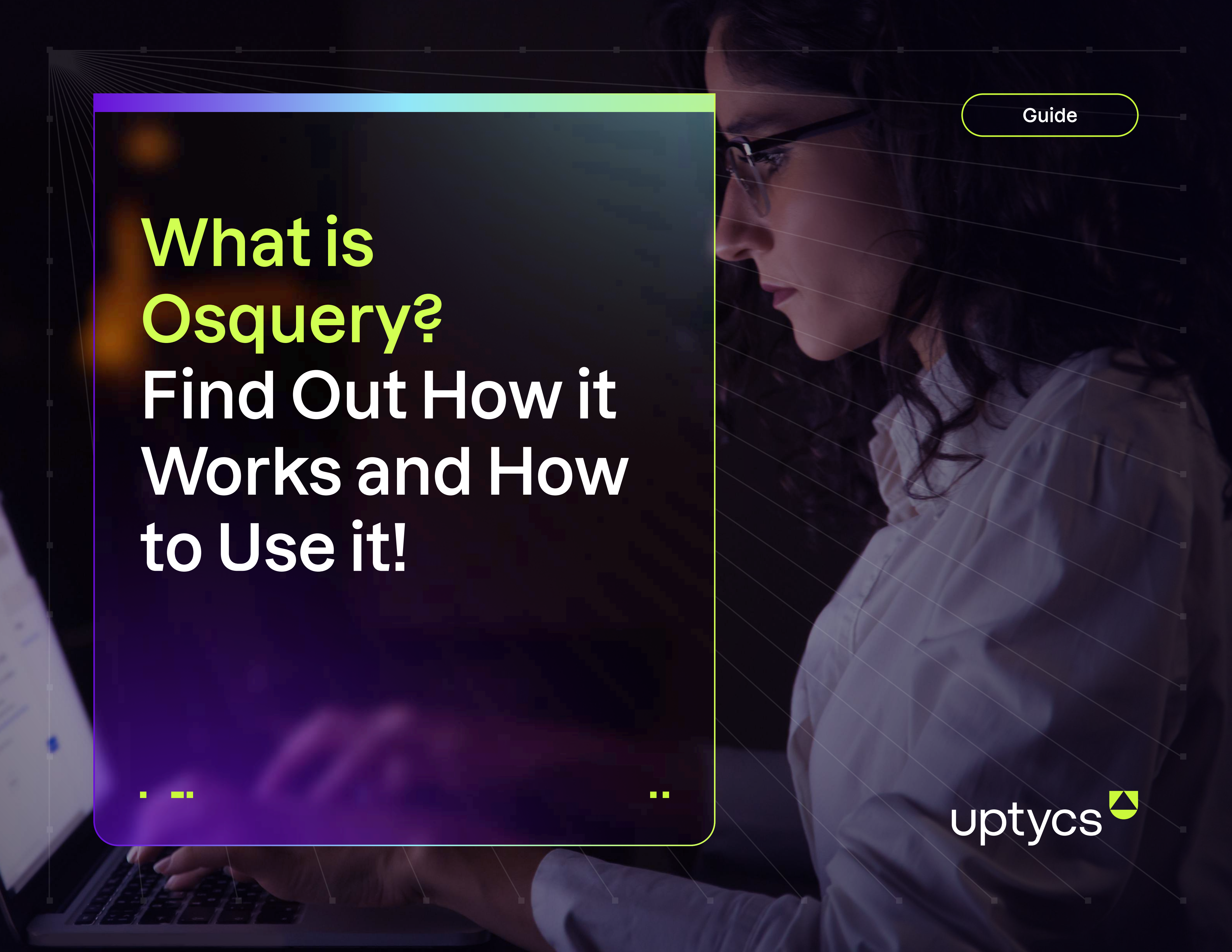
Guide

What is Osquery? Find Out How it Works and How to Use it!



uptycs

Table of Contents

Overview	01
What Is Osquery?	02
Getting Started With Osquery	08
Osquery: Build vs Buy	16
Running Osquery At Scale	22

Overview

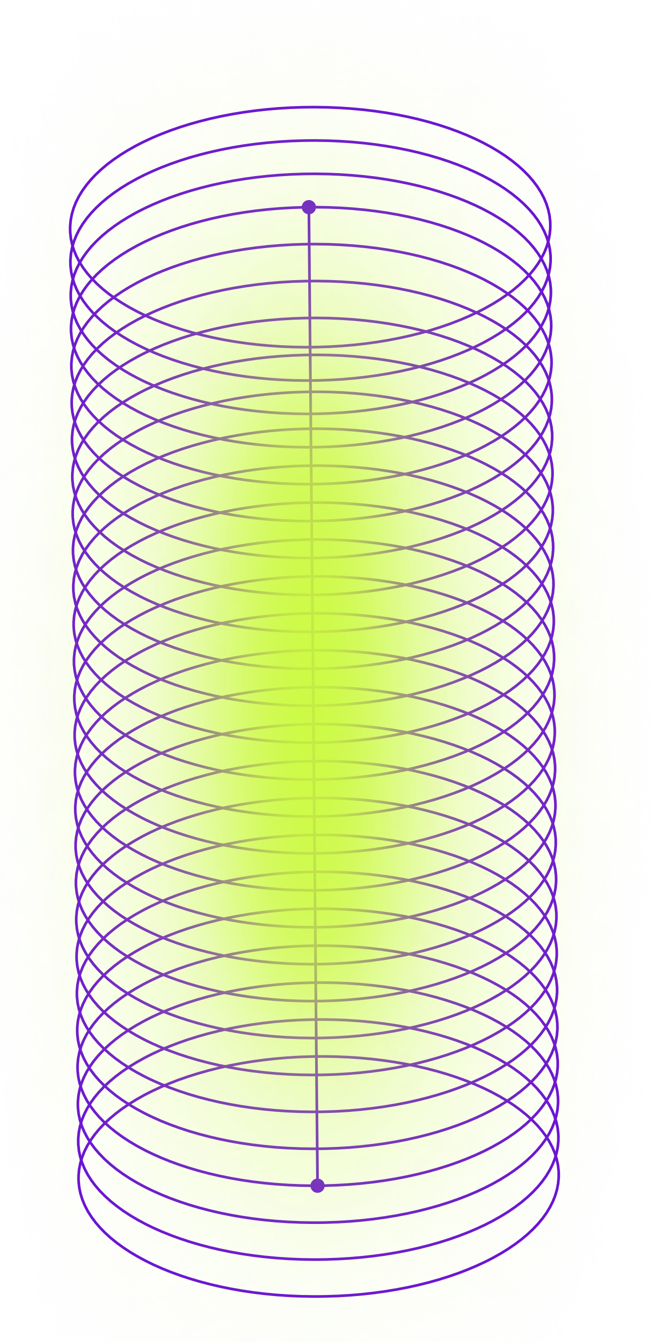
You're probably well aware visibility depends on data—and this is where things can get complicated. Today's organizations are overwhelmed with endpoint agent options, many of which hog resources, require complicated setup, use an esoteric language, and only work with one operating system.

