#HACKTIVITY
IT SECURITY
FESTIVAL 2019

A decade of infosec tools

from where we were to what we need now

October 2019, 26th

Thomas DEBIZE

Who am I ? Well just another infosec passionate

Thomas DEBIZE

- https://github.com/maaaaz

A bit of context

The infosec field has quite evolved during the **last decade**, especially around **tool crafting**

Some **old sweet dreams** now come true

- Scan the **entire IPv4** space in few minutes/hours/days
- Query **all OSINT information** you want
- Pwn **large** Windows corporate **infrastructures**
- **Fuzz** anything you want
- **Storing** entire earth's hashes and passwords
- …

Coding became **social**

- Infosec people **enhanced** their **coding skill**
- Infosec people now with the **will** to write **good** and **practical** tools
- Some exhibitions **devoted** to the tool crafting art (Black Hat Arsenal)

➔ More security folks are writing **more and more good quality and reliable tools**

More tools allowing **attack AND defense**

- More **recognition** for the Blue side
- Moving from the **"breaking" era**, to the **"securing and building" era**
- Finally taking advantage of **data visualization** (graphs etc.)

A bit of context

But in the same time we still rely on some old school core tools

Author : Fyodor

```
---[  Phrack Magazine   Volume 7, Issue 51 September 01, 1997, article 11 of 17


-------------------------[  The Art of Port Scanning


--------[  Fyodor <fyodor@dhp.com>


[ Abstract ]

This paper details many of the techniques used to determine what ports (or
similar protocol abstraction) of a host are listening for connections.  These
ports represent potential communication channels.  Mapping their existence
facilitates the exchange of information with the host, and thus it is quite
useful for anyone wishing to explore their networked environment, including
hackers.  Despite what you have heard from the media, the Internet is NOT
all about TCP port 80.  Anyone who relies exclusively on the WWW for
information gathering is likely to gain the same level of proficiency as your
average AOLer, who does the same.  This paper is also meant to serve as an
introduction to and ancillary documentation for a coding project I have been
working on.  It is a full featured, robust port scanner which (I hope) solves
some of the problems I have encountered when dealing with other scanners and
when working to scan massive networks.  The tool, nmap, supports the following:

    - vanilla TCP connect() scanning,
    - TCP SYN (half open) scanning,
    - TCP FIN (stealth) scanning,
    - TCP ftp proxy (bounce attack) scanning
    - SYN/FIN scanning using IP fragments (bypasses packet filters),
    - UDP recvfrom() scanning,
    - UDP raw ICMP port unreachable scanning,
    - ICMP scanning (ping-sweep), and
    - reverse-ident scanning.

The freely distributable source code is appended to this paper.
```

A bit of context

A bit of context

2003 Top 75 Tools Results

From: Fyodor <fyodor () insecure org>
Date: Sun, 4 May 2003 00:33:30 -0700

Hello everyone,

Thanks for the fantastic response to the Nmap user survey! It is now closed, but recorded 1854 responses -- that blew away our goal of 1500 and is over 50% greater than the 2000 survey! I haven't analyzed all the questions/comments yet, but I did go through your recommended tools and create a most-loved list as I did in 2000. Thanks to the increased responses, I was able to expand the list from "Top 50" to

It is worth noting that almost half of the 2003 top 50 are new to the list. Congratulations to these rising stars:

GFI LANguard: A commercial network security scanner for Windows
Ettercap: In case you still thought switched LANs provide much extra security
Nikto: A more comprehensive web scanner
Kismet: A powerful wireless sniffer
SuperScan: Foundstone's Windows TCP port scanner
Fport: Foundstone's enhanced netstat
Network Stumbler: Free Windows 802.11 Sniffer
N-Stealth: Web server scanner
AirSnort: 802.11 WEP Encryption Cracking Tool
NBTScan: Gathers NetBIOS info from Windows networks
Cain & Abel: The poor man's L0phtcrack
XProbe2: Active OS fingerprinting tool
SolarWinds Toolsets: A plethora of network discovery/monitoring/attack tools
THC-Amap: An application fingerprinting scanner
OpenSSL: The premier SSL/TLS encryption library
Honeyd: Your own personal honeynet
Achilles: A Windows web attack proxy
Brutus: A network brute-force authentication cracker
Stunnel: A general-purpose SSL cryptographic wrapper
Paketto Keiretsu: Extreme TCP/IP
SPIKE Proxy: HTTP Hacking
THC-Hydra: Parallized network authentication cracker

The questions giving birth to this study

In this **myriad** of newly-created tools during that decade

- How are these new tools **built** ?
- Where are they **hosted** ?
- **How long** are they maintained ?
- Are they **really better designed** than the old ones ?

All in all, how did it **evolve** ?

01 A bit of context

02 Tools during the last decade

03 Make your tool great (again)

04 Wrapping it up

Study scope and limitations

What did I do ?

- Analyze metadata from **4 major publication sources** of infosec tools

Packetstorm	Toolswatch	Kitploit	n0where
https://www.packetstormsecurity.com/files/tags/tool	http://www.toolswatch.org	https://www.kitploit.com	https://n0where.net

Manual crawl — **vulners.com feed**

How ?

- With **Dataiku Data Science Studio** free edition, an amazing all-in-one tool. Mostly Excel on steroids.
- You should try it https://www.dataiku.com/learn/portals/tutorials.html

How many records have been analyzed ? On which period ?

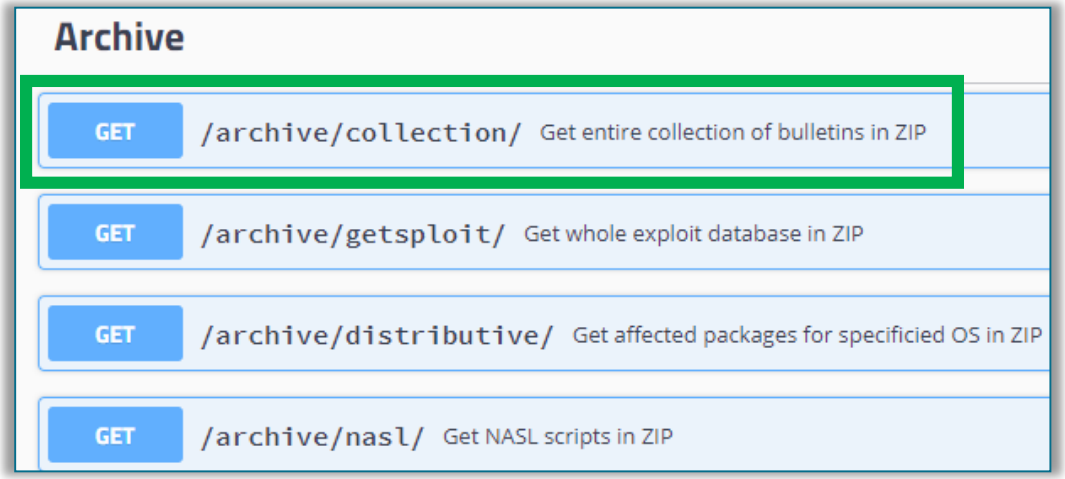
Packetstorm	Toolswatch	Kitploit	n0where
6872 records	1642 records	3331 records	1052 records
From 01/1994 to 10/2019	From 12/2010 to 06/2019	From 12/2010 to 10/2019	From 06/2010 to 03/2019

Slight off-topic greetings for vulners.com

vulners.com indexes a lot of cool sources and provides a **free API** to get **structured data**

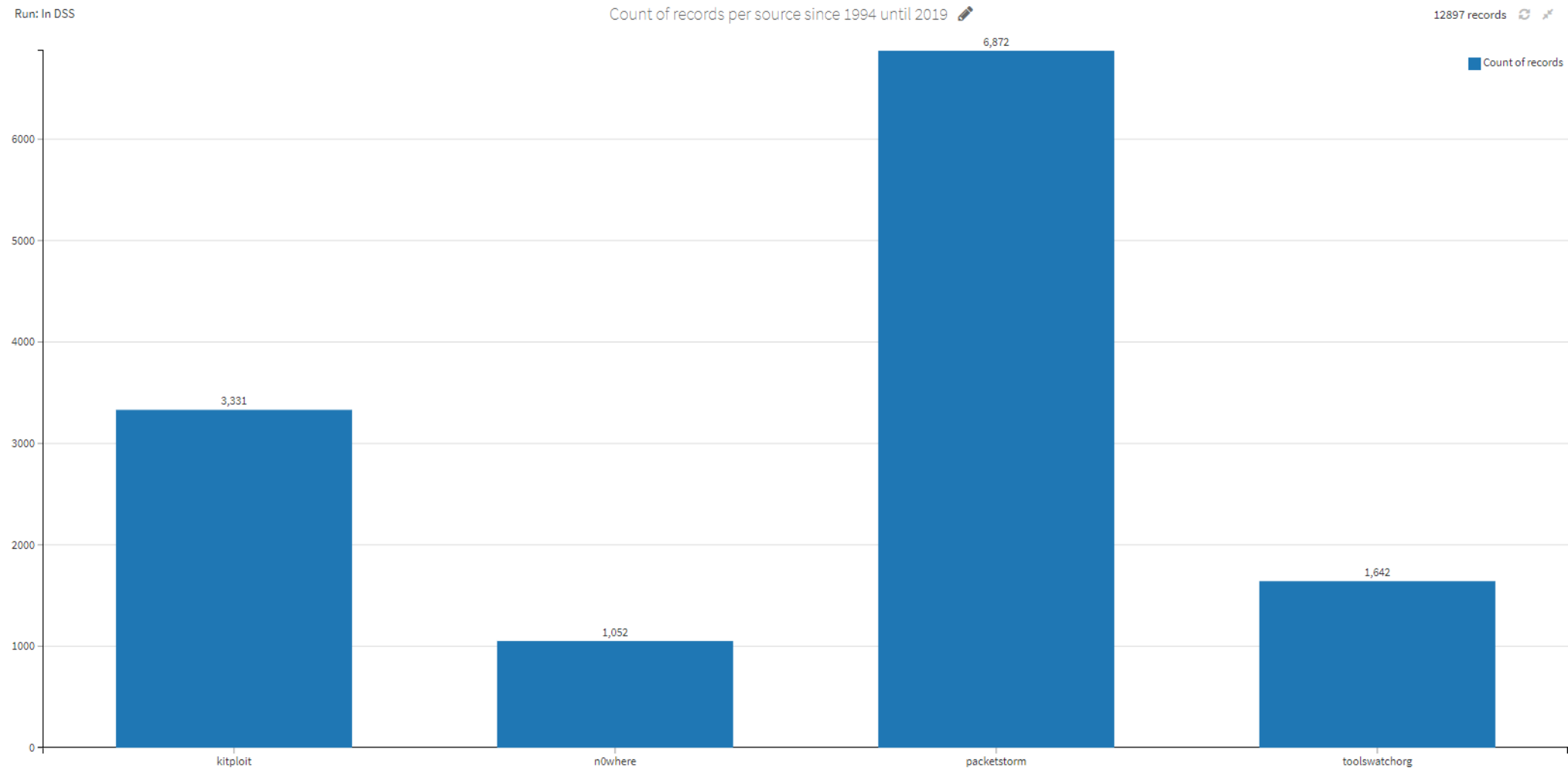
- Blogs
- Vulnerability feeds
- IOC feeds
- Exploit feeds
- Vendors
- General news websites
- …

Thanks to them !



