BruCON 2015 // osquery workshop

Javier Marcos / Facebook
Ted Reed / Facebook

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 agent

- https://github.com/facebook/osquery
- https://osquery.io
- 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?

[concept]

```
SELECT pid, name, uid FROM processes
```

why SQL?

[attributes] [concept]

```
SELECT pid, name, uid FROM processes
```

why SQL?

```
SELECT pid, name, uid FROM processes

WHERE uid != 0
    [constraints]
```

why SQL?

[attribute]

```
SELECT pid, name, username FROM processes
JOIN users ON processes.uid=users.uid
WHERE uid != 0
```

[join]

download and install osquery: https://osquery.io/downloads

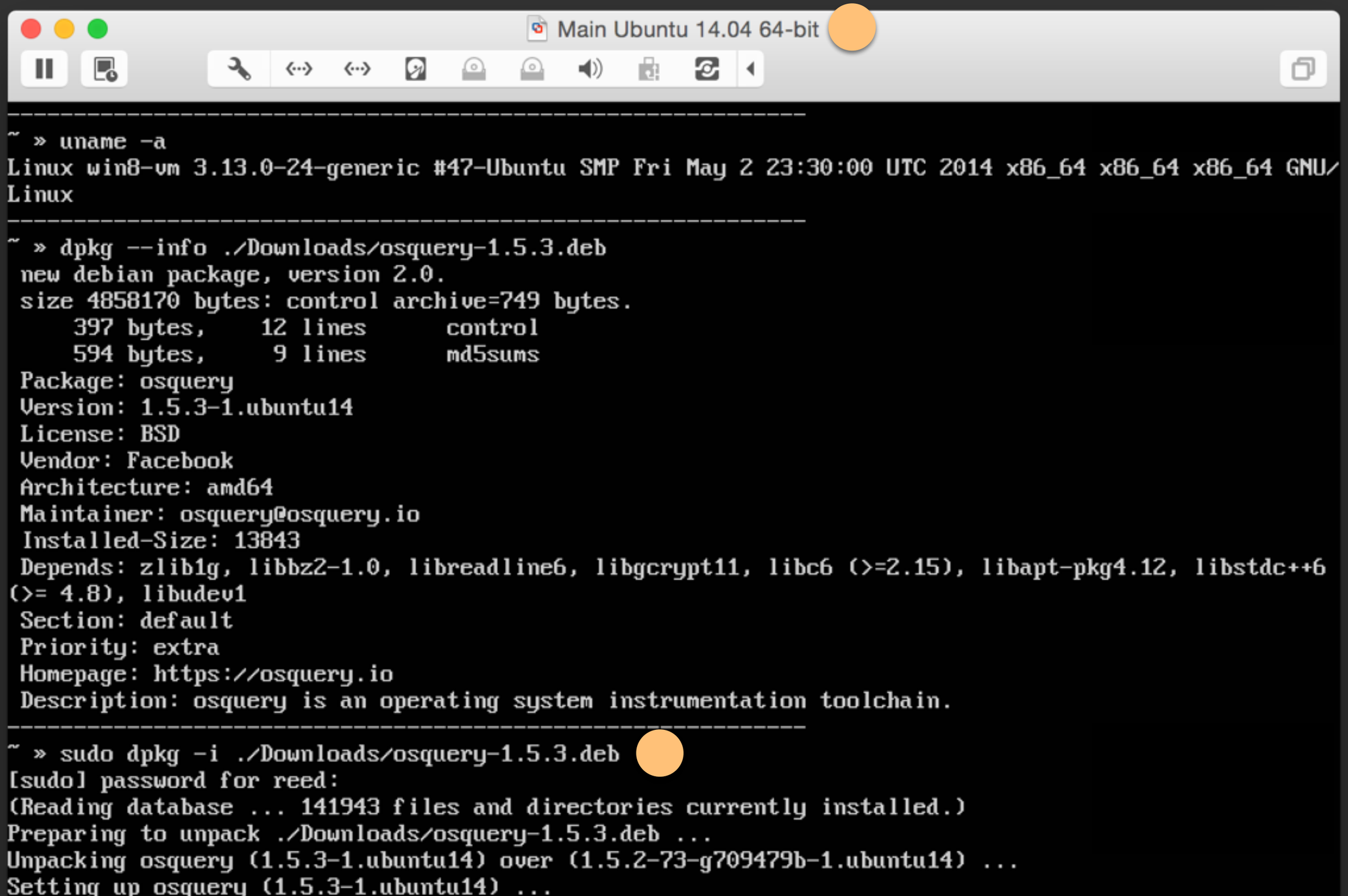
OS X 10.9, 10.10, 10.11

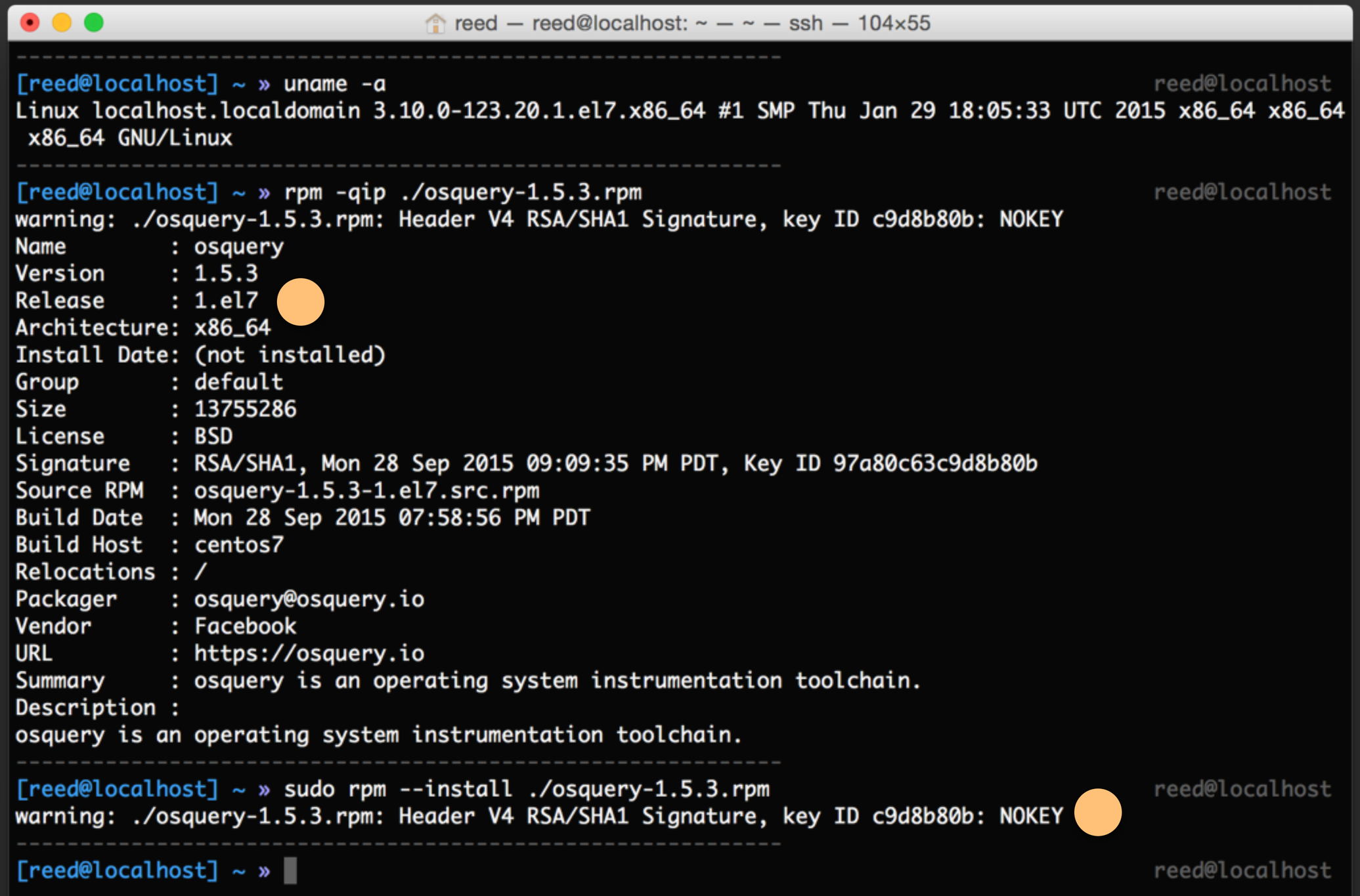
CentOS 6.6 or 7.1

Ubuntu 12.04 or 14.04

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

