- # Requirements for E2EV Voting Systems When Embedded in or Packaged as an Internet Voting System
- **David Jefferson**
- **Draft 1**
-

The nature of these requirements

- These are requirements for embedding and E2EV system in an Internet voting environment.
- The are over and above the requirements for the core E2EV itself.
- We do not consider usability or accessibility requirements here. Some of these requirements will make accessibility and usability more difficult to achieve. Still, these are *requirements*, and if they cannot be met, or cannot be met simultaneously with usability and accessibility requirement, then we have to recommend not implementing an E2EV Internet voting system.

- **Certification and recertification requirements**
  - An Internet voting system is an *open* system, meaning that large parts of it (the mix of client hardware and software in fact) are unknown, unsecured, un certified, and completely out of control of election officials.
  - But at least if used in federal elections an Internet voting system is also a *national security* system, and thus must be subject the highest security requirements.
  - Any Internet voting system used in a public primary or general election in the U.S. for federal or state legislative, executive, or judicial office, or recall election, or statewide initiative or referendum, must meet all of the requirements in this document.
    - Reduced security requirements might be appropriate for county, municipal, or other kinds of elections
  - *Any E2EV Internet voting system should be certified by competent professionals as meeting*

the requirements in this document, and any reports they issue must be public whether the report recommends certification or not.

- *Any time there is a change in the voting system client or server side or the E2EV system, all of the requirements listed below must be re-established and recertified.*
  - This includes
    - new supported hardware platforms, OS's, browsers, etc.
    - bug fixes and security patches to voting client and/or server
    - changes or upgrades to voting client or server in response to
      - detected bugs or security vulnerabilities
      - changes in law
      - changes in threat environment
- *The requirements listed below must be re-established and recertified every four years even if there are no changes*
  - The relevant threat environment will certainly have changed, or at least must be re-evaluated to see if it has

Reliability requirements

- *The entire voting service (server side) must have a proven MTBF of >168 hours (1 week) under peak expected voting loads the entire time*
  - Must be demonstrated in multiple tests of actual mock elections
  - MTBF requirement applies during normal peak operation, not during attacks, e.g. DDoS
- *If service goes down for any reason other than regional natural disaster or malicious attack, service must be restored in no more than 10 minutes*
  - That is 0.999 uptime
  - Must be demonstrated in by failures in actual mock election situations, e.g. tested by sudden loss of power to any server
  - Should have warm spare in second data center that can take over in case of major failure
  - System must be staffed at all times to guarantee the 10 minute recovery time
- *Ballot casting should be atomic WRT server failures*

- If a voter receives the final “Thank you for voting” confirmation, then she/he can be certain the ballot was recorded
- If a server side failure occurs, no voter’s balloting can be left in an intermediate state.
- The ballot must either be securely and completely cast and the voter marked as having voted, or no ballot is recorded and the voter is considered not to have voted
- If the voter is uncertain, he/she is free to attempt to vote again
- If the system and the law allows a voter to cast multiple votes with only the last one counting, or to cast a partial ballot with the option of modifying it later, then each voting session must be atomic wrt server failures, and if a failure occurs during the voter’s last session, then the votes cast as of his or her previous session will count.

- *Client side software (applications, apps, scripts, etc.) should be free of known bugs on a wide range of platform and software stack combinations intended to be usable as voting terminals*

- *Operators of voting system must document exactly what client configurations are requires or supported*
  - which versions of hardware platforms (PCs, mobile devices, etc.)
  - which versions of which operating systems for those platforms
  - which versions of which browsers, plugins, protocols, or other software applications, apps, components, and plugins
  - which common components, plugins, or other software interfere with voting (e.g. flash blockers, popup blockers, script blockers, etc.)
  - what configuration choices the voter must make to successfully vote (e.g. permit Javascript)
  - what configuration choices the voter might wish to make to more strongly protect his/her vote privacy
    - e.g. disable cookies, run privacy-protecting browser plugins, vote from virtual machine that is later destroyed, log out of social networks, disable remote control and remote administration tools, disable incoming connections, etc.

- *System must be extensively tested on a wide range of platform and software combinations, with test results made public*

General security requirements

Eligibility

Registration

- online registration
- automated registration fraud
- change of credentials

- I don't know yet what to write here regarding requirements. But obviously any automated registration fraud can be used to affect the out come of elections.

