18. Usable Security for Developers

Blase Ur and Mainack Mondal
May 30th, 2018
CMSC 23210 / 33210

THE UNIVERSITY OF CHICAGO

Security, Usability, & Privacy Education & Research

Today's class

- Making security usable for developers
  - Motivation
  - Sources of security advice
  - Crypto APIs
  - Additional aspects

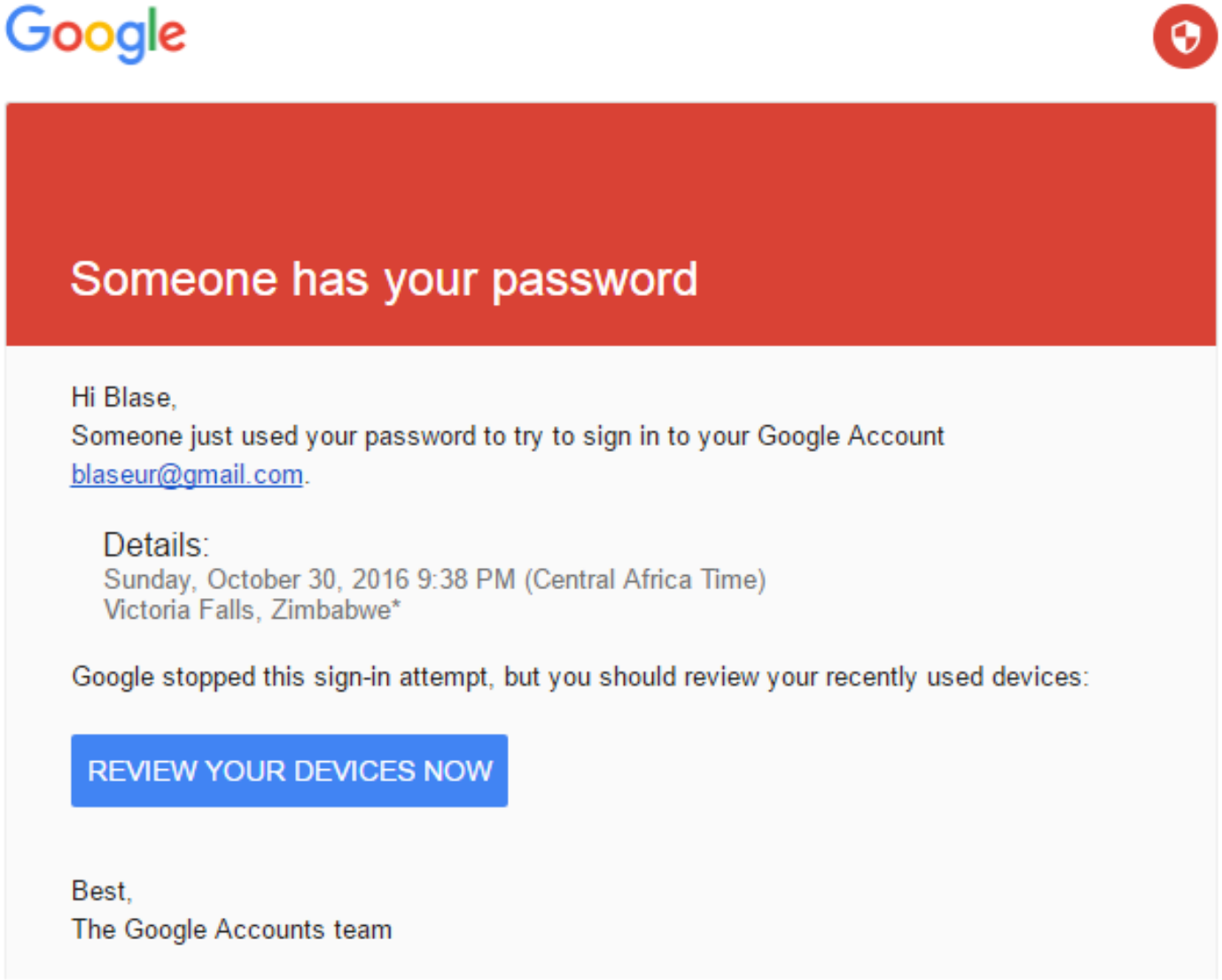
Developers Are Users, Too!

Security and human error

"Not long ago, [I] received an e-mail purporting to be from [my] bank. It looked perfectly legitimate, and asked [me] to verify some information. [I] started to follow the instructions, but then realized this might not be such a good idea … [I] definitely should have known better."

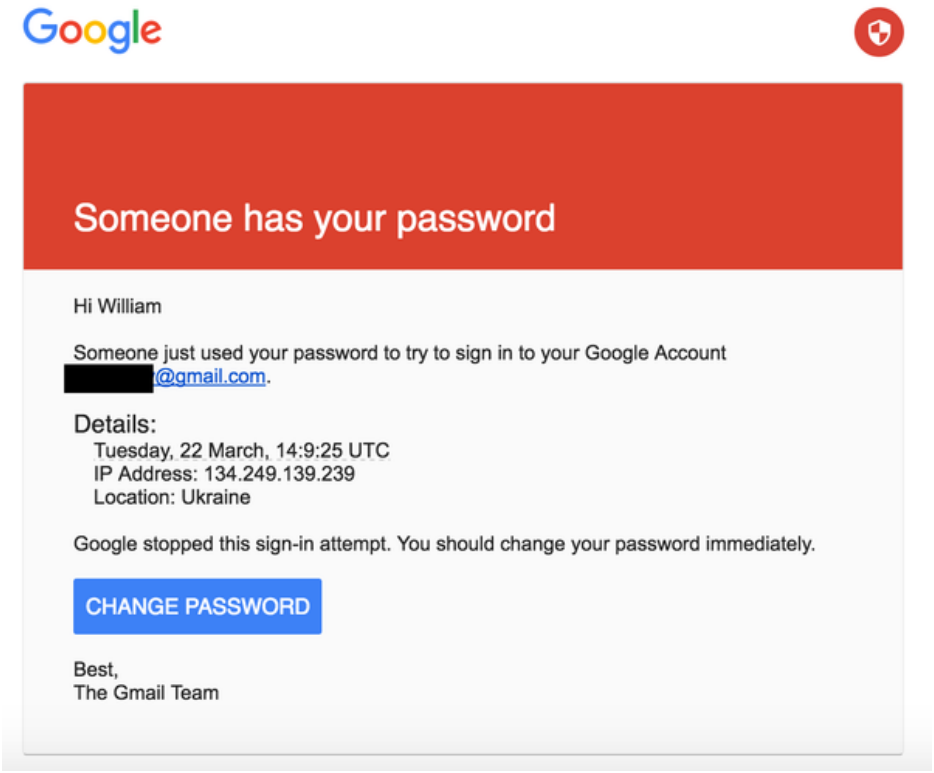
-- former FBI Director Robert Mueller

Security and human error



Security and human error

- John Podesta (more precisely an aide) receives the following:



https://mobile.nytimes.com/2016/12/13/us/politics/russia-hack-election-dnc.html

Security and human error

- IT services writes back:

From: Charles Delavan <cdelavan@hillaryclinton.com>
Date: March 19, 2016 at 9:54:05 AM EDT
To: Sara Latham <slatham@hillaryclinton.com>, Shane Hable <shable@hillaryclinton.com>
Subject: Re: Someone has your password

Sara,

This is a legitimate email. John needs to change his password immediately, and ensure that two-factor authentication is turned on his account.

He can go to this link: https://myaccount.google.com/security to do both. It is absolutely imperative that this is done ASAP.

