Open Raven

Building Open Raven’s Data Store Fingerprinting (DMAP)

February 2020

Overview: Perception vs. Reality

TABLE OF CONTENTS

Historically organizations have been led to believe the most significant risk of a data breach comes from external attackers. In reality, the majority of data breaches happen as a result of internal mishandling of data. In fact, in 2019, accidental data breaches eclipsed the number of intentional hacks for sheer amount of data exposed. Furthermore, by 2025, 99% of cloud security failures will be attributed to end users, not their cloud service providers (Gartner).

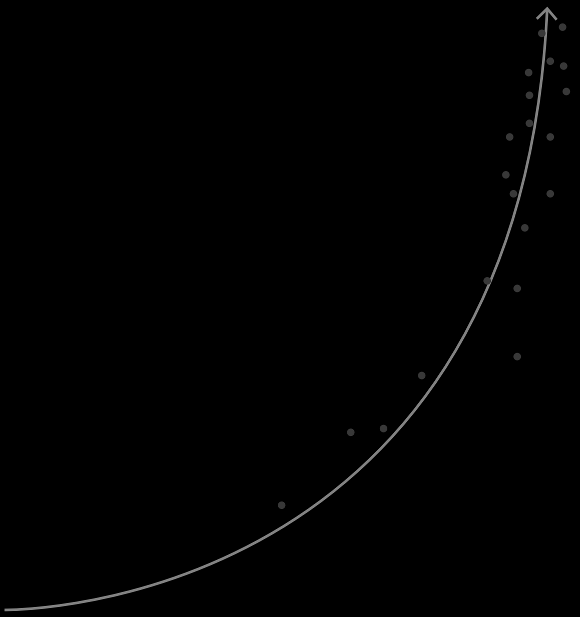
While data misuse and accidental breaches bear no malicious intent, they do have serious ramifications for the consumers and entities whose data is compromised. They can also have an equally devastating effect on the organizations responsible, who suffer brand reputation decline, loss of revenue, and worse yet, the loss of customer trust. In the two years since its inception, GDPR fines for data privacy violations have eclipsed $126 million and the statute has resulted in over 160,000 breach notifications.

If one of the driving factors behind data breaches is mistakes made by organization's protecting the data, the most straight-forward means of preventing them is to make it easier to secure the data itself and consequently avoid the resulting mistakes. This effort should begin with simply understanding what data repositories are present in an organization. This is the focus of Open Raven's DMAP service and this whitepaper.

The Data Breach Epidemic

A major cause of the data misuse problem is that today's Infrastructure is increasingly owned by developers, not IT. These developers are moving faster than security can keep pace, and their incentives typically prize speed or completeness over hygiene. The consequence? In 2019, over 6 billion records were made freely accessible not because of cyber infiltration from state-sponsored adversaries, but because of misconfigured databases, backups, endpoints and services (Dark Reading).

This issue is multiplied by the sheer volume and variety of data available. Every 2 years, the amount of data in the world doubles (IDC). By 2025, it will double every 12 hours. Even the organizations creating this data can't keep up with it – 70% of this data goes unused.



What's more, not every asset created is equal, with some – like data-bearing systems – demanding much more focus than others. A lab device, for example, isn't the same as a database of customer data that supports a critical application. If the information within that database becomes exposed, it requires immediate action across an organization, for example:

- Security, IT, Legal, Communications, Executives and others must mobilize to respond to the issue accordingly.
- Notifications must go out to customers, partners and everyone else impacted (which leads to subsequent loss of trust in the brand).
- Regulating entities may enforce penalties on the organization for breach of compliance (e.g., HIPAA in healthcare and FDIC in finance, etc.).

The damaging result of such an exposure is that individual lives are impacted and jobs are negatively affected. So, a functional heuristic for minimizing any chance of data exposure and its consequences is to start with knowing what type of data, and how much of it, you have in your organization, and most importantly, where that data is stored.

Answering the Question: Where's My Data?

