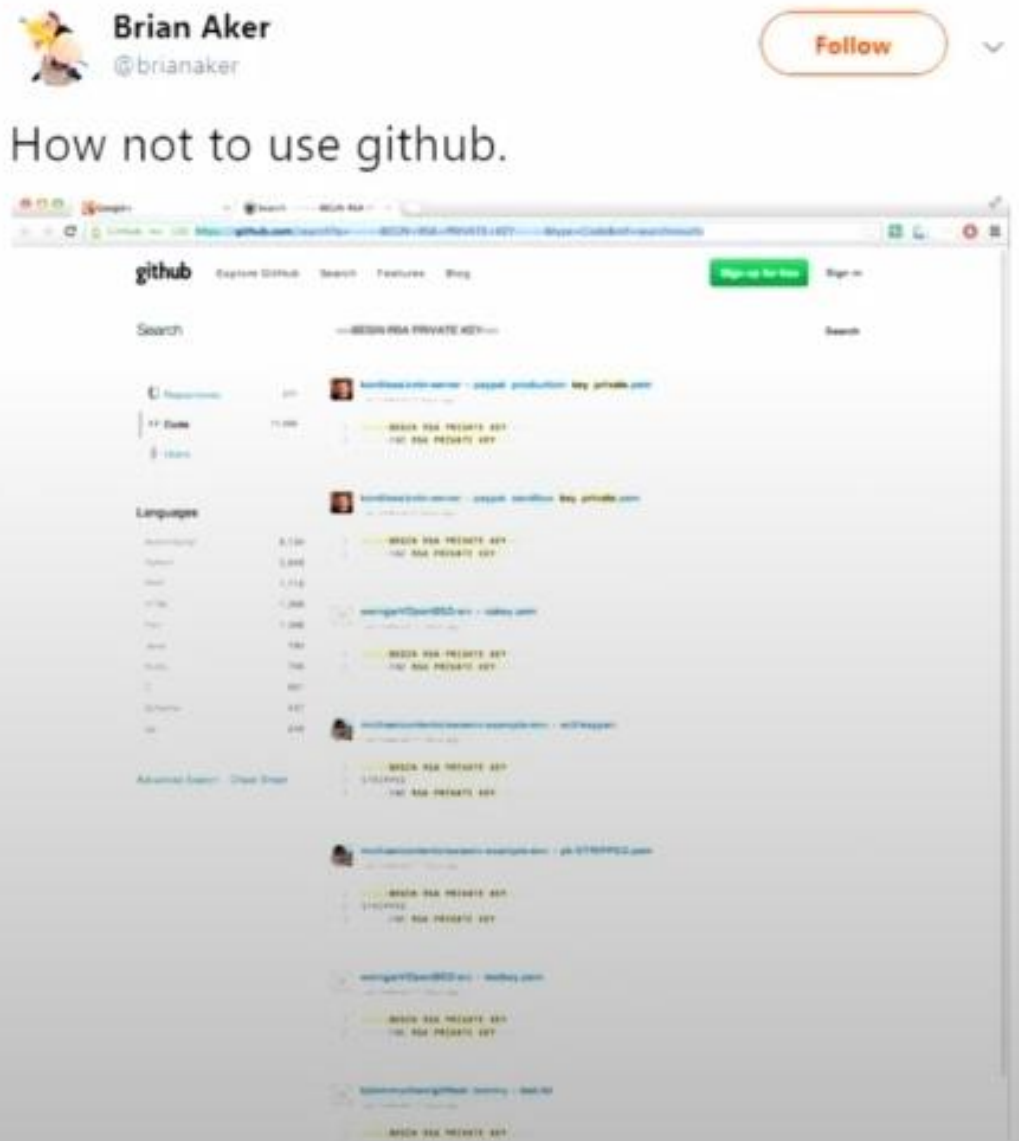
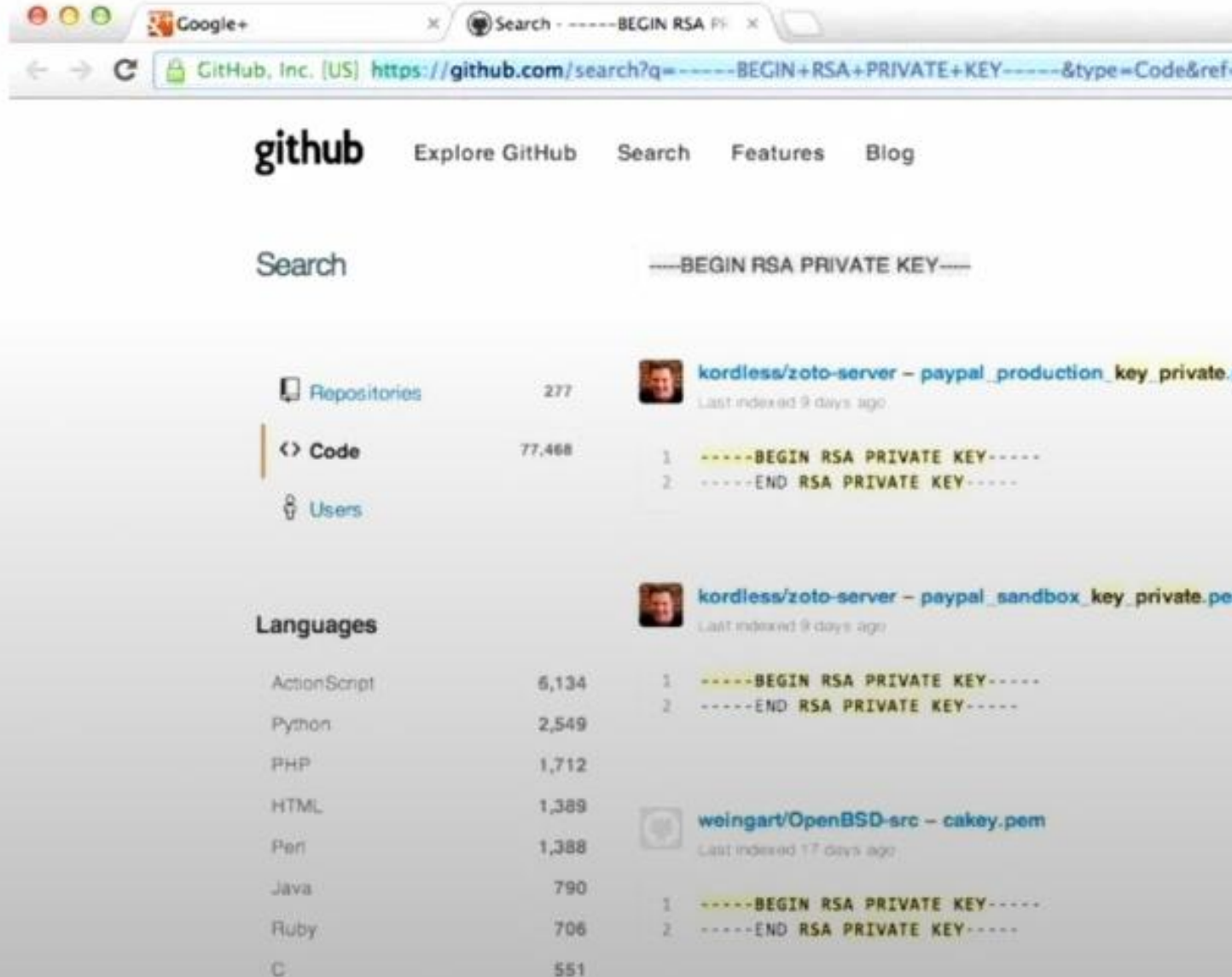
How Bad Can It **Git**?

