

**Testimony before the
Committee on House Administration
Election Subcommittee Hearing on Election Reform
March 15, 2007**

**Britain J. Williams, Ph.D., Professor Emeritus
Kennesaw State University**

Introduction

I would like to begin by thanking the Committee for the opportunity to appear before you. I have worked in the arena of computer based voting systems for over 20 years and appreciate this opportunity to share with you my experience and opinions on this important matter of open source software for voting systems. I will begin with some background information and then conclude with some specific recommendations.

Background

The following definition and description of open source software is intended to give the Committee a sense of what the various panel members intend when they take the position that voting system software should be *open source*. The two key points in the following are that under open source our voting system software would be “made available to the general public with either relaxed or non-existent intellectual property restrictions” and that this “allows the users (i.e. the general public) to create user-generated software.”

The examples that are listed are mostly very specialized applications that are not in use by the ‘general public.’ For example, OpenOffice.org¹ is designed to compete with Microsoft Office. Although this product is free, I would be surprised to learn that a single member of this Committee has replaced their Microsoft Office suite with OpenOffice.org.

Definition: *Open source* describes the principles and methodologies to promote open access to the **production** and **design** process for various goods, products, resources and technical [conclusions](#) or advice. The term is most commonly applied to the [source code](#) of [software](#) that is made available to the general public with either relaxed or non-existent [intellectual property](#) restrictions. This allows users to create [user-generated software content](#) through either incremental individual effort, or [collaboration](#).

[Open source software](#) — software whose source code is published and made available to the public, enabling anyone to copy, modify and redistribute the source code without paying royalties or fees. Open source code evolves through

¹ <http://en.Wikipedia.org/wiki/OpenOffice.org>

community cooperation. These communities are composed of individual programmers as well as very large companies. Examples of open-source software products are:

- [Linux kernel](#) - operating system kernel based on [Unix](#)
- [Eclipse](#) - An IDE primarily for doing Java development, but has enough plug-ins to make it a software that can do virtually anything from programming in multiple technologies to creating Word documents and checking e-mail
- [Apache](#) - HTTP web server
- [Tomcat web server](#) - Java web/servlet-container
- [Blender](#) - 3D graphics application
- [Moodle](#) - course management system
- [Mozilla Firefox](#) - web browser
- [Mozilla Thunderbird](#) - e-mail client
- [OpenOffice.org](#) - office suite
- [OpenSolaris](#) - Unix Operating System from Sun Microsystems
- [Project.net](#) - Commercial Open Source Project Management
- [Mediawiki](#) - [wiki](#) server software, the software that runs Wikipedia
- [Aras Innovator](#) - open source business process management enterprise software
- [Drupal](#) - [content management system](#)

- [Joomla!](#) - [content management system](#)
- [GNU Compiler Collection](#) - Programming language compiler for C, C++, Java and other languages.
- [phpBB](#) - open source bulletin board system
- [Nvu](#) - open source WYSIWYG HTML editor (webpage/website builder)
- [Audacity](#) - open source audio recording software
- [StCAD](#) - open source 3D Framework for Smalltalk
- [Adempiere](#) - open source ERP/CRM
- [FileZilla](#) - open source FTP-Client ²

Testing and Certification of Voting Systems

The primary reason that is given for requiring voting system software to be open source is that open source would allow the public to verify the accuracy of the voting system software and detect any fraudulent code that may be present in the voting system software. In other words, open source would allow an extensive and through testing of the voting system source code.

Yet nowhere in the definition of open source is testing even mentioned. The definition of open source clearly states that the purpose is to allow users to modify the software to suit their own individual needs. Clearly, this is not the intent of this Committee.

Voting systems and their associated software currently undergo extensive tests and all of these tests are open to the public. Specifically, voting systems are tested at four different levels:

Federal
State
Local Acceptance

² http://en.wikipedia.org/wiki/Open_Source

Local Logic and Accuracy

The following sections give a brief description of the tests performed at each level.

Federal Level Testing: From the mid 1990's until recently, voting systems were tested for compliance with the voting systems standards developed by the Federal Election Commission. These tests were under the direction of the National Association of State Election Directors.

As required by the HAVA, the Election Assistance Commission has developed Voluntary Voting Systems Guidelines (VVSG) and has put in place a process to ensure that voting systems comply with these Guidelines. Under EAC direction, Voting System Test Laboratories (VSTL) examine the voting systems for compliance with the Guidelines. To become a VSTL, a laboratory must first be certified by NIST and then be approved by the EAC.

Based on the results of the VSTL examinations and any other information at their disposal, the EAC will certify the voting system as being compliant with the VVSG.

One of the tests performed by the VSTL is an examination of the voting system source code.

After EAC certification has been granted, the VSTL delivers the source code and the object code along with their digital signatures to a trusted archive designated by the EAC.

