Analyzing the security of Internet Voting: SERVE, a Case Study

David Jefferson[1], Aviel D. Rubin[2], Barbara Simons[3], and David Wagner[4]

1. Introduction

The authors are four of a group of eight computer scientists and security experts who reviewed the Pentagon's $22 million program for voting over the Internet, called SERVE (Secure Electronic Registration and Voting Experiment). Shortly after the release of our report in January 2004 [JRSW04], the Pentagon decided not to implement SERVE in the 2004 election, citing security concerns. It is still possible that a program similar to SERVE could be proposed for future elections.

This paper, a much shortened version of the full report, describes the security issues that we identified with SERVE, most of which apply to Internet voting in general. To simplify the presentation, we use the present tense throughout, even though there are no longer current plans to use SERVE in any election.

1.1 What is SERVE?

SERVE is an Internet-based voting system built by Accenture and its subcontractors for the U.S. Department of Defense FVAP (Federal Voting Assistance Program). FVAP's mission is to reduce voting barriers for all citizens covered by the Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA), namely U.S. citizens who are members of the Uniformed Services, their family members, and U.S. citizens who reside outside the United States. SERVE allows UOCAVA voters both to register to vote and to vote via the Internet, from anywhere in the world. It is intended to be a complete, ITA-qualified and state-certified voting system that collects real votes.

To participate, an eligible voter first enrolls in the SERVE program. After enrollment, the voter may register to vote, and then votes in one or two short sessions from any Internet-connected PC. The PC must run a Microsoft Windows operating system and either the Internet Explorer or Netscape web browser. The browser must be configured to enable JavaScript, along with either Java or ActiveX scripting, and session cookies; no additional hardware or software is required.

When a person registers online to vote, his or her information is stored on the central web server for later download by the Local Election Official (LEO), at which point the LEO updates its database. When a person votes in the election, the completed ballot is stored on the central server and later downloaded by the LEO, who stores it for canvass. The communication between the user's web browser and the central server is protected using

[1] Lawrence Livermore National Laboratory, d_Jefferson@yahoo.com
[2] Johns Hopkins University, rubin@jhu.edu
[3] simons@acm.org
[4] University of California at Berkeley, daw@cs.berkeley.edu

the Secure Socket Layer (SSL) protocol. Once that connection is established, an ActiveX control is downloaded to the voter's PC and run to provide functionality not available in current browsers.

Besides being restricted to overseas voters and military personnel, in the 2004 trial SERVE was to be limited to people who vote in one of 50 counties in the seven states (Arkansas, Florida, Hawaii, North Carolina, South Carolina, Utah, and Washington) that agreed to participate. The 2004 trial was expected to handle up to 100,000 votes over the course of the year (as many votes as a small state), including both the primaries and the general election. By comparison, approximately 100 million votes were cast in the 2000 general election. One goal was to determine if a similar system might be suitable for expansion to all 6 million UOCAVA voters.

Limited though it is, SERVE is a real voting system. Systems similar to SERVE might eventually be offered by Accenture or other vendors for use by all voters, instead of just a limited population. For these reasons we analyze SERVE not as an experiment, but as a real voting system whose use could be significantly expanded in future years.

1.2 Our Recommendations

The following is a summary of our findings and recommendations:

a) DRE (direct recording electronic) voting systems have been widely criticized for various deficiencies and security vulnerabilities: that their software is totally closed and proprietary; that the software undergoes insufficient scrutiny during qualification and certification; that DREs are especially vulnerable to various forms of insider (programmer) attacks; and that DREs have no voter-verified audit trails (paper or otherwise) that could largely circumvent these problems. All of these criticisms of DREs apply directly to SERVE as well.

b) But in addition, because SERVE is an Internet- and PC-based system, it has numerous other fundamental security problems that leave it vulnerable to a variety of well-known cyber attacks (denial of service attacks, spoofing, automated vote buying, viral attacks on voter PCs, etc.), any one of which could be catastrophic.

c) Such attacks could occur on a large scale, and could be launched by anyone in the world from a disaffected lone individual to a well-financed enemy agency outside the reach of U.S. law. These attacks could result in large-scale, selective voter disenfranchisement, and/or privacy violation, and/or vote buying and selling, and/or vote switching even to the extent of reversing the outcome of many elections at once, including the presidential election. Some of the attacks could succeed and yet go completely undetected. Even if detected and neutralized, such attacks could have a devastating effect on public confidence in elections.

