CE 815 – Secure Software Systems

Causal Analysis (Atlas)

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.
Attack Investigation Challenges

• Failing to address these challenges lead to attack investigation false positives and false negatives

1. “Finding needle in a haystack”
   Only small number of events are relevant to an attack

2. “Connecting the dots”
   Construct end-to-end attack story out of attack-related logs, sometimes across multiple hosts
Observation

• Attack steps can be summarized as a temporal sequence of words

1. firefox.exe
   *Write*
   contract.doc

2. winword.exe
   *Read*
   contract.doc

3. winword.exe
   *Connect*
   1.2.3.4

4. winword.exe
   *Write*
   template.hta

5. powershell.exe
   *Fork*
   cmd.exe

6. cmd.exe
   *Fork*
   mahta.exe

7. mahta.exe
   *Read*
   template.hta

8. powershell.exe
   *Write*
   maintenance.vbs

9. powershell.exe
   *Execute*
   maintenance.vbs

10. powershell.exe
    *Write*
    backdoor.exe

11. svchost.exe
    *Read*
    backdoor.exe

12. svchost.exe
    *Fork*
    backdoor

13. backdoor
    *Read*
    PDF files

14. backdoor
    *Connect*
    5.6.7.8

• Attack steps can be summarized as a concise attack subgraph
Design Challenges 1

• The goal is to separate benign from malicious activities and generalize sequence extraction across various audit log types.

• Two main challenges:
  • Audit logs contain a vast number of unique entities, leading to many different sequences of arbitrary lengths.
  • Similar attack patterns can result in different sequences, but with similar contexts, which complicates model learning and can cause issues like vanishing or exploding gradients.

• Addressed by:
  • Using a custom graph-optimization to reduce complexity and obtain shorter, relevant sequences.
  • Implementing a novel technique for extracting and learning sequences that accurately represent attack patterns.
Design Challenges 2

- Learning from sequences for attack investigation, akin to "finding needles in a haystack."
- Monitoring produces imbalanced datasets with few attack sequences (needles) and many non-attack sequences (haystack).
- Imbalanced sequences significantly hinder the learning process, with models biased towards non-attack sequences, missing some attacks.
- Combat with under-sampling of non-attack sequences and over-sampling of attack sequences.
- This creates a balanced ratio between attack and non-attack sequences, facilitating more effective model learning.
Design Challenges 3

- Querying arbitrary sequences, but generating these sequences is ad-hoc and might not capture all attack entities.
- Investigators often need to find many sequences with potential attack entities, which is inefficient.
- To improve this, ATLAS has an attack investigation phase that:
  - Analyzes entities in audit logs.
  - Identifies attack entities that, when paired with an attack symptom entity, form an attack sequence.
  - More accurately and efficiently recovers attack entities to build the attack narrative.
Audit Log Pre-processing

- Build an optimized causal graph that reduces complexity without losing important semantics. Which leads to shorter sequences, enhancing learning efficacy and precision.

- ATLAS's optimization techniques include:
  - Removing nodes and edges not connected to attack nodes or the attack symptom node.
  - Dropping duplicate edges, keeping only the first occurrence of an action between entities.
  - Combining nodes and edges of identical event types, assigning the earliest timestamp to the new edge.

- This optimization does not disrupt the detection of attack patterns despite potentially altering the temporal order of events.

- The process results in an average 81.81% reduction in the number of entities in the causal graph.
Figure 2: Illustration of causal graph, neighborhood graph, events, and sequences.
Audit Log Pre-processing

Figure 4: Illustration of graph optimization in ATLAS. P: Process, S: Session, A: IP Address, D: Domain name.
Sequence Construction and Learning

- Identify temporally ordered events for attack entities from a causal graph and creates subsets of attack entities, each with two or more entities, to analyze combinations.
  - The number of subsets is calculated combinatorially and can be exponentially large with the number of entities but is usually manageable as attackers limit their footprint.
- ATLAS extracts neighborhood graphs for each attack entity to identify all causally related entities and then orders attack events by timestamps within these graphs.
  - Events are considered attacks if they involve attack entities as sources or destinations.
- Finally, ATLAS labels a series of timestamp-ordered events as an attack sequence if it contains only attack events and includes all attack events for a given subset of entities.
Sequence Construction and Learning

• Non-attack sequences are challenging to identify due to the vast number of non-attack entities.
• ATLAS does not learn benign activities but distinguishes between malicious and non-malicious activities.
• It adds a non-attack entity to attack subsets to extract non-attack sequences, allowing the model to learn the deviations.
• ATLAS extracts non-attack sequences by following the same steps used for attack sequences.
• A sequence is labeled non-attack if it doesn't match any attack sequence pattern.
Figure 5: (Middle) An example causal graph to illustrate sequence construction process. (Left) Attack sequence extraction steps. (Right) Non-attack sequence extraction steps.
Sequence Lemmatization

- ATLAS employs lemmatization to convert sequences into generalized text for semantic interpretation, similar to NLP practices.

