universitätbonn

Developers are not the Enemy!

Matthew Smith

Usable Security and Privacy Lab, Universität Bonn, Fraunhofer FKIE

I‘d like to thank

universitätbonn

- Alena Naiakshina
- Anastasia Danilova
- Christian Tiefenau
- Dr. Emanuel von Zezschwitz
- Dr. Henning Perl
- Karoline Busse
- Dr. Khaled Yakdan
- Dr. Marian Harbach
- Dr. Michael Brenner
- Dr. Sascha Fahl
- Sergej Dechand
- Yasemin Acar

Usable Security?

universitätbonn

Security is hard!

universitätbonn

The goal of Usable Security is to make it easy!

universitätbonn

Users are not the enemy!

Adams & Sasse’99

Overview

universitätbonn

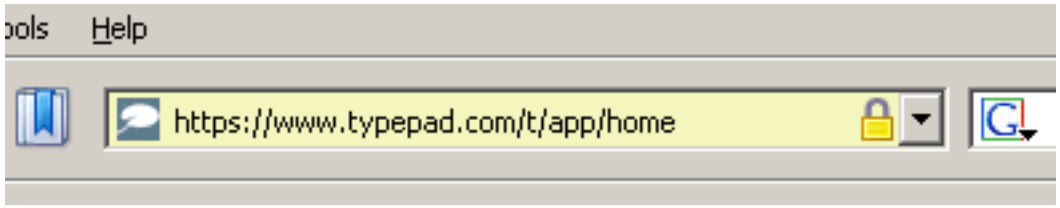
- Story 1: HTTPS
- Story 2: Passwords
- Story 3: Malware Analysis
- or Frontiers of Usable Security Methodology

Story 1
HTTPS/TLS

universitätbonn

HTTPS Part 1: Security Indicators

HTTPS Indicators (old)

- Microsoft IE 
- Mozilla 
- Firefox 

- Safari 

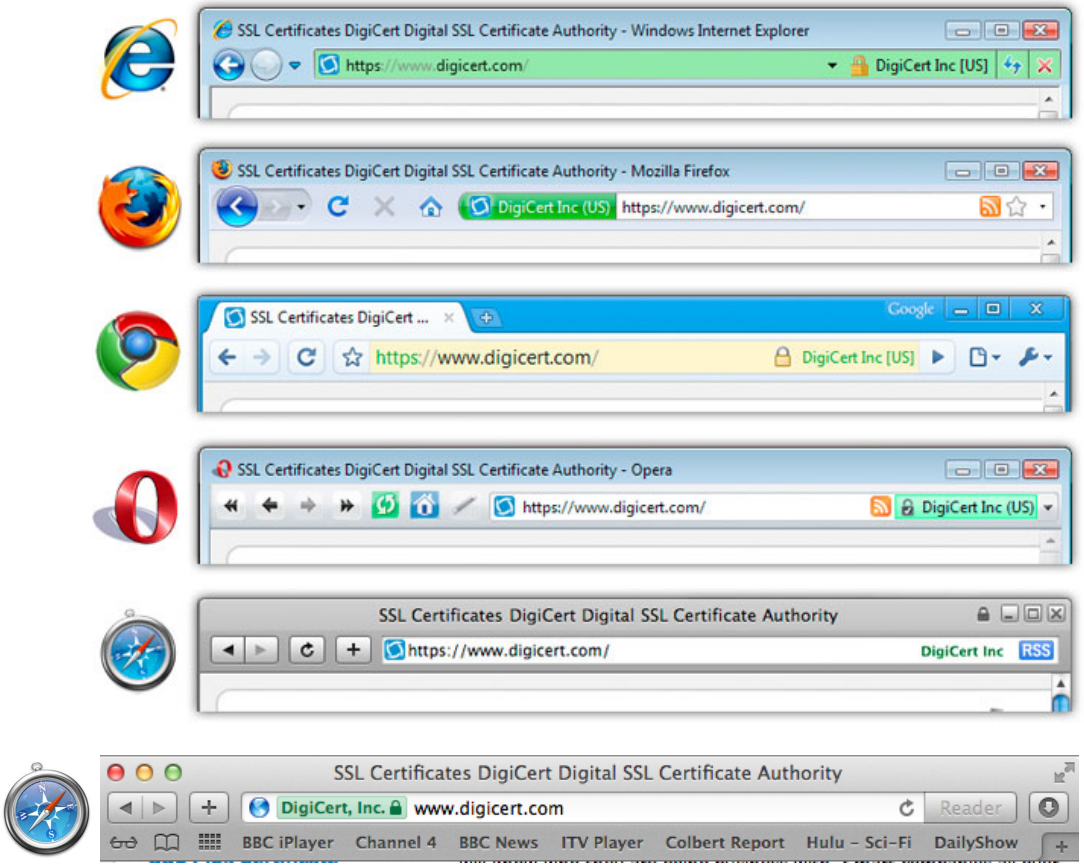
User Study

- Schechter et al. conducted a lab study with 67 participants*
  - Complete an online banking task
  - Three groups
    - Role playing
    - Role playing with hint to behave securely
    - Users' real online banking account
- Removed HTTPS security indicator
  - 100% entered their credentials
  - Even those using their real online banking credentials

* Schechter et al., The Emperor's New Security Indicators An evaluation of website authentication and the effect of role playing on usability studies, IEEE Security and Privacy 2007

