Report from Dagstuhl Seminar 17281

Malware Analysis: From Large-Scale Data Triage to Targeted Attack Recognition

Edited by

Sarah Zennou[1], Saumya K. Debray[2], Thomas Dullien[3], and Arun Lakhotia[4]

1 **Airbus – Suresnes, FR,** sarah.zennou@airbus.com
2 **University of Arizona – Tucson, US,** debray@cs.arizona.edu
3 **Google Switzerland – Zürich, CH,** thomas.dullien@gmail.com
4 **University of Louisiana – Lafayette, US,** arun@louisiana.edu

Abstract

This report summarizes the program and the outcomes of the Dagstuhl Seminar 17281, entitled "Malware Analysis: From Large-Scale Data Triage to Targeted Attack Recognition". The seminar brought together practitioners and researchers from industry and academia to discuss the state-of-the art in the analysis of malware from both a big data perspective and a fine grained analysis. Obfuscation was also considered. The meeting created new links within this very diverse community.

Seminar July 9–14, 2017 – http://www.dagstuhl.de/17281
1998 ACM Subject Classification formal methods, program analysis
Keywords and phrases big data, executable analysis, machine learning, malware, obfuscation, reverse engineering
Digital Object Identifier 10.4230/DagRep.7.7.44

1 Executive Summary

Sarah Zennou
Saumya K. Debray
Thomas Dullien
Arun Lakhotia

As a follow-up on the previous Dagstuhl Seminar 14241 on the analysis of binaries, the interest in attending this new seminar was very high. The attendance was very diverse, almost half academics and half practitioners.

Talks were arranged by topics and each day ended with an open discussion on one of the three topics: machine learning, obfuscation and practitioners' needs.

Considering the given talks, it appears that the challenges in the realm of general binary analysis have not changed considerably since the last gathering. However, the balance between the topics shows that the academic interest is now more focused on machine learning than on obfuscation. On the contrary practitioners exhibited examples showing that the sophistication level of obfuscations has tremendously increased during this last years.

The open discussions were the most fruitful part of the seminar. The discussions enabled the academics to ask practitioners about the hypotheses that are relevant to build models

Malware Analysis: From Large-Scale Data Triage to Targeted Attack Recognition, *Dagstuhl Reports*, Vol. 7, Issue 7, pp. 44–53
Editors: Sarah Zennou, Saumya K. Debray, Thomas Dullien, and Arun Lakhotia
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

for their analyses and the problems they face in their daily work. The practitioners gained awareness of the automated tools and techniques that they can expect to see emerge from research labs.

These informal exchanges will be gathered into a separate document and spread to the academic community.

Finally please note that not all people who presented have submitted their abstracts due to the sensitive nature of the content and/or the organization that the participants work for

2 Table of Contents

3 Overview of Talks

3.1 Characterizing the Strength of Code Obfuscation Against Auto-MATEd Attacks

Sebastian Banescu (BMW Group ITZ, DE)

Joint work of Sebastian Banescu, Christian Collberg, Vijay Ganesh, Zack Newsham, Alexander Pretschner
Main reference Sebastian Banescu, Christian Collberg, Vijay Ganesh, Zack Newsham, Alexander Pretschner, "Code obfuscation against symbolic execution attacks", in Proc. of the 32nd Annual Conf. on Computer Security Applications, pp. 189–200, ACM, 2016.
URL https://doi.org/10.1145/2991079.2991114

There exist several obfuscation transformations and there are many ways in which these can be combined. This talk presents a method that is meant to aid a software developer in deciding which obfuscation transformations to employ in order to protect his/her own software against known automated man-at-the-end (MATE) attacks, for a certain amount of time, against attackers having a given resource cap. A case-study based on a symbolic execution attack is used to instantiate this method and to show its utility.

