databricks

Threat detection on Kubernetes using GNN embeddings

CAMLIS 2023

Arjun Chakraborty
Staff Engineer, Databricks

©2022 Databricks Inc. — All rights reserved

About me

Security ML

And

Threat detection

(Twitter/X : @sfrosagulla)

©2022 Databricks Inc. — All rights reserved

Agenda

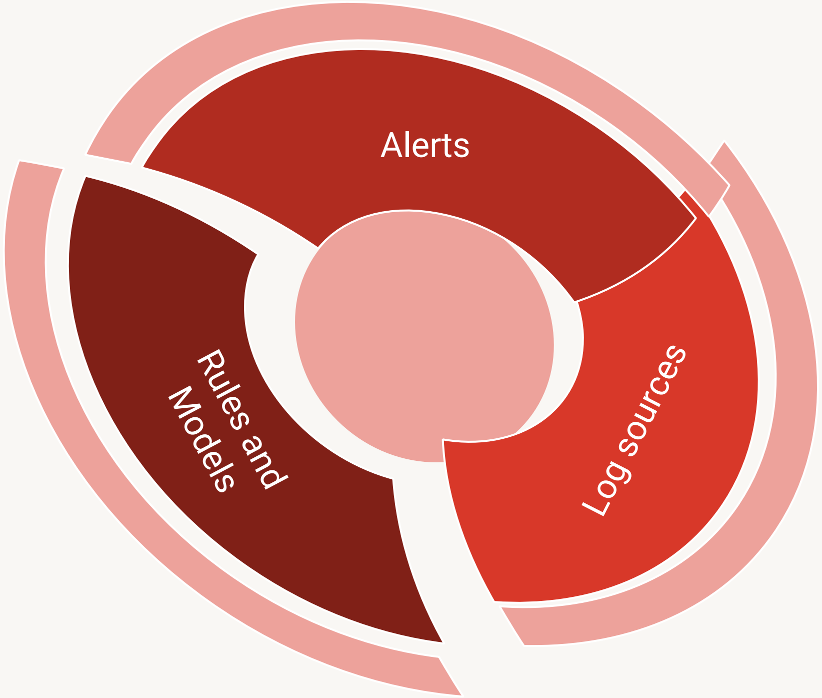
- Problem statement
- A primer on Kubernetes (K8s) logs
- K8s logs as graphs : The how and the why
- Using GNNs to build out embeddings : A walk through
- Threat hunting with GNN embeddings
- Challenges and looking ahead
- QnA

