bb-it-boost

Hochschule für Technik und Wirtschaft Berlin
University of Applied Sciences

Sichere Ausführung beliebiger Analysealgorithmen in datenschutzkritischen Infrastrukturen

Michael Witt[a], Björn Lindequist[a], Peter Hufnagl[a,b], Dagmar Krefting[a]

a Hochschule für Technik und Wirtschaft Berlin, b Institut für Pathologie, Charité-Universitätsmedizin Berlin

Contact: m.witt@htw-berlin.de

Motivation

Moderne Biobanken besitzen Softwaresysteme, welche digitale Datensätze mit Probeninformationen enthalten und diese über eine Schnittstellen bereitstellen können.

Neben Standardinformationen wie Art, Beschaffenheit und Herkunft von Proben, können in modernen Biobank-Systemen auch komplexere Daten hinterlegt werden. Ein Beispiel hierfür sind virtuelle Schnitte (Abb. 1).

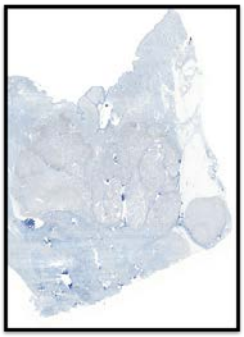

Abb. 1: virtueller Gewebeschnitt

Auf diesen digitalen Daten können beliebige Analyseverfahren angewendet werden, um neue Informationen zu generieren. Diese erweitern dann den Probendatensatz in der Biobank und sind anschließend jederzeit als Analyseergebnisse abrufbar (Abb. 2).

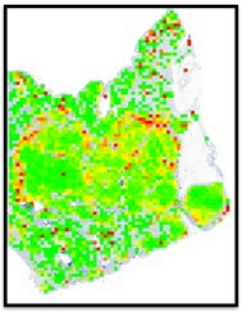

Abb. 2: Heat-Map nach Ki-67-Analyse

Eine flexible Infrastruktur ermöglicht es autorisierten Benutzern, eigene Analysealgorithmen zu übermitteln und diese auf den vorhandenen Daten auszuführen.
Medizinische Daten erfordern aufgrund der besonderen Anforderungen des Datenschutzes und ihrer Größe besondere Vorgehensweisen der Datenverarbeitung. Daher ist es z.T. erforderlich die Analyseverfahren in der selben Infrastruktur auszuführen, in der die Daten gespeichert sind. Die Ausführung unbekannter Algorithmen erfordert es jedoch, die Infrastruktur besonders gegen fehlerhafte oder schadhafte Aktionen abzusichern.

Sandboxes für Anwendungen

Moderne Betriebssysteme verfügen über Technologien, um anwendungsspezifische, abgesicherte Ausführungsumgebungen einzurichten. Diese werden als Sandbox bezeichnet. In einer Sandbox werden Zugriffe auf Ressourcen wie z.B. Dateien, Netzwerkverbindungen oder Datenbanken limitiert, um unberechtigten Zugriff zu verhindern.

Eine Möglichkeit, um eine Anwendung in der Sandbox zu überwachen, ist das Auswerten von System Calls (kurz SysCalls). SysCalls werden von einem Benutzer-Prozess verwendet, um auf Ressourcen zuzugreifen, welche vom Kernel des Betriebssystems verwaltet werden. Da genau diese Ressourcen geschützt werden müssen, stellen SysCalls eine geeignete Möglichkeit dar, um diese Anforderungen zu realisieren.

