The Rules of Engagement for Bug Bounty Programs

Aron Laszka[1], Mingyi Zhao[2], Akash Malbari[3], and Jens Grossklags[4]

[1] University of Houston
[2] Snap Inc.
[3] Pennsylvania State University
[4] Technical University of Munich

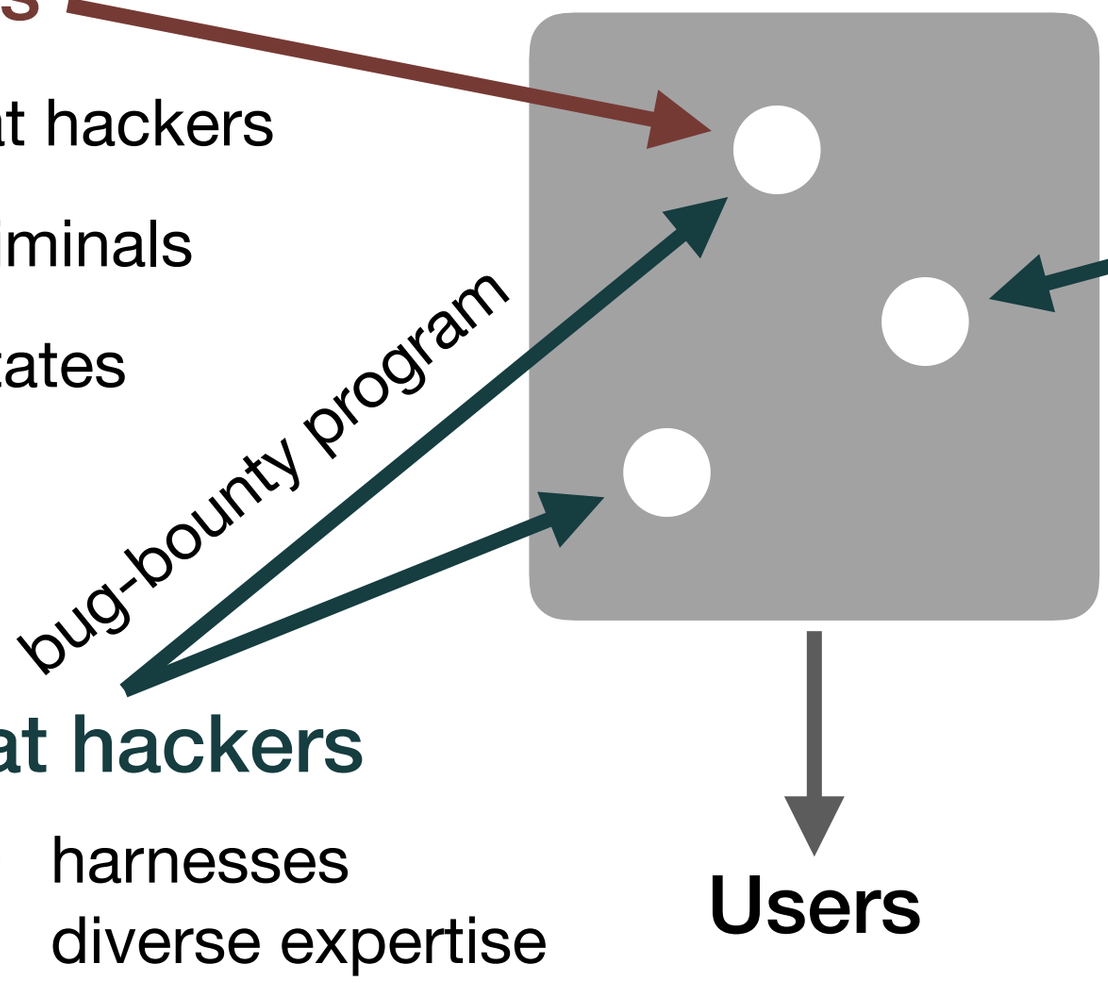
Bug-Bounty Programs

Website / software of an **organization**

facebook Google

Attackers

- black-hat hackers
- cyber criminals
- nation states



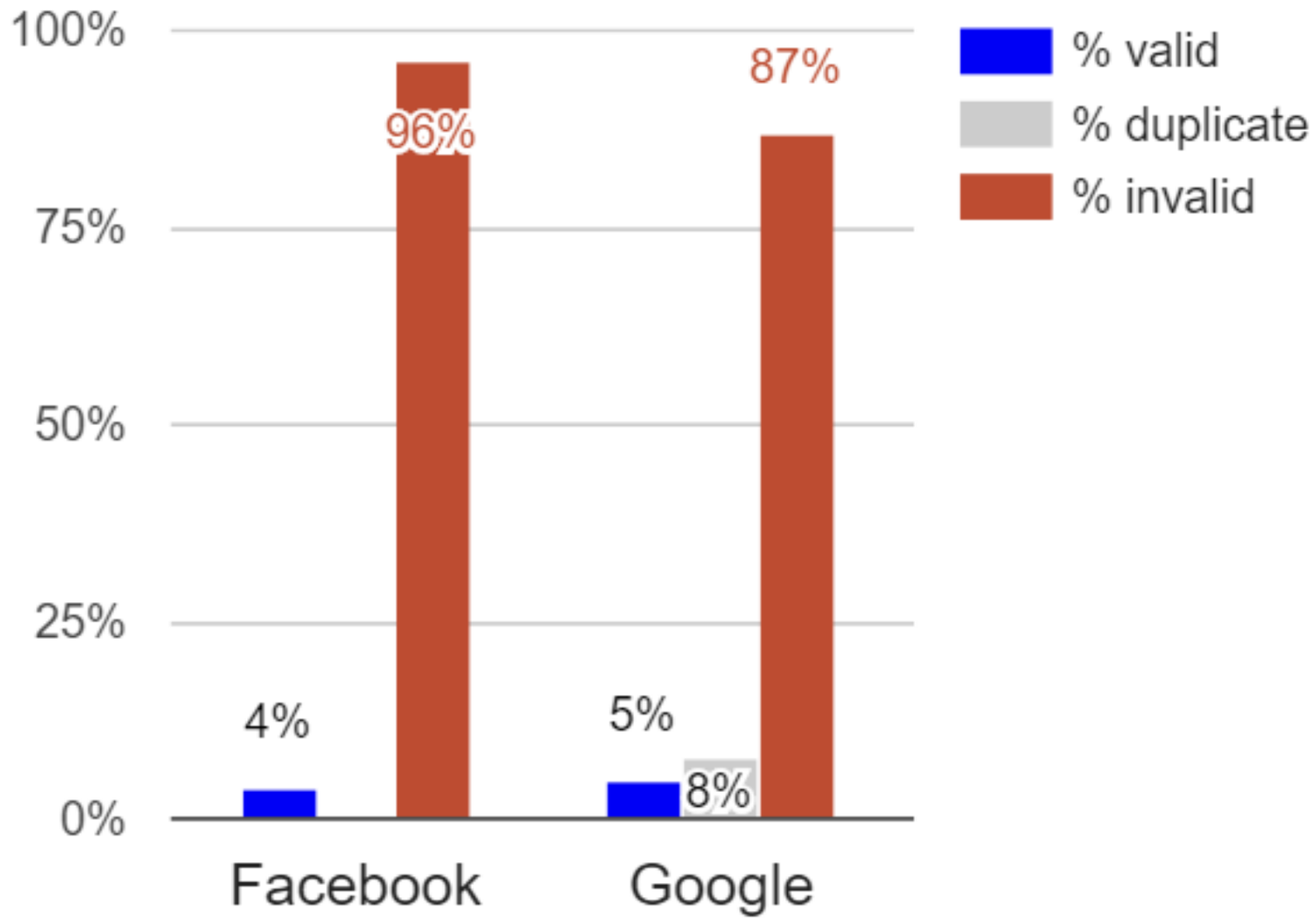
Defenders

- internal security team
- external partners (e.g., penetration testing)