©2022 Databricks Inc. — All rights reserved

Problem statement

Static detection rules need constant tuning

Low Signal to noise ratio when detection rule complexity increases

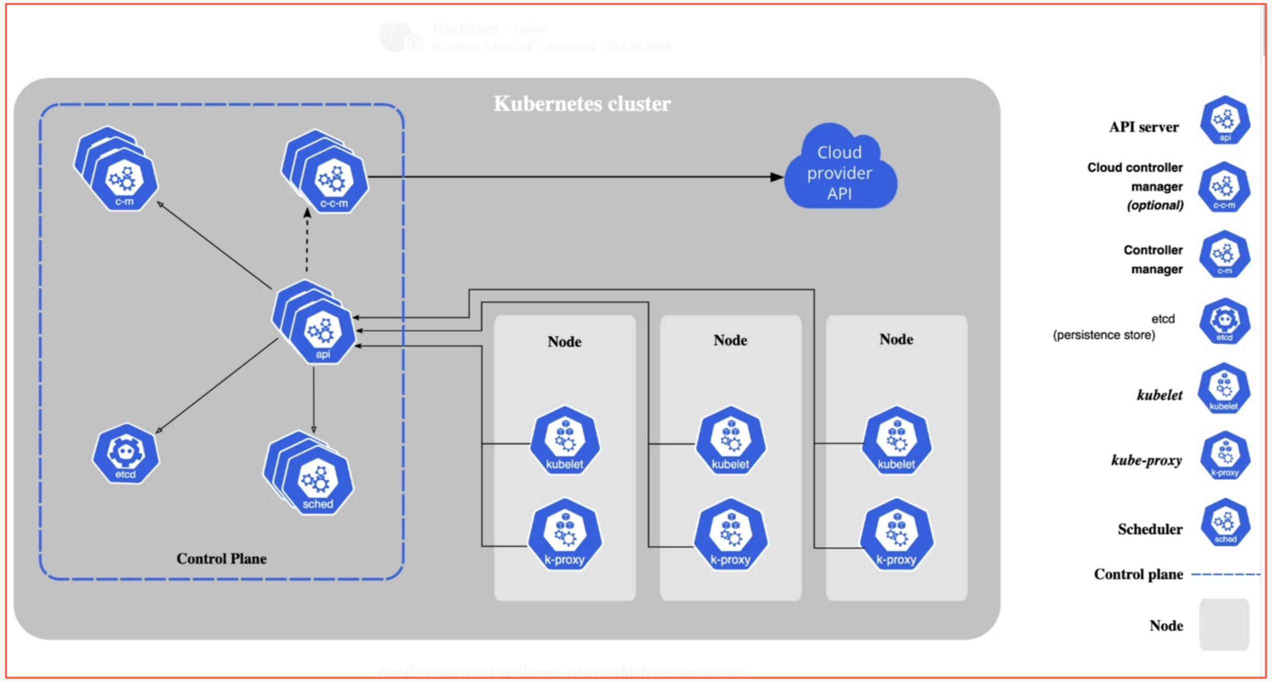


Static detection rules do not have structure or context encoded

No labeled data to build out a supervised learning problem

©2022 Databricks Inc. — All rights reserved

Monitoring the Kubernetes (K8s) API server



- The core of the Kubernetes control plane is the API server.
- The API server exposes an HTTP API that lets end users and different parts of the cluster, and external components communicate with each other.
- Kubernetes API server generates audit logs that are used for security monitoring.

What?

- What user/system is generating what traffic?
- Which users access production frequently and why?
- What requests are being rejected and why?
- How do we monitor for malicious activity within the Kubernetes cluster?

Why?

Source : https://medium.com/pareture/monitor-kubernetes-api-server-audit-in-eks-3e8e6e18e7fb

©2022 Databricks Inc. — All rights reserved

A primer on K8s audit logs

@logStream	kube-apiserver-audit-7fee5e06a15b28e78fd4fea0eae08606
@message	{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"M
@timestamp	1648236440454
annotations.authorization.k8s.io/decision	forbid
apiVersion	audit.k8s.io/v1
auditID	4c180539-f6c3-4f96-b30d-82eed2009f05
kind	Event
level	Metadata
requestReceivedTimestamp	2022-03-25T19:27:19.718793Z
requestURI	/
responseStatus.code	403
responseStatus.reason	Forbidden
responseStatus.status	Failure
sourceIPs.0	130.211.54.158
stage	ResponseComplete
stageTimestamp	2022-03-25T19:27:19.724919Z
user.groups.0	system:unauthenticated
user.username	system:anonymous
userAgent	python-requests/2.27.1
verb	get

Request information

- Originating source of request
- User or system
- Action/Verb type (GET/POST/DELETE/PATCH)
- Requesting resource and subresource

Response information

- Outcome of the request
- HTTP status code
- Returned object or details associated with resource

Metadata components

- Timestamp
- Audit ID
- Annotations or additional information provided by auditor

