Überblick: Wozu braucht man Agenten

SS2004, Michael Grenz xi2306@fh-wedel.de
Wiesbaden, den 27.04.2004

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ACL	Agent Communication Language
bzw.	beziehungsweise
etc.	et cetera
ff.	fortfolgende
FIPA	Foundation for Intelligent Physical Agents
ggf.	gegebenenfalls
ID	Intrusion Detection
IP	Internet Protocol
KI	Künstliche Intelligenz
p.a.	per anno (pro Jahr)
PC	Personal Computer
S.	Seite
TCP	Transmission Control Protocol
u.a.	und andere/unter anderem
usw.	und so weiter
u.U.	unter Umständen
vgl. a.	vergleiche auch
z.B.	zum Beispiel

1 Einführung

1.1 Zielsetzung und Abgrenzung

Das Ziel dieser Ausarbeitung besteht darin, den Leser mit dem Begriff des Softwareagenten vertraut zu machen und ihm einige mögliche Einsatzgebiete für agentenbasierte Lösungen vorzustellen. Der Schwerpunkt dieser Ausarbeitung ist die Untersuchung der Möglichkeit des Einsatzes von Agenten zum Schutz von Systemen. Im Rahmen dieser Arbeit wird ein Entwurf eines Mulitiagentensystems vorgestellt, dessen Konzepte auf ihre Alltagstauglichkeit jedoch nicht näher untersucht werden sollen.
Das Ergebnis dieser Arbeit ist in der kritischen Auseinandersetzung mit der Nützlichkeit des agentenbasierten Ansatzes im Bereich Systemsicherheit zu sehen.

1.2 Vereinbarungen

Um die Lesbarkeit dieser Arbeit zu erhöhen, werden folgende Vereinbarungen getroffen:

- Begriffe, die im Glossar erläutert sind, werden im Text stets *kursiv* ausgezeichnet.
- Auszüge aus diversen Logdateien und dem Quellcode werden durch `Schreibmaschinenschrift` hervorgehoben.

2 Agentensysteme - Grundlagen

2.1 Definition des Agentenbegriffs

Bevor man sich damit beschäftigt, wie der Einsatz von Agentensystemen zur Erhöhung von Systemsicherheit beitragen kann, muss der Begriff eines Agenten präzisiert werden. Doch bereits an dieser Stelle scheitert die Fachwelt an einer einheitlichen Begriffsdefinition. Die intuitive Definition eines Agenten ist relativ einfach. Ein Softwareagent ist danach ein Programm, das auf Anforderung und Eingabe von Daten hin eine Dienstleistung erbringt. Z.B. ist Druckerdämon ein einfacher Softwareagent. Tatsächlich existieren mehrere Agentenbegriffe, wobei manche Definitionen sich sehr stark voneinander unterscheiden. So versteht Cheong[1] unter einem Agenten ein beliebiges Programm, welches menschliches Verhalten nachahmt, indem es Aktionen ausführt, die ein Mensch an Stelle des Agenten ausführen würde. Eine etwas andere Definition präsentieren Stuart Russel und Peter Norvig[2]. Sie verstehen unter einem Agenten eine Einheit, die in der Lage ist, deren Umgebung durch Sensoren wahrzunehmen und mit Hilfe von Aktoren in dieser Umgebung zu agieren. Von einer etwas anderen Definition des Agentenbegriffs geht Michael Wooldridge aus[3]. Danach ist ein beliebiges Programm genau dann ein Agent, wenn es in der Lage, autonom zu handeln und mit anderen Systemen zu kommunizieren. Die in mancher Literaturquelle zu findenden Eigenschaften „Lernfähigkeit“ und „Mobilität“ eines Agentensystems sind, seiner Meinung nach, für die Definition des Agentenbegriffs optional.

Ein Agent muss nach Wooldridge[4] also drei Eigenschaften erfüllen. Er muss in

[1] Vgl. a. [CHEONG 1996] S. 5.
[2] Vgl. a. [RUSSEL 2003] S. 32 f.
[3] Vgl. a. [WOOLDRIDGE 2002] S. XI.
[4] Vgl. a. [WOOLDRIDGE 2002] S. 23.

der Lage sein, Perzepte wahr zu nehmen und auf diese reagieren zu können. Ein Agent muss zielgerichtet handeln können und es muss mit den anderen Agenten kommunizieren können. Tanenbaum[5] unterscheidet außerdem zwischen mehreren Agententypen, wobei er folgende Eigenschaften hervorhebt:

Eigenschaft	**Beschreibung**
Autonom	Kann eigenständig agieren
Reaktiv	Reagiert rechtzeitig auf Änderungen in seiner Umgebung
Proaktiv	Initiiert Aktionen, die seine Umgebung beeinflussen
Kommunikativ	Kann Informationen mit Benutzern und anderen Agenten austauschen
Mobil	Kann von einem System auf ein anderes migrieren
Adaptiv	Lernfähig

Tabelle 1: Agentensysteme können durch folgende Eigenschaften unterschieden werden.

2.1.1 Abgrenzung zum Client/Server-Modell

Die beschriebenen Eigenschaften eines Agenten machen den Unterschied zu dem klassischen Client/Server-Modell deutlich. Der Client übermittelt einen Auftrag mit den dazugehörigen Daten an den Server. Dieser arbeitet den Auftrag ab und sendet die Ergebnisse an den Client zurück. Während der Ausführung des Auftrages muss eine permanente Verbindung zwischen den beiden Systemen bestehen. Dies ist mit einem höheren Datenvolumen verbunden und ist in vielen Fällen gar nicht erst möglich[6]. Das autonome und proaktive Handeln von Agenten macht die

[5] Vgl. a. [TANENBAUM 2003] S. 203.

[6] Man denke beispielsweise an die Weltraumroboter, deren Verbindung mit der Steuerzenztrale in der Regel aufgrund physikalischer Gegebenheiten sporadisch und von kurzer Dauer ist.

permanente Verbindung zwischen den beiden Systemen überflüssig, denn Agenten sind nicht nur in der Lage selbständige, sondern auch langfristig-sinnvolle Entscheidungen zu treffen.

2.1.2 Abgrenzung zu Expertensystemen

Die klassischen Expertensysteme besitzen keine Perzeptoren und Aktoren; sie sind nicht in einer Umgebung integriert. Expertensysteme besitzen daher keinerlei „sozialen Fähigkeiten“ und sind nicht fähig, reaktiv und zielgerichtet zu handeln.

2.1.3 Agenten und Künstliche Intelligenz (KI)

Wichtig ist die Abgrenzung zwischen dem typischen Verständnis von KI und dem Agentenbegriff: KI beschäftigt sich vorwiegend mit der Nachahmung menschlicher Fähigkeiten, wie das Lernen, Erkennen von Mustern etc.. Agentensysteme nutzen dagegen lediglich die Errungenschaften von KI, um den Maschinen, eine unabhängige Entscheidungsfindung zu ermöglichen. Außerdem vernachlässigt KI die sozialen Eigenschaften eines Systems, wobei die Kommunikationsfähigkeit zu den Schlüsseleigenschaften eines Agenten gehört.

2.2 Multiagentensysteme

Richtig interessant wird die Diskussion über Agentensysteme dann, wenn mehrere Agenten im Spiel sind. Diese werden zu einem Multiagentensystem, wenn die einzelnen Agenten in der Lage sind, miteinander zu kommunizieren, was z.B. mit Hilfe des Nachrichtenaustauschs geschehen kann. Der einzelne Agent agiert,

um die Interessen seines Auftraggebers zu vertreten[7]. Damit er dies auch in einer Multiagentenumgebung machen kann, müssen die einzelnen Agenten in der Lage sein, miteinander zu kooperieren, die eigenen Handlungen zu koordinieren und Verhandlungen zu führen. Dieses Erkenntnis macht es uns deutlich, dass bei der Entwicklung eines Agentensystems nicht nur sein interner Aufbau (agent design) sondern auch seine „sozialen Fähigkeiten“ (society design) von Bedeutung sind[8].

2.3 Wirtschaftlichkeitsaspekte des Einsatzes von Agentensystemen

Im Zusammenhang mit Agentensystemen ist außerdem die Wirtschaftlichkeitsbetrachtung sinnvoll, denn aus dem dezentralen Aufbau und der Fähigkeit von Agenten, auch ohne ständige Überwachung von der Seite des Inhabers zu handeln, können Kosteneinsparungen resultieren. Ein erfolgreiches Beispiel für die Kosteneinsparungen durch den Einsatz von Agentensystemen demonstriert Michael Wooldridge an der NASAs Mission „Deep Space 1“. Im Jahr 1998 wurden von NASA zum ersten Mal Agenten zur Erforschung des Weltalls eingesetzt, was sich sehr positiv auf der Kostenstruktur der Projekte widerspiegelte. Unter anderem konnte die dreihundert Mann starke Überwachungscrew für die Roboter, die in den vorhergehenden Missionen einen großen Teil der Kostenbelastung darstellte, wegrationalisiert werden. Ein weiteres wesentlich alltagstauglicheres Beispiel zur Verwendung von Agentensystemen ist bei ihm einige Zeilen später zu finden. Ein Ausschreibungs- und Verhandlungsassistent, der die Angebote von mehreren Urlaubsanbietern einholt, mit ihnen verhandelt und anschließend eine für seinen

[7] Selbstverständlich können sich die Interessen mehrerer Auftraggeber voneinander unterscheiden.