One of the most effective ways to minimize risk of data exposure is for organizations to have a constantly updated view of their data stores which allows them to take appropriate action quickly when an irregularity is identified. Said another way, they should have a ready means of answering the question of "Where is my data?" and ultimately "How is it protected?" These questions are deceptively hard to answer for many reasons:

- Data is growing at unprecedented rates.
- Data is commonly duplicated for not only backups, but also for technical support and data science efforts to glean insights the organization can use across business functions, like customer support and sales lead-generation.
- A large (and growing) number of people handle data – from DevOps, to IT, to security teams and now even the boardroom (Risk & Insurance).
- Most organizations straddle on-premise and the cloud infrastructure (IaaS, SaaS) with no unified view into what is moving between the two or how data is being stored.
- Responsible data handling practices are often not a priority within an organization, lacking in training, tools and general awareness.

The issue with so many people invested in an organization's data is that best practices for handling that data are often unclear. Who should have access to that data? When should they have access to that data? Who are they sharing that data with? By what means are they sharing that data? Will they be stripped of that access when they leave the organization and how should that access be removed?

Casting a wide net

A brief exploration of where an organization of at least modest size is storing data further reveals why it is hard to answer fundamental questions about data security. Firstly, traditional relational databases have not disappeared. They are alive and well, if not swelling with data that spans many years. They're in both the cloud and in on-premises where they can grow to sizes that defy easy analysis. They sit on laptops and servers supporting applications both simple and complex and at times can even be found supporting Internet of Things devices as either dynamic or persistent storage.

NoSQL data stores (e.g., ElasticSearch, Hadoop, MongoDB) are also commonplace on corporate networks and even more so in cloud environments. The type and behavior of non-relational data stores has exploded in recent years due to a variety of factors but perhaps most meaningfully the popularity of data science and trend towards using data pipelines to support modern applications. Data pipelines allow organizations to combine a variety of data repositories into a single back-end with each data store being used for a specific purpose well suited to its specific attributes (e.g., Postgres for driving the UI, Druid for time series data, S3 for backups, etc.). Data Science drives a massive volume of data whereas a visible trend towards data pipelines is a force behind data store heterogeneity inside an organization. And the trend towards new types of data stores should be expected to continue unabated as graph data back-ends gain popularity, SQL databases adapt to cloud native environments (e.g. cockroachDB) and so on.

Wrapping your arms around where and what types of data stores is a challenge that goes well beyond finding the usual RDBMS and NoSQL data stores for structured data. There are massive, often even larger piles of unstructured data on file servers from NetApp appliances to SharePoint. SaaS applications are increasingly taking over the load from on-premises file stores and placing unstructured data in the hands of third party providers in the cloud.

Thus, building a map that identifies and plots the data stores of a modern organization can be incredibly challenging, requiring one to explore many different areas, from cloud to on-premises to partner networks, while encountering an incredibly diverse set of repositories, each with its own unique attributes. The tools we have historically to tackle this problem typically leave us with a best guess at the operating system along with the running ports and services. This is a far cry from a clear label of a data repository and leaves considerable manual effort to the user to determine what's actually running on a server, instance, container, etc.

> If data is "the new oil" as many assert, why haven't we put more effort into properly locating and labeling our oil wells?

Answering the wuestion:
Where’s my data?

Yesterday’s fingerprinting

Beyond the sheer complexity of current environments is the fact that they are also frequently changing due to a move to faster development cycles (i.e., Agile and DevOps) and transitions to much more dynamic cloud environments. Classic fingerprinting techniques such as those used for identifying operating systems by gauging TCP, UDP and ICMP behavior presumed fairly static OS behavior. If we are to properly identify a data store, we can’t necessarily afford the same luxury of presuming behavior will remain as it is today. Yesterday’s accurate fingerprint – a tactic used to correlate data sets to identify network services, operating system number and version, software applications, databases, configurations, etc. – may no longer deliver the same results as before, requiring both broad visibility and regular updates to maintain efficacy.

Answering the question:
Where’s my data?

The permissions problem

What type of data store attributes one can use to positively identify a data store will also depend on the level of access to the network and host system. Ideally, one has full administrative access that allows for examining the file system or even network traffic.
The reverse is often the case, where minimal access is available restricting analysis techniques to checking for listening ports and exploring what can be gleaned by probing the service.

Given that the goal is to discover the unknown data stores that may become future data leaks, the scenario where one possesses nothing beyond a handful of ports to interrogate remotely introduces added risk. This is why a data repository fingerprinting service must leverage a wide set of attributes and be able to state its conclusions in probabilistic terms, from low to high confidence. Ideally, it would also adjust the confidence ratings over time as more analysis is performed and considered.

