NISTIR 7682

Information System Security Best Practices for UOCAVA-Supporting Systems

Andrew Regenscheid
Geoff Beier
Santosh Chokhani
Paul Hoffman
Jim Knoke
Scott Shorter

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

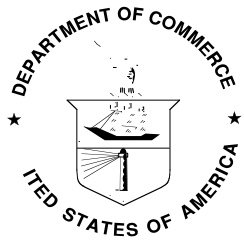
[This page intentionally left blank.]

NISTIR 7682

Information System Security Best Practices for UOCAVA-Supporting Systems

Andrew Regenscheid
Geoff Beier
Santosh Chokhani
Paul Hoffman
Jim Knoke
Scott Shorter

September 2011



U.S. Department of Commerce
Rebecca M. Blank, Acting Secretary

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

[This page intentionally left blank.]

ACKNOWLEDGEMENTS

The authors, Andrew Regenscheid of NIST, Paul Hoffman of the VPN Consortium, 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, and John Wack of NIST, Matt Masterson, and James Long of the Election Assistance Commission, and Carol Paquette, Mark Skall, Tom Caddy and Karen Scarfone for their feedback on drafts of this document.

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 voters who are away from their residences to obtain information and descriptions about voter registration and voting procedure, and how to request, 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 postal mail delivery around the world, Information Technology (IT) systems can be used to facilitate 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.
- 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 consequences for the integrity of the election and 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 remote voting but does not define a specific architecture or configuration.

Component and system selection guidance

The security features outlined in this document rely on components that are frequently, but not always, found in commercially available IT products. In some cases, a product may appear to offer a feature but fails 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 to support UOCAVA voting understand these components and the features required to implement them both when purchasing a turn-key system or selecting components to assemble into a system.

Component and system configuration guidance

In most cases, the IT products used to support 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. Second, this document lists 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. Jurisdictions will have customized their configurations depending on the architecture or implementation of their remote voting system.

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 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 system 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 administrators when assigning individuals to fulfill roles defined by the system design.

1 Introduction

State and local election officials have various responsibilities under the Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA), many of which involve information security. These state and local jurisdictions have begun to use information technology (IT) systems and the Internet to facilitate UOCAVA voting; for example, they are required to make voter registration, absentee ballot applications, and general election information available electronically. These IT systems are often used to distribute election information to voters, send and collect voter registration and ballot request forms, and deliver blank ballots.

1.1 Purpose and Scope

This document provides voting jurisdictions with security best practices for IT and networked systems that are used to support UOCAVA voting by sending or receiving voter registration or ballot request materials, or by delivering blank ballots to voters. Some of these best practices are unique to voting systems, but most are similar to, or the same as, best practices in IT and networked systems in general. For the latter, this document summarizes and points to other security-related documents published by NIST.

This document follows NISTIR 7551, *A Threat Analysis on UOCAVA Voting Systems*, which documents the threats to UOCAVA voting systems using electronic technologies for all aspects of the remote voting process. While NISTIR 7551 discusses 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 security best practices to help jurisdictions create UOCAVA voting systems based on security practices used in other IT applications.

The practices described in this document are broadly applicable to voting systems supporting UOCAVA that rely on IT systems, most of which run over the Internet. They supplement the other safeguards already in use in those voting systems, and possibly replace those practices that are out of date.

There are some topics not covered in this document. Remote voting techniques such as remote voting kiosks and voting over the Internet from personal computers, and using secure email such as S/MIME and OpenPGP for electronic ballot return, are out of scope because they are rarely used in UOCAVA voting systems and have a very different set of security challenges than the systems described here.

1.2 Intended Audience

This document is aimed at IT administrators who are implementing or maintaining systems that support UOCAVA. 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. The reader is assumed to have a medium to high degree of technical literacy of computers and networking.

This document refers to system designers, implementers, operators, auditors, and administrators as roles relative to the system used to support UOCAVA voting. Those terms 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.

2 Overview of UOCAVA-Supporting Systems

2.1 System Overview

There are many different ways to support UOCAVA voters, and each jurisdiction must put together its own system for such support. This document covers some common parts of UOCAVA system architectures, and shows how to secure those parts against both normal and extraordinary threats. The two components that are covered in most detail are Internet-based or Internet-assisted delivery of blank ballots and voter registration.

2.1.1 Voter Registration and Ballot Request

In most jurisdictions, overseas and military voters must register in the jurisdiction where they are eligible to vote absentee in order to be qualified to vote in future elections, although some jurisdictions waive registration for military voters. A common method for voters to submit this information is the Federal Post Card Application (FPCA), a standard federal form that all states are required to accept. In addition, each state has its own registration form that reflects its specific registration requirements. Both the state specific forms and the FPCA request the following information from voters: name, date of birth, sex, race, home address and political party preference. They also ask for various forms of contact information, including telephone number, fax number, e-mail address, and mailing address.

Many jurisdictions make voter registration and ballot request forms available on their web sites, or are willing to e-mail them to voters upon request. Depending on local procedures and state law, some jurisdictions will accept completed voter registration or ballot request forms from voters over e-mail or allow voters to upload scanned forms to web sites. A growing number of jurisdictions are creating web sites that allow voters to fill out a web form to submit updates to their voter registration information. In these cases, proper operational, managerial and technical security controls must be implemented to ensure sensitive personally-identifiable material from voters is kept secure.

2.1.2 Electronic Ballot Delivery

Blank ballots are sometime created in electronic format and delivered to voters electronically. In UOCAVA environments, electronic delivery of ballots over the Internet overcomes many of the obstacles of delivering paper ballots in a timely and verifiable fashion. Such ballots are commonly formatted as PDF files which the voter can print locally and return by postal mail.

Blank ballots can be delivered to voters by email or over the Web. The choice of how to deliver ballots involves many variables. Some considerations include:

- Some jurisdictions have recent email addresses of non-local voters, making email delivery possible.

- Ballot availability may be restricted based on the ability to authenticate the voter
- Local policy might require that ballots be encrypted for delivery

Note, this document only covers delivery of blank ballots to voters, not electronic voting itself (i.e., ballot return). Thus, it is expected that voted ballots described within this document will be printed and sent back to the jurisdiction via postal mail.

2.2 IT and Networking Component Overview

Different voting systems have different computing and network components, but most have many components in common. They include:

- Computers used as web and email servers (as well as other public services)
- Server software and jurisdiction-specific configurations
- Network devices such as routers and firewalls
- Identification and authorization systems
- Shared networks, particularly the Internet
- Desktop and laptop servers used to manage other elements of the voting system

These elements are described in more detail in the remainder of this document based on their interactions with the security requirements discussed.

3 Identification and Authentication

A primary goal of voting systems is to ensure that every ballot is cast by a legitimate voter. 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 can be performed on the system and what information may be accessed. In addition to authenticating voters, every IT system used to support UOCAVA voting will have other classes of users, particularly administrators, who have 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. Having insufficient authentication for a system administrator can have a much more negative effect on an election than having insufficient authentication of a particular voter because the system administrator has heightened privileges that allow them to affect the validity of votes from many voters.

Identification and authentication in face-to-face environments are quite different than in electronic environments. In most cases, electronic authentication (particularly over the Internet) gives much less assurance than in face-to-face environments. For example, seeing a person who is holding a government-issued photo identity card such as a drivers license or passport gives much more assurance than seeing a copy of the photo identity card that was emailed. It should be noted that face-to-face voting normally employs much less stringent verification on government-issued identification than other environments, such as in aviation security screening. Still, physical interaction with physical identification such as drivers' licenses gives a greater opportunity for better authentication than online systems.

In this discussion, the person who is asserting his or her identity is called the *claimant* and the party trying to assess the authenticity of the identity is the *verifier*. NIST SP 800-63 Rev. 1, *Draft Electronic Authentication Guideline*, provides guidelines for implementing electronic authentication that is used over open networks such as the Internet. It defines levels of assurance that are associated with various forms of authentication and lists the types of authentication that a verifier might use for authenticating a remote user's identification. Electronic authentication relies on *tokens*, which are either information that is only known to the person and the verifier, or a hardware device that can generate information that the verifier knows can only come from that device. A summary of the types of tokens that could be used in UOCAVA systems is:

- Handwritten signatures – This is the same type of token used by jurisdictions to authenticate local voters. Because it is easy to photocopy

signatures, it is common to require that signatures used for authentication must be original signatures, not copies (i.e., signatures used for authentication purposes must be "wet signatures").

