Detection and Incident Response

With osquery

Javier Marcos
@javutin

WOPR SUMMIT



$ whoami

- Security Engineer/Incident Responder
- Open source contributor (github.com/javuto)
- Former IBM, Facebook, Uber and Airbnb
- Current BitMEX



Agenda

Part 1: osquery, let's talk about it

- What is it?
- osqueryi basics
- osquery tables
- Package files

(break)



Agenda

Part 2: Scaling osquery

- Do you need a Daemon? osqueryd!
- Flags and configuration files
- Scheduled queries, packs and watchdog
- Remote API: TLS endpoint

(break)

Agenda

Part 3: IR using osquery

- File Integrity Monitoring
- Yara rule hunting
- Extensions

(EOF)



osquery shell

```
ssh -p 2222 osquery@192.168.1.2

(Password: woprsummit)
```



osquery packages

MacOS:	`brew install osquery`
Windows:	`choco install osquery`
APT Linux:	`sudo apt-get install osquery`
RPM Linux:	`sudo yum install osquery`
FreeBSD:	`pkg install osquery`

https://osquery.io/downloads

What is osquery?

- Explore your operative system using SQL
- Host visibility motivated by intrusion detection

🚨 100% OS API usage, no fork execve 🚨

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



osquery motivation

- What machines have chrome extension abc123 installed?
- How many file descriptors were open yesterday by hour?
- Is anything bridging routes from VPN to LAN?



Why use SQL?

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

- Core concepts of SQL are platform agnostic
- Most devs and administrators know SQL



Why use SQL?

[concept]

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



Why use SQL?

```
                [attributes]                [concept]
SELECT pid,name,uid FROM processes
```



Why use SQL?

[attributes] [concept]

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

```
WHERE uid != 0
```

[constraints]



Why use SQL?

[attributes] [concept]

```
SELECT pid,name,uid FROM processes

JOIN users ON processes.uid=users.uid
                                [join]

WHERE uid != 0
      [constraints]
```



osqueryi basics

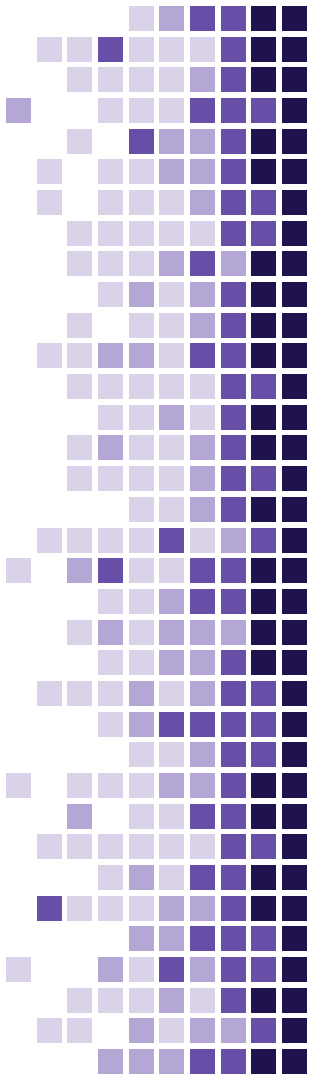
```
osquery> .help
Welcome to the osquery shell. Please explore your OS!
You are connected to a transient 'in-memory' virtual database.

.all [TABLE]     Select all from a table
.bail ON|OFF     Stop after hitting an error
.echo ON|OFF     Turn command echo on or off
.exit            Exit this program
.features        List osquery's features and their statuses
.headers ON|OFF  Turn display of headers on or off
.help            Show this message
```



osqueryi basics

```
osquery> .tables
  => acpi_tables
  => apt_sources
  => arp_cache
  => augeas
  => authorized_keys
  => block_devices
  => carbon_black_info
  => carves
  => chrome_extensions
  => cpu_time
  => cpuid
  => crontab
  => curl
  => curl_certificate
  => deb_packages
  => device_file
  => device_hash
  => device_partitions
  => disk_encryption
  => dns_resolvers
  => docker_container_labels
  => docker_container_mounts
  => docker_container_networks
  => docker_container_ports
```





