AKMIS 2024

The 3rd Workshop on Application of Knowledge Methods in Information Security

11-13 June 2024, Smolenice, Slovakia

BOOK OF ABSTRACTS

organized by

- Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, FEI STU
- Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, FMPH UK
- Institute of Informatics, Slovak Academy of Sciences, II SAS
- Mathematical Institute, Slovak Academy of Sciences, MI SAS

EDITORS:

- ***Štefan Balogh***

  affiliated with Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava

- ***Martin Homola***

  affiliated with Faculty of Mathematics, Physics and Informatics Comenius University in Bratislava

- ***Ján Mojžiš***

  affiliated with Institute of Informatics, Slovak Academy of Sciences

ISBN 978-80-974468-1-9

EAN 9788097446819

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-19-0220.

PREFACE:

Increasing amounts of available data are currently pushing forward almost any area of human interest. In information security, such data sources contain knowledge that can be facilitated to improve detection, protection, and timely response against ever-more frequent security threats. The AKMIS Workshop series aims to bring together scientists, researchers, and practitioners to provide an open platform for discussion about recent research developments and future trends in facilitating data and knowledge to improve information security. AKMIS aims at inclusivity, and the book of the abstracts was issued by the project ORBIS consortium.

This volume contains the proceedings from the 3rd AKMIS Workshop, which took place in Smolenice, Slovakia, June 11–13, 2024. This year, AKIMS received 14 submissions in the form of extended abstracts. Each submission received two professional reviews from the program committee or additional reviewers. All 14 submissions turned out to be relevant to the workshop; therefore, they were accepted and are included in this volume.

AKMIS 2024 was held under the general patronage of the ORBIS project, supported by the SRDA agency under contract No. APVV-19-0220. The workshop co-chairs would like to thank all our colleagues, authors, and participants for their help in organizing the workshop, for nice presentations, and for fruitful discussions. We wish to have the chance to organize another workshop, AKMIS. Finally, we would like to thank all the reviewers for their insightful and precise reviews, which helped improve the quality of the published abstracts.

The list of the reviewers at AKMIS 2024: Štefan Balogh, Martin Homola, Martin Kenyeres, Ján Mojžiš, Ivana Budinská, Elena Štefancová, Peter Anthony, Damas Gruska, Peter Švec, and Zekeri Adams.

Bratislava, June 2024

Štefan Balogh
Ivana Budinská
Martin Homola
Ján Mojžiš
Martin Kenyeres

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-19-0220.

TOPICS:

The workshop is organized with the aim to bring together scientists, researchers and practitioners from any country and provide an open platform for discussion about recent research developments and future trends in creative and informal atmosphere. Application of various methods from knowledge extraction, management and artificial intelligence, such as:

- automated reasoning,
- information retrieval,
- data mining,
- machine learning,
- knowledge representation,
- ontologies reasoning, explanation,
- neural networks, argumentation, automated question answering,
- description logics learning algorithms,
- and others

with applications in

- system monitoring,
- information intelligence sharing,
- malware detection,
- malware features analysis,
- ontology based security,
- context reasoning,
- ontology models for security domain,
- incident detections, share schemes and responds strategies,
- and others

The fields of interests of the workshop include topics related to knowledge sharing in various domains. Therefore the topics are not limited to the information security domain.

This research was funded by the Scientific Grant Agency of the Slovak Republic grant number APVV-19-0220.

CONTENTS:

1. ***Diagnosability as opacity***
   authored by ***Damas Gruska***
2. ***Can distributed consensus algorithms for data aggregation be used to mitigate malware threats?***
   authored by ***Martin Kenyeres, Ivana Budinska, and Jan Mojzis***
3. ***Adversarial examples for concept learning based malware detection models***
   authored by ***Peter Svec***
4. ***Platform for sharing cybersecurity information's based on semantic data representation***
   authored by ***Stefan Balogh***
5. ***Tailoring logic explained for explainable malware detection and characterization***
   authored by ***Peter Anthony et al.***
6. ***Towards Building an Improved Malware Detection System: Investigating the LightGBM Model***
   authored by ***Onoja Monday, Martin Homola, and Abayomi Jegede***
7. ***Exploring meta-modelling languages toward conceptual meta-modelling ontology design patterns. A study on ML2 and PURO***
   authored by ***Zekeri Adams et al.***
8. ***Enhancing dynamic malware analysis through ontology integration***
   authored by ***Zekeri Adams et al.***
9. ***Comparison of concept learning tools***
   authored by ***Alexander Simko et al.***
10. ***Are my explanations any good?***
    authored by ***Martin Homola***
11. ***Investigation of the possibility of using matrix in asymmetric cryptographic systems***
    authored by ***Alisher Mavlonov and Karol Nemoga***
12. ***Malware clustering of PE files in current era: experiences, limitations, and ways forward***
    authored by ***Martin Mocko and Daniela Chuda***
13. ***Detection methods of eBPF-based rootkits in linux***
    authored by ***Peter Strycek and Roderik Ploszek***
14. ***Adversarial examples for machine learning based malware detection versus ontological approach***
    authored by ***Pavol Zajac***

Diagnosability as opacity

Damas Gruska

Department of Applied Informatics, Comenius University in Bratislava,
Mlynská dolina, 842 48 Bratislava, Slovakia
gruska@fmph.uniba.sk

Abstract. The opacity security property ([BKR05]) expresses the ability of a potential attacker to obtain classified information about the system. However, this property can also be used inversely to express the ability of the defender to obtain information about the state of the current attack and thus set an appropriate defensive strategy. As a basic formalism, we use attack trees (see [Sch99]). The simple and intuitive descriptions of attack trees have allowed for various extensions of the concept. These include adding defense [Kor+14], protection [AG19], and countermeasures [RKT12] nodes to model how to stop an attacker from reaching its target. Researchers have proposed the enrichment of attack trees with various attributes [Bul+20], notably cost [EK19], [Dew+12], [LS23b] and time [AG21], [LS23a]. The idea here we consider attack tress enriched with options to express some temporal properties of the attack in terms of duration and in what time an attack on a particular part of the system can be executed, as well as the cost of each step of the attack. We assume, that the defender cannot identify the initial stages of the attack, where and when the attack started but has information, on which parts of the system have already been successfully attacked. The question we ask is under what conditions the defender can diagnose the initial phase of the attack based on this information. This diagnosis can be used either for the defense itself or for changes in the design of the system to make it more robust. For this, we use the initial-state opacity property, which expresses that the attacker cannot detect the initial state of the system. We transform the attack tree into a nondeterministic finite automaton and show how diagnosability can be reduced to opacity for different kinds of information available to the defender (attacked parts of the system, time or cost).

Keywords: Attack Trees, opacity, diagnosability

Acknowledgments. This work was supported by the Slovak Research and Development Agency under the contract No. APVV-19-0220.

References

[Sch99] Bruce Schneier. “Attack trees”. In: *Dr. Dobb’s journal* 24.12 (1999), pp. 21–29.

[BKR05] Jeremy W. Bryans, Maciej Koutny, and Peter Y.A. Ryan. "Modelling Opacity Using Petri Nets". In: *Electronic Notes in Theoretical Computer Science* 121 (2005). Proceedings of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP 2004), pp. 101–115. ISSN: 1571-0661. DOI: https://doi.org/10.1016/j.entcs.2004.10.010. URL: https://www.sciencedirect.com/science/article/pii/S1571066105000277.

[Dew+12] Rinku Dewri et al. "Optimal security hardening on attack tree models of networks: a cost-benefit analysis". In: *International Journal of Information Security* 11 (2012), pp. 167–188.

[RKT12] Arpan Roy, Dong Seong Kim, and Kishor S Trivedi. "Attack countermeasure trees (ACT): towards unifying the constructs of attack and defense trees". In: *Security and communication networks* 5.8 (2012), pp. 929–943.