White-hat hackers

- harnesses diverse expertise
- signals security

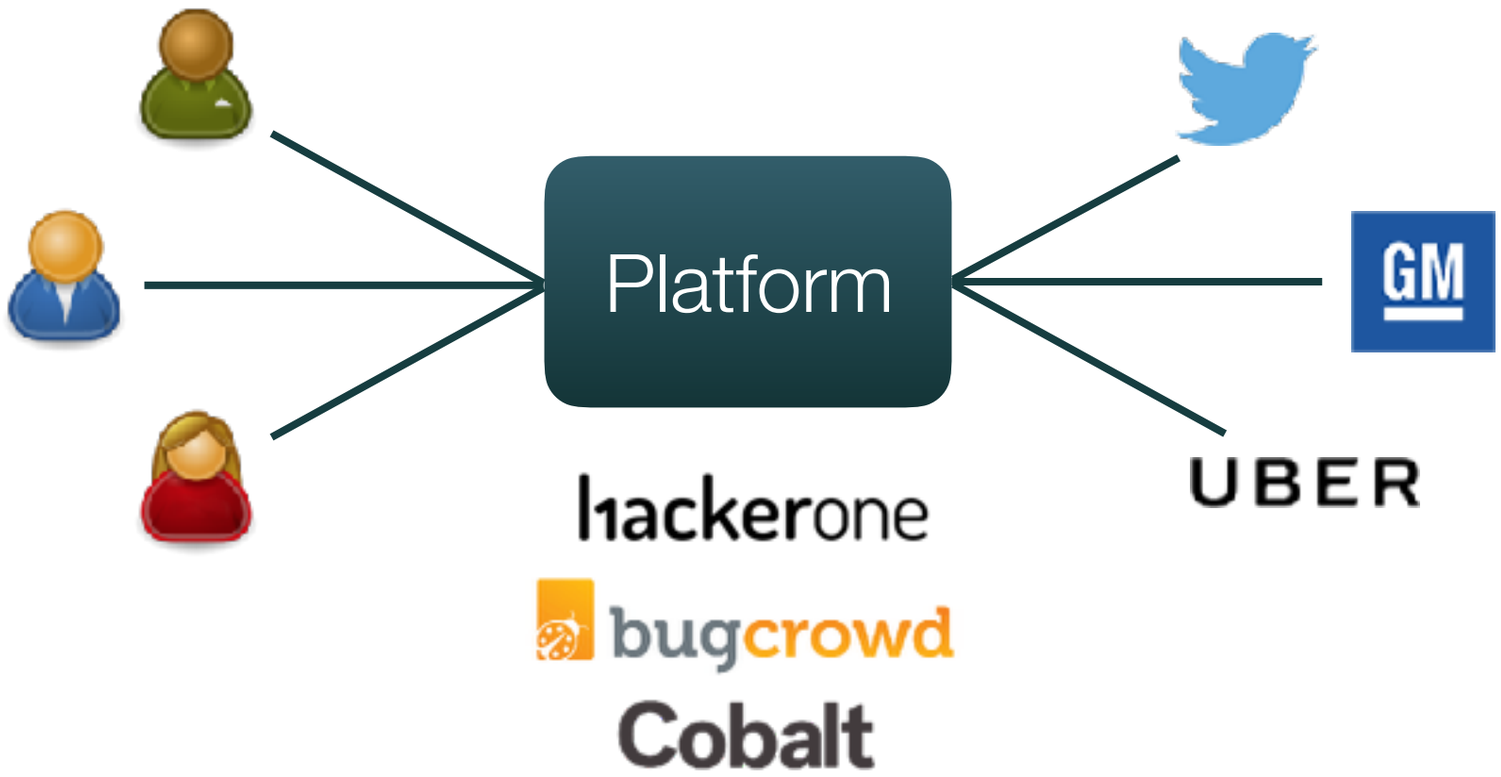
Problem with Bug-Bounty Programs



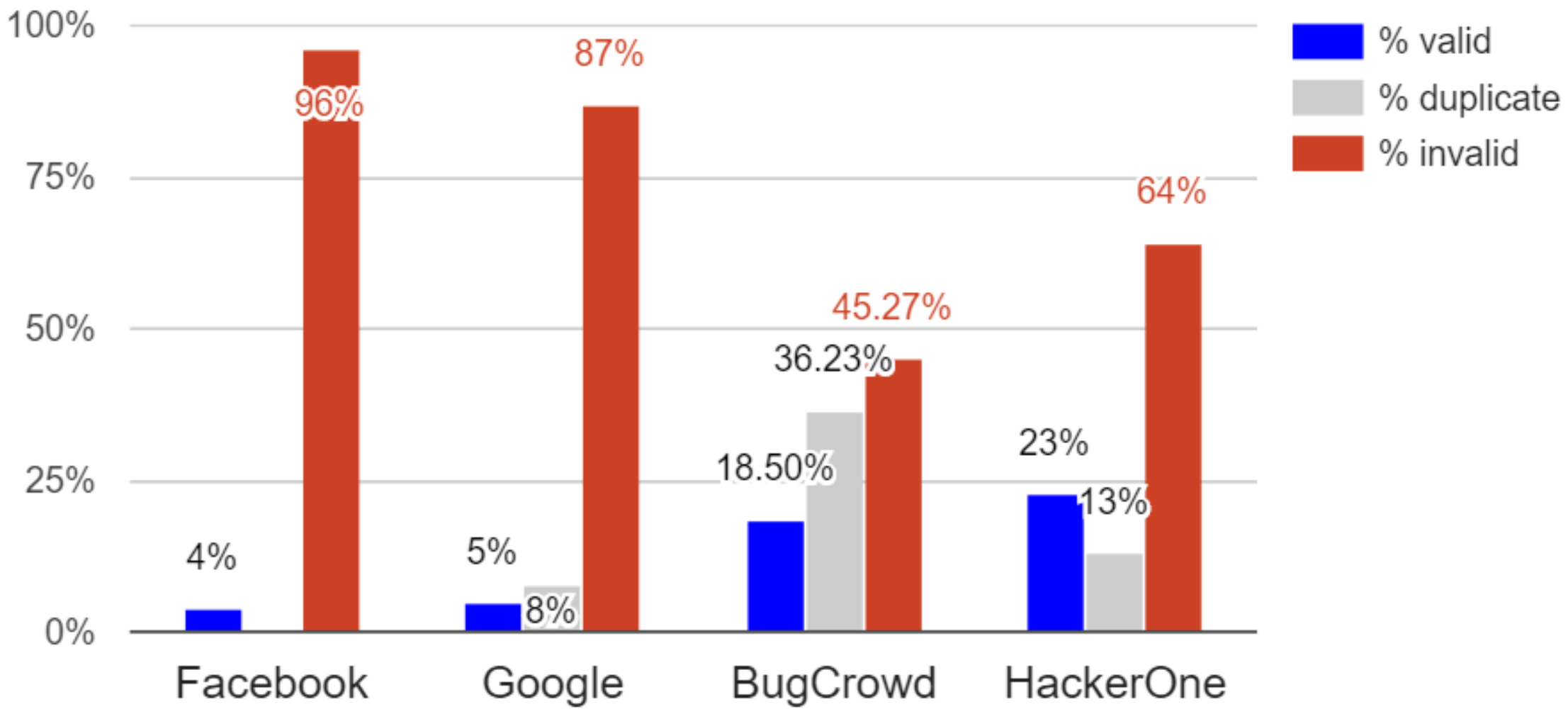
- Key challenge that “companies face in running a public program at scale is managing **noise**, or the proportion of **low-value reports** they receive” (HackerOne)

Bug-Bounty Platforms

- Connect white-hat hackers and organizations
- Facilitate setting up a program (infrastructure, payments, etc.), resolve trust issues between hackers and organizations
- Allows filtering hackers (and reports) based on their reputation



Problem with Bug-Bounty Programs



It is not that hard to keep white hats away…

*but how to **attract** the ones that do good work?*

Prior Analysis of Bug-Bounty Programs

- Prior work found “highly significant **positive correlation** between the **expected reward** offered and the **number of vulnerabilities received** by that organization per month” [1]
  - “Roughly speaking, a $100 increase in the expected vulnerability reward is associated with an additional 3 vulnerabilities reported per month”

VARIABLES	(1) # Vuln.	(2) # Vuln.	(3) # Vuln.
Expected Reward (R_i)	0.04***	0.03***	0.03***
	(0.01)	(0.01)	(0.01)
Alexa [log] (A_i)		-2.52*	-2.70**
		(1.20)	(1.21)
Platform Manpower (M_i)			10.54
			(10.14)
Constant	3.21*	16.12**	-133.05
	(1.88)	(6.39)	(143.66)
R-squared	0.35	0.39	0.40

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Is it all about the money?

