Experts Are Not Infallible

The Need for Usable System Security

Matthew Smith

Usable Security and Privacy Lab, Universität Bonn

Security Fails – a lot

Online banking users get their credentials phished

Comodo Hack: 37,000 Legitimate Certificates Issued by CAs for Unqualified Names

Stuxnet Virus sets back Iran's Nuclear Program by 2 Years.
Physical damage to facilities

Sony Hack 2011: Personal Information from Approximately 24.6 Million Sony OE Accounts stolen

Security Fails – a lot

Online banking users get their credentials phished

Comodo Hack: 37,000 Legitimate Certificates Issued by CAs for Unqualified Names

Stuxnet Virus sets back Iran's Nuclear Program by 2 Years.
Physical damage to facilities

Sony Hack 2014: Over 100 TB stolen without anybody noticing. Including emails, medical records and unreleased scripts and films

Security Fails – a lot

Online banking users get their credentials phished

Comodo Hack: 37,000 Legitimate Certificates Issued by CAs for Unqualified Names

Stuxnet Virus sets back Iran’s Nuclear Program by 2 Years.
Physical damage to facilities

Sony Hack 2014 No. 2: Hacker Group Lizard Squad Takes Down Sony's PlayStation Network for a couple of days

Security is hard!

Our goal is to make it easy

Solution: Usable Security and Privacy

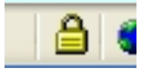
- Three seminal papers are seen as the origin of Usable Security and Privacy research:
  - Zurko and Simon's: "User-Centered Security"
  - Adams and Sasse's: "Users Are Not the Enemy"
  - Whitten and Tygar's "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0"
- All argued that users should not be seen as the problem to be dealt with,
  - but that security experts need to communicate more with users, and adopt user-centered design approaches.

Usable Security Research Example: HTTPS

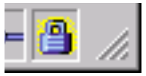
HTTPS Part 1:
Security Indicators

HTTPS Indicators (old)

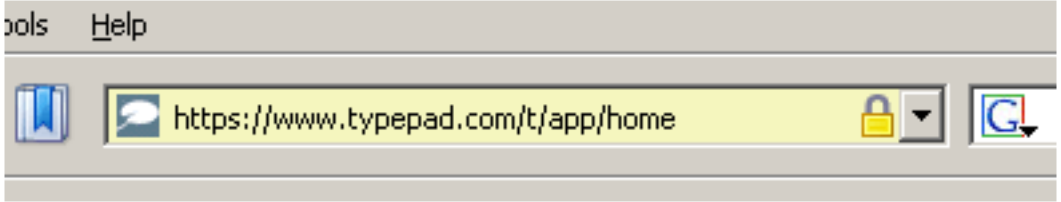
- Microsoft IE



- Mozilla

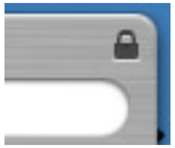


- Firefox



https://www.typepad.com/t/app/home

- Safari



The Emperor’s New Security Indicators An evaluation of website authentication and the effect of role playing on usability studies

(2007)

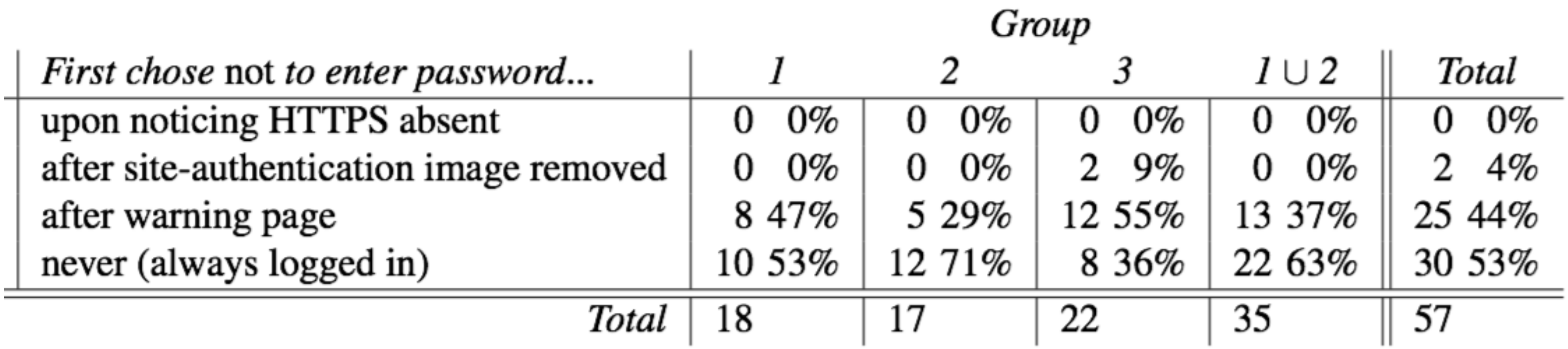
Stuart E. Schechter
MIT Lincoln Laboratory

Rachna Dhamija
Harvard University & CommerceNet

Andy Ozment
MIT Lincoln Laboratory & University of Cambridge

Ian Fischer
Harvard University

Study Results

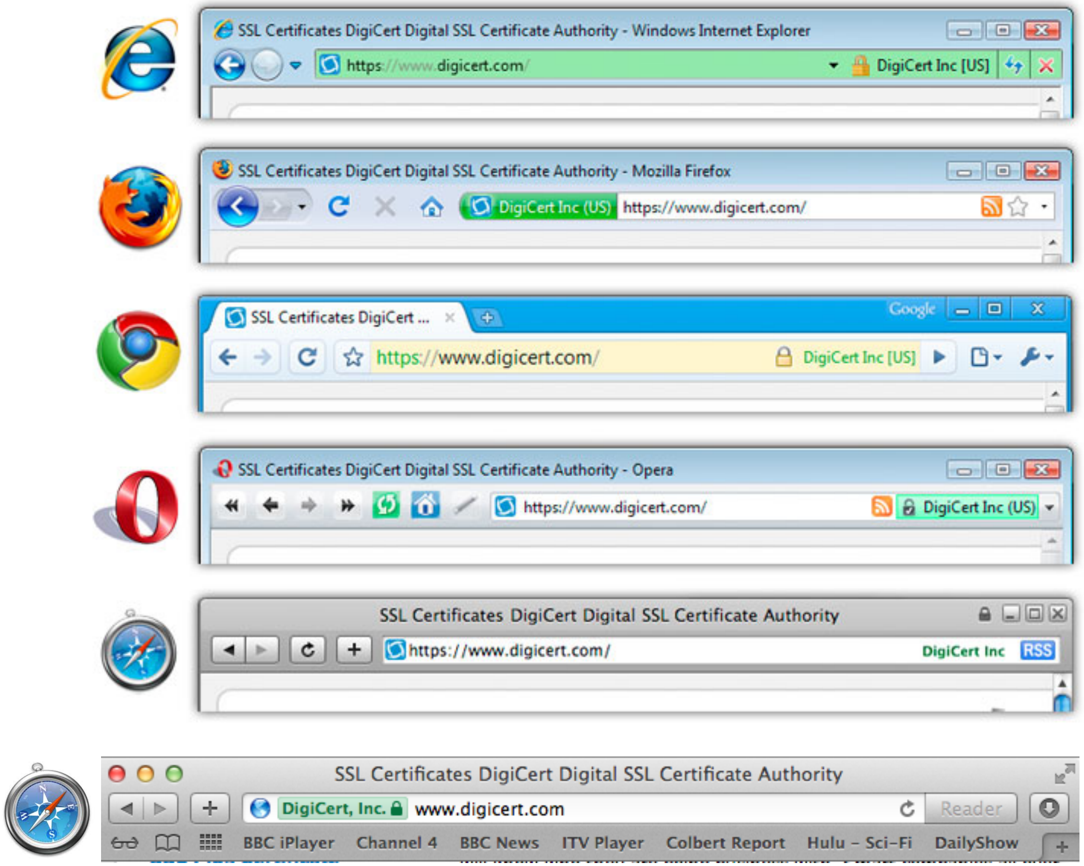


	Group									
First chose not *to enter password...*	*1*		*2*		*3*		*1* ∪ *2*		*Total*	
upon noticing HTTPS absent	0	0%	0	0%	0	0%	0	0%	0	0%
after site-authentication image removed	0	0%	0	0%	2	9%	0	0%	2	4%
after warning page	8	47%	5	29%	12	55%	13	37%	25	44%
never (always logged in)	10	53%	12	71%	8	36%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

- All participants entered their passwords after HTTPS indicators were removed, including all 27 who were using their own account credentials.

HTTPS Indicators (newer)

- Made more visible
- Security “signals”
  - Green = all is well
- But things still change on a regular basis
- Effectiveness still isn’t great



SSL Certificates DigiCert Digital SSL Certificate Authority - Windows Internet Explorer

https://www.digicert.com/ DigiCert Inc [US]

SSL Certificates DigiCert Digital SSL Certificate Authority - Mozilla Firefox

DigiCert Inc (US) https://www.digicert.com/

SSL Certificates DigiCert ...

https://www.digicert.com/ DigiCert Inc [US]

SSL Certificates DigiCert Digital SSL Certificate Authority - Opera

https://www.digicert.com/ DigiCert Inc (US)

