(19) Deutsches Patent- und Markenamt

(10) **DE 10 2011 077 611 A1** 2012.12.20

(12) # Offenlegungsschrift

(21) Aktenzeichen: **10 2011 077 611.7**
(22) Anmeldetag: **16.06.2011**
(43) Offenlegungstag: **20.12.2012**

(51) Int Cl.: ***G06F 21/00*** (2011.01)

(71) Anmelder:
Universität Bremen, 28359, Bremen, DE

(74) Vertreter:
Fink Numrich Patentanwälte, 80634, München, DE

(72) Erfinder:
Elfers, Carsten, Dipl.-Inf., 28215, Bremen, DE; Birkholz, Henk, Dipl.-Inf., 28219, Bremen, DE; Edelkamp, Stefan, Dr., 28205, Bremen, DE; Sohr, Karsten, Dr., 28359, Bremen, DE

(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:

LANE, T.; BRODLEY, C. E.: Temporal Sequence Learning and Data Reduction for Anomaly Detection. ACM Transactions on Information and System Security, Vol. 2, August 1999, No. 3, S. 295 – 331. URL: http://www.aaai.org/Papers/FLAIRS/2005/Flairs05-107.pdf [recherchiert am 05.06.2012]

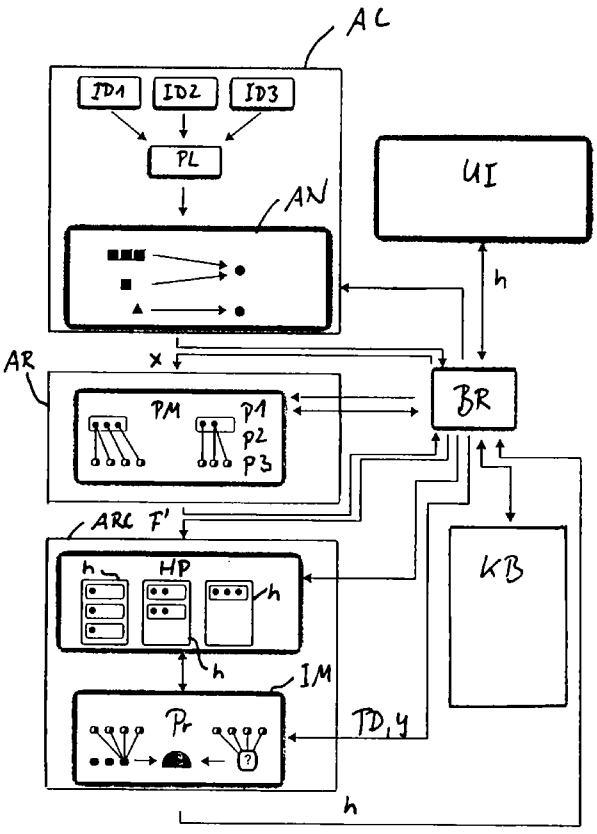
WONG, A.K.Y. [et al.]: Similarity and Logic Based Ontology Mapping for Security Management. Proceedings of the 8th International Florida Intelligence Research Society Conference, 2005. URL: http://www.aaai.org/Papers/FLAIRS/2005/Flairs05-107.pdf [recherchiert am 05.06.2012]

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Verfahren zum rechnergestützten Erkennen von Angriffen auf ein Computernetz**

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren zum rechnergestützten Erkennen von Angriffen auf ein Computernetz. In diesem Verfahren werden Sequenzen (x) von einer oder mehreren Beobachtungen im Computernetz detektiert. Anschließend wird eine jeweilige Sequenz (x) mit Mustern (p1, p2, p3) von jeweils einer oder mehreren semantischen Aussagen (y^1_1, y^1_2, y^1_3) aus einer ontologischen Wissensbasis (KB) verglichen, wobei für jedes Muster ein oder mehrere Ähnlichkeitsmaße (F') des Musters (p1, p2, p3) mit einer oder mehreren Gruppen von Beobachtungen aus der jeweiligen Sequenz (x) ermittelt wird. Für eine jeweilige Sequenz (x) werden dann basierend auf den Ähnlichkeitsmaßen (F') der Muster (x) eine oder mehrere Wahrscheinlichkeitsverteilungen (Pr) für eine Mehrzahl von vorgegebenen Angriffen (y) ermittelt, wobei eine jeweilige Wahrscheinlichkeitsverteilung (Pr) die Wahrscheinlichkeiten des Vorliegens der vorgegebenen Angriffe (y) für eine Gruppe von Beobachtungen der jeweiligen Sequenz (x) repräsentiert, wobei die Wahrscheinlichkeitsverteilung oder Wahrscheinlichkeitsverteilungen (Pr) auf einem probabilistischen Modell basieren, das mittels Trainingsdaten (TD) trainiert ist, gemäß denen Muster (p1, p2, p3) aus der Wissensbasis (KB) mit vorgegebenen Angriffen (y) korreliert werden.



Beschreibung

[0001] Die Erfindung betrifft ein Verfahren und ein System zum rechnergestützten Erkennen von Angriffen auf ein Computernetz.

[0002] Für die zuverlässige Detektion von Angriffen auf Computernetzen ist es aus dem Stand der Technik bekannt, sog. IDS-Systeme (IDS = Intrusion Detection System) zu verwenden. Dabei wird über den Einsatz verschiedener Arten von Software-Sensoren versucht, möglichst viele unterschiedliche Angriffstypen zuverlässig zu erkennen. Es sind anomalie-basierte IDS-Systeme bekannt, welche bisher unbekannte Angriffe detektieren, jedoch zu einer hohen Rate von Fehlalarmen für detektierte Beobachtungen führen, bei denen es sich nicht um einen Angriff handelt. Im Stand der Technik werden ferner signatur-basierte IDS-Systeme beschrieben, welche zwar nur bekannte Angriffe detektieren können, jedoch zuverlässiger dahingehend sind, dass seltener Fehlalarme ausgegeben werden. IDS-Systeme unterscheiden sich ferner in Bezug auf ihren Detektionshorizont. Netzbasierte Detektions-Sensoren überwachen die gesamten Leitungen in einem Computernetz, wohingegen host-basierte Detektions-Sensoren einzelne Computersysteme in Bezug auf Auffälligkeiten, z. B. im Arbeitsspeicher und im Dateisystem, überwachen. Der Betrieb verschiedener heterogener IDS-Systeme in einem Computernetz erfordert einen hohen Integrations- und Betriebsaufwand und erschwert es einem Benutzer, die Vielzahl von unterschiedlichen Meldungen von potentiellen Angriffen richtig zu werten und entsprechende Gegenmaßnahmen zur Abwehr der Angriffe einzuleiten.

[0003] In der Druckschrift [1] ist eine Einrichtung offenbart, mit der das Verhalten von Nutzern einer Datenverarbeitungseinheit unter Verwendung eines semantischen Netzes analysiert wird. In dem Dokument [2] wird ein rechnergestütztes Verfahren beschrieben, bei dem Ereignisse in einem Computernetz miteinander korreliert werden, um hierdurch Angriffe auf das Computernetz zu erkennen.

[0004] Aufgabe der Erfindung ist es, Angriffe auf ein Computernetz zuverlässig mit einem rechnergestützten Verfahren zu erkennen.

[0005] Diese Aufgabe wird durch das Verfahren gemäß Patentanspruch 1 bzw. das System gemäß Patentanspruch 13 gelöst. Weiterbildungen der Erfindung sind in den abhängigen Ansprüchen definiert.

[0006] Im Rahmen des erfindungsgemäßen Verfahrens werden in einem Schritt a) Sequenzen von einer oder mehreren Beobachtungen bzw. Ereignissen im Computernetz detektiert. Die Sequenzen können dabei verschiedene Kombinationen von zeitlich zurückliegenden Beobachtungen (inklusive der aktuellen Beobachtung) umfassen, wobei sich die Sequenzen auch in der Anzahl der darin enthaltenen Beobachtungen unterscheiden können. Die Detektion der Beobachtungen erfolgt mit an sich bekannten IDS-Systemen, wie z. B. Snort, oder einer Kombination aus mehreren IDS-Systemen. Es sind dabei viele verschiedene Ereignissen bzw. Beobachtungen bekannt, welche detektiert werden können. Ereignisse bzw. Beobachtungen können sich beispielsweise auf TCP-Scans im Computernetz, auf Pings, auf fehlerhafte Logins und dgl. beziehen.

[0007] In einem Schritt b) des Verfahrens wird eine jeweilige Sequenz mit Muster von jeweils einer oder mehreren semantischen Aussagen aus einer ontologischen Wissensbasis verglichen, wobei für jedes Muster eine oder mehrere Ähnlichkeitsmaße des Musters mit einer oder mehreren Gruppen von Beobachtungen aus der jeweiligen Sequenz ermittelt wird. Vorzugsweise wird dabei für jede Beobachtung aus der Sequenz eine Gruppe von Beobachtungen gebildet, welche die jeweilige Beobachtung und alle zeitlich zurückliegenden Beobachtungen aus der Sequenz umfasst. Eine Gruppe von Beobachtungen kann ggf. auch eine einzelne Beobachtung umfassen.

[0008] Erfindungsgemäß wird das Verhalten des Computernetzes basierend auf einer Ontologie beschrieben, wobei Ontologien semantisches Wissen über Begrifflichkeiten und deren Relationen in geeigneter Weise in digitalisierter Form repräsentieren. Ontologien sind im Bereich der Informatik an sich bekannt und sie ermöglichen, über Schlussfolgern mit einem geeigneten Reasoner semantisches Wissen abzuleiten. Im Rahmen des Vergleichs der Gruppen von Beobachtungen mit einem Muster wird dabei überprüft, ob die entsprechenden semantischen Aussagen des Musters, welche auch als Bedingungen bzw. Constraints bezeichnet werden, auf die Beobachtungen der entsprechenden Gruppe zutreffen, was gleichbedeutend damit ist, dass das Muster mit der Gruppe von Beobachtungen übereinstimmt. Erfindungsgemäß wird nunmehr jedoch nicht rein auf Übereinstimmung und Nicht-Übereinstimmung geprüft, sondern es erfolgt ein toleranter Mustervergleich, mit dem auch nur eine teilweise Übereinstimmung ermittelt werden kann, was durch ein entsprechendes Ähnlichkeitsmaß repräsentiert wird. Das maximale Ähnlichkeitsmaß steht dabei für eine Übereinstimmung und das

minimale Ähnlichkeitsmaß für keine Übereinstimmung. Alle dazwischenliegenden Werte zeigen eine teilweise Übereinstimmung an.

[0009] Die in Schritt b) ermittelten Ähnlichkeitsmaße werden in Schritt c) des erfindungsgemäßen Verfahrens in geeigneter Weise weiterverarbeitet. Dabei werden für eine jeweilige Sequenz basierend auf den Ähnlichkeitsmaßen der Muster eine oder mehrere Wahrscheinlichkeitsverteilungen für eine Mehrzahl von vorgegebenen Angriffen ermittelt, wobei eine jeweilige Wahrscheinlichkeitsverteilung die Wahrscheinlichkeiten des Vorliegens der vorgegebenen Angriffe für eine Gruppe von Beobachtungen der jeweiligen Sequenz repräsentiert. Die Wahrscheinlichkeitsverteilung oder Wahrscheinlichkeitsverteilungen basieren dabei auf einem probabilistischen Modell, das mittels Trainingsdaten trainiert ist, gemäß denen zumindest ein Teil der Muster aus der Wissensbasis mit den vorgegebenen Angriffen korreliert werden. Korrelation bedeutet dabei, dass für entsprechende vorgegebene Angriffe festgelegt wird, ob das Muster als übereinstimmend mit dem Angriff bzw. nicht übereinstimmend mit dem Angriff gewertet wird. Gegebenenfalls können die Trainingsdaten auch so ausgestaltet sein, dass eine teilweise Übereinstimmung festgelegt wird. Über die Trainingsdaten, welche z. B. durch Expertenwissen modelliert sind, fließt somit beim Trainieren des probabilistischen Modells Wissen über die Zuordnung entsprechender Muster zu Angriffen ein. Der Begriff des Angriffs ist hier und im Folgenden weit zu verstehen. Er umfasst neben bösartigen Angriffen unbefugter Dritter ggf. auch gutartige Angriffe, welche nicht Angriffe im eigentlichen Sinn sind, sondern ein normales Verhalten des Computernetzes repräsentieren.

[0010] Die Verwendung eines probabilistischen Modells, mit dem entsprechende Wahrscheinlichkeitsverteilungen beschrieben werden, weist den Vorteil auf, dass das Verfahren lernfähig wird, da das Modell ggf. mit neuen Trainingsdaten in regelmäßigen Abständen wieder neu gelernt werden kann und somit die Detektion von Angriffen verbessert werden kann. Über die durch das erfindungsgemäße Verfahren ermittelten Wahrscheinlichkeitsverteilungen, welche z. B. über eine Benutzerschnittstelle ausgegeben werden können, bekommt ein Benutzer eine zuverlässige Rückmeldung, wie wahrscheinlich bestimmte Angriffe für die detektierte Sequenz von Beobachtungen bzw. Gruppen von Beobachtungen aus der Sequenz sind. Gegebenenfalls können die ermittelten Wahrscheinlichkeitsverteilungen auch weiterverarbeitet werden, wie weiter unten anhand der Ausgabe von Hypothesen mit entsprechenden Prioritätswerten beschrieben wird.

