CSE 484 / CSE M 584: Computer Security and Privacy

Mobile Platform Security [finish], Usable Security [start]

Spring 2017

Franziska (Franzi) Roesner
franzi@cs.washington.edu

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

Admin

- Monday is a holiday! 

- Project checkpoint #2 **due Wed, 5/31, 11:59pm**
- Lab 3 **due Fri June 2, 8pm**

Challenges with Isolated Apps

So mobile platforms isolate applications for security, but…

1. Permissions: How can applications access sensitive resources?
2. Communication: How can applications communicate with each other?

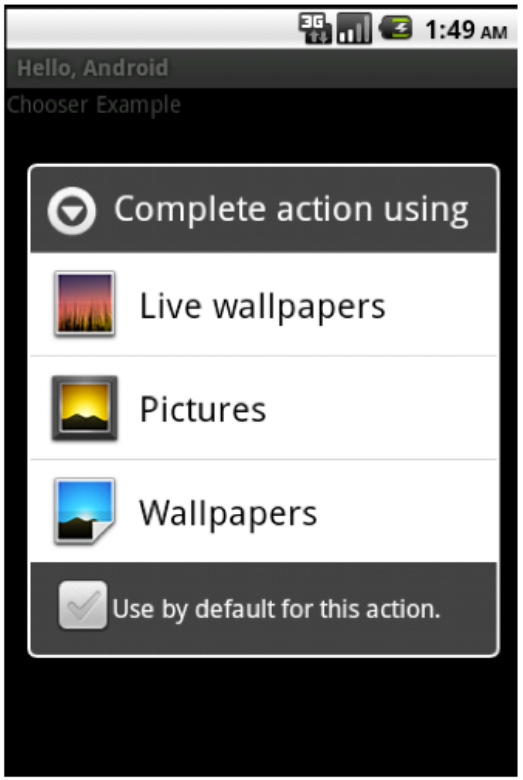
(2) Inter-Process Communication

- Primary mechanism in Android: Intents
  - Sent between application components
    - e.g., with `startActivity(intent)`
  - Explicit: specify component name
    - e.g., com.example.testApp.MainActivity
  - Implicit: specify action (e.g., ACTION_VIEW) and/or data (URI and MIME type)
    - Apps specify Intent Filters for their components.

[Chin et al.]

Unauthorized Intent Receipt

- Attack #1: Eavesdropping / Broadcast Thefts
  - Implicit intents make intra-app messages public.
- Attack #2: Activity Hijacking
  - May not always work:
- Attack #3: Service Hijacking
  - Android picks one at random upon conflict!



[Chin et al.]

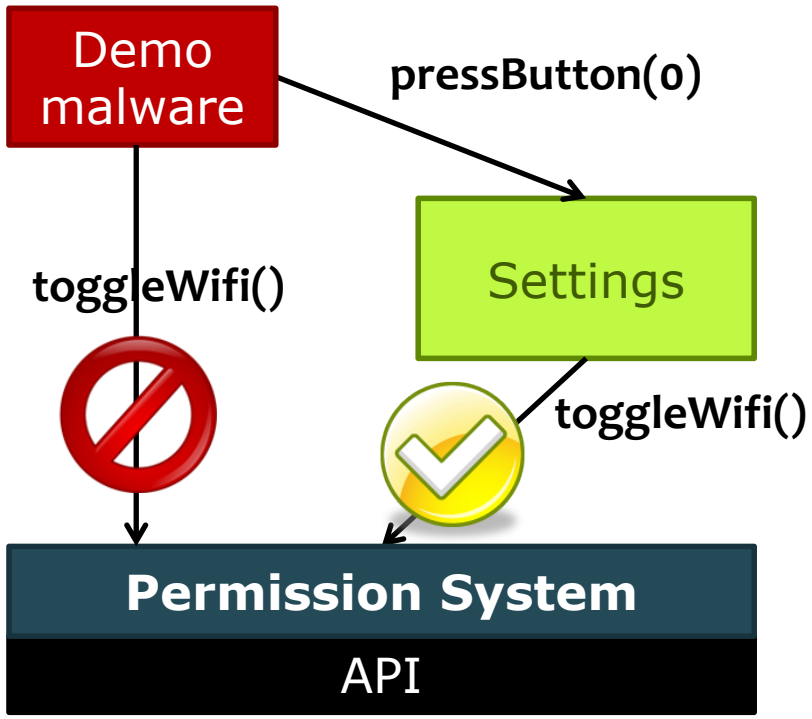
Intent Spoofing

- Attack #1: General intent spoofing
  - Receiving implicit intents makes component public.
  - Allows data injection.
- Attack #2: System intent spoofing
  - Can't directly spoof, but victim apps often don't check specific "action" in intent.

[Felt et al.]

Permission Re-Delegation

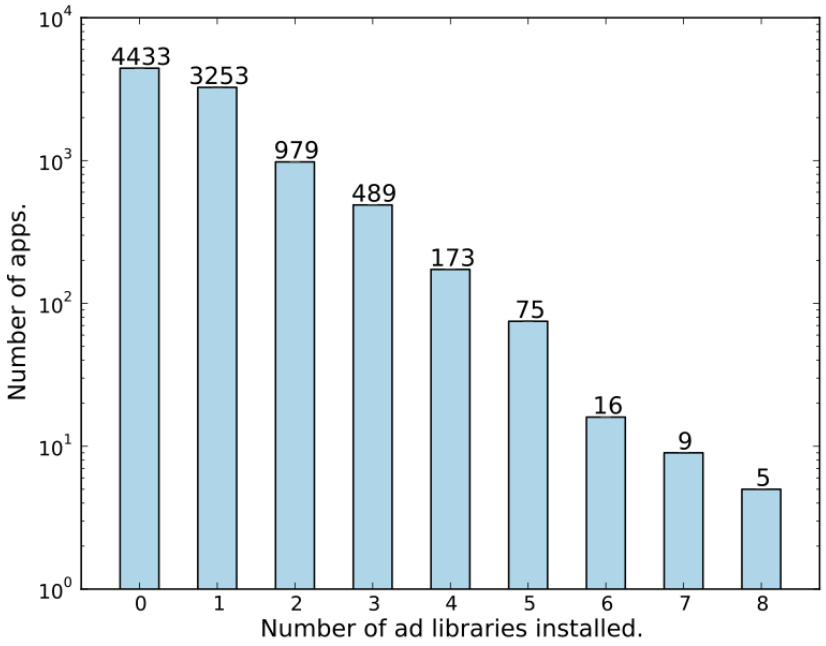
- An application without a permission gains additional privileges through another application.
- Demo video
- Settings application is deputy: has permissions, and accidentally exposes APIs that use those permissions.



More on Android…

Aside: Incomplete Isolation

Embedded UIs and libraries always run with the host application's permissions! (No same-origin policy here…)



[Shekhar et al.]

Ad from ad library

Map from Google library

Social button from Facebook library

Android Application Signing

- Apps are signed
  - Often with self-signed certificates
  - Signed application certificate defines which user ID is associated with which applications
  - Different apps run under different UIDs
- Shared UID feature
  - Shared Application Sandbox possible, where two or more apps signed with same developer key can declare a shared UID in their manifest

Shared UIDs

- App 1: Requests GPS / camera access
- App 2: Requests Network capabilities

- Generally:
  - First app can't exfiltrate information
  - Second app can't exfiltrate anything interesting
- With Shared UIDs (signed with same private key)
  - Permissions are a superset of permissions for each app
  - App 1 can now exfiltrate; App 2 can now access GPS / camera

File Permissions

- Files written by one application cannot be read by other applications
  - Previously, this wasn't true for files stored on the SD card (world readable!) – Android cracked down on this
- It is possible to do full file system encryption
  - Key = Password/PIN combined with salt, hashed

Memory Management

- Address Space Layout Randomization to randomize addresses on stack
- Hardware-based No eXecute (NX) to prevent code execution on stack/heap
- Stack guard derivative
- Some defenses against double free bugs (based on OpenBSD's dmalloc() function)
- etc.