[8] Vgl. a. [WOOLDRIDGE 2002] S. 3 ff.

Auftraggeber maßgeschneiderte Lösung präsentiert, würde uns bei der Suche nach dem optimalen Urlaubsziel viel Zeit und Ärger ersparen. Besonders hoch könnten außerdem die Einsparungen durch den Einsatz von Agentensystemen in den Ausschreibungsverfahren der öffentlichen Hand sein[9]. Die durch den Einsatz von Agentensystemen beschleunigte automatisierte Vorgehensweise würde die Bürokratiekosten drastisch reduzieren, so dass die Steuergelder eine vernünftigere Verwendung finden könnten[10].

2.4 Agentenumgebungen

Im Abschnitt 2.1 haben wir den Agenten als ein System definiert, welches in der Lage ist, in einer Umgebung autonom zu agieren, um ein bestimmtes Ziel zu erreichen. Die Umgebung, in der ein Agent integriert ist, ist von entscheidender Bedeutung für seinen Aufbau, denn sie bestimmt, welchen Anforderungen dessen Design entsprechen muss und welche Komplexität es erreicht. Eine sehr ausführliche Beschreibung der Eigenschaften von Agentenumgebungen geben Stuart Russel und Peter Norvig[11]. Sie unterscheiden zwischen folgenden Umgebungseigenschaften:

- **zugänglich („accessible“) / unzugänglich („inaccessible“)**
  In einer zugänglichen Umgebung kann ein Agent alle Informationen über seine Umgebung schnell und zuverlässig mit Hilfe seiner Perzeptoren erhalten. Auf die meisten Anwendungsbereiche der realen Welt trifft dies nicht zu.

- **deterministisch („deterministic“)/nicht-deterministisch („non-deterministic“)**
  Von einer deterministischen Umgebung wird gesprochen, wenn jede Aktion

[9] Die bekanntlich sehr langwierig und so ineffizient sind, dass deren Nutzen häufig angezweifelt wird.
[10] Vgl. a. [WOOLDRIDGE 2002] S. 5 f..
[11] Vgl. a. [RUSSEL 2003] S. 38 ff..

eines Agenten zu einem genau bestimmbaren eindeutigen Zustand führt. Die reale Welt stellt keine deterministische Umgebung dar.

- **episodisch („episodic“) / nicht-episodisch („non-episodic“)**
  In einer episodischen Umgebung besteht kein Zusammenhang zwischen der aktuellen Aktion und den vorhergehenden Aktionen, so dass jede Aufgabe ohne Kenntnis der vergangenen und zukünftigen Ereignisse bewältigt werden kann.

- **statisch („static“) / dynamisch („dynamic“)**
  Eine dynamische Umgebung wird durch eine Vielzahl von Faktoren beeinflußt, die sich durchaus der Kontrolle eines Agenten entziehen können. Eine statische Umgebung wird lediglich durch die Agentenaktionen verändert und ist deswegen wesentlich leichter zu handhaben.

- **diskret („discrete“) / kontinuierlich („continous“)**
  Eine diskrete Umgebung läßt nur eine feste endliche Anzahl von Zuständen und Aktionen zu und kann in der Regel relativ leicht beschrieben werden. Die Beschreibung kontinuierlicher Umgebungen kann dagegen nur wesentlich schwerer oder gar nicht erfolgen.

Nach diesem Schema stellen nicht-zugängliche, nicht-deterministische, nicht-episodische, dynamische und kontinuierliche Umgebungen den Agentendesigner vor Probleme, die nur mit einem erheblichen Aufwand zu lösen sind.

Die Schlüsselproblematik eines Agentensystems besteht darin, zu bestimmen, welche Aktionen es ausführen soll, um seine Designziele zu erfüllen. Besonders deutlich wird diese Problematik, wenn man bedenkt, dass Entscheidungen auch langfristige Konsequenzen bedingen können. Dies soll anhand eines einfachen Beispiels veranschaulicht werden. Ein Bandbreitenmanagement-Agent verteilt eine

knappe Ressource (z.B. eine Leitungskapazität) zwischen mehreren Systemen. Würde dieser Agent bestimmte Clients im Konfliktfall konsequent bevorzugen und ihnen die Leitungskapazität zuweisen, könnte seine Entscheidung, kurzfristig gesehen, absolut richtig sein[12]. Auf längere Sicht könnte es allerdings bedeuten, dass manche Clients gar keine Ressourcen zugewiesen bekämen, was nicht dem Designziel des Agenten entsprechen würde.

2.5 Anwendungsgebiete für Agentensysteme

Agentensysteme können aufgrund ihrer Eigenschaften auf vielfältige Weise eingesetzt werden. Besonders nützlich sind Agentensysteme, wenn man in einem komplexen verteilten System Dienstleistungen in selbständigen Einheiten verfügbar machen will. Die zur Verfügung gestellten Dienste können relativ einfach (z.B. Datenbankabfragen), jedoch auch von komplexer Natur sein[13]. Auf Interaktion von Agenten mit verschiedenen Aufgaben und Fähigkeiten basierend können komplexere Softwaresysteme gebaut werden. Die Vorteile von Agentensystemen liegen in deren Modularisierung, Skalierbarkeit und Erweiterbarkeit. Am häufigsten werden Agenten in folgenden Bereichen eingesetzt:

- **Verteilte/konkurrente Systeme** Mögliche Einsatzgebiete sind: Flugverkehrsüberwachung, Business Process Management, verteiltes Sensoring, Produktionsüberwachung.
- **Netzwerke** Informationssammlung und -management.
- **Human-Computer Interfaces** Nachrichtenassistenten, Spam-Filtering.

[12] Denn die Leitungskapazitäten wären in diesem Fall ebenfalls permanent ausgelastet.
[13] Planungsaufgaben gehören z.B. zu den komplexeren Diensten.

Im Folgenden werden einige Anwendungsgebiete von Agentensystemen näher diskutiert.

2.5.1 Geschäftsprozessoptimierung

In seinem Buch beschreibt Michael Wooldridge den Einsatz eines Multiagentensystems zwecks Geschäftsprozessbeschleunigung. So werden in einem Großunternehmen mit klar definierten Geschäftsprozessen mehrere Agenten eingesetzt, um den Ablauf der Geschäftsprozesse zu beschleunigen. Um einem Kunden möglichst schnell ein, für das Unternehmen wirtschaftliches, Angebot zu machen, verhandeln mehrere Agenten miteinander, wobei jeder dieser Agenten, für die von ihm zu vertretende Organisation, Abteilung oder Person, Angebote einholen und abgeben darf. Um sinnvolle Verhandlungen zu ermöglichen, werden Agenten mit bestimmten Ressourcen ausgestattet, wodurch eine Priorisierung unternehmensinterner Interessen möglich wird. In der Regel können Geschäftsprozesse nur sehr schwer durch ein zentrales System gesteuert werden. Durch den Einsatz eines Multiagentensystems kann das natürliche Verhalten menschlicher Geschäftspartner nachgeahmt werden.

2.5.2 Verteiltes Sensoring

Beim verteilten Sensoring werden mehrere mit Sensoren[14] ausgestattete Agenten eingesetzt, um ein Gebiet (z.B. einen Flughafen) zu überwachen. Die Aufgabe der Systeme besteht darin, sämtliche Bewegungen der überwachten Objekte/Subjekte zu verfolgen. Diese Aufgabe kann durch die Kommunikation von Agenten untereinander effizienter erfüllt werden. So erleichtert die Warnung eines Agenten über den zu erwartenden Eintritt des Objektes/Subjektes in das Überwachungsgebiet

[14] z.B. Kameras und Bewegungsmelder

eines anderen Agenten dessen Aufgabe beträchtlich.