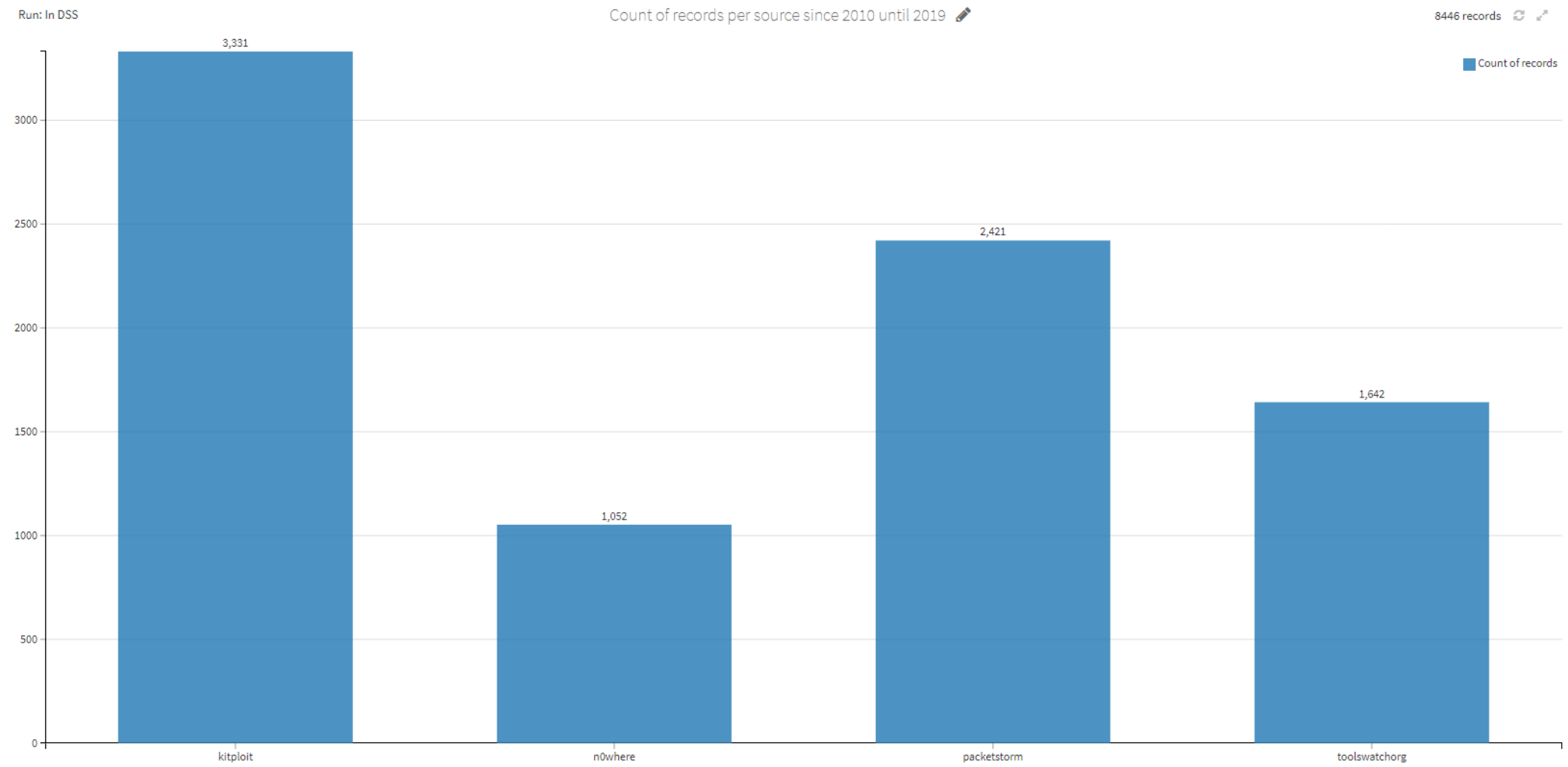
First, the source dataset,
also known as “the base of all biases”

Distribution of records per source **since 1994**



Irrelevant dataset as solely **Packetstorm** was existing **before 2010** ☺

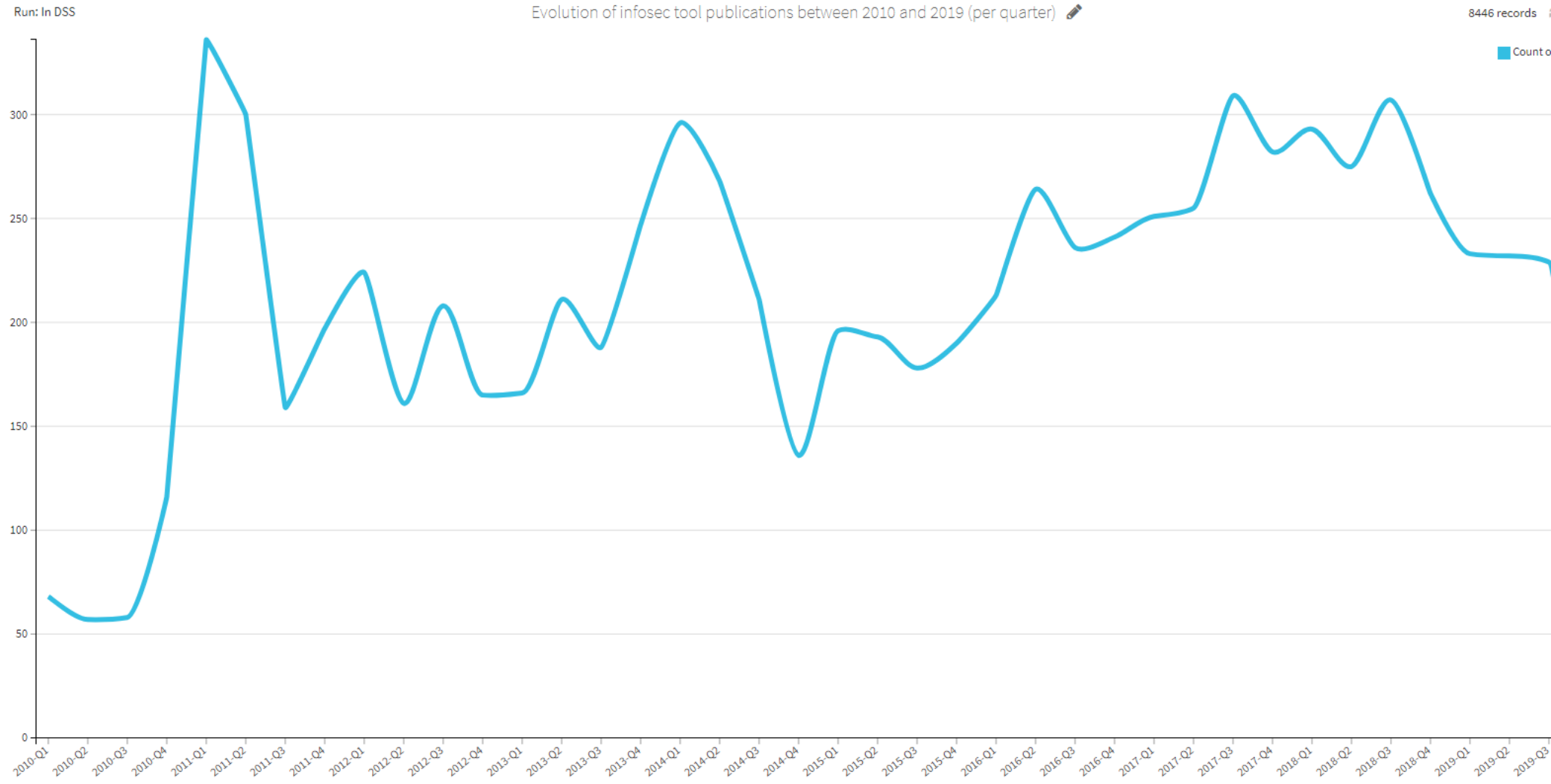
Distribution of records per source **since 2010**



Relevant dataset as all sources **have quite the same order of magnitude for publications since 2010**

