

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

What Adults Will (and Won't) Do to Prove Their Age: Empirical Evidence from a Deceptive Web Experiment

Yanzi Lin and Cheng Zhang, *Carnegie Mellon University;* Madelyne Xiao, *Princeton University;* Lorrie Faith Cranor and Sarah Scheffler, *Carnegie Mellon University*

https://www.usenix.org/conference/usenixsecurity26/presentation/lin-yanzi

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 35th USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Artifact Appendices to the Proceedings of the 35th USENIX Security Symposium is sponsored by



جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology



USENIX Security '26 Artifact Appendix: What Adults Will (and Won't) Do to Prove Their Age: Empirical Evidence from a Deceptive Web Experiment

Yanzi Lin[*] Cheng Zhang[*] Madelyne Xiao[†] Lorrie Faith Cranor[*] Sarah Scheffler[*]

[*]Carnegie Mellon University [†]Princeton University

A Artifact Appendix

A.1 Abstract

This artifact package supports a web experiment that measures how adults respond to age verification systems when deciding whether to access age-restricted content. It contains the following components:

- Study platform: a React and Firebase web application that randomly assigned participants to one of seven age verification conditions, presented the corresponding interface, and logged their activity during the experiment.
- Demo video: a recording (`study_flow.mov`) showing the flow of the participants through the web experiment.
- Survey data and analysis code: anonymized CSV data from the experiment and post-task survey, along with R and Python scripts that generate all statistical results and figures reported in the paper.
- A complete appendix containing the demographics of the survey participants, complete statistical test results, screenshots from the study site, screenshots of the AgeGuardian interface for all seven conditions, recruitment ad designs and targeting parameters, the survey instrument, and the qualitative codebook.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

Evaluators face no security risks when executing the artifact.

With respect to data privacy, the platform runs locally using placeholder Firebase credentials, does not perform real verification, and does not collect any personally identifiable information that the evaluators may have provided when interacting with the simulated age verification service.

The default Qualtrics survey links are live and are owned by the research group. Completing the survey will submit responses to the research group's Qualtrics account. Evaluators should review Qualtrics' privacy policy and **avoid entering real personal information**. For isolated testing, evaluators can instead substitute their own survey URLs, as described in the README.

A.2.2 How to access

The artifact is available at `https://doi.org/10.5281/zenodo.20343090`. It contains two sub-directories:

- `experiment-software/`
- `survey-data-and-analysis-code/`

Each directory has its own README with setup and run instructions.

A.2.3 Hardware dependencies

The artifact should run on any modern laptop or desktop computer. A smartphone with a camera is needed to evaluate the QR-code scanning feature of the artifact.

A.2.4 Software dependencies

The study platform requires Node.js 18 or later, npm, and a modern web browser. For local development, we recommend Firefox or Safari; additional browser-specific notes are provided in the README. Optional data-collection functionality requires the Firebase CLI, a Firebase project, and a Qualtrics account.

The data-analysis scripts require Python 3 and the Python packages listed in `requirements.txt`, which can be installed and run via `setup.sh`. The statistical analyses additionally require R; running `setup.R` installs the required R packages.

A.2.5 Benchmarks

None.

A.3 Set-up

A.3.1 Installation

To install and run the study platform, download the repository, navigate to the `experiment-software` subdirectory, and run `npm install && npm start`. The platform will then be available at `http://localhost:3000`. Evaluators who wish to test data-collection functionality should additionally follow the "Configure Firebase" instructions in the README.

To perform data analysis, navigate to the `survey-data-and-analysis-code` subdirectory and run `./setup.sh`. This script creates a Python virtual environment, installs the dependencies, and registers a Jupyter kernel. For the R analyses, run `Rscript setup.R` to install the required packages.

A.3.2 Basic Test

To test the study platform, first run `npm start` from the `experiment-software` subdirectory. Then open the following example URL in Firefox or Safari:

```
http://localhost:3000/?user_id=test&c=2&p=rf
```

A successful run displays the landing page. Clicking "Begin Study" should load the consent form in an iframe. After the consent form is submitted, the Important Notice screen should appear, followed by the government-ID condition specified by `c=2`. Completing the age-verification flow, or clicking "Exit Study", should redirect to the survey.

To test the Python analysis, after running `setup.sh`, open `generate_graphs.ipynb` and run all cells. No input parameters are required beyond the anonymized data files already included in the repository. A successful run produces nine PNG files in the `graphs/` sub-directory, and they should match Figure 3 through Figure 7 in the paper.

To test the R analysis, after running `setup.R`, knit or run all cells in `RQ1-RQ2-analysis.Rmd`. No input parameters are required. A successful run prints the chi-squared test result (with $\chi^2 = 619.82$, $p < .001$) and the pairwise proportions test results, matching the statistic reported in the Appendix (Section A.2.2) of the paper.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): The platform implements the seven verification conditions listed in paper's Table 1: checkbox, three government-ID reassurance variants, government-ID with liveness check, AI age estimation, and email age estimation. These conditions can be selected using the URL parameter `c=1` through `c=7`.

