Verbesserte Systemsicherheit durch Kombination von IDS und Firewall

Utz Roedig, Ralf Ackermann, Marc Tresse, Lars Wolf, Ralf Steinmetz
Utz.Roedig, Ralf.Ackermann, Marc.Tresse, Lars.Wolf, Ralf.Steinmetz@KOM.tu-darmstadt.de

Zusammenfassung

Ausgangspunkt jeglicher Aktivität im Bereich IT-Sicherheit ist die Erstellung einer Security Policy. In dieser werden die schutzbedürftigen Objekte und Werte und die gegen sie gerichteten Bedrohungen beschrieben sowie das angestrebte Sicherheitsniveau definiert. Neben herkömmlichen technischen Maßnahmen, wie z.B. dem Einsatz von Firewalls zur Abschirmung von Netzbereichen und Endsystemen sowie von kryptographischen Algorithmen zur Sicherung der Vertraulichkeit und Überprüfung der Unversehrtheit von Daten, werden zur Umsetzung einer solchen Security Policy vermehrt Intrusion Detection Systeme (IDS) eingesetzt. Übereinstimmend wird heute eingeschätzt, daß sich durch deren Verwendung ein höheres Niveau der Systemsicherheit erreichen läßt. 1999 wurden in 37% der Unternehmen, für die Sicherheit ein wichtiges Thema darstellt, IDS-Komponenten benutzt (Vorjahr 29%) [7]. Neben der generellen Verfügbarkeit einzelner zur Erhöhung der Systemsicherheit einsetzbarer Komponenten ist auch die Effizienz ihres Zusammenwirkens ein entscheidendes Kriterium für das erreichbare Sicherheitsniveau. Dieses Zusammenwirken ist – insbesondere auch bei Verwendung relativ neuer Komponenten, wie z.B. der ID Systeme – bisher jedoch teilweise nicht gegeben, bzw. nicht entsprechend optimiert. Innerhalb dieses Beitrags werden wir einen Ansatz vorstellen, der durch die aktive Verknüpfung der Komponenten Firewall und IDS, die Effizienz beider Systeme steigern und zusätzliche Möglichkeiten erschließen kann. Da diese Kopplung spezielle Implikationen auf Design und auszuwählende Mechanismen des zu integrierenden IDS hat, werden wir ein IDS-Modell vorstellen, welches für diesen Zweck optimiert wurde. Der Beitrag umfaßt die Beschreibung einer exemplarischen Implementierung unseres Ansatzes und die Vorstellung und Bewertung erster Einsatzerfahrungen.

1 Basismechanismen der Systemsicherheit

1.1 Ausgangspunkt - Security Policy

Eine Security Policy definiert die Rahmenbedingungen, die erfüllt werden müssen, um ein angestrebtes Sicherheitsniveau zu erreichen. Aus ihr abgeleitete Vorgaben können in formale (wie z.B. Zuständigkeiten, Juristische Absicherung) und technische (z.B. einzusetzende Hard- und Software, Regeln für deren Betrieb) Richtlinien unterschieden werden.

Die technischen Richtlinien und Maßnahmen, auf die wir uns nachfolgend konzentrieren werden, müssen sich neben der Forderung nach prinzipieller Eignung auch an dem Kriterium der

effizienten und performanten praktischen Umsetzbarkeit unter den vorliegenden Randbedingungen messen lassen. Wesentliche Methoden und Verfahren, die im Rahmen verschiedener Komponenten realisiert sind, werden nachfolgend kurz vorgestellt.

1.2 Sicherheitskomponente - Firewall System

Mit Hilfe der Komponente Firewall können Policies durchgesetzt werden, die auf der Reglementierung des Datenverkehrs zwischen Netzbereichen bzw. internen und externen Netzen basieren. So können Firewalls z.B. die Aufgabe haben, ein privates Netz vor unerlaubten Zugriffen aus einem externen Netz zu schützen oder aber auch unerwünschte Zugriffe der Anwender innerhalb des privaten Netzes auf Dienste des externen Netzes zu unterbinden.

Firewalls, die jeweils an einem definierten Punkt, den alle durch sie zu behandelnden Datenströme durchlaufen müssen, zum Einsatz kommen, können in der Regel nicht alle Elemente einer Security Policy innerhalb einer Organisation umsetzen.

1.3 Sicherheitskomponente - Intrusion Detection System

Ein Intrusion Detection System (nachfolgend auch ID System oder IDS) ist in der Lage, komplexe Zustände und Abläufe innerhalb einer IT-Umgebung zu bewerten und entweder als regulär oder als sicherheitskritisch zu klassifizieren. Dies gilt insbesondere auch für einzelne, autonom betrachtet legitime Aktionen, die im Zusammenhang als potentielle Angriffe bewertet werden müssen.

