



Osquery

Building things atop
Osquery

Hello, and welcome to my talk. Building things a top Osquery.

Hugh Neale

- Director of Zecurity
(Cybersecurity operations platform atop Osquery)
- SecOps at a startup bank. **We spent all our time integrating security tools.**
- Osquery replaced a number of vendors.



@hughneale
@zecurity

<https://www.zecurity.com/>
<https://medium.com/@zecurity/>

Who am I?

Why we chose Osquery, as a swiss army knife

Osquery has replaced a collection of other tools, not that it necessitated a large time commitment to integrate with other tools.

Agenda

- What we've built you can build atop Osquery.
 - Inventory management
 - Monitoring
 - Vulnerability management
 - IAM & AAA
 - Networking
 - Data loss protection
 - Compliance & risk

30 minutes, 30 slides let's go!

- What is Osquery
- Using remote distributed queries
- 30 slides, 30 minutes buckle up

Inventory

- Hardware
- OS information
- Installed apps & packages
- Running processes
- Networking (ARP)
- Users & Groups

```
|Hughes-MacBook-Pro:~ hughneale$ osqueryi
Using a virtual database. Need help, type '.help'
|osquery> .mode line
|osquery> SELECT hostname, cpu_subtype, cpu_brand, physical_memory, hardware_vendor, hardware_model FROM system_info;
    hostname = hughs-macbook-pro.local
    cpu_subtype = Intel x86-64n Haswell
    cpu_brand = Intel(R) Core(TM) i7-4980HQ CPU @ 2.80GHz
    physical_memory = 17179869184
    hardware_vendor = Apple Inc.
    hardware_model = MacBookPro11,5
|osquery> SELECT name, version, build, platform FROM os_version;
    name = Mac OS X
    version = 10.14.5
    build = 18F132
```

Osquery Tables

platform_info, os_version, system_info

apps, programs, rpm_packages, deb_packages

processes, process_events

arp_table

users, groups

- Talk about the structure of the slides
- Really hard to know what's on your estate
- Once you've got Osquery installed you can start to run these queries
- I'll touch on apps and versions / vulnerabilities
- What binaries are being run
- ARP tables for finding important nodes

```
SELECT hostname, cpu_subtype, cpu_brand, physical_memory, hardware_vendor,
hardware_model FROM system_info;
SELECT name, version, build, platform FROM os_version;
```

Asset

Name: Hugh's Laptop
Hostname: Hugh's-MacBook-Pro.local
Owner: Hugh Neale
Tags:
Serial: G8WM
Type: LAPTOP
Platform: DARWIN

Team: Zecurity
Public IP: 81.98.131.11
Location: Islington, London EC1
Interface IP(s): fe80::c45c:f9f3:9465:f64d%utun0, fe80::1026:f8ff:fea7:dd41%awdl0, 192.168.1.127.94.0.3, 127.94.0.1, 127.94.0.5, 127.94.0.2, 127.94.0.0.3
Osquery version: 3.3.0
Santa version: 0.9.30
Definitions: 13th Jun 2019, 21:42:24 (4 min)
Last seen: 13th Jun 2019, 21:46:39 (a few seconds ago)
Last updated: 13th Jun 2019, 21:46:39 (a few seconds ago)
Created: 10th Jun 2017, 15:08:47 (2 years ago)

Interface	Type	Address
utun0	OTHER	fe80::c45c:f9f3:9465:f64d: ffff:ffff:ffff:ffff:
awdl0	ETHERNET	fe80::1026:f8ff:fea7:dd4f: ffff:ffff:ffff:ffff:
en0	ETHERNET	192.168.1.158 255.255.255.0 ac:bc:32:8c:21:99

PID	Name
85455 (1)	syncdefaults /System/Library/PrivateFrameworks/SyncedDefaults.framework/Support/syncdefaultsd
85454 (431)	Google Chrome Helper com.google.Chrome.helper / 3729.169 /Applications/Google Chrome.app/Contents/Versions/74.0.3729.169/Google Chrome Helper
85450	Google Chrome Helper com.google.Chrome.helper / 3729.169 /Applications/Google Chrome.app/Contents/Versions/74.0.3729.169/Google Chrome Helper

Innstor PenDrive
S/N:4253097777

Yubico Yubikey 4 OTP+U2F+CCID
S/N:0

Yubico Yubikey 4 OTP+U2F+CCID
S/N:0

IP: UnKnown, 41.212.122.222
First seen: 10th May 2019, 12:48:02
Last seen: 15th May 2019, 17:56:51 (a month ago)
Time at location: 5 hours

IP: UnKnown, 79.223.202.129
First seen: 11th May 2019, 16:36:48
Last seen: 14th May 2019, 07:25:55 (a month ago)
Time at location: 63 hours



5

- Quickly walk through the screen shots

Process monitoring and visibility
for EDR, binary analysis,
fingerprinting, Virus Total

Processes

What can you do
with processes?

