Ein Intrusion–Warning–System für dynamische Koalitionsumgebungen[1]

Martin Lies, Marko Jahnke,
Sven Henkel, Michael Bussmann
Forschungsgellschaft für Angewandte
Naturwissenschaften e.V. (FGAN)
Neuenahrer Strasse 20
53347 Wachtberg-Werthoven
{lies|jahnke|henkel|bus}@fgan.de

Jens Tölle
Universität Bonn
Institut für Informatik, Abt. IV
Römerstr. 164, 53117 D-Bonn
toelle@cs.uni-bonn.de

Zusammenfassung

Wir stellen ein Intrusion-Warning-System (IWS) vor, mit dem der Meldungsfluss von Sicherheitswerkzeugen unterschiedlicher Domänen zusammengefasst und ausgewertet werden kann. Hiermit können sicherheitskritische Situationen frühzeitig erkannt und das Sicherheitspersonal in die Lage versetzt werden, angemessen zu reagieren. Im Gegensatz zu anderen Ansätzen für kooperierende Intrusion-Detection-Systeme (IDS) wird eine modifizierte sternförmige Meta-IDS–Architektur für die Verteilung von Ereignismeldungen bei Angriffen verwendet. Der vorgestellte Ansatz benötigt keine zusätzlichen Informationen über die Art der lokalen Sicherheitssysteme oder die lokalen Sicherheitsrichtlinien. Das vorgeschlagene Meta-IDS ist besonders für den Einsatz in dynamischen Koalitionsnetzwerken wie die von kooperierenden Strafverfolgungsbehörden oder von Streitkräften geeignet.

Zum Erbringen der Meta-IDS–Dienste wurde eine existierende verteilte IDS–Architektur um drei wesentliche Funktionen erweitert:
Frühzeitige Anomaliewarnung – ein graphbasierter Anomaliedetektor wird als ein adaptives Frühwarnmodul für großflächige Angriffe, z.B. Internet-Würmer, verwendet. *Informationsfilterung* – Meldungen werden beim Verlassen der lokalen Domäne entsprechend der domänenspezifischen Sicherheitsrichtlinien anonymisiert. *Informationsverdichtung* – zusätzliche Filter zur Datenreduzierung auf der Basis von vordefinierten Abhängigkeitsregeln steigern die Handhabbarkeit des Datenflusses.

[1]Erschienen in: Tagungsband 11. DFN-CERT/PCA-Workshop "Sicherheit in vernetzten Systemen", Februar 2004

1 Einführung

Da viele Netzwerke in Koalitionsumgebungen (*Coalition Environments, CE*) den Datenaustausch von unklassifizierten – oder mit den entsprechenden Sicherheitsmechanismen auch von klassifizierten – Daten über das öffentliche Internet durchführen, können so enorme Kosten für dedizierte Leitungen gespart werden. Auf der anderen Seite sind diese Netzwerke verschiedenen Bedrohungen ausgesetzt. Beispiele für diese Umgebungen sind internationale Kooperationen von Strafverfolgungsbehörden, Allianzen von Wirtschaftsunternehmen oder militärische Koalitionen (z.B. NATO).

Es ist offensichtlich, dass großflächige sicherheitsrelevante Aktivitäten schneller aufgedeckt werden können, wenn Zugriff auf eine größere Menge von angriffsabhängigen Ereignismeldungen aus einer Vielzahl von Quellen möglich ist. Ein allgemein bekanntes Beispiel sind Internet–Würmer, wie etwa “MS-Blast” und sein Nachfolger “Nachi/Welchia”. Das Zusammenführen von verschiedenen Informationsquellen gewinnt noch an Bedeutung, wenn es um das Erkennen von koordinierten Angriffen gegen eine Vielzahl von Zielsystemen geht, sobald hinter jeder einzelnen Aktivität eine Strategie steht.

Ein Hauptziel aus Sicht eines Sicherheitsanalytikers besteht darin, alle verfügbaren Angriffsinformationen zusammenzuführen, um mehr Daten für seine Analysewerkzeuge zu erhalten. Es gibt zwei alternative Architekturansätze, um koalitionsweite Meldungen zu verarbeiten: *verteilte kooperierende IDS* oder ein *zentralisiertes Meta-IDS*, das Daten von lokalen Sicherheitsprogrammen (IDS, Firewalls, Virenscannern, usw.) entgegennimmt, sammelt und verarbeitet. Bei dem hier beschriebenen *Intrusion–Warning–System* handelt es sich um eine derartige Meta-IDS–Infrastruktur mit einem integrierten Anomalieerkennungsmodul, das zur Frühwarndiagnose und zur Entscheidungsunterstützung eingesetzt werden kann.

Diese Arbeit ist wie folgt strukturiert: Abschnitt 2 betrachtet die wesentlichen Fragen für die Konstruktion eines Meta-IDS für CEs. Abschnitt 3 beschreibt die allgemeine Architektur eines solchen Systems. Anschliessend werden in den Abschnitten 4, 5 und 6 die neuen Komponenten beschrieben, die für die Lösung der vorgestellten Fragen notwendig sind. In Abschnitt 7 werden das Konzept und die bisher erzielten Ergebnisse diskutiert. Abschnitt 8 stellt einige verwandte Arbeiten vor. Zum Abschluss werden in Abschnitt 9 unsere Ansätze zusammengefasst und mögliche Erweiterungen skizziert.

2 Probleme und Lösungsansätze für ein CE-Meta-IDS

Bevor ein robustes und effektives IDS für CE-Szenarien entworfen werden kann, müssen verschiedene Fragen beantwortet werden. Einige dieser Fragen wurden in der Literatur bereits identifiziert (siehe Abschnitt 8). Für unsere speziellen Anforderungen ergeben sich folgende Punkte:

- *Effiziente Analyse-Algorithmen*
  Um einen Nutzen aus den gesammelten Informationen zu ziehen, werden an erster Stelle effiziente Algorithmen benötigt, die dabei helfen, großflächige Aktivitäten zu erkennen. Auf der einen Seite benötigen wir Korrelationsverfahren, um koordinierte Aktivitäten zu

erkennen. Andererseits sind Anomaliedetektoren als “Frühwarnsystem” notwendig, um das lokale Sicherheitspersonal in den einzelnen Domänen mit den notwendigen Informationen zu versorgen.

- *Richtlinien für die Informationsweitergabe*
  Eine der wichtigsten zu lösenden Fragen, um den koalitionsweiten Austausch von Angriffsinformationen zu realisieren, sind die unterschiedlichen Richtlinien für die Informationsweitergabe (*Information Sharing Policies, IShPs*), da Informationen in den IDS-Meldungen enthalten sein können, die nicht einmal an die engsten Koalitionspartner gelangen sollen. Für die Lösung dieses Problems werden Mechanismen benötigt, um Angriffsinformationen zu filtern.

