Security Analysis of the Democracy Live Online Voting System

Michael Specter, *MIT;* J. Alex Halderman, *University of Michigan*

https://www.usenix.org/conference/usenixsecurity21/presentation/specter-security

This paper is included in the Proceedings of the 30th USENIX Security Symposium.

August 11–13, 2021

978-1-939133-24-3

Open access to the Proceedings of the 30th USENIX Security Symposium is sponsored by USENIX.

Security Analysis of the Democracy Live Online Voting System

Michael A. Specter
MIT
specter@mit.edu

J. Alex Halderman
University of Michigan
jhalderm@eecs.umich.edu

Abstract

Democracy Live's OmniBallot platform is a web-based system for blank ballot delivery, ballot marking, and online voting. In early 2020, three states—Delaware, West Virginia, and New Jersey—announced that they would allow certain voters to cast votes online using OmniBallot, but, despite the well established risks of Internet voting, the system has never before undergone a public, independent security review.

We reverse engineered the client-side portion of OmniBallot, as used in Delaware, in order to detail the system's operation and analyze its security. We find that OmniBallot uses a simplistic approach to Internet voting that is vulnerable to vote manipulation by malware on the voter's device and by insiders or other attackers who can compromise Democracy Live, Amazon, Google, or Cloudflare. In addition, Democracy Live, which had no privacy policy prior to our work, receives sensitive personally identifiable information—including the voter's identity, ballot selections, and browser fingerprint—that could be used to target political ads or disinformation campaigns. Even when OmniBallot is used to mark ballots that will be printed and returned in the mail, the software sends the voter's identity and ballot choices to Democracy Live, an unnecessary risk that jeopardizes the secret ballot.

We recommend changes to make the platform safer for ballot delivery and marking. However, we conclude that using OmniBallot for electronic ballot return represents a severe risk to election security and could allow attackers to alter election results without detection. In response to our findings, Delaware and New Jersey halted their use of OmniBallot for online voting, but it remains available in other jurisdictions, as do similar tools that likely face the same serious risks.

1 Introduction

COVID-19 has forced states to prepare for the possibility that voters may not be able to vote safely in person in coming elections, and many jurisdictions are turning to forms of online ballot delivery and return to facilitate remote participation. One avenue for doing so is Democracy Live's OmniBallot system, a web-based platform that can be used for blank ballot delivery, ballot marking, and online voting.

OmniBallot has long been used to let voters print ballots that will be returned through the mail, but in early 2020, for the first time, three states announced plans for large classes of voters to use it to return their ballots online. New Jersey recently made the online voting option available to voters with disabilities, calling the move "a pilot for if we need to use it more broadly in the future" [27]. West Virginia allows not only the disabled but also military voters and residents overseas to vote online using OmniBallot [39]. Most significantly, Delaware [24] offered OmniBallot online voting during the presidential primary to all voters who were sick or were self-quarantining or social distancing to avoid exposure to SARS-CoV-2—practically the entire state [11, 24].

Increasing voter access is a laudable goal. Voters who are sick, disabled, or stationed overseas sometimes face substantial obstacles to participation, and the coronavirus pandemic threatens to disrupt in-person voting for everyone. However, elections also face substantial risks from attackers—risks that are magnified when delivering or returning ballot online. Election officials have the complicated job of weighing these risks in light of the access needs of their constituencies.

For online voting, the consensus of election security experts and national security experts is that the risks are unacceptable. Numerous studies of Internet voting systems used or slated for use in real elections have uncovered critical security flaws (e.g., [26, 29, 31, 51, 52, 65]). The National Academies of Sciences, Engineering, and Medicine concluded that "no known technology guarantees the secrecy, security, and verifiability of a marked ballot transmitted over the Internet," and that, "[a]t the present time, the Internet (or any network connected to the Internet) should not be used for the return of marked ballots" [41]. In light of Russia's attacks on U.S. election infrastructure during the 2016 presidential election, the Senate Select Committee on Intelligence has recommended that "[s]tates should resist pushes for online voting," including for military voters [61]. As recently as May 2020, the Cyberse-

curity and Infrastructure Security Agency, Federal Bureau of Investigation, U.S. Election Assistance Commission, and National Institute of Standards and Technology privately warned states that "electronic ballot return technologies are high-risk even with [risk-mitigation] controls in place," and that attacks "could be conducted from anywhere in world, at high volumes, and could compromise ballot confidentiality, ballot integrity, and/or stop ballot availability" [63].

Despite these risks, OmniBallot has not previously been the subject of a public, independent security review,[1] and there is little public documentation about its functionality. Democracy Live even claims that the online ballot return capability should not be considered Internet voting at all, but rather a "secure portal" or "document storage application" [45]. (In fact, it completely matches the definition of Internet voting as used by security experts [1] and by the Election Assistance Commission [59].) Nor have similar ballot delivery and marking products from other vendors been rigorously analyzed. This makes it difficult for voters, election officials, and other policymakers to understand whether the technologies are safe.

In this paper, we present the first public, independent analysis of OmniBallot's security and privacy. We obtained the portion of the software that runs in voters' browsers, reverse engineered it, and created a minimal compatible server in order to study the system's design and operation. Using Delaware's deployment as a model, we describe how the system functions, assess the risks of its various modes of operation, and offer a series of recommendations for the company and for election officials. The analysis was current as of June 7, 2020 and may not reflect later system changes. Our key findings include:

1. OmniBallot's electronic ballot return (online voting) function uses a simplistic approach that cannot achieve software independence [46] or end-to-end verifiability [9], two key goals for secure Internet voting. It also makes extensive use of third-party services and infrastructure: the servers and voter data are hosted in Amazon's cloud, and the client executes JavaScript from both Google and Cloudflare. As a result, votes returned online can be altered, potentially without detection, by a wide range of parties, including Democracy Live itself, insiders at any of these three large tech firms, and attackers who gain access to any of the companies' systems or to a voter's client.
2. The OmniBallot online ballot marking mechanism as used in Delaware needlessly risks violating ballot secrecy by sending the voter's identity and ballot selections to Democracy Live, *even when the voter opts to print the ballot and return it physically through the mail.* There is no technical reason why this information needs to be transmitted over the Internet, and some other jurisdictions have configured OmniBallot to mark the ballot client-side.
3. There are important security and privacy risks even when OmniBallot is used only for delivering blank ballots, including the risk that ballots could be misdirected or subtly manipulated in ways that cause them to be counted incorrectly. Although these risks can be mitigated through careful election procedures, officials need to ensure that the necessary protections are in place, including rigorous post-election audits.
4. In all modes of operation, Democracy Live receives a wealth of sensitive personally identifiable information: voters' names, addresses, dates of birth, physical locations, party affiliations, and partial social security numbers. When ballots are marked or returned online, the company also receives voters' ballot selections, and it collects a browser fingerprint during online voting. This information would be highly valuable for political purposes or for election interference, as it could be used to target ads or disinformation campaigns based on the voter's fine-grained preferences. Nevertheless, OmniBallot had no privacy policy prior to our work, and it is unclear whether there were any effective legal limitations on the company's use of the data.

In this time of widespread social disruption, election officials face intense pressure to make remote voter participation easier and available to more people, but as use of online ballot delivery and return grows, so will the risk that a successful attack could change the result of a major election. We hope that our work will be helpful for states deciding how to conduct upcoming elections in light of COVID-19, and that our analysis of OmniBallot can serve as a template for further security scrutiny of online ballot distribution and return products more generally. Without greater technical transparency and analysis, voters and election officials will be unable to accurately weigh the tradeoffs between risk and access.

2 A Tour of OmniBallot

Much of what is publicly known about OmniBallot comes from a small number of sources, including a FAQ provided by Democracy Live [16], information posted on various sites for jurisdictions' deployments (e.g., [15]), and press statements by the company. In this section, we provide a more complete picture of the system's operation and adoption, based on our own examination of the software.

2.1 Modes of Operation

Each jurisdiction's OmniBallot deployment takes the form of a website at a unique URL. The platform is highly con-

[1] Democracy Live claims that audits have been conducted by the National Cybersecurity Center (a private entity) [43] and ShiftState Security [15], though only high-level summaries of these audits appear to be public. NCC and ShiftState were also claimed to have performed audits of the online voting app Voatz [40], which was later found to have basic, severe security failings [51, 55].

figurable, and jurisdictions can customize the available languages, accessibility options, voter lookup and authentication functions, and available features. Most importantly, jurisdictions can configure the platform to provide any subset of the three modes of operation listed below:

Online blank ballot delivery. The voter downloads a blank ballot corresponding to their home address and/or party affiliation. The ballot is delivered as a PDF file. Most jurisdictions instruct voters to print it, mark it manually, and physically return it to the election authorities.