- First and foremost processes

<https://medium.com/@zecurity/process-monitoring-with-osquery-22c6f38fc239>

Process trees

```
[Hughes-MacBook-Pro:~ hughneale$ osqueryi
Using a virtual database. Need help, type '.help'
osquery> WITH RECURSIVE
    ...> rc(pid, parent, name) AS (
    ...>     SELECT pid, parent, name FROM processes WHERE pid = 55334
    ...>     UNION ALL
    ...>     SELECT p.pid, p.parent, p.name FROM processes AS p, rc
    ...>     WHERE p.pid = rc.parent
    ...>     AND p.pid != 0
    ...> )
    ...>     SELECT pid, parent, name FROM rc LIMIT 20;
+-----+-----+
| pid | parent | name   |
+-----+-----+
| 55334 | 55310 | osqueryd |
| 55310 | 55309 | bash    |
| 55309 | 484   | login   |
| 484   | 1      | Terminal |
| 1     | 0      | launchd|
+-----+-----+
osquery>
```

Osquery Tables

processes



process_events



process_envs, process_file_events,
process_memory_map,
process_namespaces,
process_open_files



Make a note of the icons for the tables

Parent relationships are important to understand where a process came from

- Who ran it
- What was the context for it running
- EDR

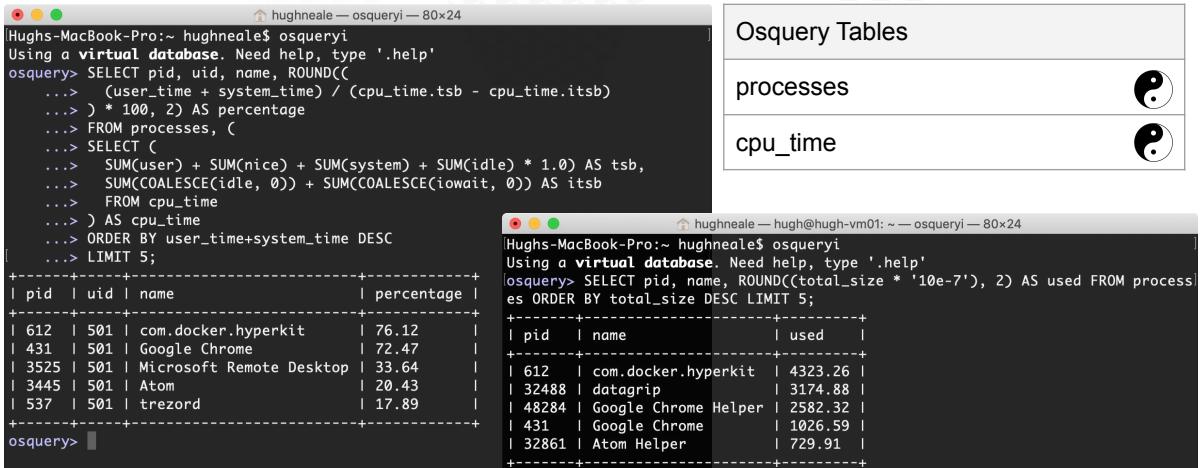
Talk through the recursive query

We can create a nice EDR visualization

WITH RECURSIVE

```
rc(pid, parent, name) AS (
    SELECT pid, parent, name FROM processes WHERE pid = 55334
    UNION ALL
    SELECT p.pid, p.parent, p.name FROM processes AS p, rc
    WHERE p.pid = rc.parent
    AND p.pid != 0
)
SELECT pid, parent, name FROM rc LIMIT 20;
```

Performance monitoring



Hughes-MacBook-Pro:~ hughneale\$ osqueryi
Using a **virtual database**. Need help, type '.help'
osquery> SELECT pid, uid, name, ROUND((
...> (user_time + system_time) / (cpu_time.ts - cpu_time.ts)
...>) * 100, 2) AS percentage
...> FROM processes, (
...> SELECT (
...> SUM(user) + SUM(nice) + SUM(system) + SUM(idle) * 1.0) AS ts,
...> SUM(COALESCE(idle, 0)) + SUM(COALESCE(iowait, 0)) AS tsb
...> FROM cpu_time
...>) AS cpu_time
...> ORDER BY user_time+system_time DESC
...> LIMIT 5;
+-----+-----+-----+-----+
| pid | uid | name | percentage |
+-----+-----+-----+-----+
612	501	com.docker.hyperkit	76.12
431	501	Google Chrome	72.47
3525	501	Microsoft Remote Desktop	33.64
3445	501	Atom	20.43
537	501	trezord	17.89
+-----+-----+-----+-----+
osquery>

Hughes-MacBook-Pro:~ hughneale\$ osqueryi
Using a **virtual database**. Need help, type '.help'
osquery> SELECT pid, name, ROUND((total_size * '10e-7'), 2) AS used FROM processes ORDER BY total_size DESC LIMIT 5;
+-----+-----+-----+
| pid | name | used |
+-----+-----+-----+
612	com.docker.hyperkit	4323.26
32488	datagrip	3174.88
48284	Google Chrome Helper	2582.32
431	Google Chrome	1026.59
32861	Atom Helper	729.91
+-----+-----+-----+

We've got a customer who is use monitoring for production workloads
Talk more about this query

