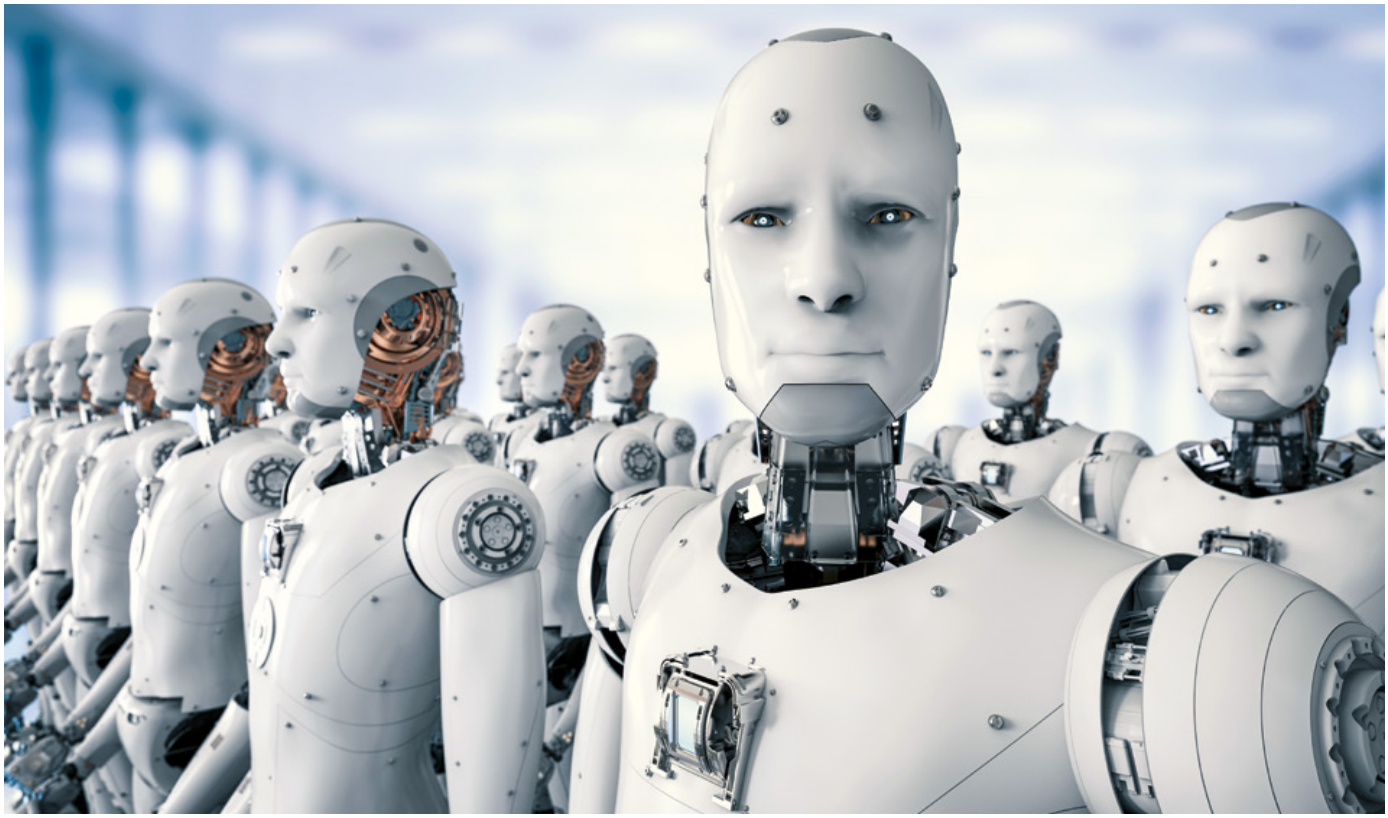
Die Kinder des Talos

AI gestützte IT-Sicherheitslösungen

Die Informationstechnologie hat schon zahlreiche Segen hervorgebracht. Der letzte große Sprung, Machine Learning (ML), hat die Kreativität in die Schaltkreise gebracht und damit neue Anwendungsfälle erschlossen, welche für die bisherige Software zu komplex sind. Aber wir werden noch regelmäßig mit den Schwächen der IT-Sicherheit konfrontiert. Könnte man nicht eine ML-Lösung entwickeln, die die IT-Sicherheit erhöht? Ja, kann man und macht man!