Online ballot marking. Voters use the website to mark their ballot selections and download the completed ballot as a PDF file. Online marking makes it easier for voters with certain disabilities to fill out their ballots independently. It also allows the website to prevent overvotes and to warn voters about undervotes, reducing errors. The resulting PDF file can be printed and returned physically. Some jurisdictions, including Delaware, also give voters the option to return it via email or fax.

Online ballot return. In some deployments, voters can use OmniBallot to mark their ballots and transmit them to the jurisdiction over the Internet through a service operated by Democracy Live. Like in Washington, D.C.'s attempted Internet voting system [65], jurisdictions print the ballots they receive and then tabulate them with other absentee ballots.

2.2 Deployments

Most instances of OmniBallot appear to be hosted at predictable paths of the form https://sites.omniballot.us/*n*/app, where *n* is the locality's numeric FIPS code [57]. Statewide deployments use two-digit numbers, and counties and cites use five-digit numbers. We visited all pages with these URL formats in May 2020 and found instances for seven state governments and 98 smaller jurisdictions in 11 states.

Nearly all OmniBallot customers offer online ballot delivery, and we found 70 that offer online ballot marking, but only a few appear to allow online ballot return. We found six jurisdictions that have the Internet voting option available:

- Jackson County, OR
  https://sites.omniballot.us/41029/app
- Umatilla County, OR
  https://sites.omniballot.us/41059/app
- Pierce County, WA
  https://sites.omniballot.us/53053/app
- King Conservation District, WA
  https://sites.omniballot.us/kcd/app
- State of West Virginia
  https://sites.omniballot.us/54/app
- State of Delaware
  https://ballot.elections.delaware.gov/app

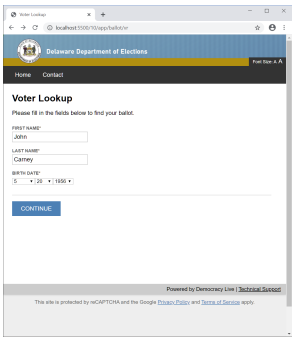
New Jersey also announced plans to use Democracy Live for online voting [38, 53] and reportedly did use it for local school board elections in May 2020, but we were not able to locate a deployment for the state.

2.3 The Voter's Perspective

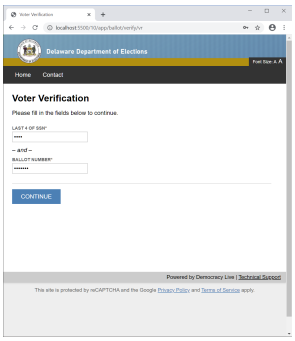
We now describe how OmniBallot works from a voter's perspective. Screenshots in Figure 1 illustrate each step. We use the Delaware deployment as a concrete example, noting some of the differences in other deployments where applicable.

1. **Welcome.** Voters visit the main URL of the website and are greeted by a welcome screen. The voter clicks a button to "Mark My Official Ballot."
2. **Voter lookup.** The voter enters their first and last name and birthdate, and the site locates them in the voter registration database. If multiple voters match, the site lists their street addresses and asks the voter to choose one.
3. **Verify voter.** In Delaware, voters entered the last four digits of their social security numbers and a "ballot number" provided by the state in an email sent by the election administrators. These were verified by the server before the voter is allowed to proceed. Some other deployments we examined did not use this verification step.
4. **Return type.** Delaware let voters opt to return their ballots by mail, by fax, by email (using a webmail portal), or through OmniBallot's Internet voting mechanism ("electronic return"). If mail, fax, or email return was selected, voters could either mark their ballots using the site and generate PDF files to return or retrieve blank ballot PDFs and mark them manually.
5. **Ballot marking.** The voter can scroll through the ballot and make selections. Write-in candidates can be entered using the keyboard where permitted. The site will refuse to mark more than the allowed number of candidates.
6. **Selection review.** A summary screen shows the selections in each race (or a warning if the voter made fewer than the allowed number of sections). The voter can return to the ballot to change selections or proceed to casting.
7. **Signature.** Voters are instructed to sign their names with the mouse or touch screen, or to type their names. The result is captured as a bitmap image. Some other jurisdictions do not allow a typed signature and instruct voters that their signature must match the signature on file with the jurisdiction.[2]

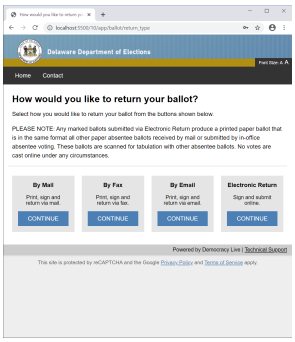
[2] On-screen signatures often differ dramatically from signatures made on paper [20].



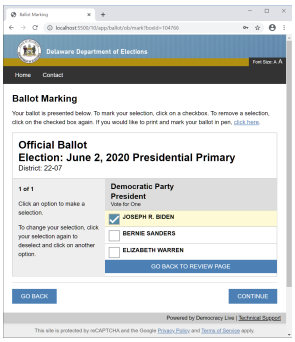
(a) Voter Lookup



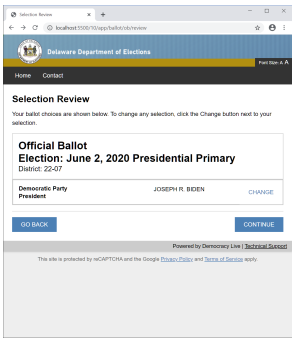
(b) Verify Voter



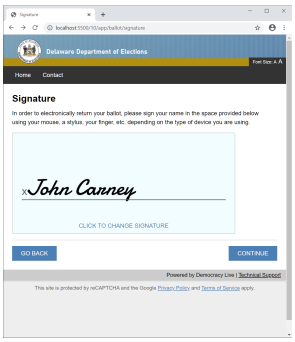
(c) Return Type



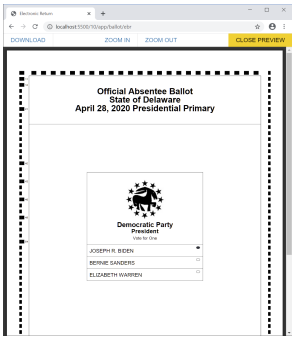
(d) Ballot Marking



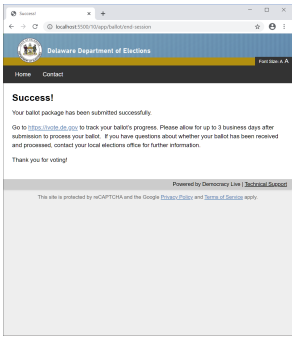
(e) Selection Review



(f) Signature



(g) Preview



(h) Ballot Submitted

Figure 1: **Online voting with Democracy Live**, as used in Delaware. The voter's identity and ballot selections are transmitted over the Internet to generate a PDF ballot. Election officials later retrieve the ballot files and tabulate the votes. All screenshots in this paper were captured with a local stand-in server.

8. **Electronic return.** Voters are shown a preview of their return packages (which includes their identification information and signature page) and their completed ballot. These are PDF files that the site renders with JavaScript.

9. **Ballot submitted.** When voters are satisfied, they click a button to submit the ballot over the Internet. In Delaware, voters could check whether a ballot in their name has been accepted using their ballot numbers. However, unlike the confirmations provided by E2E-V systems, this mechanism could not protect the ballot selections from modification.

Alternatively, if voters choose to download a blank ballot or to mark a ballot to send via mail, fax, or email, they follow a different path through the site. There is no signature screen after marking the ballot, and instead the voter is provided with a downloadable PDF file of the ballot and return package.

3 System Architecture and Client Operations

From the client's perspective, each OmniBallot site is a single-page web app. The app is written using the AngularJS framework and implemented as a combination of static HTML, JavaScript, CSS, and JSON-based configuration files. This code runs in the voter's browser and performs all steps of the voting process via a series of API calls to services controlled by Democracy Live. Below, we explain how we performed our analysis, describe the overall architecture of the platform, and provide details of the web app's operation.

3.1 Reverse-Engineering Methodology

Researchers have conducted numerous independent analyses of electronic voting systems by acquiring voting equipment, reverse engineering it, and testing it in a controlled environment (see [30] and references therein). Safely testing an *online* voting system is more challenging. Such systems necessarily have server-side components that (unless source code is available) cannot be replicated in the lab. Accessing non-public server functionality might raise legal issues and would be ethically problematic if it risked unintentionally disrupting real elections [47].

To avoid these issues, we constrained our analysis to publicly available portions of the OmniBallot system. Following similar methodology to Halderman and Teague [31] and, more recently, Specter et al. [51], we obtained the client-side OmniBallot software, which is available to any member of the public, reverse-engineered it, and implemented our own compatible server in order to drive the client without interacting with the real voting system. Of course, this approach limits our ability to identify vulnerabilities in Democracy Live's *server-side* code and infrastructure—an important task for future work—but we were able to learn many details about the platform's design and functionality.

