

Fill in the B I a n k s: Empirical Analysis of the Privacy Threats of Browser Form Autofill

Xu Lin, Panagiotis Ilia, Jason Polakis
University of Illinois at Chicago

xlin48@uic.edu



November 10, 2020



Nobody likes to fill out forms

The image displays a collage of various web forms, illustrating the frustration of filling them out. The forms are layered and semi-transparent, showing different sections of a user interface. A blue box highlights the 'Sign Up' section with the text 'It's free and always will be.' A pink box highlights the 'Billing Address' section. A white box highlights the 'Shipping Address' section. A light blue box highlights the 'Contact us' section. A red box highlights a 'required.' label. A blue box highlights a 'DOWNLOAD NOW' button. A white box highlights a privacy policy notice.

Sign Up

It's free and always will be.

Shipping Address

First Name Last Name

Address Apt

City State ZIP Code

Email address for receipt Primary Phone

CONTINUE

Billing Address

Street

Number

Zip code

Shipping Address

Street

Number

Zip code

Contact us

Name Company Name

Phone Number Email Address

Which describes you best?

Comments

Submit

required.

DOWNLOAD NOW

We're committed to your privacy. HubSpot uses the information you provide to us to contact you about our relevant content, products, and services. You may unsubscribe from these communications at any time. For more information, check out our [Privacy Policy](#).

Form autofill

New Member Registration

First Name *

John
123 Example Street

Clear form

Manage addresses...

Chicago

Last Name *





Addree Line2

State/Province * **Zip/Postal Code ***


Phone *

Email Address *

Credit card

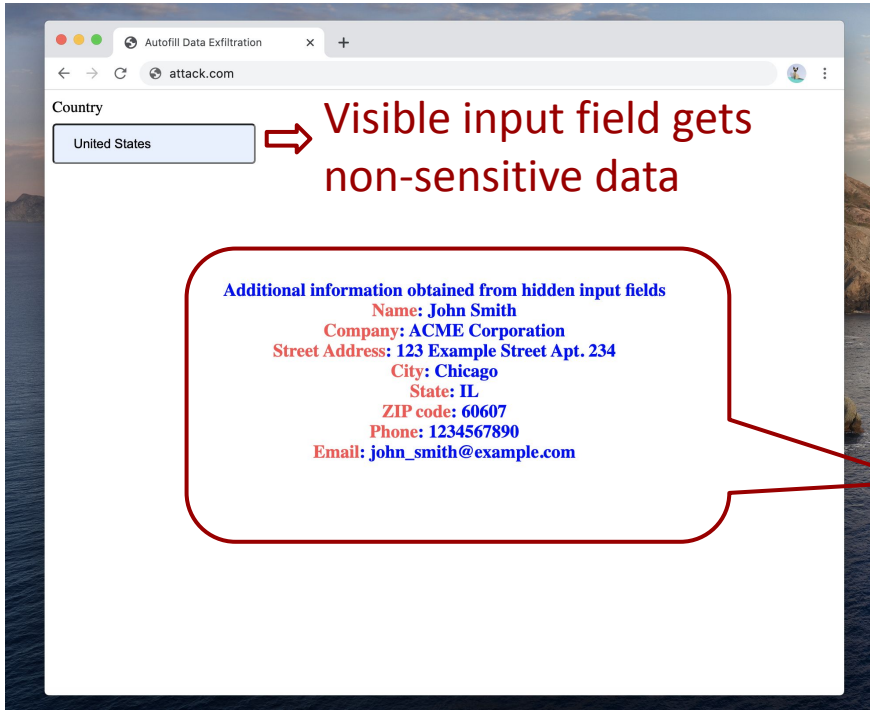
Card number

 Visa •••• 2345
Expires on 06/20

Problem: Stealthy data exfiltration

- Malicious websites can obtain sensitive user data
 - Without the user's **knowledge** or **consent**
- We demonstrate 2 types of attacks:
 - Using **visually hidden** form elements
 - Exploiting **autocomplete preview** functionality

