CSE 484 / CSE M 584: Computer Security and Privacy

Mobile Platform Security (finish)

Fall 2016

Ada (Adam) Lerner
lerner@cs.washington.edu

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

Security Mindset: Customs

- Exchange on Reddit comment thread
- Started with an observation about the world:
  - "I tried to ship something to Venezuela, but it would have cost $80 shipping and $1420 in taxes and duty import fees!"

Security Mindset: Customs

- Problem: Extremely high customs fees.

- Solution?

[–] AnonymousGuy767 552 points 1 day ago

This is why you lie and say the box is a gift or $2 value. Chinese sellers do it all day every day on ebay.

I would also suggest indicating in the description that the items are broken or malfunctioning and you're returning for refund. Makes them less likely to be stolen by customs.

permalink source embed save save-RES parent report give gold reply

Lie about the value of the item, or, better, claim it's broken!

[–] snotrokit 57 points 1 day ago

That won't make it past the customs inspection. They snatch it up in a heartbeat then throw the recipient in jail for fraud

permalink source save save-RES parent report give gold reply

"That won't make it past the customs inspection. They snatch it up in a heartbeat then throw the recipient in jail for fraud."

“That can’t be right. Otherwise I could just send packages of people I don’t like in other countries with fake packing slips to have them arrested.”

[–] AnonymousGuy767 157 points 1 day ago

That can't be right. Otherwise I could just send packages to people I don't like in other countries with fake packing slips to have them arrested.

permalink source save save-RES parent report give gold reply

Mobile Malware Attack Vectors

- Unique to phones:
  - Premium SMS messages
  - Identify location
  - Record phone calls
  - Log SMS
- Similar to desktop/PCs:
  - Connects to botmasters
  - Steal data
  - Phishing
  - Malvertising

Mobile Malware Examples

"ikee is never going to give you up"

[Zhou et al.]

(Android) Malware in the Wild

What does it do?

	Root Exploit	Remote Control		Financial Charges			Information Stealing		
		Net	SMS	Phone Call	SMS	Block SMS	SMS	Phone #	User Account
# Families	20	27	1	4	28	17	13	15	3
# Samples	1204	1171	1	256	571	315	138	563	43

What's Different about Mobile Platforms?

- Applications are isolated
  - Each runs in a separate execution context
  - No default access to file system, devices, etc.
  - **Different than traditional OSes** where multiple applications run with the same user permissions!

- **App Store:** approval process for applications
  - Market: Vendor controlled/Open
  - App signing: Vendor-issued/self-signed
  - User approval of permissions

Two Types of App We Want to Defend Against

- Malware
- Legit, but privacy invasive

(1) Permission Granting Problem

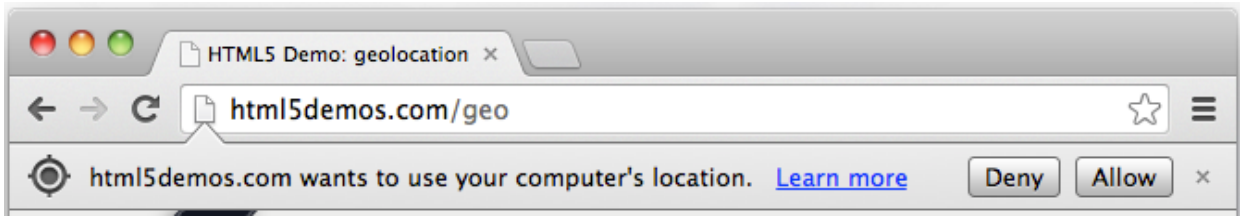
Smartphones (and other modern OSes) try to prevent such attacks by limiting applications' access to:

- System Resources (clipboard, file system).
- Devices (camera, GPS, phone, …).

How should operating system grant permissions to applications?

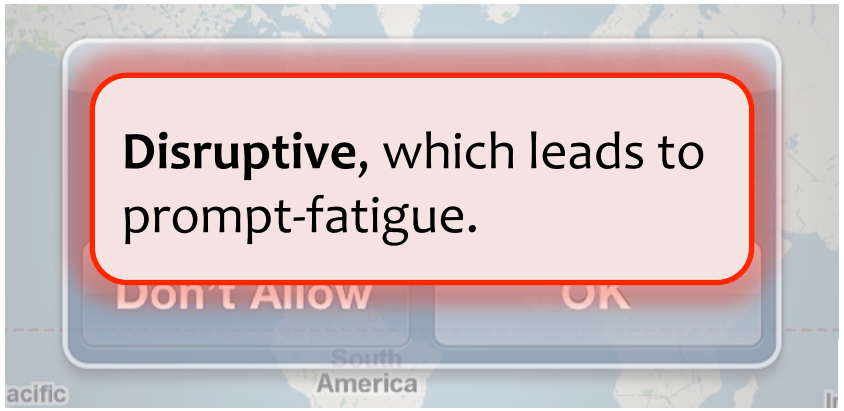
State of the Art

Prompts (time-of-use)

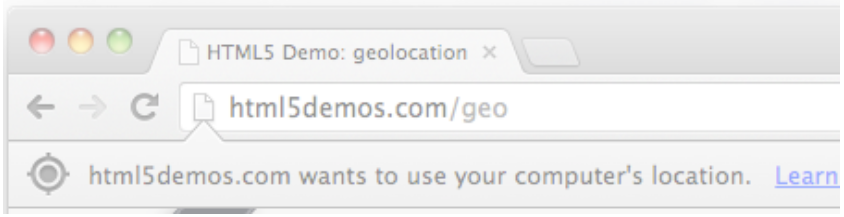


State of the Art

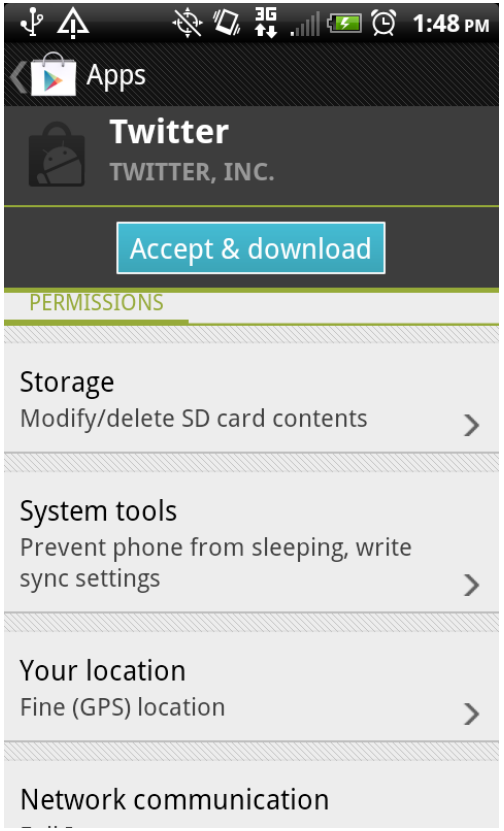
Prompts (time-of-use)



Disruptive, which leads to prompt-fatigue.



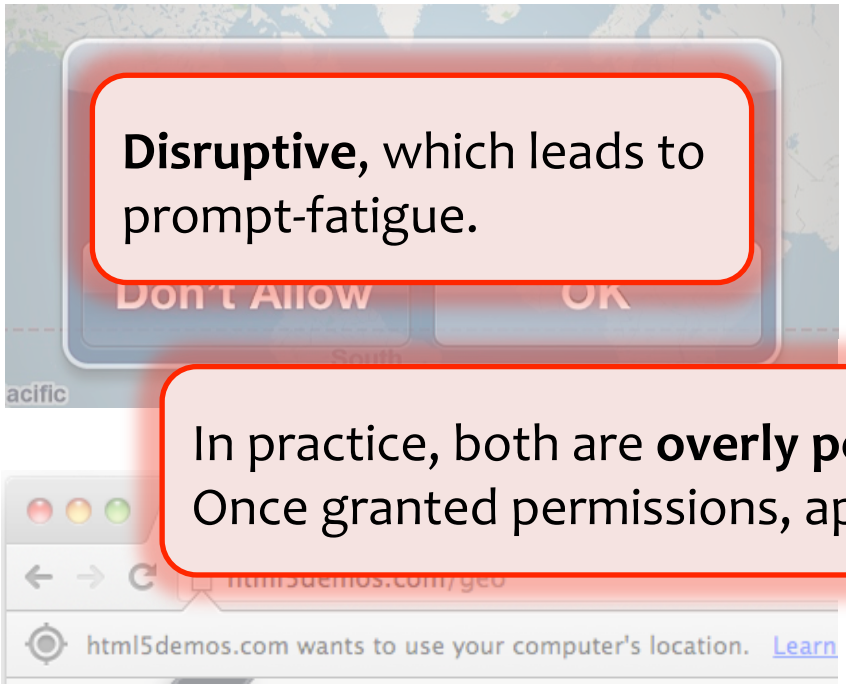
Manifests (install-time)

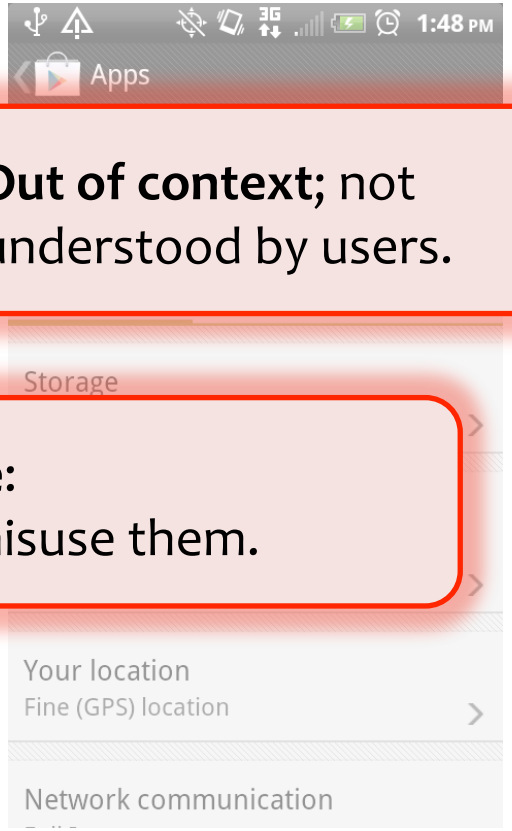


State of the Art

Prompts (time-of-use)

Manifests (install-time)





Disruptive, which leads to prompt-fatigue.

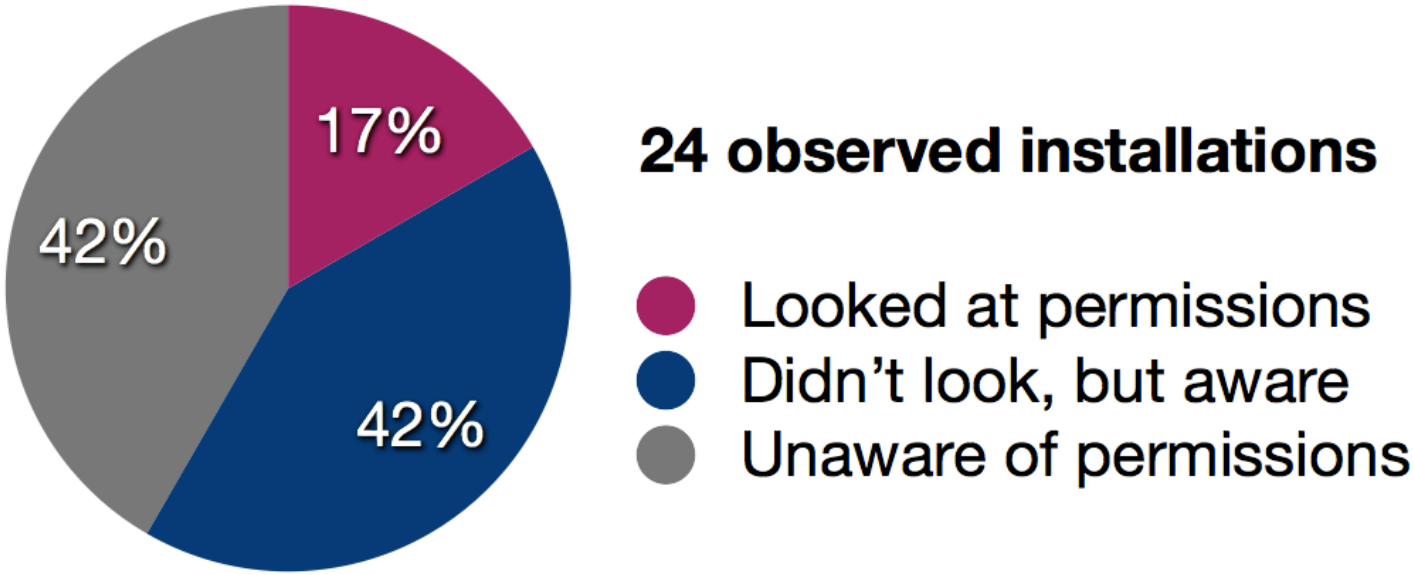
Out of context; not understood by users.