[Kor+14] Barbara Kordy et al. "Attack–defense trees". In: *Journal of Logic and Computation* 24.1 (2014), pp. 55–87.

[AG19] Aliyu Tanko Ali and Damas P Gruska. "Attack Protection Tree." In: *CS&P*. 2019.

[EK19] Julia Eisentraut and Jan Křetínskỳ. "Expected cost analysis of attack-defense trees". In: *Quantitative Evaluation of Systems: 16th International Conference, QEST 2019, Glasgow, UK, September 10–12, 2019, Proceedings 16*. Springer. 2019, pp. 203–221.

[Bul+20] Ahto Buldas et al. "Attribute evaluation on attack trees with incomplete information". In: *Computers & Security* 88 (2020), p. 101630.

[AG21] Aliyu Tanko Ali and Damas P Gruska. "Attack Trees with Time Constraints." In: *CS&P*. 2021, pp. 93–105.

[LS23a] Milan Lopuhaä-Zwakenberg and Mariëlle Stoelinga. "Attack time analysis in dynamic attack trees via integer linear programming". In: *International Conference on Software Engineering and Formal Methods*. Springer. 2023, pp. 165–183.

[LS23b] Milan Lopuhaä-Zwakenberg and Mariëlle Stoelinga. "Cost-damage analysis of attack trees". In: *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE. 2023, pp. 545–558.

Can distributed consensus algorithms for data aggregation be used to mitigate malware threats?

Martin Kenyeres[1], Ivana Budinská[1], and Ján Mojžiš[1]

[1] Institute of Informatics, Slovak Academy of Sciences, Dubravska cesta 9, 845 07 Bratislava, Slovakia
martin.kenyeres@savba.sk

As shown in the literature, malware is found to be one of the most serious threats to many modern digital technologies [1-2]. This cyber threat can cause serious damage to computers and even disrupt the operation of the whole system. Over the past years, we have witnessed the creation of numerous mechanisms to mitigate malware threats [1-2]. As malware permanently evolves (making it more sophisticated and harmful), older-generation techniques do not pose an effective tool against many of these attacks [1]. As seen in [3], distributed consensus algorithms for data aggregation find the application to suppress or even overcome many negative environmental factors affecting the quality of service (e.g., noise, radiation, temperature, weather conditions, etc.) in numerous modern technologies (wireless sensor networks, the Internet of Things, etc.). These identified facts motivate us to address the applicability of the mentioned algorithms as a tool against malware and thus potentially find a novel way of the protection against malware. In [3], we consider scenarios where the sensor readings in sensor networks are negatively affected by Gaussian noise, causing the measured values to differ from the observed unknown parameter. In this paper, we change the configuration of the average consensus algorithm with the Perron matrix and the implemented stopping criterion in order to examine the applicability of the mentioned algorithm to suppressing Gaussian noise in wireless sensor networks. In Figure 1, we show an example of how the inner states can be skewed due to noise.

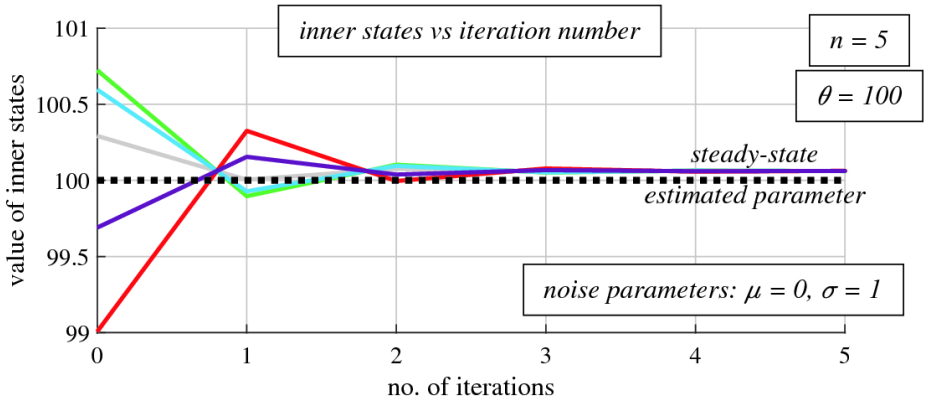

Fig. 1. Example of evolution of inner states when sensor readings are affected by Gaussian noise – steady state differs from value of estimated parameter. [3]

So, it is seen that the examined algorithm with the Perron matrix can operate with a small error when the sensor readings have been affected by Gaussian noise causing their deviation. In Figure 2, we provide the results of how three distributed average consensus algorithms with the Perron matrix (namely, the algorithm based on the graph order, the maximum degree, and the Laplacian eigenvalues) can compensate for incorrect sensor data caused by Gaussian noise. From the figure, it is seen that every applied algorithm can significantly suppress this noise whereby the application of data aggregation mechanisms can significantly increase the estimation precision.

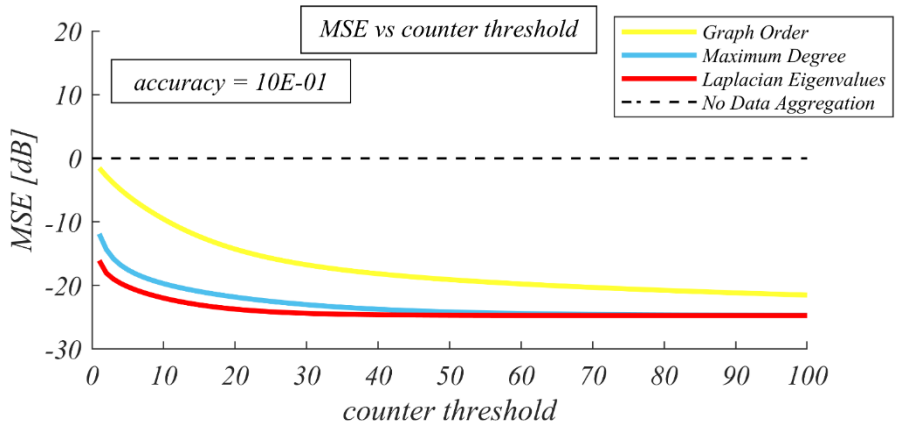

Fig. 2. Analysis of how three distributed average consensus algorithms with Perron matrix can alleviate incorrect sensor readings – mean square error (MSE) as function of *counter threshold* for *accuracy* = 10E-01 is depicted. [3]

In the submitted paper [4], we examine the applicability of the Metropolis-Hastings algorithm, a fully distributed consensus approach for data aggregation, to suppressing/overcoming Gaussian noise, like in [3]. We observe that the examined algorithm can alleviate the incorrect sensor readings up to 24.74 dB, which is a significant increase in the estimation precision again. Thus, this research confirms the applicability of another consensus-based algorithm to suppressing negative environmental factors. The presented results justify our planned research activities, i.e., the investigation of the applicability of distributed consensus algorithms for data aggregation to suppressing/overcoming malware and its effective localization. Therefore, our intention is to find a novel way of how to protect computer systems against malware in an effective manner.

Acknowledgments. This work was supported by the Slovak Scientific Grand Agency VEGA under the contract 2/0135/23 "Intelligent sensor systems and data processing" and by the Slovak Research and Development Agency under the contract No. APVV-19-0220 and No. SK-SRB-23-0038.

References

1. Gopinath, M., Sethuraman, S.C.: A comprehensive survey on deep learning based malware detection techniques. Computer Science Review **47**, 100529 (2023)
2. Aslan, Ö.A., Samet, R.: A comprehensive review on malware detection approaches. IEEE access **8**, 6249-6271 (2020).
3. Kenyeres, M., Kenyeres, J.: Average Consensus with Perron Matrix for Alleviating Inaccurate Sensor Readings Caused by Gaussian Noise in Wireless Sensor Networks. Lecture Notes in Networks and Systems **1**, 391-405 (2022).
4. Kenyeres, M., Kenyeres, J.: Using Metropolis-Hastings Algorithm to Suppress Incorrect Sensor Data in Wireless Sensor Networks. Submitted.