- Passwords – These are commonly short strings of letters, numbers, and possibly punctuation that the claimant is expected to memorize or to have stored in a password management tool. Section 4.3 of NIST SP 800-118, *Guide to Enterprise Password Management*, describes password management tools and their uses. Numeric PINs are a type of password that are all-numeric and often shorter than typical passwords.
- Identifying prompts – These are usually questions whose answers are known to few people, including the claimant, such as "what city were you born in" or "your first pet's name," and are often only used for low-value authentication.
- Printed sets of secrets – This might be a sheet of paper or a small booklet that is unique to each claimant and which contains numerous secret values. The verifier prompts the claimant to reveal one of the values by its position (such as "enter the number that is in the second column in the tenth row of page 5").
- Out-of-band hardware access – This type of authentication relies on the claimant having their own hardware that the verifier can initiate communications to. For example, if the claimant registered a phone number with the verifier ahead of time, the verifier can tell the claimant a secret, and then call the claimant on the registered phone number and ask for the secret.
- Single-factor One Time Password Device –These hardware devices spontaneously generate new passwords on-demand or at set intervals, and display them on the device. Users of single-factor one time password devices do not need to unlock the device before it will generate passwords. These devices are typically used in combination with other types of authentication tokens. For example, the verifier might authenticate the claimant by asking for the correct memorized password and one-time password.
- Multi-factor One Time Password Device – These hardware devices generate new one time passwords only after being unlocked by the claimant. For example, the claimant might unlock the multi-factor one time password device by entering a PIN directly onto the device, or using a biometric (e.g., fingerprint) reader on the device. Typically the one time password generated are displayed on the device and manually input into another system by the claimant for transmission to the verifier.
- Cryptographic Software – These are cryptographic keys that are stored on disk or some other unprotected media that typically must be unlocked before use (e.g., using a password). For example, the cryptographic keys might be stored in an encrypted format, using passwords to decrypt them.

Authentication is accomplished by having the claimant interact with the verifier using a cryptographic protocol.

- Cryptographic Hardware – These devices contain a protected cryptographic key that typically must be unlocked before use (e.g., using a PIN or biometric). These devices usually use the cryptographic key to digitally sign challenges from the verifier. A smart card is a common type of a hardware cryptographic hardware device.

Agreeing on the type of token that will be used for future authentication is called *issuance*. Issuance normally happens in person because of the chicken-and-egg problem of not being able to authenticate a request for issuance. However, one can use one token to authenticate a request for another. It is quite common to use a handwritten signature as authentication for a request for a token that can be used for electronic authentication when in-person issuance is not possible.

3.1 Authenticating People

The jurisdiction is the verifier when authenticating voters and people who act in administrative role. The jurisdiction and the claimant must agree on the mechanism for authentication before a voter asks to perform an action that requires authentication (such as changing their registration information).

All authentication mechanisms require that the verifier keep some record of what was presented by the claimant (e.g., the handwritten signature) or given to the claimant (e.g., the one time password generator) at the time of issuance. When authenticating, the verifier compares what is presented with that original information.

If the authentication mechanism is a handwritten signature (as in the case of non-electronic voting), the issuance information is an original signature or a copy thereof. Even if someone who wants to impersonate the voter sees the signature or copy, they still have to reproduce it in a wet-signed duplicate, which is considered hard; this is why bank checks have worked as well as they have for over a hundred years. Note, however, that banks currently do not rely solely on visual inspection of signatures for validation of checks, and modern signature verification tools use machine learning algorithms that are rarely used in voting contexts.

The following shows likely considerations for authenticating voters with the different types of authentication systems:

Authentication type	Security Considerations	Deploying and Verifying
Handwritten signatures	Currently universally used for in-person voting transactions, thus strong enough for remote transactions.	If a ballot or information update form is delivered electronically, the claimant needs to have access to a printer. The claimant needs to be able to send the wet-signed paper to the verifier.
Passwords	Users often use the same passwords at multiple sites and/or choose weak passwords, making impersonation attacks fairly easy. No hardware is required, making this the easiest electronic token available.	Password can be chosen by the claimant or the verifier. Storing or transmitting unencrypted passwords makes attacks easier.
Identifying prompts	Generally not used for voting systems because the answers may be easy to guess or may be easy to determine from public systems.	Prompts need to be chosen by the verifier. Storing unencrypted prompt responses makes attacks easier. Normally more than one type of prompt is used in a single system.
Printed sets of secrets	Can be made secure against impersonation attacks by having the secrets be at least 40 bits long; these secrets are still easy to type.	Verifier must get the printed material to the claimants, and claimants must have the material available when asserting their identity. Storage of secrets and prompts should be encrypted.
Out-of-band hardware access	The verifier must assume that the hardware being accessed is still controlled by the claimant. For example, if the claimant has lost their cell phone, the new possessor can impersonate the claimant.	A second communication system (such as a phone system) must be deployed and available to people who are doing the verification.

Single-factor One Time Password Device	These are usually small, hand-held cards and therefore can be lost or stolen. If these used as the only authentication factor, the new possessor could impersonate the claimant. Therefore, these should be used with another authentication factor, such as a memorized password.	The claimant needs to be able to receive a short prompt and, within less than a minute, access the device and repeat back a short message from the device to the verifier.
Multi-factor One Time Password Device	These are usually small, hand-held cards and therefore can be lost or stolen. The new possessor has to be able to unlock the device (e.g., by guessing the PIN) in order to impersonate the claimant.	The factor used to unlock the device must be set prior to deploying the device. This could be having users set or memorize a PIN, or having the device learn a biometric.
Cryptographic Software	The security of the authentication mechanism depends on claimants keeping their private keys secret.	The claimant needs to possess the private key, and the verifier needs to trust that the public key associated with the private key belongs to the claimant. Private keys are usually protected with passwords.
Cryptographic hardware devices	These are usually small, hand-held cards and therefore can be lost or stolen. Many of these cards are protected with PINs or passwords; the new possessor has to be able to guess the password in order to impersonate the claimant. When implemented properly, this is a very strong authentication mechanism.	The claimant needs to have a device that reads the cryptographic device (e.g., a smart card and card reader) connected to the computer they are using while authenticating.

3.2 Authenticating Voters

A potential voter's identity needs to be authenticated before they can cast a ballot in an election. Election jurisdictions have always had methods for identifying and authenticating voters at polling places. Voting remotely, such as is enabled by UOCAVA, changes the ways that people are identified because the voter is not seen in person. Jurisdictions have typically authenticated absentee ballots submitted by UOCAVA voters using hand signatures, but may use forms of electronic authentication as they deploy electronic and Internet-based delivery methods for election materials. Establishing trusted agents to perform in-person ID verification for voter credentialing for remote (particularly overseas) voters is difficult and may be beyond the capabilities of a particular jurisdiction office.

As described in Section 1, this document does not cover the case of electronic ballot return by voters. Jurisdictions may, however, require authentication of a person's identity for actions other than voting. For example, jurisdictions may require authentication of identity before allowing someone to change the information stored for a registered voter. While it is more common to authenticate marked ballots once they've been returned, some jurisdictions may wish to also authenticate potential voters prior to sending them blank ballots. For many jurisdictions, remote electronic authentication of voters will serve as a secondary authentication mechanism, with handwritten signature verification on returned ballots serving as the primary authentication mechanism.

Like banking web sites, most jurisdictions use passwords for authentication, even though these are considered fairly weak in the security community. Passwords are familiar to users, do not require use of special hardware by the voter, and can be used in a variety of locations. The security risk of using passwords for authentication is high but can be mitigated. NIST SP 800-118, *Guide to Enterprise Password Management*, describes the use of passwords; Section 3 of that document describes threat mitigation in great detail. As noted there, one of the best mitigation strategies is for passwords to be assigned by the verifier because the verifier can use rules for creating passwords that are likely to be much more secure than those that are typically chosen by the people who will use them.

If the authentication mechanism is a password, the jurisdiction has multiple choices for how to store the issuance information. They can store the password just as it was entered, but if the file in which the password is ever compromised by an attacker, that attacker can impersonate the voter with no effort at all. Because of this, most security-aware organizations who store passwords for verification do so by repeatedly encrypting the password with another value. If an attacker accesses this file, they must perform much more work to retrieve the password.

