UNIVERSITY
OF TWENTE.

Detecting Anomalous Misconfigurations in AWS Identity and Access Management Policies

Thijs van Ede, Niek Khasuntsev, Bas Steen & Andrea Continella

Contact: t.s.vanede@utwente.nl

pwc

Misconfigurations

Capital One Attacker Exploited Misconfigured AWS Databases

After bragging in underground forums, the woman who stole 100 million credit applications from Capital One has been found guilty.

Misconfigurations

Capital One Attacker Exploited
Miscon

After bragging i
Capital One has

Three million senior citizens' info exposed by SeniorAdvisor

A security breach at SeniorAdvisor, a review website, compromised over three million elderly adults' personal information in August. WizCase researchers observed that a misconfigured Amazon S3 bucket exposed details including individuals' names, numbers, and email addresses. The information pertained to

Misconfigurations

Capital One Attacker Exploited
Miscon

After bragging i
Capital One has

Three million senior citizens' info exposed by SeniorAdvisor

A security breach at SeniorAdvisor, a review website, compromised over three
se researchers
ails including
nation pertained to

Misconfigurations

Capital One Att[...]
Miscon[...]

After bragging i[...]
Capital One has[...]

Three
by Se[...]

May 2022: 23 Million Files Exposed in Pegasus Airlines Breach

In May 2022, a security firm discovered an unprotected AWS S3 bucket containing 6.5 terabytes of "Electronic Flight Bag" information, including navigation information, proprietary software, and personal information pertaining to Pegasus Airlines crew members. Once notified of the exposed information, Pegasus Airlines promptly secured the unprotected S3 bucket.

A security b[...]

[...]se researchers
[...]ails including
[...]ation pertained to

Misconfigurations

Capital One Atta
Miscon

After bragging i
Capital One has

Three
by Se

May 2022: 23 Million Files Exposed in Pegasus Airlines Breach

In May 2022, a security firm discovered an unprotected AWS S3 bucket containing 6.5 terabytes of "Electronic Flight Bag" information, including navigation information, proprietary software, and personal information pertaining to Pegasus Airlines crew members. Once notified of the exposed information, Pegasus Airlines promptly secured the unprotected S3 bucket.

A security b

July 2021: PeopleGIS Exposes Sensitive Data for Over 80 Municipalities

In July 2021, a group of ethical hackers at WizCase discovered a vulnerability affecting at least 80 municipalities in the United States. This breach resulted from misconfigured Amazon S3 buckets related to MapsOnline, a service run by the software company PeopleGIS. It's unclear whether the misconfiguration was made by PeopleGIS or by the municipalities in question.

Misconfigurations

May 2022: 23 Million Files Exposed in Pegasus

Capital One Att...

Miscon...

After bragging i...

Capital One has

Three

by Se

A security

...et containing 6.5 terabytes of

... proprietary software, and

...nce notified of the exposed

...ucket.

Jul

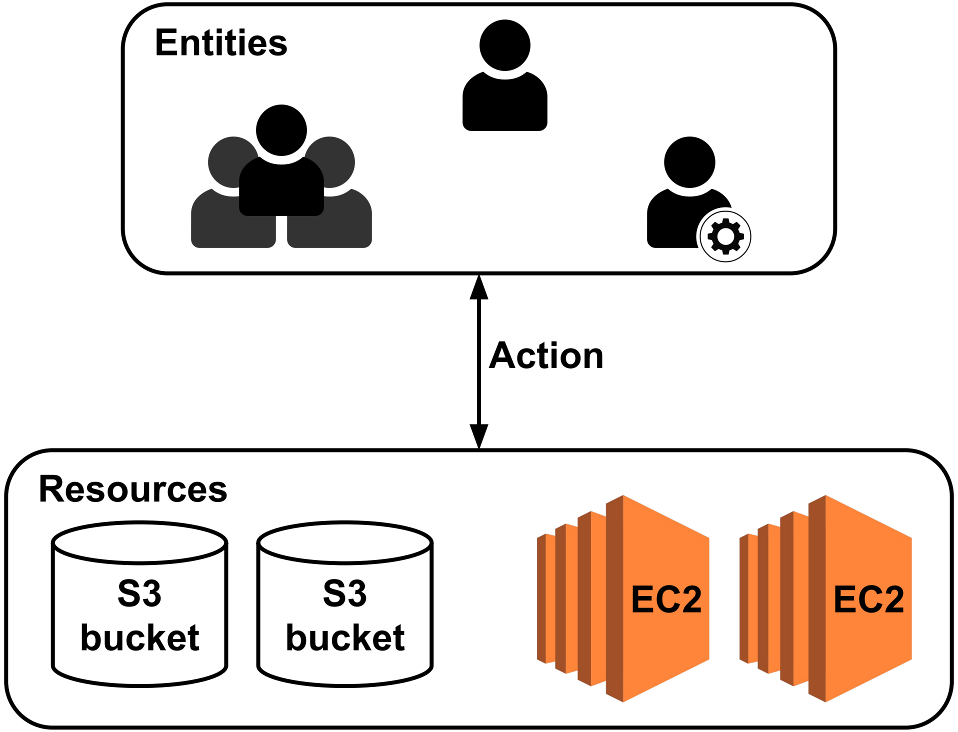
Ov

...ata for

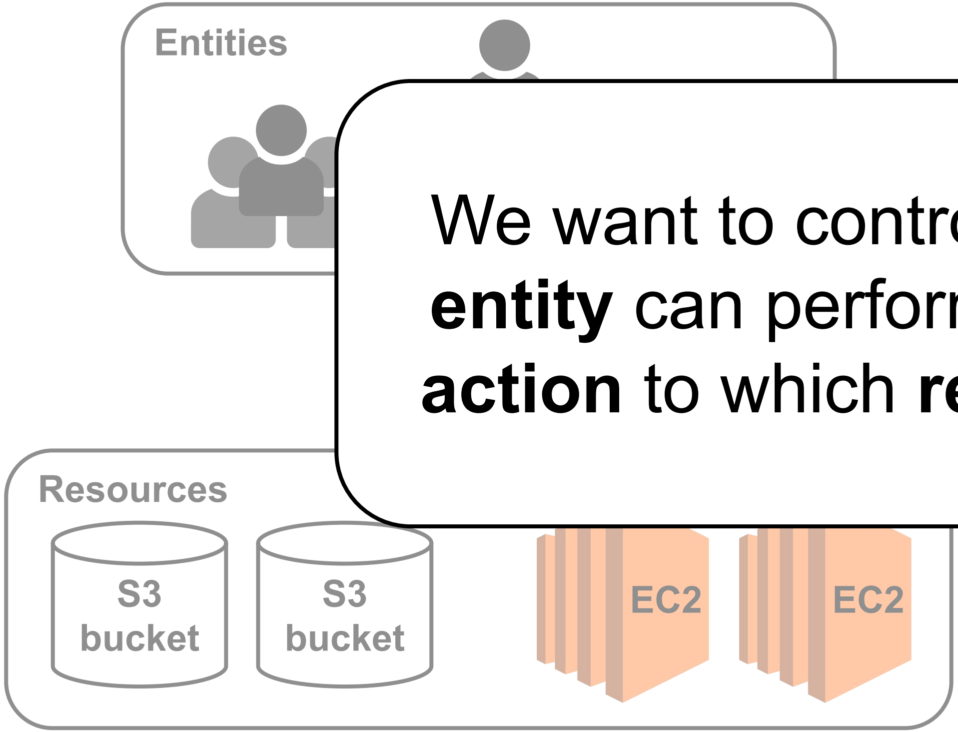
Misconfigured cloud environments are a problem!

In July 2021, a group of ethical hackers at WizCase discovered a vulnerability affecting at least 80 municipalities in the United States. This breach resulted from misconfigured Amazon S3 buckets related to MapsOnline, a service run by the software company PeopleGIS. It's unclear whether the misconfiguration was made by PeopleGIS or by the municipalities in question.

