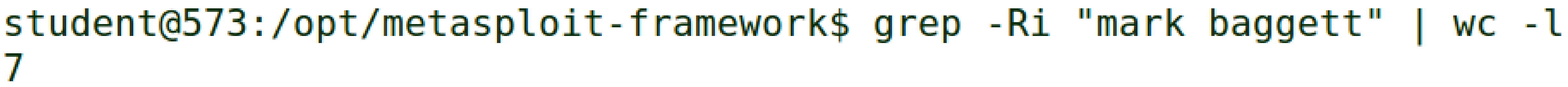
Enhance your Threat Hunting

with these open source tools

@markbaggett

Get-ADUser -Filter "Mark Baggett"| fl -Properties *

- Mark Baggett
- Penetration Testing and Incident Response Consulting
- Senior SANS Instructor
- Author of SANS SEC573 Automating InfoSec with Python
- Masters in Information Security Engineering
- GSE #15
- DoD Advisor, Former CISO 18+ years commercial



```
student@573:/opt/metasploit-framework$ grep -Ri "mark baggett" | wc -l
7
```

These first two tools integrate into your SIEM or ZEEK IDS

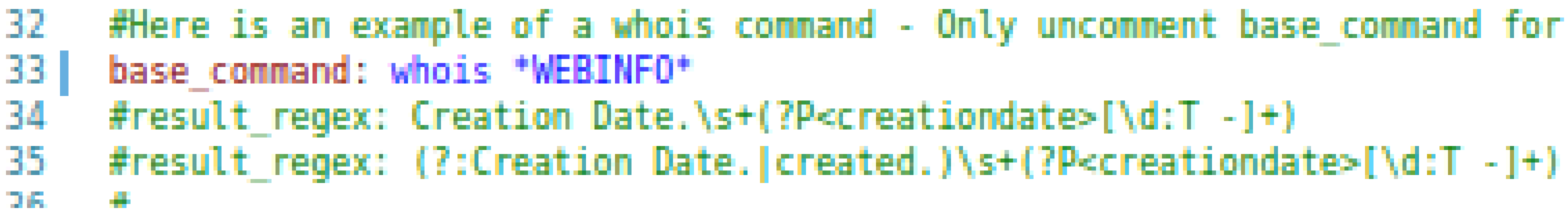
- Enrich the data your SIEM is already collecting
- These web services run on a single host on your network
- Each service listens on a different TCP port

APIify - Make ANYTHING query-able by your SEIM or ZEEK

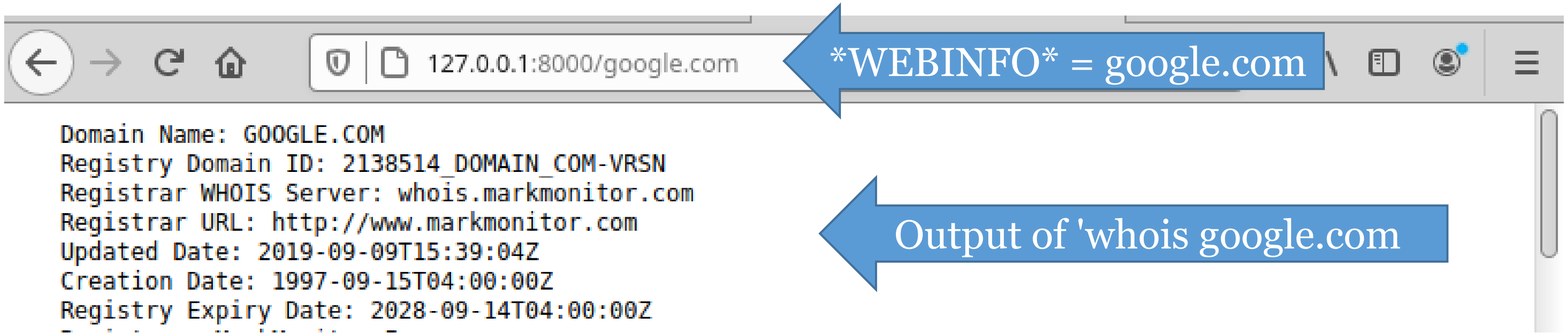
- API is intended to automate those first few steps you take with every investigation you do.
- APIify automates running any command and making it available to you SEIM:
  - Cached responses and detailed statistics
  - Run any command line tool and consume this with your SEIM
  - Define regular expressions to select what data that is returned
- Lets look at a sample use case…
  - Make APIify do what domain_stats 1.0 did… BUT BETTER, FASTER AND MORE SCALEABLE!!

Using APIify to replace the old domain_stats

- apiify.yaml configurations control which command is run
- *WEBINFO* is replaced with what ever is typed on the URL



```
#Here is an example of a whois command - Only uncomment base_command for
base_command: whois *WEBINFO*
#result_regex: Creation Date.\s+(?P<creationdate>[\d:T -]+)
#result_regex: (?:Creation Date.|created.)\s+(?P<creationdate>[\d:T -]+)
#
```