3.3 Authenticating System Administrators and Election Officials

The tradeoffs for authenticating people who manage voting systems are quite different than those for authenticating voters. Many of the types of device-based

tokens that are difficult in practice to distribute to voters, particularly remote voters, have much better security properties than passwords. Any small difficulty associated with distributing and administering these better mechanisms may be outweighed by their better security. That is, even if it is not terribly convenient for a system administrator to need to use a device-based authentication mechanism, doing so protects a system that itself protects the validity and secrecy of elections.

Note that different voting systems allow different types of authentication tokens. Many (but, unfortunately, not all) systems allow one or more types of strong authentication for administrative access. Jurisdictions that produce their own voting systems can choose one or more of these types of authentication in their designs. It is important to remember that role-based authorization (such as giving different rights to a system administrator than to an auditor) can be based on different types of authentication; people whose roles require less security can use authentication mechanisms that are easier to deploy.

3.4 Authenticating Jurisdiction-Administered Servers

Users need to authenticate the servers that they connect to so they can be sure that the information they receive comes from the source that they expect. Essentially all server authentication today is done with digital signatures through the TLS security protocol.

When a voter uses the Internet to connect securely to a server that they think is administered by a jurisdiction, they use their web browser and TLS. The first steps in that protocol are to authenticate the server to the user by comparing the domain name that the user accessed with the name in the certificate presented by the server in the TLS handshake and to be sure that the server knows the secret key associated with the certificate. The voter's browser then checks if the certificate is issued by a trusted certificate authority (CA) and, if so, allows the user to proceed securely to the intended web site. Note that there is a serious but unsolved problem with the extremely large number of CAs and the fact that CAs do not incur almost any liability if they issue erroneous certificates that could mislead voters into trusting that they were on a jurisdiction's site when in fact they were led somewhere else.

4 Host Protection

The two major parts of an electronic voting system that need to be protected are the computers (hosts) on which processing happens and the network that is between those computers. This section covers how to protect the computers; Section 5 covers how to protect the network. Both parts of an electronic voting system also have ongoing protection such as audits and policy reviews; these are covered in Section 6.

4.1 Types of UOCAVA Hosts

UOCAVA hosts fall into two broad categories:

- Hosted voting system servers – Voting system servers are those with which voters interact. A common example of these is web servers that voters connect to from their personal computers to get voting information, request paper ballots, get electronic ballots, and update their registration information. If a jurisdiction contacts voters through email, the email server used to send messages would also be a hosted voting system server.
- Management stations – These are systems that only jurisdiction IT and network administrators interact with. Typically, these hosts are used to manage and monitor hosted voting system servers, networks, and personal computers used by jurisdiction employees. Note that these management stations may manage and monitor both UOCAVA and non-UOCAVA systems at the same time.

Both hosted voting system servers and management stations may be local to the jurisdiction or may be remote (particularly, overseas) but controlled by the jurisdiction through contracts with service providers. In fact, if the jurisdiction has outsourced some of its IT functions, the management stations are likely to be owned and controlled by contracted company, not the jurisdiction.

The difference between the two types of hosts is due mostly to who can access them. Hosted voting system servers are by design accessible to Internet users, whereas management stations are often on networks that are protected by firewalls. Note that not all management stations are on protected networks: a common example is a PC used by a jurisdiction IT staff to manage systems from home or while travelling.

Protection of personally-owned PCs used by voters and remote voting kiosks are not covered here. The vulnerabilities associated with these systems, and the mitigations for those vulnerabilities, are quite different than what is described in this document.

4.2 Protecting Voting Servers and Management Stations

Voting system servers and management stations can be vulnerable to a wide variety of attacks from the Internet. Servers are normally at fixed, easily-determined locations, which makes a prolonged attack easier to mount. Management stations at fixed locations that are not protected by a firewall have a

similar attack profile. In fact, management stations that are fully protected from the Internet can still be the target of attack if another computer on the protected network is compromised, such as by malware that was delivered in email or by web browsing.

Jurisdictions that have voting servers and management stations that can be reached from the Internet need to assume that attackers will want to take control of these computers, even if the attacker is uninterested in the voting aspect of the system.

4.2.1 Management Access Control

Computer management entails any modification to the computer that changes the way that the computer operates. *Management access control* is restricting who can manage the computer to a limited number of known people.

For example, on a PC used in a personal work setting, setting the electronic clock back by an hour will have minimal impact on the use of the computer; on a server that is handling requests for electronic ballots, making such a change (even with auditing) could have huge effects on the security of the voting system. Other management tasks can similarly be benign or serious; changes such as rebooting, patching the operating system, limiting the ability of a user to write files over a certain size, or even where DNS resolution information is obtained need to be considered in light of all the operational uses of the computer. Similarly, one might allow certain people rights to change only a few settings on a server, but it is almost impossible to prevent anyone from rebooting a computer if they have physical access to it.

Every server has at least one way to manage it, and often has at least a few. Some servers are managed directly on the server themselves, using keyboards and monitors attached directly to the servers. More and more, however, servers are managed by workstations (often regular PCs) that access them through local networks and/or the Internet. In the latter case, management access control for the server also means access control for any workstation that can manage the server through its remote interface. Thus, the scope of access control is often much wider than just that of the server itself. It is important to recognize that the management of any particular computer can be done in many ways, not just one.

Controlling management of servers requires attention to at least three areas:

- Minimize the number of users who can manage a computer to the bare minimum needed to reliably maintain the system. This is not as simple as it initially sounds: having too few administrators makes recovering from emergencies difficult because it may be hard to reach anyone who has management authorization, but allowing too many increases the risk that any one might be impersonated by an attacker.
- Use strong authentication for every user who is allowed to administer the computer. Use of passwords that might be easily guessed or copied from

other servers to which an attacker may have access is not strong enough for servers of high value.

- Record all logins to a server in a way that even an administrator cannot easily change. Anyone who can impersonate a user who has administrative privileges can often make changes that are difficult to trace unless reliable audit trails are kept.

Access control goes well beyond these three topics, but implementing them greatly reduces exposure to typical attacks on servers and makes such attacks easier to detect and possibly fix.

4.2.2 Anti-malware

All server operating systems are susceptible to malware, although there is a wide range of vulnerability. Malware can be spread through many mechanisms, such as exploiting security holes in web browsers, mail attachments, and open services that have programming errors. The goal of almost every attacker is to get administrative access to the server; from there, they can change the system to allow later access.

Server operating systems that have a history of exploitation by malware usually have anti-malware available from the operating system's vendor, other vendors, or both. Using anti-malware on such systems is necessary for good server hygiene. However, installing anti-malware is usually barely sufficient for the task. Because attackers are constantly changing their malware, constantly keeping this anti-malware software up to date is also necessary. Some anti-malware software has daily updates, and all servers that use that software should update whenever there is a new release.

Note that not all server operating systems have significant malware problems, and thus there is little market for anti-malware on those operating systems. However, all operating systems have vulnerabilities that are discovered after release, and thus it is still necessary to perform updates on a regular basis. This is similar in concept to updating anti-malware, although the mechanism for updating operating systems is usually more cumbersome than updating anti-malware. Closing known vulnerabilities helps prevent exploitation by new malware that would not be detected by even by an up-to-date malware scanner.

Management stations are often normal PCs running specialized software that controls the voting servers. Normal PCs are often susceptible to the wide range of malware infecting the Internet. This leads to two main strategies for preventing management stations from getting and passing along malware: run anti-malware conscientiously and restrict the use of any software other than the management software.

4.2.3 Configuration Management

Changes to the configuration of a voting server can have significant consequences on all aspects of the voting system. For example, a change to the networking software could cause some previously-acceptable communication to be rejected

and previously-unacceptable communication to start being accepted. Another example is adding a piece of additional monitoring software: such a seemingly-benign change could slow the system significantly and/or possibly block the monitoring of existing software.

NIST SP 800-128, *Guide for Security Configuration Management of Information Systems*, provides guidelines for managing the configuration of systems such as servers and the networks on which they run. It emphasizes the need to keep records of baseline configurations (known-good starting points in the lifetime of a system) and maintaining configuration management plans, particularly with respect to system security. Section 2.2 of SP 800-128 gives a good overview of the process of configuration management.