3.2 Operation Avalanche: Not your average botnet take down

Thomas Barabosch (Fraunhofer FKIE – Bonn, DE)

The Avalanche network was one of the major cyber crime platforms offering services such as traffic obfuscation and money mule management. It hosted infamous malware families like Goznym, Matsnu and Urlzone. In late 2016, several international partners from law enforcement, industry and academia took down this network in a coordinated operation. Fraunhofer FKIE participated in this take down and carried out the technical analysis. I discuss the work we did prior to the take down, e.g. how we analyzed 130 TB of malware traffic, how we tracked 10+ malware families at once and what kind of obfuscations we faced in practice. This talk should be beneficial to participants since it gives insights into a four year long operation against a major cyber crime network.

3.3 Deobfuscation: semantic analysis to the rescue

Sébastien Bardin (CEA LIST, FR)

Joint work of Sébastien Bardin, Robin David, Jean-Yves Marion
Main reference Sébastien Bardin, Robin David, Jean-Yves Marion: "Backward-Bounded DSE: Targeting Infeasibility Questions on Obfuscated Codes", in Proc. of the 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017, pp. 633–651, IEEE Computer Society, 2017.
URL http://dx.doi.org/10.1109/SP.2017.36

Malware comprehension, i.e. deep malware analysis in order to understand their behavior, may be necessary in case of infection of a critical asset. Yet, in the case of targeted attack,

this analysis is highly challenging due to the strong obfuscation methods used for malware protection, and there is a clear need for more sophisticated and automated tools than currently available syntactic and dynamic approaches. In this talk, we present how semantic analysis coming from source-level safety analysis can be adapted to the context of binary-level deobfuscation, as well as their strengths and limitations.

3.4 How Professional Hackers Understand Protected Code while Performing Attack Tasks

Bjorn De Sutter (Ghent University, BE)

Joint work of Mariano Ceccato, Paolo Tonella, Cataldo Basile, Bart Coppens, Bjorn De Sutter, Paolo Falcarin, Marco Torchiano
Main reference Mariano Ceccato, Paolo Tonella, Cataldo Basile, Bart Coppens, Bjorn De Sutter, Paolo Falcarin, Marco Torchiano: "How professional hackers understand protected code while performing attack tasks", in Proc. of the 25th International Conference on Program Comprehension, ICPC 2017, Buenos Aires, Argentina, May 22-23, 2017, pp. 154–164, IEEE Computer Society, 2017.
URL http://dx.doi.org/10.1109/ICPC.2017.2

Code protections aim at blocking (or at least delaying) reverse engineering and tampering attacks to critical assets within programs. Knowing the way hackers understand protected code and perform attacks is important to achieve a stronger protection of the software assets, based on realistic assumptions about the hackers' behaviour. However, building such knowledge is difficult because hackers can hardly be involved in controlled experiments and empirical studies. The FP7 European project ASPIRE has given the project researchers the unique opportunity to have access to the professional penetration testers employed by the three industrial partners. In particular, we have been able to perform a qualitative analysis of three reports of professional penetration test performed on protected industrial code. Our qualitative analysis of the reports consists of open coding, carried out by 7 annotators and resulting in 459 annotations, followed by concept extraction and model inference. We identified the main activities: understanding, building attack, choosing and customizing tools, and working around or defeating protections. We built a model of how such activities take place. We used such models to identify a set of research directions for the creation of stronger code protections.

3.5 Similarity Analysis in Verona & IMDEA

Roberto Giacobazzi (University of Verona, IT), Mila Dalla Preda, and Niccolò Marastoni

We present the problems and main challenges in automated similarity analysis of high-level code. After an introduction to abstract interpretation and its precision as complete abstractions, we introduce the REHA, a tool for similarity analysis of Android applications. REHA scales well over large code. The talk ends with a discussion on the open questions and future developments in automated similarity analysis in malware detection and early threat detection.

3.6 The many Dimensions of Relationships

Tim Kornau-von Bock und Polach (Google Switzerland – Zürich, CH)

Joint work of Tim Kornau-von Bock und Polach, Bruno Montalto

Function based similarity, detections, challenges and open questions.

