Introduction to Osquery Workshop

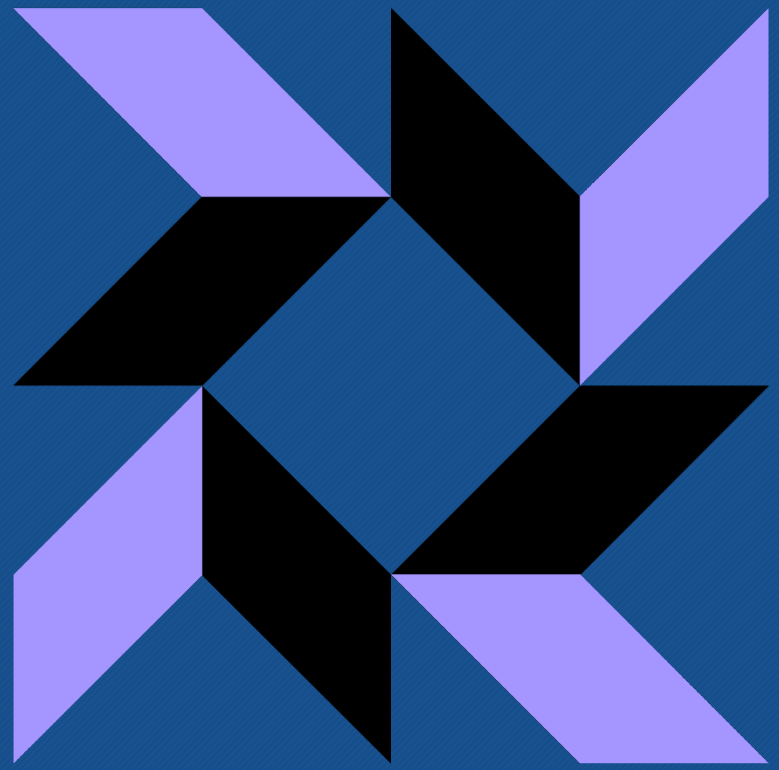
1

2019 Pass the SALT Workshop

ALZETTE
INFORMATION SECURITY

Overview

- Introduction to Osquery
- Osquery Basics
- SQL Refresher
- Osquery Configuration and Extensions
- Fleet Management
- Osquery and Elastic Stack



Introductory Workshop!

- This is an introductory workshop
- You probably won't hear/see a lot of new things if you have:
  - Already used osquery;
  - Followed SANS SEC599, etc.;
- **If you are stuck, please do not suffer in silence!**

Workshop VM

- ais_workshop_xubuntu-18.04.2-desktop-amd64
- VMware Workstation, Player, or Fusion
  - You can try VirtualBox too, but you are on your own with that… sorry! ☺
- 8 GB RAM
- 30-50 GB disk space
- **Keyboard layout: EN-US !!!**
- Workshop VM (Ubuntu) user/pass: **user / Workshop1234%**
  - Normally, it should not require password for login and sudo

About David

- Managing partner at Alzette Information Security (@AlzetteInfoSec)
- Network penetration testing, security architectures, security monitoring, incident response
- Instructor at SANS Institute: FOR572
- BSides Luxembourg organizer https://bsideslux.lu
- Twitter: @DavidSzili
- E-mail: david.szili@alzetteinfosec.com
- Blog: http://jumpespjump.blogspot.com

Introduction to Osquery

6

2019 Pass the SALT Workshop

ALZETTE
INFORMATION SECURITY

About Osquery

What is osquery?

- Build for:
  - Security
  - Compliance
  - Operations (DevOps)
- Everything in SQL!
  - Exposes the operating system as a relational database
- Developed by Facebook

Why osquery?

- (Free) Open Source Software
- Cross-platform
  - One platform for monitoring
  - Native packages for supported operating systems
- Large-scale host monitoring or threat hunting
- Growing Community

