CSE 484 / CSE M 584 Computer Security: Lab 2 & Click Jacking

TA: Thomas Crosley
tcrosley@cs

Thanks to Franzi Roesner, Adrian Sham, and Vitaly Shmatikov for many previous slides

Logistics / Reminders

- Submit account info for Lab #2
  - Link: http://goo.gl/forms/rXbXqXKWdY
- Homework #2 due **tomorrow** (8pm).
- Next office hour:
  - Kevin and Thomas: 2-3pm
- Lab #2: Web security
  - Should be out tomorrow

XSS review

- Cross-site scripting (XSS) is a type of computer security vulnerability typically found is web applications.
- Allows the attacker to inject JavaScript into web pages viewed by other users.
- JavaScript can do a lot of things, like reading cookies and ex-filtrating them.
- Sanitize/validate your input
- Browser detection

PHP review

- A **server**-side programming language
- File extension is .php
- Before a webpage is sent to you, PHP code is executed by the server
- You won't see the PHP code, only html
- PHP can be use to set and read cookies for authentication
- You will need a basic PHP script to receive captured cookies

Quick demo of XSS

Back story to Lab #2

- You finally decide to show your click-happy Computer Security TAs who's da boss.
- Use XSS attacks to steal your TA's cookies, and therefore access your gradebook to change your grade.
- Use a SQL Injection to add yourself to Franzi's good list.

Basic setup

- Give the TAs (codered.cs) a link with a XSS vulnerability.
- TAs will ‘visit’ this link, and cookie will be stolen.
- The process of stealing cookie involves sending it to a place you control.
- Save the cookie, read it, and use it to log in and change your grade.
- Easy!

What you will need

- Firefox, latest version should be OK
  - Chrome ~~***might***~~ won't work
- Firebug add-on for Firefox
- Setup a location to collect your ~~stolen~~ liberated cookies
  - Good place is homes.cs, FAQ here: https://homes.cs.washington.edu/FAQ.html

Overview of setup

codered.cs

Hacker (you)

homes.cs

Tips

- Be mindful of Same Origin Policy
  - Don't redirect codered
- Run JavaScript locally before sending to codered
- When URL encoding, be careful of new-lines in XSS
  - Browser might stop executing at newline
- Talk to us if something feels wrong / confusing

Click Jacking

- Clickjacking happens when an attacker uses different techniques to hijack clicks meant for their page and routing them to another
- Multiple techniques
  - Transparent UI elements on top of a button or link
  - Timing based attacks

https://www.owasp.org/index.php/Clickjacking

Example

- Video of click jacking
- https://www.youtube.com/watch?v=9V4_emKyAg8
- User is asked to play a game
- Button is quickly switched to a ‘save’ button

- Following slides by Vitaly Shmatikov
- http://www.cs.utexas.edu/~shmat/courses/cs361s/clickjack.ppt

Clickjacking (UI Redressing)

[Hansen and Grossman 2008]

- Attacker overlays multiple transparent or opaque frames to trick a user into clicking on a button or link on another page

- Clicks meant for the visible page are hijacked and routed to another, invisible page

Clickjacking in the Wild

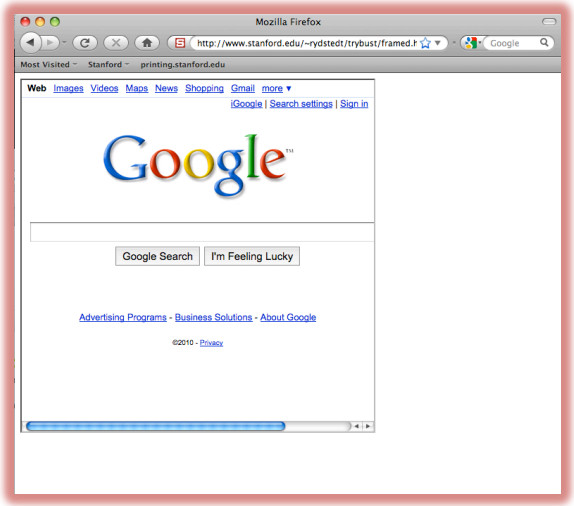
- Google search for “clickjacking” returns 624,000 results… this is not a hypothetical threat!
- Summer 2010: Facebook worm superimposes an invisible iframe over the entire page that links back to the victim's Facebook page
  - If victim is logged in, automatically recommends link to new friends as soon as the page is clicked on
- Many clickjacking attacks against Twitter
  - Users send out tweets against their will

