Autonomous Security – Eine neuartige Architektur für netzwerkbasierte Intrusion Detection und Response

Sahin Albayrak,
Katja Luther, Rainer Bye, Stephan Schmidt,
Aubrey-Derrick Schmidt und Karsten Bsufka*

Zusammenfassung

Autonomous Security stellt eine Rahmenarchitektur für die Realisierung von Sicherheitsmechanismen dar, die in der Lage sind, ihre Umgebung wahrzunehmen und mit den gesammelten Informationen, autonom auf Bedrohungen zu reagieren. Diese Architektur unterstützt somit Endanwender und Administratoren bei der Abwehr von Bedrohungen und ermöglicht es diesen, die Komplexität heutiger IT-Systeme zu beherrschen und die schnelle Ausbreitung neuartiger Malware einzuschränken. Das vorliegende Papier präsentiert unsere bisherigen Forschungsergebnisse in der Entwicklung der *Autonomous Security*-Architektur (ASA). ASA kombiniert biologisch inspirierte Anomalieerkennungsverfahren mit Peer-to-Peer-Kollaborationsverfahren, um sowohl bekannte Angriffe als auch neuartige Angriffsmuster zuverlässig und effizient zu erkennen. Diese Kombination bildet in Verbindung mit einem spiel- und optimierungstheoretisch motivierten Verteilungskonzept ein neuartiges netzwerkbasiertes Intrusion Detection- und Response-System. Die Auswertung dieser Verfahren und eine umfassende Evaluierung von ASA erfolgt mittels des agentenbasierten Netzwerksicherheitssimulators *NeSSi*.

1 Einleitung

Angriffe über Computernetzwerke (z.B. durch Viren, Würmer oder Trojaner, im Folgenden auch Malware genannt) werden in Zukunft schneller, intelligenter und besser koordiniert handeln. Hierbei nimmt die Anzahl und Vielfalt der Angriffe deutlich zu. Im Symantec Internet Security Threat Report[1] wird von der höchsten Anzahl neu aufgetretener Software-Schwächen

*DAI-Labor, Technische Universität Berlin, Ernst-Reuter Platz 7, 10587 Berlin. Email: {sahin.albayrak, katja.luther, rainer.bye, stephan.schmidt, aubrey-derrick schmidt, karsten.bsufka}@dai-labor.de

[1] http://www.symantec.com/threatreport

(2526 Fälle) im zweiten Halbjahr 2007 gesprochen, die jemals in den letzten 10 Jahren in einem vergleichbaren Zeitraum registriert wurden; darüber hinaus wird angemerkt, dass 79% dieser Schwachstellen einfach auszunutzen seien. Dies spricht für die weiterhin wachsende Bedrohung, denen Computersysteme durch Malware ausgesetzt sind.

Weit verbreitete Sicherheitslösungen (Firewalls, Anti-Virus Software, Anti-Spyware Software) eignen sich gut, um bekannte Malware und Angriffe zu erkennen. Ebenfalls geeignet für die Erkennung bekannter Angriffe sind signaturbasierte Intrusion Detection Systeme (IDS); unter Voraussetzung von flexiblen Signaturen bzw. Regeln können diese Systeme auch ähnliche neuartige Angriffe erkennen. Anomaliebasierte Verfahren der Angriffserkennung bewerten den Zustand bzw. das Verhalten eines Systems und können somit auch unbekannte Angriffe erkennen, leiden aber oftmals unter einer hohen falsch positiv Rate. Dies kann dazu führen, dass ein IDS unbenutzbar wird, weil Benutzer/Administratoren durch die Anzahl der Alarmmeldungen überfordert werden.

Die Zielsetzung des vorgestellten Systems ist es, durch intelligente Kombination von Erkennungsverfahren und Verteilung von Sensoren in Netzwerken und auf Endgeräten, die Fehlerraten zu minimieren. Die Kooperation in einem Netzwerk kann hierbei insbesondere über die Nutzung von P2P-Technologien, die sich durch Fehlertoleranz und Selbstorganisation auszeichnen, realisiert werden.

Weiterhin sollen automatische Reaktionen auf Angriffe vorgeschlagen und vorbereitet bzw. eingeleitet werden.Um diese Zielsetzung zu erreichen, müssen Detektionseinheiten unter Berücksichtigung eines beschränkten Ressourcenbudgets im Netzwerk[2] verteilt werden. Dieses *Deployment* auf die Netzknoten soll eine möglichst effiziente Detektion ermöglichen und muss vordefinierten Optimalitätskriterien genügen. Auf diese Weise entfallen beispielsweise redundante Operationen wie das mehrfache Untersuchen des gleichen Netzwerkverkehrs, und die zu verwendenden Erkennungsalgorithmen können speziell auf die Art des beobachteten Verkehrs zugeschnitten werden.

Neben der technisch korrekten Umsetzung, d.h. insbesondere einer robusten und zuverlässigen Erkennung von Angriffen aller Art, sind für den Erfolg einer solchen skizzierten Sicherheitslösung hierbei drei Faktoren verantwortlich. Erstens, es muss möglich sein die Gesamtarchitektur auf einfache Weise als Sicherheitslösung in existierenden Netzwerkinfrastrukturen zu integrieren. Zweitens, durch ihren Einsatz muss sie in ihrer Eigenschaft als *autonomes* System den Automatisierungsanteil deutlich erhöhen, um Sicherheitsexperten dabei zu unterstützen, den steigenden Administrationsaufwand in komplexen Netzwerken zu beherrschen. Zu guter Letzt soll sie ökonomisch arbeiten; dies bedeutet insbesondere, dass ein unmittelbarer Nutzen für Netzwerkbetreiber, -administratoren und -benutzer nachweisbar ist.

Bevor wir in Abschnitt 3 unsere Lösung vorstellen, gehen wir in Abschnitt 2 auf verwandte Ansätze und Vorarbeiten ein. Im Anschluss präsentieren wir in Abschnitt 4 mögliche Einsatzszenarien für unsere Lösung und in Abschnitt 5 eine Simulationsumgebung für diese Szenarien, die zur Evaluation unserer Lösung eingesetzt wird. Zum Abschluss geben wir in Abschnitt 6 noch einen Ausblick auf unsere künftigen Arbeiten.

[2] Vor allem innerhalb eines großen und komplexen Netzwerkes.

2 Stand der Forschung

ASA umfasst die Erkennung von Angriffen und die autonome Adaptation an die Gegebenheiten des überwachten Netzwerkes. Diese Teilbereiche der Architektur basieren auf unterschiedlichen Forschungsgebieten, die im folgenden Abschnitt vorgestellt werden. Hierzu gehören Maschinelles Lernen und Anomalieerkennung, die Peer-to-Peer-Technologie für ein ausfallsichere Kommunikation und das intelligente Deployment der Detektionskomponenten.

2.1 Anomalieerkennung

Innerhalb von *ASA* werden unterschiedliche Anomalieerkennungsmechanismen eingesetzt, neu entwickelt und miteinander kombiniert. Diese basieren auf Algorithmen des Maschinellen Lernens (ML) zur Klassifizierung und Gruppierung unterschiedlicher Audit-Daten sowie auf kooperativen Ansätzen in netzwerkbasierten Intrusion Detection-Systemen (IDS).

Maschinelle Lernalgorithmen (in der Anomalieerkennung) Im Bereich des Maschinellen Lernens findet man vor allem statistisch motivierte Algorithmen wie Support Vector Machines (SVM) oder Hidden Markov Models (HMM) und einige biologisch motivierte Algorithmen wie neuronale Netze, genetische Algorithmen oder *Artificial Immune Systems* (AIS). Die Grundlagen des AIS wurden vor allem von Forrest *et al.* [16] in den frühen 90er Jahren entwickelt. Diese Ansätze sind aber sehr ressourcenintensiv und auch anfällig für hohe False Positive Raten.

Aickelin *et al.* [1] nimmt durch die Gefahrentheorie und das Konzept der „Künstlichen Gewebe“ Metainformationen aus der Umgebung in die Entscheidung des künstlichen Immunsystems mit auf. Die Gefahrentheorie verknüpft die Erkennung von Anomalien mit dem Auftreten von Signalen, die meist gleichzeitig mit Angriffen auftreten [32] Ein bestimmtes Verhalten wird in diesem Kontext nur dann als Anomalie gewertet, wenn simultan eine Kostimulation durch ein Gefahrensignal erfolgt. Dieser Ansatz bietet die Möglichkeit, Probleme beim Einordnen von neuem, aber normalem Verhalten zu vermeiden, indem Umgebungsinformationen im Klassifizierungsprozess berücksichtigt werden. Dies kann zu einer Reduzierung der Fehlalarmrate von AIS Systemen führen.

