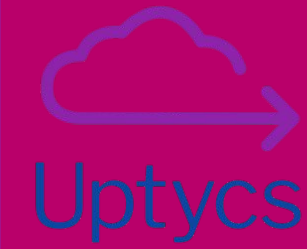
Defending Environments and Hunting Malware with Osquery



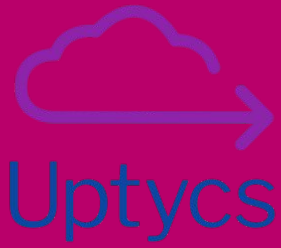
I'm Guillaume Ross

Uptycs - osquery analytics at scale

@gepeto42 on Twitter



gross @ uptycs.com



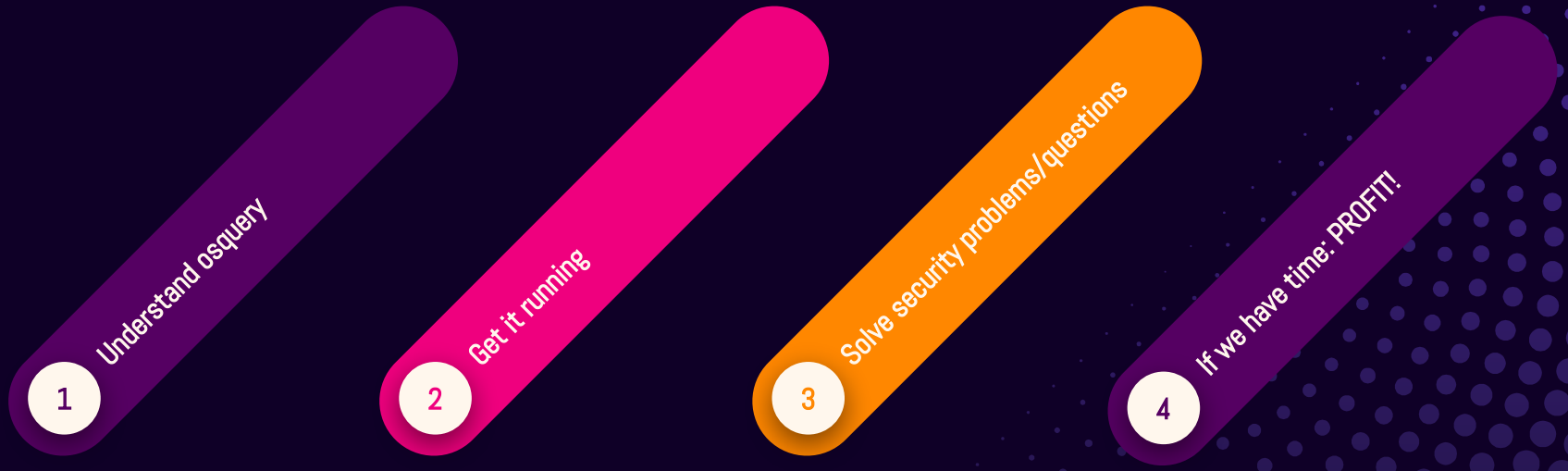
The guy who does not look like me is Julian Wayte

Uptycs - osquery analytics at scale

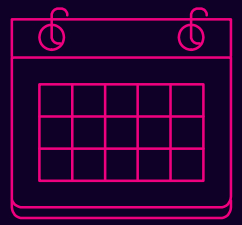
jwayte @ uptycs.com



What we'll do today



SCHEDULE



- **Hour 1:** Setup, a few use cases on Linux, feel free to try on Mac and Windows too.
- **Hour 2** - Joins, Events, monitoring, containers, Windows registry.
- **Hour 3** - FIM, Augeas, centralized logging
- **Hour 4:** Extensions, other fun use cases, detection, etc.

We will take short breaks between sections.

WHAT IS OSQUERY?



- Open Source agent (Maintained by Facebook & Community)
- Cross-platform (Mac, Linux, Windows, Docker Support)
- Turns requests for info into SQL tables you query with SQLite based syntax
  - Mostly read-only*

SQL



- https://www.sqlite.org/lang.html
  - We will not do very long or complex queries, learn the basics then you can iterate.
- Surprisingly(?), every french guide about SQL seems to be about wine and alcohol abuse:

Figure VII.1 : Création de la base Dégustation

```
VINS (NV, CRU, MILLESIME, DEGRE, QUALITE)
BUVEURS (NB, NOM, ADRESSE, TYPE)
ABUS (NB, NV, DATE, QUANTITE)

CREATE SCHEMA AUTHORIZATION DEGUSTATION
CREATE TABLE VINS (NV INT, CRU CHAR(12), MILLESIME INT, DEGRE DEC(3,1), QUALITE CHAR)
CREATE TABLE BUVEURS (NB INT, NOM CHAR(20), ADRESSE CHAR(30), TYPE CHAR(4))
CREATE TABLE ABUS(NB INT, NV INT, DATE DEC(6), QUANTITE SMALLINT)
```

HOW DOES IT WORK?



- Daemon runs scheduled “queries” or “query packs”.
- Results are usually human readable in pretty tables, or machine readable - usually JSON.
- Results logged locally, over syslog, TLS and more.
- Interactive version available - we will use this for the workshop!