What if one data collection agent was lightweight, configurable, easily accessible, worked in all systems, and used a common language like SQL? No more struggling to standardize data between disparate systems or learning obscure languages to run your queries. Sounds too good to be true?

This universal endpoint agent exists. It's called Osquery and it can collect and normalize data independent of operating systems while increasing visibility across your infrastructure.

In 2014, Facebook created open-source Osquery as a high-performance agent to monitor security on Linux and Mac operating systems. Windows support was launched in 2016, and stewardship of the Osquery project transferred to the Linux Foundation in 2019. A committed group of organizations and community members continues to nurture Osquery through ongoing development and education.

Considering all the advantages this agent offers, it's time to learn more about Osquery.

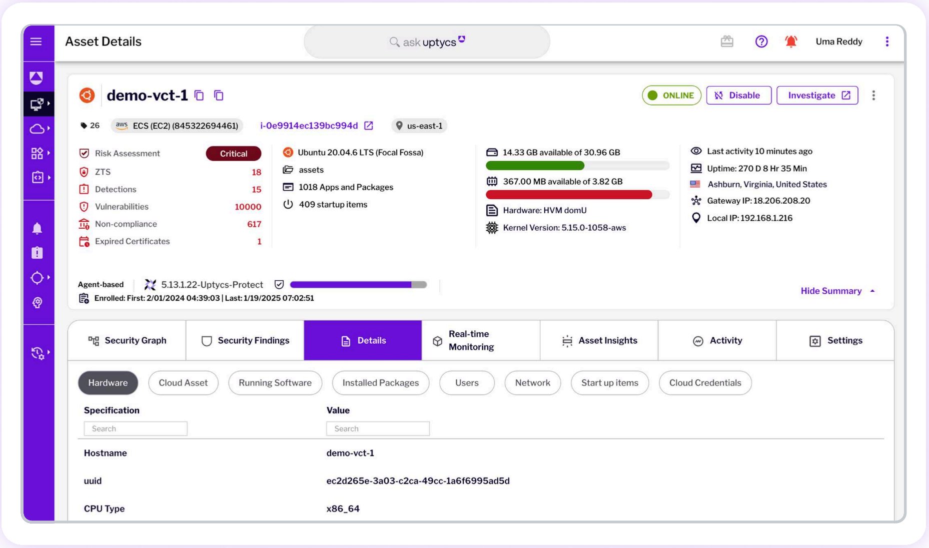


01 What is Osquery?

Osquery is an operating system instrumentation agent that provides a unique and refreshing approach to security. It delivers a single-agent solution using a universal query language to collect rich datasets for multiple use cases. Osquery simplifies the process of understanding your infrastructure by exposing an operating system as a high-performance relational database.

Osquery is a universal endpoint agent that supports multiple operating systems, including macOS, Linux, FreeBSD, and Windows. Additionally, Osquery can monitor Docker containers, providing flexibility and data visibility across various platforms. One of the most powerful features of Osquery is its ability to collect and normalize data independently of the operating system. This eliminates the need for custom scripts for each platform, allowing organizations to run the same queries across different environments with ease.

Osquery is capable of collecting a wide range of data types, which are far richer than standard log files. Some examples of the information Osquery can gather include running processes; user logins; loaded kernel modules; open network connections; browser plugins; hardware events; file hashes; sockets; mounts and storage volumes; installed software packages; and more. This extensive data collection makes Osquery a powerful tool for security, compliance, and troubleshooting purposes, providing comprehensive visibility across systems.



demo-vct-1

26 | ECS (EC2) (845322694461) | i-0e9914ec139bc994d | us-east-1

Risk Assessment	Critical
ZTS	18
Detections	15
Vulnerabilities	10000
Non-compliance	617
Expired Certificates	1

Ubuntu 20.04.6 LTS (Focal Fossa)
assets
1018 Apps and Packages
409 startup items

14.33 GB available of 30.96 GB
367.00 MB available of 3.82 GB
Hardware: HVM domU
Kernel Version: 5.15.0-1058-aws

Last activity 10 minutes ago
Uptime: 270 D 8 Hr 35 Min
Ashburn, Virginia, United States
Gateway IP: 18.206.208.20
Local IP: 192.168.1.216

Agent-based | 5.13.1.22-Uptycs-Protect
Enrolled: First: 2/01/2024 04:39:03 | Last: 1/19/2025 07:02:51

Specification	Value
Hostname	demo-vct-1
uuid	ec2d265e-3a03-c2ca-49cc-1a6f6995ad5d
CPU Type	x86_64

The exciting news for users? With Osquery, running queries no longer requires specialized expertise. Your team can write SQL-based queries to explore data across all operating systems and infrastructure. Osquery uses SQL tables to represent abstract concepts such as running processes, loaded kernel modules, open network connections, browser plugins, hardware events, and file hashes.

The interactive version of Osquery, Osqueryi, is a stand-alone console shell. This version can collect many types of information without running as root, uses an in-memory default database, and doesn't connect or communicate with the Osqueryd daemon. You can use Osqueryi to mock-up queries and begin exploring your operating system.

Osqueryd is a high-performance, low-footprint, host-monitoring daemon that drives insight by monitoring your infrastructure changes.

With our osqueryd, your team can schedule queries to run across your entire infrastructure, and subscribe to events as they happen.

This version effectively accumulates and logs query and event data that reflects systemic changes and various OS events. With osqueryd, logging is seamless, using a plugin architecture where-in, various logging targets can be configured and integrated into your organization's log aggregation pipeline.

The data generated by osqueryd queries can be invaluable in providing a snapshot of your operating system's configuration, security posture, functioning, and overall condition. The events data can be useful to identify launching of new programs, monitoring files, network connections, users logging in and so on.

A key feature of Osquery is the use of query packs, which are collections of pre-built queries designed for specific tasks like incident response, vulnerability management, or compliance monitoring.

These query packs can be scheduled to run at specific intervals, ensuring continuous monitoring without overwhelming the system. It is important to note that query packs only gather the data necessary for compliance or security audits, but teams must still analyze and act upon the data. While Osquery provides built-in query packs, smart organizations often customize these packs to better fit their specific needs, optimizing performance and minimizing the load on their systems.