<https://medium.com/@zercurity/process-monitoring-with-osquery-22c6f38fc239>

```
SELECT pid, uid, name, ROUND((  
    (user_time + system_time) / (cpu_time.ts - cpu_time.ts)  
) * 100, 2) AS percentage  
FROM processes, (  
    SELECT (  
        SUM(user) + SUM(nice) + SUM(system) + SUM(idle) * 1.0) AS ts,  
        SUM(COALESCE(idle, 0)) + SUM(COALESCE(iowait, 0)) AS tsb  
    FROM cpu_time  
) AS cpu_time  
ORDER BY user_time+system_time DESC  
LIMIT 5;
```

```
SELECT pid, name, ROUND((total_size * '10e-7'), 2) AS used FROM processes  
ORDER BY total_size DESC LIMIT 5;
```

Process hashing & users

```
[Hughes-MacBook-Pro:~ hughneale$ osqueryi
Using a virtual database. Need help, type '.help'
osquery> SELECT DISTINCT h.md5, p.name, u.username
...> FROM processes AS p
...> INNER JOIN hash AS h ON h.path = p.path
...> INNER JOIN users AS u ON u.uid = p.uid
...> ORDER BY start_time DESC
...> LIMIT 5;
+-----+-----+-----+
| md5      | name    | username |
+-----+-----+-----+
| add306f504de3f50ec547de00ccb86db | osqueryd   | hughneale |
| a71017e283350c35a78df7a673a2137d | syncdefaultsd | hughneale |
| f85b6b07e9f14a0645d3599ee822b954 | Preview    | hughneale |
| bd0e6a57439d45704ed98a8129d68b05 | quicklookd  | hughneale |
| 9bd617e4f4fcab3249ba578468372852 | QuickLookSatellite | hughneale |
+-----+-----+-----+
osquery>
```

Osquery Tables

processes	
process_events	
hashes	
users	

Note: **--read_max=524288000**
to hash (almost) all the
binaries.

Virus total

Read max gotcha

```
SELECT DISTINCT h.md5, p.name, u.username
FROM processes AS p
INNER JOIN hash AS h ON h.path = p.path
INNER JOIN users AS u ON u.uid = p.uid
ORDER BY start_time DESC
LIMIT 5;
```

```
SELECT h.sha256, p.pid, p.name, CASE p.start_time WHEN -1 THEN
(time.unix_time-(uptime.total_seconds-p.start_time)) ELSE p.start_time END AS
execution_time, p.path,
u.uid_signed AS uid
FROM processes AS p, uptime, time
INNER JOIN hash AS h ON h.path = p.path
INNER JOIN users AS u ON u.uid = p.uid
WHERE h.sha256 <> "
AND p.pid = 9200
ORDER BY execution_time ASC;
```

Vulnerabilities and software updates

```
[Hughes-MacBook-Pro:~ hughneale$ osqueryi
Using a virtual database. Need help, type '.help'
osquery> SELECT name, bundle_name, bundle_version FROM apps ORDER BY last_opened_time DESC LIMIT 5;
+-----+-----+-----+
| name          | bundle_name      | bundle_version |
+-----+-----+-----+
| Preview.app   | Preview          | 944.6.16.1      |
| Activity Monitor.app | Activity Monitor |           |
| Calculator.app | Calculator        | 123            |
| VLC.app       | VLC media player | 3.0.7          |
| QuickTime Player.app | QuickTime Player | 935.3          |
+-----+-----+-----+
osquery>
```

Osquery Tables

apps, package_install_history,  signature

programs 

rpm_packages, deb_packages 

On Linux you can query to see what repos are being used

Which can show us what packages are installed

We can see this information across all assets and know what's available to update cross platform

Comparing versions

Reverse lookup against known hashes of packages (package extraction)

Pulling data from CVDB, exploit DB etc

Same can be done with NPM and python packages (if you have devs)

```
SELECT name, bundle_name, bundle_version FROM apps ORDER BY last_opened_time DESC LIMIT 5;
```

Vulnerabilities and software updates

Filename	Version	Owner / URL	Published
tcpdump_4.2.1~ubuntut2.1_amd64.deb cc91a5b173a0730ccb3cc32689db44dfb9cb0ba47b7514f77664code95c12	4.2.1~ubuntut2.1	Ubuntu Developers	9 years ago
tcpdump_4.2.1~ubuntut2.2_amd64.deb 0de99309a4d474a620209b0879977b7b1e0d9780fcac729abb91483ce8e9333	4.2.1~ubuntut2.2		
tcpdump_4.5.1~ubuntut1.1_amd64.deb b349ef55050873dc50b0767018b938779c0dbd97386b679a4047cff598226c	4.5.1~ubuntut1.1		
tcpdump_4.5.1~ubuntut1.1_amd64.deb 98a0e0146f814ec	4.5.1~ubuntut1.1_amd64		
tcpdump_4.7.4~ubuntut1.1_amd64.deb 627ad908f73aa4e	4.7.4~ubuntut1.1_amd64		
tcpdump_4.9.1~ubuntut1.1_amd64.deb 2206689871671dd9	4.9.1~ubuntut1.1_amd64		
tcpdump_4.9.1~ubuntut1.1_amd64.deb a7ca70694a2c3f7	4.9.1~ubuntut1.1_amd64		

High Risk
This package has been identified as vulnerable and carries a HIGH risk to your system if exploited. You should update the affected package immediately.

