Honeynet data capture - practical analysis of rootkits

Alexandre Dulaunoy

ASBL CSRRT-LU (Computer Security Research and Response Team Luxembourg)
http://www.csrrt.org/

January 22, 2010

Introduction

After the "basic" network analysis, you have now a better understanding of what happened in the network. Not everything is clear (it's impossible to reach the truth) but you are facing various part of the attacks and you would like to dig into the components used during the attacks. Some components seem to be software but how to make analysis ?

Analysis approach

There are two ways to analyze unknown components on a compromised system. The dynamic analysis approach and the static analysis approach. Static analysis is the most difficult way and takes a lot time to be properly done but the results can be very good. On the other side, the dynamic analysis is often faster to realize and give sometimes sufficient results but introduce more potential risks.

Dynamic analysis

Dynamic analysis means that you are analyzing software during its execution on a computer or in a virtual machine.

- "Black box approach". You are studying the component without digging the internal but only the flows (input/output) of the program, its interactions with the external interface (read/write access to files) or its effects on the environment (e.g. smart-card and power usage).
- "Post mortem approach". You are studying the effect on the system after the execution of the component. Effects can be memory effects, file access, temporary data created in the file system,...
- "Conventional approach". You are studying the execution of the component actively with system call tracker, memory analyzer, real time debugger/wrapper, open files tracker, ...

Static analysis

- "Classical static analysis" is the method to look at the component without any execution on the component itself. You can look a the string contained in the binary, compares the hashes fingerprint of the software,
- "Disassembly analysis" is to recover the machine-language code of a component. This can be a complex and long task to analyze large software but realivility small software can be disassembled.
- "Decompilation analysis" is to recover the source code of the compo- nent from machine-language. Sometimes compiler (from source code to machine-language) adds a lot of information in the compiled pro- gram.

Q and A

- Thanks for listening.
- a@foo.be