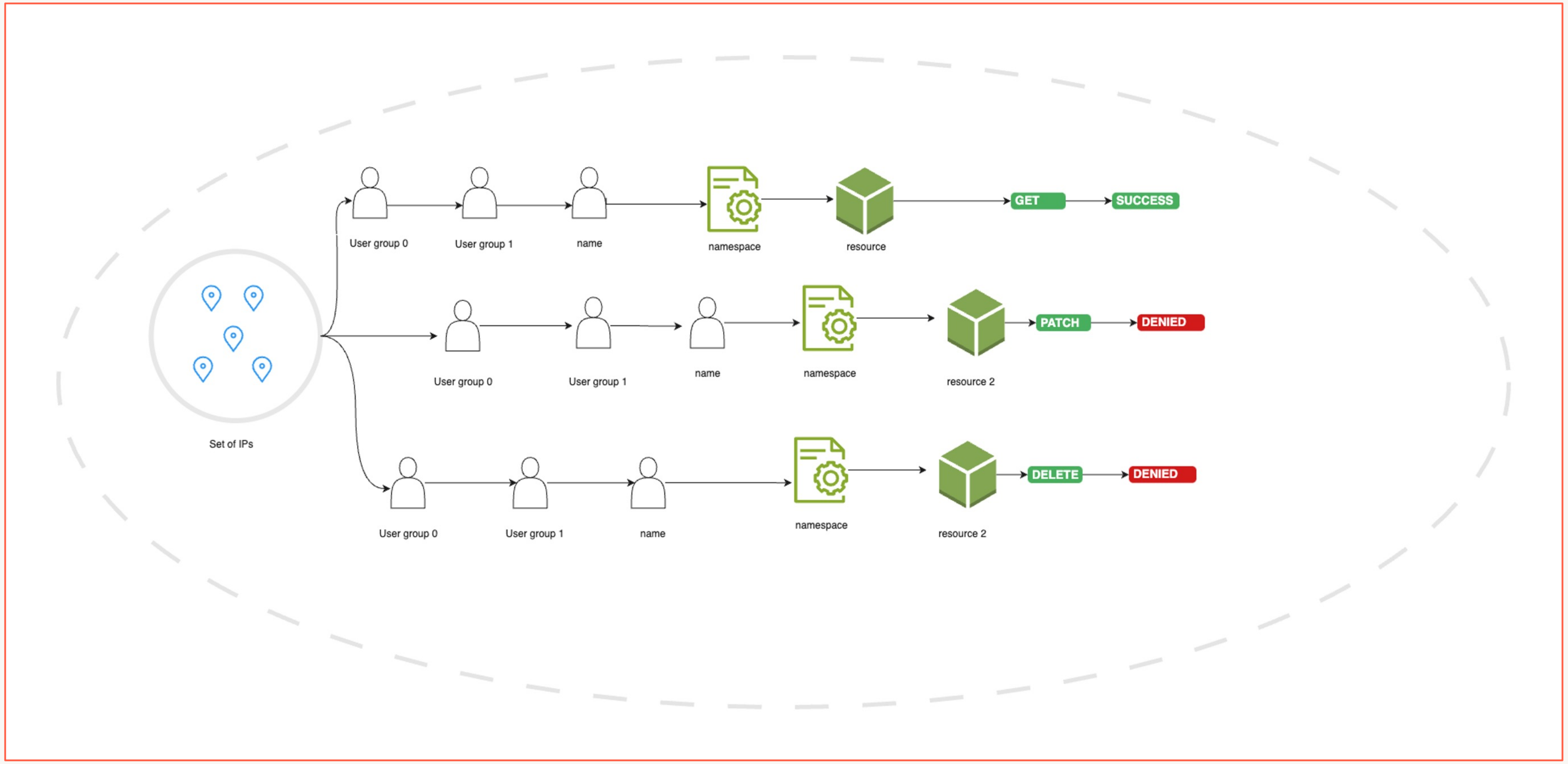
Logging stages

- Request Received
- Response Started
- Response Completed

Source : https://medium.com/pareture/monitor-kubernetes-api-server-audit-in-eks-3e8e6e18e7fb

©2022 Databricks Inc. — All rights reserved

What is a K8s session?



K8s API server audit logs details action on a resource

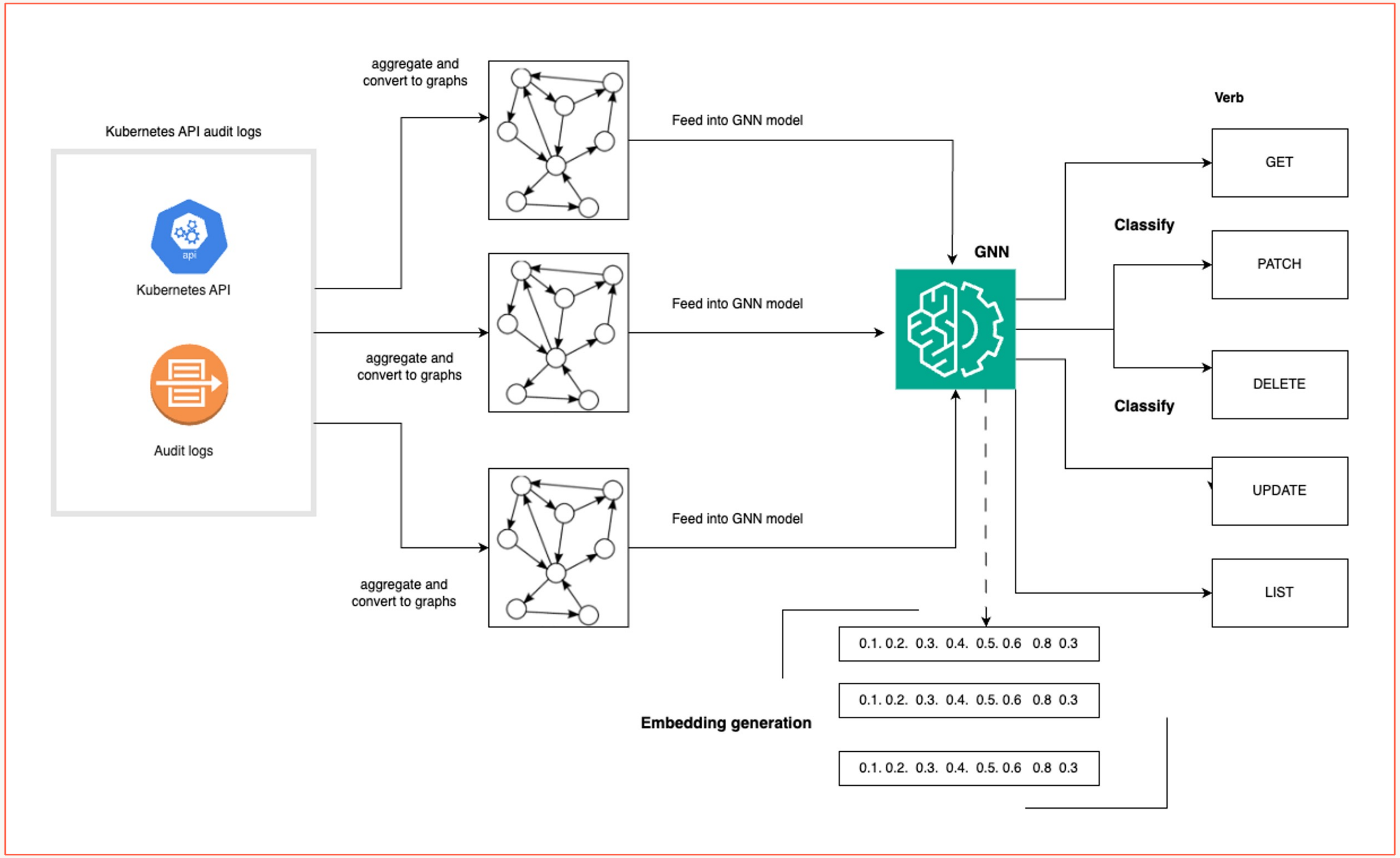
Dependency between each of the components associated with an action/verb (such as **GET/PATCH/DELETE**)

