Open Ended Vulnerability Testing for Software Independent Voting Systems

Prepared at the direction of the STS Subcommittee of the TGDC

May 16, 2007
Version 1.3

The Technical Guidelines Development Committee is an advisory group to the Election Assistance Commission (EAC), which produces Voluntary Voting System Guidelines (VVSG). Both the TGDC and EAC were established by the Help America Vote Act of 2002. NIST serves as a technical advisor to the TGDC.

TABLE OF CONTENTS

1 Introduction

1.1 Background

1.1.1 U.S. Election Assistance Commission (EAC)

The U.S. Election Assistance Commission (EAC) was established by the Help America Vote Act of 2002 (HAVA). The Commission serves as a national clearinghouse and resource for information and review of procedures with respect to the administration of Federal elections. HAVA was enacted to establish a program to provide funds to States to replace punch card voting systems, to establish the Election Assistance Commission to assist in the administration of Federal elections and to otherwise provide assistance with the administration of certain Federal election laws and programs, to establish minimum election administration standards for States and units of local government with responsibility for the administration of Federal elections.

1.1.2 Technical Guidelines Development Committee (TGDC)

HAVA also established a 15-member Technical Guidelines Development Committee (TGDC) to assist the Executive Director of the Election Assistance Commission (EAC) in the development of Voluntary Voting System Guidelines (VVSG). HAVA named the Director of the National Institute of Standards and Technology (NIST) to chair the TGDC. In addition, HAVA requires NIST to provide the TGDC with technical support necessary to carry out its duties.

The duties of the TGDC include the gathering and analysis of data and information related to the security of computers, human factors, voter privacy, and methods to detect and prevent fraud.

In Resolution 17-05, the TGDC requested that NIST perform research and draft standards documents requiring testing of voting systems that includes a significant amount of open-ended research for vulnerabilities by an analysis team supplied with complete source code and system documentation and operational voting system hardware. This responsibility now has been taken up by the Security and Transparency Subcommittee (STS).

The TGDC recently passed a resolution for requiring voting machines to provide the software independent verifiability of cast ballots. For a definition of “software independence” see [SI].

1.1.3 Security and Transparency Subcommittee (STS)

The EAC has also approved formation of subcommittees, including the STS. The purpose of the STS subcommittee is to deal with relevant issues including security, transparency, human factors, privacy, core standards requirements, and testing of voting systems.

1.2 Definition of OEVT

Vulnerability testing is an attempt to bypass or break the security of a system or a device. Like functional testing, vulnerability testing can falsify a general assertion (namely, that the system or device is secure) but it cannot verify the security (show that the system or device is secure in all cases). Vulnerability testing is also referred to as penetration testing. Vulnerability testing can be performed using a test suite or it can be open-ended. Open ended vulnerability testing involves the testing of a system or device using the experience and expertise of the tester; using the knowledge of system or device design and implementation; using the publicly available knowledge base of vulnerabilities in the system or device; using the publicly available knowledge base of vulnerabilities in similar system or device; using the publicly available knowledge base of vulnerabilities in similar and related technologies; and using the publicly available knowledge base of vulnerabilities generally found in hardware and software (e.g., buffer overflow, memory leaks, etc.)

1.3 Purpose of this Document

The purpose of this document is to define the scope of and approach to Open Ended Vulnerability Testing (OEVT) prescribed in VVSG "2007" in light of the software independent verification of cast ballots. Having software independent verifiability reduces the reliance on accuracy and security of the voting machine. For example, if a paper ballot that has potentially been examined for accuracy by the voter and is used in vote count audit, then the accuracy and security of the voting machine is less important. The motivation for this paper is to identify areas where OEVT resources should be spent to mitigate known and anticipated threats, and to reduce the threat of undetected compromise of election outcome.

