

# No boundaries: Exfiltration of personal data by session-replay scripts

**Steven Englehardt**

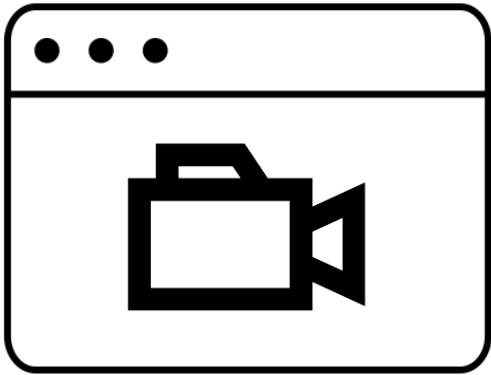
@s\_englehardt  
senglehardt.com

Joint work with:  
Gunes Acar and  
Arvind Narayanan

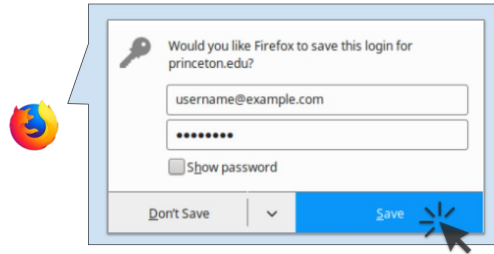


# No boundaries series: Privacy vulnerabilities arising from directly embedded third parties

## Session Recording



## Autofill abuse



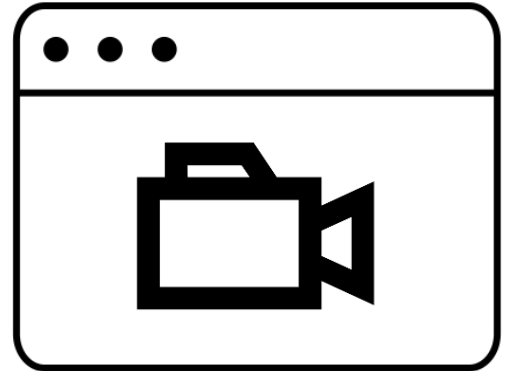
**More to come!**

*"No boundaries: Exfiltration of personal data by session-replay scripts" (freedom-to-tinker.com)*

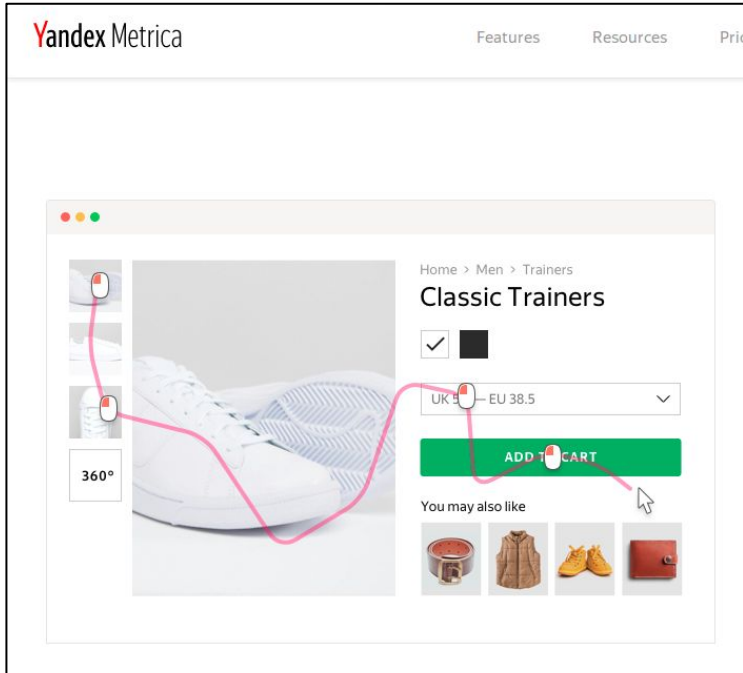
*"No boundaries for user identities: Web trackers exploit browser login managers" (freedom-to-tinker.com)*

# What are session recording scripts?

- Session recording scripts create a “video” of all of a user’s actions on a site.
- Publishers can later review the videos.



# Why use session recording scripts?



Answer questions like:

- Who are my most valuable customers?
- Who added items to the cart but didn't convert?
- Where do users leave the onboarding flow?
- Where are users frustrated?

*More than just site optimization:* Jornaya (LeadID) uses recordings to “prove consent” for data collection

Jornaya | The Surest Signals of Consumer Intent (formerly LeadID) - Mozilla Firefox

Jornaya | The Surest Sig x +

www.jornaya.com

PLATFORM SUPPORT SIGN UP LOGIN

### VISUAL PLAYBACK

Page Load 0:00

0:00 / 19:26

Match Data Audit Results FAQs

#### MATCH DATA

Enter values to see if those values match what was actually entered during the lead event.

