CE 815 – Secure Software Systems

Causal Analysis (Atlas)

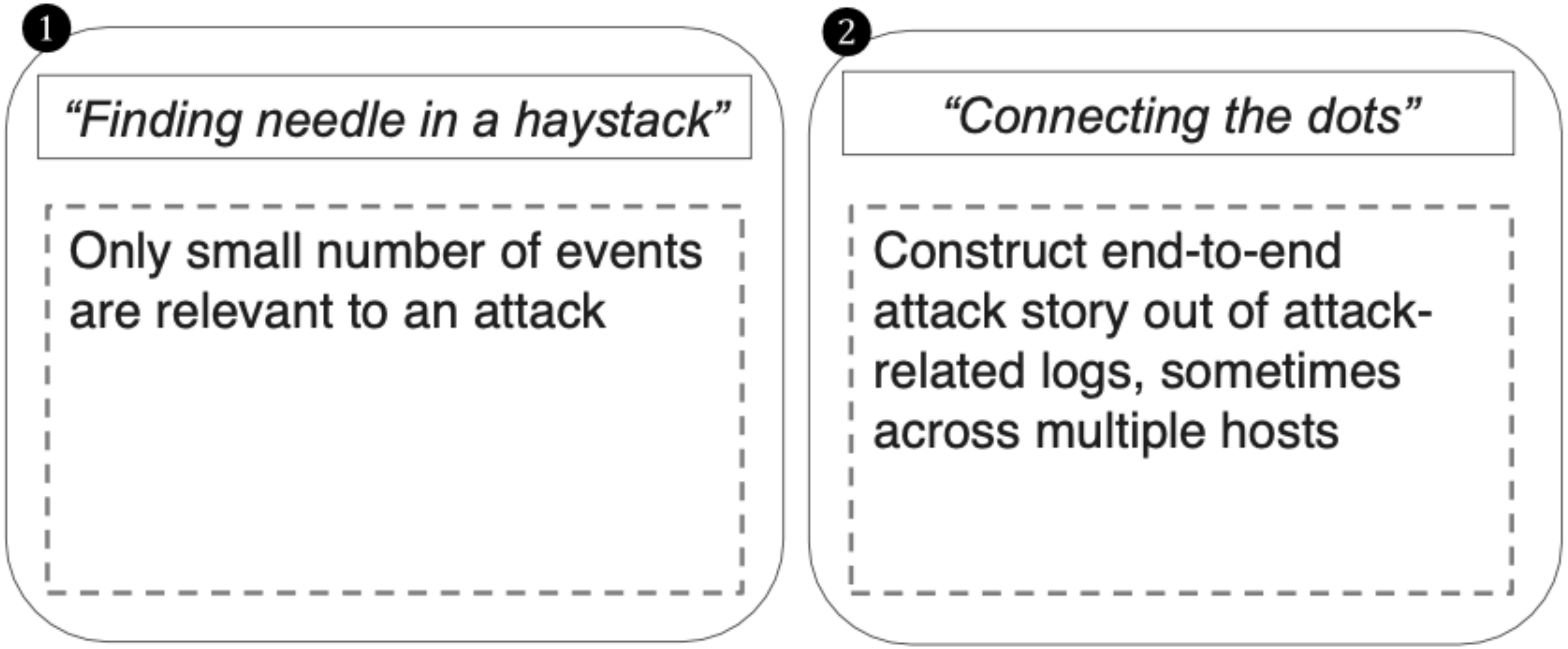
Mehdi Kharrazi
Department of Computer Engineering
Sharif University of Technology