Start a Vagrant (Ubuntu 14.04): https://goo.gl/D2Owus


Main Ubuntu 14.04 64-bit
~ » 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) ...


reed — reed@localhost: ~ — ~ — ssh — 104×55
[reed@localhost] ~ » uname -a
reed@localhost
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
x86_64 GNU/Linux
[reed@localhost] ~ » rpm -qip ./osquery-1.5.3.rpm
reed@localhost
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 97a80c63c9d8b80b
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.
[reed@localhost] ~ » sudo rpm --install ./osquery-1.5.3.rpm
reed@localhost
warning: ./osquery-1.5.3.rpm: Header V4 RSA/SHA1 Signature, key ID c9d8b80b: NOKEY
[reed@localhost] ~ »
reed@localhost

Install osquery-1.5.3

Select a Destination

- Introduction
- **Destination Select**
- Installation Type
- Installation
- Summary

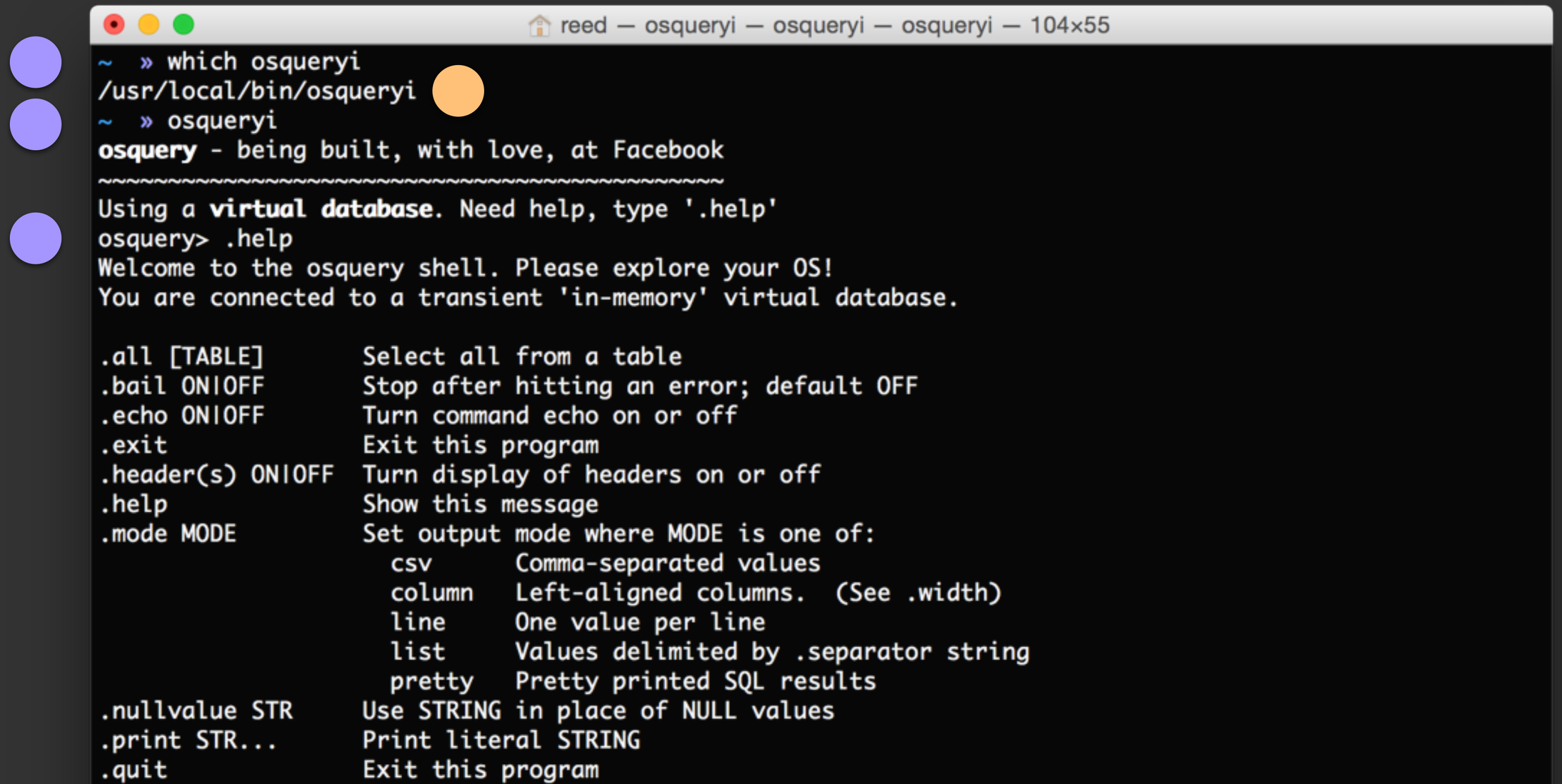
Select the disk where you want to install the osquery-1.5.3 software.

