AntiRE - An Executable Collection of Anti-Reversing Techniques

Daniel Plohmann and Christopher Kannen

Fraunhofer FKIE
D-53113 Bonn, Germany
⟨*plohmann*, *kannen*⟩{at}cs.uni-bonn.de

Reverse engineering is a methodology used to recover high-level information about structure and semantics of low-level analysis subjects. In the context of software, it is applied e.g. to establish understanding of embedded algorithms and protocols in order to achieve interoperability with software that is only available in compiled, binary form. This procedure is not always desired by creators of software as their products may contain intellectual property that they want to guard from competitors. Thus, protection schemes for binary code have been developed since decades.

Nowadays, such protection schemes are massively applied to malicious software (malware). The main reason is that the modifications performed on the binary code allow evasion from detection by security products. As a side effect, if the employed techniques or their effects are not known to malware analysts performing reverse engineering on malware samples, the analysis time can increase dramatically.

AntiRE is a collection of such anti analysis approaches, gathered from various sources like [1] and [2]. While these techniques by themselves are nothing new, we believe that their integration into a single, executable file provides a comprehensive overview, suitable for directly studying their behaviour in a harmless context without additional efforts. AntiRE includes different methods to detect or circumvent debuggers, fool execution tracing, and disable memory dumping. Furthermore, it can detect the presence of different virtualization environments and gives examples on how to twarth static analysis.

The techniques are implemented as isolated, minimal examples and accompanied by descriptions. Depending on their mode of operation, they are either treated as test or demonstration. A technique is defined as test if it yields information about presence of an analysis environment, in this case the result is returned. All test results are summarized at the end of execution. Otherwise, if the technique only serves as a demonstration, it does not influence the test statistics. To provide a better overview, the techniques are organized in groups ordered by similarity. When executed in a debugger, the tests and demonstrations can be easily accessed through a binary "table of contents", realized as a long sequence of commented function calls. When executed, our tool additionally generates output to a console window, showing the descriptions and optionally the test result. This can be used to quickly check an analysis environment for robustness against the included mechanisms.

AntiRE has been released as an open source project [3] and willingly accepts feedback and participation. The project will be extended continually.

References

[1] Peter Ferrie: *The "Ultimate" Anti-Debugging Reference*, 2011.
`http://pferrie.host22.com/papers/antidebug.pdf`

[2] Ange Albertini: *corkami.com - reverse engineering experiments and documentations*, 2011.
`http://www.corkami.com`

[3] Daniel Plohmann and Christopher Kannen: *AntiRE*, 2012.
`https://bitbucket.org/fkie_cd_dare/simplifire.antire`