Heute befinden wir uns im Informationszeitalter und werden regelmäßig von den Fähigkeiten der Computer oder deren Software beeindruckt. Vor nicht allzu langer Zeit haben Menschen ihre Fähigkeiten in mental anspruchsvollen Spielen sehr hoch bewertet, Spiele in denen die besten Menschen nun von Computerprogrammen geschlagen wurden:

- Schach: Ein deterministisches Spiel, es ist im Grunde genommen mit Algebra lösbar.
- Jeopardy: Computer sind durch schnellen Zugriff auf riesige Datenmengen besser als die menschlichen Champions.
- Go: Dieses Spiel wird seit 2400 Jahren von vielen Menschen gespielt und perfektioniert.

Dank *Machine Learning* (ML) sind Computer nun kreativ und zeigen uns, wie wenig wir über dieses Spiel wissen.

Die Informationstechnologie hat schon zahlreichen Segen hervorgebracht, die unser Leben fundamental verändert haben. Verwaltungsaufgaben, Kommunikation und das Medienangebot von heute sind in allen Aspekten besser als vor dreißig Jahren. Erst der letzte große Sprung: ML hat die Kreativität in die Schaltkreise gebracht und damit neue Anwendungsfälle erschlossen, welche für die bisherige Software zu komplex sind.
Das wohl populärste Beispiel einer ML-Lösung ist die Suchfunktion von Google, dicht gefolgt von den Empfehlungen, die wir von Amazon, Netflix oder Facebook erhalten. Auch Microsoft und Baidu investieren Milliarden in diese Technologien – die Hälfte der zehn wertvollsten Unternehmen unserer Zeit sind führend in dem Bereich Künstliche Intelligenz (KI).
In Bild-, Sprach-, und Texterkennung sind computergestützte Lösungen nun besser als jemals zuvor. Autonom fahrende Autos, Züge und Flugzeuge werden in naher Zukunft fest in unserem Alltag integriert sein. Welche weiteren Aufgaben von Computersoftware übernommen werden, ist schwer einzugrenzen. Limitierende Faktoren scheinen hier nicht technologischer Natur zu sein. Wir stehen erst am Anfang, und nicht wenige Forscher versuchen bereits, die Singularität [Wiki-a] zu terminieren.
Diese Entwicklungen finden statt, während wir regelmäßig mit den Schwächen der IT-Sicherheit konfrontiert werden.

Daten von Unternehmen und Regierungen werden entwendet, die öffentliche Meinung gezielt manipuliert und Malware-Opfer erpresst. Daten sind die wichtigste Ressource unserer Zeit, sie sind begehrt und umkämpft.
Könnte man nicht eine ML-Lösung entwickeln, um die IT-Sicherheit zu erhöhen? Ja, kann man und macht man! Zahlreiche Forschungsergebnisse aus Universitäten, R&D-Abteilungen werden von alten und neuen Unternehmen genutzt, um neue Lösungen in die Gesellschaft zu tragen.
Im Folgenden werden einige dieser Lösungen genauer betrachtet. Im Fokus stehen diejenigen, welche heute tatsächlich nutzbar sind. Zu Beginn wird ein Forschungsergebnis genauer betrachtet und erläutert, um aufzuzeigen, wie ML funktioniert.

