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
Cyber Threats Are Everywhere

Microsoft confirms Lapsus$ hackers stole source code via ‘limited’ access

Cisco latest victim of Russian cyber attack using SolarWinds

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

1. ... **password**
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

[ShadeWatcher]
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

---

**Alice**

<table>
<thead>
<tr>
<th>Iron Man</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thor</td>
</tr>
<tr>
<td>The Avengers</td>
</tr>
<tr>
<td>Little Women</td>
</tr>
</tbody>
</table>
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

![Diagram showing relationships between movies and users](image-url)
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?
**Input:** Audit records collected by system auditing frameworks (e.g., Linux Audit)

**Output:** Detection signals for adversarial system entity interactions
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](image1)

![Bipartite Graph](image2)
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**:

\[
KG = \{(h, r, t) | h, t \in \{\text{system entities}\}, r \in \{\text{system call and interactions}\}\}
\]

- **passwd** → **gtcache** (\(\text{passwd, read, gtcache}\))
- **passwd** → **162.66.239.75** (\(\text{passwd, interact, 162.66.239.75}\))
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

![Diagram of connectivity modeling](image)
First-order Connectivity Modeling

![Graph showing first-order connectivity modeling with points representing different entities and system entities.]

- **Embedding Space**
  - Entity
  - Read
  - Write
  - Create
  - Delete

- **System Entity**
  - `/etc/cron.daily/libvirt-bin`
  - `/etc/passwd`

---

**Fall 1402**

**CE 815 - Causal Analysis**
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.

  ![Diagram](image)

  - Detection $h \times t$
  - Likelihood: 3.65
  - Threshold

- 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 extent 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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Ground Truth</th>
<th>True Positive</th>
<th>False Negative</th>
<th>False Positive Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARPA TC Dataset</td>
<td>68K malicious &amp; 8M benign interactions</td>
<td>68,087</td>
<td>10</td>
<td>0.332%</td>
</tr>
<tr>
<td>Simulated Dataset</td>
<td>39 malicious &amp; 3M benign interactions</td>
<td>37</td>
<td>2</td>
<td>0.137%</td>
</tr>
</tbody>
</table>

**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

<table>
<thead>
<tr>
<th>KG Embedding</th>
<th>One-hot</th>
<th>TransE</th>
<th>TransH</th>
<th>TransR</th>
<th>TransR</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNN</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>AUC Value</td>
<td>0.966</td>
<td>0.971</td>
<td>0.974</td>
<td>0.763</td>
<td>0.996</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Component</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing</td>
<td>PG Construction</td>
<td>40.47 minutes</td>
</tr>
<tr>
<td></td>
<td>Interaction Extraction</td>
<td>4.13 minutes</td>
</tr>
<tr>
<td>Training</td>
<td>System Entity Embedding</td>
<td>12.27 hours</td>
</tr>
<tr>
<td></td>
<td>Information Propagation</td>
<td>6.45 hours</td>
</tr>
<tr>
<td>Testing</td>
<td>Interaction Classification</td>
<td><strong>8.16 seconds</strong></td>
</tr>
</tbody>
</table>

**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