FACULTY OF ENGINEERING
AND ARCHITECTURE

WEB ELECTION SYSTEM

Information Security 2018-2019

Xavier Claerhoudt
Jens-Joris Decorte
Julien Marbaix
Michiel Van Gendt
Maarten Vangeneugden

GHENT
UNIVERSITY

Web election system

Project for Information Security

Xavier Claerhoudt
Jens-Joris Decorte
Julien Marbaix
Michiel Van Gendt
Maarten Vangeneugden

2018-2019

Contents

1 Security requirements

1.1 Authentication

It's important that we check who the client is.

1.2 The anonimity-authentication conundrum

According to the Belgian constitution, the election must be secret, that is, a voter and the vote mustn't be linked to each other.

On paper ballots, this is not a problem; a voter can show up with an identity card, after which the person fills in a ballot, and drops it in a box, without anyone looking at it. This removes a possible trust issue; the voter knows that nobody got to see the ballot.

This is a problem that conflicts with the required authentication: The web booth requires voters to provide an identity card to prove they are who they are. The possibilities offered by a physical presence are gone.
This is less of a problem if we assume the government is a trustworthy party, and will honor the privacy of the voters. A large part of this analysis rests on that viewpoint (i.e. the government is trustworthy).

Using Tor is no good either: While that system allows for surfing anonymously, it does not guard against (mandatory) identification[1], which is required for the integrity of the vote[1].

[1] Every voter must only cast one vote. If the possibility exists that one person can vote multiple times, the integrity of the vote disappears.

2 Security (counter)measures

This section describes feasible attacks[2] that an online election will have to defend itself against.

2.1 The government is trustworthy

All the following security measures critically depend on the assumption that the government is a trustworthy party; that implies that we assume the informaticians responsible for the infrastructure are also trustworthy, the data does not get tampered with by anyone with access to the voting system, ...

2.2 Authentication

The system will request that the voter provides the eID card, to check if that person is eligible to vote.
This also serves as a way to block bots from entering the website; no identity card means no access.

2.3 Preserving anonimity

Part of maintaining **confidentiality** comes from requiring all connections use HTTPS, to avoid sniffing attacks. This also makes spoofing harder to pull off.

When it comes to the registration of the vote, the backend server will feature two different database tables: One stores what people have already voted, and another stores the votes themselves. Both tables are not linked to each other in any way.

2.4 Integrity

2.4.1 HTTPS enforcement

To make sure that the voter's connection can't be intercepted, we can contact popular browser distributors, and ask if they can enable HSTS preloading for our website. Of course, HSTS is also enforced from the server side.

[2] This means that we won't be discussing attacks if the chance they'd actually take place is negligible.

2.4.2 Voter bribery / Vote buying

Numerous stories exist in historical references about how (mostly poor) people would "sell" their vote, essentially casting a ballot with the briber's vote, in the voter's name.
The ability to sell your vote gives poor people a strong incentive to not vote in their best interest, but rather let the rich decide what's "best" for the country, tilting every election in favour of a plutocracy.

the Belgian prohibition on identifying ballot marks[4] effectively stops this practice, but in the web booth we lack physical assurances. As a countermeasure, our system offers the ability to generate a "bogus token", that allows the user to make a token that displays a vote for anybody the person wants. The mere knowledge that bribing somebody doesn't guarantee that the bribee actually voted as agreed, might very well be enough reason to deter bribers.

2.5 Non-repudiation

After the voter casts the ballot, the website will present a token (a QR code) that, when scanned, displays for whom the vote was cast.

2.6 DDoS attacks

Maintaining **availability** is required to offer people the chance to vote.
To avoid an "accidental" DDoS, the invitation letter provides a designated, non-compulsory time slot, copying the technique currently used for Belgian elections.

Malicious DDoS attacks are possible. We propose the following measures to guard against this:

Designate 12 different voting servers One for each province, plus Brussels Capital Region. The remaining server remains stand-by, and takes over when one of the other servers can't take the load anymore.

Check for possible DDoS attacks using a secondary service Services like Cloudflare can be used to detect incoming attacks, allowing legitimate voters to cast their ballot with few problems.

3 Remaining vulnerabilities

With web voting, we are not able to counter all possible attacks. The most prominent vulnerabilities are discussed in this section.