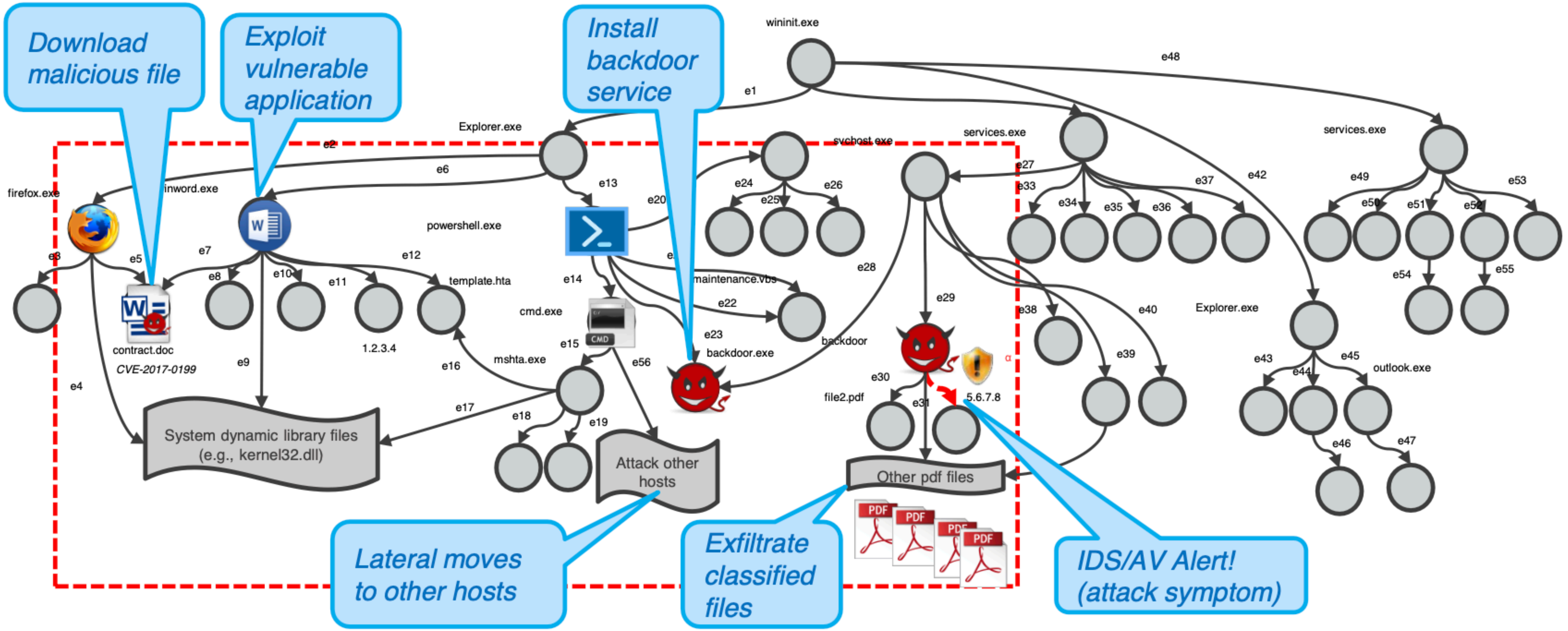
Acknowledgments: Some of the slides are fully or partially obtained from other sources. A reference is noted on the bottom of each slide, when the content is fully obtained from another source. Otherwise a full list of references is provided on the last slide.

ATLAS: A Sequence-based Learning Approach for Attack Investigation, A. Alsaheel, Y. Nan, S. Ma, L. Yu, G. Walkup, Z. Berkay Celik, X. Zhang, and D. Xu, Usenix Security 2021.

Attack Investigation Challenges

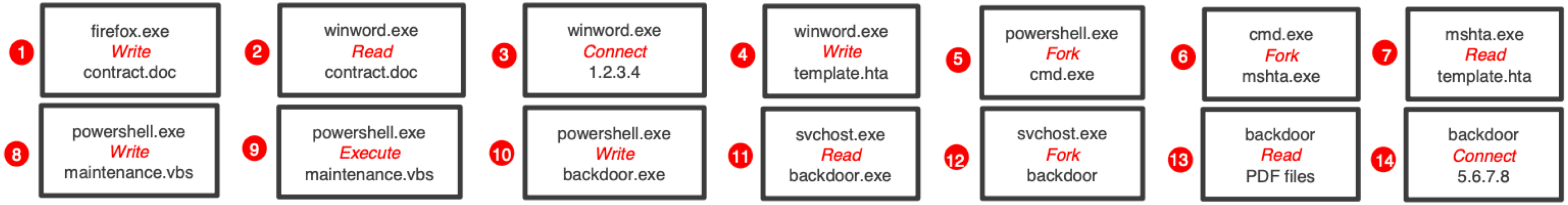
- Failing to address these challenges lead to attack investigation false positives and false negatives