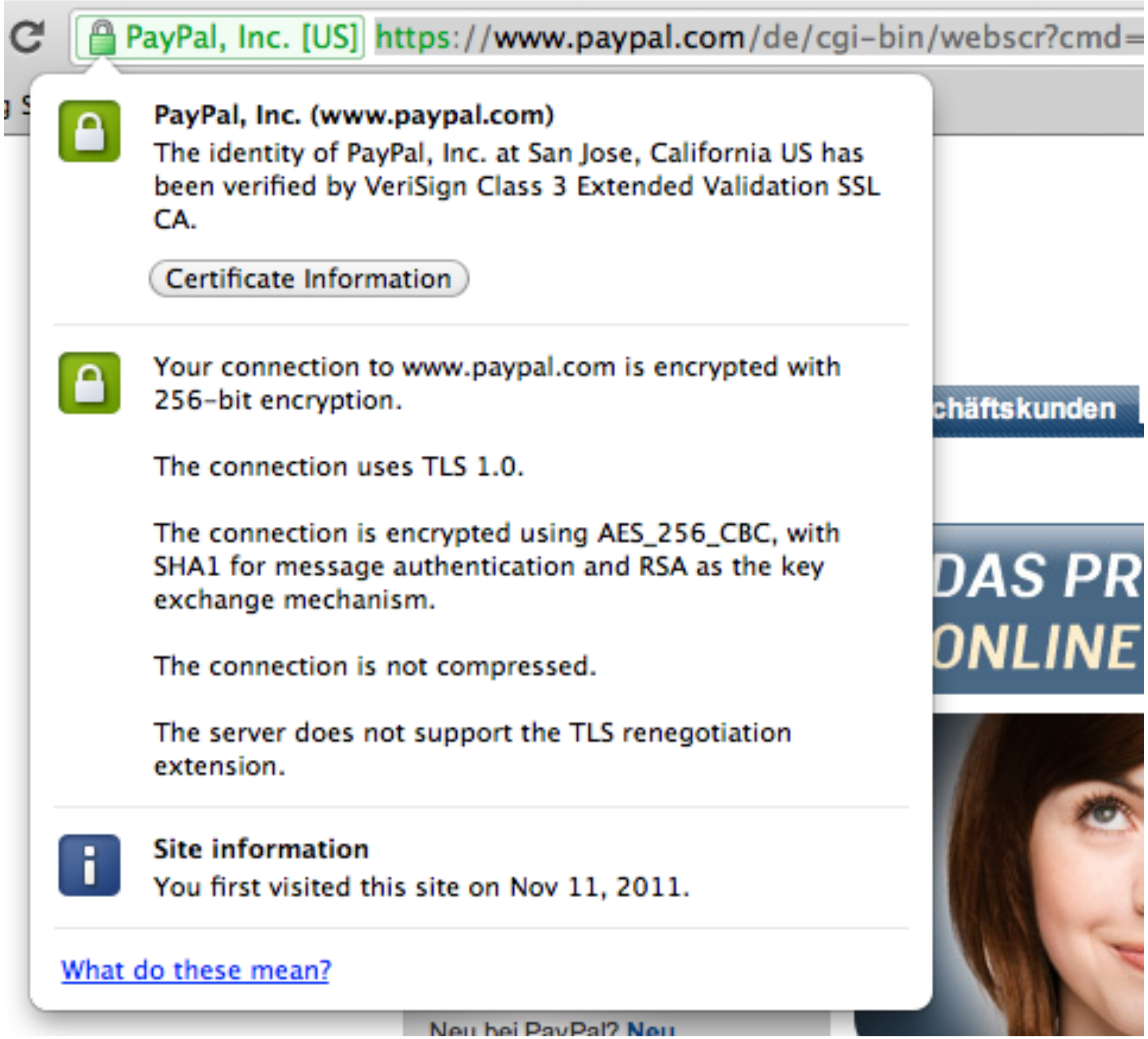
SSL Certificates DigiCert Digital SSL Certificate Authority

https://www.digicert.com/ DigiCert Inc

SSL Certificates DigiCert Digital SSL Certificate Authority

DigiCert, Inc. www.digicert.com

Would you trust…?



PayPal, Inc. [US] https://www.paypal.com/de/cgi-bin/webscr?cmd=

PayPal, Inc. (www.paypal.com)

The identity of PayPal, Inc. at San Jose, California US has been verified by VeriSign Class 3 Extended Validation SSL CA.

Your connection to www.paypal.com is encrypted with 256-bit encryption.

The connection uses TLS 1.0.

The connection is encrypted using AES_256_CBC, with SHA1 for message authentication and RSA as the key exchange mechanism.

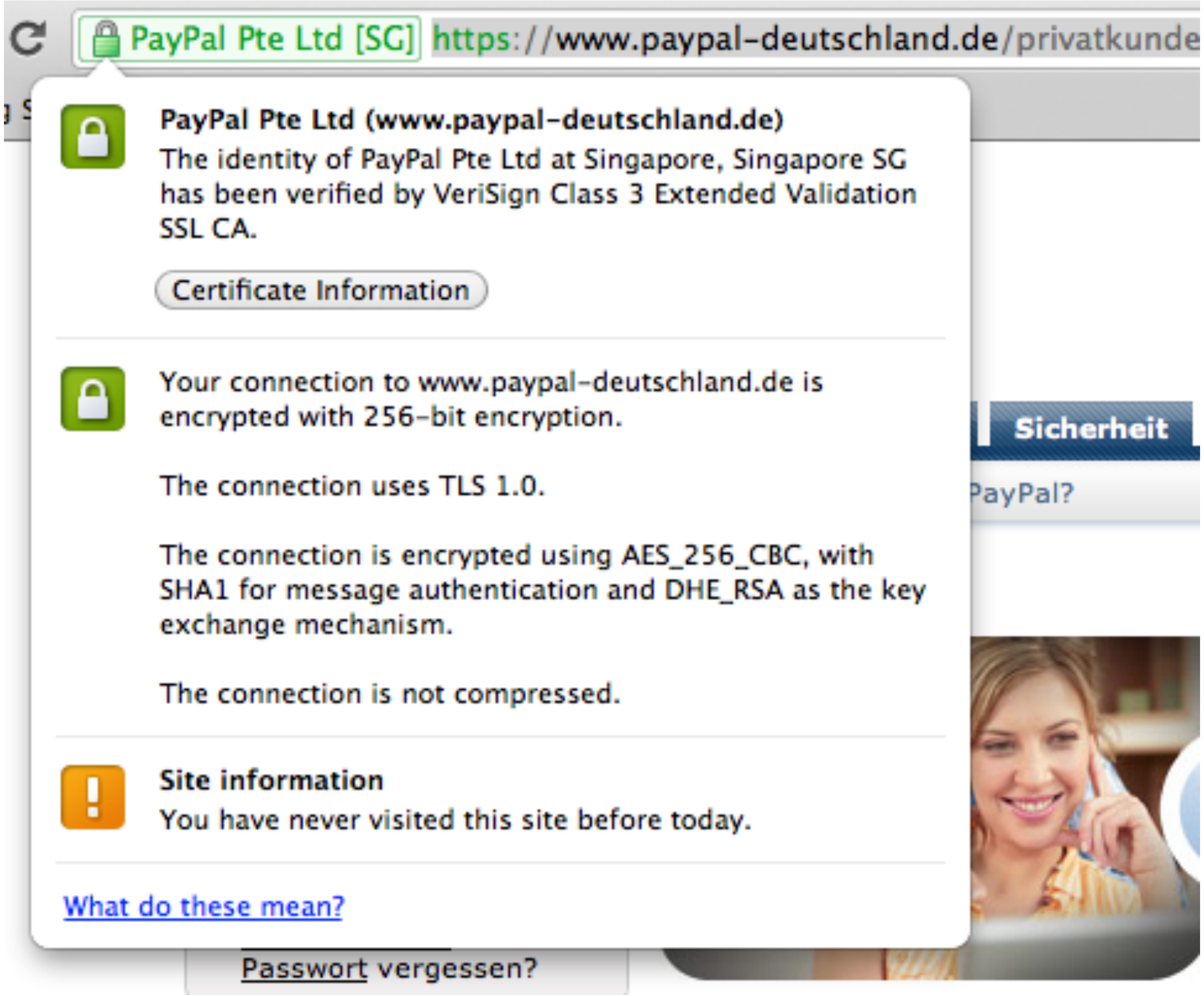
The connection is not compressed.

The server does not support the TLS renegotiation extension.

Site information

You first visited this site on Nov 11, 2011.

Would you trust…?



PayPal Pte Ltd (www.paypal-deutschland.de)
The identity of PayPal Pte Ltd at Singapore, Singapore SG has been verified by VeriSign Class 3 Extended Validation SSL CA.

Certificate Information

Your connection to www.paypal-deutschland.de is encrypted with 256-bit encryption.

The connection uses TLS 1.0.

The connection is encrypted using AES_256_CBC, with SHA1 for message authentication and DHE_RSA as the key exchange mechanism.

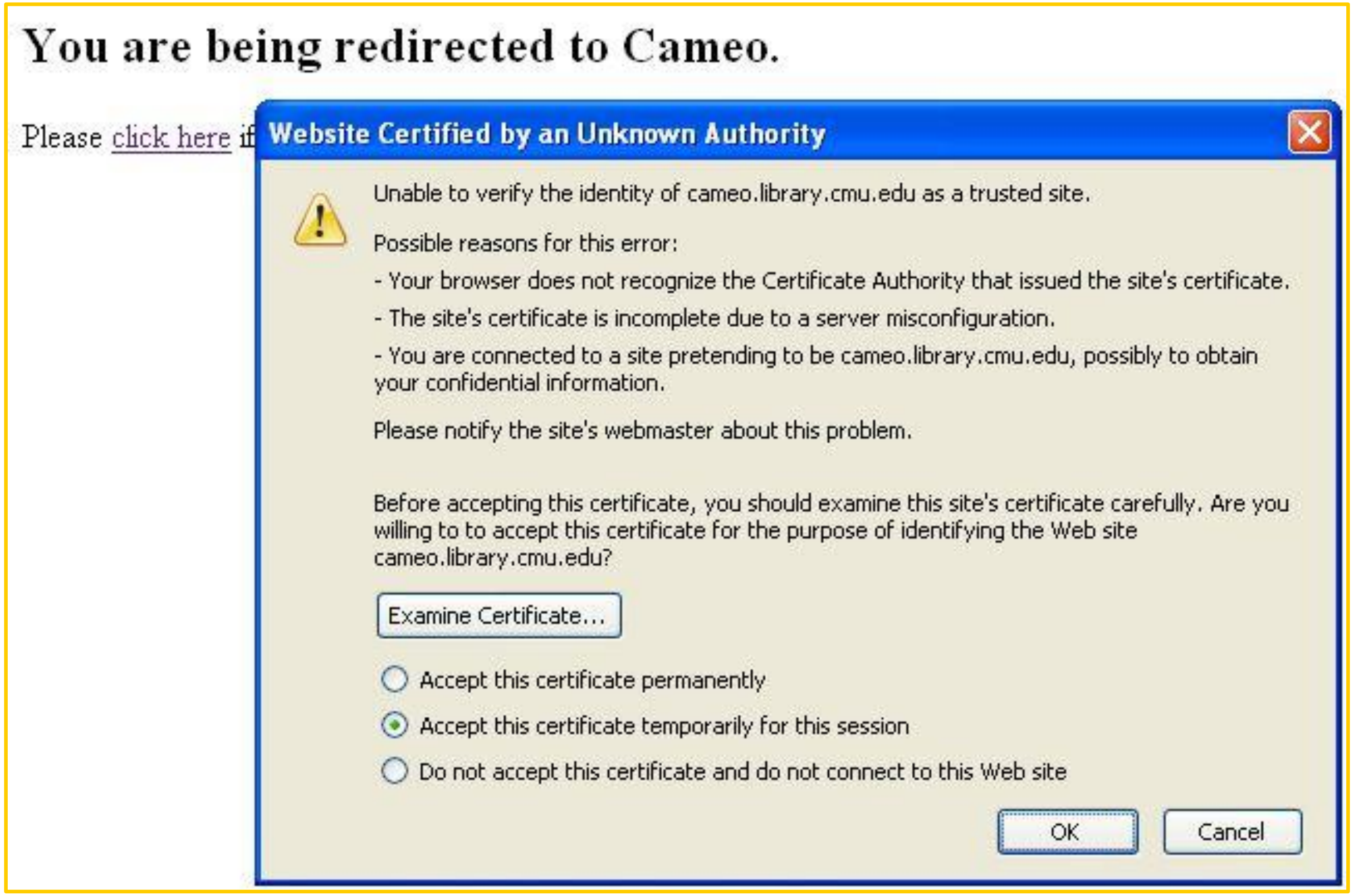
The connection is not compressed.

