CE 815 – Secure Software Systems

Causal Analysis (ShadeWatcher)

Mehdi Kharrazi
Department of Computer Engineering
Sharif University of Technology

Acknowledgments: Some of the slides are fully or partially obtained from other sources. A reference is noted on the bottom of each slide, when the content is fully obtained from another source. Otherwise a full list of references is provided on the last slide.

Review

- APT attacks, Causal analysis, Provenance graph
- Poirot
  - Looking for known attack in audit log
  - Graph matching was the problem
  - They know attack scenario
- Atlas
  - Learn benign and attack sequences
  - Start from a symptom try to construct attack story
  - Is there any way to find attacks with no prior knowledge?
- Introduce anomaly detection

SHADEWATCHER: Recommendation-guided Cyber Threat Analysis using System Audit Records, J. Zeng, X. Wang, J. Liu, Y. Chen, Z. Liang, T.S. Chua, Z. Leong Chua, IEEE Security & Privacy, 2022.

Cyber Threats Are Everywhere

MICROSOFT \ TECH \ WINDOWS

Microsoft confirms Lapsus$ hackers stole source code via 'limited' access

Lap

By Mi

If you

Home / Technology / Tech / Cisco latest victim of Russian cyber attack using SolarWinds

Cisco latest victim of Russian cyber attack using SolarWinds

Cisco

gover

the da

NICHOLAS THOMPSON BRIAN BARRETT SECURITY SEP 24, 2020 12:00 PM

How Twitter Survived Its Biggest Hack—and Plans to Stop the Next One

On July 15, Twitter melted down. On Election Day, that's not an option.

Microsoft

cisco

How to combat cyber threats through attacker's footprints left in systems?

Analyze Cyber Threat using System Auditing

Audit records are a valuable source for analyzing cyber threats:

- Provide a low-level view by monitoring **system entity interactions**
- Navigated through a **provenance graph** that describes a system's historical contexts

1. … **password**
2. **gtcache**, **read**, **/etc/passwd**
3. **gtcache**, **clone**, **ztmp**
4. **ztmp, send, 162.66.239.75**
5. …

Data Exfiltration

Analyze Cyber Threat using System Auditing

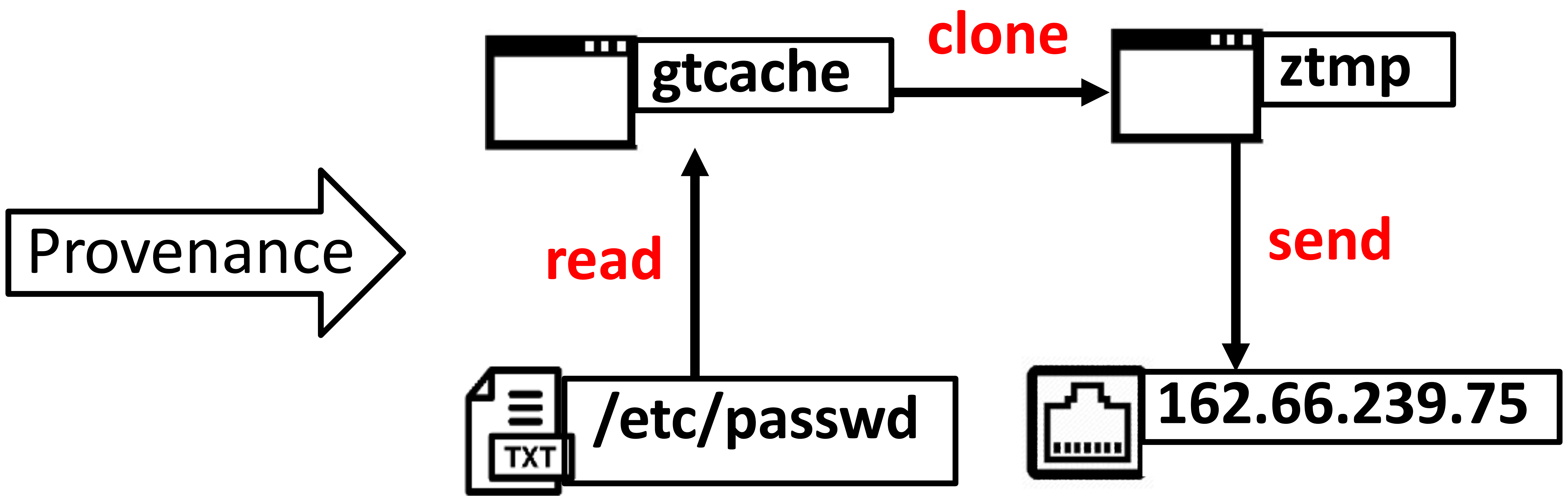
Audit records are a valuable source for analyzing cyber threats:

- Provide a low-level view by monitoring **system entity interactions**
- Navigated through a **provenance graph** that describes a system's historical contexts

password

1. …
2. **gtcache**, **read**, **/etc/passwd**
3. **gtcache**, **clone**, **ztmp**
4. **ztmp**, **send**, **162.66.239.75**
5. …

Data Exfiltration



System auditing connects separate attack steps, presenting the **overall** attack scenario

Previous Approaches using Audit Records

Statistics-based approaches [NDSS’18, NDSS’19, …]:

- Quantify audit records’ degrees of suspicion by their historical frequency
- **False-positive** prone

Specification-based approaches [USENIX Security’17, CCS’19, S&P’19, …]:

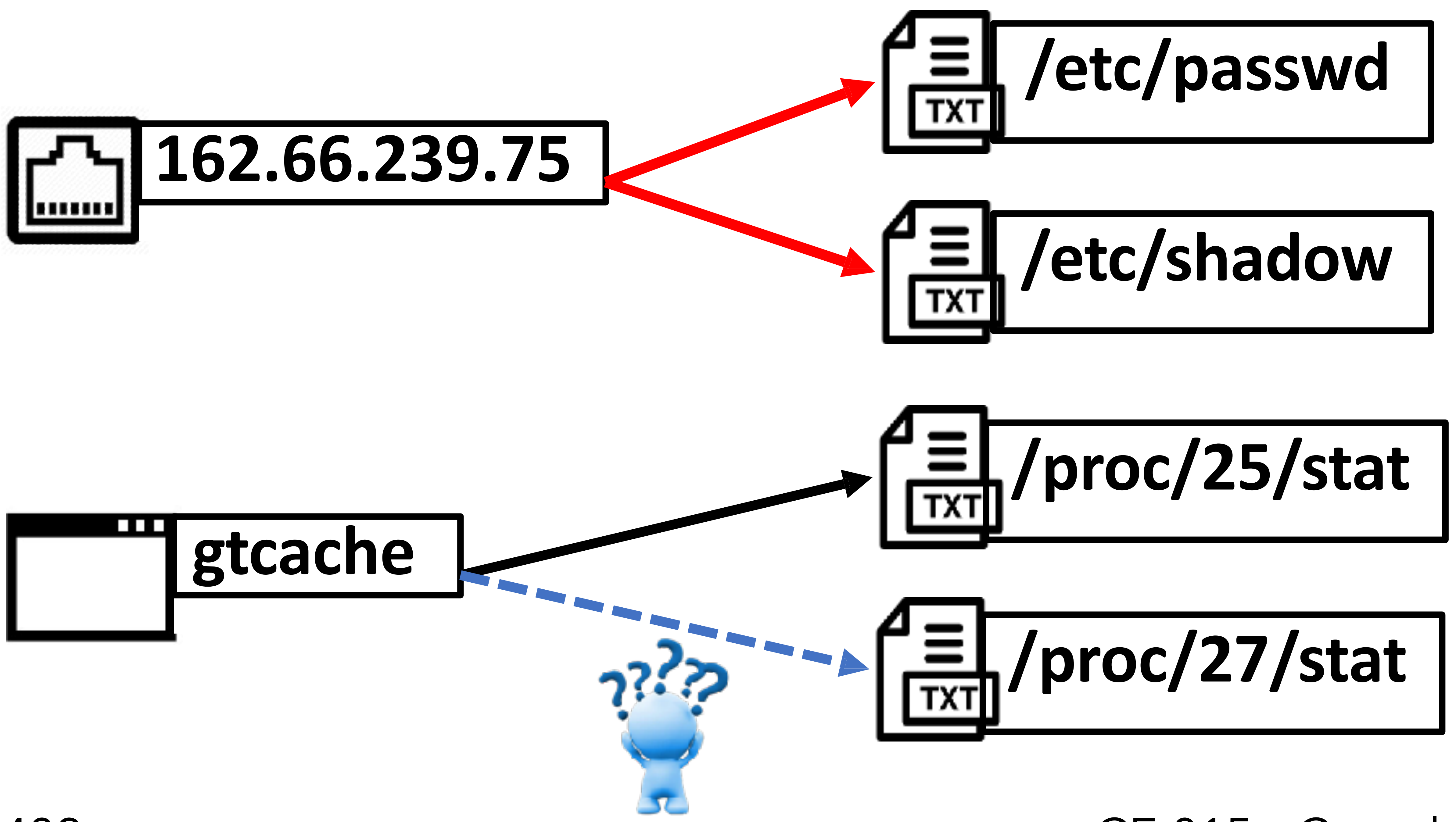
- Match audit records against a knowledge base of security policies
- **Time-consuming** and **error-prone** to develop

Learning-based approaches [NDSS’20, USENIX Security’21, …]:

- Train a model of benign behaviors and detect deviations
- Produce detection signals at a **coarse-grained** level, leading to **extensive** manual efforts for attack investigation

Our Observation

- Cyber threats can be revealed by determining **how likely** a system entity would **interact** with another entity
  - Unlikely (or “Unintended”) interactions indicate cyber threats
  - Estimate such likelihood with **historical** system entity interactions



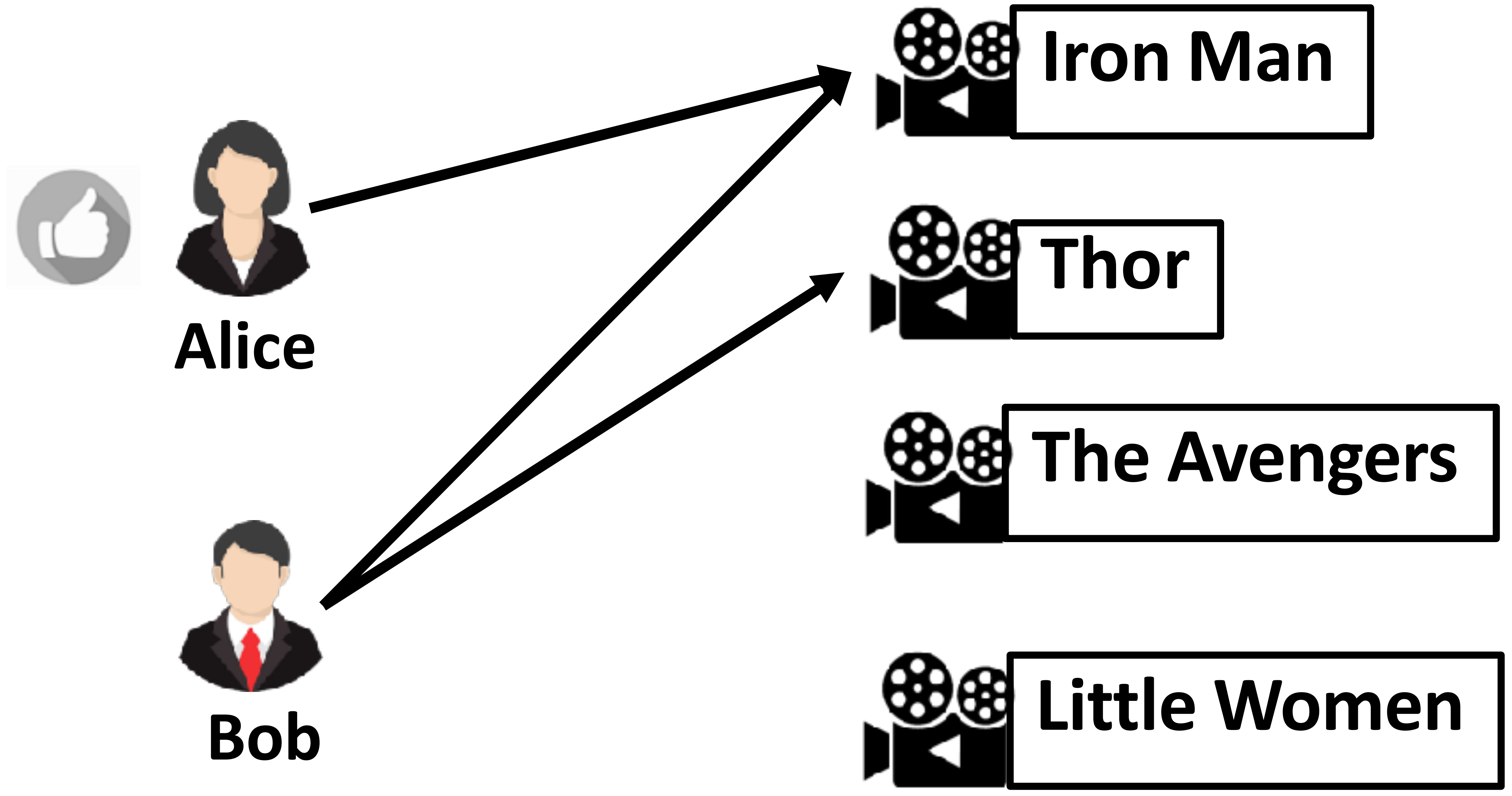
Sensitive files normally **do not** interact with public networks!