In practice, both are **overly permissive**:
Once granted permissions, apps can misuse them.

[Felt et al.]

Are Manifests Usable?

Do users pay attention to permissions?



… but 88% of users looked at reviews.

[Felt et al.]

Are Manifests Usable?

Do users understand the warnings?

	Permission	n	Correct Answers	
1 Choice	READ_CALENDAR	101	46	45.5%
	CHANGE_NETWORK_STATE	66	26	39.4%
	READ_SMS$_1$	77	24	31.2%
	CALL_PHONE	83	16	19.3%
2 Choices	WAKE_LOCK	81	27	33.3%
	WRITE_EXTERNAL_STORAGE	92	14	15.2%
	READ_CONTACTS	86	11	12.8%
	INTERNET	109	12	11.0%
	READ_PHONE_STATE	85	4	4.7%
	READ_SMS$_2$	54	12	22.2%
4	CAMERA	72	7	9.7%

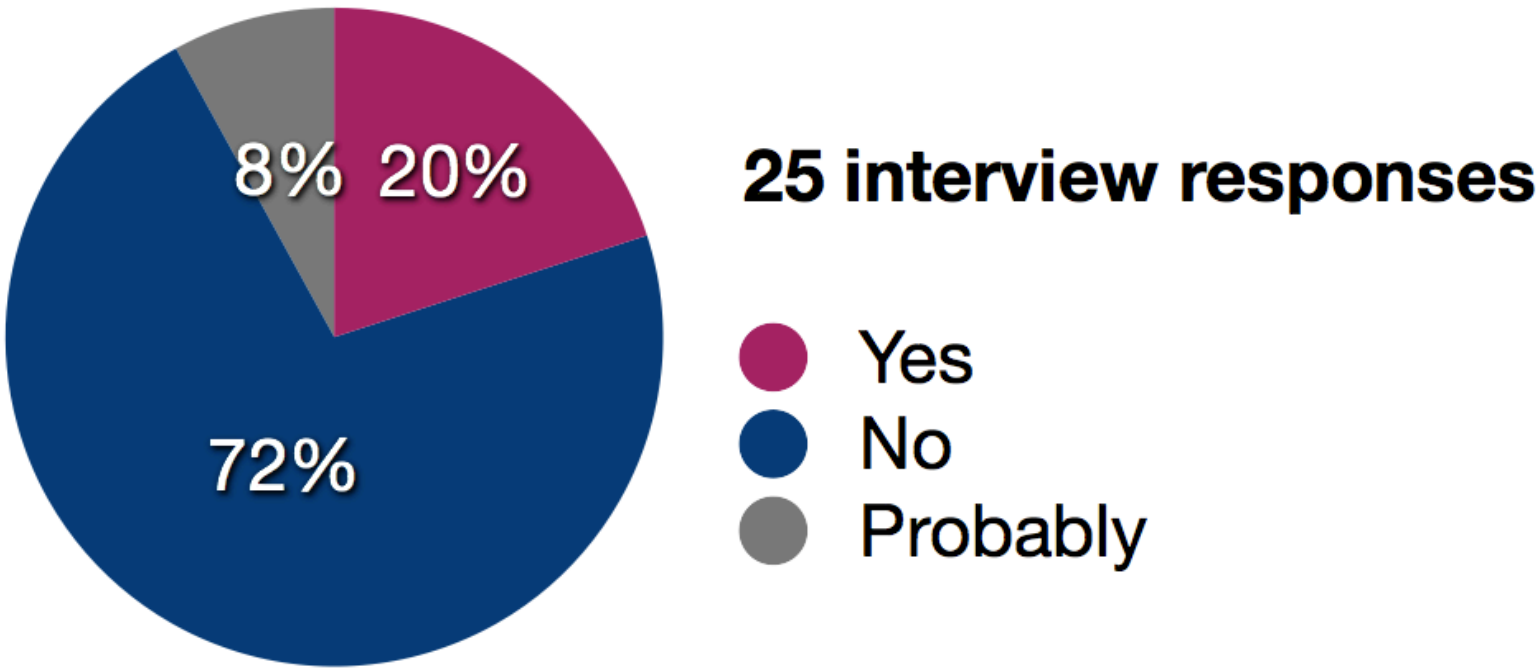
Table 4: The number of people who correctly answered a question. Questions are grouped by the number of correct choices. n is the number of respondents. (Internet Survey, $n = 302$)

[Felt et al.]

Are Manifests Usable?

Do users act on permission information?

"Have you ever not installed an app because of permissions?"

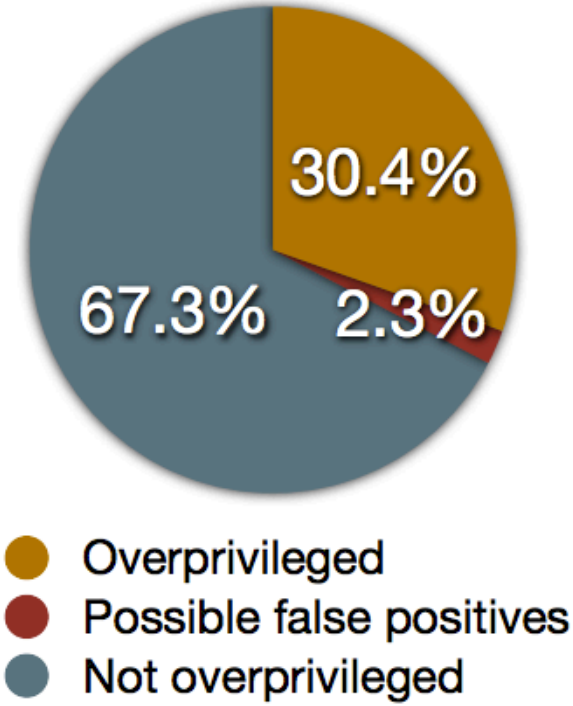


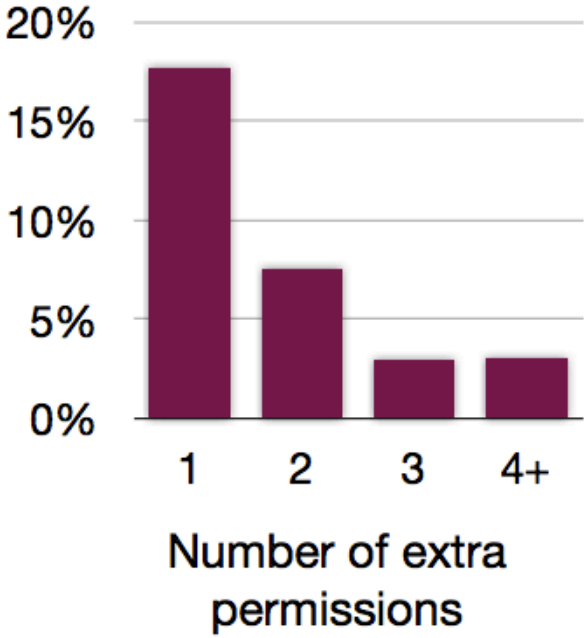
[Felt et al.]

Over-Permissioning

- Android permissions are badly documented.
- Researchers have mapped APIs → permissions.

www.android-permissions.org (Felt et al.), http://pscout.csl.toronto.edu (Au et al.)





Why is Over-Permissioning Bad?

- **Over-permissioning:** app has permission to access resources but never accesses them.

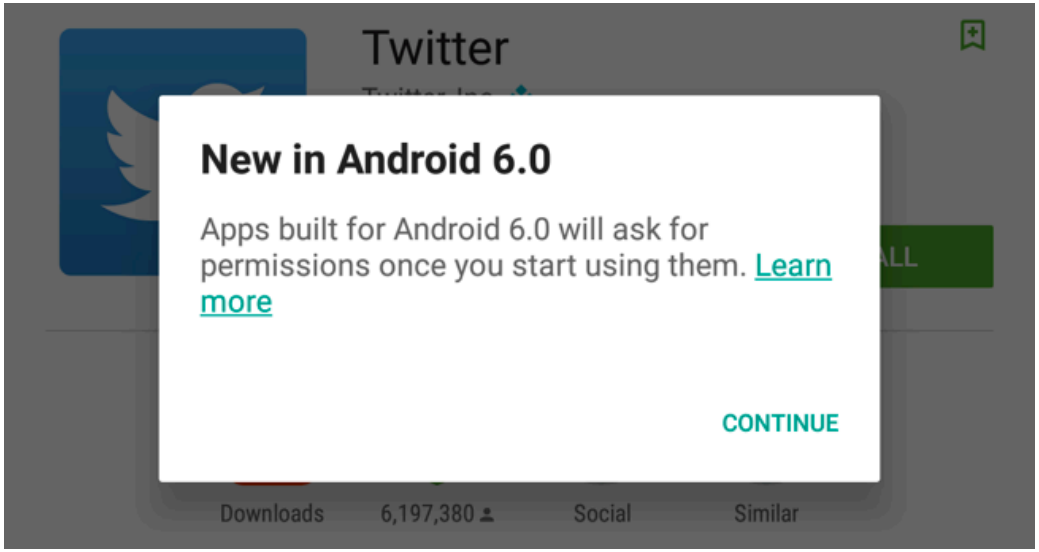
- If the app never uses the extra permissions, why is it bad that it has them?

Manifests rely on the user to make good choices at install time

- It's not clear that users know how to make the right choice – or that there IS a right choice.

- I don't want ANY app to access my camera at all times. I just want apps to access my camera when they need to for legitimate purposes!

Android 6.0: Prompts!



- First-use prompts for sensitive permission (like iOS).
- Big change! Now app developers need to check for permissions or catch exceptions.

Promps rely on the user to make good choices at use time

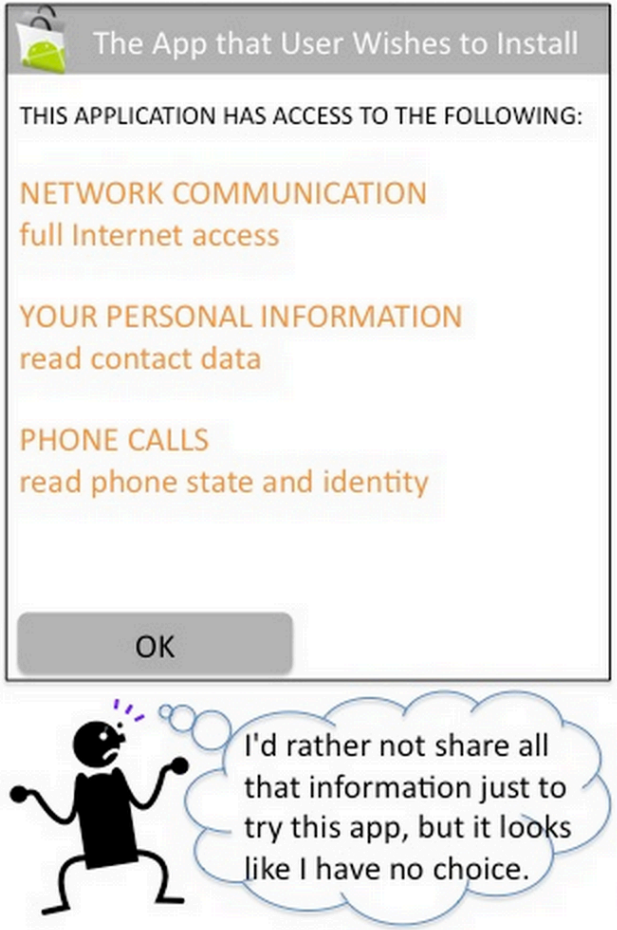
- It's not clear that users know how to make the right choice at use time either.

