Intrusion Detection Management

28. Januar 2003

Kurzbeschreibung

Sicherheitswerkzeuge wie IDS, Scanner und Firewalls gibt es bereits viele. Sie alle haben jedoch gemeinsam, dass verschiedene Programme dieser Art untereinander nicht kompatibel sind und ihr Entwicklungsschwerpunkt technikorientiert ist. Hier setzt das *Intrusion Detection Management* (IDM) an. Zum einen soll es eine einheitliche Schnittstelle für Sicherheitswerkzeuge bieten, von der aus sie zentral verwaltet werden können, zum anderen gilt es eine benutzerorientierte Schnittstelle zur Verfügung zu stellen, die sicherheitsrelevante Vorgänge des zu überwachenden Netzes übersichtlich darstellt und ein effektives Eingreifen ermöglicht.

1 Einleitung

Bereits in den 80er Jahren erstmals entworfen, haben sich *Intrusion Detection Systems* (IDS) zu einem wichtigen Bestandteil der IT-Sicherheit entwickelt. Sie ergänzen bereits vorhandene Sicherheitsstrukturen, wie beispielsweise Firewalls, die nicht selbstständig in der Lage sind, eine Paketanalyse durchzuführen. Sie unterstützen bzw. ersetzen signaturbasierte Maßnahmen, z.B. Virenscanner und liefern, sofern Honigtopfverfahren dem Bereich der IDS zugeordnet werden, wichtige Erkenntnisse über unbekannte Schwachpunkte des Systems und Strategien von Angreifern. Weiterhin ermöglichen IDS das Netz von innen gegen Angriffe abzusichern. Dieser Aspekt spielt insofern eine Rolle, da eine Umfrage unter Sicherheitsexperten ergeben hat, dass rund 40% der Angriffe von authorisierten Benutzern initiiert wurden [larsen99].

Betrachtet man die rapide zunehmende Anzahl von Angriffsversuchen gerade innerhalb der letzten Jahre (siehe Abb. 1), zeigt sich, dass geeignete Maßnahmen getroffen werden sollten, um möglichen Schaden von Systemen abwenden zu können.

Diesen Zahlen steht eine Untersuchung von mittelständischen Unternehmen in Nordrhein-Westfalen gegenüber [cz02]. Hierbei zeigte sich, das 40% der Befragten keine Firewall oder Virenschutz verwenden, in 70% aller Fälle wird keine Datensicherung durch Backups durchgeführt. Ein Angriff auf ein solches Unternehmen könnte schwerwiegende Verluste mit sich bringen. Abbildung 2 zeigt die Ergebnisse bezüglich der unternehmensspezifischen Internet-Sicherheitsrichtlinien. Für diese Einstellung gibt es verschiedene Gründe, die von Bedienbarkeit über Kosten bis hin zur fehlenden Sensibilisierung reichen.

Im Rahmen dieser Arbeit soll der Punkt der Bedienbarkeit herausgegriffen, analysiert und optimiert werden. Denn auch im Bereich der Angriffswerkzeuge hat sich

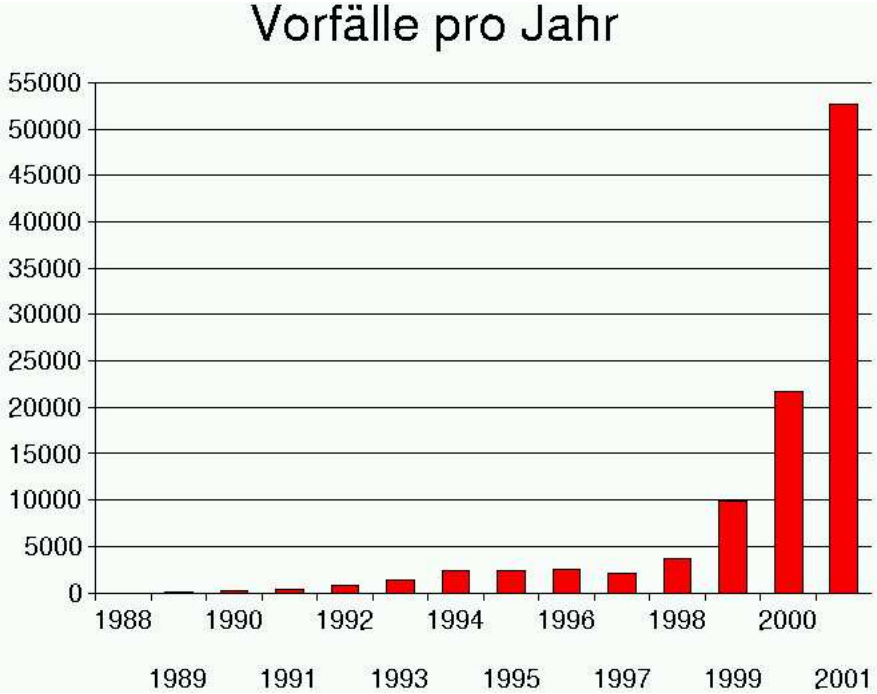

Abbildung 1: Zunahme von Angriffen (Quelle: CERT/CC)

auf diesem Gebiet viel getan. Wie in Abbildung 3 zu sehen ist, nimmt trotz steigender Angriffszahlen das Basiswissen, wie ein Angriff im Detail funktioniert, stetig ab [jallen00]. Wie der Grafik zu entnehmen ist, wurde Mitte der 90er Jahre erstmals eine grafische Benutzeroberfläche für ein Angriffswerkzeug verwendet. Heutzutage ist es ein leichtes, sich einen Virus zusammen zu “klicken” oder einen Trojaner, z.B. NetBus oder Back Orifice, einzuschleusen und zu bedienen.

