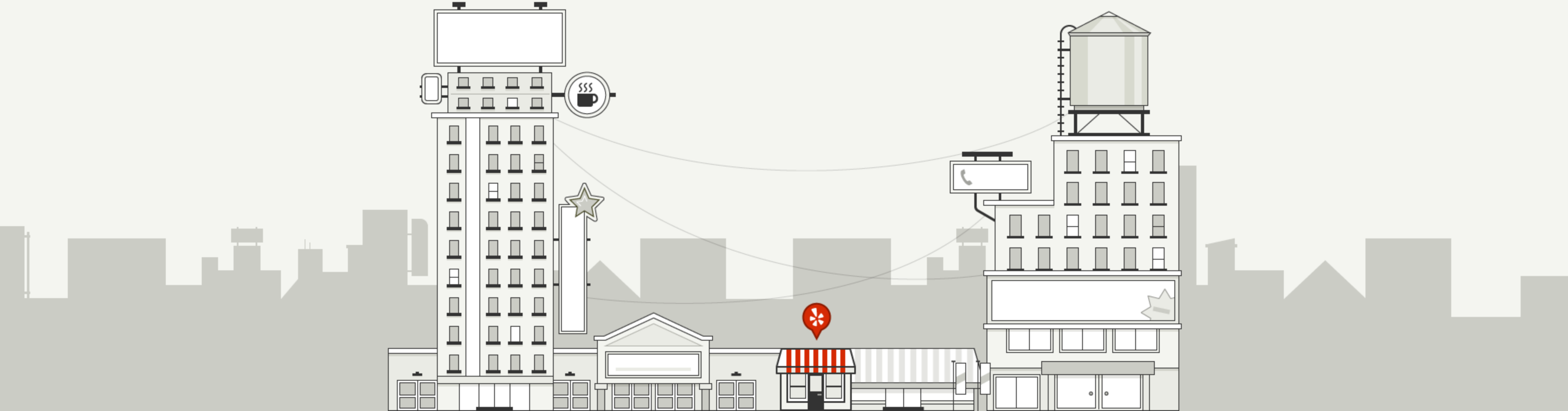
OSXCollector

Automated forensic evidence collection & analysis for OS X

Jakub (Kuba) Sendor
@jsendor

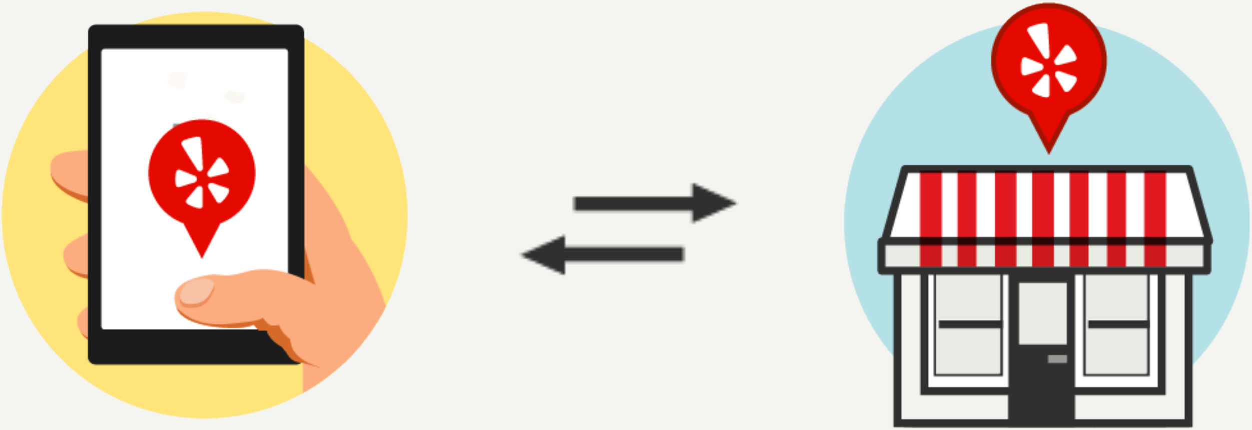


whoami

- Joined Yelp security team in July 2014.
- Mostly involved in malware incident response.
- Also working on automating our security processes.
- Previously worked at SAP in Sophia Antipolis (France) in the Security & Trust research group.
- Graduated in 2011 from AGH University of Science and Technology in Kraków (Poland) and Telecom ParisTech/Institut Eurecom (France).

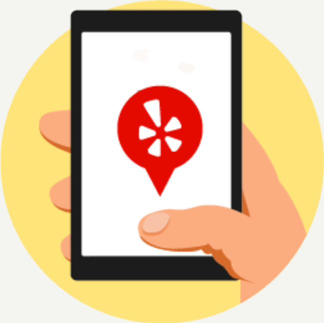
Yelp’s Mission:

Connecting people with great local businesses.



Yelp Stats:

As of Q2 2015



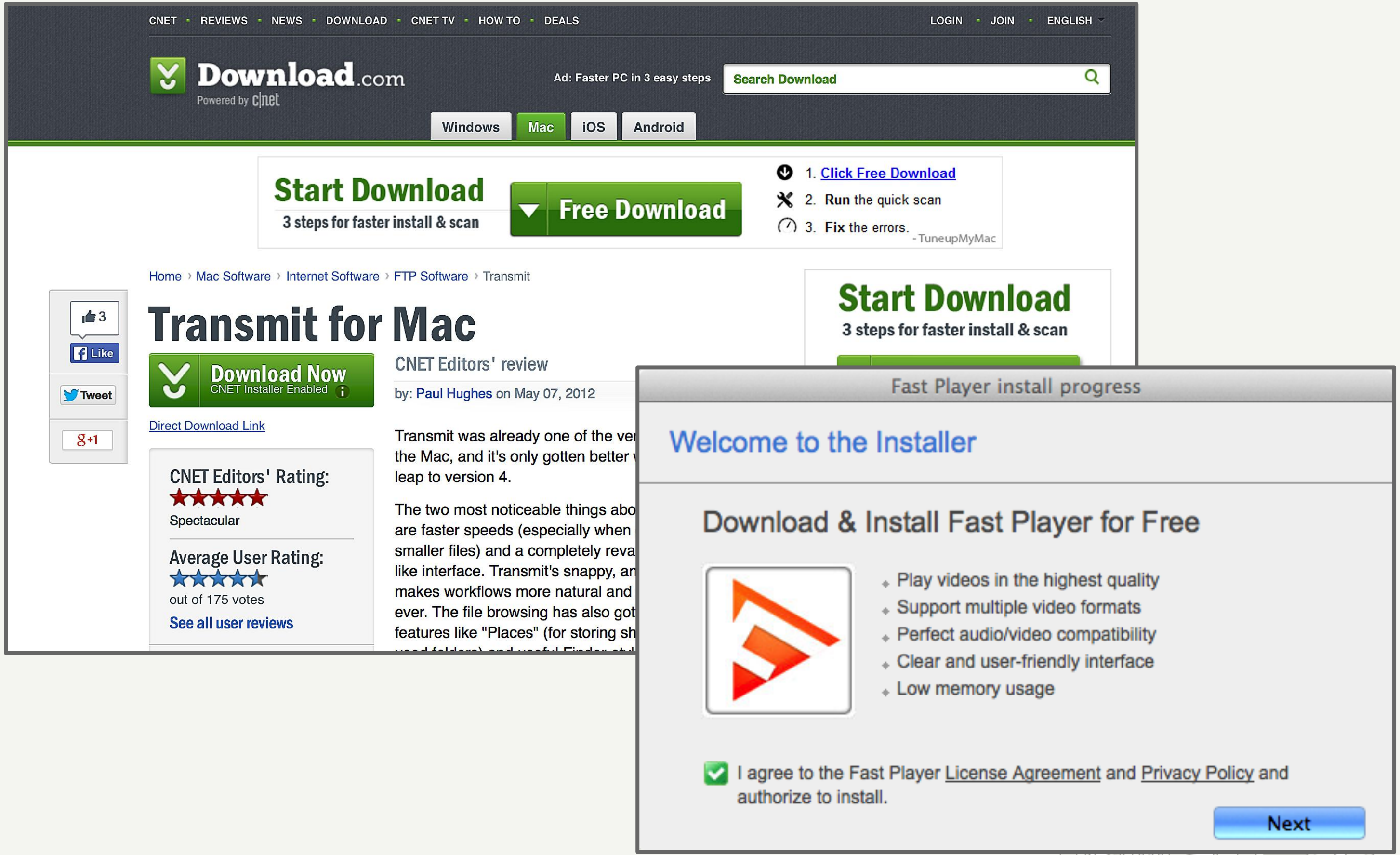
83M

83M

68%

32

>3k employees, most of them using Macs



CNET · REVIEWS · NEWS · DOWNLOAD · CNET TV · HOW TO · DEALS LOGIN · JOIN · ENGLISH

Download.com
Powered by cnet

Ad: Faster PC in 3 easy steps

Search Download

Windows Mac iOS Android

Start Download
3 steps for faster install & scan

Free Download

1. Click Free Download
2. Run the quick scan
3. Fix the errors.
- TuneupMyMac

Home › Mac Software › Internet Software › FTP Software › Transmit

Transmit for Mac

3 Like Tweet g+1

Download Now
CNET Installer Enabled

Direct Download Link

CNET Editors' Rating:
★★★★★
Spectacular

Average User Rating:
★★★★★
out of 175 votes

See all user reviews

CNET Editors' review

by: Paul Hughes on May 07, 2012

Transmit was already one of the ver the Mac, and it's only gotten better leap to version 4.