Should gtcache interact with /proc/27/stat? **Yes!**

Recommendation as a Similar Problem

A Similar problem has been explored in **Recommendation Systems**:

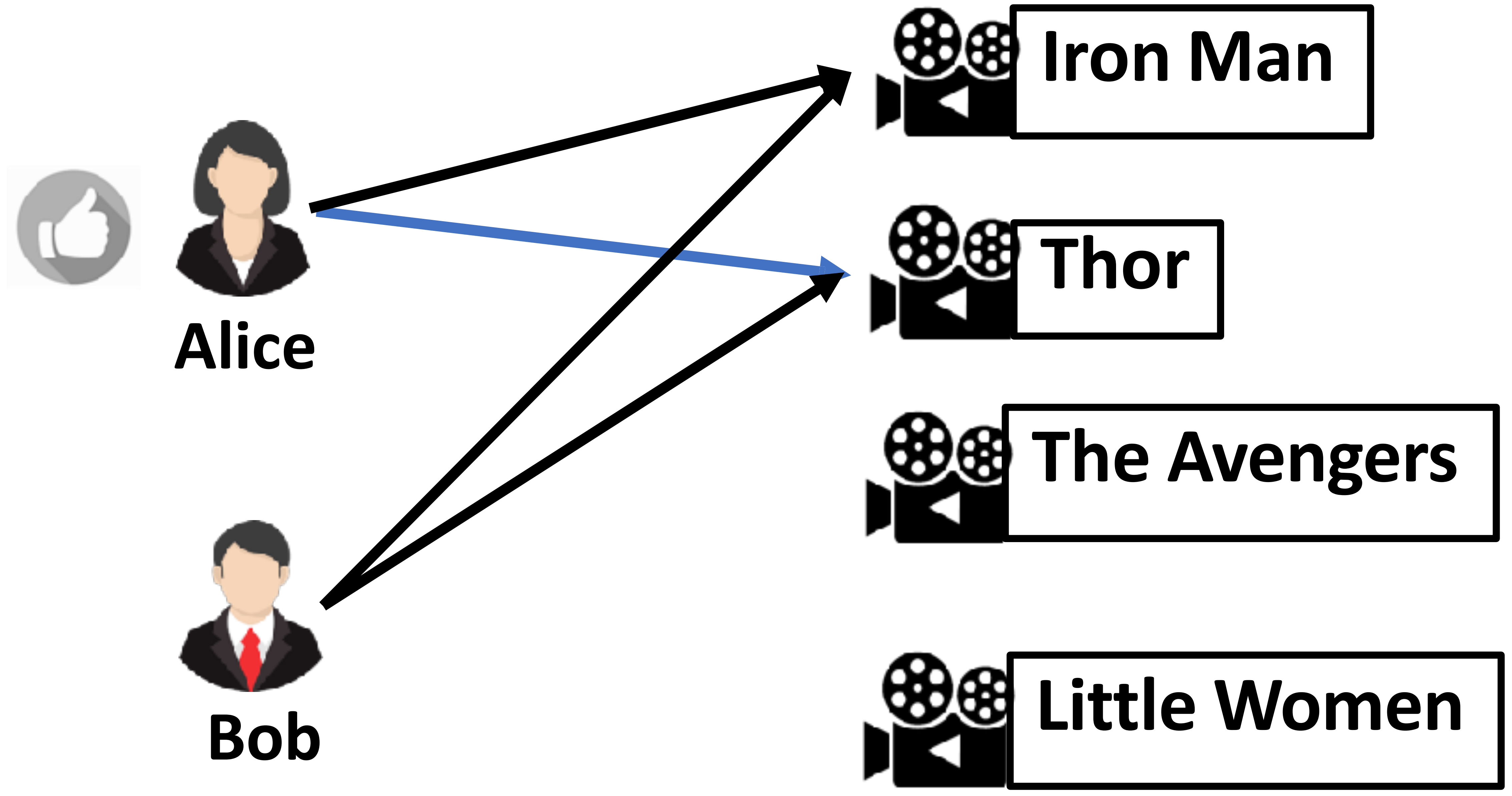
- Determine **how likely** a user would **interact** with an item
- **Similar** users share preferences on items: **historical** user-item interactions
- Item side information forms **high-order connectivity** that links **similar** items



Recommendation as a Similar Problem

A Similar problem has been explored in **Recommendation Systems**:

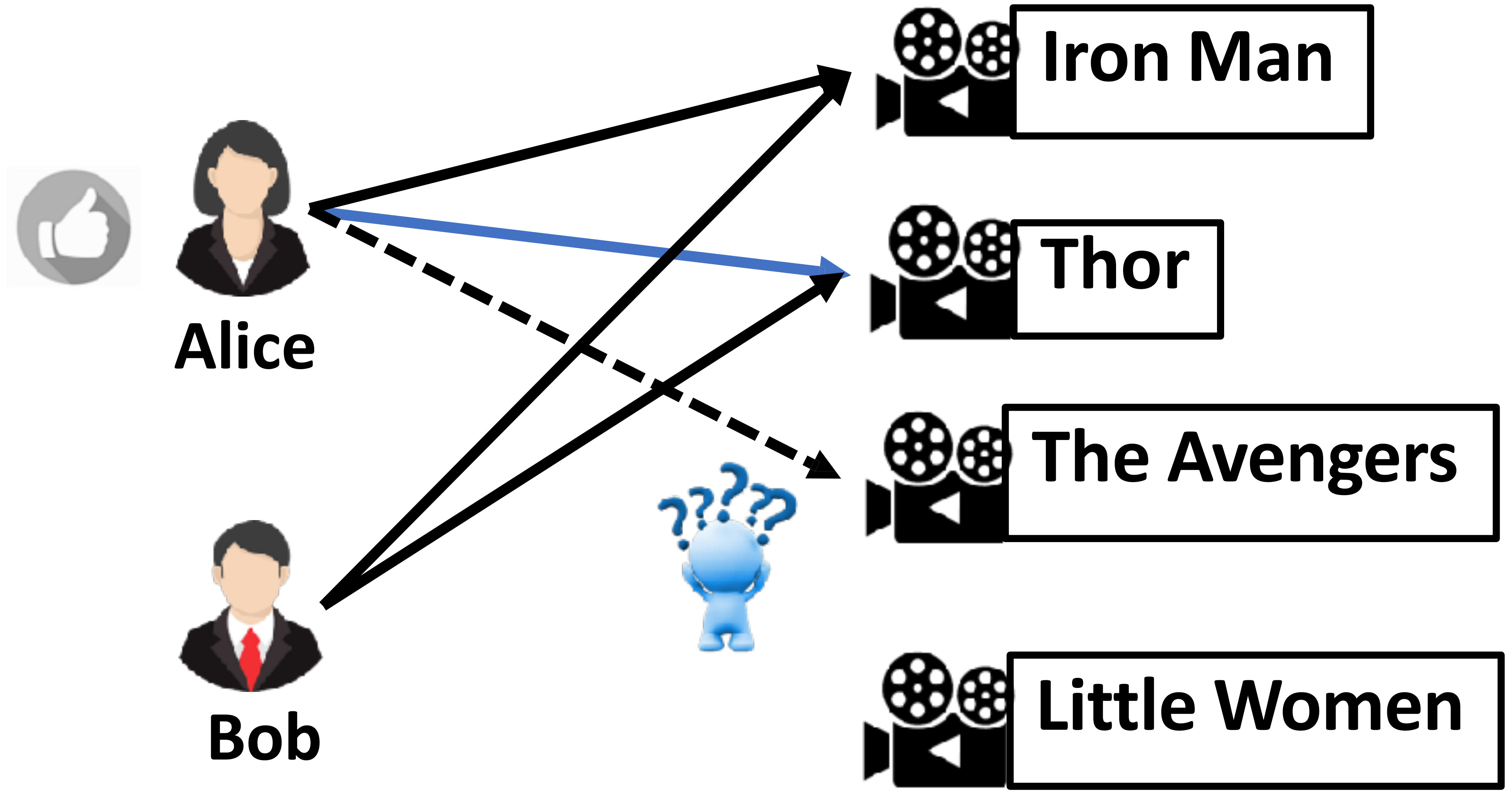
- Determine **how likely** a user would **interact** with an item
- **Similar** users share preferences on items: **historical** user-item interactions
- Item side information forms **high-order connectivity** that links **similar** items



