# Rumours of our Demise Have Been Greatly Exaggerated

Michael Gianarakis
Julian Berton

# Obligatory Introduction Slide

## Michael Gianarakis

**@mgianarakis**

Director of SpiderLabs, Asia-Pacific & Japan

SecTalks Brisbane

Have also spoken at the *"equally as good"* WAHCKon hacking conference (❤️ Nanomebia)

Flat Duck Justice Warrior 🦆

## Julian Berton

**@julianberton**

Application Security Engineer at SEEK

OWASP Melbourne Chapter Lead

Web developer in a previous life

Climber of rocks

# Why This Presentation?

- There is a fair bit of hype surrounding crowdsourced security testing and the result-oriented economic model

- Many have claimed that "traditional" pentesting is dead and the industry will be "Uberised" as a inevitable future

- Most of the discussion on this topic is either from the bug hunters (great) or from the bounty companies themselves (mixed bag) - very few address the point of view of an organisation trying to manage their security

- Intends to address the realities of running a bounty and where they fit in an organisation's security testing framework

# Bug Bounties

# Bug Bounty Basics

- Concept is simple
- Providing a mechanism for security researchers to submit a bug in a system or application usually with some incentive (cash or kudos) tied to doing so
- Pioneered and established by the likes of Mozilla, Microsoft and Google

# Bug Bounty Basics

- More recently various startups have entered the space offering to host or manage bug bounties for organisations and offer them to their platform or security testers

- Companies such as Bugcrowd, HackerOne, Synack

- Refer to them as HaaS (Hacking as a Service) providers in the talk (as opposed to "traditional" pen test providers)

# Different Types of Bounties

- Public bounties - bounty programs that invite participation from the public

- Private bounties - invite only programs

- Timed bounties - usually limited to the HaaS companies, a timed bounty is a bounty (typically private) that is only open for a short period of time

# Bug bounties are essentially pen testing with a different economic and resource model

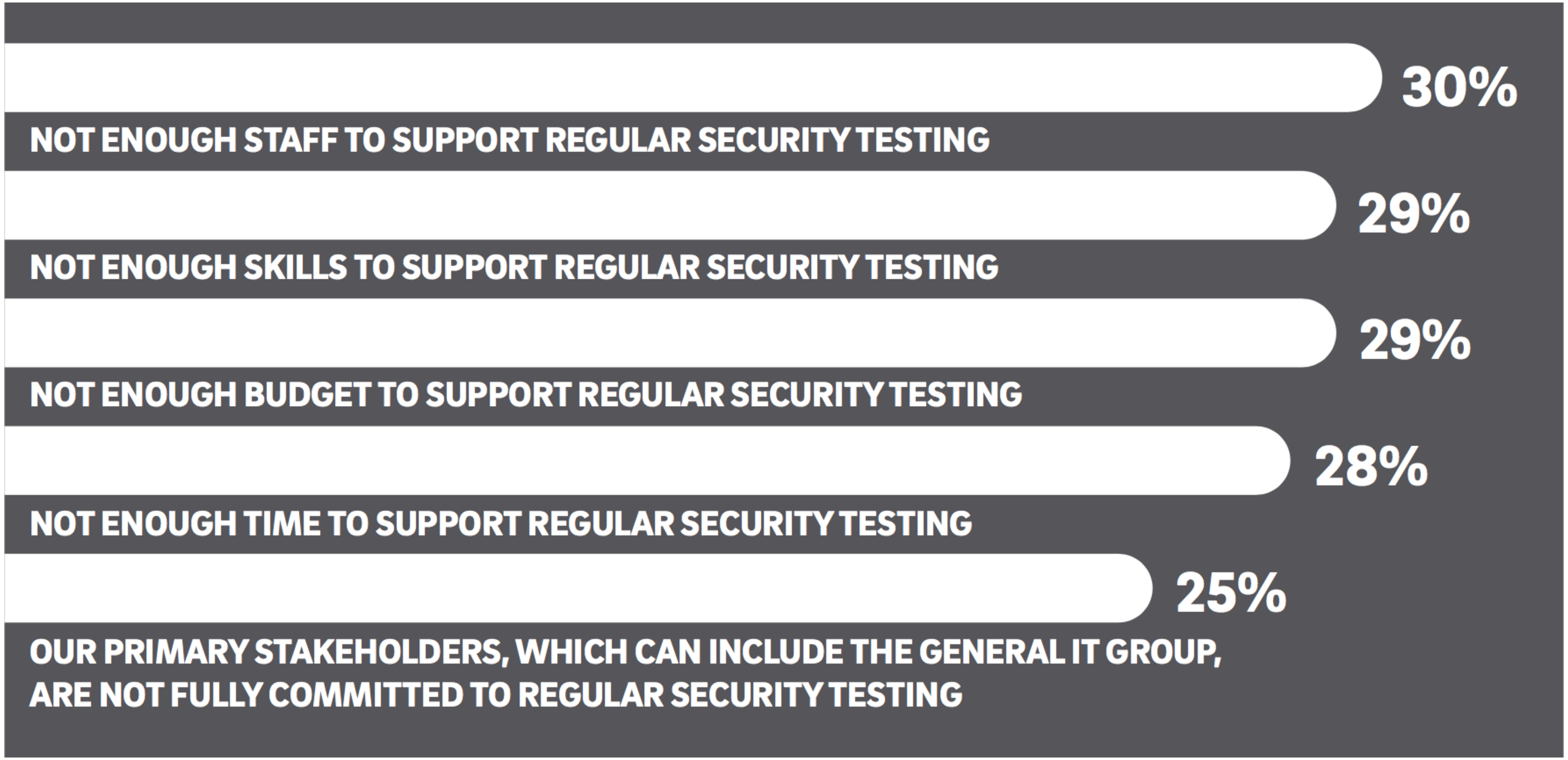
That’s what makes them interesting

# The Hype

# Why you should pay attention

- There is a lot hype surrounding bug bounties - primarily driven by the VC funded Silicon Valley marketing departments
- Bug bounties and HaaS providers represent some interesting innovation in the security testing space
- Can be a great compliment to your appsec program
- If you perform security testing you should explore the benefits and tradeoffs

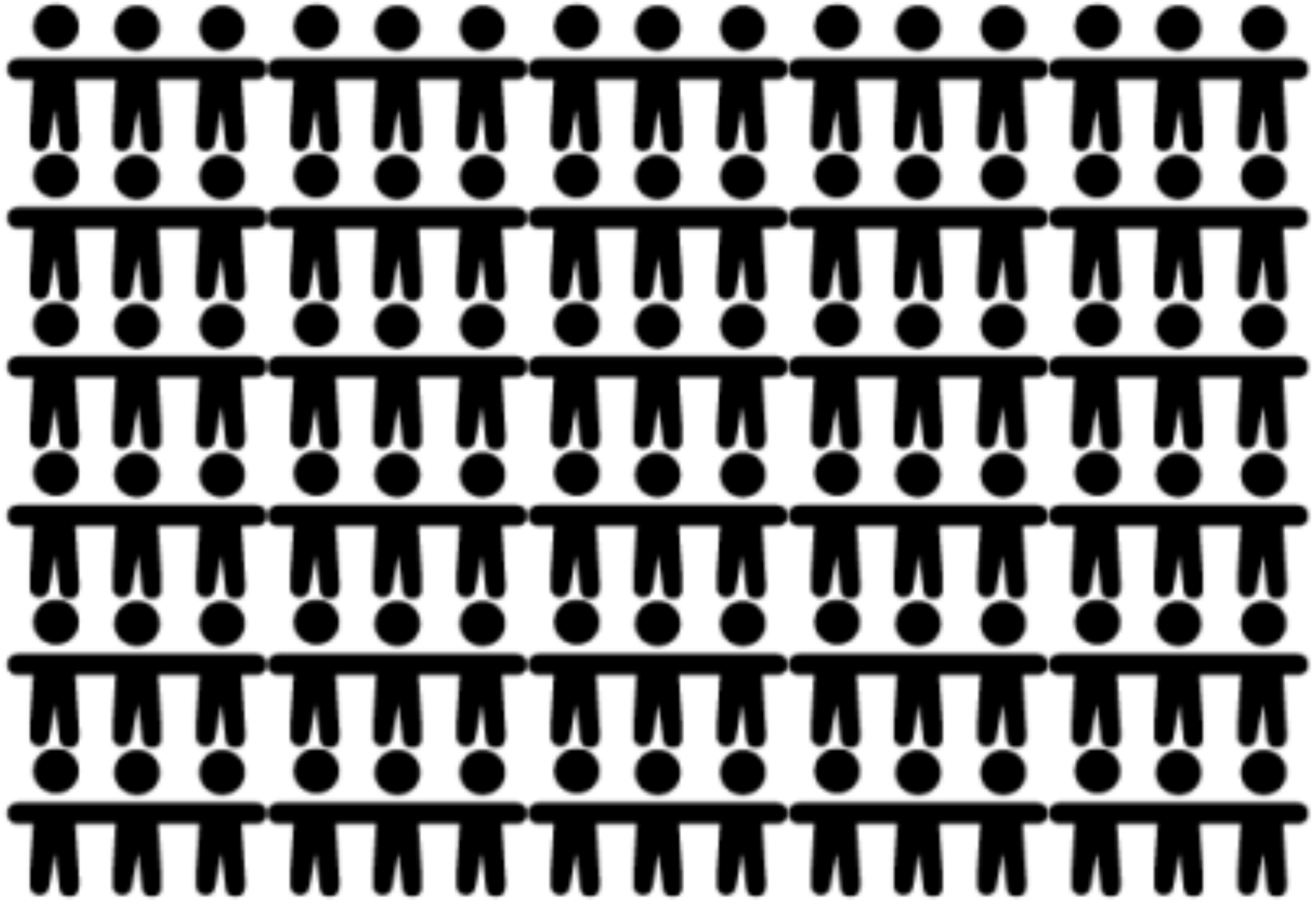
# Security Testing Challenges



https://www.trustwave.com/Resources/Library/Documents/Australian-Security-Testing-Practices-and-Priorities/

# Skills Shortage

Tech Team



Security Team

# Evolving Development Practices

## Then

3-6 month deploy to prod cycles (think waterfall)