In the context of a voting server, a "configuration change" could be almost any change to the settings and applications running on the server, even those not necessarily associated with the voting software on the server. In addition, hardware changes, such as adding new memory or changing the network switch to which the server is connected, constitute configuration changes.

Thus, it is critical to start with a configuration that is both secure and is proven to work well as a whole (that is, all the software is known to work together). When the jurisdiction is sure that this configuration is correct, it is marked in the configuration management system as the baseline and changes to this baseline are then logged. NIST has created a set of checklists and benchmarks for a wide variety of software that can be used for creating baseline configurations. The checklists can be found on the NIST web site at http://checklists.nist.gov/, and the methodology used to create the checklists is described in NIST SP 800-70rev1, *National Checklist Program for IT Products – Guidelines for Checklist Users and Developers*.

Further, it is critical to always track every change to any configuration on a voting server, even if the change initially seems inconsequential. The type of planning needed for this level of tracking is described in section 3.1 of SP 800-128. In order to be sure that all changes, even innocent-looking ones, are tracked, it is essential to limit the people allowed to make any sort of software change to those who understand configuration management and participate in the management tracking that has been instituted for the system in question. Normal software updates for both operating systems and application software inherently cause changes to a configuration. This does not mean that they should not be done, just that every such change be done under the control of the configuration management system with the ability to roll back to a known-good state if the changes stop the system from performing its task or inadvertently reduces the security of the system.

If new problems appear on the server, even after other changes have been applied, being able to look through the configuration change log can help pinpoint the changes that caused the problems; such monitoring is covered in section 3.4 of SP 800-128. There are many software packages that can be used to log configuration changes, but even keeping dated notes in a text file is better than

nothing. Any configuration management should be done on a server other than the one that is being tracked, or at least the change database should be backed up to a different server so a change that is disastrous does not also take down the configuration management system itself.

Note that some of the logging discussed here might also need to be audited. Even if the configuration data itself does not need to be audited separately, it at least made available to those who are auditing the overall system in which the computers participate. It is good to work with auditors to plan for audits well before they happen in order to reduce surprises during the audit process that will force major configuration changes.

4.2.4 Lifecycle Management

The lifecycle of servers and management stations is often easy to ignore, but it is an important to monitor as part of the protection of the systems. In some organizations, lifecycle management is also considered part of the configuration management of a system. NIST SP 800-64rev2, *Security Considerations in the System Development Life Cycle*, describes the processes that lead to sound lifecycle management.

Lifecycle management involves many people. Section 2.3 of SP 800-64rev2 lists the many people in an organization that might be involved with lifecycle management for hardware and software.

The lifecycle of the hardware portion of a server or PC includes acquisition, modification, and decommissioning. Hardware acquisition is usually not an important step, although some jurisdictions might require only US-manufactured hardware systems. Hardware modifications such as adding RAM or upgrading a hard drive can have important ramifications on the software that is running on the computer, and thus needs to be logged.

Hardware decommissioning is often the most important part of lifecycle management in that all data retained in the hardware must be destroyed before the hardware is disposed. A safe way to do this destruction is to do a secure full-disk erase of all hard drives in the system. Simply erasing all "user data" has been repeatedly proven to be an insufficient protection against exposure of sensitive data on systems. Many operating systems retain user data in "system data" files that would normally not be deleted if only deleting user data. Section 3.5 of SP 800-64rev2 lists many steps in decommissioning that are often overlooked.

The lifecycle of the software portion of a server or PC includes acquisition and modification, although rarely includes decommissioning. In this case, "acquisition" consists of two phases: purchasing and installation. Protecting a server during purchase is usually not an issue. However, Section 3.2 of SP 800-64rev2 describes how to perform risk assessment which is often overlooked in software purchasing. The methods used for installing purchased software, however, can have implications on security if the new software affects how previously-installed software performs. For example, adding software that

purposely restricts the ability to use other software, such as anti-malware, can cause security problems if the blocked software is actually part of the security setup for the server. Thus, it is very important to monitor the logs of all server software after installing new software to be sure that the older software continues to perform as expected.

Software updates (sometimes called *patches*) can affect not only the updated software but also other software on the system. This is particularly true for updates made to operating system software. Most modern operating systems include utilities that often have security holes and thus will be updated when general operating system updates are applied. These changes, which can be critical to the security of the server, may have negative effects on other software, particularly software that requires a particular version of the operating system or its utilities. In some ways, updating software is similar to adding new software to a system, and it is very important to monitor the logs of all server software after updating software to be sure that the all software on the system continues to perform as expected. Thus, the considerations from Section 3.4 of SP 800-64rev2 on the management of operations become particularly relevant.

Some electronic voting systems come as integrated solutions that contain both the hardware and software. The lifecycle management for these unified systems is in some ways easier because there is a single target for management. However, some of the operations that are performed by system integrators is harder to track and can have serious effects on the security of the systems. Unified solutions should not be considered "better" because of potential reductions in lifecycle management needs; instead, they must be seen as having different needs for lifecycle management.

4.2.5 Secure Backup

Making copies of the software and data on a voting server is a double-edged sword. It is required for stability but it exposes all the software and data to possible compromise. That is, each time there is a back up of a critical part of a voting server, that backup needs to be secured as well as the original server. This tradeoff can cause some organizations to not back up often enough to be useful in an emergency, but it can also cause other jurisdictions to use less-than-adequate security for their backups.

The security policies that apply to the voting server must also apply to all backups of sensitive data and applications on the voting server. This includes deciding who has physical access to the backups, who is authorized to read the data on the backups, who can make subsequent copies of the backed-up material, and who can read the data itself. Duplicating the policies for the original data for backup data is often easier than enforcing those policies because many organizations have different people handle their original data and their backups. In such cases, however, doubling the number of people who have access to backups may significantly increase the risk of the backup data being improperly exposed.

In order to assess the security of their backup system, jurisdictions need written backup procedures as part of the operational step of lifecycle management. These procedures list not only how the backup is made (such as what data is backed up and on what media), but also where and how the backups are stored and who has physical access to the backup media.

4.2.6 Web Server and Application Security

Many of the servers used by jurisdictions for assisting voting are web servers. Web servers are different than other Internet servers in that potential attackers have studied web servers in greater detail than application-specific servers. Thus, they have all the same security issues as generic servers that are exposed to the Internet, but are susceptible to greater attacks because of the acquired skills of a larger set of attackers.

Because of the widespread use of public web servers, NIST SP 800-42v2, *Guidelines on Securing Public Web Servers*, details the procedures that server administrators should follow in order to reduce the possibility of security flaws. For many organizations, flaws that expose private data (e.g., data associated with voters) are considered very damaging. For voting jurisdictions, flaws that allow an attacker to successfully impersonate a web server can be even more devastating because voters could be given incorrect information about how and where to cast ballots, which in turn can lead to flawed elections and loss of confidence from voters. Following the guidelines in SP 800-42v2, particularly those in Section 5 of that document, can go a long way towards reducing exposure to both errors and attacks on jurisdiction-run web servers.

Good web server hygiene is a complete field unto itself and much of it depends on the software that is chosen for the web server. Not only do different HTTP servers (such as Apache, Microsoft IIS, and Lighttpd) have different exposures to attacks, common additions to web servers (such as the PHP language and content management systems such as WordPress) also present their own attack possibilities. Some of the more important considerations in securing web servers include:

- Apply security patches for the web server software in use as soon as they are available. Web server vulnerabilities are tracked closely by the community of attackers, so applying patches in instances where a jurisdiction's server is vulnerable is critical to maintaining a secure system.
- Similarly, apply security patches for the additional web software packages in use as soon as they are available. These packages are easily detected by attackers and often can open the same types of attack vectors as the web server software itself.
- Constantly screen for cross-site scripting (XSS) vulnerabilities using firewalls and external screening services or web application scanners. Cross-site scripting is a mechanism for inserting scripts controlled by the attacker onto pages hosted on the web server. Their purpose is to gain access to private information that is used by the user's browser, particularly

site passwords and cookies. Most modern web browsers attempt to prevent cross-site scripting attacks by limiting the private information only to trusted web pages. However if an attacker can get their script onto a trusted page, they can masquerade as legitimate page content and access the private information.

