Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

A knowledge-assisted visual malware analysis system: Design, validation, and reflection of KAMAS

Markus Wagner [a,b,*], *Alexander Rind* [a,b], *Niklas Thür* [a], *Wolfgang Aigner* [a,b]

[a] *St. Pölten University of Applied Sciences, St. Pölten, Austria*
[b] *Vienna University of Technology, Vienna, Austria*

ARTICLE INFO

Article history:
Received 25 June 2016
Received in revised form 12 November 2016
Accepted 5 February 2017
Available online 11 February 2017

Keywords:
Malicious software
Malware analysis
Behavior-based
Design study
Prototype
Visualization
Visual analytics
Interactive
Knowledge generation

ABSTRACT

IT-security experts engage in behavior-based malware analysis in order to learn about previously unknown samples of malicious software (malware) or malware families. For this, they need to find and categorize suspicious patterns from large collections of execution traces. Currently available systems do not meet the analysts' needs which are described as: visual access suitable for complex data structures, visual representations appropriate for IT-security experts, provision of workflow-specific interaction techniques, and the ability to externalize knowledge in the form of rules to ease the analysis process and to share with colleagues. To close this gap, we designed and developed KAMAS, a knowledge-assisted visualization system for behavior-based malware analysis. This paper is a design study that describes the design, implementation, and evaluation of the prototype. We report on the validation of KAMAS with expert reviews, a user study with domain experts and focus group meetings with analysts from industry. Additionally, we reflect on the acquired insights of the design study and discuss the advantages and disadvantages of the applied visualization methods. An interesting finding is that the arc-diagram was one of the preferred visualization techniques during the design phase but did not provide the expected benefits for finding patterns. In contrast, the seemingly simple looking connection line was described as supportive in finding the link between the rule overview table and the rule detail table which are playing a central role for the analysis in KAMAS.

1. Introduction

Malware (malicious software) is undoubtedly one of today's greatest threats to the Confidentiality/Integrity/Availability (CIA) triangle of information security (Stoneburner et al., 2002). It has become a common tool in digital theft, corporate and national espionage, spam distribution and attacks on infrastructure availability. When security professionals analyze malware in a real world setting, they have large volumes of complex and heterogeneous data at their disposal. In behavior-based malware analysis, they explore traces of malware execution in the form

[*] *Corresponding author.*
E-mail addresses: markus.wagner@fhstp.ac.at (M. Wagner), alexander.rind@fhstp.ac.at (A. Rind), niklas.thuer@fhstp.ac.at (N. Thür), wolfgang.aigner@fhstp.ac.at (W. Aigner).

http://dx.doi.org/10.1016/j.cose.2017.02.003
0167-4048/

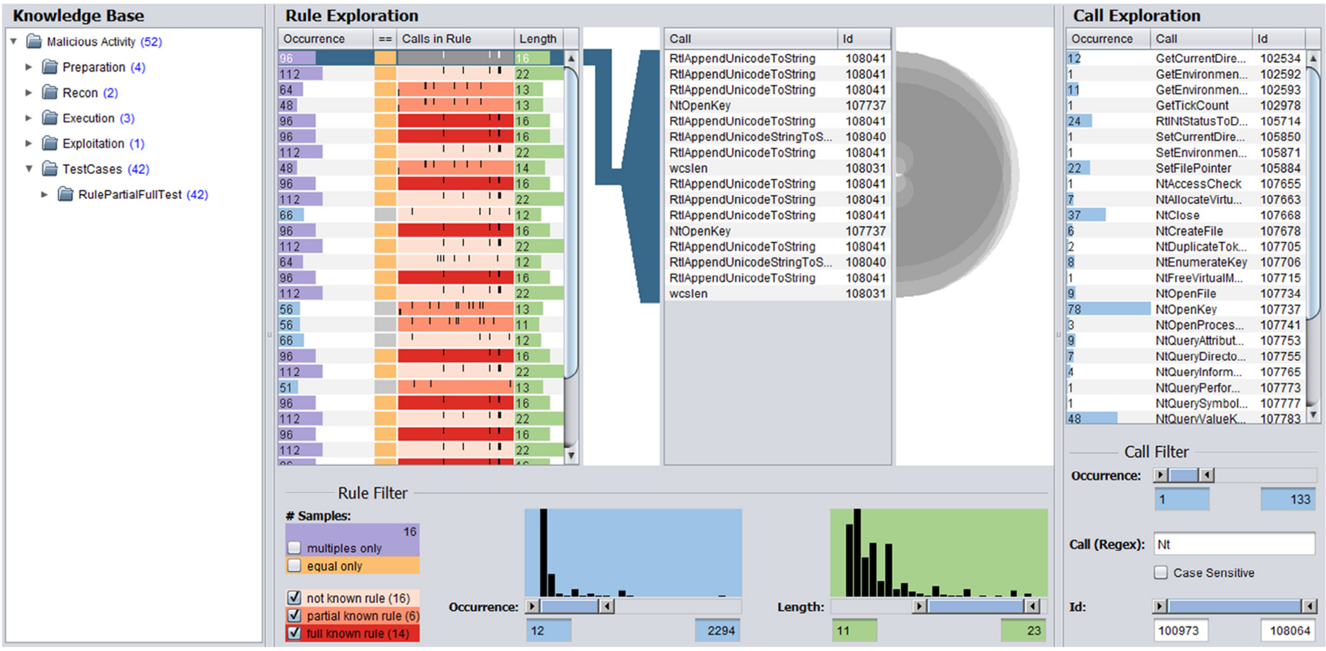
of system call sequences (rules) and frequently occurring subsequences of potentially malicious code (Egele et al., 2008). Their workflow involves the tasks of selecting different rules, categorizing them, and storing them in a database as well as manual adaption and/or tuning of found rules (Wagner et al., 2014). In addition to challenging analysis methods, "implicit knowledge" (Chen et al., 2009) or "tacit knowledge" (Wang et al., 2009) about the data, the domain experience or prior experience are often required to make sense of the data and not become overwhelmed. By externalizing some of the domain experts' implicit knowledge, it can be made available as explicit knowledge and stored in a knowledge database (KDB) (Chen et al., 2009).