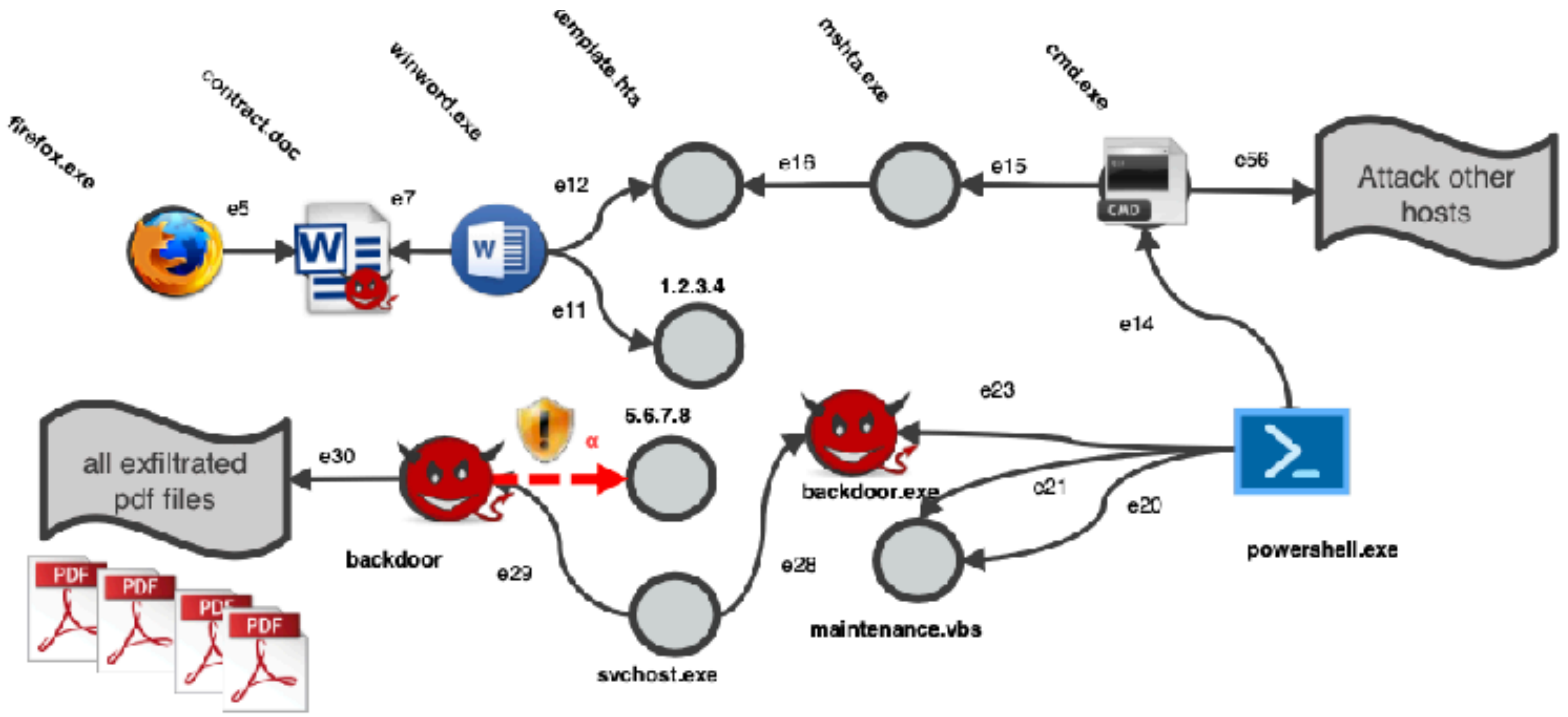

Download malicious file
Exploit vulnerable application
Install backdoor service
Lateral moves to other hosts
Exfiltrate classified files
IDS/AV Alert! (attack symptom)
wininit.exe
Explorer.exe
firefox.exe
winword.exe
powershell.exe
svchost.exe
services.exe
contract.doc
CVE-2017-0199
1.2.3.4
template.hta
cmd.exe
mshta.exe
maintenance.vbs
backdoor.exe
backdoor
file2.pdf
5.6.7.8
outlook.exe
System dynamic library files (e.g., kernel32.dll)
Attack other hosts
Other pdf files

