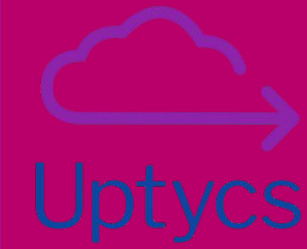
# Defending Environments and Hunting Malware with osquery




# I'm Guillaume Ross

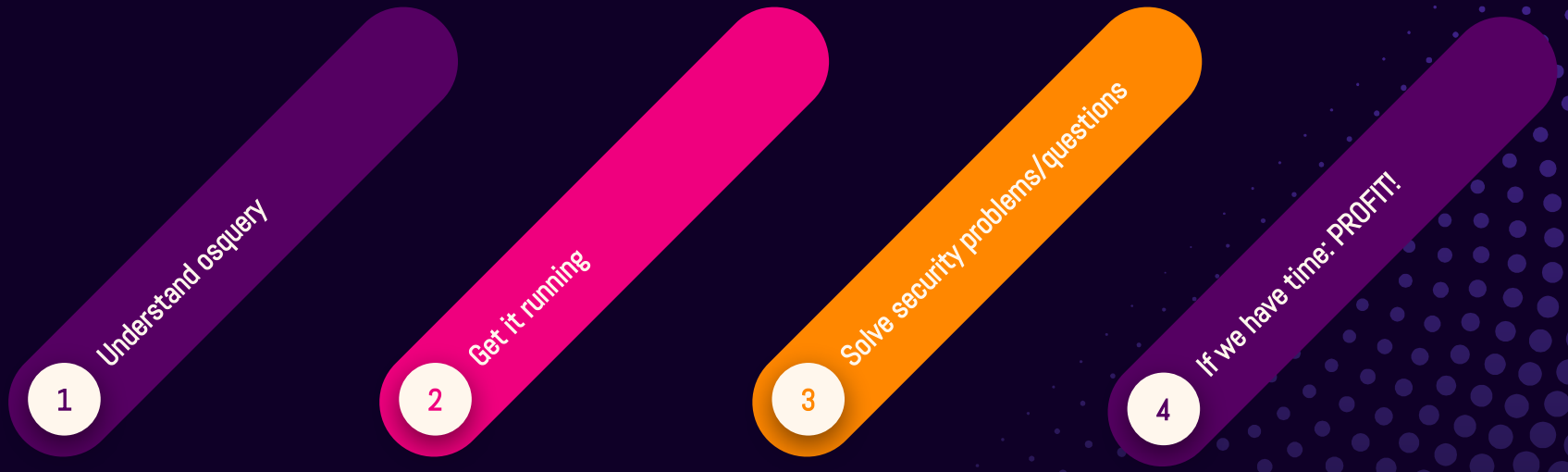
Uptycs - osquery analytics at scale

@gepeto42 on Twitter

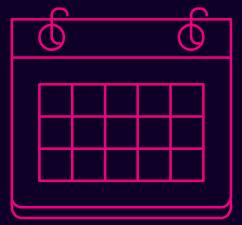


*gross* @ uptycs.com

# 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** - Events, configuration monitoring, containers, Windows registry.
- **Hour 3** - Augeas, more advanced use cases, extensions.
- **Hour 4:** Hunting, malware, fun!