It's All About iFrame

- Any site can frame any other site

  <iframe
    src="http://www.google.com/..."></iframe>

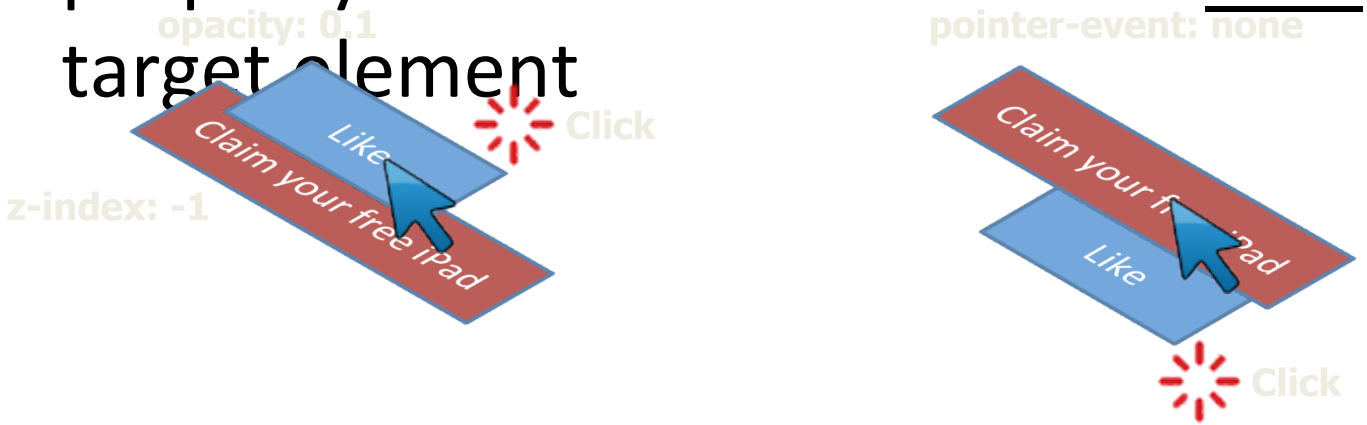
- HTML attributes
  - Style
  - Opacity defines visibility percentage of the iframe
    - 1.0: completely visible
    - 0.0: completely invisible



Hiding the Target Element

["Clickjacking: Attacks and Defenses"]

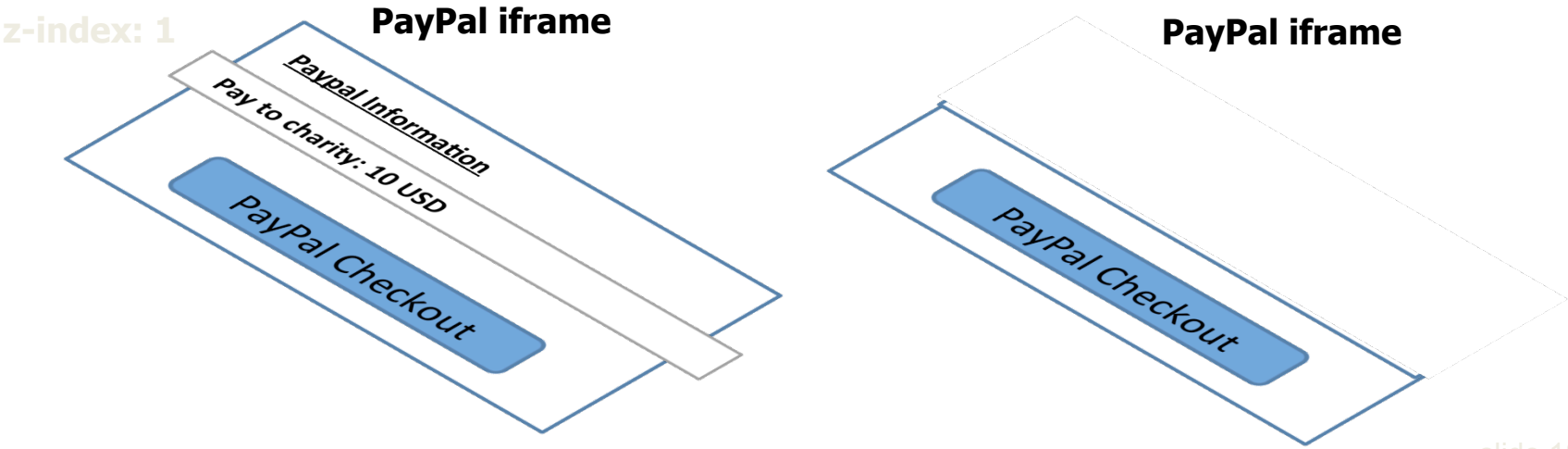
- Use CSS `opacity` property and `z-index` property to hide target element and make other element float under the target element
- Using CSS `pointer-events: none` property to cover other element over the target element