Identity and Access Management

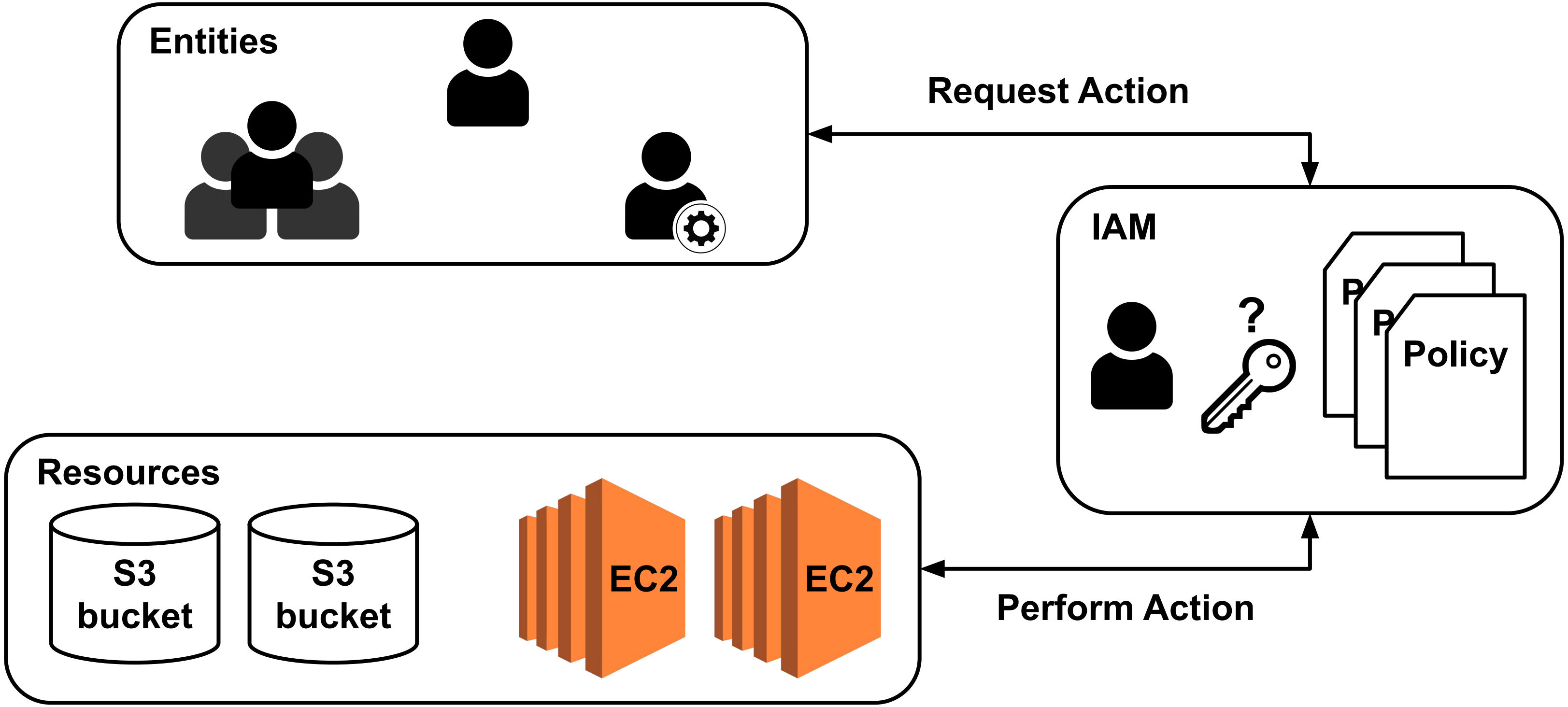


Identity and Access Management

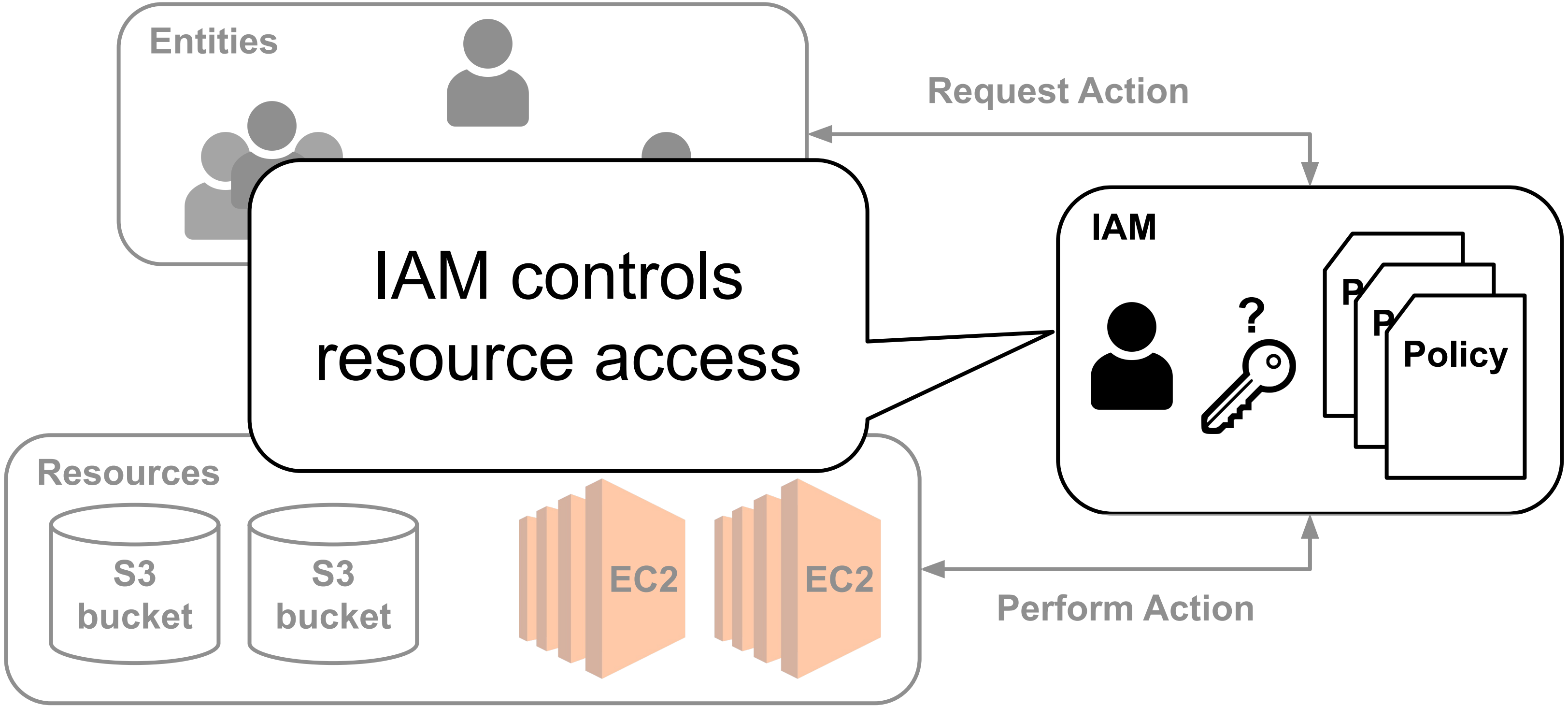


We want to control which **entity** can perform which **action** to which **resource**