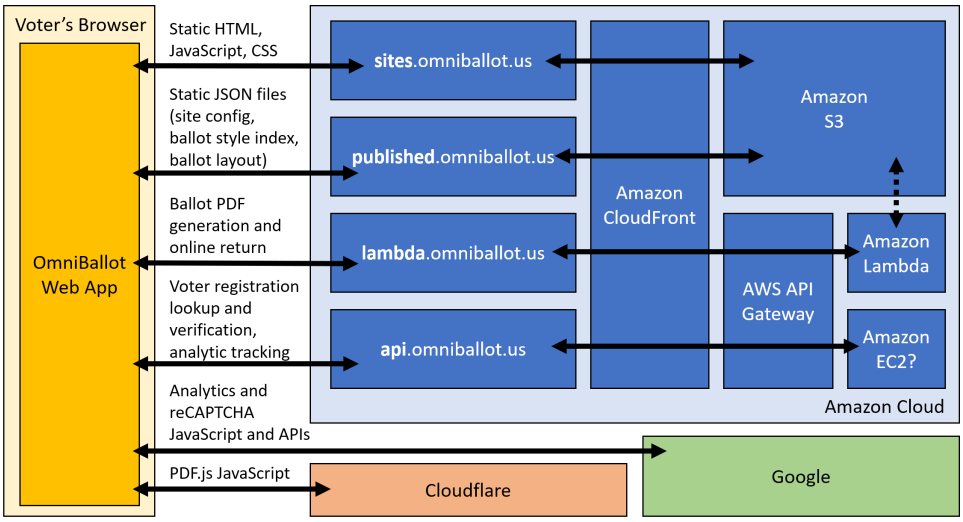


Figure 2: **OmniBallot architecture.** The web app runs in the browser and uses HTTPS to load files and call REST-like APIs from several domains. When voting online or marking a ballot, the app sends the voter's identity and ballot selections to Democracy Live services running in Amazon's cloud. The app runs JavaScript loaded from Amazon, Google, and Cloudflare, making all three companies (as well as Democracy Live itself) potential points of compromise for the election system.

For our analysis, we focused on the instance of OmniBallot deployed in Delaware, which was available at https://ballot.elections.delaware.gov/. As of June 7, 2020, the site used OmniBallot version 9.2.11, which we believe was the most recent version of the system at that time. We began by visiting the site and saving copies of the files that comprise the client. We beautified [35] the minified JavaScript files and ensured that they would not communicate with any live election services by replacing references to `*.omniballot.us` domains with `localhost` and disabling Google's services.

Next, we iteratively reverse-engineered the code to understand each server API call and the format of the expected response, repeating this process until we could complete the voting process using a local stand-in server we created. Finally, we confirmed and extended our reconstruction of the system's operation by inspecting HTTP traces captured by a Delaware voter while using the live system.

Other than accessing resources that are available to the general public, the authors had no interaction with the OmniBallot servers. At no point did we attempt to log in as a real voter or cast a ballot in a real election.

3.2 Service Architecture

The web app communicates with several servers to load static files or make API calls, as illustrated in Figure 2. Four of these services are controlled by Democracy Live and hosted in Amazon Web Services: `{sites, published, lambda, api}.omniballot.us`; all use Amazon CloudFront as a CDN and have HTTPS certificates for `*.omniballot.us`. The app also loads JavaScript libraries from Google (Google Analytics and reCAPTCHA [64]) and Cloudflare (PDF.js).

The `sites` and `published` servers appear to be backed by Amazon S3. The `sites` server hosts the static HTML, JavaScript, and CSS of the web app, with different paths containing different jurisdictions' deployments or different versions of the code. The `published` server hosts static JSON files that specify the configuration of each deployment (`site-config.json`), provide an index of ballot styles (`lookups.json`), and define each ballot. The `site-config.json` file defines the appearance and workflow of the web app, allowing individual app instances to be heavily customized for each jurisdiction.

The `api` server handles voter lookup and authentication. It provides a REST-like API that allows clients to query for specific voter and ballot information as JSON-encoded HTTP queries and responses. The service is hosted through AWS API Gateway, and may be backed by an Amazon EC2 instance. The `lambda` server uses a similar API format to process ballot PDF generation requests and online ballot return submissions, and it appears to be backed by code running on the Amazon Lambda serverless computing platform. Calls to both servers include an `x-api-key` HTTP header set to a hard-coded value.

3.3 Client–Server Interactions

In Delaware, the client-server interactions proceeded along the following lines:

1. The browser visits https://ballot.elections.delaware.gov/ and loads the base HTML page, which defines the site configuration file as https://published.omniballot.us/10/site-config.json and loads the app's base code from

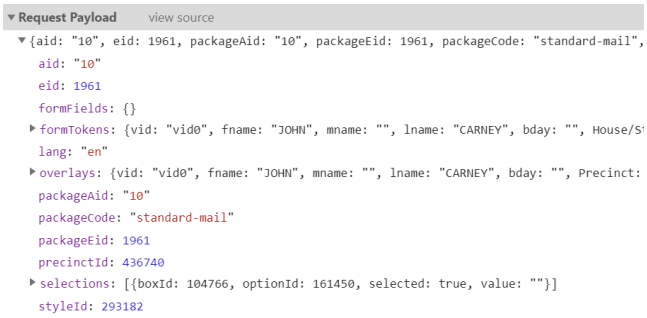


Figure 3: In Delaware, **marked ballot generation** took place on OmniBallot servers. The app sent a POST request (*above*) that included the voter's identity and ballot selections. The server returned the marked ballot as a PDF file. Online voting used a similar request format, with the addition of a browser fingerprint. Marking ballots server-side increases risks to election integrity and ballot secrecy.

https://sites.omniballot.us/v9_2_11/combined.js. The app dynamically loads 24 other JavaScript modules from under the same path. It also loads the Google Analytics library from https://www.googletagmanager.com and the reCAPTCHA library from https://www.gstatic.com.

2. The app looks up the voter's registration information by making a POST request to https://api.omniballot.us/vr/*db*/voters/lookup. This request (and all later POST requests) includes headers for the reCAPTCHA API as an abuse protection mechanism. The request contains the voter's first and last names and date of birth. The server responds with the registration data, including a unique id (*voter_id*), whether the user is a "standard" or military (UOCAVA) voter (*voter_type*), and their party (*voter_party*) and precinct.

3. The app verifies the voter's identity by making a POST request to https://api.omniballot.us/vr/*db*/voter/*voter_id*/verify. The request includes the election ID as well as the ballot number and partial social security number entered by the user. If verification succeeds, the server returns a signed JSON Web Token that authenticates the voter_id.

4. To find available elections, the app sends a GET request to https://api.omniballot.us/accounts/*account_id*/current elections?voter_type=*type*&voter_party=*party*. The server returns a JSON object for each election with the election name, ID, parent_id, and opening and closing dates. The app then locates the appropriate ballot design by loading https://published.omniballot.us/10/*parent_id*/styles/lookups.json, which is a data structure that associates ballot styles with precincts, parties, and voter types. The ballot itself is defined in a static JSON object retrieved from https://published.omniballot.us/10/*parent_id*/styles/*style_id*.json.

5. If the voter chooses to return the ballot via postal mail, fax, or email, the web app generates a ballot PDF file by making a POST request to https://lambda.omniballot.us/packagebuilder/v2. The request includes an HTTP `Authorization: Bearer` header that contains the voter authentication token acquired above. The request body, shown in Figure 3, specifies the election, the ballot style, and the voter's name and other registration information. If the voter is marking the ballot, it also includes the ballot selections, encoded as an array of race and selection identifiers. The server returns a URL to a PDF file containing the generated ballot. The file is hosted in Amazon S3, and the URL is a pre-signed object URL [5] with a five-minute expiration.

6. Online ballot return uses a similar API. The app makes a POST request to https://lambda.omniballot.us/ebr/build with the same authorization header. The request contains the same kinds of data as ballot marking, including the voter's identity, registration information, and ballot selections. In addition, the request contains a browser fingerprint generated using FingerprintJS [62] and a base64-encoded PNG image of the voter's signature. The server returns a ballot ID and URLs from which the client can retrieve PDF files of the marked ballot and return package. These are rendered in the browser using the PDF.js library, which is retrieved from `cdnjs.cloudflare.com`.

7. Finally, to submit the ballot online, the client makes a POST request to https://lambda.omniballot.us/ebr/submit, again including the authorization header. The request contains the voter_id and the ballot_id from the previous step, but the ballot selections are not resent. Based on Democracy Live's statements about using Amazon ObjectLock [4], we assume that this API call causes the server to place the return package and ballot PDFs into an ObjectLock-enabled S3 bucket for delivery to election officials. The server sends a response indicating success, and the voting process is complete.