- If the server accesses data from an SQL-based database, assiduously check all user input for *SQL injection* attacks. These attacks, which are still quite common on the Internet, look for web sites that pass insufficiently-processed user input to database back-ends and then send carefully-crafted input that will cause exposure of database records, and possibly allow destruction of databases.

Most voting web servers that send or receive sensitive information use the TLS protocol to cryptographically protect connections. TLS requires that every server have a certificate that contains its public key and an assertion from a trusted certificate authority (CA) that the public key is associated with the domain name used for the web server. The certificate used by a web server must not be expired and must be signed by a CA that is trusted by the user. Different web clients have different sets of trusted authorities, and this forces web server administrators to choose authorities that are trusted by all possible users of their secure web server.

A small number of voting jurisdictions use web services in Service Oriented Architectures (SOAs) for processing votes that were received electronically or were manually entered from paper ballots. NIST SP 800-95, *Guide to Secure Web Services*, lists many of the risks of using SOAs, and lists procedures that web services customers should take to protect themselves from loss of data confidentiality.

4.2.7 Email Security

Electronic voting systems may use email for sending ballots, sending election notifications, and other UOCAVA election materials. They may also use email for non-authenticated incoming mail, such as communications between jurisdictions and voting authorities.

Sending and receiving mail uses the SMTP protocol, which does not have any inherent authentication. NIST SP 800-45v2, *Guidelines on Electronic Mail Security*, describes the significant security issues that come with unauthenticated mail sending and receipt. Many SMTP servers support TLS for authenticating the server; that is, the initiator of an email exchange can authenticate the responding SMTP server using TLS with certificates. As long as both parties share trust in the same CA, the initiator can be sure it is communicating with the desired server. There is no common way to authenticate SMTP initiators. Using TLS with SMTP also provides encryption and integrity protection for the SMTP session.

The origin of messages sent over SMTP can be validated with three similar protocols: DKIM, SPF, and SenderID. Of the three, only DKIM is an Internet standard, and it is more widely deployed than the earlier SPF and SenderID

protocols. Note that these protocols do not provide encryption or integrity protection; instead, they only allow the sending organization to assert that mail messages that claim to come from a particular domain name in fact do so.

4.3 Special UOCAVA Host Considerations

4.3.1 Protecting Data at Rest

Jurisdictions often store personally identifiable information (PII), voted ballots, and other private data on drives that need to be periodically backed up. The backed-up data must be protected with the same vigilance as the original data. If the original data is stored encrypted with keys of a certain strength and only usable by certain people, the backup needs to use the same strength keys (or stronger) and have the same access controls (or be even more restrictive).

Maintaining data covers two different topics: preventing unauthorized viewing of private information and maintaining the integrity of the stored data. The latter is extremely important for voting jurisdictions. The same tools used to prevent viewing of private data are also used to prevent changing of stored data by unauthorized parties, namely encryption and access controls. In this case, access control has two parts: access to viewing and updates. Normally, backups of data should never be updated; instead, the data is changed in its original location and a new backup is performed. This helps assure the integrity of the backups and keeps the access control rules clear, namely that people can only create new backups, not modify existing ones.

Protecting backups of data is complicated by the fact many backups are, by design, meant to be kept in a different location than the original data. In order to prevent loss of data due to a physical disaster such as fire in a data center, keeping off-site backups is a standard practice for most organizations. However, it is difficult to maintain the physical security of such backups identically as the original data because there are normally different staff at the storage site. Because of this, off-site backups should be encrypted with keys that are only known to people who have access to the original data.

4.3.2 Protecting Databases

Database servers, and the data they contain, have come under more frequent attacks in recent years. The personal data in registration databases, polling books, and so on, do not at first appear to be of value to typical Internet miscreants. However, all personally identifiable information (PII) can have value when combined with other data, such as stolen credit card numbers.

Protecting database servers is different than protecting web servers in that database servers are usually not directly accessed from the Internet. Instead, they are only accessed using custom programs running on web servers. However, this lack of direct connection to the Internet does not make them at all immune to attack. People looking to dump the contents of databases will try to fill in web forms in ways that will exploit bugs in the custom programs accessing the database servers.

It is common for attackers to try to inject database access commands in text fields in forms, hoping that the controlling programs are not scanning the input carefully before it is passed to the database server. In recent years, these *script injection* attacks have caused databases to reveal a great deal of personal information that the site operators thought was protected. To reduce the likelihood of script injection attacks:

- Rigorously check the values in every field of a web form, looking for any characters that should not be in that type of data, and also looking for patterns that look like database commands.
- Limit the number of fields that allow user input.
- Monitor the logs of the database server, looking for anomalous queries coming from the web server.

4.3.3 Document Delivery Over Fax

Many jurisdictions use facsimile (fax) systems to send and receive forms and voting information to UOCAVA and other remote voters. Nearly all fax transmissions are over standard telephone lines, which means that neither party can protect the network connection. Further, there is no widely-used standard for fax encryption. Thus, information sent by fax is at risk for possible interception or modification. Jurisdictions should carefully weigh the risks of fax transmission of election materials against the possible alternatives prior to using fax to send or receive sensitive information.

Some faxes are sent over the Internet, which would give them the same security properties as other documents sent over the Internet. However, most Internet fax systems are not end-to-end, meaning that the recipient still receives the fax on hardware connected to the phone system.

5 Network Protection

5.1 Types of UOCAVA Networks

The rapid expansion of the Internet and the continuing advancement of networking technologies has made defining particular network configurations more complicated. Networks in UOCAVA environments have the additional attribute of having long-distance components that are often not controlled by the election jurisdiction. This document covers the security practices for three types of networks:

- Links from remote to local systems run by election jurisdictions – These are sometimes dedicated leased lines, but could also be normal Internet links.
- Networks between end users and externally hosted voting systems – Some jurisdictions outsource operation of systems used to support UOCAVA voting. Typically, these systems allow voters to use whatever local Internet connection they have to connect to the voting system, and the voting system is connected over the Internet to the jurisdiction.
- Local area networks (LANs) – The security aspects of these are approximately the same as for other types of networks, although hardware switches can help in segmenting these networks.

5.2 Firewall Devices

In order to have any control of the data flowing through its network, an organization must make sure there are only a small number of connection points between the protected network and other networks. At each connection point, there should be a firewall device that controls both what comes in to the protected network and what goes from the protected network to other networks. NIST SP 800-41rev1, *Guidelines on Firewalls and Firewall Policy*, describes how to choose and deploy the firewalls that protect a network.

Section 3 of SP 800-41rev1 shows many typical network architectures and shows where firewalls fit into the design of protected networks. UOCAVA networks that are controlled by a voting jurisdiction, such as those between remote parts of a voting system or a LAN, are typical of the architectures people think of when deploying firewalls. However, most UOCAVA jurisdictions also must deal with remote users on their own computers accessing parts of a protected network, and thus the remote access to or through a firewall becomes much more important. Placement of firewalls in a network becomes extremely important because openings that are not protected by firewalls can lead to attacks on the network that are difficult to find and fix.

There are many types of firewall devices, some of which are more appropriate for protecting networks of devices that are all controlled by a jurisdiction, but others of which are more appropriate for allowing outsiders (in this case, voters and those interested in registering) to have limited access to some of the computers

in a protected network. Section 2 of SP 800-41rev1 describes each of the types of firewall devices that might be used. Jurisdictions that let voters have access to servers at the border or inside of its networks need to consider how to use web application firewalls and/or firewalls with network access control, and need to design their networks based on those choices.

Modern firewalls are fairly flexible and therefore complex devices. Most firewalls can implement a wide variety of security policies (such as "allow incoming traffic only to these hosts," "block all incoming traffic unless it is from this range of addresses," etc.). Section 4 of SP 800-41rev1 describes firewall policies and how they can be implemented in various firewall configurations.

After a security policy is established, each firewall at the perimeter of a protected network needs to be configured to meet that policy. If a network has multiple places where traffic from outside the network can enter and/or exit, that network needs multiple firewalls, each of which is configured with the same policy. Every firewall has a different method for configuration, which makes implementing multi-vendor networks difficult but not impossible. Even in a single-firewall network, it is important to be sure the configuration of the firewall fulfills all of the parts of the security policy.

It is common for organizations that have systems placed remotely, such as a voting jurisdiction that has overseas servers, to have multiple networks that need to be linked together. This linking is often done using firewalls to segment the network into smaller networks with connections between them. A firewall that inspects the source and destination of each packet can be used to keep a particular set of addresses on just one segment of a network. Segmented networks are not necessarily more secure than a single unified network, but they may be easier to administer. Network segmentation is covered in Section 3 of SP 800-41rev1.