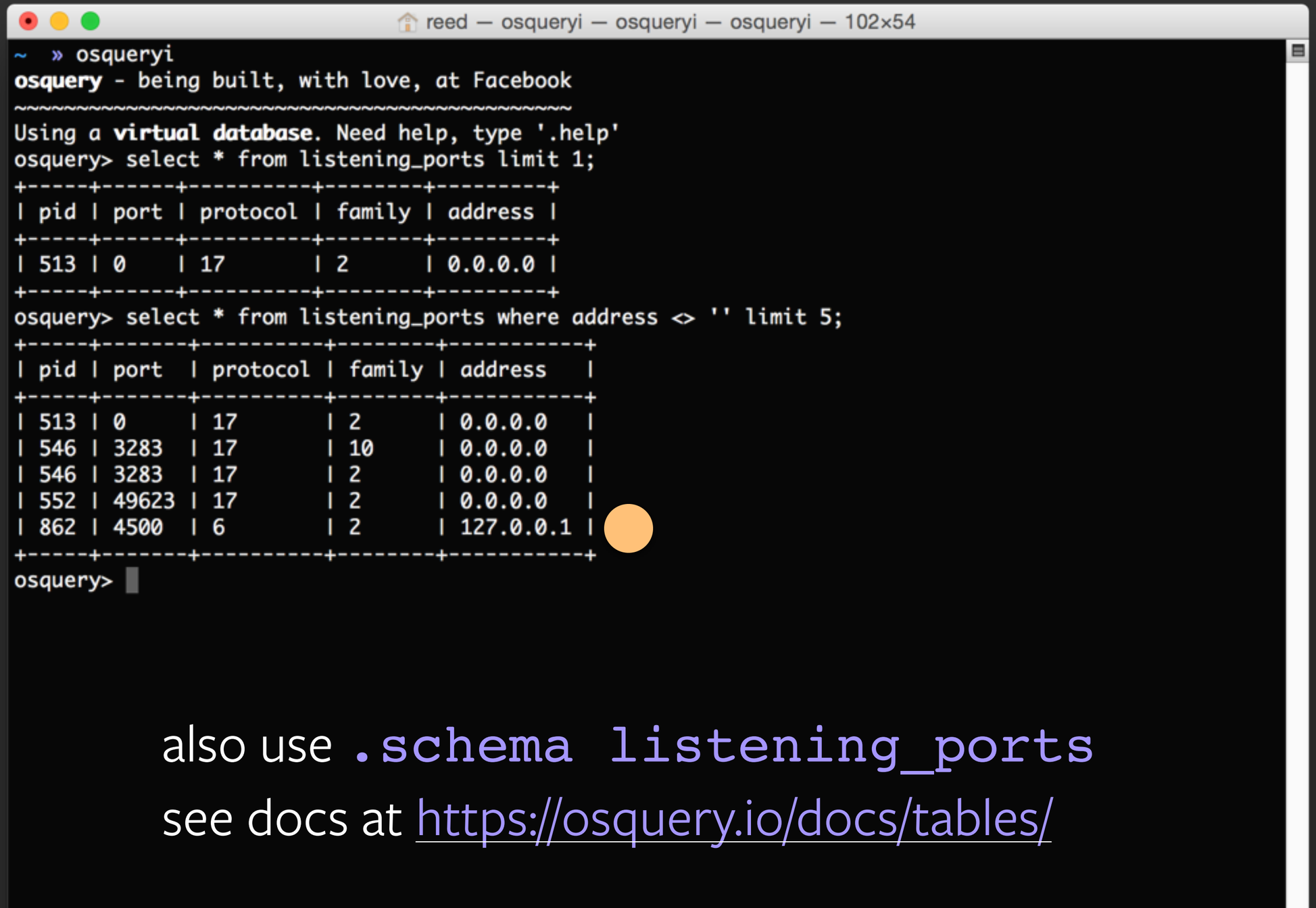
Macintosh HD
31.64 GB available
249.67 GB total

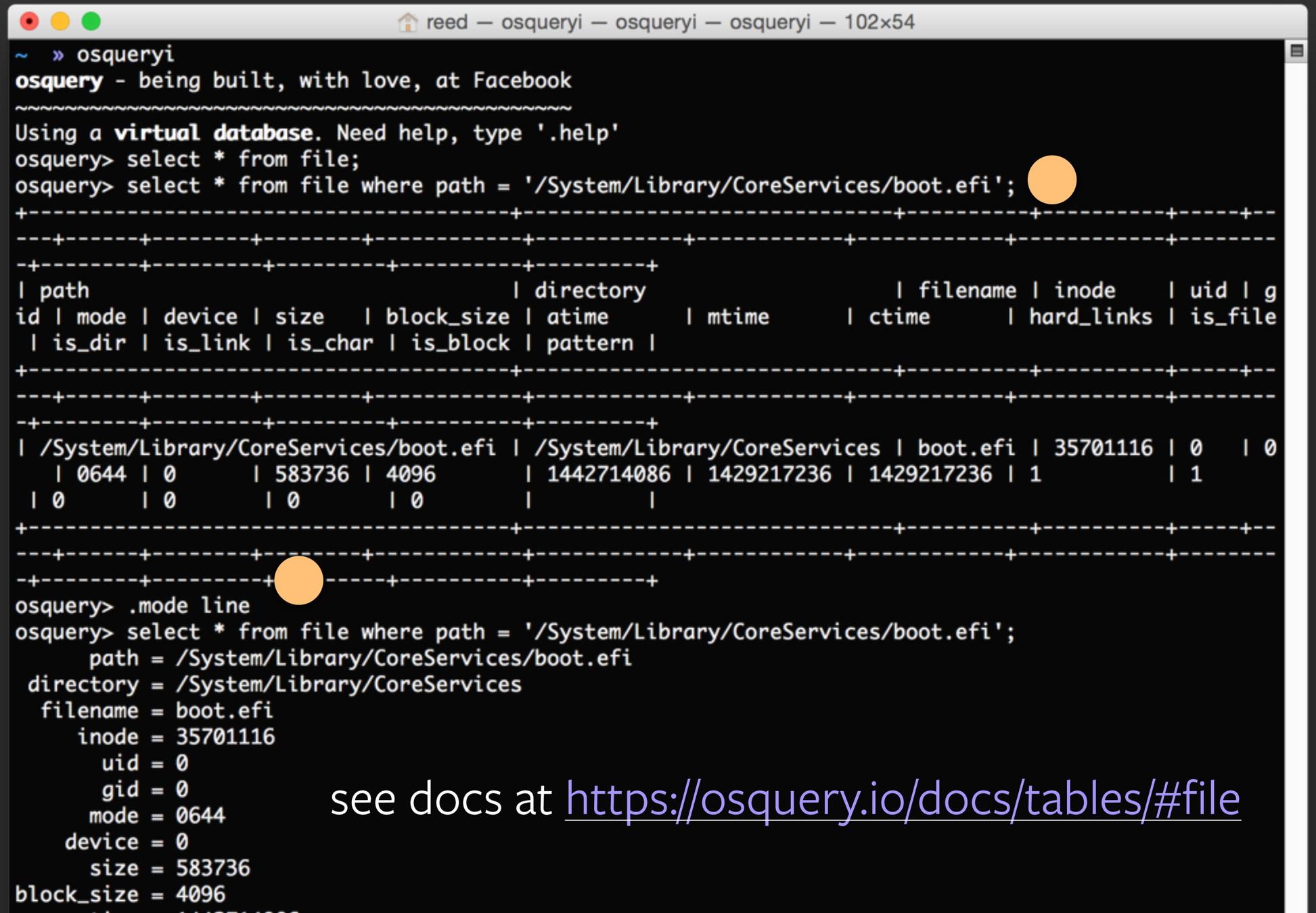
Installing this software requires 13.2 MB of space.