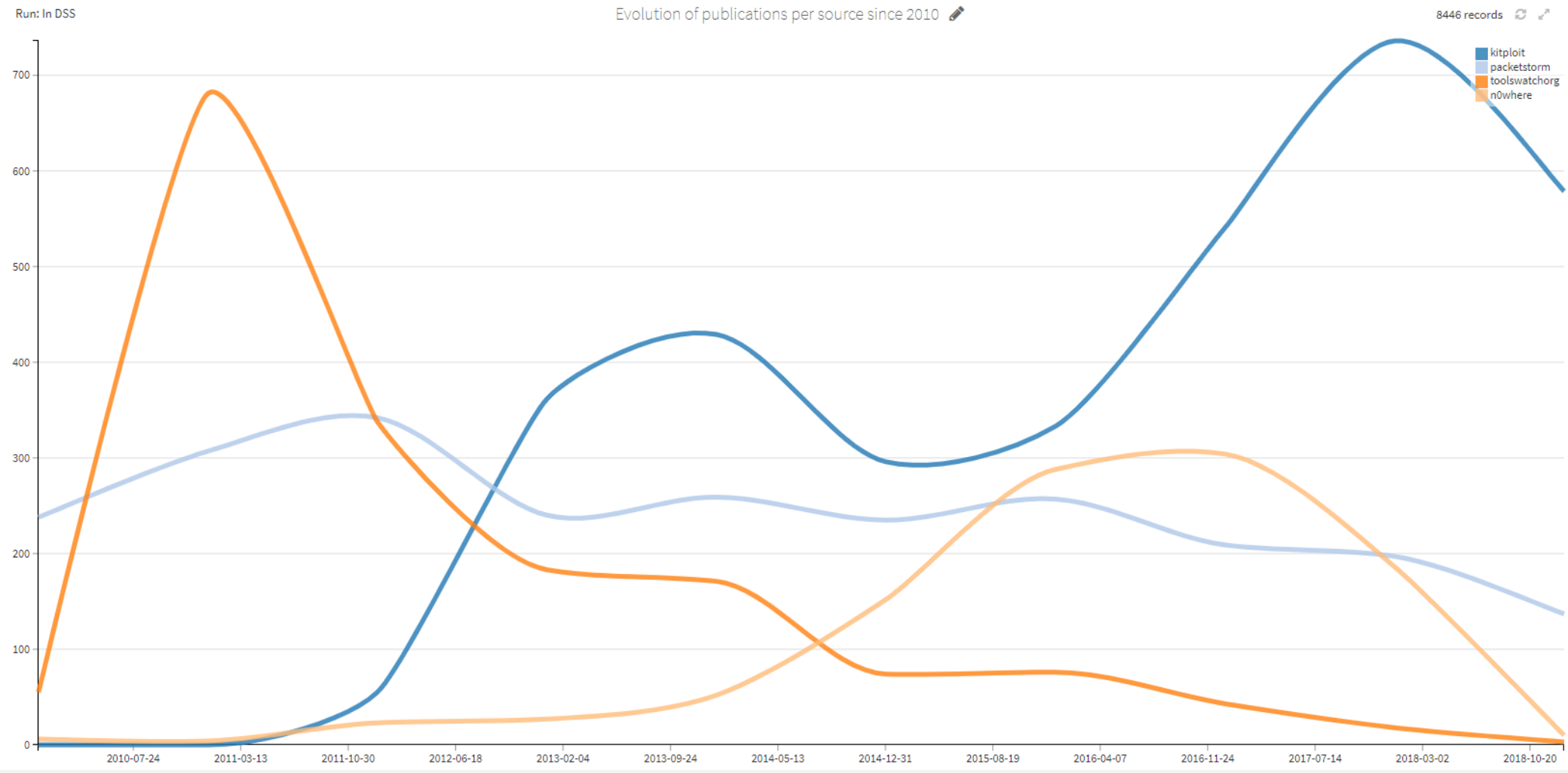
Then, some evolution figures

Evolution of the **number of publications** (per quarter) since 2010



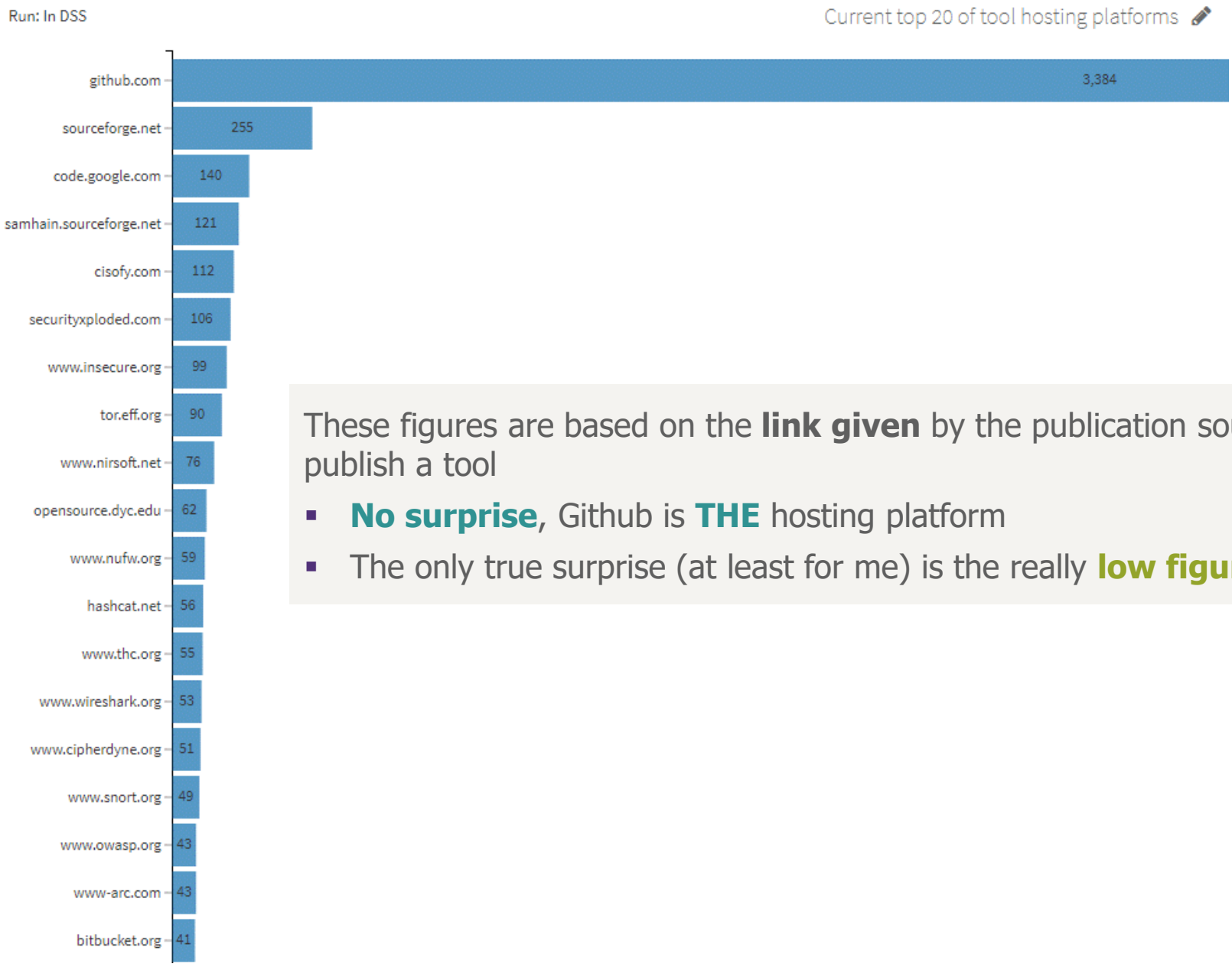
- **More and more** publications

Evolution of the **number of publications** per source since 2010



- **Less and less** publications from **toolswatch** on their website
- **Packetstorm**, the old school reference, used to **maintain** its rhythm of publications, but now tends to **diminish** it, while **Kitploit the "decade newcomer"** tends to become the **new reference source**

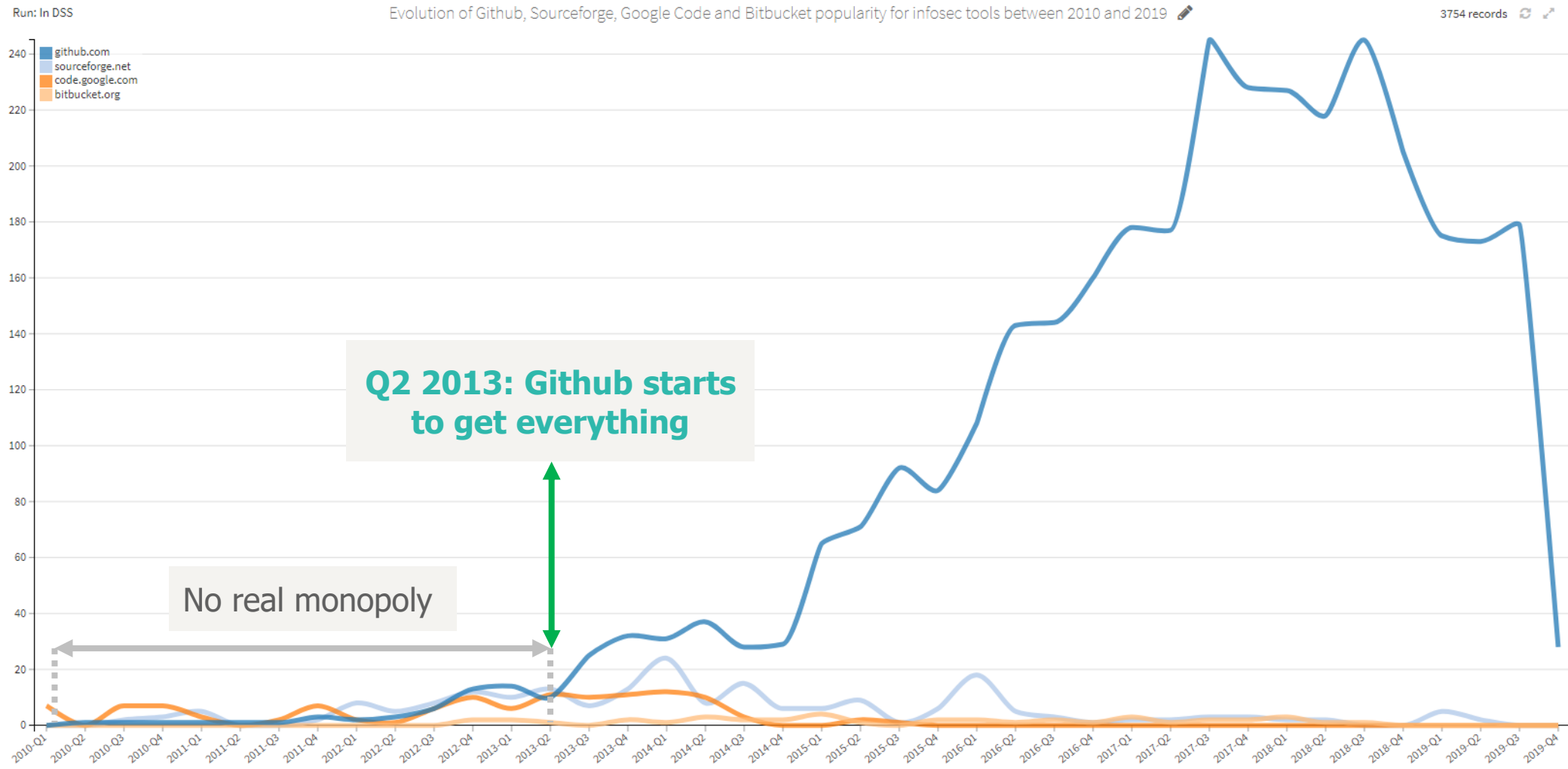
Distribution of tool **hosting platforms** (to date)



