

Analysis of LA County's ISB Program

Murali Mohan, Matthew Bernath, Blake King

1. Background to Internet Voting

In recent years, there has been increasing interest toward developing internet voting systems. Many governments around the world, including the United States, France, and Canada, have allocated funds to the research and development of internet-connected election systems [1]. While governments and developers claim their systems are secure, studies done by researchers have found a multitude of flaws which jeopardize voter privacy and vote integrity. An example of such a study was in 2010, when the municipality of Washington DC held a mock election with an internet voting system which was compromised within days of its deployment [2]. Attackers were able to gain access to information about who cast almost all of the votes, and had access to change any vote of their choosing.

The ultimate goal of shifting towards internet systems is well-intentioned. Election officials desire to increase the availability and accessibility of voting for all citizens. The internet could reduce backups at voting centers, help disabled citizens cast their votes easier, and potentially increase the speed at which votes can be tallied. The advantages of the internet are numerous, but they come at a major cost to the security of an election. An overwhelming majority of the security community agrees that online voting should not be deployed in real elections until significant security improvements can be made. The abject failure of previous systems, however, has not stopped municipalities from attempting to increase accessibility of their existing voting systems through the use of the Internet.

2. ISB Introduction

LA County, the nation's most populous county, has spent the last decade developing technology to combat long voting lines at the polls [3]. Their most recent development, the "Los Angeles County Interactive Sample Ballot" (ISB), aims to decrease the time a voter spends at the vote center by allowing them to make their ballot selections at home. The ISB was first deployed in the Municipal and Special Elections held in March 2020, and most recently used in the 2020 Presidential election.

The ISB is a tool that allows users to go to a website and fill out their registration information/address to get a sample ballot associated with their precinct. They can then fill out their selections on the sample ballot website and print a QR code called a "Poll Pass". The voter can then scan their Poll Pass at any vote center within LA County to load their selections onto a ballot-marking device (BMD). After verifying their selections, the BMD then prints out a ballot containing their selections for tabulation.

While it has been demonstrated numerous times that internet voting systems are not secure and should not be used in vital elections [4, 5], the ISB differs in that no votes are cast or tabulated over the internet. It is simply a system which allows voters to pre-select candidates and carry their selections to a voting center. In spite of this, connecting any part of the voting process to the internet introduces many of the same vulnerabilities of existing internet voting systems. The ISB has drawn little national attention and has yet to be publicly reviewed, though a private audit was conducted by the state of California [6]. The purpose of this paper is to conduct a system security review of the ISB and its impact on the security of the election system.

3. Methods

Constraints

The first step for conducting our system review of the ISB was to decide on a set of constraints in order to conform to United States law and ethical boundaries. Due to the proximity of our investigation to the November 2020 general election, we were worried about appearing as a malicious attacker to election officials. This would cause undue stress to those election officials and possibly divert critical resources away from analyzing other real potential threats. Therefore, we chose to create our own duplicate application, limiting our interaction with the actual ISB application as much as possible. Due to these constraints, we did not explore any security vulnerabilities related to LA County's backend servers or databases. The scope of this review is therefore limited to issues found in the publicly available client-side code and the role the ISB plays in the election system as a whole.

Getting Data

In order to create our own deployment of the application that we *could* interact with, we needed to replicate the system as closely as possible. This included getting a copy of the entire client-side application source, as well as a copy of the HTTP request/response data between a voter and the ISB system. This way, we could use the captured response data from the backend to replicate the backend response to our frontend interactions.

In order to get the client-side JavaScript code, we were able to just download the minified code directly from the browser. Additionally, the site also had the unminified React source code of the application available in the "sources" section of the Chrome Developer Tools, so we were able to download a copy of that for analysis (instead of having to reverse engineer the minified bundle). This is good from a transparency perspective, as it allows anyone in theory to perform an audit of the site (as we have).

To get a copy of actual API responses from the production application, we worked with a consenting LA County resident to use the application as intended. We observed the resident fill out a sample ballot and generate a Poll Pass. After they finished using the application, we were able to use the Google Chrome Network tools to save a HTTP archive (.HAR file) of all requests and responses recorded by the browser during the interaction.

