TU Darmstadt
FB Informatik
Fachgebiet Sicherheit in der Informationstechnik
Frau Prof. C. Eckert

Ausarbeitung im Rahmen des Seminars
"Anomalieerkennung durch künstliche Immunsysteme"

Investigating the Roles of Negative Selection and Clonal Selection in an Artificial Immune System for Network Intrusion Detection

Anne Paul

Wintersemester 2003 / 04

1 Einführung

Computernetzwerke werden immer umfangreicher, firmeninterne Netzwerke und das Internet sind nicht mehr weg zu denken. Aber es geschehen auch immer mehr Angriffe auf diese Netze. Daraus entsteht der Wunsch Computernetze abzusichern. So genannte Intrusion Detection Systeme haben zum Ziel Angriffe zu erkennen und abzuwehren. Allerdings ist ein nicht zu vernachlässigender Administrationsaufwand mit ihrem Betrieb verbunden. Die Natur hat in Millionen von Jahren ein System zum Schutz des Körpers entwickelt, das Immunsystem [11]. Ein solches System wäre ideal für Computernetzwerke. So kam die Idee auf, das menschliche Immunsystem zum Vorbild zu nehmen, und so künstliche Immunsysteme für Computernetzwerke zu entwickeln.

Im Folgenden wird vor dem Hintergrund des menschlichen Immunsystems und derzeitiger Intrusion Detetction Systeme ein Algorithmus zur klonalen Selektion unter Verwendung eines *negative selection operators* nach Kim und Bentely [11] dargestellt. Darüber hinaus wird ein kurzer Ausblick auf dynamische klonale Selektion sowie auf Ideen der Danger Theory gewährt. Die Danger Theory könnte die Entwicklung künstlicher Immunsysteme maßgeblich vorantreiben.

2 Das menschliche Immunsystem

Das menschliche Immunsystem schützt den Körper vor Angriffen von außen. Es verteidigt ihn gegen körperfremde Substanzen und Krankheitserreger. Dazu ist es notwendig, dass das Immunsystem zwischen körpereigenen (*self*) und körperfremden (*non-self*) Elementen unterscheiden kann. Seiner Aufgabe gerecht wird das leistungsstarke, effektive Immunsystem durch einen komplexen Aufbau. Die Grundlagen der Funktionsweise des menschlichen Immunsystems umfassend zu erläutern ist nicht Aufgabe dieser Ausarbeitung. Allerdings sollen im Folgenden einzelne Aspekte angesprochen werden, die für die Umsetzung in ein Intrusion Detetction System von Interesse sind.

2.1 Welche Eigenschaften hat das menschliche Immunsystem?

Das Immunsystem besteht aus drei wesentlichen Phasen: der Entwicklung der genetischen Bibliothek, der negativen Selektion und er der klonalen Selektion.

Unser körpereigenes Immunsystem erfüllt drei wesentliche Kriterien: es ist verteilt, organisiert sich selbst und ist vollständig.

Die Verteiltheit zeigt sich darin, dass das Immunsystem aus unterschiedlichen im Körper verteilten Organen besteht (z.B. dem Knochenmark und dem Thymus) und aus Zellen, die durch die Blutbahnen zirkulieren. Es gibt kein zentrales Steuerungsorgan, dass das Immunsystem von einem einzigen Punkt aus koordiniert. So können Immunantworten auf erkannte Gefahren schnell und (bedingt) lokal geschehen.

Das Immunsystem organisiert sich selbst. Hier spielt wiederum die Abwesenheit eines zentralen Steuerungsorgans eine Rolle. Antikörper werden scheinbar zufällig aus der genetischen Bibliothek gebildet. Es gibt zuerst keine Beschränkung in irgendeiner Weise. Antikörper entstehen als zufällige Kombinationen der Gensegmente der genetischen Bibliothek. Anschließend

durchlaufen die neu gebildeten Antikörper die Stufe der negativen Selektion. Hierbei werden ihnen körpereigene Zellen vorgeführt, auf die sie nicht reagieren dürfen. Falls ein Antikörper dennoch eine körpereigene Zelle als gefährlich erkennt und sich daran bindet, stirbt dieser Antikörper ab.

In der letzten Stufe, der klonalen Selektion, vermehren sich Antikörper, die sich als besonders gut heraus gestellt haben. Antikörper, die zwar funktionsfähig aber nicht effektiv sind, sterben nach einer bestimmten Lebensdauer ab. Es gibt also in der Entwicklung der Antikörper kein zentrales Organ, dass vorgibt, welche Antikörper gebildet werden sollen, sondern nur eine Vorgehensweise, mit der der Körper effektive Antikörper bildet.

Das letzte wichtige Kriterium, das das körpereigene Immunsystem erfüllt, ist die Vollständigkeit. Der Körper ist in der Lage, mit einer relativ geringen Anzahl an Antikörpern eine große Anzahl an Pathogenen zu erkennen. Darüber hinaus wird gesammeltes Wissen über effektive Beseitigung von Pathogenen wirksam wieder verwendet. Die effektiven Antikörper bleiben, wie oben beschrieben, lange am Leben. Der Körper lernt effektiv. Nicht zuletzt hat der Körper die Möglichkeit, über die genetische Bibliothek schnell eine große Anzahl von unterschiedlichen Antikörpern zu erzeugen [6].

Das menschliche Immunsystem ist zusammenfassend ein System, dass ohne äußere Steuerung effektiv den Körper vor Gefahren schützt. Dazu verfügt es über Möglichkeiten sich an veränderte Bedingungen anzupassen.

2.2 Wichtige Begriffe

Im Folgenden werden stichwortartig einige der im Weiteren mehrfach auftauchenden Begriffe kurz erläutert.

Antigen

Als Antigen bezeichnet mal alle körperfremden Stoffe. Sie werden im Allgemeinen vom Immunsystem erkannt und lösen eine Immunreaktion aus. So ist der Körper in der Lage sich vor Gefahren von außen zu schützen.

Antikörper

Antikörper werden vom Körper als Antwort auf ein erkanntes Antigen gebildet. Sie binden das Antigen und machen es damit unschädlich.

Genotyp

Der Genotyp eines Organismus ist sein vollständiges Erbgut. Das Erbgut stammt anteilig von beiden Elternteilen.

Phänotyp

Der Phänotyp ist das Erscheinungsbild eines Organismus. Er hängt maßgeblich vom Genotyp dieses Lebewesens ab, wird aber auch stark durch die Umwelt des Organismus geprägt. Der Phänotyp eines Organismus verändert sich im Laufe seines Lebens.

Negative Selektion

Antikörper durchlaufen in ihrer Entwicklung die Phase der negativen Selektion. Hier werden den Antikörpern körpereigene Zellen vorgeführt. Werden diese *self* Zellen von den Antikörpern gebunden, sind diese Antikörper offensichtlich defekt. Sie sterben ab.

Klonale Selektion oder klonale Expansion

Die klonale Selektion ist die Vermehrung und Differenzierung[1] reifer Antikörper. Die Differenzierung erfolgt mittels Mutation und Crossover.

3 Intrusion Detection Systeme

Intrusion Detection Systeme (IDS) sind automatisierte Systeme zur Erkennung von Eindringlingen in Computer Systemen. Sie sind sozusagen die Alarmanlage eines Netzwerks. Die Hauptaufgaben von IDS liegen in der Auffindung von unautorisiertem Gebrauch, von falscher Verwendung sowie von Missbrauch des Systems.

Man unterscheidet *host level* IDS und *network based* IDS. *Host level* IDS werden heute nur noch selten verwendet. Sie überwachen den Traffic eines einzelnen Hosts. *Network based* IDS sind dagegen heute weit verbreitet. Sie sind in der Lage mehrere Hosts und den Traffic eines gesamten Netzwerks zu kontrollieren.

Im Wesentlichen werden zwei Methoden angewendet: Missbrauch Erkennung (*misuse detection*) und Anomalie Erkennung (*anomaly detection*). Bei der Missbrauch Erkennung wird nach bekannten Angriffsmustern im Traffic gesucht. Zur Anomalie Erkennung werden Profile von regulärem Gebrauch des Systems erstellt. Sie umfassen zum Beispiel die Größe des normalen Netzwerk Traffics oder Informationen über normales Verhalten von Nutzern des Systems. Grobe Abweichungen von diesen Profilen werden erkannt und als Gefahr identifiziert. Die Stärke der Anomalie Erkennung ist, dass vorher unbekannte Gefährdungen erkannt werden können. Es ist kein Wissen über die Art des Angriffs notwendig, da nicht nach bekannten Angriffsmustern gesucht wird. [11].

Kim und Bentley [6] identifizieren drei Design Ziele für *network based* IDS. Da *network based* IDS verteilte Komponenten überwachen, ist es nur konsequent, sie ebenfalls verteilt zu erstellen. Des Weiteren ist es wünschenswert, dass ein *network based* IDS in der Lage ist sich selbst zu organisieren. Es soll ohne zentrales Steuerungsorgan funktionieren und selbst lernen. Damit einfällt die manuelle Steuerung von außen. Schließlich soll jedes IDS effizient arbeiten und das überwachte Netzwerk nicht unnötig belasten.

4 Künstliche Immunsysteme

Künstliche Immunsysteme (*Artificial Immune Systems* – AIS) haben zum Ziel, die Fähigkeiten des menschlichen Immunsystems in die Welt der IDS zu übertragen. Viele Fähigkeiten des Immunsystems sind wünschenswerte Eigenschaften für IDS. So drängt sich der Gedanke auf, IDS mittels Erkenntnissen über die Funktionsweise des Immunsystems zu erstellen.

Kim und Bentley stellen die Architektur eines AIS vor [8]. Das AIS besteht aus einem primären und mehreren sekundären IDS. Das primäre IDS übernimmt Aufgaben analog zur Erstellung der genetischen Bibliothek und der negativen Selektion im Körper. Die genetische Bibliothek enthält genetische Segmente effektiver Anomalie Detektoren. Sie entwickelt sich ständig. In der Phase der negativen Selektion werden die aus der genetischen Bibliothek zufällig gebildeten Detektoren getestet und ggf. aussortiert.

[1] Unter der Differenzierung von Antikörpern versteht man die gemäß einem Muster erfolgte Umwandlung.

Die sekundären IDS übernehmen die eigentliche Überwachung des Systems. Hier wird der Netzwerk Verkehr mit Hilfe der generierten Detektoren überwacht. Falls eine Anomalie erkannt wird, wird der entsprechende Detektor geklont (klonale Expansion) und an andere sekundäre IDS weitergereicht. Des Weiteren werden die Gene dieses Detektors in die genetische Bibliothek des primären IDS aufgenommen, falls sie dort noch nicht vorhanden sein sollten.

Diese Architektur eines AIS erfüllt die Design Eigenschaften, die an ein IDS gestellt werden.

5 Algorithmus zur negativen Selektion

Ein grundlegender Algorithmus zur negativen Selektion wurde von Forrest et al 1994 vorgestellt [5]. Forrest et al haben den Algorithmus zuerst zur Virenerkennung eingesetzt, stellen aber selbst fest, dass die Anwendungsbereiche weitreichender sind.

Der Algorithmus soll Veränderungen erkennen. Dazu generiert er Detektoren, die Veränderungen aufspüren. Die Generierung und Arbeitsweise der Detektoren orientiert sich stark an Erkenntnissen über die Arbeitsweise des menschlichen Immunsystems. Dabei werden die Detektoren als Analogon zu Antikörpern gesehen. Als *self*, und damit in Analogie zu den körpereigenen Zellen, werden alle unkritischen Elemente des zu betrachtenden Umfeldes eingestuft, so zum Beispiel reguläre Nutzer, fehlerfreie Dateien und ähnliches. Als *non-self*, also als Pathogene, werden alle schädlichen Elemente betrachtet, beispielsweise Angriffe, Viren oder nicht autorisierte Benutzer.

Der Algorithmus basiert auf der Annahme, dass zur Zeit der Erstellung der Detektoren das zu betrachtende Umfeld störungsfrei ist. Es wird also davon ausgegangen, dass das überwachte System in seiner Erscheinung vollständig als *self* eingestuft werden kann. Um das System handhabbar zu machen, wird es häufig auf binärem Niveau betrachtet. Jedes einzelne Element von *self* wird als binärer String identifiziert[2]. Soweit notwendig, werden diese Strings zerteilt, so dass Teilstrings identischer Länge vorliegen.

Der Algorithmus besteht aus zwei Phasen, der Phase der Generierung von Detektoren und anschließend, der Phase der Überwachung des Systems.

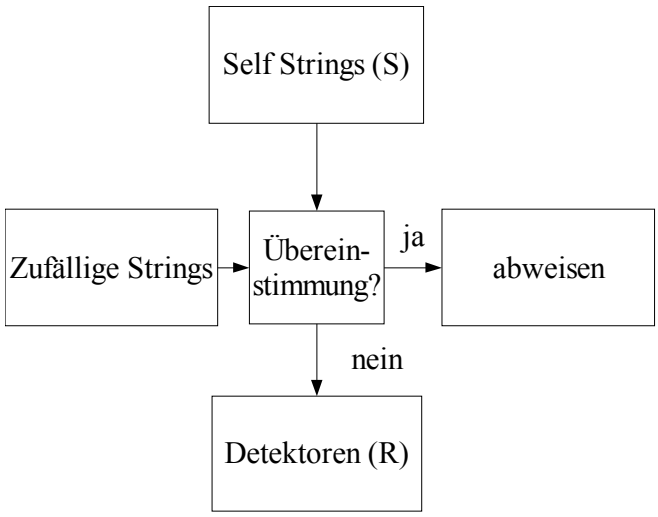

Abbildung 1 (Generierung von Detektoren)

5.1 Generierung von Detektoren

Angenommen *self* (*S* sei die Menge der Strings in *self)*, habe drei Elemente: 0011 1001 und 1101. Sollten später während der Überwachung des Systems andere Elemente auftauchen, so soll der Algorithmus dies erkennen.

[2] Es ist durchaus möglich, die Stings über einem anderen Alphabet als {0,1}* zu definieren. Allerdings verringert die binäre Darstellung den Rechenaufwand.

Zuerst werden nun wahllos binäre Strings erzeugt, mit dem Ziel einige von ihnen als Detektoren nutzen zu können. Diese Menge der Detektoren sei *R*. Beispielsweise seien die zufällig generierten Strings die Elemente 1010 1001 0000 1100 und 1101.

In einem zweiten Schritt werden die potentiellen Detektoren nun an den Elementen aus *S* getestet. Dieser Teilschritt entspricht der negativen Selektion wie sie das menschliche Immunsystem vornimmt um schlechte Antikörper auszusortieren. Alle potentiellen Detektoren, die einem Element aus *S* entsprechen, werden gelöscht. Sie kommen als Detektoren nicht in Frage. So werden nach Durchführung des Tests nur die Elemente 1010, 0000 und 1100 in *R* enthalten sein. 1001 und 1101 sind identisch zu Elementen in *S* und somit als Detektoren nicht geeignet.

In realitätsnahen Umgebungen werden *S* und *R* Strings höherer Länge enthalten. Eine vollständige Übereinstimmung wie oben beschrieben ist sehr rar. Daher relaxiert man die vollständige Übereinstimmungsfunktion häufig zu einer *r-contiguous* Übereinstimmungsfunktion. Dabei stimmen zwei Strings überein, wenn sie an wenigstens *r* benachbarten Stellen übereinstimmen. So stimmen 1100110 und 0100011 beispielsweise an 3 Stellen überein. Wird aber eine 4-contiguous Funktion angewendet, so stimmen die beiden Strings nicht überein.

5.2 Wie muss die Menge der Detektoren dimensioniert werden?

Percus et al [15] leiten eine Formel für die Wahrscheinlichkeit her, dass ein Detektor ein bestimmtes Antigen erkennt. Diese Wahrscheinlichkeit *P* hängt ab von *m*, der Anzahl der Buchstaben im Alphabet, sowie von *r*, der Anzahl der benachbarten Stellen des Strings, die übereinstimmen müssen. Darüber hinaus verändert sich P abhängig von *l*, der Länge jedes Strings.

Die Wahrscheinlichkeit *P* setzt sich aus mehreren Komponenten zusammen. Beginnt man an der linken Seite des Strings, so ist die Wahrscheinlichkeit, dass das Wort an r Stellen mit dem Detektor übereinstimmt gleich m^{-r}. Der übereinstimmende Abschnitt kann aber nicht nur ganz links im String liegen, sondern an genau *l-r* unterschiedlichen Stellen des Strings starten. Allerdings darf der übereinstimmende Abschnitt an keiner anderen Stelle des Strings beginnen; das geschieht mit einer Wahrscheinlichkeit von $m^{-r}(m-1)/m$. So setzt sich die gesamte Formel als $\mathbf{P = m^{-r} [(l-r) (m-1) / m +1]}$ zusammen.

Aus dieser Formel kann nun sehr einfach *r* errechnet werden. *m* und *l* sind bekannte Größen und *P* wird auf genau den gewünschten Wert gesetzt.

Forrest et al geben darüber hinaus Formeln an, die die Anzahl der notwendigen Detektoren berechnen [5].

5.3 Überwachung des Systems

Die Überwachung des Systems ist denkbar einfach. Die Elemente aus *S* können sich über die Zeit hinweg ändern. Kontinuierlich werden sie mit Elementen aus *R* überprüft. Sollte eine Übereinstimmung vorliegen, wird diese erkannt. Offensichtlich ist eine Veränderung in *S* aufgetreten, die nicht der als *self* bekannten Form entspricht.

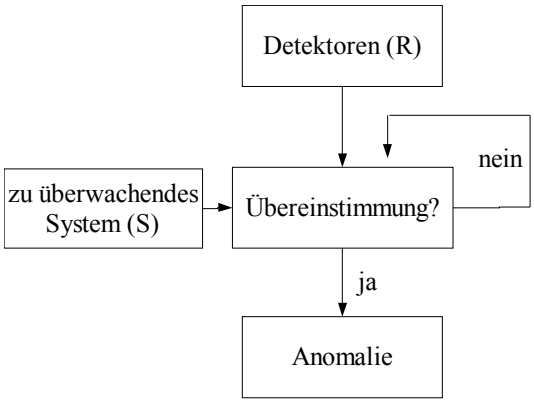

Abbildung 2 (Überwachung des Systems)

5.4 Was kann der Algorithmus leisten?

Der Algorithmus zur negativen Selektion erkennt, falls sich ein Element der zu überwachenden *self* Menge soweit verändert, als das er nicht mehr dem ursprünglichen Muster entspricht. Der Algorithmus kann aber das Entfernen einzelner Strings aus *S* nicht erkennen, er überprüft nur alle vorhandenen Elemente auf Veränderung.

6 Negative Selektion in einem IDS

Wie bereits beschrieben (siehe 4) bestehen auffällige Ähnlichkeiten zwischen den Fähigkeiten des menschlichen Immunsystems und den Anforderungen an ein IDS.

Aus diesem Blickwinkel betrachtet, stellt sich die Frage, ob der von Forrest et al vorgestellte Algorithmus [5], der sich an der negativen Selektion bei der Bildung von Antikörpern im menschlichen Immunsystem orientiert, Verwendung in Intrusion Detection Systemen finden kann. Kim und Bentley [9] haben hierzu Versuche durchgeführt. Der Algorithmus zur negativen Selektion wurde verwendet, um Netzwerk Traffic zu beobachten und Veränderungen festzustellen.

6.1 Aufbau des Experiments

Es wurden TCP Header Pakete festgehalten, die zwischen dem internen LAN und externen Servern ausgetauscht wurden, sowie von Pakete, die im Intra-LAN versendet wurden. Vier Gruppen von Feldern wurden aus den Verbindungen extrahiert: Verbindungs-Identifikatoren, auf bekannte Verwundbarkeiten hinweisende Informationen, 3-way handshake und Traffic Intensität. Die Verbindungs-Identifikatoren wurden protokolliert um einzelne Verbindungen zu identifizieren. Bestimmte Felder in TCP header weisen darauf hin, ob Sender- oder Empfänger-Port durch bekannte Angriffe verwundbar sind. Angriffe verletzen häufig die Auflagen des 3-way handshake, deshalb ist es sinnvoll zu

Externe Verbindungen	
{(2, *),(*, 80)}	5292
{(2, *),(*, 53)}	919
{(2, *),(*, 113)}	255
{(2, *),(*, 25)}	192
{(2, *),(*, trusted)}	187
{(2, *),(*, not trusted)}	756
{(2, 53),(*, *)}	940
{(2, 25),(*, *)}	352
{(2, 113),(*, *)}	145
{(2, trusted),(*, *)}	114
{(2, not trusted),(*, *)}	6050
internes LAN	
{(2, *),(2, trusted)}	190
{(2, *),(2, not trusted)}	189

Tabelle 1 (Übersicht der Verbindungen)

überprüfen ob sie eingehalten wurden. Eine auffällig hohe Intensität des Netzwerk Traffics kann auf Angriffe hinweisen.

Diese Informationen wurden kumuliert und daraus 13 unterschiedliche Klassen erstellt, die *self* widerspiegeln (siehe Tabelle 1). Ein Tupel {(a, b), (c, d)} bedeutet dabei, eine Verbindung von Host a, Port b nach Host c, Port d.[3] Alle internen Hosts werden durch die Zahl 2 symbolisiert. Externe Hosts wurden nicht identifiziert.

Für die Tests wurde nur die Klasse der Verbindungen {(2,*),(*,25)} mit 192 Verbindungen betrachtet. Es handelt sich hierbei um Verbindungen von einem internen Host zum smtp Port eines externen Hosts. Dabei wurden mit Hilfe von 154 Verbindungen Detektoren generiert, die verbleibenden 38 Verbindungen wurden zum Testen der Detektoren verwendet.

6.2 Erkenntnisse

Mit Hilfe der Formel von Forrest et al [6] (siehe 5.2) konnten Abschätzungen über die Größenordnung der Anzahl der benötigten Detektoren getroffen werden. Soll eine 4-contiguous Übereinstimmungsfunktion angewendet werden, so benötigt das System rechnerisch 955 Detektoren maximal 20% fälschliche Fehlermeldungen zu erzeugen. Pro generierten Detektor braucht das System 535 Versuche. Diese Zahlen sind offensichtlich hoch. Noch hinzuzufügen ist, dass mit dieser Konstellation innerhalb eines Tages kein einziger Detektor generiert werden konnte.

Für weitere Tests wurde daher eine 9-contiguous Funktion verwendet. So war es möglich, in akzeptabeler Zeit Detektoren zu generieren. Es wurden mit Rücksicht auf die Durchführbarkeit des Tests in 5 Durchläufen jeweils 1000 Detektoren generiert. Die erzeugten Detektoren wurden anschließend genutzt, um vier Angriffe[4] sowie das zufällige erzeugen von Strings und regulären Netzwerk Traffic zu beobachten. Der Anteil der erkannten Angriffe wurde festgehalten. Allerdings konnten nur sehr geringe Anteile der Angriffe erkannt werden, die höchste Rate liegt bei 15,9%.

Dieser geringe Erfolg war absehbar, da nur 1000 Detektoren generiert wurden. Für eine 80%ig Erkennung müssten $643 * 10^6$ Detektoren erzeugt werden. Der hierzu notwendige Rechenaufwand ist allerdings zu hoch.

6.3 Skalierung

Damit steht der praktischen Anwendung des Algorithmus zur negativen Selektion ein großes Skalierungsproblem entgegen. Die Anwendung des Algorithmus stößt bereits bei kleinen Testfällen an nicht überwindbare Grenzen der Rechenleistung. Damit ist der Algorithmus in der vorliegenden Form als IDS nicht einsetzbar.

Der Ansatz, negative Selektion zu verwenden, scheint aber durchaus anwendbar zu sein. Kim und Bentley [9] schlagen vor, die negative Selektion einzusetzen, um schlechte Detektoren zu erkennen und nicht um fähige zu erzeugen, wie es der Algorithmus zur negativen Selektion vornimmt.

[3] port Zuweisungen: port 25: smtp, port 53: DNS, port 80: http, port 113: authentication service
[4] IP spoofing, rlogin oder ftp Passworte erraten, scanning attack und network hopping attack

7 NICHING

Forrest et al haben untersucht [4], welche Art von Detektoren sich durch genetische Algorithmen entwickeln.

7.1 Welche Detektoren entwickeln sich?

Bei Detektoren kann man, analog zu den Antikörpern im menschlichen Immunsystem, von Generalisten und Spezialisten sprechen. Generalistische Detektoren erkennen eine breite Masse von Anomalien, allerdings können sie dadurch nicht speziell auf einzelne Angriffe reagieren. Dies ist die Aufgabe der Spezialisten. Die Untersuchung zeigt, dass genetische Algorithmen dazu neigen Spezialisten auszubilden, wenn die Anzahl der Detektoren steigt.

7.2 Modifizierter Algorithmus zur negativen Selektion

Aufgrund dieser Überlegungen schlagen Kim und Bentley [7] eine modifizierte Form des Algorithmus von Forrest et al [4] (siehe *5*) vor.

Zuerst muss, genau wie im ursprünglichen Algorithmus, ein *self* Profil erstellt werden. Aus diesem Profil sollen mit Hilfe des modifizierten Algorithmus effektive Detektoren generiert werden. Dazu hat jeder potentielle Detektor (ein so genannte pre-Detektor) einen Fitness Wert. Der eigentliche Algorithmus besteht aus 10 Schritten:

1. *D* Detektoren werden zufällig erzeugt und der Fitness Wert jedes Detektors wird mit null initialisiert.
2. Aus den generierten Detektoren werden *N* Detektoren zufällig ausgewählt.
3. Ein Element des *self* Profils wird zufällig ausgewählt.
4. Jeder der *N* Detektoren wird mit dem ausgewählten *self* Profil verglichen und der Wert der Ähnlichkeit festgehalten.
5. Der Fitness Wert des Detektors mit der geringsten Ähnlichkeit wird erhöht. Die Fitness Werte der übrigen Detektoren werden nicht verändert.
6. Schritte 2-5 werden wiederholt. Forrest et al [4] schlagen als Anzahl der Wiederholungen etwa drei mal die Anzahl der zu bildenden Detektoren vor.
7. Die *p%* der Detektoren mit den höchsten Fitness Werten werden als Eltern für neue Detektoren vorgesehen. Mit ihrer Hilfe werden neue Detektoren gebildet. Hierbei kommen aus der Genetik bekannt Mechanismen (z.B. Crossover und Mutation) zum Einsatz.
8. Die *q%* der Detektoren mit den niedrigsten Fitness Werten werden gelöscht, somit entsteht Platz für neue Detektoren.
9. Es werden neue Detektoren generiert. In der Menge der neuen Detektoren sind unter anderem die neu generierten Detektoren aus Schritt 7 enthalten.
10. Die Schritte 2-8 werden so lange wiederholt, bis die Fitness Werte der verbleibenden Detektoren keine signifikanten Änderungen mehr vorweisen.

8 Algorithmus zur klonalen Selektion unter Verwendung eines Negative Selection Operators

Der oben beschriebene Algorithmus (siehe 7.2) ist zuerst sehr theoretisch. Kim und Bentley haben weitergehende Überlegungen zu seiner Umsetzbarkeit angestellt [10].

Besonders hervorzuheben sind die Unterscheidung zwischen Genotyp und Phänotyp der Detektoren und die hieraus resultierende Matching Funktion.

8.1 Genotyp und Phänotyp

Alle Detektoren bestehen aus unterschiedlichen Genen. Jedes dieser Gene kann *n* Werte annehmen. So kann das Gen 1 in Abbildung 3 (Genotyp und Phänotyp nach Kim und Bentley [10]) zum Beispiel die Werte "TCP", "UDP" und "TMCP" annehmen. Ein Detektor ist binär dargestellt. Ein einzelnes Gen ist immer n+1 Stellen lang.

Die Darstellung eines Gens ist relativ einfach. Die erste Stelle des Gens ist meistens auf 0 gesetzt, dies bedeutet, dass die Ausprägung dieses Gens zu betrachten sind. Sollte die erste Stelle eine 1 beinhalten, werden die folgenden Stellen nicht mehr betrachtet. Die 1 symbolisiert, dass für den Phänotyp dieses Detektors dieses spezielle Gen alle Werte annehmen darf.

Die folgenden Stellen beziehen sich auf die einzelnen Werte des Gens. An der n+1-ten Stelle des Gens ist die Information über den n-ten Wert des Gens zu finden. Eine 1 bedeutet, dass dieser Wert angenommen wird, das Gegenteil wird von der 0 symbolisiert.

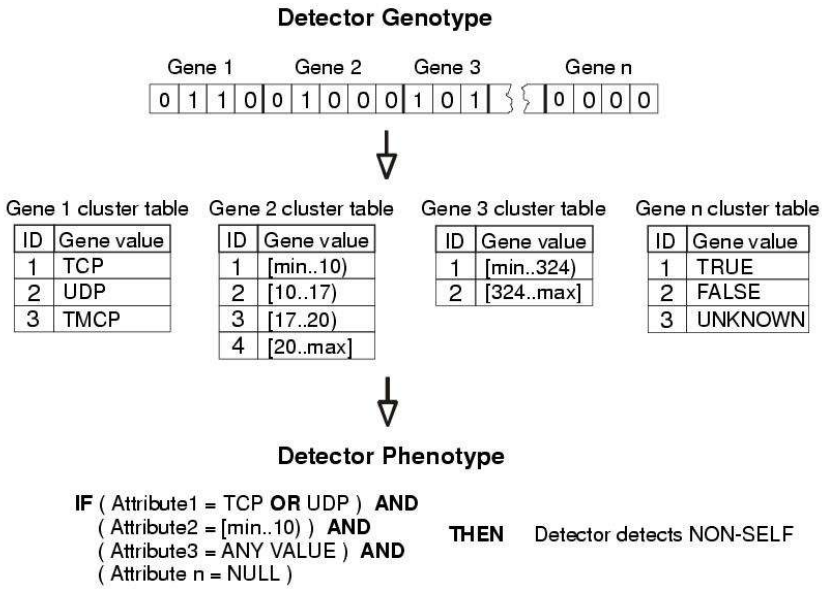

Abbildung 3 (Genotyp und Phänotyp nach Kim und Bentley [10])

Die Informationen über die einzelnen Werte in einem Gen werden als OR interpretiert. Im Beispiel bedeutet dies, dass das Gen 1 "TCP" oder "UDP" annehmen kann. Die Verbindung der Gene untereinander wird über AND realisiert. Damit ergibt sich, wie in Abbildung 3 (Genotyp und Phänotyp nach Kim und Bentley [10]) anschaulich dargestellt aus jedem Detektor Genotyp ein zugehöriger Phänotyp.

Des Weiteren kann man in Abbildung 3 (Genotyp und Phänotyp nach Kim und Bentley [10]) sehen, dass Gene nicht nur diskrete Werte annehmen können. Um dennoch mit diesen Ausprägungen arbeiten zu können, wird eine

Klasseneinteilung vorgenommen. Hierzu bedienen sich Kim und Bentley eines Algorithmus zur Diskretisierung.

8.2 Matching Funktion

Forrest et al schlagen als *Matching* Funktion eine *r-contiguous* Funktion vor [5], die auf der Ebene des Genotyps des Detektors prüft. Dagegen benutzen Kim und Bentley eine *Matching* Funktion, die auf dem Phänotyp Level des Detektors operiert. Diese *Matching* Funktion wird verwendet, um zu prüfen, ob ein Antigen von einem Detektor als Pathogen erkannt wird. Hierzu werden zuerst die Ausprägungen der Gene des Antigens mit denen des Phänotyps des Detektors verglichen. Stimmen Antigen und Detektor in allen Genen überein, wird das Antigen als *non-self* identifiziert.

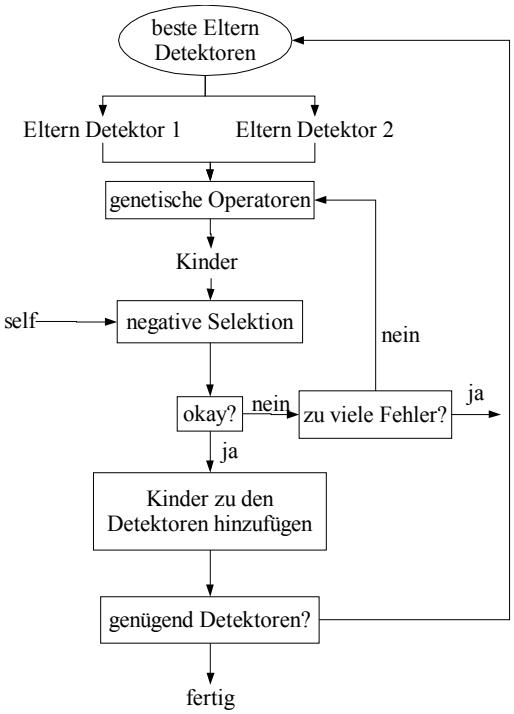

Abbildung 4 (Reproduktion und negative Selektion)

Diese Bedingung kann wiederum dadurch gelockert werden, dass keine vollständige Übereinstimmung gefordert wird, sondern eine *r-contiguous* Funktion, die in diesem Fall auf der Ebene des Phänotyps angewendet wird. Damit wird ein Antigen als Pathogen erkannt, wenn es mit dem Detektor an *r* aufeinander folgenden Stellen übereinstimmt [11].

8.3 Reproduktion und negative Selektion

Die Phase der Reproduktion läuft im Algorithmus von Kim und Bentley in mehreren Schritten ab. Der Algorithmus ist in Abbildung 4 (Reproduktion und negative Selektion) dargestellt.

Aus zwei Elternteilen werden mittels genetischer Operationen (Crossover und Mutation) Kinder gebildet. Diese Kinder unterziehen sich anschließend der negativen Selektion, bei der sie an *self* Elementen getestet werden. Haben die Kinder die negative Selektion zufrieden stellend durchlaufen, werden sie als Detektoren beibehalten.

Wenn die Kinder allerdings in der negativen Selektion aussortiert werden, greift ein weiterer Mechanismus. Sollten aus zwei bestimmten Eltern mehr als *M* mal fehlerhafte Kinder erzeugt worden sein, scheinen diese Elternteile fehlerhafte Informationen zu enthalten. Daher wählt der Algorithmus zwei neue Elternteile aus. Diese neuen Elternteile werden dann zur weiteren Generierung von Kindern genutzt.

8.4 Vollständiger Algorithmus

Kim und Bentley [10] testen den vollständigen Algorithmus. Der Algorithmus setzt sich aus den oben beschriebenen Phasen zusammen, er ist in Abbildung 5 (vollständiger Algorithmus) dargestellt.

Wie man leicht sieht, wurde an der Struktur des Algorithmus nicht viel verändert. Dagegen gibt es Veränderungen im Detail wie oben beschrieben. Dank dieser Veränderungen konnten Kim und Bentley den Algorithmus im Einsatz als IDS testen.

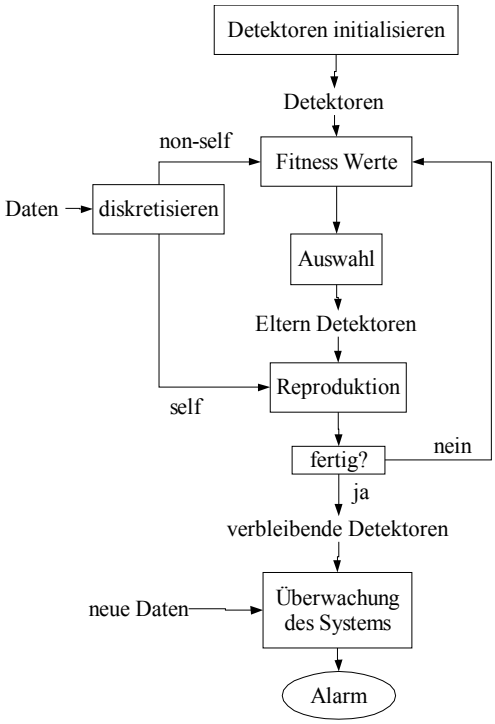

Abbildung 5 (vollständiger Algorithmus)

8.5 Tests

Kim und Bentley haben in ihren Tests [10] sehr positive Ergebnisse erzielt. Die Testfälle entsprechen den Fällen, mit deren Hilfe der ursprüngliche Algorithmus von Forrest et al getestet wurde [9]. Der modifizierte Algorithmus zeigte sich als performant. So ist die Erkennung von 95% der Anomalien in den Testdaten keine Seltenheit. Die Rate der als Anomalie erkannten *self* Antigene liegt teilweise weit unter 10%.

Unterschiedliche Parameter Konfigurationen wurden verglichen, so dass daraus eine Empfehlung entwickelt wurde. Offensichtlich führen Tests mit einer hohen Anzahl zufällig generierter Detektoren (*D*) zu guten Ergebnissen. Daher sollte *D* stets so groß gewählt werden, wie das vorhandene System es zulässt. Ein großes *D* erhöht den Rechenaufwand. Dagegen ist die im Anzahl der *non-self* Elemente (vergleiche Abbildung 5 (vollständiger Algorithmus)), die verwendet werden, um die Fitness Werte zu ermitteln, offenbar unerheblich, falls *D* richtig gewählt wurde. So sollte die Anzahl der *non-self* Elemente auf 1 gesetzt werden, in diesem Fall ist der Rechenaufwand minimal.

Auffällig ist ebenfalls, dass die negative Selektion, die im modifizierten Algorithmus als Operator angewendet wird, einen großen positiven Einfluss hat.

9 Dynamische Klonale Selektion

Die Umgebung jedes Netzwerks und das Verhalten in jedem realen Netzwerk verändern sich täglich. Damit steht ein IDS vor dem Problem, laufend neue Schemen analysieren zu müssen. Es ist gut möglich, dass zum Beispiel ein für heutige Verhältnisse auffällig hoher Traffic in einem halben Jahr völlig normal ist. Somit muss sich jedes IDS der Herausforderung stellen, dynamisch auf diese Veränderungen zu reagieren. Das menschliche Immunsystem ist dazu in der Lage.

Kim und Bentley stellen einen Algorithmus vor, *DynamiCS*, der einem AIS erlaubt sich dynamisch an sich verändernde Umgebungen anzupassen [12]. Im Laufe der Zeit werden jugendliche Detektoren immer wieder gegen *self* und *non-self* getestet und beweisen so ihre Fähigkeit als effektiver Detektor dem AIS zur Verfügung zu stehen. Nach einer festgelegten Zeit werden sie zu erwachsenen Detektoren und überwachen das System. Zeigen sie hierbei besonders große Erfolge, können sie sogar zu *memory* Detektoren werden. Nach einer festgelegten Lebensdauer sterben erwachsene Detektoren ab. Wichtig bei diesem Algorithmus

ist die Einbindung des Faktors Zeit. Nicht nur spielen unterschiedliche Zeitfenster eine Rolle in der Entwicklung der Detektoren. Darüber hinaus werden die Detektoren immer wieder mit *self* und *non-self* Profilen des Systems zu unterschiedlichen Zeitpunkten konfrontiert.

Erste Tests zur Anwendbarkeit von *DynamiCS* lassen auf Erfolge hoffen. Die Ansätze werden daher weiter verfolgt.

10 Danger Theory

Die so genannte Danger Theory ist eine neue Theorie in der Immunologie. Diese Theorie ist noch sehr umstritten. Dennoch hofft man, dass die daraus gewonnen Erkenntnisse im Bereich der AIS Einsatz finden werden. Bereits jetzt ist absehbar, dass diese neue Theorie wichtige Ansätze für AIS liefern wird, unabhängig davon, ob die Danger Theory wirklich das Verhalten des menschlichen Immunsystems weiter erklären wird [2].

10.1 Ideen der Danger Theory

Die Danger Theory basiert auf der Überlegung, dass eine einfache *self non-self* Unterscheidung Zellen im Körper zu grob ist. So gibt es zum Beispiel keine Immunreaktion auf Essen, dass sehr wohl Fremdkörper im menschlichen Körper darstellt. Des Weiteren verändert sich das *self non-self* Profil jedes Systems mit seiner Lebensdauer.

Aickelin und Cayzer [1] streichen das zentrale Element der Danger Theory heraus: das menschliche Immunsystem reagiert nicht auf *non-self* Elemente, sondern auf Gefahr (danger).

Die Danger Theory geht davon aus, dass eine Zelle, die sich in Gefahr befindet, ein so genanntes *danger signal* aussendet. Um welche Art von Signal es sich dabei handelt ist noch nicht genau bekannt. Auf dieses *danger signal* reagieren dann alle Antikörper, die sich in der *danger zone* um die entsprechende Zelle herum befinden. Sie durchlaufen den Prozess der klonalen Selektion und beheben den Schaden an der betroffenen Zelle. Antikörper, die zu weit von der betroffenen Zelle entfernt sind, also außerhalb der *danger zone*, werden nicht aktiviert.

Eine Einführung in die Danger Theory von Polly Matzinger findet sich in [13].

10.2 Umsetzung in AIS

Für die Umsetzung der Erkenntnisse der Danger Theory in die Entwicklung von AIS sind derzeit drei Punkte von besonderem Interesse: *key types* von Angriffen zu erkennen, den Grad der Stärke des Angriffs zu regeln sowie die Umsetzung der *Danger Zones*.

