Osquery: The Basics

## **Task 1 – Introduction**

* **Osquery** is an open-source endpoint visibility tool developed by Facebook.
* Treats the OS as a high-performance relational database.
* Lets you run **SQL-like queries** to gather system information.
* Works across **Windows, Linux, macOS**.

## **Task 2 – Connect with the Lab**

* Ensure connection to the lab VM or test environment where Osquery is installed.
* Can run via **CLI** (osqueryi) or as a service.
* Lab setup ensures safe experimentation without affecting your host OS.

## **Task 3 – Osquery: Interactive Mode**

* Launch using:

bash

CopyEdit

osqueryi

* **Interactive Mode** allows live SQL queries.
* Example:

sql

CopyEdit

SELECT \* FROM processes LIMIT 5;

* Use .help for available commands.
* .tables lists all available tables.
* .schema <table> shows table structure.

## **Task 4 – Schema Documentation**

* Each **table** represents a set of OS data (processes, users, network, etc.).
* **.schema** shows columns and data types.
* Example:

sql

CopyEdit

.schema processes

* Official Osquery schema reference: https://osquery.io/schema

## **Task 5 – Creating SQL Queries**

* Osquery supports **standard SQL syntax**.
* Example queries:

sql

CopyEdit

SELECT name, pid FROM processes WHERE name LIKE '%chrome%';

SELECT \* FROM users WHERE username='administrator';

SELECT address, port FROM listening\_ports WHERE port=80;

* You can **filter, join, group** like in normal SQL.

## **Task 6 – Challenge and Conclusion**

* Challenge involves:
  + Exploring .tables and .schema.
  + Running custom queries to gather process, user, and network information.
* **Conclusion**:
  + Osquery is powerful for endpoint investigation.
  + Works well for **threat hunting**, **incident response**, and **asset inventory**.
  + SQL familiarity is key to using it effectively.

    