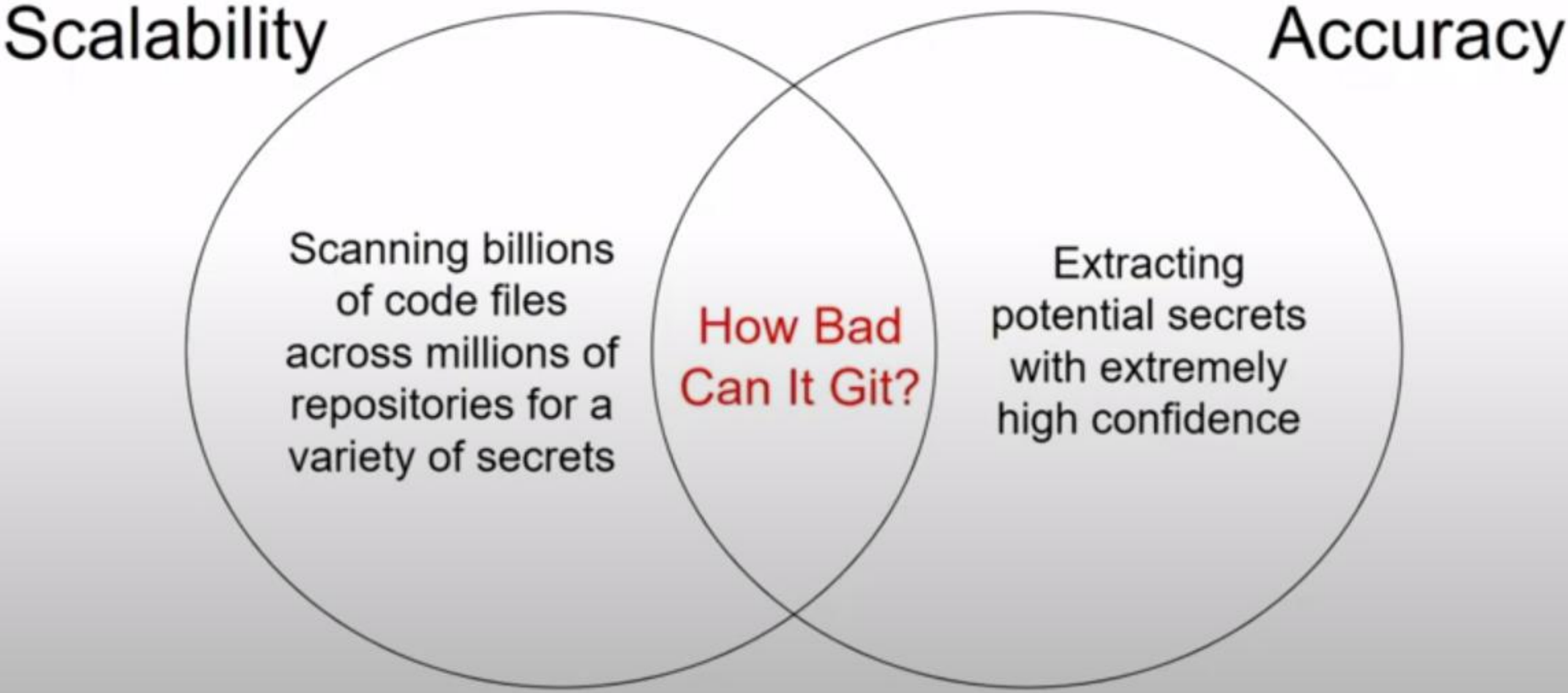
Secret Leakage in Public GitHub Repositories