One software stack per company (e.g. C#, .NET, SQL Server and IIS

Ratio of security people to developers/infrastructure is skewed

## Now

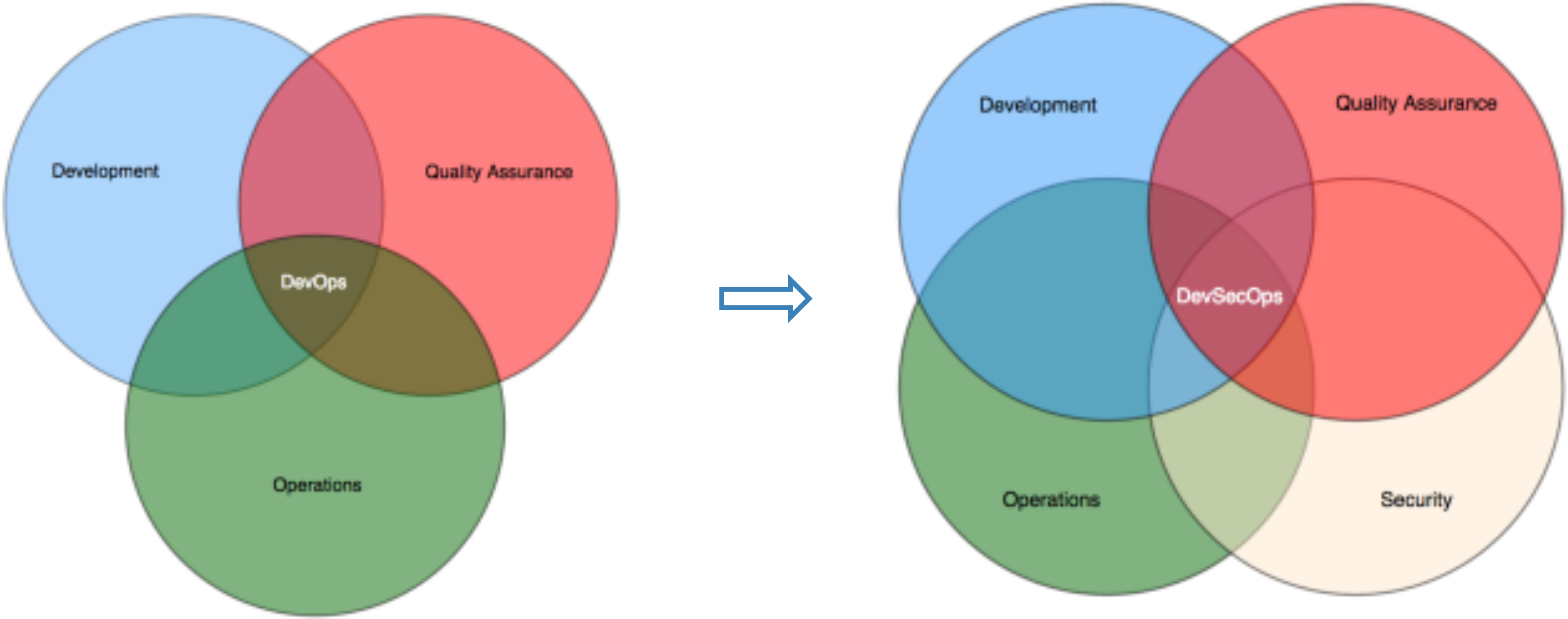
CD/CI, deploy to prod daily (move faster)

Agile development practices

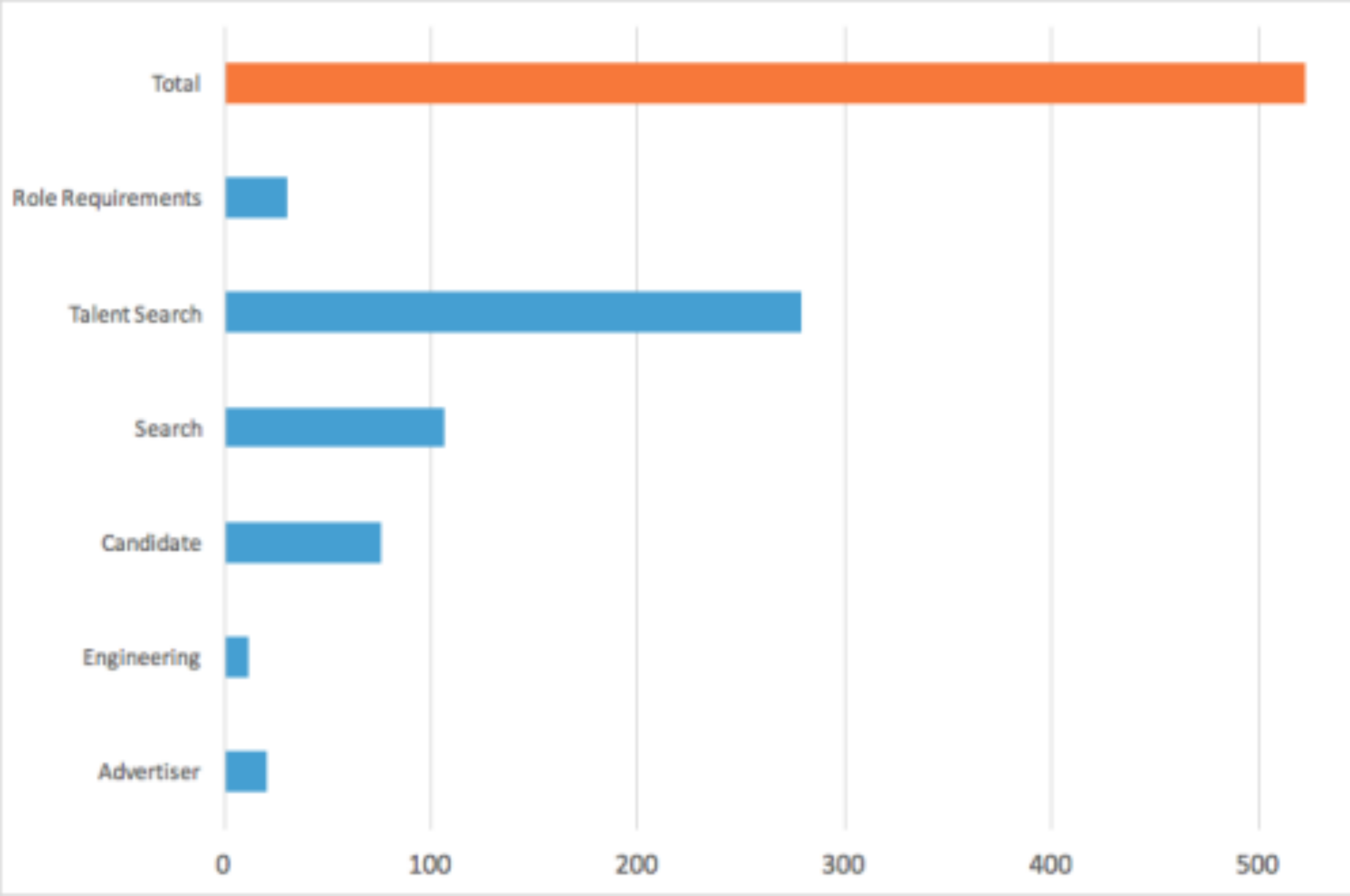
Developers do everything = devops practices

Ratio of security people to developers/infrastructure still skewed

# Evolving Development Practices



# Evolving Development Practices



**~30 times a day**

# Growing Complexity

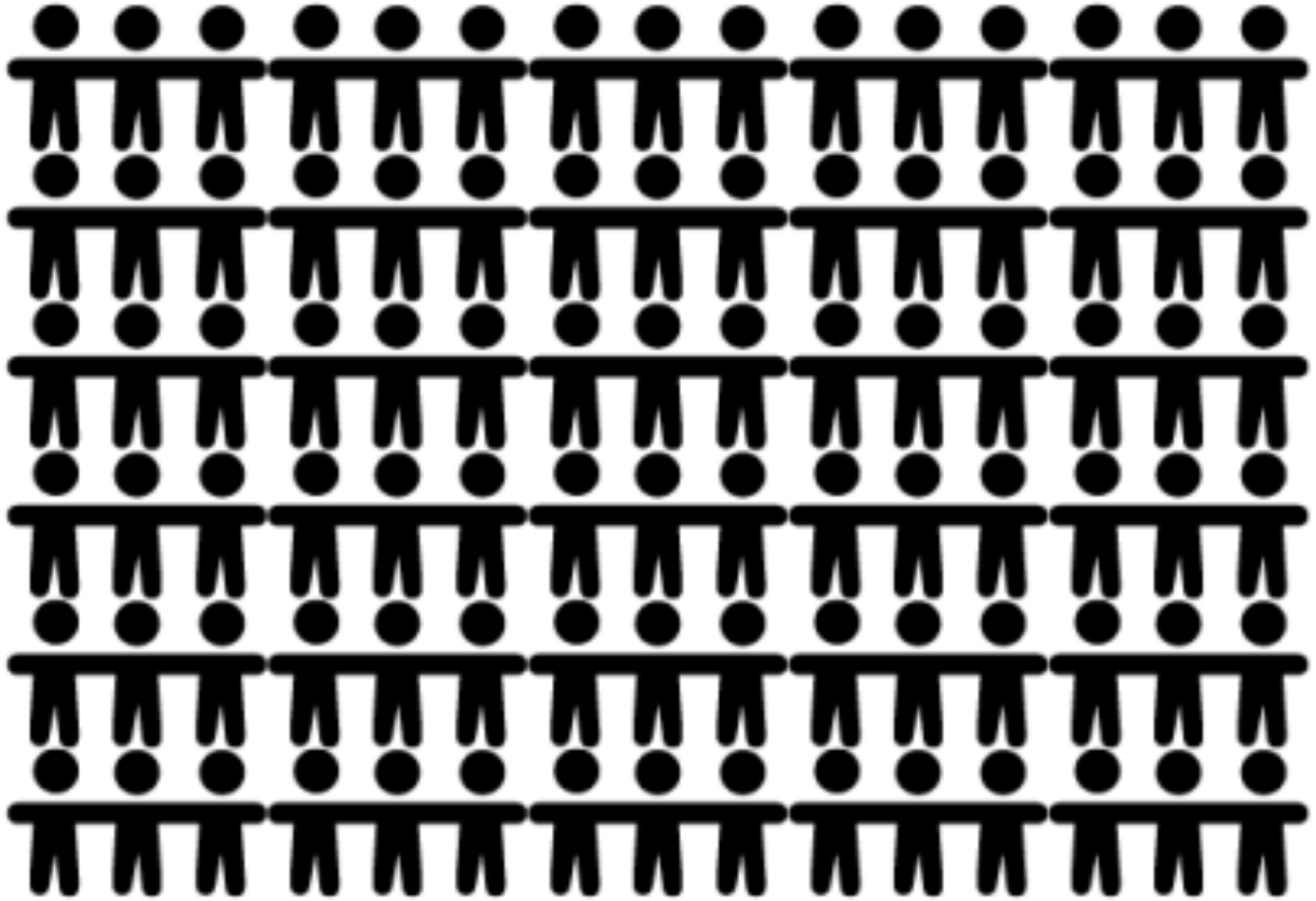
~150 different tools, languages, platforms and frameworks