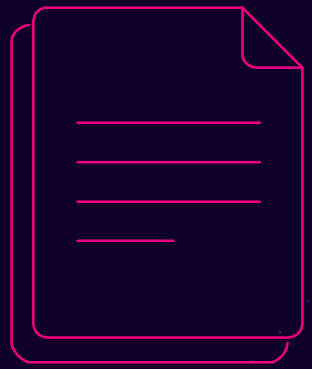
CONNECT



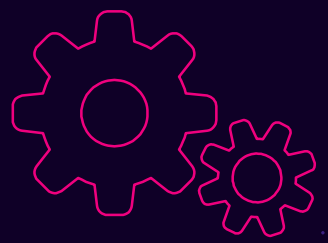
- SSH into your VM!
- Screen and tmux are preinstalled - or just open 3-4 SSH connections to it
- irssi is installed and preconfigured to join #osqueryworkshop on Freenode
- Join with any client you want from your laptop
- Good for copy pasting sample queries, asking questions, tag @gepeto or @jwayte if needed

WHAT FILES ARE INVOLVED?

- Binaries in /usr/bin/
  - osqueryi : interactive
  - osqueryd: daemon
- Config in /etc/osquery/
  - osquery.conf
  - osquery.flags
- DB in /var/osquery/osquery.db
- Logs in /var/log/osquery/osquery.INFO



EXAMPLE ABSTRACTION



- ps ax | grep httpd
- `SELECT * FROM processes WHERE name LIKE '%httpd%';`

No huge DB built over time on the agent. When queried, info is obtained (Also: Events stream).

```
"Conduit": {
  "query" : "select * from launchd where path like '%com.conduit.loader.agent.plist' or name = 'com.conduit.loader.agent.plist' or path like '%com.perion.searchprotectd.plist' or name =
  'com.perion.searchprotectd.plist';",
  "interval" : "3600",
  "version": "1.4.5",
  "description" : "(http://www.thesafemac.com/arg-conduit/)",
  "value" : "Artifact used by this malware"
},
"Genieo": {
  "query" : "select * from launchd where \
    name = 'com.genieo.completer.download.plist' OR \
    name = 'com.genieo.completer.update.plist' OR \
    name = 'com.genieo.completer.ltvbit.plist' OR \
    name = 'com.installer.completer.download.plist' OR \
    name = 'com.installer.completer.update.plist' OR \
    name = 'com.installer.completer.ltvbit.plist' OR \
    name = 'com.genieoinnovation.macextension.plist' OR \
    name = 'com.genieoinnovation.macextension.client.plist' OR \
    name = 'com.genieo.engine.plist';",
  "interval" : "3600",
  "version": "1.4.5",
  "description" : "(http://www.thesafemac.com/arg-genieo/)",
  "value" : "Artifact used by this malware"
},
"GenieoPart2": {
  "query" : "select * from launchd where program_arguments like '/Users/%/Library/Application Support/%/%.app/Contents/MacOS/App% -trigger download -isDev % -installVersion % -firstAppId %
  -identity %';",
  "interval" : "3600",
  "version": "1.4.5",
  "description" : "New version of Genieo",
  "value" : "Artifact used by this malware"
},
"HackingTeam_Mac_RAT1": {
  "query" : "select * from file where path = '/dev/ptmx0';",
  "interval" : "3600",
  "version": "1.4.5",
  "description" : "Detect RAT used by Hacking Team",
  "value" : "Artifact used by this malware"
},
"HackingTeam_Mac_RAT2": {
  "query" : "select * from apps where bundle_identifier = 'com.ht.RCSMac';",
  "interval" : "3600",
  "version": "1.4.5",
  "description" : "Detect RAT used by Hacking Team",
  "value" : "Artifact used by this malware"
},
```

PIN THIS!

https://osquery.io/schema/



Without a browser:

.tables / .schema $table

LAB #1!

Let's get you in *osqueryi*

> You can SSH into the VM I provided, or you can install the latest osquery on your system. As long as you can run *osqueryi*, it's fine!

INSTALLING

- Best option: SSH into VMs provided for Linux
- Mac and Linux: Get PKG from osquery.io
  - Linux: sudo dpkg -i package.deb (it's in files/ in learnosquery home on the Linux boxes provided)
- Windows: Chocolatey is easiest for lab
  - choco install osquery

You do not need all 3 - many exercises work on multiple.

Have at least one Linux system working - remote one is fine if Wi-Fi survives.

TESTING

- Mac / Linux: just run *osqueryi* in terminal.
- Windows: Run *osqueryi* from C:\ProgramData\osquery in command line. (it might not be in your path)
- `SELECT * FROM uptime;`

Let's get this party started!!



USEFUL COMMANDS

- `.help`
- Quitting is .exit or .quit - this ain't vi!
- `.mode line`
- Try: `SELECT * FROM users;` **(shortcut: .all users)**
- `.mode pretty`
- `SELECT * FROM processes;`
- On Linux: `SELECT * FROM shadow;` - any issue?
- Check out: `.features` and `.show`

AT SCALE

Pick Queries

Using *osqueryi* and the schema, determine queries or query packs to be run periodically.