- *Sicherheitsrichtlinien*
  Die vorgegebenen Sicherheitsrichtlinien, die von den lokalen IDS durchgesetzt werden sollen, können sich in verschiedenerlei Hinsicht unterscheiden. Beispielsweise können in einer Domäne Portscans als potentieller Angriff aufgefasst werden, während in einer anderen Domäne eine solche Aktivität völlig ignoriert wird. Hieraus erhält man einen heterogenen Pool von Ereignismeldungen mit verschiedenen Prioritäten, Granularitäten, Vertrauensverhältnissen, etc. Um diesen Pool bewältigen zu können, kann das System entweder Mechanismen zur Ereignismeldungs-“Normierung” über die eingehenden Daten verwenden oder Analysemethoden benutzen, die unabhängig von den hier aufgeführten Eigenschaften sind.

- *Architektur*
  Es gibt unterschiedliche Wege, um eine Architektur für kooperative IDS zu realisieren. In früheren Ansätzen, wie z.B. von Frincke et al. [4] beschrieben, wurde der Ausdruck “Kooperatives IDS” als Einsatz direkter Kommunikationsverbindungen zwischen den IDS-Knoten (oder allgemeiner: lokale Meldungssammeleinrichtungen) der betroffenen Domänen interpretiert. Für n Domänen würde man so ein Netzwerk mit $n \cdot (n-1)$ Verbindungen benötigen. Für dynamische CEs hat dieses Modell nachfolgend aufgeführte Nachteile und ist somit nicht anwendbar:

  – *Unnötige Verkehrslast*
    In extremen Konstellationen – wie etwa einer Maschenstruktur von IDS-Knoten, in der alle verfügbaren Informationen mit allen Teilnehmern geteilt werden – wird viel mehr Netzlast als notwendig verursacht.

  – *Separate Informationsbereinigungsrichtlinien (Information Sanitizing Policies)*
    Da Informationen über Angriffe zu jeder anderen Domäne separat übertragen werden, muss jeweils eine eigene Richtlinie für die Informationsbereinigung zur Umsetzung der IShP erstellt werden, die abhängig von dem bestehenden Vertrauensgrad ist.

  – *Kontinuierliche Rekonfiguration*
    In dynamische CEs können neue Domänen aufgenommen werden und existierende Vertrauensgrade können sich ändern. Dadurch sind kontinuierliche Anpassungen der betroffenen Richtlinien notwendig.

  – *CE-Autoritätsstrukturen*
    Die konventionellen Strukturen (wie z.B. der NATO) sind oft hierarchisch angeord-

net. Dies würden individuelle, miteinander kommunizierende IDS-Knoten konterkarieren.

Auch eine Ringstruktur der IDS-Knoten ist nicht als Alternative zu Peer-to-Peer-Verbindungen denkbar, weil die Informationen zwischen zwei Domänen u. U. viele weitere Domänen durchlaufen müssen, was bzeüglich der IShP problematisch sein dürfte. Als eine realisierbare Struktur hat sich dagegen eine sternförmige Architektur mit bidirektionalen Verbindungen von den einzelnen Domänen zu den zentralen Meldungsverarbeitungs-Einheiten erwiesen.

- *Datenformat und Protokolle*
  Um Ereignismeldungen zu benutzen, die von unterschiedlichen Sicherheitstools verschiedener Hersteller stammen, benötigt man ein vereinbartes Datenformat und Meldungstransportprotokoll. Glücklicherweise hat die *Intrusion Detection Working Group (IDWG)* des IETF hierfür Vorschläge veröffentlicht, wie das XML-basierte IDMEF (*Intrusion Detection Message Exchange Format*, [2]) und das IDXP-Profil [3] für das BEEP-Protokoll [13]. Hersteller kommerzieller Systeme geben an, diese Formate zu unterstützen, um interoperabel mit anderen Systemen zu sein.

Um Ereignismeldungen zuverlässig zustellen zu können, benötigt man entsprechende Kommunikationskanäle. Hierfür sind Mechanismen zur Sicherstellung der Informationsintegrität, Vertraulichkeit und Authentisierung notwendig. Diese Dienste können durch Hinzufügen einer Kryptoebene auf dem Kommunikationskanal, wie etwa SSL/TLS als Teil von IDXP/BEEP [3], realisiert werden.

Obwohl Ziele wie *koordinierte Reaktion auf Angriffe* (Co-ordinated Intrusion Response) sehr wichtig sind, um auf verteilte Angriffe reagieren zu können, werden diese im Rahmen dieses Beitrages nicht behandelt.

3 Die Architektur

Anstelle der üblicherweise für Kooperationszwecke vorgeschlagenen Peer-to-Peer-Architekturen wurde ein anderer Ansatz gewählt. Alle Ereignismeldungen fließen an eine Zentraleinheit für die Datenanalyse. Anschließend werden die gewonnenen Informationen in Form von Warnmeldungen an die lokalen IDS-Knoten wieder verteilt. Diese Architektur wird im folgenden als *Meta-IDS* bezeichnet und basiert auf in früheren Arbeiten entwickelten, generischen IDS-Infrastrukturkomponenten [5]. Sie besteht im wesentlichen aus zentralen Einheiten (*Management-Konsolen*) und Aufschaltpunkten an den Domänengrenzen (*IDS-Gateways, GWs*), dargestellt in Abb. 1.

Management-Konsolen verfügen über Speicher- und Verarbeitungsstufen für Ereignismeldungen sowie über verschiedene grafische Benutzerschnittstellen (GUIs) für die visuelle Offline-Analyse und für die Darstellung eintreffender Meldungen in Echtzeit. Die darin enthaltenen Analysemodule sind Anomaliedetektoren, deren Arbeitsweise in Abschnitt 4 erläutert wird.

Konsolen erhalten Ereignismeldungen von den Gateways, deren Aufgabe es ist, innerhalb ihrer Domäne alle auftretenden Meldungen zu sammeln und von den Konsolen erzeugte Warnungen

für die entsprechenden Stellen zugänglich zu machen. Die Arbeitsweise der Gateways wird in Abschnitt 5 diskutiert.

