

CS 253: Web Security

Administrative Stuff

- Website: `cs253.stanford.edu`
- 5-6 assignments
- 7-8 guest lectures
- Use Piazza for questions
- Share anonymous feedback (use form on website)
- Assignment 0 released tonight

LEARN YOU THE NODE.JS FOR MUCH WIN!

Select an exercise and hit Enter to begin

-
- » HELLO WORLD**
 - » BABY STEPS**
 - » MY FIRST I/O!**
 - » MY FIRST ASYNC I/O!**
 - » FILTERED LS**
 - » MAKE IT MODULAR**
 - » HTTP CLIENT**
 - » HTTP COLLECT**
 - » JUGGLING ASYNC**
 - » TIME SERVER**
 - » HTTP FILE SERVER**
 - » HTTP UPPERCASERER**
 - » HTTP JSON API SERVER**
-

HELP

CHOOSE LANGUAGE

CREDITS

CHECK FOR UPDATE

EXIT

Origins of this course

- CS 241: Secure Web Programming
- Last taught in 2011 by Dan Boneh and John Mitchell
- My favorite class at Stanford
- Inspired me to start looking for vulnerabilities
- I wanted to bring back the course – that's what CS 253 is!

Some stories

 Fill up your hard disk with ju ↶ ×

← → ⌂ ⓘ www.filldisk.com

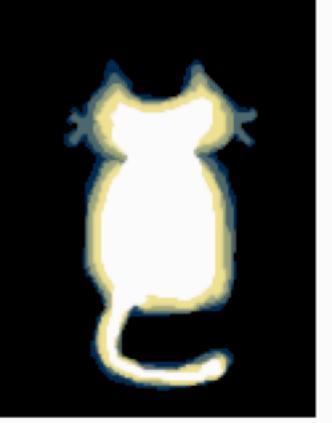
The Joys of HTML5

Introducing the new HTML5 Hard Disk Filler™ API

Oh hai there... Filling your hard disk with lots of cats...

Used 15 MB of disk space!

