Network and Distributed System Security (NDSS) Symposium 2025

CASPR: Context-Aware Security Policy Recommendation

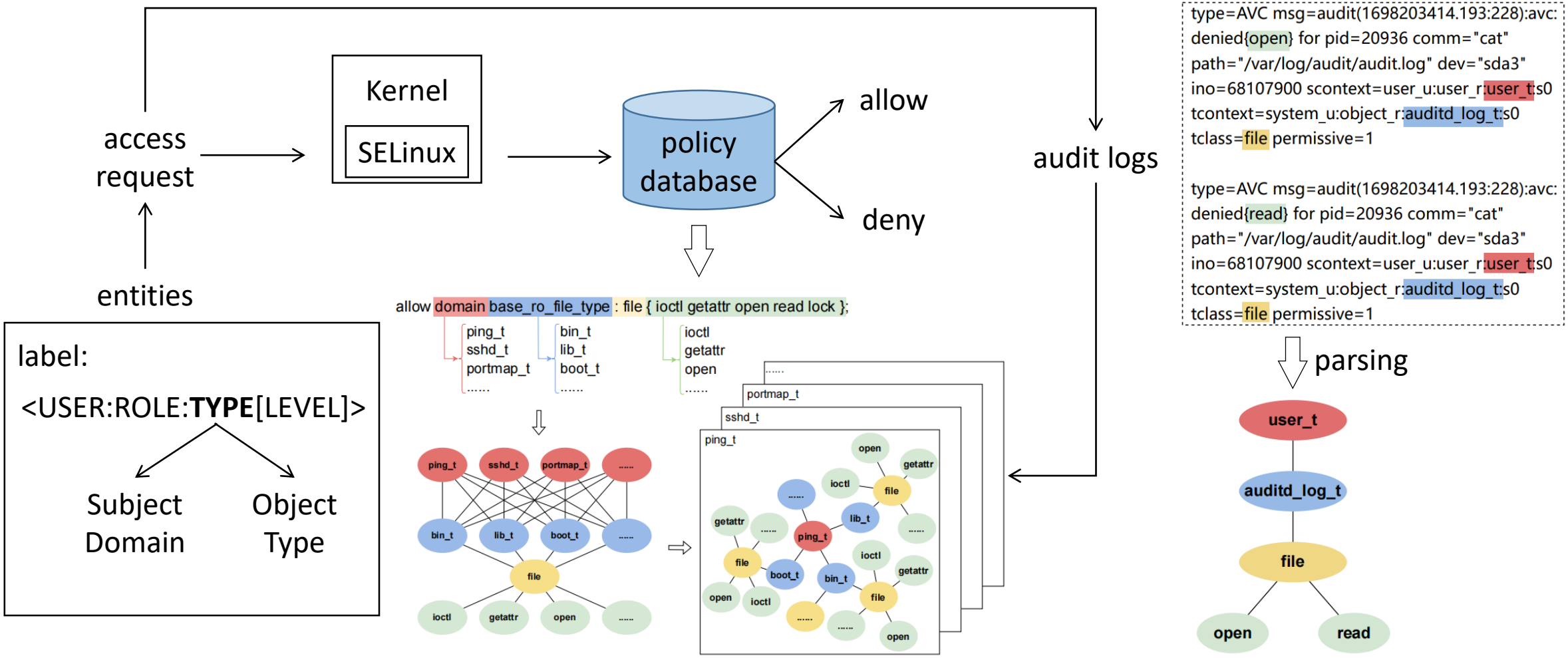
中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