Pick Channel

Do you already have a centralized ELK? Graylog? Splunk?

Do you need to support workstations out of the office?

Alerting

Create alerts for specific queries

Hunting

TLS allows real-time queries

Logging to files/ELK: Harder to change configs on the fly

STITCHING IT TOGETHER

Deployment + Config

- Chef
- Puppet
- SCCM
- Munki
- etc.

Management

- Zentral
- Kolide Fleet
- Uptycs
- Doorman

Data Bus

- Kafka
- Logstash/Beats
- Uptycs
- etc.

Storage

- Elastic
- Postgres
- Uptycs
- etc.

QUERY PACKS?

A collection of queries to schedule

`/usr/share/osquery/packs` - **look at incident-response.conf**

More readable:

`https://github.com/osquery/osquery/tree/master/packs`

QUERY PACKS?

Many already available!

- sudo cp osquery.example.conf /etc/osquery/osquery.conf
- Edit with vi/nano/whatever
- Uncomment the packs - make sure the last one uncommented does not end with a comma
- sudo systemctl start osqueryd
- sudo tail -f /var/log/osquery/osqueryd.results.log

QUERY PACKS?

Do you see empty results?

```
==> /var/log/osquery/osqueryd.INFO.20190730-180056.31039 <==
I0730 18:03:52.157404 31049 scheduler.cpp:100] Executing scheduled query pack_ossec-rootkit_monkit: select * from file where path in ('/lib/defs');
I0730 18:03:52.176695 31049 query.cpp:106] Storing initial results for new scheduled query: pack_ossec-rootkit_monkit
```

SQL!

```
SELECT * FROM users;

SELECT * FROM users LIMIT 1;

SELECT COUNT(*) FROM users;

SELECT uid, gid, username, description,
directory FROM users;
```

SQL!

What's in the description field for **learnosquery?**

It's the Full Name, Room Number, Work Phone, Home Phone, Other.

Nobody uses UNIX accounts as a directory but...

SQL!

We can split results in different fields at least.

Hint: Select only the username and description, try to extract the first value of the description using SPLIT()

SQL!

```
SELECT username, split(description, ',',0) AS
describe1 FROM users;
```

Can someone explain how it works?

How it's a tad inaccurate?

SLIGHTLY MORE SQL!



SQL - Ordering

```
SELECT uid, gid, username, description,
directory FROM users;
```

How would you list users in alphabetical descriptions?

Reminder: https://www.sqlite.org/lang.html

Worth pinning for the next 3 hours!

SQL - Ordering

```
SELECT uid, gid, username, description,
directory FROM users ORDER BY description;
```

How would you sort in ASCENDING or DESCENDING order?

> OK but surely we won't always just be selecting the entire contents of a table right??
>
> - All y'all

SQL - WHERE and LIKE

Pick your own table and try a WHERE, then a LIKE with a wildcard (%)

SQL - WHERE and LIKE

LIKE makes strings case-insensitive where = does not.

Mac:

```
SELECT * FROM apps WHERE name LIKE 'microsoft%';
SELECT * FROM apps WHERE name='Microsoft Word.app';
```

Linux:

```
SELECT * FROM deb_packages WHERE name LIKE 'PyThOn%';
```

Windows:

```
SELECT * FROM programs WHERE name LIKE '%Google%';
```

DATES and MATH

- **Dates/times can be formatted**
- `SELECT local_time FROM time;`
  - `SELECT datetime(local_time, 'unixepoch', 'localtime') AS formatted_time FROM time;`
- **We can do math too - keep it in mind in case you need to convert units (KB/MB/etc.) or add values.**

DATES and MATH

```
SELECT path, type, blocks_available, blocks_size FROM
mounts WHERE path = ‘/’;
```

```
osquery> SELECT path, type, blocks_available, blocks_size FROM mounts WHERE path = '/';
+------+------+-----------------+-------------+
| path | type | blocks_available | blocks_size |
+------+------+-----------------+-------------+
| /    | ext4 | 5628141         | 4096        |
+------+------+-----------------+-------------+
```

I SUCK AT MATH BUT...

It seems we can multiply blocks by their size, then convert into a more readable format than bytes?

```
[osquery> SELECT path, type, blocks_available, blocks_size FROM mounts WHERE path = '/';
+------+------+-----------------+-------------+
| path | type | blocks_available | blocks_size |
+------+------+-----------------+-------------+
| /    | ext4 | 5628141         | 4096        |
+------+------+-----------------+-------------+
```

I SUCK AT MATH BUT...

```
SELECT path, type, round((blocks_available *
blocks_size*10e-10),2) AS gigs_free FROM mounts WHERE
path = '/';
```

```
osquery> SELECT path, type, round((blocks_available * blocks_size*10e-10),2) as gigs_free from mounts where path = '/';
+------+------+-----------+
| path | type | gigs_free |
+------+------+-----------+
| /    | ext4 | 23.05     |
+------+------+-----------+
```


NO NEED TO...
POP A CALC
imgflip.com

BREAK!

This place better have coffee!

LAB #2

Joins, Events, monitoring, containers, Windows registry.