i frame storing 5 mb each

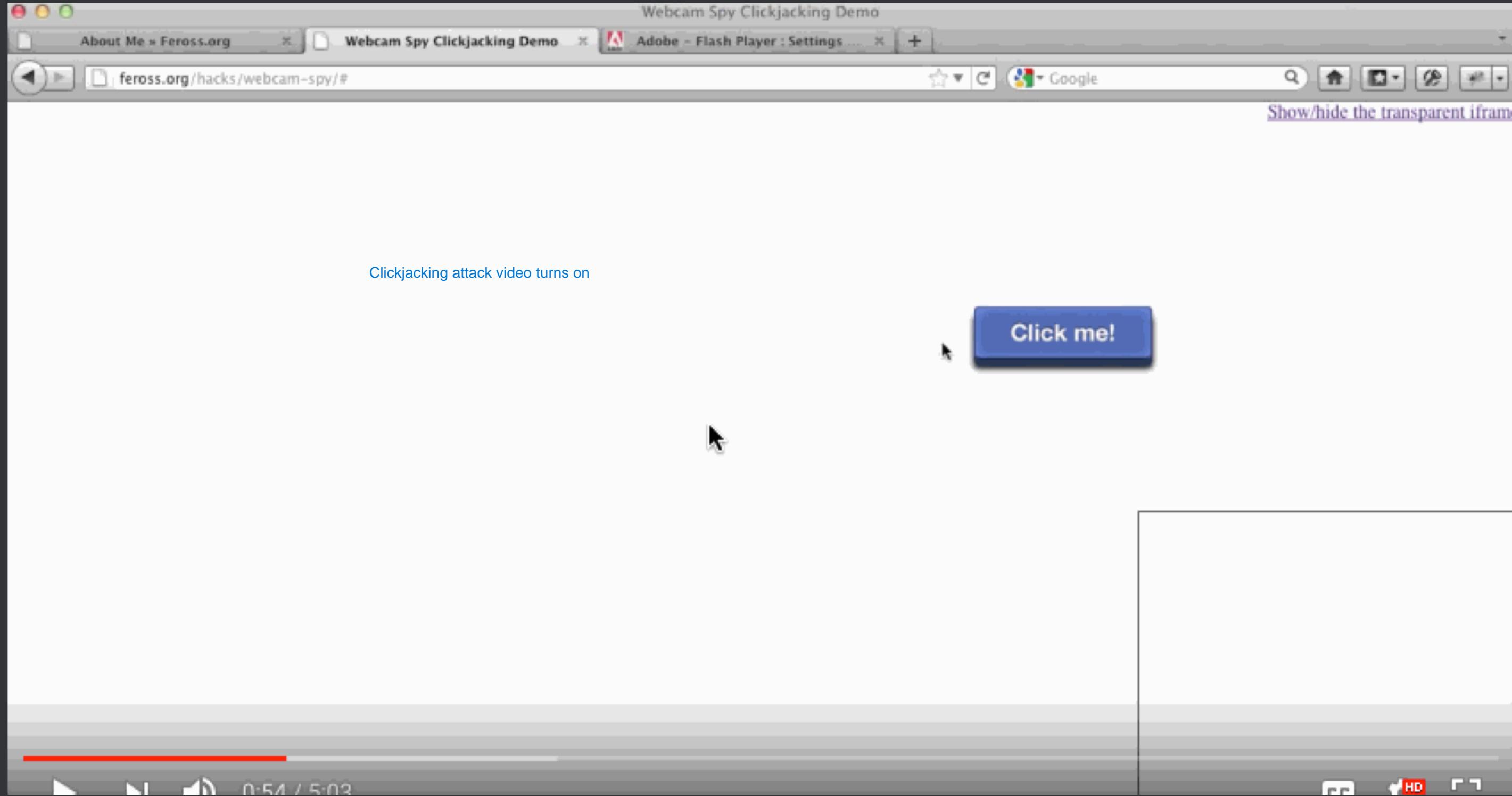


Stop the madness! (gives your disk space back)

Works in Chrome, Safari (iOS and desktop), Opera and IE. Firefox is immune to this hackery.

Every site can store 5 MB

1. **filldisk.com**
2. **filldisk.com**
3. **filldisk.com**
4. **filldisk.com**
5. **filldisk.com**
6. **filldisk.com**
7. **filldisk.com**
... and so on ...



