

**Heartbleed is discovered at Google,
the bug was introduced in 2012**

~500 000 websites are open to attack

Fixed openssl library is released

**The Canada Revenue Agency reports a theft of
Social Insurance Numbers belonging to 900
taxpayers**

```
→ client git:(master) g
```



Effects of high-profile incidents on code

Felix Wolff

Master seminar *Code Repository Mining*

23. January 2018 | winter term 2017/2018

Agenda: Creating a system weakness scanner



- Recap: CVE, CWE and data
- Client-Server architecture
- Data analysis & design choices
- Data procurement & structure
- Future Work
- Competition

Recap: CVE, CWE and data

Common Vulnerabilities and Exposures (CVE)

- ID-based (CVE-2014-0160) – for a vulnerability affecting some product
- NIST entry reveals CWE, CPE, descriptions and references

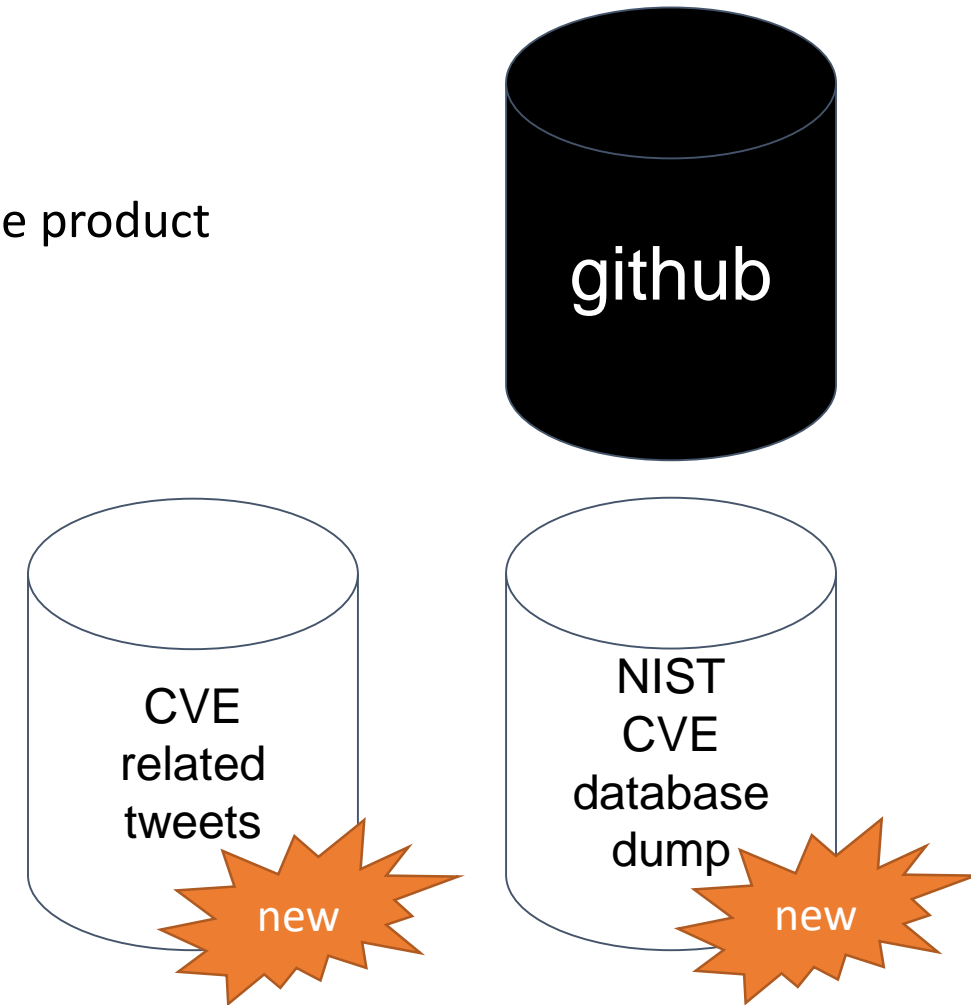
Common Weakness Enumeration (CWE)

- ID-based (CWE-119)
- NIST entry reveals description, examples

Common Platform Enumeration (CPE)

- `cpe:2.3:a:openssl:openssl:1.0.1:*:*:*:*:*:*`

National Institute of Standards and Technology (NIST)



Recap: CVE, CWE and data

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

Hyperlink	Resource	Type	Source	Name
http://it.slashdot.org/comments.pl?sid=4821073&cid=46310187		External Source	MISC	http://it.slashdot.org/comments.pl?sid=4821073&cid=46310187
http://support.apple.com/kb/HT6146	Vendor Advisory	External Source	CONFIRM	http://support.apple.com/kb/HT6146
http://support.apple.com/kb/HT6147	Vendor Advisory	External Source	CONFIRM	http://support.apple.com/kb/HT6147
http://support.apple.com/kb/HT6148	Vendor Advisory	External Source	CONFIRM	http://support.apple.com/kb/HT6148
http://support.apple.com/kb/HT6150		External Source	CONFIRM	http://support.apple.com/kb/HT6150
https://news.ycombinator.com/item?id=7281378		External Source	MISC	https://news.ycombinator.com/item?id=7281378
https://www.cs.columbia.edu/~smb/blog/2014-02/2014-02-23.html		External Source	MISC	https://www.cs.columbia.edu/~smb/blog/2014-02/2014-02-23.html
https://www.cs.columbia.edu/~smb/blog/2014-02/2014-02-24.html		External Source	MISC	https://www.cs.columbia.edu/~smb/blog/2014-02/2014-02-24.html
https://www.imperialviolet.org/2014/02/22/applebug.html	Exploit	External Source	MISC	https://www.imperialviolet.org/2014/02/22/applebug.html

Technical Details

Vulnerability Type [\(View All\)](#)

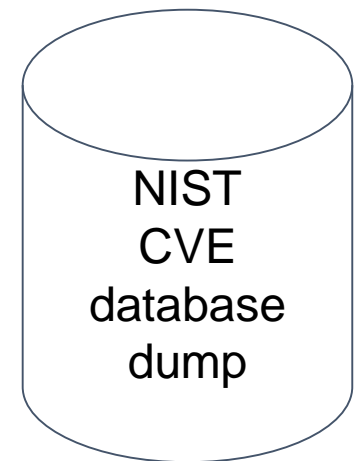
- Input Validation [\(CWE-20\)](#)