For example, a compliance pack contains queries searching for systemic changes or anomalies concerning compliance. Your team can choose from a list of built-in query packs covering commonly desired use cases like hardware monitoring, incident response, compliance, and macOS attacks. Built-in query packs aren't updated often, so they're best used as examples. Smart organizations embrace Osquery's flexibility by customizing query packs to meet their specific needs.

Osquery offers many choices, but the options aren't so numerous when it comes to the storage of data.

Some event-based data can be cached in RocksDB, which provides Osquery users with a local, embedded storage option for fast, convenient data persistence.

RocksDB works well as a temporary function, but it's not a centralized, long-term data store. Other options for storage include security information and event management (SIEM) systems (like Splunk), or security analytics platforms (like Uptycs) for threat intelligence, correlations, and anomaly detection.

Lookout®

"Uptycs has been instrumental for our FedRamp authorization and ISO 27001 certification."

Grant Kahn
Director, Security Enginerring Lookout

What Problems Does Osquery Solve?

Osquery has many applications, but it does particularly well in the following areas.

Visibility

The most critical problem Osquery solves for organizations is visibility across all systems and infrastructure.
Many teams don't know what machines make up their fleet, what programs are running, if configurations are correct, and if passwords are current. Osquery provides a clear view of all operating systems, ensuring all machines are set up and performing correctly.

"It just gives you so much visibility and protection. Without the knowledge of what's going on in your system, you're at risk. With 250 subject areas' worth of info, Osquery is unrivaled in the breadth of visibility it provides, in addition to how lightweight and easy it is to roll out."

Julian Wayte
Senior Security Solutions Engineer

Osquery is unrivaled in the breadth of visibility it provides, in addition to how lightweight and easy it is to roll out.

Compliance

Osquery provides the necessary information to stay compliant with the growing number of data privacy protection laws. Maintaining compliance starts with making sure all your machines and infrastructure configure within current compliance guidelines.

Compliance often requires documented evidence proving you've successfully monitored and protected your critical files. Osquery queries deliver the data to verify the integrity of sensitive files, maintain data privacy compliance, and keep your organization off the hot seat with regulators.

Data Standardization

Employing a universal, cross-operational endpoint agent that runs queries and stores data in SQL format will make your security team's job easier. They can enjoy increased production by conducting investigations in a more structured fashion.

For example, Windows teams can conduct Linux issue investigations because transferring data between systems is no longer a problem. With a universal solution like Osquery, standardized data smoothly flows between agents and systems to simplify the security process.

Security

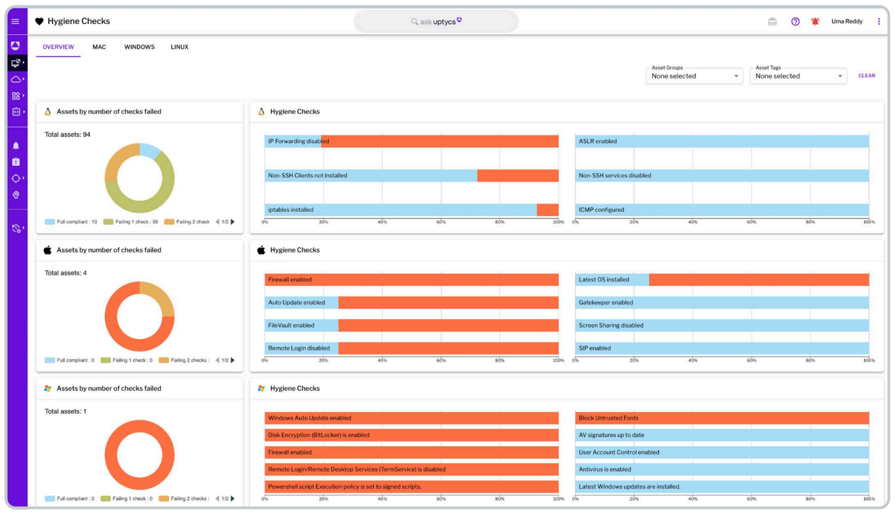
Osquery can function as an Endpoint Detection and Response (EDR) tool. By allowing organizations to query machines for both preemptive threat detection and post-incident investigation, Osquery performs a crucial role in system auditing, compliance, and threat hunting. Its SQL-based query functionality lets security teams quickly collect data on running processes, network connections, and file changes, making it an essential tool for maintaining security visibility across all endpoints.

Osquery can help any security team by providing a continuous stream of events as they occur on the particular system. These events can be funneled into a platform like Uptycs and enhanced with alerts, signals and detections in a way that security teams remain on top of all major security events as they occur within the fleet. This can be followed up with threat hunting, forensics, and intrusion detection by providing real-time views of every running process and network connection.

Who Uses Osquery?

The advantages of Osquery are useful across organizations of any size, but it's the larger players that benefit the most from Osquery's capabilities. Sizeable companies tasked with capturing vast amounts of data across many machines can use the tool's ability to scale.

Does your organization have a large production environment with multiple operating systems? Are you staffed well enough to be proactive in your security posture? If so, Osquery could be just what your team needs to gain priceless insight into the configuration, security, compliance, and performance of your infrastructure.



Takeaways: The Basics Of Osquery

When it comes to data collection agents, organizations have far too many complicated, disconnected choices. Many are rigid frameworks, locking you into one operating system, draining your system's resources, requiring detailed configuration, or demanding expertise in some esoteric language. Gaining visibility into your operating systems and infrastructure doesn't need to be this difficult.

Osquery breaks out of the traditional siloed approach to endpoint agents. It streamlines security as a lightweight, single-agent solution for collecting and normalizing data in a simple, universal language that's independent of operating systems. This unique tool is easily accessible to all and standardizes data, breaking down the walls between different machines and systems.

Here's a simple breakdown of some pros and cons. While Osquery offers powerful data collection and flexibility, there are several pros and cons to consider when implementing it in your environment.

Pros:

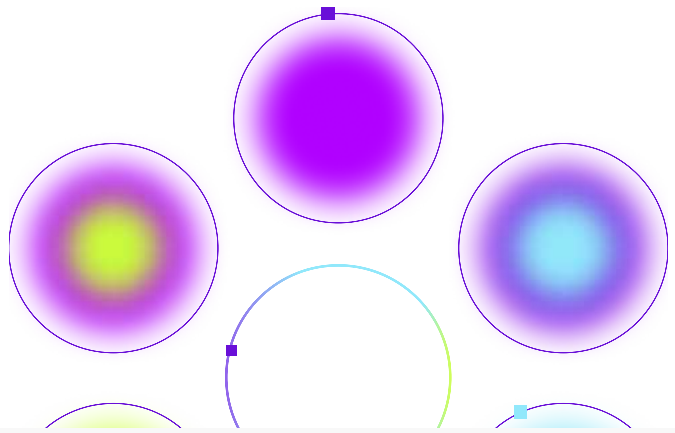
- **Simplicity:** Easy to use for teams familiar with SQL.
- **Customizable:** Highly customizable for specific organizational needs.
- **Extensive Visibility:** Provides access to endpoint data that was previously difficult or impossible to gather.