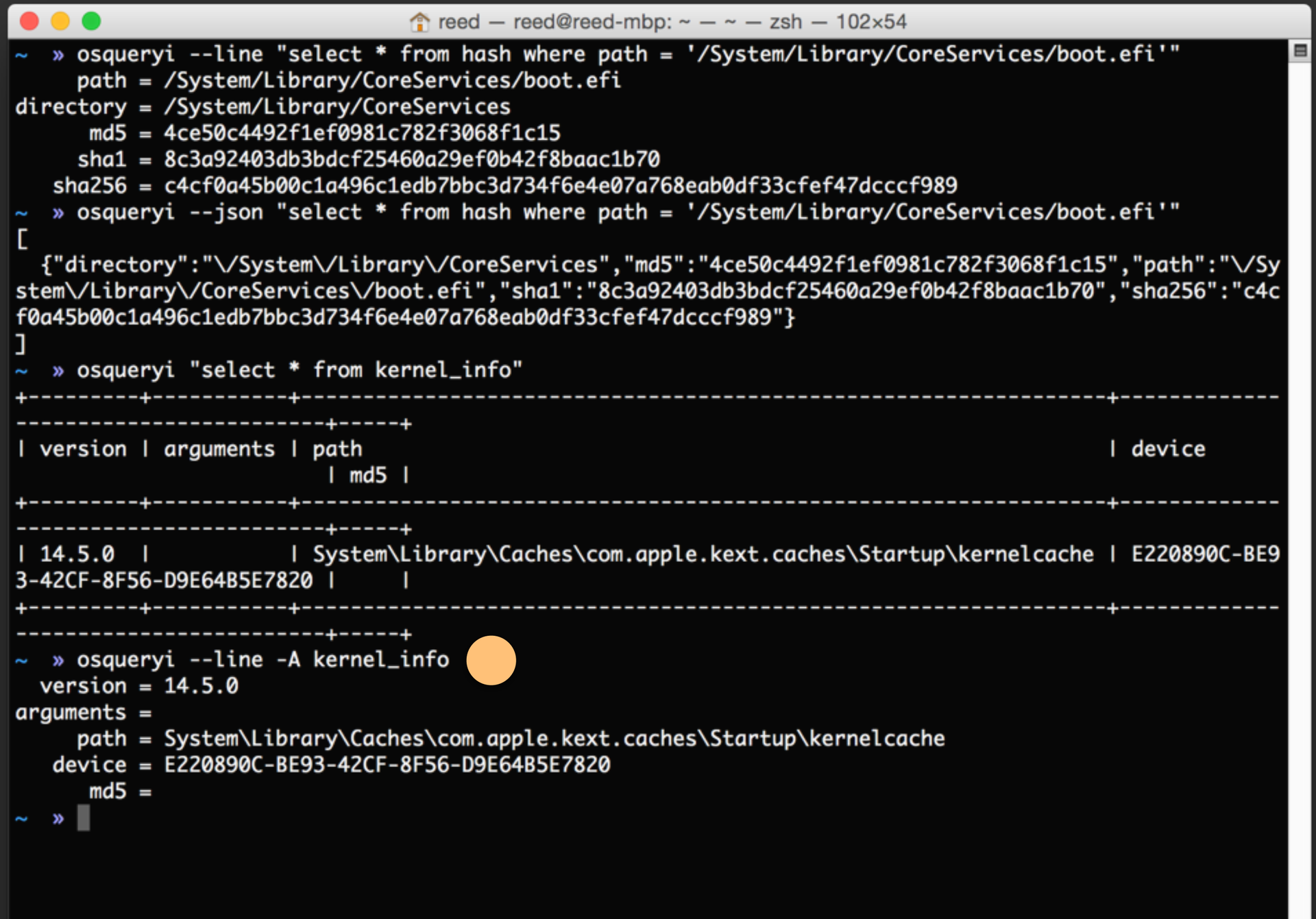
Go Back | Continue

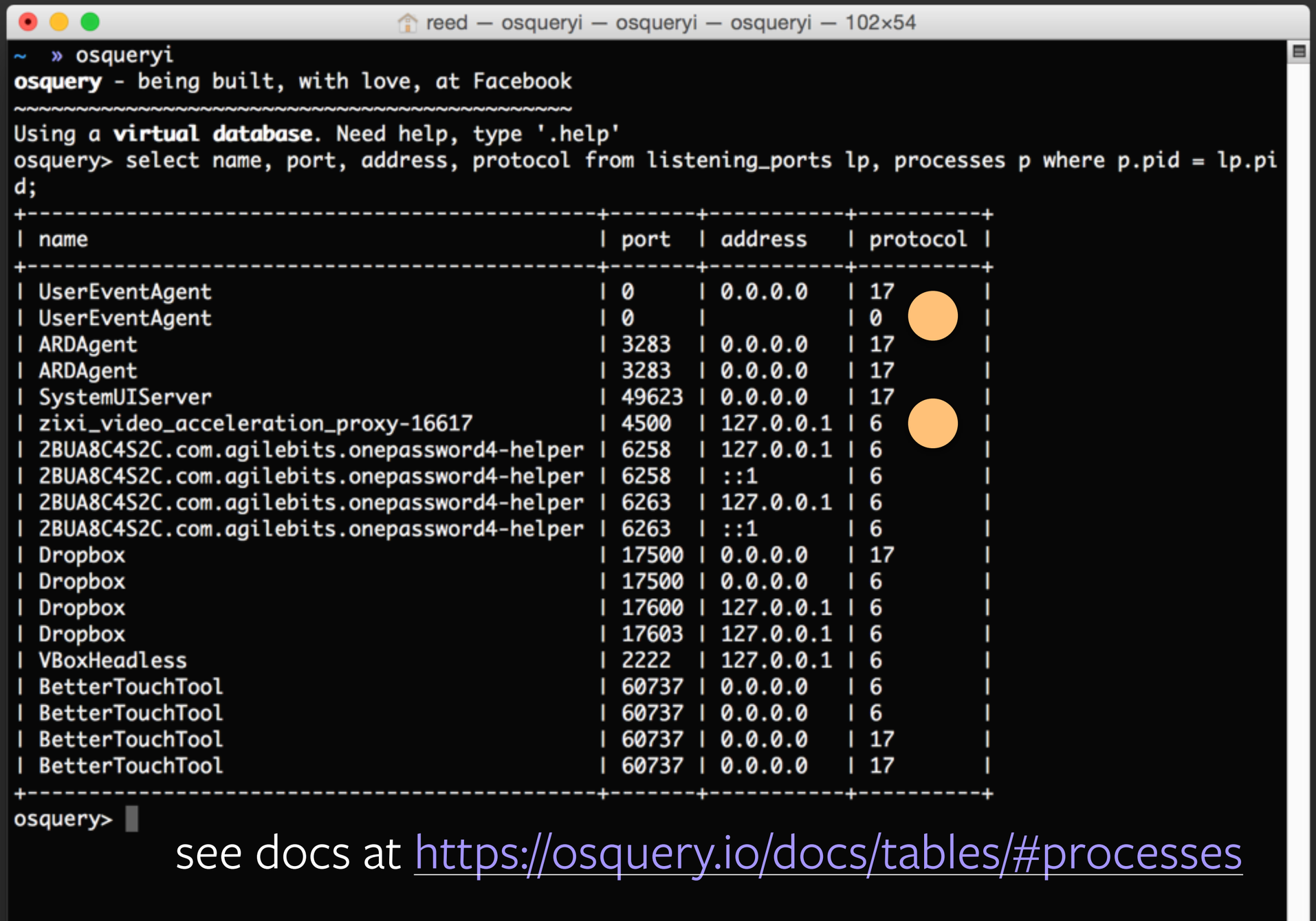
run `osqueryi` and inspect the basic shell help menu