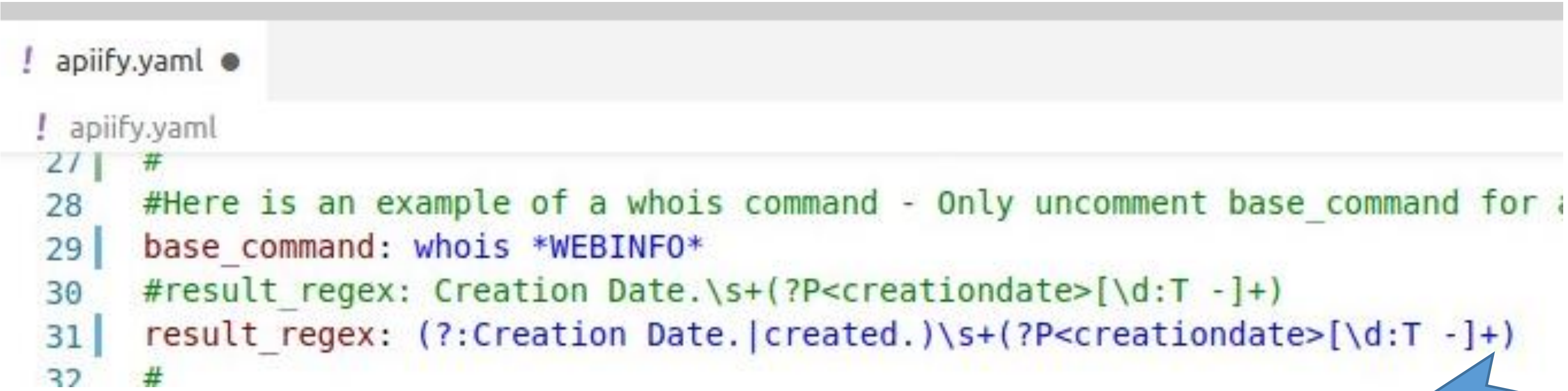
WEBINFO = google.com

```
Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
```

Output of 'whois google.com

Enable JSON responses

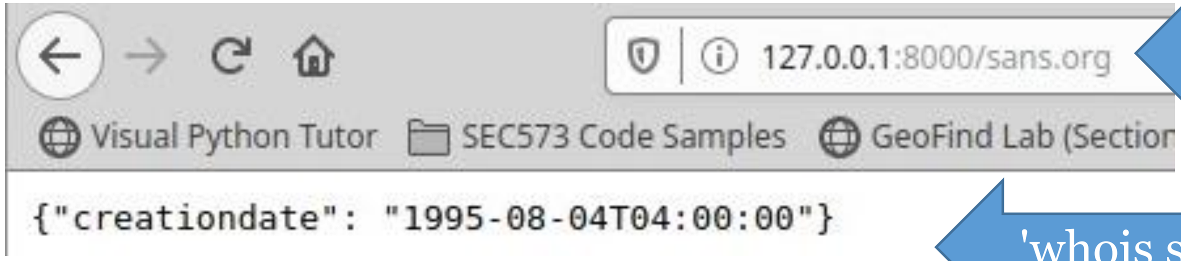
- The "result_regex" option can provide an optional regex



```
#
#Here is an example of a whois command - Only uncomment base_command for
base_command: whois *WEBINFO*
#result_regex: Creation Date.\s+(?P<creationdate>[\d:T -]+)
result_regex: (?:Creation Date.|created.)\s+(?P<creationdate>[\d:T -]+)
#
```

Optional REGEX

- Now you get JSON responses

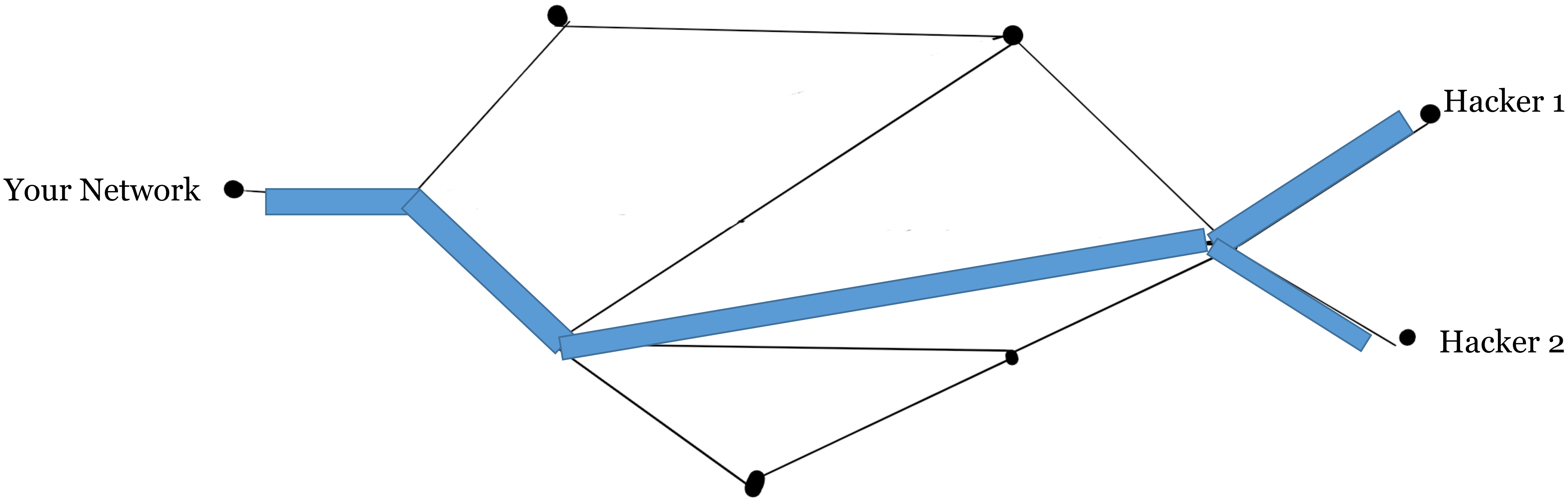


```
{"creationdate": "1995-08-04T04:00:00"}
```