- Still only checks on first use – the app can still use the resource for any reason it wants, at any time now or in the future.

[Hornyack et al.]

Improving Permissions: AppFence

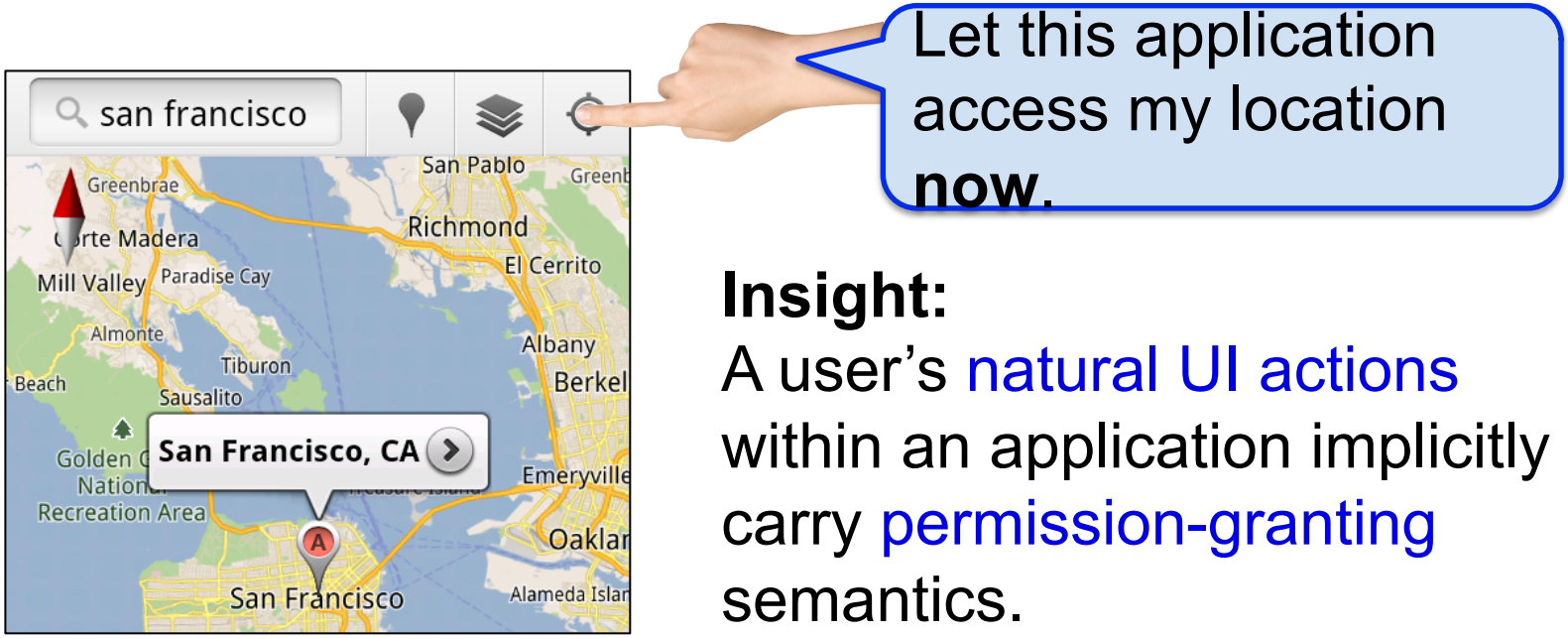
Today, ultimatums give app developers an unfair edge in obtaining permissions.



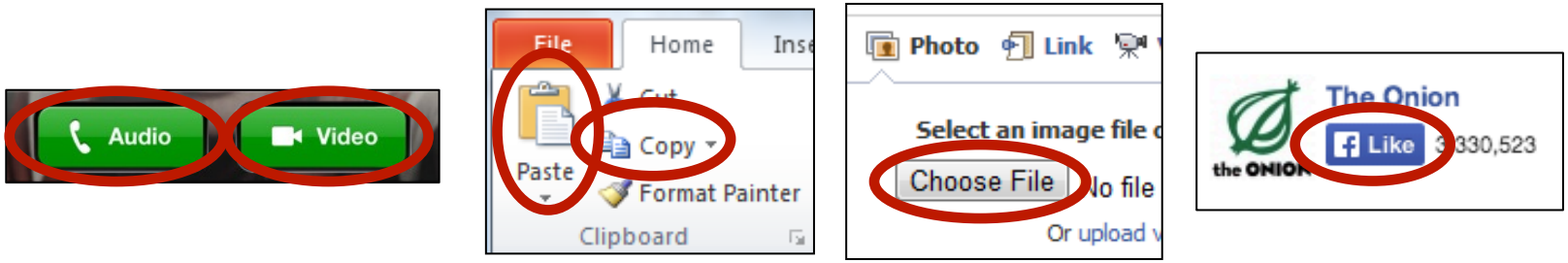
AppFence can enable new interfaces that give users control over the use of their info.

[Roesner et al.]

Improving Permissions: User-Driven Access Control

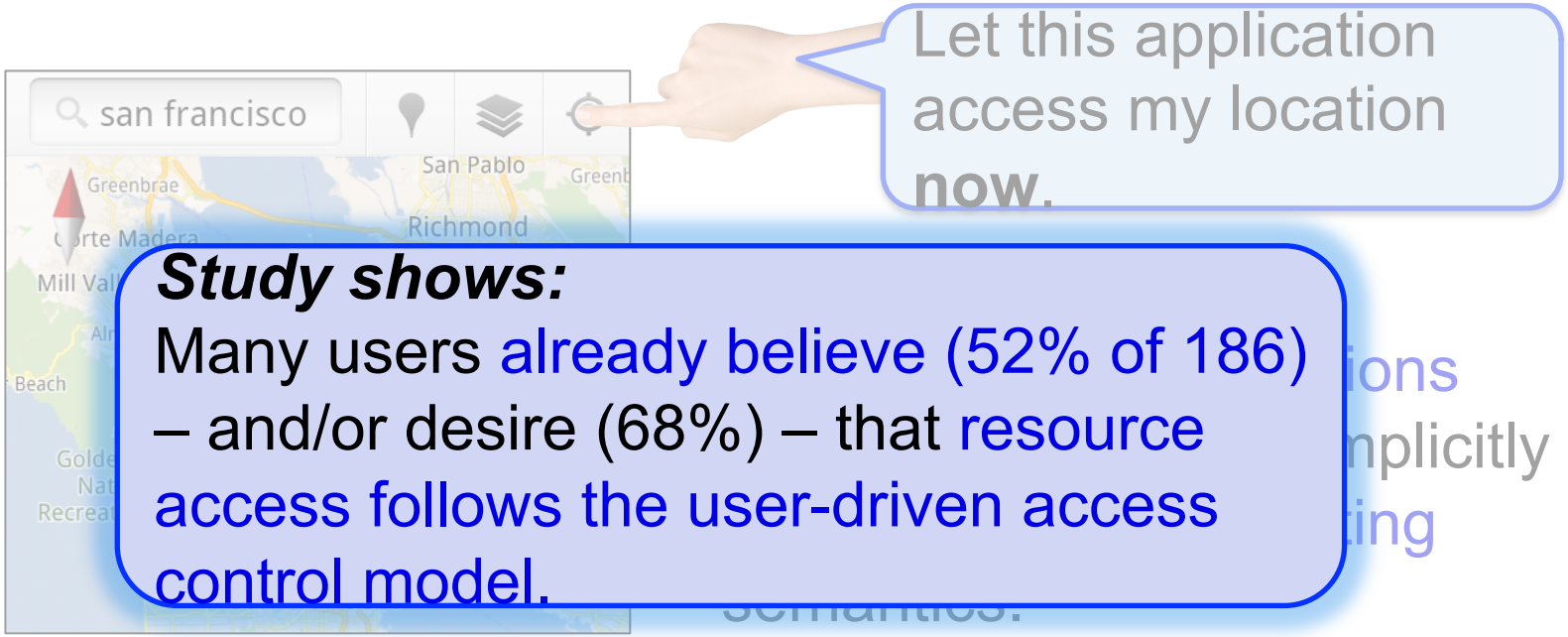


Insight:
A user's natural UI actions within an application implicitly carry permission-granting semantics.

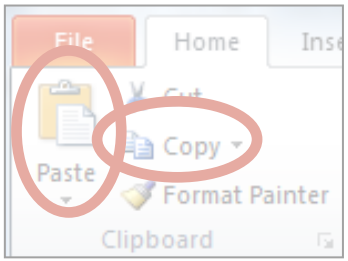


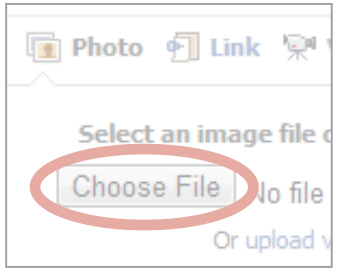
[Roesner et al.]

Improving Permissions: User-Driven Access Control

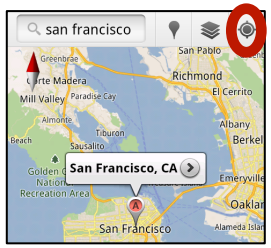


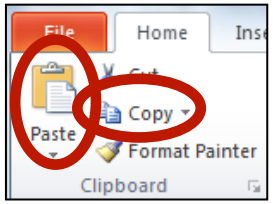
Study shows:
Many users already believe (52% of 186) – and/or desire (68%) – that resource access follows the user-driven access control model.

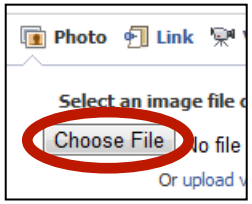




New OS Primitive: Access Control Gadgets (ACGs)







Approach: Make resource-related UI elements first-class operating system objects (access control gadgets).

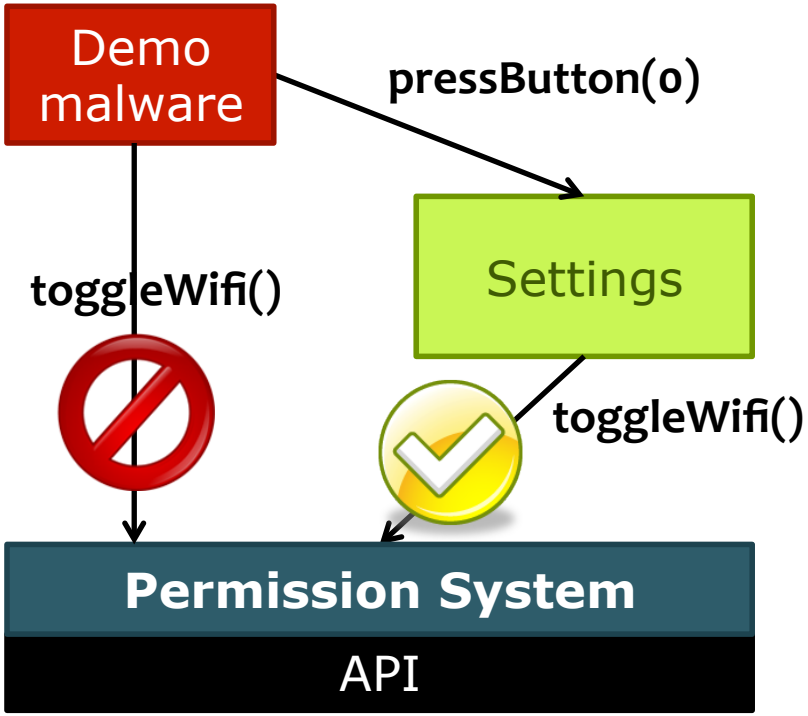
- To receive resource access, applications must embed a system-provided ACG.
- ACGs allow the OS to capture the user's permission granting intent in application-agnostic way.

