Automated discovery and analysis of code similarities in malware

Degree programme: Master of Science in Engineering | Specialisation: Information and Communication Technologies
Thesis advisor: Prof. Dr. Endre Bangerter
Expert: Paolo Palumbo (F-Secure)

Code reuse in malware is very common, as it lowers the effort of developing new versions or iterations. Such reuse gives malware writers an edge over analysts. The detection of code similarities and reuse can considerably simplify or even automate the analysis of malware. In this work, we have identified the principal steps and difficulties towards automating the discovery of code similarities, and present a system which is able to handle it.

System

The figure on the left provides an overview of the system we developed. In a first step, the malware's code gets automatically extracted using memory tracing, which is a technique of recording the memory changes of a software under execution. Memory tracing allows us to generically circumvent common obfuscation techniques like packing, encryption, multi-staged loading and injections to find malicious code, no matter when or where it appears. Furthermore, memory tracing is resilient against anti-analysis techniques, like debugger detection, system- or API call flooding, invalid instructions, CPU-hogging, etc.
After extraction, the code is disassembled, stored and indexed in a database. From there a clustering algorithm groups similar code together by comparing disassembled code functions. As malware analysis usually encompasses thousands of samples, the question of scalability is addressed by applying data deduplication, reducing computational complexities and retaining metadata to avoid recomputations. The metadata is also used to propagate knowledge about code between samples, thus can be used to automate some aspects of reverse engineering malware. Various tools of visualizing and exploring code similarities and malware relations were developed, providing an easy access to the data.

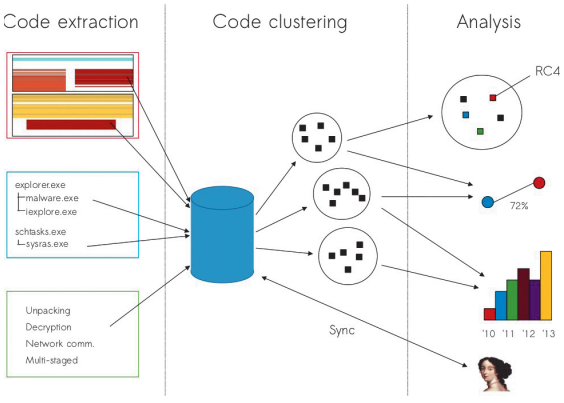
Results

We conducted a case study with CosmicDuke, a cyber-espionage malware, to evaluate the real-world applicability of the system. We analyzed the code similarities and reuse of samples that were active between 2010 and 2015. The image on the right shows the evolution of CosmicDuke's code. The vertical axes signify how much code has been reused from the previous samples, in some cases up to 80% of a sample's code is reused. We were able to show the modular design of CosmicDuke through the detection of code similarities, what parts changed infrequently over the years and how much of the code can be automatically analyzed with our system.

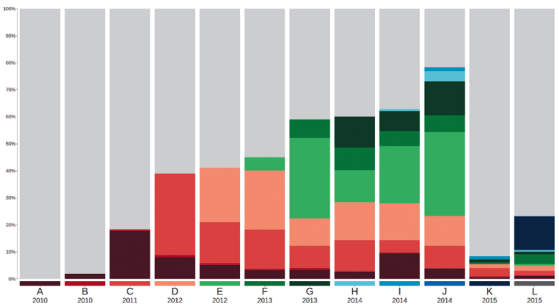
Jonas Wagner

Conclusion

The system we developed is able to automatically extract code from malware samples, while defeating common obfuscation and anti-analysis techniques. Code similarities are automatically revealed, thus significantly lowering the amount of manual labor to analyse malware samples or entire families. We were able to prove the real-world applicability with a case study about the six-year evolution of the CosmicDuke malware.



Overview of the system



CosmicDuke's code evolution over six years