Inter-dependencies between different actions based on IPs, user groups and resources.

Aggregate multiple such records to create a defined K8s session

©2022 Databricks Inc. — All rights reserved

Using GNNs to generate embeddings



- Generate embeddings based on classifying graphs into different verb types
- This is a multi label graph classification problem
- Embeddings generated can be used for multiple downstream tasks associated with anomaly detection.
- Size of embeddings can vary depends on the dataset and some level of experimentation

©2022 Databricks Inc. — All rights reserved

Graph formulation for K8s audit logs

Graph definition

- G = (V,E,X,Y)
- V = Set of nodes in the graph
- X = Node feature matrix
- E = adjacency/connecting matrix
- Y = labels

Set of nodes

- IP
- User group
- Request object
- Resource
- Decision

Feature matrix and labels

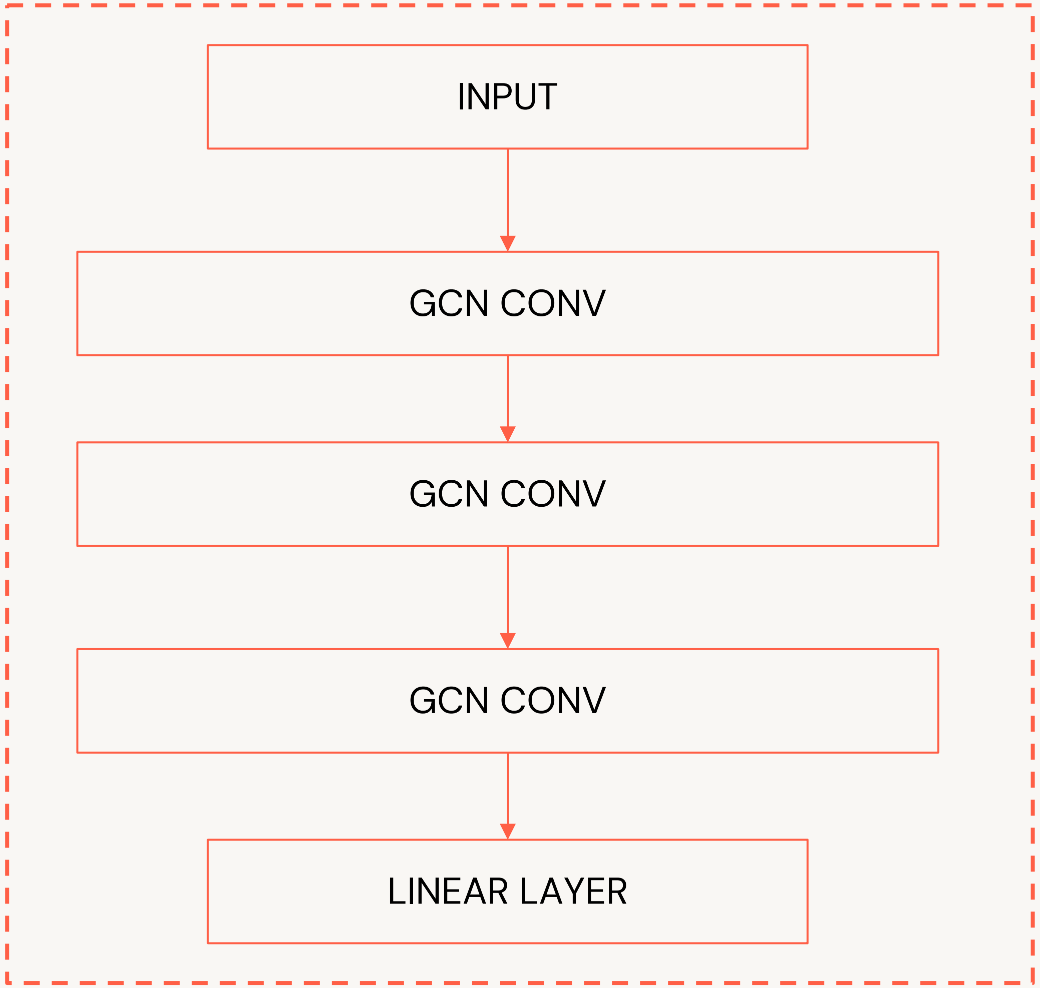
- 40 dim feature matrix
- Feature hashing based on string
- **Y = Type of verb**
- **Classification task**