[0011] Probabilistische Modelle zur Beschreibung entsprechender Wahrscheinlichkeitsverteilungen sind an sich aus dem Stand der Technik bekannt. In einer besonders bevorzugten Ausführungsform werden dabei Conditional Random Fields verwendet, mit denen sich sehr gut Sequenzen von Beobachtungen beschreiben lassen. Conditional Random Fields beschreiben die Daten, aus denen sie generiert werden, unter der Verwendung einer Exponentialfunktion. In der detaillierten Beschreibung werden Conditional Random Fields nochmals näher erläutert.

[0012] In einer besonders bevorzugten Ausführungsform des erfindungsgemäßen Verfahrens werden die Ähnlichkeitsmaße in Schritt b) derart bestimmt, dass zunächst ermittelt wird, ob ein jeweiliges Muster aus der ontologischen Wissensbasis mit der entsprechenden Gruppe von Beobachtungen der jeweiligen Sequenz (komplett) übereinstimmt. Diese Ermittlung kann in geeigneter Weise mit auf der Wissensbasis arbeitenden Reasonern, wie z. B. Pellet, erreicht werden. Im Falle, dass das jeweilige Muster mit der Gruppe von Beobachtungen übereinstimmt, wird diesem Muster das maximale Ähnlichkeitsmaß mit der Gruppe von Beobachtungen (d. h. Übereinstimmung) zugeordnet. Im Falle, dass das jeweilige Muster nicht mit der Gruppe von Beobachtungen übereinstimmt, werden die semantischen Aussagen im Muster schrittweise zu allgemeineren Aussagen abstrahiert, wobei nach jedem Abstraktionsschritt die Gruppe von Beobachtungen mit dem aus dem Abstraktionsschritt resultierenden abstrahierten Muster verglichen wird. Im Falle, dass das abstrahierte Muster mit der Gruppe von Beobachtungen übereinstimmt, wird ein Wert einer Ähnlichkeitsfunktion, der sich für das abstrahierte Muster ergibt, mit dem Ähnlichkeitsmaß des jeweiligen Musters gleichgesetzt. Dabei ist der Wert der Ähnlichkeitsfunktion umso geringer, je allgemeiner das abstrahierte Muster ist. Die Allgemeinheit eines Musters wird durch die Anzahl und Allgemeinheit der darin enthaltenen semantischen Aussagen bestimmt. In der detaillierten Beschreibung wird ein Beispiel für eine im erfindungsgemäßen Verfahren verwendbare Ähnlichkeitsfunktion beschrieben. Im Falle, dass über die Abstraktionsschritte kein abstrahiertes Muster gefunden werden kann, welches mit der Gruppe von Beobachtungen übereinstimmt, wird dem jeweiligen Muster ein minimales Ähnlichkeitsmaß mit der Gruppe von Beobachtungen (d. h. keine Übereinstimmung) zugeordnet. Man macht sich in dieser Ausführungsform der Erfindung die Erkenntnis zunutze, dass für eine Gruppe von Beobachtungen, die mit dem ursprünglichen Muster nicht übereinstimmt, über die Abstraktion der entsprechenden semantischen Aussagen im Muster ein Maß geschaffen werden kann, mit dem die Ähnlichkeit des Musters mit der Gruppe von Beobachtungen beschrieben wird. Die Abstraktion von semantischen Aussagen in allgemeinere Aussagen ist im Rahmen einer entsprechenden Beschreibung der Ontologie basierend auf einer Beschreibungssprache problemlos realisierbar.

[0013] In einer besonders bevorzugten Ausführungsform der Erfindung werden die semantischen Aussagen der Muster in der ontologischen Wissensbasis durch eine Beschreibungslogik repräsentiert, welche in an sich bekannter Weise als Entitäten Klassen, Relationen, Individuen in der Form von instantiierten Klassen und Variablen in der Form von Stellvertretern für mehrere Individuen umfasst. Dabei beschreibt eine semantische Aussage eine Relation zwischen zwei Entitäten, von denen eine Entität ein Individuum oder eine Variable ist, wobei die semantische Aussage oder die semantischen Aussagen in einem Muster mit einem oder mehreren Operatoren kombiniert sein können, wobei die zur Kombination verwendbaren Operatoren logische Operatoren und insbesondere die Operatoren UND, ODER und Negation umfassen. Um ggf. auch zeitliche Zusammenhänge beim Vergleich der Muster mit der Gruppe von Beobachtungen zu modellieren, umfassen die zur Kombination verwendbaren Operatoren vorzugsweise auch zeitliche Operatoren, mit denen eine semantische Aussage für eine oder mehrere Beobachtungen auf eine oder mehrere zeitlich zurückliegende Beobachtungen abgebildet wird.

[0014] Um Muster, in denen logische Operatoren verwendet werden, in geeigneter Weise mit Gruppen von Beobachtungen zu vergleichen, werden in einer besonders bevorzugten Ausführungsform die an sich bekannten De Morgan'schen Regeln verwendet. Dabei wird ein Muster, welches eine oder mehrere Negationen für eine oder mehrere Kombinationen von semantischen Aussagen umfasst, vor der Durchführung der schrittweisen Abstraktion basierend auf den De Morgan'schen Regeln derart gewandelt, dass das gewandelte Muster nur noch eine oder mehrere Negationen für eine oder mehrere einzelne semantische Aussagen enthält, wobei die Abstraktion einer Negation einer semantischen Aussage durch eine Tautologie (d. h. die semantische Aussage ist immer gültig) repräsentiert wird.

[0015] In einer weiteren Ausgestaltung des erfindungsgemäßen Verfahrens werden die Angriffe in bösartige, von unbefugten Dritten durchgeführten Angriffe und in gutartige, den Normalbetrieb des Computernetzes widerspiegelnde Angriffe kategorisiert. Gegebenenfalls können die bösartigen Angriffe auch nochmals in weitere Kategorien, wie z. B. verdächtige Angriffe und gefährliche Angriffe, untereilt werden. Dabei wird einer jeweiligen Sequenz von Beobachtungen ein Prioritätswert zugewiesen, wobei der Prioritätswert um so größer ist, je höher die Wahrscheinlichkeit ist, dass alle Gruppen von Beobachtungen der jeweiligen Sequenz zu einem bösartigen Angriff gehören, und wobei der Prioritätswert um so niedriger ist, je größer die Wahrscheinlichkeit ist, dass alle Gruppen von Beobachtungen der jeweiligen Sequenz zu einem gutartigen Angriff gehören. Diese Wahrscheinlichkeiten werden aus den in Schritt c) ermittelten Wahrscheinlichkeiten bestimmt. Über den Prioritätswert wird somit festgelegt, wie verdächtig eine entsprechende Sequenz an Beobachtungen in Bezug auf das Vorliegen eines bösartigen Angriffs ist.

[0016] In einer besonders bevorzugten Ausführungsform werden die jeweiligen Sequenzen mit deren Prioritätswerten als Hypothesen in einem Hypothesenpool über eine Benutzerschnittstelle ausgegeben, wobei ein jeweiliger Hypothesenpool eine vorbestimmte Anzahl an Hypothesen mit den größten Prioritätswerten enthält. D. h., in einem Hypothesenpool sind nur eine bestimmte Anzahl der am gefährlichsten eingestuften Hypothesen enthalten. Über den Prioritätswert wird einem Benutzer dabei vermittelt, wie wahrscheinlich ein bösartiger Angriff auf das Computernetz ist. Bei dem Hinzukommen von neuen Beobachtungen kann der Hypothesenpool in geeigneter Weise aktualisiert werden, indem durch die Hinzunahme der Beobachtungen neue Sequenzen an Beobachtungen gebildet werden. In einer bevorzugten Variante enthält ein Hypothesenpool mehrere Listen von Hypothesen, die sich darin unterscheiden, dass jede Liste Hypothesen mit einer unterschiedlichen Anzahl an Beobachtungen enthält.

[0017] Um die Qualität der Erkennung von Angriffen im erfindungsgemäßen Verfahren zu verbessern, wird in einer bevorzugten Ausführungsform das probabilistische Modell in zeitlichen Abständen mit neuen Trainingsdaten trainiert, wobei die neuen Trainingsdaten aus neu detektierten Sequenzen von Beobachtungen abgeleitet werden.

[0018] In einer bevorzugten Variante werden die neuen Trainingsdaten derart ermittelt, dass für eine neu detektierte Sequenz über eine Benutzerschnittstelle eine Eingabe eines Benutzers abgefragt wird, über welche der Benutzer die neu detektierte Sequenz als übereinstimmend mit einem oder mehreren vorgegebenen Angriffen spezifiziert, wobei basierend auf dem Vergleich gemäß Schritt b) übereinstimmende Muster mit dem maximalen Ähnlichkeitsmaß mit der neu detektierten Sequenz (d. h. mit allen Beobachtungen der Sequenz) ermittelt werden und diese übereinstimmenden Muster in den neuen Trainingsdaten als übereinstimmend mit dem oder den über den Benutzer als übereinstimmend spezifizierten Angriffen eingestuft werden. Vorzugsweise werden ferner die Muster mit einem anderen als dem maximalen Ähnlichkeitsmaß in den neuen Trainingsdaten als nicht übereinstimmend mit dem oder den über den Benutzer als übereinstimmend spezifizierten Angriffen eingestuft.

[0019] In einer weiteren Ausführungsform des oben beschriebenen Trainings kann der Benutzer bei seiner Eingabe in die Benutzerschnittstelle automatisiert unterstützt werden. In dieser Variante wird basierend auf den für die neu detektierte Sequenz in Schritt c) ermittelten Wahrscheinlichkeitsverteilungen dem Benutzer über die Benutzerschnittstelle ein Vorschlag für die Spezifikation der neu detektierten Sequenz als übereinstimmend mit einem oder mehreren vorgegebenen Angriffen ausgegeben. In der detaillierten Beschreibung ist ein Beispiel einer Funktion angegeben, mit der ein solcher Vorschlag realisiert werden kann.

[0020] Das Training des probabilistischen Modells kann im Rahmen des erfindungsgemäßen Verfahrens mit an sich bekannten Algorithmen durchgeführt werden. In einer besonders bevorzugten Ausführungsform wird das probabilistische Modell basierend auf dem an sich bekannten Improved-Iterative-Scaling-Algorithmus trainiert.

[0021] Neben dem oben beschriebenen Verfahren betrifft die Erfindung ferner ein System zum rechnergestützten Erkennen von Angriffen auf ein Computernetz. Das System ist dabei derart ausgestaltet, dass in dem System das erfindungsgemäße Verfahren bzw. eine oder mehrere Varianten des erfindungsgemäßen Verfahrens durchführbar sind. Im Besonderen umfasst das System ein Detektionsmittel, um Schritt a) des erfindungsgemäßen Verfahrens durchzuführen, ein Vergleichsmittel, um Schritt b) des erfindungsgemäßen Verfahrens durchzuführen, und ein Berechnungsmittel, um Schritt c) des erfindungsgemäßen Verfahrens durchzuführen.

[0022] Die Erfindung betrifft darüber hinaus ein Computerprogrammprodukt mit einem auf einem maschinenlesbaren Träger gespeicherten Programmcode zur Durchführung des erfindungsgemäßen Verfahrens bzw. einer oder mehrerer bevorzugter Varianten des erfindungsgemäßen Verfahrens, wenn der Programmcode auf einem Computer ausgeführt wird.

[0023] Darüber hinaus betrifft die Erfindung ein Computerprogramm mit einem Programmcode zur Durchführung des erfindungsgemäßen Verfahrens bzw. einer oder mehrerer Varianten des erfindungsgemäßen Verfahrens, wenn der Programmcode auf einem Computer ausgeführt wird.

[0024] Ausführungsbeispiele der Erfindung werden nachfolgend anhand der beigefügten Figuren detailliert beschrieben.

[0025] Es zeigen:

[0026] Fig. 1 eine schematische Darstellung der Komponenten einer Architektur zur Erkennung von Angriffen auf ein Computernetz basierend auf einer Ausführungsform der Erfindung; und

[0027] Fig. 2 eine Darstellung, welche beispielhaft die Abstraktion eines Musters verdeutlicht, wobei diese Abstraktion im Rahmen eines Mustervergleichs im erfindungsgemäßen Verfahren genutzt wird.

[0028] Die in Fig. 1 gezeigte Architektur stellt ein sog. SIEM-System (SIEM = Security Intrusion and Event Management) dar, welches dazu dient, die Geräte in einem Computernetzwerk zu überwachen und kritische Angriffe von Dritten in geeigneter Weise zu erkennen. In der Architektur der Fig. 1 wechselwirkt eine Vielzahl von Komponenten miteinander, was durch entsprechende Pfeile bzw. Doppelpfeile angedeutet ist. Sofern im Rahmen dieser Wechselwirkung erfindungswesentliche Informationen übertragen werden, befinden sich an den Pfeilen bzw. Doppelpfeilen entsprechende Bezugszeichen, welche diese Informationen spezifizieren.

