Context Description: Posted Dec. 1, 2006

This draft report was prepared by NIST staff at the request of the Technical Guidelines Development Committee (TGDC) to serve as a point of discussion at the Dec. 4-5 meeting of the TGDC. Prepared in conjunction with members of a TGDC subcommittee,
the report is a discussion draft and does not represent a consensus view or recommendation from either NIST or the TGDC. It reflects the conclusions of NIST research staff for purposes of discussion. The TGDC is an advisory group to the Election Assistance Commission, which produces voluntary voting system guidelines and was established by the Help America Vote Act. NIST serves as a technical advisor to the TGDC.

The NIST research and the draft report's conclusions are based on interviews and discussions with election officials, voting system vendors, computer scientists, and other experts in the field, as well as a literature search and the technical expertise of its authors. It is intended to help in developing guidelines for the next generation of electronic voting machine to ensure that these systems are as reliable, accurate, and secure as possible. Issues of certification or decertification of voting systems currently in place are outside the scope of this document and of the TGDC's deliberations.

COTS Discussion Paper

1 TGDC Resolution #14-05

TGDC Resolution #14-05 reads:

> The TGDC has considered the advisability of using Commercial Off-The-Shelf Software ("COTS Software") within voting systems, from a security perspective. It has concluded that, generally speaking, the use of COTS software introduces excessive and unnecessary risk and should be avoided, while specific well-motivated exceptions to this rule may be required upon occasion. The TGDC directs NIST to research and draft standards documents requiring:
>
> 1. That the use of COTS software within voting systems is not allowed unless it meets specific exceptional conditions, and
>
> 2. That the criteria for exceptions shall be drafted by NIST.

This resolution was motivated by concern that COTS software that is not up to voting system standards of security, etc. would be exempted from test lab evaluations and thus wind up in a certified voting system. Previous versions of the Guidelines were somewhat confusing about which requirements and evaluations were applicable to COTS software and hardware.

In the current working draft, the only test lab evaluation from which COTS software can easily be exempted is source code review, and then only if the COTS software is in fact completely unmodified.

For all other evaluations, including security evaluations, if the test lab determines that an evaluation is made redundant by previous certifications or relevant field performance, that determination must be documented and justified in the test plan, which must then be approved by the EAC. Any "rotten COTS" whose field performance is inadequate or irrelevant to its expected use in voting systems would not have a strong case for exemption from test lab evaluations, and we would not expect such an exemption to be proposed by a test lab or approved by the EAC.

The same approach addresses hardware COTS concerns. The exemption from environmental testing that [4] granted to COTS products was on the condition or assumption that applicable external regulations were sufficient to show conformity with the requirements of the Guidelines. But we are now finding that this often is not the case: e.g., as noted in [6], the RF immunity requirements in [4] are more stringent than any generally applied to consumer electronics. Similarly, a COTS office computer could be FCC Class A for electromagnetic emissions, but [4] requires the more stringent Class B. We would not expect a test plan that exempted a hardware product from evaluations for RF immunity or electromagnetic emissions based on these inadequate prior certifications to be proposed by a test lab or approved by the EAC.

We believe that this fully addresses the concern that an unwise exemption from test lab scrutiny would allow a voting system that relies on "rotten COTS" to attain certification. This approach has been discussed in teleconferences of the Security and Transparency Subcommittee and the Core Requirements and Testing Subcommittee and appears to be an acceptable response to the resolution.

2 COTS authentication

If a test plan is approved that accepts previous certifications or field performance of a COTS product in lieu of some test lab evaluation, it remains to be shown that the relevant component of the voting system is in fact identical to the COTS product that it is asserted to be. Consensus of the Security and Transparency Subcommittee and the Core Requirements and Testing Subcommittee is that this is best established by having the procurement and integration of the COTS components be performed by or witnessed by the test lab as part of the initial system build (before certification testing begins). COTS products must be procured from canonical sources to demonstrate their authenticity.

3 Borderline cases

To avoid an effective ban on the use of COTS software in voting systems, COTS software is excused from the requirement to deliver source code to the test lab and the requirement to conform to coding conventions specified by the Guidelines. This pragmatically motivated double standard has caused some borderline cases to emerge. First, Windows CE[1] is alleged to be a COTS product, but to deploy it requires source code customization, as it is effectively being ported to a new platform each time. Second, there are cases where the integration of vendor-developed code with COTS products cannot be accomplished without violating coding conventions specified by the Guidelines because the COTS product determines the interface. Third, there are questions whether Hypertext Markup Language files and other "non-software" files could, through ambiguity in the Guidelines, face scrutiny under the coding conventions. Finally, there are questions regarding the applicability of requirements to code generated by a COTS package. This case was handled in previous Guidelines, but not very clearly.