We retrieved the code from the ISB website on October 19, 2020. The HAR file was also captured on the same day. This was the version of the code that was used in our analysis.

Replicating the Website

Once we scraped the website content of the ISB website, those files were then hosted on an Apache server without modification, as the website source code used predominantly relative references to local resources. Due to our constraints, we did not attempt to replicate the lavote.net's web server configuration and so we left that mostly out of our analysis as well.

4. System

The ISB system consists mainly of three components: the ISB website, the QR code Poll Pass generated by the ISB, and ballot marking devices (BMDs) at LA County vote centers. The ISB website is predominantly written in JavaScript (generated from React), CSS, and HTML.

The main landing page for the website outlines the ISB process for voters (see Fig. 1). It has options to set the language of the website, supporting 13 different languages. For voters with visual impairments, there are also settings for high contrast mode and enlarged text.

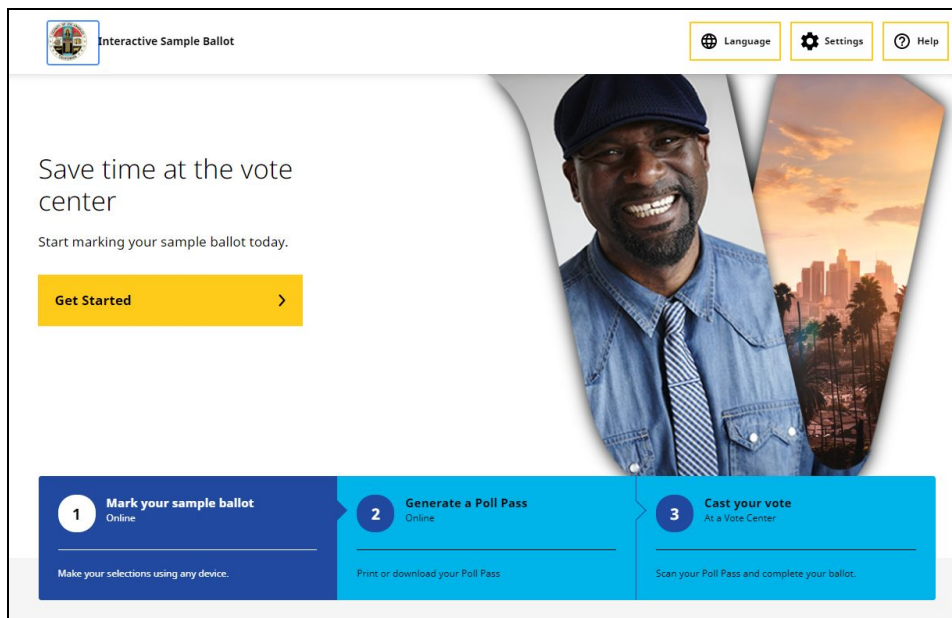


Fig. 1: **isb.lavote.net Landing Page.** This is what a voter is presented with when visiting the ISB website. When they click “Get Started” they are sent to the “voter lookup” page.

To use the website, a voter needs to perform a “voter lookup” or “address lookup” in order for the website to return information about the races in which they are eligible to vote. If the user elects to have their ballot retrieved via a voter lookup, they provide some voter registration information (last name, birth date, and house number). An HTTP GET request is then generated with query parameters including the election ID (an identifier for the current elections) and the voter-provided information. The response from the ISB server will be data encoded in JSON, which contains the description of the voter’s ballot.

If the user instead elects to use address lookup, the process is slightly different. When a voter enters their address, GET requests are generated for portions of the address. The requests will be up to three characters of the last separated portion of an address. For example, if the address “123 Main St” is submitted, a JSON file containing every valid address on a street starting with the letters “MAI” would be sent to the client. The user’s browser could determine which precinct they were in based on that information and generates another request for the ballot information for the voter.

Before a voter can make their selections, they are prompted with a message “Do you want to store your selections for later use?” There are two options given to a user (see Fig. 2). The first option, “Yes,” comes with a warning that a user’s data will be stored locally and can be retrieved later. This option will store a voter’s selections in their browser’s local storage. This storage will persist after a user closes their browser and even survive a reboot of their device. The second option, “No,” only temporarily stores their selections in the session storage for that browser tab. The website advises this option for voters using shared devices.