Jing Liu

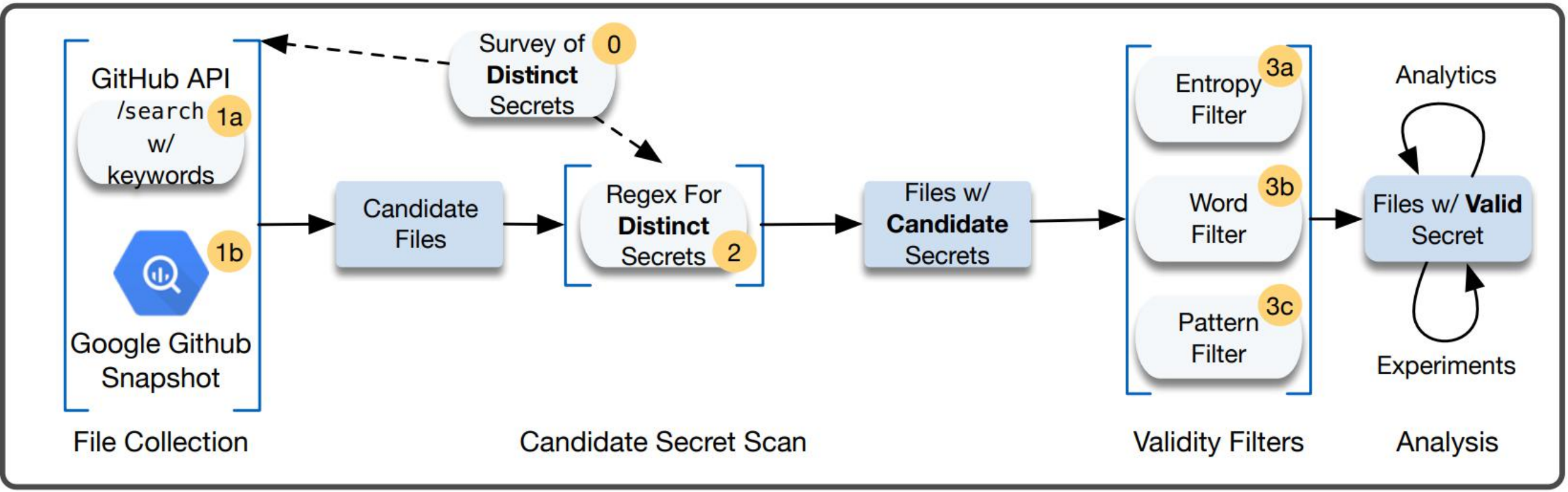
GitHub Code Search






Scalability
Accuracy
Scanning billions of code files across millions of repositories for a variety of secrets
How Bad Can It Git?
Extracting potential secrets with extremely high confidence

various phases to identify secrets



Key Challenge 1: What is secret?