Site information
You have never visited this site before today.

What do these mean?

HTTPS Part 2:
Security Warnings

Firefox 2 Warning



You are being redirected to Cameo.

Please click here if

Website Certified by an Unknown Authority

Unable to verify the identity of cameo.library.cmu.edu as a trusted site.

Possible reasons for this error:

- Your browser does not recognize the Certificate Authority that issued the site's certificate.
- The site's certificate is incomplete due to a server misconfiguration.
- You are connected to a site pretending to be cameo.library.cmu.edu, possibly to obtain your confidential information.

Please notify the site's webmaster about this problem.

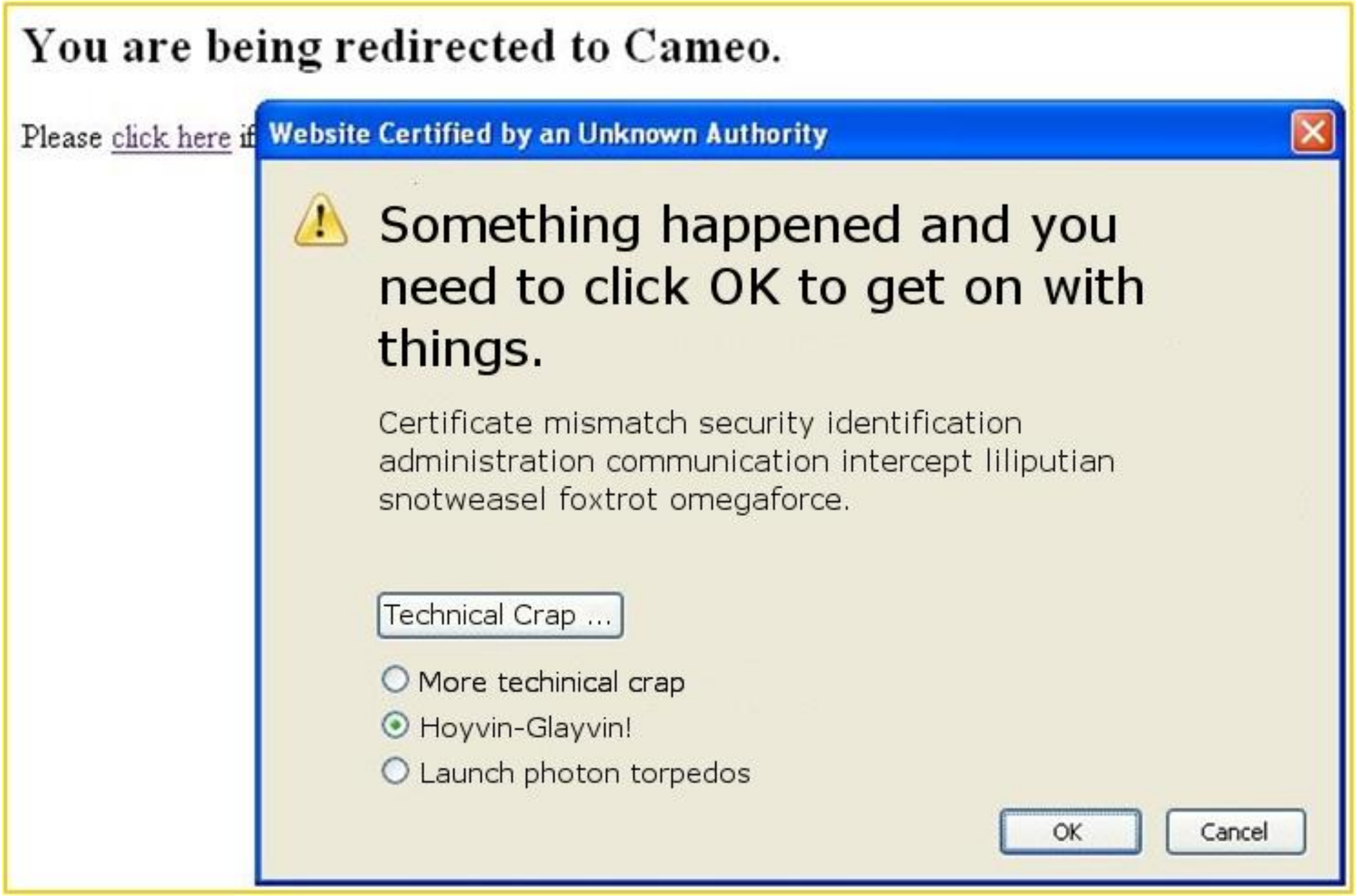
Before accepting this certificate, you should examine this site's certificate carefully. Are you willing to to accept this certificate for the purpose of identifying the Web site cameo.library.cmu.edu?

○ Accept this certificate permanently

● Accept this certificate temporarily for this session

○ Do not accept this certificate and do not connect to this Web site

What users actually see



You are being redirected to Cameo.

Please click here if

Something happened and you need to click OK to get on with things.

Certificate mismatch security identification administration communication intercept liliputian snotweasel foxtrot omegaforce.

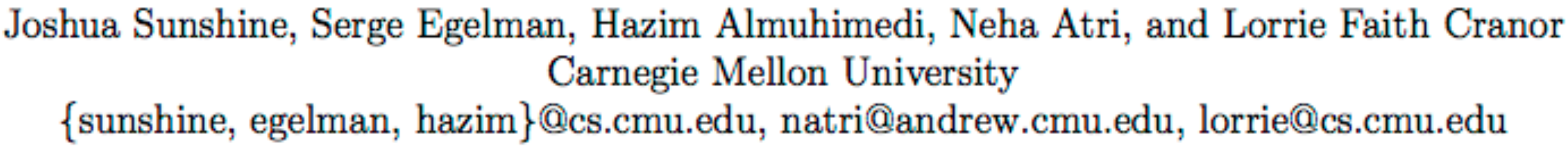
○ More techinical crap

● Hoyvin-Glayvin!

○ Launch photon torpedos

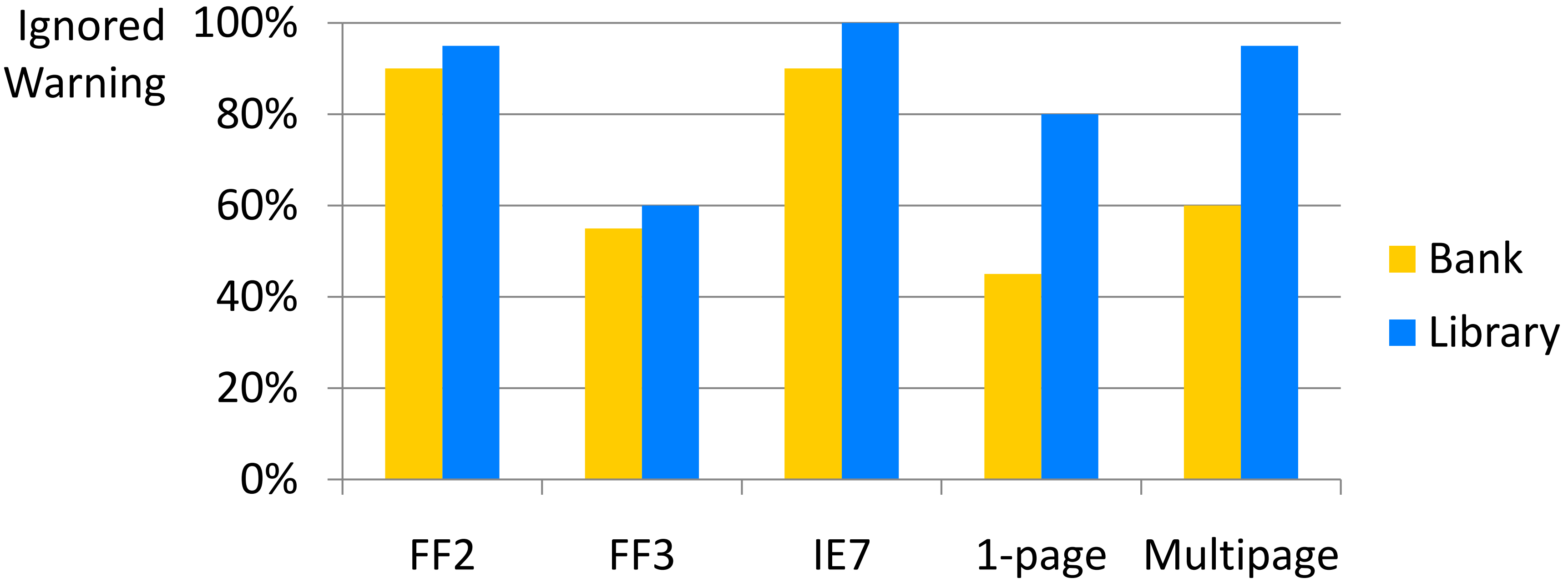
Adapted from Jonathan Nightingale

Crying Wolf: An Empirical Study of SSL Warning Effectiveness (2009)



Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor
Carnegie Mellon University
{sunshine, egelman, hazim}@cs.cmu.edu, natri@andrew.cmu.edu, lorrie@cs.cmu.edu

Library vs Bank Results



- In native warning conditions, no significant difference in reactions at library and bank
- In new warning conditions, users more likely to heed warnings at bank than at library

On the Challenges in Usable Security Lab Studies: Lessons Learned from Replicating a Study on SSL Warnings
(2011)