Misc Thoughts From Mobile Security

[Felt et al.]

Permission Re-Delegation

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



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

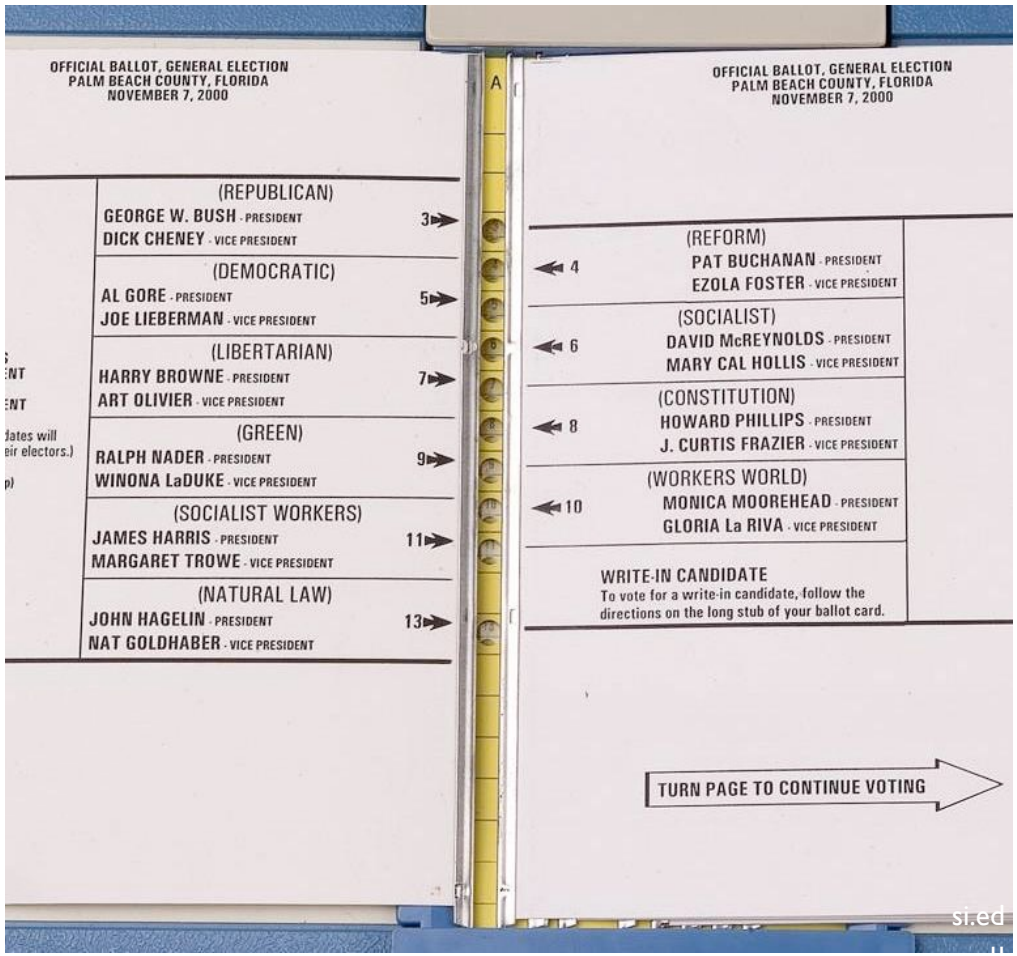
[https://developer.android.com/about/dashboards/index.html]

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.0%
4.1.x	Jelly Bean	16	7.2%
4.2.x		17	10.0%
4.3		18	2.9%
4.4	KitKat	19	32.5%
5.0	Lollipop	21	16.2%
5.1		22	19.4%
6.0	Marshmallow	23	7.5%

Data collected during a 7-day period ending on May 2, 2016.
Any versions with less than 0.1% distribution are not shown.

USABLE SECURITY

Poor Usability Causes Problems



AP

Importance in Security

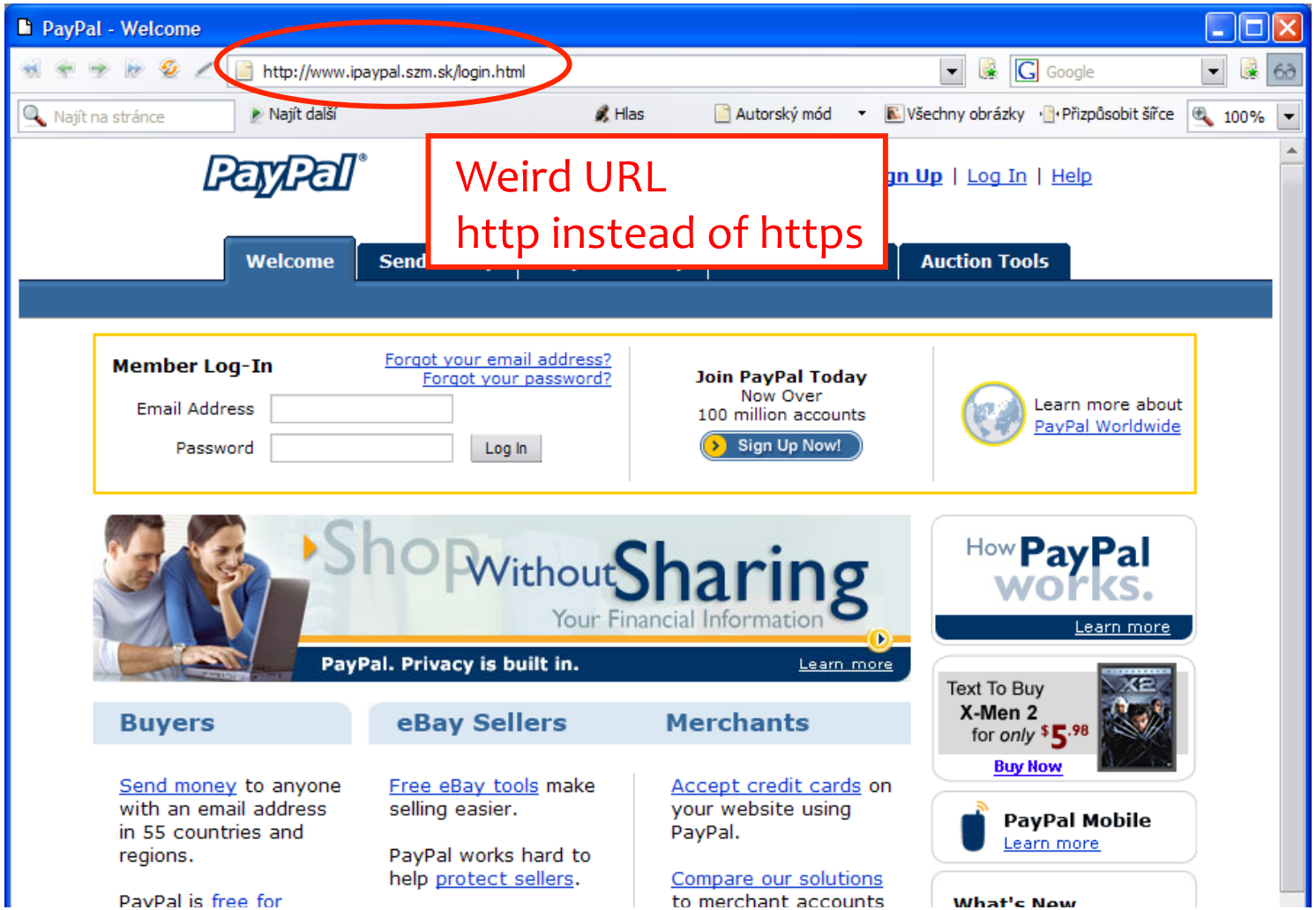
- Why is usability important?
  - People are the critical element of any computer system
    - People are the real reason computers exist in the first place
  - Even if it is **possible** for a system to protect against an adversary, people may use the system in other, **less secure** ways

Today

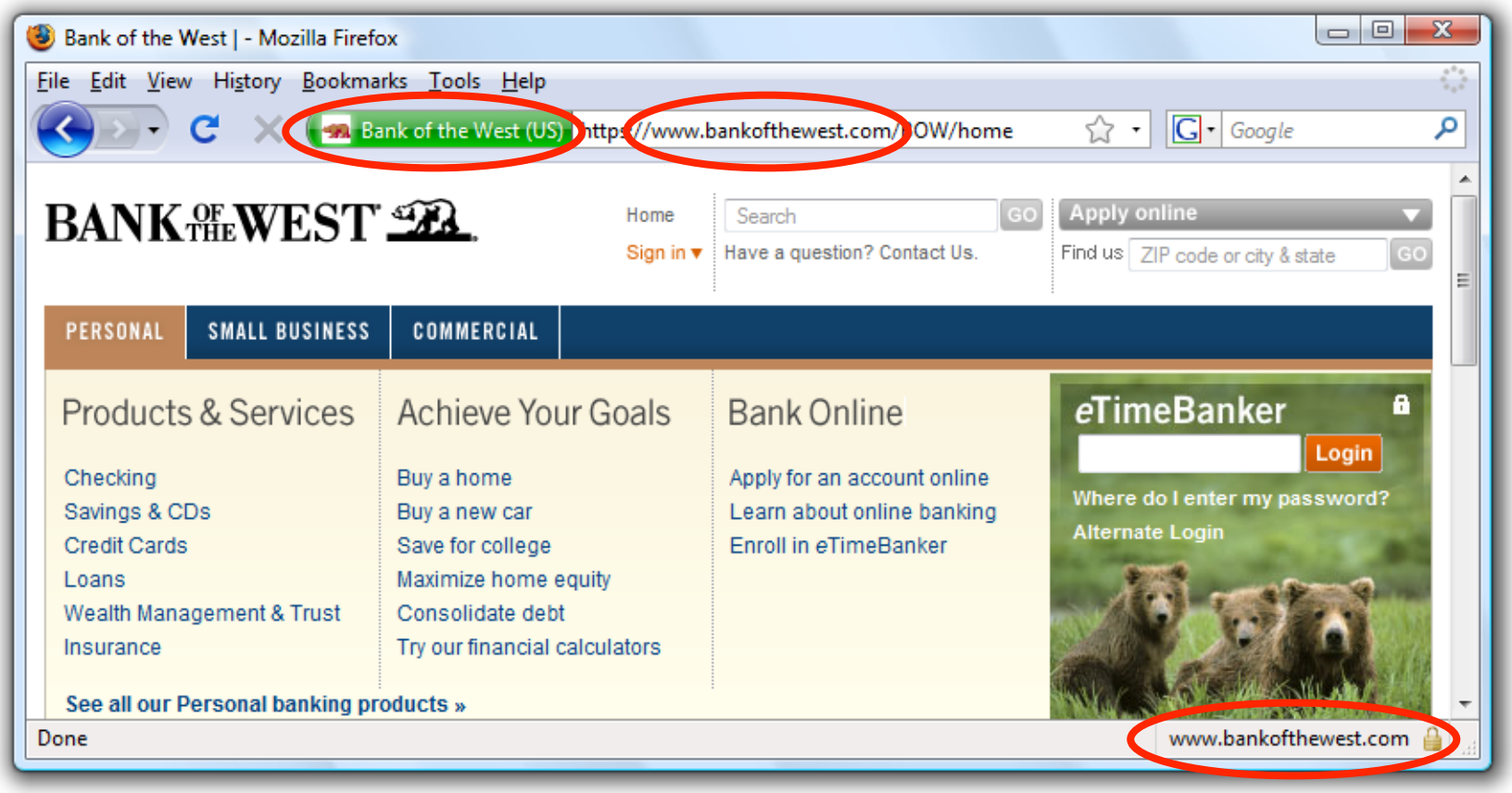
- 3 case studies
  - Phishing
  - SSL warnings
  - Password managers
- Step back: root causes of usability problems, and how to address