Package		
Name	Score	Risk
Vulnerability Risk	90 / 100	Critical
Age Risk	a year ago	High

Package Health		
Name	Score	Risk
Vulnerability Risk	60 / 100	High
Age Risk	5 years ago	Critical

Package Vulnerabilities		
Name	Published	CVSSv2
CVE-2017-12895 (CRITICAL)	14th Sep 2017	
CVE-2017-12897 (CRITICAL)	14th Sep 2017	
CVE-2017-13017 (CRITICAL)	14th Sep 2017	
CVE-2017-13037 (CRITICAL)	14th Sep 2017	
CVE-2017-12999 (CRITICAL)	14th Sep 2017	
CVE-2017-13051 (CRITICAL)	14th Sep 2017	
CVE-2017-11542 (CRITICAL)	23rd Jul 2017	
CVE-2017-11541 (CRITICAL)	23rd Jul 2017	
CVE-2017-11543 (CRITICAL)	23rd Jul 2017	
CVE-2017-11108 (HIGH)	8th Jul 2017	

Other Versions		
Version	Published	Risk
1.30+dfsg-6	24th Apr 2019	
1.30+dfsg-5	4th Feb 2019	
1.30+dfsg-3	20th Nov 2018	
1.30+dfsg-2	16th May 2018	
1.30+dfsg-1	13th May 2018	

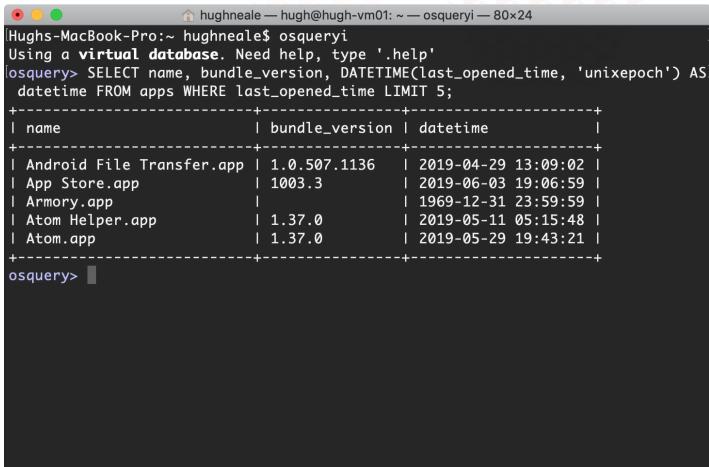


11

This is TCP dump issue

Note that we've been able to group other similar packages and suggest updates

Finding unused software licenses



A screenshot of a macOS terminal window titled "hughneale — hugh@hugh-vm01: ~ — osqueryi — 80x24". The window displays the output of an osquery query. The command run was ".help" followed by "SELECT name, bundle_version, DATETIME(last_opened_time, 'unixepoch') AS datetime FROM apps WHERE last_opened_time LIMIT 5;". The resulting table shows the following data:

name	bundle_version	datetime
Android File Transfer.app	1.0.507.1136	2019-04-29 13:09:02
App Store.app	1003.3	2019-06-03 19:06:59
Armory.app		1969-12-31 23:59:59
Atom Helper.app	1.37.0	2019-05-11 05:15:48
Atom.app	1.37.0	2019-05-29 19:43:21

- Who has Microsoft office installed and when was the last time they used it?
- Use binary hashes to detect version from db
- Fuzzy match version information

Mac helpfully shows you the last time that application ran

On linux and windows you'll have to store all the process information yourself to work out who ran what and when

```
SELECT name, bundle_version FROM apps ORDER BY last_opened_time DESC  
LIMIT 5;
```

```
SELECT name, bundle_name, bundle_version, DATETIME(last_opened_time,  
'unixepoch') AS datetime FROM apps WHERE last_opened_time > date('now', '-1  
month') ORDER BY last_opened_time DESC LIMIT 5;
```

IAM & AAA

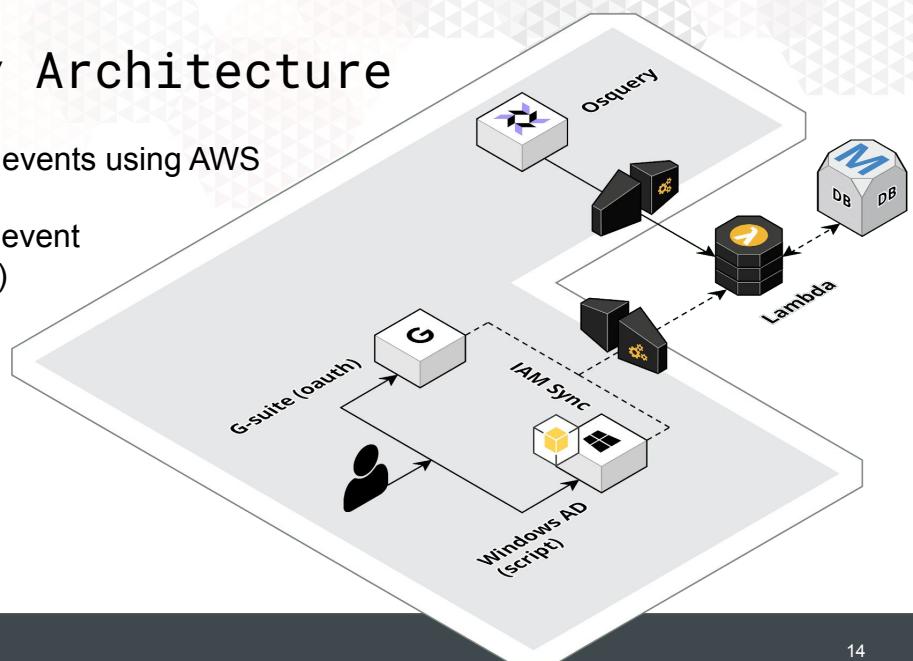
Identity & Access Management
Authentication Access &
Authorisation