Recommendation as a Similar Problem

A Similar problem has been explored in **Recommendation Systems**:

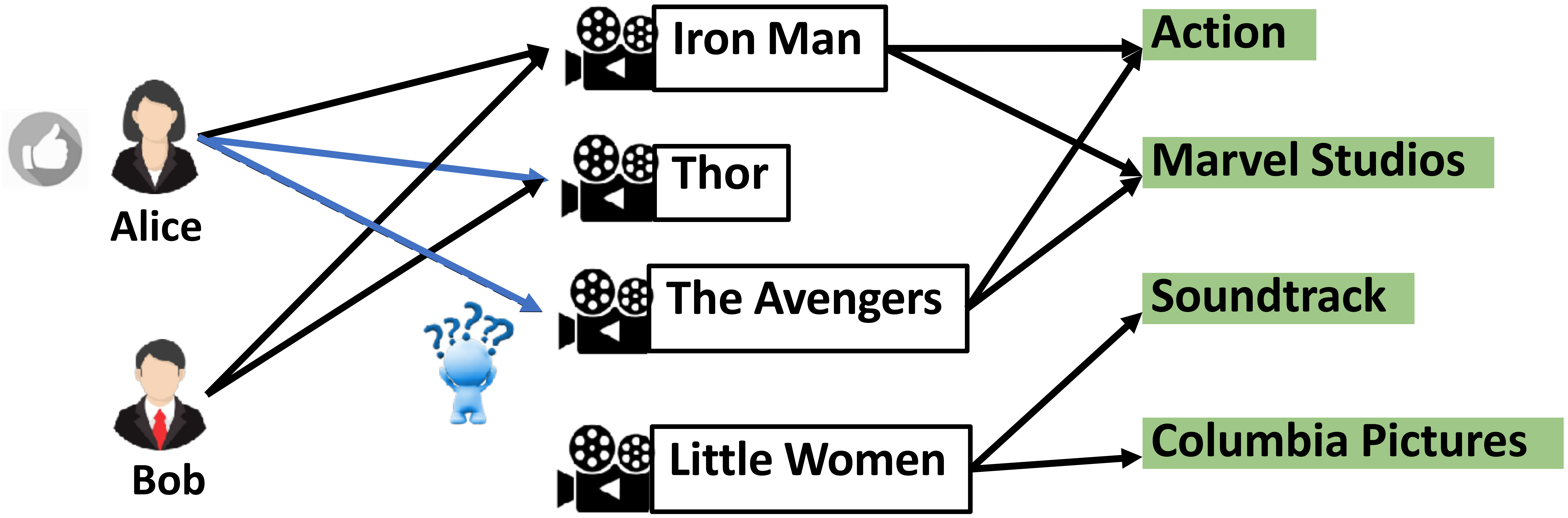
- Determine **how likely** a user would **interact** with an item
- **Similar** users share preferences on items: **historical** user-item interactions
- Item side information forms **high-order connectivity** that links **similar** items



Recommendation as a Similar Problem

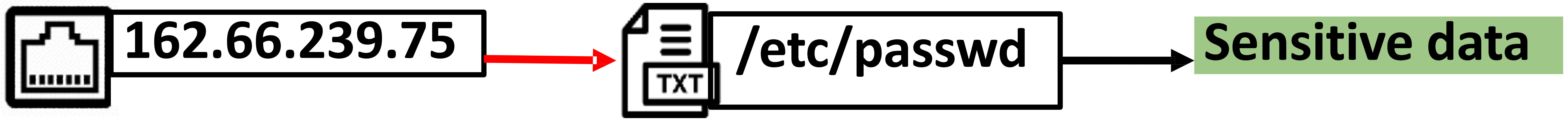
A Similar problem has been explored in **Recommendation Systems**:

- Determine **how likely** a user would interact with an item
- **Similar** users share preferences on items: **historical** user-item interactions
- Item side information forms **high-order connectivity** that links **similar** items