https://mobile.nytimes.com/2016/12/13/us/politics/russia-hack-election-dnc.html

Security and human error

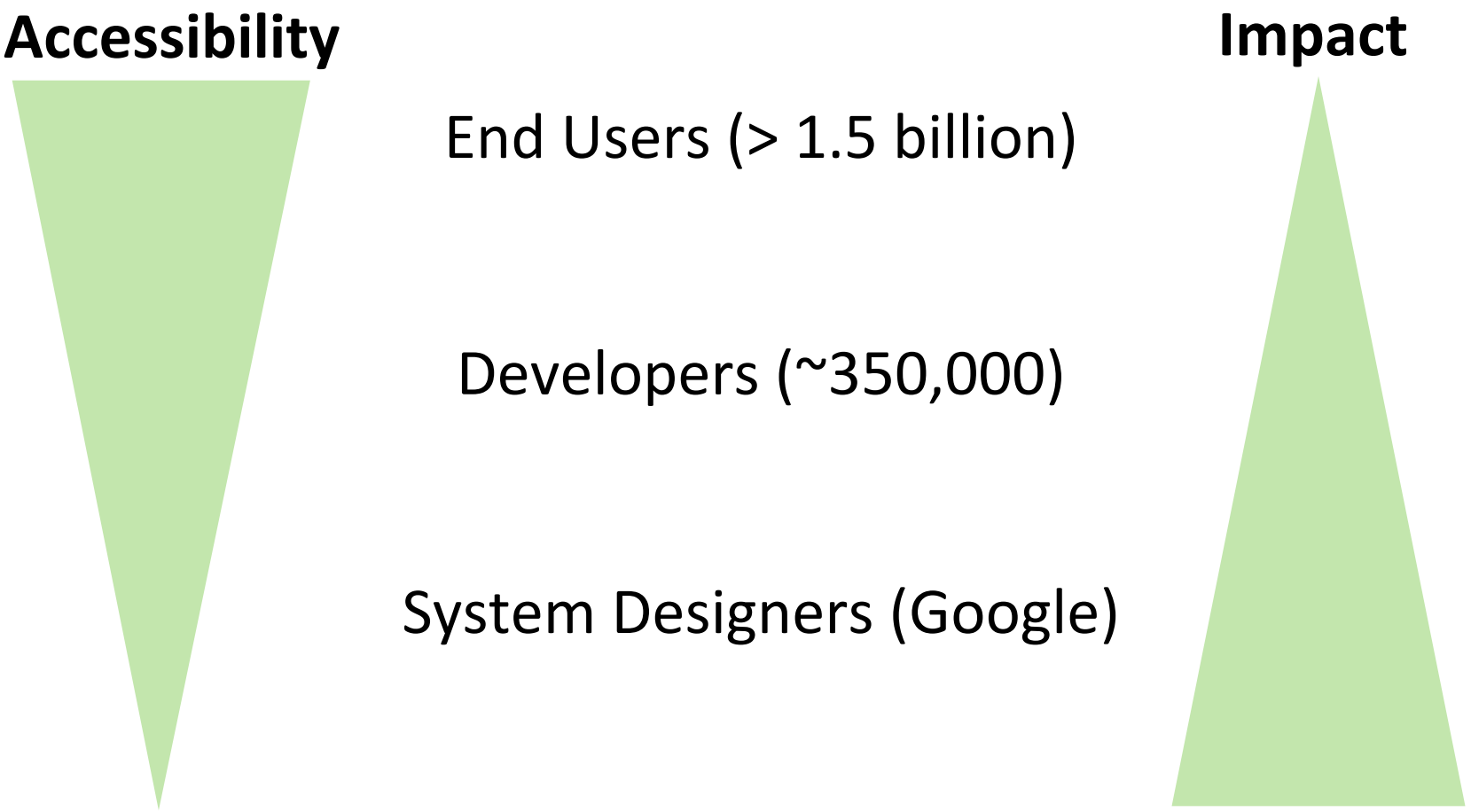
On September 2015, a call was transferred from the main DNC switchboard to the Help Desk; I was handed the phone by a Help Desk staff member who stated that the FBI was looking for the person in charge of technology at the DNC. I took the call, and learned that the FBI thinks the DNC has at least one compromised computer on its network and that the FBI wanted to know if the DNC is aware, and if so, what the DNC is doing about it. I asked if the person calling, who stated he was Special Agent [redacted] can provide me with the means of identifying whom he claims to be. He did not provide me with an adequate response, but I did stay on the phone and talked about potential risks to the DNC, without giving him any identifiable information about the DNC, its personnel, or its assets. I did say that the DNC has, in the past, received phishing attack attempts, and ransom-ware type of attacks. The Special Agent told me to look for a specific type of malware dubbed "dukes" by the US intelligence community and in cyber-security circles.

https://mobile.nytimes.com/2016/12/13/us/politics/russia-hack-election-dnc.html

~~Why are users stupid or lazy?~~

How can we make security more usable?

Beyond end users for more impact



Example: Android

What about software developers?

Developers are experts, right?

Or not.

```
goto fail;
goto fail;
```

http://www.slate.com/content/dam/slate/articles/technology/future_tense/2016/06/160610_FT_Barbie-promo.jpg.CROP.cq5dam_web_1280_1280_jpeg.jpg
https://i.ytimg.com/vi/tQms037U72w/maxresdefault.jpg

Why are **developers** stupid or lazy?

How can we make **secure programming** easier?

Lessons learned: Usec for end users

- You are not your user
- Security is a secondary concern
- More is not always better

You are not your user

- Confusing warnings and error messages
- Too much security jargon
- Don’t assume security knowledge just because they know how to program
- Design for usability, evaluate it explicitly

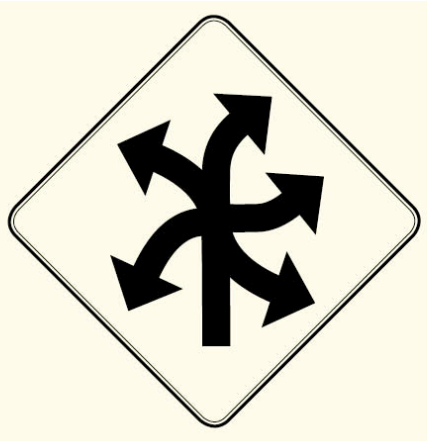
Security is secondary

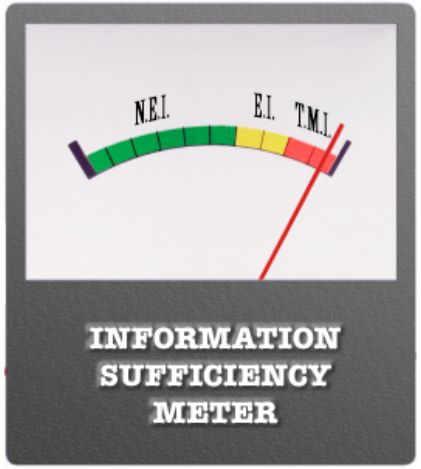
- No one turns on their computer to do "security"
  - Functionality, time to market, maintainability, etc.
  - May (appear to) conflict with security
- Attention and time are limited!
- Try: Take developer out of the loop
- Try: Persuasive design

https://www.scripps.org/sparkle-assets/images/multitastking-600x375.jpg

More is not always better

- Too much advice is overwhelming
  - Hard to roll it back
- Can't just keep asking users (developers) to do and remember more stuff



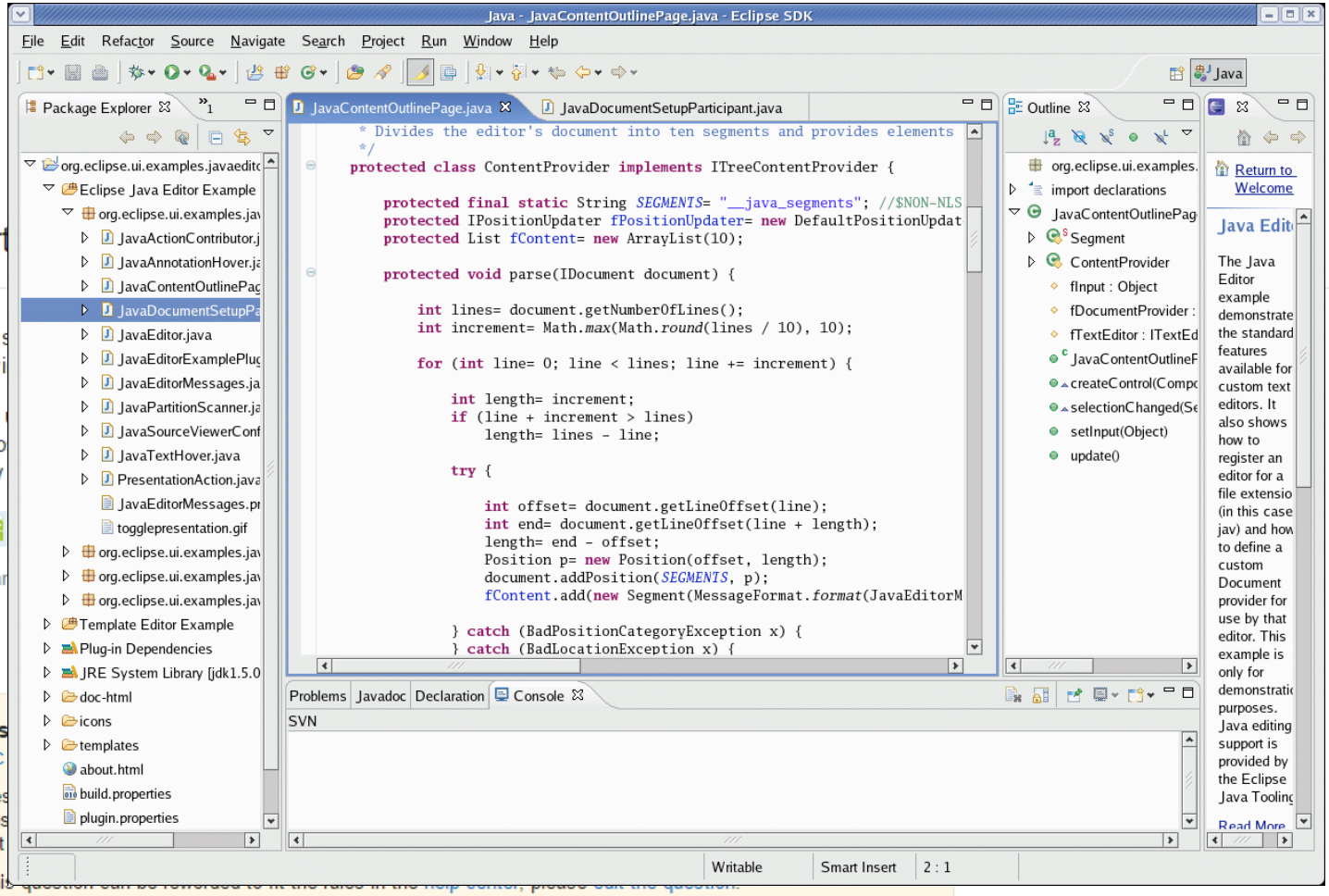


https://rpseawright.files.wordpress.com/2013/10/too-much-info.png

http://blogs.babycenter.com/wp-content/uploads/2011/09/too-much-sugar.jpg

YOU GET WHERE YOU'RE LOOKING FOR (IEEE S&P 2016)

Has this happened to you?