3.1 Malicious SSL issuer

While we make HTTPS mandatory, we're putting trust in SSL providers that they will not abuse the power we give them. We could hand out our own certificates, but that would either throw issuer invalidity errors, or we'd have to assume every household has the technical knowledge to check a certificate for authenticity. We're already assuming they know their systems are secure enough, and that's a high bar to pass, let alone being able to handle digital signatures.

3.2 Malicious voting coercion

3.2.1 Voter intimidation

To make sure people can vote the way they want to vote, it's imperative that all possible ways to intimidate voters are diminished.
The Belgian approach to this, is to forbid "identifying marks" on the (paper) ballot, so a person cannot be forced to mark their ballot for somebody else to check. And of course, only the voter is allowed inside the voting booth.[3]

This is impossible to pull off with a web voting booth; an abusive spouse can always be watching, one might feel a form of peer pressure when friends are around, or an employer might force employees to vote the "right" way. There is no way we can be certain voters aren't intimidated.

3.3 Client malware

We must presume that the voter uses a device that is reasonably secure. If the client is compromised, certificates might be altered and the voter would be none the wiser.
A website cannot prevent this. A compromised computer is able to undo every security mechanism we put in place to secure the vote of the affected voter.

[3] If a Belgian voter requires help with the voting process, that person can ask the local election official for assistance inside the voting booth. This exists to accomodate for e.g. visually impaired voters. These exceptions are always recorded in the procès-verbal.

4 Specific security details

The details of the decisions that were made for the different elements of this project are described in this section.

4.1 HTTPS settings

Certificate is an 2048 bits RSA key, with enabled forward secrecy. The website will offer support for different cipher suites to accomodate for browsers without the latest patches. It will however, provide a list of "preferred ciphers", starting with $\text{TLS}_{\text{ECDHERSAWITHAES256GCMSHA384}}$. This implies support for SSL is disabled, only TLS 1.2 and 1.3 will be available.

To avoid malicious certificate issuing, we utilize Certificate Authority Authorization to limit which CA can create a certificate for the website.

Forward secrecy must be enabled as well .

4.2 Software

We decided to use Django (v2.1) as the framework on which we will build the web voting booth. The main reasons for our choice are:

Security as primary feature This would be a very important part of our project, so a framework that proactively deals with that is a big plus.

Python Easy language to work in, few pitfalls, and understood by the entire team.

Out-of-the-box ready Creating the web booth should happen as quickly as possible. Having all the necessary "web plumbing" already done is a big boost to our productivity.

4.3 eID authentication

The Belgian government provides a library that allows developers to work with eID cards. This system is implemented in the website, and asks for the voter's identity card, which can be provided using a card reader. When the identify is confirmed, the voter is granted access to the voting ballot. [4]

[4] In the interests of full disclosure: We were not able to implement the PyKCS module in the prototype, given the timespan for this project. As a substitute, the website will ask for the The prototype does act like it checks for an identity card, but the actual authentication does not happen.

5 Conclusion

What we've described in this report, is an attempt to make sure that an election could happen over the internet, while also guaranteeing the same integrity that paper voting offers.

However: The amount of assumptions that we had to make in order to just begin this project, is an affront to a democratic election process; assuming everyone that has access to critical parts of the system is trustworthy, is essentially placing the integrity of an election in just a couple of hands.

Our conclusion is thus: **Given the current state of informatics, it is not possible to build a secure web based election system.** While it may become feasible in the future.

Assuming that the authorities can be trusted with all this is almost begging for committing fraud.

This report should be taken as a summary of why internet voting is a terrible idea. These findings are in line with those of prominent informaticians like Dr.h.c.mult. Richard Stallman[3], Dr.h.c. Bruce Schneier[2], who've already called attention to the issue, and they too, recommend using paper voting for securing elections.

6 Bibo

References

[1] The Tor Project. So I'm totally anonymous if I use Tor? URL: `https://2019.www.torproject.org/docs/faq.html.en#AmITotallyAnonymous`.

[2] Bruce Schneier. "Securing elections". In: (). URL: `https://www.schneier.com/blog/archives/2018/04/securing_electi_1.html`.

[3] Richard Stallman. "Computerized voting machines". In: (). URL: `https://stallman.org/evoting.html`.

[4] Bart Verhulst. "Hoe moet u stemmen? Wat moet, wat mag, wat mag niet?" In: (). URL: `https://www.vrt.be/vrtnws/nl/drafts/Politiek/hoe-moet-u-stemmen-wat-moet-wat-mag-wat-mag-niet/`.