Doorman

An osquery fleet manager

About me

Marcin Wielgoszewski

- Security engineer at a digital asset (cryptocurrency) exchange
- Previously
  - Matasano Security (now NCC Group)
  - Gotham Digital Science

git.io/vof8M

Outline

- Brief introduction to osquery
- Overview of a typical osquery deployment
- How we use osquery
- Managing our osquery fleet with Doorman
- Demo
- Doorman in production
- Summary

Introduction to osquery

osquery

Enables the collection of low-level information from an operating system

- Exposes the information as a database you can query via SQL
- Queries can be ad-hoc or run on a scheduled interval
- Changes in state between query runs is logged
- Compatible with Linux (Ubuntu and CentOS), MacOS, Windows
- Maintains a relatively small footprint

Sample osquery queries

Determine if OS X user has screensaver require a password and the delay before asking:

```
osquery> select username, key, value from (select * from users where directory
like '/Users/%') u, preferences p where p.path = u.directory || '/Library/
Preferences/com.apple.screensaver.plist';
+---------------+---------------------+-------+
| username      | key                 | value |
+---------------+---------------------+-------+
| marcin        | askForPassword      | 1     |
| marcin        | askForPasswordDelay | 0     |
| marcin        | tokenRemovalAction  | 0     |
+---------------+---------------------+-------+
```

Sample osquery queries

Query all non-Apple kernel extensions:

```
osquery> select name, version from kernel_extensions where name not like
'com.apple.%' and name != '__kernel__' order by name;
+--------------------------------------+---------+
| name                                 | version |
+--------------------------------------+---------+
| com.viscosityvpn.Viscosity.tap       | 1.0     |
| com.viscosityvpn.Viscosity.tun       | 1.0     |
| org.virtualbox.kext.VBoxDrv          | 5.0.16  |
| org.virtualbox.kext.VBoxNetAdp       | 5.0.16  |
| org.virtualbox.kext.VBoxNetFlt       | 5.0.16  |
| org.virtualbox.kext.VBoxUSB          | 5.0.16  |
|--------------------------------------|---------|
```

Sample osquery queries

Identify processes listening on a local port which originate from /tmp

```
osquery> select name, address, port, cwd, cmdline from listening_ports
join processes using (pid) where family = 2 and protocol = 6 and cwd like
'%/tmp%' or path like '%tmp%';

+-----------+-----------+------+--------------+----------------+
| name      | address   | port | cwd          | cmdline        |
+-----------+-----------+------+--------------+----------------+
| python2.7 | 127.0.0.1 | 5001 | /private/tmp | python test.py |
+-----------+-----------+------+--------------+----------------|
```

A typical osquery deployment

A typical osquery deployment

- Endpoints are centrally managed
  - Chef, Puppet
- Logs are collected and aggregated locally
  - Logstash, Splunk, Rsyslog
- Logs ultimately end up in ELK or Splunk for later analysis

https://osquery.readthedocs.io/en/stable/deployment/log-aggregation/

Our problem

- Laptops have a different threat model than our servers
- Employees are expected to manage their own laptops, apply updates, and abide by our security policies and basic security requirements
- These policies reduce our visibility into a considerable part of of our environment

Important considerations

1. Avoid creating a central point of compromise by installing a sanctioned RAT on everyone's machine
    A. No remote code execution (i.e., no Chef, Casper, etc)
2. Avoid introducing and/or exposing a path to sensitive internal infrastructure to the Internet
    A. ELK on the Internet? No way!
3. Avoiding installing more software than we have to, and if we do, keep it as lightweight as possible
    A. Need to figure out how to manage configuration and log aggregation

