CSE 484: Computer Security and Privacy

Usability
Side-Channels

Spring 2023

David Kohlbrenner
dkohlbre@cs

Thanks to Franzi Roesner, Dan Boneh, Dieter Gollmann, Dan Halperin, David Kohlbrenner, Yoshi Kohno, Ada Lerner, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...
Logistics

• Monday is a guest lecture, not recorded!
  • Come and ask questions, it’ll be interesting stuff

• Next Friday is not recorded either, and is also an interesting topic

• No class Wednesday (24th)
  • I will have to cancel my office hours as well

• Lab 3 (patch for sploit1) due Monday

• FP part 1 (RCAs for 2 of sploit{2,3,4}) due in a week+
Usability and Security - Warnings
Challenge: Meaningful Warnings

See current designs for different conditions at https://badssl.com/. 

[Felt et al.]
Firefox vs. Chrome Warning

33% vs. 70% clickthrough rate

[Image: Firefox and Chrome warning messages]

[Felt et al.]
Experimenting w/ Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.
Experimenting with Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td>67.9%</td>
<td>17,479</td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.

![Warning Design](image)

Figure 1. The default Chrome SSL warning (Condition 1).
Experimenting w/ Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td>67.9%</td>
<td>17,479</td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td>68.9%</td>
<td>17,977</td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td>66.5%</td>
<td>18,049</td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td>68.8%</td>
<td>18,084</td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.

Figure 1. The default Chrome SSL warning (Condition 1).

Figure 4. The three images used in Conditions 2-4.

[Felt et al.]
Experimenting w/ Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td>67.9%</td>
<td>17,479</td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td>68.9%</td>
<td>17,977</td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td>66.5%</td>
<td>18,049</td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td>68.8%</td>
<td>18,084</td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td>56.1%</td>
<td>20,023</td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td>55.9%</td>
<td>19,297</td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.

Figure 2. The mock Firefox SSL warning (Condition 5).
Experimenting w/ Warning Design

<table>
<thead>
<tr>
<th>#</th>
<th>Condition</th>
<th>CTR</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Control (default Chrome warning)</td>
<td>67.9%</td>
<td>17,479</td>
</tr>
<tr>
<td>2</td>
<td>Chrome warning with policeman</td>
<td>68.9%</td>
<td>17,977</td>
</tr>
<tr>
<td>3</td>
<td>Chrome warning with criminal</td>
<td>66.5%</td>
<td>18,049</td>
</tr>
<tr>
<td>4</td>
<td>Chrome warning with traffic light</td>
<td>68.8%</td>
<td>18,084</td>
</tr>
<tr>
<td>5</td>
<td>Mock Firefox</td>
<td>56.1%</td>
<td>20,023</td>
</tr>
<tr>
<td>6</td>
<td>Mock Firefox, no image</td>
<td>55.9%</td>
<td>19,297</td>
</tr>
<tr>
<td>7</td>
<td>Mock Firefox with corporate styling</td>
<td>55.8%</td>
<td>19,845</td>
</tr>
</tbody>
</table>

Table 1. Click-through rates and sample size for conditions.

Figure 3. The Firefox SSL warning with Google styling (Condition 7).
Opinionated Design Helps!

![Warning message from a website about a security certificate not being trusted]

<table>
<thead>
<tr>
<th>Adherence</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.9%</td>
<td>4,551</td>
</tr>
</tbody>
</table>
Opinionated Design Helps!

![Security Certificate Warning]

Adherence

<table>
<thead>
<tr>
<th>Percentage</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.9%</td>
<td>4,551</td>
</tr>
<tr>
<td>32.1%</td>
<td>4,075</td>
</tr>
<tr>
<td>58.3%</td>
<td>4,644</td>
</tr>
</tbody>
</table>

[Felt et al.]
Today’s warnings (2022)
Deprecated encryption schemes

This site can’t provide a secure connection

rc4.badssl.com uses an unsupported protocol.

Error code: SSL_ERROR_NO_CIPHER_OVERLAP

- The page you are trying to view cannot be shown because the authenticity of the received data could not be verified.
- Please contact the website owners to inform them of this problem.

Learn more...
Expired certificates

Your connection is not private
Attackers might be trying to steal your information from expired.badssl.com (for example, passwords, messages, or credit cards). Learn more

Warning: Potential Security Risk Ahead
Firefox detected an issue and did not continue to expired.badssl.com. The website is either misconfigured or your computer clock is set to the wrong time.

It's likely the website's certificate is expired, which prevents Firefox from connecting securely. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

What can you do about it?
Your computer clock is set to 12/7/2022. Make sure your computer is set to the correct date, time, and time zone in your system settings, and then refresh expired.badssl.com.

If your clock is already set to the right time, the website is likely misconfigured, and there is nothing you can do to resolve the issue. You can notify the website's administrator about the problem.

Learn more...
Self-signed certificates

Your connection is not private

Attacks might be trying to steal your information from self-signed.badssl.com (for example, passwords, messages, or credit cards). Learn more

NET:ERR_CERT_AUTHORITY_INVALID

To get Chrome's highest level of security, turn on enhanced protection

Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to self-signed.badssl.com. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

Learn more...
Untrusted Root certificate

Your connection is not private

Attacks might be trying to steal your information from untrusted-root.badssl.com (for example, passwords, messages, or credit cards). Learn more

NET-ERR CERT AUTHORITY INVALID

Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to untrusted-root.badssl.com. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

What can you do about it?

The issue is most likely with the website, and there is nothing you can do to resolve it.

If you are on a corporate network or using anti-virus software, you can reach out to the support teams for assistance. You can also notify the website’s administrator about the problem.

Learn more...