Partial Overlays and Cropping

["Clickjacking: Attacks and Defenses"]

- Overlay other elements onto an iframe using CSS `z-index` property or Flash Window Mode `wmode=direct` property
- Wrap target element in a new iframe and choose CSS position offset properties



Drag-and-Drop API

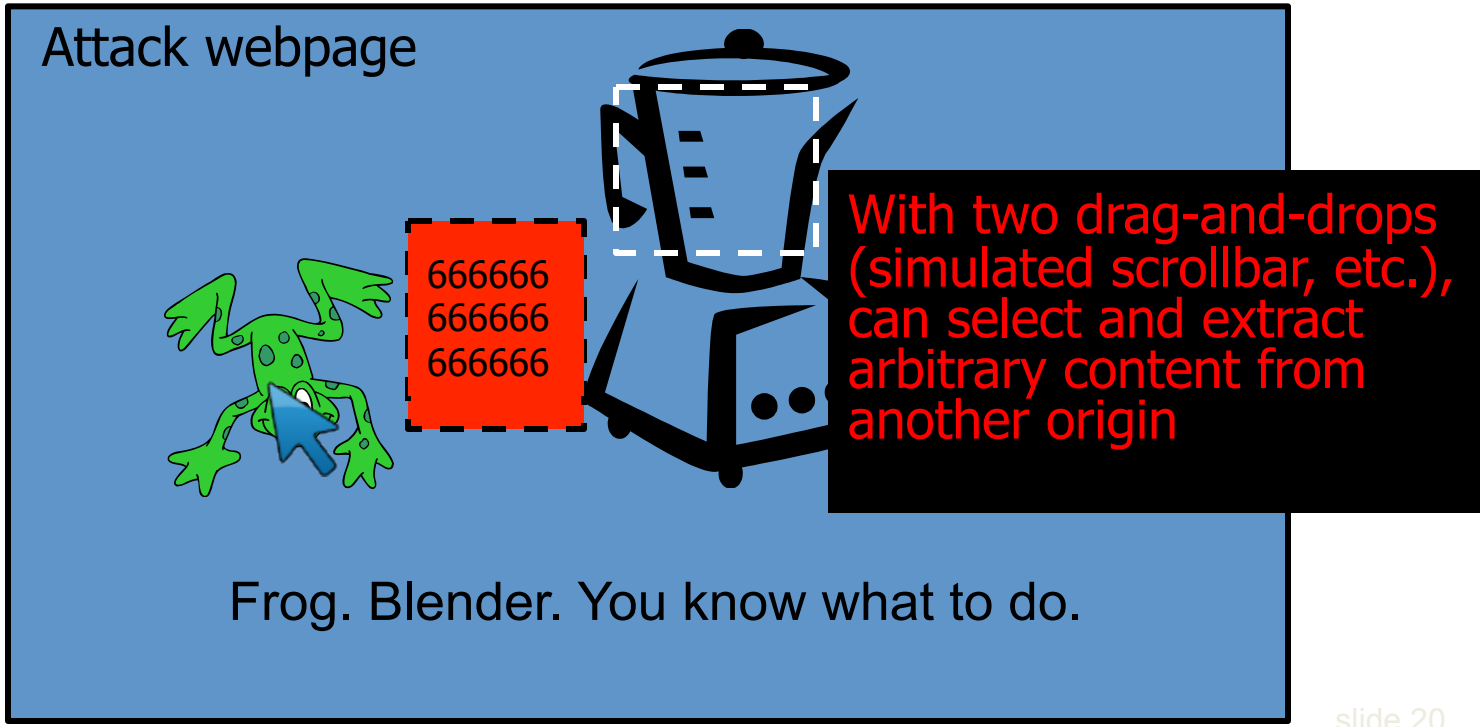
["Next Generation Clickjacking"]