[See http://source.android.com/tech/security/index.html]

Android Fragmentation

- Many different variants of Android (unlike iOS)
  - Motorola, HTC, Samsung, …
- Less secure ecosystem
  - Inconsistent or incorrect implementations
  - Slow to propagate kernel updates and new versions

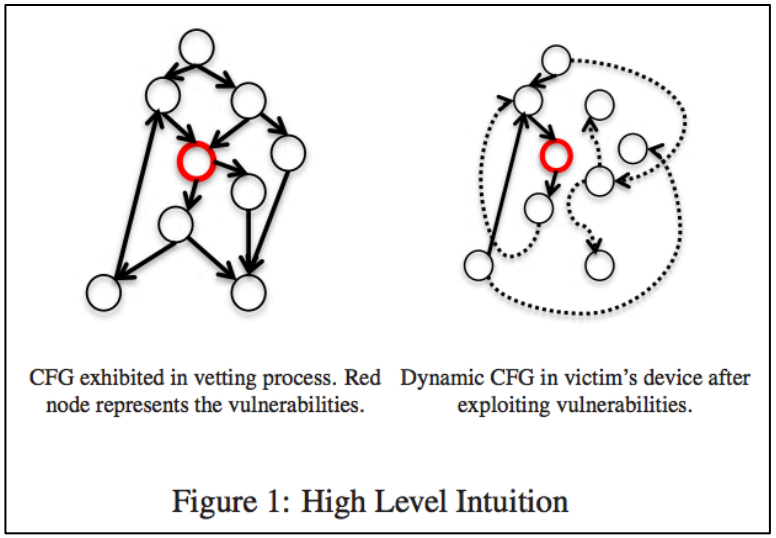
[https://developer.android.com/about/dashboards/index.html]

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.8%
4.1.x	Jelly Bean	16	3.2%
4.2.x		17	4.6%
4.3		18	1.3%
4.4	KitKat	19	18.8%
5.0	Lollipop	21	8.7%
5.1		22	23.3%
6.0	Marshmallow	23	31.2%
7.0	Nougat	24	6.6%
7.1		25	0.5%

Data collected during a 7-day period ending on May 2, 2017.
Any versions with less than 0.1% distribution are not shown.

What about iOS?

- Apps are sandboxed
- Encrypted user data
  - See recent news…
- App Store review process is (maybe) stricter
  - But not infallible: e.g., see Wang et al. "Jekyll on iOS: When Benign Apps Become Evil" (USENIX Security 2013)
- No "sideloading" apps
  - Unless you jailbreak



CSE 484 / CSE M 584: Computer Security and Privacy

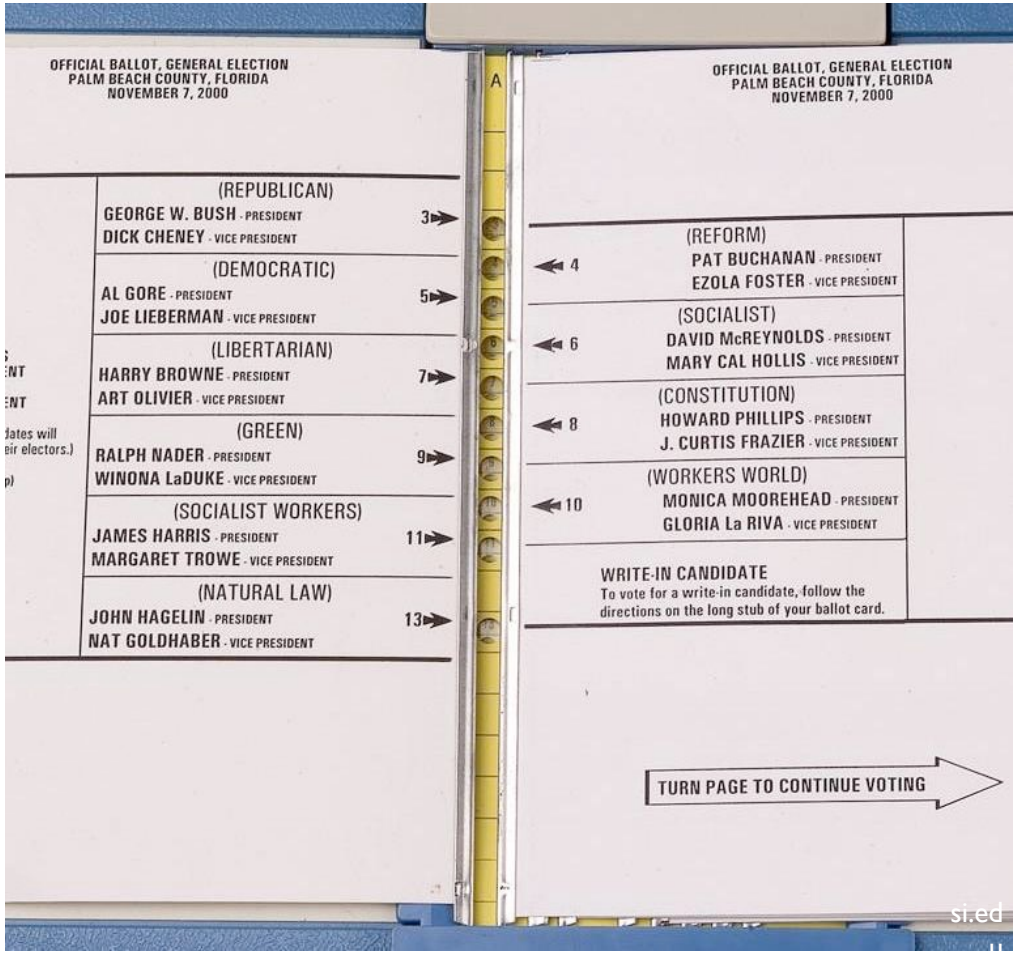
Mobile Platform Security [finish], Usable Security [start]

Spring 2017

Franziska (Franzi) Roesner
franzi@cs.washington.edu

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

Poor Usability Causes Problems



AP

Importance in Security

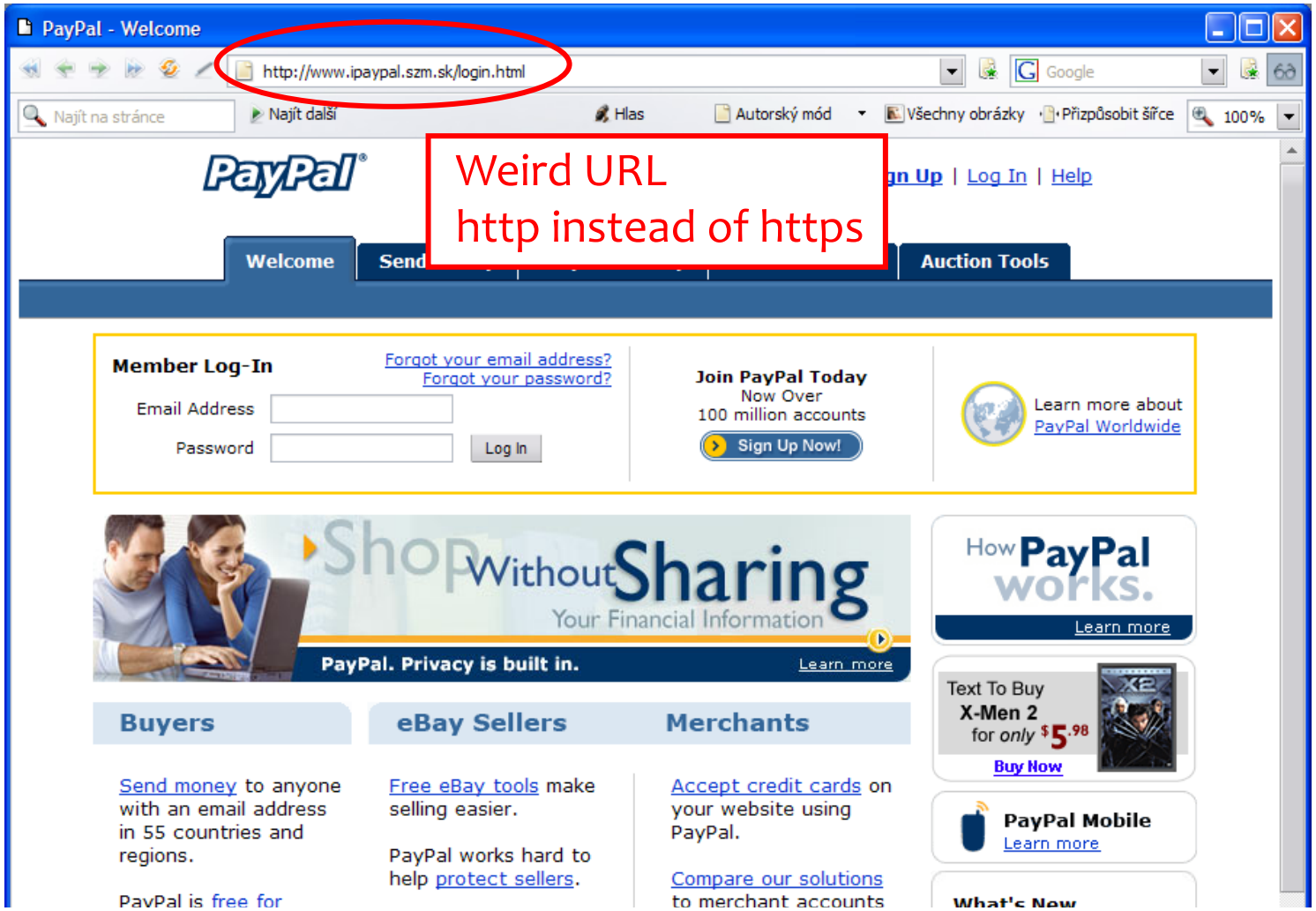
- Why is usability important?
  - People are the critical element of any computer system
    - People are the real reason computers exist in the first place
  - Even if it is **possible** for a system to protect against an adversary, people may use the system in other, **less secure** ways