中国科学院大学
University of Chinese Academy of Sciences

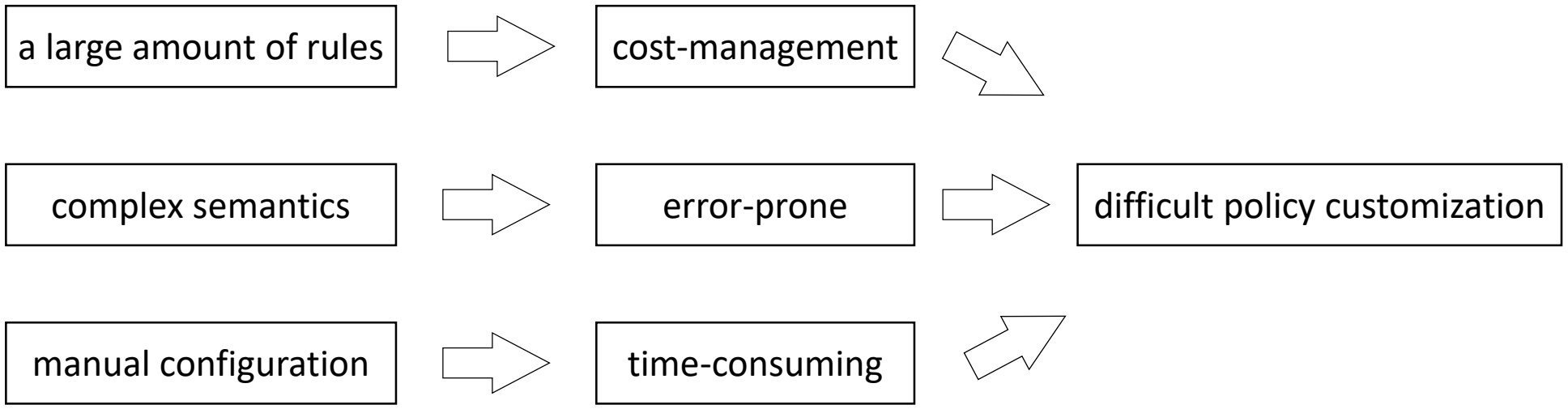
Lifang Xiao, Hanyu Wang, Aimin Yu, Lixin Zhao, Dan Meng

Background of Access Control based on SELinux

SELinux (Security Enhanced Linux) controls interactions between entities through security policies to achieve the least privilege to prevent unauthorized access.



Challenges of Policy Recommendation



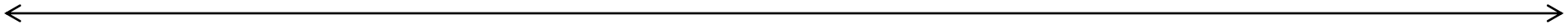
Previous policy analysis lacks policy recommendation for **newly defined types**, which lacks rules used as a reference.

Our insights:

Context-aware information → Similar privileges → Policy recommendations

Case Study

Proper configuration of policy



Under-privileging

Disrupt normal operations

If the administrator **changes the default port** of Apache HTTP server from port 80 to port 3131 or **stores content in a non-standard directory**. However, the new directory and port have not been appropriately labeled for httpd access. As a result, the server returns error due to **SELinux denying access**.