Authentication

Authentication of service

- *(Not sure what requirement should be here. The intent is to somehow ascertain that the E2EV software on the client side is genuine. Presumably that E2EV software will authenticate the remote server.)*

Authentication of voter

- *The voting service must by itself securely authenticate the voter* (verify identify the voter and verify his/her registration and/or eligibility according to law to vote in

the election) *before allowing him/her to cast a ballot (or modify or replace a previously cast ballot).*

- Authentication must not be done through third party intermediaries such as Facebook, iCloud, Google, Yahoo, Amazon, etc. that offers authentication services.

- *There must be no way to reasonable way to automate forging or invalidation of a large number of voter authentications*
  - This can only be established by competent security analysis, not testing
- *The security of authentication must not be affected by any potential breach of any public or commercial databases*
  - Authentication must not us personal information, gov't or commercial account identifiers, etc.
- *It should not be possible for an attacker to impersonate voters even if the entire server database used for authentication is compromised.*
- *Authentication secrets must be changeable or revokable at any time at the behest of either the voter of election officials*
  - In some cases of security breach it may be necessary to require all voters in a jurisdiction to re-establish credentials
- *All voter authentication secrets must be changed at least once in every two-year election cycle.*

Availability

- *In a federal election the voting system must remain available even during a large distributed denial of service attack. It must be able to continue correct operation during a sustained DDoS attack on any combination of server side IP addresses (whether at the primary server data center or its ISP) at a total level of 100 Gb/s with no more than 15s degradation of response time to voters during the attack.*
  - The threshold of 100 Gb/s should be evaluated every two year election cycle to see if it has to be raised due to newer DDoS attack technology
  - The ability to survive this level of DDoS attack must be actually demonstrated in the actual network configuration to be used prior to each federal election.
  - Reduced DDoS defense requirements might be acceptable for non-federal elections

Privacy

- *Vote privacy must be preserved end-to-end (insofar as mathematically possible)*
  - vote privacy cannot be waived by voters
  - violations of vote privacy are not generally detectable
  - violations of vote privacy are irreversible
  - violations of vote

- violations of vote privacy enable vote coercion and vote selling
- vote privacy cannot be verified by testing; it can only be ascertained by expert analysis of architecture and code

Client side privacy

Client side malware attacks

Vote privacy must not be violated even in the presence of arbitrary malicious code on the client platform, including phony client software, malicious client wrappers, MITM code between the user and the E2EV interface, malicious browser plugins or scripts, keyloggers, etc.

- This will seriously complicate the user interface an usability of the system, but is absolutely essential.

Client side monitoring systems

Voting should not be permitted from client platforms known to have remote monitoring software installed that could be used to monitor or log voting activity and that cannot be turned off by the voter

- All mobile platforms had, and probably still do have, such remote monitoring software.

Receipt freedom

There must be no way for voters to prove to another party any information regarding how they voted in any race (beyond what is mathematically deducible from the final distribution of votes).

- It is mathematically forced if, for example, all the votes in the precinct are for the same candidate in a race.
- This property cannot be established by testing; it can only be established by expert analysis

Client side effects: communication and memory

No side communication

- *The client software of the voting system must not send data to any IP address except those associated with the voter server and the basic infrastructure servers of the Internet*
  - Basic infrastructure includes routers and DHCP, NTP and DNS servers
  - Client could be leaking authentication credentials, or actual vote information
  - The client should not provide any information to third parties, e.g. Facebook, Twitter, etc. regarding the act of voting;
  - There must be no tracking devices or tracking logic in the vote client
- No persistent traces of votes
  - *The client software must leave no files or other persistent data on the platform regarding the vote transaction: no cookies or other session files, no temporary files.*
    - Users may be advised to turn off browser history data, cookies, logging data, and other tools that might retain a record of the vote transaction whether the vote data itself or metadata.
    - The only exception would be an optional file containing information needed for subsequent verification that the voter's ballot is included in the election canvass
- Transient traces of votes
  - *The client software should explicitly erase (i.e. overwrite) all transient copies of vote-transaction data, e.g. data in registers, caches, RAM, and virtual memory*
    - It should not be possible even for client side forensic tools to retrieve any information regarding the voting transaction after the voting session is ended

Server side privacy

- *Once it is determined that a ballot will be counted, the ballot should be irrevocably separated from the identification of the voter who cast it. It must be*

information theoretically impossible to determine who cast a particular ballot, or what ballot was cast by a particular voter.

- Again, except for what can be deduced from the distribution of votes in a precinct.

- *If the voting system permits voters to modify or replace their previously cast ballots, the system should retain only the latest votes in each race, and retain no record of the previous modified votes.*

Integrity

Client remote control and remote administration

The voting system should not support platforms that have remote administration or remote control tools installed that cannot be turned off by the voter

- All mobile platforms have had,and may still have, such systems in place.