random appearance & unique signature -> small probability of collision

Domain	Platform/API	Key Type	Target Regular Expression
Social Media	Twitter	Access Token	`[1-9][0-9]+-[0-9a-zA-Z]{40}`
	Facebook	Access Token	`EAACEdEose0cBA[0-9A-Za-z]+`
	Google YouTube	API Key	`AIza[0-9A-Za-z\-_]{35}`
		OAuth ID	`[0-9]+-[0-9A-Za-z_]{32}\.apps\.googleusercontent\.com`
	Picatic	API Key	`sk_live_[0-9a-z]{32}`
Finance	Stripe	Standard API Key	`sk_live_[0-9a-zA-Z]{24}`
		Restricted API Key	`rk_live_[0-9a-zA-Z]{24}`
	Square	Access Token	`sq0atp-[0-9A-Za-z\-_]{22}`
		OAuth Secret	`sq0csp-[0-9A-Za-z\-_]{43}`
	PayPal Braintree	Access Token	`access_token\$production\$[0-9a-z]{16}\$[0-9a-f]{32}`
	Amazon MWS	Auth Token	`amzn\.mws\.[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`
Communications	Google Gmail	(see YouTube)	(see YouTube)
	Twilio	API Key	`SK[0-9a-fA-F]{32}`
	MailGun	API Key	`key-[0-9a-zA-Z]{32}`
	MailChimp	API Key	`[0-9a-f]{32}-us[0-9]{1,2}`
Storage	Google Drive	(see YouTube)	(see YouTube)
IaaS	Amazon AWS	Access Key ID	`AKIA[0-9A-Z]{16}`
	Google Cloud Platform	(see YouTube)	(see YouTube)

Key Challenge 2:How to scan at scale

Two complementary approaches

	GitHub Search API	GitHub BigQuery Snapshot
Search Qualifier	Trigger search keywords	Loose regex
Perspective	Continuous latest commits	Historical
Coverage	99% of all public commits in near real time, despite rate limiting	Every public licensed repo
Primary Repo Type	Actively developed	Mature

Key Challenge 3: How to achieve accuracy

Avoid false positives that passed the regex tests
eg. `AKIA[0-9A-Z]{16}` `AKIAXXXEXAMPLEKEYXXX`

- An entropy filter, which catches strings with deviant **shannon entropy**

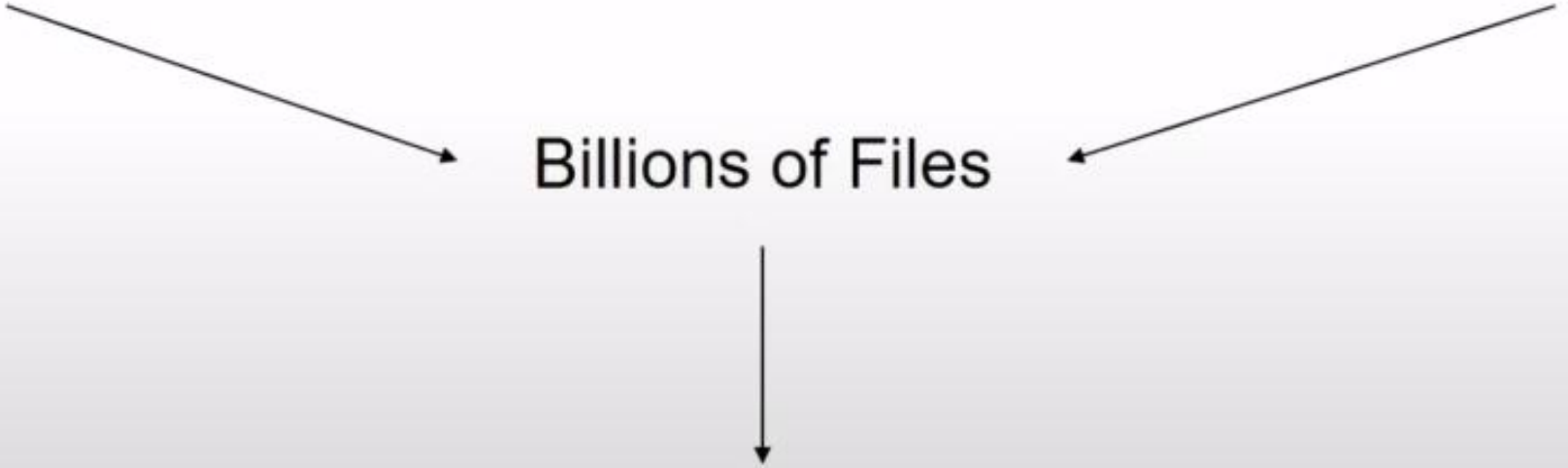
$$H(X) = -\sum_{i=0}^{n} P(x_i) \log_2 P(x_i)$$