Your privacy is important

Do you want to store your selections for later use?

Answer YES to allow your web browser to store your selections.

Recommended only for private devices. Your selections will be stored on this device, so you can retrieve them at a later time. Your selections will not be shared.

Yes, only I use this device

Answer NO to prevent your web browser from storing your selections.

Recommended for public or shared devices. Your selections will not be stored on this device. Your selections will not be shared.

✓ No, others use this device

Fig. 2: **Privacy Options of ISB.** The ISB displays clear options to the user and bolds the implications of each choice.


After the voter makes their privacy selection and their respective ballot is retrieved. The voter is then prompted to select their choices for each entry on the ballot one screen at a time (see Fig. 3). If a voter wants to have a write-in candidate for a race, they can mark their selection as a write-in but are unable to provide the actual name of the candidate they want. They will have to provide their write-in name at their vote center. Once the voter reaches the end of the process, they are sent to a final review page where all of their selections can be checked and edited if necessary.

STATE MEASURE 14

Vote YES or NO


AUTHORIZES BONDS CONTINUING STEM CELL RESEARCH. INITIATIVE STATUTE. Authorizes \$5.5 billion state bonds for: stem cell and other medical research, including training; research facility construction; administrative costs. Dedicates \$1.5 billion to brain-related diseases. Appropriates General Fund moneys for repayment. Expands related programs. Fiscal Impact: Increased state costs to repay bonds estimated at about \$260 million per year over the next roughly 30 years.

YES on Measure 14



Info

NO on Measure 14



Info

Fig. 3: **Ballot Selection Process of ISB.** Ballot races and initiatives are presented to the voter one page at a time. Each screen gives a brief overview of what is being voted for and allows the user to get more information on each option.

Once their selections are confirmed, the voter is given the options to print, download, or enlarge their generated “Poll Pass”. This Poll Pass is a QR code that contains their voting selections for the current election (see Fig. 4). The print option generates a PDF that can be printed. The download option creates a JPG containing the QR code. The enlarge option renders a larger version of the QR code, which a user can take a picture of with their cell phone.

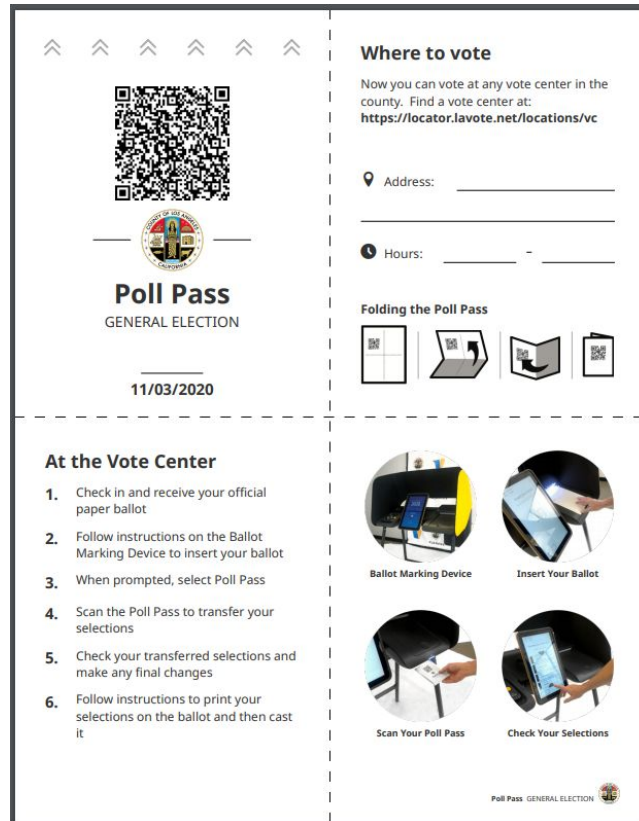


Fig. 4: **Poll Pass Generated by ISB.** This is a mock Poll Pass generated by the ISB website. It provides the voter with a QR code of their vote selections, and additional instructions arriving at the polling location.

The pass is to be taken to any vote center in LA County to be scanned on a BMD. Once scanned, a voter has the opportunity to review their selections and enter any write-ins on the BMD itself. Their selections are then printed, and the printout is turned in as their official ballot to be tabulated on a different machine.

