MEMORANDUM

Date: November 26, 2003

To: Dean Heller, Secretary of State

From: Marc McDermott, Chief
Electronic Services Division

Subject: Diebold and Sequoia Voting Machine Security

Summary

I believe the Diebold electronic voting machine, operating on the software analyzed in the Johns Hopkins report and the SAIC Risk Assessment Report, represents a legitimate threat to the integrity of the election process. Conversely, based on available information with regard to the Sequoia Voting System, I believe the Sequoia electronic voting machine represents a much more secure option because of the increased security of the customer (voter) interface and by the fact that the Sequoia operating software has not been made available on the Internet.

Supporting Information

I reviewed the information supplied with your letter of October 2, 2003 regarding an analysis (the Johns Hopkins report) of Diebold electronic voting machines. This analysis, written by personnel from the Information Security Institute of Johns Hopkins University and the Department of Computer Science of Rice University, was based on Diebold electronic voting machine software that had been put on the Internet. The analysis pointed out numerous security problems and areas of vulnerability with the Diebold machines. Unfortunately, after reviewing the report, I would agree with the report's findings and believe that the majority of the problems listed are valid.

Based on my experience in counter-terrorism and as Chief of the Gaming Control Board's Electronic Services Division, I believe the primary target for a hacker would be the voter interface. At these points, the public comes into physical contact with the voting machines. If someone wanted to attack a voting machine, the normal voting process would allow this person access to a machine for several minutes. During this time, when he was supposed to be voting, the attacker could try to gain entry to the machine's software to either change how the machine operates or change the stored voter information. The only interfaces or data entry points available to the attacker at this time are the touch screen and the smart card reader. Of these, only the smart card reader provides

a reasonable path to get deep enough into the voting machine's operating software to pose a threat to the machine. It is in this critical interface that the Diebold and Sequoia machines differ significantly. According to the Johns Hopkins report, the Diebold machines send and accept data from the smart card reader in plain, unencrypted text. I agree with the Johns Hopkins report's conclusion that this is a poor choice. As explained on pages 11 and 12 of that report, the clear-text messages allow for a compromise of both the administrative password and the password used to authenticate the terminal to the smart card. As a result, I believe that the availability of smart cards, smart card readers and smart card development tools, combined with both the Diebold software and the Johns Hopkins report on the Internet, represent a real threat to any election using the Diebold machines.

Conversely, the Sequoia machine provides basic encryption of the information sent between the smart card and the voting machine. This reason alone would be enough to recommend the Sequoia over the Diebold system, as I would see an obvious flaw such as sending unencrypted data to the smart card as indicative of other serious security issues. However, in addition to a software security concern, Diebold's source code for their voting machine was put on the Internet. This immediately raises many questions regarding the security of the entire Diebold software development and management process. There should be very tight internal controls for software development of this kind. Since this information got to the Internet, it casts a shadow of uncertainty on the security consciousness of the entire development team.

The Diebold system has been evaluated by Johns Hopkins and SAIC and both evaluations have indicated serious security deficiencies. However, as stated in the SAIC report, when taken in the context of a real election, the internal controls governing the election will greatly reduce or completely eliminate the majority of the deficiencies found. For example, if a password for a voting terminal is extracted and fake smart cards are made, they would only be valid on one particular voting machine. At a polling place, voting officials, not voters, choose what terminal a voter is to use. As there are many voting terminals at most polling places, many fake cards would be necessary and would have to be distributed to many confederates hoping that one would be assigned to the terminal where the card would work. Additionally, there is a very good chance that any errors such as extra votes or other irregularities caused by these fake cards would be detected after the polls closed and the votes were tallied. The problem is that, if even one fake card was shown to work, even if it was discovered after the polls closed, it may cast doubt on the entire election process regardless of whether the election was actually flawed or not.

I have reviewed the article concerning the Sequoia software that was left unprotected on a publicly available server. I see this as unfortunate but vastly different from the Diebold incident. In the Sequoia case, the software was "leaked" by a government contractor not by Sequoia. Also, the software that was

leaked was for systems that display election results, i.e. systems that use raw voter data not systems, such as the voting machine, that gather raw data. This is a tremendous difference because as long as the raw voter data is intact, the true outcome of an election can be determined. In Diebold's case, the software released was for the voting machines that gather the raw data which is inherently more serious. The next significant difference is that the Sequoia code released was binary machine code. This type of software is significantly more difficult to use. The article states that the binary code must be reverse engineered in order to understand how it works. I agree with that assessment. The article also states that this process "is not hard to do". I strongly disagree with that assessment. Although it is certainly possible to reverse engineer binary code, I believe, based on my work with assembly language and binary programming that the reverse engineering process for binary code would be at least hundreds, if not thousands of times more difficult than analyzing software where the source code and programmer comments were available.

There are other points of attack that, depending on the access gained, could seriously disrupt an election. For example, successful attacks on the system that develops the election definition files that are loaded into the voting machines or the back-end server that evaluates the results could create enough confusion to delay election results until the errors are determined and corrected. Typically, though, these areas are much more difficult to access. In an actual election setting, the set of internal controls, physical locks, secured areas and limited access computers, if configured correctly, greatly reduce the risk of these systems being compromised regardless of the strengths or weaknesses of security features in the actual software. I believe this extra security, above and beyond the security of the particular software, greatly reduces the likelihood that someone will be able to access this part of the system, and increases the expertise required of an attacker that chooses this portion of the election system as a target.

As part of my analysis, I asked numerous questions regarding Sequoia's implementation of the voter interface as well as the entire election package. I believe, based on the answers I received and the results in the two reports on the Diebold system that the Sequoia units are more secure. This, however, does not mean that they are perfect. There are several areas where the security of the Sequoia system could be improved over time. Suggestions regarding some proposed improvements could be compiled and forwarded to your office upon request

While I believe Sequoia's approach to be inherently better than Diebold's, the damage done to Diebold by having its source code placed on the Internet cannot be overstated. Secrecy does not, by itself, ensure security. However, the probability of success of most attacks increases as information regarding the target increases. Bank robbers, for example, study their targets before robbing them. Computer hackers electronically survey their targets by "footprinting" them

before attacking. I would expect people who attack voting systems to follow the same pattern. Therefore, since Diebold's source code has been made available I would expect people to be studying this software and developing hardware/software items to take advantage of the Diebold voting machines in some future election.

On the other hand, the Sequoia voting machine software is still secret. As people have not had the opportunity to evaluate the Sequoia software, they have a significantly lower likelihood of mounting a successful attack on a Sequoia voting machine. However, based on what I have learned about Diebold's and Sequoia's voting systems, perhaps an evaluation by a trusted independent entity, such as SAIC that was employed by Maryland, and a set of technical standards listing minimum security performance criteria, would add security and peace of mind concerning the voting process to all.

I hope this provides the information you requested concerning the Diebold and Sequoia voting systems. Please contact my supervisor, Board Member Scott Scherer, or myself at your convenience if you have any questions or require any additional information.