[0029] Gemäß Fig. 1 wird mit an sich bekannten IDS-Systemen (IDS = Intrusion Detection System) das Computernetz überwacht, was in der Box AC (AC = Alert Collection) angedeutet ist. Dabei sind beispielhaft drei IDS-Systeme ID1, ID2 und ID3 gezeigt, wobei hierfür Systeme wie Snort, LML, IAS und dergleichen eingesetzt werden. Diese Systeme sind in dem Computernetz verteilt und ermitteln bzw. sammeln Sequenzen von Ereignissen in der Form von sog. Alerts. Diese Alerts werden in dem bekannten IDMEF-Format bereitgestellt und anschließend von der an sich bekannten Software Prelude (mit PL bezeichnet) weiterverarbeitet. Im Rahmen der Verarbeitung der IDMEF-Nachrichten erfolgt eine Alert-Aggregation sowie eine syntaktische und semantische Alert-Normalisierung, was in Fig. 1 durch die Box AN angedeutet ist. Die syntaktische und semantische Normalisierung von Alerts ist dem Fachmann geläufig und wird deshalb nicht weiter im Detail beschrieben. Zur semantischen Normalisierung enthält die in Fig. 1 mit KB bezeichnete ontologische Wissensbasis, die durch die Beschreibungssprache OWL-DL repräsentiert wird, eine Relation für jede mögliche Ausgabe der IDS-Systeme, um hierdurch festzulegen, wie normalisiert und kategorisiert werden soll. Im Rahmen der Aggregation der Alerts wird ferner ein an sich bekannter Burst-Detektor verwendet, der eine Vielzahl von zeitlich kurz aufeinanderfolgenden Ereignissen bzw. Beobachtungen mit der gleichen IP-Adresse, dem gleichen Port

und den gleichen Klassifikationswerten aggregiert. Die aggregierten und normalisierten Alerts sind in Fig. 1 mit x bezeichnet und werden im Rahmen eines Alert-Verfeinerungs-Prozesses AR (AR = Alert Refinement) einem sog. toleranten Mustervergleich unterzogen. Dabei werden Kombinationen von zurückliegende Beobachtungen (bis einschließlich der aktuellen Beobachtung), die weiter unten auch als Hypothesen bezeichnet werden, verarbeitet. Die Kombinationen von zurückliegenden Beobachtungen sind Sequenzen von Beobachtungen im Sinne von Anspruch 1.

[0030] Im Rahmen des Mustervergleichs wird auf die ontologische Wissensbasis KB zurückgegriffen. Dabei werden modellierte Muster von miteinander verknüpften semantischen Aussagen, die in der Wissensbasis basierend auf der Beschreibungssprache OWL-DL hinterlegt sind, auf Übereinstimmung mit den entsprechenden Sequenzen von Beobachtungen geprüft. D. h., es wird ermittelt, ob die verknüpften Aussagen der einzelnen Muster auf eine erfasste Sequenz von Beobachtungen zutreffen. Hierfür wird ein an sich bekannter Reasoner (z. B. Pellet) zur Ableitung von Schlussfolgerungen verwendet. Eine Sequenz stellt im Kontext des Mustervergleichs auch Gruppen von Beobachtungen aus einer Sequenz dar. In Fig. 1 ist der Mustervergleich allgemein mit Bezugszeichen PM bezeichnet und beispielhaft sind drei Muster p1, p2 und p3 angegeben. Der tolerante Mustervergleich PM zeichnet sich dabei dadurch aus, dass im Falle, dass durch den Reasoner keine Übereinstimmung einer Sequenz von Beobachtungen mit einem entsprechenden Muster gefunden wird, ein Ähnlichkeitswert des Musters mit der entsprechenden Sequenz von Beobachtungen ermittelt wird, wobei hierfür die einzelnen Muster abstrahiert werden, wie weiter unten noch näher beschrieben wird.

[0031] Die Ähnlichkeitswerte, welche in Fig. 1 mit F' bezeichnet sind, werden anschließend einer Angriffserkennung ARC (ARC = Attacksequence Recognition) zugeführt, welche einen Hypothesenpool HP sowie ein Interpretations- und Klassifikations-Modul IM umfasst. Im Rahmen der Angriffserkennung ARC werden für die Sequenzen von Beobachtungen Hypothesen h in Bezug auf das Vorliegen vorgegebener Angriffe bestimmt, indem entsprechende Wahrscheinlichkeitsverteilungen Pr der Angriffe für die jeweiligen Sequenzen über das Modul IM ermittelt werden. Hierüber wird für die einzelnen Hypothesen ein Prioritätswert bestimmt, welcher widerspiegelt, wie wahrscheinlich ein sicherheitsbedrohlicher Angriff vorliegt. Die Hypothesen werden in dem Pool gemäß diesen Prioritätswerten angeordnet. Aus dem Hypothesenpools HP fallen die Hypothesen mit den geringsten Prioritätswerten heraus, wodurch Platz für neue Hypothesen mit höheren Prioritätswerten generiert wird. In Rahmen der Erfassung von neuen Beobachtungen im Betrieb des Systems werden die Hypothesen kontinuierlich überprüft und durch die neuen Beobachtungen erweitert, um neue, wahrscheinlichere Angriffshypothesen zu generieren.

[0032] Die entsprechende Wahrscheinlichkeitsverteilungen für eine Hypothese h werden in der hier beschriebenen Ausführungsform über das an sich bekannte probabilistische Modell der Conditional Random Fields ermittelt, wobei dieses Modell im Folgenden auch mit CRF abgekürzt wird. Dieses Modell bestimmt die Wahrscheinlichkeiten Pr, dass entsprechende Gruppen von Beobachtungen aus der jeweiligen Hypothese einem vorgegebenen Angriff entsprechen. Das CRF-Modell wird auf entsprechenden Trainingsdaten TD gelernt, gemäß denen Zuordnungen bzw. Nichtzuordnungen von mit y bezeichneten Angriffen zu entsprechenden, in der ontologischen Wissensbasis hinterlegten Mustern festgelegt sind. In der hier beschriebenen Ausführungsform wird das CRF-Modell während der Laufzeit kontinuierlich weitertrainiert, wobei hierfür neue Trainingsdaten basierend auf neu hinzukommenden Sequenzen von Beobachtungen herangezogen werden. Dieses Trainieren wird weiter unten noch im Detail beschrieben.

[0033] Um einen Benutzer über vermutete Angriffe zu informieren, werden diesem die entsprechenden Hypothesen h aus dem Hypothesenpool und insbesondere die Hypothese mit dem größten Prioritätswert über eine graphische Benutzerschnittstelle UI präsentiert. Hierfür wird eine Web-Schnittstelle genutzt. Der Nutzer wird somit darüber informiert, ob von dem System vermutet wird, dass für bestimmte Sequenzen an Beobachtungen gefährliche Angriffe vorliegen. Ferner kann der Benutzer ggf. auch entscheiden, ob er im Rahmen eines Trainings die generierte Hypothese korrigiert, wobei in diesem Fall ein neues Trainingsdatum für das Interpretations- und Klassifikations-Modul IM generiert wird. Ebenso kann der Benutzer ggf. die für den toleranten Mustervergleich verwendeten Muster in Abhängigkeit seines Wissens über den Angriff remodellieren. In Fig. 1 ist ferner eine Komponente BR wiedergegeben, welche einen sog. AMQP-Broker darstellt, der basierend auf dem an sich bekanntem AMQP-Protokoll (AMQP = Advanced Message Queuing Protocol) die Kommunikation zwischen den unterschiedlichen Modulen durchführt und verwaltet.

[0034] Die hier beschriebene Ausführungsform der Erfindung enthält als zwei wesentliche Komponenten den toleranten Mustervergleich und die Hypothesengenerierung über das CRF-Modell. Ferner kann in einer speziellen Variante ein geeignetes Training des CRF-Modells während der Laufzeit des Verfahrens durchgeführt

werden. Dabei kann der Benutzer ggf. beim Annotieren von Trainingsdaten unterstützt werden. Im Folgenden werden die soeben genannten Komponenten im Detail erläutert.

[0035] Im Rahmen des toleranten Mustervergleichs wird eine ontologische Wissensrepräsentation basierend auf einer Beschreibungslogik verwendet, welche in der Wissensbasis KB abgelegt ist. Die Repräsentation von Wissen in der Form einer Ontologie ist dabei an sich bekannt. In der hier beschriebenen Ausführungsform wird die Ontologie verwendet, welche auf der Webseite http://www.fides-security.org/ hinterlegt ist. Diese Ontologie stellt eine geeignete logische Beschreibung von entsprechenden, in Computernetzen über IDS-Systeme erfassten Ereignissen bzw. Beobachtungen dar. Gegebenenfalls können auch andere Ontologien zum Einsatz kommen. Wie jede Ontologie umfasst die verwendete Ontologie Konzepte, semantische Relationen bzw. Rollen und Individuen. Konzepte stellen dabei Klassen von Objekten dar, welche hierarchisch strukturiert sein können (d. h. zu einer Klasse existieren ggf. eine oder mehrere untergeordnete (speziellere) bzw. übergeordnete (allgemeinere) Klassen). Ein Individuum ist dabei die Instantiierung einer entsprechenden Klasse. Eine semantische Relation bzw. Rolle beschreibt eine binäre Relation zwischen Konzepten/Individuen. Die Semantik wird im Rahmen der Ontologie durch die Interpretation (eine Interpretation kann als die entsprechende Domäne betrachtet werden) der Menge von Konzeptnamen, der Menge von Relationsnamen und der Menge aller Namen von Individuen definiert. Ein Konzept C wird dabei durch ein Konzept D subsumiert, falls für alle Interpretationen I $C^I \subseteq D^I$ gilt.

[0036] Der tolerante Mustervergleich beruht darauf, dass sukzessive die einzelnen Muster aus der Wissensbasis generalisiert werden, sofern eine betrachtete Sequenz bzw. Gruppe von Beobachtungen nicht mit dem entsprechenden Muster übereinstimmt. Für jedes generalisierte Muster wird immer wieder überprüft, ob dieses Muster nunmehr mit der Sequenz an Beobachtungen übereinstimmt. Sobald ein übereinstimmendes Muster gefunden wird, wird ein Ähnlichkeitsmaß ermittelt, das die Ähnlichkeit des ursprünglich nicht generalisierten Musters mit der entsprechenden Sequenz an Beobachtungen beschreibt. Ein Muster stellt dabei eine logische Verknüpfung von mehreren semantischen Aussagen (auch als Constraint bezeichnet) dar und kann ggf. auch aus einer einzelnen semantischen Aussage bestehen. In der hier beschriebenen Ausführungsform enthält die Wissensbasis semantische Aussagen, welche gemäß der nachfolgenden Definition 1 festgelegt sind.

Definition 1:

[0037] Eine Entität E ist entweder (a) ein Individuum, (b) eine Klasse oder (c) eine Variable, welche der Stellvertreter für eine Vielzahl von Individuen ist. Eine semantische Aussage $\gamma(x)$ ist in Bezug auf eine Sequenz von Beobachtungen x definiert als $\gamma = eRe'$, wobei e eine Entität, e' eine weitere Entität und R eine semantische Relation zwischen diesen Entitäten darstellt. Dabei ist entweder e oder e' ein Individuum oder eine Variable. Nachfolgend wird eine semantische Aussage durch γ bezeichnet.

[0038] Die Gültigkeit der semantischen Aussage wird dadurch überprüft, dass die relevanten Beobachtungen als Individuen bzw. Variablen in der semantischen Aussage eingesetzt werden und dann die Gültigkeit dieser Aussage überprüft wird.

[0039] Im erfindungsgemäßen Verfahren wird das Konzept der teilweise übereinstimmenden Muster eingeführt. Dieses Konzept wird in der hier beschriebenen Ausführungsform über die nachfolgende Definition 2 beschrieben.

Definition 2:

[0040] Ein Muster p besteht aus einem Satz von semantischen Aussagen und logischen Verknüpfungen (sofern vorhanden) unter diesen. Ein teilweise übereinstimmendes Muster für eine Sequenz an Beobachtungen x wird durch eine reellwertige Funktion im Wertebereich [0,1] beschrieben. Der Wert einer solchen Funktion wird als Grad der Übereinstimmung bzw. Übereinstimmungsgrad bezeichnet und wird im folgenden als F' definiert.

[0041] Jede semantische Aussage in einem Muster kann durch ein entsprechendes Anfrage-Tripel in Beschreibungslogik ausgedrückt werden. Dies ermöglicht eine einfache Transformation der Muster in eine Anfrage-Sprache, wie z. B. die aus dem Stand der Technik bekannte Anfrage-Sprache SPARQL. SPARQL kann von einer Vielzahl von sog. Reasoner interpretiert werden, welche die Anfrage auswerten können und hierüber bestimmen können, ob ein Muster mit einer entsprechenden Sequenz an Beobachtungen übereinstimmt. In der hier beschriebenen Ausführungsform wird Pellet als Reasoner verwendet, um eine Übereinstimmung zwischen einem Muster und einer Sequenz von Beobachtungen zu bestimmen. Sollte keine Übereinstimmung

festgestellt werden, wird das entsprechende Muster schrittweise abstrahiert, bis ein abstrahiertes Muster gefunden wird, das übereinstimmt. Diese Abstraktion wird weiter unten näher beschrieben.

[0042] Zur Veranschaulichung wird ein Beispiel beschrieben, wie eine semantische Aussage über einen Reasoner geeignet verarbeitet werden kann. Es wird davon ausgegangen, dass die semantische Aussage γ_1 die Beschränkung ist, dass eine Beobachtung (z. B. eine IDMEF-Nachricht) mit dem Attributwert von classification das gleiche wie das Individuum portsweep ist. Ferner wird davon ausgegangen, dass die semantische Aussage γ_2 die Beschränkung darstellt, dass der Attributwert von classification das gleiche wie das Individuum ping sein muss. Ferner liegt ein Muster p vor, welches durch die ODER-Verknüpfung $\gamma_1 \vee \gamma_2$ gegeben ist. Nunmehr kann die Anfrage „Ist das Muster p für classification = portscan erfüllt?" in die folgende SPARQL-Anfrage umgewandelt werden:

{ns:portscan	owl:sameAs	ns:portsweep}	UNION
{ns:portscan	owl:sameAs	ns:ping}	

[0043] Diese Anfrage ist offensichtlich nicht erfüllt, da ein Portscan nicht das Gleiche wie ein Portsweep oder ein Ping ist.