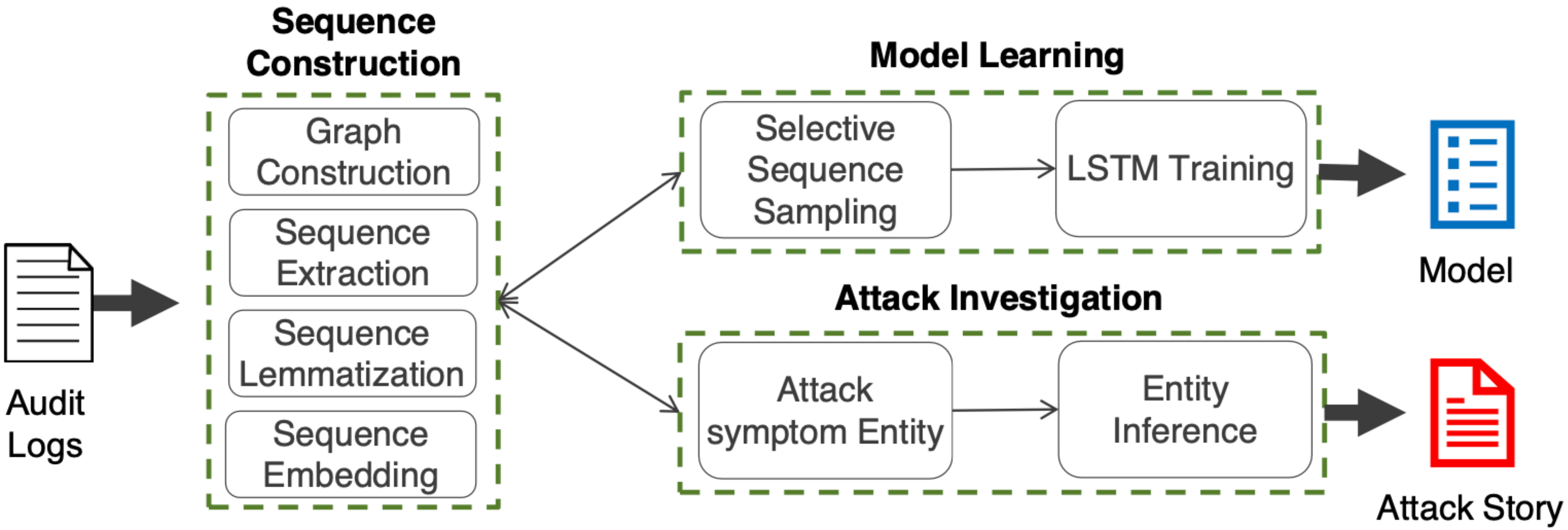
Observation

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



- Attack steps can be summarized as a concise attack subgraph





Sequence Construction
Graph Construction
Sequence Extraction
Sequence Lemmatization
Sequence Embedding
Audit Logs
Model Learning
Selective Sequence Sampling
LSTM Training
Model
Attack Investigation
Attack symptom Entity
Entity Inference
Attack Story


a
Sequence-based Model Learning
Non-attack entity
Attack entity
1
Graph Construction
Audit Logs
2
Sequence Construction
Sequence Lemmatization
Process write File; Process Execute File; ….;
3
Selective Sequence Sampling
Normal Seq.
Attack Seq.
Balanced Sequences
4
Sequence embedding
Process write File; Process Execute File; ….;
Sequence to Numerical vector representation
5
LSTM Training
Non-attack Seq.
Attack Seq.
Model
b
Attack Investigation
Attack Symptom Entity (e.g., malicious host)
Sequence Construction and Lemmatization
Sequence embedding
Model
Attack/non-attack entity inference
Attack Story

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.

