開発者向け
ユーザブルセキュリティ研究の動向

2022/10/24 ユーザブルセキュリティワークショップ

日本電信電話株式会社 社会情報研究所

金井 文宏

自己紹介：金井 文宏（Fumihiro Kanei）

- 所属/役職
  - 2015年 日本電信電話株式会社 入社
  - 社会情報研究所 所属　（研究員）
- 出身大学
  - 横浜国立大学
    修士: 2015卒, 博士: 2021卒
- 研究分野
  - ユーザブルセキュリティ
    - › 開発者向けユーザブルセキュリティ
      CSS2020 最優秀論文賞, ACSAC2021 採択
  - モバイルセキュリティ
    - › Androidアプリ/マルウェアの解析
  - Webセキュリティ
    - › 広告不正検知

本日の内容

- 開発者向けユーザブルセキュリティ研究の概要
- 主な研究トピック
- 開発者向けユーザ調査を行うにあたって

本日の内容

- 開発者向けユーザブルセキュリティ研究の概要
- 主な研究トピック
- 開発者向けユーザ調査を行うにあたって

開発者向けユーザブルセキュリティ研究とは？

- ユーザブルセキュリティ研究：
  - **人間的側面**からセキュリティ技術を検討する研究分野
  - ユーザ属性とコンテキストの組み合わせによる様々な検討領域が存在

非専門家（Non-expert）
エンドユーザ など

専門家（Expert）
ソフトウェア開発者
システム管理者
SOCアナリスト
CISO など

開発者向けユーザブルセキュリティ研究

ユーザブルセキュリティ研究のうち、対象とするユーザ属性として**開発者**に着目した研究領域

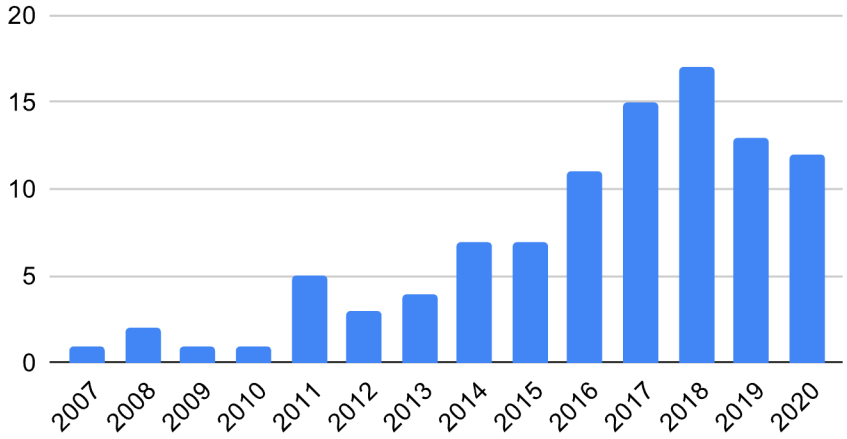
なぜ「開発者」に着目する必要があるか？

- システムの設計・実装段階でセキュリティ対策を講じる事で脆弱性に起因する被害を未然に防ぎたい
- 性能的に優れたセキュリティ技術でも現場の**開発者視点で効果や利便性**が高くないと活用は困難
  - 例：とても高精度な脆弱性検査ツールらしいけど、
    ウチの環境だと導入しづらい…
    どれくらい効果があるかわからないから導入判断が難しい…