Vulnerable software and versions [Switch to CPE 2.2](#)

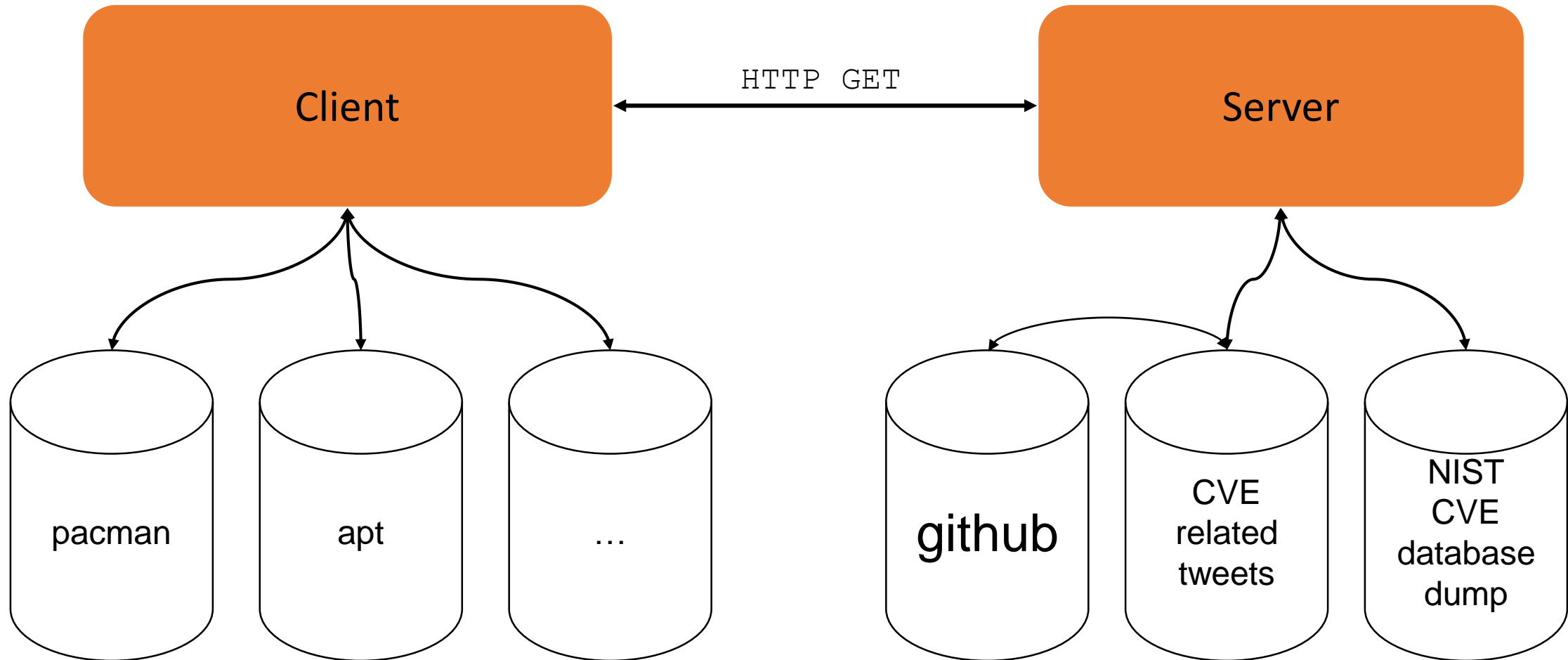
+ Configuration 1

+ OR

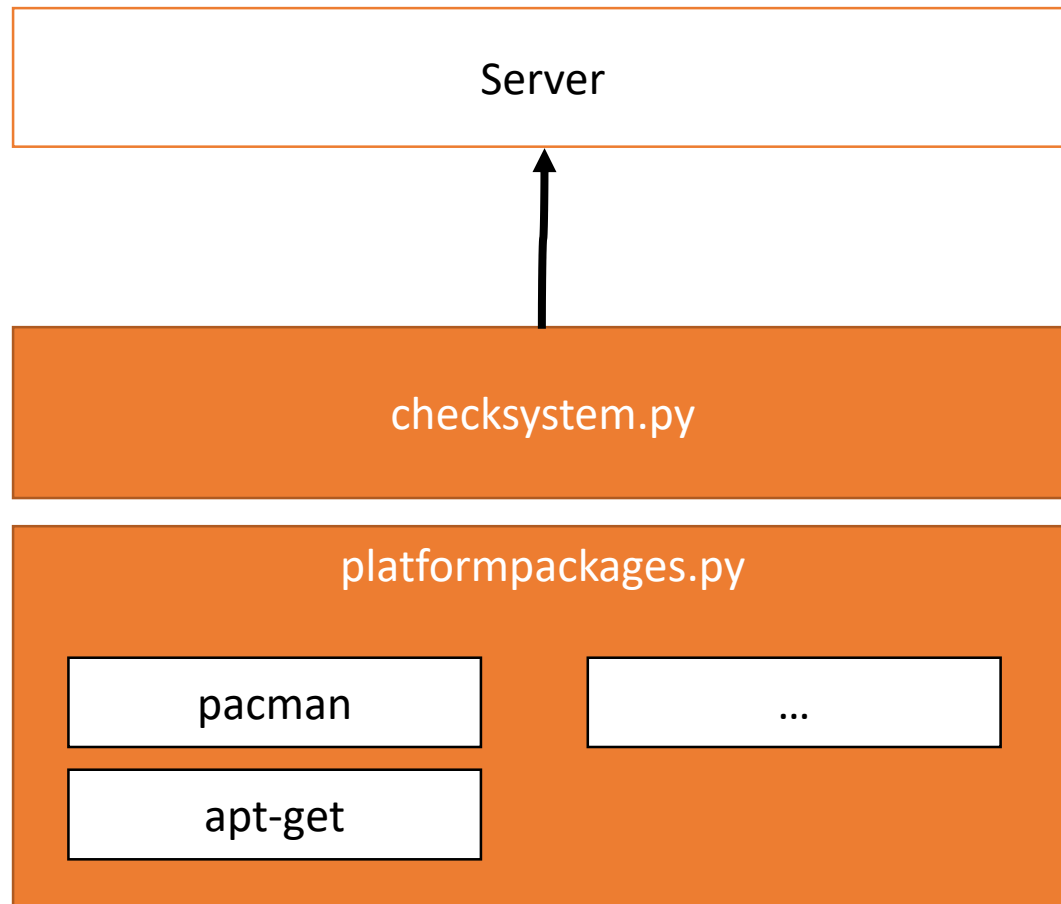
* cpe:2.3:o:apple:iphone_os:6.0:*:*:*:*:*