2.5.3 Informationssammlung und -verwaltung

Durch die rasche Entwicklung des Internet werden wir zunehmend mit dem Problem der Informationsüberflutung konfrontiert. Eine sinnvolle Nutzung des aktuellen Medienangebots ist heute ohne Nutzung von Suchmaschinen kaum möglich. Dabei stellen uns diese lediglich sehr rudimentäre Funktionalitäten zur Verfügung; die Suchergebnisse können relativ leicht durch gezielte Manipulationen böswilliger Subjekte beieinträchtigt werden. Außerdem wird die Skalierbarkeit der jetzigen Lösungen des Öfteren bezweifelt. Personale Informationsagenten können uns die Suche nach den Informationen beträchtlich erleichtern, indem sie für uns eine personifizierte Suche durchführen und uns anschließend lediglich die für uns relevanten Information präsentieren.[15]

2.5.4 E-Commerce

Auf der Suche nach dem preisgünstigsten Produkt können Agenten uns ebenfalls behilflich sein. Ein Produkt kann Angebote mehrerer Händler einholen, diese miteinander vergleichen und seinem Auftraggeber anschließend das günstigste präsentieren. Außerdem können Agenten bei der Durchführung von Online-Auktionen eingesetzt werden (auction bots).

[15] Vgl. a. [WOOLDRIDGE 2002] S. 248 f..

2.5.5 Weitere Einsatzgebiete

Zusätzlich zu den aufgezählten Anwendungsgebieten, können Agenten in den Bereichen „Human-Computer Interfaces“, „soziale Simulationen“, „virtuelle Umgebungen“, „industrielles Systemmanagement“, „Weltraumforschung“, „Flugverkehrsüberwachung“[16] etc. eingesetzt werden. An dieser Stelle soll auf die genannten Einsatzgebiete jedoch nicht näher eingegangen werden. Nähere Informationen zu diesem Thema entnehmen Sie bitte den aufgeführten Literaturquellen.

2.6 Zusammenfassung

In diesem Kapitel wurde der Begriff eines Agenten diskutiert und Unterschiede des agentenbasiereten Ansatzes zu den anderen, dem Leser bereits bekannten, Konzepten verdeutlicht. Der Beschreibung von Agentenumgebungen folgte eine Auflistung der möglichen Anwendungsbereiche für ein Agentensystem, die anhand einiger Beispiele verdeutlicht wurden.

[16] Vgl. a. [WOOLDRIDGE 2002] S. 245 ff..

3 Angriffserkennung

3.1 Methoden der Angriffserkennung

Heutzutage existiert eine Vielzahl von Methoden zur Erkennung von Angriffen. Eine der einfachsten Möglichkeiten besteht darin, die Datenpakete nach dem Vorkommen bestimmter Bitsequenzen, die eventuell auf das Vorhandensein von *malicious* Code hinweisen können, zu überprüfen. Eine Sammlung solcher Sequenzen wird auch als *Signaturdatenbank* bezeichnet. Signaturbasierte Angriffserkennung erkennt die Angriffe zwar sehr zuverlässig,funktioniert jedoch in vielen Fällen nicht, da viele Vertreter von *Malware* in der Lage sind, ihren eigenen Code selbständig zu reorganisieren, ohne dabei die Verbreitungs- und Schädigungseigenschaften zu verlieren. Einen etwas anderen Ansatz verfolgt die heuristische Analyse. Heuristische Scanner halten nach Aktivitäten Ausschau, die für eine Malware typisch sind. Der Verdächtigkeitsgrad solcher Aktionen wird durch einen Satz von Regeln beschrieben. Eine verdächtige Aktivität könnte vom Scanner z.B. gemeldet werden, wenn:

- eine Software ein ungewöhnliches Verhalten an den Tag legt, indem sie beispielsweise versucht, bestimme Schutzmechanismen des Betriebssystems auszuhebeln.
- ein angebliches Textverarbeitungsprogramm innerhalb kürzester Zeit Tausende von Werbemails verschickt.
- ein Benutzer sich in der für ihn ungewohnten Zeit anmeldet und verdächtige Aktionen durchführt[17].
- etc..

[17] Man denke z.B. an einen Bankangestellten, der sich am Wochenende mit seinem Account einloggt und eine hohe Summe auf ein Kaiman-Inseln-Konto überweist.

Das Hauptproblem der heuristischen Angriffserkennung liegt in der Bestimmung der Grenze für die Verdächtigkeit einer Aktivität. Ein System mit einer sehr niedrig gesetzten Grenze würde viele Fehlalarme produzieren. Eine viel zu großzügig gewählte Warnstufe würde dagegen dafür sorgen, dass das System viele Angriffe einfach übersieht. Der Hauptvorteil der heuristischen Methode liegt in der Möglichkeit, absolut neue Angriffe zu entdecken, welche durch die signaturbasierte Prüfung nicht entdeckt werden können. Um die Beschreibung der folgenden Konzepte zu vereinfachen, wird in Zukunft lediglich auf die signaturbasierte Prüfung eingegangen. Andere Methoden der Angriffserkennung lassen sich jedoch ebenfalls problemlos in die im Rahmen dieser Arbeit vorgestellten Konzepte integrieren.

3.2 Integritätskontrolle des Signatursatzes

Bei der signaturbasierten Erkennung von Angriffen ist die Qualität der Erkennung von den verwendeten Signaturen abhängig. Dies bedeutet, dass ein Agent nach dem Empfang von neuen Signatursätzen in der Lage sein muss, sicherzustellen, dass die von ihm empfangenen Daten während des Transfers nicht verändert wurden. Ein Angreifer (Malloy) könnte z.B. während der Datenübertragung die von Alice an Bob gesendeten Signaturen so verändern, dass diese keine Daten zu einem bestimmten Angriff enthalten. Diesen Angriff könnte Malloy anschließend bei Bob anwenden, ohne die Gefahr zu laufen, dass Bob diesen Angriff erkennt. Die gefälschten Signaturen könnten außerdem eine Sicherheitslücke in Bobs Antivirensoftware ausnutzen und zu einer Fehlfunktion führen. Einem solchen „Man in the Middle“-Angriff können Agenten mit Hilfe eines Integrity-Checks widerstehen.

Alice berechnet vor dem Übersenden von Daten an Bob eine Checksumme ihrer Daten und veröffentlicht diese auf einem vertrauenswürdigen System oder übermittelt diese über einen sicheren Kanal an Bob. Nach dem Datenempfang berechnet

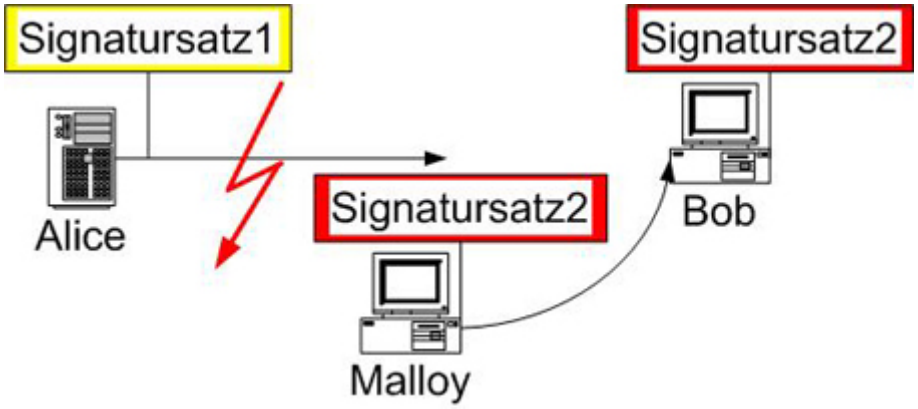

Abbildung 1: Man in the Middle-Agriff. Malloy faelscht Signaturen waehrend der Datenuebertragung.

Bob die Checksumme der von ihm empfangenen Daten und vergleicht sie mit dem von Alice veröffentlichten Wert2w. Sind beide Checksummen nicht identisch, wurden die Daten während der Datenübermittlung verändert.

Eine Checksumme wird in aller Regel nach der *CRC*-Methode berechnet. Auch die kleinste Veränderung der übermittelten Daten bedingt eine völlig neue Prüfsumme. Unsere Agenten verwenden zur Berechnung der Prüfsumme *MD5*. Da das von uns zur Berechnung der Prüfsumme verwendete Verfahren nicht geheim und somit dem potenziellen Angreifer bekannt ist, könnte dieser für seine gefälschten Daten eine neue Prüfsumme berechnen und diese dem Bob übermitteln. Ein solcher Angriff kann z.B. mit Hilfe von digitalen Signaturen vereitelt werden, denn diese beinhalten außer einer Prüfsumme noch zusätzliche Informationen, welche uns die Identität der signierenden Person offenbaren.[18].

[18]Vgl. a. [FUHS 1995].