開発者の認識/行動を分析し理解することで、
開発者にとって利用しやすいセキュリティ対策技術を創出したい

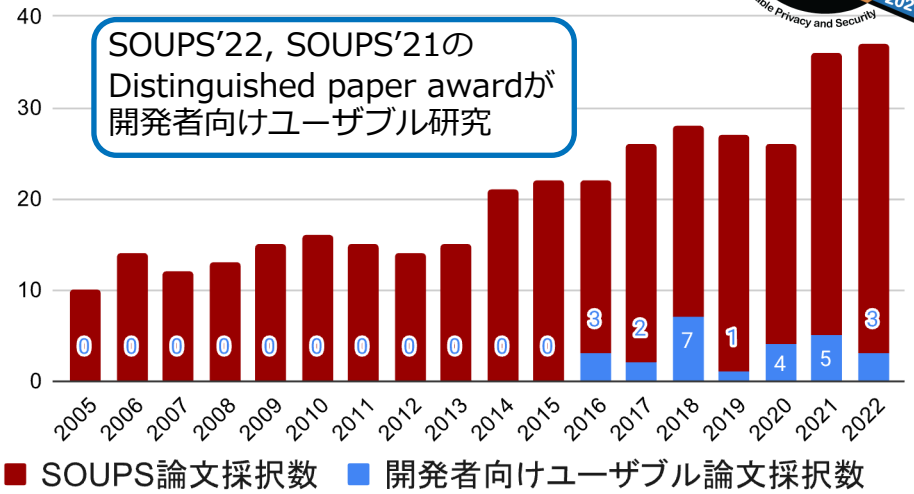
論文数から見る
開発者向けユーザブルセキュリティ研究

開発者向けユーザブルセキュリティ研究に関する論文件数*



2016年頃から急速に論文件数が増加

SOUPSにおける開発者向けユーザブルセキュリティ論文

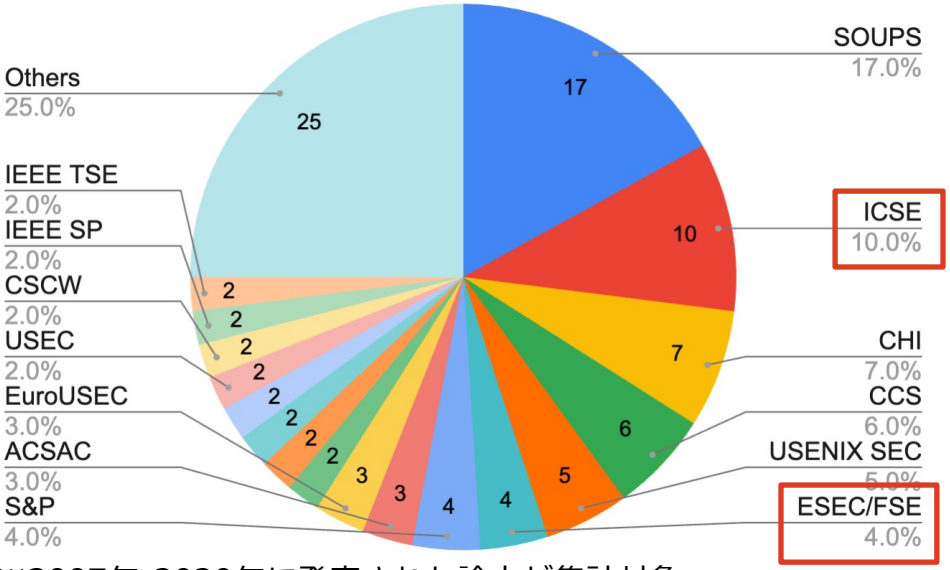


近年のSOUPS論文における1～2割が開発者向けユーザブルセキュリティ研究

*参考元：Mokhberi et al. *SoK : Human , Organizational , and Technological Dimensions of Developers ' Challenges in Engineering Secure Software*,（EuroUSEC'21）
Kaur et al. *Where to Recruit for Security Development Studies: Comparing Six Software Developer Samples*,（SEC'21）
Tahaei et al. A *Survey on Developer-Centred Security*,（EuroUSEC'19）

国際会議ごとに見てみると...

国際会議 / 論文誌ごとの開発者向け
ユーザブルセキュリティ論文の発表件数*



※2007年-2020年に発表された論文が集計対象

- **ソフトウェア工学分野**でも
  ユーザブルセキュリティに関連する
  論文が増加中
  - ICSE, ESEC/FSE, ASE（SW工学分野Tier1会議）で、数年前からHCIのセッションが新設
- **IEEE SecDev** が2016年から開始
  - セキュアなシステム開発に関する研究に特化した国際会議

*参考元：Mokhberi et al. *SoK : Human , Organizational , and Technological Dimensions of Developers ' Challenges in Engineering Secure Software*,（EuroUSEC'21）
Kaur et al. *Where to Recruit for Security Development Studies: Comparing Six Software Developer Samples*,（SEC'21）
Tahaei et al. A *Survey on Developer-Centred Security*,（EuroUSEC'19）

本日の内容

- 開発者向けユーザブルセキュリティ研究の概要
- 主な研究トピック
- 開発者向けユーザ調査を行うにあたって

主な研究トピック

- 開発プロセスの理解 / セキュリティ阻害要因の特定
- 暗号API / 静的解析ツール
- 開発者が参照する情報とセキュリティ

主な研究トピック：

開発プロセスの理解 / セキュリティ阻害要因の特定

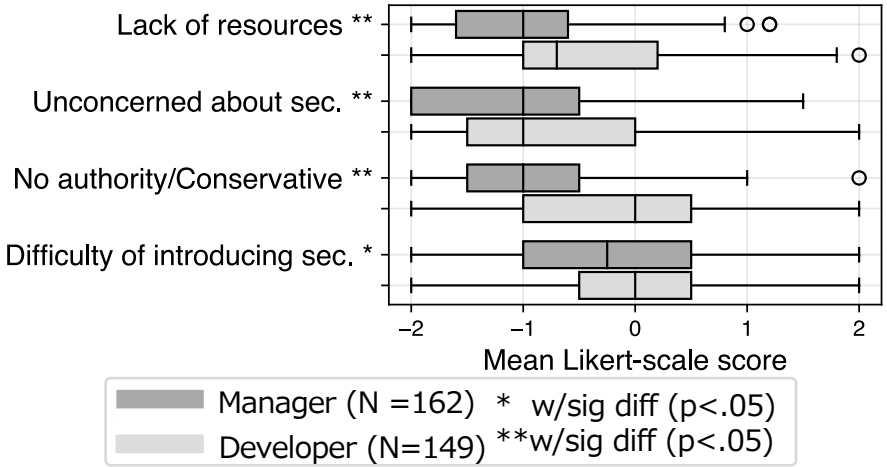
- 開発者が**セキュアな開発を行うためのプロセスの理解**や、**セキュアな開発を妨げている要因**の特定を目的に、開発者の認識・行動を調査する研究