Client-Server architecture

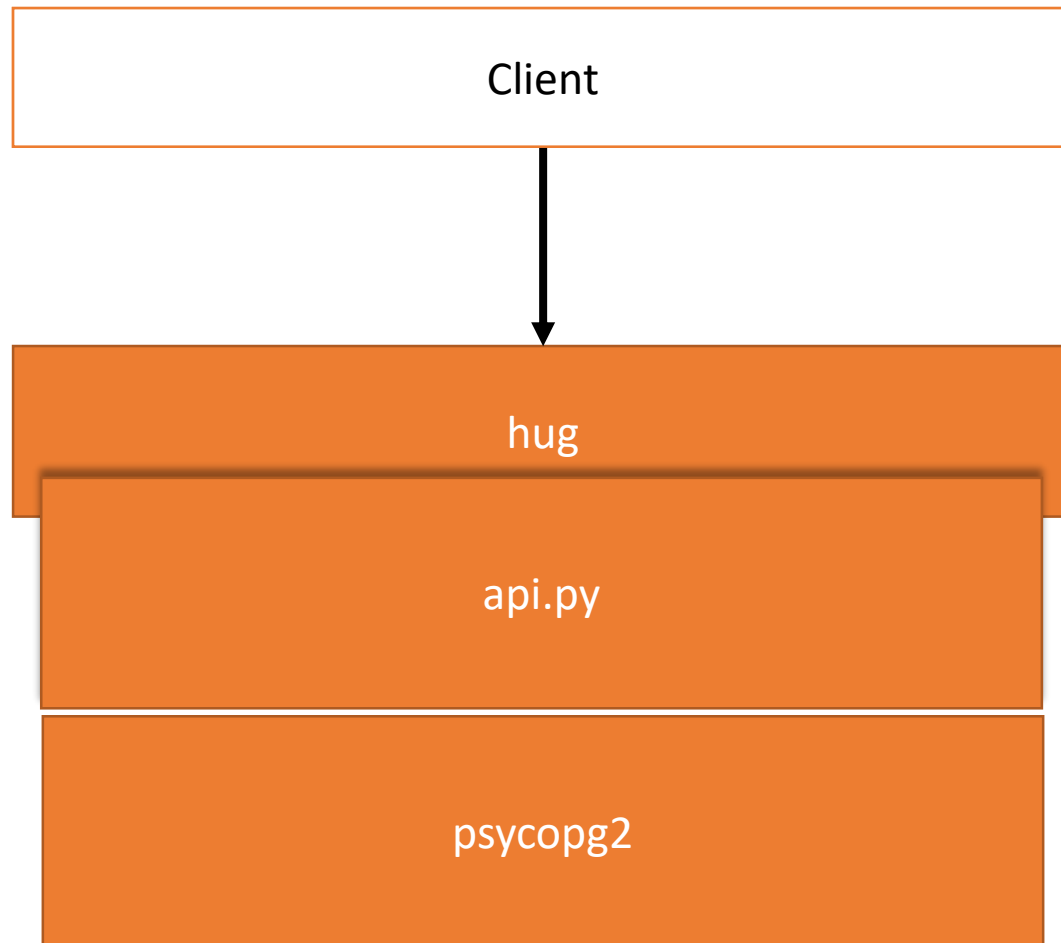


Client



- Python 3
- Performs three simple functions
 - Gather installed packages
 - Send HTTP requests
 - Print results
- Easily extensible for other platforms