- A words filter, which catches strings containing common dictionary words of length at least 5
- A pattern filter looking for repeated characters (e.g. ‘AAAA’), ascending characters (‘ABCD’) and descending characters (‘DBCA’)

What did we find?

6 months of continuous Search API scanning

BigQuery snapshot of all of GitHub as of **April 2018**



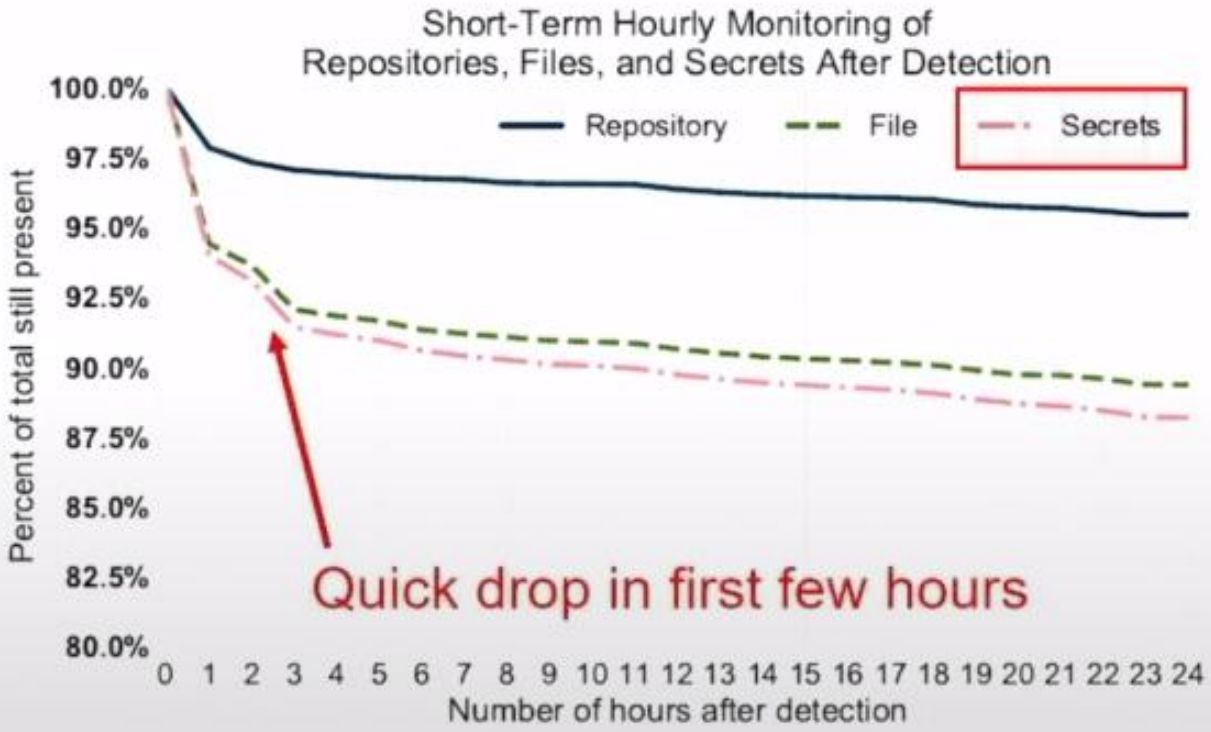
4,394,476 code files total downloaded for offline analysis