1.4 Goal of OEVT

The goal of OEVT is to discover architecture, design and implementation flaws that have crept into the system which may not be detected using systematic functional, reliability, and security testing and can be exploited to change the outcome of an election, can provide erroneous results for an election, can cause denial of service, can compromise secrecy of vote, or can compromise of security audit log. The OEVT team may need to prioritize its search due to the schedule and resource limitations. The OEVT team should focus on flaws that can be exploited for vote-tampering or provide erroneous election outcomes as the most important. The next priority should be denial of service attacks during the election, followed by attacks that reveal how some of the voters voted.

The goal of OEVT also includes attempts to discover logic bombs, time bombs or other Trojan Horses that may have been introduced in the system hardware, firmware or software in order to change the outcome of an election or in order to provide erroneous results for an election. These types of problems can be reduced by using development security controls consisting of physical security, personnel security, procedural security, and technical security for the development environment. These types of problems may be detected by examining the code thoroughly and by conducting code correspondence.

These types of problems may also be detected by code analysis tools designed to detect suspicious and malicious code segments. Thus, OEVT includes examination of code, and analysis of code using automated tools to assess the possibility of logic bombs, time bombs or other Trojan Horses in the voting system software.

Another goal of OEVT is to identify the nature of collusion that may be required to compromise the security of the voting system.

The TGDC request to NIST stated that the open-ended vulnerability search and research should include those involving adversaries with significant financial and technical resources. This coupled with the fact that most exploitation and associated automation scripts become public, implies that the OEVT needs to be rigorous. In other words, difficulty of the exploit alone should not be used to rule out a vulnerability.

1.5 Scope of this Document

Scope of this document is limited to OEVT for Software Independence. Other forms of testing will be addressed in other document(s). For example, the functional testing, reliability testing, security functional testing, parallel testing, random testing, pre-election testing, etc. are outside the scope of this document.

While the OEVT requirements in this paper apply to all voting machines, prioritization of OEVT resources are suggested based on the "Software Independence".

Scope of OEVT includes the entire voting system, including the documented use procedures.

1.6 Assumptions

It is assumed that for the OEVT, the voting machines will be configured similar to how they are to be used in election.

1.7 Document Organization

Section 2 describes the scope of OEVT. Section 3 describes the OEVT activities. Appendix A contains the vulnerabilities that have been surmised or demonstrated on existing voting systems. Appendix B contains additional flaw hypotheses that have not been necessarily tested.

2 OEVT Scope and Priorities

2.1 OEVT Scope

The TGDC request to NIST stated that the open-ended vulnerability search and research should not exclude those involving collusion between multiple parties (including vendor insiders). This is interpreted to mean that the scope of OEVT should include pretty much all aspects, including but not limited to the following:

- Voting System Security
- Voting System Physical security while voting machines are:
  - In storage
  - Being configured
  - Being transported, and
  - Being used
- Voting System Use Procedures

2.2 OEVT Priorities

The OEVT team should make its own prioritization of the exploitation scenarios based on the availability of time and resources and the OEVT team's determination of which exploitation scenarios seem to offer the highest payoff.

This section contains some suggestions for the OEVT team for prioritization. The suggestions are made in the following areas: threat scenarios; COTS investigation; Internet based threats; and cryptographic investigation.

2.2.1 Threat Priorities

The OEVT team should prioritize its activities based on the impact of the threats to the voting process. The following priorities are suggested:

- Threats that can be exploited to change the outcome of an election and flaws that can provide erroneous results for an election should have the highest priority;
- Threats can cause denial of service during the election should be considered of very high priority;
- Threat that can compromise secrecy of vote should be considered of high priority;
- Threat to disclosure or modification metadata (e.g., security audit log) that does not change the outcome of the election, cause denial of service during the election, or does not compromise the secrecy of ballot should be considered of lower priority.

2.2.2 COTS

If the voting machine qualifies for SI[1] and the voting machine security depends on the security features or correctness of the COTS, the OEVT team should rely on publicly known vulnerabilities and other suspected vulnerabilities (e.g., NIST CVE database) for OEVT on COTS[2]. Additional comprehensive search for flaws in COTS should be undertaken if resources are available after these vulnerabilities and voting application vulnerabilities are examined. In other words, in order to prioritize scarce OEVT resources, existing research should be relied on for COTS.