WEBINFO = sans.org

'whois sans.org' regex results

Another Use Case: Traceroutes to identify shared infrastructure



traceroute #1	Hop 1 = 1.1.1.1, Hop 2 = 5.5.5.5, Hop 3 = 200.200.200.200
traceroute #2	Hop 1 = 1.1.1.1, Hop 2 = 5.5.5.5, Hop 3 = 200.200.200.200

Finding Shared Infrastructure

- I want to TRACEROUTE to every IP that generates some ZEEK alert
- Collect the path from the traceroute and record it in my SEIM
- A traceroute to google.com takes about 1 minute and 6 seconds

```
root@573:~# time traceroute www.google.com
traceroute to www.google.com (108.177.122.103), 30 hops max, 60 byte packets
 1  homefirewall.localdomain (x.x.x.1)  12.506 ms  22.639 ms  22.568 ms
...
22  108.177.122.103 (108.177.122.103)  16.196 ms *  28.300 ms

real	1m6.117s
user	0m0.000s
sys	0m0.028s
```

Make traceroute Faster!

- Traceroute has some useful options

-n	Do not resolve DNS Names for hops
-f #	Skill the first # number of hops on my side
-q #	Repeat the trace # number of times (default is 3)

- Additionally use TCP port 80 for reliability

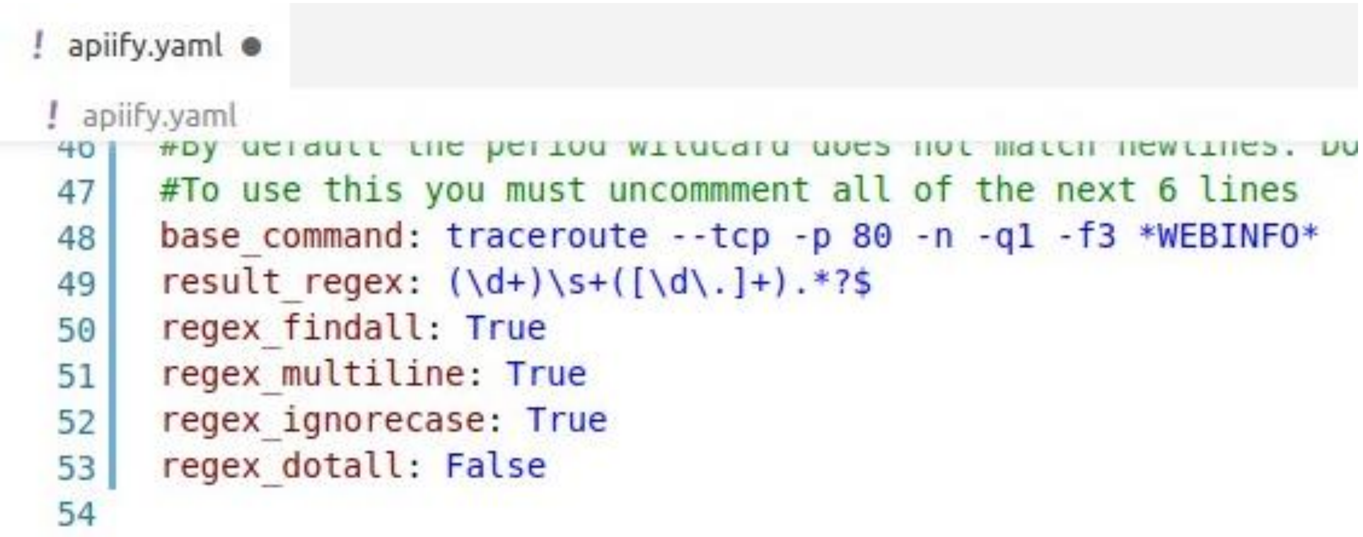
```
root@573:~# time traceroute --tcp -p 80 -n -q1 -f3  google.com
traceroute to google.com (64.233.177.139), 30 hops max, 60 byte packets
 3  208.188.184.1  28.377 ms
21  64.233.177.139  33.881 ms

real	0m0.398s
user	0m0.000s
sys	0m0.006s
```

EXCEPTIONAL expect 10 seconds per IP the FIRST TIME ONLY

Just change the apiify.yaml file!

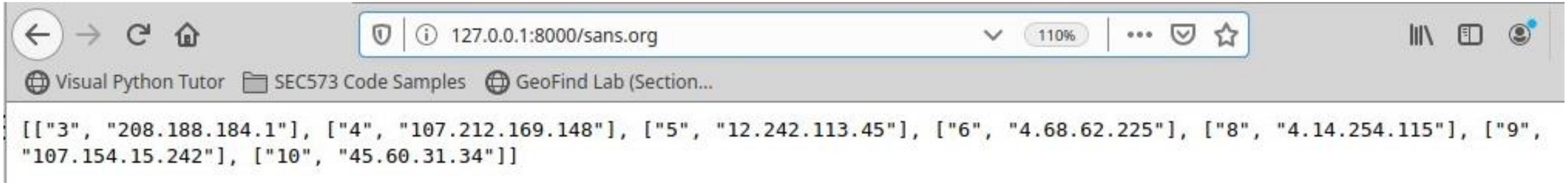
- Another few changes to apiify.yaml



```
#By default the period wildcard does not match newlines. Do
#To use this you must uncommment all of the next 6 lines
base_command: traceroute --tcp -p 80 -n -q1 -f3 *WEBINFO*
result_regex: (\d+)\s+([\d\.]+).*?$
regex_findall: True
regex_multiline: True
regex_ignorecase: True
regex_dotall: False