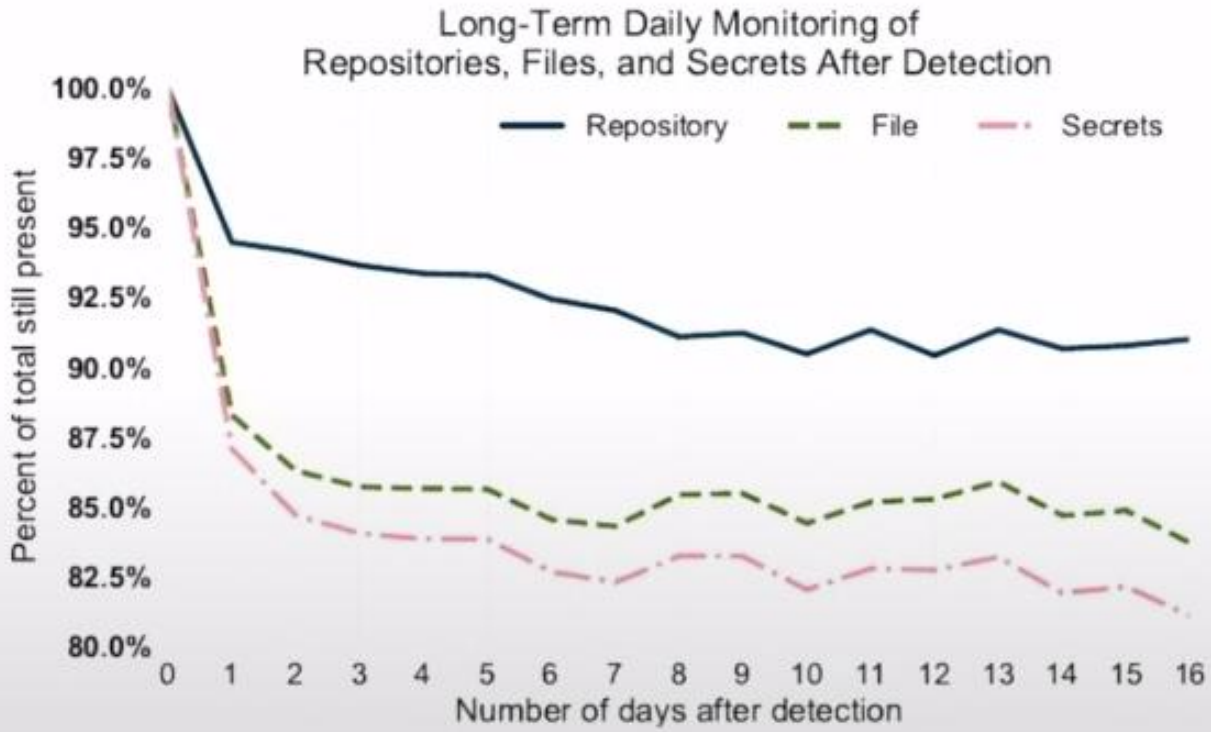
201,642 total unique valid secrets found

1,793 median secrets found **per day**

Secret	# Total	# Unique	% Single-Owner
Google API Key	212,892	85,311	95.10%
RSA Private Key	158,011	37,781	90.42%
Google OAuth ID	106,909	47,814	96.67%
General Private Key	30,286	12,576	88.99%
Amazon AWS Access Key ID	26,395	4,648	91.57%
Twitter Access Token	20,760	7,935	94.83%
EC Private Key	7,838	1,584	74.67%
Facebook Access Token	6,367	1,715	97.35%
PGP Private Key	2,091	684	82.58%
MailGun API Key	1,868	742	94.25%
MailChimp API Key	871	484	92.51%
Stripe Standard API Key	542	213	91.87%
Twilio API Key	320	50	90.00%
Square Access Token	121	61	96.67%
Square OAuth Secret	28	19	94.74%
Amazon MWS Auth Token	28	13	100.00%
Braintree Access Token	24	8	87.50%
Picatic API Key	5	4	100.00%
TOTAL	**575,456**	**201,642**	**93.58%**

Do users realize when they commit secrets?





*Most secrets **are never removed** from the default branch,*
*and many that are **can still be found in Git history***

Why does this happen?

