TANSIVTx: Time-Accurate Network Simulation Interconnecting VMs with Hardware Virtualization towards stealth analysis

Executive summary: This project aims at confining malware samples for analysis in a secure environment while ensuring that these samples do not evade the analysis by detecting the environment. The specific challenge to address in this project is to support the analysis of malware samples that check the verisimilitude of network performances to detect analysis environments. To this end the project combines experiences from the Inria/Myriads team on simulation-based studies of distributed systems and from the DGA in cybersecurity and virtualization.

Advisors:

- Martin Quinson (ENS-Rennes, IRISA, team Myriads) Martin.Quinson@ens-rennes.fr
- Louis Rilling (DGA, team Myriads) Louis.Rilling@irisa.fr
- Matthieu Simonin (Inria Rennes) Matthieu.Simonin@inria.fr

Team: Myriads. **Laboratory:** IRISA, Rennes (head: Guillaume Gravier – guig@irisa.fr).

Required skills: Networks and VMs; System programming on Linux.
Appreciated but not mandated skills: Deep understanding of OSes; x86 assembly; Programming in C; Programming in Rust.

Context and Description

Malware analysts often rely on sandboxes to study malware and their interaction with the environment. In this context, the malware payload is executed in a virtual machine (VM) on top of a custom hypervisor. Various analysis tools can then safely analyze the malware execution from outside of the VM.

Attackers have however developed different evasion techniques, to detect sandboxes and hide their malicious behavior [1]. A class of evasion techniques relies on timing analysis. For instance, a malware can compare several time references to detect discrepancies that can be caused by the analysis environment. To the best of our knowledge, the currently known evasion techniques solely rely on comparing time references that are local to the VM (e.g. execution loops' timings). Some sandboxes ensure that all local time references remain consistent to achieve stealthiness with that regard [2, 3, 4].

The TANSIV project focuses on making sandboxes stealth with respect to timing-analysis-based evasion techniques using network interactions as part of their time references. Such evasions are easy enough to implement to be considered as real threats from the analyst perspective. In the general case, a malware targeting a specific victim could compare its network environment with a fingerprint embedding the knowledge of the victim environment, including the local network size, performance, and geo-location. Similarly, with no prior knowledge of the victim environment, the evasion decision could be taken based on the performance of the communication to its Command and Control (C&C) servers as the network latency must follow approximately a distribution known in advance.

To add timing-based network-fingerprinting resistance to sandboxes, the TANSIV approach consists in interconnecting a sandbox and the network end-points of its environment with a scalable, performance-accurate, discrete-event network simulator. TANSIV decouples the wall-clock execution time from the time perceived inside the sandbox and its network environment, as the simulator coordinates the progression of time on all network end-points and when communications are delivered to their target end-points.

In a first step, we have implemented a prototype of TANSIV with the Qemu PC emulator and the SimGrid simulator [5]. The emulation mode of Qemu especially allows TANSIV to precisely control the progression of time in the VM. However, although some sandboxes are based on Qemu [6, 7], the emulation mode has 3 major drawbacks:

- wall-clock execution speed (as perceived by the human analyst) is much slower than in a real environment;
- emulation mode is easily detected from inside the VM (e.g. it is not cycle-accurate) and is a clear sign of a sandboxed execution;
- many sandboxes rather use hardware-assisted virtualization [8, 9], both for better speed and stealthiness.

The goal of the TANSIVTx project is twofold. First, we should extend TANSIV to hardware-assisted virtualization, for which we must design solutions to precisely control the progression of time. Second, the added precise yet decoupled control of time may interfere with the time-related tooling of sandboxes. Therefore, the approach must be tailored to minimize the required changes on the sandbox tools.

The different approaches will be implemented in open-source sandboxes and experimentally evaluated, taking into account the achievable execution speed, the portability to new sandboxes, as well as the analysis power gained for analysts.

Bibliography

[1] A. Afianian, S. Niksefat, B. Sadeghiyan and D. Baptiste. *Malware Dynamic Analysis Evasion Techniques: A Survey*, ACM Computing Surveys 52(6), 2019.

[2] D. C. D'Elia, E. Coppa, F. Palmaro and L. Cavallaro. *On the Dissection of Evasive Malware*, IEEE Transactions on Information Forensics and Security, vol. 15, 2020.

[3] C. Kruegel. *Full System Emulation: Achieving Successful Automated Dynamic Analysis of Evasive Malware*, BlackHat 2014.

[4] T. Roccia and C. Shah. *Evolution of Malware Sandbox Evasion Tactics – A Retrospective Study*, 2019. https://www.mcafee.com/blogs/other-blogs/mcafee-labs/evolution-of-malware-sandbox-evasion-tactics-a-retrospective-study/

[5] H. Casanova, A. Giersch, A. Legrand, M. Quinson and F. Suter. *Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms*, Journal of Parallel and Distributed Computing 74(10), 2014. http://hal.inria.fr/hal-01017319.

[6] B. Dolan-Gavitt, J. Hodosh, P. Hulin, T. Leek and R. Whelan. *Repeatable Reverse Engineering with PANDA*, 5th Program Protection and Reverse Engineering Workshop, Los Angeles, California, December 2015. https://apps.dtic.mil/sti/pdfs/AD1034415.pdf

[7] https://github.com/Cisco-Talos/pyrebox

[8] T. Lengyel, S. Maresca, B. Payne, G. Webster, S. Vogl and A. Kiayias. *Scalability, Fidelity and Stealth in the DRAKVUF Dynamic Malware Analysis System*, 30th Annual Computer Security Applications Conference (ACSAC), 2014.

[9] https://github.com/thalium/icebox

[10] H. Tazaki, F. Urbani, E. Mancini, M. Lacage, D. Camara, T. Turletti and W. Dabbous. *Direct Code Execution: Revisiting Library OS Architecture for Reproducible Network Experiments*. 9th International Conference on emerging Networking EXperiments and Technologies (CoNEXT'13), 2013. https://hal.inria.fr/hal-00880870

[11] H. Lee, J. Seibert, E. Hoque, C. Killian and C. Nita-Rotaru. *Turret: A Platform for Automated Attack Finding in Unmodified Distributed System Implementations*, 34th International Conference on Distributed Computing Systems (ICDCS), 2014.