Goal #1

The attacker mindset

Goal #2

The defender mindset

Extra credit policy

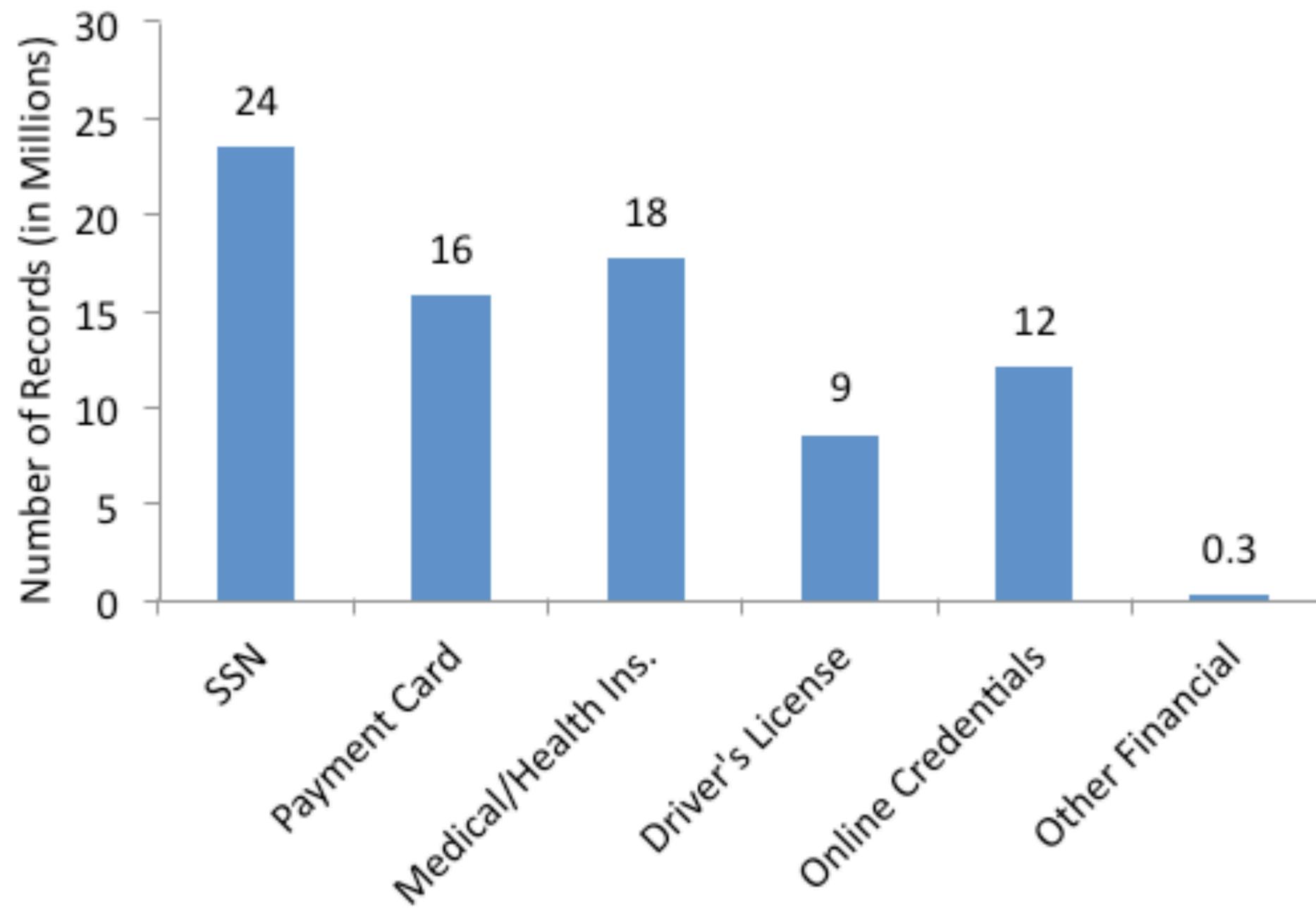
- Anyone who finds a web security vulnerability during the quarter will receive extra credit (1 to 50 points)
- **YOU MUST USE RESPONSIBLE DISCLOSURE**
 - If Stanford web app, must use Stanford Bug Bounty program
 - Do not attack servers you do own, do not destroy data
 - You are responsible for your own actions
 - If you are unsure, come speak with us

Why is computer security hard?

- Lots of buggy code
- Social engineering is very effective
- There's money to be made by finding and exploiting vulnerable systems
 - Marketplace for vulnerabilities
 - Marketplace for owned machines / stolen data
 - Many methods to profit from owned machines / stolen data

Why attack a computer system?

- Spam
 - Sent from legitimate IP address, less likely to be blocked
- Denial of service
 - Attack competitors, or seek ransom
- Infect visiting users with malware
 - Infect one server, use it to infect hundreds of thousands of clients
- Data theft
 - Steal credentials, credit card numbers, intellectual property



Why attack a computer system in 2019?

- Mine cryptocurrency
- Ransomware
- Political motivation

What is web security?

- Browser security
 - e.g. Same Origin Policy – Isolate sites from each other, while running in the same browser

Bank of America - Banking, Cre × W Wikipedia × +

← → ⌂ 🔒 wikipedia.org

WIKIPEDIA
The Free Encyclopedia

English
5 935 000+ articles

日本語
1 169 000+ 記事

Русский
1 569 000+ статей

Italiano
1 554 000+ voci

Português
1 014 000+ artigos

Español
1 546 000+ artículos

Deutsch
2 345 000+ Artikel

Français
2 141 000+ articles

中文
1 074 000+ 條目

Polski
1 360 000+ haset



javascript - Why does Google prepend while(1); to their JSON responses

← → ⌛ stackoverflow.com/questions/2669690/why-does-google-prepend-while1-to-their-json-responses

stackoverflow Products Customers Use cases Search... Log in Sign up

Home PUBLIC Stack Overflow Tags Users Jobs TEAMS What's this? First 10 Free

Why does Google prepend while(1); to their JSON responses?