- 研究例
  - 開発時のセキュリティ実践内容とベストプラクティスの乖離を分析（SOUPS'18）
  - セキュア開発を行うモチベーション・戦略・阻害要因を調査（CHI'19）
  - デベロッパ/マネージャ双方の視点からセキュア開発を妨げる要因を調査（ACSAC'21）
  - Ethnographic studyにより開発現場でセキュリティの文化が形成されていくプロセスを分析（SOUPS'21） ※**Distinguished paper award**

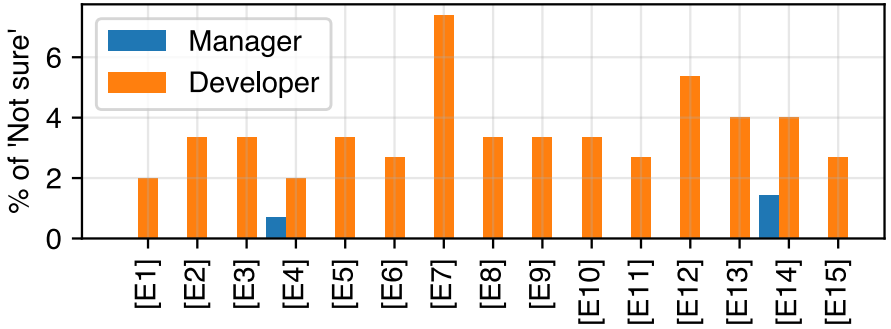
Assal et al. *Security in the Software Development Lifecycle*（SOUPS'18）
Assal et al. *"Think secure from the beginning": A Survey with Software Developers*（CHI'19）
Kanei et al. *A Cross-role and Bi-national Analysis on Security Efforts and Constraints of Software Development Projects* (ACSAC'21)
Tuladhar et al. *An analysis of the role of situated learning in starting a security culture in a software company* (SOUPS'21)

論文紹介： セキュア開発の阻害要因（ACSAC'21）

- デベロッパ/マネージャ双方に対するアンケート調査により、
  セキュア開発を妨げる要因を定量的に調査・比較



セキュリティ阻害要因に対する回答のスコア
全く同意できない (-2) -強く同意できる (+2)



セキュリティ対策 [E1-E15] の実施状況に関する質問に
「分からない（Not sure）」と回答した参加者の割合

⇛ デベロッパ/マネージャの間に**セキュリティに対する認識・行動のギャップが**存在
e.g., デベロッパには意思決定権が無い など

Kanei et al. *A Cross-role and Bi-national Analysis on Security Efforts and Constraints of Software Development Projects* (ACSAC'21)

セキュア開発の実践を妨げる要因の例

- **開発者の課題**
  - スキル・経験の不足
  - セキュリティに対する誤解・楽観
  - 機能実装を優先（セキュリティは非機能要件）
- **技術の課題**
  - 暗号API/ライブラリの誤った利用
  - 利用しやすい脆弱性検査ツールの欠如
- **組織の課題**
  - リソース不足（時間、予算、人）
  - セキュリティ文化の欠如
  - ガイドライン/ポリシーの欠如
  - 管理/経営層の理解・サポート不足
  - チーム間のコミュニケーション

他にどのような課題があるか？
調査が不十分な領域は？

参考：Mokhberi et al. *SoK : Human , Organizational , and Technological Dimensions of Developers ' Challenges in Engineering Secure Software* (EuroUSEC'21)