2.2.3 Internet Based Attacks

The OEVT team should not consider the voting machine vulnerabilities that require Internet connectivity for exploitation if the voting machine is not connected to the Internet during the election and otherwise. However, if the voting machine is connected to another machine which in turn may have been connected to the Internet, Internet based attacks may be plausible.

2.2.4 Cryptography

If cryptography is used in the voting system, the OEVT team should prioritize the cryptographic algorithms, protocols, and implementation analysis. The following is an illustrative list of cryptographic analysis related items. While reasonably comprehensive, the list is not necessarily exhaustive:

- The following activities should have the highest priority:
    - Cryptographic algorithm, key size, and mode of operation that are not FIPS approved (per applicable FIPS) or not FIPS recognized (per NIST Special Publications).
    - Cryptographic protocol analysis if the cryptographic protocol is neither a FIPS approved nor IETF approved standards. The analyst shall be an expert in cryptographic protocols and shall independently determine which of the security services (e.g., confidentiality, integrity, source authentication, non-repudiation, replay protection, etc.) are required. The expert shall then analyze the cryptographic protocol to ensure that the desires security services are provided by the protocol and there are no flaws in the protocol with respect to these required services.
    - Analysis of options for cryptographic protocols that are FIPS approved or IETF approved standards..
    - Cryptographic protocol implementation.
    - Various side channel attacks such as power analysis, error injection, and timing analysis.
    - Key management analysis to ensure that the secret and private keys are protected from disclosure and from unauthorized modifications. The scope shall include key management analysis to ensure that the public keys are protected from unauthorized modifications.

[1] For the definition of Software Independence, see [SI].
[2] For the definition of COTS, see [COTS}.

    - Key management implementation.

- The following activities should be low priority since chances of discovering flaws in these are low:
    - Cryptographic algorithm, key size, and mode of operation analyses that are FIPS approved or FIPS recognized.
    - Cryptographic protocol analysis for protocols that are FIPS approved or IETF approved standards.
    - Cryptographic algorithm implementation that have been validated under NIST Cryptographic Algorithm Validation Program (CAVP).

3 OEVT Approach

The OEVT team should develop a penetration analysis and testing approach. This section contains some suggestion in this area.

The OEVT is should consider the following phased approach:

- Perform OEVT on voting system components such as DRE, Audit Machine, Central Tabulator, etc.

- Perform OEVT on the integrated voting computer system that includes all the system components

- Perform OEVT on the integrated voting computer system, including the voting system use procedures

The OEVT team is provided all the documentation and a voting system and becomes fully familiar with the voting system, including the procedures used to install, configure and operate the system. The documentation includes the system architecture, system design, system test plans and procedures, and testing laboratory test plans, test procedures and test results.

The following sections describe the OEVT activities associated with each of these phases.

3.1 OEVT for Individual Voting System Component

The OEVT team should develop a penetration analysis and testing approach for voting system components. The OEVT team should consider the following activities in developing their approach:

1. The OEVT team becomes familiar with the component architecture and design. The OEVT team performs a security analysis of the component. In addition to performing the security analysis of the overall components, the following should be analyzed, if applicable:
    a) Analyze the security of all input interfaces to the component. This analysis should consider malicious input scenarios such as malformed data, invalid values for the various fields, and overflow and underflow of input.

2. The OEVT team performs code review and code correspondence. The OEVT team also uses automated tools to analyze the code for buffer overflow, memory leaks, dead code, and otherwise suspicious code. The results of this step are used to identify issues and flaw hypotheses.