Usable Security Roadmap

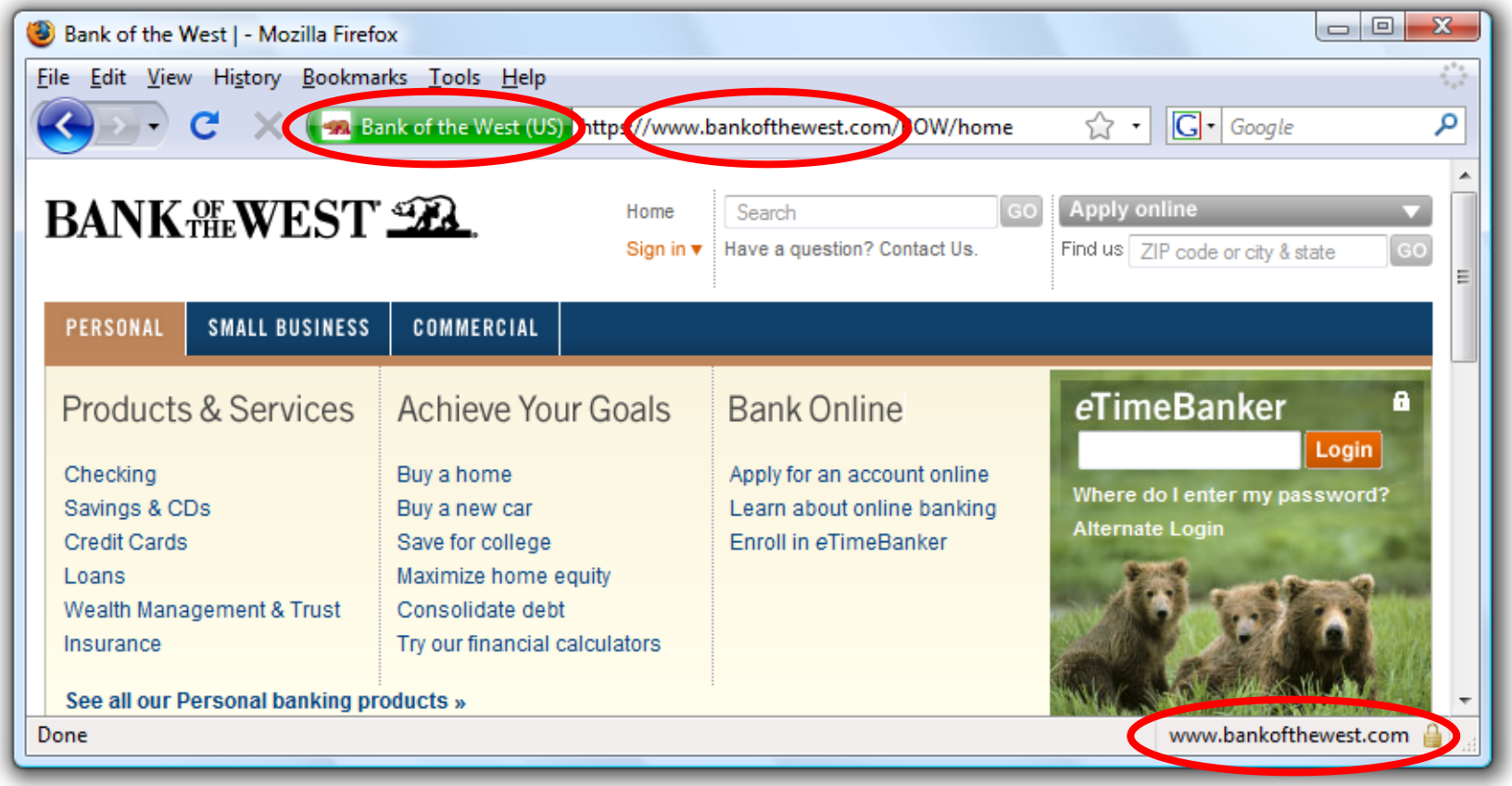
- 2 case studies
  - Phishing
  - SSL warnings
- Step back: root causes of usability problems, and how to address

Case Study #1: Phishing

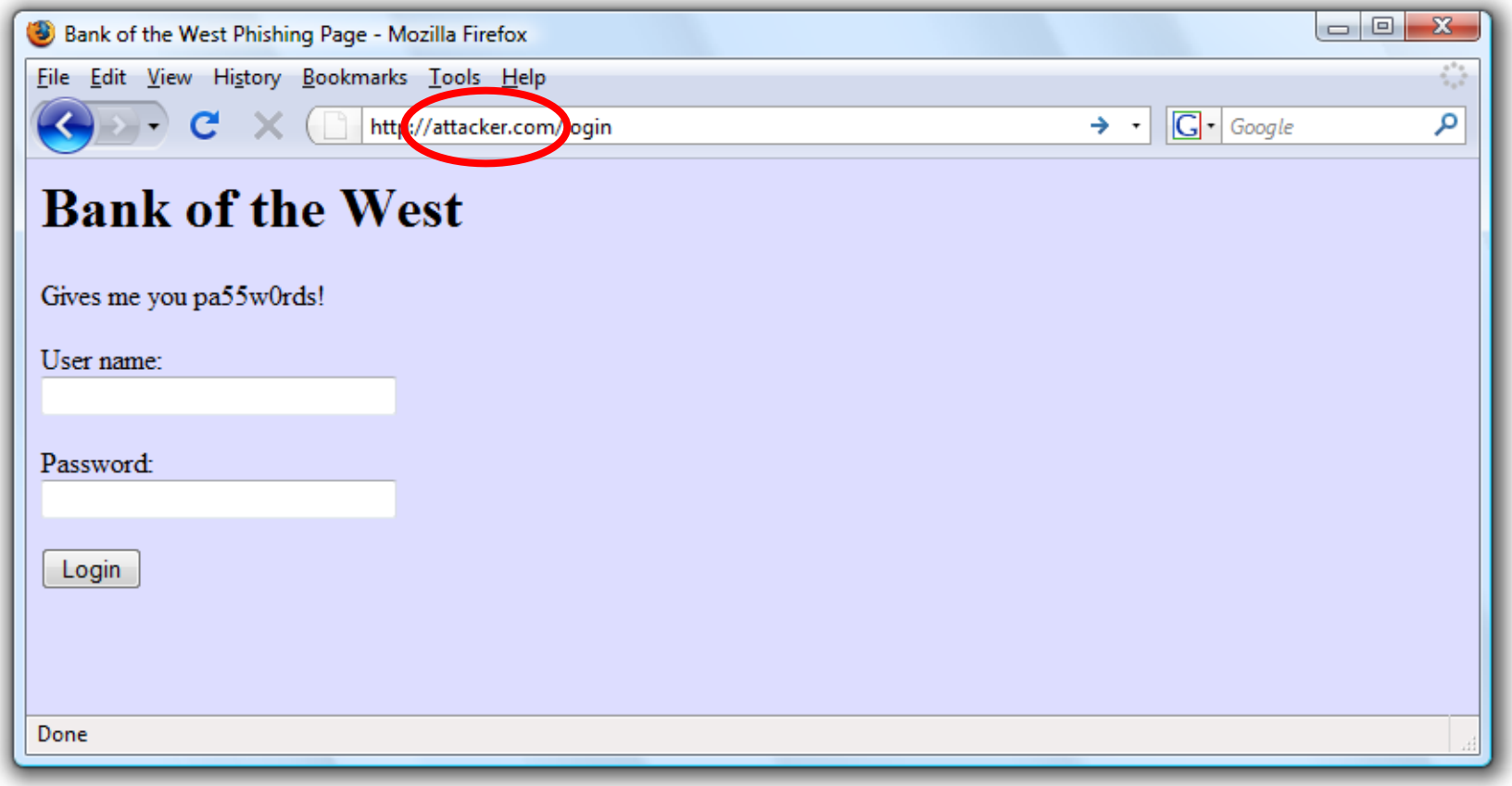
A Typical Phishing Page



Safe to Type Your Password?

