Ballot Acrobatics

Altering Electronic Ballots using Internal PDF Scripting

Henry D. Herrington
Advised by Jennifer Rexford

Submitted in Partial Fulfillment
Of the Requirements for the Degree of
Bachelor of Arts
Department of Computer Science
Princeton University

April 2022

Abstract

In recent decades, it has become increasingly common to transmit absentee ballots electronically as PDF files in American elections. Current PDF ballot policies provide certain voter groups, often military and overseas voters, with a more convenient alternative to the existing mail-based system of physical ballot transmission. This thesis offers an overview of the current role of PDF technology in voting procedures in the United States, and analyzes how the internal scripting capabilities of PDFs allow adversaries to manipulate PDF ballots and compromise election integrity. We conclude that, although they provide convenience and accessibility, PDF ballots as currently used in US elections jeopardize election security due to their vulnerability to scripting-based manipulation attacks. Future areas of research include analyzing the security of other data storage formats in the context of electronic ballots, investigating the security of electronic ballot transmission over networks, and developing new solutions for secure, accessible voting.

Acknowledgments

Thank you to my thesis advisor Jennifer Rexford for her countless thoughtful contributions to this project, and for her guidance throughout the process. Thank you also to Andrew Appel for initially suggesting the topic of PDF ballot scripting attacks, and for his many enlightening discussions on the subject. Thank you also to Arvind Narayanan, Kartikeya Kandula, Matthew Bernhard, and Michael A. Specter, for taking the time to meet with me and offer their expertise on both information security and online voting practices. And thank you to my parents, for everything.

Contents

1. Introduction

Secure elections are an integral part of healthy democracy. Not only do they allow voters to exercise political power, but they also establish respect for the legitimacy of government. Even minor voting security flaws have the potential to swing important elections in winner-take-all systems, and they can erode the basic trust between citizens and state that is needed for democracy to function.

While security is an important part of voting system design, accessibility and convenience are also desirable goals. Convenient voting procedures encourage democratic participation, and in turn, produce more representative election results. This goal of accessibility was the impetus for original absentee voting procedures in the United States, which were enacted during the Civil War to enfranchise soldiers who were unable to return to their local polling places [16]. Since then, many new voter groups have been granted eligibility to vote absentee across the US, and have traditionally done so through the postal system.

In recent decades, the internet has been used as a new tool to make voting more convenient. Many states now allow certain voter groups to receive and send their absentee ballots electronically as PDFs, as an alternative to physical ballots sent through the mail. This thesis will investigate how this pursuit of convenience in the American voting system can jeopardize election security. Specifically, it will explore how the internal scripting capabilities of PDFs make them an insecure file format for electronic ballots.

I will document the current uses of PDF ballots in US elections, describe and demonstrate several PDF ballot attacks, and then offer conclusions on PDF ballot usage given these findings. While PDFs are certainly a viable medium for communication in many instances, this paper may caution excitement about their use in the most critical aspects of the American election system.

2. Related Work

Prior research has analyzed the risks associated with different aspects of online voting, but there has been little discussion of how PDF scripting presents a danger in this space.

One category of scholarship surrounding online voting security has investigated specific online voting products, such as Voatz and Democracy Live's Omniballot [27, 26]. This research, however, has largely focused on the high-level software design of these applications that enables malicious insiders, third-party software vendors, and network adversaries to view and manipulate ballots. It does not analyze the risks inherent in PDF ballots that exist independently from these applications.

Other studies on online voting security have specifically examined the security of the PDF file format for ballot usage, but have largely neglected to include scripting attacks in this discussion. A National Institute of Standards and Technology report on electronic ballot transmission best practices has identified that PDF scripting can validate voter input in blank ballots, but offers no analysis of the ways in which adversaries can use this technology to manipulate voter preferences [19]. Additionally, research papers such as "UnclearBallot" have explored how scanners can be corrupted by malicious drivers to generate manipulated PDF ballots [3]. Although related to PDF ballot vulnerabilities, this type of external attack does not investigate the exploitability of internal PDF scripting capabilities.

Finally, studies of generic PDF vulnerabilities have identified the dangers of scripting, but have not linked them to current PDF ballot usage. In their report "Insecure Features of PDF Documents", Müller, et al. mention that scripting-based PDF form manipulation could in theory be used in deceptive financial attacks, but they do not consider the use case of electronic ballots [17].

This thesis will draw upon work in each of these areas to offer an analysis of how PDF scripting specifically can be exploited to compromise election security.

3. Current Uses of PDFs in US Elections

3.1. PDF Usage Overview

The US has a highly decentralized election system in which voting laws are largely decided on the state level and elections are often administered by individual counties. The result is a diverse patchwork of election policy between states that determines which voter groups may use PDF voting materials in different ways. Despite this heterogeneity, in general, there are several critical interactions in most states between a voter and the government during an election that can involve PDF usage. These interactions are voter registration[1], absentee ballot request[2], and voting an absentee ballot [3]. In each of these instances, the government will usually distribute the relevant blank voting material to the voter, and the voter will fill it out and return it to the government. Figure 1 illustrates this model of government and voter interaction.

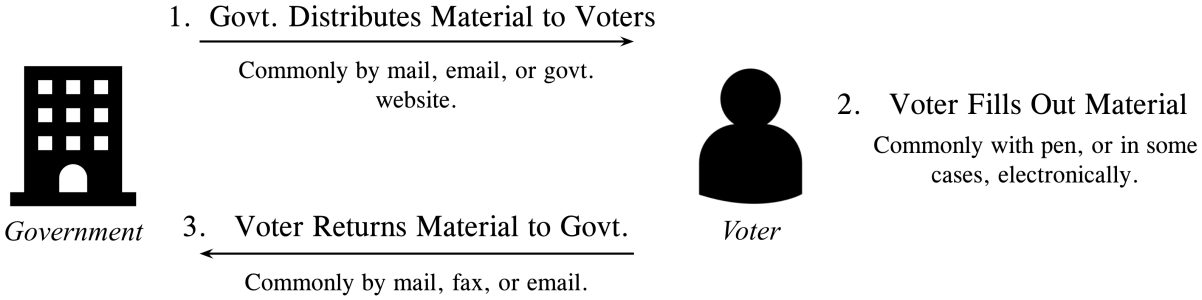


Figure 1: A general model of communication between government and voter. Material may refer to voter registration documents, ballot request forms, and absentee ballots. Descriptions provide a general picture of the election process, but are non-exhaustive and do not apply to all election material in all states [11, 12].

[1] North Dakota does not practice voter registration, and therefore does not distribute or collect voter registration material [20].

[2] A handful of states, like Washington, conduct all elections by mail. In these states, ballot request forms are generally not used, though similar forms, such as replacement ballot applications, may exist [30].

[3] Absentee voting is permitted for some voter group in all states [18].

These communications, when they occur, are always permitted in both directions via traditional mail. Depending on the state, voter group, and material sent, they can also happen electronically via email, fax, and website[4]. All of these electronic forms of communication have the potential to utilize PDF technology. Email communication almost always involves attaching voting material as a PDF, faxing communication can be done via online faxing services that transmit uploaded PDFs over fax, and web portals can permit PDF upload.

3.2. Why PDF Ballots are Uniquely Dangerous

To illustrate how PDF technology has seeped into American election procedures, I will present an overview of PDF usage for each of these types of voting material. However, this thesis will focus primarily on the vulnerabilities of PDF ballots, and not PDF voter registration forms or PDF absentee ballot request forms. This is because, due to their anonymous nature, hacked ballots are far more difficult to detect and correct, and therefore present a greater threat to election security.

The unique security puzzle surrounding ballots stems from their politically-desirable properties of anonymity. Specifically, voters should have no way of proving how they voted, and neither voters nor election officials should have the ability to connect submitted ballots to specific voters. A challenging result of these desired properties is that if an adversary manipulates a PDF ballot cleverly, a voter will have no way of knowing that their vote was tampered with, even after election results are made public.

Contrast this with attacks on other PDF election material, such as voter registration forms and absentee ballot request forms. The PDFs containing these materials will certainly be technically susceptible to the same types of attacks discussed in this thesis regarding PDF ballots. Attackers can manipulate the contents of these forms and, as a result, potentially

[4]There are also many other less common ways voting material might be distributed. To get a sense of the wide variety of voting policies across states, note that Oklahoma officially accepts ballot request forms by telegraph, in addition to mail, fax, email, and custom state web portal [23].

delay the delivery of voting material. For some voters, this could even prevent them from voting on time. However, in the long term, a voter will be able to detect these attacks when they don't receive their material, or in theory, if they ask the government to reproduce the information that they originally submitted. This ability to detect attacks allows voters to correct their actions and avoid similar attacks in the future. Given that PDF ballots by design do not allow for this sort of tampering detection and correction, they present the highest security risk and will be the main focus of this thesis.

3.3. PDF Usage for Voter Registration and Absentee Ballot Requests

PDFs can play a role in election procedures before absentee voters even receive their ballots. During the processes of voter registration and absentee ballot request, the relevant documents are often transmitted between government and voter in PDF format.

For distribution, state governments will almost always make blank voter registration and absentee ballot request forms available for download as PDFs on their websites. Table 3 in the Appendix contains a collection of PDF ballot request forms distributed by states in this way. Additionally, all citizens can use a federal voter registration form known as the National Voter Registration Application (NVRA), publicly available as a downloadable PDF [29]. Military and overseas voters also have the option to request an absentee ballot in any state using the Federal Post Card Application (FPCA), a federally-supplied generic absentee ballot request form [8]. The FPCA is an alternative to state-specific ballot request forms, and is also publicly available for download as a PDF. The majority of these non-ballot election material PDFs contain interactive form fields, which means that they can be directly marked electronically, as described in Section 4.2. Some ballot request forms, like those of Kansas and Texas, even utilize electronic form manipulation tools to help voters validate inputs and clear form fields. As we will see later, this programmatic form manipulation is the same technology adversaries can use to maliciously alter ballot materials.