# The Crowd-Sourced Future

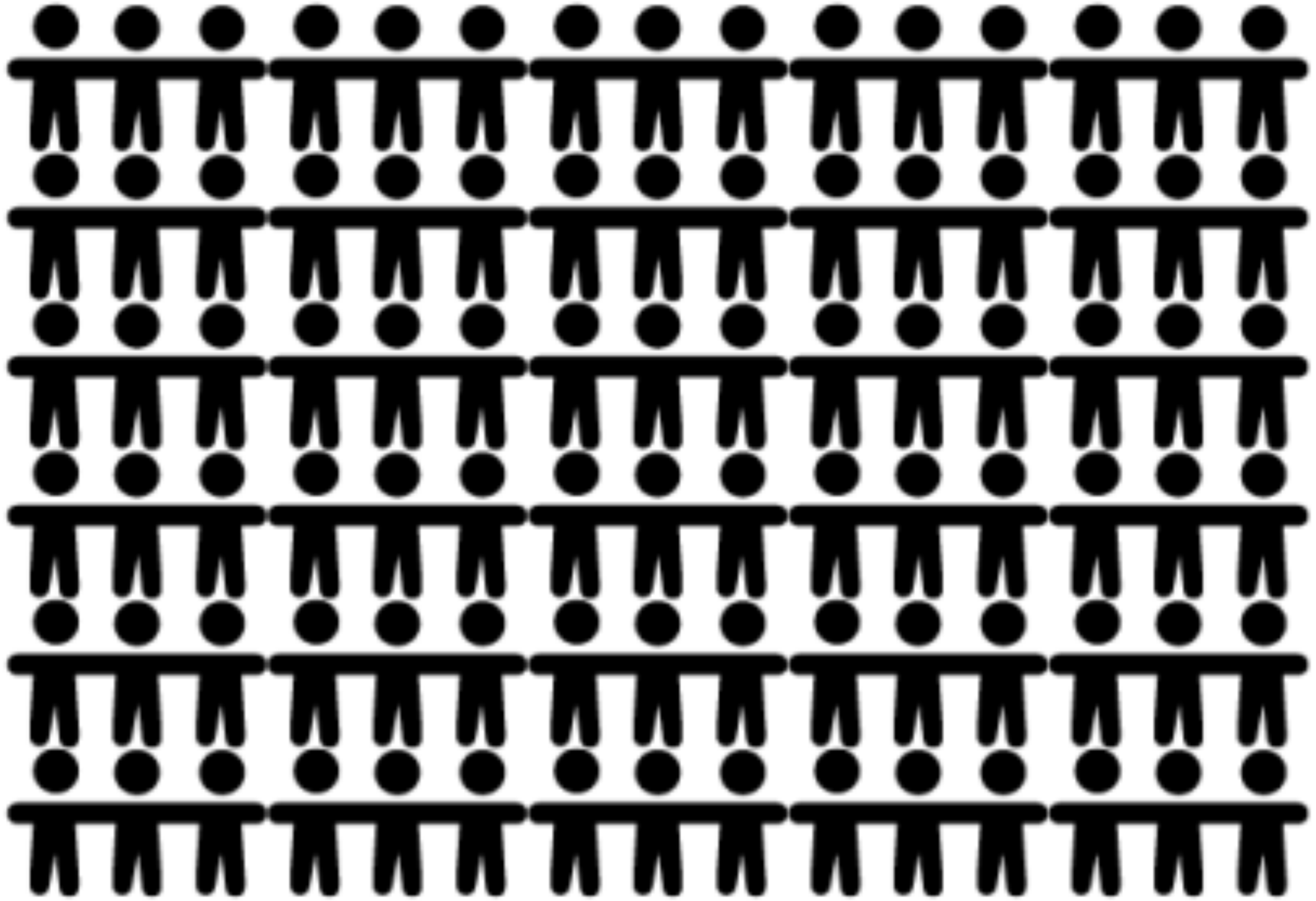
- Bug bounties address the skills shortage via crowd-sourcing
- Unlocks access to a vast resource pool - Bugcrowd and HackerOne claim testers in the tens of thousands but in theory the resource pool is potentially much greater than that
- Even private/invite-only bounties can give access to a larger and more diverse resource pool than what you might find with traditional in-house or contract testing teams

# The Crowd-Sourced Future

Tech Team



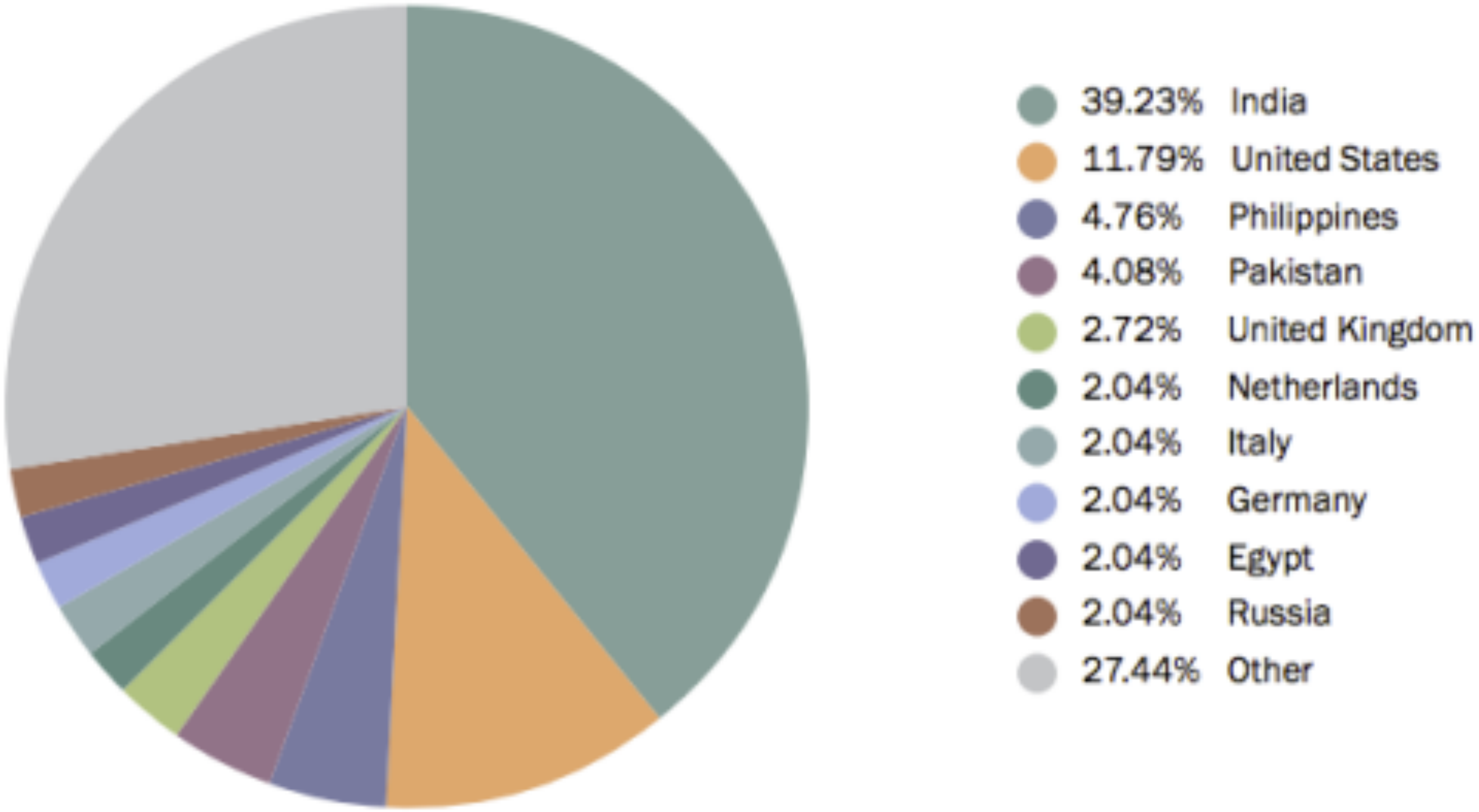
Security Team



# The Crowd-Sourced Future

- The benefits of the crowd-sourced model are obvious
  - Scales well - tap into 100s of testers instantly
  - Diverse skills sets - researchers specialised in certain classes of bugs
  - Can lead to high quality bugs

# The Crowd-Sourced Future



https://pages.bugcrowd.com/2016-state-of-bug-bounty-report

# The Result-Based Economic Model

- Organisations running bug bounty programs pay out based on the successful bug submissions - which represent genuine, validated, non-duplicated vulnerabilities

- This flips the switch on how most companies pay for for vulnerabilities

- Instead of paying for a resources time (be it in-house or a consultant) to find the vulnerabilities you are paying for the bug itself.

- The real innovation of the bug bounty model

# The Result-Based Economic Model

- The central benefit to this model is that there are less compromises that you have make compared to traditional testing activities
  - You don’t have to limit yourself to a small number of testers
  - You don’t have to limit yourself to a set timeframe
  - You don’t have to limit scope to the same extent

# The Reality

# Can You Run a Bounty?

- Do you have security aware people to manage the program?
- What is the security maturity of the systems you want to test?
- Do you have the budget and traction to fix security in a timely manner?

# Can You Run a Bounty?

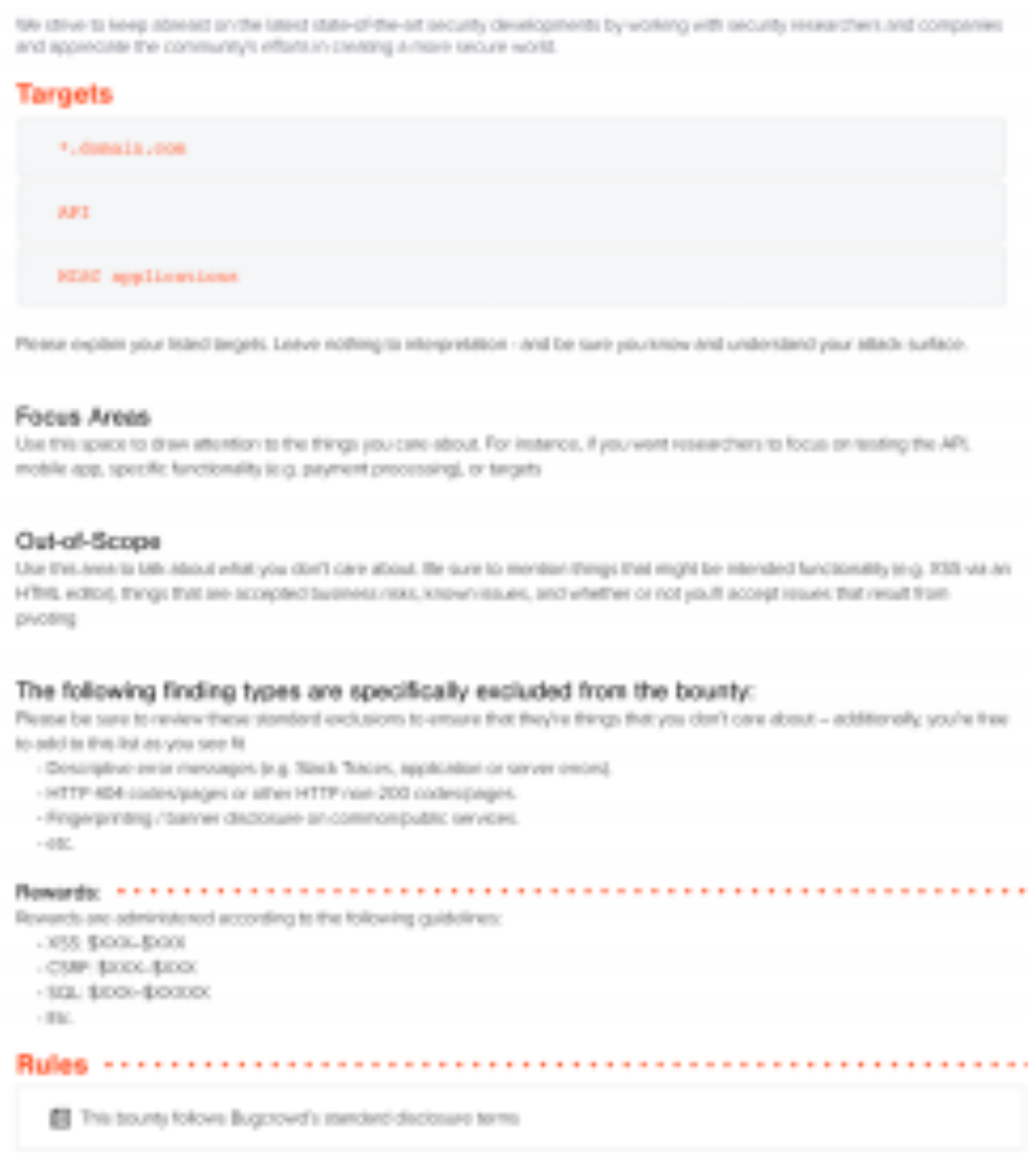
- How fragile are your systems?
- Can testing be performed on production? No? Do you have a publicly available test environment?
- Can the production app detect and block attacks if they are affecting customers or degrading service?

# SEEK’s Private Timed Bounty

- 50 researchers invited and were paid for bugs found.
- Testing occurred on production systems.
- 3 apps in scope.

# The Brief

- Overview of company and targets.
- Targets - sites that are in scope.
- Focus Areas - Draw attention to things you care about.
- Out-of-Scope - Areas that are off limits.
- Issue Exclusions - Issues you will not reward.
- Rewards - What you will reward for issues found.