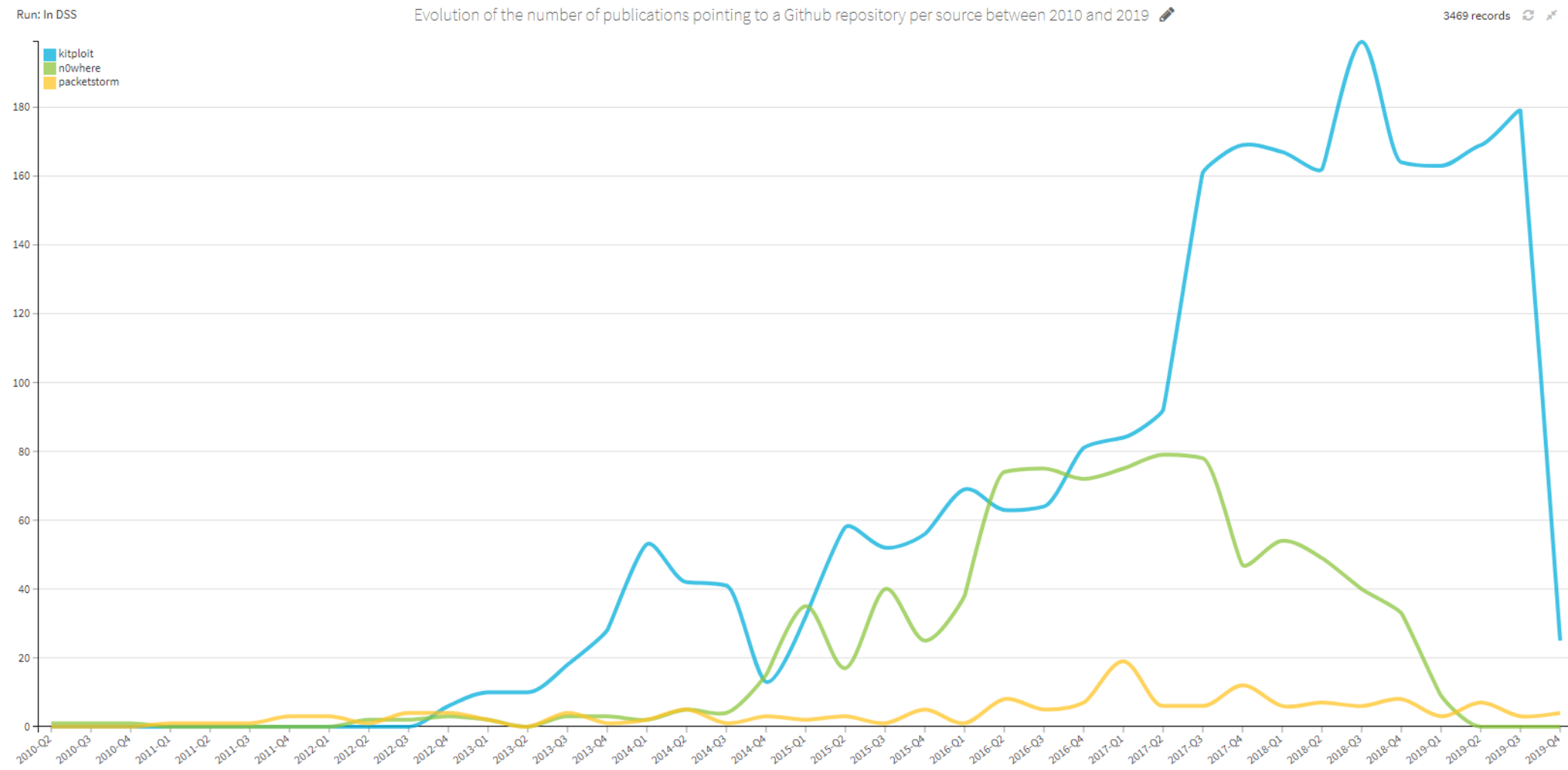
These figures are based on the **link given** by the publication sources when they publish a tool

- **No surprise**, Github is **THE** hosting platform
- The only true surprise (at least for me) is the really **low figure for Bitbucket**

Evolution of **Github, Sourceforge, Google Code and Bitbucket popularity** for infosec tools between 2010 and 2019



Evolution of the **number of publications pointing to a Github repository** per source between 2010 and 2019



- **Packetstorm** is not following the trend, hence continuing to bring diversity for tool sources

Ok, whatever, so if everything seems to be hosted on Github,
let’s focus on Github !

Some statistics about for the **2300+ Github** repositories analyzed

Stars

Average: 1053
Median: 292
Std Dev: 2968

Forks

Average: 188
Median: 70
Std Dev: 451

Watchers

Average: 1053
Median: 292
Std Dev: 2968
(1 star induces 1 watch)

Releases

Average: 5
Median: 0
Std Dev: 18

Size

Average: 15 MB
Median: 993 KB
Std Dev: 61 MB

Commits

Average: 522
Median: 71
Std Dev: 1941

Maintenance duration

in days, last commit – first commit on master

Average: 910 (2,5 years)
Median: 626 (1,7 years)
Std Dev: 976 (2,6 years)

