KASPERSKY lab

Lernfähige Systeme und menschliche Expertise

„Wie gewährleisten die fortschrittlichen Algorithmen von Kaspersky Lab bestmöglichen Schutz vor Cyberbedrohungen für Ihr Unternehmen?“

www.kaspersky.de

#truecybersecurity

Lernfähige Systeme und menschliche Expertise

Jedes Jahr gewinnt Kaspersky Lab in unabhängigen Tests viele Auszeichnungen im Bereich der Cybersicherheit. Unser HuMachine™-Ansatz ist ein zentraler Bestandteil unseres Erfolgs: Er kombiniert ein globales Cybergehirn, das von Big Data, Algorithmen und lernfähigen Systemen angetrieben wird, mit der unvergleichlichen Expertise unserer Sicherheitsteams, um Bedrohungen der nächsten Generation zu bekämpfen.

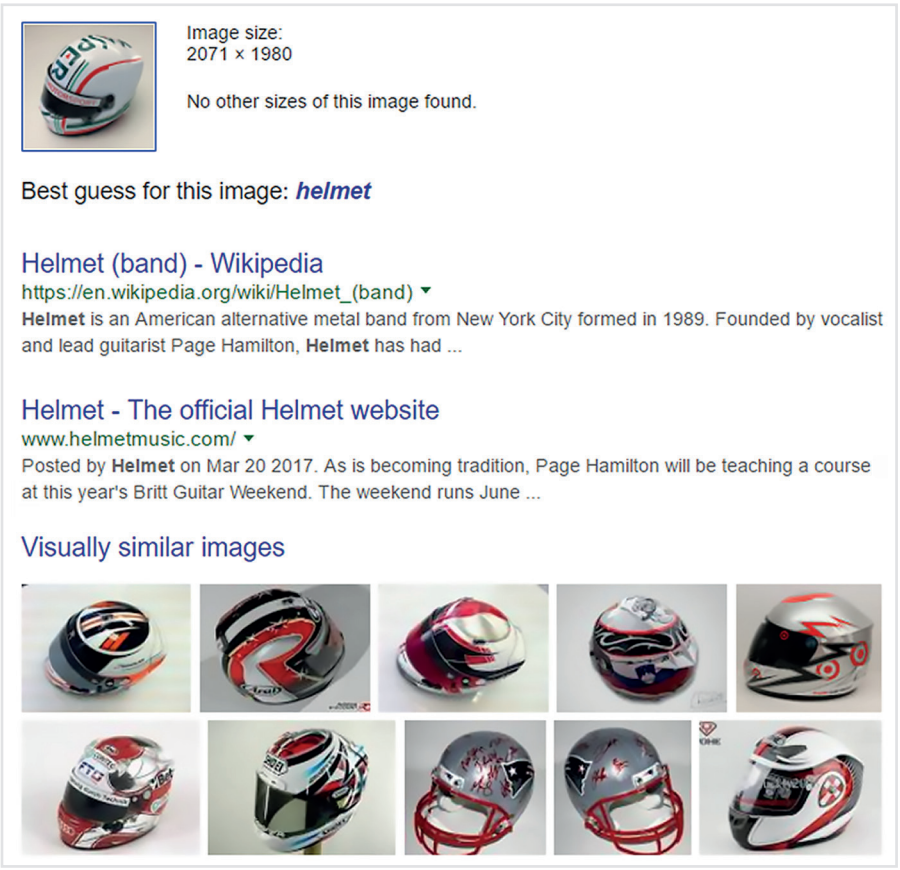
Wir bieten Ihnen einen Einblick in das Herz unserer Anti-Malware-Infrastruktur und stellen Ihnen hierbei unsere Algorithmen und deren Rolle bei der Verteidigung gegen die gefährlichsten Bedrohungen für Unternehmen wie Ihres vor.

„Wie gewährleisten die fortschrittlichen Algorithmen von Kaspersky Lab bestmöglichen Schutz vor Cyberbedrohungen für Ihr Unternehmen?“

Unser klassischer Ansatz für automatische Erkennung

Unsere Virendatenbank enthält Proben erkennbarer Bedrohungen nach Erkennungsnamen wie Backdoor.Win32.Hupigon.abc gruppiert. Wenn ein neuer, bisher unbekannter Virus erkannt wird, durchsuchen wir unsere Sammlung nach ähnlichen Proben. Dieses Suchprinzip ähnelt grob dem der Google-Suche. Der einzige Unterschied besteht darin, dass die Google-Suche textbasiert ist, während unsere Suche auf Dateimerkmalen basiert. Im einfachsten Szenario – wenn die Probe erfolgreich entpackt wurde – können wir die Strings extrahieren, die für die Malware-Funktionalität verantwortlich sind, und verwenden sie ähnlich wie eine Suchmaschine Stichwörter.