First Name  ✓

Last Name  ✗

Phone

Email

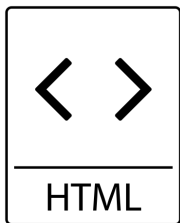
To see if additional fields match, enter the data values in the fields below

## WELCOME TO JOE'S LEAD SHACK

In today's world, prospective surfers come from all walks of life. To help you take a break from the surf to find special offers specific to the benefits to using Joe's Lead Shack:

- ✓ **Flexibility!** Search offers from our partners on your schedule
- ✓ **Convenience!** Hundreds of offers in one convenient location

**The problem:** recordings require a **ton** of data



Full page source and text



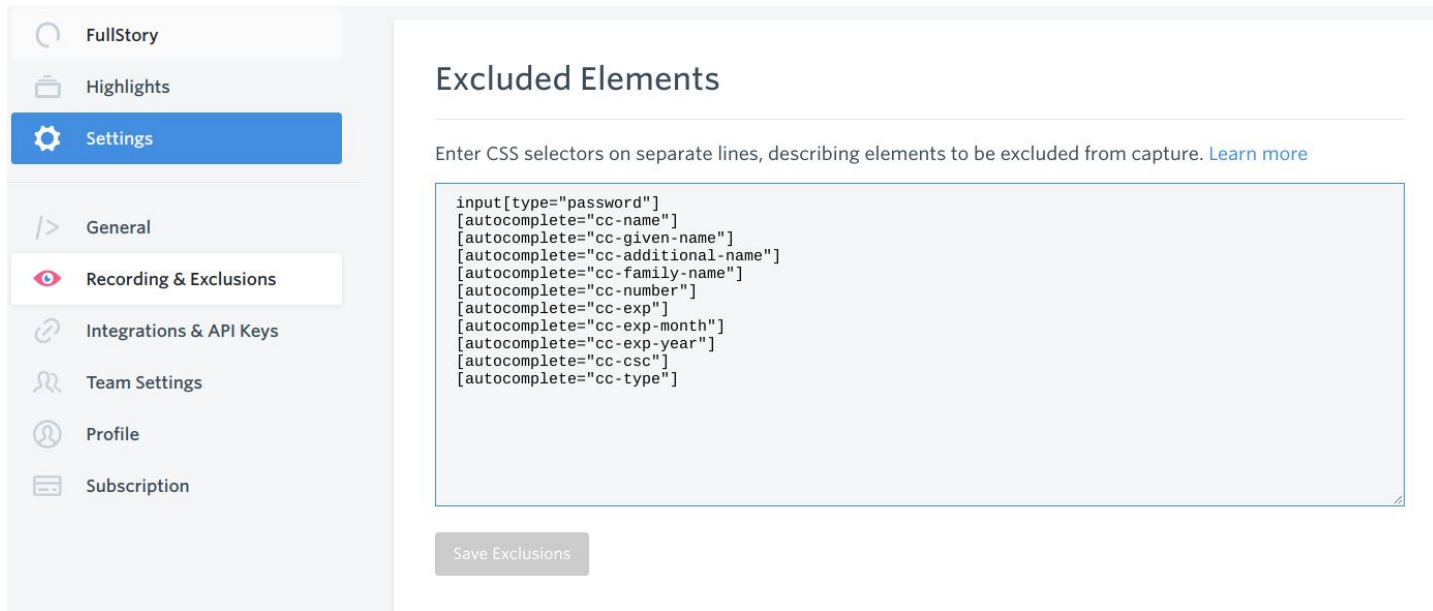
Mouse movements & clicks



Keypresses



# Scripts have automated redaction...



The screenshot displays the FullStory settings interface. On the left is a sidebar with navigation options: FullStory, Highlights, Settings (highlighted in blue), General, Recording & Exclusions (highlighted in white), Integrations & API Keys, Team Settings, Profile, and Subscription. The main content area is titled 'Excluded Elements' and contains a text input field with the following CSS selectors:

```
input[type="password"]
[autocomplete="cc-name"]
[autocomplete="cc-given-name"]
[autocomplete="cc-additional-name"]
[autocomplete="cc-family-name"]
[autocomplete="cc-number"]
[autocomplete="cc-exp"]
[autocomplete="cc-exp-month"]
[autocomplete="cc-exp-year"]
[autocomplete="cc-csc"]
[autocomplete="cc-type"]
```

Below the text area is a 'Save Exclusions' button.