Visually hidden elements



Country
United States

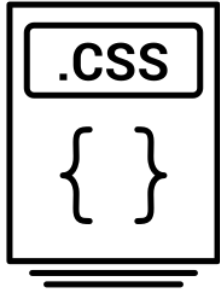
⇒ Visible input field gets non-sensitive data

Additional information obtained from hidden input fields
Name: John Smith
Company: ACME Corporation
Street Address: 123 Example Street Apt. 234
City: Chicago
State: IL
ZIP code: 60607
Phone: 1234567890
Email: john_smith@example.com

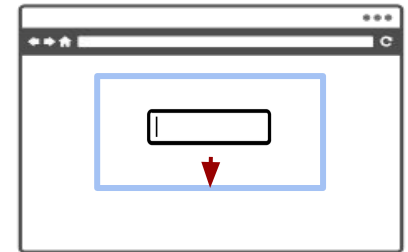
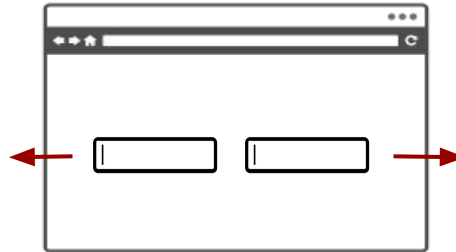
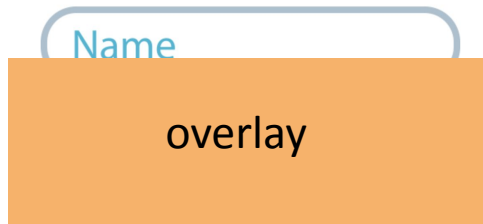
- Hidden form elements in the page
- Filled automatically by browsers when autofill is triggered

Hidden input fields get sensitive data

Visually hidden elements



- CSS display property
- CSS visibility property
- CSS opacity property
- Covered by overlay
- Non-effective size
- Off-screen placement
- Ancestor's overflow

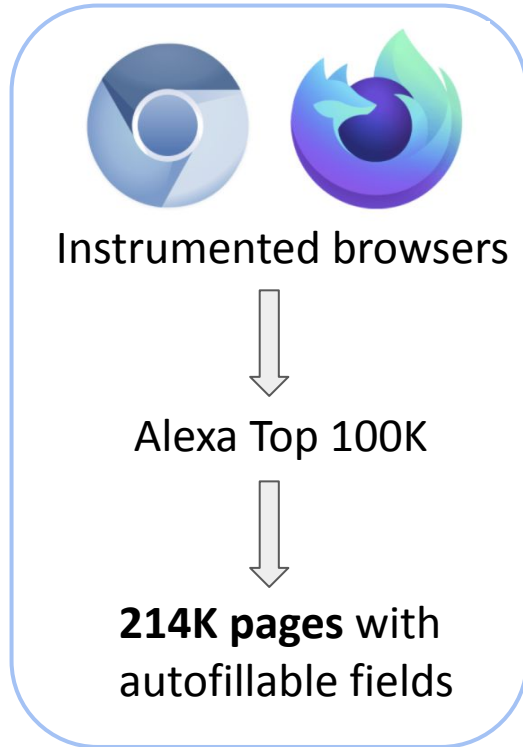


```
{overflow: hidden;}
```

Visually hidden elements

Techniques	Firefox	Chrome	Brave	Edge	Safari	Opera
CSS Display	✓	✗	✗	✗	✗	✗
CSS Visibility	✓	✗	✗	✗	✗	✗
CSS Opacity	✓	✓	✓	✓	✓	✓
Covered by overlay	✓	✓	✓	✓	✓	✓
Non-effective size	✓	✓	✓	✓	✓	✓
Off-screen placement	✓	✓	✓	✓	✓	✓
Ancestor's overflow	✓	✓	✓	✓	✓	✓

Visually hidden elements - Measurement



	Firefox	Chrome
Sites w/ autofilled forms	21,589	31,621
Sites w/ hidden fields	24.52%	5.82%

