How AuditEngine Works

Raymond Lutz, Citizens' Oversight Projects. Adapted from "Auditing Elections Using Ballot Images and AuditEngine: General Background", version 2, 2022-10-04.

PDF

Download this page as a PDF, to read offline or to attach to an email.

What AuditEngine does

AuditEngine audits the tabulation of an election using ballot images. It reads every ballot image and evaluates the votes on it. It does not use any software or hardware of the voting system except for normal exports. It then compares its result with the official result separately for each ballot.

Citizens' Oversight Projects, a nonpartisan 501(c)(3) nonprofit organization, develops AuditEngine.

AuditEngine runs in the cloud. It has about 45 separate steps called "stages." In the largest stages, many computers process the ballot images at once. A stage that would take weeks on one computer then takes minutes. Other stages include manual steps, such as mapping the targets on each ballot style, and analyzing the results.

This page describes the general method. Reports on specific audits refer to it for background and terms.

Why audit ballot images

Voting systems and ballot images

A voting system includes the Election Management System (EMS), which handles most elements of the election. These include laying out the ballots, tabulating ballots, and reporting the results.

Some districts have scanners in polling places that scan and internally tabulate each ballot. If configured, a scanner can warn the voter of an overvote or a blank ballot. In other districts, polling places transport ballots to the central office for scanning. Mail ballots are also scanned centrally on high-speed scanners.

Modern voting systems use scanners that scan the entire sheet on both sides, creating a pair of digital images. The images are typically at 200 pixels per inch, higher than the typical computer screen. The system then reads the votes from this image. So the image is a critical record of the election. It should be preserved, and it is usually a public record.

We reviewed the ten battleground states of the 2020 general election. We looked at the most populous jurisdictions, which hold 80% of the voters in each state. Of those jurisdictions, 93% used equipment that created ballot images.

What a ballot image audit checks

The images are created early in the process. So an audit of the images can detect a later change to the tabulation. It can also measure how well the voting system judged voter intent.

Other audits check the tabulation too. A full hand count counts every ballot by hand. A risk-limiting audit counts a random sample of ballots or batches. Both methods handle paper and rely on people to count, which costs time and adds error. A ballot image audit can compare every ballot in every contest with the official result. That works no matter how close a contest is.

These methods can work together. The images are a check on the paper ballots, and the paper ballots are a check on the images. A complete audit program reviews both.

What a tabulation audit does not cover

A ballot image audit covers the tabulation, from the scanned ballot to the final result. It does not cover what happens before a ballot is cast, such as voter eligibility or signatures. It does not show that the set of ballots is complete. Audits of the chain of custody, of eligibility and of procedures address those questions.

AuditEngine can compare the number of ballots with the number of voters recorded as voting.

AuditEngine never receives the identity of any voter. This was a design decision. It means AuditEngine has no way to link a vote to a voter.

Remaining risk in the images

An audit of images assumes the images match the paper ballots. In theory, a skilled attacker could alter images as they are made.[1] Security controls in the scanners reduce this risk. Comparing a sample of images with their paper ballots reduces it further.

We have proposed that each scanner compute a secure hash of each image as it creates the image. The scanner would create a digital cryptographic signature of the hash using a private key

built into the hardware and unknown to anyone. Any later change to an image would then be detectable. The hashes could be published before the images are released.[2] Voting system vendors may claim to have these security features, but we need widespread adoption of standard methods.

Ballots and vendors

Most U.S. jurisdictions primarily use equipment from three vendors. They are Election Systems & Software (ES&S), Dominion Voting Systems and Hart InterCivic. AuditEngine reads the data from ES&S and Dominion systems. Support for Hart is in progress. Newer vendors include Clear Ballot and VotingWorks.

Two kinds of ballots matter for the audit.

A hand-marked paper ballot has ovals or rectangles, called targets. The voter darkens a target with a pen to choose candidates or yes/no options. The voter sees each choice while marking it, and the text that they were presented on the ballot is preserved for viewing in the audit.

A ballot marking device (BMD) is a touch screen device that prints a ballot summary card. The card lists the voter's choices as text. It also encodes them in a barcode or QR code. Voting systems usually read only the barcode, which voters cannot read. The ballot summary card is not what the voter was shown when they voted, and there is no record of the exact presentation of the contests they were shown.

Voters rarely check the printed card. In a University of Michigan study, voters missed over 93% of errors on BMD printouts.[3] J. Alex Halderman, a computer science professor there, showed another risk. Malicious software on a BMD could change the votes in the QR code and leave the printed text alone.[4]

AuditEngine reads the printed text on BMD cards using optical character recognition (OCR). It does not rely on barcodes. So it would catch a change to the barcode, as long as the text shows what the voter chose. If OCR cannot read an ES&S card at all, AuditEngine reads its barcode. It marks that card as not independently checked, and lists it in a report. If the text itself were altered, only the voter could detect it. Districts that use BMDs should guide each voter to check the printed card, which would reduce that risk.[5]

The data

Ballot images

The ballot image is the first digital record of a ballot. It shows whether a voter filled a target, checked it or circled it. Once the vote is extracted, that detail is gone.