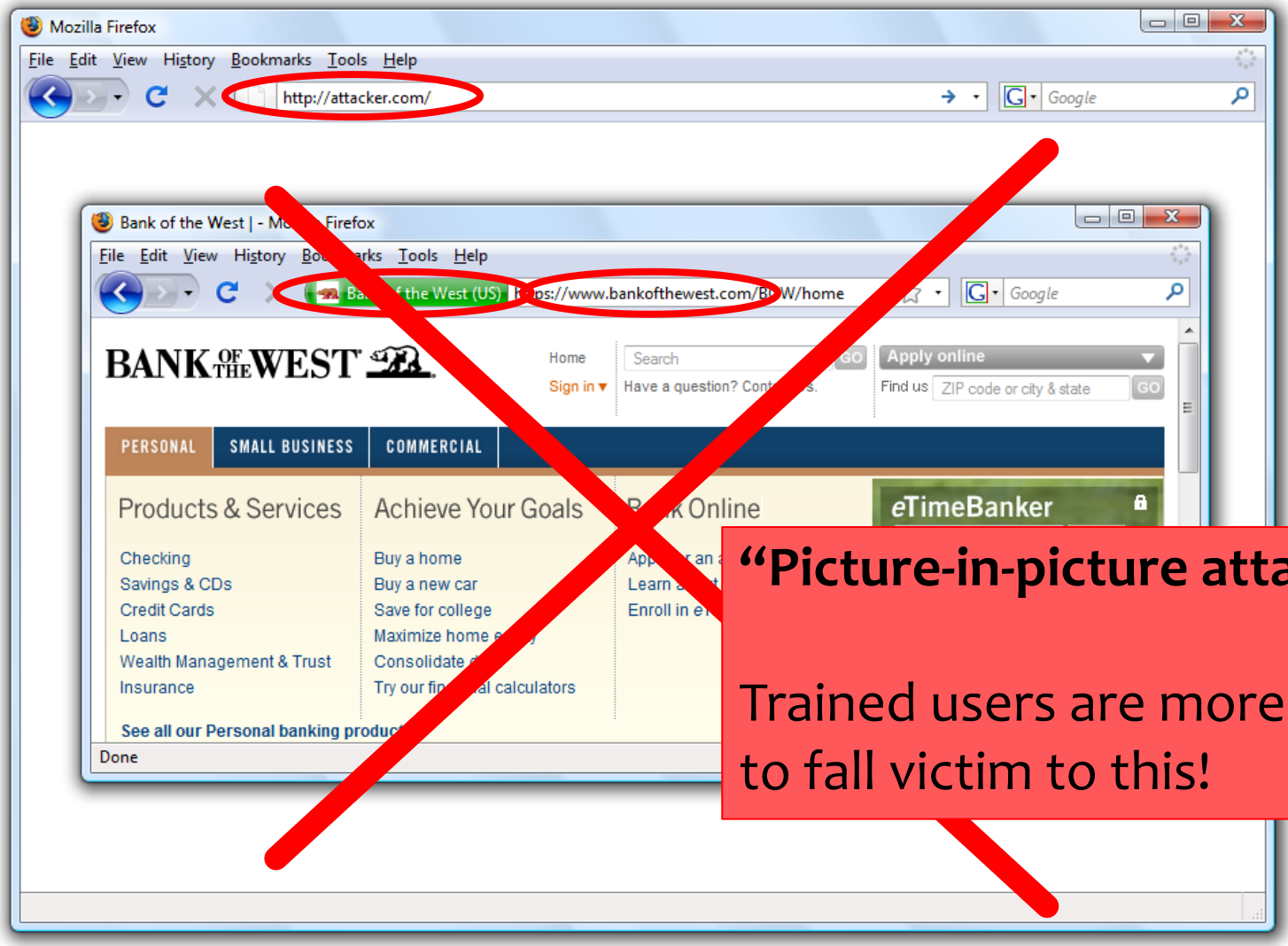


Safe to Type Your Password?



Safe to Type Your Password?

Safe to Type Your Password?



"Picture-in-picture attacks"

Trained users are more likely to fall victim to this!

Experiments at Indiana University

- Reconstructed the social network by crawling sites like Facebook, MySpace, LinkedIn and Friendster
- Sent 921 Indiana University students a spoofed email that appeared to come from their friend
- Email redirected to a spoofed site inviting the user to enter his/her secure university credentials
  - Domain name clearly distinct from indiana.edu
- 72% of students entered their real credentials into the spoofed site

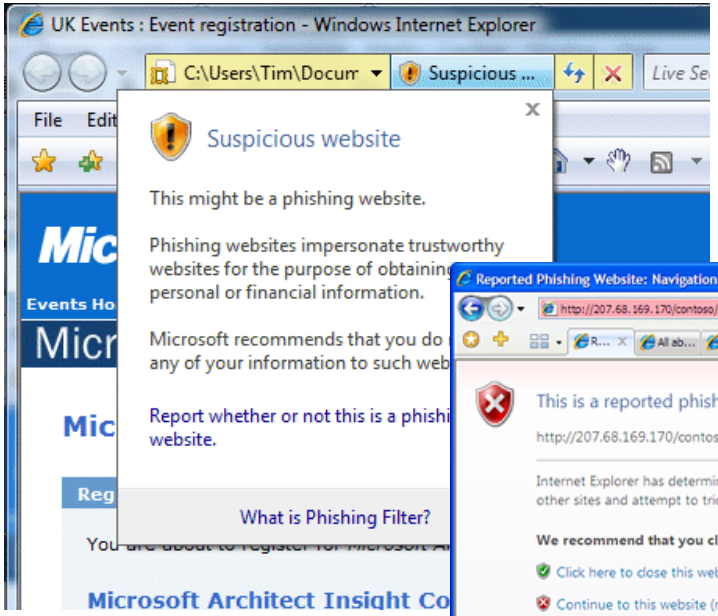
More Details

- Control group: 15 of 94 (16%) entered personal information
- Social group: 349 of 487 (72%) entered personal information

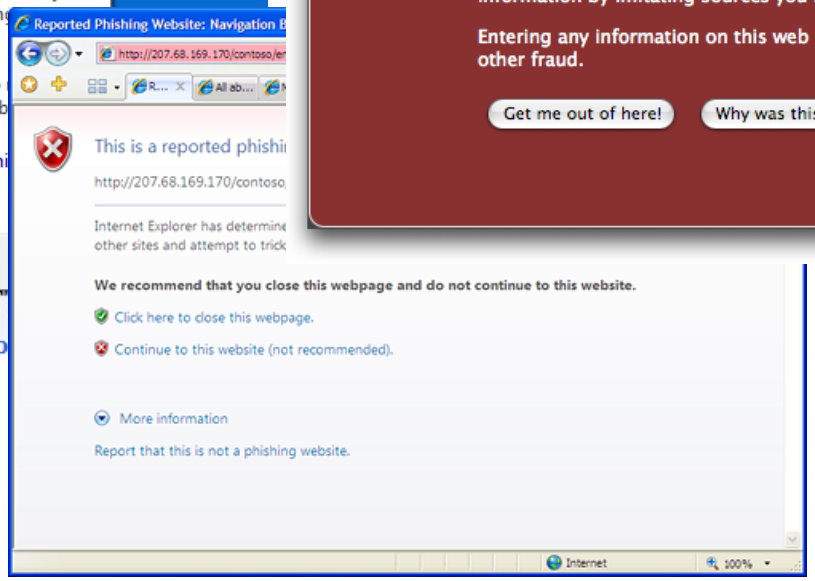
- 70% of responses within first 12 hours
- Adversary wins by gaining users' trust

- Also: If a site looks "professional", people likely to believe that it is legitimate

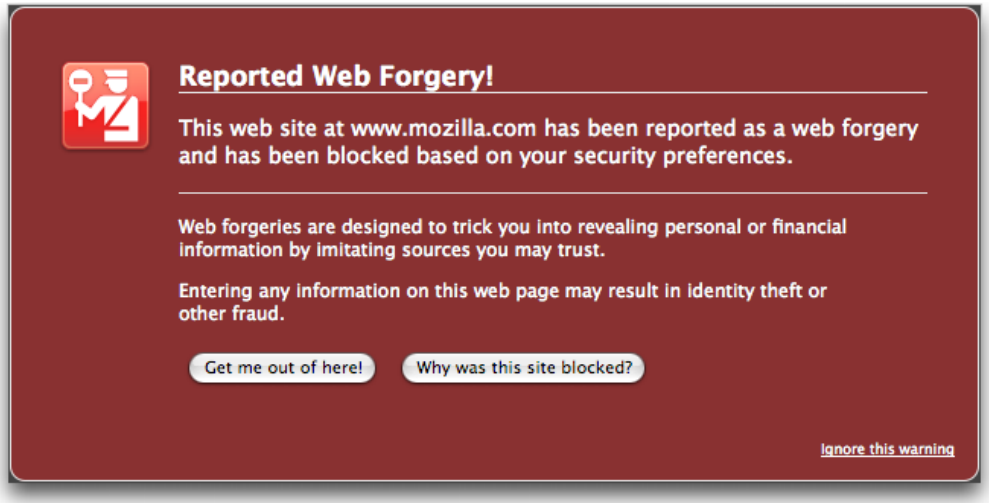
Phishing Warnings



Passive (IE)



Active (IE)



Active (Firefox)

[Egelman et al.]