5. Evaluation

There are a few characteristics expected from any voting system to be used in an election: voter privacy (ensuring a voter's vote cannot be revealed), voter integrity (ensuring a voter's vote is counted as cast), coercion resistance (ensuring that a third party cannot verify that a voter voted a certain way), availability (ensuring all eligible voters have the opportunity to vote), and usability (ensuring eligible voters are able to cast a ballot properly regardless of disabilities). It is often helpful to keep these characteristics in mind when evaluating the security of an election system because attacks often affect one or more of them.

Due to the lasting impact an election can have on a nation, election systems are often seen as a high value target for potential attackers. When conducting a national election, potential attackers can range anywhere from individuals with malicious intent all the way to nation-states with very large resources and an agenda against another nation. The goal of an attack on the election system is often to either change the result, or diminish the trust in the results of the process, and this can be achieved by exploiting one of the key characteristics of an election system mentioned above.

When conducting our evaluation of the ISB, we first looked to see if the ISB had the same faults as other electronic voting systems that have been previously demonstrated to be woefully insecure. As a result of the system's client-side design, no votes are transmitted over the internet, which severely limits the types and impacts of any potential attacks.

Threat Model

When evaluating the ISB, we considered both the ISB system itself, and how adding the ISB to the election system affects the security of the rest of the election process. Our threat model considered several main categories of attacks: phishing, malware on the client's device, network access (either a MITM or some other way to compromise HTTPS traffic), control over third-party dependencies, and non-technical. The attacks range in level of sophistication, but as previously mentioned, potential attackers may have a wide range of capabilities.

Phishing Attacks

When hosting any website, a phishing attack is a potential attack vector outside of the developers' control. Due to the client-side source code being publicly available in plain text, replicating and hosting a duplicate version of the ISB website is a trivial task. After replication, an attacker would simply have to convince a victim to visit their site instead of LA County's in order to be successful.

Once a voter is phished, an attacker could modify the JavaScript code served on the site to view all of their selections, omit races in the election, and even produce a modified ballot QR code. Another subtle but possible attack is to reorder the candidates within individual races - research shows that the ballot order can give a roughly 5% increase in likelihood of winning for candidates in lower-profile races [7].

While phishing attacks are largely out of the control of the developers, LA County made several poor choices which makes them even more susceptible to this type of attack. First, LA County hosts the ISB website on "https://isb.lavote.net/", a ".net" domain instead of utilizing their access to a more trusted ".gov" domain. We conducted a search of similar domain names and found names such as "lavotes.net", "votela.net", and "lavote.ca" available for purchase.

Another issue is the lack of DMARC, DKIM, and SPF records for subdomains of lavote.net. The lack of these records would enable attackers to spoof emails as being from lavote.net subdomains [8], which in turn could be a vector for getting phishing links to real voters from what appears to be the genuine ISB website.

Malware on Client Device

Due to the large number of potential users of the ISB site, it is very likely that some voters' computers will be compromised at some level, possibly leading to client-side attacks.

If an attacker has access to a compromised device, they could again perform attacks such as viewing the voter's selections, omitting races in the election, reordering candidates, and replacing the voter's QR code with one of their choosing. Another possible attack could be to deny a voter access to their ballot by producing an error when they enter their information. While a voter is not denied the access to vote at the poll, it would add frustration for the voter during the process. This attack has mainly the same results as phishing, but voters who visit the legitimate ISB website are now vulnerable as well.

Network Access

Much like any website, the ISB is vulnerable to network-based attacks. There are two important pieces of information exchanged between a voter's browser and the ISB website: voter personal identifying information sent from the browser to the website, and a voter's ballot information returned from the website to the client. It is important to also reiterate that a filled-out ballot is never sent back to the ISB website. This design choice by the developers limits many of the most severe network attacks, but several potential attacks are still plausible.

If an attacker had the means to intercept, decode, or inject malicious data into a voter's network stream, they could take advantage of this information. Although this would require a somewhat sophisticated attack (breaking HTTPS encryption), the feasibility of this is increased by the fact the ISB website supports ciphers (see Appendix B) marked as weak by Qualys SSL Labs [9].