The two most noticeable things abo are faster speeds (especially when smaller files) and a completely reva like interface. Transmit's snappy, an makes workflows more natural and ever. The file browsing has also got features like "Places" (for storing sh

Start Download
3 steps for faster install & scan

Fast Player install progress

Welcome to the Installer

Download & Install Fast Player for Free

- Play videos in the highest quality
- Support multiple video formats
- Perfect audio/video compatibility
- Clear and user-friendly interface
- Low memory usage

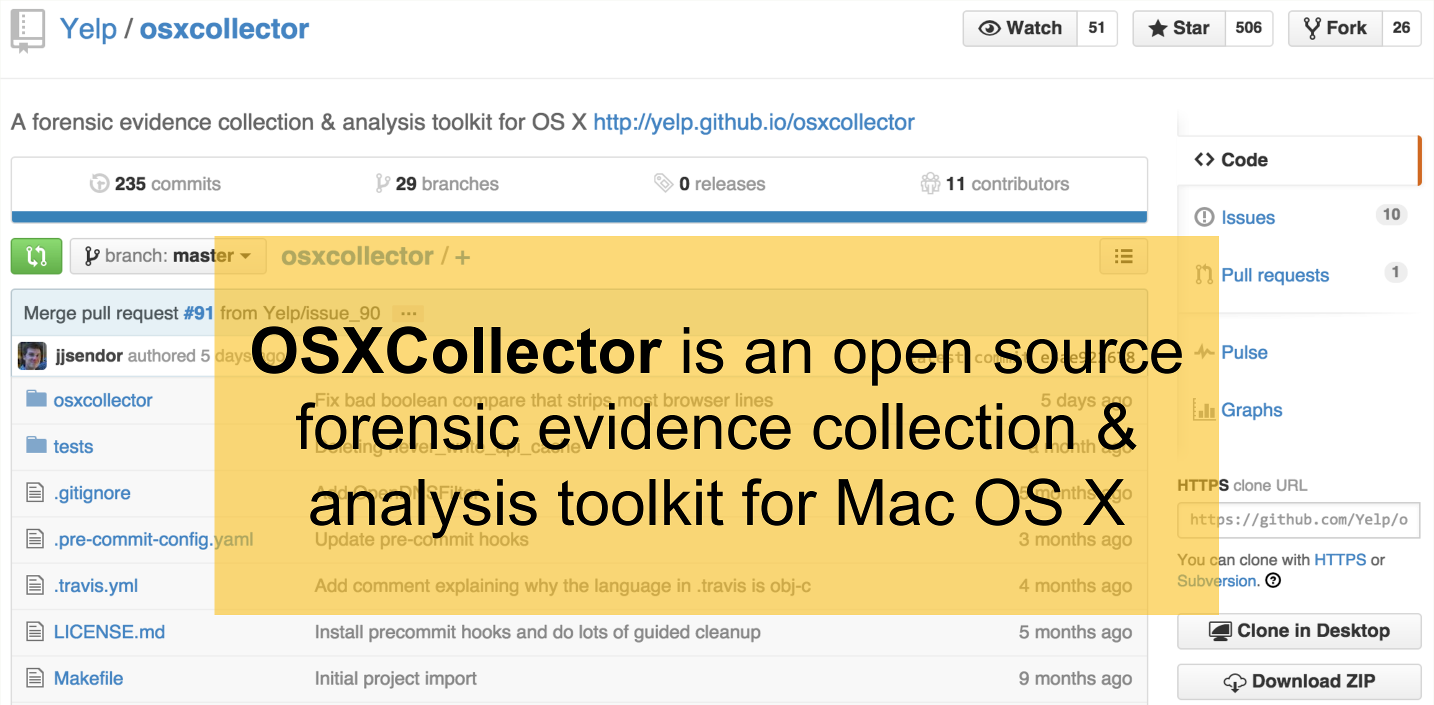
☑ I agree to the Fast Player License Agreement and Privacy Policy and authorize to install.

Next

Virus Download

OS X Grumpy Cat
Introducing OS X 10.FU
The world's most advanced operating system just got Grumpier

https://github.com/Yelp/osxcollector



Yelp / **osxcollector**

Watch 51 | Star 506 | Fork 26

A forensic evidence collection & analysis toolkit for OS X http://yelp.github.io/osxcollector

235 commits | **29** branches | **0** releases | **11** contributors

branch: **master** | osxcollector / +

Merge pull request **#91** from Yelp/issue_90 …

jjsendor authored 5 days ago | latest commit [illegible]

osxcollector	Fix bad boolean compare that strips most browser lines	5 days ago
tests	Deleting [illegible]_write_api_cache	a month ago
.gitignore	Add OpenDNSFilter	5 months ago
.pre-commit-config.yaml	Update pre-commit hooks	3 months ago
.travis.yml	Add comment explaining why the language in .travis is obj-c	4 months ago
LICENSE.md	Install precommit hooks and do lots of guided cleanup	5 months ago
Makefile	Initial project import	9 months ago

HTTPS clone URL

https://github.com/Yelp/o

You can clone with HTTPS or Subversion.

OSXCollector is an open source forensic evidence collection & analysis toolkit for Mac OS X

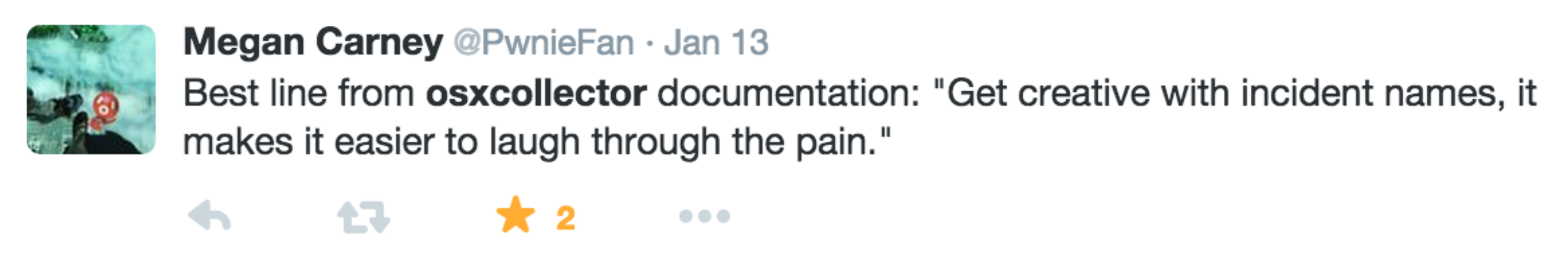
@jsendor

OSXCollector is easy to run

1 Python file

0 dependencies

```
$ sudo osxcollector.py --id DelayedHedgehog
Wrote 35394 lines.
Output in DelayedHedgehog-2015_01_20-19_38_38.tar.gz
$
```



Megan Carney @PwnieFan · Jan 13

Best line from **osxcollector** documentation: "Get creative with incident names, it makes it easier to laugh through the pain."

The output is JSON

JSON is beautiful.

JSON is easy to manipulate.

```
{
  "file_path": "/System/Library/Extensions/Apple_iSight.kext/Contents/MacOS/Apple_iSight",
  "sha2": "19b7b85eaedb17d9565dce872f0d1ea8fc0761f508f28bedcc8606b828cbf614",
  "sha1": "99005b68295c202fd359b46cd1411acea96b2469",
  "md5": "b8cc164b6546e4b13768d8353820b216",
  "ctime": "2014-12-05 16:50:39",
  "mtime": "2014-09-19 00:16:50",
  "osxcollector_section": "kext",
  "osxcollector_incident_id": "DelayedHedgehog-2015_01_20-19_38_38",
  "osxcollector_plist_path": "/System/Library/Extensions/Apple_iSight.kext/Contents/Info.plist",
  "osxcollector_bundle_id": "com.apple.driver.Apple_iSight",
  "signature_chain": [
    "Software Signing",
    "Apple Code Signing Certification Authority",
    "Apple Root CA"
  ]
}
```

OS X stores lots of data in SQLite DBs

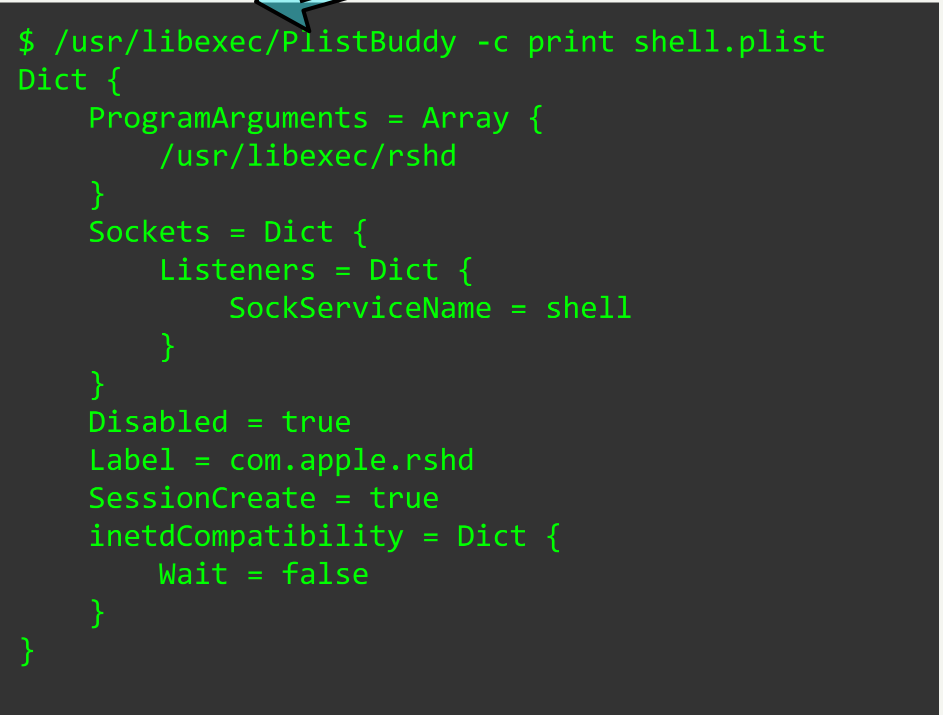
```
# Dump a sqlite DB in a dozen lines of code
with connect(sqlite_db_path) as conn:
    conn.cursor.execute('SELECT * from sqlite_master WHERE type = "table"')
    table_names = [table[2] for table in tables.fetchall()]

    for table in table_names:
        rows = conn.cursor.execute('SELECT * from {0}'.format(table_name))
        column_descriptions = [col[0] for col in conn.cursor.description]
        for row in rows.fetchall():
            record = dict([(key, val) for key, val in zip(column_descriptions, row)])
```