We strive to keep abreast on the latest state-of-the-art security developments by working with security researchers and companies and appreciate the community's efforts in creating a more secure world.

**Targets**

```
*.domain.com
```

```
API
```

```
MISC applications
```

Please explain your listed targets. Leave nothing to interpretation - and be sure you know and understand your attack surface.

**Focus Areas**

Use this space to draw attention to the things you care about. For instance, if you want researchers to focus on testing the API, mobile app, specific functionality (e.g. payment processing), or targets

**Out-of-Scope**

Use this area to talk about what you don't care about. Be sure to mention things that might be intended functionality (e.g. XSS via an HTML editor), things that are accepted business risks, known issues, and whether or not you'll accept issues that result from pivoting

**The following finding types are specifically excluded from the bounty:**

Please be sure to review these standard exclusions to ensure that they're things that you don't care about – additionally, you're free to add to this list as you see fit

- Descriptive error messages (e.g. Stack Traces, application or server errors).
- HTTP 404 codes/pages or other HTTP non-200 codes/pages.
- Fingerprinting / banner disclosure on common/public services.
- etc.

**Rewards:**

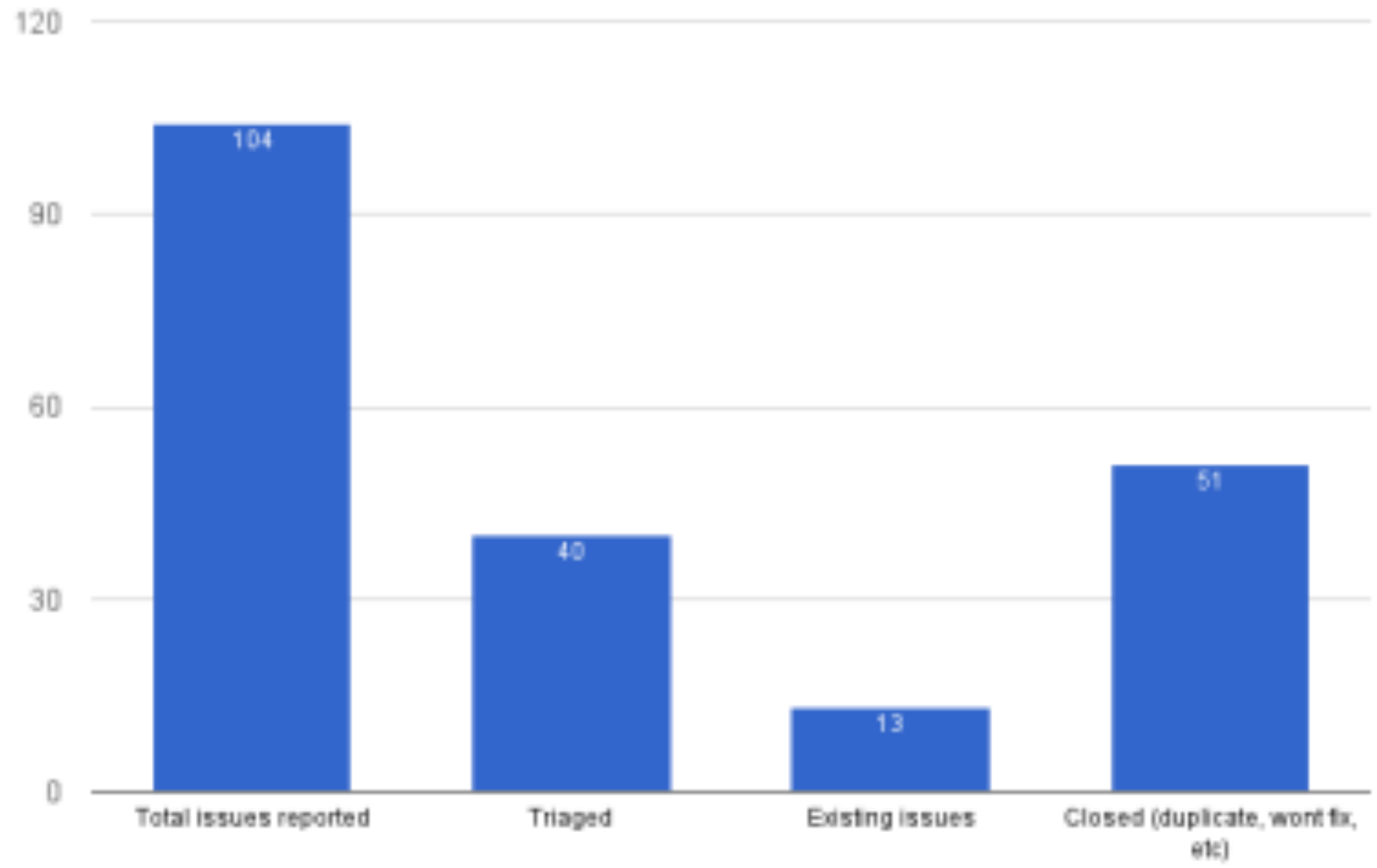
Rewards are administered according to the following guidelines:

- XSS: $XXX–$XXX
- CSRF: $XXX–$XXX
- SQL: $XXX–$XXXXX
- Etc.

**Rules**

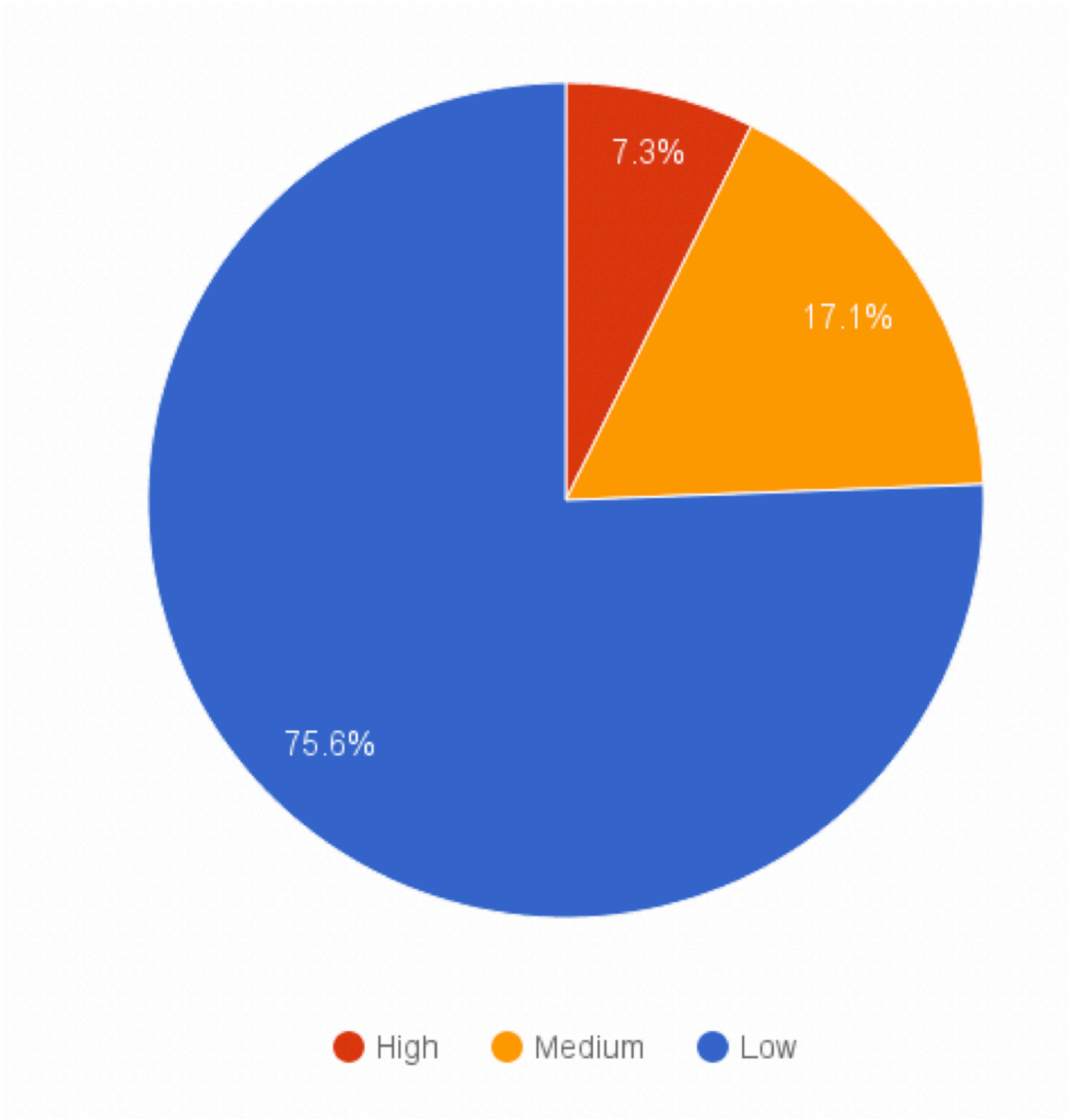
This bounty follows Bugcrowd's standard disclosure terms

# Submissions



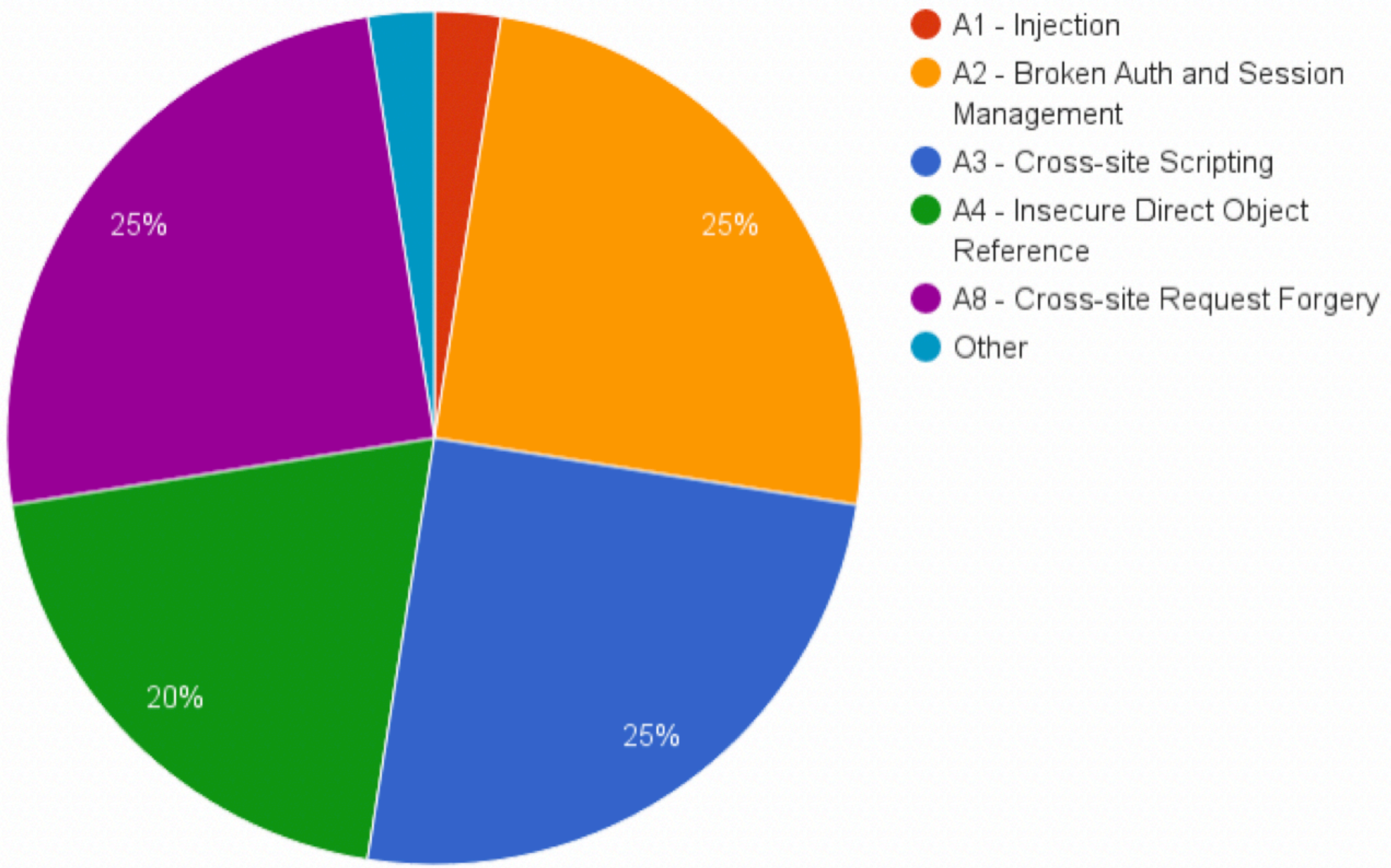
104 issues were reported in total, with 40 being verified issues