Osquery History

2014 OCT 29:
Announcement

2016 SEP 27 /
2016 Oct 4:
Osquery for Windows
(Trail of Bits)

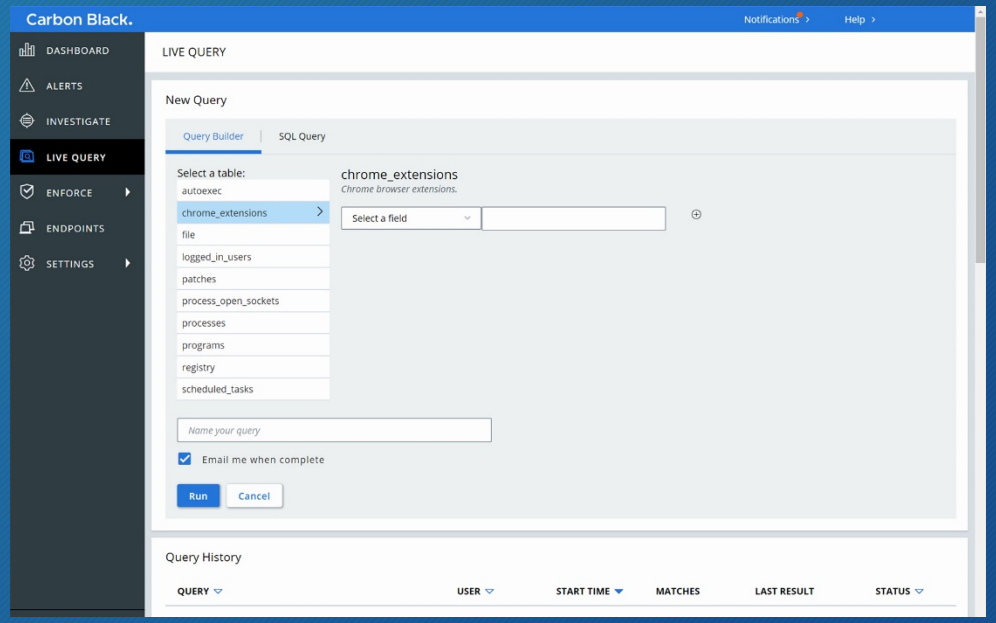
2018 APR 25:
v3.2.4 - First stable release in 3.0.0 series

2019 JUN 28:
osquery 4.0.0 released

ALZETTE
INFORMATION SECURITY

Osquery flavours

Carbon Black LiveOps™



osql

- Osquery open source "soft-fork" from Trail of Bits
- https://blog.trailofbits.com/2019/04/18/announcing-the-community-oriented-osquery-fork-osql/
- https://osql.io