# Scripts have automated redaction...

- FullStory
- Highlights
- Settings**

- General
- Recording & Exclusions**
- Integrations & API Keys
- Team Settings
- Profile
- Subscription

## Excluded Elements

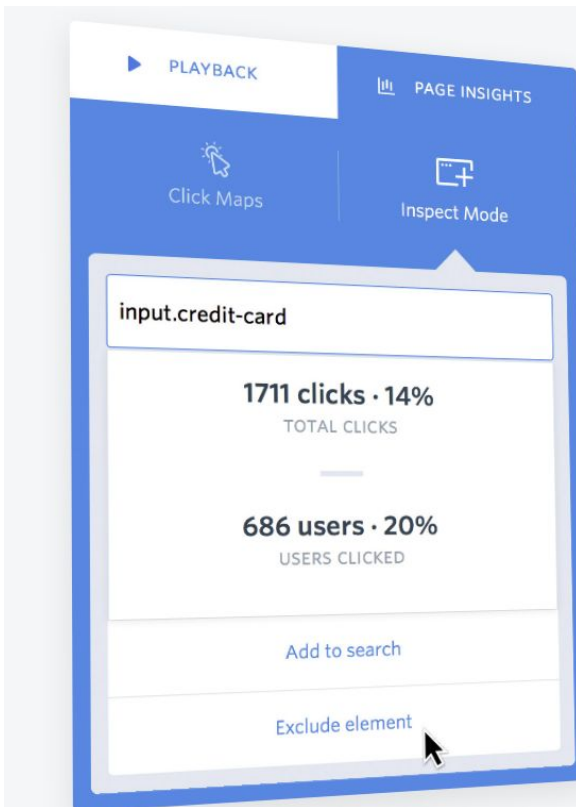
Enter CSS selectors on separate lines, describing elements to be excluded from capture. [Learn more](#)

```
input[type="password"]
[autocomplete="cc-name"]
[autocomplete="cc-given-name"]
[autocomplete="cc-additional-name"]
[autocomplete="cc-family-name"]
[autocomplete="cc-number"]
[autocomplete="cc-exp"]
[autocomplete="cc-exp-month"]
[aut
[aut
[aut
```

Redacted Field	FullStory	UserReplay	SessionCam	Hotjar	Yandex	Smartlook
Name	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Email	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Phone	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Address	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/> †	<input type="radio"/>	<input type="radio"/>
SSN	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
DOB	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Password	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
CC Number	<input checked="" type="radio"/>	<input checked="" type="radio"/> *	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
CC CVC	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
CC Expiry	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Save

# Scripts also support manual redaction



The screenshot shows a web analytics dashboard with a blue header. On the left, there are tabs for 'PLAYBACK' and 'PAGE INSIGHTS'. Below these are icons for 'Click Maps' and 'Inspect Mode'. A white box highlights a specific element with the ID 'input.credit-card'. The box displays the following statistics: '1711 clicks · 14%' with 'TOTAL CLICKS' below it, and '686 users · 20%' with 'USERS CLICKED' below it. At the bottom of the box, there are two buttons: 'Add to search' and 'Exclude element', with a mouse cursor hovering over the latter.

**Easily protect your user's privacy.**

Exclude sensitive customer data from ever leaving your customer's browser by using our in-app point and click system.

# Session recording scripts are (too) easy to integrate

Smartlook.com

## Instructions for webmasters that manage your site:

Insert the following code into your website between tags <head> and </head>. The moment you have a visitor on your website, recordings will start to show within a few minutes.

```
<script type="text/javascript">  
  window.smartlook||(function(d) {  
    var o=smartlook=function(){ o.api.push(arguments)},h=d.getElementsByTagName('head')[0],  
    var c=d.createElement('script');o.api=new Array();c.async=true;c.type='text/javascript';  
    c.charset='utf-8';c.src='https://rec.smartlook.com/recorder.js';h.appendChild(c);  
  })(document);  
  smartlook('init', 'd83e650fa9d18b86892b229b95a2f5a4d84a4e7b');  
</script>
```

COPY CODE

00:00:26

Tick tock tick tock... every minute you lose a lot of video material!

DONE, STOP ME THE RECORDINGS

Set up Hotjar with one script in a matter of seconds.

Hotjar works out of the box on most popular platforms including:

A timer counting how long it takes you to embed their code.

# How can things go wrong?

We found session recordings containing:

- Health data (Walgreens)
- Student data (Gradescope)
- Credit Card data (Bonobos)
- Purchase data (Lenovos)

in a relatively small manual review of sites.

# Recording redactions miss sensitive data

The image shows a web browser window with a form titled "Add New Card" and a network developer tool open on the right. The form contains the following fields:

- NAME (As it appears on your card):** John Doe (highlighted with a blue box)
- CARD NUMBER:** 4111111111111111 VISA (highlighted with a red box)
- MONTH:** 10
- YEAR:** 2020
- CVV:** 456
- COUNTRY:** United States
- FIRST NAME:** John
- LAST NAME:** Doe