# Severity



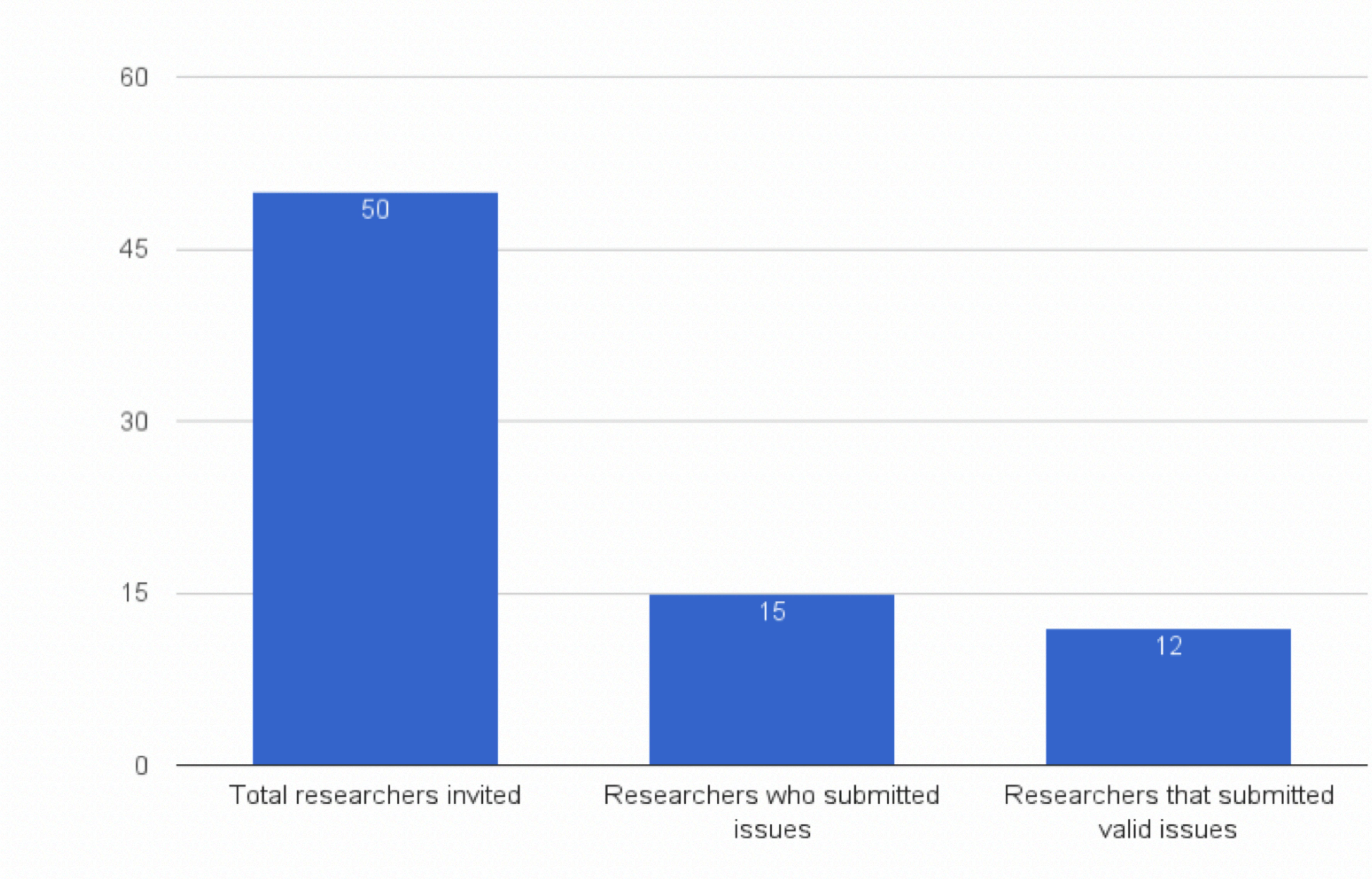
3 High, 7 Medium and 31 Low issues were reported

# Issues by Category



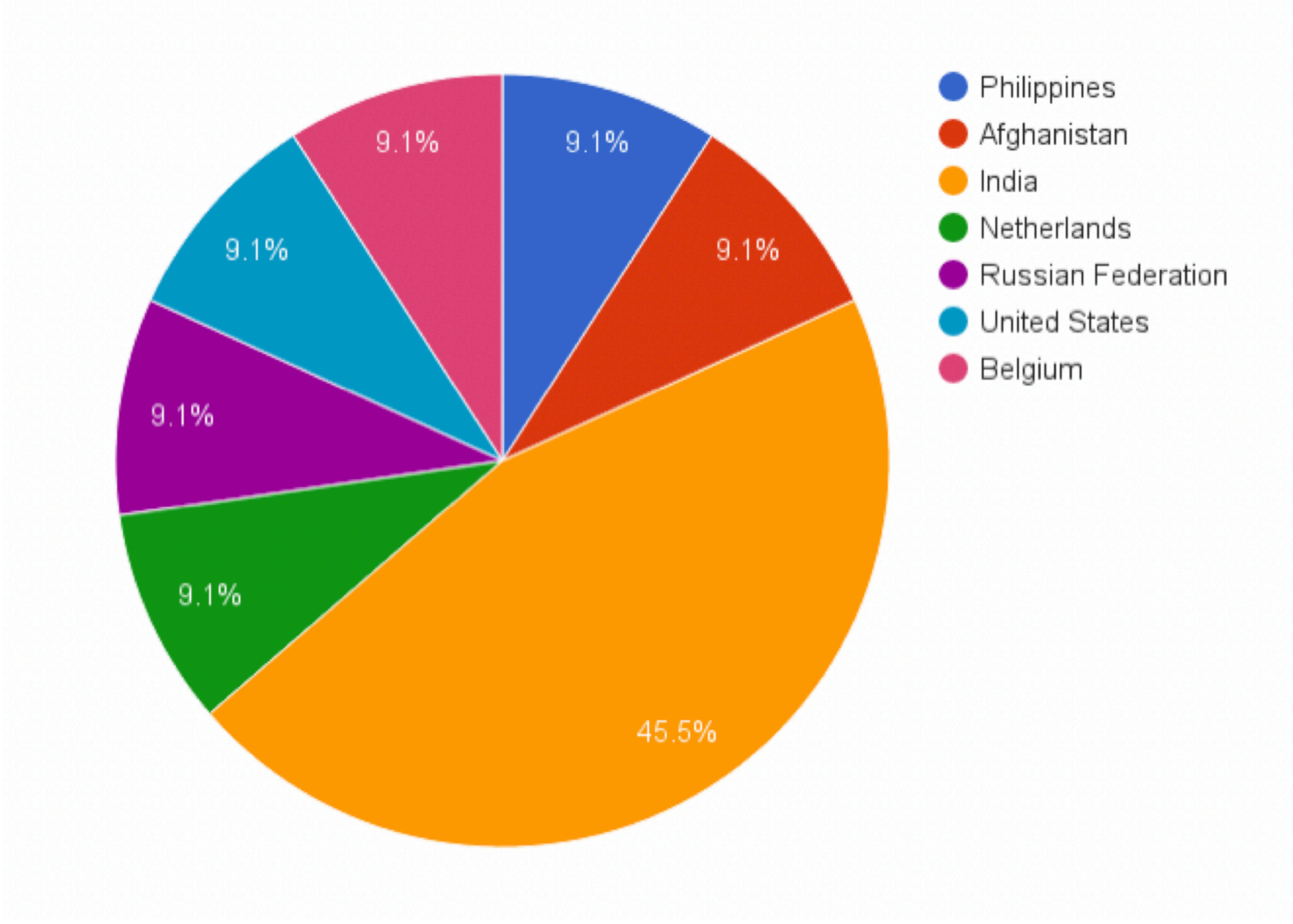
97.5% of all issues are categorised in the OWASP Top 10

# About the Researchers



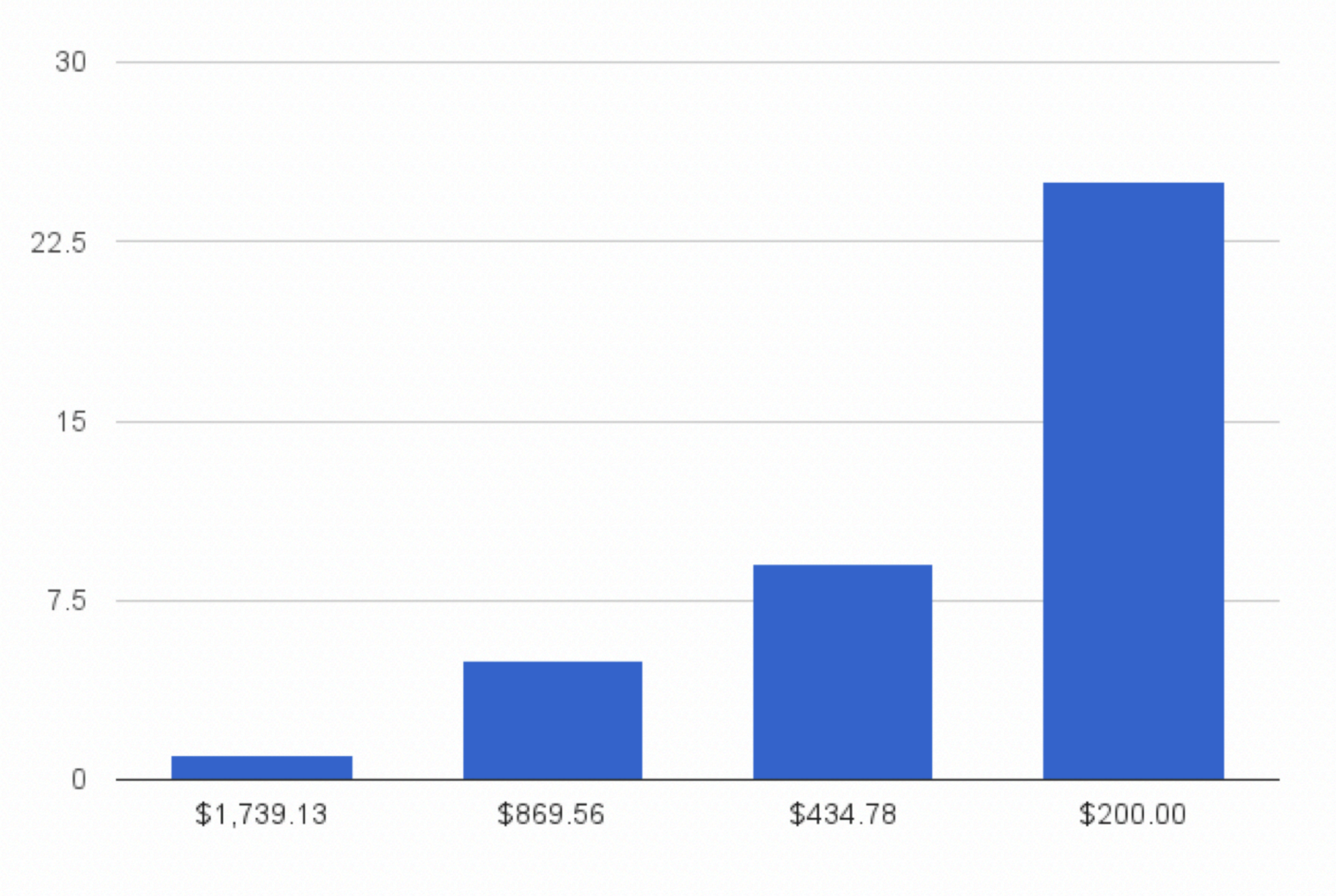
50 researchers were invited, 15 submitted and 12 were valid