d) It is impossible to estimate the probability of a successful cyber-attack; but we show that the attacks we are most concerned about are quite easy to perpetrate. In some cases there are kits readily available on the Internet that could be modified or used

d

Comment [1]: I'm not sure it's good tactics to link this to the DRE debate; our arguments can stand alone. DRJ: Maybe I misunderstand, but I don't think the mere mention of public confidence as an issue is a reference to the DRE debate. I, too, think this SERVE argument must stand alone.

directly for attacking an election. And we must consider the obvious fact that a U.S. general election offers one of the most tempting targets for cyber-attack in the history of the Internet, whether the attacker's motive is overtly political or simply self-aggrandizement.

e) The vulnerabilities we describe cannot be fixed by design changes or bug fixes in SERVE. Instead, they are fundamental in the architecture of the Internet and of the PC hardware and software that is ubiquitous today. The vulnerabilities cannot be eliminated for the foreseeable future without a wholesale redesign and replacement of much of the hardware and software security systems in the PC and the Internet, or else some unforeseen radical security breakthrough(s).

f) We have examined numerous variations on SERVE; however, all such variations suffer from the same kinds of fundamental vulnerabilities. Regrettably, we cannot recommend any of them. We do suggest a kiosk architecture as a starting point for designing an alternative voting system with similar aims to SERVE, but which does *not* rely on the Internet or on unsecured PC software. (See [JRSW04] Appendix C).

g) A seemingly successful voting experiment in a U.S. presidential election involving seven states would likely be viewed by most people as strong evidence that SERVE is a reliable, robust, and secure voting system. Such an outcome would encourage expansion of the program by FVAP in future elections, or the marketing of the same voting system by vendors to jurisdictions all over the United States, and other countries as well.

However, the fact that no successful attack is detected does not mean that none occurred. Many attacks, especially if cleverly hidden, would be extremely difficult to detect, even in cases when they change the outcome of a major election. A "successful" trial of SERVE in 2004 is the top of a slippery slope toward even more vulnerable systems in the future. (The existence of SERVE was cited as justification for Internet voting in the Michigan Democratic caucuses earlier this year.)

h) Like the proponents of SERVE, we believe that there should be better support for voting for members of the military overseas. Still, because the danger of successful, large-scale attacks is so great, we reluctantly recommend shutting down the development of SERVE immediately and not attempting anything like it in the future until the security problems of the PC and the Internet are resolved.

The remainder of this paper explains some of the reasoning behind these conclusions.

1.3 Vulnerabilities in SERVE

Because the Internet knows no national boundaries, an election held over the Internet is vulnerable to attacks from anywhere in the world. Not only could a political party attempt to manipulate an election by attacking SERVE, but so could individual hackers, criminals, terrorists, organizations such as the Mafia, and *even other countries*. There is

no need to postulate a large conspiracy or highly sophisticated adversaries; many of the attacks we describe could be mounted by lone individuals with college-level training in computer programming. In this section we give a short description of a variety of attacks that can be mounted against SERVE.

Lack of voter-verified audit trail and insider attacks. Paperless DRE (direct recording electronic) voting systems have been widely criticized because they are unauditable. There is no way that a voter can verify that the vote recorded inside the machine is the same as the vote that he or she entered and saw displayed on the machine's screen, and if serious problems subsequently occur in the canvass of the votes (which happens all to frequently), there is no *independent* audit trail of the votes to help resolve the problem. Voter verification is the only readily available effective defense against programmed insider attacks. Every argument about the need for voter verification and auditability that has been made about DREs applies essentially unchanged to SERVE.

Privacy. The privacy of SERVE ballots is protected using encryption. When the ballot is cast, it is encrypted during transmission over the Internet and decrypted at the central server. Once received, the ballot is separated from the voter's identity and the anonymous ballot is re-encrypted so that only the LEO of the voter's district can read it. These encrypted ballots are stored at the central server and can be downloaded (in randomly re-ordered form) upon request by LEOs.

This architecture introduces several privacy risks. First, a LEO could deduce how voters in his/her precinct have voted by downloading votes from SERVE so frequently that he/she gets at most one new vote and voter name each time. This would allow a curious LEO to infer how each individual voted. Second, the brief existence of cleartext ballots on the server introduces a risk that SERVE system administrators could view how individuals voted. Likewise, if SERVE machines were subverted by hackers, the privacy of all votes could be compromised. Third, SERVE's retention of encrypted ballots for 18 months or longer could compromise voter privacy if this information were to land in the wrong hands and old system keys were exposed.

