protocol [LDAP], Network Information System [NIS])
5. Email services (e.g., Simple Mail Transfer Protocol (SMTP[8]))
6. Language compilers and libraries
7. System development tools
8. System and network management tools and utilities, including Simple Network Management Protocol (SNMP).

5.7 Host Based Intrusion Detection and Prevention

Each server and workstation should contain and operate a host based intrusion detection and prevention system. At a minimum, the tool should detect and prevent modification of all executable files and addition of any executable files. It should detect and prevent attempts to modify system files. The tool should monitor for and alert administrators to modification of access rights to key system files.

Consult http://checklists.nist.gov for detailed guidance on configuration of host-based intrusion prevention systems for specific operating systems.

5.8 Malware Protection

Each server and workstation should be configured with malware protection that can detect viruses, Trojans, worms, spyware and rootkits. The malware protection software should be configured for the following:

1. Regular periodic scan
2. Scan removable media
3. Real-time on-access file scanning

On hosts where real-time on-access file scanning interferes with the voting application, real-time scans of newly created files may be configured instead of full on-access scanning.

5.9 Backup and Restore

The system should provide backup and restore capabilities for all servers. If the backup and restore functions provide cryptographic checksum (e.g., digital signature, MAC, HMAC, or hash) protection, the protection should be enabled and configured. The checksum should be stored separately from the backup media. Prior to restoring the server from the backup media, operators should confirm the integrity of the backups using the checksum.

Workstations may not require backup and restore capabilities if they do not store critical data for the voting system.

5.10 Voting System Application Security

In order to support the functions performed by the election officials and voters, voting systems will require applications. Examples of the applications include: Web Server, Web Server Application, DBMS, and DBMS applications.

The following subsections describe application level security controls.

[8] This capability may be required if the application uses automated e-mail as the mechanism to provide ballots.

5.10.1 Application Level Identification & Authentication

The application level I&A is used to authenticate individuals who require use of the application. Examples include voters and election officials. They are likely to be authenticated to the Web based application using the means described in Section 3.1.6

5.10.2 Application Discretionary Access Control

The application DAC should be used to protect all voting application data. The application DAC should permit the election officials and voters to perform their functions under the control of the application only. Role based access control is well-suited for application level DAC for voting system applications. In most situations, the voting system designer should be able to implement role based access control using group or role mechanism provided by the application or the underlying operating system.

5.10.3 Application Account Management

The application should be configured such that only the authorized administrators can create application accounts. The accounts may be required for the election officials. See http://checklists.nist.gov/ for detailed guidance on configuration of specific operating systems.

5.10.4 Application Event Log

The application event logs should be protected from unauthorized examination and modification using operating system DAC.

Where feasible, the application event logs should also be protected from unauthorized examination and modification using application-enforced DAC.

See section 3 for a discussion of the use of DAC to counter threats confidentiality and integrity.

The application event logs should use the operating system clock for time stamping the events.

The following list of events should be logged by the application:

1. Application startup
2. Application shutdown
3. Login and logout
4. Administrative actions
5. Changes to application configuration
6. Changes to ballot configuration
7. Change in authentication values (e.g., password, certificate)
8. Event log
    a) Change to list of events to be logged
    b) Event log deletion
    c) Overwrite of event log
    d) Backup of event log
    e) Change in event log space allocation (e.g. log roll threshold, maximum log size)
9. Account Management
    a) Creation
    b) Deletion
    c) Modification
    d) Changes to privileges
10. All ballots generated, excluding ballot number

5.10.5 General Application Security Practices

Voting system applications should be designed and implemented using the following principles:

1. Applications should be developed using a well-understood coding convention.
2. No operating system "system files" should be accessed.
3. Applications should not execute with impersonation (e.g., impersonation in Windows and SUID in Unix).
4. Applications should not interact with other applications.
5. When accessing a system object, its full path name should be used. This protects against path variable related errors as well as malicious attempts to subvert the system.
6. Use of hard or symbolic links (e.g., shortcuts for Windows) should be disabled.
7. All executable files should be placed in a folder that does not have the modify permission for anyone.
8. All user input should be validated.
9. Protection against buffer overflows and memory leaks should be provided.
10. No services should be provided until the user is properly authenticated.
11. No third-party scripts or executable code should be used without verifying the source code.

Additionally, vulnerability analysis and remediation should be performed and documented as described in Section 7.2.

5.10.6 Web Application Security Practices

This section was developed using NIST SP 800-44 Version 2, Guidelines on Securing Public Web Servers [SP800-44], and NIST SP 800-123, Guide *to General Server Security* [SP800-123]. Readers may consult these documents for background and additional details.