State Level Testing: After a voting system is certified by the EAC, each state that wishes to consider using the voting system conducts a state level test of the system. Historically, these state level tests have been little more than a review of the voting system for compliance with state law; however, most states have now responded to the increasing concern for voting system security by implementing state level tests that approach the rigor of the EAC tests.

Many states require the vendor to submit source code to the state. The state conducts a review of the source code and then archives the source code for future reference as needed.

Local Level Acceptance Testing: Most states will not allow a local jurisdiction to purchase a voting system until that system has received EAC Certification and State Certification. Upon delivery, the local jurisdiction conducts tests, called acceptance tests, that test the system for compliance

with the conditions of the procurement and to verify that the system delivered is identical to the system that underwent EAC and State Certification.

Local Logic and Accuracy Testing: Prior to each election, the local jurisdiction conducts tests called Logic and Accuracy Tests (L & A Tests). These tests are a simulated test of all of the ballot styles and the entire set of voting system devices that are to be used in the upcoming election.

The voting system configuration for each precinct is set up and the ballot styles for that precinct loaded on the devices. Then ballots are cast on the system in accordance with a known pattern of votes. The precinct is then closed and the results recorded on the voting system are compared to the results of the known pattern of voting. The local election officials must account for any discrepancies.

These L & A tests are public tests that must be advertised in the local legal organ prior to the tests.

Organizational Use of Open Source Code

Every agency in government and every major business entity have software that is considered mission critical. I am not aware of a single organization that makes their mission critical software available to the general public. The reason is simply that open source software is vulnerable to attack from everyone from teen age hackers to foreign terrorists.

Voting system source code is mission critical to successful elections. Placing this source code in the hands of hackers and terrorists clearly creates the potential for harm to the integrity of elections. In addition, substantial harm can be done to a voting system by well-meaning members of the public. On the other hand, there are advantages to be gained from making this source code available to responsible reviewers.

It is recommended that the EAC be granted the authority to make voting system source code available to responsible individuals. Persons wishing to review voting system source code should be required to make application to the EAC; providing their credentials for reviewing the software, their 'need to know', and the specific voting system software they wish to review. A recipient of voting system software should be required to sign a nondisclosure agreement and to return or destroy the software when their review is completed. Source code should only be provided to individuals, not organizations.

A Specific Recommendation

The following gives a recommended outline for allowing access to voting system source code. This recommendation is based on the belief that voting system source code should only be issued to individuals (not organizations) that are working under the direction a state or local election official. It also contains a feature that will allow the identification of any source code that is leaked to any unauthorized individual or organization. Finally, it is strongly recommended that there be specific, well defined penalties for violating the confidentiality of voting system source code. It has been previously demonstrated that US Patent Laws and laws designed to protect proprietary information are not sufficient to protect voting system source code.

1. The cognizant election official must file with the EAC an application with the EAC requesting that a specific individual or group of individuals be allowed access to the source code for a specific voting system. This application must clearly state the reason for the request.

2. Each individual named in the application must then provide the EAC with the following information:
 - The reason the individual wishes access to the source code.
 - The qualifications of the individual to evaluate source code.
 - The schedule that the individual intends to adhere to while reviewing the source code.

3. The individual must sign a non-disclosure statement agreeing that (s)he will not disclose the source code to anyone that has not been approved by the EAC. The agreement must also specify that the individual cannot release any report or press release based on the review of the source code until the report or press release has been approved by the EAC. This agreement should clearly state the penalty for disclosing the source code to any unauthorized individual.

4. Once the EAC approves the application, the individual must undergo background checks by the Office of Homeland Security.

5. When steps 1, 2, 3, and 4 have been successfully completed the EAC will furnish the individual with a digitally signed copy of the source code. This digital signature must be unique to the point that it cannot be altered or duplicated.

6. When the schedule in step two expires the individual must either return the digitally signed source code to the EAC or apply for an extension.

Thank you

Again, I wish to thank the Committee for the opportunity to address these important issues. I sincerely hope that I have made at least a small contribution to the work of this committee.

In closing, I would like to state that the opinions presented in this paper are entirely my own. They do not represent the opinions of Kennesaw State University or the Office of the Georgia Secretary of State.

Respectfully submitted:

Britain J. Williams, Ph.D.
Professor Emeritus of Computer Science and Information Systems
Kennesaw State University

Brit Williams is Professor Emeritus of Computer Science and Information Systems at Kennesaw State University. He has worked in the field of computing since 1957. He has directed large computer centers and computer networks in industry, government, and academia. One of his primary research interests since 1986 has been computer-based voting systems. He was a consultant to the FEC during the development of the 1990 Voting System Standards and the 2002 Voting System Standards. He was a member of the NASED Voting Systems Board and Chair of the NASED Voting Systems Board Technical Committee from their inception until 2007. He represents NASED on the Technical Guidelines Development Committee created by the Help America Vote Act. Dr. Williams has been conducting certification evaluations of computer-based voting systems for the State of Georgia since 1986. He also has assisted the states of Pennsylvania, Maryland and Virginia with certification evaluations of computer-based voting systems.