Anti-Anti-Sandbox - Notwendigkeit einer nicht detektierbaren Sandbox Umgebung

Dr. Björn Stelte
Stab Technische Innovation und Koordination
Bayerisches Landesamt für Verfassungsschutz
innovation@lfv.bayern.de

***Zusammenfassung*—Die Analyse von professionell entwickelten Schadprogrammen ist für die Abwehr von Angriffen essentiell. Mittels sogenannter Anti-Debug Techniken wird die Analyse jedoch zu einer zeitraubenden Angelegenheit. Da Zeit kostbar ist, werden häufig Sandbox-Umgebungen für eine erste Verhaltens-Analyse verwendet. Hierzu darf die Sandbox-Umgebung nicht als solche auffallen. Genau dieser Umstand wird jedoch häufig kaum beachtet.**

I. Einleitung

Die Analyse von Schadprogrammen ist ein wichtiger Bestandteil bei der Abwehr eines erkannten elektronischen Angriffs. Die Analyse mittels einer Sandbox-Umgebung verspricht hierbei wichtige erste Erkenntnisse, bspw. hinsichtlich der genutzten Rückkanalwege. Es ist allerdings zu beachten, dass das Verhalten eines Programms durch die System-Umgebung bestimmt sein kann. So kann bspw. mittels einer Erkennung einer Sandbox ein Programm die Ausführung des Schadcode-Programmteils unterdrücken und eine Analyse läuft ins Leere (vgl. [1]). Diese Technik nennt man Anti-Sandbox oder auch Anti-Forensik. Im nachfolgenden Beitrag sollen Überlegungen zur Abwehr der Anti-Sandbox Techniken vorgestellt werden, mit dem Ziel weiterhin eine schnelle Analyse von Schadprogrammen durchführen zu können bspw. ohne einen aufwendigen Reverse-Engineering Prozess zunächst zu durchlaufen.

II. Anti-Sandbox Methoden

Die Erkennung einer Sandbox Umgebung kann durch unterschiedliche Methoden erfolgen. Eine Liste von Möglichkeiten kann bspw. bei [2] gefunden werden. Zu analysierende Programme können neben verräterischen Einträgen bspw. im BIOS auch die Seriennummern einzelner Komponenten, wie etwa CPU, GPU, Speicherbausteine, etc., validieren und ihr Verhalten entsprechend der Umgebung steuern [3].

```
Handle 0x0001, DMI type 1, 27 bytes
System Information
        Manufacturer: HP
        Product Name: HP Pavilion Power Desktop 580-1xx
        Version:
        Serial Number: 4CE80
        UUID: 44FE5566-24AF-E5A4-7304-
        Wake-up Type: Power Switch
        SKU Number: 3ES41
        Family: 103C_53311M HP Pavilion
```

Abbildung 1. Auszug aus einer exemplarischen DMIDECODE Ausgabe.

Unter Linux sind mittels bekannter Programme, wie z. B. DMIDECODE, viele dieser Informationen einfach auslesbar. Bei [4] finden sich erste Überlegungen, wie eine VirtualBox Umgebung grob abgesichert werden kann. Weitergehende Möglichkeiten bestehen auch im Auslesen von angeschlossenen Geräten, wie dem angeschlossenen Monitor oder interner Geräte, wie etwa der Festplatte. Auch diese Komponenten müssen für einen ganzheitlichen Ansatz in Betracht gezogen werden.

III. Anti-Anti-Sandbox oder wie eine Sandbox aussehen muss

Gerade die bekannten Sandbox-Umgebungen werden oft auch von Entwicklern der Schadprogramme genutzt, um ihre Anti-Forensik Erkennungen zu schärfen. Ein Augenmerk der Forensiker muss daher eine Verbesserung der Sandbox-Umgebung hinsichtlich einer von einem normalen Rechners nicht unterscheidbaren Umgebung sein. Daher ist u.a. eine Möglichkeit glaubhafte Angaben zur Hardware zu erhalten notwendig, um in der Host-Forensik nicht blind zu werden. Viele Ansätze existieren bzgl. der Fragestellung, wie Anti-Debug und Anti-Sandbox Techniken detektiert werden können. Der Aspekt der Veränderung der Hardware-Angaben hinsichtlich einer wirksamen Abwehr der Anti-Forensik ist bislang kaum betrachtet worden. Kommerzielle Systeme existieren sehr wohl auf dem Markt, jedoch ist die Umsetzung der Unterbindung der Sandbox-Detektion teilweise eher dem Marketing zuzuschreiben.

IV. Zusammenfassung und Ausblick

Die erfolgreiche Abwehr elektronischer Angriffe kann nur durch eine schnelle und effiziente Abwehr erfolgen. Eine funktionierende Sandbox-Analyse ist daher ein wichtiger Baustein und sollte als solcher an die vorhandenen Anti-Sandbox Techniken angepasst werden.

Literatur

[1] Rudd, E. M., Rozsa, A., Günther, M., & Boult, T. E. (2017). A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions. IEEE Communications Surveys & Tutorials, 19(2), 1145-1172.
[2] https://www.cyberbit.com/blog/endpoint-security/anti-vm-and-anti-sandbox-explained/ (Stand 16.4.2018)
[3] Claud Xiao, Cong Zheng & Yanhui Jia (2017). New IoT/Linux Malware Targets DVRs, Forms Botnet. Unit 42 paloalto networks Report
[4] http://blog.michaelboman.org/2014/01/making-virtualbox-nearly-undetectable.html (Stand 16.4.2018)