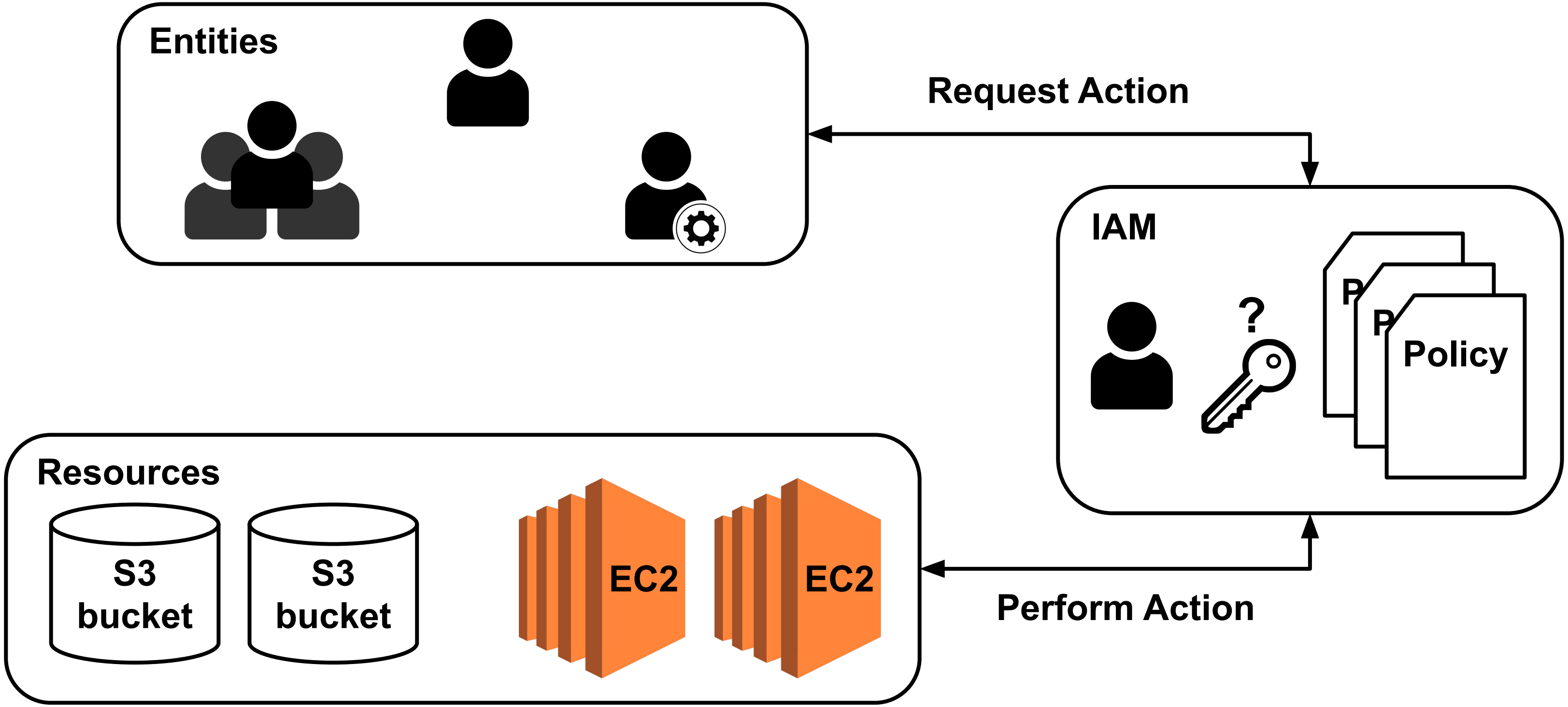
Identity and Access Management



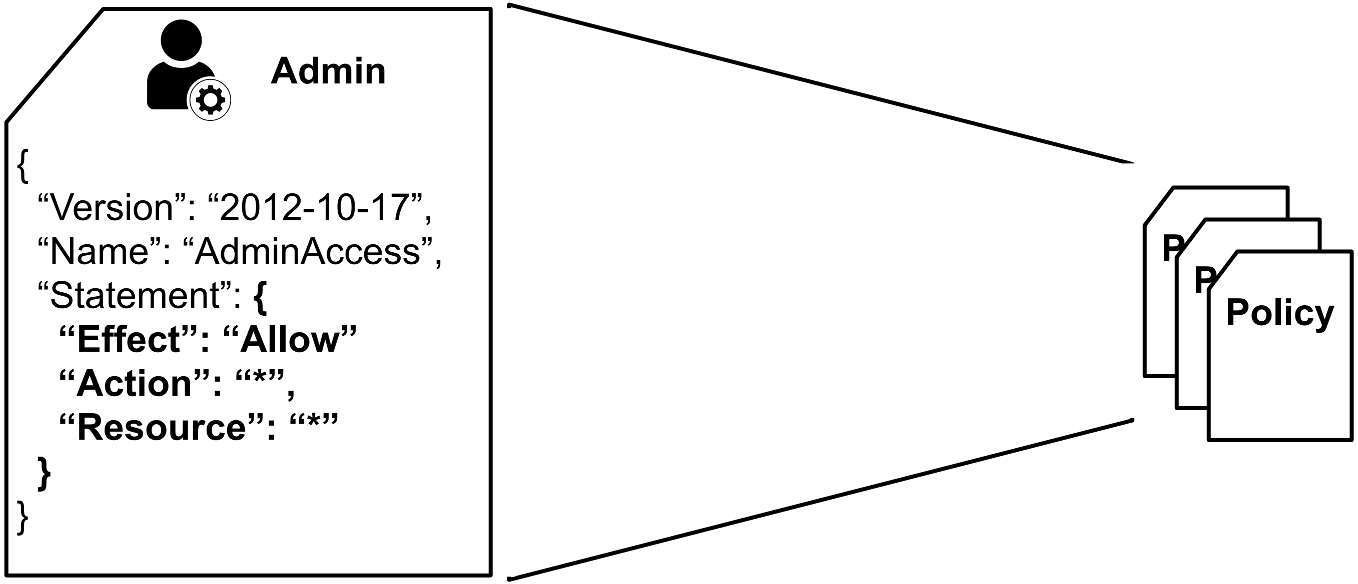
Identity and Access Management



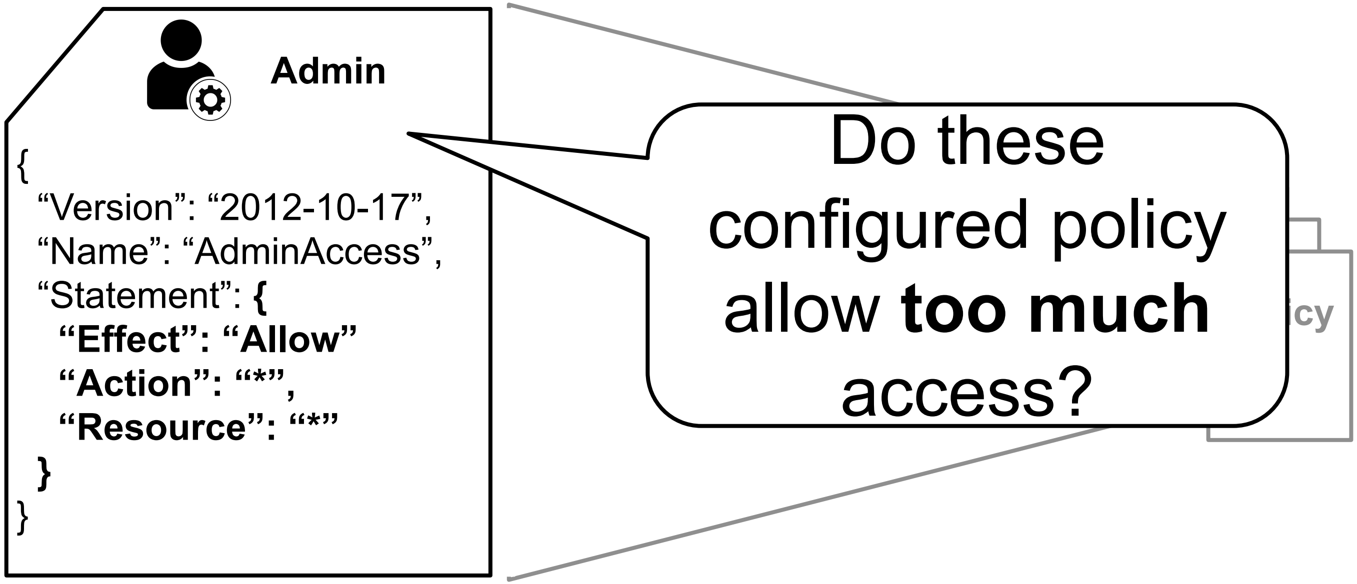
Identity and Access Management



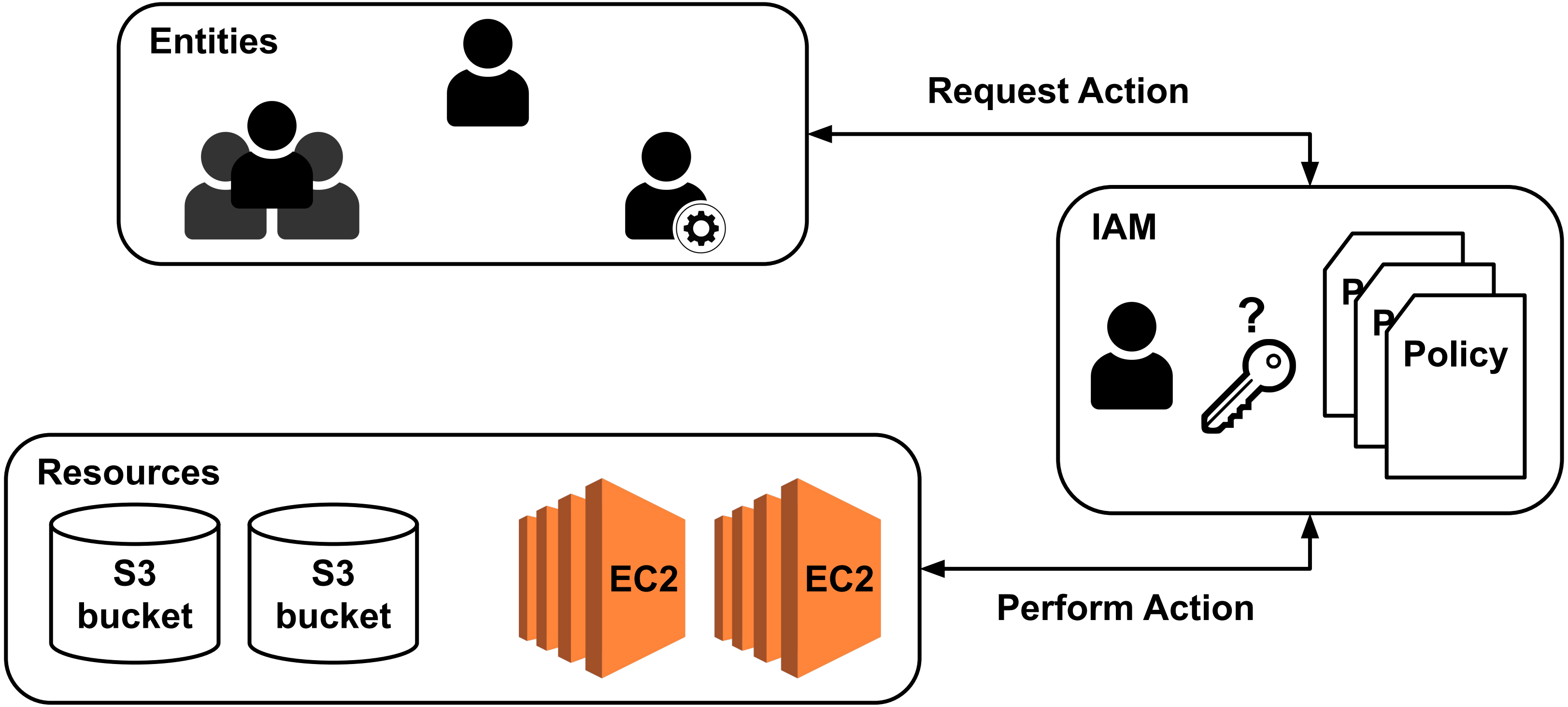
Identity and Access Management



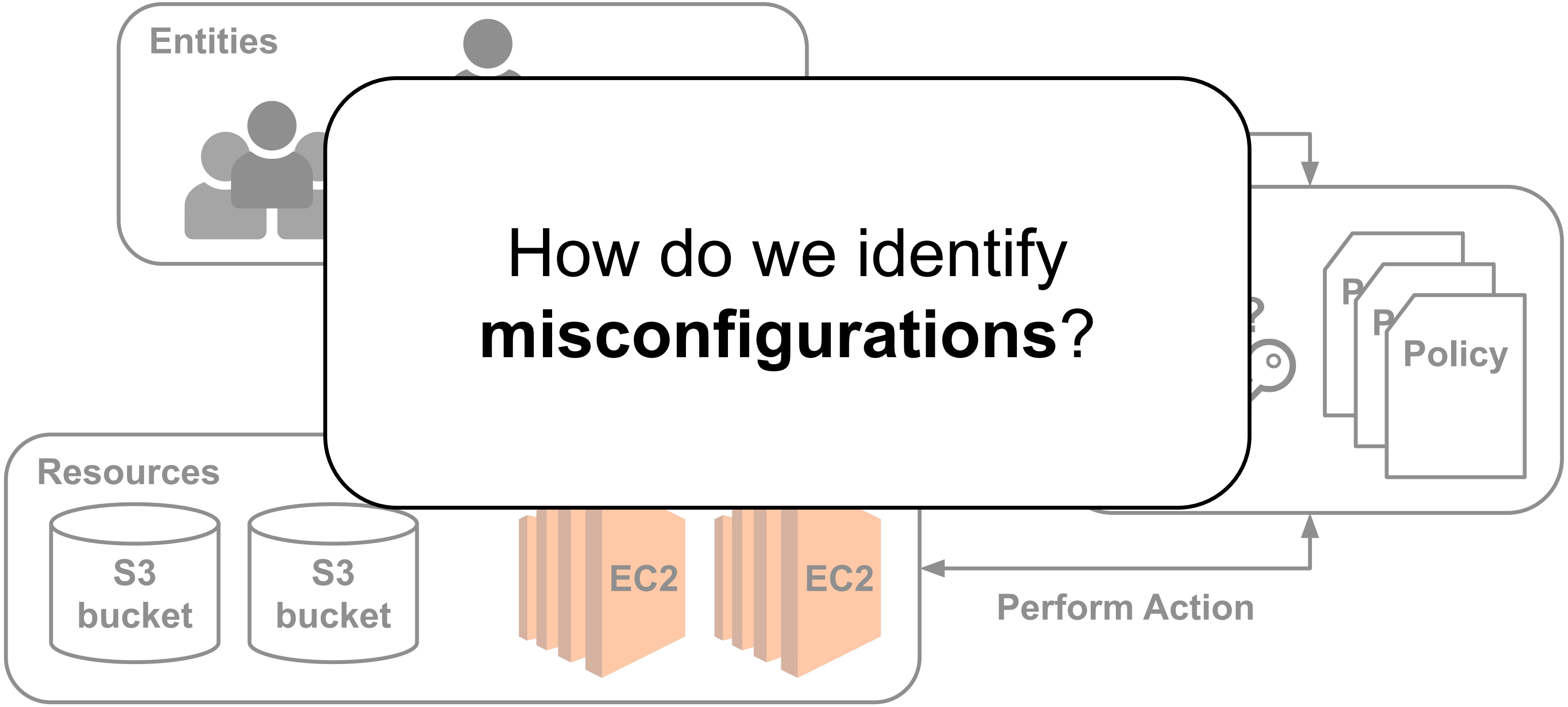
Identity and Access Management



Identity and Access Management



Identity and Access Management



How do we identify **misconfigurations**?