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



universitätbonn

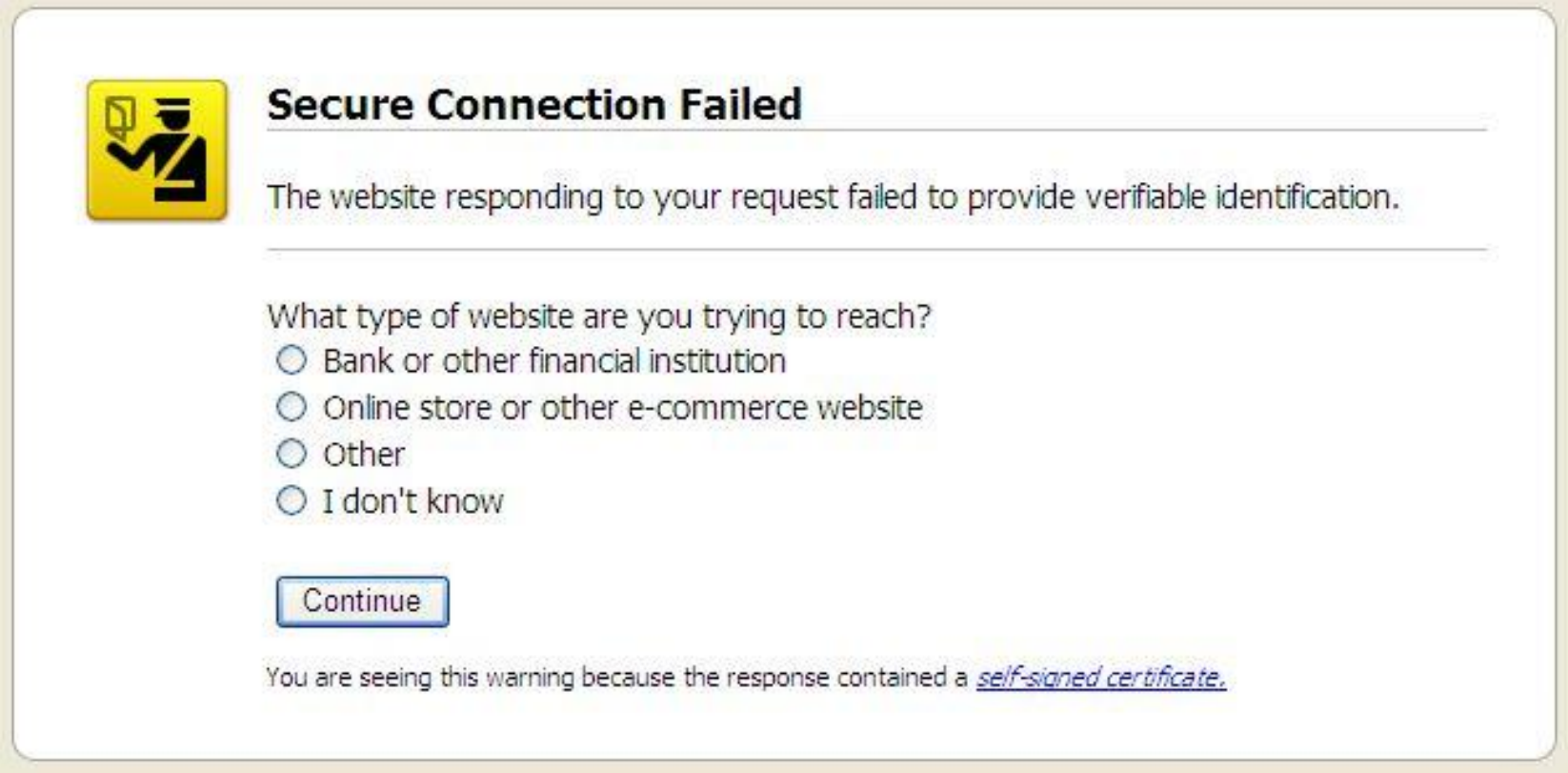
HTTPS Part 2: Security Warnings

Firefox 2 Warning

What users actually see

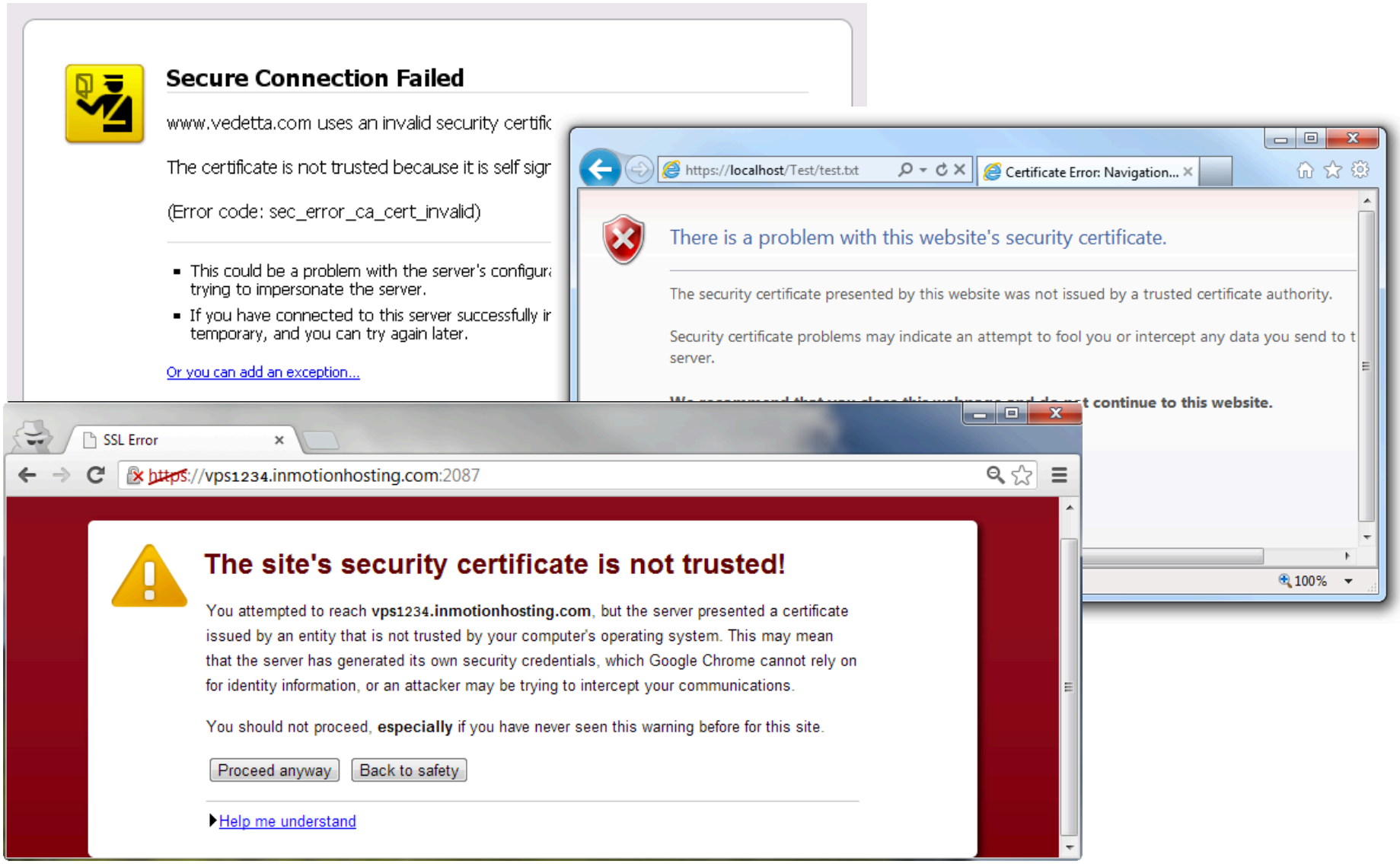
Adapted from Jonathan Nightingale

USEC Warnings



Sunshine et. al. Crying Wolf, Usenix Security 2009

Newer HTTPS Warnings

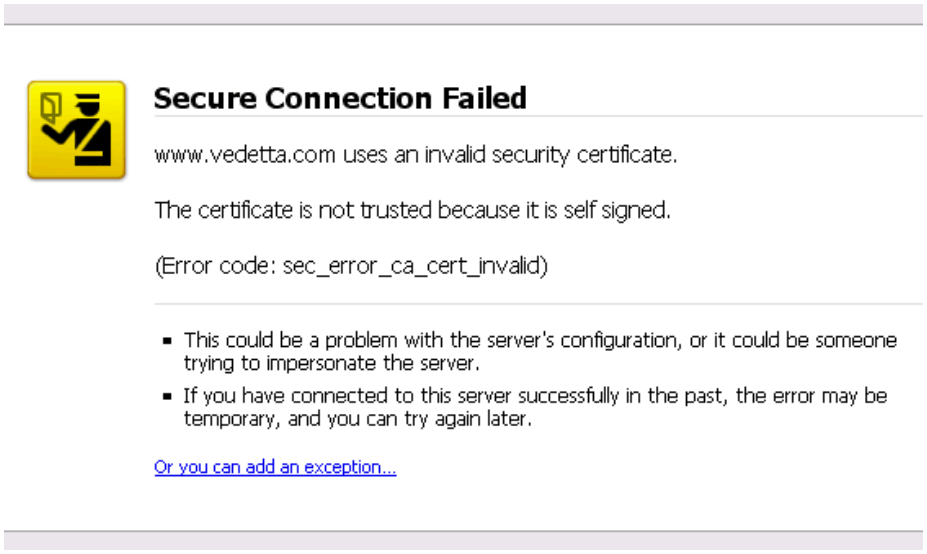


HTTPS: Administrator Mistakes

Akhawe et al: Server misconfigurations lead to

15.400 false positive per **1** true positive

certificate warnings[1]



[1] Akhawe et al. (WWW '13)

HTTPS: Administrator Mistakes

Akhawe et al: Server misconfigurations lead to

15.400 false positive per **1** true positive certificate warnings[1]

15.400 to 1 odds shouldn't be dealt with on the end-user level
but on the system level

[1] Akhawe et al. (WWW '13)

Developers are not the enemy!

Green & Smith IEEE S&P Magazine'16

Frontiers of Usable Security

- End-users are only a small part of the HTTPS ecosystem
- Administrators are responsible for (mis)configuration of 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...

universitätbonn

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%)

USEC Studies with Administrators

universitätbonn

- ~610k 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

- 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, etc.
- Better Tool Support
  - OpenSSL command line tool too complicated
  - Server configuration cumbersome, especially for v-hosts
  - Auto-Update Reminder
  - Notification of problems

Published at ACM AsiaCCS'14


Let's
Encrypt

Sneak Peak

- Study with 32 computer science students

CA	Success	Fail
CA-C	28	4
CA-T	16	16

universitätbonn

Developers

HTTPS Usage on Android

The default Android HTTPS API implements correct certificate validation.

What could possibly go wrong?

Trust me! I know what I‘m doing!

And it does 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

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:

```
// 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
        }

} };
```

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

Common: Blaming Developers

"It's all the developers' fault!"

Solutions?

universitätbonn

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

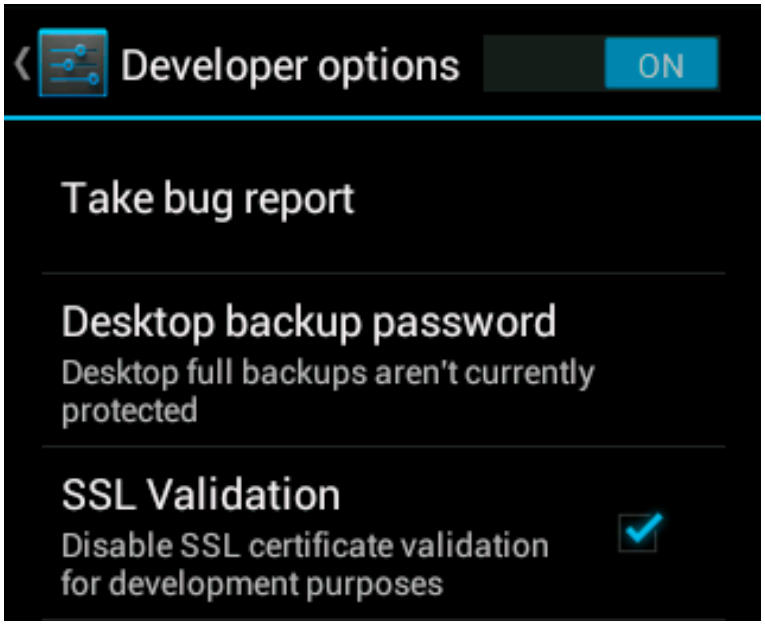
- We contacted 80 developers of broken apps
  - informed them ✓
  - offered further assistance ✓
  - asked them for an interview ?

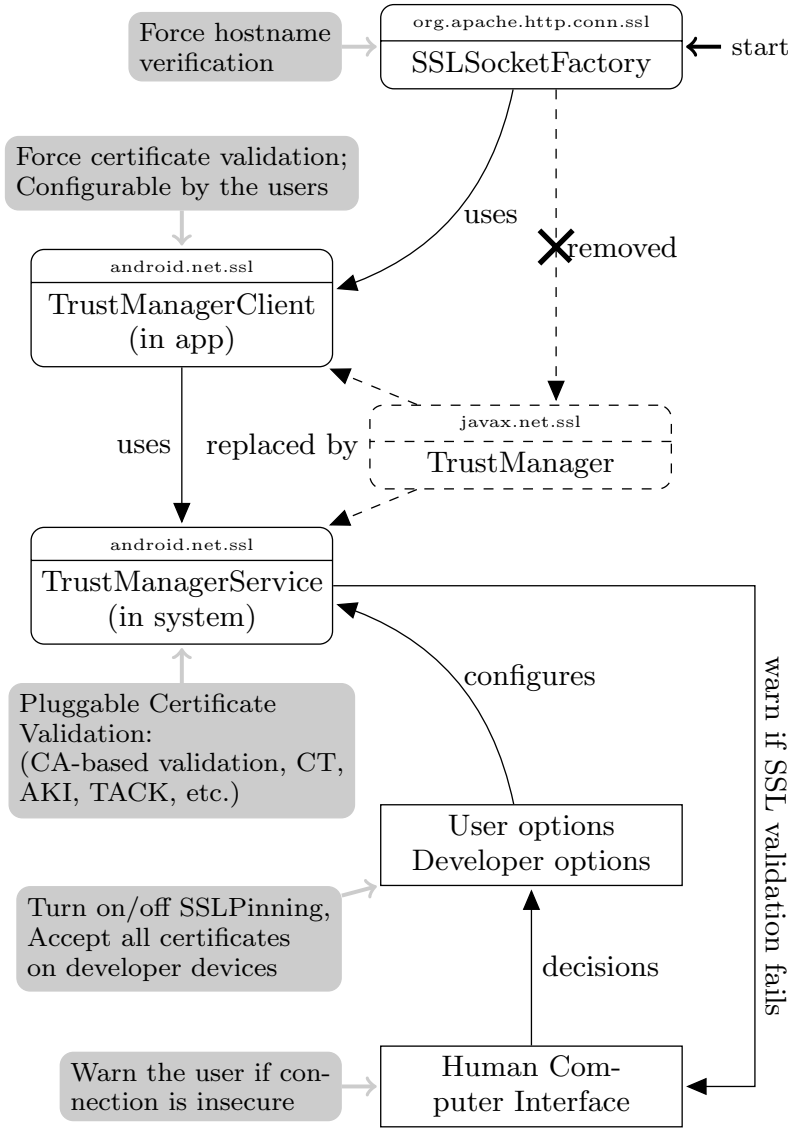
- 15 developers agreed ✓

A New Approach to TLS on Android

Central TLS service for Android

- Force TLS validation
- Supports self-signed certificates
- Certificate Pinning
- Standardised user interaction
- Alternate Cert validation strategies