Cons:

- **High Data Storage Costs:** Osquery can generate a large amount of data, which can become expensive to store, especially when using SIEM solutions like Elastic.
- **Complex Data Analysis:** Analyzing and interpreting the collected data for security, compliance, and threat detection can be complex, often requiring additional tools or platforms.
- **Resource Impact:** Queries can sometimes pull more data than expected, impacting system performance if not optimized.

More on Osquery basics

- Osquery website and schema
- Osquery official documentation
- The Osquery Project
- Introducing Osquery
- Host intrusion detection with Osquery - Osquery introductory talk
- Osquery Security Use Cases and Solutions
- 3 useful ways Osquery can help with security compliance



02 Getting Started With Osquery

Successful organizations all recognize one undeniable truth—the security and prosperity of business today depends on data. Effectively gathering, processing, and applying data is critical in today's competitive landscape.

This brings us to a critical question all organizations must ask. Is your team obtaining enough quality data to maintain the necessary insight into your operating systems and infrastructure?

If you're like most enterprises today, the answer is probably not. However, that's where Osquery comes in. Fondly described as an "esoteric tool" by Justin Mitzimberg, a senior security engineer at Uptycs, this operating system instrumentation framework excels at gathering large volumes of data. Created by Facebook in 2014, Osquery can provide a powerful, versatile, open-source option for organizations looking to scale data collection.

On average, Osquery generates approximately 110MB of data per endpoint, per day. This amount can vary depending on the functions of the monitored assets and the types of data being collected.

For organizations dealing with large-scale deployments, it is essential to consider the storage and management of this data, particularly when retaining it for extended periods. Data collected by Osquery can be stored in SIEM solutions like Splunk or centralized in a security analytics platform like Uptycs for comprehensive threat detection, compliance monitoring, and anomaly detection.

Despite its multitude of benefits, Osquery comes with many challenges starting with the technology's esoteric nature. Facebook open-sourced the agent but didn't follow up with in-depth instructions for use.

With Osquery, users face many decisions, including configuration, management, data storage, and maintaining visibility on their own. Below, we look at how to get started with this occasionally challenging yet deeply empowering data collection tool.

Osquery Pre-Install Considerations

Before downloading and installing Osquery, organizations must do their homework—starting with a thorough evaluation of resources and goals. The following are pre-install considerations your team should keep in mind before moving forward with Osquery.

SQL

Does your team have a comfortable working knowledge of SQL practices? Queries in Osquery depend on SQL, so having staff with these skills is critical.

Setup & Management

You will need a deployment infrastructure to deliver Osquery to your environment and a configuration plan in place to manage the tool. Keep in mind the system manager software you select should be capable of deploying the agent to the endpoint as well as managing ongoing configuration via files, over time. Popular solutions include traditional mobile device management (MDM) vendors such as Microsoft Intune or Jamf; DevOps tooling such as Ansible, Puppet, and Chef; and dedicated Osquery fleet managers such as Uptycs, Kolide, or osctrl.

Data Storage

Osquery can generate lots of data, so you will need a capable central storage location. Storage doesn't need to be Osquery-specific—it can be a centralized log saver like Elastic, Splunk, or Sumo Logic.

What's the problem you want to solve?

It's essential to begin with a clearly defined, addressable problem for your queries. Don't forget to map the Osquery schema to your specific issue—this will help you determine which data to gather and analyze.

Downloading & Installing Osquery

The Osquery system packages are straightforward and simple to download, but there are some significant differences between the three main operating systems. Let's take a quick look at each.

MacOS

Apple created an effortless installation with the macOS package installer. There are no package or library dependencies and you will be responsible for managing and deploying all updates. Please see here for more details on installing Osquery on macOS.

Linux

With Linux, a universal package is created for each distribution system. Packages contain Osquery(d) (the Osquery Daemon), shell, example configuration, and startup scripts. Please see here for more details on installing Osquery on Linux.

Windows

It's recommended to install the Windows version of Osquery using the Chocolatey package manager or the latest official binaries available here.

Remember, downloading an Osquery package is only your first step. You will also need a tool to run queries and push configurations out. Ideally, you want your system to pull down the configuration. In the event it doesn't, you will need third-party assistance to enable this function. You might consider engaging a security analytics platform that receives a configuration from the cloud or using a separate orchestration tool like Chef or Puppet.

Using Osquery For The First Time

First-time Osquery users should begin on the interactive command-line interface (CLI) known as Osquery. This introductory interface doesn't require any major configuration and it provides an excellent starting point for Osquery novices to begin working with queries.

Get started by running some basic local queries, such as **SELECT * FROM users or SELECT * FROM processes LIMIT 10**, and learn how the tables work.

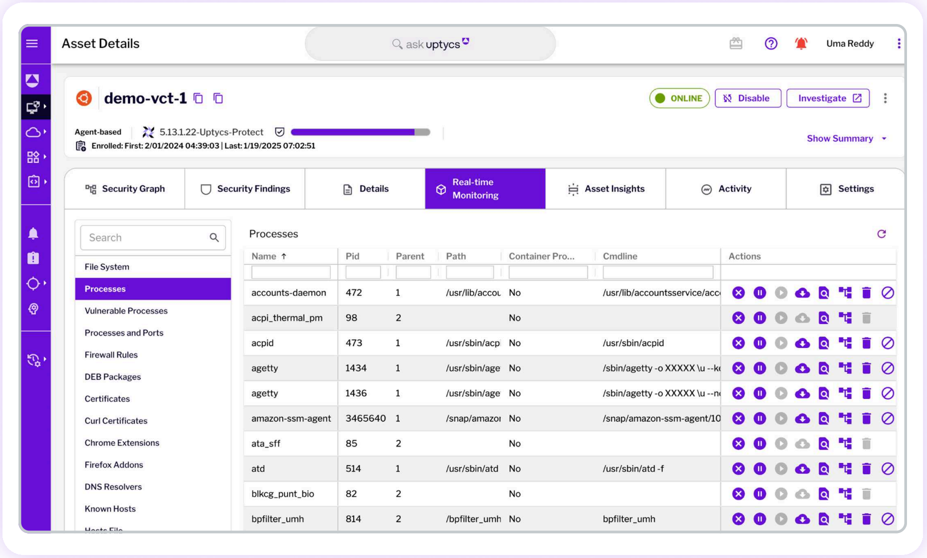
Continue by examining the default query packs (bundles of queries constructed to address common needs, such as compliance) and enabling those that spark interest.