Existing solutions

- Cloud Custodian

Existing solutions

- Cloud Custodian
  - Rule-based

Existing solutions

- Cloud Custodian
  - Rule-based
  - Requires manual tweaking of rules

Existing solutions

- Cloud Custodian
  - Rule-based
  - Requires manual tweaking of rules
- P-Diff

Existing solutions

- Cloud Custodian
  - Rule-based
  - Requires manual tweaking of rules
- P-Diff
  - Learns control policies from access logs

Existing solutions

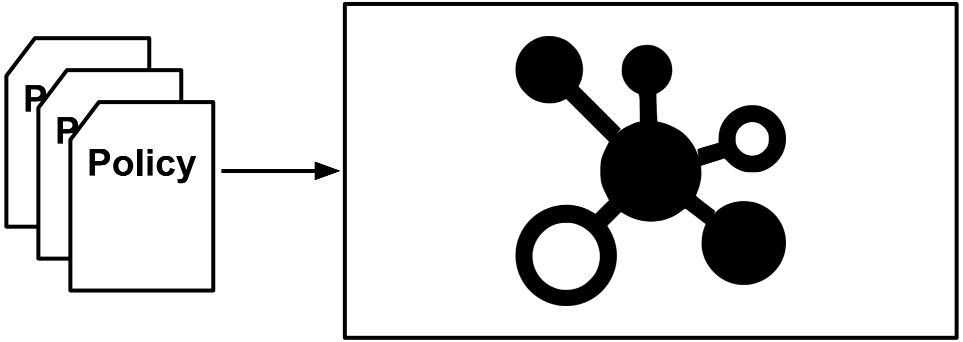
- Cloud Custodian
  - Rule-based
  - Requires manual tweaking of rules
- P-Diff
  - Learns control policies from access logs
  - Reactive approach

Idea

- Most policies are properly configured

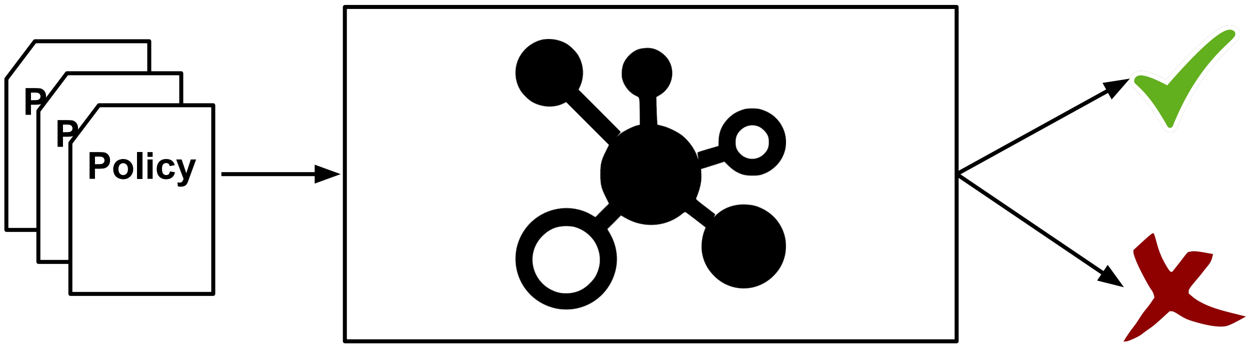
Idea

- Most policies are properly configured
- Use **anomaly detection** to learn properly configured policies



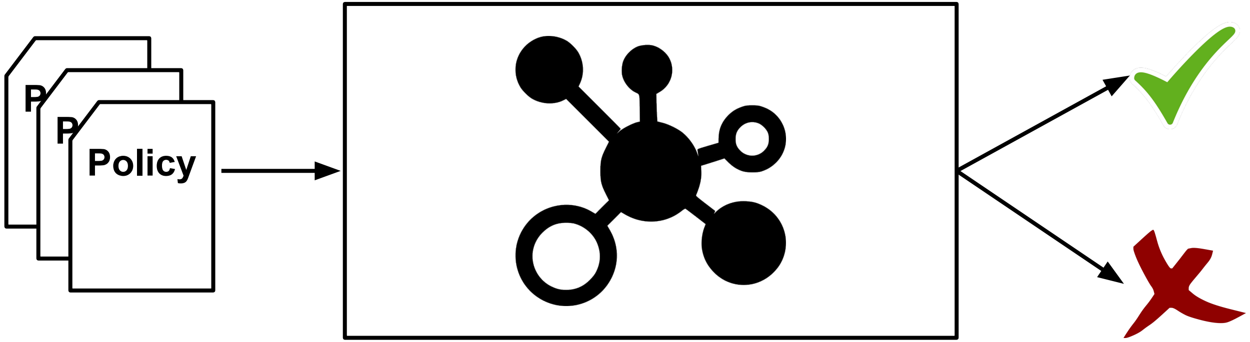
Idea

- Most policies are properly configured
- Use **anomaly detection** to learn properly configured policies
- Any found **anomalies** will likely be **misconfigurations**



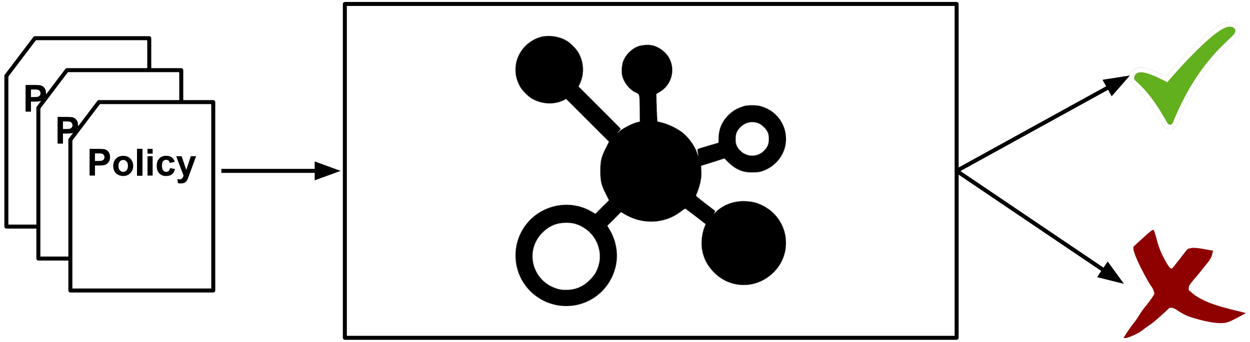
Challenges

- Policies are **specific** to the **context** of the organization



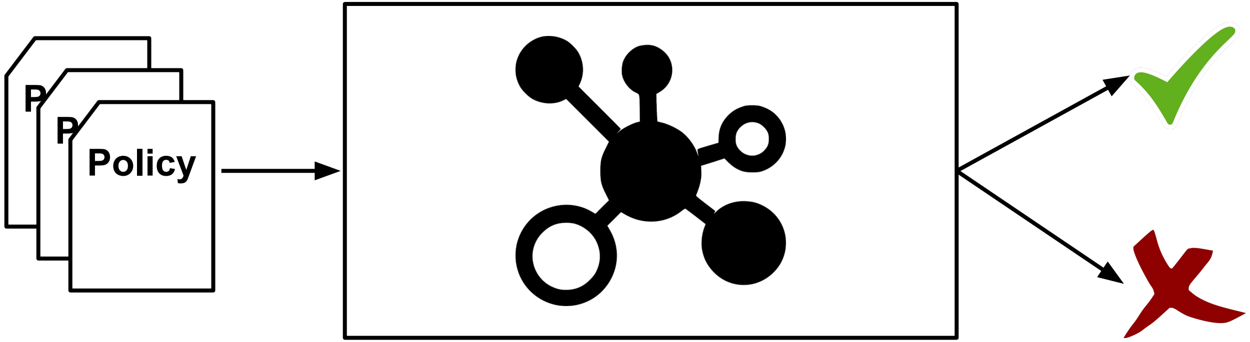
Challenges

- Policies are **specific** to the **context** of the organization
- Checks must be **proactive** to ensure policies are not abused



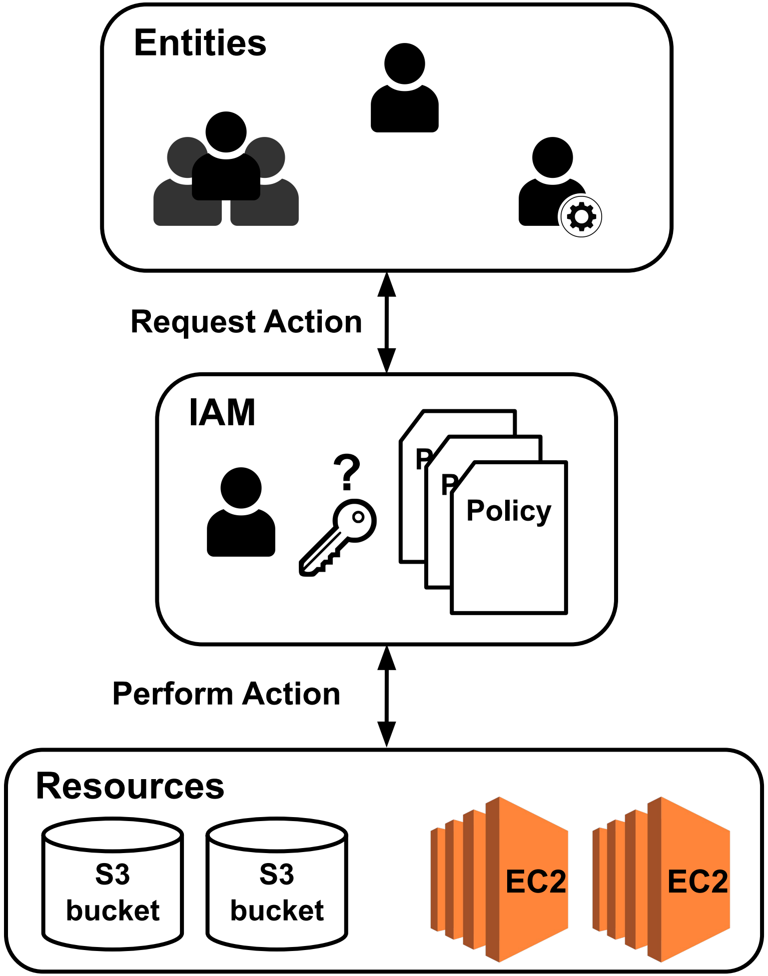
Challenges

- Policies are **specific** to the **context** of the organization
- Checks must be **proactive** to ensure policies are not abused
- Checks must be **low maintenance** to ensure adoption