Over-privileging

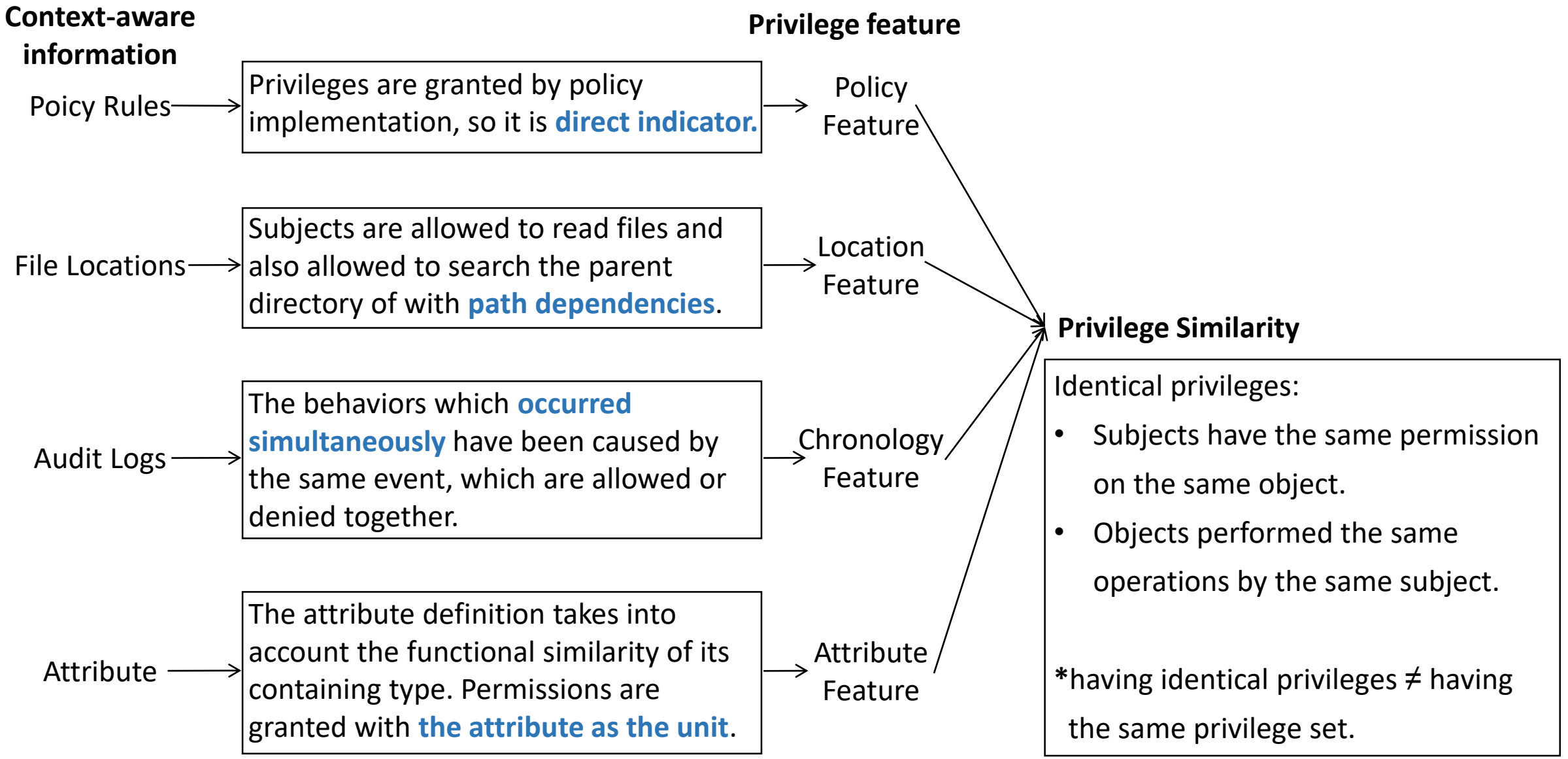
Execute malicious attacks

CVE-2019-13272:
SElinux policy enables attackers to hjack a child process by gaining control of parent process. They can then manipulate the child process through the ptrace function, which is **typically denied by standard SELinux policy**.

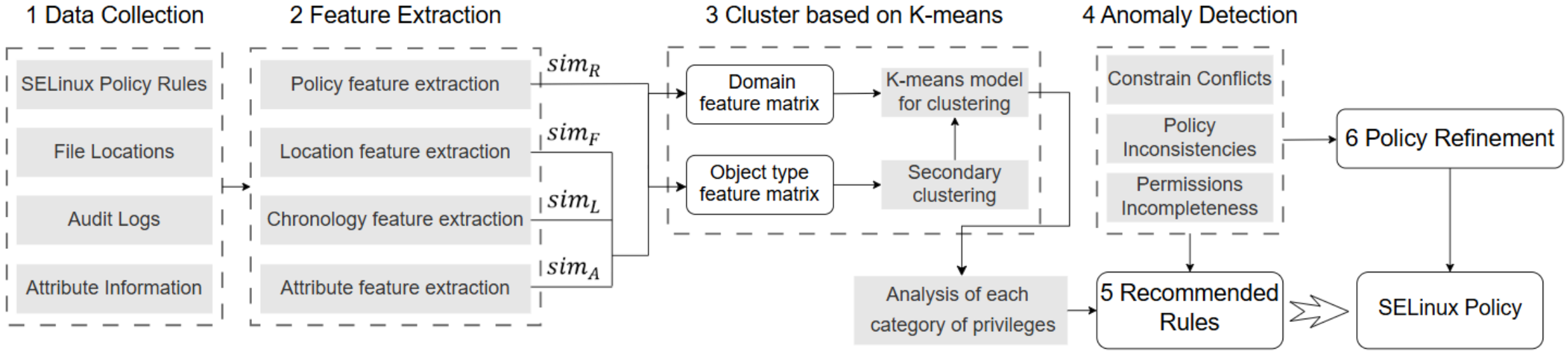
Goal:

Recommend modifications of inappropriate rules for better management of permissions.