3. The OEVT team reviews the developer and testing laboratory test to gain insight into the degree of rigor applied. The purpose of the reviews is to examine the

comprehensiveness of testing and use that to identify areas that are more likely to have flaws. The interfaces that are not tested at all or tested poorly, may be good avenues of investigation. Thus, some of the examples this examination are:

a) How thoroughly each external interface is tested. Testing of each interface should have included:
    i. Trying nominal values, boundary values, and erroneous values for all the inputs.
    ii. Create all possible error conditions/codes for the interface.
b) How thoroughly interfaces among modules are tested.
c) How much of the code is covered by developer and independent laboratory testing.
d) Which known or potential vulnerabilities the component was tested for. For example, did the testing include applicable scenarios from the databases such as CVE and the voting forums?

4. The OEVT team becomes familiar with previous security analysis and penetration testing conducted for the component and for the older versions of the component.

5. The OEVT team uses its knowledge of the system internals and analysis from the steps above, and its own penetration testing expertise to brainstorm possible ways to break into the system. During the brainstorming phase, no evaluation of the hypotheses is carried out. The goal is to develop hypotheses to break the system.

6. The hypotheses are recorded.

7. The OEVT team removes the hypotheses whose associated vulnerabilities have been disproved by the developer testing, by independent laboratory testing, or by previous security analysis and penetration testing.

8. The OEVT evaluates the flaws hypothesized and flaws proven by previous security analysis and penetration testing activities for the component. The OEVT team adds the plausible flaws to the hypotheses.

9. The OEVT team identifies any known vulnerability tests that have not been conducted by performing a search for the known vulnerabilities in the database. The OEVT team adds these vulnerabilities to the list of hypotheses developed in Step 6 above. These vulnerabilities are marked as high pay off since they are derived from known vulnerabilities.

10. The OEVT team makes a broad search for vulnerabilities and flaw hypotheses and their applicability to the voting system. The vulnerabilities and hypotheses would come from the public database.
    a) The OEVT team considers vulnerabilities and hypotheses in similar systems.
    b) The OEVT team analyzes vulnerabilities and hypotheses in other systems to determine their applicability to the voting system being tested.

c) The OEVT team considers the hypotheses

11. The OEVT team adds these to the list of hypotheses.

12. The OEVT team uses its knowledge of the system to identify inputs and internal probes that will induce errors that are either externally visible or internally handled by the system but were not exercised by the developer testing or by independent laboratory testing. The OEVT team adds these to the flaw hypotheses.

13. The OEVT team also identifies inputs and internal probes that will invoke code segments that were not be exercised by the developer testing or by independent laboratory testing. The OEVT team adds these to the flaw hypotheses.

14. The OEVT team evaluates the cumulative hypotheses and rejects the ones that are not plausible.

15. The OEVT team prioritizes the remaining hypotheses based on payoff potential, damage the flaw can cause if present, and effort involved to test them.

16. The OEVT team takes the top hypotheses and develops and executes the penetration scenarios. The OEVT team need not execute the entire penetration attack if code examination, other tests (penetration or otherwise), and analysis demonstrate that a given attack is likely to succeed.

17. Based on the results of the penetration scenarios, new hypotheses are developed and step 16 is repeated.

18. The OEVT team performs penetration testing of the COTS used in the component to determine if the COTS security can be bypassed or otherwise privilege (e.g., administrative) can be gained to negate the COTS configuration assumptions on which the voting application security is based.

The OEVT team can forego performing OEVT on a component that can not impact the outcome of an election. This determination must be based on an independent security analysis by the OEVT team.

The OEVT team can reduce the degree of rigor in performing OEVT on a component based on the number and severity of the hypothesized vulnerabilities.

3.2 OEVT for Integrated Voting System

The same OEVT team that performs the OEVT on the computer system components should perform OEVT on the overall voting computer system. The OEVT team should develop a penetration analysis and testing approach for voting system computer system. The OEVT team should consider the following activities in developing their approach:

1. The OEVT team uses component OEVT to brainstorm possible ways to break into the overall system. During the brainstorming phase, no evaluation of the hypotheses is carried out. The goal is to develop hypotheses to break the system. The focus is on how the interfaces among components can be manipulated using a computer system component or communication channel to defeat the security of the voting system.