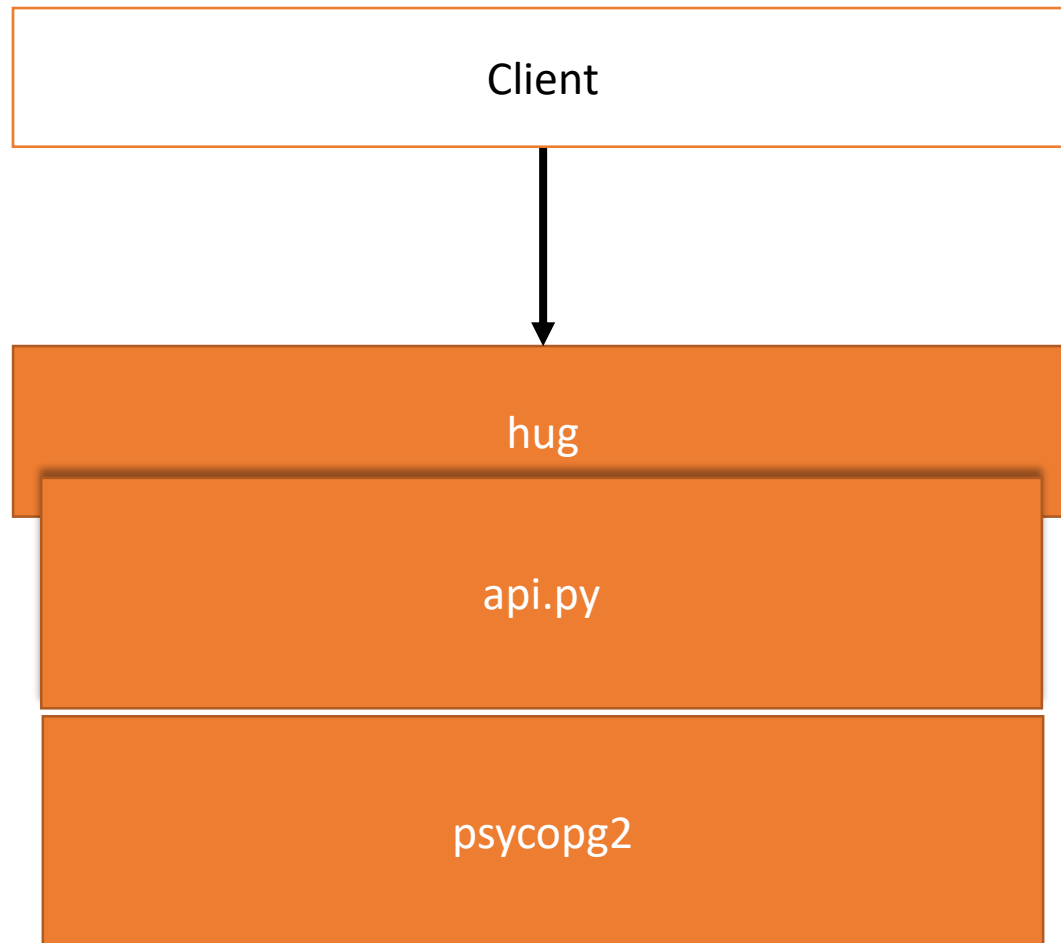
Server



- Python 3, HTTP-API via hug
- Answers client requests via multiple SQL queries
- Mean response time: 5s

```
@hug.get('/product-weaknesses')
def product_weaknesses(name, version):
    return_info = { product_name: name...
    # ... abridged...
    return return_info
```

Server: Request handling



- for an incoming request for product `p` and version `v1`
 - query all CVE `c` and version `v2` combinations that are known for this product
 - iterate over results
 - For each combination `(c,v2)` check for match of `v1` with `v2`
 - Save matches
- query information for each matched CVE
 - if CVE has CWE set, try to fetch additional info
 - try to query source recommendation
 - try to query user recommendation

Server: Implementation challenges

Version comparison

`1.0.1beta1 == 1.0.1beta2 ?`

Luckily Python offers help:

```
from pkg_resources import  
parse_version  
install = parse_version('1.0.1beta1')  
cve_ver = parse_version('1.0.1beta2')  
install == cve_ver # now possible
```

Speed-relevant drawback:

A part of what could have been processed in the database now is inside the application layer!

Product name matching

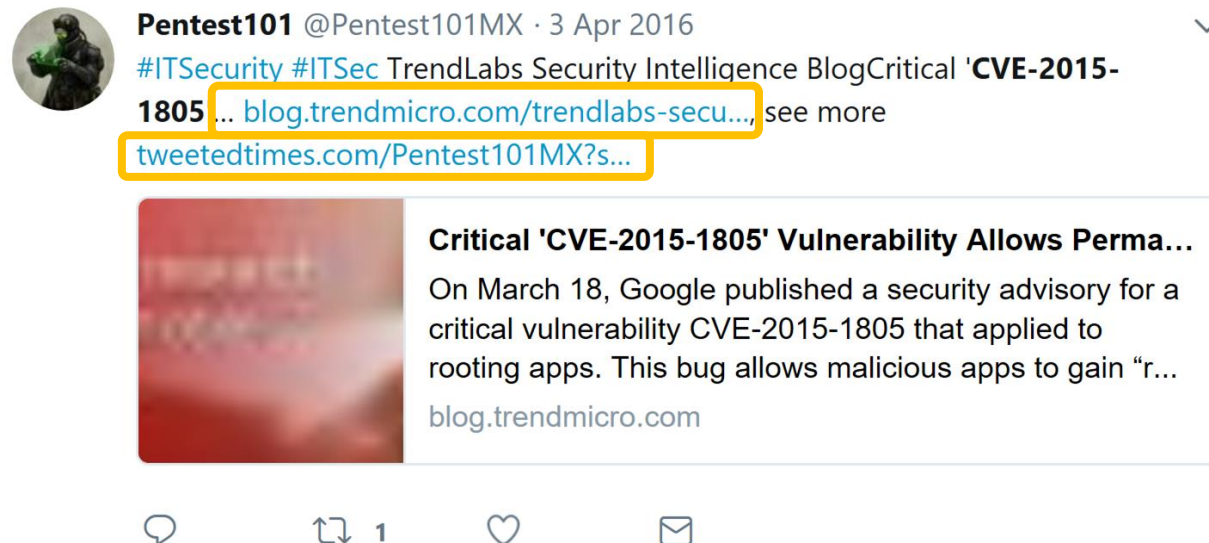
```
SELECT cveid,name,version  
FROM cve_per_product_version  
WHERE name = '{0}'
```

- Only absolute product name matches are considered
 - CVEs are reported for `linux_kernel` – Arch Linux calls it `linux`
- Needs reliable heuristic to avoid false positives
 - Experiments with Levenshtein/Edit-distance thresholds proved to be error-prone

Data analysis & design decisions

Which sources do we recommend?