The network developer tool shows a list of requests. Several requests have their response bodies redacted with "4111111111111111" or "41111111111111111111", which are highlighted with red boxes. These redactions correspond to the sensitive data entered in the form.

```
4: {When: 385423, Kind: 18, Args: [104, "John Do"]}
5: {When: 385424, Kind: 15, Args: [79]}
6: {Kind: 4, When: 385442, Args: [1072, "value", "2017-11-14"]}
7: {When: 385488, Kind: 14, Args: [69]}
8: {When: 385566, Kind: 18, Args: [104, "John Doe"]}
9: {When: 385567, Kind: 15, Args: [69]}
10: {Kind: 4, When: 385692, Args: [1072, "value", "2017-11-14"]}
11: {When: 385678, Kind: 9,...}
12: {When: 385945, Kind: 9,...}
13: {When: 386012, Kind: 9, Args: [386148, 56, 48, -8, 3125, ...]}
14: {When: 386162, Kind: 9, Args: [386192, 48, 48, 0, 0, 297, ...]}
15: {When: 386195, Kind: 9,...}
16: {When: 386295, Kind: 9,...}
17: {When: 386393, Kind: 12, Args: [72, 313]}
18: {When: 386394, Kind: 24, Args: [1041]}
19: {When: 386396, Kind: 17, Args: [1047]}
20: {When: 386399, Kind: 59, Args: [1046, 0]}
21: {When: 386495, Kind: 13, Args: [72, 313]}
22: {When: 386496, Kind: 16, Args: [1047, 72, 313, 45, 299, ...]}
23: {When: 387544, Kind: 14, Args: [52]}
24: {When: 387638, Kind: 18, Args: [107, "4"]}
25: {When: 387639, Kind: 15, Args: [52]}
26: {When: 388087, Kind: 14, Args: [49]}
27: {When: 388166, Kind: 18, Args: [107, "41"]}
28: {When: 388167, Kind: 15, Args: [49]}
29: {When: 388328, Kind: 14, Args: [49]}
30: {When: 388422, Kind: 18, Args: [107, "411"]}
31: {When: 388423, Kind: 15, Args: [49]}
32: {Kind: 4, When: 388444, Args: [1048, "class",...]}
33: {Kind: 4, When: 388444, Args: [1072, "value", "2017-11-14"]}
34: {When: 388567, Kind: 14, Args: [49]}
35: {When: 388670, Kind: 18, Args: [107, "4111"]}
36: {When: 388671, Kind: 15, Args: [49]}
37: {When: 389375, Kind: 14, Args: [49]}
38: {When: 389454, Kind: 18, Args: [107, "411111"]}
39: {When: 389455, Kind: 15, Args: [49]}
40: {When: 389567, Kind: 14, Args: [49]}
41: {When: 389630, Kind: 18, Args: [107, "411111"]}
42: {When: 389631, Kind: 15, Args: [49]}
43: {When: 389775, Kind: 14, Args: [49]}
```

# Recording includes CVV field → Not PCI compliant?

The screenshot shows a web browser window with the URL <https://bonobos.com/account/wallet>. The page title is "Your Wallet | Bonobos". The main content area is titled "Wallet" and contains a form for "Add New Card". The form fields are:

- NAME (As it appears on your card): John Doe
- CARD NUMBER: 4111111111111111 VISA
- MONTH: 10
- YEAR: 2020
- CVV: 456

The network tab shows several requests with headers containing sensitive information. The headers for requests 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, and 43 are visible. The headers for requests 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, and 43 are visible. The headers for requests 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, and 43 are visible.

## Technical Guidelines for PCI Data Storage

	Data Element	Storage Permitted	Protection Required	PCI DSS Req. 3.4
Cardholder Data	Primary Account Number (PAN)	Yes	Yes	Yes
	Cardholder Name <sup>1</sup>	Yes	Yes <sup>1</sup>	No
	Service Code <sup>1</sup>	Yes	Yes <sup>1</sup>	No
	Expiration Date <sup>1</sup>	Yes	Yes <sup>1</sup>	No
Sensitive Authentication Data <sup>2</sup>	Full Magnetic Stripes Data <sup>3</sup>	No	N/A	N/A
	CAV2/CVC2/CVV2/CID	No	N/A	N/A
	PIN/PIN Block	No	N/A	N/A

From:  
[https://www.pcisecuritystandards.org/pdfs/pci\\_fs\\_data\\_storage.pdf](https://www.pcisecuritystandards.org/pdfs/pci_fs_data_storage.pdf)

# What happened?

The screenshot shows a web browser window with the URL `https://bonobos.com/account/wallet`. The page title is "Your Wallet | Bonobos". The browser's developer tools are open, showing the Network tab with a list of requests. The form on the page is titled "Wallet" and has several fields:

- Add New Card** (Cancel)
- NAME** (As it appears on your card):  (highlighted with a blue box)
- CARD NUMBER**:   (highlighted with a red box)
- MONTH**:  **YEAR**:  **CVV**:
- Set Card as Default** (checkbox)
- COUNTRY**:
- FIRST NAME**:  **LAST NAME**:
- ADDRESS**:
- Chat with us** (button)

The Network tab shows a list of requests. The first few requests are highlighted with blue boxes in the original image, showing headers and response data. The response data for these requests includes:

- Request 4: `{When: 385423, Kind: 18, Args: [104, "John Do"]}`
- Request 5: `{When: 385424, Kind: 15, Args: [79]}`
- Request 6: `{Kind: 4, When: 385442, Args: [1072, "value", "2017-11-14]}`
- Request 7: `{When: 385488, Kind: 14, Args: [69]}`
- Request 8: `{When: 385566, Kind: 18, Args: [104, "John Doe"]}` (highlighted with a blue box)
- Request 9: `{When: 385567, Kind: 15, Args: [69]}`
- Request 10: `{Kind: 4, When: 385692, Args: [1072, "value", "2017-11-14]}`
- Request 11: `{When: 385878, Kind: 9, ...}`
- Request 12: `{When: 385945, Kind: 9, ...}`
- Request 13: `{When: 386812, Kind: 9, Args: [386148, 56, 48, -8.3125, ...]}`
- Request 14: `{When: 386162, Kind: 9, Args: [386192, 48, 48, 0, 0, 299]}`
- Request 15: `{When: 386195, Kind: 9, ...}`
- Request 16: `{When: 386295, Kind: 9, ...}`
- Request 17: `{When: 386393, Kind: 12, Args: [72, 313]}`
- Request 18: `{When: 386394, Kind: 24, Args: [1041]}`
- Request 19: `{When: 386396, Kind: 17, Args: [1047]}`
- Request 20: `{When: 386399, Kind: 59, Args: [1046, 0]}`
- Request 21: `{When: 386495, Kind: 13, Args: [72, 313]}`
- Request 22: `{When: 386496, Kind: 16, Args: [1047, 72, 313, 45, 299]}`
- Request 23: `{When: 387544, Kind: 14, Args: [52]}`
- Request 24: `{When: 387638, Kind: 18, Args: [107, "4"]}` (highlighted with a red box)
- Request 25: `{When: 387639, Kind: 15, Args: [52]}`
- Request 26: `{When: 388887, Kind: 14, Args: [49]}`
- Request 27: `{When: 388166, Kind: 18, Args: [107, "41"]}` (highlighted with a red box)
- Request 28: `{When: 388167, Kind: 15, Args: [49]}`
- Request 29: `{When: 388328, Kind: 14, Args: [49]}`
- Request 30: `{When: 388422, Kind: 18, Args: [107, "411"]}` (highlighted with a red box)
- Request 31: `{When: 388423, Kind: 15, Args: [49]}`
- Request 32: `{Kind: 4, When: 388444, Args: [1048, "class", ...]}`
- Request 33: `{Kind: 4, When: 388444, Args: [1072, "value", "2017-11-14]}`
- Request 34: `{When: 388567, Kind: 14, Args: [49]}`
- Request 35: `{When: 388670, Kind: 18, Args: [107, "4111"]}` (highlighted with a red box)
- Request 36: `{When: 388671, Kind: 15, Args: [49]}`
- Request 37: `{When: 389375, Kind: 14, Args: [49]}`
- Request 38: `{When: 389454, Kind: 18, Args: [107, "41111"]}` (highlighted with a red box)
- Request 39: `{When: 389455, Kind: 15, Args: [49]}`
- Request 40: `{When: 389567, Kind: 14, Args: [49]}`
- Request 41: `{When: 389630, Kind: 18, Args: [107, "411111"]}` (highlighted with a red box)
- Request 42: `{When: 389631, Kind: 15, Args: [49]}`
- Request 43: `{When: 389775, Kind: 14, Args: [49]}`

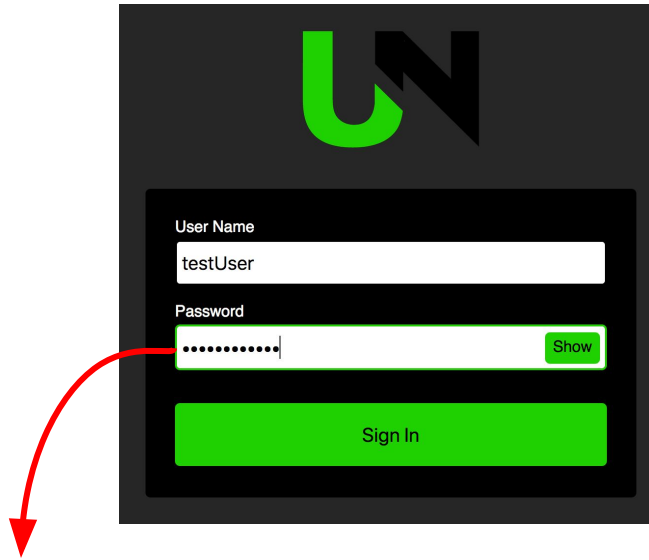
Bonobos used:

```
<input type="text"></input>
```

Bonobos should have used:

```
<input type="text"  
autocomplete="cc-number"></input>
```