Vote Buying/Selling. Vote selling is a problem in all elections, but it is a special concern for Internet voting, since large scale vote buying and selling can be automated. During the 2000 presidential election we saw the creation of several web sites to facilitate vote swapping between Gore and Nader voters. While the Gore/Nader swapping depended on the honor system and no money changed hands, a similar approach could be used with SERVE to provide *enforced* vote swapping or vote bartering services, or to purchase votes from SERVE voters.

The most straightforward vote-buying scheme would involve the selling of personally identifiable information and the voter's password or private key. One possible defense would be for SERVE to prohibit the submission of multiple votes from the same Internet address. This is not a strong defense, however, because a purchaser of votes could fool SERVE into thinking that the votes were coming from different addresses, and because legitimate users often appear to come from the same IP address.

Another approach to vote-buying would be for the buyer to provide the seller with a version of the SERVE ActiveX component that is modified to ensure that the voting is done according to the wishes of the vote purchaser. There does not seem to be any way for SERVE to defend against this style of vote buying.

Large-Scale Impact. When voting is conducted at physical precincts on mechanical devices or with paper ballots, whatever vote manipulation occurs happens on a relatively small scale. By contrast, since SERVE is vulnerable to many different types of attacks, a significant percentage of votes cast over the Internet could be vulnerable. A single malicious party could potentially affect tens of thousands of votes cast through SERVE, while it is extremely unlikely that any single person could conduct vote fraud on such a large scale in existing non-electronic elections.

Table 1 is a summary of the major vulnerabilities we have identified in SERVE, along with our assessment of them.

Threat	**Skill needed**	***Consequences***	**Realistic?**	**Countermeasures**
denial of service attack (various kinds)	low	disenfranchisement (possibly selective disenfranchisement)	common on the Internet	no simple tools; requires hours of work by network engineers; launchable from anywhere in the world
Trojan horse attack on PC to prevent voting	low	disenfranchisement	There are a million ways to make a complex transaction such as voting fail.	can mitigate risk with careful control of PC software; reason for failure may never be diagnosed
on-screen electioneering	low	voter annoyance, frustration, distraction, improper influence	trivial with today's web	nothing voter can do to prevent it; requires new law
spoofing of SERVE (various kinds)	low	vote theft, privacy compromise, disenfranchised voters	Web spoofing is common and relatively easy	none exists; likely to go undetected; launchable by anyone in the world
client tampering	low	disenfranchisement	one example: change permissions on cookie file. Many other trivial examples	none exists for all possible mechanisms. Too difficult to anticipate all attacks; likely never diagnosed.
insider attack on system servers	medium	complete compromise of election	Insider attacks are the most common, dangerous, and difficult to detect of all security violations	none within SERVE architecture; voter verified ballots needed; likely undetected
automated vote buying/selling	medium	disruption of democracy	very realistic, since voter willingly participates	none exists; buyers may be out of reach of U.S. law
coercion	medium	disruption of democracy	harder to deploy than vote buying/selling, but man in the middle attacks make it achievable with average skill	none exists; likely to go undetected.
SERVE-specific virus	medium or high	vote theft, privacy compromise, disenfranchised voters	Some attacks require only experimentation with SERVE; others require leak of SERVE specs or code and	virus checking software can catch known viruses, but not new ones; likely to go undetected

			resourceful attacker	
Trojan horse attack on PC to change votes or spy on them	high	vote theft, privacy compromise	widely available spyware would be a good starting point	can mitigate risk with careful control of PC software; harder to control at cybercafe or other institutionally managed networks; likely to go undetected

Table 1 This table describes, for each potential threat to SERVE, what skill is required by the attacker, the consequences of a successful attack, how realistic the attack is, and what countermeasures might be used to thwart the attack.

The remaining sections of the paper explicate in some detail three of the most important of the vulnerabilities in SERVE. For further information see the original report [JRSW04].

2. Lack of Control of the Voting Environment

Perhaps the greatest challenge with Internet voting arises from the fact that electoral authorities do not have control over all the equipment used by voters. Since SERVE's voters can vote on their own computers or on computers controlled by others, third parties might be able to gain control of a large number of computers used for voting. Such attacks could result in the loss of voter privacy, disenfranchisement, or vote alteration without anyone, including the voter and election officials, noticing or detecting any problem.