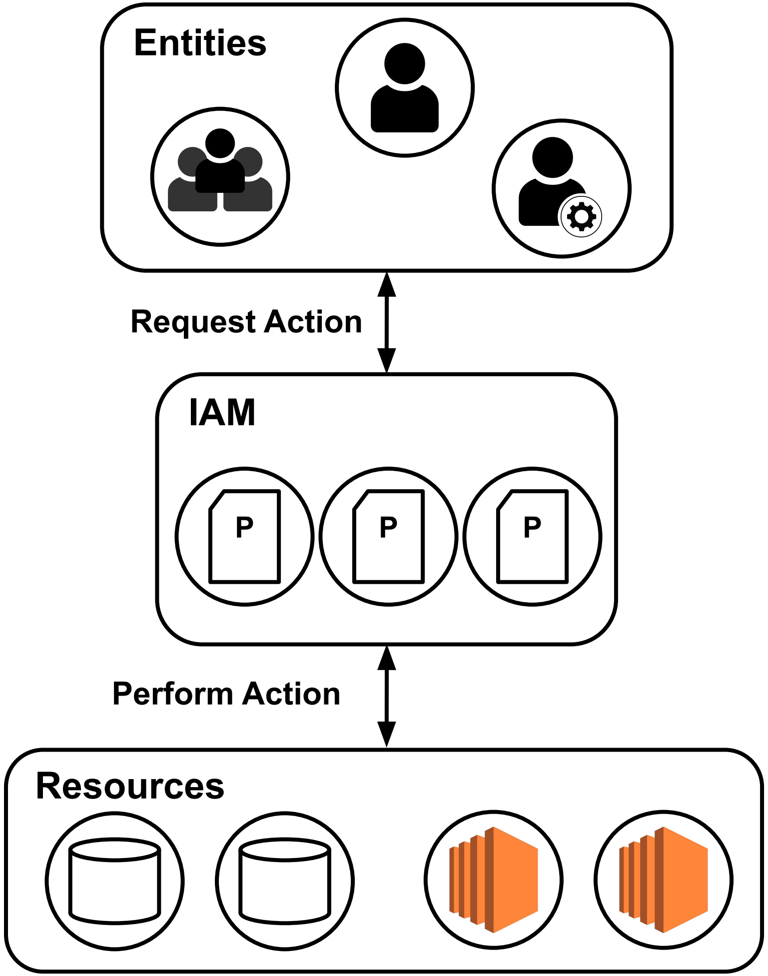
Approach

- Model policies as a graph



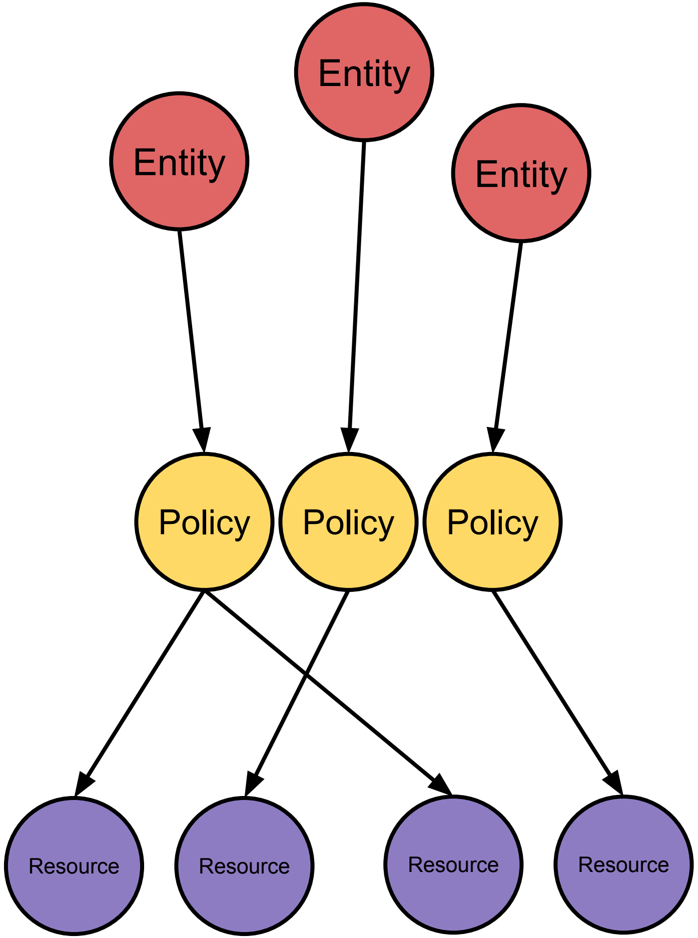
Approach

- Model policies as a graph



Approach

- Model policies as a graph

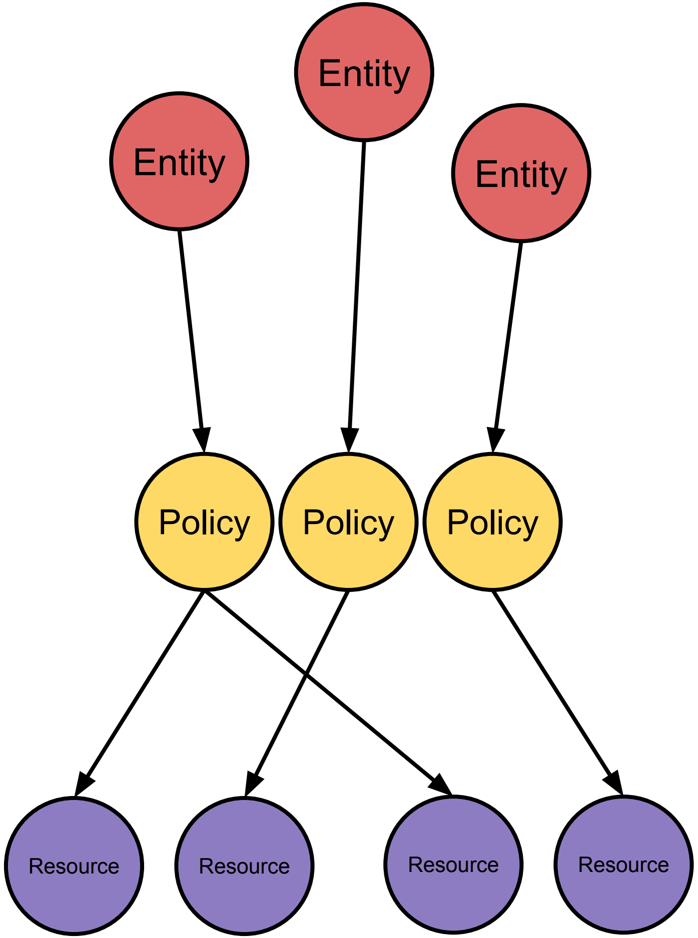


Approach

- Model policies as a graph
  - A policy can have multiple statements

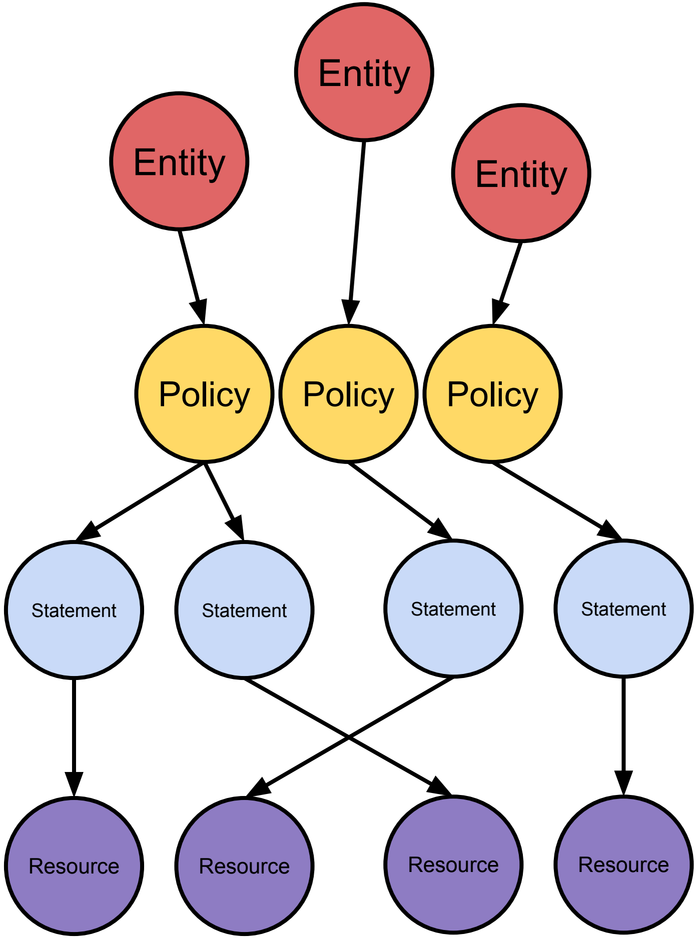
Admin

```
{
  "Version": "2012-10-17",
  "Name": "AdminAccess",
  "Statement": {
    "Effect": "Allow"
    "Action": "*",
    "Resource": "*"
  }
}
```