# About the Researchers

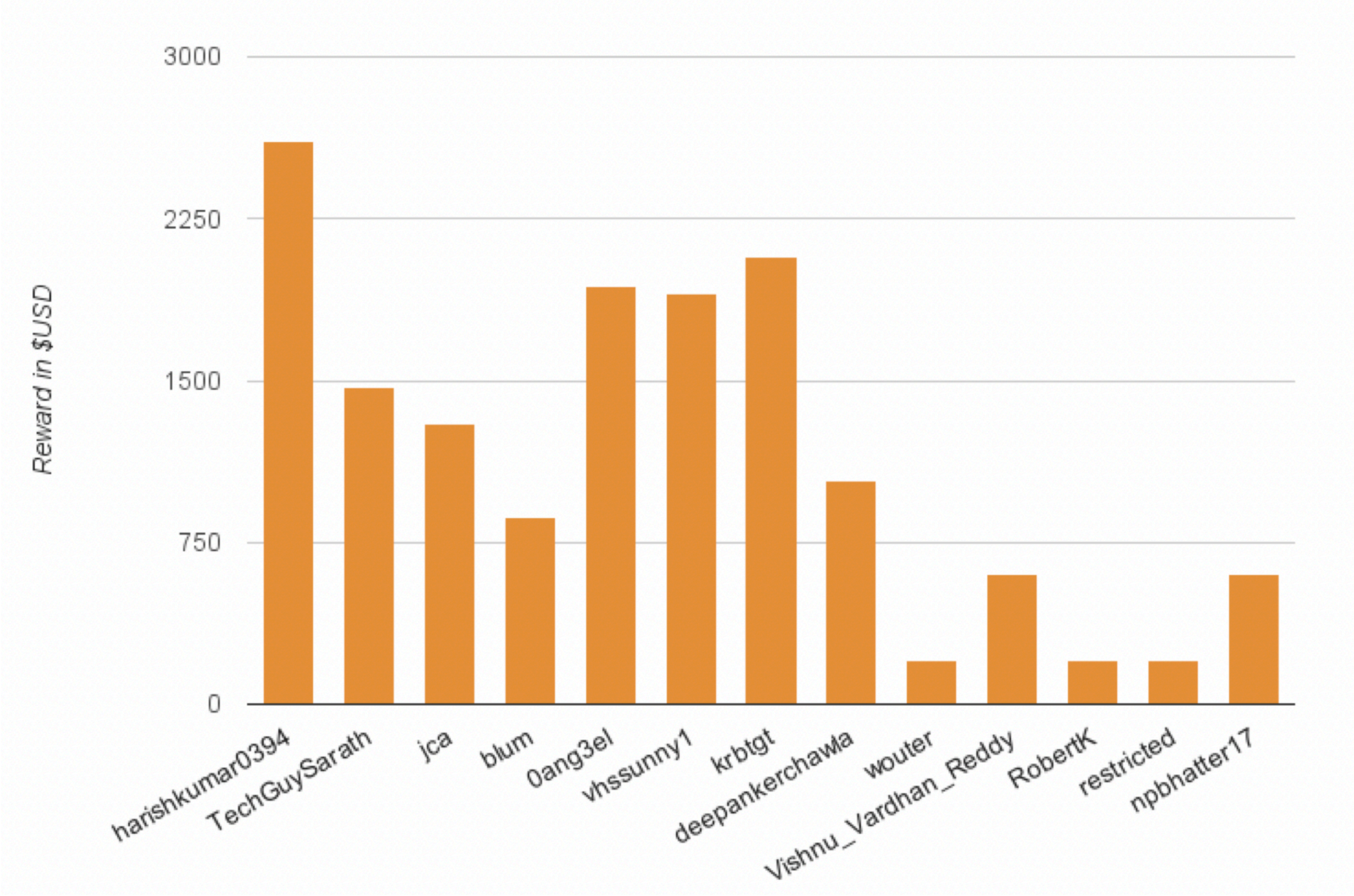


12 researchers who submitted valid issues came from

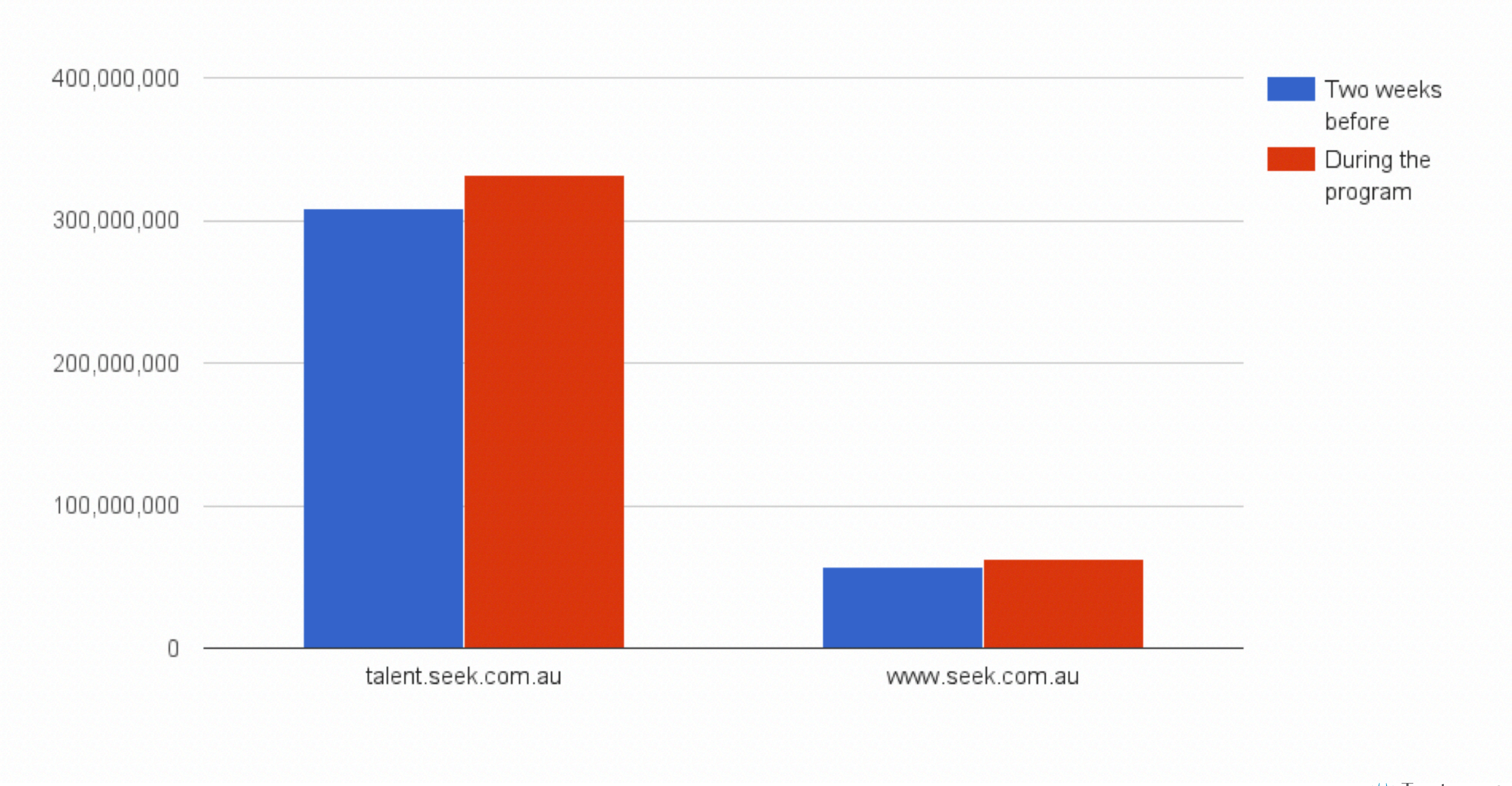
# Reward Distribution



# Reward Distribution



# Traffic


400,000,000
300,000,000
200,000,000
100,000,000
0
talent.seek.com.au
www.seek.com.au
🟦 Two weeks before
🟥 During the program

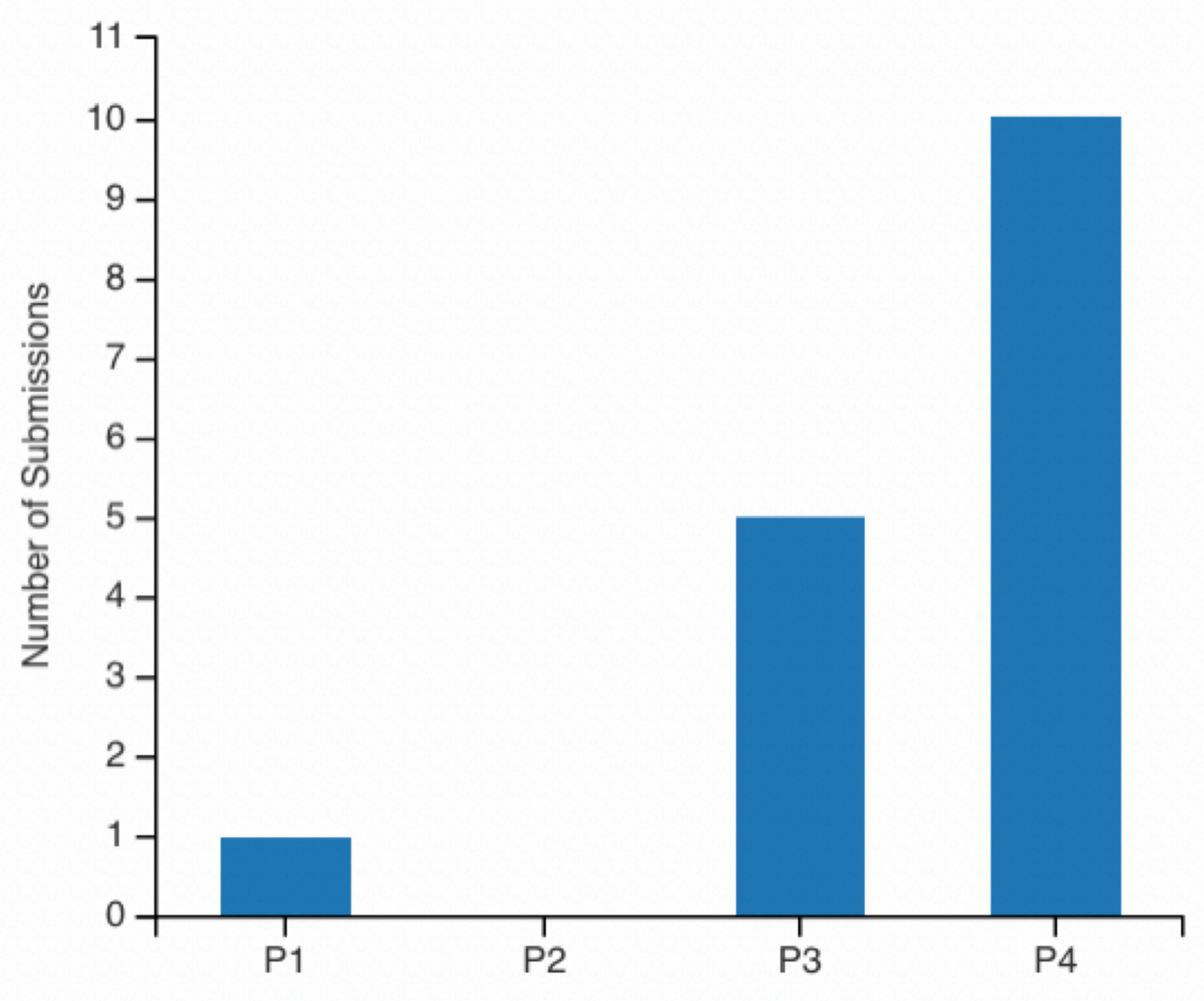
# SEEK’s Private Ongoing Bounty

- Ongoing, private, managed program (started November 2016).
- 50 researchers invited initially.
- Testing occurs on production systems.
- 3 apps in scope + 2 mobile apps.