ALZETTE INFORMATION SECURITY

Uptycs

Osquery Basics

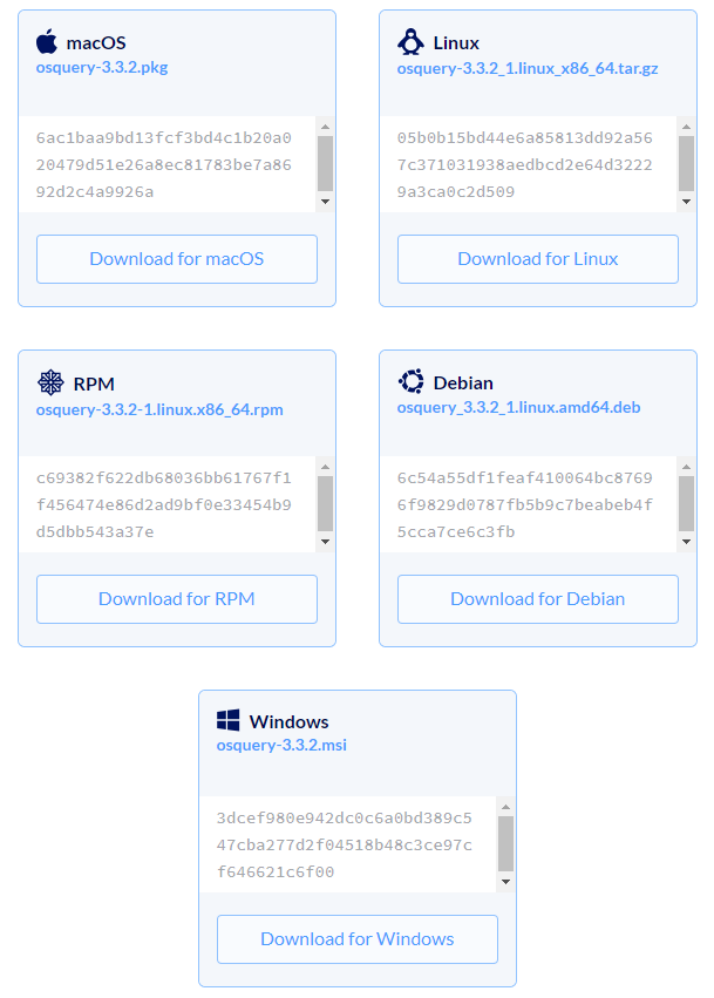
10

2019 Pass the SALT Workshop

ALZETTE
INFORMATION SECURITY

Installation

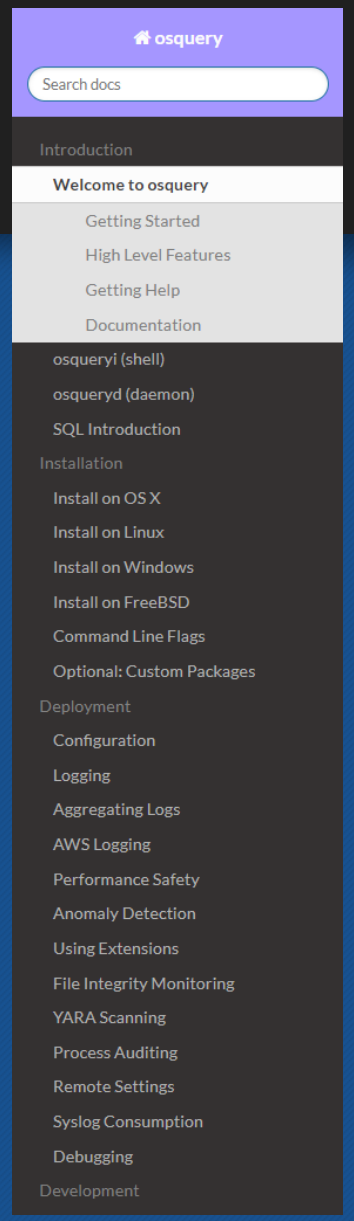
- Built and signed by the osquery team
- Uses minimal number of run-time library dependencies
  - Binaries are a bit big (~20MB)
- Packages for:
  - macOS
  - Linux (Tarball, RPM, DEB)
  - Windows (MSI)
- https://osquery.io/downloads
- Alternative downloads: darwin, apt, yum, freebsd, chocolatey repositories



ALZETTE
INFORMATION SECURITY

Getting Help

- **Osquery Documentation**
  - https://osquery.readthedocs.io/en/stable/
- Osquery Slack
  - https://osquery-slack.herokuapp.com/
- Osquery E-mail (for long-form questions)
  - osquery@fb.com
- Osquery Github
  - https://github.com/facebook/osquery/issues



ALZETTE INFORMATION SECURITY

Main Components

osqueryi

- Interactive query console
- Provides an SQL interface
- Completely standalone, no communication with a daemon
- Does not require elevated privileges (root/Administrator), but not every table can be queried in this case

osqueryd

- Host monitoring daemon
- Distributed, high-performance, low-footprint
- Schedules queries to be executed across an entire infrastructure
- Aggregates query results and generates logs

Osquery SQL and schema

- Superset of SQLite's SQL
  - **SELECT only! (without using extensions)**
  - You can still create run-time tables/VIEWs
- "SQL As Understood By SQLite":
  - https://www.sqlite.org/lang.html

```
osquery> .help
osquery> .tables
osquery> .schema
```

- **Osquery schema documentation**: https://osquery.io/schema
- More than 200 tables in total!
  - All platforms: ~40
  - MacOS: ~160
  - FreeBSD: ~40
  - Linux: ~130
  - Windows: ~73

Using osqueryi

- Used for:
  1. Developing queries
  2. Exploring a single system
- Side note:
  - There is no connection between interactive and daemon mode
  - However, osqueryi and osqueryd are the same binary!
  - You can run osqueryi in daemon mode and osqueryd interactively ☺
- Linux/BSD/MacOS:
  - $ {sudo} osqueryi
- Windows:
  - Osqueryi is not in the path by default
  - C:\ProgramData\osquery\osqueryi.exe {in an Administrator console}

Osquery Shell and Schema Hands-On

2019 Pass the SALT Workshop

SQL Refresher

17

2019 Pass the SALT Workshop

ALZETTE
INFORMATION SECURITY

SELECT (1)

- **SELECT** statement
  - **FROM**: defines input data
  - **WHERE**: boolean expression evaluated for each row
  - **GROUP BY**: Groups the result-set by one or more columns
  - **HAVING**: boolean expression evaluated once for each group (can use aggregate functions)
  - **DISTINCT/ALL**: no duplicate rows/all rows displayed
- https://www.sqlite.org/lang_select.html