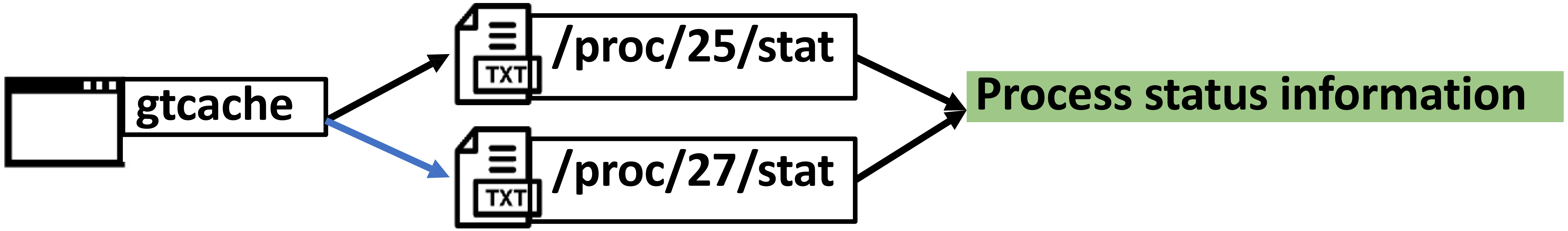


Recommendation-guided Cyber Threat Analysis

Observation: Similar system entities share preferences on interactions



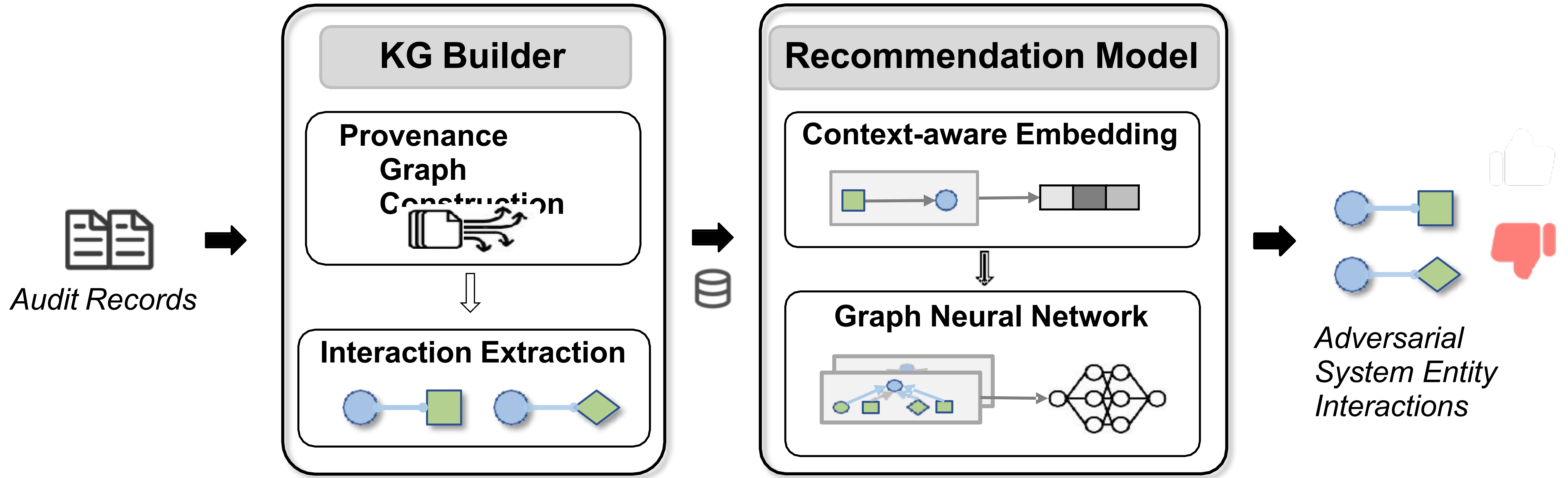
Insight: Identify high-order connectivity based on side information of system entities to better uncover their similarities



We formulate cyber threat analysis as a recommendation task:

How likely a system entity would **"prefer"** its interactive entities?

SHADEWATCHER: Overview

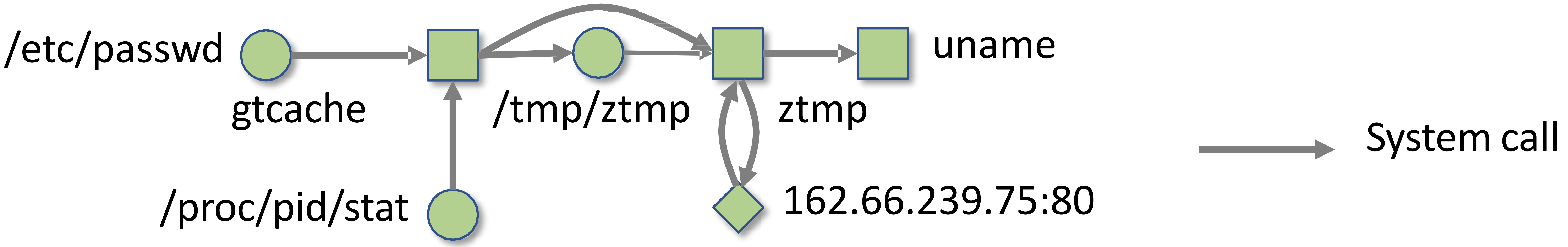


Input: Audit records collected by system auditing frameworks (e.g., Linux Audit)

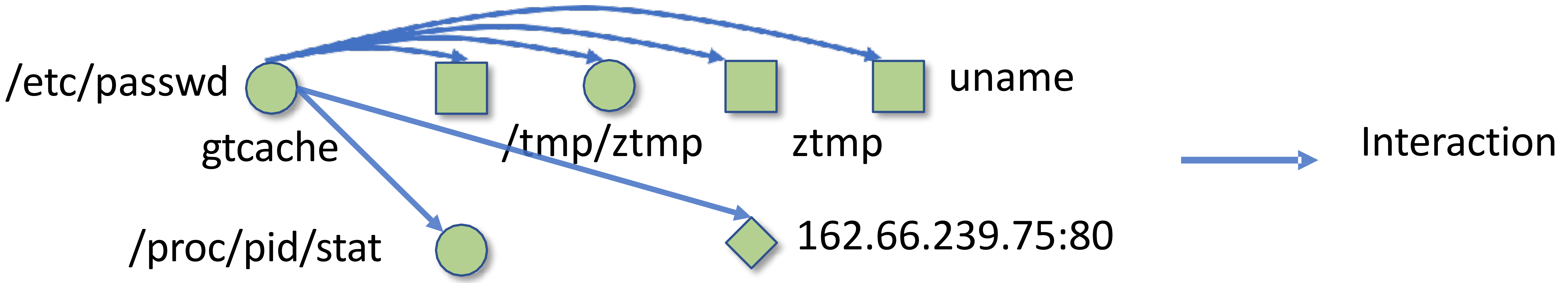
Output: Detection signals for adversarial system entity interactions

Knowledge Graph Builder

- Given audit records on end hosts, we parse them into a **provenance graph (PG)** and extract system entity interactions into a **bipartite graph (BG)**.