Other important considerations

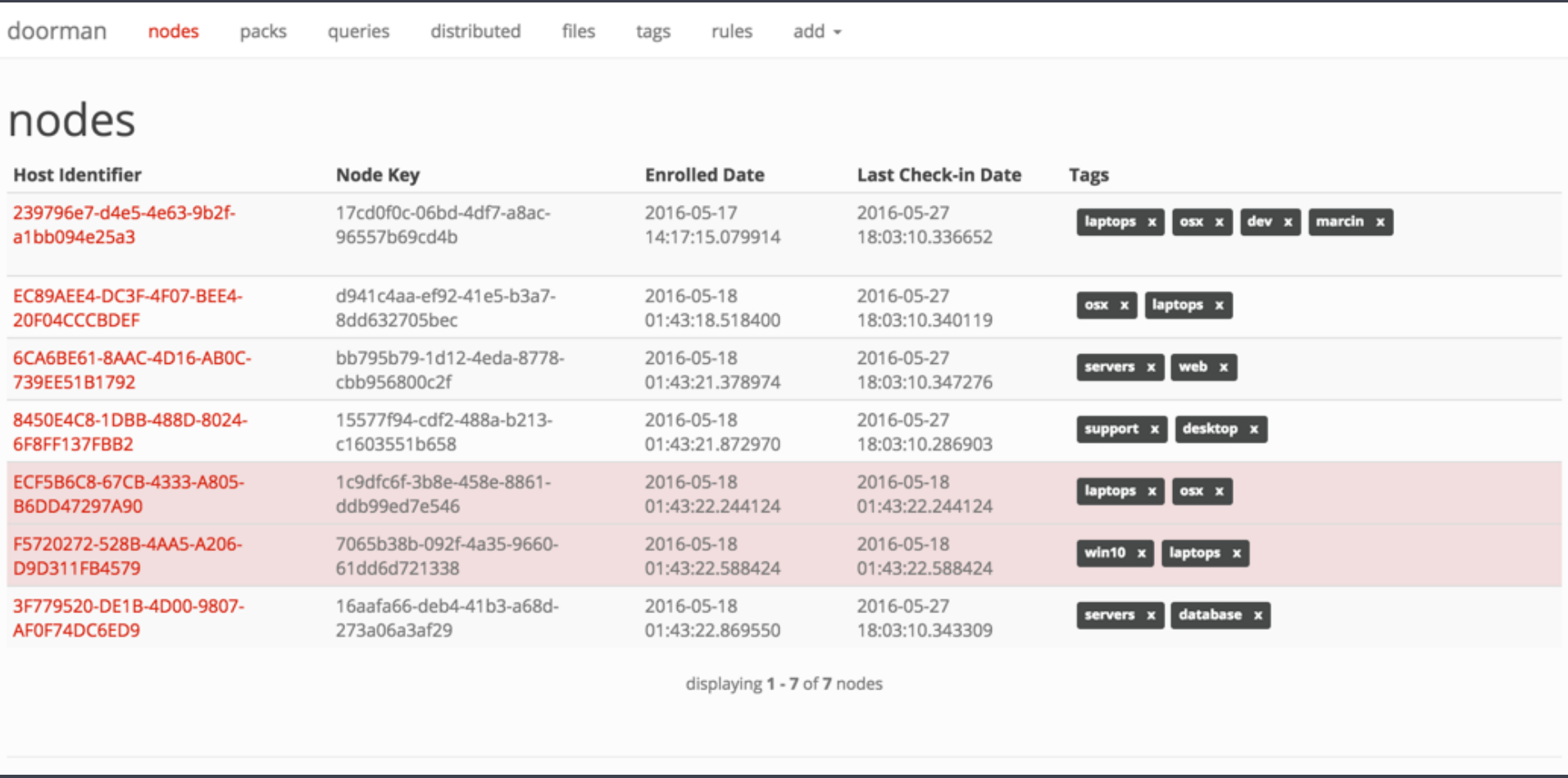
4. Not all employees may connect to our VPN, or remote working conditions may prevent them
    A. Laptops might be turned off for extended periods
    B. Need to be able to re-establish contact afterward
5. Respectful of our employee’s privacy and system performance
    A. Nothing that pegs CPU for minutes at a time while opening an archive
    B. No undocumented kernel hooks, etc
    C. Don’t support ability to snoop users’ browser history or what Nickelback songs they enjoy