[1] Zhao et al.: An Empirical Study of Web Vulnerability Discovery Ecosystems. Proc. of ACM CCS 2015.

The Rules of Engagement

- We analyze the **descriptions** of **bug-bounty programs** to find out what **rules** contribute the most to the **success** of a program
- **Qualitative analysis:** taxonomy of program rules
- **Quantitative analysis:** relation between rules and success

Dataset

- Source: HackerOne (`https://www.hackerone.com/`)
- Descriptions for 111 public programs downloaded January 2016
- Detailed history for 77 programs
  - rule description changes, bugs resolved, and hackers thanked
  - for each program, computed the rate of bugs resolved and hackers thanked (per year) for the time period in which the January 2016 version of the description was in effect

Problem: program rule description may be arbitrary text

Qualitative Study

- We **manually** evaluated **111 program descriptions**
- Taxonomy of rule statements
    1. in-scope
    2. out-of-scope
    3. eligible vulnerabilities
    4. ineligible vulnerabilities
    5. prohibited actions
    6. participation restrictions
    7. legal clauses
    8. submission guidelines
    9. public disclosure guidelines
    10. reward evaluation
    11. deepening engagement
    12. company statements

Taxonomy:
Scope and Eligibility

- **In-scope** and **out-of-scope**: define the scope of the program
  - e.g., allow / forbid working on core production site, APIs, mobile applications, and desktop applications
  - **staging sites**: some organizations allow / require white hats to work on staging sites that are provided by the organization
- **Eligible** and **non-eligible vulnerabilities**: specify the types of vulnerabilities that white hats should find
  - e.g., SQL injection, remote code execution, potential for financial damage, “issues that are very clearly security problems”

Taxonomy:
Restrictions and Legal Clauses

- **Prohibited actions**: list further instructions on what white hats should not do
  - e.g., automated scanners, interfering with other users, social engineering
- **Participation restrictions**: exclude certain individuals from participating in the program
  - e.g., employees, individuals of certain nationalities
- **Legal clauses**: promise not to bring legal action white hats if rules are followed, or remind them to comply with laws

Taxonomy:
Submission and Public Disclosure Guidelines

- **Submission guidelines**: specify the bug report contents
  - e.g., specific format, screenshots, pages visited
- **Public disclosure guidelines**: forbid / allow disclosing vulnerabilities to other entities (for some time period or until they have been fixed)
  - default period of secrecy on HackerOne: 180 days
- **Reward evaluation**: specifies an evaluation process that is used to determine whether a submission is eligible for rewards
  - e.g., reward amounts for specific types of vulnerabilities, areas of a site, and various other conditions
  - **duplicate report clause**: specifies if duplicate reports will be rewarded

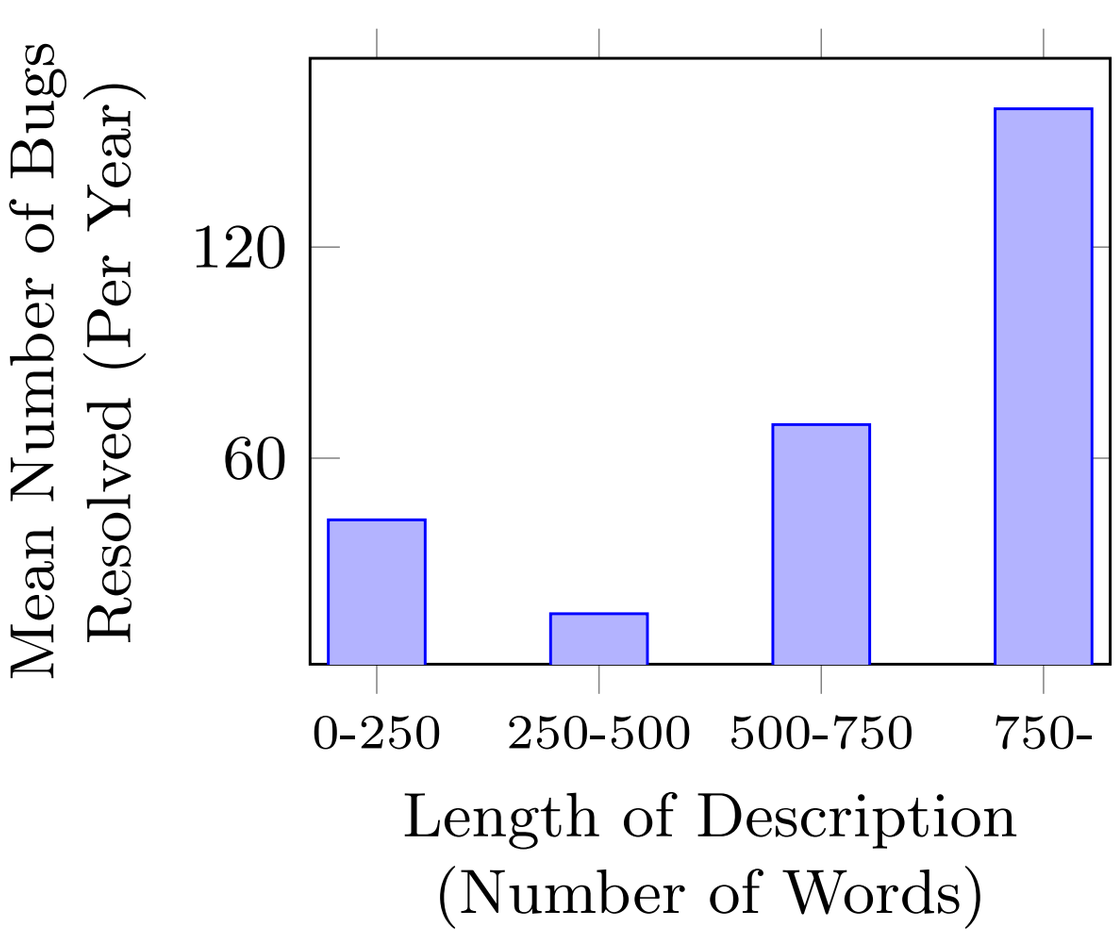
Taxonomy:
Deepening Engagement and Company Statements