reed — osqueryi — osqueryi — osqueryi — 102×54
~ » osqueryi
osquery - being built, with love, at Facebook
Using a virtual database. Need help, type '.help'
osquery> select * from listening_ports limit 1;
+-----+------+----------+--------+---------+
| pid | port | protocol | family | address |
+-----+------+----------+--------+---------+
| 513 | 0 | 17 | 2 | 0.0.0.0 |
+-----+------+----------+--------+---------+
osquery> select * from listening_ports where address <> '' limit 5;
+-----+-------+----------+--------+-----------+
| pid | port | protocol | family | address |
+-----+-------+----------+--------+-----------+
513	0	17	2	0.0.0.0
546	3283	17	10	0.0.0.0
546	3283	17	2	0.0.0.0
552	49623	17	2	0.0.0.0
862	4500	6	2	127.0.0.1
+-----+-------+----------+--------+-----------+
osquery>
also use .schema listening_ports
see docs at https://osquery.io/docs/tables/


reed — osqueryi — osqueryi — osqueryi — 102×54
~ » osqueryi
osquery - being built, with love, at Facebook
Using a virtual database. Need help, type '.help'
osquery> select * from file;
osquery> select * from file where path = '/System/Library/CoreServices/boot.efi';
osquery> .mode line
osquery> select * from file where path = '/System/Library/CoreServices/boot.efi';
path = /System/Library/CoreServices/boot.efi
directory = /System/Library/CoreServices
filename = boot.efi
inode = 35701116
uid = 0
gid = 0
mode = 0644
device = 0
size = 583736
block_size = 4096
see docs at https://osquery.io/docs/tables/#file


reed — reed@reed-mbp: ~ — ~ — zsh — 102×54
~ » osqueryi --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 = c4cf0a45b00c1a496c1edb7bbc3d734f6e4e07a768eab0df33cfef47dcccf989
~ » osqueryi --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":"c4cf0a45b00c1a496c1edb7bbc3d734f6e4e07a768eab0df33cfef47dcccf989"}
]
~ » osqueryi "select * from kernel_info"
| version | arguments | path | device | md5 |
| 14.5.0 | | System\Library\Caches\com.apple.kext.caches\Startup\kernelcache | E220890C-BE93-42CF-8F56-D9E64B5E7820 | |
~ » osqueryi --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

The most value comes from the `osqueryd` daemon

This uses a JSON-config to set options and define a schedule

Config:

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

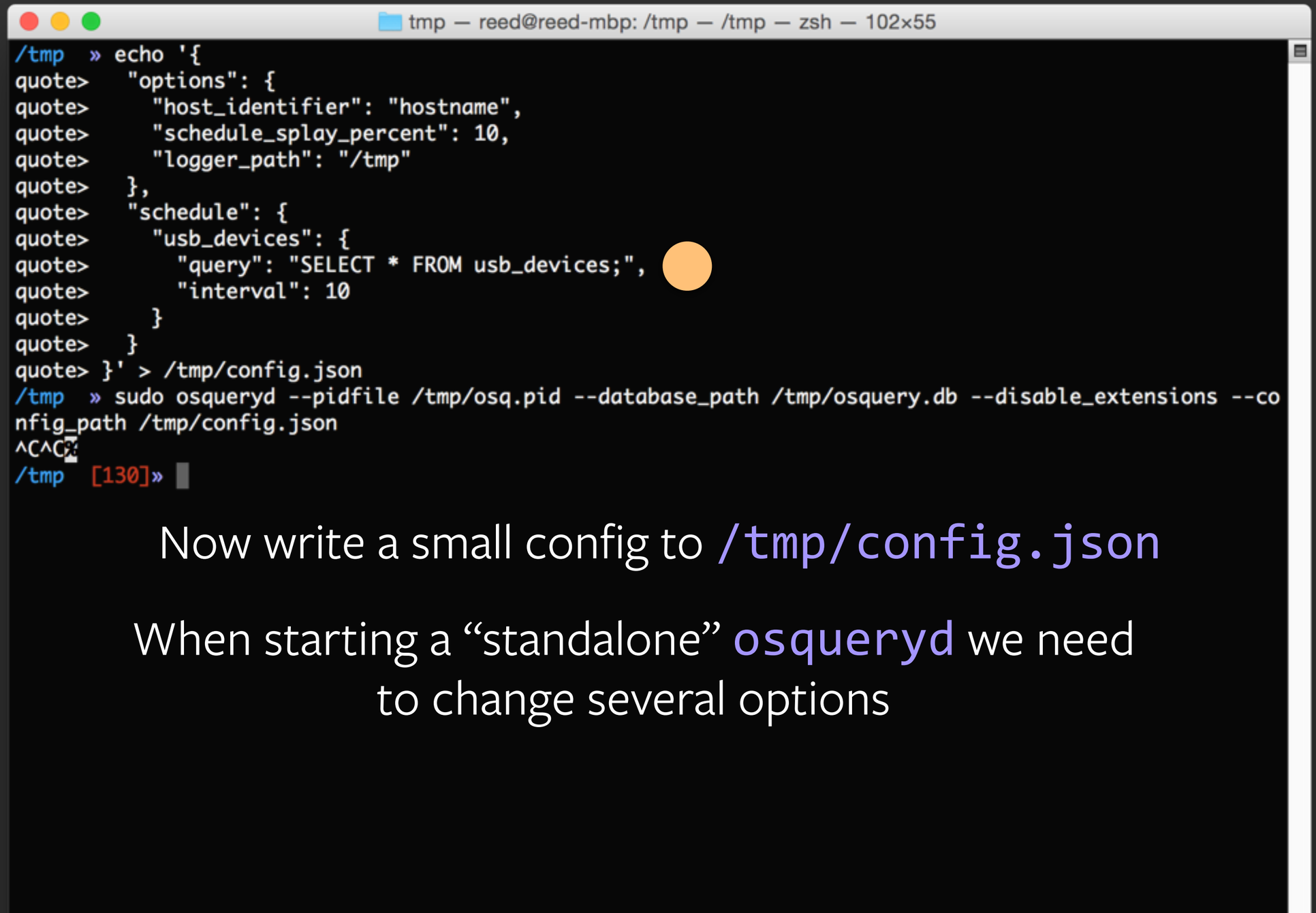
Log (single line):

