F A C H H O C H S C H U L E W E D E L
M A S T E R T H E S I S

in der Fachrichtung
Computer Science

Thema:

Agentensysteme und Infrastruktursicherheit

Verwendung von Agenten zum Schutz von Systemen

Eingereicht von:	Michael Grenz Rudolfstr. 13 65197 Wiesbaden E-Mail: <name>@web.de, <name>=michaelgrenz
Erarbeitet im:	8. Semester
Abgegeben am:	29 August 2004
Referent (FH Wedel):	Prof. Dr. Sebastian Iwanowski FH Wedel Feldstrasse 143 22880 Wedel Germany Tel.: +49 (0) 4103 8048 63 Tel.: +49 (0) 4103 8048 91063 E-Mail: <name>@fh-wedel.de, <name>=iw

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ACL	Agent Communication Language
bzw.	beziehungsweise
etc.	et cetera
f.	folgend(e)
ff.	fortfolgende
FIPA	Foundation for Intelligent Physical Agents
ggf.	gegebenenfalls
ID	Intrusion Detection
IP	Internet Protocol
KI	Künstliche Intelligenz
PC	Personal Computer
S.	Seite
s.a.	siehe auch
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 Arbeit besteht darin, den Leser mit dem Begriff des Softwareagenten vertraut zu machen und ihm, nach einer Auseinandersetzung mit dieser Thematik, eine Einsatzmöglichkeit für Agenten aufzuzeigen, nämlich im Bereich des Systemschutzes.

Die vorliegende Arbeit setzt sich aus sieben Kapiteln zusammen. Der Auseinandersetzung mit der Thematik von Agentensystemen und der Erklärung theoretischer Funktionsgrundlagen der Angriffserkennung folgt die Darstellung einer möglichen Vorgehensweise zur Gewährleistung des Systemschutzes mit Hilfe von Agentensystemen. Die im Kapitel „Agentenbasierter Schutz“ vorgestellten Konzepte werden im darauf folgenden Kapitel „Mathematische Grundlagen“ mit Hilfe statistischer Untersuchungen untermauert. Im Abschnitt „Implementierungsdetails“ werden einige Besonderheiten der im Rahmen dieser Arbeit stattgefundenen Implementierung erläutert. Im letzten Kapitel dieser Arbeit werden anschließend deren Ergebnisse und die daraus zu ziehenden Erkenntnisse übersichtlich zusammengefasst.

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* markiert.
- 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 sich manche Definitionen 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 noch etwas anderen Definition des Agentenbegriffs geht Michael Wooldridge aus[3]. Danach ist ein beliebiges Programm genau dann ein Agent, wenn es in der Lage ist, 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 der Lage sein, Perzepte wahr zu nehmen und auf diese reagieren zu können. Ein Agent muss zielgerichtet handeln und mit den anderen Agenten kommunizieren können. Tanenbaum[5] unterscheidet zusätzlich zwischen mehreren Agententypen,

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

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: Hervorhebung bestimmter Agenteneigenschaften.

2.1.1 Abgrenzung zum Client/Server-Modell

Die beschriebenen Eigenschaften eines Agenten verdeutlichen uns den Unterschied zu dem klassischen Client/Server-Modell. 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. Das Serversystem stellt dabei dem Client eine bestimmte Funktionalität zur Verfügung. Die Anfragen des Client-Systems sind dafür maßgebend, welche Aktionen im nächsten Verarbeitungsschritt durchgeführt werden. Wenn eine Folge von mehr oder weniger komplizierten Aktionen vom Client/Server-System durchgeführt werden soll, um ein bestimmtes Ergebnis zu erzielen, muss der Client die Ausführungsreihenfolge dieser Aktionen verwalten, indem er dem Server nach der Beendigung eines Verarbeitungsschrittes eine neue Anfrage sendet. Dies ist mit einem hohen Verwaltungsaufwand von Seiten des Clients verbunden, insbesondere dann, wenn mehrere Aufträge parallel durchgeführt und verwaltet werden müssen. Im Zweifel[6] wird der Client dazu gezwungen, die Aktionsreihenfolge neu zu konstruieren und eine neue Serie von Aufträgen an den Server zu senden. Dieser Aufwand kann durch den Einsatz von Agentensystemen verringert werden. Dank der Fähigkeit

[6] Z.B. wenn der Server eine bestimmte Aktion nicht ausführen kann.

von Agenten, autonom und proaktiv zu Handeln, muss der Auftraggeber den Ausführungsstatus nicht mehr überwachen. Ein Agent kann selbständige und auch langfristig-sinnvolle Entscheidungen treffen und ist somit in der Lage, beim Fehlschlagen bestimmter Aktionen[7], eine Entscheidung über die Aktionsalternativen zu treffen und diese auch umzusetzen. Beim Client/Server-System ist dies nicht möglich.

2.1.2 Agenten und Objektorientierter Ansatz

Sowohl Agenten als auch Objekte gehören zu den Programmierparadigmen. Viele Merkmale sind diesen beiden Ansätzen gemeinsam. Agenten und Objekte können als Exemplare aufgefasst werden, die bestimmte Eigenschaften besitzen und in der Lage sind, Aktionen durchzuführen[8]. Agenten unterscheiden sich von Objekten jedoch in dem Grad ihrer Autonomie. Sowohl Agenten, als auch Objekte haben Kontrolle über ihre internen Zustände[9]. Gleichzeitig üben Objekte keine Kontrolle über ihr Verhalten aus. Denn sobald ein Objekt eine seiner Methoden freigibt[10], verliert es die Kontrolle darüber, wann diese Methode ausgeführt wird. Um ein zusammenhängendes System aus mehreren Objekten zu erstellen, müssen Objekt-Methoden freigegeben werden. Das ist relativ unproblematisch, solange alle Bestandteile des Systems dasselbe Ziel verfolgen. Bei einem Agentensystem wird diese Eigenschaft jedoch nicht vorausgesetzt, denn ein Agent vertritt die Interessen seines Besitzers. In einem Multiagentensystem können Agenten mehreren unterschiedlichen Besitzern gehören, die ihrerseits unterschiedliche Ziele verfolgen können. Aus diesem Grund wird ein Agent **A** nicht immer eine Aktion **a** ausführen, nur weil ein anderer Agent **B** das möchte[11]. Bei einem Agenten spricht man deswegen nicht vom Aufruf einer bestimmtem Aktion[12], sondern von der Anfrage, diese Aktion auszuführen. Die endgültige Entscheidung, ob die Ak-

[7] z.B. bei der Änderung seiner Umgebung

[8] Vgl. a. [OA 1996], [WOOLDRIDGE 2002] S. 25

[9] Interne Zustände von Objekten werden in der Variablenwerten des Objektes festgehalten.

[10] Dies kann z.B. durch die `public`-Deklaration geschehen.

[11] Vgl. a. [WOOLDRIDGE 2002] S. 26

[12] Dies ist bei einem Objekt der Fall.

tion ausgeführt wird oder nicht, liegt beim Agenten selbst. Bei einem Objekt ist dagegen die Entscheidung des Aufrufers maßgebend[13]. Die Autonomie von Agenten ist flexibel, denn sie basiert auf dem reaktiven, proaktiven und sozialen Verhalten. Es ist möglich, Objekte zu bauen, die ein ähnlich flexibles autonomes Verhalten an den Tag legen; dies ist jedoch ebenfalls kein Bestandteil des klassischen Objektorientierten Ansatzes[14].

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 dem 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 zusammengefasst, wenn die einzelnen Agenten in der Lage sind, miteinander zu kommunizie-

[13] Natürlich kann einem Objekt die Autonomie-Eigenschaft beigebracht werden, indem es programmiert wird, vor der Ausführung einer Aktion, Prüfungen durchzuführen, die ihm für die Entscheidung zuständig sind, ob eine bestimmte Aktion ausgeführt werden soll oder nicht. Ein solches Verhalten ist allerdings nicht der Bestandteil des ursprünglichen Objektorientierten Modells. Vgl. a. [WOOLDRIDGE 2002] S. 26

[14] In seinem Buch verweist Michael Wooldridge auf ein weiteres Unterscheidungsmerkmal zwischen Objekten und Agenten - die Kontrollfunktion. Tatsächlich ist es so, dass ursprünglich mehrere Objekte in einem System eine gemeinsame Kontrollkomponente besaßen. Inzwischen ist eine „Multi-Threaded“ Umgebung für Objekte ein fester Bestandteil moderner Objektorientierter Sprachen und ist als Konzept aus der Objektorientierten Programmierung nicht mehr wegzudenken (z.B. in Java). Aus diesem Grund ist die Validität dieses Unterscheidungsmerkmals mehr als fragwürdig. Vgl. a. [WOOLDRIDGE 2002] S. 26

ren, was z.B. mit Hilfe des Nachrichtenaustauschs geschehen kann. Der einzelne Agent agiert, um die Interessen seines Auftraggebers zu vertreten[15]. Damit er dies auch in einer Multiagentenumgebung tun kann, müssen die einzelnen Agenten in der Lage sein, miteinander zu kooperieren, die eigenen Handlungen zu koordinieren und gegebenenfalls Verhandlungen zu führen.

Ein aus mehreren Agenten bestehendes System (Multiagentensystem) kann als klassisches verteiltes System verstanden werden. Bei diesem verteilten System stellen mehrere verteilte Objekte diverse Dienste bereit und interagieren miteinander über eine Kommunikationskomponente. Ein verteiltes System wird oft als „Zusammenschluss unabhängiger Computer, welcher sich für den Benutzer als ein einzelnes System präsentiert“, definiert. Dabei unterscheidet man in der Regel zwischen dem Client/Server-System, der verteilten Anwendung und dem verteilten Betriebssystem. Idealerweise erscheint dieses, aus mehreren Komponenten bestehende System dem Benutzer gegenüber als eine Einheit.

Da ein Multiagentensystem eine besondere Art eines verteilten Systems darstellt, erfüllt es ebenfalls alle Anforderungen, die für ein verteiltes System gelten:

- **Ortstransparenz:** Dienste und Ressourcen können auf mehrere Lokationen verteilt werden und müssen dem Auftraggeber des Agenten nicht bekannt sein. Dieser erteilt dem Agenten einen Auftrag, welcher vom Agenten unter Benutzung der ihm zur Verfügung stehenden verteilten Ressourcen erfüllt werden muss, und sieht folglich nicht die Lokationen der entsprechenden Ressourcen.

- **Zugriffstransparenz:** Die Zugriffstransparenz für den Benutzer resultiert aus der bereits erläuterten Ortstransparenz. Die Zugriffstransparenz eines einzelnen Agenten geht (im idealen Fall) dagegen aus der einheitlichen Agentenplattform hervor.

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

- **Nebenläufigkeitstransparenz:** Da ein Multiagentensystem per Definition aus mehreren Agenten besteht, die mehrere Besitzer haben können, ist es absolut natürlich, dass in einem Multiagentensystem mehrere Agenten parallel auf Dienste und Ressourcen zugreifen können.

- **Skalierungstransparenz:** Das Hinzufügen/Entfernen zusätzlicher Agenten bzw. Ressourcen geschieht transparent für den Endbenutzer.

- **Migrationstransparenz:** Die Migration von Diensten und Ressourcen bleibt vom Benutzer unbemerkt, solange der entsprechende Agent noch in der Lage ist, mit den ihm zur Verfügung stehenden Ressourcen seinen Auftrag zu erfüllen.

- **Leistungstransparenz:** Die Verteilung von Ressourcen auf einzelne Agenten geschieht transparent und ist von den Einstellungen einzelner Agenten abhängig.

- **Replikationstransparenz:** Aus Performance-Gründen können mehrere Agenten über gleiche oder ähnliche Ressourcen/Eigenschaften verfügen. Dadurch steigert sich nicht nur die Leistungsfähigkeit des Multiagentensystems, sondern auch dessen Fehler- und Ausfalltoleranz.

- **Fehler- und Ausfalltransparenz:** Beim Ausfall eines oder mehrerer Agenten kann das Multiagentensystem in der Regel seine Dienste weiterhin fortsetzen[16].

Darüber hinaus besitzt ein Multiagentensystem weitere Eigenschaften, die für ein verteiltes System nicht üblich sind. So wird ein verteiltes System von einer zentralen Stelle gesteuert, welche die soeben genannten Eigenschaften gewährleisten soll. Ein Multiagentensystem ist auf diese Einschränkung nicht angewiesen und erfordert folglich einen geringeren Pflegeaufwand. Ein Multiagentensystem besitzt außerdem eine „Ad Hoc“-Eigenschaft; es kann plötzlich entstehen und genauso schnell wieder aufgelöst werden, sobald bestimmte Ziele erreicht worden

[16] Vgl. a. [WIKIPEDIA 2004]

sind. Das Entstehen eines Mulitagentensystems erfordert lediglich ein Kommunikationsprotokoll, welches von seinen Bestandteilen (Agenten) unterstützt werden muss.

Im Gegensatz zu einem verteilten System vertreten die Agenten Interessen unterschiedlicher Auftraggeber. Diese Interessen sind nicht nur für die vom Agentennetz zu erfüllenden Ziele, sondern auch für die dem Agentensystem zur Verfügung stehenden Ressourcen maßgebend. Aus diesem Grund sind bei der Entwicklung eines Agentensystems nicht nur der interne Aufbau seiner Einheiten (Agent Design) sondern auch deren „soziale Fähigkeiten“ (Society Design) von Bedeutung[17].

2.3 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[18]. 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

[17] Vgl. a. [WOOLDRIDGE 2002] S. 3 ff.
[18] 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 ist dagegen wesentlich komplizierter oder kann gar nicht erst erfolgen.

Nach dem oben aufgeführten 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 bei Kapazitätsengpässen konsequent

bevorzugen und ihnen die Leitungskapazität zuweisen, könnte seine Entscheidung, kurzfristig gesehen, absolut richtig sein[19]. Auf längere Sicht könnte es allerdings bedeuten, dass manche Clients gar keine Ressourcen zugewiesen bekämen, was nicht dem Designziel des Agenten entsprechen würde.

2.4 Anwendungsgebiete für Agentensysteme

Agentensysteme können aufgrund ihrer Eigenschaften auf vielfältige Weise eingesetzt werden. Besonders nützlich sind Agentensysteme, wenn in einem komplexen verteilten System bestimmte Dienstleistungen in selbständigen Einheiten verfügbar gemacht werden sollen. Die zur Verfügung gestellten Dienste können relativ einfach sein (z.B. Datenbankabfragen), jedoch auch eine komplexe Natur besitzen[20]. Auf der 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	Flugverkehrsüberwachung, Business Process Management, verteiltes Sensoring, Produktionsüberwachung
Netzwerke	Informationssammlung und -management.
Human-Computer Interfaces	Nachrichtenassistenten, Spam-Filtering.

Tabelle 2: Die häufigsten Einsatzgebiete für Agentensysteme.

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

2.5 Zusammenfassung

In diesem Kapitel wurde der Begriff eines Agenten diskutiert und die Unterschiede des agentenbasiereten Ansatzes zu den anderen, dem Leser bereits bekannten, Konzepten verdeutlicht. Der Beschreibung von Agentenumgebungen folgte eine kurze Auflistung der möglichen Anwendungsbereiche für ein Agentensystem. Im nächsten Kapitel beschäftigen wir uns ausführlich mit einem einzelnen Einsatzgebiet für Agentensysteme - der Erkennung von Angriffen auf Computersysteme.

3 Angriffserkennung

3.1 Notwendigkeit des Systemschutzes

An dieser Stelle erwartet der Leser einen für die Arbeit mit einer solchen Thematik üblichen „Einschüchterungsabschnitt“, in dem sämtliche, einen Anwender bedrohenden Gefahren ausnahmslos aufgezählt werden und der Autor anschließend zur Schlussfolgerung kommt, dass jeder Anwender sein System durch den Einsatz einer Fülle diverser Sicherheitsprodukte um jeden Preis sicherer machen muss[21]. Üblicherweise wird in solcher Literatur darauf hingewiesen, dass nur das einwandfreie Zusammenspiel diverser Sicherheitssoftware zur wirklichen Sicherheit beiträgt und gleichzeitig eine bestimmte Produktart oder gar ein Programmpaket als besonders zuverlässig und benutzerfreundlich angepriesen, ohne gleichzeitig die Schutzbedürfnisse der angesprochenen Anwender zu analysieren. Diese Arbeit soll jedoch nicht dem Weg folgen, welchen bereits viele andere Autoren gegangen sind und zu einer Verkaufsveranstaltung ausarten[22]. Viel mehr soll an dieser Stelle eine kurze nüchterne Auseinandersetzung mit den Grundlagen der Systemsicherheit erfolgen, um eine Basis für das Verständnis weiterer Ausführungen zum Hauptthema dieser Arbeit zu schaffen.

3.2 Arten von Angriffen

Bereits vor langer Zeit hat man Versuche eingestellt, Systeme per Hand zu kompromittieren. Diese Vorgehensweise ist sehr zeitintensiv und ergibt nur in seltensten Fällen eine lohnenswerte Ausbeute. Man ist viel mehr dazu übergegangen, sich bestimmter automatisierter Werkzeuge zu bedienen, denn diese erlauben es, eine Vielzahl von Systemen in einer kurzen Zeit und mit einer kalkulierbaren Ge-

[21] Üblich dabei ist die Empfehlung von Antivirenprodukten, *Firewalls*, Integritätscheckern, Content-Blockern, Antispyware-Software und diverser anderer Vertreter der Software dieser Art, wobei in seltensten Fällen auf das Zusammenspiel dieser Komponenten geachtet wird.

[22] Das Interesse der Öffentlichkeit am Thema „Sicherheit und Datenschutz“ ist in der letzten Zeit sehr stark geworden. Es wäre an dieser Stelle viel zu einfach, die grundlegenden Angriffstechnicken mit den Sicherheitsvorfällen der letzten Zeit zu vermischen und somit dem Leser eine spannende, jedoch gleichzeitig geistig wenig anspruchsvolle Geschichte anzubieten.

fahr für den Angreifer, erkannt zu werden, anzugreifen. Diese Werkzeuge weisen bestimmte Funktionalität auf; eine Möglichkeit, sie einzuordnen, kann der Abbildung 1 entnommen werden[23]. Dabei soll diese nicht als eine endgültig festgelegte Taxonomie verstanden werden, denn das Umfeld der beschriebenen Programme ist sehr heterogen und kann nur sehr schwer eindeutig klassifiziert werden[24].

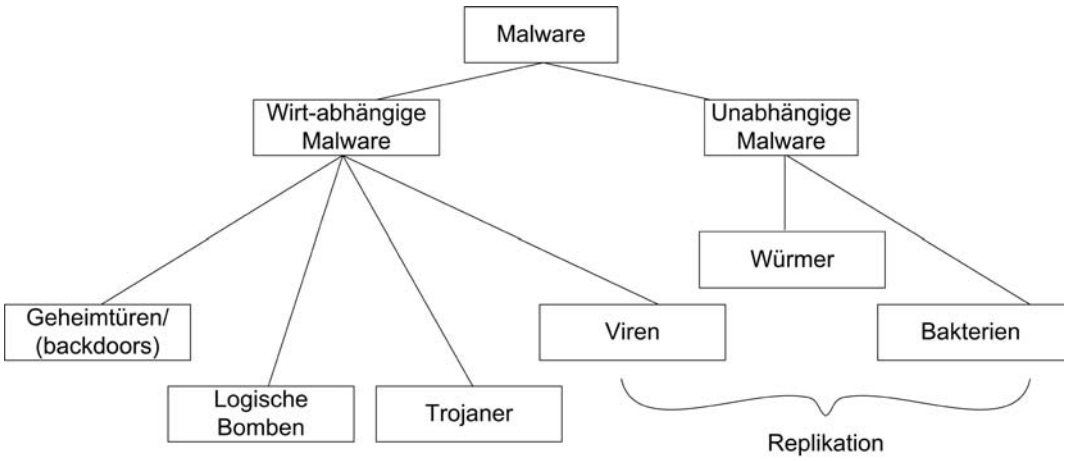

Abbildung 1: Taxonomie der *Malware*.

Die Unterscheidung zwischen den Systemen in der dargestellten Taxonomie basiert darauf, ob ein Schädling einen Wirt braucht[25] oder nicht. Geheimtüren (*Backdoors*), logische Bomben und *Trojanische Pferde* bedürfen eines solchen Wirts, wobei *Viren*, *Bakterien* und *Würmer* in der Lage sind, sich selbständig zu verbreiten.

- Unter *Backdoors* versteht man die Möglichkeit für einen Hacker, sich einen unerlaubten Zugang zu einem System zu verschaffen, auch wenn alle Sicherheitslücken geschlossen sind.

- *Trojanische Pferde* sind selbstständige Programme mit einer verdeckten Schadensfunktion. Von einem klassischen *Computervirus* unterscheiden sich

[23] In der Fachwelt existiert derzeit keine Einigkeit über die Taxonomie der *Malware*. Im Rahmen dieser Arbeit wird von der in der Abbildung 1 dargestellten Einteilung ausgegangen. Vgl. a. [NEUMANN 1995] S. 98 f., [KLANDER 1997] S. 391 f.

[24] So besitzen die aktuellen *Viren* und *Würmer* in der Regel Funktionalität von *Trojanischen Pferden* und *logischen Bomben*. Vgl. a. [SCHMIDT 2004]

[25] In der Regel - eine Datei.

die „*Trojanische Pferde*“ dadurch, dass sie sich nicht selbstständig verbreiten, sondern an ihr „Wirtsprogramm“ gebunden sind. Sie nisten sich oft im Betriebssystem ein, sind also für durchschnittliche PC-Nutzer kaum zu erkennen.

- *Computerviren* sind Programme, die sich selbst mit oder ohne Zutun des Anwenders vervielfältigen können, wobei die neuen Ableger sich ihrerseits weiter vermehren.

- *Bakterien* werden häufig mit den *Viren* verwechselt. Der Hauptunterschied zwischen Bakterien und Computerviren besteht darin, dass *Bakterien* keine Schadensroutine beinhalten. Sie reproduzieren sich jedoch wiederholt, was zu einem Denial of Service (*DoS*) führen kann.

- *Würmer* sind eine *Malware*-Art, die in der letzten Zeit besonders viele Schlagzeilen macht[26]. *Würmer* sind in der Lage, sich selbst zu reproduzieren und selbständig zu verbreiten, indem sie z.B. Kopien von sich selbst per eMail an diverse Empfänger versenden.

Jede dieser Schädlingsarten kann in mehrere Unterarten eingeteilt werden. So unterscheidet man z.B. zwischen Speicher-residenten *Viren*, polymorphen *Viren*, Bootsektor-*Viren* etc.. Mehr Informationen zu diesem Thema können der weiterführenden Literatur entnommen werden[27].

Die aufgeführten Schädlinge führen bestimmte Angriffe gegen die Systeme aus. Diese Angriffe können von vielfältiger Natur sein, denn sie basieren auf den Eigenschaften einzelner Schädlinge. Grundsätzlich kann dabei zwischen zwei Arten von Angriffen unterschieden werden.

3.2.1 Passive Angriffe

Ein passiver Angriff beinhaltet das Abhören von Daten, die über einen Kommunikationskanal versendet werden. Manche passive Angriffe verwenden jedoch

[26] Vgl. a. [SCHMIDT 2004]
[27] Vgl. a. [STALLINGS 2000] S. 304 ff.

Techniken, die wesentlich raffinierter sind als das triviale Abhören einer Leitung. So kann beispielsweise elektromagnetische Abstrahlung von Bildschirmen und Computerboards mit Hilfe spezieller Werkzeuge gemessen und analysiert werden. Im Rahmen des passiven Angriffs werden keine Daten verändert, wobei man hier wiederum zwischen drei grundlegenden Angriffsmöglichkeiten unterscheidet:

- **Das Abhören von Daten.** Dies ist wohl die offensichtlichste Form eines passiven Angriffes, in der das Ausspähen von Informationen im Vordergrund steht, die entweder unzureichend verschlüsselt oder gar in Klartext übermittelt werden. Besonders anfällig zu dieser Form des Angriffs sind z.B. veraltete *Protokolle*, wie *FTP*, *TELNET*, *PAP* etc., welche die sensiblen Benutzerdaten praktisch im Klartext übermitteln. Abhängig von der Art der belauschten Daten, erlauben die durch diesen Angriff gewonnenen Informationen dem Angreifer, eine falsche Identität anzunehmen oder verschaffen ihm Vorteile gegenüber anderen Menschen, die über diese Informationen nicht verfügen oder sie für vertraulich halten.

- **Die Analyse des Kommunikations-Verhaltens.** Die im Rahmen dieses Angriffes gesammelten Informationen erlauben es dem Angreifer festzustellen, wer mit wem zu welchem Zeitpunkt kommuniziert. Diese können später nach bestimmten Regelmäßigkeiten untersucht werden und verraten dem Angreifer Gewohnheiten der Kommunikationspartner. So kann sich ein Angreifer z.B. die Struktur unterschiedlicher Geschäftsprozesse eines Unternehmens erschließen.

- **Datenflussanalyse.** Die Datenflußanalyse ähnelt in ihrer Struktur sehr der Analyse des Kommunikationsverhaltens. Bei diesem Angriff sind die übermittelten Daten dem Angreifer ebenfalls nicht zugänglich[28]. Der Angreifer ist jedoch trotzdem in der Lage (z.B. anhand des erhöhten Kommunikationsaufkommens), Rückschlüsse auf bestimmte Vorgänge zu ziehen. Diese Technik wird besonders oft von den Geheimdiensten eingesetzt, die zwar nicht immer in der Lage sind, die übermittelten Daten zu dekodieren, die

[28] Denn die Kommunikation kann z.B. verschlüsselt erfolgen.

Veränderung des Kommunikationsvolumens in bestimmten Regionen mit der Zunahme terroristischer Aktivitäten jedoch trotzdem verbinden können.

Passive Angriffe sind relativ schwer festzustellen. Aus diesem Grund steht bei der Abwehr solcher Angriffe nicht die Reaktion im Vordergrund, sondern eine gut durchdachte Konzeption der Systeme vor ihrer Inbetriebnahme. So kann z.B. die Verwendung von Kabeln mit spezieller Abschirmung das Abhören von Datenleitungen unmöglich machen. Die Datenflussanalyse kann z.B. durch Generierung eines kontinuierlichen fiktiven Datenverkehrs erschwert werden. Aus diesem Grund steht die Abwehr von passiven Angriffen nicht im Vordergrund nachfolgender Ausführungen. Wir werden uns vorwiegend mit der Abwehr aktiver Angriffe beschäftigen.

3.2.2 Aktive Angriffe

Aktive Angriffe sind vorwiegend gegen die Verfügbarkeit, Integrität oder Authentizität von Daten gerichtet. Aktive Angriffe können sein:

- **Verzögerung oder das wiederholte Senden von Informationen.** Hierdurch kann der Empfänger zu einer falschen Aktion gezwungen werden und könnte z.B. eine Geldtransaktion mehrfach veranlassen.
- **Veränderung übermittelter Daten.** Der Angreifer könnte z.B. in einer digital übermittelten Rechnung die Höhe der zu überweisenden Summe oder die Daten des gewünschten Empfängers verändern.
- **Denial of Service (*DoS*).** Der Angreifer hindert einen Anwender an der Nutzung von Diensten. Er überlastet unbefugt ein System, damit es seinen eigentlichen Aufgaben nicht nachkommen kann.
- **Maskeradeangriff.** Bei dieser Art von Angriffen nimmt der Angreifer eine falsche Identität an und agiert mit den Berechtigungen des gefälschten Systems oder Benutzers.

- **Session Hijacking.** Der Angreifer übernimmt eine bestehende Verbindung zwischen zwei Kommunikationspartnern und setzt seinen Angriff nach dem *Maskerade*-Prinzip fort.

- **Ausnutzung von Softwarefehlern.** Die Fehlerfreiheit heutiger Software kann, abgesehen von wenigen Ausnahmen, nur sehr schwer oder erst gar nicht bewiesen werden. Ein Angreifer nutzt die Softwarefehler mit Hilfe von *Exploits* aus[29].

3.3 Schutz der Systeme

Heutzutage wird verstärkt über den vermeintlichen digitalen Krieg gesprochen. Tatsächlich ist es so, dass boshafte Aktivitäten im Internet in der letzten Zeit beinahe unerträglich geworden sind. Nervende Spamflut[30], täglich neue Varianten diverser Schädlinge, ständige Meldungen über die auf Grund von *DoS*-Angriffe nicht erreichbaren Systeme. Diese Entwicklung ist alles andere als zufrieden stellend[31]. Aus diesem Grund verlangen viele Mitglieder der Internet-Gemeinschaft nach einer aktiven Abwehr von Angriffen[32]. Die im Folgenden vorgestellten Konzepte sollen sich von den vielfach vorgeschlagenen Formen der aktiven Verteidigung und der Gegenangriffe distanzieren. Die Verfolgung von Internet-Verbrechern kann und darf nicht die Angelegenheit von Durchschnittsbürgern sein, denn viel zu

[29] Vgl. a. [DOMHAN 2001]

[30] Zunehmend werden E-Mails zum Klau von Online-Identitäten eingesetzt. Aus diesem Grund kann Spam als eine Art des aktiven Angriffs eingestuft werden. Auch wenn die Spam-Bekämpfung nicht zur eigentlichen Aufgabenstellung dieser Arbeit gehört, soll hier nicht unerwähnt bleiben, dass das beschriebene Agentennetz zu einem effektiven Werkzeug gegen Spam entwickelt werden kann. Die einfachste Möglichkeit besteht dabei im Austausch von *Hash*-Werten empfangener Spam-Mails und von IP-Adressen der Spam-Versender. Die einschlägig bekannten Spammer und deren Nachrichten könnten dadurch vom Agentennetz schnell identifiziert und gefiltert werden. Vgl. a. [BLEICH 2004], [METZGER 2003]

[31] Wie viele andere Internetnutzer wäre der Autor dieses Textes gerne bereit, eine gewisse Geldsumme aufzuwenden, um mit einem der vielen Spamversender über die negative Seite seines Tuns zu diskutieren, wobei die Anwendung körperlicher Gewalt während dieser Diskussion nicht ausgeschlossen werden könnte.

[32] Vgl. a. [SCHNEIER 2001]

leicht kann der Angreifer seine Identität fälschen (*Maskeradeangriff*), so dass sich der Gegenangriff gegen unschuldige Personen und unbeteiligte Organisationen richten könnte. Nichtsdestotrotz stellt das im Rahmen dieser Arbeit ausgearbeitete Konzept des Agentennetzes ein hervorragendes Werkzeug für die aktive Angriffsbekämpfung und die Durchführung von Gegenangriffen dar. Das im folgenden Kapitel beschriebene Agentennetz ermöglicht eine sehr einfache Erfassung der Angreifer-Aktivitäten. Die Bündelung von Ressourcen einzelner Agenten erhebt ein größeres Agentennetz zu einem mächtigen Gegner, gegen den vereinzelte Angreifer keine Chancen haben. So kann das Netz z.B. die Aktivitäten des Angreifers wesentlich ausführlicher protokollieren, als die einzelnen Systeme es unabhängig voneinander tun könnten. Die dadurch gesammelten Beweise wären eine große Hilfe für die Ermittlungsbehörden. Das Agentennetz könnte auch z.B. einen besonders lästigen Angreifer mit einem gerichteten *DoS*-Angriff unschädlich machen, indem es z.B. seine Leitung mit Datenpaketen überflutet. Dies wird durch den Zusammenschluss von einzelnen Einheiten zu einem Gesamtsystem möglich.

Ein Multiagentensystem verfügt über das Mehrfache an Ressourcen seiner Bestandteile[33]. Dies ermöglicht nicht nur die Durchführung aktiver Angriffe, sondern auch eine effizientere Immunisierung des gesamten Systems z.B. durch eine schnellere Signaturverteilung durch optimierte Leitungsauslastung. Die gemeinsame Abwehr von Angriffen ist kein vollkommen neues Konzept. Bereits die heutzutage üblichen Antivireninfrastrukturen beherrschen den Signaturenupdate für Client-Systeme per Upload-Anweisung des Zentral-Rechners. Der Hauptvorteil eines agentenbasierten Systems besteht darin, dass der Zusammenschluss von Agenten zu einer Einheit durch das autonome Agieren von Agenten dezentral erfolgt und auch dezentral gesteuert wird[34]. Für einen Angreifer ist es nicht möglich, das Gesamtsystem durch die Ausschaltung seiner einzelnen Bestandteile[35] zu zerstören. Autonome Agenten sind in der Lage, die Struktur ihres Netzes

[33] einzelne Agenten

[34] Der Verwaltungsaufwand des Gesamtsystems wird auf seine Bestandteile verteilt, was zur Robustheit des agentenbasierten Systems beiträgt.

[35] Wie z.B. der Angriff auf den Signaturserver beim klassischen System.

selbständig zu ändern[36], ohne dass ein administrativer Eingriff notwendig ist.

Die Ansätze des Gegenangriffs sollen im Rahmen dieser Arbeit nicht ausgearbeitet werden, denn obwohl deren technische Umsetzung relativ einfach ist, ist die Rechtmäßigkeit solcher Maßnahmen äußerst fraglich. In den folgenden Ausführungen liegt der Schwerpunkt auf den passiven Formen des Systemschutzes, welche von der technischen Seite nicht weniger anspruchsvoll sind als die aktiven Maßnahmen und gleichzeitig von der rechtlichen Seite vollkommen unbedenklich sind.

3.4 Methoden der Angriffserkennung

Angriffe können auf vielfältige Weise erkannt werden. Grundsätzlich unterscheidet man zwischen den folgenden drei Methoden:

- Signaturbasierte Angriffserkennung
- Generische Angriffserkennung
- Heuristische Analyse

3.4.1 Verwendung bekannter Signaturen

Es handelt sich dabei um die einfachste Technik, bei welcher der Datenstrom auf einschlägig bekannte Bitsequenzen geprüft wird. Diese Bitsequenzen heißen Signaturen und sind charakteristisch für bestimmte Schädlinge bzw. deren Angriffe. 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.

[36] Z.B. durch die Anpassung der Kontaktlisten oder ihres Verhaltens an den aktuellen Umgebungszustand. Änderungen der bestehenden Leitungskapazität, das Hinzukommen neuer Verbindungsmöglichkeiten etc.

3.4.2 Generische Erkennung

Im Rahmen der generischen Angriffserkennung werden Varianten der bereits bekannten Schädlinge bzw. Angriffe erkannt. Besonders effektiv ist diese Methode bei *Makroviren*, da hier die für einen Schädling typischen Sequenzen besonders leicht erkannt werden können. Diese Sequenzen bleiben auch bei Virenvarianten stabil, auch wenn deren Verhalten sich teilweise sehr stark voneinander unterscheidet. Generische Erkennung schlägt jedoch in der Regel bei neuen, noch nicht bekannten Angriffen fehl.

3.4.3 Heuristische Analyse

Die heuristische Analyse ist in der Lage, Angriffe zu erkennen, deren Signaturen nicht in der Signaturdatenbank aufgeführt sind. Innerhalb der heuristischen Analyse arbeitet man entweder mit statischen Methoden (man sucht nach verdächtigen Datenkostrukten) oder man führt eine dynamische Analyse durch die Code-Emulation durch. Das verdächtige Datenfragment wird dabei in einer geschützten emulierten Umgebung auf die Schadensfunktion untersucht.

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 z.B. vom Scanner 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[37].

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

- etc..

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.

3.5 Integritätskontrolle des Signatursatzes

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.

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. Denn 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 mehr enthalten. Diesen Angriff könnte Malloy anschließend bei Bob anwenden, ohne die Gefahr zu laufen, dass Bob diesen Angriff erkennen oder abwehren kann. 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.

Dazu berechnet Alice vor dem Übersenden von Daten an Bob eine Checksum-

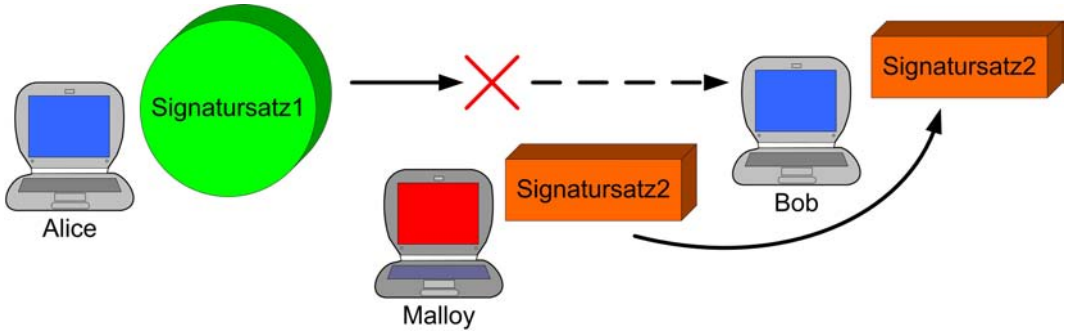

Abbildung 2: Man in the Middle-Angriff (Malloy verändert Signaturen während der Datenübertragung).

me ihrer Daten und veröffentlicht diese auf einem vertrauenswürdigen System oder übermittelt sie über einen sicheren Kanal an Bob. Nach dem Datenempfang berechnet Bob die Checksumme der von ihm empfangenen Daten und vergleicht sie mit dem von Alice veröffentlichten Wert. Sind beide Checksummen nicht identisch, wurden die Daten während der Datenübermittlung verändert. Selbstverständlich könnte Alice ihre Signaturen an Bob direkt über einen sicheren Kanal senden. Die gesicherte Übertragung eines kompletten Signatursatzes wäre allerdings wesentlich aufwendiger als die Übermittlung des relativ kurzen *Hash*-Wertes. Das direkte Versenden des Signatursatzes von Alice zu Bob setzt voraus, dass die beiden Kommunikationspartner während der Datenübertragung eine Verbindung zueinander besitzen, was abhängig von der Zeit und der Lokation kostspielig sein kann. Die Benutzung eines *Hash*-Wertes ermöglicht eine indirekte Datenübertragung, wobei der Signatursatz auf anderen Systemen zwischengespeichert werden kann. Durch die Zwischenspeicherung des Signatursatzes können Alice und Bob ihre Kosten der Datenübertragung optimieren, indem sie die für sich jeweils günstigste Verbindung aussuchen. Eine direkte Verbindung zwischen Alice und Bob muss in dem Fall nur während der Übertragung der kurzen Checksumme bestehen.

Eine Checksumme wird in aller Regel nach der *CRC*-Methode berechnet. Auch die kleinste Veränderung der übermittelten Daten[38] bedingt eine völlig neue

[38] z.B. das „Kippen“ eines Datenbits

Prüfsumme. Das zur Berechnung der Prüfsumme verwendete Verfahren muss öffentlich bekannt sein. Das bedeutet, dass ein potenzieller Angreifer, dieses auf seine gefälschten Daten anwenden könnte, um z.B. eine neue Prüfsumme zu berechnen und diese an Bob zu 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[39].

3.6 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[40]. Die verfügbaren Rechenkapazitäten der mobilen Geräte sind in der Regel geringer als die der stationären Systeme[41]. 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 an erster Stelle 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 niemandem garantiert werden. Diesem Umstand muss bei der Konzeption der Angriffserkennungssoftware für die mobilen Geräte Rechnung getragen werden. Im Folgenden wird gezeigt, wie trotz der relativ geringen Leistungsfähigkeit die-

[39] Vgl. a. [FUHS 1995]

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

[41] Am häufigsten sind die Einschränkungen der Rechenleistung durch die Akkukapazität und die aufwendige Wärmeabfuhr bedingt.

ser Systeme, deren angemessener Schutz durch den Einsatz von Agentensystemen gewährleistet werden kann.

3.7 Der klassische Aufbau einer Sicherheitsinfrastruktur

Vor einiger Zeit war ich für eine Lebensversicherungsgesellschaft tätig. In dieser Branche stellen Informationen, bzw. deren Vertraulichkeit die Geschäftsgrundlage dar. Dementsprechend aufmerksam wird das Thema „Sicherheit und Datenschutz“ behandelt. Während meiner Tätigkeit für das Unternehmen arbeitete ich ein Konzept zur Einführung eines Intrusion Detection Systems aus und führte eine Pilotinstallation durch. Während der Durchführung von Security Audits lernte ich den Aufbau der Antiviren-Infrastruktur kennen. Der unternehmensweite Einsatz des Produktes eines der führenden Antivirensoftwareherstellers 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 der hohen Qualifikation und eines enormen Zeitaufwandes konnte es das Systemadministratoren-Team nicht verhindern, dass es immer wieder zu Sicherheitszwischenfällen kam.

In der Abbildung 3 ist der vereinfachte Aufbau des Systems dargestellt. Der Administrationsrechner bezog seine Updates automatisch von der Webseite des Softwareherstellers und versorgte damit die Client-Stationen. Von der Administrationskonsole aus konnte der Administrator die Regelwerke der entsprechenden Clients ändern und auch bestimmte Befehle[42] an diese übermitteln.

Der auf den ersten Blick sichere Aufbau hatte jedoch einige Schwachstellen. Die Versorgung der Client-Rechner mit den aktuellen Signaturen war eine der Komplikationen. In der Datenbank des Administrationsservers wurden die von ihm zu betreuenden PCs eingetragen. Dieser prüfte in periodischen Zeitabständen, ob seine Clients online waren und aktuelle Signaturen verwendeten. Diese Vorgehens-

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

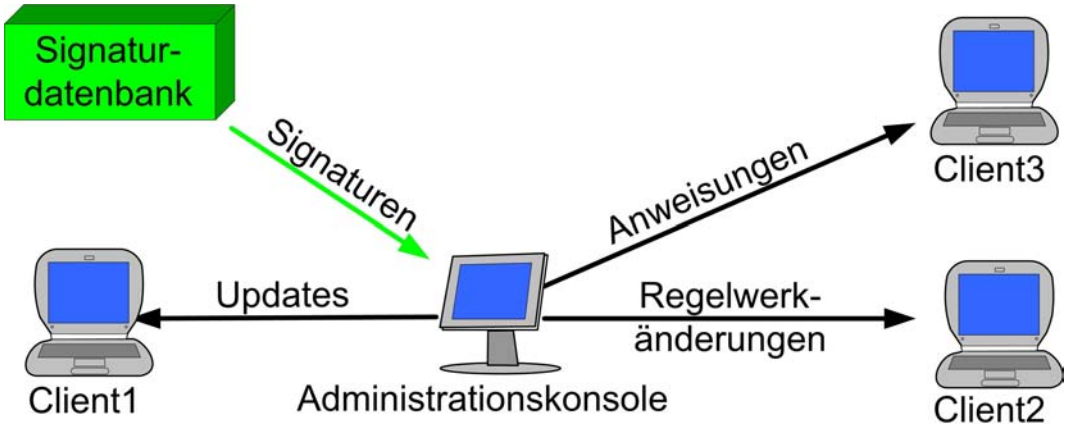

Abbildung 3: Der klassische Aufbau einer Antivireninfrastruktur.

weise führte zu erheblichen Problemen zu Beginn des Arbeitstages, da Hunderte von PCs fast gleichzeitig eingeschaltet wurden. Der Update-Server entdeckte sie 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 verschiedener Systeme waren zu einem Zeitpunkt selten auf dem gleichen Stand, was bei der für Computerviren üblichen rasanten Verbreitungsgeschwindigkeit eine Katastrophe bedeuten könnte. 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. Diese „Lösung“ brachte allerdings weitere Probleme mit sich. Obwohl die eingesetzten Büro-PCs für den Dauerbetrieb ausgelegt waren, kam es zunehmend zu Hardwareausfällen, was zu einer Erhöhung von Reparaturkosten führte[43].

Ein anderer interessanter Aspekt dieser Vorgehensweise ist an dieser Stelle ebenfalls zu erwähnen: Antivirenhersteller können auf neue Angriffe lediglich rea-

[43]Ein weiteres Problem könnte z.B. die Erhöhung der Brandgefahr sein.

gieren. Sobald eine neue *Malware* bekannt wird, versuchen sie, die dazugehörige Signatur zu ermitteln und diese an ihre Kundschaft weiterzugeben. Diese Prozedur kann Stunden in Anspruch nehmen, was bedeutet, dass in dieser Zeit die Systeme dem neuen Angriff schutzlos ausgeliefert sind[44]. Ein weiterer großer Nachteil der beschriebenen Infrastruktur bestand darin, dass die Antivirenclients nur lokal[45] 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 seitens 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 und die Funktionsweise 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 beschriebene Problematik zwar entschärfen, jedoch nicht vollständig 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 Antivirensoftware-Herstellers. Sobald sie (z.B. aufgrund eines *DoS*-Angriffs) nicht erreichbar werden, können die Signaturupdates nicht mehr weiterverbreitet werden und sämtliche Clients bleiben in der Zeit ohne einen vernünftigen Schutz.

[44] 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 Signaturupdates ohne Schutz, etwas übertrieben ist.

[45] Auf dem lokalen System.

Diese und viele andere Probleme sind mit dem Einsatz der zentralisierten Architektur verbunden. Im folgenden Kapitel werden einige, auf dem Agentenansatz basierende, Konzepte präsentiert, die viele der beschriebenen Schwächen nicht besitzen und in der Lage sind, einen angemessenen Systemschutz zu gewährleisten.

3.8 Zusammenfassung

In diesem Kapitel wurden Grundlagen der Angriffserkennung vorgestellt. Der Leser wurde außerdem mit einigen damit verbundenen Problematiken konfrontiert. Die Notwendigkeit der Authentizität des bei der Angriffserkennung verwendeten Signatursatzes wurde erläutert und ein technischer Vorschlag zu dessen Gewährleistung wurde gemacht. Der Darstellung des klassischen Aufbaus einer Sicherheitsinfrastruktur am Beispiel der Antivirensoftware folgte die Beschreibung der Flaschenhals-Eigenschaften einiger Infrastrukturbestandteile, was dem Leser die Notwendigkeit einer besseren Lösung offenbarte.

4 Agentenbasierter Schutz

4.1 Ressourceneinsparung durch Alarmstufendifferenzierung

Im vorhergehenden Kapitel wurde der klassische Aufbau einer Antivireninfrastruktur und dessen Nachteile erläutert. Es wurde gezeigt, dass Systeme häufig unnötig viel Ressourcen zur Erkennung von Angriffen verbrauchen[46]. Dabei verfügen die Systeme nicht über gleich viele Ressourcen und können daraus folgernd nicht die gleichen Softwareprodukte zur Gewährleistung des eigenen Schutzes verwenden.

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[47], um sich eine längere Zeit im „Stresszustand“ zu befinden. Es liegt deswegen auf der Hand, das System nur

[46] Dabei spielen in die Angriffserkennung investierte Ressourcen eine wesentlich wichtigere Rolle, als man es auf den ersten Blick glauben könnte. Am vierzehnten Juni 2004 wurde der erste *Computerwurm* offiziell registriert, dessen primäre Angriffsziele mobile Geräte (*Bluetooth*-Technologie unterstützende Handys und PDAs) waren. Vgl. a. [FERRIE 2004]. Obwohl dieser *Wurm* über keinerlei Schadensfunktionen verfügte und seine Ausbreitungsgeschwindigkeit aufgrund der eingeschränkten *Bluetooth*-Reichweite nicht mit der seiner gewöhnlichen Brüder verglichen werden konnte, setzte er ein deutliches Zeichen. Denn zu diesem Zeitpunkt existierte lediglich ein einziges Antivirenprodukt, das in der Lage war, mobile Geräte vor ähnlichen Angriffen zu schützen. Sehr deutlich wurde dadurch die Tatsache, dass die Antivirenhersteller besonders im mobilen Sektor noch Einiges nachzuholen haben. In Zukunft ist das vermehrte Auftreten ähnlicher Vorfälle zu erwarten.

[47] Man denke beispielsweise an die mobilen Geräte, bei denen Energieversorgung die Rechenleistung der Geräte strikt limitiert.

dann stark auszulasten, wenn dies wirklich notwendig ist. Im Folgenden wird dieses Prinzip anhand eines Modells demonstriert.

4.2 Gedankenexperiment - Modellaufbau

Bei dem Aufbau für unser Gedankenexperiment gehen wir von einem fiktiven Computersystem aus. Wir gehen weiterhin von der Annahme aus, dass für dieses System zehn Arten von Angriffen sowie die Signaturen dieser Angriffe bekannt sind. Um ein ankommendes Datenpaket mit einer Signatur zu überprüfen, verbraucht das System eine Ressourceneinheit. Unser System besitzt insgesamt zehn Ressourceneinheiten und ist somit zwar in der Lage, alle zehn Angriffssignaturen im Speicher zu halten und die ankommenden Datenpakete damit zu überprüfen, hat allerdings in diesem Fall keine Kapazitäten mehr, seinen anderen Aufgaben nachzugehen. Die natürliche Lösung besteht darin, das System nur mit einem Bruchteil des Signatursatzes zu betreiben, so dass die Angriffserkennung nicht die kompletten Systemressourcen für sich beansprucht. Im in der Abbildung 4 dargestellten 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 es sich, in den Zustand „yellow“ zu wechseln. Jetzt arbeitet das System mit einem erweiterten Signatursatz und hat zwar weniger freie Ressourcen zur Verfügung, kann jedoch dafür 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.

4.3 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, der

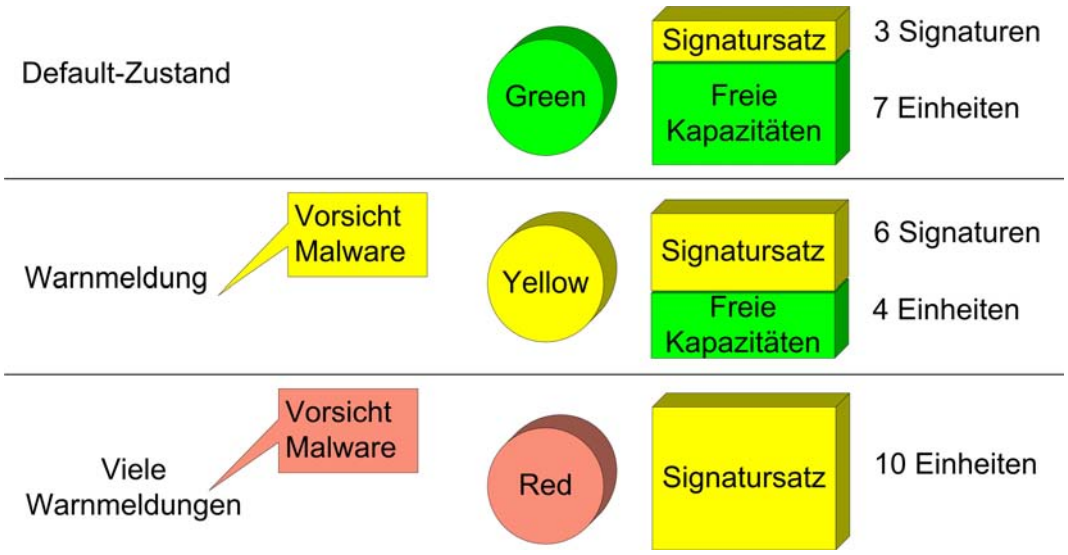

Abbildung 4: Zusammenhang zwischen der aktuellen Alarmstufe und dem verwendeten Signatursatz.

zu Stoßzeiten zu einem Flaschenhals werden könnte. Durch die Eliminierung zentraler Systeme verschwinden außerdem die bei den Angreifern besonders beliebten Ziele, die bildlich gesehen, die Achillesfersen von Infrastrukturen darstellen.

Im agentenbasierten Ansatz wird jeder Rechner durch einen oder durch mehrere Agenten geschützt. Die Konfigurationen von Agenten können 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 müssen ein gemeinsames *Protokoll* unterstützen, welches ihnen Nachrichten- und Datenaustausch ermöglicht. Die Abbildung 5 veranschaulicht den Aufbau einer solchen Struktur.

Die Agenten eins und zwei arbeiten mit vergleichsweise kleinen Regelwerken. Agent drei ist dagegen leistungsfähig genug, um von einem erweiterten Signatursatz (alle bekannten Signaturen) und Heuristik Gebrauch zu machen. Auf den ersten Blick scheinen die ersten beiden Systeme einen geringeren Schutz als das dritte System zu besitzen. Gleichzeitig verfügt der Agent3 über die notwendigen Voraussetzungen, um alle gegen das System bekannten Angriffe zu erkennen[48].

[48] Agent3 ist somit in der Lage, Angriffe zu erkennen, die nicht vom Agent1 und Agent2 erkannt

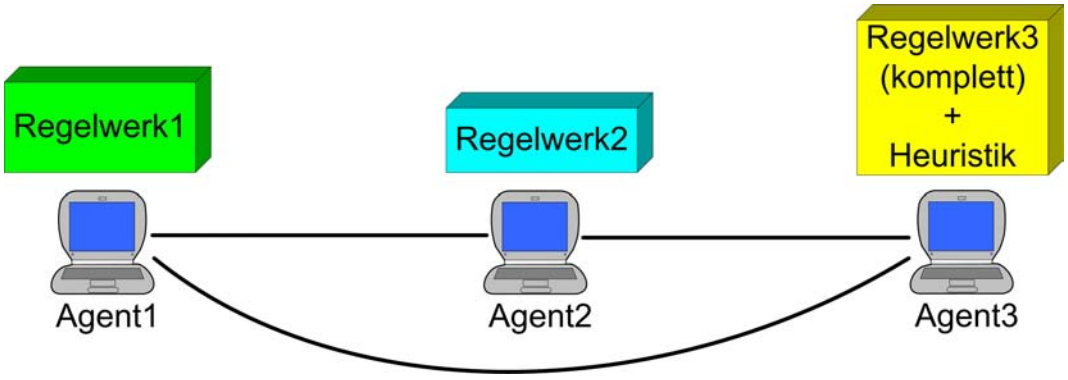

Abbildung 5: Verwendung unterschiedlicher Regelwerke.

Sobald dieser einen solchen Angriff feststellt, übermittelt er eine entsprechende Warnung an die beiden Nachbar-Systeme mit dem Hinweis auf die Signatur des von ihm erkannten Angriffs. Unter der Voraussetzung, dass der dritte Agent in seiner Datenbank einen funktionierenden Patch gegen den Angriff hat, kann er diesen ebenfalls an seine Nachbarn senden, so dass sie sich aus eigener Kraft von der Infektion befreien können. Die beiden Agenten müssen anschließend die entsprechende Signatur in ihre Signatursätze aufnehmen, um vor diesem Angriff in Zukunft geschützt zu sein. Die beschriebenen Prozesse werden in der Abbildung 6 veranschaulicht.

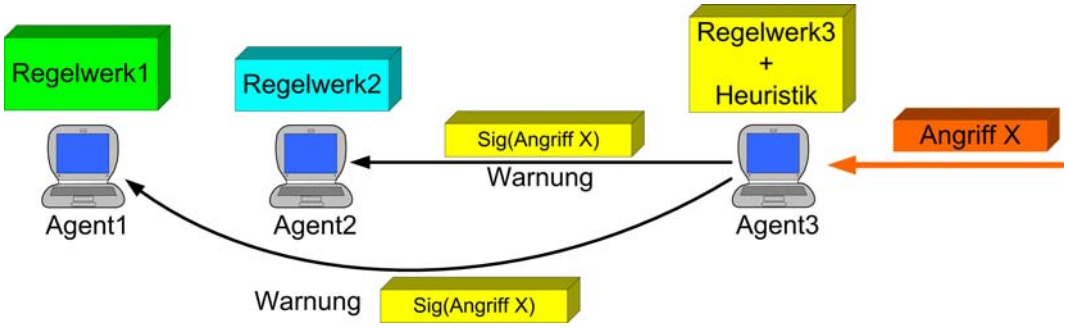

Abbildung 6: Bekanntgabe der Angriffssignatur.

Eine solche Vorgehensweise ermöglicht einen besonders sparsamen Umgang mit den Ressourcen der zu schützenden Rechner. Die Aufnahme einer neuen Signatur in die Signaturdatenbank kann mit einer Schutzimpfung verglichen werden. Die

werden können.

wenigsten Menschen kommen auf die Idee, sich im Hochsommer einer Grippeimpfung zu unterziehen, denn sie ist in der Regel vollkommen überflüssig und belastet lediglich das Immunsystem. Auf der anderen Seite würde sich jeder vernünftige Mensch eine Schutzimpfung gegen Malaria gönnen, wenn er sich im Sommer die tropischen Wälder als Urlaubsziel ausgesucht hat. Es macht Sinn, die Signatur eines neuen Angriffes erst dann in die Signaturdatenbank aufzunehmen, wenn dieser Angriff wahrscheinlicher als andere Angriffe wird. Wenn die Gefahr des Angriffes nicht mehr gegeben ist und die Agenten eine bestimmte Art von Angriffen nicht zu erwarten haben, können die entsprechenden Signaturen als nicht mehr aktuell eingestuft und wieder aus dem Signatursatz entfernt werden[49], was ebenfalls mit der natürlichen Vorgehensweise zu vergleichen wäre.

Die dezentralisierte Struktur eines Multiagentensystems hat weitere angenehme 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 Kapazitätsengpässe zu vermeiden. Dies kann am einfachsten auf der Basis des von uns eingeführten Gedankenmodells dargestellt werden.

Wir kehren zu dem von uns behandelten Gedankenmodell zurück, bei dem Agent3 einen Angriff feststellen konnte und nun versucht, die ihm bekannten Agents über diese Gefahr zu benachrichtigen. Seine Ressourcen reichen jedoch lediglich aus, um die Signatur des Angriffs dem Agent2 zu übermitteln[50]. Nichtsdestotrotz kann Agent1 die von ihm benötigte Signatur in seine Datenbank auf-

[49] 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.

[50] So könnte Agent3 danach z.B. aufgrund eines Netzwerkfehlers nicht mehr verfügbar sein.

nehmen, indem er diese vom Agent2 bezieht. Abbildung 7 veranschaulicht den beschriebenen Prozess. Agent2 wäre außerdem in der Lage, einen Teil der Serveraufgaben zu übernehmen und die ihm bekannten Agenten über die Existenz einer Bedrohung und das Vorhandensein einer Signatur des Angriffs zu benachrichtigen. Dieser Vorgang könnte so lange fortgesetzt werden, bis alle Rechner im bedrohten Netzwerksegment über einen aktualisierten Signatursatz verfügen.

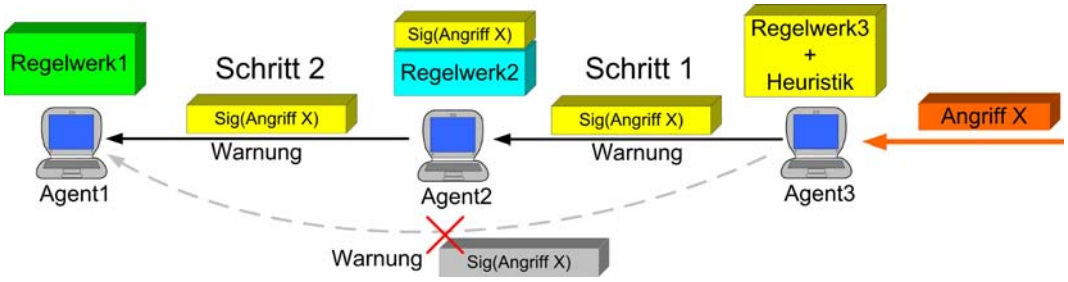

Abbildung 7: Bezug von Signaturen durch Agent2 von Agent1 bei der Nichtverfügbarkeit von Agent3.

Es ist klar zu sehen, dass der beschriebene Aufbau leicht von einem Angreifer ausgenutzt werden kann, indem er einige Falschmeldungen generiert und diese an die Teilnehmer des Netzwerks verschickt. Agenten würden diese an die anderen Hosts weiterleiten, was dazu führt, dass das gesamte Multiagentensystem sich bereits nach einer kurzen Zeit im „Stresszustand“ befindet[51]. 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 der Lage sind, ihr Vertrauen zu einem System zum Ausdruck zu bringen und es gegebenenfalls zu ändern. So können beispielsweise einige Falschmeldungen von Seiten eines Agenten dazu führen, dass seine Warnungen in Zukunft ignoriert werden[52]. Um das „Hochschaukeln“ der Warnstufe des gesamten Systems zu vermeiden, kann die Tatsache ausgenutzt werden, dass *Malware* sich in der Regel nicht wahllos verbreitet[53]. Die Wahrscheinlichkeit, dass sich der

[51] Agenten arbeiten mit unnötig vielen Signaturen, überflüssiger Netzwerkverkehr wird erzeugt.

[52] Eine Nachricht könnte z.B. als eine Falschmeldung eingestuft werden, wenn es zu keinem Angriff mit gemeldeter Signatur kommt. Ebenfalls verdachtserregend sind zu viele unterschiedliche Warnungen, wenn sie innerhalb einer kurzen Zeitspanne von einem System kommen.

[53] Dies resultiert aus der Tatsache, dass die Verbindung nach außen von Rechnern eines Netz-

Angriff an erster Stelle gegen die Nachbarn eines Systems und nicht gegen einen Rechner am anderen Ende der Welt richtet, 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. Eine ähnliche Vorgehensweise wird zum Beispiel bei *ICMP*-Paketen verwendet[54] und sorgt dafür, dass das Netz nicht von alten, aus welchen Gründen auch immer, nicht mehr notwendigen Paketen überflutet wird. An dieser Stelle könnte man z.B. ein Datenpaket mit einem Zeitzähler ausstatten, der seine Lebenszeit nicht durch die Anzahl von Hops, sondern anhand des zeitlichen Faktors bestimmen könnte. Allerdings ist die Zeitmessung und insbesondere der Zeitabgleich innerhalb einer großen inhomogenen Architektur wesentlich problematischer als die Verwendung dieser einfachen und gleichzeitig effektiven Methode.

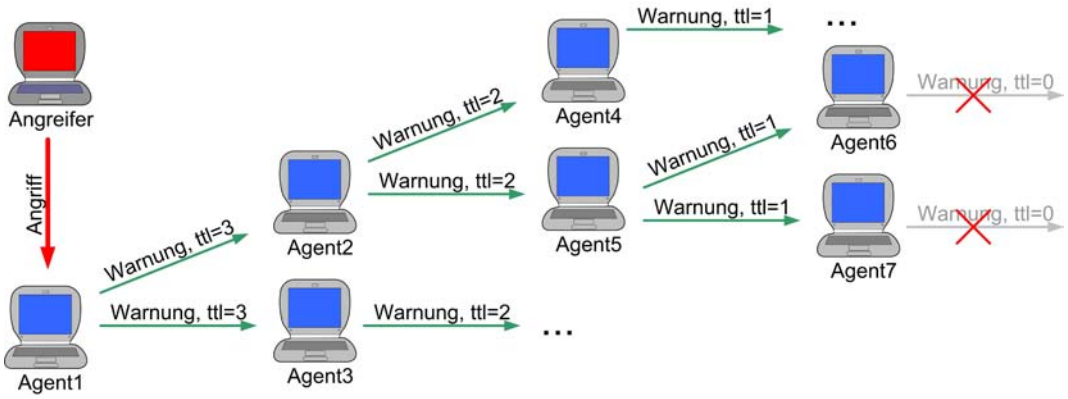

Abbildung 8: Jede Warnmeldung wird mit einem Zähler versehen, der bei jedem Hop decrementiert wird. Beim Zählerstand 0 (null) wird die Nachricht verworfen.

Der beschriebene kleine Trick sorgt dafür, dass es dem Angreifer wesentlich schwerer fällt, das Netz mit Falschmeldungen zu überfluten. Dies hindert ihn jedoch nicht daran, die Nachrichten in einer großen Anzahl zu generieren und

werks in der Regel über eine zentrale Stelle erfolgt, die widerrum relativ gut geschützt ist. Deswegen breiten sich die Schädlinge normalerweise wesentlich intensiver innerhalb von abgeschlossenen Netzsegmenten aus.

[54] Vgl. a. [POSTEL 1981]

diese an die Rechner im Netzwerk zu verschicken. Obwohl Meldungen nur eine beschränkte Anzahl von Hops weitergeleitet werden und der Schaden des Angriffs somit begrenzt ist, könnte der Angreifer bei einer hohen Kapazität der ihm zur Verfügung stehenden Leitung viel Unheil in den Systemen in seiner Nähe anrichten und den Betrieb des Netzes beträchtlich stören. Besonders gefährlich wäre ein von Seiten mehrerer Angreifer durchgeführter Angriff. Diese könnten ihre Netzwerkkapazitäten bündeln und größere Netzsegmente außer Funktion setzen. Deswegen muss gewährleistet werden, dass nicht jeder beliebiger Agent zum Mitglied des Netzes werden kann. Nur diejenige Agenten, die ein gewisses Vertrauen der Netzmitglieder genießen[55], dürfen sich mit dem Agentennetz verbinden.

Um die Vertrauensbeziehungen zwischen den Mitgliedern eines Netzwerks aufzubauen bedarf es zuerst eines Instruments, um die Authentizität der Mitglieder zu gewährleisten. Dafür eignet sich der Austausch von öffentlichen Schlüsseln der jeweiligen Agenten. Jedes Mitglied des Agentennetzes muss folglich seinen öffentlichen Schlüssel den anderen Mitgliedern verfügbar machen. Sobald ein Agent das Agentennetz betritt und die Verbindung zu den Netzmitgliedern aufnimmt, erhält er eine Nachricht von dem Host, mit dem er den Initialisierungshandshake durchführen will. Diese Nachricht ist mit dem öffentlichen Schlüssel des Newcomers verschlüsselt, so dass nur er in der Lage ist, diese zu entschlüsseln und dem Anfrager-Agenten im Klartext zurückzusenden[56]. Die beschriebene Vorgehensweise ist der Abbildung 9 zu entnehmen.

Dazu speichert jeder Agent seinen öffentlichen Schlüssel auf einem zentralen Schlüsselverwaltungsserver oder sendet diesen am Anfang einer Kommunikationssession an seinen Kommunikationspartner. Man kann nur bedingt sicherstellen, dass der öffentliche Schlüssel, welchen man von einem Agenten erhält, auch tatsächlich ihm gehört. Deswegen ist eine solche Infrastruktur einem *Man in the*

[55] Agenten, die von den anderen Netzmitgliedern zu der Netzbenutzung autorisiert sind.

[56] Die Antwort könnte auch mit Hilfe des öffentlichen Schlüssels des Anfrager-Hosts verschlüsselt werden, um z.B. den Nachrichteninhalt als Session-Key für die Verbindung zu verwenden.

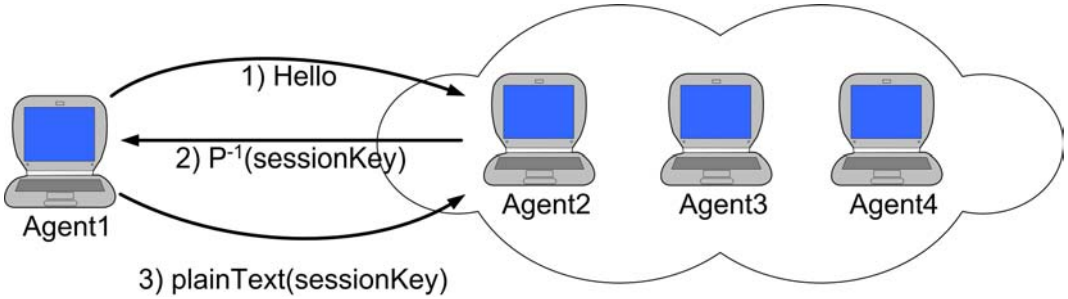

Abbildung 9: Das Identifizieren von Agenten innerhalb des Netzes.

Middle-Angriff hilflos ausgesetzt.

Die beschriebene Problematik ist dem folgenden Szenario zu entnehmen. Agent1 will sich mit dem Agentennetzwerk verbinden und sendet am Anfang einer Sitzung seinen öffentlichen Schlüssel an Agent2. Ein Angreifer (Agent3) schleicht sich noch vor dem Verbindungsaufbau in den Kommunikationskanal der beiden Parteien, fängt die Nachricht von Agent1 ab und sendet seinen Schlüssel an Agent2. Dasselbe passiert auch in der Gegenrichtung. Agent3 fängt den Schlüssel von Agent2 ab und sendet seinen eigenen Schlüssel an Agent1. Agent1 und Agent2 haben nun keine Ahnung davon, dass sie nicht den Schlüssel ihres Gegenübers besitzen. Sämtliche Kommunikation zwischen den beiden Parteien verläuft nun über den Angreifer. Er ist in der Lage, die Nachrichten der beiden Kommunikationspartner zu entschlüsseln und diese, mit Hilfe der ihm jetzt bekannten Schlüssel zu verschlüsseln und weiter zu versenden.

Die Überprüfbarkeit der Schlüssel muss also gewährleistet werden. Die Vertrauensbeziehungen zwischen den Mitgliedern des Netzes könnten nach dem Vorbild der Zertifizierungsstellen gestaltet werden. In diesem Fall würde eine zentrale Stelle die Schlüssel der Kommunikationspartner mit ihrem eigenen Signieren, was deren Authentizität bestätigen würde. Der Vorteil besteht in einer sehr einfachen Implementierung und Verwaltung der sich daraus ergebenden Infrastruktur. Gleichzeitig wäre die Sicherheit des Netzes von dieser Stelle abhängig, denn würde sie z.B. aufgrund der Kompromittierung durch einen Hacker falsche Informatio-

nen liefern, wäre die Sicherheit des gesamten Netzes gefährdet. Unübersehbar ist außer dieser nicht erwünschten Abhängigkeit der relativ hohe Verwaltungsaufwand, denn die Speicherung und die Verwaltung entsprechender Datensätze erfordert eine komplexe Infrastruktur. Diese muss ständig gepflegt und vor Angriffen geschützt werden, was verständlicherweise mit gewissen Kosten verbunden ist.

Die Netzgemeinschaft hat auf diese Frage bereits vor langer Zeit eine Antwort gefunden. Die Lösung besteht darin, die oben beschriebenen Informationen nicht an einer zentralen Stelle zu speichern, sondern die Datenhaltung den Mitgliedern des Netzes zu überlassen. Der Vorteil dieser Lösung besteht in ihrer völligen Unabhängigkeit von sämtlichen Zertifizierungsstellen. Erkauft wird dieser Vorteil durch die nicht mehr so triviale Implementierung und die Tatsache, dass im Gegensatz zum zentralisierten Ansatz, die hundertprozentige Authentizität des Kommunikationspartners nur in Ausnahmefällen festgestellt werden kann[57]. Im Folgenden soll das erwähnte Konzept näher erläutert werden.

4.4 Vertrauensnetz

Das Vorhandensein einer vertrauenswürdigen Stelle, an 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. 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[58]. Es ist vollkommen ausreichend, wenn einige uns bereits bekannte Gäste

[57] z.B. Vertrauenspfade der Länge eins

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

die Identität dieser Person bestätigen, indem diese uns beispielsweise als ein/e alte/r 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 Schlüssel 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 sich diese. Dasselbe gilt für die Disjunktheit der Vertrauenspfade: je mehr disjunkte Pfade vom Agenten1 zum Agenten2 führen, desto vertrauenswürdiger ist dieser. Abbildung 10 veranschaulicht das Prinzip der Vertrauenspfade.

Die Einstufung der Vertrauenswürdigkeit eines Agenten kann z.B. nach dem in der Tabelle 3 dargestellten Schema erfolgen[59].

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

Tabelle 3: Vertrauensbeziehungen innerhalb des Agentennetzes.

bestätigt.

[59] Vgl. a. [FEISTHAMMEL 2002]

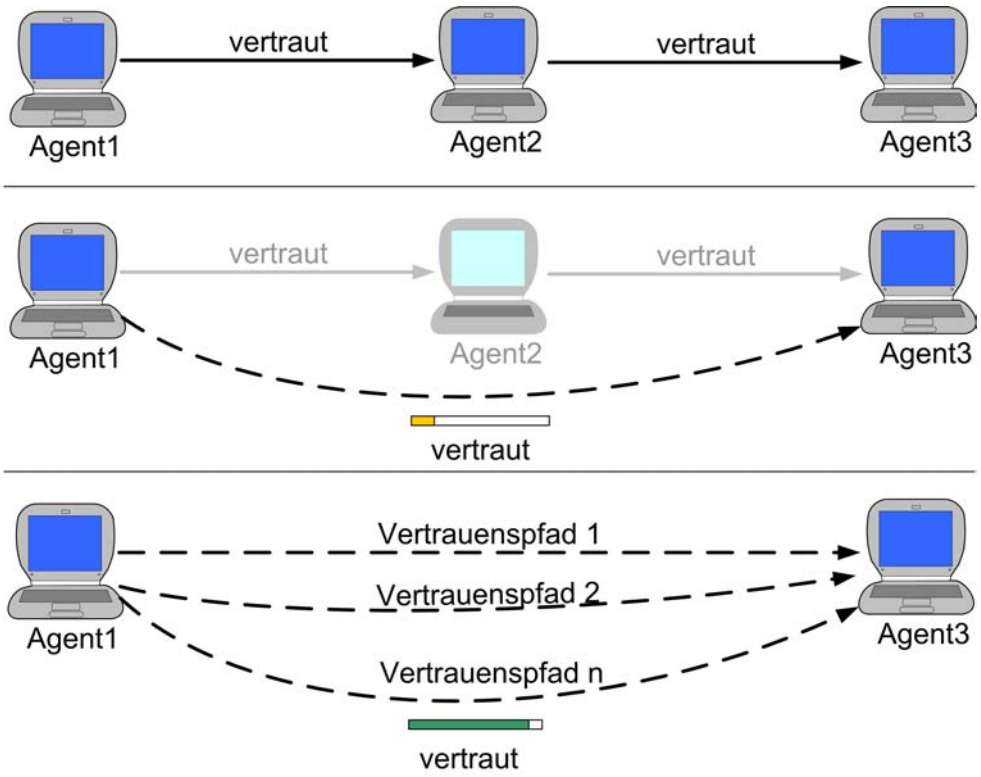

Abbildung 10: Vertrauensbeziehung zwischen Agent1 und Agent3.

4.5 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 besprochenen Konzepte noch keinen Einsatz in der aktuellen Antivirensoftware finden, werden einige von ihnen bereits seit geraumer Zeit in anderen Bereichen der Informationstechnologie eingesetzt, so dass deren Einsatz in einem produktiven Umfeld zwecks Angriffserkennung durchaus denkbar ist.

5 Mathematische Grundlagen

5.1 Der Einstieg

Nachdem in den vorhergehenden Kapiteln die Nachteile des klassischen Ansatzes zur Angriffserkennung ausreichend diskutiert und die Vorteile der im Rahmen dieser Arbeit entworfenen Lösung dargestellt wurden, sollen nun die beschriebenen positiven Eigenschaften der agentenbasierten Lösung mit Hilfe einiger Kalkulationen untermauert werden. Um eine sinnvolle Berechnungsbasis zu schaffen wird hier ein mathematisches Modell entworfen, welches es ermöglicht, die Güte einzelner Vorgehensweisen in Zahlen auszudrücken. Die auf dieser, zugegebenermaßen sehr theoretischen, Basis errechneten Ergebnisse werden anschließend mit Hilfe der während statistischer Untersuchungen erzeugten Daten verifiziert. Diese Daten wurden im Rahmen mehrerer Testläufe mit unterschiedlicher Parametrisierung gewonnen und berücksichtigen die Individualität des Agentenverhaltens.

5.2 Berechnungsgrundlagen

Um eine einheitliche Auswertung der Ergebnisse zu ermöglichen, wird an dieser Stelle Folgendes vorausgesetzt:

- Das Agentennetz besteht aus dreißig (30) Knoten.
- Jedes System im Testfeld wird von (genau) einem Agenten geschützt.
- Jeder Agent besitzt zehn (10) Ressourceneinheiten[60].
- Diese Ressourceneinheiten können vom Agenten mit Angriffssignaturen belegt werden.
- Das globale Ziel des Agentensystems besteht darin, einen möglichst guten

[60] Die Einteilung wurde nach dem Vorbild von prozentuellen Angaben vorgenommen. Benutzung einer Einheit entspricht dabei einer zehnprozentigen Auslastung des Systems, Verwendung von zwei Einheiten - zwanzigprozentigen Systemauslastung etc..

Schutz der Systeme mit einem möglichst kleinem Signatursatz zu gewährleisten[61].

- Es existieren genau zehn (10) Angriffe gegen ein System.
- Für jeden Angriff existiert genau eine Angriffssignatur. Ein Angriff kann also anhand seiner Signatur eindeutig identifiziert werden.
- Ein Angreifer kennt genau einen (1) dieser Angriffe und führt den ihm bekannten Angriff gegen jeden Knoten im Netz (dreißig mal) aus[62].
- Das Ziel des Angreifers besteht darin, möglichst viele Systeme im Netz zu kompromittieren.

Die im Folgenden verwendeten Ansätze der Wahrscheinlichkeitsrechnung sollen im Rahmen dieser Arbeit nicht näher erläutert werden, da sie deren Umfang sprengen würden. An dieser Stelle sei auf die weiterführende Literatur verwiesen[63].

5.2.1 Kalkulation auf Basis des klassischen Ansatzes

Die Aufbaubeschreibung des klassischen Ansatzes ist ausführlich im Abschnitt 3.7 (Der klassische Aufbau einer Sicherheitsinfrastruktur) beschrieben worden. Danach kommunizieren die einzelnen Agenten nicht miteinander und benutzen den

[61] Da es nur bedingt möglich ist, den Schutz eines Systems und die Menge verbrauchter Ressourcen miteinander in Beziehung zu setzen, wird hier die Güte einzelner Verfahren miteinander verglichen. Maßgebend ist dabei die Anzahl kompromittierter Systeme, welche unter der Verwendung eines der Ansätze mit einer bestimmten Anzahl von Signaturen zu erwarten ist.

[62] Ein dem Angreifer bekannter Angriff entspricht dabei einem *Exploit*. Ein Hacker kennt in der Regel nur einen Teil des für ein System existierenden *Expolits* und versucht den ihm bekannten Satz von *Exploits* an möglichst vielen System auszuprobieren, um diese zu kompromittieren. Ein System kann entweder die entsprechende Sicherheitslücke aufweisen oder nicht. Daraus ergibt sich die für den Angreifer maximal sinnvolle Anzahl von Angriffen, die der Anzahl von Systemen im Netz entspricht.

[63] Vgl. a. [LIPSCHUTZ 1989], [BASLER 1989], [ALTHOFF 1985]

gleichen Signatursatz. Die Berechnung der entsprechenden Wahrscheinlichkeit für einen erfolgreichen Angriff gestaltet sich trivial:

> Verwendet jeder Agent je n Signaturen, so beträgt die Wahrscheinlichkeit, dass der Angreifer eine richtige Signatur benutzt und das System kompromittiert $\frac{10-n}{10}$. Würde es dem Angreifer gelingen, einen Angriff durchzuführen, dessen Signatur von den Agenten nicht benutzt wird, hätte er bei zehn Angriffen zehn Systeme kompromittieren können. Die Berechnung der Erwartungswerte erfolgt nach dem folgenden Schema: $\mu = E[X] := \sum x_i p_i$, wobei x_i die Rolle einer diskreten zufälligen Variablen spielt (Anzahl kompromittierter Systeme) und p_i die entsprechende Wahrscheinlichkeit darstellt. Daraus ergibt sich die in der Tabelle 4 dargestellte Kalkulation.

Da sämtliche Systeme nach dem Prinzip des im vorhergehenden Kapitel erläuterten klassischen Modells funktionieren, haben sie alle den gleichen Signatursatz. Dies bedeutet, dass die aufgeführten Erwartungswerte sich erst bei der Berechnung des durchschnittlichen Erfolgs nach mehreren Durchläufen ergeben. Bei der Durchführung von dreißig Angriffen (eine Runde) wären entweder alle dreißig oder keins der Systeme kompromittiert. Würden die Agenten mit Signaturen arbeiten, die sich voneinander unterscheiden (Streuung des Signatursatzes), so würden die Erwartungswerte auch in diesem Fall den Werten in der Tabelle 4 entsprechen. Denn die Wahrscheinlichkeit eines erfolgreichen Angriffs ist in jedem Einzelfall lediglich von der Größe des Signatursatzes abhängig[65] und bleibt in jedem Einzelfall bei einer gegebenen Größe des Signatursatzes konstant.

5.2.2 Kalkulation auf Basis des agentenbasierten Ansatzes

Im Folgenden werden die Vorteile des idealisierten agentenbasierten Ansatzes besprochen. Der grundlegende Unterschied zur klassischen Vorgehensweise besteht

[64] Erwartete Anzahl kompromittierter Systeme nach dreißig Angriffen.

[65] Unter der Annahme, dass der Angriff mit einer bestimmten Signatur gleich wahrscheinlich ist.

Anzahl Signaturen (n)	**Wahrscheinlichkeit eines erfolgreichen Angriffs (p)**	**Erwartungswert (x)**[64]
0	1	30
1	$\frac{9}{10}$	27
2	$\frac{8}{10}$	24
3	$\frac{7}{10}$	21
4	$\frac{6}{10}$	18
5	$\frac{5}{10}$	15
6	$\frac{4}{10}$	12
7	$\frac{3}{10}$	9
8	$\frac{2}{10}$	6
9	$\frac{1}{10}$	3
10	0	0

Tabelle 4: Wahrscheinlichkeit eines erfolgreichen Angriffs in Abhängigkeit von der Anzahl verwendeter Signaturen.

in der Fähigkeit von Agenten, miteinander zu kommunizieren. Beim Auftreten einer bestimmten Gefahr sind Agenten in der Lage, sich gegenseitig zu warnen. Im idealen Fall erreicht diese Warnmeldung sämtliche Agenten, die von der Gefahr betroffen werden könnten, so dass sie rechtzeitig ihre Regelwerke aktualisieren und den Angriff erfolgreich abwehren können.

Die Kalkulation von Erwartungswerten ist an dieser Stelle ebenfalls trivial, auch wenn etwas komplizierter als beim klassischen Ansatz. Mehrere Angriffe können dabei am einfachsten als eine mehrstufige Versuchsreihe beschrieben werden[66]. Sehr interessant ist dabei die Frage, ob die Erfolgswahrscheinlichkeit eines Ereignisses (ein erfolgreicher Angriff) abhängig von den vorhergehenden Angriffen ist. Tatsächlich ist es so, dass ein Angreifer in seinem aktuellen Angriff nur

[66] Vgl. a. [SACHS 2003] S. 83 ff.

dann erfolgreich sein kann, wenn er beim vorhergehenden Angriff ebenfalls Erfolg verzeichnen konnte. Denn in unserem idealen Fall hat ein Angreifer keine Chance mehr, ein System zu kompromittieren, sobald ein einziger seiner Angriffe von einem Agenten erkannt wurde. Tatsächlich ist es so, dass in diesem Fall lediglich die Länge der Versuchsreihe variiert[67]. Dieser Versuch kann mit dem „Ziehen mit Zurücklegen“[68] verglichen werden. Maßgebend für die Einteilung ist dabei die Erfolgswahrscheinlichkeit bei jedem Angriff, denn diese bleibt innerhalb der Versuchsreihe (die eine variable Länge aufweist) konstant. Die Wahrscheinlichkeit, x Systeme zu kompromittieren, errechnet sich in unserem Modell wie folgt: $p_x = \left(\frac{10-n}{10}\right)^x$, wobei n die Anzahl der Signatursätze eines Agenten ist[69]. Etwas eindeutiger ist die Einordnung der Versuchsreihe beim bereits beschriebenen klassischen Ansatz. Da mehrere Systeme in diesem Fall vollkommen unabhängig voneinander agieren, beeinflusst der Erfolg oder der Misserfolg eines Angriffs weder die Erfolgswahrscheinlichkeit der nachfolgenden Angriffe, noch die Länge der Versuchsreihe. Beim klassischen Ansatz ist die Erfolgswahrscheinlichkeit bei einer festen Größe des Signatursatzes konstant und bei einem dreißigstufigen Experiment gleich $p_x = \left(\frac{10-n}{10}\right)^{30}$.

Um den Leser an dieser Stelle nicht mit monotonen Berechnungstabellen zu langweilen und damit seinen Lesefluss zu stören, wurde die detaillierte Kalkulation der Wahrscheinlichkeiten von erfolgreichen Angriffen im Anhang 3 zusammengefasst. Anhand der dort aufgeführten Daten erkennt man deutlich, dass die Verwendung des agentenbasierten Ansatzes sogar bei einer geringen Anzahl von Angriffen zu einem wesentlich besseren Schutz der Systeme unter Verwendung einer kleinen Signaturanzahl beiträgt. Tatsächlich ist es so, das die Erhöhung der Anzahl von Angriffen die Vorteile des agentenbasierten Ansatzes noch mehr zum Ausdruck bringt. Verwendet beispielsweise jeder Agent eine einzige Signatur zur

[67] Sobald ein Agent den n-ten Angriff erkennt, ändert sich die Erfolgswahrscheinlichkeit für den Angreifer auf 0 (null) und die Versuchsreihe wird unterbrochen (da die Durchführung weiterer Angriffe für den Angreifer nicht mehr sinnvoll ist) und hat die Länge n.

[68] Vgl.a. [KELLER 2003] S. 45 ff.

[69] Die Variable x ist dabei die Länge der Versuchsreihe.

Erkennung von Angriffen und ist er zusätzlich in der Lage, eine weitere Signatur in seinen Datensatz aufzunehmen, ergibt sich die in der Tabelle 5 dargestellte Kalkulation[70].

Anzahl von Angriffen:	10	20	30	40	50	60	70	80	90	100
Anzahl kompromittierter Systeme:	5.51	7.78	8.58	8.85	8.95	8.98	8.99	8.99	8.99	8.99

Tabelle 5: Anzahl kompromittierter Systeme in Abhängigkeit von der Angriffsanzahl bei einem aus einer Signatur bestehenden Signatursatz.

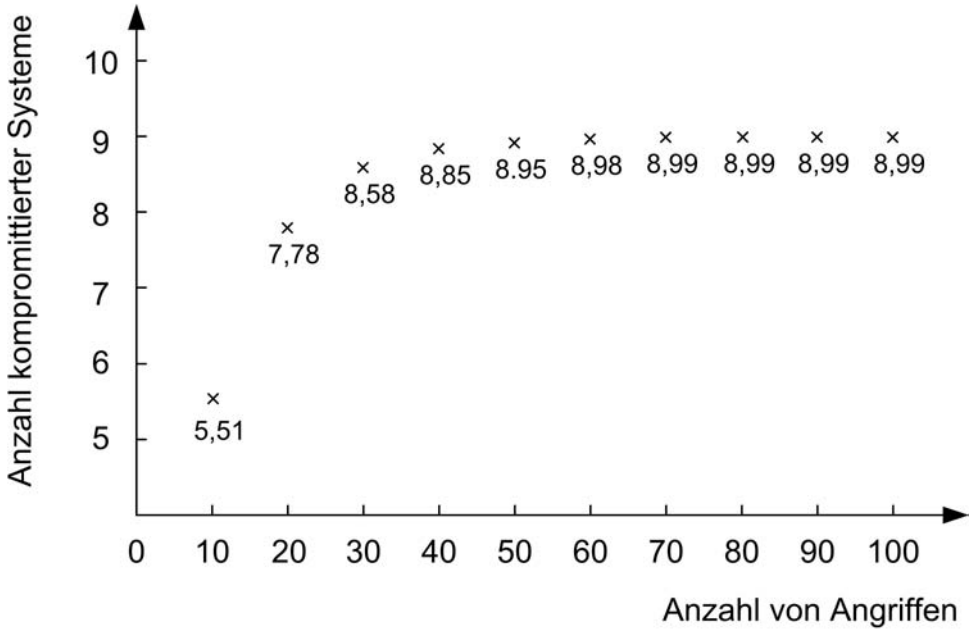

Abbildung 11: Anzahl kompromittierter Systeme in Abhängigkeit von der Angriffsanzahl bei einem aus einer Signatur bestehenden Signatursatz.

Ein Angriff auf ein Systemverbund aus hundert (100) Systemen mit jeweils neun (9) Signatursätzen hat beim klassischen Ansatz einen Erwartungswert von genau zehn (10) Systemen. Dieser Wert ist um eins höher als der des aufgeführten

[70] Tatsächlich ist es so, dass die Zahlenfolge gegen neun Systeme von unten konvergiert und das, obwohl jeder Agent lediglich eine Signatur in seinem Speicher hält. Die Genauigkeit der in der Tabelle 5 dargestellten Ergebnisse beträgt zwei wesentliche Ziffern.

Agentennetzes und erfordert etwa das Neunfache der Signaturgröße beim agentenbasierten Ansatz. Vergleicht man dagegen den Verlauf von Erkennungsraten des Agentennetzes und des klassischen Ansatzes mit jeweils einer Signatur ergibt sich das in der Abbildung 12 dargestellte Verhältnis. Die Abbildung 12 offenbart uns die Überlegenheit des agentenbasierten Ansatzes gegenüber dem klassischen besonders deutlich.

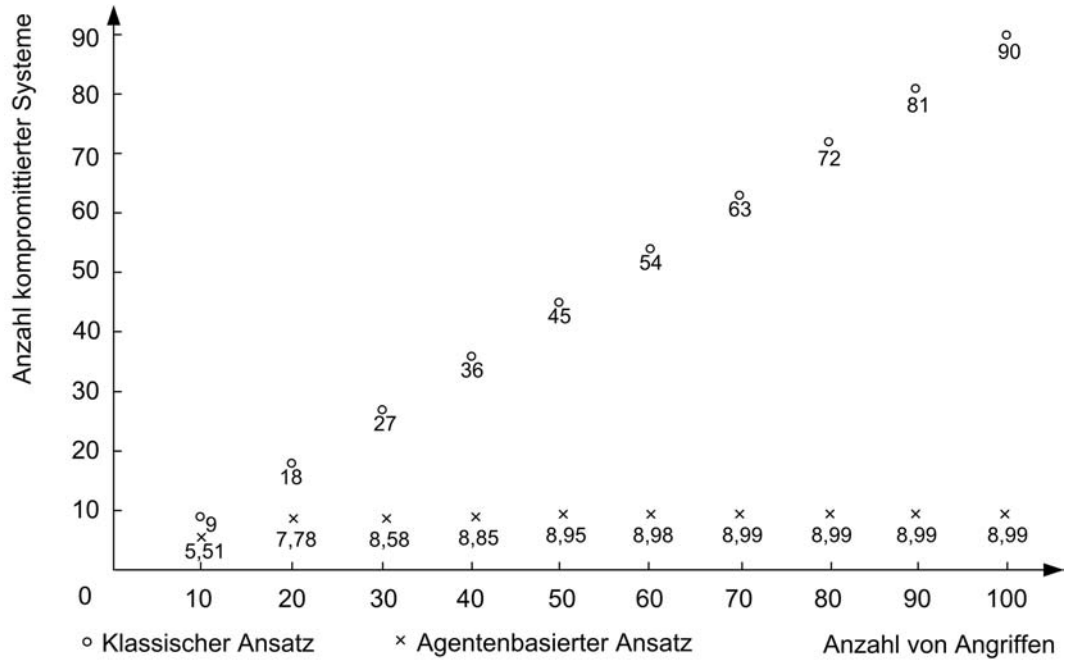

Abbildung 12: Vergleich der Erkennungsergebnisse des agentenbasierten und des klassischen Ansatzes bei einer Signatur.

Bei hundert (100) Hosts ergibt sich ein Verhältnis von $\frac{9}{100}$ zwischen den kompromittierten Systemen und deren Gesamtzahl. In einem aus zehntausend (10.000) Systemen bestehenden Unternehmensnetzwerk wäre das Verhältnis gleich $\frac{9}{10.000}$. Wenn man davon ausgeht, dass ein System, unter der Verwendung des heutzutage üblichen klassischen Ansatzes, durchschnittlich ein Mal im Jahr aufgrund eines Angriffes kompromittiert wird, wäre jeder unserer von Agenten geschützten Systeme etwa ein Mal in tausendeinhundertelf (1.111) Jahren kompromittiert. Die mit Hilfe der heutzutage üblichen Vorgehensweise zu erzielenden Ergebnisse wären also etwa tausend Mal schlechter.

In unseren Berechnungen sind wir von dem Idealfall ausgegangen, dass jeder Agent alle anderen Agenten immer warnen kann. In einer realen Umgebung ist das leider nicht möglich[71]. Wir könnten diesen Sachverhalt als Basis für ein weiteres Modell nehmen, bei dem jede Warnung einen Agenten nur mit einer Wahrscheinlichkeit unter 100 % erreichen kann. Für dieses Modell des Agentennetzes sollen dieselben Forderungen gelten, wie für die bereits beschriebenen Modelle. Um die Rechenschritte besser nachvollziehen zu können, betrachten wir zuerst ein Zahlenbeispiel:

- Jeder Agent wird mit 5 (fünf) Signaturen initialisiert.
- Beim Feststellen eines Angriffs sendet jeder Agent eine Warnmeldung an die Mitglieder des Netzes, die im Durchschnitt jedoch nur 10% aller Agenten erreicht.

Der beschriebene Sachverhalt wird in der Abbildung 13 mit Hilfe eines Entscheidungsbaumes veranschaulicht[72].

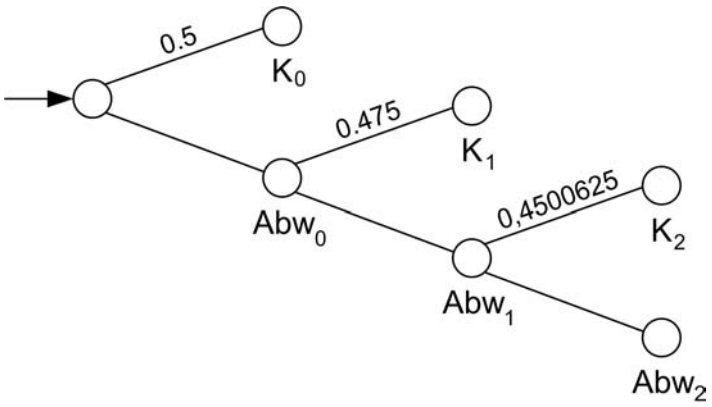

Abbildung 13: Jede Warnmeldung erreicht einen bestimmten Prozentsatz von Agenten.

K_i ist dabei die Wahrscheinlichkeit, dass ein System durch den Angriff i kompromittiert wird und Abw_i - die Wahrscheinlichkeit, dass der Angriff abgewehrt

[71] So kann z.B. eine Warnung mit einer kleinen Hop-Zahl verfallen, bevor sie ein System erreicht oder der Angreifer kann einen neuen Angriff durchführen, obwohl die Warnmeldung noch nicht von allen Agenten empfangen wurde.

[72] Vgl. a. [SACHS 2003]

werden kann. Bei K_0 und Abw_0 bedarf es keiner Berücksichtigung der Empfangswahrscheinlichkeit, denn vor dem ersten Angriff erhält noch kein System eine Warnmeldung. Die Ausgangswahrscheinlichkeit $P_0 = 0,5$ kann für K_0 übernommen werden. Beim zweiten Angriff (K_1 und Abw_1) kann das System genau dann kompromittiert, wenn:

1. Der Agent die Signatur des Angriffs nicht kennt und
2. der Agent noch keine Angriffswarnungen erhalten hat.

Im Fall K_1 erhält der Agent keine Warnung, wenn das erste Angriffsopfer kompromittiert wurde (K_0) oder der Angriff abgewehrt werden konnte, die Angriffswarnung beim zweiten Agenten jedoch nicht angekommen ist ($P(Abw_0) \cdot 0,9$)[73]. $P(Abw_1)$ ist dabei die Gegenwahrscheinlichkeit eines erfolgreichen Angriffs und errechnet sich wie folgt: $1 - P(K_1)$. Analog erhält man die Werte für ein beliebiges K_i die sich für den von uns besprochenen Fall mit der folgenden Formel berechnen lassen:

$$P(K_{i+1}) = (1 - 0,5) \cdot \prod(P(K_i) + P(Abw_i) \cdot 0,9)$$

In unserem konkretem Beispiel tritt K_0 mit einer Wahrscheinlichkeit von 0,5 (50%) ein. Um die Eintrittswahrscheinlichkeit für K_1 zu berechnen, müssen wir die von uns gewählten Parameter (0,5 und 0,1) in die Formel einsetzen:

$$P(K_1) = (1-0,5)\cdot(P(K_0)+P(Abw_0)\cdot(1-0,1)) = 0,5\cdot(0,5+0,5\cdot0,9) = 0,475 \quad (47,5\%)$$

Analog dazu errechnet sich die Eintrittswahrscheinlichkeit für K_2 ($P(K_2)$):

$$P(K_2) = (1-0,5)\cdot((P(K_0)+P(Abw_0)\cdot(1-0,1))\cdot(P(K_1)+P(Abw_1)\cdot(1-0,1)) =$$

$$= 0,5 \cdot ((0,5 + 0,5 \cdot 0,9) \cdot (0,475 + 0,525 \cdot 0,9)) = 0,4500625 \quad (45,00625\%)$$

[73] 0,9 = (1 - 0,1) ist dabei die Wahrscheinlichkeit, dass eine Warnmeldung den Agenten nicht erreicht.

Anhand dieses Zahlenbeispiels wird der rekursive Charakter der Formel sehr gut erkennbar. Ersetzen wir nun die konkreten Zahlen in unserem Zahlenbeispiel durch allgemeine Symbole, erhalten wird die generische Formel für die Berechnung der Erfolgswahrscheinlichkeit des i-ten Angriffs. Diese Wahrscheinlichkeit ist von drei Variablen abhängig: Signaturenanzahl, Erfolgswahrscheinlichkeit einer Warnungsweiterleitung und der laufenden Nummer des Angriffs und lautet:

$$P(K_{i+1}) = (1 - P(Abw)) \cdot \prod (P(K_i) + P(Abw_i) \cdot (1 - p_w))^{74}$$

Die Kompromittierungswahrscheinlichkeiten für einige Zahlenbeispiele sind dem Anhang 4 zu entnehmen. In der Abbildung 14 wird der Vergleich zwischen dem klassischen Ansatz unter Verwendung von 7 (sieben) Signaturen und dem agentenbasierten Ansatz unter Verwendung von 1 (einer) und 5 (fünf) Signaturen veranschaulicht. Die Weiterleitungswahrscheinlichkeit für eine Warnmeldung beträgt dabei 10 (zehn) Prozent. Trotz der relativ geringen Wahrscheinlichkeit für den erfolgreichen Empfang einer Warnmeldung hat das Agentennetz mit lediglich einer Signatur nach 30 (dreißig) Angriffen fast dieselben Erkennungsraten wie das klassische System mit (7) sieben Signaturen. Bei 5 (fünf) Signaturen und der gleichen Weiterleitungswahrscheinlichkeit von 10 (zehn) Prozent ist das Agentennetz bereits nach 10 (zehn) Angriffen sicherer als das klassische Modell mit 7 Signaturen. Beim dreißigsten Angriff ist das Agentennetz gar etwa sechs Mal performanter als das klassische System mit einem um zwei Einheiten größeren Signatursatz.

Die Abbildung 15 veranschaulicht die Wahrscheinlichkeiten für die Kompromittierung eines durch einen Agenten geschützten Systems beim n-ten Angriff. In allen drei Fällen besteht der Signatursatz von Agenten aus einer einzigen Signatur. Variable Größe ist in diesem Fall die Weiterleitungswahrscheinlichkeit der Angriffswahrnung, die hier jeweils 10% (zehn), 20% (zwanzig) und 30% (dreißig) beträgt. Durch das □-Symbol gekennzeichneten Werte spiegeln die Performance

[74] p_w ist dabei die Wahrscheinlichkeit einer erfolgreichen Weiterleitung; P(Abw) ist die Abwehrwahrscheinlichkeit des ersten Agriffs.

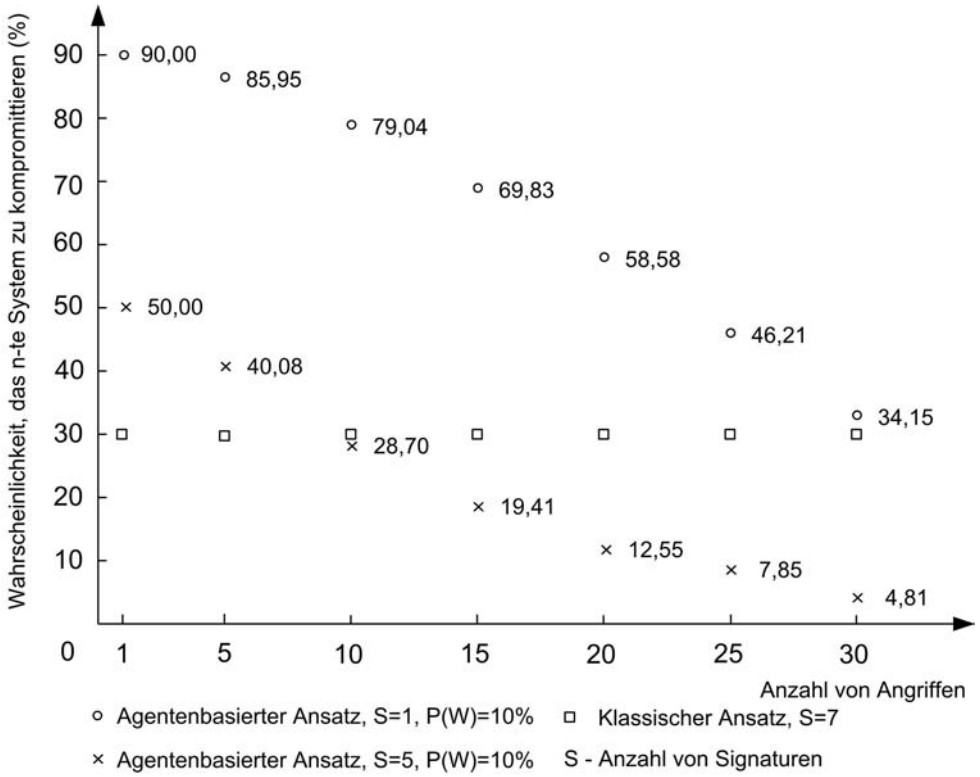

Abbildung 14: Erkennungsraten des Agentennetzes bei einer zehnprozentigen Empfangswahrscheinlichkeit.

des klassischen Ansatzes bei sieben Signaturen wider[75]. Erwartungsgemäß verbessern sich die Erkennungsraten des Agentennetzes überproportional zur Steigerung der Empfangswahrscheinlichkeit. So ist beispielsweise ein Mitglied des Agentennetzes nach 30 (dreißig) Angriffen bei einer Weiterleitungswahrscheinlichkeit von 30% (dreißig) etwa 3 (drei) mal besser geschützt als ein Mitglied des gleichen Netzes bei einer Weiterleitungswahrscheinlichkeit von 20 (zwanzig) Prozent.

Eine Auflistung der zu erwartenden Performance-Ergebnisse des Agentennetzes bei einer variablen Weiterleitungswahrscheinlichkeit ist dem Anhang 4 zu entnehmen. Es muss allerdings an dieser Stelle gesagt werden, dass die anhand des beschriebenen Modells errechneten Werte nicht der realistischen Gegebenheiten entsprechen. Diese Erkenntnis resultiert aus der Tatsache, dass sowohl die Angreifer als auch die Guardian-Agenten dieselbe Infrastruktur benutzen. Im Kapitel „Agentenbasierter Schutz“ wurde bereits erwähnt, dass die *Malware*-Angriffe

[75]Dieser Wert soll hier als Vergleich dienen.

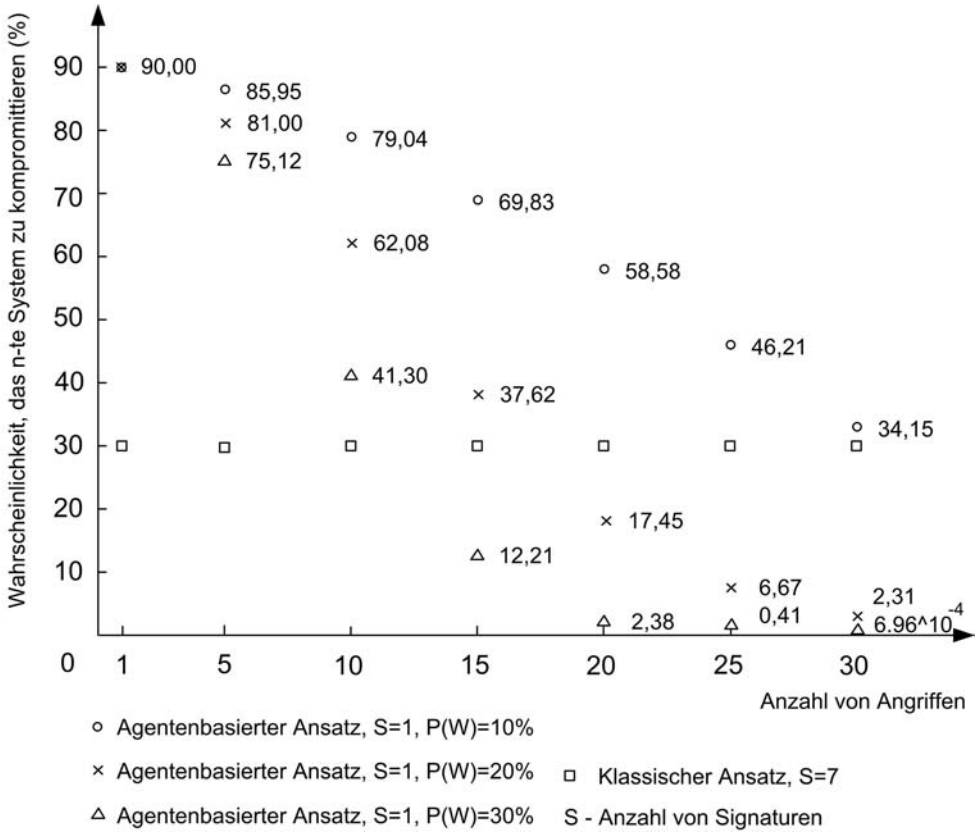

Abbildung 15: Erkennungsraten des Agentennetzes bei einer variablen Empfangswahrscheinlichkeit.

nicht wahllos stattfinden und sich meistens zuerst in geschlossenen/benachbarten Strukturen abspielen[76]. Die Tatsache, dass eine Angriffswarnung einen bestimmten Agenten nicht erreicht, ist unmittelbar mit der Wahrscheinlichkeit eines Angriffs auf diesen Agenten verbunden. Denn je schlechter ein Agent von einem Netzsegement zu erreichen ist, desto geringer ist die Wahrscheinlichkeit, dass er aus diesem oder einem der benachbarten Netzsegmente attackiert wird. Mir erscheint es zu diesem Zeitpunkt unmöglich, eine mathematisch korrekte Beziehung zwischen der Erreichbarkeit eines Agenten und der Wahrscheinlichkeit des gegen ihn durchgeführten Angriffs herzustellen[77].

5.3 Performancedaten des simulierten Agentennetzes

Natürlich handelt es sich bei den soeben vorgestellten Berechnungen um idealisierte Fälle. Tatsächlich ist es so, dass die errechneten Werte von einer Implementierung nicht erreicht werden können. Dies liegt vor allem daran, dass die Agenten per Definition keine homogene Struktur darstellen und im Auftrag ihrer (zum Teil unterschiedlichen) Inhaber handeln, was aus den unterschiedlichen Zielsetzungen der Agenten resultiert.

Im Rahmen dieser Arbeit wurde die Basisstruktur des Agentennetzes implementiert. Anschließend wurde ein Agentennetz aus 30 (dreißig) Nodes erzeugt,

[76] Besonders interessant wird dieser Gedankengang dann, wenn man berücksichtigt, dass die Netzstrukturen in der Realität inhomogen sind, was z.B. durch die Leistungsfähigkeit einzelner Systeme bedingt ist. In *GNUtella*-Netzen versucht man durch die Lastverlagerung auf besonders leistungsstarke Netzteilnehmer (Ultrapeers) die angestrebte übergeordnete Infrastrukturhomogenität herzustellen. Vgl. a. [SINGLA 2001]. Die Aussagekraft der Weiterleitungswahrscheinlichkeit ist in einer solchen inhomogenen Struktur mehr als fragwürdig.

[77] Da der Routing in der Regel von Datenpaketen dynamisch erfolgt, kann die Erreichbarkeit eines Systems nicht als eine konstante Größe behandelt werden. So können beispielsweise zwei Datenpakete über zwei vollkommen unterschiedliche Wege zu einem System gelangen. Berücksichtigt man zusätzlich, dass ein modernes System in der Regel mehrere Schnittstellen zu der Außenwelt besitzt (z.B. Ethernet-Anbindung, *Bluetooth*, WLAN etc.) hat man keine Chance mehr, dieses komplexes Gebilde in einem Modell zusammenzufassen.

das als Basis für statistische Untersuchungen diente. Die während dieser Untersuchungen erzeugten Daten können dem Anhang 1 entnommen werden[78]. Im Folgenden soll nun deren Auswertung im Vergleich mit den theoretisch erzielbaren Ergebnissen erfolgen. Dabei muss hier erneut gesagt werden, dass die durch die Implementierung erzielten Ergebnisse sich lediglich auf eine bestimmte während der Tests festgelegte Agentennetz- und Verhaltensstruktur der Agenten beziehen.

In der aktuellen Implementierung wurden Agenten mit individuellen Verhaltensmustern ausgestattet, die in die Konfigurationsdatei des entsprechenden Agenten eingetragen wurden. Eine Beispiel-Konfigurationsdatei eines Agenten ist dem Anhang 2 zu entnehmen. Mit Hilfe der in dieser Datei enthaltenen Parameter, ist es möglich, einem Agenten, bestimmte Verhaltensmerkmale zuzuordnen. Abhängig von der Parametrisierung kann ein Agent die Warnmeldungen nur an bestimmte Agenten weiterleiten, von denen er bereits Warnungen erhalten hat, er kann das Weiterleiten von Warnungen komplett verweigern oder dies nach dem Zufallsprinzip tun. Ein Agent kann jedoch auch so konfiguriert werden, dass er jede bei ihm eingetroffene Warnmeldung weiterleitet. Ebenfalls individuell kann das Verhalten eines Agenten beim Eintreffen einer Warnmeldung sein. Ein Agent kann entweder diese verwerfen oder seine Signaturdatenbank nach einer bestimmten Anzahl von Warnungen eines Typs updaten, was ebenfalls seiner Konfigurationsdatei zu entnehmen ist.

5.4 Aufbau der Testreihe

Die statistischen Daten wurden aus einem Netzwerk von dreißig (30) Agenten gesammelt. Fünf dieser Agenten leiteten die Warnmeldungen nach dem Zufallsprinzip an andere Agenten weiter; jeder Agent besaß eine bestimmte Anzahl von Knoten in seiner Kontaktliste; die Mitglieder dieser Liste wurden nach dem Zufallsprinzip mit einer Wahrscheinlichkeit von 50 % ausgesucht und mit der ent-

[78] Die im Anhang 1 dargestellten Ergebnisse wurden nach der vierten wesentlichen Nachkommastelle abgeschnitten.

sprechenden Warnmeldung versorgt. Fünf weitere Mitglieder des Netzes leiteten die Warnmeldungen nur an diejenigen Agenten weiter, von denen sie zu diesem Zeitpunkt bereits Warnungen erhalten haben. Dies bedeutete, dass solche Agenten besonders am Anfang einer Testreihe keine Warnungen weitergeleitet haben und somit zu einem schlechteren Ergebnis beigetragen haben. Jeder Agent wurde mit einer aus zwei Einträgen bestehenden Kontaktliste initialisiert. Mit Hilfe eines Werkzeugs, welches im Rahmen der Implementierung erstellt wurde[79], wurde anschliessend geprüft, ob der aus dem Agentennetz erzeugte Graph zusammenhängend ist. Wenn ein Mitglied des Agentennetzes keine Verbindung zu den Agenten aus seiner Kontaktliste aufnehmen konnte[80], unternahm er weitere Verbindungsversuche, bis eine Verbindung zum Agentennetz hergestellt werden konnte. Die durchschnittliche Größe der Kontaktliste eines Agenten erwies sich parallel zu der Größe des Signatursatzes als maßgebend für die Performance des Agentennetzes. Dies lag vor allem daran, dass ein Agentennetz mit einer hohen durchschnittlichen Größe der Kontaktliste[81] die entsprechenden Warnmeldungen an wesentlich mehr Agenten weiterleiten konnte. Es hat sich auch gezeigt, dass Agentennetze, die sich durch eine geringe mittlere quadratische Abweichung der Kontaklist-Größen auszeichnen, bessere Erkennungwerte liefern als Netze mit derselben durchschnittlichen Größe der Kontaklisten und größeren Unterschieden in den Kontaklistgrößen. Dies ist zum Beispiel an den Erkennungsraten des Agentennetzwerks mit acht und neun Signaturen deutlich zu sehen[82]. Das Agentennetz mit neun Signaturen (Durchscnittswert: 0,6 kompromittierte Systeme) weist eine etwas schlechtere Erkennungsrate als ein Netzwerk mit acht Signaturen (Durchscnittswert: 0,4 kompromittierte Systeme) auf, was durch eine kleinere Durchschnittsgröße der Konstaktliste und eine etwas höhere Standardabweichung der Listengrößen zu erklären ist. Das Netzwerk erlaubte sich einen einzigen „Ausrut-

[79] Dazu mehr im Kapitel „Implementierungsdetails“.

[80] Dies war z.B. dann der Fall, wenn die Kontaktaufnahme zu einem Zeitpunkt erfolgte, als noch keines der Kontaklistenmitglieder im Netz angemeldet war.

[81] Die Informationen zu den Größen entsprechender Kontaktlisten sind dem Anhang 1 zu entnehmen.

[82] Siehe Anhang 3.

scher“ im Durchlauf elf (11)[83]. Um ein solches unvorhergesehenes Verhalten des Agentennetzes zu vermeiden, soll man versuchen, dessen Struktur möglichst homogen zu halten. Man könnte z. B. versuchen, die prozentuellen Anteile von Agenten mit unterschiedlichem Verhaltenweisen in allen Netzsegmenten etwa gleich zu halten oder die Anzahl von Hops, mit denen eine Meldung initialisiert wird, an die Gegebenheiten des Netzes anzupassen[84].

In der Abbildung 16 werden die Performance-Ergebnisse verschiedener Vorgehensweisen miteinander verglichen. Die Anzahl kompromittierter Systeme bei der Verwendung des klassischen Ansatzes wurde dem Gedankenexperiment entnommen und ist durch das $\times$-Symbol gekennzeichnet. Die Leistung des implementierten Agentennetzes wird durch die mit dem $\square$-Symbol gekennzeichneten Werte dargestellt. Sie ist trotz der zum Teil sehr ungünstigen Beschaffenheit des Netzes[85] wesentlich höher als die des klassischen Modells. Unter Verwendung von mehr als zwei Signaturen entspricht die Leistung des implementierten Agentennetzes weitestgehend dem theoretisch maximal erreichbaren Wert[86].

Die durch das Gedankenexperiment ermittelten Ergebnisse sind die Performance-Obergrenze für ein Agentennetz, das den am Anfang des Kapitels getroffenen

[83] Siehe Anhang 1 - Ergebnisse der statistischen Untersuchungen bei neun (9) Signaturen.

[84] Bei den Tests wurden alle Meldungen pauschal mit einem Wert von drei möglichen Hops initialisiert. Bei einer durschnittlichen Größe der Kontakliste von neun (9) Hosts braucht eine Meldung mindestens 2 (zwei) Hops, um das Netz aus dreißig Agenten zu erreichen. Da die Kontaktlisten verschiedener Agenten zum Teil redundant sind, war diese Zahl nicht immer ausreichend, um alle Mitglieder des Netzes mit Meldungen zu versorgen. Während der Testläufe wurde die Anzahl von Hops bewusst klein gehalten, um die Datenlast in Grenzen zu halten. Die Erhöhung dieser Zahl würde bessere Erkennungsraten mit sich bringen.

[85] Ein sehr hoher prozentueller Anteil an Hosts, die Warnmeldungen nicht weiterleiten oder dies nicht regelmäßig tun, eine geringe Anzahl von Hosts in den Kontaklisten und schließlich eine sehr kurze Testreihe von dreißig (30) Angriffen sorgen für die Ergebnisse, die im Alltagsbetrieb unter schlechten Umständen zu erwarten wären. Die Zusammenstellung der Testinfrastruktur wurde mit Absicht relativ ungünstig gehalten, damit die Ergebnisse den Realitätserwartungen entsprechen.

[86] Hier durch das $\bigcirc$-Symbol dargestellt.

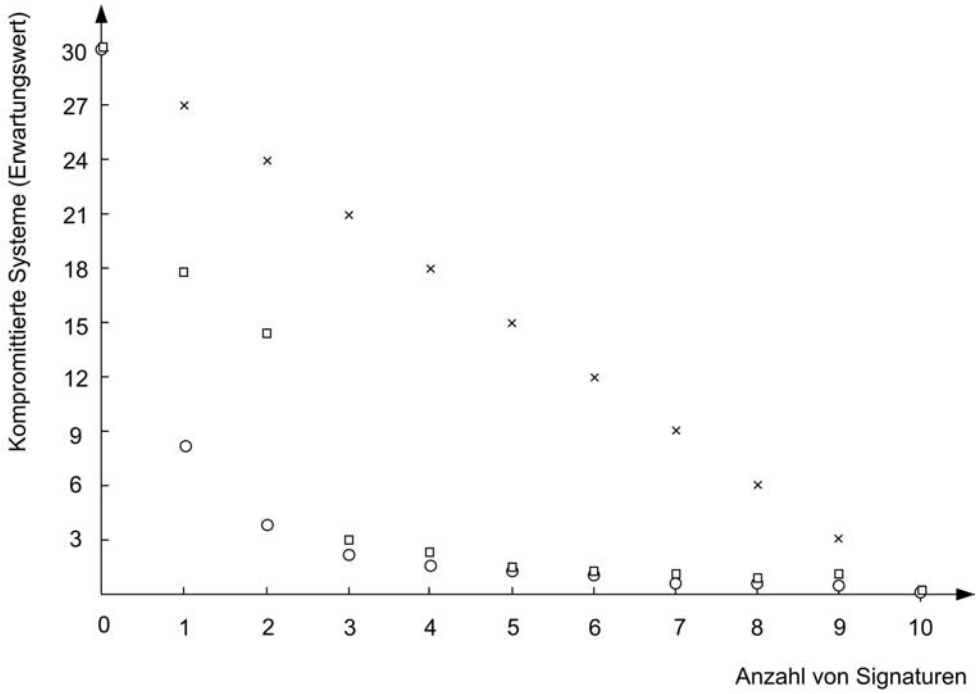

Abbildung 16: Vergleich von Performance-Ergebnissen unterschiedlicher Modelle.

Annahmen/Voraussetzungen entspricht. Dieser Wert, dessen mathematische Korrektheit bewiesen werden kann, wird von einer realen Implementierung niemals erreicht. Mit Hilfe der Test-Implementierung des Agentennetzes, welche die im Abschnitt 5.2 beschriebenen Berechnungsgrundlagen ebenfalls erfüllt, wurde ein Versuch unternommen, Leistungsfähigkeit des Agentennetzes unter realistischeren Bedingungen zu untersuchen. Die Beschaffenheit des Testnetzes wurde suboptimal gehalten[87], um von einer realen Implementierung zu erwartenden Ergebnisse zu erzeugen.

5.5 Zusammenfassung

In diesem Kapitel wurden die grundlegenden statistischen Berechnungen durchgeführt, die die Überlegenheit des agentenbasierten Ansatzes gegenüber dem klassischen Ansatz bewiesen. Im folgenden Kapitel werden einige Details über den

[87] Nicht jede Angriffswarnung erreichte alle Agenten, die ihrerseits nicht immer solche Warnmeldungen versendeten oder weiterleiteten.

Aufbau und die Funktionsweise des Agentennetzes erläutert. Außerdem werden die Grundlagen zur Steuerung des Netzes anhand von einigen Beispielen beschrieben.

6 Implementierungsdetails

6.1 Framework

Die im Rahmen dieser Arbeit stattgefundene Implementierung fand auf Basis des *JADE*-Frameworks statt und erweiterte dessen Basisfunktionalität durch einige Features. Die Entscheidung, dieses Framework zu benutzen wurde zu keinem Zeitpunkt der Implementierung bereut, zumal die Erfahrungen mit zwei anderen Frameworks, die am Anfang der Implementierung gesammelt wurden alles andere als positiv waren. *JADE*-Framework bietet außer einer, leicht verständlichen und ausführlichen Dokumentation, einen sehr breiten Funktionsumfang, der es ermöglicht, Fehler in der Implementierung sehr effizient zu beseitigen. Als besonders angenehm erwies sich in der Implementierungsphase die große und hilfsbereite *JADE*-Entwicklergemeinde.

Das *JADE*-Framework verfügt über eine Funktionalität, die es leicht möglich macht, die eingebauten Features in einer Multiagentenumgebung zu testen. Als sehr hilfreich erwies sich die graphische Oberfläche der Umgebung, die intuitiv zu bedienen war. Mit Hilfe von wenigen Handgriffen war es möglich, Sniffer-Agenten zu erzeugen, die die Überwachung des Datenverkehrs innerhalb des Agentennetzes sehr einfach gestaltet haben. Besonders interessant ist die Möglichkeit des Frameworks, die Infrastruktur des Agentennetzes auf mehrere physikalische Maschinen zu verteilen. Dadurch ist es möglich, auch statistische Untersuchungen mit einem größeren Agentennetz durchzuführen.

6.2 Verbindungsaufbau

Um sich mit dem Agentennetzwerk zu verbinden bedarf ein Agent lediglich der Adresse eines einzigen Agenten, der bereits Mitglied dieses Netzwerks ist. Der neue Agent schickt diesem eine Meldung (einen Ping), welche anschließend von dem Agenten an alle von ihm bekannten Agenten weitergeleitet wird. Die Agen-

ten nehmen den neuen Host in ihre Listen auf und verschicken ihm ihrerseits Meldungen (Pongs), die es ihm ermöglichen, sie in seine Kontaktliste aufzunehmen. Die Meldungen enthalten eine Liste mit Empfängern als Parameter, um das mehrfache Verschicken von Initialisierungsnachrichten zu vermeiden[88].

Die Weiterleitung von Warnmeldungen ist vom individuellen Verhalten einzelner Agenten abhängig. Ein Agent kann die Warnmeldung grundsätzlich an alle Agenten in seiner Kontaktliste versenden oder diese für sich behalten. Er kann jedoch auch nach dem Zufallsprinzip vorgehen und die Warnmeldungen zufällig an Agenten verschicken. Die aktuelle Implementierung erlaubt es außerdem, die Warnmeldungen lediglich an diejenigen Agenten zu versenden, die zu einem bestimmten Zeitpunkt bereits Warnmeldungen verschickt haben. Dieses Verhalten kann mit Hilfe der Parametrisierung erreicht werden und ist vom Parameterwert `warnotheragents` abhängig[89]. Um das mehrfache Berücksichtigen einer Meldung zu vermeiden, werden diese mit einem Timestamp (einer eindeutigen ID) versehen[90].

Alle im Agentennetz verschickten Meldungen werden mit einem Zähler versehen. Dieser Zähler sagt aus, wie oft ein Paket im Netz weitergeleitet werden kann. Bei jedem Hop wird der Zähler decrementiert. Beim Erreichen des Zählerstands Null wird die Meldung verworfen. Dadurch wird vermieden, dass die Meldungen unendlich lang im Netz kursieren. Wird eine Meldung mit einem zu hoch gesetzten Zähler (z.B. 1000) verschickt, kann ein Agent vor der Weiterleitung der Meldung ihren Zähler auf eine sinnvolle Größe reduzieren (z.B. 5 (fünf)).

[88] Eine ähnliche Vorgehensweise wird seit Jahren in *GNUtella*-Netzen erfolgreich benutzt. Vgl. a. [KNOWBUDDY 2004]

[89] Mehr Informationen zur Parametrisierung von Agenten entnehmen Sie bitte dem Anhang 2.

[90] Dies ist beispielsweise wichtig, wenn die Zusammenstellung des aktuellen Signatursatzes von der Häufigkeit der Warnmeldungen über einen bestimmten Angriff abhängig ist.

6.3 Kommunikation zwischen Agenten

Kommunikationsfähigkeit ist einer der Schlüsseleigenschaften eines Agenten. Ein gemeinsames Kommunikationsprotokoll ist deswegen für das Funktionieren des Agentennetzes von entscheidender Bedeutung. Die Nachrichtenheader der aktuellen Implementierung sind an den ACL[91]-Vorschlag von *FIPA* angelehnt[92]. Da nicht alle in der *FIPA*-Spezifikation beschriebenen Kommunikationsheader von der aktuellen Implementierung verwendet werden[93], wurde nur ein Teil der *FIPA*-Spezifikation übernommen.

Eine *FIPA*-konforme ACL-Message besteht aus mehreren Bestandteilen, die wiederum nicht alle in einer gegebenen Implementierung ihre Verwendung finden müssen. Aus Platzgründen werden in der Tabelle 6 nur die in der Implementierung verwendeten Message-Bestandteile aufgelistet. Mehr Informationen zu den hier nicht erwähnten Headern entnehmen Sie bitte der *FIPA*-Spezifikation[94]. Jede Nachricht besitzt außerdem einen bestimmten Typ. Die in der aktuellen Implementierung verwendeten Nachrichtentypen sind in der Tabelle 7 aufgelistet.

[91] Agent Communication Language
[92] Vgl. a. [TANENBAUM 2003] S. 206
[93] So unterstützen die Agenten in der aktuellen Implementierung keine Verhandlungen und können keine Ausschreibungen tätigen.
[94] Vgl. a. [FIPA 2004]
[95] S.a. REPLY_TO
[96] Bei dem er jedoch nicht eindeutig sicher ist, dass es sich dabei um einen Angriff handelt.

SENDER	Identifiziert den Absender der Nachricht, kann jedoch auch ausgelassen werden, wenn der Absender anonym bleiben will. Dies kann z.B. dann sinnvoll sein, wenn die Nachricht vom aktuellen Agenten lediglich weitergeleitet wird und er nicht der eigentliche Empfänger der zu erwartenden Antwort ist[95].
RECEIVER	Dieses Feld muss mit einem Wert belegt werden und identifiziert die Empfänger der Nachricht. Das in der Implementierung verwendete *JADE*-Framework erlaubt (*FIPA*-konforme) Multicast-Nachrichten, bei denen mehrere Agenten als Empfänger vorgesehen sind.
REPLY_TO	Bei der Belegung dieses Headers soll die Antwort auf die Nachricht nicht an den eigentlichen Absender, sondern an den/die in diesem Feld eingetragenen Agenten gesendet werden. Dies ist z.B. dann sinnvoll, wenn die Nachricht lediglich weitergeleitet wird.
CONTENT	In diesem Feld wird der eigentliche Inhalt einer Nachricht transportiert. Jeder Agent kann diesen auf seine individuelle Weise interpretieren.
CONVERSATION_ID	Wird in der Regel verwendet, um einen Nachrichtenaustausch zwischen mehreren Agenten zu ermöglichen, indem Nachrichten mit Identifikationsnummern versehen werden, damit ein Agent die von ihm erhaltene Nachricht einer eindeutigen Kommunikationssession zuordnen kann. In der gegebenen Implementierung enthält dieses Feld eine eindeutige Nachricht-ID, um eine mehrfache Auswertung einer Nachricht durch einen Agenten zu vermeiden.

Tabelle 6: Die in der aktuellen Implementierung verwendeten Nachrichtenheader.

6.4 Agentennetz

Die folgende kurze Anleitung zur Steuerung des Agentennetzes erhebt keinerlei Ansprüche auf die Vollständigkeit der Beschreibung. Vielmehr soll dem Leser ein Überblick über die Funktionalität der Implementierung gegeben werden. Mehr Einzelheiten über die Optimierung des Netzes und die Durchführung von Messungen entnehmen Sie bitte der Implementationsdokumentation.

Nachrichtenzweck	Bedeutung
FAILURE	Beschreibt eine Aktion, die von einem Agenten veranlasst wurde und aus bestimmten Gründen nicht ausgeführt werden konnte. Wird hauptsächlich zur Fehlererkennung und -behebung verwendet. Eine Nachricht vom Typ „FAILURE“ wird beispielsweise dann erhalten, wenn keine Verbindung zur Agentenplattform hergestellt werden kann.
INFORM	Ein Agent erhält eine Meldung vom Typ „INFORM“, wenn ein anderer Agent ihm bestimmte Informationen, die er selbst für wahr hält, geben will und dabei der Meinung ist, dass der Empfänger-Agent über diese Information noch nicht verfügt. Unsere Agenten können Pakete vom Typ „INFORM“ verwenden, wenn die einen Angriff feststellen und diesen weiter melden oder wenn sie dem Agentennetz beitreten, um die anderen darüber zu informieren. Dieser Meldungstyp wird außerdem vom Angreifer verwendet, um die Kommunikation zwischen den Agenten zu stören, indem er gefälschte Meldungen mit *malicious* Inhalten produziert und diese als Inform-Pakete an Agenten versendet.
NOT_UNDERSTOOD	Meldungen von diesem Typ werden verschickt, wenn ein Agent das von ihm erhaltene Paket nicht verarbeiten kann. Dies kann beispielsweise dann der Fall sein, wenn der Kommunikationspartner nicht die einzuhaltende Spezifikation beachtet und Meldungen erzeugt, die nicht in das vereinbarte Modell passen. Meldungen vom Typ „NOT_UNDERSTOOD“ werden in der aktuellen Implementierung zur Fehlerverfolgung und -behebung verwendet.
QUERY_IF	Wird benutzt, um die Bestätigung einer Aussage (oder deren Negierung) zu erhalten. Der anfragende Agent kennt nicht den Wahrheitswert der von ihm gesendeten Aussage und ist der Meinung, dass ein anderer Agent in der Lage ist, ihm diese zu beantworten. Nachrichten von diesem Typ sind für den Fall reserviert, dass ein Agent bei einem verdächtigen Datenpaket[96]andere Agenten darum bittet, es für ihn zu analysieren.

Tabelle 7: Die von Agenten akzeptierten Nachrichtentypen.

6.4.1 Steuerung des Agentennetzes

Der Start des Agentennetzes setzt das Vorhandensein des *JADE*-Frameworks auf dem entsprechenden Zielhost voraus[97]. Die Multiagentenplattform kann mit einem einzigen Kommando gestartet werden:

[97]Mehr Informationen zur Einrichtung des *JADE*-Frameworks entnehmen Sie bitte dem der Distribution beiliegenden „*JADE* Administrator's Guide“.

```
java jade.Boot [-gui]
```

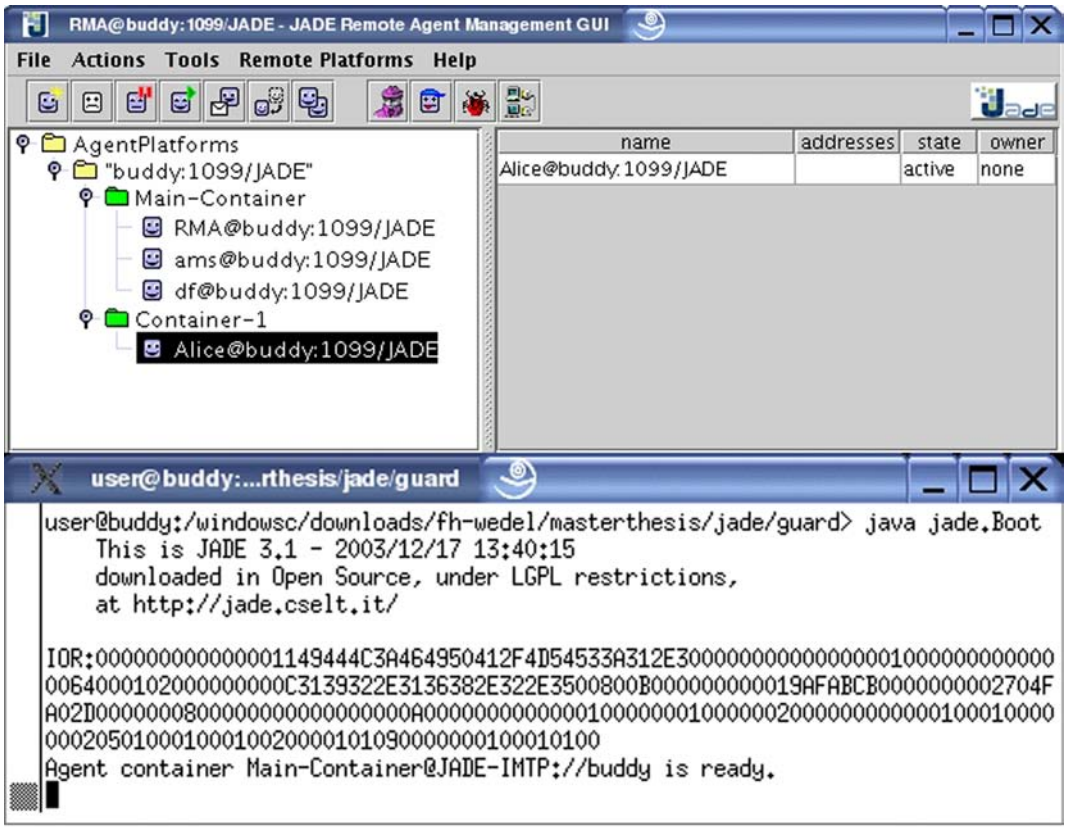

Abbildung 17: Start der Multiagentenplattform mit und ohne graphische Benutzeroberfläche.

Der optionale Parameter `-gui` sorgt dafür, dass die Plattform mit einem leicht zu bedienenden graphischen Konfigurationswerkzeug gestartet wird. Nun kann man bereits die vorkonfigurierten Agenten (Sniffer bzw. Dummy-Agenten) erzeugen oder den eigenen Agenten der Plattform hinzufügen. Die Guardian-Agenten können mit folgendem Kommando dem Agentennetz hinzugefügt werden:

```
java jade.Boot -container AgentName:guard.Guard
```

Der Paramter `AgentName` entspricht dabei dem Namen der Konfiguaraitionsdatei eines Agenten mit der Erweiterung `.dat`. In dieser Datei kann das Agentenverhalten individualisert werden. Im Anhang 2 ist der Aufbau einer Beispiel-Konfigurationsdatei eines Agenten aufgelistet. Ein Agent kann auch im Hintergrund

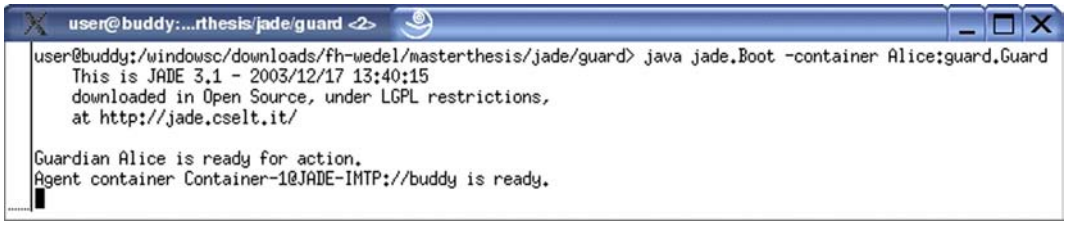

Abbildung 18: Das Hinzufügen eines Agenten zu der Multiagentenumgebung.

gestartet werden; dadurch wird es möglich in einer Konsole mehrere Agenten laufen zu lassen. Dies kann z.B. auf einem Unix/Linux-System durch den folgenden Aufruf erfolgen:

```
java jade.Boot -container AgentName:Guard &
```

Besonders hilfreich ist dieses Feature dann, wenn ein aus vielen Nodes bestehendes Agentennetz erzeugt werden muss. So wurde während der Sammlung von statistischen Daten das Netzwerk aus dreißig Agenten mit Hilfe des folgenden einfachen Bash-Skriptes gestartet:

```
#!/bin/bash

NUMBEROFAGENTSTOSTART=30

# This simple shell script starts a number of Agents specified
# in the variable "NUNMBEROFAGENTSTOSTART"

echo "*****************************************************"
echo "This script will now start $NUMBEROFAGENTSTOSTART Agents"

for ((i=0; i < $NUMBEROFAGENTSTOSTART; i++)) do
        java java jade.Boot -container "Agent$i:guard.Guard()" &
        sleep 10;
done
echo "*****************************************************"
echo "$NUMBEROFAGENTSTOSTART Agents were successfully started"
```

Obwohl Agenten mit einer Logging-Funktionalität ausgestattet sind und alle ihre Aktionen protokollieren können, macht es Sinn, die Meldungen einzelner Agenten an einer zentralen Stelle zwecks einfacherer Datenauswertung zu sammeln. Dafür ist der so genannte Logger-Agent zuständig. Er kann durch einen einfachen Aufruf gestartet werden.

```
java jade.Boot Logger:guard.Logger -container
```

Ein Logger-Agent sammelt sämtliche Informationen, die er von Agenten erhält, formatiert diese und gibt sie aus. Um zu einem späteren Zeitpunkt, Ergebnisse bestimmter Tests nachvollziehen zu können, werden diese Meldungen in einer Datei festgehalten. Durch einen optionalen Parameter ist es möglich, dem Logger-Agenten den Namen dieser Datei mitzuteilen:

```
java jade.Boot  -container "Logger.guard.Logger(session.log)"
```

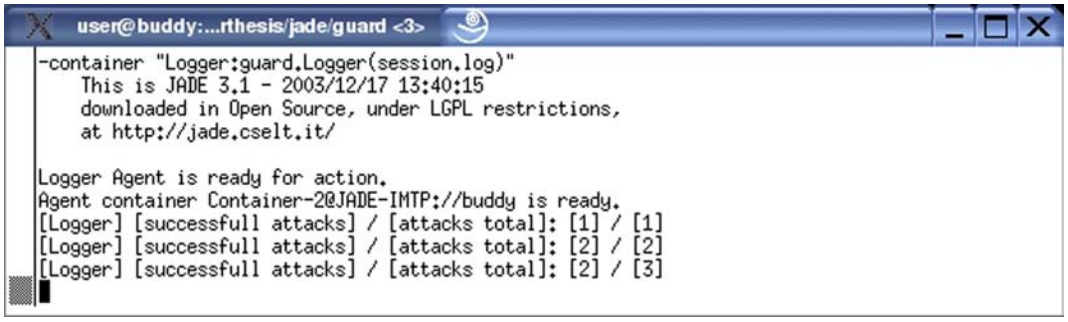

Abbildung 19: Ein Logger-Agent protokolliert die Zustandsveränderungen des Agentennetzes.

6.4.2 Statistische Untersuchungen

Nachdem das Agentennetz initialisiert ist, kann man mit Hilfe eines weiteren Agenten Angriffe gegen bestimmte Mitglieder dieses Netzes durchführen. Dazu braucht man den Namen des Opfer-Rechners und eine Angriffssignatur z.B.:

```
java jade.Boot Malloy:guard.Attacker(Alice maliciouscode1) -container
```

Der Attacker-Agent kann außerdem mit einem einzigen Befehl angewiesen werden, eine große Anzahl anderer Agenten anzugreifen. Dieses Feature war beim Durchführen von statistischen Untersuchungen besonders hilfreich. Außerdem verfügt der Attacker-Agent über weitere Features, um z.B. das Agentennetzwerk in den Initialzustand zu versetzen, damit die statistischen Untersuchungen der Vorrunden, aktuelle Testergebnisse nicht beeinflussen. Dies geschieht mit Hilfe vom Attacker-Agent, der ein bequemes Werkzeug zum Testen und Verwalten des Netzes darstellt.

6.5 Ausblick

Die im Rahmen dieser Arbeit entstandene Implementierung verfügt über die Basisfunktionalität, die es möglich macht, das Konzept des eine Infrastruktur schützenden Agentennetzes zu verifizieren. Gleichzeitig wurden einige in vorhergehenden Kapiteln besprochene Features[98] in der aktuellen Implementierung noch nicht vollständig umgesetzt. Der Code der Implementierung wurde bewusst ausführlich dokumentiert, in der Hoffnung, dass weitere Informatiker, die sich für die beschriebenen Konzepte interessieren, die Arbeit an diesem, meiner Meinung nach, sehr interessanten Thema fortsetzen.

6.6 Zusammenfassung

In diesem Kapitel wurden einige Details der im Rahmen dieser Arbeit stattgefundenden Implementierung erläutert. Besondere Aufmerksamkeit galt dabei den von Agenten verwendeten Nachrichtentypen, denn diese bilden die Kommunikationsbasis des Agentennetzes und sind daher von entscheidender Bedeutung für dessen problemlose Funktionieren. Ein weiterer Abschnitt wurde dem wichtigen Thema „Verbindungsaufbau mit dem Agentennetz“ gewidmet. Der Rest des Kapitels beinhaltete eine kurze Anleitung zum Start und zur Steuerung des Agentennetzes.

[98] z.B. Verschlüsselung der Kommunikation, Vertrauensstellungen zwischen den Agenten etc.

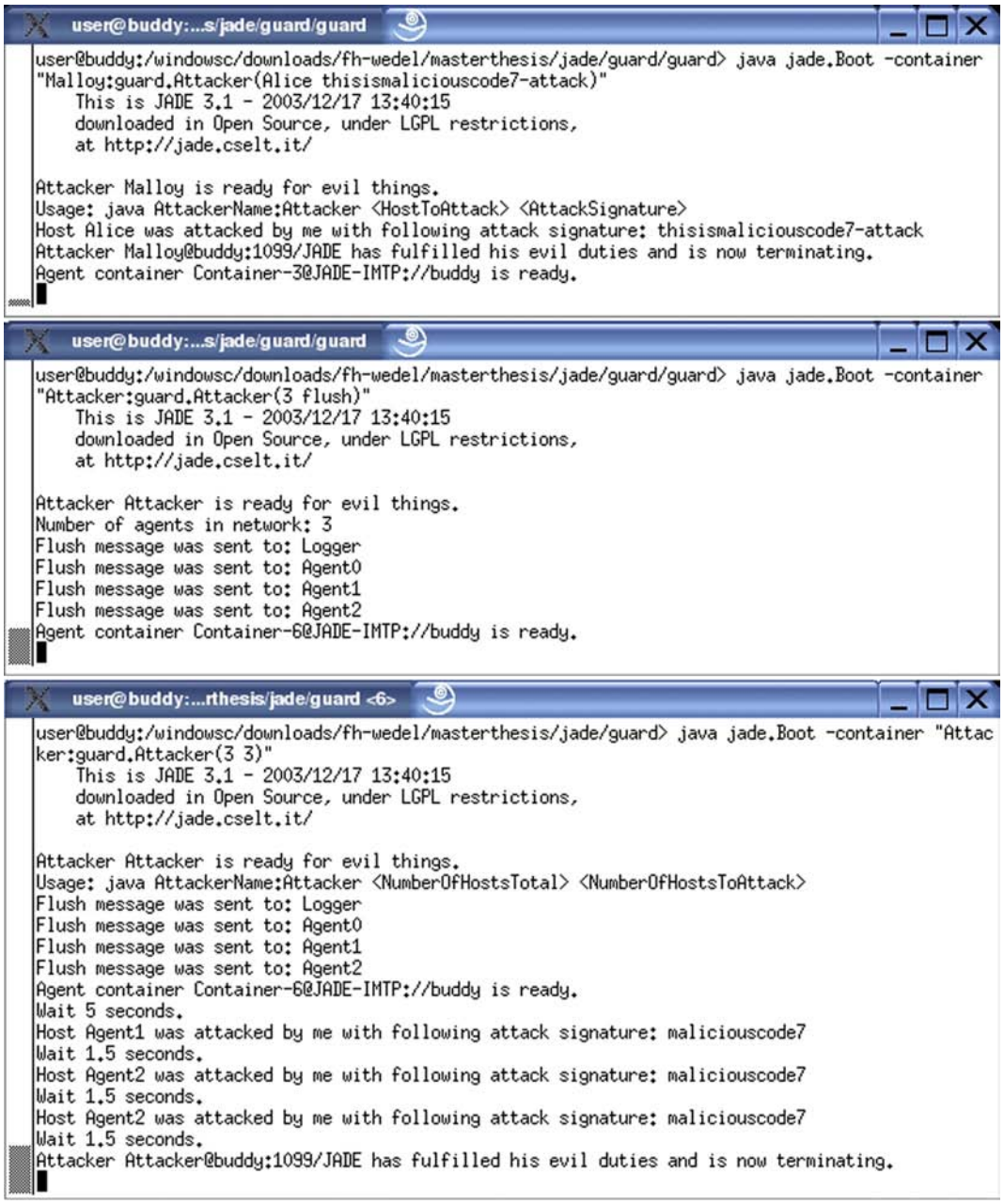

Abbildung 20: Mit Hilfe des Attacker-Agenten ist es nicht nur möglich, bestimmte Agenten anzugreifen, sondern auch die Kommandos an das Agentennetz zu senden, um es beispielsweise zu reinitialisieren.

7 Fazit

Die zentrale Problemstellung dieser Arbeit, Erstellung eines auf dem Agentenansatz basierten Modells zum Schutz von Systemen, wurde gelöst. Die deutliche Überlegenheit des im Rahmen der Arbeit vorgestellten Ansatzes wurde anhand von mathematischen Berechnungen bewiesen und auch mit Hilfe von Implementierung und anschließender statistischer Untersuchungen verifiziert. Die im Rahmen dieser Arbeit stattgefundene Implementierung fand auf Basis des *JADE*-Frameworks statt und bereicherte dessen Funktionalität durch einige neue Features.

Die Erstellung des beschriebenen Modells verlangte eine Auseinandersetzung mit den Grundlagen von Angriffserkennung und dem Vergleich des klassischen und des agentenorientierten Ansatzes. Diese Aufgabe wurde deswegen zu einer interessanten Herausforderung, weil die Fachliteratur nur wenige Informationen zu diesem Thema liefert[99].

Der Aufbau der entwickelten Umgebung und der Agenten wurde bewusst modular gehalten. Der Implementierungscode wurde nach Möglichkeit ausführlich dokumentiert in der Hoffnung, dass einige Interessenten den Ansatz als Gedankenanstoß für ihre Arbeit empfinden werden und das Agentennetz durch weitere Features vervollständigen. Besonders wünschenswert wäre dabei nicht nur die Umsetzung der im Rahmen dieser Arbeit angesprochenen Konzepte, sondern auch das Einbringen von vollständig neuen Ideen, die das Modell des Agentennetzes robuster und effizienter gestalten und die Angriffserkennung innerhalb des Agentennetzes noch zuverlässiger machen könnten.

[99] Als besonders frustrierend erwiesen sich diverse Lektüren von Büchern über *Computerviren*, weil sie sich in der Regel weder mit den Methoden der Angriffserkenung auseinandersetzen, noch eine saubere Eingrenzung des Gefahrenpotentials aufweisen.

8 Anhang 1 (Ergebnisse statistischer Untersuchungen)

Anzahl Angriffe (gesamt)/erfolgreiche Angriffe

Runde	1	5	10	15	20	25	30
1	1	5	10	15	20	25	29
2	1	5	9	14	20	22	22
3	1	5	10	15	19	19	21
4	0	4	7	8	8	8	11
5	1	5	8	11	11	12	13
6	1	1	1	3	3	3	4
7	1	4	4	4	5	5	6
8	1	5	9	12	14	14	14
9	1	5	10	15	20	25	30
10	1	5	10	15	20	25	26
11	1	5	10	15	20	25	30
12	1	5	7	8	9	9	9
13	1	5	7	8	9	10	11
14	1	5	10	15	20	21	21
15	1	5	10	15	20	23	23
16	1	5	6	6	7	8	8
17	1	4	4	4	5	6	6
18	1	5	9	14	19	24	29
19	1	5	10	14	15	16	17
20	1	5	10	15	20	24	29
Arithm. Mittel:	0,95	4,65	8,05	11,3	14,2	16,2	17,95
Varianz:	0,0500	0,8711	6,6816	19,3789	40,2737	62,8000	83,1026
Standardabweichung:	0,2236	0,9333	2,5849	4,4022	6,3462	7,9246	9,1161

Tabelle 8: Statistische Ergebnisse des Agentennetzes mit einer Signatur.

Größe der Kontaktliste eines Agenten

4	2	7	5	7
4	3	7	9	9
3	4	3	1	10
5	2	7	4	14
17	8	5	4	5
5	5	7	1	9
Summe:				176,0000
Arithm. Mittel:				5,8667
Varianz:				12,8092
Standardabweichung:				3,5790

Tabelle 9: Struktur des Agentennetzes (Testreihe 1).

Anzahl Angriffe (gesamt)/erfolgreiche Angriffe							
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
1	1	5	9	14	18	23	28
2	1	5	10	15	17	17	17
3	1	5	10	15	20	25	30
4	1	4	5	6	6	6	6
5	1	5	7	7	7	7	7
6	1	5	10	15	20	25	30
7	1	5	10	15	20	24	29
8	1	5	10	15	19	19	19
9	0	0	0	0	0	0	0
10	1	4	5	6	9	9	9
11	1	5	10	13	16	17	17
12	1	5	6	7	8	9	9
13	0	0	0	0	0	0	0
14	1	5	10	15	17	17	17
15	1	5	9	11	12	12	12
16	1	4	8	11	11	11	11
17	1	5	10	13	13	13	13
18	1	5	10	14	14	15	15
19	1	5	6	6	6	6	6
20	1	5	10	13	15	15	15
Arithm. Mittel:	0,9	4,35	7,75	10,55	12,4	13,5	14,5
Varianz:	0,0947	2,3447	10,3026	24,7868	40,2526	58,1579	85,0000
Standardabweichung:	0,3078	1,5313	3,2098	4,9786	6,3445	7,6261	9,2195

Tabelle 10: Statistische Ergebnisse des Agentennetzes mit zwei Signaturen.

Größe der Kontaktliste eines Agenten				
4	8	12	3	8
2	2	1	4	7
3	8	4	12	8
9	4	4	2	4
9	3	12	13	15
4	12	15	9	7
Summe:				208,0000
Arithm. Mittel:				6,9333
Varianz:				17,3057
Standardabweichung:				4,1600

Tabelle 11: Struktur des Agentennetzes (Testreihe 2).

Anzahl Angriffe (gesamt)/erfolgreiche Angriffe							
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
1	0	0	0	0	0	0	0
2	1	3	8	11	12	13	13
3	0	0	0	0	0	0	0
4	1	3	3	3	3	3	3
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	1	3	3	3	3	3	3
8	1	4	4	4	4	4	4
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1
13	0	0	0	0	0	0	0
14	1	3	8	10	10	10	10
15	1	4	4	4	4	4	4
16	0	0	0	0	0	0	0
17	1	4	9	11	14	16	16
18	1	3	3	3	3	3	3
19	1	2	2	2	2	2	2
20	0	0	0	0	0	0	0
Arithm. Mittel:	0,55	1,55	2,3	2,65	2,85	3	3
Varianz:	0,2605	2,6816	8,8526	14,0289	18,0289	21,5789	21,5789
Standardabweichung:	0,5104	1,6376	2,9753	3,7455	4,2461	4,6453	4,6453

Tabelle 12: Statistische Ergebnisse des Agentennetzes mit drei Signaturen.

Größe der Kontaktliste eines Agenten				
3	8	23	19	13
19	9	19	19	26
22	19	19	12	22
5	12	19	23	27
24	11	13	25	25
22	3	23	27	23
Summe:				534,0000
Arithm. Mittel:				6,9333
Varianz:				51,2000
Standardabweichung:				7,1554

Tabelle 13: Struktur des Agentennetzes (Testreihe 3).

Anzahl Angriffe (gesamt)/erfolgreiche Angriffe							
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
1	0	2	4	4	4	4	4
2	0	0	0	0	0	0	0
3	1	1	2	2	2	2	2
4	1	5	7	7	7	7	7
5	1	2	2	2	2	2	2
6	1	4	4	4	4	4	4
7	0	0	0	0	0	0	0
8	0	0	0	1	1	1	1
9	1	2	2	2	2	2	2
10	0	0	0	0	0	0	0
11	1	5	5	5	5	5	5
12	0	1	2	2	2	2	2
13	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1
15	1	4	5	5	5	5	5
16	1	4	4	4	4	4	4
17	1	5	5	5	5	5	5
18	0	0	0	0	0	1	1
19	0	0	1	1	1	1	1
20	1	1	1	1	1	1	1
Arithm. Mittel:	0,6	1,9	2,3	2,35	2,35	2,4	2,4
Varianz:	0,2526	3,5684	4,5368	4,3447	4,3447	4,1474	4,1474
Standardabweichung:	0,5026	1,8890	2,1300	2,0844	2,0844	2,0365	2,0365

Tabelle 14: Statistische Ergebnisse des Agentennetzes mit vier Signaturen.

Größe der Kontaktliste eines Agenten				
6	17	11	6	20
11	1	13	13	19
8	2	12	17	19
8	9	19	11	17
11	11	10	10	10
8	17	8	2	23
Summe:				349,0000
Arithm. Mittel:				11,6333
Varianz:				31,4816
Standardabweichung:				5,6108

Tabelle 15: Struktur des Agentennetzes (Testreihe 4).

Anzahl Angriffe (gesamt)/erfolgreiche Angriffe							
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1
4	0	1	1	1	1	1	1
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	1	4	4	4	4	4	4
9	0	1	2	2	2	2	2
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	1	3	3	3	3	3	3
13	1	4	5	5	5	5	5
14	1	1	1	1	1	1	1
15	1	2	2	2	2	2	2
16	1	5	5	5	5	5	5
17	1	1	1	1	1	1	1
18	0	1	1	1	1	1	1
19	1	1	1	1	1	1	1
20	0	0	1	1	1	1	1
Arithm. Mittel:	0,5	1,3	1,45	1,45	1,45	1,45	1,45
Varianz:	0,2632	2,3263	2,5763	2,5763	2,5763	2,5763	2,5763
Standardabweichung:	0,5130	1,5252	1,6051	1,6051	1,6051	1,6051	1,6051

Tabelle 16: Statistische Ergebnisse des Agentennetzes mit fünf Signaturen.

Größe der Kontaktliste eines Agenten				
6	5	8	13	15
4	4	5	8	14
7	4	8	9	7
1	6	8	11	5
6	8	8	6	12
7	6	8	14	18
Summe:				241,0000
Arithm. Mittel:				8,0333
Varianz:				14,4471
Standardabweichung:				3,8009

Tabelle 17: Struktur des Agentennetzes (Testreihe 5).

Anzahl Angriffe (gesamt)/erfolgreiche Angriffe							
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
1	0	1	1	1	1	1	1
2	1	2	2	2	2	2	2
3	1	3	3	3	3	3	3
4	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0
6	1	4	4	4	4	4	4
7	0	0	0	0	0	0	0
8	0	1	1	1	1	1	1
9	0	0	0	0	0	0	0
10	0	2	2	2	2	2	2
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	1	3	3	3	3	3	3
15	0	0	0	0	0	0	0
16	1	1	1	1	1	1	1
17	1	3	4	4	4	4	4
18	0	0	0	0	0	0	0
19	0	2	2	2	2	2	2
20	0	0	0	0	0	0	0
Arithm. Mittel:	0,35	1,15	1,2	1,2	1,2	1,2	1,2
Varianz:	0,2395	1,7132	1,9579	1,9579	1,9579	1,9579	1,9579
Standardabweichung:	0,4894	1,3089	1,3992	1,3992	1,3992	1,3992	1,3992

Tabelle 18: Statistische Ergebnisse des Agentennetzes mit sechs Signaturen.

Größe der Kontaktliste eines Agenten				
7	2	14	7	6
13	14	2	8	19
9	14	2	13	7
1	4	13	16	17
4	4	21	14	17
4	10	4	18	16
Summe:				300,0000
Arithm. Mittel:				10,0000
Varianz:				35,4483
Standardabweichung:				5,9538

Tabelle 19: Struktur des Agentennetzes (Testreihe 6).

Anzahl Angriffe (gesamt)/erfolgreiche Angriffe							
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
1	0	0	0	0	1	1	1
2	0	1	1	1	1	1	1
3	1	2	2	2	2	2	2
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	1	1	1	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	2	2	2	2	2	2
11	0	2	2	2	2	2	2
12	0	2	2	2	3	3	3
13	0	0	0	0	0	0	0
14	1	1	1	1	1	1	1
15	0	1	1	1	1	1	1
16	1	3	3	3	3	3	3
17	0	1	1	1	1	1	1
18	0	1	1	1	1	1	1
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
Arithm. Mittel:	0,2	0,85	0,85	0,85	0,95	0,95	0,95
Varianz:	0,1684	0,8711	0,8711	0,8711	0,9974	0,9974	0,9974
Standardabweichung:	0,4104	0,9333	0,9333	0,9333	0,9987	0,9987	0,9987

Tabelle 20: Statistische Ergebnisse des Agentennetzes mit sieben Signaturen.

Größe der Kontaktliste eines Agenten				
1	2	12	3	20
18	3	20	2	21
2	18	20	19	18
18	2	18	2	21
15	20	17	12	1
19	18	20	20	18
Summe:				400,0000
Arithm. Mittel:				13,3333
Varianz:				61,4023
Standardabweichung:				7,8360

Tabelle 21: Struktur des Agentennetzes (Testreihe 7).

Anzahl Angriffe (gesamt)/erfolgreiche Angriffe							
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0
6	0	1	1	1	1	1	1
7	0	0	0	0	0	0	0
8	1	1	1	1	1	1	1
9	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	1	1	1	1	1	1	1
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1
Arithm. Mittel:	0,35	0,4	0,4	0,4	0,4	0,4	0,4
Varianz:	0,2395	0,2526	0,2526	0,2526	0,2526	0,2526	0,2526
Standardabweichung:	0,4894	0,5026	0,5026	0,5026	0,5026	0,5026	0,5026

Tabelle 22: Statistische Ergebnisse des Agentennetzes mit acht Signaturen.

Größe der Kontaktliste eines Agenten				
5	2	16	11	12
11	15	5	16	16
6	8	8	6	22
4	17	11	19	9
2	2	7	1	4
11	6	15	11	8
Summe:				286,0000
Arithm. Mittel:				9,5333
Varianz:				31,0161
Standardabweichung:				5,5692

Tabelle 23: Struktur des Agentennetzes (Testreihe 8).

Anzahl Angriffe (gesamt)/erfolgreiche Angriffe							
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
Runde	**1**	**5**	**10**	**15**	**20**	**25**	**30**
1	0	0	0	0	0	0	0
2	0	1	1	1	1	1	1
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	1	2	2	2	3	4	5
12	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1
14	1	2	2	2	3	3	3
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	1	1	1	1
20	0	0	0	0	0	0	0
Arithm. Mittel:	0,2	0,35	0,35	0,4	0,5	0,55	0,6
Varianz:	0,1684	0,4500	0,4500	0,4632	0,8947	1,2079	1,6211
Standardabweichung:	0,4104	0,6708	0,6708	0,6806	0,9459	1,0990	1,2732

Tabelle 24: Statistische Ergebnisse des Agentennetzes mit neun Signaturen.

Größe der Kontaktliste eines Agenten				
13	6	13	16	4
6	1	13	2	3
15	4	6	13	1
1	14	14	6	14
5	15	4	23	18
4	13	3	15	5
Summe:				270,0000
Arithm. Mittel:				9,0000
Varianz:				36,8966
Standardabweichung:				6,0743

Tabelle 25: Struktur des Agentennetzes (Testreihe 9).

9 Anhang 2 (Beispielkonfiguration eines Agenten)

```
# This is a config file for a guardian agent
# Possible values are: [always, sometimes, never]
# "always": (default): agent always warns other agents
# "never":  agent never warns other agents about an attack
# "sometimes": agent sends warnings depending on the value of
# "warnagentsrandomly" and "warntalkingagentsonly"

warnotheragents=sometimes

# if the value of "warnotheragents" is "sometimes" and
# "warnagentsrandomly"=yes, then agent sends warning messages
# depending on his actual mood (randomly)
# if warnotheragents=always or warnotheragents=never, this
# parameter will not be activated

warnagentsrandomly=no

# if the value of "warnotheragents" is "sometimes" and
# warntalkingagentsonly=yes then only the agents, who send warning
#  messages will be informed about an attack. Default value: no

warntalkingagentsonly=yes

# number of signatures, an agent loads during his initialization

numberofsignatures=2

# signature list to load

signature0=maliciouscode2
signature1=maliciouscode3

# number of warnings to apply signature (integer value)

warningstoapplysignature=1

# number of hosts, an agent tries to contact on boot (integer value)

numberofknownagents=2

# names of the hosts to load

agent0=agent8
agent1=agent20
```

10 Anhang 3 (Statistische Berechnungen)

10.1 Erwartungswert bei einer (1) Signatur

Die in den folgenden Tabellen dargestellten Ergebnisse wurden nach der vierten wesentlichen Nachkommastelle abgeschnitten.

Kompromittierte Systeme (x)	**Wahrscheinlichkeit**	**Erwartungswert**
0 Systeme:	0.1000	0.0000
1 System:	$9.0000 \cdot 10^{-2}$	$9.0000 \cdot 10^{-2}$
2 Systeme:	$8.1000 \cdot 10^{-2}$	$1.6200 \cdot 10^{-1}$
3 Systeme:	$7.2900 \cdot 10^{-2}$	$2.187 \cdot 10^{-1}$
4 Systeme:	$6.5610 \cdot 10^{-2}$	$2.6244 \cdot 10^{-1}$
5 Systeme:	$5.9049 \cdot 10^{-2}$	$2.9524 \cdot 10^{-1}$
6 Systeme:	$5.3144 \cdot 10^{-2}$	$3.1886 \cdot 10^{-1}$
7 Systeme:	$4.7829 \cdot 10^{-2}$	$3.3480 \cdot 10^{-1}$
8 Systeme:	$4.3046 \cdot 10^{-2}$	$3.4437 \cdot 10^{-1}$
9 Systeme:	$3.8742 \cdot 10^{-2}$	$3.4867 \cdot 10^{-1}$
10 Systeme:	$3.4867 \cdot 10^{-2}$	$3.4867 \cdot 10^{-1}$
11 Systeme:	$3.1381 \cdot 10^{-2}$	$3.4519 \cdot 10^{-1}$
12 Systeme:	$2.8242 \cdot 10^{-2}$	$3.3891 \cdot 10^{-1}$
13 Systeme:	$2.5418 \cdot 10^{-2}$	$3.3044 \cdot 10^{-1}$
14 Systeme:	$2.2876 \cdot 10^{-2}$	$3.2027 \cdot 10^{-1}$
15 Systeme:	$2.0589 \cdot 10^{-2}$	$3.0883 \cdot 10^{-1}$
16 Systeme:	$1.8530 \cdot 10^{-2}$	$2.9648 \cdot 10^{-1}$
17 Systeme:	$1.6677 \cdot 10^{-2}$	$2.8351 \cdot 10^{-1}$
18 Systeme:	$1.5009 \cdot 10^{-2}$	$2.7017 \cdot 10^{-1}$
19 Systeme:	$1.3508 \cdot 10^{-2}$	$2.5666 \cdot 10^{-1}$
20 Systeme:	$1.2157 \cdot 10^{-2}$	$2.4315 \cdot 10^{-1}$
21 Systeme:	$1.0941 \cdot 10^{-2}$	$2.2977 \cdot 10^{-1}$
22 Systeme:	$9.8477 \cdot 10^{-3}$	$2.1664 \cdot 10^{-1}$
23 Systeme:	$8.8629 \cdot 10^{-3}$	$2.0384 \cdot 10^{-1}$
24 Systeme:	$7.9766 \cdot 10^{-3}$	$1.9143 \cdot 10^{-1}$
25 Systeme:	$7.1789 \cdot 10^{-3}$	$1.7947 \cdot 10^{-1}$
26 Systeme:	$6.4610 \cdot 10^{-3}$	$1.6798 \cdot 10^{-1}$
27 Systeme:	$5.8149 \cdot 10^{-3}$	$1.5700 \cdot 10^{-1}$
28 Systeme:	$5.2334 \cdot 10^{-3}$	$1.4653 \cdot 10^{-1}$
29 Systeme:	$4.7101 \cdot 10^{-3}$	$1.3659 \cdot 10^{-1}$
30 Systeme:	$4.2391 \cdot 10^{-2}$	1.2717
SUMME:	**1.0000**	**8.6184**

Tabelle 26: Berechnung des Erwartungswertes für eine Signatur.

10.2 Erwartungswert bei zwei (2) Signaturen

Kompromittierte Systeme (x)	**Wahrscheinlichkeit**	**Erwartungswert**
0 Systeme:	0.2000	0.0000
1 System:	0.1600	0.1600
2 Systeme:	0.1280	0.2560
3 Systeme:	0.1024	0.3072
4 Systeme:	$8.1920 \cdot 10^{-2}$	0.3276
5 Systeme:	$6.5536 \cdot 10^{-2}$	0.3276
6 Systeme:	$5.2428 \cdot 10^{-2}$	0.3145
7 Systeme:	$4.1943 \cdot 10^{-2}$	0.2936
8 Systeme:	$3.3554 \cdot 10^{-2}$	0.2684
9 Systeme:	$2.6843 \cdot 10^{-2}$	0.2415
10 Systeme:	$2.1474 \cdot 10^{-2}$	0.2147
11 Systeme:	$1.7179 \cdot 10^{-2}$	0.1889
12 Systeme:	$1.3743 \cdot 10^{-2}$	0.1649
13 Systeme:	$1.0995 \cdot 10^{-2}$	0.1429
14 Systeme:	$8.7960 \cdot 10^{-3}$	0.1231
15 Systeme:	$7.0368 \cdot 10^{-3}$	0.1055
16 Systeme:	$5.6294 \cdot 10^{-3}$	$9.0071 \cdot 10^{-2}$
17 Systeme:	$4.5035 \cdot 10^{-3}$	$7.6561 \cdot 10^{-2}$
18 Systeme:	$3.6028 \cdot 10^{-3}$	$6.4851 \cdot 10^{-2}$
19 Systeme:	$2.8823 \cdot 10^{-3}$	$5.4763 \cdot 10^{-2}$
20 Systeme:	$2.3058 \cdot 10^{-3}$	$4.6116 \cdot 10^{-2}$
21 Systeme:	$1.8446 \cdot 10^{-3}$	$3.8738 \cdot 10^{-2}$
22 Systeme:	$1.4757 \cdot 10^{-3}$	$3.2466 \cdot 10^{-2}$
23 Systeme:	$1.1805 \cdot 10^{-3}$	$2.7153 \cdot 10^{-2}$
24 Systeme:	$9.4447 \cdot 10^{-4}$	$2.2667 \cdot 10^{-2}$
25 Systeme:	$7.5557 \cdot 10^{-4}$	$1.8889 \cdot 10^{-2}$
26 Systeme:	$6.0446 \cdot 10^{-4}$	$1.5716 \cdot 10^{-2}$
27 Systeme:	$4.8357 \cdot 10^{-4}$	$1.3056 \cdot 10^{-2}$
28 Systeme:	$3.8685 \cdot 10^{-4}$	$1.0831 \cdot 10^{-2}$
29 Systeme:	$3.0948 \cdot 10^{-4}$	$8.9750 \cdot 10^{-3}$
30 Systeme:	$1.2379 \cdot 10^{-3}$	$3.7138 \cdot 10^{-2}$
SUMME:	**1.0000**	**3.9950**

Tabelle 27: Berechnung des Erwartungswertes für zwei Signaturen.

10.3 Erwartungswert bei drei (3) Signaturen

Kompromittierte Systeme (x)	**Wahrscheinlichkeit**	**Erwartungswert**
0 Systeme:	0.3000	0.0000
1 System:	0.2100	0.2100
2 Systeme:	0.1469	0.2939
3 Systeme:	0.1028	0.3086
4 Systeme:	$7.2029 \cdot 10^{-2}$	0.2881
5 Systeme:	$5.0420 \cdot 10^{-2}$	0.2521
6 Systeme:	$3.5294 \cdot 10^{-2}$	0.2117
7 Systeme:	$2.4706 \cdot 10^{-2}$	0.1729
8 Systeme:	$1.7294 \cdot 10^{-2}$	0.1383
9 Systeme:	$1.2106 \cdot 10^{-2}$	0.1089
10 Systeme:	$8.4742 \cdot 10^{-3}$	$8.4742 \cdot 10^{-2}$
11 Systeme:	$5.9319 \cdot 10^{-3}$	$6.5251 \cdot 10^{-2}$
12 Systeme:	$4.1523 \cdot 10^{-3}$	$4.9828 \cdot 10^{-2}$
13 Systeme:	$2.9066 \cdot 10^{-3}$	$3.7786 \cdot 10^{-2}$
14 Systeme:	$2.0346 \cdot 10^{-3}$	$2.8485 \cdot 10^{-2}$
15 Systeme:	$1.4242 \cdot 10^{-3}$	$2.1364 \cdot 10^{-2}$
16 Systeme:	$9.9698 \cdot 10^{-4}$	$1.5951 \cdot 10^{-2}$
17 Systeme:	$6.9789 \cdot 10^{-4}$	$1.1864 \cdot 10^{-2}$
18 Systeme:	$4.8852 \cdot 10^{-4}$	$8.7934 \cdot 10^{-3}$
19 Systeme:	$3.4196 \cdot 10^{-4}$	$6.4973 \cdot 10^{-3}$
20 Systeme:	$2.3937 \cdot 10^{-4}$	$4.7875 \cdot 10^{-3}$
21 Systeme:	$1.6756 \cdot 10^{-4}$	$3.5188 \cdot 10^{-3}$
22 Systeme:	$1.1729 \cdot 10^{-4}$	$2.5804 \cdot 10^{-3}$
23 Systeme:	$8.2106 \cdot 10^{-5}$	$1.8884 \cdot 10^{-3}$
24 Systeme:	$5.7474 \cdot 10^{-5}$	$1.3793 \cdot 10^{-3}$
25 Systeme:	$4.0232 \cdot 10^{-5}$	$1.0058 \cdot 10^{-3}$
26 Systeme:	$2.8162 \cdot 10^{-5}$	$7.3222 \cdot 10^{-4}$
27 Systeme:	$1.9713 \cdot 10^{-5}$	$5.3227 \cdot 10^{-4}$
28 Systeme:	$1.3799 \cdot 10^{-5}$	$3.8638 \cdot 10^{-4}$
29 Systeme:	$9.6597 \cdot 10^{-6}$	$2.8013 \cdot 10^{-4}$
30 Systeme:	$2.2539 \cdot 10^{-5}$	$6.7618 \cdot 10^{-4}$
SUMME:	**1.0000**	**2.3332**

Tabelle 28: Berechnung des Erwartungswertes für drei Signaturen..

10.4 Erwartungswert bei vier (4) Signaturen

Kompromittierte Systeme (x)	**Wahrscheinlichkeit**	**Erwartungswert**
0 Systeme:	0.4000	0.0000
1 System:	0.2400	0.2400
2 Systeme:	0.1440	0.2880
3 Systeme:	$8.6399 \cdot 10^{-2}$	0.2592
4 Systeme:	$5.1840 \cdot 10^{-2}$	0.2073
5 Systeme:	$3.1103 \cdot 10^{-2}$	0.1555
6 Systeme:	$1.8662 \cdot 10^{-2}$	0.1119
7 Systeme:	$1.1119 \cdot 10^{-2}$	$7.8382 \cdot 10^{-2}$
8 Systeme:	$6.7184 \cdot 10^{-3}$	$5.3747 \cdot 10^{-2}$
9 Systeme:	$4.0310 \cdot 10^{-3}$	$3.6279 \cdot 10^{-2}$
10 Systeme:	$2.4186 \cdot 10^{-3}$	$2.4186 \cdot 10^{-2}$
11 Systeme:	$1.4511 \cdot 10^{-3}$	$1.5963 \cdot 10^{-2}$
12 Systeme:	$8.7071 \cdot 10^{-4}$	$1.0448 \cdot 10^{-2}$
13 Systeme:	$5.2242 \cdot 10^{-4}$	$6.7915 \cdot 10^{-3}$
14 Systeme:	$3.1345 \cdot 10^{-4}$	$4.3883 \cdot 10^{-3}$
15 Systeme:	$1.8807 \cdot 10^{-4}$	$2.8211 \cdot 10^{-3}$
16 Systeme:	$1.1284 \cdot 10^{-4}$	$1.8055 \cdot 10^{-3}$
17 Systeme:	$6.7706 \cdot 10^{-5}$	$1.1510 \cdot 10^{-3}$
18 Systeme:	$4.0623 \cdot 10^{-5}$	$7.3123 \cdot 10^{-4}$
19 Systeme:	$2.4374 \cdot 10^{-5}$	$4.6311 \cdot 10^{-4}$
20 Systeme:	$1.4624 \cdot 10^{-5}$	$2.9249 \cdot 10^{-4}$
21 Systeme:	$8.7747 \cdot 10^{-6}$	$1.8427 \cdot 10^{-4}$
22 Systeme:	$5.2648 \cdot 10^{-6}$	$1.1582 \cdot 10^{-4}$
23 Systeme:	$3.1589 \cdot 10^{-6}$	$7.2655 \cdot 10^{-5}$
24 Systeme:	$1.8953 \cdot 10^{-6}$	$4.5488 \cdot 10^{-5}$
25 Systeme:	$1.1372 \cdot 10^{-6}$	$2.8430 \cdot 10^{-5}$
26 Systeme:	$6.8232 \cdot 10^{-7}$	$1.7740 \cdot 10^{-5}$
27 Systeme:	$4.0939 \cdot 10^{-7}$	$1.1053 \cdot 10^{-5}$
28 Systeme:	$2.4563 \cdot 10^{-7}$	$6.8778 \cdot 10^{-6}$
29 Systeme:	$1.4738 \cdot 10^{-7}$	$4.2740 \cdot 10^{-6}$
30 Systeme:	$2.2107 \cdot 10^{-7}$	$6.6322 \cdot 10^{-6}$
SUMME:	**1.0000**	**1.4999**

Tabelle 29: Berechnung des Erwartungswertes für vier Signaturen.

10.5 Erwartungswert bei fünf (5) Signaturen

Bei fünf Signaturen handelt es sich um einen Sonderfall, denn sowohl die Wahrscheinlichkeit eines positiven Agriffs, als auch die Wahrscheinlichkeit einer erfolgreichen Abwehr sind gleich. In solchen Fällen spricht man von einem Laplace-Experiment.

Kompromittierte Systeme (x)	**Wahrscheinlichkeit**	**Erwartungswert**
0 Systeme:	0.5000	0.0000
1 System:	0.2500	0.2500
2 Systeme:	0.1250	0.2500
3 Systeme:	$6.2500 \cdot 10^{-2}$	0.1875
4 Systeme:	$3.1250 \cdot 10^{-2}$	0.1250
5 Systeme:	$1.5625 \cdot 10^{-2}$	$7.8125 \cdot 10^{-2}$
6 Systeme:	$7.8125 \cdot 10^{-3}$	$4.6875 \cdot 10^{-2}$
7 Systeme:	$3.9062 \cdot 10^{-3}$	$2.7343 \cdot 10^{-2}$
8 Systeme:	$1.9531 \cdot 10^{-3}$	$1.5625 \cdot 10^{-2}$
9 Systeme:	$9.7656 \cdot 10^{-4}$	$8.7890 \cdot 10^{-3}$
10 Systeme:	$4.8828 \cdot 10^{-4}$	$4.8828 \cdot 10^{-3}$
11 Systeme:	$2.4414 \cdot 10^{-4}$	$2.6855 \cdot 10^{-3}$
12 Systeme:	$1.2207 \cdot 10^{-4}$	$1.4648 \cdot 10^{-3}$
13 Systeme:	$6.1035 \cdot 10^{-5}$	$7.9345 \cdot 10^{-4}$
14 Systeme:	$3.0517 \cdot 10^{-5}$	$4.2724 \cdot 10^{-4}$
15 Systeme:	$1.5258 \cdot 10^{-5}$	$2.2888 \cdot 10^{-4}$
16 Systeme:	$7.6293 \cdot 10^{-6}$	$1.2207 \cdot 10^{-4}$
17 Systeme:	$3.8146 \cdot 10^{-6}$	$6.4849 \cdot 10^{-5}$
18 Systeme:	$1.9073 \cdot 10^{-6}$	$3.4332 \cdot 10^{-5}$
19 Systeme:	$9.5367 \cdot 10^{-7}$	$1.8119 \cdot 10^{-5}$
20 Systeme:	$4.7683 \cdot 10^{-7}$	$9.5367 \cdot 10^{-6}$
21 Systeme:	$2.3841 \cdot 10^{-7}$	$5.0067 \cdot 10^{-6}$
22 Systeme:	$1.1920 \cdot 10^{-7}$	$2.6226 \cdot 10^{-6}$
23 Systeme:	$5.9604 \cdot 10^{-8}$	$1.3709 \cdot 10^{-6}$
24 Systeme:	$2.9802 \cdot 10^{-8}$	$7.1525 \cdot 10^{-7}$
25 Systeme:	$1.4901 \cdot 10^{-8}$	$3.7252 \cdot 10^{-7}$
26 Systeme:	$7.4505 \cdot 10^{-9}$	$1.9371 \cdot 10^{-7}$
27 Systeme:	$3.7252 \cdot 10^{-9}$	$1.0058 \cdot 10^{-7}$
28 Systeme:	$1.8626 \cdot 10^{-9}$	$5.2154 \cdot 10^{-8}$
29 Systeme:	$9.3132 \cdot 10^{-10}$	$2.70083 \cdot 10^{-8}$
30 Systeme:	$9.3132 \cdot 10^{-10}$	$2.7939 \cdot 10^{-8}$
SUMME:	**1.0000**	**0.9999**

Tabelle 30: Berechnung des Erwartungswertes für fünf Signaturen.

10.6 Erwartungswert bei sechs (6) Signaturen

Kompromittierte Systeme (x)	**Wahrscheinlichkeit**	**Erwartungswert**
0 Systeme:	0.6000	0.000
1 System:	0.2400	0.2400
2 Systeme:	$9.6000 \cdot 10^{-2}$	0.1920
3 Systeme:	$3.8400 \cdot 10^{-2}$	0.1152
4 Systeme:	$1.5360 \cdot 10^{-2}$	$6.1440 \cdot 10^{-2}$
5 Systeme:	$6.1440 \cdot 10^{-3}$	$3.0720 \cdot 10^{-2}$
6 Systeme:	$2.4576 \cdot 10^{-3}$	$1.4745 \cdot 10^{-2}$
7 Systeme:	$9.8304 \cdot 10^{-4}$	$6.8812 \cdot 10^{-3}$
8 Systeme:	$3.9322 \cdot 10^{-4}$	$3.1457 \cdot 10^{-3}$
9 Systeme:	$1.5729 \cdot 10^{-4}$	$1.4155 \cdot 10^{-3}$
10 Systeme:	$6.2915 \cdot 10^{-5}$	$6.2914 \cdot 10^{-4}$
11 Systeme:	$2.5166 \cdot 10^{-5}$	$2.7682 \cdot 10^{-4}$
12 Systeme:	$1.0066 \cdot 10^{-5}$	$1.2079 \cdot 10^{-4}$
13 Systeme:	$4.0265 \cdot 10^{-6}$	$5.2344 \cdot 10^{-5}$
14 Systeme:	$1.6106 \cdot 10^{-6}$	$2.2548 \cdot 10^{-5}$
15 Systeme:	$6.4425 \cdot 10^{-7}$	$9.6636 \cdot 10^{-6}$
16 Systeme:	$2.5770 \cdot 10^{-7}$	$4.1231 \cdot 10^{-6}$
17 Systeme:	$1.0308 \cdot 10^{-7}$	$1.7523 \cdot 10^{-6}$
18 Systeme:	$4.1232 \cdot 10^{-8}$	$7.4217 \cdot 10^{-7}$
19 Systeme:	$1.6493 \cdot 10^{-8}$	$3.1336 \cdot 10^{-7}$
20 Systeme:	$6.5971 \cdot 10^{-9}$	$1.3194 \cdot 10^{-7}$
21 Systeme:	$2.6388 \cdot 10^{-9}$	$5.5415 \cdot 10^{-8}$
22 Systeme:	$1.0555 \cdot 10^{-9}$	$2.3221 \cdot 10^{-8}$
23 Systeme:	$4.2221 \cdot 10^{-10}$	$9.7108 \cdot 10^{-9}$
24 Systeme:	$1.6888 \cdot 10^{-10}$	$4.0532 \cdot 10^{-9}$
25 Systeme:	$6.7553 \cdot 10^{-11}$	$1.6888 \cdot 10^{-9}$
26 Systeme:	$2.7021 \cdot 10^{-11}$	$7.0256 \cdot 10^{-10}$
27 Systeme:	$1.0808 \cdot 10^{-11}$	$2.9183 \cdot 10^{-10}$
28 Systeme:	$4.3234 \cdot 10^{-12}$	$1.2105 \cdot 10^{-10}$
29 Systeme:	$1.7293 \cdot 10^{-12}$	$5.0152 \cdot 10^{-11}$
30 Systeme:	$1.1529 \cdot 10^{-12}$	$3.4587 \cdot 10^{-11}$
SUMME:	**1.0000**	**0.6666**

Tabelle 31: Berechnung des Erwartungswertes für sechs Signaturen.

10.7 Erwartungswert bei sieben (7) Signaturen

Kompromittierte Systeme (x)	**Wahrscheinlichkeit**	**Erwartungswert**
0 Systeme:	0.7000	0.0000
1 System:	0.2100	0.2100
2 Systeme:	$6.3000 \cdot 10^{-2}$	0.1260
3 Systeme:	$1.8899 \cdot 10^{-2}$	$5.6699 \cdot 10^{-2}$
4 Systeme:	$5.6700 \cdot 10^{-3}$	$2.2680 \cdot 10^{-2}$
5 Systeme:	$1.7009 \cdot 10^{-3}$	$8.5049 \cdot 10^{-3}$
6 Systeme:	$5.1029 \cdot 10^{-4}$	$3.0617 \cdot 10^{-3}$
7 Systeme:	$1.5308 \cdot 10^{-4}$	$1.0716 \cdot 10^{-3}$
8 Systeme:	$4.5926 \cdot 10^{-5}$	$3.6741 \cdot 10^{-4}$
9 Systeme:	$1.3778 \cdot 10^{-5}$	$1.2400 \cdot 10^{-4}$
10 Systeme:	$4.1334 \cdot 10^{-6}$	$4.1334 \cdot 10^{-5}$
11 Systeme:	$1.2400 \cdot 10^{-6}$	$1.3640 \cdot 10^{-5}$
12 Systeme:	$3.7200 \cdot 10^{-7}$	$4.4641 \cdot 10^{-6}$
13 Systeme:	$1.1160 \cdot 10^{-7}$	$1.4508 \cdot 10^{-6}$
14 Systeme:	$3.3480 \cdot 10^{-8}$	$4.6873 \cdot 10^{-7}$
15 Systeme:	$1.0044 \cdot 10^{-8}$	$1.5066 \cdot 10^{-7}$
16 Systeme:	$3.0132 \cdot 10^{-9}$	$4.8212 \cdot 10^{-8}$
17 Systeme:	$9.0398 \cdot 10^{-10}$	$1.5367 \cdot 10^{-8}$
18 Systeme:	$2.7119 \cdot 10^{-10}$	$4.8814 \cdot 10^{-9}$
19 Systeme:	$8.1358 \cdot 10^{-11}$	$1.5458 \cdot 10^{-9}$
20 Systeme:	$2.4407 \cdot 10^{-11}$	$4.8814 \cdot 10^{-10}$
21 Systeme:	$7.3222 \cdot 10^{-12}$	$1.5376 \cdot 10^{-10}$
22 Systeme:	$2.1966 \cdot 10^{-12}$	$4.8326 \cdot 10^{-11}$
23 Systeme:	$6.5900 \cdot 10^{-13}$	$1.5157 \cdot 10^{-11}$
24 Systeme:	$1.9770 \cdot 10^{-13}$	$4.7448 \cdot 10^{-12}$
25 Systeme:	$5.9310E-14 \cdot 10^{-14}$	$1.4827 \cdot 10^{-12}$
26 Systeme:	$1.7793 \cdot 10^{-14}$	$4.6261 \cdot 10^{-13}$
27 Systeme:	$5.3379 \cdot 10^{-15}$	$1.4412 \cdot 10^{-13}$
28 Systeme:	$1.6013 \cdot 10^{-15}$	$4.4838 \cdot 10^{-14}$
29 Systeme:	$4.8041 \cdot 10^{-16}$	$1.3931 \cdot 10^{-14}$
30 Systeme:	$2.0589E-16 \cdot 10^{-16}$	$6.1767 \cdot 10^{-15}$
SUMME:	**1.0000**	**0.4285**

Tabelle 32: Berechnung des Erwartungswertes für sieben Signaturen.

10.8 Erwartungswert bei acht (8) Signaturen

Kompromittierte Systeme (x)	**Wahrscheinlichkeit**	**Erwartungswert**
0 Systeme:	0.8000	0.0
1 System:	0.1600	0.1600
2 Systeme:	$3.2000 \cdot 10^{-2}$	$6.4000 \cdot 10^{-2}$
3 Systeme:	$6.4000 \cdot 10^{-3}$	$1.9200 \cdot 10^{-2}$
4 Systeme:	$1.2800 \cdot 10^{-3}$	$5.1200 \cdot 10^{-3}$
5 Systeme:	$2.5600 \cdot 10^{-4}$	$1.2800 \cdot 10^{-3}$
6 Systeme:	$5.1200 \cdot 10^{-5}$	$3.0720 \cdot 10^{-4}$
7 Systeme:	$1.0240 \cdot 10^{-5}$	$7.1680 \cdot 10^{-5}$
8 Systeme:	$2.0480 \cdot 10^{-6}$	$1.6384 \cdot 10^{-5}$
9 Systeme:	$4.0960 \cdot 10^{-7}$	$3.6864 \cdot 10^{-6}$
10 Systeme:	$8.1920 \cdot 10^{-8}$	$8.1920 \cdot 10^{-7}$
11 Systeme:	$1.6384 \cdot 10^{-8}$	$1.8022 \cdot 10^{-7}$
12 Systeme:	$3.2768 \cdot 10^{-9}$	$3.9321 \cdot 10^{-8}$
13 Systeme:	$6.5536 \cdot 10^{-10}$	$8.5196 \cdot 10^{-9}$
14 Systeme:	$1.3107 \cdot 10^{-10}$	$1.8350 \cdot 10^{-9}$
15 Systeme:	$2.6214 \cdot 10^{-11}$	$3.9321 \cdot 10^{-10}$
16 Systeme:	$5.2428 \cdot 10^{-12}$	$8.3886 \cdot 10^{-11}$
17 Systeme:	$1.0485 \cdot 10^{-12}$	$1.7825 \cdot 10^{-11}$
18 Systeme:	$2.0971 \cdot 10^{-13}$	$3.7748 \cdot 10^{-12}$
19 Systeme:	$4.1943 \cdot 10^{-14}$	$7.9691 \cdot 10^{-13}$
20 Systeme:	$8.3886 \cdot 10^{-15}$	$1.6777 \cdot 10^{-13}$
21 Systeme:	$1.6777 \cdot 10^{-15}$	$3.5232 \cdot 10^{-14}$
22 Systeme:	$3.3554 \cdot 10^{-16}$	$7.3819 \cdot 10^{-15}$
23 Systeme:	$6.7108 \cdot 10^{-17}$	$1.5435 \cdot 10^{-15}$
24 Systeme:	$1.3421 \cdot 10^{-17}$	$3.2212 \cdot 10^{-16}$
25 Systeme:	$2.6843 \cdot 10^{-18}$	$6.7108 \cdot 10^{-17}$
26 Systeme:	$5.3687 \cdot 10^{-19}$	$1.3958 \cdot 10^{-17}$
27 Systeme:	$1.0737 \cdot 10^{-19}$	$2.8991 \cdot 10^{-18}$
28 Systeme:	$2.1474 \cdot 10^{-20}$	$6.0129 \cdot 10^{-19}$
29 Systeme:	$4.2949 \cdot 10^{-21}$	$1.2455 \cdot 10^{-19}$
30 Systeme:	$1.0737 \cdot 10^{-21}$	$3.2212 \cdot 10^{-20}$
SUMME:	**1.0000**	**0.2500**

Tabelle 33: Berechnung des Erwartungswertes für acht Signaturen.

10.9 Erwartungswert bei neun (9) Signaturen

Kompromittierte Systeme (x)	**Wahrscheinlichkeit**	**Erwartungswert**
0 Systeme:	0.9000	0.0
1 System:	$9.0000 \cdot 10^{-2}$	$9.0000 \cdot 10^{-2}$
2 Systeme:	$9.0000 \cdot 10^{-3}$	$1.8000 \cdot 10^{-2}$
3 Systeme:	$9.0000 \cdot 10^{-4}$	$2.7000 \cdot 10^{-3}$
4 Systeme:	$9.0000 \cdot 10^{-5}$	$3.6000 \cdot 10^{-4}$
5 Systeme:	$9.0000 \cdot 10^{-6}$	$4.5000 \cdot 10^{-5}$
6 Systeme:	$9.0000 \cdot 10^{-7}$	$5.4000 \cdot 10^{-6}$
7 Systeme:	$9.0000 \cdot 10^{-8}$	$6.3000 \cdot 10^{-7}$
8 Systeme:	$9.0000 \cdot 10^{-9}$	$7.2000 \cdot 10^{-8}$
9 Systeme:	$9.0000 \cdot 10^{-10}$	$8.1000 \cdot 10^{-9}$
10 Systeme:	$9.0000 \cdot 10^{-11}$	$9.0000 \cdot 10^{-10}$
11 Systeme:	$9.0000 \cdot 10^{-12}$	$9.9000 \cdot 10^{-11}$
12 Systeme:	$9.0000 \cdot 10^{-13}$	$1.0800 \cdot 10^{-11}$
13 Systeme:	$9.0000 \cdot 10^{-14}$	$1.1700 \cdot 10^{-12}$
14 Systeme:	$9.0000 \cdot 10^{-15}$	$1.2600 \cdot 10^{-13}$
15 Systeme:	$9.0000 \cdot 10^{-16}$	$1.3500 \cdot 10^{-14}$
16 Systeme:	$9.0000 \cdot 10^{-17}$	$1.4400 \cdot 10^{-15}$
17 Systeme:	$9.0000 \cdot 10^{-18}$	$1.5300 \cdot 10^{-16}$
18 Systeme:	$9.0000 \cdot 10^{-19}$	$1.6200 \cdot 10^{-17}$
19 Systeme:	$9.0000 \cdot 10^{-20}$	$1.7100 \cdot 10^{-18}$
20 Systeme:	$9.0000 \cdot 10^{-21}$	$1.8000 \cdot 10^{-19}$
21 Systeme:	$9.0000 \cdot 10^{-22}$	$1.8900 \cdot 10^{-20}$
22 Systeme:	$9.0000 \cdot 10^{-23}$	$1.9800 \cdot 10^{-21}$
23 Systeme:	$9.0000 \cdot 10^{-24}$	$2.0700 \cdot 10^{-22}$
24 Systeme:	$9.0000 \cdot 10^{-25}$	$2.1600 \cdot 10^{-23}$
25 Systeme:	$9.0000 \cdot 10^{-26}$	$2.2500 \cdot 10^{-24}$
26 Systeme:	$9.0000 \cdot 10^{-27}$	$2.3400 \cdot 10^{-25}$
27 Systeme:	$9.0000 \cdot 10^{-28}$	$2.4300 \cdot 10^{-26}$
28 Systeme:	$9.0000 \cdot 10^{-29}$	$2.5200 \cdot 10^{-27}$
29 Systeme:	$9.0000 \cdot 10^{-30}$	$2.6100 \cdot 10^{-28}$
30 Systeme:	$1.0000 \cdot 10^{-30}$	$3.0000 \cdot 10^{-29}$
SUMME:	**1.0000**	**0.1111**

Tabelle 34: Berechnung des Erwartungswertes für neun Signaturen.

10.10 Erwartungswert bei zehn (10) Signaturen

Bei zehn (10) Signaturen beträgt der Erwartungswert 0 (null), da jeder Angriff vom Agenten abgewehrt werden kann.

10.11 Erwartungswert bei null (0) Signaturen

Da der Agent in diesem Fall keinen Schutz bietet, wären nach dreißig (30) Angriffen dreißig (30) Systeme kompromittiert. (Erwartungswert beträgt dreißig (30) Systeme.)

11 Anhang 4 (Kompromittierungswahrscheinlichkeiten)

11.1 Kompromittierungswahrscheinlichkeiten bei einer (1) Signatur

Die in den folgenden Tabellen dargestellten Ergebnisse wurden nach der vierten wesentlichen Nachkommastelle abgeschnitten.

n/P(w)[100]	10%	30%	50%	70%	90%
1	0.9000	0.9000	0.9000	0.9000	0.9000
2	0.8910	0.8730	0.8550	0.8370	0.8190
3	0.8812	0.8397	0.7930	0.7414	0.6855
4	0.8708	0.7993	0.7109	0.6073	0.4915
5	0.8595	0.7512	0.6081	0.4403	0.2666
6	0.8475	0.6951	0.4890	0.2678	$9.0654 \cdot 10^{-2}$
7	0.8345	0.6316	0.3641	0.1305	$1.6461 \cdot 10^{-2}$
8	0.8207	0.5618	0.2483	$5.1115 \cdot 10^{-2}$	$1.8900 \cdot 10^{-3}$
9	0.8060	0.4879	0.1550	$1.7163 \cdot 10^{-2}$	$1.9222 \cdot 10^{-4}$
10	0.7904	0.4130	$8.9514 \cdot 10^{-2}$	$5.3552 \cdot 10^{-3}$	$1.9255 \cdot 10^{-5}$
11	0.7738	0.3402	$4.8763 \cdot 10^{-2}$	$1.6266 \cdot 10^{-3}$	$1.9259 \cdot 10^{-6}$
12	0.7563	0.2729	$2.5570 \cdot 10^{-2}$	$4.8985 \cdot 10^{-4}$	$1.9259 \cdot 10^{-7}$
13	0.7379	0.2133	$1.3112 \cdot 10^{-2}$	$1.4712 \cdot 10^{-4}$	$1.9259 \cdot 10^{-8}$
14	0.7186	0.1630	$6.6421 \cdot 10^{-3}$	$4.4152 \cdot 10^{-5}$	$1.9259 \cdot 10^{-9}$
15	0.6983	0.1221	$3.3431 \cdot 10^{-3}$	$1.3246 \cdot 10^{-5}$	$1.9259 \cdot 10^{-10}$
16	0.6773	$8.9944 \cdot 10^{-2}$	$1.6771 \cdot 10^{-3}$	$3.9742 \cdot 10^{-6}$	$1.9259 \cdot 10^{-11}$
17	0.6554	$6.5388 \cdot 10^{-2}$	$8.3998 \cdot 10^{-4}$	$1.1922 \cdot 10^{-6}$	$1.9259 \cdot 10^{-12}$
18	0.6328	$4.7054 \cdot 10^{-2}$	$4.2034 \cdot 10^{-4}$	$3.5768 \cdot 10^{-7}$	$1.9259 \cdot 10^{-13}$
19	0.6096	$3.3602 \cdot 10^{-2}$	$2.1025 \cdot 10^{-4}$	$1.0730 \cdot 10^{-7}$	$1.9259 \cdot 10^{-14}$
20	0.5858	$2.3860 \cdot 10^{-2}$	$1.0515 \cdot 10^{-4}$	$3.2191 \cdot 10^{-8}$	$1.9259 \cdot 10^{-15}$
21	0.5615	$1.6873 \cdot 10^{-2}$	$5.2581 \cdot 10^{-5}$	$9.6574 \cdot 10^{-9}$	$1.9259 \cdot 10^{-16}$
22	0.5369	$1.1896 \cdot 10^{-2}$	$2.6292 \cdot 10^{-5}$	$2.8972 \cdot 10^{-9}$	$1.9259 \cdot 10^{-17}$
23	0.5121	$8.3700 \cdot 10^{-3}$	$1.3146 \cdot 10^{-5}$	$8.6917 \cdot 10^{-10}$	$1.9259 \cdot 10^{-18}$
24	0.4871	$5.8800 \cdot 10^{-3}$	$6.5732 \cdot 10^{-6}$	$2.6075 \cdot 10^{-10}$	$1.9259 \cdot 10^{-19}$
25	0.4621	$4.1264 \cdot 10^{-3}$	$3.2866 \cdot 10^{-6}$	$7.8225 \cdot 10^{-11}$	$1.9259 \cdot 10^{-20}$
26	0.4372	$2.8935 \cdot 10^{-3}$	$1.6433 \cdot 10^{-6}$	$2.3467 \cdot 10^{-11}$	$1.9259 \cdot 10^{-21}$
27	0.4126	$2.0280 \cdot 10^{-3}$	$8.2167 \cdot 10^{-7}$	$7.0402 \cdot 10^{-12}$	$1.9259 \cdot 10^{-22}$
28	0.3884	$1.4208 \cdot 10^{-3}$	$4.1083 \cdot 10^{-7}$	$2.1120 \cdot 10^{-12}$	$1.9259 \cdot 10^{-23}$
29	0.3646	$9,9520 \cdot 10^{-4}$	$2.0541 \cdot 10^{-7}$	$6.3362 \cdot 10^{-13}$	$1.9259 \cdot 10^{-24}$
30	0.3415	$6,9694 \cdot 10^{-4}$	$1.0270 \cdot 10^{-7}$	$1.9008 \cdot 10^{-13}$	$1.9259 \cdot 10^{-25}$

Tabelle 35: Kompromittierungswahrscheinlichkeiten bei einer Signatur.

[100] n - Nummer des Angriffs; P(w) - Wahrscheinlichkeit des Empfangs einer Warnmeldung.

11.2 Kompromittierungswahrscheinlichkeiten bei zwei (2) Signaturen

n/P(w)[101]	10%	30%	50%	70%	90%
1	0.8000	0.8000	0.8000	0.8000	0.8000
2	0.7840	0.7520	0.7200	0.6880	0.6560
3	0.7670	0.6960	0.6192	0.5377	0.4529
4	0.7491	0.6325	0.5013	0.3637	0.2298
5	0.7304	0.5628	0.3763	0.2017	$7.0557 \cdot 10^{-2}$
6	0.7107	0.4890	0.2589	$8.9008 \cdot 10^{-2}$	$1.1536 \cdot 10^{-2}$
7	0.6901	0.4140	0.1630	$3.2248 \cdot 10^{-2}$	$1.2734 \cdot 10^{-3}$
8	0.6687	0.3412	$9.4788 \cdot 10^{-2}$	$1.0402 \cdot 10^{-2}$	$1.2880 \cdot 10^{-4}$
9	0.6466	0.2738	$5.1886 \cdot 10^{-2}$	$3.1964 \cdot 10^{-3}$	$1.2895 \cdot 10^{-5}$
10	0.6237	0.2141	$2.7289 \cdot 10^{-2}$	$9.6609 \cdot 10^{-4}$	$1.2896 \cdot 10^{-6}$
11	0.6003	0.1636	$1.4017 \cdot 10^{-2}$	$2.9048 \cdot 10^{-4}$	$1.2896 \cdot 10^{-7}$
12	0.5763	0.1226	$7.1068 \cdot 10^{-3}$	$8.7203 \cdot 10^{-5}$	$1.2896 \cdot 10^{-8}$
13	0.5518	$9.0350 \cdot 10^{-2}$	$3.5786 \cdot 10^{-3}$	$2.6166 \cdot 10^{-5}$	$1.2896 \cdot 10^{-9}$
14	0.5271	$6.5694 \cdot 10^{-2}$	$1.7957 \cdot 10^{-3}$	$7.8504 \cdot 10^{-6}$	$1.2896 \cdot 10^{-10}$
15	0.5022	$4.7280 \cdot 10^{-2}$	$8.9947 \cdot 10^{-4}$	$2.3551 \cdot 10^{-6}$	$1.2896 \cdot 10^{-11}$
16	0.4772	$3.3767 \cdot 10^{-2}$	$4.5014 \cdot 10^{-4}$	$7.0655 \cdot 10^{-7}$	$1.2896 \cdot 10^{-12}$
17	0.4522	$2.3979 \cdot 10^{-2}$	$2.2517 \cdot 10^{-4}$	$2.1196 \cdot 10^{-7}$	$1.2896 \cdot 10^{-13}$
18	0.4275	$1.6957 \cdot 10^{-2}$	$1.1261 \cdot 10^{-4}$	$6.3590 \cdot 10^{-8}$	$1.2896 \cdot 10^{-14}$
19	0.4030	$1.1956 \cdot 10^{-2}$	$5.6312 \cdot 10^{-5}$	$1.9077 \cdot 10^{-8}$	$1.2896 \cdot 10^{-15}$
20	0.3789	$8.4126 \cdot 10^{-3}$	$2.8157 \cdot 10^{-5}$	$5.7231 \cdot 10^{-9}$	$1.2896 \cdot 10^{-16}$
21	0.3554	$5.9100 \cdot 10^{-3}$	$1.4079 \cdot 10^{-5}$	$1.7169 \cdot 10^{-9}$	$1.2896 \cdot 10^{-17}$
22	0.3325	$4.1475 \cdot 10^{-3}$	$7.0397 \cdot 10^{-6}$	$5.1507 \cdot 10^{-10}$	$1.2896 \cdot 10^{-18}$
23	0.3103	$2.9084 \cdot 10^{-3}$	$3.5198 \cdot 10^{-6}$	$1.5452 \cdot 10^{-10}$	$1.2896 \cdot 10^{-19}$
24	0.2889	$2.0384 \cdot 10^{-3}$	$1.7599 \cdot 10^{-6}$	$4.6357 \cdot 10^{-11}$	$1.2896 \cdot 10^{-20}$
25	0.2683	$1.4281 \cdot 10^{-3}$	$8.7997 \cdot 10^{-7}$	$1.3907 \cdot 10^{-11}$	$1.2896 \cdot 10^{-21}$
26	0.2487	$1.0003 \cdot 10^{-3}$	$4.3998 \cdot 10^{-7}$	$4.1721 \cdot 10^{-12}$	$1.2896 \cdot 10^{-22}$
27	0.2300	$7.0052 \cdot 10^{-4}$	$2.1999 \cdot 10^{-7}$	$1.2516 \cdot 10^{-12}$	$1.2896 \cdot 10^{-23}$
28	0.2123	$4.9051 \cdot 10^{-4}$	$1.0999 \cdot 10^{-7}$	$3.7549 \cdot 10^{-13}$	$1.2896 \cdot 10^{-24}$
29	0.1956	$3.4343 \cdot 10^{-4}$	$5.4998 \cdot 10^{-8}$	$1.1264 \cdot 10^{-13}$	$1.2896 \cdot 10^{-25}$
30	0.1798	$2.4043 \cdot 10^{-4}$	$2.7499 \cdot 10^{-8}$	$3.3794 \cdot 10^{-14}$	$1.2896 \cdot 10^{-26}$

Tabelle 36: Kompromittierungswahrscheinlichkeiten bei zwei Signaturen.

[101] n - Nummer des Angriffs; P(w) - Wahrscheinlichkeit des Empfangs einer Warnmeldung.

11.3 Kompromittierungswahrscheinlichkeiten bei drei (3) Signaturen

n/P(w)[102]	10%	30%	50%	70%	90%
1	0.7000	0.7000	0.7000	0.7000	0.7000
2	0.6789	0.6369	0.5950	0.5529	0.5110
3	0.6572	0.5676	0.4745	0.3799	0.2861
4	0.6346	0.4940	0.3498	0.2150	0.1022
5	0.6114	0.4190	0.2361	$9.6888 \cdot 10^{-2}$	$1.9644 \cdot 10^{-2}$
6	0.5877	0.3459	0.1459	$3.5637 \cdot 10^{-2}$	$2.3117 \cdot 10^{-3}$
7	0.5635	0.2780	$8.3612 \cdot 10^{-2}$	$1.1580 \cdot 10^{-2}$	$2.3597 \cdot 10^{-4}$
8	0.5389	0.2178	$4.5302 \cdot 10^{-2}$	$3.5679 \cdot 10^{-3}$	$2.3648 \cdot 10^{-5}$
9	0.5140	0.1667	$2.3677 \cdot 10^{-2}$	$1.0793 \cdot 10^{-3}$	$2.3653 \cdot 10^{-6}$
10	0.4890	0.1250	$1.2118 \cdot 10^{-2}$	$3.2460 \cdot 10^{-4}$	$2.3653 \cdot 10^{-7}$
11	0.4640	$9.2238 \cdot 10^{-2}$	$6.1328 \cdot 10^{-3}$	$9.7456 \cdot 10^{-5}$	$2.3653 \cdot 10^{-8}$
12	0.4392	$6.7119 \cdot 10^{-2}$	$3.0852 \cdot 10^{-3}$	$2.9243 \cdot 10^{-5}$	$2.3653 \cdot 10^{-9}$
13	0.4145	$4.8335 \cdot 10^{-2}$	$1.5473 \cdot 10^{-3}$	$8.7736 \cdot 10^{-6}$	$2.3653 \cdot 10^{-10}$
14	0.3903	$3.4535 \cdot 10^{-2}$	$7.7488 \cdot 10^{-4}$	$2.6321 \cdot 10^{-6}$	$2.3653 \cdot 10^{-11}$
15	0.3665	$2.4532 \cdot 10^{-2}$	$3.8774 \cdot 10^{-4}$	$7.8964 \cdot 10^{-7}$	$2.3653 \cdot 10^{-12}$
16	0.3433	$1.7353 \cdot 10^{-2}$	$1.9394 \cdot 10^{-4}$	$2.3689 \cdot 10^{-7}$	$2.3653 \cdot 10^{-13}$
17	0.3207	$1.2237 \cdot 10^{-2}$	$9.6992 \cdot 10^{-5}$	$7.1068 \cdot 10^{-8}$	$2.3653 \cdot 10^{-14}$
18	0.2989	$8.6113 \cdot 10^{-3}$	$4.8500 \cdot 10^{-5}$	$2.1320 \cdot 10^{-8}$	$2.3653 \cdot 10^{-15}$
19	0.2780	$6.0501 \cdot 10^{-3}$	$2.4251 \cdot 10^{-5}$	$6.3961 \cdot 10^{-9}$	$2.3653 \cdot 10^{-16}$
20	0.2579	$4.2461 \cdot 10^{-3}$	$1.2126 \cdot 10^{-5}$	$1.9188 \cdot 10^{-9}$	$2.3653 \cdot 10^{-17}$
21	0.2387	$2.9776 \cdot 10^{-3}$	$6.0631 \cdot 10^{-6}$	$5.7565 \cdot 10^{-10}$	$2.3653 \cdot 10^{-18}$
22	0.2206	$2.0870 \cdot 10^{-3}$	$3.0315 \cdot 10^{-6}$	$1.7269 \cdot 10^{-10}$	$2.3653 \cdot 10^{-19}$
23	0.2034	$1.4622 \cdot 10^{-3}$	$1.5157 \cdot 10^{-6}$	$5.1808 \cdot 10^{-11}$	$2.3653 \cdot 10^{-20}$
24	0.1872	$1.0242 \cdot 10^{-3}$	$7.5789 \cdot 10^{-7}$	$1.5542 \cdot 10^{-11}$	$2.3653 \cdot 10^{-21}$
25	0.1720	$7.1725 \cdot 10^{-4}$	$3.7894 \cdot 10^{-7}$	$4.6628 \cdot 10^{-12}$	$2.3653 \cdot 10^{-22}$
26	0.1577	$5.0223 \cdot 10^{-4}$	$1.8947 \cdot 10^{-7}$	$1.3988 \cdot 10^{-12}$	$2.3653 \cdot 10^{-23}$
27	0.1444	$3.5164 \cdot 10^{-4}$	$9.4737 \cdot 10^{-8}$	$4.1965 \cdot 10^{-13}$	$2.3653 \cdot 10^{-24}$
28	0.1321	$2.4618 \cdot 10^{-4}$	$4.7368 \cdot 10^{-8}$	$1.2589 \cdot 10^{-13}$	$2.3653 \cdot 10^{-25}$
29	0.1206	$1.7234 \cdot 10^{-4}$	$2.3684 \cdot 10^{-8}$	$3.7768 \cdot 10^{-14}$	$2.3653 \cdot 10^{-26}$
30	0.1100	$1.2065 \cdot 10^{-4}$	$1.1842 \cdot 10^{-8}$	$1.1330 \cdot 10^{-14}$	$2.3653 \cdot 10^{-27}$

Tabelle 37: Kompromittierungswahrscheinlichkeiten bei drei Signaturen.

[102] n - Nummer des Angriffs; P(w) - Wahrscheinlichkeit des Empfangs einer Warnmeldung.

11.4 Kompromittierungswahrscheinlichkeiten bei vier (4) Signaturen

n/P(w)[103]	10%	30%	50%	70%	90%
1	0.6000	0.6000	0.6000	0.6000	0.6000
2	0.5760	0.5279	0.4800	0.4320	0.3840
3	0.5515	0.4532	0.3551	0.2602	0.1711
4	0.5268	0.3788	0.2406	0.1254	$4.3461 \cdot 10^{-2}$
5	0.5019	0.3082	0.1493	$4.8664 \cdot 10^{-2}$	$6.0462 \cdot 10^{-3}$
6	0.4769	0.2443	$8.5799 \cdot 10^{-2}$	$1.6257 \cdot 10^{-2}$	$6.3752 \cdot 10^{-4}$
7	0.4519	0.1889	$4.6580 \cdot 10^{-2}$	$5.0621 \cdot 10^{-3}$	$6.4118 \cdot 10^{-5}$
8	0.4272	0.1429	$2.4375 \cdot 10^{-2}$	$1.5365 \cdot 10^{-3}$	$6.4155 \cdot 10^{-6}$
9	0.4027	0.1062	$1.2484 \cdot 10^{-2}$	$4.6262 \cdot 10^{-4}$	$6.4159 \cdot 10^{-7}$
10	0.3786	$7.7725 \cdot 10^{-2}$	$6.3202 \cdot 10^{-3}$	$1.3893 \cdot 10^{-4}$	$6.4159 \cdot 10^{-8}$
11	0.3551	$5.6220 \cdot 10^{-2}$	$3.1800 \cdot 10^{-3}$	$4.1694 \cdot 10^{-5}$	$6.4159 \cdot 10^{-9}$
12	0.3322	$4.0302 \cdot 10^{-2}$	$1.5950 \cdot 10^{-3}$	$1.2509 \cdot 10^{-5}$	$6.4159 \cdot 10^{-10}$
13	0.3100	$2.8698 \cdot 10^{-2}$	$7.9882 \cdot 10^{-4}$	$3.7529 \cdot 10^{-6}$	$6.4159 \cdot 10^{-11}$
14	0.2886	$2.0336 \cdot 10^{-2}$	$3.9972 \cdot 10^{-4}$	$1.1259 \cdot 10^{-6}$	$6.4159 \cdot 10^{-12}$
15	0.2681	$1.4359 \cdot 10^{-2}$	$1.9994 \cdot 10^{-4}$	$3.3777 \cdot 10^{-7}$	$6.4159 \cdot 10^{-13}$
16	0.2485	$1.0113 \cdot 10^{-2}$	$9.9992 \cdot 10^{-5}$	$1.0133 \cdot 10^{-8}$	$6.4159 \cdot 10^{-14}$
17	0.2298	$7.1101 \cdot 10^{-3}$	$5.0001 \cdot 10^{-5}$	$3.0399 \cdot 10^{-8}$	$6.4159 \cdot 10^{-15}$
18	0.2121	$4.9922 \cdot 10^{-3}$	$2.5001 \cdot 10^{-5}$	$9.1199 \cdot 10^{-9}$	$6.4159 \cdot 10^{-16}$
19	0.1954	$3.5020 \cdot 10^{-3}$	$1.2501 \cdot 10^{-5}$	$2.7359 \cdot 10^{-9}$	$6.4159 \cdot 10^{-17}$
20	0.1796	$2.4551 \cdot 10^{-3}$	$6.2506 \cdot 10^{-6}$	$8.2079 \cdot 10^{-10}$	$6.4159 \cdot 10^{-18}$
21	0.1649	$1.7203 \cdot 10^{-3}$	$3.1253 \cdot 10^{-6}$	$2.4623 \cdot 10^{-10}$	$6.4159 \cdot 10^{-19}$
22	0.1511	$1.2051 \cdot 10^{-3}$	$1.5626 \cdot 10^{-6}$	$7.3871 \cdot 10^{-11}$	$6.4159 \cdot 10^{-20}$
23	0.1383	$8.4404 \cdot 10^{-4}$	$7.8134 \cdot 10^{-7}$	$2.2161 \cdot 10^{-11}$	$6.4159 \cdot 10^{-21}$
24	0.1264	$5.9104 \cdot 10^{-4}$	$3.9067 \cdot 10^{-7}$	$6.6484 \cdot 10^{-12}$	$6.4159 \cdot 10^{-22}$
25	0.1153	$4.1383 \cdot 10^{-4}$	$1.9533 \cdot 10^{-7}$	$1.9945 \cdot 10^{-12}$	$6.4159 \cdot 10^{-23}$
26	0.1051	$2.8973 \cdot 10^{-4}$	$9.7668 \cdot 10^{-8}$	$5.9835 \cdot 10^{-13}$	$6.4159 \cdot 10^{-24}$
27	$9.5765 \cdot 10^{-2}$	$2.0284 \cdot 10^{-4}$	$4.8834 \cdot 10^{-8}$	$1.7950 \cdot 10^{-13}$	$6.4159 \cdot 10^{-25}$
28	$8.7106 \cdot 10^{-2}$	$1.4200 \cdot 10^{-4}$	$2.4417 \cdot 10^{-8}$	$5.3852 \cdot 10^{-14}$	$6.4159 \cdot 10^{-26}$
29	$7.9154 \cdot 10^{-2}$	$9.9407 \cdot 10^{-5}$	$1.2208 \cdot 10^{-8}$	$1.6155 \cdot 10^{-14}$	$6.4159 \cdot 10^{-27}$
30	$7.1865 \cdot 10^{-2}$	$6.9588 \cdot 10^{-5}$	$6.1042 \cdot 10^{-9}$	$4.8466 \cdot 10^{-15}$	$6.4159 \cdot 10^{-28}$

Tabelle 38: Kompromittierungswahrscheinlichkeiten bei vier Signaturen.

[103] n - Nummer des Angriffs; P(w) - Wahrscheinlichkeit des Empfangs einer Warnmeldung.

11.5 Kompromittierungswahrscheinlichkeiten bei fünf (5) Signaturen

n/P(w)[104]	10%	30%	50%	70%	90%
1	0.5000	0.5000	0.5000	0.5000	0.5000
2	0.4750	0.4250	0.3750	0.3250	0.2750
3	0.4500	0.3516	0.2578	0.1714	$9.5562 \cdot 10^{-2}$
4	0.4253	0.2832	0.1621	$7.2004 \cdot 10^{-2}$	$1.7775 \cdot 10^{-2}$
5	0.4008	0.2223	$9.4214 \cdot 10^{-2}$	$2.5230 \cdot 10^{-2}$	$2.0618 \cdot 10^{-3}$
6	0.3768	0.1704	$5.1545 \cdot 10^{-2}$	$8.0148 \cdot 10^{-3}$	$2.1001 \cdot 10^{-4}$
7	0.3533	0.1280	$2.7101 \cdot 10^{-2}$	$2.4494 \cdot 10^{-3}$	$2.1041 \cdot 10^{-5}$
8	0.3305	$9.4569 \cdot 10^{-2}$	$1.3917 \cdot 10^{-2}$	$7.3902 \cdot 10^{-4}$	$2.1045 \cdot 10^{-6}$
9	0.3083	$6.8881 \cdot 10^{-2}$	$7.0557 \cdot 10^{-3}$	$2.2208 \cdot 10^{-4}$	$2.1045 \cdot 10^{-7}$
10	0.2870	$4.9640 \cdot 10^{-2}$	$3.5527 \cdot 10^{-3}$	$6.6661 \cdot 10^{-5}$	$2.1045 \cdot 10^{-8}$
11	0.2665	$3.5487 \cdot 10^{-2}$	$1.7827 \cdot 10^{-3}$	$2.0001 \cdot 10^{-5}$	$2.1045 \cdot 10^{-9}$
12	0.2470	$2.5219 \cdot 10^{-2}$	$8.9293 \cdot 10^{-4}$	$6.0007 \cdot 10^{-6}$	$2.1045 \cdot 10^{-10}$
13	0.2284	$1.7844 \cdot 10^{-2}$	$4.4686 \cdot 10^{-4}$	$1.8002 \cdot 10^{-6}$	$2.1045 \cdot 10^{-11}$
14	0.2108	$1.2586 \cdot 10^{-2}$	$2.2353 \cdot 10^{-4}$	$5.4007 \cdot 10^{-7}$	$2.1045 \cdot 10^{-12}$
15	0.1941	$8.8580 \cdot 10^{-3}$	$1.1179 \cdot 10^{-4}$	$1.6202 \cdot 10^{-7}$	$2.1045 \cdot 10^{-13}$
16	0.1785	$6.2241 \cdot 10^{-3}$	$5.5902 \cdot 10^{-5}$	$4.8606 \cdot 10^{-8}$	$2.1045 \cdot 10^{-14}$
17	0.1638	$4.3685 \cdot 10^{-3}$	$2.7952 \cdot 10^{-5}$	$1.4582 \cdot 10^{-8}$	$2.1045 \cdot 10^{-15}$
18	0.1501	$3.0637 \cdot 10^{-3}$	$1.3976 \cdot 10^{-5}$	$4.3746 \cdot 10^{-9}$	$2.1045 \cdot 10^{-16}$
19	0.1374	$2.1474 \cdot 10^{-3}$	$6.9884 \cdot 10^{-6}$	$1.3123 \cdot 10^{-9}$	$2.1045 \cdot 10^{-17}$
20	0.1255	$1.5045 \cdot 10^{-3}$	$3.4942 \cdot 10^{-6}$	$3.9371 \cdot 10^{-10}$	$2.1045 \cdot 10^{-18}$
21	0.1145	$1.0538 \cdot 10^{-3}$	$1.7471 \cdot 10^{-6}$	$1.1811 \cdot 10^{-10}$	$2.1045 \cdot 10^{-19}$
22	0.1044	$7.3804 \cdot 10^{-4}$	$8.7356 \cdot 10^{-7}$	$3.5434 \cdot 10^{-11}$	$2.1045 \cdot 10^{-20}$
23	$9.5075 \cdot 10^{-2}$	$5.1679 \cdot 10^{-4}$	$4.3678 \cdot 10^{-7}$	$1.0630 \cdot 10^{-11}$	$2.1045 \cdot 10^{-21}$
24	$8.6472 \cdot 10^{-2}$	$3.6183 \cdot 10^{-4}$	$2.1839 \cdot 10^{-7}$	$3.1891 \cdot 10^{-12}$	$2.1045 \cdot 10^{-22}$
25	$7.8572 \cdot 10^{-2}$	$2.5332 \cdot 10^{-4}$	$1.0919 \cdot 10^{-7}$	$9.5673 \cdot 10^{-13}$	$2.1045 \cdot 10^{-23}$
26	$7.1332 \cdot 10^{-2}$	$1.7734 \cdot 10^{-4}$	$5.4598 \cdot 10^{-8}$	$2.8701 \cdot 10^{-13}$	$2.1045 \cdot 10^{-24}$
27	$6.4708 \cdot 10^{-2}$	$1.2415 \cdot 10^{-4}$	$2.7299 \cdot 10^{-8}$	$8.6105 \cdot 10^{-14}$	$2.1045 \cdot 10^{-25}$
28	$5.8656 \cdot 10^{-2}$	$8.6911 \cdot 10^{-5}$	$1.3649 \cdot 10^{-8}$	$2.5831 \cdot 10^{-14}$	$2.1045 \cdot 10^{-26}$
29	$5.3134 \cdot 10^{-2}$	$6.0840 \cdot 10^{-5}$	$6.8247 \cdot 10^{-9}$	$7.7495 \cdot 10^{-15}$	$2.1045 \cdot 10^{-26}$
30	$4.8103 \cdot 10^{-2}$	$4.2589 \cdot 10^{-5}$	$3.4123 \cdot 10^{-9}$	$2.3248 \cdot 10^{-15}$	$2.1045 \cdot 10^{-27}$

Tabelle 39: Kompromittierungswahrscheinlichkeiten bei fünf Signaturen.

[104] n - Nummer des Angriffs; P(w) - Wahrscheinlichkeit des Empfangs einer Warnmeldung.

11.6 Kompromittierungswahrscheinlichkeiten bei sechs (6) Signaturen

n/P(w)[105]	10%	30%	50%	70%	90%
1	0.4000	0.4000	0.4000	0.4000	0.4000
2	0.3760	0.3280	0.2799	0.2320	0.1840
3	0.3525	0.2618	0.1791	0.1072	$4.8870 \cdot 10^{-2}$
4	0.3297	0.2038	0.1056	$4.0238 \cdot 10^{-2}$	$7.0365 \cdot 10^{-3}$
5	0.3076	0.1551	$5.8409 \cdot 10^{-2}$	$1.3205 \cdot 10^{-2}$	$7.4821 \cdot 10^{-4}$
6	0.2863	0.1158	$3.0910 \cdot 10^{-2}$	$4.0835 \cdot 10^{-3}$	$7.5325 \cdot 10^{-5}$
7	0.2658	$8.5128 \cdot 10^{-2}$	$1.5933 \cdot 10^{-2}$	$1.2367 \cdot 10^{-3}$	$7.5376 \cdot 10^{-6}$
8	0.2463	$6.1763 \cdot 10^{-2}$	$8.0934 \cdot 10^{-3}$	$3.7209 \cdot 10^{-4}$	$7.5381 \cdot 10^{-7}$
9	0.2277	$4.4379 \cdot 10^{-2}$	$4.0794 \cdot 10^{-3}$	$1.1172 \cdot 10^{-4}$	$7.5381 \cdot 10^{-8}$
10	0.2102	$3.1656 \cdot 10^{-2}$	$2.0480 \cdot 10^{-3}$	$3.3526 \cdot 10^{-5}$	$7.5381 \cdot 10^{-9}$
11	0.1936	$2.2460 \cdot 10^{-2}$	$1.0261 \cdot 10^{-3}$	$1.0058 \cdot 10^{-5}$	$7.5381 \cdot 10^{-10}$
12	0.1779	$1.5873 \cdot 10^{-2}$	$5.1359 \cdot 10^{-4}$	$3.0176 \cdot 10^{-6}$	$7.5381 \cdot 10^{-11}$
13	0.1633	$1.1186 \cdot 10^{-2}$	$2.5692 \cdot 10^{-4}$	$9.0531 \cdot 10^{-7}$	$7.5381 \cdot 10^{-12}$
14	0.1496	$7.8684 \cdot 10^{-3}$	$1.2849 \cdot 10^{-4}$	$2.7159 \cdot 10^{-7}$	$7.5381 \cdot 10^{-13}$
15	0.1369	$5.5264 \cdot 10^{-3}$	$6.4256 \cdot 10^{-5}$	$8.1478 \cdot 10^{-8}$	$7.5381 \cdot 10^{-14}$
16	0.1251	$3.8776 \cdot 10^{-3}$	$3.2130 \cdot 10^{-5}$	$2.4443 \cdot 10^{-8}$	$7.5381 \cdot 10^{-15}$
17	0.1141	$2.7188 \cdot 10^{-3}$	$1.6065 \cdot 10^{-6}$	$7.3330 \cdot 10^{-9}$	$7.5381 \cdot 10^{-16}$
18	0.1040	$1.9054 \cdot 10^{-3}$	$8.0330 \cdot 10^{-6}$	$2.1999 \cdot 10^{-9}$	$7.5381 \cdot 10^{-17}$
19	$9.4754 \cdot 10^{-2}$	$1.3348 \cdot 10^{-3}$	$4.0165 \cdot 10^{-6}$	$6.5997 \cdot 10^{-10}$	$7.5381 \cdot 10^{-18}$
20	$8.6176 \cdot 10^{-2}$	$9.3496 \cdot 10^{-4}$	$2.0082 \cdot 10^{-6}$	$1.9799 \cdot 10^{-10}$	$7.5381 \cdot 10^{-19}$
21	$7.8301 \cdot 10^{-2}$	$6.5473 \cdot 10^{-4}$	$1.0041 \cdot 10^{-6}$	$5.9397 \cdot 10^{-11}$	$7.5381 \cdot 10^{-20}$
22	$7.1084 \cdot 10^{-2}$	$4.5844 \cdot 10^{-4}$	$5.0207 \cdot 10^{-7}$	$1.7819 \cdot 10^{-11}$	$7.5381 \cdot 10^{-21}$
23	$6.4481 \cdot 10^{-2}$	$3.2097 \cdot 10^{-4}$	$2.5103 \cdot 10^{-7}$	$5.3457 \cdot 10^{-12}$	$7.5381 \cdot 10^{-22}$
24	$5.8449 \cdot 10^{-2}$	$2.2471 \cdot 10^{-4}$	$1.2551 \cdot 10^{-7}$	$1.6037 \cdot 10^{-12}$	$7.5381 \cdot 10^{-23}$
25	$5.2945 \cdot 10^{-2}$	$1.5731 \cdot 10^{-4}$	$6.2758 \cdot 10^{-8}$	$4.8112 \cdot 10^{-13}$	$7.5381 \cdot 10^{-24}$
26	$4.7931 \cdot 10^{-2}$	$1.1012 \cdot 10^{-4}$	$3.1379 \cdot 10^{-8}$	$1.4433 \cdot 10^{-13}$	$7.5381 \cdot 10^{-25}$
27	$4.3368 \cdot 10^{-2}$	$7.7092 \cdot 10^{-5}$	$1.5689 \cdot 10^{-8}$	$4.3300 \cdot 10^{-14}$	$7.5381 \cdot 10^{-26}$
28	$3.9219 \cdot 10^{-2}$	$5.3966 \cdot 10^{-5}$	$7.8448 \cdot 10^{-9}$	$1.2990 \cdot 10^{-14}$	$7.5381 \cdot 10^{-27}$
29	$3.5451 \cdot 10^{-2}$	$3.7777 \cdot 10^{-5}$	$3.9224 \cdot 10^{-9}$	$3.8970 \cdot 10^{-15}$	$7.5381 \cdot 10^{-28}$
30	$3.2031 \cdot 10^{-2}$	$2.6444 \cdot 10^{-5}$	$1.9612 \cdot 10^{-9}$	$1.1691 \cdot 10^{-15}$	$7.5381 \cdot 10^{-29}$

Tabelle 40: Kompromittierungswahrscheinlichkeiten bei sechs Signaturen.

[105] n - Nummer des Angriffs; P(w) - Wahrscheinlichkeit des Empfangs einer Warnmeldung.

11.7 Kompromittierungswahrscheinlichkeiten bei sieben (7) Signaturen

n/P(w)[106]	10%	30%	50%	70%	90%
1	0.3000	0.3000	0.3000	0.3000	0.3000
2	0.2790	0.2370	0.1950	0.1530	0.1110
3	0.2588	0.1827	0.1165	$6.2286 \cdot 10^{-2}$	$2.2188 \cdot 10^{-2}$
4	0.2396	0.1379	$6.5043 \cdot 10^{-2}$	$2.1401 \cdot 10^{-2}$	$2.6620 \cdot 10^{-3}$
5	0.2214	0.1022	$3.4637 \cdot 10^{-2}$	$6.7410 \cdot 10^{-3}$	$2.7257 \cdot 10^{-4}$
6	0.2042	$7.4726 \cdot 10^{-2}$	$1.7918 \cdot 10^{-2}$	$2.0541 \cdot 10^{-3}$	$2.7324 \cdot 10^{-5}$
7	0.1879	$5.3983 \cdot 10^{-2}$	$9.1197 \cdot 10^{-3}$	$6.1919 \cdot 10^{-4}$	$2.7331 \cdot 10^{-6}$
8	0.1727	$3.8663 \cdot 10^{-2}$	$4.6014 \cdot 10^{-3}$	$1.8602 \cdot 10^{-4}$	$2.7332 \cdot 10^{-7}$
9	0.1584	$2.7512 \cdot 10^{-2}$	$2.3113 \cdot 10^{-3}$	$5.5832 \cdot 10^{-5}$	$2.7332 \cdot 10^{-8}$
10	0.1450	$1.9485 \cdot 10^{-2}$	$1.1583 \cdot 10^{-3}$	$1.6751 \cdot 10^{-5}$	$2.7332 \cdot 10^{-9}$
11	0.1326	$1.3754 \cdot 10^{-2}$	$5.7983 \cdot 10^{-4}$	$5.0257 \cdot 10^{-6}$	$2.7332 \cdot 10^{-10}$
12	0.1211	$9.6845 \cdot 10^{-3}$	$2.9008 \cdot 10^{-4}$	$1.5077 \cdot 10^{-6}$	$2.7332 \cdot 10^{-11}$
13	0.1105	$6.8073 \cdot 10^{-3}$	$1.4508 \cdot 10^{-4}$	$4.5232 \cdot 10^{-7}$	$2.7332 \cdot 10^{-12}$
14	0.1007	$4.7790 \cdot 10^{-3}$	$7.2553 \cdot 10^{-5}$	$1.3569 \cdot 10^{-7}$	$2.7332 \cdot 10^{-13}$
15	$9.1644 \cdot 10^{-2}$	$3.3521 \cdot 10^{-3}$	$3.6279 \cdot 10^{-5}$	$4.0709 \cdot 10^{-8}$	$2.7332 \cdot 10^{-14}$
16	$8.3319 \cdot 10^{-2}$	$2.3498 \cdot 10^{-3}$	$1.8140 \cdot 10^{-5}$	$1.2212 \cdot 10^{-8}$	$2.7332 \cdot 10^{-15}$
17	$7.5681 \cdot 10^{-2}$	$1.6465 \cdot 10^{-3}$	$9.0703 \cdot 10^{-6}$	$3.6638 \cdot 10^{-9}$	$2.7332 \cdot 10^{-16}$
18	$6.8686 \cdot 10^{-2}$	$1.1534 \cdot 10^{-3}$	$4.5352 \cdot 10^{-6}$	$1.0991 \cdot 10^{-9}$	$2.7332 \cdot 10^{-17}$
19	$6.2289 \cdot 10^{-2}$	$8.0779 \cdot 10^{-4}$	$2.2676 \cdot 10^{-6}$	$3.2974 \cdot 10^{-10}$	$2.7332 \cdot 10^{-18}$
20	$5.6448 \cdot 10^{-2}$	$5.6565 \cdot 10^{-4}$	$1.1338 \cdot 10^{-6}$	$9.8923 \cdot 10^{-11}$	$2.7332 \cdot 10^{-19}$
21	$5.1122 \cdot 10^{-2}$	$3.9605 \cdot 10^{-4}$	$5.6690 \cdot 10^{-7}$	$2.9677 \cdot 10^{-11}$	$2.7332 \cdot 10^{-20}$
22	$4.6271 \cdot 10^{-2}$	$2.7728 \cdot 10^{-4}$	$2.8345 \cdot 10^{-7}$	$8.9031 \cdot 10^{-12}$	$2.7332 \cdot 10^{-21}$
23	$4.1858 \cdot 10^{-2}$	$1.9412 \cdot 10^{-4}$	$1.4172 \cdot 10^{-7}$	$2.6709 \cdot 10^{-12}$	$2.7332 \cdot 10^{-22}$
24	$3.7847 \cdot 10^{-2}$	$1.3589 \cdot 10^{-4}$	$7.0863 \cdot 10^{-8}$	$8.0128 \cdot 10^{-13}$	$2.7332 \cdot 10^{-23}$
25	$3.4206 \cdot 10^{-2}$	$9.5133 \cdot 10^{-5}$	$3.5431 \cdot 10^{-8}$	$2.4038 \cdot 10^{-13}$	$2.7332 \cdot 10^{-24}$
26	$3.0902 \cdot 10^{-2}$	$6.6595 \cdot 10^{-5}$	$1.7715 \cdot 10^{-8}$	$7.2115 \cdot 10^{-14}$	$2.7332 \cdot 10^{-25}$
27	$2.7907 \cdot 10^{-2}$	$4.6618 \cdot 10^{-5}$	$8.8578 \cdot 10^{-9}$	$2.1634 \cdot 10^{-14}$	$2.7332 \cdot 10^{-26}$
28	$2.5195 \cdot 10^{-2}$	$3.2633 \cdot 10^{-5}$	$4.4289 \cdot 10^{-9}$	$6.4903 \cdot 10^{-15}$	$2.7332 \cdot 10^{-27}$
29	$2.2738 \cdot 10^{-2}$	$2.2843 \cdot 10^{-5}$	$2.2144 \cdot 10^{-9}$	$1.9471 \cdot 10^{-15}$	$2.7332 \cdot 10^{-28}$
30	$2.0516 \cdot 10^{-2}$	$1.5990 \cdot 10^{-5}$	$1.1072 \cdot 10^{-9}$	$5.8413 \cdot 10^{-16}$	$2.7332 \cdot 10^{-29}$

Tabelle 41: Kompromittierungswahrscheinlichkeiten bei sieben Signaturen.

[106] n - Nummer des Angriffs; P(w) - Wahrscheinlichkeit des Empfangs einer Warnmeldung.

11.8 Kompromittierungswahrscheinlichkeiten bei acht (8) Signaturen

n/P(w)[107]	10%	30%	50%	70%	90%
1	0.1999	0.1999	0.1999	0.1999	0.1999
2	0.1839	0.1519	0.1199	$8.7999 \cdot 10^{-2}$	$5.5999 \cdot 10^{-2}$
3	0.1689	0.1133	$6.7199 \cdot 10^{-2}$	$3.1820 \cdot 10^{-2}$	$8.4223 \cdot 10^{-3}$
4	0.1549	0.0831	$3.5857 \cdot 10^{-2}$	$1.0255 \cdot 10^{-2}$	$9.0608 \cdot 10^{-4}$
5	0.1418	$6.0305 \cdot 10^{-2}$	$1.8571 \cdot 10^{-2}$	$3.1501 \cdot 10^{-3}$	$9.1347 \cdot 10^{-5}$
6	0.1296	$4.3304 \cdot 10^{-2}$	$9.4583 \cdot 10^{-3}$	$9.5198 \cdot 10^{-4}$	$9.1422 \cdot 10^{-6}$
7	0.1183	$3.0875 \cdot 10^{-2}$	$4.7739 \cdot 10^{-3}$	$2.8622 \cdot 10^{-4}$	$9.1429 \cdot 10^{-7}$
8	0.1079	$2.1899 \cdot 10^{-2}$	$2.3983 \cdot 10^{-3}$	$8.5926 \cdot 10^{-5}$	$9.1430 \cdot 10^{-8}$
9	$9.8322 \cdot 10^{-2}$	$1.5473 \cdot 10^{-2}$	$1.2020 \cdot 10^{-3}$	$2.5783 \cdot 10^{-5}$	$9.1430 \cdot 10^{-9}$
10	$8.9457 \cdot 10^{-2}$	$1.0903 \cdot 10^{-2}$	$6.0174 \cdot 10^{-4}$	$7.7353 \cdot 10^{-6}$	$9.1430 \cdot 10^{-10}$
11	$8.1311 \cdot 10^{-2}$	$7.6678 \cdot 10^{-3}$	$3.0105 \cdot 10^{-4}$	$2.3206 \cdot 10^{-6}$	$9.1430 \cdot 10^{-11}$
12	$7.3841 \cdot 10^{-2}$	$5.3851 \cdot 10^{-3}$	$1.5057 \cdot 10^{-4}$	$6.9620 \cdot 10^{-7}$	$9.1430 \cdot 10^{-12}$
13	$6.7002 \cdot 10^{-2}$	$3.7782 \cdot 10^{-3}$	$7.5297 \cdot 10^{-5}$	$2.0886 \cdot 10^{-7}$	$9.1430 \cdot 10^{-13}$
14	$6.0751 \cdot 10^{-2}$	$2.6490 \cdot 10^{-3}$	$3.7651 \cdot 10^{-5}$	$6.2658 \cdot 10^{-8}$	$9.1430 \cdot 10^{-14}$
15	$5.5045 \cdot 10^{-2}$	$1.8564 \cdot 10^{-3}$	$1.8826 \cdot 10^{-5}$	$1.8797 \cdot 10^{-8}$	$9.1430 \cdot 10^{-15}$
16	$4.9843 \cdot 10^{-2}$	$1.3005 \cdot 10^{-3}$	$9.4134 \cdot 10^{-6}$	$5.6392 \cdot 10^{-9}$	$9.1430 \cdot 10^{-16}$
17	$4.5108 \cdot 10^{-2}$	$9.1090 \cdot 10^{-4}$	$4.7067 \cdot 10^{-6}$	$1.6917 \cdot 10^{-9}$	$9.1430 \cdot 10^{-17}$
18	$4.0800 \cdot 10^{-2}$	$6.3787 \cdot 10^{-4}$	$2.3534 \cdot 10^{-6}$	$5.0753 \cdot 10^{-10}$	$9.1430 \cdot 10^{-18}$
19	$3.6887 \cdot 10^{-2}$	$4.4663 \cdot 10^{-4}$	$1.1767 \cdot 10^{-6}$	$1.5225 \cdot 10^{-10}$	$9.1430 \cdot 10^{-19}$
20	$3.3334 \cdot 10^{-2}$	$3.1270 \cdot 10^{-4}$	$5.8835 \cdot 10^{-7}$	$4.5677 \cdot 10^{-11}$	$9.1430 \cdot 10^{-20}$
21	$3.0112 \cdot 10^{-2}$	$2.1892 \cdot 10^{-4}$	$2.9417 \cdot 10^{-7}$	$1.3703 \cdot 10^{-11}$	$9.1430 \cdot 10^{-21}$
22	$2.7191 \cdot 10^{-2}$	$1.5326 \cdot 10^{-4}$	$1.4708 \cdot 10^{-7}$	$4.1110 \cdot 10^{-12}$	$9.1430 \cdot 10^{-22}$
23	$2.4546 \cdot 10^{-2}$	$1.0728 \cdot 10^{-4}$	$7.3544 \cdot 10^{-8}$	$1.2333 \cdot 10^{-12}$	$9.1430 \cdot 10^{-23}$
24	$2.2151 \cdot 10^{-2}$	$7.5106 \cdot 10^{-5}$	$3.6772 \cdot 10^{-8}$	$3.6999 \cdot 10^{-13}$	$9.1430 \cdot 10^{-24}$
25	$1.9985 \cdot 10^{-2}$	$5.2576 \cdot 10^{-5}$	$1.8386 \cdot 10^{-8}$	$1.1099 \cdot 10^{-13}$	$9.1430 \cdot 10^{-25}$
26	$1.8027 \cdot 10^{-2}$	$3.6804 \cdot 10^{-5}$	$9.1930 \cdot 10^{-9}$	$3.3299 \cdot 10^{-14}$	$9.1430 \cdot 10^{-26}$
27	$1.6256 \cdot 10^{-2}$	$2.5763 \cdot 10^{-5}$	$4.5965 \cdot 10^{-9}$	$9.9897 \cdot 10^{-15}$	$9.1430 \cdot 10^{-27}$
28	$1.4657 \cdot 10^{-2}$	$1.8034 \cdot 10^{-5}$	$2.2982 \cdot 10^{-9}$	$2.9969 \cdot 10^{-15}$	$9.1430 \cdot 10^{-28}$
29	$1.3213 \cdot 10^{-2}$	$1.2624 \cdot 10^{-5}$	$1.1491 \cdot 10^{-9}$	$8.9907 \cdot 10^{-16}$	$9.1430 \cdot 10^{-29}$
30	$1.1909 \cdot 10^{-2}$	$8.8370 \cdot 10^{-6}$	$5.7456 \cdot 10^{-10}$	$2.6972 \cdot 10^{-16}$	$9.1430 \cdot 10^{-30}$

Tabelle 42: Kompromittierungswahrscheinlichkeiten bei acht Signaturen.

[107] n - Nummer des Angriffs; P(w) - Wahrscheinlichkeit des Empfangs einer Warnmeldung.

11.9 Kompromittierungswahrscheinlichkeiten bei neun (9) Signaturen

n/P(w)[108]	10%	30%	50%	70%	90%
1	$9.9999 \cdot 10^{-2}$	$9.9999 \cdot 10^{-2}$	$9.9999 \cdot 10^{-2}$	$9.9999 \cdot 10^{-2}$	$9.9999 \cdot 10^{-2}$
2	$9.0999 \cdot 10^{-2}$	$7.2999 \cdot 10^{-2}$	$5.4999 \cdot 10^{-2}$	$3.7000 \cdot 10^{-2}$	$1.8999 \cdot 10^{-2}$
3	$8.2728 \cdot 10^{-2}$	$5.2698 \cdot 10^{-2}$	$2.9012 \cdot 10^{-2}$	$1.2058 \cdot 10^{-2}$	$2.2248 \cdot 10^{-3}$
4	$7.5139 \cdot 10^{-2}$	$3.7722 \cdot 10^{-2}$	$1.4927 \cdot 10^{-2}$	$3.7192 \cdot 10^{-3}$	$2.2694 \cdot 10^{-4}$
5	$6.8190 \cdot 10^{-2}$	$2.6832 \cdot 10^{-2}$	$7.5749 \cdot 10^{-3}$	$1.1254 \cdot 10^{-3}$	$2.2740 \cdot 10^{-5}$
6	$6.1836 \cdot 10^{-2}$	$1.8998 \cdot 10^{-2}$	$3.8161 \cdot 10^{-3}$	$3.3852 \cdot 10^{-4}$	$2.2745 \cdot 10^{-6}$
7	$5.6035 \cdot 10^{-2}$	$1.3407 \cdot 10^{-2}$	$1.9153 \cdot 10^{-3}$	$1.0163 \cdot 10^{-4}$	$2.2745 \cdot 10^{-7}$
8	$5.0745 \cdot 10^{-2}$	$9.4390 \cdot 10^{-3}$	$9.5951 \cdot 10^{-4}$	$3.0498 \cdot 10^{-5}$	$2.2746 \cdot 10^{-8}$
9	$4.5928 \cdot 10^{-2}$	$6.6340 \cdot 10^{-3}$	$4.8021 \cdot 10^{-4}$	$9.1502 \cdot 10^{-6}$	$2.2746 \cdot 10^{-9}$
10	$4.1546 \cdot 10^{-2}$	$4.6570 \cdot 10^{-3}$	$2.4022 \cdot 10^{-4}$	$2.7451 \cdot 10^{-6}$	$2.2746 \cdot 10^{-10}$
11	$3.7564 \cdot 10^{-2}$	$3.2664 \cdot 10^{-3}$	$1.2014 \cdot 10^{-4}$	$8.2354 \cdot 10^{-7}$	$2.2746 \cdot 10^{-11}$
12	$3.3949 \cdot 10^{-2}$	$2.2897 \cdot 10^{-3}$	$6.0077 \cdot 10^{-5}$	$2.4706 \cdot 10^{-7}$	$2.2746 \cdot 10^{-12}$
13	$3.0669 \cdot 10^{-2}$	$1.6043 \cdot 10^{-3}$	$3.0040 \cdot 10^{-5}$	$7.4119 \cdot 10^{-8}$	$2.2746 \cdot 10^{-13}$
14	$2.7696 \cdot 10^{-2}$	$1.1238 \cdot 10^{-3}$	$1.5020 \cdot 10^{-5}$	$2.2235 \cdot 10^{-8}$	$2.2746 \cdot 10^{-14}$
15	$2.5003 \cdot 10^{-2}$	$7.8706 \cdot 10^{-4}$	$7.5105 \cdot 10^{-6}$	$6.6707 \cdot 10^{-9}$	$2.2746 \cdot 10^{-15}$
16	$2.2565 \cdot 10^{-2}$	$5.5113 \cdot 10^{-4}$	$3.7552 \cdot 10^{-6}$	$2.0012 \cdot 10^{-9}$	$2.2746 \cdot 10^{-16}$
17	$2.0360 \cdot 10^{-2}$	$3.8588 \cdot 10^{-4}$	$1.8776 \cdot 10^{-6}$	$6.0036 \cdot 10^{-10}$	$2.2746 \cdot 10^{-17}$
18	$1.8365 \cdot 10^{-2}$	$2.7016 \cdot 10^{-4}$	$9.3882 \cdot 10^{-7}$	$1.8010 \cdot 10^{-10}$	$2.2746 \cdot 10^{-18}$
19	$1.6562 \cdot 10^{-2}$	$1.8913 \cdot 10^{-4}$	$4.6941 \cdot 10^{-7}$	$5.4032 \cdot 10^{-11}$	$2.2746 \cdot 10^{-19}$
20	$1.4933 \cdot 10^{-2}$	$1.3240 \cdot 10^{-4}$	$2.3470 \cdot 10^{-7}$	$1.6209 \cdot 10^{-11}$	$2.2746 \cdot 10^{-20}$
21	$1.3462 \cdot 10^{-2}$	$9.2689 \cdot 10^{-5}$	$1.1735 \cdot 10^{-7}$	$4.8629 \cdot 10^{-12}$	$2.2746 \cdot 10^{-21}$
22	$1.2134 \cdot 10^{-2}$	$6.4885 \cdot 10^{-5}$	$5.8676 \cdot 10^{-8}$	$1.4588 \cdot 10^{-12}$	$2.2746 \cdot 10^{-22}$
23	$1.0935 \cdot 10^{-2}$	$4.5420 \cdot 10^{-5}$	$2.9338 \cdot 10^{-8}$	$4.3766 \cdot 10^{-13}$	$2.2746 \cdot 10^{-23}$
24	$9.8542 \cdot 10^{-3}$	$3.1795 \cdot 10^{-5}$	$1.4669 \cdot 10^{-8}$	$1.3130 \cdot 10^{-13}$	$2.2746 \cdot 10^{-24}$
25	$8.8785 \cdot 10^{-3}$	$2.2256 \cdot 10^{-5}$	$7.3346 \cdot 10^{-9}$	$3.9390 \cdot 10^{-14}$	$2.2746 \cdot 10^{-25}$
26	$7.9986 \cdot 10^{-3}$	$1.5579 \cdot 10^{-5}$	$3.6673 \cdot 10^{-9}$	$1.1817 \cdot 10^{-14}$	$2.2746 \cdot 10^{-26}$
27	$7.2051 \cdot 10^{-3}$	$1.0906 \cdot 10^{-5}$	$1.8336 \cdot 10^{-9}$	$3.5451 \cdot 10^{-15}$	$2.2746 \cdot 10^{-27}$
28	$6.4898 \cdot 10^{-3}$	$7.6342 \cdot 10^{-6}$	$9.1682 \cdot 10^{-10}$	$1.0635 \cdot 10^{-15}$	$2.2746 \cdot 10^{-28}$
29	$5.8450 \cdot 10^{-3}$	$5.3440 \cdot 10^{-6}$	$4.5841 \cdot 10^{-10}$	$3.1905 \cdot 10^{-16}$	$2.2746 \cdot 10^{-29}$
30	$5.2639 \cdot 10^{-3}$	$3.7408 \cdot 10^{-6}$	$2.2920 \cdot 10^{-10}$	$9.5717 \cdot 10^{-17}$	$2.2746 \cdot 10^{-30}$

Tabelle 43: Kompromittierungswahrscheinlichkeiten bei neun Signaturen.

[108] n - Nummer des Angriffs; P(w) - Wahrscheinlichkeit des Empfangs einer Warnmeldung.

Glossar

Backdoor (Hintertür) Ein geheimer Zugang, den die Programmierer in ihre Software eingebaut haben, der den Zugriff auf das System ermöglicht.

Bakterien Eine Form von *Malware*, die keine Schadensroutine beinhaltet, jedoch sich wiederholt selbst reproduziert und möglicherweise über das komplette System verbreitet.

Bluetooth Englisch: "Blau Zahn" wurde von den Initiatoren Ericsson, IBM, Intel, Nokia und Toshiba zum Leben erweckt. Dabei entstand eine interessante Entwicklung im Bereich der Kurzstrecken-Kommunikation per Funk. Bis zu acht (8) Geräte können wechselseitig miteinander kommunizieren. U.a. sind folgende Funktionen möglich: Datenaustausch zwischen Notebook, PDA und Handy, Drucker, wireless LAN, Dial-Up-Adapter für drahtlose Nutzung von Netzwerk, ISDN und analogen Telefonleitungen (z.B. per Stecker für die TAE-Dose), drahtlose Headsets für moderne Handys, Fernsteuerung und vieles mehr.

Computervirus Bezeichnung für Programme, die sich, wenn sie einmal geladen sind, beliebig vervielfältigen können und den Sinn und Zweck verfolgen, den Betriebsablauf eines Computers zu stören.

Computerwurm Eine Art von *Malware*. Programm mit Schadensfunktionen, das sich nach Befall eines Rechners automatisch über das Netzwerk weiter zu verbreiten versucht.

CRC32 siehe *CRC*.

CRC Cyclic Redundancy Check Prüfsumme, wird in Übertragungsprotokollen und Packern verwendet. Üblich sind 16 oder 32 Bit lange Varianten, kurz: CRC16 und *CRC32*. Eine *CRC* stellt den Rest aus einer Polynomdivision dar. Implementationen sind allgemein als Quelltext erhältlich.

DES Digital Encryption Standard. Symmetrischer Verschlüsselungsalgorithmus mit einer effektiven 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 und Quellcode bestätigen.

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

Exploit Software, bzw. Code, das spezifische Schwächen beziehungsweise Fehlfunktionen bestimmter Softwareprodukte ausnutzt.

FIPA Foundation for Intelligent Physical Agents (*FIPA*) ist eine non-profit Vereinigung von Firmen und Forschungseinrichtungen. Ihr Ziel ist die Spezifikation einer generischen Agententechnologie, um einen hohen Grad an Interoperabilität zwischen agentenbasierten Anwendungen zu erreichen.

Firewall Technik in Form von Hard- und/oder Software, die den Datenfluß zwischen einem privaten und einem ungeschützten Netzwerk (z.B. LAN und Internet) kontrolliert bzw. ein internes Netz vor Angriffen aus dem Internet schützt.

FTP File Transfer Protocol. Internetstandard zur Übertragung von Text- und Binärdateien.

GnuPG *GNU* Privacy Guard ist eine Open-Source Verschlüsselungs-Software. *GnuPG* ist, im Gegensatz zum de-facto-Standard *PGP*, frei von Rechten Dritter (*PGP* nutzt das patentierte *IDEA*) und unterliegt nicht der amerikanischen Gesetzgebung (z.B. Exportbeschränkungen). Die Weiterentwick-

lung von *GnuPG* wird vom Bundeswirtschaftsministerium gefördert, das leistungsstarke Verschlüsselungstechnologie der Öffentlichkeit zur Verfügung stellen will.

GNUtella Dezentrale Peer-to-Peer-Verwaltung unter *GPL* zum Austausch von Dateien.

GNU *GNU*'s not UNIX. Das *GNU*-Projekt wurde 1984 von Richard Stallman in seinem Buch "Open Sources - The first Software-sharing Community" initiiert, um ein vollständiges Unix-artiges Betriebssystem zu entwickeln, das freie Software ist – das *GNU* System.

GPL *GNU* Public License.

Hashalgorithmus siehe *Hash*

Hashfunktion siehe *Hash*.

Hash Als *Hash* bezeichnet man eine eindeutige Prüfsumme einer Datei oder eines Datensatzes. Mit der *Hashfunktion* können somit Daten auf Ihre Vollständigkeit überprüft werden. Tauschbörsen und Filesharing-Programme greifen auf sogenannte *Hashs* zurück um Dateien eindeutig zu kennzeichnen. Da der *Hash* eine kurze Prüfsumme ist, müssen nur sehr wenige Daten übermittelt werden um eine Datei eindeutig zu identifizieren. *Hashs* sind in der Regel kryptographische Zeichenketten mit einer festgelegten Länge. Beispiele für *Hashalgorithmen* sind z.B. MD4, *MD5*, SHA1, HAVAL, RIPEMD160, RIPEMD128, SNEFRU, TIGER, GOST, *CRC32* und CRC32B.

ICMP Internet Control Message Protocol. Kontrollprotokoll auf ISO-/OSI-Schicht 3, z. B. basiert der PING auf *ICMP*-Echo-Request und -Response.

IDEA *IDEA* steht für International Data Encryption Algorithm und ist ein blockorientierter, konventioneller Verschlüsselungsalgorithmus, der an Eidgenössischen Technischen Hochschule der Schweiz entwickelt wurde. *IDEA* arbeitet mit 64 Bit Blöcken und mit einer Schlüssellänge von 128 Bit und ist we-

sentlich sicherer als *DES* (64 Bit bzw. 56 Bit). *IDEA* ist in den USA sowie in den meisten europäischen Ländern patentiert. Die nicht-kommerzielle Nutzung ist frei.

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.

JADE Java Agent DEvelopment Framework ist ein Open-Source Projekt, dessen Ziel darin besteht, der Entwicklergemeinde ein bequemes Werkzeug zur Erstellung von Agenten bzw. Multiagentensystemen zur Verfügung zu stellen.

Load Balancing Lastverteilung auf mehrere Rechner.

Makrovirus *Computervirus*, der die Makroprogrammiersprache eines Anwendungsprogramms benutzt, um seinen Code zu replizieren und damit Schaden anzurichten. Er kann betriebssystemübergreifend sein und befällt keine Programme, sondern einzelne Dokumente (z.B. Word oder Excel). Oftmals werden *Makroviren* über Anhänge zu eMails weiterverbreitet.

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*.

Maskeradeangriff (Masquerading) Eine Art von Angriffen, bei welcher der Angreifer eine falsche Identität annimmt und mit den Berechtigungen des gefälschten Systems oder Benutzers agiert.

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*.

PAP Abkürzung von "Point Authorizution Protocol", Authentifizierungsmethode für *PPP*, die auf User-Namen basiert und das Paßwort unverschlüsselt überträgt.

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.

PPP Abkürzung für "Point to Point Protocol", was wörtlich "*Protokoll* für die Übertragung von Punkt zu Punkt" bedeutet und 1991 von der IETF (Internet Engineering Task Force) definiert wurde.

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

RSA Ein Algorithmus zum Signieren und asymmetrischen 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.

TELNET Das Standard-*Protokoll* im Internet für remote login. Damit kann man zu einem anderen Host über das Internet eine interaktive Verbindung aufbauen, als ob man direkt an diesem via Terminal angeschlossen wäre.

Trojanisches Pferd Ein *Trojanisches Pferd* ist ein selbstständiges Programm mit einer verdeckten Schadensfunktion. Im Betriebssystem eines Computers

kann sich ein *Trojanisches Pferd* häufig unbemerkt entfalten und wichtige Daten zerstören oder Passwörter ausspionieren und an eine geheime Adresse weiterleiten. Häufig gaukeln *Trojanische Pferde* vor, nützlich oder harmlos unterhaltend zu sein. Von einem klassischen *Computervirus* unterscheiden sich die *Trojanische Pferde* vor allem dadurch, dass sie sich nicht selbstständig vermehren, sondern an ihr *Wirtsprogramm* gebunden sind. Sie nisten sich oft im Betriebssystem der Rechner ein, sind also für durchschnittliche PC-Nutzer kaum zu erkennen.

Virus siehe *Computervirus*.

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 Schlüsseleigentümern 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.

Wurm Siehe *Computerwurm*

Literatur

[ALTHOFF 1985] Althof, H. (1985) *Wahrscheinlichkeitsrechnung und Statistik.* Stuttgart, Carl Ernst Poeschel Verlag GmbH, ISBN 3-476-50212-0.

[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.

[BASLER 1989] Basler, H. (1989) *Grundbegriffe der Wahrscheinlichkeitsrechnung und Statistischen Methoden.* Heilderberg, Physica-Verlag, ISBN 3-7908-0435-5.

[BLEICH 2004] BLEICH, H. (2004) *Auf Phishzug.* Heise Verlag, c't 17/2004 S. 178.

[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 (Konzepte und Design).* München, Pearson Education Deutschland GmbH, ISBN 3-8273-7022-1.

[DOMHAN 2001] Domhan, G. (2001) *LAN-Sicherheitskonzept und Aufbau eines Firewall-Systems.* <http://www.domhan.de> (Click-Datum 29.08.2004).

[FEISTHAMMEL 2002] Feisthammel, P. (Juni 2002) *Das web of trust (Vertrauensnetz).* <http://www.rubin.ch/pgp/weboftrust.de.html> (Click-Datum 29.08.2004).

[FERRIE 2004] Ferrie, P. Szor, P., Stanev, R. Mouritzen, R. (Juni 2004) *Security Responce (epoc.cabir).* <http://securityresponse.symantec.com/avcenter/venc/data/epoc.cabir.html> (Click-Datum 29.08.2004).

[FIPA 2004] Foundation for Intelligent Physical Agents *Content Language Specifications.* <http://www.fipa.org/repository/cls.php3> (Click-Datum 29.08.2004).

[FUHS 1995] Fuhs, H. (Mai 1995) *Methoden zur Entdeckung von Computerviren.* <http://www.fuhs.de/de/fachartikel/artikel_de/methviren.shtml> (Click-Datum 29.08.2004).

[KELLER 2003] Keller, K. (Juni 2003) *Wahrscheinlichkeitsrechnung und Statistik (Bedingte Wahrscheinlichkeiten und stochastische Unabhängigkeiten).* <http://www.rz.rwth-aachen.de/mata/downloads/statistik/kap3.pdf> (Click-Datum 29.08.2004)

[KLANDER 1997] Klander, L. (1997) *Hacker Proof (The Ultimate Guide to Network Security).* Las Vegas, Jamsa Press, ISBN 1-884133-55-X.

[KNOWBUDDY 2004] Knowbuddy, (Mai 2004) *Knowbuddy's Gnutella FAQ.* <http://www.rixsoft.com/Knowbuddy/gnutellafaq.html> (Click-Datum 29.08.2004).

[LIPSCHUTZ 1989] Lipschutz, S. (1989) *Wahrscheinlichkeitsrechnung (Theorie und Anwendung).* Frankfurt am Main, HAAG + HERCHEN Verlagsbüro GmbH, ISBN 0-07-084361-9.

[METZGER 2003] Metzger, J. (2003) *Ein Multiagenten-basiertes Peer-To-Peer Netzwerk zur verteilten, effizienten Spam-Filterung.* <http://www.virtosphere.de/schillo/teaching/Diplomarbeit/JoergMetzger/-Joerg.Metzger.Diplomarbeit.pdf> (Click-Datum 29.08.2004).

[NEUMANN 1995] Neumann, P. G. (1995) *Computer Related Risks.* New York, Addison-Wesley, ISBN 0-201-55805-X.

[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.

[OA 1996] O. A., (April 1996) *Objekt: CAD-Lexikon.* <http://www.blien.de/ralf/cad/db/objekt.htm> (Click-Datum 29.08.2004).

[POSTEL 1981] Postel, J., (September 2004) *RFC 792: Internet Control Message Protocol (Darpa Interner Program Protocol Specification).* <http://www.ietf.org/rfc/rfc0792.txt> (Click-Datum 29.08.2004).

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

[SACHS 2003] Sachs, M. (2003) *Wahrscheinlichkeitsrechnung und Statistik (für Ingenieurstudenten an Fachhochschulen).* München, Carl Hanser Verlag, ISBN 3-446-22202-2.

[SCHMIDT 2004] SCHMIDT, J. (2004) *Phatbot im Spiegel von Sasser.* Heise Verlag, c't 11/2004 S. 58.

[SCHNEIER 2001] Schneier, B. (Mai 2001) *Defense Options: What Military History Can Teach Network Security (Part 2).* <http://www.schneier.com/crypto-gram-0105.html> (Click-Datum 29.08.2004).

[SINGLA 2001] Singla, A. Rohrs, C. (Dezember 2001) *Ultrapeers: Another Step Towards Gnutella Scalability.* <http://rfc-gnutella.sourceforge.net/Proposals/Ultrapeer/Ultrapeers.htm> (Click-Datum 29.08.2004).

[STALLINGS 2000] Stallings, W. (2000) *Network Security Essentials (Applications and Standards).* USA, Prentice Hall Inc., ISBN 0-13-016093-8.

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

[WIKIPEDIA 2004] WIKIPEDIA, o.V., (Juni 2004) *Verteiltes System.* ‹http://de.wikipedia.org/wiki/Verteilte_Systeme› (Click-Datum 29.08.2004).

[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 MultiAgent Systems.* Chichester, England, John Wiley & Sons, ISBN 047149691X.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Wiesbaden, 29. August 2004

_____ _____

Vorname Nachname