```

- And now your SIEM or ZEEK contain things like this …



```
[["3", "208.188.184.1"], ["4", "107.212.169.148"], ["5", "12.242.113.45"], ["6", "4.68.62.225"], ["8", "4.14.254.115"], ["9",
"107.154.15.242"], ["10", "45.60.31.34"]]
```

No SIEM - No Problem - dump_cache.py

- Tell API-ify.py to commit its data to disk
  - `http://127.0.0.1:8000/save`
- Then you can use dump_cache.py to search all the data with it sorted in key, age, hit count, or data order
- Key is *WEBINFO*

```
student@573:~/apiify$ python dump_cache.py -h
usage: dump_cache.py [-h] [-s {key,age,count,data}]

optional arguments:
  -h, --help            show this help message and exit
  -s {key,age,count,data}, --sort {key,age,count,data}
                        Specify how you want the data sorted. Default is key.
```

Inspecting cached results reveals "related" IP Addresses

- Run the ISC block list through it

```
student@573:~/apiify$ wget -O- https://isc.sans.edu/block.txt | cut -f1 | grep -P -e
"^\d" | xargs -I {} wget -O- http://127.0.0.1:8000/{}
```

- Look for relationships with a suspect IP

```
student@573:~/apiify$ python dump_cache.py -s data | grep 80.82.70.0 -C1
93.174.93.0, 2020-07-29 12:36:50.062515, 9, b'[["3", "208.188.184.1"], ["5",
"12.242.113.6"], ["6", "216.66.24.133"], ["7", "184.105.80.161"], ["8",
"184.105.223.166"], ["9", "72.52.92.165"], ["10", "72.52.92.214"]]'
80.82.70.0, 2020-07-29 12:37:10.092859, 50, b'[["3", "208.188.184.1"], ["5",
"12.242.113.6"], ["6", "216.66.24.133"], ["7", "184.105.80.161"], ["8",
"184.105.223.166"], ["9", "72.52.92.165"], ["10", "72.52.92.214"]]'
89.248.174.0, 2020-07-29 12:37:41.493890, 1, b'[["3", "208.188.184.1"], ["5",
"12.242.113.6"], ["6", "216.66.24.133"], ["7", "184.105.80.161"], ["8",
"184.105.223.166"], ["9", "72.52.92.165"], ["10", "72.52.92.214"]]'
```

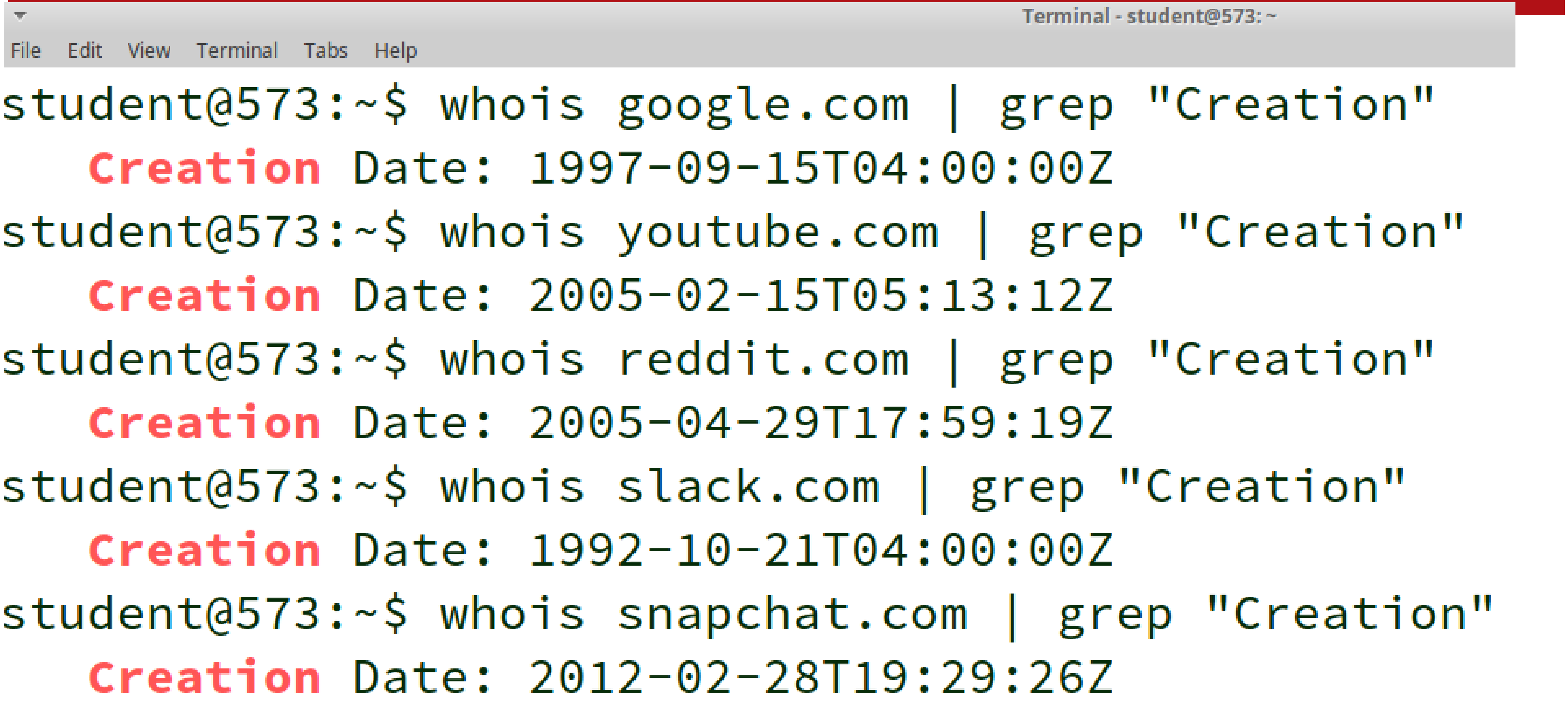
APIIFY Sample Configurations include

- PING
- Entire WHOIS record
- Select Just the Creation Date from WHOIS
- Traceroute with just the last hop
- Full Traceroute to host
- Geolocation Lookup IP with Web API
- Query ISC API for IP based Threat intelligence

Domain_stats is dead. Long Live Domain_stats.