- Modern browsers support drag-and-drop API
- JavaScript can use it to set data being dragged and read it when it's dropped
- Not restricted by the same origin policy:
  data from one origin can be dragged to a frame
  of another origin
  - Reason: drag-and-drop can only be initiated by user's mouse gesture, not by JavaScript on its own

Abusing Drag-and-Drop API

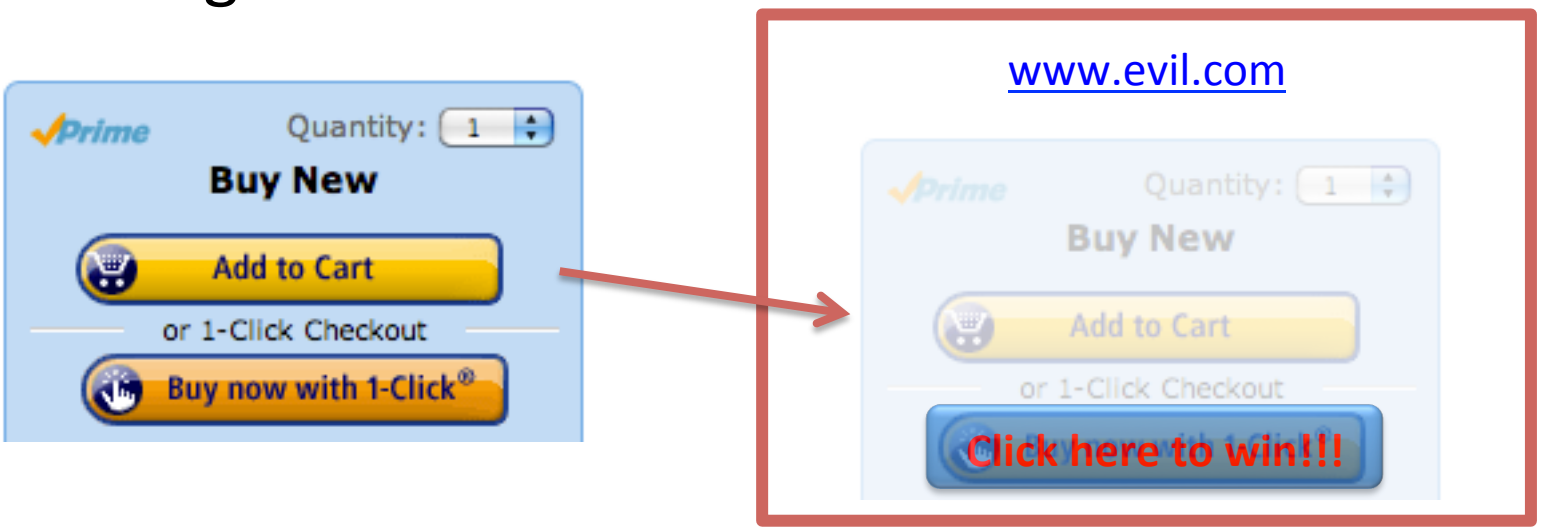
["Next Generation Clickjacking"]

1. Bait the user to click and start dragging
2. Invisible iframe with attacker's text field under mouse cursor, use API to set data being dragged
3. Invisible iframe from another origin with a form field



Clickjacking

- Trick users into interacting with sensitive user interfaces in another domain.
  - Using invisible iframes:



  - Exploit predictable user timing: http://lcamtuf.coredump.cx/ffgeo2/

Fake Cursors

["Clickjacking: Attacks and Defenses"]

- Use CSS `cursor` property and JavaScript to simulate a fake cursor icon on the screen

Real cursor icon

Fake cursor icon

cursor: none

Clickjacking using the Cursor

Figure 1: **Cursor spoofing attack page.** *The target Flash Player webcam settings dialog is at the bottom right of the page, with a "skip this ad" bait link remotely above it. Note there are two cursors displayed on the page: a fake cursor is drawn over the "skip this ad" link while the actual pointer hovers over the webcam access "Allow" button.*

[Figure from Huang et al., "Clickjacking: Attacks and Defenses", USENIX Security, 2012]

Keyboard “Strokejacking”

[“Clickjacking: Attacks and Defenses”]

- Simulate an input field getting focus, but actually the keyboard focus is on target element, forcing user to type some unwanted information into target element