Context-Aware Information

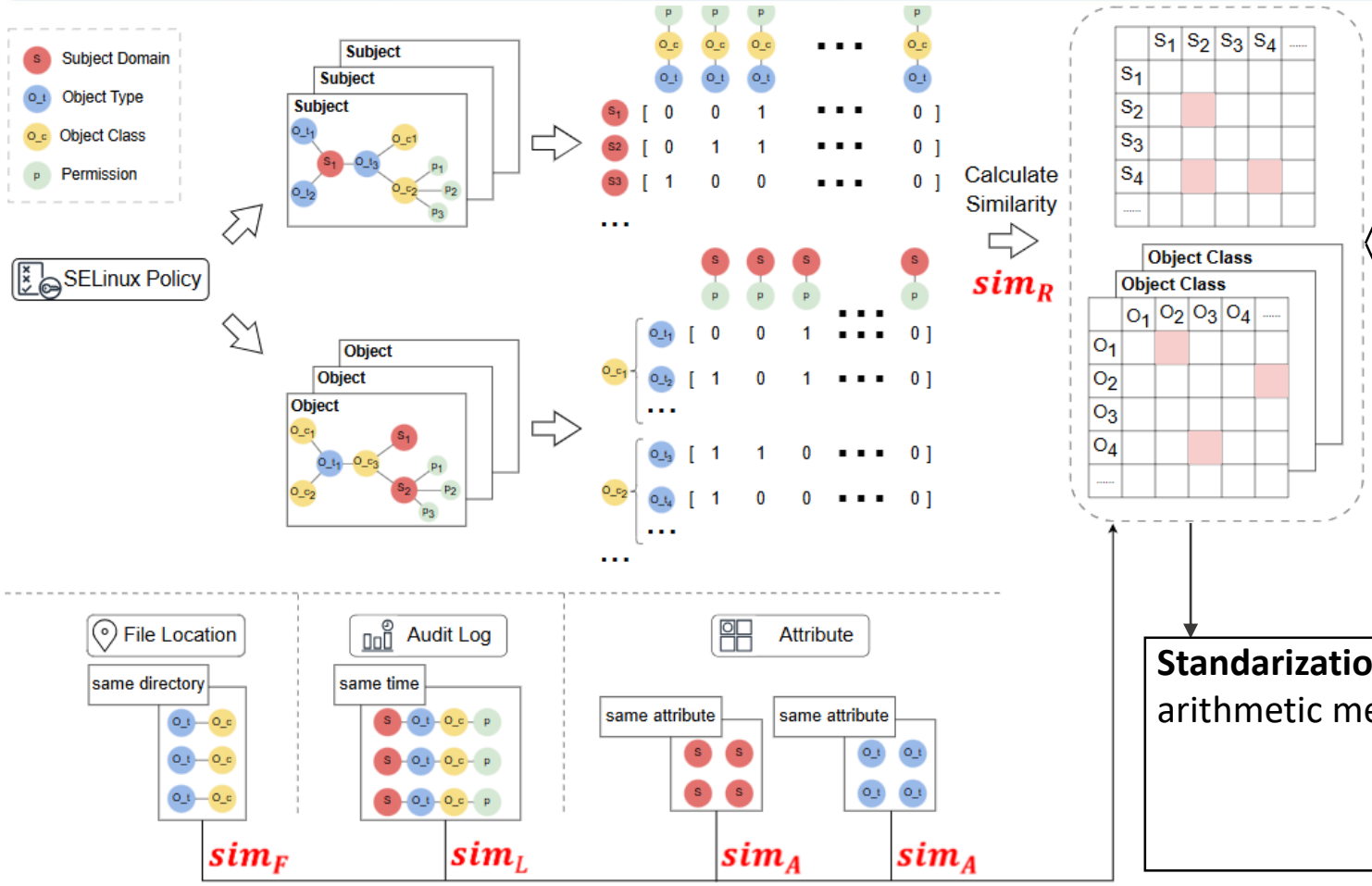


CASPR -- Workflow



For **privilege computation** and **policy recommendation**, CASPR uses **context-aware information as features** and domains and object types as samples. We train a **K-means model** with a feature matrix of privilege similarity. To address the issue of a huge number of object types in SELinux, we adopt a **secondary clustering** approach when clustering the objects of the policy. CASPR not only recommends rules but also performs anomaly detection to ensure effective implementation of policy.

CASPR -- Context-aware Feature Computation



- For similarity of security policies:

$$sim(a,b) = \frac{len(a\&b)}{len(a|b)}$$

- For similarity of other features:

The value is set to 1, if two samples exhibit similarity in the dimensions of the features.

  - path dependency
  - frequent simultaneous access
  - belonging to the same attribute

Standarization: Convert to a distribution with an arithmetic mean of 0 and a standard deviation of 1.

$$x_{standard} = \frac{x - \mu}{\sigma}$$

Formalization: Scale matrix to the range of 0 to 1.

$$x_{normal} = \frac{x - min(x)}{max(x) - min(x)}$$

CASPR -- Rule Recommendation Based on Clustering

Algorithm 1: Domains and object types clustering.