主な研究トピック：

暗号API / 静的解析ツール

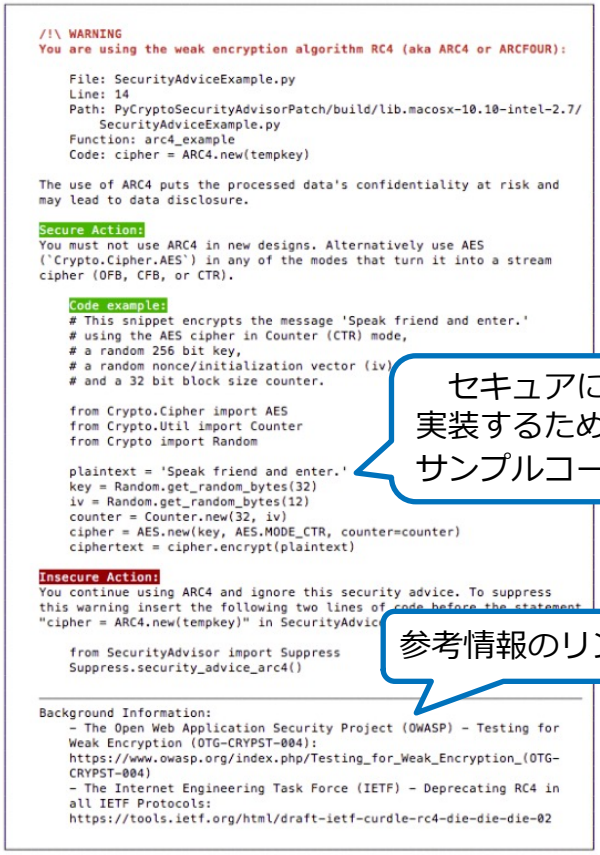
- 開発者が利用する**暗号API、静的解析ツール**などソフトウェアの
  セキュリティに関わる各種ツール/技術のユーザビリティに着目した研究

- 研究例
  - Pythonの有名な暗号化ライブラリ５種のユーザビリティ評価 (S&P17)
  - 暗号APIの適切な利用方法に関するアドバイスを提示する環境を提案し評価（SOUPS'18）
  - 静的解析ツール（Clang SA, Libfuzzer）のユーザビリティを評価（SOUPS'21）
  - 静的解析ツールの出力内容が脆弱性を修正する上でどれくらい効果的か調査（CHI'21）

Acar et al. *Comparing the Usability of Cryptographic APIs* (S&P17)
Gorski et al. *Developers Deserve Security Warnings, Too: On the Effect of Integrated Security Advice on Cryptographic API Misuse* (SOUP'18)
Plöger et al. *A Qualitative Usability Evaluation of the Clang Static Analyzer and libFuzzer with CS Students and CTF Players* (SOUPS'21)
Tahaei et al. *Security Notifications in Static Analysis Tools: Developers' Attitudes, Comprehension, and Ability to Act on Them* (CHI'21)

論文紹介：暗号APIのデザイン（SOUPS'18）

- 暗号APIを正しく利用するために効果的なセキュリティアドバイス表示を検討
  - 暗号APIの誤った利用を検知し、利用方法に関する効果的なセキュリティアドバイスを表示する開発環境を提案
  - 提案環境により正しく暗号APIを利用できる開発者の割合が有意に増加（26%→50%）



セキュリティアドバイスの例

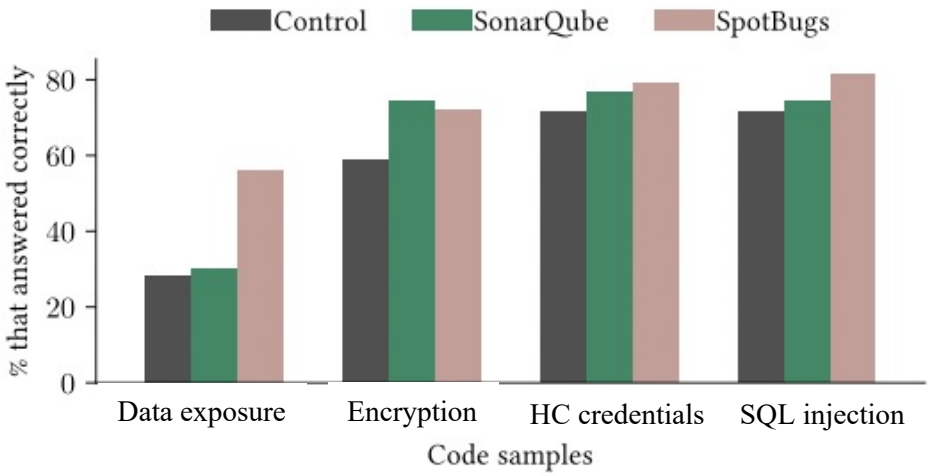
Gorski et al. *Developers Deserve Security Warnings, Too: On the Effect of Integrated Security Advice on Cryptographic API Misuse* (SOUP'18)

論文紹介：静的解析ツールの出力内容 (CHI'21)

- 既存の静的解析ツールの出力内容が、脆弱性を修正する上でどれくらい効果的であるかを調査
  - 静的解析ツールの出力を見た場合/見なかった場合に、脆弱なサンプルコードを正しく修正できるか比較
  - 静的解析ツールの出力を見ても67%～86%の開発者が少なくとも1つ以上のサンプルコードの修正方法を誤って回答した

⇛ **出力内容の見せ方に改善の余地**あり
e.g., フレーズ選び、情報の構造化 など

静的解析ツールにより正答率は全体的に向上するが、統計的に有意な変化は一部のみ



脆弱性の修正方法を正しく回答できた参加者の割合

Tahaei et al. *Security Notifications in Static Analysis Tools: Developers' Attitudes, Comprehension, and Ability to Act on Them* (CHI'21)