[0044] Die Aussagen aus dem obigen Beispiel beschreiben jedoch sog. Reconnaissance-Angriffe. Sofern kein anderes Muster erfüllt ist, kann dies einen guten Hinweis auf die Art der Beobachtung liefern. Dies kann auf einfache Weise durch eine geeignete Subsumption erreicht werden, d. h. jeder Portsweep, Ping und Portscan kann durch eine Klasse mit der Bezeichnung „Scan" subsumiert werden. Beispielsweise kann die Aussage γ_1, die Aussage γ_2 oder beide Aussagen zu der Bedingung abstrahiert werden, dass der Portscan ein Scan-Ereignis sein muss. Jede dieser Abstraktionen des Musters ist dann erfüllt. Jedoch ist es wünschenswert, die kleinstmögliche Abstraktion zu finden, um den größtmöglichen semantischen Aussagegehalt des ursprünglichen Musters beizubehalten, d. h. es sollte entweder die Aussage γ_1 oder die Aussage γ_2 und nicht beide abstrahiert werden.

[0045] Um nunmehr für eine entsprechende Sequenz von Beobachtungen, welche mit dem ursprünglichen Muster nicht übereinstimmt, geeignete abstrahierte Muster zu finden, wird die Relation $\geq_g$ verwendet, welche beschreibt, dass eine Aussage die Abstraktion einer anderen Aussage ist. Das heißt, eine Aussage γ_1 ist generischer oder abstrakter oder genauso generisch oder abstrakt wie die Aussage γ_2 (bezeichnet durch $\gamma_1 \geq_g \gamma_2$) falls für alle Interpretationen γ_2^I von γ_2 eine Interpretation γ_1^I von γ_1 existiert, so dass $\gamma_2^I \subseteq \gamma_1^I$.

[0046] Im Folgenden werden hochgestellte Indizes zur Bezeichnung von verschiedenen Stufen der Spezialisierung verwendet. Eine 0 bezeichnet dabei den abstraktesten Fall, wohingegen größere Zahlen eine zunehmende Spezialisierung und damit eine geringere Abstraktion darstellen. Je größer die Zahl ist, desto geringer ist somit die Abstraktion. Beispielsweise ist p^0 die direkte Abstraktion von p^1. Fig. 2 zeigt ein Beispiel, wie im Rahmen des erfindungsgemäßen Verfahrens ein Muster p^1 auf dem Spezialisierungsniveau L1 in das abstraktere Muster p^0 auf dem Spezialisierungsniveau L0 gewandelt werden kann. Das Muster p^1 beschreibt dabei die Verknüpfung zu mehreren semantischen Aussagen. Dabei wird die semantische Aussage γ_1^1 UND-verknüpft mit einer Negation. Die Negation bezieht sich auf eine ODER-Verknüpfung. Gemäß dieser ODER-Verknüpfung wird die semantische Aussage γ_3^1 mit der negierten semantischen Aussage γ_2^1 verknüpft. Aus diesem Muster p^1 kann das Muster p^0 abgeleitet werden. In diesem Muster p^0 sind die nächst-abstrakteren Aussagen γ_1^0, γ_2^0 und γ_3^0 der ursprünglichen Aussagen γ_1^1, γ_2^1 und γ_3^1 enthalten.

[0047] Bei der Abstraktion des Musters p^1 ist zu beachten, dass zunächst alle Negationen zu den Blättern (d. h. zu den Constraints) durch das Anwenden der Regeln von De Morgan (d. h. $\neg(a \wedge b) = \neg a \vee \neg b$ und $\neg(a \vee b) = \neg a \wedge \neg b$) propagiert werden müssen. Dies ist erforderlich, um zu detektieren, welches Constraint negiert ist. Die Konstruktion von dieser negationalen Normalform kann vor der Abstraktion der Muster durchgeführt werden oder bereits vorab auf alle Muster angewendet werden. Anschließend werden schrittweise zunächst einzelne Aussagen abstrahiert, bis ein abstrahiertes Muster gefunden wird, das mit der Sequenz an Beobachtungen übereinstimmt. Wird ein solches abstrahiertes Muster nicht gefunden, werden schrittweise immer mehrere semantische Aussagen abstrahiert, bis schließlich ein abstrahiertes Muster gefunden wird. Das Suchen nach einem übereinstimmenden Muster kann z. B. mit dem an sich bekannten Divide-and-Conquer-Algorithmus durchgeführt werden. Wird kein übereinstimmendes Muster gefunden, ist das ursprüngliche Muster nicht übereinstimmend mit der Sequenz x an Beobachtungen. Bei der Abstraktion der semantischen Aussagen werden folgende Regeln angewendet:

– eine negierte semantische Aussage wird zu einer Tautologie abstrahiert, da das vorliegende Modell einer solchen semantischen Aussage alle Individuen bis auf das negierte umfasst. Beispielsweise hat sich die semantische Aussage γ_3^0 in dem Beispiel der Fig. 2 in die Tautologie t gewandelt. Mit anderen Worten sind alle Individuen gültige Ergebnisse bis auf das negierte. Eine Abstraktion dieser Aussage, gemäß der die gültigen Ergebnisse erweitert werden, muss das negierte Individuum beinhalten und ist somit eine Tautologie.
– Falls die Entität der abstrahierten semantischen Aussage eine Klasse oder ein Individuum ist, wird dieses durch eine abstraktere Klasse basierend auf der Definition $\geq_g$ ersetzt. Die eingesetzten Werte aus den Beobachtungen sind fest und nicht abstrahiert, nur die Bedingungen werden abstrahiert, in denen diese auftreten können.
– Die Relation einer abstrahierten semantischen Aussage muss ggf. ersetzt werden, um sicherzustellen, dass der Satz von Interpretationen der Aussage zunimmt. Beispielsweise muss eine Identitäts-Relation durch eine geeignete (transitive) Subklassen-Relation ersetzt werden (z. B. wird owl:sameAs ersetzt durch rdf:subClassOf).

[0048] Im folgenden wird dargelegt, wie ein Maß $\theta(\gamma^j, \gamma^k)$ für die semantischen Aussagen γ^j und γ_k festgelegt wird, um die Ähnlichkeit einer abstrahierten semantischen Aussage γ von dem ursprünglichen Spezialisierungsniveau j zu dem abstrahierten Spezialisierungsniveau k zu quantifizieren. Dabei bezeichnet $\gamma^\perp$ die ursprüngliche semantische Aussage auf dem speziellsten Niveau $\perp$ und $\theta(\gamma^j)$ steht für $\theta(\gamma^j, \gamma^j)$ Dieses Maß wird auf den Wert 1 gesetzt, falls die semantische Aussage nicht abstrahiert wird, und nimmt ab, falls die semantische Aussage immer abstrakter wird. Das Maß bleibt dabei immer > 0.

[0049] Das obige Maß kann auch als eine Ähnlichkeitsbeziehung betrachtet werden, welche aussagt, wie gut die Aussage γ^j die Aussage γ^k beschreibt bzw. wie ähnlich diese Aussagen sind. Die Eigenschaften einer entsprechenden Ähnlichkeitsfunktion sind basierend auf der nachfolgenden Definition 3 festgelegt.

Definition 3:

[0050] Ein Ähnlichkeitsmaß θ ist eine reellwertige Funktion zwischen [0,1], die durch folgende Eigenschaften definiert ist:

$\forall \gamma^j, \gamma^k: \theta(\gamma^j, \gamma^k) \geq 0$ (positive Definitheit)

$\forall \gamma^j, \gamma^k: \theta(\gamma^j, \gamma^k) = \theta(\gamma^k, \gamma^j)$ (Symmetrie)

$\forall \gamma^j, \gamma^k: \theta(\gamma^j, \gamma^k) \leq \theta(\gamma^j, \gamma^j)$ (Identität)

$\forall \gamma^j < k: \theta(\gamma^j, \gamma^{k+1}) < \theta(\gamma^j, \gamma^k)$ (Monotonität)

[0051] Die Werte einer solchen Ähnlichkeitsfunktion für entsprechende semantische Aussagen werden zu einem Übereinstimmungsgrad des Musters kombiniert, indem der Fusions-Operator $F'(\theta_1, ..., \theta_n)$ angewendet wird. Dies ist erforderlich, um die Semantik der logischen Operatoren beim Abstrahieren der Muster zu berücksichtigen. In der hier beschriebenen Ausführungsform wird eine probabilistische Fusion verwendet, welche auf einer Bayes'schen Ähnlichkeitsinterpretation des Baums von logischen Operatoren in jedem Muster basiert. Diese Fusion ist basierend auf der nachfolgenden Definition 4 festgelegt:

Definition 4:

[0052] Die Fusionsfunktion F' eines Musters p ist rekursiv in Bezug auf eine Ähnlichkeitsfunktion θ von Constraints γ definiert, die sich aus logischen Operatoren zusammensetzen.

$$F'\left(\gamma_1^i \wedge \gamma_2^j\right) = F'\left(\gamma_1^i\right) \cdot F'\left(\gamma_2^j\right)$$

$$F'\left(\gamma_1^i \vee \gamma_2^j\right) = \max\left(F'\left(\gamma_1^i\right), F'\left(\gamma_2^j\right)\right)$$

$$F'\left(\neg\gamma^i\right) = \left\{\begin{array}{ll} 1 - F'\left(\gamma^i\right), & \text{für } i = \perp \\ \beta \cdot F'\left(\gamma^i\right), & \text{sonst} \end{array}\right\}$$

$$F'\left(\gamma^i\right) = \theta\left(\gamma^i\right).$$

[0053] Dabei bezeichnet $\beta \in [0,1]$ einen Bestrafungsfaktor, um zusätzlich die Abstraktion von Negationen zu bestrafen. Um eine Sequenz an Beobachtungen x hin zu Beschränkungen (welche unabhängig von der Sequenz an Beobachtungen sind) zu propagieren, kann die Fusionsfunktion auch als unabhängig von x betrachtet werden. Der obige Faktor β kann von der verwendeten Ähnlichkeitsfunktion und der Tiefe der Ontologie (d. h. der Anzahl an Spezialisierungsniveaus) abhängen.

[0054] Der Disjunktions-Teil einer Bayes'schen Fusion wurde durch den max-Operator ersetzt, um sicherzustellen, dass die Fusionsfunktion monoton fallend in Bezug auf ein beliebiges θ ist. Diese Eigenschaft ist in Kombination mit den weiter unten beschriebenen Conditional Random Fields von Vorteil. Die Konjunktion der Fusionsfunktion kann als ein deterministischer UND-Knoten in dem Sinne eines Bayes'schen Netzwerks aufgefasst werden. Durch die Annahme von unabhängigen semantischen Aussagen kann das Ergebnis effizient berechnet werden. Beispielsweise kann das Muster p^1 aus Fig. 1 repräsentiert werden durch:

$$F'(p^1) = F'(\gamma_1^1) \cdot (1 - \max(F'(\gamma_2^1), 1 - F'(\gamma_2^1)))$$

[0055] Die Negation ist anders zu interpretieren als in einer entsprechenden Bayes'schen bedingten Wahrscheinlichkeitstabelle, um sicherzustellen, dass eine zunehmende Abstraktion zu einer abnehmenden Fusionsfunktion führt. Diese unterschiedliche Interpretation resultiert daraus, dass die Negation einer abstrahierten semantischen Aussage eine Spezialisierung ist (d. h. die Menge der möglichen Lösungen nimmt bei der Abstraktion ab). Die hier verwendete Interpretation einer Abstraktion, wonach ein Satz an Lösungen (Modellen) einer Aussage immer zunimmt (was das Komplement einer Negation/Spezialisierung ist), führt dazu, dass auch das Komplement der Bayes'schen Interpretation für die Negation verwendet werden muss. Die Abstraktion einer Negation kann jedoch einen hohen Einfluss auf die Semantik haben. Deshalb wird ein Straffaktor β eingeführt. Dies führt zu der Monotonität von F' in Bezug auf θ, was sehr nützlich ist, um die am wenigsten spezifische Abstraktion zu finden.

[0056] Mit den obigen Eigenschaften wird eine teilweise Ordnung der Muster in Bezug auf die Generalität der in diesen Muster enthaltenen Aussagen definiert. Basierend darauf muss das Muster mit dem größten Wert von F' gefunden werden, wobei der Wert von F' das Ähnlichkeitsmaß zum ursprünglichen, noch nicht abstrahierten Muster ist.