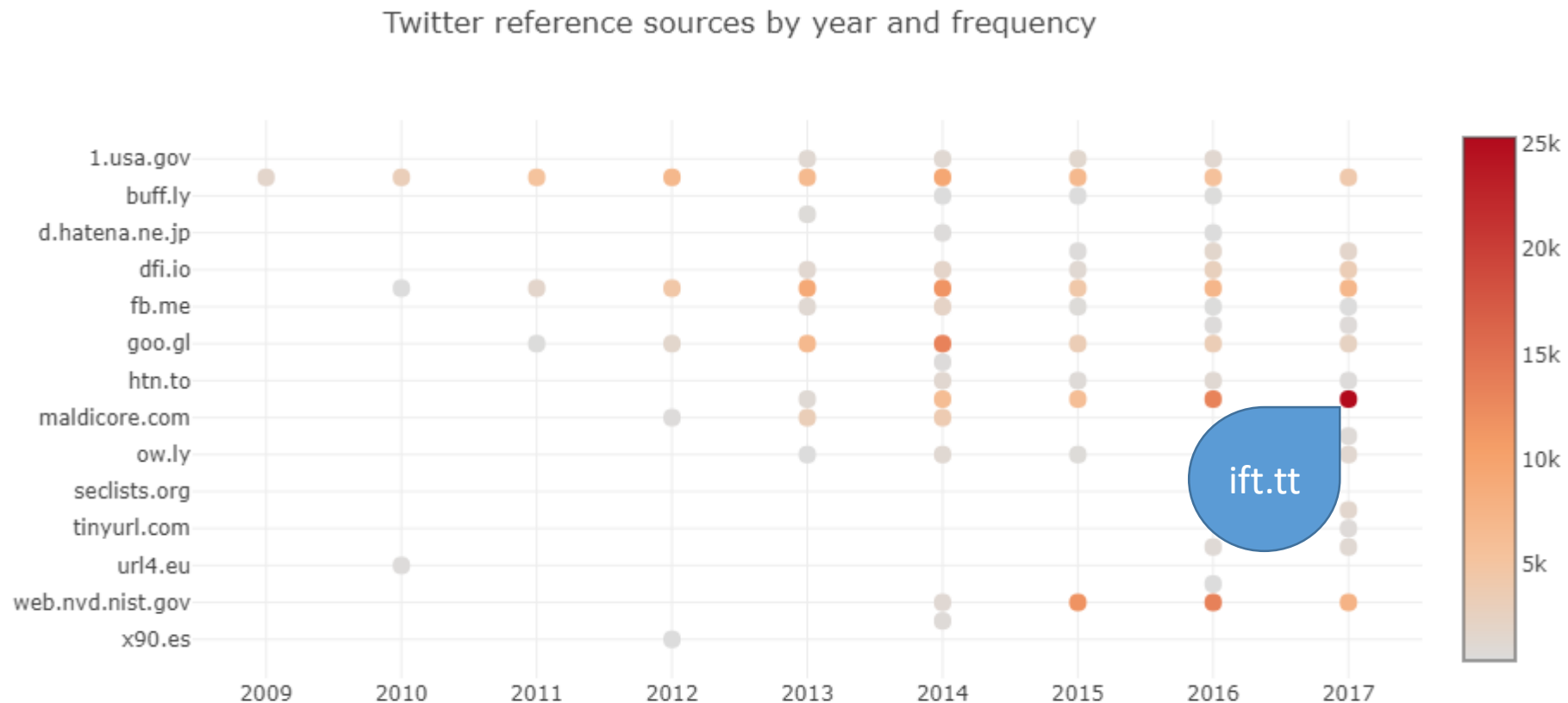
We put high hopes into the community-curated sources from Twitter...



Data analysis & design decisions

... unfortunately, link shorteners make up the majority of source domains

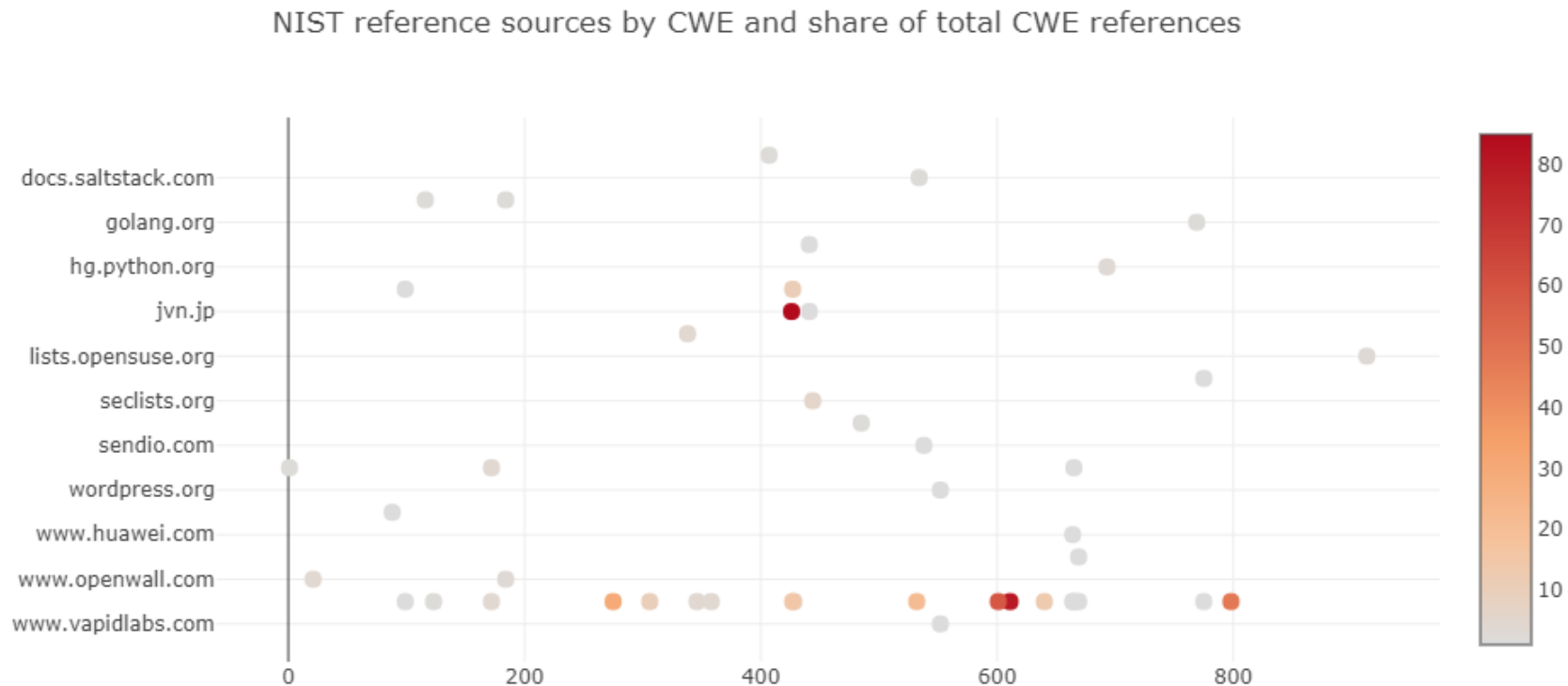
→ No trends or CWE experts deductible from tweet contents



Data analysis & design decisions

Official NIST references are suited for identifying knowledgeable sources for specific weakness types