Grundlage der Arbeit eines jeden ID Systems ist die Verfügbarkeit geeigneter zu bewertender Daten, um ein umfassendes Bild über die Aktionen der im IT-System agierenden Prozesse und Nutzer erlangen zu können. Die Art der Informationsbeschaffung, -aufbereitung, -bewertung und der ausgelösten Aktionen ist je nach Ausprägung des IDS überaus vielgestaltig [8]. Die Möglichkeit zur geeigneten Kombination von möglichst vielen sicherheitskritische Abläufe überdeckenden Mechanismen ist ein wesentliches Kriterium für die Bewertung der Leistungsfähigkeit eines Gesamtsystems.

Der Aspekt der effektiven und flexiblen Informationsbereitstellung und die Möglichkeit zur unmittelbaren Auslösung von Aktivitäten bei Entdeckung von Angriffen werden für unsere weitere Betrachtung eine besonders wichtige Rolle spielen.

2 Kombination von Firewall und Intrusion Detection

2.1 Motivation

Grundsätzlich durchlaufen die gesamten an der Kommunikation mit externen Objekten beteiligten Datenströme die Firewall Systeme. Diese bieten daher einen sehr gut geeigneten Punkt der Analyse sicherheitsrelevanter Abläufe, wobei wahlweise sowohl eine Betrachtung auf Netzwerk- als auch auf Applikationsebene erfolgen kann.

Eine solche Vollständigkeit der Sicht auf die gesamte externe Kommunikation ist mit anderen Ansätzen – wie z.B. mittels der Auswertung der syslog-Meldungen aller im lokalen Netz vorhandenen Hosts oder der Analyse von durch im Promiscous Mode arbeitenden Probes gewonnenen Paket-Rohdaten – nur bedingt effizient möglich. Die Klassifikatoren der Paketfilter sowie die in den Proxies realisierten Protokoll-Maschinen einer Firewall können zusätzlich eine Vorselektion der an die IDS weiterzuleitenden Daten übernehmen. Damit können IDS-Auswertemechanismen entweder gezielt angesprochen oder diese von einem unnötig großen zu verarbeitenden Datenvolumen entlastet werden.

Werden durch ein Intrusion Detection System Angriffe festgestellt, so ist es oft wünschenswert, auf diese unmittelbar durch Auslösung von über eine reine Benachrichtigung hinausgehenden Aktionen zu reagieren. Die aktive Kopplung von Firewall und IDS inklusive der Realisierung von Rückkopplungsmechanismen ermöglicht die Umsetzung eines Regelkreises. Innerhalb dessen können die Ergebnisse der Auswertung durch das IDS unmittelbar die beteiligten Firewalls beeinflussen und z.B. als Bedrohung klassifizierte Datenströme durch diese unterbinden.

Nicht zuletzt erlaubt eine aktive und rückgekoppelte Integration von Firewall und IDS eine adaptive Arbeit des Gesamtsystems, so können z.B. Art und Umfang der von der Firewall weitergeleiteten zu analysierenden Daten entsprechend der aktuellen Situation angepaßt werden.

Nach einer Einordnung in das Umfeld verwandter Ansätze werden in den nachfolgenden Abschnitten Modell und prototypische Umsetzung einer entsprechend dieser Überlegungen entworfenen Firewall-IDS-Kombination vorgestellt.

2.2 Arbeiten im Umfeld

Es existieren bereits verschiedene Ansätze, die versuchen, ID Systeme stärker mit Firewall Systemen interagieren zu lassen. Diese können in die folgenden hier beschriebenen Kategorien eingeordnet werden.

2.2.1 ID Systeme mit Firewall-Logfile Auswertung

Verschiedene kommerzielle ID Systeme haben die Möglichkeit, die von Firewalls erzeugten Log-Daten als eine Quelle für ihre Datenbasis zu nutzen. Zu diesen System gehören Axents ITA [4] und das Computer Misuse Detection System von SAIC, die in der Lage sind, die Log-Daten verschiedener Firewall-Typen in die Auswertung mit einzubeziehen. Nachteil dieser Systeme ist, das nur text-basierende Informationen an das ID System übergeben werden können, weiterhin fehlt die Möglichkeit zur rückwirkenden Interaktion mit der Firewall. Wird ein Angriff erkannt, kann das System eine Meldung generieren, es ist aber ohne weitere Vorkehrungen in der Regel nicht in der Lage, unmittelbar Schutzmaßnahmen einzuleiten.

2.2.2 Nutzung der Konfigurationssprachen aktiver Proxies und Filter

Die Konfigurationssprache aktiver Paketfilter und Proxies kann teilweise so eingesetzt werden, daß dadurch das Verhalten eines ID Systems nachgebildet werden kann. Zu den Systemen, die

dies ermöglichen, zählt die Checkpoint Firewall [5] mit ihrer Sprache INSPECT. Sprachsyntax und Konfigurations- sowie Interaktionsmöglichkeiten der Paketfilter ermöglichen es, eingeschränkt Pattern Matching Engines nachzubilden, die Nachbildung einer Statistical Anomaly Detection ist jedoch bei den uns bekannten Systemen nicht möglich.

