Poster: Design of an Anomaly-based Threat Detection & Explication System

Robert Luh
Josef Ressel Center TARGET,
St. Pölten UAS, Austria
& De Montfort University Leicester
Leicester, UK

Sebastian Schrittwieser
Josef Ressel Center TARGET
St. Pölten UAS
St. Pölten, Austria

Stefan Marschalek
Josef Ressel Center TARGET
St. Pölten UAS
St. Pölten, Austria

Helge Janicke
De Montfort University Leicester
Leicester, UK

Edgar Weippl
SBA Research
Vienna, Austria

ABSTRACT

The poster corresponding to this summary depicts a proposition of a system able to explain anomalous behavior within a user session by considering anomalies identified through their deviation from a set of baseline process graphs. We adapt star structures, a bipartite representation used to approximate the edit distance between two graphs. Relevant processes are selected from a dictionary of benign and malicious traces generated through a sentiment-like bigram extraction and scoring system based on the log likelihood ratio test. We prototypically implemented smart anomaly explication through a number of competency questions derived and evaluated by a decision tree. The determined key factors are ultimately mapped to a dedicated APT attack stage ontology that considers actions, actors, as well as target assets.

CCS CONCEPTS

•Security and privacy → Intrusion/anomaly detection and malware mitigation; •Mathematics of computing → *Graph algorithms;*

KEYWORDS

Intrusion detection, malware, anomaly, behavioral analysis, knowledge generation, graph

ACM Reference format:
Robert Luh, Sebastian Schrittwieser, Stefan Marschalek, Helge Janicke, and Edgar Weippl. 2017. Poster: Design of an Anomaly-based Threat Detection
& Explication System. In *Proceedings of SACMAT'17, June 21–23, 2017, Indianapolis, IN, USA, ,* 2 pages.
DOI: http://dx.doi.org/10.1145/3078861.3084162

The financial support by the Austrian Federal Ministry of Science, Research and Economy and the National Foundation for Research, Technology and Development is gratefully acknowledged.

SACMAT'17, June 21–23, 2017, Indianapolis, IN, USA
 ACM ISBN 978-1-4503-4702-0/17/06.
DOI: http://dx.doi.org/10.1145/3078861.3084162

1 INTRODUCTION

The system introduced on the corresponding poster to the original paper published at ICISSP 2017 [7] is designed to primarily combat advanced persistent threats (APTs) and the malicious software utilized by this class of cyber-attacks. APTs are highly targeted to one specific entity (organization, system, device) and usually cause significantly more damage than bulk attacks in terms of privacy breaches, or monetary damage. APTs increasingly affect less prominent targets; 60% of APTs target small and medium businesses in retail, finance, and healthcare sectors. Today's threat mitigation strategies such as signature-based detection systems are not effective against these attacks.

For threat definition and initial modeling we use an adapted version of the Cyber Kill Chain [2] combined with our own APT ontology [6]. This ontology models actors and assets, APT attack stages (from reconnaissance to actions on objective), individual attack actions and their semantic description, as well as events and anomalies that can be captured by various data providers.

2 SYSTEM DESIGN

Figure 1 depicts the proposed system and its components. There are six stages in the process of collecting, processing, and analyzing the potentially malicious behavioral data:

Data collection – Surveying several data providers [4], we designed a kernel driver able to capture various events in a Microsoft Windows environment: Process, thread, file, registry, image load operations, and network events are captured. This is complemented by a Netflow component.

Event linking – Events are linked by their process ID (PID) and thread ID (TID) as well as their timestamps, creating "smart traces" that consider process and thread context while retaining most of their internal chronology. Mimicry attacks are largely prevented.

Grammar inference – Sequitur is used to infer rules frequently seen in the trace [5]. This step includes lossless recursive compression and the application of a semantic labeling mechanism for inferred rules (i.e. compound events).

Sentiment mining – We use an approach akin to sentiment mining to identify relevant OS processes [9] to determine likely trace event pairs using the log likelihood ratio test. After learning a set of bigrams typically found in benign and malicious scenarios, we compile a dictionary and apply scoring to generate knowledge and