Next, graduate to the daemon version of Osquery—Osquery(d)—and begin scheduling queries and working with packs related to specific areas of interest.

Browse through the Osquery website, especially the schema, to grasp the tool's immense power and functionality. The schema is where your organization can begin customizing the power of Osquery, tailoring queries to deliver specific data and insight. Exploring the schema is often the "ah-ha" moment when it becomes apparent the only limit to Osquery may be your team's imagination.

(Within Osquery itself, **PRAGMA table_info (tablename)** is also a useful tool for exploring Osquery tables.)

Best practices suggest initially working with regular tables—which offer point-in-time snapshots of your ecosystem and require less storage—then advancing to Osquery events, which store data in the included RocksDB database. Events use operating system features to record in real-time, which is critical for security but can generate volume that can overload systems. For this reason, you need to be careful with the configuration of the cache size and frequency of queries. It's crucial your team establishes a balanced rhythm—not so many queries as to cause overload, but enough to generate a steady supply of smart data.



Agent-based 5.13.1.22-Uptycs-Protect
Enrolled: First: 2/01/2024 04:39:03 | Last: 1/19/2025 07:02:51

Processes

Name ↑	Pid	Parent	Path	Container Pro...	Cmdline	Actions
accounts-daemon	472	1	/usr/lib/accou	No	/usr/lib/accountsservice/acc	
acpi_thermal_pm	98	2		No		
acpid	473	1	/usr/sbin/acp	No	/usr/sbin/acpid	
agetty	1434	1	/usr/sbin/age	No	/sbin/agetty -o XXXXX \u --ke	
agetty	1436	1	/usr/sbin/age	No	/sbin/agetty -o XXXXX \u --n	
amazon-ssm-agent	3465640	1	/snap/amazo	No	/snap/amazon-ssm-agent/10	
ata_sff	85	2		No		
atd	514	1	/usr/sbin/atd	No	/usr/sbin/atd -f	
blkcg_punt_bio	82	2		No		
bpfilter_umh	814	2	/bpfilter_umh	No	bpfilter_umh	

Despite its open-source affordability, systemic flexibility, and vast use-case potential, Osquery is far from simple. After the initial enthusiasm fades, your organization can feel alone and unsure without specific guidance. At this point, many organizations realize the immense challenge of centralizing data and maintaining visibility. After all, how does your team go from a log file to a dashboard that shows everything across an entire fleet of machines? These gaps can create frustration and drain momentum.

> **Enterprise-grade Osquery requires the ability to effectively store, access, and utilize data in an ongoing manner.**

Vast amounts of unusable data are worthless and successful enterprise-grade Osquery requires the ability to effectively store, access, and utilize data in an ongoing manner. Here is where engaging a security analytics platform, like Uptycs unified CNAPP & XDR, makes a lot of sense. Security analytics platforms can remove the heavy lifting by effectively handling data and maintaining visibility for maximum utility and impact.

Queries To Get You Started With Osquery

Organizations getting started with Osquery face an important question—what queries should my team begin running? We've put together a list of 10 suggested queries to help you get up and running.

Note: The following examples make use of JOINs, which can require an intermediate understanding of SQL.

Users (users) - Search for all local user accounts that exist on a machine.

- **macOS:**
  SELECT * FROM users WHERE username NOT LIKE '_%' ESCAPE '\';
  Note: Escape is required in this query, otherwise '_' matches any single character.

- **Linux (Ubuntu):**
  SELECT * FROM users WHERE gid < 65534 AND uid >= 1000;
  Note: uid 1000 is where users typically start on Linux.

Linux and macOS: The following query finds all users who have actual login shells (i.e. not "bin/false" or "bin/true", with allowances for them living in different places, /usr/bin/false or /bin/false, for example).

```
SELECT * FROM users WHERE shell NOT LIKE '%false' AND
shell NOT LIKE '%true';
```

Processes (processes) - This shows the top 10 memory hogs running on a system.

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

Process Open Sockets (process_open_sockets) - Searches for processes making network connections, other than web connections.

```
SELECT s.pid, p.name, local_address, remote_address,
family,  protocol, local_port, remote_port FROM
process_open_sockets s JOIN  processes p ON s.pid = p.pid
WHERE remote_port NOT IN (80, 443) AND  local_port NOT
IN (0) AND family = 2;
```

Listening Ports (listening_ports) - Looks for open ports on a system.

```
SELECT DISTINCT p.pid, p.name, l.port FROM listening_ports
AS l JOIN processes ON l.pid = p.pid WHERE l.address =
'0.0.0.0';
```

Chrome Extensions (chrome_extensions) - Search for the browser extensions running Chrome. (Since Osqueryd is typically running as root, this should be joined to the users table). The following query eliminates duplicates, and shows all extensions for all users:

```
SELECT DISTINCT c.name, u.username FROM users u JOIN
chrome_extensions c USING (uid) ORDER BY c.name;
```

Logged-In Users (logged_in_users) - Shows who is currently logged in to a system.

```
SELECT liu.*, p.name, p.cmdline, p.cwd, p.root FROM
logged_in_users liu, processes p WHERE liu.pid = p.pid;
```

File (file) - Returns information about the specified file on disk. The following queries, adapted from Fritz Ifert-Miller's blog post, finds all files in all user Downloads folders across macOS to one level of folder depth.

Equivalent queries for Windows & Unix are also noted below:

macOS:

```
SELECT file.path, users.username AS owner,
groups.groupname AS groups,
datetime(file.btime,'unixepoch') AS created,
datetime(file.mtime,'unixepoch') AS last_mod,
ROUND((file.size * 10e-7),4) AS size_mb FROM file JOIN users
USING (uid) JOIN groups USING (gid) WHERE path LIKE '/
Users/%/Downloads/%%' ORDER BY last_mod DESC;
```

Linux:

```
SELECT file.path, users.username AS owner,
groups.groupname AS groups,
datetime(file.btime,'unixepoch') AS created,
datetime(file.mtime,'unixepoch') AS last_mod,
ROUND((file.size * 10e-7),4) AS size_mb FROM file JOIN users
USING (uid) JOIN groups USING (gid) WHERE path LIKE '/
home/%/Downloads/%%' ORDER BY last_mod DESC;
```

Windows:

```
SELECT file.path, users.username AS owner,
datetime(file.btime,'unixepoch') AS created,
datetime(file.mtime,'unixepoch') AS last_mod,
ROUND((file.size * 10e-7),4) AS size_mb FROM file JOIN
users USING (uid) WHERE path LIKE 'c:\Users\%
\Downloads\%%' ORDER BY last_mod DESC;
```

Shell History (shell_history, for Mac or Linux) - Search for the executed commands on the system.

```
SELECT uid, username, shell, command FROM users JOIN
shell_history USING (uid);
```

Sudoers (sudoers, for Mac or Linux) - This displays sudo rules present on a system. Very simplistically, the second query noted below will address the question most people are likely to care about: Who can run commands as root? People with more complicated sudoers rules will likely know if they are specifying binaries, etc, and can use the first query or modify as appropriate.

```
SELECT * FROM sudoers;

SELECT * FROM sudoers WHERE rule_details LIKE '%ALL';
```

Collecting the right data requires asking the right questions. Justin Mitzimberg has some parting advice for perfecting queries: "Know the schema. For anyone getting started, they must know the data's location to know how to ask for it. The only way to ask the right questions is to understand how, and it comes back to knowing the schema."