Bei Kaspersky Lab verfügen wir über ein automatisiertes System, das sowohl die Analyse der Dateien als auch die Klassifikation der Bedrohungen übernimmt.



Google-Service, der das Internet nach ähnlichen Bildern durchsucht

Dieses System sortiert die eingehenden Virenproben und fügt Hashes hinzu, um Erkennungen zu identifizieren und zu definieren. Ein einfacher Hash-Datensatz deckt zwar die Erkennung von nur einer Datei ab, jedoch lassen sich auf diese Weise Fehlalarme vermeiden.

Wenn Malware auftritt, die keiner der Proben in unserer Datenbank ähnelt, wissen wir, dass es sich entweder um etwas völlig Neues oder eben nicht um Malware handelt. Und hier kommt die Expertise unserer Virenanalysten ins Spiel. Wenn die Analysten eine Probe als Malware identifizieren, schaffen sie damit einen neuen Referenzwert, sodass künftig auch veränderte Versionen der Malware als solche erkannt werden können.

Heuristikbasierter Ansatz für automatische Erkennung

Eine rein Hash-basierte Erkennung ist nur die halbe Miete: nur eine minimale Dateiänderung (z. B. ein zusätzliches Byte am Ende), und schon ist die Datei nicht mehr erkennbar. Deshalb wenden wir auf die gesamte Familie der Malware-Proben, also z. B. Backdoor.Win32.Hupigon.abc, auch ein automatisches heuristisches Erkennungssystem an. Mithilfe eines Emulators erstellt das heuristikbasierte System Ausführungsprotokolle aller Proben, findet gemeinsame Ausführungsmuster und erstellt einen einzelnen ausführungsbasierten heuristischen Datensatz. Der Vorteil dieses Ansatzes ist es, dass neue Malware-Versionen, die ein ähnliches Verhalten aufweisen, erkannt werden, selbst wenn der Inhalt geändert wurde.

Sehen wir uns den Prozess, über den heuristische Datensätze erstellt werden, einmal näher an. Das automatische System nutzt maschinelles Lernen, um wichtige Ausführungssequenzen zu extrahieren. Es „weiß" hierbei nicht, welchen genauen Zweck die jeweiligen Befehlssequenzen haben. Das System soll nur erkennen, ob die untersuchte Ausführungssequenz – oder die Kombination von Sequenzen – einer Malware-Familie entspricht und dementsprechend in legitimen Dateien nicht auftreten sollte. Nach einigen Durchgängen werden die effektivsten Indikatoren und ihre Kombinationen automatisch in Datensätzen zusammengefasst.

Im Gegensatz zu diesem System versteht der menschliche Analyst exakt, was die Probe „im Schilde führt" – trotz ihrer Versuche, den Emulator des heuristischen Systems abzuschütteln. Nun kann der Analyst das verdächtige Verhalten der Probe umgehend dokumentieren.

Diese beiden Ansätze funktionieren parallel – insbesondere, wenn die Ergebnisse der automatischen Erkennung nicht eindeutig sind und die Zweitmeinung eines Experten erforderlich ist. Die automatischen sowie die von Menschenhand erstellten Datensätze arbeiten Hand in Hand und gewährleisten eine erfolgreiche Erkennung in perfekter HuMachine™-Harmonie.

```
KERNEL32!LoadLibrary(0x004020B6 "KERNEL32.dll");
KERNEL32!GetTickCount();
KERNEL32!LoadLibrary(0x00403000 "kernel32.dll");
KERNEL32!LoadLibrary(0x0040302C "urlmon.dll");
urlmon!URLDownloadTofile(,0x00403061 "http                                    ",0x004030C5
"c KERNEL32!Sleep()
KERNEL32!DeleteFile(0x004030C5 "c:\\boot.bak");
urlmon!URLDownloadTofile(,0x0040308F "http                                    ",0x004030B9
"c:\\4
```

Ausführungsprotokoll von „Trojan-Downloader.Win32.Small.aon"