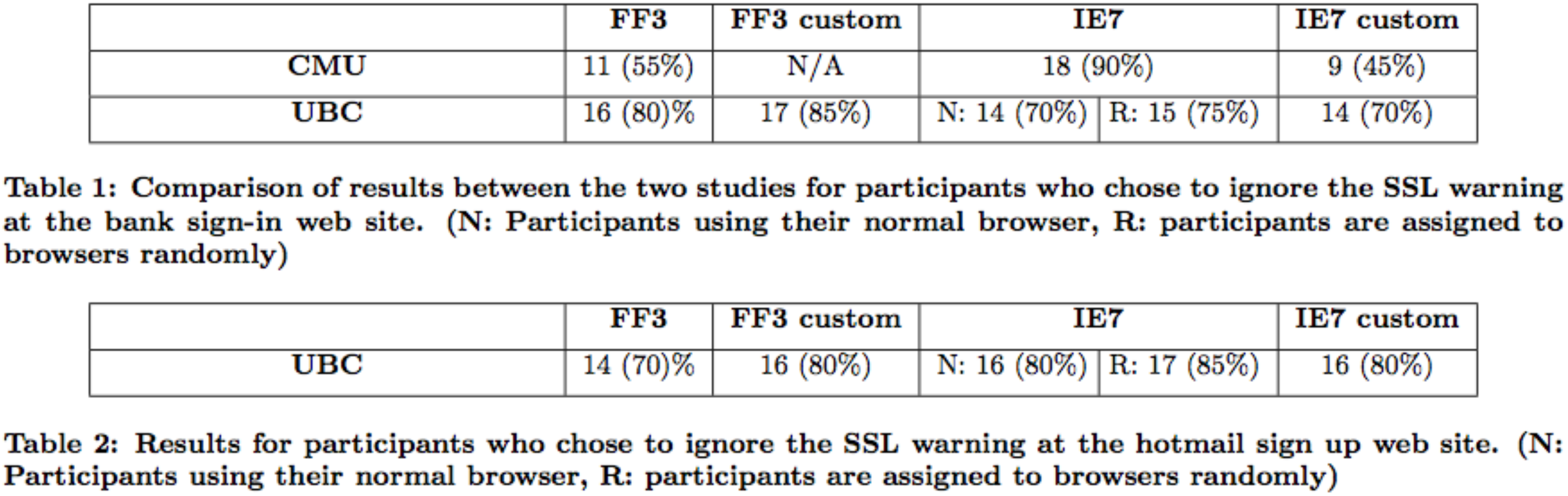
Andreas Sotirakopoulos
University of British Columbia
Vancouver, BC, Canada
andreass@ece.ubc.ca

Kirstie Hawkey
Dalhousie University
Halifax, NS, Canada
hawkey@cs.dal.ca

Konstantin Beznosov
University of British Columbia
Vancouver, BC, Canada
beznosov@ece.ubc.ca

Effect of Browser/Warning on Behavior

- No statistically significant differences were observed between the various conditions in the study.
- There was also no significant differences between participants who were randomly assigned IE7 and native IE7 users



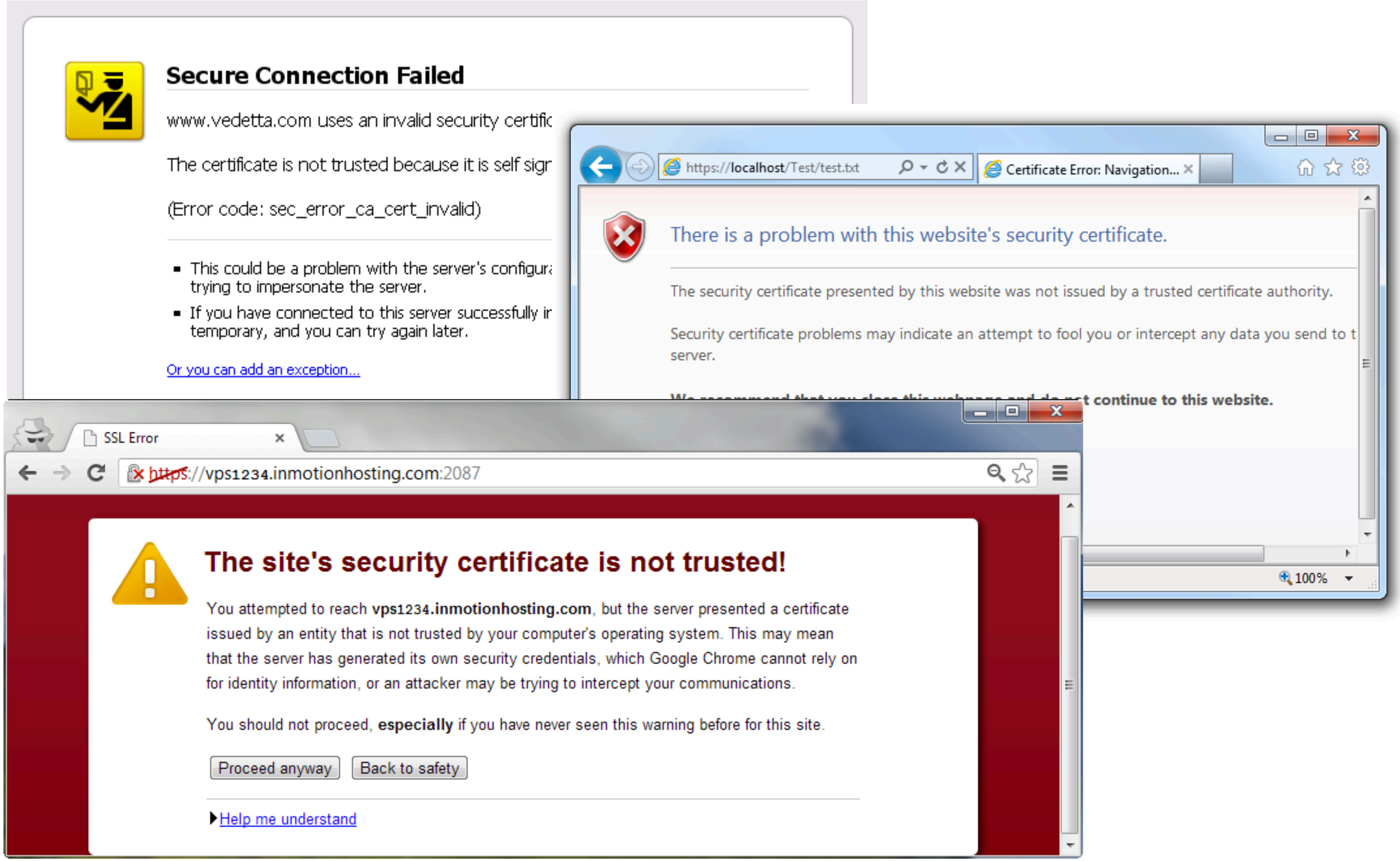
	FF3	FF3 custom	IE7		IE7 custom
CMU	11 (55%)	N/A	18 (90%)		9 (45%)
UBC	16 (80)%	17 (85%)	N: 14 (70%)	R: 15 (75%)	14 (70%)

Table 1: Comparison of results between the two studies for participants who chose to ignore the SSL warning at the bank sign-in web site. (N: Participants using their normal browser, R: participants are assigned to browsers randomly)

	FF3	FF3 custom	IE7		IE7 custom
UBC	14 (70)%	16 (80%)	N: 16 (80%)	R: 17 (85%)	16 (80%)

Table 2: Results for participants who chose to ignore the SSL warning at the hotmail sign up web site. (N: Participants using their normal browser, R: participants are assigned to browsers randomly)

Current HTTPS Warnings



Secure Connection Failed

www.vedetta.com uses an invalid security certific

The certificate is not trusted because it is self sigr

(Error code: sec_error_ca_cert_invalid)

- This could be a problem with the server's configura trying to impersonate the server.
- If you have connected to this server successfully ir temporary, and you can try again later.

Or you can add an exception…

https://localhost/Test/test.txt
Certificate Error: Navigation...
100%

There is a problem with this website's security certificate.

The security certificate presented by this website was not issued by a trusted certificate authority.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to t server.

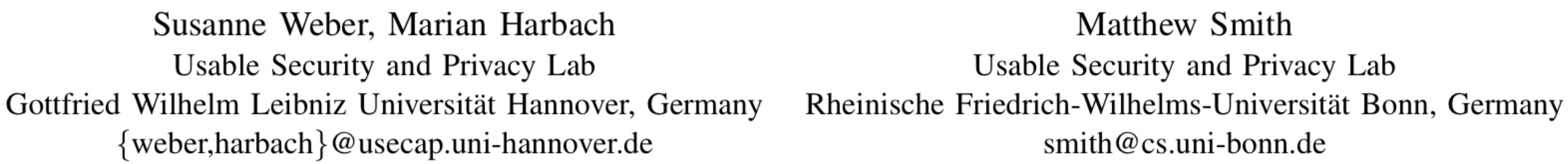
We recommend that you close this webpage and do not continue to this website.

The site's security certificate is not trusted!

You attempted to reach **vps1234.inmotionhosting.com**, but the server presented a certificate issued by an entity that is not trusted by your computer's operating system. This may mean that the server has generated its own security credentials, which Google Chrome cannot rely on for identity information, or an attacker may be trying to intercept your communications.

You should not proceed, **especially** if you have never seen this warning before for this site.

Participatory Design for Security-Related User Interfaces
(2015)

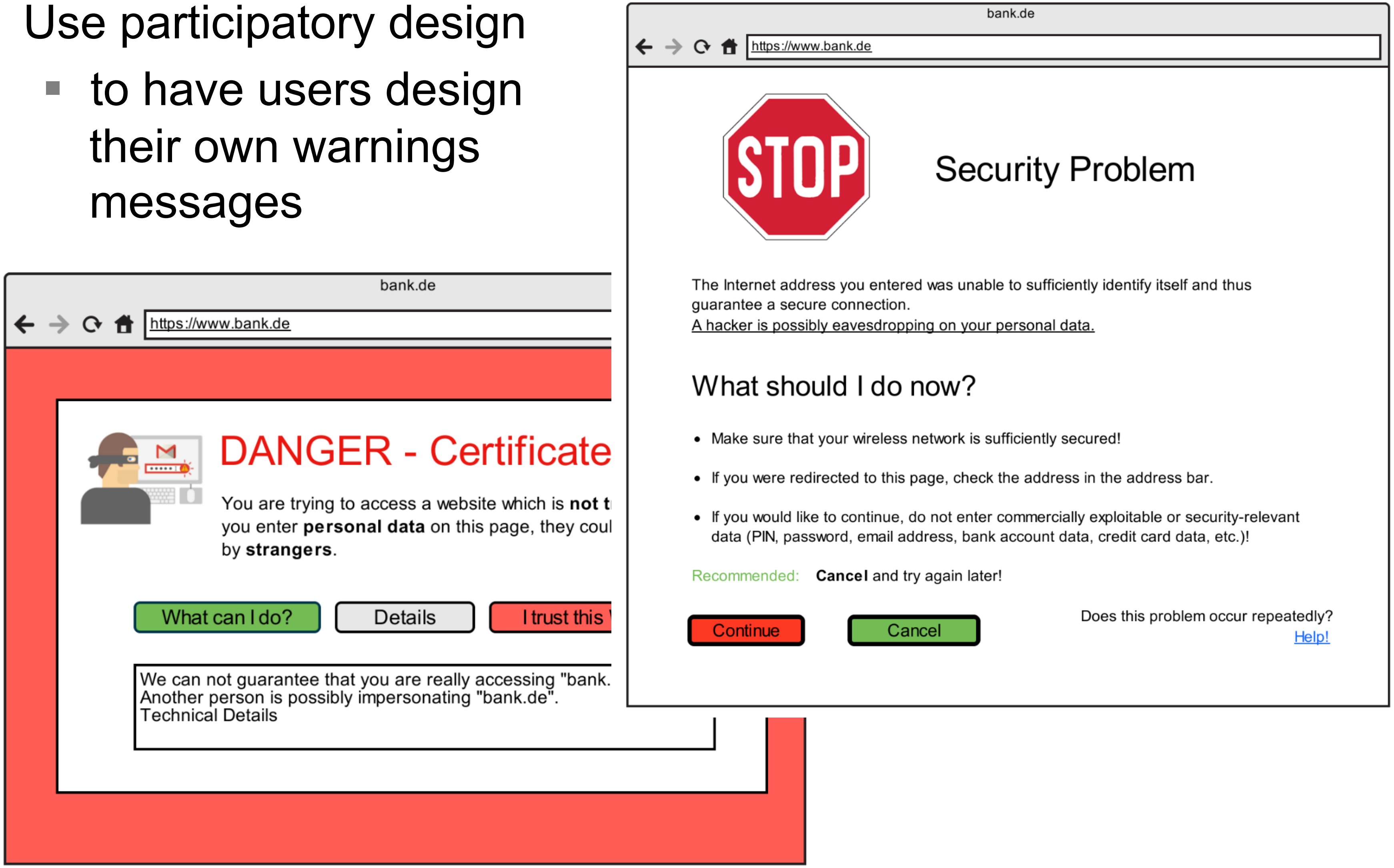


Susanne Weber, Marian Harbach
Usable Security and Privacy Lab
Gottfried Wilhelm Leibniz Universität Hannover, Germany
{weber,harbach}@usecap.uni-hannover.de