[0057] In dem Beispiel der Fig. 2 kann γ_1^1 beispielsweise die Aussage darstellen, dass ein Portscan-Ereignis durch ein IDS-System empfangen wurde. γ_2^1 kann die Aussage darstellen, dass die IP-Adresse der Quelle dieses Ereignisses aus einem internen Netzwerk extrahiert ist. Ferner kann γ_3^3 die Aussage darstellen, dass die IP-Adresse der Quelle des Ereignisses zu dem PC des Administrators gehört. In Bezug auf die logischen Zusammensetzungen in Fig. 2 passt dieses Muster auf einen Portscan von einem Host aus dem internen Netz, der nicht der PC des Administrators ist. D. h., dieses Muster sollte sicherstellen, dass kein PC des internen Netzwerks einen Portscan außer der PC des Administrators durchführt. Dieses Muster ist aufgrund der vielfachen Negationen nicht intuitiv verständlich, stellt jedoch ein gutes Beispiel dar, um den Einfluss der Fusionsfunktion wiederzugeben. Anstatt eines Portscan-Ereignisses wird nunmehr ein Ping-Ereignis betrachtet, welches sich auf eine sehr ähnliche Situation bezieht, da es auch ein sog. Reconnaissance-Ereignis von einem möglichen Angreifer darstellt und deshalb in der Ontologie strukturell sehr ähnlich ist. Beispielsweise kann ein Ping und ein Portscan zu dem Konzept bzw. der Klasse Reconnaissance-Ereignis oder Scan-Ereignis zusammengefasst werden. Das beschriebene Muster ist nicht erfüllt, da dieses Muster nur entworfen wurde, um auf Portscans zu passen. Jedoch kann mit dem oben beschriebenen toleranten Mustervergleich das Muster in Bezug auf abstrahiert werden, um das Ping-Ereignis mit zu umfassen. Beispielsweise kann $\theta(\gamma_1^1) = 0{,}5$ gelten (γ_1^1 ist abstrahiert zu γ_1^0). Es wird somit angenommen, dass das Muster dem Ähnlichkeitswert von 0,5 entspricht, da alle anderen semantischen Aussagen erfüllt sind. Betrachtet man das Ereignis, dass der pingende PC der PC

des Administrators ist, muss die Aussage γ_3^1 auch abstrahiert werden, d. h. es gilt $\theta(\gamma_3^1) = 0{,}1$ (dabei ist $\beta = 0{,}2$ gesetzt). Dies führt zu einem Gesamtähnlichkeitsmaß von nur 0,05, was bedeutet, dass das Muster durch das Ereignis eines pingenden PCs, welcher der Administrator-PC ist, bei weitem nicht erfüllt wird. Falls weitere der obigen semantischen Aussagen nochmals abstrahiert werden müssen, nimmt der Ähnlichkeitsgrad weiter ab.

[0058] Bei der Verwendung der obigen Fusionsfunktion F' gemäß Definition 4 ist noch eine geeignete Funktion $\theta(\gamma^k)$ festzulegen. Die Erfinder konnten ermitteln, dass eine geeignete Wahl für die Ähnlichkeitsfunktion θ in Abhängigkeit von einer semantischen Aussage γ^k in Bezug auf die Anzahl der Abstraktionen $\perp - k$ wie folgt lautet:

$$\theta(\gamma^k) = \left(\frac{1}{|\mathbf{p}|-1}\right)^{\perp-k}$$

[0059] Dabei bezeichnet |p| die Anzahl an Mustern in der ontologischen Wissensbasis. Man erkennt aus dieser Gleichung, dass im Falle von keiner Abstraktion (d. h. Exponent 0) der Ähnlichkeitswert bei 1 liegt und somit die Übereinstimmung zwischen dem Muster und der entsprechenden Sequenz an Beobachtung darstellt. Je größer die Anzahl von Abstraktionen ist, desto kleiner wird der Ähnlichkeitswert.

[0060] In der hier beschriebenen Ausführungsform erhält man für eine betrachtete Sequenz von Beobachtungen für jedes Muster aus der Wissensbasis ein Ähnlichkeitsmaß, das die Ähnlichkeit des jeweiligen Musters zu der Sequenz an Beobachtungen beschreibt. Das Ähnlichkeitsmaß ist dabei 1, wenn über einen Reasoner eine Übereinstimmung der Sequenz an Beobachtungen mit dem Muster festgestellt wird. Ist dies nicht der Fall, wird das entsprechende Muster schrittweise abstrahiert, bis eine Übereinstimmung durch den Reasoner festgestellt wird. Über den Grad der Abstraktion wird dann ein entsprechender Wert des Ähnlichkeitsmaßes festgelegt, der größer 0 und kleiner 1 ist. Ist es nicht möglich, ein abstrahiertes Muster zu finden, wird das Ähnlichkeitsmaß auf 0 gesetzt.

[0061] Die ermittelten Ähnlichkeitsmaße werden anschließend im Rahmen eines probabilistischen Modells dazu verwendet, um für eine Sequenz an Beobachtungen zu bestimmen, wie wahrscheinlich für diese Sequenz vorbestimmte Angriffe vorliegen. In der hier beschriebenen Ausführungsform werden Conditional Random Fields (auch mit CRF abgekürzt) als probabilistisches Modell verwendet. CRF sind an sich aus dem Stand der Technik bekannt und werden hauptsächlich im Bereich der Sprachverarbeitung eingesetzt. Die Verwendung von CRF ermöglicht es, die starken Unabhängigkeits-Annahmen, welche typischerweise mit bekannten Hidden-Markov-Modellen gemacht werden, abzuschwächen. Im Unterschied zu Hidden-Markov-Modellen ermöglichen CRF mehrfach überlappende und abhängige Merkmale, die geeigneter sind, um einen sequentiellen Kontext in der Form von zeitlich aufeinander folgenden Beobachtungen in einem Computernetz zu beschreiben.

[0062] In der nachfolgenden Definition 5 werden zunächst die CRF definiert.

Definition 5:

[0063] Conditional Random Fields sind für einen Label bzw. eine Klasse $y \in y$ definiert, und zwar unter der Bedingung eines Vektors an Beobachtungen x, wobei ein Satz von reellwertigen Gewichten λ und ein entsprechender Satz von reellwertigen Merkmalsfunktionen f mit $f_i \in f$ verwendet wird. Hieraus ergibt sich folgende bedingte Wahrscheinlichkeit:

$$Pr(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\left(\sum_{i=1}^{n}\lambda_i f_i(\mathbf{x}, y)\right)$$

[0064] Erfindungsgemäß stellen y entsprechend vorgegebene Angriffe dar und x ist wiederum die entsprechende Sequenz an Beobachtungen, die im Computernetz detektiert wurde. In der obigen Wahrscheinlichkeitsfunktion ist eine Normalisierungsfunktion Z enthalten, um sicherzustellen, dass das Ergebnis eine korrekte Wahrscheinlichkeitsmassenfunktion ist. Die Normalisierungsfunktion ist durch nachfolgende Definition 6 gegeben.

Definition 6:

[0065] Die Normalisierungsfunktion Z eines Conditional Random Fields ist für eine Beobachtungssequenz x über die Summe aller möglichen Labels y (d. h. im vorliegenden Fall über die Summe aller möglichen vorgegebenen Angriffe) wie folgt definiert:

$$Z(\mathbf{x}) = \sum_{y \in \mathbf{y}} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}, y) \right)$$

[0066] Die obige CRF-Funktion ist durch einen Satz von gewichteten Merkmalsfunktionen definiert, welche jeweils einen Aspekt der Eingabedaten darstellen. Im Rahmen der hier beschriebenen Ausführungsform werden die Gewichte $\lambda_i \in \lambda$ in einem geeigneten Training gelernt, welches weiter unten beschrieben wird und auf Trainingsdaten basiert, gemäß denen den Mustern entsprechende Angriffe zugeordnet bzw. nicht zugeordnet werden. Zunächst wird jedoch erläutert, wie mit den bereits trainierten Modellen (d. h. mit bereits festgelegten Gewichten) die obige Wahrscheinlichkeit bestimmt werden kann, wobei hierfür die entsprechenden Merkmalsfunktionen mit Werten der obigen Fusionsfunktion F' korreliert werden.

[0067] Wie bereits erwähnt, stellen die Label $y \in y$ Angriffe dar. Zum Beispiel könnte ein Angriff der Versuch einer SQL-Injektion sein. Betrachtet man diskrete Zeitschritte {1, ..., t} und einen diskreten Satz an Beobachtungen $x = \{x_1, ..., x_t\}$ ist es das Ziel, eine Teilmenge der Beobachtungen zu finden, welche am wahrscheinlichsten einen Angriff darstellt. Jede Teilmenge stellt dabei eine Sequenz von Beobachtungen im Sinne der Ansprüche dar und wird als eine Hypothese $h \in h$ betrachtet. Beispielsweise kann $\{x_1, x_2, x_4\}$ als eine Hypothese und $\{x_1, x_4\}$ als eine andere Hypothese betrachtet werden. Eine Hypothese $h \in h$ ist eine Untermenge der Beobachtungen $x_h \subseteq x$ des Satzes an zurückliegenden Beobachtungen.

[0068] Die Hypothesen können sich somit in ihren Beobachtungen und in der Anzahl ihrer Beobachtungen unterscheiden. In der hier beschriebenen Ausführungsform werden die wahrscheinlichsten Angriffe (y_j mit $j \in \{1, ..., t\}$), welche zu einem Satz von Beobachtungen einer Hypothese gehören, durch die Wahrscheinlichkeitsmassenfunktion der Angriffe über die Zeit gegeben die Beobachtungen ermittelt. Diese Wahrscheinlichkeitsmassenfunktion kann auf einfache Weise durch CRF berechnet werden.

[0069] Die Bestimmung der Wahrscheinlichkeitsmassenfunktion wird im Folgenden an einem Beispiel erläutert. Es wird ein erster Angriff y_1 betrachtet, der den Versuch einer SQL-Injektion darstellt. Ferner wird ein zweiter Angriff y_2 betrachtet, der das normale Verhalten des Computersystems widerspiegelt (normales Verhalten des Computernetzes wird als gutartiger Angriff eingestuft). Die Wahrscheinlichkeitsmassenfunktion in Abhängigkeit von der Zeit einer gegebenen Hypothese mit drei Beobachtungen ist dabei in Tabelle 1 wiedergegeben. Dabei ist zu beachten, dass der Index bei einem dick gedruckten y verschiedene Zeitpunkte repräsentiert, wohingegen der Index bei einem normal gedruckten y verschiedene Angriffe wiedergibt. In dem Beispiel der Tabelle 1 enthält die Hypothese eine Sequenz aus drei Beobachtungen, wobei die erste Beobachtung zu dem Versuch einer SQL-Injektion gehört, die zweite Beobachtung ein normales Verhalten des Computersystems widerspiegelt (y_2) und für die dritte Beobachtung beide Angriffe gleich wahrscheinlich sind. In der Tabelle 1 bezeichnet x_{h1} bzw. $x_{h1:2}$ bzw. $x_{h1:3}$ die Menge an Beobachtungen für unterschiedliche Zeitschritte. Somit enthält x_{h1} die Beobachtung zum Zeitschritt 1, $x_{h1:2}$ die Beobachtungen zu den Zeitschritten 1 und 2 sowie $x_{h1:3}$ die Beobachtungen bis zum Zeitschritt 3 (d. h. die Beobachtungen zum Zeitschritt 1, 2, und 3).

Tabelle 1:

y_1	$Pr(y_1\|x_{h1})$	y_2	$Pr(y_2\|x_{h1:2})$	y_3	$Pr(y_3\|x_{h1:3})$
y_1 y_2	0,9 0,1	y_1 y_2	0,1 0,8	y_1 y_2	0,5 0,5

[0070] Im Folgenden wird nunmehr erläutert, wie die Ähnlichkeitsmaße, die über den toleranten Mustervergleich ermittelt wurden, in die Merkmalsfunktionen der Wahrscheinlichkeitsmassenfunktion gemäß CRF einfließen. Dieser Zusammenhang wird durch die nachfolgende Definition 7 repräsentiert.

Definition 7:

[0071] Eine (teilweise übereinstimmende) Merkmalsfunktion $f_i(x, y) \in f$ ist eine reellwertige Funktion in dem Intervall [0,1] in Abhängigkeit von einer Sequenz an Beobachtungen x und einer Klasse (d. h. einem Angriff) $y \in y$ aus einer Menge an vorgegebenen Angriffen y. Der Wert einer solchen Funktion wird Übereinstimmungsgrad genannt und durch die oben beschriebene Fusionsfunktion des toleranten Mustervergleichs wie folgt bestimmt:

$$f_i(\mathbf{x}, y) = \begin{cases} F'(p(\mathbf{x})) & falls\ das\ Merkmal\ f_i\ dem\ Angriff\ zugeordnet\ ist \\ 0 & sonst \end{cases}$$

[0072] Die Abhängigkeit der Merkmalsfunktion f_i von y ergibt sich dabei durch die Zuordnung der entsprechenden Muster zu dem Angriff, wobei diese Zuordnung für jeden Angriff unterschiedlich ist, was weiter unten nochmals an einem Beispiel verdeutlicht wird. Gemäß der obigen Formel existiert für jede Kombination eines Musters mit einem Angriff eine Merkmalsfunktion $f_i \in f$. Der Wert dieser Funktion entspricht dem Ähnlichkeitsmaß F' des Musters in Bezug auf die Beobachtung x, falls das Muster zu dem entsprechenden Angriff gehört. Ist dies nicht der Fall, ist das entsprechende Merkmal 0. Jede Merkmalsfunktion ist dabei genau einem Angriff zugeordnet. Es gibt somit $|y| \times |p|$ mögliche Merkmalsfunktionen. Durch eine fortlaufende Indizierung der Merkmalsfunktionen werden schrittweise für jeden Angriff $y_1, y_2, ...$ die Merkmalsfunktionen der Reihe nach den einzelnen Mustern $p_1, p_2, ...$ zugeordnet, wodurch sich die entsprechende Zuordnung der Merkmalsfunktionen zu den Angriffen ergibt. Dies wird durch die nachfolgende Tabelle 2 verdeutlicht, welche zwei Angriffe y_1, y_2 und zwei Muster p_1 und p_2 enthält. Wie man erkennt, sind die Merkmalsfunktionen f_1 und f_2 dem Angriff y_1 zugeordnet und die Merkmalsfunktionen f_3 und f_4 dem Angriff y_2.