Es besteht die Hoffnung, über die Erkennung von *key types* von Angriffen die Erfolgsrate von AIS zu erhöhen. Aikelin und Cayzer [2] nennen als Beispiel einen Distributed Denial of Service (DDoS) Angriff, bei dem nicht wie "klassischerweise" ein *ping* ausgeführt wird, sondern ein *traceroute* Befehl. Ist das AIS in der Lage, nach dem *key type* dieses Angriffs zu suchen, also das stupide wiederholte Ausführen eines Befehls von extern, so kann er erkannt werden. Sucht das AIS nur nach *ping* Befehlen, wird dieser abgewandelte DDoS nicht erkannt.

Das menschliche Immunsystem kann darüber hinaus Immunantworten unterschiedlicher Stärke erzeugen. Auch in einem AIS ist dies wünschenswert. Angriffe werden mittels bekannter Angriffssignaturen und einem Anomalie *threshold* erkannt. Dabei führt eine zu strenge Limitierung zu vielen *false postive alerts*, durch eine zu offene Beschränkung werden wirkliche Angriffe eventuell nicht identifiziert. Ein gutes Mittelmaß ist wichtig um eine optimale Angriffsidentifizierung zu gewährleisten. Allerdings wechselt dieses Mittelmaß abhängig vom Zustand des zu überwachenden Systems. Daher ist es ein Ziel, ein AIS so zu gestalten, dass es sich dynamisch an den Zustand des Systems anpasst. Hier liegt ein direktes Analogon zum menschlichen Immunsystem vor, dass in der Lage ist, die Stärke der Immunantwort auf ein Pathogen in unterschiedlicher Stärke hervorzurufen.

Die Danger Theory geht darüber hinaus davon aus, dass Signale zu nahe liegenden Sensoren übermittelt werden, um die Immunantwort auf ein Pathogen auszulösen. Dabei ist die Definition des Begriffs "nahe liegend" noch zu klären. Im Zusammenhang mit AIS könnte das bedeuten, dass, falls zum Beispiel ein Angriff auf einem Web Server des Netzes vermutet wird, gezielt alle anderen Web Server darauf überprüft werden. So sollte es möglich sein, eine schnelle und gezielte Erkennung des Angriffs sicher zu stellen.

11 Zusammenfassung

Das menschliche Immunsystem ist das Vorbild beim Design von künstlichen Immunsystemen. Ein erster Algorithmus zur negativen Selektion hat bei näherer Betrachtung große Skalierungsprobleme. Eine mögliche Lösung dieser Probleme liegt in dem von Kim und Bentley vorgeschlagenen erweiterten Algorithmus [11]. Dieser Algorithmus nutzt die klonale Selektion und verwendet dabei einen *negative selection operator*. Dieser Ansatz zeigt in Tests gute Resultate. Neue, viel versprechende Ideen zur Konstruktion von künstlichen Immunsystemen kommen aus der Danger Theory. Vielleicht wird die Forschung in Zukunft in diese Richtung tendieren.

12 Literaturangaben

[1] – Aickelin, U., Cayzer, S., **The Danger Theory and Its Applications to Artificial Immune Systems**, Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS 2002), University of Kent at Canterbury, UK, Sptember 9th-11th, 2002

[2] - Aickelin, U., Bentley, P., Cayzer, S., Kim, J., and McLeod, J., **Danger Theory: The Link between AIS and IDS?**, Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS) Edinburgh, pp.147-155, September 1-3, 2003

[3] - Bentley, P., **Natural Design by Computer**. Article accompanying keynote speech in Proc of the AAAI Symposium on Computational Synthesis, Stanford University, Palo Alto, California. March 2003.

[4] – Forrest, S., Javornik, B., Smith, R. E., and Perelson, A. S. **Using Genetic Algorithms to Explore Pattern Recognition in the Immune System.** Evolutionary Computation, 1(3) (1993) 191-211

[5] - Forrest, S., Perelson, A.S., Allen, L., and Cherukuri, R. **Self-nonself discrimination in a computer**., IEEE Symposium on Research in Security and Privacy, pages 202-212, Oakland, CA, May 16-18 1994. Springer-Verlag.

[6] - Kim, J. and Bentley, P., **The Human Immune System and Network Intrusion Detection**, Proceedings of the 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99).

[7] - Kim, J., Bentley, P., **Negative Selection and Niching by an Artificial Immune System for Network Intrusion Detection**, In Proceedings of GECCO'99, pp. 149-158.

[8] - Kim, J, Bentley, P., **An Artificial Immune Model for Network Intrusion Detection,** 7th European Conference on Intelligent Techniques and Soft Computing (EUFIT'99), Aachen, Germany.

[9] – Kim, J., Bentley, P., **An Evaluation of Negative Selection in an Artificial Immune System for Network Intrusion Detection**, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)

[10] – Kim, J., Bentely, P., **The Artificial Immune System for Network Intrusion Detection: An Investigation of Clonal Selection with a Negative Selection Operator,** Graduate Student Workshop (2001)

[11] - Kim, J. and Bentley, P. J., **Investigating the Roles of Negative Selection and Clonal Selection in an Artificial Immune System for Network Intrusion Detection** Submitted to the *Special Issue on Artificial Immune Systems in IEEE Transactions of Evolutionary Computation* , 2001

[12] - Kim, J. and Bentley, P. J., **Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Dynamic Clonal Selection** , the Congress on Evolutionary Computation (CEC-2002), Honolulu, pp.1015 - 1020, May 12-17, 2002

[14] – Matzinger, P., **The Real Function of the Immune System or Tolerance and the Four D's (Danger; Death; Destruction and Distress),** http://cmmg.biosci.wayne.edu/asg/polly.htm (Stand 11.01.2004)

[15] - Percus, J., Percus, O., Perelson, A., **Predicting the size of the T-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination,** Proc Natl Acad Sci U S A. 1993 Mar 1;90(5):1691-1695