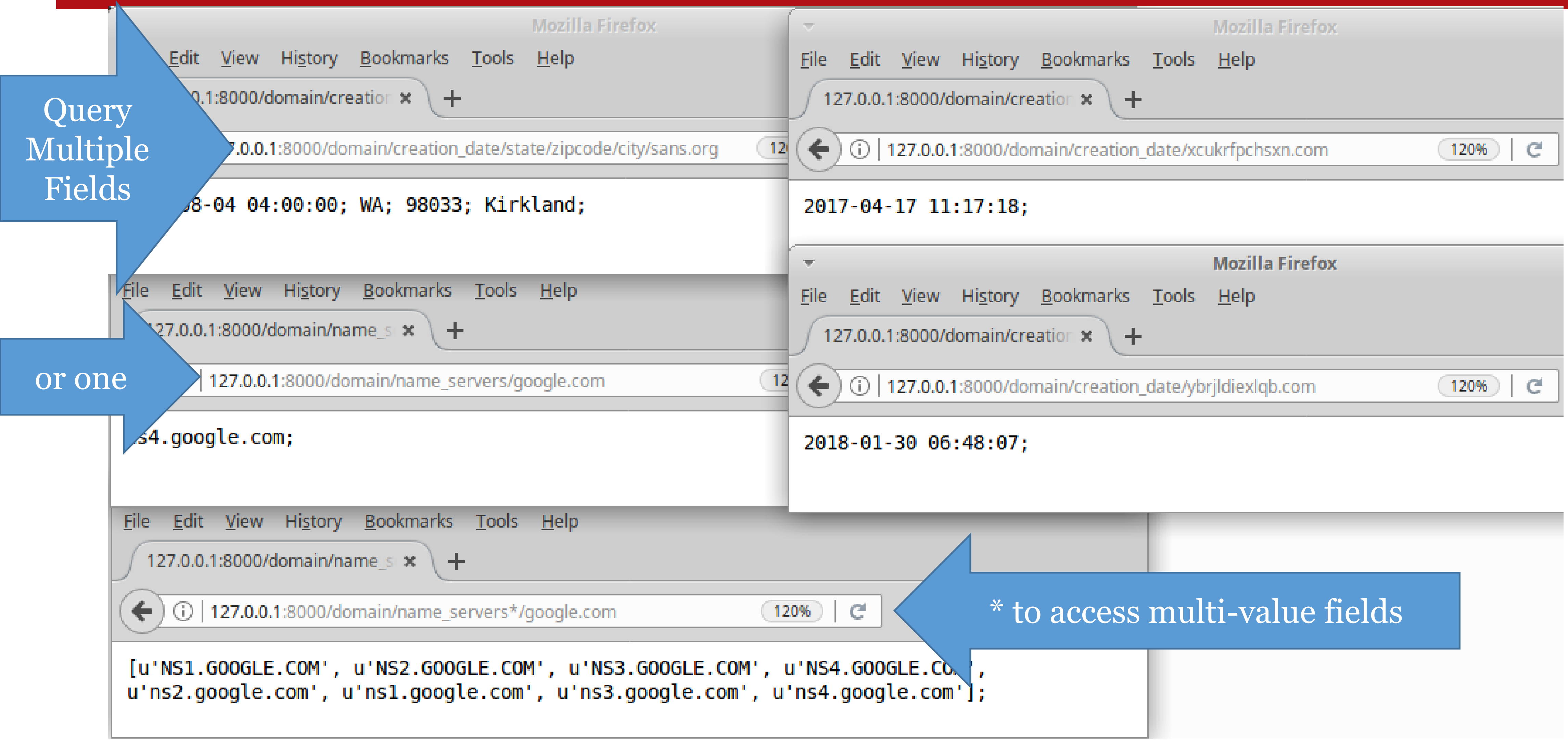
- I wrote a tool named domain_stats.
- People liked it.
- It had performance issues at high volumes.
  - Whois queries are rate limited
  - Cache growth is unlimited. Memory consumption is huge.
- People really only used one feature - Domain Creation dates

"Normal" Domain Creation Dates



```
student@573:~$ whois google.com | grep "Creation"
   Creation Date: 1997-09-15T04:00:00Z
student@573:~$ whois youtube.com | grep "Creation"
   Creation Date: 2005-02-15T05:13:12Z
student@573:~$ whois reddit.com | grep "Creation"
   Creation Date: 2005-04-29T17:59:19Z
student@573:~$ whois slack.com | grep "Creation"
   Creation Date: 1992-10-21T04:00:00Z
student@573:~$ whois snapchat.com | grep "Creation"
   Creation Date: 2012-02-28T19:29:26Z
```