plist == property list

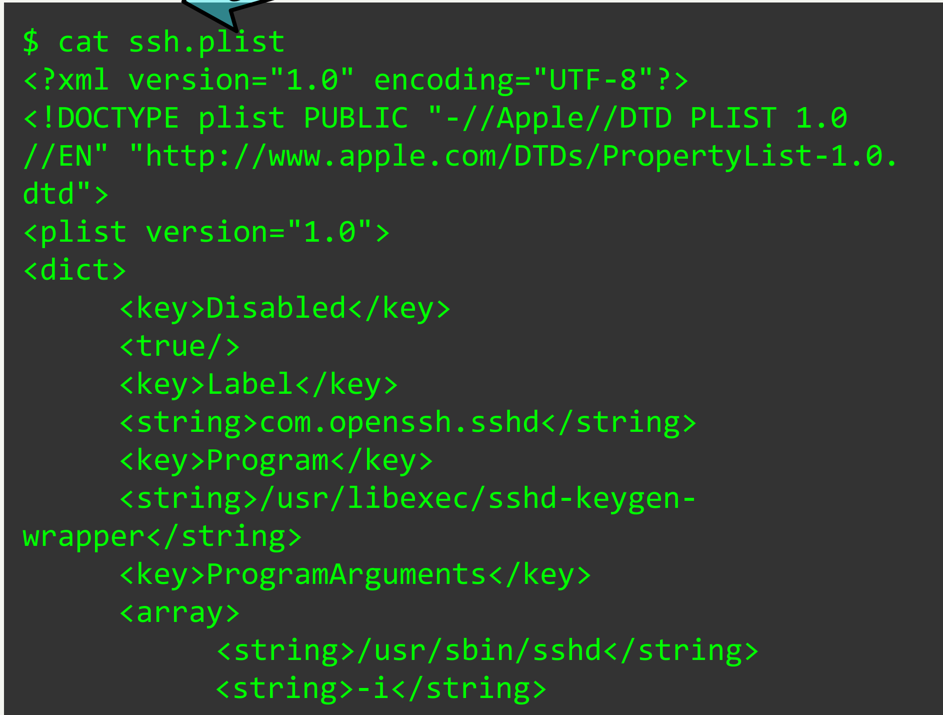
sometimes binary, sometimes plain text

BINARY

UTF-8



```
$ /usr/libexec/PlistBuddy -c print shell.plist
Dict {
    ProgramArguments = Array {
        /usr/libexec/rshd
    }
    Sockets = Dict {
        Listeners = Dict {
            SockServiceName = shell
        }
    }
    Disabled = true
    Label = com.apple.rshd
    SessionCreate = true
    inetdCompatibility = Dict {
        Wait = false
    }
}
```



```
$ cat ssh.plist
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0
//EN" "http://www.apple.com/DTDs/PropertyList-1.0.
dtd">
<plist version="1.0">
<dict>
     <key>Disabled</key>
     <true/>
     <key>Label</key>
     <string>com.openssh.sshd</string>
     <key>Program</key>
     <string>/usr/libexec/sshd-keygen-
wrapper</string>
     <key>ProgramArguments</key>
     <array>
           <string>/usr/sbin/sshd</string>
           <string>-i</string>
```

OSXCollector uses Foundation

Foundation is a *nice* Objective-C wrapper.

```
import Foundation

# Look! Incredibly long objc style function names!
plist_nsdata, error_message = Foundation.NSData.dataWithContentsOfFile_options_error_(
                              plist_path, Foundation.NSUncachedRead, None)

# Seriously, incredibly long function names!
plist_dict, _, _ = Foundation.NSPropertyListSerialization. \
                   propertyListFromData_mutabilityOption_format_errorDescription_( \
                   plist_nsdata, Foundation.NSPropertyListMutableContainers, \
                   None, None)
```

Forensic Collection

OS System Info

Applications

Web Browser Info

Kernel Extensions

Quarantines

Email Info

Downloads

Startup Items

Groups & Accounts

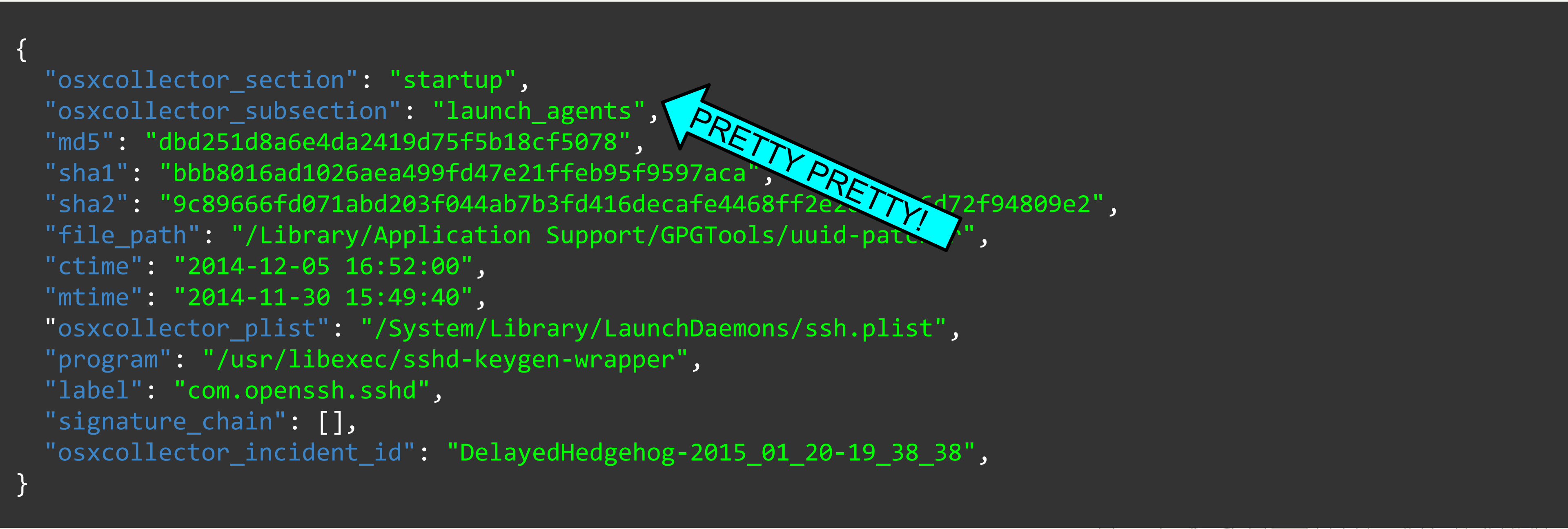
Common keys in entries

path, hashes, timestamps, signature chain, ...

```
{
  "file_path": "/System/Library/Extensions/Apple_iSight.kext/Contents/MacOS/Apple_iSight",
  "sha2": "19b7b85eaedb17d9565dce872f0d1ea8fc0761f508f28bedcc8606b828cbf614",
  "sha1": "99005b68295c202fd359b46cd1411acea96b2469",
  "md5": "b8cc164b6546e4b13768d8353820b216",
  "ctime": "2014-12-05 16:50:39",
  "mtime": "2014-09-19 00:16:50",
  "osxcollector_section": "kext",
  "osxcollector_incident_id": "DelayedHedgehog-2015_01_20-19_38_38",
  "osxcollector_plist_path": "/System/Library/Extensions/Apple_iSight.kext/Contents/Info.plist",
  "osxcollector_bundle_id": "com.apple.driver.Apple_iSight",
  "signature_chain": [
    "Software Signing",
    "Apple Code Signing Certification Authority",
    "Apple Root CA"
  ]
}
```

Startup items run on boot

Malware running at startup is basically game over.



```
{
  "osxcollector_section": "startup",
  "osxcollector_subsection": "launch_agents",
  "md5": "dbd251d8a6e4da2419d75f5b18cf5078",
  "sha1": "bbb8016ad1026aea499fd47e21ffeb95f9597aca",
  "sha2": "9c89666fd071abd203f044ab7b3fd416decafe4468ff2e2[illegible]d72f94809e2",
  "file_path": "/Library/Application Support/GPGTools/uuid-patc[illegible]",
  "ctime": "2014-12-05 16:52:00",
  "mtime": "2014-11-30 15:49:40",
  "osxcollector_plist": "/System/Library/LaunchDaemons/ssh.plist",
  "program": "/usr/libexec/sshd-keygen-wrapper",
  "label": "com.openssh.sshd",
  "signature_chain": [],
  "osxcollector_incident_id": "DelayedHedgehog-2015_01_20-19_38_38",
}
```

Timestamps are important in forensics

Timestamps get stored in a lot of ways.