In Intrusion Detection Systemen werden überwachte und unüberwachte Lernverfahren des Maschinellen Lernens benutzt, um anormales Verhalten bzw. Angriffe zu erkennen [37, 4]. Diese Verfahren basieren auf der Überprüfung der Zugehörigkeit von sogenannten Feature Vektoren zu bestimmten Angriffsklassen oder zu bekannten Clustern. Diese Feature Vektoren beschreiben das Verhalten des überwachten Systems. Kim *et al.* [23] entwickelten ein IDS auf Basis von SVMs, sowie einer Kombination aus genetischen Algorithmen und SVMs. Dies führte zu guten Erkennungsergebnissen, da SVM ein überwachtes Verfahren ist, waren in der Trainingsmenge auch Angriffe enthalten und die Fehlalarmquote war relativ hoch. In einem anderen Ansatz verglichen Mukkamala *et al.* [34] die Ergebnisse eines SVM-basierten Erkennungsmechanismus mit den Ergebnissen, die durch die Klassifikation durch neuronale Netze erhalten wurden und fanden für beide Verfahren gute Ergebnisse, wobei jedoch die SVM-basierte Methode ressourcenschonender ist.

Erste Erfahrungen mit einem kombinierten Detektionsverfahren wurden ebenfalls von Mukkamala *et al.* [36] beschrieben. Sie verknüpften SVMs, neuronale Netze und das statistische Verfahren MARS (Multivariate Adaptive Regression Splines) mit verbesserten Ergebnissen linear miteinander.

Die bisherigen Lernverfahren wurden in den meisten Fällen mit Trainingsdaten getestet, die sowohl Angriffsdaten als auch normale Daten enthielten. Es wurde vor allem der KDD Cup 1999 Datensatz[3] verwendet. Dieser enthält vier verschiedene Hauptklassen von Angriffen, die von den Lernverfahren richtig klassifiziert werden müssen. Allerdings enthalten die Testdaten Unterklassen, die nicht in den Trainingsdaten vorkommen. Analog zu signaturbasierten Systemen können daher völlig neue Angriffe eventuell nicht korrekt klassifiziert werden. Aus diesem Grund werden neben den bereits erwähnten künstlichen Immunsystemen auch unüberwachte Lernverfahren wie Self-Organizing Map (SOM) [2, 39] oder Clustering [19, 38] für Anomalieerkennung in Kombination mit angriffsfreien Trainingsdaten verwendet. Hierfür werden mit den Trainingsdaten Cluster gebildet und während der Detektion der Abstand zu diesen Clustern ermittelt. Befinden sich Feature Vektoren außerhalb der normalen Cluster, so werden sie als anormal angenommen.

Kooperative Intrusion Detection-Systeme Neben den oben beschriebenen Algorithmen ist vor allem der Aufbau des Immunsystems für dessen Robustheit und Fehlertoleranz entscheidend. Es gibt keine zentrale Steuereinheit, deren Ausfall das Funktionieren des gesamten Systems gefährden würde, sondern seine hochgradige Robustheit resultiert aus dem mehrschichtigen Aufbau und einem hohen Grad an Kooperation und Regulation unter den Elementen[21].

Diesen Aspekt des biologischen Immunsystems auf ein Netzwerksicherheitssystem abzubilden würde bedeuten, ein System aus vielen kleinen Entitäten zu entwickeln, die unterschiedliche Aufgaben haben, aber untereinander kooperieren.

Einen anderen Kooperationsansatz verfolgen Yang *et al.* [47] und Mukkamala *et al.* [35]. In diesen Systemen werden die Ergebnisse unterschiedlicher Detektionskomponenten kombiniert. Yang *et al.* [47] beschreibt den naheliegenden Ansatz die Vorteile von „misuse detection“ und „anomaly detection“ miteinander zu kombinieren. Das System verknüpft ein Expertensystem zur signaturbasierten Erkennung mit einem Anomalieerkennungsalgorithmus, der auf Basis von Clusteranalyse arbeitet. Außerdem werden die Ergebnisse der Anomalieerkennung in Regeln für das Expertensystem überführt, um erneute Angriffe schneller und sicherer zu erkennen. Durch diese Kombination konnte die Detektionsrate deutlich erhöht werden.

Ebenfalls eine Verbesserung des Gesamtergebnisses konnten Mukkamala *et al.* [35] durch die Kombination von verschiedenen Anomalieerkennungsverfahren erreichen. Eine weitere Steigerung wird durch intelligentere Verknüpfung der Verfahren erwartet. Ein vielversprechender Ansatz hierfür ist das sogenannte „Ensemble Learning“, indem verschiedene schwächere Klassifizierer oder Clusterer zu einem besseren kombiniert werden.

[3]Der KDD Cup ist ein Data Mining Wettbewerb, der von der ACM SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining) organisiert wird. Der Datensatz (`http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`) ist eine Version des DARPA Datensatzes (Lincoln Laboratory, Massachusetts Institute of Technology, `http://www.ll.mit.edu/IST/ideval/data/data_index.html`)

Ensemble Learning Unter dem Begriff „Ensemble Learning“ fasst man Verfahren zusammen, die nicht nur einen Klassifikator oder Clusterer in die Entscheidung über ein Problem einbeziehen, sondern mehrere („Ensemble“). Die beiden bekanntesten Ensemble Learning-Algorithmen sind Bagging [8] und Boosting [33, 40]. Bagging basiert auf der Annahme, dass durch die Verwendung einer Reihe von unabhängig erzeugten, kleineren Trainingsmengen ein besseres Lernergebnis bei überwachten Lernverfahren erzielt wird, als mit einer einzigen großen Trainingsmenge [8]. Aber auch im Bereich des unüberwachten Lernens wurde Bagging schon von Dudoit *et al.* genutzt[14]. Sie verwendeten Bagging zur Optimierung des Clusterings von Micro-Array Daten in der Bioinformatik.

Boosting hingegen verwendet verschiedene schwache Klassifikatoren eines Algorithmus, die unterschiedlich gewichtet werden. Während des Trainings werden aufgrund der falsch klassifizierten Trainingsdaten Gewichtungen für die einzelnen Klassifikatoren ermittelt; hierbei wird besonderes Augenmerk auf schwer klassifizierbare Daten gelegt; d.h. Klassifikatoren, die in der Lage sind, schwer einzuordnende Daten korrekt zu klassifizieren, erhalten ein höheres Gewicht. Diese Verfahren können für die Verknüpfung verschiedener Detektionseinheiten auf einem Knoten sowie für die Kooperation zwischen den Knoten genutzt werden.

Wie in der Einleitung beschrieben sind die heutigen IDS, wenn sie signaturbasiert arbeiten, sicher und schnell in der Erkennung bekannter Angriffe, bei neuen Angriffen jedoch haben sie große Probleme. Alternativ können mit den vorgestellten Algorithmen Anomalien erkannt werden. Dieses Verfahren ist jedoch sehr anfällig für Fehlalarme und somit auch nur bedingt einsetzbar. Ziel sollte es sein, Systeme zu entwickeln, die zum einen die Vorteile der signaturbasierten und anomaliebasierten Systeme verknüpfen, sowie durch die Verwendung mehrerer Verfahren des Maschinellen Lernens die Anzahl der Fehlalarme zu minimieren. Hierfür bieten sich die oben erwähnten Verfahren zum Verknüpfen der Detektionsergebnisse und eine ausfallsichere Kommunikation wie zum Beispiel innerhalb eines P2P-Netzwerkes an. Eine solche Overlay-Struktur bietet auch die Möglichkeit verteiltes Lernen, wie im Boosting Algorithmus von Lazarevic *et al.* [27] beschrieben, anzuwenden und so die aufwendigen Berechnungen großer Datenmengen zu verteilen.

2.2 Peer-to-Peer (P2P) Systeme

P2P-Applikationen haben neben dem klassischen File Sharing[4] auch in vielen andern Gebieten wie der IP-Telefonie[5] oder dem Anbieten von TV-Inhalten Einzug gehalten[6]. Die Vorteile von P2P-Systemen für Verteilte Angriffserkennung werden von König [25] anhand ihrer Schlüsseleigenschaften aufgezeigt:

Die **dezentrale Struktur** von Peer-to-Peer-Systemen erlaubt, Auswertungen an wenig belastete Knoten zu delegieren und erhöht die Skalierbarkeit eines Systems. Insbesondere bei sehr großen Netzen unterstützen hier *Partially Centralized Architectures* mit dynamischen Super-Peers [3]. Die **Fehlertoleranz** wird durch die Abwesenheit eines Single-Point-Of-Failures erhöht; bei Ausfall eines Knotens können Auswertungen von anderen Knoten übernommen

[4] Z.B. `http://www.gnutella.com`

[5] Z.B. `http://www.skype.com`

[6] Z.B. `http://www.joost.com`

werden. Die **Selbstorganisation** schließlich, erlaubt die dynamische Aufteilung anstehender Aufgaben auf die Peers.

Trotz ihrer Vorteile existieren nur vereinzelte Sicherheitssysteme, die auf P2P-Technologien bzw. Prinzipien basieren. Vlachos *et al.* [44] stellen ein kooperatives Intrusion Detection-System vor, welches auf JXTA[7] basiert. Es justiert anhand der Frequenz von Angriffsereignissen die Sicherheitseinstellungen der Browser der teilnehmenden Knoten.

Janakiraman *et al.* [20] beschreiben konzeptionell ein System, in dem alle Teilnehmer eines Overlay-Netzwerkes über Angriffsversuche benachrichtigt werden. Bei Peer-to-Peer Systemen stellt insbesondere die Vertrauensverwaltung eine große Herausforderung dar. Hierfür wird von den Autoren ein *Web of Trust*-Ansatz zwischen den teilnehmenden Knoten vorgeschlagen. Xiong *et al.* stellen ein reputationsabhängiges, feedback-basiertes Vertrauensmodell auf, das auf besondere Weise die Dynamik von P2P-Systemen adressiert [46].

Des Weiteren präsentieren Zhou *et al.* ein System, welches die Vorteile strukturierter, auf verteilten Haschtabellen basierender Systeme ausnutzt: durch einen globalen *Publish-Subscribe* Mechanismus, werden verdächtige IP-Adressen in einem Overlay-Netzwerk eindeutig zugeordnet. Die “Meldung” einer IP Adresse erfolgt dezentral, in logisch verschiedenen Netzwerkabschnitten. Die global eindeutige Zuordnung erlaubt die Unterscheidung, ob eine verdächtige IP Adresse ein lokales Problem darstellt oder ein Teil eines Massenphänomens ist [50]. Auch das DOMINO System stellt einen globalen Ansatz zur Organisation von Angriffsinformation dar. Hierbei tauschen sogenannte “Axis Nodes”, ausgewählte Knoten in Netzwerken, z.B. die Master- Knoten eines hierarchischen Intrusion Detection Systems, über ein Overlay Nachrichten aus. Ein konkretes Anwendungsszenario stellt hier ebenfalls der Austausch von Blacklists von IP Adressen dar [48].

Gruppenorganisation, abhängig von den Interessen der Teilnehmer oder einem konkreten Sicherheitsszenario, spielt ebenfalls eine wichtige Rolle. Dieses wird bisher beim Einsatz von Peer-to-Peer Technologie in der Computer- und Netzwerksicherheit wenig betrachtet. Existierende Arbeiten in der allgemeinen Forschung zum Thema Peer-to-Peer Netzwerke sind klassischerweise motiviert durch Effizienzsteigerungen bei Suchoperationen. Hierfür beschreiben Loeser *et al.* [30] einen Ansatz zur Gruppenformation basierend auf ähnlichen Eigenschaften von Teilnehmern eines P2P-Netzwerkes. Des Weiteren schlagen Khambatti *et al.* [22] eine Möglichkeit zur Strukturierung von P2P-Overlay-Netzwerken auf Grundlage der Interessen der einzelnen Teilnehmer vor. Ein effizienter Algorithmus zur verteilten Suche nach Teilnehmern mit ähnlichen Eigenschaften wird hier ebenfalls vorgeschlagen. Dieser basiert auf *Bloom Filter*-Datenstrukturen, die sich insbesondere für die Überprüfungen von Inklusionsbeziehungen eignen, die bei der Suche nach Knoten mit gleichen Eigenschaften eine zentrale Rolle spielen.

2.3 Adaptive Konfiguration

Im Rahmen einer Autonomous Security-Architektur müssen die Detektionskomponenten unter Berücksichtigung eines beschränkten Ressourcenbudgets im Netzwerk verteilt werden. Dieses *Deployment* auf die Netzknoten soll eine möglichst effiziente Detektion ermöglichen und

[7] https://jxta.dev.java.net/

muss vordefinierten Optimalitätskriterien genügen. Diese Kriterien sollen gewährleisten, dass für die Detektion verfügbare Ressourcen optimal eingesetzt werden; auf diese Weise sollen beispielsweise redundante Operationen wie das mehrfache Untersuchen des gleichen Netzwerkverkehrs entfallen oder die zu verwendenden Detektionskomponenten der Art des beobachteten Verkehrs angepasst werden. Bloem *et al.* [6] haben untersucht, wie sich ein solches so genanntes *Monitor Placement Problem* als Problem der *linearen Programmierung* formulieren lässt. Unter Berücksichtigung des Angreiferverhaltens sind von Kodialam *et al.* auch auf spieltheoretischen Konzepten basierende Ansätze untersucht worden [24].

Üblicherweise unterscheidet man zwischen *statischer* und *dynamischer* Konfiguration [43]. Die Berechnung des ursprünglichen Deployments der Detektionskomponenten bzw. einer Rekonfiguration unabhängig von Laufzeitparametern wie beobachteter Netzwerkverkehr oder Angriffsindikationen sind klassische statische Konfigurationsmaßnahmen.

Dynamische Konfigurationsmaßnahmen hingegen sind von zeitlichen Änderungen bestimmter Parameter der Detektionsarchitektur abhängig. Wird beispielsweise von einer Detektionskomponente ein erhöhter Anteil infizierten Verkehrs registriert, so erfordert diese Situation eine Reaktion des IDS. Gegenmaßnahmen können dabei der Einsatz weiterer Detektionskomponenten in der Umgebung oder die Erhöhung der Sampling-Rate der in Frage stehenden Komponente darstellen. Ein spezifikationsbasierter Ansatz zur automatisierten Einleitung von Gegenmaßnahmen wurde von Balepin *et al.* beschrieben [5].

Neben der Relevanz für die Detektionseffizienz wirkt sich das Deployment auch auf die Kooperation zwischen den Komponenten aus. So kann z.B. bei bestimmten Kooperationsverfahren die Entfernung zwischen kommunizierenden Peers für die Geschwindigkeit der Nachrichtenübermittlung entscheidend sein, so dass abhängig vom verwendeten Routing-Protokoll und der Netzwerktopologie möglichst kurze Kommunikationswege zwischen den Knoten der Sicherheitsarchitektur gewährleistet werden müssen. Dies kann unter Verwendung von *Betweenness Centrality*-Algorithmen, die von Brandes [7] beschrieben wurden, berechnet werden. Bloem *et al.* haben gezeigt, dass sich dieser Ansatz auf IP-basierte Computernetzwerke übertragen lässt [6].

In den oben genannten Kontexten können mit Hilfe von Verfahren aus Optimierungs- und Spieltheorie optimale Deployments von Sicherheitskomponenten in einem Netzwerk anhand vorgegebener Kosten-/Nutzenfunktionen berechnet werden. Die Klasse solcher *Monitor Placement Problems* wurde beispielsweise von Cantieni *et al.* [12] untersucht. Es existieren eine Vielzahl von mathematischen Standardverfahren, beispielsweise aus den Bereichen der linearen und konvexen Optimierung, anhand derer sich Deployment-Lösungen numerisch effizient berechnen lassen.

Heutige IDS arbeiten größtenteils starr regelbasiert und beziehen nicht die topologischen Eigenschaften des Netzwerks bei der Auswahlentscheidung von Gegenmaßnahmen mit ein. Die Herausforderung besteht darin, die realen Netzwerke korrekt mathematisch zu modellieren und insbesondere die Zielfunktion und die einschränkenden Nebenbedingungen des Monitor Placement-Problems so zu formulieren, dass es die realen Bedingungen innerhalb des Netzwerks widerspiegelt, um optimale Detektionsergebnisse und Reaktionsmechanismen innerhalb des Sicherheitsarchitektur zu erreichen.

3 Die Autonomous Security-Architektur

ASA stellt eine neuartige, auf spieltheoretischen und biologisch motivierten Ansätzen basierende Sicherheitsarchitektur dar. Der Einsatz kooperativer Lernverfahren und Algorithmen ermöglicht dabei eine flexible Behandlung bekannter Angriffsmuster sowie eine Reduktion der Fehlerrate bei der Erkennung neuartiger Angriffe. Diese Verfahren fassen wir unter dem Begriff *Autonomous Security* zusammen.

Abbildung 1 zeigt beispielhaft zwei ASA-Knoten, die über eine P2P-Schnittstelle miteinander kommunizieren können. ASA-Knoten bestehen aus fünf verschiedenen Komponenten: Die *Detektionskomponente* erlaubt die Integration von 1..*n* sogenannten *Detection Units* (DU), die einen beliebigen Erkennungsalgorithmus, anomalie- oder signaturbasiert darstellen können. Die Eingabe an dieses Modul erfolgt über Sensoren, wie z.B. Netzwerkverkehr-Sniffer oder Betriebssystemaufrufmonitore. Die Komponente stellt diese Daten verwalteten Erkennungsalgorithmen zur Verfügung. Zusätzlich werden Rückgabewerte einzelner DUs auch wieder über die Detektionskomponente zur Verfügung gestellt. Dies erlaubt dann die Kombination von Algorithmen um beispielsweise wie beim Ensemble Learning (vgl. Abschnitt 2.1) mehrere “schwache” Klassifikatoren zu kombinieren.

Die Evaluationskomponente bewertet die Eingaben aus dem Erkennungsmodul, hat Zugriff auf die Kommunikationsschnittstelle und spricht Empfehlungen für die “Configuration & Response”-Komponente aus. Aus diesen Empfehlungen, kombiniert mit vorher festgelegten Security Policies, erfolgt eine (Neu)-Konfiguration oder Initiierung von Gegenmaßnahmen; Beispiele für diese Policies sind von einem Administrator festgelegte IP Addressbereiche, mit denen eine Kooperation erlaubt ist, oder explizite White- oder Blacklists, die generell oder abhängig von einem verwendeten Protokoll festgelegt sind. Die Kooperationskomponente (“Cooperation Component”) erlaubt die Bildung von Detektionsgruppen. Diese stellen einen Zusammenschluss von Teilnehmern mit einem gemeinsamen Interesse, z.B. einem gemeinsamen, verteilten Detektionsalgorithmus, dar. Zur Formulierung dieses Interesses existiert eine Ontologie, mittels der Knoten einerseits selber beschrieben sind aber auch andererseits potenzielle Kooperationspartner spezifizieren können. Konkrete Bereiche dieser Ontologie sind bisher Hardware-Eigenschaften, Betriebssystem, Software-Konfiguration sowie Netzwerkeigenschaften und verwendete Detektionsalgorithmen. Eine weitere Verbesserung der Skalierbarkeit und Robustheit von ASA wird durch den Einsatz von Peer-to-Peer-Techniken erreicht. Zudem wird die Entstehung eines Single-Point-of-Failure, der in klassischen hierarchischen Intrusion Detection-Systemen zu finden ist, vermieden. Die Anwendung der genannten Techniken versetzt eine Autonomous Security-Architektur somit in die Lage, neue Angriffe zu erkennen und Gegenmaßnahmen an die jeweilige Situation anzupassen. Für den Test und die Bewertung der Algorithmen verwenden wir den von uns entwickelten Netzwerksicherheitssimulator *NeSSi* (vgl. Abschnitt 5). In den folgenden Abschnitten werden kurz bisherige Arbeiten in der Domäne *AS* vorgestellt:

3.1 Kooperatives Künstliches Immunsystem mit P2P Kommunikation

Die verteilte Detektion kann beispielsweise durch ein kooperativ arbeitendes *Künstliches Immunystem* (AIS) realisiert werden, wie es von uns in [31] beschrieben wurde. Dabei tauschen

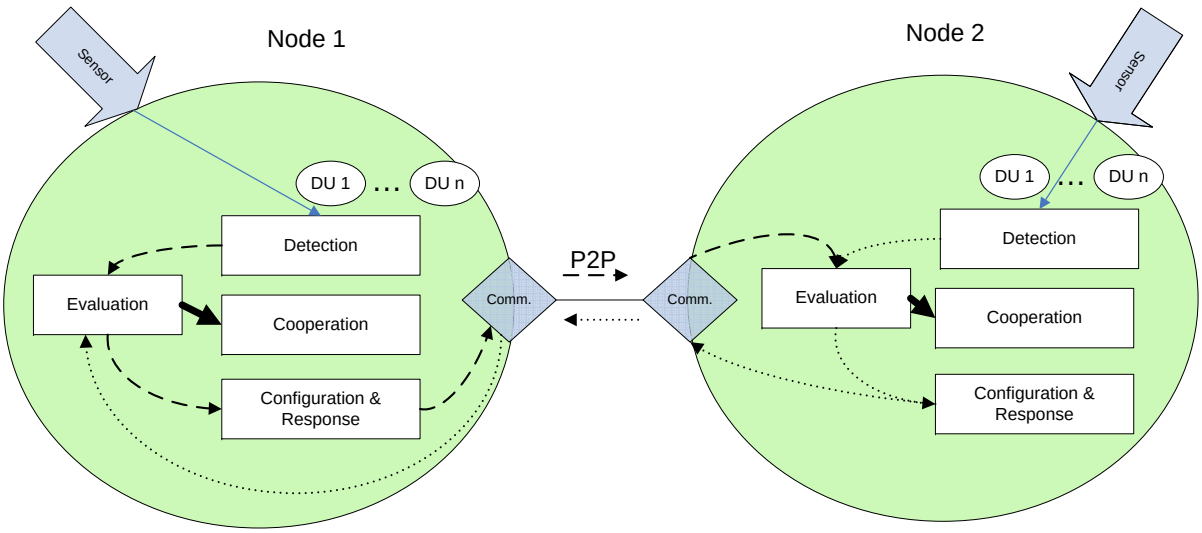

Abbildung 1: *Autonomous Security-Architektur (ASA)*. Jeder beteiligte Knoten ist mit Funktionalität zur Erkennung von Anomalien bzw. Angriffen ausgestattet. Die Netzwerkknoten können basierend auf einer P2P-Infrastruktur untereinander Informationen austauschen, um Detektionsereignisse zu evaluieren und gegebenenfalls Gegenmaßnahmen einzuleiten.

Agenten, die mit einem AIS ausgestattet sind über ein Peer-to-Peer-System Statusinformationen aus. Dies ist beispielsweise ein Infektionsstatus, der mittels zweier unterschiedlicher Ausprägungen des AIS-Algorithmus auf jedem Knoten berechnet wird. In einer Gruppe von Knoten innerhalb eines Peer-to-Peer Systems können die verschiedenen Agenten sich nun austauschen und basierend darauf einen globalen Infektionsstatus ermitteln. Dieses System wird mittels verschiedener Angriffsszenarien wie Wurmangriffen bzw. bei Hacking-Versuchen typischen Port Scanning-Aktivitäten evaluiert. Als Resultat kann hier die Fehlalarmquote deutlich gesenkt werden, die ein typisches Problem bei Anomaliedetektionsverfahren darstellt. Eine weitergehende Verbesserung der Erkennungsergebnisse wird durch die Einbindung weiterer Verfahren des Maschinellen Lernens und Ensemble Learnings erreicht.

3.2 Spieltheoretische Verteilungsmodelle für Detektionskomponenten

Bei der Verteilung von Detektionskomponenten in netzwerkbasierten IDS muss davon ausgegangen werden, dass dem Angreifer die Position und Konfiguration von Detektionskomponenten innerhalb des Netzwerks zu jedem Zeitpunkt bekannt ist, und er rational und intelligent handelt, um diese zu umgehen. Diese Annahme motiviert ein spieltheoretisches Konzept zur Verteilung von Detektionskomponenten. Ein solches Konzept wurde von Alpcan *et al.* entwickelt [42]. Darin wird ein Ansatz beschrieben, wie ein Intrusion Detection-System als nicht kooperatives Zwei-Personen-Spiel modelliert und berechnet werden kann. Dabei werden symmetrische Auszahlungsmatrizen für beide Spieler verwendet, welche die numerische Quantifizierung des Nutzens eines erfolgreichen Angriffs bzw. dessen Abwehr ermöglicht. Ein entscheidender Vorteil der Spieltheorie besteht darin, dass bei korrekter Domänenmodellierung eine Garantie gegeben werden kann, dass unter Annahme lokaler Erkennungsraten an den einzelnen Knoten eine feste Mindesterkennungsrate im gesamten Netzwerk erreicht wird. Dies ist insbesondere unabhängig von der gewählten Strategie des Angreifers.

4 Autonomous Security Anwendungsszenarien

Anhand dreier Szenarien werden wir im Folgenden praktische Einsatzmöglichkeiten der vorgestellten Autonomous Security Architektur erläutern.

4.1 Autonomous Security im Heimbereich

Das erste Szenario behandelt die Absicherung eines Heimnetzwerkes, welches aus unterschiedlichsten, IP-fähigen Endgeräten (PC, Fernseher, Herd, Steckdosen, ...) besteht und indem eine Vielzahl von Diensten (Überwachung und Steuerung des Energieverbrauchs, personalisierte Informationsdienste, ...) angeboten werden. Ein solches Heimnetzwerk wird beispielsweise in [15] vorgestellt. Nutzer eines solchen Heimnetzwerkes sind im Allgemeinen keine IT- oder Sicherheitsexperten, von daher benötigen sie eine Sicherheitslösung, die autonom Angriffe erkennen, erste Reaktionen einleiten und Nachrichten an Nutzer und Anbieter der Heimnetzwerklösung senden kann. Eine beispielhafte Absicherung eines solchen Netzwerkes mit Hilfe eines Systems auf Basis von ASA ist in Abbildung 2 dargestellt.

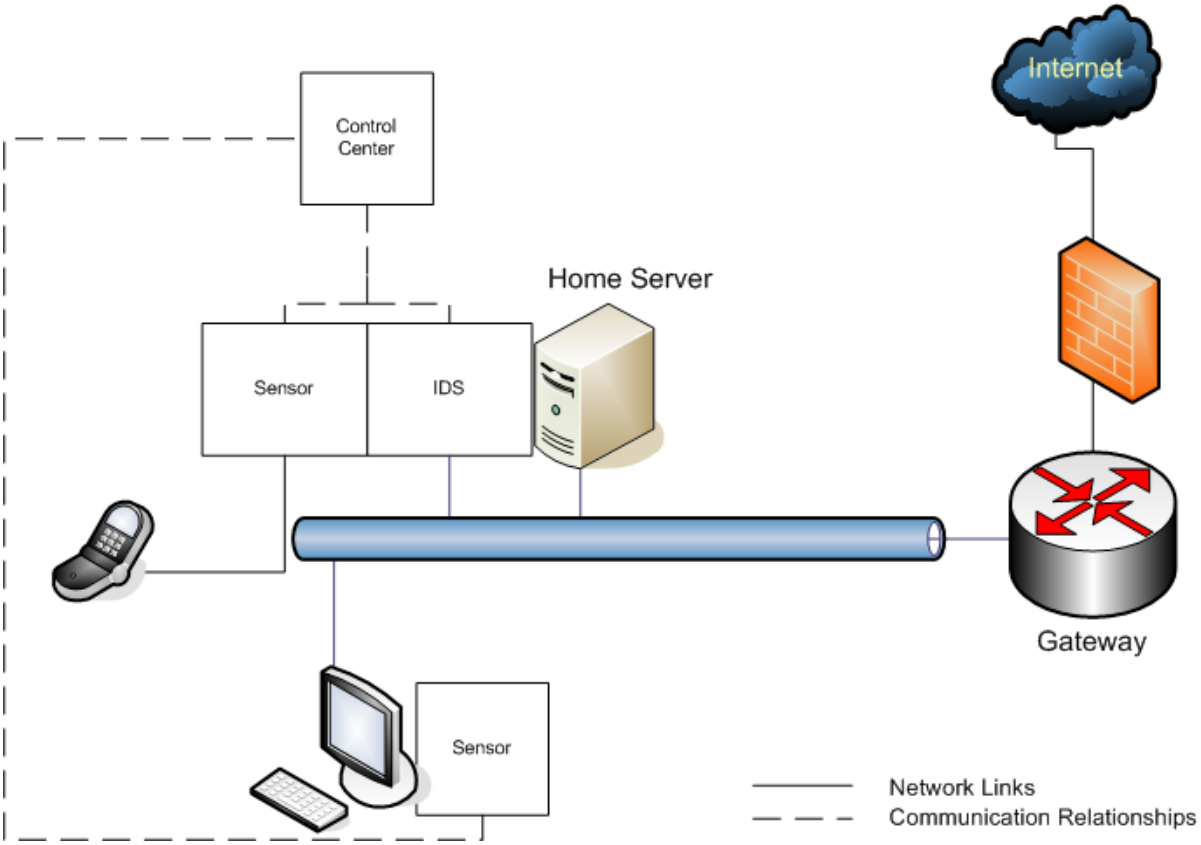

Abbildung 2: AS-System in einem Heimnetzwerk

Dieses AS-System besteht aus unterschiedlichen Sensoren, die ihre Detektionsergebnisse untereinander abstimmen und einen zentralen Agenten (*Control Center*) im Heimnetzwerk benachrichtigen, falls sie Angriffe erkannt haben. Dieser Agent wiederum ist dafür zuständig,

Benachrichtigungen an Nutzer und Betreiber zu versenden und das Einleiten von Gegenmaßnahmen zu koordinieren.

Zur Erkennung von Angriffen auf ein Heimnetzwerk werden unterschiedliche Sensortypen eingesetzt:

- **Sicherheitsprodukt-Sensor:** Hierbei handelt es sich um einen Sensor, der gängige Sicherheitsprodukte (Firewall, kommerzielles IDS, Anti-Virus Software, ...) überwacht, beispielsweise indem er eine Firewall-Log-Datei analysiert.
- **Netzwerkdatenverkehr-Sensor:** Der Sensor überwacht den Datenverkehr im Netzwerk.
- **Angriffsmuster-Sensor:** Sucht nach Mustern für bekannte Angriffe, beispielsweise in Log-Dateien von Anwendungen oder als Anwendungs-Firewall.
- **Anomalie-Sensor:** Dieser werden Messwerte eines Endgerätes oder eines Netzwerkdatenverkehr-Sensors verwendet um Anomalien zu erkennen, die auf einen Angriff hindeuten.

Zur Verbesserung der Erkennungsergebnisse, insbesondere zur Reduzierung von Fehlalarmen, kooperieren die Sensoren miteinander, bevor der *Control Center*-Agent über einen Angriff informiert wird. Dieser wiederum kann sich der Entscheidung der Sensoren anschließen oder zusätzliche Informationsquellen (lokale Sicherheitsrichtlinien oder Benachrichtigungen des Heimnetzanbieters über aktuelle Angriffe/Schwachstellen) und Entscheidungsstrategien (Gewichtungen für Sensoren, Kooperation mit *Control Center*-Agenten in anderen Heimnetzwerken) nutzen, um eine endgültige Entscheidung zu treffen. Ein erster Prototyp des beschriebenen Szenarios wird aktuell realisiert.

4.2 Autonomous Security für kleine und mittelständische Unternehmen

Ein alternatives Einsatzgebiet für ein AS-System sind kleine und mittelständische Unternehmen (KMU). Diese Unternehmen besitzen teilweise kein dediziertes Personal für die Administration und Überwachung der Sicherheitsinfrastruktur eines Unternehmens. Ähnlich wie bereits in Heimnetzwerken können AS-Sensoren in einem KMU-Netzwerk installiert werden. Jedoch fällt hier dem *Control Center*-Agenten eine wichtigere Rolle zu.

Bei der Kooperation mit Agenten in Netzwerken in einem anderen Unternehmen, muss darauf geachtet werden, dass keine vertraulichen Daten preisgegeben werden. Weiterhin bietet es sich an, in einem solchen System eine Zusammenarbeit mit einer IT-Sicherheitsberatungsfirma zu realisieren, die in den Prozess der Bewertung von erkannten Anomalie einbezogen wird und für die Umsetzung von Gegenmaßnahmen zuständig ist, die nicht automatischen eingeleitet werden.

4.3 Autonomous Security und kritische Infrastrukturen

In den bisher vorgestellten Szenarien erfolgte der Einsatz von Anwendungen auf Basis von in relative überschaubaren Netzen und die Administration eines Autonomous Security Systems

erfolgte durch eine zentrale Stelle. Dies ist jedoch keine notwendige Bedingung für ein System auf der Basis von ASA. Ein komplexeres Einsatzgebiet für ASA ist die Realisierung eines Systems zur Überwachung von kritischen Infrastrukturen [9].

Ziel eines derartigen Systems ist es Angriffe frühzeitig zu erkennen und Gegenmaßnahmen einzuleiten. Im Prinzip unterscheidet sich dieser Ansatz nicht von dem für KMUs, jedoch gibt es Unterschiede bezüglich der Skalierung und der benötigten Strategien bei der Risikobewertung. Sind im KMU-Szenario die einzelnen KMUs (mehr oder weniger) unabhängig voneinander, muss bei der Risikobewertung für kritische Infrastrukturen die Abhängigkeiten zu anderen kritische Infrastrukturen mit berücksichtigt werden. Die IT-Systeme einer Bank sind beispielsweise auf Operator-Netzwerke angewiesen und beide wiederum benötigen eine funktionierende Energieversorgung; andersherum besteht auch eine Abhängigkeit vom Energieversorger zu einem Operator-Netzwerk.

Um Angriffe auf IT-Systeme einer kritischen Infrastruktur zu erkennen werden innerhalb der Netzwerke und auf ausgewählten Endgeräten Sensoren installiert. Hierbei handelt es sich wieder um die schon bekannten Netzwerkverkehrs-, Anomalie- und Angriffsmustersensoren. Der Hauptunterschied zu den vorherigen Szenarien besteht in den Auswertungs- und Entscheidungsstrategien. Durch den Einsatz des Serviceware Framework JIAC wird der Austausch dieser Strategien erleichtert. Die von JIAC Agenten verwendeten Agentenbeschreibungssprache JADL [26] ist darauf ausgelegt, Zielbeschreibungen festzulegen und auszutauschen.

Welche Besonderheiten müssen Agenten in einem AS-System in ihren Strategien berücksichtigen? Bei der Analyse von Sensordaten muss eine Bewertung erfolgen, wie groß die Gefährdung für die existenziell wichtige IT-Systeme einer kritischen Infrastruktur ist, welche Teile des IT-Netzes funktionstüchtig bleiben muss und welche Teile verzichtbar sind. Im Falle eines DDoS Angriffes auf Server in Estland [18, 28] wurde als Gegenmaßnahme die Anbindung an das weltweite Internet getrennt [13]. Die Entscheidung, sich vom weltweiten Internet zu trennen, verdeutlicht die Notwendigkeit Entscheidungen nicht nur lokal für eine Infrastruktur zu treffen, sondern diese Entscheidungen zwischen Infrastrukturbetreibern zu koordinieren. Entweder wie in [9] vorgeschlagen über eine zentrale Instanz oder durch dezentrale Kooperationsverfahren zwischen Agenten innerhalb der einzelnen Infrastrukturen.

Im Falle von kritischen Infrastrukturen lassen sich die effektivsten Entscheidungsstrategien und die benötigten Daten für eine Kooperation mit anderen Infrastrukturbetreibern kaum durch theoretische Überlegungen und praktische Tests zuverlässig feststellen. Mit dem Netzwerksicherheitssimulator NeSSi lassen sich unterschiedliche Entscheidungsstrategien und Angriffsszenarien simulieren, bevor sie Anwendung in einer realen Infrastruktur finden.

5 Netzwerksicherheitssimulator NeSSi

Der am DAI-Labor entwickelte *Ne*twork *S*ecurity *Si*mulator *NeSSi* ermöglicht es, zu Testzwecken Verkehrsdaten auf Paketebene für die Bewertung von Sicherheitsarchitekturen zu generieren. Auf diese Weise ist es möglich, detaillierte Verkehrsanalysen vorzunehmen sowie mittels parametrisierte Profile automatisch Angriffsszenarien zu generieren und auszuwerten. Abbildung 3 zeigt die grafische Oberfläche des Simulators. NeSSi wurde auf Basis des Agenten-Frameworks JIAC [17] entworfen und weist daher eine verteilte und erweiterba-

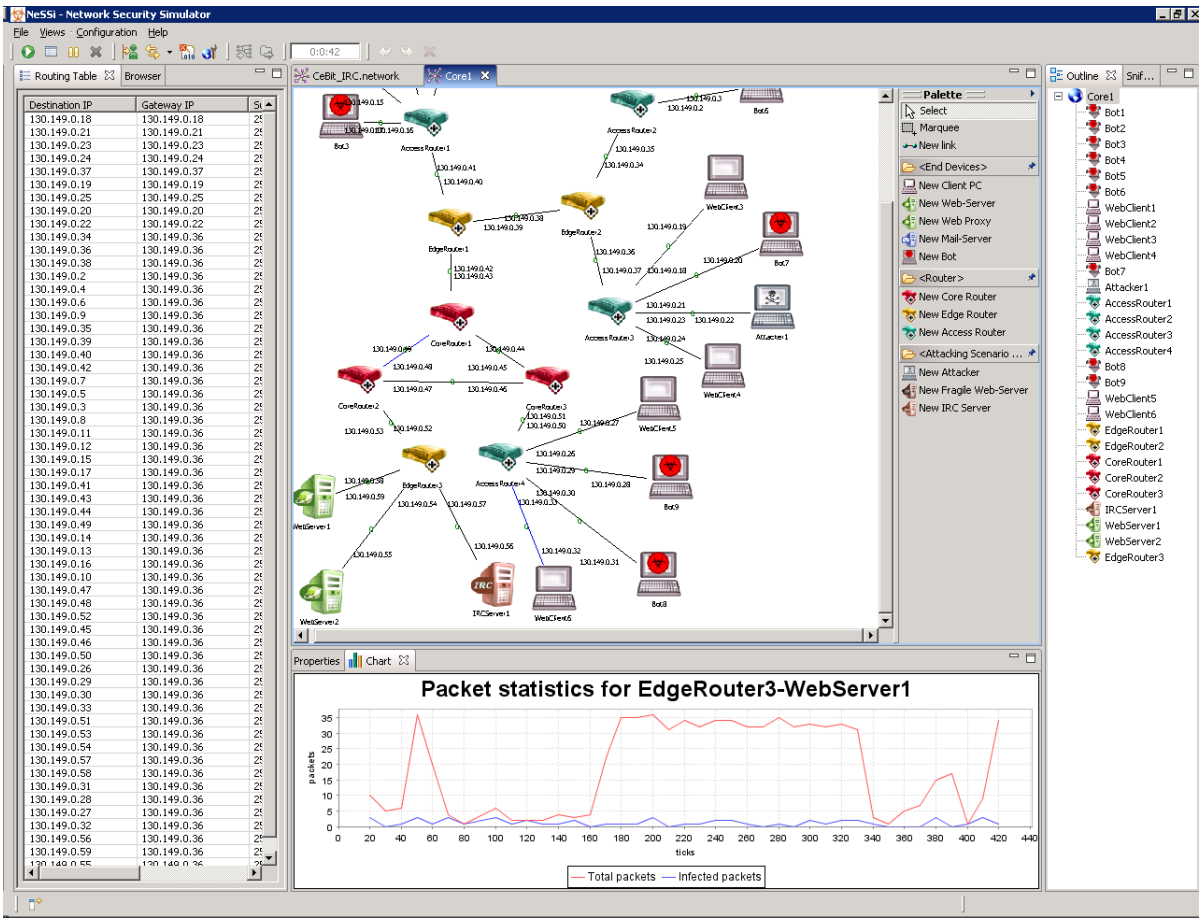

Abbildung 3: Grafische Oberfläche des Netzwerksicherheitssimulators NeSSi

re Architektur auf. Eine standardisierte Schnittstelle erlaubt die Einbindung sowohl signatur- als auch anomaliebasierter Erkennungsalgorithmen und stellt somit eine einfach zu nutzende Testumgebung für netzwerkbasierte Intrusion Detection-Systeme dar. NeSSi ist ein paketbasiertes Simulationstool und unterscheidet sich von anderen Simulationstools für Netzwerksicherheit (z.B. [45], [29], [49]) insbesondere durch seine umfassende Unterstützung von Szenarien auf der Anwendungsschicht des TCP/IP-Standardschichtenmodells. NeSSi erlaubt unter anderem die Ausführung von SMTP-, HTTP- und IRC-Protokollen und -Szenarien; nicht standardmäßig unterstützte Protokolle können über eine dokumentierte Plugin-Schnittstelle eingebunden werden. NeSSi hat sich sowohl in Forschung und Lehre als Evaluationstool bewährt. Unter anderem wurde, wie in Abschnitt 3.1 und 3.2 beschrieben, ein kooperatives künstliches Immunsystem [31] und ein spieltheoretisches Verteilungsmodell [42] validiert und publiziert. Neben den in Abbildung 4 gezeigten in NeSSi integrierten Report zur Evaluierung unterschiedlicher Detektionsalgorithmen, können auch Reports zur Evaluierung verschiedener Deploymentstrategien genutzt werden. Detailliertere Informationen zu NeSSi finden sich in [11].

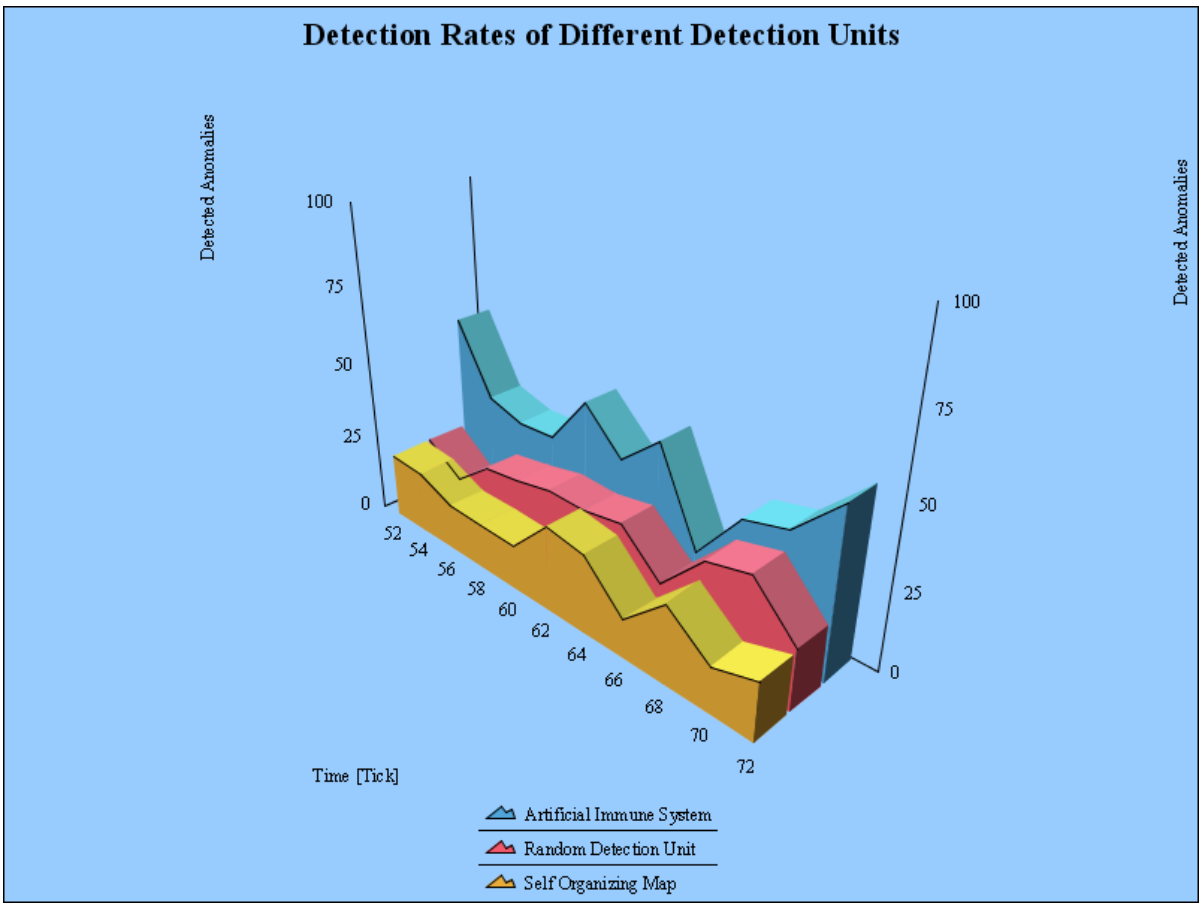

Abbildung 4: Auswertung von Detektionsalgorithmen

6 Zusammenfassung und Ausblick

In diesem Papier wurde ein erster Ansatz für eine Rahmenarchitektur für die Realisierung von Sicherheitsmechanismen vorgestellt, welche in der Lage sind, ihre Umgebung wahrzunehmen und mit den gesammelten Informationen, autonom auf Bedrohungen zu reagieren. Grundlegend besteht diese Architektur aus drei Teilsystemen: Anomalieerkennung, P2P-basierte Kooperation und dem reaktiven Deployment von Detektionskomponenten. Es werden Anwendungsszenarien vorgestellt, die aus dem Bereich Heimnetzwerke, kleine Unternehmensnetzwerke und kritische Infrastrukturen kommen. Abschliessend wird der Netzwerk Sicherheitssimulator NeSSi vorgestellt, auf dessen Basis die Funktionalität der Architektur evaluiert werden soll.

Aktuell liegen unsere Forschungsschwerpunkte in der Weiterentwicklung von ASA-Basisfunktionalitäten, wie z.B. dem Hinzufügen weiterer Sensoren, der Verwendung bereits erzielter Ergebnisse im Bereich der Erkennung von Malware-Ausbreitungen in großen Netzwerken [31, 10], sowie der Evaluation von mathematischen Verfahren für das effiziente Deployment von Detektionskomponenten. An der Integration weiterer verwandter Arbeiten [41] wird ebenfalls gearbeitet.

Erste ASA Versionen mit NeSSi wurden bereits im Lehrbetrieb eingesetzt, um es Studenten zu ermöglichen, Angriff, Detektion und Gegenmaßnahmen auf Basis Realitäts-naher Netzwerke zu simulieren.

Danksagungen

Teile dieser Arbeit wurden gefördert durch ein Projekt der Deutsche Telekom Laboratories und mit Mitteln des Bundesministeriums für Wirtschaft und Technologie unter dem Förderkennzeichen 01MG541 - 01MG547. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Weiterhin danken die Autoren den beiden Projektteams für Unterstützung und Diskussionen bei der Erstellung dieses Papiers, insbesondere bedanken wir uns bei Ahmet Camtepe, Arik Messerman und Sebastian Feuerstack. Weiterhin danken wir Tansu Alpcan für die gemeinsame Diskussionen und Anregungen bezüglich des Themas Autonomous Security.

Literatur

[1] AICKELIN, U. und S. CAYZER: *The Danger Theory and Its Application to Artificial Immune Systems*. In: TIMMIS, J. und P. J. BENTLEY (Hrsg.): *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS 2002)*, S. 141–148, Canterbury, UK, 2003. University of Kent at Canterbury Printing Unit.

[2] ALBAYRAK, S., C. SCHEEL, D. MILOSEVIC und A. MÜLLER: *Combining Self-Organizing Map Algorithms for Robust and Scalable Intrusion Detection*. In: MOHAMMADIAN, M. (Hrsg.): *Proceedings of International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA 2005)*, S. 123–130. IEEE Computer Society, 2005.

[3] ANDROUTSELLIS-THEOTOKIS, S. und D. SPINELLIS: *A survey of peer-to-peer content distribution technologies*. ACM Computing Surveys, 36(4):335–371, 2004.

[4] AXELSSON, S.: *Intrusion Detection Systems: A Survey and Taxonomy*. Techn. Ber. 99-15, Department of Computer Engineering Chalmers University of Technology Göteborg, Sweden, März 2000.

[5] BALEPIN, I., S. MALTSEV, J. ROWE und K. N. LEVITT: *Using Specification-Based Intrusion Detection for Automated Response*. In: VIGNA, G., E. JONSSON und C. KRÜGEL (Hrsg.): *Recent Advances in Intrusion Detection, 6th International Symposium, RAID 2003, Pittsburgh, PA, USA, September 8-10, 2003, Proceedings*, Bd. 2820 d. Reihe *Lecture Notes in Computer Science (LNCS)*, S. 136–154. Springer-Verlag, 2003.

[6] BLOEM, M., T. ALPCAN, S. SCHMIDT und T. BAŞAR: *Malware Filtering for Network Security Using Weighted Optimality Measures*. In: *Proc. of 2007 IEEE Multi-conference on Systems and Control*. IEEE, 2007. to appear.

[7] BRANDES, U.: *A Faster Algorithm for Betweenness Centrality*. Journal of Mathematical Sociology, 25(2):163–177, 2001.

[8] BREIMAN, L.: *Bagging predictors*. Machine Learning, 24(2):123–140, 1996.

[9] BSUFKA, K., O. KROLL-PETERS und S. ALBAYRAK: *Intelligent Network-Based Early Warning Systems*. In: LOPEZ, J. (Hrsg.): *Critical Information Infrastructures Security, First International Workshop, CRITIS 2006, Samos, Greece, August 31 - September 1, 2006, Revised Papers*, Bd. 4347 d. Reihe *Lecture Notes in Computer Science (LNCS)*, S. 103–111. Springer-Verlag, 2006.

[10] BYE, R., K. LUTHER, S. A. ÇAMTEPE, T. ALPCAN, ŞAHIN ALBAYRAK und B. YENER: *Decentralized Detector Generation in Cooperative Intrusion Detection Systems*. In: MASUZAWA, TOSHIMITSU; TIXEUIL, S. (Hrsg.): *Stabilization, Safety, and Security of Distributed Systems 9th International Symposium, SSS 2007 Paris, France, November 14-16, 2007 Proceedings*, Lecture Notes in Computer Science , Vol. 4838. Springer, 2008.

[11] BYE, R., S. SCHMIDT, K. LUTHER und S. ALBAYRAK: *Application-level simulation for network security*. submitted to SimuTools 2008, 2008.

[12] CANTIENI, G. R., G. IANNACCONE, C. BARAKAT, C. DIOT und P. THIRAN: *Reformulating the monitor placement problem: Optimal network-wide sampling*. Technical Report, Intel Research, Februar 2005.

[13] DONNER, M.: *Cyberassault on Estonia*. IEEE Security and Privacy, 5(4):4, 2007.

[14] DUDOIT, S. und J. FRIDLYAND: *Bagging to improve the accuracy of a clustering procedure*. Bioinformatics, 19(9):1090–199, 2003.

[15] FEUERSTACK, S., M. BLUMENDORF, G. LEHMANN und S. ALBAYRAK: *Seamless Home Services*. In: *Developing Ambient Intelligence. Proceedings of the First International Conference on Ambient Intelligence Developments (AmID'06)*, S. 1–10, 2006.

[16] FORREST, S., A. S. PERELSON, L. ALLEN und R. CHERUKURI: *Self-nonself Discrimination in a Computer*. In: *Proceedings of the IEEE Symposium on Research in Security and Privacy*, S. 202–212. IEEE Computer Society Press, 1994.

[17] FRICKE, S., K. BSUFKA, J. KEISER, T. SCHMIDT, R. SESSELER und S. ALBAYRAK: *Agent-based telematic services and telecom applications*. Communications of the ACM, 44(4):43–48, April 2001.

[18] GOTH, G.: *The Politics of DDoS Attacks*. IEEE Distributed Systems Online, 8(8):art. no. 0708–o8003, 2007.

[19] GUAN, Y., A. GHORBANI und N. BELACEL: *Y-means: A Clustering Method for Intrusion Detection*. In: *Canadian Conference on Electrical and Computer Engineering.*, Montréal, Québec, Canada., Mai 2003.

[20] JANAKIRAMAN, R., M. WALDVOGEL und Q. ZHANG: *Indra: A Peer-to-Peer Approach to Network Intrusion Detection and Prevention*. In: *Proceedings of IEEE WETICE 2003*, Juni 2003.

[21] JANEWAY, JR., C. A., P. TRAVERS, M. WALPORT und M. J. SHLOMCHIK: *Immunobiology : the immune system in health and disease*. Garland Publishing, New York, 2001.

[22] KHAMBATTI, M., K. RYU und P. DASGUPTA: *Structuring Peer-to-Peer Networks Using Interest-Based Communities*. In: ABERER, K., M. KOUBARAKIS und V. KALOGERAKI (Hrsg.): *Databases, Information Systems, and Peer-to-Peer Computing – First International Workshop, DBISP2P 2003 Berlin, Germany, September 7 - 8, 2003 Revised Papers*, Bd. 2944 d. Reihe *Lecture Notes in Computer Science (LNCS)*, S. 48–63. Springer, 2004.

[23] KIM, D. S. und J. S. PARK: *Network-Based Intrusion Detection with Support Vector Machines*. In: *Information Networking, Networking Technologies for Enhanced Internet Services - International Conference, ICOIN 2003, Cheju Island, Korea, February 12-14, 2003, Revised Selected Papers*, Bd. 2662 d. Reihe *Lecture Notes in Computer Science (LNCS)*, 2003.

[24] KODIALAM, M. und T. LAKSHMAN: *Detecting Network Intrusions via Sampling: A Game Theoretic Approach*. In: *Proceedings IEEE INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, Bd. 3, S. 1880–1889, Apr. 2003.

[25] KÖNIG, H.: *Peer-to-Peer Intrusion Detection Systeme für den Schutz sensibler IT-Infrastrukturen*. In: CREMERS, A. B., R. MANTHEY, P. MARTINI und V. STEINHAGE (Hrsg.): *GI Jahrestagung (2)*, Bd. 68 d. Reihe *LNI*, S. 638–642. Gesellschaft für Informatik, GI, 2005.

[26] KONNERTH, T., B. HIRSCH und S. ALBAYRAK: *JADL — an Agent Description Language for Smart Agents*. In: BALDONI, M. und U. ENDRISS (Hrsg.): *Declarative Agent Languages and Technologies IV*, Bd. 4327 d. Reihe *LNCS*, S. 141–155. Springer Verlag, 2006.

[27] LAZAREVIC, A. und Z. OBRADOVIC: *The distributed boosting algorithm*. In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, S. 311 – 316, San Francisco, California, 2001. ACM Press.

[28] LESK, M.: *The New Front Line: Estonia under Cyberassault*. IEEE Security and Privacy, 5(4):76–79, 2007.

[29] LILJENSTAM, M., J. LIU, D. NICOL, Y. YUAN, G. YAN und C. GRIER: *RINSE: The Real-Time Immersive Network Simulation Environment for Network Security Exercises*. PADS, 00:119–128, 2005.

[30] LOESER, A., F. NAUMANN, W. SIBERSKI, W. NEJDL und U. THADEN: *Semantic overlay clusters within super-peer networks*. In: ABERER, K., V. KALOGERAKI und M. KOUBARAKIS (Hrsg.): *Databases, Information Systems, and Peer-to-Peer Computing*, Bd. 3367 d. Reihe *Lecture Notes in Computer Science (LNCS)*, S. 33–47. Springer-Verlag, 2004.

[31] LUTHER, K., R. BYE, T. ALPCAN, S. ALBAYRAK und A. MÜLLER: *A Cooperative AIS Framework for Intrusion Detection*. In: *Proceedings of the IEEE International Conference on Communications (ICC 2007)*, 2007.

[32] MATZINGER, P.: *The Danger Model: A Renewed Sense of Self*. Science, 296(5566):301–305, April 2002.

[33] MEIR, R. und G. RÄTSCH: *An introduction to boosting and leveraging*. In: *Advanced Lectures on Machine Learning (LNAI2600)*, 2003.

[34] MUKKAMALA, S., G. JANOSKI und A. SUNG: *Intrusion Detection Using Neural Networks and Support Vector Machines*. In: *International Joint Conference on Neural Networks (IJCNN '02)*, 2002.

[35] MUKKAMALA, S. und A. H. SUNG: *Detecting Denial of Service Attacks Using Support Vector Machines*. In: *Proceedings of International Conference on Fuzzy Systems*, 2003.

[36] MUKKAMALA, S., A. H. SUNG und A. ABRAHAM: *Intrusion detection using ensemble of intelligent paradigms*. Journal of Network and Computer Applications, 28:167–182, 2005.

[37] PATCHA, A. und J.-M. PARK: *An overview of anomaly detection techniques: Existing solutions and latest technological trends*. In: *Proceedings of Computer Networks: The International Journal of Computer and Telecommunications Networking*, 2007.

[38] PORTNOY, L., E. ESKIN und S. STOLFO: *Intrusion detection with unlabeled data using clustering*. In: *Proceedings of ACM Workshop on Data Mining Applied to Security (DMSA) 2001.*, 2001.

[39] RHODES, B. C., J. A. MAHAFFEY und J. D. CANNADY: *Multiple Self-Organizing Maps for Intrusion Detection*. In: *23rd National Information Systems Security Conference - PROCEEDINGS, PAPERS, and SLIDE PRESENTATIONS*, 2000. `http://csrc.nist.gov/nissc/2000/proceedings/2000proceedings.html` (2007-04-19).

[40] SCHAPIRE, R. E.: *The boosting approach to machine learning: An overview*. In: DENISON, D. D., M. H. HANSEN, C. HOLMES, B. MALLICK und B. YU (Hrsg.): *Nonlinear Estimation and Classification*. Springer, 2003.

[41] SCHMIDT, A.-D., F. PETERS, F. LAMOUR und S. ALBAYRAK: *Monitoring Smartphones for Anomaly Detection*. In: *Mobilware 2008*, 2008. to appear.

[42] SCHMIDT, S., T. ALPCAN, S. ALBAYRAK und A. MÜLLER: *A Monitor Placement Game for Intrusion Detection*. In: *Proc. of CRITIS, 2nd International Workshop on Critical Information Infrastructures Security*, Lecture Notes in Computer Science. Springer, 2007. to appear.

[43] STAKHANOVA, N., S. BASU und J. WONG: *A taxonomy of intrusion response systems*. International Journal of Information and Computer Security, 1(1/2):169–184, 2007.

[44] VLACHOS, V., S. ANDROUTSELLIS-THEOTOKIS und D. SPINELLIS: *Security applications of peer-to-peer networks*. Computer Networks, 45(2):195–205, Juni 2004.

[45] WEI, S., J. MIRKOVIC und M. SWANY: *Distributed Worm Simulation with a Realistic Internet Model*. PADS, 00:71–79, 2005.

[46] XIONG, L. und L. LIU: *PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities*. IEEE Transactions on Knowledge and Data Engineering, 16(7):843–857, 2004.

[47] YANG, D., C. HU und Y. CHEN: *A framework of cooperating intrusion detection based on clustering analysis and expert system*. In: *InfoSecu '04: Proceedings of the 3rd International Conference on Information Security*, S. 150–154, New York, NY, USA, 2004. ACM Press.

[48] YEGNESWARAN, V., P. BARFORD und S. JHA: *Global Intrusion Detection in the DOMINO Overlay System.*. In: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2004*, 2004.

[49] YUN, J. B., E. K. PARK, E. G. IM und H. P. IN: *A Scalable, Ordered Scenario-Based Network Security Simulator*. In: *Systems Modeling and Simulation: Theory and Applications*, Bd. 3389/2005 d. Reihe *Lecture Notes in Computer Science (LNCS)*, S. 487–494. Springer-Verlag, 2005.

[50] ZHOU, C. V., S. KARUNASEKERA und C. LECKIE: *A Peer-to-Peer Collaborative Intrusion Detection System*. In: *Proceedings of the IEEE International Conference on Networks (ICON 2005)*, S. 118–123, November 2005.