That doesn’t seem right ….

- Answer suggests to trust all certs
  - Many real apps [Fahl+ 2012]
- Some interviewees: pasted from internet

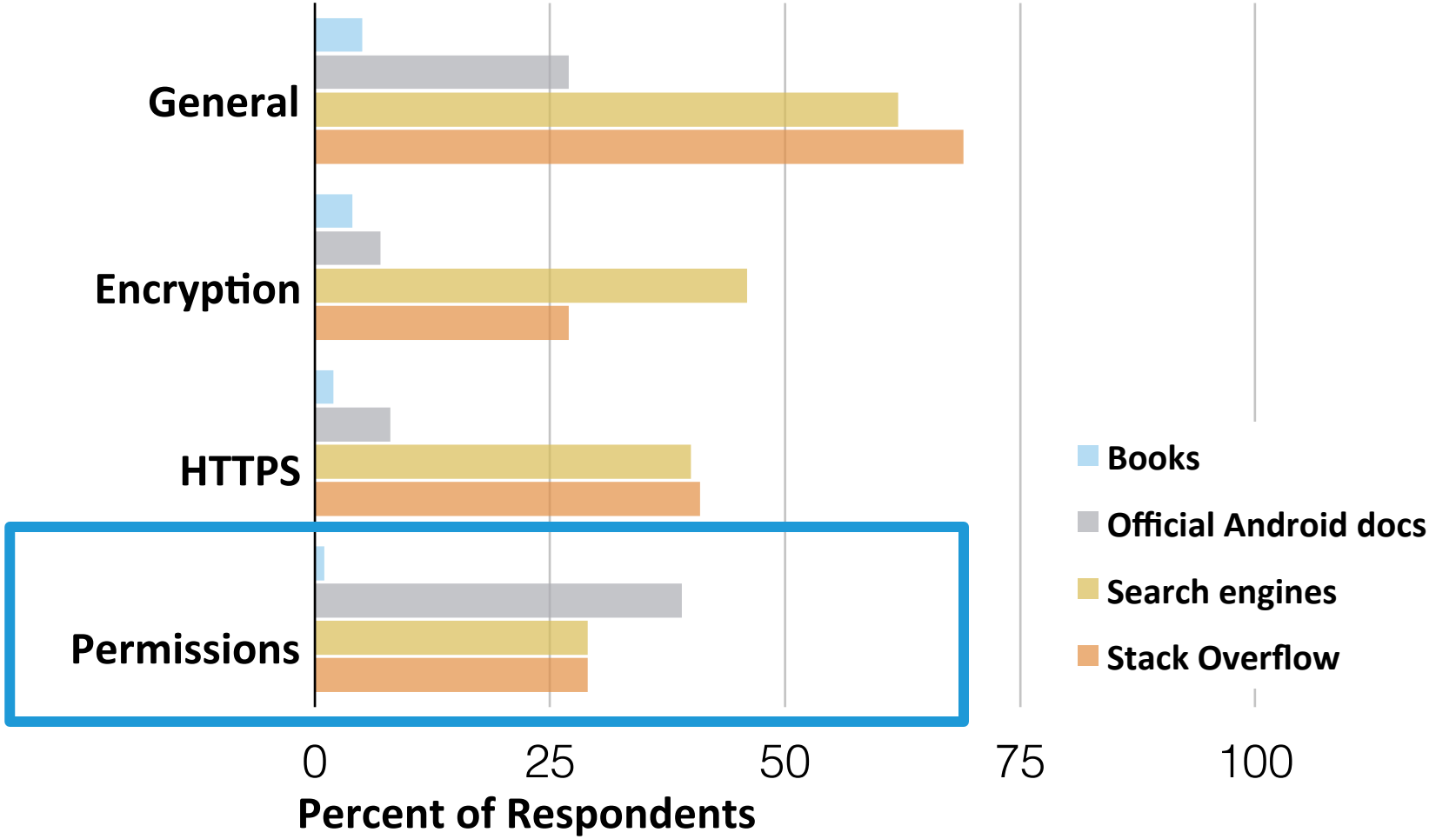
Stack Overflow considered insecure

- “Everyone knows” copy-paste from the internet is bad for security
  - Particularly for “amateur” app devs?
- Can we measure this empirically?
- How does it contrast with official docs?
- What do real devs do?

Online developer survey

- Sent 50k invites, collected from Play
  - 295 valid responses
- Strategy for help with security/permissions
- General use of programming resources

Where do you look up …

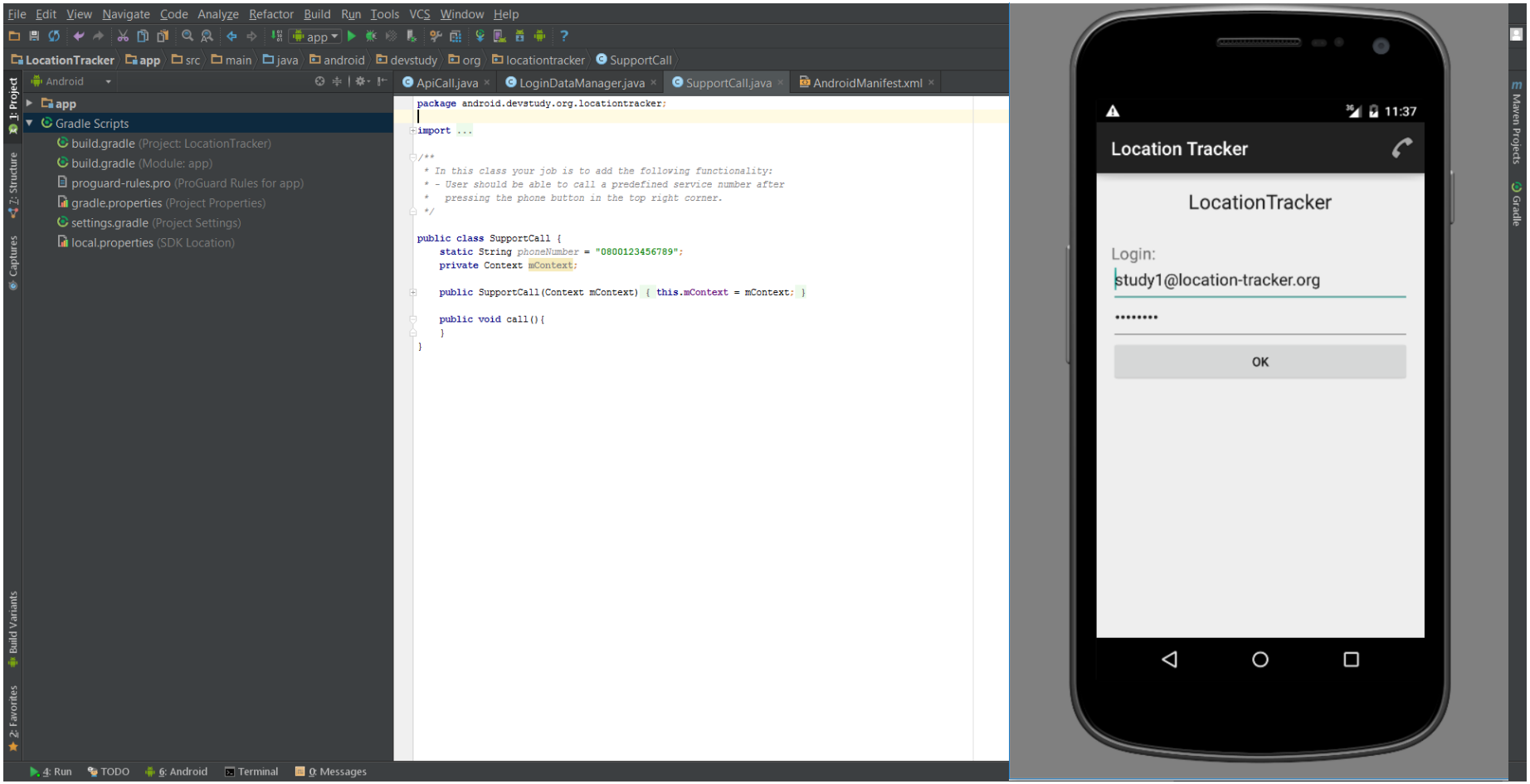


69% Stack overflow, 62% search engines, 27.5% official

Next, a lab study

- Complete four short programming tasks
  - Designed to have secure/insecure solutions
- Resources constrained by condition:
  - Official docs, Stack Overflow, book, free choice
- Exit interview
- Not primed for security or privacy!