2.2.3 Begrenzte Interaktion von Firewall und IDS

Bei der Kombination Checkpoint Firewall-1 und RealSecure [3], die eine sehr aktuelle Entwicklung darstellt, wurde ein Firewall Produkt mit einem IDS Produkt gekoppelt. Hierbei wird die Firewall durch das ID System im Bedarfsfall neu konfiguriert, die Firewall selbst wird nicht von dem IDS als Datenquelle genutzt.
Unser Ansatz geht über diesen insbesondere wegen seiner expliziten Ausrichtung auf die Realsierung eines Regelkreises und wegen der offenen, nicht an ein spezielles Produkt oder einen speziellen IDS Mechanismus gebundenen Umsetzung hinaus.

3 Modellierung

3.1 Modellierung

Für die Realisierung eines ID Systems gibt es in der Literatur verschiedene Vorschläge [8]. Um ein für seine Zwecke spezialisiertes IDS-Modell zu finden oder zu erstellen, müssen zunächst ein Funktionsprinzip ausgewählt und nachfolgend verschiedene grundlegende Mechanismen festgelegt werden.

3.1.1 Funktionsprinzip

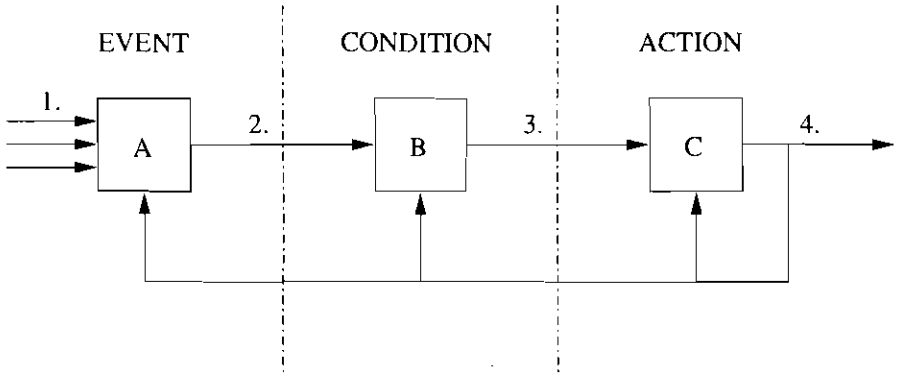
Es gibt eine Reihe von Möglichkeiten, die Funktionsweise eines ID Systems festzulegen, es existiert jedoch ein generelles Funktionsprinzip, welches 1987 in [2] eingeführt wurde. Nach diesem generellen Ansatz können die Bestandteile von ID Systemen stets den folgenden drei Hauptkomponenten zugeordnet werden:

- Event Generator – dieser bereitet die im System anfallenden Daten auf
- Detection Engine – diese realisiert die Analyse der aufbereiteten Daten
- Activity Profile – dieses hält Informationen über den Sicherheits-Zustand des Systems

Da diese Abstraktion nicht auf die Interaktion mit anderen Komponenten eingeht, wird von uns für die hier vorgesehene Firewall-IDS Interaktion das ECA- (Event-Condition-Action) Prinzip mit Rückkopplung, welches aus dem Bereich der aktiven Datenbanken [1] stammt, verwendet. Das besondere Merkmal der Umsetzung dieses Prinzips für unseren Zweck liegt in der Realisierung eines Regelkreislaufs, der durch die Rückkopplungen möglich wird. Die drei Komponenten des ECA-Modells sind folgendermaßen definiert:

- Event – Relevantes Ereignis, das entweder periodisch oder einmalig auftreten kann
- Condition – Definition der Datenprüfung
- Action – Reaktion des Systems auf das Ergebnis der Datenprüfung

Abbildung 1 illustriert das Funktionsprinzip unseres Ansatzes. Zunächst treten Kombinationen der verschiedenen Events ein (1). Diese Informationen repräsentieren gerade ablaufende Systemaktivitäten. Aus der Gesamtheit der Events werden durch die sogenannte Event Mask (A) nur die für den Betrachter interessanten Events herausgefiltert. Durch diesen Filter hat man



1. Eintreten von Ereignissen (Event)
2. gefilterte relevante Ereignisse (Event/Condition)
3. Ergebnis der Datenprüfung (Condition)
4. Folgeaktion (Action)
A. Event Mask (Event/Condition)
B. Datenprüfung (Condition)
C. Vordefinierte mögliche Folgeaktionen (Action)

Bild 1: Anwendung des ECA-Prinzips im IDS-Umfeld

die Möglichkeit, die anfallende Informationsmenge auf ein geeignetes Maß zu reduzieren und gleichzeitig auf die Qualität der für die nachfolgenden Schritte zur Verfügung stehenden Informationen einzuwirken.

Die so gefilterten Events (2) werden in einer weiteren Komponente (B) einer Prüfung unterzogen, die entscheidet, ob potentiell verdächtige Systemaktivitäten vorliegen. Dies geschieht durch einen Regelsatz, der auf die Ereignisse angewendet wird. Das resultierende Ergebnis (3) wird in der nächsten Verarbeitungsstufe (C) dazu verwendet, eine entsprechende Systemreaktion (4) auszuwählen und auszulösen. Es besteht zusätzlich die Möglichkeit, ein Feedback zu erzeugen: Reaktionen des Systems können benutzt werden, um als Event zu wirken oder das Verhalten der Komponenten (A), (B) und (C) zu beeinflusssen.