Web-based voting applications should be designed and implemented using the following principles:

1. All the principles listed in Section 5.10.5.
2. The system should include protection mechanisms against web bots.
3. A single hard drive or logical partition should be dedicated for Web content. Furthermore,
    a) This drive/partition should not contain any other information.
    b) All directories and subdirectories in this drive/partition should be exclusively for Web server content files, including graphics but excluding scripts and other programs
4. A single directory should be used exclusively for all external scripts or programs executed as part of Web server content (e.g., Common Gateway Interface (CGI), Active Server Pages (ASP)). This directory should not contain anything except external scripts or programs, and the web server should not be configured to execute scripts or programs located elsewhere.
5. A complete Web content access matrix should be developed that identifies which directories and files within the Web server document directory are restricted and which are accessible (and by whom).
6. Directory listings by the web users should be disabled.
7. Execution of scripts that are not exclusively under the control of administrative accounts should be disabled. This action is accomplished by creating and controlling access to the separate directory intended to contain authorized scripts.
8. Server Side Includes (SSI), or their execution, should be disabled.
9. Web content generation code should be scanned or audited.
10. No process except web server administration processes should be able to write to web content files. This can be accomplished by using the operating system discretionary access controls on the web content files and directories.
11. Dynamically generated pages should not contain dangerous metacharacters (e.g., & ; ` ' \ " | * ? ~ < > ^ ( ) [ ] { } $ \n \r\0)
12. Character set encoding should be explicitly set in each page.

13. Special characters or HTML tags should be processed so that they cannot be used for exploitation.
14. Cookies should be examined to ensure they do not contain any unexpected data.
15. Input validation should be performed by the web application so that the web application's security mechanisms cannot be bypassed when a malicious user tampers with data he or she sends to the application, including Hypertext Transfer Protocol (HTTP) requests, headers, query strings, cookies, form fields, and hidden fields. This mechanism also protects against Cross-Site Scripting (XSS) and Structured Query Language (SQL) injection attacks.
16. In many cases, there should not be a need to permit the users to upload files to the Web Server. If such a need were determined,
    a) Uploads should not be readable by the Web server. This can be accomplished by using the operating system discretionary access controls on upload files and directories.
    b) Uploads should be limited to a defined directory. The directory and its subfolders should not be readable by the Web server.
17. All sample scripts should be removed from the operational system.
18. Cross Site Request Forgery (CSRF) attacks should be prevented by making sure that neither an attacker nor a script running on the attacker's website has sufficient information to construct a valid request authorizing an action (with significant consequences). This can be done by inserting unpredictable challenge tokens associated with the user session into each request into URLs or forms that cause actions to be performed on behalf of the user.
19. The web application should be protected against TLS renegotiation attacks. The TLS renegotiation extension protects against these attacks. In lieu of, or in addition to, the use of TLS renegotiation extension, web pages and applications should be designed so that when a step up authentication occurs, inputs provided by the client that resulted in the need to negotiate higher authentication level are ignored and the client is required to resubmit the request after the requisite authentication is complete.
20. Follow community recommended best practices for web application development for specific languages or frameworks, e.g., .NET, PHP, Java, Ajax, etc.

Systems administrators may consult NIST SP800-44 *Guidelines on Securing Public Web Servers* for a list of tools for vulnerability scanning and log analysis.

Additionally, vulnerability analysis and remediation should be performed and documented as described in Section 7.2.

5.11 Workstation Network Protections

When workstations are on a separate network from the servers in an overseas voting system, the workstation network should be protected using similar mechanisms to the voting system network. This network protection should use a multi-layered approach by using a firewall to block remote network-based attacks as well as IPS/IDS in case some attack attempts are not stopped at the firewall.

5.11.1 Firewall

In addition to the host-based firewalls installed on each workstation, the workstation network should be protected by one or more dedicated firewalls. The requirements for the workstation network firewall are the same as those for the host network firewall, detailed in section 4.1.

5.11.2 Intrusion Detection System

In addition to the host-based IDS installed on each workstation, network-based IDS should be employed on the workstation network. The guidelines for IDS configuration detailed in section 4.2.6 also apply to the workstation network.

5.11.3 Virtual Private Network

The workstation should be connected to the voting system using the VPN detailed in Section 4.3.

6 Operational Controls

The controls described in this section apply to the voting system, firewall, IDS/IPS protecting the voting system, and mail server used for fulfilling voting system functions. As applicable, the requirements apply to the hardware, operating system software, application software, and cryptographic equipment. The guidelines in this section will be most beneficial to system administrators and technical staff charged with routine operation of UOCAVA systems.