- This retains original sequence semantics, aiding in model learning.

- ATLAS's vocabulary of 30 words abstracts entities and actions in sequences into four types: process, file, network, and actions.

- It parses sequences, lemmatizes entities, and maps them to vocabulary, like transforming:
  - `<system/process/malicious.exe read /user/secret.pdf>` to `<system_process read user_file>`.

- Post-lemmatization, sequences resemble "sentence-like" structures that maintain the semantics of generalized patterns.
## Table 1: Abstracted vocabulary set for lemmatization

<table>
<thead>
<tr>
<th>Type</th>
<th>Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>process</td>
<td>system_process, lib_process, programs_process, user_process</td>
</tr>
<tr>
<td>file</td>
<td>system_file, lib_file, programs_file, user_file, combined_files</td>
</tr>
<tr>
<td>network</td>
<td>ip_address, domain, url, connection, session</td>
</tr>
<tr>
<td>actions</td>
<td>read, write, delete, execute, invoke, fork, request, refer, bind receive, send, connect, ip_connect, session_connect, resolve</td>
</tr>
</tbody>
</table>
Selective Sequence Sampling

- Imbalance example: average attack entities 61 vs. non-attack entities 21,000.
- Training on such imbalanced data risks bias towards the majority class or failure to learn about the minority class.
- ATLAS balances the dataset by undersampling non-attack sequences to a similarity threshold.
- It then oversamples attack sequences through mutation to match the number of non-attack sequences.
- Simple duplication or random removal of sequences can lead to overfitting or missing patterns.
- To avoid this, employs specialized undersampling and oversampling mechanisms.
Embedding and Learning

- Applies word2vec and other embedding techniques to capture semantic relationships between words.
- Compiles a corpus of lemmatized attack and non-attack sequences from audit logs for training word embeddings.
- Employs LSTM networks for learning from sequences, which are effective in various NLP tasks.
Implementation

- Built using Python version 3.7.7.
- Comprises approximately 3,000 lines of code for all components.
- Processes Windows security events with Sysmon for file operations and network connections.
- Handles Firefox logs to track visited webpages.
- Utilizes TShark for capturing DNS logs.
- Employs the LSTM model from the Keras library with TensorFlow as the back-end.
Dataset

- Implemented ten attacks based on real-world APT campaign reports to generate audit logs.
- Created a controlled testbed environment for generating these logs.
- Construction of Benign System Events:
  - Emulated diverse normal user activities alongside attack execution.
  - Manually generated benign activities such as web browsing, email reading, and file downloading.
  - Scheduled benign activities randomly within an 8-hour daytime window.
- Details of Attack Implementation and Emulation:
  - On average, generated 20,088 unique entities with 249K events per attack.
    - Entity 28 (attack) 20K (non-attack)
    - Event 17K (attack) 275K (non-attack)
Table 2: Overview of implemented APT attacks for ATLAS evaluation.

<table>
<thead>
<tr>
<th>Attack ID</th>
<th>APT Campaign</th>
<th>Exploiting CVE by attack</th>
<th>Attack Features†</th>
<th>Size (MB)</th>
<th>Log Type (%)</th>
<th>Total</th>
<th># entity</th>
<th># event</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>System</td>
<td>Web</td>
<td>DNS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-1</td>
<td>Strategic web compromise [17]</td>
<td>2015-5122</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>381</td>
<td>97.11%</td>
<td>2.24%</td>
</tr>
<tr>
<td>S-2</td>
<td>Malvertising dominate [22]</td>
<td>2015-3105</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>990</td>
<td>98.58%</td>
<td>1.09%</td>
</tr>
<tr>
<td>S-3</td>
<td>Spam campaign [39]</td>
<td>2017-11882</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>521</td>
<td>96.82%</td>
<td>2.43%</td>
</tr>
<tr>
<td>S-4</td>
<td>Pony campaign [18]</td>
<td>2017-0199</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>448</td>
<td>97.08%</td>
<td>2.24%</td>
</tr>
<tr>
<td>M-1</td>
<td>Strategic web compromise [17]</td>
<td>2015-5122</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>851.3</td>
<td>96.89%</td>
<td>1.32%</td>
</tr>
<tr>
<td>M-2</td>
<td>Targeted GOV phishing [34]</td>
<td>2015-5119</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>819.9</td>
<td>97.39%</td>
<td>1.36%</td>
</tr>
<tr>
<td>M-3</td>
<td>Malvertising dominate [22]</td>
<td>2015-3105</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>496.7</td>
<td>99.11%</td>
<td>0.52%</td>
</tr>
<tr>
<td>M-4</td>
<td>Monero miner by Rig [28]</td>
<td>2018-8174</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>653.6</td>
<td>98.14%</td>
<td>1.24%</td>
</tr>
<tr>
<td>M-5</td>
<td>Pony campaign [18]</td>
<td>2017-0199</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>878</td>
<td>98.14%</td>
<td>1.24%</td>
</tr>
<tr>
<td>M-6</td>
<td>Spam campaign [39]</td>
<td>2017-11882</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>725</td>
<td>98.31%</td>
<td>0.96%</td>
</tr>
<tr>
<td>Avg.</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>676.5</td>
<td>97.76%</td>
<td>1.46%</td>
</tr>
</tbody>
</table>