osqueryi basics

```
osquery> pragma table_info('system_info');
+-----+--------------------+---------+---------+------------+----+
| cid | name               | type    | notnull | dflt_value | pk |
+-----+--------------------+---------+---------+------------+----+
| 0   | hostname           | TEXT    | 0       |            | 0  |
| 1   | uuid               | TEXT    | 0       |            | 0  |
| 2   | cpu_type           | TEXT    | 0       |            | 0  |
| 3   | cpu_subtype        | TEXT    | 0       |            | 0  |
| 4   | cpu_brand          | TEXT    | 0       |            | 0  |
| 5   | cpu_physical_cores | INTEGER | 0       |            | 0  |
| 6   | cpu_logical_cores  | INTEGER | 0       |            | 0  |
| 7   | cpu_microcode      | TEXT    | 0       |            | 0  |
```



osquery tables

- 229 tables in version 3.3.2
- 4 different platforms
  - Mac, windows, linux and freebsd
- Data easy to collect and to join

https://osquery.io/schema/3.3.2



osquery tables

- acpi_tables
- arp_cache
- apps
- authorized_keys
- autoexec
- battery
- block_devices
- browser_plugins
- certificates
- cpu_time

...

- cpu_info
- crontab
- cups_jobs
- deb_packages
- disk_info
- dns_resolvers
- docker_info
- drivers
- etc_hosts
- elf_info

...

- etc_services
- event_taps
- file
- iptables
- kernel_info
- known_hosts
- launchd
- mounts
- preferences

...

And many more!

https://osquery.io/schema/3.3.2



Tables execute when used

```
osquery> SELECT datetime FROM time;
+----------------------+
| datetime             |
+----------------------+
| 2019-03-01T04:16:07Z |
+----------------------+
```

...


A FEW
MOMENTS LATER



Tables execute when used

```
osquery> SELECT datetime FROM time;
+----------------------+
| datetime             |
+----------------------+
| 2019-03-01T04:20:18Z |
+----------------------+
```

...



Tables execute when used

```
SELECT datetime FROM time;
2019-03-01T04:16:07Z

                    ...

SELECT datetime FROM time;
2019-03-01T04:20:18Z
```



Tables with parameters

```
osquery> SELECT directory FROM file WHERE path =
‘/etc/issue’;

+-----------+
| directory |
+-----------+
| /etc      |
+-----------+
```



Tables with parameters

```
osquery> SELECT md5 FROM file JOIN hash USING
(path) WHERE path = ‘/etc/issue’;

+----------------------------------+
| md5                              |
+----------------------------------+
| b954418e6a50d4d4cb8f02776d867550 |
+----------------------------------+
```



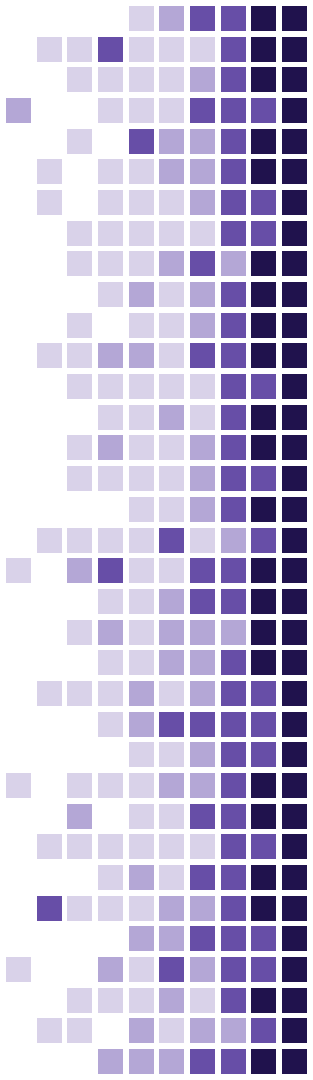
Tables easy to collect

```
osquery> SELECT * FROM rpm_packages;

osquery> SELECT * FROM users;

osquery> SELECT * FROM kernel_modules;

osquery> SELECT * FROM startup_items;
```