6.1 Facility Controls

The site and room for the voting system should have the following controls:

1. The voting system site and room should have physical security controls to protect highly sensitive systems.
2. The voting system should have a reliable power source to ensure system availability commensurate with commercial systems.
3. The voting system should have reliable air conditioning to ensure system availability commensurate with commercial systems.
4. The voting system should have protections against water and fire hazards commensurate with commercial systems.

6.2 Media Storage and Off-site Backup

Media and backups should be stored in a location with controls commensurate with those specified in Section 6.1.

Media and backups should be under the same multi-person control as the live system. This may be achieved using a combination of logical and physical controls, e.g. by encrypting the backup data and storing the keys separately from the activation data needed to access them.

The system administrator should use manual or automated means to keep records of all media which are loaded with data from the voting system.

These records should be sufficiently detailed to positively identify the media.

The storage location of all media containing voting system data should be recorded.

The system administrator should use manual or automated means to record all access to the backup or archival media.

Access to media and backups should be audited using the same process and frequency as access to the live system.

When media will no longer be used to store voting system data, the media should be destroyed or sanitized in accordance with the practices defined for removal of the system from service. See Section **Error! Reference source not found.** for additional details.

6.3 Personnel Security Controls

6.3.1 Position Categorization

For the system administrator and election official positions:

1. Risk designations should be developed;
2. Screening criteria for individuals filling these positions should be developed; and
3. Individuals nominated for these positions should undergo a screening process.

6.3.2 Separation of Duties

A system administrator should not be assigned an election official role and vice versa.

Physical, technical, procedural controls should be employed such that physical and logical access to the voting system, and performance of administrative tasks requires two system administrators. Note that this may require that the administrator use local consoles only to login and perform their tasks.

6.3.3 Qualifications, Experience, and Training

The system administrators and election officials should meet the following requirements related to performance of their duties:

1. They should successfully complete an appropriate training program commensurate with their role.
2. They should have demonstrated the ability to perform their duties.
3. They should not be assigned other responsibilities that would interfere or conflict with their ability to perform their duties.

The system administrators and election officials should be provided system manuals, user manuals, and procedures required to perform their duties.

6.4 Event Log Processing

On a UOCAVA system where components have been configured in conformance with the practices described in this document, the event logs will constitute a record of all significant activity. Appropriate management and processing of these logs is important to ensure the integrity of every system function. For detailed guidance on log management, consult NIST SP 800-92, *Guide to Computer Security Log Management* [SP800-92].

6.4.1 Frequency of Event Log Processing

Event logs should be processed frequently enough that no data is lost or overwritten. During election, this may mean daily processing or more frequently; testing should be performed to establish a safe frequency.

Where possible, an automated alert should be triggered well before the event log storage becomes full so that the system administrators can back up the event log.

6.4.2 Frequency of Event Log Review

During an election, the event logs should be reviewed daily. The objective of the review should be to determine if suspicious activities are taking place and if the event log processing schedule is appropriate. Section 6.4.3 contains additional details on events to examine.

6.4.3 Vulnerability Assessments

The developer of the UOCAVA system should supply a vulnerability assessment with the system documentation. This documentation includes potential approaches that an adversary could take in an attempt to subvert or disrupt the operation of the voting system. It also includes guidance advising system operators and administrators as to how such attempts might be detected and prevented. A critical element of this is monitoring the system's event logs.

The following are typical examples of events which could indicate attempts to subvert the system:

- Excessive number of events
- Failed login attempts
- Excessive password changes to the same account in a short period of time
- Creation of accounts
- Changes to account profiles
- Account lock out events
- Gaps in event logs
- Modification of critical system files
- Read access to sensitive files
- Installation of programs
- File access failures
- Changes to audit profile
- Changes to authentication policy
- Changes to file metadata (e.g., ownership, access control list, etc.)
- All accesses to databases and files containing PII

These and similar events described in the documentation delivered with the voting system should be monitored. Consult the system documentation for additional events that are indicative of attempts to violate the voting system security.

6.5 Backup and Archive

System backups sufficient to recover from failures should be made on at least a daily basis during the election. Recovery from these backups should be tested as part of system deployment prior to the election.

Event logs should be archived for retention based on the election records retention requirements.

System backups and event logs should meet the requirements for facility security specified in Section 6.1.

Access to system backups and event logs should be under multi-person system administrator control as specified in Section 6.3.2.

6.6 Configuration Management

The configuration of the components comprising a UOCAVA system should be managed according to a formal, documented policy and procedures. The policy and procedures should be periodically reviewed to ensure that the controls established for the various components of the system are maintained when the policy is adhered to and the procedures are followed.

6.6.1 Baseline Configuration

When the components of the system are deployed, the baseline configuration should be documented. This should include all details necessary to deploy the component into the production system. When changes are made, the documentation of the baseline should be updated as part of the change management process.

