

Malware-Klassifizierung und -Clustering mit MIST

Philipp Trinius

Universität Mannheim, D-68159 Mannheim
philipp.trinius{at}informatik.uni-mannheim.de

Aktuell wird bei der Erkennung und Klassifizierung von Malware größtenteils auf signaturbasierte Verfahren zurückgegriffen. Die dabei eingesetzten Signaturen beschreiben lediglich Byte-Sequenzen von als böse eingestuftem Code und lassen sich daher relativ einfach mit Hilfe von polymorphem Code oder über *Code-Obfuscation* aushebeln. Der an der Universität Mannheim verfolgte und hier vorgestellte Ansatz zur Klassifizierung von Malware setzt daher keine klassischen Signaturen ein sondern versucht, allein auf Basis des Malware-Verhaltens eine zuverlässige Aussage zu treffen. Dazu wird die Malware in der CWSandbox ausgeführt und das beobachtete Verhalten protokolliert. Erste Versuche einer automatisierten Klassifizierung von Malware auf Basis dieser Reports brachte durchaus vielversprechende Ergebnisse [Rieck et al., DIMVA '08].

Um die maschinelle Verarbeitung der Daten zu optimieren und somit eine noch zuverlässigere Klassifizierung sowie *sauberes* Clustering der Malware zu erreichen, wurde MIST – Malware Instruction Set – entwickelt. Dabei handelt es sich um eine eigens zu diesem Zweck entwickelte Sprache, die das Verhalten von Malware abbilden kann. Die folgende Auflistung zeigt in (a) die CWSandbox- und in (b) die MIST- Codierung des API-Befehls zum Laden einer Bibliotheks-Datei.

(a) `<load_dll filename="H:\WINDOWS\system32\kernel32.dll" successful="1" address="$7C800000" end_address="$7C907000" size="1077248"/>`

(b) 02|02||00107000||02|0002|000011|0000||7C800000||7C907000||1||

In MIST entsprechen die einzelnen Attribute hexadezimalen Werten und werden durch eine doppelte Pipe voneinander getrennt. Während numerische Attribute, wie die Dateigröße, einfach hex-codiert übernommen werden, werden die Werte der übrigen Attribute, z.B. Dateinamen, in Tabellen eingetragen. In die MIST-Darstellung wird in diesem Fall lediglich der entsprechende Index übernommen. Neben der verkürzten Darstellung erfolgt eine Ordnung der Attribute. Dabei werden die Parameter der API-Befehle “entsprechend ihrer Konstanz” von links nach rechts angeordnet, wobei sehr variable oder für die Klassifizierung irrelevante Attribute vollständig entfallen oder ihre Werte ersetzt werden können. Beispielsweise werden Prozess-IDs durch den Dateinamen (inkl. Pfad) des Prozesses ersetzt. Für obiges MIST-Wort ergibt sich somit die folgende Sortierung: `load_dll|size|filename|address|end_address|successful`. Die Sortierung nach “Konstanz” wird auch für Pfadangaben enthaltende Attributwerte übernommen. Diese werden nach `extension|path|filename|paramter` sortiert. Das Ziel der Sortierung ist in beiden Fällen, verschiedene Granularitäten bei der Klassifizierung und dem Clustering der Reports zu ermöglichen.

Als Input werden zur Zeit CWSandbox-Reports in XML-Darstellung verwendet. Neben diesen Reports benötigt der Konvertierungsalgorithmus lediglich die dazugehörige Schemabeschreibung. Das MIST-Alphabet selbst baut sich erst während der Konvertierung der Reports auf, indem es beim Auftreten unbekannter Attributwerte um diese erweitert wird.

Um die Eignung von MIST zur Klassifizierung und dem Bilden von Malware-Familien zu bestimmen, wurden die MIST-Reports eines Test-Sets im ersten Schritt mit Hilfe der *Edit-Distance* und dem *Fuzzy hashing*-Algorithmus *ssdeep* untersucht. Im Vergleich mit Analysen auf den CWSandbox-Reports wurden die Malware-Binaries mit einem bedeutend höheren Prozentsatz ihren Familien zugeordnet, und auch der Abstand zwischen den einzelnen Familien war prozentual größer.