what is osquery?

Explore your operating system using SQL

Host visibility motivated by intrusion detection

100% OS API usage, no *fork execve*

Facebook’s host intrusion detection detection agent

• [https://github.com/facebook/osquery](https://github.com/facebook/osquery)
• [https://osquery.io](https://osquery.io)
• [https://osquery.readthedocs.org](https://osquery.readthedocs.org)
why SQL?

SELECT pid, name, uid FROM processes

OS concepts are shared on Mac, Linux, and Windows

the “concepts” have attributes:
user ids, process ids, descriptors, ports, paths

most developers and administrators know SQL
why SQL?

SELECT pid, name, uid FROM processes
why SQL?

SELECT pid, name, uid FROM processes
why SQL?

SELECT pid, name, uid FROM processes

WHERE uid != 0

[constraints]
why SQL?

SELECT pid, name, username FROM processes
JOIN users ON processes.uid = users.uid
WHERE uid != 0
download and install osquery: https://osquery.io/downloads

OS X 10.9, 10.10, 10.11
CentOS 6.6 or 7.1
Ubuntu 12.04 or 14.04

if you do not have access to any locally, let us know

```bash
~ » uname -a
Linux win8-vm 3.13.0-24-generic #47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014 x86_64 x86_64 x86_64 GNU/
Linux

~ » dpkg --info ./Downloads/osquery-1.5.3.deb
new debian package, version 2.0.
size 4858170 bytes: control archive=749 bytes.
  397 bytes,  12 lines     control
  594 bytes,   9 lines     md5sums
Package: osquery
Version: 1.5.3-1.ubuntu14
License: BSD
Vendor: Facebook
Architecture: amd64
Maintainer: osquery@osquery.io
Installed-Size: 13843
Depends: zlib1g, libbz2-1.0, libreadline6, libgcrypt11, libc6 (>=2.15), libapt-pkg4.12, libstdc++6
(>= 4.8), libudev1
Section: default
Priority: extra
Homepage: https://osquery.io
Description: osquery is an operating system instrumentation toolchain.

~ » sudo dpkg -i ./Downloads/osquery-1.5.3.deb
[sudo] password for reed:
(Reading database ... 141943 files and directories currently installed.)
Preparing to unpack ./Downloads/osquery-1.5.3.deb ... Unpacking osquery (1.5.3-1.ubuntu14) over (1.5.2-73-g709479b-1.ubuntu14) ...
Setting up osquery (1.5.3-1.ubuntu14) ...
```
uname -a
Linux localhost.localdomain 3.10.0-123.20.1.el7.x86_64 #1 SMP Thu Jan 29 18:05:33 UTC 2015 x86_64 x86_64 GNU/Linux

rpm -qip /osquery-1.5.3.rpm
warning: /osquery-1.5.3.rpm: Header V4 RSA/SHA1 Signature, key ID c9d8b80b: NOKEY
Name : osquery
Version : 1.5.3
Release : 1.el7
Architecture: x86_64
Install Date: (not installed)
Group : default
Size : 13755286
License : BSD
Signature : RSA/SHA1, Mon 28 Sep 2015 09:09:35 PM PDT, Key ID 97a80c63c9db80b
Source RPM : osquery-1.5.3-1.el7.src.rpm
Build Date : Mon 28 Sep 2015 07:58:56 PM PDT
Build Host : centos7
Relocations : /
Packager : osquery@osquery.io
Vendor : Facebook
URL : https://osquery.io
Summary : osquery is an operating system instrumentation toolchain.
Description : osquery is an operating system instrumentation toolchain.

sudo rpm --install /osquery-1.5.3.rpm
warning: /osquery-1.5.3.rpm: Header V4 RSA/SHA1 Signature, key ID c9d8b80b: NOKEY
Select the disk where you want to install the osquery-1.5.3 software.

Macintosh HD
31.64 GB available
249.67 GB total

Installing this software requires 13.2 MB of space.
run `osqueryi` and inspect the basic shell help menu
also use `.schema listening_ports`
see docs at https://osquery.io/docs/tables/
see docs at https://osquery.io/docs/tables/#file
osquery --line "select * from hash where path = '/System/Library/CoreServices/boot.efi'"
path = /System/Library/CoreServices/boot.efi
directory = /System/Library/CoreServices
  md5 = 4ce50c4492f1ef0981c782f3068f1c15
  sha1 = 8c3a92403db3bdcf25460a29ef0b42f8baac1b70
  sha256 = c4cf0a45b00c1a496c1ed9bbbc3d734f6e4e07a768eab0df33c8ef47dcccfc989