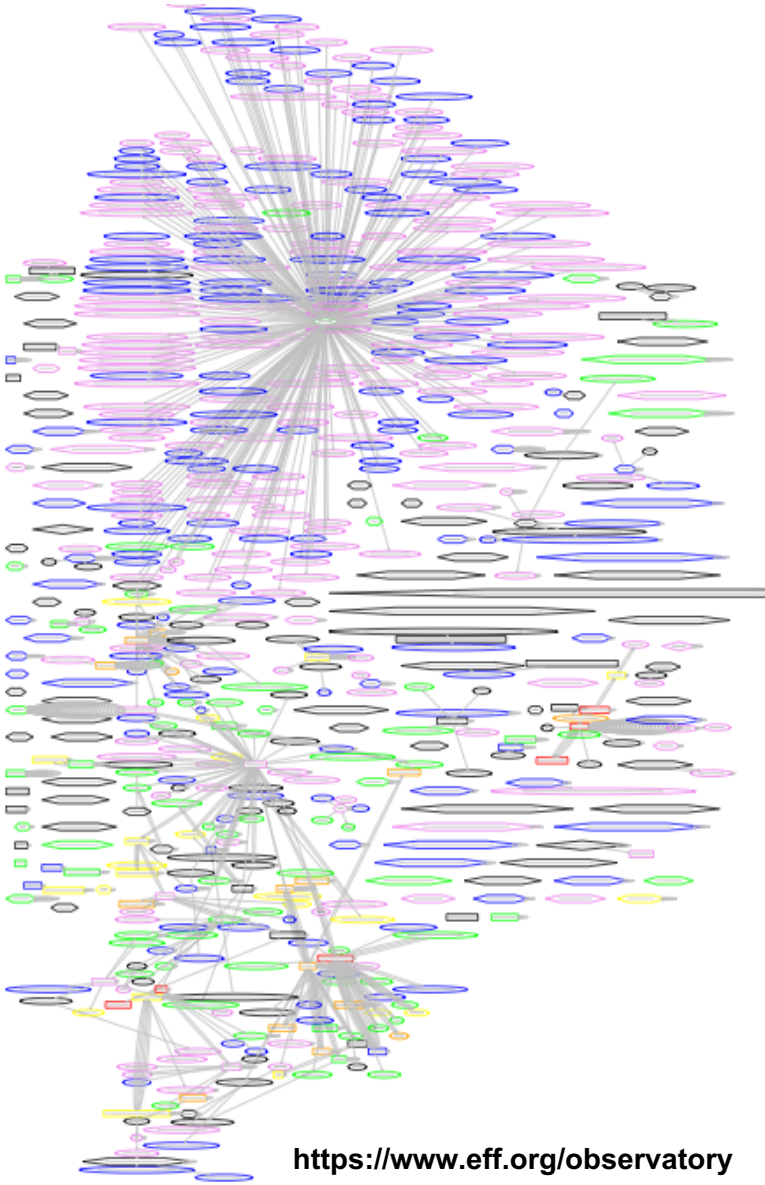


universitätbonn

CA Infrastructure

Problems with the CA 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

universitätbonn

- Up-and-coming PKIs
  - DANE
  - Certificate Transparency (Google)
  - ARPKI/SCION (ETH Zürich)

- All offer better security
  - All are more complex
  - How will developers cope?
  - How will administrators cope?
  - How will users cope?

Story 2

Passwords

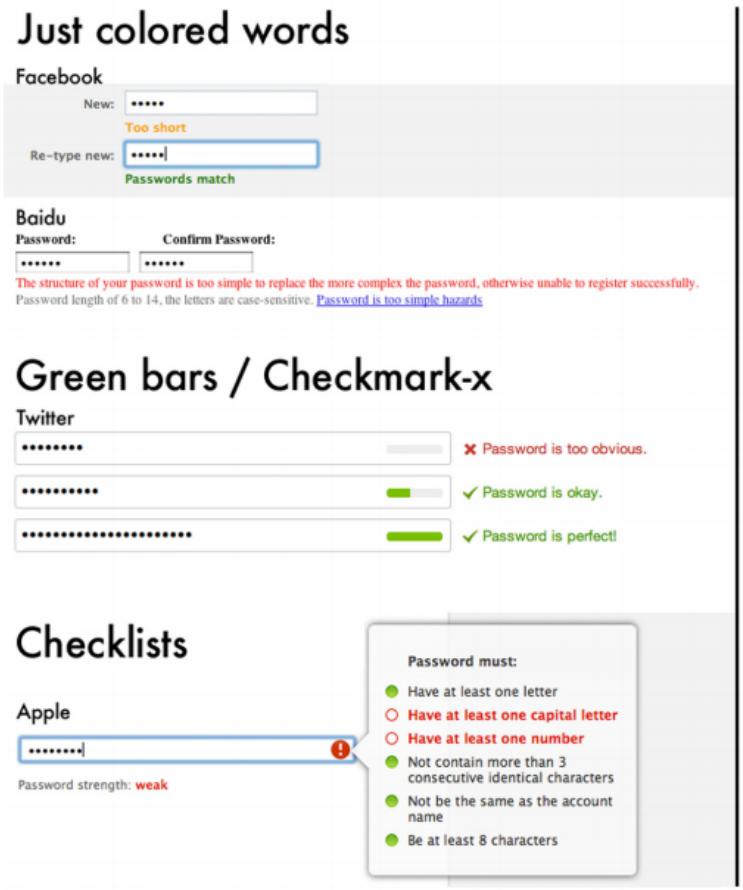
Password Advice

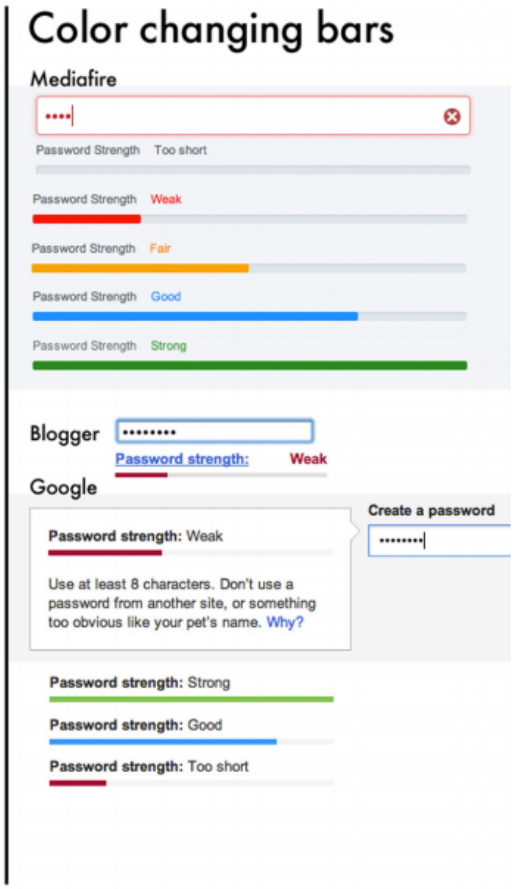
- Passwords are still a mainstay of modern security
  - and a very common cause of security problems
- Common password advice
  - make it long and random
  - use special characters
  - don't write it down
  - change it often
  - don't re-use across services
- Password problems lead to
  - lost productivity
  - recovery cost
  - frustrated users who try and circumvent system

good technical advice

bad usability advice

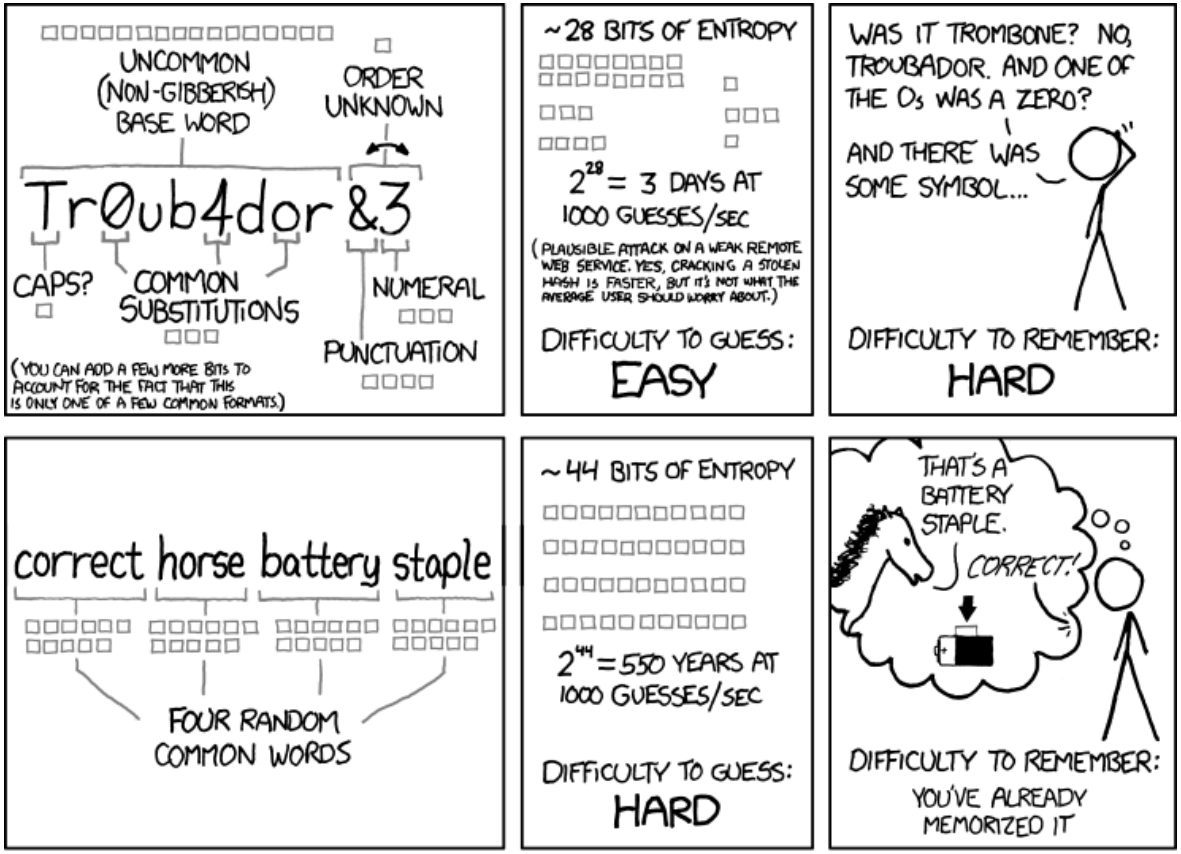
Password Meters





Ur et al. How Does Your Password Measure Up?
The Effect of Strength Meters on Password Creation, USENIX Security'12

Passphrases



Shay et al. Correct horse battery staple: Exploring the usability of system-assigned passphrases, SOUPS‘12

universitätbonn

THE VERGE TECH SCIENCE CULTURE CARS REVIEWS LONGFORM VIDEO MORE

US & WORLD | TECH | CYBERSECURITY

Yahoo says all 3 billion user accounts were impacted by 2013 security breach

by Natt Garun | @nattgarun | Oct 3, 2017, 5:07pm EDT

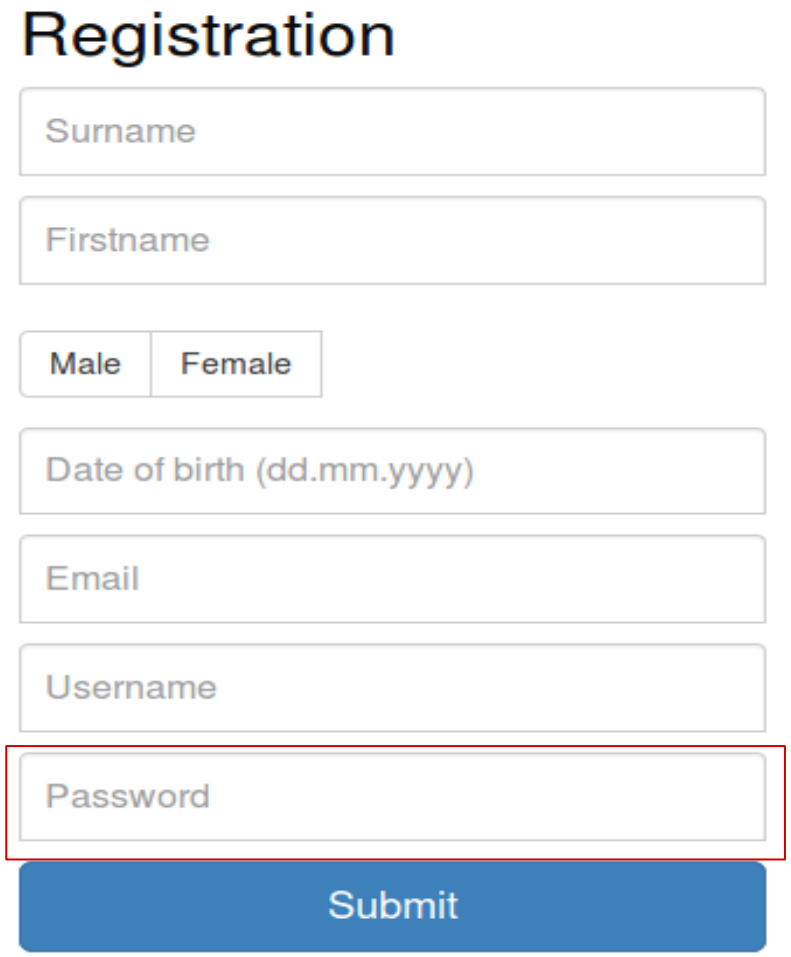
Study Design

Role-playing scenario

- Social networking platform of the University of Bonn.
- Code for user registration and user authentication.