2. The OEVT team adds the known vulnerabilities and publicly available hypotheses for the overall computer system.

3. The OEVT team evaluates the hypotheses and rejects the ones that are not plausible or have been covered by the developer testing, independent laboratory testing, or by component OEVT.

4. The OEVT team prioritizes the remaining hypotheses based on payoff potential, damage the flaw can cause if present, and effort involved to test them. In determining the potential damage, the OEVT team analyzes if flaw or compromise in one component can be used to compromise other components. For example, if a virus in one component can infect other components over time, or if privilege escalation in one component can result in administrative access to other components.

5. The OEVT team takes the top hypotheses and develops and executes the penetration scenarios.

6. Based on the results of the penetration scenarios, the OEVT team develops new hypotheses and repeats steps 4 and 5.

3.3 OEVT for Fielded Voting System

The OEVT for the fielded system includes testing to determine if the integrated system and the physical and procedural security can be defeated to impact the outcome of an election.

1. The OEVT team uses overall computer system OEVT to brainstorm possible ways to break into the overall system by compromising physical, procedural, and personnel security controls. During the brainstorming phase, no evaluation of the hypotheses is carried out. The goal is to develop hypotheses to break the system.

2. The OEVT team evaluates the hypotheses and rejects the ones that are not plausible or have been covered by the developer testing, independent laboratory testing, by component OEVT, or by the integrated voting system OEVT.

3. The OEVT team prioritizes the remaining hypotheses based on payoff potential, damage the flaw can cause if present, and effort involved to test them. In

determining the potential damage, the OEVT team analyzes how the use procedures and recommended physical security protect against insider threat. For example, what harm can a person with privilege access (e.g., administrative access) to one component (e.g. Central Election Management System) can cause to the election, what harm a single individual can cause if in control of a removable storage device that contains election software and/or data.

4. The OEVT team takes the top hypotheses and develops and executes the penetration scenarios.

5. Based on the results of the penetration scenarios, the OEVT team develops new hypotheses and repeats steps 3 and 4.

3.4 OEVT Resources

3.4.1 Information and System

The OEVT team is provided the entire Technical Data Package (TDP), including but not limited to the following:

- Threat analysis describing threats mitigated by the voting system
- Security architecture describe the philosophy or protection (i.e., security model) and how threats to the voting system are mitigated
- High level design of the system
- Any other documentation provided to the testing laboratory
- Source code
- Operational voting system configured for election, but with the ability for the OEVT team to reconfigure it
- Testing reports from the developer and from the testing laboratory
- Physical and procedural security procedures

3.4.2 Team Composition

The OEVT team should consist of 3-5 software engineering, information security, and penetration testing experts. The team should consist of the following:

- 1-2 Penetration Testing Experts
- 1-2 Information Security Experts
- 1 Software Engineer

Each member of the OEVT team should have the following qualifications:

- Good knowledge of work done to date on voting system design, research and analysis conducted on voting system security, and known and suspected flaws in voting systems;
- Good knowledge of the information presented at the October 2005 NIST Workshop on threat to voting systems;

- Bachelor's degree in computer science or related field from a reputed school with Grade Point Average of B+ or better. Examples of related fields are Mathematics; Software Engineering; Information Security; Electrical Engineering, and
- Seven years of experience in design, implementation, security analysis, or testing of technologies or products involved in the voting systems

In addition to the above requirements,

- Four of the seven years of experience of each penetration testing expert should be in penetration testing of information systems;
- Four of the seven years of experience of each information security expert should be in security analysis and testing of information systems;
- Four of the seven years of experience of each software engineer should be in design and development of information system software and firmware;

In addition, the OEVT team will require consultation from an elections expert who is familiar with election procedures; how the voting systems and installed and used; and how votes are counted.