Enriching data with Google,
Active Directory

Tracking users across machines
down to the commands
they execute.

Zercurity Architecture

- Processing of events using AWS serverless
- Workflows for event triggers (SWF)
- Sync between IAM repos



SSH keys

```
hughneale@Hughes-MacBook-Pro:~ hughneale$ osqueryi
Using a virtual database. Need help, type '.help'
osquery> SELECT * FROM user_ssh_keys;
+-----+-----+
| uid | path           | encrypted |
+-----+-----+
| 501 | /Users/hughneale/.ssh/example | 0         |
| 501 | /Users/hughneale/.ssh/id_rsa   | 1         |
| 501 | /Users/hughneale/.ssh/id_rsa_zercurity | 1         |
+-----+-----+
osquery>
```

Osquery Tables

user_ssh_keys



authorized_keys



Monitoring for non-encrypted
keys `SELECT * FROM
authorized_keys;`

Quick note on checking for non-encrypted keys

Also the authorized_keys table to check what access servers have

```
SELECT * FROM user_ssh_keys;
SELECT * FROM authorized_keys;
```

SIEM

```
hughneale — hugh@hugh-vm01: ~ — ssh -p 5226 hugh@81.98.53.22 — 80x24
osquery> SELECT username, p.name AS process,
...> DATETIME(time, 'unixepoch') AS datetime, host FROM last AS l
...> LEFT JOIN processes AS p ON p.pid = l.pid
...> WHERE host <> '' AND host NOT LIKE ':pts%'
...> ORDER BY time DESC LIMIT 10;
+-----+-----+-----+
| username | process | datetime           | host      |
+-----+-----+-----+
| hugh    | sshd    | 2019-06-13 09:46:06 | 192.168.1.1
| hugh    |          | 2019-06-12 17:28:55 | 192.168.1.1
| hugh    |          | 2019-06-11 11:36:06 | 192.168.1.1
| hugh    | gdm-x-session | 2019-06-09 07:36:29 | :1
| hugh    |          | 2019-06-09 07:32:43 | 192.168.1.1
| runlevel | migration/7 | 2019-06-08 18:49:39 | 4.15.0-51-generic
| reboot   |          | 2019-06-08 18:49:30 | 4.15.0-51-generic
| hugh    |          | 2019-06-02 16:41:51 | 192.168.1.1
+-----+-----+-----+
osquery>
osquery>
osquery>
osquery>
osquery>
osquery>
osquery>
```

Osquery Tables

last



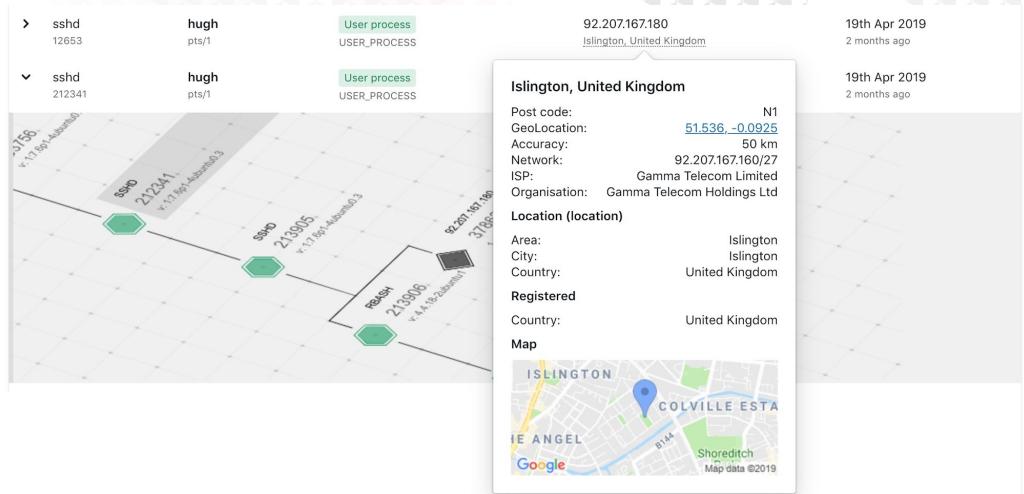
Note: that you can **only** JOIN against running processes

Processes that no longer exist cannot be joined.