3.3 Kapazitätsbeschränkungen

In seinem Buch beschreibt Michael Wooldridge einige Trends, welche die Entwicklung moderner Computersysteme kennzeichnen. Er macht den Leser darauf aufmerksam, dass Menschen ihre Aufgaben zunehmend den Computersystemen anvertrauen. Diese Systeme werden ihrerseits zunehmend kleiner und mobiler[19]. Die verfügbaren Rechenkapazitäten der mobilen Geräte sind in der Regel geringer als die der stationären Systeme[20]. Das bedeutet, dass den mobilen Systemen grundsätzlich weniger Rechenkapazitäten zur Verfügung stehen, um ihren eigenen Schutz zu gewährleisten. Die von den Netzbetreibern durchgeführte Überwachung der zentralen Knotenpunkte ist sehr rechenaufwendig und erscheint unter der Annahme, dass viele Systeme zunehmend direkt (ohne die Infrastruktur des Betreibers zu nutzen) miteinander kommunizieren, für wenig aussichtslos. Doch gerade diese mobilen Systeme müssen vor Angriffen geschützt werden, denn sie beinhalten in der Regel sehr sensible Daten. Die Komplexität der Software, welche in den mobilen Geräten ihren Einsatz findet, übersteigt die der einstigen Mainframes. Die Fehlerfreiheit der eingesetzten Software kann deswegen von niemanden garantiert werden. Diesem Umstand muss bei der Konzeption der Angriffserkennugnssoftware für die mobilen Geräte Rechnung getragen werden. Im Folgenden wird gezeigt, wie trotz der relativ geringen Leistungsfähigkeit mobiler Systeme, deren Schutz durch den Einsatz von Agentensystemen verbessert werden kann.

3.4 Der klassische Aufbau einer Sicherheitsinfrastruktur

Vor einiger Zeit war ich für eine Lebensversicherungsgesellschaft tätig. In dieser Branche stellen Informationen, bzw. deren Vertraulichkeit die Geschäftsgrundlage

[19] Vgl. a. [WOOLDRIDGE 2002] S. 1 ff.

[20] Die häufigsten Einschränkungen der Rechenkapazität werden durch die Akkukapazität und die aufwendige Wärmeabfuhr bedingt.

dar. Dementsprechend aufmerksam wird das Thema Sicherheit und Datenschutz behandelt. Während meiner Tätigkeit für das Unternehmen wurde von mir ein Konzept zur Einführung eines Intrusion Detection Systems ausgearbeitet und eine Pilotinstallation durchgeführt. Während der Durchführung von Security Audits lernte ich den Aufbau der Antiviren-Infrastraktur kennen. Der unternhemensweite Einsatz der Antivirensoftware eines führenden Herstellers war für meinen damaligen Arbeitgeber eine sehr kostspielige Angelegenheit. Die aufgebaute Infrastruktur und Konsequenz bei der Durchsetzung ausgearbeiteter Sicherheitsrichtlinien waren beispielhaft. Dies hinderte das eingesetzte System jedoch nicht daran, nicht vernünftig zu funktionieren. Trotz seiner hohen Qualifikation und eines enormen Zeitaufwandes konnte der Systemadministrator es nicht verhindern, dass es immer wieder zu Sicherheitszwischenfällen kam.

In der Abbildung 2 ist der vereinfachte Aufbau des Systems abgebildet. Der Administrationsrechner bezog seine Updates automatisch von der Webseite des Herstellers und versorgte damit die Client-Stationen. Von dieser Konsole konnte der Administrator die Regelwerke der entsprechenden Clients ändern und auch bestimmte Befehle[21] an diese übermitteln.

Der auf den ersten Blick sichere Aufbau hatte jedoch einige Schwachstellen. Die Versorgung der Client-Rechner mit den aktuellsten Signaturen stellte eine der Komplikationen dar. In der Datenbank des Administrationsservers wurden die von ihm zu betreuenden PCs eingetragen. Dieser prüfte in periodischen Zeitabständen, ob seine Clients online sind und aktuelle Signaturen verwenden. Diese Vorgehensweise führte zu erheblichen Problemen zu Beginn des Arbeitstages, da Hunderte von PCs fast gleichzeitig eingeschaltet wurden. Der Update-Server entdeckte sie

[21] z.B. den Befehl zum Neustart des Systems.

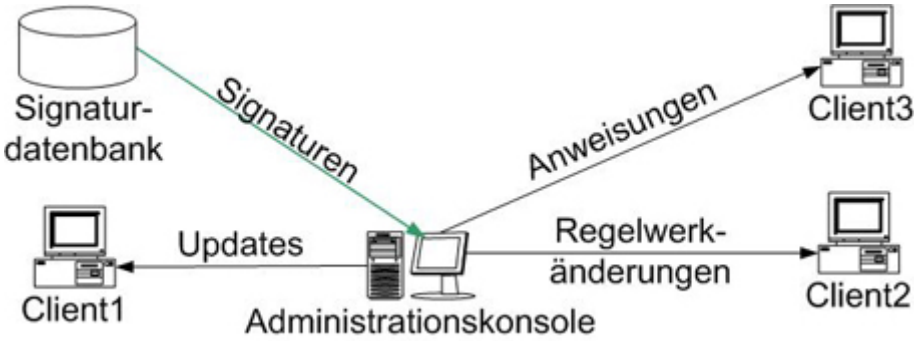

Abbildung 2: Der klassische Aufbau einer Sicherheitsinfrastruktur.

und versuchte, die Systeme mit den aktuellen Signaturen zu versorgen. Wurde während des Update-Vorgangs ein neuer Signatursatz verfügbar, fing der Server mit dem Update der Systeme wieder von vorne an. Die Signatursätze der Systeme zu einem Zeitpunkt selten auf dem gleichen Stand. Bei der für Computerviren üblichen rasanten Verbreitungsgeschwindigkeit könnte dies in einer Katastrophe enden. Diese Vorgehensweise bedingte jedoch ein weiteres Problem. Da neue Signaturen kurz nach dem Systemstart eingespielt wurden und das System einen Sicherheitscheck mit den aktualisierten Signaturen vornahm, war der Rechner in der ersten Viertelstunde nach dem Bootvorgang kaum zu bedienen. Die Unternehmensleitung erkannte schnell das „Problem“ und man entschloss sich dazu, die PCs nachts laufen zu lassen und es somit auf Kosten der Stromrechnung zu lösen. Die „Lösung“ brachte allerdings zwei weitere Probleme mit sich. Obwohl die eingesetzten Büro-PCs für den Dauerbetrieb ausgelegt waren, kam es zunehmend zu den Hardwareausfällen, was zu einer Erhöhung von Reparaturkosten führte[22]. Ein weiterer Nachteil bestand darin, dass die Systeme jetzt nachts unbeaufsichtigt liefen. Ein weiterer Aspekt ist an dieser Stelle ebenfalls zu erwähnen: Antivirenhersteller können auf neue Angriffe lediglich reagieren. Sobald eine neue *Malware* bekannt wird, versuchen sie, deren Signatur zu ermitteln und diese an die Kunden

[22]Ein weiteres Problem könnte z.B. eine erhöhte Brandgefahr sein.

weiterzugeben. Diese Prozedur kann Stunden in Anspruch nehmen, was bedeutet, dass in dieser Zeit die Systeme dem neuen Angriff schutzlos ausgeliefert sind[23]. Ein weiterer großer Nachteil der beschriebenen Infrastruktur bestand darin, dass die Clients nur lokal agiert haben. Um einen akzeptablen Schutz der Systeme zu gewährleisten war man gezwungen, die jeweiligen Scanner-Clients mit dem kompletten Signatursatz und einer eingeschalteten Heuristik laufen zu lassen. Dies führte zu einer schlechten Performance und sorgte für ständige Beschwerden von der Seite der Anwender.

Die beschriebenen Probleme beruhten nicht nur auf der Schwäche der eingesetzten Software und lagen nicht auf der Seite der Administration. Vielmehr waren sie durch den Aufbau des Systems bedingt. Die geschilderten Probleme traten in einem mittelgroßen Netzwerk auf. Bei dem Einsatz in einem größeren Netzwerk wäre eine Eskalation der Situation zu erwarten. Der Einsatz mehrerer Administrationsserver und das dadurch mögliche Load-Balancing würde die Problematik zwar entschärfen jedoch nicht lösen. Einer der größten Nachteile der zentralen Lösung ist die Tatsache, dass zentrale Systeme kritisch für die Funktionsfähigkeit der restlichen Systeme sind. So stellt z.B. der Administrationsserver einen Single Point of Failure dar. Beim Ausfall des Servers können die restlichen Systeme nicht mit den aktualisierten Signaturen versorgt werden. Dasselbe gilt auch für die Systeme des Antiviren-Herstelers. Sobald sie (z.B. aufgrund eines *DoS*-Angriffs) nicht erreichbar sind, können die Administratoren ihre Signaturen nicht mehr aktualisieren.