Matthew Smith
Usable Security and Privacy Lab
Rheinische Friedrich-Wilhelms-Universität Bonn, Germany
smith@cs.uni-bonn.de

User Involvement

- Use participatory design
  - to have users design their own warnings messages



bank.de

https://www.bank.de

DANGER - Certificate

You are trying to access a website which is **not t**
you enter **personal data** on this page, they coul
by **strangers**.

What can I do? | Details | I trust this '

We can not guarantee that you are really accessing "bank.
Another person is possibly impersonating "bank.de".
Technical Details

bank.de

https://www.bank.de

Security Problem

The Internet address you entered was unable to sufficiently identify itself and thus guarantee a secure connection.
A hacker is possibly eavesdropping on your personal data.

What should I do now?

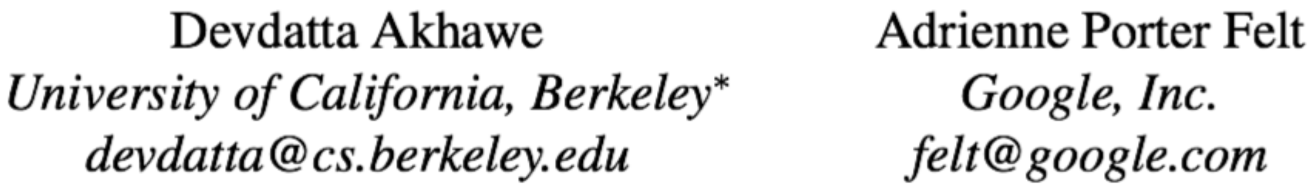
- Make sure that your wireless network is sufficiently secured!
- If you were redirected to this page, check the address in the address bar.
- If you would like to continue, do not enter commercially exploitable or security-relevant data (PIN, password, email address, bank account data, credit card data, etc.)!

Recommended: **Cancel** and try again later!

Continue | Cancel

Does this problem occur repeatedly?
Help!

Alice in Warningland:
A Large-Scale Field Study of Browser Security Warning Effectiveness
(2013)



Devdatta Akhawe
*University of California, Berkeley**
devdatta@cs.berkeley.edu

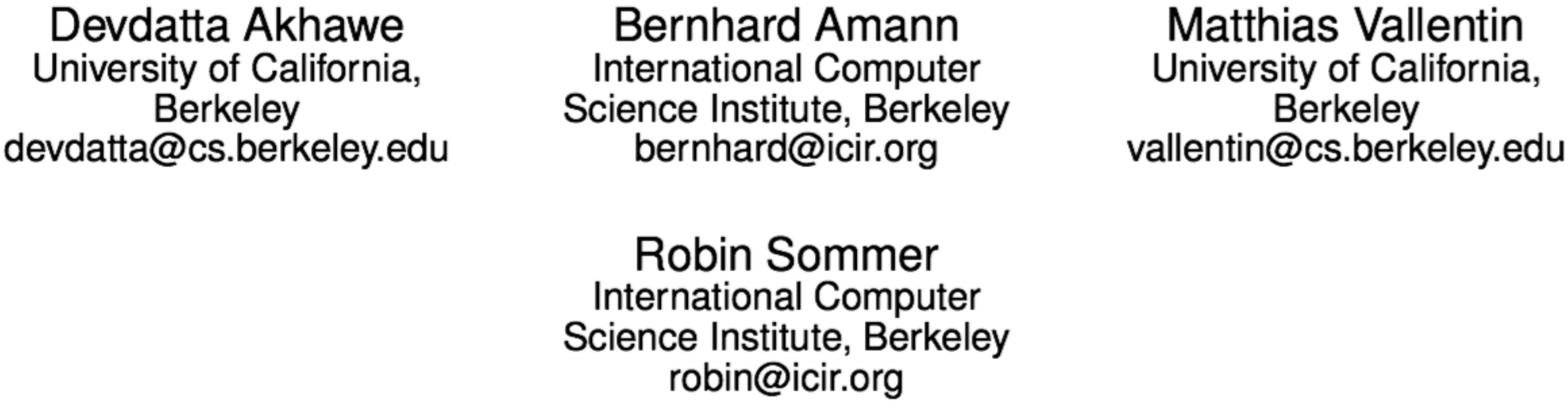
Adrienne Porter Felt
Google, Inc.
felt@google.com

Real World Analysis

- Studied the click-through rate for malware and HTTPS warnings
- Malware
  - Firefox 7.2%
  - Chrome 23.2%
- Phishing
  - Firefox 9.1%
  - Chrome 18.0%
- HTTPS
  - Firefox 33.0%
  - Chrome 70.2%

Here's My Cert, So Trust Me, Maybe? Understanding TLS Errors on the Web

(2013)



Devdatta Akhawe
University of California, Berkeley
devdatta@cs.berkeley.edu

Bernhard Amann
International Computer Science Institute, Berkeley
bernhard@icir.org

Matthias Vallentin
University of California, Berkeley
vallentin@cs.berkeley.edu

Robin Sommer
International Computer Science Institute, Berkeley
robin@icir.org

Real World Analysis

- Studied TLS activity of more than 300,000 users
  - collected certificates passively at egress points of ten network sites
  - over a nine-month period
  - validated certificate chains using browser logic locally
  - 98,46% of the filtered connections validate correctly, implying a false warning rate of 1,54%
- In a scenario with a hypothetical MITMA chance of 1 in 1.000.000
  - 1.000.000 connections would produce 15.401 warnings
  - out of which 15.400 would be false warnings

15.400 to 1

USEC needs you!

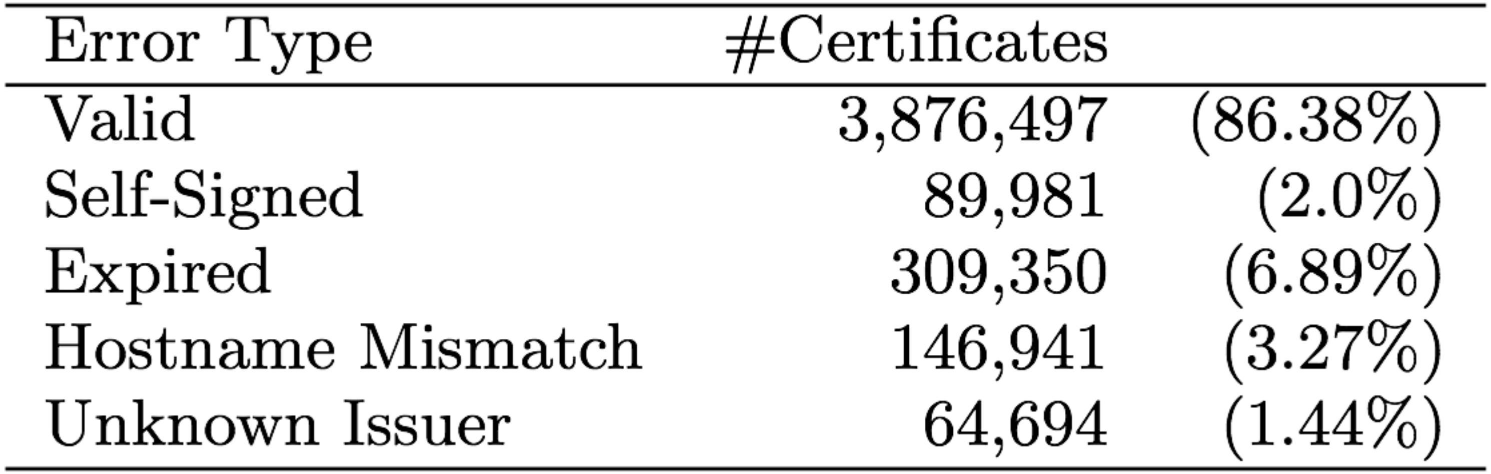
There's more...

- End-users are only a small part of the HTTPS ecosystem
- Administrators are responsible for (mis)configuration web-servers
- Developers are responsible for (mis)using HTTPS in their applications
- Alternative PKI designs might make things better – they might also make them worse...

Chapter 1: Administrators

Scope of the problem

- We used HTTPS certificates collected by Google's web-crawler
  - Period of 12 months
  - ~55.7 million different hosts
  - ~4,49 million different X.509 certificates
  - We extracted all certificates that did not validate correctly based on the Firefox browser logic



Error Type	#Certificates	
Valid	3,876,497	(86.38%)
Self-Signed	89,981	(2.0%)
Expired	309,350	(6.89%)
Hostname Mismatch	146,941	(3.27%)
Unknown Issuer	64,694	(1.44%)

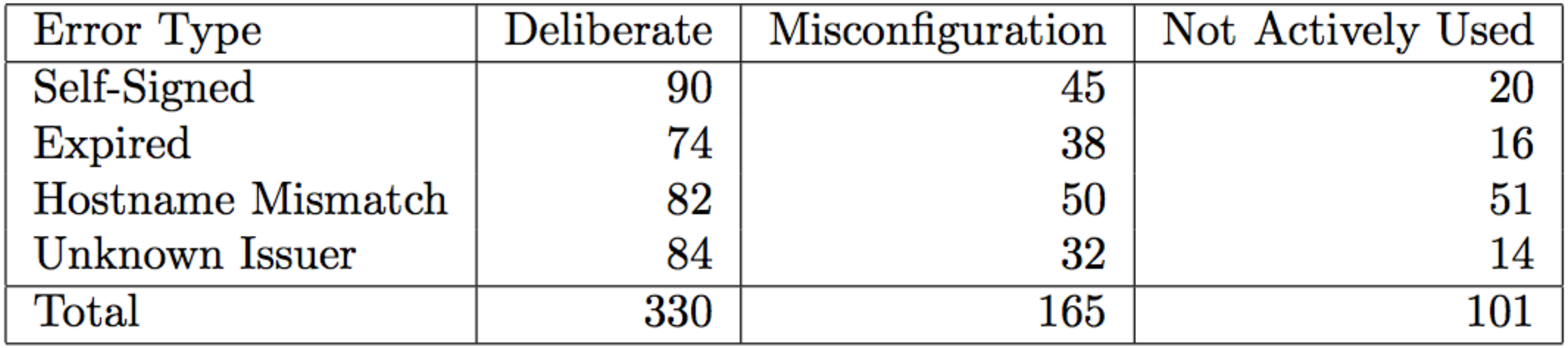
Solutions?

