UNIVERSITÄT BONN

Deception Task Design in Developer Password Studies:

Exploring a Student Sample

Alena Naiakshina **Anastasia Danilova**

Christian Tiefenau Matthew Smith

Symposium on Usable Privacy and Security (SOUPS), Baltimore, MD, USA 2018

Motivation

- Design of end-user studies
  - Ample experience
  - Best-practice knowledge
- Lacking knowledge for developer studies
- Further insights into why developers struggle with end-user password storage

Motivation

- Design of end-user studies
  - Ample experience
  - Best-practice knowledge

- Lacking knowledge for developer studies → **Meta-level**

- Further insights into why developers struggle with end-user password storage

Motivation

- Design of end-user studies
  - Ample experience
  - Best-practice knowledge

- Lacking knowledge for developer studies → **Meta-level**

- Further insights into why developers struggle with end-user password storage → **Primary study**

Meta-level

UNIVERSITÄT BONN

- Qualitative vs. quantitative approach
  - Extended study:
    Why Do Developers Get Password Storage Wrong? A Qualitative Usability Study (CCS'17)
- Deception
  - Primed group: request to store end-user passwords securely
  - Fahl et al.[1] found no significant difference within the priming conditions in an end-user password study.
- Task length

[1] Fahl et al. On the ecological validity of a password study. (SOUPS'13)

Qual vs. Quant Setup

1. Hypotheses (researcher 1)

2. 67 participants registered for the study

3. Condition assignment: framework & priming
(balanced for programming experience)

4. 40 participants took part in the study

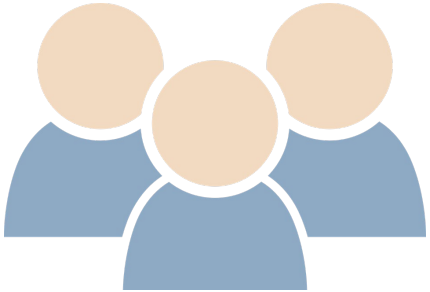
5. Study
- Implementation task: 8 hours
- Survey
- Interview

- First 20 used for **qualitative analysis**
- Analysis by researchers 2 & 3
- Published at CCS'17

- All 40 used for **quantitative study**
- Analysis by researchers 1-4

Demographics

- 40 participants
  - 6 female, 31 male, 3 prefer not to say
- Students
  - 33 Computer Science, 6 Media Informatics, 1 other
  - 12 BSc, 26 MSc Students, 2 other
- Age: mean 25

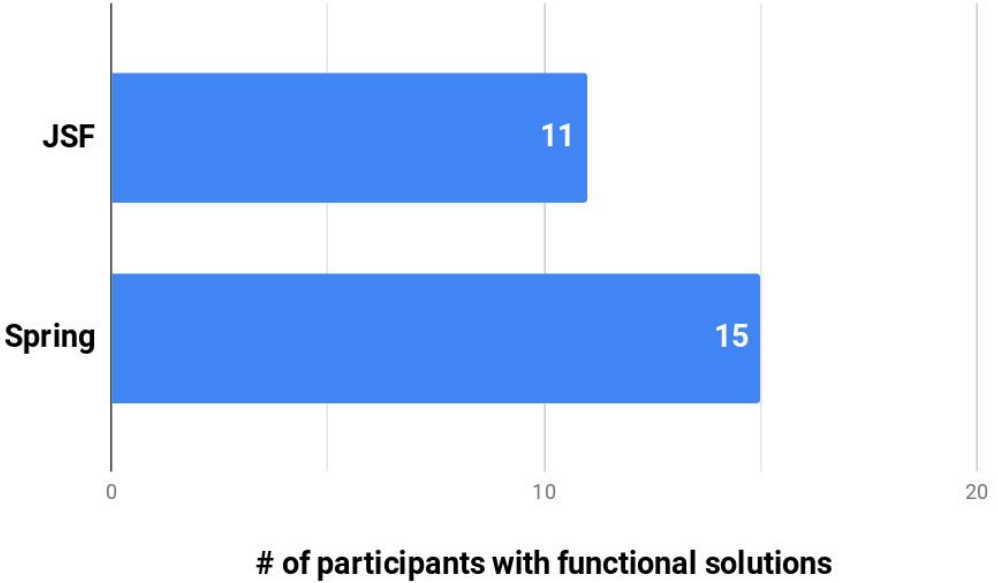


UNIVERSITÄT BONN

Results - Primary study

Framework → functionality

- H - **Framework** has an effect on the likelihood of **achieving functional solutions**.