This slide dedicated in remembrance of domain_stats 1.0



Query Multiple Fields

7.0.0.1:8000/domain/creation_date/state/zipcode/city/sans.org

```
8-04 04:00:00; WA; 98033; Kirkland;
```

or one

127.0.0.1:8000/domain/name_servers/google.com

```
s4.google.com;
```

127.0.0.1:8000/domain/name_servers*/google.com

```
[u'NS1.GOOGLE.COM', u'NS2.GOOGLE.COM', u'NS3.GOOGLE.COM', u'NS4.GOOGLE.CO ',
u'ns2.google.com', u'ns1.google.com', u'ns3.google.com', u'ns4.google.com'];
```

* to access multi-value fields

127.0.0.1:8000/domain/creation_date/xcukrfpchsxn.com

```
2017-04-17 11:17:18;
```

127.0.0.1:8000/domain/creation_date/ybrjldiexlqb.com

```
2018-01-30 06:48:07;
```

INTRODUCING

Domain Stats 2.0

Domain Stats 2.0!!!!!!

- Domain_stats has been completely reimagined to focus on what people used… The domain creation date
- Focuses on optimized delivery of the following data:
  - seen_by_web - Age of the domain - ie First Contact on the web
  - seen_by_you - First contact date for your network
  - seen_by_isc - First contact date for the domain_stats hivemind hosted by ISC
  - Alerts system from ISC and on the very first request ever for a domain
- ZEEK Deployment script
- Docker/container image deployment

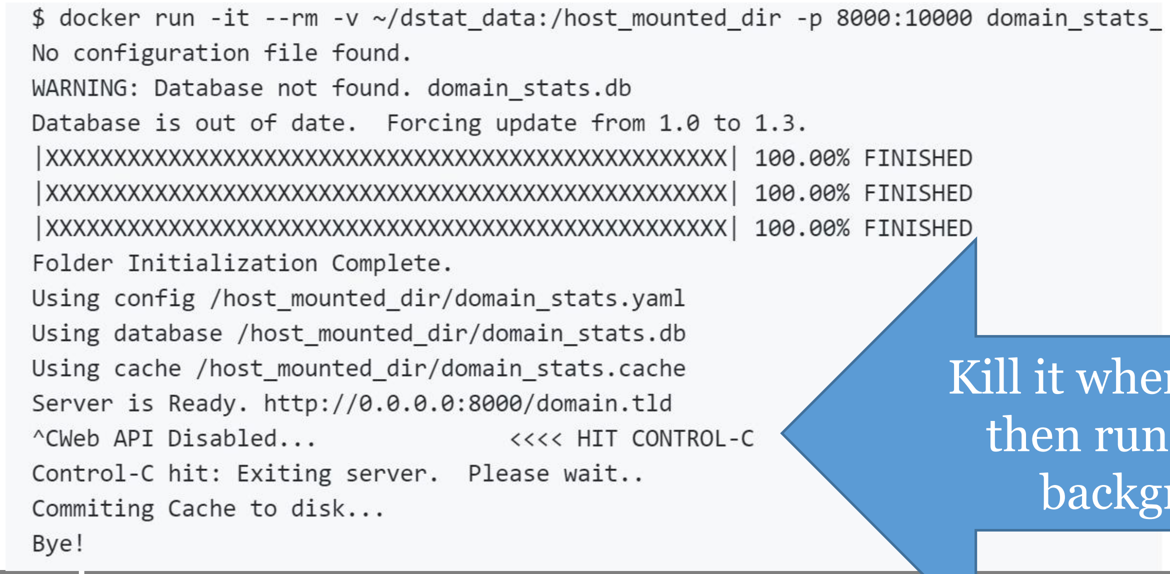
Focus is on resolution at the speed of logs!

- The whois servers will block you if you have any significant amount of requests. This was the main catalyst for a rewrite of the tool.
- Program maintains a local database of most common domains registration "seen_by_web" so no network requests are required.
- Memory cache minimizes Disk IO delays for most recent used items
- All responses are cached or committed to database until the domain registration expiration date.
- Only do slow network request for uncommon domains and only do it once
- Even optimize those infrequent domains by having ISC is purchase access to direct feeds so they can eliminate this bottle neck, optimize lookups with caches, provide alerts and "First Seen" data

How easy is it to deploy with Security Onion?

- Create a docker image, a directory to store your data and run it!
- Your first run should be interactive (-it) so you can see it download updates and configure itself.

```
$ docker build --tag domain_stats_image http://github.com/markbaggett/domain_stats.git
$ mkdir ~/dstat_data
$ docker run -it --rm -v ~/dstat_data:/host_mounted_dir -p 8000:10000 domain_stats_image
```



```
$ docker run -it --rm -v ~/dstat_data:/host_mounted_dir -p 8000:10000 domain_stats_
No configuration file found.
WARNING: Database not found. domain_stats.db
Database is out of date.  Forcing update from 1.0 to 1.3.
|XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX| 100.00% FINISHED
|XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX| 100.00% FINISHED
|XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX| 100.00% FINISHED
Folder Initialization Complete.
Using config /host_mounted_dir/domain_stats.yaml
Using database /host_mounted_dir/domain_stats.db
Using cache /host_mounted_dir/domain_stats.cache
Server is Ready. http://0.0.0.0:8000/domain.tld
^CWeb API Disabled...              <<<< HIT CONTROL-C
Control-C hit: Exiting server.  Please wait..
Commiting Cache to disk...
Bye!
```