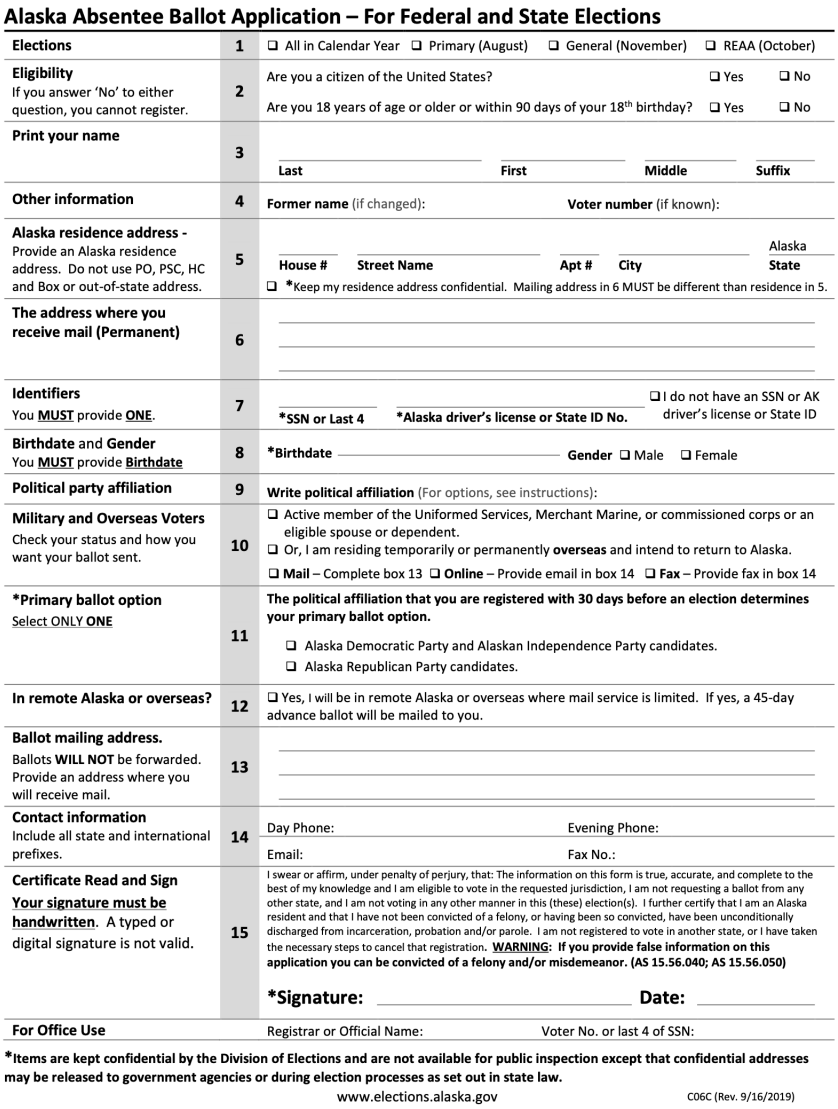
Voters are generally encouraged to download these forms onto their personal machines and fill them out either on paper after printing, or digitally in a PDF reader. Most states have instructions on these materials that disallow completing these forms without having to print them out at some point. As illustrative examples, Alaska's ballot request form, pictured in Figure 2, warns that a "digital signature is not valid," and Idaho's voter registration form prompts voters explicitly to "print and sign this form" [28, 13]. However, many other forms are ambiguous in terms of if it is permissible to complete them digitally.

Once these forms are complete, it is common for voters to return them via mail, fax, or as attachments over email. For military and overseas voters, the 2009 Military and Overseas Voter Empowerment ("Move") Act requires that all states accept their voter registration and ballot request forms by at least one form of electronic communication [1]. Even for non-military-and-overseas voters, many states allow their ballot request forms to be returned by email and fax. For the specific state policies on non-military-and-overseas ballot request form submission methods, see Table 4 in the Appendix. If these election materials are returned as PDFs, they may exist as scanned versions of printed and signed forms, or potentially as native PDFs that have remained digitally intact.

3.4. PDF Usage for Absentee Ballots

Whereas many states allow all voters to receive and send their voter registration and ballot request material as PDFs, PDF usage for actual ballots is much more limited. No state authorizes all of their voters to receive or send ballots as PDFs. In fact, in many states, permission to use PDF ballots is limited to military and overseas voters. In this section, I will focus my description of PDF ballot usage on the policies pertaining to military and overseas voters, which are often the most permissive.

Starting with the distribution of blank PDF ballots, military and overseas voters are able to receive their absentee ballot as a PDF in all states. This is due to the Uniformed and



Alaska Absentee Ballot Application – For Federal and State Elections

Elections	1	❑ All in Calendar Year ❑ Primary (August) ❑ General (November) ❑ REAA (October)
Eligibility If you answer 'No' to either question, you cannot register.	2	Are you a citizen of the United States? ❑ Yes ❑ No Are you 18 years of age or older or within 90 days of your 18th birthday? ❑ Yes ❑ No
Print your name	3	**Last** **First** **Middle** **Suffix**
Other information	4	**Former name** (if changed): **Voter number** (if known):
Alaska residence address - Provide an Alaska residence address. Do not use PO, PSC, HC and Box or out-of-state address.	5	**House #** **Street Name** **Apt #** **City** Alaska **State** ❑ *Keep my residence address confidential. Mailing address in 6 MUST be different than residence in 5.
The address where you receive mail (Permanent)	6	
Identifiers You **MUST** provide **ONE**.	7	***SSN or Last 4** ***Alaska driver's license or State ID No.** ❑ I do not have an SSN or AK driver's license or State ID
Birthdate and **Gender** You **MUST** provide **Birthdate**	8	***Birthdate** **Gender** ❑ Male ❑ Female
Political party affiliation	9	**Write political affiliation** (For options, see instructions):
Military and Overseas Voters Check your status and how you want your ballot sent.	10	❑ Active member of the Uniformed Services, Merchant Marine, or commissioned corps or an eligible spouse or dependent. ❑ Or, I am residing temporarily or permanently **overseas** and intend to return to Alaska. ❑ **Mail** – Complete box 13 ❑ **Online** – Provide email in box 14 ❑ **Fax** – Provide fax in box 14
***Primary ballot option** Select ONLY **ONE**	11	**The political affiliation that you are registered with 30 days before an election determines your primary ballot option.** ❑ Alaska Democratic Party and Alaskan Independence Party candidates. ❑ Alaska Republican Party candidates.
In remote Alaska or overseas?	12	❑ Yes, I will be in remote Alaska or overseas where mail service is limited. If yes, a 45-day advance ballot will be mailed to you.
Ballot mailing address. Ballots **WILL NOT** be forwarded. Provide an address where you will receive mail.	13	
Contact information Include all state and international prefixes.	14	Day Phone: Evening Phone: Email: Fax No.:
Certificate Read and Sign **Your signature must be handwritten**. A typed or digital signature is not valid.	15	I swear or affirm, under penalty of perjury, that: The information on this form is true, accurate, and complete to the best of my knowledge and I am eligible to vote in the requested jurisdiction, I am not requesting a ballot from any other state, and I am not voting in any other manner in this (these) election(s). I further certify that I am an Alaska resident and that I have not been convicted of a felony, or having been so convicted, have been unconditionally discharged from incarceration, probation and/or parole. I am not registered to vote in another state, or I have taken the necessary steps to cancel that registration. **WARNING: If you provide false information on this application you can be convicted of a felony and/or misdemeanor. (AS 15.56.040; AS 15.56.050)** ***Signature:** **Date:**
For Office Use		Registrar or Official Name: Voter No. or last 4 of SSN:

***Items are kept confidential by the Division of Elections and are not available for public inspection except that confidential addresses may be released to government agencies or during election processes as set out in state law.**

www.elections.alaska.gov C06C (Rev. 9/16/2019)

Figure 2: A screenshot of Alaska's Absentee Ballot Application, available publicly as an interactive PDF form at https://www.elections.alaska.gov/doc/forms/C06C.pdf [28].

Overseas Citizens Absentee Voting Act (UOCAVA), enacted in 1986. UOCAVA allows all military and overseas voters to vote using the Federal Write-In Absentee Ballot (FWAB), which is a generic ballot made available as a PDF online [2, 9]. The FWAB is intended as a back-up resource if a specific election ballot doesn't arrive in time, but it will be accepted for federal elections in all states. Additionally, many states will accept the FWAB for state and local elections as well. The FWAB PDF contains interactive form fields that allow voters

to mark their ballot electronically, although voters are instructed to print the ballot before returning it.