Provenance Graph

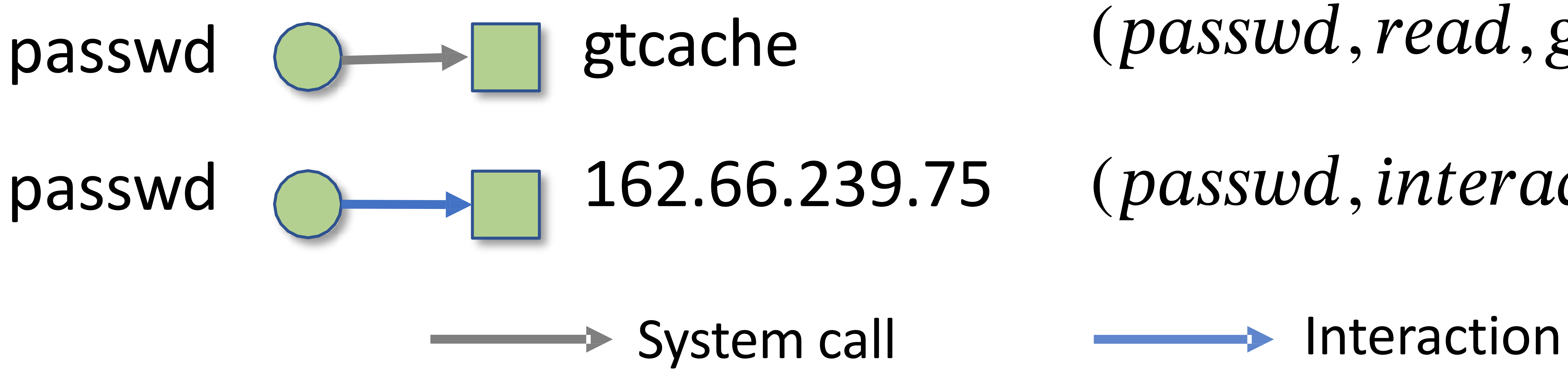


Bipartite Graph

Knowledge Graph Builder (cont.)

- System entities' side information is not encoded in a PG or BG
- However, side information can be inferred from the context in which system entities are used
- To incorporate high-order connectivity, we combine system entity contexts (side information) and interactions into a **knowledge graph**:

$$KG = \{(h,\ r,\ t) \mid h, t \in \{system\ entities\},\ r \in \{system\ call\ and\ interactions\}\}$$