Operator	Description
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between a certain range
LIKE	Search for a pattern
IN	Specify multiple values

SELECT (2)

- More on **SELECT** statement:
  - **ORDER BY**: the list of expressions in the ORDER BY determine the order in which rows are returned
    - **ASC**: smaller values returned first
    - **DESC**: larger values returned first
  - **LIMIT**: upper bound on the number of rows returned
    - **OFFSET**: the first X number of rows are omitted from the results
- Compound SELECT Statements
  - **UNION ALL**: returns all the rows from two SELECTs
  - **UNION**: like UNION ALL, but duplicate rows are removed
  - **INTERSECT**: returns the intersection of the results of two SELECTs
  - **EXCEPT**: Returns the subset of rows returned by the left SELECT that are not returned by the right-hand SELECT

Aggregate Functions

Function	Description
avg(X)	Returns the average value of all non-NULL X within a group
count(X)	Returns a count of the number of times that X is not NULL in a group
count(*)	Returns the total number of rows in the group
group_concat(X)	Returns a string which is the concatenation of all non-NULL values of X
group_concat(X,Y)	group_concat(X) and Y is used as the separator between instances of X
max(X)	Returns the maximum value of all values in the group
min(X)	Returns the minimum non-NULL value of all values in the group
sum(X)	Returns the (integer) sum of all non-NULL values in the group
total(X)	Returns the (float) sum of all non-NULL values in the group

JOIN

- **INNER JOIN** (or just **JOIN**): combines column values of two tables based upon the join predicate (**ON** keyword)
  - **USING**: specifies a list of one or more columns as a condition
  - **NATURAL INNER JOIN**: automatically tests for equality between the values of every column that exists in both tables
- **LEFT OUTER JOIN** (or just **LEFT JOIN**): returns all values from the left table, even if there is no match with the right table
  - **ON**, **USING**, **NATURAL**: works the same way as in INNER JOINs
- **CROSS JOIN**: matches every row of the first table with every row of the second table

Osquery Complex Query Example


```
osquery> SELECT datetime(logged_in_users.time,'unixepoch') AS
datetime,logged_in_users.type,logged_in_users.user,users.uid,logged_in_users.tty,logged_in_users.
pid,processes.name AS process_name,processes.path
    ...> FROM logged_in_users
    ...> LEFT JOIN processes USING(pid)
    ...> LEFT JOIN users ON users.username = logged_in_users.user;
+---------------------+-----------+----------+------+------+------+--------------+--------------+
| datetime            | type      | user     | uid  | tty  | pid  | process_name | path         |
+---------------------+-----------+----------+------+------+------+--------------+--------------+
| 2019-03-26 21:35:00 | boot_time | reboot   |      | ~    | 0    |              |              |
| 2019-03-26 21:35:13 | login     | LOGIN    |      | tty1 | 834  | agetty       | /sbin/agetty |
| 2019-03-26 21:35:14 | user      | user     | 1000 | tty7 | 1248 | sh           | /bin/dash    |
| 2019-03-26 21:35:39 | runlevel  | runlevel |      | ~    | 53   |              |              |
+---------------------+-----------+----------+------+------+------+--------------+--------------+
```

ALZETTE
INFORMATION SECURITY

Osqueryi Hands-On

2019 Pass the SALT Workshop

Osquery Configuration and Extensions

2019 Pass the SALT Workshop

Using osqueryd

- Osqueryd is the host monitoring daemon
- It aggregates query results over time and generates logs
- Allows to:
  1. Schedule queries
  2. Record OS state changes, including file and directory changes, hardware events, network events, etc.

- Configuration and query schedule

```
{
  "osquery_info": {
    "query": "SELECT * FROM osquery_info;",
    "interval": 300,
    "snapshot": true
  }
}
```

- Logging and reporting
- Query Packs

Flags and Flagfile

- Osqueryi and osqueryd use optional command line (CLI) flags to:
  - Control initialization
  - Disable/enable features
  - Select plugins
- List of flags: https://osquery.readthedocs.io/en/stable/installation/cli-flags/