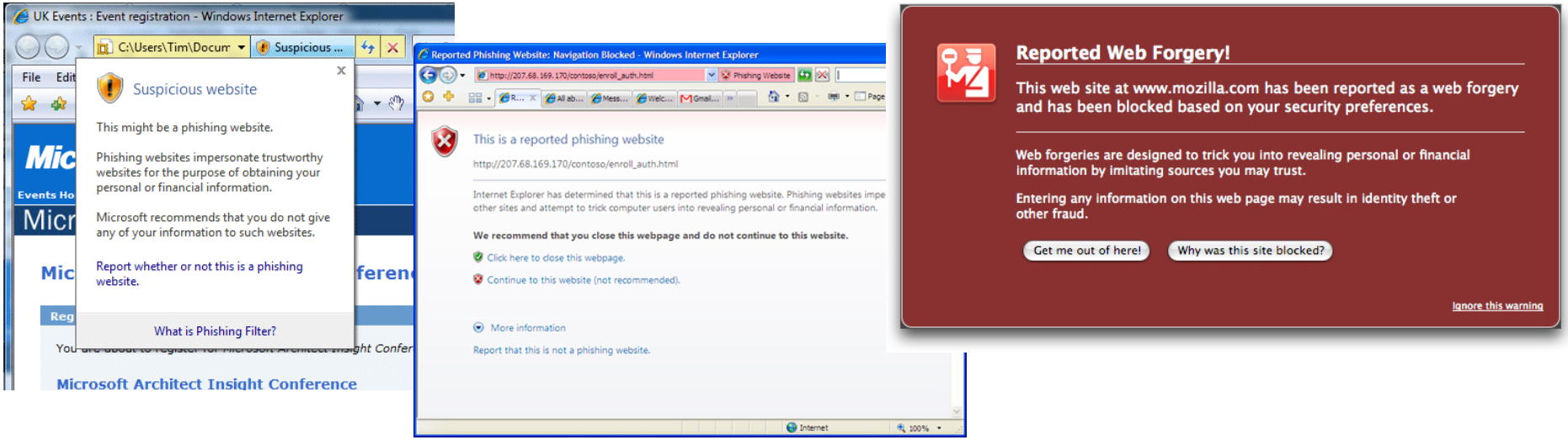
Are Phishing Warnings Effective?

- CMU study of 60 users
- Asked to make eBay and Amazon purchases
- All were sent phishing messages in addition to the real purchase confirmations
- Goal: compare active and passive warnings

[Egelman et al.]

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



Passive (IE) Active (IE) Active (Firefox)

[Egelman et al.]

User Response to Warnings

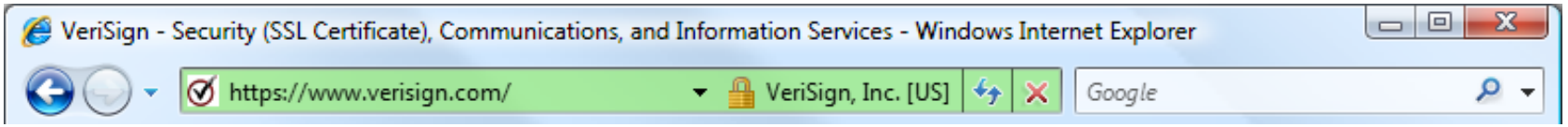
- Some fail to notice warnings entirely
  - Passive warning takes a couple of seconds to appear; if user starts typing, his keystrokes dismiss the warning
- Some saw the warning, closed the window, went back to email, clicked links again, were presented with the same warnings… repeated 4-5 times
  - Conclusion: "website is not working"
  - Users never bothered to read the warnings, but were still prevented from visiting the phishing site
  - Active warnings work!

[Egelman et al.]

Why Do Users Ignore Warnings?

- Don't trust the warning
  - "Since it gave me the option of still proceeding to the website, I figured it couldn't be that bad"
- Ignore warning because it's familiar (IE users)
  - "Oh, I always ignore those"
  - "Looked like warnings I see at work which I know to ignore"
  - "I thought that the warnings were some usual ones displayed by IE"
  - "My own PC constantly bombards me with similar messages"