osquery --json "select * from hash where path = '/System/Library/CoreServices/boot.efi'"
[
  {
    "directory": "/System/Library/CoreServices",
    "md5": "4ce50c4492f1ef0981c782f3068f1c15",
    "path": "/System/Library/CoreServices/boot.efi",
    "sha1": "8c3a92403db3bdcf25460a29ef0b42f8baac1b70",
    "sha256": "c4cf0a45b00c1a496c1ed9bbbc3d734f6e4e07a768eab0df33c8ef47dcccfc989"
  }
]

osquery "select * from kernel_info"
+

| version | arguments | path | device |
+---------+-----------+------+--------|
|         | md5       |      |        |
|         |           |      |        |
|         |           |      |        |
| 14.5.0  |           |      |        |
|         |           |      |        |
|         |           |      |        |
|         |           |      |        |

osquery --line -A kernel_info

version = 14.5.0
arguments =
  path = System/Library/Caches/com.apple.kext.caches/Startup/kernelcache
  device = E220890C-BE93-42CF-8F56-D9E64B5E7820
  md5 =
see docs at https://osquery.io/docs/tables/#processes

<table>
<thead>
<tr>
<th>name</th>
<th>port</th>
<th>address</th>
<th>protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserEventAgent</td>
<td>0</td>
<td>0.0.0.0</td>
<td>17</td>
</tr>
<tr>
<td>UserEventAgent</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>ARDAGroup</td>
<td>3283</td>
<td>0.0.0.0</td>
<td>17</td>
</tr>
<tr>
<td>ARDAGroup</td>
<td>3283</td>
<td>0.0.0.0</td>
<td>17</td>
</tr>
<tr>
<td>SystemUIserver</td>
<td>49623</td>
<td>0.0.0.0</td>
<td>17</td>
</tr>
<tr>
<td>zixi_video_acceleration_proxy-16617</td>
<td>4500</td>
<td>127.0.0.1</td>
<td>6</td>
</tr>
<tr>
<td>ZBUA8C452C.com.agilebits.onepassword4-helper</td>
<td>6258</td>
<td>127.0.0.1</td>
<td>6</td>
</tr>
<tr>
<td>ZBUA8C452C.com.agilebits.onepassword4-helper</td>
<td>6258</td>
<td>::1</td>
<td>6</td>
</tr>
<tr>
<td>ZBUA8C452C.com.agilebits.onepassword4-helper</td>
<td>6263</td>
<td>127.0.0.1</td>
<td>6</td>
</tr>
<tr>
<td>ZBUA8C452C.com.agilebits.onepassword4-helper</td>
<td>6263</td>
<td>::1</td>
<td>6</td>
</tr>
<tr>
<td>Dropbox</td>
<td>17500</td>
<td>0.0.0.0</td>
<td>17</td>
</tr>
<tr>
<td>Dropbox</td>
<td>17500</td>
<td>0.0.0.0</td>
<td>17</td>
</tr>
<tr>
<td>Dropbox</td>
<td>17600</td>
<td>127.0.0.1</td>
<td>6</td>
</tr>
<tr>
<td>Dropbox</td>
<td>17603</td>
<td>127.0.0.1</td>
<td>6</td>
</tr>
<tr>
<td>VBoxHeadless</td>
<td>2222</td>
<td>127.0.0.1</td>
<td>6</td>
</tr>
<tr>
<td>BetterTouchTool</td>
<td>60737</td>
<td>0.0.0.0</td>
<td>6</td>
</tr>
<tr>
<td>BetterTouchTool</td>
<td>60737</td>
<td>0.0.0.0</td>
<td>6</td>
</tr>
<tr>
<td>BetterTouchTool</td>
<td>60737</td>
<td>0.0.0.0</td>
<td>17</td>
</tr>
<tr>
<td>BetterTouchTool</td>
<td>60737</td>
<td>0.0.0.0</td>
<td>17</td>
</tr>
</tbody>
</table>
The most value comes from the **osqueryd** daemon. This uses a JSON-config to set options and define a schedule.

```json
{
   "options": {
      "host_identifier": "hostname",
      "logger_path": "/tmp"
   },
   "schedule": {
      "usb_devices": {
         "query": "SELECT * FROM usb_devices",
         "interval": 10
      }
   }
}
```

**Log (single line):**

```json
{
   "name": "usb_devices",
   "hostIdentifier": "reed-mbp.local",
   "unixTime": "1444120356",
   "columns": {
      "model": "USB Laser Mouse",
      "model_id": "c069",
      "vendor": "Logitech",
      "vendor_id": "046d"
   },
   "action": "added"
}
```
The most value comes from the osqueryd daemon. This uses a JSON-config to set options and define a schedule.

The schedule is a set of QUERY and INTERVAL pairs. The logs are changes in the output of the queries.

These queries can be organized into packs, and distributed alongside the osquery package or internally.
Now write a small config to `/tmp/config.json`.