The current draft defines new terms to clarify exactly which requirements are applicable in each case. In the new terms, "logic" is used in lieu of software, firmware, or what have you because the software/firmware/etc. distinction is not germane to the level of scrutiny that the logic should receive from a test lab. Tabulation code needs to be evaluated by a test lab regardless whether it is stored in nonvolatile memory or on a hard drive. The ability to waive evaluation of embedded COTS firmware such as is found in a keyboard controller is provided by the response to TGDC Resolution #14-05 (i.e., the relevant field performance suffices to show conformity).

The new terms are as follows.

application logic: Software, firmware, or hardwired logic from any source that is specific to the voting system, with the exception of border logic.

border logic: Software, firmware, or hardwired logic that is developed to connect application logic to COTS or third-party logic. Note: Although it is typically developed by the voting system vendor, border logic is constrained by the requirements of the third-party or COTS interface with which it must interact. It is not always possible for border logic to achieve its function while conforming to standard coding conventions. For this reason, border logic should be minimized relative to application logic and where possible, wrapped in a conforming interface. An example of border logic that could not be so wrapped is a customized boot manager that connects a bootable voting application to a COTS BIOS.

configuration data: Non-executable input to software, firmware, or hardwired logic.

core logic: Subset of application logic that is responsible for vote recording and tabulation.

COTS: Software, firmware, device or component that is used in the United States by many different people or organizations for many different applications and that is incorporated into the voting system with no vendor- or application-specific modification. Note: (1) The expansion of COTS as Commercial Off-The-Shelf is no longer helpful, since much of what satisfies the requirements is non-commercial software that is not available in stores. The acronym COTS is used here only because it is familiar to the audience. (2) By requiring "many different applications," this definition deliberately prevents any application logic from receiving a COTS designation.

non-executable: Declarative or informative in nature; not subject to interpretation as a sequence of imperative instructions as in a functional programming language.

third-party logic: Software, firmware, or hardwired logic that is neither application logic nor COTS; e.g., general-purpose software developed by a third party that is either customized (e.g., ported to a new platform, as is Windows CE) or not widely used, or code generated by a COTS package.

The applicability of requirements and test lab evaluations to these newly defined categories is summarized in Table 1.

Categories	Level of scrutiny	Tested ?	Source code required?	Coding standards enforced?	Shown to be correct?
COTS	Black box	Yes	No	No	No
third-party logic, border	Clear box	Yes	Yes	No	No

logic, configuration data					
application logic	Coding standards	Yes	Yes	Yes	No
core logic	Logic verification	Yes	Yes	Yes	Yes

Table 1 Levels of scrutiny

The relevant precedent is detailed in the following subsections.

3.1 Black box testing

[4] sends mixed signals with respect to COTS software: the requirements are there, but the test lab is instructed not to test them.

> The requirements of this section apply generally to all software used in voting systems, including software furnished by an external provider (for example, providers of COTS operating systems and web browsers) where the software may be used in any way during voting system operation. — [4] I.5.1.1.
>
> Unmodified software is not subject to code examination. — [4] I.5.1.1.
>
> Voting system application software, including commercial off-the-shelf (COTS) software, shall be designed in a modular fashion. However, COTS software is not required to be inspected for compliance with this requirement. — [4] I.5.2.3.
>
> Unmodified, general purpose COTS non-voting software (e.g., operating systems, programming language compilers, data base management systems, and Web browsers) is not subject to the detailed examinations specified in this section. — [4] II.5.2.

The hardware requirements also send mixed signals. First there is a statement waiving testing when a prior certification or performance history is sufficient, but then there is a blanket exclusion from the non-operating tests regardless whether there is sufficient evidence.

> Unmodified COTS hardware will not be subject to all tests. Generally such equipment has been designed to rigorous industrial standards and has been in wide use, permitting an evaluation of its performance history. To enable reduced testing of such equipment, vendors shall provide the manufacturer specifications and evidence that the equipment has been tested to the equivalent of these Guidelines. — [4] II.4.2.1.
>
> Systems exclusively designed with system-level COTS hardware whose configuration has not been modified in any manner are not subject to this segment of hardware testing. — [4] II.4.6.1.

3.2 Clear box testing

Source code provided by third parties and embedded in software modules for compilation or interpretation shall be provided in human readable form to the accredited test lab. — [4] I.5.1.1.

Vendors shall submit a record of all user selections made during software installation as part of the Technical Data Package. The vendor shall also submit a record of all configuration changes made to the software following its installation. The accredited test lab shall confirm the propriety and correctness of these user selections and configuration changes. — [4] I.5.1.1.

Portions of COTS software that have been modified by the vendor in any manner are subject to review. — [4] II.5.2.

Source code generated by a COTS package and embedded in software modules for compilation or interpretation shall be provided in human readable form to the accredited test lab. — [4] II.5.2.

The accredited test lab may inspect the COTS generated software source code in preparation of test plans and to provide some minimal scanning or sampling to check for embedded code or unauthorized changes. — [4] II.5.2.

For purposes of applying the Standards, firmware is considered a form of software. — [2] I.A (definition of firmware). This text was replaced in [4].