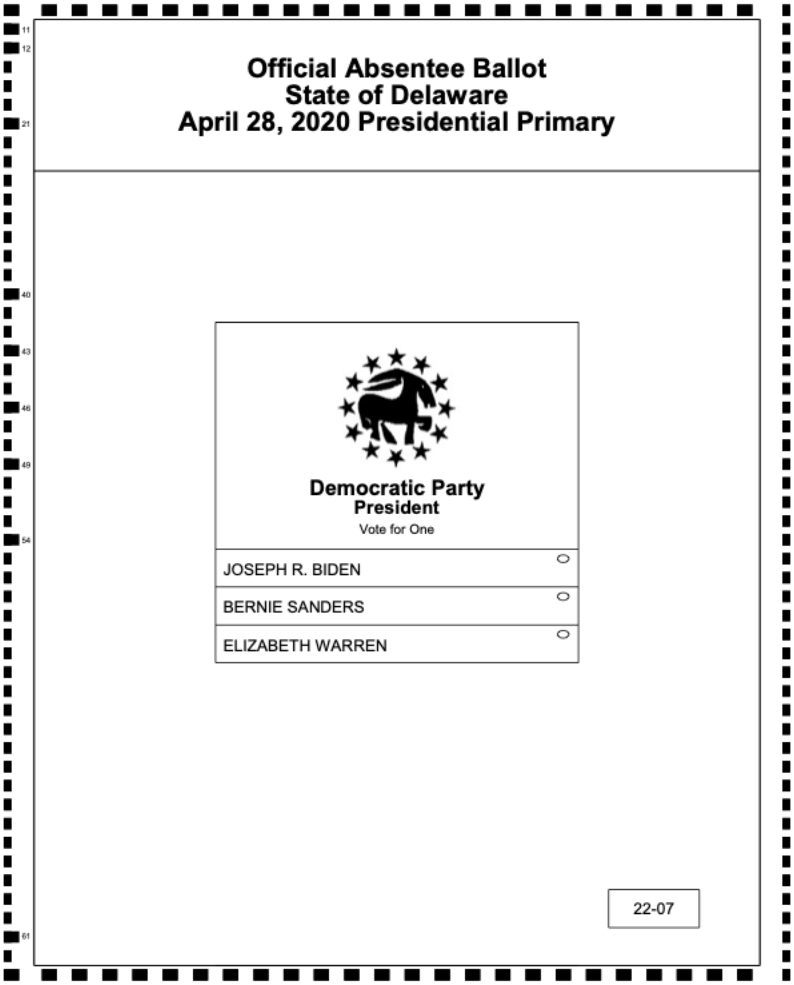


Figure 3: A screenshot of a Delaware PDF ballot used for the 2020 Presidential primaries [25].

In addition to this generic federal PDF ballot, election-specific ballots are also distributed as PDFs to military and overseas voters in many states. The aforementioned Move Act additionally requires that states establish at least one electronic method of transmitting blank absentee ballots to military and overseas voters [1]. This is often done by making blank PDF ballots available, either through a web portal or via email.

Generally, states are even more cautious about accepting absentee ballots as PDFs than they are about distributing them as PDFs. Nonetheless, the majority of states will allow military and overseas voters to return their ballot electronically as a PDF. To see which states specifically allow ballots to be returned as PDFs, please consult Table 5 in the Appendix.

Often, voters who are allowed to return PDF ballots are instructed to print their ballots and fill them out physically before scanning and returning them. However, in certain states, voters are permitted to mark and return their ballots without ever using a physical pen or paper. For example, Nevada's EASE system allows certain voters to receive and fill out their ballots electronically, and then download a PDF of their completed ballot to submit directly as an email attachment [22]. In Nevada, EASE is accessible for both military and overseas voters as well as voters with disabilities [22]. Similar commercial products exist that allow voters to receive blank ballots, mark them electronically, and, in some cases, download and submit their completed ballots as PDFs. One such product is Democracy Live's Omniballot, which has been used in elections in over 20 states [5, 6]. Before discussing the different security exploits possible in each of these PDF ballot contexts, I will provide a brief overview of PDF scripting.

4. Background on PDF Scripting

4.1. PDF Structure

PDF files have a sophisticated internal structure that allows for much more than just the presentation of text and images. PDF file code can be categorized into four ordered sections: a header, body, cross reference (xref) table, and trailer.

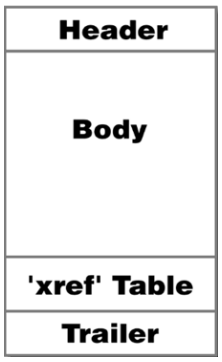


Figure 4: PDF file structure [15].

The header simply contains the PDF version, the body contains different representations of objects within the PDF, the xref table contains the locations of the body objects, and the trailer, which is commonly the first section read by PDF readers, points to the start of the xref table [14, 24]. With this flexible internal structure, PDFs possess capabilities such as the ability to play audio and video, open webpages, open system files, display 3-D content, run JavaScript code, and more. The scripting attacks in this thesis will focus on this ability of PDFs to run JavaScript code.

4.2. PDF Annotations and JavaScript

PDF annotations are a class of PDF objects including comments and form fields, that, in contrast to standard PDF text and images, are easily manipulated by anyone reading the

document. Such annotated form fields, also referred to as interactive form fields, include text boxes, dropdowns, checkboxes, radio buttons, and more. Notably, one type of form field, the button, can be displayed in the document as an uploaded image of custom size and position.

Additionally, the content and visibility of all PDF form fields can be manipulated programmatically using internal JavaScript. This JavaScript can be triggered by events such as clicking on a form field, but it can also be triggered by less noticeable events, such as opening the document, closing the document, and timed JavaScript intervals. Furthermore, certain PDF readers such as Adobe Acrobat allow JavaScript programs to refer to other JavaScript scripts, which enables a PDF script to modify and even remove itself and other scripts from the document. The ability of this internal JavaScript to manipulate the PDF contents and itself, all based on unobtrusive user input, will form the basis of many of the following attacks.

4.3. A Note on PDF Readers

As noted in Section 4.2, many of the attacks in this paper will rely on the capacity of PDFs to execute internal JavaScript code, which can only be accomplished while a PDF is opened in a PDF reader. Notably, the capabilities of PDF files are restricted to varying degrees depending on the PDF reader in which they are opened. Figure 5 illustrates the diversity of PDF privileges between different PDF readers, focusing particularly on potential insecurities.

This chart accurately reflects that two of the most popular PDF readers – Acrobat Reader DC and Acrobat Pro DC – support programmatic form modification. Since this chart was published, newer versions of readers like Google Chrome have also expanded functionality to support scripting-based form modification. To view the demonstrations in Section 7 and Section B correctly, please use an advanced and up-to-date PDF reader

Attack Category			DoS		Invasion of Privacy			Disclosure			Manipulation			RCE
Application	Version		Infinite loop	Deflate bomb	URL invocation	Evitable metadata	Revision recovery	Form data leakage	Local file leakage	Credential theft	Form modification	File write access	Content masking	Code execution
Acrobat Reader DC	(2019.008.20081)	Windows	●	●	●	–	–	○	○	○	●	○	○	○
Foxit Reader	(9.2.0.9297)		●	●	○	–	–	○	○	●	○	○	○	○
PDF-XChange Viewer	(2.5.322.9)		●	●	●	–	–	●	●	●	●	○	●	○
Perfect PDF Reader	(8.0.3.5)		●	●	●	–	–	●	○	●	○	○	○	○
PDF Studio Viewer	(2018.1.0)		○	●	●	–	–	●	◐	○	○	○	○	●
Nitro Reader	(5.5.9.2)		●	●	●	–	–	●	○	●	○	○	●	●
Acrobat Pro DC	(2017.011.30127)		●	●	●	○	◐	○	○	○	●	○	○	○
Foxit PhantomPDF	(9.5.0.20723)		●	●	○	●	○	○	○	●	○	○	○	○
PDF-XChange Editor	(7.0.326.1)		●	●	●	○	○	●	●	●	●	●	○	○
Perfect PDF Premium	(10.0.0.1)		●	●	●	●	○	●	○	●	○	○	○	○
PDF Studio Pro	(12.0.7)		○	●	●	○	○	●	◐	○	○	○	○	●
Nitro Pro	(12.2.0.228)		●	○	●	●	○	●	○	●	○	○	●	●
Nuance Power PDF	(3.0.0.17)		●	●	●	●	○	○	●	●	○	○	●	○
iSkysoft PDF Editor	(6.4.2.3521)		●	○	○	●	◐	○	○	○	○	○	○	○
Master PDF Editor	(5.1.36)		●	○	●	○	○	●	◐	●	○	●	○	○
Soda PDF Desktop	(11.0.16.2797)		●	●	●	●	○	○	○	●	○	○	●	○
PDF Architect	(7.0.23.3193)		●	●	●	●	○	○	○	●	○	○	●	○
PDFelement	(6.8.0.3523)		●	○	○	●	◐	○	○	○	○	○	○	○
Preview	(10.0.944.4)	Mac	●	●	○	–	–	○	○	○	○	○	○	○
Skim	(1.4.37)		●	●	○	–	–	○	○	○	○	○	○	○
Evince	(3.2.11)	Linux	○	●	○	–	–	○	○	○	○	○	○	◐
Okular	(0.26.1)		●	●	○	–	–	○	○	○	○	○	○	◐
MuPDF	(1.14.0)		●	○	○	–	–	○	○	○	○	○	○	◐
Chrome	(70.0.3538.67)	Web	◐	●	●	–	–	●	○	○	○	○	○	○
Firefox	(66.0.2)		◐	●	●	–	–	○	○	○	○	○	○	○
Safari	(11.0.3)		○	○	○	–	–	○	○	○	○	○	○	○
Opera	(57.0.3098.106)		◐	◐	●	–	–	●	○	○	○	○	○	○
Edge	(42.17134.1.0)		○	○	○	–	–	○	○	○	○	○	○	○

● Application vulnerable ◐ Vulnerability limited ○ Not vulnerable – No editing

Figure 5: The different levels of permission granted to PDF files based on PDF reader (chart from [17]).

like Adobe Acrobat or Google Chrome. Currently, the most up-to-date working versions of these applications are 2022.001.20112 (Acrobat Reader DC and Acrobat Pro DC), and 100.0.4896.127 (Google Chrome). Other readers, like Apple's Preview, restrict much PDF functionality by design, such as the execution of JavaScript code. As a result, these limited PDF readers will not correctly run the demonstrations, even if up-to-date.