A LOT OF JOINTS

THIS SECTION IS BROUGHT TO YOU BY...



I LOAD TWO JOINS IN THE MORNING

Look at the processes and users table

processes

Details on processes running

users

Details on each user account

I LOAD TWO JOINS IN THE MORNING

Look at the processes and users table

```
SELECT * FROM processes LIMIT 1;
SELECT pid, name, cmdline FROM processes LIMIT 5;
```

I LOAD TWO JOINS IN THE MORNING

Why would we want to join those?

I LOAD TWO JOINS IN THE MORNING

Join on UID! - Give short names to tables when selecting them to make it easier to refer to them

```
SELECT p.pid, p.name, u.uid, u.username FROM processes p join users u
on u.uid=p.uid;
```

I LOAD TWO JOINS IN THE MORNING

Try with Shell History

As learnosquery

```
SELECT * FROM shell_history;
```

As root

```
SELECT * FROM shell_history;
```

I LOAD TWO JOINS IN THE MORNING

Try with Shell History

In interactive mode, by default, you get your own user's. This won't work for the daemon!

```
SELECT * FROM shell_history WHERE shell_history.uid IN (SELECT uid FROM
users);
```

I LOAD TWO JOINS IN THE MORNING

"Workstation" Example

Start by looking at the tables - you can replace chrome_extensions with shell_history if you don't have Chrome.

```
SELECT * FROM chrome_extensions;
SELECT * FROM users;
```

What matches? The UID.

```
SELECT * FROM chrome_extensions WHERE chrome_extensions.uid IN (SELECT uid FROM users);
```

I LOAD TWO JOINS IN THE MORNING

Proper join to see who has what extension

```
SELECT users.username, chrome_extensions.name,
chrome_extensions.description FROM users CROSS JOIN chrome_extensions
USING (uid);
```

username	name	description
gross	Slides	Create and edit presentations
gross	Docs	Create and edit documents
gross	Google Drive	Google Drive: create, share and keep all your stuff in one place.
gross	YouTube	
gross	uBlock Origin	Finally, an efficient blocker. Easy on CPU and memory.
gross	Sheets	Create and edit spreadsheets
gross	Postman	
gross	HTTPS Everywhere	Encrypt the Web! Automatically use HTTPS security on many sites.
gross	Google Docs Offline	Get things done offline with the Google Docs family of products.

I LOAD TWO JOINS AT NIGHT

What process isn't stored on disk yet is in memory?

```
SELECT processes.pid, users.username, processes.path,
processes.on_disk FROM processes LEFT JOIN users ON processes.uid =
users.uid;
```

Just want the suspicious stuff? IF_SUSPICIOUS=1? Almost!

```
SELECT processes.pid, users.username, processes.path,
processes.on_disk FROM processes LEFT JOIN users ON processes.uid =
users.uid where processes.on_disk = 0;
```

I LOAD TWO JOINS AT NIGHT

What tables require users?

- Account_policy_data
- Authorized_keys
- Browser_plugins
- Chrome_extensions
- Firefox_addons
- Known_hosts
- Opera_extensions
- Safari_extensions
- Shell_history



WE CAN EVEN DO THIS WITH WINDOWS REGISTRY!

..Stay tuned.

SSH KEYS

Public and Private Keys

With your group - find ways to find info about public and private SSH keys with osqueryi. You can upload your own SSH public keys to the VMs to generate more data if desired.

You can generate private keys too (do not put actual private keys on the lab!)

SSH KEYS

Public and Private Keys

Private Keys - useful to detect clear text ones.

```
SELECT user_ssh_keys.uid, user_ssh_keys.path, user_ssh_keys.encrypted
FROM user_ssh_keys LEFT JOIN users on user_ssh_keys.uid = users.uid;
```

Public keys - useful to know who can login to a system and match against known keys

```
SELECT * FROM users JOIN authorized_keys USING (uid);
```

EXPLORE

Let's take a few minutes to look at the schema
Look for “event” tables too.

QUICK!

- What packages are installed?
- What time is it?
- Who's logged in?
- What's the MD5 hash of /etc/shadow?

QUICK!

- What packages are installed?
  - `Apps, programs, APT, homebrew..`
- What time is it?
- Who's logged in?
- What's the MD5 hash of /etc/shadow?
  - `SELECT md5 FROM hash WHERE path = ‘/etc/shadow’;`

EVENTS

- Cached “real time” query: data gets stored constantly, cache is emptied when query runs.
- Must be enabled via a flag - or rather - the disable-events flag must be turned off.
- `osqueryi --audit_allow_config=true --audit_allow_sockets=true --audit_persist=true --disable_audit=false --events_expiry=1 --events_max=50000 --logger_plugin=filesystem --disable_events=false`

ARGH

YOU EXPECT US TO RUN THAT COMMAND WITH ALL THOSE FLAGS ALL THE TIME? WE'LL SPEND ALL DAY PRESSING UP TO FIND THE RIGHT COMMAND AGAIN!!!