4 Security Analysis

We now assess the security and privacy risks of the OmniBallot platform. We analyze risks created when OmniBallot is used in each of three modes—blank ballot delivery, ballot marking, and online ballot return—and we discuss how (or whether) they can be mitigated. We consider three main classes of adversaries:

Adversaries with access to the voter's device. The client-side adversaries with which we are most concerned are ones with the ability to alter the behavior of the voter's web browser, such as by modifying HTTP requests or responses or injecting JavaScript into the context of the site. Several kinds of threat actors have these capabilities, including system administrators, other people with whom the voter shares the device (e.g., an abusive partner), and remote attackers who control

malware on the device, such as bots or malicious browser extensions.

Client-side malware is especially concerning because many devices are already infected by malicious software that could be remotely updated to attack OmniBallot. For instance, Microsoft this year took down a botnet controlled by Russian criminals that had infected more than nine million PCs [48]. Botnets are sometimes rented or sold to other parties to perpetrate attacks [32]. Similarly, researchers recently uncovered more than 500 malicious Chrome extensions in use by millions of people [33], and a popular *legitimate* Chrome extension was hijacked and modified to forward users' credentials to a server in Ukraine [34]. Attackers could use these strategies to target large numbers of OmniBallot voters.

Adversaries with access to OmniBallot server infrastructure. The platform's architecture makes server-side adversaries extremely powerful. Depending on which services they compromised, they could change the code delivered to clients, steal sensitive private information, or modify election data, including voted ballots. Potential attackers with such access include: (1) software engineers and system administrators at Democracy Live; (2) insiders at Amazon, which owns and operates the physical servers; and (3) external attackers who manage to breach the servers or Democracy Live's development systems.

Adversaries with control of third-party code. Beyond its reliance on Amazon's cloud, OmniBallot incorporates a wide range of third-party software and services, including AngularJS, FingerprintJS, PDF.js, Google Analytics, and reCAPTCHA. Since all this code runs within the app's browser context, it has the ability to access sensitive data or introduce malicious behavior. In recent years, attackers have hijacked several popular JavaScript libraries to target users of software that incorporates them (e.g., [56]). Moreover, OmniBallot clients load some libraries directly from Google and Cloudflare, putting these companies (as well as Amazon) in a position to surreptitiously modify the web app's behavior.

Even large, sophisticated companies are not beyond being compromised by nation states—see, e.g., Operation Aurora, in which China infiltrated Google and a number of other high-tech companies [67]. While Amazon, Google, and Cloudflare have significant incentives to protect their infrastructure and reputations, they also have large stakes in the outcome of major elections, and individual employees or small teams within the companies may feel strong partisan sympathies and have sufficient access to attack OmniBallot. Furthermore, even if these companies' services were perfectly secure against insiders and exploitation, voters may still be distrustful of their ability to handle votes impartially—just as some of the public does not trust the Washington Post under Jeff Bezos's ownership—weakening the perceived legitimacy of elections.

The subsections that follow discuss attacks that these threat actors could carry out against OmniBallot's blank ballot delivery, online ballot marking, and electronic ballot return features, and against voters' privacy. We omit some important categories of attacks, including denial-of-service attacks and attacks against voter authentication, due to limits of what we can learn without access to the servers or detailed local election procedures. Table 1 summarizes our analysis.

4.1 Risks of Blank Ballot Delivery

OmniBallot's safest mode of operation is online delivery of blank ballots that will be printed, manually marked, and returned physically through postal mail or drop off. (Returning the ballots via email or fax leads to severe risks, which we discuss separately.) Online blank-ballot delivery can provide a valuable enhancement to vote-by-mail systems, but election officials must implement rigorous safeguards to protect against several categories of attacks.

Ballot design manipulation. One mode of attack would be to alter the ballot design. For instance, an attacker could change or omit certain races or candidates or substitute a ballot from a different locality. Such changes might be spotted by well informed voters, but other, harder to detect modifications could cause votes to be counted for the wrong candidate when tabulated by a scanner. For instance, attackers could modify bar codes or timing marks, or shift the positions of selection targets. Conducting these attacks would be straightforward for adversaries with control of the client device, server infrastructure, or third-party code.

To protect against ballot design manipulation, officials first need to check that each returned ballot matches the voter's assigned ballot style, using careful procedures to preserve ballot secrecy. Next, since visual inspection likely cannot detect all modifications that would cause tabulators to miscount the votes, officials either need to count the ballots by hand or manually "remake" the ballots (transfer the votes onto pre-printed ballots) before scanning them. An effective alternative would be to perform a risk-limiting audit [36] (which is necessary in any case to protect against other kinds of error and fraud), but Delaware, West Virginia, and New Jersey do not conduct state-wide RLAs.

Ballot misdirection. Another way to attack blank ballot delivery would be to modify the ballot return instructions, rather than the ballot itself, in order to cause voted ballots to be sent to the wrong place or be delayed until too late to count. In Delaware, OmniBallot included the return instructions and a printable envelope in the same PDF file as the ballot. The attacker could replace the entire delivery address or simply change the zip code or postal bar code to route the ballot to a distant sorting facility. Since OmniBallot verifies the voter's identity before providing the return package, an attacker could decide which ballots to misdirect based on the voter's place of residence or party affiliation.

Voters might detect that their ballots have been misdirected if the jurisdiction provides a ballot tracking service. However, the attacker could simultaneously mail a different ballot in

Configuration	Attacker Capability			Risk
	Manipulate Ballot Design	Compromise Ballot Secrecy	Invisibly Change Votes	
Blank Ballot Printing	(C)(S)(T)			Moderate
Marked Ballot Printing	(C)(S)(T)	(C)(S)(T)		High
Online Ballot Return	(C)(S)(T)	(C)(S)(T)	(C)(S)(T)	Severe

Table 1: **OmniBallot risks.** We show what kinds of attacks are possible when OmniBallot is used in different modes, if an attacker compromises the voter's client (C), Democracy Live's services (S), or third-party infrastructure (T). Ballot designs can be manipulated in all cases. When ballots are marked online, Democracy Live servers see the voter's identity and selections. When ballots are returned online, attackers could potentially change votes without being detected.

the voter's name—but with votes for the attacker's preferred candidates—reusing the voter's identity information taken from the web app. This would make it appear to voters that their ballots had been received.

Officials can partially defend against misdirection by providing correct ballot return instructions through prominent channels other than OmniBallot, such as on other official sites and in the media. We also recommend that states coordinate with the Postal Service to ensure that postal workers are on the lookout for misdirected ballots.

4.2 Risks of Online Ballot Marking

Using OmniBallot to mark ballots online, print them, and return them physically raises greater risks than blank ballot delivery. (Again, marking ballots online and returning them via email or fax leads to severe risks, which we discuss separately.) Some of the risks can be mitigated with careful procedures, but others are difficult to avoid, especially if online ballot marking is widely used.

Enhanced ballot misdirection and manipulation. OmniBallot's online ballot marking configuration could allow attackers to see the voter's selections before the ballot is generated, allowing them to surgically suppress votes for a particular candidate by misdirecting or modifying only those ballots. The attacker could also reorder the candidates, move the selection targets or timing marks, or encode false votes within barcodes, so that the ballot appears (to a human) to be marked for the voter's selected candidate but will be counted by an optical scanner as a vote for a different candidate. These risks make the procedural defenses discussed in § 4.1 even more crucial when jurisdictions offer online ballot marking. However, "remaking" the ballot by reading the votes from a barcode, as some jurisdictions do, introduces further security risks, since attackers could change the barcodes without detection. Instead, absent a risk-limiting audit, officials must manually transcribe the human-readable selections to a preprinted ballot.

Ballot mismarking. Online marking enables a simpler style of ballot manipulation that may be impossible to procedurally mitigate: mismark the ballot so that one or more races reflect the attacker's choices instead of the voter's.

Of course, voters could detect this by carefully reviewing their ballots before returning them. However, recent research involving ballot marking devices—which are susceptible to analogous attacks—finds that the vast majority of voters fail to detect errors on machine-marked paper ballots [10]. OmniBallot users who did notice a problem would likely discard the erroneous ballot and use the system to mark another; the attacker could recognize this repeat attempt and mark the new ballot correctly. Even if a few voters alerted election officials, the voters would have no way to prove that the system misbehaved, so officials would have difficulty distinguishing an attack from isolated human error [7].

Prompting voters to carefully review their ballots may increase error detection to a limited extent. However, modeling suggests that the improvement may not be sufficient to detect outcome-changing fraud in close elections unless use of electronic ballot marking is limited to a small subset of voters [10].

Compromising ballot secrecy. Online ballot marking carries an elevated risk that attackers could compromise the voter's secret ballot. Attackers with the ability to alter or inject code into the web app could exfiltrate the voter's identity and ballot choices. Moreover, since the web app sends the voter's identity and ballot choices to `lambda.omniballot.us` in order to generate the marked ballot PDF file, an attacker with only passive access to the data processed by this service can learn voters' ballot selections, even when the ballot is returned physically.

Furthermore, the ballot return package, including the voter's identity and marked ballot, is saved locally to the voter's computer before being printed. This creates a risk that client-side attackers, including other local users, could gain access to the file. Even if voters delete the files, forensic tools may allow adversaries to recover the ballots long into the future [25].