Skeleton app, emulator



Task 1: Secure networking

- Convert HTTP to HTTPS
  - In presence of X.509 cert error

- Sample secure solution:
  - Accept only this cert
- Sample insecure solution:
  - Accept all certs

http://5zin.com/certificate-of-authenticity-template.html

Task 2: Inter-component comms

- Given a service, limit access to only apps from same developer

- Sample secure solution:
  - Define a “signature” permission
- Sample insecure solution:
  - Export publicly

Task 3: Secure storage

- Store user ID and password locally

- Sample secure solution:
  - Private shared preference
- Sample insecure solution:
  - Public on SD card

http://www.routercheck.com/administrator-password/

Task 4: Least permissions

- Dial a customer-support phone number

- Sample secure solution:
  - Dial but don't call
- Sample insecure solution:
  - Call (extra permission)

http://wizwas.com/index.php/2009/11/02/the-door-to-yesterday-6/

Evaluation

- Correctness: Does it compile and work?
- Security: If it works, was solution secure?
  - Coded manually in predefined categories
- Self-reported sentiment
  - Security thinking
  - Correctness and usefulness of resources

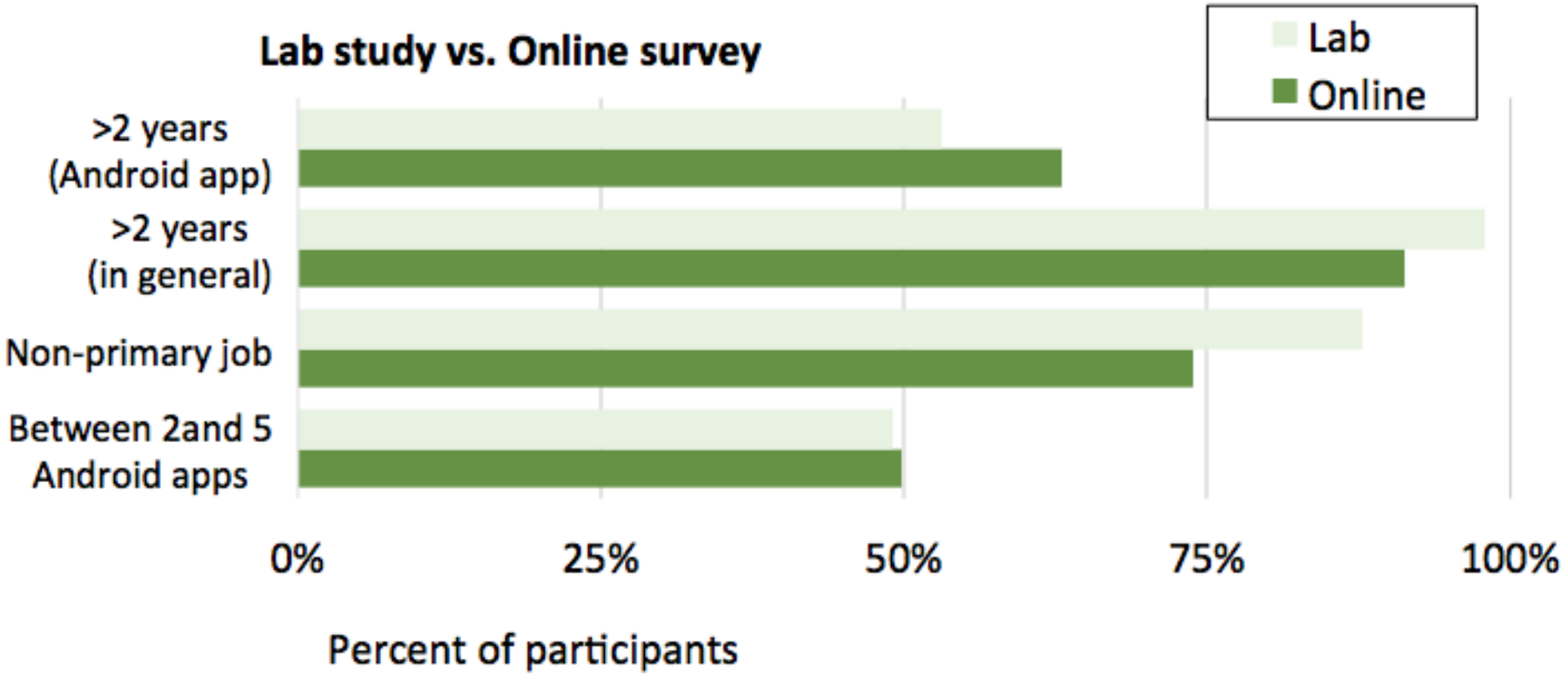
Recruitment

- In/around 3 universities, U.S. and Germany
  - Email, flyers, craigslist, developer forums
- 1+ Android course or 1+ yrs pro
- Pass basic Android knowledge questions

Participants

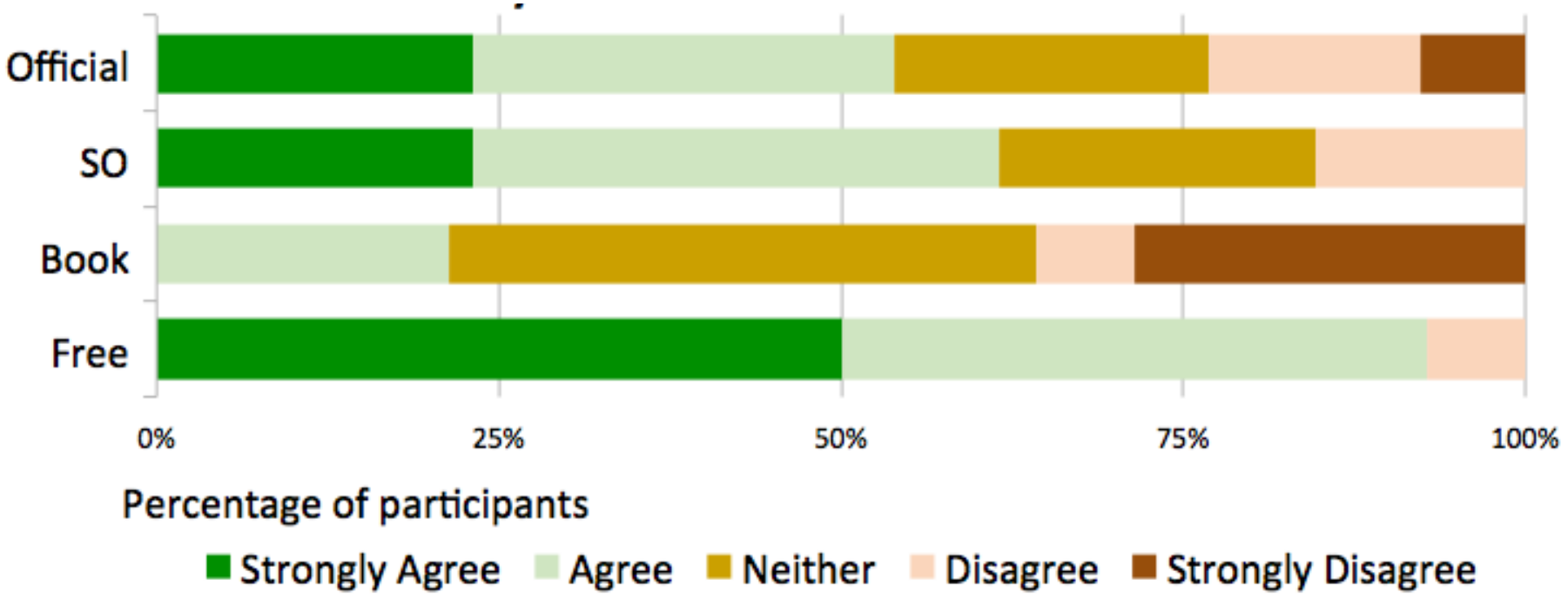
- 54 total
- 13 or 14 per condition
- 12 U.S., 42 Germany
- Ages 18-40; median 25
- 46 men, 8 women
- 14 professional, 40 non-professional