Common hypothesis: developer inexperience

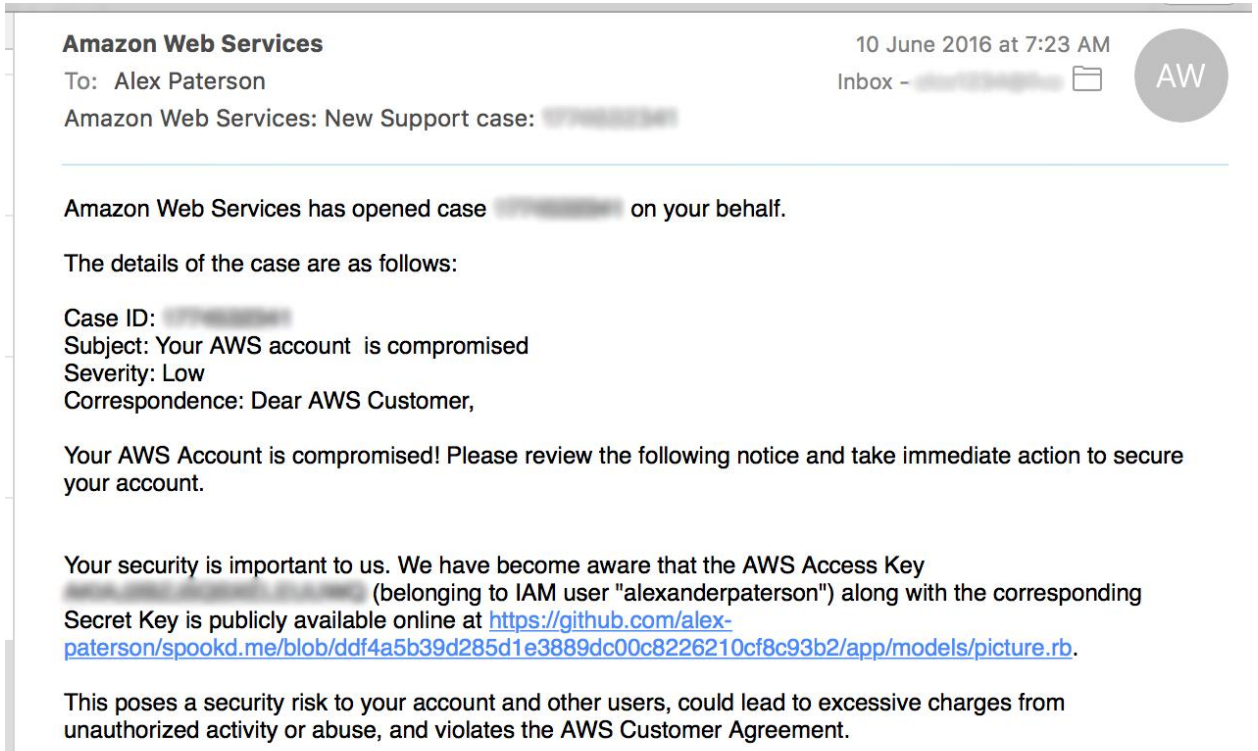
We conducted statistical tests across the following variables:

- Developer experience
- Repo maturity
- Repo activity (contributors, contributions, watchers, etc.)

*We found **no statistically significant difference** between any available variable and leakage. It happens to **everyone** in **every type** of project, regardless of experience.*

New Situations

- Microsoft acquired GitHub, bringing free private repos
- China-US trade war makes Gitee an alternative
- COVID-19 has caused growing demand of remote development
- Tech giants cooperate to scan repositories for leaked API keys



Amazon Web Services 10 June 2016 at 7:23 AM

To: Alex Paterson Inbox - AW

Amazon Web Services: New Support case:

Amazon Web Services has opened case on your behalf.

The details of the case are as follows:

Case ID:
Subject: Your AWS account is compromised
Severity: Low
Correspondence: Dear AWS Customer,

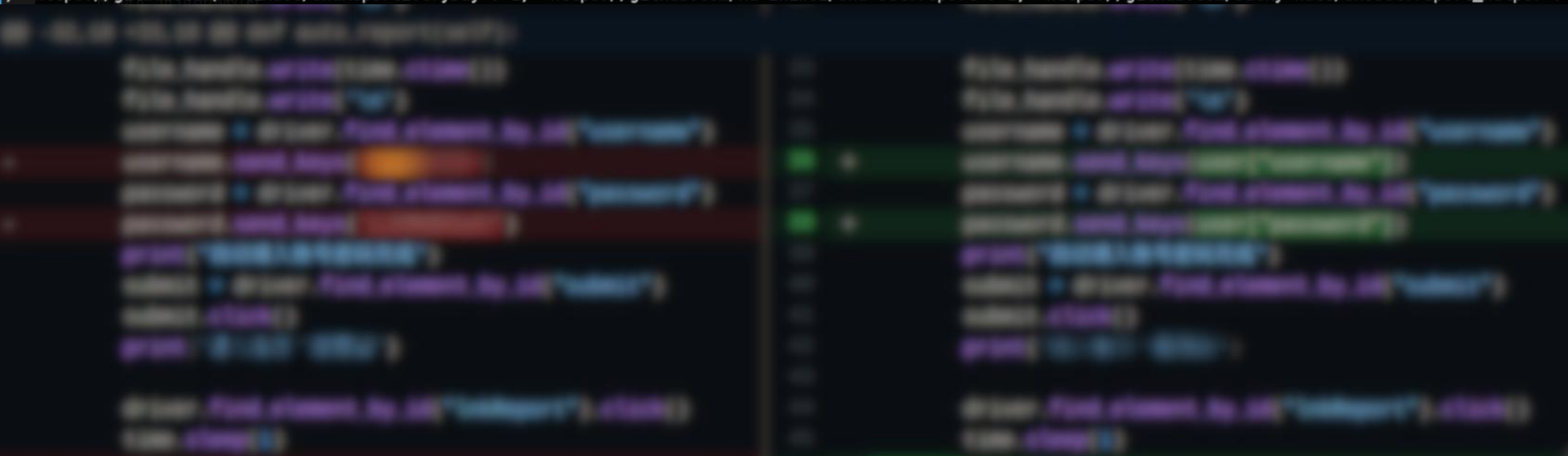
Your AWS Account is compromised! Please review the following notice and take immediate action to secure your account.