In this paper, we present a visualization design study in the context of malware analysis. Visualization tools "provide visual representations of datasets designed to help people carry out tasks more effectively" (Munzner, 2014). Thus, they are particularly useful "to augment human capabilities rather than replace people with computational decision-making methods" (Munzner, 2014). Malware analysis lends itself very well to visualization, because the experience of analysts plays a central role in reconstructing the obfuscated behavior of malware. Users' needs in the context of behavior-based malware analysis were analyzed systematically in previous work (Wagner et al., 2014) and are summarized in Section 2. Currently, there are no interactive visualization tools available which cover all the needs of the malware analysis experts. To close this gap, we developed a novel Knowledge-Assisted Visual Malware Analysis System (KAMAS) (see Fig. 1). Additionally, we demonstrate how the visualization can benefit from explicit domain knowledge.

In order to achieve the best possible results, we followed the paradigm of problem-oriented research, i.e., collaborating with real users to solve their tasks (Sedlmair et al., 2012). Therefore, we worked in accordance with the *nested model for visualization design and validation* by Munzner (Munzner, 2009), which divides visualization design into four levels combined with appropriate validation methods (see Section 2). Specifically, we focused on the third and fourth level of Munzner's model (Munzner, 2009) (third level: visual encoding and interaction design, fourth level: algorithm design). Moreover, we evaluated the prototype's usability in relation to the interaction metaphors and knowledge representation needed by IT-security experts. Thus, the main contributions of our research are:

- We present the concept and implementation of KAMAS as systematically designed, developed and evaluated instantiation of an interactive visual method for handling large amounts of complex data in behavior-based malware analysis.
- We show that applying knowledge-assisted visual analytics methods allows domain experts to externalize their implicit knowledge and profit from this explicit knowledge in their analysis workflow.
- To provide evidence for the effectiveness of the developed methods, we provide a rigorous and reproducible validation of the introduced techniques with malware analysis experts.

This paper is organized as follows (see Fig. 2 for a graphical overview): Section 2 provides background information about the problem domain and provides a summary of our previous work on problem characterization and abstraction in relation to malicious software analysis. Section 3 presents related work in problem-oriented visualization research, malware analysis, and knowledge-assisted visualization. Section 4 provides a detailed description of our KAMAS prototype and Section 5 demonstrates the application of KAMAS in a usage scenario. Section 6 describes the used validation methods and



Knowledge Base

- Malicious Activity (52)
  - Preparation (4)
  - Recon (2)
  - Execution (3)
  - Exploitation (1)
  - TestCases (42)
    - RulePartialFullTest (42)

Rule Exploration

Occurrence	==	Calls in Rule	Length
96			16
112			22
64			13
48			13
96			16
96			16
112			22
48			14
96			16
112			22
66			12
96			16
112			22
64			12
96			16
112			22
56			13
56			11
66			12
96			16
112			22
51			13
96			16
112			22
96			16
112			22

Call	Id
RtlAppendUnicodeToString	108041
RtlAppendUnicodeToString	108041
RtlAppendUnicodeToString	108041
NtOpenKey	107737
RtlAppendUnicodeToString	108041
RtlAppendUnicodeStringToS...	108040
RtlAppendUnicodeToString	108041
wcslen	108031
RtlAppendUnicodeToString	108041
RtlAppendUnicodeToString	108041
RtlAppendUnicodeToString	108041
NtOpenKey	107737
RtlAppendUnicodeToString	108041
RtlAppendUnicodeStringToS...	108040
RtlAppendUnicodeToString	108041
wcslen	108031

Rule Filter

# Samples: 16
□ multiples only
□ equal only

☑ not known rule (16)
☑ partial known rule (6)
☑ full known rule (14)

Occurrence: 12 – 2294

Length: 11 – 23

Call Exploration

Occurrence	Call	Id
12	GetCurrentDire...	102534
1	GetEnvironmen...	102592
11	GetEnvironmen...	102593
1	GetTickCount	102978
24	RtlNtStatusToD...	105714
1	SetCurrentDire...	105850
1	SetEnvironmen...	105871
22	SetFilePointer	105884
1	NtAccessCheck	107655
7	NtAllocateVirtu...	107663
37	NtClose	107668
6	NtCreateFile	107678
2	NtDuplicateTok...	107705
8	NtEnumerateKey	107706
1	NtFreeVirtualM...	107715
9	NtOpenFile	107734
78	NtOpenKey	107737
3	NtOpenProces...	107741
9	NtQueryAttribut...	107753
7	NtQueryDirecto...	107755
4	NtQueryInform...	107765
1	NtQueryPerfor...	107773
1	NtQuerySymbol...	107777
48	NtQueryValueK...	107783

Call Filter

Occurrence: 1 – 133

Call (Regex): Nt
□ Case Sensitive

Id: 100973 – 108064

Fig. 1 – Screenshot of the KAMAS interface during the exploration of a cluster grammar that includes combined system and API call sequences (rules) gathered during the execution of malicious software to find malicious subsequences (patterns).