Demographics: lab vs. online



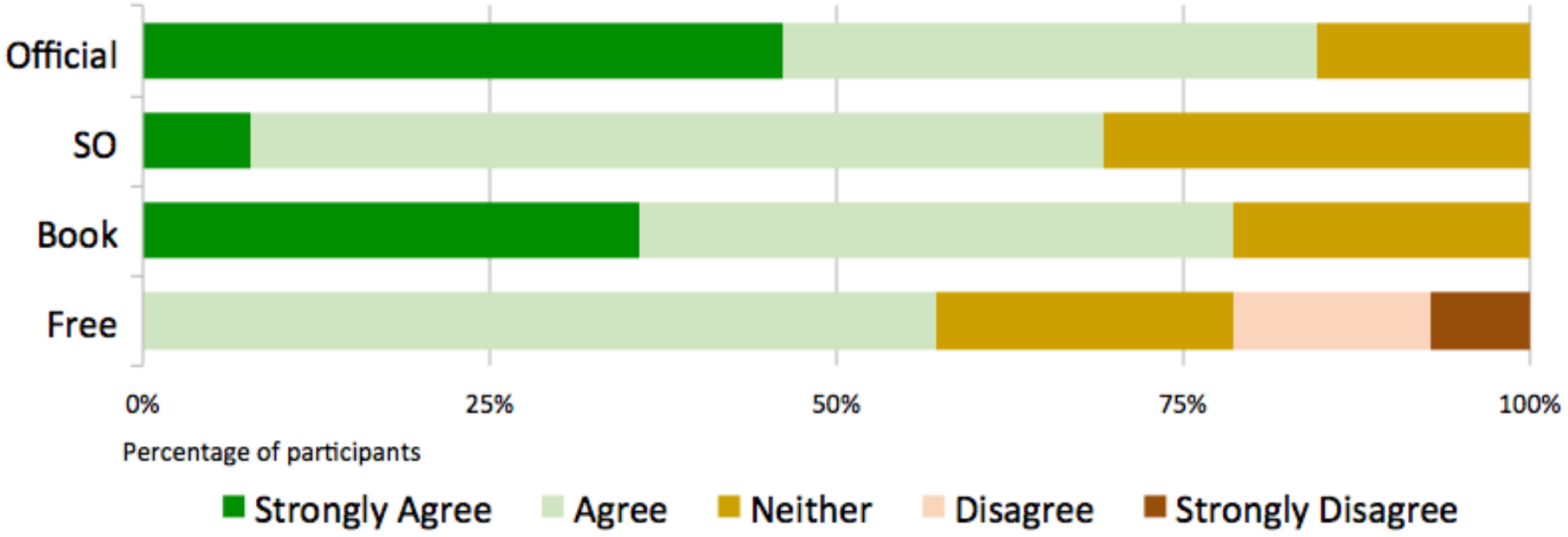
Many similarities; Lab had more formal education

Resource was easy to use



Free choice was easiest; book was worst

Resource was correct

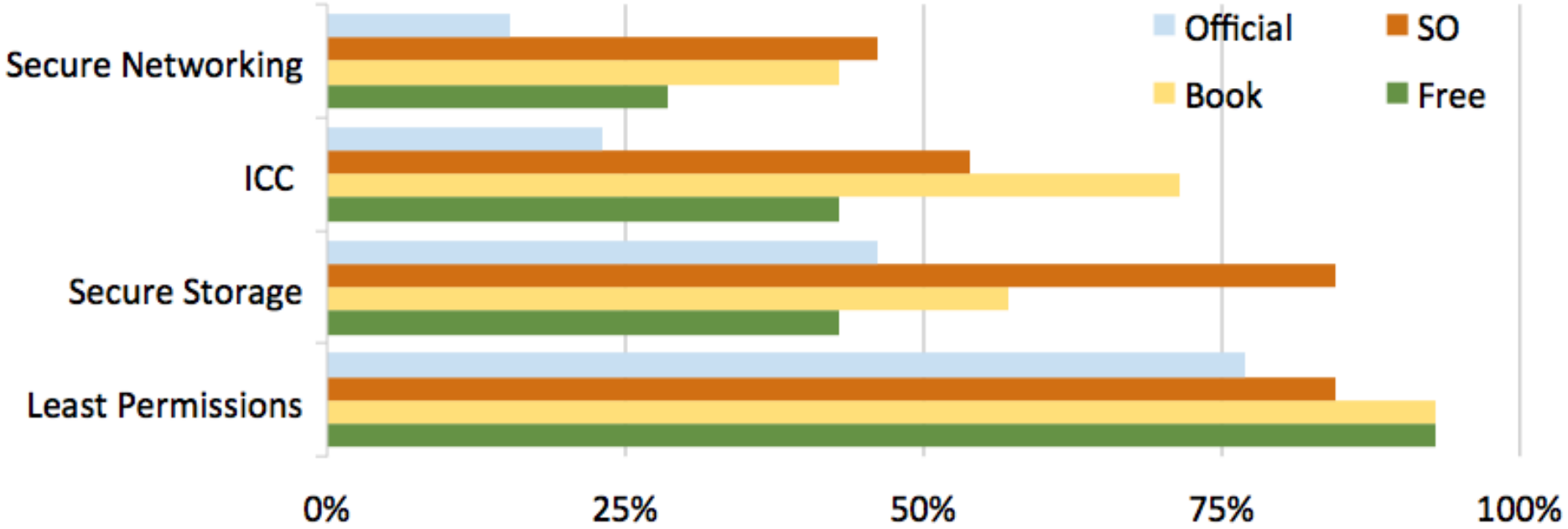


Books, official docs considered most correct

Security thinking

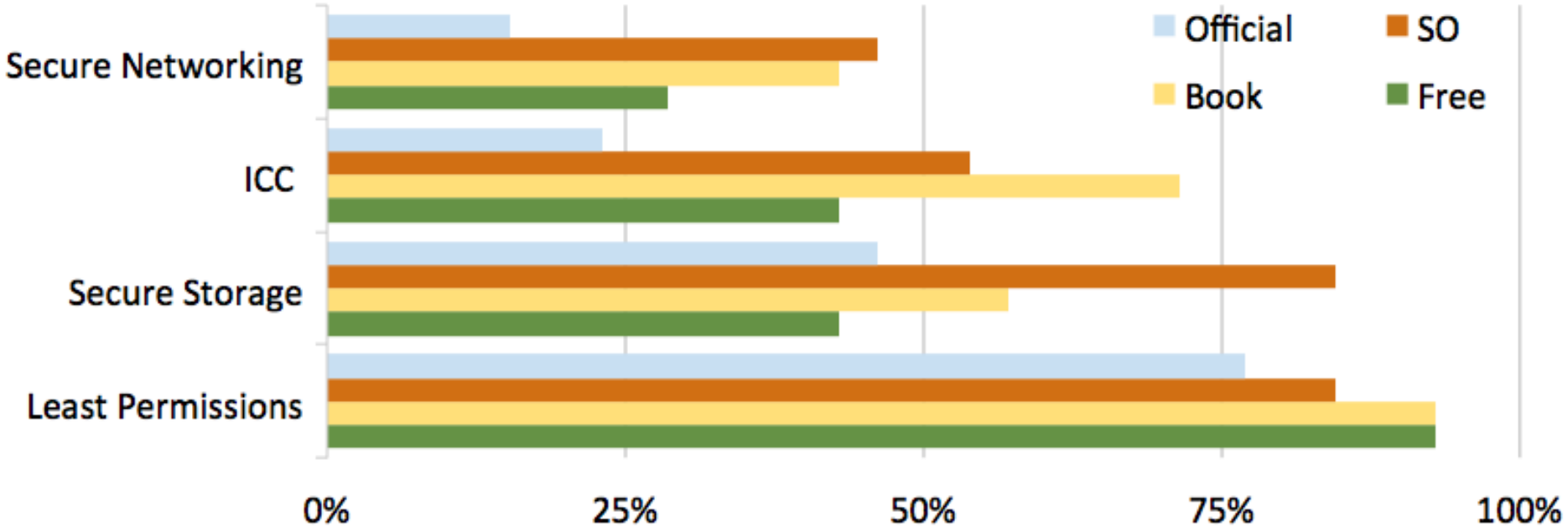
- Observed via think-aloud:
  - 16% thought about it
  - 5% said they ignored it for study / time
- Self-reported: 60% thought about it
- No significant difference in conditions

Functional correctness



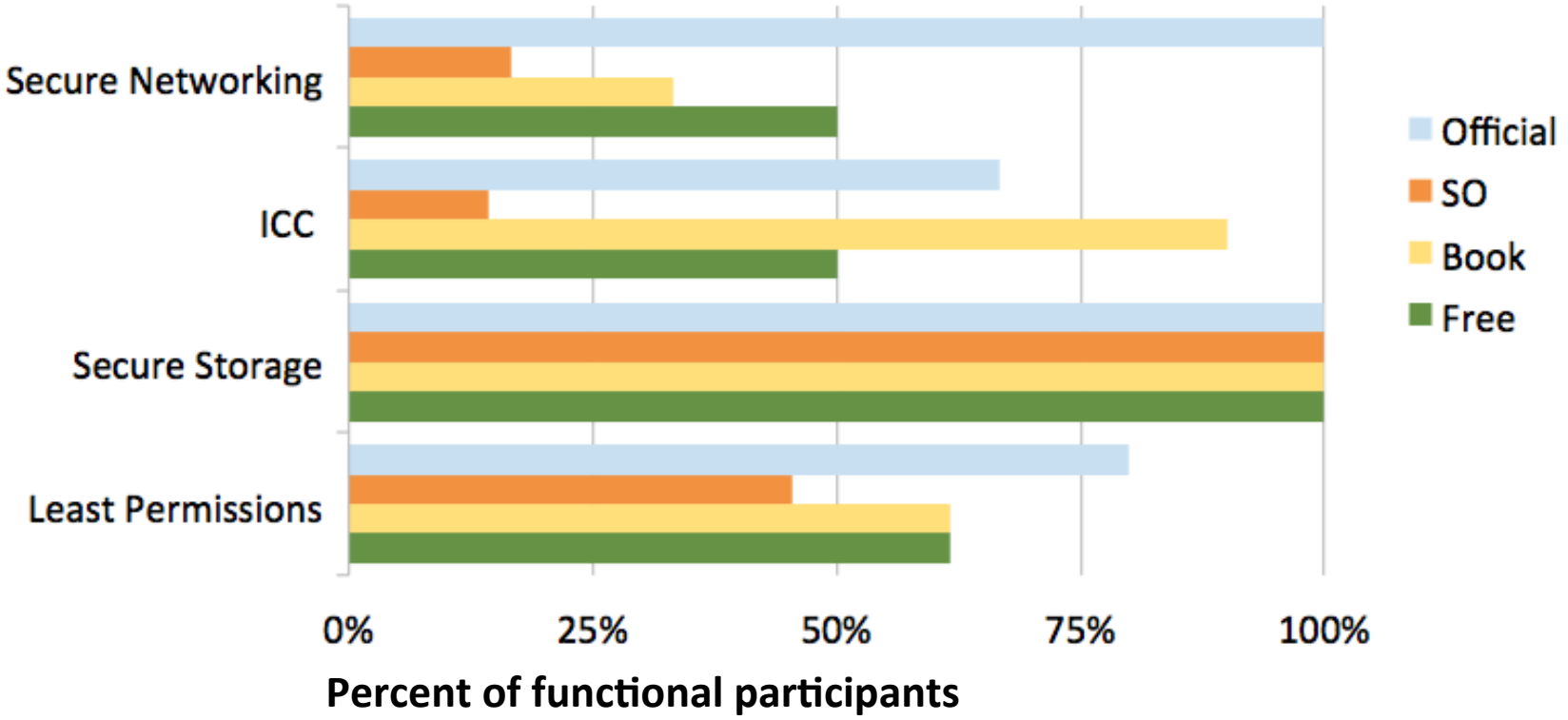
- SO (67%) and Book (66%) performed best
- Official (40%) performed worst
  - Significantly worse than SO

