METHOD OF INTERNET-BASED VOTING USING DOMAIN NAME SYSTEM

Background

Voting machine technology has faced a lot of challenges in securing the ballot, protecting the tally count, while providing audit trails of those who voted[1].

Summary

This solution is to build the application on a secure DNS infrastructure to leverage the latter's multi-tiered environment with modifications that leverages digital signatures and ballot tokenization (for auditing purposes) to secure the ballot / tally. The method and system described herein relates to a method of internet-based voting using Domain Name System (DNS) to tally candidates votes from a ballot. The system incorporates entities that is found on the DNS infrastructure: Stub, Recursive and Authoritative resolvers – and records the votes across all these hosts.

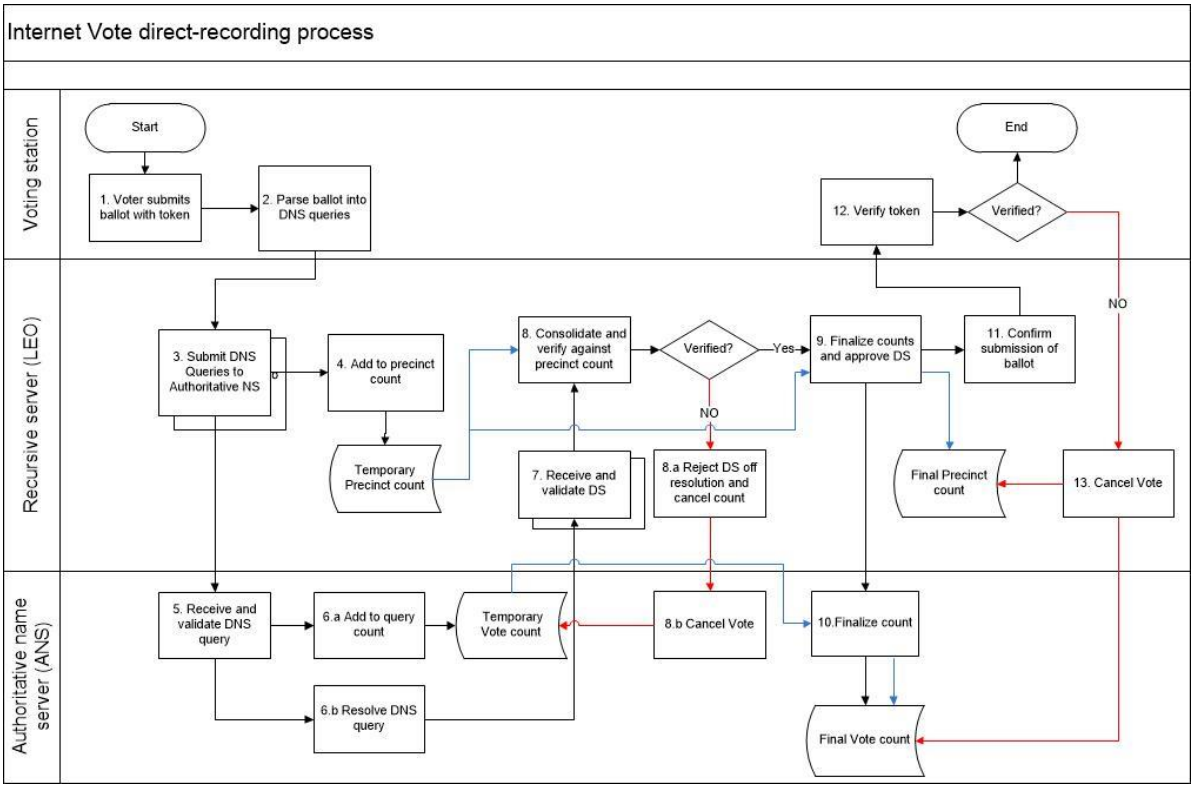
The stub resolver is a ballot application running on a computer. It is responsible for recording the votes and parsing the results into DNS queries. The recursive and authoritative DNS servers preserve their roles in recording the request and resolving the query with the appropriate response.

This approach – running on its own virtual private network (VPN) – elevates the architecture from two-tiered (Client-Server) to a multi-tiered (Stub-Recursive-Authoritative) environments; making it more robust in protecting and securing the tally / count.