Managing our fleet with osquery and Doorman

Doorman

An osquery fleet manager

- Tags identify and associate nodes with packs and queries (ultimately comprising an osquery configuration)
- Schedule ad-hoc queries to be run
- Provides an “at-a-glance” view of results
    - Optionally log results elsewhere via log plugins (if you want to keep ELK)
- Create rules and alerts when specific conditions apply
    - Result returned contains a specific key / value


doorman
nodes
packs
queries
distributed
files
tags
rules
add
nodes
Host Identifier
Node Key
Enrolled Date
Last Check-in Date
Tags
239796e7-d4e5-4e63-9b2f-a1bb094e25a3
17cd0f0c-06bd-4df7-a8ac-96557b69cd4b
2016-05-17 14:17:15.079914
2016-05-27 18:03:10.336652
laptops x
osx x
dev x
marcin x
EC89AEE4-DC3F-4F07-BEE4-20F04CCCBDEF
d941c4aa-ef92-41e5-b3a7-8dd632705bec
2016-05-18 01:43:18.518400
2016-05-27 18:03:10.340119
osx x
laptops x
6CA6BE61-8AAC-4D16-AB0C-739EE51B1792
bb795b79-1d12-4eda-8778-cbb956800c2f
2016-05-18 01:43:21.378974
2016-05-27 18:03:10.347276
servers x
web x
8450E4C8-1DBB-488D-8024-6F8FF137FBB2
15577f94-cdf2-488a-b213-c1603551b658
2016-05-18 01:43:21.872970
2016-05-27 18:03:10.286903
support x
desktop x
ECF5B6C8-67CB-4333-A805-B6DD47297A90
1c9dfc6f-3b8e-458e-8861-ddb99ed7e546
2016-05-18 01:43:22.244124
2016-05-18 01:43:22.244124
laptops x
osx x
F5720272-528B-4AA5-A206-D9D311FB4579
7065b38b-092f-4a35-9660-61dd6d721338
2016-05-18 01:43:22.588424
2016-05-18 01:43:22.588424
win10 x
laptops x
3F779520-DE1B-4D00-9807-AF0F74DC6ED9
16aafa66-deb4-41b3-a68d-273a06a3af29
2016-05-18 01:43:22.869550
2016-05-27 18:03:10.343309
servers x
database x
displaying 1 - 7 of 7 nodes

voltatype / recent activity / logs

Host Identifier	Node Key	Enrolled Date	Last Check-in Date
voltatype	61d24e88-2ebe-4144-9dba-1902c4943310	2016-06-03 19:06:10.792812	2016-06-06 20:11:28.916239

Packs

No packs currently associated with this node. Use the tag functionality to associate nodes with packs.

Queries

usb_devices - Retrieves the current list of USB devices in the target system.

Query	`select * from usb_devices;`
Interval	`10`
Platform	all
Version	1.4.5
Value	General security posture.
Show Removed?	True
Packs	it-compliance
Tags	marcin x