Where possible, automated mechanisms (e.g., SCAP-validated scanning tools) should be used to monitor and report the configuration of each component. Any deviation from the documented secure baseline should be flagged for review, and should trigger either a change to the component's configuration or an update to that documentation.

6.6.2 Configuration Change Control

All proposed changes to the system should first be formally proposed, reviewed and approved using a change control process that meets the requirements defined in the configuration policy. This process should record the rationale for and approval of each change to the system's configuration.

Where feasible, configuration changes should be deployed using automated tools that can ensure that the changes being deployed are the same as those that have been approved.

The analysis of each proposed change should focus on ensuring that all required security controls are maintained.

After a change has been deployed, the change control process should ensure that the deployed configuration change matches the documented configuration change and that the baseline configuration is updated.

6.6.3 System Hardware and Software Inventory

The system administrator should use manual or automated means to keep records of hardware and software installed on the voting system.

The system administrator should use manual or automated means to record all events related to updates to, and the disposition of, the hardware and software.

All hardware and software should contain sufficient information for precise identification of a configuration. This may include manufacturer, make and model, version number, and revision number. Where feasible, this information should be collected, documented and monitored for changes using automated mechanisms. For an expanded list of configuration items that may apply, see NIST SP 800-40 Version 2, *Creating a Patch and Vulnerability Management Program* [SP800-40].

6.6.4 Cryptographic Material inventory

The system administrator should use manual or automated means to keep records of hardware and firmware used in cryptographic modules.

The system administrator should use manual or automated means to record all events related to updates to, or the disposition of, the cryptographic hardware and firmware.

All cryptographic hardware and firmware should contain sufficient information to identify precisely which hardware is in use at a given time. This may include manufacturer, make and model, version number, and revision number; consult the documentation supplied with the system for details.

6.7 Disaster Recovery

The voting system should contain a disaster recovery plan from various failures such as:

1. Facility unavailability
2. Cryptographic module failure
3. Hardware failure
4. Software failure

The disaster recovery plan should undergo a successful test one week prior to start of election.

6.8 Ongoing Testing

6.8.1 Penetration Testing

The voting system should undergo penetration testing after it is fully deployed to ensure that the vulnerability assessment is conducted against the exact configuration that will be used to conduct the election. This testing should take place as near to the start date of the election as is feasible, to enable the penetration testers to take advantage of the most recent known vulnerabilities, while at the same time providing system owners, administrators and vendors an opportunity to mitigate any discovered vulnerabilities. The testing should be conducted by experienced experts in penetration testing. The testers should be provided with all the system design documentation available to the voting system developer and should use information from this documentation to retrieve information on potential vulnerabilities from the National Vulnerability Database (NVD). Any vulnerability identified by the penetration testing should be resolved before the system is deemed fit for conducting the election.

See NIST SP 800-115, *Technical Guide to Information Security Testing and Assessment* [SP800-115], for additional guidelines.

6.8.2 Network Configuration Monitoring

The voting system network configuration should be verified using the penetration testing, network mapping, and IDS/IPS tools as close to the start of the election as is feasible, allowing time for system administrators to resolve any problems that are discovered.

The voting system network configuration should be monitored on an ongoing basis by the IDS/IPS tools.

The voting system firewall rules should be examined and verified to be accurate and enforced by using the penetration testing, network mapping, and IDS/IPS tools.

6.8.3 Availability Monitoring and Load Testing

Prior to the start of the election, the voting system should be tested under anticipated peak load conditions and the response time should be verified to be within target goal. The load should be created for each class of user functions the voting system supports, i.e.

1. Registration Database Update
2. Obtain a Ballot
3. Cast a Ballot

This testing should be conducted once the projected load is known and with sufficient lead time to address concerns raised by load testing.

During the election, the voting system should be monitored using automated or manual means to ensure that all the user functions listed above are available.

6.8.4 Compliance Audit

Prior to the start of the election and in conjunction with penetration testing, the voting system should undergo a compliance audit to ensure that the voting system has controls in place to meet the requirements specified in this document. Any

deficiencies identified by the compliance audit should be corrected and an incremental audit should be conducted to ensure that all the deficiencies are closed prior to the start of election.

The audit should take place only after the final configuration has been put into place for the election. Scheduling considerations for the audit should balance the need to audit the configuration that's actually used during the election period with the need to allow enough time to address any concerns raised by the audit.

6.9 Incident Handling

The voting system operator should have incident reporting and handling systems and processes in place. These should provide the following functions:

1. There should be a mechanism for voters, election officials, and system administrators to report security incidents.
2. Reported security incidents should be kept in a secure manual or automated database.
3. Only authorized development and system administration personnel should have the ability to access the database for both review and updates
4. Each open security incident should be assigned to an individual as recorded in the database.
5. The database should maintain the status of the incident in terms of whether it is being investigated, has been confirmed, being fixed, or has been fixed.