3.1.2 Mechanismen

Die im folgenden beschriebenen Mechanismen sind wesentlich für die Festlegung der Eigenschaften einer IDS [8]. Sie werden von uns unter dem Aspekt ausgeprägt, daß das resultierende ID System für eine enge Zusammenarbeit mit einer Firewall spezialisiert ist.

Detection
Einer IDS können folgende Detection-Verfahren zugrunde liegen:

- Statistical Anomaly Detection:
  Ein IDS, dem dieses Verfahren zugrunde liegt, versucht anhand statistischer Abweichungen im Verhalten der Nutzer oder der Systeme einen Verstoß oder drohenden Verstoß

gegen die Security Policy zu erkennen. Ein solches IDS benötigt Normalwerte für die Zustände bestimmter Elemente, um mit Hilfe von Algorithmen, denen statistische Verfahren zugrunde liegen, eine Abweichung von der festgelegten oder bestimmten Norm zu erkennen. Betrachtete Parameter können zum Beispiel die verwendete Bandbreite innerhalb eines Subnetzes, oder die Login Dauer eines Benutzers sein.

- Pattern Matching Detection:
  Dieses Verfahren vergleicht die dem IDS zur Verfügung stehenden Daten mit bekannten Angriffsmustern. Dazu ist innerhalb des IDS eine Pattern-Matching Engine nötig, die auf verschiedene Arten (z.B. Finite State Machine, Neuronales Netz, Expertensystem) implementiert werden kann.

Die Anwendung einer Methode schließt die Verwendung der anderen nicht aus. Einige Angriffe die von unserem System registriert werden können, lassen sich besser durch die erste, andere besser durch das zweite Verfahren erkennen, daher ist es sinnvoll beide in das Gesamtsystem zu integrieren.

Layer

ID Systeme können ihre Daten, die als Entscheidungsgrundlage dienen, aus verschiedenen Schichten innerhalb des OSI-Schichtenmodells erhalten. Man kann dabei folgende grobe Unterscheidung festlegen:

- System-Level ID
  Die dem ID System zur Verfügung stehenden Daten werden oberhalb des Network Layers gewonnen. Innerhalb dieser Daten finden sich im wesentlichen Informationen, die Aufschluß über das Benutzerverhalten liefern.

- Network-Level ID
  Die dem ID System zur Verfügung stehenden Daten werden durch unmittelbare Analyse der Netzwerkkommunikation gesammelt. Diese Daten enthalten im wesentlichen Informationen über Kommunikationsverhältnisse, die zwischen verschiedenen Rechnern bzw. Netzen bestehen.

Bei Nutzung einer Firewall als Datenquelle für ein IDS ist es möglich, beide Methoden der Datengewinnung einzusetzen. Innerhalb einer Firewall bzw. eines Firewall-Systems finden Paket-Filter (Network Level) und Proxies (System Level) Verwendung. Mit diesen sind Komponenten verfügbar, die direkt und ohne großen zusätzlichen Aufwand diese Daten liefern können.

Prüf-Intervalle

Entscheidend für die Effizienz eines ID Systems ist der Abstand der Prüfintervalle. Diese legen fest, zu welchen Zeitpunkten das IDS Operationen auf seinem Datenbestand durchführt, um Angriffe zu erkennen. Es gibt dabei folgende grundsätzlichen Möglichkeiten:

- Real Time Systems
  Bei Real Time Systemen wird versucht, alle Verstöße gegen die geltende Security Policy zu dem Moment zu erkennen, zu dem sie stattfinden. Nachteil dieses Verfahrens ist sei-

ne Ressourcenintensität. Es ist deshalb nicht immer eine Real Time Prüfung aller Daten möglich.

- Interval based Systems
  Intervall-basierende Systeme greifen in periodischen Abständen auf ihre Datenbasis zu, um einen Angriff erkennen zu können. Bei Wahl eines zu großen Prüfungs-Intervalls ist es allerdings denkbar, daß der Angriff schon durchgeführt wurde, und nur noch im Nachhinein diagnostiziert werden kann.

Für die realisierte Firewall-IDS-Kombination wird ein Intervall-basierendes System verwendet, bei dem die Intervall-Zeiten sich dynamisch an die Situation anpassen können. Bei auffälligem Verhalten kann die Intervall-Zeit innerhalb des ID Systems verkürzt werden. Dadurch wird „bei zunehmender Gefahr" das ID System „aufmerksamer".

4 Prototypische Implementierung

Als Grundlage der Implementierung wurde das in [6] und [9] beschriebene und als Verteilte Multimedia Firewall Architektur (VDMFA) bezeichnete und in Abbildung 2 gezeigte System verwendet. Die Funktionsweise der Firewall-Komponente dieses modularen und erweiterbaren Systems wird im nächsten Abschnitt kurz dargestellt, eine weitergehende Beschreibung findet sich in [6][9].