Asked 9 years, 5 months ago Active 6 months ago Viewed 509k times

3910 1762

Why does Google prepend `while(1);` to their (private) JSON responses?

For example, here's a response while turning a calendar on and off in [Google Calendar](#):

```
while(1);[['u',[['smsSentFlag','false'],['hideInvitations','true'],['remindOnRespondedEventsOnly','true'],['hideInvitations_remindOnRespondedEventsOnly','false'],['Calendar ID stripped for privacy','false']],['smsVe
```

I would assume this is to prevent people from doing an `eval()` on it, but all you'd really have to do is replace the `while` and then you'd be set. I would assume the eval prevention is to

A red arrow points from the right side of the image towards the Salesforce advertisement on the right side of the Stack Overflow page. The advertisement features the Salesforce logo and the slogan "ENGINEER MUCH MORE THAN SOFTWARE." It includes a photo of two people working on a laptop.

What is web security?

- Server app security
 - Attackers can run arbitrary HTTP clients; can send anything to server

```
curl  
-d '{"user":"Alice", "permission":"admin"}'  
-H "Content-Type: application/json"  
-X POST http://example.com/data
```

What is web security?

- Client app security
 - Prevent user from being attacked while using web app locally

Stealing passwords from McDonald's users

Reflected XSS through AngularJS sandbox bypass causes password exposure of McDonald users.

By abusing an insecure cryptographic storage vulnerability ([link](#)) and a reflected server cross-site-scripting vulnerability ([link](#)) it is possible to steal and decrypt the password from a McDonald's user. Besides that, other personal details like the user's name, address & contact details can be stolen too.

Proof of Concept

Reflected XSS through AngularJS sandbox escape

McDonalds.com contains a search page which reflects the value of the search parameter (q) in the source of the page. So when we search on for example *****-test-reflected-test-***** the response will look like this:

116,228 views | Aug 31, 2019, 03:41am

Critical 'Backdoor Attack' Warning Issued For 60 Million WordPress Users



Davey Winder Senior Contributor

[Cybersecurity](#)

I report and analyse breaking cybersecurity and privacy stories

f

t

in



Feross Aboukhadijeh

Creates a new admin user with these credentials:

- Username: **wpservices**
- Password: **w0rdpr3ss**

What is web security?

- Protect the user
 - From social engineering
 - From trackers, private data being leaked

BIZ & IT —

How a college student tricked 17k coders into running his sketchy script

Infecting military and government software engineers is easier than you may think.

DAN GOODIN - 6/14/2016, 7:10 AM



Collusion for Chrome



This site is informed when you visit the following sites:

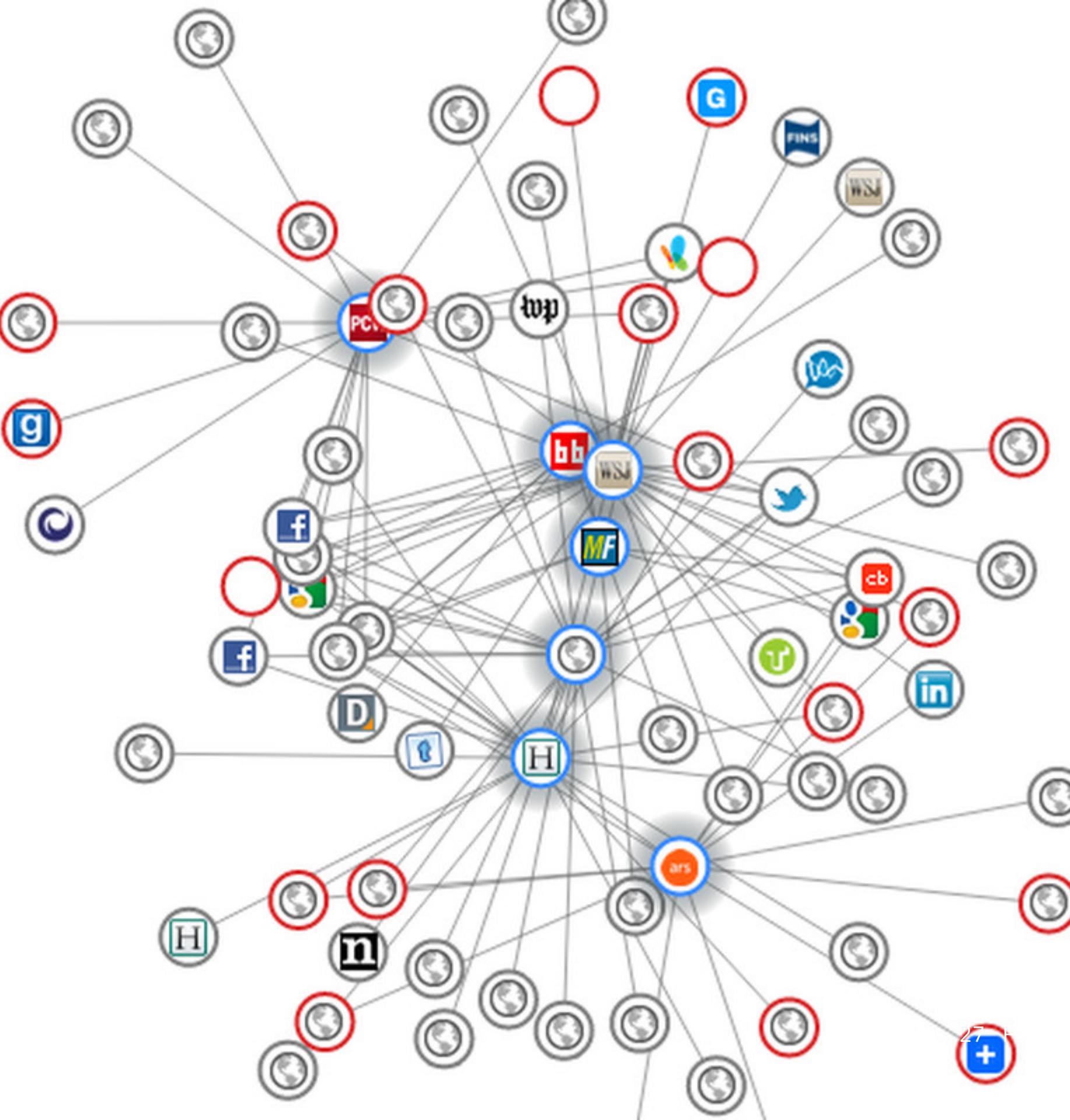
- [wsj.com](#)

[Reset the graph](#)

[Hide the sidebar](#)

[Show the instructions](#)

Privacy: We collect info about the sites you go to only to illustrate how they're connected. This info is stored locally on your computer and can be deleted anytime by resetting the graph or quitting your browser.



Why is web security hard?

- Extremely ambitious goal – Run untrusted code securely
- Different sites interacting in the same tab ("mashups")
- Low-level features; hardware access
- Desire for high performance
- APIs were not designed from first principles; evolved
- Strict backwards compatibility requirements
 - "Don't break the web"

"Modern web applications are built on a tangle of technologies that have been developed over time and then haphazardly pieced together. Every piece of the web application stack, from HTTP requests to browser-side scripts, comes with important yet subtle security consequences. To keep users safe, it is essential for developers to confidently navigate this landscape."

– Tangled Web

The browser has a seemingly impossible task

Sites – even malicious ones – can:

- Download content from **anywhere**
- Spawn worker processes
- Open sockets to a server, or even to another user's browser
- Display media in a huge number of formats
- Run custom code on the GPU
- Save/read data from the filesystem

Differing visions for the web

- Simple document viewer?
- Powerful application platform?