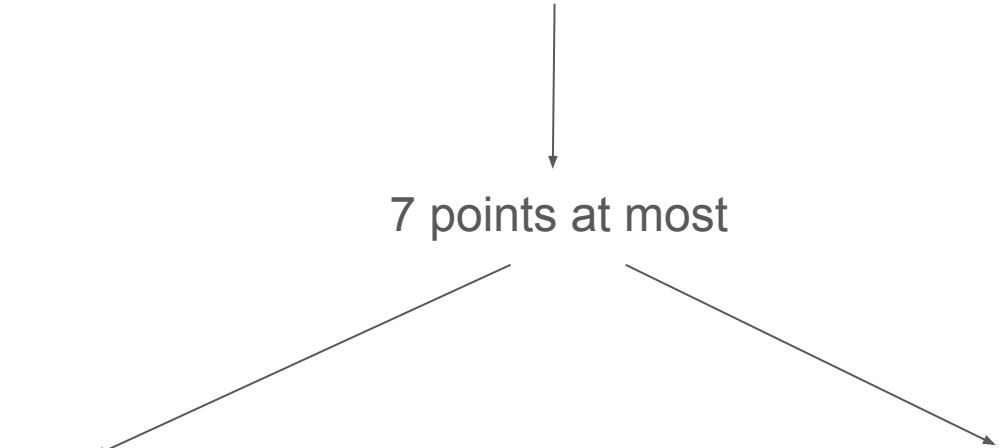


→ *Not statistically significant*

→ *We only had a power of 0.17, so this effect is worth looking at in follow-up studies*

Framework → security

- H - **Framework** has an effect on the **security score** of participants attempting security.



Min **2**, Median **5.5**, Mean **4.3**, Max **6**

Min, Median, Mean, Max **6**

Framework → security

- H - **Framework** has an effect on the **security score** of participants attempting security.

→ *Statistically significant before multiple testing correction*

→ *We think it is likely that a larger sample would confirm the trend that Spring participants earn higher scores than JSF participants.*

Java experience

UNIVERSITÄT BONN

- Acar et al.[1] → programming language experience has an effect on the security of participants' solutions

- H - Years of **Java experience** have an effect on the **security scores**.

  → *Not statistically significant in our study*

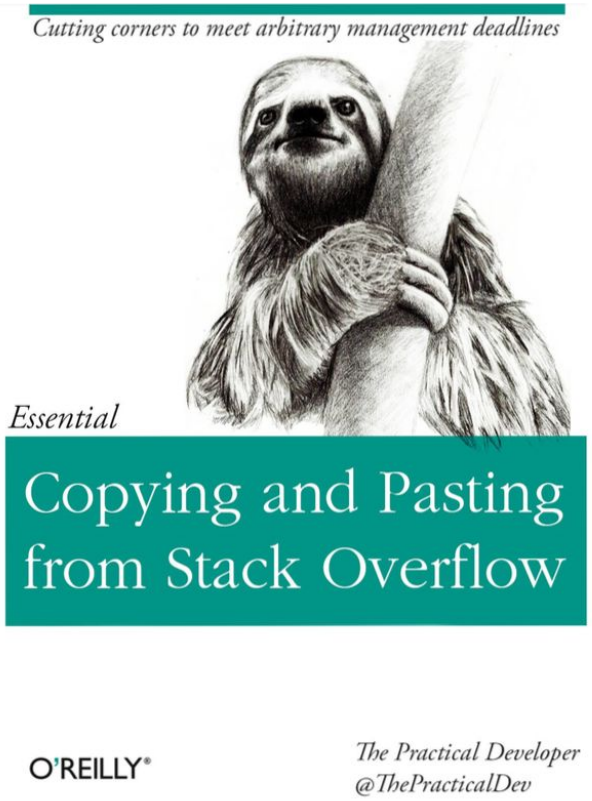
  → *Our student sample had a much smaller range of programming experience*