Choosing the approach

Application fingerprinting (i.e., TCP/IP fingerprinting) is not a well-studied subject compared to its predecessor, OS fingerprinting. While TCP/IP (sequence number, TTL values, etc.) is a well-defined standard, application protocols have no enforced commonality, so TCP/IP fingerprinting provides an ineffective model for framing an application fingerprinting problem such as positively identifying data stores.

Ideally, the solution shouldn't rely on applications running on their standard ports. We need to generate a certain degree of entropy through application behaviors in order to create enough data to positively identify a data store. This is similar to creating a map of a fingerprint for physical security – the user must re-position their finger on a reader (e.g., the iPhone screen) several times to ensure it is read and properly mapped from enough angles.

Some application protocols speak HTTP, which generally offers a wealth of entropy sources such as status codes, header values and content. CouchDB, for example, offers the following.

```
> nc couchdb-server 5984
GET / HTTP/1.1

HTTP/1.1 200 OK
Cache-Control: must-revalidate
Content-Length: 208
Content-Type: application/json
Date: Thu, 30 Jan 2020 02:11:14 GMT
Server: CouchDB/2.3.1 (Erlang OTP/19)
X-Couch-Request-ID: 2491a24c5c
X-CouchDB-Body-Time: 0

{"couchdb":"Welcome" <content snipped>}
```

Other applications remain silent until their protocol is spoken, which presents another challenge. The banner identification is reliable, but it requires manual labor since it doesn't scale and is only a small fraction of the solution since most services don't even present banners.

Protocols may be binary or text-based, or a mixture of both. For example, the MySQL banner contains both binary values and text strings (8.0.18 is the MySQL version, while the authentication method is caching_sha2_password). Other binary values here indicate packet length, sequence numbers, nonces, etc.:

```
> nc mysql-server 3306
J
8.0.18  <0O6ar7•••••k9>#ri51caching_sha2_password
```

With all these variables, building manual pattern detection (regexes, etc.) doesn't scale and is cost prohibitive.

Open Raven Data Store Fingerprinting (DMAP)

Deploying effective machine learning

DMAP is a classifier at its core. While developers can write a manual classifier to match up fingerprint characteristics, this is a task well-suited to machine learning algorithms for several reasons. First, the number of data stores is large and constantly expanding making it diminishingly reasonable to manually keep up. Similarly, the number of potential attributes will exceed the human capacity to discover patterns. Finally, many data stores are derivative or exhibit similar behaviors, meaning there may be only nuanced differences available to differentiate them.

Once machine learning was chosen to be the classifier engine for Open Raven's DMAP, several algorithms were considered. Linear and non-linear regression were not appropriate choices because the data in this case is categorical, and these classifiers are better suited for numeric data. Neural networks were also considered but have several shortcomings for the problem domain, the most prominent of which is that neural networks are not well suited to non-binary classification inputs (for example, HTTP response codes).

A decision tree (and later a decision forest) was chosen for its characteristics that made it well suited for this type of classification problem:

- It provides accurate results even with small data sets, allowing theories to be tested early on with little time investment.
- It provides predictions based on probabilities, meaning more nuanced results. This enabled DMAP to pass along the confidence in the application classification, articulating for example where it may have 95% confidence that an application is MySQL vs. a situation where it's only 51% confident. This type of transparency is key to building trust in the results.
- A decision tree is tolerant of dirty data, allowing for factors such as varying network latency, lost packets or early disconnects.
- A decision tree provides probabilistic results which offer a reasonable (and labeled) "guess" for edge cases. Users can see how much confidence Open Raven's model has with the provided results.
- It is conceptually easy to understand and interpret by a human for smaller dimension data sets, meaning Open Raven's team could verify the efficacy of DMAP in early development.
- To overcome some of the decision tree's limitations, a "random forest" was chosen. This provides protection against overfitting, where a tree is over-customized to fit the training data at the expense of providing accurate guidance on unseen data.