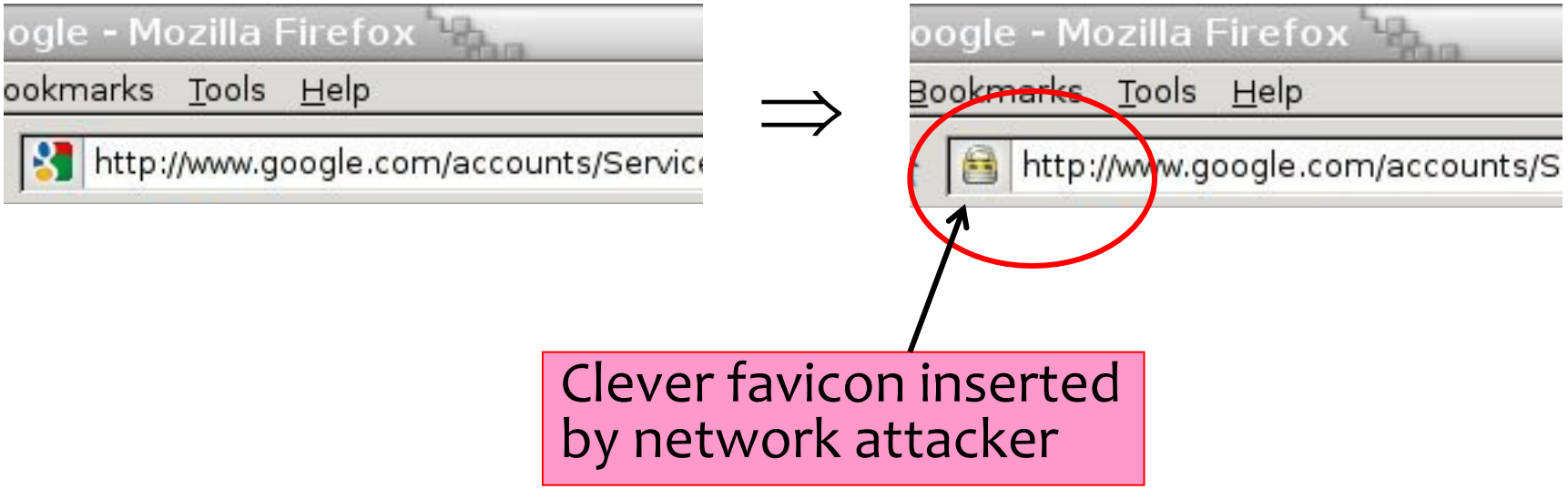
The Lock Icon



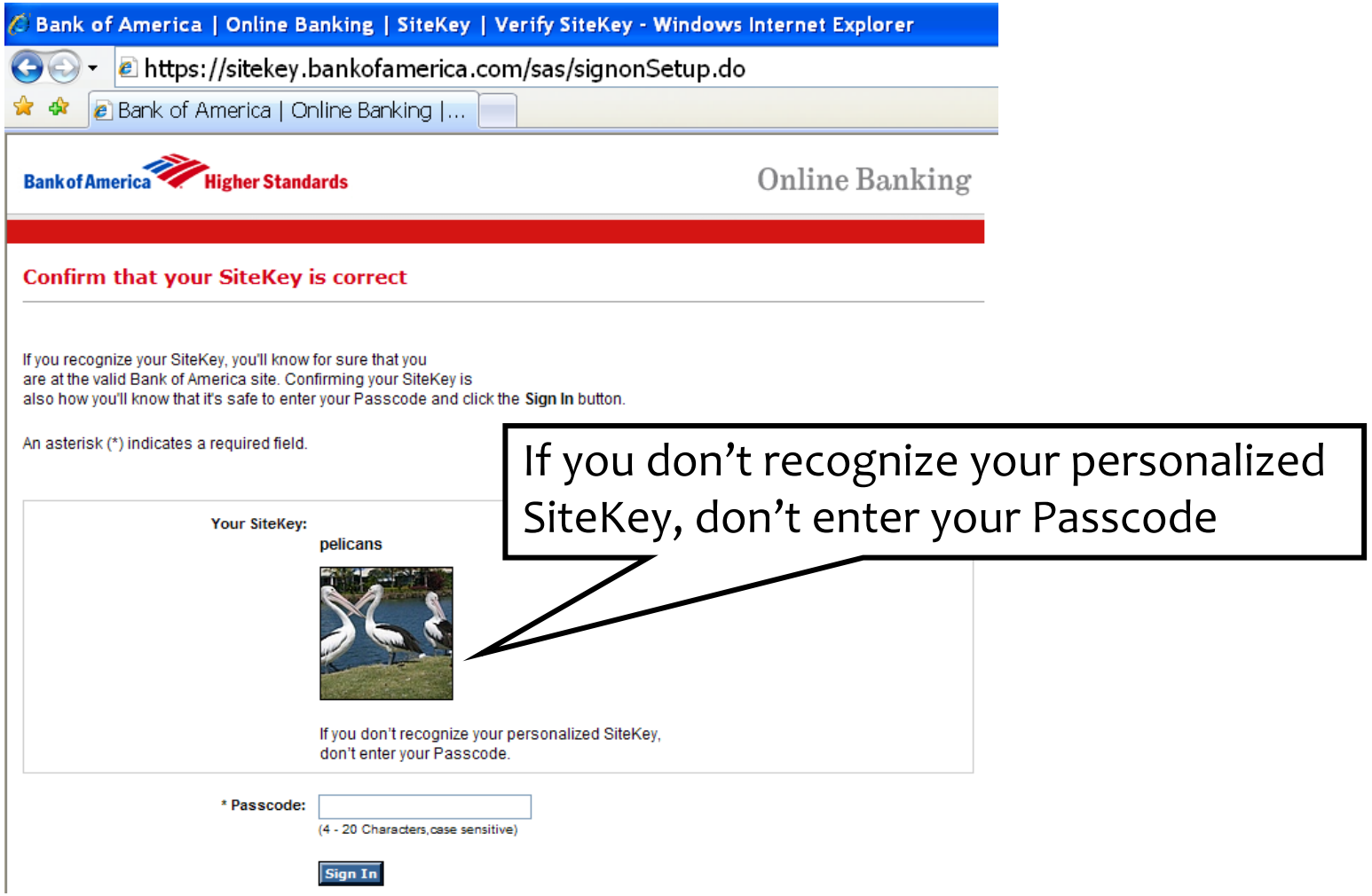
- Goal: identify secure connection
  - SSL/TLS is being used between client and server to protect against active network attacker
- Lock icon should only be shown when the page is secure against network attacker
  - Semantics subtle and not widely understood by users
  - Whose certificate is it??
  - Problem in user interface design

[Moxie Marlinspike]

Will You Notice?



Site Authentication Image (SiteKey)



Do These Indicators Help?

- "The Emperor's New Security Indicators"
  - http://www.usablesecurity.org/emperor/emperor.pdf

		Group				
Score	*First chose* not *to enter password...*	*1*	*2*	*3*	*1 ∪ 2*	*Total*
0	upon noticing HTTPS absent	0 0%	0 0%	0 0%	0 0%	0 0%
1	after site-authentication image removed	0 0%	0 0%	2 9%	0 0%	2 4%
2	after warning page	8 47%	5 29%	12 55%	13 37%	25 44%
3	never (always logged in)	10 53%	12 71%	8 36%	22 63%	30 53%
	Total	18	17	22	35	57

Users don't notice the **absence** of indicators!

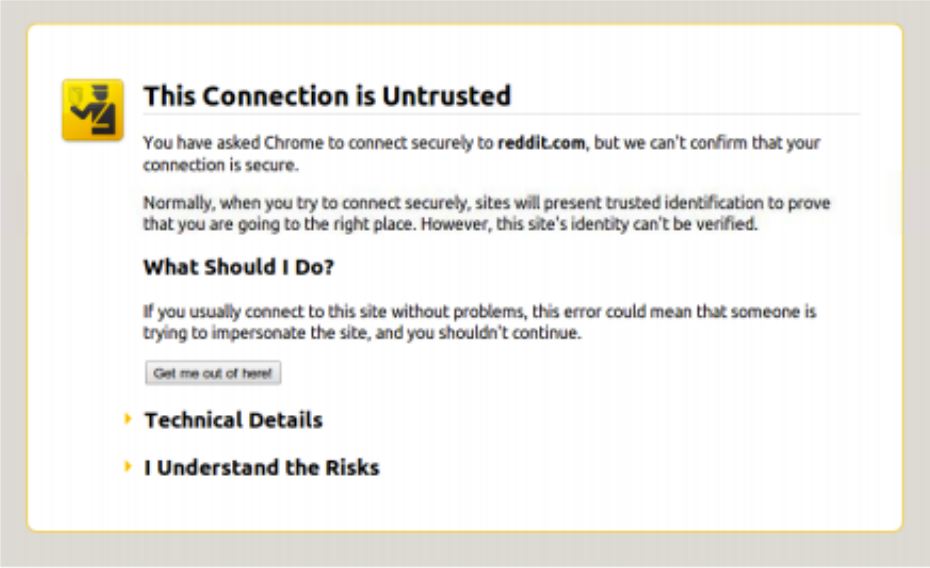
Case Study #2: Browser SSL Warnings

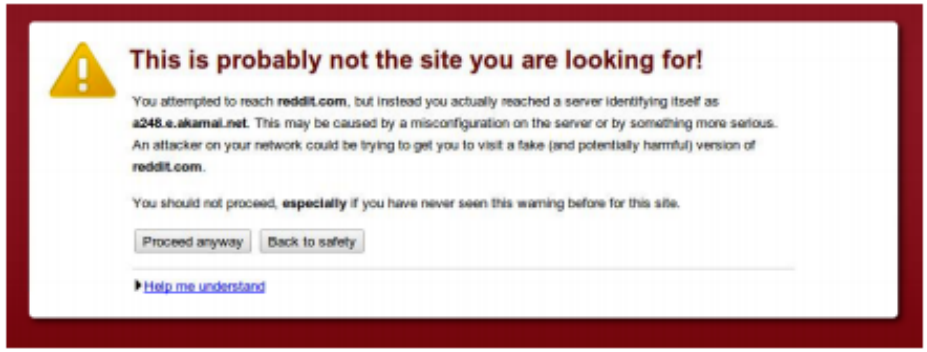
- Design question: How to alert the user if a site's SSL certificate is untrusted?

[Felt et al.]

Firefox vs. Chrome Warning

33% vs. 70% clickthrough rate





[Felt et al.]

Experimenting w/ Warning Design

#	Condition	CTR	N
1	Control (default Chrome warning)		
2	Chrome warning with policeman		
3	Chrome warning with criminal		
4	Chrome warning with traffic light		
5	Mock Firefox		
6	Mock Firefox, no image		
7	Mock Firefox with corporate styling		

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

[Felt et al.]