- **Flagfile**: flags can be set within environment variables or via a "master" flag file

```
--tls_hostname=ws-vm
--tls_server_certs=C:\ProgramData\osquery\ws-vm.pem
--host_identifier=uuid
--enroll_tls_endpoint=/api/v1/osquery/enroll
--config_plugin=tls
--config_tls_endpoint=/api/v1/osquery/config
--config_tls_refresh=10
--disable_distributed=false
--distributed_plugin=tls
--distributed_interval=10
--distributed_tls_max_attempts=3
--distributed_tls_read_endpoint=/api/v1/osquery/distributed/read
--distributed_tls_write_endpoint=/api/v1/osquery/distributed/write
--logger_plugin=tls
--logger_tls_endpoint=/api/v1/osquery/log
--logger_tls_period=10
--enroll_secret_path=C:\ProgramData\osquery\osquery.secret
```

Configuration

- Osquery "configuration" is read from a config plugin
  - Set to **filesystem** by default
  - HTTP/TLS request using the **tls** config plugin
- The response data must be in JSON format
- Configuration details: https://osquery.readthedocs.io/en/stable/deployment/configuration

- Components in a configuration include
  - **Daemon options and feature settings**
  - **Query Schedule**: the set of SQL queries and intervals
  - **File Change Monitoring**: categories and paths of monitored files and directories
- Filesystem config plugin default locations:
  - Windows: **C:\ProgramData\osquery\osquery.conf**
  - Linux: **/etc/osquery/osquery.conf** and **/etc/osquery/osquery.conf.d/**
  - MacOS: **/var/osquery/osquery.conf** and **/var/osquery/osquery.conf.d/**

Packs

- Configuration supports sets of queries called packs
- Packs are distributed with osquery and labeled based on broad categories
- In an osquery configuration JSON
  - Packs can be defined as a top-level-key and consist of pack name to pack content JSON data structures
  - Pack value may also be a string. In case of the **filesystem** plugin, these strings are considered paths.

```
{
  "options": {
    "enable_monitor": "true“
  },
  "packs": {
    "osquery-monitoring": "/usr/share/osquery/packs/osquery-monitoring.conf",
    "incident-response": "/usr/share/osquery/packs/incident-response.conf“
  }
}
```

```
{
  "options": {
    "enable_monitor": "true“
  },
  "packs": {
    "osquery-monitoring": {
      "queries": {...}
    },
    "incident-response": {
      "queries": {...}
    }
  }
}
```

Logging

- Osqueryd uses logger plugins:
  - **filesystem** (default)
  - **tls**
  - syslog (for POSIX),
  - windows_event_log (for Windows)
  - kinesis
  - firehose
  - kafka_producer
- Log types: status and result logs
- **Status logs**:
  - Generated by the Glog logging framework
  - Logger plugins may intercept these
- **Results logs**: Results of scheduled queries are logged to the "results log"
  - **Differential logs**: Differential changes between the last (most recent) query execution and the current execution
  - **Snapshot logs**: A snapshot is an 'exact point in time' set of results, no differentials

Eventing Framework

- Scheduled queries have limitations
  - Volatile events like process execution
- To overcome this, osquery has the Eventing (pubsub) Framework
  - Aggregating operating system information asynchronously at event time
  - Storing related event details in the osquery backing store
  - Performing a lookup to report stored rows query time
- Almost every pubsub-based table ends with a **_events** or **_changes**
- Note that this reporting pipeline is much more complicated!
  1) Requires additional configuration
  2) As events occur, the rows returned by a query will compound, so queries should always include a time range
  3) The buffered events will eventually expire! Buffer is set to 1 day by default
  4) Eventing Framework will not really work with osqueryi

Eventing Framework Example

File Integrity Monitoring

- Available for Linux and Darwin
- The list of files/directories to monitor is defined in the osquery configuration
- Can use standard wildcards "*" or SQL-style wildcards "%“ for the path definitions
  - %: Match all files and folders for one level
  - %%: Match all files and folders recursively

```
{
  "schedule": {...},
  "file_paths": {
    "homes": [
      "/root/.ssh/%%",
      "/home/%/.ssh/%%“
    ],
    "etc": [
      "/etc/%%“
    ]
  },
  "exclude_paths": {
    "homes": [
      "/home/user/.ssh/%%“
    ]
  }
}
```

Extensions

- Osquery supports proprietary tables, config plugins, and logger plugins
- Thrift-based extensions API
- Osqueryd may "autoload" these extensions and monitor their performance
- Trail of Bits extensions:
  - https://github.com/osql/extensions