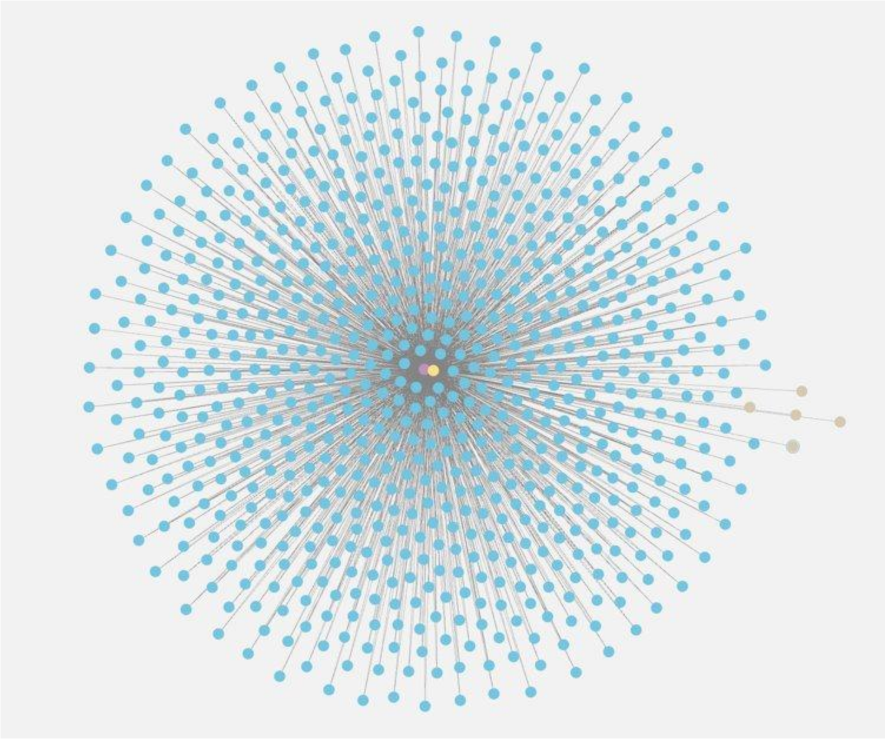
Approach

- Model policies as a graph
  - A policy can have multiple statements



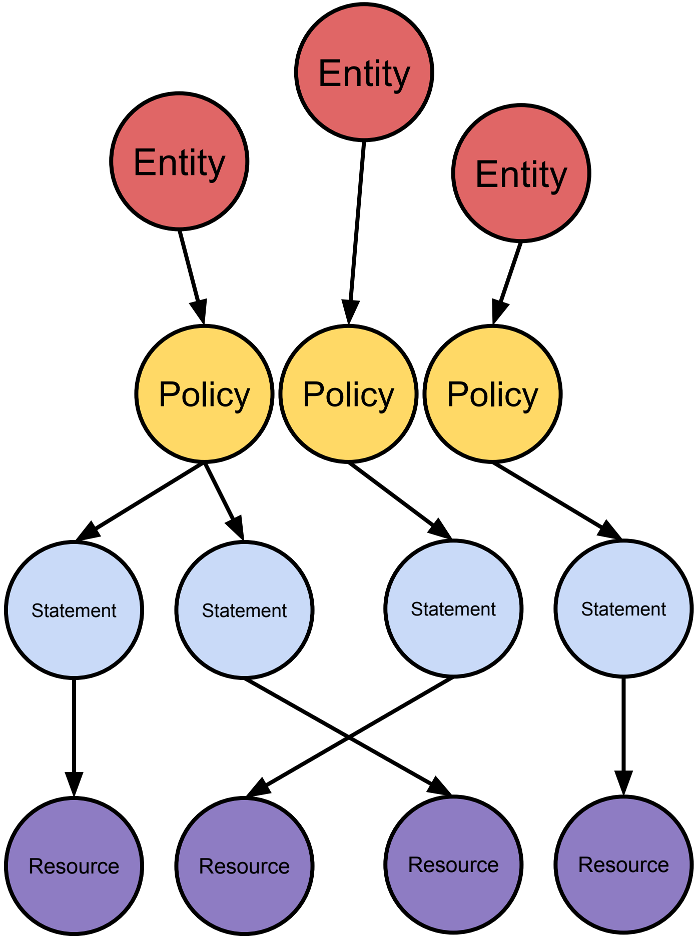
Approach

- Model policies as a graph
  - A policy can have multiple statements



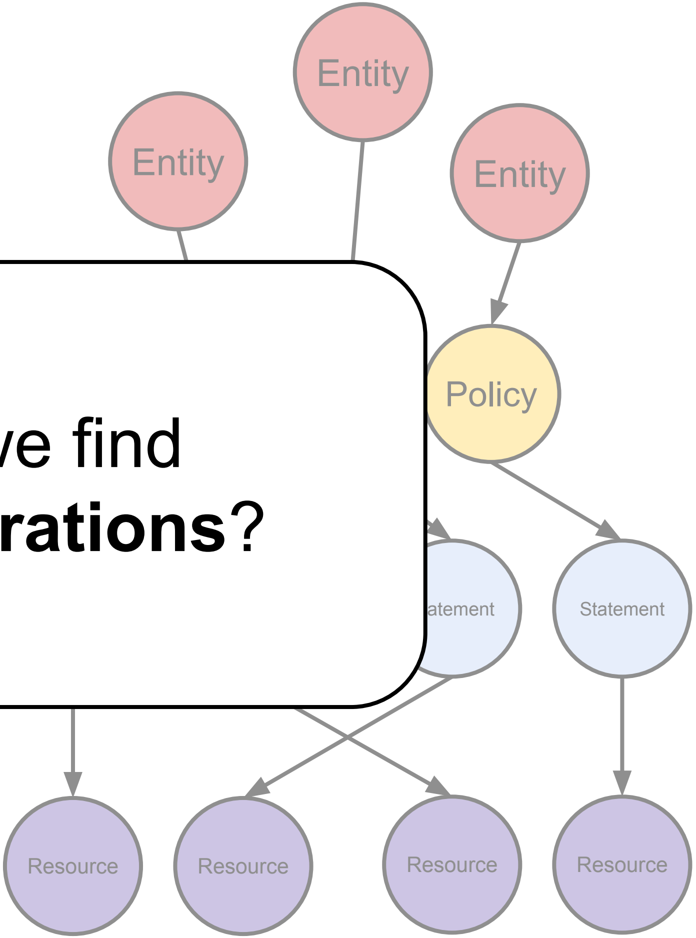
Approach

- Model policies as a graph
  - A policy can have multiple statements



Approach

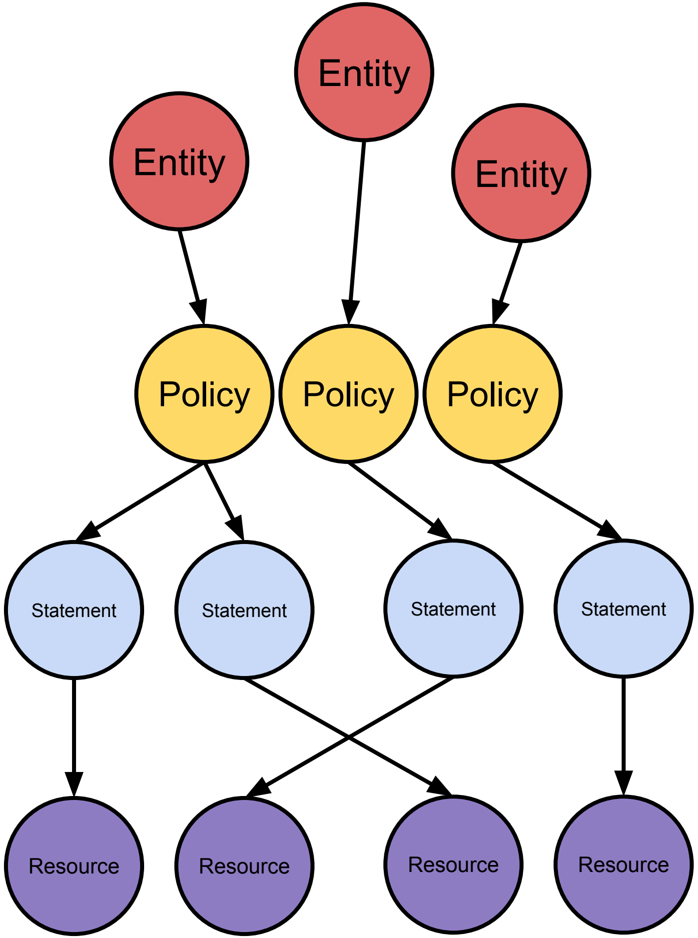
- Model policies as a graph
  - A policy c
    statemen



How do we find **misconfigurations**?

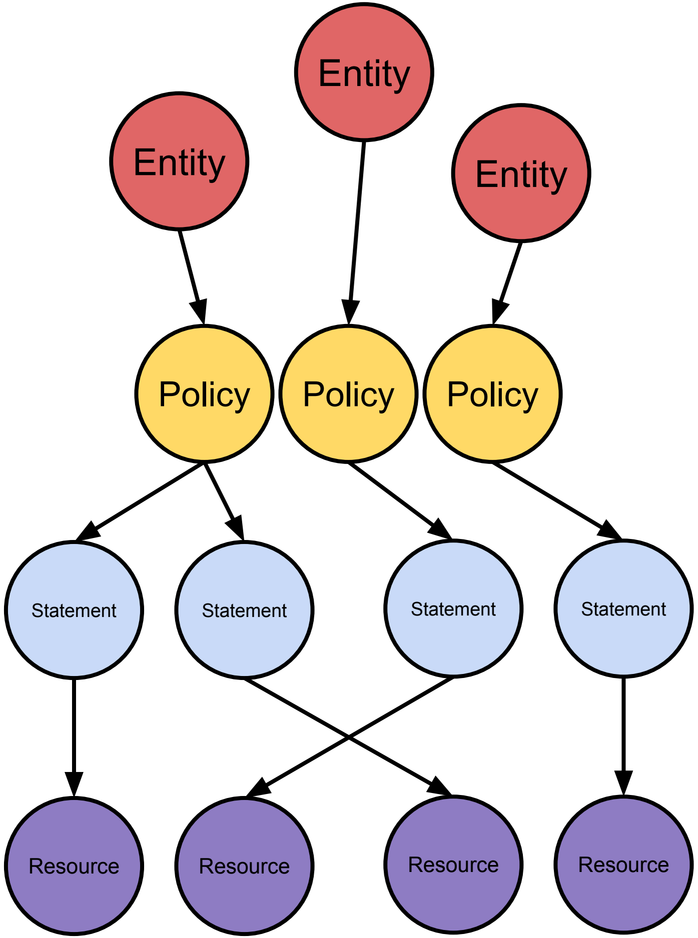
Approach

- Model policies as a graph
  - A policy can have multiple statements
- Policies are **specific** to the **context** of the organization



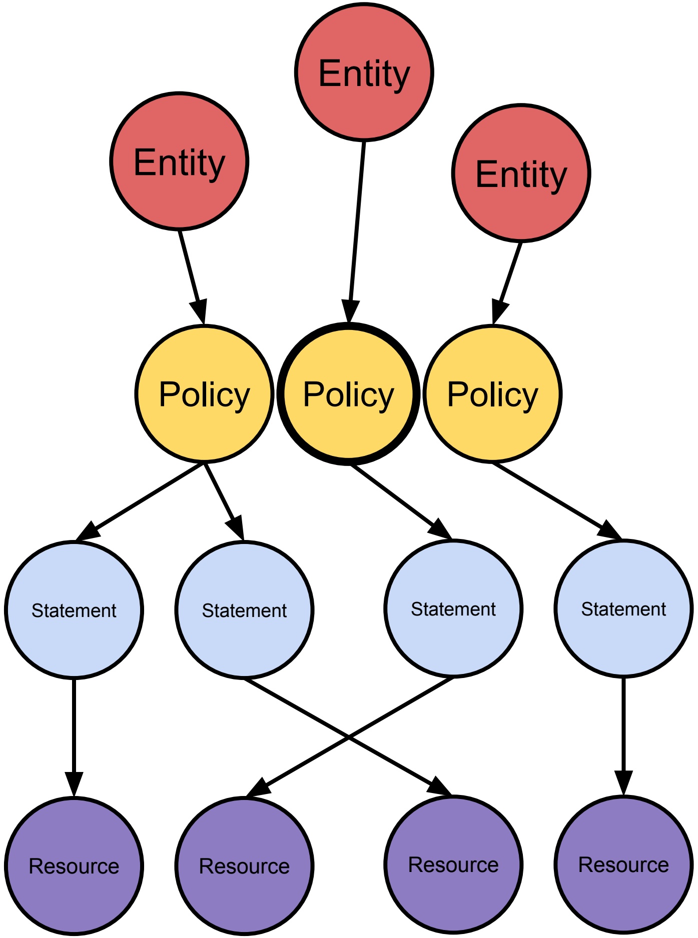
Approach

- Model policies as a graph
  - A policy can have multiple statements
- Policies are **specific** to the **context** of the organization
- Model the context of policies using **Node2vec**