Diese und viele andere Probleme Sind mit dem Einsatz der zentralisierten Ar-

[23]Der Autor ist sich dessen bewußt, dass das signaturbasierte Scannen nur eine der Angriffserkennungsmethoden ist und dass die Behauptung, die Systeme seien in der Zeit zwischen dem Erscheinen einer neuen Malware und dem Aufspielen der entsprechenden Signatur ohne Schutz, leicht übertrieben ist.

chitektur verbunden. Im folgenden Kapitel werden einige, auf dem Agentenasatz basierende, Konzepte präsentiert, die viele der beschriebenen Schwächen nicht besitzen und trotzdem in der Lage sind, einen angemessenen Systemschutz zu gewährleisten.

3.5 Zusammenfassung

In diesem Kapitel wurden Grundlagen der Angriffserkennung vorgestellt. Der Leser wurde außerdem mit einigen damit verbundenen Problematiken konfrontiert. Die Notwendigkeit der Authenzität des bei der Angriffserkennung verwendeten Signatursatzes und die Flaschenhals-Eigenschaften einiger Infrastrukturbestandteile wurden anhand von Beispielen erläutert. Anschließend wurde eine vereinfachte Darstellung der klassischen Sicherheitsinfrastruktur besprochen und die damit verbundenen Schwierigkeiten erläutert.

4 Agentenbasierter Schutz

4.1 Ressourceneinsparungen durch Alarmstufendifferenzierung.

Im vorhergehenden Kapitel wurde der klassische Aufbau einer Antivireninfrastruktur und dessen Nachteile skizziert. Es wurde gezeigt, dass Systeme häufig unnötig viel Ressourcen zur Erkennung von Angriffen verbrauchen. Eine bloße Betrachtung der Naturvorgänge kann uns Lösungsideen zu diesem Problem liefern. Im Stresszustand kann ein Mensch unglaubliche Taten vollbringen. Gleichzeitig sind wir nicht in der Lage, uns in einem solchen Zustand über einen längeren Zeitraum aufzuhalten, denn dies bedingt einen überproportional hohen Ressourcenverbrauch. Ein ähnliches Prinzip könnte man bei der Erkennung von Angriffen nutzen. Ein Scanner kann permanent mit dem kompletten Satz von Signaturen und einer ressourcenhungrigen Heuristik arbeiten. Dieser „Dauerstresszustand“ würde das System einigermaßen zuverlässig schützen. Nicht jedes Gerät verfügt jedoch über die Ressourcen[24], um sich in dem „Stresszustand“ eine längere Zeit zu befinden. Auf der Hand liegt es deswegen, das System nur dann stark auszulasten, wenn dies wirklich notwendig ist. Im Folgenden wird dieses Prinzip anhand eines Models demonstriert.

Für ein System sind zehn Arten von Angriffen und deren Signaturen bekannt. Um ein ankommendes Datenpaket mit einer Signatur zu überprüfen braucht ein System eine Ressourceneinheit. Jedes dieser Systeme besitzt zehn Ressourceneinheiten und ist somit zwar in der Lage, alle zehn Signaturen im Speicher zu halten und die ankommenden Datenpakete damit zu überprüfen, hat allerdings in diesem Fall keine Kapazitäten mehr, um den anderen Aufgaben nachzugehen. Die natürliche Lösung besteht darin, das System nur mit einem Teil der bekannten

[24] Man denke beispielsweise an die mobilen Geräte, bei denen die Energieversorgung strikte Limits für die Rechenleitung der Geräte setzt.

Signaturen zu betreiben, so dass die Angriffserkennung nicht die kompletten Systemressourcen für sich beansprucht. In unserem Beispiel verwendet das System im Default-Zustand („green“) für die Signaturen drei Ressourceneinheiten. Sobald mehrere Warnungen von Seiten anderer Rechner bei diesem System ankommen, entscheidet sich es, in den Zustand „yellow“ zu wechseln. Jetzt arbeitet dieses System mit einem erweiterten Signatursatz und hat zwar weniger freie Ressourcen zur Verfügung, kann dafür jedoch auch wesentlich mehr Angriffe erkennen. Die Alarmsstufe „red“ bringt das System dazu, alle ihm bekannten Signaturen in den Speicher zu laden, so dass die für das System maximale Sicherheitsstufe erreicht worden ist. Abbildung 3 veranschaulicht dieses Prinzip.

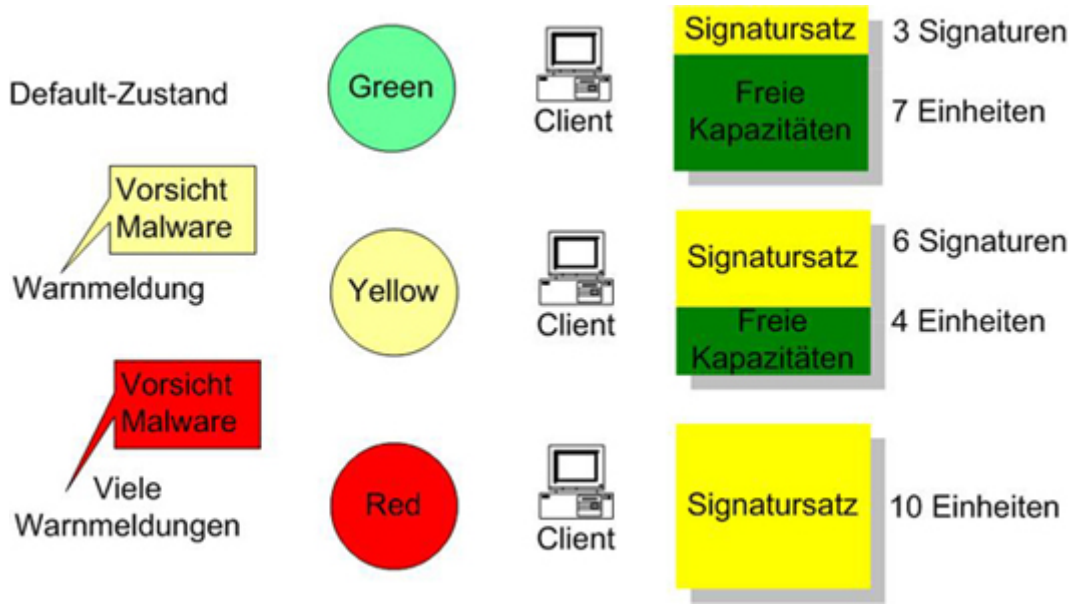

Abbildung 3: Zusammenhang zwischen der aktuellen Alarmstufe und dem verwendeten Signatursatz.

Das Senden von *Malware*-Warnungen an sämtliche Teilnehmer in einem Netzsegment (Broadcast) wäre keine gute Lösung, denn sie würde zu einer unnötig hohen Datenlast und zu einem potenziellen *DoS* führen, da auch ein Angreifer in

der Lage wäre, die Warnmeldungen zu falsifizieren. Im nächsten Abschnitt setzen wir uns mit diesem und einigen anderen interessanten Problemen auseinander und betrachten die Leichtigkeit, mit der sie durch den Einsatz von Agentensystemen gelöst werden können.

4.2 Agentenbasierte Lösung

Der Einsatz autonomer Agenten erlaubt es uns, auf eine hierarchische Struktur zu verzichten. Die Dezentralisierung ermöglicht eine ausgeglichene Auslastung der Netzwerkkapazitäten; es existiert kein zentraler Administrationsserver mehr, der zu Stoßzeiten zu einem Flaschenhals werden könnte. Durch die Eliminierung zentraler Systeme verschwinden außerdem die bei den Angreifern besonders beliebte Ziele, die bildlich gesehen, Achillesfersen der Infrastrukturen darstellen.

Jeder Rechner wird durch einen oder durch mehrere Agenten geschützt. Die Konfigurationen von Agenten könnten sich durchaus voneinander unterscheiden. So können Agenten, z.B. abhängig von den ihnen zur Verfügung stehenden Ressourcen, mit einer ungleichen Anzahl von Regeln arbeiten. Agenten unterstützen ein gemeinsames *Protokoll*, welches ihnen Nachrichten- und Datenaustausch ermöglicht.

Die Abbildung 4 veranschaulicht den Aufbau einer solchen Struktur.