All Issues

Average: 224
Median: 16
Std Dev: 2030

Open issues

Average: 25
Median: 3
Std Dev: 96

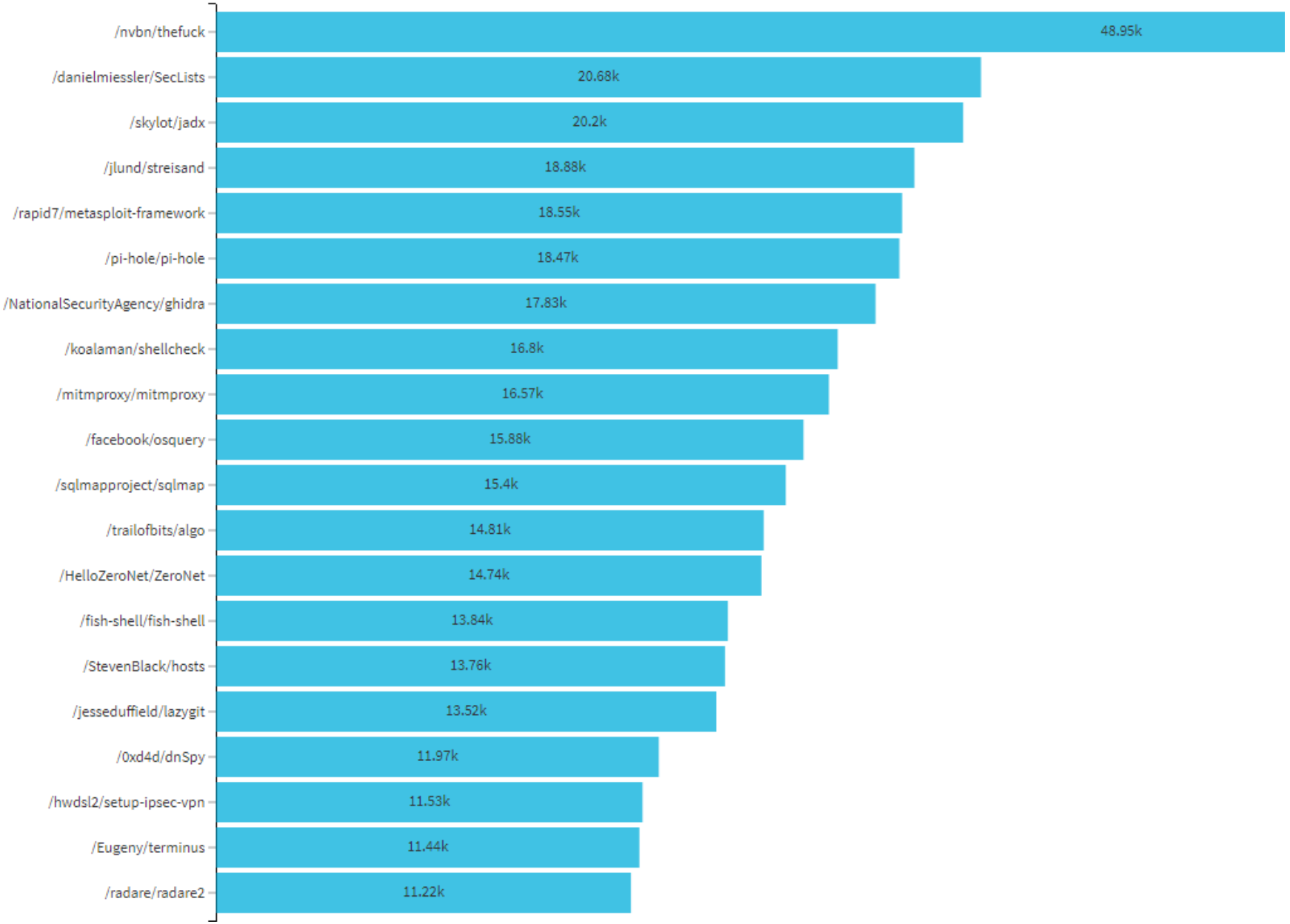
All Pull Requests

Average: 69
Median: 4
Std Dev: 398

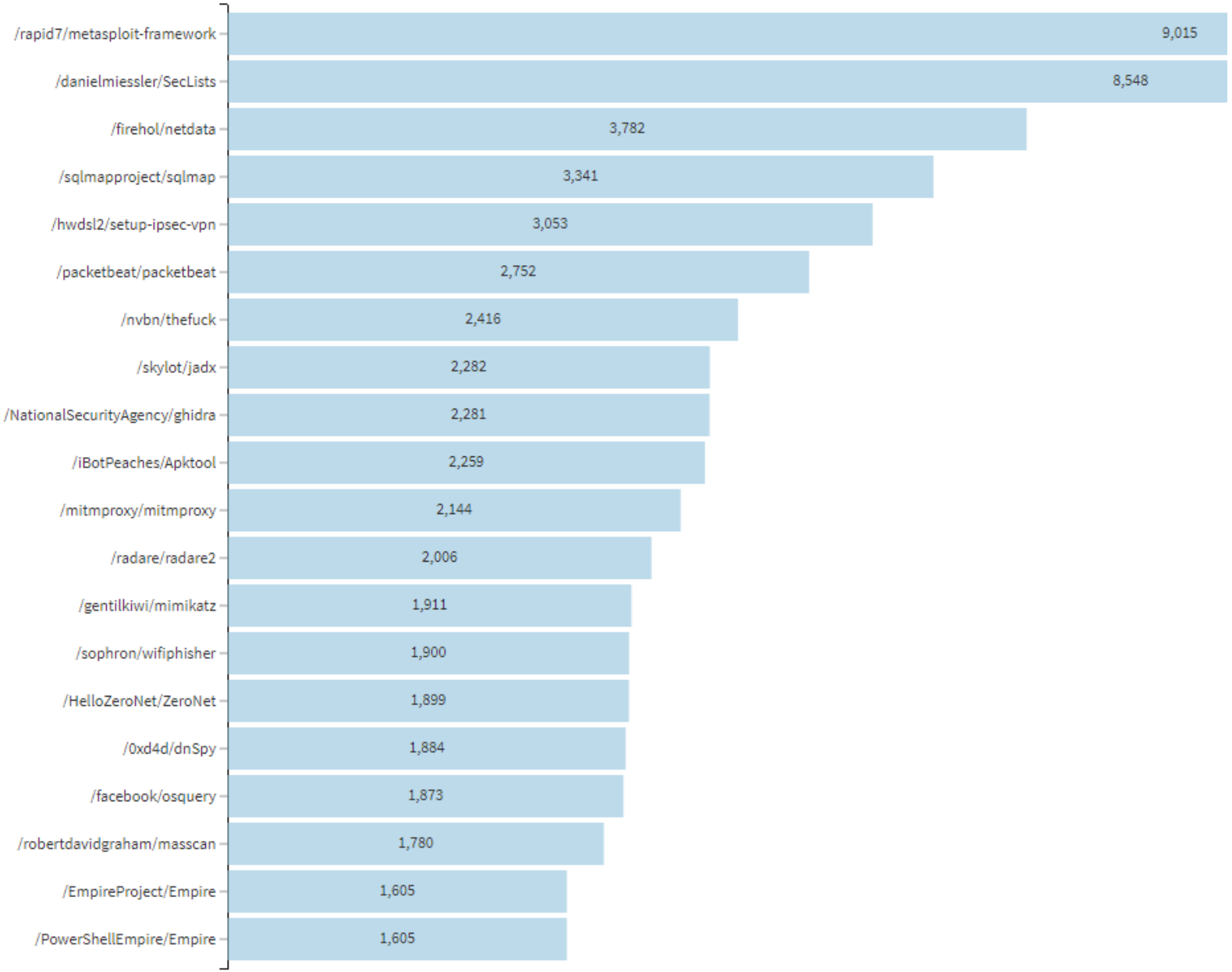
Open Pull Requests

Average: 2
Median: 0
Std Dev: 9

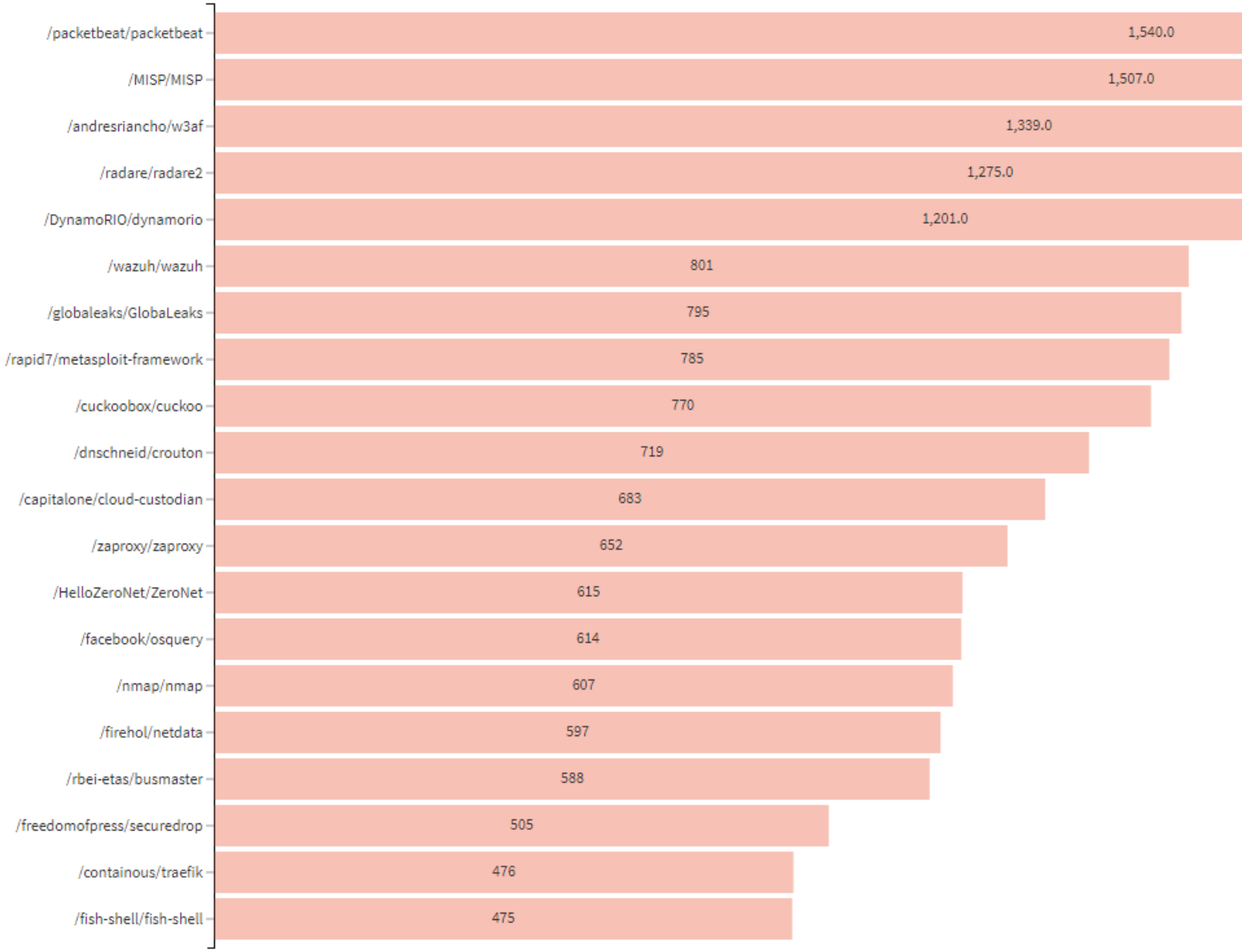
Top 20 of the **most starred** infosec tools on Github



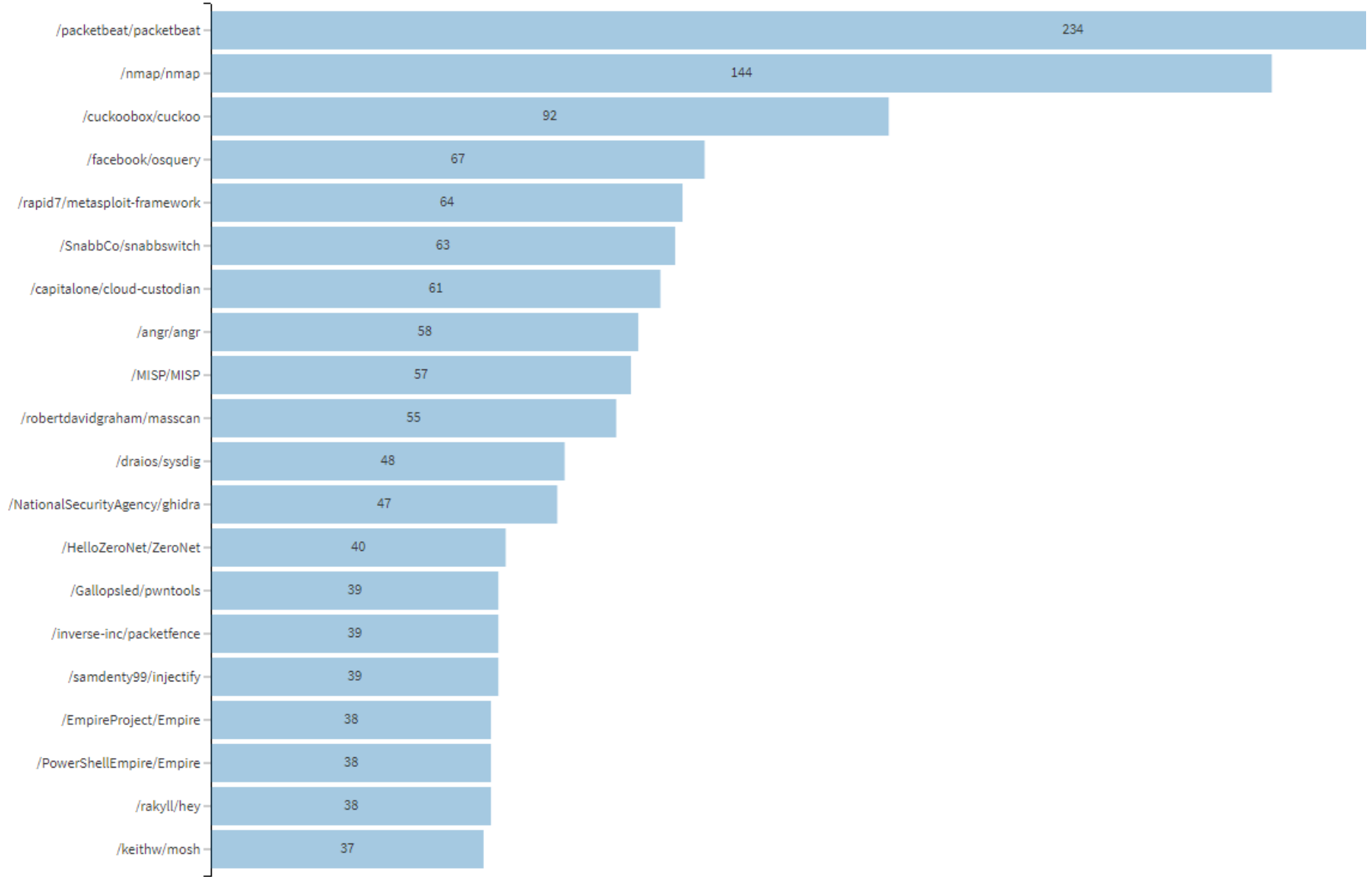
Top 20 of the **most forked** infosec tools on Github