Input: Policy, Location, Log, Attribute
Output: Clustering Results of subjects and objects.

```
1  rules← GetPrivilege(policy);
2  relationship_location← GetNearFile(Location);
3  relationship_chronology← GetEvent(Log);
4  relationship_type← GetAttribute(Attribute);
5  subject← GetDomain(policy+added_domain);
6  for rule ∈ rules do
7  |  subject_vector[subject].update(rule[object_type]+ rule[object_class]
8  |  +rule[permission]);
9  end
10 for s1 ∈ subjects do
11 |  for s2 ∈ subjects do
12 |  |  matrix1[s1][s2].add(sim(s1_vector, s2_vector));
13 |  end
14 end
15 for relationship(s1, s2) in relationship do
16 |  matrix1[s1][s2].add(sim_i);
17 end
18 results_subject← K-means(martrix1);
19 objects← GetObjectType(rules+added_type) ;
20 for class ∈ rules[object_class] do
21 |  for rule ∈ rules do
22 |  |  object_vector[object].update(rule[domain]+ rule[permission]);
23 |  end
24 |  for o1 ∈ objects do
25 |  |  for o2 ∈ objects do
26 |  |  |  matrix2[o1][o2].add(sim(o1_vector, o2_vector));
27 |  |  end
28 |  end
29 |  for relationship(o1, o2) in relationship do
30 |  |  matrix2[o1][o2].add(sim_i);
31 |  end
32 |  results_object ← K-means(martrix2);
33 end
34 return results_subject, results_object;
```

Capture Features

Subjects Clustering

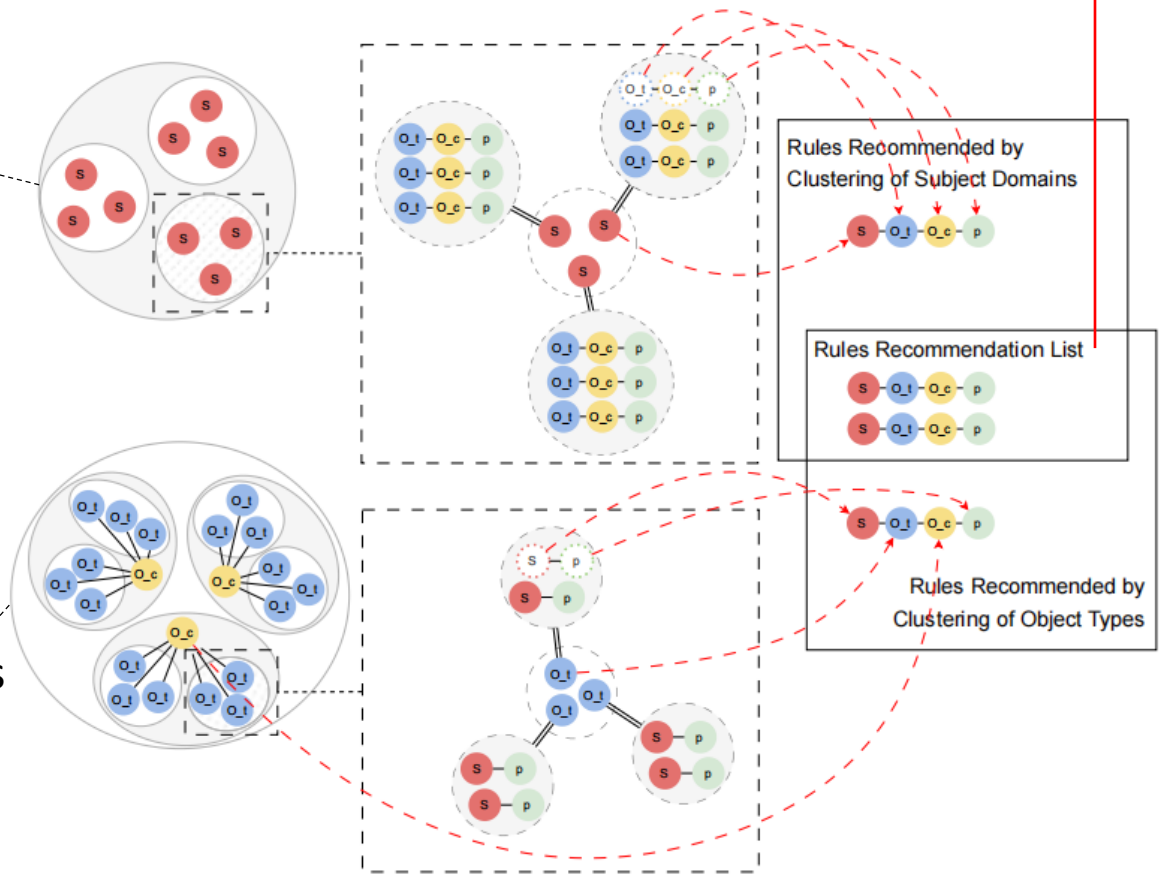
K-means model

Secondary Clustering of Objects

Clustering results indicates multiple identical privileges, not the same privilege sets.