主な研究トピック：

開発者が参照する情報とセキュリティ

- **開発者が参照する情報の安全性**（＝それらを参照するとソフトウェアがどれくらい安全に/脆弱になりやすいか）に着目した研究

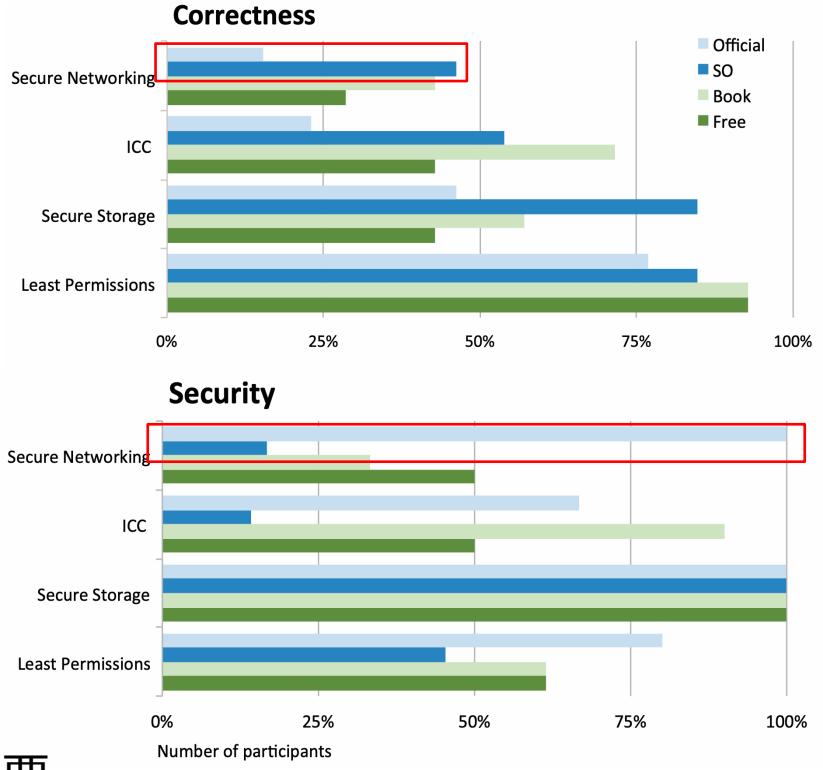
- 研究例
  - 開発者が参照する情報ソースとソフトウェアの機能性/安全性の関係を調査（S&P'16）
  - Stack Overflow上のスニペットに警告ナッジを表示するシステムを提案し評価（SEC'19）
  - Web検索で脆弱なスニペットを表示されにくくするランク付け手法を提案し評価（CCS'21）
  - 開発者にとって利用しやすくかつ安全な情報を提示するシステムLet's Hashを提案し評価（SOUPS'22） **※Distinguished paper award**

Acar et al. *You Get Where You're Looking For The Impact of Information Sources on Code Security*（S&P'16）
Fischer et al. *Stack Overflow Considered Helpful ! Deep Learning Security Nudges Towards Stronger Cryptography* (SEC'19)
Fischer et al. *The Effect of Google Search on Software Security: Unobtrusive Security Interventions via Content Re-ranking* (CCS'21)
Geierhaas et al. *Let's Hash: Helping Developers with Password Security* (SOUPS'22)

論文紹介：情報ソースの比較（S&P'16）

- 参照する情報ソースを制限した状態で参加者にプログラミングタスクを行ってもらうラボ実験を実施
  - 情報ソース：公式ドキュメント, 書籍, Stack Overflow, 自由（何でも可）
- 成果物の機能性/安全性を比較
  - Stack Overflowを参照すると機能的に正しく実装しやすい反面、脆弱になりやすい
  - 公式ドキュメントを参照すると安全に実装しやすい反面、機能的に正しく実装しにくい

⇛ **公式ドキュメントのユーザビリティ向上**が必要



機能を**正しく**実装できた参加者の割合（上）と
機能を**安全に**実装できた参加者の割合（下）

Acar et al. *You Get Where You're Looking For The Impact of Information Sources on Code Security*（S&P'16）