The Computers. Voters' personal computers are unlikely to be as carefully defended as corporate ones, and hence voters' machines are especially susceptible to attack. Attacks can be easily automated; hackers routinely scan thousands or even millions of computers in search of those that are easiest to compromise. A relatively easy way to disenfranchise the voter is to disable ActiveX or Web cookies so that it is no longer possible to vote through SERVE. Alternatively, a malicious third party could cast an unauthorized ballot that appears to come from the voter.

A shared computer, for example at a cybercafe or public library, is even more insecure. The owner, the system administrator, or even a prior visitor could have installed remote spying or subversion software.

Voting from workplaces entails similar risks. One study found that 62% of major US corporations monitor employee's Internet connections, and more than one-third store and review files on employees' computers.[5]

The Software. Pre-installed software applications also pose risks. Backdoors placed in software and activated when a user tries to vote could invisibly monitor or subvert the voting process. Software security vulnerabilities could allow a remote attacker to take

[5] http://www.amanet.org/research/pdfs/ems_short2001.pdf

complete control of a computer using remote control software such as PCAnywhere or BackOrifice. Successful penetration of even well-defended computers is routine.

Viruses and Worms. One of the most dangerous forms of remote attack is a virus or worm that spreads itself and contains a malicious payload designed to take control of machines and wreak havoc with a future election. Since virus checking software defends against only previously known viruses, virus checkers often are unable to keep up with the spread of new viruses and worms. In 2001 the Code Red worm infected 360,000 computers in 14 hours, and in 2003 the Slammer worm brought down many ATM machines and compromised many Internet hosts.[6] Modern worms are even more virulent, are often spread by multiple methods, are able to bypass firewalls and other defenses, and can be difficult to analyze. For example, it took quite a while to determine that SoBig.F was a Trojan horse designed to plant spam engines.[7]

Attackers can build new viruses, or modify existing viruses sufficiently that they will avoid detection. Virus construction kits are available on the Internet. In addition, the attacker has the advantage that he/she can test new versions of viruses using the same publicly available virus checkers that potential victims use, thus confirming that the virus will not be detected before its release.

Web sites. A dangerous hybrid attack involves placing malicious content on specially chosen websites. For instance, an attacker with a vendetta against one candidate might booby-trap the website of that candidate, so that those who visit the candidate's website are unable to vote using SERVE. Such selective disenfranchisement might eliminate several hundred votes for a candidate, enough to throw the election to his/her opponent.

3. Spoofing and Man-in-the-Middle Attacks

In *man-in-the-middle attacks* the adversary interposes itself between legitimate communicating parties and simulates each party to the other. To simplify the discussion, we focus primarily on ways that a man-in-the-middle attack can subvert voter privacy, although the same general technique can be used for other attacks, e.g. vote buying.

The use of SSL does little to mitigate man-in-the-middle attacks on privacy. Any man-in-the-middle could act as an SSL gateway, forwarding application data between the voter and the vote server unaltered. The attacker could see all of the traffic by decrypting and re-encrypting as communications pass between the two. In effect, the attacker would communicate using two SSL sessions, one between itself and the voter, and the other between itself and the vote server, and neither would know that there was a problem. These attacks are possible because the voter's browser does not verify that it is talking to the real SERVE web server – only that it is talking to someone in possession of a valid SSL certificate (which could be an attacker).

Man-in-the-middle attacks also could be used to disenfranchise voters by spoofing the

[6] http://securityresponse.symantec.com/avcenter/venc/data/w32.sqlexp.worm.html
[7] http://securityresponse.symantec.com/avcenter/venc/data/w32.sobig.f@mm.html

entire interaction with the voter. SERVE has some safeguards in place, but they assume that the voter knows exactly what to expect from the voting experience; it is likely that an attacker could create a voting experience that the voter would believe is real. Similarly, voters could be led to believe that they registered successfully, when in fact they were communicating directly with an adversary instead of the legitimate registration server. The voters would discover when attempting to vote that they were not registered, but at that point there might be nothing they could do.

Perhaps the most serious consequence of man-in-the-middle attacks is that attackers could engage in election fraud by spoofing the voting server and observing how the voter votes. If the vote is to the attacker's liking, the voter is redirected to SERVE's legitimate voting site. If the attacker does not like the vote, then the entire voting session is spoofed; in this case, the user thinks he/she has voted, but in fact the vote will not be received or counted by SERVE.