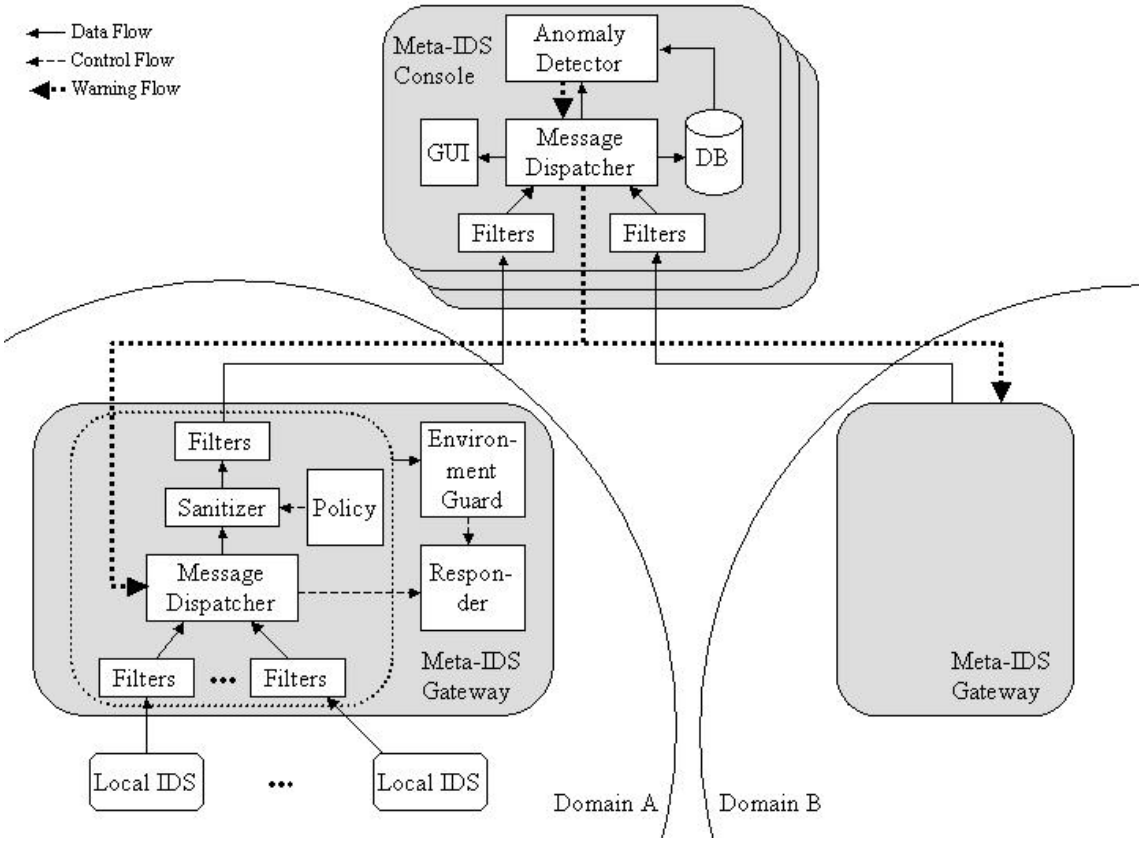

Abbildung 1: Architekturkomponenten des Meta-IDS.

Um einen Schutz gegen Denial-of-Service-Angriffe zu gewährleisten, werden alle in [6] geschilderten Maßnahmen angewendet, so z.B. die Überwachung der Sensor-Agent-Prozessumgebungen sowie die redundante Auslegung der Management-Konsolen für den Fall einer Netzverbindungs- oder Prozessterminierung.

4 Anomalieerkennung im Ereignismeldungs-Datenmodell

Die Struktur des Gesamtsystems erlaubt einen weiteren Ansatz zur Erkennung ungewöhnlicher Vorgänge im System. Die Ereignismeldungen der in den Domänen installierten ID-Systeme gelangen über die Gateways zum Meta-IDS und werden dort zusammengeführt. Diese Zusammenführung von Daten aus unterschiedlichen Quellen ist nun Ausgangsbasis für einen Ansatz zur Beurteilung der “Normalität” des aktuellen Meldungsaufkommens. Somit wird an dieser Stelle ein Ansatz der Anomalieerkennung eingesetzt. Die Details dazu werden in den folgenden Abschnitten erläutert.

4.1 Das Umfeld

Eine der wesentlichen Herausforderungen für eine am Meta-IDS installierte Anomalieerkennungskomponente ist ihr Umfeld. Am Meta-IDS treffen über die Gateways eine große Zahl Ereignismeldungen aus den angeschlossenen Domänen ein. Es ist jedoch auf Grund der Konzeption des Gesamtsystems nur sehr begrenzt möglich, Annahmen über die Art und die Häufigkeit der Meldungen zu machen. Aus leicht ersichtlichen Gründen kann sowohl die Frequenz der eintreffenden Ereignismeldungen, deren Inhalte und deren Aufbau stark von der konkreten Installation abhängen.

Einer der Hauptgründe für diese Eigenschaft ist die dezentrale Administration der Ereignismeldungen generierenden Systeme in den einzelnen Domänen. Es ist das berechtigte Interesse jeder einzelnen Domänenverwaltung Ereignismeldungen nur sehr überlegt an die Kooperationspartner weiterzureichen. Diese Zurückhaltung kann sich deshalb auch in der Anonymisierung einzelner Ereignismeldungen, im Auslassen von Detailinformationen oder in der Filterung bestimmter Ereignismeldungen äußern.

Ebenfalls auf Grund der dezentralen Administration ist zu bedenken, dass in den Domänen unterschiedliche Systeme und Produkte eingesetzt werden können. Dies führt zu uneinheitlichen Ereignismeldungen. Identische Ereignisse können damit durch in Form und Anzahl unterschiedliche Ereignismeldungen weitergegeben werden.

4.2 Das Konzept

Die grundlegende Idee zur Erfassung des aktuellen Systemzustandes und der Erkennung von Abweichungen von diesem Normalzustand (Anomalien) ist die kontinuierliche Überwachung der eintreffenden Ereignismeldungen. Das hier eingesetzte Verfahren ist ursprünglich für die Überwachung und Anomalieerkennung des Verkehrs in Netzwerken entwickelt worden (siehe [14]) und funktioniert folgendermaßen:

Es hat sich gezeigt, dass die typischen Strukturen über die Zeit recht stabil sind, also nur selten grundlegende Veränderungen auftreten. Aus diesem Grund kann man regelmäßig in kurzen Abständen den aktuellen Verkehr sammeln (möglich durch Netzwerk-Monitoring-Geräte oder auch durch Packetsniffer, in geswitchten Netzwerken an einem Spiegelport eines Switches) und in Form einer Verkehrsmatrix abspeichern. Diese Verkehrsmatrix kann als Graph interpretiert werden. Knoten des Graphen sind damit an der Kommunikation beteiligte Geräte während Kanten der Kommunikation zwischen zwei Geräten entspricht. Die Kanten sind dabei mit der Stärke der Kommunikation gewichtet.

Diese Graphen können mittels Graph-Clustering-Algorithmen in Teilgraphen zerlegt werden. Das Clustering dieses Graphen repräsentiert somit die typische Kommunikationsstruktur innerhalb des überwachten Netzes.