Table 3: Ground-truth information of each implemented attack, including the number of entities, events, sequences and balanced sequences.

<table>
<thead>
<tr>
<th>Attack ID</th>
<th>#Attack Entity</th>
<th>#Non-attack Entity</th>
<th>#Attack Event</th>
<th>#Non-attack Event</th>
<th>#Attack Seq.</th>
<th>#Non-attack Seq.</th>
<th>#Balanced Seq.*</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-1</td>
<td>22</td>
<td>7,445</td>
<td>4,598</td>
<td>90,467</td>
<td>42</td>
<td>14,243</td>
<td>1,388</td>
</tr>
<tr>
<td>S-2</td>
<td>12</td>
<td>34,008</td>
<td>15,073</td>
<td>382,879</td>
<td>43</td>
<td>13,388</td>
<td>1,386</td>
</tr>
<tr>
<td>S-3</td>
<td>26</td>
<td>8,972</td>
<td>5,165</td>
<td>123,152</td>
<td>21</td>
<td>8,600</td>
<td>2,598</td>
</tr>
<tr>
<td>S-4</td>
<td>21</td>
<td>13,016</td>
<td>18,062</td>
<td>107,551</td>
<td>32</td>
<td>12,238</td>
<td>1,244</td>
</tr>
<tr>
<td>M-1</td>
<td>28</td>
<td>17,565</td>
<td>8,168</td>
<td>243,507</td>
<td>83</td>
<td>26,764</td>
<td>2,682</td>
</tr>
<tr>
<td>M-2</td>
<td>36</td>
<td>24,450</td>
<td>34,956</td>
<td>249,365</td>
<td>82</td>
<td>27,041</td>
<td>2,748</td>
</tr>
<tr>
<td>M-3</td>
<td>36</td>
<td>24,424</td>
<td>34,979</td>
<td>299,157</td>
<td>81</td>
<td>27,525</td>
<td>2,710</td>
</tr>
<tr>
<td>M-4</td>
<td>28</td>
<td>15,378</td>
<td>8,236</td>
<td>250,512</td>
<td>79</td>
<td>27,076</td>
<td>2,746</td>
</tr>
<tr>
<td>M-5</td>
<td>30</td>
<td>35,671</td>
<td>34,175</td>
<td>667,337</td>
<td>78</td>
<td>25,915</td>
<td>2,540</td>
</tr>
<tr>
<td>M-6</td>
<td>42</td>
<td>19,580</td>
<td>9,994</td>
<td>344,034</td>
<td>70</td>
<td>23,473</td>
<td>2,598</td>
</tr>
<tr>
<td>Avg.</td>
<td>28</td>
<td>20,051</td>
<td>17,341</td>
<td>275,796</td>
<td>61</td>
<td>20,626</td>
<td>2,264</td>
</tr>
</tbody>
</table>

* The sampled number of attack and non-attack sequences are identical.
### Table 4: Entity-based and event-based investigation results.