Your security is important to us. We have become aware that the AWS Access Key (belonging to IAM user "alexanderpaterson") along with the corresponding Secret Key is publicly available online at https://github.com/alex-paterson/spookd.me/blob/ddf4a5b39d285d1e3889dc00c8226210cf8c93b2/app/models/picture.rb.

This poses a security risk to your account and other users, could lead to excessive charges from unauthorized activity or abuse, and violates the AWS Customer Agreement.

Not limited to API Keys...
You are not safe!

```
total_count (round to 100): 500
Found 182 targets: defaultdict(<class 'int'>, {'https://github.com/dislazy/blog': 1, 'https://github.com/servantbi/photo': 1, 'https://github.com/Trouvaille0198/SHU-course-helper': 1, 'https://github.com/sls-652/SelectCourse': 1, 'https://github.com/khs1994-docker/lnmp': 1, 'https://github.com/DongZhouGu/SHU-self report-mail': 2, 'https://github.com/lework/kainstall': 2, 'https://github.com/yhyDewily/shu_homework_platform': 1, 'https://github.com/Trouvaille0198/litttle-site': 1, 'https://github.com/GuoJuna/blog': 1, 'https://github.com/xuxianDe/SHU_HealthReport': 1, 'https://github.com/Steve235lab/Auto_SelfReport-for-SHU': 2, 'https://github.com/joeky888/dotfile': 1, 'https://github.com/crazyhubox/RoomUse': 1, 'https://github.com/crazyhubox/Shu_pwKey': 1, 'https://github.com/hidacow/SHU-CourseHelper': 1, 'https://github.com/BlueFisher/SHU-selfreport': 2, 'https://github.com/wyp2019/-': 2, 'https://github.com/Silicon-He/SHU-lessons-helper': 1, 'https://github.com/crazyhubox/SHU_report_public': 3, 'https://github.com/chinggg/AutoSHU': 1, 'https://github.com/fansichao/blog-csdn01': 1, 'https://github.com/Lanszhang131/DailyReport_SHU': 1, 'https://github.com/zsksmhq/dailyReport': 1, 'https://github.com/panghaibin/shuasr': 1, 'https://github.com/Pcrab/SHU-Daily-Report': 1, 'https://github.com/Menamot/Daliy_Report': 2, 'https://github.com/Conanzhanghz/HealthReportPython': 3, 'https://github.com/Ezreal147/reportshu': 2, 'https://github.com/Shu-Huai/SHU-Self-Report': 2, 'https://github.com/Microdust12/script': 1, 'https://github.com/DanicCheng/blog': 2, 'https://github.com/mrlixuec/shuReportEveryDay': 1, 'https://github.com/xu-zhiwei/shu-selfreport': 1, 'https://github.com/Jacky-hate/SHUselfreport_helper':
```



Suggestions

- Consider private repositories
- Use Gitee instead of GitHub
- Use environment variables instead of hard-coded passwords
- Install tools like git-secret to prevent commiting secrets

Open Source != Open Secret

Ethical Statement

- First and foremost, we only work with **publicly available** data,not private data or data derived from interaction with human participants.
- Second, apart from our search queries, our methodology is **passive**. All secrets that we collect were already exposed when we find them, thus this research does not create vulnerabilities
- Furthermore, we **never attempt to use** any of the discovered secrets other than for the analytics

Thanks for listening