Aus der Forschung …

An der Universität Belfast wurde kürzlich ein neues Verfahren vorgestellt, welches mittels ML Schadsoftware (Malware) auf dem Betriebssystem Android identifiziert [Yer13]. Hierbei wurde Android als Zielsystem gewählt, weil es die größte Verbreitung hat und der potenzielle Einflussbereich der Lösung dementsprechend groß ist.
Schon vor dem Projekt gab es Ansätze zur Schadsoftware-Erkennung. Diese nutzen *on-device anomaly* oder *behavioral based detection*. In diesem Projekt wurde stattdessen eine *statische Analyse* auf Android Packages angewandt. Ein wesentlicher Vorteil dieses Vorgehens ist, dass die untersuchte Schadsoftware ihr Verhalten nicht erkennungsmechanisch zur Laufzeit anpassen kann.
Android-Anwendungen sind in der Programmiersprache Java geschrieben. Zur Analyse von Android-Anwendungen werden die Programme in einem ersten Schritt *reverse engineered*. Das Android SDK kompiliert den Sourcecode zusammen mit allen dazugehörigen Dateien in ein komprimiertes Android Package. Diese Archivdateien werden APKs genannt und haben die Endung .apk.
Mit dem Programm *Baksmali* [GitH] werden die APK-Dateien dekompiliert, innerhalb dieses Reverse-Engineering-Prozesses werden die Merkmale aus den ursprünglichen APK-Dateien ermittelt und festgehalten. Diese Merkmale sind z. B. API-Aufrufe, ausgeführte Linux-Systemkommandos und durch die App angefragte Zugriffsberechtigungen. Ebenfalls wird festgestellt, ob der Quellcode verschlüsselt ist oder ob das Archiv weitere APK- oder Java-Archiv-Dateien enthält.
Im nächsten Schritt werden die ermittelten Merkmale in einen Merkmalsvektor übertragen. Dieser enthält alle Merkmale in numerischer Form.

Abb. 1: Talos

Als Verfahren zur Klassifizierung der Anwendung wird der Bayes-Klassifikator angewendet. Der Bayes-Klassifikator ordnet jedes Objekt der Klasse zu, zu der es mit der größten Wahrscheinlichkeit gehört, oder bei der durch die Einordnung die wenigsten Kosten entstehen. Genau genommen handelt es sich um eine mathematische Funktion, die jedem Punkt eines Merkmalsraums [Wiki-b] eine Klasse zuordnet.
Für die Klassifizierung von Schadsoftware werden die beiden Klassen *Verdächtig* und *Gutartig* definiert.
Mit dem Bayes-Klassifikator kann nun für jedes Merkmal die Wahrscheinlichkeit bestimmt werden, welche besagt, zu welcher der beiden Klassen das Merkmal gehört. Die Merkmale können nun danach sortiert werden, mit welcher Wahrscheinlichkeit sie mit Schadsoftware in Verbindung stehen.

Talos

Talos (**siehe Abbildung 1**) ist ein bronzener Riese der griechischen Mythologie. Dreimal täglich umfliegt er die Insel Kreta. Wenn sich Piraten nähern, bewirft er deren Schiffe mit riesigen Felsen. Falls die Angreifer die Insel dennoch betreten, wird Talos ihnen zu Land begegnen.

Die Bewohner Kretas werden automatisch von dieser autonomen Instanz geschützt. Talos schafft Sicherheit.

Kasten 1

Nachdem die Rahmenbedingungen definiert wurden, wurde das neue Verfahren auf tausend unterschiedliche Malware-Beispiele angewandt, welche in 49 Kategorien eingeteilt wurden. Als Kontrollgruppe wurden weitere tausend Android-Apps untersucht, bei welchen es sich um ganz normale Software aus dem App-Store handelt.
Die Ergebnisse der Untersuchung zeigen, dass 15 bis 20 Merkmale ausreichend sind, um 90 Prozent der Schadsoftware richtig zu erkennen.
Dieser ML-Ansatz stellte sich in der Untersuchung als Verfahren mit dem besten Ergebnis für das Erkennen von Schadsoftware heraus. Ein weiterer Vorteil dieses Verfahrens war die Möglichkeit zur Identifizierung von Merkmalen, welche – in den Trainingsdaten – ausschließlich von Schadsoftware benutzt wurden. Zu solchen Merkmalen zählten unter anderem das Vorhandensein von *embedded Jars*, der Zugriff auf Informationen wie *DeviceID* oder die *SIM-Seriennummer* sowie das Ausführen von *pm install*.
Ein nicht weiter untersuchter Vorteil des Einsatzes des Bayes-Klassifikators ist die Verwendung von Expertenwissen. Sicherheitsexperten könnten durch die Analyse der ermittelten Eigenschaften womöglich weitere Informationen gewinnen.
Statt des Bayes-Klassifikators könnten auch andere ML-Verfahren auf die erhobenen Daten angewandt werden. Anbieten würde sich für die Klassifizierung unter anderem eine *Support-Vector-Machine*, *Neuronale Netzwerke* und *Entscheidungsbäume*.
Diese Lösung sollte es in Zukunft einfacher gestalten, Android Malware zu identifizieren, wobei die Idee auch auf andere Betriebssysteme übertragbar ist. Wahrscheinlich werden weitere Lösungen dieser Art folgen, denn ML eignet sich hervorragend, um Malware zu identifizieren.