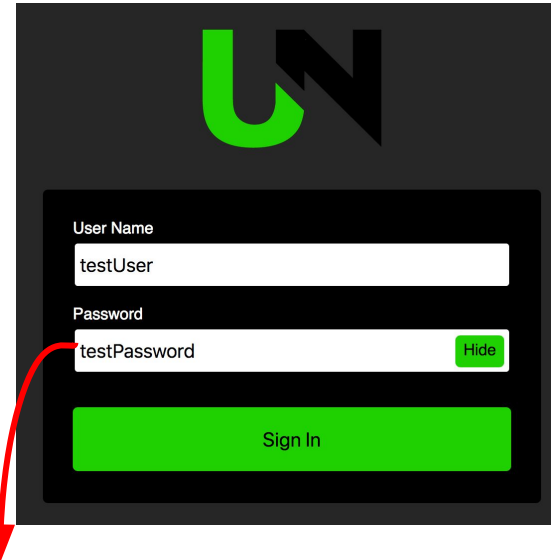
# Unexpected input types can also cause password leaks



```
<input type="password"></input>
```

(automatically redacted with [type="password"] rule)

Show password  
clicked...



```
<input type="text"></input>
```

(automatic redaction fails)



# Walgreens misses fields during redaction

The screenshot shows a web browser window titled "Prescription Checkout" with the URL [https://www.walgreens.com/pharmacy/prescriptioncheckout\\_new.jsp](https://www.walgreens.com/pharmacy/prescriptioncheckout_new.jsp). The page has two radio buttons: "Pick Up in Store" (selected) and "Ship to Home".

The main content area is divided into two sections:

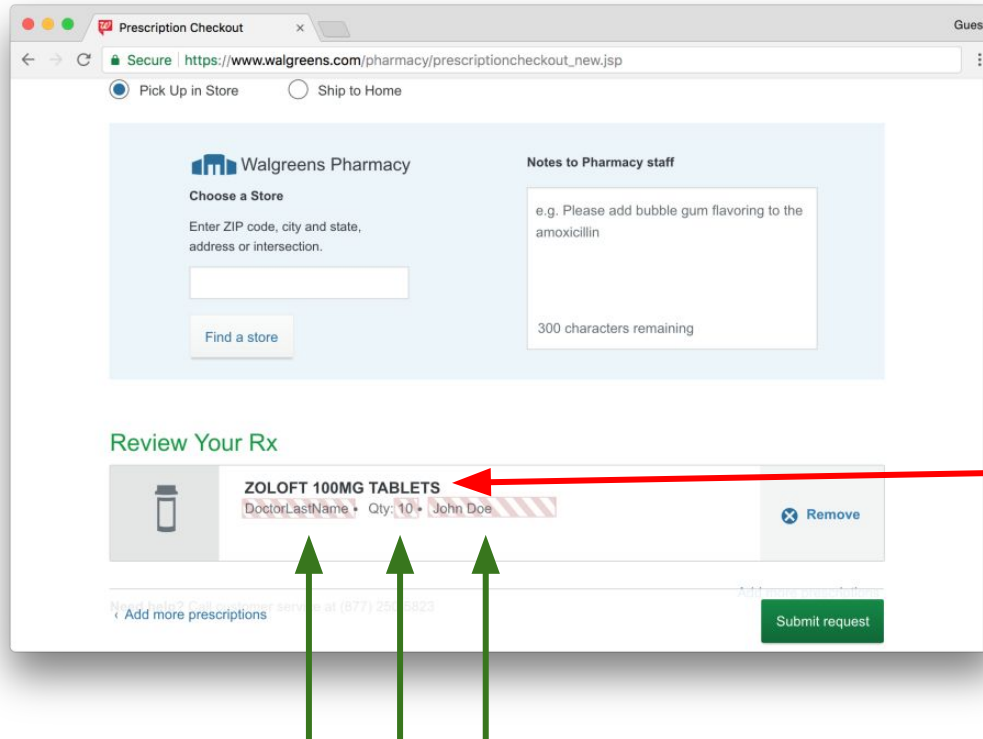
- Walgreens Pharmacy**: Includes a "Choose a Store" section with a text input field for "Enter ZIP code, city and state, address or intersection." and a "Find a store" button.
- Notes to Pharmacy staff**: Includes a text area with the example text "e.g. Please add bubble gum flavoring to the amoxicillin" and a "300 characters remaining" indicator.

The "Review Your Rx" section displays a prescription card for "ZOLOFT 100MG TABLETS". The card includes a pill icon, the drug name, and a redacted area for "DoctorLast Name" and "John Doe". The quantity is "Qty: 10". A "Remove" button is visible on the right. Three green arrows point to the redacted fields.

At the bottom of the page, there is a "Submit request" button and a link for "Add more prescriptions".