5.3 Encryption and Integrity Protection

Data that passes over public networks can be inspected and/or changed by various types of attackers. Such attacks can have a devastating effect on the organization that runs the network. For example, a voting jurisdiction that runs a UOCAVA network might have voter registration information and possibly even votes (e.g., the contents of mailed-in absentee ballots) passing over its network. An attacker who can change registration or voting information can potentially change the outcome of an election. Even if an attacker can only see this information, revealing that ability can greatly reduce the public's trust in the election jurisdiction.

To prevent such attacks, the public links in a network needs to be protected with cryptography. The two primary types of protection are encryption (the scrambling of data so an attacker cannot understand it) and integrity protection (preventing forged data from being accepted on the network). Encryption and integrity protection are usually provided at the same time. Even though it is technically feasible to have a network that provides integrity protection without

encryption and vice versa, most businesses want both, so they use a single network protection system that provides both.

Different cryptographic algorithms and different sizes of keys offer different levels of protection from attack. Therefore, it is important for an organization to be sure to use both the correct algorithms and the proper size keys for their needs. NIST's recommendations for the algorithms and key sizes that are acceptable to use to protect government data in non-national security systems are found in NIST SP 800-131A, *Recommendation for the Transitioning of Cryptographic Algorithms and Key Lengths*. Section 2 lists the encryption algorithms that are recommended; Sections 3, 5, and 6 lists the recommendations for key sizes. This document augments the advice given in NIST SP 800-57 Part 1, *Recommendation for Key Management – General*, and NIST SP 800-57 Part 3, *Application-Specific Key Management Guidance*. The former describes best practices for key sizes and cryptographic algorithms, while the latter talks about the type of key management that should be used in specific protocols such as IPsec and TLS.

Note, using encryption and integrity protection is appropriate between all types of networks, not just those that are connected with public links over the Internet. "Private" links such as leased lines can be snooped on and have their traffic changed by attackers; the only thing preventing this is a trust that the service provider has configured every router and switch between the ends of the link correctly. If there is a single mistake in the configurations, the traffic may be visible and vulnerable to modification.

5.3.1 Common Cryptographic Protocols

TLS is used to protect web-based traffic (that is, traffic run over the HTTP protocol).[1] TLS is widely used for protecting point-to-point traffic, notably between a web client and a web server. TLS provides both encryption and integrity protection. Many protocols other than HTTP can be protected with TLS as well, but for voting jurisdictions, TLS is almost exclusively thought of in terms of web traffic.

IPsec is the best-known protocol for protecting client-to-network and network-to-network traffic. IPsec is normally thought of as a protocol for virtual private networks (VPNs). A VPN creates a private data network while using public networks (typically the Internet) while providing both encryption and integrity protection to all data in the protected network. Most corporate firewall products include IPsec capabilities, making it much easier for a organization to connect their networks with IPsec at the same time as using firewalls to filter traffic. Note, IPsec can also be used to segment networks with cryptographic protection between each sub-network. Section 4 of NIST SP 800-77, *Guide to IPsec VPNs*, describes in detail how to use IPsec for secure network designs.

TLS/SSL can also be used to create VPNs, which are typically referred to as SSL VPNs. NIST SP 800-113, *Guide to SSL VPNs*, covers the technologies used in typical SSL VPNs. Section 2.2 of that document describes the common use cases

[1] TLS is the successor to SSL, and the two names are often used interchangeably.

for SSL VPNs, which are mostly for roaming access users, not fixed networks such as are typical in UOCAVA environments. In fact, SSL VPNs can be used in some of the same environments where IPsec VPNs are used, but they offer no greater security than IPsec VPNs. Choosing which type of VPN to deploy usually depends on the operational ease of use. If there are many remote access users with unmanaged PCs, SSL VPNs are often appropriate. If the network consists mostly or solely of gateway devices, then IPsec VPNs are usually more appropriate.

S/MIME is the most widely-used standard for digitally signing and/or encrypting email. Many email products come with S/MIME built in, and others have free S/MIME extensions that can be added easily. An email message that is signed with S/MIME before being sent can be checked by the recipient to be sure that no one has tampered with the message. A message that is encrypted with S/MIME prevents someone watching the network traffic from reading the body of the email message (although note the headers of the message are sent unencrypted). OpenPGP is a standard similar to S/MIME, and it is also widely used in email systems.

In order to use S/MIME effectively, both the sender and the receiver must share a mutually-trusted certificate authority (CA). There are many commercial CAs, although only some of them issue certificates for S/MIME. There are also many non-commercial CAs that might used by UOCAVA voters, including the US Government and US Department of Defense CAs. OpenPGP software usually uses a very different trust model than S/MIME, and does not normally have certificate authorities; this makes it harder to use in UOCAVA systems unless the voting jurisdiction already has a trust relationship with numerous other OpenPGP users.

As described in Section 2 of NIST SP 800-49, *Federal S/MIME V3 Client Profile*, different mail software supports different features of S/MIME, and network administrators need to be careful all systems can read and generate the S/MIME messages that are required for any voter information sent through secured email.

5.4 Authentication of Endpoints

Network security relies on at least one party in every communication being fully identified. In many cases, it relies on all parties being identified to the satisfaction of the other parties. In voting systems, these parties are most often human users (such as potential voters, system administrators, and auditor) and computer systems (such as web servers, email servers, and network infrastructure). Some methods for identifying human users and computer systems are similar, others are very different.

The identities of users and systems are verified using authentication mechanisms. In many voting applications, it is very important to identify the user or system you are interacting with in order to not disclose information to, or receive forged information from, the wrong entities.

A system may need to authenticate a user before granting them access to sensitive information or network services. For example, a voting jurisdiction probably wants to authenticate a person before allowing them to update their

ballot delivery information; some jurisdictions may even require authentication before delivering blank ballots. Another typical use is authenticating users before allowing administration of networking systems and equipment.

A user may need to authenticate a system before the user is willing to divulge personal information that can be used to impersonate the user later. For example, a user would want to be sure they are talking to a trusted server before the user gives his or her password or data used for password recovery such as their mother's maiden name. System-to-system communication often also relies on both systems being able to authenticate the others' identity.

Section 3 covered identification and authentication of users for both local access to machines and access to network resources. As described there, low-impact systems frequently use passwords over an encrypted channel to authenticate users. Medium-impact systems often require multi-factor to authenticate users. For high-impact systems, cryptographic hardware devices such as smart cards can be used to authenticate users.

Authenticating machines normally involves stronger forms of authentication mechanisms. Instead of passwords or multi-factor authentication, machine authentication is almost always done with cryptographic authentication mechanisms using strong keys stored on the system. A strong key is one that contains so much unpredictable material an attacker could not possibly guess the key even if he or she used phenomenally expensive systems for an extremely long time.

Machine authentication comes in two broad categories: those that use asymmetric public keys (such as digital signatures and public key encryption) and those that use shared secrets. Both can be equally secure for authenticating machines, but they are used quite differently in practice.

- Authentication based on public keys requires the verifier either have a copy of the public key or, more often, trust a third party that assures the public key given by the claimant is in fact theirs. The latter is how essentially all web browsers using TLS allow users to authenticate the servers to which they connect.
- Authentication based on shared secrets requires the two parties to have already exchanged the key they will use for communication. This exchange takes place out-of-band, meaning it uses a different protocol than the one being protected.

If an attacker can get a copy of a machine's authentication key, that attacker can impersonate the machine. In most current deployments, keys are stored on hosts on normal storage media such as hard drives. This relies on the security of the system to be as strong for protecting the keys as it is for protecting other system-critical information and processes. For example, most keys can only be read and written by someone who has the highest authorization access on a computer. Some high-impact systems store their keys in hardware using hardware security modules (HSMs). HSMs have much better properties to protect

the keys from being read by an attacker, but rely on operational changes that are too onerous for many organizations.

5.5 Certificates, Keys, and Trust Anchors

Network devices such as web servers, mail servers, and firewalls, are normally identified to other devices using the cryptographic methods described above. These methods most commonly use digital certificates for identification. In a small number of voting systems, most notably with secure email, people are identified with certificates. In short, a digital certificate is a signed assertion that the cryptographic key in the certificate is associated with a particular person or system. Users of certificates rely on trusted third parties (often called "certificate authorities" or "CAs") to make those assertions.