4.3 Risks of Online Ballot Return

OmniBallot's online ballot return mode carries similar risks to online ballot marking as well as severe additional risk that cast votes could be changed at large scale without detection. These risks cannot be adequately mitigated with procedural changes or readily available technology.

Lack of end-to-end verifiability. Computer scientists have been working for more than 30 years to develop principled techniques for secure remote voting [8]. These protocols use an approach called "end-to-end verifiability" (E2E-V), which (among other properties) allows each voter to independently check that their vote is correctly recorded and included in the election result [9]. Cryptographic E2E-V protocols such as Helios [2] accomplish this without requiring the voter to trust a particular client device or the official election software or servers. These technologies are promising—both for remote voting and as an added layer of protection for traditional voting [37]—but they are also complex and difficult to implement correctly [29]. For this reason, although experts hold that E2E-V should be a requirement for any Internet voting system, they simultaneously caution that "no Internet voting system of any kind should be used for public elections before end-to-end verifiable *in-person* voting systems have been widely deployed and experience has been gained from their use" [21].

OmniBallot does not attempt to achieve E2E verifiability. Instead, it uses a protocol that provides no way for voters, officials, or Democracy Live itself to verify that the ballot selections a voter chooses are the same as what officials receive. Consequently, an attacker with control of the voter's client, of Democracy Live's infrastructure, or of any of the third-party services from which the client loads JavaScript, could change recorded votes. Unlike ballot marking with physical return, where the voter has a chance to review the printed ballot that is sent for tabulation, voters have no practical ability to detect vote-changing attacks involving online ballot return. Nor do election officials. Democracy Live itself would have little opportunity to detect attacks that were perpetrated by client-side malware or third-party infrastructure.

Vote-changing attacks. Recall that OmniBallot's online voting is accomplished by making two API calls to `lambda.omniballot.us`: one that submits the voter's identity and selections and receives a ballot ID and a URL for the marked ballot PDF file, and another that submits the ballot ID and causes the ballot to be delivered to election officials. Both requests are authenticated with a bearer token that is provided after checking the voter's identity.

One way to subvert this process would be to inject malicious code into the web app. This could be accomplished with local malware (such as a malicious browser extension) or by delivering malicious code as part of the JavaScript that OmniBallot loads from Amazon, Google, and Cloudflare servers. Insiders at these companies or at Democracy Live could attempt such an attack, as could external attackers who compromised any of the companies' infrastructure.

Once in control of the client, the attacker could cause the web app to substitute ballot selections of the attacker's choosing. To hide the changes from the voter, the attacker would simply have to generate a separate ballot PDF file to display to the voter that *did* match the voter's selections. This could be accomplished by modifying the real ballot PDF file using client-side code. As a result, the web app would show a ballot containing the selections the voter intended, but the ballot that got cast would have selections chosen by the attacker. The attack would execute on the client, with no unusual interactions with Democracy Live, so there would be no reliable way for the company (or election officials) to discover it.

Attackers with control of the `lambda.omniballot.us` service—such as malicious insiders at Democracy Live or at Amazon, or external attackers who penetrated either company's systems—would have a separate way of changing votes. Malicious code on this server could return one PDF to the voter and store a different one for delivery and counting. Voters would have no way to notice the change.

Insufficient controls. Available documents give us some visibility into Democracy Live's server-side defenses and internal controls. These controls appear to have either limited or no ability to prevent the attacks we have described.

The company says that voted ballots are stored immutably in Amazon S3 using AWS Object Lock [16].[3] While an immutable store does provide some security benefits, it cannot prevent the attacks described above. Object Lock can only protect files from modification after they are stored, so it cannot prevent attacks that modify the ballot before it is placed in S3. It also cannot protect ballots from modification by insiders at Amazon with internal access to the storage system. Moreover, Democracy Live appears to use Object Lock in "governance mode," which means the protections can be bypassed by the root user or other insider accounts with special permissions [17].

Following a pilot of electronic ballot return during a January 2020 election held by Washington State's King Conservation District, Democracy Live conducted what it called a "post election security audit" in order to "verif[y] the integrity of the [. . .] election" and "identify potential malfeasance on the part of Democracy Live employees." An unpublished report by the company [17] explains that the "audit" consisted of a review of log entries created by Amazon's AWS CloudTrail log service [6], and it lists ten specific log queries that were performed. We note that these queries did not cover all vectors by which insiders or other attackers could have modified votes. For instance, although the audit included looking for log entries that would occur if an employee logged in under the root account or attempted to remove a restriction

[3] Object Lock refers to a configuration of Amazon's S3 storage service that allows the developer to designate certain classes of information unmodifiable for various retention periods and configurations [4].

Voter Private Information	Configuration		
	Blank Ballot Delivery	Online Ballot Marking	Online Ballot Return
IP address/coarse physical location	✓+	✓+	✓+
Delaware voter ID number	✓+	✓+	✓+
Name, address, and date of birth	✓*	✓*	✓*
Party affiliation	✓*	✓*	✓*
Partial social security number	✓	✓	✓
Vote selections		✓	✓
Browser fingerprint			✓

Table 2: **Access to privacy-sensitive data.** We show what data is shared with Democracy Live when using OmniBallot in each mode offered in Delaware. A + indicates that the information is also sent to Google; a * indicates that Google can infer it. All data is implicitly sent to AWS.

on bypassing Object Lock, it apparently did not search for attempts to modify the software downloaded by clients or the software running the `lambda` service. As we have explained, changing either piece of software would be sufficient to allow an attacker to view and alter votes.

Such a limited analysis is insufficient to verify the integrity of an election, as it cannot detect the full range of sophisticated threats that public elections face. No matter how comprehensive, server-side logs cannot protect against client-side attacks or attacks conducted through third-party services, since such events would occur outside of Democracy Live's control. Likewise, no level of auditing or procedural controls can eliminate the threat that attackers will introduce malicious functionality into software without detection, and deliberate vulnerabilities can be extremely subtle and difficult to detect (e.g., [12, 23]). Internal audits also provide little assurance against the threat that the employees who conduct them are themselves malicious. Finally, reviewing logs is necessarily retrospective, so, even if a vote-changing attack was uncovered, detection would likely occur only after the election. Since Internet voting lacks voter-verified paper records from which the correct votes could be recovered, officials might be forced to rerun the election.

4.4 Risks of Email-Based Ballot Return

Like other modes of online voting, email-based ballot return faces severe security risks that cannot be adequately mitigated with available technology or controls. Different OmniBallot jurisdictions use widely varying procedures for email-based return; here we focus on the way it is implemented in Delaware. Even after discontinuing OmniBallot, Delaware allowed voters to return ballots by email.

Delaware voters who choose to return their ballots by email are instructed to use Egress Switch [54], a "secure email" platform produced by U.K.-based Egress Software Technologies, Ltd. Rather than directly emailing the ballot, voters visit https://switch.egress.com and sign up for accounts using their email addresses. After proving that they have received a confirmation code sent to that address, the voter can log in to a webmail interface, compose a message to a Delaware elections email address, and attach the voted ballot as a PDF file. The recipient receives an email notification that the message is available and can log in to the same system to retrieve it.

A full analysis of Egress Switch is beyond the scope of this paper, but we note that it is effectively serving as a second Internet voting platform, with broadly similar risks to OmniBallot's online return mode, including a reliance on large tech companies for trusted infrastructure. Egress appears to be hosted in Microsoft's cloud and to store encrypted messages in Amazon S3 servers located in the U.K. Routing domestic voters' ballots through a foreign jurisdiction may weaken the legal protections surrounding ballot secrecy and exposes voters to a greater risk of surveillance or other attacks by a foreign government [13].

Depending on the voter's existing email provider, Egress Switch may offer privacy advantages, particularly as the sender may only view sent messages for a limited time. On the other hand, it centralizes voted ballots on a single third-party platform, which must be trusted to deliver them without modification. As with OmniBallot, Switch itself, and the third-parties it trusts, can see and change the ballot before it is delivered, and there is no apparent mechanism by which voters can independently confirm that their voted ballots have been received by election officials without modification.

4.5 Risks to Voters' Privacy

OmniBallot has access to a large amount of privacy-sensitive data (see Table 2): voters' names, addresses, dates of birth, party affiliations, and other voter registration fields; their coarse physical locations from their IP addresses; their partial

social security numbers; and, in either the ballot marking or online voting configurations, their actual ballot selections.

In addition, when votes are cast online, OmniBallot's client-side code takes a fingerprint of the browser and sends it to the server with the voter's registration data and ballot selections. If Democracy Live shared this data with other sites, they could recognize the voter's browser and associate it with their identity and votes. Browser fingerprints are incredibly privacy invasive [22]—they can uniquely track a browser even after the user has taken defensive measures such as clearing cookies, as well as between private browsing and normal browser modes [66].