Any problems with commercial products used in the voting system should be resolved in conjunction with the commercial product vendor in order to fix the vulnerability.

6.10 Removal from Service

Prior to removal from service or disposal of equipment, the following activities should be undertaken:

1. All cryptographic equipment should be zeroed out.
2. All event logs on the computer systems should be archived.
3. All files on the computer systems should be deleted.
4. Hard drives and other storage media used by system equipment should be sanitized before those components are disposed of or repurposed. Section 5 of NIST SP 800-88, *Guidelines for Media Sanitization* [SP800-88], contains descriptions of sanitization methods. Degaussing or destroying hard drives can provide high assurance that any sensitive data previously stored on the drive is not recoverable. If storage media will be repurposed, organizations may clear the drive by using a secure eraser tool to overwrite the hard drive with random data. For some ATA hard drives which support the Secure Erase command, a better option may be to use a tool to securely purge a drive using this special-purpose command in the ATA specification.
5. The computer system should be powered off for few minutes prior to release of the equipment.

7 Assurance Requirements

The controls described in this section apply to the servers and workstations that comprise the voting system, firewall, IDS/IPS protecting the voting system, and mail server used for fulfilling voting system functions. Where appropriate, these requirements apply to the hardware, operating system software, application software, and cryptographic equipment. The purpose of these assurance requirements is to establish confidence that the system as a whole has been both evaluated and determined to meet the security requirements of the application, and that the system is being operated in the same configuration that was evaluated.

This section should be used by system designers, both in the selection of components and as a checklist for documentation that should accompany the overall system. It should be used by personnel charged with administration and deployment of UOCAVA systems as a reference to documentation that will accompany the system. By following these guidelines, designers and implementers can ensure that the IT systems being deployed will enforce the controls discussed in previous sections.

7.1 Documentation Requirements

The documentation described in this section should be provided by the designer of each system or component and should be evaluated along with the system being deployed. Its purpose is to ensure that the system is deployed and maintained in a configuration with the same security controls as the system whose security was evaluated prior to selection.

7.1.1 Administration Guidance

The requirements specified in this section should also be applied during the selection or development of products that comprise a UOCAVA system. Each product used to support UOCAVA voting should be accompanied by detailed guidance documentation in the following areas:

1. Secure Delivery, Installation, and Start-up Guides
2. Administration Guide
3. Maintenance, Upgrade, and Flaw Remediation Procedures

This guidance should be evaluated along with the system to ensure that it is sufficient to bring the system into a secure operational state and that the configuration which results from following this guidance is identical to that which was evaluated and determined to meet the security requirements.

7.1.1.1 Secure Delivery, Installation, and Start-up Guides

All components supplied as part of a UOCAVA system should be accompanied by detailed documentation of the procedures necessary to deploy the components in the secure configuration that was used to certify their suitability for use in the voting system. These should include

- Guidance for validating the integrity of the hardware and software components that will be deployed as part of the UOCAVA system
- Documentation of the installation procedures necessary for a secure configuration
- Documentation of the procedures required to place the system in a secure operational state

The totality of this documentation should be sufficiently detailed that administrators can verify that the components being deployed are

- Complete, as selected by the system designer
- Do not differ from those that were evaluated and determined to provide the security features required by the UOCAVA system
- Configured identically to the components whose security was evaluated
- Are operating in a secure state once all components have been installed

Designers should consult [CEMv3.1] for further detail on evaluating whether component documentation is sufficient to achieve these goals.

The administrator should use the secure delivery guide to confirm the completeness and validity of all delivered system components.

The administrator should use the same secure installation and start-up procedures that were used to evaluate security as part of the system design to install the system and bring it into an operational state.

7.1.1.2 Administration Guide

Components deployed as part of a UOCAVA system should be accompanied by guidance documentation for system administrators. This documentation should describe each user role necessary for the operation of the system. The description of these roles should include the functions and privileges accessible to and required for each role and detail mechanisms for restricting them. This documentation should also explain those restrictions necessary in order to operate the UOCAVA system in a secure manner.

System designers should ensure that this guidance is clear, comprehensive and compatible with operation of all components in the context of an election. Designers should consult [CEMv3.1] for guidance on evaluating the administrative documentation that accompanies system components.

The administrator should use the administrator guidance that has been evaluated in this context to manage the system.

7.1.1.3 Maintenance, Upgrade, and Flaw Remediation Procedures

Over the lifecycle of IT products, threats evolve and flaws are discovered. To ensure continued secure operation of a system, components need to be accompanied by procedures for maintaining system security, applying upgrades and addressing flaws. The documentation describing these procedures should be clear, detailed, and sufficient to ensure that each component is maintained in the secure state established in the installation, start-up and administration guides.