Tabelle 2:

passt auf	p_1	p_2
falls $y = y_1$ falls $y = y_2$	$f_1(x, y) = F'(p_1(x))$ $f_3(x, y) = F'(p_1(x))$	$f_2(x, y) = F'(p_2(x))$ $f_4(x, y) = F'(p_2(x))$

[0073] In der hier beschriebenen Ausführungsform wird basierend auf der Ermittlung der obigen Wahrscheinlichkeitsverteilungen für entsprechende Hypothesen ein sog. Hypothesenpool gebildet, in dem die Hypothesen für verschiedene Sequenzen an Beobachtungen basierend auf entsprechend ermittelten Prioritätswerten angeordnet sind. Je höher der Prioritätswert ist, desto wahrscheinlicher ist es, dass für die entsprechende Sequenz an Beobachtungen ein gefährlicher Angriff vorliegt. Die Ermittlung geeigneter Prioritätswerte wird weiter unten näher beschrieben. Im Hypothesenpool werden somit die Hypothesen mit den höchsten Prioritäten gehalten, wobei Hypothesen mit geringen Prioritäten im Pool nach unten wandern und schließlich verworfen werden. In jedem neuen Zeitschritt wird eine neue Hypothese (mit exakt einer Beobachtung) für die neue Beobachtung generiert. Ferner wird jede Hypothese in dem Hypothesenpool kopiert und durch die neue Beobachtung erweitert. Dies führt dazu, dass der Hypothesenpool die Menge der momentanen Hypothesen plus der zusätzlichen neuen Hypothesen enthält. Falls der Hypothesenpool eine vorgegebene Anzahl an Hypothesen überschreitet, werden die Hypothesen mit den geringsten Prioritäten aus dem Pool entfernt, um den Rechenaufwand zu begrenzen. In einer Realisierung der Erfindung werden fünf Listen mit maximal fünf Hypothesen in jeder Liste betrachtet, wobei jede Liste eine Hypothese mit einer vorbestimmten Länge betrifft. D. h., die erste Liste enthält alle Hypothesen mit nur einer Beobachtung, die zweite Liste alle Hypothesen mit zwei Beobachtungen usw. Wird eine Hypothese aus der ersten Liste um eine Beobachtung erweitert, wandert diese Hypothese in die zweite Liste aufgrund der größeren Länge. Nach jedem Zeitschritt werden die schlechtesten Hypothesen gemäß den geringsten Prioritätswerten aus jeder Liste herausgeworfen, so dass in der Liste immer nur fünf Hypothesen verbleiben. Dies ist nochmals beispielhaft an nachfolgender Tabelle 3 verdeutlicht.

Tabelle 3:

Position	Hypothese (Länge 1)	Priorität	Hypothese (Länge 2)	Priorität
1	$\{x_2\}$	0,5	$\{x_1, x_2\}$	1,0
2	$\{x_1\}$	0,3	$\{x_1, x_3\}$	0,9
3	$\{x_3\}$	0,2	$\{x_5, x_6\}$	0,8

4	$\{x_4\}$	0,2	$\{x_1, x_6\}$	0,5
5	$\{x_6\}$	0,2	$\{x_3, x_5\}$	0,3
verwerfen ...	$\{x_5\}$ ...	0,1 ...	$\{x_1, x_4\}$ ...	0,2 ...

[0074] Optional kann ein Faktor verwendet werden, gemäß dem die Priorität von älteren Hypothesen herabgesetzt wird. Ferner kann ggf. die Größe des Hypothesenpools dynamisch vergrößert bzw. verkleinert werden, um hierdurch den Rechenaufwand an die verfügbaren Ressourcen anzupassen.

[0075] Die Einführung der obigen Hypothesen ermöglicht es, ggf. zeitliche Verknüpfungen bzw. Operatoren im Rahmen der Ontologie zu verwenden. Auf diese Weise können Abhängigkeiten über die Zeit berücksichtigt werden. Zum Beispiel kann ein fehlgeschlagener Versuch eines Logins kritischer eingestuft werden, wenn zuvor ein Portscan erfolgt ist. Es können dabei drei zeitliche Operatoren berücksichtigt werden:

– Der Operator „momentan": Dieser Operator bildet eine semantische Aussage auf eine momentane Gruppe von Beobachtungen ab.

– Der Operator „vorher" (bezeichnet mit prev): Gemäß diesem Operator wird eine semantische Aussage auf eine Gruppe von Beobachtungen zum vorhergehenden Zeitpunkt abgebildet. Falls $\gamma(x_t)$ die semantische Aussage für die Beobachtungsgruppe x_t zum Zeitpunkt t ist, so gilt $prev(\gamma(x_t)) = \gamma(x_{t-1})$.

– Der Operator „vorvorher" (bezeichnet mit pprev): Gemäß diesem Operator wird eine semantische Aussage auf Gruppen von Beobachtungen zu mehreren vorhergehenden Zeitpunkten im Sinne einer Disjunktion abgebildet. Falls $\gamma(x_t)$ die Aussage für die Beobachtungsgruppe x_t zum Zeitpunkt t ist, so gilt $pprev(\gamma(x_t)) = \gamma(x_1) \vee ... \vee \gamma(x_{t-1})$. Aufgrund der obigen Fusionsfunktion F' ist dies äquivalent zu $pprev(\gamma(x_t)) = \max_{i=1}^{t-1}\gamma(x_i)$.

[0076] Es gibt eine maximale Menge von Beobachtungen, welche zu einer Hypothese gehören, um hierdurch den Rechenaufwand des obigen pprev-Operators nicht zu groß werden zu lassen.

[0077] Der oben beschriebene Hypothesenpool kann nunmehr in geeigneter Weise über eine Benutzerschnittstelle ausgegeben werden, so dass ein Benutzer durch einen entsprechenden Prioritätswert darüber informiert wird, wie wahrscheinlich für entsprechende Sequenzen an Beobachtungen ein Angriff vorliegt. Um diese Prioritätswerte zu ermitteln, muss bereits ein trainiertes CRF vorliegen. Hierfür werden entsprechende Trainingsdaten benötigt, und im Folgenden wird erläutert, wie diese Trainingsdaten bestimmt werden können. Die Trainingsdaten können a priori durch einen Experten vorgegeben werden und anschließend während der Laufzeit durch neue Beispiele in der Form von Beobachtungen ergänzt werden. Die Trainingsdaten werden dabei über Matrizen beschrieben, welche für vorbestimmte Angriffe angeben, ob diesem Angriff ein Muster zugeordnet werden kann bzw. nicht zugeordnet werden kann bzw. ob keine Information hinsichtlich der Zuordnung des Musters zum Angriff vorliegt. Ein Beispiel einer solchen Matrix ist die nachfolgende Tabelle 4.

Tabelle 4:

	p_1	p_2	p_3	Bedrohung
y_1	m	¬m	¬m	d
y_2	¬m	m	u	n

[0078] Diese Matrix beruht auf Expertenwissen, wobei die einzelnen Spalten die verschiedenen Muster p_1 bis p_3 und die Zeilen verschiedene Angriffe y_1 und y_2 bezeichnen.

[0079] Durch einen entsprechenden Eintrag m wird dabei eine Zuordnung bzw. Übereinstimmung des entsprechenden Musters mit dem Angriff spezifiziert, wohingegen ¬m eine Nicht-Zuordnung bzw. Nicht-Übereinstimmung des entsprechenden Musters mit dem Angriff festlegt. u bezeichnet dabei, dass keine Information hierzu vorliegt. In der hier beschriebenen Ausführungsform werden die einzelnen Angriffe ferner dahingehend charakterisiert, wie gefährlich sie sind. n bezeichnet dabei einen normalen Betrieb des Computernetzes und somit einen gutartigen Angriff, s einen verdächtigen Angriff und d einen gefährlichen Angriff.

[0080] Die Tabelle kann durch Experten weiter ergänzt werden. Der Wert u wird hierbei benötigt, wenn neue Muster und Angriffe hinzugefügt werden. Wird beispielsweise ein neues Muster p_4 und ein neuer Angriff y_3 hinzugefügt, ergibt sich eine Erweiterung der Matrix, die in nachfolgender Tabelle 5 wiedergegeben ist:

Tabelle 5:

	p_1	p_2	p_3	p_4	Bedrohung
y_1	m	¬m	¬m	u	d
y_2	¬m	m	u	u	n
y_3	u	u	u	m	s

[0081] Um die Trainingsdaten nunmehr in geeigneter Weise zum Lernen der Conditional Random Fields (d. h. der entsprechenden Gewichte λ_i) zu berücksichtigen, werden für die Matrix aus Trainingsdaten in an sich bekannter Weise entsprechende gemeinsame Wahrscheinlichkeiten ermittelt, welche in dem Trainings-Algorithmus verwendet werden. Als Trainings-Algorithmus wird der an sich bekannte Improved-Iterative-Scaling-Algorithmus eingesetzt, der in der Druckschrift [3] beschrieben ist.

[0082] Im Folgenden wird die Berechung der entsprechenden gemeinsamen Wahrscheinlichkeiten aus Trainingsdaten kurz umrissen. Es wird ein Satz T von Trainingsbeispielen betrachtet. Jede Reihe der entsprechenden Matrix aus Trainingsdaten repräsentiert ein Trainingsbeispiel. Jedes Trainingsbeispiel $T \in T$ enthält eine Liste an Muster mit den entsprechenden Werten m, ¬m bzw. u, welche durch die Variable v repräsentiert werden. Die dem jeweiligen Trainingsbeispiel zugeordneten Angriffe werden durch die Variable y repräsentiert. In der folgenden Tabelle 6 ist ein Beispiel eines Satzes von Trainingsbeispielen wiedergegeben:

Tabelle 6:

$T \in T$	v	y
T_1	$\{p_1 = m, p_2 = \neg m, p_3 = u\}$	$\{y_2\}$
T_2	$\{p_1 = m, p_2 = m, p_3 = u\}$	$\{y_1\}$
T_3	$\{p_1 = m, p_2 = u, p_3 = m\}$	$\{y_1, y_2\}$

[0083] Es wird angenommen, dass zwei Sätze von Muster-Werten $v = \{p_1 = \{m, \neg m, u\}, ..., p_n = \{m, \neg m, u\}\}$ (d. h. die Substitution der Beobachtungen durch entsprechende Werte m, ¬m bzw. u) dann und nur dann äquivalent sind ($v \equiv v'$), wenn gilt:

$$\neg(\exists i.((p_i \in \mathrm{v} = m \wedge p'_i \in \mathrm{v}' = \neg m) \vee (p_i \in \mathrm{v} = \neg m \wedge p'_i \in \mathrm{v}' = m)))$$

[0084] Die Muster-Werte v können aus einer Beobachtung x für einen Satz von Mustern p wie folgt extrahiert werden:

$$\mathrm{v}(\mathbf{x}) = \left\{p_1(\mathbf{x}) \in \mathbf{p} = \mu\left(p_1(\mathbf{x})\right) \in \{m, \neg m\}, \ldots, p_n(\mathbf{x}) \in \mathbf{p} = \mu\left(p_n(\mathbf{x})\right) \in \{m, \neg m\}\right\}$$

wobei

$$\mu\left(p(\mathbf{x})\right) = \begin{cases} m & falls\ F'\left(p(\mathbf{x})\right) = 1, \\ \neg m & sonst \end{cases}$$

[0085] Die empirische Wahrscheinlichkeit $\tilde{p}(x)$, welche im Improved-Iterative-Scaling-Algorithmus verwendet wird, wird wie folgt berechnet:

$$\tilde{p}(\mathbf{x}) = \frac{\left|\left\{\mathrm{v}' \in \mathbf{T} \middle| \mathrm{v}' \equiv \mathrm{v}(\mathbf{x}) \in \mathbf{T}\right\}\right|}{|\mathbf{T}|}$$

[0086] Die empirische gemeinsame Wahrscheinlichkeit $\tilde{p}(x, y)$ wird bestimmt durch

$$\frac{N(y,\mathbf{T},x)}{|\mathbf{T}|},$$

wobei N(y, T, x) die Anzahl des Auftretens von y mit dem äquivalenten Muster-Wert v(x) in dem Satz T an Trainingsbeispielen ist.

[0087] Diese Wahrscheinlichkeiten werden in dem Gradienten des Improved-Iterative-Scaling-Algorithmus verwendet, um die Gewichte des Conditional Random Fields zu bestimmen. Zwecks näherer Ausführung des Improved-Iterative-Scaling-Algorithmus wird auf Druckschrift [3] verwiesen.

[0088] Wie bereits erwähnt, kann die obige Matrix aus Tabelle 4 bzw. 5 immer weiter durch Expertenwissen ergänzt werden. Jedoch zeigen unspezifizierte Werte u an, dass in einem solchen Fall kein entsprechendes Merkmal generiert wird. Gemäß Tabelle 5 können die Muster p_1, p_2 und p_3 für den Angriff y_3 erfüllt sein oder nicht. y_3 wird dabei auch für Musterkombinationen von y_1 und y_2 betrachtet, was wenig informativ ist, da y_3 in den meisten Fällen nicht ausgeschlossen werden kann. Durch Hinzufügen von Expertenwissen bzw. Referenzdaten aus Beobachtungen können die diskriminativen Eigenschaften von y_3 erhöht werden.