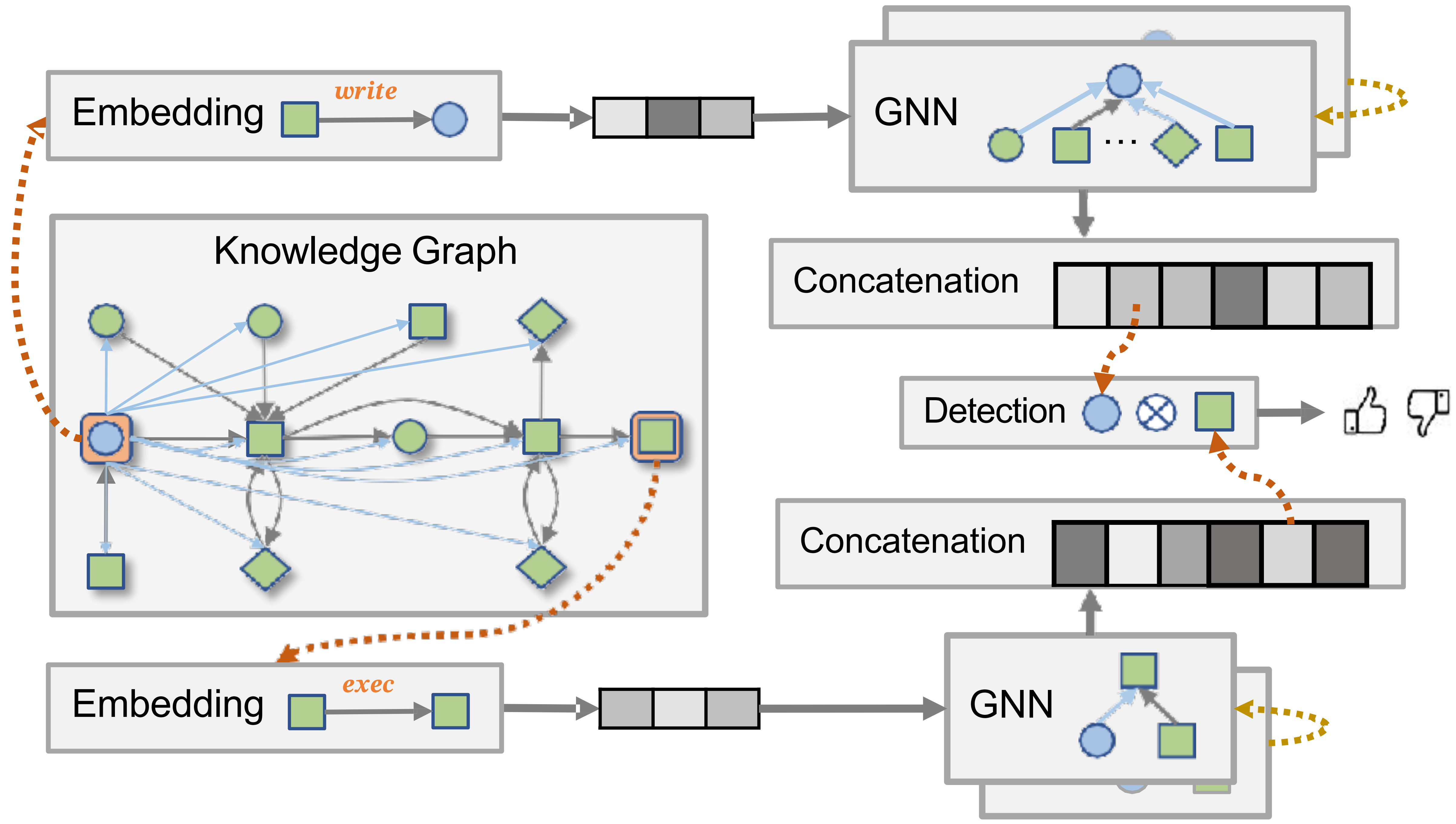
$$(passwd, read, gtcache)$$

$$(passwd, interact, 162.66.239.75)$$

Recommendation Model

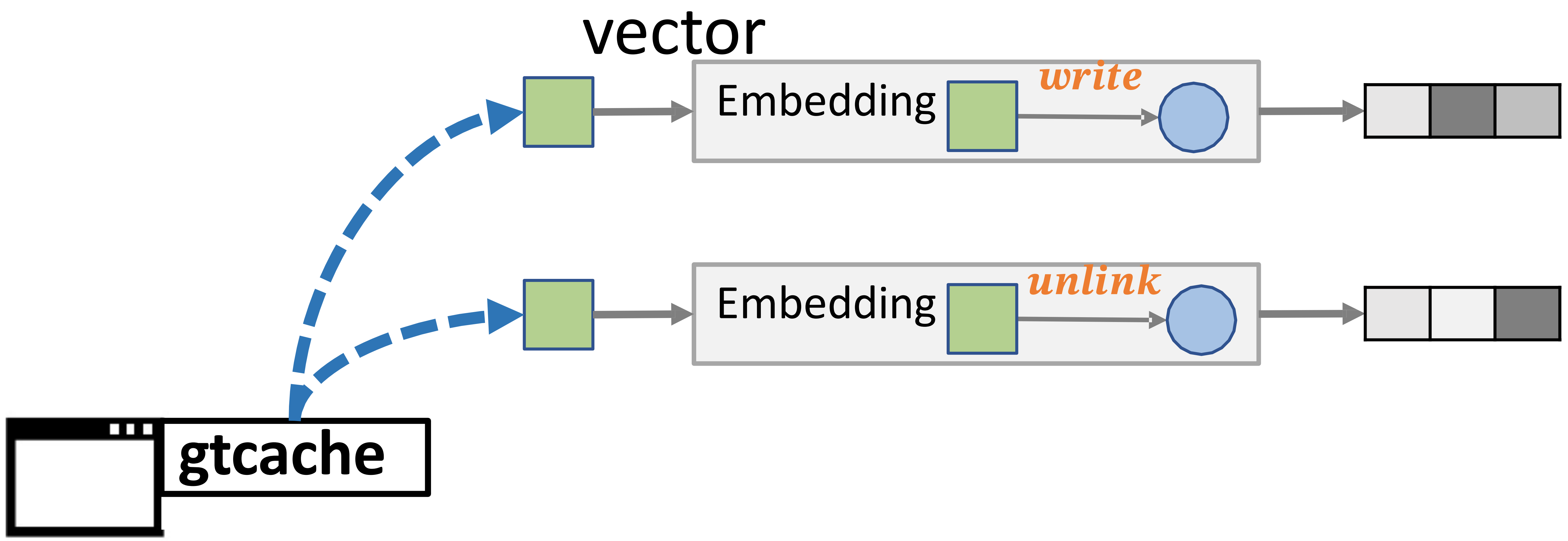
Key Idea: use **different-order** connectivities in a KG to model the **likelihood** of system entity interactions, identifying anomalous ones as cyber threats

- Model first-order connectivity to parameterize system entities as embeddings (i.e., vectors)
- Model higher-order connectivity by propagating embeddings from neighbors via GNNs
- Classify system entity interactions into normal and anomalous



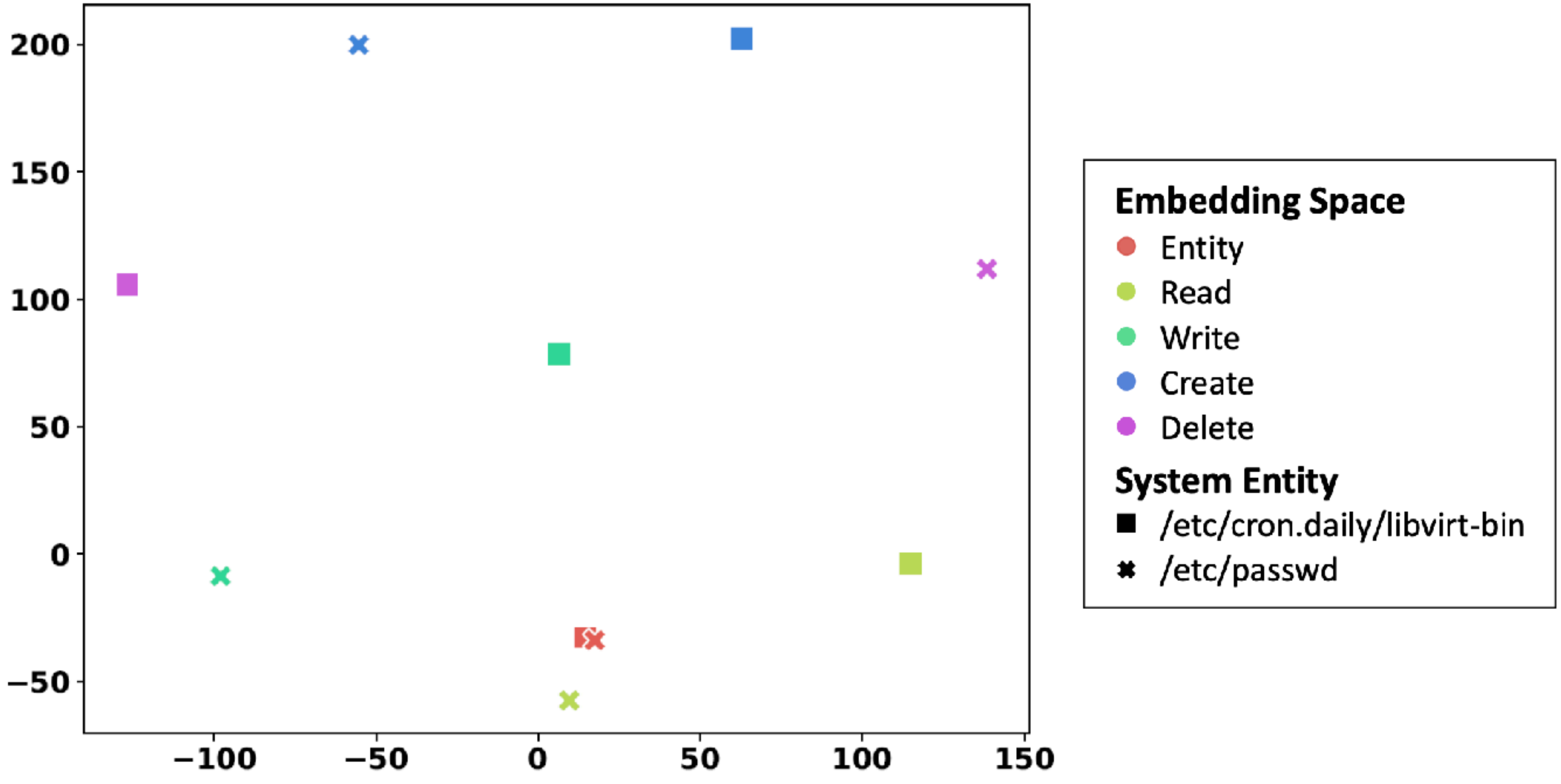
First-order Connectivity Modeling

- Model first-hop connections in a KG
  - System contexts (side information) decide the semantics of system entities
  - Use the KG embedding method (TransR): defines t = $h + r$ in $KG = \{ (h, r, t) \}$
  - Assign distinct semantics to the same entity conditioned on different relations