- CLI flags for extension auto-loading:

```
--extensions_autoload=/etc/osquery/extensions.load
--extensions_timeout=3
--extensions_interval=3
```

- Extensions.load file example (osquery.ext is an executable):

```
/usr/lib/osquery/extensions/osquery.ext
```

- Manually Loading Extensions:

```
osqueryi {--allow_unsafe} --extension
/path/to/extension.ext
```

ALZETTE INFORMATION SECURITY

Source: https://osquery.readthedocs.io/en/stable/deployment/extensions/

Osquery Configuration and Extensions Hands-On

2019 Pass the SALT Workshop

Fleet Management

2019 Pass the SALT Workshop

Fleet Management Options

- **Kolide Fleet**: https://kolide.com/fleet
  - (Free) Open Source Software from Kolide: https://github.com/kolide/fleet
  - Paid: Kolide Cloud (SaaS)
- **Doorman**: https://github.com/mwielgoszewski/doorman
  - (Free) Open Source Software from Marcin Wielgoszewski
- STG: https://github.com/OktaSecurityLabs/sgt
  - (Free) Open Source Software from Okta
  - "Built Entirely on AWS"
- (osquery-fleet? : https://github.com/sandstorm/osquery-fleet)

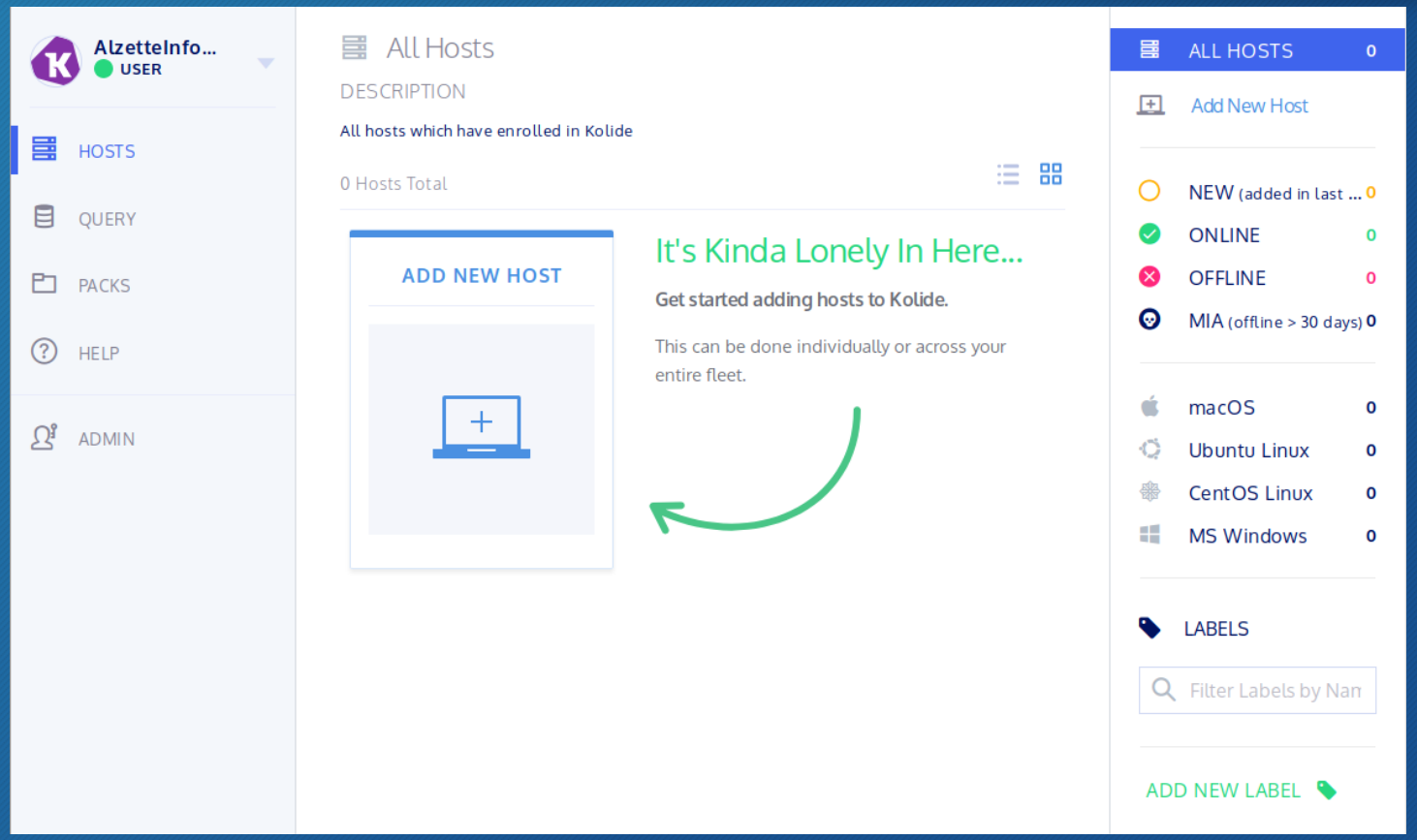
About Kolide Fleet (and Kolide Launcher)