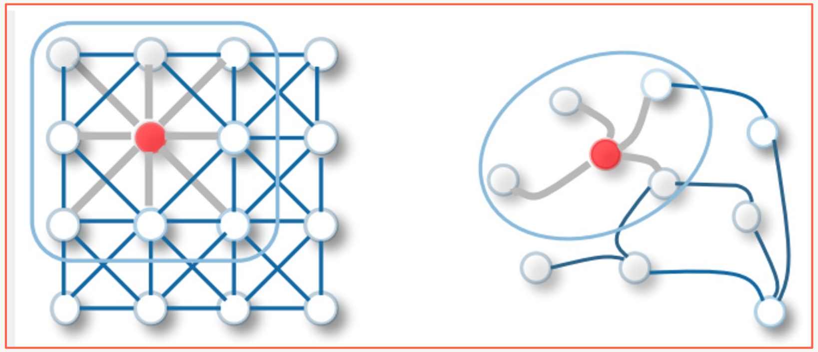
Aggregation

- Aggregating K8 audit logs based on 1 minute time intervals to define a "session"
- Each session to be considered its own graph.
- Each session to be associated with verb

©2022 Databricks Inc. — All rights reserved

Model type and architecture





A GCN is a specific instance of a graph neural network that modifies the idea of a CNN to work with graph data and generate node embeddings

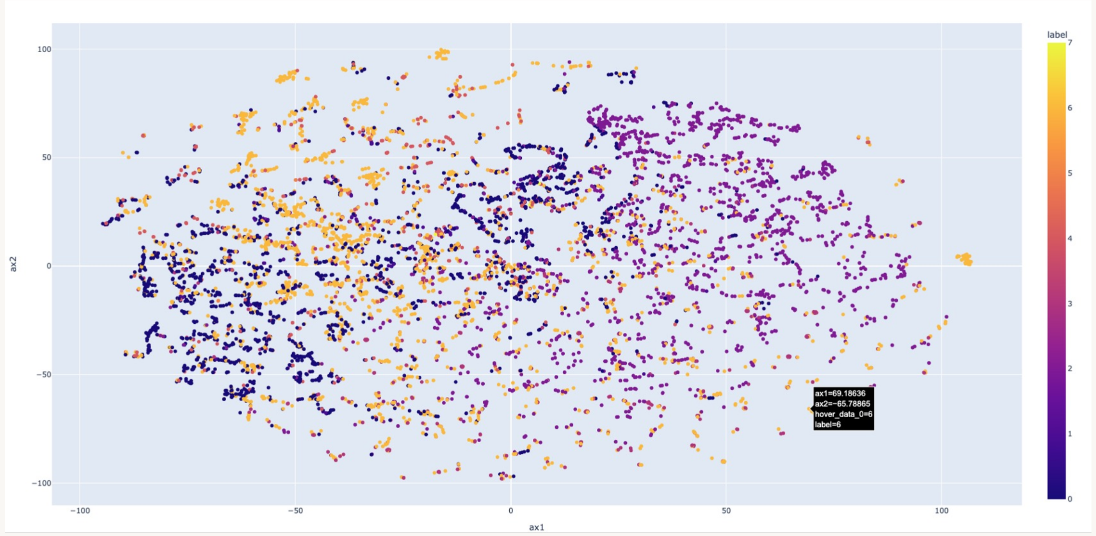
Model specifications

- 3 layer GCN with a linear layer attached at the end.
- node embeddings are passed through a global_mean_pool to generate graph embeddings
- The input is associated with the node features.