✗ Blindly recommend rules ignoring differences between types.

✓ Generate lists seperately and calculate the intersaction.



CASPR -- Anomaly Detection and Policy Refinement

Policy integrity and availability requires **all the necessary rules** to perform the behavior instead of **adding one single rule** to execute a certain operation.

Grammar and semantic anomalies	Constraint file	Form	Illlustration
Constraint Conflicts	seinfo --constrain	$< Attr, c, \{p_i\} >$	Only types in attribute $Attr$ have permissions p_i on objects of class c.
Policy Inconsistencies	file_patterns.spt	$< beh: [\{beh\_f\}, \{beh\_s\}] >$	If a subject performs beh behavior, it needs privileges not only to perform beh_s on the file but also to perform beh_f on its parent directory.
Permission Incompleteness	obj_perm_sets.spt	$< c: \{beh: \{p_i\}\} >$	If a subject performs beh behavior on an object of class c, it needs to gain permissions of p_i.

Evaluation -- Real-World Evaluation

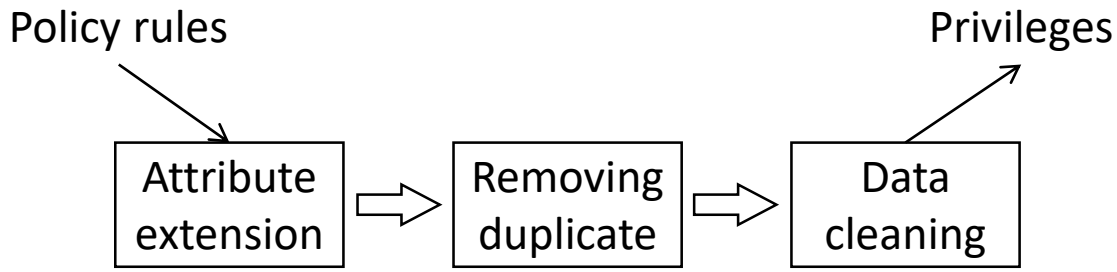
Operating Systems	SELinux Version	Rules	Privileges	File Locations	Audit Logs	Attribute Mappings
CentOS6	3.7.19-312.el6	304,755	128,258,009	3,904	8,255	8,814
CentOS7	3.13.1-268.el7	99,054	173,514,894	5,372	8,748	14,950
CentOS8	3.14.3-139.el8	108,038	197,389,960	5,609	8,326	13,581
Ubuntu20	2:2.20190201-8	96,758	43,008,323	4,458	6,854	9,402
Ubuntu22	2:2.20210203-10	90,772	35,880,295	4,287	7,749	8,871
Ubuntu24	2:2.20240202-1	36,400	37,213,711	4,484	7,753	8,935

Context-aware information list for different operating systems

	Accuracy	Precision	Recall	F1-score	FPR
CentOS6	91.163%	92.859%	94.987%	93.905%	15.214%
CentOS7	92.439%	93.472%	94.627%	94.046%	11.302%
CentOS8	92.687%	93.085%	93.508%	93.323%	14.365%
Ubuntu20	90.584%	90.925%	92.945%	93.457%	12.283%
Ubuntu22	91.422%	92.649%	94.384%	93.986%	14.744%
Ubuntu24	91.196%	91.394%	93.481%	93.847%	14.497%
Avg.	91.582%	92.397%	93.982%	93.761%	13.734%

Performance of CASPR for different versions of SELinux policy

- **Data Process**



- **Universality between different versions**

  We employ CASPR in **six operating systems** to demonstrate the adaptability of CASPR to various versions of SELinux policy rule recommendations.