Two Groups

- Secure password storage
- API usability.



Frameworks

universitätbonn

JSF

- Manual level of support
- No built-in functions for hashing

Spring

- Opt-in support
- Built-in functions for hashing

4 Conditions

universitätbonn

	Framework	Level of Support	(Non-)Priming	Label
1	JSF	manual	Priming	JP
2	JSF	manual	Non-Priming	JN
3	Spring	opt-in	Priming	SP
4	Spring	opt-in	Non-Priming	SN

Participant Demographics

- 20 participants
  - 3 female, 17 male
- Students: 18 Computer Science, 2 Media Informatics
  - 7 BSc, 13 MSc Students
- Mean age 24 years
  - Range: 19-27 years

- 8 hours to complete study
- Post study interview

Security Score

universitätbonn

- The end-user password is **salted** *(+1)* and **hashed** *(+1)*.
- The **derived length of the hash** is **at least 160 bits** long *(+1)*.
- The **iteration count** for key stretching is
  - at least 1 000 *(+0.5)* or 10 000 *(+1)* for **PBKDF2** and
  - at least 2^{10} = 1 024 for **bcrypt** *(+1)*.
- A **memory-hard hashing function** is used *(+1)*.
- The **salt** value is generated **randomly** *(+1)*.
- The **salt** is **at least 32 bits** in length *(+1)*.

Results

Non-Primed Group

universitätbonn

How many participants had a basic background knowledge of **hashing**?

9/10

How many participants have

managed to store

the user **passwords securely**?

Task not Security Related

Umm, actually literally when I was in the project
***I didn't feel** much like that*
*it was **related to security**. (JN5)*

Responsibility

I would ask my supervisor about it. [...]
*There is definitely **another person***
that understood these kinds of things. (JN3)

Misconceptions

I assumed that the connection
will be a secure connection
like with an HTTPS connection,
so everything should come encrypted. (JN1)

universitätbonn

Developers are not the enemy!

Primed Group

universitätbonn

How many participants had a basic background knowledge of **hashing**?

9/10

universitätbonn

How many participants have

managed to store

the user **passwords securely**?

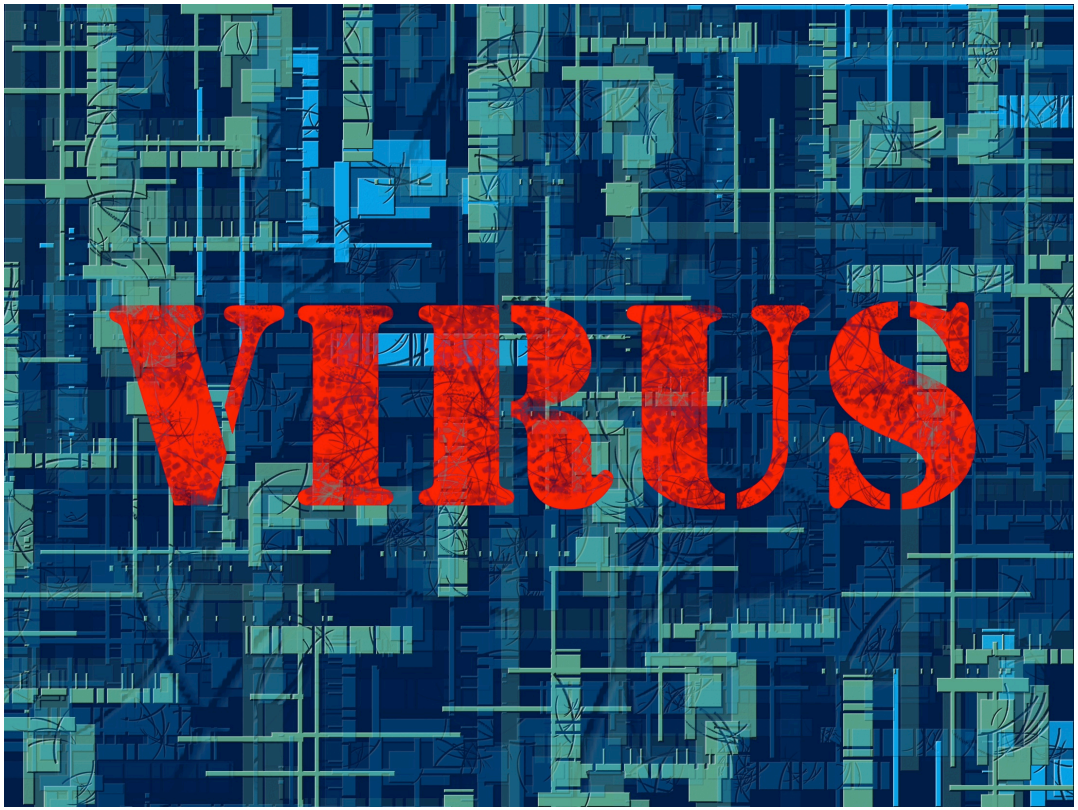
Primed Group

- ➔ 7/10 included at least some security
- ➔ 4/10 participants received 6 points.
- ➔ 3/4 were in the Spring group

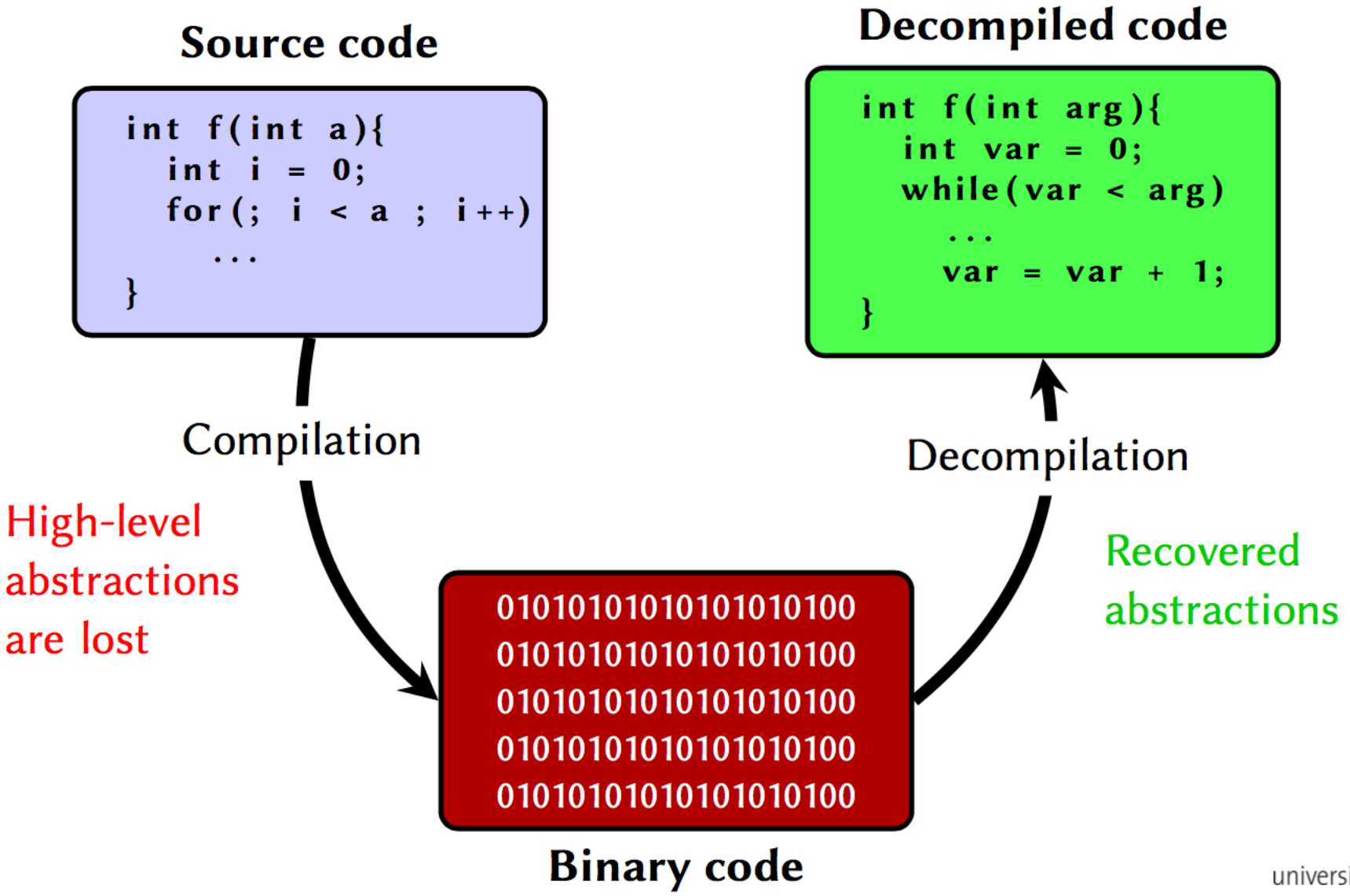
	Hash Function	Sec	Func
JP1	-	0	Y
JP2	PBKDF2(SHA256)	5.5	Y
JP3	SHA256	2	Y
JP4	PBKDF2(SHA1)	6	Y
JP5	-	0	Y
SP1	BCrypt	6	Y
SP2	MD5	1	Y
SP3	BCrypt	6	N
SP4	BCrypt	6	Y
SP5	-	0	N