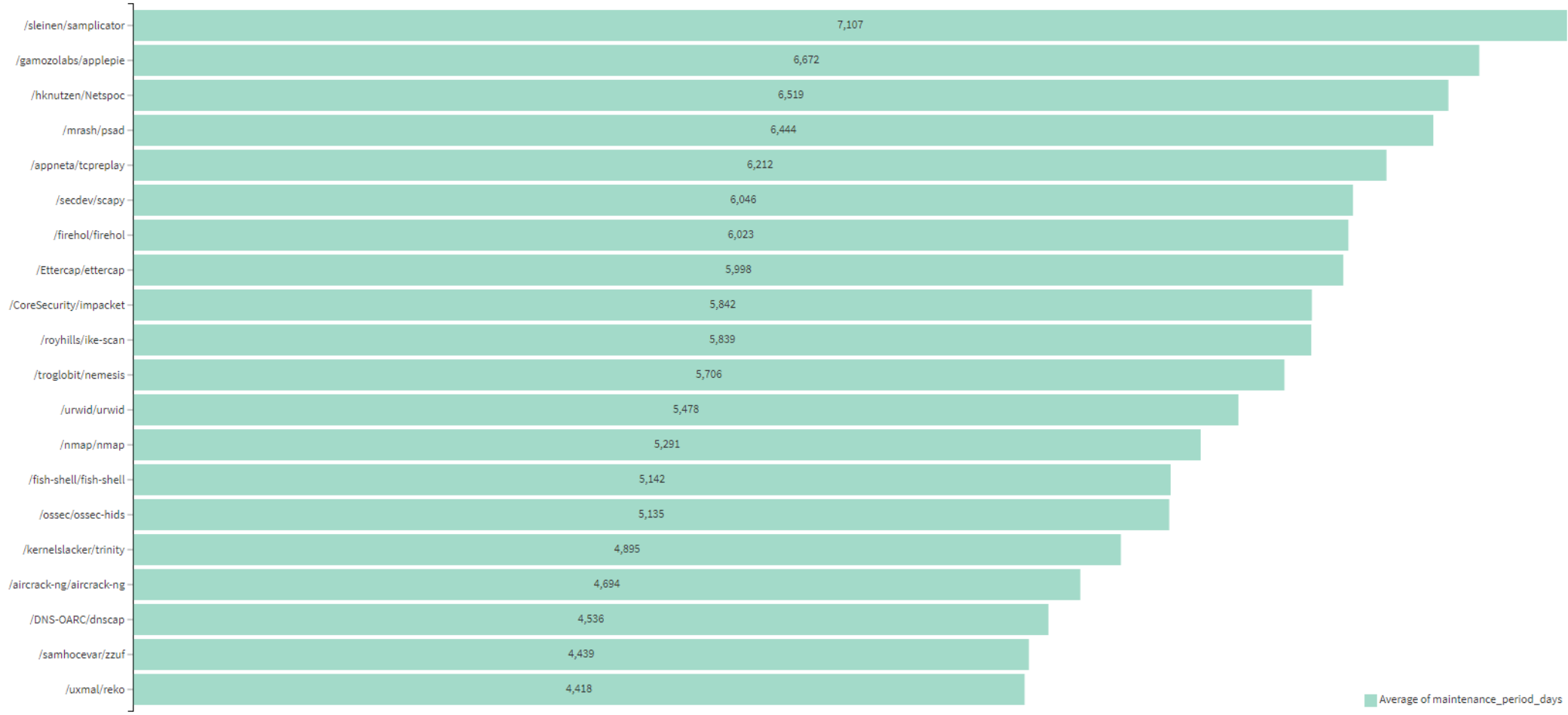
Top 20 of infosec tools on Github with the **biggest number of open issues**



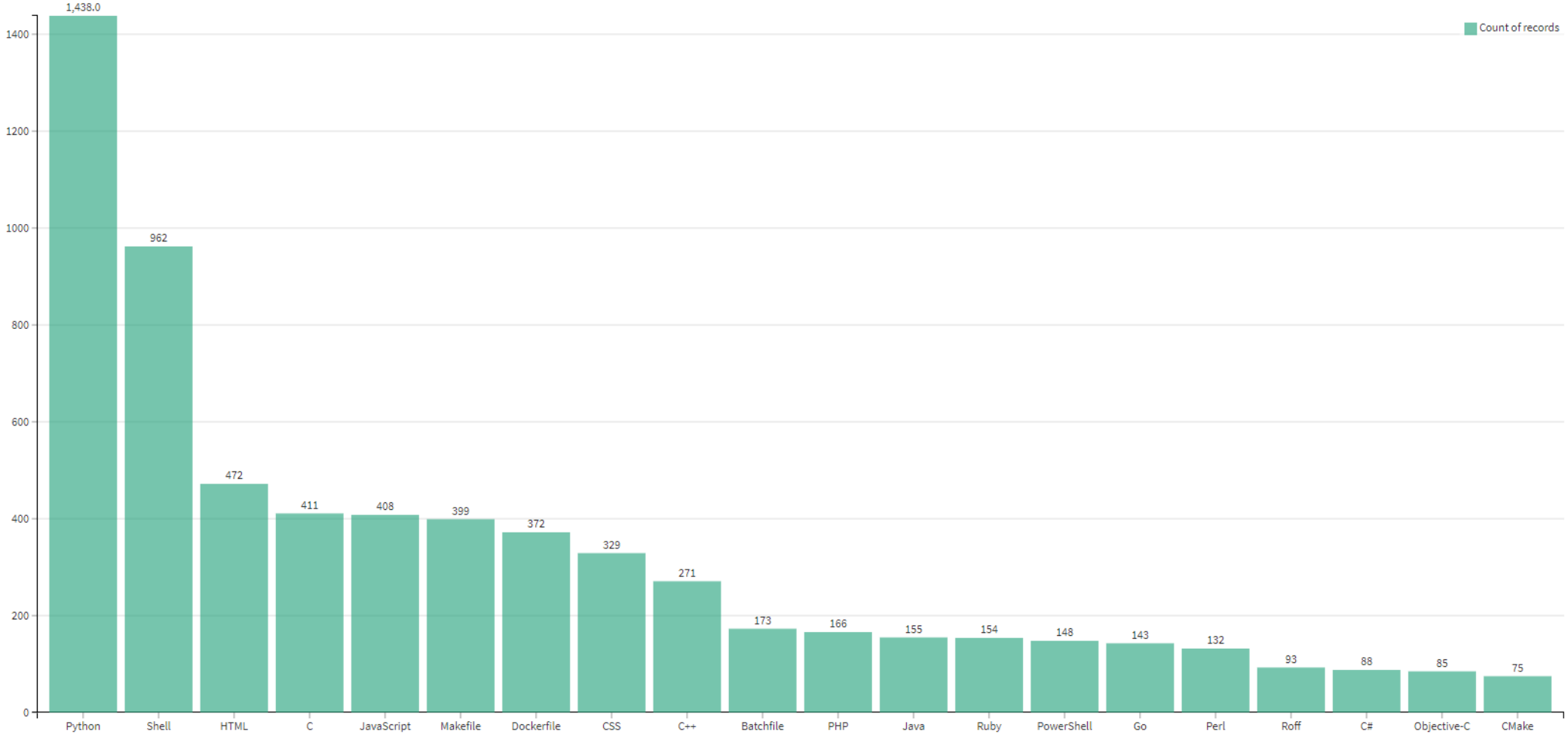
Top 20 of infosec tools on Github with the **biggest number of open pull requests**