5/19/2023

CSE 484 - Spring 2023
Address Bar behaviors (2022)
Does anything stand out?

• Canvas
• What makes warnings hard, especially over time?
• Why do Firefox and Chrome make different warning designs?
Case Study #2: Phishing

• **Design question:** How do you help users avoid falling for phishing sites?
A Typical Phishing Page

Weird URL
http instead of https
Safe to Type Your Password?
Safe to Type Your Password?

![Phishing website example](image)

**Bank of the West**

Gives me you pa55w0rds!

User name:

Password:

Login
Safe to Type Your Password?
Safe to Type Your Password?

“Picture-in-picture attacks”
Trained users are more likely to fall victim to this!
Phishing Warnings (2008)

Passive (IE)

Active (IE)

Active (Firefox)
Active vs. Passive Warnings

- Active warnings significantly more effective
  - Passive (IE): 100% clicked, 90% phished
  - Active (IE): 95% clicked, 45% phished
  - Active (Firefox): 100% clicked, 0% phished

[Image: Screenshots of passive and active warnings in Internet Explorer and Firefox]

[Note: Image credit: Egelman et al.]
If you don’t recognize your personalized “SiteKey”, don’t enter your Passcode.
Modern anti-phishing

• Largely driven by Google Safe Browsing
  • Browser sends 32-bit prefix of hash(url)
  • API says: good or bad
Modern warnings

Deceptive site ahead

Attackers on testsafebrowsing.appspot.com may trick you into doing something dangerous like installing software or revealing your personal information (for example, passwords, phone numbers, or credit cards). Learn more
Deceptive site ahead

Firefox blocked this page because it may trick you into doing something dangerous like installing software or revealing personal information like passwords or credit cards.

Advisory provided by Google Safe Browsing.
The page ahead may try to charge you money

These charges could be one-time or recurring and may not be obvious.
The site ahead contains malware

Attackers currently on testsafebrowsing.appspot.com might attempt to install dangerous programs on your computer that steal or delete your information (for example, photos, passwords, messages, and credit cards). Learn more
Which warning is ‘better’?

• For user security?
• For user agency?
• For user understanding?
• For... what?
Side channels
Side-channels: conceptually

• A program’s implementation (that is, the final compiled version) is different from the conceptual description

• Side-effects of the difference between the implementation and conception can reveal unexpected information
  • Thus: Side-channels
Detour: Covert-channels

• We’ll see many unusual ways to have information flow from thing A to thing B

• If this is an *intentional* usage of side effects, it is a covert channel

• *Unintentional* means it is a side-channel

• The same *mechanism* can be used as a covert-channel, or abused as a side-channel
Side Channel Attacks

• Most commonly discussed in the context of cryptosystems
• But also prevalent in many contexts
  • E.g., we discussed the TENEX password implementation
  • E.g., we discussed browser fingerprinting
Why should we care about side-channels?

• Compromises happen via ‘simple’ methods
  • Phishing
  • Straight-forward attacks

• Embedded systems \textit{do} see side-channel attacks

• “High Security” systems \textit{do} see side-channel attacks
And they are getting more impactful...

• “The Secret Network has been vulnerable to the xAPIC and MMIO vulnerabilities that were publicly disclosed on August 9, 2022. These vulnerabilities could be used to extract the consensus seed, a master decryption key for the private transactions on the Secret Network. Exposure of the consensus seed would enable the complete retroactive disclosure of all Secret-4 private transactions since the chain began. We have helped Secret Network to deploy mitigations, especially the Registration Freeze on October 5, 2022.”
Timing Side-Channels

• Duration of a program (or operation) reveals information
• TENEX case
TENEX attack (for real)

- TENEX had an early *memory paging system*

- The original attack used page faults, not timing
  - Timing would’ve also worked 😊
Timing side-channels: round 2

• Cryptographic implementations fall down
  • #1 target for timing attacks
  • Extremely common to find vulnerabilities

• “Timing Attacks on Implementations of Diffe-Hellman, RSA, DSS, and Other Systems”
  • Was very far from the last paper on the topic
Attacking cryptography with side-channels

• ANY leakage is bad
  • E.g. 1 bit of key leaking is ‘catastrophic’

• Cryptographic implementations are complex
  • Many layers of protocols
Example Timing Attacks

• **RSA**: Leverage key-dependent timings of modular exponentiations

• **Block Ciphers**: Leverage key-dependent cache hits/misses
Cache side-channels
Cache side-channels

• **Idea:** The cache’s current state implies something about prior memory accesses

• **Insight:** Prior memory accesses can tell you a lot about a program!
Cache Basics

- **Cache lines**: fixed-size units of data
- **Cache set**: holds multiple cache lines
- **Set index**: assigns cache line to cache set
- **Eviction**: removing cache lines to make room
- **L1, L2, L3**: different levels of cache
- **Inclusive**: lines in L1/L2 must also be in L3

Many thanks to Craig Disselkoen for the animations.
Cache Attacks: Structure

- Pre-Attack
- Active Attack
- Analysis

Many thanks to Craig Disselkoen for the animations.
Timing threshold
Eviction set

Timing threshold
Eviction set

Activation threshold
Victim access if
\[ \text{time} > \text{threshold} \]

Prior attack
Victim accesses targeted set

Pre-existing data
Attacker’s data
Victim’s data

Many thanks to Craig Disselkoen for the animations.
FLUSH + RELOAD

• Even simpler!

• Kick line L out of cache

• Let victim run

• Access L
  • Fast? Victim touched it
  • Slow? Victim didn’t touch it
Cache attacks wrapup

• Cache attacks are a core element of many side-channels

• Generally “assumed to work” these days

• New variations/tricks/mitigations published constantly

• Randomized caches are the current hotness