Die Hosts eins und zwei arbeiten mit vergleichsweise kleinen Regelwerken. Agent drei ist dagegen leistungsfähig genug, um von einem erweiterten Signatursatz und Heuristik Gebrauch zu machen. Auf den ersten Blick scheinen die ersten beiden Hosts, einen geringeren Schutz als der dritte Rechner zu besitzen. Global gesehen ist dies jedoch nicht der Fall, denn Host3 verfügt über die notwendigen Voraus-

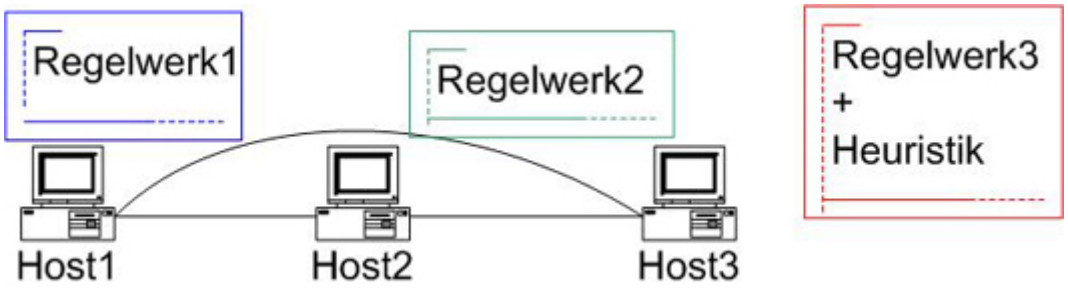

Abbildung 4: Verwendung unterschiedlicher Regelwerke.

setzungen, um alle ihm bekannten Angriffe zu erkennen[25]. Sobald dieser einen solchen Angriff feststellt, übermittelt er eine entsprechende Warnung an die beiden Client-Rechner mit dem Hinweis auf die Signatur des von ihm erkannten Angriffs. Die beiden Rechner müssen jetzt lediglich die entsprechende Signatur in ihre Signatursätze aufnehmen, um von diesem Angriff ebenfalls geschützt zu sein. Die beschriebenen Prozesse werden in der Abbildung 5 veranschaulicht.

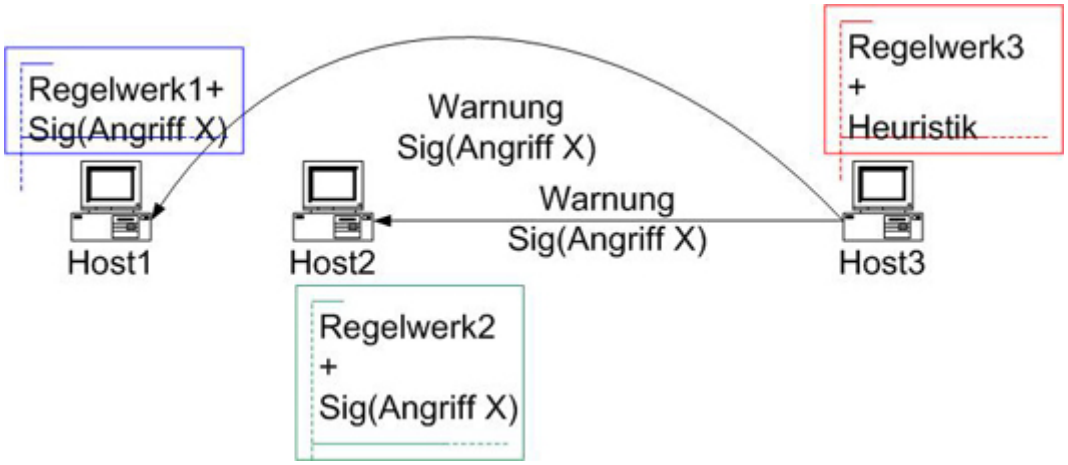

Abbildung 5: Bekanntgabe der Angriffssignatur.

[25]Host3 ist somit in der Lage, Angriffe zu erkennen, die nicht von Host1 und Host2 erkannt werden könnten.

Eine solche Vorgehensweise ermöglicht einen besonders sparsamen Umgang mit den Ressourcen der zu schützenden Rechner. Dies ist beispielsweise in den mobilen Systemen von einer besonderen Bedeutung, denn deren Rechenleistung dieser Systeme ist durch die Akkukapazität und Wärmeabfuhr beschränkt. Die Aufnahme einer neuen Signatur in den Datensatz kann mit einer Schutzimpfung verglichen werden. Die wenigsten Menschen kommen auf die Idee, sich im Hochsommer einer Grippeimpfung zu unterziehen, denn sie ist in der Regel vollkommen überflüssig und würde lediglich das Immunsystem belasten. Auf der anderen Seite würde sich jeder vernünftige Mensch eine Schutzimpfung gegen Malaria gönnen, wenn er im Sommer die tropischen Wälder als Urlaubsziel aufsuchen würde. Es macht Sinn, die Signatur eines neuen Angriffes erst dann in den Signatursatz aufzunehmen, wenn dieser wahrscheinlich wird. Wenn die Gefahr des Angriffes nicht mehr gegeben ist und die Agenten eine bestimmte Sorte von Angriffen nicht zu erwarten haben, können die entsprechenden Signaturen als nicht mehr aktuell eingestuft und wieder aus dem Speicher entfernt werden[26].

Die dezentralisierte Struktur eines Multiagentensystems hat auch weitere positive Eigenschaften. Wir haben bei der Auseinandersetzung mit einem auf dem Client/Server-Ansatz basierenden System gesehen, dass der zentrale Server zum Flaschenhals der Infrastruktur werden kann. Dies ist z.B. dann der Fall, wenn zu viele Clients mit den aktuellen Signaturen versorgt werden müssen und der Server nicht über die dafür notwendigen Kapazitäten verfügt. Die dezentralisierte Struktur eines Multiagentensystems kann dazu verwendet werden, die Kapazitätenengpässe zu vermeiden.

[26] Die *„In-The-Wild“*-Listen und Gefährlichkeitseinstufungen unterschiedlicher *Malware*, welche die Hersteller von Antivirensoftware regelmäßig veröffentlichen, können z.B. dafür genutzt werden, um das Verhältnis zwischen der Größe des verwendeten Signatursatzes und dem Schutz des Systems zu optimieren.

Wir kehren zu dem von uns behandelten Gedankenmodell zurück, bei dem Host3 einen Angriff feststellen konnte und jetzt versucht, die ihm bekannten Hosts über diese Gefahr zu benachrichtigen. Seine Ressourcen reichen jedoch lediglich aus, die Signatur des Angriffs dem Host2 zu übermitteln[27]. Nichtsdestotrotz kann Host1 die von ihm benötigte Signatur in seine Datenbank aufnehmen, indem er diese vom Host2 bezieht. Abbildung 6 veranschaulicht den beschriebenen Vorgang. Host2 wäre außerdem in der Lage, einen Teil der Serveraufgaben zu übernehmen und die ihm bekannten Hosts über die Existenz einer Bedrohung und das Vorhandensein einer Signatur des Angriffs benachrichtigen. Dieser Prozess könnte so lange fortgesetzt werden, bis alle Rechner im bedrohten Netzwerksegment über einen aktualisierten Signatursatz verfügen.

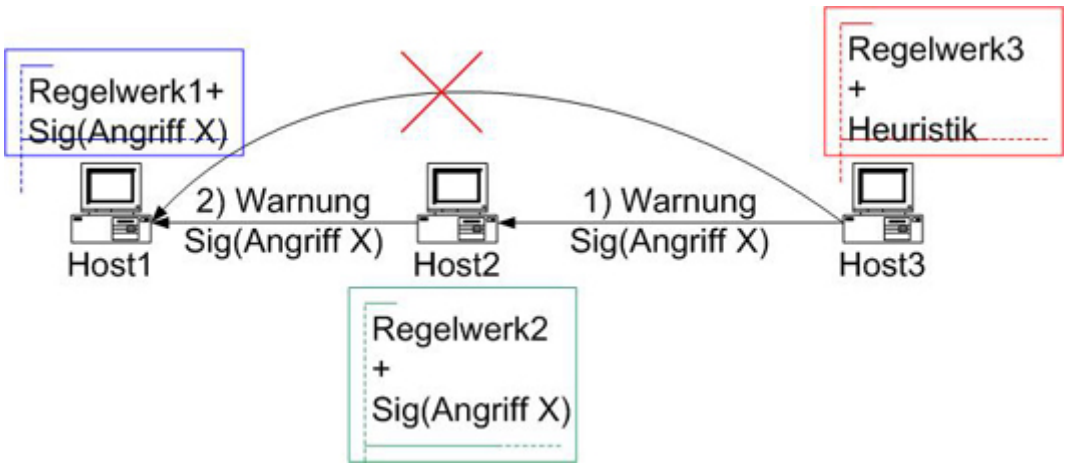

Abbildung 6: Bei der Nichtverfuegbarkeit von Host3 kann Host1 die Signatur des Angriffs vom Host2 beziehen.