System designers should evaluate these procedures to ensure that they maintain the security of the system.

The administrator should use the system maintenance procedures to carry out preventive and corrective maintenance.

The administrator should use the upgrade procedures to regularly patch the system. The administrator should ensure that all systems and applications have the proper patches and security updates applied.

The administrator should use the system flaw remediation procedures to inform the system designer and the system vendor of applicable incidents as discussed in Section 6.9.

7.1.2 Design Documents

The requirements specified in this section should be applied during the selection of the products. These documents should describe a system that meets the security functional requirements of the application in question. The system should be evaluated against these documents prior to the deployment, to ensure that the product design is sound, the delivered system meets the design requirements, and that the design process included at least the following documents:

1. Functional Specification
2. Complete External Interfaces Specification consisting of the following for each interface:
    a) Inputs
    b) Processing (high level description)
    c) Outputs
    d) Errors
    e) Exceptions and Side Effects
3. System Architecture consisting of the following:
    a) Description of Major Functional Components
    b) External IT Entities
    c) System Interfaces
    d) Application Work-flow
4. High Level Design

7.2 Vulnerability Analysis

The documentation specified in this section should be analyzed during the selection of the products to ensure that comprehensive vulnerability testing was conducted by the vendor, and that the vulnerability testing included the following documents:

1. Vulnerability analysis methodology
2. Databases searched to conduct vulnerability analysis, including queries made to the NVD
3. Vulnerability analysis finding
4. Vulnerability confirmation or refutation (e.g., based on in-depth analysis or empirical penetration testing)
5. Actions taken to close any identified vulnerabilities
6. Residual vulnerabilities and proposed mitigations for these

System designers should review this documentation to ensure that IT components deployed will meet the security requirements of the UOCAVA system.

7.3 Testing Requirements

The requirements specified in this section should be applied during the selection of the products to ensure that comprehensive security testing was conducted, and that the security testing included the following documents:

1. Test plan and degree to which the external interface specification was tested. It is required that the external interface testing was comprehensive. Testing is considered comprehensive if every external interface is tested for nominal and boundary conditions and every error for each external interface is exercised.
2. Test cases and test procedures
3. Automated test scripts
4. Test results

System designers who are making use of COTS components should review the security testing documentation which accompanies these, ensure that IT components deployed will meet the security assurance requirements of the UOCAVA system, and reference the component testing documentation when documenting the entire system.

8 References

8.1 Documents and Papers

CEMv3.1	Common Methodology for Information Technology Security Evaluation – Evaluation methodology, July 2009, Version 3.1 Revision 3 Final http://www.niap-ccevs.org/cc_docs/CEMV3.1R3.pdf
GAO08536	GAO Report 08-536, Privacy: Alternatives Exist for Enhancing Protection of Personally Identifiable Information, May 2008, http://www.gao.gov/new.items/d08536.pdf
HAVA	Help America Vote Act of 2002 http://www.fec.gov/hava/law_ext.txt
NISTIR-7298	Glossary of Key Information Security Terms http://csrc.nist.gov/publications/nistir/NISTIR-7298_Glossary_Key_Infor_Security_Terms.pdf
NISTIR-7551	A Threat Analysis on UOCAVA Voting Systems, December 2008 http://www.nist.gov/itl/vote/upload/uocava-threatanalysis-final.pdf
OMB0404	E-Authentication Guidance for Federal agencies, OMB Memorandum M-04-04, December 16, 2003 http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf
PILOTREQ	UOCAVA Pilot Program Testing Requirements, March 24, 2010 http://www.eac.gov/program-areas/voting-systems/docs/requirements-03-24-10-uocava-pilot-program
RFC2865	Remote Authentication Dial In User Service (RADIUS), June 2000 http://www.ietf.org/rfc/rfc2865.txt
RFC5280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008 http://www.ietf.org/rfc/rfc5280.txt?number=5280
RUBIN	Security Considerations for Remote Electronic Voting over the Internet, Avi Rubin, AT&T Labs – Research
SERVE	A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE), Avi Rubin, et. al., 2004
SP800-122	Guide to Protecting the Confidentiality of Personally Identifiable Information (PII), April 2010. http://csrc.nist.gov/publications/drafts/800-122/Draft-SP800-122.pdf
SP800-123	Guide to General Server Security, July 2008 http://csrc.nist.gov/publications/nistpubs/800-123/SP800-123.pdf
SP800-191	Guideline for The Analysis Local Area Network Security, 9 November 1994 http://csrc.nist.gov/publications/fips/fips191/fips191.pdf
SP800-24	PBX Vulnerability Analysis http://csrc.nist.gov/publications/nistpubs/800-24/sp800-24pbx.pdf
SP800-41	Guidelines on Firewalls and Firewall Policy (Draft), July 2008 http://csrc.nist.gov/publications/drafts/800-41-Rev1/Draft-SP800-41rev1.pdf