Adversarial Examples for Concept Learning Based Malware Detection Models

Peter Švec[0000−0002−8315−5301]

Institute of Computer Science and Mathematics, Faculty of Electrical Engineering and Information Technology Slovak University of Technology, Ilkovičova 3, 81219 Bratislava, Slovakia
peter.svec1@stuba.sk

Machine learning models are known to be vulnerable against inputs specially crafted by an attackers. These inputs, also referred to as adversarial examples, are introducing a small changes to the input data that cause a misclassification. The first concept of adversarial examples was proposed by Szegedy et al. [9]. Generating an adversarial examples for malware classifiers is a different task compared to image classification domain. While in images, we can change arbitrary bytes, it is no longer possible in executable files (where strict format is used). Hence the main goal of an attacker is to modify the executable file, without disrupting the original malicious behaviour, so that model classifies it as benign. There have already been proposed number of attacks in the domain of traditional machine learning algorithms and malware detection [6][5][4][3].

In this work, we propose similar attacks for models (or more specifically class expressions) developed using concept learning algorithms [2]. As baseline models we used class expressions by Švec et al. [8]. These models were trained using *PE Malware Ontology* [7], which is based on popular EMBER dataset [1]. Our proposed attacks are based on modifying various features in ontology such as debugging symbols, signatures, section entropies or API calls. We tested our attacks on two different ontological datasets for algorithms OCEL, CELOE, PARCEL and SPACEL [11][10]. We also discussed two different scenarios under which attacks can occur: *black-box* (attacker has no knowledge about class expressions and he can only send his samples and observe outputs) and *white-box* (attacker has full knowledge about class expressions).

Acknowledgents. This research was sponsored by the Slovak Republic under grants APVV-19-0220 (ORBIS) and by the EU H2020 programme under Contract no. 952215 (TAILOR)

References

1. Anderson, H.S., Roth, P.: EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. ArXiv e-prints (Apr 2018)
2. Bühmann, L., Lehmann, J., Westphal, P., Bin, S.: Dl-learner structured machine learning on semantic web data. In: Companion Proceedings of the The Web Conference 2018. pp. 467–471 (2018)

3. Kolosnjaji, B., Demontis, A., Biggio, B., Maiorca, D., Giacinto, G., Eckert, C., Roli, F.: Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables. 2018 26th European Signal Processing Conference (EUSIPCO) pp. 533–537 (2018)
4. Kreuk, F., Barak, A., Aviv-Reuven, S., Baruch, M., Pinkas, B., Keshet, J.: Deceiving End-to-End Deep Learning Malware Detectors using Adversarial Examples (2018)
5. Park, D., Khan, H., Yener, B.: Generation and Evaluation of Adversarial Examples for Malware Obfuscation. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). pp. 1283–1290 (2019)
6. Suciu, O., Coull, S.E., Johns, J.: Exploring Adversarial Examples in Malware Detection. 2019 IEEE Security and Privacy Workshops (SPW) pp. 8–14 (2018)
7. Švec, P., Balogh, Š., Homola, M., Kl'uka, J.: Knowledge-based dataset for training pe malware detection models. arXiv preprint arXiv:2301.00153 (2022)
8. Švec, P., Balogh, Š., Homola, M., Kl'uka, J., Bisták, T.: Semantic data representation for explainable windows malware detection models. arXiv preprint arXiv:2403.11669 (2024)
9. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014), http://arxiv.org/abs/1312.6199
10. Tran, A.C., Dietrich, J., Guesgen, H.W., Marsl, S.: Parallel symmetric class expression learning. Journal of Machine Learning Research **18**(64), 1–34 (2017)
11. Tran, A.C., Dietrich, J., Guesgen, H.W., Marsland, S.: An approach to parallel class expression learning. In: Rules on the Web: Research and Applications: 6th International Symposium, RuleML 2012, Montpellier, France, August 27-29, 2012. Proceedings 6. pp. 302–316. Springer (2012)

Platform for sharing cybersecurity information's based on semantic data representation.

Štefan Balogh, [1]

[1] Faculty of Electrical Engineering, Slovak Technical University, Bratislava

Abstract. In this paper we discuss the new web platform for cybersecurity data sharing.

Keywords: data sharing, cybersecurity, semantic datastore.

1 Project Idea

Data sharing is highly desired in the security domain, significantly aiding in improving protection against attacks and providing appropriate responses to incidents [1]. Various types of information can be shared, such as details on newly discovered vulnerabilities, attack techniques, or insights from malware analysis, including its functionalities and exploitation techniques used for system breaches, among others [2]. Recognizing the need to share different types of information, we have developed a solution that addresses this requirement. The solution allows the use of a local knowledge repository and integration with external knowledge sources. For this purpose, we have utilized a semantic database. The system is designed to provide a web interface over the semantic database, where organizations can store information, they choose to retain (see figure 1). The stored information can be displayed to local users or shared with other organizations using our solution via a published URL. To access the shared knowledge, it's not necessary to have our solution; any semantic database that can function as an RDF or OWL endpoint will suffice. Our goal is to enable each organization or security group to create their own endpoint to store and share information from their findings or analyses with allied institutions or the public sector. More about this concept can be found in the article [2]. Through this solution, we aim to realize the ideas presented in the article.

Fig. 1. Displayed entity structure.

2 Solution Description

To The system for sharing information and knowledge in the security domain is essentially a web application designed to store and share information and knowledge from various areas of the security domain, facilitating further interconnection based on existing relationships. This includes insights from malware analysis, exploit techniques, software vulnerabilities, and other useful information about attacks.

The system is modeled similarly to Wikidata[1] that collects structured data. Wikidata stores not only the data but also their sources and links to other databases, enhancing the diversity of available knowledge while providing the ability to verify the data based on the available sources.

The application also features bilingual functionality, making it available in both Slovak and English. This bilingualism applies to everything except the data from the security domain presented on the website, as this data is sourced from other automated sources and maintained in its original language.

3 Data Storage Method

For effective data storage, search, and sharing, the system uses ontology and semantic representation of data and the concept of Linked Data [3], which involves linking data on the web to make them easily connectable and processable. These data are linked using URI (Uniform Resource Identifier), enabling the creation of extensive data networks.

Currently, we use the Blazegraph database[2] to manage and query the necessary RDF data model. This approach significantly improves the file loading speed and the overall performance of the application. The application currently includes data from the MITRE ATT&CK database, categorized into techniques, mitigations, groups, software, and tactics. The ontological model also includes data from Hybrid Analysis[3] as can be seen in figure 2.

Future expansions of the ontological model will involve adding data from the CVE database and other sources focused on the security domain. We communicate with the Blazegraph database via REST API, and an HTTP wrapper has been created to execute SPARQL queries. The system also handles the automated updating of existing ontologies and parsing of .owl files to extract predicates.