OSXCollector normalizes them.

```
{
  "file_path": "/System/Library/Extensions/Apple_iSight.kext/Contents/MacOS/Apple_iSight",
  "sha2": "19b7b85eaedb17d9565dce872f0d1ea8fc0761f508f28bedcc8606b828cbf614",
  "sha1": "99005b68295c202fd359b46cd1411acea96b2469",
  "md5": "b8cc164b6546e4b13768d8353820b216",
  "ctime": "2014-12-05 16:50:39",
  "mtime": "2014-09-19 00:16:50",
  "osxcollector_section": "kext",
  "osxcollector_incident_id": "DelayedHe[illegible]-2015_01_20-19_38_38",
  "osxcollector_plist_path": "/System/Library[illegible]nsions/Apple_iSight.kext/Contents/Info.plist",
  "osxcollector_bundle_id": "com.apple.driver.App[illegible]ght",
  "signature_chain": [
    "Software Signing",
    "Apple Code Signing Certification Authority",
    "Apple Root CA"
  ]
}
```

VERY NORMALIZED

Hashes are *still* important in forensics

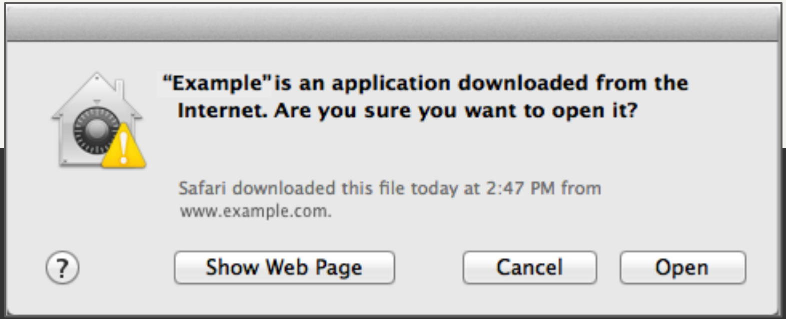
```
{
  "file_path": "/System/Library/Extensions/Apple_iSight.kext/Contents/MacOS/Apple_iSight",
  "sha2": "19b7b85eaedb17d9565dce872f0d1ea8fc0761f508f28bedcc8606b828cbf614",
  "sha1": "99005b68295c202fd359b46cd1411acea96b2469",
  "md5": "b8cc164b6546e4b13768d8353820b216",
  "ctime": "2014-12-05 16:50:39",
  "mtime": "2014-09-19 00:16:50",
  "osxcollector_section": "kext",
  "osxcollector_incident_id": "DelayedHedgehog-2015_01_20-1[illegible]3_38",
  "osxcollector_plist_path": "/System/Library/Extensions/Apple_iSight.kext/Contents/Info.plist",
  "osxcollector_bundle_id": "com.apple.driver.Apple_iSight",
  "signature_chain": [
    "Software Signing",
    "Apple Code Signing Certification Authority",
    "Apple Root CA"
  ]
}
```

STILL USEFUL

@jsendor

Quarantines track downloaded content

They live forever in a plist.



"Example"is an application downloaded from the Internet. Are you sure you want to open it?

Safari downloaded this file today at 2:47 PM from www.example.com.

```
{
  "osxcollector_section": "quarantines",
  "osxcollector_username": "jsendor",
  "LSQuarantineAgentName": "Google Chrome",
  "LSQuarantineAgentBundleIdentifier": "com.google.Chrome",
  "LSQuarantineDataURLString": "https://cachefly.alfredapp.com/Alfred_2.5.1_308.zip",
  "LSQuarantineEventIdentifier": "6FA87446-1249-4578-83E4-4BBCF7AEA4A3",
  "LSQuarantineOriginURLString": "http://www.alfredapp.com/",
  "osxcollector_db_path": "/Users/ivanlei/Library/Preferences/com.apple.LaunchServices.QuarantineEventsV2",
  "osxcollector_table_name": "LSQuarantineEvent",
  "osxcollector_incident_id": "DelayedHedgehog-2015_01_20-19_38_38",
  "LSQuarantineTimeStamp": "2014-12-05 14:40:33"
}
```

xattr-wherefrom

No need to search around in browser history.

```
{
  ..

  "md5": "0b984ecc39d5b33e4f6a81ade4e8dbf1",
  "xattr-quarantines": [
    "0001;5541127e;Google Chrome;63B2C485-1F64-4ADE-A95C-72F7087FA172"
  ],
  "signature_chain": [],
  "xattr-wherefrom": [
    "http://trojans.evildownloads.com/Trojan.app",
    "http://trojans.evildownloads.com/latest-trojans/"
  ],
  "osxcollector_incident_id": "DelayedHedgehog-2015_01_20-19_38_
  "file_path": "/Users/jdoe/Downloads/Trojan.app",
}
```

THIS IS BAAAD

OS X doesn't care if startups and kext are signed

But I kinda do, so OSXCollector lists the signature chain.

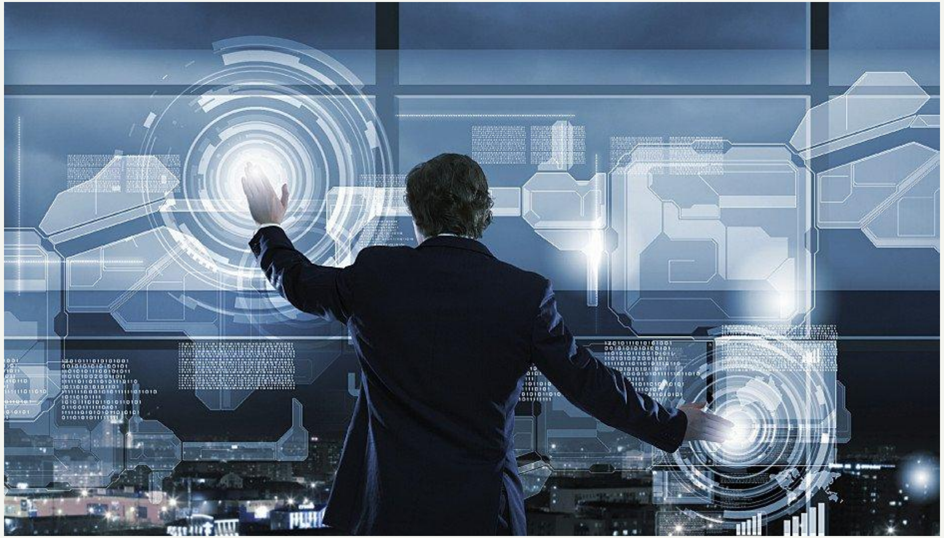
```
{
  "osxcollector_section": "startup",
  "osxcollector_subsection": "launch_agents",
  "md5": "dbd251d8a6e4da2419d75f5b18cf5078",
  "sha1": "bbb8016ad1026aea499fd47e21ffeb95f9597aca",
  "sha2": "9c89666fd071abd203f044ab7b3fd416decafe4468ff2e20a50b6d72f94809e2",
  "file_path": "/Library/Application Support/GPGTools/uuid-patcher",
  "ctime": "2014-12-05 16:52:00",
  "mtime": "2014-11-30 15:49:40",
  "osxcollector_plist": "/System/Library/LaunchDaemons/ssh.plist",
  "program": "/usr/libexec/sshd-keygen-wrapper",
  "label": "com.openssh.sshd",
  "signature_chain": [],
  "osxcollector_incident_[illegible]"DelayedHedgehog-2015_01_20-19_38_38",
}
```

SWELL!

Forensic collection is hard work.

Forensic analysis is fun.

Part science, part art.



Manual analysis with **grep** and **jq** works pretty well

grep a time window

```
$ cat foo.json | grep '2014-01-01 11:3[2-8]'
```

only urls in a time window

```
$ cat foo.json | grep '2014-01-01 11:3[2-8]' | jq 'select( has("url")).url'
```

grep a single user

```
$ cat INCIDENT32.json | jq 'select( .osxcollector_username=="jsendor")|.'
```