Functionality by task



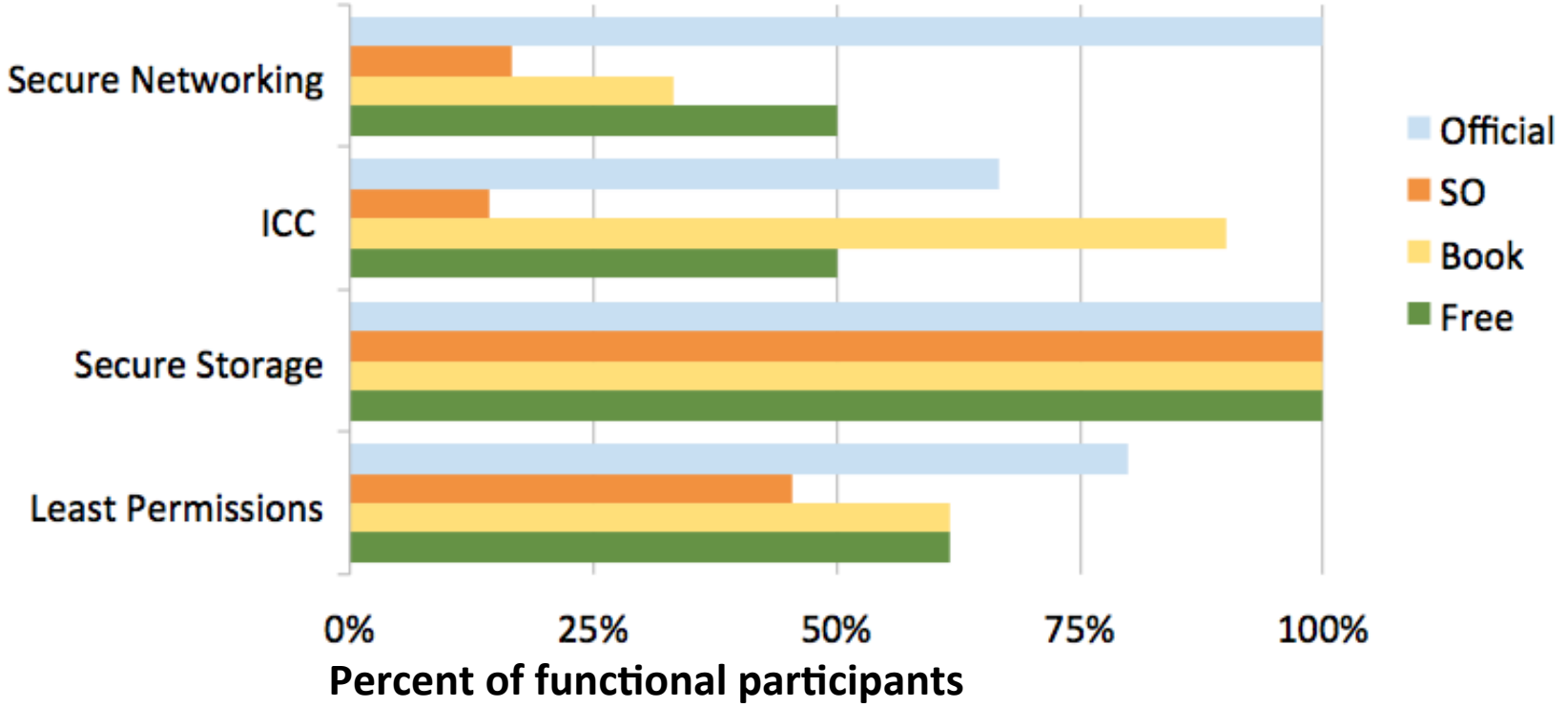
- Easiest: Least permissions (87%)
- Hardest: Secure networking (33.3%)

But what about security?



SO worst (51%), Official best (86%) (significant)

Security by task



- Storage: 100% of functional solutions secure
- Networking: Only 39%

Professionals vs. students

- More functional
- But not significantly more secure!

Lookup behavior

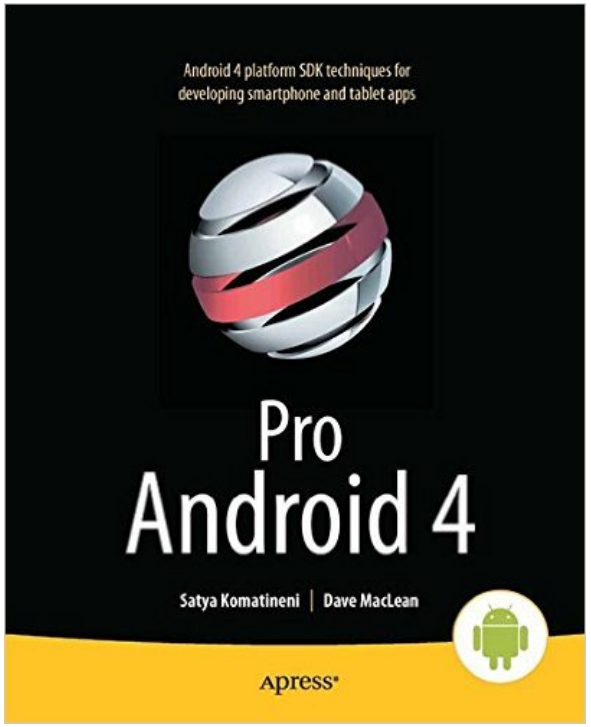
- Official: scrolling, clicking internal links
- Stack Overflow: many search resets
- Free choice:
  - Everyone used official, all but one used SO
  - One picked up a book!
  - Results closest to SO

A closer look at Stack Overflow

- Collected via browser history
- 149 unique pages, 41 relevant
- 20 with code snippets
  - 7 only secure, 10 only insecure, 3 both
  - 3 insecure have warnings

So now what?

- If you want functional, secure code:
- Cut off the internet, give your devs a book!



JOKE

Real takeaways

- Stack Overflow: quick, functional solutions
  - Official docs don't
- But, it's less secure than official or books
- We need resources that integrate both!
  - Add a security rating to influence upvote?
  - Integrate Q&A into official docs?
  - Use SO to identify trouble spots, provide code snippets in the official docs?

Comparing Crypto APIs

Getting crypto right is hard

- Developers must pick:
  - algorithm
  - mode of operation
  - key size, etc.
- Challenging, error prone (ICSE'16)
- Alternatives claim to be more usable
  - libsodium, keyczar, cryptography.io
- Is this really true?

Online developer study

- Short python tasks, secure/insecure solutions
  - Symmetric: key gen./storage, encrypt/decrypt
  - Asymmetric: also certification validation
- One of 5 libraries:
  - PyCrypto, M2Crypto, cryptography.io, keyczar, PyNacl
- Exit survey

Library	Current Version	Designed for Usability	Symmetric Key Generation	Symmetric Encryption/Decryption	Secure Symmetric Key Storage	Asymmetric Key Generation	Asymmetric Encryption/Decryption	Secure Asymmetric Key Storage	Certificate Validation
PyCrypto	2.6.1	○	●	●	●	●	●	●	○
M2Crypto	0.25.1	○	●	●	●	●	●	●	●
cryptography.io	1.4	●	●	●	●	●	●	●	●
Keyczar	0.716	●	◐	●	○	◐	●	○	◐
PyNaCl	1.0.1	●	●	●	○	●	●	○	◐

● = fully applies; ◐ = partly applies; ○ = does not apply

Not all libs support all tasks well

Certificate validation

Goal: Verify that the SSL certificate from the central Citizen Measure server was issued by the Let's Encrypt Certificate Authority to ensure that citizen reports are not being intercepted. You have to validate the certificate's digital signature and common name. For your convenience, the SSL certificate from the Citizen Measure server is stored in ./citizenMeasureCertificate.pem and the Let's Encrypt Certificate Authority certificate in ./leca.pem. You can take also a look at the Let's Encrypt X3 Root CA and the server certificate.