This data about the voter would be valuable to many parties: advertisers, political candidates, or attackers seeking to conduct disinformation campaigns. Notably, Democracy Live appears to be silent about whether, or for how long, they store this data, how they use it, or whether it will be shared or sold to third parties. Prior to our work, OmniBallot included no terms of service or privacy policy (though it did link to Google's, as sites that use reCAPTCHA are required to do).

OmniBallot also makes extensive use of first- and third-party tracking mechanisms to monitor voters' interactions with the platform. It sends Google Analytics extensive browser configuration information, the URLs of pages the voter visits within the app, whether they are a UOCAVA voter, and the voter's ID number. In Delaware, the same ID number is used in the state's publicly available voter file, where it is associated with the voter's full name, address, phone number, birth year, and party. Google could use the ID field to personally identify the voter and potentially to associate the voter's identify with other tracking cookies.[4]

4.6 Risk Summary

Below, we briefly summarize our findings concerning OmniBallot's three main modes of operation. Our assessment of their relative risk accords with recent guidance by the U.S. Cybersecurity and Infrastructure Security Agency [58, 63].

Blank ballot delivery. When OmniBallot is used to deliver blank ballots for printing, attackers could modify certain voters' ballots or return instructions to omit candidates, cause votes to be scanned incorrectly, or delay or misdirect mail-in returns. These risks can be largely mitigated with rigorous election procedures, and, with such protections in place, we consider the overall risk to be moderate.

Online ballot marking. Using OmniBallot to mark and print ballots carries greater risks. Attackers can learn the voter's selections and target ballots for a disfavored candidate by misdirecting them or causing them to be scanned as a vote for somebody else. Attackers could also mark the ballot for different candidates than the voter intended, which, although visible, many voters would likely fail to detect. Voter education and procedural defenses can only mitigate these attacks to an extent, so we consider the risk to be high. As the risk further increases when online marking is widely used, we recommend limiting its deployment.

Online ballot return. When ballots are returned over the Internet using OmniBallot, there is no way for voters to confirm that their votes have been transmitted without modification, and attackers could change votes in ways that would be difficult for voters, officials, or Democracy Live to detect. Attacks could be conducted through client-side malware, compromise of third-party services such as Amazon and Google, or infiltration of Democracy Live. Administrative controls and audits cannot prevent such attacks. Given the possibility for undetected changes to election results, we consider the risks of online voting to be severe.

5 Recommendations

Based on our analysis, we offer a series of recommendations for election administrators, policymakers, and Democracy Live in order to help protect the integrity of elections conducted using OmniBallot and safeguard voters' privacy. These are in addition to the procedural defenses discussed in § 4. Many of these recommendations apply more generally to all systems for online voting or ballot delivery and marking that jurisdictions may be using or considering.

We conveyed these recommendations and a summary of our findings to the U.S. Cybersecurity and Infrastructure Security Agency, which communicated them to state officials, and we discussed them with Democracy Live's management team. In response, Democracy Live made some limited improvements, such as adding a privacy policy. Delaware and New Jersey discontinued use of OmniBallot for online voting [49], but Delaware continued to allow webmail-based ballot return.

Eliminate electronic ballot return. OmniBallot's online ballot return functions run counter to the clear scientific consensus, as expressed by the National Academies [41], that the Internet should not be used for the return of marked ballots. Our analysis shows that votes cast online using OmniBallot could be surreptitiously changed without voters, officials, or Democracy Live being able to detect the attack. Given the risks, we recommend that elections administrators refrain from using online ballot return, including ballot return via email. Instead, administrators should focus on improving the efficiency and accessibility of physical ballot return paths, which carry fewer risks of large-scale manipulation.

Limit the use of online ballot marking. In the ideal case, online ballot marking provides valuable usability and accessibility benefits. For absentee voters with disabilities that make it impossible to mark ballots by hand, such a tool could provide greater independence and privacy. At the same time,

[4] This behavior appears to be in violation of the Google Analytics terms of service [28], which prohibit sending personally identifiable information to Google.

it carries higher risks of ballot misdirection, manipulation, and mismarking than blank ballot delivery, and research with ballot-marking devices suggests that most voters will fail to spot altered ballots, even if prompted to check [10]. As online marking becomes used more widely, it becomes a more attractive target, and the risk that attacks could change election outcomes increases rapidly. For these reasons, we recommend offering online marking only to voters who could not otherwise mark a ballot independently, and not to the general public. Furthermore, marked ballots should always be printed and physically returned.

Mark ballots using client-side code. OmniBallot's design, as used in Delaware, creates unnecessary risks to ballot secrecy and integrity by sending the voters' selections, coupled with their identities, to an online service when generating marked ballots. These risks could be avoided by marking ballots locally in the browser, using client-side code.

Democracy Live already offers an option to do this. OmniBallot deployments in California, Virginia counties, and Washington, D.C. use an alternative online marking approach called "Secure Select," in which marked ballots are generated without sending selections to a server [50]. After downloading the return package, the voter is redirected to a page on `ss.liveballot.com`, which delivers JavaScript for generating the marked ballot entirely within the browser.

In addition to Delaware, jurisdictions in Colorado, Florida, Ohio, Oregon, Washington State, and West Virginia appear to use the more dangerous server-side marking mechanism. We recommend that they switch to client-side marking.

Implement risk-limiting audits. When OmniBallot is used to deliver blank ballots that are marked by hand and physically returned, this generates a strongly voter-verified record of voters' choices. However, attackers can still manipulate the ballot design in ways that would cause votes to be miscounted when tabulated by an optical scanner. To mitigate this, we recommend that officials perform risk-limiting audits (RLAs) [36], which limit the probability that the election outcome differs from the outcome that would be found by a full hand-count. As with in-person voting, RLAs are an essential defense against error and fraud.

Reduce unnecessary trust in third parties. OmniBallot's security depends not only on the security of Democracy Live's code and procedures, but also on the security of services provided by Amazon, Google, and Cloudflare. Attackers that breach their systems (or rogue employees within the companies) could alter votes that are returned electronically. Democracy Live can reduce this risk, to an extent, by removing inessential dependencies (e.g., Google Analytics) and applying subresource integrity [3] to static libraries (e.g., PDF.js). However, eliminating all reliance on third-party may be inadvisable, as it is difficult, if not impossible, for a small company like Democracy Live to deliver the same level of infrastructure security and resilience as a leading cloud provider.

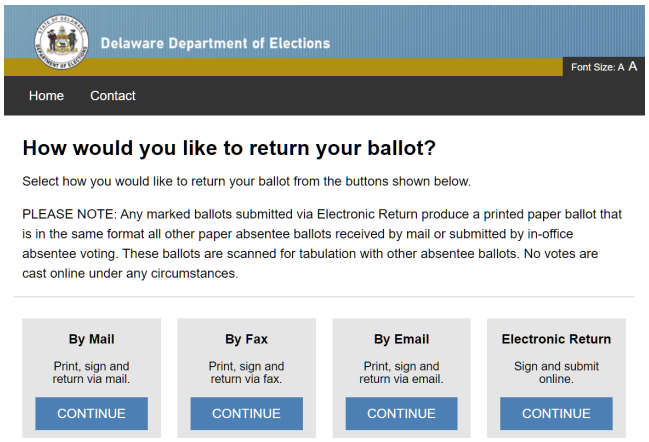


Figure 4: **Misleading statements about online voting.** The Delaware app stated that, "No votes are cast online under any circumstances." In fact, both email and electronic return cast the ballot over the Internet. Such mischaracterizations make it harder for voters to understand the risks of their selected return path.

Require a privacy policy. Despite having access to a wide range of sensitive personally identifiable information, OmniBallot had no privacy policy, leaving voters uniformed about what legal limitations, if any, restrict the company's use of this data. For example, it remains unclear whether the company could legally share such data with political campaigns, law enforcement, foreign governments, or ad tech companies. Moreover, due to OmniBallot's reliance on third-party services, Amazon and Google store or receive some or all of this data. Statutory requirements, Democracy Live's contracts with third parties, and contractual obligations to election jurisdictions may offer some legal protections, but these are largely invisible to voters.

At our recommendation, Democracy Live recently posted a privacy policy that covers all OmniBallot instances and prohibits the company from using voters' information for any purpose unrelated to servicing their ballots [18]. However, the policy does not provide explicit limits and guarantees about the retention, protection, and disposal of this data.

Increase transparency and facilitate independent review. Transparency and independent technical analysis are important for ensuring that election software is as secure as possible and for helping officials and the public understand the technology's risks. Yet Democracy Live and Delaware have made accurate public understanding of these risks more difficult through misleading statements as to whether OmniBallot is a form of online voting (e.g., Figure 4), and ours is the first public, independent security analysis of the software.