Um der Erkennung zu entgehen, ändert der Cyberkriminelle möglicherweise die Funktionalität seiner Malware. Hierbei bestehen jedoch Einschränkungen. Gehen wir einmal davon aus, dass die Malware über grundlegende Funktionen verfügt: Sie soll eine Datei über einen schädlichen Link herunterladen, sie auf der Festplatte speichern und ausführen (Trojaner-Downloader). Es gibt maximal 10 Möglichkeiten, per Programmierung etwas aus dem Internet herunterzuladen, und sogar nur fünf, um eine ausführbare Datei zu starten. Wenn der Cyberkriminelle all diese Möglichkeiten ausgeschöpft und herausgefunden hat, dass jede Methode erkannt wird, bleibt ihm nicht viel übrig, als aufzugeben und sich für seinen Angriff ein Unternehmen zu suchen, das über keine Sicherheitslösung bzw. über eine Lösung ohne die nötigen Tools zur Ausführungsanalyse verfügt.

Möglicherweise hat er jedoch ein Ass im Ärmel: Wenn er die Details der Emulation kennt, kann er versuchen, den Emulationsprozess zu unterbrechen, indem er beispielsweise lange Ausführungsverzögerungen einfügt oder Systemparameter abfragt, die der Emulator nicht bereitstellen kann. Einige dieser Tricks stellen selbst Indikatoren für eine Erkennung dar, die Funktionalität einer Probe kann aber auch über eine weitere Methode ermittelt werden: den System Watcher. Dieses System überwacht die Aktivitäten von Prozessen im tatsächlichen Betriebssystem.

System Watcher und Verhaltenserkennung

Im Gegensatz zum Emulator ist der System Watcher ein echtes Verhaltenserkennungssystem, das auf Protokollen von realen Probenausführungen basiert. Das macht es unmöglich, das System auszutricksen. Es verfügt über eine eigene Sammlung von Verhaltensdatensätzen, die in vielen Punkten dem emulatorbasierten Erkennungssystem ähneln.

Die Protokollierung durch den System Watcher fällt jedoch deutlich breiter aus, als es während der Emulation möglich wäre. Und im Gegensatz zur Emulation findet diese Protokollierung dauerhaft statt: Jedes verdächtige Verhalten innerhalb eines bestimmten Kontextes wird berücksichtigt und zwischengespeichert, bis genügend Beweise für eine Erkennung gesammelt wurden. Wenn eine schädliche Aktivität erkannt wird, wird die entsprechende Aktion einfach rückgängig gemacht.

Wie auch das Emulationssystem spielt der System Watcher sowohl für die lokale Erkennung direkt beim Kunden als auch für unsere interne Laborarbeit eine wichtige Rolle. Der System Watcher arbeitet transparent und wirkt sich nicht nachteilig auf die überwachten Prozesse aus.

Eine dauerhafte lokale Verhaltensanalyse schafft eine äußerst leistungsstarke Erkennungsebene. Noch effektiver ist es jedoch, die gesamte Leistung der Kaspersky-Infrastruktur zu nutzen, um verdächtige Dateien auszuführen, ihr Verhalten zu untersuchen und die Bedrohungserkennung über das Kaspersky Security Network (KSN) vorzunehmen.

Sandboxes, KSN und Experten

Mit unserem HuMachine™-Ansatz testen wir ständig neue Proben – schädliche und unbekannte – in unseren internen Sandboxing-Systemen zur Verhaltensanalyse. Einige dieser Sandboxes imitieren Endbenutzersysteme mit Standardsoftware, während die leistungsstärksten Systeme über extrem fein abgestufte Protokollierungsfunktionen verfügen, die eine extrem präzise Erkennung ermöglichen.

Abfragen verdächtiger Aktivitäten als Anzeichen schädlichen Verhaltens

Die Sandbox-Protokolle und System Watcher-Ausführungsstatistiken, die wir von freiwilligen KSN-Teilnehmern erhalten, werden sowohl automatisiert als auch von Experten bearbeitet. Der automatisierte Teil übernimmt übernehmen hierbei zwei wichtige Prozesse: Die Protokolle der Ausführung neuer schädlicher Proben werden mithilfe lernfähiger Systeme untersucht, um neue Erkennungsindikatoren zu finden. Mithilfe statischer Protokolle werden auch bislang noch unbekannte Proben ermittelt. Selbst wenn der Entwickler der Malware also raffiniert genug ist, den Großteil der lokalen Erkennungsebenen zu umgehen – was im Normalfall nur über umfassende Untersuchungen und Vorabtests möglich ist –, bleibt er am Ende doch erfolglos.