In this talk the current and past research challenges about executable and function similarity will be discussed. We will discuss scale of executable / function similarity search at Google and its applicability to malware. We will demonstrate use cases from an executable and function similarity perspective to show where such a system is relevant in practice. We will highlight some future directions of research and also demonstrate failures during the development of this system.

3.7 A morphological approach to detect code similarities and to analyse x86 binaries

Jean-Yves Marion (LORIA & INRIA – Nancy, FR)

Binary code analysis is a complex process which can be performed nowadays only by skilled cybersecurity experts whose workload just keeps increasing. Uses cases include vulnerabilities detection, testing, clustering and classification, malware analysis, etc... We develop a tool named Cyber-Detect, which is based on the reconstruction of an high level semantics for the binary code. Control flow graphs provide a fair level of abstraction to deal with the binary codes they represent. After applying some graph rewriting rules to normalize these graphs, our software tackles the subgraph search problem in a way which is both efficient and convenient for that kind of graphs. This technique is described as morphological analysis as it recognizes the whole shape of the malware. That being said, some pitfalls still need to be considered. First of all, the output can only get as good as the input data. And it is known that static disassembly cannot produce the perfect control flow graph since this problem is undecidable. As a matter of facts, malware heavily use obfuscation techniques such as opaque predicates to hide their payloads and confuse analyses. Dynamic analysis should then be used along with static disassembly to combine their strengths. Another dangerous pitfall feared by every expert is the so-called false positives rate : false alarms that make them waste indeed a precious time assessing the reality of the threat.

3.8 Side-Channel Based Detection of Malicious Software

J. Todd McDonald (University of South Alabama – Mobile, US)

Joint work of J. Todd McDonald, Joel A. Dawson, Patrick H. Luckett, Lee M. Hively

Timing and power are two potential indicators for the presence of malicious software execution. There is nascent research in using such indicators for training of intrusion or anomaly detection

systems. Of potential interest are non-linear methods that have a theoretic foundation and are adaptable to a wide range of anomaly detection problems, not just cyber. A nonlinear approach based on Taken's time delay embedding theorem has seen some early success in detecting rootkit execution. The approach relies on phasespace representation of the dynamics of a computer system and leverages graph-based difference to indicate a change in normal state. The technique is different from other traditional approaches that rely on statistical analysis methods based on machine learning and may provide a practical out-of-band approach for early indication of zero-day threats.

3.9 On the Availability of High-Quality Malware Data Sets

Daniel Plohmann (Fraunhofer FKIE – Bonn, DE)

In this presentation, the lack of availability of comprehensive, accurately labelled malware data sets is thematized. First, a typical sequence of steps an analyst may use during malware identification is outlined. This is then used to to further motivate the knowledge fragmentation that is reportedly perceived by many malware analysts. Next, a selection of popular malware data sets is quickly examined, showing that there is no coherent corpus focusing on unpacked malware samples.

We then present our approach for a manually curated, high-quality malware corpus. This corpus focuses on providing coverage for as many distinct families as possible, while limiting itself to single unpacked samples for a given variant. The method of collaborative collection and inventorization is explained, along with a short evaluation of the current contents.

3.10 Measuring and Defeating Anti-Instrumentation-Equipped Malware

Mario Polino (Polytechnic University of Milan, IT) and Stefano Zanero (Polytechnic University of Milan, IT)

Joint work of Mario Polino, Andrea Continella, Sebastiano Mariani, Stefano DÁlessio, Lorenzo Fontana, Fabio Gritti, and Stefano Zanero
Main reference Mario Polino, Andrea Continella, Sebastiano Mariani, Stefano D'Alessio, Lorenzo Fontana, Fabio Gritti, Stefano Zanero: "Measuring and Defeating Anti-Instrumentation-Equipped Malware", in Proc. of the Detection of Intrusions and Malware, and Vulnerability Assessment - 14th International Conference, DIMVA 2017, Bonn, Germany, July 6-7, 2017, Proceedings, Lecture Notes in Computer Science, Vol. 10327, pp. 73–96, Springer, 2017.
URL https://doi.org/10.1007/978-3-319-60876-1_4