[1] https://www.wikidata.org

[2] https://blazegraph.com/

[3] https://www.hybrid-analysis.com

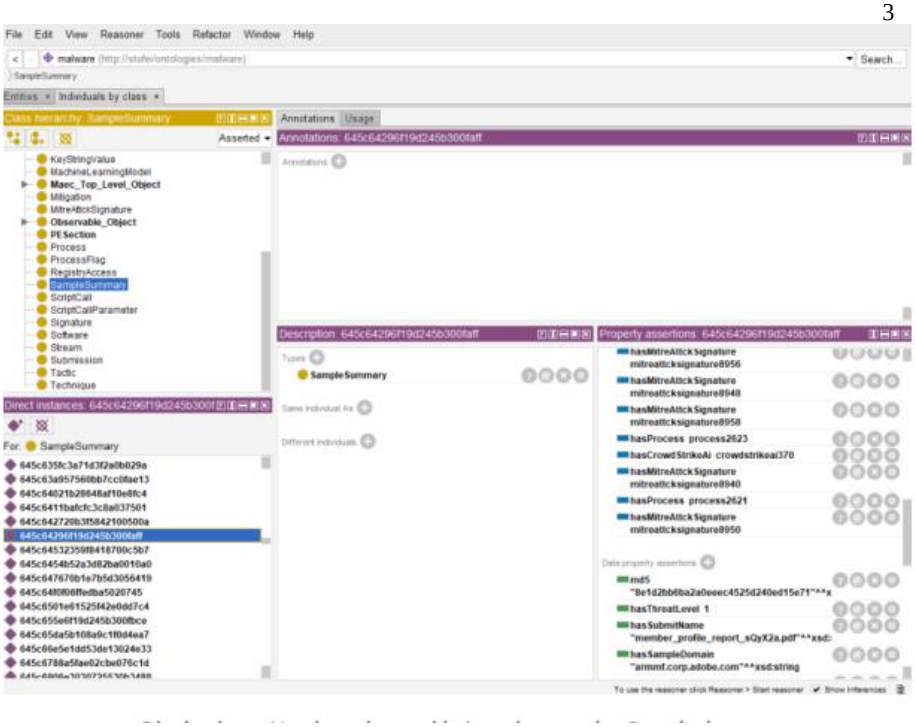

Fig. 2. Populated ontology displayed in Protégé

4 Web Interface

The system uses a configuration file that allows customization of the names displayed to the user. The user uploads an ontology from which a configuration file is created. This file contains basic names parsed from the ontology. After logging in, the user can modify these names as desired. For example, changing the name "Title" to "Name."

The user can access and modify data properties, object properties, searchable attributes, and basic ontology information. The second functionality of the configuration file is for searching. When searching for an entity, all configuration files are combined, and their searchable data (i.e., properties by which the entities can be searched) are used.

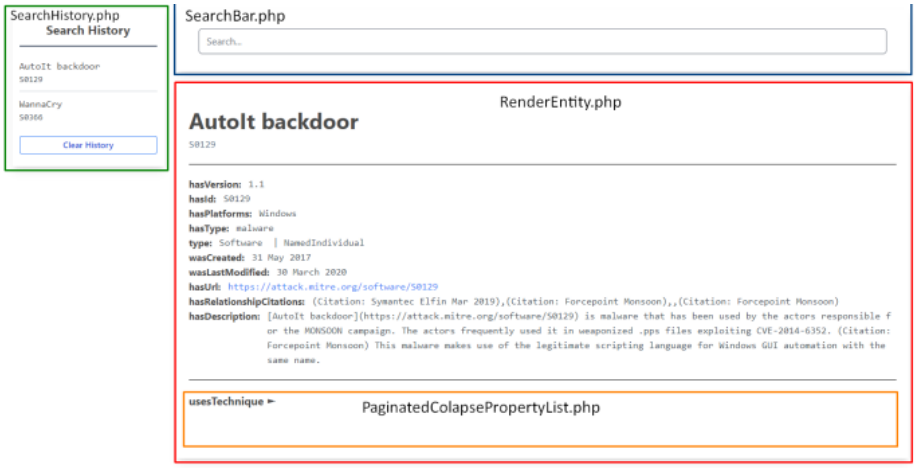

Fig. 3. Layout of the web application

5 Conclusion

In this way, we have created a universal platform that can be stored in a Docker file and, after a simple installation, can ensure the storage of required information from the security domain and facilitate its sharing (a feature of the semantic database). Each institution or organization thus has a simple platform to use for sharing their knowledge and accessing knowledge from other providers. A similar principle is used by the MISP platform[4] and also described in [4], but its solutions does not store knowledge in OWL (RDF) form and does not use a semantic database with properties that allow quick and efficient integration of various sources.

Our platform also offers a simple and user-friendly front-end for displaying stored knowledge see figure 3. The potential of semantic databases for data integration is exemplified by the Wikidata platform, which integrates a vast amount of diverse information based on this principle. This is why we chose this approach, and we expect our solution to be practically utilized.

Acknowledgments. This work was supported by the Slovak Research and Development Agency under the contract No. APVV-19-0220.

[4] https://www.misp-project.org/

References

1. Baker, Laura (2019), Sharing Information is the Key, https://www.cyberwyoming.org/sharing-information-is-the-key/, last accessed 2021/08/01.
2. Balogh, Š. (2021). Knowledge and datasets as a resource for improving artificial intelligence. In Data Science and Intelligent Systems: Proceedings of 5th Computational Methods in Systems and Software 2021, Vol. 2 (pp. 828-837). Springer International Publishing.
3. Bizer, C., Heath, T., & Berners-Lee, T. (2023). Linked data-the story so far. In *Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web* (pp. 115-143).
4. Takahashi, T., Panta, B., Kadobayashi, Y., & Nakao, K. (2018). Web of cybersecurity: Linking, locating, and discovering structured cybersecurity information. International Journal of Communication Systems, 31(3), e3470.

Tailoring Logic Explained for Explainable Malware Detection and Characterization

Peter Anthony[1,*], Francesco Giannini[2], Michelangelo Diligenti[2], Martin Homola[1], Marco Gori[2], Štefan Balogh[3], and Ján Mojžiš[4]

[1] Department of Applied Informatics, Comenius University Bratislava, Slovakia
anthony2, homola}@uniba.sk
[2] Department of Information Engineering and Mathematics, University of Siena, Italy
{francesco.giannini, michelangelo.diligenti, marco.gori}@unisi.it
[3] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, Slovakia
stefan.balogh@stuba.sk
[4] Institute of Informatics, Slovak Academy of Sciences, Slovakia
jan.mojzis@savba.sk

Abstract. Logic Explained Networks (LENs) have emerged as a promising paradigm for explainable machine learning, offering interpretable insights by leveraging First-Order Logic (FOL) formulations of human-understandable predicates. This paper explores the application of LENs in the complex domain of malware detection, particularly focusing on their effectiveness over extensive real-world datasets, such as the EMBER malware dataset comprising 800,000 labeled samples with numerous features. By maximizing the trade-off between accuracy and interpretability, LENs exhibit a capacity to distill compact and meaningful explanations, aligning with classifier predictions. Our study demonstrates the adaptability of LENs to the challenges posed by large-scale datasets inherent in malware detection. We showcase their capability to discriminate malware effectively, rivaling state-of-the-art black-box models while outperforming traditional interpretable-by-design methods. Moreover, we introduce an innovative approach to augment the fidelity of class-level explanations extracted by LENs, enhancing their utility in understanding and combating malicious software. Through rigorous experimentation and evaluation, we establish LENs as a robust malware detection framework, offering insights that are both meaningful and comprehensible. This research contributes to the advancement of explainable AI techniques in security applications and underscores the potential of LENs in addressing real-world challenges in diverse domains.

Keywords: Malware Detection · Logic Explained Network · Explainable AI.

