BiG2-KAMAS: Supporting Knowledge-Assisted Malware Analysis with Bi-Gram Based Valuation

Niklas Thür[1], Markus Wagner[1], Johannes Schick[1], Christina Niederer[1], Jürgen Eckel[3],
Robert Luh[2], Wolfgang Aigner[1]

[1]University of Applied Sciences, St. Pölten, Austria

[2]Josef Ressel Center for Unified Threat Intelligence on Targeted Attacks, Austria

[3]IKARUS Security Software GmbH, Austria

Email: [1,2]first.last@fhstp.ac.at, [3]eckel.j@ikarus.at

Abstract

Malicious software, short *malware*, refers to software programs that are designed to cause damage or to perform unwanted actions on the infected computer system. The behavior-based analysis of malware typically utilizes tools that produce lengthy traces of observed events, which have to be analyzed manually or by means of individual scripts. Due to the growing amount of data extracted from malware samples, analysts are in need of an interactive tool that supports them in their exploration efforts. In this respect, the use of visual analytics methods and stored expert knowledge helps the user to speed up the exploration process and, furthermore, to improve the quality of the outcome. In this paper, the previously developed KAMAS concept is extended with components such as a bi-gram based valuation approach to cover further malware analysts' needs. The components have been integrated a new prototype which was evaluated by two domain experts in a detailed user study.

Index Terms: K.6.1 [Information Interfaces and Presentation]: User Interfaces—User-centered design—Evaluation/methodology;

1 Introduction & Related Work

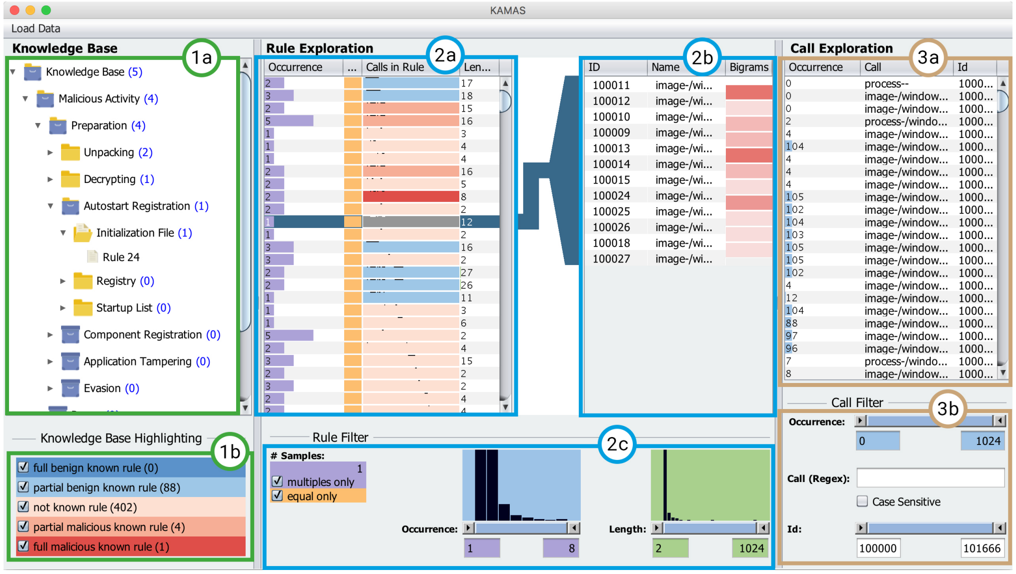
Malicious software (*malware*) is one of the biggest threats to computer systems these days [6]. Malware includes viruses, trojan horses, worms, rootkits, scareware, and spyware [6]. By now there are millions of malicious programs and the number is increasing every day. *Malware analysis* is commonly defined as “the art of dissecting malware to understand how it works, how to identify it, and how to defeat or eliminate it” [6]. Egele et al. [3] presented a general literature for malware analysis techniques and tools. For the categorization of such systems, Wagner et al. [8] published a survey of different visualization systems for malware analysis and developed a novel ’Malware Visualization Taxonomy’. To cover all of the malware analyst's needs, Wagner et al. [7] performed a problem characterization and abstraction elaborating the analysts needs in behavior-based malware analysis. In a design study for behavior-based knowledge-assisted malware analysis, a novel system called KAMAS was presented [10]. The malware analyst's workflow involves the tasks of examining potentially malicious rules, selecting them, categorizing them, and storing the found rules in a knowledge database (KDB) [10]. A focus group meeting with members of an Austrian IT security company, an IT security university research department, and the developers of the initial KAMAS prototype was conducted to identify the need for additional features requested by domain experts to extend the KAMAS design study [10]. We developed an interactive prototype to extend the KAMAS design study [10] with the new feature of **Bi**-**G**ram supported **G**eneric **K**nowledge-**A**ssisted **M**alware **A**nalysis **S**ystem (BiG2-KAMAS). The new features at hand include a *generic data loading process*, the *extension of the knowledge database (KDB)* for benign rules and the implementation of a *bi-gram based valuation approach* of Luh et al. [5]. A bi-gram is an n-gram where the length of $n = 2$. An n-gram, in turn, is a coherent sequence of n elements. In this approach the elements are system or API calls. Each bi-gram has a score in the range [-1, 1], which indicates whether this pair of calls is malicious or benign. These features are evaluated in a user study to verify if the new features enhance the analysts' workflow.