Walgreens makes thorough use of redaction

# Walgreens misses fields during redaction

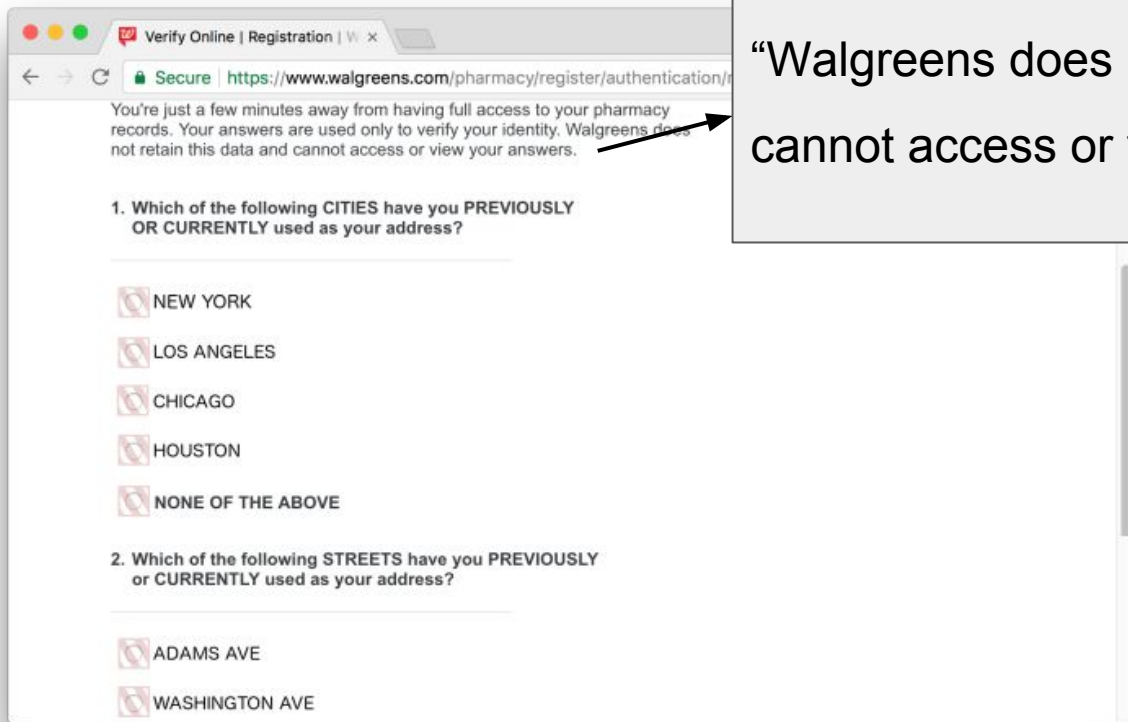


But prescription information is missed!

(the user's full name included was not redacted on the previous page)

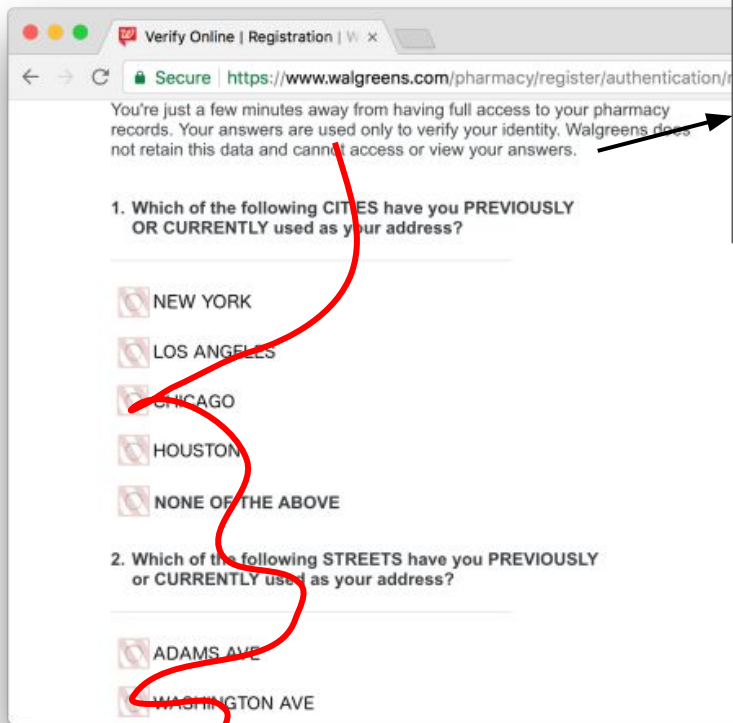
Walgreens makes thorough use of redaction

# Recording redactions miss sensitive data



“Walgreens does not retain this data and cannot access or view your answers.”

# Recording redactions miss sensitive data



“Walgreens does not retain this data and cannot access or view your answers.”

Although selection inputs redacted, mouse trace is still recorded.