Sequence Extraction

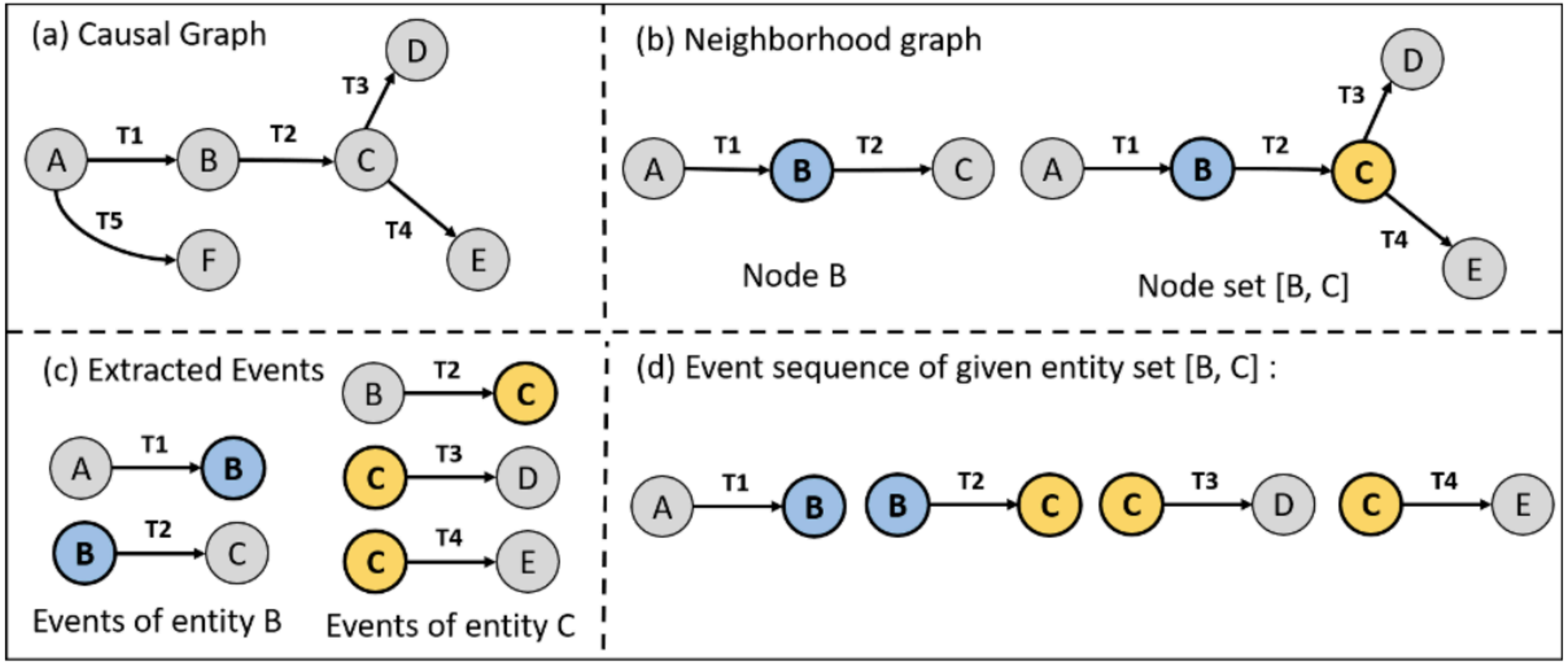


Figure 2: Illustration of causal graph, neighborhood graph, events, and sequences.

Audit Log Pre-processing

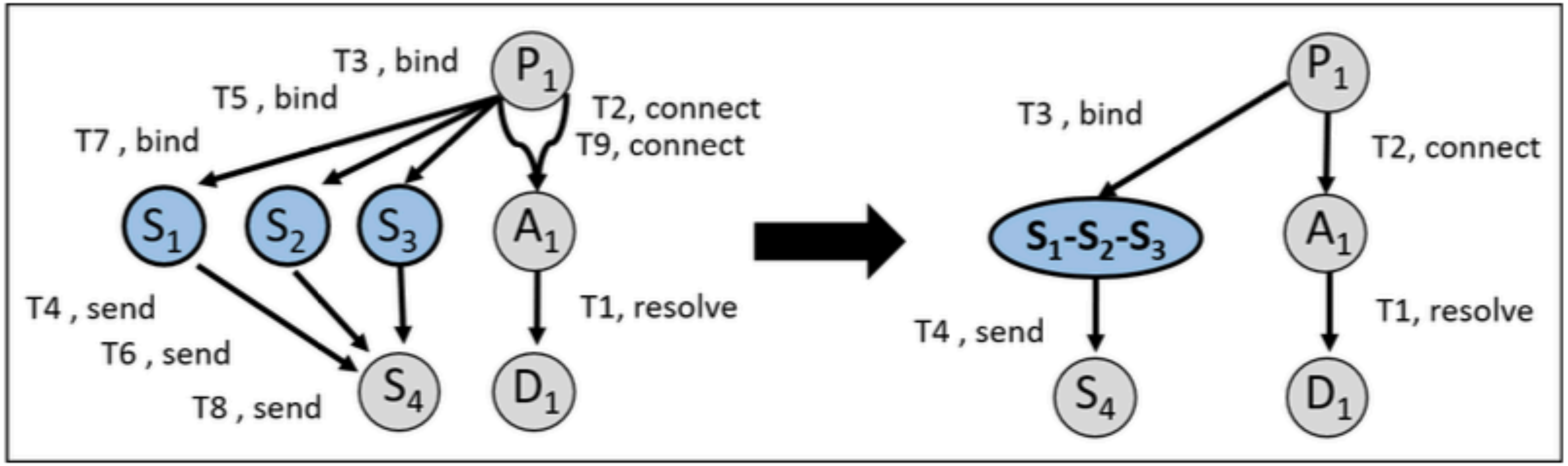


Figure 4: Illustration of graph optimization in ATLAS. `P`: Process, `S`: Session, `A`: IP Address, `D`: Domain name.

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.