osquery files in Linux

- **deb/rpm**

```
/etc/osquery/osquery.conf      ← Config
/var/log/osquery               ← Logs
/usr/bin                       ← Bins
/usr/share/osquery/packs       ← Packs
```



osquery files in Mac OS

- **brew/pkg**

```
/var/osquery/osquery.conf      ← Config
/var/log/osquery               ← Logs
/usr/local/bin                 ← Bins
/var/osquery/packs             ← Packs
```



osquery files in Windows

- **choco/msi**

```
C:\ProgramData\osquery\osquery.conf  ← Config
C:\ProgramData\osquery\log           ← Logs
C:\ProgramData\osquery\              ← Bins
C:\ProgramData\osquery\packs         ← Packs
```



Quiz!

- What is the system hostname?
- What users exist on the system?
- What processes are running?



Quiz!

- What is the system hostname?

```
SELECT hostname FROM system_info;
```

- What users exist on the system?

- What processes are running?



Quiz!

- What is the system hostname?

```
SELECT hostname FROM system_info;
```

- What users exist on the system?

```
SELECT uid, username FROM users;
```

- What processes are running?



Quiz!

- What is the system hostname?

```
SELECT hostname FROM system_info;
```

- What users exist on the system?

```
SELECT uid, username FROM users;
```

- What processes are running?

```
SELECT pid, name, path FROM processes;
```



Quiz!

- What is the username and the shell of the user that has a running process?

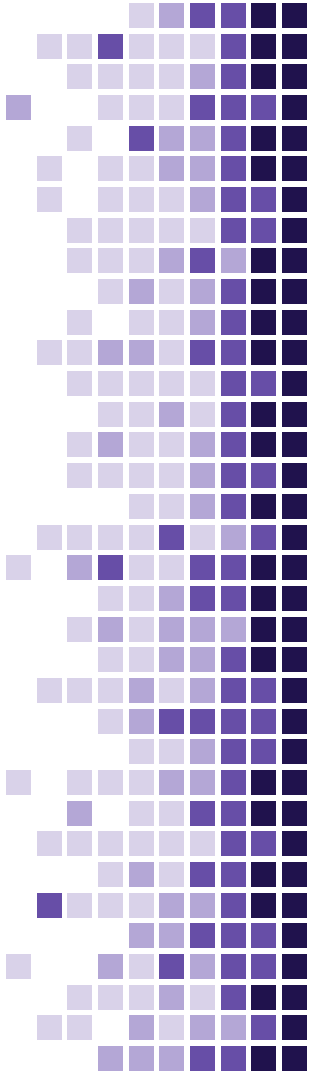
Quiz!

- What is the username and the shell of the user that has a running process?

```
SELECT p.pid, p.name, p.path, u.username,
u.shell FROM processes AS p JOIN users AS u ON
p.uid = u.uid;
```



Questions so far?