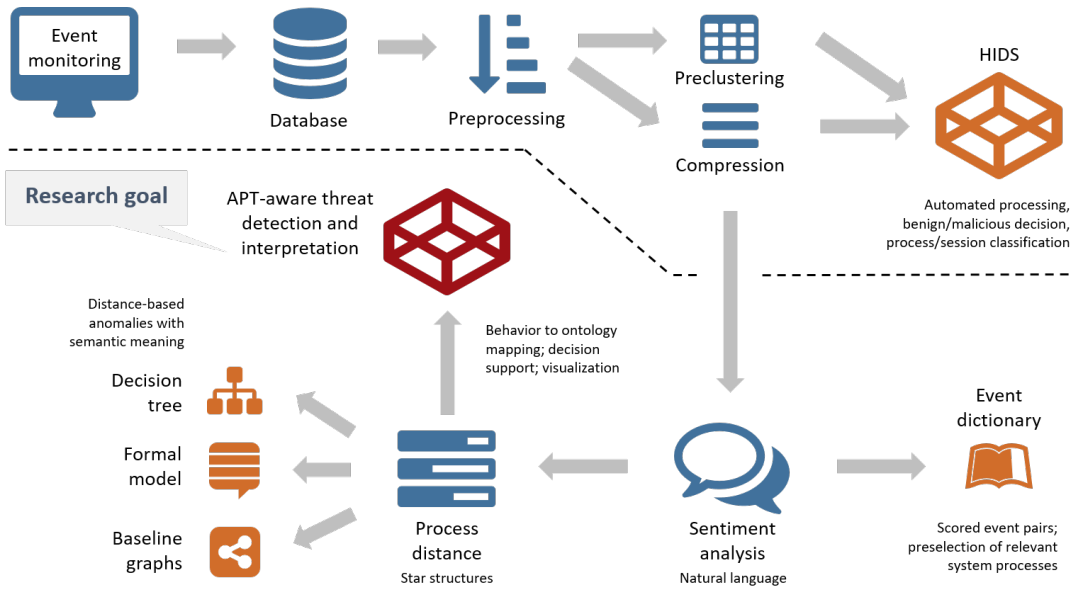

Figure 1: APT detection & explication system processing stages

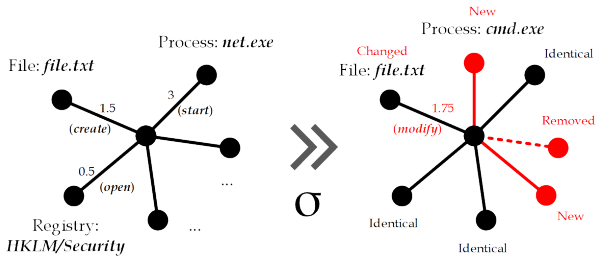

Figure 2: Transformation σ of baseline to target graph for example `svchost.exe` process [7]

ultimately determine which processes, event types and parameters contribute most to the good/bad decision.

Star construction – Events are broken down to star structures [1] $G = (U, V, E)$, where U and V are nodes and E is the respective edge. The edge label describes the basic operation. Both are used for minimal cost calculation based on bipartite graph matching using the Kuhn-Munkres algorithm [3]. For example: $G(svchost.exe, 1.5, file.txt)$ describes the creation of a file by OS process `svchost.exe`; edit operations σ are determined by type of event and type of operation E (create, modify, delete).

Anomaly detection & explication – Baseline templates for benign/known process behavior are created using Malheur heuristic clustering [8]. This automatically determines prototypes and value thresholds for anomaly detection. New traces are checked against these templates and the edit distance is calculated (see Figure 2).

Anomaly explication is two-pronged: Deviating events such as new processes, altered file operations, and the like are summarized in a human-readable report. These reports are then fed to a decision tree rooted in the six APT categories, sans weaponization [2].

3 CONCLUSION

The introduced star structure-based anomaly explication system is able to detect and interpret anomalous deviations in operating system process behavior. The returned output of detailed state changes as well as a tendency towards a specific APT stage or action is expressed through the mapping of semantic key factors to a dedicated attack ontology. The process was prototypically implemented and successfully tested using real-world process data captured on several company workstations. Please refer to [7] and the poster for additional information.

REFERENCES

[1] Xin Hu, Tzi-cker Chiueh, and Kang G Shin. 2009. Large-scale malware indexing using function-call graphs. In *16th conference on Computer and communications security*. ACM, 611–620.

[2] Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research* 1 (2011), 80.

[3] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

[4] Robert Luh, Stefan Marschalek, Manfred Kaiser, Helge Janicke, and Sebastian Schrittwieser. 2016. Semantics-aware detection of targeted attacks: a survey. *Journal of Computer Virology and Hacking Techniques* (2016), 1–39.

[5] Robert Luh, Gregor Schramm, Markus Wagner, and Sebastian Schrittwieser. 2017. Sequitur-based Inference and Analysis Framework for Malicious System Behavior. In *1st International Workshop on FORmal methods for Security Engineering, ICISSP*. https://doi.org/10.5220/0006250206320643

[6] Robert Luh, Sebastian Schrittwieser, and Stefan Marschalek. 2016. TAON: An ontology-based approach to mitigating targeted attacks. In *iiWAS 2016*. ACM.

[7] Robert Luh, Sebastian Schrittwieser, Stefan Marschalek, and Helge Janicke. 2017. Design of an Anomaly-based Threat Detection & Explication System. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*. 397–402. https://doi.org/10.5220/0006205203970402

[8] Konrad Rieck, Philipp Trinius, Carsten Willems, and Thorsten Holz. 2011. *Automatic analysis of malware behavior using machine learning*. Journal of Computer Security.

[9] Sebastian Schrittwieser Robert Luh and Stefan Marschalek. 2017. LLR-based sentiment analysis for kernel event sequences. In *31st IEEE International Conference on Advanced Information Networking and Applications (AINA)*.