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)
```

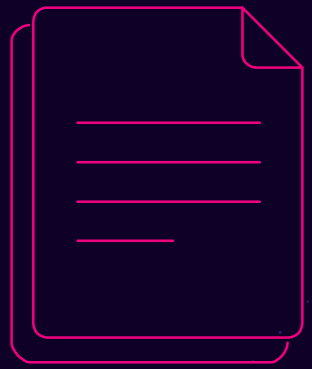
# HOW DOES IT WORK?



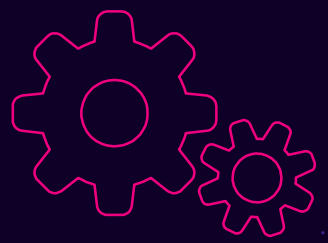
- Daemon runs scheduled “queries” or “query packs”.
- Results are usually human readable in pretty tables, or machine readable - usually JSON.
- Results logged locally, over syslog, TLS and more.
- Interactive version available - we will use this for the workshop!

# 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 apache
- `SELECT * FROM processes WHERE name LIKE '%apache%';`

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* and have admin privileges, it's fine!

# INSTALLING

- Best option: SSH into VMs provided for Linux
- Just install Ubuntu or CentOS
- Mac and Linux: Get PKG from osquery.io
  - Linux: dpkg -i package.deb
- Windows: Chocolatey is easiest for lab (or wait for RDP VM)
  - 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!!



# RUN OSQUERYI

- `.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 on the daemon, usually for a specific purpose.

Check out `/usr/share/osquery/` on Linux

Or `/var/osquery/packs` on Mac

Or `C:\ProgramData\Osquery\share\osquery\packs`

# QUERY PACKS?

Many already available!

Malware, compliance, hardening, performance…

Uncomment a few in osquery.conf

```
systemctl start osqueryd
```

# SQL!

```
SELECT * FROM users;

SELECT * FROM users LIMIT 1;

SELECT COUNT(*) FROM users;

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

# 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

# “ WHERE would you LIKE to go Today?

- SQL'ified Microsoft tagline from 1994

# 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.**

# I LOAD TWO JOINS IN THE MORNING

**If you don't have Chrome replace chrome_extensions with shell_history**

**chrome_extensions**

Per-user list of Chrome Extensions

**users**

List of users, including domain accounts on Windows.

# I LOAD TWO JOINS IN THE MORNING

Cross-platform 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?

# I LOAD TWO JOINS IN THE MORNING

Cross-platform 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 or shell command

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

```
SELECT users.username, shell_history.time, shell_history.command FROM
users CROSS JOIN shell_history USING (uid);
```

# 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;
```



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!)

Requires Mac or Linux, will not work on Windows.

# 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 authorized_keys.uid, authorized_keys.key_file FROM
authorized_keys LEFT JOIN users ON authorized_keys.uid = users.uid;
```

## SSH KEYS

# What would you constantly monitor now that you understand user_ssh_keys and authorized_keys?

# BREAK!

This place better have coffee!

# EXPLORE

Let's take a few minutes to look at the schema

# 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’;`

# LAB #2!

Events, Config Mgmt, Containers!