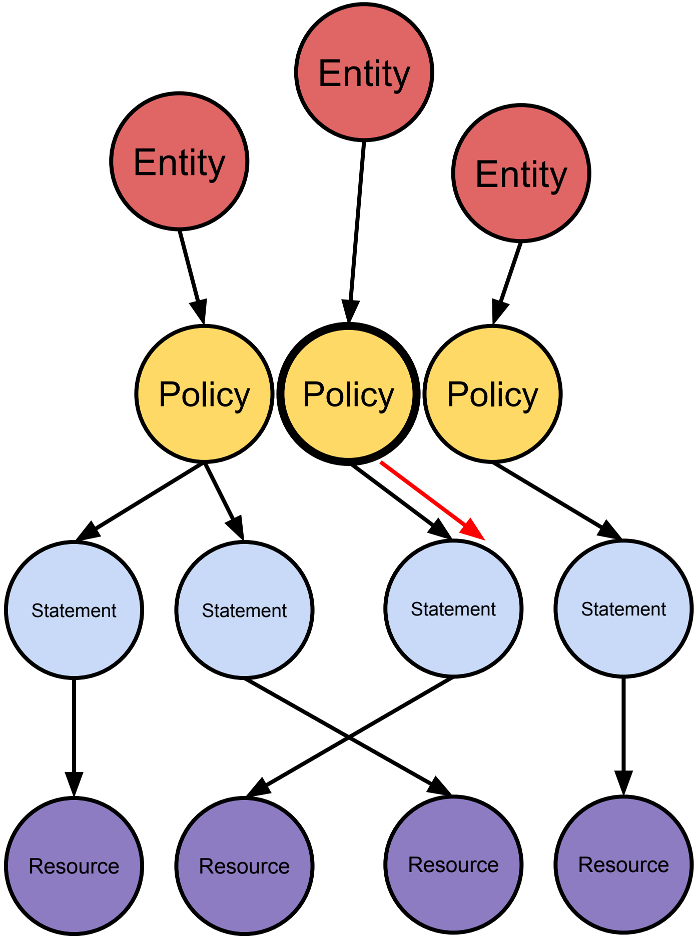
Approach - Node2vec

- Select a starting **policy node**



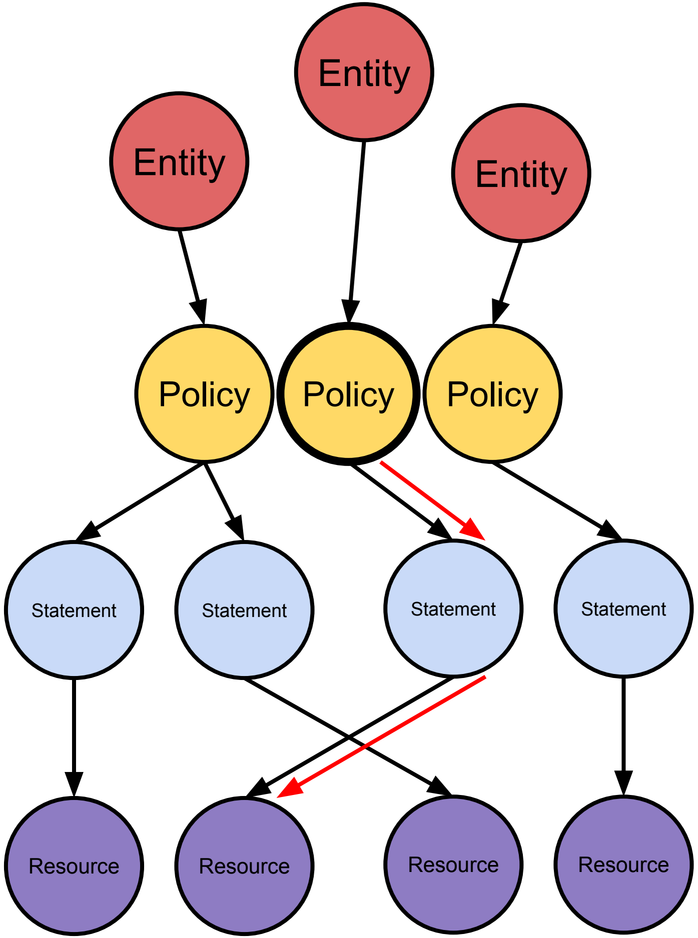
Approach - Node2vec

- Select a starting **policy node**
- Perform random walks



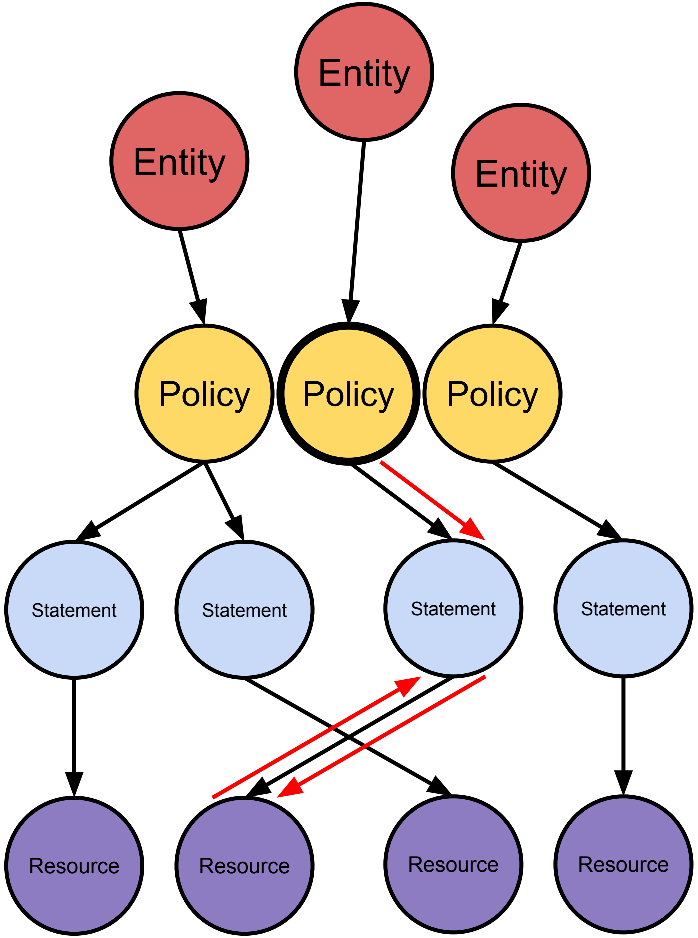
Approach - Node2vec

- Select a starting **policy node**
- Perform random walks
  - Collect information about visited nodes and edges



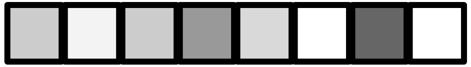
Approach - Node2vec

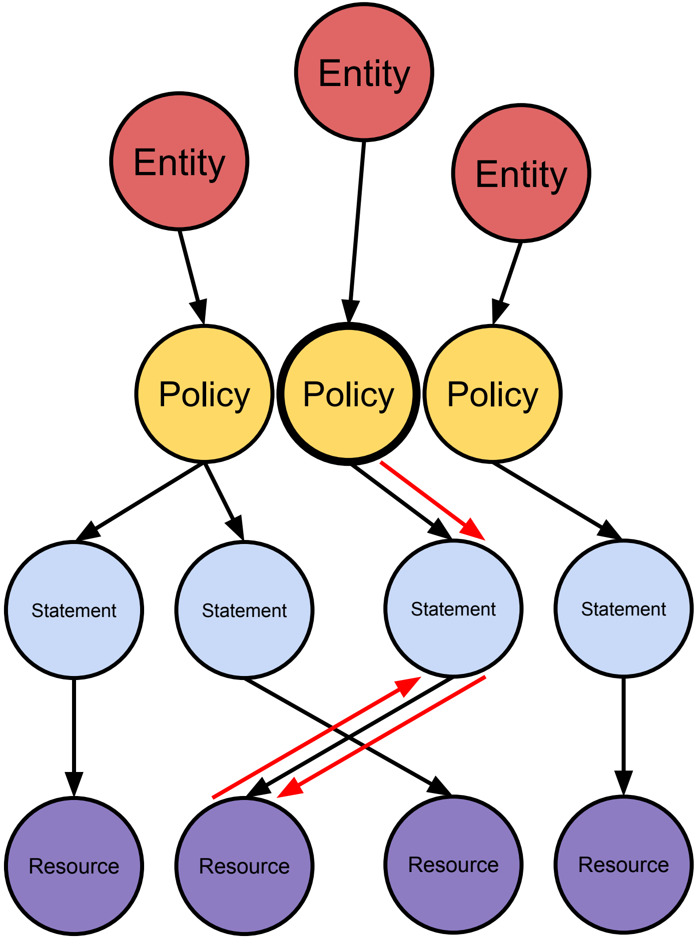
- Select a starting **policy node**
- Perform random walks
    - Collect information about visited nodes and edges



Approach - Node2vec

- Select a starting **policy node**
- Perform random walks
    - Collect information about visited nodes and edges
- Store information in a fixed length vector

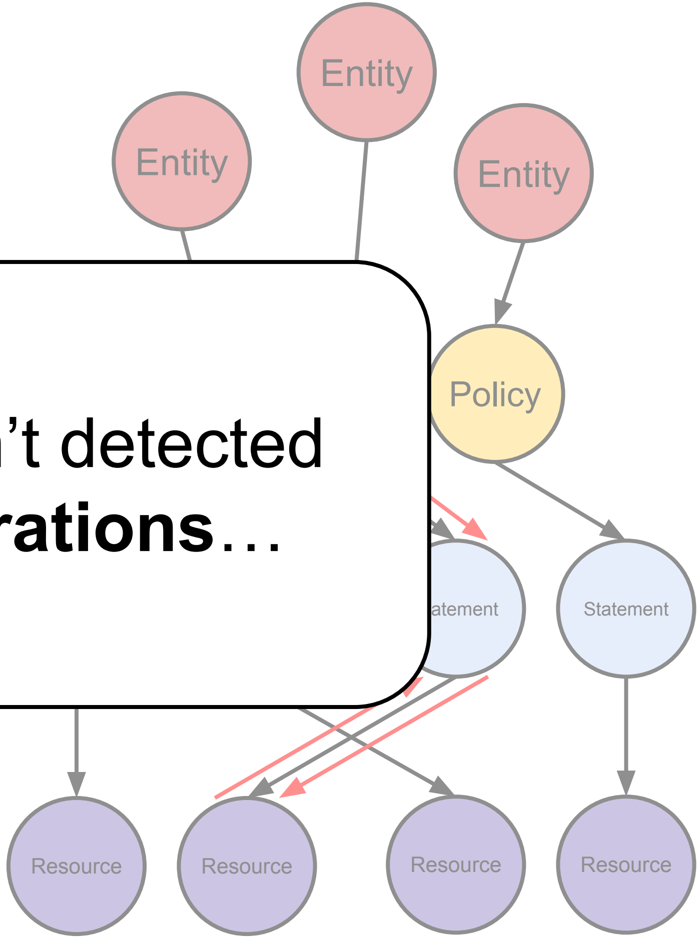




Approach - Node2vec

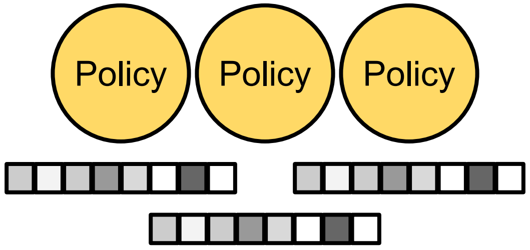
- Select a starting **policy node**
- Perform rand
  - Collect i
    visited n
- Store inform
  length vecto