# Submission Timeline

Number of Submissions
0
1
2
3
4
5
6
7
8
9
10
11
12
13
2016-09-21
2016-10-07
2016-10-23
2016-11-08
2016-11-24
2016-12-10
2016-12-26
2017-01-11
2017-01-27
2017-02-12

# Severity



# Risk Mitigation

## Risk

A researcher could perform testing that brings down or disrupts production (if testing on production systems).

## Mitigation

- Program brief state's Denial of Service on any in scope targets.
- Ban researcher from program. They will stop as they will not get paid and get negative points on the HaaS.
- If you have the ability (e.g. a WAF) you can block the IP address that is causing the issues.
- Use a testing environment for the bug bounty program.

# Risk Mitigation

## Risk

A researcher could interact with real customers and steal real customer data.

## Mitigation

- The brief states not to interact with real customers. Ban researcher from program.
- Existing security controls will prevent most customers being affected.
- Parts of the site that are too hard to test without interacting with customers are taken out of scope.

# Risk Mitigation

## Risk

A researcher could exploit a vulnerability and steal sensitive data.

## Mitigation

- In the brief it states issues should be reported immediately and sensitive data must not be exfiltrated.
- Bonuses are rewarded for getting access to sensitive data and systems, incentivising them to report the issue quickly.

# Risk Mitigation

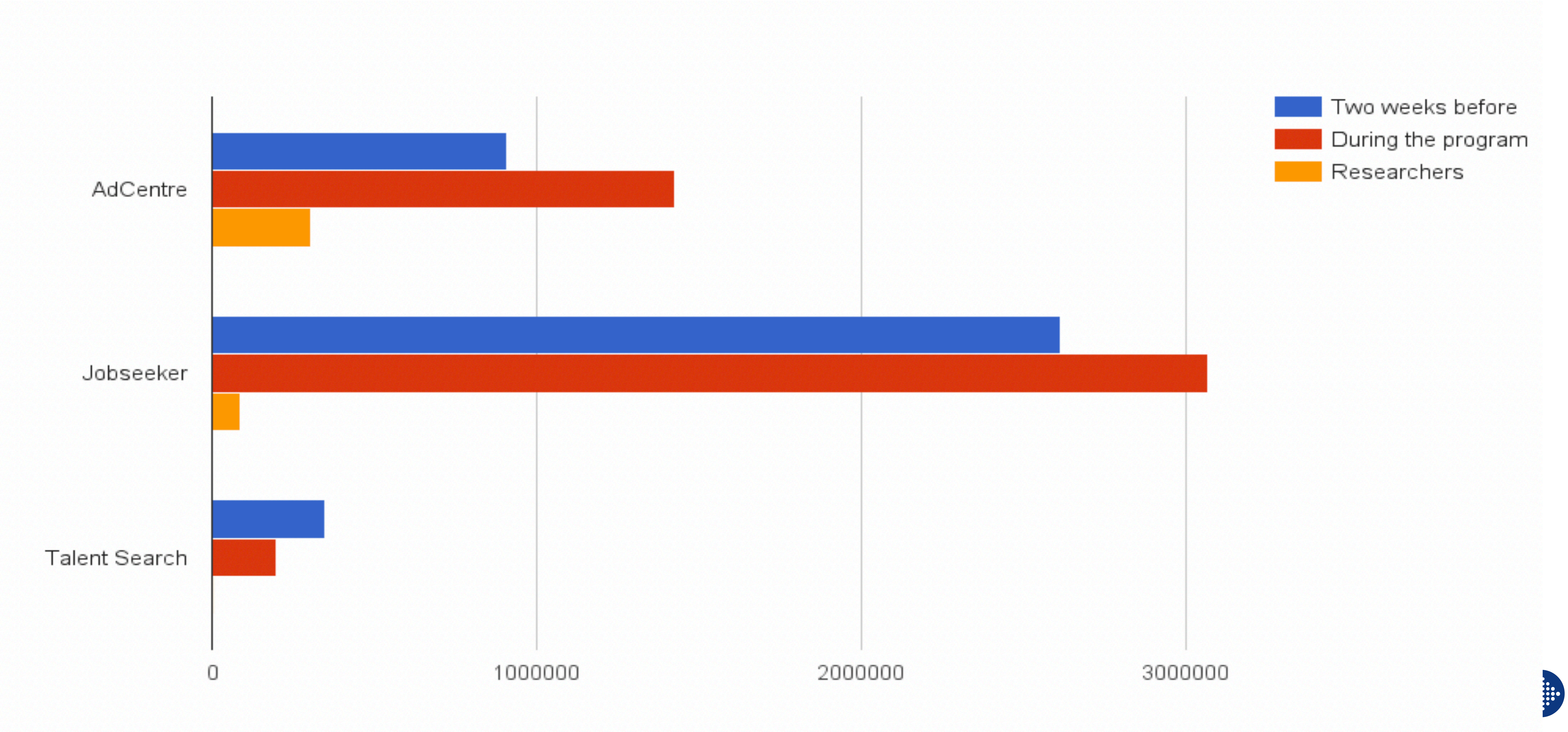
## Risk

A researcher could publicly disclose an issue during or after the program.

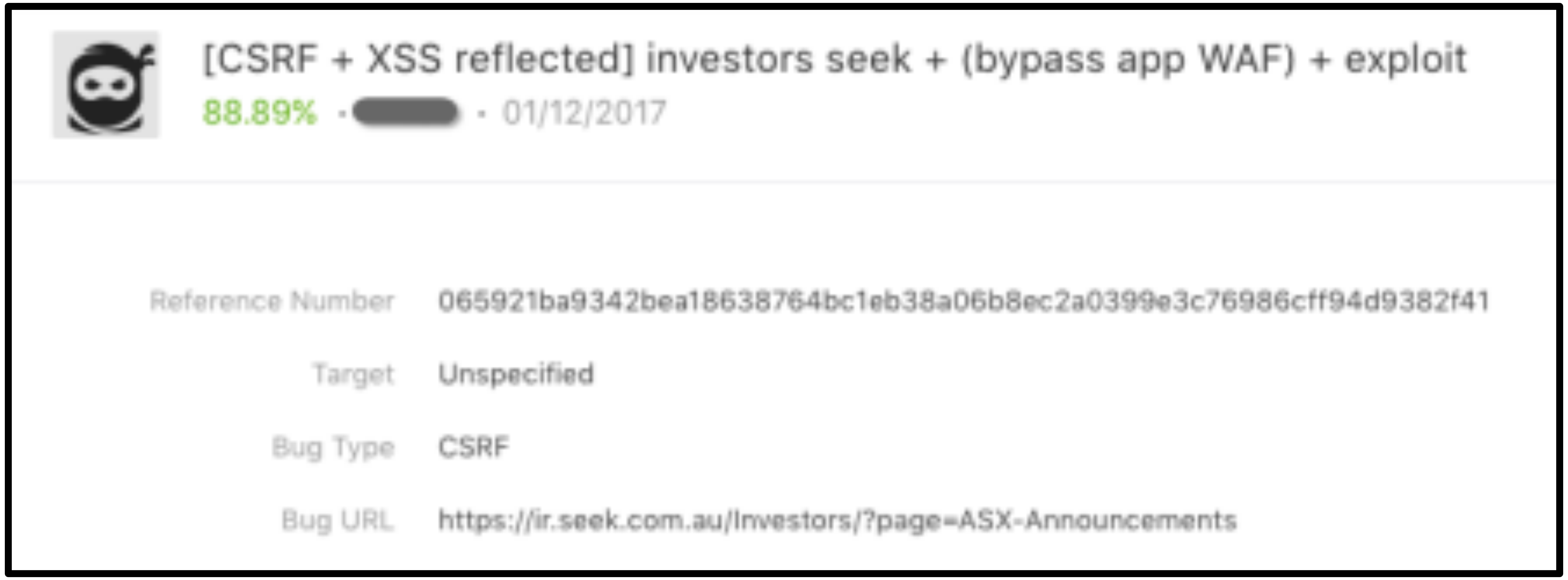
## Mitigation

- They will not receive a reward, will be banned from the program and their reputation score will suffer.
- Ensure that the business is capable and ready to fix reported issues (especially the high issues) as quickly as possible. So that the risk is minimised if it did go public.

# Lessons Learnt - Managing the Crowd



# Lessons Learnt - Managing the Crowd

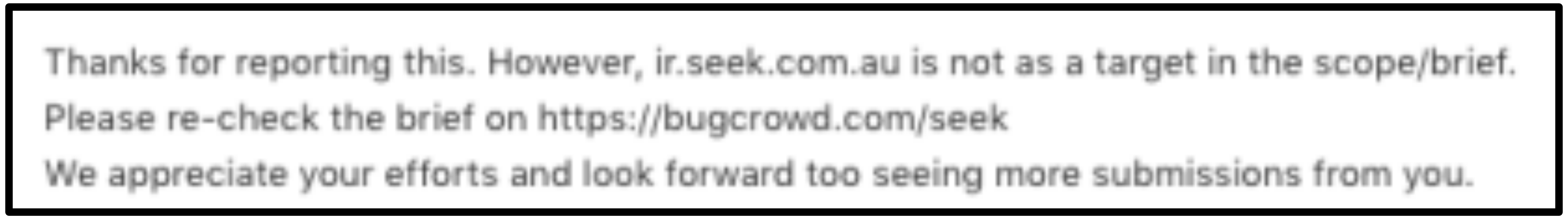


[CSRF + XSS reflected] investors seek + (bypass app WAF) + exploit

88.89% · [redacted] · 01/12/2017

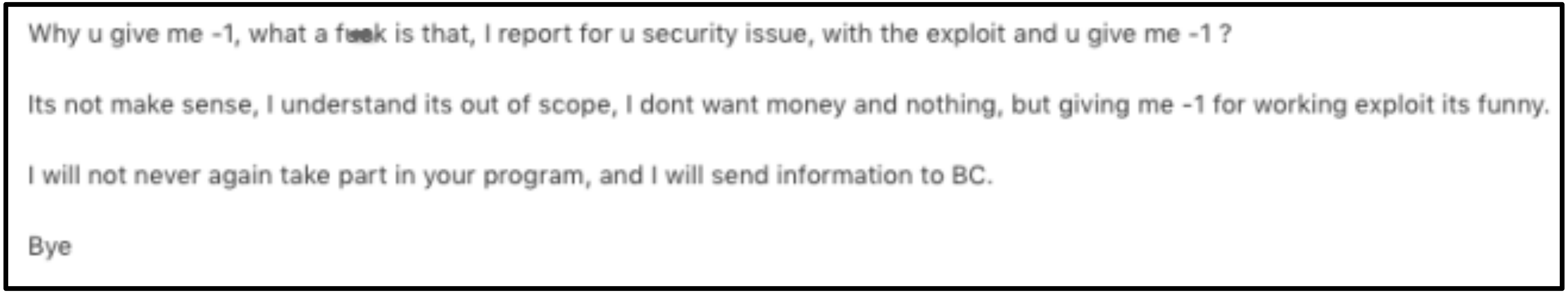
| | |
|---|---|
| Reference Number | 065921ba9342bea18638764bc1eb38a06b8ec2a0399e3c76986cff94d9382f41 |
| Target | Unspecified |
| Bug Type | CSRF |
| Bug URL | https://ir.seek.com.au/Investors/?page=ASX-Announcements |

⇩



Thanks for reporting this. However, ir.seek.com.au is not as a target in the scope/brief.
Please re-check the brief on https://bugcrowd.com/seek
We appreciate your efforts and look forward too seeing more submissions from you.