So equipment should be set to keep the images. Election offices should preserve and publish them for every election. Some ES&S equipment can be set to delete images. A later election can also overwrite the images from an earlier one.

Federal law requires election records to be kept for 22 months after a federal election.[6] The U.S. Department of Justice has said this includes records in digital form. Ballot images are such records.

The EMS exports the images. For our purposes, they should go into ZIP archives of about 10,000 to 50,000 ballots each. AuditEngine accepts the image formats the vendors produce, including PDF, TIFF, PNG, PBM and JPG. Current Dominion systems use TIFF, and current ES&S systems use PDF. One ballot sheet, front and back, typically takes about 20 KB to 500 KB. Ballots printed without a gray background are easier to read and make smaller files.

The exporting guides explain how to export the data from each vendor's system. Sending Election Data to AuditEngine explains how to send it.

Privacy

Ballot images are anonymous.[7] Nothing in an image proves who cast it. A name written on a ballot does not prove who voted it, as anyone could write some other name. Sometimes people mistakenly initial changes. Some districts redact marks of this type by adding a black block in that area of the image.

Rare ballot styles are a risk to privacy. If only a few voters receive a style, other data might reveal how they voted. We recommend that jurisdictions keep rare styles to a minimum. Ballots in rare styles can be left out of the image set without harming the audit. We also recommend that it be a crime to match a vote to a voter and then use or publish it.

AuditEngine checks each audit for rare styles. Rare styles can be avoided by designing the election district by avoiding small groups.

AuditEngine does not need people to look at most ballots. People review only the ballots with a variant. A variant is a disagreement with the voting system, an overvote, a write-in or an ambiguous gray mark. That is usually a few percent of the ballots.

Every ballot can be researched in full in AuditEngine, so others can check its results. As in science, trust comes from others repeating the analysis.

Cast vote records

A cast vote record (CVR) is the voting system's reading of one ballot sheet. The CVR files hold the CVRs for all the sheets. The vote totals in the CVR files should match the official totals, or come very close.

With CVR files, AuditEngine compares its reading with the official reading for each ballot. Without them, it can compare totals only by contest or by precinct. That still checks the outcome. But it gives little help in finding the ballots that caused a difference.[8]

CVR files must be machine readable. ES&S provides spreadsheets, with one row per ballot sheet. Dominion provides JSON files. Dominion's files include the original reading of each ballot and any change made in adjudication. NIST has published a Common Data Format for CVRs.[9] Vendors do not fully follow it yet, so AuditEngine reads each vendor's own format.

The term "cast vote record" has a second use. ES&S can export a PDF for each ballot that shows how it was read. Dominion prints a similar summary, called the AuditMark, as the third page of each image file.

AuditEngine does not rely on these summaries of single ballots. A consolidated file is more secure, because it is harder to change a few ballots in it without notice. This documentation uses "CVR files" to mean the consolidated files.

The CVR files should include these fields for each sheet:

- the ballot ID, which identifies the sheet,
- the ballot style, which sets the contests, the language and the order of options,
- the precinct,
- the batch, which helps staff find the paper ballot.

Some scanners print the ballot ID on the paper as they scan it. The ID then appears in the image too, which links the image to its paper ballot. The ballot style in the CVR should match the style code printed on the ballot.

Configuration files

AuditEngine uses a few configuration files.

The Election Information File (EIF) lists each contest and its options. It gives each name as it appears in the CVR, in other reports and on the ballots. It gives the number of votes allowed in each contest and the number of write-in lines. It also gives the text printed on BMD cards. AuditEngine can generate most of the EIF from the CVR files, and can fill out the rest by inspecting

ballots. Since some of this information is derived later in the process, the fields that are corrected are included in the "LATER_EIF" so the early stages need not be rerun.

The target map gives the location of each target on each ballot style. Vendors do not provide target maps in any standard form, but even if they did, it would be a violation of independence to use them.

The section Map the styles explains how AuditEngine builds the map.

The job settings file holds the settings for one audit. It names the image archives and the CVR files. It also records facts about the district for the reports.

The audit process

An audit runs as a series of stages. The stages fall into four phases.

Set up and check the data

An "election" is created in AuditEngine for a given "district". Each district can be provided a single consistent upload link. The ballot images, CVR files and ballot style masters are uploaded through it.

These data files are placed in a staging area and then adopted to the secure area for further processing.

An audit job is then created for the election, with its job settings. If the jurisdiction publishes a list of its files with a hash of each, AuditEngine hashes each file. It reports whether each file matches the list, to show the files are identical.