If an attacker could intercept and decode the voter's traffic to the website, they would have access to the voter's personal identifying information (specifically last name, DOB, and house number). An attacker could see what the elections and ballot measures a voter is eligible to vote on, but this information is also publicly available. Since voter ballot selections are never sent over the network, a network-based attack would not be able to determine this information and compromise voter privacy or integrity in this way.

If an attacker is able to modify the data received by the client from the website, they could give the voter a faulty ballot by modifying or changing the returned JSON file from the ISB website. When a voter makes selections on a fraudulent sample ballot, the ballot could potentially be rejected at the vote center once their Poll Pass was scanned, forcing them to vote manually. The attacker could also reorder or hide the candidates on a ballot, potentially influencing the outcome while producing valid Poll Pass. An attacker could return results to a voter's browser indicating their voter information could not be found, denying them access to the ISB system entirely, and forcing them to fill out their ballot at the vote center. Finally, an attacker could inject JavaScript code into the existing script of the website, allowing an attacker to exfiltrate a voter's selections directly from the browser.

Control Over Third-Party Dependencies

During our review of the ISB source code, we found a total of 23 external libraries used for various purposes such as generating the QR code and generating the PDF the QR code is saved on (see Appendix A). These dependencies were hosted locally, in lieu of making external requests to resources outside the control of the developers. There is also one API request to Google Maps used in the address lookup to display the user's entered address back to them on a map. While it is a positive that these dependencies are hosted locally, this still opens a wide attack vector for injecting malicious code into the ISB website. If an attacker were able to compromise any one of these dependencies, they could potentially perform many or all of the previously mentioned attacks. It is unknown whether the ISB developers took security measures and audited these libraries before including them.

Non-Technical

Along with the technical vulnerabilities, the ISB also introduces a few non-technical attacks into the election system as a whole. These attacks include slate voting, coercion, leaking selections to others on a shared computer, and leaking votes to anyone who can see the QR code.

Slate voting is when a political party or other group sends a voter a pre-filled ballot with candidates selected to align with their views. Using the ISB, a political party or candidate could mass distribute prefilled QR codes as their slate. With a traditional paper slate, a voter would at least need to

read the ballot and fill out the corresponding candidates individually, providing a forced review of the slate. With the QR form of slate voting introduced by the ISB, voters can instantly fill out their ballot without ever having to look at it, omitting any review. Used maliciously, an attacker could pose as a group with certain interests and provide a slate. This malicious slate could potentially differ from what the voter expects, especially on lower-turnout downballot races in which candidates are not well known.

The ISB can also add potential for coercion at the vote center. A voter at the poll booth should take little to no time completing their vote, since scanning a QR code is much faster than manually voting for individual candidates. If a coercer was able to watch how long it took a victim to vote, it may provide them with a new way of ensuring the voter submitted the coercer's selection instead of their own.

Filling out the ISB at home or on a shared computer could potentially reveal a voter's votes to others with access to the computer. A voter could leave the tab open on accident, or as mentioned earlier, the application gives the user the option of downloading the ballot when finished or saving their unfinished choices for later and not deleting them after the tab is closed. Any person with access to the computer the vote ISB system was used on may easily gain access to the voter's potential selections. An attacker could also potentially change the voter's selections without their knowledge before they generate their Poll Pass.

The last non-technical attack is to simply use a camera to scan the QR code a voter is holding. If an attacker is able to scan the barcode, they can get an encoded list of the candidates the voter intends to vote for. This can be decoded fairly easily by just making requests to the ISB site and determining which candidates correspond to a given ID. This would be an easy way to potentially reveal who an individual voted for at the vote center if their QR code is not kept secret.

Mitigated Vulnerabilities

During our review of the ISB system, there were several potential attacks we investigated that appear to be mitigated. For instance, the ISB does not allow filling in write-in candidates online - voters have to go to the voting center. As a result, the potential attack vector of malformed write-in candidates was avoided. The site also uses a CDN provider called Incapsula (which we were able to find references to in the site) which prevents any availability issues as a result of a potential denial-of-service attack. The QR code in the Poll Pass also does not have any identifying information about the voter aside from the ballot type.