source : https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7b

©2022 Databricks Inc. — All rights reserved

TSNE based distribution of embeddings

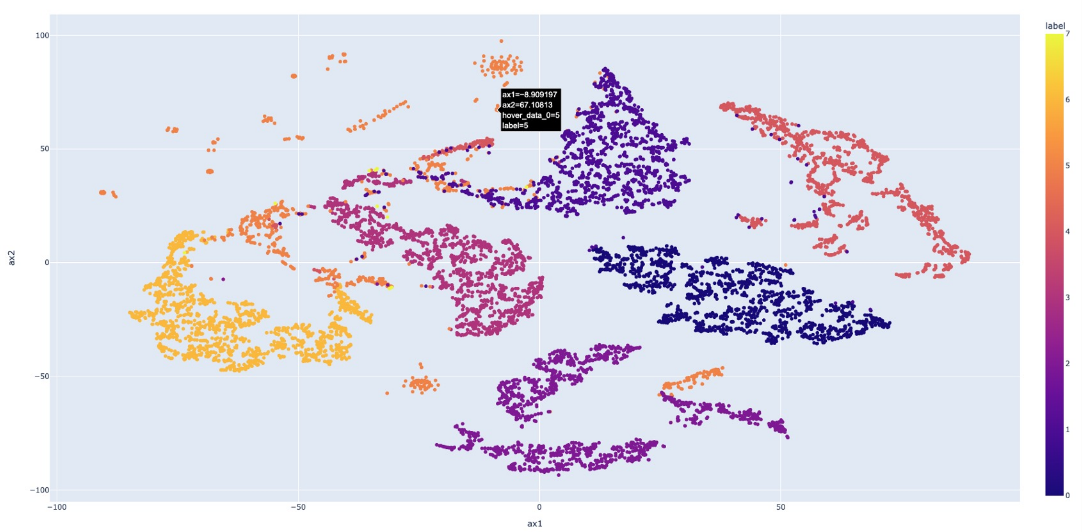


BEFORE training

Training data size : 600K graphs

Time period : 2 months

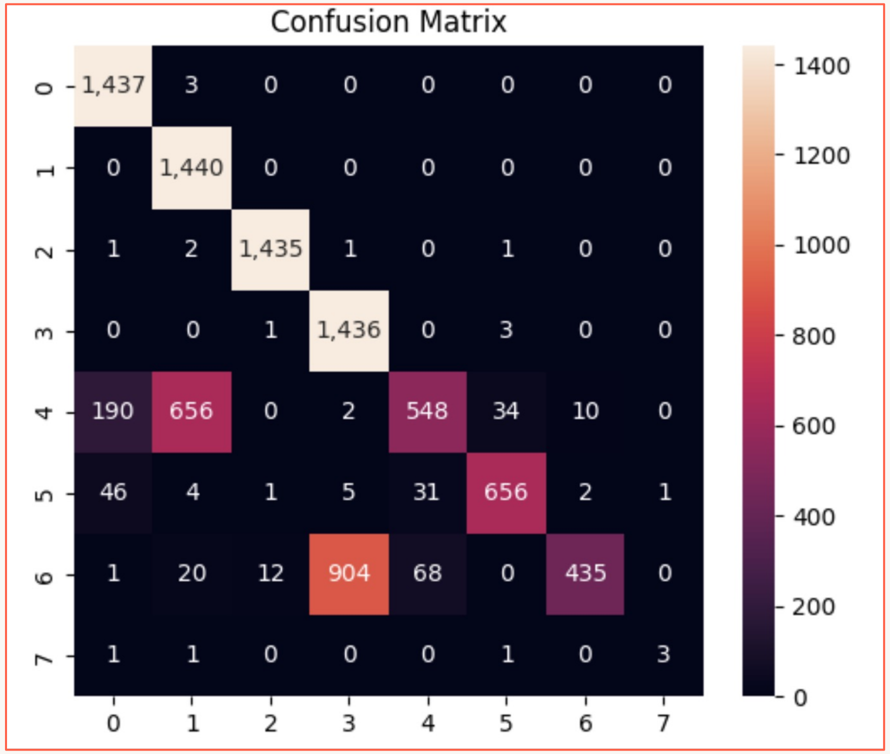
Average number of nodes : 206

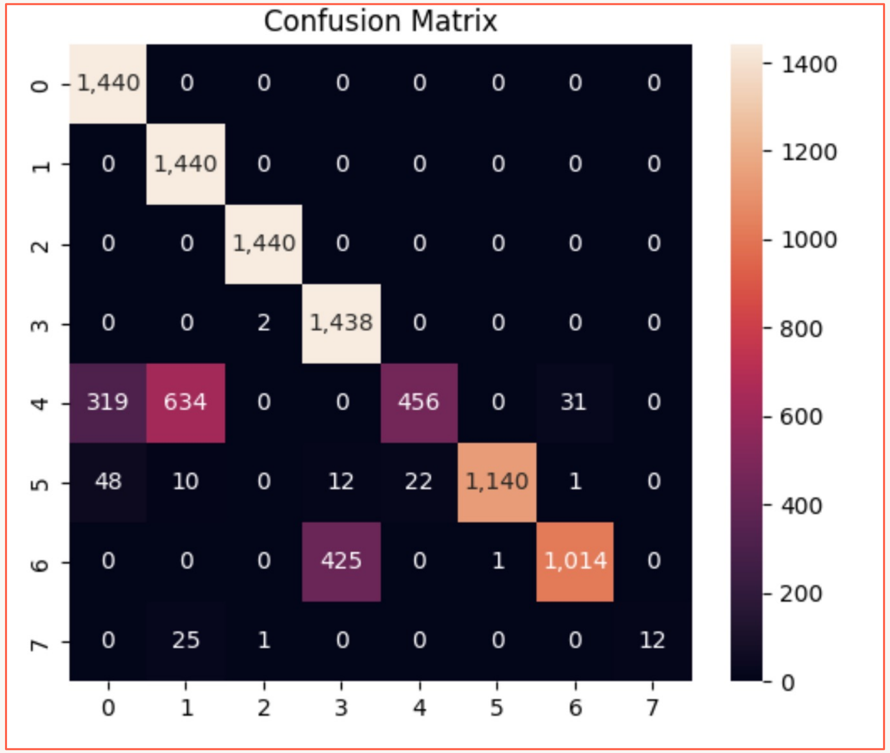


AFTER training

©2022 Databricks Inc. — All rights reserved

Model results





No of evaluation samples : 9392

F1-score : 0.804

No of evaluation samples : 9911

F1-score : 0.840

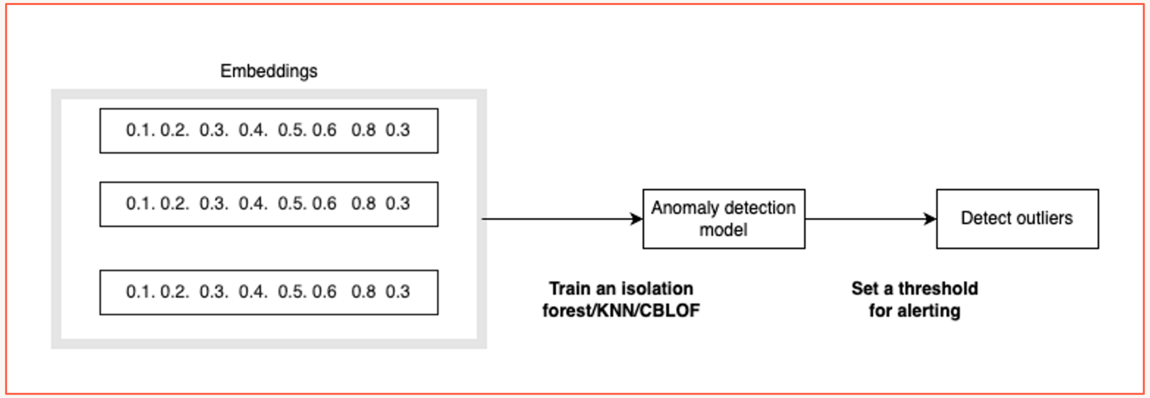
Two different evaluation samples from different regions

©2022 Databricks Inc. — All rights reserved

Threat detection with embeddings

From the POV of an analyst : What can we actually detect?

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account	Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking	Denial of service
Application vulnerability	Application exploit (RCE)		Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files	
Exposed Dashboard	SSH server running inside container					Instance Metadata API	Writable volume mounts on the host	
							Access Kubernetes dashboard	
							Access tiller endpoint	



- Lot of the MITRE use cases listed above can be detected by writing **precise heuristic based detection rules.**
- Some common use cases :
  - Listing of K8s secrets
  - Kubernetes cronjob
  - Kubeconfig file