3.3 Coding standards

The requirements of this section apply generally to all software used in voting systems, including software furnished by an external provider (for example, providers of COTS operating systems and web browsers) where the software may be used in any way during voting system operation. — [4] I.5.1.1.

All software components designed or modified for election use shall be tested in accordance with the applicable procedures contained in this section. — [4] II.5.2.

3.4 Logic verification

These coding standards are a means to an end, the end being an ITA evaluation of the code's correctness to a high level of assurance. The TGDC requests that NIST recommend standards to be used in evaluating the correctness of voting system logic, including but not limited to software implementations. — TGDC Resolution #29-05.

The logic verification approach was presented at the September 2005 TGDC meeting. No change of direction was recommended by the TGDC at that time.

> There are three types of indicia used to assess system accuracy, reliability, and correctness. One involves the absolute logical correctness of all ballot processing software. In this case, no margin for error exists. — [1] 7.1.1.
>
> The testing process involves the assessment of absolute correctness of all ballot processing software, for which no margin for error exists. — [2] I Overview.

The above language was retained in [3], but it does not appear in [4].

> If a malfunction is due to a defect in software, then the test shall be terminated and system returned to the vendor for correction. — [4] II.1.8.2.6.b.

4 On a more precise definition of COTS

Some discussion occurred in the Security and Transparency Subcommittee towards providing a more precise definition of COTS than was given above. However, issues that were raised in that discussion were never brought to a conclusion.

The following definition is quoted from [5] though very similar text appears in [6].

> QUALIFICATION: If a component is to be deemed "COTS" for the purpose of the VVSG, it must meet the following criteria:
>
> > a. it must publicly available at the time a vendor submits a voting system containing that component for evaluation. (It may be for sale or may be available for free.)
> >
> > b. it must be in widespread use outside of the vendor's use of that component (e.g. at least 10,000 instances of that component in other products or applications),
> >
> > c. it must be maintained by an organization that has been in existence for at least seven years,
> >
> > d. that organization must carefully mark each version of that product with explicit version numbers, and, in the case of software, provide cryptographic hashes of the software versions in a public manner.

Unresolved issues:

(b) A benchmark something like this would be helpful, but one generally cannot verify the number of deployments. For software that is licensed or registered, one has a count of the number of such licenses or registrations. For software that is neither licensed nor registered, all one has is a count of the number of times it has been downloaded from its site of origin (downloads from mirror sites would not be traceable).

(c) The objective of this requirement was to increase the likelihood of an organizational commitment to continued support, but the relationship between organizational age and support longevity in the technology sector is highly unclear. Mature organizations are known to unilaterally discontinue support for their own products or the products of former competitors whom they have bought out. Contractual terms requiring continued support may be difficult to enforce through bankruptcies, takeovers, etc.

5 Proposal for approved COTS list

Excerpted from [7].

> There were two things discussed in the telephone conference yesterday that I think would go a long way toward relieving the COTS problem. These could be included in the Guidelines or in the EAC procedures, or both.
>
> 1. Develop a list of devices and software that are pre-approved as COTS for use in voting systems. Initially this list would be empty. The next point describes how the list would be populated.
>
> 2. When a vendor submits a voting system to an ITA, it must clearly identify all devices and software that the vendor considers to be COTS. These items would be evaluated by the ITA as specified in the Guidelines. The subsequent ITA report to the EAC would indicate whether or not the ITA recommends the COTS devices and/or software for inclusion in the list of pre-approved COTS. This recommendation would go through the review and approval process as set up by the EAC. If the device and/or software is approved it would be added to the list of approved COTS.
>
> Another approach could be to allow a vendor to submit proposed COTS devices and software to the EAC approval process prior to including them in a voting system. This pre-approval would keep the vendors from expending research and development funds on devices that do not qualify as COTS.

For the approved COTS list to be relevant in a voting system evaluation, it would be necessary to establish that the use of the COTS product in the new system is comparable to its use in the previously approved system and is in fact the same version and configuration of the product.

6 References

[1] Performance and Test Standards for Punchcard, Marksense, and Direct Recording Electronic Voting Systems, January 1990 edition with April 1990 revisions, in Voting System Standards, U.S. Government Printing Office, 1990. Available at http://josephhall.org/fec_vss_1990_pdf/1990_VSS.pdf.

[2] 2002 Voting Systems Standards, available from http://www.eac.gov/election_resources/vss.html.

[3] Voluntary Voting System Guidelines Version I Initial Report, 2005-05-09, available from http://vote.nist.gov/VVSGVol1&2.pdf.

[4] 2005 Voluntary Voting System Guidelines, Version 1.0, 2006-03-06, available from http://www.eac.gov/vvsg_intro.htm.

[5] E-mail from Ron Rivest, "Thought on COTS Software," 2006-05-17.

[6] Memo from Stephen Berger, "Supplemental Guidance on COTS," 2006-06-06.

[7] E-mail from Brit Williams, "More COTS," 2006-06-30.

Notes

[1] Commercial equipment and materials are identified in order to describe certain procedures. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.