- **Running time**

  Initialization: 1 hour

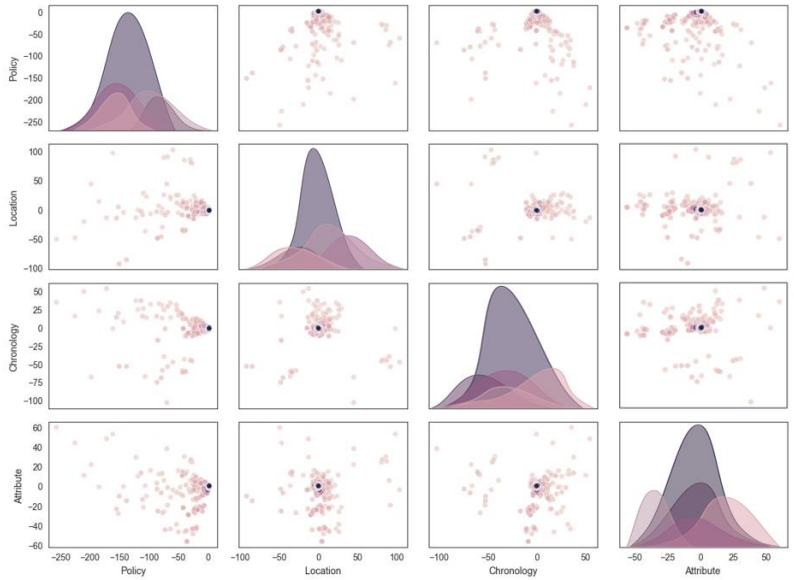
  Clustering subject domains: 1.5 min

  Clustering object types: 8 min

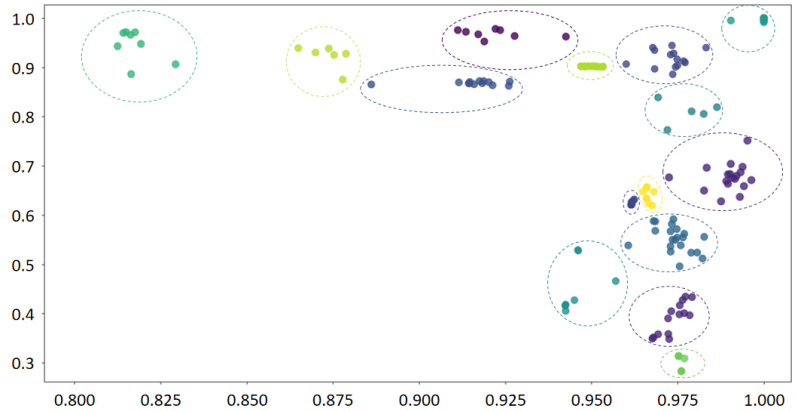
  Policy recommendation: 3min

Evaluation -- Clustering of Subjects and Objects

Clustering results



Clustering results of subject domains



Clustering results of object types

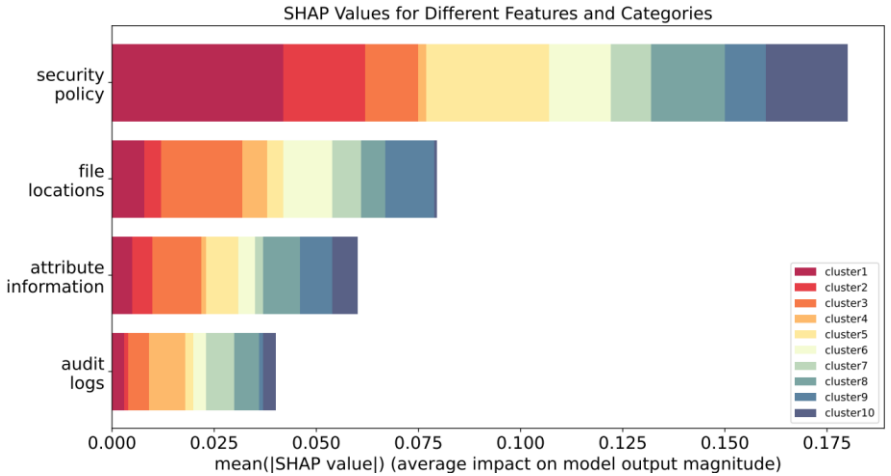
Through **PCA dimensionality reduction** in clustering, it is clear that there are multiple domains **clustered into one category** based on selecting context-aware information as features.

Both subject domains and object types have apparent clustering effect.

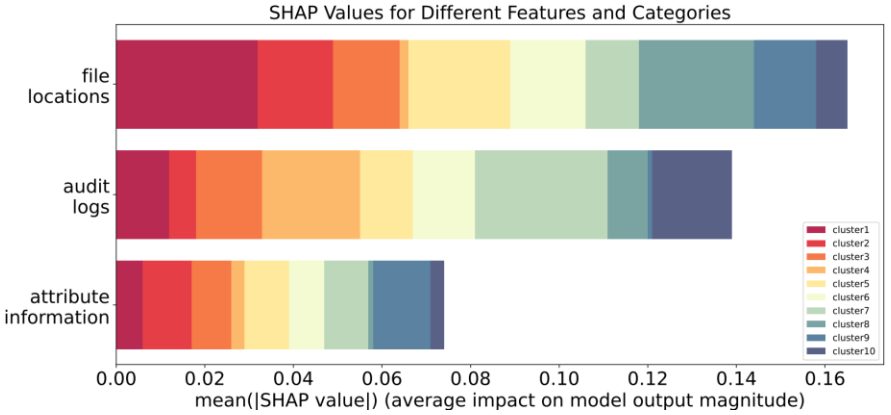
We use the **SHAP interpretable program**, which assigns a value to each feature and explains the relevance of the feature to the classification results.

