# System and Network Security (S22CS8.403) Beamer Presentation

Research Paper : The Effect of Google Search on Software Security, ACM CCS 2021

Sudipta Halder [2021202011]
Nitin Kumar [2021202020]

Department of Computer Science and Information Security
International Institute of Information Technology, Hyderabad

March 10, 2022

# Importance of Google Search

- Google Search has become the way to go for Software Developers when they are stuck in a coding problem.
- Programmers tend to prefer programming forums like Stack Overflow, Quora more as compared to official documentation or books[1].
- People use Google Search(91%) more than directly using programming forums like Stack Overflow to look for answers(36%)[4].

# Security Concerns!!

- Results obtained in Google Search or other programming forums are not always secure!!
- one of the top three Google Search results has about one fourth chance of resulting into insecure code[2].
- About one third of the code examples related to cryptography to be insecure in Stack Overflow and apart from that they were also used in numerous apps(1.9 lakhs)[3].

# Difficulty in finding secure solutions

- Google does not have any direct indication whether a solution is secure or not.
- Finding secure solutions in Stack Overflow is also difficult.
- Stack Overflow does not have a indicator which tells this code snippet is secure or not.
- Most people generally pick the most voted answer in Stack Overflow without checking whether it is secure or not.
- Cryptographic code analysis tools like FixDroid, LGTM, CodeQL are available as plugin but then most of the users don't use them for their daily work.

# Online Surveys

- Two online surveys were performed.
- **Survey 1** with 192 people.
- **Survey 2** with 218 people.

| | | | | |
|---|---|---|---|---|
| | | **Age** | | |
| Mean = 32.03/30.94 | Median = 29.5/29 | Stddev = 10.34/8.59 | Min = 18 | Max = 74/59 |
| | | **Country of Origin** | | |
| USA = 40/60 | Germany = 23/19 | Brazil/India = 13/18 | China/Brazil = 11/9 | Other = 105/112 |
| | | **Gender** | | |
| Male = 171/188 | Prefer not to say = 5/22 | Other = 6/1 | | Female = 10/7 |
| | | **Level of Education Achieved** | | |
| High School = 30/21 | Bachelor= 80/102 | Master = 44/60 | PhD = 18/11 | Other = 20/24 |
| | | **Professional** | | |
| Yes = 133/161 | No = 52/33 | | | N/A = 7/24 |
| | | **Security Background** | | |
| Yes = 56/46 | No = 129/143 | | | N/A = 7/29 |
| | | **Java Years of Experience** | | |
| < 1 year = 51/61 | 1 to 2 years = 35/40 | > 2 years = 82/80 | | N/A = 25/37 |
| | | **Java Primary Focus of Job** | | |
| Yes = 31/49 | No = 158/144 | | | N/A = 3/25 |

Figure: Detailed data about demographics of participants for Study 1 (N = 192) and Study 2 (N = 218)[2]

# Agenda

- **Survey 1** focuses on the ratio of secure and insecure codes in top 10 Google Search results.
- **Proposed solution:** Using security based re-ranking and changing the ordering of the Google Search results in terms of security.
- **Survey 2** focuses on testing how this re-ranking performs in real life, how it is changing the ration of secure and insecure results in top 10 Google Search results and how it affects security of developed program.

# Choosing Dataset for labelling

- **Secure Webpage:** If top accepted answer in that webpage turns out to be secure.
- **Insecure Webpage:** If top accepted answer in that webpage turns out to be insecure.
- **Dataset:** TUM-CRYPTO, consisting of several secure and insecure JAVA code examples from Stack Overflow on the topics cipher, hash, hnv, hnvor, IV, KEY, tls, tm.

# Study 1

- **Goal:** Checking top ten Google Search results ($t_{10}$) and finding the ratio of secure and insecure Stack Overflow results among them.

- **Task:** Programming assignments based on AES encryption, consisting of initializing vector(IV), cryptographic key(KEY), symmetric cipher(CIPHER)[2].

# Study 1 Results

- The survey 1 result is visible in the below Figure 2.