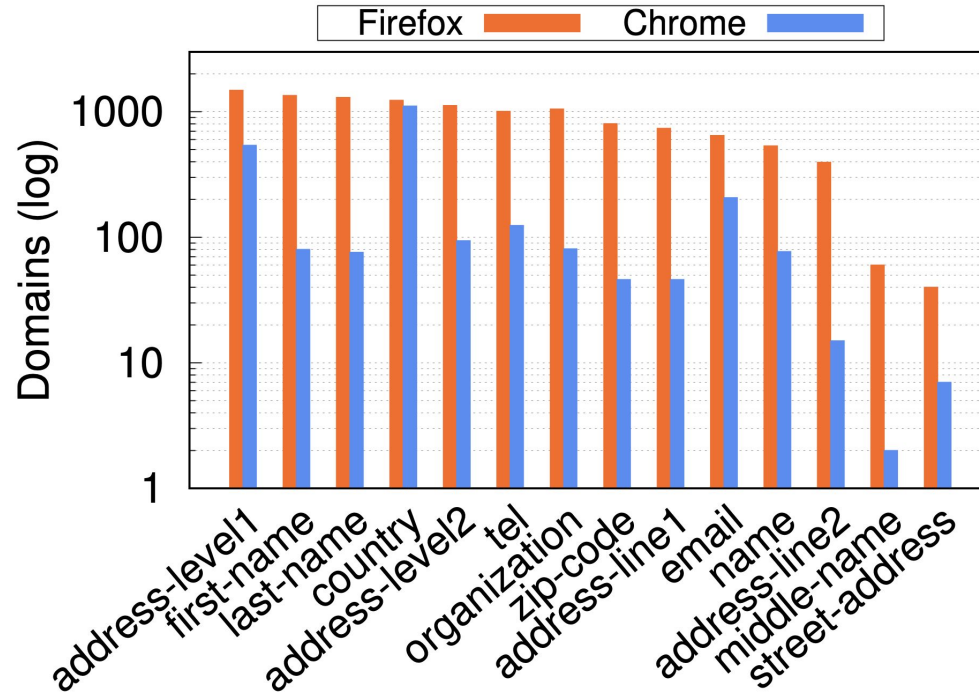
- **Chrome** fills forms in **46.5%** more websites
- **Firefox** fills almost **3x** forms with **hidden fields**

Concealment techniques

Technique	Firefox		Chrome	
	Domains	Fields	Domains	Fields
display_none_ancestor	9,177	12,675	692	1,111
display_none	1,271	2,134	468*	758*
covered	1,129	1,554	769	1,119
visibility_hidden	109	211	117*	143*
off_screen	94	131	249	497
off_ancestors_overflow	88	131	91	144
non_effective_size	61	74	53	75
transparent_ancestor	23	42	75	123
transparent	11	11	27	43
visibility_hidden_ancestor	1	1	-	-

*Chrome only autofills <select> fields hidden with these techniques.

Types of hidden autofilled fields



Cautious users may avoid using autofill.

Are they safe?

Autofill preview attack

- Does not require users to trigger autofill
 - Runs when user clicks on a field and values are previewed
- Chain together several techniques to bypass browsers' restrictions
 - Field-type mismatch
 - Side-channel leakage
 - Dynamic element replacement

Autofill preview attack

First Name *	Last Name *
<input type="text"/>	<input type="text"/>
Address2 *	Email Address
<input type="text"/>	<input type="text"/>
Street Address	City *
<input type="text"/>	<input type="text"/>

- Preview values are displayed in overlay fields that are **not** part of the DOM
 - **Not** accessible to the page (i.e., through JavaScript)

Autofill preview attack: Field-type mismatch

Email Address

<input> element

Your email..