Other Findings

There were two other things we came across in our investigation that we found interesting. The first is that we found a developer's username "dmurillo" in the filesystem on the real isb.lavote.net website. This potentially reveals information about a developer/sysadmin that could be used in a spear-phishing operation to gain access to the server. We also ran BURP professional web vulnerability scanner on our deployment of the ISB application and it returned no significant findings.

6. Results / Recommendations

Recommendations to LA County

There are a few areas of improvement we suggest for LA County to improve the impact of the ISB on the election system's security as a whole:

Transition to a .gov domain. As .gov domains are exclusive to government organizations/agencies, hosting the ISB application on a .gov domain (which LA County already has

through its lacounty.gov site) would make phishing attacks far less feasible, as similar domains (e.g. votela.net, lavotes.net, etc.) would be impossible to replicate for a .gov domain.

Consider not allowing users to save selections locally. If a voter chooses to save their selections locally, it opens them up to many of the attacks we mentioned earlier. Removing this option would make users complete their ballots in one sitting, removing that potential attack vector entirely.

Remind users to be cautious when downloading Poll Pass. A simple reminder for users to be cognizant of their Poll Pass PDF (i.e. deleting after downloading, or not downloading at all) would help voters keep their selections secret and prevent the next person using that computer from possibly viewing their selections.

Continually audit 3rd-party libraries. We are not sure if the developers have performed an audit on their 23 external dependencies before including them in the codebase. If they have not already done so, we recommend the developers perform an initial audit and audit again every time they decide to update one of these dependencies.

Setup DKIM and SPF records. Currently, the DKIM and SPF records for the lavote.net domain are not setup correctly, enabling potential email spoofing from *@lavote.net - configuring them correctly can reduce the plausibility of a phishing attack.

Use only secure TLS ciphers. As mentioned earlier, a few of the ciphers that the ISB website can use are potentially insecure. Reconfiguring the server should hopefully reduce the risk associated with network attacks, but may come at a usability cost for older devices that may not support newer encryption methods.

Recommendations to Other Jurisdictions

We also have a few recommendations for other jurisdictions who are looking into implementing similar systems to LA County's ISB in the future.

Reuse as much existing election infrastructure as possible. The ISB system does a good job of this by relying on the existing voter identification system in use at voting locations (because voters still have to cast their vote in person). By forcing voters to cast in person, it's essentially a slightly more streamlined form of having voters create a list of candidates to vote for. Re-using existing mechanisms that already exist helps ensure that the system is no less secure than traditional in-person voting.

Remind voters to double-check their BMD choices. All of the attacks we have found against the ISB system so far can be thwarted by a voter double checking their choices and confirming they are correct. However, recent literature shows that less than 10% of voters actually verify their ballots [10]. Interventions are critical to ensuring that voters actually do check their ballots, which the security of the ISB system relies on.

7. Further Work

Further ISB Investigations

There are a few areas of interest we wanted to investigate further but were unable to due to our constraints mentioned earlier. These consist of continuing the investigation of the ISB system as a whole, specifically including the backend server that provides election data to clients (which is a central point of failure for the ISB system), as well as the tabulation/QR code parsing step at the BMD.

Remote Accessible Vote by Mail

It has also come to our attention that LA County uses a similar system for mail-in voting called Remote Accessible Vote by Mail (RAVBM). It uses the same front-end application for users to input their selections. Voters that get sent a regular vote-by-mail envelope can simply use the RAVBM site, print out a ballot with a QR code (instead of a Poll Pass), and send it back using the existing vote-by-mail system for tabulation. This site is vulnerable to all of the same vote leaking/changing/reordering attacks of the ISB application through the same mechanisms as described in section 5, as the source code heavily overlaps between the two applications. There could also potentially be an attack vector with write-in candidates as users are allowed to enter their own selections, though we have not investigated this further.

Those same flaws, however, are *much* more concerning for the RAVBM system - the voter does not have the opportunity to verify their ballot at the BMD that users of the ISB system have. If the QR code listed on the mail-in ballot (which is used for counting as opposed to the listed selections) does not match the listed votes, the voter has no way of knowing, and the changed votes will be counted instead of their intended selections. Furthermore, since mail in votes cannot be tabulated in LA County until after election day [11], any malicious client that prints out invalid ballots would go undetected and the voter may not have a chance to rectify it - their vote could potentially go uncounted.