Detailed Description

The embodiments described herein relate to a private DNS infrastructure designed to handle on-line voting. It comprises the following:

[1] https://www.blackhat.com/us-18/briefings/schedule/#lessons-from-virginia---a-comparative-forensic-analysis-of-winvote-voting-machines-11107



The Ballot

1. The suggested approach of this proposal is to have a software application that displays a blank ballot:
    a. The application allows for the voter to select his candidates
    b. The application allows for the inclusion of a digital token for risk-auditing purposes
    c. The application allows for full submission of the ballot.
2. As stated above, embodiments of the present application involve parsing the individual candidate vote into DNS queries. Embodiments would include a source IP/station address as Recursive Servers would relate the Voting Stations as stub resolvers. Once the ballot is submitted (and the individual votes parsed into DNS queries), two-tiered servers would come into play.
    a. Another embodiment includes a digital token (and other forms) that identifies and authenticates the existence of a real voter who completed the ballot. The digital token is bound to the individual DNS queries.
    b. All ballots and parsed DNS query records are stored on the Voting station's local DB for auditing purposes.

Two-Tiered servers (Recursive and Authoritative)

3. The recursive server would receive the DNS queries. Embodiments to this process includes the following:
    a. Verification of authenticated stub resolver as well as authentication of vote validity[2]. If the queries are not verified / validated, the entire set of queries may be rejected.
    b. Verification of the ballot's digital token on each of the DNS query.
4. The DNS requests would also be added to a temporary "Precinct" count. The idea is to provide a local count of votes for back-up/replication purposes. This architecture is designed to be analogous to a Local Election Office (LEO). Additional variation may validate the total count of ballots per LEO/Precinct.
5. DNS queries would be received by the Authoritative Name Server (ANS). Embodiments to this process include whitelisting (Access Control Lists [ACL] to determine which Recursive servers can send DNS queries) to ensure security of the relationship between LEO and the ANS and is feasible for supervised voting[3]. With whitelisting (as well as possible embodiments to validate the DNS queries, anew), ANS can validate the source of the queries prior to further processing.
6. Upon receipt / validation, the ANS does the following:
    a. Add the query to the Vote associated to the candidate on a temporary vote count.
    b. Storing the digital token as part of the record – for auditing purposes.
    c. Resolving the query with an associated IP[4] address. Addition to resolution is the incorporation of a Digital Signature (DS) via DNS Security Extensions (DNSSEC)[5].
7. Recursive will use the resolution as well as DS to validate the vote count:
    a. If DS does not match – as in the case of a (unauthorized) SQL injection - the resolutions would be rejected. Variation of this process may include ANS resending the packets, if/when appropriate.
    b. Attached digital token is also validated. If it does not match stored digital token, it is rejected as well – as noted on the next step.
8. Recursive servers would consolidate all DNS resolutions against the precinct count/ballot:
    a. If rejected, the recursive server would send a cancellation of vote count to the ANS.

[2] For example, only one vote for President, one vote for Vice-President, a limit on the number of votes for the board, etc. This has to be defined before the voting takes place.

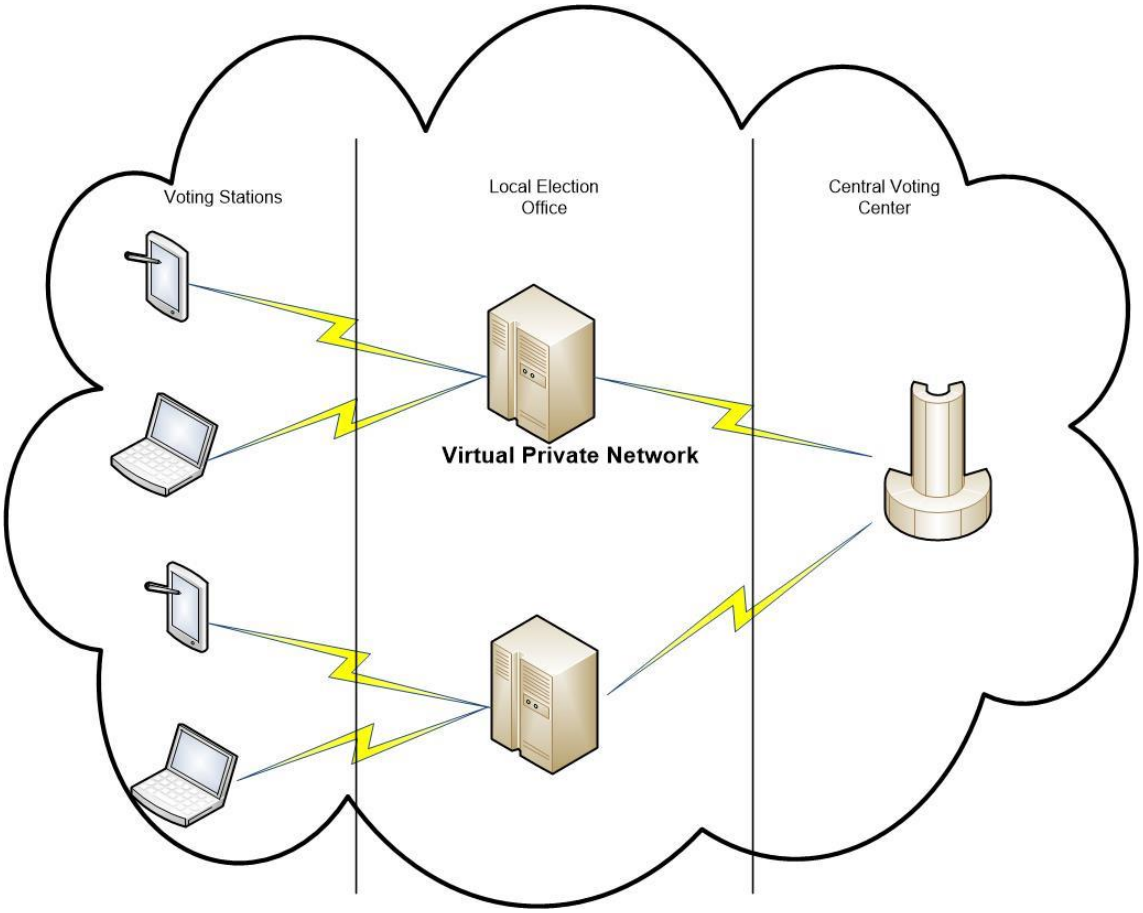
[3] Andreu Riera et al., Internet Voting: Embracing Technology in Electoral Processes, in ELECTRONIC GOVERNMENT: DESIGN, APPLICATIONS AND MANAGEMENT 80 (Åke Grönlund ed., 2002).