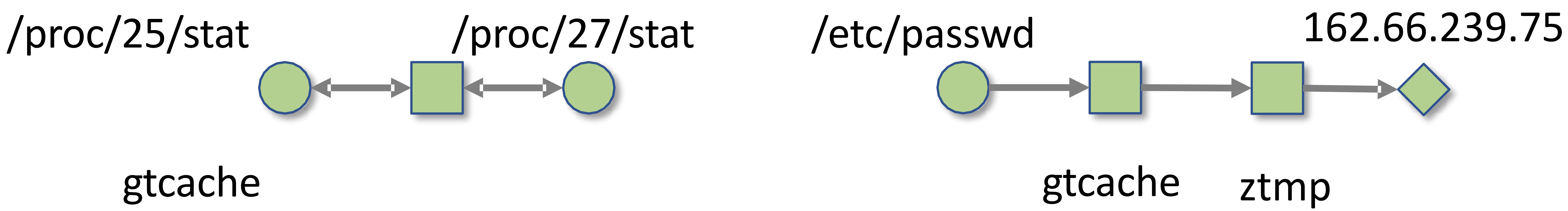
Two **different** embeddings

First-order Connectivity Modeling

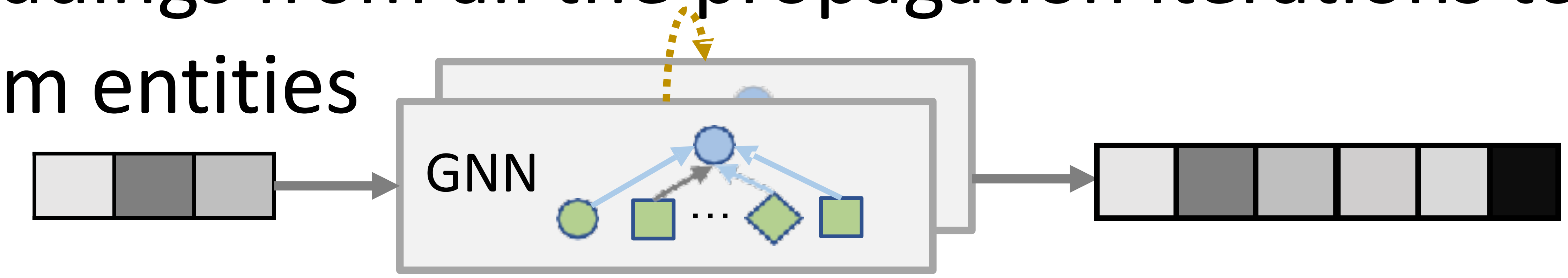


Higher-order Connectivity Modeling

- Model multi-hop paths in a KG
  - (1) Supplement similarities among system entities; (2) Exhibit how system entities influence each other

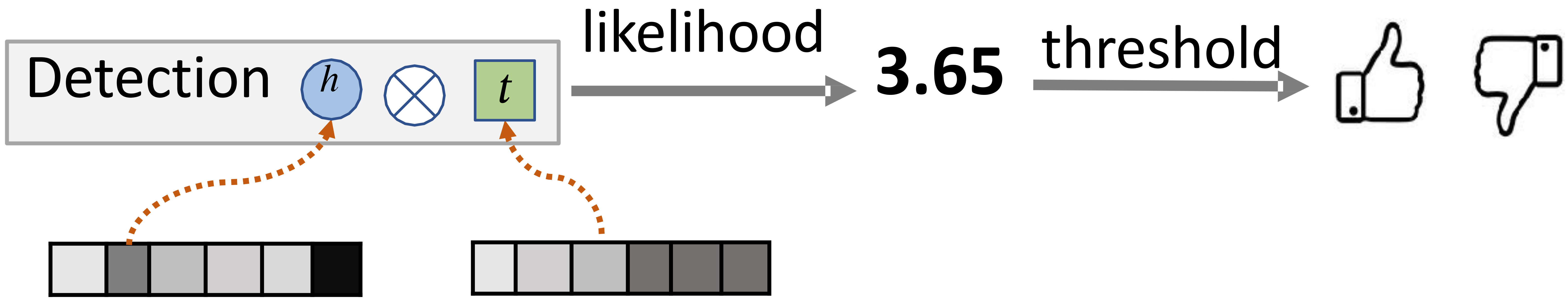


  - Adopt a graph neural network (GNN) to iteratively propagate embeddings along with multi-hop paths in a KG
  - Aggregate the embeddings from all the propagation iterations to form the final embeddings of system entities



Learning to Cyber Threat Analysis

- Given system entity interactions, we apply inner product on system entity embeddings to predict how likely a system entity would **not** interact with another entity.



- To keep up with evolving system entity interactions, we enable dynamic updates of the recommendation model with analyst feedback on detection signals.

Evaluation

- **Experimental datasets:**
  - **Six real-world cyber-attacks** simulated in a testbed environment:

    Configuration Leakage, Content Destruction, Cheating Student, Illegal Storage, Passwd Gzip Scp, and Passwd Reuse
  - **Four APT attacks** from the DARPA Transparent Computing (TC) dataset

    Extension Backdoor, Firefox Backdoor, Pine Backdoor, and Phishing Executable
- **Evaluation aspects:**
  - How **effective** is SHADEWATCHER as a threat detection system?
  - To what extend do first-order and high-order information **facilitate** analysis?
  - How **efficient** is SHADEWATCHER in deployment?

Effectiveness in Cyber Threat Detection

- Identify cyber threats based on system entity interactions in the DARPA TC dataset and Simulated dataset

Dataset	Ground Truth	True Positive	False Negative	False Positive Rate
DARPA TC Dataset	**68K** malicious & **8M** benign interactions	68,087	10	0.332%
Simulated Dataset	**39** malicious & **3M** benign interactions	37	2	0.137%

SHADEWATCHER distinguishes benign and malicious interactions with high accuracy

Study of Recommendation-guided Analysis

- Compare different KG embedding algorithms
- Study the importance of high-order information propagated by GNNs

KG Embedding	One-hot	TransE	TransH	TransR	TransR
GNN	**Yes**	**Yes**	**Yes**	**No**	**Yes**
AUC Value	0.966	0.971	0.974	0.763	**0.996**

SHADEWATCHER

SHADEWATCHER achieves the best performance (AUC):

- High-order information is **beneficial** to cyber threat analysis
- It is important to **distinguish** semantics under different relation contexts

System Efficiency

Measure the runtime overhead on the DARPA TC dataset at different phases: audit record **processing**, recommendation **training**, and cyber threat **testing**

Phase	Component	Mean
Processing	PG Construction	40.47 minutes
	Interaction Extraction	4.13 minutes
Training	System Entity Embedding	12.27 hours
	Information Propagation	6.45 hours
Testing	Interaction Classification	**8.16 seconds**

SHADEWATCHER pinpoints cyber threats from nearly a million interactions **within seconds**

Conclusion

- propose SHADEWATCHER:
  - Analyze cyber threats through recommendations on system entity interactions
  - Model a system entity’s preferences on its interactive entities

- Key insights:
  - Similar system entities share preferences on interactions
  - High-order information can better correlate similar system entities

Acknowledgments

- [ShadeWatcher] SHADEWATCHER: Recommendation-guided Cyber Threat Analysis using System Audit Records, J. Zeng, X. Wang, J. Liu, Y. Chen, Z. Liang, T.S. Chua, Z. Leong Chua, IEEE Security & Privacy, 2022.