5. Scripting Attacks on PDF Ballots

5.1. Overview of Attack Space

To organize this discussion of PDF ballot attacks, I will categorize PDF ballots using two parameters. These parameters are the status of the PDF file as either native or non-native, and the status of the ballot contents as either blank or marked. The resulting four types of PDF ballots and a summary of their corresponding vulnerabilities are displayed in Table 1 below.

	Blank Ballot	Marked Ballot
Native PDF	**Blank Native Ballot**: Candidates can be removed or rearranged.	**Marked Native Ballot**: Individual votes can be altered, and potentially captured.
Non-Native PDF	**Blank Non-Native Ballot**: Entire ballot can be overwritten to remove or rearrange candidates.	**Marked Non-Native Ballot**: Entire ballot can be overwritten to alter votes.

Table 1: An organization of the attacks possible on four different PDF ballot types, based on the PDF file's digital status and the ballot's completion status.

Before describing the attacks on these ballot types, I will define the terminology used for categorization. Native PDFs are PDFs whose content, such as text and graphics, has been generated digitally and not printed or scanned. Compared to non-native PDFs, the content of native PDFs is visually crisp and precise, even when zoomed in on. Figure 6 presents a visual comparison between native and non-native PDF content.

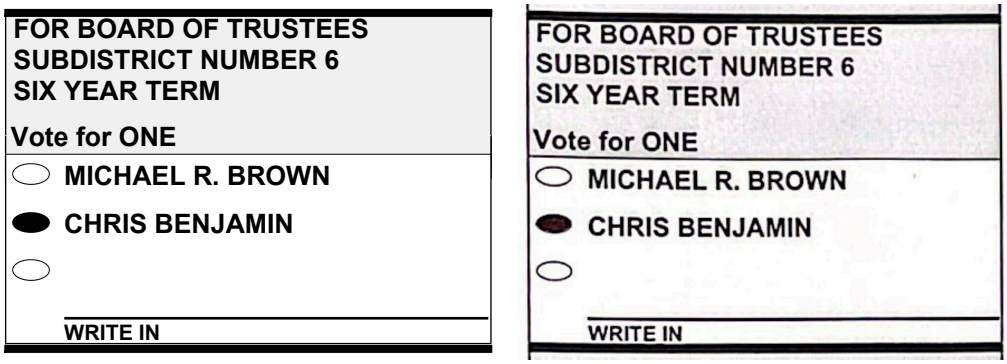


Figure 6: A visual comparison of native and non-native PDF elements. On the left, native elements from a sample PDF ballot. On the right, a scanned non-native version of the same ballot [4].

Furthermore, native PDFs are more likely to include interactive form fields, as described in Section 4.2. To contrast native PDFs, non-native PDFs are simply PDFs that do not contain digitally-generated content, such as photos and scanned images. Even native PDFs that are printed and then scanned are then considered non-native PDFs, because they've lost the original digital representation of their content. The difference between blank ballots and marked ballots is, more obviously, that blank ballots contain no indication of voter preference and marked ballots contain votes.

In the remainder of this section, I will describe the different attacks possible for each of these ballot types. For each quadrant, I will explain why the ballot type is vulnerable from a technical perspective, and referencing demonstrations, explain what an attack on the ballot type may look like in a real election. Finally, I will explore how evidence of attacks can be removed using self-modifying JavaScript code, which is a technique applicable in all ballot type quadrants.

5.2. Attacks on Marked Native Ballots

A marked native ballot, when hacked, is able to manipulate votes by covering existing voter preferences with form fields generated to visually mimic native ballot content. This allows these ballots to present one set of preferences to a voter and a separate set of altered preferences to election officials. First I will describe how these hacked ballots might enter an election, and then I will demonstrate several specific versions of this type of hacked ballot.

One avenue through which attackers might produce hacked marked native ballots is ballot-marking software like EASE or Omniballot (described in Section 3.4). If election hackers can compromise a voter's personal machine, this type of ballot creation software can be corrupted to produce hacked ballots instead of legitimate ones, without the voter's knowledge. Importantly, these attacks don't rely on the fact that the ballot-marking software was designed maliciously, although this should also be a concern. Simply by nature of operating on a voters' personal machine, this software is liable to become a vehicle for malicious code, even if it was not originally intended to produce corrupted ballots. Another potential avenue for attack is corrupted PDF readers, which in theory could inject malicious JavaScript into a native ballot as a voter or election official is viewing it electronically. It is important to note that internal PDF scripts have no way to access non-annotation content, so internal scripts that wish to hide and reveal specific ballot elements must have knowledge of the elements' location prior to being injected. Both ballot-marking software and PDF readers have access to this information, so hacking these systems will involve programmatically understanding the ballot layout so that the injected JavaScript can accurately place form fields. Because ballot-marking software has a built-in knowledge of ballot layout, it may be a more convenient attack vector for adversaries. Several specific examples of hacked ballots that could result from these attacks are described below, and the corresponding hacked PDF demonstrations are available in Section 7.

I will now describe the malicious code in Demonstration A, which contains a hacked marked native ballot that toggles its votes every minute. Unbeknownst to the voter, each ballot bubble in Demonstration A is actually represented by the button form field. Button form fields can take the appearance of an image, and in this demonstration, in the place of each ballot bubble, there are two buttons – one displaying an empty bubble image and another displaying a full bubble image. At any given time, one of these buttons is hidden and the other is visible. These form fields are named to reflect which of the four votes they count towards and which candidate within that vote they represent. For example, Emily Stone's button form fields are named "vote4_cd2_empty" and "vote4_cd2_full", because she is the second candidate in the fourth vote. As described in section 4.2, all form fields have a display property that can be set to hidden or visible, among other options, according to the fields of an Adobe-defined display object. So in Demonstration A, Emily Stone's ballot bubble could be filled with the following document-level JavaScript:

```
this.getField("vote4_cd2_empty").display = display.hidden;
this.getField("vote4_cd2_full").display = display.visible;
```

Listing 1: JavaScript code to programmatically mark a ballot bubble in a native marked ballot.

We have inserted a document-level JavaScript function called SelectBubble that, given a vote number and candidate number, essentially does the work of selecting that candidate and deselecting all other candidates in that vote.

The trick to altering votes in this seemingly static ballot, then, is just a matter of calling SelectBubble on non-user triggers. In this demonstration, the non-user trigger is a JavaScript interval that is started by a document-level script when the document is opened. Every second, this interval will run a function to check if the minute has changed, based on an instance of the JavaScript Date object. When the minute has changed, the

program will execute a function UpdateForm which in turn makes calls to SelectBubble with predetermined arguments based on the minute parity.

This demonstration alters the ballot contents every minute, but there are other conditions attackers could use to prompt attacks, such as time zone and date. These conditions are particularly dangerous in the context of absentee ballots, because if attackers use time zone to conditionally manipulate a ballot, that ballot may appear legitimate to a voter overseas, while displaying a fraudulent vote in the jurisdiction where it is counted. Furthermore, such a location-based attack could also be conditioned to only occur close to the election date. As examples, Demonstration B presents a ballot that displays a different set of voter preferences on the East Coast versus anywhere else in a non-Eastern time zone. Demonstration C presents a ballot that will only change its contents based on a future election date.

A final note on marked native ballots is that they in theory could be interactive ballots that allow voters to enter preferences directly into form fields. Generally, PDF ballots do not exist as interactive PDFs. This being said, the majority of ballot request form PDFs and voter registration form PDFs exist as interactive PDFs, as mentioned in Section 3.3 and Table 3. I present demonstrations of attacks on interactive PDFs not only because of these existing uses, but also to discourage any future efforts to create interactive PDF ballots. In non-interactive native marked ballots, form fields are injected as a tool to display phony information. But in interactive ballots, attackers can manipulate votes by simply interacting with these existing forms. Demonstrations A through C, which operate on non-interactive ballots, all apply to interactive ballots as well. By unchecking the "read only" option on all form fields, these ballots will become interactive and remain insecure in the same ways. Additionally, interactive ballots give attackers a way to capture existing votes, and not just overwrite votes as before. Demonstration E gives an example of how interactive ballots can be manipulated based on previously entered votes, and Demonstration F showcases how captured votes in an interactive ballot might be exfiltrated to adversaries. Specifically, in

Demonstration F, interactive form data is collected and then sent to an adversarial server as part of a URL query.

5.3. Attacks on Blank Native Ballots

Blank native ballots also present an opportunity for adversaries to deceive voters, but using a slightly different attack model. The majority of blank PDF ballots we found are native PDFs, and are often intended to be printed and filled out physically. In this case, these ballots can be attacked by using JavaScript and hidden form fields, as in Section 5.2, to remove or rearrange candidates before the voter fills out the ballot. Although rearranging candidates would not alter a ballot vote counted by a human, in jurisdictions that scan ballots and count votes based on bubble position, rearranging candidates could result in an incorrectly counted vote. The vehicle for this sort of attack could again involve a corrupted PDF reader that injects malicious code into PDF ballots, either on a voter's compromised machine or on an election official's compromised machine before being sent to voters. This malicious script could utilize forms to either alter specific ballot elements, or simply overwrite the entire legitimate blank ballot with a hidden layer containing an entirely new phony blank ballot.