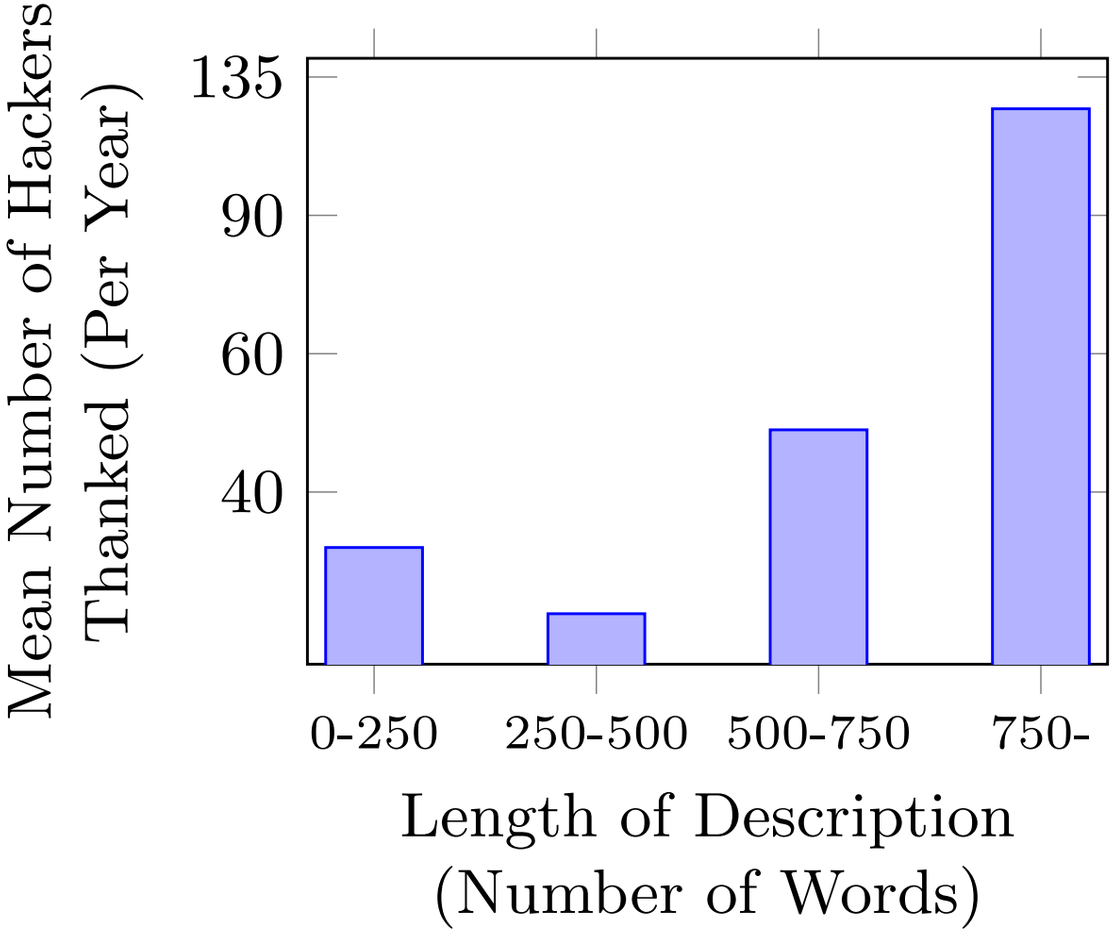
- **Deepening engagement**: statements provide instructions for white hats on how they can better engage in vulnerability research for the organization
  - e.g., “capture the flag” challenges
  - **test accounts**: some organizations allow / require white hats to create dedicated test accounts
  - **downloadable source code**: some organization provide source code
- **Company statements**:
  - demonstrate an organization’s willingness to improve security and to collaborate with the white hat community
  - not directly provide instructions or reward-relevant information

Quantitative Study

- Based on 77 programs with detailed history
- Measures of success:
  number of **bugs resolved** per year,
  number of **hackers thanked** per year
- Predictors
  - basic properties of program rule descriptions
  - statements and clauses identified by the taxonomy