SP800-44	Guidelines on Securing Public Web Servers, Version 2, September 2007 http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf
SP800-45	Guidelines on Electronic Mail Security, Version 2, February 2007 http://csrc.nist.gov/publications/nistpubs/800-45-version2/SP800-45v2.pdf
SP800-63	Electronic Authentication Guideline, Draft NIST Special Publication 800-63-1, December 8, 2008 http://csrc.nist.gov/publications/drafts/800-63-rev1/SP800-63-Rev1_Dec2008.pdf
SP800-88	Guidelines for Media Sanitization, September 2006 http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_rev1.pdf
SP800-92	Guide to Computer Security Log Management, September 2006 *http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf*
SP800-94	Guide to Intrusion Detection and Prevention Systems (IDPS), February 2007 http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf
TCSEC	Trusted Computer System Evaluation Criteria, 1985 http://csrc.nist.gov/publications/history/dod85.pdf
UOCAVA	The Uniformed and Overseas Citizens Absentee Voting Act http://www.usdoj.gov/crt/voting/misc/activ_uoc.php
UOCAVA-BP	Best Practices for Facilitating Voting by U.S. Citizens Covered by the Uniformed and Overseas Citizens Absentee Voting Act, September 2004 http://www.dos.state.pa.us/election_reform/lib/election_reform/Best_Practices_for_Facilitating_Voting_by_US_Citizens_Covered_by_the_UOCAVA_EAC.pdf

8.2 Useful Websites

Federal Voting Assistance Program	http://www.fvap.gov/
How E-voting Works, Kevin Bonsor and Jonathan Strickland:	http://people.howstuffworks.com/e-voting.htm
US Election Assistance Program, Resources for Overseas Citizens and Military Voters:	http://www.eac.gov/voter/overseas-citizens-and-military-voters
National Checklist Program (NCP)	http://checklists.nist.gov/
National Vulnerability Database (NVD)	http://nvd.nist.gov/
Security Content Automation Protocol (SCAP) specifications	http://scap.nist.gov/

9 List of Acronyms

ABAC	Attribute Based Access Control
ACE	Access Control Entry
ACL	Access Control List
AES	Advanced Encryption Standard (a symmetric key based data encryption algorithm)
AH	Authentication Header
ASP	Active Server Pages
CA	Certification Authority
CAVP	Cryptographic Algorithm Validation Program
CBAC	Capability Based Access Control
CBC	Cipher Block Chaining
CCM	Counter with CBC Message Authentication Code
CD	Compact Disc
CGI	Common Gateway Interface
CMAC	Cipher-based Message Authentication Code
COTS	Commercial-Off-The-Shelf
CRL	Certificate Revocation List
CSRF	Cross Site Request Forgery
DAC	Discretionary Access Control
DDoS	Distributed Denial of Service
DH	Diffie Hellman
DMZ	Demilitarized Zone
DNS	Domain Name Service
DoS	Denial of Service
DVD	Digital Video Disc
EAC	Election Assistance Commission
FTP	File Transfer Protocol
HAVA	Help America Vote Act
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
I&A	Identification and Authentication
ICMP	Internet Control Message Protocol
ID	Identifier
IDS	Intrusion Detection System
IE	Internet Explorer (Microsoft web browser application software)
IETF	Internet Engineering Task Force

IMAP	Interactive Mail Access Protocol
IP	Internet Protocol
IPS	Intrusion Prevention System
IPSec	Internet Protocol Security
IT	Information Technology
KBA	Knowledge Based Authentication
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MAC	Mandatory Access Control
MAC	Message Authentication Code
MITM	Man-In-The-Middle
NBA	Network Behavior Analysis
NCP	National Checklist Program
NetBIOS	Network Basic Input/Output System
NFS	Network File System
NIS	Network Information System
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
NVD	National Vulnerability Database
OCSP	Online Certificate Status Protocol
OTP	One Time Password
PBAC	Privilege Based Access Control
PBX	Private Branch Exchange
PII	Personally Identifiable Information
PIN	Personal Identification Number
PKI	Public Key Infrastructure
POP	Post Office Protocol
RADIUS	Remote Authentication Dial In User Service
RAID	Redundant Array of Inexpensive Disks
RBAC	Role Based Access Control
RDBMS	Relational Data Base Management System
RFC	Request For Comment (series of standards developed by IETF)
RSA	Rivest, Shamir, Adelman (a public key cryptography algorithm)
SAN	Storage Area Network
SASL	Simple Authentication and Security Layer
SCAP	Security Content Automation Protocol
SHA-1	Secure Hash Algorithm (Version 1) – a FIPS Standard
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol

SP	Special Publication (A National Institute of Standards and Technology publication series)
SQL	Structured Query Language
SSH	Secure Shell
SSI	Server Side Include
SSL	Secure Socket Layer
SSN	Social Security Number
TCP	Transmission Control Protocol
TDES	Triple Data Encryption Standard (a symmetric key based data encryption algorithm)
TLS	Transport Layer Security
UDP	User Datagram Protocol
UOCAVA	Uniformed Overseas Citizens Absentee Voting Act
URL	Uniform Resource Locator
USB	Universal Serial Bus
VPN	Virtual Private Network
WORM	Write-Once Read Many
XSS	Cross-Site Scripting

10 Glossary

Access Control	The process of granting or denying specific requests: 1) for obtaining and using information and related information processing services
Access Control Entry (ACE)	An entity and the type of permission granted to that entity, contained on an Access Control List
Access Control List	A register of: 1. users (including groups, machines, processes) who have been given permission to use a particular system resource, and 2. the types of access they have been permitted.
Certificate	A digital representation of information which at least 1. identifies the certification authority issuing it, 2. names or identifies its subscriber, 3. contains the subscriber's public key, 4. identifies its operational period, and 5. is digitally signed by the certification authority issuing it.
Certificate Revocation List (CRL)	A list of revoked public key certificates created and digitally signed by a Certification Authority.
Certification Authority (CA)	A trusted entity that issues and revokes public key certificates
Commercial-Off-The-Shelf (COTS)	Hardware and software IT products that are ready-made and available for purchase by the general public
Cross-Site Request Forgery (CSRF)	A type of web exploit where an unauthorized party causes commands to be transmitted by a trusted user of a website without that user's knowledge
Demilitarized Zone (DMZ)	A network created by connecting two firewalls. Systems that are externally accessible but need some protections are usually located on DMZ networks
Denial of Service (DoS)	The prevention of authorized access to resources or the delaying of time-critical operations.
Discretionary Access Control (DAC)	The basis of this kind of security is that an individual user, or program operating on the user's behalf is allowed to specify explicitly the types of access other users (or programs executing on their behalf) may have to information under the user's control.
Distributed Denial of Service (DDoS)	A Denial of Service technique that uses numerous hosts to perform the attack
Hash-based Message Authentication Code (HMAC)	A message authentication code that uses a cryptographic key in conjunction with a hash function.
Identification and Authentication (I&A)	The process of establishing the identity of an entity interacting with a system
Intrusion Detection System (IDS)	Software that looks for suspicious activity and alerts administrators.
Intrusion Prevention System (IPS)	System which can detect an intrusive activity and can also attempt to stop the activity, ideally before it reaches its targets.
Man-In-The-Middle (MITM)	An attack where the adversary positions himself in between the user and the system so that he can intercept and alter data traveling between them.
Mandatory Access Control (MAC)	Access controls (which) are driven by the results of a comparison between the user's trust level or clearance and the sensitivity designation of the information.

Message Authentication Code	A cryptographic checksum on data that uses a symmetric key to detect both accidental and intentional modifications of the data.
Metacharacter	A character that has some special meaning to a computer program and therefore will not be interpreted properly as part of a literal string.
Network Behavior Analysis	Examination of network traffic to identify threats, usually as part of an IDS or IPS.
Nonce	A value used in security protocols that is never repeated with the same key. For example, challenges used in challenge-response authentication protocols generally must not be repeated until authentication keys are changed, or there is a possibility of a replay attack. Using a nonce as a challenge is a different requirement than a random challenge, because a nonce is not necessarily unpredictable.
Out Of Band	Used to refer to information transmitted through a separate communications channel.
Personally Identifiable Information (PII)	This is information which can be used, alone or in combination with other information, to distinguish or trace an individual's identity.
Public Key Infrastructure (PKI)	A set of policies, processes, server platforms, software and workstations used for the purpose of administering certificates and public-private key pairs, including the ability to issue, maintain, and revoke public key certificates.
Token	Something a user possess and controls used to authenticate the user's identity.
Transport Layer Security (TLS)	An authentication and encryption protocol widely implemented in browsers and web servers. HTTP traffic transmitted using TLS is known as HTTPS.
UOCAVA	Uniformed Overseas Citizens Absentee Voting Act
UOCAVA Systems	Information technology systems which enable uniformed and overseas United States citizens to vote.
XSS	Cross-Site Scripting (XSS) is a security flaw found in some web applications that enables unauthorized parties to cause client-side scripts to be executed by other users of the web application