5.4. Attacks on Marked Non-Native Ballots

Marked non-native ballots are more widely accepted than marked native ballots, as photos or scanned images of ballots filled out by hand. In this case, the attack vector could again be a corrupted PDF reader that a voter uses to inspect their marked ballot before submitting it electronically. Given the randomness inherent in scanned images, it would be even more difficult for corrupted PDF readers to convincingly place individual form fields to alter the ballot with precision. Unlike native PDFs, non-native PDFs introduce an additional uncertainty surrounding image conditions like light level, camera blur, and image rotation angle, that can make small-scale alterations difficult. The more realistic attack option, then,

is to completely overwrite the existing ballot by programmatically revealing an existing image layer that covers the entire document. An example of this attack is available in Demonstration D. This attack can be made more convincing at scale by altering the lighting, rotation angle, grain, and other features of the phony ballot using various annotation layers stored within the PDF file, all at different levels for different victims.

5.5. Attacks on Blank Non-Native Ballots

Much less common than blank native ballots are blank non-native ballots. The attack model in this scenario is similar to the model described for blank native ballots in Section 5.3, in that candidates might be removed or rearranged in a corrupted PDF reader prior to ballot marking. The difference however, as explained in Section 5.4 on marked non-native ballots, is that non-native ballots will almost always require completely overwriting the provided ballot, due to the randomness inherent in photos and scans.

5.6. Removing Evidence of Malicious Code

Upon initial inspection, a drawback of these PDF scripting attacks is that the embedded JavaScript is discoverable after the fact, which would constitute evidence of tampering if uncovered. However, the nature of JavaScript and the structure of PDF scripts makes it such that attackers have the option to programmatically add, modify, or remove scripts altogether at any point in time. Consider the code in Listing 2, for example. This script, which exists at the PDF document level, illustrates how an attacker could create a malicious script, run it, delete it, and then delete the program that just ran all of that in a few short lines.

```
function scriptA() {
    this.addScript("scriptB", "console.println('scriptB ran');");
    scriptB();
    this.removeScript("scriptB");
    this.removeScript("scriptA");
 }
```

Listing 2: An example of self-modifying code used to create, run, and delete scripts within a PDF document.

Such code can be combined with any of the previous internal JavaScript attacks to remove concrete evidence of malicious scripting. A demonstration of self-deleting code within a self-manipulating ballot can be found in Demonstration G.

6. External Attacks

This paper has focused on the internal JavaScript capabilities of PDFs as a vulnerability in ballot usage. The demonstrated attacks so far have relied on the realistic assumption that some machine, either a voter's or an election official's, is compromised during the election process. This assumption, which should be made when designing a secure election system, opens up a range of other attacks possible on PDF ballots that do not rely on internal JavaScript, but instead manipulate PDF ballots externally. In addition to the aforementioned internal scripting attacks, an area of my ongoing experimentation is these external attacks.

Although this research is not as fully developed, I will briefly outline my external attack projects and how they might be extended. One specific type of external attack I have focused on is manipulating PDF ballot contents using computer vision. Specifically, I utilized the OpenCV package in Python to detect and fill ballot bubbles in a blank ballot.

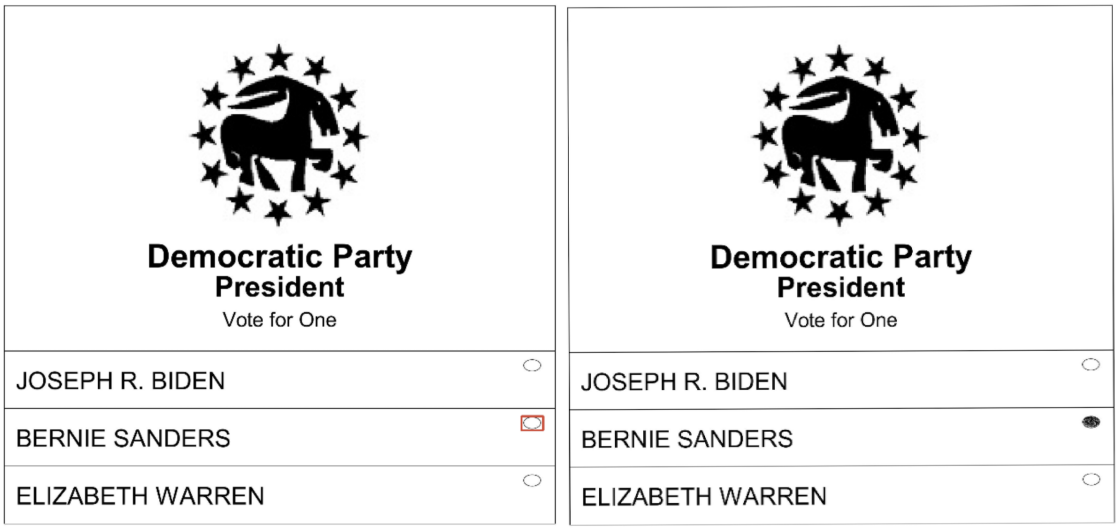


Figure 7: Side by side, the results of bubblefind.py (left) and bubblefill.py (right), both provided with template images for the "Sanders" candidate field, empty ballot bubble, and filled ballot bubble. On the left, the given ballot bubble is identified, and on the right, it is replaced with the image of a full ballot bubble [25].

Within the Demonstration H folder, bubblefind.py is able to locate a specific candidate's bubble within a ballot given a template image of that candidate field and empty bubble. As an extension, bubblefill.py will, additionally provided with an image of a filled bubble, mark a voter preference for the given candidate. Currently, these programs require that attackers possess an accurate reference image of the blank ballot, so that template images of bubbles and candidate fields can be provided for identification. In the future, the OpenCV library offers an opportunity to identify and mark ballot bubbles without template images. Specifically, the HoughCircles method included in the OpenCV library can be used to detect circles, which I made some progress extending into the detection of ballot bubble ellipses. If this work is continued, it offers an attractive attack model for adversaries to identify and manipulate ballots, without having to inject malicious internal JavaScript and deceptive form fields.

7. Demonstration Attacks on PDF Ballots

All demonstration ballots are available at https://github.com/henryprinceton/senior-thesis. Below, Table 2 contains a brief description of each demonstration. As mentioned in Section 4.2, the JavaScript capabilities of these PDFs require viewing them in an up-to-date and advanced PDF reader, such as Adobe Acrobat or the Google Chrome browser. Other readers, such as Apple Preview, will restrict the JavaScript capabilities of PDF files even if up-to-date, and accordingly, cannot run the demonstrations. Furthermore, PDF readers such as Adobe Acrobat Pro DC include tools for inspecting the full JavaScript used in each demonstration.

Demo	Description
A	Native marked ballot manipulation based on minute parity
B	Native marked ballot manipulation based on time zone
C	Native marked ballot manipulation based on set (election) date
D	Non-native marked ballot overwrite every five seconds
E	Native interactive form manipulation based on previously written content
F	Native interactive ballot data capture using server communication
G	Self-modifying JavaScript code within a PDF ballot
H	CV external modification of PDF ballot with template

Table 2: A description of each of the hacked PDF ballot demonstrations, all available in the linked GitHub repository.

Demo A

In even minutes, the ballot will vote for Michael R. Brown, Bryan Cribbs, Emily Stone, and "No" on the middle proposition (Voter Set 1), and in odd minutes, it will vote for Chris Benjamin, Ryan P. Miller, Jenny Wagoner, and "Yes" on the middle proposition (Voter Set 2).

Demo B

If viewed in Eastern Time, this ballot will vote for Voter Set 2 as opposed to Voter Set 1.

Demo C

Starting on April 4, 2023, the ballot will vote for Voter Set 2 as opposed to Voter Set 1.

Demo D

Every five seconds, the visibility of a fraudulent scanned ballot, displaying Voter Set 2, will toggle to either overwrite or reveal an underlying legitimate scanned ballot, displaying Voter Set 1.

Demo E

Every time the document is opened, it will vote for the first candidate in each vote unless the previous voter cast their vote for that candidate. In this case, the second candidate for that vote will be selected.

Demo F

Every time a bubble is selected or deselected, this PDF will check to see if all votes have a candidate marked. If so, it will exfiltrate ballot data to an adversarial server. The JavaScript accomplishes this by collecting the candidate number indicated in each vote and compiling it into a URL query string. The JavaScript then automatically launches a webpage using a URL containing the dynamically compiled query string. This can be tested using the provided server files in the GitHub repository in conjunction with the PDF ballot. Adobe Acrobat will warn users that the document would like to visit an unknown webpage, but this warning will not appear for previously visited pages or, presumably, in less-secure PDF readers.

Demo G

Ten seconds after opening this ballot, it will switch from Voter Set 1 to Voter Set 2. However, an inspection of the JavaScript code, achievable through a PDF reader like Adobe Acrobat Pro DC, will reveal no traces of JavaScript after the switch.

Demo H

This folder contains several files used in the experimentation of detecting ballot bubbles with OpenCV. The file bubblefind.py will detect the bubble corresponding to Bernie Sanders, and the file bubblefill.py will similarly fill in this bubble.

8. Conclusion

8.1. Summary

In the domain of voting security and ballot technology, PDF files possess often overlooked capabilities that make them inherently at-risk for manipulation attacks. Although online voting might increase accessibility and voter turn out, it also makes the election process more dependent on the security of voters' and election officials' machines. If a voter is using once-legitimate ballot marking software that has been hacked, or if an election official or voter is using a corrupted PDF reader, hacked PDF ballots with malicious JavaScript code are liable to enter into the system, compromising the integrity of the election. It is particularly difficult to trust that third-party ballot marking software such as Omniballot is sufficiently protected against potential corrupting attacks, considering that many of these software manufacturers don't disclose their software to the public. Because it is infeasible to expect all voters and election officials to maintain secure machines and secure software, it is advisable to minimize the usage of PDF ballots in official elections.