Building a machine learning classifier requires sample data. Given that there are no readily available datasets, the first step in building Open Raven's DMAP was to create it internally. To do this, Open Raven devoted considerable resources to developing a scalable fingerprint ingestion workflow that utilizes both cloud and on-premises resources (see below for Fingerprint Ingestion).

> Against a running application, the Open Raven Fingerprinter will make multiple connections over TCP, each connection performing a specific fingerprint (test). These fingerprints are collected and stored on a per-application basis, mapped to a known application type.

Model building

When a decision tree model is built, fingerprints are converted into machine learning features before being fed to the training model. Each fingerprint will result in one or more machine learning features, as shown below:

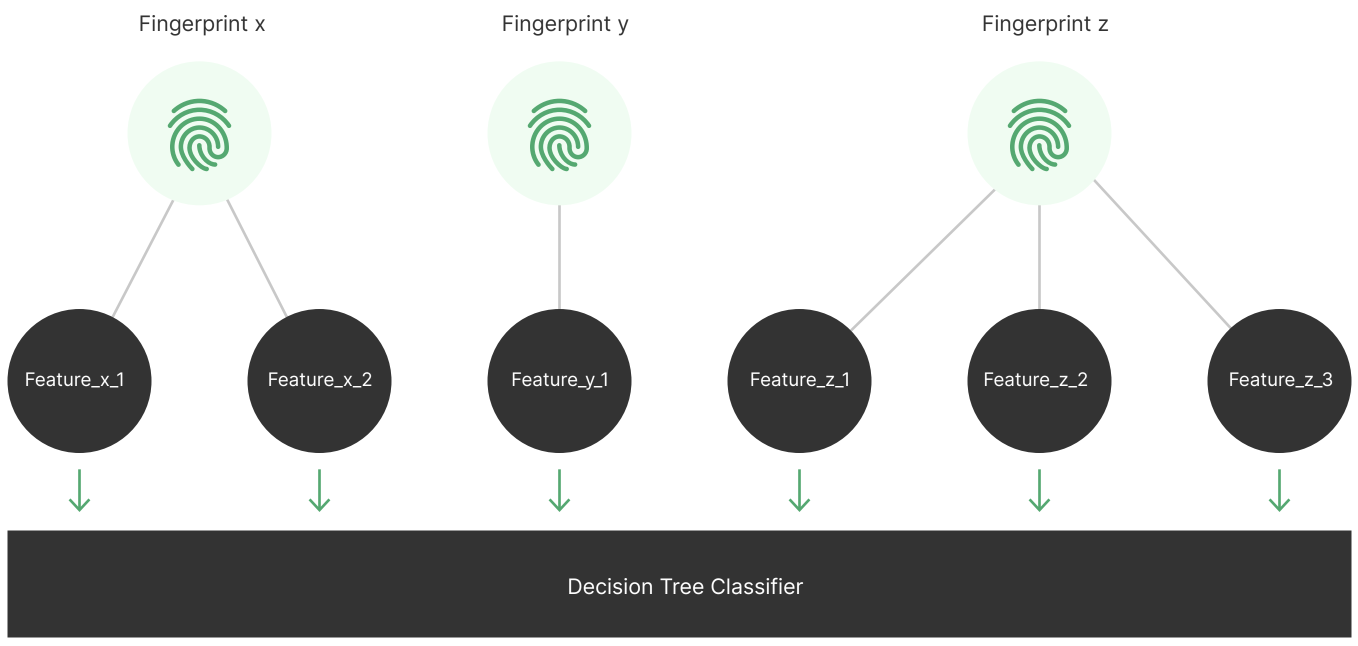


Figure 1: Fingerprint Features

The above is repeated for each application ingested and finally split-tested to ensure accurate predictions.

Fingerprint ingestion

Fingerprint ingestion is Open Raven's primary method of supervised training. The cloud-based ingestion workflow is shown below (Figure 2). This flow utilizes AWS Fargate with Docker containers, allowing Open Raven to quickly spin up application infrastructure for fingerprint ingestion and to scale it down when no longer needed.