voltatype / recent activity / logs

distributed queries 2

chrome_extensions 21

usb_devices 103

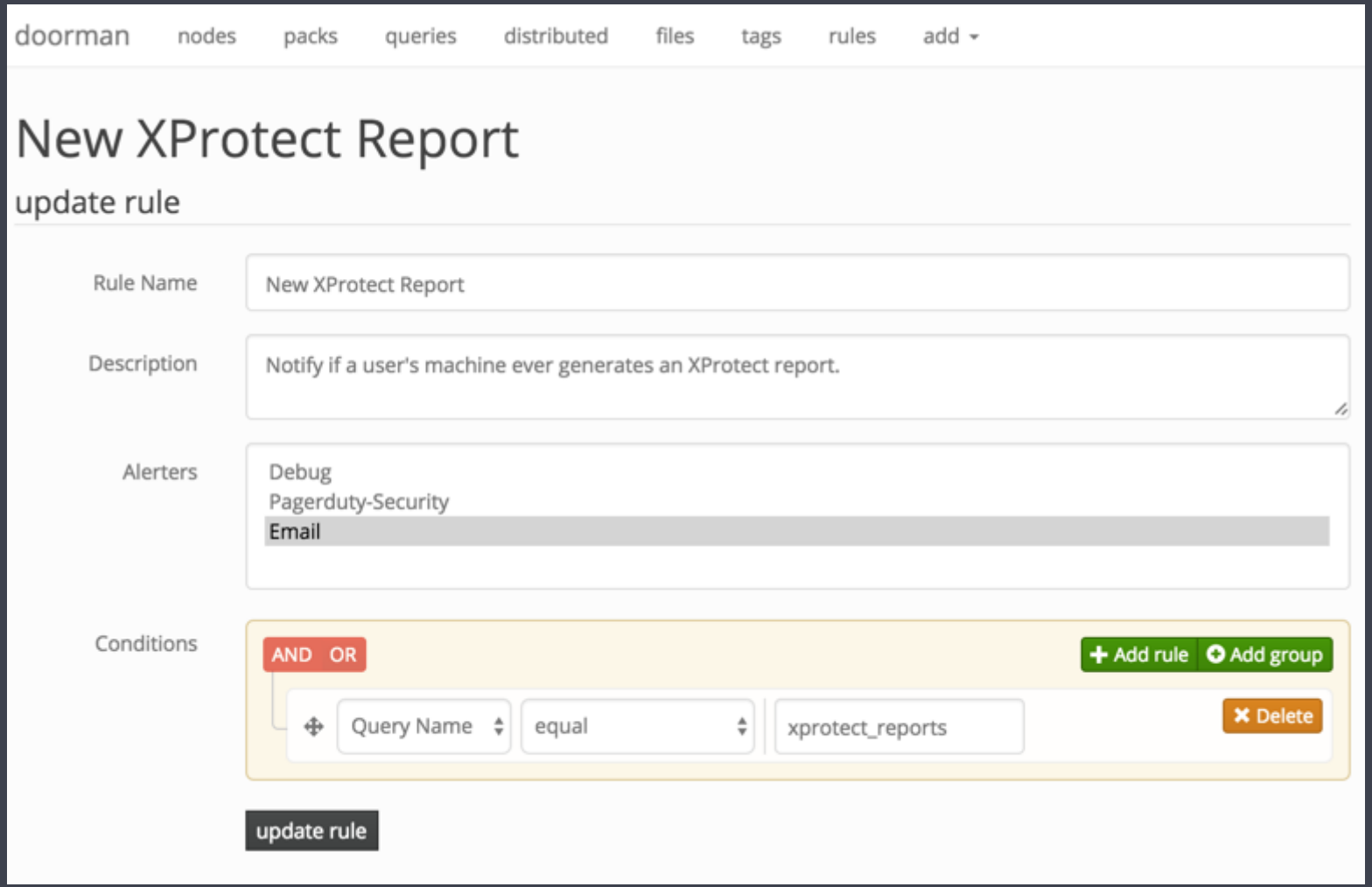
activity	timestamp	model	model_id	removable	serial	usb_address	usb_port	vendor	vendor_id
removed	2016-06-04 20:02:25	Security Key by Yubico	0120	1	0	31	5	Yubico	1050
removed	2016-06-04 16:14:15	Java Token	080f	1	0	18	1	FT	096e
removed	2016-06-04 16:14:06	Display Audio	1107	0		4	4	Apple Inc.	05ac
added	2016-06-04 16:14:06	Internal Memory Card Reader	8406	0		17	4	Apple	05ac
added	2016-06-04 16:14:06	Apple Internal Keyboard / Trackpad	0262	0	0	19	12	Apple Inc.	05ac
added	2016-06-04 16:14:06	Java Token	080f	1	0	18	1	FT	096e
removed	2016-06-04 16:14:06	Apple Internal Keyboard / Trackpad	0262	0	0	29	12	Apple Inc.	05ac
removed	2016-06-04 16:14:06	Apple Keyboard	024f	0	0	9	2	Apple Inc.	05ac