In [0]:

```
import nacl

def validate(certificate, root_certificate, hostname="citizen-measure.tk"):
    """
    Purpose:
        Validate the given certificate's digital signature and common name.

    Arguments:
        certificate: The certificate to validate.
        hostname: The server's hostname.

    Return value:
        validationresult: True if validating the certificate is correct, False otherwise.

    Notes:
        - The Citizen Measure server certificate can be found at ./citizenMeasureCertificate.pem
        - The Let's Encrypt Certificate Authority certificate can be found at ./leca.pem
        - If you used any other information source to solve this task than the linked documentation (e.g. a post on StackOverflow, a blog post or a discussion in a forum), please provide the link right below:
        - additional information sources go here (e.g. https://stackoverflow.com/questions/415511/how-to-get-current-time-in-python)
    """

    # This is where your code goes
    return False

# This is to test the code for this task.
certificate = open("./citizenMeasureCertificate.pem").read()
root_certificate = open("./leca.pem").read()
assert validate(certificate, root_certificate, "citizen-measure.tk"), "Certificate validation failed."
print "Task completed! Please continue."
```

Run and Test

Get unstuck | NOT solved, Next Task | Solved, Next Task

Skeleton code, online code editor

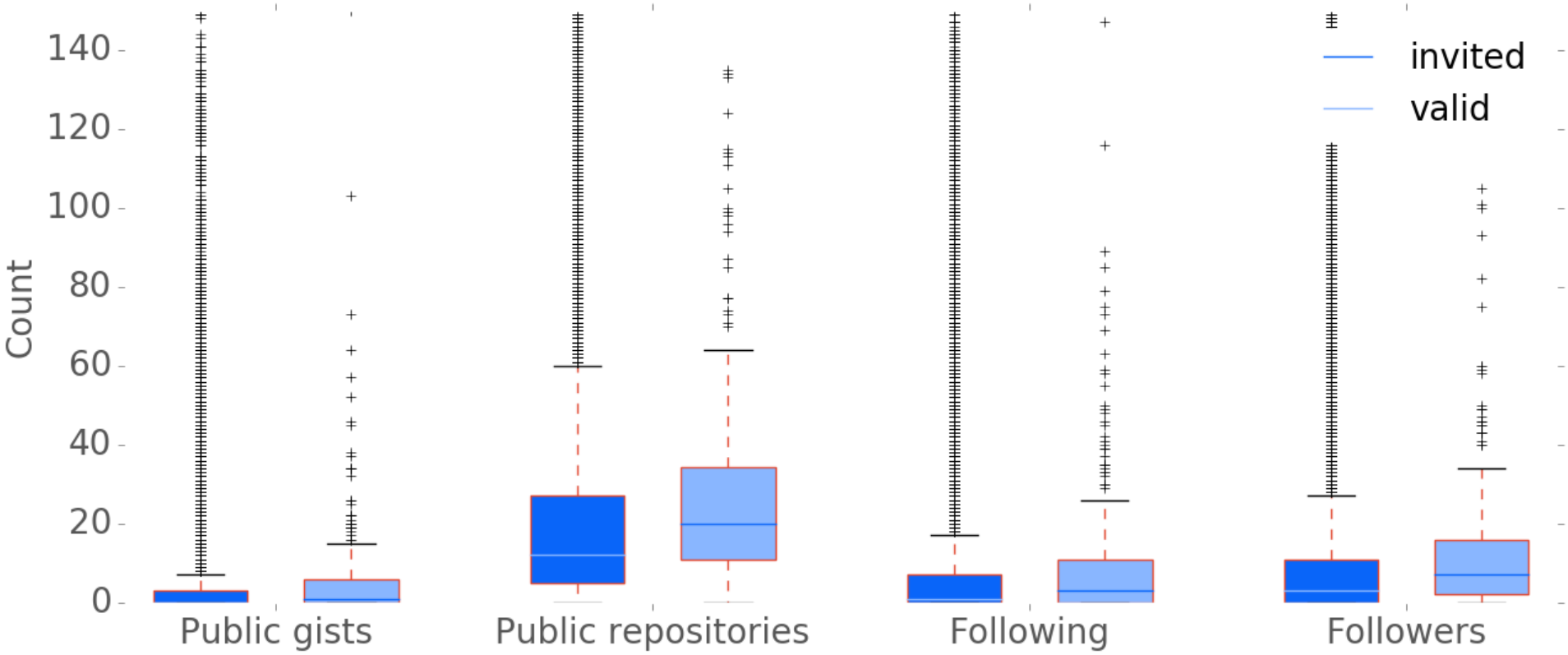
Evaluation

- Correctness: Runs without errors, “works”
- Security: Manually coded
  - Predefined categories, 2 independent coders
- Self-report
  - Security thinking
  - System Usability Scale (SUS)
  - New API scale we designed
- Primarily analyzed w/ multiple regression

Recruitment via Github

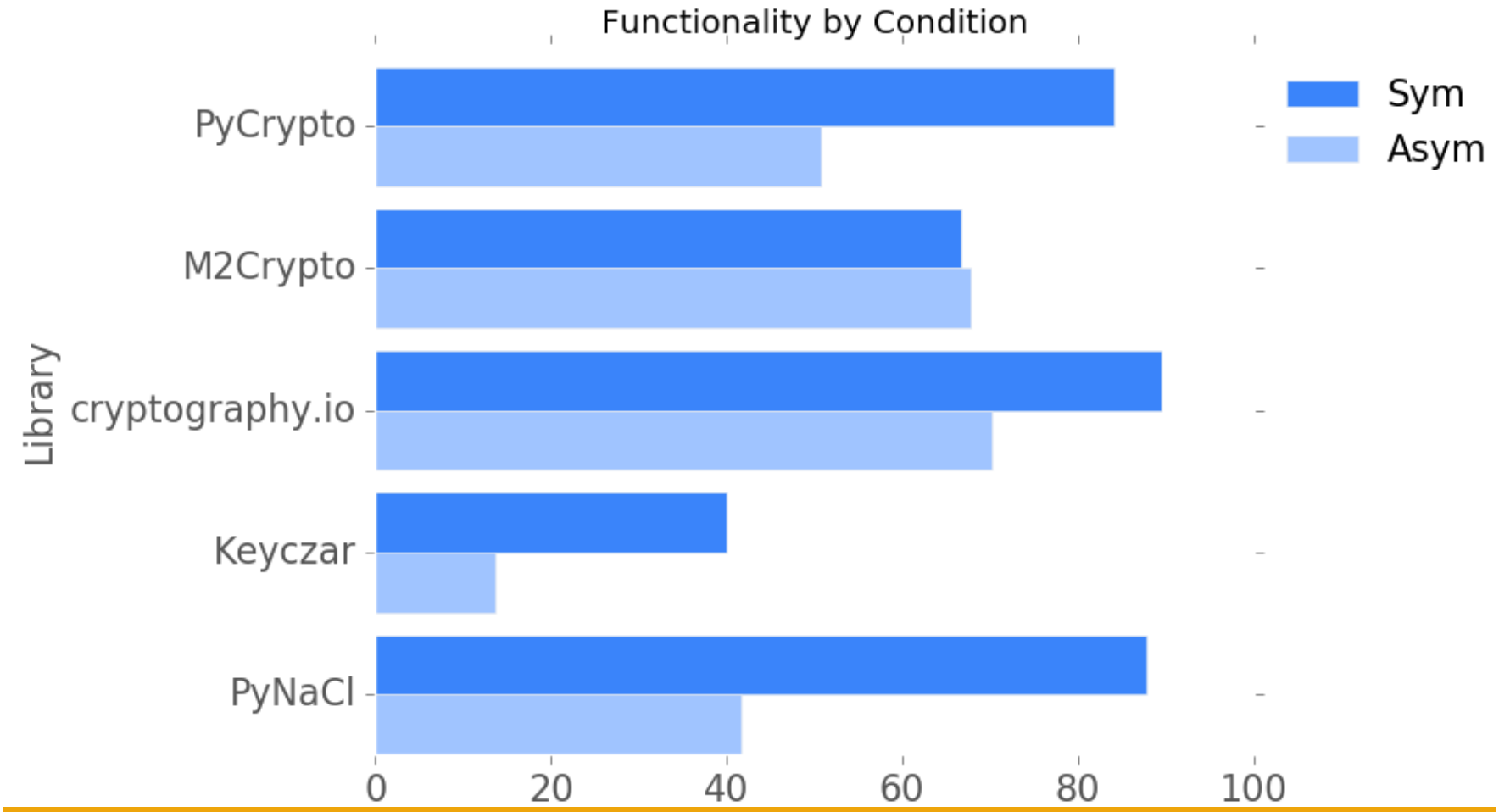
- Scraped committers to 100k Python repos
- Invited random 50k of these
- Final, “valid” sample: 256
  - 208 professionals
  - 198 w/ no security background
  - 1571 who consented; many dropped out