Collecting all the pid information offline can help build a better picture.

```
SELECT username, p.name AS process,
DATETIME(time, 'unixepoch') AS datetime, host FROM last AS l
LEFT JOIN processes AS p ON p.pid = l.pid
WHERE host <> '' AND host NOT LIKE ':pts%'
ORDER BY time DESC LIMIT 10;
```

SIEM



EDR

Syslog, ASL & Windows Events

```
↑ hughneale — hugh@hugh-vm01: ~ — osquery — 80x24
[Hughs-MacBook-Pro:~ hughneale$ osqueryi
Using a virtual database. Need help, type '.help'
osquery> SELECT DATETIME(time, 'unixepoch') AS datetime, facility, level, SUBSTR(message, 0, 20) AS message FROM asl ORDER BY time DESC LIMIT 10;
+-----+-----+-----+
| datetime | facility | level | message
+-----+-----+-----+
| 2019-06-13 10:31:18 | user | 3 | objc[3525]: Attempt |
| 2019-06-13 10:30:18 | user | 3 | objc[3525]: Attempt |
| 2019-06-13 10:29:18 | user | 3 | objc[3525]: Attempt |
| 2019-06-13 10:28:18 | user | 3 | objc[3525]: Attempt |
| 2019-06-13 10:27:18 | user | 3 | objc[3525]: Attempt |
| 2019-06-13 10:26:18 | user | 3 | objc[3525]: Attempt |
| 2019-06-13 10:26:18 | com.apple.asl.statistics | 5 | ASL Sender Statistics |
| 2019-06-13 10:25:18 | user | 3 | objc[3525]: Attempt |
| 2019-06-13 10:24:18 | user | 3 | objc[3525]: Attempt |
| 2019-06-13 10:23:18 | user | 3 | objc[3525]: Attempt |
+-----+-----+-----+
osquery>
```

Osquery Tables

syslog_events [#4810](#)



asl



windows_events



- ToB extention to import
https://github.com/osql/extensions/tree/master/darwin_unified_log
- Someone please approve PR #4810
- ASL remove

SELECT * FROM asl ORDER BY time DESC LIMIT 10;

Osquery lets you observe
network sockets.

Wash data against OS
threat feeds

Compare and contrast
Netflow data

Networking

Because you need more events.

Network & sockets

```
[Hughes-MacBook-Pro:~ hughneale]$ osqueryi
Using a virtual database. Need help, type '.help'
osquery> SELECT processes.pid, processes.name, remote_address, remote_port FROM
process_open_sockets LEFT JOIN processes ON processes.pid = process_open_sockets
.pid WHERE remote_address <> '' AND remote_address != '::' AND remote_address !=
'127.0.0.1' AND remote_address != '0.0.0.0' AND remote_port = 443 LIMIT 10;
+-----+-----+-----+
| pid | name           | remote_address | remote_port |
+-----+-----+-----+
| 473 | Google Chrome Helper | 35.157.226.4 | 443
| 473 | Google Chrome Helper | 216.58.210.195 | 443
| 473 | Google Chrome Helper | 63.34.186.40 | 443
| 473 | Google Chrome Helper | 157.240.1.53 | 443
| 473 | Google Chrome Helper | 172.217.169.14 | 443
| 473 | Google Chrome Helper | 216.58.204.37 | 443
| 473 | Google Chrome Helper | 192.229.233.50 | 443
| 473 | Google Chrome Helper | 13.107.6.171 | 443
| 473 | Google Chrome Helper | 63.34.186.40 | 443
| 473 | Google Chrome Helper | 216.58.210.231 | 443
+-----+-----+-----+
osquery>
```

Osquery Tables

socket_events 

process_open_sockets 

listening_ports 

Wash data against open
source threat intel feeds.

- New Osquery has batching support to improve performance
- Don't mention trade offs - everyone knows

```
SELECT processes.pid, processes.name, remote_address, remote_port FROM
process_open_sockets LEFT JOIN processes ON processes.pid =
process_open_sockets.pid WHERE remote_address <> '' AND remote_address != '::'
AND remote_address != '127.0.0.1' AND remote_address != '0.0.0.0' AND
remote_port = 443 LIMIT 10;
```

Wifi Survey (Mac Only)

```
↑ hughneale — hugh@hugh-vm01: ~ — osqueryi — 80x24
[Hugh's MacBook-Pro:~ hughneale$ osqueryi
Using a virtual database. Need help, type '.help'
osquery> SELECT interface, channel, country_code FROM wifi_status;
+-----+-----+
| interface | channel | country_code |
+-----+-----+
| en0       | 40      | FR          |
+-----+-----+
osquery> SELECT interface FROM interface_details WHERE type = 71;
osquery> SELECT bssid, rssi, noise FROM wifi_survey LIMIT 5;
+-----+-----+
| bssid     | rssi   | noise    |
+-----+-----+
| 48:d3:43:0b:ee:df | -90  | 0        |
| b8:c1:a2:39:10:ec | -54  | 0        |
| d2:05:c2:a8:50:09 | -71  | 0        |
| c0:05:c2:a8:50:0f | -80  | 0        |
| c0:ff:d4:1f:b0:c0 | -80  | 0        |
+-----+-----+
osquery>
```

Osquery Tables

wifi_status	
interface_details	
wifi_survey	

Use the [Google geolocate](#) API with bssid, rssi, noise to get a lat, lng.