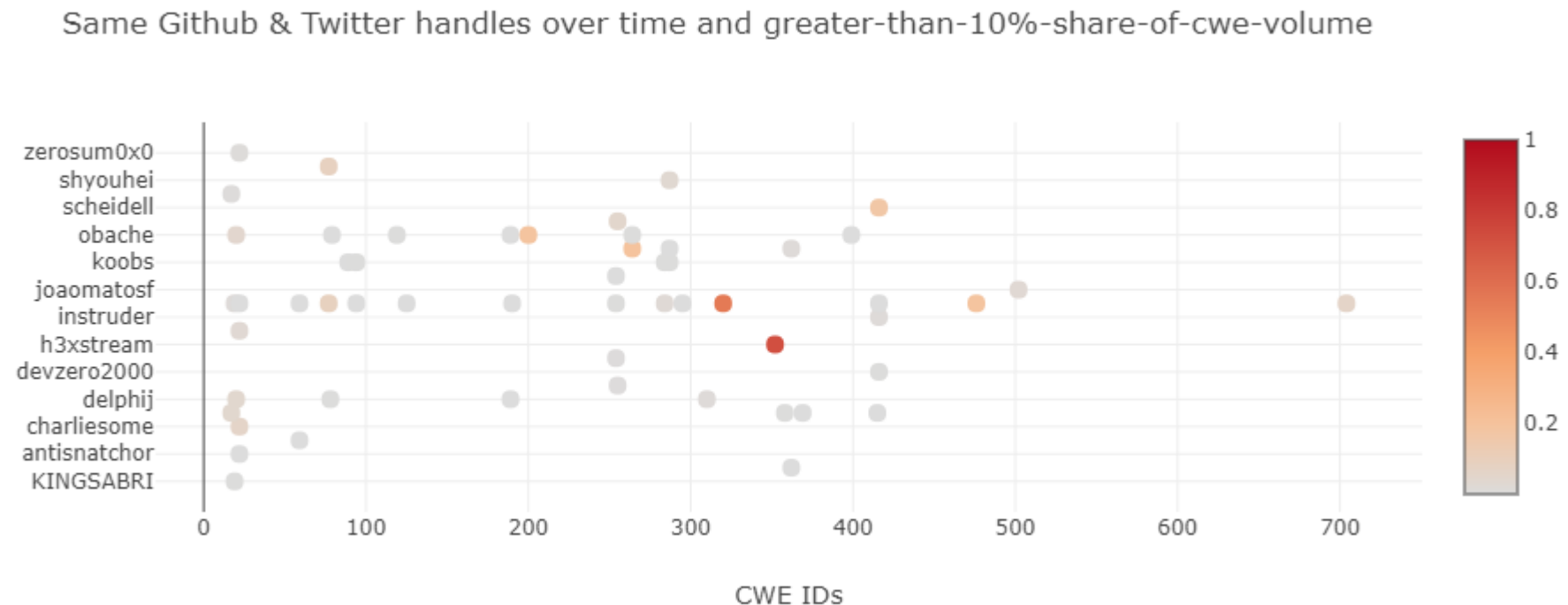
Which sources (domains) are most often referenced for any given CWE?



Data analysis & design decisions

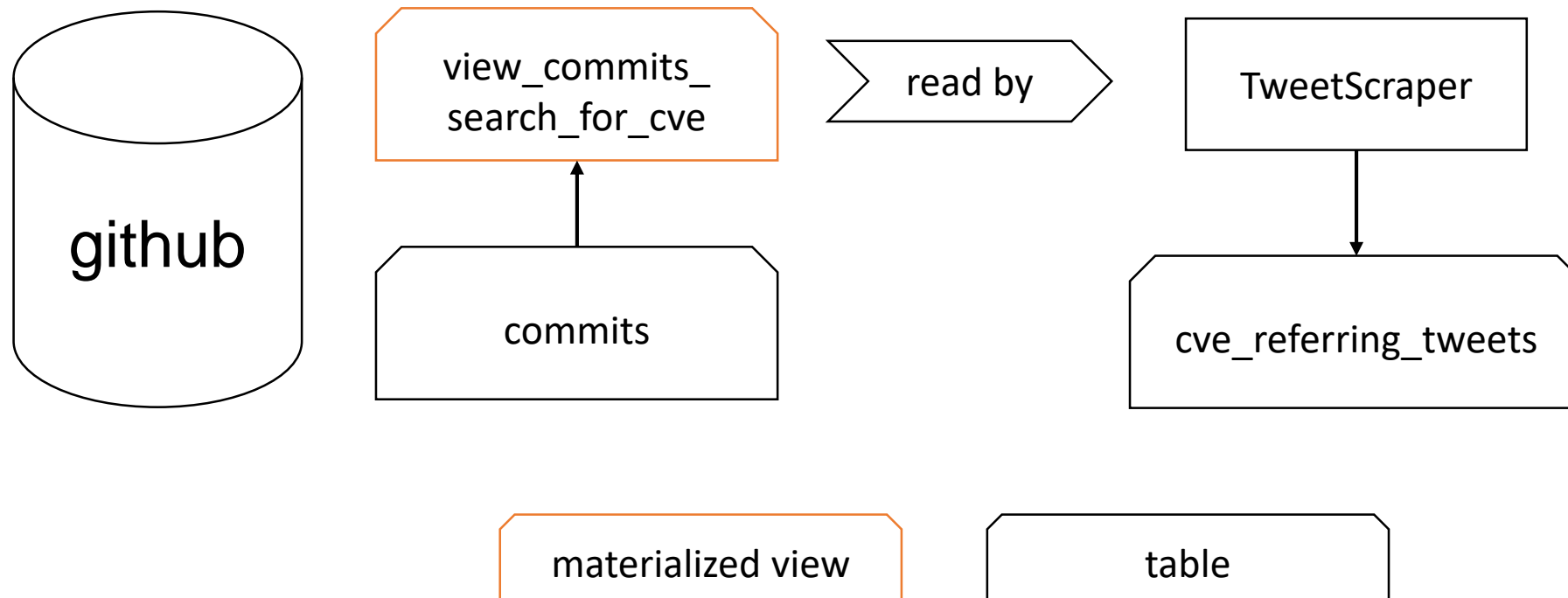
Recommending knowledgeable people by matching Github & Twitter handles
→ Expertise deductible by clustering tweeted CVE references via CWE

Which users made 10% or more of all tweets related to any given CWE?