Story 3

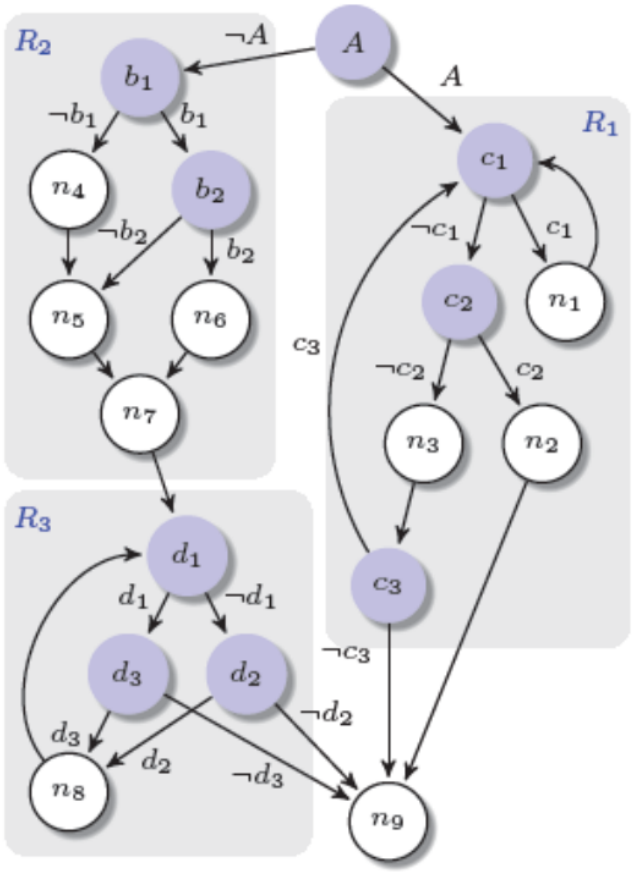
Security Analysis

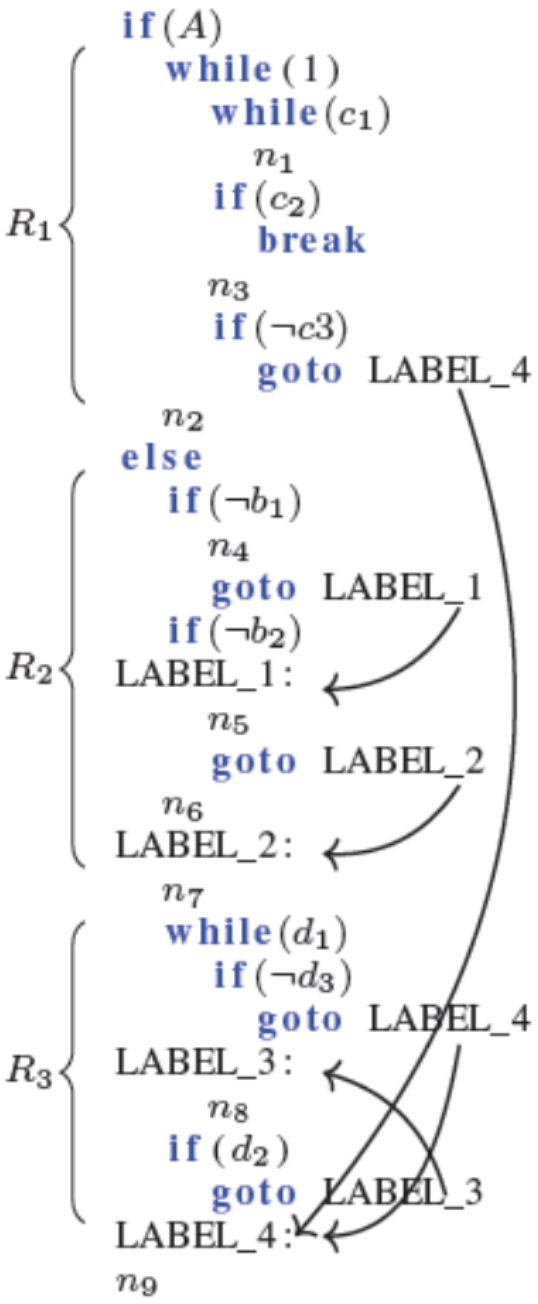


Malware Analysis



Control Flow Graph

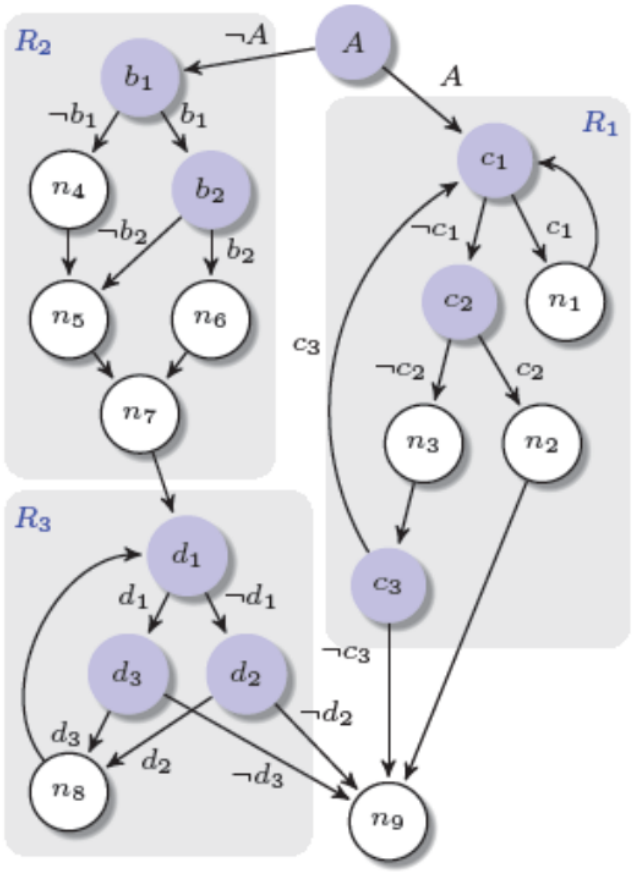




Decompiling a P2P Zeus sample with Hex-Rays

- 1,571 goto for 49,514 LoC
- 1 goto for each 32 LoC

Control Flow Graph




```
if (A)
  do
    while (c1)
      n1
      if (c2)
        n2
        break
    n3
  while (c3)
else
  if (¬b1)
    n4
  if (b1 ∧ b2)
    n6
  else
    n5
  n7
  while ((d1 ∧ d3) ∨ (¬d1 ∧ d2))
    n8
n9
```

- DREAM Decompiler
  - Pattern independent CFG structuring
  - No more gotos!
  - Most compact code

Usability Problems

universitätbonn

- Complex expressions
  - *(too) Many variables*
  - *Code in* loop statements
  - *Pointer expressions*
- Control Flow
  - *Duplicate/inlined code*
  - *Complex loop structure*
- No Semantics
  - Special API function
  - Magic number of file types

Simda Malware

universitätbonn

```
void *__cdecl sub_10006390(){
  __int32 v13; // eax@14
  int v14; // esi@15
  unsigned int v15; // ecx@15
  int v16; // edx@16
  char *v17; // edi@18
  bool v18; // zf@18
  unsigned int v19; // edx@18
  char v20; // dl@21
  char v23; // [sp+0h] [bp-338h]@1
  int v30; // [sp+30Ch] [bp-2Ch]@1
  __int32 v36; // [sp+324h] [bp-14h]@14
  int v37; // [sp+328h] [bp-10h]@1
  int i; // [sp+330h] [bp-8h]@1
  // [...]
  v30 = *"qwrtpsdfghjklzxcvbnm";
  v32 = *"ghjklzxcvbnm";
  v33 = *"lzxcvbnm";
  v31 = *"psdfghjklzxcvbnm";
  v35 = aQwrtpsdfghjklz[20];
  v37 = *"eyuioa";
  v34 = *"vbnm";
  v38 = *"oa";
  v39 = aEyuioa[6];
  // [...]
  v14 = 0;
  v15 = 3;
  if ( v13 > 0 )
  {
    v16 = 1 - &v23;
    for ( i = 1 - &v23; ; v16 = i )
    {
      v17 = &v23 + v14;
      v19 = (&v23 + v14 + v16) & 0x80000001;
      v18 = v19 == 0;
      if ( (v19 & 0x80000000) != 0 )
        v18 = ((v19 - 1) | 0xFFFFFFFE) == -1;
      v20 = v18 ? *(&v37 + dwSeed / v15 % 6)
                : *(&v30 + dwSeed / v15 % 0x14);
++v14;
v15 += 2;
*v17 = v20;
if ( v14 >= v36 )
break; }
}
```

```
LPVOID sub_10006390(){
  char * v1 = "qwrtpsdfghjklzxcvbnm";
  char * v2 = "eyuioa";
  // [...]
  int v13 = 3;
  for(int i = 0; i < num; i++){
    char v14 = i % 2 == 0 ? v1[(dwSeed / v13) % 20]
                          : v2[(dwSeed / v13) % 6];
v13 += 2;
v3[i] = v14;
}
```

DREAM++

dwSeed = 0x45AE94B2
results in?

User Study

- We recruited 21 students who successfully took part in our malware bootcamp over the last 5 years and
- 9 malware analysis professionals

- $3 \times 2 \times 2$ mixed-subjects design
- 3 decompilers (within-subjects)
  - Hex-Rays
  - DREAM
  - DREAM++
- 2 levels of experience (between-subject)
  - Students and Professionals
- 2 groups of malware analysis tasks (split-plot)
  - 3 medium and 3 hard task (within-subjects)