4.1 VDMFA und Integration des Intrusion Detection Systems

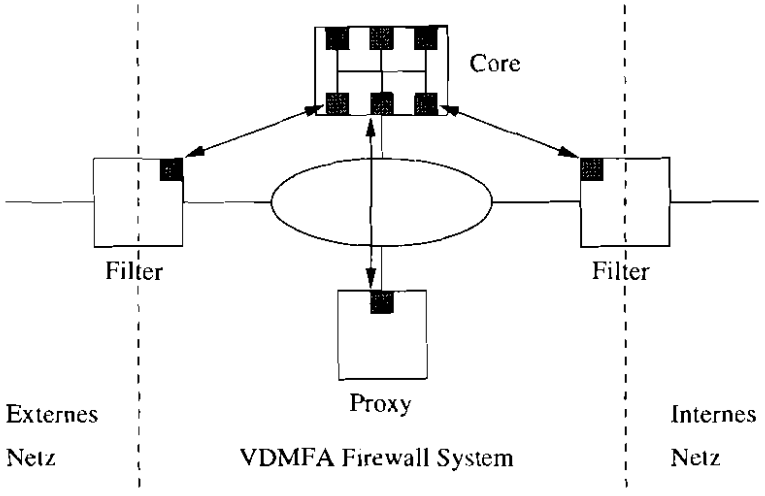


Bild 2: VDMFA Firewall

Die VDMFA Firewall besteht aus mehreren Komponenten, einer zentralen Steuerungs-Komponente (Core) sowie einer Vielzahl angeschlossener Systeme (z.B. Paket-Filter, Proxies). Die Core Komponente kann komponentenübergreifende Informations- und Steuerungsaufgaben durchführen, wie z.B. die Weitergabe von in einem Proxy gewonnenen Informationen zur Änderung der Konfiguration eines Paketfilters.

Einzelne Funktionen, die innerhalb der Core-Komponente implementiert sein müssen, sind auf verschiedene Module verteilt. Diese Module können aufgrund des verwendeten Java-Ansatzes zur Laufzeit des Systems ge- und entladen werden. Die Core-Komponente stellt eine Plattform zur Verfügung, die verschiedene Funktionalitäten tragen kann.

Die Umsetzung des hergeleiteten IDS Modells erfolgt, indem innerhalb der VDMFA Core Komponente geeignete Module hinzugeladen werden. Dabei werden Firewall und Intrusion Detection auf Basis der innerhalb der VDMFA genutzten generischen Kommunikationsmechanismen eng verknüpft, ohne daß die Notwendigkeit zur Realisierung auf einem System besteht.

4.2 Spezielle Aspekte der Umsetzung des Modells

Wie beschrieben, wird das IDS innerhalb des Core in verschiedene Module aufgeteilt. Es ergibt sich die in Abbildung 3 gezeigte Umsetzung entsprechend des ECA-Funktionsprinzips:

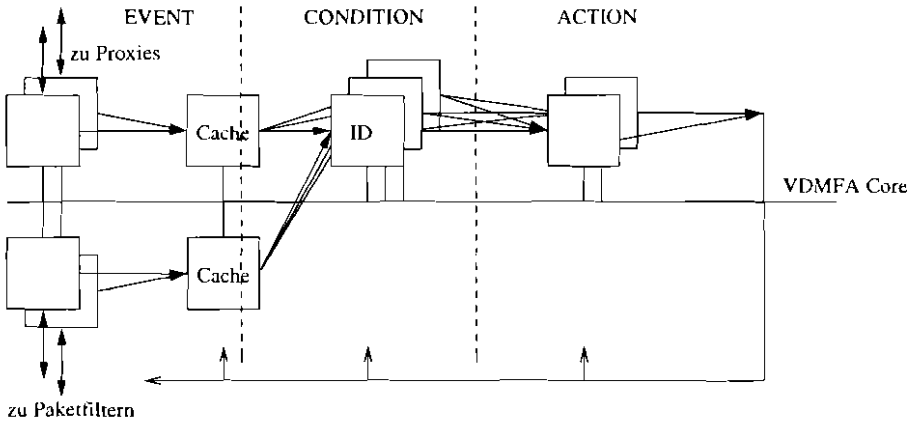


Bild 3: ECA-basiertes IDS unter Nutzung von VDMFA Komponenten

Events werden von den Paketfiltern oder den angeschlossenen Proxies ausgelöst. Die zu diesen Events gehörenden und sie näher charakterisierenden Daten werden nach einer Vorfilterung (Event Mask) innerhalb der zur Zwischenspeicherung benutzten Datenstrukturen (in sogenannten Caching-Modulen) abgelegt. Es werden für die verschiedenen Datentypen spezielle Caching-Module verwendet, die auf die Zwischenspeicherung des entsprechenden Datentyps spezialisiert sind. Mögliche Datentypen sind z.B. Textzeilen (analog zu syslog-Meldungen) oder Byteblöcke, die von den angeschlossenen Paketfiltern eintreffende Netzwerkpakete repräsentieren.

Die Condition Prüfung wird von einer oder mehreren Instanzen eines ID Moduls durchgeführt. Jedes der ID Module kann dabei für eine andere Aufgabe (Angriffstyp) spezialisiert sein.

