Vortessence Rule Engine

Fachgebiet: Informatik - IT-Security
Betreuer: Prof. Dr. Endre Bangerter, Prof. Hansjürg Wenger
Experte: Dr. Igor Metz (Glue Software Engineering AG)

Gezielte Cyberattacken unter dem Einsatz von Malware nehmen stetig zu. Die Memory-Forensik als Mittel zur Erkennung solcher Angriffe wird daher immer wichtiger. Das Ziel des Projekts Vortessence ist es, diese Analysen zu vereinfachen, indem anhand einer Whitelist Anomalien in Speicherabbildern automatisch detektiert werden können. In dieser Thesis wurde eine Regelsprache entwickelt, in der die Regellogik der Whitelist in Form von Regel-Dateien beschrieben werden kann.

Ausgangslage

Memory-Forensik ist die Analyse von Arbeitsspeicher-Abbildern zur Detektion von Cyberattacken. Bisher erfordert es sehr viel Handarbeit und detaillierte Systemkenntnisse, um Malware mit einem Tool wie zum Beispiel dem quelloffenen Memory-Forensik-Framework Volatility zu finden. Vortessence ist ein Projekt des Security Engineering Labs der BFH, welches auf Volatility aufsetzt. Es automatisiert diese Analysen, indem es den Zustand «gesunder» Systeme in einer Whitelist beschreibt. Anhand dieser Whitelist findet Vortessence bei der Analyse Anomalien, wie z. B. unbekannte Prozesse, welche als Detektionen ausgegeben werden.

Der Workflow lässt sich wie folgt beschreiben: Mit Volatility werden Speicherabbilder analysiert. Die Ausgabe der Analyse wird anschliessend in die Datenbank von Vortessence eingelesen. Je nach Anwendungsfall werden, mit den daraus gewonnen Daten, die Whitelist aufgebaut oder durch Vergleich mit der bestehenden Whitelist Detektionen erstellt.

Die Problematik des bisherigen Aufbaus von Vortessence war, dass die Regellogik fest einprogrammiert war. Damit war es nur schwer möglich, ohne Kenntnisse vom internen Aufbau von Vortessence, neue Regeln zu schreiben sowie bestehende zu verstehen oder abzuändern.

Ziele

Das Ziel dieser Bachelor Thesis war es, für Vortessence eine Regelsprache zu entwickeln, in der die Regellogik beschrieben werden kann. Dafür musste eine Rule Engine entwickelt werden, welche diese Sprache interpretieren kann. Mit der Auftrennung von Regel- und Applikationslogik sollte es Memory-Forensik-Spezialisten einfach möglich sein, neue Regeln zu schreiben und/oder bestehende abzuändern.

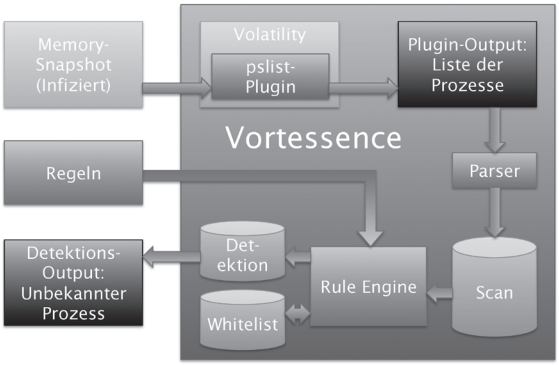
Ergebnisse

Als Regelsprache wurde eine Python-API definiert. Memory-Forensik-Spezialisten können damit Python-Module entwickeln, welche Regeln zur automatisierten Erkennung von Memory-Anomalien enthalten. In Vortessence wurde eine Rule Engine eingebaut, welche diese Regeln zur Laufzeit importieren und ausführen kann.

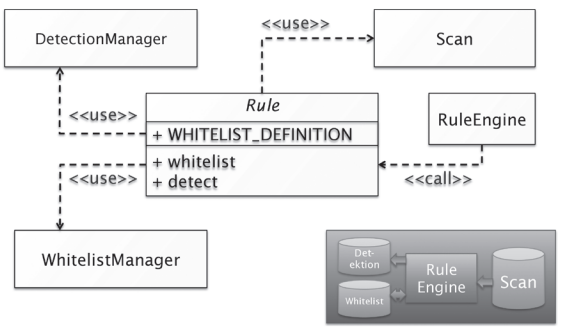
Die bestehende Regellogik wurde komplett in die neu entwickelte Regelsprache migriert. Mittels Praxis-Tests wurde gezeigt, dass die neue Regellogik gleichwertig mit der alten ist. Die API als Regelsprache bietet folglich die nötige Flexibilität, um Regeln für Vortessence zu beschreiben. Gleichzeitig wurde auch die Lesbarkeit der Regeln erhöht, so dass die Regellogik für Spezialisten leichter verständlich ist.

Christian Bürgi

Patrick Haring



Übersicht Aufbau und Workflow von Vortessence am Beispiel der Analyse von Prozessen.



Klassendiagramm der Python API zum Verfassen von Regeln für Vortessence