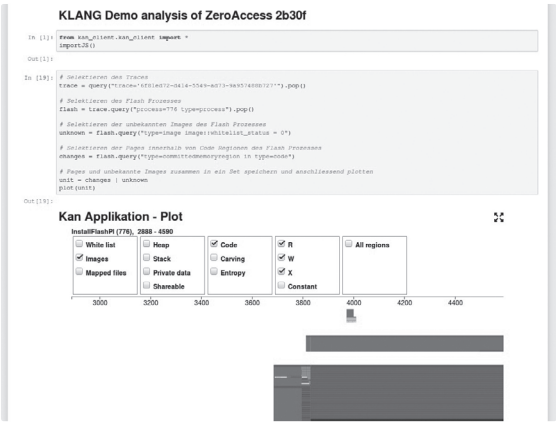
KAN Language: Interaktive, query-basierte Analyse von Malware

Studiengang: BSc in Informatik | Vertiefung: IT-Security
Betreuer: Prof. Dr. Endre Bangerter, Jonas Wagner
Experte: Dr. Igor Metz

Täglich entstehen neue Angriffsmuster unter dem Einsatz von Malware. Mit der Technologie des Memory Tracing wird die Funktionsweise solcher analysiert und dokumentiert. Im Rahmen dieser Thesis wurde eine kontextabhängige Abfragesprache entwickelt und implementiert, durch welche die Analyse effizienter, ergonomischer und nachvollziehbarer gemacht wird.

Ausgangslage

Das Security Engineering Lab der BFH entwickelt und betreibt die Memory Tracing-Applikation KAN. Beim Memory Tracing wird, im Gegensatz zur herkömmlichen Memory-Forensik, nicht nur ein einzelnes Abbild des Arbeitsspeichers analysiert, sondern dessen Zustandsänderungen über einen bestimmten Zeitabschnitt. Dabei werden in sehr kurzen Zeitabständen Abbilder des Arbeitsspeichers erstellt, um möglichst viele Zustände einer beobachteten Malware festzuhalten. Die Metadaten der erfassten Speicherabbilder werden anschliessend in eine Datenbank abgelegt und dienen zur Rekonstruktion der Aktivitäten der Prozesse. Diese Metadaten beinhalten unter anderem Informationen zum Typ und Inhalt von Memory-Regionen sowie Informationen zu den Prozessen, Threads und Systemcalls. Das Ziel des Analytikers ist es, anhand dieser Informationen die Funktionsweise der Malware zu verstehen.
Die Problematiken im aktuellen KAN-System sind die Performance-Einbussen durch den OR-Mapper, die Menge an Code, der für die Analyse geschrieben werden muss, sowie das Fehlen eines geeigneten Werkzeugs zu deren Dokumentation.



Ausschnitt einer Memory-Analyse mit der neuen KAN Language

Ziele

Die Ziele dieser Bachelor-Thesis waren die Implementation einer effizienten Sprache für die Datenabfrage über eine interaktive, web-fähige Shell sowie die Integration der Visualisierung von aufgezeichneten Memory-Regionen.

Ergebnisse

Als Lösung der bestehenden Probleme wurde die KAN Language implementiert. Diese Sprache ermöglicht es dem Analytiker die benötigten Daten gezielt und ohne Overhead aus der Datenbank zu selektieren. Die Auswahl erfolgt mittels sogenannter KAN-Entitäten wie Process, Heap, Stack, Code, Image oder Syscall und ihren jeweiligen Attributen. Eine Abfrage in einem KAN Language Query kann wie folgt aussehen:
query(«trace=›6f81ed72-d414-5549-ad73-9a957488b727› process=776 wlerror::snap_start›3800 wlerror::error_type‹››› image::whitelist_status=1 type=image type=wlerror»)
Der Aufruf des Parsers erfolgt über eine dazu entwickelte REST API. Im KAN Backend übersetzt der Parser anschliessend das KAN Language Query in ein gültiges SQL Select Statement, wobei die Zuordnung der Entitäten und ihrer Attribute zu den jeweiligen Tabellen und Spalten in der Datenbank gemacht wird. Um die Performance zu erhöhen, wurde für die Umwandlung des SQL Result Set zurück in KAN-Entitäten der KAN Entity Mapper geschrieben, welcher den bestehenden OR-Mapper ersetzt. Damit konnte die Performance, im Vergleich zur bisherigen Lösung, um den Faktor 17 verbessert werden.
Als interaktive, web-fähige Shell dient das Jupyter Notebook mit einem IPython Kernel, womit sich die Malwareanalyse auch elegant dokumentieren lässt.

Fazit

Die KAN Language und der KAN Entity Mapper verbessern Performance sowie Ergonomie gleichermassen. Einen grossen Mehrwert bieten auch die interaktiv bedienbaren Visualisierungen. Somit kann sich der Analytiker wieder voll und ganz auf die Auswertung konzentrieren.

Dominique Pascal Helfer

Patrick Schläpfer