In order for identification using certificates to be trustworthy, the secrets that are associated with the keys in certificates must be kept private; otherwise, an attacker who knows the secret could impersonate the holder of the keys. This requirement puts a lot of pressure on individuals to do proper key management. The three parts of NIST SP 800-57, *Recommendation for Key Management*, describe the issues with maintaining the secrecy of keys and the use of certificates for identification. In specific, Section 2 of Part 3 of this series lists many best practices for using keys in certificates.

In order for systems such as web browsers to work with certificates, they must have a set of *trust anchors* that are trusted to associate cryptographic keys with devices and people. A trust anchor is the key for a certificate authority who issues certificates (or authorizes others to do so on its behalf). The set of trust anchors used by an application or operating system is called the trust anchor *store*. Trust anchor stores must be managed carefully because if an attacker can get its own key in the trust anchor store of an application, or if he can subvert the trust anchor that is already in an application's trust anchor store, the attacker can impersonate systems with whom the application communicates.

There are many different ways a CA might create a certificate for a web server or email user (the process is called *enrollment* although that term is rarely used on CA web sites). Because of this, when asking a CA to create a certificate for you, you need to first find their enrollment instructions and be sure they work for the web server or email client for which you want a certificate. Most often, the process involves telling your software to create a *certificate request*, delivering that certificate request to the CA, receiving email from the CA to validate you are authorized to request a certificate, performing that validation, and then receiving the certificate itself.

If your intended users do not already trust the CA with whom you have enrolled, those users must add a trust anchor for that CA in their web browser, email client, or operating system. Again, the steps to do this vary widely between different types of software. Also, note some users will be very hesitant to add a trust anchor because most software (for good reason) gives dire warning about adding trust anchors. In most cases, jurisdictions will be able to obtain certificates from a trust anchor supported by default in common browsers.

Jurisdictions that want to be able to validate signed email from military personnel need to install the trust anchor for the US Department of Defense Root CA. This certificate can currently be found at <http://dodpki.c3pki.chamb.disa.mil/rootca.html>.

5.6 Other Network Protection

Networks that have many individual users often want to limit who has access to the network, or at least limit access to certain parts of the network. This type of fine-grained admission to a network requires network access control, sometimes abbreviated NAC. NIST SP 800-46rev1, *Guide to Enterprise Telework and Remote Access Security*, particularly section 3 of that document, describes network access control systems and how they can be placed in a network to grant the specific access to users that a network administrator would want.

Security systems such as firewalls, VPNs, and network access control do not always succeed in the goal of keeping unwanted traffic from a network. Because of this, some network administrators deploy intrusion detection systems (IDSs) and intrusion prevention systems (IDPs) in parallel with firewalls to look for many different types of unwanted traffic. As explained in NIST SP 800-94, *Guide to Intrusion Detection and Prevention Systems (IDPS)*, these devices are useful in profiling the type of traffic coming onto a network and looking for common attacks. However, managing IDSs and particularly IDPs can be very labor-intensive because most networks have complicated and hard-to-predict traffic patterns, in that these devices need to produce a lot of logs in order to be useful.

As described earlier in this section, network segmentation can make management of each segment easier. Firewalls and VPN devices can be used to segment networks at their edges. In a LAN, however, network segmentation can be achieved more easily with managed switches. Low-end, unmanaged switches create fast connections based on traffic patterns, but managed switches also allow configuration to restrict access to certain ports (and therefore the networks connected to the ports) based on policies. Many managed switches allow access to a particular port based on authentication protocols using passwords and certificates. Managed switches cost much less than firewalls or VPNs, and they require much less setup and operational overhead to run.

6 Ongoing Voting System Protection

Maintaining the security of electronic voting systems takes more than just planning and one-time execution of preventative steps: security must be monitored and acted on throughout the life of the system. Sections 4 and 5 of this document give advice about how to plan for protecting hosts and networks in a voting system, and discuss some aspects of how to maintain security day-to-day. However, IT threats faced by election system usually evolve, so paying attention to security every day can be just as important as planning and proper initial setup.

6.1 System Audits and Record Keeping

The core practice for ongoing security is auditing of IT systems. Observing the statuses of the various parts of a system allows an administrator to find where the system is vulnerable to threats or, if not found ahead of time, the part of the system that was vulnerable to a successful attack. By their very nature, voting systems are subject to audits and record keeping to detect voter fraud. This section covers system audits and record keeping specific to host and network security that can be quite different than the type of attacks seen on non-electronic voting.

Some voting audits may also require IT system audits as one part of the overall audit. These voting audits will probably specify what type of auditing is needed for hosts and networks, but a jurisdiction should strongly consider going beyond the minimum required by voting audits for their IT auditing practices. Collecting more information can help detect attempted attacks that might be missed by collecting only the minimum amount of information required.

It is relatively rare to see an attack in progress and recognize it as an attack, so keeping records of all audits is necessary. Good record keeping is useful for finding when and where an attack happened, but also for finding patterns of unsuccessful attacks as part of ongoing assessments about how to improve the security of a system. The value of the latter should not be underestimated: auditing stored audits can be very valuable to preventing attacks that take research on the part of an attacker.

Monitoring events can happen either in an automatic, continuous fashion, or sporadically by people who look through event logs and so on. Continuous monitoring is far more reliable for capturing data that can be used to analyze or prevent attacks, but sporadic monitoring by humans is required to detect anomalous events missed by automated programs. It is impossible to say either how detailed continuous monitoring records should be or how often sporadic human monitoring should take place: such judgments depend on the nature of the voting system and the value of various attacks to the attackers. Section 3 of NIST Draft SP 800-137, *Information Security Continuous Monitoring for Federal Information Systems and Organizations*, describes the process needed for both automatic and sporadic monitoring of networked computer systems.

6.1.1 Host Audits

The servers and personal computers that run in a voting network are as subject to attack as any system on the Internet. Thus, it is important to audit as much information in these systems as you would any other network-connected system. The types of information typically monitored in hosts include:

- User logins
- Running of administrative software
- Addition and removal of applications
- Patching of applications and the operating system

Hosts on voting systems may have additional IT-related auditing requirements, such as monitoring changes to voter databases or logging the number of requests for particular types of ballots.

6.1.2 Network System Audits

Networks themselves are rarely audited. That is, it is rare to try to perform a complete capture and audit of all information flowing over a network connection. Instead, the systems that make up the network have their software and event logs audited. These systems include all types of routers and firewalls, and some jurisdictions will even monitor local switches. The types of information typically monitored in hosts include:

- User logins to management software
- Event logs from firewalls and intrusion detection systems
- All configuration changes
- Use of encryption for connections
- Patching of system software
- Changes to hardware subsystems

Users of networks sometimes add unauthorized systems to the network. A common example is users who sometimes add wireless access points in order to improve local connectivity. In fact, these can unintentionally allow outsiders to access the network in ways unanticipated by the network administrator. Another common example is computers that have not been vetted by the IT department being added temporarily, but still causing havoc.

To prevent such unintended additions, many network administrators will perform system scans to see all the computers and network devices on the network. They then compare the results of the scan with a known inventory of allowed systems. The presence of such systems should be logged before removing the system from the network (or allowing the system if, in fact, it conforms with the network security policy).

6.1.3 Log Security

Network security audits themselves need to be secured because they contain information that can be used to attack a network. For example, a typical audit will tell an attacker what the system administrators did not see when being probed for vulnerabilities. As described in NIST Draft SP 800-137, auditing information is normally kept offline or on systems that have different access control mechanisms than the systems that are being monitored.

The audit logs for electronic voting systems can have even more stringent requirements than those of normal networked systems if they may contain information about voters that can be used to surmise those voters' votes. Although it is unlikely these hosts contain actual votes, there are types of information that represent voting patterns that may be considered sensitive. If a log does not contain any personally-identifying information about voters or votes cast, the security of the log should be as high as for the logs of normal in-person voting. However, if the logged data (even if it is summarized data) contains more than what is available for in-person voting, the logs should probably be as secure as the data itself.