論文紹介：適切な情報選択の補助（SEC'19）

- Stack Overflow上のコードスニペットに対して安全性に関するナッジ（警告）を付与するシステムを提案し評価

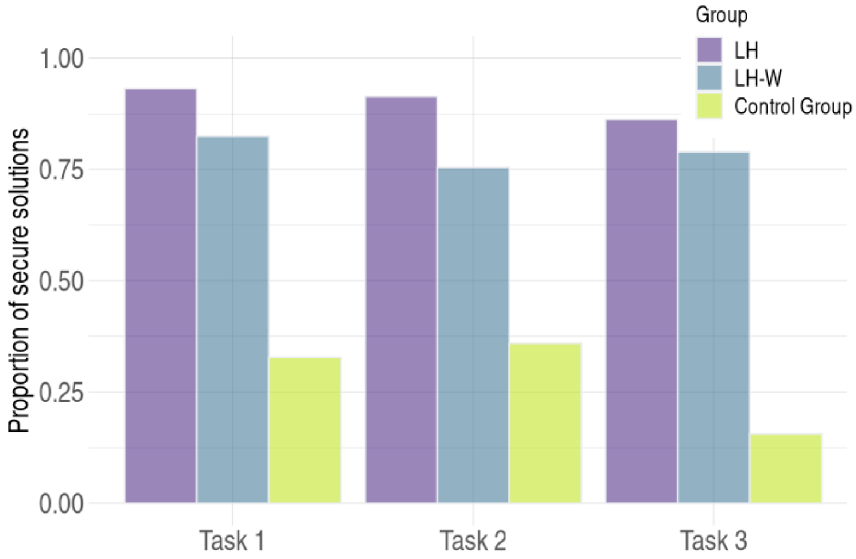
Stack Overflowのスニペット上に表示されるナッジ

- スニペットが Secure or Insecure どちらかを機械学習ベースの手法で判定（AUC-ROC: 0.992）
- 判定結果に基づき、Stack Overflow上のスニペットにナッジを表示
- 当該システムを利用することでセキュアに機能を実装できる開発者の割合が有意に増加

Fischer et al. *Stack Overflow Considered Helpful ! Deep Learning Security Nudges Towards Stronger Cryptography* (SEC'19)

論文紹介： 安全かつ利用しやすい開発リソース（SOUPS’22）※Distinguished paper award

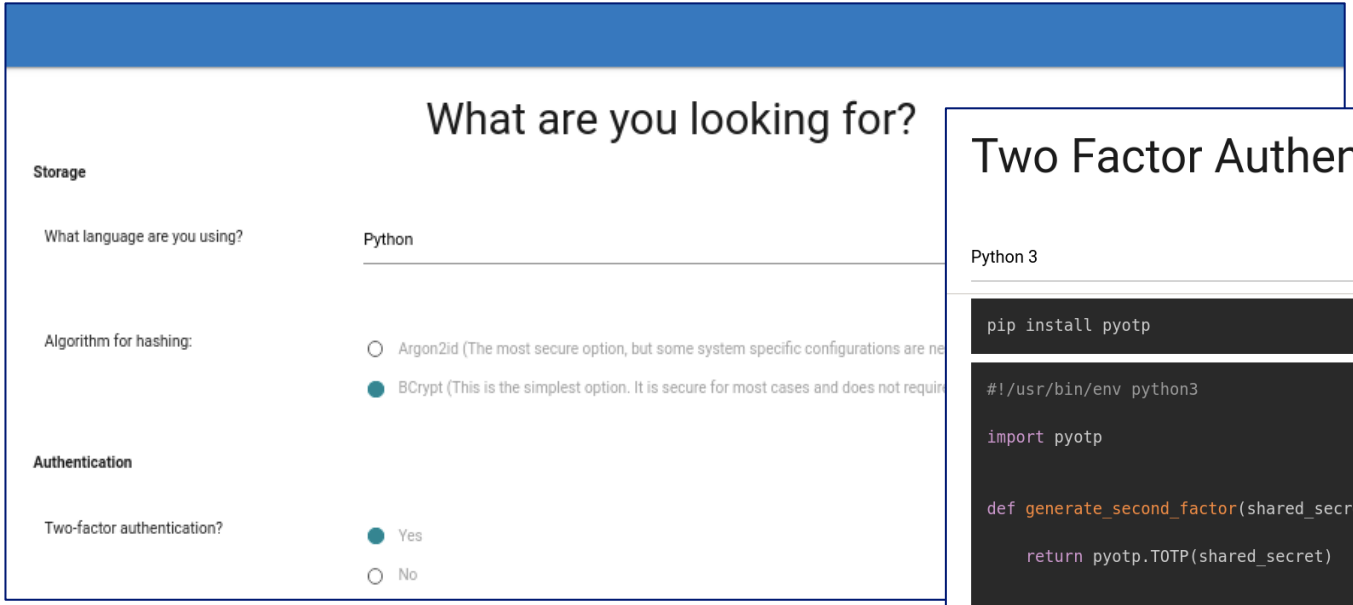
- パスワードに関する機能をセキュアに実装可能かつ Ready-to-Useなスニペットを提示するシステム Let’s Hash を提案
- デザインのコンセプト：
  「Stack Overflowのように簡単に使えて
  公式ドキュメントのように安全に実装可能」
- Let’s Hashを利用することで、
  セキュアにパスワード機能を
  実装できる確率が有意に増加
  （5倍～32倍）