Simda Malware

universitätbonn

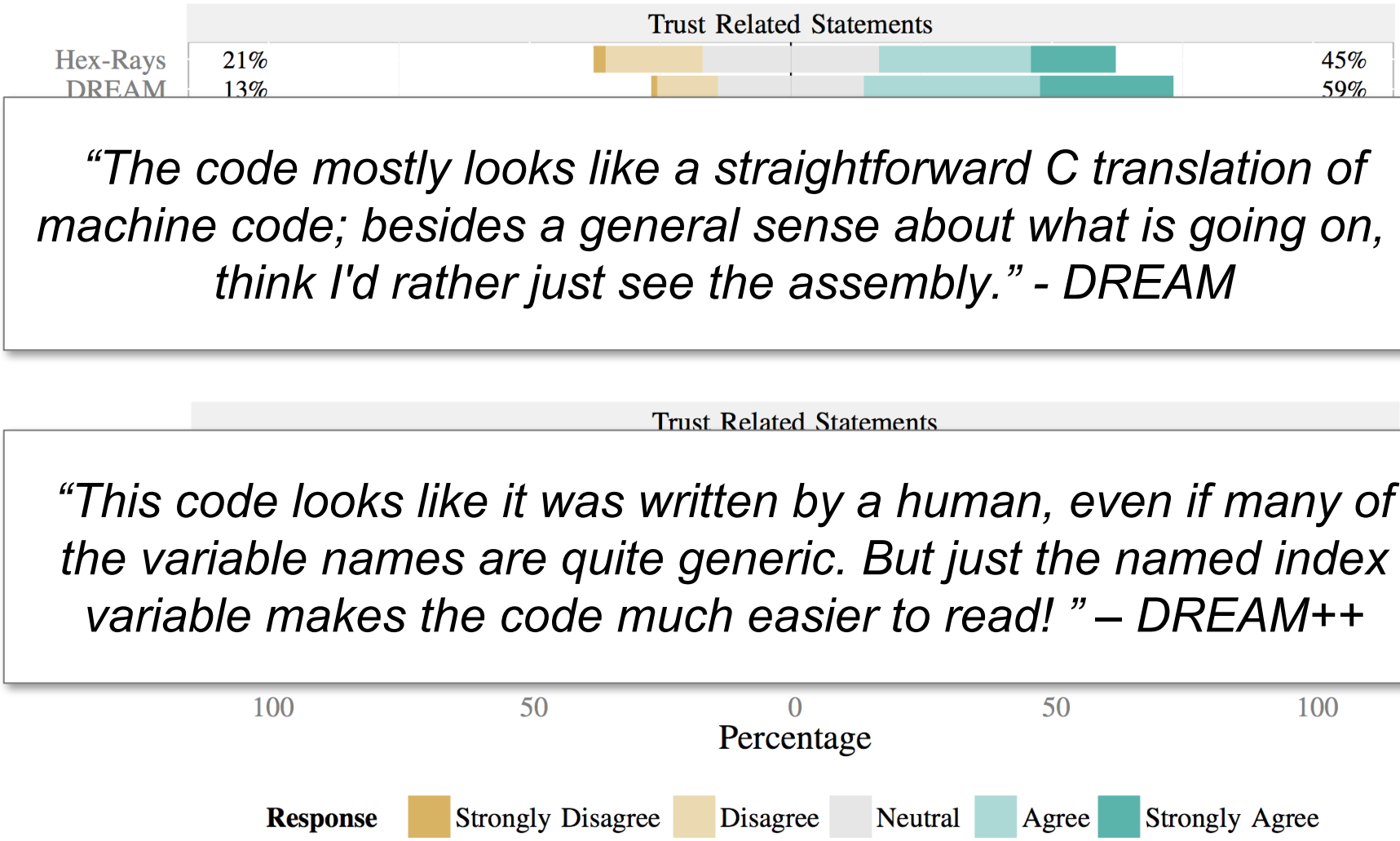
```
void *__cdecl sub_10006390(){
  __int32 v13; // eax@14
  int v14; // esi@15
  unsigned int v15; // ecx@15
  int v16; // edx@16
  char *v17; // edi@18
  bool v18; // zf@18
  unsigned int v19; // edx@18
  char v20; // dl@21
  char v23; // [sp+0h] [bp-338h]@1
  int v30; // [sp+30Ch] [bp-2Ch]@1
  __int32 v36; // [sp+324h] [bp-14h]@14
  int v37; // [sp+328h] [bp-10h]@1
  int i; // [sp+330h] [bp-8h]@1
  // [...]
  v30 = *"qwrtpsdfghjklzxcvbnm";
  v32 = *"ghjklzxcvbnm";
  v33 = *"lzxcvbnm";
  v31 = *"psdfghjklzxcvbnm";
  v35 = aQwrtpsdfghjklz[20];
  v37 = *"eyuioa";
  v34 = *"vbnm";
  v38 = *"oa";
  v39 = aEyuioa[6];
  // [...]
  v14 = 0;
  v15 = 3;
  if ( v13 > 0 )
  {
    v16 = 1 - &v23;
    for ( i = 1 - &v23; ; v16 = i )
    {
      v17 = &v23 + v14;
      v19 = (&v23 + v14 + v16) & 0x80000001;
      v18 = v19 == 0;
      if ( (v19 & 0x80000000) != 0 )
        v18 = ((v19 - 1) | 0xFFFFFFFE) == -1;
      v20 = v18 ? *(&v37 + dwSeed / v15 % 6)
                : *(&v30 + dwSeed / v15 % 0x14);
++v14;
v15 += 2;
*v17 = v20;
if ( v14 >= v36 )
break; }
}
```

```
LPVOID sub_10006390(){
  char * v1 = "qwrtpsdfghjklzxcvbnm";
  char * v2 = "eyuioa";
  // [...]
  int v13 = 3;
  for(int i = 0; i < num; i++){
    char v14 = i % 2 == 0 ? v1[(dwSeed / v13) % 20]
                          : v2[(dwSeed / v13) % 6];
v13 += 2;
v3[i] = v14;
}
```

DREAM++

dwSeed = 0x45AE94B2
results in?
v17 = "cihunemyror"
v3 = "cihunemyror"

Results



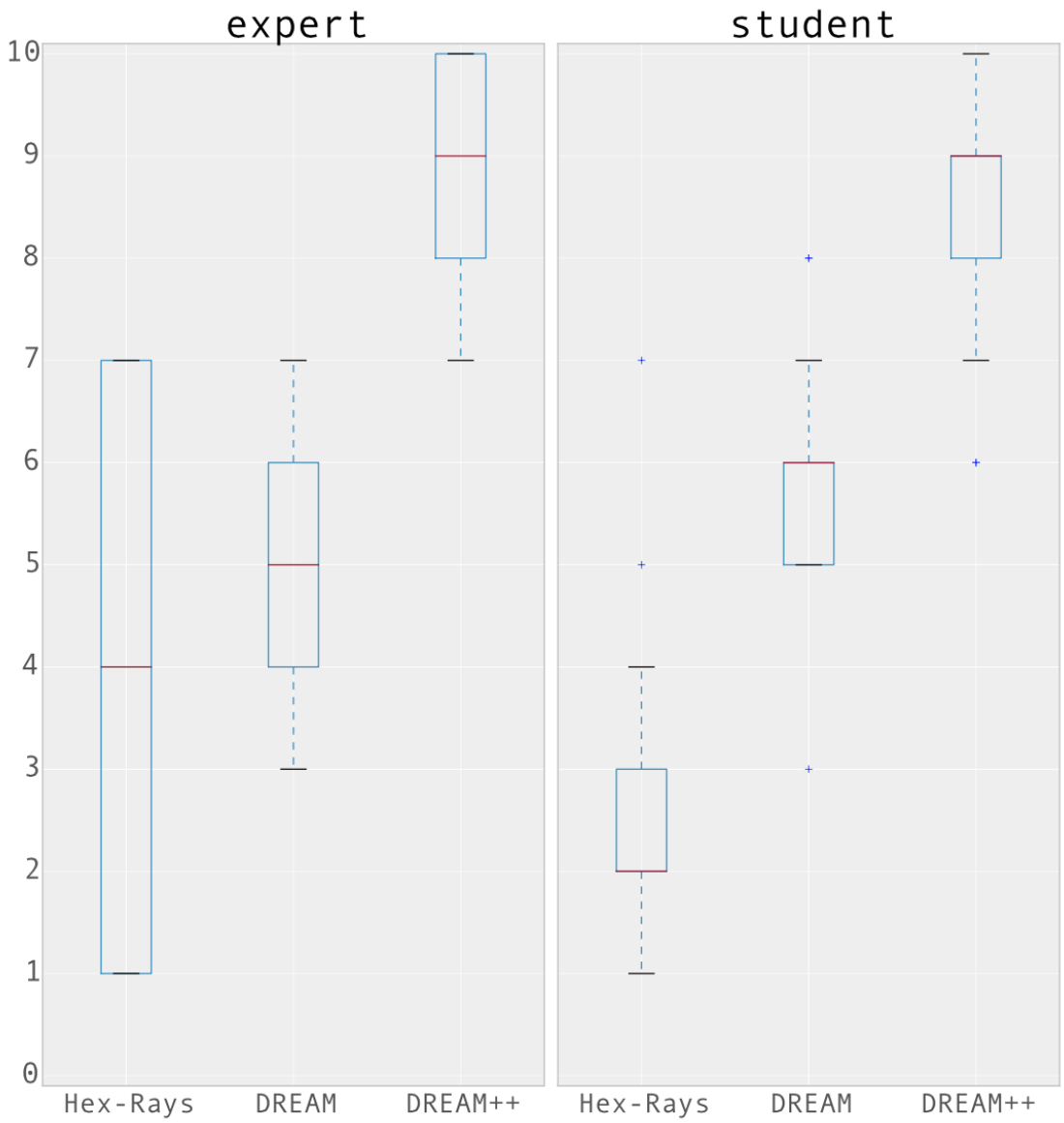
"The code mostly looks like a straightforward C translation of machine code; besides a general sense about what is going on, I think I'd rather just see the assembly." - DREAM

"This code looks like it was written by a human, even if many of the variable names are quite generic. But just the named index variable makes the code much easier to read! " – DREAM++

Results

Decompiler	Avg. Score	p	Pass	Fail	p
Students					
DREAM^{++}	70.24		30	12	
DREAM	50.83	0.002	16	26	0.002
Hex-Rays	37.86	<0.001	11	31	<0.001
Experts					
DREAM^{++}	84.72		15	3	
DREAM	79.17	0.234	15	3	0.570
Hex-Rays	61.39	0.086	9	9	0.076

Ranking Results



Usable Vulnerability Analysis

code intelligence

USECFrontiers

universitätbonn

Frontiers of Usable Security

ERC

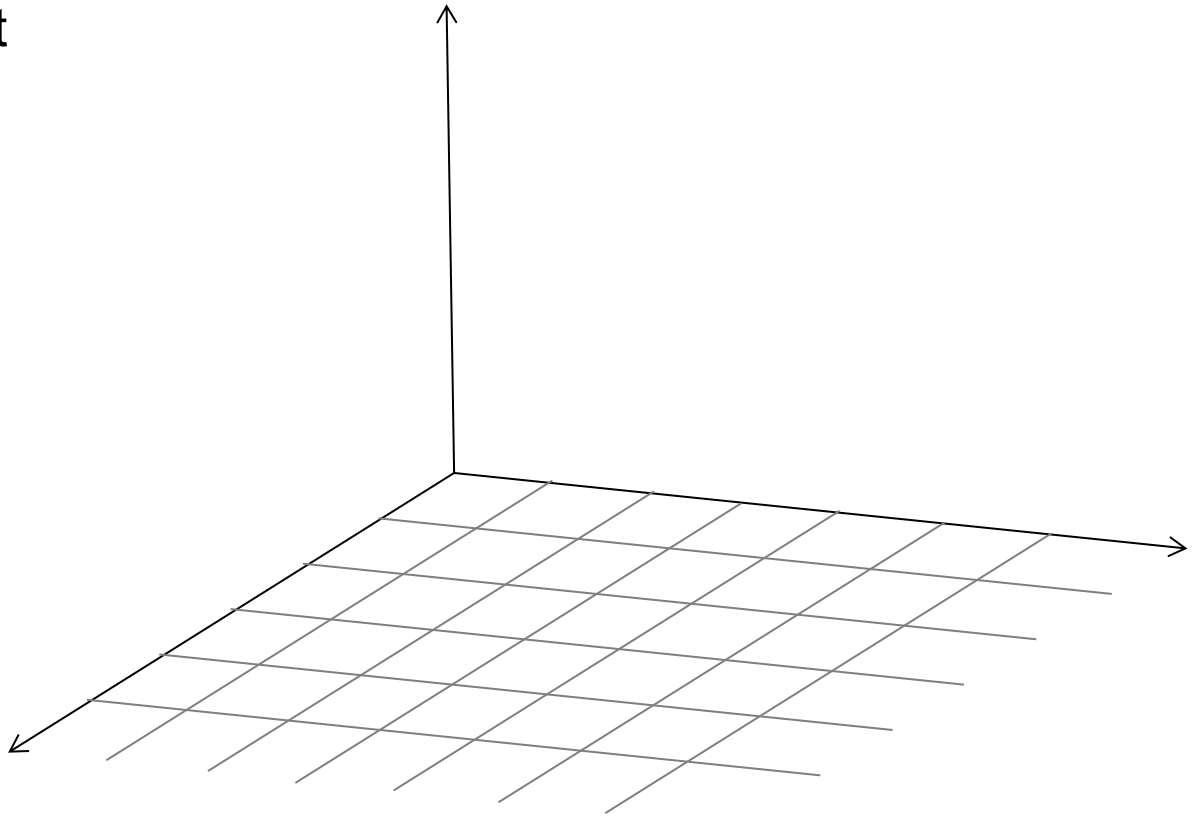
USECFrontiers

Multi-dimensional Problem

Fundamental Objectives

- F1.1 Incentives
- F1.2 Task Design
- F1.3 Type of Participant
- F1.4 Priming/Deception
- F1.5 Self-reporting
- F1.6 Type of Study