DeepInstinct

Ein Start-up-Unternehmen aus Tel Aviv bietet eine proprietäre *Deep Learning*-Lösung an, um Malware auf Endgeräten in Unternehmensnetzwerken zu identifizieren oder sie zu stoppen. DeepInstinct [DI] besteht aus drei Teilen:

- Einem *Cloud-Modul* in der Firmenzentrale des Anbieters. Dieses sammelt kontinuierlich neue Informationen zu Cyberthreads und verteilt diese an die Kunden.
- Einem *Servermodul*, welches beim Kunden installiert ist. Dieses ist eine Schnittstelle zwischen dem Cloud-Mo-

dul und den Clients im Unternehmensnetzwerk. Es bezieht die aktuellsten Informationen aus dem Cloud-Modul und wendet eine kundenspezifische Policy an, um die Clients zu verwalten.
- Das letzte Modul wird direkt auf die zu schützenden Endgeräte installiert. Es nutzt die Informationen der beiden anderen Module, um Malware auf den Endgeräten zu identifizieren oder um zu verhindern, dass Malware Schäden verursacht – auch wenn das befallene Endgerät offline ist.

Diese Lösung konkurriert direkt mit den bisherigen Angeboten der Antiviren-Softwarehersteller, aber auch diese haben die Zeichen der Zeit erkannt und versuchen in verschiedenen Bereichen, ihre Lösungen mittels ML zu verbessern. *DeepInstinct* ist auch nicht auf Malware für einzelne Betriebssysteme reduziert, sondern verfolgt einen ganzheitlichen Ansatz.

IBM Cognitive SOC

IBM entwickelt unter dem Namen Watson [Wiki-c] diverse AI-Lösungen, darunter *Cognitive SOC*. Diese Lösung soll den Benutzern dabei helfen, schneller an sicherheitsrelevante Neuigkeiten zu gelangen.
Sicherheitskritische Informationen sind sehr heterogen im Internet verteilt, man findet sie in Whitepapern, Foren, Artikel, Logdateien usw. IBM verwendet Deep Learning zur Texterkennung, um sicherheitsrelevante Informationen im Internet zu finden, sie aufzubereiten und den Kunden auf einem zentralen Dashboard bereitzustellen.
Diese Lösung unterstützt IT-Sicherheitsexperten dabei, schneller an relevante Informationen zu gelangen. Hierdurch können diese schneller und gezielter auf Gefahren reagieren und ihren Job noch besser machen.

Apache Spot

Von Apache gibt es ein neues Incubator-Projekt, welches wesentlich von Intel und Coursera gefördert wird: *Spot* [ASpot]. Dieses Projekt hat das Ziel, Angriffe oder Anomalien im Netzwerkverkehr zu identifizieren, um sie gegebenenfalls zu verhindern.
Um dies zu erreichen, werden innerhalb eines Netzwerks die Telemetrie-Daten von unterschiedlichen Netzwerkkomponenten oder Anwendungen zentral in einer Hadoop-Instanz abgelegt (**siehe Abbildung 2**).
Diese Daten werden anschließend mittels ML analysiert, hierbei wird nach verdächtigen Datenpaketen oder Verbindungen gesucht. Die Analyseergebnisse werden dann auf einem zentralen Dashboard dargestellt. Neben den dort bereitgestellten Analysewerkzeugen gibt es auch Möglichkeiten, den Netzwerkverkehr zu begrenzen.
Falls tatsächlich schadhafter Netzwerkverkehr identifiziert wurde, können die neuen Erkenntnisse auf einer zentralen Plattform geteilt werden, um die Angriffsfläche der Malware auch außerhalb des eigenen Netzwerks weiter zu reduzieren. Die Rahmenbedingungen der Analyse sind für jede Instanz frei konfigurierbar.
Schon vorher gab es zahlreiche Ansätze, die dieses oder ähnliche Probleme gelöst haben. Hier werden die Technologien *Big Data* und *Machine Learning* injiziert, um bessere Ergebnisse zu erzielen.