[4] IPv6 would be preferred version.

[5] RFC 4641

   b. Authoritative Name Server (or ANS) would cancel the appropriate counts against its temporary vote count storage.
9. If verified, recursive server will finalize its Precinct count and approve the DS.
10. ANS would take the approved DS to finalize its vote count by finalizing the temporary vote count.
11. Recursive server would confirm the submission of the entire ballot.
12. Voting station verifies the completion of the process by validating the digital token.
    a. Another embodiment, prior to ending this process, is the ability to print out the entire ballot as it is recorded on the LEO and ANS.
13. If the verification fails, recursive server will send a cancellation of votes on its database as well as the recorded vote on the ANS.

Architectural diagram of the system



The architecture that supports this system is a private Domain Name System. Its structure is based on trusted connections amongst servers (Authoritative and Recursive) as well as Voting stations where the application resides.

The Central Voting Center is the Authoritative Name Server (ANS).

1. It is configured to have a top-level domain (TLD) that is not resolvable by public internet (DNS) queries. To accomplish this, the TLD is not registered by ICANN or any other Internet authority.
2. Furthermore, connects and (only) accepts DNS queries from trusted Recursive servers. The trust relationship is established by network-network interface (NNI).
3. When the ANS resolves a DNS query, it does not provide a TTL (time-to-live) value. In other words, no caching. This ensures that each DNS query is recorded and resolved.

The Local Election Office is represented by the recursive server.

1. The recursive server connects and (only) accepts DNS queries from trusted voting stations. The connectivity can be established with VPN, IP whitelisting and other methods of securing the connection.
2. The recursive server will synchronize with the authoritative server through on private IP interconnects. These domains are only resolvable by network equipment; not by end-users. (They will exclusively use network-network interface; not user-network interface.)
3. In handling DNS queries:
    a. Recursive servers may keep a total number of voters in their 'sphere of influence'. This number can be used for risk-based auditing.
    b. Recursive servers may set rate-limits to catch possible fraudulent votes.
    c. Recursive servers will not keep a cache of DNS records (Zero value on TTL) to ensure that each query / resolution is recorded accordingly.

Both Authoritative and Recursive name servers will employ digital signatures as applied through DNS Security Extensions (DNSSEC). This authenticates the delivery of the query / response between the servers.

The Voting station (or Polling station) is represented by a computing device; also referred to as the stub resolver:

1. The device may be any internet-capable computer (including laptops, tablets, etc) that can host the ballot application.
2. The device should have the capability to establish a secure connection with the recursive server; be it, VPN or other methods.
3. The device should have the capability to:
    a. Parse selected votes into DNS queries
    b. Accept a digital token that can be in any form. This token can be stored on the TXT section of the DNS query.
    c. Record the votes on its local drive for verification and risk-based auditing purposes.