[0089] Im Folgenden bezeichnet |y| die Menge an vorgegebenen Angriffen und |p| die Menge an Muster. Das Hinzufügen eines neuen Angriffs kann den Aussagegehalt der Matrix durch entsprechende Bewertungen für die |p| Muster verbessern. Das Hinzufügen eines neuen Musters ermöglicht es, insgesamt |y| neue Werte für den neuen Angriff in die Matrix hinzuzufügen. Für eine große Anzahl an Angriffen und Mustern kann die entsprechende Ergänzung der Matrix sehr aufwändig sein. Demzufolge ist es vorteilhaft, Referenzdaten zu verwenden, um automatisch entsprechende Werte in der Matrix zu bestimmen, selbst wenn neue Muster hinzugefügt werden. Ferner sind nicht alle Angriffe durch die verwendeten IDS-Systeme bzw. Muster unterscheidbar. Dies kann dadurch berücksichtigt werden, dass Trainingsdaten hinzugefügt werden, welche statistisch interpretiert werden. Zum Beispiel kann p_1 nur in 25% der Fälle y_1 zugeordnet sein und in 75% der Fälle nicht y_1 zugeordnet sein.

[0090] Stehen Referenzdaten in der Form von bereits vorhandenen Sequenzen von Beobachtungen zur Verfügung, können Trainingsdaten über eine Interaktion mit einem Benutzer generiert werden. Dabei gibt ein Benutzer für die entsprechende Sequenz an Beobachtungen ein, welchen Angriff bzw. welche Angriffe er dieser Sequenz zuordnet. Anschließend wird für die Sequenz mit dem oben beschriebenen toleranten Mustervergleich abgeleitet, welche Muster mit dieser Sequenz zu 100% übereinstimmen (Ähnlichkeitsmaß von 1). Für diese Muster wird dann der entsprechende Wert m für die vom Benutzer spezifizierten Angriffe in der Matrix hinterlegt. Für Muster, die nicht bzw. nur teilweise übereinstimmen, wird der Wert ¬m in der Matrix hinterlegt.

[0091] Gegebenenfalls besteht auch die Möglichkeit, dass der Benutzer in dem obigen Annotationsprozess, bei dem er für eine Sequenz von Beobachtungen entsprechende Angriff spezifiziert, automatisch unterstützt wird. Dabei kann die obige Wahrscheinlichkeitsverteilung Pr(y|x) genutzt werden. Ein entsprechender Vorschlag val(y, x) für eine Annotation, gemäß der für eine Beobachtung x = {x_1, ..., x_t} ein vorbestimmter Angriff festgelegt wird, kann wie folgt bestimmt werden:

$$val(y \in \mathbf{y}, \mathbf{x}) = \begin{cases} m & y > \dfrac{\max_{y \in \mathbf{y}} \Pr(y|\mathbf{x}) + \dfrac{1}{|\mathbf{y}|}\sum_{y \in \mathbf{y}} \Pr(\mathbf{y}|\mathbf{x})}{2} \\ \neg m & sonst \end{cases}$$

[0092] Im Rahmen dieser Ausführungsform wird der Benutzer über eine entsprechende Benutzerschnittstelle gefragt, ob er den rechnergestützt generierten Vorschlag annehmen möchte oder Zuweisungen der Angriffe verändern möchte. Die Berechnung von val(y, x) berücksichtigt, dass mehrere Angriffe für eine gegebene Beobachtung möglich sein können. Dies ist jedoch nur eine Heuristik, um den Benutzer zu unterstützen. Die oben beschriebene Matrix mit den darin enthaltenen Trainingsdaten muss ferner nicht widerspruchsfrei sein. Die Wahrscheinlichkeit des obigen unspezifischen Werts u wird im Rahmen der Ermittlung der gemeinsamen Wahrscheinlichkeiten für den Trainings-Algorithmus immer mit 1 gezählt. Das heißt, beim Zählen der Menge von übereinstimmenden und nicht übereinstimmenden Mustern werden die unspezifischen Muster immer mitgezählt. Die Wahrscheinlichkeit, dass ein Muster p mit einem unspezifischen Wert u mit einem Angriff y übereinstimmt, wird auf Pr(y, p_1 = u) = 1 gesetzt und in Kombination mit den anderen Mustern auf Pr(y, p_1 = u, p_2) =

Pr(y, p_2). Die anderen erforderlichen empirischen Wahrscheinlichkeiten, die für das Training benötigt werden, werden auf einfache Weise durch deren relative Frequenz aus der Matrix abgeleitet.

[0093] Wie bereits oben erwähnt, wird das Training der Conditional Random Fields basierend auf dem an sich bekannten Improved-Iterative-Scaling-Algorithmus durchgeführt. In der hier beschriebenen Ausführungsform wurde dabei als Wertebereich für λ [–10;10] festgelegt. Im Rahmen des Algorithmus wird mit $\lambda_i = 0$ begonnen. Bis zu einer Konvergenz des Algorithmus wird für jedes λ_i ein sog. Gewichts-Update-Wert δ_i berechnet, mit dem der Wert λ_i aktualisiert wird (d. h. $\lambda_i = \lambda_i + \delta_i$). Dabei wird sichergestellt, dass λ_i in dem vorgegebenen Wertebereich bleibt. Zur Berechnung des Gewichts-Updates wird das Newton-Verfahren verwendet, mit dem die Gradientengleichung aus der Druckschrift [3] gelöst wird.

[0094] Im Folgenden wird noch dargelegt, wie die oben beschriebenen Prioritäten für die einzelnen Hypothesen aus dem Hypothesenpool ermittelt werden können. Dabei werden auch die Klassifikationen der einzelnen Angriffe in normale (n), verdächtige (s) und gefährliche (d) Angriffe berücksichtigt. Die Berechnung der Priorität basiert auf probabilistischer Inferenz und wird durch nachfolgende Definition 9 beschrieben.

Definition 9:

[0095] Die Priorität Pri wird durch den Logarithmus der Wahrscheinlichkeit Pr (über CRF bestimmt) berechnet, wonach die gegebenen Beobachtungen über die Zeit t (einer Hypothese) $x = \{x_1, ..., x_t\}$ als Ganzes zu einem gefährlichen oder verdächtigen Angriff gehören (d. h. $Pr(T = d \vee T = s|x) = \Pi_{i=1}^{t} Pr(T = d \vee T = s|x_i)$), geteilt durch die Wahrscheinlichkeit, dass alle Beobachtungen über die Zeit t einen gutartigen Angriff und somit einen normalen Betrieb des Computernetzes darstellen (d. h. $Pr(T = n|x) = \Pi_{i=1}^{t} Pr(T = n|x_i)$):

$$\mathrm{Pri}(\mathbf{x}) = \log_{10}\left(\frac{\Pr(T = d \vee T = s|\mathbf{x}) + 2\Pr(T = d|\mathbf{x})}{\Pr(T = n|\mathbf{x})}\right)$$

[0096] Dabei bezeichnet T die Menge aller möglichen Bedrohungen T = {d, s, n}. Ferner bezeichnet x_i in den bedingten Wahrscheinlichkeiten alle zurückliegenden Beobachtungen (beginnend mit x_i) bis zur Beobachtung x_i. Die obige Priorität Pri enthält eine spezielle Gewichtung der verdächtigen und gefährlichen Bedrohungen. Die auf diese Weise ermittelte Priorität nimmt stark zu bei kleinen Wahrscheinlichkeiten von Beobachtungen, die zu normalen (gutartigen) Angriffen gehören. Sie nimmt ferner für gefährliche und verdächtige Angriffe zu, jedoch nimmt sie stärker für explizite gefährliche Angriffe zu. Deshalb bevorzugt der Hypothesenpool gefährliche und verdächtige Angriffe gegenüber normalen Hypothesen, insbesondere wenn gefährliche Angriffe auftreten.

[0097] Das erfindungsgemäße Verfahren wurde von den Erfindern im Rahmen von Simulationen getestet. Es konnte in der Tat bestätigt werden, dass die mit der obigen Formel ermittelten Prioritäten sehr gut den Sachverhalt widerspiegeln, dass Sequenzen mit hoher Priorität gefährliche Angriffe auf das Computernetz darstellen.

[0098] Die im Vorangegangenen beschriebenen Ausführungsformen der Erfindung weisen eine Reihe von Vorteilen auf. Insbesondere wird eine intelligente Korrelation von Beobachtungen bzw. Ereignissen ermöglicht, die über statistisch modellierte Regeln hinausgeht. Im Gegensatz zu herkömmlichen Verfahren können auch bisher nicht bekannte Angriffssequenzen durch die Abstraktion entsprechender Muster korreliert und bewertet werden. Dadurch wird eine schnelle Anpassung auf neue Bedrohungen ermöglicht. Zudem können Erfahrungen von bereits versuchten Angriffen verwendet werden, um das System einfach unternehmensspezifisch anzupassen. Ein weiterer Vorteil ist, dass konkretes Angriffswissen in der Form von Beobachtungen nicht notwendiger Weise vorliegen muss, jedoch optional zur Optimierung des Modells verwendet werden kann. Dies ermöglicht den Austausch von modelliertem Wissen auch über Unternehmensgrenzen hinweg und die Anpassung an das jeweilige Unternehmen durch konkrete Beobachtungen im Nachhinein. Die Erfindung kann somit als lernfähiges, abstraktionsfähiges und regel-inspiriertes System angesehen werden.

Literaturverzeichnis:

[1] DE 694 28 631 T2
[2] US 7,039,953 B2
[3] Adam L. Berger, Vincent J. Della Pietra, Stephen A. Della Pietra. A maximum entropy approach to natural language processing. Band 2, Seiten 39 bis 71, Cambridge, MA, USA, März 1996. MIT Press.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Nicht-Patentliteratur

- http://www.fides-security.org/ [0035]

Patentansprüche

1. Verfahren zum rechnergestützten Erkennen von Angriffen auf ein Computernetz, bei dem:
a) Sequenzen (x) von einer oder mehreren Beobachtungen im Computernetz detektiert werden;
b) eine jeweilige Sequenz (x) mit Muster (p1, p2, p3) von jeweils einer oder mehreren semantischen Aussagen (γ_1^1, γ_2^1, γ_3^1) aus einer ontologischen Wissensbasis (KB) verglichen wird, wobei für jedes Muster ein oder mehrere Ähnlichkeitsmaße (F') des Musters (p1, p2, p3) mit einer oder mehreren Gruppen von Beobachtungen aus der jeweiligen Sequenz (x) ermittelt wird;
c) für eine jeweilige Sequenz (x) basierend auf den Ähnlichkeitsmaßen (F') der Muster (x) eine oder mehrere Wahrscheinlichkeitsverteilungen (Pr) für eine Mehrzahl von vorgegebenen Angriffen (y) ermittelt werden, wobei eine jeweilige Wahrscheinlichkeitsverteilung (Pr) die Wahrscheinlichkeiten des Vorliegens der vorgegebenen Angriffe (y) für eine Gruppe von Beobachtungen der jeweiligen Sequenz (x) repräsentiert, wobei die Wahrscheinlichkeitsverteilung oder Wahrscheinlichkeitsverteilungen (Pr) auf einem probabilistischen Modell basieren, das mittels Trainingsdaten (TD) trainiert ist, gemäß denen Muster (p1, p2, p3) aus der Wissensbasis (KB) mit vorgegebenen Angriffen (y) korreliert werden.

2. Verfahren nach Anspruch 1, bei dem das probabilistische Modell auf Conditional Random Fields beruht.

3. Verfahren nach Anspruch 1 oder 2, bei dem die Ähnlichkeitsmaße (F') in Schritt b) derart bestimmt werden, dass zunächst ermittelt wird, ob ein jeweiliges Muster (p1, p2, p3) aus der ontologischen Wissensbasis (KB) mit der entsprechenden Gruppe von Beobachtungen der jeweiligen Sequenz (x) übereinstimmt, wobei im Falle, dass das jeweilige Muster (p1, p2, p3) mit der Gruppe von Beobachtungen übereinstimmt, diesem Muster das maximale Ähnlichkeitsmaß (F') mit der Gruppe von Beobachtungen zugeordnet wird, und wobei im Falle, das das jeweilige Muster (p1, p2, p3) nicht mit der Gruppe von Beobachtungen übereinstimmt, die semantischen Aussagen (γ_1^1, γ_2^1, γ_3^1) in diesem Muster schrittweise zu allgemeineren Aussagen abstrahiert werden, wobei nach jedem Abstraktionsschritt die Gruppe von Beobachtungen mit dem aus dem Abstraktionsschritt resultierenden abstrahierten Muster verglichen wird, wobei im Falle, dass das abstrahierte Muster mit der Gruppe von Beobachtungen übereinstimmt, ein Wert einer Ähnlichkeitsfunktion, der sich für das abstrahierte Muster ergibt, mit dem Ähnlichkeitsmaß des jeweiligen Musters (p1, p2, p3) gleichgesetzt wird, wobei der Wert der Ähnlichkeitsfunktion (F') umso niedriger ist, je allgemeiner das abstrahierte Muster ist, und wobei im Falle, dass kein abstrahiertes Muster gefunden werden kann, welches mit der Gruppe von Beobachtungen (x) übereinstimmt, dem jeweiligen Muster ein minimales Ähnlichkeitsmaß mit der Gruppe von Beobachtungen zugeordnet wird.