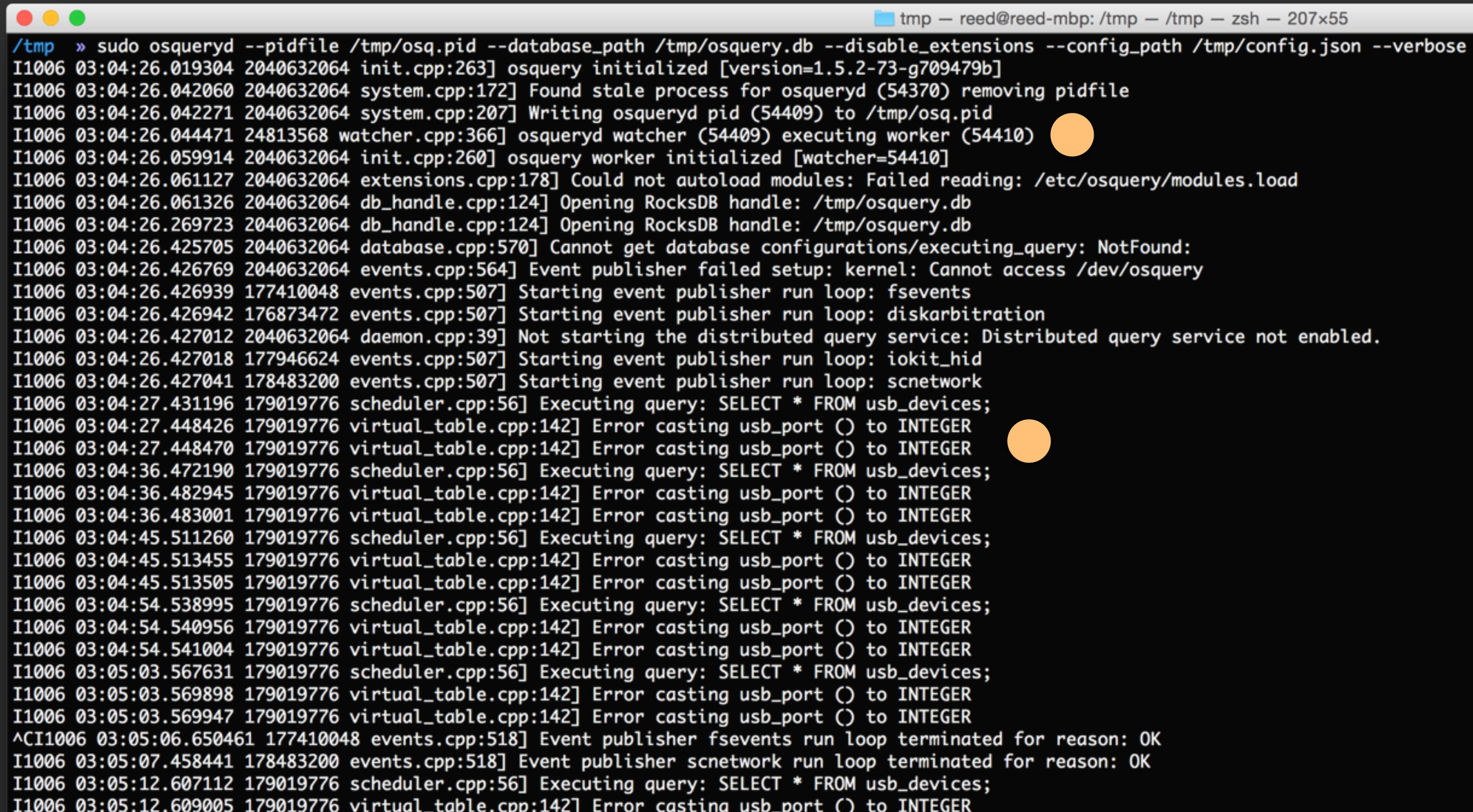
```
{
  "name": "usb_devices",
  "hostIdentifier": "reed-mbp.local",
 "unixTime": "1444120356",
  "columns": {
    "model": "USB Laser Mouse",
    "model_id": "c069",
   "vendor": "Logitech",
    "vendor_id": "046d"
  },
  "action": "added"
}
```

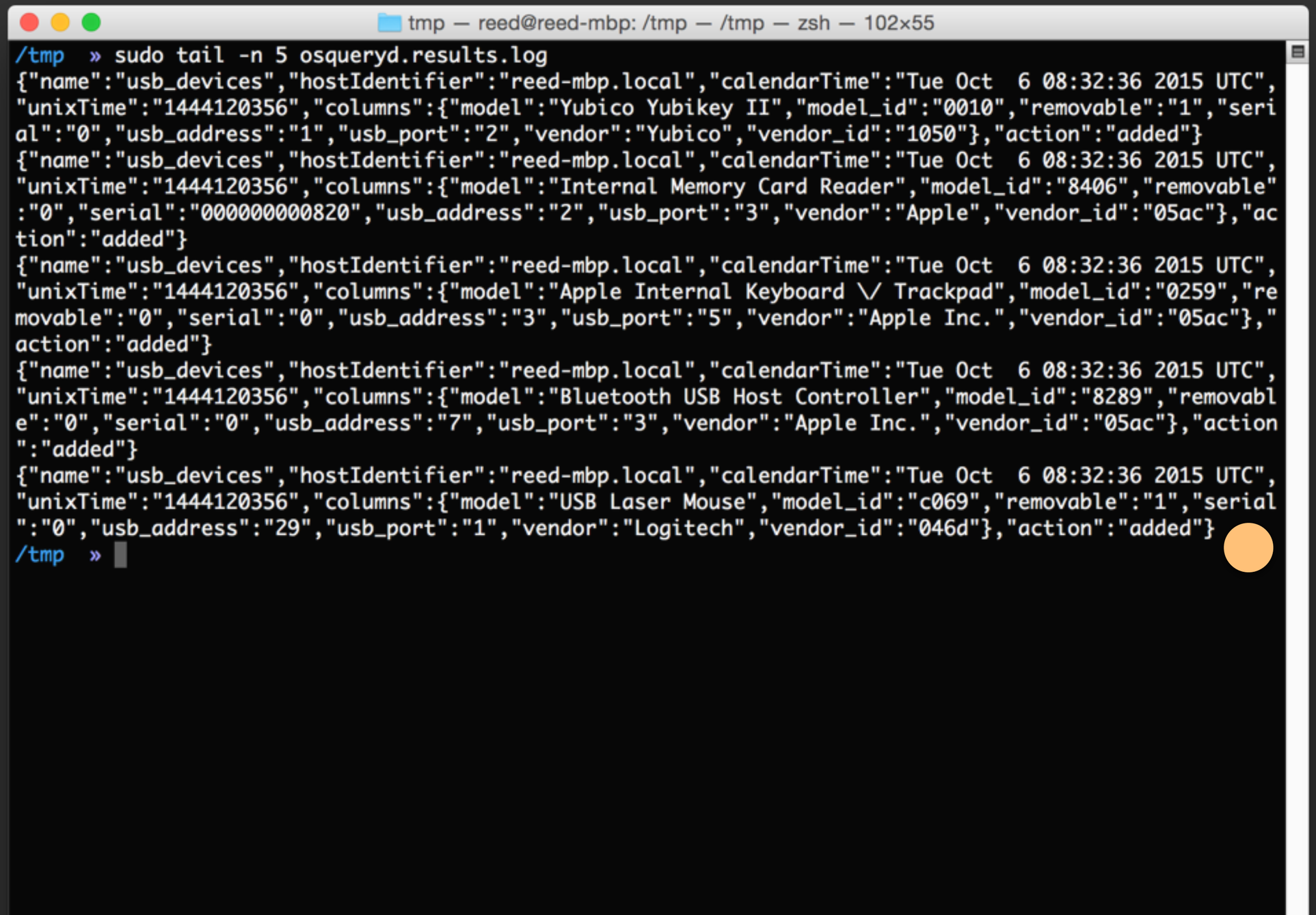
The most value comes from the `osqueryd` daemon
This uses a JSON-config to set options and define a schedule