2 Bi-gram Concept

This section describes the new features of the BiG2-KAMAS system. Since the BiG2-KAMAS prototype is based on the prototype of Wagner et al. [10], it also uses a data-oriented design concept [4]. The KDB was integrated to support the user during their analysis tasks and is based on the malware behavior schema of Dornhackl et al. [2]. The KDB is located at the left side of the prototype (see Figure 1:1a) and is implemented in a hierarchical tree structure. In the BiG2-KAMAS prototype the KDB was extended by one additional category to store the benign rule data (‘benign activity’).
Element Coloring: For the rule highlighting as well as the bi-gram visualization, a sequential coloring scheme from red to blue was selected. Red 🟥 indicates that the rule or bi-gram is malicious and a blue 🟦 one stands for a benign rule or bi-gram.
Bi-Gram Visualization: The bi-gram approach is visualized in the third column of the call overview table (see Figure 1:2b). For the bi-gram based valuation two different visualization approaches were implemented: First, if the width of the bi-gram column is bigger than 75px, the prototype visualizes the bi-gram values as bar charts, whereby each bar starts in the middle of the bi-gram column. If the bi-gram score is between 0 and -1, the bi-gram is malicious and visualized from the middle to the left in red. If the bi-gram score is between 0 and 1 the bi-gram is benign and the bar chart is visualized from the middle to the right side in blue. The visualization approach was chosen to give the user a quick but still precise overview of the bi-gram based scores. If the width of the bi-gram column is smaller than 75px, the bar charts are hardly recognizable. Thus, the system switches to the second visualization designed along the ‘semantic zoom’ [1] concept. Thereby, the bi-gram values are visualized as a colored filled rectangle. To visualize the value of the malicious or benign bi-gram, the system changes the alpha value of the displayed color. Therefore, the darker the color, the higher the value. Since the difference of an alpha value between 255 and 240 is not easy to recognize, we decided to implement only four graduation steps for the alpha value. The visualization with the alpha value is less precise than the visualization with the bar charts but easier to comprehend.

3 Evaluation & Discussion

Usability study: For the prototype validation, a user study with two domain experts was conducted. Each test took approximately one hour in which the domain experts validated the functionality as well



- Knowledge Base (5)
  - Malicious Activity (4)
    - Preparation (4)
      - Unpacking (2)
      - Decrypting (1)
      - Autostart Registration (1)
        - Initialization File (1)
          - Rule 24
        - Registry (0)
        - Startup List (0)
      - Component Registration (0)
      - Application Tampering (0)
      - Evasion (0)

Occurrence	...	Calls in Rule	Len...
2			17
3			18
2			15
5			16
1			3
1			4
1			4
2			16
2			5
2			8
2			2
1			12
1			2
3			16
3			2
2			27
2			26
1			11
1			3
1			6
5			2
2			4
3			15
2			2
3			2
2			4

ID	Name	Bigrams
100011	image-/wi...	
100012	image-/wi...	
100010	image-/wi...	
100009	image-/wi...	
100013	image-/wi...	
100014	image-/wi...	
100015	image-/wi...	
100024	image-/wi...	
100025	image-/wi...	
100026	image-/wi...	
100018	image-/wi...	
100027	image-/wi...	

Occurrence	Call	Id
0	process--	1000...
0	image-/window...	1000...
0	image-/window...	1000...
2	process-/windo...	1000...
4	image-/window...	1000...
104	image-/window...	1000...
4	image-/window...	1000...
4	image-/window...	1000...
4	image-/window...	1000...
105	image-/window...	1000...
102	image-/window...	1000...
104	image-/window...	1000...
103	image-/window...	1000...
105	image-/window...	1000...
105	image-/window...	1000...
102	image-/window...	1000...
4	image-/window...	1000...
12	image-/window...	1000...
104	image-/window...	1000...
88	image-/window...	1000...
97	image-/window...	1000...
96	image-/window...	1000...
7	process-/windo...	1000...
8	image-/window...	1000...