Attack and Non-attack Sequence Extraction

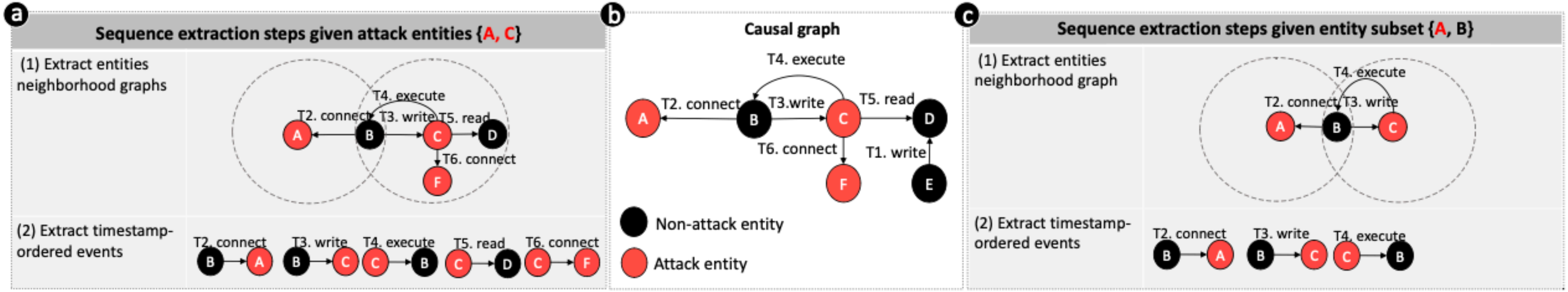


Figure 5: (Middle) An example causal graph to illustrate sequence construction process. (Left) Attack sequence extraction steps. (Right) Non-attack sequence extraction steps.

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.

Sequence Lemmatization

Table 1: Abstracted vocabulary set for lemmatization

Type	Vocabulary
process	system_process, lib_process, programs_process, user_process
file	system_file, lib_file, programs_file, user_file, combined_files
network	ip_address, domain, url, connection, session
actions	read, write, delete, execute, invoke, fork, request, refer, bind receive, send, connect, ip_connect, session_connect, resolve

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.

Attack ID	APT Campaign	Exploiting CVE by attack	Attack Features†							Size (MB)	Log Type (%)			Total	
			PL	**PA**	**INJ**	**IG**	**BD**	**LM**	**DE**		System	Web	DNS	# entity	# event
S-1	Strategic web compromise [17]	2015-5122	√		√	√	√		√	381	97.11%	2.24%	0.65%	7,468	95.0K
S-2	Malvertising dominate [22]	2015-3105	√		√	√	√		√	990	98.58%	1.09%	0.33%	34,021	397.9K
S-3	Spam campaign [39]	2017-11882		√	√	√	√		√	521	96.82%	2.43%	0.75%	8,998	128.3K
S-4	Pony campaign [18]	2017-0199		√	√	√	√		√	448	97.08%	2.24%	0.68%	13,037	125.6K
M-1	Strategic web compromise [17]	2015-5122	√		√	√	√	√	√	851.3	96.89%	1.32%	1.32%	17,599	251.6K
M-2	Targeted GOV phishing [34]	2015-5119	√		√	√	√	√	√	819.9	97.39%	1.36%	1.25%	24,496	284.3K
M-3	Malvertising dominate [22]	2015-3105	√		√	√	√	√	√	496.7	99.11%	0.52%	0.37%	24,481	334.1K
M-4	Monero miner by Rig [28]	2018-8174		√	√	√	√	√	√	653.6	98.14%	1.24%	0.62%	15,409	258.7K
M-5	Pony campaign [18]	2017-0199	√		√	√	√	√	√	878	98.14%	1.24%	0.62%	35,709	258.7K
M-6	Spam campaign [39]	2017-11882		√	√	√	√	√	√	725	98.31%	0.96%	0.73%	19,666	354.0K
Avg.	-	-	-	-	-	-	-	-	-	**676.5**	**97.76%**	**1.46%**	**0.73%**	**20,088**	**249K**

† **PL**: Phishing email link. **PA** : Phishing email attachment. **INJ**: Injection. **IG**: information gathering. **BD**: backdoor. **LM**: Lateral movement. **DE**: Data ex-filtration.

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

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

* The sampled number of attack and non-attack sequences are identical.

Table 4: Entity-based and event-based investigation results.

ID	Symptom entity	Entity-based Investigation Results							Event-based Investigation Results						
		TP	TN	FP	FN	Precision %	Recall %	F1-score %	TP	TN	FP	FN	# Precision %	# Recall %	F1-score %
S-1	malicious host	22	7,445	0	0	100.00%	100.00%	100.00%	4,598	90,467	0	0	100.00%	100.00%	100.00%
S-2	leaked file	12	34,008	2	0	85.71%	100.00%	92.31%	15,073	382,876	3	0	99.98%	100.00%	99.99%
S-3	malicious host	24	8,972	0	2	100.00%	92.31%	96.00%	5,155	123,152	0	10	100.00%	99.81%	99.90%
S-4	leaked file	21	13,011	5	0	80.77%	100.00%	89.36%	18,062	107,506	45	0	99.75%	100.00%	99.88%
M-1	leaked file	28	17,562	3	0	90.32%	100.00%	94.92%	8,168	243,504	3	0	99.96%	100.00%	99.98%
M-2	leaked file	36	24,445	5	0	87.80%	100.00%	93.51%	34,956	249,316	49	0	99.86%	100.00%	99.93%
M-3	malicious file	35	24,423	1	1	97.22%	97.22%	97.22%	34,978	299,147	10	1	99.97%	100.00%	99.98%
M-4	malicious file	24	15,378	0	4	100.00%	85.71%	92.31%	8,161	250,512	0	75	100.00%	99.09%	99.54%
M-5	malicious host	30	35,665	6	0	83.33%	100.00%	90.91%	34,175	667,329	8	0	99.98%	100.00%	99.99%
M-6	malicious host	41	19,573	7	1	85.42%	97.62%	91.11%	9,993	343,959	75	1	99.26%	99.99%	99.62%
Avg.	-	**27**	**20,048**	**3**	**1**	**91.06%**	**97.29%**	**93.76%**	**17,332**	**275,777**	**19**	**9**	**99.88%**	**99.89%**	**99.88%**