- So what should we do to help the administrators?
  - Create better configuration tools?
  - Reduce the complexity of the entire system?

Find out where the problems lie

- ~4,49 million “bad” certificates
  - We picked a random sample of 50,000
  - Pruned non-current certs down to 46,145
  - And contacted the admins
- We sent 40,473 emails to webmaster@domain.com
- and 5,672 to addresses embedded in the certs.
- Of the 46,145 emails we sent
  - 37,596 could not be delivered to the intended recipient,
  - leaving us with 8,549 successfully delivered surveys
  - 755 complete responses to our survey (~8%)

Find out where the problems lie



Error Type	Deliberate	Misconfiguration	Not Actively Used
Self-Signed	90	45	20
Expired	74	38	16
Hostname Mismatch	82	50	51
Unknown Issuer	84	32	14
Total	330	165	101

- Reasons given in survey
  - ~21% sub-domains/virtual hosts/ redirects
  - ~16% to difficult
  - ~16% for a small group of users
  - ~7% NSA, PRISM & co.
  - ~5% untrusted CA
  - ~3% default configuration
  - ~2% mistake
  - …

- Risk perception
  - ~70% very small
  - ~3% very high
  - ~11% didn‘t know there were warnings

Administrators' wish list

- Lower Price for CA-signed certificates
  - Price is perceived too high for little effort on the CA's side
  - Free CA-signed certificates
  - Cheaper wildcard certificates
- Allow CACert
  - More trust in CACert's web of trust model
- Better Support for Non-Validating Certificates
  - Support for trust-on-first-use, Pinning, TACK
- Better Tool Support
  - OpenSSL command line tool too complicated
  - Server configuration cumbersome, especially for v-hosts
  - Auto-Update Reminder
  - Notification of problems

Chapter 2: Developers

Trust me I‘m an Engineer

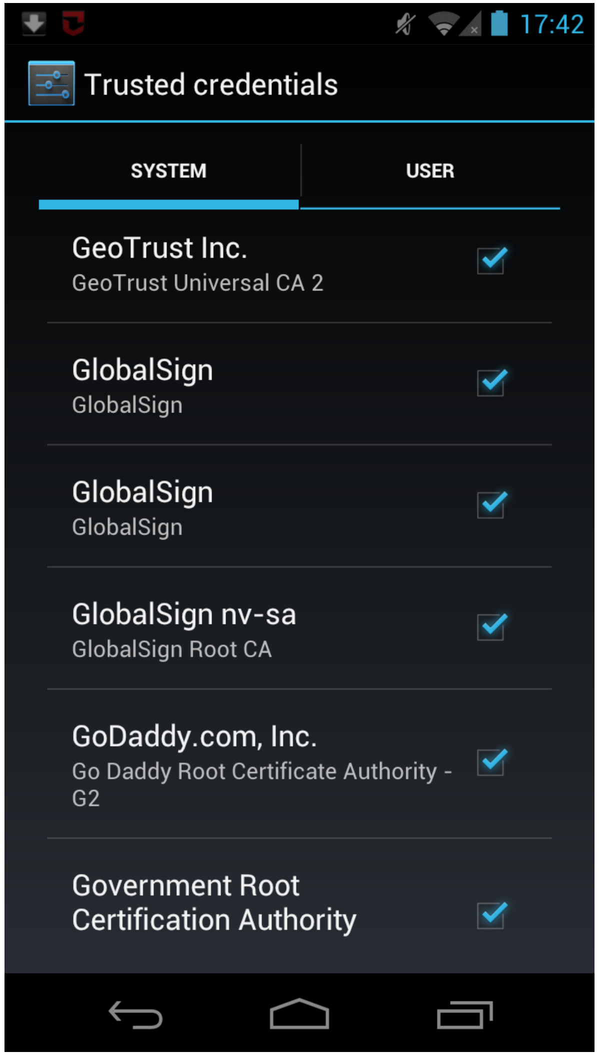
HTTPS Usage on Android

The default Android HTTPS API implements correct certificate validation.

What could possibly go wrong?

HTTPS Usage on Android and iOS

- A server needs a certificate that was signed by a trusted Certificate Authority
  - (~130 pre-installed CAs)
- For non-trusted certificates a custom workaround is needed
- Error handling requires custom code
- Additional security measures such as pinning or Certificate Transparency require custom code



17:42

Trusted credentials

GeoTrust Inc. ☑
GeoTrust Universal CA 2

GlobalSign ☑
GlobalSign

GlobalSign ☑
GlobalSign

GlobalSign nv-sa ☑
GlobalSign Root CA

GoDaddy.com, Inc. ☑
Go Daddy Root Certificate Authority - G2

Government Root Certification Authority ☑

But it does seem to go wrong...

Q: I am getting an error of „javax.net.ssl.SSLException: Not trusted server certificate“.

[...]

I have spent 40 hours researching and trying to figure out a workaround for this issue.

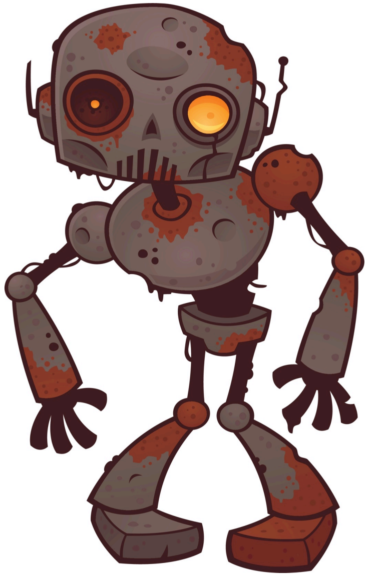
A: Look at this tutorial

http://blog.antoine.li/index.php/2010/10/android-trusting-ssl-certificates

stackoverflow.com

SSL Static Code Analysis

- Analysis of 13,500 popular, free apps from Google's Play Market
  - 92.8 % of the apps use the Internet permission
  - 91.7 % of networking API calls are HTTP(S) related
  - 0.8 % exclusively HTTPS URLs
  - 46.2 % mix HTTP and HTTPS
- 17.28 % of all apps that use HTTPS include code that fails in SSL certificate validation
  - 1070 include critical code
  - 790 accept all certificates
  - 284 accept all hostnames

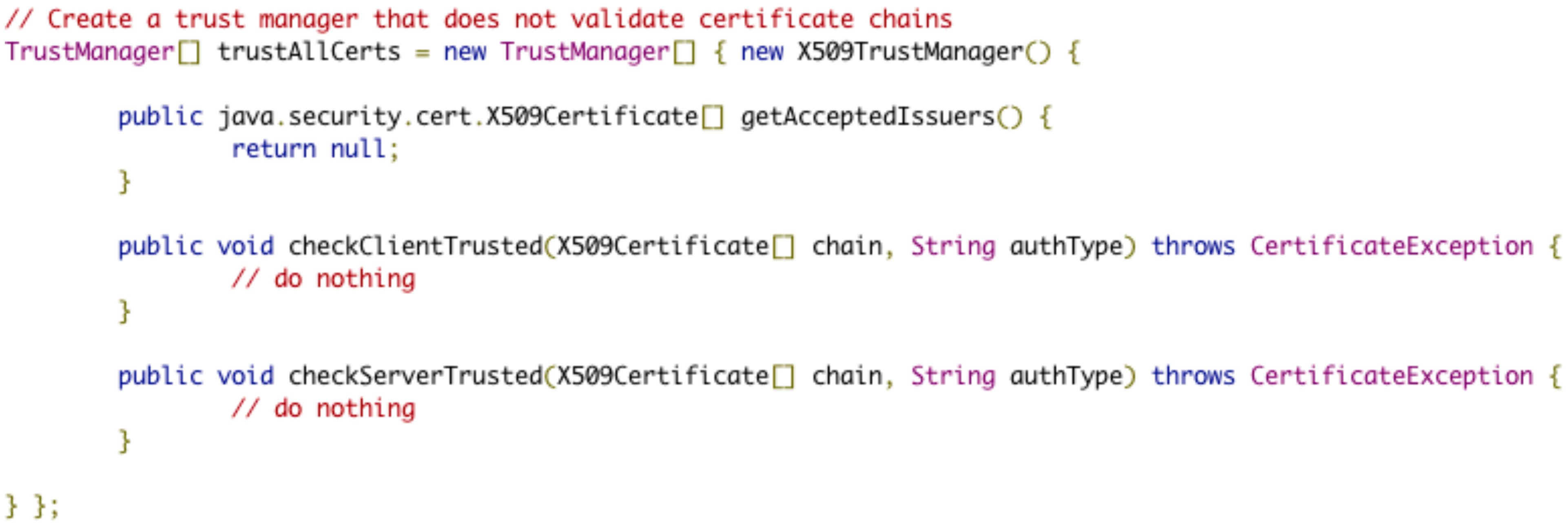


Manual App Testing Results

- Cherry-picked 100 apps
  - 21 apps trust all certificates
  - 20 apps accept all hostnames
- Captured credentials for:
  - American Express, Diners Club, Paypal, bank accounts, Facebook, Twitter, Google, Yahoo, Microsoft Live ID, Box, WordPress, remote control servers, arbitrary email accounts, and IBM Sametime, among others.