Plötzliche Änderungen dieser Strukturen werden als Anomalie aufgefasst und gemeldet. Dazu ist eine Metrik erforderlich, die Ähnlichkeiten von aufeinanderfolgenden Clusterings bewertet.

4.3 Anpassung an Ereignismeldungen

Das im Bereich der Verkehrstrukturen genutzte Verfahren benötigt nun einige Anpassungen an das hier betrachtete Ereignismeldungsmodell. Viele der am Meta-IDS eintreffenden Meldungen können analog den oben betrachteten Verkehrsgraphen unmittelbar für den Aufbau eines Graphen verwendet werden. In dieser Kategorie fallen alle Ereignismeldungen, die detailliert Ziel (das betroffene System) sowie Quelle (vermuteter Auslöser) eines Ereignisses angeben. In Abhängigkeit des Detaillierungsgrades kann es vorkommen, dass Gateways mancher Domänen für manche Ereignismeldungen nur das betroffene System angeben, jedoch keine Angabe über einen (vermuteten) Auslöser machen können oder wollen. Je nach Konfiguration einzelner ID-Systeme und Gateways kann es sein, dass solche Angaben überhaupt nicht vorgesehen sind. Um solche Ereignismeldungen trotzdem in geeigneter Form im Ereignismeldungsgraph zu berücksichtigen, kann den Domänen für bestimmte Ereignismeldungstypen ein Pseudoknoten zugeordnet werden, der über Ereignismeldungen repräsentierende Kanten mit betroffenen Systemen repräsentierende Knoten verbunden wird.

Ein solcher mittels Graph-Clustering-Verfahren aufgeteilte Graph beschreibt somit die typische Struktur der eintreffenden Ereignismeldungen. Abweichungen der typischen Meldungsstruktur sind auch hier wieder als Anomalie anzusehen und zu melden. Dabei können die im Kontext der Anomalieerkennung in Netzwerkverkehrsstrukturen genutzten Vergleichsmaße eingesetzt werden.

4.4 Methoden zur Bewertung

Zur Bewertung des Systems wird auf mehrere Datenquellen zurückgegriffen. Einer der Testdatensätze ist dabei eine Zusammenstellung von über 400 MByte realen Ereignismeldungen, die den Normalbetrieb an einem der Einsatzorte widerspiegelt. Eine weitere Testumgebung zur Bewertung ist ein System, in dem normale Meldungen gezielt und reproduzierbar mit künstlich generierten Ereignismeldungen gemischt werden können. Dieses System kann mit Eingabedaten versorgt werden, die beispielsweise von einem Wurmausbreitungssimulator generiert werden. Dieser Simulator bildet die Ausbreitung eines Wurmes am Beispiel von Ausbreitungsmechanismen realer (z.B. Code Red v2, Code Red II,...) oder fiktiver Würmer nach.

4.5 Herausforderungen

Auch bei dieser Anwendung von Anomalieerkennungsverfahren besteht wieder die bekannte Herausforderung, dass solche Verfahren grundsätzlich anfällig sind für die Erzeugung von Fehlalarmen (sogenannte *False Positives*) oder reale Bedrohungen nicht erkennen (*False Negatives*). Auch die hier angewandten Methoden können diese Problematik nicht grundsätzlich umgehen, sondern es ist durch sorgfältige Abstimmung aller Parameter im konkreten Einsatzszenario darauf zu achten, dass die Anzahl der False Positves und False Negatives akzeptabel bleibt. Grundsätzlich ist anzumerken, dass Verfahren dieser Kategorie nicht ausschließlich verwendet werden sollten und immer nur zusätzliche Informationen zum aktuellen Gesundheitszustand des überwachten Netzes geben können.

5 Richtlinien-konfigurierbare Gateways für Ereignismeldungen

Damit Ereignismeldungen bereinigt (d.h. insbesondere anonymisiert) werden können, sind Gateways (GWs) notwendig. Die Funktionsfähigkeit der GWs wird vom Meta-IDS kontrolliert, jedoch ist die IShP abhängig von der lokalen Domäne. Daher kann die Konfiguration des GWs nur Aufgabe des lokalen Sicherheitspersonals sein. Da die Umsetzung der IShPs zu den wichtigsten Aspekten beim Einsatz von IDS von CEs sind, wird im folgenden die Informationsbereinigung innerhalb des GWs genauer vorgestellt. Dieser Prozess kann als ein Problem der bedingten Musterübereinstimmungsprüfung und -transformation (*Conditional Pattern Matching and Transformation, CPMT*) für Ereignismeldungen aufgefasst werden (zur Notation und zur formalen Definition, siehe [7]).

Aus der lokalen IShP kann eine Menge von bedingten Transformationsregeln der Form $R = (\{E^M\}, \emptyset, E^T)$ mit einem *Matching-Template* E^M und einem *Transformations-Template* E^T generiert werden. Unglücklicherweise kann ein sinnvoller Informationsbereinigungsprozess, wie er in den GWs benötigt wird, nicht ausschliesslich auf einer statischen Textsubstitution basieren. Wenn beispielsweise IP-Adressen als Bestandteil von Meldungen durch feste Werte ersetzt werden sollen, gehen alle Informationen über die Topologie des Netzwerks verloren. Offensichtlich behindert dies den Erkennungsprozess, insbesondere, wenn Verkehrsbezogene Anomalien entdeckt werden sollen. Daher benötigen wir eine flexiblere Methode, um Transformationsregeln zu spezifizieren: *Submatching-Referenzen*. Beispiel hierfür stellt der "Substitute"–Befehl des Standard-Unix-Tools sed(1) dar, der Zeichenketten ersetzt:

```
s/192\.22\.([0-9]{1,3}\.[0-9]{1,3})/191\.72\.\1/.
```

Hierbei wird im Falle eines Matchings der Inhalt des qualifizierten Teilausdruck `([0-9]{1,3}\.[0-9]{1,3})` ermittelt und anstatt der Submatching-Referenz `\1` in der neu zu erzeugenden Zeichenkette verwendet. In diesem Beispiel wird also das 2-Byte Präfix der angegebenen Adresse statisch ersetzt.

Die Vorgehensweise dieser Transformation wird in Abbildung 2 dargestellt.