Malware authors constantly develop new techniques in order to evade analysis systems. Previous works addressed attempts to evade analysis by means of anti-sandboxing and anti-virtualization techniques, for example proposing to run samples on bare-metal. However, state-of- the-art bare-metal tools fail to provide richness and completeness in the results of the analysis. In this context, Dynamic Binary Instrumentation (DBI) tools have become popular in the analysis of new malware samples because of the deep control they guarantee over the instrumented binary. As a consequence, malware authors developed new techniques,

called anti- instrumentation, aimed at detecting if a sample is being instrumented. We propose a practical approach to make DBI frameworks more stealthy and resilient against anti-instrumentation attacks. We studied the common techniques used by malware to detect the presence of a DBI tool, and we proposed a set of countermeasures to address them. We implemented our approach in Arancino, on top of the Intel Pin framework. Armed with it, we perform the first large-scale measurement of the anti-instrumentation techniques employed by modern malware. Finally, we leveraged our tool to implement a generic unpacker, showing some case studies of the anti- instrumentation techniques used by known packers.

3.11 Formally Proved Security of Assembly Code Against Power Analysis

Pablo Rauzy (University of Paris VIII, FR), Sylvain Guilley, and Zakaria Najm

Main reference Pablo Rauzy, Sylvain Guilley, Zakaria Najm: "Formally proved security of assembly code against power analysis - A case study on balanced logic", J. Cryptographic Engineering, Vol. 6(3), pp. 201–216, 2016.
URL http://dx.doi.org/10.1007/s13389-015-0105-2

In his keynote speech at CHES 2004, Kocher advocated that side-channel attacks were an illustration that formal cryptography was not as secure as it was believed because some assumptions (e.g., no auxiliary information is available during the computation) were not modeled. This failure is caused by formal methods' focus on models rather than implementations. In this paper we present formal methods and tools for designing protected code and proving its security against power analysis. These formal methods avoid the discrepancy between the model and the implementation by working on the latter rather than on a high-level model. Indeed, our methods allow us (a) to automatically insert a power balancing countermeasure directly at the assembly level, and to prove the correctness of the induced code transformation; and (b) to prove that the obtained code is balanced with regard to a reasonable leakage model, and we show how to characterize the hardware to use the resources which maximize the relevancy of the model. The tools implementing our methods are then demonstrated in a case study in which we generate a provably protected PRESENT implementation for an 8-bit AVR smartcard.

3.12 Advanced Semantics Based Binary Code Similarity Comparison

Dinghao Wu (Pennsylvania State University – State College, US)

Joint work of Dinghao Wu, Jiang Ming, Dongpeng Xu, Yufei Jiang

Binary code comparison has many applications in malware analysis and software engineering. Previous semantics-based work focuses on extracting and comparing of semantics of basic blocks. This block-centric method has limitations on optimizations and obfuscations that merge or split basic blocks. To address the limitations, I will present a method that uses Equivalence Checking of System Call Aligned Segments. With two sequences of instructions, obtained from two traces with the same input on two programs, we first align the system

calls involved. With two aligned system calls (one in each sequence, or trace), we slice back on the data and control dependences from the call sites on their parameters. After slicing, we get two trace segments with only instructions related to these two system calls. Then we compute their weakest preconditions, and compare their equivalence using a constraint solver. Our experiments show quite promising results.