EVENTS

Configure /etc/osquery/osquery.flags (it's in queries.txt or https://evil.plumbing/defcon27/osquery.flags)

```
--audit_allow_config=true
--audit_allow_sockets=true
--audit_persist=true
--disable_audit=false
--events_expiry=1
--events_max=50000
--logger_plugin=filesystem
--disable_events=false
```

Start osqueryi: `sudo osqueryi --flagfile /etc/osquery/osquery.flags`

EVENTS

Does anyone have osquery running on their laptop or in a VM with USB support?

```
SELECT * FROM hardware_events;
```

Insert USB stick

Run it again. Hey, can this help for our DLP needs?

EVENTS

SELECT * FROM user_events;

Windows: Grab Windows Events!

`--windows_event_channels=System,Application,Setup,Security`

You can then query for **only specific event IDs. Perfect for centralizing workstation logs!**

(there's a similar syslog feature for other OSes)

EVENTS..ABOUT FILES!

SELECT * FROM file_events;

Wait, that's empty?

Hmmm. Perhaps we need to be more specific. We'll see it in the next hour!

EVENTS

- ⊗ Try to select from process_events table
- ⊗ What happens?

EVENTS

- Empty - because no process has run since you started osqueryi with events enabled!
- Start a 2nd SSH window (or screen etc)
  - ping 8.8.8.8
  - Top
- Check process_events and socket_events
- Query it again…. Still empty?

EVENTS

- *User_events*
- *Hardware_events* (won't work in our cloud VMs but you can try it on your laptop with a USB stick)
- Use the *osquery_events* table to see what events are active

GRAB YOUR WINDOWS!

- Any Windows VM will do.
- If you don't have one, most contexts work on other OSes except the Registry stuff. Sit with someone who's got one.



Windows in Enterprise: Big vectors

- Lateral Movement
- Macros from Internet
- Browser related issues



Osquery on Windows

- Schema has the list of Windows tables
- *Registry* is a crazy source of information.



How can we...

ensure our systems do not exhibit lateral movement issues?

Without reconfiguring the machine itself, focus on detecting good/bad config.



445 + 3389

Mac and Linux : Use port 22 for exercise

WinRM would be nice to have on Windows too.

Do we need to bother with FW Rules?

Perhaps tracking activity is enough?

Do we need to bother with FW Rules?

Perhaps tracking activity is enough?

process_open_sockets

Now that's interesting...

```
SELECT *
FROM process_open_sockets
WHERE local_port=(your port(s)
here);
```



```
SELECT *
FROM process_open_sockets
WHERE local_port=(your port(s)
here)
AND remote_address NOT LIKE
‘0.0.0.0’ (WHITELIST HERE);
```

LAPS / Registry

LAPS is configured in the registry.

Tell me how to track if the password is rotated more than every month.

Download a .reg to simulate LAPS being installed:

http://evil.plumbing/defcon27/LapsSettings.reg.zip

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft Services\AdmPwd

LAPS / Registry

Start with this..

```
SELECT data, path FROM registry
WHERE key =
'HKEY_LOCAL_MACHINE\Software\Policies\Microsoft
Services\AdmPwd\';
```

```
    ...> AND name='PasswordLength'
    ...> AND data < 31;
+------+-----------------------------------------------------------------------------+
| data | path                                                                        |
+------+-----------------------------------------------------------------------------+
| 14   | HKEY_LOCAL_MACHINE\Software\Policies\Microsoft Services\AdmPwd\PasswordLength |
+------+-----------------------------------------------------------------------------+
osquery>
```

Or we can track a specific value as a maximum:

```
SELECT data, path FROM registry
WHERE key LIKE
'HKEY_LOCAL_MACHINE\Software\Policies\Microsoft
Services\AdmPwd'
AND name='PasswordLength'
AND data < 31;
```

BROWSERS

1) Is Chrome or Firefox installed?
2) Does every user have uBlock Origin in Chrome?

Bonus questions:

- In a real world scenario I would add: Is IE configured to send all requests except whitelisted sites to Edge. How do we confirm?
- How do we hunt down potentially bad Chrome extensions?

MACROS FROM INTERNET

- How do we tell what Office app will allow the user to run a macro on a file downloaded from the Internet?
- Hint 1: The setting is in the registry
- Simulate from
  - **https://evil.plumbing/defcon27/officereg.zip**
- Hint 2: It's a per user setting.
- Hint 3: HKEY_USERS\[SID]\Software\Policies\Microsoft\office\16.0

SPLIT AND JOIN

- Take your Office query and map it to users.
- Hint: SPLIT And wildcards...

STEP 1

- What data are we looking for?
- Blockcontentexecution not set to 1 = dangerous

```
SELECT * FROM registry WHERE key LIKE
'HKEY_USERS\%\Software\Policies\Microsoft\office\16.0\%\security' AND name='blockcontentexecutionfrominternet' AND data!=1
```

STEP 2

- Who's who?
- That first wildcard actually corresponds to the user's SID!

```
SELECT * FROM registry WHERE key LIKE
'HKEY_USERS\%\Software\Policies\Microsoft\office\16.0\%\security' AND name='blockcontentexecutionfrominternet' AND data!=1
```

STEP 2

- SPLIT and JOIN on the subquery

```
SELECT username, data, split(path, '\', 1) AS sid
FROM
(SELECT data, path FROM registry
WHERE key LIKE 'HKEY_USERS\%\SOFTWARE\Policies\Microsoft\office\16.0\%\security' and name='blockcontentexecutionfrominternet')
JOIN users ON users.uuid = sid;
```

```
+----------+------+-----------------------------------------------+
| username | data | sid                                           |
+----------+------+-----------------------------------------------+
| g        | 1    | S-1-5-21-2940226973-2973024380-1756164060-1001 |
+----------+------+-----------------------------------------------+
osquery>
```

QUICK!