You specify folder outside docker where data is stored long term

Kill it when its done, then run it in the background

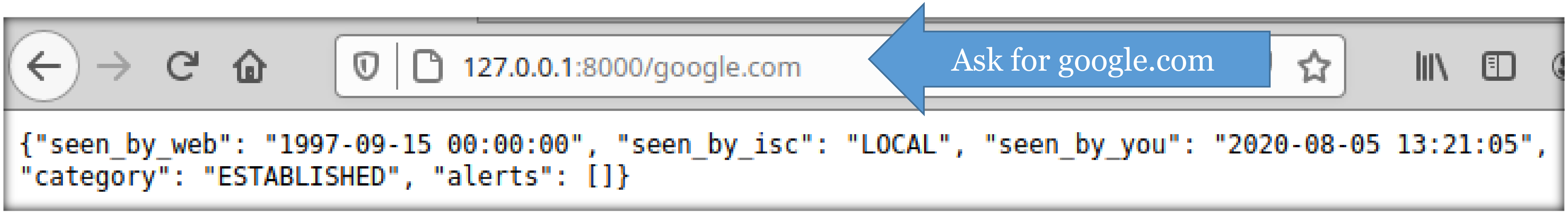
Once installed let it run!

- After initial configuration use "docker run" again without "-it"
- Stop and start the service with docker start and docker stop

```
$ docker run -d --name domain_stats -v ~/dstat_data:/host_mounted_dir -p 8000:8000 domain_stats_image
$ docker stop domain_stats
$ docker start domain_stats
```

- "domain_stats.yaml" has many configuration option
- All data is kept outside the container in the specified folder ("dstat_data" in the example above)
- Point your SEIM, ZEEK or custom apps at the URL

Domain_stats In Action - Normal Domains

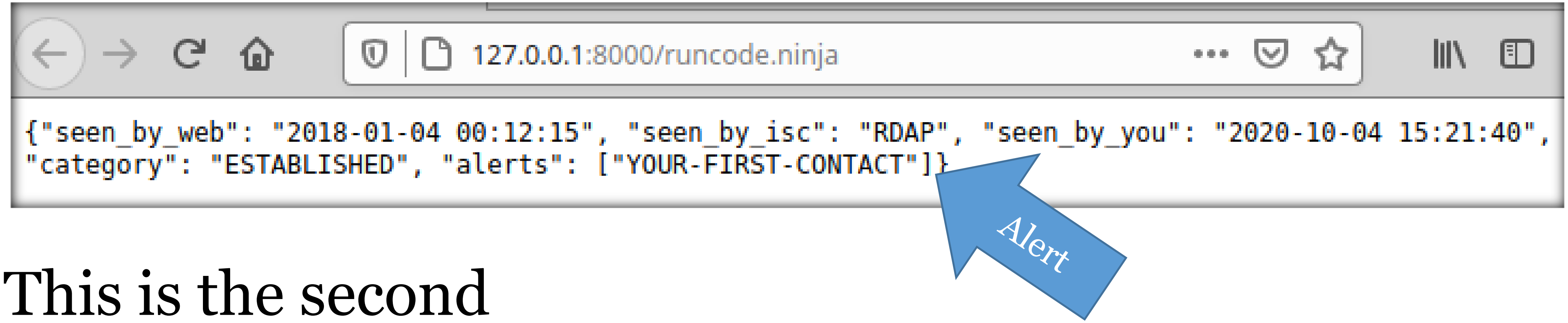


```
{"seen_by_web": "1997-09-15 00:00:00", "seen_by_isc": "LOCAL", "seen_by_you": "2020-08-05 13:21:05",
"category": "ESTABLISHED", "alerts": []}
```

- "`seen_by_web`" is domain registration date
- "`seen_by_isc`" - Local, RDAP or date the ISC first saw the domain
- "`seen_by_you`" is the date your organization first saw this domain used
- Category:
  - ESTABLISHED - means registration is > 2 years old
  - NEW - means it is a newly registered domain and deserves some scrutiny
- Alerts: "Your First Contact", "ISC First Contact" and more

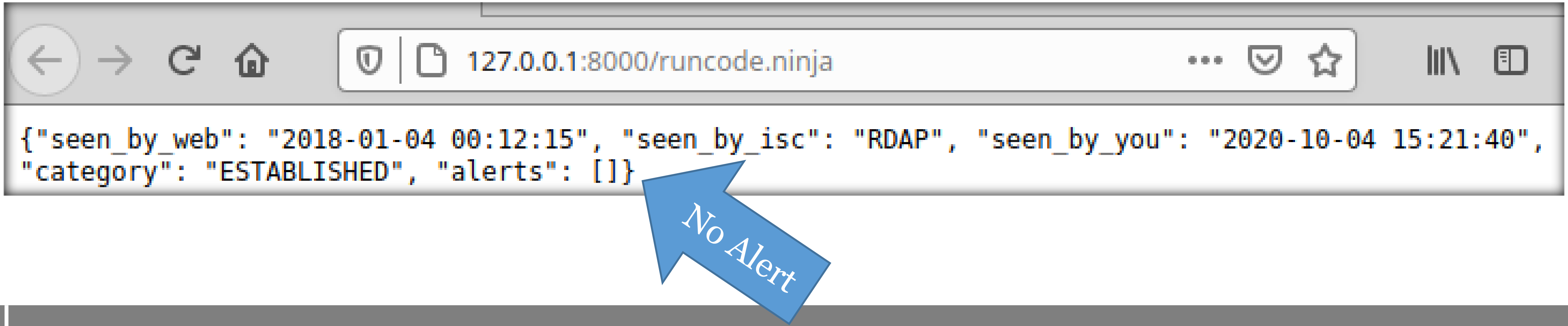
Domain_stats In Action - New to you

- Here is the first time we ever lookup `runcode.ninja`



```
{"seen_by_web": "2018-01-04 00:12:15", "seen_by_isc": "RDAP", "seen_by_you": "2020-10-04 15:21:40",
"category": "ESTABLISHED", "alerts": ["YOUR-FIRST-CONTACT"]}
```