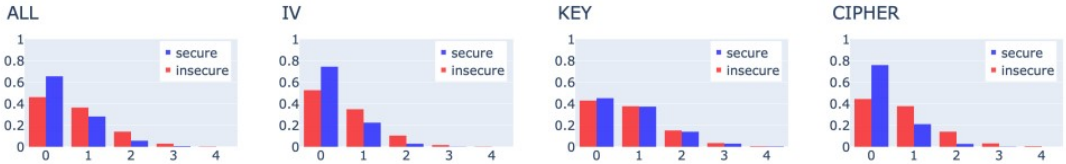


Figure: Binomial distribution of secure (blue) and insecure (red) Stack Overflow results over the top ten results $t_{10}$. X: count of secure/insecure results, Y: probability of secure/insecure.[2]

- In $t_3$, $t_5$, $t_{10}$(where $t_n$: top n Google Search results), the probability of insecure result was significantly higher than probability of secure result.
- If a search result ends up getting place in $t_{10}$, high chance that it will end up in $t_3$ rather than below ranks.

# Tackling the problem of insecure results

- Security based re-ranking technique to be used for Search results.
- New label introduced for KEY and IV "Secure Best Practice Examples"
- The Secure Best Practice Examples are different from normal Secured examples in the sense that the later may not be functionally complete even though it is secure. In that case the solution is of no use since it is not complete.
- In case of CIPHER, only secure and insecure categories were kept because JAVA has a standard and secure way of initializing CIPHER using JAVA SDK.

# Clustering

- Semi supervised clustering algorithm called DBSCAN was used for clustering.
- The datapoints(webpages here) were then sent for clustering.
- After successful clustering, a representative was chosen from each cluster and it's source code was reviewed manually to decide whether it was best practice or secure in case of KEY, IV where as secure or insecure in case of CIPHER.

# Re-Ranking

- A parameter search was performed to find a optimal boost value $b_s$ for secure results in range [-1.0, 1.0] to rerank the Stack Overflow webpages such that the secure best practice results get the highest priority followed by only secure results followed by insecure results with least priority[2].

- The Figure 3 shows how the best $b_s$ was selected.

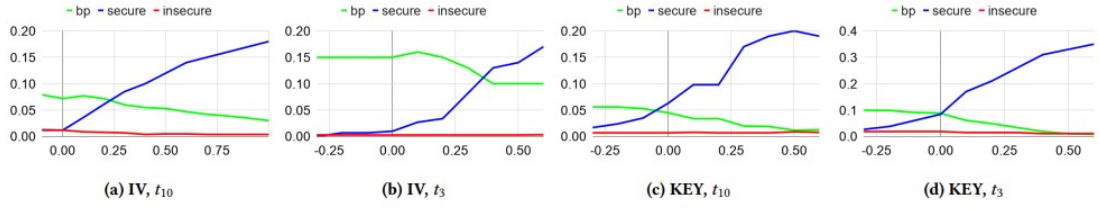


Figure: Parameter search for boosting results for IV and KEY. X: boost values, Y: probability of secure/secure best practice/insecure in top ten results.[2]

# Re-Ranking results visualization

- From Figure 4, it is now visible that how re-ranking helped to increase the probability that secure best practice results get highest priority followed by only secure results followed by insecure results across all tasks IV, KEY, CIPHER and in search result $t_{10}$, $t_5$, $t_5$ where $t_n$ = top n search results.


Figure: Binomial distribution of boosted secure best-practice (green), secure (blue), and insecure results (red) for IV, KEY, and CIPHER over $t_{10}$ X: count of secure best-practice, secure or insecure results in $t_{10}$ Y: probability of secure best practice/secure/insecure $t_{10}$.[2]

# Study 2 Task and Results

- **Assignment:** Search feasible solutions online for the same tasks(KEY, IV, CIPHER) and write JAVA codes to execute them.
- **Groups**
  - **Control Group:** Supposed to use Google's default search bar.
  - **Treatment Group:** Supposed to use security based re-rankable search bar.
- **Results**
  - The mean security jumped from 1.6(control group) to 1.9(treatment group) which indicates that the treatment group submitted more secure code snippets[2].
  - The mean functionality jumped from 2.35(control group) to 2.58(treatment group) which indicates that the treatment group submitted more functional solutions using the modified search engine[2].