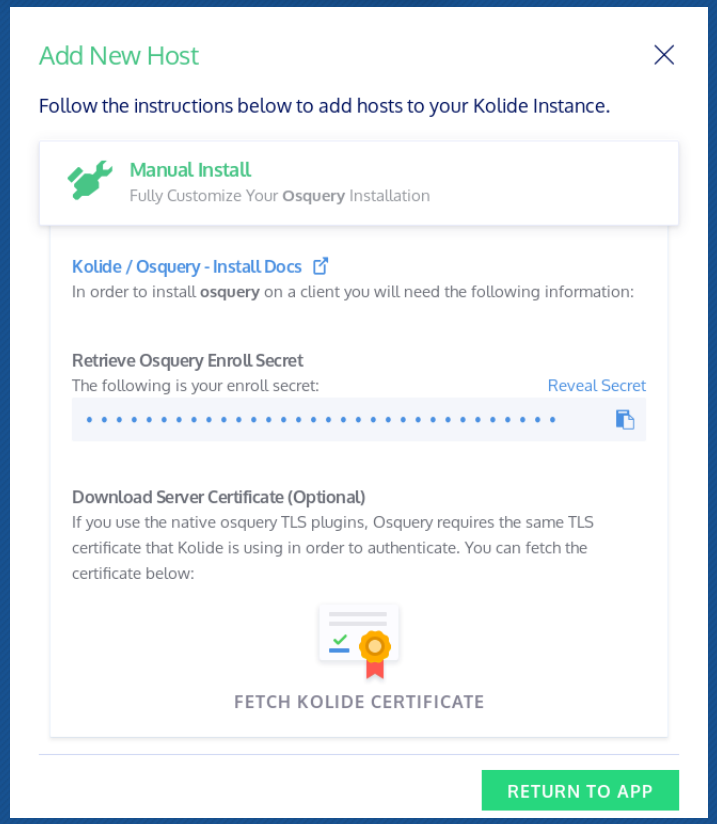
- Open Source Osquery Manager
- Compatible with every major platform
- Designed to work with **Launcher** (Osqery deployment)
- Features:
  - Query dynamic sets of hosts
  - Run queries repeatedly with Packs
  - Create labels populated with hosts matching a query
  - Export results
- **fleetctl**: provides scriptable, CLI based access to osquery on your entire fleet

Kolide Fleet Installation and Configuration

1) Install and configure MySQL
2) Install Redis
3) Generate TLS certificate for Kolide Fleet server
4) Install Kolide Fleet (https://dl.kolide.co/bin/fleet_latest.zip)
5) Configure Kolide Fleet
   a) Create fleet.config
   b) Create MySQL database
   c) Create fleet.service
6) Start Kolide Fleet

Kolide Fleet Interface and Deployment





ALZETTE
INFORMATION SECURITY

Kolide Fleet Hands-On

2019 Pass the SALT Workshop

Osquery and Elastic Stack

2019 Pass the SALT Workshop

Filebeat Configuration

- Filebeat osquery module can be used
- JSON messages can be sent to:
  - Elasticsearch
  - Logstash

```
- module: osquery
  result:
    enabled: true

    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: ["/tmp/osquery_result"]

    # If true, all fields created by this module are prefixed with
    # `osquery.result`. Set to false to copy the fields in the root
    # of the document. The default is true.
    #var.use_namespace: true
```

```
#============================== Filebeat inputs ===============================

filebeat.inputs:
# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.
- type: log
  # Change to true to enable this input configuration.
  # enabled: true
  enabled: false
  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /tmp/osquery_result

#============================== Filebeat modules ===============================

filebeat.config.modules:
  # Glob pattern for configuration loading
  path: ${path.config}/modules.d/*.yml

  # Set to true to enable config reloading
  #reload.enabled: false
  reload.enabled: true

  # Period on which files under path should be checked for changes
  #reload.period: 10s

#================================ Outputs ======================================

#----------------------------- Logstash output --------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["localhost:5044"]
```

