Research: MLS PostgreSQL

Table of Contents

Overview

One of the four main components of the COPS Platform provides secure Data Storage that meets Multi-Level Security guidelines.

The initial Data Storage component will contain PostgreSQL with support for MLS SELinux labels. There are two main options:

- Crunchy MLS PostgreSQL
  - https://www.crunchydata.com/products/crunchy-mls-postgresql/
- SE-PostgreSQL
  - https://wiki.postgresql.org/wiki/SEPostgreSQL_SELinux_Overview

Specific Pros and Cons are listed in this decision log: Which MLS Database should COPS use?

- The Decision Log includes the Oracle Database with MLS extensions. While not strictly based on PostgreSQL, it is still worth comparing.

This page focuses on research related to the Crunchy Data offering and SE PostgreSQL

Most of the research was completed as part of **COPS-99** - Getting issue details... STATUS

Crunchy MLS PostgreSQL

Crunchy Data is one of the partners that helps maintain PostgreSQL. They offer a number of support-based contracts and usability improvements for use with PostgreSQL.

Details

There is not much information available on their website, but there is a slide deck from 2016 that describes the work-in-progress to add row support to the sepgsql Provider:

- http://joeconway.com/presentations/mls-postgres-scale14x-2016.pdf
- Also attached in case the link goes dark:

So it looks like they just added row-level access control to the sepgsql provider, and they keep that separate to offer as a commercial add-on for PostgreSQL.

Phone conversation with Crunchy Data

Danny sent an info request through the website on Jan 2, 2020 and got a call back from Bob Laurence (the CEO of Crunchy Data). Danny described our application simply as one where multiple applications need to pull data out of a database based on security labels for a U.S. Government customer. No more details than that and no commitments. Here are notes from the discussion:

- Crunchy has worked with a number of government organizations to provide an integrated and security-hardened solution for Multi-Level Security databases.
- They are the most trusted solution for MLS Databases (among their customers, at least).
- The MLS components are not open-source to the outside world because their customers want to protect the capability. It goes above and beyond the capabilities of the open-source SE-PostgreSQL.
    - It integrates with SELinux in a unique way
    - Row-level access control
    - They created a Management GUI for dealing with security labeling (to make it more user friendly)
    - But the source code is available to government customers that have paid for the product
- According to Bob, it takes a 2 hour briefing to summarize the differences between their MLS solution and that of the open source SE-PostgreSQL.
- In terms of how we would work together:
    - Crunchy Data could be a subcontractor to Skyward Federal, providing integration support and a copy of the software for the government users.
    - It is possible that Crunchy already has a relationship with the office that COPS is supporting. If so, then we could leverage that relationship for integration support to use Crunchy's PostgreSQL.
    - Once we have some concrete plans to use Crunchy Data's solution, we should contact them and work out more details.

References

- Marketing page claiming row-level access control
    - https://www.crunchydata.com/products/crunchy-mls-postgresql/
- Github repos (none seem related to SELinux or MLS)
    - https://github.com/CrunchyData?utf8=%E2%9C%93&q=&type=&language=
- Product announcement from 2015 (which mentions "the U.S. Government's Centralized Super Computer Facility")
    - https://www.prnewswire.com/news-releases/crunchy-data-solutions-announces-open-source-multi-level-security-enabled-postgresql-300104763.html
- Conference presentation from 2016
    - https://postgresconf.org/conferences/2016/program/proposals/mls-postgresql-implementing-multi-level-security-in-postgresql-with-rls-and-selinux

SE-PostgreSQL

In the current version of PostgreSQL, the MLS SELinux labeling capabilities are handled through a provider named 'sepgsql'. It has some specific limitations (and has had the same limitations since PostgreSQL 9.3 in 2014 (according to the documentation, at least)).

Details

SELinux labels (security contexts can be associated with):

- Databases
- Tables
- Views
- Columns

Labels are changed using a SQL command named 'SECURITY LABEL.'

Limitations

- Data Definition Language (DDL) Permissions
    - Due to implementation restrictions, some DDL operations do not check permissions.
- Data Control Language (DCL) Permissions
    - Due to implementation restrictions, DCL operations do not check permissions.
- Row-level access control
    - PostgreSQL supports row-level access, but sepgsql does not.
- Covert channels
    - sepgsql does not try to hide the existence of a certain object, even if the user is not allowed to reference it. For example, we can infer the existence of an invisible object as a result of primary key conflicts, foreign key violations, and so on, even if we cannot obtain the contents of the object. The existence of a top secret table cannot be hidden; we only hope to conceal its contents.

The source code for sepgsql is not very complex, so one option is to fix some of these limitations and contribute the changes back to the PostgreSQL project: https://git.postgresql.org/gitweb/?p=postgresql.git;a=history;f=contrib/sepgsql;h=23256de641cae637649f7a22fbe659e123266f4a;hb=HEAD

References

- Transition from internal patches to a separate MLS Provider
  - https://selinuxproject.org/page/NB_SQL_9.3
- Specific limitations of the current provider implementation
  - https://www.postgresql.org/docs/12/sepgsql.html#SEPGSQL-LIMITATIONS