Email Address

<select> element

-- select your email --

Email Address

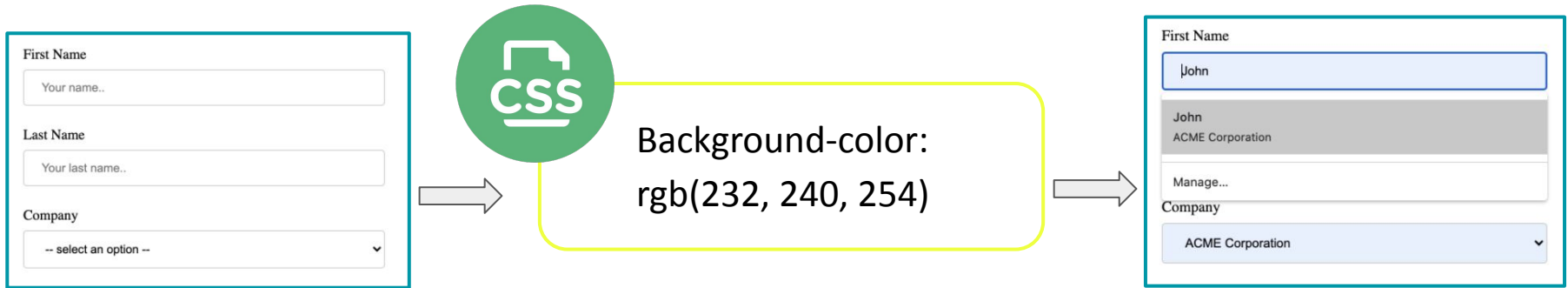
john_smith@example.com

```
<html>  
<select id="email" autocomplete="email">  
  <option>john@example.com</option>  
  <option>smith@example.com</option>  
  <option>john_smith@example.com</option>  
</select>  
</html>
```



Autofill preview attack: Side-channel leakage

- 22 style properties that change when a *matching* value is previewed
 - Accessible through JavaScript
- Reveals that a value is previewed, but it does not reveal the actual value.



Autofill preview attack: Value inference

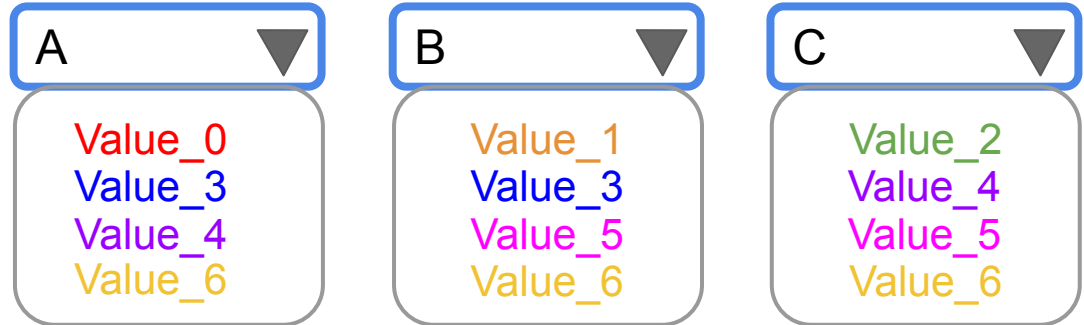
How the attacker can infer the preview value?

⇒ **By replicating values across multiple menus**

1 drop-down menu: Value_0(A),
Value_1(B), Value_2(C)

2 drop-down menus: Value_3(AB),
Value_4(AC), Value_5(BC)

3 drop-down menus:
Value_6(ABC)



Autofill preview attack: Value inference

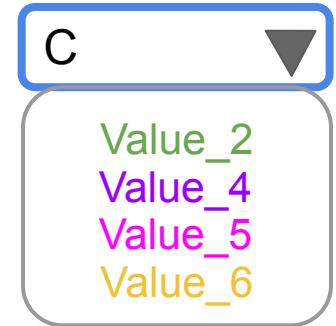
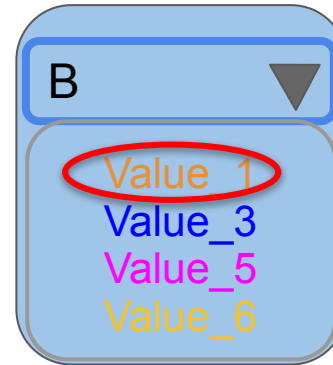
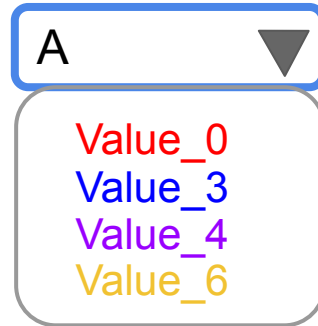
How the attacker can infer the preview value?