<table>
<thead>
<tr>
<th>ID</th>
<th>Symptom entity</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Precision %</th>
<th>Recall %</th>
<th>F1-score %</th>
<th>TP</th>
<th>TN</th>
<th># Precision %</th>
<th># Recall %</th>
<th>F1-score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-1</td>
<td>malicious host</td>
<td>22</td>
<td>7,445</td>
<td>0</td>
<td>0</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>4,598</td>
<td>90,467</td>
<td>0</td>
<td>0</td>
<td>100.00%</td>
</tr>
<tr>
<td>S-2</td>
<td>leaked file</td>
<td>12</td>
<td>34,008</td>
<td>2</td>
<td>0</td>
<td>85.71%</td>
<td>100.00%</td>
<td>92.31%</td>
<td>15,073</td>
<td>382,876</td>
<td>3</td>
<td>0</td>
<td>100.00%</td>
</tr>
<tr>
<td>S-3</td>
<td>malicious host</td>
<td>24</td>
<td>8,972</td>
<td>2</td>
<td>0</td>
<td>100.00%</td>
<td>92.31%</td>
<td>96.00%</td>
<td>5,155</td>
<td>123,152</td>
<td>0</td>
<td>10</td>
<td>100.00%</td>
</tr>
<tr>
<td>S-4</td>
<td>leaked file</td>
<td>21</td>
<td>13,011</td>
<td>5</td>
<td>0</td>
<td>80.77%</td>
<td>100.00%</td>
<td>89.36%</td>
<td>18,062</td>
<td>107,506</td>
<td>45</td>
<td>0</td>
<td>99.75%</td>
</tr>
<tr>
<td>M-1</td>
<td>leaked file</td>
<td>28</td>
<td>17,562</td>
<td>3</td>
<td>0</td>
<td>90.32%</td>
<td>100.00%</td>
<td>94.92%</td>
<td>8,168</td>
<td>243,504</td>
<td>3</td>
<td>0</td>
<td>99.96%</td>
</tr>
<tr>
<td>M-2</td>
<td>leaked file</td>
<td>36</td>
<td>24,445</td>
<td>5</td>
<td>0</td>
<td>87.80%</td>
<td>100.00%</td>
<td>93.51%</td>
<td>34,956</td>
<td>249,316</td>
<td>49</td>
<td>0</td>
<td>99.86%</td>
</tr>
<tr>
<td>M-3</td>
<td>malicious file</td>
<td>35</td>
<td>24,423</td>
<td>1</td>
<td>1</td>
<td>97.22%</td>
<td>97.22%</td>
<td>97.22%</td>
<td>34,978</td>
<td>299,147</td>
<td>10</td>
<td>1</td>
<td>99.97%</td>
</tr>
<tr>
<td>M-4</td>
<td>malicious file</td>
<td>24</td>
<td>15,378</td>
<td>0</td>
<td>4</td>
<td>100.00%</td>
<td>85.71%</td>
<td>92.31%</td>
<td>8,161</td>
<td>250,512</td>
<td>0</td>
<td>75</td>
<td>100.00%</td>
</tr>
<tr>
<td>M-5</td>
<td>malicious host</td>
<td>30</td>
<td>35,665</td>
<td>6</td>
<td>0</td>
<td>83.33%</td>
<td>100.00%</td>
<td>90.91%</td>
<td>34,175</td>
<td>667,329</td>
<td>8</td>
<td>0</td>
<td>99.98%</td>
</tr>
<tr>
<td>M-6</td>
<td>malicious host</td>
<td>41</td>
<td>19,573</td>
<td>7</td>
<td>1</td>
<td>85.42%</td>
<td>97.62%</td>
<td>91.11%</td>
<td>9,993</td>
<td>343,959</td>
<td>75</td>
<td>1</td>
<td>99.26%</td>
</tr>
<tr>
<td><strong>Avg.</strong></td>
<td></td>
<td><strong>27</strong></td>
<td><strong>20,048</strong></td>
<td><strong>3</strong></td>
<td><strong>1</strong></td>
<td><strong>91.06%</strong></td>
<td><strong>97.29%</strong></td>
<td><strong>93.76%</strong></td>
<td><strong>17,332</strong></td>
<td><strong>275,777</strong></td>
<td><strong>19</strong></td>
<td><strong>9</strong></td>
<td><strong>99.88%</strong></td>
</tr>
</tbody>
</table>

TP and TN stands for correctly reported attack and non-attack (normal) entities/events. FP and FN stands for incorrectly labeled attack and non-attack (normal) entities/events.
Figure 8: Effectiveness of causal graph optimization of given audit logs for attack investigation. The percentages on the bars show the percentage of the logs reduction.
Attack Story Recovery
Conclusion

• ATLAS is a framework for identifying and reconstructing cyber attack stories from audit logs.

• It uses causality analysis, natural language processing, and machine learning techniques.

• The approach models and recognizes high-level attack patterns via sequence-based analysis.

• Evaluation on 10 real-world APT scenarios demonstrated high precision and efficiency in recovery of attack steps.
Acknowledgments