  → *Suggests that it might not be necessary to balance programming experience when working with **students**.*

[1] Acar et al. Security developer studies with github users: Exploring a convenience sample. (SOUPS'17)

Copy and paste

- Previous work:
  - “Because **Stack Overflow contains many insecure answers, Android developers** who rely on this resource **are likely to create less secure code**.” [1]
  - “We show that 196,403 (15%) of the 1.3 million **Android applications contain vulnerable code snippets** that were very **likely copied from Stack Overflow**.” [2]



[1] Acar et al. You get where you’re looking for: The impact of information sources on code security. (SP’16)
[2] Fisher et al. Stack overflow considered harmful? The Impact of Copy & Paste on Android Application Security. (CoRR abs’17)

Copy and paste

Our results:

- **Significant positive effect** of copy/paste events

- All secure solutions came from participants who copied and pasted security code

- **0%** of participants who did not copy/paste created a secure solution

UNIVERSITÄT BONN

Results - Meta-level

Priming hypotheses

H - **Priming** has an effect on the likelihood of participants **attempting security**.

Primed group	Non-Primed group
14/20	**2**/20

→ *Statistically significant*

Priming hypotheses

H - **Priming** does not have an effect on **achieving a secure solution** once the attempt is made.

Primed group	Non-Primed group
12/14	**0**/2

→ *Statistically significant before multiple testing correction*

Task length

- Short tasks
  - Common in previous work, straight forward, feasible
  - No distraction tasks, i.e., clear focus on security
- One-day time frame
  - Allows distraction tasks
  - 8 hours: longest time we could reasonably ask participants to remain in a lab setting
  - Full-screen capture, history of all code, copy/paste events, website history etc.
- Multi-day time frame
  - Trade-off between ecological validity and the ability to gather high-fidelity data

Task design

UNIVERSITÄT BONN

- Priming task
  - **14**/20 interacted with **security libraries/APIs**
  - 6/20 did not attempt to add security → no interaction with security libraries/APIs

→ Shorter, API usability focused study

→ Discover **usability problems of security APIs**

Task design

UNIVERSITÄT BONN

- Deception task
  - 2/20 interacted with security libraries/APIs
  - **18**/20 did **not attempt** to add security

    → **no interaction with security libraries/APIs**

→ Why do developers **not add security without study countermeasures** or **being prompted**.

→ More work needed to validate the ecological validity of deception in this context.

Deception task

- **2** attempted but failed
- **2** erroneously thought it was secure
- **2** security not part of the task
- **3** functionality more important
- **8** were not aware
- **3** no reason

- Realistic behavior

- Real world
  - Many password database compromises

Laboratory setting advantages

- You can monitor what participants google
  - 20/38 used Google when answering the survey
  - 6 framework-related topics
  - 14 password storage-related topics

- **4**/16 **non-primed participants without security attempts** searched **how to store user passwords securely** while answering the survey

- **7**/12 **primed participants** with secure solutions searched for **additional password storage security details** to answer questions of the survey

Quantitative vs. Qualitative

UNIVERSITÄT BONN

- Compared two frameworks with A/B test.
- We are confident that deception changes the behavior of participants dramatically.

Already...

- ...highlighted many of the problems faced by developers.
- ...delivered a good indication of priming effect.

We did not find much to add to the conclusions of the

qualitative study!

Study design

- Many valuable insights can be gained without the need for larger sample sizes.
- Recruitment of participants biggest challenge.
- The extra 20 participants did not add much in the way of insights.

→ We recommend doing **qualitative studies** in order to investigate the **usability of APIs!**

Take-aways

Priming / Deception

Allows us to study two completely different aspects:

API usability vs. **security awareness**

Google

Our participants googled survey questions

Beware of knowledge questions!

Copy/Paste

Positive effect on the security of our participants' code

UNIVERSITÄT BONN

Do more **qualitative** developer studies in lab!