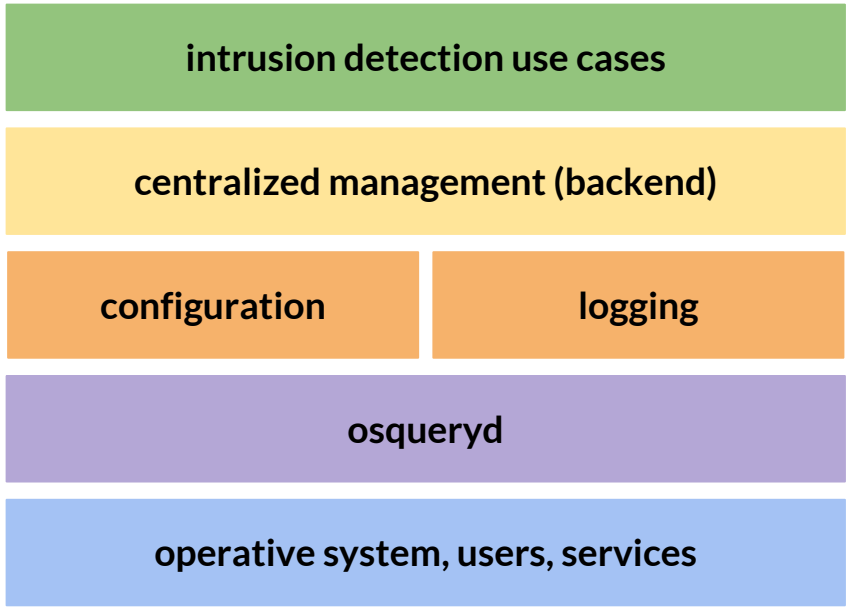




The osquery daemon: osqueryd

- Init, systemd, launchd, win service
- Queries executed on schedule
- Logs for daemon status and query results
- Heavily configurable

The osquery daemon: osqueryd





osquery.flags

- Flagfile can bootstrap how to config

```
$ osqueryd --flagfile /etc/osquery/osquery.flags
```

- It is common to use chef/puppet to write flags

```
$ osqueryd/osqueryi --help
```

osquery.conf - options

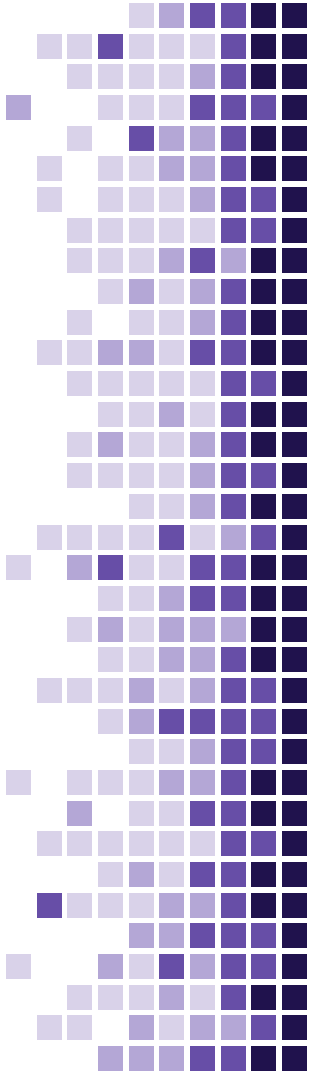
```
$ osquery[d-i] --config_path /path/to/osquery.conf
```

```
"options": {
  "config_plugin": "filesystem",
  "logger_plugin": "filesystem",
  "schedule_splay_percent": "10",
  "utc": "true"
   ...
}
```



osquery.conf - schedule

```
"schedule": {
  "example_query1": {
    "query": "SELECT * FROM users;",
    "interval": 60
  },
  "example_query2": {
    "query": "SELECT * FROM processes;",
    "interval": 3600
  },
}
```





Scheduled queries

`query`: The exact query string to run

`interval`: Run the query every this seconds

`platform`: Restrict query to this platform

`shard`: Only run on this % of hosts

`snapshot`: Return all results on each execution



osquery.conf - decorators

```
"decorators": {
  "load": [
    "SELECT uuid FROM system_info;"
  ],
  "always": [
    "SELECT pid FROM osquery_info;"
  ]
}
```

osquery.conf - packs

```
"packs": {
  "osquery-monitoring": "osquery-monitoring.conf",
  "incident-response": "incident-response.conf",
  "it-compliance": "it-compliance.conf",
  "osx-attacks": "osx-attacks.conf",
  "vuln-management": "vuln-management.conf"
  "hardware-monitoring": "hardware-monitoring.conf",
  "ossec-rootkit": "ossec-rootkit.conf",
  "windows-hardening": "windows-hardening.conf",
  "windows-attacks": "windows-attacks.conf"
},
```



osquery.conf - packs

```
// incident-response.conf
"queries": {
  "launchd": {
    "query" : "select * from launchd;",
    "interval" : "3600",
    "platform" : "darwin",
    "version" : "1.4.5",
  },
  ...
```



osqueryd watchdog

- osqueryd by default works on a single worker
- Periodically inspects CPU/memory usage
- **restart if**: Over 60% CPU usage for 9 s
- **restart if**: Over 200M memory allocated