Case Study #1: Phishing

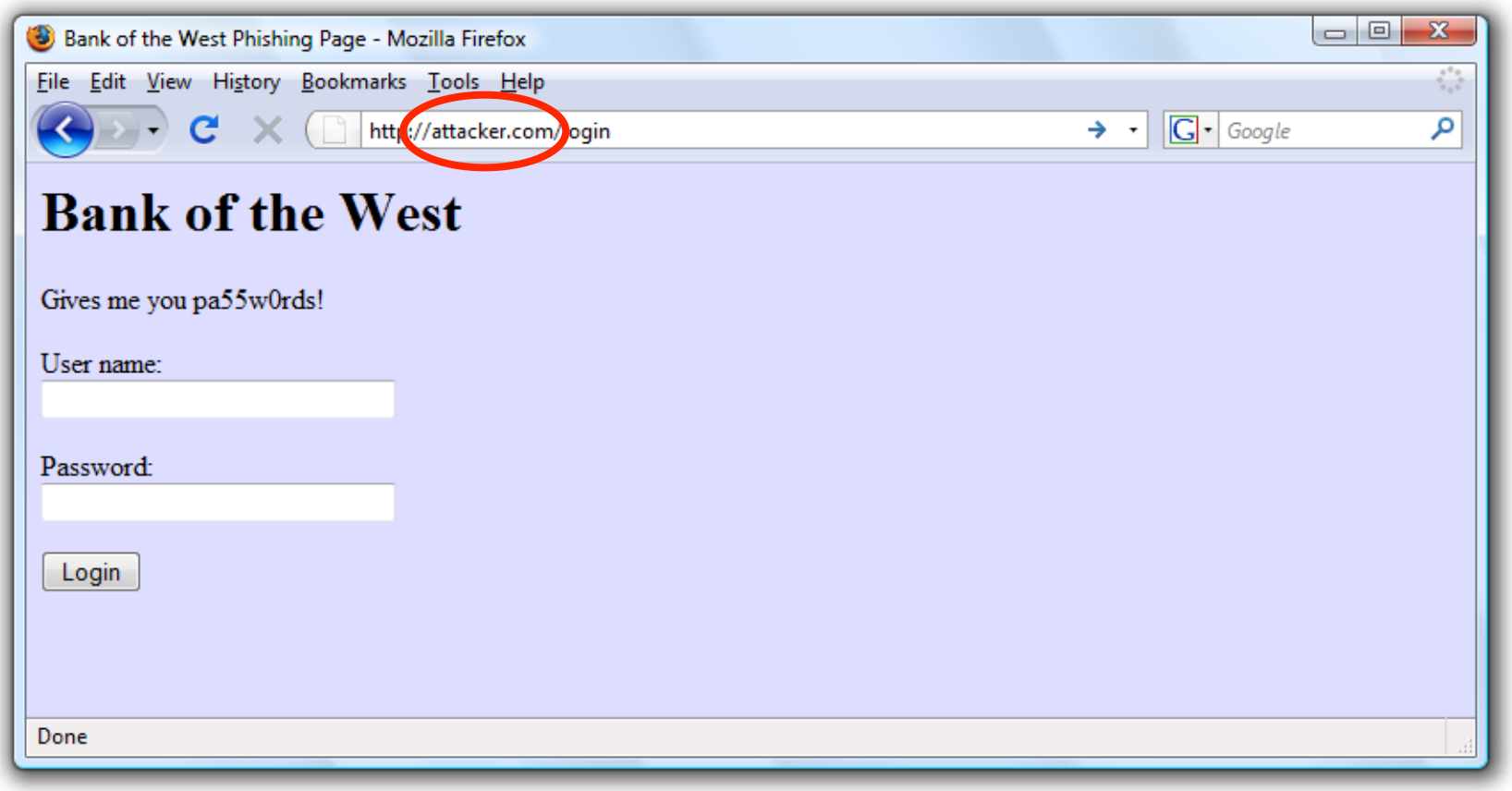
A Typical Phishing Page



Safe to Type Your Password?

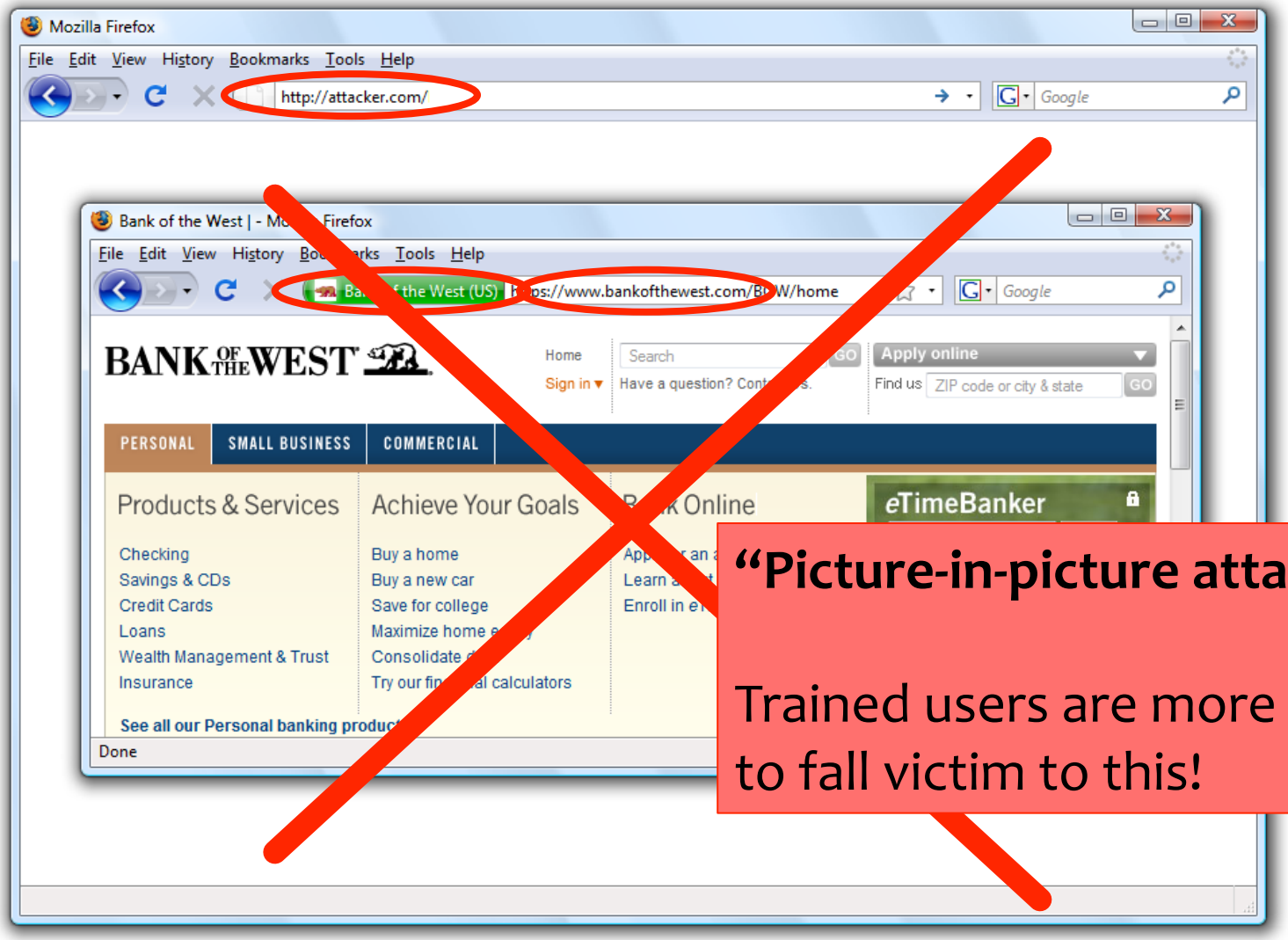


Safe to Type Your Password?



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

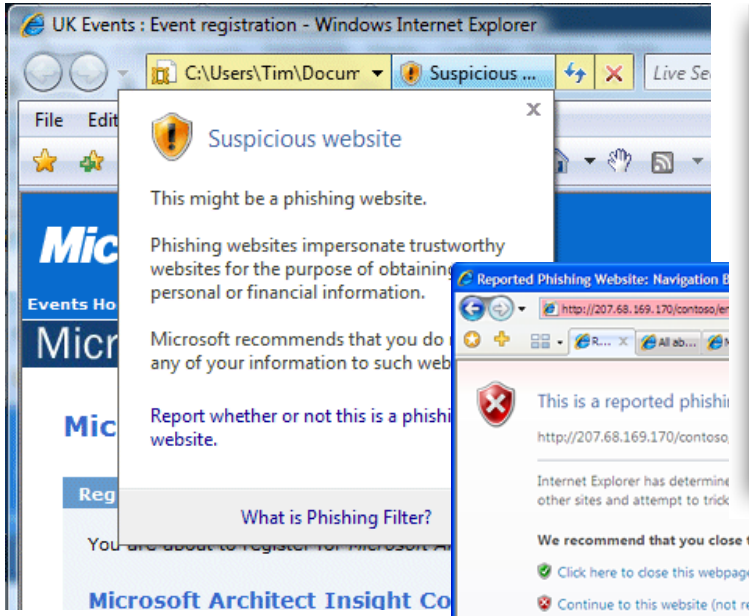
More Details

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

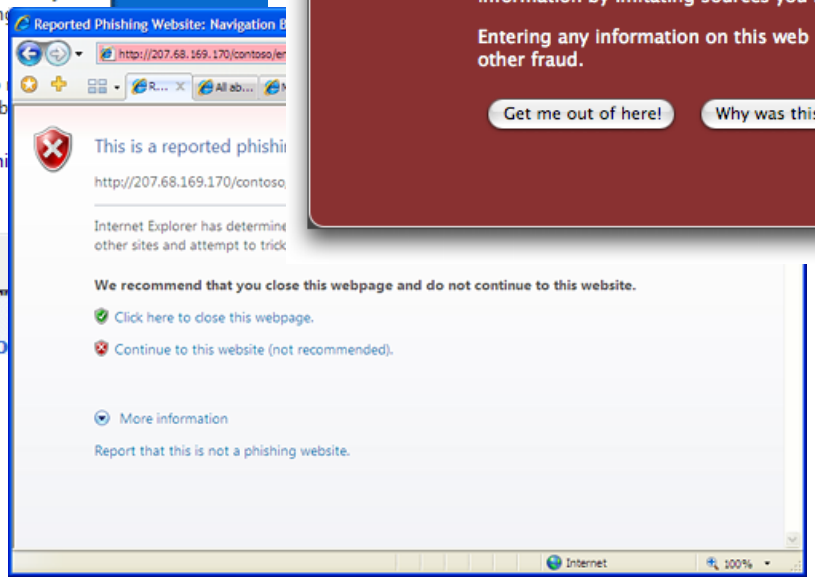
- 70% of responses within first 12 hours
- Adversary wins by gaining users' trust

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

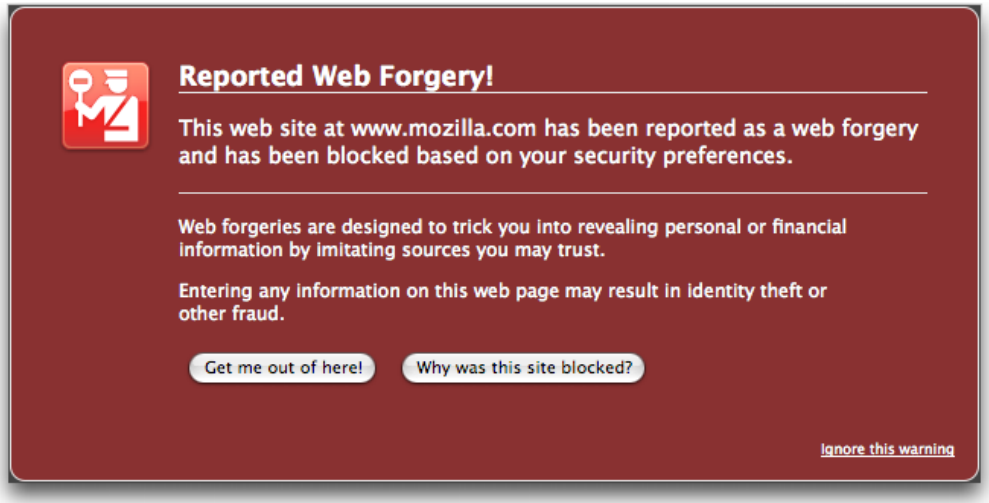
Phishing Warnings



Passive (IE)



Active (IE)



Active (Firefox)

[Egelman et al.]