Demo

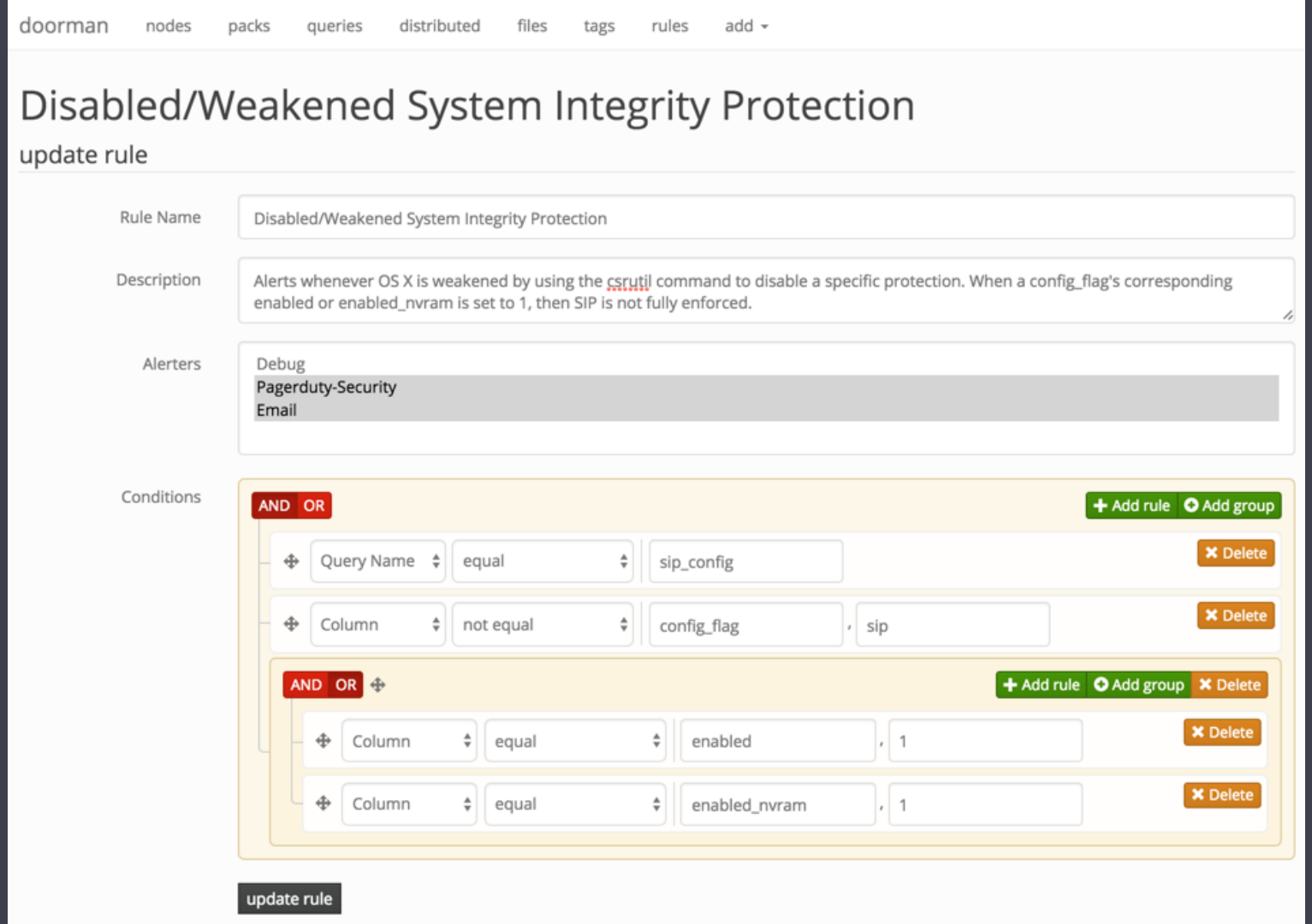
Doorman

Create rules to alert when configuration drifts or violates policy

- For example,
  - A new browser extension is installed
  - Security protections are disabled (SIP, ALF, Filevault, anti-virus, etc)
  - Unauthorized hardware is inserted
  - LaunchAgent is installed
- Alert via PagerDuty, Email, etc


doorman
nodes
packs
queries
distributed
files
tags
rules
add
New XProtect Report
update rule
Rule Name
New XProtect Report
Description
Notify if a user's machine ever generates an XProtect report.
Alerters
Debug
Pagerduty-Security
Email
Conditions
AND OR
Add rule
Add group
Query Name
equal
xprotect_reports
Delete
update rule