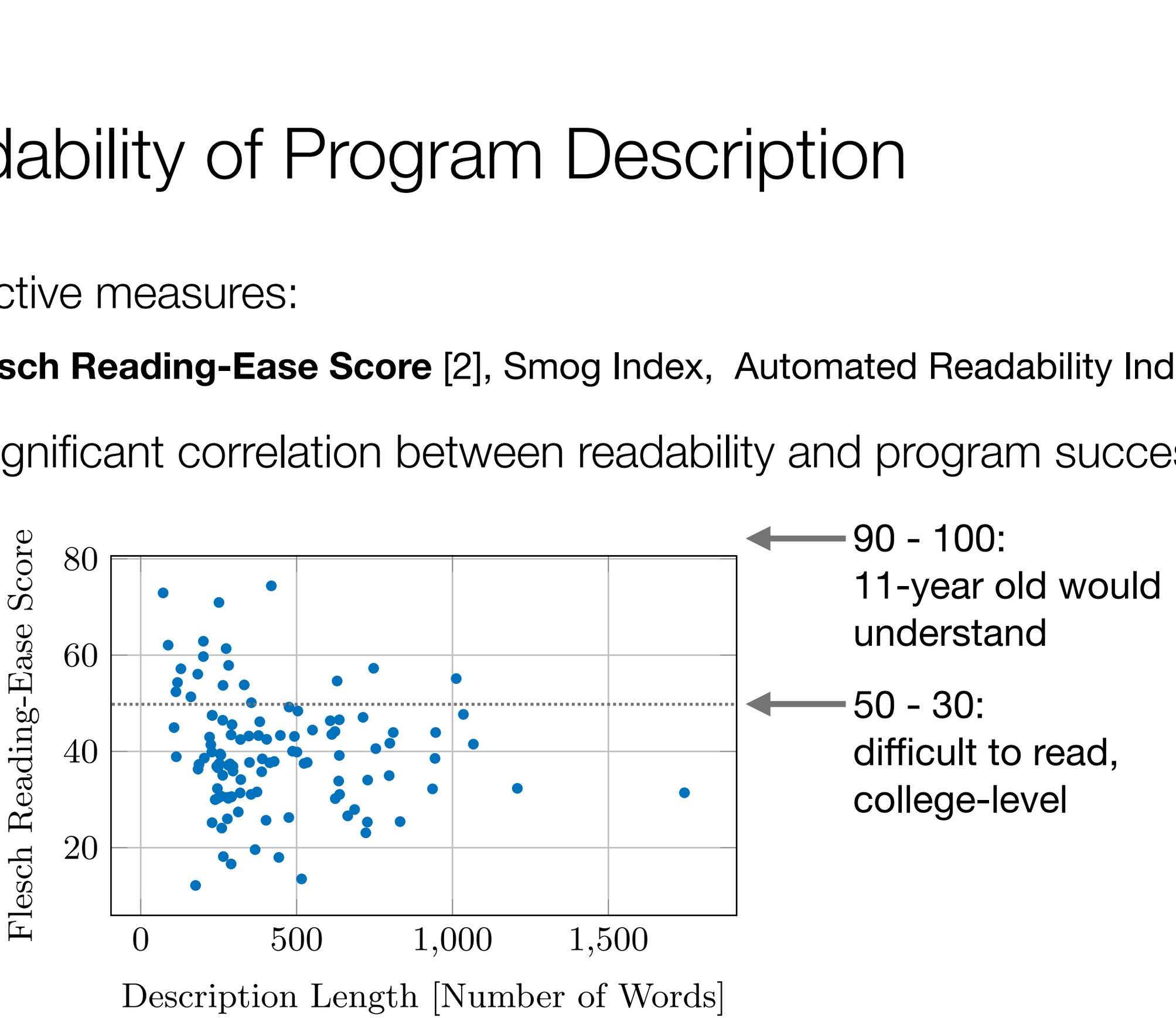
Length of Program Description





Readability of Program Description

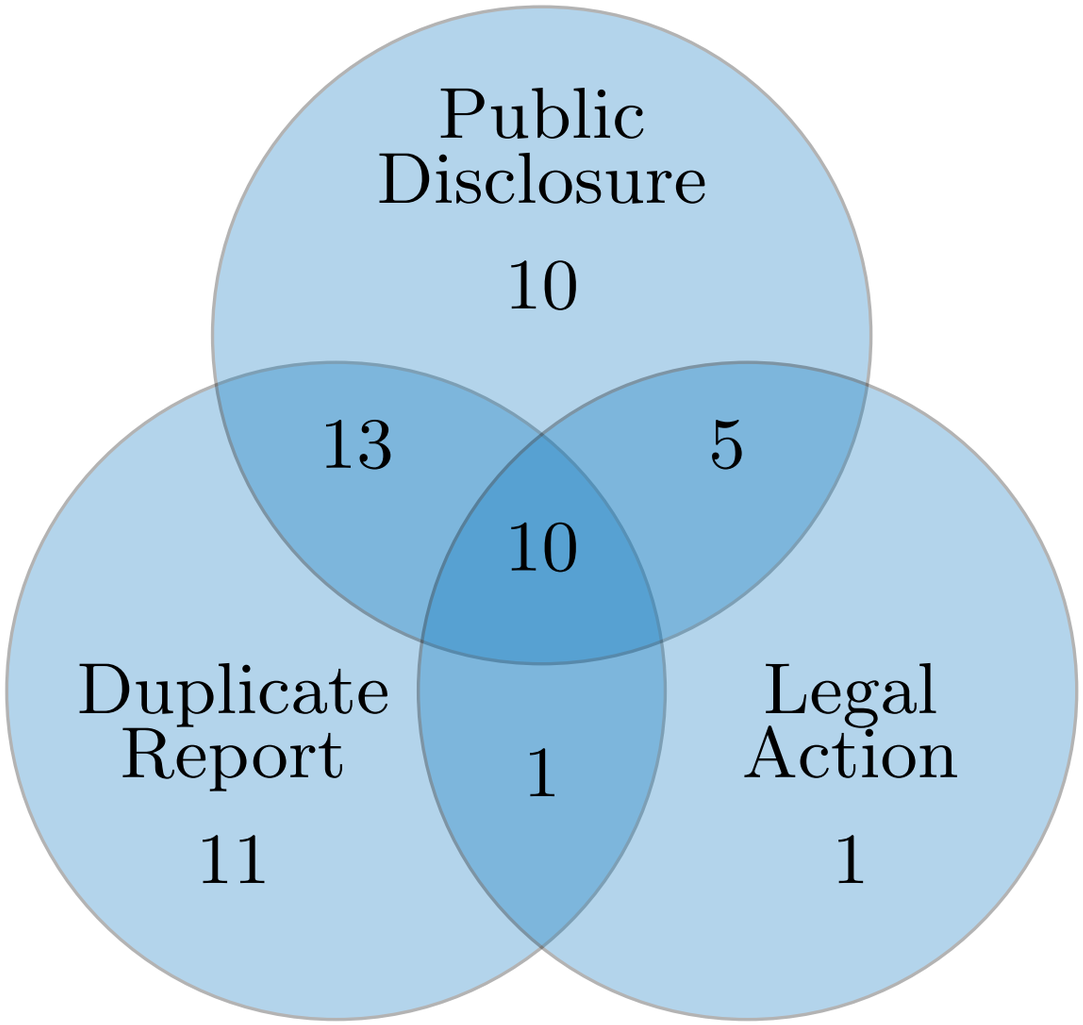
- Objective measures:
  - **Flesch Reading-Ease Score** [2], Smog Index, Automated Readability Index
- No significant correlation between readability and program success