4. Denial-of-service attacks

Attacks in which legitimate users are prevented from using the system by malicious activity such as overloading the election web server are known as *denial-of-service attacks*. A particularly nasty variant of denial-of-service attack is the *distributed-denial-of-service* (DDoS) attack. In this scenario, an attacker typically takes control of many computers in advance by spreading a custom-crafted virus or worm. In computer security jargon, the compromised machines are often known as "zombies" or "slaves," because the attacker leaves behind hidden software that causes infected machines to blindly obey subsequent commands from the attacker. Automated tools for mounting DDoS attacks have been circulating among the hacker community since at least 1999 [HW01], and hackers routinely amass large "zombie networks" of compromised machines. In February 2000, major DDoS attacks were mounted against several high-profile web sites, including CNN, Yahoo and eBay.[8] It was later discovered that these damaging attacks had been perpetrated by a lone teenager not on US soil.[9]

Since then, DDoS attacks have become routine. One study recorded over 10,000 denial-of-service attacks during a three-week period in 2001 [MVS01]. In 2001, the Code Red worm infected 360,000 computers in 14 hours; it contained code to mount a DDoS attack on the White House website. (Fortunately, the DDoS attack was deflected at the last minute).[10] In 2003, an Internet election in Canada was disrupted by a denial-of-service attack on Election Day.[11] These are not isolated examples; it is all too easy to mount DDoS attacks, and the culprits are rarely caught.

If an attacker were to mount a large-scale denial-of-service attack that renders SERVE's voting service unavailable on Election Day, it would call into question the validity of the election and effectively disenfranchise large numbers of UOCAVA voters. Alternatively, network services could be knocked out or degraded for areas where a particular

[8] http://www.nipc.gov/investigations/mafiaboy.htm
[9] http://www.cnn.com/2000/TECH/computing/04/18/hacker.arrest.01/
[10] http://www.symantec.com/avcenter/venc/data/codered.worm.html
[11] http://cbc.ca/stories/2003/01/25/ndp_delay030125

demographic is known to vote for a particular party, possibly modifying the outcome of the election. Detection of a selective disenfranchisement attack might be possible, but it is not clear how to respond; once polls close, there may be no good choices. We expect that denial-of-service attacks could disenfranchise a substantial fraction of the SERVE population, and there seems to be little that SERVE can do to defend against such attacks.

5. Conclusions

Because of space constraints, we have mentioned only a few of the possible attacks. These attacks depend on fundamental vulnerabilities in the current PC architecture (e.g. to malicious code) and in the Internet (e.g. to spoofing and denial of service attacks). They can be launched by anyone in the world, and in many cases may be successful while remaining completely undetected, Consequently, we conclude that Internet voting in general, and SERVE in particular, cannot be made secure for use in real elections for the foreseeable future.

Acknowledgements

We thank Kim Alexander, Dr. Steve Bellovin, Lillie Coney, Prof. David Dill, Prof. Doug Jones, Yoshi Kohno, Prof. Deirdre Mulligan, Prof. Ron Rivest, Prof. Gene Spafford, Dr. Rebecca Mercuri, and Adam Stubblefield for helpful comments.

References

[AK96] Ross Anderson, Markus Kuhn, "Tamper Resistance - a Cautionary Note," *Second Usenix Workshop on Electronic Commerce*, pages 1-11, November, 1996.

[Garman81] John R. Gamran, "The "bug" heard round the world," *ACM Software Engineering notes*, 6(5):3, October, 1981.

[HW01] Kevin J. Houle, George M. Weaver, "Trends in Denial of Service Attack Technology", October 2001.

[JRSW04] David R. Jefferson, Aviel D. Rubin, Barbara Simons, David Wagner, A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE), http://servesecurityreport.org/

[Kohno03] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, Dan S. Wallach, "Analysis of an Electronic Voting System", *IEEE Security and Privacy 2004*.

[MVS01] David Moore, Geoffrey M. Voelker, Stefan Savage, "Inferring Internet Denial-of-Service Activity", *Usenix Security 2001*.

[MPSSSW03] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, Nicholas Weaver, "Inside the Slammer Worm", *IEEE Security and Privacy 2003*.

[Pfleeger03] Charles P. Pfleeger, Shari Lawrence Pfleeger, *Security in Computing*, third

edition, Prentice Hall, 2003.

[SPW02] Stuart Staniford, Vern Paxson, Nicholas Weaver, “How to Own the Internet in Your Spare Time”, *Usenix Security 2002*.