Spamassasin

Spamassasin [SpamA] ist ein Klassiker, der seit 2004 unsere E-Mail-Postfächer vor unerwünschtem Spam schützt. Um besser zwischen erwünschten E-Mails und Spam zu unterscheiden, wird ML eingesetzt. Verwendet wird ein *Bayesscher Filter*, welcher nach der Installation trainiert wird. Ausschlaggebend für die Bewertung sind hierbei die in der E-Mail enthaltenen Wörter. Dieser Anwendungsfall ist für ML sehr naheliegend, daher wird er von zahlreichen weiteren E-Mail-Programmen verwendet.

Cleverhans

Die neuen ML-Lösungen bringen auch neue Schwächen mit sich. Cleverhans ist ein Projekt, welches einen Blog sowie eine Open-Source-Bibliothek beinhaltet. Im Blog [CH] werden Sicherheitsschwachstellen und Schutzmechanismen erläutert. Die bereitgestellte Bibliothek stellt Werkzeuge und Beispiele zur Verfügung, um diese Schwachstellen zu testen oder aufzuzeigen. Die Schutzziele lassen sich, wie gewohnt, mit dem klassischen CIA-Modell (Confidentiality, Integrity, Availibility) abbilden und sind sowohl während der Entwicklungsphase als auch im Betrieb relevant.
Cleverhans stellt ein Werkzeug bereit, mit welchem man Bilder so manipulieren kann, dass die Bilderkennungsprogramme unerwartete Ergebnisse liefern. In einem ihrer Beispiele wird ein Bild mit einem Panda gezeigt. Dies hat die geteste-