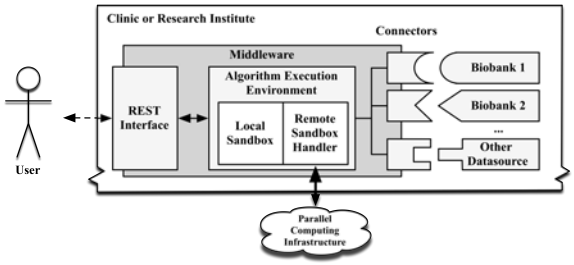

Abb. 3: Middleware mit Sandbox zur Ausführung von empfangenen Algorithmen

Überwachung von SysCalls

SysCalls arbeiten direkt an der Schnittstelle zwischen Anwendung und Betriebssystem. Über 300 SysCalls sind in modernen 64-Bit Linux-Systemen verfügbar. Die Formulierung von Regeln für eine SysCalls-basierte Sandbox ist deshalb komplex und fehleranfällig.

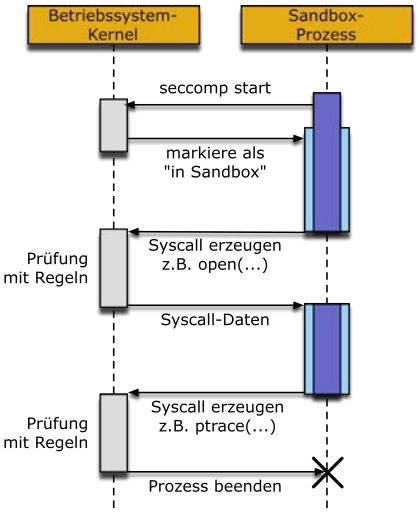

Abb. 4: Prozessüberwachung mittels SecComp

Aus diesem Grund ermöglicht es das hier vorgestellte Konzept, einen Prozess mittels pTrace zu überwachen. Damit werden alle SysCalls aufgezeichnet, und eine spätere Auswertung ermöglicht.
Diese Überwachung bedeutet erheblichen Mehraufwand bei der Ausführung und erhöht damit die Laufzeit.
Wurde jedoch der Prozess einmalig überwacht und zu beschränkende Sys-Calls identifiziert, ist eine Überwachung mit pTrace nicht länger notwendig.
Der Mehraufwand tritt somit nur einmalig auf.

Die Überwachung von Regeln im Kernel geschieht mit der SecComp-Technologie. Damit prüft der Kernel beim Bearbeiten von SysCalls die Regeln und beendet ggf. den Sandbox-Prozess bei einem erkannten Regel-Verstoß (Abb. 4).

Auswertung

Für die Analyse des Konzepts wurde eine Beispiel-Anwendung in einem OpenStack-Cluster ausgeführt.
Vier Modi zur Überwachung mit pTrace wurden gegenübergestellt. Ein Anstieg der Laufzeit auf bis zu 420% ist erkennbar (Abb. 5).
Während der Überwachung können pro Ausführung bis zu 780MB Daten entstehen.

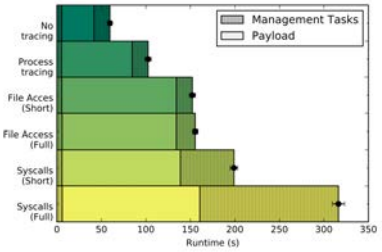

Abb. 5: Laufzeitvergleich Prozess-Überwachung

Es ist mittels automatisierter Verarbeitung möglich, mit diesen Informationen über verwendete SysCalls Regel-Sets für die SecComp-Sandbox zu erstellen.
Abb. 6 zeigt die Auswertung solcher Regel-Sets. Selbst bei komplexen Sandboxes aus über 2.000 Regeln ist kein signifikanter Laufzeitanstieg erkennbar.
SecComp eignet sich demnach für die Überwachung auf Ebene von SysCalls in Standard Linux-Systemen.

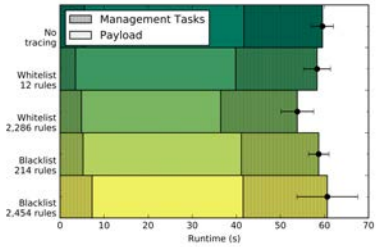

Abb. 6: Laufzeitvergleich Sandboxing