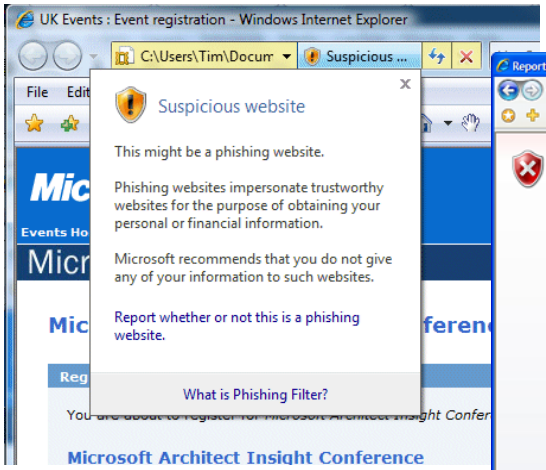
Are Phishing Warnings Effective?

- 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 <u>active</u> and <u>passive</u> warnings

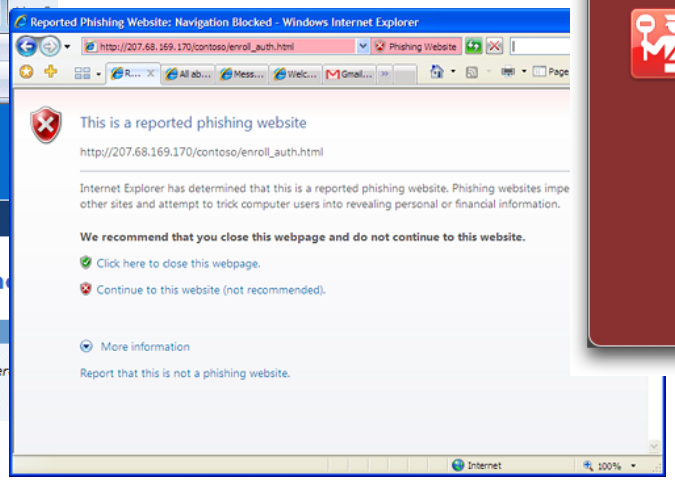
[Egelman et al.]

Active vs. Passive Warnings

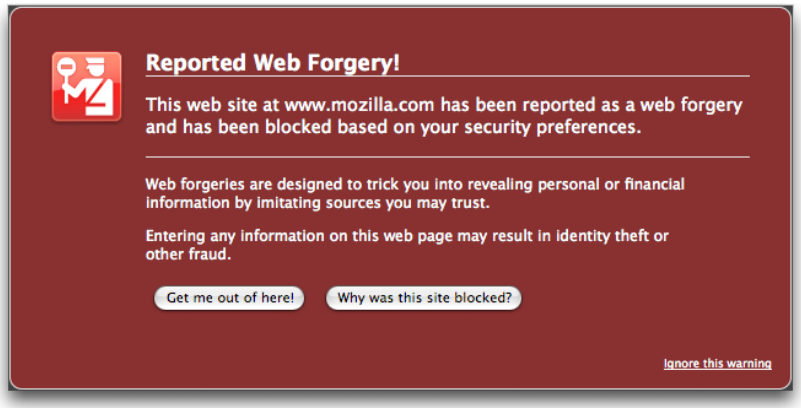
- Active warnings significantly more effective
  - Passive (IE): 100% clicked, 90% phished
  - Active (IE): 95% clicked, 45% phished
  - Active (Firefox): 100% clicked, 0% phished



Passive (IE)



Active (IE)



Active (Firefox)

[Egelman et al.]

User Response to Warnings

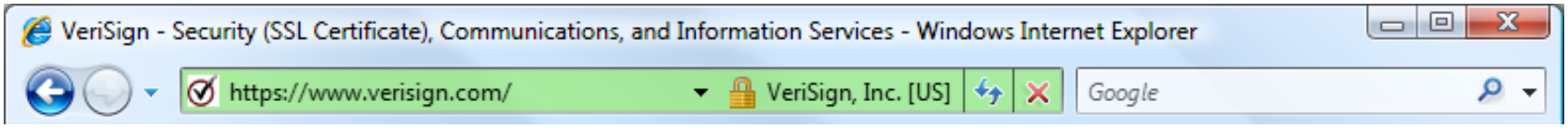
- Some fail to notice warnings entirely
  - Passive warning takes a couple of seconds to appear; if user starts typing, his keystrokes dismiss the warning
- Some saw the warning, closed the window, went back to email, clicked links again, were presented with the same warnings… repeated 4-5 times
  - 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"

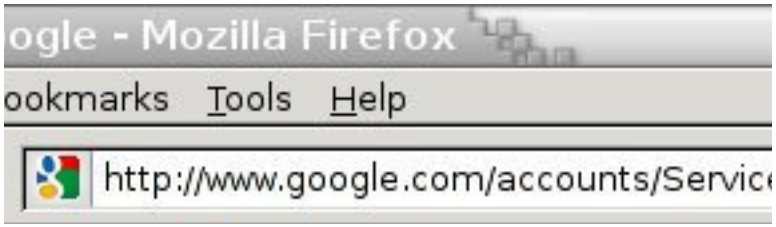
The Lock Icon



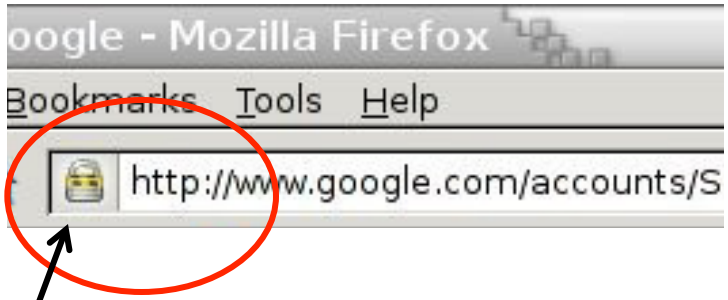
- Goal: identify secure connection
  - SSL/TLS is being used between client and server to protect against active network attacker
- Lock icon should only be shown when the page is secure against network attacker
  - Semantics subtle and not widely understood by users
  - Whose certificate is it??
  -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

[Moxie Marlinspike]

Will You Notice?

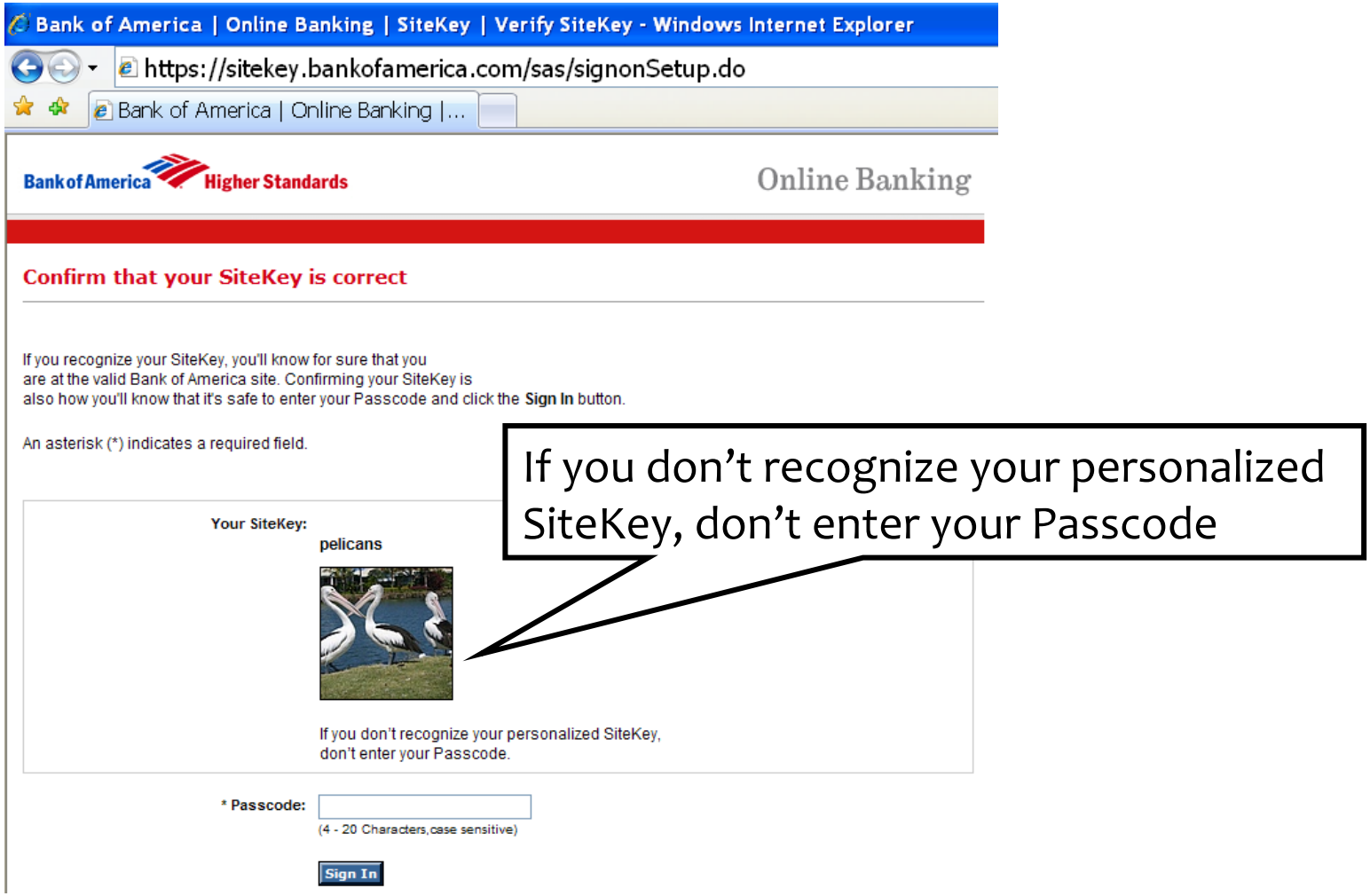


⇒



Clever favicon inserted by network attacker

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!