In der Zwischenzeit nutzen Experten maschinell erstellte Indikatoren, um effektive Verhaltensdatensätze aufzubauen, die denen der emulierten Ausführung ähneln, aber eine deutlich breitere Auswahl an nutzbaren Indikatoren bieten.

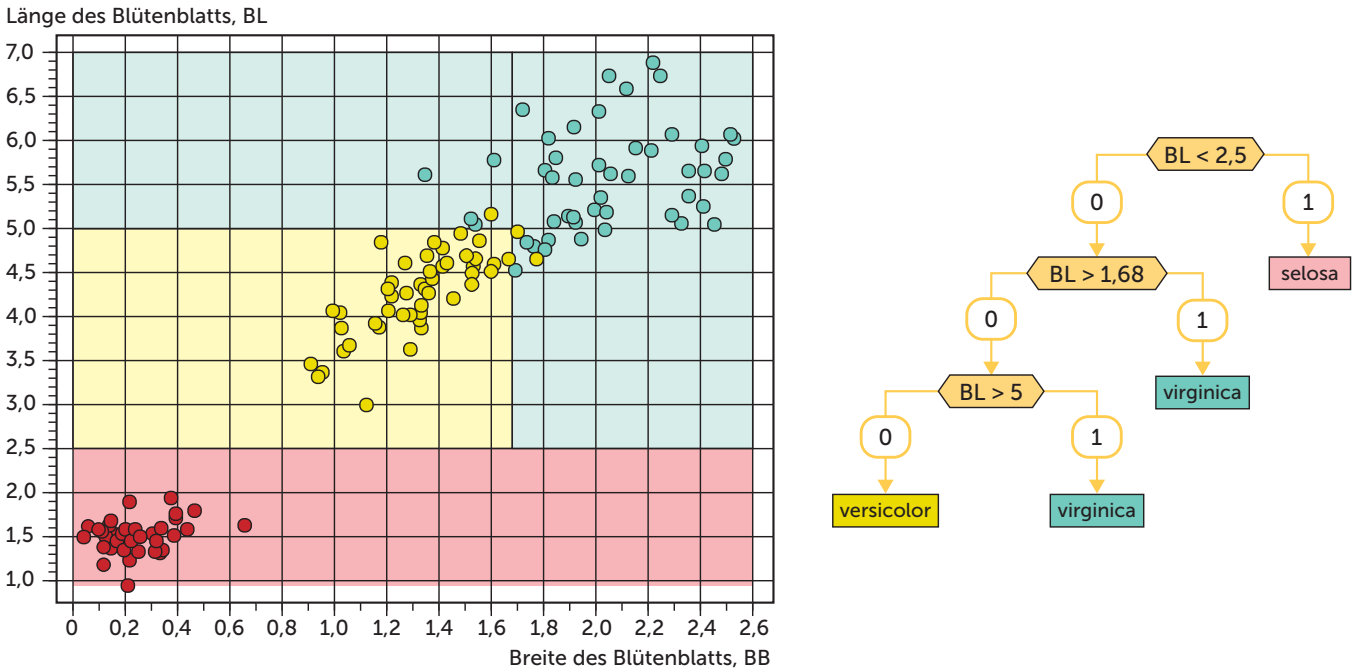
Smarte Datensätze

Neben den oben genannten Prozessen, die auf maschinellem Lernen basieren, sind noch weitere maschinelle Erkennungsebenen vorhanden, die verschiedenste Malware-Familien auffinden können. Diese Ebenen nennen wir „smarte Datensätze“.

Antiviren-Datensätze basierend auf Entscheidungsbäumen

Der laborinterne automatische Teil dieses Systems analysiert dieselbe, oben erwähnte Sammlung von Proben und erstellt oder verbessert die Datensätze mithilfe von Entscheidungsbäumen. So ist es möglich, Dateien in Klassen einzuteilen und Kriterien festzulegen, die den Merkmalen dieser Dateien entsprechen.