Experimenting w/ Warning Design

#	Condition	CTR	N
1	Control (default Chrome warning)	67.9%	17,479
2	Chrome warning with policeman		
3	Chrome warning with criminal		
4	Chrome warning with traffic light		
5	Mock Firefox		
6	Mock Firefox, no image		
7	Mock Firefox with corporate styling		

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

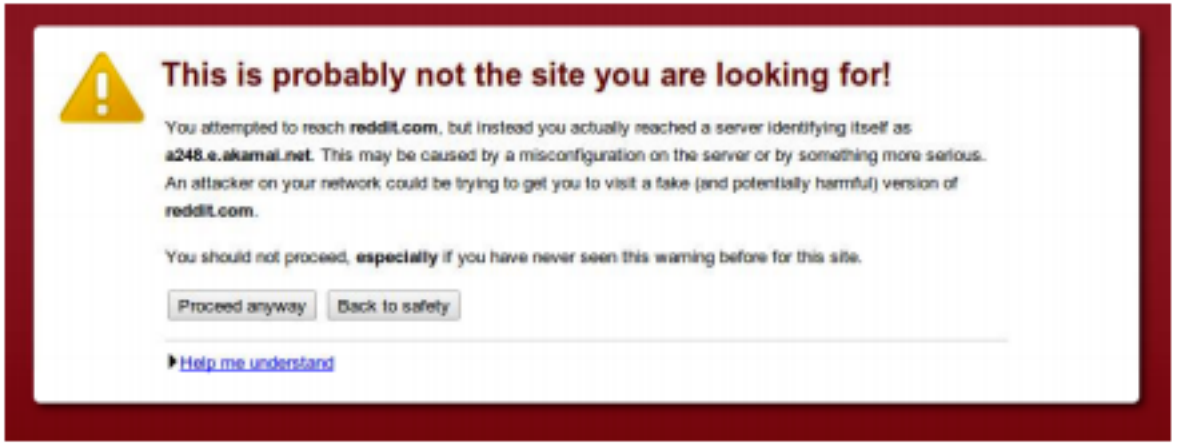


Figure 1. The default Chrome SSL warning (Condition 1).

[Felt et al.]

Experimenting w/ Warning Design

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

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

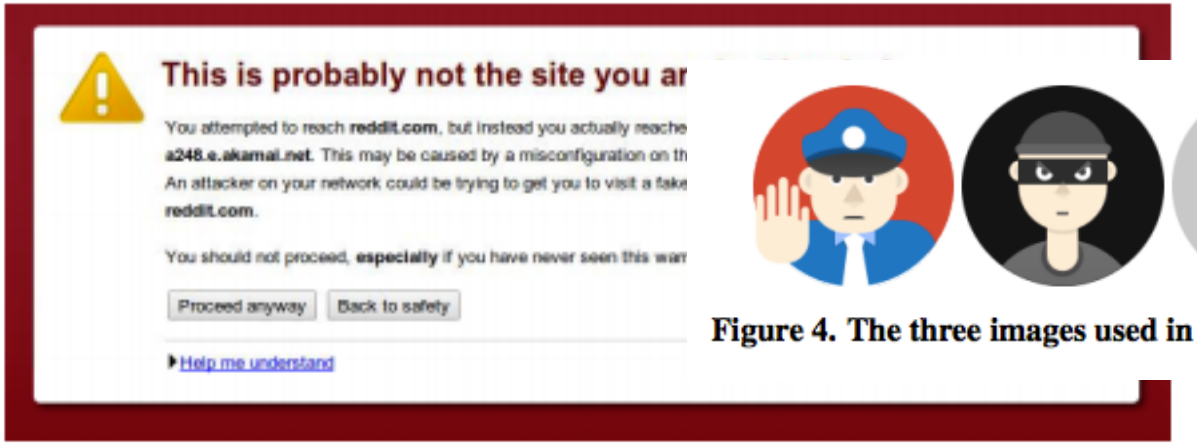


Figure 4. The three images used in Conditions 2-4.

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

[Felt et al.]

Experimenting w/ Warning Design

#	Condition	CTR	N
1	Control (default Chrome warning)	67.9%	17,479
2	Chrome warning with policeman	68.9%	17,977
3	Chrome warning with criminal	66.5%	18,049
4	Chrome warning with traffic light	68.8%	18,084
5	Mock Firefox	56.1%	20,023
6	Mock Firefox, no image	55.9%	19.297
7	Mock Firefox with corporate styling		

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

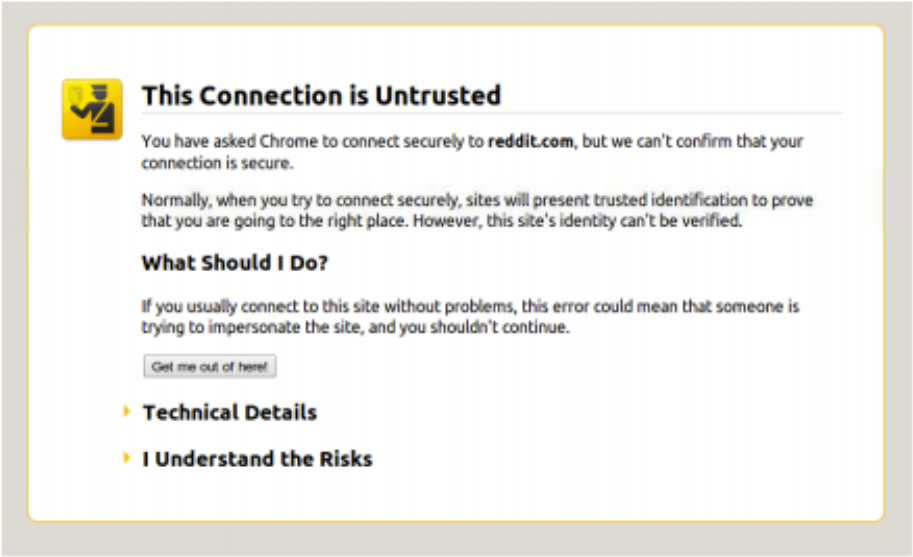


Figure 2. The mock Firefox SSL warning (Condition 5).

[Felt et al.]

Experimenting w/ Warning Design

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

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

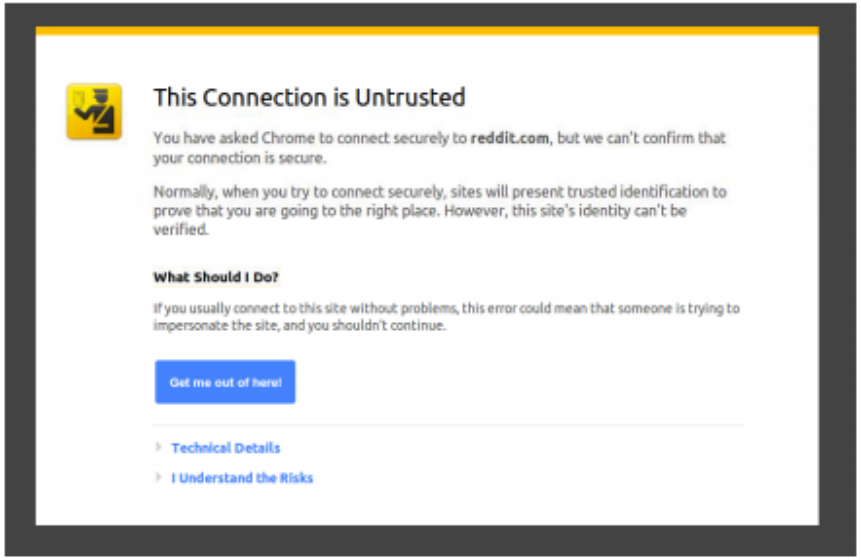
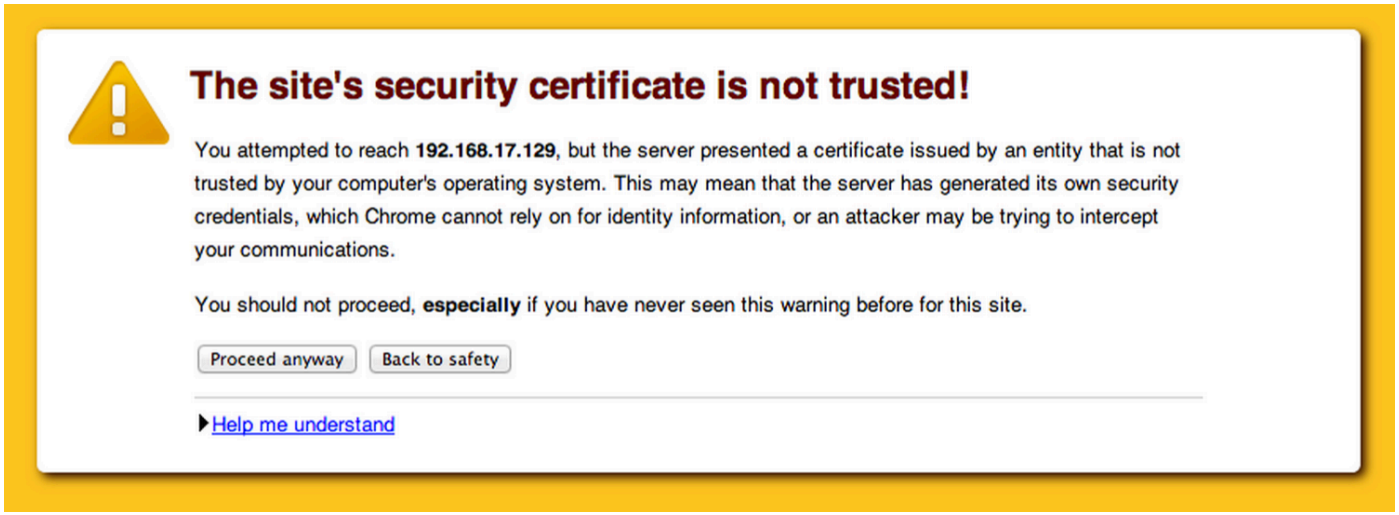


Figure 3. The Firefox SSL warning with Google styling (Condition 7).