X


doorman
nodes
packs
queries
distributed
files
tags
rules
add
Disabled/Weakened System Integrity Protection
update rule
Rule Name
Disabled/Weakened System Integrity Protection
Description
Alerts whenever OS X is weakened by using the csrutil command to disable a specific protection. When a config_flag's corresponding enabled or enabled_nvram is set to 1, then SIP is not fully enforced.
Alerters
Debug
Pagerduty-Security
Email
Conditions
AND OR
Add rule
Add group
Query Name
equal
sip_config
Delete
Column
not equal
config_flag
sip
Delete
AND OR
Add rule
Add group
Delete
Column
equal
enabled
1
Delete
Column
equal
enabled_nvram
1
Delete
update rule

Demo

Doorman

Leverages osquery's built-in TLS remoting plugin

- Nodes are configured to "poll" Doorman's HTTP endpoints periodically
  - Retrieve updated configurations (packs, queries, file integrity monitoring)
  - Result logs, status logs, and distributed queries
- Communication is pinned to a set of TLS server certificates
- Polling nature of TLS remoting avoids the need for central management or complex log aggregation and collection
  - https://osquery.readthedocs.io/en/stable/deployment/remote/

Deploying osquery on OS X

Installed during laptop provisioning

- pkg installer contains all the required files and config settings
  - Remoting endpoints are configured to respective Doorman API endpoints
  - Interval at which those endpoints are called
  - Shared enrollment secret and TLS server certificates
  - Some tables are disabled for privacy reasons (shell_history, file)
  - Result buffer size in the event osqueryd cannot reach Doorman
  - Installs a LaunchDaemon to start osqueryd automatically
- Updates to osquery distributed manually to users

Managing our osquery fleet with Doorman

Doorman allows us to safely collect osquery results without exposing sensitive, internal infrastructure to the Internet

- No need to put Logstash out on the Internet, or give everyone VPN to collect results
- No need to install and manage additional log aggregation agents on the laptops

Using osquery, we gain visibility into our laptops without sacrificing performance, security, and privacy

Doorman in Production

- Python Flask / Celery web application
- Postgres database
- Message queue
    - We use Redis
- API and manager applications can be deployed as separate wsgi apps
    - We deploy the API to be accessible externally behind a load balancer
- Currently managing <50 nodes w/ a single t2.medium instance in AWS

Doorman in Production (one year later)

- Relatively stable over the past year
- Added database indexes helped improve UI responsiveness
- Enrollment notifications to validate laptop build process is being followed
- Need better notification capabilities to detect when a node goes offline for an extended period, or has ceased reporting valid results
    - Backlog of osquery results, poor connectivity, nginx timeouts, HTTP compression, local database corruption

Scaling Doorman

Flexible architecture *should* make Doorman easy to scale

- Multiple API servers can be deployed separately
- Increase number of Celery workers
- PostgreSQL is most likely going to be the bottleneck

With that said, haven't run into any scalability concerns (and shouldn't at our size), yet

- If anyone is running 5000+ nodes, come talk to me

Summary

- Doorman and osquery provides us visibility into an otherwise unmanaged fleet
- Don’t expose additional attack surface via remote access capabilities
- Maintain transparency with end users via detailed logging of queries
- Establishes a baseline configuration for our environment
- Query a set of nodes on an ad-hoc basis for information

Thanks!

- Andrew Dunham (and Stripe) for committing engineering time to development
- Diogo Mónica (for hosting this track at QConNY!)
- Dan Guido (Trail of Bits)
- Teddy Reed and Mike Arpaia (Facebook)

git.io/vof8M