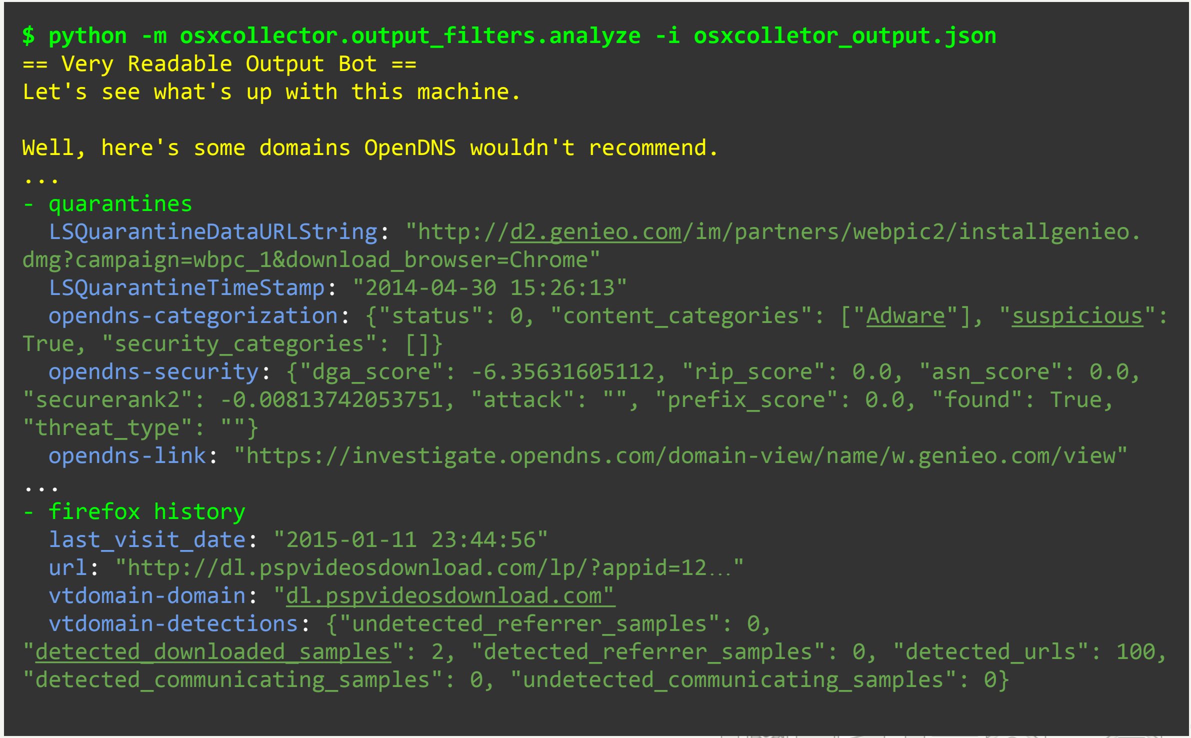
REUTERS

We can automate this!

step 1: analyze

step 2: ???

step 3: profit

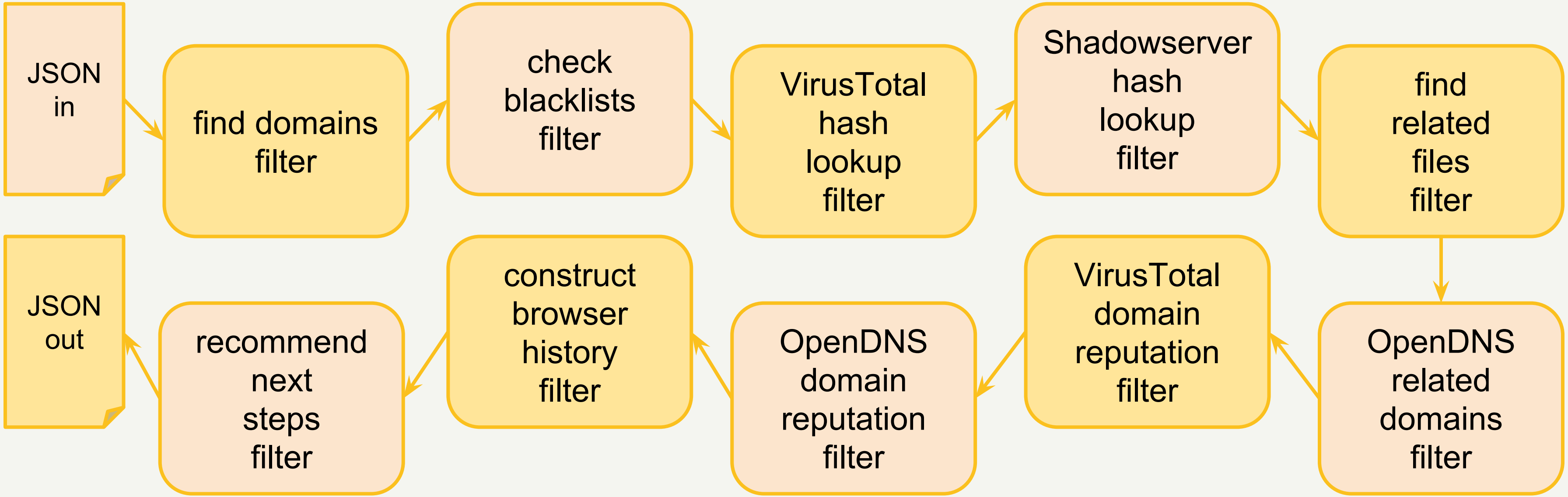


```
$ python -m osxcollector.output_filters.analyze -i osxcolletor_output.json
== Very Readable Output Bot ==
Let's see what's up with this machine.

Well, here's some domains OpenDNS wouldn't recommend.
...
- quarantines
  LSQuarantineDataURLString: "http://d2.genieo.com/im/partners/webpic2/installgenieo.
dmg?campaign=wbpc_1&download_browser=Chrome"
  LSQuarantineTimeStamp: "2014-04-30 15:26:13"
  opendns-categorization: {"status": 0, "content_categories": ["Adware"], "suspicious":
True, "security_categories": []}
  opendns-security: {"dga_score": -6.35631605112, "rip_score": 0.0, "asn_score": 0.0,
"securerank2": -0.00813742053751, "attack": "", "prefix_score": 0.0, "found": True,
"threat_type": ""}
  opendns-link: "https://investigate.opendns.com/domain-view/name/w.genieo.com/view"
...
- firefox history
  last_visit_date: "2015-01-11 23:44:56"
  url: "http://dl.pspvideosdownload.com/lp/?appid=12…"
  vtdomain-domain: "dl.pspvideosdownload.com"
  vtdomain-detections: {"undetected_referrer_samples": 0,
"detected_downloaded_samples": 2, "detected_referrer_samples": 0, "detected_urls": 100,
"detected_communicating_samples": 0, "undetected_communicating_samples": 0}
```

@jsendor

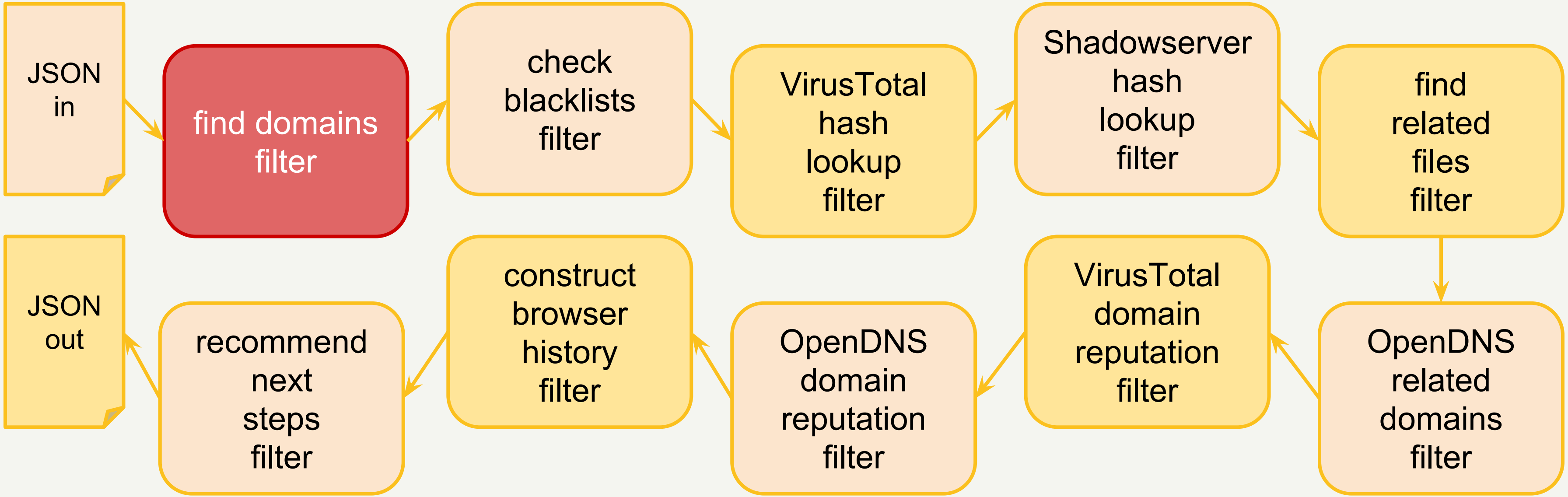
Enter OSXCollector Output Filters

Automated analysis with output filters


JSON in
find domains filter
check blacklists filter
VirusTotal hash lookup filter
Shadowserver hash lookup filter
find related files filter
OpenDNS related domains filter
VirusTotal domain reputation filter
OpenDNS domain reputation filter
construct browser history filter
recommend next steps filter
JSON out

@jsendor

Automated analysis with output filters


JSON in
find domains filter
check blacklists filter
VirusTotal hash lookup filter
Shadowserver hash lookup filter
find related files filter
OpenDNS related domains filter
VirusTotal domain reputation filter
OpenDNS domain reputation filter
construct browser history filter
recommend next steps filter
JSON out