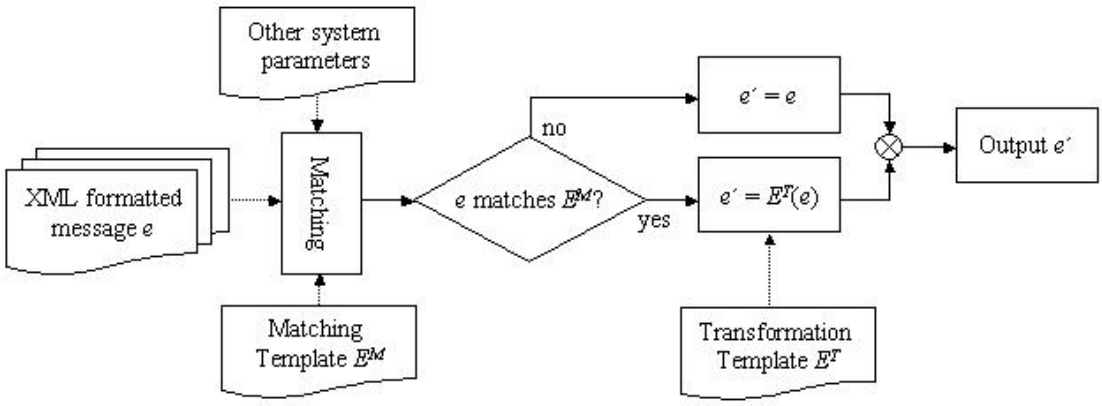

Abbildung 2: Ablauf der bedingten Meldungstransformation.

Um die Möglichkeiten der Transformationsregeln noch erweitern zu können, ist es offensichtlich sinnvoll, Systemparameter – wie den Zeitpunkt des letzten Matchings, die aktuelle Systemzeit oder die IP-Adresse des GWs – als Parametermenge einzubeziehen.

6 Module zur Aggregation von Ereignismeldungen

In unserem Ansatz gibt es zwei Aggregationsverfahren, die den Umgang mit hohen Meldungsraten durchführbar machen: *Redundanzfilter* und das Anwenden von vordefinierten Erkennungsregeln für miteinander korrelierte Elementarereignisse (*Kombinationsdetektor*). Wie auch die im vorangegangenen Abschnitt beschriebenen Verfahren zur Informationsbereinigung können wir die Kombinationserkennung auf ein CPMT-Problem reduzieren.

Sowohl Redundanzfilter als auch Kombinationsdetektoren können als Filtermodule an jeder Stelle unserer Architektur (vgl. Abb. 1) eingefügt werden. Unser Ansatz beinhaltet wohlgemerkt nicht das Generieren von Korrelationsregeln, jedoch sobald diese spezifiziert wurden – sei es manuell oder durch einen entsprechenden Algorithmus – ist man in der Lage, diese Regeln sehr flexibel auf den Meldungsfluss anzuwenden.

6.1 Filterung von Redundanzen

Um die Anzahl von Ereignismeldungen reduzieren zu können, ist es offensichtlich notwendig, Redundanzen zu vermeiden. Deshalb werden mehrere Meldungen mit einem “ähnlichen” Inhalt zu einer einzelnen, neuen Meldung zusammengefasst. Um den Fusionierungsprozess der angesammelten Meldungen zu spezifizieren, erweitern wir die Matching-Templates E^M und definieren dadurch, welche Ereignisse als “ähnlich” angesehen werden können.

Die Ähnlichkeitseigenschaft muss flexibel konfigurierbar sein, da sie in hohem Maße von Domänen-spezifischen Parametern abhängt, wie etwa den verwendeten Sicherheitswerkzeugen und den durchzusetzenden Richtlinien. Da wir nicht nur *absolute* (initiale) *Matchings* benötigen, sondern auch *relative Matchings* (d.h. abhängig von vorhergehenden, wenn die Differenz definiert wurde), müssen wir den Matchingvorgang erweitern. Wir führen ein zweites Matching-Template E_1^M ein, das auf Submatchings des initialen Matchings E_0^M verweist.

Beispiel: Es sollen alle Meldungen (im IDMEF-Format [2]) mit identischer Ereignisklassifikation und Quell-/Ziel–IP-Adresse gefiltert werden. Gleichzeitig soll der Entstehungszeitpunkt der Meldungen innerhalb einer Sekunde liegen.

So muss

- E_0^M einen geeigneten Teilausdruck für den Entstehungszeitpunkt der Meldung und für die Quell-/Ziel–Adresse besitzen,
- E_1^M einen bedingten Ausdruck für die Quell-/Ziel–Adresse haben, der identisch zu den entsprechenden Werten des Ereignisses *e* ist, das mit dem initialen Matching-Templates E_0^M übereinstimmt und

- E_1^M einen bedingten Ausdruck für den Wert des Entstehungszeitpunktes beinhalten, der nicht um mehr als eine Sekunde vom Entstehungszeitpunkt des initialen Matchings entfernt ist.

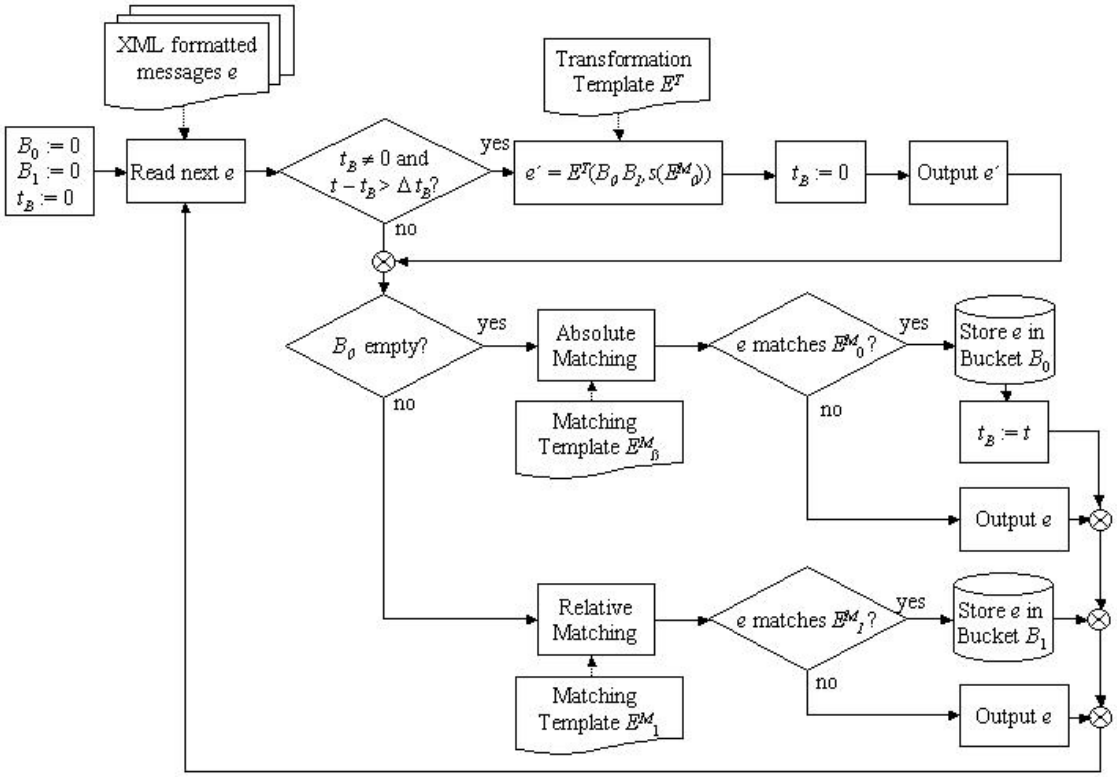

Abbildung 3: Ablauf der Redundanzfilterung durch bedingte Meldungstransformation.