Alle ID Module verwenden den Datenbestand der Caching Module, um ihre Entscheidungen zu treffen. Entscheidet ein ID Modul, daß eine Aktion ausgeführt werden soll (z.B. weil ein Angriff entdeckt wurde), so kann dies entweder nach Nachladen eines entsprechenden Moduls (z.B. Mail Client zur Benachrichtigung) oder durch Aktivierung der Funktionalität bereits vorhandener Module (z.B. der Paketfilter) erfolgen.

Innerhalb dieses Modells sind nun auch die Parameter des IDS Modells zu finden. Meldungen der System Level ID werden durch die angeschlossenen Proxies in einem Text Format an ein textorientiert arbeitendes Caching-Modul weitergeleitet. Meldungen der Network Level ID werden als Pakete an die paketbasierten Caching Module weitergegeben. Die ID Module prüfen nachfolgend in periodischen Abständen die Inhalte der Caching Module. Treten bestimmte Zustände ein, verkürzen die ID Module das Prüf-Intervall, dadurch wird ein adaptives, intervall-basierendes System realisiert.

Innerhalb der ID Elemente erfolgt die Prüfung mittels Statistical Anomaly und/oder Pattern Matching Detection. Die Detection Engines werden bei Erzeugung der ID Module durch eine entsprechende Sprache parametrisiert.

4.3 ID Modul Aufbau

Die wesentliche Funktionalität der Intrusion Detection wird innerhalb der ID Module realisiert. Das Verhalten der ID Module wird mittels der Firewall Skriptsprache bei Start des Moduls definiert. Um die in 3.1.2 geforderten Mechanismen umsetzen zu können, ist das ID Modul wie in Bild 4 dargestellt aufgebaut:

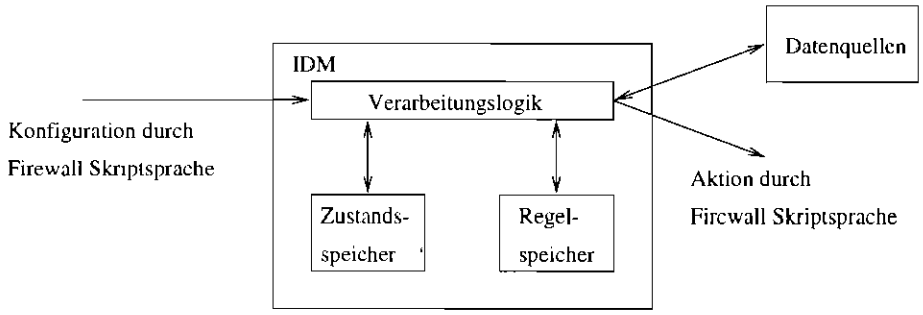


Bild 4: Innerer Aufbau und Kommunikationsbeziehungen des ID Moduls

Jedes ID Modul besitzt eine Verarbeitungslogik, die programmiert durch eine IDS Sprache die Intrusion Detection durchführt. Um ihre Aufgabe zu erfüllen benötigt diese Verarbeitungslogik Schnittstellen zu verschiedenen Elementen innerhalb der Firewall. Die Verarbeitungslogik benötigt Zugriff auf einen internen Regelspeicher (RS), in dem die abzuarbeitenden Konstrukte der IDS Sprache abgelegt sind. Diese Regeln werden von der Verarbeitungslogik periodisch abgearbeitet. Die Verarbeitungslogik greift dabei auf die angeschlossenen Datenquellen (Caching Module) zu und analysiert diese entsprechend den Konstrukten der IDS Sprache. Die geforderte Statistical Anomaly Detection kann hierdurch formuliert werden.
Um die Pattern Matching Detection realisieren zu können, muß die Verarbeitungslogik eine Pattern Matching Engine beinhalten. Diese ist, aufgrund der einfacheren Implementierbarkeit als bspw. ein neuronals Netz, hier als eine Menge parallel arbeitender Finite State Machinen (FSM) implementiert, die einen oder mehrere Zustandswechsel pro Zyklus durchführen. Deshalb ist ein Zustandsspeichers nötig, in dem die aktuellen Zustände der FSMs abgelegt werden können. Zustandswechsel werden durch Konstrukte der IDS Sprache ausgelöst. Die für die Detection-Methoden eventuell notwendigen Norm- bzw. Zwischenwerte werden in speziellen

Caching Modulen abgelegt.
Die IDS Sprache zur Regelformulierung und Parameterisierung bestimmt wesentlich die Funktionsweise und die Möglichkeiten des ID Systems. Deshalb ist ihr Aufbau und ihre Verwendung innerhalb des ID Moduls nachfolgend beschrieben.