- How would you find out who's using what Chrome extension(s)?
- How would you go about finding specific vulnerable Chrome extensions? (Say we know the current Webex version had a Oday used in the wild)
- Bonus: On a real machine, check this registry path to see files where macros where enabled manually:

```
HKEY_USERS\%\Software\Microsoft\Office\%\Security\Trusted Documents\TrustRecords
```

BACK TO LINUX!

CONTAINER TIME

A FEW DOCKER BEST PRACTICES

- Inventory containers
- Know what ports you're exposing (host/guest)
- Never run privileged
- Tweet "C O N T A I N E R S" to @**joeynoname**

If you are using your own Linux VM, install Docker:

https://docs.docker.com/install/linux/docker-ce/ubuntu/

SSH VM have it already.

A FEW DOCKER BEST PRACTICES

Start a container in one terminal:

```
sudo docker run -it ubuntu bash
```

Start a second and third one:

```
sudo docker run --rm -it --security-opt
seccomp=unconfined debian:jessie
```

```
sudo docker run --rm -it --privileged
--security-opt seccomp=unconfined debian:jessie
```

A FEW DOCKER BEST PRACTICES

```
SELECT id, privileged, security_options,
cgroup_namespace FROM docker_containers;

SELECT * FROM docker_container_processes WHERE id
= '[one container ID here]';
```

LAB 3

FIM - Augeas - Centralized Logging -

File Integrity Monitoring

EVENTS..ABOUT FILES!

Edit osquery.conf

This is where FIM is controlled - or it can be in query packs.

You can paste this below the query packs section:

```
  "file_paths": {
    "homes": [
      "/root/.ssh/%%",
      "/home/%/.ssh/%%"
    ],
    "etc": [
      "/etc/%%"
    ],
      "home": [
      "/home/%%"
    ],
      "tmp": [
      "/tmp/%%"
    ],
      "www": [
      "/var/www/%%"
      ]
},
```

https://evil.plumbing/defcon27/fim.txt

EVENTS..ABOUT FILES!

Start osqueryi with the config and flag file:

sudo osqueryi --config-path /etc/osquery/osquery.conf --flagfile /etc/osquery/osquery.flags

SELECT * FROM file_events;

Then, go create a file somewhere in those paths and query again.

EVENTS..ABOUT FILES!

1) Great, easy way to configure FIM.
2) Hashes can be piped into anything, correlated with threat intel etc.
3) Don't go wild and try to monitor the entire file system with osqueryd :)

IOCs

IOCs

- First, what connects where?

```
select * from socket_events;
```

IOCs

- Open socket connections to specific remote machines can be found with SQL of the form:
  - `SELECT * FROM socket_events WHERE remote_address in ('A', 'B', 'C');`
  - `Use the IP of the IRC server you're on - or ping 8.8.8.8 in another terminal and try`
- You can search for specific command line processes using SQL of the form:
  - SELECT pid, parent, uid, cmdline FROM processes WHERE cmdline LIKE '%ping%';

IOCs

```
osquery> SELECT pid, local_address, remote_address, state FROM process_open_sockets WHERE remote_address in
 ('65.52.17.132', '54.165.17.209', '52.191.194.66');
+-----+---------------+----------------+------------+
| pid | local_address | remote_address | state      |
+-----+---------------+----------------+------------+
| -1  | 134.209.58.230 | 54.165.17.209  | TIME_WAIT |
| -1  | 134.209.58.230 | 54.165.17.209  | TIME_WAIT |
+-----+---------------+----------------+------------+
```

```
osquery> SELECT pid, parent, uid, cmdline FROM processes WHERE cmdline LIKE '%54.165.17.209%';
+------+--------+------+-----------------------------------+
| pid  | parent | uid  | cmdline                           |
+------+--------+------+-----------------------------------+
| 2511 | 2504   | 1017 | netcat -q 30 54.165.17.209 4443 |
+------+--------+------+-----------------------------------+
```

IOCs - Parent Process

- Select * from processes;
- Subqueries are your friends
- Select the processes table, for specific process names, look at the parent, see what the parent is, log when unexpected. For example

```
osquery> select pid, name, path, parent from processes where name='services.exe'
    ...> ;
+-----+--------------+------+--------+
| pid | name         | path | parent |
+-----+--------------+------+--------+
| 568 | services.exe |      | 476    |
+-----+--------------+------+--------+
```

Parent Process

What's 476? Wininit. Expected!

```
osquery> select pid, name, path, parent from processes where pid=476;
+-----+-------------+------+--------+
| pid | name        | path | parent |
+-----+-------------+------+--------+
| 476 | wininit.exe |      | 400    |
+-----+-------------+------+--------+
```