⇒ **By replicating values across multiple menus**

1 drop-down menu: Value_0(A),
Value_1(B), Value_2(C)

2 drop-down menus: Value_3(AB),
Value_4(AC), Value_5(BC)

3 drop-down menus:
Value_6(ABC)



Autofill preview attack: Value inference

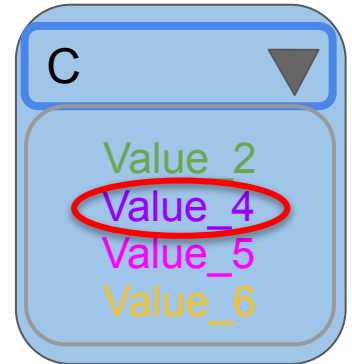
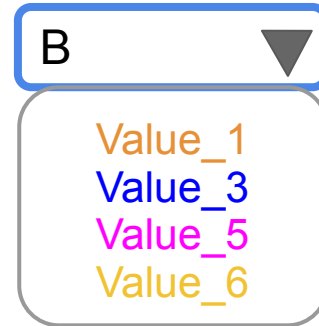
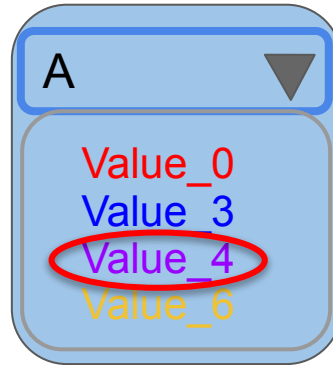
How the attacker can infer the preview value?

➔ **By replicating values across multiple menus**

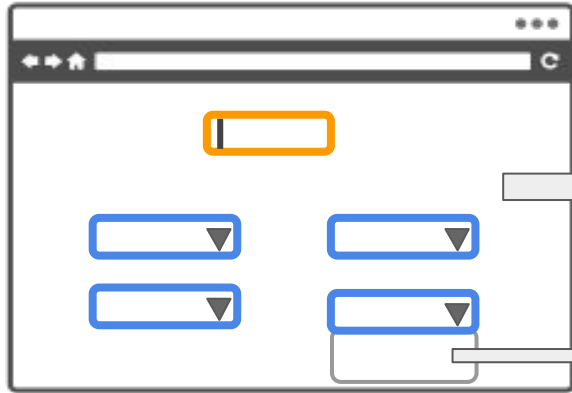
1 drop-down menu: Value_0(A),
Value_1(B), Value_2(C)

2 drop-down menus: Value_3(AB),
Value_4(AC), Value_5(BC)

3 drop-down menus:
Value_6(ABC)



Autofill preview attack: Browser Constraints



Autofill up to **200**
form elements (199
drop-down menus)

up to 512 entries

Phone

1234567890

Your phone..

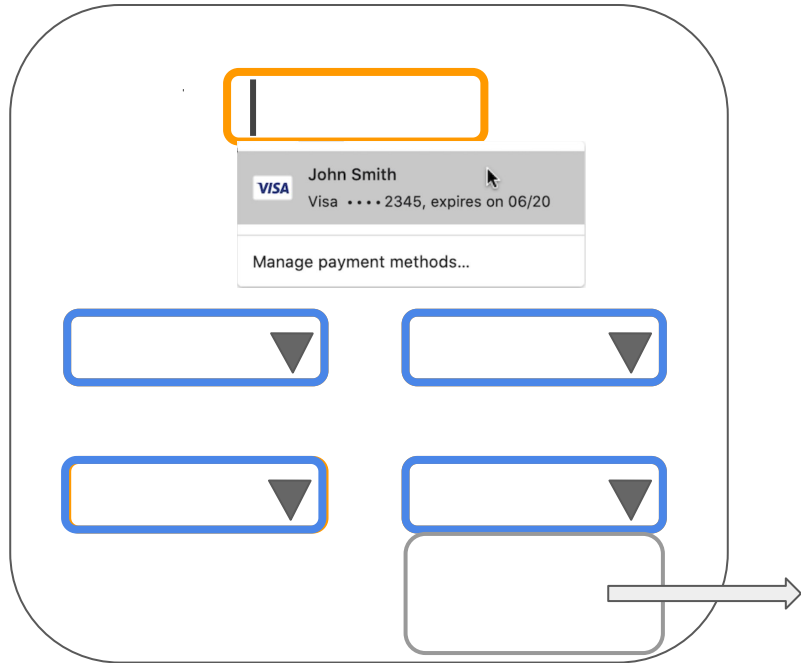
Your phone..