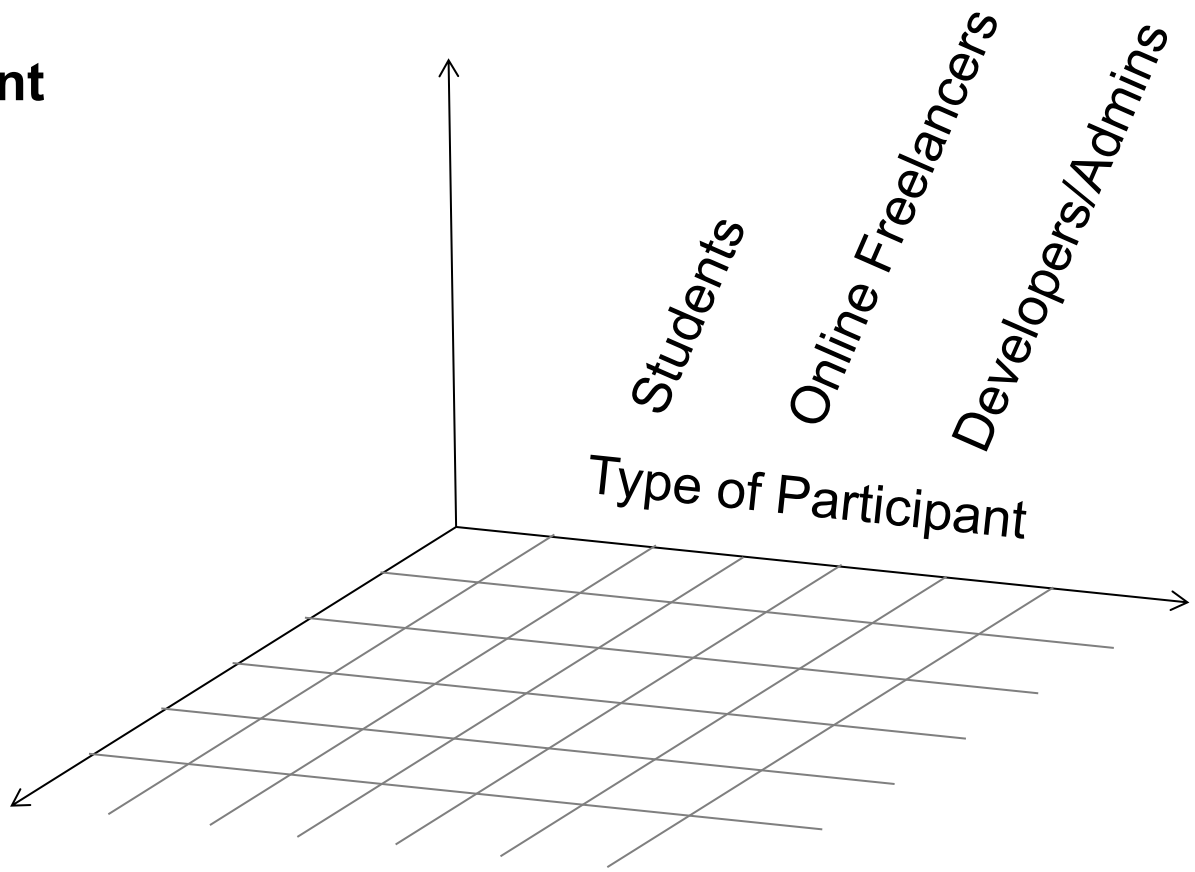
- F2 Security APIs
- F3 Risk Perception & Mental Models



Multi-dimensional Problem

Fundamental Objectives

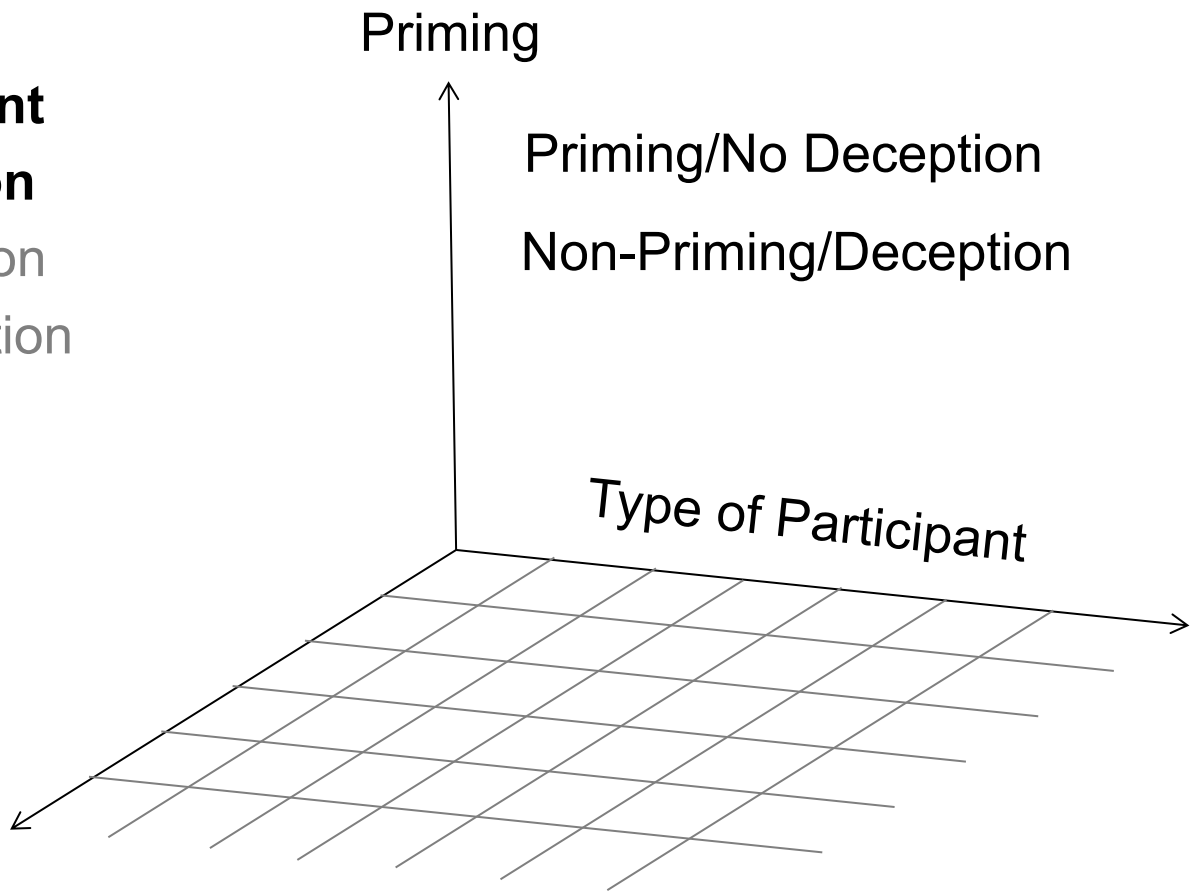
- F1.1 Incentives
- F1.2 Task Design
- **F1.3 Type of Participant**
  - Students
  - Online Freelancers
  - Developers/Admins
- F1.4 Priming/Deception
- F1.5 Self-reporting
- F1.6 Type of Study



Multi-dimensional Problem

Fundamental Objectives

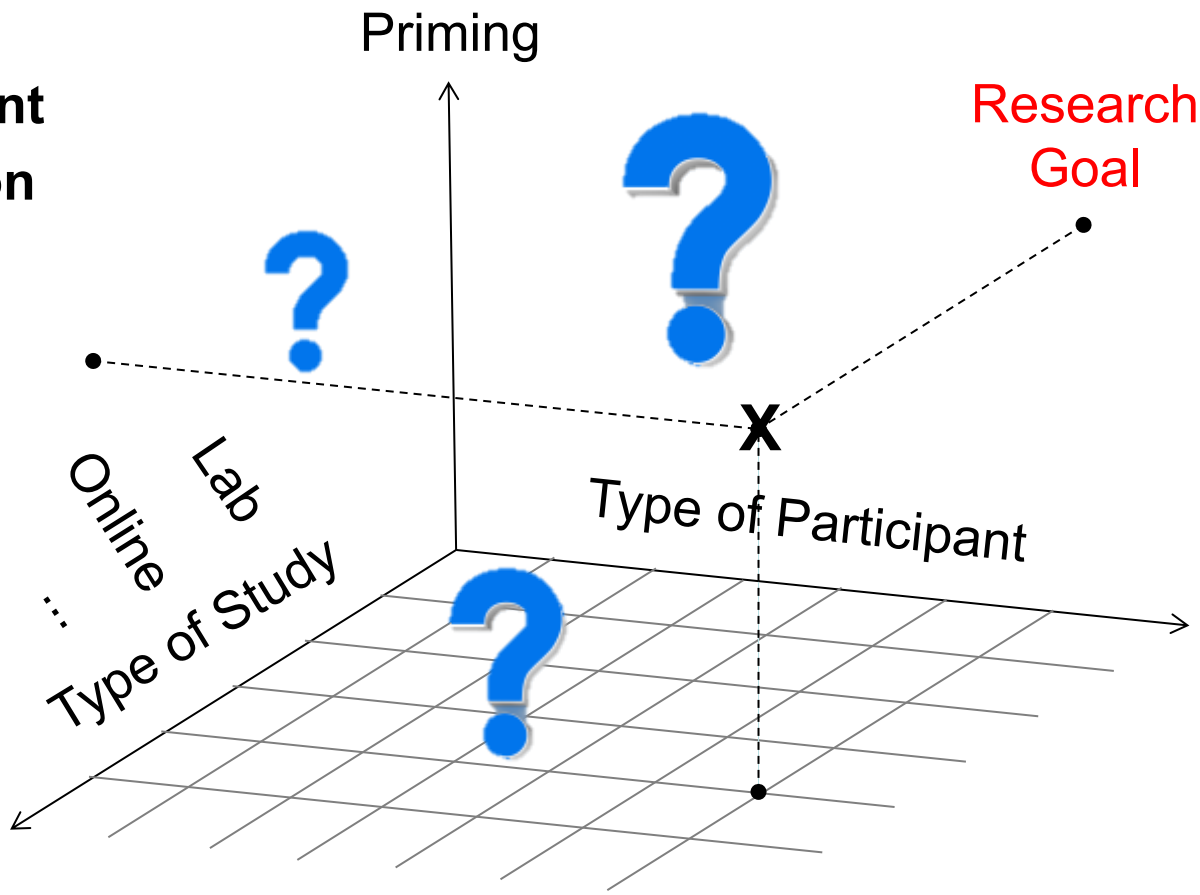
- F1.1 Incentives
- F1.2 Task Design
- **F1.3 Type of Participant**
- **F1.4 Priming/Deception**
  - Priming/No Deception
  - Non-priming/Deception
- F1.5 Self-reporting
- F1.6 Type of Study