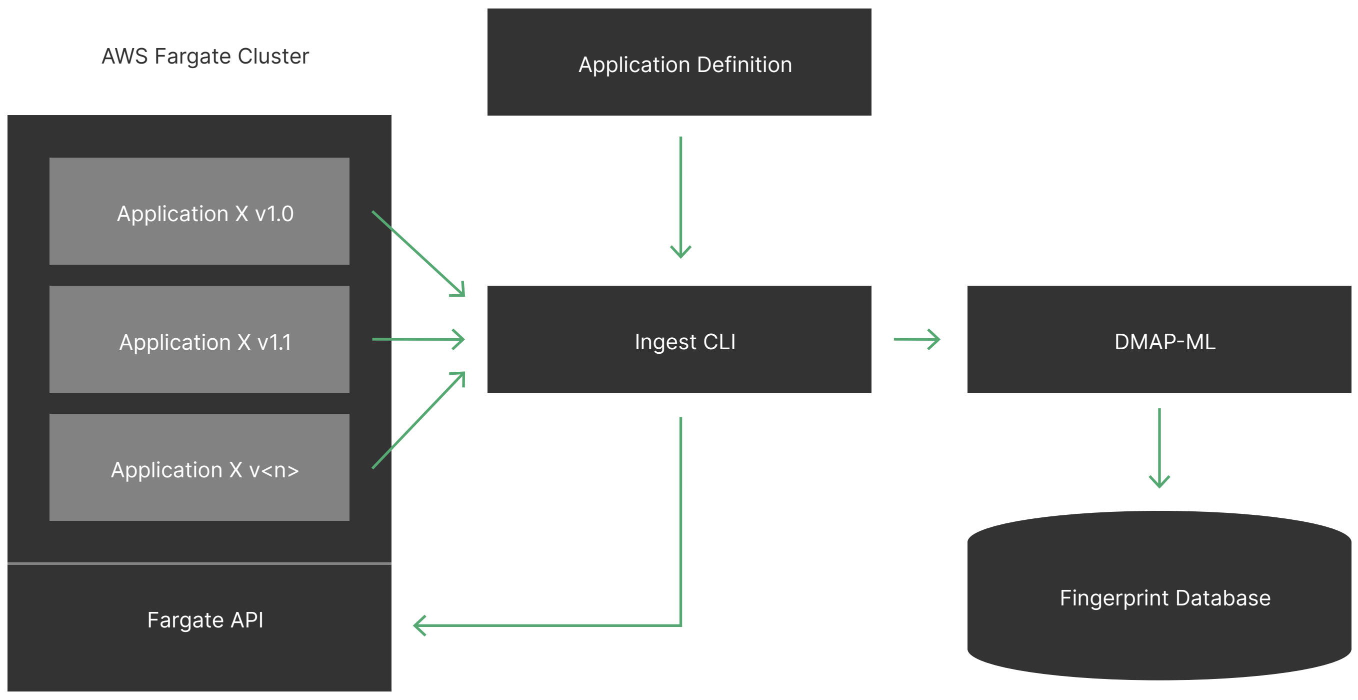


Figure 2 - DMAP Ingestion Flow

User Contributions

While Open Raven's initial dataset and model is based on fingerprint ingestion, its workflow design allows (and encourages) user contributions and refinement. The flow looks like Figure 3 below. When application predictions are shown to the user, they can override existing predictions with their own feedback. This feedback loop is integrated in real-time into Open Raven's DMAP predictive engine in DMAP-ML, allowing DMAP to leverage the newly gained knowledge for future predictions.