☑ full benign known rule (0)
☑ partial benign known rule (88)
☑ not known rule (402)
☑ partial malicious known rule (4)
☑ full malicious known rule (1)

# Samples: 1
☑ multiples only
☑ equal only

Occurrence: 1 – 8
Length: 2 – 1024

Occurrence: 0 – 1024
Call (Regex):
☐ Case Sensitive
Id: 100000 – 101666

Figure 1: The BiG2-KAMAS prototype and it's three sections: Section 1 shows the knowledge base, section 2 shows the rule exploration area and section 3 shows the rule exploration area.

as the visual interface design. Both participants have more than five years of experience in the field of malware analysis. Each participant was tested individually and had already tested the KAMAS prototype at least once. They tested the prototype in two scenarios with various input data to validate the generic data loading process. Both participants mentioned that the bi-gram visualization is very helpful to identify potentially malicious or benign call sequences, helping to decide whether a rule is malicious or not. Additionally, it could be valuable to implement a rule creation process where the analyst can build rules based on the known system and API calls [9]. **Fulfilled feature requests:** Furthermore, the performed user study confirmed that the following three feature requests are fulfilled by the BiG2-KAMAS prototype:

Generic data loading: The BiG2-KAMAS prototype is structured to enable the generic loading of any kind of data. To make this possible the input data as well as the prototype's database are based on unique identifiers instead of the actual values. Only with the corresponding translation table the system can translate the IDs to the actual values. Thus, it is possible to load any data as long as there is a translation table available.

Extend the KDB with benign rules: To fulfill this requirement the KDB was extended with an additional category for benign activity. The KDB's highlighting and filter pipeline were extended to identify and filter partially and fully benign rules, which are highlighted in blue to avoid a red and green hues for colorblind people [11, p. 124].

Implementation of bi-gram based valuation: To support the bi-gram approach [5] the prototype's rule detail table was adopted. Since many domain experts mentioned that the arc-diagram visualization is not very helpful [10], it was replaced by the bi-gram visualization. Thus, the bi-gram based valuation is implemented in two different approaches based on the available width ($size > 75px$:= bar chart, $size \leq 75px$:= alpha channel coloring).

In general, this work presented a design study for a 'Bi-Gram Supported Generic Knowledge-Assisted Malware Analysis System' (BiG2-KAMAS) including a description, demonstration and validation for its new implemented features.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF) via the "KAVA-Time" project (P25489) and by the Austrian Federal Ministry of Science, Research and Economy under the FFG Innovationscheck (no. 856429). We would also like to thank all focus group members and test participants who have agreed to volunteer in this project.

References

[1] R. Bade, S. Schlechtweg, and S. Miksch. Connecting time-oriented data and information to a coherent interactive visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pp. 105–112. ACM, NY, 2004. doi: 10.1145/985692.985706

[2] H. Dornhackl, K. Kadletz, R. Luh, and P. Tavolato. Malicious behavior patterns. pp. 384–389. IEEE, 2014. doi: 10.1109/SOSE.2014.52

[3] M. Egele, T. Scholte, E. Kirda, and C. Kruegel. A survey on automated dynamic malware-analysis techniques and tools. 44(2):6:1–6:42, 2008.

[4] R. Fabian. Data-Oriented Design, 2013. `http://www.dataorienteddesign.com/dodmain/dodmain.html`, accessed on August 11, 2017.

[5] R. Luh, S. Schrittwieser, and S. Marschalek. LLR-based Sentiment Analysis for Kernel Event Sequences. IEEE, 2017.

[6] M. Sikorski and A. Honig. *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. No Starch Press, 1 ed., 2012.

[7] M. Wagner, W. Aigner, A. Rind, H. Dornhackl, K. Kadletz, R. Luh, and P. Tavolato. Problem characterization and abstraction for visual analytics in behavior-based malware pattern analysis. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, VizSec '14. ACM, 2014. doi: 10.1145/2671491.2671498

[8] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner. A survey of visualization systems for malware analysis. In *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015. doi: 10.2312/eurovisstar.20151114

[9] M. Wagner, A. Rind, G. Rottermanner, C. Niederer, and W. Aigner. Knowledge-assisted rule building for malware analysis. In *Proceedings of the 10th Forschungsforum der österreichischen Fachhochschulen*. FH des BFI Wien, Vienna, Austria, 2016.

[10] M. Wagner, A. Rind, N. Thür, and W. Aigner. A knowledge-assisted visual malware analysis system: Design, validation, and reflection of KAMAS. *Computers & Security*, 67:1–15, 2017. doi: 10.1016/j.cose.2017.02.003

[11] C. Ware. *Information Visualization: Perception for Design*. Elsevier. Google-Books-ID: UpYCSS6snnAC.