Acknowledgments:
This work was supported by the Slovak Research and Development Agency under the contract No. APVV-19-0220

Towards Building an Improved Malware Detection System: Investigating the LightGBM Model

Onoja Monday[1] , Martin Homola[2] and Abayomi Jegede[3]

[1] Federal University of Health Sciences, Otukpo 145, Otukpo, Nigeria
[2] Comenius University in Bratislava, Slovakia
[3]University Jos P.M.B 2084 Jos, Nigeria
monday.onoja@fuhso.edu.ng, martin.homola@fmph.uniba.sk, jegedea@unijos.edu.ng,

Abstract: LightGBM is the best of the Gradient Boosting Decision Tree algorithm, which is suitable for malware detection. Previous studies did not evaluate the training time of the model for malware detection. There is a need to determine the time required for training the LightGBM model since it is necessary to decide on its effectiveness and further improvement. Our preliminary investigation reveals the training time of the LightGBM algorithm can be determined for Windows malware detection using the Malimg dataset. Further studies we carried out also show that the performance of the LightGBM algorithm can be improved by hybridizing with the pre-trained XceptionCNN model to extract relevant and robust features before passing them to the LightGBM model for training, thereby building an improved, efficient, and effective Malware Detection System. This significant reduction in the training time makes it possible for the model to converge quickly and train a large sum of data within a relatively short period. Overall, the reduction in detection time and improved detection accuracy will minimize damage to files stored in computer systems in the event of a malware attack. However, Explanations are not being derived for classification of malicious or benign software behaviors, even at a very high accuracy of 100% TPR. These explanations allow malware experts to build trust in the results/output generated by machine learning algorithms, thereby increasing the chances of users acting based on a system's output and for better assessment and improvement of the system. Hence, there's a need for dynamic malware detection using explainable AI for improved Detection, Better Threat Intelligence, and Adaptive Response Strategies.

Keywords: Machine Learning; LightGBM; Malware Detection; Windows Malware Malimg Dataset, Explainable AI.

Exploring meta-modelling languages towards conceptual meta-modelling ontology design patterns. A study on ML2 and PURO.

Zekeri Adams[1], Martin Homola[1], Ján Kl'uka[1], and Vojtech Svatek[2]

[1] Comenius University in Bratislava, Faculty of Mathematics, Physics, and Informatics.
{homola, jan, zekeri.adams}@fmph.uniba.sk
[2] Prague University of Economics and Business, Czech Republic
svatek@vse.cz

Abstract. In the field of ontology engineering, conceptual modeling serves as the backbone for representing and structuring domain-specific knowledge. As the complexity and scope of ontologies continue to grow, there is an increasing need for effective methodologies and tools to streamline the design and implementation process. One promising approach is the use of ontology design patterns (ODPs), which are reusable solutions to common modeling problems. ODPs provide a structured way to capture best practices and facilitate the creation of high-quality ontologies. Meta-modeling languages play a pivotal role in the development of these design patterns. By offering a higher level of abstraction, meta-modeling languages enable the definition and manipulation of models and their relationships, thereby enhancing the expressiveness and flexibility of ontology design. Our investigation delves into two meta-modeling languages, ML2 and PURO, both equipped with higher-order constructs tailored for modelling subject domains with intricate features. These languages are formalized in first-order logic, providing a structured framework to accommodate complexities often absent in models within such domains. Our objective is to synthesize the unique attributes of PURO and ML2 to create a comprehensive framework for modeling across complex domains. Through the utilization of the PURO modeler, we demonstrate the practical significance of these languages by identifying issues in multi-level taxonomic structures, using segments from Wikidata as illustrative examples.

Keywords: Meta-modeling · Ontology design pattern · Conceptual modeling.

1 Introduction

Meta modeling involves modeling the structure and relationships of other models. It provides an abstract representation of the characteristics and elements that can exist within models of a given domain. Thus, such domains are endowed

with higher order constructs where classes are instances of other classes. The multi-level conceptual modelling theory, MLT[1] formalized in first-order logics provides a framework for conceptual ontology design patterns for domains with multi level classification. Models built with this theory are expressed in the textual language ML2[3] and the graphical language PURO[2] with the aid of the PURO modeler.

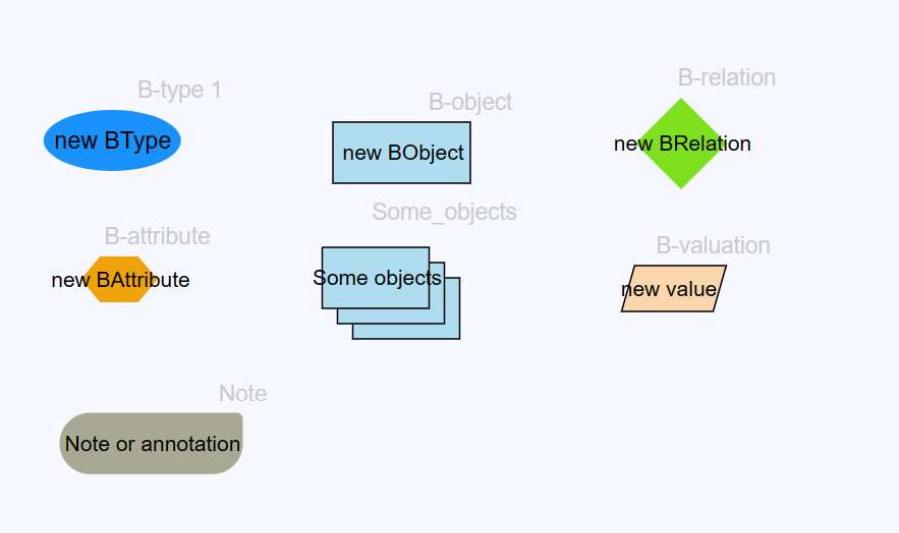

Fig. 1. PURO modeler

2 Results

Despite prior findings on the existence of anti-patterns in Wikidata and the improvements made, we illustrate with the use of the PURO modeler to establish some patterns and anti-patterns still exist in the knowledge base.

- Entity of second-order class instantiate a second-order class.

Using the query

- SELECT DISTINCT ?x ?y WHERE
- ?x wdt:P31/(wdt:P279*) wd:Q24017414.
- ?y wdt:P31/(wdt:P279*) wd:Q24017414.
- ?x wdt:P31/(wdt:P279*) ?y.

31 results of these patterns were found in Wikidata. For instance, item Cellphone model(Q19723444) is an instance of Type of manufactured goods(Q22811462), both of which are second-order class. This violates the rule for a homogeneous B-type.

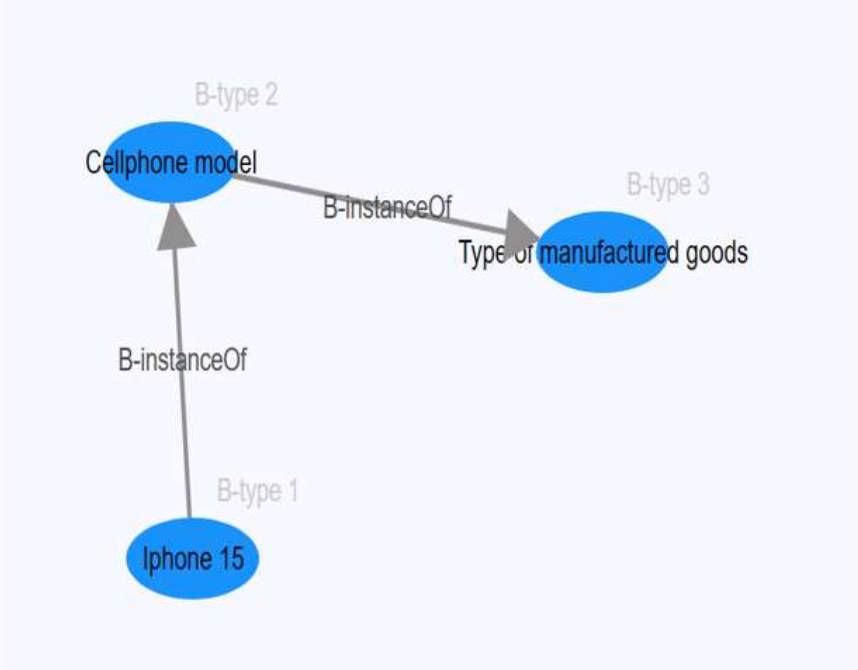

Fig. 2. An anti-pattern in Wikidata

References

1. Fonseca CM, Almeida JPA, Guizzardi G, de Carvalho VA. Multi-level conceptual modeling: Theory, language and application. Data Knowl Eng. 2021;134:101894
2. Dudáš M, Hanzal T, Svátek V, Zamazal O. OBOWLMorph: Starting Ontology Development from PURO Background Models. In: OWLED 2015. vol. 9557 of LNCS. Springer; 2015. p. 14-20
3. Almeida JPA, Carvalho VA, Brasileiro F, Fonseca CM, Guizzardi G. Multi-Level Conceptual Modeling: Theory and Applications. In: ONTOBRAS. vol. 2228 of CEUR-WS; 2018. .
4. Svátek V, Homola M, Kl'uka J, Vacura M. Metamodeling-Based Coherence Checking of OWL Vocabulary Background Models. In: OWLED. vol. 1080.