Unlike in-person voting equipment, which is tested by federally accredited labs for compliance with the EAC's Voluntary Voting System Guidelines [60], there are no federal standards or certification processes for platforms like OmniBallot. This means local and state officials are largely dependent on

the vendors themselves when assessing such products. Officials should insist that systems like OmniBallot be subjected to public examination by independent security experts before considering them for use. Such evaluation has exposed critical vulnerabilities in Internet voting systems in the past (e.g., [29, 65]), preventing flawed technologies from putting elections at risk. That OmniBallot has been used before without reported problems—predominately for small populations and for low-risk blank-ballot delivery—does not establish that it can be used safely for online voting or with large numbers of voters in high-stakes elections.

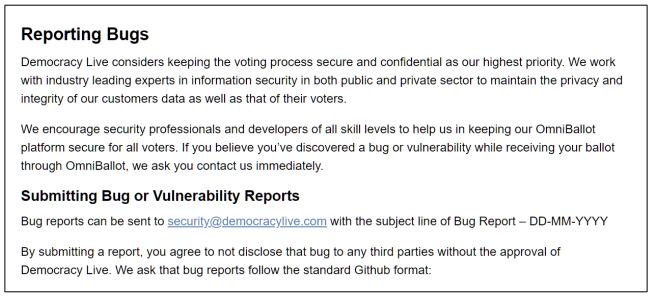
To facilitate independent analysis, we recommend that Democracy Live adopt a vulnerability disclosure policy that follows best practices, such as NTIA's CVD policy template [44], and make OmniBallot's source code available for scrutiny. The company's reporting guidelines at the time of our analysis (Fig. 5) prohibited further disclosure of reported problems without their permission. After we made our findings public, they adopted a new policy [14] modeled after Disclose.io's CVD template [19]. The new policy permits disclosure post-mitigation, but there are no set timelines nor any apparent recourse if the company excessively delays or chooses not to fix a problem. These policies may discourage responsible disclosure and could prevent researchers from alerting officials or the public about flaws that go unfixed.

It is notable that ours is the fourth security analysis of a deployed Internet voting system in less than year to find significant risks to election integrity [26, 29, 51]. In each of these cases, the researchers were presented with nontrivial barriers to analysis, ranging from incomplete documentation and lack of source code availability to restrictive vulnerability disclosure policies. This trend points to the possibility that current market incentives do not favor security or transparency for such systems. Our work should serve as further evidence to policymakers that regulatory intervention may be necessary.

6 Conclusions

Elections administrators have the complicated job of ensuring that all eligible voters have the ability to vote, while simultaneously safeguarding against some of the world's most sophisticated attackers. Some voters, including those with certain disabilities and some overseas servicemembers, have long faced significant obstacles to participation. Now, with the emergence of the COVID-19 pandemic, all voters may need better options for voting safely.

We find that OmniBallot's ballot delivery and marking modes have the potential to be valuable tools for helping voters participate, *if* used with specific precautions and changes. Blank ballot delivery, when used to print ballots, mark them by hand, and return them physically, appears to have only moderate risks if the precautions we recommend are applied, and it can cut in half the round-trip time of voting by mail. Online marking of vote-by-mail ballots is riskier, especially when widely used, and marking ballots server-side adds additional, unnecessary risks. However, with client-side marking and the procedural defenses we propose, the risks can be reduced to a level that may be acceptable for voters who otherwise could not mark a ballot independently. Our suggested changes would not impede accessibility and would result in greater protection for these voters.



Reporting Bugs

Democracy Live considers keeping the voting process secure and confidential as our highest priority. We work with industry leading experts in information security in both public and private sector to maintain the privacy and integrity of our customers data as well as that of their voters.

We encourage security professionals and developers of all skill levels to help us in keeping our OmniBallot platform secure for all voters. If you believe you've discovered a bug or vulnerability while receiving your ballot through OmniBallot, we ask you contact us immediately.

Submitting Bug or Vulnerability Reports

Bug reports can be sent to security@democracylive.com with the subject line of Bug Report – DD-MM-YYYY

By submitting a report, you agree to not disclose that bug to any third parties without the approval of Democracy Live. We ask that bug reports follow the standard Github format:

Figure 5: **Democracy Live's vulnerability reporting guidelines** stipulated that researchers who reported problems could not further disclose them without permission. Although it is unclear if this policy is enforceable, such restrictions run counter to best practices and may chill responsible disclosure.

Online ballot return, however, represents a severe danger to election integrity and voter privacy. At worst, attackers could change election outcomes without detection, and even if there was no attack, officials would have no way to prove that the results were accurate. No available technology can adequately mitigate these risks [41], so we urge jurisdictions not to deploy OmniBallot's online voting capabilities or similar systems.

In response to our findings, Delaware and New Jersey announced that they would halt use of OmniBallot [49] for online return, though Delaware continued allowing online voting using the Egress Switch webmail service, which is not necessarily more secure. Meanwhile, 19 states allow at least some voters to return ballots via email, fax, or a web portal [42], and many more offer online ballot delivery and marking using OmniBallot or similar products. There is an urgent need for further security scrutiny of these technologies to help officials assess the risks and to ensure that voters who need to participate remotely can do so as safely as possible.

Acknowledgements

We thank Andrew Appel, Matt Bernhard, Nakul Bajaj, Rachel Goodman, Susan Greenhalgh, David Jefferson, Ron Rivest, Jonathan Rudenberg, Andrew Sellars, Daniel Weitzner, and Aryana Ensafi Halderman for insightful feedback and other assistance. This material is based upon work supported by the National Science Foundation under Grant No. CNS-1518888, the Andrew Carnegie Fellows Program, Google's ASPIRE program, and MIT's Internet Policy Research Initiative.

References

[1] AAAS Center for Scientific Evidence in Public Issues. Letter to governors and secretaries of state on the insecurity of online voting, April 9, 2020. https://www.aaas.org/programs/epi-center/internet-voting-letter.

[2] B. Adida. Helios: Web-based open-audit voting. In *17th USENIX Security Symposium*, pages 335–348, 2008.

[3] D. Akhawe, F. Braun, F. Marier, and J. Weinberger. Subresource integrity, 2016. https://www.w3.org/TR/SRI/.

[4] Amazon Web Services. S3 Object Lock overview. https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lock-overview.html.

[5] Amazon Web Services. Share an object with others. https://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html.

[6] Amazon Web Services. What is AWS CloudTrail? https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html.

[7] A. W. Appel, R. A. DeMillo, and P. B. Stark. Ballot-marking devices (BMDs) cannot assure the will of the voters. *Election Law Journal*, 19(3), 2020.

[8] J. Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, Sept. 1987. https://www.microsoft.com/en-us/research/publication/verifiable-secret-ballot-elections/.

[9] M. Bernhard, J. Benaloh, J. A. Halderman, R. L. Rivest, P. Y. Ryan, P. B. Stark, V. Teague, P. L. Vora, and D. S. Wallach. Public evidence from secret ballots. In *2nd Intl. Joint Conf. on Electronic Voting*, E-Vote-ID, 2017.

[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? In *41st IEEE Symposium on Security and Privacy*, 2020.

[11] J. C. Carney. Sixth modification of the declaration of a state of emergency for the State of Delaware due to a public health threat, Mar. 2020. https://governor.delaware.gov/wp-content/uploads/sites/24/2020/03/Sixth-Modification-to-State-of-Emergency-03242020.pdf.

[12] S. Checkoway, J. Maskiewicz, C. Garman, J. Fried, S. Cohney, M. Green, N. Heninger, R.-P. Weinmann, E. Rescorla, and H. Shacham. A systematic analysis of the Juniper Dual EC incident. In *23rd ACM Conference on Computer and Communications Security*, CCS, 2016.

[13] C. Culnane, M. Eldridge, A. Essex, and V. Teague. Trust implications of DDoS protection in online elections. In *2nd Intl. Joint Conf. on Electronic Voting*, E-Vote-ID, 2017.

[14] Democracy Live. Disclosure policy. Accessed Oct. 10, 2020. https://democracylive.com/disclosure-policy/.

[15] Democracy Live. OmniBallot frequently asked questions. https://sites.omniballot.us/kcd/app/faq.

[16] Democracy Live. OmniBallot Online is an online sample ballot and electronic ballot system. https://democracylive.com/omniballot-online/.

[17] Democracy Live. Post election security audit: King Conservation District, Feb. 2020.

[18] Democracy Live. Privacy policy, June 15, 2020. https://democracylive.com/privacy-policy/.

[19] Disclose.io. Election CVD terms, June 15, 2020. https://github.com/disclose/terms/blob/master/vertical/core-terms-US-2020-ELECTIONS.md.

[20] E. Dreyfuss. Is your wobbly, illegible touchscreen signature still you? Wired, May 31, 2019. https://www.wired.com/story/is-your-wobbly-illegible-touchscreen-signature-still-you/.

[21] S. Dzieduszycka-Suinat, J. Murray, J. R. Kiniry, D. M. Zimmerman, D. Wagner, P. Robinson, A. Foltzer, and S. Morina. The future of voting: End-to-end verifiable Internet voting. U.S. Vote Foundation, 2015. https://www.usvotefoundation.org/E2E-VIV.