8.2. Communicating Findings

Because this thesis explores the intersection of computer science and politics, a goal of mine was to communicate the technical findings plainly, so that even non-computer scientists could understand the key takeaways. In the spirit of this effort, I also composed a condensed version of this thesis that is embedded with a working demonstration of a corrupted ballot. This article, intended for policymakers, voters, and other potentially non-technical readers, can be found in Section B of the Appendix, and additionally at https://github.com/henryprinceton/senior-thesis.

8.3. Future Work

As the world continues to digitize and streamline its societal processes, there will continue to be a lively discussion surrounding the viability of large-scale online voting in the US. This thesis has provided an analysis of one way in which current American online voting practices may be insecure, but it is hardly a comprehensive analysis of online voting security. Future research might analyze the security of other non-PDF data storage formats in the context of electronic ballots. Additionally, future work could continue identifying ways in which the transmission of electronic ballots might prove insecure in the field of network security. Research may also focus on inventing new technology that reimagines how accessible voting can be accomplished without sacrificing election security.

References

[1] 111th Congress, "Military and Overseas Voter Empowerment Act," 2009. [Online]. Available: https://www.fvap.gov/uploads/FVAP/Policies/moveact.pdf

[2] 99th Congress, "Uniformed and Overseas Citizens Absentee Voting Act," 1986. [Online]. Available: https://www.fvap.gov/uploads/FVAP/Policies/uocavalaw.pdf

[3] M. Bernhard *et al.*, "UnclearBallot: Automated Ballot Image Manipulation," in *Electronic Voting*. Berlin, Heidelberg: Springer-Verlag, p. 14–31. Available: https://doi.org/10.1007/978-3-030-30625-0_2

[4] Cass County, Missouri, "April 2022 Sample Ballot," 2022. Available: https://www.casscounty.com/DocumentCenter/View/2969/April-Sample-Ballot

[5] Democracy Live, "OmniBallot Portal." Available: https://democracylive.com/omniballot-portal/

[6] Democracy Live, *OmniBallot Paper Return*, Oct 2020. Available: https://www.youtube.com/watch?v=NFz5asVo5Qo

[7] District of Columbia Board of Elections, "Casting your Vote: Military and Overseas Voters (UOCAVA)." Available: https://www.dcboe.org/Voters/Casting-Your-Vote/Military-and-Overseas-Voters

[8] Federal Voting Assistance Program, "Federal Post Card Application (FPCA)." Available: https://www.fvap.gov/uploads/FVAP/Forms/fpca2013.pdf

[9] Federal Voting Assistance Program, "Federal Write-In Absentee Ballot (FWAB)." Available: https://www.fvap.gov/uploads/FVAP/Forms/fwab2013.pdf

[10] Federal Voting Assistance Program, "Voting Assistance Guide." Available: https://www.fvap.gov/guide/chapter2

[11] FlatIcon, *Building*. Available: https://www.flaticon.com/premium-icon/building_795521

[12] FlatIcon, *User Black Close Up Shape*. Available: https://www.flaticon.com/free-icon/user-black-close-up-shape_32438

[13] Idaho State Government, "Idaho Absentee Ballot Application 2022 Calendar Year." Available: https://sos.idaho.gov/elections/forms/absentee_ballot_request.pdf

[14] Infosec Institute, "PDF File Format: Basic Structure [Updated 2020]." Available: https://resources.infosecinstitute.com/topic/pdf-file-format-basic-structure/

[15] D. Lukan, *PDF Structure*. Infosec Institute. Available: https://resources.infosecinstitute.com/topic/pdf-file-format-basic-structure/

[16] MIT Election Data + Science Lab, "Voting by mail and absentee voting," Mar 2021. Available: https://electionlab.mit.edu/research/voting-mail-and-absentee-voting

[17] J. Müller *et al.*, *Vulnerability Report: Insecure Features of PDF Documents*. Ruhr-Universität Bochum and Horst-Görtz-Institut. Available: https://pdf-insecurity.org/download/pdf-dangerous-paths/PDF_Features_Disclosure.pdf

[18] National Conference of State Legislatures, "Voting Outside the Polling Place: Absentee, All-Mail and other Voting at Home Options," Mar 2022. Available: https://www.ncsl.org/research/elections-and-campaigns/absentee-and-early-voting.aspx

[19] National Institute of Standards and Technology, "Security Best Practices for the Electronic Transmission of Election Materials for UOCAVA Voters," U.S. Department of Commerce, Washington, D.C., Tech. Rep. NISTIR 7711, 2011. Available: https://csrc.nist.gov/publications/detail/nistir/7711/final

[20] North Dakota State Government, "When do eligible voters register for an election?" Available: https://vip.sos.nd.gov/PortalListDetails.aspx?ptlhPKID=79&ptlPKID=7

[21] Office of the Missouri Secretary of State, "Military and Overseas Voting Access Portal." Available: https://www.sos.mo.gov/elections/goVoteMissouri/registeroverseas

[22] Office of the Nevada Secretary of State, "EASE - Overview." Available: https://www.nvsos.gov/sos/elections/voters/uniformed-overseas-citizens/ease-overview

[23] Oklahoma State Election Board, "Standard Absentee Voters." Available: https://oklahoma.gov/elections/voters/absentee-voting/standard-absentee-voters.html

[24] L. Rosenthol, "Intro to PDF - Leonard Rosenthol," Dec 2015. Available: https://www.youtube.com/watch?v=KmP7pbcAl-8.

[25] M. A. Specter, *Official Absentee Ballot State of Delaware April 28, 2020 Presidential Primary.*

[26] M. A. Specter and J. A. Halderman, "Security Analysis of the Democracy Live Online Voting System," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 3077–3092. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/specter-security

[27] 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 (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 1535–1553. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/specter

[28] State of Alaska Division of Elections, "Alaska Absentee Ballot Application – for Federal and State Elections." Available: https://www.elections.alaska.gov/doc/forms/C06C.pdf

[29] U.S. Election Assistance Commission, "National Voter Registration Application (NVRA)." Available: https://www.eac.gov/sites/default/files/eac_assets/1/6/Federal_Voter_Registration_ENG.pdf

[30] Washington Secretary of State, "Frequently Asked Questions on Voting by Mail." Available: https://www.sos.wa.gov/elections/faq_vote_by_mail.aspx

A. Election Material Transmission Policy Tables

Blank Ballot Request Form PDFs				
State	BRF Link	Interactive	Comment	Source
Alabama	AL BRF	No	-	AL Source
Alaska	AK BRF	Yes	"Digital signature is not valid"	AK Source
Arizona	AZ BRF	No	-	AZ Source
Arkansas	AR BRF	No	-	AR Source
California	CA BRF	No	Form for replacement absentee ballot	CA Source
Colorado	CO BRF	Yes	"Scan the signed form"	CO Source
Connecticut	CT BRF	Yes	-	CT Source
Delaware	DE BRF	Yes	-	DE Source
Florida	-	-	No formal template provided	FL Source
Georgia	GA BRF	No	"No electronic signatures allowed"	GA Source
Hawaii	HI BRF	Yes	"Please print clearly in black ink"	HI Source
Idaho	ID BRF	Yes	"Please print and sign this form"	ID Source
Illinois	IL BRF	No	-	IL Source
Indiana	IN BRF	No	-	IN Source
Iowa	IA BRF	Yes	-	IA Source
Kansas	KS BRF	Yes	Uses form action to clear forms	KS Source
Kentucky	-	-	Not publicly available	KY Source
Louisiana	LA BRF	Yes	-	LA Source
Maine	ME BRF	Yes	-	ME Source
Maryland	MD BRF	Yes	"No electronic signatures allowed"	MD Source
Massachusetts	MA BRF	Yes	-	MA Source
Michigan	MI BRF	Yes	"Take a picture of the form"	MI Source
Minnesota	MN BRF	Yes	"May be returned ... scanned"	MN Source
Mississippi	-	-	Not publicly available	MS Source
Missouri	MO BRF	Yes	-	MO Source
Montana	MT BRF	No	-	MT Source
Nebraska	NE BRF	Yes	"Must be physically signed"	NE Source
Nevada	NV BRF	-	Ballots mailed to all, no BRFs	NV Source
New Hampshire	NH BRF	Yes	-	NH Source
New Jersey	NJ BRF	No	"Print and sign your name"	NJ Source
New Mexico	NM BRF	No	-	NM Source
New York	NY BRF	Yes	-	NY Source
North Carolina	NC BRF	Yes	"No electronic signatures allowed"	NC Source
North Dakota	ND BRF	Yes	-	ND Source
Ohio	OH BRF	Yes	-	OH Source
Oklahoma	OK BRF	No	-	OK Source

Oregon	OR BRF	Yes	-	OR Source
Pennsylvania	PA BRF	Yes	-	PA Source
Rhode Island	RI BRF	Yes	-	RI Source
South Carolina	-	-	Not publicly available	SC Source
South Dakota	SD BRF	Yes	-	SD Source
Tennessee	TN BRF	Yes	"Digital Signature Not Accepted"	TN Source
Texas	TX BRF	Yes	Uses form action to clear forms	TX Source
Utah	-	-	Ballots mailed to all, no BRFs	UT Source
Vermont	VT BRF	Yes	-	VT Source
Virginia	VA BRF	No	-	VA Source
Washington	WA BRF	No	Form for special ballot requests	WA Source
West Virginia	WV BRF	No	-	WV Source
Wisconsin	-	No	Not publicly available	WI Source
Wyoming	WY BRF	Yes	-	WY Source

Table 3: This table documents the different state-specific ballot request forms made available on state websites. The interactive column refers to the use of interactive form fields within the PDF. In some cases, BRFs are taken from individual counties if state-wide forms are not provided. Additionally, in some cases where BRFs are not used, return methods for similar resources, such as ballot replacement forms, are displayed.