Im Gegensatz zu (zustandslosen) bedingten Transformationen, bei denen ein absolutes Matching für alle Templates E^M erforderlich ist, benötigen wir zwei Speicherplätze (*Buckets*) B_0 und B_1, die bereits übereinstimmende Ereignisse solange aufnehmen, bis eine maximale Speicherdauer Δt_B überschritten wird. Der dazugehörige Algorithmus erzeugt aus einer Folge von Eingabeereignissen e eine Folge von Ausgabeereignissen e', in der "ähnliche" Ereignisse zusammengefasst sind. Das Verfahren wird in [7] formal beschrieben und arbeitet, wie in Abbildung 3 skizziert.

Man beachte, dass dieser einfache Algorithmus nicht voraussetzt, dass "ähnliche" Ereignisse sequentiell und ohne andere dazwischenliegende Meldungen eintreffen. Zusätzlich ist es möglich, mehrere Buckets parallel zu benutzen. Die Anzahl dieser Buckets muss aber begrenzt sein, um Überlastsituationen und DoS-Angriffe zu vermeiden. Weiterhin ist es möglich, anstatt die Transformation der gespeicherten Ereignisse erst nach Beendigung der Zusammenfassungsphase durchzuführen, nach jedem Teil-Matching das spätere Ausgabeereignis e' entsprechend anzupassen. Da in diesem Falle nur ein Bucket für ein Ereignis benötigt wird, reduziert sich der Speicheraufwand erheblich.

6.2 Kombinationsdetektoren

Wir definieren *Ereigniskombinationen* als korrelierte Mengen von Ereignismeldungen. Detektoren für diese Kombinationen wenden in unserem Ansatz vordefinierte Matching-Regeln auf die Ereignisse an. Daher handelt es sich um einen verallgemeinerten Fall der Redundanzfilter, die im vorhergehenden Abschnitt beschrieben wurden. Statt eines absoluten und eines relativen Matchings spezifiziert man eine Menge von relativen Matching-Templates $E^M = \{E_1^M, \ldots, E_n^M\}$, um eine Ereigniskombination beschreiben zu können. Jedes Matching kann sich dabei nicht nur auf Submatchings im initialen Template E_0^M beziehen, sondern auch auf Submatchings in den übrigen Templates. Unter Zuhilfenahme mehrerer Buckets $B_i, 0 \leq i < n$ ist es möglich, auf alle in den Buckets gespeicherten Ereignisse zu verweisen, indem E^T entsprechend erweitert wird.

Mit diesem Ansatz können wir verschiedene Korrelationsbeziehungen modellieren, die für die Erkennung sicherheitsrelevanter Vorkommnisse von Bedeutung sind. So lassen sich beispielsweise folgende Korrelationseigenschaften zwischen zwei Matching-Templates E_0^M und E_1^M ausdrücken:

- *Zusammenhänge der Meldungs-Klassifikation*:
  Definiere Ausdrücke in E_0^M und E_1^M, die eine Teilmenge von möglichen Ereignisklassifikationen (z.B. Aufzählungen oder Bereiche von IDs aus einer Angriffsdatenbank) beschreiben.
- *Zeitliche Zusammenhänge*:
  Definiere einen Teilausdruck in E_0^M, der den Zeitwert beinhaltet, und einen arithmetischen Ausdruck in E_1^M, der den Zusammenhang spezifiziert (z.B. eine maximale Zeitdifferenz der Detektion).
- *Räumliche Zusammenhänge*:
  Definiere einen Teilausdruck in E_0^M, der den Adresswert beinhaltet, und einen arithmetischen Ausdruck in E_1^M, der den Zusammenhang spezifiziert (z.B. Zugehörigkeit zum selben Subnetz).

Gleichzeitig können frei wählbare Kombinationen definiert werden, um so komplexere Zusammenhänge spezifizieren zu können. Ein Beispiel ist “Suche alle Meldungen mit einer ’TCP Portscan’-ähnlichen Klassifikation, die von der selben Adresse innerhalb einer Zeitdauer von einer Stunde kommen”.

7 Bisherige Ergebnisse

Die hier beschriebene Architektur wurde bereits als Prototyp mit dem Namen “MSIDI” (Meta SIDI) implementiert, welcher auf der existierenden verteilten IDS-Infrastruktur “SIDI” (*Survivable Intrusion Detection Infrastructure*, siehe auch [6]) basiert. Einzelne Instanzen von SIDI liefern Ereignismeldungen über zusätzlich entwickelte Gateways an das Meta-IDS, wo alle Meldungen mit den weiter oben beschriebenen Funktionen verarbeitet und analysiert werden.

7.1 Anomalieerkennung

Zur Zeit wird das Anomalieerkennungssystem in die bestehende IDS-Infrastruktur integriert. Dazu finden aktuell Portierungsarbeiten der bisher eingesetzten Java-Lösung in C/C++ statt, um eine reibungslose Zusammenarbeit zu gewährleisten.

In der ersten Erprobungsstufe findet eine Beschränkung auf den ursprünglich für die Netzwerkanomalieerkennung entwickelten Ansatz statt, d.h. es werden zur Erkennung typischer Ereignismeldungsstrukturen ausschließlich Source-Target-Beziehungen für den Aufbau der Graphen verwendet. Die Portierung der bisher verwendeten Algorithmen geschieht aber schon mit Rücksicht auf die eingeplanten Erweiterungen des Verfahrens.

7.2 Verarbeitung von Ereignismeldungen

Die in den Abschnitten 5 und 6 beschriebenen Ansätze der Informationsbereinigung, der Redundanzfilterung und der Erkennung von Ereigniskombinationen können auf die bedingte Ereignismeldungs-Transformation reduziert werden. Ein häufig gewählter Ansatz, um komplexe Transformationen von XML-formatierten Daten durchzuführen, ist die Anwendung von XSLT-Stylesheets [15]. Unglücklicherweise beinhalten diese Empfehlungen keine Regulären Ausdrücke (REs), und derzeit sind auch keine geeigneten Implementierungen verfügbar. Deshalb haben wir für einen ersten Meta-IDS-Prototypen einen XML-Prozessor implementiert, der XSLT-Sheets um POSIX.1-REs sowie um zusätzliche boolesche und arithmetische Ausdrücke erweitert. Er wendet das beschriebene Matching- und Transformationsverfahren auf IDMEF-Meldungen an.