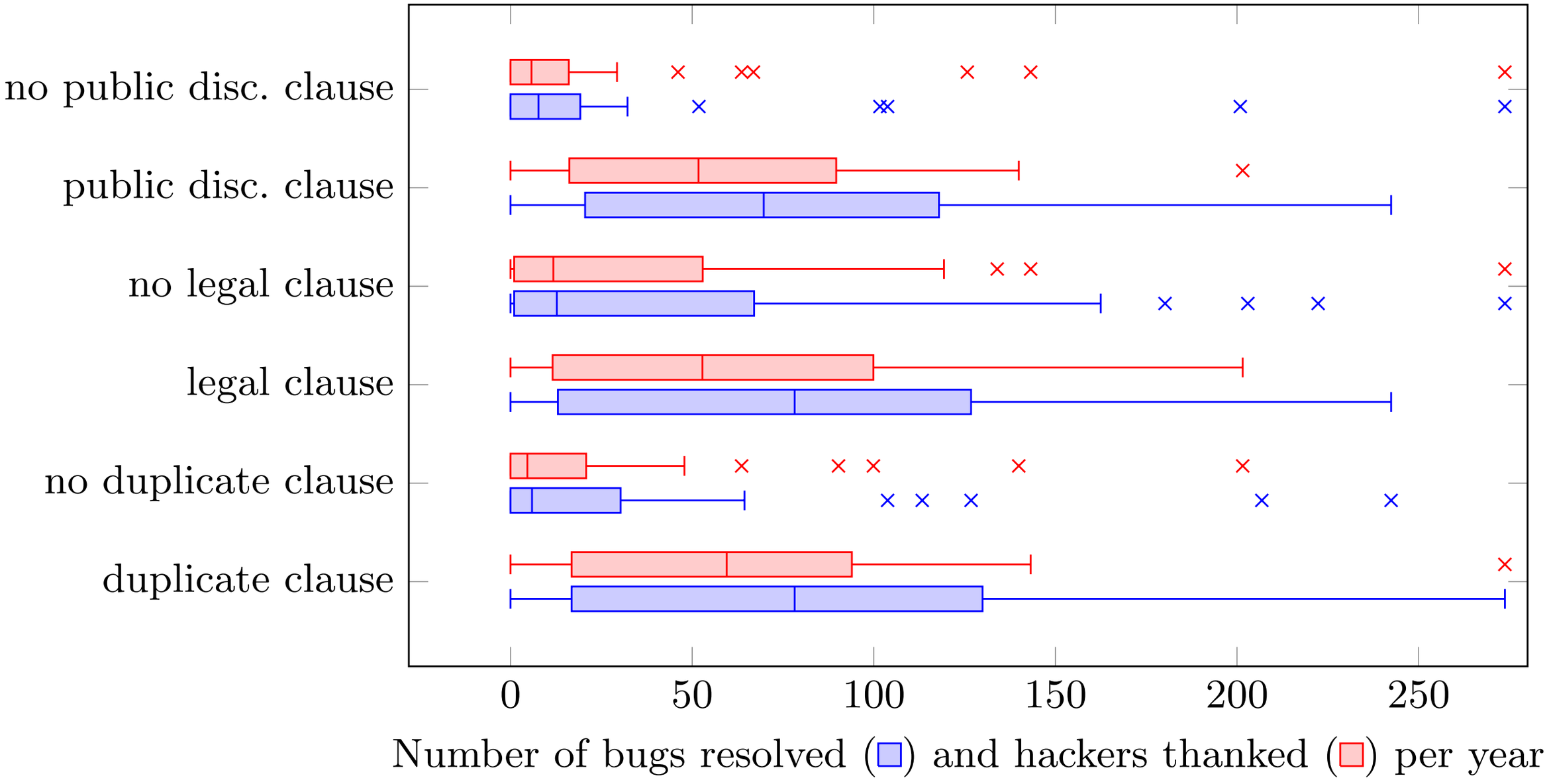
[2] Flesch, R.: A new readability yardstick. Journal of Applied Psychology 1948(32), 221–233.