Know the schema. The only way to ask the right questions is to understand how, and it comes back to knowing the schema.

Takeaways: Getting Started With Osquery

Osquery can deliver substantial business value by providing insight into the configuration, security, and behavior of your operating systems and infrastructure. This powerful data-gathering instrumentation framework shines with operating system flexibility and broad use-case utility.

But make no mistake—getting started with Osquery will require the dedication of skilled staff, deep resources, and a pioneering hands-on spirit to manage the tool and process the data with limited guidance. There isn't a one-size-fits-all recipe for delivering value—every organization must customize their version of Osquery, and partnerships with vendors is worth consideration.

The takeaway here is that running Osquery isn't for every organization—especially those looking for an easy plug-and-play panacea. There will be heavy lifting, especially when it comes to storage, visibility, and translation—ask any team about the challenges of getting data from management logs to digestible dashboards. However, for organizations with the staffing, resources, and commitment to succeed, Osquery undoubtedly proves its worth delivering invaluable insight that helps secure operating systems and infrastructure.

More on Osquery first steps

- Osquery official documentation
- Osquery schema
- Osquery query packs
- SQL introduction for Osquery
- How RocksDB is used in Osquery
- 10 pitfalls on the path to Osquery bliss
- How Osquery uses sqlite3 and RocksDB
- Catching everything with Osquery events

03 Osquery: Build vs Buy

At this point, you're well aware of Osquery powerful ability to collect valuable data. Your team is ready to begin harnessing the many benefits this unique tool can provide. However, you're facing a crucial decision—one that could determine the success or failure of your venture into Osquery.

To build or to buy? To embrace the challenge of building out your own agent, or enjoy similar functionality minus a bit of the heavy lifting by purchasing an Osquery solution from a vendor?

Some organizations chose the hands-on, build-your-own approach to Osquery—shouldering all the responsibility from agent deployment and configuration to accessing the data. Others prefer to team with a vendor—allowing a third party to handle the logistics, while still benefiting from the data-driven insight.

Both building and buying Osquery come with distinct advantages and disadvantages, and both can result in success. Ultimately, the decision to build or buy boils down to an organization's unique resources, needs, and preferences.

So, which path is right for your organization?

Below, we explore both options, providing knowledge and insight to help you make this important decision.

Building your own Osquery solution

Building with Osquery starts with a simple download of the open source agent. After installation, your organization has many choices for customizing Osquery. It can be easy to get overwhelmed with such decisions, but start with the following foundational considerations.

Configuration and Deployment

How will you deliver Osquery to the endpoints in your environment? Do you have the tools needed to reconfigure Osquery quickly and effectively? You will need a deployment infrastructure to push the agent out to all the endpoints at scale. Chef, Puppet, and Munki (macOS) can automate this provisioning and configuration step for independent users.

Inspection

You will need to inspect endpoints for data and possibly select tools to supplement the Osquery agent in this task. Prometheus is an example of such a supplementary data inspection tool.

Management

You will need a management tool for the fleet of hosts running the Osquery agent. Ideally, this tool will include a TLS server, generate commands at scale, and oversee scheduled query packs. Doorman and Fleet are examples of management assistant tools for independent builders.

Transport

Endpoint data collected by Osquery must move to a centralized management solution and then a data store. You must decide if your data will be sent over TLS or an existing logging pipeline. With logging pipelines, machines must always be on that same network, which can work for servers but not for workstations. TLS servers are the most common means of transport for an endpoint to management transfer in scaled Osquery solutions.

If using TLS, you have different choices. Kolide is an option to get up and running. If your team is proficient with Elastic, Splunk, or Sumo Logic these can also provide data transport options.

Storage

Once collected, data must be stored, and here is where building your own Osquery outlay can become costly. There are no uniform backend storage solutions. However, many organizations channel their data to SIEM repositories with providers like Elastic, Splunk, or Sumo.

It's important to keep in mind how much data Osquery can generate and the fact that some SIEMs charge by volume. This can result in high transfer and storage costs when large amounts of non-optimized data channel into SIEMs.

Visualization

How will you identify what's important in the data you collect? You must engage a tool like Kibana to organize and extract the valuable insight needed for fleet visibility, vulnerability management, compliance, and audits.

Additional Considerations For Osquery Builders

If you go the build route, be sure to consider these important questions.

What's your plan for updating Osquery?

Your Osquery deployment will require periodic configuration updating. Ideally, your deployment infrastructure tool will have a way to update the open source agent, but it may be necessary to activate changes manually. In this case, Julian Wayte, senior security solutions engineer at Uptycs, offers some advice for manual updates:

"After uninstalling the old version and installing the new version, it's important to maintain all configuration changes. Be sure to rewrite any custom configurations after deploying the new version to maintain continuity."

How will you manage and scale storage?

Backend storage isn't a set-and-forget proposition with Osquery. The continuing challenge of reaching and maintaining scale will require analysis, forecasting, and expansion. As an Osquery builder, you will need to allocate resources and staff to the management of this service.

How will you monitor the performance of your storage environment?

Like any other open source consideration, the performance of your storage environment must be monitored. An effective way to gauge performance is through the speed of questioning across your fleet. How quickly are questions receiving answers in your environment? A rapid flow of questions and answers indicates your storage solution is operating at optimum efficiency.

How will you monitor the performance of your Osquery deployment?

When it comes to general Osquery performance and what to watch, Julian Wayte has some advice for the do-it-yourselfers: "When monitoring the performance of deployed agents, keep an eye out for the memory or CPU hogs that are capable of swallowing up resources, overloading systems, and leaving you dead in the water."

Smart organizations actively employ resource utilization tools, like the native Windows Resource Monitor, to avoid the pitfalls of resource depletion and system overload.

How will you address bugs and new features?

For an organization to truly take advantage of open source Osquery, it will need to have C++ developers, notes Seshu Pasam, chief software architect at Uptycs. Developers are required to fix bugs or add features the organization needs. The alternative is to post bug fixes and requests to the Osquery Slack channel and GitHub.