The **policy feature** is more decisive in determining identical privileges. When clustering new types, the **attribute feature** has less contribution.

Contribution of context-aware feature based on SHAP interpretability analysis



Mean absolute value of SHAP for clustering of existing types



Mean absolute value of SHAP for clustering of new types

Evaluation -- Rule Recommendation

Threshold Setting	True Malicious (TP)	False Malicious (FP)	True Benign (TN)	False Benign (FN)	True Positive Rate (TPR)	False Positive Rate (FPR)	Accuracy	F1-score
n=2	69.631%	30.369%	41.903%	58.097%	47.895%	35.726%	53.938%	61.153%
n=4	73.985%	26.015%	49.729%	50.271%	63.444%	38.154%	62.854%	65.982%
n=6	78.697%	21.303%	56.694%	43.306%	69.764%	32.298%	69.003%	70.897%
n=8	82.722%	17.278%	64.263%	35.737%	76.353%	27.275%	75.014%	75.732%
n=10	93.472%	6.528%	90.612%	9.388%	94.627%	11.302%	92.439%	94.046%
n=12	89.503%	10.497%	80.034%	19.966%	87.988%	17.649%	89.503%	85.457%

Performance of CASPR recommendation rules under different thresholds

- **Performance of rule recommendation**

CASPR achieves 92.439% accuracy, 93.472% precision, 94.627% recall, and 94.046% F1-score.

Despite few false sample, CASPR reduces the effort for manual configuration and makes a **positive effort to narrow the attack surface**.

- **Recommend rules for new types**

For the new types, the crucial feature of policy is absent, causing its accuracy is **slightly lower**, which is 82.366%.

- **Accuracy at different thresholds**

We use the average number of samples in each category, iie., $n = all\_samples/k$ as the **criterion for clustering granularity**.

When $n = 10$, the accuracy of the recommendation reaches the highest.

Evaluation -- Comparison with Baseline and Anomaly Detection

Comparison with other policy recommendation methods

Metrics	Threshold	Evaluation Indicators				
		Accuracy	Precision	Recall	F1-score	FPR
CASPR	n=10, $0.2<\theta<0.3$	92.439%	93.472%	94.627%	94.046%	11.302%
EASEAndroid	m=10, σ=55%, $Dist$=1	78.193%	84.166%	80.067%	82.348%	25.935%
SEPAL	-	88.436%	84.794%	89.286%	86.983%	12.078%

- **EASEAndroid:**
  - nearest-neighbors classifier + pattern-to-rule distance measure judge
  - **policy feature**
- **SEPAL:**
  - wide & deep learning model
  - **attribute + user ID + NLP-based features**

Anomaly Detection

Out of **the recommended 97,583 rules**, we identify 168 anomalies, including 46 constraint conflicts, 54 policy inconsistencies and 58 permission incompleteness.

We **artificially generate some sets of rules** to assess the effectivenes of anomaly detection.

Eliminating these anomalies requires r**ecalculating the rules involved in the anomalies**. If the majority are not recommended, they are deleted. If the majority are recommended, they are classfied in the recommended list.

Conclusion

- **Privilege calculation based on context-aware information:**

  We innovatively introduce **context-aware information** for privilege calculation and integrate them, including policy rules, file locations, audit logs, and attribute information.

- **Rule recommendation and anomaly detection:**

  We propose CASPR, **a rule recommendation and anomaly detection method**, which establishes a privilege similarity matrix based on context-aware information to cluster the domains and object types and automatically recommends policy rules and detects and refines anomalies.

- **Experimental effectiveness:**

  We perform experiments to prove that **CASPR improves the accuracy of policy recommendations with higher adaptability and feasibility.** We employ CASPR in multiple operating systems to illustrate its universality. We also demonstrate the clustering effect and the contribution of each context-aware feature. The average accuracy of CASPR achieves 91.582%, which exeeds other rule recommendation methods.

Network and Distributed System Security (NDSS) Symposium 2025

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

中国科学院大学
University of Chinese Academy of Sciences

CASPR: Context-Aware Security Policy Recommendation

Lifang Xiao, Hanyu Wang, Aimin Yu, Lixin Zhao, Dan Meng

Thank you for your attention!