Trusting all Certificates

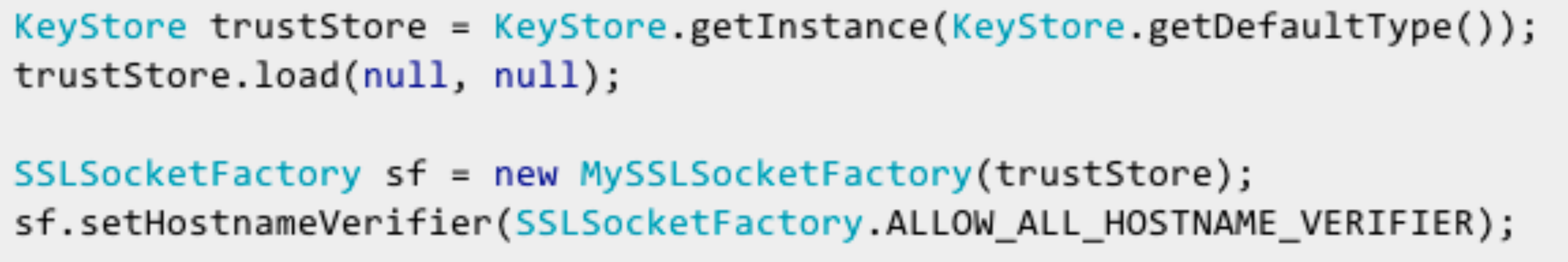
- Correct HTTPS certificate validation is easy
  - Only a (costly) trusted CA signed certificate required
- What some Apps do:



```java
// Create a trust manager that does not validate certificate chains
TrustManager[] trustAllCerts = new TrustManager[] { new X509TrustManager() {

        public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                return null;
        }

        public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException {
                // do nothing
        }

        public void checkServerTrusted(X509Certificate[] chain, String authType) throws CertificateException {
                // do nothing
        }

} };
```

Allowing all Hostnames

- What other Apps do:
  - Check CA signature, but allow mallory.com for google.com



```
KeyStore trustStore = KeyStore.getInstance(KeyStore.getDefaultType());
trustStore.load(null, null);

SSLSocketFactory sf = new MySSLSocketFactory(trustStore);
sf.setHostnameVerifier(SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER);
```

Anti-Virus Example

- ZonerAV
  - Anti-Virus app for Android
  - Awarded best free anti-virus app for Android by av-test.org
- Virus signature updates via HTTPS GET
  - The good thing: It uses SSL
  - Unfortunately: The wrong way

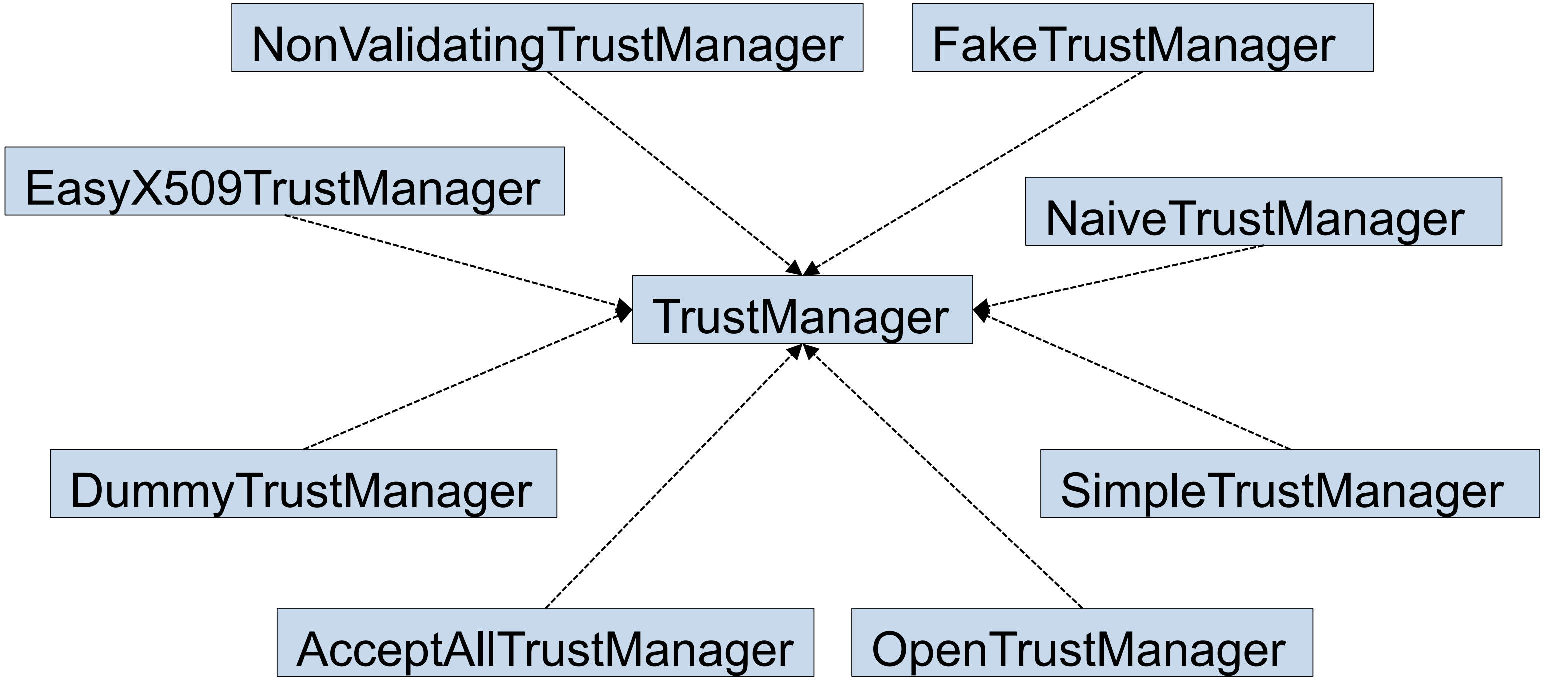
```
static final HostnameVerifier DO_NOT_VERIFY = new HostnameVerifier()
{
        public boolean verify(String paramString, SSLSession paramSSLSession)
        {
          return true;
        }
};
```

  - Zoner fixed the bug immediately!

Zoner AV

TrustManager Implementations

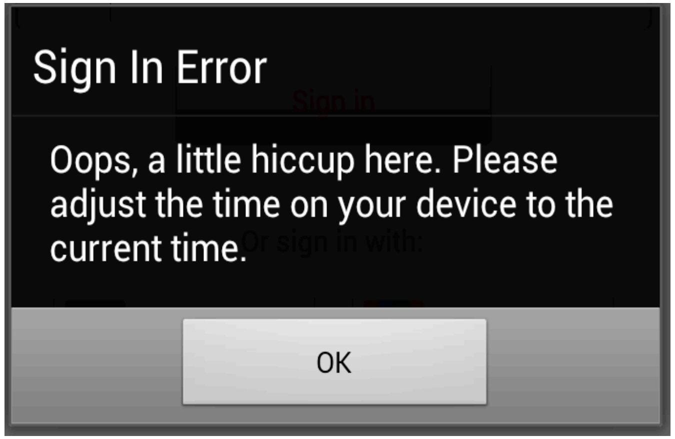
- 22 different TrustManager implementations



- and all turn effective certificate validation off

How Do (Good) Apps React to MITMAs?

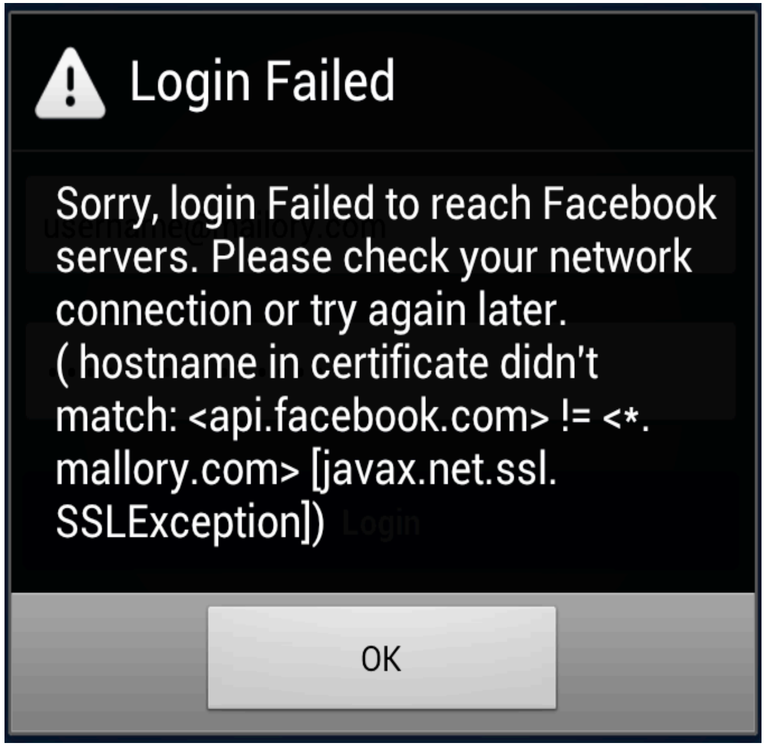
- Technically they do not endanger the user
- However they suffer from serious usability problems



Sign In Error

Oops, a little hiccup here. Please adjust the time on your device to the current time.

Flickr



Login Failed

Sorry, login Failed to reach Facebook servers. Please check your network connection or try again later. (hostname in certificate didn't match: <api.facebook.com> != <*. mallory.com> [javax.net.ssl. SSLException])

Facebook

Common: Blaming Developers

“It’s all the developers’ fault!”

Solutions?

So what should we do to help the developers?

Security experts need to communicate more with developers, and adopt developer-centered design approaches.

Talking To Developers

- Finding broken HTTPS in Android and iOS apps is good…

  …knowing what the root causes are is even better

- We contacted 80 developers of broken apps ✓
  - informed them ✓
  - offered further assistance ?
  - asked them for an interview

- 15 developers agreed ✓

Novice Developers

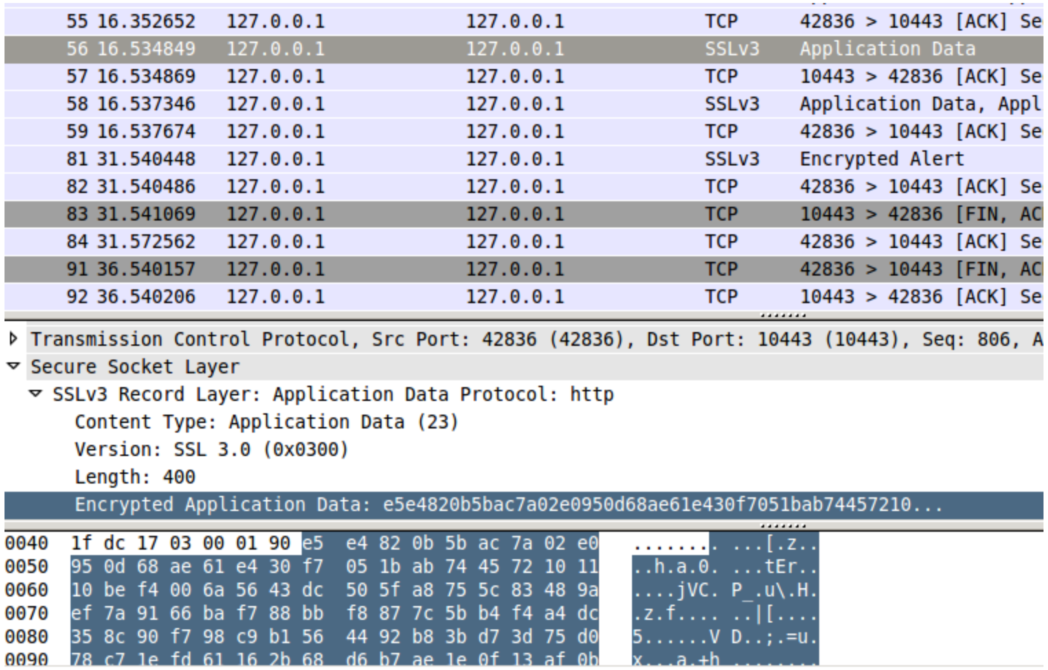
“This app was one of our first mobile apps and when we noticed that there were problems with the SSL certificate, we just implemented the first working solution we found on the Internet.”