It may appear easy and inexpensive to build Osquery yourself, but the devil is in the details.

How will you assess the costs of building?

It may appear building Osquery yourself is the most cost-effective means of acquisition. However, most fail to consider critical details, like the volume of data, that often isn't realized until deployment is underway. Julian Wayte reflects on years of watching customers discover how much the build-it-yourself option can cost:

"It may appear relatively easy and inexpensive to build Osquery yourself, but what happens is the devil is in the details. After you deploy the agent and start getting into deeper water, it becomes a bit of a 'gotcha' moment when the real challenges and costs of scale and handling larger volumes of event-based data become apparent."

Julian Wayte
Senior Security Solutions Engineer

In the final analysis—despite initial appearances—building osquery from scratch isn't necessarily the least expensive option.

The Complete Guide to Osquery

Learn more about the features and functions of Osquery, the open source universal endpoint agent.

Get the Guide

Considerations For Osquery Buyers

If you're looking for the functionality of Osquery without the responsibility or resource commitment of building, then buying an Osquery solution might be for you. You'll face fewer decisions than builders, but you'll also need to make important choices as a buyer.

Total cost is always a primary consideration, but what else should buyers of Osquery solutions consider? We've made a list of questions that all potential purchasers of Osquery solutions should keep in mind.

Can I be alerted when there is a specific event, combination of events, compliance or vulnerability incidents?

While Osquery is good for generating data and also stream events to an external log stream, one needs to consider the ability to make sense of the large amounts of data using analytics, monitoring and alerting systems around it.

Will it make configuration/deployment easier?

You'll want to purchase an Osquery solution that quickly and seamlessly provisions configuration, pushing the agent out to all endpoints at scale.

Will it include configuration management and the receiving and storing of data?

The Osquery solution you purchase should consist of robust endpoint management to oversee your fleet of hosts running Osquery. Endpoint management should control and monitor data transport from the endpoints to a centralized management solution and a data store.

Will it allow me to access data in the ways I need?

Flexibility is key here, so be careful to avoid solution packages that freeze your data. You want to purchase a solution where data smoothly flows in and then out into other systems. Make sure the solution uses an established query language like SQL for added flexibility.

In what format will I receive my data?

Make sure your Osquery solution offers diverse and thorough data visualization options. In addition, you want data in the most digestible and usable format for your organization's needs, so make sure the platform can deliver here.

Takeaways: Osquery Build/Buy Considerations

One of the most significant benefits of Osquery is the amount of flexibility the tool allows. Osquery provides powerful data collection benefits no matter which route you choose, either building or buying. Open source Osquery comes with the freedom to do as much as you desire on your own, without getting locked into a vendor relationship. Conversely, you can also enjoy the benefits a vendor brings without the responsibility or resource commitment of building out the Osquery agent on your own.

The takeaway here is regardless of whether you build or buy, there really is no wrong choice—it all depends on your organization's resources, needs, and preferences. You and your team should seek to understand both paths, the pros and cons of each option, and then select the one that best fits your organization. When it comes to the choice of building or buying osquery, it really is all about you.

More about building vs buying Osquery

- Osquery security solutions: Build or buy?
- Deploying Osquery at scale: A comprehensive list of open source tools
- Scaling Osquery with pure open source tech
- How Stripe is actioning the Osquery API at scale

Greenlight

"Uptycs contextualizes threat activity across K8s, cloud services, and laptops. We've dramatically shortened our threat investigation time."

Anwar Reddick
Director Of Information Security Greenlight Financial

04 Running Osquery At Scale

You know Osquery's value for gathering data, gaining visibility into architecture, and helping keep your organization secure. You're sold on Osquery and aware of the two options—either download and build using the open source agent yourself or buy the tool from a third-party vendor.

But what happens when your team wants to run queries across your entire fleet? What proves beneficial on a handful of endpoints can only be better on all your machines, right? So, can Osquery deliver on an enterprise scale?

The answer depends—mainly on your organization's decision to build out the open source workflow yourself or buy a fully-functional and enhanced agent from a third-party vendor. Many of Osquery's open source functions can scale, but there are limits. Osquery builders inevitably encounter a ceiling when looking to extend the agent across an entire fleet.

In this section we'll examine running Osquery at scale—what a full-featured, functional Osquery deployment looks like, what it can do, and what it takes to manage.

We'll also explore how Osquery can impact various advanced and emerging topics in the cybersecurity industry.

What does Osquery at scale look like?

A fully-functioning Osquery-based approach is a marvel of precision—the agent seamlessly deploying across an expansive fleet of assets, running queries, and automatically upgrading with minimal workload interference or resource depletion. Mass amounts of event data gathered across hundreds of thousands of machines, processed and analyzed on the backend in real-time.

However, scaling Osquery doesn't come without challenges, and volume is first on the list. Seshu Pasam, Chief Software Architect at Uptycs, explains: "Scale requires being able to handle, ingest, and make sense of high volumes of data. These high volumes, generated from scaling Osquery, can seriously challenge your organization if you're not prepared."

High volumes of data, generated from scaling Osquery, can seriously challenge your organization if you're not prepared.

The added workloads of scale demand more resources and skillsets, which can stretch budgets. More data also means higher SIEM bills, which can snowball and compound running Osquery at scale. Seshu notes: "With rising storage costs associated with increased query and event gathering, many organizations will cut back on the number of queries/ events, only running 50 tables instead of 300 to save money. But this can result in teams missing out on important data."

What Osquery At Scale Can Do

When Osquery is working at full capacity, it can address a variety of needs for your organization.

Security configuration management

Do you have insight into all the machines in your fleet? Who uses them, and where do they connect? How about what programs they run, and what data they hold? Scaling Osquery across all of your devices can provide this critical information, drastically simplifying fleet management. Your security team can see who is connecting and where, making unauthenticated and unauthorized entry more difficult.

Compliance

Compliance is a perfect example where the limited scope of open source Osquery leaves many builders stuck in a "what's next" quandary. The open source agent is capable of running queries to check compliance—just not many of them.

With open source Osquery, a small fraction of basic compliance checks can run, but the ability to scale is limited. Users working with a full-scale, Osquery-powered, analytics platform provider can run many types of compliance checks such as CIS Benchmarks, FedRAMP, and SOC 2.

Vulnerabilities

While package versions can be compared against known vulnerable versions using opensource Osquery, Uptycs provides a precompiled list of known vulnerabilities and intricate mechanisms of detecting them beyond version compare.

Detection Of Suspicious Behavior

The more endpoint devices Osquery covers, the higher your odds of discovering suspicious behavior and lateral movement before it becomes a threat.