# 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.



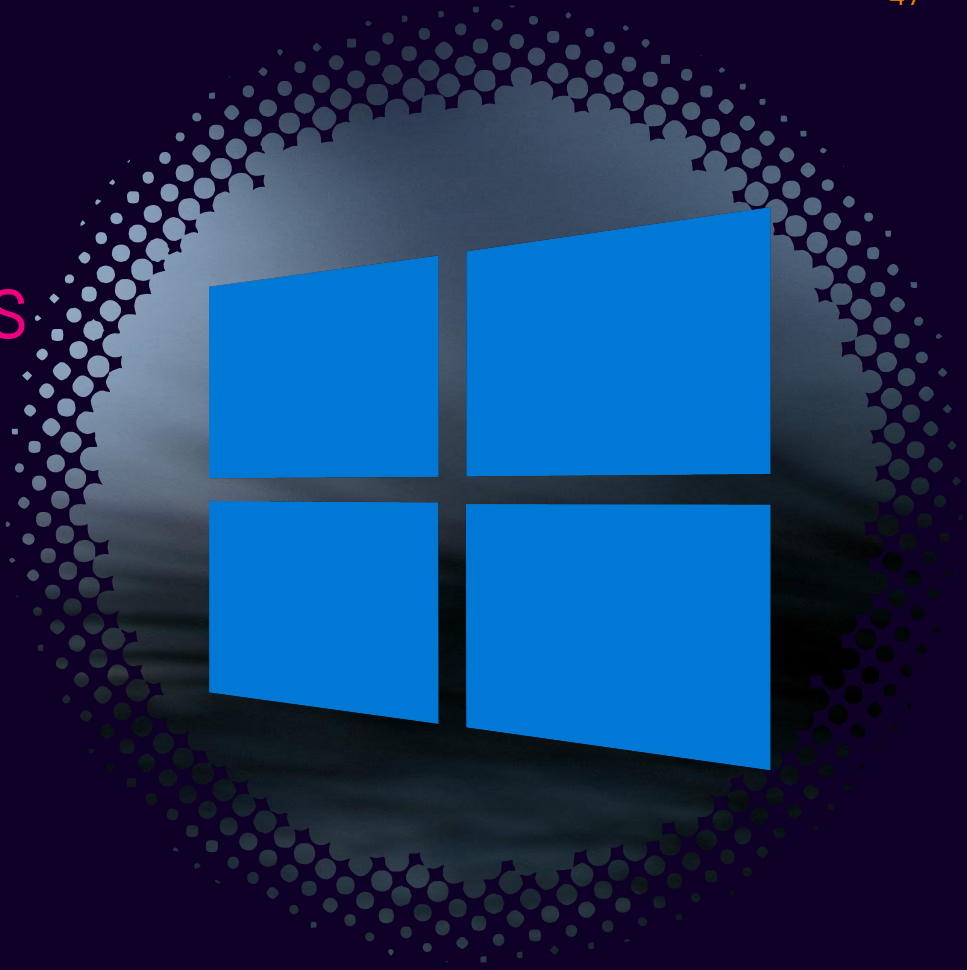
# Windows in Enterprise: Big vectors

- 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?

**10min**. Do not reconfigure 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.

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? (Test system has Chrome)
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
- Hint 2: It's a per user setting.
- Hint 3: HKEY_USERS\[SID]\Software\Policies\Microsoft\office\16.0

# IF TIME ALLOWS SPLIT AND JOIN?

- Take your Office query and map it to users.
- Hint: SPLIT!
- Those who are going faster - keep working on this one until noon until you have a clean query mapping Office results to Users.

# 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)

# 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

Events - Extensions - Augeas

# EVENTS

Configure /etc/osquery/osquery.flags

```
--disable_audit=false
--audit_allow_config=true
--audit_persist=true
--audit_allow_process_events=true
--audit_allow_sockets
```

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

# EVENTS

```
SELECT * FROM process_events;
SELECT * FROM socket_events;
```

Records everything as it runs! Try to start some commands in other shells and query process_events again, or open up links and browse to a website and query socket_events.

What is missing?

# 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.

# EVENTS..ABOUT FILES!

Let's create a FIM Pack in /usr/share/osquery/packs/fim.conf (already there on SSH VM)

Edit osquery.conf and add this near other packs:

"workshop-fim": "/usr/share/osquery/packs/fim.conf",

(See .workshop files in /etc/osquery on SSH VMs)

# EVENTS..ABOUT FILES!

Start osqueryi with the config and flag file:

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

SELECT * FROM file_events;

Then, go create a file somewhere 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 :)

# 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/sudoers';

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

INSERT MIND BLOWN DOT GIF. Infinite configuration tracking possibilities.

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

# EXTENSIONS

# EXTENSIONS

Add-on “packages” to osquery that can be exposed as tables. Some can be read-only, some can write.

Good example: Trail of Bits extensions. Download from https://github.com/trailofbits/osquery-extensions/releases

Located in /etc/osquery/ on SSH VM.

# EXTENSIONS

Manually trying extensions:

```
osqueryi --extension /path/to/your_extension.ext
```

Run wget https://evil.plumbing in a terminal in ssh

Then, load osqueryi and the Trail of Bits extension

# EXTENSIONS

```
osqueryi --extension
/etc/osquery/trailofbits_osquery_extensions.ext
SELECT * FROM HostBLackList;

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

Try wget again. BOOM, sinkholed!

# EXTENSIONS

Start a *ping* in another terminal to 8.8.8.8

Block it with the extension

Did it work?

# EXTENSIONS

You can write your own.

Write extensions allow osquery to become a response agent too.

# 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**