Es ist leicht zu sehen, dass der beschriebene Aufbau leicht von einem Angreifer ausgenutzt werden kann, indem er einige Falschmeldungen generiert und diese ver-

[27]So könnte Client3 z.B. aufgrund eines Netzwerkfehlers nicht mehr verfügbar sein.

schickt. Agenten würden diese an die anderen Hosts weiterleiten, was dazu führt, dass das gesamte Multiagentensystem sich nach einer kurzen Zeit im „Stresszustand“ befindet. Sämtliche Systeme arbeiten in diesem Fall mit dem vollen Signatursatz und gehen somit recht verschwenderisch mit ihren Ressourcen um. Das beschriebene Problem kann gelöst werden, wenn Agenten in er Lage sind, ihr Vertrauen zu einem System zu ändern. So können beispielsweise einige falsche Meldungen eines Systems dazu führen, dass dessen Warnungen in Zukunft ignoriert werden. Um das „Hochschaukeln“ der Warnstufe des gesamten Systems zu vermeiden, kann die Tatsache ausgenutzt werden, dass *Malware* sich in der Regel nicht wahllos verbreitet[28]. Die Wahrscheinlichkeit, dass die Malware versucht, zuerst die Nachbarn eines Systems anzugreifen und nicht einen Rechner am anderen Ende der Welt ist relativ hoch. Die Warnmeldungen sollen deswegen mit einem Zähler ausgestattet werden: bei jeder Weiterleitung der Meldung (einem Hop) wird der Zähler decrementiert. Sobald der Zählerstand null beträgt, wird die Meldung vom System nicht mehr weitergeleitet.

4.3 Vertrauensnetz

Das Vorhandensein einer vertrauenswürdigen Stelle, auf der die Prüfsummen der Signaturen gespeichert sind, erleichtert die Konzeption des Multiagentensystems beträchtlich. Die zentrale Stelle ist jedoch gleichzeitig ein willkommener Angriffspunkt. Würde es einem Angreifer gelingen, diese zu kompromittieren, wäre er in der Lage, den Schutz aller Systeme, welche dieser Stelle vertrauen, auszuhebeln.

[28] Dies resultiert aus der Tatsache, dass die Verbindung nach außen von Rechnern eines Netzwerks in der Regel über eine zentrale Stelle erfolgt, die widerrum relativ gut geschützt ist. Deswegen breiten sich die Schädlinge in der Regel wesentlich intensiver innerhalb von abgeschlossenen Netzsegmenten aus.

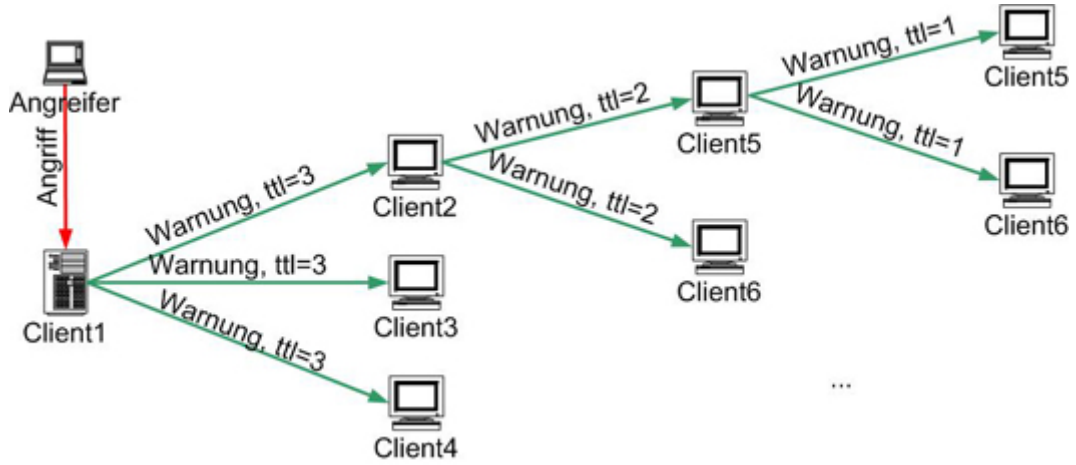

Abbildung 7: Jede Warnmeldung wird mit einem Zaehler versehen, der bei jedem Hop decrementiert wird. Beim Zaehlerstand 0 wird die Nachricht verworfen.

Dabei ist die Existenz einer zentralen Stelle, deren Aufgabe darin besteht, die Authentizität bestimmter Benutzer oder Informationen zu bestätigen (mittels einer *digitalen Signatur*), alles andere als natürlich. Wenn wir auf einer Feier eine Person vorgestellt bekommen, verlangen wir nicht sofort nach ihrem Personalausweis[29]. Es ist vollkommen ausreichend, wenn einige uns bereits bekannte Gäste die Identität dieser Person bestätigen, indem diese uns beispielsweise als ein alter Freund/in vorgestellt wird. Genau dieses Prinzip liegt dem von *PGP* verwendeten Vertrauensnetz (*Web of Trust*) zugrunde und kann bei der Konzeption eines Multiagentensystems verwendet werden. Agent1 kann dem Agenten2 vertrauen, wenn er einem Agenten3 vertraut, der seinerseits dem Agenten2 vertraut. Die entsprechenden Schlüsseln werden von Agenten gegenseitig unterschrieben, so dass sich dadurch die so genannten Vertrauenspfade ergeben. Diese Pfade sollen möglichst kurz und disjunkt sein, da sie Ketten von Bestätigungen darstellen. Je kürzer der Vertrauenspfad zwischen den Schlüsseln zweier Agenten ist, desto vertrauenswürdiger erscheinen

[29] In diesem Fall spielt Personalausweis die Rolle eines Zertifikats, das von einer vertrauenswürdigen Stelle (Meldebehörde) ausgestellt wird und die Authenzität des Inhabers bestätigt.

sich diese. Dasselbe gilt für die Disjunktheit der Vertrauenspfade: je mehr disjunkte Pfade vom Agenten1 zum Agenten2 führen, desto vertrauenswürdiger ist dieser. Abbildung 7 veranschaulicht die sich dadurch ergebenden Vertrauensbeziehung.

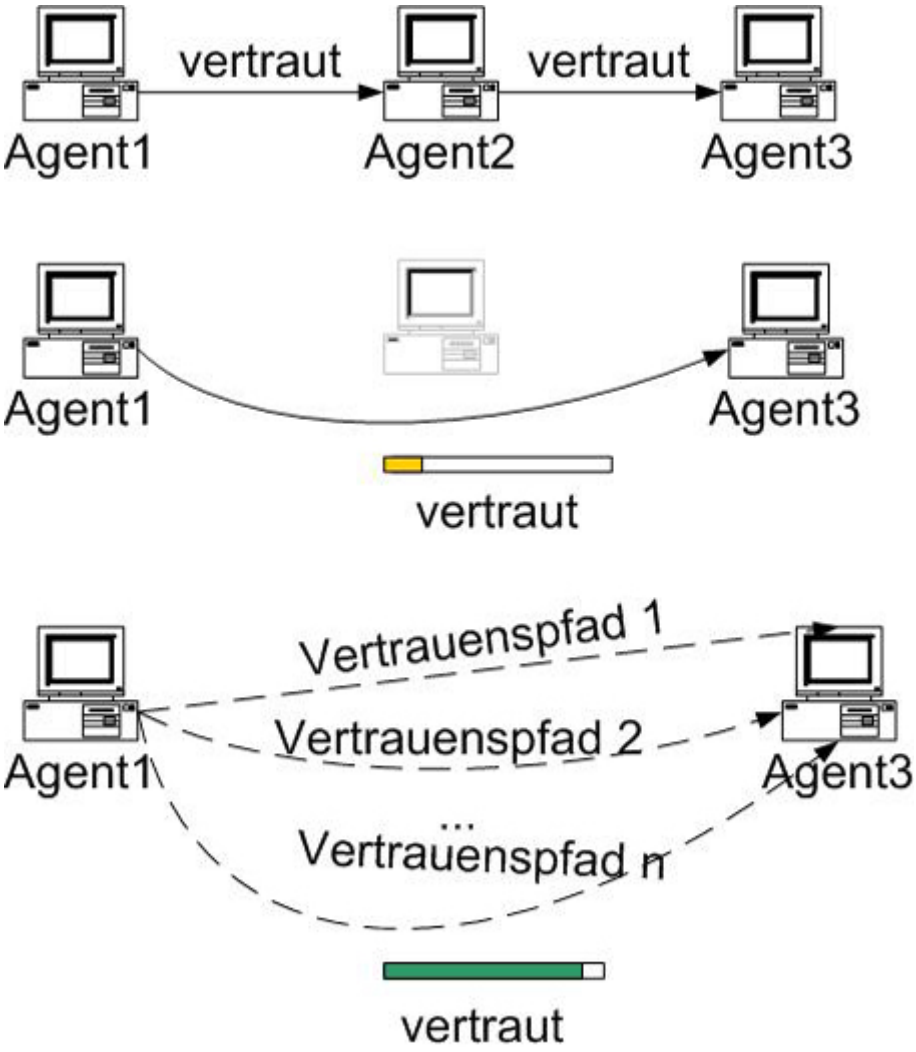

Abbildung 8: Vertrauensbeziehung zwischen Agent1 und Agent2.