Data procurement: Tweets

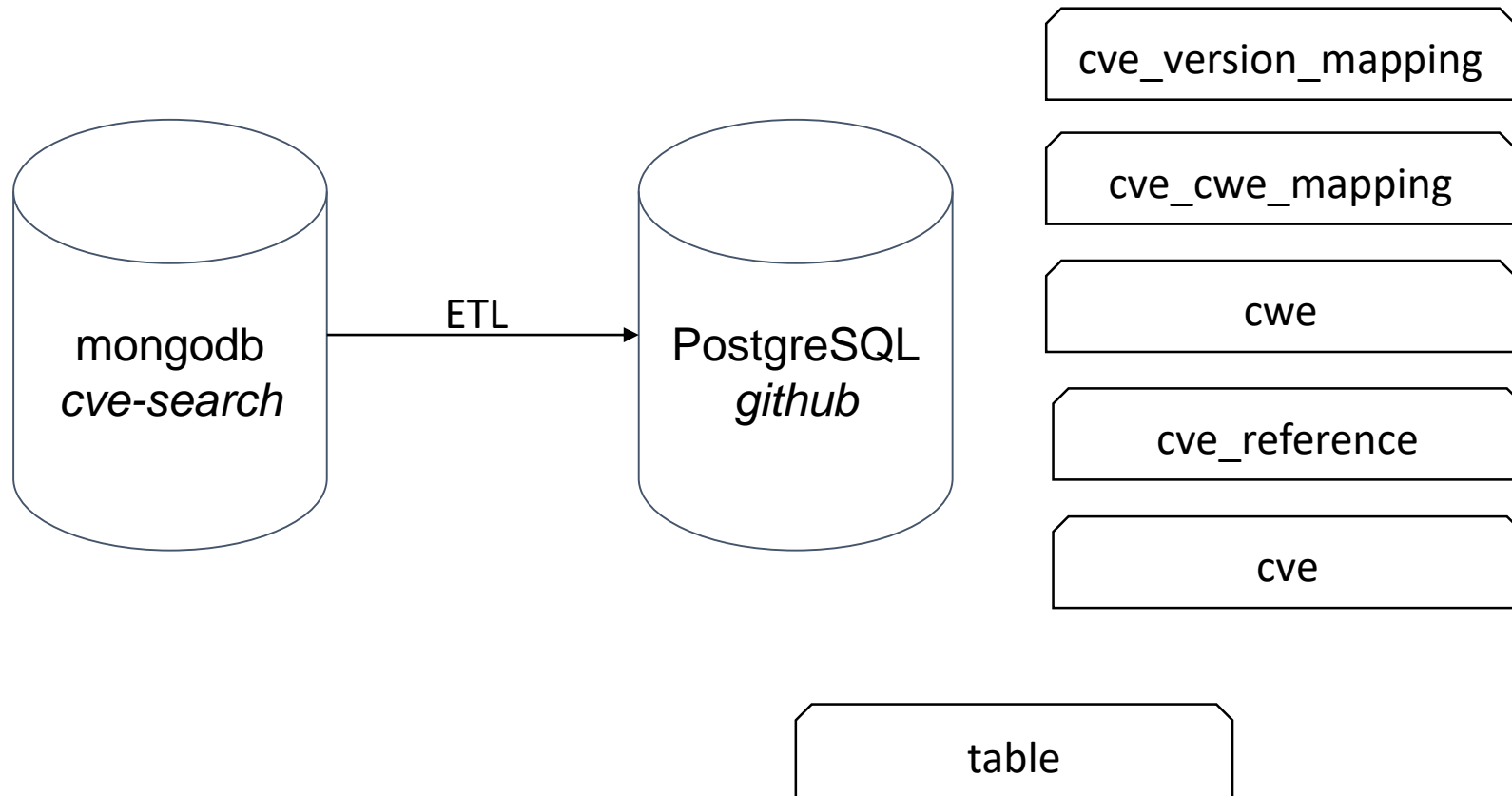
All CVE IDs found inside commits table are used as an input for TweetScraper to crawl tweets



Data procurement: NIST data

Getting hold of a NIST CVE and CWE database dump was straightforward:

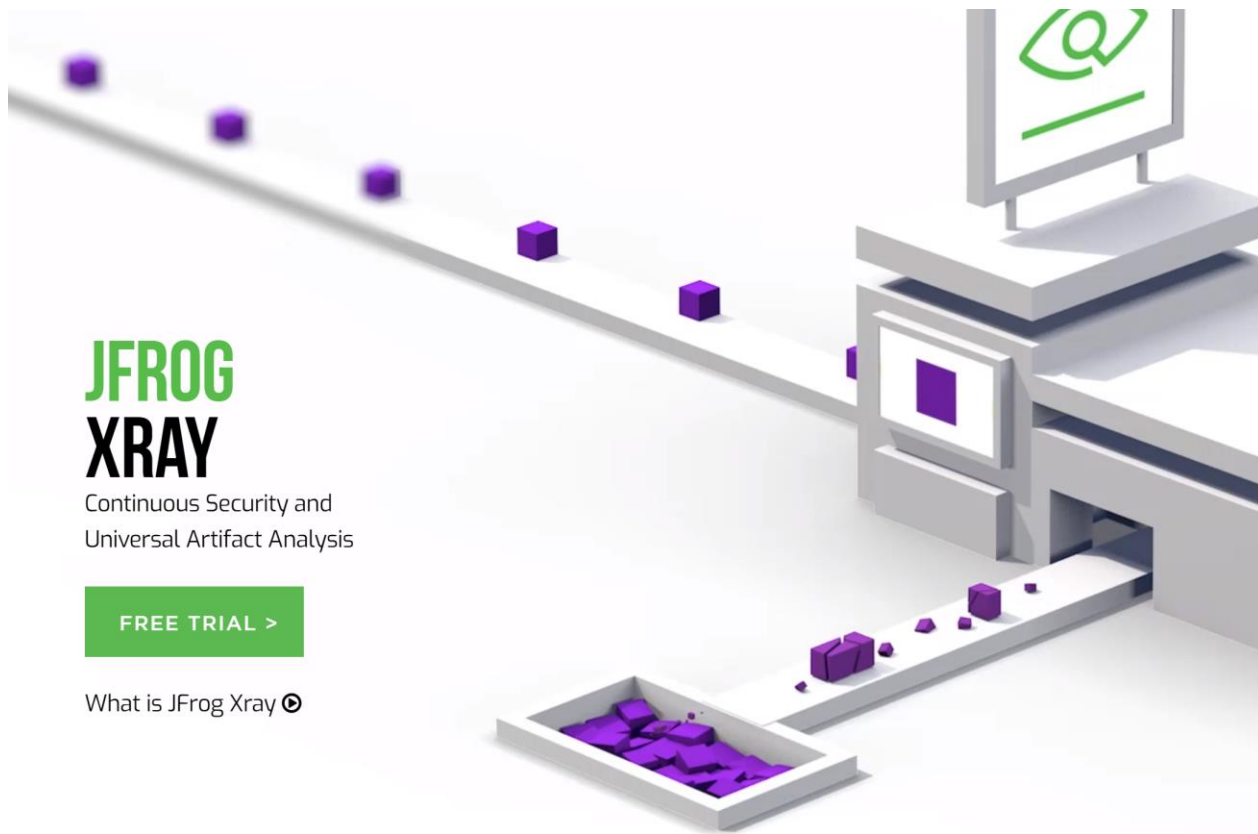
Write an ETL script for the cve-search application database: github.com/cve-search/cve-search