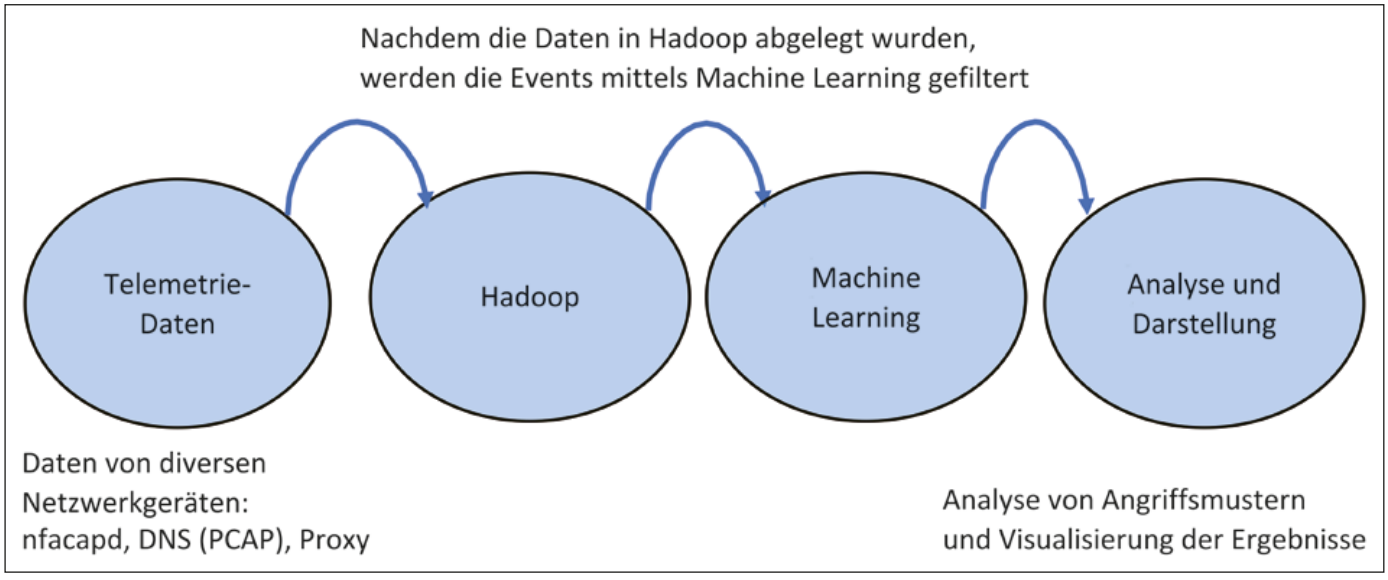


Abb. 2: Funktionsweise von Apache Spot

te Software mit einer Sicherheit von 57,7 Prozent ebenso erkannt. In diesem Bild wurden anschließend einige Pixel verändert. Für das menschliche Auge sind diese Änderungen nicht sichtbar, die getestete Software hat nun jedoch mit einer Sicherheit von 99,3 Prozent Gibbons auf dem Bild erkannt.

Die Schwächen von ML-gestützter Bilderkennung werden von Cleverhans gezielt ausgenutzt.

Dies sind sehr aufregende Neuigkeiten, wenn man die potenziellen Anwendungsfälle der neuen Bilderkennungslösungen berücksichtigt: Autonom fahrende Autos und Gesichtserkennung zur Terrorbekämpfung, um zwei der brisanteren zu nennen.

Leider ist dieses Sicherheitsrisiko nicht das einzige. Dass weitere folgen werden, ist abzusehen.

Fazit

ML-Lösungen haben qualitative und sicherheitsrelevante Qualitätsaspekte, die bei der Entwicklung berücksichtigt und getestet werden müssen. In den neuen Sicherheitslösungen können Risiken dieser Art eine schwer kalkulierbare Dynamik annehmen.

Machine Learning ist die neue Technologie, welche die IT-Landschaft von morgen maßgeblich formen wird. Sie ist da. Sie wird bleiben und unser Leben verändern. Kreative Hacker werden sie nutzen, umgehen und angreifen. IT-Sicherheitsexperten müssen sie nutzen und meistern, um den neuen Gefahren begegnen zu können. Softwaretester werden ihre Fähigkeiten erweitern müssen, um die Qualität der neuen Lösungen zu prüfen.

Die Kinder des Talos sind angekommen, sie sind jung und haben Schwächen, sie haben das Spiel verändert und sie kämpfen an der Seite ihrer Meister. Heißt sie willkommen, sie sollen unser Segen sein!

Die Autoren:
Toni Gansel (Sogeti), Sven Podlech (aiso-lab)

Literatur & Links

[ASpot] Apache Spot, siehe: **http://spot.incubator.apache.org/**

[CH] Cleverhans-Blog von I. Goodfellow, N. Papernot, siehe: **http://www.cleverhans.io/**

[DI] DeepInstinct, siehe: **https://www.deepinstinct.com/**

[GitH] smali/baksamli, siehe: **https://github.com/JesusFreke/smali/wiki**

[SpamA] SpamAssasin's Bayesian Classifier, siehe: **https://spamassassin.apache.org/full/3.0.x/dist/doc/sa-learn.html**

[Wiki-a] https://de.wikipedia.org/wiki/Singularität_(Systemtheorie)

[Wiki-b] https://de.wikipedia.org/wiki/Merkmalsraum

[Wiki-c] https://de.wikipedia.org/wiki/Watson_(Künstliche_Intelligenz)

[Yer13] S. Y. Yerima, S. Sezer, G. McWilliams, I. Muttik, A New Android Malware Detection Approach Using Bayesian Classification, in: IEEE 27. Int. Conf. AINA, 2013, siehe: **https://arxiv.org/ftp/arxiv/papers/1608/1608.00848.pdf**