# Study 2 Results continued...

- Also it was noticed that the treatment group(45.8%) faced 15.8% more secured results than the control group(30.8%)[2].
- In case of best practice results, treatment group got 36.9% whereas control group got 0.4%[2].
- In case of insecure results, treatment group got 17.3% which is 51.5% less than the control group(68.8%)[2].
- It was also observed that in the treatment group, the sum of votes per rank was significantly higher almost across all ranks($t_1$ to $t_{10}$ except $t_9$) than the control group. This is very useful since if it could have happened otherwise, the developers could have left the customized search engine since it would be continuously predicting less upvoted results at the top[2].

# What we are gaining using this re-ranking service??

- Re-Ranking is showing promising results in terms of security, functional correctness and completeness.
- Solution(Re-Rankable service) is user friendly since no need to install, setup or learn anything new.
- Significant from Company's point of view.
- Tech companies need to give security training to its employees.
- Even after that the programmers mainly rely on Google Search or Stack Overflow type platforms when they are stuck.
- This work will address the problem at the root and hence even if they search in Google they will end up with most relevant and secure results only most of the time and without any extra effort.

# Critiques

- **Assumption**
  - Number of people participated in surveys(192 in Study 1 and 218 in Study 2) is very small in number.
  - Number of countries participated is also very low(only 5 countries).
  - TUM-CRYPTO dataset is 3 years old. Programming forums data change every moment. Also, dataset contains very limited number of examples.
- **Technical Approach**
  - Task set up using only 3 keywords(KEY, IV, CIPHER). TUM-CRYPTO dataset contains 8 examples(cipher, hash, hnv, hnvor, IV, key, tls, tm). Could have designed a task which contains more examples.
  - TUM-CRYPTO dataset is 3 years old. Programming forums data change every moment.
  - Users were allowed to click beyond top ten search results, but study was done on top ten search results only.
  - DBSCAN used for semi supervised clustering. Could have used more faster algorithm like HDBSCAN.

# Critiques continued

- **Technical Approach**
  - Webpage labelled secure/insecure by looking at the topmost answer. This would declare a webpage insecure even if other answers except topmost one are secure.
- **Analysis/Results**
  - Fails when no secure solution is present in internet.
  - TUM-CRYPTO dataset is based on JAVA only. Could have found a dataset containing examples from different programming languages and more security keywords(CCA, CPA, PRF etc.).
  - Only Google Search Engine was modified. Not flexible for other search engine users. Also, data not recorded in study if anyone uses other search engine.
  - Even after manually checking each cluster, we were able to achieve a precision of 0.81. Although there were no false positives, we still think that the precision could have been made much better.
  - Re ranking might affect the probability to get the desired solution to problem. Person may be looking for a stub of code which he can write into his module and then add security as per his module, not be finding the direct secure solution.

# Improvements and Extension

- Running the same surveys with more number of people and people from more countries.
- We can check at least 50% of the answers of a webpage to declare it secure best practice/secure/insecure instead of only analyzing the topmost answer.
- We should explicitly instruct users to click on top ten search results only.
- Running the surveys on a better dataset which would support more examples related to cryptography and support more programming languages along with JAVA.
- Use more faster algorithm HDBSCAN instead of DBSCAN.
- Revamp interface of Stack Overflow to give users a visual feedback whether answer is secure or insecure. Can also add a feature to upvote and downvote the answers based on security.

# Improvements and Extension continued..

- Classify developers in terms of experience and compare their performance with and without the help of modified search engine. Check whether the junior developers using modified search engine can outperform the senior developers without using the modified search engine.
- Running same experiment in offline laboratory to prevent participants from using books, other search engines, GitHub Repositories.
- Can use cryptographic code analysis tools like FixDroid, LGTM, CodeQL along with modified Search Engine and default Google Search Engine and check their performance agianst each other.

# References


[1] Yasemin Acar et al. "You get where you're looking for: The impact of information sources on code security". In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2016, pp. 289–305.

[2] Felix Fischer, Yannick Stachelscheid, and Jens Grossklags. "The Effect of Google Search on Software Security: Unobtrusive Security Interventions via Content Re-ranking". In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021, pp. 3070–3084.

[3] Felix Fischer et al. "Stack Overflow Considered Harmful?" In: ().

[4] Michael Hucka and Matthew J Graham. "Software search is not a science, even among scientists: A survey of how scientists and engineers find software". In: *Journal of Systems and Software* 141 (2018), pp. 171–191.