(C2): The study platform implements the three non-completion paths described in §3.1 of the paper: closing the browser tab, reaching the inactivity timeout, or clicking the "Exit Study" button.

(C3): Participants' age verification completion behavior varies significantly by condition. The reassurance statements did not affect completion.

(C4): Checkbox and email age estimation are rated significantly more comfortable than other methods; Participants perceived government-ID-based methods and AI age estimation as riskier than checkbox or email verification; Checkbox was rated as significantly less effective than all other methods; GovID methods were rated significantly less convenient than all other methods.

(C5): Participants who reported higher comfort and lower perceived risk had significantly higher odds of completing age verification; perceived effectiveness was not significantly associated with completion.

A.4.2 Experiments

(E1): Verification-condition flows *[10 human-minutes]*: This experiment evaluates claim (C1) by checking that the study platform implements all seven verification conditions listed in Table 1 of the paper.
Preparation: Start the development server as described in §A.3.1, and open the platform in Firefox or Safari. For testing: open each URL in a new incognito or private window. Otherwise, changing the value of c will keep loading the first condition tested, because the platform saves URL parameters to the browser's session storage so a participant keeps their assigned condition if their page refreshes mid-study.
Execution: For each value $N \in \{1, \ldots, 7\}$, open:
`http://localhost:3000/?user_id=test&c=N&p=rf`
Results: Each value of `c` should load a verification condition that matches the corresponding row of Table 1.

(E2): Non-completion paths *[10 human-minutes]*: This experiment evaluates claim (C2) by checking that the study platform implements the three non-completion paths described in the paper.
Preparation: Start the development server as described in §A.3.1. To verify event logging, configure a Firebase project by following the README's "Configure Firebase" instructions; otherwise, the evaluator can still verify the user-visible behavior of each non-completion path locally.
Execution: Open any condition, for example:
`http://localhost:3000/?user_id=test&c=2&p=rf`
Then exercise the following three paths, each time with a fresh run: click "Exit Study", leave the tab idle until the inactivity countdown completes, and close the browser tab before completion.
Results: If "Exit Study" is clicked, the page should be

redirected to the exit survey. For tab closure or timeout, if configuration step 7 in README is performed, the email account entered in the consent form should receive an email containing a link to the survey. If Firebase is configured, the corresponding participant event documents should appear according to the data-collection schema described in the README.

(E3): Experiment data analysis *[5 human-minutes]*: This experiment evaluates claim (C3) by checking that the participants' age verification behavior matches the result reported in Table 3, Section A.2.2 in the Appendix.
Preparation: Ensure R and the required packages are installed by running `setup.R`. Confirm that `anonymized_experiment_data.csv` is present in the same directory as `RQ1-RQ2-analysis.Rmd`.
Execution: Open `RQ1-RQ2-analysis.Rmd` in RStudio and knit it (or run all chunks in order from the command line).
Results: Inspect the chi-squared test output and the pairwise proportion test output and confirm that they match with what is reported in Table 3 (Section A.2.2 in the Appendix).

(E4): Likert construct analysis *[10 human-minutes]*: This experiment evaluates claim (C4) by checking that participants' self-reported comfort, risk, effectiveness, and convenience ratings match the results reported in Tables 2, 4, 5, and 6 in the Appendix.
Preparation: Ensure R and the required packages are installed by running `setup.R`. Confirm that `anonymized_survey_data.csv` is present in the same directory as `RQ3-likert-analysis.Rmd`.
Execution: Open `RQ3-likert-analysis.Rmd` in RStudio and knit it (or run all chunks in order from the command line).
Results: Inspect the four Kruskal-Wallis test outputs (comfort, risk, effectiveness, convenience) and their pairwise Wilcoxon rank-sum test outputs, and confirm that they match what is reported in Tables 2, 4, 5, and 6 (Appendix A.2.1, A.2.3–A.2.5).

(E5): Completion regression *[5 human-minutes]*: This experiment evaluates claim (C5) by checking that the logistic regression of completion on comfort, risk, and effectiveness matches the result reported in Table 11 in the Appendix.
Preparation: Ensure R and the required packages are installed by running `setup.R`. Confirm that `anonymized_survey_data.csv` is present in the same directory as `completion_and_recruitment_regressions.Rmd`.
Execution: Open `completion_and_recruitment_regressions.Rmd` in RStudio and knit it (or run all chunks in order from the command line). The first chunk fits the logistic regression of completion on comfort, risk, effectiveness, and condition; the second chunk fits a separate regression on recruitment platform and gender (Appendix A.1) and is not part of this claim.
Results: Inspect the output of the first model ("LOGISTIC REGRESSION: COMPLETION COMFORT + RISK + EFFECTIVENESS + CONDITION") and confirm the results match Table 11 in the Appendix.

A.5 Notes on Reusability

The study platform is designed to be extensible. Researchers may adapt it to evaluate additional age-verification methods by adding new verification-condition components and routing parameters. The platform can also be adapted to other realistic study contexts.

The analysis code may also be reused for similar survey-based studies.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2026/.