[22] P. Eckersley. How unique is your web browser? In *10th Privacy Enhancing Technologies Symposium*, PETS, 2010.

[23] E. Felten. The Linux backdoor attempt of 2003. Freedom to Tinker, 2013. https://freedom-to-tinker.com/2013/10/09/the-linux-backdoor-attempt-of-2003/.

[24] S. Gamard. Delaware presidential primary: Here's how to vote from home on June 2 due to coronavirus. Delaware News Journal, May 5, 2020. https://www.delawareonline.com/story/news/politics/2020/05/05/heres-how-vote-absentee-delaware-due-coronavirus/3048049001/.

[25] S. L. Garfinkel. Carving contiguous and fragmented files with fast object validation. *Digital Investigation*, 4:2–12, Sept. 2007.

[26] P. Gaudry. Breaking the encryption scheme of the Moscow Internet voting system. *arXiv preprint arXiv:1908.05127*, 2019. https://arxiv.org/pdf/1908.05127.pdf.

[27] E. Geller. Coronavirus boosts push for online voting despite security risks. Politico, May 1, 2020. https://www.politico.com/news/2020/05/01/coronavirus-online-voting-229690.

[28] Google. Google Analytics terms of service, June 2019. https://marketingplatform.google.com/about/analytics/terms/us/.

[29] T. Haines, S. J. Lewis, O. Pereira, and V. Teague. How not to prove your election outcome. In *41st IEEE Symposium on Security and Privacy*, 2020.

[30] J. A. Halderman. Practical attacks on real-world e-voting. In F. Hao and P. Y. A. Ryan, editors, *Real-World Electronic Voting: Design, Analysis and Deployment*, page 145–171, 2016.

[31] J. A. Halderman and V. Teague. The New South Wales iVote system: Security failures and verification flaws in a live online election. In *5th Intl. Joint Conf. on E-voting and Identity*, E-VoteID, 2015.

[32] N. Hastings, R. Peralta, S. Popoveniuc, and A. Regenscheid. Security considerations for remote electronic UOCAVA voting. National Institute of Standards and Technology, NISTIR 7770, 2011. https://www.nist.gov/system/files/documents/itl/vote/NISTIR-7700-feb2011.pdf.

[33] J. Kaya and J. Rickerd. Security researchers partner with Chrome to take down browser extension fraud network affecting millions of users. Duo Security Blog, 2020. https://duo.com/labs/research/crxcavator-malvertising-2020.

[34] B. Krebs. Browser extensions: Are they worth the risk? Krebs on Security, Sept. 18, 2018. https://krebsonsecurity.com/2018/09/browser-extensions-are-they-worth-the-risk/.

[35] E. Lielmanis. beautify-web/js-beautify. https://github.com/beautify-web/js-beautify.

[36] M. Lindeman and P. B. Stark. A gentle introduction to risk-limiting audits. *IEEE Security & Privacy*, 10(5):42–49, 2012.

[37] Microsoft Defending Democracy Program. ElectionGaurd, 2019. https://github.com/microsoft/electionguard.

[38] Mobile Voting Project. New Jersey announces accessible voting is coming to may elections, May 2020. https://mobilevoting.org/2020/05/new-jersey-announces-accessible-voting-is-coming-to-may-elections/.

[39] Mobile Voting Project. West Virginia expands online voting option in upcoming primary election for citizens with disabilities, Apr. 2020. https://mobilevoting.org/2020/04/west-virginia-expands-online-voting-option-in-upcoming-primary-election-for-citizens-with-disabilities/.

[40] L. Moore and N. Sawhney. Under the hood: The West Virginia mobile voting pilot, 2019. https://www.nass.org/sites/default/files/2019-02/white-paper-voatz-nass-winter19.pdf.

[41] National Academies of Sciences, Engineering, and Medicine. *Securing the Vote: Protecting American Democracy*. The National Academies Press, Washington, DC, 2018. https://www.nap.edu/catalog/25120/securing-the-vote-protecting-american-democracy.

[42] National Conference of State Legislatures. Electronic transmission of ballots, 2019. https://www.ncsl.org/research/elections-and-campaigns/internet-voting.aspx.

[43] National Cybersecurity Center. NCC King County audit summary 2020, March 3, 2020. https://cyber-center.org/ncc-king-county-audit-summary-2020/.

[44] NTIA Safety Working Group. "Early stage" coordinated vulnerability disclosure template, 2016. https://www.ntia.doc.gov/files/ntia/publications/ntia_vuln_disclosure_early_stage_template.pdf.

[45] M. Parks. States expand Internet voting experiments amid pandemic, raising security fears. NPR News, April 28, 2020. https://www.npr.org/2020/04/28/844581667/states-expand-internet-voting-experiments-amid-pandemic-raising-security-fears.

[46] R. L. Rivest. On the notion of 'software independence' in voting systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3759–3767, 2008.

[47] D. G. Robinson and J. A. Halderman. Ethical issues in e-voting security analysis. In *2nd Workshop on Ethics in Computer Security Research*, WECSR, 2011.

[48] D. E. Sanger. A botnet is taken down in an operation by Microsoft, not the government. The New York Times, March 10, 2020. https://www.nytimes.com/2020/03/10/us/politics/microsoft-botnets-malware.html.

[49] S. Schmidt. Delaware drops Internet-based voting system used by some absentee voters amid security concerns. Delaware Public Media, June 16, 2020. https://www.delawarepublic.org/post/delaware-drops-internet-based-voting-system-used-some-absentee-voters-amid-security-concerns.

[50] SLI Compliance. Democracy Live Secure Select 1.0 California certification security and telecommunications test report, 2017. https://votingsystems.cdn.sos.ca.gov/vendors/demlive/sli-dl-sectel.pdf.

[51] M. A. Specter, J. Koppel, and D. 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*, Aug. 2020.

[52] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman. Security analysis of the Estonian Internet voting system. In *21st ACM Conference on Computer and Communications Security*, CCS, 2014.

[53] T. Starks. States dabble with online voting. Politico, Apr. 30, 2020. https://www.politico.com/newsletters/morning-cybersecurity/2020/04/30/states-dabble-with-online-voting-787248.

[54] State of Delaware. Egress user guide for external users. https://elections.delaware.gov/information/pdfs/Egress%20Guide%20for%20External%20Users.pdf.

[55] Trail of Bits. Our full report on the Voatz mobile voting platform, Mar. 2020. https://blog.trailofbits.com/2020/03/13/our-full-report-on-the-voatz-mobile-voting-platform/.

[56] Trend Micro. Hacker infects Node.js package to steal from bitcoin wallets, Nov. 2018. https://www.trendmicro.com/vinfo/hk-en/security/news/cybercrime-and-digital-threats/hacker-infects-node-js-package-to-steal-from-bitcoin-wallets.

[57] U.S. Census Bureau. 2017 FIPS codes, 2017. https://www.census.gov/geographies/reference-files/2017/demo/popest/2017-fips.html.

[58] U.S. Cybersecurity and Infrastructure Security Agency. Risk management for electronic ballot delivery, marking, and return (draft). Published by The Guardian, May 2020. https://www.scribd.com/document/460491458/CISA-Guidelines-on-Internet-Voting.

[59] U.S. Election Assistance Commission. A survey of Internet voting, 2011. https://www.eac.gov/sites/default/files/eac_assets/1/28/SIV-FINAL.pdf.

[60] U.S. Election Assistance Commission Technical Guidelines Development Committee. Recommendations for requirements for the Voluntary Voting System Guidelines 2.0, Feb. 2020. https://www.eac.gov/sites/default/files/TestingCertification/2020_02_29_vvsg_2_draft_requirements.pdf.

[61] U.S. Senate Select Committee on Intelligence. Russian active measure campaigns and interference in the 2016 U.S. election, Volume 1: Russian efforts against election infrastructure, 2019. https://www.intelligence.senate.gov/sites/default/files/documents/Report_Volume1.pdf.

[62] Valve. Fingerprint.js. https://github.com/Valve/fingerprintjs2.

[63] D. Volz. Agencies warn states that Internet voting poses widespread security risks. The Wall Street Journal, May 8, 2020. https://www.wsj.com/articles/agencies-warn-states-that-internet-voting-poses-widespread-security-risks-11588975848.

[64] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[65] S. Wolchok, E. Wustrow, D. Isabel, and J. A. Halderman. Attacking the Washington, D.C. Internet voting system. In *16th Intl. Conf. on Financial Cryptography and Data Security*, FC, 2012.

[66] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi. Host fingerprinting and tracking on the web: Privacy and security implications. In *19th Network and Distributed System Security Symposium*, NDSS, 2012.

[67] K. Zetter. Google hack attack was ultra sophisticated, new details show. Wired, Jan. 14, 2010. https://www.wired.com/2010/01/operation-aurora/.