The web is robust

"It's all too easy to criticize, lament, and create paranoid scenarios about the 'unsound security foundations' of the web. Truth is, all of that criticism is true, and yet the web has proven to be an incredibly robust platform."

– Ilya Grigorik, Google web performance engineer

Goal #3

Learn to architect secure systems

This course

- Part 1: Browser security model, Same origin policy
- Part 2: Client security: attacks, defense
- Part 3: Server security: attacks, defense, authentication, TLS
- Part 4: Writing secure code

HTML

Introduction

This article is a review of the book *Dietary Preferences of Penguins*, by Alice Jones and Bill Smith. Jones and Smith's controversial work makes two hard-to-swallow claims about penguins:

- First, that penguins actually prefer tropical foods such as bananas and pineapple to their traditional diet of fish
- Second, that tropical foods give penguins an odor that makes them unattractive to their traditional predators

...

Introduction

This article is a review of the book **Dietary Preferences of Penguins**, by Alice Jones and Bill Smith. Jones and Smith's controversial work makes three hard-to-swallow claims about penguins:

First, that penguins actually prefer tropical foods such as bananas and pineapple to their traditional diet of fish

Second, that tropical foods give penguins an odor that makes them unattractive to their traditional predators

<h1>Introduction</h1>

<p>

This article is a review of the book Dietary Preferences of Penguins, by Alice Jones and Bill Smith. Jones and Smith's controversial work makes three hard-to-swallow claims about penguins:

</p>

First, that penguins actually prefer tropical foods such as bananas and pineapple to their traditional diet of fish

Second, that tropical foods give penguins an odor that makes them unattractive to their traditional predators

Introduction

This article is a review of the book *Dietary Preferences of Penguins*, by Alice Jones and Bill Smith. Jones and Smith's controversial work makes two hard-to-swallow claims about penguins:

- First, that penguins actually prefer tropical foods such as bananas and pineapple to their traditional diet of fish
- Second, that tropical foods give penguins an odor that makes them unattractive to their traditional predators

...

```
<!doctype html>
<html lang='en'>
<head>
  <meta charset='utf-8' />
  <title>Dietary Preferences of Penguins</title>
</head>
<body>
  <h1>Introduction</h1>

  <p>
    This article is a review of the book Dietary Preferences...
  </p>
</body>
</html>
```

Uniform Resource Locators (URLs)

`https://example.com:4000/a/b.html?user=Alice&year=2019#p2`

Protocol

Hostname

Port

Path

Query

Fragment

Ways to specify a URL

- Full URL: `2019 News`
- Relative URL: `September News`
 - Same as `http://stanford.edu/news/2019/september`
- Absolute URL: `Events`
 - Same as `http://stanford.edu/events`
- Fragment URL: `Jump to Section 3`
 - Scrolls to `` within page
 - Same as `http://stanford.edu/events#section3`

Lots of HTML tags

- ``
- `<video>, <audio>`
- `<canvas>`
- `<link>, <style>`
- `<script>`

Include CSS in a page

```
<!-- External CSS file -->
<link rel='stylesheet' href='/path/to/styles.css' />
```

```
<!-- Inline CSS -->
<style>
  body {
    color: hot-pink;
  }
</style>
```

Include JavaScript in a page

```
<!-- External JS file -->  
<script src='/path/to/script.js'></script>
```

```
<!-- Inline JS -->  
<script>  
  window.alert('hi there!')  
</script>
```

JavaScript

- Fun
- Flexible
- Immediate feedback
- Pre-installed on every device in the world
- Dev environment is pre-installed too, so easy to start writing code

Node.js

- JavaScript on the command line
- Adds built-in functions for filesystem, HTTP, DNS, and sockets
 - Stuff that belongs in a scripting language, but not a browser
- Also adds module system, binary data support
 - Less necessary because JavaScript has improved rapidly