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.

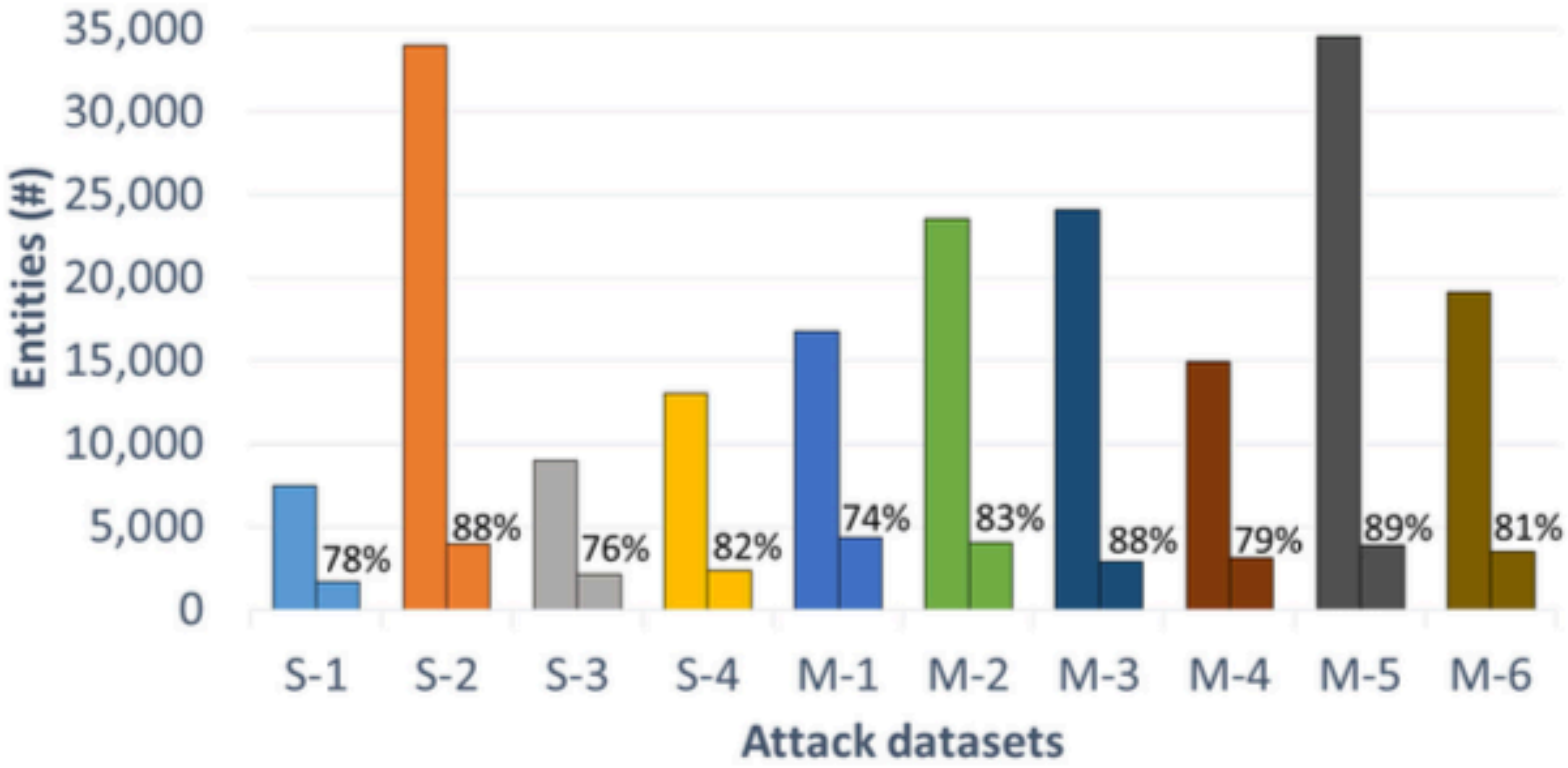
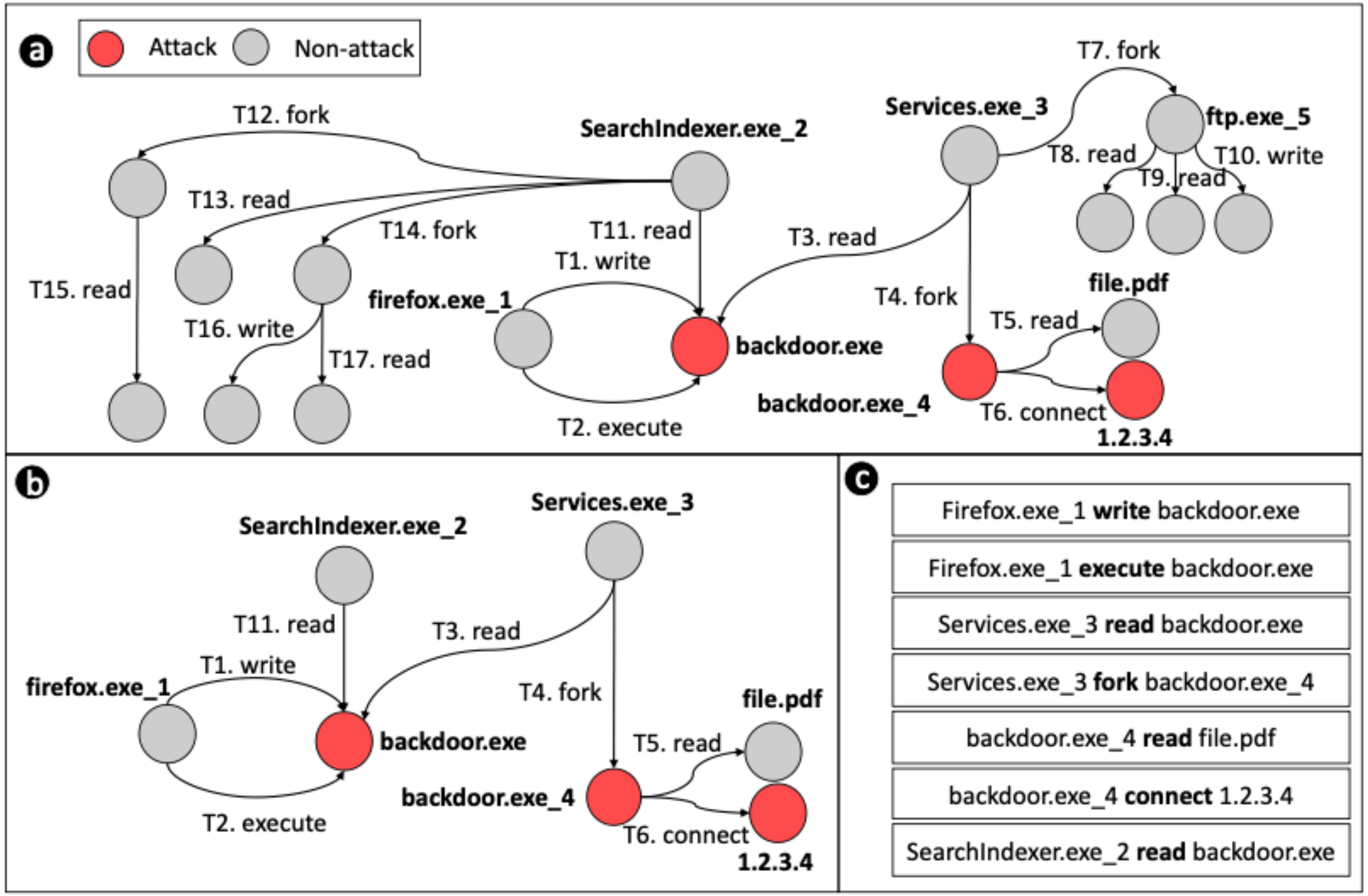


Figure 8: Effectiveness of causal graph optimization of given audit logs for attack investigation. The percentages on the bars show the percentage of the logs reduction.

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

- [Atlas] ATLAS: A Sequence-based Learning Approach for Attack Investigation, A. Alsaheel, Y. Nan, S. Ma, L. Yu, G. Walkup, Z. Berkay Celik, X. Zhang, and D. Xu, Usenix Security 2021.