IOCs - Process Tree

- Can we visualize it a bit?

https://evil.plumbing/defcon27/pstree.txt

IOCs - Process Tree

2	10885	bash	systemd->screen->bash	0->1000->1000
2	17988	bash	systemd->screen->bash	0->0->0
2	2129	(sd-pam)	systemd->systemd->(sd-pam)	0->0->0
2	30491	(sd-pam)	systemd->systemd->(sd-pam)	0->1001->1001
2	30692	bash	systemd->screen->bash	0->1001->1001
2	4882	bash	systemd->screen->bash	0->0->0
2	16816	sshd	systemd->sshd->sshd	0->0->0
2	19208	sshd	systemd->sshd->sshd	0->0->0
2	20294	sshd	systemd->sshd->sshd	0->0->0
2	10953	containerd-shim	systemd->containerd->containerd-shim	0->0->0
2	20622	containerd-shim	systemd->containerd->containerd-shim	0->0->0
3	10897	sudo	systemd->screen->bash->sudo	0->1000->1000->0
3	30722	irssi	systemd->screen->bash->irssi	0->1001->1001->1001
3	4892	su	systemd->screen->bash->su	0->0->0->0
3	16918	bash	systemd->sshd->sshd->bash	0->0->0->0
3	19285	bash	systemd->sshd->sshd->bash	0->0->0->0
3	20393	bash	systemd->sshd->sshd->bash	0->0->0->0
3	10980	bash	systemd->containerd->containerd-shim->bash	0->0->0->0
3	20649	bash	systemd->containerd->containerd-shim->bash	0->0->0->0
4	10898	docker	systemd->screen->bash->sudo->docker	0->1000->1000->0->0
4	4893	bash	systemd->screen->bash->su->bash	0->0->0->0->1001
4	20235	osqueryi	systemd->sshd->sshd->bash->osqueryi	0->0->0->0->0->0
4	20578	sudo	systemd->sshd->sshd->bash->sudo	0->0->0->0->0->0
5	20579	docker	systemd->sshd->sshd->bash->sudo->docker	0->0->0->0->0->0->0

IOCs - Process Tree

- But what to monitor?
- Create queries for suspicious parents/child - ex: cmd.exe child of winword.exe - etc.

Parent Process

```
SELECT name as bad_parent_child_name, pid bad_parent_child_pid
FROM processes WHERE
pid=(SELECT parent FROM processes WHERE parent!=(SELECT pid from processes where name='wininit.exe')
AND LOWER(name)='services.exe')
OR pid=(SELECT pid FROM processes WHERE parent!=(SELECT pid from processes where name='wininit.exe')
AND LOWER(name)='services.exe');
```

This example + MANY more can be found on Filippo Mottini's GitHub:

https://bit.ly/2X9bAyy

Hashing Process Executables

- To match to threat feeds
- To make it easy to be sure it's the same process across hosts

1. The *hash* table is your friend
2. The *process_events* table contains all processes
3. Use DISTINCT if you want to eliminate duplicate values
4. Match on "path"

Hashing Process Executables

```
+-----------+------------------------------------------------------------------+
| path      | sha256                                                           |
+-----------+------------------------------------------------------------------+
| /bin/ping | dc4bb71234363d8f6eefc68f6dfd0f02596fff0ab514363691c03c35edbc1934 |
+-----------+------------------------------------------------------------------+
```

Take 5 minutes - collaborate in groups if needed!

Hashing Process Executables

```
+-----------+------------------------------------------------------------------+
| path      | sha256                                                           |
+-----------+------------------------------------------------------------------+
| /bin/ping | dc4bb71234363d8f6eefc68f6dfd0f02596fff0ab514363691c03c35edbc1934 |
+-----------+------------------------------------------------------------------+
```

```
SELECT path, md5 FROM hash WHERE path IN (SELECT
       DISTINCT path FROM process_events);
```

Hashing Process Executables

- Grab all hashes like we did and centralize them
- Look for specific malicious ones (less storage intensive, blacklist approach, not ideal but sometimes awesome during incidents)

```
SELECT path, sha256 FROM hash WHERE path IN (SELECT
DISTINCT path FROM process_events) AND sha256 in
(‘hash1’, ‘hash2’);
```

AUGEAS

AUGEAS

1) I don't know how to pronounce it.
2) Open source project of its own. http://augeas.net/
3) "Lens" are basically definitions of how config files are made (ex: sshd config file, sudoers file, etc).
4) We can leverage those lenses to read files as tables!

AUGEAS

```
SELECT * FROM augeas WHERE path='/etc/ssh/sshd_config';
```

INSERT MIND BLOWN DOT GIF. Infinite configuration tracking possibilities - in key:value format!

We use this a lot at Uptycs to build hardening reports! (CIS, FedRAMP, etc.)

AUGEAS

Try those:

- /etc/crontab
- /etc/hosts
- /etc/sudoers
- /etc/shadow

What’s an offensive use case you could think of with this?

EXTENSIONS

Extensions

Osquery is read-only - and by design does not include a kitchen sink.

What if we want one?

- Being able to control firewall rules
- Responding to incidents in general
- Configuring apps like Google Santa

Extensions

TrailOfBits has great extensions.