ALZETTE
INFORMATION SECURITY

Logstash Pipeline Configuration

- Logstash file needs to be placed to:
  - /etc/logstash/conf.d/
- /etc/logstash/logstash.yml has:
  - config.reload.automatic: true
  - config.reload.interval: 5s

```
input {
    beats {
        port => 5044
    }
}
filter {
    # When not using the osquery filebeat module
    #json {
    #    source => "message"
    #    remove_field => [ "message" ]
    #}
    date {
        # When not using the osquery filebeat module
        #match => [ "unixTime", "UNIX" ]
        match => [ "json.unixTime", "UNIX" ]
    }
}
output {
    elasticsearch {
        hosts => ["localhost:9200"]
    }
}
```

Kibana Discovery

Discover - Kibana - Mozilla Firefox

Kolide Fleet | Discover - Kibana

ws-vm:5601/app/kibana#/discover?_g=h@44136fa&_a=h@0c06fee

kibana

12 hits

New Save Open Share Inspect Auto-refresh Last 15 minutes

Search... (e.g. status:200 AND extension:PHP)

Options Refresh

Add a filter +

Discover
Visualize
Dashboard
Timelion
Canvas
Machine Learning
Infrastructure
Logs
APM
Dev Tools
Monitoring
Management
Default
Collapse

logstash-*

Selected fields
? _source

Available fields
@timestamp
t @version
t _id
t _index
_score
t _type
t beat.hostname
t beat.name
t beat.version
t event.dataset
t fileset.module
t fileset.name
t host.name
t input.type
t json.action
t json.calendarTime
json.counter
t json.decorations.host_uuid
t json.decorations.hostname
json.epoch
t json.hostIdentifier

March 28th 2019, 02:47:30.720 - March 28th 2019, 03:02:30.720 — Auto

Count

02:48:00 02:49:00 02:50:00 02:51:00 02:52:00 02:53:00 02:54:00 02:55:00 02:56:00 02:57:00 02:58:00 02:59:00 03:00:00 03:01:00 03:02:00

@timestamp per 30 seconds

Time | _source

March 28th 2019, 03:02:21.802

prospector.type: log @timestamp: March 28th 2019, 03:02:21.802 event.dataset: osquery.result offset: 16,064 source: /tmp/osquery_result fileset.module: osquery fileset.name: result tags: beats_input_raw_event host.name: ws-vm beat.version: 6.6.2 beat.hostname: ws-vm beat.name: ws-vm @version: 1 input.type: log log.file.path: /tmp/osquery_result json.epoch: 0 json.snapshot: { "instance_id": "e155122d-add4-45d7-9e26-9b9a2f481ce3", "version": "3.3.2", "build_platform": "windows", "start_time": "1553738125", "config_valid": "1", "watcher": "11008", "config_hash": "44c48e3f62e5d494e7b68306b48554c6f032983d", "build_distro": "windows10", "pid": "11056", "extensions": "active", "uuid": "BC4FFF81-548F-11CB-980F-9F751BD61797" } json.hostIdentifier: BC4FFF81-548F-11CB-980F-9F751BD61797 json.u

Table JSON

View surrounding documents View single document

@timestamp	March 28th 2019, 03:02:21.802
t @version	1
t _id	MiUIwmkBvfCuWabEr_dB
t _index	logstash-2019.03.28
# _score	-
t _type	doc
t beat.hostname	ws-vm
t beat.name	ws-vm
t beat.version	6.6.2
t event.dataset	osquery.result
t fileset.module	osquery
t fileset.name	result
t host.name	ws-vm

ALZETTE INFORMATION SECURITY

Osquery and Elastic Stack Hands-On

2019 Pass the SALT Workshop

Questions and Answers

2019 Pass the SALT Workshop

References

- Osquery Website and Osquery Schema
  - https://osquery.io
  - https://osquery.io/schema
- Osquery Docs
  - https://osquery.readthedocs.io
- Kolide Website
  - https://kolide.com
- Elastic Website
  - https://www.elastic.co