6.1.4 Local Policy Audits

Many jurisdictions have their own security policies. These policies sometimes apply only to the voting aspect of a jurisdiction, but are often inherited from the larger government agency of which the jurisdiction is just one part. Audits of the security practices of a voting jurisdiction may therefore involve separate compliance reviews for separate security policies. These audits can usually be done concurrently because the policies will often have large (usually intentional) overlaps.

6.2 Qualifications and Training

It is important the jurisdictions designing custom Internet-connected voting systems use current best practices in security. This is also true for jurisdictions selecting such systems from vendors: it is not sufficient to believe that all vendors are using security best practices that apply to each jurisdiction. Security practices are implemented by jurisdiction staff and contractors, so having all of those people be able to determine which practices are best is the first step to their implementation.

Different positions have different roles and responsibilities for security. For example, a database administrator has different security objectives than someone who maintains the operating system for web servers used by the jurisdiction. It is thus important that all the security roles and responsibilities for every position are clearly defined and documented.

Once the security responsibilities are laid out, the jurisdiction must ensure that each employee or contractor is qualified for the position(s) they have. This involves determining if each person has the necessary skills and experience to conduct the specific jobs(s) they perform. Note that in typical jurisdiction, a single person will have multiple security-related roles.

When evaluating how well the technical qualifications that a person has are matched for the security skills needed for a particular role, many factors need to be taken into account. They may already have relevant, related experience in security-sensitive tasks such as operating and/or designing systems with security components; they may have certifications in security technologies; and they may have recent education or training without having the opportunity to use it.

A jurisdiction can actively help raise the level of security skills through training programs for all staff. Such security awareness and training programs can help everyone know the jurisdiction's policies and procedures. Section 3 of NIST SP800-50, *Building an Information Technology Security Awareness and Training Program,* describes how to design such a program, and the rest of the document covers important topics such as how to evaluate training programs after they have been implemented.

In addition, a jurisdiction can create or purchase targeted ongoing training for people in specific security-sensitive roles, as described in Section 2.3 of SP800-50. This can help assure that technical staff are proficient in the technologies with which they work. Training can also help election officials and their management understand the risk inherent in the decisions they make.

Some of the more intensive training programs can lead to certification for the trainee. There are a variety of certifications for security personnel from various independent organizations, and each certification has its own level of value and appropriateness for particular tasks. Some of the many types of security certifications include proof of skills such as:

- designing and selecting online systems in which security is an important factor
- day-to-day IT operations of systems with security components
- security management for executives such as Chief Security Officer (CSO)
- managing the security aspects of networking systems for specific hardware and software vendors
- writing software that has security aspects, particularly cryptography

The first two of these are the most valuable in planning for and deploying voting systems connected to the Internet, although it is difficult to directly map the claims of a certification system on many of the tasks that jurisdictions assign to staff and contractors.

6.3 Incident Response Planning

Monitoring electronic voting systems is important for determining when something important has happened, but monitoring must be followed up with incident response. Note, incident response entails responding to known attacks and, just as significantly, responding to events that are even slightly suspicious.

The latter category is often overlooked because it causes a large number of "false positive" reports, but it is a critical part of attack prevention.

Section 2 of NIST SP 800-61rev1, *Computer Security Incident Handling Guide*, details the kinds of incidents for which responses are needed. It emphasizes the need for response planning, including setting up response teams and publishing the response plans so that everyone involved knows their responsibilities. Although some of the recommendations at the end of the section are specific to US government agencies, most of them apply just as well to any organization that needs to deal with computer and/or network incidents.

6.4 Media Control

Many different types of data stored on a computer or network device can be of value to an attacker. Although it is much more common for attackers to try to access valuable data over the Internet, having direct unfettered access to the media on which the data is stored is of huge value to an attacker. Thus, it is critical the media on which the data is stored are not directly available to any attacker, even after these media have been taken out of use.

Similarly, all media used for backups must be stored with at least the same level of safety as is used for the live data. Safely storing backups is different than protecting media that are actively being used because actively-used media are in systems that themselves are usually physically protected. Backup media, on the other hand, are normally kept in unattended locations where many types of media are stored together. Anyone with access to the storage location may be easily able to access particular backup media. Given this problem, normal monitoring of backup media usually involves a plan for destroying old backups that are no longer used.

Controlling election media is also critical for preventing the injection of malware that can then be propagated to users of a jurisdiction's online systems. It is very common for miscreants to use generally-trusted sites that are not adequately protected as launching points for hidden distribution of malware. To prevent being the source of such attacks, jurisdictions need to have close physical control and chain-of-custody tracking for all their electronic media and Internet servers.

6.5 Cryptographic Validation of Hosts and Network Equipment

Nearly all voting systems use cryptography for some of their security features. It is important the cryptographic functions in such systems conform to widely-accepted standards and are implemented using industry best practices. Such conformance assures systems are using algorithms that have been vetted by experts throughout the security field; this, in turn, reduces the likelihood of security breaches due to poorly-chosen cryptographic functions.

NIST's FIPS 140 series of requirements and certifications is probably the best-known set of conformance and best-practice standards available. FIPS 140 is the anchor of a program at NIST called the Cryptographic Module Validation Program (CMVP). US government agencies purchasing equipment that uses cryptography are required to verify the cryptography is certified to conform to FIPS 140, and

other industries have also made FIPS 140 certifications into requirements as well. More information on CMVP and the FIPS 140 program can be found at <http://csrc.nist.gov/groups/STM/cmvp>.

Compliance with cryptographic standards is often considered a one-time check at the time of purchase or deployment, but it really should be part of ongoing audits. A vendor's systems can lose its certification, such as if there is a software or hardware upgrade that breaks compliance. Also, compliance specifications themselves can evolve, and a system that complied with an older version of a specification may not comply with requirements specified in the newer version. Thus, checking for certification should be done periodically as part of normal security auditing practices.

7 References

NIST SP 800-41rev1, *Guidelines on Firewalls and Firewall Policy*

NIST SP 800-42v2, *Guidelines on Securing Public Web Servers*

NIST SP 800-45v2, *Guidelines on Electronic Mail Security*

NIST SP 800-46rev1, *Guide to Enterprise Telework and Remote Access Security*

NIST SP 800-49, *Federal S/MIME V3 Client Profile*

NIST SP 800-57 Part 1, *Recommendation for Key Management – General*

NIST SP 800-57 Part 3, *Application-Specific Key Management Guidance*

NIST SP 800-61rev1, *Computer Security Incident Handling Guide*

NIST SP 800-63 Rev. 1, *Draft Electronic Authentication Guideline*

NIST SP 800-64rev2, *Security Considerations in the System Development Life Cycle*

NIST SP 800-70rev1, *National Checklist Program for IT Products – Guidelines for Checklist Users and Developers*

NIST SP 800-77, *Guide to IPsec VPNs*

NIST SP 800-94, *Guide to Intrusion Detection and Prevention Systems (IDPS)*

NIST SP 800-95, *Guide to Secure Web Services*

NIST SP 800-113, *Guide to SSL VPNs*

NIST SP 800-118, *Guide to Enterprise Password Management*

NIST SP 800-128, *Guide for Security Configuration Management of Information Systems*

NIST SP 800-131, *Recommendation for the Transitioning of Cryptographic Algorithms and Key Lengths*

NIST Draft SP 800-137, *Information Security Continuous Monitoring for Federal Information Systems and Organizations*

NISTIR 7551, *A Threat Analysis on UOCAVA Voting Systems*

Checklists and benchmarks for creating baseline configurations: <http://checklists.nist.gov/>

FIPS 140 and NIST's Cryptographic Module Validation Program (CMVP): <http://csrc.nist.gov/groups/STM/cmvp/>

8 Glossary

Authentication - The process of establishing confidence in the claimed identity of a user or system

Claimant - The person who is asserting his or her identity

Enrollment - The process that a Certificate Authority (CA) uses to create a certificate for a web server or email user

Issuance - Agreeing on the type of token that will be used for future authentication

Management control - Restricting who can manage the computer to a limited number of known people

Management stations – Systems with which only IT and network administrators interact

SQL injection - Attacks that look for web sites that pass insufficiently-processed user input to database back-ends

Tokens - Either information that is only known to the person and the verifier, or a hardware device that can generate information that the verifier knows can only come from that device

Trust anchor - The key for a certificate authority who issues certificates or authorizes others to do so on its behalf

Verifier - The party trying to assess the authenticity of an identity