4. Verfahren nach einem der vorhergehenden Ansprüche, bei dem die semantischen Aussagen (γ_1^1, γ_2^1, γ_3^1) der Muster (p1, p2, p3) in der ontologischen Wissensbasis (KB) durch eine Beschreibungslogik repräsentiert werden, welche als Entitäten Klassen, Relationen, Individuen in der Form von instantiierten Klassen und Variablen in der Form von Stellvertretern für mehrere Individuen umfasst, wobei eine semantische Aussage (γ_1^1, γ_2^1, γ_3^1) eine Relation zwischen zwei Entitäten beschreibt, von denen eine Entität ein Individuum oder eine Variable ist, wobei die semantische Aussage oder die semantischen Aussagen (γ_1^1, γ_2^1, γ_3^1) in zumindest einem Teil der Muster (p1, p2, p3) mit einem oder mehreren Operatoren kombiniert sind, wobei die zur Kombination verwendbaren Operatoren logische Operatoren und insbesondere die Operatoren UND, ODER und Negation umfassen.

5. Verfahren nach Anspruch 4, bei dem die zur Kombination verwendbaren Operatoren ferner zeitliche Operatoren umfassen, mit denen eine semantische Aussage (γ_1^1, γ_2^1, γ_3^1) für eine oder mehrere Beobachtungen auf eine oder mehrere zeitlich zurückliegende Beobachtungen abgebildet wird.

6. Verfahren nach Anspruch 4 oder 5 in Kombination mit Anspruch 3, bei dem in Schritt b) ein Muster (p1, p2, p3), welches eine oder mehrere Negationen für eine oder mehrere Kombinationen von semantischen Aussagen (γ_1^1, γ_2^1, γ_3^1) umfasst, vor der Durchführung der schrittweisen Abstraktion basierend auf den De Morgan'schen Regeln derart gewandelt wird, dass das gewandelte Muster nur noch eine oder mehrere Negationen für eine oder mehrere einzelne semantische Aussagen enthält, wobei die Abstraktion einer Negation einer semantischen Aussage (γ_1^1, γ_2^1, γ_3^1) durch eine Tautologie repräsentiert wird.

7. Verfahren nach einem der vorhergehenden Ansprüche, bei dem die vorgegebenen Angriffe (y) in bösartige, von unbefugten Dritten durchgeführte Angriffe und in gutartige, den Normalbetrieb des Computernetzes widerspiegelnde Angriffe kategorisiert werden und einer jeweiligen Sequenz (x) von Beobachtungen ein Prioritätswert zugewiesen wird, wobei der Prioritätswert umso größer ist, je höher die Wahrscheinlichkeit ist, dass alle Gruppen von Beobachtungen der jeweiligen Sequenz (x) zu einem bösartigen Angriff gehören, und wobei der Prioritätswert umso niedriger ist, je größer die Wahrscheinlichkeit ist, dass alle Gruppen von Beobachtun-

gen der jeweiligen Sequenz (x) zu einem gutartigen Angriff gehören, wobei diese Wahrscheinlichkeiten aus den in Schritt c) ermittelten Wahrscheinlichkeiten bestimmt werden.

8. Verfahren nach Anspruch 7, bei dem die jeweiligen Sequenzen (x) mit deren Prioritätswerten als Hypothesen (h) in einem Hypothesenpool (HP) über eine Benutzerschnittstelle ausgegeben werden, wobei ein Hypothesenpool (HP) eine vorbestimmte Anzahl an Hypothesen (h) mit den größten Prioritätswerten enthält.

9. Verfahren nach einem der vorhergehenden Ansprüche, bei dem das probabilistische Modell in zeitlichen Abständen mit neuen Trainingsdaten (TD) trainiert wird, wobei die neuen Trainingsdaten (TD) aus neu detektierten Sequenzen (x) von Beobachtungen abgeleitet werden.

10. Verfahren nach Anspruch 9, bei dem die neuen Trainingsdaten (TD) derart ermittelt werden, dass für eine neu detektierte Sequenz (x) über eine Benutzerschnittstelle eine Eingabe eines Benutzers abgefragt wird, über welche der Benutzer die neu detektierte Sequenz (x) als übereinstimmend mit einem oder mehreren vorgegebenen Angriffen (y) spezifiziert, wobei basierend auf dem Vergleich gemäß Schritt b) übereinstimmende Muster (p1, p2, p3) mit dem maximalen Ähnlichkeitsmaß mit der neu detektierten Sequenz ermittelt werden und diese übereinstimmenden Muster (p1, p2, p3) in den neuen Trainingsdaten (TD) als übereinstimmend mit dem oder den über den Benutzer als übereinstimmend spezifizierten Angriffen eingestuft werden.

11. Verfahren nach Anspruch 10, bei dem basierend auf den für die neu detektierte Sequenz in Schritt c) ermittelten Wahrscheinlichkeitsverteilungen (Pr) dem Benutzer über die Benutzerschnittstelle ein Vorschlag für die Spezifikation der neu detektierten Sequenz (x) als übereinstimmend mit einem oder mehreren vorgegebenen Angriffen (y) ausgegeben wird.

12. Verfahren nach einem der vorhergehenden Ansprüche, bei dem das probabilistische Modell basierend auf dem Improved-Iterative-Scaling-Algorithmus trainiert ist oder trainiert wird.

13. System zum rechnergestützten Erkennen von Angriffen auf ein Computernetz, umfassend:
– ein Detektionsmittel, um Sequenzen (x) von Beobachtungen im Computernetz zu detektieren;
– ein Vergleichsmittel, um eine jeweilige Sequenz (x) mit Mustern (p1, p2, p3) von jeweils einer oder mehreren semantischen Aussagen (γ_1^1, γ_2^1, γ_3^1) aus einer ontologischen Wissensbasis (KB) zu vergleichen, wobei für jedes Muster ein oder mehrere Ähnlichkeitsmaße (F') des Musters (p1, p2, p3) mit einer oder mehreren Gruppen von Beobachtungen aus der jeweiligen Sequenz (x) ermittelt wird;
– ein Berechnungsmittel, um für eine jeweilige Sequenz (x) basierend auf den Ähnlichkeitsmaßen der Muster eine oder mehrere Wahrscheinlichkeitsverteilungen (Pr) für eine Mehrzahl von vorgegebenen Angriffen (y) zu ermitteln, wobei eine jeweilige Wahrscheinlichkeitsverteilung die Wahrscheinlichkeiten der jeweiligen vorgegebenen Angriffen für eine Gruppe von Beobachtungen der jeweiligen Sequenz (x) repräsentiert, wobei die Wahrscheinlichkeitsverteilung oder Wahrscheinlichkeitsverteilungen (Pr) auf einem probabilistischen Modell basieren, das mittels Trainingsdaten (TD) trainiert ist, gemäß denen Muster (p1, p2, p3) aus der Wissensbasis (KB) mit vorgegebenen Angriffen (y) korreliert werden.

14. System nach Anspruch 13, welches derart ausgestaltet ist, dass mit dem System ein Verfahren nach einem der Ansprüche 2 bis 12 durchführbar ist.

15. Computerprogrammprodukt mit einem auf einem maschinenlesbaren Träger gespeicherten Programmcode zur Durchführung eines Verfahrens nach einem der Ansprüche 1 bis 12, wenn der Programmcode auf einem Computer ausgeführt wird.

16. Computerprogramm mit einem Programmcode zur Durchführung eines Verfahrens nach einem der Ansprüche 1 bis 12, wenn der Programmcode auf einem Computer ausgeführt wird.

Es folgen 2 Blatt Zeichnungen

Anhängende Zeichnungen

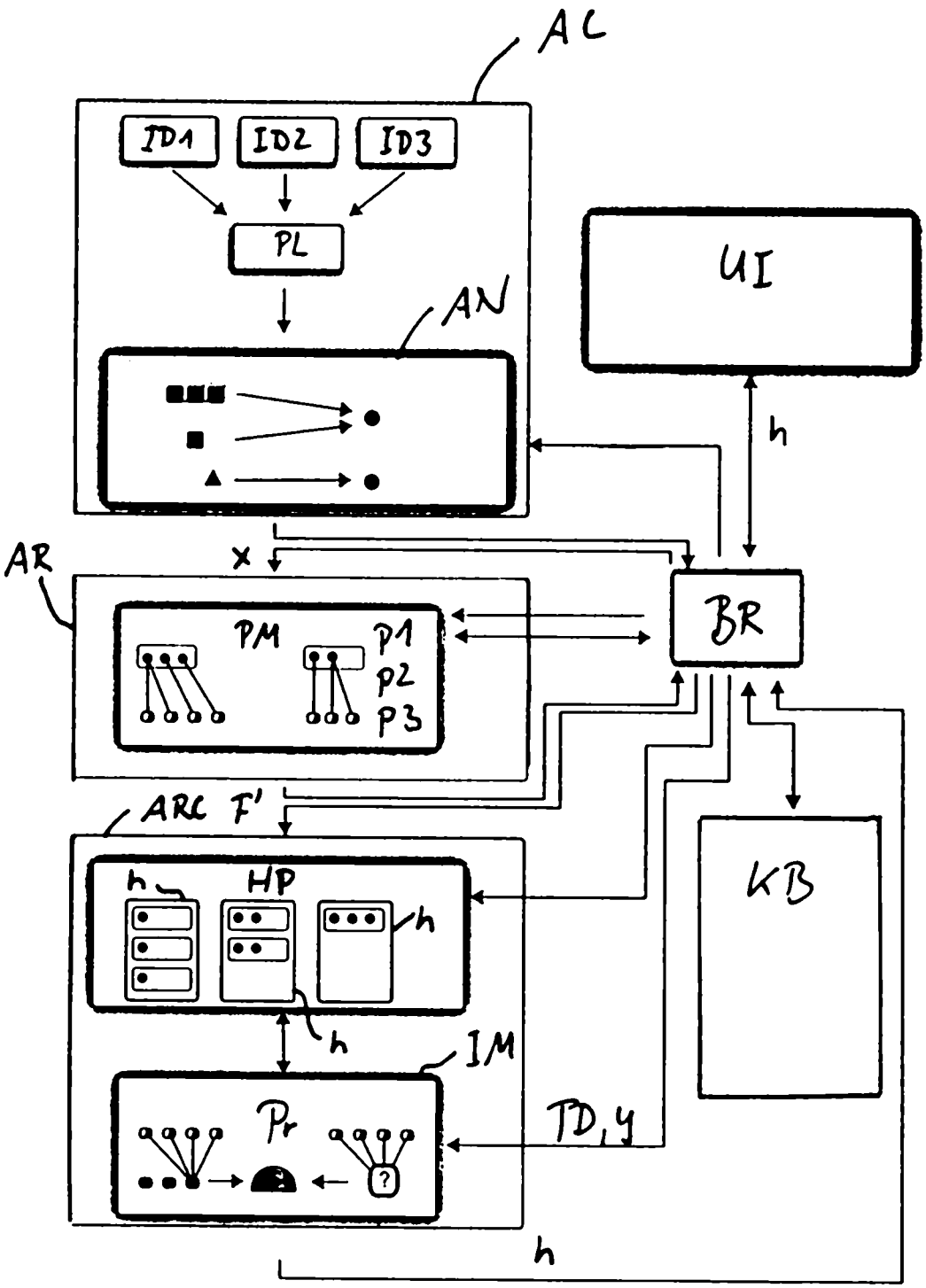


Fig. 1

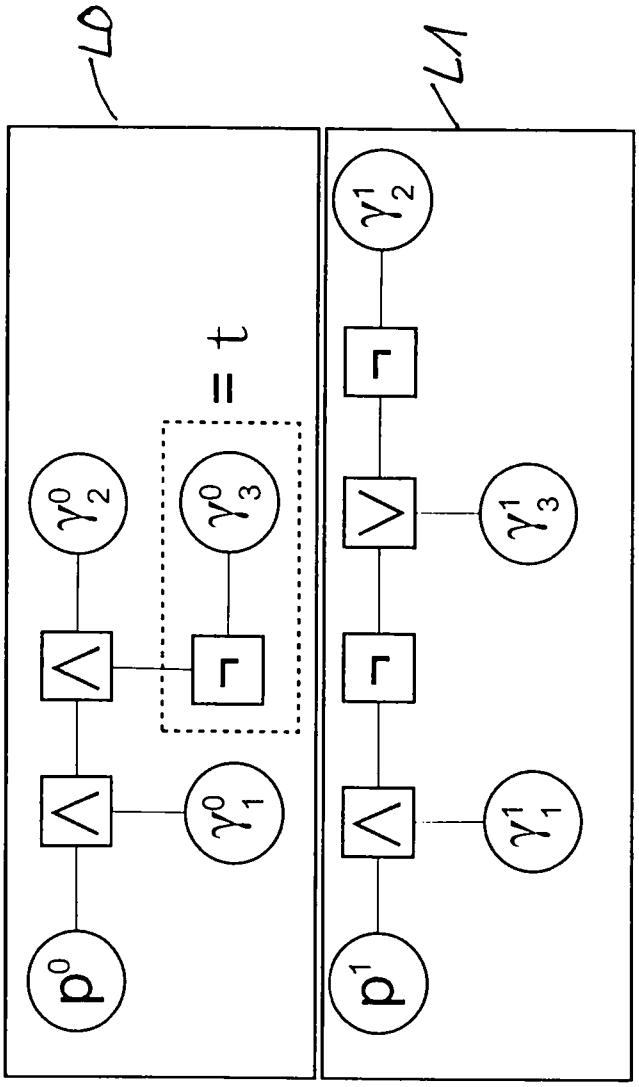

L0
L1
= t
γ_2^0
γ_3^0
γ_1^0
p^0
γ_2^1
γ_3^1
γ_1^1
p^1

Fig. 2