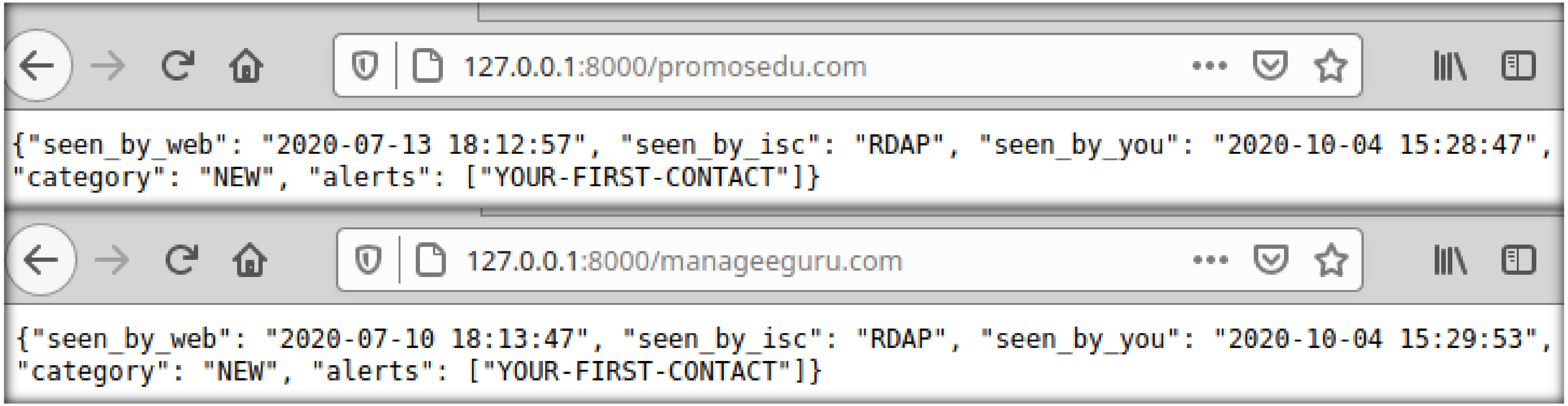
- This is the second



```
{"seen_by_web": "2018-01-04 00:12:15", "seen_by_isc": "RDAP", "seen_by_you": "2020-10-04 15:21:40",
"category": "ESTABLISHED", "alerts": []}
```

New to you, New to the world

- Here are the result from a few domain identified as evel by malc0de.com



127.0.0.1:8000/promosedu.com

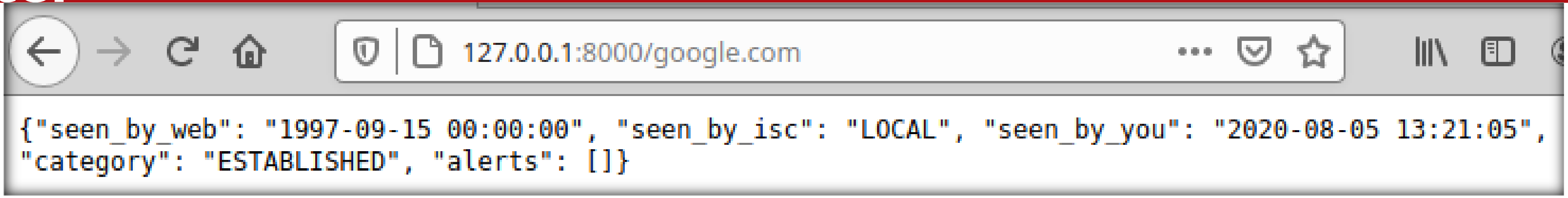
```
{"seen_by_web": "2020-07-13 18:12:57", "seen_by_isc": "RDAP", "seen_by_you": "2020-10-04 15:28:47",
"category": "NEW", "alerts": ["YOUR-FIRST-CONTACT"]}
```

127.0.0.1:8000/manageeguru.com

```
{"seen_by_web": "2020-07-10 18:13:47", "seen_by_isc": "RDAP", "seen_by_you": "2020-10-04 15:29:53",
"category": "NEW", "alerts": ["YOUR-FIRST-CONTACT"]}
```

- The "NEW" category and "YOUR-FIRST-CONTACT" alert makes these stand out from the other domains in my logs

New to you, New to the world, New to the Internet Storm Center



```
{"seen_by_web": "1997-09-15 00:00:00", "seen_by_isc": "LOCAL", "seen_by_you": "2020-08-05 13:21:05",
"category": "ESTABLISHED", "alerts": []}
```

- This tool is ready for you to use TODAY
- Name resolution is limited to
  - LOCAL - Your localized database prepopulate with 1000s of domains.
  - RDAP - Today the protocol has limited eTLD support
- Pending Enhancement:
  - Your lookups CAN be proxied through Internet Storm center to support all domains via whois
  - This enabled community base "Seen by ISC" alerts and first seen dates

What if the tool doesn't do exactly what you need?

- Let me know. I'm happy to support these tools!
- I enjoy writing tools and I hope these are truly useful to you.
- But more than anything I want to teach you to write your own tools.

- Give a man a fish, and you feed him for a day; show him how to catch fish, and you feed him for a lifetime. - Proverbs 12:10