Acknowledgments:
This work was supported by the Slovak Research and Development Agency under the contract No. APVV-19-0220

ENHANCING DYNAMIC MALWARE ANALYSIS THROUGH ONTOLOGY INTEGRATION.

Zekeri Adams[1], Monday Onoja[1], Martin Homola[1], Ján Kl'uka[1], and Štefan Balogh[2]

[1] Comenius University in Bratislava, Faculty of Mathematics, Physics, and Informatics.
{monday.onoja, zekeri.adams, homola, jan}@fmph.uniba.sk
[2] Faculty of Electrical Engineering and Information Technology Slovak University of Technology Ilkovicova 3, Slovakia
stefan.ocds@gmail.com

Abstract. Dynamic malware analysis involves executing potentially malicious software in a controlled environment to observe its behaviour in real-time. This method is essential for detecting sophisticated malware that employs evasion techniques to avoid static analysis. Ontology, as a structured framework for organizing knowledge, can significantly enhance dynamic malware analysis by providing a formal representation of the concepts, relationships, and rules within this domain. By integrating ontological models, dynamic analysis can achieve a higher level of precision and efficiency in identifying and categorizing malware behaviours. Furthermore, the representation of malware features with human understandable concepts will enhance interpretability. Although, deep learning approaches have been applied to extract malware features during dynamic analysis, but they are limited in exploiting heterogeneous information from different types of arguments. As such it is imperative to apply ontological approach which provides a framework to integrate information from diverse sources. In this work, we aimed at developing an ontology-based framework for dynamic malware analysis, enhance feature extraction and representation through ontological models, improve malware detection and classification accuracy using ontology-integrated techniques, and create a comprehensive knowledge base to support automated and scalable analysis.

Keywords: Dynamic analysis · Malware· Ontology.

Acknowledgments:This work was supported by the Slovak Research and Development Agency under the contract No. APVV-19-0220.

Comparison of concept learning tools

Alexander Šimko[1], Peter Švec[2], Martin Homola[1], Claudia d'Amato[3], Umberto Straccia[4], and Francesco Giannini[5]

[1] Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Mlynská dolina, 84248 Bratislava, Slovakia
alexander.simko@uniba.sk, homola@fmph.uniba.sk

[2] Institute of Computer Science and Mathematics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia
peter.svec1@stuba.sk

[3] Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Campus Via Orabona 4, 70215 Bari, Italy
claudia.damato@uniba.it

[4] Istituto di Scienza e Tecnologie dell'Informazione, CNR, Pisa, Italy
umberto.straccia@isti.cnr.it

[5] Department of Information Engineering and Mathematics, University of Siena, Italy
francesco.giannini@unisi.it

Abstract. In this paper we report our work on an experimental evaluation of concept learning tools.

Keywords: concept learning · description logics · malware detection

Concept learning are methods that given an input ontology, a set of positive, and a set of negative examples, try to come up with a concept expression that characterizes (ideally) all the positive and none of the negative examples. To this day, several tools and algorithms for concept learning exists, such as DL-Learner [5], DL-Focl [6], and Evolearner [4]. Recently, concept learning was successfully used for finding characterizations of computer malware [7]. The experiments in [7] were carried out on a description logic ontology representation of the EMBER dataset [3] using DL-Leaner. Our aim is to extend the evaluation to DL-Focl and Evolearner. To this day there is no experimental study comparing all three tools. The malware ontology of [7] has low expressiveness, and serves mainly as the dictionary of symbols used when searching for a concept expression. Our comparison will therefore be carried out also on more expressive ontologies, specifically the Financial [1] and Geoskill [2] ontologies. The current state of our work is that we have generated several datasets. For each of the three ontologies (Financial, Geoskill, Malware) we have nine datasets: three datasets with 125 negative and 125 positive examples, three datasets with 250 negative and 250 positive examples, and three dataset with 500 negative and 500 positive examples. When running experiment a dataset will be further split into a training and testing set using k-fold cross validation. In the case of the financial

and geoskill datasets, the concept expressions that we are trying to find using the tools are known beforehand. The sets of positive and negative examples were deduced from a randomly generated concept expressions as in [6]. In the malware datasets, no such target concept expressions are known beforehand. The positive and negative examples describe the real world malware.

Acknowledgements. This research was sponsored by the Slovak Republic under the grant APVV-19-0220 (ORBIS) and by the EU under the H2020 grant no. 952215 (TAILOR) and under Horizon Europe grant no. 101079338 (TERAIS).

References

1. Financial dataset, http://www.cs.put.poznan.pl/alawrynowicz/financial.owl
2. Geoskills dataset, https://github.com/i2geo/GeoSkills
3. Anderson, H.S., Roth, P.: EMBER: An open dataset for training static PE malware machine learning models. arXiv preprint arXiv:1804.04637 (2018)
4. Heindorf, S., Blübaum, L., Düsterhus, N., Werner, T., Golani, V.N., Demir, C., Ngomo, A.N.: Evolearner: Learning description logics with evolutionary algorithms. In: Laforest, F., Troncy, R., Simperl, E., Agarwal, D., Gionis, A., Herman, I., Médini, L. (eds.) WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022. pp. 818–828. ACM (2022). https://doi.org/10.1145/3485447.3511925
5. Lehmann, J.: DL-learner: Learning concepts in description logics. The Journal of Machine Learning Research **10**, 2639–2642 (2009). https://doi.org/10.5555/1577069.1755874
6. Rizzo, G., Fanizzi, N., d'Amato, C.: Class expression induction as concept space exploration: From DL-Foil to DL-Focl. Future Generation Computer Systems **108**, 256–272 (2020)
7. Švec, P., Balogh, Š., Homola, M.: Experimental evaluation of description logic concept learning algorithms for static malware detection. In: ICISSP. pp. 792–799 (2021)

Are My Explanations Any Good?*

Martin Homola and Elena Štefancová

Comenius University in Bratislava, Mlynská dolina, 8428 Bratislava, Slovakia
{homola,elena.stefancova}@fmph.uniba.sk

Keywords: Malware detection · Explanations · Quality.

In the recent works, the ORBIS consortium has focused on different approaches to "explainable malware detection", a task to classify malware samples in such a way that a human-readable explanation is also provided – mostly in form of a logical formula or a concept description. But are these explanations indeed useful for their intended users?

This talk will focus on the different types of explanations in the malware domain and then on different ways how to evaluate their quality and especailly usefuness to the users.

* This work was sponsored by the Slovak Republic under the grant no. APVV-19-0220 (ORBIS).