Wie funktioniert das? Sehen wir uns hierzu ein Beispiel an, das auf dem sogenannten Iris Flower Dataset, einem typischen Testfall für statistische Klassifikationstechniken, basiert. Gehen wir von 150 Blumen aus: jeweils 50 Proben der Iris setosa, der Iris virginica und der Iris versicolor. Um die Aufgabe zu vereinfachen, untersuchen wir die beiden aussagekräftigsten Merkmale dieser Blumen: die Länge (BL) und Breite (BB) des Blütenblatts. Durch die Untersuchung der Merkmale jeder Probe werden Daten zusammengetragen, die zur Erstellung eines Entscheidungsbaums genutzt werden können, der daraufhin jede neue Iris einer der drei Klassen zuordnen kann – über ein Frage-Antwort-Prinzip wie das folgende:



Im Diagramm: Achsen der beiden aussagekräftigsten Eigenschaften (von vier); zwei Klassen wurden präzise erkannt, während in der 3. Klasse drei Fehler auftraten.
Quellen: Coursera/Yandex

Unsere Antiviren-Engine nutzt exakt dieselbe Art von Entscheidungsbaum. Jeder Baum wird sorgfältig konstruiert und dem Benutzer bereitgestellt. Die ausgewählten Merkmale einer einzelnen Datei, die auf dem Computer des Endbenutzers ausgeführt wird, werden extrahiert und durchlaufen jeden Entscheidungsbaum. Der Baum verwendet die Antworten dann, um zu entscheiden, ob eine Datei schädlich ist.

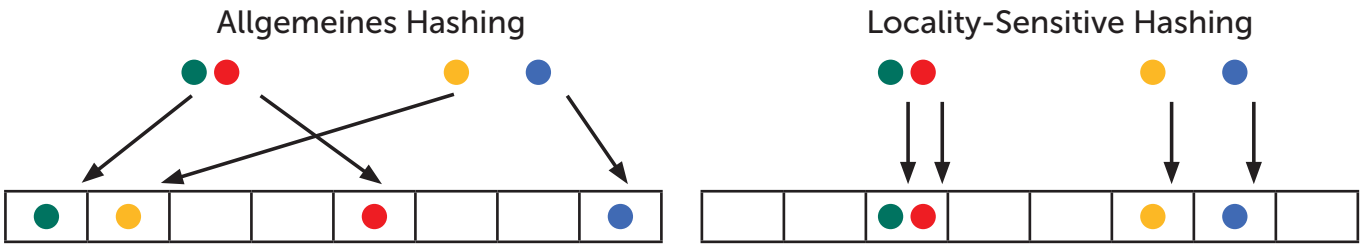
Der Vorteil dieses Ansatzes ist seine universelle Erkennung: Jeder Baum wird im Labor mithilfe einer kleinen Untergruppe von Proben erstellt, auf dem Computer des Endbenutzers erkennt der Baum jedoch auch Proben, die es bisher nicht in unser Labor geschafft haben. So wird beispielsweise im oben gezeigten Bild jeder Punkt im roten Bereich als Iris setosa erkannt. Ein einzelner Entscheidungsbaum ersetzt im Durchschnitt Tausende Hash-Datensätze.

Für die Erstellung von Entscheidungsbäumen ist maschinelles Lernen unerlässlich. Zwar können Experten den Systemen lange Listen von Merkmalen zuführen, sie können die baumbasierten Datensätze jedoch nicht selbst erstellen. Nur ein Automatismus kann die Daten extrahieren und anwenden, um die besten Merkmale auszuwählen und – was noch wichtiger ist – Entscheidungsregeln zu erstellen, die auf diesen Merkmalen basieren. Der Experte überwacht nur das Ergebnis und steuert den Prozess.

Locality-Sensitive Hashing

Modelle mit Entscheidungsbäumen sind zwar äußerst hilfreich, bieten jedoch einen Nachteil: Zwar werden sie im Labor automatisch erstellt, jedoch können sie nur auf dem Host (also dem Endbenutzer-PC), auf dem die jeweilige Datei untersucht wird, effektiv arbeiten. Ein Cloud-System, das auf diesem Prinzip basiert, würde beträchtlichen Netzwerkverkehr verursachen, den es in den meisten Fällen zu vermeiden gilt.

Hash-basierte Cloud-Systeme hingegen verursachen deutlich weniger Datenverkehr. Ein typischer kryptografischer Hash wie MD5 oder SHA-256 entspricht jedoch fast immer nur einer Datei. Das Ausbleiben einer zweiten Datei im selben Hash sorgt einerseits dafür, dass Fehlalarme ausgeschlossen sind. Andererseits wäre es hilfreich, über einen Hash zu verfügen, der für sämtliche Malware ein und derselben Familie identisch ist. So ließe sich der Hash nicht durch kleine Dateiänderungen umgehen. Und genau das funktioniert mit dem sogenannten Locality-Sensitive Hashing (LSH). Anfragen, die zu einer Erkennung mithilfe dieses Hash führen, können über die Cloud gesendet werden.