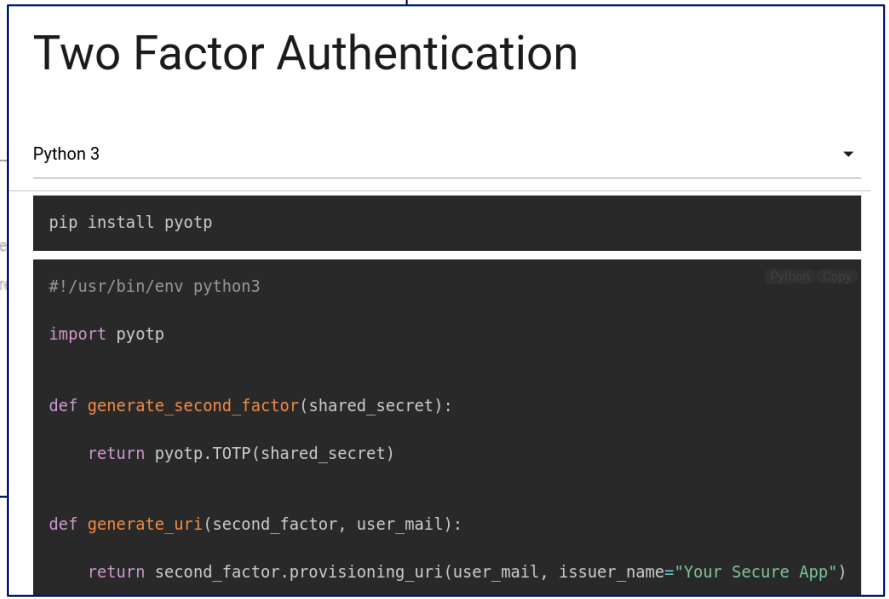


プログラミングタスクにおいて
セキュアな実装ができた開発者の割合

Geierhaas et al. *Let’s Hash: Helping Developers with Password Security* (SOUPS’22)

論文紹介： 安全かつ利用しやすい開発リソース（SOUPS’22）※Distinguished paper award





1. 実装したい機能に関する質問を提示
2. Ready-to-useなスニペットを表示

Geierhaas et al. *Let’s Hash: Helping Developers with Password Security* (SOUPS’22)

その他の研究トピック

- **Security Champions / Advocates**（e.g., セキュリティ提唱者への質的調査）
  - Haney et al. "It’s Scary. . . It’s Confusing. . . It’s Dull": How Cybersecurity Advocates Overcome Negative Perceptions of Security (SOUPS’18)
  - Tahaei et al. Privacy Champions in Software Teams: Understanding Their Motivations, Strategies, and Challenges (CHI’21)
- **開発者の教育/啓発**（e.g., セキュア開発に関する効率的な教育方法を提案/評価）
  - Weir et al. Light-Touch interventions to improve software development security (SecDev’18)
  - Weir et al. Interventions for Software Security: Creating a Lightweight Program of Assurance Techniques for Developers（ICSE-SEIP’19）

本日ご紹介した以外にも開発者向けユーザブルセキュリティに関する様々なトピックが研究されていますので、冒頭にご紹介した国際会議の論文を是非チェックしてみて下さい

本日の内容

- 開発者向けユーザブルセキュリティ研究の概要
- 代表的な研究トピック
- 開発者向けユーザ調査を行うにあたって

開発者向けユーザ調査における注意点

- エンドユーザ向けユーザ調査における基本的な注意点は開発者を調査対象にした場合でも同様
  - 適切な属性を持つ参加者の選定
  - 実験環境の一定化
  - 不良回答への対応
  - 各種バイアスの軽減
  - Ecological Validity
  - 倫理的配慮　　など

適切な属性を持つ参加者の選定

- 開発者向けユーザ調査では**開発者特有のユーザ属性**も考慮した調査設計/参加者募集が必要

一般的なユーザ属性の例

- 年齢
- 性別
- 母国語
- 教育的バックグラウンド
- ハンディキャップ　など

開発者特有のユーザ属性の例

- 立場：プロ, フリーランス, 学生
- プログラミングスキルの有無
- 役職：デベロッパ、マネージャ…
- 業務内容：実装、テスト…
- 所属企業：大企業、中小企業

例えば下記のような調査設計は Ecological Validityに問題があるとされる

- プロ向けの開発サポートツールの評価に大学生を募集

開発者向けユーザ調査における注意点：

開発者のリクルーティング

- 一般的にエンドユーザと比較して開発者のリクルーティングは難しい
  - スキルのある参加者を募集しづらい、コンタクト先が少ない、費用がかかりやすい、など

- 既存研究で用いられている開発者のリクルーティング方法 [SEC'22]
  - 研究者のコネクション活用（e.g., 個人的な連絡先、スノーボールサンプリング、MLでのCS学生募集）
  - 有料サービス（e.g., Prolific, Upwork, Freelancer）
  - ソーシャルメディア（e.g., Twitter, Facebook Groups）
  - オンラインフォーラム/ブログ (e.g., Reddit)
  - ネットワーキング（e.g., LinkedIn）
  - メール募集（e.g., Github, Google Play）