@jsendor

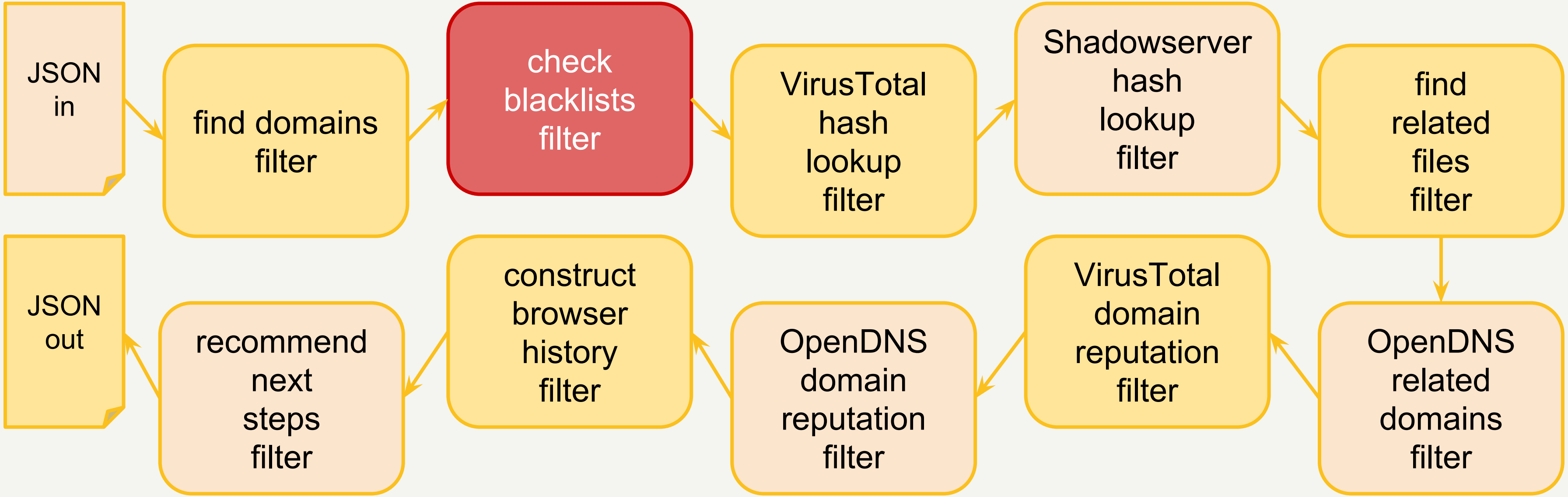
find domains filter

```
{
  "url": "https://biz.yelp.com"
}
```

```
{
  "url": "https://biz.yelp.com",
  "osxcollector_domains": [
    "biz.yelp.com",
    "yelp.com"
  ]
}
```

a lot of filters add a single piece of info

Automated analysis with output filters



JSON in
find domains filter
check blacklists filter
VirusTotal hash lookup filter
Shadowserver hash lookup filter
find related files filter
OpenDNS related domains filter
VirusTotal domain reputation filter
OpenDNS domain reputation filter
construct browser history filter
recommend next steps filter
JSON out

@jsendor

check blacklist filter

```
{
  "url": "https://www.evil.com",
  "osxcollector_domains": [
    "www.evil.com",
    "evil.com"
  ]
}
```

domain_blacklist.txt

evil.com
streaming-football.com
downloads.com

```
{
  "url": "https://www.evil.com",
  "osxcollector_domains": [
    "www.evil.com",
    "evil.com"
  ],
  "osxcollector_blacklist": [
    "domains"
  ]
}
```

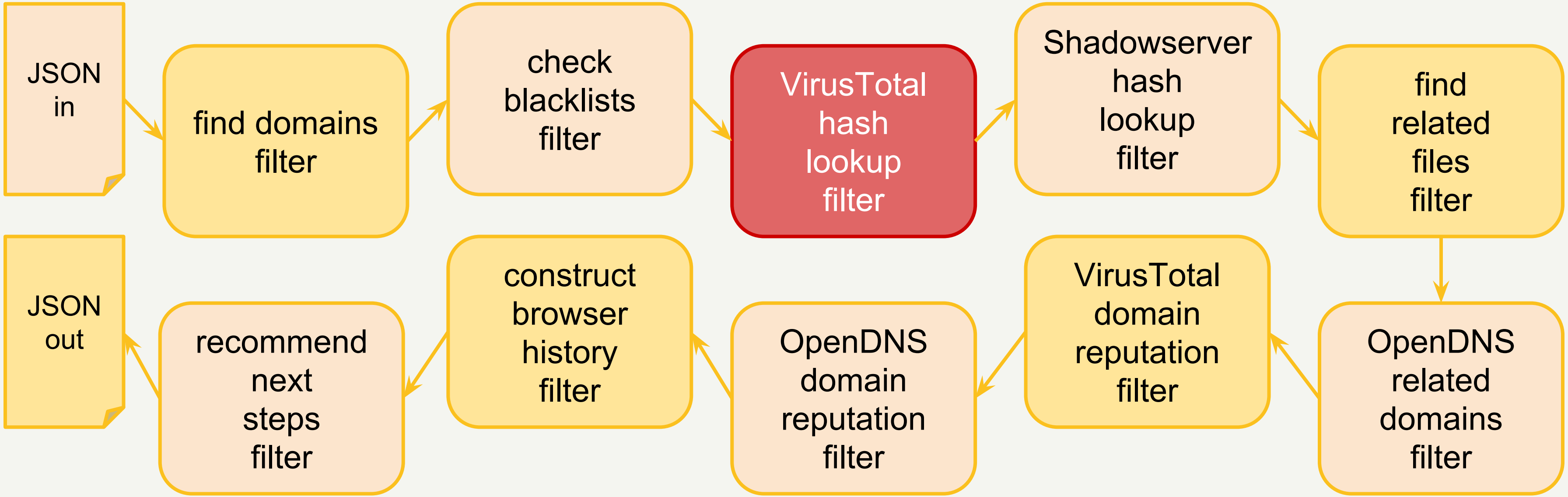
Match any key.

Regex or exact match.

Built in smarts for turning domains into regex.

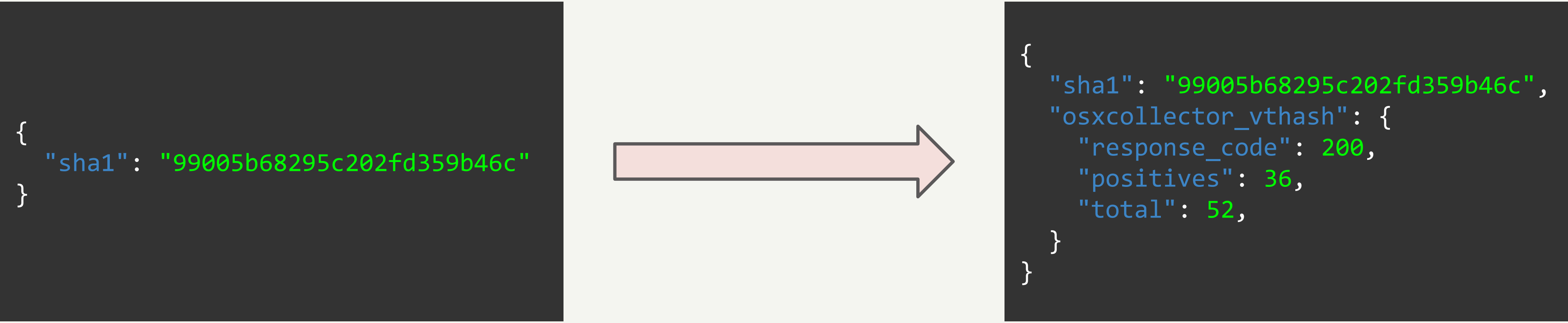
@jsendor

Automated analysis with output filters


JSON in
find domains filter
check blacklists filter
VirusTotal hash lookup filter
Shadowserver hash lookup filter
find related files filter
OpenDNS related domains filter
VirusTotal domain reputation filter
OpenDNS domain reputation filter
construct browser history filter
recommend next steps filter
JSON out

@jsendor

VirusTotal hash lookup filter



```
{
  "sha1": "99005b68295c202fd359b46c"
}
```

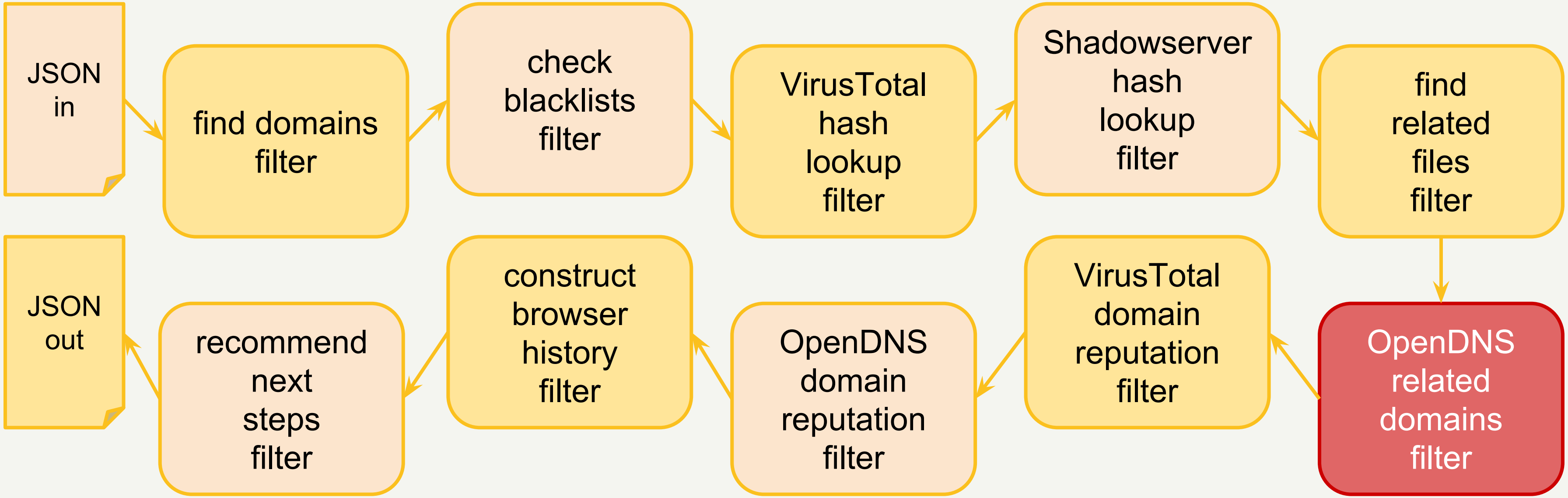
```
{
  "sha1": "99005b68295c202fd359b46c",
  "osxcollector_vthash": {
    "response_code": 200,
    "positives": 36,
    "total": 52,
  }
}
```

API output filter base does the heavy lifting.