Duplicate Reports, Legal Actions, and Public Disclosure

- **Duplicate report clause**:
  specifies if duplicate reports will be rewarded
- **Legal action clause**:
  informs white hats under what conditions it may (or may not) bring a lawsuit against them
- **Public disclosure clause**:
  forbids / allows white hats to disclose a vulnerability to other entities (for some time period or until it has been fixed)



Duplicate Reports, Legal Actions, and Public Disclosure



Staging Sites, Test Accounts, and Downloadable Source Code

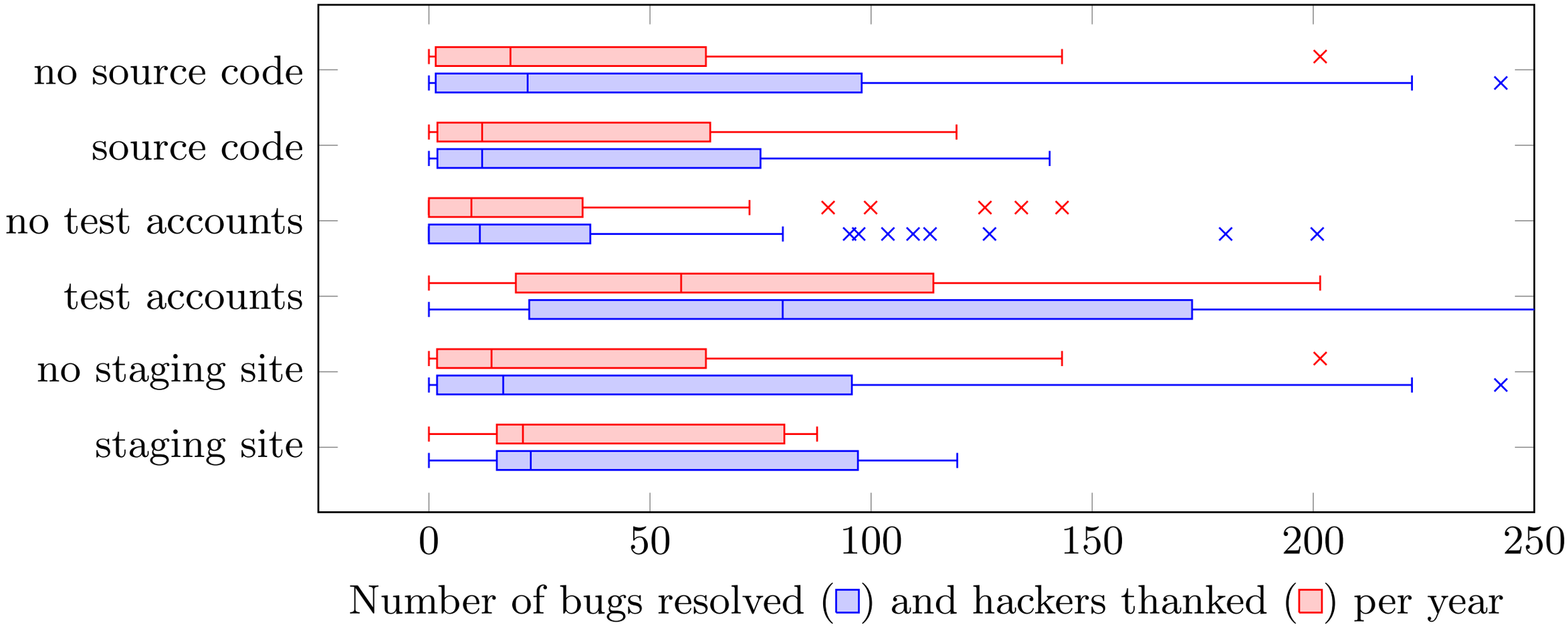
How much help do organizations provide to white hats?

- **Staging sites**:
  allow / require white hats to work on staging sites that are provided by the organization

- **Test accounts**:
  allow / require white hats to create dedicated test accounts

- **Downloadable source code**:
  provide downloadable source code for the software / service

Staging Sites, Test Accounts, and Downloadable Source Code



Regression Analysis

- Dependent variable:

  number of bugs resolved V

- Predictors:
  - average bounty B
  - Alexa rank A
  - previous features

VARIABLES	(1) V	(2) V	(3) V
Length of the rule (L)	0.18***	0.09*	0.01
Average bounty (B)		0.12*	0.09*
Age of the program (T)		0.05	0.13***
Log(Alexa rank) (A)		-4.65	-4.20
Has legal clause (LE)			23.04
Has duplicate report clause (DU)			47.39*
Has public disclosure clause (DI)			60.41**
Has staging site (ST)			1.10
Asks to use test accounts (TA)			1.01
Asks to download source (DS)			45.56*
Constant	-15.21	23.21	-14.40
R-squared	0.27	0.43	0.57

*** p<0.01, ** p<0.05, * p<0.1

Conclusion

- Limitation of our study
  - only public programs (no publicly available data for private ones)
  - only the white hats' success is measurable, not their effort
- Lessons learned
  - there are factors (beside expected amount bounty) that are crucial for the success of a program
  - platforms should help bug-bounty programs to define these rules
- Future work
  - extending the scope of our analysis to a larger number of programs, employing natural language processing and text mining

Thank you for your attention!

Questions?

Aron Laszka: alaszka@uh.edu / www.aronlaszka.com
Mingyi Zhao: rvlfly@gmail.com
Jens Grossklags: jens.grossklags@in.tum.de