Osquery excels at detecting anomalies across the enterprise, and the greater the scale, the easier to spot suspicious activity. When your organization scales deployments, increasing endpoint telemetry, security teams can correlate more data, which increases the odds of quickly identifying anomalies.

Threat Hunting

Threat hunting is an option when scaling Osquery, but the impact appears different for builders and buyers. Fully functional Osquery deployments play nicely with YARA, the tool used for malware research and detection.

The open source Osquery agent runs into limitations at scale because no standard deployment exists. Open source is endpoint software with functions differing for every user, and no two users will configure and operate the tool exactly alike. Seshu aptly sums it up:

“It’s hard to talk of scaling threat hunting, or any service for that matter, when there is no uniform way open source Osquery is deployed and used. Osquery has many flags that change the way it functions and how it enables or disables various features.”

When it comes to threat hunting, builders of the open source version must proceed independently with limited options in scaling Osquery across the enterprise.

It’s important to remember Osquery’s utility extends beyond just security. IT and DevOps teams can also benefit from Osquery’s versatility in data collection. Osquery tables provide necessary details for both hardware fleet and software inventory use cases.

Maintaining Osquery

The maintenance of Osquery across your fleet will be defined by your choice to build or buy the agent. As a builder of open source Osquery, you will need to shoulder all responsibility for incorporating certain functions into your management process. As a buyer of the tool, you will need to make sure you select a vendor that prioritizes the execution of these managing functions.

Here’s what you need to keep in mind to maintain your Osquery deployment across the enterprise.

Tracking Updates

Like all software, Osquery requires continual updates. You will need to keep an eye out across your fleet for updates as they become available. The open source agent has no uniform update protocol, so it falls on each user to determine the necessary actions.

Monitoring Fleet Status

Open source builders need to shoulder responsibility for two moving parts—the flags file, which dictates the configuration of Osquery, and the configuration itself, which governs operating details like tables and intervals. Open source builders must manage both, usually with the assistance of a third-party tool like Chef or Puppet.

Your team will also want to keep track of the resources your Osquery deployment uses. How much is too much? Also, be sure to confirm all machines function and deliver data properly.

Managing Osquery Performance

It's critical to keep an eye on the performance impact of the agent on your machines. Successful organizations put a great deal of thought into managing osquery performance. In a presentation at Osquery@scale 2020, Zachary Wasserman did an excellent job illustrating the depth of thought and effort that managing Osquery performance requires. His summarized takeaway is threefold:

1. Be strategic in how you present new queries.
2. Use the necessary tools to write queries that provide critical insights without burdening your systems.
3. Continually monitor query performance, starting from deployment and throughout the lifespan.

Understanding Cloud Security Posture Management

Strengthen your cloud security—discover how CSPM helps prevent misconfigurations and reduce risk.

Get the Guide

Advanced Osquery Topics

How does Osquery work with containers, cloud workloads, and Augeas? Can Osquery impact an organization's Zero Trust security posture? Let's explore the possibilities for your enterprise.

Containers

Osquery gathers details and provides visibility without installing anything inside a container. The open source agent provides a handful of tables that can look inside a container, but visibility is somewhat limited.

In an ideal environment, Osquery can provide some function with containers, but data gaps can exist. Your team can query most tables, but not all will be relevant, useful, or even applicable to containers. In short, Osquery can work in containers, but the process can be clunky and may not be the most effective way to gather data in these environments.

Cloud Workloads

Osquery doesn't care if you deploy on a virtual machine or in the cloud. However, keep in mind, while Osquery can be deployed on cloud services where compute is involved like AWS EC2, AWS ECS/EKS on EC2, etc., it is not applicable to services like AWS S3. Osquery does include a few tables specific to certain cloud providers.

Augeas

Osquery works with Augeas, a tool that reads and modifies configuration files. The Osquery interface includes Augeas tables, which allow for the in-depth search of configuration data across all endpoints in seconds. Osquery delivers peace-of-mind value by running Augeas queries to confirm the proper configuration of your entire fleet.

Zero Trust

Zero Trust is a hot-button topic with today's security professionals—and running Osquery can enhance your organization's Zero Trust posture from two angles.
First, before implementing a Zero Trust policy, you will need a comprehensive knowledge of all fleet endpoints. Who is running what applications? Where are they connecting? This information, which Osquery provides, is critical in the design process of your organization's Zero Trust environment.

Second, running Osquery proves useful in keeping tabs on your entire fleet. Queries can continuously check for malicious activity across the enterprise, delivering real-time reports to stay ahead of potential threats. Osquery data can provide irrefutable proof to validate your Zero-Trust state—and that's invaluable come audit time.

Conclusion

Osquery has proven utility in the modern enterprise—gathering data for valuable insight into systems and infrastructure. Fleet management, compliance, detecting suspicious behavior, and threat hunting are areas where the osquery agent can significantly impact your organization's security and efficiency levels. Osquery can also provide value when working with containers, cloud workloads, and in Zero Trust environments.

When it comes to scaling Osquery successfully, it boils down to your team's decision to build your own open source version or work with a third-party provider. Those building out the open source agent, especially when extending to higher functions or scale, may hit a ceiling of function and value. At this point, your organization may want to consider partnering with a vendor to harness the full capability of your Osquery deployment.

For those choosing to team with a full-service vendor platform, the benefits of scaling Osquery are well worth the investment.

More on running Osquery at scale

- Get started using Osquery for container security
- Use cloudquery and Osquery to simplify your cloud monitoring
- Kubequery brings the power of Osquery to Kubernetes clusters
- How Stripe is actioning the Osquery API at scale
- The Osquery Agent. Performance Tuning @Scale
- Osquery from a Cloud-Native and Zero Trust Perspective
- MuleSoft: Cloud governance and compliance with osquery
- Deploying Osquery@scale: Optimizing Telemetry Collection & Resource Utilization
- Monitoring ephemeral infrastructure with Osquery
- Osquery Across Compliance, Monitoring, Risk, and Threat Hunting

uptycs

Uptycs is dedicated to leading security innovations in hybrid cloud environments, ensuring robust protection and enabling our customers to innovate safely and efficiently. Included in the 2024 CNAPP Market Guide, Uptycs provides comprehensive security solutions that bridge the gap from code to cloud. Our platform excels in Cloud Workload Protection (CWPP), Vulnerability Management, Cloud Security Posture Management (CSPM), Detection & Response, Software Pipeline Security, XDR, and Risk & Compliance. Trusted by leading enterprises like PayPal and Comcast, Uptycs transforms potential vulnerabilities into fortified security, ensuring your digital environments are safeguarded from development through runtime.

Secure Everything from Dev to Runtime

Learn more at Uptycs.com