Future Work

- Optimization
 - speedup, indices
 - View optimization to reduce number of joins
 - Version matching inside PostgreSQL
- Timeliness
 - Efficient update mechanism for Tweets
 - Remove cve-search dependency
- Data quality
 - Resolve shortened URLs from Tweets
 - Reliable product matching heuristic
- Further developments
 - Notification service to replace constant system scanning

Competition: JFRog XRay



- Connects to JFROG Artifactory
- Monitors artifacts for security problems, performance issues and code quality
- Security issue checking is nebulous, a screenshot shows a CVE
- 14-day trial, need to be in touch with sales rep...

Competition: Github

Data services

Use the data from your repository to power these enhanced features. If you'd like to enable the [dependency graph](#), vulnerability alerts, and services like it, we'll need additional permissions.

☐ **Allow GitHub to perform read-only analysis of this repository**

By checking the "Allow GitHub to perform read-only analysis of this repository" checkbox, you're agreeing to GitHub's [Terms of Service](#) and granting us permission to perform **read-only** analysis of this private repository.
[Learn more about how we use your data.](#)

☐ **Dependency graph**

Access badass-group's dependencies, sub-dependencies, versions, and related repositories on GitHub.

☐ **Vulnerability alerts**

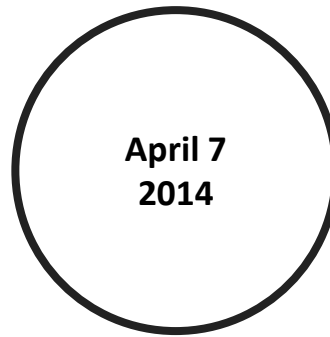
Receive alerts for known security vulnerabilities found in dependencies.

- Activate under repository settings
- Released November 2017
- Scans dependencies for known issues
- *Vulnerabilities that have CVE IDs...*
- *...Javascript and Ruby—Python support coming in 2018*

Sources:

<https://github.com/>

<https://github.com/blog/2470-introducing-security-alerts-on-github>



**Heartbleed is discovered at Google,
the bug was introduced in 2012**

~500 000 websites are open to attack

Fixed openssl library is released

**The Canada Revenue Agency can update their
systems to prevent data theft**

Data procurement: Tweets

Searching for tweets is not straightforward

Introduction

The Twitter API platform offers three tiers of search APIs:

thanks for nothing

Standard This search API searches against a sampling of recent Tweets published in the past 7 days. Part of the 'public' set of APIs.


Premium Free and paid access to the last 30 days of Tweets. Built on the reliability and full-fidelity of our enterprise data APIs, provides the opportunity to upgrade your access as your app and business grows.

Enterprise Paid (and managed) access to either the last 30 days of Tweets, or access to the entire Tweet archive. Provides full-fidelity data, direct account management support, and dedicated technical support to help with integration strategy.

Data procurement: Tweets

If the API does not come to you, you bring the API to them:

Forked and extended existing TweetScraper with PostgreSQL saving capability

 **flxw / TweetScraper**
forked from [jonbakerfish/TweetScraper](#)

Unwatch ▾

1


★ Star


0


🔗 Fork


40


<> Code

 Pull requests 0

 Projects 0


 Wiki


 Insights


 Settings


TweetScraper is a Scrapy crawler/spider for Twitter Search without using API

[Add topics](#)

 44 commits

 1 branch

 0 releases

 5 contributors

Branch: master ▾

New pull request


Create new file


Upload files


Find file

Clone or download ▾


This branch is 1 commit ahead, 1 commit behind [jonbakerfish:master](#).

 Pull request

 Compare

 **flxw** Implement Postgres adapter for crawler

Latest commit 3f4d883 on Dec 14, 2017

 **TweetScraper**

Implement Postgres adapter for crawler

a month ago

Client: Extensibility

```
import platform
```

```
distro = platform.linux_distribution()[0]
```

```
def get_package_list():
```

```
    package_list = []
```

```
    if distro == "arch":
```

```
        import pacman
```

```
        package_list = [ {
```

```
            "name":    p['id'],
```

```
            "version":p['version']
```

```
        } for p in pacman.get_installed() ]
```

```
    elif distro == 'debian':
```

Enable support for a new distribution in three steps:

- Insert new condition
- Establish connection to package manager
 - Assumption:
nobody changed the system default
- Set package_list to defined format

Data structure: Recommending sources

Allows us to
recommend a
source

cve_referring_tweets

376 590

cve

438 129

n_mapping

2 406 760

_mapping

59 548

ve

97 426

```
SELECT *, nrefs/nrefstotal::float AS refshare
FROM (
  SELECT DISTINCT cr.domain,
    cvecwe.cweid,
    COUNT(cr.domain) OVER (PARTITION BY cvecwe.cweid) AS nrefstotal,
    COUNT(cr.domain) OVER (PARTITION BY cr.domain, cvecwe.cweid) AS nrefs
  FROM view_cverefence_extracted_domains cr
  JOIN cve_cwe_classification cvecwe ON cr.cveid = cvecwe.cveid
) a
WHERE nrefs/nrefstotal::float > 0.1
```

materialized view

table

Data structure: Recommending users