Client side malware

The voting system must not be vulnerable to malware designed to modify votes before they are input to the E2EV system.

- This is not something that can be tested — it has to be determined analytically by experts
- This will seriously complicate the human interface and usability of the voting system, but is absolutely essential
- Malware can be in many forms: completely phony or “alternative” client app, client wrapper, client side MITM, browser plugin, client APT, etc.

- Authentication of client
  - *The voting system server must authenticate that it is communicating with a genuine vote client during a voting session*
    - This will complicate, but not eliminate, the possibility of client side malware
  - Server side integrity attacks
    - Penetration and APT attacks
      - (I don't know what to write about this.)
    - Insider attacks
      - (I don't know what to write about this.)

Security requirements specific to elections

Double vote prevention

- *The voting system should not record more than one vote for any voter in any race*
  - This is not meant to exclude systems that allow the voter to vote multiple time with only the last one counting, or that allow the voter to modify his or her vote up to the end of the election window

Advertising / online electioneering

- *The voting system client must not display or permit displaying of any advertising or commercial logos in the window that contains the voting session, other than that for the election jurisdiction itself.*

- *The voting system client must not display any links to other sites in the window except for help in the mechanics of voting.*

Coercion prevention

- *There must be no way for voters to prove to another party any information regarding how they voted in any race (beyond what is mathematically deducible from the final distribution of votes).*
  - It is mathematically forced if, for example, all the votes in the precinct are for the same candidate in a race.
  - This property cannot be established by testing; it can only be established by expert analysis
  - This is a repeat of a requirement under *privacy* above
- *Voting should not be permitted from client platforms known to have remote monitoring software installed that could be used to monitor or log voting activity and that cannot be turned off by the voter*
  - All mobile platforms had, and probably still do have, such remote monitoring software.
  - This is a repeat of a requirement under *privacy* above

Vote buying / selling prevention

- Vote buying is a growing threat as anonymous crypto currencies spread
- *There must be no way for voters to prove to another party any information regarding how they voted in any race (beyond what is mathematically deducible from the final distribution of votes).*
  - It is mathematically forced if, for example, all the votes in the precinct are for the same candidate in a race.
  - This property cannot be established by testing; it can only be established by expert analysis
  - This is a repeat of a requirement under *privacy* above

Verifiability / auditability

- *The list of voters who voted online should be published.*
  - The point of this is to detect possible ballot box stuffing in case votes are somehow cast in the name of legitimate voters who did not in fact vote.
- What to do if verification fails
  - *There must be clear technical and legal procedures for how to proceed in the event that voters can prove that their votes were not received accurately or counted, or if the official election verification*

application does not verify that the Internet part of the election was correct.

- *The E2EV core system, in the event that it does not verify that the online votes cast, must be capable of giving a close upper bound on the number of ballots that may have been affected.*
  - This will guide officials of courts in their decision regarding what to do.
- *Official verification applications, like voting software, must be published in source form, along with documentation, build directions, and a standard cryptographic hash of the source code.*

Transparency requirements

Open documentation

- *All aspects of the design, architecture, algorithms and documentation for the entire Internet voting system (not just the E2EV core) should be published and available for free download by anyone.*
- *As the system changes, all documentation must be kept up to date. No new version of an E2EV Internet voting system may be certified until all documentation is up to date.*

- *No nondisclosure agreement or any other contract shall be required to download and study the Internet voting system.*

Open source

- *The source code, build scripts, issue tracking system, security features, and related development information for the entire Internet voting system (all versions for all platforms) shall be made publicly available for free download and inspection by anyone.*
- *In particular the source code for all parts of the E2EV Internet voting system shall be made publicly available under a license that allows anyone who wishes dto ownload the code, build it, instrument it, and test.*
- *No nondisclosure agreement or any other contract shall be required to download, instrument, build, test, and publish test results for an E2EV Internet voting system.*

Verification application

- *Official verification applications, like voting software, must be published in source form, along with documentation, build directions, and a standard cryptographic hash of the source code.*

Logging & forensic data

- *The Internet voting system should keep detailed logs of all relevant activity, written to write-once media in append-only mode.*
- *The log data should be as complete as possible, consistent with maximum possible vote privacy.*
  - If there is a tradeoff between vote privacy and the identification of the perpetrators of fraud, the decision should be made in favor of vote privacy.
- *The log data and documentation of its meaning and format should be available for public download so that anyone can download, inspect, and publish concerns based on the logs.*