Invited vs. participated



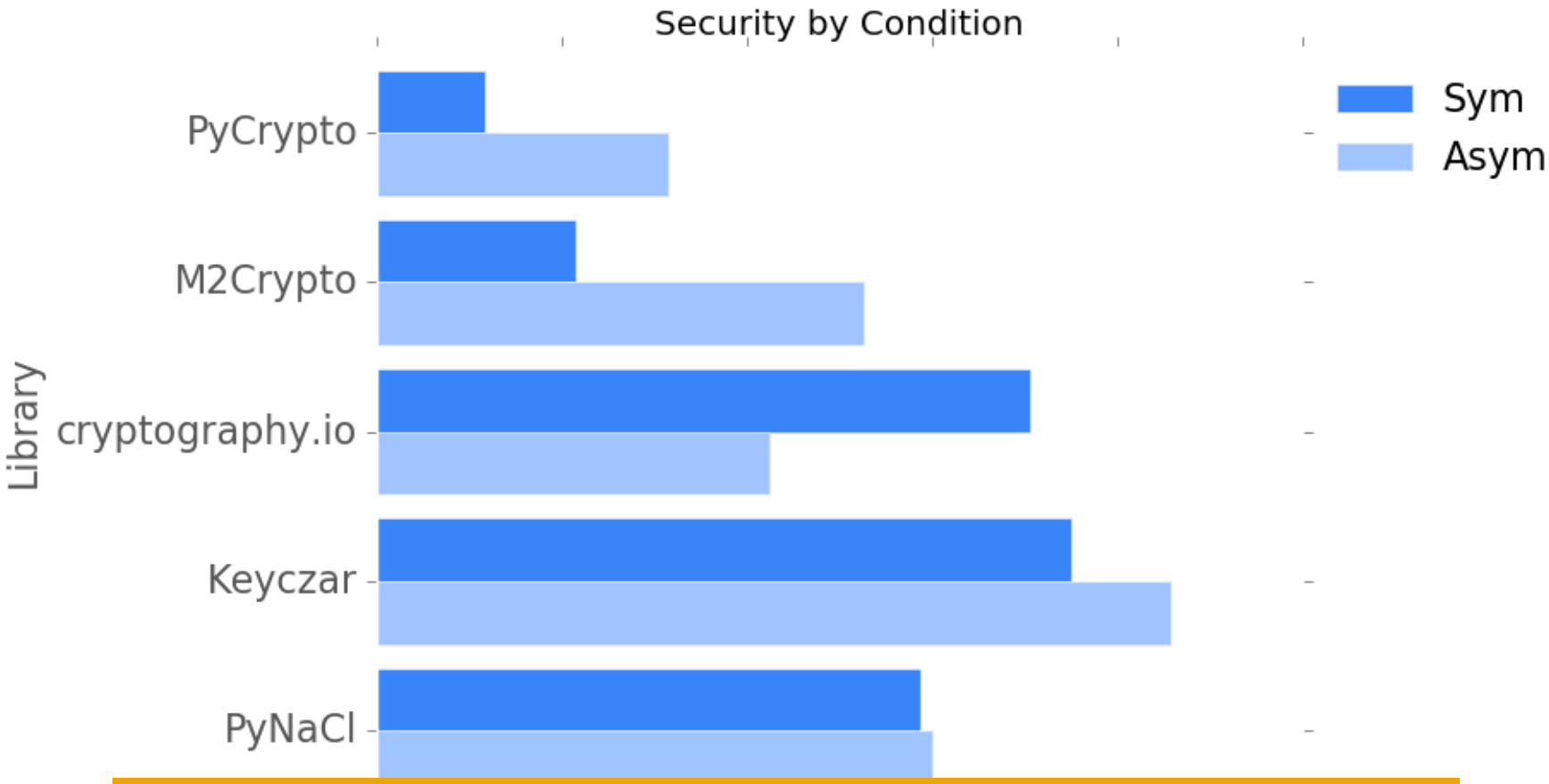
Many similarities; Participants slightly more active

Functionality by library



Keyczar, m2crypto worst; c&p helps (significant)

Security (among functional)



"simplified" libs are most secure;
asymmetric more secure than symmetric

Self-reported data

- Believed secure but weren't: 20% of tasks!
  - Not different by library
- SUS: Nothing better than mediocre
  - Most disliked: keyczar, m2crypto, asymm
  - Very similar to functionality results
- From our scale: Documentation is key!
  - Keyczar: "Your documentation is bad and you should feel bad."

Participant background

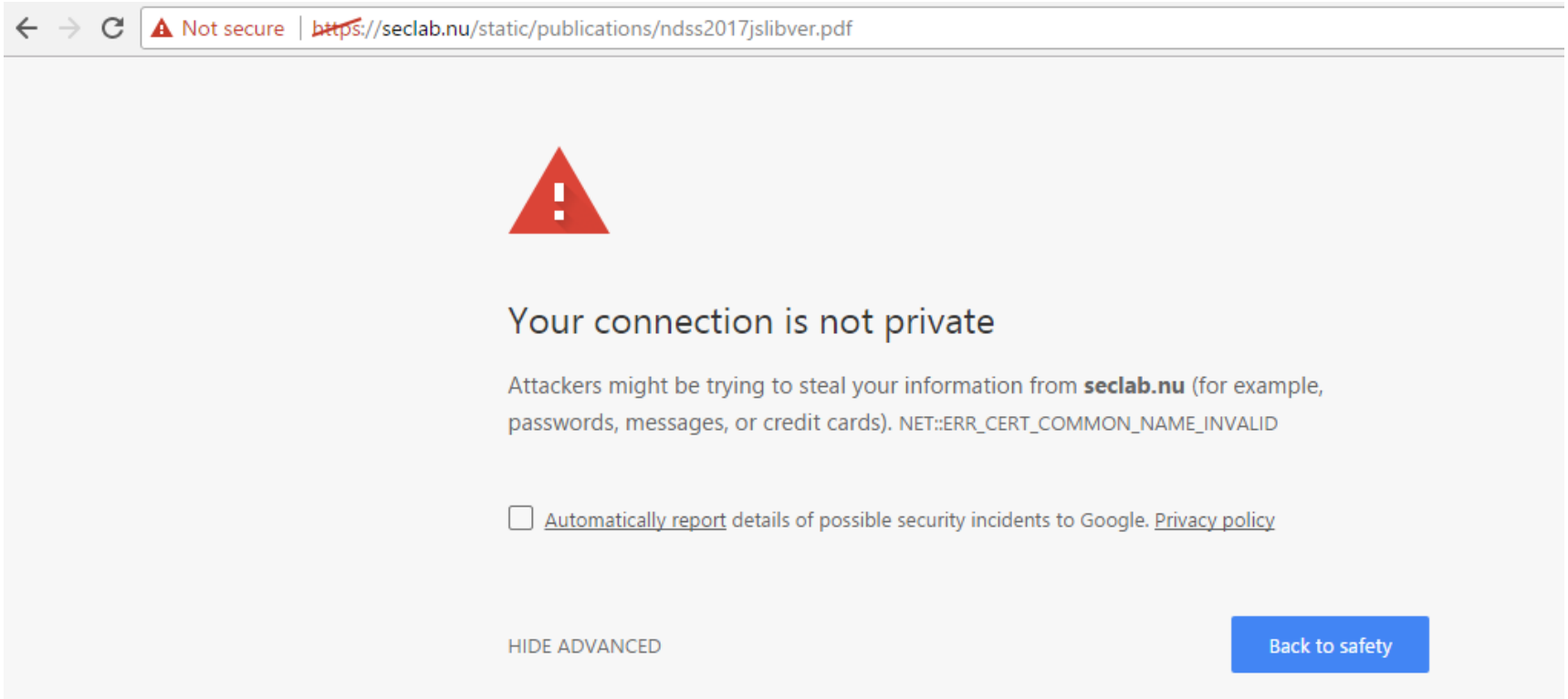
- Experience level:
  - High if python is your job, or programming in python > 5 years
  - Did not matter on any metric
- Security background:
  - **Almost** mattered to security results
  - Helps with usability reports
- Library experience: maybe helps usability

Takeaways

- Implementing crypto is (still) hard
- Simplified APIs **do** promote security
  - Sort of!
- Documentation, full-featured-ness are key!

What else can go wrong?

Example from today's readings



Other Developer Concerns

- AWS (or other) access tokens
  - Don’t commit them to GitHub
- Credentials for MySQL, etc.
  - Don’t leave them in web-accessible directories (in case PHP crashes)
  - Don’t let people pick them
  - Don’t let them be spit out by verbose error messages

Other Developer Concerns

- Don’t keep legacy databases around
  - bcrypt vs. MD5
- Don’t allow password access for SSH
- Don’t allow remote access to your database
- Don’t use outdated Javascript libraries for your website

Configuring HTTPS

What can go wrong?

- Hacking Team *was* a consulting company that contracted with governments
- Many operational security errors
- Sys admin's password: P4ssword

http://pastebin.com/raw/0SNSvyjJ