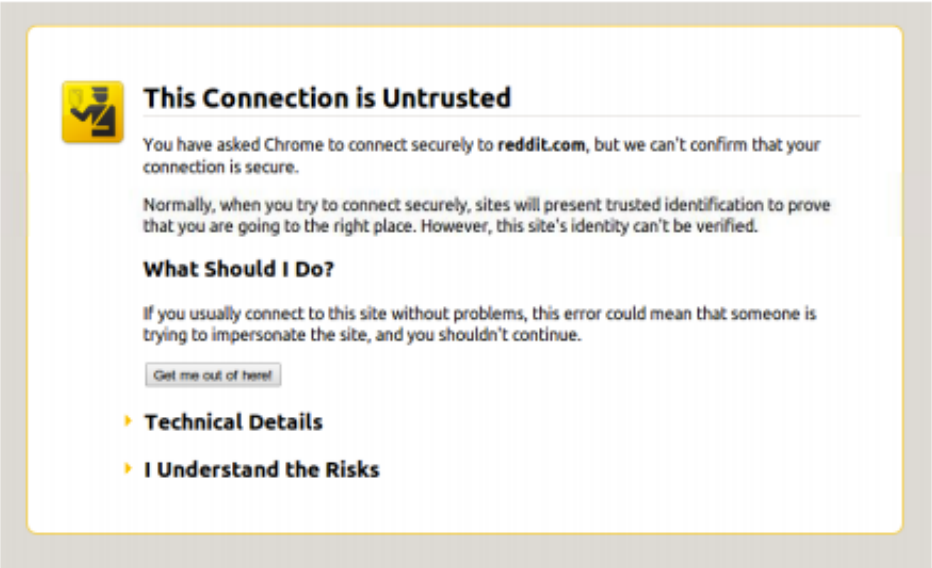
Case Study #2: Browser SSL Warnings

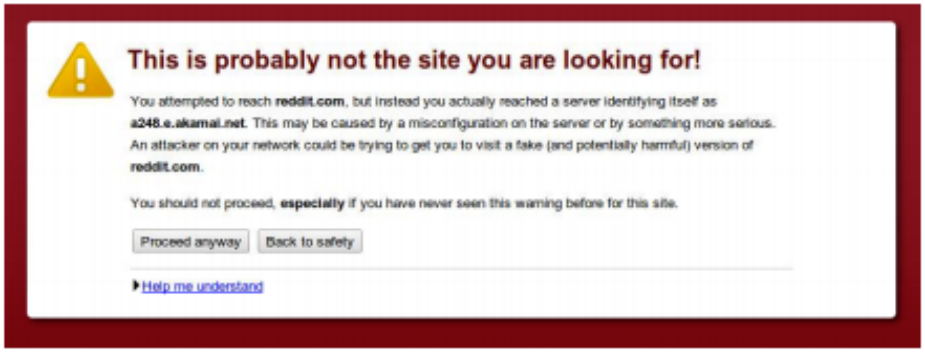
- Design question: How to alert the user if a site's SSL certificate is untrusted?

[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.

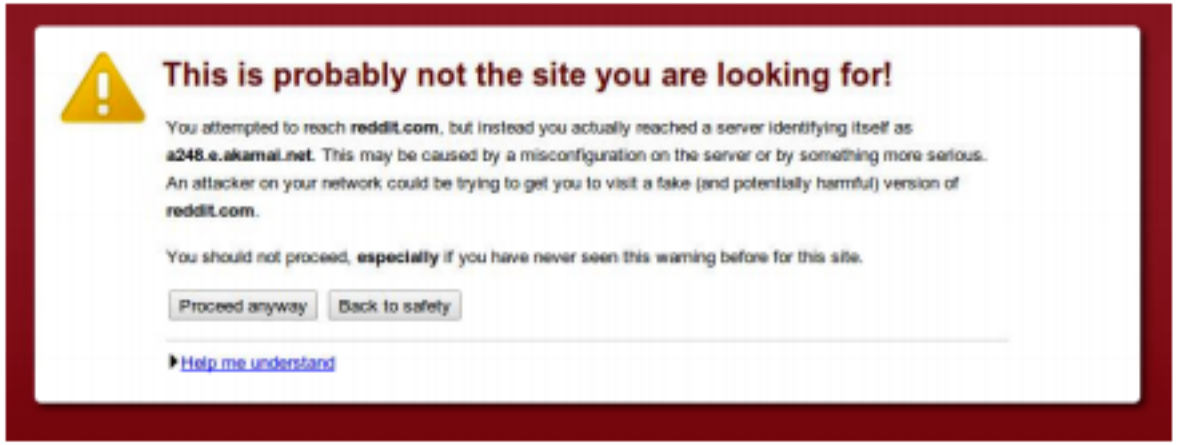


Figure 1. The default Chrome SSL warning (Condition 1).

[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.

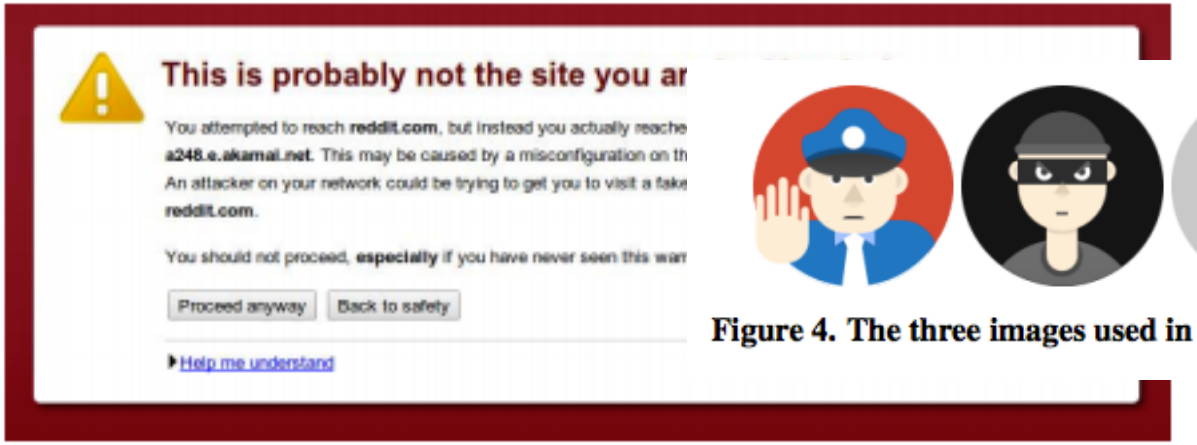


Figure 4. The three images used in Conditions 2-4.

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.

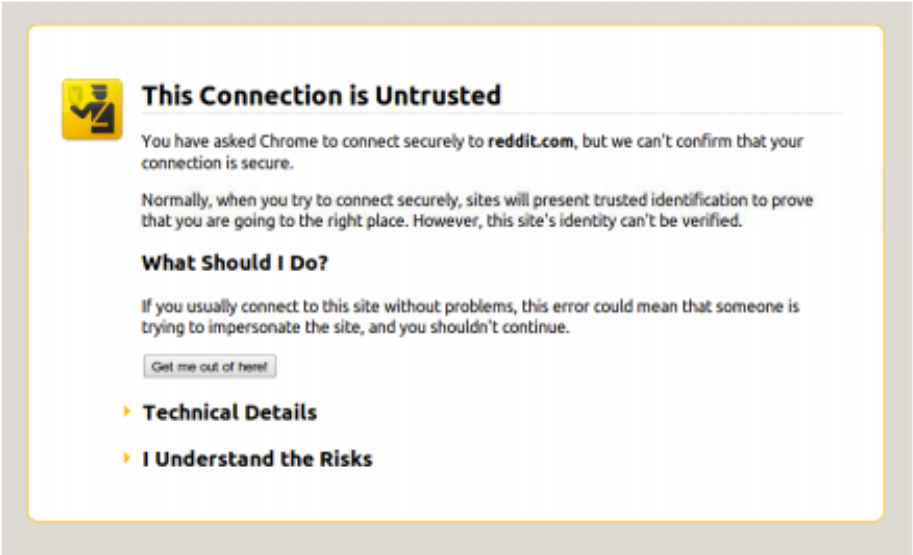


Figure 2. The mock Firefox SSL warning (Condition 5).

[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.

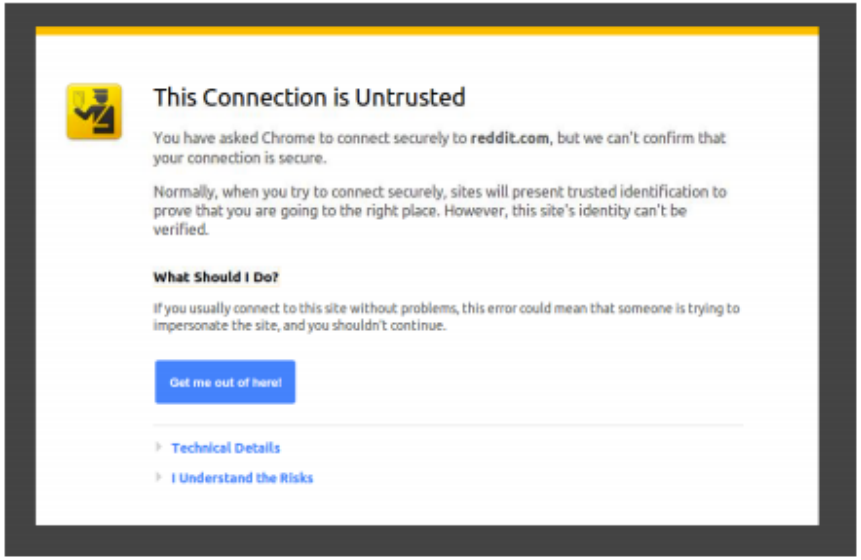
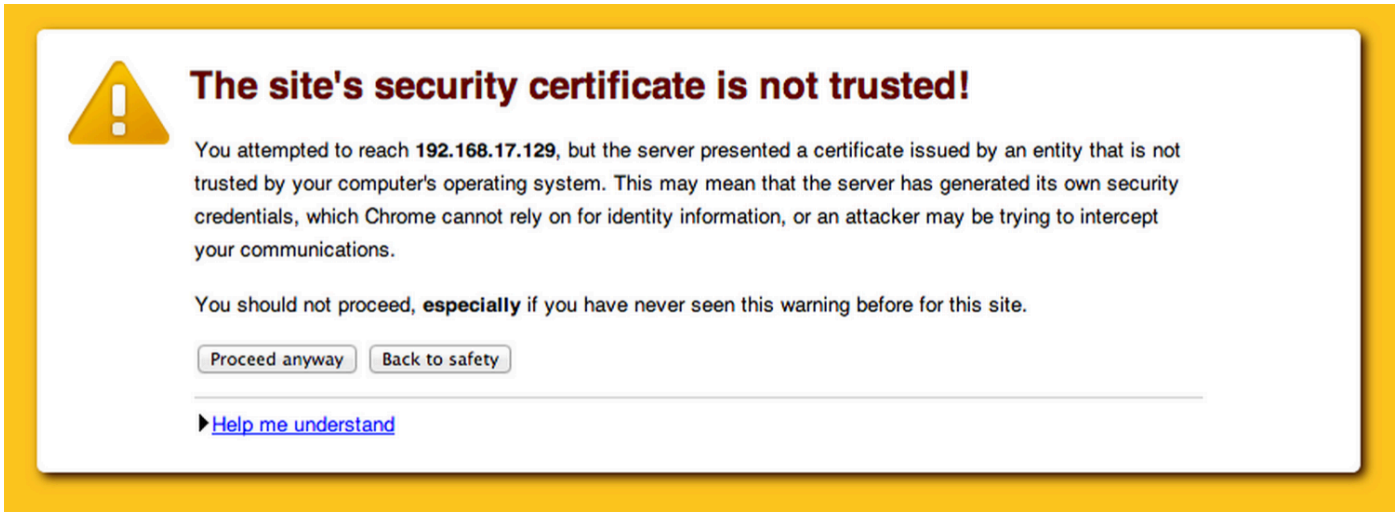


Figure 3. The Firefox SSL warning with Google styling (Condition 7).

[Felt et al.]

