Let's Hash
Helping Developers with Password Security

Lisa Geierhaas*
University of Bonn

Anna-Marie Orloff
University of Bonn

Matthew Smith
University of Bonn, Fraunhofer FKIE

Alena Naiakshina
Ruhr-University Bochum
Background

2016 IEEE Symposium on Security and Privacy

You Get Where You’re Looking For
The Impact of Information Sources on Code Security

Yasemin Acar, Michael Backes, Sascha Fahl, Doowon Kim†, Michelle L. Mazeuke‡, Christian Stransky
CISPA, Saarland University; †University of Maryland, College Park

A Password-Storage Field Study with
Freelance Developers

Alena Naiakshina
University of Bonn
naiakshina@cs.uni-bonn.de

Anastasia Danilova
University of Bonn
danilova@cs.uni-bonn.de

Eva Gerlitz
University of Bonn
gerlitz@uni-bonn.de

Emanuel von Zezschwitz
University of Bonn, Fraunhofer FKIE
zeugschwitz@cs.uni-bonn.de

Matthew Smith
University of Bonn, Fraunhofer FKIE
smith@cs.uni-bonn.de
Background

You Get Where You’re Looking For
The Impact of Information Sources on Code Security

Yasemin Acar, Michael Backes, Sascha Fahl, Doowon Kim¹, Michelle L. Mazurek¹, Christian Stransky
CISPA, Saarland University; ¹University of Maryland, College Park

Usable resources
=! secure code

Secure solutions
via copy-paste

"If you want, I can store the encrypted password."
A Password-Storage Field Study with Freelance Developers

Alena Naiakshina
University of Bonn
naiakshi@cs.uni-bonn.de

Anastasia Danilova
University of Bonn
danilova@cs.uni-bonn.de

Eva Gerlitz
University of Bonn
gerlitz@uni-bonn.de

Emanuel von Zeischwitz
University of Bonn, Fraunhofer FKIE
zeischwitz@cs.uni-bonn.de

Matthew Smith
University of Bonn, Fraunhofer FKIE
smith@cs.uni-bonn.de
Coding Resources

Usable?

Usable and secure!
Let’s Hash

Multiple languages

Code fragments ready to use

Different security-sensitive topics

Storage

```
import bcrypt

def hash_password(password):
    # bcrypt.gensalt() takes an argument (in the form of eg (round
    # which increases security, but also the cost factor
    pw_hash = bcrypt.hashpw(password.encode(), bcrypt.gensalt())
    return pw_hash

def verify(pw_hash, password):
    return bcrypt.checkpw(password.encode(), pw_hash)
```
Let’s Hash “Wizard”

Wizard-like UI with guiding questions

What are you looking for?

- Language
  - Python
  - Java

- Algorithm for hashing:
  - BCrypt (This is the simplest option. It is secure for most cases and does not require fine tuning.)
Developer Study

Participants from Freelancer.com

Three programming tasks + survey

n=179
Developer Study

Three groups:

LH: Let’s Hash

LH-W: Let’s Hash (with wizard)

Control group (commonly used resources)
Results

Secure solutions by task and group
Results – Error Types

Security error types by task and group
Results - Feedback

- Trustworthy, because it is not a forum post
- Easier to use than other resources
- I would use it again
Let’s Hash
Helping Developers with Password Security

Lisa Geierhaas*
University of Bonn

Anna-Marie Ortlloff
University of Bonn

Matthew Smith
University of Bonn, Fraunhofer FKIE

Alena Naiakshina
Ruhr-University Bochum

• Introduced Let’s Hash: A website to help developers with tasks around password security
• Developer study (n=179): Developers using Let’s Hash had a significantly higher chance of producing secure code
• Resources that are usable and secure can significantly improve code quality

https://www.letsauth.dev