Support for rate limits & response caching issues10s of requests at once.

@jsendor

Automated analysis with output filters



JSON in
find domains filter
check blacklists filter
VirusTotal hash lookup filter
Shadowserver hash lookup filter
find related files filter
OpenDNS related domains filter
VirusTotal domain reputation filter
OpenDNS domain reputation filter
construct browser history filter
recommend next steps filter
JSON out

@jsendor

OpenDNS related domains filter

```
{
  "url": "https://www.evil.com",
}
```

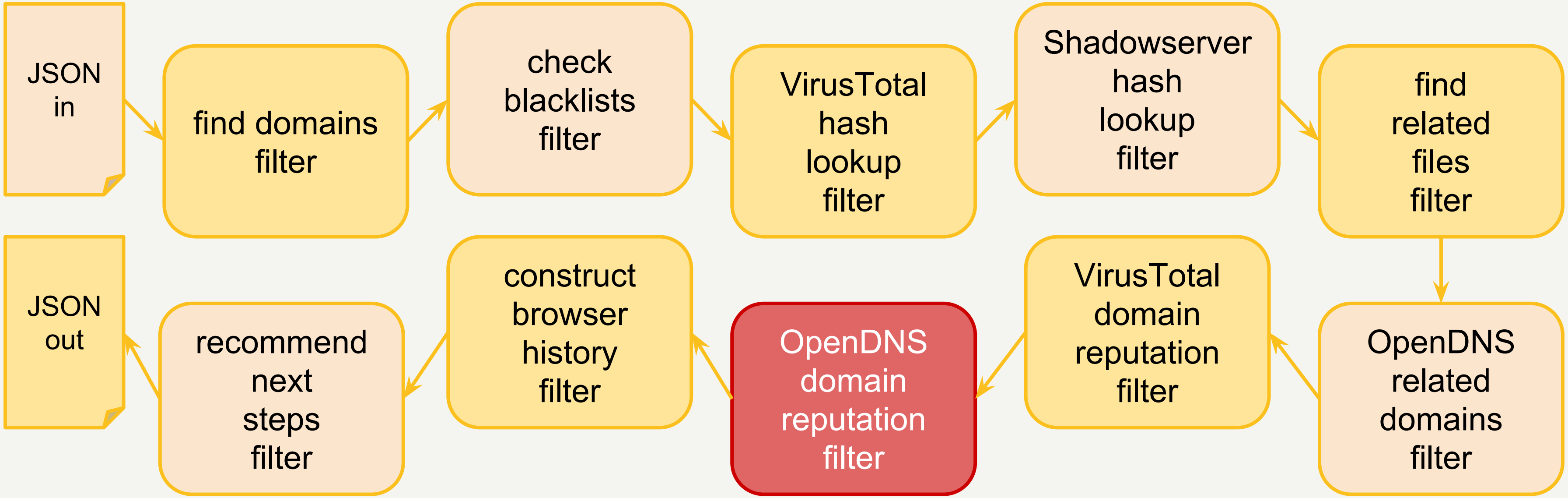
```
{
  "url": "https://www.evil.com",
  "osxcollector_related": {
    "domains": [
       "double-evil.com",
       "free-lunch.org",
       "torrent-malware.net"
    ]
  }
}
```

Judge domains by the company they keep.

Domains related to suspicious domains are usually suspicious themselves.

@jsendor

Automated analysis with output filters



JSON in

find domains filter

check blacklists filter

VirusTotal hash lookup filter

Shadowserver hash lookup filter

find related files filter

OpenDNS related domains filter

VirusTotal domain reputation filter

OpenDNS domain reputation filter

construct browser history filter

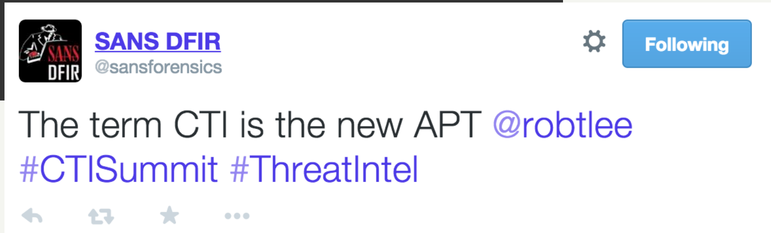
recommend next steps filter

JSON out

OpenDNS domain reputation filter

Premium Cyber Threat Intel (CTI)

```
{
  "url": "https://www.evil.com",
}
```

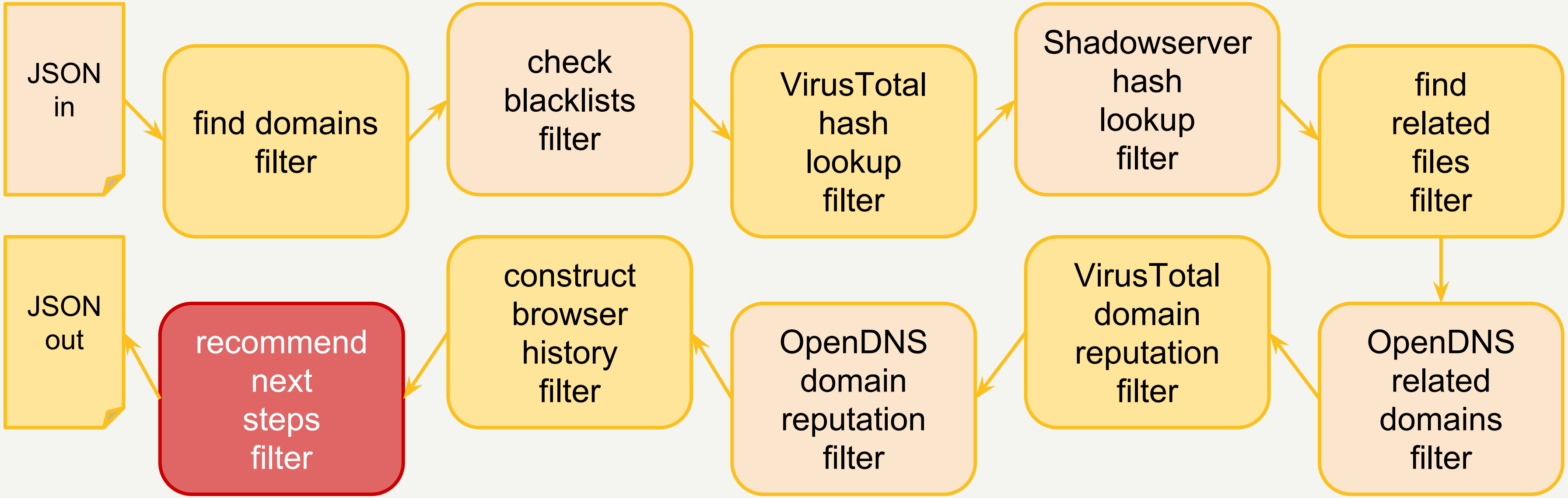


SANS DFIR
@sansforensics

The term CTI is the new APT @robtlee #CTISummit #ThreatIntel

```
{
  "url": "https://www.evil.com",
  "osxcollector_opendns": {
    "domain": "evil.com",
    },
    "security": {
      "found": true,
      "dga_score": -3,
      "securerank2": -23,
      "asn_score": -57,
      "prefix_score": -62,
      "rip_score": -99,
    }
  }
}
```