Non-Military-and-Overseas Ballot Request Form Submission Methods					
State	Mail	Email	Fax	Comment	Source
Alabama	Yes	No	No	-	AL Source
Alaska	Yes	Yes	Yes	-	AK Source
Arizona	Yes	Yes	Yes	-	AZ Source
Arkansas	Yes	Yes	Yes	-	AR Source
California	Yes	*u*	*u*	Form for replacement absentee ballot	CA Source
Colorado	Yes	No	Yes	-	CO Source
Connecticut	Yes	No	Yes*	*Mailed original ballot also required	CT Source
Delaware	Yes	Yes	Yes	-	DE Source
Florida	Yes	Yes	Yes	-	FL Source
Georgia	Yes	Yes	Yes	-	GA Source
Hawaii	Yes	*u*	*u*	-	HI Source
Idaho	Yes	No	No	-	ID Source
Illinois	Yes	*u*	*u*	-	IL Source
Indiana	Yes	Yes	Yes	-	IN Source
Iowa	Yes	No	No	-	IA Source
Kansas	Yes	*u*	*u*	-	KS Source
Kentucky	Yes	-	-	Not publicly available	KY Source
Louisiana	Yes	No	Yes	-	LA Source
Maine	Yes	*u*	*u*	-	ME Source
Maryland	Yes	Yes	Yes	-	MD Source
Massachusetts	Yes	Yes	Yes	-	MA Source
Michigan	Yes	Yes	No	-	MI Source
Minnesota	Yes	Yes	Yes	-	MN Source
Mississippi	Yes	No	No	-	MS Source
Missouri	Yes	No	Yes	-	MO Source
Montana	Yes	No	No	-	MT Source
Nebraska	Yes	Yes	Yes	-	NE Source
Nevada	-	-	-	No ballot request form	NV Source
New Hampshire	Yes	Yes	Yes	-	NH Source
New Jersey	Yes	No	No	-	NJ Source
New Mexico	Yes	No	No	-	NM Source
New York	Yes	No	No	-	NY Source
North Carolina	Yes	No	No	-	NC Source
North Dakota	Yes	*u*	*u*	-	ND Source
Ohio	Yes	No	No	-	OH Source
Oklahoma	Yes	Yes	Yes	-	OK Source
Oregon	Yes	No	No	-	OR Source
Pennsylvania	Yes	No	No	-	PA Source

Rhode Island	Yes	No	No	-	RI Source
South Carolina	Yes	Yes	Yes	-	SC Source
South Dakota	Yes	No	No	-	SD Source
Tennessee	Yes	Yes	Yes	-	TN Source
Texas	Yes	Yes*	Yes*	*Mailed original ballot also required	TX Source
Utah	-	-	-	No ballot request form	UT Source
Vermont	Yes	Yes	No	-	VT Source
Virginia	Yes	Yes	Yes	-	VA Source
Washington	Yes	*u*	*u*	Form for special ballot requests	WA Source
West Virginia	Yes	Yes	Yes	-	WV Source
Wisconsin	-	-	-	Not publicly available	WI Source
Wyoming	Yes	Yes	No	-	WY Source

Table 4: This table documents the different methods by which non-military-and-overseas ballot request forms are accepted by the government after being filled in. Military and overseas voters, under the 2009 Move Act, always have the right to return their BRFs electronically [1]. The symbol *u* indicates unspecified. This is table is non-exhaustive in terms of submission methods, and many states offer other methods of submitting absentee ballot requests that are not listed. Additionally, in some cases where BRFs are not used, return methods for similar resources, such as ballot replacement forms, are displayed.

State PDF Ballot Submission Policies					
State	Mail	Email	Fax	Web Portal	Email, Fax, or Web Portal
Alabama	Yes	No	Yes	Yes	Yes
Alaska	Yes	No	Yes	No	Yes
Arizona	Yes	Yes	Yes	Yes	Yes
Arkansas	Yes	No	No	No	No
California	Yes	No	Yes	No	Yes
Colorado	Yes	Yes	Yes	Yes	Yes
Connecticut	Yes	No	No	No	No
Delaware	Yes	Yes	Yes	No	Yes
Florida	Yes	No	Yes	No	Yes
Georgia	Yes	No	No	No	No
Hawaii	Yes	Yes	Yes	No	Yes
Idaho	Yes	No	No	No	No
Illinois	Yes	No	No	No	No
Indiana	Yes	Yes	Yes	No	Yes
Iowa	Yes	Yes	Yes	No	Yes
Kansas	Yes	Yes	Yes	No	Yes
Kentucky	Yes	No	No	No	No
Louisiana	Yes	No	Yes	No	Yes
Maine	Yes	Yes	Yes	No	Yes
Maryland	Yes	No	No	No	No
Massachusetts	Yes	Yes	Yes	No	Yes
Michigan	Yes	No	No	No	No
Minnesota	Yes	No	No	No	No
Mississippi	Yes	Yes	Yes	No	Yes
Missouri	Yes	Yes	Yes	Yes	Yes
Montana	Yes	Yes	Yes	No	Yes
Nebraska	Yes	Yes	Yes	No	Yes
Nevada	Yes	Yes	Yes	Yes	Yes
New Hampshire	Yes	No	No	No	No
New Jersey	Yes	No	No	No	No
New Mexico	Yes	Yes	Yes	No	Yes
New York	Yes	No	No	No	No
North Carolina	Yes	Yes	Yes	Yes	Yes
North Dakota	Yes	Yes	Yes	Yes	Yes
Ohio	Yes	No	No	No	No
Oklahoma	Yes	No	Yes	No	Yes
Oregon	Yes	Yes	Yes	No	Yes

Pennsylvania	Yes	No	No	No	No
Rhode Island	Yes	Yes	Yes	No	Yes
South Carolina	Yes	Yes	Yes	No	Yes
South Dakota	Yes	No	No	No	No
Tennessee	Yes	No	No	No	No
Texas	Yes	No	Yes	No	Yes
Utah	Yes	Yes	Yes	No	Yes
Vermont	Yes	No	No	No	No
Virginia	Yes	No	No	No	No
Washington	Yes	Yes	Yes	No	Yes
West Virginia	Yes	Yes	Yes	Yes	Yes
Wisconsin	Yes	No	No	No	No
Wyoming	Yes	No	No	No	No
Totals	**50**	**23**	**30**	**8**	**30**

Table 5: This table documents the different methods by which marked ballots are accepted by the government in PDF form, specifically from military and overseas voters. Instances in which ballots can be emailed or faxed but also must be mailed are marked as "No" for email and fax. Totals refer to the numbers of "Yes" policies. Data compiled from the directory of the Federal Voting Assistance Program [10, 7, 21].

B. Condensed Thesis Article

The following is a condensed version of this thesis, including a working demonstration of a hacked ballot. Available independently at https://github.com/henryprinceton/senior-thesis.

ALTERING ELECTRONIC BALLOTS USING PDF SCRIPTING

Henry D. Herrington
Princeton University
henryherrington@gmail.com

April 7, 2022

I. INTRODUCTION

In recent decades, it has become increasingly common to transmit absentee ballots electronically as PDFs in American elections. From a security perspective, PDF ballots may seem analogous to physical paper ballots because both are used to display voter preferences. However, PDFs have many additional capabilities, including the ability to run JavaScript code, that make them insecure for ballot usage. In this article, I will explain current PDF ballot usage in the US, and then demonstrate why PDF is an insecure file format in this context.

Current PDF Ballot Usage in the US

All states permit some form of absentee voting, but laws on which voters are eligible to receive and return electronic ballots vary from state to state [7]. Military and overseas voters, for example, can receive their ballots electronically in all states, and are permitted to return their ballots as PDF email attachments in over half of all states [1], [5]. In addition to the electronic distribution and submission of ballots, some states also provide electronic means of marking ballots, such as Nevada's EASE system. This service allows certain voters to receive and fill out their ballots electronically, and then download a PDF of their completed ballot to submit as an email attachment [8]. In Nevada, EASE is accessible for both military and overseas voters as well as voters with disabilities [8]. Similar commercial products exist that allow voters to receive and mark their ballots electronically, and then download their completed ballots to submit as PDFs. One such product is Democracy Live's Omniballot, which has been used in elections in over 20 states [3], [4].

PDF Capabilities

Although there are clear accessibility and convenience benefits associated with PDF ballots, sometimes less apparent are the security tradeoffs of this technology. Unlike paper, PDF files have many additional capabilities outside of displaying images that make them more exploitable as ballots. PDFs can, for example, play video and audio, open webpages, launch other files, and, most relevant to this article, execute internal JavaScript code. I will now demonstrate how this scripting capability of PDFs makes them insecure for ballot usage.