- Grab the binary: https://github.com/osql/extensions/releases
- Unzip
- Try it out ...
- `sudo osqueryi --allow-unsafe --extension /path/to/your_extension.ext`
- Make sure you have a ping to evil.plumbing running and working

Extensions

```
SELECT * FROM HostBLackList;

INSERT INTO HostBlacklist (domain, sinkhole,
address_type) VALUES ("evil.plumbing", "127.0.0.1",
"ipv4");
```

Try wget again. BOOM, sinkholed!

Check results:

```
SELECT * FROM HostBlackList;
```

EXTENSIONS

You can write your own.

Write extensions allow osquery to become a response agent too.

CENTRALIZED LOGGING

Centralized Logging & Management

Options:

- Log to syslog, let whatever you already have deal with it.
- Log to file, grab it with Filebeat, nxlog, Splunk, whatever.
- Log to a TLS endpoint.
  - Advantages: allows centralized configuration & real-time queries
  - Uptycs, Zentral, Kolide Fleet

Centralized Logging & Management

We'll do filebeat + graylog - start or restart osqueryd first.

- `sudo systemctl restart osqueryd`
- `sudo tail -f /var/log/osquery/osqueryd.results.log`
  - Stuff going in in JSON format? Good

Centralized Logging & Management

- Download Filebeat for Ubuntu
- sudo dpkg -i -i filebeat-7.3.0-amd64.deb
- Replace contents of /etc/filebeat/filebeat.yml with provided in queries.txt or https://evil.plumbing/defcon27/filebeat.yml.txt and **set the port to 4000 if your machine has an EVEN number or 4001 if your machine has an ODD number**

Centralized Logging & Management

- We could get the daemon running BUT for easier troubleshooting, in a separate SSH or screen:
  - `sudo filebeat -e -c filebeat.yml -d "publish"`
  - You can keep tailing osqueryd.results.log

Centralized Logging & Management

Make sure you extract the JSON from the message

From here - sky is the limit - Elastic and Splunk pros or just those better than me and with more time are gonna have fun!

LAB 4

Yara - CURL Tables - ATC - Misc

YARA

YARA

1) Must specify rules + paths to monitor in config
2) Events are available
3) Let's try on-demand scanning

YARA

1) Copy sample config from osquery documentation
   a) https://osquery.readthedocs.io/en/stable/deployment/yara/
2) Edit the sig_groups to point to
   a) Sig_group_1: /etc/osquery/yara/group1.sig and /etc/osquery/yara/group1b.sig
   b) Sig_group_2: /etc/osquery/yara/group2.sig
3) Download the 3 files from https://evil.plumbing/osquery/ and put them in the right places.

YARA

1) Copy sample config from osquery documentation
   a) https://osquery.readthedocs.io/en/stable/deployment/yara/
2) Edit the sig_groups to point to
   a) Sig_group_1: /etc/osquery/yara/group1.sig and /etc/osquery/yara/group1b.sig
   b) Sig_group_2: /etc/osquery/yara/group2.sig
3) Download the 3 files from https://evil.plumbing/osquery/ and put them in the right places.
4) Download the .docx file from and place it on your Linux box in /home/*user*/downloads/ - it has DDE enabled in it.

YARA

1) Of our 3 signatures, the third one is made to detect DDE.
2) Query all of the downloads folder for all rules. What happens?
3) Query all of the downloads folder**s** of all users. How?

RAPID FIRE

1) Curl + curl_certificate
    a) Monitor the cert of your Intranet/365/SSL VPN - great way to see how often people are getting MiTM'd!
2) Auto Table Construction
3) Wifi Survey

CURL_CERTIFICATE

Try it:

```
SELECT * FROM curl_certificate WHERE hostname
=’www.google.com:443’;
```

CURL_CERTIFICATE

```
osquery> select * from curl_certificate where hostname='www.google.com:443';
+--------------------+------------------------------+--------------+-------------------+-----------------+------------------------------+
| hostname           | common_name                  | organization | organization_unit | serial_number   | issuer_common_name           |
+--------------------+------------------------------+--------------+-------------------+-----------------+------------------------------+
| www.google.com:443 | redirector.to.captive.portal | RedirectCO   | CaptivePortal     | 951F804930C976B1 | redirector.to.captive.portal |
+--------------------+------------------------------+--------------+-------------------+-----------------+------------------------------+
osquery>
```

ATC

Blog post by Kolide:
https://blog.kolide.com/build-custom-osquery-tables-using-atc-ab112a30674c

Ability to generate osquery tables from SQLite Databases. Interesting to parse all those DBs on macOS.

ATC

```
SELECT * FROM curl_certificate where
hostname='www.canada.ca:443';
```

Add “and issuer_organization NOT LIKE ‘%your CA%’ to get reports when MiTM is occuring.

Curl Table can be used with other queries, to integrate with APIs, etc.

WIFI_SURVEY

```
SELECT * FROM wifi_survey;
```

Mac only.

Why would we want this?

THANKS!

Get in touch!



@gepeto42 on Twitter

osquery-slack.herokuapp.com

Uptycs Blog:
https://www.uptycs.com/blog