- Talk about the context of IAM
 - Helps tie together data from Google gsuite for mobile devices
 - Better visibility
 - Provides context around user access

SELECT interface FROM interface_details WHERE type = 71;

SELECT interface, channel, country_code FROM wifi_status;

SELECT bssid, rssi, noise FROM wifi_survey LIMIT 5;

Combining queries can let you monitor USB device file transfers.

DLP

Data loss prevention using FIM and hardware_events

Data loss protection (DLP)

```
hughneale — hugh@hugh-vm01:/etc/rsyslog.d — osqueryd • sudo — 80x24
Using a virtual database. Need help, type '.help'
osquery> SELECT action, DATETIME(time, 'unixepoch') AS datetime, vendor, mounts.path
FROM disk_events LEFT JOIN mounts ON mounts.device = disk_events.device;
+-----+-----+-----+
| action | datetime | vendor | path |
+-----+-----+-----+
| add   | 2019-06-13 14:57:01 | Innostor | /Volumes/HUGHLOU |
| add   | 2019-06-13 14:57:01 | Innostor |           |
+-----+-----+-----+
osquery>
osquery> SELECT action, uid, SUBSTR(target_path, 18) AS path, SUBSTR(md5, 0, 8)
AS hash, DATETIME(time, 'unixepoch') AS datetime FROM file_events WHERE sha1 <>
'' AND target_path NOT LIKE '%DS_Store';
+-----+-----+-----+-----+
| action | uid | path          | hash    | datetime      |
+-----+-----+-----+-----+
| CREATED | 99 | .Trashes        | d41d8cd | 2019-06-13 15:18:54 |
| CREATED | 99 | .Trashes/501     | d41d8cd | 2019-06-13 15:18:54 |
| CREATED | 99 | gozney-investment-deck.pdf | d41d8cd | 2019-06-13 15:18:56 |
| CREATED | 99 | gozney-investment-deck.pdf | ca4a434 | 2019-06-13 15:18:57 |
| UPDATED | 99 | gozney-investment-deck.pdf | ca4a434 | 2019-06-13 15:18:58 |
+-----+-----+-----+-----+
osquery>
```

Osquery Tables

usb_devices	 
file_events	 
hardware_events	 
mounts	 
disk_events	

```
{"file_paths": {
  "homes": [
    "/Volumes/%%"
  ]
}}
```



23

legal case

We hashed files from sharepoint and filenames can be tracked

Osquery config

```
sudo osqueryi --disable_audit=false --verbose --disable_events=false
```

```
SELECT action, DATETIME(time, 'unixepoch') AS datetime, vendor, mounts.path
FROM disk_events LEFT JOIN mounts ON mounts.device = disk_events.device;
```

```
Osq.conf = {"file_paths": {
  "homes": [
    "/Volumes/%%"
  ]
}}
```

```
sudo osqueryi --disable_audit=false --verbose --disable_events=false --config_path
./osq.conf
```

```
SELECT action, uid, SUBSTR(target_path, 18) AS path, SUBSTR(md5, 0, 8) AS
hash, time FROM file_events WHERE sha1 <> '' AND target_path NOT LIKE
'%DS_Store';
```

We've built thousands of
Compliance queries atop Osquery
for CIS, NIST, CE, ASD top 8.

Compliance

Use common frameworks or
write your own.

Query packs

We'll put out something on Github around this.

Compliance

CIS Debian 9

CIS-DEBIAN-9-100 // Pending

4.1.6

Ensure events that modify the system's network environment are collected.

Not enough data yet to score this test.

CIS Windows 10

(L1) Ensure 'MSS: (AutoAdminLogon) Enable Automatic Logon (not recommended)' is set to 'Disabled'

This setting is disabled from the Windows core feature in Windows XP and Vista. If this feature is enabled, this setting is not disabled. If you configure a computer for automatic logon, anyone who can physically gain access to the computer can also gain access to everything that is on the computer, including any network or networks to which the computer is connected. Also, if you enable automatic logon, the password is stored in the registry in plain text. This password is stored in the registry under the key that stores this value is remotely readable by the Authenticated Users group.

For additional information, see Microsoft Knowledge Base article 324737: How to turn on automatic logon in Windows.

The recommended state for this setting is Disabled.

Rationale

If you configure a computer for automatic logon, anyone who can physically gain access to the computer can also gain access to everything that is on the computer, including any network or networks that the computer is connected to. Also, if you enable automatic logon, the password is stored in the registry in plain text. This password is stored in the registry under the key that stores this value is remotely readable by the Authenticated Users group. As a result, this entry is appropriate only if the computer is physically secured and if you ensure that untrusted users cannot remotely see the registry.