Top 20 of infosec tools on Github with the **longest maintenance period**

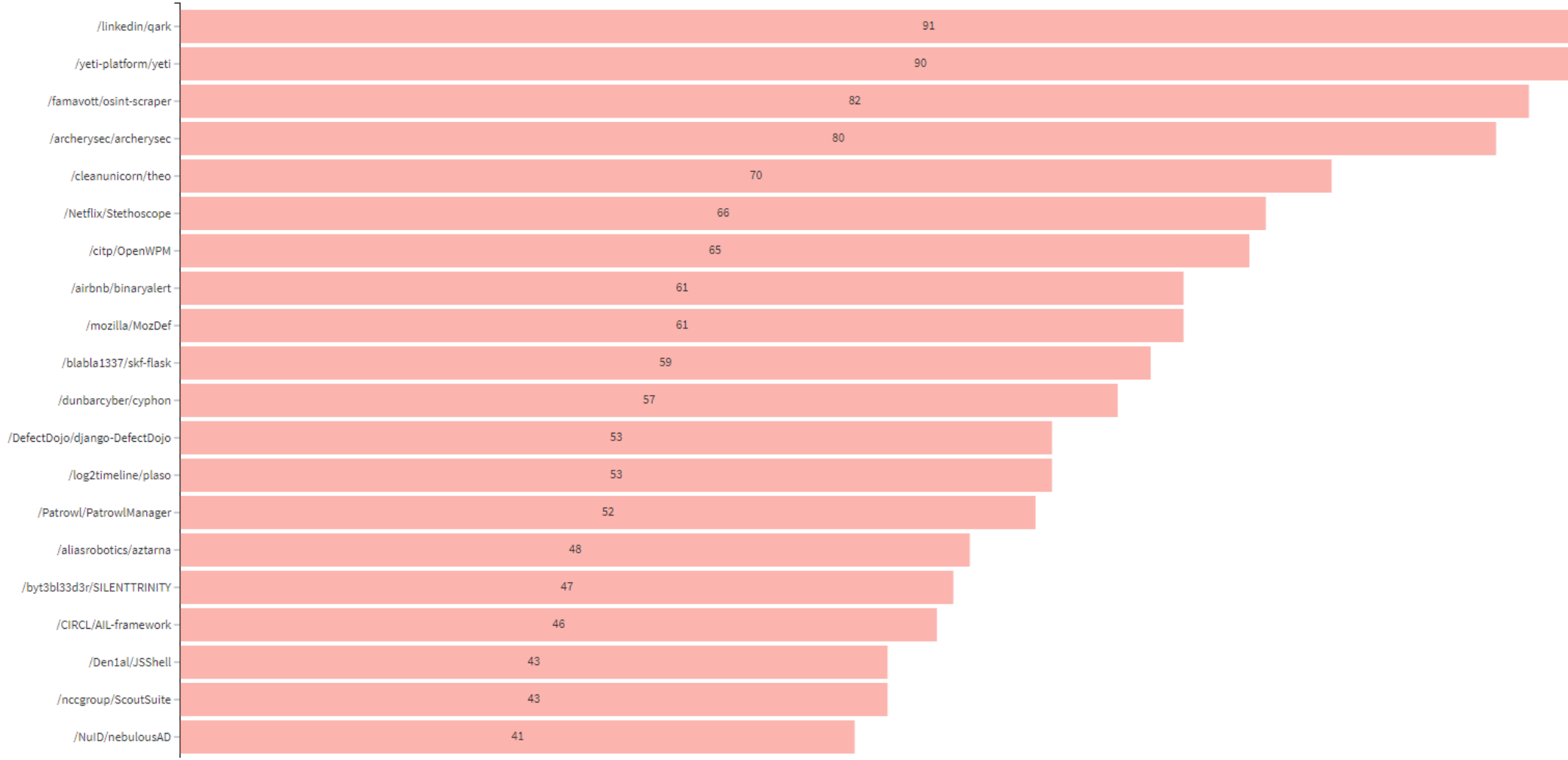


Top 20 of the **most frequent programming languages** in infosec tools

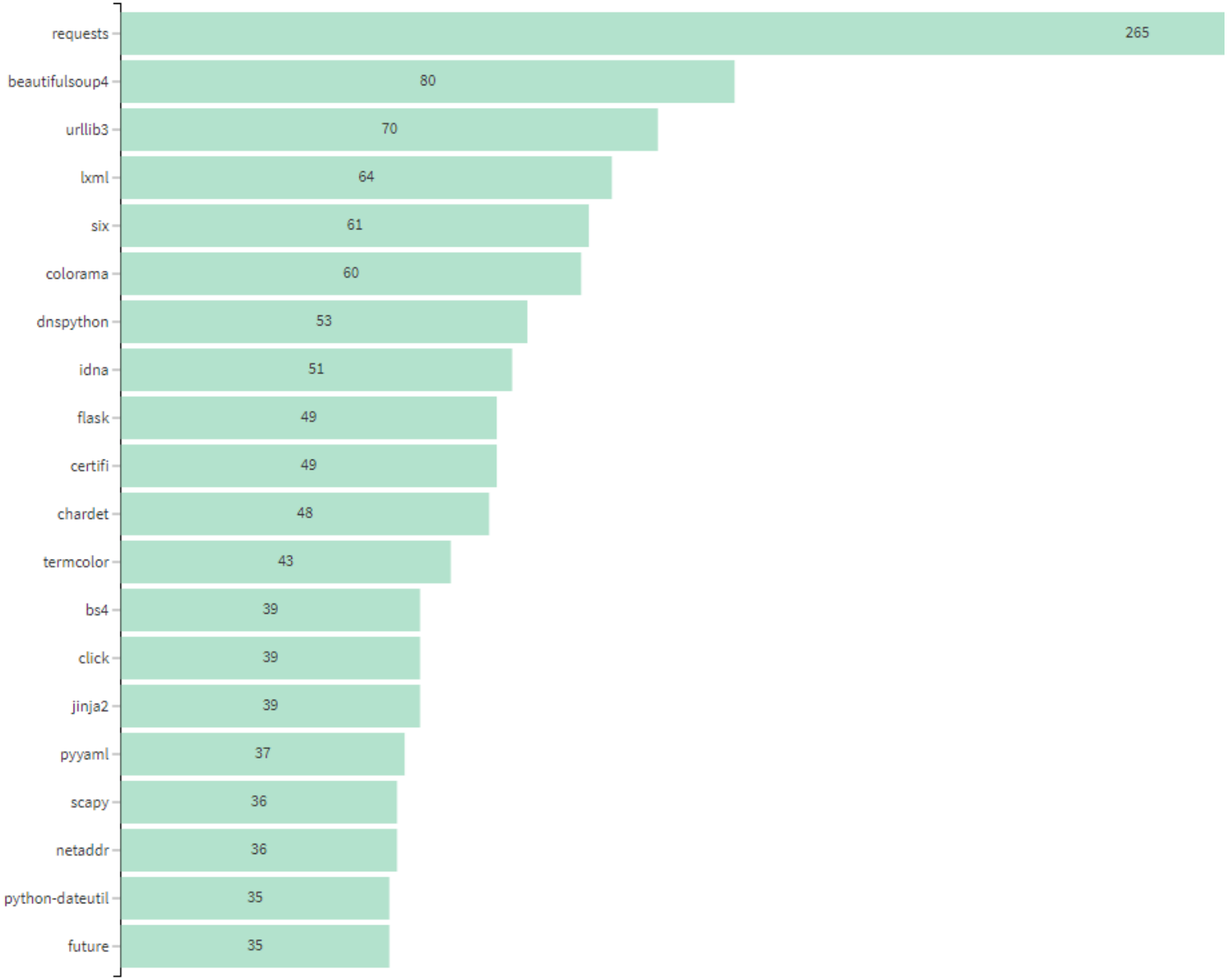


Ok, whatever, so if infosec tools are mostly developed in Python, let’s focus on Python !

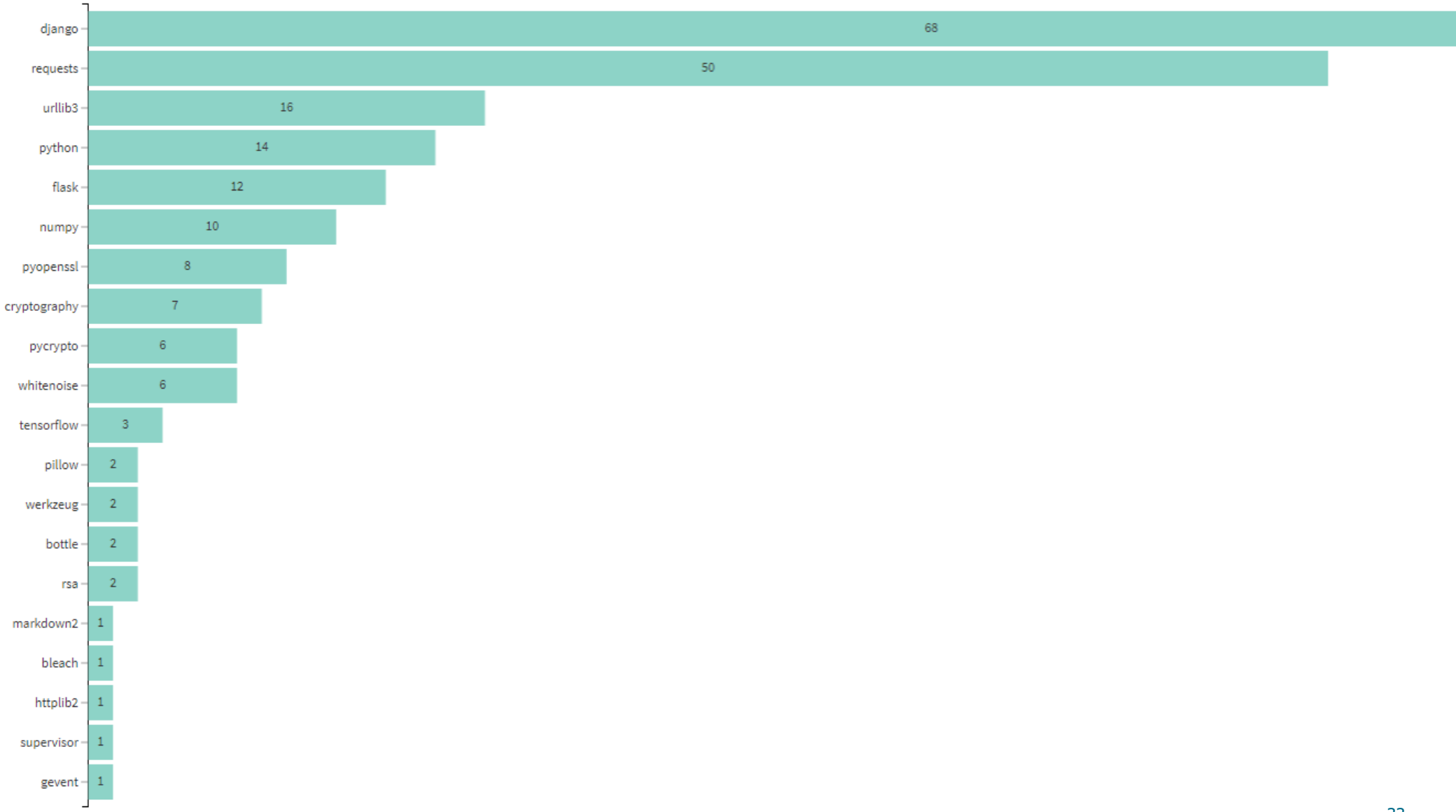
Top 20 of infosec tools in Python having the **biggest number of dependencies**



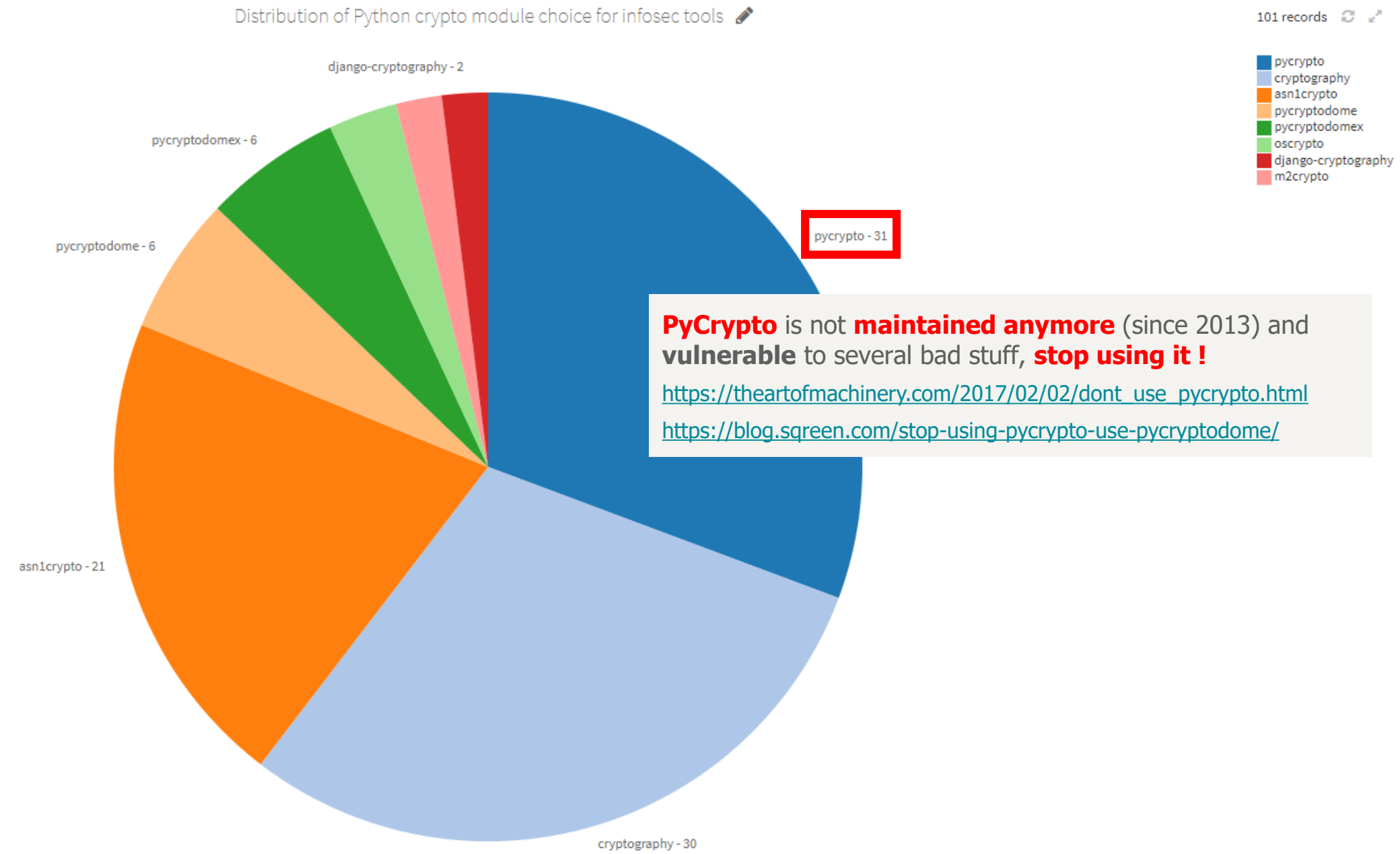
Top 20 of the **most used Python 3rd-party modules** in infosec tools