-- select your phone --

Not autofillable

- Size constraints:
 - Max number of candidate values: **40,662**
- Type constraints:
 - Do not support *credit card numbers* in drop-down menus.
 - Only autofill the first element of *phone number* type

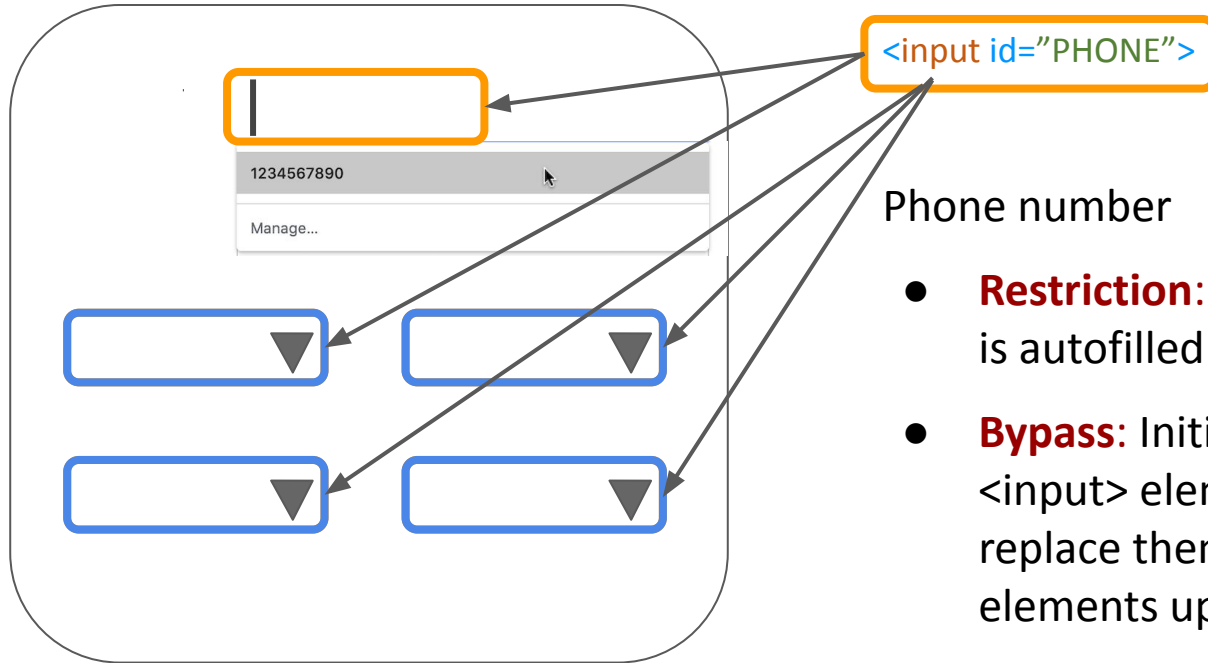