In dem folgenden Beispiel wird ein kombiniertes Matching- und Transformations-Template (sog. *Transformations-Sheet*) spezifiziert:

```
<address>
  192\.22\.([0-9]{1,3})$v1$\.([0-9]{1,3})$v2$
  <condition>
   ($v2<255)
  </condition>
  <transform>
    <xsl:copy>
     191.72.<xsl:value-of select="$v1"/>.<xsl:value-of select="$v2"/>
    </xsl:copy>
  </transform>
</address>
```

Die Matchings der letzten beiden Bytes einer IP-Adresse `192.22.x.y` werden weiterhin als Variablen `$v1` und `$v2` referenziert. Die zusätzliche Bedingung dafür, dass die Transformation durchgeführt wird, ist, dass `$v2` nicht der traditionellen Broadcastsuffix 255 entsprechen darf.

Mit diesen Transformationstechniken haben wir die Informationsbereinigungs-Funktionen der Gateways auf Basis der GNOME libxml2 und libxslt in Form von C++-Filterklassen im Rahmen des SNAF/SIDI-Frameworks (vgl. [5]) implementiert.

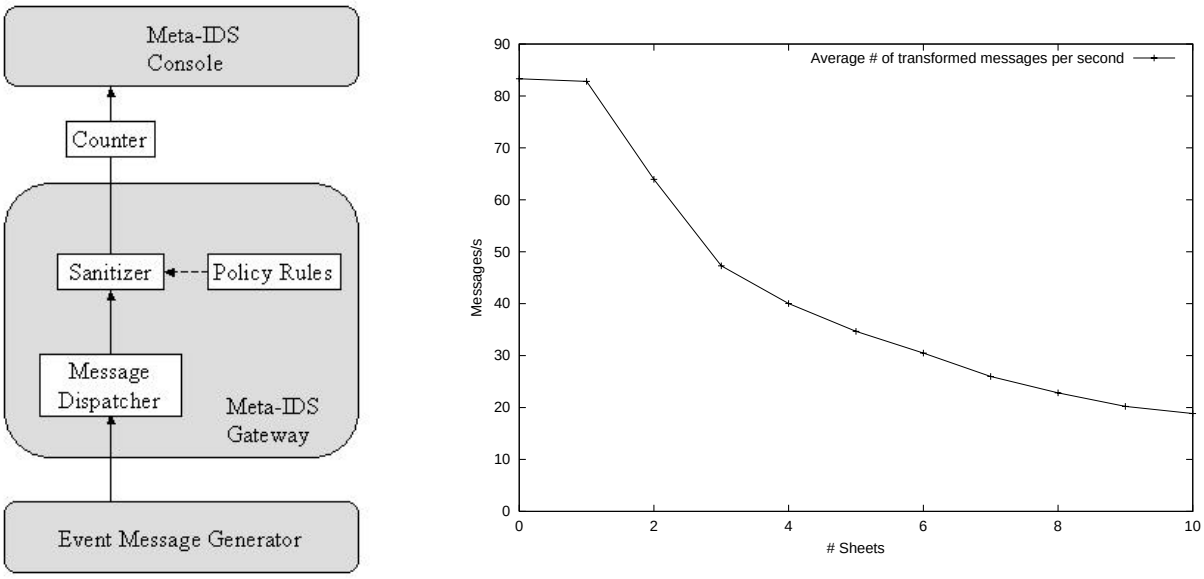

Abbildung 4: Szenario und Ergebnisse der Durchsatzmessungen für die Informationsbereinigung bei Ereignismeldungen.

Um die Effizienz der Implementierung festzustellen, wurde zunächst eine Anzahl von Transformations-Sheets erstellt, die beispielsweise IP-Adressen und Adressbestandteile, Hostnamen, Zeitangaben sowie Produktangaben über Sicherheitswerkzeuge anonymisieren. Dann wurde in Abhängigkeit der Anzahl der Transformationen die Anzahl der Ereignismeldungen pro Zeiteinheit gemessen, die unser Informationsbereinigungs-Gateway zu transformieren in der Lage ist. Unter Verwendung eines PIII/1GHz-PCs mit 256MB RAM und Debian GNU/Linux bewältigt ein Gateway die in Abbildung 4 dargestellten Durchsatzraten (ohne Codeoptimierung). Während dieser Tests bestätigte sich die Vermutung, daß das Zusammenfassen von Eigenschaften mehrerer Templates – sofern anwendbar – zu einer signifikanten Verbesserung der Performanz führt. Dies ist durch die Tatsache begründet, dass Meldungsbestandteile nicht mehrfach auf Übereinstimmungen geprüft werden müssen. Allerdings gestaltet sich eine derartige Zusammenfassung nichttrivial und wird schnell unübersichtlich.

Die Realisierung der Redundanzfilterung wurde bereits – ebenfalls auf der Basis von XSLT-Stylesheets – in Angriff genommen und wird in Kürze in die Gateways integriert werden.

8 Verwandte Arbeiten

Kooperative IDS wurden bereits in zahlreichen Publikationen untersucht.

Der Ansatz von Kim et al. [9] weist einige Analogien zu unserem Architekturansatz auf. Im Gegensatz dazu wird in unserem Ansatz nicht davon ausgegangen, dass ausschließlich die beschriebenen dazugehörigen Architekturkomponenten über die betroffenen Domänen verteilt eingesetzt werden, sondern dass beliebige Sicherheitswerkzeuge als Informationslieferanten herangezogen werden können. Dies kommnt den Anforderungen für dynamische CEs eher

entgegen, da jede einzelne Domäne ihre eigenen Vorstellungen von den zu benutzenden Produkten und deren Herstellern hat.

Eine frühe Arbeit über kooperative IDS von Frincke et al. [4] identifiziert viele Probleme bezüglich der Kooperation, wie etwa Konflikte mit verschiedenen Richtlinien und der Informationsweitergabe. Unter diesen Voraussetzungen und in dem Kontext der Verwendung verschiedener lokaler Werkzeuge in den betroffenen Domänen, kann diese Arbeit als Grundlage für unseren Ansatz angesehen werden. Ein grundlegender Unterschied zu unserer Arbeit besteht in der beschriebenen Architektur, die auf direkten Kommunikationsverbindungen zwischen den verteilten Datensammelstellen basiert.