Die mehrfarbigen Punkte (Dateien) auf der linken Seite werden mit dem traditionellen Ansatz gehasht – die Hashes haben nichts gemeinsam. Auf der rechten Seite kommt der LSH-Hash zum Einsatz – so erhalten Dateien, die einander stark ähneln, identische Hashes.
Quellen: 0110.be

Wie berechnen wir das Maß an Ähnlichkeit zwischen den Dateien? Sehen wir uns hierzu folgendes Beispiel an:

Angenommen, Datei A wird durch folgende numerische Eigenschaften charakterisiert:

31, 83, 98, 86, 183, 79, 67, 153, 77, 67

Datei B weist demgegenüber leichte Unterschiede auf:

27, 89, 93, 81, 190, 71, 67, 161, 75, 69

Alle Zahlen können „abgerundet" werden, indem sie durch 10 geteilt werden. Dadurch ergibt sich Folgendes:

Datei A: 3, 8, 9, 8, 18, 7, 6, 15, 7, 6
Datei B: 2, 8, 9, 8, 19, 7, 6, 16, 7, 6

Wie Sie sehen, sind die Werte der einzelnen Merkmale jetzt nahezu identisch.

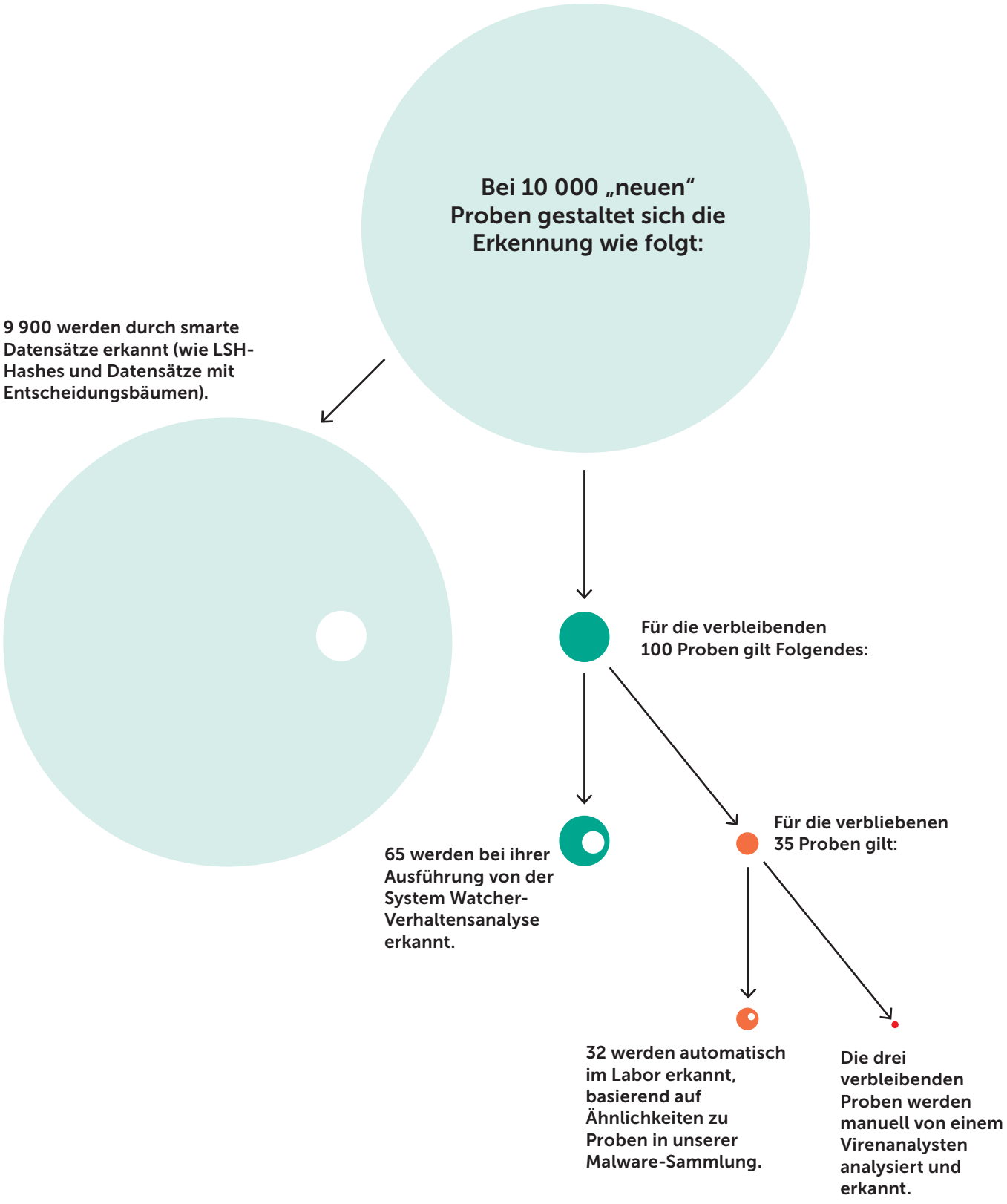
Hier ein anderer Ansatz: Wir berechnen das arithmetische Mittel der Zahlen in der ersten sowie in der zweiten Hälfte der beiden Dateien. Dadurch ergibt sich Folgendes:

Datei A: 96, 88
Datei B: 96, 88

In diesem Fall sind die LSH-Hashes identisch.

Die Herausforderung dieses Ansatzes besteht darin, Merkmale auszuwählen, die innerhalb einer Malware-Familie leicht abweichen, aber dennoch unterschiedlich genug sind, um in einer spezifischen sauberen Datei erkannt zu werden. Diese Merkmale müssen dann quantifiziert werden. Hierbei werden sie – vereinfacht ausgedrückt – verarbeitet, um ihre Präzision zu mindern. Auch das ist nur durch maschinelles Lernen und Automatismen möglich. Der Prozess wird jedoch von einem Experten eingestellt.

Der Pfad der Malware



Alle Proben werden unabhängig davon, ob sie es in die Sammlung geschafft haben, ständig neu analysiert, um neue Erkennungen zu ermöglichen. Hierbei kommen Verallgemeinerungstechnologien wie heuristische, baumbasierte und LSH-basierte Datensätze zum Einsatz. Wenn eine Probe zuvor nur über den individuellen Hash erkannt wurde, wird die Erkennung durch maschinelles Lernen verallgemeinert, um die Probe einer größeren „Familie“ von Malware zuzuordnen, die von einem einzelnen Datensatz beschrieben wird. Daraufhin wird der individuelle Hash-Datensatz gelöscht.

Fehlalarme vermeiden

Die Geschichte der heuristischen Erkennung, die auf maschinellem Lernen basiert, wäre nicht vollständig, ohne das Problem der Fehlalarme zu erwähnen. Wie bei jeder Methode, bei der das Prinzip der Verallgemeinerung zum Einsatz kommt, bergen diese Techniken das Risiko von Fehlern, die zu Fehlalarmen führen. Unerwartete Veränderungen der Bedrohungslandschaft können die Wahrscheinlichkeit solcher Fehlalarme erhöhen. Neben ständigen Anpassungen der Erkennungsmodelle ist also auch die dauerhafte Kontrolle über Fehlalarme erforderlich.

Kaspersky-Produkte beinhalten automatisierte Mechanismen für die Nachverfolgung, rechtzeitige Deaktivierung und Korrektur fehlerhafter Datensätze. Um unseren Kunden bestmögliche Ergebnisse zu ermöglichen, werden sämtliche Datensätze – auch die automatisch vom System erstellten – dauerhaft von den erfahrensten Analysten überwacht. Sie stellen sicher, dass die Datensätze regelmäßig gründlich getestet und angepasst werden, um bestmögliche Erkennungsraten zu gewährleisten und gleichzeitig die Anzahl von Fehlalarmen so gering wie möglich zu halten. Wie unabhängige Tests immer wieder beweisen, können wir das ziemlich gut.

Alle hierin beschriebenen Technologien und Ansätze bilden die Grundlage für das, was wir als „True Cybersecurity" bezeichnen. Und wir entwickeln unsere Technologien beständig weiter um auch der nächsten Generation von Bedrohungen einen Schritt voraus zu sein.

Informationen zur Internetsicherheit: **www.viruslist.de**
Informationen zu Partnern in Ihrer Nähe finden Sie hier:
http://www.kaspersky.com/de/partner_finden

www.kaspersky.de