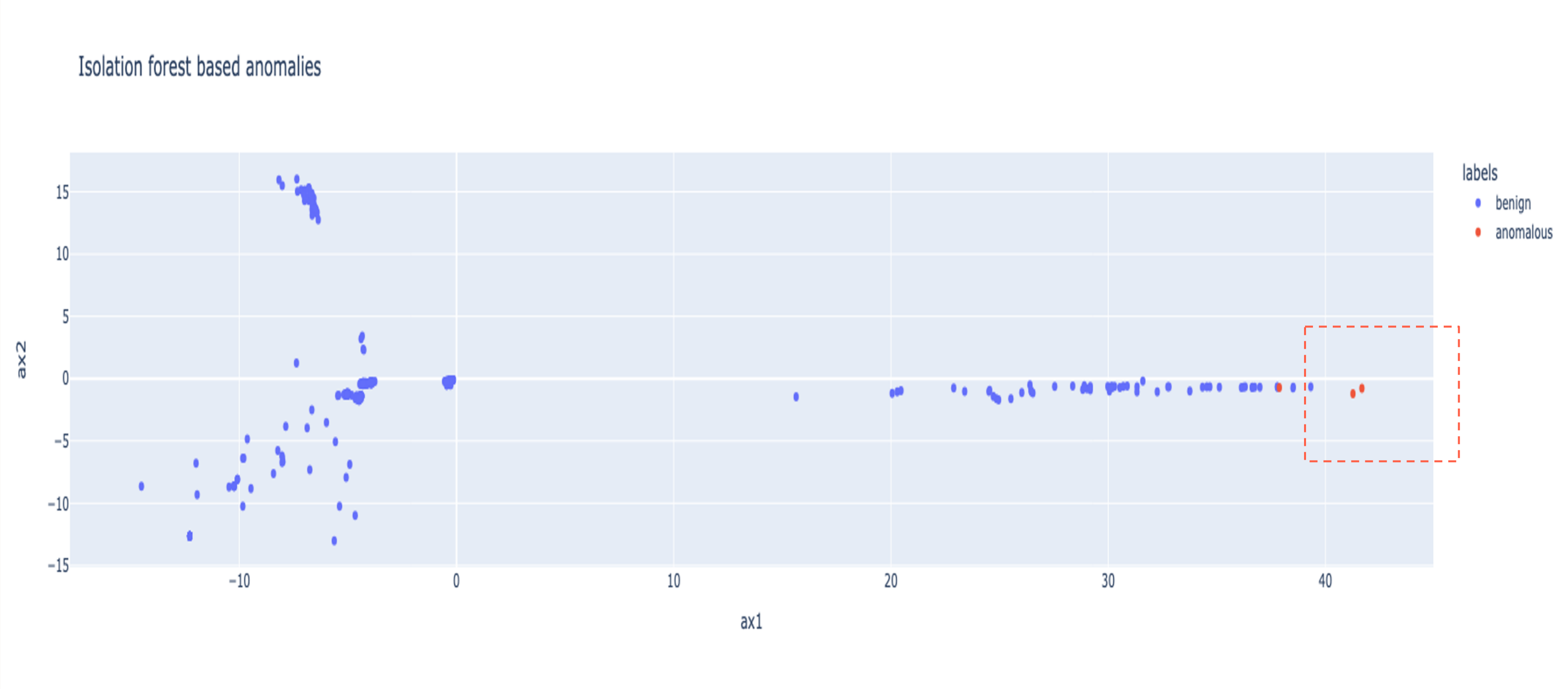
- Embedding based models in this case are “umbrella” models.
- Best way to use these models is by correlating their results with heuristic based detection rules.
- Infrastructure is prone to malicious attacks. But it’s also prone to bad hygiene -> **expect benign true positives**.

source : https://attack.mitre.org/matrices/enterprise/containers/

©2022 Databricks Inc. — All rights reserved

Using embeddings downstream

- Trained an isolation forest on the trained model embeddings with a low contamination factor (~ 0.001 - 0.006)
- Created an evaluation dataset of 420 graphs to see if we can find unusual activity patterns.
- Use trained isolation forest on evaluation set embeddings and extract anomalies



How do we evaluate if anomalies these are legitimate?

- Unusual number of nodes
- Set of source IPs associated with the K8s session
- Namespaces associated with the K8s session
- Usernames associated with the K8s session

3 data points that are considered anomalous in this case

©2022 Databricks Inc. — All rights reserved

After some digging…

Why were the anomalous points anomalous?

Threat hunt	Results
What was the action/verb associated with the anomalous graph?	• The action/verb was “CREATE”
What was the size of the anomalous graphs?	• The size of the anomalous graphs were the lowest in that K8s session for the verb “CREATE”
Were there any new namespaces associated with the anomalous graph?	• There was one new namespace associated with the anomalous graph • The namespace was associated with a new **container hardening process introduced by engineering**
Were there any new usernames associated with the anomalous graph?	• New username associated with container hardening process
Were there any new IPs associated with the anomalous graph?	• New IPs associated with the hardening process

Anomalous process → Benign True positive

©2022 Databricks Inc. — All rights reserved

Challenges and looking ahead

Challenges

- Interpretability
- An anomalous event is not always malicious

Looking ahead

- **Threat hunting will continue to remain an important part of detection**
- **Specific detection rules + Models still the way to go for threat detection**
- **Red teaming to make sure adversary simulation is realistic**
- **Work with your IR/SOC team to tune models at a consistent cadence.**

©2022 Databricks Inc. — All rights reserved

Shoutout to the Databricks Detection and Response teams!

Questions?

©2022 Databricks Inc. — All rights reserved

References/Acknowledgements

- https://attack.mitre.org/matrices/enterprise/containers/
- https://medium.com/stanford-cs224w/task/home
- https://medium.com/pareture/monitor-kubernetes-api-server-audit-in-eks-3e8e6e18e7fb
- https://mlabonne.github.io/blog/posts/2022_02_20_Graph_Convolution_Network.html
- https://docs.lacework.net/onboarding/kubernetes-audit-logs-overview
- Elliptic, https://www.kaggle.com/ellipticco/elliptic-data-set
- https://rish-16.github.io/posts/gnn-math/
- T. Kipf and M. Welling, Semi-Supervised Classification with Graph Convolutional Networks (2017). arXiv preprint arXiv:1609.02907. ICLR 2017
- https://towardsdatascience.com/how-to-do-deep-learning-on-graphs-with-graph-convolutional-networks-7d2250723780
- https://docs.aws.amazon.com/eks/latest/userguide/control-plane-logs.html

©2022 Databricks Inc. — All rights reserved