4.3.1 Entwurf der IDS Sprache

Wie aus Abschnitt 4.3 hervorgeht, muß ein Benutzer mittels der IDS Sprache in der Lage sein, Intrusion Detection Code zu erstellen, d.h. Angriffserkennungen und Reaktionen darauf zu programmieren. Nachfolgend findet sich eine Liste notwendiger Eigenschaften einer Sprache für das IDS:

- Die Umsetzung des ECA-Modells muß möglich sein. Das bedeutet Bedingungen bzw. Überprüfungskriterien und Folgeaktionen müssen spezifizierbar sein.
- Ausdrücke zur Beschreibung der Abweichung vom Normalverhalten des Systems sind nötig, um die Statistical Anomaly Detection zu realisieren.
- Ausdrücke zur Definition von endlichen Automaten sowie zur Beschreibung von Zustandsübergängen sind nötig, um die Pattern Matching Detection zu realisieren.

Weiterhin ist es notwendig festzulegen, welche Sprachkonstrukte (Sequenz, Auswahl, Iteration) benötigt werden, und ob bzw. welche Datentypen verwendet werden. Auch muß entschieden werden, ob der Quellcode interpretiert oder compiliert wird.
Für das IDS wurde eine interpretative Sprache gewählt, da die Firewall, mit der das ID-System interagieren soll, bereits eine interpretierbare Konfigurationssprache (Firewall Skriptsprache) verwendet. Zusätzlich zum Vorteil der Nutzung bereits vorhandener Softwarekomponenten ergibt sich damit die Möglichkeit, die Firewall Skriptsprache in die IDS Sprache zu integrieren. Dies vereinfacht zum einen den Zugang zu Firewall-Daten, zum anderen die rückwirkende Interaktion mit der Firewall in Form von Befehlen. Innerhalb des Bedingungsteils einer IDS-Regel können bestimmte Firewall-Systemparameter abgefragt werden, während im Aktionsteil z.B. Anweisungen zur Änderung von Firewall-Filterregeln durchgeführt werden können; und zwar unter Verwendung des Firewall-Befehlssatzes. Somit ist die Kopplung von Firewall und IDS auch auf der Sprachebene vorhanden.
Hinsichtlich der notwendigen Sprachkonstrukte zeigt sich, daß nur Bedingungen und Zustände wirklich benötigt werden. Sie dienen der Umsetzung des Condition/Action-Prinzips und sind das Bindeglied zwischen Datenprüfung und Aktionsauslösung. Aufgrund des verwendeten Polling-Ansatzes, bei dem die im Regelspeicher stehenden IDS-Regeln periodisch abgearbeitet werden, werden keine Schleifenkonstrukte benötigt. Da alle vorliegenden Regeln sequentiell abgearbeitet werden, benötigt man keine speziellen Anweisungsblöcke. Es werden keine speziellen Datentypen benötigt, da alle zu speichernden Informationen in den Caches gespeichert werden und der aktuelle IDS-Zustand in den Endlichen Automaten festgehalten wird.

4.3.2 IDS Sprach Aufbau

Die von uns gewählte IDS Sprache besitzt die folgende Syntax (in BNF-Form):

```
rule            -> FSM_number 'if' condition '->' action
FSM_number      -> integer
condition       -> 'Compare' value operator value
                   | 'CacheAnalyzer' string parameter
action          -> 'if' condition '->' action
                   | shell_cmd ';' shell_cmd
                   | shell_cmd
parameter       -> value parameter |
value           -> shell_cmd | integer | string
operator        -> '<' | '<=' | '=' | '>=' | '>'
```

In der hier angegebenen BNF-Form sind die lexikalischen Definitionen für integer, string und shell_cmd aus Gründen der Übersichtlichkeit nicht angegeben.
Zu Beginn jeder IDS-Regel steht die Nummer der FSM, für den diese gültig ist. Eine 0 bedeutet, daß es sich um eine zustandslose Regel handelt, die keine FSM benötigt. Das folgende Konstrukt realisiert zum Beispiel die im ECA Modell definierte Prüfung einer Bedingung mit dem anschließendem Auslösen einer Aktion:

```
0 if CONDITION -> ACTION
```

Der CONDITION Ausdruck enthält dabei den Namen einer JAVA-Klasse, gefolgt von den von ihr benötigten Parametern. Als Parameter können auch in der Firewall Skriptsprache formulierte Befehle verwendet werden, wenn sie einen Integer- oder Stringwert zurückliefern. Die dort verwendeten Klassen sind auf ihre Aufgabe spezialisiert, bei Bedarf können neue Analyse Klassen entwickelt werden, die neue Bedingungen prüfen. Auf diese Weise läßt sich der Sprachumfang flexibel erweitern. In der oben dargestellten BNF sind die beiden bisher verfügbaren Klassen angegeben. Compare vergleicht zwei Werte miteinander, CacheAnalyzer kann verschiedene Prüfoperationen auf den Daten der Cache Module ausführen. Wenn die Bearbeitung der Bedingung 'WAHR' ergeben hat, wird der ACTION Ausdruck bearbeitet.
Der Ausdruck ACTION besteht dabei entweder aus einem weiteren CONDITION/ACTION Paar oder aus einem in der Firewall Skriptsprache formulierten Befehl. Dabei besteht die Möglichkeit, über die xy-Notation einzelne Rückgabeparameter der Analyse Klasse zu referenzieren, wobei x die Nummer der gewünschten Bedingung angibt und y das benötigte Element.

```
0 if A -> if B -> log $2$1
```

würde das Ergebnis der Überprüfung der zweiten Bedingung als Log-Meldung ausgeben. Der Vorteil des vorgestellten Sprachaufbaus ist, daß sich aufgrund der Integration der Firewall Skriptsprache und aufgrund seiner Erweiterbarkeit sehr viele Angriffe mit IDS-Regeln beschreiben lassen. Falls die Mächtigkeit der Sprache nicht ausreicht, muß eine neue Analyseklasse erstellt werden oder ein Firewall-Modul um ein neues Skriptsprachen-Kommando erweitert werden.

