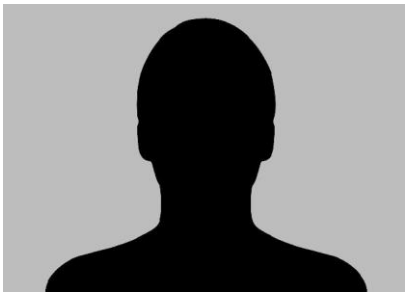
Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

InIT Institut für angewandte Informationstechnologie

Antivirus Software Attack Surface Security Analysis



Diplomand/in
Daniel Jampen

Dozent/in
Bernhard Tellenbach

In den diversen Cyber-Angriffen im letzten Jahrzehnt werden immer häufiger Schwachstellen in Antivirus Produkten ausgenützt. Die grosse Angriffsfläche und die hohen Berechtigungen, mit welchen diese Programme ausgeführt werden, machen sie zu einem attraktiven Ziel. Obwohl Sicherheitsforscher ihre Bemühungen zur Auffindung von Fehlern in solcher Software verstärkt haben, werden immer wieder neue Schwachstellen publik.

Die Knappheit der öffentlich verfügbaren Informationen über die Architektur und das Innenleben von Antivirus Produkten stellt eine weitere Hürde dar, welche Sicherheitsforscher überwinden müssen, um überhaut mit der Suche nach Schwachstellen beginnen zu können. Die Verfügbarkeit solcher Informationen würde das Bootstrapping von Forschungsvorhaben beschleunigen und schnellere und gezieltere Sicherheitstests ermöglichen.

In dieser Arbeit leisten wir mehrere Beiträge um diese Herausforderungen für die populärste Antivirus-Software-Engine, die mpengine von Windows Defender, anzugehen. Zuerst präsentieren wir eine Zusammenfassung von öffentlich verfügbaren Informationen, die wir für den Kontext der Schwachstellensuche als relevant erachten.Diese Zusammenfassung zeigt die in der Verganheit oft anfälligen Komponenten, die grundlegenden Arbeitsabläufe, wichtige Einstiegspunkte, Parser für Dateiformate und die Funktionen und Limitierungen des Binary Emulators sowie der JavaScript-Engine. Zweitens prüfen wir, ob diese Informationen noch auf die aktuellste Version von mpengine zutreffen und präsentieren zusätzliche, eigene Ergebnissen. Durch diese stellten wir fest, dass die meisten Informationen zwar immer noch gültig, jedoch ziemlich unvollständig sind - z.B. haben wir Software Code für 144 Dateiformat-Parser gefunden, welche bisher unbekannt warden.Anschliessend zeigen wir, dass mit diesen Ergebnissen ein Angreiffer die Signaturprüfung sowie den Emulator der mpengine mit Hilfe von Fingerprinting Techniken und einem, auf Verschlüsselung basierenden, Packer leicht umgehen kann.

Zum Abschluss geben wir Einblicke in unsere Forschungsarbeiten über mögliche Einstiegspunkte für ein auf Dateiformat-Parser ausgerichtetes Fuzzing Harness.

Um einen Antivirus Emulator zu umgehen, kann ein Fingerprinting der emulierten Umgebung verwendet werden. Die Abbildung zeigt die statische Liste der derzeit laufenden Prozesse, die einer emulierten Anwendung im Binary-Emulator von Windows Defender angezeigt wird. Mittels solchen Informationen kann eine Malware herauszufinden, ob sie in einem Antivirus-Emulator ausgeführt wird. Ist dies der Fall, ändert diese ihr Verhalten so, dass sie harmlos wirkt und verhindert damit die Entdeckung der Malware durch die Antivirus-Software.