INVESTIGATION OF THE POSSIBILITY OF USING MATRIX IN ASYMMETRIC CRYPTOGRAPHIC SYSTEMS

Karol Nemoga[1][0000−0002−4613−1765] and Alisher Mavlonov[2][0000−0001−8987−6110]

[1] Mathematical Institute, Slovak Academy of Sciences,
Štefánikova 49, SK-814 73 Bratislava, Slovakia
nemoga@mat.savba.sk
[2] Urgench State University,
14, Kh.Alimdjan str, 220100 Urgench, Uzbekistan
{mavlonosher}@gmail.com

Abstract. In this presentation we will look at how quantum computers solve problems used in modern cryptographic systems, and what post-quantum cryptographic systems exist. We focus on the scheme, that developed in the direction of multivariate cryptography. Our presentation presents a symmetric cryptosystem in a finite field utilizing a matrix A_{mxn} and its right inverse B_{nxm} for $m > n$. This property of matrices, particularly evident when the dimensions are equal $m = n$, is actively used in AES. This utilization involves matrix transformations integral to the SubBytes() and InvSubBytes() functions. Moreover, a similar design principle is employed in the O'zDSt 1105:2006 data encryption algorithm, where it constitutes essential components of the AralashHolat() and TesAralashHolat() transformations. Additionally, we introduce a one-way function based on a special matrix with a zero determinant in a finite field, a concept not currently existing in asymmetric cryptography within our scheme. This approach fortifies defenses, diversifies cryptosystems, and addresses current vulnerabilities. We delve into matrix based asymmetric algorithms in finite fields, seeking to fortify our digital realm against the quantum unknown. This research promises post-quantum security, aligns with existing cryptographic abstractions, and fosters agility in an evolving landscape.

References

1. T. Acar, J. Benaloh, C. Costello, and D. Shumow, "Evaluating post-quantum asymmetric cryptographic algorithm candidates", ST/post-quantum-2015/presentations/session7-shumow-dan. pdf, pp. 6-10, 2015.
2. D. Coppersmith, A. M. Odlzyko, and R. Schroeppel, "Discrete logarithms in GF (p)", Algorithmica, vol. 1, no. 1, pp. 1–15, 1986.
3. O'zDSt 1105:2006., State Standard of Uzbekistan. Information technology. Cryptographic information protection. Hash function. - Tashkent. Uzbek Agency for Standardization, Metrology and Certification. 2006. (in Russian).

4. A. Mavlonov, "Investigation of the possibility of using matrix multiplication in asymmetric cryptographic systems", Academic research in educational sciences, vol. 2, no. 10, pp. 89–93, 2021

Acknowledgments:
This research was supported by the Slovak Republic under Grant APVV-19-0220 (ORBIS) and by the Vega Grant 2/0119/23.

Malware clustering of PE files in current era: experiences, limitations, and ways forward

Martin Mocko[1,2][0000−0001−8982−0141] and Daniela Chudá[2,3][0000−0002−3873−9308]

[1] Brno University of Technology, Brno, Czech republic
[2] Kempelen Institute of Intelligent Technologies, Bratislava, Slovakia
name.surname@kinit.sk
[3] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, Slovakia
daniela.chuda@stuba.sk

Abstract. During the last few years, a few large public malware datasets have been released, which significantly improve the potential of comparability of research in the malware domain. This gave rise to many malware detection papers which have shown interesting results. However, the same kind of attention has yet to be paid to malware clustering - or, more broadly - clustering of Portable Executable (PE) files. In this extended abstract, we offer a brief look into three exciting topics in malware clustering - the impact of large datasets and the subsequent usage of large(r) amount of clusters, the influence of deep clustering on malware clustering, and the potential of contrastive (CL) and self-supervised learning (SSL) to improve the performance of malware clustering.

Keywords: Malware clustering · Large datasets · Representation learning.

1 Introduction

Since 2017-2018, the situation of malware datasets in the malware research community has begun improving. In 2018, Ember v2 [1] was published, the first significant contribution to the research community. Along with it, the code for extracting all of its features was published in an open-source GitHub repository[4]. This later enabled the publication of other significant malware research datasets - namely Bodmas [2] and Sorel [3], with the latter being arguably the first ever public dataset to reach almost 20 million samples. These publications allowed researchers to create various exciting research works. However, this situation has not translated too well into the task of malware clustering, which does not see as much attention from the research community as it should.

There are various issues with malware clustering - such as lack of comparability, lack of publications on large real-world datasets, labeling problems, lack of consensus on how clustering solutions should be set up, and the usage of different

[4] https://github.com/elastic/ember

evaluation metrics. We focus on addressing some of these issues in our research work. The clustering of PE files can be used for various tasks, such as helping domain experts prioritize what new incoming never-before-seen samples should be subjected to a deeper analysis. Another example is using clustering to represent the data space and sampling from these clusters when creating a representative dataset for downstream ML (e.g., detection or classification) models.

We divide our so-far conducted extensive experiments into two sets: 1. shallow clustering model experiments to establish the state-of-the-art and 2. deep embedded clustering (DEC) experiments to improve the situation. The first set of experiments with shallow clustering models showed that what works on small datasets may not necessarily translate to larger sets. We employed two public datasets - Bodmas and Ember, and one private industry dataset - Security. The worst results in terms of clustering Homogeneity were achieved for the Ember dataset - only in the 40-44% range. More detailed results of our experiments can be found in Figure 1.

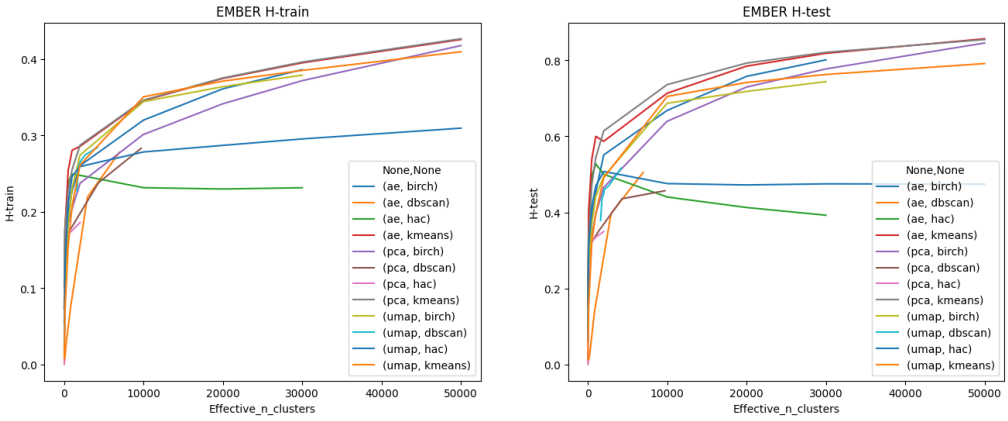

Fig. 1. RO1 experiment combinations of (representation, clusterer) that were run on the EMBER dataset. Both Homogeneity on train and test sets is reported. X-axis corresponds to the number of clusters that the clusterer created.

The latter, DEC experiments, were conducted to create an end-to-end clustering solution and with the aim to improve the clustering situation. However, these experiments proved unsuccessful. The best results are usually achieved before Epoch 0, when DEC has just been initialized using K-Means and the actual DEC training has not yet started. All other subsequent training epochs saw worse results. The more epochs that were run, the worse the Homogeneity was. This research direction of utilizing deep clustering has, therefore, been discontinued.

2 Ways forward

Our research in malware clustering has established the SotA in shallow clustering methods while utilizing relatively commonly used representation methods. However, as we mentioned, the clustering quality may still need improvement for some datasets. We propose recommendations for improving the clustering quality, the main ways of improving the resulting clustering quality we see is by improving the underlying representation created from the malware features. We can employ more advanced ways of creating the embedded feature space to improve the representation. Contrastive (CL) and self-supervised learning (SSL) both utilize sample pairs, potentially improving the embeddings. What remains is choosing the proper method for the task and figuring out the best way to create sample pairs to improve the representation quality. Multiple approaches for creating sample pairs may be utilized, such as distance functions, complex rules, or synthetic samples.