Die kooperierenden Softwareeinheiten, die in [10] und in [16] beschrieben werden, besitzen ähnliche Konzepte wie unser Meta-IDS-Gateway, wobei abermals die Infrastruktur auf einem Peer-to-Peer–Netz basiert.

Zahlreiche Papiere beschreiben Ansätze, um Korrelationsbeziehungen zwischen Ereignismeldungen finden zu können (z.B. [1], [11], [12]) und sind von mehr oder weniger formalen Modellen abhängig. Da unser System nur Mechanismen zum Erkennen von bereits definierten Zusammenhängen bereitstellt, sind Off-line–Korrelationsverfahren notwendig, um die entsprechenden Regeln zu erstellen. Durch den sehr allgemein gewählten Ansatz für die Spezifikation solcher Regeln, scheint es leicht möglich, extern detektierte Korrelationsbeziehungen zu importieren.

Die Arbeit von Julisch über “Alert Clustering” [8] hatte großen Einfluss auf die Art, wie wir Ähnlichkeiten von Meldungen definieren, um Redundanzen filtern zu können.

9 Zusammenfassung und Ausblick

Dieses Papier präsentiert eine Architektur für ein kooperatives Intrusion-Warning-System für dynamische heterogene Koalitionsumgebungen. Bereits aufgrund der Struktur werden verschiedene Anforderungen für dynamische Koalitionsumgebungen erfüllt, was für andere Ansätze – insbesondere für diejenigen, die von einer direkten Kommunikation Domänenspezifischer Einheiten ausgehen – nicht der Fall ist.

Als Basis für die Realisierung diente eine existierende verteilte IDS-Infrastruktur. Die wichtigsten Erweiterungen stellen zentrale meldungsverarbeitende Einheiten mit integriertem Anomalieerkennungsmodul sowie Module für die Informationsbereinigung und Datenreduktion dar. Damit entsteht die Möglichkeit, frühzeitig Warnungen über großflächige sicherheitsrelevante Aktivitäten zu liefern.

Alle Aufgaben, die die Verarbeitung von Ereignismeldungen betreffen, wurden auf bedingtes Pattern-Matching und Transformation zurückgeführt, das in Form eines erweiterten XSLT-Stylesheet-Prozessors realisiert wurde. Somit ist man in der Lage, den Meldungsfluss äußerst flexibel zu filtern, wobei sich die Performanz-optimierte Umsetzung der Informationsweitergabe-Richtlinien in XSLT-Stylesheets als durchaus nichttrivial erwiesen hat.

Zur Zeit wird an der Vervollständigung der Implementation sowie an diversen Optimierungen des bestehenden Systems gearbeitet. Für die Zukunft sind verschiedene Erweiterungen

des Systems – insbesondere hinsichtlich der Miteinbeziehung weiterer, in Ereignismeldungen enthaltener Informationen in den Anomalieerkennungsprozess – angedacht. Zudem wird an einem kontinuierlich laufenden Testszenario mit realen Ereignismeldungen aus verschiedenen entmilitarisierten Zonen (DMZ) von Forschungsnetzwerken gearbeitet.

Literaturverzeichnis

[1] F. Cuppens and A. Miège. Alert Correlation in a Cooperative Intrusion Detection Framework. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 202–215, Oakland, CA, May 2002. IEEE Computer Society, Technical Committee on Security and Privacy, IEEE Computer Society Press.

[2] D. Curry and H. Debar. Intrusion Detection Message Exchange Format – Data Model and Extensible Markup Language (XML) Document Type Definition. IETF Internet Draft draft-ietf-idwg-idmef-xml-10.txt, January 2003. IETF IDWG.

[3] B. Feinstein, G. Matthews, and J. White. The Intrusion Detection Exchange Protocol. IETF Internet Draft draft-ietf-idwg-beep-idxp-07.txt, October 2002. IETF IDWG.

[4] D. A. Frincke, D. Tobin, J. C. McConnell, J. Marconi, and D. Polla. A Framework for Cooperative Intrusion Detection. In *Proc. 21st NIST-NCSC National Information Systems Security Conference*, pages 361–373, 1998.

[5] M. Jahnke. An Open and Secure Infrastructure for Distributed Intrusion Detection Sensors. In *Proceedings of the NATO Regional Conference on Communication and Information Systems (RCMCIS'02), Zegrze, Poland*, October 2002.

[6] M. Jahnke. Schutz verteilter Intrusion-Detection-Systeme gegen Denial-of-Service-Angriffe. In *10. DFN-CERT/PCA-Workshop "Sicherheit in vernetzten Systemen"*, pages J–1–J–12, Hamburg, February 2003.

[7] M. Jahnke, J. Tölle, M. Bussmann, and S. Henkel. Components for Cooperative Intrusion Detection in Dynamic Coalition Environments. Presented at NATO/RTO IST Symposium "Adaptive Defence in Unclassified Networks", Toulouse, April 2004.

[8] K. Julisch. Mining Alarm Clusters to Improve Alarm Handling Efficiency. In *Proceedings of the NATO/RTO IST Workshop on Inforensics and Incident Response, The Hague, The Netherlands*, October 2002.

[9] Byoung-Koo Kim, Jong-Su Jang, and Tai M. Chung. Design of Network Security Control System for Cooperative Intrusion Detection. *Lecture Notes in Computer Science*, 2344:389–, 2002.

[10] G. Koutepas, F. Stamatelopoulos, V. Hatzigiannakis, and B. Maglaris. An Adaptable Inter-Domain Infrastructure Against DoS Attacks. In *Proc. of the International Conferences on Advances in Infrastructures for e-Business, e-Education, e-Science, e-Medicine on the Internet (SSGRR 2003)*, L'Acquila, Italy, January 2003.

[11] B. Morin, L. Mé, H. Debar, and M. Ducassé. M2D2: A formal data model for IDS alert correlation. volume 2516, page 115, 2002.

[12] P. Ning, Y. Cui, and D. Reeves. Constructing attack scenarios through correlation intrusion alerts. Technical Report TR-2002-12, Department of Computer Science, North Carolina State University, August 14 2002.

[13] M. Rose. RFC 3080: The Blocks Extensible Exchange Protocol Core. http://www.ietf.org/rfc/rfc3080.txt, March 2001.

[14] J. Tölle and C. de Waal. A Simple Traffic Model Using Graph Clustering For Anomaly Detection. Proc. of Applied Simulation and Modelling (ASM) Crete, Greece, June 2002.

[15] W3C. W3C Recommendation 16: XSL Transformations (XSLT) Version 1.0. http://www.w3.org, 1999.

[16] Q. Zhang and R. Janakiraman. Indra: A Distributed Approach to Network Intrusion Detection and Prevention. Technical report, University of Washington, St. Louis, 2003.