（過去にやってる研究はあるが）
GitHubやGoogle Play 経由の開発者の募集はサービスの利用規約違反のため非推奨 [CHI'22]

Tahaei et al. *Recruiting Participants With Programming Skills: A Comparison of Four Crowdsourcing Platforms and a CS Student Mailing List* (CHI'22)
Kaur et al. *Where to Recruit for Security Development Studies from: Comparing Six Software Developer Samples* (SEC'22)

論文紹介：開発者のスクリーニング (ICSE'21)

- プログラミングスキルがある開発者を募集するためのスクリーニング質問を設計

- 提案されたスクリーニング質問により、以下の参加者を区別できるか評価

1. プログラミングの講義を受けた学生 + 大学教授
   = プログラミングスキルのある人

2. 行動経済学科の学生（プログラミング経験無し）
   + Clickworker で開発経験無しと申告している人
   = プログラミングスキルのない人

⇛ 1と2のグループ間で正答率に有意な差が見られた

```
main{
    print(func("hello world"))
}

String func(String in){
    int x = len(in)
    String out = ""
    for (int i = x -1; i >= 0; i--){
        out.append(in[i])
  }
  return out
}
```

簡単なコード例を提示

Q. What is the parameter of the function?

- String out
- String in
- I don't know
- int i = x-1; i>=0; i--
- Outputting a String
- int x = len(in)

プログラミングスキルのある人であれば正解を選択できるような質問を実施

提案されたスクリーニング質問（一部抜粋）

Danilova et al. *Do you really code? Designing and Evaluating Screening Questions for Online Surveys with Programmers* (ICSE'21)

開発者のリクルーティング：

論文紹介：募集チャネルの比較 (SEC'22, CHI'22)

- 開発者の募集チャネルごとに集まった参加者の開発経験・スキル、セキュリティ意識等がどのように異なるか比較

	CHI'22	SEC'22
比較対象	• MTurk • Prolific • Appen • Clickworker • CS学生	• MTurk • Prolific • Upwork • Freelancer • Google Play • CS学生
推奨する開発者の募集方法	• 少人数の調査の場合 ⇛ **CS学生**の募集を検討すべき • 大規模調査の場合 ⇛ **Prolific** での募集を検討すべき	• 少人数調査の場合 ⇛ **Upwork**での募集を検討すべき • 大規模調査の場合 ⇛ **MTurk**での募集を検討すべき

※ 両論文で実験設定や比較内容、推奨される募集方法が異なるので、両論文を熟読の上、自身の研究におけるRQ / 仮説などを踏まえて募集方法を決定する事を推奨します

Tahaei et al. *Recruiting Participants With Programming Skills: A Comparison of Four Crowdsourcing Platforms and a CS Student Mailing List* (CHI'22)
Kaur et al. *Where to Recruit for Security Development Studies from: Comparing Six Software Developer Samples* (SEC'22)

開発者向けユーザ調査の方法論：

調査設計にあたってその他に参考にすべき研究

- プログラミングタスクの説明を行う際の様々なDeception（e.g., 本来の実験目的を伝えるかどうか、タスクの依頼元を詐称するかどうか, etc.）が実験結果に与える影響を分析（CCS'17, SOUPS'18, CHI'19）
- セキュア開発に関する自己効力感（＝自分がどれくらいセキュアにソフトウェアを開発できると思うか）を図る指標SSD-SESを提案（CHI'20）
- オンラインでプログラミングタスク等のラボ実験を行うためのプラットフォームを提案（SOUPS'22）

既存研究を参考に確立された実績のある方法を採用すべき（我流はNG）

Naiakshina et al. *Why Do Developers Get Password Storage Wrong? A Qualitative Usability Study* (CCS'17)
Naiakshina et al. *Deception Task Design in Developer Password Studies: Exploring a Student Sample* (SOUPS'18)
Naiakshina et al. *"If you want, I can store the encrypted password": A Password-Storage Field Study with Freelance Developers* (CHI'19)
Votipka et al. *Building and Validating a Scale for Secure Software Development Self-Efficacy* (CHI'20)
Huaman et al. *If You Can ' t Get Them to the Lab : Evaluating a Virtual Study Environment with Security Information Workers* (SOUPS'22)

まとめ

- 開発者向けユーザブルセキュリティは、脆弱性に起因する被害の防止に向けた重要かつホットな研究分野

- 開発者向けユーザブルセキュリティにける主な研究トピック
  - 開発プロセスの理解 / セキュリティ阻害要因の特定
  - 暗号API / 静的解析ツール
  - 開発者が参照する情報とセキュリティ

- 開発者向けユーザ調査の際には開発者独自のユーザ属性を考慮すべき
- 既存研究を参考に確立された実績のある方法を採用すべき