II. DEMONSTRATION

The following page contains a demonstration PDF ballot that was adapted from an existing sample ballot from Cass County, Missouri [2]. The ballot has already been marked electronically, and could easily be the downloaded product of some ballot marking software. At this point, take note of which candidates are voted for. Once you do, you might feel comfortable closing this document and sending it off to election officials to have your vote counted. A non-technical voter may assume that once they stop interacting with this document, their vote is essentially secure. However, this ballot will actually alter the candidates it votes for based on the current time. In a real election, this ballot would be designed to alter its votes after the ballot is transmitted. For demonstration purposes, this ballot will conveniently flip its votes every minute. Specifically, in even minutes, it will vote for Michael R. Brown, Bryan Cribbs, Emily Stone, and "No" on the middle proposition, and in odd minutes, it will vote for Chris Benjamin, Ryan P. Miller, Jenny Wagoner, and "Yes" on the middle proposition. Importantly, this type of PDF ballot hacking can occur even if the voter uses a legitimate ballot marking software that was not originally designed to produce corrupted ballots. So long as an election hacker compromises a voter's machine, they can corrupt this software to generate hacked PDF ballots like the one demonstrated. Please see Section VII on troubleshooting if you encounter any difficulties with the demonstration.

OFFICIAL BALLOT

GENERAL MUNICIPAL ELECTION
CASS COUNTY, MISSOURI
TUESDAY, APRIL 5, 2022

NOTICE OF ELECTION

Notice is hereby given that the General Municipal Election will be held in the County of Cass on Tuesday, April 5, 2022 as certified to this office by the participating entities of Cass County. The ballot for the Election shall be in substantially the following form.

JUNIOR COLLEGE DISTRICT OF METROPOLITAN KANSAS CITY, MISSOURI

FOR BOARD OF TRUSTEES
SUBDISTRICT NUMBER 6
SIX YEAR TERM

Vote for ONE

- ○ **MICHAEL R. BROWN**
- ● **CHRIS BENJAMIN**
- ○ _____ **WRITE IN**

EAST LYNNE SCHOOL DISTRICT NO. 40

FOR BOARD MEMBER
THREE YEAR TERM

Vote for TWO

- ● **RYAN P. MILLER**
- ○ **BRYAN CRIBBS**
- ○ _____ **WRITE IN**
- ○ _____ **WRITE IN**

PROPOSITION STUDENTS, GROWTH, AND SAFETY

"Shall the Board of Education of the East Lynne School District No. 40, Missouri, borrow money in the amount of Five Hundred Thousand Dollars ($500,000) for the purpose of providing funds for site development, construction, equipping, and furnishing the reconfiguration of current spaces to address recent growth within the district, to replace and/or repair roofs; to implement safety and security improvements, including secure entrances; to the extent funds are available, complete other repairs and improvements to the existing facilities of the District; and issue general obligation bonds for the payment therof resulting in an estimated increase to the debt service property tax levy of $0.2400 per one hundred dollars of assessed valuation?

If this proposition is approved, the adjusted debt service levy of the School District is estimated to increase from $0.0000 to $0.2400 per one hundred dollars of assessed valuation of real and personal property."

- ● **YES**
- ○ **NO**

HARRISONVILLE R-IX SCHOOL DISTRICT

QUESTION:
To choose by ballot two (2) directors who shall serve as members of the Board of Education of said School District for a term of three (3) years each.

Vote for TWO

- ○ **BRITTNEY SEXTON**
- ○ **EMILY STONE**
- ● **JENNY WAGONER**
- ○ **DAVID W. REECE**
- ○ **DAVID PETERMAN**
- ○ _____ **WRITE IN**
- ○ _____ **WRITE IN**

III. TECHNICAL DETAILS

The basis of this vulnerability is the use of PDF form fields, which can be altered programmatically using internal PDF scripting. I now will describe the specific form fields and PDF scripts in this demonstration.

A. Form Fields

PDFs support interactive form fields ranging from text boxes, to dropdowns, to checkboxes, to radio buttons, and more. All of these form fields are manipulable using internal JavaScript. This demonstration uses the simple button form field to display custom ballot bubbles. Button form fields can take the appearance of an image, and in this demonstration, each ballot bubble is created using two buttons – one button displaying an empty bubble and another displaying a full bubble. At any given time, one of these buttons is hidden and the other is visible.

These form fields are named to reflect which of the four votes they count towards and which candidate within that vote they represent. For example, Emily Stone's button form fields are named "vote4_cd2_empty" and "vote4_cd2_full", because she is the second candidate in the fourth vote.

B. Internal JavaScript

All form fields have a display property that can be set to hidden or visible, among other options, according to the fields of an Adobe-defined display object. So as an example, Emily Stone's ballot bubble could be filled with the following document-level JavaScript:

```
this.getField("vote4_cd2_empty").display = display.hidden;
this.getField("vote4_cd2_full").display = display.visible;
```

We have inserted a document-level JavaScript function called SelectBubble that, given a vote number and candidate number, essentially does the work of selecting that candidate and deselecting all other candidates in that vote.

The trick to altering votes in this seemingly static ballot, then, is just a matter of calling SelectBubble on non-user triggers. In this demonstration, the non-user trigger is a JavaScript interval that is started by a document-level script when the document is opened. Every second, this interval will run a function to check if the minute has changed, and when it has, it will execute a function UpdateForm which in turn makes calls to SelectBubble with predetermined arguments based on the minute parity.

IV. EXTENSIONS

This is a simple demonstration used to show an easily-detectable attack on a non-interactive native PDF ballot. However, the scripting principles underlying this demonstration apply to a much broader range of PDF ballots, and can be used in much sneakier attacks. I will briefly describe how PDF scripting can be used to manipulate interactive PDF ballots, as well as non-native PDF ballots such as scanned images.

This demonstration naturally applies to interactive PDF ballots, which are built using the same form fields that this malicious JavaScript targets. In the above demonstration, all button form fields have been set to read-only, so that they can no longer be interacted with by user clicks. This was done to mimic how a downloaded, marked, non-interactive ballot might look. However, by removing this read-only setting, this document could quickly be turned into an interactive ballot that allows users to mark their votes, with all of the demonstrated vulnerabilities.

Furthermore, even non-native PDF ballots, such as photos or scanned images of filled ballots, are liable to PDF scripting attacks. Because PDF scripting allows malicious code to hide and display images, a hidden image of an entire pre-filled ballot could easily be programmatically revealed to "overwrite" the entirety of a legitimate scanned ballot. This means that even non-native PDF ballots can still be manipulated in a compromised PDF.

In all of these ballot types, there also exist far more sophisticated scripting attacks. These attacks could, for example, manipulate ballots only after a specific date, could execute based on the machine's time zone, could remove candidates from blank ballots sent to voters, could allow hackers to target only a small percentage of corrupted ballots, and could programmatically delete evidence of tampering after the fact. Demonstrations and further discussion of many of these attack extensions are explored in the full version of this article [6].

V. CONCLUSIONS

In the domain of voting security and ballot technology, PDF files possess often overlooked capabilities that make them inherently at-risk for manipulation attacks. Although online voting might increase accessibility and voter turn out, it also makes the election process more dependent on the security of voters' and election officials' machines. If a voter is phished, if an election official or voter is using a corrupted PDF reader, or if a voter is using a once-legitimate ballot marking software that has been hacked, PDF ballots with malicious JavaScript code are liable to enter into the system, compromising the integrity of the election. It is particularly difficult to trust that third-party ballot marking software such as Omniballot is sufficiently protected against potential corrupting attacks, considering that many of these software manufacturers don't disclose their software to the public. Because it is infeasible to expect all voters and election officials to maintain secure machines and secure software, it is advisable to minimize the usage of PDF ballots in official elections.

VI. ACKNOWLEDGEMENTS

Thank you to my advisor Jennifer Rexford for her encouragement and countless contributions to this project. Thank you also to Andrew Appel, who initially suggested investigating PDF ballot manipulation and who regularly provided helpful guidance. Thank you also to Arvind Narayanan, Kartikeya Kandula, Matthew Bernhard, and Michael A. Specter for taking the time to meet with me and offer their expertise on both information security and online voting practices.

VII. TROUBLESHOOTING

If the included demonstration appears to not be working, please consider the following. This document must be in its original PDF form, and not in some other image file format. Additionally, it must be viewed in an advanced and up-to-date PDF viewer like Adobe Acrobat or the Google Chrome browser. Currently, the most up-to-date working versions of these applications are 22.001.20085 (Acrobat Reader DC and Acrobat Pro DC), and 99.0.4844.83 (Chrome). Other readers, like Apple's Preview, will restrict much PDF functionality such as the execution of JavaScript code, and will not run the demonstration even if up-to-date. An original version of this file can be found at: `https://github.com/henryprinceton/senior-thesis`.

REFERENCES

[1] 111th Congress, "Military and Overseas Voter Empowerment Act," 2009. [Online]. Available: https://www.fvap.gov/uploads/FVAP/Policies/moveact.pdf

[2] Cass County, Missouri, "April 2022 Sample Ballot," 2022. [Online]. Available: https://www.casscounty.com/DocumentCenter/View/2969/April-Sample-Ballot

[3] Democracy Live, "OmniBallot Portal." [Online]. Available: https://democracylive.com/omniballot-portal/

[4] ——, *OmniBallot Paper Return*, Oct 2020. [Online]. Available: https://www.youtube.com/watch?v=NFz5asVo5Qo

[5] Federal Voting Assistance Program, "Voting Assistance Guide." [Online]. Available: https://www.fvap.gov/guide/chapter2

[6] Henry D. Herrington, "Ballot Acrobatics: Altering Electronic Ballots using Internal PDF Scripting," Undergraduate Thesis, Princeton University, April 2022.

[7] National Conference of State Legislatures, "Voting Outside the Polling Place: Absentee, All-Mail and other Voting at Home Options," Mar 2022. [Online]. Available: https://www.ncsl.org/research/elections-and-campaigns/absentee-and-early-voting.aspx

[8] Office of the Nevada Secretary of State, "EASE - Overview." [Online]. Available: https://www.nvsos.gov/sos/elections/voters/uniformed-overseas-citizens/ease-overview