Die Einstufung der Vertrauenswürdigkeit eines Agenten kann z.B. nach folgendem Schema erfolgen[30]:

[30]Vgl. a. [FEISTHAMMEL 2002].

Vertrauensstufe	**Bedeutung**
undefiniert	Alle von diesem Agenten ausgehenden Vertrauenspfade werden ignoriert.
teilweise	Mindestens zwei Agenten müssen einen dritten Agenten für vertrauenswürdig erklären, damit es ihm vertraut werden kann.
voll	Mindestens ein Agent muss die Vertrauenswürdigkeit eines dritten Agenten bestätigen.
absolut	Hier sind sämtliche Vertrauenspfade mit der Länge = 1 enthalten.

Tabelle 2: Vertrauensbeziehungen.

4.4 Zusammenfassung

In diesem Kapitel wurden Ansätze besprochen, wie Systemressourcen durch eine Alarmstufendifferenzierung geschont werden können. Es wurden einige Konzepte vorgestellt, welche die Basis für die im Rahmen dieser Arbeit stattfindenden Implementierung bilden. Ein besonders großer Wert wurde auf die Dezentralisierung der vorgestellten Ansätze gelegt. Obwohl die hier vorgestellten Konzepte noch keine Verwendung in der aktuellen Antivirensoftware finden, werden einige von ihnen bereits seit geraumer Zeit in anderen Bereichen der Informationstechnologie eingesetzt, so dass deren Verwendung in einem produktiven Umfeld zwecks Angriffserkennung durchaus denkbar ist.

5 Zusammenfassung

In dieser Seminararbeit wurden unterschiedliche Definitionen von Agenten vorgestellt und deren Eigenschaften aufgezeigt. Einer Beschreibung der Eigenschaften von Agenten(systemen) folgten einige konkrete Anwendungsbeispiele. Die Hauptaufgabe dieser Ausarbeitung lag darin, dem Leser aufzuzeigen, wie der Einsatz von Agenten die vorhandene Sicherheitsarchitektur positiv verändern kann.

Glossar

CRC Cyclic Redundancy Check Prüfsumme, wird in Übertragungsprotokollen und Packern verwendet. Üblich sind 16 oder 32 Bit lange Varianten, kurz: CRC-16 und CRC-32. Eine CRC stellt den Rest aus einer Polynomdivision dar. Implementationen sind allgemein als Quelltext erhältlich.

DES Digital Encryption Standard. Symmetrischer Verschlüsselungsalgorithmus mit einer Schlüssellänge von 56 Bit. Kann nach dem heutigen Stand der Technik relativ leicht geknackt werden.

Digitale Signatur Aus den zu signierenden Daten und dem Geheimschlüssel wird mittels eines Einweg-Hashalgorithmus eine digitale Signatur erzeugt, deren Echtheit man mit dem öffentlichen Schlüssel überprüfen kann. Wird die Datei oder die Signatur verändert, ergibt sich bei der Überprüfung der Signatur eine Fehlermeldung. Mit digitalen Signaturen kann man die Echtheit von digitalen Dokumenten wie beispielsweise Texten, Fotografien, Quellcode bestätigen.

DoS Denial of Service. Hindert einen Anwender an der Nutzung von Diensten. Unbefugte überlasten ein System, damit es den eigentlichen Aufgabe nicht nachkommen kann. Dabei wird zum Beispiel ein Server der mit dem Internet verbunden ist, mit sinnlosen Datenpaketen überflutet.

In-The-Wild Als *'In-The-Wild'* werden die Viren bezeichnet, die in der Öffentlichkeit verbreitet sind. Dafür ist Voraussetzung, daß der Virus in mindestens zwei unterschiedlichen voneinander unabhängigen Regionen aufgetreten ist. s.a. *'In-The-Zoo'*

In-The-Zoo 'In-the-Zoo'-Viren sind Viren die nicht verbreitet sind, sondern nur in Forschungsumgebungen existieren.

Integrity-Checker Ein Programm, welches Veränderungen an Dateien feststellen kann. Diese Veränderungen treten z. B. auf, wenn eine Malware ein Programm infiziert hat. Der Integrity Checker sucht nach solchen Veränderungen und markiert entsprechende Dateien als verdächtig.

Malicious Software siehe *Malware*.

Malware Kunstwort aus *malicious* (englisch für 'boshaft') und Software. Software, die primär schädliche Auswirkungen für den User hat, wie z.B. Viren, Würmer oder Trojanische Pferde.

MD5 Message Digest Version 5 *(MD5)* ist der bekannteste kryptographische Prüfsummenalgorithmus. MD5 weist die wichtige Eigenschaft auf, dass er sich viel effizienter berechnen lässt als bspw. *DES* oder *RSA*.

PGP Pretty Good Privacy. Eine von Philip Zimmermann in den USA entwickelte, weit verbreitete Verschlüsselungssoftware. PGP benutzt den patentierten Algorithmus IDEA und fordert für kommerzielle Anwender den Erwerb einer Lizenz. Der Quellcode von PGP ist öffentlich nicht verfügbar, die Integrität der Software wird von Experten in Frage gestellt.

Protokoll Ein Satz von Regeln und Vereinbarungen, der den Informationsfluss in einem Kommunikationssystem steuert.

RSA Ein Algorithmus zum Signieren und asymetrischen Verschlüsseln von Daten. RSA steht für Rivest, Shamir, und Adelman, die Erfinder des Algorithmus. Dieser Algorithmus ist von Patenten geschützt und daher nicht frei verwendbar.

Web of Trust Netzwerk gegenseitigen Vertrauens). Schlüsselunterschriften werden auch in einem als Web of Trust bekannten Schema benutzt, um die Gültigkeit auch auf Schlüssel auszudehnen, die nicht direkt von Ihnen selbst,

sondern von anderen Personen signiert worden sind. Dabei ist nicht das Vertrauen in die andere Person, sondern das Vertrauen in deren Fähigkeit, Schlüssel sorgfältig zu authentifizieren und richtig zu signieren entscheidend. Verantwortungsbewußte Benutzer, die eine gute Schlüsselverwaltung praktizieren, können das Verfälschen des Schlüssels als einen praktischen Angriff auf sichere Kommunikation mit Hilfe von GnuPG abwehren.

Literatur

[ANONYMOUS 2000] Anonymous. (2000) *Maximum Security (A Hacker's Guide to Protecting Your Internet Site and Network).* USA, Sams.net Publishing, ISBN 1-57521-268-4.

[CHEONG 1996] Cheong, F.C. (1996) *Internet Agents (Spiders, Wanderers, Brokers and Bots).* USA, New Riders, ISBN 1-56205-463-5.

[COULOURIS 2002] Coulouris, G. Dollimore, J. Kindberg, T. (2002) *Verteilte Systeme (Kozepte und Design).* München, Pearson Education Deutschland GmbH, ISBN 3-8273-7022-1.

[FEISTHAMMEL 2002] Feisthammel, P. (Juni 2002) *Das web of trust (Vertrauensnetz).* <http://www.rubin.ch/pgp/weboftrust.de.html> (Click-Datum 16.04.2004).

[FUHS 1995] Fuhs, H. (Mai 1995) *Methoden zur Entdeckung von Computerviren.* <http://www.fuhs.de/de/fachartikel/artikel_de/methviren.shtml> (Click-Datum 09.04.2004).

[NORTHCUTT 2001] Northcutt, S. Novak, J. (2001) *IDS: Intrusion Detection-Systeme (Spurensuche im Internet).* Bonn, mitp-Verlag, ISBN 3-8266-0727-9.

[NORTHCUTT 2003] Northcutt, S. Zelster, L. Winters, S. Frederick, K.K. Ritchey, R.W. (2003) *Inside Network Perimeter Security.* USA, Indiana, Indianapolis, New Riders, ISBN 0-7357-1232-8.

[RUSSEL 2003] Russel, S. Norvig, P. (2003) *Artificial Intelligence (A Modern Approach).* USA, Prentice Hall, ISBN 0-13-080302-2.

[TANENBAUM 2003] Tanenbaum, A. Marten van Steen. (2003) *Verteilte Systeme (Grundlagen und Paradigmen).* München, Pearson Education Deutschland GmbH, ISBN 3-8273-7057-4.

[WILLIAMS 1999] Williams, J. (1999) *Bots and Other Internet Beasties.* USA, Sams.net Publishing, ISBN 1-57521-016-9.

[WOOLDRIDGE 2002] Wooldridge, M. (Februar 2002) *An Introduction to Multi-Agent Systems.* Chichester, England, John Wiley & Sons, ISBN 047149691X.