Top 20 of the **most vulnerable 3rd-party modules** in infosec Python tools



Distribution of **Python crypto module choice** for infosec tools

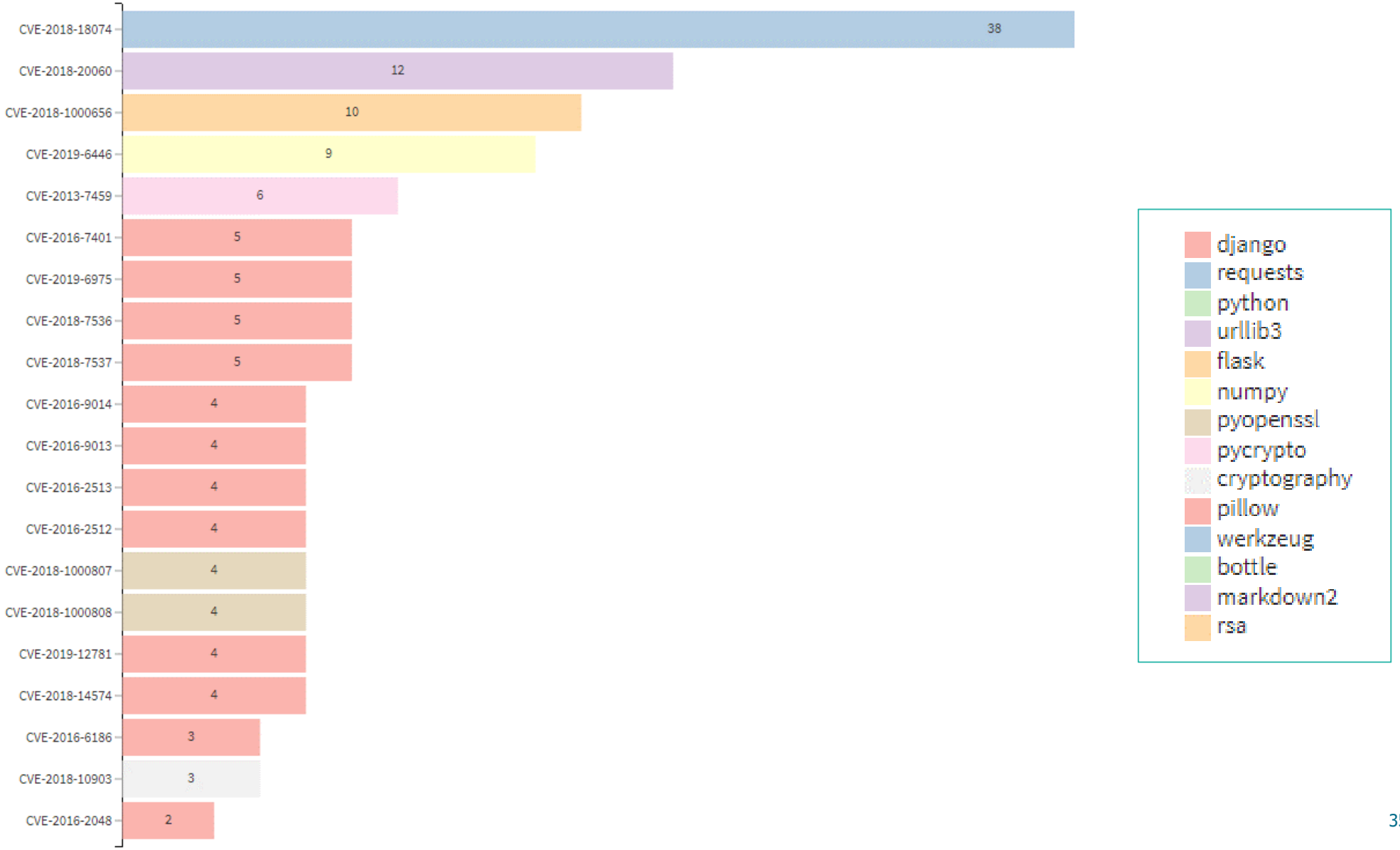


PyCrypto is not **maintained anymore** (since 2013) and **vulnerable** to several bad stuff, **stop using it !**

https://theartofmachinery.com/2017/02/02/dont_use_pycrypto.html

https://blog.sqreen.com/stop-using-pycrypto-use-pycryptodome/

Top 20 of the **most common vulnerabilities** in infosec Python tools using **3rd-party modules**



So you do want to access the data ?

Code, details, and output datasets of the study are available on Github

`https://github.com/maaaaz/adecadeofinfosectools`

01 A bit of context

02 Tools during the last decade

03 Make your tool great (again)

04 Wrapping it up

Golden rules for modern tools (from my personal experience)

Use a standard argument parsing library and accept arguments

Build it with modularity to ease public contributions

Use asynchronous execution
(IO bounded → multithreading
CPU bounded → multiprocessing)

Make it usable worldwide
UTF-8 ! UTF8 ! UTF-8 !

Provide multiple verbosity levels

Package it and make it easily installable

Provide prebuilt binaries or containers
(it helps attackers AND defenders)

Encrypt traffic

Provide easy-to-parse output in CSV / JSON

Support NTLM authentication

Support Kerberos authentication

Support HTTP proxy traversal

Support SOCKS proxification

Allow single and bulk input

Use non vulnerable dependencies

01 A bit of context

02 Tools during the last decade

03 Make your tool great (again)

04 Wrapping it up

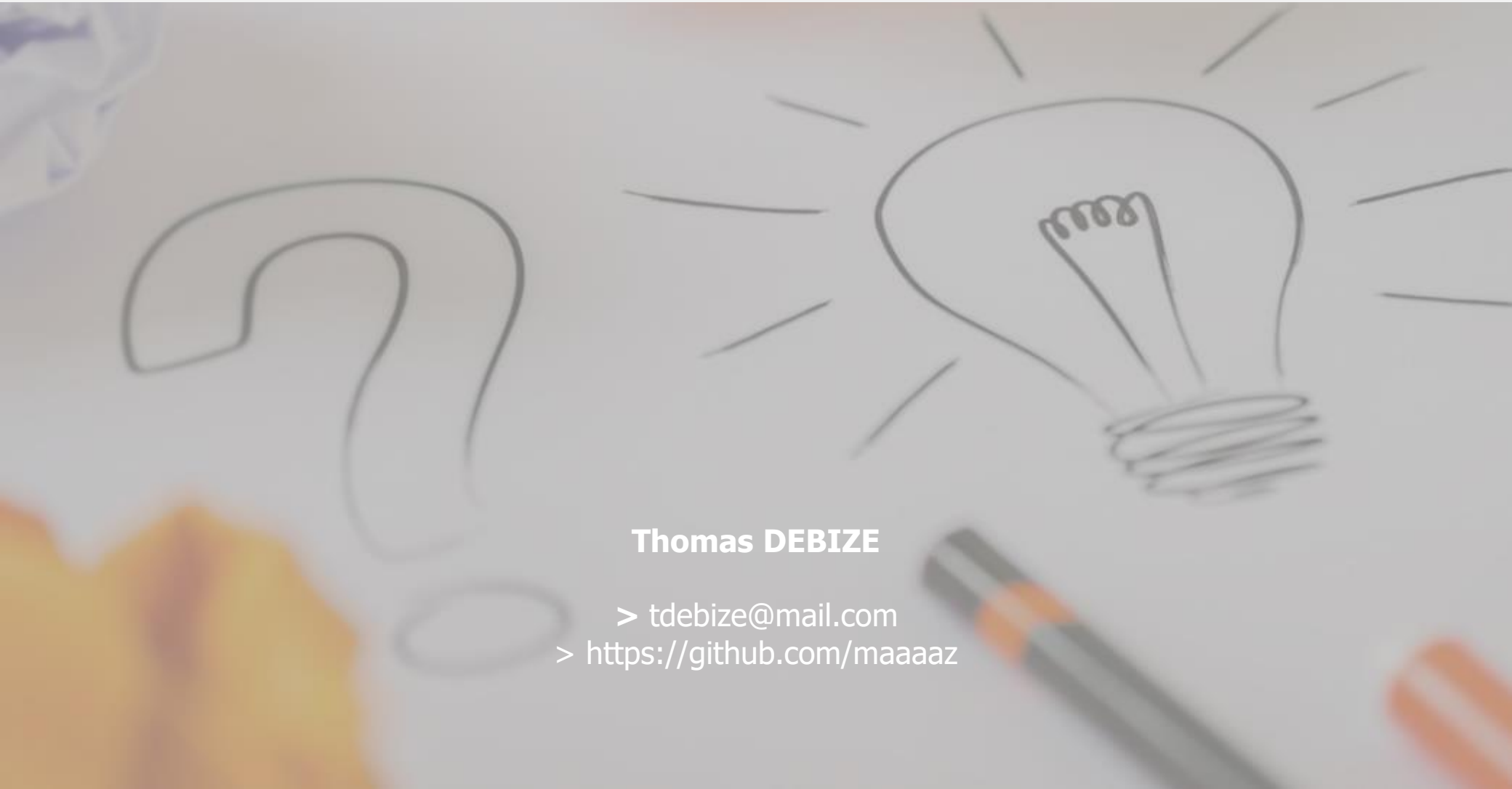
[INSERT A CONCLUSION HERE]

Good tools **work,**

Better tools **scale,**

Great tools **last.**

Questions ?



Thomas DEBIZE

> tdebize@mail.com
> https://github.com/maaaaz