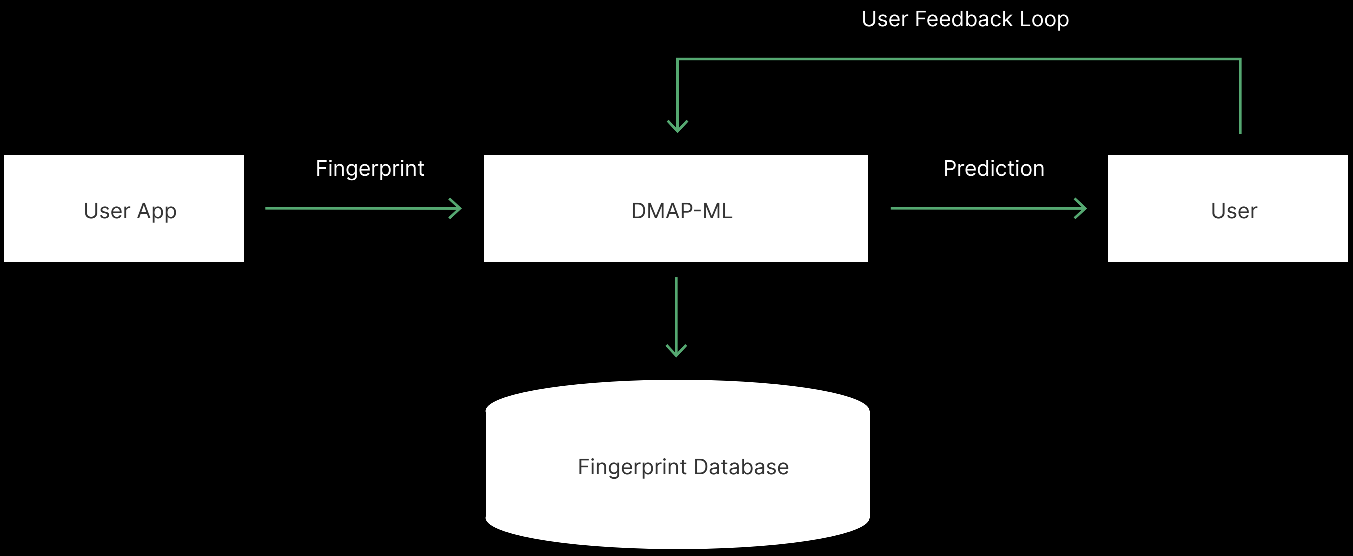


Figure 3 - Feedback Loop

A user feedback loop both enhances Open Raven's ability to predict already known applications, especially in yet unseen versions or editions. This enables Open Raven to learn about new software applications in real-time as users ingest and provide feedback.

Building Open Raven's DMAP

To understand how Open Raven's DMAP works, we need to examine its inner workings and how it is built.

DMAP is a cloud-centric and distributed architecture. DMAP-ML runs inside the Open Raven management cluster (Asgard), while DMAP runs within the customers' Open Raven (Odin) cluster. For users that wish to map their enterprise (non-cloud) networks, the DMAP-Scanner runs locally on-premises and feeds back to DMAP.