CIS Ubuntu 18.04 LTS

CIS CentOS 7

Asset compliance status

Passing	Failing	Total
2	3	5

Assets

Asset	Result	Datetime
AD installation 1	PASSED	4 hours ago 11th Jun 2019, 13:10:51
AD installation 1	PASSED	11 hours ago 10th Jun 2019, 00:15:50
AD installation 2	PASSED	a day ago 10th Jun 2019, 11:22:40
AD installation 2	PASSED	3 hours ago 11th Jun 2019, 14:38:10
AD installation 2	PASSED	15 hours ago 10th Jun 2019, 02:08:10
AD installation 2	PASSED	a day ago 10th Jun 2019, 13:44:05

Osquery Tables

registry	
plist	
carves	
system_controls	
augeas	

Compliance

```
SELECT COUNT(*) AS passed FROM system_controls WHERE name = 'net.ipv4.tcp_syncookies' AND current_value = 0 AND config_value = 0;
+-----+
| passed |
+-----+
| 1      |
+-----+
SELECT COUNT(*) AS passed FROM registry WHERE key = 'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System' AND name = 'MaxDevicePasswordFailedAttempts' AND data = '1';
+-----+
| passed |
+-----+
| 0      |
+-----+
SELECT * FROM augeas WHERE path = '/etc/apache2/...'
```

Few examples of what we can run

Auditing

- Osquery provides a whole lot of EVENTED TABLES for monitoring and auditing system changes.

Osquery Tables		
user_events		
file_events		
process_file_events		
socket_events		

Osquery Tables		
hardware_events		
powershell_events		
disk_events		
process_events		
selinux_events		
syslog_events		
user_interaction_events		
yara_events		

Risk

Key changes

Below are some key changes Zercurity has observed over the last week

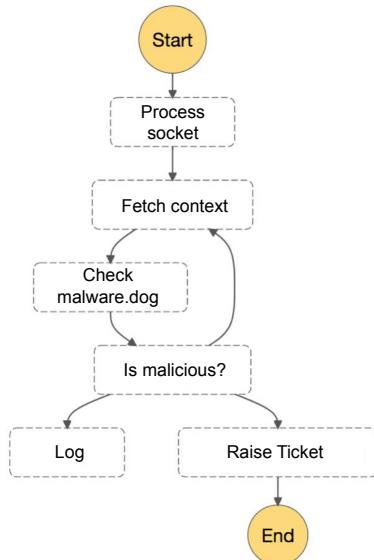
Critical	High	Medium	Low
318 No change	362 No change	312 No change	8 No change

Below is a summary of your top open issues. Issues are scored from 0 to 100. 100 being the most critical issue and in need of immediate attention.

Key new or critical outstanding issues	
100	ubuntu EOL has 8 known CRITICAL vulnerabilities (CVE-2016-4658, CVE-2016-4448, CVE-2016-4658, CVE-2016-4448, CVE-2016-4448, CVE-2016-4658, CVE-2016-4448, CVE-2016-4448).
100	Ubuntu test box (Vulnerable) has 4 known CRITICAL vulnerabilities (CVE-2016-4658, CVE-2016-4448, CVE-2016-4658, CVE-2016-4448).
98	ubuntu EOL has 110 known CRITICAL vulnerabilities (CVE-2017-2885, CVE-2017-7870, CVE-2016-9942, CVE-2016-9941, CVE-2013-7459, CVE-2017-7870, CVE-2016-9427, CVE-2017-7870, CVE-2016-10195, CVE-2017-2885). Though only showing 20.
98	Ubuntu test box (Vulnerable) has 180 known CRITICAL vulnerabilities (CVE-2018-1126, CVE-2018-1126, CVE-2015-8271, CVE-2016-9427, CVE-2016-10195, CVE-2017-12987, CVE-2017-13008, CVE-2017-13037, CVE-2017-12991, CVE-2017-12897). Though only showing 20.

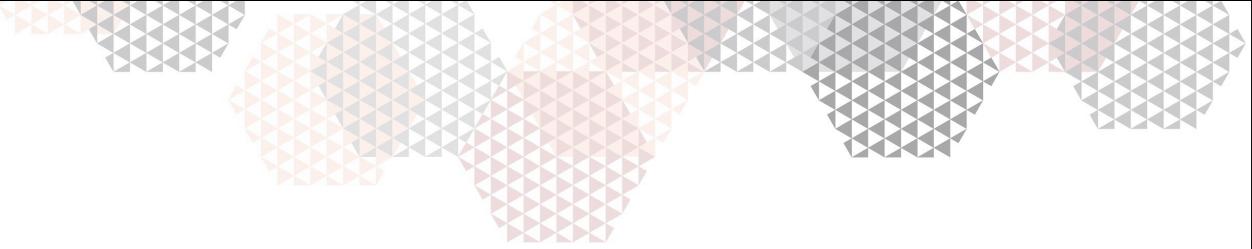
- Talk about assigning risk to different queries and effort
- Using enriched data per team to identify risks in the BUs
- Anon data to benchmarking in verticals
- SSO OKTA

Workflows



Using workflows to automate tasks and results that come in

- Notify a user
- Is this something the end user can address
- Interact with another service
- AWS Workflows (SWF)



Boom, that's it.

@hughneale / @zecurity

hugh@zecurity.com

End with how awesome Osquery is.
Much more than just SELECT statements.

You can build lots of things a top OSquery