When starting a “standalone” `osqueryd` we need to change several options.
sudo tail -n 5 osqueryd.results.log
{"name": "usb_devices", "hostIdentifier": "reed-mbp.local", "calendarTime": "Tue Oct 6 08:32:36 2015 UTC", "unixTime": "1444120356", "columns": {"model": "Yubico Yubikey II", "model_id": "0010", "removable": "1", "serial": "0", "usb_address": "1", "usb_port": "2", "vendor": "Yubico", "vendor_id": "1050"}, "action": "added"}
{"name": "usb_devices", "hostIdentifier": "reed-mbp.local", "calendarTime": "Tue Oct 6 08:32:36 2015 UTC", "unixTime": "1444120356", "columns": {"model": "Internal Memory Card Reader", "model_id": "8406", "removable": "0", "serial": "0000000000820", "usb_address": "2", "usb_port": "3", "vendor": "Apple", "vendor_id": "05ac"}, "action": "added"}
{"name": "usb_devices", "hostIdentifier": "reed-mbp.local", "calendarTime": "Tue Oct 6 08:32:36 2015 UTC", "unixTime": "1444120356", "columns": {"model": "Apple Internal Keyboard ⌘ Trackpad", "model_id": "0259", "removable": "0", "serial": "0", "usb_address": "3", "usb_port": "5", "vendor": "Apple Inc.", "vendor_id": "05ac"}, "action": "added"}
{"name": "usb_devices", "hostIdentifier": "reed-mbp.local", "calendarTime": "Tue Oct 6 08:32:36 2015 UTC", "unixTime": "1444120356", "columns": {"model": "Bluetooth USB Host Controller", "model_id": "8289", "removable": "0", "serial": "0", "usb_address": "7", "usb_port": "3", "vendor": "Apple Inc.", "vendor_id": "05ac"}, "action": "added"}
{"name": "usb_devices", "hostIdentifier": "reed-mbp.local", "calendarTime": "Tue Oct 6 08:32:36 2015 UTC", "unixTime": "1444120356", "columns": {"model": "USB Laser Mouse", "model_id": "0069", "removable": "1", "serial": "0", "usb_address": "29", "usb_port": "1", "vendor": "Logitech", "vendor_id": "046d"}, "action": "added"}
On OS X use /var/osquery/osquery.example.conf

```
sudo cp /var/osquery/com.facebook.osqueryd.plist /Library/LaunchDaemons
```

```bash
sudo launchctl load /Library/LaunchDaemons/com.facebook.osqueryd.plist
```
What can you do with all the logs?

osquery + logstash forwarder + ELK

ELK
- Elastic Search
- Logstash
- Kibana
client configuration
logstash forwarder

```json
logstash-forwarder.conf
{
    "network": {
        "servers": [ "LOGSTASH_SERVER_IP:LOGSTASH_SERVER_PORT" ],
        "ssl ca": "/path/to/logstash-forwarder.crt",
        "timeout": 15
    },
    "files": [
        { "paths": [ "/var/log/osquery/osqueryd.results.log" ],
        "fields": { "type": "osquery_json" }
        }
    ]
}
```
server configuration

logstash

01-lumberjack-input.conf:

```bash
input {
  lumberjack {
    port => 5000
    type => "logs"
    ssl_certificate => "/path/to/file.crt"
    ssl_key => "/path/tofile.key"
    codec => "json"
  }
}
```

10-osquery.conf

```bash
filter {
  if [type] == "osquery_json" {
    json {
      source => "message"
    }
    date {
      match => [ "unixTime", "UNIX" ]
    }
  }
}
```
installing ELK

https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-4-on-ubuntu-14-04

https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-4-on-centos-7
configuration docs

All osquery docs kept in the Github repo and hosted using RTD


https://github.com/facebook/osquery/tree/master/docs/wiki
AWS lab

Log into an AWS node:

lab-centos7-1  lab-ubuntu14-1
lab-centos7-2  lab-ubuntu14-2
lab-centos7-3  lab-ubuntu14-3  osquery.io
lab-centos7-4  lab-ubuntu14-4
lab-centos7-5  lab-ubuntu14-5

Ubuntu14 machines username is ubuntu
CentOS7 machines username is centos

User passwords are handed out in the workshop
AWS lab

Verify that `osqueryd` is running
Inspect the config: `/etc/osquery/osquery.conf`

Use Kibana to detect your actions and try to find the Azazel and a host with a simple rootkit

https://lab.osquery.io

only available during the workshop
all development happens in the open, on GitHub

the problem that osquery solves isn’t unique to Facebook

- https://github.com/facebook/osquery
- https://osquery.io
- https://osquery.readthedocs.org

➡ @osquery
➡ @teddyreedv
➡ @javutin