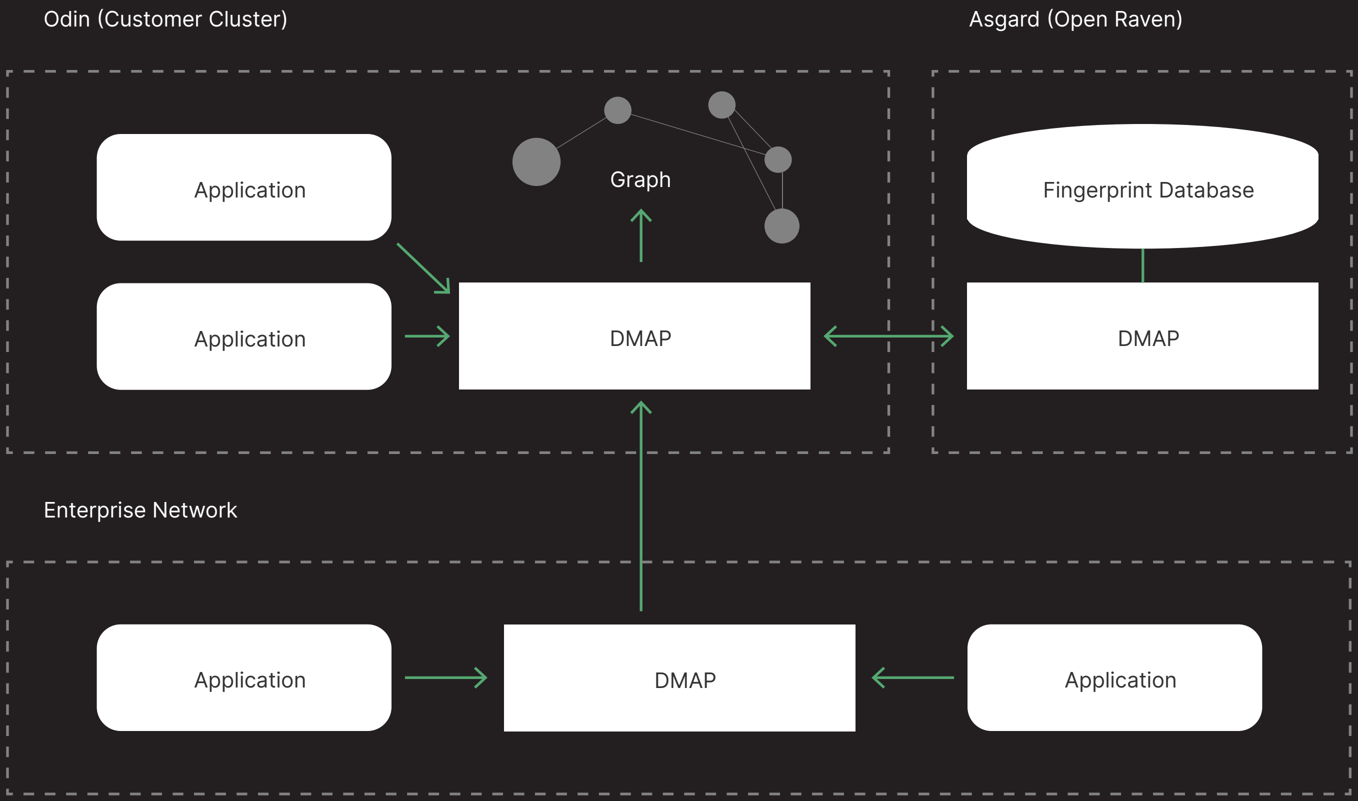


Figure 4 - DMAP Architecture

Managing Data Store Profiles

Data stores can be bucketed into four main categories based on how noisy they are:

Church mouse quiet

Postgres will sit on an open connection and offer no responses until you speak its language. This may disconnect after a certain number of characters read, or it may never disconnect at all.

Speaks only when spoken to

Redis will sit quietly on an open socket but will happily give you an error message when you send data that isn't a valid protocol frame.

Gabbers

On connect, MySQL and SSH both readily send a banner that identifies the application, version and other service data.

HTTP hipsters

Splunk and Mongo offer a wealth of entropy by responding to HTTP requests.

These different types of data stores necessitate tailored approaches to match their unique "personalities" and ensure accurate identification. For those applications that require reception of a valid protocol frame before sending responses, conventional fingerprinting is limited in accuracy. In order to boost accuracy, Open Raven is developing application-specific fingerprint tests. While this approach will boost identification of these services, it doesn't scale as it requires considerable engineering research and development time.

Amazon Web Service (AWS) Fargate, on the other hand, enables Open Raven to dynamically spin up thousands of services without requiring fixed infrastructure. Combined with the fact that the overwhelming majority of existing data store applications are available as Docker images, it's easier than ever to generate sample data. Unfortunately, not all Docker images are compatible with Fargate. This means Open Raven had to develop secondary methods of generating large numbers of sample sources.

Future Open Raven DMAP Enhancements

DMAP is currently in its early days and will evolve and improve over time. In its next phases, some bigger picture items Open Raven will tackle are:

Port Correlation

Currently, Open Raven's DMAP assesses each port individually. This works well for applications that reside on a single port (MySQL, PostgreSQL, etc.), but leaves entropy on the table for services that listen on 2+ ports (Hadoop, MSSQL, etc.).

Banner identification with neural networks

Currently, Open Raven's DMAP assesses each port individually. This works well for applications that reside on a single port (MySQL, PostgreSQL, etc.), but leaves entropy on the table for services that listen on 2+ ports (Hadoop, MSSQL, etc.).

Conclusion

The foundation of protecting your data is knowing where it resides, how it is stored and how it is being secured. This is no mean feat in a modern enterprise that straddles cloud and corporate networks, has hundreds of different types of data stores, myriads of people handling sensitive data, third parties who both store and handle data as well... all while regulators watch with increasing scrutiny and the media report lapses in the daily news. Simply put, it has been far too hard with available solutions to maintain basic levels of data security.

DMAP from Open Raven is a step forward to regaining control of data protection by making it easier to understand what data stores are present in any environment. Once data locations can be identified, they can be inventoried and classified. They can then be assessed. They can be secured. And then, breaches can be avoided and we can return our attention to where it belongs: unlocking the potential of the data itself.