The next stages collect metadata for each ballot. The metadata comes from the image archives, the CVR files and the images themselves. AuditEngine reads the style code from the proprietary coding printed on each ballot. BMD ballots are the exception, because their style is provided in the QR code or barcode, which AuditEngine does not use for this, per policy. For these ballots, AuditEngine derives a basic style from the contests the CVR files show on each ballot. Jurisdictions should print the BMD style on the card so it can be read to avoid needing the CVR. The metadata from these various sources is compared and a common set resolved.

The images are inspected for any exact repeated files, which will not occur unless the images were uploaded incorrectly. The ballot_id may also repeat and these must be resolved.

A report shows any missing, repeated or extra images and CVR records. It also describes the styles, which are the different ballot layouts with different contests, languages, or parties.

Map the styles

For hand-marked ballots, AuditEngine creates a template for each ballot style from the images.

These templates are a combination of many randomly chosen images from the same style. The mapping process determines the location of each target and associates it with the contest and option. This process can be accomplished in three ways.

- Using ballot style masters, which are PDF files that describe each style in detail. These can be parsed and exactly matched because they are not scanned data. This is the most foolproof method, but it does rely on a voting system export.
- Using machine learning and OCR of ballot images to locate targets and match the adjacent text. This can be used if the ovals are printed (not invisible red-ink) and if the ballot is in a common format, like portrait with several vertical columns.
- Using TargetMapper. This method uses human perception and data entry and can map any case no matter what, but it does require time to complete, using several days of work. If the other methods are not successful, the mapping can be completed this way.

AuditEngine then produces two different sets of redline proofs, which show each target outlined and labeled. The proofs let the map be reviewed and verified by eye, and they are used no matter how the mapping is performed. Although these proofs can normally catch any human or machine errors, any remaining mistakes will be detected later in the audit when the comparison is performed.

Extract the votes

AuditEngine reads every ballot image. It aligns the image, measures the darkness of each target and decides the vote. Its threshold adapts to each voter's marks and to the density of the image. It reads BMD cards by OCR. Reports then show the marks found and the totals for each contest.

Compare and report

When CVR files are available, AuditEngine compares its evaluation with the official evaluation of every ballot. The result is the Reconciliation Report. It lists each disagreement and links to the ballot images involved. Without CVR files, the report lists the ballots with overvotes, write-ins and ambiguous gray marks, which can guide further review.

The Reconciliation Report can run to thousands of pages, so it is meant to be read on screen. It gives details on selected contests. By default these are the first ten contests and every contest

with a margin under 10%. They also include the contests where variants or disagreements are largest compared with the margin.

Reviewers can then look at the ballots that need it in the AdjudiTally app. They can review overvotes, write-ins, undervotes and ambiguous marks.

An index links all the reports. A narrative report describes the findings of a finished audit.

About the author

Raymond Lutz founded Citizens' Oversight Projects and is its executive director. The organization has provided oversight of elections for over 15 years. Lutz has a master's degree in electronics and software engineering. He has worked in the document management, office imaging and medical device industries. He is the lead developer of AuditEngine.

---

1. Matthew Bernhard, Kartikeya Kandula, Jeremy Wink and J. Alex Halderman, "UnclearBallot: Automated Ballot Image Manipulation", University of Michigan. Our response is at copswiki.org/Common/M1976. ↵
2. The Internet Engineering Task Force has a working group on Supply Chain Integrity, Transparency and Trust (SCITT). Its work would let data such as ballot images be registered so that any change can be detected. We follow and contribute to it. ↵
3. "New study finds voters not detecting ballot errors, potential hacks", University of Michigan, January 2020. Matt Bernhard and J. Alex Halderman found that voters missed over 93% of errors on printed ballots they filled out on BMDs. ↵
4. "Secret report finds flaw in Georgia voting system, but state in the dark", Atlanta Journal-Constitution. ↵
5. In a directed review, the touch screen asks the voter to compare the printed card with the choices made on screen. It would slow voting. ↵
6. 52 U.S.C. 20701 requires election officers to keep all records relating to an election for 22 months. The U.S. Department of Justice wrote that "jurisdictions must therefore also retain and preserve records created in digital or electronic form," in "Federal Law Constraints on Post-Election Audits", 2021-07-28. ↵
7. Some states still print a unique serial number on ballots. The stated purpose is to pull the ballot of a voter who dies after voting early. The practice should stop, because it lets an official find the votes of any voter. ↵
8. Some systems, such as Clear Ballot, rescan the paper ballots. That avoids relying on the voting system's scans. But the new scans cannot be matched to the official reading of each ballot. They can only be compared in totals, by batch or by precinct, which makes a difference hard to trace. ↵

9. NIST, "Cast Vote Records Common Data Format Specification, Version 1.0". ↩