We still haven't detected misconfigurations…



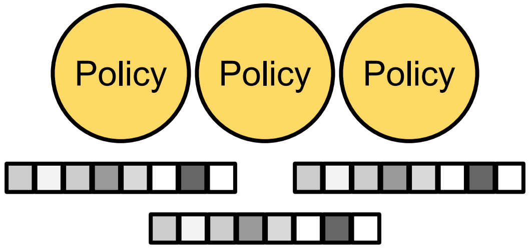
Approach - Anomaly detection

- Each policy node is represented by a vector



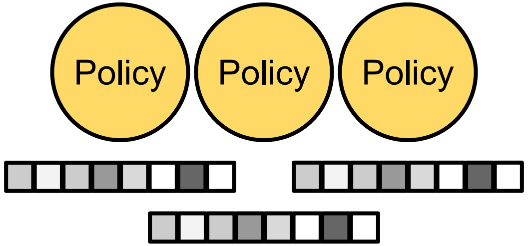
Approach - Anomaly detection

- Each policy node is represented by a vector
- We can train an anomaly detection model to find anomalous policies



Approach - Anomaly detection

- Each policy node is represented by a vector
- We can train an anomaly detection model to find anomalous policies
  - One-Class SVM
  - Local Outlier Factor
  - Isolation Forest
  - Robust Covariance

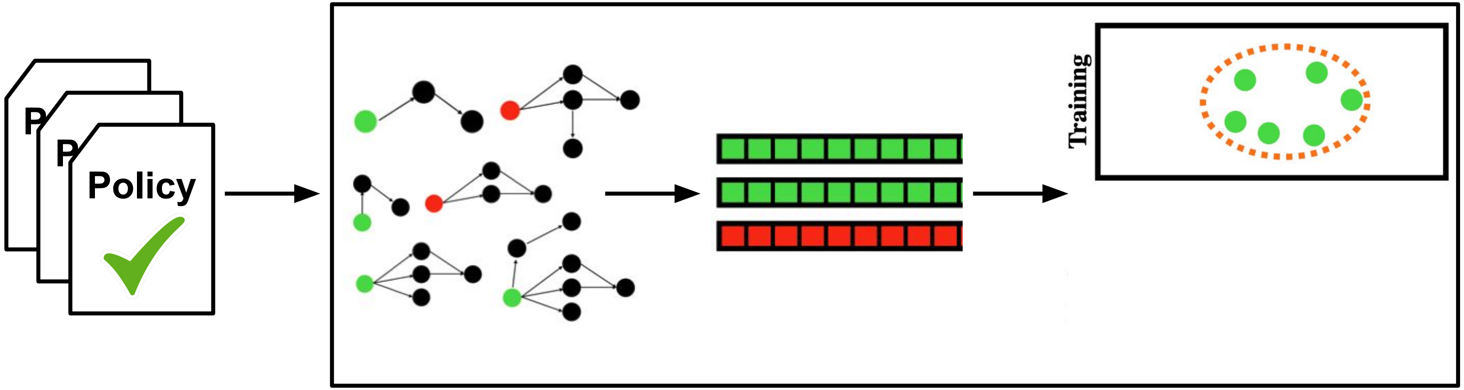


How does this work in practice?

- Security operators manually verify a set of policies

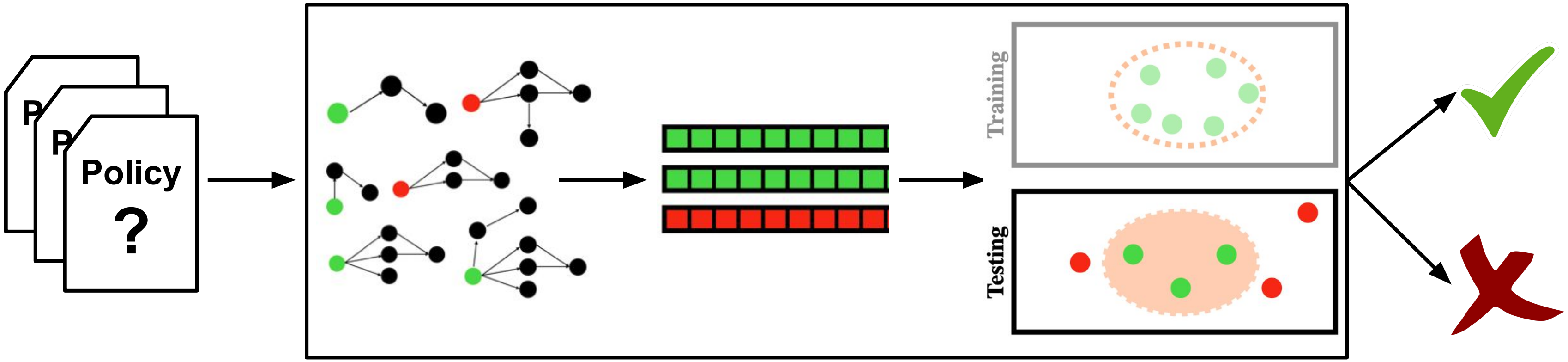
How does this work in practice?

- Security operators manually verify a set of policies
- We run our approach and train the anomaly detector



How does this work in practice?

- Security operators manually verify a set of policies
- We run our approach and train the anomaly detector
- When new policies are added, we run our pipeline to check if we find an anomaly



Evaluation - Are anomalies misconfigurations?

- Evaluated on 3 real-world datasets

Dataset	Total number of					Number of
	employees	policies	users	groups	roles	collections
1	12,000	842	0	0	55	8
2	130	812	0	0	34	2
3	4	826	2	1	10	12

Evaluation - Are anomalies misconfigurations?

- Evaluated on 3 real-world datasets
  - Dataset 1 & 2 are SSO users

Dataset	Total number of employees	Total number of policies	Total number of users	Total number of groups	Total number of roles	Number of collections
1	12,000	842	0	0	55	8
2	130	812	0	0	34	2
3	4	826	2	1	10	12

Evaluation - Are anomalies misconfigurations?

- Evaluated on 3 real-world datasets
  - Dataset 1 & 2 are SSO users
  - Data was periodically collected using our tool

Dataset	Total number of employees	policies	users	groups	roles	Number of collections
1	12,000	842	0	0	55	8
2	130	812	0	0	34	2
3	4	826	2	1	10	12

Evaluation - Are anomalies misconfigurations?

- Evaluated on 3 real-world datasets
- Compared with rule-based Cloud Custodian

Dataset	Total number of					Number of
	employees	policies	users	groups	roles	collections
1	12,000	842	0	0	55	8
2	130	812	0	0	34	2
3	4	826	2	1	10	12

Evaluation - Are anomalies misconfigurations?

- Evaluated on 3 real-world datasets
- Compared with rule-based Cloud Custodian
- Increased detection of **misconfigurations**

		Our approach			Cloud Custodian All rules			Cloud Custodian Selected rules		
	DS	Prec.	Recall	F1-score	Prec.	Recall	F1-score	Prec.	Recall	F1-score
Misconf.	1	66.67%	66.67%	66.67%	7.89%	10.34%	4.48%	100.00%	10.34%	9.37%
	2	70.00%	63.34%	66.67%	13.73%	17.07%	7.61%	100.00%	17.07%	14.58%
	3	75.00%	50.00%	60.00%	15.38%	11.32%	6.52%	100.00%	11.32%	10.17%
Overall	1	91.58%	91.58%	91.58%	97.93%	97.60%	97.76%	98.99%	98.98%	98.57%
	2	92.03%	92.31%	92.15%	97.40%	97.09%	97.24%	98.75%	98.73%	98.28%
	3	94.97%	95.45%	95.03%	98.93%	97.88%	96.87%	98.12%	98.08%	97.33%

Evaluation - Are anomalies misconfigurations?

- Evaluated on 3 real-world datasets
- Compared with rule-based Cloud Custodian
- Increased detection of **misconfigurations** but more **FPs**

	DS	Our approach			Cloud Custodian All rules			Cloud Custodian Selected rules		
		Prec.	Recall	F1-score	Prec.	Recall	F1-score	Prec.	Recall	F1-score
Misconf.	1	66.67%	66.67%	66.67%	7.89%	10.34%	4.48%	100.00%	10.34%	9.37%
	2	70.00%	63.34%	66.67%	13.73%	17.07%	7.61%	100.00%	17.07%	14.58%
	3	75.00%	50.00%	60.00%	15.38%	11.32%	6.52%	100.00%	11.32%	10.17%
Overall	1	91.58%	91.58%	91.58%	97.93%	97.60%	97.76%	98.99%	98.98%	98.57%
	2	92.03%	92.31%	92.15%	97.40%	97.09%	97.24%	98.75%	98.73%	98.28%
	3	94.97%	95.45%	95.03%	98.93%	97.88%	96.87%	98.12%	98.08%	97.33%

Evaluation - Ar nfigurations?

- Evaluated on 3
- Compared with an
- Increased dete s but more **FPs**

Precision and recall can be **tuned** in anomaly detector

	DS	Our approach			Cloud Custodian All rules			Cloud Custodian Selected rules		
		Prec.	Recall	F1-score	Prec.	Recall	F1-score	Prec.	Recall	F1-score
Misconf.	1	66.67%	66.67%	66.67%	7.89%	10.34%	4.48%	100.00%	10.34%	9.37%
	2	70.00%	63.34%	66.67%	13.73%	17.07%	7.61%	100.00%	17.07%	14.58%
	3	75.00%	50.00%	60.00%	15.38%	11.32%	6.52%	100.00%	11.32%	10.17%
Overall	1	91.58%	91.58%	91.58%	97.93%	97.60%	97.76%	98.99%	98.98%	98.57%
	2	92.03%	92.31%	92.15%	97.40%	97.09%	97.24%	98.75%	98.73%	98.28%
	3	94.97%	95.45%	95.03%	98.93%	97.88%	96.87%	98.12%	98.08%	97.33%

Conclusion

Using **anomaly detection** in IAM policies:

- **Increases** the number of detected **misconfigurations**
- **Incorrectly** flags **slightly more** policies than rule-based solutions
- Requires **fewer** manual steps than rule-based solutions

`https://github.com/utwente-scs/misdet-code`

Questions?

Using anomaly detection in IAM policies:

- **Increases** the number of detected **misconfigurations**
- **Incorrectly** flags **slightly more** policies than rule-based solutions
- Requires **fewer** manual steps than rule-based solutions

`https://github.com/utwente-scs/misdet-code`

Thijs van Ede
t.s.vanede@utwente.nl
@EdeThijs

pwc
UNIVERSITY OF TWENTE.