Dynamic element replacement



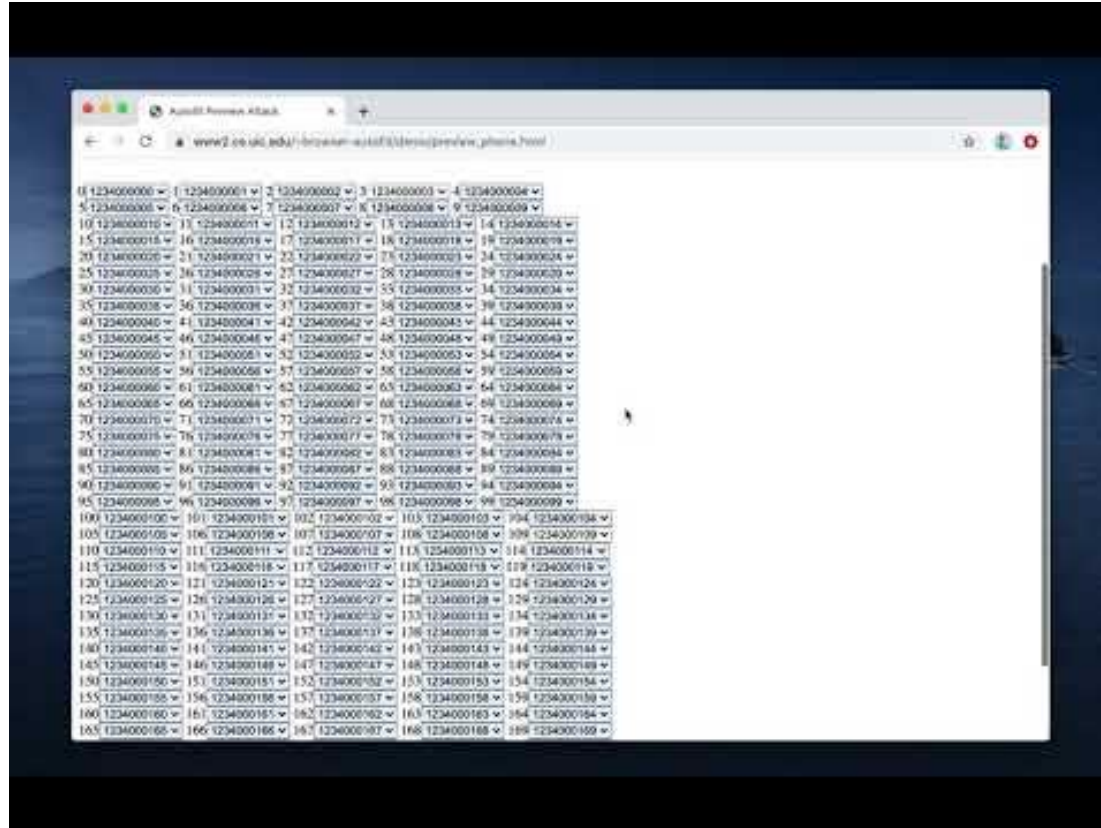
Credit card number

- **Restriction:** not autofillable in drop-down menus
- **Bypass:** initially place `<input>` elements, dynamically replace them with `<select>` elements upon click
- Unlimited number of entries in each drop-down menu