[Felt et al.]

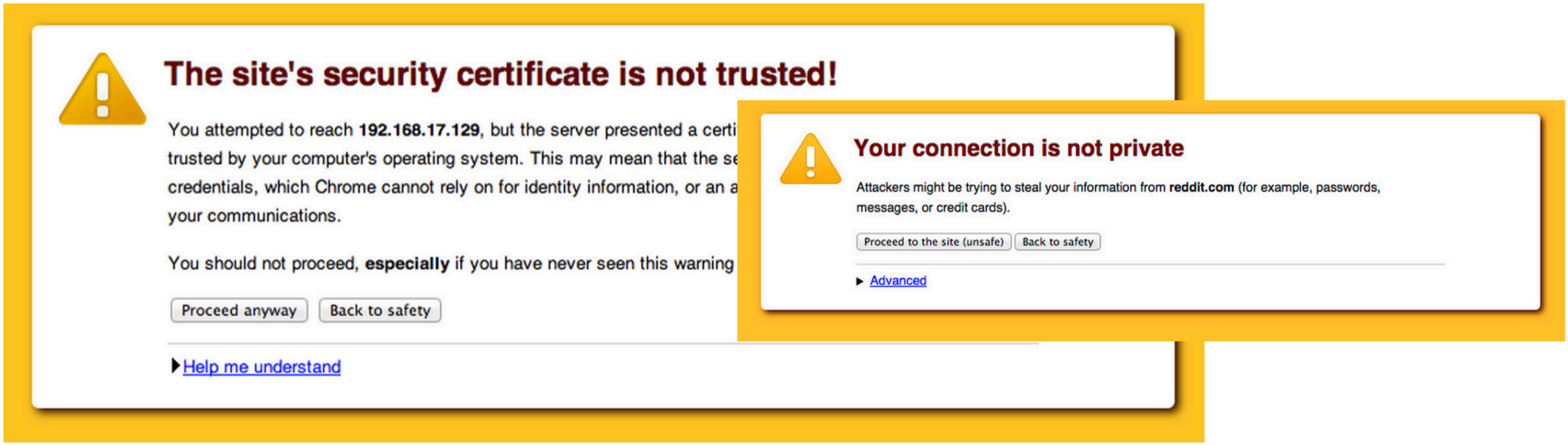
Opinionated Design Helps!

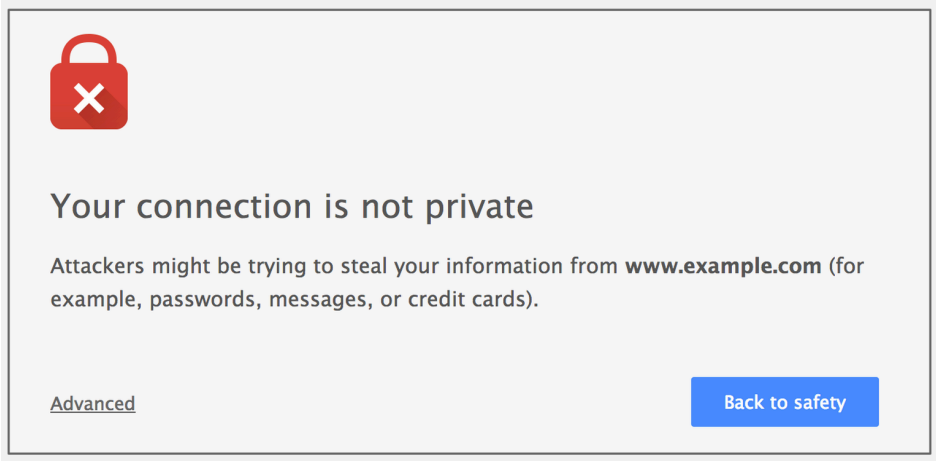


Adherence	N
30.9%	4,551

[Felt et al.]

Opinionated Design Helps!

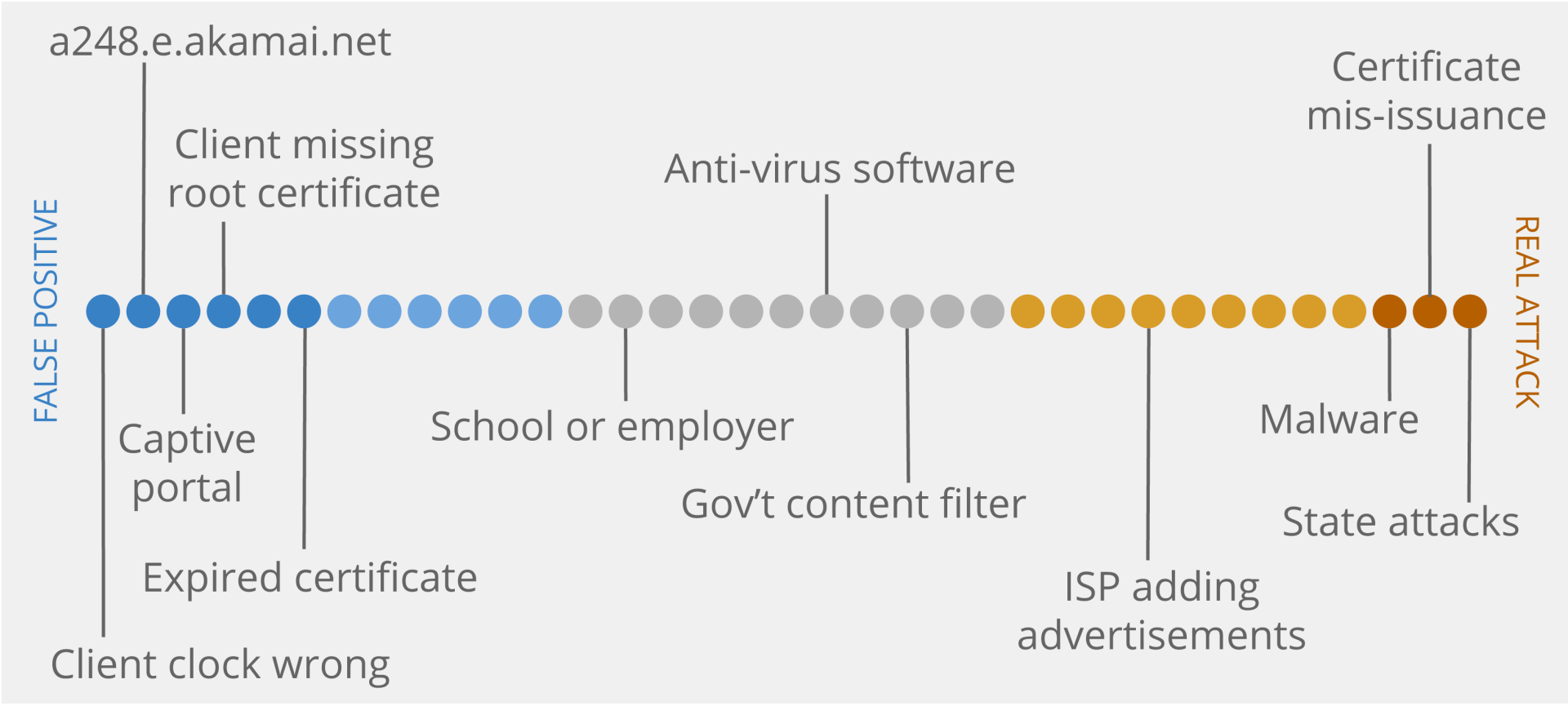




Adherence	N
30.9%	4,551
32.1%	4,075
58.3%	4,644

[Felt et al.]

Challenge: Meaningful Warnings



Stepping Back: Root Causes?

- Computer systems are complex; users lack intuition
- Users in charge of managing own devices
  - Unlike other complex systems, like healthcare or cars.
- Hard to gauge risks
  - "It won't happen to me!"
- Annoying, awkward, difficult
- Social issues
  - Send encrypted emails about lunch?...

How to Improve?

- Security education and training
- Help users build accurate mental models
- Make security invisible
- Make security the least-resistance path
- …?