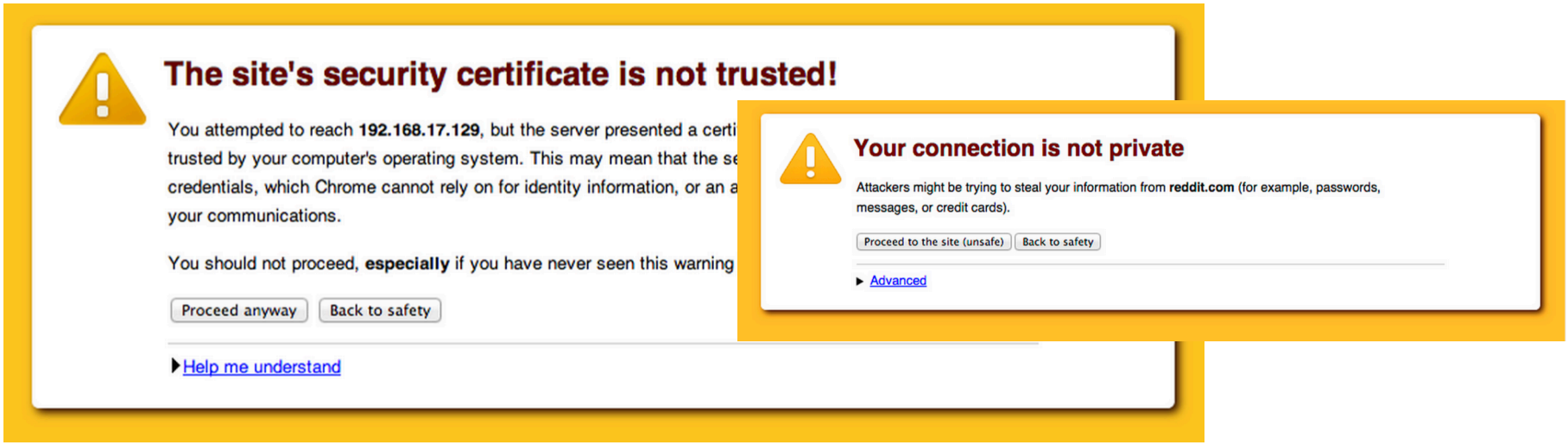
Opinionated Design Helps!

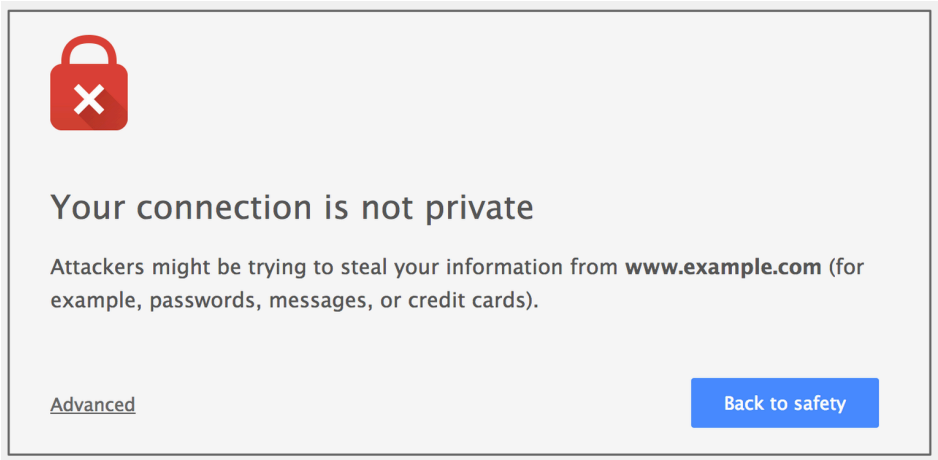


Adherence	N
30.9%	4,551

[Felt et al.]

Opinionated Design Helps!

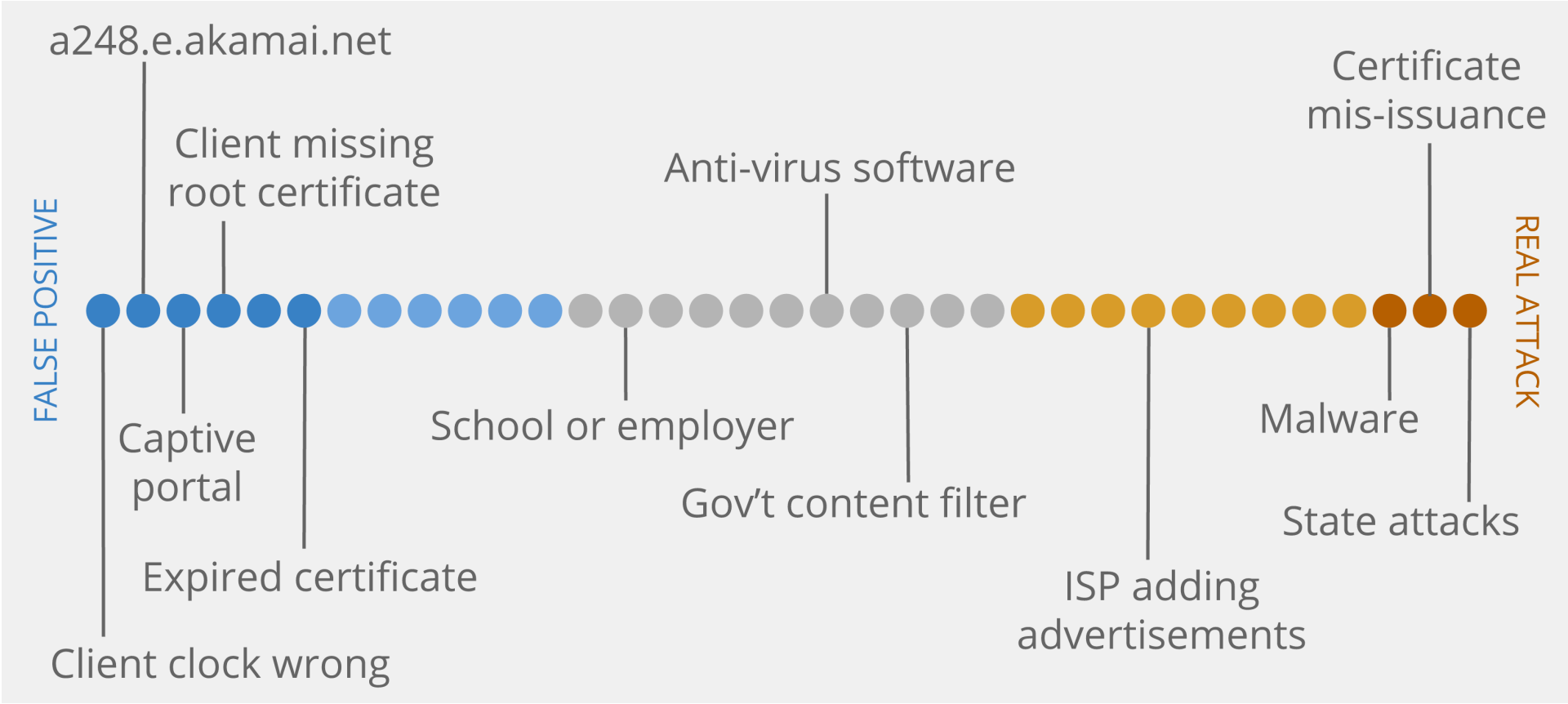




Adherence	N
30.9%	4,551
32.1%	4,075
58.3%	4,644

[Felt et al.]

Challenge: Meaningful Warnings



Password Managers

- Separate application and/or extension in your browser.
- Remembers and automatically enters passwords on your behalf.
- Seems possibly **easier** than remembering all your passwords. Is it **more secure?**

Question

- **Q.** What are the root causes of usability issues in computer security?

Issue #1: Complexities, Lack of Intuition

Real World

We can see, understand, relate to.

Electronic World

Too complex, hidden, no intuition.

Issue #1: Complexities, Lack of Intuition

- Mismatch between perception of technology and what really happens
  - Public keys?
  - Signatures?
  - Encryption?
  - Message integrity?
  - Chosen-plaintext attacks?
  - Chosen-ciphertext attacks?
  - Password management?
  - …

Issue #2: Who's in Charge?

Real World

Electronic World

Users want to feel like they're in control.

Where analogy breaks down: *Adversaries* in the electronic world can be *intelligent*, *sneaky*, and *malicious*.

Complex, hidden, but *doctors manage*

Complex, hidden, and *users manage*

Issue #2: Who's in Charge?

- Systems developers should help protect users
  - Usable authentication systems
  - Usable privacy settings (e.g., on social media)
  - User-driven access control
- Software applications help users manage their applications
  - Anti-virus software
  - Anti-web tracking browser add-ons
  - PwdHash, Keychain for password management
  - Some say: Can we trust software for these tasks?

Issue #3: Hard to Gauge Risks

"It won't happen to me!" (Sometimes a reasonable assumption, sometimes not.)

Schneier on Security

A weblog covering security and security technology.

« The Emergence of a Global Infrastructure for Mass Registration and Surveillance | Main | PDF Redacting Failure »

May 02, 2005

Users Disabling Security

It's an old story: users disable a security measure because it's annoying, allowing an attacker to bypass the measure.

> A [redacted] accused in a deadly courthouse rampage was able to enter the chambers of the judge slain in the attack and hold the occupants hostage because the door was unlocked and a buzzer entry system was not activated, a sheriff's report says.

Security doesn't work unless the users want it to work. This is true on the personal and national scale, with or without technology.

Issue #4: No Accountability

- Issue #3 is amplified when users are not held accountable for their actions
  - E.g., from employers, service providers, etc.
  - (Not all parties will perceive risks the same way)

- Also, recall that a user's poor security choices may affect **other** people
  - E.g., compromise account of user with weak password, then exploit a local (rather than remote) vulnerability to get root access

Issue #5: Annoying, Awkward, or Difficult

- Difficult
  - Remembering 50 different, "random" passwords
- Awkward
  - Lock computer screen every time leave the room
- Annoying
  - Browser warnings, virus alerts, forgotten passwords, firewalls

- Consequence:
  - Changing user's knowledge may **<u>not</u>** affect their behavior

Issue #6: Social Issues

- Public opinion, self-image
  - Only "nerds" or the "super paranoid" follow security guidelines
- Unfriendly
  - Locking computers suggests distrust of co-workers
- Annoying
  - Sending encrypted emails that say, "what would you like for lunch?"

Issues with Usability

1. Lack of intuition
   - See a safe, understand threats. Not true for computers.
2. Who's in charge?
   - Doctors keep your medical records safe, you manage your passwords.
3. Hard to gauge risks
   - "It would never happen to me!"
4. No accountability
   - Asset-holder is not the only one you can lose assets.
5. Awkward, annoying, or difficult
6. Social issues

Question

- **Q.** What approaches can we take to mitigate usability issues in computer security?

Response #1: Education and Training

- Education:
  - Teaching technical concepts, risks
- Training
  - Change behavior through:
    - Drill
    - Monitoring
    - Feedback
    - Reinforcement
    - Punishment
- May be <u>part</u> of the solution – but not <u>the</u> solution

Response #2: Security Should Be Invisible

- Security should happen
  - Naturally
  - By Default
  - Without user input or understanding

- Recognize and stop bad actions
- Starting to see some invisibility
  - SSL/TLS
  - VPNs
  - Automatic Security Updates
  - User-driven access control

Response #2: Security Should Be Invisible

- "Easy" at extremes, or for simple examples
  - Don't give everyone access to everything
- But hard to generalize
- Leads to things not working for reasons user doesn't understand
- Users will then try to get the system to work, possibly further <u>reducing</u> security
  - E.g., "dangerous successes" for password managers

Response #3: "3 Word UI": "Are You Sure?"

- Security should be invisible
  - Except when the user tries something dangerous
  - In which case a warning is given
- But how do users evaluate the warning? Two realistic cases:
  - Always heed warning. But see problems / commonality with Response #2 ("security should be invisible")
  - Always ignore the warning. If so, then how can it be effective?

Response #4: Focus on Users, Use Metaphors

- Clear, understandable metaphors:
  - Physical analogs; e.g., red-green lights
- User-centered design: Start with user model
- Unified security model across applications
  - User doesn't need to learn many models, one for each application
- Meaningful, intuitive user input
  - Don't assume things on user's behalf
  - Figure out how to ask so that user can answer intelligently

Response #5: Least Resistance

- "Match the most comfortable way to do tasks with the least granting of authority"
  - Ka-Ping Yee, Security and Usability
- Should be "easy" to comply with security policy
- "Users value and want security and privacy, but they regard them only as secondary to completing the primary tasks"
  - Karat et al, Security and Usability