Multi-dimensional Problem

Fundamental Objectives

- F1.1 Incentives
- F1.2 Task Design
- **F1.3 Type of Participant**
- **F1.4 Priming/Deception**
- F1.5 Self-reporting
- **F1.6 Type of Study**
  - Qual/Quant
  - Lab
  - Online
  - Field
  - Within/Between
  - Interviews
  - Focus Groups



Novel Approach

Primary Study

Randomized control trial: Create a backend service including user accounts

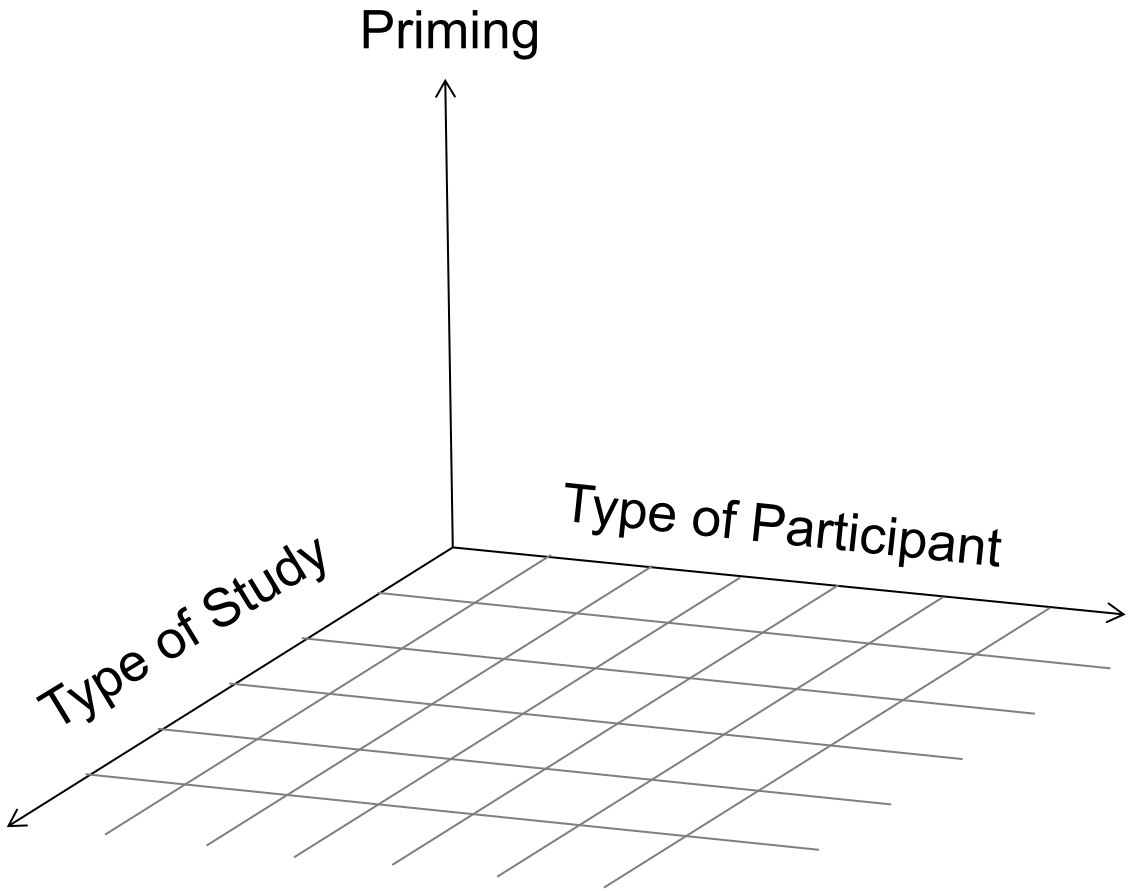
Control condition

- JCA

Treatment condition

- Spring

Meta-Study



Novel Approach

Primary Study

Randomized control trial: Create a backend service including user accounts

Control condition

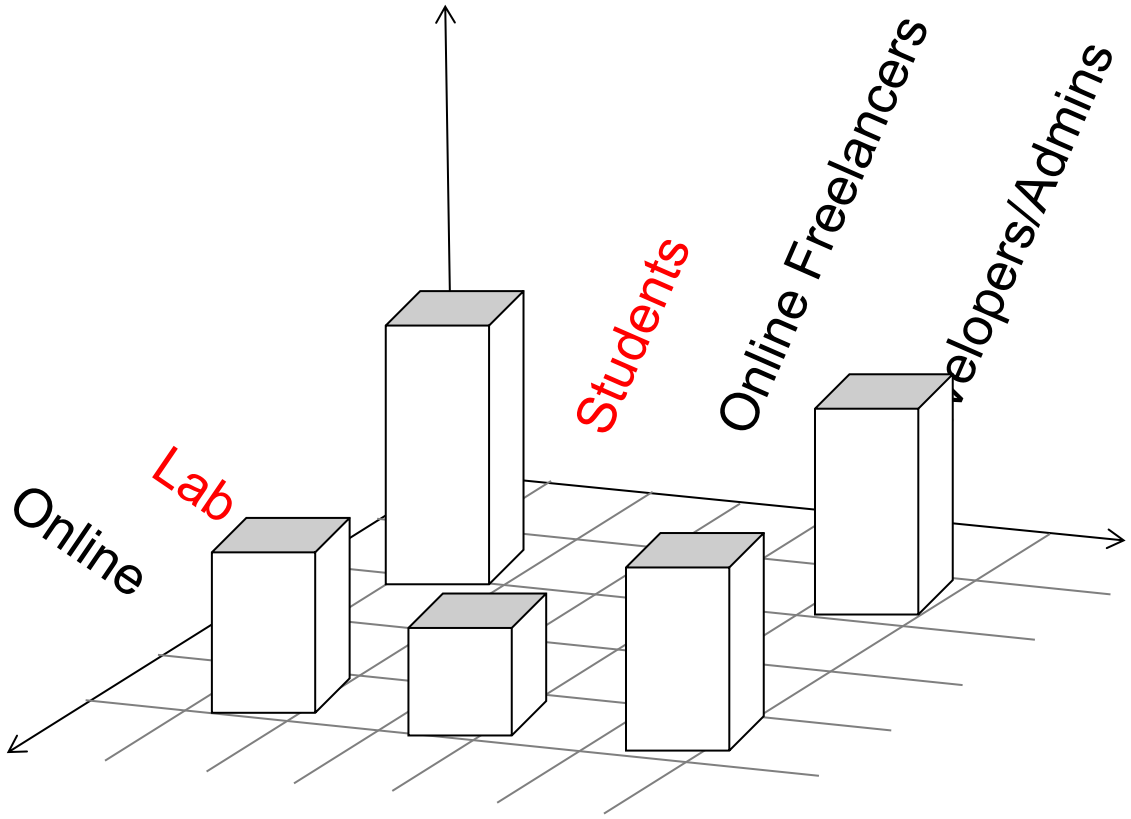
- JCA

Treatment condition

- Spring

Meta-Study

F1.3 Participants F1.6 Type of Study



Novel Approach

Primary Study

Randomized control trial: Create a backend service including user accounts

Control condition

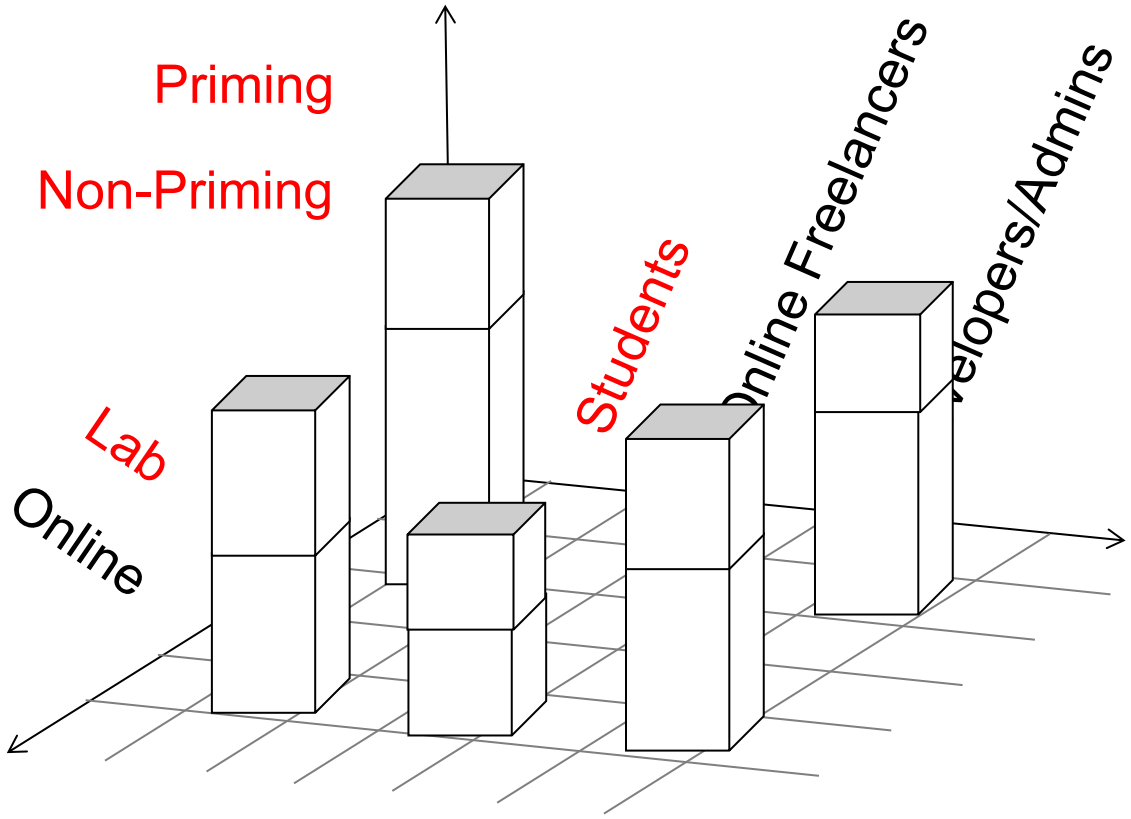
- JCA

Treatment condition

- Spring

Meta-Study

F1.4 Priming/Deception




We are here
Usable Security for Administrators and Developers holds huge potential!

More Interesting Work on DevUSEC

universitätbonn

Yasemin Acar & Sascha Fahl @ Uni-Hannover

- Fischer et al. *Stack Overflow Considered Harmful? The Impact of Copy & Paste on Android Application Security*, IEEE S&P’17
- Acar et al. *Leading By (Insecure) Example: How Internet Resources Might be Helping You Develop Faster But Less Securely*, IEEE S&P Magazine’17
- Acar et al. *You Get Where You're Looking For - The Impact of Information Sources on Code Security*, IEEE S&P‘16

universitätbonn

Developers are not the enemy!

Green & Smith IEEE S&P Magazine'16