Automated analysis with output filters



JSON in

find domains filter

check blacklists filter

VirusTotal hash lookup filter

Shadowserver hash lookup filter

find related files filter

OpenDNS related domains filter

VirusTotal domain reputation filter

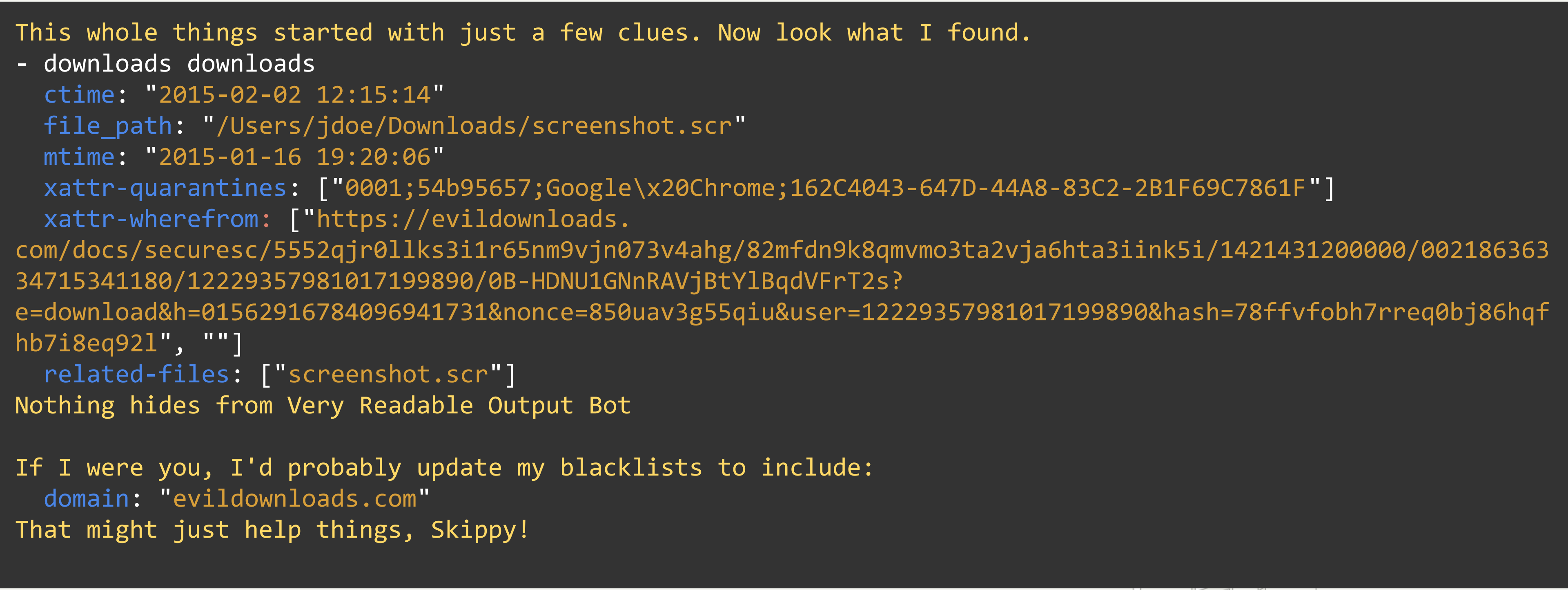
OpenDNS domain reputation filter

construct browser history filter

recommend next steps filter

JSON out

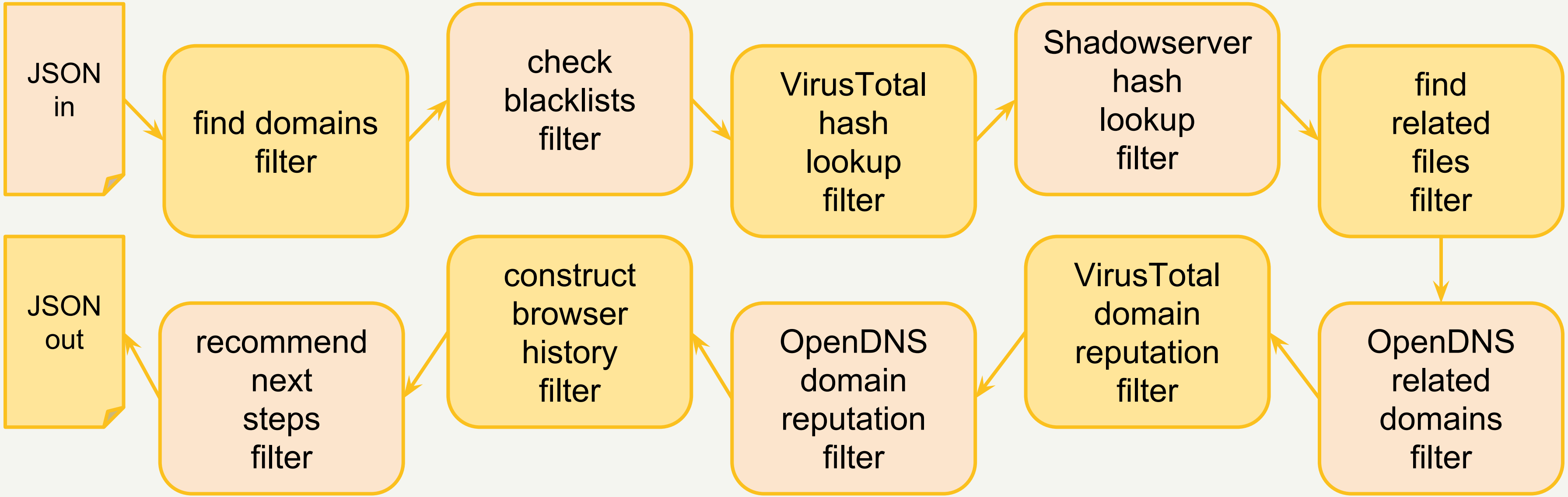
Recommend next steps



```
This whole things started with just a few clues. Now look what I found.
- downloads downloads
  ctime: "2015-02-02 12:15:14"
  file_path: "/Users/jdoe/Downloads/screenshot.scr"
  mtime: "2015-01-16 19:20:06"
  xattr-quarantines: ["0001;54b95657;Google\x20Chrome;162C4043-647D-44A8-83C2-2B1F69C7861F"]
  xattr-wherefrom: ["https://evildownloads.
com/docs/securesc/5552qjr0llks3i1r65nm9vjn073v4ahg/82mfdn9k8qmvmo3ta2vja6hta3iink5i/1421431200000/002186363
34715341180/12229357981017199890/0B-HDNU1GNnRAVjBtYlBqdVFrT2s?
e=download&h=01562916784096941731&nonce=850uav3g55qiu&user=12229357981017199890&hash=78ffvfobh7rreq0bj86hqf
hb7i8eq92l", ""]
  related-files: ["screenshot.scr"]
Nothing hides from Very Readable Output Bot

If I were you, I'd probably update my blacklists to include:
  domain: "evildownloads.com"
That might just help things, Skippy!
```

@jsendor

Automated analysis with output filters



JSON in

find domains filter

check blacklists filter

VirusTotal hash lookup filter

Shadowserver hash lookup filter

find related files filter

OpenDNS related domains filter

VirusTotal domain reputation filter

OpenDNS domain reputation filter

construct browser history filter

recommend next steps filter

JSON out

Threat Intel API

`https://github.com/Yelp/threat_intel`

Query Threat Intel Feeds:

virustotal

OpenDNS Investigate

shadowserver

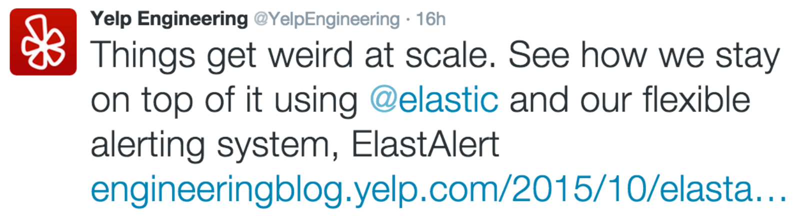
Call OpenDNS API endpoints

```
from threat_intel.opendns import InvestigateApi
investigate = InvestigateApi(<INVESTIGATE-API-KEY-HERE>, cache_file_name="/tmp/cache.opendns.json")

domains = ["google.com", "baidu.com", "bibikun.ru"]
investigate.security(domains)
```

```
{
  "baidu.com": {
    "found": true,
    "dga_score": 0,
    "rip_score": 0,

    ..

  }
}
```

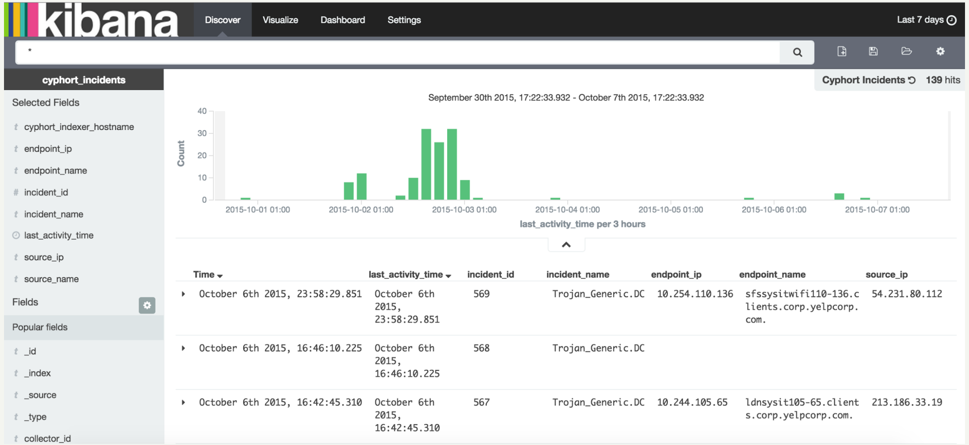
ElastAlert

http://engineeringblog.yelp.com/



Yelp Engineering @YelpEngineering · 16h

Things get weird at scale. See how we stay on top of it using @elastic and our flexible alerting system, ElastAlert engineeringblog.yelp.com/2015/10/elasta…



Time	last_activity_time	incident_id	incident_name	endpoint_ip	endpoint_name	source_ip
October 6th 2015, 23:58:29.851	October 6th 2015, 23:58:29.851	569	Trojan_Generic.DC	10.254.110.136	sfssysitwifi110-136.clients.corp.yelpcorp.com.	54.231.80.112
October 6th 2015, 16:46:10.225	October 6th 2015, 16:46:10.225	568	Trojan_Generic.DC			
October 6th 2015, 16:42:45.310	October 6th 2015, 16:42:45.310	567	Trojan_Generic.DC	10.244.105.65	ldnsysit105-65.clients.corp.yelpcorp.com.	213.186.33.19

@jsendor

https://github.com/Yelp/osxcollector

Lemme know if you use it.

Send pull requests.

Questions? Let's talk!

kuba@yelp.com

@jsendor

We are hiring! visit **yelp.com/careers**