References

1. Anderson, Hyrum S., and Phil Roth. Ember: an open dataset for training static pe malware machine learning models. arXiv preprint arXiv:1804.04637 (2018).
2. Yang, Limin, et al. BODMAS: An open dataset for learning based temporal analysis of PE malware. 2021 IEEE Security and Privacy Workshops (SPW). IEEE, 2021.
3. Harang, Richard, and Ethan M. Rudd. SOREL-20M: A large scale benchmark dataset for malicious PE detection. arXiv preprint arXiv:2012.07634 (2020).

Detection Methods of eBPF-Based Rootkits in Linux

Peter Strýček and Roderik Ploszek[0000−0002−3192−0630]

Slovak University of Technology in Bratislava, Slovakia
Institute of Computer Science and Mathematics
{xstrycek, roderik.ploszek}@stuba.sk

This paper presents proposal of new detection methods for eBPF-based rootkits, a novel threat vector in the Linux operating system. Our research involved a detailed analysis of six eBPF rootkit projects—*TripleCross*, *ebpfkit*, *Boopkit*, *nysm*, *Bad BPF*, and *evilBPF*—showcasing the potential of eBPF technology to obscure malicious artifacts.

eBPF (extended Berkeley Packet Filter) technology allows users to load and run programs in special instruction set in kernel space, thus extending the kernel without the need to change the kernel source code or compiling a loadable kernel module [2]. While this technology is beneficial for system monitoring, it presents new vectors for rootkit development. Rootkits are malicious programs designed to hide their presence and activities, complicating detection efforts. Kernel-level rootkits may stealthily modify kernel data structures to achieve a variety of malicious goals, which may include hiding malicious user space objects, installing backdoors, logging keystrokes, and disabling firewalls [1]. Due to their nature, eBPF-based rootkits may only modify user space memory. While not as dangerous as kernel module rootkits, they still pose a significant threat.

We explored and evaluated both existing and newly developed detection techniques across various scenarios. These scenarios included active and passive states of rootkits and situations where rootkits were installed before or after the deployment of a security solution. Our detection methods included a kernel module for enumerating eBPF programs, a tool called *proclook* to identify hidden processes, and a scanner to locate traced functions.

Key findings demonstrated that attaching an eBPF program incurs a noticeable performance impact, which can be exploited for rootkit detection. The research also identified significant security implications due to the capability of eBPF-based rootkits to modify user values and override function return values, thereby concealing their presence and preventing the deployment of security measures.

Our contributions include *proclook*, which successfully detected hidden processes like *Boopkit*, and a loadable kernel module that identified inconsistencies between the kernel and user space views of running eBPF programs. We also proposed a cross-view detection approach for active *ftrace* using eBPF. The technical details of the solution and a more detailed context can be found in [3].

Acknowledgments. We would like to thank Mr. Peter Košinár for his valuable suggestions. This work was supported by the Slovak Research and Development Agency

under the Contract no. APVV-19-0220 and by the Science Grant Agency - project VEGA 1/0105/23.

References

1. Baliga, A., Ganapathy, V., Iftode, L.: Detecting kernel-level rootkits using data structure invariants. IEEE Transactions on Dependable and Secure Computing **8**(5), 670–684 (2010)
2. eBPF.io authors: eBPF Documentation, https://ebpf.io/what-is-ebpf/
3. Strýček, P.: Detection of Linux rootkits utilizing eBPF. Master's thesis, Institute of Computer Science and Mathematics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, https://opac.crzp.sk/?fn=detailBiblioForm&sid=9A7D98113A4E7D6F09AB76A0EB46, RN: FEI-16607-103137

Adversarial examples for Machine Learning based malware detection versus ontological approach

Pavol Zajac *

Slovak University of Technology in Bratislava, Slovakia

Abstract

Machine learning is often proposed as a suitable tool for malware detection and classification [2]. Learning is based on different features that result both from static analysis:

- string analysis,
- bytes and op-code N-grams,
- API function calls,
- entropy,
- image representation,
- fuction and control flow graphs;

as well as from dynamic analysis:

- memory and register's usage,
- instruction traces,
- network traffic,
- API call traces.

The extracted features from a large dataset of malware and non-malware samples are used for training the ML model. The trained model then typically acts as a black box for malware detection/classification.

From the security perspective, the main disadvantage of ML models is the existence of adversarial attacks. Adversarial Windows malware samples can bypass machine learning-based detection relying on static code analysis by perturbing relatively few input bytes. [1].

Recently we have conducted experiments with adversarial attacks on popular modern object detector YOLO [3]. The adversarial attacks can be automated relatively easily by modifying existing tools. The result of the work is essentially hiding in plain sight (see Figure 1), the ML detection performance is significantly reduced by small changes that are essentially imperceptible to human eye.

The rise of the adversarial techniques and the ease of use of the existing attack tools can thus help even non-sophisticated malware creators to hide their malware from detection by ML models.

*This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-19-0220.

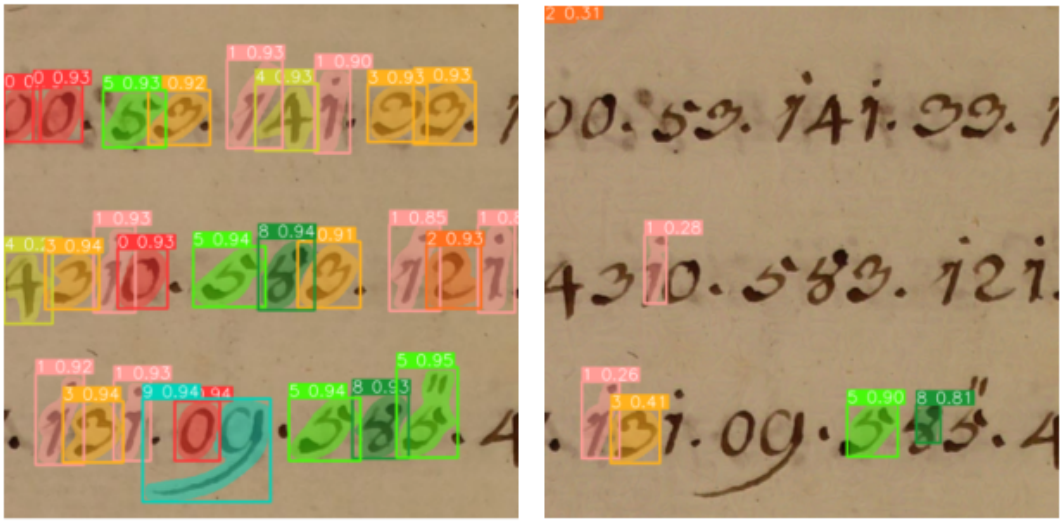

Figure 1: Adversarial example attack on YOLOv8 detector [3].

Adversarial examples rely on small changes that typically do not affect core features of the examined malware samples, but affect the ML detector due to inherent approximations embedded in the model. In the context of our project ORBIS, existence of malware samples is a strong motivation to enhance malware detection by a more systematic approach that can be based on ontological representation of data, and logical inference of malware features.

On the other hand, we can ask whether our approach is also vulnerable to the existence of specific types of adversarial examples. A continuation of our research project can try to answer two core research questions: How our existing tools behave in the presence of known adversarial malware examples? Are there any specific ways to attack our detection models that are specific to our approach, but are easily detectable with other malware detectors?

References

[1] Demetrio, L., Coull, S. E., Biggio, B., Lagorio, G., Armando, A., and Roli, F. Adversarial exemples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection. *ACM Transactions on Privacy and Security (TOPS) 24*, 4 (2021), 1–31.

[2] Gibert, D., Mateu, C., and Planes, J. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications 153* (2020), 102526.

[3] Meliš, L. Adversarial machine learning (in slovak). Master's thesis, FEI STU in Bratislava, 2024.