# Lessons Learnt - Managing the Crowd



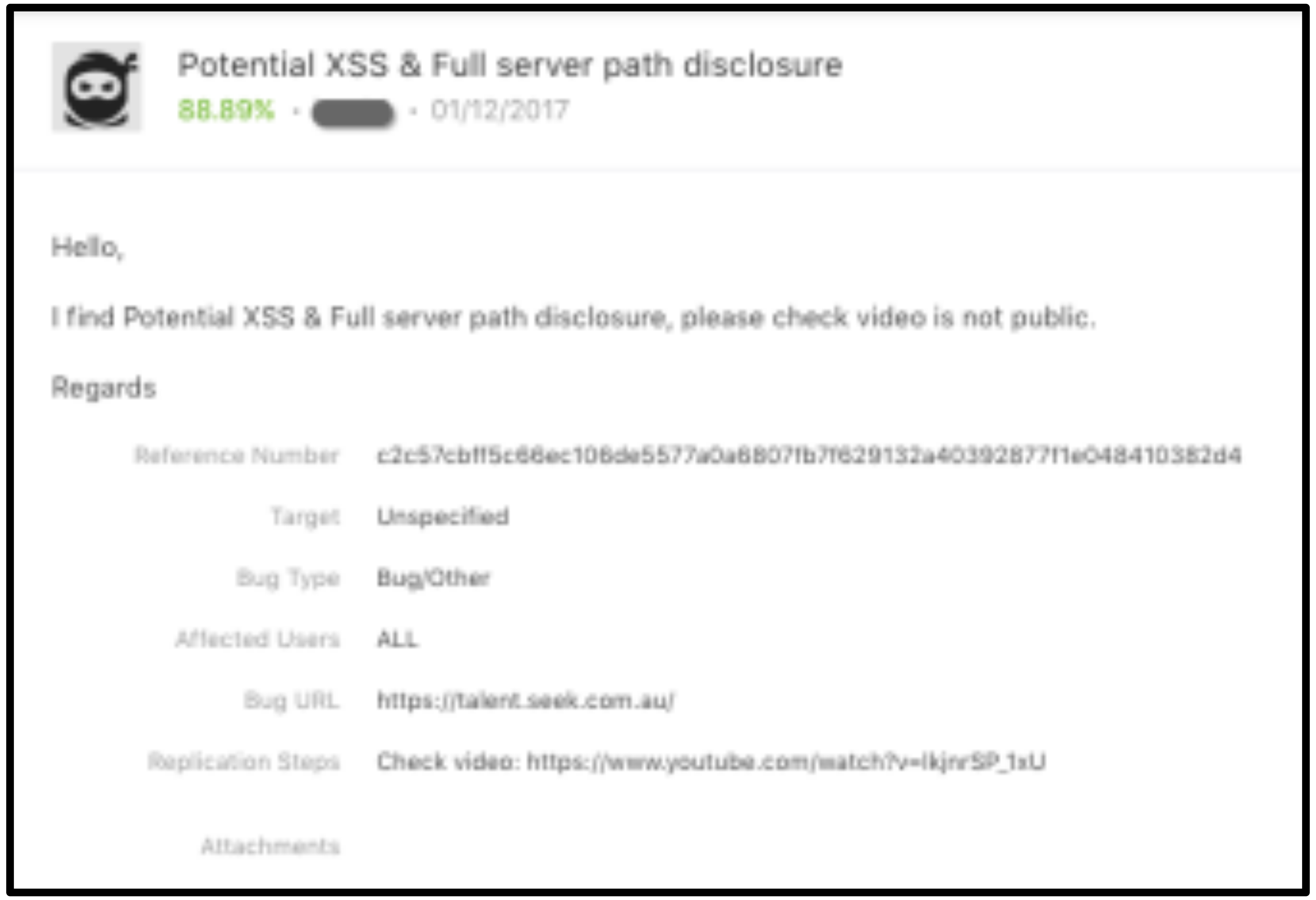
Why u give me -1, what a f[redacted]k is that, I report for u security issue, with the exploit and u give me -1 ?

Its not make sense, I understand its out of scope, I dont want money and nothing, but giving me -1 for working exploit its funny.

I will not never again take part in your program, and I will send information to BC.

Bye

# Lessons Learnt - Managing the Crowd



Potential XSS & Full server path disclosure

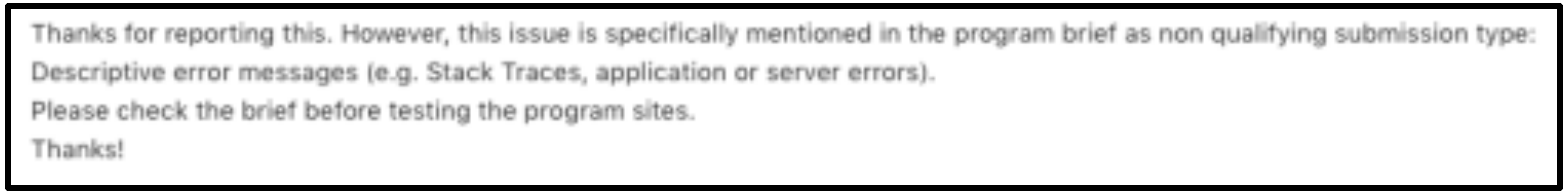
88.89% · [redacted] · 01/12/2017

Hello,

I find Potential XSS & Full server path disclosure, please check video is not public.

Regards

| | |
|---|---|
| Reference Number | c2c57cbff5c66ec106de5577a0a6807fb7f629132a40392877f1e048410382d4 |
| Target | Unspecified |
| Bug Type | Bug/Other |
| Affected Users | ALL |
| Bug URL | https://talent.seek.com.au/ |
| Replication Steps | Check video: https://www.youtube.com/watch?v=lkjnrSP_1xU |
| Attachments | |



Thanks for reporting this. However, this issue is specifically mentioned in the program brief as non qualifying submission type:
Descriptive error messages (e.g. Stack Traces, application or server errors).
Please check the brief before testing the program sites.
Thanks!

# Lessons Learnt

- Limited control over researcher's actions.
- Unsure if attacks were coming from a real hacker or a researcher.
- Keep the program brief as simple as possible.
- Reward bonuses to focus testing on certain applications or issue types.
- Respond to researchers in a reasonable time frame. Even for invalid issues.
- Testers will eventually trigger operational alerts (Prod testing only).

# Revisiting the Economics

- The result-based economic model can be more flexible but it’s not automatically cost-effective
- Marketing from the HaaS providers like to compare bug bounties to point-in-time penetration tests but it’s not a worthwhile comparison - the model is too different
- The common price-per-bug measure is a trap

# Revisiting the Economics

- Given that bounties are ongoing and longer term when modelling the economics of running a program you should use something more akin to Total Cost of Ownership analysis
- Commonly overlooked elements when performing the economic analysis:
    - Management fees (if using a HaaS provider)
    - Internal management of the program (even if using a HaaS provider)
    - Increased load on production equipment and processes
    - Downtime, outage or failure expenses
    - Diminished performance (i.e. opportunity cost if site is slow or down)

# Revisiting the Economics

- Managing the incentives are also not straightforward
  - Have to account for the variability of the payout - the cost is driven by the results (more results = more cost)
  - You are competing with other bounty providers for resources - in a way you become a vendor to the testers
  - Payout size directly influences the quality of the testers and the submissions - in “traditional” pen-testing you might pay more for low-end bugs but you typically pay less for high-end bugs

# Compliance - The Elephant In the Room

- Compliance artificially creates economic incentive to perform testing and drives most of the industry.

- Can be internal (internal audit, policy etc.) or external (PCI, CBEST etc.)

- This is why most of us have jobs.

# Compliance Testing

- Compliance testing is based around assurance and verification
  - Determine that a level of control has been established and maintained
  - This is why the "checklist approach" is so prevalent in compliance based testing and why every QSA asks to see your methodology.

# Compliance Testing

- The incentives in the results-based model don't incentivise testers for compliance testing.
  - Compliance testing is about verification - even if everything is fine or likely to be fine you still need to verify and more importantly evidence compliance with the control objectives.
  - For a bug hunter spending time verifying controls for a company has no ROI vs. chasing the bug.
  - Only way to get around them is to pay them for the verification activities - but then you are back to "traditional" testing.

# Liability

- One of the big hurdles to overcome with this approach for most companies is managing liability.

- Most large organisations have a risk management team and a vendor management team. Bug bounties typically don't make it past there on liability grounds.

- There is a level of risk tolerance required at the moment

# Liability

- Even when using a HaaS where does the liability sit if there is an issue caused by a tester?
  - The standard legal protections (e.g. MSAs, NDAs) do not extend to anonymous testers
  - Enforcing action against anonymous users, cross jurisdiction is probably not possible
  - Liability extends to amount of management contract not the payouts and contracts for most HaaS providers governed by US law

# Liability

- There is still a lot of unanswered questions and ground to cover in this area before more "traditional" organisations get on board.
- The HaaS providers are likely to evolve to meet this problem as they try to target organisations outside generally progressive tech companies
- Will be interesting to see how this develops.

# Bottom Line

# Should I run a bug bounty?

# Maybe

# There is no silver bullet in information security

I feel like we've been over this before…..

# Key Takeaways

- Bug bounties are just one tool that can be used to manage your security risk.

| Training | Inception | Development | Deployment | Monitoring |
|---|---|---|---|---|
| Web security training program for tech teams.<br>Security awareness and improve security culture (i.e. Brown bags, email updates, etc). | Review system design for security weaknesses.<br>Develop attack scenarios for high risk projects. | Add security specific tests into test suite.<br>Adopt security standards and security release plans. | Automate security scanning tools into build pipeline.<br>Automatically scan infrastructure and code for outdated and vulnerable components. | Perform manual security testing for complex or high value components.<br>Implement a continuous testing program (e.g. A bug bounty program). |

# Key Takeaways

- Bug bounties have a lot of inherent benefits but there are a number of considerations that need to be understood and accounted for
- Always evaluate against your requirements
- Don't just blindly follow a HaaS or a pen test provider or any other vendor for that matter - do your homework

# Questions?

**Michael Gianarakis**

@mgianarakis
au.linkedin.com/in/michaelgianarakis
meetup.com/sectalks-brisbane
eightbit.io

**Julian Berton**
@julianberton
au.linkedin.com/in/julianberton
meetup.com/Application-Security-OWASP-Melbourne
bertonjulian.github.io