Intermediate Developers

“We use self-signed certificates for testing purposes and the easiest way to make them working is to remove certificate validation. Somehow we must have forgotten to remove that code again when we released our app.“

Expert Developers (kind of...)

“[...] When I used Wireshark to look at the traffic, Wireshark said that this is a proper SSL protected data stream and I could not see any cleartext information when I manually inspected the packets. So I really cannot see what the problem is here.”



```
55 16.352652  127.0.0.1     127.0.0.1     TCP    42836 > 10443 [ACK] Se
56 16.534849  127.0.0.1     127.0.0.1     SSLv3  Application Data
57 16.534869  127.0.0.1     127.0.0.1     TCP    10443 > 42836 [ACK] Se
58 16.537346  127.0.0.1     127.0.0.1     SSLv3  Application Data, Appl
59 16.537674  127.0.0.1     127.0.0.1     TCP    42836 > 10443 [ACK] Se
81 31.540448  127.0.0.1     127.0.0.1     SSLv3  Encrypted Alert
82 31.540486  127.0.0.1     127.0.0.1     TCP    42836 > 10443 [ACK] Se
83 31.541069  127.0.0.1     127.0.0.1     TCP    10443 > 42836 [FIN, AC
84 31.572562  127.0.0.1     127.0.0.1     TCP    42836 > 10443 [ACK] Se
91 36.540157  127.0.0.1     127.0.0.1     TCP    42836 > 10443 [FIN, AC
92 36.540206  127.0.0.1     127.0.0.1     TCP    10443 > 42836 [ACK] Se

▷ Transmission Control Protocol, Src Port: 42836 (42836), Dst Port: 10443 (10443), Seq: 806, A
▽ Secure Socket Layer
  ▽ SSLv3 Record Layer: Application Data Protocol: http
      Content Type: Application Data (23)
      Version: SSL 3.0 (0x0300)
      Length: 400
      Encrypted Application Data: e5e4820b5bac7a02e0950d68ae61e430f7051bab74457210...

0040  1f dc 17 03 00 01 90 e5  e4 82 0b 5b ac 7a 02 e0   ........ ...[.z..
0050  95 0d 68 ae 61 e4 30 f7  05 1b ab 74 45 72 10 11   ..h.a.0. ...tEr..
0060  10 be f4 00 6a 56 43 dc  50 5f a8 75 5c 83 48 9a   ....jVC. P_.u\.H.
0070  ef 7a 91 66 ba f7 88 bb  f8 87 7c 5b b4 f4 a4 dc   .z.f.... ..|[....
0080  35 8c 90 f7 98 c9 b1 56  44 92 b8 3b d7 3d 75 d0   5......V D..;.=u.
0090  78 c7 1e fd 61 16 2b 68  d6 b7 ae 1e 0f 13 af 0b   x...a.+h ........
```

Expert Developers (time constrained)

“The app accepts all SSL certificates because some users wanted to connect to their blogs with self-signed certs and […] because Android does not provide an easy-to-use SSL certificate warning message, it was a lot easier to simply accept all self-signed certificates.”

vs.

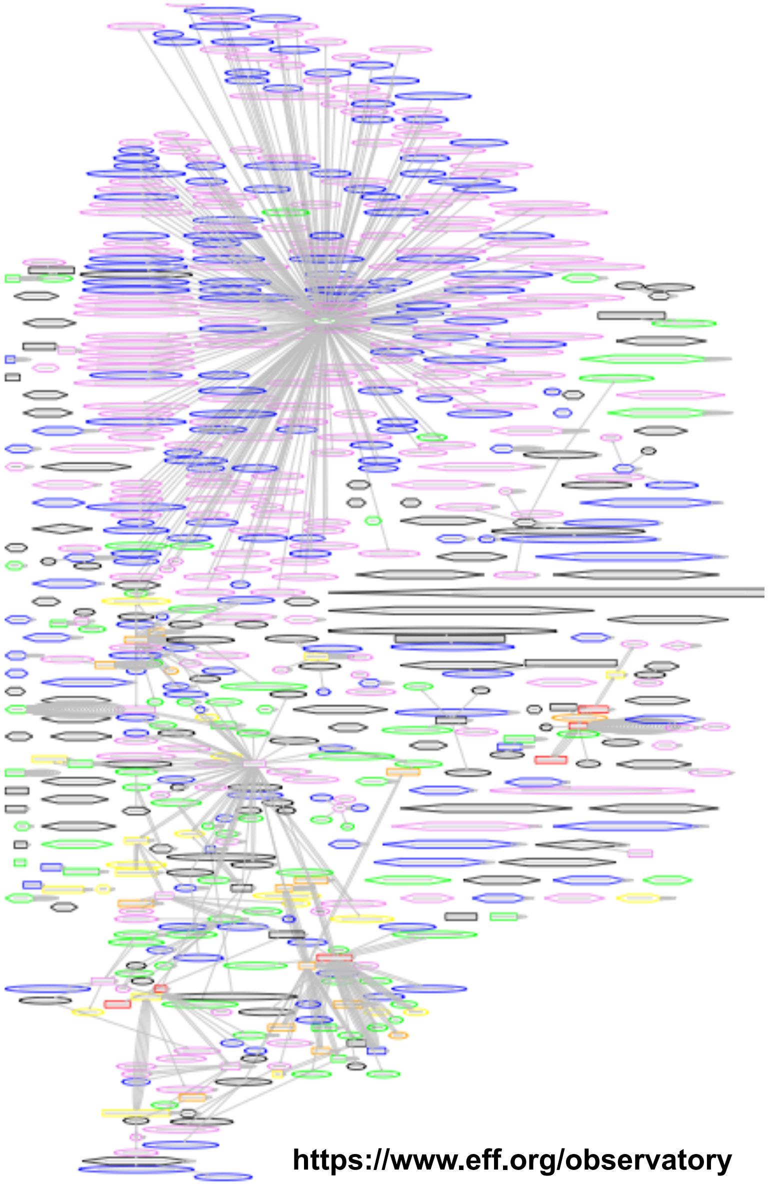
Developer Survey Summary

- Self-Signed Certificates – Development.
  - Developers commonly wish to use self-signed certificates for testing purposes and hence want to turn off certificate validation during testing.
- Self-Signed Certificates – Production.
  - A few developers wanted to use self-signed certificates in their production app for cost and effort reasons.
- Code Complexity.
  - Developers described the code-level customization features of HTTPS as too complex and requiring too much effort.
- Certificate Pinning / Trusted Roots.
  - Developers liked the idea of having an easy way to limit the number of trusted certificates and/or certificate authorities.
- Global Warning Message.
  - Developers requested global HTTPS warning messages since they described building their own warning messages as too challenging.

Chapter 3:
System Design

Problems with the infrastructure

- Approximately 100-200 trusted root CAs in
  - Firefox, Chrome, IE Explorer, Windows, Mac OS, Linux
  - Extended to ~650 via CA hierarchies
  - EFF Map of these organizations
- SSL / HTTPS only as strong as the weakest link
  - Weak (email-based) authentication with many CAs
  - Targeted attacks against CAs - a real world threat
  - No CA scopes



https://www.eff.org/observatory

Up-and-coming PKIs

- Up-and-coming PKIs
  - DANE
  - Certificate Transparency
  - ARPKI (Perrig et. al – next door at SENT)
- All promise better security
  - All are more complex
  - How will developers cope?
  - How will administrators cope?
  - How will users cope?

So what do we do now?

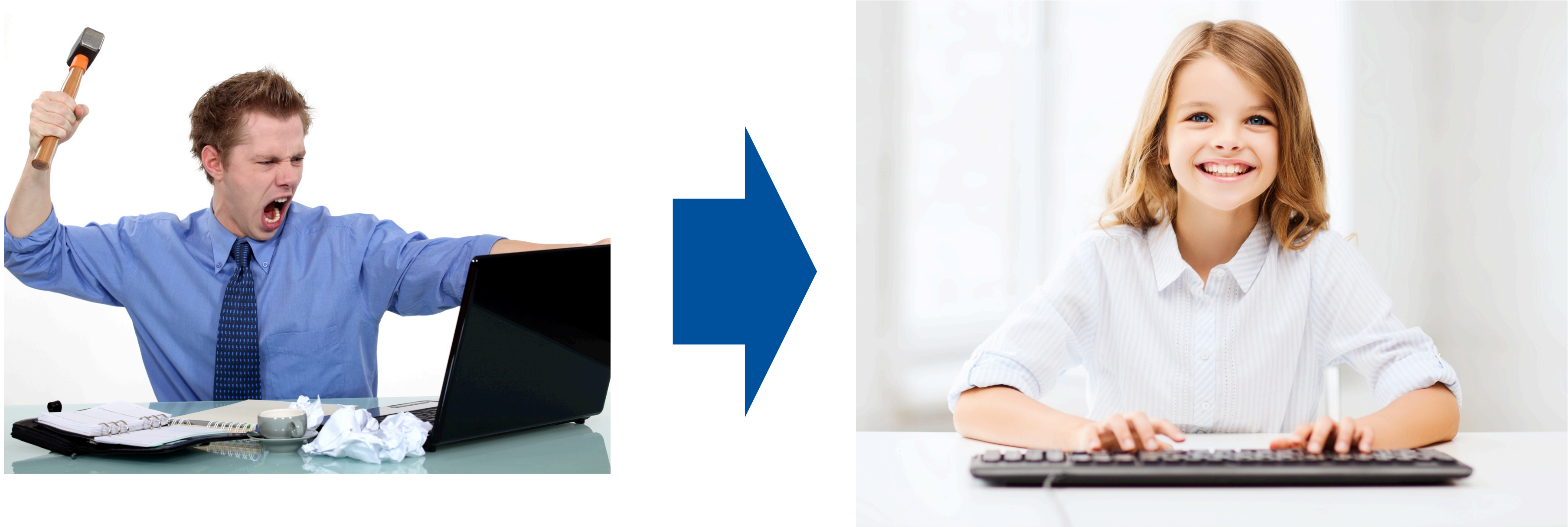
Frontiers of Usable Security

- Administrators and developers are humans too
  - We should be supporting them just as much – if not more – than end-users
  - Especially during systems design

- Short term goals:
  - Talk with administrators and developers
  - Find out where the problems lie
  - Extract and implement wish-lists

- Long term goal: Usable Systems Security
  - Design entire IT-Ecosystem with administrators and developers in mind

Experts Are Not The Enemy (either)



Let’s give them our support