4.4 Einsatzbeispiel

Bei ersten Versuchen wurde nachgewiesen, daß die geschaffene Firewall-IDS Kombination in der Lage ist, Angriffe abzuwehren, die eine Firewall allein nicht erkannt hätte, und ein ID System allein nicht hätte verhindern können. Im Folgenden ist ein Beispiel eines solchen Angriffs und seiner Abwehr gegeben:

- Von einem externen Rechner wurde mit Hilfe des Scanners NMAP ein Portscan auf Rechner innerhalb eines internen Netzwerkes durchgeführt.
- Durch den Scan werden in sehr schneller Folge TCP-Pakete mit gesetztem SYN-Bit erzeugt und an die Scan-Ziele gesendet. Da diese Pakete bei normalem Netzwerkverkehr ebenfalls verwendet werden, muß die Firewall (in diesem Fall eine Paketfilter-Firewall) diese passieren lassen.
- Die IDS erkannte die ungewöhnliche Häufung dieses Pakettyps (Statistical Anomaly Detection), und eine nachfolgende Prüfung des Datenbestandes (Pattern Matching Detection) zeigte, daß diese Pakete mehrheitlich von einem Rechner gesendet wurden. Das IDS erkannte daraufhin einen Port Scan.
- Der angreifende Rechner wurde durch eine Konfigurationsmeldung an die Firewall vom angegriffenen Netzsegment isoliert. Dadurch wurden weitere Aktionen des Angreifers (wie z.B. Login-Versuche bei gefundenen Diensten) unmittelbar unterbunden.

5 Bewertung

Innerhalb dieses Beitrags wurde ein Ansatz vorstellt, der durch die aktive Verknüpfung der Komponenten Firewall und IDS die Effizienz beider Systeme steigert. Es wurde ein IDS Modell sowie dessen prototypische Umsetzung für eine Firewall-IDS Kombination vorgestellt. Erste Versuche mit dem geschaffenen System haben gezeigt, daß nicht nur theoretisch sondern auch praktisch ein Vorteil durch die Kombination der beiden Systeme Systeme entsteht.

Durch die Kombination der beiden Systeme kann ein höheres Sicherheitsniveau erreicht werden, als dies durch die Verwendung der beiden Systeme ohne Interaktionsmöglichkeiten möglich ist. Dies läßt sich folgendermaßen begründen:

- Das ID System ist in der Lage, nicht nur Angriffe zu erkennen, sondern auch aktiv zu reagieren.
- Die Firewall ist in der Lage, Angriffe zu erkennen, die sie zuvor nicht bemerken konnte.

Das Gesamtsystem ist damit in der Lage, sich gegen mehr Angriffe zu behaupten, als es die einzelnen Komponenten für sich selbst genommen können. Die Relevanz des beschriebenen Systems ergibt sich aus den festgestellten Defiziten bereits existierender Produkte oder Produktkombinationen ähnlicher Art. Der hier vorgestellte Prototyp ist sicher nur ein Anfang, er zeigt aber die prinzipielle Wirksamkeit dieses Ansatzes.

Literatur

[1] Aktive Datenbanksysteme - Konzepte und Mechanismen
K.R. Dittrich, S. Gatziu, Thomson's Aktuelle Tutorien, Thomson Publishing, 1996.

[2] An intrusion-detection model
Dorothy E. Denning, Procedings of the 1986 IEEE Symposium on Security on Privacy, 1986.

[3] Check Point RealSecure Datasheet
http://www.checkpoint.com/products/firewall-1/realsecureds.html, August 1999.

[4] Intruder Alert
http://www.axent.com/product/smsbu/ITA/default.htm, August 1999.

[5] Stateful Inspection Firewall Technology
http://www.checkpoint.com/products/technology/stateful1.html, August 1999.

[6] VDMFA, eine verteilte dynamische Firewallarchitektur für Multimedia-Dienste
Christoph Rensing, Utz Roedig, Ralf Ackermann, Lars Wolf, Ralf Steinmetz
Kommunikation in Verteilten Systemen, Springer Verlag, 1999.

[7] Gut gerüstet
Information Week, Information Week, page 14, August 1999.

[8] Intrusion Detection - Network Security beyond the Firewall
Terry Escamilla. Wiley Computer Publishing, 1998.

[9] DDFA Concept Utz Roedig, Ralf Ackermann, Christoph Rensing
Technical Report KOM-TR-1999-04, KOM, 1999.