The OEVT team will need consultation assistance from a cryptography expert (unless one of the information security experts is also expert in cryptography) if the voting system uses cryptography. The level of effort of the cryptography expert will depend on the degree to which cryptography is used to meet the VVSG 2007 requirements and to ensure vote integrity.

3.4.3 Level of Effort

The OEVT team should spend about 3-6 weeks worth of effort in performing the OEVT. Given the team composition of 3-5 persons, this means 9-30 person weeks of effort. The following factors should be considered in determining the level of effort:

- Size and Complexity of the voting system;
- Comprehensiveness of the testing laboratory analysis and testing activities;
- Number of vulnerabilities found in previous security analysis and testing of the voting system and its prior versions; and
- Flaw hypotheses surmised by the OEVT team. This may require that the OEVT team perform some preliminary investigation prior to determining the time they plan to spend performing OEVT.

3.5 OEVT Results

The OEVT team should record its finding and provide these findings to the EAC and the sponsor of OEVT. Sponsor of OEVT could be EAC, a State Election Board, voting system vendor, or a testing laboratory. The report should contain all the information discovered during the conduct of the OEVT, including but not limited to:

- Names, organizational affiliations, summary qualifications, and resumes of the members of the OEVT;

- Time spent by each individual on the OEVT activities;

- List of hypotheses considered;

- List o hypotheses rejected and rationale;

- List of hypotheses tested, testing approach, and testing outcomes;

- List of hypotheses not tested and their rationale (e.g., too costly, low payoff; minimal damage, etc.);

- List and description of remaining vulnerabilities in the voting system.
    - Description of each vulnerability should include how the vulnerability can be exploited and what is the nature of the impact. Some of the examples of the impact are over-count of ballots for a candidate; undercount for a candidate; very slow response time during election; erasure of votes; lack of availability of the voting machine during election; etc.
    - For each vulnerability, the OEVT team should identify the VVSG 2007 requirement(s) violated
    - The OEVT team should flag those vulnerabilities as serious if the vulnerability can result in:
        - Violation of one or more VVSG 2007 requirements; or
        - Change the outcome of an election;
        - Erroneous results for an election;
        - Denial of service (lack of availability) during the election.

- Recommendations in terms of OEVT time and effort for this system and other voting systems.

The testing laboratory should examine the OEVT results and update their compliance assessment of the voting system based on the OEVT. The testing laboratory should use the following approach in updating their compliance assessment:

- The testing laboratory should examine each remaining vulnerability that is marked serious by the OEVT team. Unless the testing laboratory can provide an explanation as to how the vulnerability can not be exploited, it should form the basis for non-compliance; and

- The testing laboratory should examine each of the remaining vulnerabilities to ensure that it does not result in violation of any of the VVSG 2007 requirements. If a vulnerability can result in violation of one or more VVSG 2007 requirements, the voting system should be judged non-compliant in that area unless the testing

laboratory can provide an explanation as to how the vulnerability can not be exploited.

Appendix A: Known Vulnerabilities

This appendix contains the list vulnerabilities that have been exhibited by the current voting machines and voting systems. It also includes the vulnerabilities that have been surmised, but not fully demonstrated.

Appendix B: Flaw Hypotheses

This appendix provides a list of flaw hypotheses. The hypotheses are different from known vulnerabilities, but known vulnerabilities have played a part in generating them.

These hypotheses are for illustrative purposes only. The OEVT tester should use the methodology described in this document and the actual voting system to develop applicable hypotheses.

References

[COTS] COTS Discussion Paper, October 16, 2006.

[SI] Four Approaches to SI and Accessibility, March 9, 2007

List of Acronyms

CVE	Common Vulnerability and Exposures
EAC	Election Assistance Commission
FIPS	Federal Information Processing Standard
HAVA	Help America vote act
NIST	National Institute of Standards and Technology
OEVT	Open Ended Vulnerability Testing
STS	Security and Transparency Subcommittee
TGDC	Technical Guidelines Development Committee
U.S.	United States
VVSG	Voluntary Voting System Guidelines