### Review Grades for Homework1Assignment

● REGRADE REQUESTS OPEN ● GRADES NOT PUBLISHED

MINIMUM	MEDIAN	MAXIMUM	MEAN	STD DEV
2.5	2.75	3.0	2.75	0.35

2 Students

Search 🔍

NAME	EMAIL	SCORE/3.0	GRADED?	VIEWED?	TIME (EST)
Gunes Acar	gunes@princeton.edu	2.5	✓	👁	Dec 20 7:21pm
Steven Englehardt	ste@princeton.edu	3.0	✓	👁	Dec 20 5:06pm

Gradescope recordings included:

- Student name
- Student emails
- Student grades
- Professor comments

# FullStory forbids PII sharing?

1.3 “Sensitive Data” means any information that: (a) requires a high degree of protection by law and where loss or unauthorized disclosure would require notification by Customer to government agencies, individuals or law enforcement, (b) any information that, if made public, could expose individuals to a risk of physical harm, fraud, or identity theft. Sensitive Data includes, but is not limited to, social security numbers or other government-issued identification numbers, financial account numbers, credit card or debit card numbers, CVVs, credit report information or other personal financial information, health or medical information or other information that is subject to international, federal, state, or local laws or ordinances now or hereafter enacted regarding data protection or privacy, including, but not limited to, the Health Insurance Portability and Accountability Act, the Health Information Technology for Economic and Clinical Health Act, the Fair Credit Reporting Act, the Children’s Online Privacy Protection Act and the Gramm-Leach-Bliley Act

First party restricted from sharing sensitive data:

*“Customer agrees that it will not provide any Sensitive Data to FullStory.”*

# Session recordings are widespread

- 14+ analytics company offer recording services
  - Present on 99,174 of the top 1 million sites
- Evidence of recording on 7,918 sites.
  - Likely a lower bound as recording scripts sample users

Session recording present on ~1 - 10% of the top 1 million sites. We found several severe PII leaks after manually reviewing ~30 sites.

→ **How many more leaks are out there?**

# Recording companies don't always handle data well

## Recordings on HTTPS pages played back over HTTP

- At time of measurement: Yandex, Smartlook, and Hotjar were doing this.
- Smartlook has since fixed this

 **Not secure** | example.com

## Password length leaked in recording

- At time of measurement: Smartlook and UserReplay were doing this.
- Smartlook has since fixed this

`secret_password` → `*****`      `while`      `abc123` → `*****`



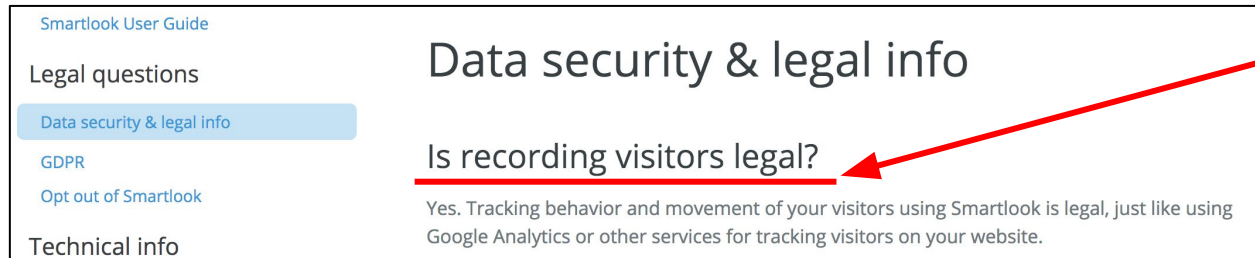
# Takeaways

1. Recordings contain sensitive information
2. Redaction is difficult and brittle.

Example rule:

```
form[name=\"financialInfoForm\"]>table:last-child>tbody>tr>td>table:first-child>tbody>tr>td>table>tbody>tr>td>table>tbody>tr>td>table>tbody>tr:nth-child(3)
```

3. Are users comfortable being watched?



Smartlook User Guide

- Legal questions
- Data security & legal info
- GDPR
- Opt out of Smartlook
- Technical info

## Data security & legal info

### Is recording visitors legal?

Yes. Tracking behavior and movement of your visitors using Smartlook is legal, just like using Google Analytics or other services for tracking visitors on your website.

Needing to ask “Is this legal?” should give you pause.

# Thank you!

## Blog post series:

No boundaries: Exfiltration of personal data by session-replay scripts

→ <https://freedom-to-tinker.com/2017/11/15/no-boundaries-exfiltration-of-personal-data-by-session-replay-scripts/>

No boundaries for user identities: Web trackers exploit browser login managers

→ <https://freedom-to-tinker.com/2017/12/27/no-boundaries-for-user-identities-web-trackers-exploit-browser-login-managers/>

Website operators are in the dark about privacy violations by third-party scripts

→ <https://freedom-to-tinker.com/2018/01/12/website-operators-are-in-the-dark-about-privacy-violations-by-third-party-scripts/>

Image assets from the Noun Project: recording by Guru, browser windows by DTDesign, HTML File by Burak Kucukparmaksiz, mouse click by Tomas Knopp, Keyboard by Arthur Shlain