While RAVBM can be a great tool to increase the accessibility of the voting system, LA County needs to keep in mind the known vulnerabilities of internet voting systems before implementing future systems that may be more connected than this.

8. Conclusion

Overall, the system design of the election system accounts for many vulnerabilities. The ISB is safer than fully online systems because it still requires voters to cast their votes in person, relying on existing mechanisms for authentication. This is much safer than a fully online election system, and is only marginally more insecure than a traditional in-person election.

However, as previously mentioned, any aspect of the voting process that is connected to the internet comes with attacks from adversaries outside of the developers' control. The introduction of adversaries at different levels could in theory compromise both voters' privacy and the integrity of their votes to be cast. Almost all of the attacks mentioned in this paper are inherent to *any* online system - the protection of user data from client-side malware is an unsolved problem. Jurisdictions need to keep these in mind when exploring even simple ways to connect parts of their voting system to the internet.

References

- [1] “Internet Voting.” [Online]. Available: <https://verifiedvoting.org/internetvoting/>.
- [2] Wolchok, Scott, Wustrow, Eric, Isabel, Dawn, and Halderman, J. Alex, “Attacking the Washington, D.C. Internet Voting System,” in *Financial Cryptography and Data Security*, vol. 7397, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 114–128.
- [3] VSAP – LAC Voter-Centered Approach. [Online]. Available: <https://vsap.lavote.net/>. [Accessed: 11-Dec-2020].
- [4] J. Alex Halderman. Internet voting: What could go wrong? San Francisco, CA, January 2016. USENIX As-sociation
- [5] Michael A. Specter, James Koppel, and Daniel Weitzner. The Ballot is Busted Before the Blockchain: A Security Analysis of Voatz, the First Internet Voting Application Used in U.S. Federal Elections. In 29th USENIX Security Symposium (USENIX Security 20), pages 1535–1553. USENIX Association, August 2020
- [6] “Los Angeles County VSAP,” *Los Angeles County VSAP :: California Secretary of State*. [Online]. Available: <https://www.sos.ca.gov/elections/ovsta/voting-technology-vendors/los-angeles-county-vsap>.
- [7] M. Meredith and Y. Salant, “On the Causes and Consequences of Ballot Order Effects,” *Polit Behav*, vol. 35, no. 1, pp. 175–197, Jan. 2012, doi: 10.1007/s11109-011-9189-2.
- [8] “How to Combat Fake Emails,” *Cyber.gov.au*. [Online]. Available: <https://www.cyber.gov.au/acsc/view-all-content/publications/how-combat-fake-emails>. [Accessed: 11-Dec-2020].
- [9] “SSL/TLS Deployment Best Practices,” *Qualys SSL Labs - Projects / SSL/TLS Deployment Best Practices*. [Online]. Available: <https://www.ssllabs.com/projects/best-practices/>. [Accessed: 11-Dec-2020].
- [10] M. Bernhard, A. McDonald, H. Meng, J. Hwa, N. Bajaj, K. Chang, and J. A. Halderman, “Can Voters Detect Malicious Manipulation of Ballot Marking Devices?” [Online]. Available: <https://jhalderm.com/pub/papers/omniballot-20.pdf>.
- [11] J. Myers, “What you should know about how and when California counts ballots,” *Los Angeles Times*, 02-Nov-2020. [Online]. Available: <https://www.latimes.com/california/story/2020-11-02/2020-election-california-ballot-count>.

Appendix A

List of all 3rd party dependencies that the ISB application uses (in the node-modules folder of the React application):

@pdf-lib	pdf-lib	rgbcolor
axios	pdfmake/build	setimmediate
@babel/runtime	process	stackblur-canvas/src
canvg/dist/browser	qr.js/lib	strict-uri-encode
decode-uri-component	qrcode.react/lib	timers-browserify
is-buffer	query-string	webpack/buildin
Lodash	react-dom	whatwg-fetch
pako	regenerator-runtime	

Appendix B

List of TLS ciphers marked as weak by Qualys SSL Labs:

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA