Aufbau eines ML-basierten Intrusion Detection Systems (IDS)

Fachmodul
Eastern Switzerland University of Applied Sciences

Moritz Bättig & Christoph Landolt

Verfasser:	Moritz Bättig	moritz.baettig@ost.ch
	Christoph Landolt	christoph.landolt@ost.ch
Referent:	René Pawlitzek	rene.pawlitzek@ost.ch
Korreferent:	Klaus Frick	klaus.frick@ost.ch
Industriepartner:	Noser Engineering	
Institute:	INF Institut für Ingenieurinformatik	
	ICE Institut für Compuational Engineering	
Datum:	07.01.2022	

Inhaltsverzeichnis

1 Einleitung **4**

2 Intrusion Detection Systems (IDS) Grundlagen **5**

2.1 Standardisierung und IDS-Komponenten 5

2.1.1 Ereigniskomponenten 6

2.1.2 Analysekomponenten 6

2.1.3 Datenbankkomponenten 9

2.1.4 Reaktionskomponenten 9

2.2 IDS-Kategorien 10

2.2.1 Hostbasierte IDS (HIDS) 10

2.2.2 Netzwerkbasierte IDS (NIDS) 11

2.3 Deep packet inspection (DPI) 12

2.3.1 Deep packet inspection and Encryption 12

2.4 Visualisierung von Intrusion Detection Events 13

2.4.1 Output Stream 13

2.4.2 Tabelle 14

2.4.3 Dashboard 14

2.4.4 Security Event Visualization 15

3 Marktanalyse IDS **18**

3.1 Snort 18

3.1.1 Snort-Architektur 18

3.2 SolarWinds Security Event Manager (SEM) 20

3.3 Cisco Secure IPS (NGIPS) 20

3.4 McAfee Network Security Platform 20

3.5 Trend Micro TippingPoint Threat Protection System (TPS) 20

3.6 Amazon GuardDuty 21

3.7 IBM Security Network IPS 21

3.8 Zusammenfassung der Markttrends 22

4 Netzwerksicherheitsarchitekturen **23**

4.1 Klassischer Einsatz eines IDS 23

4.1.1 Schutz von Netzwerkzonen 23

4.1.2 Schutz von ausgewählten Systemen 24

4.1.3 Kombination von netzwerk- und hostbasiertem IDS 25

4.2 Schutz zum Internet 25

4.2.1 Webapplication Firewall (WAF) 26

4.2.2 Proxy-Server 26
4.2.3 Mail-Relay-System 27
4.3 Schutz der Clients 27
4.3.1 802.1X port-based Network Access Control (PNAC) 28
4.4 Schlussfolgerungen Netzwerksicherheitsarchitekturen 29

5 Machine Learning Grundlagen 29
5.1 Supervised Learning 30
5.1.1 Klassifizierung 31
5.1.2 Regression 31
5.1.3 Einsatz von Supervised Learning zur Angriffserkennung 32
5.2 Unsupervised Learning 33
5.2.1 Clustering 33
5.2.2 Dimensionsreduktion 34
5.2.3 Einsatz von Unsupervised Learning zur Angriffserkennung . . . 35
5.3 Reinforcement Learning 36
5.3.1 Einsatz von Reinforcement Learning zur Angriffserkennung . . . 36
5.4 Neuronale Netzwerke 37
5.4.1 Modell eines Neurons 38
5.4.2 Aktivierungsfunktionen 40
5.5 Learning from Data Streams 41
5.6 Data Types 43
5.6.1 Datensätze 43
5.6.2 Merkmalsauswahl 45

6 Schlussfolgerungen und Ausblick 46

1 Einleitung

Durch die steigende Vernetzung in der Industrie steigt auch das Risiko durch Cyberangriffe. In der Industrie 4.0 können die beiden Trends zur horizontalen Vernetzung entlang der Lieferketten und zur vertikalen Vernetzung, sprich der Vernetzung von Produktion und IoT-Devices mit den Business-Prozessen und ERP-Systemen des Unternehmens, identifiziert werden.

Damit ein Unternehmen langfristig erfolgreich am Markt bestehen kann, ist es wichtig die Cyber-Risiken zu minimieren. Zu diesem Zweck werden neben herkömmlichen Sicherheitssystemen wie Firewalls und Antiviren-Programmen auch Intrusion Detection Systeme (IDS) eingesetzt, um mögliche Angriffe frühzeitig zu erkennen und diesen entgegenzuwirken. Aufgrund der Diversität und der stetigen Veränderung der modernen IT-Umgebungen ist es sehr schwer mittels statischen Signaturen die IT-Umgebung und damit auch die Produktionsressourcen in der modernen Fabrik effektiv zu schützen.

Diese Arbeit beschäftigt sich mit der Frage, wie Cyber-Angriffe auf IT-Systeme effektiv erkannt werden können und wie durch stetige Analyse der Netzwerkverbindungen dynamische Modelle zur Angriffserkennung geschaffen werden können. Weiter soll aufgezeigt werden wie der aktuelle Stand der Forschung im Bereich Einsatz von maschinellem Lernen in Intrusion Detection Systemen ist.

Zur Beantwortung der Frage dient die Funktionsweise eines modernen IDS, welches auf Basis wissenschaftlicher Publikationen und wissenschaftlicher Werke aus den Fächern IT-Sicherheit, Verteilte Systeme, theoretische Informatik, Statistik und Data Analytics beschrieben wurde.

Im Bericht wird als Einführung die Funktionsweise moderner IDS beschrieben und die wichtigsten Funktionen für eine effektive Angriffserkennung wie Deep Package Inspection aber auch die Visualisierung von sicherheitsrelevanten Events. Anschliessend werden die wichtigsten Produkte am Markt im Bereich IDS und deren Funktionsweise beschrieben. In einem weiteren Schritt wird aufgezeigt wie moderne Computer-Netzwerke heute bereits geschützt werden und welche Technologien zum Einsatz kommen. Zum Schluss folgt eine Zusammenfassung der wichtigsten Verfahren und Datensätze aus dem Bereich Machine Learning für die Analyse sicherheitsrelevanter Events.

Dieser Bericht wurde im Rahmen des Fachmoduls als Vorbereitung auf die Bachelorarbeit am Institut für Ingenieurinformatik INF (INF) und Institut für Computational Engineering (ICE) der Ostschweizer Fachhochschule OST durchgeführt und dient als Grundlage für die Weiterbearbeitung des Themas als Forschungsprojekt.

2 Intrusion Detection Systems (IDS) Grundlagen

Unter einem Intrusion Detection System (IDS) versteht man ein System zum Schutz von IT-Systemen, welches die frühzeitige Erkennung von Attacken auf Computernetzwerke ermöglicht. Darüber hinaus ermöglicht ein Intrusion Detection System (IDS) das Sammeln von Informationen über neue Angriffsarten, welche zur präventiven Verbesserung des Schutzes des Computernetzwerkes verwendet werden können [32].
Häufig werden Intrusion Detection Systems (IDS) nach Art der Audit-Daten in hostbasierte IDS und netzwerkbasierte IDS so wie nach Analysetechnik in Anomalieerkennung (Anomaly-based) und Signaturanalyse (Signature-based) eingeteilt [32].
Moderne IDS verfügen neben den Sensorkomponenten häufig auch über reaktive Komponenten. Diese erlauben unmittelbar nach oder während der Erkennung eines Angriffes auf diesen zu reagieren. Diese Systeme werden häufig als reaktive IDS oder als Intrusion Prevention Systems (IPS) bezeichnet [32]. Da der Begriff IPS häufig aus marketingtechnischen Gründen nicht nur für reaktive IDS sondern auch für andere Netzwerkkomponenten wie Firewalls verwendet wird [32], ist bei der weiteren Verwendung des Begriffs IPS immer ein reaktives IDS gemeint.

2.1 Standardisierung und IDS-Komponenten

Die Defense Advanced Research Projects Agency (DARPA) standardisierte die IDS-Architektur durch das Common Intrusion Detection Framework (CIDF), um die Wiederverwendung von IDS-Komponenten und die Kooperation zwischen verschiedenen IDS zu ermöglichen. Das CIDF sieht Ereigniskomponenten, Analysekomponenten, Datenbankkomponenten und Reaktionskomponenten als die vier möglichen IDS-Komponenten vor [32].

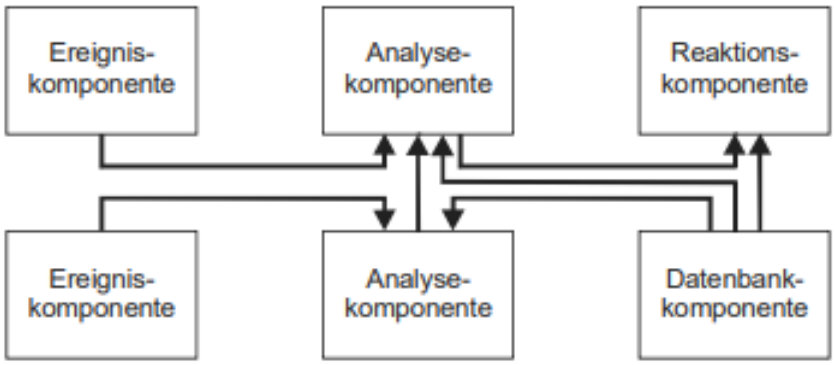


Abbildung 1: Interaktion der IDS-Komponenten [32]

2.1.1 Ereigniskomponenten

Um eine automatische Erkennung von Sicherheitsverletzungen zu ermöglichen, braucht es ein Verfahren zur Protokollierung von sicherheitsrelevanten Abläufen oder Zuständen von IT-Systemen. Von diesem Verfahren, auch Audit genannt, können zwei Arten unterschieden werden: zustandsbasiertes Audit und transitions- bzw. aktionsbasiertes Audit. Beim zustandsbasierten Audit wird der Zustand eines IT-Systems periodisch zur späteren Analyse aufgezeichnet. Da dies zu grossen und schwer handhabbaren Datenmengen führt, werden in der Praxis nur sicherheitsrelevante Zustände aufgezeichnet. Bei einem transitions- bzw. aktionsbasiertem Audit werden sicherheitsrelevante Aktivitäten aufgezeichnet, um durch spätere Analyse entscheiden zu können, ob ein sicherheitskritischer Systemzustand erreicht wurde oder nicht [32].

2.1.2 Analysekomponenten

Diese Komponenten führen die eigentliche Sicherheitsverletzungserkennung im IDS durch. Mittels entsprechender Verfahren werden die Audit-Daten analysiert, um den Zustand des Systems als sicherheitskonform oder sicherheitsverletzend zu klassifizieren. Folgende vier Werte werden typischerweise zur Bewertung herangezogen [32]:

- True Positive: Das System wird korrekt als sicherheitsverletzend klassifiziert.
- True Negative: Das System wird korrekt als sicherheitskonform klassifiziert.
- False Positive: Das System wird fälschlicherweise als sicherheitsverletzend klassifiziert.
- False Negative: Das System wird fälschlicherweise als sicherheitskonform klassifiziert.

Es existieren zwei grundsätzliche Analysetechniken zur Einbruchserkennung:

- Anomalie- oder Verhaltenserkennung
- Signatur- oder Regelanalyse

Anomalie- oder Verhaltenserkennung

Bei der Anomalieerkennung werden Sicherheitsverletzungen aufgrund von Abweichungen von Normen identifiziert. Es wird somit zwischen normalen und anomalen Abläufen und Zuständen unterschieden. Dazu muss angenommen werden, dass ein normales Verhalten existiert und mess- bzw. beschreibbar ist. Zur Erzeugung der erforderlichen Referenzinformationen geschieht dies durch das Erlernen bzw. Messen oder durch das

Spezifizieren des normalen Verhaltens [32].
Bei der ersten Methode ist eine Lernphase des Systems erforderlich, die regelmässig oder kontinuierlich wiederholt wird. Problematisch ist, dass Sicherheitsverletzungen während der Lernphase durch andere Mechanismen erkennt und verhindert werden müssen, da diese sonst in die Referenzprofile normalen Verhaltens einfliessen.
Bei der spezifikationsbasierten Anomalieerkennung wird das normale Verhalten explizit und formal spezifiziert. Die verwendeten Spezifikationen repräsentieren eine Grammatik und ein entsprechender Parser überprüft die Konformität beobachteter Aktionen und Zustände zu den Spezifikationen [32].

Signatur- oder Regelanalyse
Bei der Signaturanalyse wird eine umfangreiche Datenbank mit Angriffssignaturen verwaltet. Eine Signatur oder eine Regel kann eine Liste von Merkmalen eines einzelnen Pakets oder verwendeten Protokolls sein, zum Beispiel das Tupel Quell- Zielport oder eine bestimmte Bitreihenfolge. Die Analysekomponente des IDS durchsucht jedes durchlaufende Paket und vergleicht dessen Merkmale mit der Signaturdatenbank, sodass sie im Falle einer Übereinstimmung einen Alarm auslösen kann.
Dieser Ansatz ist am weitesten verbreitet aber hat einige Nachteile. Es benötigt Vorwissen über einen Angriff, um eine genaue Signatur zu erstellen. Deshalb ist das IDS völlig blind für neue Angriffe, die noch nicht von ihm aufgezeichnet wurden. Andererseits kann es sich bei einer Signaturübereinstimmung schnell um einen Fehlalarm handeln, da nur gezielte Eigenschaften abgeglichen werden und weitere Informationen eventuell nicht verarbeitet werden. Letztlich muss jedes Paket mit einer möglicherweise riesigen Signaturdatenbank abgeglichen werden, was das IDS mit der Verarbeitung überfordern kann. Wenn die Paketanalyse mit dem Netzwerkdurchsatz nicht mithalten kann, könnten bösartige Pakete übersehen werden [36].

Es existieren verschiedenste Signaturdatenbanken, die von IDS-Herstellern bereitgestellt und gepflegt werden. Das open-source IDS Snort zum Beispiel stellt ein Community Ruleset und ein Subscriber Ruleset zur Verfügung. Ersteres ist für alle Nutzer frei erhältlich und wird von den Nutzern gepflegt und aktuell gehalten. Das Subscriber Ruleset wird von Cisco Talos entwickelt, getestet und gepflegt. Als Abonnent erhält man die Signaturen in Echtzeit.

Vergleich der Analyseansätze
Ein grosser Unterschied der beiden Ansätze ist, dass bei der lernbasierten Anomalieerkennung keine explizite Festlegung hinsichtlich sicherheitskonformer oder -verletzender Aktionen und Zustände getroffen werden müssen. Somit ist es prinzipiell möglich, neue

bzw. unbekannte Sicherheitsverletzungen zu erkennen. Bei der regelbasierten Anomalieerkennung ist das nicht der Fall, da die Merkmale des Angriffs bereits in der Signaturdatenbank vorhanden sein müssen.
Andererseits ist es schwierig, verhaltensspezifische Merkmale zu finden und die Veränderbarkeit des Nutzerverhaltens über längere Zeiträume einzuschätzen. Diese signifikante inhärente Unschärfe führt bei der Anomalieerkennung oft zu unakzeptablen Fehlalarmen.
Ein weiteres Problem der Verhaltenserkennung besteht in der Notwendigkeit, die Sicherheitskonformität des Systems während der Lernphase durch andere Mechanismen sicherzustellen. Ausserdem basiert der Ansatz lediglich auf der Hypothese, dass sich Sicherheitsverletzungen in anomalem Verhalten manifestieren. Deshalb zeigt sich in der Praxis bei diesen Systemen eine nicht vernachlässigbare Falsch-Negativ-Rate.
Ein Hauptvorteil der signaturbasierten Analyse liegt in den scharfen und nachvollziehbaren Ergebnissen, durch die konkret aufgetretene Sicherheitsverletzungen beschrieben werden können. Das Leistungsvermögen von signaturbasierten Systemen steht und fällt jedoch mit dem Umfang der Qualität und der Aktualität der verwendeten Merkmalsbeschreibungen. Wegen der Komplexität der Signaturentwicklung und den daraus resultierenden Entwicklungszeiten können Signaturen erst einige Zeit nach dem Auftauchen entsprechender Sicherheitslücken zum Einsatz kommen. Deshalb besteht hier ein permanentes Unvollständigkeitsproblem und auch dieser Ansatz weist in der Praxis Falsch-Negative auf [32].

Da in der Praxis jeder der Analyseansätze jeweils nur für das Finden bestimmter Teilmengen von Sicherheitsverletzungen geeignet ist, wird zunehmend eine Kombination der Ansätze angewandt. Durch diese Methode kann eine Verringerung der Fehlalarme erreicht werden [32].

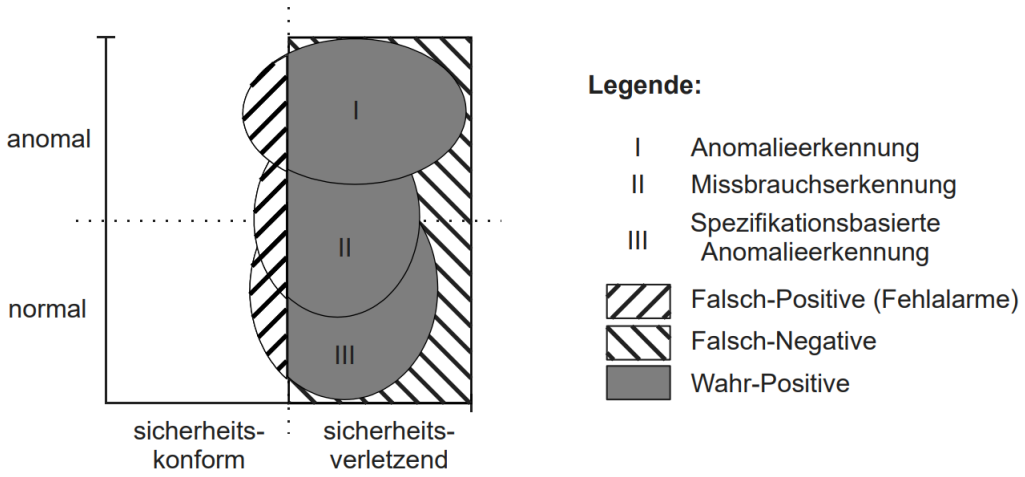


Abbildung 2: Vergleich der Analyseansätze [32]

2.1.3 Datenbankkomponenten

Bei der Angriffserkennung und -analyse entstehen Ereignisdaten, die zur späteren Weiterverwendung abgespeichert werden müssen. Da in kurzer Zeit sehr grosse Datenmengen anfallen können, ist es notwendig, dafür ein Datenbanksystem zu verwenden, um einerseits die anfallende Datenmenge aufzunehmen und zuverlässig zu verwalten sowie andererseits Zugriffszeiten zu minimieren [10].

2.1.4 Reaktionskomponenten

Nachdem Sicherheitsverletzungen durch die Analysekomponenten erkannt wurden, werden die Reaktionskomponenten des IDS veranlasst, entsprechende Reaktionen durchzuführen. Man unterscheidet zwischen passiven und aktiven Reaktionen:

- Passive Reaktion: Informationen über den Angriff werden an den Benutzer des IDS weitergeleitet und die Ergreifung weiterer Massnahmen wird ihm überlassen.
- Aktive Reaktion: Das IDS löst automatisch oder halbautomatisch eine Aktion aus. Möglich sind hierbei z.B. das Blockieren bestimmter Netzwerdienste, das Benachrichtigen umgebender Systeme und die Sammlung zusätzlicher Informationen.

2.2 IDS-Kategorien

Basierend auf der Position in einem System, fällt ein IDS in eine der beiden Kategorien:

- Hostbasiert
- Netzwerkbasiert

Je nach Anwendung können beide IDS-Kategorien Vorteile und Nachteile haben.

2.2.1 Hostbasierte IDS (HIDS)

Hostbasierte Intrusion Detection Systeme überwachen den Sicherheitszustand bestimmter einzelner Systeme oder Hosts. Diese hostbasierten Agenten, welche oft auch Sensoren genannt werden, werden somit typischerweise auf Geräten installiert, die als anfällig für potenzielle Angriffe gelten. Für jedes Gerät wird ein separates HIDS benötigt, da sich dieses nur auf die Überwachung eines einzelnen Hosts beschränkt. Ein HIDS sammelt Daten über bestimmte Ereignisse auf dem zu überwachenden Host, die häufig vom Betriebssystem in Form eines Audit Trails zur Verfügung gestellt werden. Dieser Trail enthält Informationen über die Objekte, die ein Event ausgelöst haben oder anderweitig mit dem Event in Verbindung stehen. Aufgrund dieser Daten kann festgestellt werden, welcher Prozess, welches Programm oder welcher Benutzer die Sicherheitsverletzung ausgelöst hat [40].
Ein häufiger Kritikpunkt von hostbasierten Intrusion Detection Systemen hängt mit der Menge der Daten zusammen, die das System liefern kann. Je detaillierter das IDS Informationen über potenzielle Sicherheitsverletzungen sammeln soll, desto stärker häufen sich grosse Datenmengen an, die sehr viel Speicher belegen. Grundsätzlich wird es für einen Angreifer schwieriger, den Prüfungsprozess zu umgehen, je grösser die gesammelte Datenmenge ist. Jedoch macht es die Komplexität und Grösse der Daten in der Praxis einfacher, für Eindringlinge, ihre Spuren zu verbergen. Diese Ironie ist oft eine Hürde, die ein Entwickler von hostbasierten IDS bewältigen muss. Ein weiterer Nachteil ist, dass ein HIDS den Netzwerkverkehr nicht sehen kann, da diese Art von Intrusion Detection Systeme nur für lokale Hosts entwickelt wurden. Ausserdem ist ein HIDS sehr stark vom Betriebssystem abhängig. Falls ein Angreifer ein Schlupfloch im Betriebssystem findet, könnte dieser auch direkt das HIDS umgehen [40].
Hostbasierte IDS haben jedoch auch einige Vorteile. Wie bereits erwähnt kann oft angegeben werden, wer auf was zugegriffen oder wer was ausgelöst hat. Somit kann in vielen Fällen die Person oder der Prozess identifiziert werden, der die Sicherheitsverletzung verursacht hat. HIDS können auch sehr gut das Verhalten eines bestimmten Benutzers beobachten. Somit kann man Attacken während des Angriffs oder sogar vor dem Systemzugriff stoppen [40].

2.2.2 Netzwerkbasierte IDS (NIDS)

Netzwerkbasierte Intrusion Detection Systeme sammeln ihre Informationen vom Netzwerk selbst anstatt von jedem einzelnen Host. Die Ereigniskomponente arbeitet auf Grundlage eines Abhörkonzepts, d. h. es werden Informationen aus dem Netzwerkverkehrsstrom gesammelt, während sich die Daten in einem bestimmten Netzwerksegment bewegen. Das IDS prüft auf Angriffe oder auf unregelmässiges Verhalten, indem es die Inhalte und Header-Informationen aller Pakete, die sich durch das Netzwerk bewegen, untersucht [40].

Die Verwendung von Netzwerkdaten als primäre Informationsquelle ist in mehrfacher Hinsicht wünschenswert. Zum einen beeinträchtigt die Ausführung der Netzwerküberwachung nicht die Leistung anderer Programme, da die einzelnen Pakete nur dann ausgelesen werden, wenn sie sich durch das überwachte Netzwerksegment bewegen und keine Software auf einem Host ausgeführt werden muss. Des Weiteren ist die Netzwerküberwachung für den Systembenutzer transparent, was auch dafür sorgt, dass ein potenzieller Angreifer das IDS nicht lokalisieren und deaktivieren kann. Ausserdem sind netzwerkbasierte IDS äusserst mobil. Sie überwachen nur den Verkehr über ein bestimmtes Netzwerksegment und sind unabhängig von den Betriebssystemen, auf denen sie installiert sind [40].

Netzwerkbasierte Intrusion Detection Systeme haben jedoch auch ihre Tücken. Ein grosses Problem ist die Skalierbarkeit. Netzwerküberwachungssysteme müssen jedes Paket untersuchen, das sich durch das bestimmte Segment bewegt und es hat sich gezeigt, dass ein IDS ab einem Netzwerkdurchsatz von 100 Mbps überfordert sein kann. In den heutigen weitverbreiteten Gigabit-Netzwerken könnten potenzielle Angreifer diese Sicherheitslücke ausnutzen. Die strategische Platzierung von Netzwerksensoren kann hier Abhilfe schaffen, aber Systeme mit hohem Datenverkehr werden trotzdem immer mit diesem Problem konfrontiert sein.
Verschlüsselung und Switching sind zwei weitere Einschränkungen der netzwerkbasierten IDS. Die Ereigniskomponente kann bei einer verschlüsselten Kommunikation nicht auf den Inhalt oder auf das Übertragungsprotokoll des Pakets zugreifen, sondern nur auf den Header. Falls sich im zu überwachenden Netzwerk Switches befinden, werden die Verbindungen zwischen zwei bestimmten Hosts isoliert, sodass ein Host nur den an ihn adressierten Datenverkehr sehen kann. In diesen Fällen ist ein netzwerkbasierter Monitor auf die Überwachung eines einzelnen Hosts reduziert, wodurch ein grosser Teil der Absicht des NIDS zunichtegemacht wird [40].

2.3 Deep Packet Inspection (DPI)

Deep Packet Inspection (DPI) beschreibt eine Methode, bei der das gesamte Datenpaket geprüft wird, sobald dieses einen bestimmten Checkpoint im Netzwerk traversiert [12]. Dieses Verfahren ermöglicht ein effektives Aufspüren von Malware, Spam oder weiteren Angriffen auf Computersysteme.
Die Idee der Deep Package Inspection (DPI) wird in modernen Netzwerken besonders häufig bei Schnittstellen zum Internet verwendet. Beispielsweise bei Proxy-Servern, welche den Datenverkehr der User im Internet analysiert, Spam Filtern auf E-Mail Relay-Servern oder auch Web Application Firewalls (WAF), welche den Schutz der firmeneigenen Webserver und Webservices vor Angriffen aus dem Internet sicherstellt. Es gibt aber zwei Nachteile bei der Deep Package Inspection (IDP):

1. Verschlüsselter Datenverkehr kann nicht analysiert werden. Daher muss der ganze Datenverkehr entschlüsselt werden. Dazu muss zum einen der Private-Key der jeweiligen Kommunikation bekannt sein, zum andern führt diese Entschlüsselung des Datenverkehrs zu Performance-Einbussen [31]. Daher ist die Deep Package Inspection oft nur auf Schnittstellen mit dem Internet begrenzt und wird nicht im ganzen Netzwerk eingesetzt.

2. Das Entschlüsseln des Datenverkehrs kann auch als Eingriff in die Privatsphäre der Benutzer aufgefasst werden [31]. Daher muss jede Organisation prüfen, ob diese Methode überhaupt zulässig ist und bei Bedarf ihre internen Reglemente entsprechend anpassen.

2.3.1 Deep Packet Inspection and Encryption

Durch den Einsatz von IPSec oder anderen Verschlüsselungstechniken wird der Einsatz von reinen Network Intrusion Detection Systems (NIDS) nahezu nutzlos [43]. Durch das steigende Bedürfnis, Vertraulichkeit und Integrität von Netzwerken sicherzustellen, geht der Trend jedoch eindeutig in Richtung von vollständig verschlüsselter Netzwerkkommunikation. Für die Deep Packet Inspection in solchen Netzwerken bietet sich lediglich eine Kombination aus Netzwerk- und Hostbasierten Intrusion Detection Systemen an, welche am Rand des Netzwerkes die Analyse der jeweiligen Pakete ermöglicht [43].
Ein anderer Ansatz verfolgt Cisco mit ihrer Encrypted Traffic Analytics. Dabei wird beim Aufbau einer verschlüsselten Verbindung das erste Datenpaket, das sogenannte Initial Data Package (IDP) abgefangen und daraus die Features für die Machine Learning Algorithmen extrahiert [8]. Die darauf folgenden verschlüsselten Datenpakete werden anhand dieser extrahierten Informationen auf ihre Integrität überprüft [8].

Trotz all dieser Techniken kann ein nahezu vollständiger Schutz von verschlüsselten Netzwerken lediglich durch Entschlüsselung des Datenverkehrs erreicht werden [55].

2.4 Visualisierung von Intrusion Detection Events

Ein Event ist eine beobachtbare Situation oder eine Veränderung in einer Umgebung, die über einen bestimmten Zeitraum hinweg auftritt. Ein Event kann ein bestimmter Zustand oder eine Zustandsänderung eines Systems sein [29].
Ein Event kann beschrieben und aufgezeichnet werden. Eine einzelne Aufzeichnung wird oft auch als Logeintrag bezeichnet. Ein Logeintrag ist ein einzelner Datensatz, der Details aus einem oder mehreren Events enthält. Ein Logeintrag wird manchmal auch als Ereignisprotokoll, Eventprotokoll, Alarm, Protokolleintrag oder Audit-Event genannt [29].
Eine Sammlung von mehreren Logeinträgen bildenen einen Log. Ein Log wird typischerweise in einer lokalen Datei, in einer Datenbank oder auf einem Netzwerkserver gespeichert. Ein Log wird auch Audit Trail genannt.
Bei der Verwendung von Intrusion Detection Systemen fallen riesige Datenmengen an. Die Events, die das System als Output liefert, müssen so dargestellt werden, dass der Nutzer nicht von der Datenflut überwältigt wird und diese schnell interpretieren kann. Im folgenden werden verschiedene Ansätze vorgestellt.

2.4.1 Output Stream

Die einfachste Art der Visualisierung von Intrusion Detection Events ist die Ausgabe des Output Streams auf einer textbasierten Konsole. Der Nachteil dieser Darstellung ist die Unübersichtlichkeit. Der Nutzer kann schnell den Überblick über die Daten verlieren.

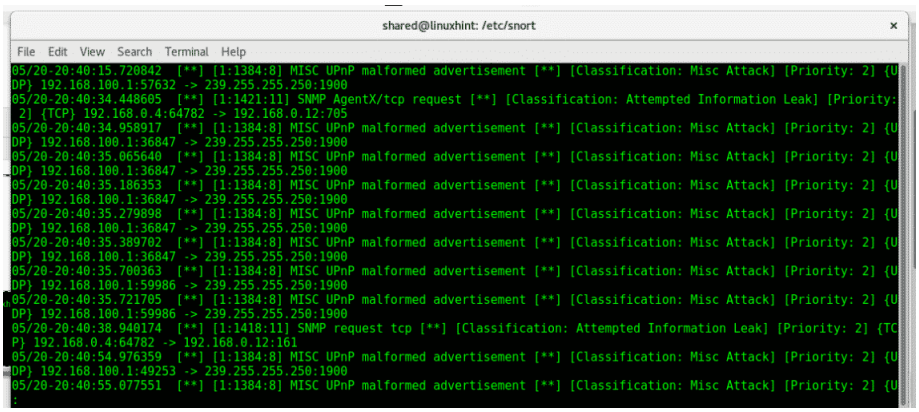


Abbildung 3: Snort Output Stream [3]

2.4.2 Tabelle

Anstatt die Events einfach textbasiert in einer Konsole auszugeben, können diese in einer Tabelle aufgelistet werden. Die Daten werden somit strukturierter und übersichtlicher dargestellt. Die grössten Vorteile einer Tabelle sind die Sortierung, Filterung und Gruppierung der Daten. Eine Gruppierung nach der Quell-IP-Adresse ist zum Beispiel nützlich, um zeitlich verzögerte Attacken desselben Ursprungs aufzulisten. Andererseits kann zum Beispiel auch eine Filterung nach bestimmten Signaturen nützlich sein.

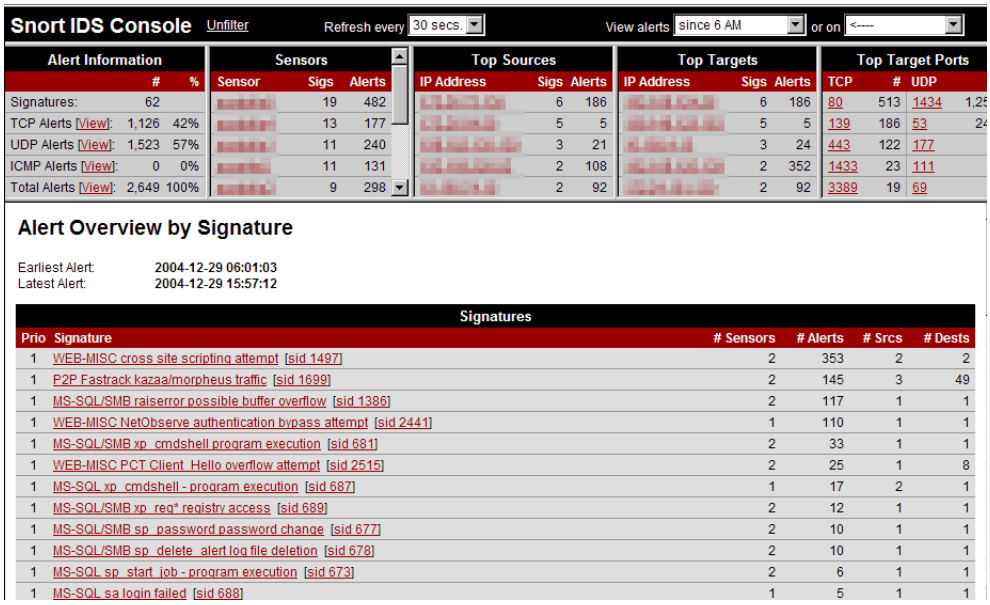


Abbildung 4: Snort IDS Console [1]

2.4.3 Dashboard

Eine fortgeschrittene und individualisierbare Art Intrusion Detection Events darzustellen kann mit Hilfe eines Dashboards erzielt werden. In einem Dashboard können verschiedene statistische Tools und Grafiken angezeigt werden, um es dem Benutzer so leicht wie möglich zu machen, den Zustand des IDS und des zu überwachenden Systems zu beobachten. Folgende unvollständige Liste zeigt auf, was für Elemente ein Dashboard enthalten könnte.

1. Geografische Karte: Mithilfe dieses Elements können Intrusion Detection Events in Bezug auf einen bestimmten Ort dargestellt werden. Eine naheliegende Anwendung wäre die Kennzeichnung der Ursprungs-IP-Adresse des Angriffs auf der Karte.

2. Häufigkeitsdiagramme: Oft ist es sinnvoll, die Häufigkeit bestimmter Eigenschaften der Intrusion Detection Events darzustellen. Anwendungen dafür könnten

Kuchendiagramme oder Histogramme sein, die Signaturarten, fehlgeschlagene Logins oder andere Merkmale übersichtlich darstellen.

3. Fortgeschrittene Suchfunktion: Oft muss nach Merkmalen von Attacken gesucht oder gefiltert werden. Durch eine Suchfunktion kann dies besonders benutzerfreundlich und vorallem schnell gestaltet werden. Sinnvoll wäre es auch, bestimmte, häufig verwendete Suchabfragen abzuspeichern damit diese jederzeit wiederverwendet werden können.

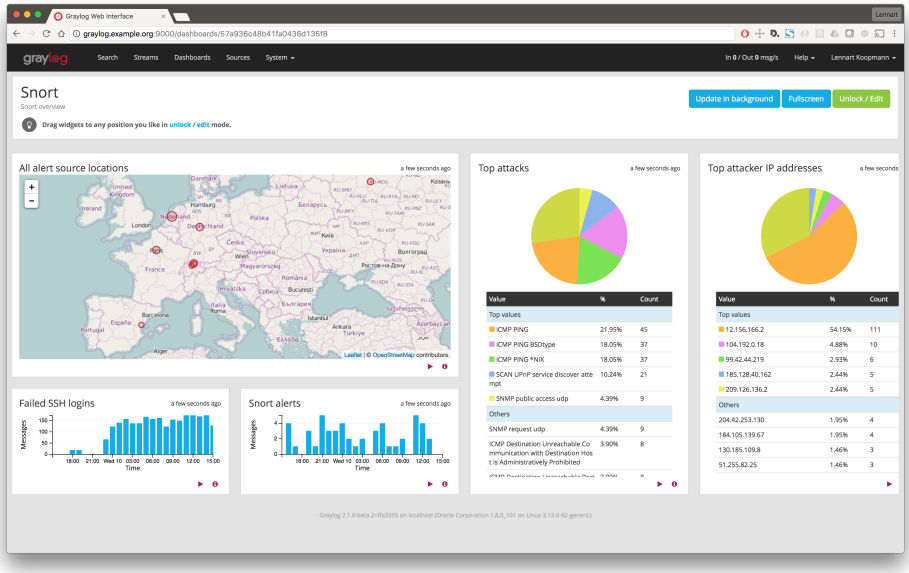


Abbildung 5: Graylog Dashboard [16]

2.4.4 Security Event Visualization

Bei anomaliebasierten Intrusion Detection Systemen werden Sicherheitsverletzungen und Angriffe durch automatische Analysen mit Klassifizierern und Algorithmen detektiert. Im folgenden werden zwei Visualisierungen erläutert, die im Zusammenhang mit Machine Learning Algorithmen zur Anomalieerkennung stehen. Die Übersichtlichkeit und Darstellung für einen Benutzer steht hier nicht mehr im Vordergrund.

Scatter Plot

Ein Scatter Plot wird verwendet um ordinale oder kontinuierliche Daten in zwei oder mehr Dimensionen darzustellen. Er eignet sich hervorragend um Cluster oder Trends in den Daten zu erkennen. Ein Beispiel ist das Auftragen der Ziel-IP-Adresse gegen die Ziel-Portnummer. In der Abbildung 6 ist eine vertikale Linie ersichtlich, die indiziert, dass auf einem Client ein Port Scan durchgeführt wurde [29].

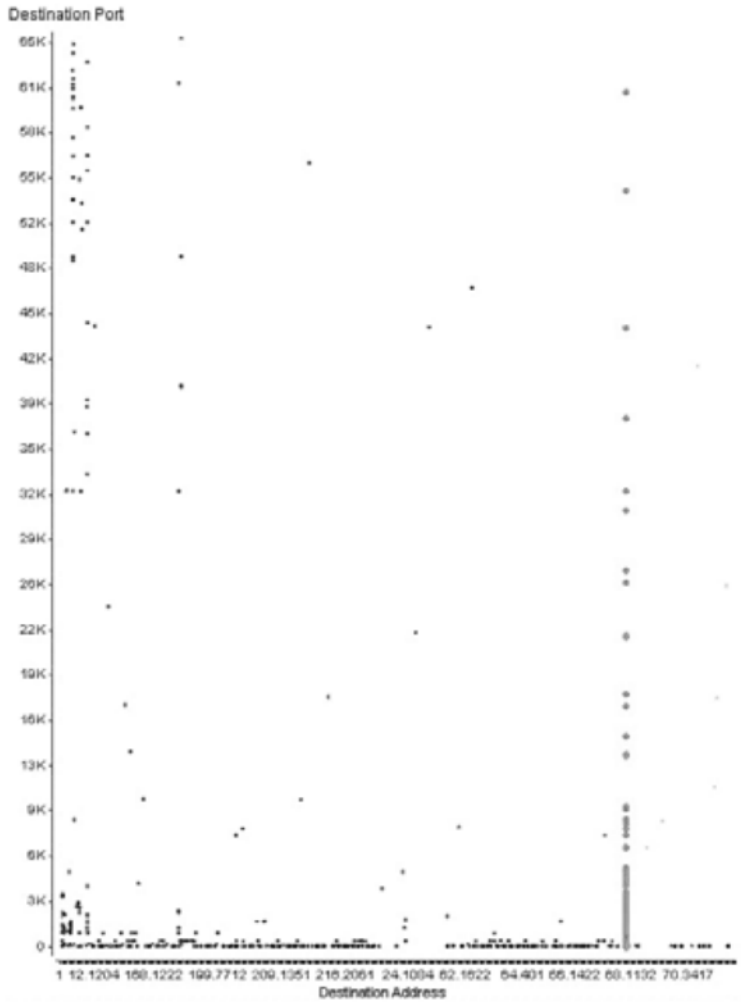


Abbildung 6: Scatter Plot [29]

Scatter Plots werden zur Visualisierung von verschiedensten Clustering-Algorithmen, wie zum Beispiel dem K-Means-Algorithmus, verwendet.

Link Graph

Intrusion Detection Events können auch in einem Link Graph dargestellt werden. In der Abbildung 7 wird ein Beispiel dargestellt, bei dem die Quell-IP-Adresse, die Ziel-IP-Adresse und die Ziel-Portnummer visualisiert werden.

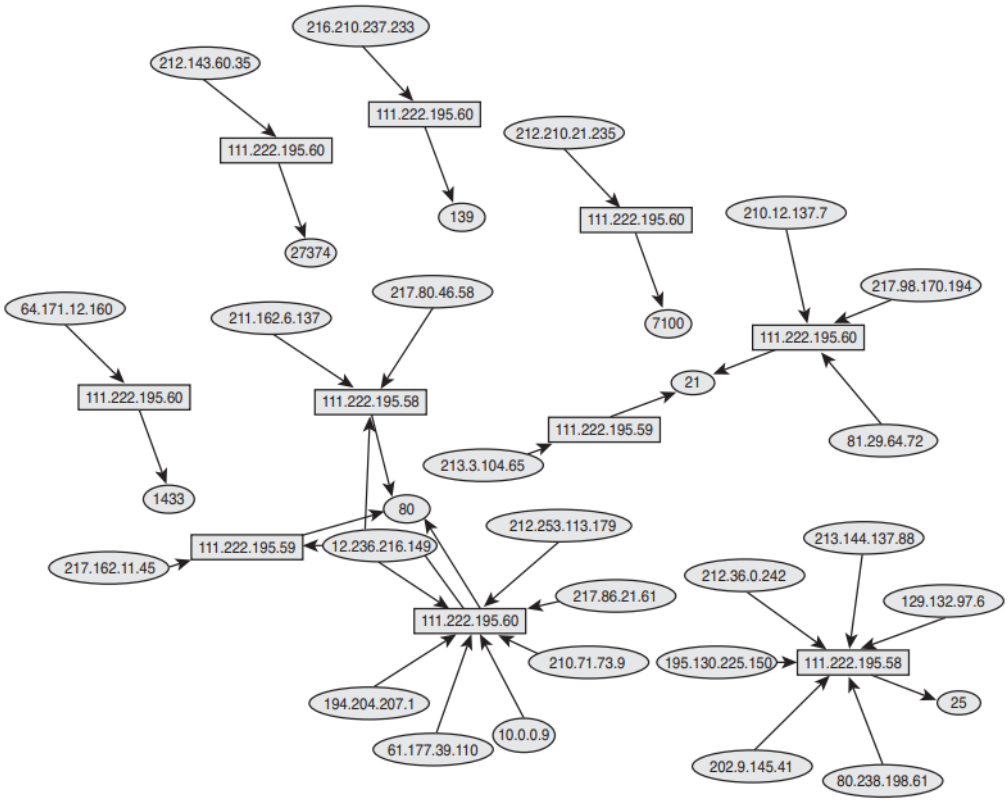


Abbildung 7: Link Graph [29]

Um abnormale von normalen Aktivitäten in einem System zu unterscheiden, kann der Junction Tree Algorithmus angewendet werden, bei dem die Link Graphen als Inputs verwendet werden. Der Algorithmus hat die Form einer Nachrichtenübermittlung auf einen Graphen, der als Junction Tree bezeichnet wird, dessen Nodes Cluster von Variablen sind. Jedes Cluster startet mit einem Potenzial der faktorisierten Dichte. Durch die Kombination dieses Potenzials mit den Potenzialen der Nachbar-Nodes, kann das Cluster den Randwert über seine Variablen berechnen [35].

3 Marktanalyse IDS

In diesem Kapitel werden einige auf dem Markt verbreitete IDS und deren Vor- und Nachteile beschrieben.

3.1 Snort

Snort ist ein open-source netzwerkbasiertes IDS. Durch den open-source Standard geniesst das System ein hohes Vertrauen und dadurch eine grosse Verbreitung unter den IDS.
Snort kann in drei verschiedenen Modi betrieben werden [48]:

1. Sniffer Mode: In diesem Modus werden die Netzwerkpakete lediglich gelesen und auf der Snortkonsole als Stream angezeigt. Der Sniffer Mode erlaubt es Netzwerkadministratoren, ähnlich wie bei Wireshark, nach Fehlern in der Netzwerkkonfiguration zu suchen.

2. Packet Logger Mode: Dieser Modus erlaubt es, den Netzwerkverkehr aufzuzeichnen und als Log abzuspeichern. Dabei wird der Netzwerkverkehr nicht aktiv überwacht. Die Log-Files können aber nachträglich zur Fehleranalyse ausgewertet werden. In einigen Unternehmen kann ein solches System erforderlich sein, um bestimmte Auflagen wie beispielsweise die ISO 27001-Norm zu erfüllen. Die ISO 27001-Norm ist der internationale Standard für Informationssicherheit und schreibt vor, dass alle Zugriffe von Benutzern, Fehlern und Events aufgezeichnet werden.

3. Network Intrusion Detection System (NIDS) Mode: Dabei handelt es sich um die wichtigste Funktionalität von Snort. Dieser Modus erlaubt es den Netzwerktraffic zu detektieren und analysieren.

In der vorliegenden Arbeit wird lediglich der NIDS-Mode von Snort genauer betrachtet.

3.1.1 Snort-Architektur

Bei der Snort-Architektur handelt es sich um eine Plug-in-Architektur wie man sie aus anderen open-source-Projekten wie beispielsweise Eclipse kennt. Dabei wird der Netzwerksniffer als Core-System betrachtet und alle anderen Komponenten kommen lediglich als Plug-in hinzu [50].
Für den Betrieb von Snort als Network Intrusion Detection System werden folgende Komponenten benötigt:

1. Sniffer: Der Paketsniffer zeichnet die Netzwerkpakete (raw data) auf und leitet diese zur weiteren Bearbeitung an den Preprocessor weiter [28].

2. Preprocessor: Bei Snort können zu verschiedenen Protokollen die entsprechenden Preprocessors als Plug-in geladen werden. Ein Preprocessor detektiert um welches Protokoll es sich bei einem bestimmten Paket handelt. So erlaubt beispielsweise das Plug-in HTTP Inspect HTTP-Pakete zu normalisieren und die entsprechenden Daten an die detection engine zur weiteren Prüfung weiterzuleiten. Weiter erlauben die Preprocessors auch eine Priorisierung bestimmter Felder [48].
   Weitverbreitete Preprocessors sind beispielsweise HTTP Inspect für den Schutz von Webservern und Webbrowsern, sfPortscan zum Schutz vor Portscans, wenn Angreifer versuchen, ihr Wissen über das Netzwerk zu verbessern oder der SMTP Preprocessor zum Schutz der Mail-Infrastruktur. Es bleibt aber zu erwähnen, dass Snort keinen verschlüsselten Datenverkehr analysieren kann [48].

3. The detection engine: Hier findet die eigentliche Überprüfung der im Preprocessor normalisierten Daten auf Angriffe statt. Dabei vergleicht Snort die Pakete mit jeder Regel in einem vorgegebenen Ruleset. Schlägt eine Regel auf ein bestimmtes Paket an, wird dieses an den Output weitergeleitet [50].

4. Output: Je nach Art der Regelaktion werden die Events protokolliert oder als Alert an den Systemadministrator gesendet. Zur grafischen Ausgabe kann hierzu die Snort-Konsole verwendet werden [28].

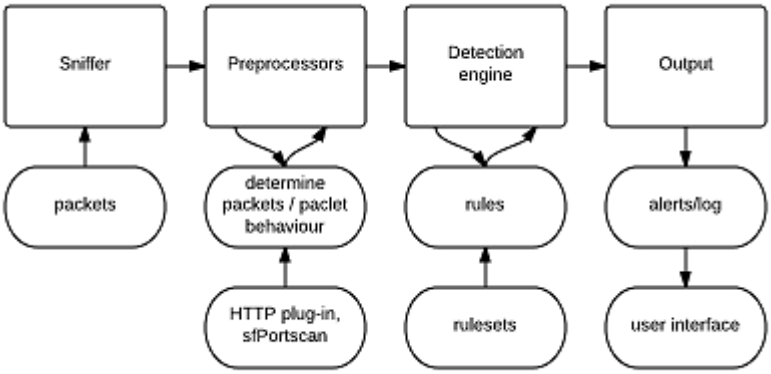


Abbildung 8: Snort Architektur [50]

Die grosse Verbreitung veranlasst durch den open-source-Standard, der hohen Integrität sowie der geringen Kosten kommt Snort in vielen Netzwerkinfrastrukturen zum

Einsatz. Dadurch, dass in heutigen Umgebungen verschlüsselte Verbindungen weitgehend zum Standard gehören, kann in modernen Umgebungen ein Grossteil der Pakete nicht mehr mittels Snort analysiert werden [48].

3.2 SolarWinds Security Event Manager (SEM)

Der SolarWinds Security Event Manager (SEM) ist kein IDS im eigentlichen Sinne, sondern eine Analyse-Umgebung, in der Tools zur Log-Analyse bereitgestellt werden. Der SEM erlaubt daher mittels eines vorgefertigten Rule-Frameworks Angriffe auf Clients, Server oder Netzwerkressourcen anhand client-, server- und netzwerkseitiger Daten festzustellen und Alerts zu generieren. Somit stellt der SEM eher eine Orchestrierung der Thread Detection über die verschiedenen Security-Tools wie beispielsweise Firewall-Logs, Webserver-Logs, Acces-Management-Logs und IDS bereit [42].

3.3 Cisco Secure IPS (NGIPS)

Da Cisco das open-source IDS Snort im Jahre 2013 übernommen hat und dieses weiterentwickelt, stellt Cisco selbst nicht noch ein zweites IDS zur Auswahl. Jedoch hat Cisco mit ihrer Produktreihe Secure IPS (NGIPS) ein Produkt geschaffen, welches die Funktionalität von Snort für Cisco Netzwerke erweitert und über die Intrusion Prevention Features einem Angriff direkt entgegenwirken kann [9].

3.4 McAfee Network Security Platform

Die McAfee Network Security Platform bietet neben der Intrusion Detection auch eine Intrusion Prevention an. Da die Firma McAfee viel Erfahrung in der Entwicklung von Antiviren-Software hat, bietet die Network Security Platform auch viel Funktionalität aus dem Bereich Deep Package Inspection. Dabei setzt die Lösung auf eine Kombination von statischer Codeanalyse, dynamischem Malware Sandboxing und Verfahren aus dem Bereich Machine Learning. Für diese Deep-Package Inspection bietet die Network Security Platform eine shared key solution zur Entschlüsselung des eingehenden und ausgehenden Netzwerkverkehrs an [30].

3.5 Trend Micro TippingPoint Threat Protection System (TPS)

Das Unternehmen TippingPoint war ein Unternehmen, welches sich auf Netzwerksicherheitsprodukte spezialisiert hatte, bevor es von der US-Japanischen IT-Sicherheitsfirma Trend Micro übernommen wurde. Trend Micro stellte ursprünglich kopiergeschützte

dongle her, bevor sich das Unternehmen einen Namen als Computer Sicherheitsunternehmen besonders in den Bereichen Endpoint Protection und Malware Protection machte. Aus diesem Grund bietet das Trend Micro TippingPoint Threat Protection System (TPS) auch viele Features aus dem Bereich Deep Package Inspection an. Um diese Funktionalität sicherzustellen, bietet die Lösung eine On-Box-SSL-Inspection an. Als Sicherheitstechnologien stellt das Trend Micro TippingPoint Threat Protection System automatische Sandbox Verfahren für verdächtigen Netzwerkverkehr bereit und setzt auf ihre patentierten Realtime Machine Learning Technologien [6]. Das im Patent US11,128,664 B1 geschützte Verfahren basiert auf einem Datastream von einer festen Länge. Die im Datenstream extrahierten Features hängen von der Art des untersuchten Netzwerk-traffics ab [6]. Eine angewendete statistische Methode funktioniert beispielsweise auf der Zählung von Sonderzeichen, Zahlen, Abständen, Buchstaben und Punktuation in einem Stream mit einer vordefinierten Länge. Dabei werden die gezählten Elemente in verschiedene Klassen eingeteilt. Anhand der so erhaltenen Daten lässt sich eine statistische Auswertung des aktiven Streams mit den vordefinierten Signaturen realisieren [6]. Weiter bietet die Lösung eine Erweiterung für hybride Cloudarchitekturenbereit [49].

3.6 Amazon GuardDuty

Viele Hersteller wie Trend Micro oder McAfee bieten eine Erweiterung der bestehenden IDS-Lösungen für hybride Cloudumgebungen an. Amazon selbst hält für den Schutz ihrer AWS-Infrastruktur jedoch ein eigenes Produkt für ihre Kunden bereit. Damit können AWS-Kunden ihre Passwörter, Daten, Workflows und ihre Infrastruktur, welche sie in der AWS-Cloud betreiben, schützen. Der grosse Vorteil dieser Lösung ist, dass diese direkt auf die Produkte der AWS-Cloud zugeschnitten ist. Jedoch benötigt der Kunde für sein eigenes LAN und zum Schutz seiner Arbeitsstationen auch weiterhin ein firmeninternes IDS [5].

3.7 IBM Security Network IPS

Die von IBM stammende Security Network Intrusion Prevention System (IPS) Lösung basiert auf dem open-source System Snort, stellt aber einige Erweiterungen zur Verfügung. Beispielsweise erweitert die IBM-Lösung Snort zu einem reaktiven System, welches direkt auf Angriffe reagieren kann. Zudem stellt die Lösung eine Unterstützung für IPv6 Netzwerke zur Verfügung [21].

3.8 Zusammenfassung der Markttrends

Durch die Betrachtung der obenstehenden Produkte konnten folgende Entwicklungen auf dem Markt von IDS festgestellt werden:

1. Einsatz von Deep Package Inspection: Viele Angriffe finden mittlerweile auf Application-Ebene statt [37]. Daher ist es wichtig, die übertragenen Daten zu analysieren und auch zu verstehen [15]. Zu diesem Zweck wird ein grosses Verständnis der Zielsysteme wie beispielsweise dem Application-Server, vorausgesetzt. Daraus lassen sich zwei Sub-Trends ableiten:

   (a) Die Entschlüsselung von Datenpaketen zur Analyse gewinnt an Bedeutung. Viele Angriffe finden mittlerweile auch über verschlüsselte Verbindungen statt [2]. Diese Angriffe sind sehr schwer und nur durch Machine Learning Methoden wie sie beispielsweise in der Cisco Encrypted Traffic Analytics eingesetzt werden [8] zu detektieren [2]. Jedoch haben solche Algorithmen oft eine sehr grosse False-Positive Rate und werden daher in der Praxis nur ungern eingesetzt [31]. Gerade in IPsec-verschlüsselten Netzwerken können Angriffe ohne Entschlüsselung des Datenverkehrs oft nicht entdeckt werden [55]. Die Hauptschwierigkeit ist das Pattern-Matching, welches in verschlüsselten Netzwerken nicht durchgeführt werden kann [34].

   (b) Damit das IDS das nötige Wissen über das Netzwerk und die darin angebotenen Services erlangen kann, bedarf es eines extrem hohen Konfigurationsaufwandes oder dem Einsatz von Machine Learning [13]. Da der Weg der Konfiguration oft fehleranfällig und statisch ist, setzen die meisten Hersteller mittlerweile auf selbstlernende Systeme [13].

2. Viele Unternehmen setzen mittlerweile teilweise oder bereits vollständig auf Cloud-Lösungen. Daher ist es wichtig, das firmeninterne Netz und die Services in der Cloud gleichermassen zu schützen. Dazu bieten einige Cloud-Anbieter wie Microsoft oder Amazon [5] geeignete Sicherheitsprodukte an. Diese bieten jedoch keinen Schutz für das firmeninterne Netzwerk und das Unternehmen ist weiterhin angreifbar. Daher bieten viele IDS-Entwickler bereits Lösungen für hybride Cloudumgebungen an.

3. Die dokumentierten Systeme zeigen deutlich, dass die Netzwerkadministratoren mittlerweile kaum mehr auf reine IDS setzen. Durch den hohen Datenverkehr in den Netzen und deren Komplexität werden praktisch nur noch IPS angeboten, welche viel schneller auf Angriffe reagieren können als ein Administrator. Zudem

sinken durch solche Systeme die Betriebskosten von Netzwerken, da es weniger administrativen Aufwand erfordert.

4 Netzwerksicherheitsarchitekturen

Angriffe auf ein Computernetzwerk kommen entweder aus dem Internet oder es wird über die Endpunkte Schadsoftware ins Netzwerk eingeschleust. Die Marktanalyse konnte aufzeigen, dass viele Hersteller nicht mehr lediglich auf einen klassischen IDS-Einsatz, sondern auf eine Orchestrierung über alle sicherheitsrelevanten Systeme setzen.
Dieses Kapitel soll die vorhandenen Sicherheitssysteme in einem modernen Computernetzwerk beschreiben.

4.1 Klassischer Einsatz eines IDS

Um ein IDS effektiv in ein bestehendes Netzwerk zu integrieren, bedarf es einer für das Unternehmen sinnvoller Netzwerkarchitektur mit einem klaren Platzierungskonzept der Services.

4.1.1 Schutz von Netzwerkzonen

Eine bewährte Technik zur Platzierung eines reinen Netzwerk-IDS ist die Platzierung der Sensoren bei den jeweiligen Netzwerkübergängen.

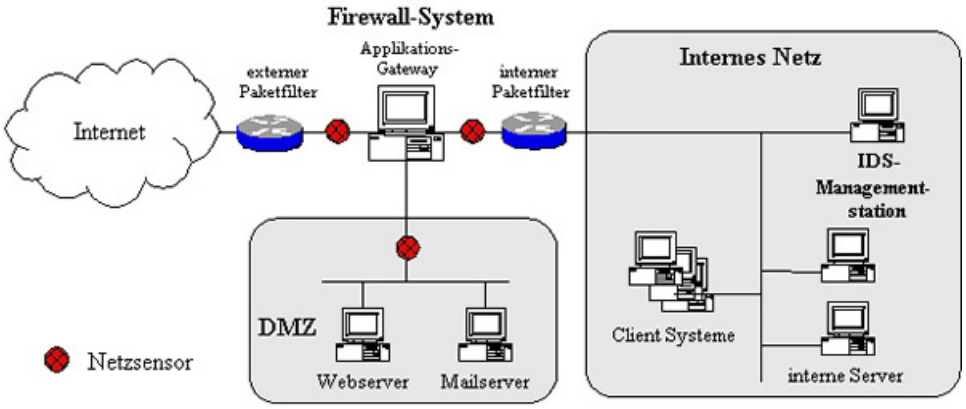


Abbildung 9: IDS zur Absicherung von Netzübergängen [7]

- Der Sensor der Internet-Firewall zeichnet alle Angriffe auf das Unternehmen auf. Diese Daten haben keine allzu grosse Relevanz für das Alerting des IDS, da diese Daten keine Aussage darüber treffen, ob die Angriffe von nachfolgenden Sicherheitskomponenten herausgefiltert werden oder nicht.
  Die hier aufgezeichneten Daten haben aber eine hohe Relevanz für die statistische

Auswertung. An dieser Stelle kann analysiert werden, wie häufig, in welcher Art und welche Systeme angegriffen werden. Über diese Analysen kann der Ausbau der firmeninternen IT-Sicherheit effektiv gesteuert werden [7].

- Die Sensorplatzierung zwischen Internet-Firewall und der DMZ ist einer der wichtigsten Indikatoren zur Erkennung und Abwehr von Angriffen auf die firmeneigene IT-Infrastruktur. Der Grund hierfür ist, dass alle Systeme, welche vom Internet her erreichbar sind, in dieser Netzwerk-Zone platziert werden sollten. Darunter fallen beispielsweise Webserver, Mail-Relay Server oder Proxy-Server [7].

- Der Sensor vor dem internen Netzwerk sollte keine Angriffe mehr detektieren, wenn alle Sicherheitsmassnahmen korrekt umgesetzt wurden. An dieser Stelle sollte es zudem keine direkte Kommunikation mit dem Internet geben, da die gesamte Kommunikation durch die Systeme der DMZ gefiltert werden sollten. Trotzdem macht der Netzsensor an dieser Stelle Sinn, um erfolgreiche Angriffe zu detektieren und den Schaden zu minimieren [7].

4.1.2 Schutz von ausgewählten Systemen

Netzsensoren haben den grossen Vorteil, dass diese sehr effektiv eine grosse Menge an Daten scannen können und daher ganze Netzwerkzonen effektiv überwachen können. Jedoch konnte in der vorliegenden Arbeit bereits festgestellt werden, dass diese Sensoren häufig bei verschlüsselten Verbindungen versagen. Daher ist es sinnvoll kritische Systeme zu identifizieren und auf diesen einen IDS Agenten, ein sogenanntes hostbasiertes IDS, zu installieren.

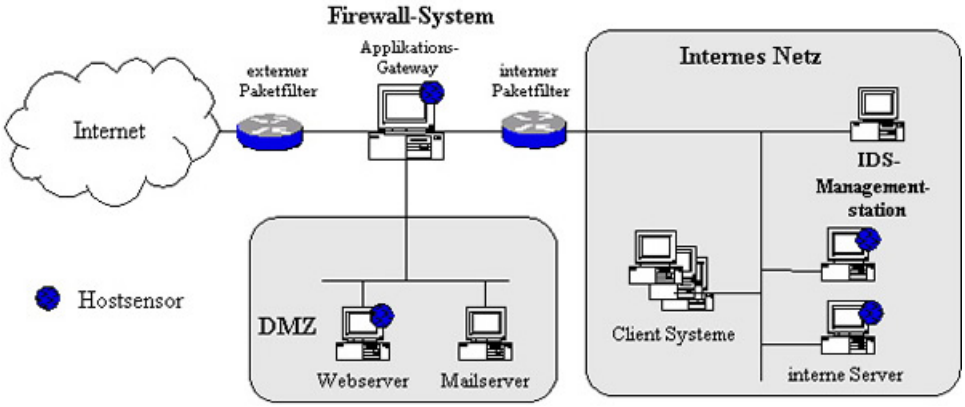


Abbildung 10: IDS zur Überwachung spezifischer Systeme [7]

Der Vorteil eines hostbasierten IDS ist, dass auf dem Host verschlüsselte Pakete entschlüsselt werden und daher der gesamte Netzwerkverkehr gescannt werden kann. Der

Nachteil dabei ist, besonders bei einem IPS, dass ein Angriff bis auf den Host gelangen kann.

4.1.3 Kombination von netzwerk- und hostbasiertem IDS

Das Bundesamt für Sicherheit in der Informationstechnik empfiehlt eine Kombination aus Netzwerksensoren und hostbasiertem IDS. Dadurch können Angriffe im Netzwerk frühzeitig erkannt werden und zusätzlich der Schutz gefährdeter Server und Clients durch ein hostbasiertes IDS erhöht werden [53]. Diese Architektur ermöglicht zudem verschlüsselte Angriffe effektiv zu detektieren.

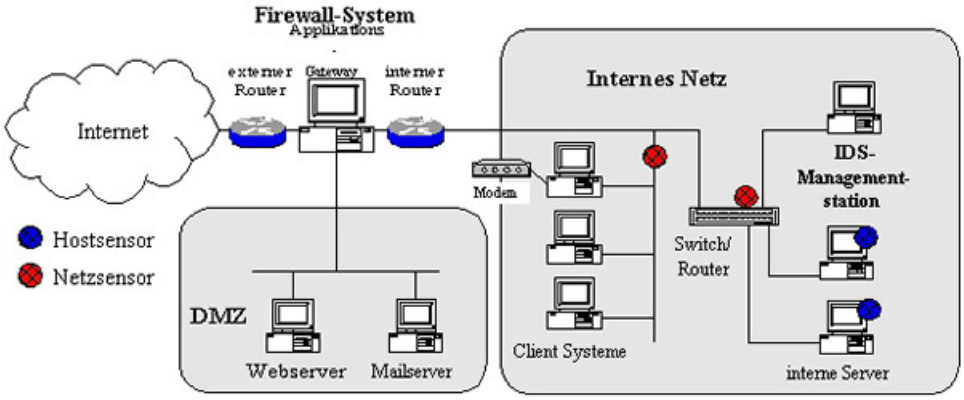


Abbildung 11: IDS zur Überwachung des internen Netzes [7]

Hierbei kann die Netzwerksegmentierung für zusätzliche Sicherheit sorgen [51]. Bei der Netzwerksegmentierung wird im internen Netzwerk für jeden Service eine zusätzliche Netzwerkzone mit bestimmten Freigaben erstellt. Diese segmentierten Zonen können effektiv mit host- und netzwerkbasierten IDS überwacht werden. Wird ein Service kompromittiert, kann der Angreifer durch die Netzwerksegmentierung nur sehr schwer weitere interne Systeme angreifen.

4.2 Schutz zum Internet

In modernen IT-Infrastrukturen werden in der DMZ eine Vielzahl an gehärteten Services installiert, um das interne Netzwerk und die darin angebotenen Services zu schützen. Zudem agieren diese Services als Interface zwischen dem Internet und den internen Services und Benutzern.
Diese Services sind im Gegensatz zu einem IDS oft nur auf dedizierte Anwendungen spezialisiert und bieten Services zur Angriffserkennung auf Appliklationsebene an wie beispielsweise Malware Erkennung, Spam-Erkennung oder Angriffe auf Webserver oder

Browser wie beispielsweise Cross-site scripting (XSS) oder SQL Injections.
Drei der wichtigsten DMZ-Sicherheitsservices werden untenstehend vorgestellt:

4.2.1 Webapplication Firewall (WAF)

Nach der gewählten Definition handelt es sich bei einer WAF, um ein auf HTTP/HTTPS spezialisiertes IPS. Eine Web Application Firewall kann den eintreffenden HTTPS-Traffic effizient entschlüsseln und auf bekannte Schwachstellen wie SQL-Injections oder Cross-Site Scripting (XSS) prüfen.
Der Vorteil einer WAF ist, dass Angriffe bereits vor dem Eintreffen auf den Webserver blockiert werden können. Zudem bieten moderne WAF-Systeme die Möglichkeit, die Authentifizierung beim Service bereits auf der WAF in der DMZ durchzuführen. Dadurch kommen keine nicht authentifizierten Verbindungen ins interne Netzwerk und jede Aktivität kann einem Benutzer zugeordnet werden.

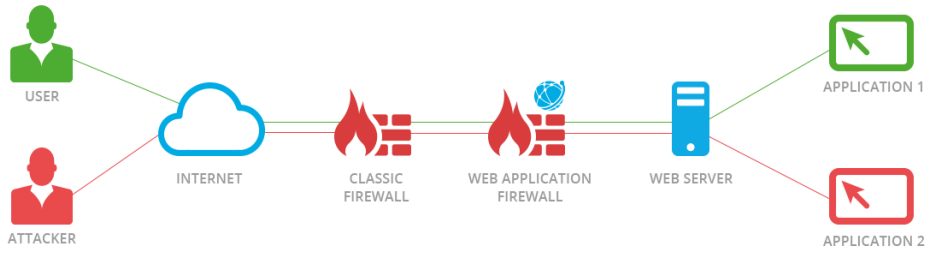


Abbildung 12: Web Application Firewall[52]

4.2.2 Proxy-Server

Ein Proxy-Server war ursprünglich eine Kommunikationsschnittstelle, die Webanfragen aus dem internen Netzwerk entgegennahm und als „Stellvertreter“ weiterleitete. Die HTTP-Responses aus dem Internet werden auf dem Proxy-Server als Webcache zwischengespeichert. Durch solche Webcaches lässt sich die Responstime langsamer Internetverbindungen minimieren, wenn häufig die selbe Webressource aufgerufen wird [25].
Moderne Proxy-Server bieten neben dem klassischen Webcache auch die Möglichkeit die HTTP-Responses auf Angriffe und Malware zu prüfen. Dabei wird der HTTP-Response auf Applikationsebene gescannt, bevor dieser an den Client im internen Netzwerk weitergeleitet wird.

4.2.3 Mail-Relay-System

Bei einem Mail-Relay-System oder SMTP-Relay handelt es sich um ein System, welches E-Mails empfangen, anhand von Blacklists und Content-Filter auf SPAM überprüfen und mittels Scaner-Software Angriffe und Schadsoftware detektieren kann. Je nach Art der Bedrohung wird das Mail bereits auf dem SMTP-Relay in die Quarantäne verschoben, oder es wird ein Flag im Mail-Header gesetzt, welches dem Mailserver anzeigt, dass es sich bei der Nachricht um SPAM handeln könnte. Danach leitet der Mail-Relay die Nachricht anhand einer Art Routing Table an das Zielsystem weiter.
Das Untenstehende Bild zeigt den Einsatz eines SMTP-Relays der Firma Symantec in der DMZ:

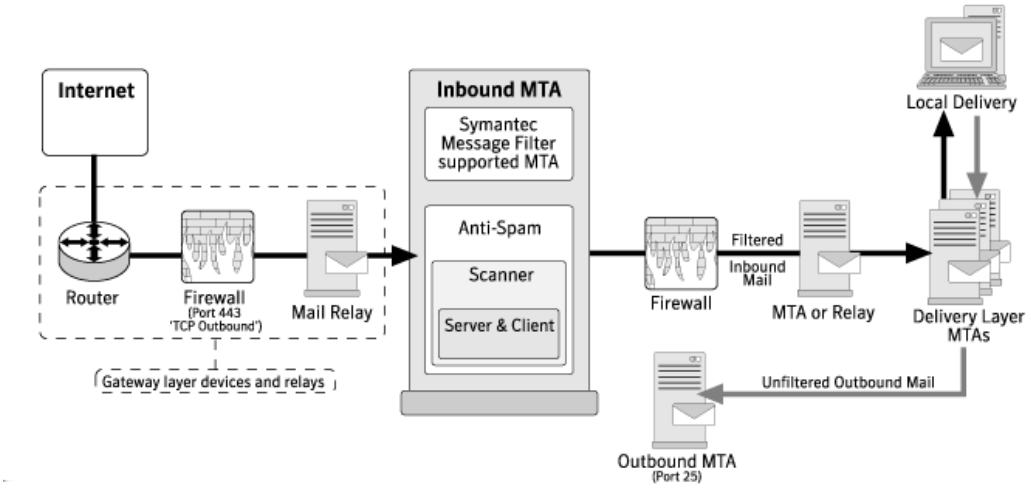


Abbildung 13: SMTP-Relay in der DMZ [45]

4.3 Schutz der Clients

Eine weitere Möglichkeit ein Netzwerk anzugreifen ist, wenn sich ein Angreifer physisch Zutritt zu einem Netzwerk verschafft. Dies kann über präparierte Devices wie Maus, Tastatur oder USB-Sticks geschehen oder durch physisches Verbinden mit einem Netzwerkport oder einem Wireless-Access-Point im Unternehmen.
Moderne Auswertungen zeigen, dass 63% der erfolgreichen Attacken von internen Quellen ausgehen [46]. Eine Auswertung von Cybersecurity Insiders ergab, dass die Organisationen die Problematik erkannt haben. Die befragten Firmen gaben zu 52% an, dass die Aufdeckung und Abwehr von internen Attacken schwieriger ist und 68% der befragten Organisationen bestätigten, dass interne Attacken häufiger wurden [20].

4.3.1 802.1X port-based Network Access Control (PNAC)

Um den Netzwerkzugang zu schützen, wurde in der Vergangenheit eine Port-Security auf den Netzwerkswitches konfiguriert. Dabei handelte es sich um eine Liste von MAC-Adressen, die sich mit dem Netzwerk verbinden durften. Diese Technik verursachte einen hohen Administrationsaufwand und war via MAC Spoofing einfach zu umgehen. Mit dem 802.1X port-based Network Access Control Standard kann der Netzwerkzugang an einem physischen Port im LAN oder per WLAN mittels eines RADIUS-Authentifizierungsservers zentralisiert werden [22].

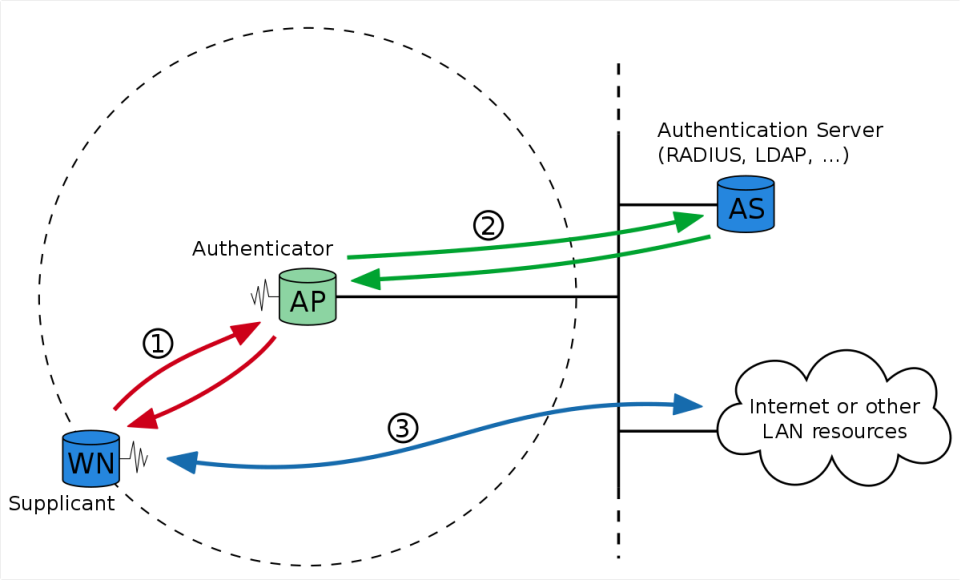


Abbildung 14: IEEE 802.1X Authentifizierung [54]

1. Der Client verbindet sich via LAN-Kabel oder WLAN mit dem Netzwerk und authentifiziert sich beim Authenticator.

2. Der Authenticator leitet die Anfrage an den RADIUS Server weiter. Der Radius Server kann beispielsweise das Computerzertifikat oder auch LDAP-Anmeldeinformationen überprüfen. Anbieter wie Microsoft gehen noch einen Schritt weiter und bieten mit dem Network Policy Server (NPS) die Möglichkeit eine Vielzahl an clientseitigen Prüfungen vor dem Netzwerkzugang durchzuführen. So kann beispielsweise geprüft werden, ob die neusten Updates installiert wurden oder das Antivieren-Programm auf dem aktuellen Stand gehalten wurde [33].

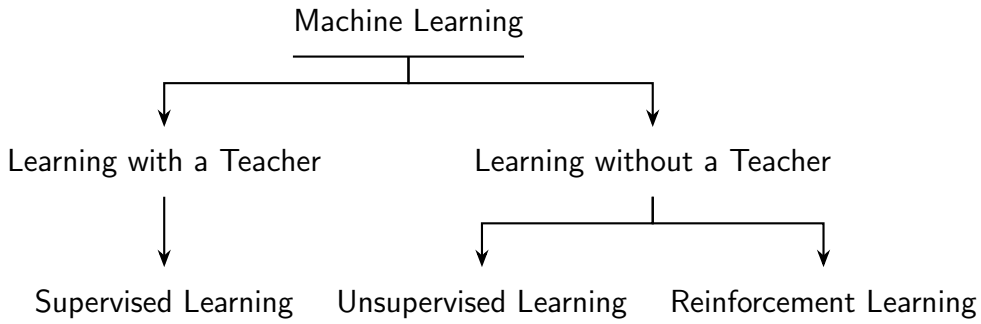
3. Erst nach einer erfolgreichen Überprüfung dieser Richtlinien erlaubt der NPS oder RADIUS-Server den Zugang zum Netzwerk.

4.4 Schlussfolgerungen Netzwerksicherheitsarchitekturen

Dieses Kapitel konnte aufzeigen, dass in modernen Netzwerken der reine Einsatz eines IDS nicht mehr ausreicht. Um Netzwerke erfolgreich zu schützen, bedarf es einer Kombination aus IDS/IPS und weiteren Komponenten. Diese Systeme sind meist auf den Schutz eines bestimmten Services ausgerichtet und agieren auf Applikationsebene. Diese Prüfungen auf dem Application-Layer ermöglichen einen zielgerichteten Schutz und verhindern das Eindringen von Schadsoftware oder Angreifern in das interne Netzwerk.
Ein erfolgreicher Schutz von Computernetzwerken kann daher nur durch eine sinnvolle Netzwerkarchitektur und Orchestrierung der sicherheitsrelevanten Services gewährleistet werden.

5 Machine Learning Grundlagen

Im Bereich des maschinellen Lernens wird sehr aktiv geforscht. Das Frauenhoferinstitut zeigte in einem Bericht aus dem Jahr 2018 auf, dass die Anzahl von Beiträgen in Fachzeitschriften bereits über 20'000 liegt und zusätzlich nochmals etwa soviele Conference-Papers kommen. Besonders stark zugenommen haben Forschungsbeiträge im Bereich Deep Learning [23].
Aufgrund der sehr aktiven Forschung ist es schwer an dieser Stelle eine komplette Übersicht über die Trends und über den aktuellen Stand der Forschung in diesem Bereich aufzuzeigen. Diese Arbeit beschränkt sich daher lediglich auf Verfahren, die zur Detektion von Angriffen auf Computernetzwerke verwendet werden können.
Computer lernen grundsätzlich sehr ähnlich wie Menschen das tun. Menschen können etwas entweder über einen Lehrer oder selbst durch Erfahrung lernen. Im Bereich des maschinellen Lernens wird daher ebenfalls in die Bereiche überwachtes Lernen (Supervised Learning) und unüberwachtes Lernen (Unsupervised Learning) oder bestärkendes Lernen (Reinforcement Learning) unterteilt [19].



5.1 Supervised Learning

Beim Supervised Learning wird dem Machine Learning Algorithmus die gewünschte Lösung für jeden Trainingsdatensatz mitgegeben. Diese Lösung wird auch Label oder Ereignisvariabel genannt und oft mit Y bezeichnet [14].

Die Parameter werden so optimiert, dass entweder die Differenz des berechneten Outputs des Machine Learning Algorithmus und des Labels oder die Misklassifikationen minimiert werden - je nachdem, ob die Lösung kontinuierlich oder kategorisch ist. Die Parameter des Machine Learning Algorithmus werden somit zu Beginn zufällig oder ungefähr gewählt. Iterativ wird nun der Output des Learners mit den Training-Labels verglichen und die Prameter werden laufend angepasst bis eine Konvergenz erreicht wurde [26].

Supervised Learning wird in folgenden zwei Teilgebieten angewandt [14]:

- Klassifizierung (Kategorien voraussagen)
- Regression (Werte voraussagen)

Folgende Algorithmen bilden eine nicht vollständige Liste der wichtigsten Supervised Learner [14]:

- k-Nearest Neighbour
- Linear Regression
- Logistic Regression
- Support Vector Machines
- Decision Trees und Random Forests

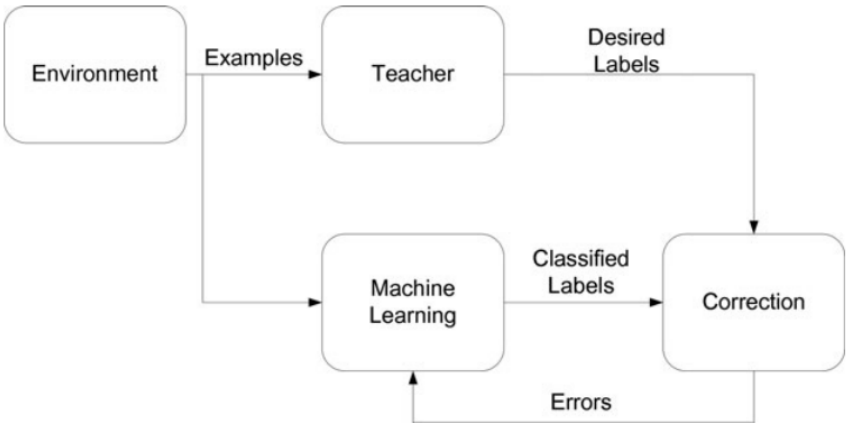


Abbildung 15: Supervised Learning Prozess [26]

5.1.1 Klassifizierung

Die Klassifizierung ist definiert als der Prozess der Zuweisung jedes Datensatzes zu einer oder mehreren vordefinierten Kategorien. Die einfachste Klassifizierungsart ist die binäre Klassifizierung, da hier der Datensatz in eine von nur zwei Kategorien - Wahr oder Falsch - eingeteilt wird. Wenn mehr als zwei, also M, $M > 2$ vordefinierte Kategorien zur Verfügung stehen spricht man von einer mehrfachen Klassifizierung. Die mehrfache Klassifizierung kann jedoch in M binäre Klassifizierungen zerlegt werden [26].

5.1.2 Regression

Die Regression ist definiert als der Prozess der Schätzung eines Ergebnisses aufgrund mehrerer Faktoren. Die Outputwerte einer Regression sind kontinuierlich, im Gegensatz zu denen der Klassifizierung, welche diskret sind[26].

Es existieren zwei Typen von Regression [26]:

- Univariate Regression: Ein Outputwert wird geschätzt.
- Multivariate Regression: Mehrere Outputwerte werden geschätzt.

Das Ziel der Regression ist es, eine Funktion zu finden, die mithilfe eines Inputverktors - den Features - ein möglichst genaues Ergebnis voraussagen kann.

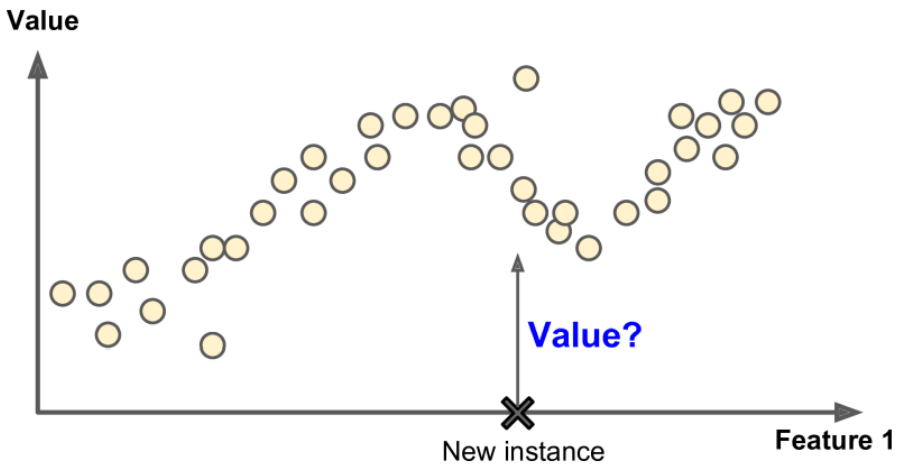


Abbildung 16: Regression [14]

5.1.3 Einsatz von Supervised Learning zur Angriffserkennung

Decision Tree

Der Decision Tree Algorithmus wird für Klassifikation sowie auch für Regression verwendet und hat eine baumartige Struktur mit Knoten, Ästen und Blättern. Jeder Knoten repräsentiert ein Attribut oder ein Feature. Ein Ast repräsentiert eine Entscheidung oder eine Regel, wobei jedes Blatt ein möglicher Output darstellt. Der Decision Tree Algorithmus selektiert automatisch die wichtigsten Merkmale aus einem potenziellen Angriff und lässt diese die Baumstruktur durchlaufen [4].

K-Nearest-Neighbour

Der KNN Algorithmus benutzt die Idee der Ähnlichkeit der Features um die Kategorie eines Datensatzes zu bestimmen. Der Datensatz wird identifiziert, indem die Distanz zu den Nachbarn berechnet wird. Die Anzahl der betrachteten Nachbarn k ist ein Hyper-Parameter, der die Qualität des Algorithmus beeinflussen kann. Merkmale eines potenziellen Angriffs können in einer hochdimensionalen Struktur angeordnet werden. Der Abstand zu den Nachbarn kann zum Beispiel mithilfe des Euklidischen Abstands berechnet werden [4].

Support Vector Machine

Beim Support Vector Machine Algorithmus werden die Features auch in einer hochdimensionalen Struktur angeordnet. Mithilfe der Support Vektoren kann eine Hyperebene zwischen den Datenpunkten konstruiert werden, die diese voneinander trennt und somit die Kategorisierung vornimmt [4].

5.2 Unsupervised Learning

Das Unsupervised Learning gehört zu den Machine Learning Methoden ohne Lehrer. Dies bedeutet, dass zu der Menge an p Features $X_1, X_2, ..., X_p$ welche in n verschiedenen Versuchen gemessen wurden, die Ergebnisvariabel Y vollkommen unbekannt ist [24]. Während es beim Supervised Learning oft darum geht, die Ergebinsvariabel Y zu bestimmen, geht es beim Unsupervised Learning meist darum, durch Anordnung und Gruppierung der Features $X_1, X_2, ..., X_p$ Muster zu erkennen und daraus neue Schlüsse zu ziehen [24].

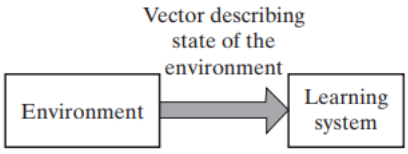


Abbildung 17: Unsupervised Learning Prozess [19]

5.2.1 Clustering

Clustering wird in der Statistik zu den explorativen Datenanalyse Methoden gezählt und ermöglicht es in Datensätzen ohne Vorwissen Abhängigkeiten aufzuzeigen [38]. Dabei werden die Daten in Untergruppen, den sogenannten Clustern, unterteilt. Durch unterteilen in Gruppen können beispielsweise bei Marktanalysen die Kunden nach verschiedenen Bedürfnissen aufgeschlüsselt werden.
Die wohl bekannteste Clustering-Methode beschreibt das K-Means Clustering. Dabei werden die Daten in k unterschiedliche Gruppen ohne Überlappung unterteilt [24]. Die Datenpunkte in der Mitte des Clusters werden Centroide genannt. Von jedem Centroid aus wird die Distanz zu den unterschiedlichen Datenpunkten gemessen und die Datenpunkte dem Cluster zugeordnet, zu dem er die geringste Distanz hat [24]. Sobald sehr stark verrauschte Daten analysiert werden, kann ein Preprocessing der Daten vor der Cluster-Analyse sinnvoll sein.
Das untenstehende Bild zeigt eine Cluster-Analyse mit $k = 3$ Clustern und zwei Features X_1 und X_2.

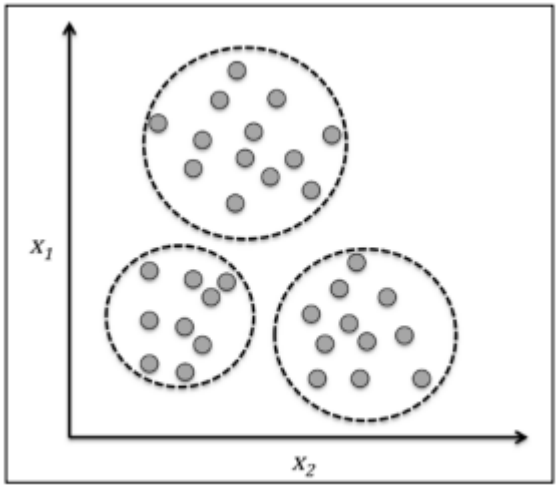


Abbildung 18: Data Clustering [38]

5.2.2 Dimensionsreduktion

Im Machine Learning wird oft mit sehr grossen Datenmengen mit vielen Features gearbeitet. Diese Daten sind in der Praxis oft verrauscht und müssen mittels Preprocessing vorbearbeitet werden. In diesem Schritt kommen oft Dimensionsreduktionsalgorithmen zum Einsatz, welche die Datensätze verkleinern und dadurch die Rechenzeit verkürzen [38]. Jedoch können durch dieses Preprocessing auch relevante Informationen herausgefiltert werden und daher die gewonnenen Informationen aus den Datensätzen verschlechtert werden. Eine gängige Methode zur Dimensionsreduktion ist es zu prüfen ob zwei Features eine hohe Korrelation haben. Sollte dies der Fall sein, werden für die weitere Bearbeitung der Daten nicht beide Features benötigt [24].
Die Abbildung 19 zeigt eine nicht lineare Dimensionsreduktion der Features X_1, X_2 und X_3 auf die Dimensionen Z_1 und Z_2. Solche Methoden werden oft angewendet, um Daten besser zu visualisieren [38].

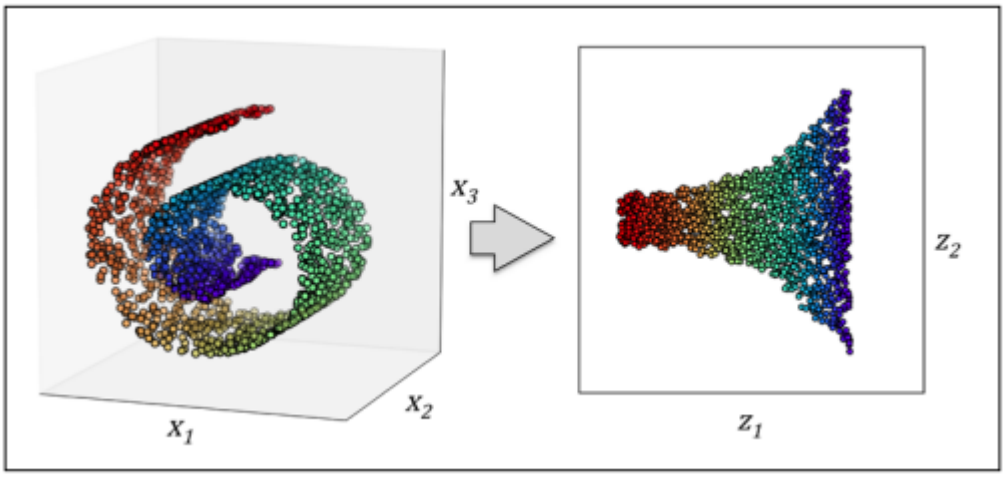


Abbildung 19: Dimensionsreduktion [38]

5.2.3 Einsatz von Unsupervised Learning zur Angriffserkennung

Werden alle HTTP-Requests gespeichert, eignen sich diese Daten für eine Cluster Analyse. Dies funktioniert unter der Voraussetzung, dass ein Grossteil der HTTP-Requests gutartiger Traffic ist und für denselben Webservice eine gewisse Ähnlichkeit aufweist. Durch eine Datenanalyse können Cluster gebildet werden. Befindet sich ein Zugriff weit entfernt von einem Centroid, kann dies auf einen Angriff hinweisen.

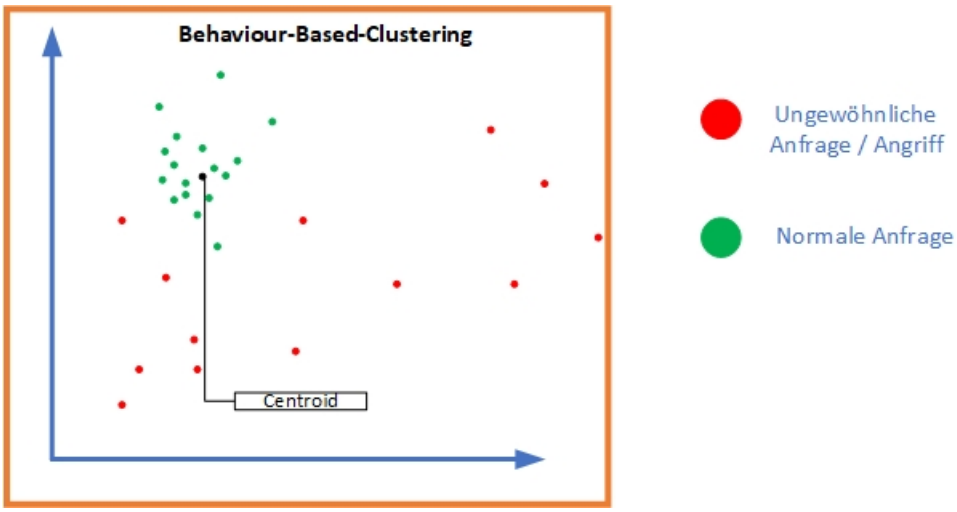


Abbildung 20: Angriffserkennung durch Cluster

5.3 Reinforcement Learning

Reinforcement Learning oder bestärkendes Lernen beschreibt ein System, das einzig durch die Interaktion mit der Umgebung versucht eine Aufgabe bestmöglich zu lösen. Das Reinforcement Learning funktioniert also ohne Lehrer, jedoch durch ein ständiges Feedback in Form einer Belohnung, welches durch Interaktion mit der Umgebung generiert wird. Wie in Abbildung 21 gezeigt, wird das Learning-System um einen Kritiker herum gebaut. Dieser Kritiker erhält ein leicht zeitlich verzögertes primary reinforcement Signal von der Umgebung als Skalar und wandelt dieses in das ebenfalls skalare höherwertige heuristic reinforcement Signal um [19]. Durch die zeitliche Verzögerung ist die Wirkung der ausgeführten Aktionen bereits messbar und das heuristic reinforcement Signal gibt dem Learning-System ein Feedback darüber, wie effektiv die gestellte Aufgabe umgesetzt wurde. Das Ziel vom Reinforcement Learning ist es, die Belohnungen zu maximieren, was über das skalare heuristic reinforcement Signal gemessen werden kann [44].

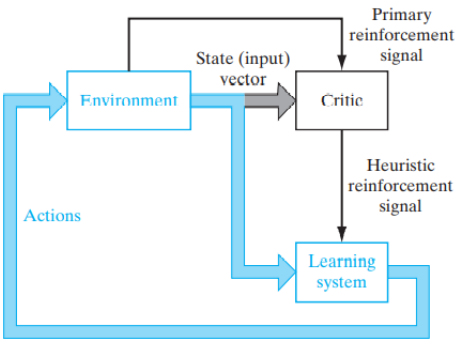


Abbildung 21: Reinforcement Learning Prozess [19]

5.3.1 Einsatz von Reinforcement Learning zur Angriffserkennung

Reinforcement Learning kann dann sinnvoll zur Angriffserkennung in Netzwerken eingesetzt werden, wenn der Kritiker erkennen kann, ob es sich um einen Angriff handelt oder nicht. In der Praxis könnte ein Reinforcement-Modell wie folgt realisiert werden:

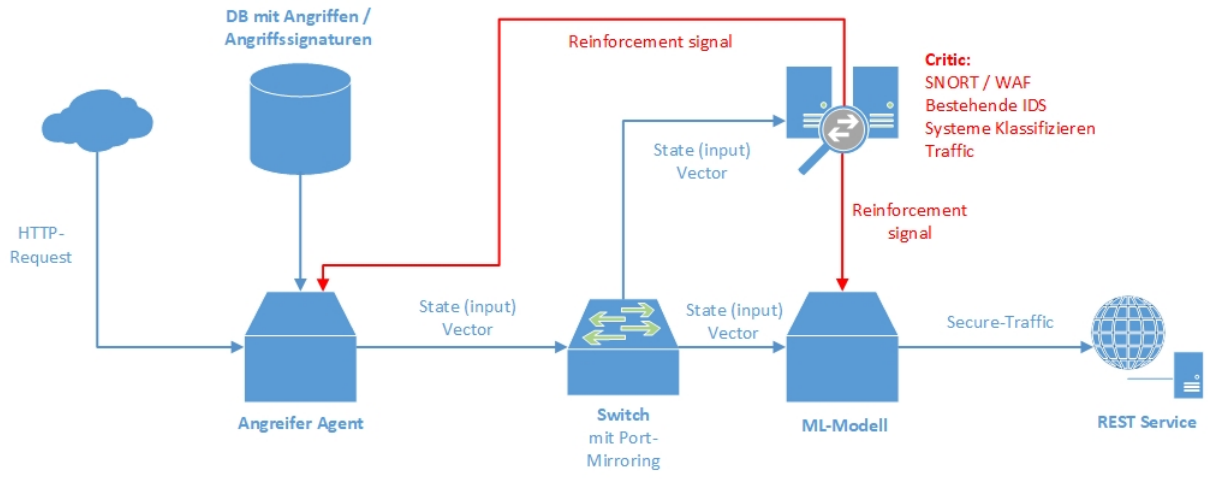


Abbildung 22: Reinforcement Learning Agent System

Trainings-Aufbau:
Der Angreifer Agent agiert als Proxy und leitet die HTTP-Requests aus dem Internet an das IDS-ML-Modell weiter. Verschiedene Kritiker, bestehend aus IDS-Systemen mit statischen Signaturdatenbanken, analysieren den Traffic und generieren das Reinforcement Signal für den Angreifer und das IDS-ML-Modell, damit diese sich stetig verbessern können.
Damit alle bekannten Angriffe eintrainiert werden können, verfügt der Angreifer über eine Datenbank von bekannten Angriffen und Angriffsignaturen, welche er senden kann. Durch diesen Aufbau generiert der Angreifer immer bessere Angriffe und das IDS-ML-Modell verbessert sich auch schrittweise, bis der Sicherheitsstandard der bestehenden IDS Produkte erreicht ist.

Produktions-Aufbau:
Nach dem Eintrainieren des IDS-ML-Modells können keine bestehenden Produkte mehr als Kritiker eingesetzt werden, da sich das Modell sonst nicht über den Stand der bestehenden Produkte hinaus entwickeln kann. Daher muss für den produktiven Einsatz ein anderes Verfahren für das Reinforcement Signal verwendet werden. Dieses Verfahren müsste in einer weiterführenden Arbeit entwickelt werden.

5.4 Neuronale Netzwerke

Neuronale Netze sind ein Programmierparadigma, welches dem menschlichen Gehirn nachempfunden ist [18]. Die elementaren Bausteine eines Neuronalen Netzwerks sind die Neuronen, welche aus einer gewichteten Eingabe, einer Summierfunktion, einer

Aktivierungsfunktion und einer Ausgabe bestehen [19]. Ein Neuronales Netz selbst besteht aus einer grossen Anzahl solcher künstlichen Neuronen, welche in verschiedene Schichten geordnet werden. Die erste Schicht wird Input-Layer und die letzte Schicht Output-Layer genannt. Alle Schichten dazwischen werden Hidden-Layer genannt [18]. Werden die Informationen nur in eine Richtung von einem Layer in den nächsten weitergeleitet, spricht man von einem Feedforward-Netzwerk [19]. Sobald das Netzwerk aber über mindestens einen Feedback-Loop verfügt, wird von einem rekurrenten Netzwerk gesprochen. Diese rekurrent verschalteten Netzwerke entsprechen eher dem biologischen Modell des menschlichen Gehirns und werden benutzt, um zeitlich codierte Informationen in den Daten zu entdecken [27].
Die Einteilung von Neuronalen Netzen nach Lernmethode ist nicht möglich, da es sich bei Neuronalen Netzwerken um eine Architektur handelt, die in allen drei Lernklassen angewendet werden kann. Aufgrund der starken Forschung im Bereich Deep-Learning, beschäftigen sich immer mehr Arbeiten mit rekurrenten Neuronalen Netzen [23].

5.4.1 Modell eines Neurons

Die Abbildung 24 zeigt das Modell eines Neurons, welches für die Informationsverarbeitung in Neuronalen Netzen eingesetzt wird. Ein Neuron besteht aus folgenden Komponenten:

1. Die Informationen x_j werden über die Synapsen oder connecting links an das Neuron k weitergeleitet. Jede Synapse j wird mittels weight w_{kj} oder strength gewichtet. In dieser Gewichtung liegt die eigentliche Intelligenz des Neurons. Beim Training eines Neuronalen Netzes werden die Werte w_{kj} jedes Neurons optimiert, bis aus den Daten x_j die gewünschten Informationen gewonnen werden können. Im Gegensatz zum menschlichen Gehirn, kann der Input in ein Computer-Neuron x sowohl ein positives als auch ein negatives Vorzeichen haben [19].

2. Die Summierungsfunktion $v_k = \sum_{j=1}^{m} w_{kj}x_j + b_k$ wird als linear compiner der gewichteten Inputdaten verwendet. Der Bias b_k wird verwendet um den Netto-Input der Aktivierungsfunktion zu erhöhen oder abzusenken [19]. Im Raum $x, y \in \mathbb{R}^2$ mit der Beziehung $y = a * x + b$ kann man sich den Bias b als Verschiebung entlang der y-Achse vorstellen:

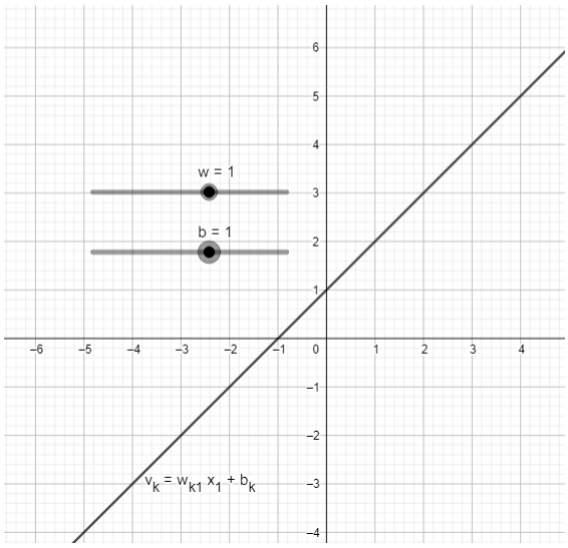


Abbildung 23: Summierungsfunktion mit einer Synapse

3. Die letzte Komponente eines Neurons stellt die Aktivierungsfunktion dar. Eine Aktivierungsfunktion verhindert, dass nicht signifikante Werte in einem Neuronalen Netz nicht zu einer Art rauschen führen und zudem wird die Ausgabeamplitude eines Neurons begrenzt. Zusätzlich wird die Aktivierungsfunktion als „squashing function“ verwendet, welche die Ausgabewerte eines Neurons in einen festgelegten Wertebereich setzt [19].

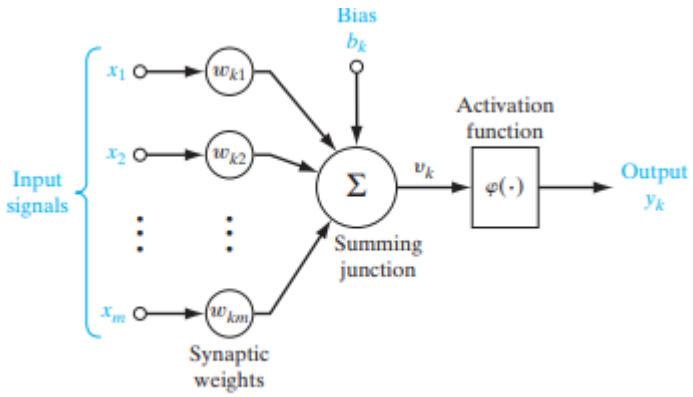


Abbildung 24: Nichtlineares Modell eines Neurons [19]

5.4.2 Aktivierungsfunktionen

Es gibt zwei Arten von Aktivierungsfunktionen:

1. Die erste Klasse von Aktivierungsfunktionen stellen die Thresholdfunktionen dar.

$$\varphi(v_k) = \begin{cases} 1 \text{ if } v_k \geq 0 \\ 0 \text{ if } v_k < 0 \end{cases}$$

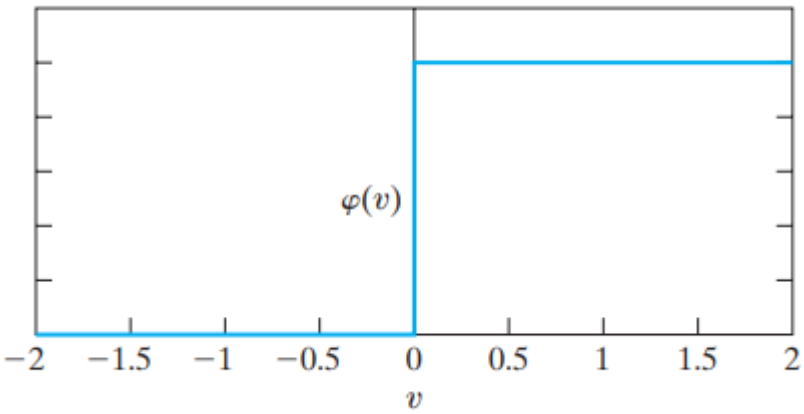


Abbildung 25: Thresholdfunktion [19]

Die Funktion ist sehr einfach im Programmcode zu implementieren. Der Nachteil ist, dass die Funktion lediglich eine harte Grenze kennt und dadurch Informationen sehr konservativ behandelt werden [39].

2. Um das abrupte Verhalten der Stufenfunktion zu verringern, kann eine S-förmige Funktion (Abbildung 26) verwendet werden. In der Theorie von Neuronalen Netzen wird eine Vielzahl von unterschiedlichen S-Funktionen verwendet. In der Praxis zeigte sich aber, dass die Funktion $y = \frac{1}{1+e^{-x}}$ einfach und von Computern sehr performant berechnet werden kann [39].
   Über den Parameter $a \in \mathbb{R}$ kann die Intensität der Aktivierungsfunktion angepasst werden. Für $a \to \infty$ ergibt sich die Thresholdfunktion [19].

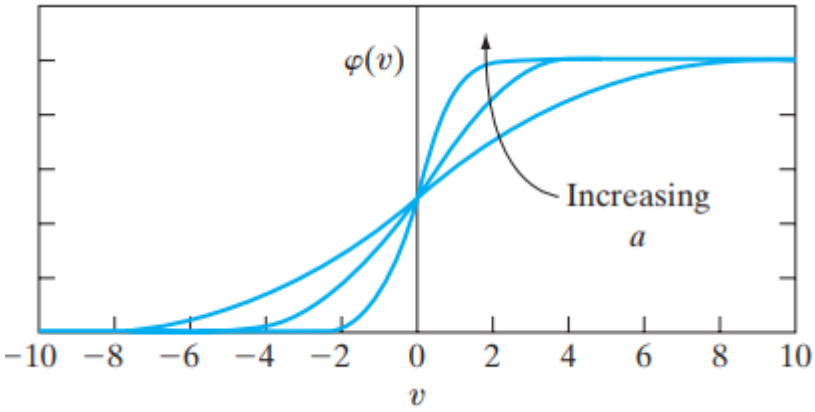


Abbildung 26: Sigmoidfunktion [19]

5.5 Lernen von Datenströmen

Die Datenmengen im Bereich des Machine Learning nehmen aufgrund der Entwicklung der Informations- und Kommunikationstechnologie rasch zu. Oft liegen diese Daten in Form von Streams vor und stellen die 5V-Charakteristik von Big Data dar: volume, velocity, value, variety und veracity. Traditionelle batchbasierte Machine Learning Modelle, die durch einen statischen, historischen Datensatz trainiert werden, können sich in dieser neuen Umgebung nicht behaupten. Zum einen ist es unmöglich alle Samples zu registrieren und zum anderen werden die alten Modelle obsolet, da sich die interne Dynamik in der Umwelt laufend verändert. Beim traditionellen Machine Learning wird angenommen, dass aktuelle und zukünfigte Datensätze genau gleich verteilt sind. Das ist jedoch in den heutigen Data Streams nicht immer der Fall. Das Lernen aus dieser ständig wachsenden, dynamischen und sich weiterentwickelnden Datenmenge erfordert flexible Lernmodelle, die sich im Laufe der Zeit selber anpassen können. Diese Learner müssen unter folgenden Bedingungen funktionieren[41]:

- Die Daten kommen in riesigen Mengen und mit einer hohen Geschwindigkeit.

- Der zufällige Zugriff auf Beobachtungen ist nicht möglich oder ist mit hohen Kosten verbunden.
- Der verfügbare Speicher ist klein im Gegensatz zur Datenmenge.
- Die Verteilung der Daten kann sich im Laufe der Zeit verändern (Concept Drift).
- Die Anzahl der Klassen kann sich im Laufe der Zeit verändern (Concept Evolution).
- Data Streams können von mehreren Quellen, Seiten oder Benutzern generiert und behandelt werden. Somit ist eine zentrale Datenverarbeitung nicht mehr effizient und Verteilte Systeme sind dazu geeigneter.

Aufgrund der oben genannten Einschränkungen müssen Modelle, die von Data Streams lernen, folgendes leisten können[41]:

- Inkrementelles Lernen, um die neuen Informationen in das Modell zu integrieren, die neu ankommen.
- Dekrementelles Lernen, um die nicht mehr nützlichen Datenmuster zu vergessen oder zu verlernen.
- Neuheitserkennung, um neue Konzepte zu lernen.

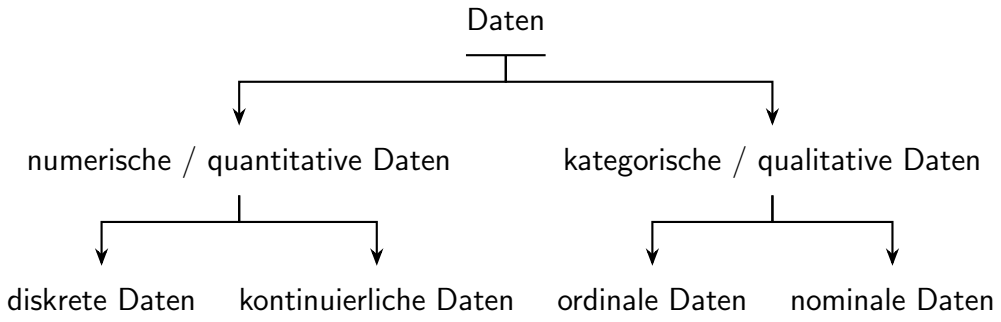
Es gibt mehrere Ansätze für maschinelles Lernen und Data Mining, die entwickelt wurden um aus Data Streams zu lernen, wenn ein Concept Drift und/oder eine Concept Evolution vorliegt.
Sogenannte informierte Methoden detektieren Veränderungen (Concept Drift) explizit durch die Überwachung von bestimmten Merkmalen wie z.B. der Leistung des Learners oder der Datenverteilung im Feature Space. Sobald eine Veränderung festgestellt wurde, lernen die informierten Methoden das Modell mit Hilfe einer aktuellen Auswahl von Datensamples um. Das Konzept kann in folgenden drei Schritten beschrieben werden [41]:

- Der Concept Drift wird durch Veränderung detektiert.
- Die Datensampels, mit denen der Learner umgelernt wird werden selektiert.
- Es wird entschieden auf welche Art und Weise der Learner an die neue Situation angepasst wird.

Das Lernen aus Data Streams mit Concept Evolution basiert auf der Verwendung von Clustering-Techniken, um die Kohäsion zwischen den Datenproben zu bewerten, die zu keiner der bestehenden Klassen gehören [41].
Die sogenannten blinden Methoden passen den Learner in regelmässigen Zeitabständen implizit an das aktuelle Konzept an, ohne dass ein Drift erkannt wird. Sie verwerfen Konzepte mit einer konstanten Geschwindigkeit, unabhängig davon, ob Änderungen aufgetreten sind oder nicht [41].

5.6 Datentypen

Bei der Analyse von Netzwerktraffic kommt es zu einer Vielzahl an unterschiedlichen Daten, die vom Machine Learning Modell anders behandelt werden müssen. In der Statistik werden folgende Datentypen unterschieden:



Quantitative Daten sind strukturiert und eignen sich für statistische Methoden, während qualitative Daten unstrukturierte Daten wie beispielsweise geschriebener Text sein kann [11]. Bei den qualitativen Daten wird nach nominalen Daten ohne logische Ordnung und nach ordinalen Daten mit einer logischen Ordnung unterschieden [11]. Quantitative Daten sind entweder diskret wie IP-Adressen, oder kontinuierlich wie ein analoger Messwert. Meist werden beim Preprocessing der Daten diese in eine Form gebracht, mit welcher weitere Muster durch statistische- oder Machine Learning-Methoden aufgedeckt werden können.

5.6.1 Datensätze

KDCup99 Dataset

Der bekannteste und am weitesten verbreitete Datensatz für Intrusion Detection ist der KDCup99 Datensatz [18]. Dieser besteht aus zwei Untergruppen. Einem Datensatz mit 5 Millionen Einträgen für das Training von Algorithmen und einem Testdatensatz mit 3 Millionen Einträgen für das Testen der Algorithmen [18]. Die Aufteilung des Datensatzes in ein Trainingsdatensatz und einen Testdatensatz ist wichtig, dass sich das Machine Learning Modell nicht auf den Datensatz „spezialisiert“ und es zu einem

overfitting kommt.
Jeder Eintrag in diesem Datensatz hat 41 Features welche in der Studie „IDS Using Machine Learning -Current State of Art and Future Directions“ [18] detailliert beschrieben sind.
Durch die weite Verbreitung des Datensatzes gibt es auch weitreichende Studien zur Performance diverser ML-Algorithmen auf diese Daten. Besonders vielversprechend erscheinen Assoziationsanalysen (Association rule mining) zur Erkennung von Angriffen. Diese Algorithmen untersuchen Korrelationen zwischen den Features, beispielsweise mit welchem Protokoll normalerweise auf eine bestimmte Destination zugegriffen wird [47]. Solche Algorithmen erkennen bis zu 99% der Angriffe [47]. Diese generieren jedoch mit einer False positive Rate von ca. 3.3% eine relativ hohe Rate an Fehlalarmen [47] und sollten möglichst zusammen mit weiteren Methoden eingesetzt werden.
Mit dem **NSL-KDD** ist eine überarbeitete Version des KDCup99 Datensatzes verfügbar, welcher von den häufigsten Fehlern befreit wurde [4].

HTTP CSIC 2010 Dataset
Ein rein auf Webapplikationen spezialisierter Datensatz ist der HTTP CSIC 2010 Datensatz welcher vom Information Security Institute CSIC (Spanish Research National Concile) herausgegeben wurde [17]. Dieser Datensatz beinhaltet 36’000 normale HTTP-Anfragen und 25’000 HTTP-Angriffe [17]. Dabei werden verschiedenste Angriffsmethoden wie SQL-Injection, XSS Attacken und weitere verwendet.
Dar Datensatz wird häufig in Publikationen, welche reine HTTP-Angriffserkennung durch ML-Algorithmen untersuchten, zitiert und verwendet [17].

Kyoto 2006+
Die Universität von Kyoto hat einen Datensatz herausgegeben, welcher Daten vom Jahr 2006 bis 2015 enthält. Dabei wurden die Daten von verschiedenen honeypots, darknet Sensoren, E-Mail-Servern und weiteren Systemen zusammengetragen. Der Datensatz beinhaltet 24 verschiedene Features [4].

UNSW-NB15
Dieser im Jahr 2015 vom Australian Center for Cyber Security herausgegebene Datensatz beinhaltet ca. 2 Millionen Einträge mit je 49 verschiedenen Features. Die darin beinhalteten Angriffe sind Backdoors, DoS, Exploits, Fuzzers, Generic, Port scans, Reconnaissance, Shellcode und worms [4].

CIC-IDS2017 und CSE-CIC-IDS2018
Der im Jahr 2017 vom Canadian Institute of Cyber Security (CIC) herausgegebene Datensatz CIC-IDS2017 beinhaltet normalen Datenverkehr und einige echte Cyber-Attacken.
Im Jahr 2018 wurde dann vom CIC und Communications Security Establishment (CSE) der Datensatz CSE-CIC-IDS2018 herausgegeben. Der Datensatz beinhaltet folgende Angriffe: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web-Attacken und Infiltration des Netzwerkes von Intern [4].

Aufgrund der stetig wachsenden Anzahl neuer Attacken werden auch immer neue Datensätze veröffentlicht. Diese Datensätze können sehr hilfreich sein, die ML-Algorithmen auf neue Angriffe zu trainieren. Jedoch ist es für die Wahl des ML-Modells besser einen älteren und besser untersuchten Datensatz zu verwenden, um über mehr wissenschaftliche Untersuchungen und Studien zu verfügen. Zudem sind diese Datensätze aufgrund der weiten Verbreitung auch besser dokumentiert [4] und stetig verbessert worden.

5.6.2 Merkmalsauswahl

Wie im Kapitel 5.6 beschrieben, können die untersuchten Features ganz unterschiedliche Daten beinhalten und diese müssen zuerst mittels Data Preprocessing encodiert werden, bevor ein ML-Algorithmus angewendet werden kann [17].
Eine weitere Herausforderung ist die grosse Anzahl an Features, die analysiert werden müssen und dadurch die Performance des ML-Modells verlangsamen können [18]. Durch dem ML-Modell vorgelagerte Feature Selection Algorithmen soll die Anzahl Features auf ein Minimum reduziert werden. Diese Algorithmen können in folgende Klassen eingeteilt werden [18]:

- Filteralgorithmen analysieren einzelne Merkmale und entscheiden anhand bestimmter Charakteristiken welche Merkmale behalten und welche verworfen werden. Filter funktionieren ohne Data-Mining-Algorithmen [18].

- Wrapper-Algorithmen basieren auf vorgegebenen Data-Mining-Ansätzen und untersuchen Teilmengen von Features auf geeignete Merkmale. Diese Algorithmen generieren die besseren Merkmale als die Filter-Ansätze, brauchen aber entsprechend mehr Rechenleistung und sind daher langsamer [18].

- Es gibt auch Ansätze welche beide Feature-Selection-Techniken kombinieren. Diese Ansätze werden Hybride Methoden genannt. Diese hybriden Ansätze liefern sehr gute Ergebnisse sind aber oft komplex zu realisieren und benötigen viel Rechenressourcen [18].