3.13 Uroboros: Reassembleable Disassembling

Dinghao Wu (Pennsylvania State University – State College, US)

Joint work of Dinghao Wu, Shuai Wang, Pei Wang

There are many disassemblers, but no one is able to disassemble an executable in a way that can be reassembled back to an executable. Uroboros is a tool we built for Reassembleable Disassembling. In this talk, I will discuss the challenges and methods to disassemble an executable into an assembly that can be reassembled it back into an executable, in a fully automated manner, and present our results.

3.14 On The Unreasonable Effectiveness of Boolean SAT Solvers

Ed Zulkoski (University of Waterloo, CA)

Joint work of Ed Zulkoski, Vijay Ganesh, Jimmy Liang, Saeed Nejati, Zack Newsham

Modern conflict-driven clause-learning (CDCL) Boolean SAT solvers routinely solve very large industrial SAT instances in relatively short periods of time. This phenomenon has stumped both theoreticians and practitioners since Boolean satisfiability is an NP-complete problem widely believed to be intractable. It is clear that these solvers somehow exploit the structure of real-world instances. However, to-date there have been few results that precisely characterize this structure or shed any light on why these SAT solvers are so efficient.

In this talk, I will present results that provide a deeper empirical understanding of why CDCL SAT solvers are so efficient, which may eventually lead to a complexity-theoretic result. Our results can be divided into two parts. First, I will talk about structural parameters that can characterize industrial instances and shed light on why they are easier to solve even though they may contain millions of variables. Second, I will talk about internals of CDCL SAT solvers, and describe why they are particularly suited to solve industrial instances.

Participants

- Radoniaina Andriatsimandefitra
Rennes, FR
- Sebastian Banescu
BMW Group ITZ, DE
- Thomas Barabosch
Fraunhofer FKIE – Bonn, DE
- Sébastien Bardin
CEA LIST, FR
- Konstantin Berlin
SOPHOS – Fairfax, US
- Paul Black
Federation University Australia – Mount Helen, AU
- Cory Cohen
Carnegie Mellon University – Pittsburgh, US
- Christian Collberg
University of Arizona – Tucson, US
- Sophia D'Antoine
Trail of Bits Inc. – New York, US
- Mila Dalla Preda
University of Verona, IT
- Robin David
Quarkslab, FR
- Bjorn De Sutter
Ghent University, BE
- Saumya K. Debray
University of Arizona – Tucson, US
- Thomas Dullien
Google Switzerland – Zürich, CH
- Roberto Giacobazzi
University of Verona, IT
- Yuan Xiang Gu
Irdeto – Ottawa, CA
- Tim Kornau-von Bock und Polach
Google Switzerland – Zürich, CH
- Arun Lakhotia
University of Louisiana – Lafayette, US
- Colas Le Guernic
Direction Generale de l'Armement – Rennes, FR
- Jean-Yves Marion
LORIA & INRIA – Nancy, FR
- Marion Marschalek
G DATA Advanced Analytics GmbH – Bochum, DE
- J. Todd McDonald
University of South Alabama – Mobile, US
- Xavier Mehrenberger
Airbus – Suresnes, FR
- Michael Meier
Universität Bonn, DE
- Bogdan Mihaila
Synopsys Finland OY – Helsinki, FI
- Craig Miles
Assured Information Security – Portland, US
- Asuka Nakajima
NTT – Tokyo, JP
- Daniel Plohmann
Fraunhofer FKIE – Bonn, DE
- Mario Polino
Polytechnic University of Milan, IT
- Pablo Rauzy
University of Paris VIII, FR
- Raphael Rigo
Airbus – Suresnes, FR
- Radwan Shushane
Columbus State University, US
- Natalia Stakhanova
University of New Brunswick at Fredericton, CA
- Ryan Stortz
Trail of Bits Inc. – New York, US
- Dinghao Wu
Pennsylvania State University – State College, US
- Yves Younan
Cisco Systems Canada Co. – Toronto, CA
- Stefano Zanero
Polytechnic University of Milan, IT
- Sarah Zennou
Airbus – Suresnes, FR
- Ed Zulkoski
University of Waterloo, CA