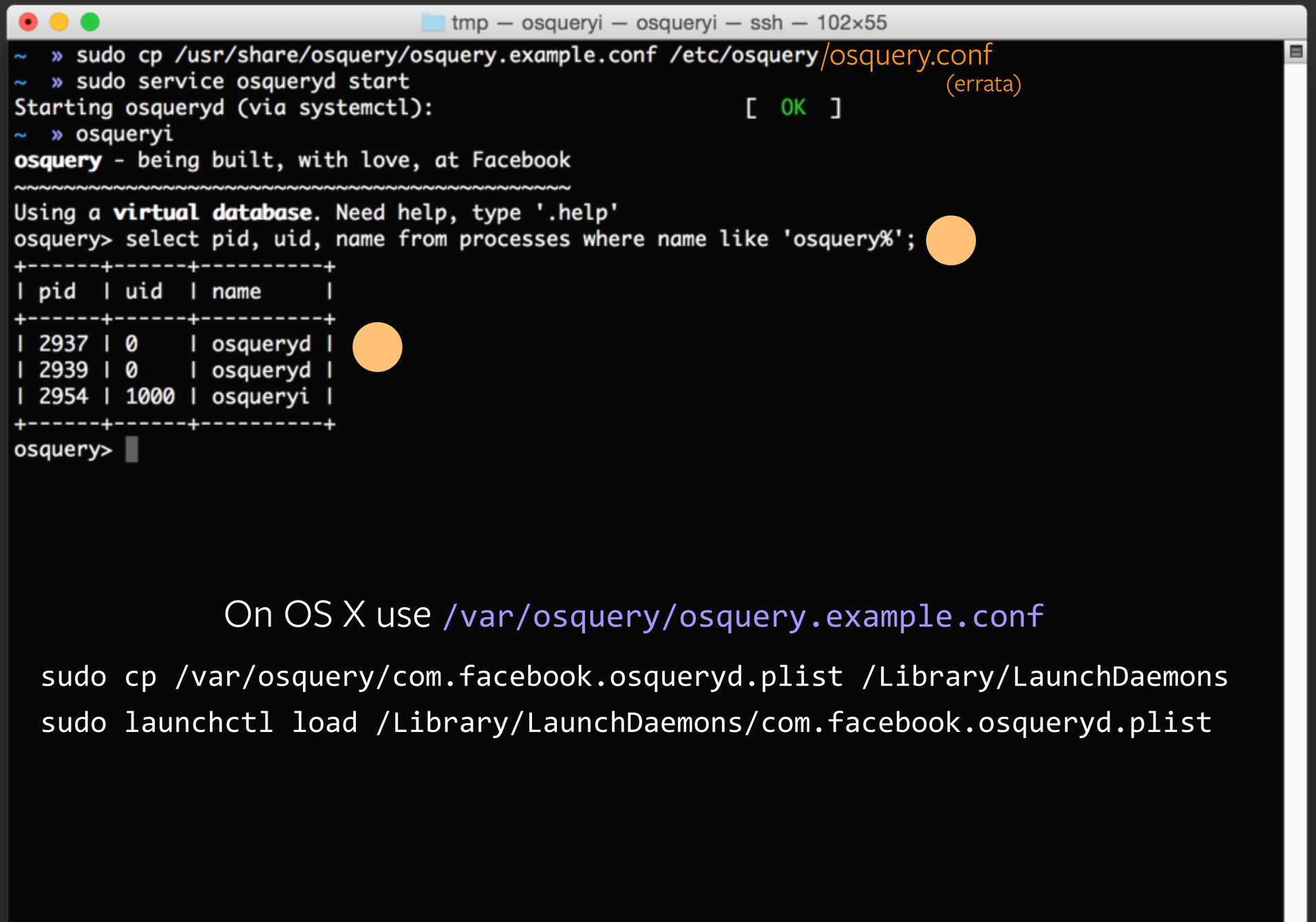
The schedule is a set of QUERY and INTERVAL pairs
The logs are changes in the output of the queries

These queries can be organized into packs, and distributed alongside the osquery package or internally


tmp — reed@reed-mbp: /tmp — /tmp — zsh — 102×55
/tmp » echo '{
quote> "options": {
quote> "host_identifier": "hostname",
quote> "schedule_splay_percent": 10,
quote> "logger_path": "/tmp"
quote> },
quote> "schedule": {
quote> "usb_devices": {
quote> "query": "SELECT * FROM usb_devices;",
quote> "interval": 10
quote> }
quote> }
quote> }' > /tmp/config.json
/tmp » sudo osqueryd --pidfile /tmp/osq.pid --database_path /tmp/osquery.db --disable_extensions --co
nfig_path /tmp/config.json
^C^C
/tmp [130]»
Now write a small config to /tmp/config.json
When starting a “standalone” osqueryd we need
to change several options