6 Schlussfolgerungen und Ausblick

Die zunehmende Vernetzung und Komplexität von Computernetzwerken führt zu einer wachsenden Anzahl an Risiken im Bereich Cybersicherheit. Für eine rechtzeitige Identifikation und zur Abwehr von Cyber-Angriffen bieten moderne Machine Learning Algorithmen gute Werkzeuge, um solche Anomalien zu erkennen.
Um ein akkurates Machine Learning Modell zu erstellen, werden aussagekräftige Features benötigt. Diese Features können meist aus einer Kombination aus Daten aus dem Header und Body durch eine Deep Package Inspection gewonnen werden. Gerade in verschlüsselten Nachrichten stösst diese Deep Package Inspection in Network IDS jedoch an ihre Grenzen.
Im Kapitel „Netzwerksicherheitsarchitekturen“ konnte aufgezeigt werden, dass durch eine Orchestrierung zwischen Network IDS und weiteren Sicherheitskomponenten, die Qualität der Features und damit auch die Sicherheit des Netzwerkes erhöht werden kann.
Diese Arbeit konnte aufzeigen, dass maschinelles Lernen sehr effektiv zur Anomalieerkennung im Netzwerkverkehr eingesetzt werden kann, sofern alle Features rechtzeitig ausgewertet werden können. In einem weiterführenden Forschungsprojekt soll im Rahmen unserer Bachelorarbeit festgestellt werden, wie effektiv REST APIs von IoT Devices durch streamorientiertes Machine Learning geschützt werden können.

Literatur

[1] *25 Best Hacking Tools*. 2011. URL: http://techniquesforhackers.blogspot.com/2011/10/25-best-hacking-tools.html (besucht am 01. 12. 2021).

[2] *7th International Conference on Networking, Systems and Security*. 7th NSysS 2020: 7th International Conference on Networking, Systems and Security (Dhaka Bangladesh). New York, NY, USA: ACM, 2020. ISBN: 9781450389051. DOI: 10.1145/3428363.

[3] David Adams. *Snort Alerts*. URL: https://linuxhint.com/snort_alerts/.

[4] Zeeshan Ahmad u. a. "Network intrusion detection system: A systematic study of machine learning and deep learning approaches". In: *Transactions on Emerging Telecommunications Technologies* 32.1 (2021). ISSN: 2161-3915. DOI: 10.1002/ett.4150.

[5] Amazon. *Amazon GuardDuty. Amazon GuardDuty User Guide*. 2021. URL: https://docs.aws.amazon.com/guardduty/latest/ug/guardduty-ug.pdf (besucht am 22. 11. 2021).

[6] Jonathan Andersson, Josiah Hagen und Brandon Niemczyk. "Intrusion prevention system with machine learning model for real-time inspection of network traffic. US201715490609;US201662431700P". H04L9/00;G06N20/00;H04L29/06;H04L29/08. Pat. US11128664 (B1) (US). TREND MICRO INC. 2021.

[7] Bundesamt für Sicherheit in der Informationstechnik. *Einführung von IntrusionDetection-Systemen*. 2002. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/IDS/Grundlagenv10_pdf.pdf?__blob=publicationFile&v=1 (besucht am 22. 11. 2021).

[8] Cisco. "Cisco Encrypted Traffic Analytics White Paper". In: (). URL: https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/enterprise-network-security/nb-09-encrytd-traf-anlytcs-wp-cte-en.html (besucht am 20. 10. 2021).

[9] Cisco. *Secure IPS (NGIPS)*. 2021. URL: https://www.cisco.com/c/en/us/products/security/ngips/index.html#~features (besucht am 22. 11. 2021).

[10] ConSecur GmbH. *Einführung von IntrusionDetection-Systemen. Grundlagen*. Hrsg. von Bundesamt für Sicherheit in der Informationstechnik. 2002.

[11] Hector Cuesta. *Practical data analysis. Transform, model, and visualize your data through hands-on projects, developed in open source tools*. eng. Community experience distilled. Birmingham: Packt Publ, 2013. 335 S. ISBN: 9781783280995.

[12] Ericka Chickowski. *Deep packet inspection explained*. AT&T Business. 2021. URL: https://cybersecurity.att.com/blogs/security-essentials/what-is-deep-packet-inspection (besucht am 05. 10. 2021).

[13] Olga Galinina u. a., Hrsg. *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019. ISBN: 978-3-030-30858-2. DOI: 10.1007/978-3-030-30859-9.

[14] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2019. ISBN: 978-1-492-03264-9.

[15] Xinyu Gong u. a. "Model Uncertainty Based Annotation Error Fixing for Web Attack Detection". In: *Journal of Signal Processing Systems* 93.2-3 (2021). PII: 1494, S. 187–199. ISSN: 1939-8018. DOI: 10.1007/s11265-019-01494-1.

[16] graylog. *Visualize and Correlate IDS Alerts with Open Source Tools*. Hrsg. von graylog. URL: https://www.graylog.org/post/visualize-and-correlate-ids-alerts-with-open-source-tools.

[17] Ayush Gupta und Avani Modak. "Anomaly Detection in HTTP Requests Using Machine Learning". In: *Machine Learning and Information Processing*. Hrsg. von Debabala Swain, Prasant Kumar Pattnaik und Tushar Athawale. Bd. 1311. Advances in Intelligent Systems and Computing. Singapore: Springer Singapore, 2021, S. 445–455. ISBN: 978-981-33-4858-5. DOI: 10.1007/978-981-33-4859-2_44.

[18] Yasir Hamid, M. Sugumaran und V. Balasaraswathi. "IDS Using Machine Learning - Current State of Art and Future Directions". In: *British Journal of Applied Science & Technology* 15.3 (2016), S. 1–22. DOI: 10.9734/BJAST/2016/23668.

[19] Simon S. Haykin. *Neural networks and learning machines*. 3rd ed. New York: Prentice Hall, 2009. xxx, 906. ISBN: 9780131471399.

[20] Holger Schulze. *Insider Threat Report 2020*. Cybersecurity Insiders. 2020. URL: https://www.cybersecurity-insiders.com/wp-content/uploads/2019/11/2020-Insider-Threat-Report-Gurucul.pdf (besucht am 28. 11. 2021).

[21] IBM Corporation. *Introducing IBM Security Network Intrusion Prevention System (IPS) products*. 2012. URL: https://www.ibm.com/docs/en/snips/4.6.0?topic=introducing-security-network-intrusion-prevention-system-ips-products (besucht am 22. 11. 2021).

[22] IEEE Communications Society. *Port-Based Network Access Control. IEEE Std. 802.1X-2010*. 2010. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5409813 (besucht am 24. 11. 2021).

[23] Inga Döbel. *Maschinelles lernen - Kompetenzen, Anwendungen und Forschungsbedarf*. Frauenhofer IAIS. 2018. URL: https://www.bigdata-ai.fraunhofer.de/content/dam/bigdata/de/documents/Publikationen/BMBF_Fraunhofer_ML-Ergebnisbericht_Gesamt.pdf (besucht am 07. 12. 2021).

[24] Gareth James u. a. *An Introduction to Statistical Learning*. Bd. 103. New York, NY: Springer New York, 2013. 441 S. ISBN: 978-1-4614-7137-0. DOI: 10.1007/978-1-4614-7138-7.

[25] Keith Ross James Kurose. *Computernetzwerke. Der Top-Down-Ansatz*. Pearson Deutschland GmbH, 2014. ISBN: 978-8-3-86894-237-8.

[26] Taeho Jo. *Machine Learning Foundations. Supervised, Unsupervised, and Advanced Learning*. 2021. ISBN: 978-3-030-65899-1.

[27] Rudolf Kruse u. a. *Computational intelligence. Eine methodische Einführung in künstliche neuronale Netze, evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*. ger. 2., überarbeitete und erweiterte Auflage. Lehrbuch. Kruse, Rudolf (VerfasserIn) Borgelt, Christian (VerfasserIn) Braune, Christian (VerfasserIn) Klawonn, Frank (VerfasserIn) Moewes, Christian (VerfasserIn) Steinbrecher, Matthias (VerfasserIn). Wiesbaden: Springer Vieweg, 2015. 515 S. ISBN: 978-3-658-10903-5. URL: http://www.computational-intelligence.eu.

[28] Labor für Verteilte Systeme der Hochschule RheinMain. *PVA WS1819-06*. 2021. URL: https://wwwvs.cs.hs-rm.de/vs-wiki/index.php/(PVA_WS1819-06) (besucht am 22. 11. 2021).

[29] Raffael Marty. *Applied Security Visualization*. 2008. ISBN: 978-0-321-51010-5.

[30] McAfee technologies. *McAfee Network Security Platform*. 2021. URL: https://www.mcafee.com/enterprise/en-us/assets/data-sheets/ds-network-security-platform-ns-series.pdf (besucht am 22. 11. 2021).

[31] Mehdi Barati, Azizol Abdullah, Ramlan Mahmod, Norwati Mustapha, Nur Izura Udzir. "FEATURES SELECTION FOR IDS IN ENCRYPTED TRAFFIC USING GENETIC ALGORITHM". In: (2012). URL: http://www.icoci.cms.net.my/proceedings/2013/PDF/PID38.pdf (besucht am 20. 10. 2021).

[32] Michael Meier. *Intrusion Detection effektiv! Modellierung und Analyse von Angriffsmustern ; mit 16 Tabellen und CD ROM*. ger. X.systems.press. Berlin und Heidelberg: Springer, 2007. 209 S. ISBN: 9783540482512. URL: http://swbplus.bsz-bw.de/bsz258398418cov.htm.

[33] Microsoft. *Network Policy Server (NPS)*. 2021. URL: https://docs.microsoft.com/en-us/windows-server/networking/technologies/nps/nps-top (besucht am 24. 11. 2021).

[34] *MILCOM 2007 - IEEE Military Communications Conference*. MILCOM 2007 - IEEE Military Communications Conference (Orlando, FL, USA). IEEE, 2007. ISBN: 978-1-4244-1512-0.

[35] Evgeniya Petrova Nikolova und Veselina Gospodinova Jecheva. *Anomaly Based Intrusion Detection Based on the Junction Tree Algorithm*. 2007.

[36] Arnold Ojugo u. a. "Genetic Algorithm Rule-Based Intrusion Detection System". In: *Journal of Emerging Trends in Computing and Information Sciences* 3 (2012). 08, S. 1182–1194.

[37] D. Raman u. a. "Efficient Machine Learning Model for Intrusion Detection—A Comparative Study". In: *Machine Learning and Information Processing*. Hrsg. von Debabala Swain, Prasant Kumar Pattnaik und Tushar Athawale. Bd. 1311. Advances in Intelligent Systems and Computing. Singapore: Springer Singapore, 2021, S. 435–444. ISBN: 978-981-33-4858-5. DOI: 10.1007/978-981-33-4859-2_43.

[38] Sebastian Raschka und Randal S. Olson. *Python machine learning. Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics*. eng. Community experience distilled. Raschka, Sebastian (VerfasserIn) Olson, Randal S. (VerfasserIn eines Vorworts). Birmingham: Packt Publishing, 2015. 454 S. ISBN: 1783555149. URL: http://proquest.safaribooksonline.com/9781783555130.

[39] Tariq Rashid. *Neuronale Netze selbst programmieren. Ein verständlicher Einstieg mit Python*. ger. 1. Auflage. Rashid, Tariq (VerfasserIn) Langenau, Frank (ÜbersetzerIn). Heidelberg: O'Reilly, 2017. 216 S. ISBN: 978-3-96009-043-4.

[40] SANS Institute. *Host- vs. Network-Based Intrusion Detection Systems*. Hrsg. von GIAC directory of certified professional. 2005.

[41] Moamar Sayed-Mouchaweh. *Learning from Data Streams in Evolving Environments. Methods and Applications*. Springer, 2019. ISBN: 978-3-319-89802-5.

[42] SolarWinds Worldwide. *Intrusion Detection Software*. 2021. URL: https://www.solarwinds.com/security-event-manager/use-cases/intrusion-detection-software (besucht am 22. 11. 2021).

[43] Ahren Studer, Cynthia McLain und Richard Lippmann. "Tuning Intrusion Detection to Work with a Two Encryption Key Version of IPsec". In: *MILCOM 2007 - IEEE Military Communications Conference*. MILCOM 2007 - IEEE Military Communications Conference (Orlando, FL, USA). IEEE, 2007, S. 1–7. ISBN: 978-1-4244-1512-0. DOI: 10.1109/MILCOM.2007.4455095.

[44] Richard S. Sutton und Andrew Barto. *Reinforcement learning. An introduction*. eng. Second edition. Adaptive computation and machine learning. Sutton, Richard S. (VerfasserIn) Barto, Andrew (VerfasserIn). Cambridge, Massachusetts und London, England: The MIT Press, 2018. 526 S. ISBN: 978-0262039246.

[45] Symantec. *Symantec™ Messaging Gateway for Service Providers 10.5. Implementation Guide*. 2013. URL: https://docplayer.net/7235160-Symantec-messaging-gateway-for-service-providers-10-5-implementation-guide.html (besucht am 23. 11. 2021).

[46] Thai H Nguyen, Sajal Bhatia. *Higher Education Social Engineering Attack Scenario, Awareness & Training Model*. Jack Welch College of Business & Technology, School of Computer Science and Engineering, Sacred Heart University. 2020. URL: https://digitalcommons.sacredheart.edu/cgi/viewcontent.cgi?article=1576&context=acadfest#:~:text=63%25%20of%20successful%20attacks%20come,control%2C%20errors%2C%20or%20fraud.&text=60%25%20of%20IT%20professional%20citing,as%20being%20at%20high%20risk.&text=SE%20attempts%20spiked%20more%20than,Q1%20to%20Q2%20of%202018.&text=sensitive%20nature%20of%20the%20information%20which%20could%20been%20compromised. (besucht am 28. 11. 2021).

[47] Ankit Thakkar und Ritika Lohiya. "A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges". In: *Archives of Computational Methods in Engineering* 28.4 (2021). PII: 9496, S. 3211–3243. ISSN: 1134-3060. DOI: 10.1007/s11831-020-09496-0.

[48] The Snort Project. *SNORT Users Manual*. 2020. URL: https://snort-org-site.s3.amazonaws.com/production/document_files/files/000/000/249/original/snort_manual.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAU7AK5ITMGOEV4EFM%2F20211122%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20211122T131958Z&X-Amz-Expires=172800&X-Amz-

SignedHeaders=host&X-Amz-Signature=38a207ad08f6fb99f1dec2652d3a38ab7221a54949a7c9543fbd8f7b3fb3e453.

[49] Trend Micro. *TippingPoint Threat Protection System*. 2021. URL: https://www.trendmicro.com/en_us/business/products/network/intrusion-prevention/tipping-point-threat-protection-system.html# (besucht am 22. 11. 2021).

[50] VICTOR TRUICA. *Understanding the Snort architecture*. 2014. URL: https://truica-victor.com/snort-architecture/ (besucht am 22. 11. 2021).

[51] vmWare. *Netzwerksegmentierung*. 2021. URL: https://www.vmware.com/de/topics/glossary/content/network-segmentation.html (besucht am 23. 11. 2021).

[52] Webland AG. *Managed Web Application Firewalls gegen Angriffe über HTTP*. 2021. URL: https://www.webland.ch/de-ch/Sicherheit/Web-Application-Firewall (besucht am 23. 11. 2021).

[53] Qingju Wang Weizhi Meng. *When Intrusion Detection Meets Blockchain Technology: A Review*. Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kongens Lyngby, Denmark. 2018. URL: https://www.researchgate.net/publication/322814269_When_Intrusion_Detection_Meets_Blockchain_Technology_A_Review (besucht am 28. 11. 2021).

[54] Wikipedia. *IEEE 802.1X*. 2021. URL: https://de.wikipedia.org/wiki/IEEE_802.1X (besucht am 24. 11. 2021).

[55] Lee Winegar. "ENABLING INTRUSION DETECTION IN IPSEC PROTECTED IPV6 NETWORKS THROUGH SECRET-KEY SHARING". In: (2005). URL: https://scholar.afit.edu/cgi/viewcontent.cgi?article=4842&context=etd (besucht am 20. 10. 2021).

Abbildungsverzeichnis

1 Interaktion der IDS-Komponenten 5
2 Vergleich der Analyseansätze 9
3 Snort Output Stream 13
4 Snort IDS Console 14
5 Graylog Dashboard 15
6 Scatter Plot 16
7 Link Graph 17
8 Snort Architektur 19
9 IDS zur Absicherung von Netzübergängen 23
10 IDS zur Überwachung spezifischer Systeme 24
11 IDS zur Überwachung des internen Netzes 25
12 Web Application Firewall 26
13 SMTP-Relay in der DMZ 27
14 IEEE 802.1X Authentifizierung 28
15 Supervised Learning Prozess 31
16 Regression 32
17 Unsupervised Learning Prozess 33
18 Data Clustering 34
19 Dimensionsreduktion 35
20 Angriffserkennung durch Cluster 35
21 Reinforcement Learning Prozess 36
22 Reinforcement Learning Agent System 37
23 Summierungsfunktion mit einer Synapse 39
24 Nichtlineares Modell eines Neurons 40
25 Thresholdfunktion 40
26 Sigmoidfunktion 41