Dynamic element replacement



Autofill preview attack: Demo

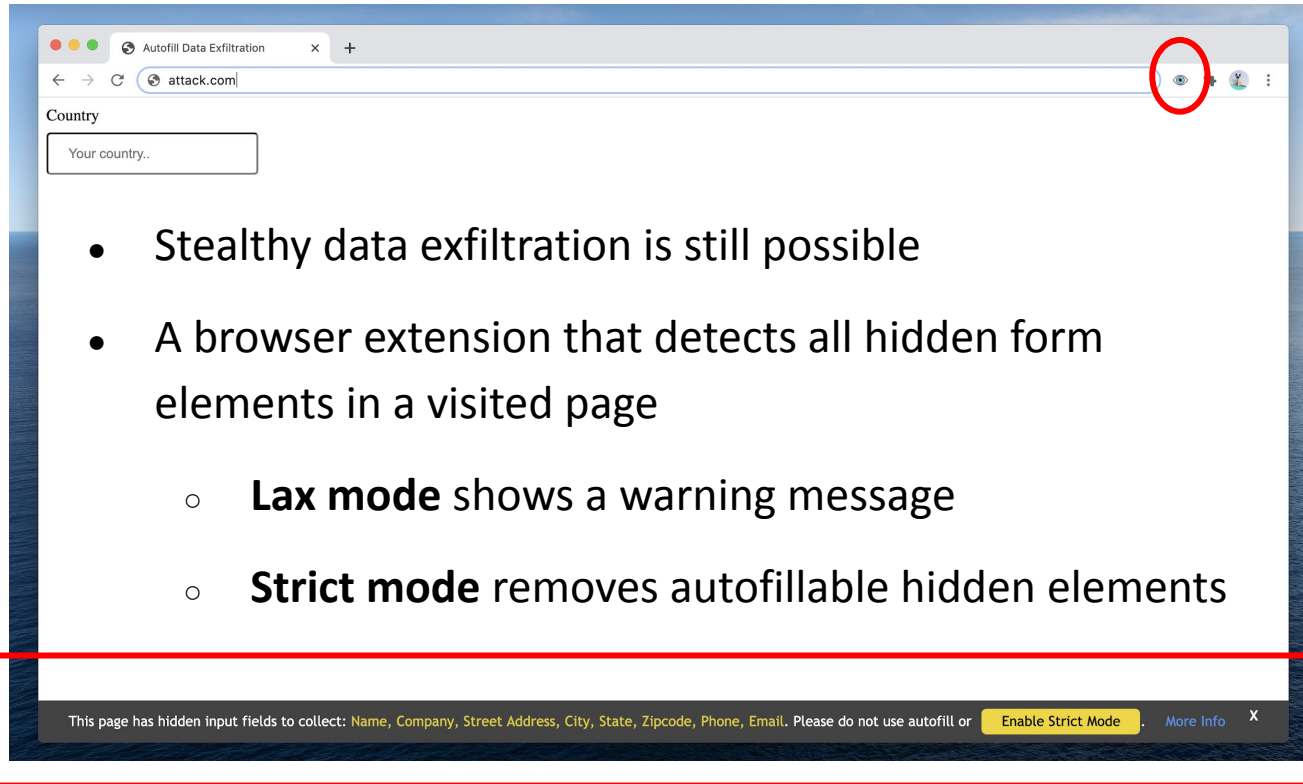


Autofill preview attack

- Affects all chromium-based browsers
- Works in incognito mode
- Bypasses **probing size limitations** for all types of information
- Probes 100k candidate values in 4-5 sec (desktops) and 5-6 sec (laptops)
- Disclosed our findings to all affected browsers
 - Assigned vulnerabilities CVE-2020-6521 and CVE-2021-21181.
 - Chrome fixed dynamic element replacement



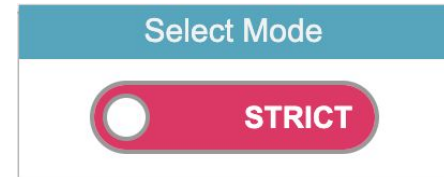
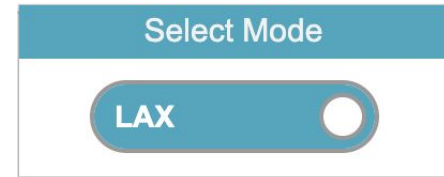
Countermeasure: Browser extension



Country

- Stealthy data exfiltration is still possible
- A browser extension that detects all hidden form elements in a visited page
 - **Lax mode** shows a warning message
 - **Strict mode** removes autofillable hidden elements

This page has hidden input fields to collect: Name, Company, Street Address, City, State, Zipcode, Phone, Email. Please do not use autofill or [Enable Strict Mode](#) [More Info](#) [X](#)



Summary

- Explored how form autofill can be exploited for stealthy data exfiltration
 - Several techniques for concealing form elements
 - Large-scale study on Alexa 100k websites
- Novel and severe side-channel attack that exploits autofill preview
 - Does not require autofill to be triggered
- Implemented and released a browser extension that prevents our attacks
- Data, demos, and code are available at <https://www.cs.uic.edu/~browser-autofill/>.