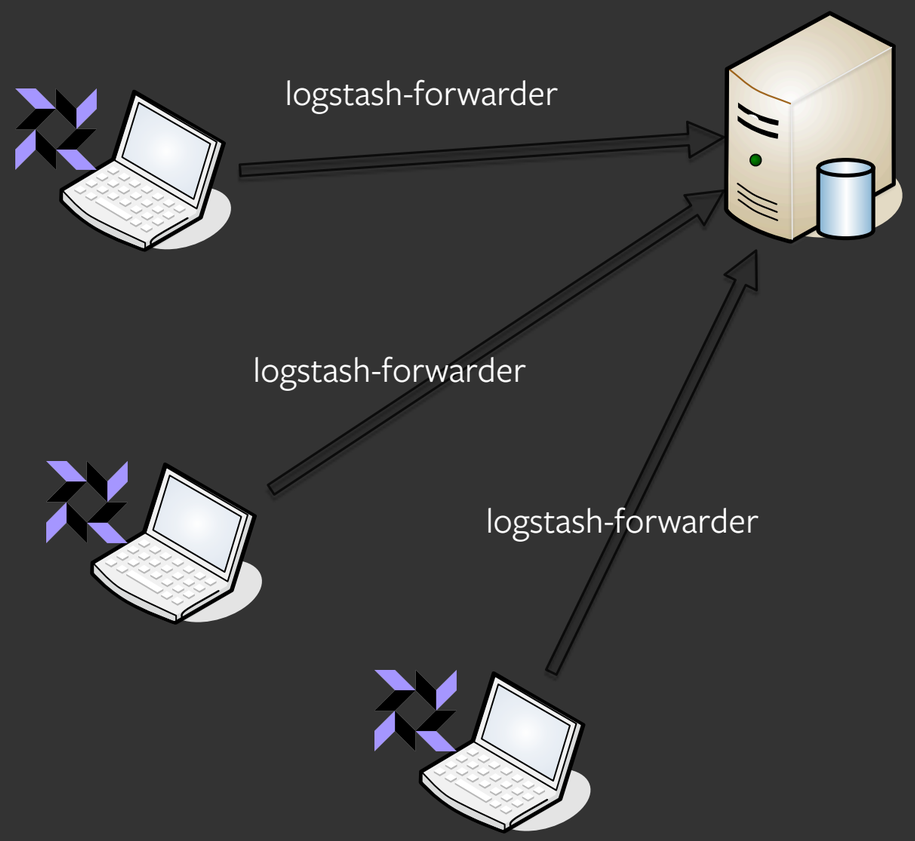

tmp — reed@reed-mbp: /tmp — /tmp — zsh — 207×55
/tmp » sudo osqueryd --pidfile /tmp/osq.pid --database_path /tmp/osquery.db --disable_extensions --config_path /tmp/config.json --verbose
I1006 03:04:26.019304 2040632064 init.cpp:263] osquery initialized [version=1.5.2-73-g709479b]
I1006 03:04:26.042060 2040632064 system.cpp:172] Found stale process for osqueryd (54370) removing pidfile
I1006 03:04:26.042271 2040632064 system.cpp:207] Writing osqueryd pid (54409) to /tmp/osq.pid
I1006 03:04:26.044471 24813568 watcher.cpp:366] osqueryd watcher (54409) executing worker (54410)
I1006 03:04:26.059914 2040632064 init.cpp:260] osquery worker initialized [watcher=54410]
I1006 03:04:26.061127 2040632064 extensions.cpp:178] Could not autoload modules: Failed reading: /etc/osquery/modules.load
I1006 03:04:26.061326 2040632064 db_handle.cpp:124] Opening RocksDB handle: /tmp/osquery.db
I1006 03:04:26.269723 2040632064 db_handle.cpp:124] Opening RocksDB handle: /tmp/osquery.db
I1006 03:04:26.425705 2040632064 database.cpp:570] Cannot get database configurations/executing_query: NotFound:
I1006 03:04:26.426769 2040632064 events.cpp:564] Event publisher failed setup: kernel: Cannot access /dev/osquery
I1006 03:04:26.426939 177410048 events.cpp:507] Starting event publisher run loop: fsevents
I1006 03:04:26.426942 176873472 events.cpp:507] Starting event publisher run loop: diskarbitration
I1006 03:04:26.427012 2040632064 daemon.cpp:39] Not starting the distributed query service: Distributed query service not enabled.
I1006 03:04:26.427018 177946624 events.cpp:507] Starting event publisher run loop: iokit_hid
I1006 03:04:26.427041 178483200 events.cpp:507] Starting event publisher run loop: scnetwork
I1006 03:04:27.431196 179019776 scheduler.cpp:56] Executing query: SELECT * FROM usb_devices;
I1006 03:04:27.448426 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
I1006 03:04:27.448470 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
I1006 03:04:36.472190 179019776 scheduler.cpp:56] Executing query: SELECT * FROM usb_devices;
I1006 03:04:36.482945 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
I1006 03:04:36.483001 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
I1006 03:04:45.511260 179019776 scheduler.cpp:56] Executing query: SELECT * FROM usb_devices;
I1006 03:04:45.513455 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
I1006 03:04:45.513505 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
I1006 03:04:54.538995 179019776 scheduler.cpp:56] Executing query: SELECT * FROM usb_devices;
I1006 03:04:54.540956 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
I1006 03:04:54.541004 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
I1006 03:05:03.567631 179019776 scheduler.cpp:56] Executing query: SELECT * FROM usb_devices;
I1006 03:05:03.569898 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
I1006 03:05:03.569947 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER
^CI1006 03:05:06.650461 177410048 events.cpp:518] Event publisher fsevents run loop terminated for reason: OK
I1006 03:05:07.458441 178483200 events.cpp:518] Event publisher scnetwork run loop terminated for reason: OK
I1006 03:05:12.607112 179019776 scheduler.cpp:56] Executing query: SELECT * FROM usb_devices;
I1006 03:05:12.609005 179019776 virtual_table.cpp:142] Error casting usb_port () to INTEGER


tmp — reed@reed-mbp: /tmp — /tmp — zsh — 102×55
/tmp » 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","seri
al":"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":"000000000820","usb_address":"2","usb_port":"3","vendor":"Apple","vendor_id":"05ac"},"ac
tion":"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","re
movable":"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","removabl
e":"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":"c069","removable":"1","serial
":"0","usb_address":"29","usb_port":"1","vendor":"Logitech","vendor_id":"046d"},"action":"added"}
/tmp »


tmp — osqueryi — osqueryi — ssh — 102×55
~ » sudo cp /usr/share/osquery/osquery.example.conf /etc/osquery/osquery.conf
(errata)
~ » sudo service osqueryd start
Starting osqueryd (via systemctl): [OK]
~ » osqueryi
osquery - being built, with love, at Facebook
Using a virtual database. Need help, type '.help'
osquery> select pid, uid, name from processes where name like 'osquery%';
pid	uid	name
2937	0	osqueryd
2939	0	osqueryd
2954	1000	osqueryi
osquery>
On OS X use /var/osquery/osquery.example.conf
sudo cp /var/osquery/com.facebook.osqueryd.plist /Library/LaunchDaemons
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

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:

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

10-osquery.conf

```
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://osquery.readthedocs.org/en/stable/deployment/configuration/

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

work on osquery with us

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