osqueryd remote API

- TLS Plugin allows for remote configuration + flags

```
--tls_client_cert     Optional path to a TLS client-auth PEM certificate

--tls_client_key      Optional path to a TLS client-auth PEM private key

--tls_hostname        TLS/HTTPS hostname for Config, Logger, and Enroll

--tls_server_certs    Optional path to a TLS server PEM certificate(s)
bundle
```

osqueryd remote API

- TLS endpoint allows **Distributed queries**

➔ On demand queries
➔ Return results immediately on a pull model

➔ Very useful for investigations

osqueryd remote API

- Options for TLS endpoint solutions

- ➔ Doorman
- ➔ Uptycs
- ➔ Kolide
- ➔ Zentral
- ➔ SGT
- ➔ Windmill
- ➔ CB LiveOps
- ➔ AlienVault

Questions?

File Integrity Monitoring (FIM)

```
"file_paths": {
  "homes": ["/home/*"]
},
"schedule": {
  "file_events": {
    "query": "SELECT * FROM file_events;",
    "interval": 300
  }
}
```

File Integrity Monitoring (FIM)

- **Events** tables: `file_events`
- Subscribe to async OS events
- osquery will buffer these events over time
- Selecting from the table shows a slice

https://osquery.readthedocs.io/en/stable/deployment/file-integrity-monitoring/

Yara rules hunting

```
"yara": {
  "signatures": {
    "sig_group_1": [
      "/tmp/foo.sig", "/tmp/bar.sig"],
    "sig_group_2": [ "/tmp/baz.sig" ]
  },
  "file_paths": {
  }
}
```

Yara rules hunting

- **Events** table: `yara_events`
- Also on-demand scanning:

```
SELECT * FROM yara WHERE path="/bin/ls" AND
sig_group="sig_group_1";
```

https://osquery.readthedocs.io/en/stable/deployment/yara/



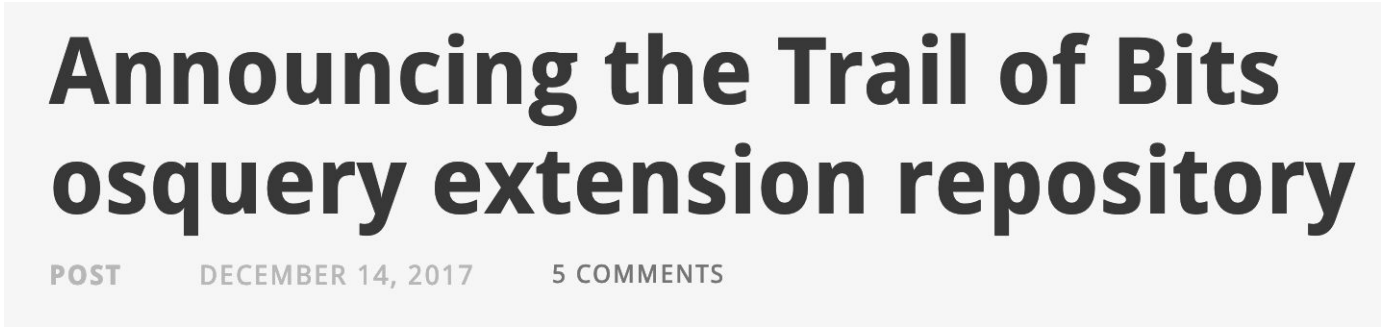
osquery extensions

```
$ osquery[d-i] --extension /path/to/my_extension.ext
```

- Write them in C++, python and golang…
- Or any other language that supports Thrift

https://osquery.readthedocs.io/en/stable/development/osquery-sdk/

osquery extensions



https://blog.trailofbits.com/2017/12/14/announcing-the-trail-of-bits-osquery-extension-repository/

osquery extensions

- darwin_unified_log
- efigy
- fwctl
- iptables
- libraries
- network_monitor
- ntfs_forensics
- opt_dependencies
- santa
- windows_sync_objects

https://github.com/trailofbits/osquery-extensions



osquery documentation

- Wiki

  https://osquery.readthedocs.io

- Code

  https://osquery.io

Thank you!

@javutin