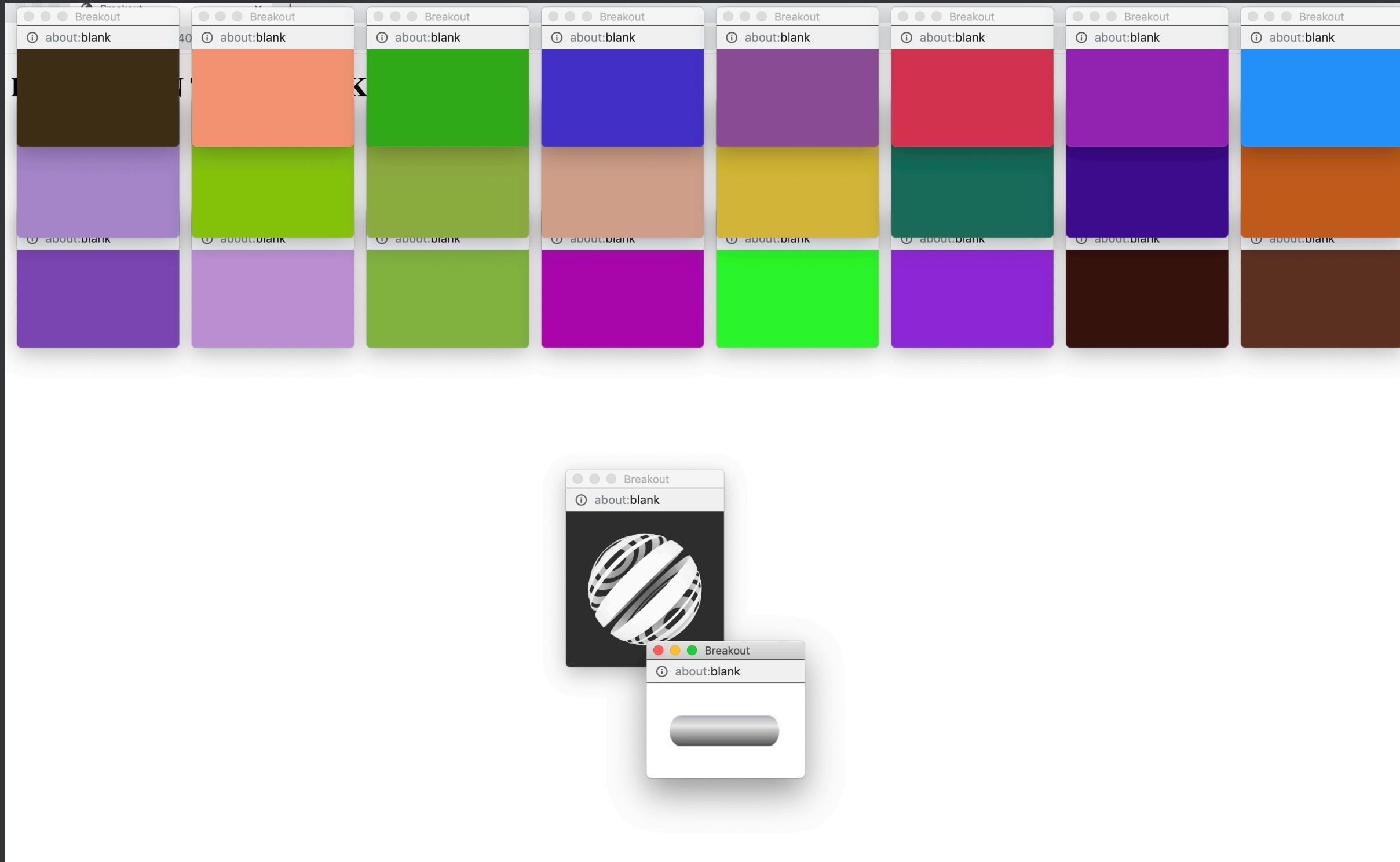
"JavaScript APIs"

- "JavaScript APIs" can come from:
 - JavaScript language specification
 - Document Object Model specification (browser)
 - Node.js built-ins
- Examples:
 - **Array**
 - **document.createElement**
 - **fs.readFile**

Crusty browser APIs

- `window.open()`
- `window.moveTo()`
- `window.resizeTo()`

Demo: Window breakout game



END