Im Bereich der Sicherheitswerkzeuge sind grafische Oberflächen ebenfalls seit einigen Jahren im Einsatz, meist wird die *Usibility* bei der Entwicklung der Werkzeuge aber zweitrangig behandelt. Ziel ist es daher eine Schnittstelle zu entwickeln, die bereits vorhandene, durchaus heterogene Quellen und Mechanismen nutzt und dem Anwender hierfür eine einheitliche Verwaltungsmöglichkeit anbietet. Diese Schnittstelle soll softwareergonomischen Aspekten genüge tragen und spezielle menschliche Fähigkeiten, z.B. die der Mustererkennung, mit einbeziehen, so dass ein schnelles erkennen bzw. reagieren auf Angriffe ermöglicht wird.

2 Untersuchung aktueller IDS

Im Vergleich zu anderen Sicherheitswerkzeugen, beispielsweise Firewalls und Virenscanner, werden IDS bisher verhältnismäßig selten eingesetzt. Hierfür gibt es verschiedene Gründe, die neben der Kostenfrage auch die Einstellung der Unternehmen zum Thema IT-Sicherheit widerspiegeln. Einige dieser Gründe sollen hier aufgeführt und analysiert werden, um sie bei der Entwicklung eines *Intrusion Detection Managements* (IDM) zu berücksichtigen.

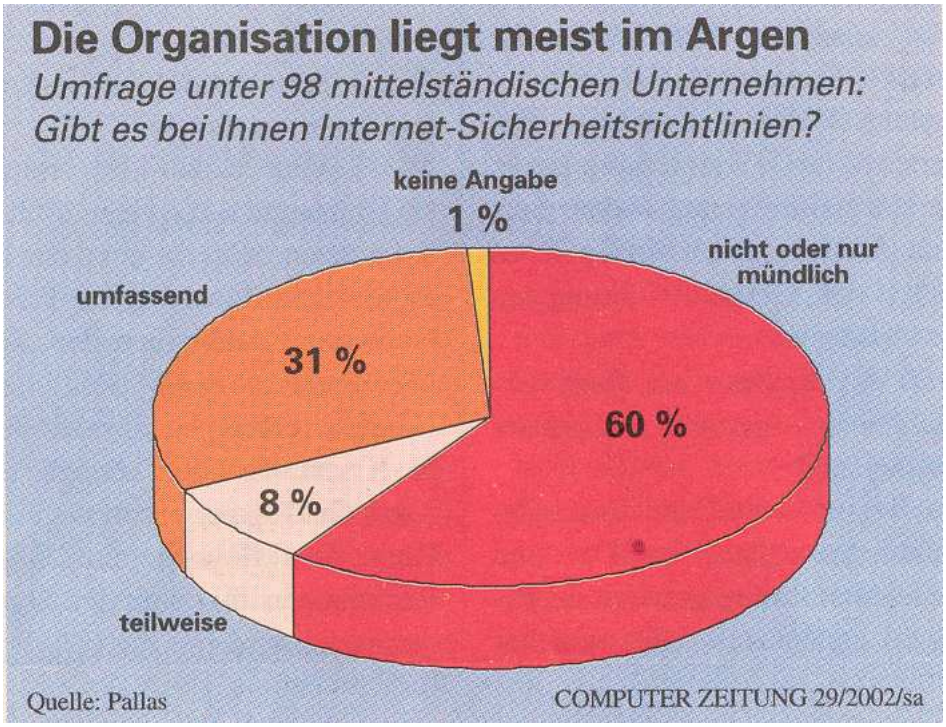

Abbildung 2: Sicherheitsbewußtsein mittelständischer Unternehmen

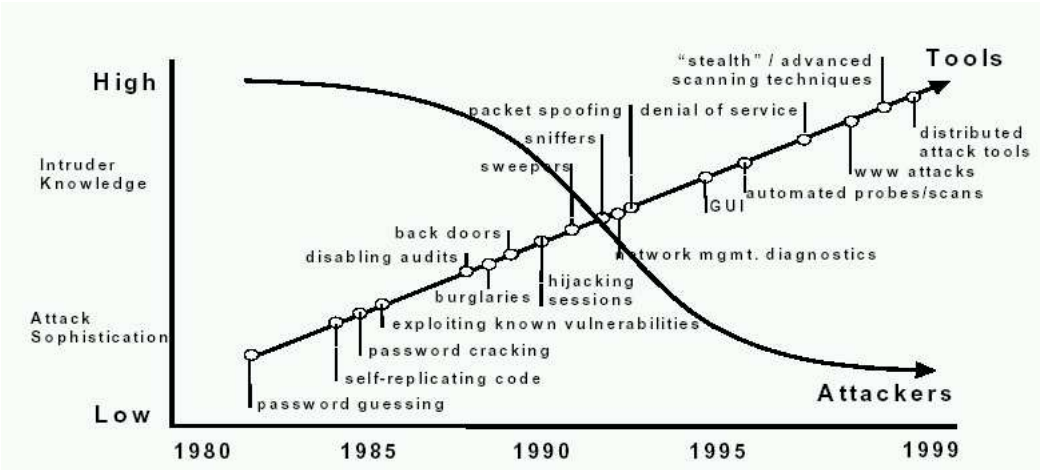

Abbildung 3: Entwicklung von Angriffswerkzeugen

2.1 Anwendungsumfrage zum Thema IDS

Um eine Einschätzung über den Einsatz von IDS zu bekommen, sowie deren Benutzbarkeit zu ermitteln, wurde im Rahmen dieser Arbeit eine Umfrage unter 300 Sicherheitsfachleuten durchgeführt. Der verwendete Fragebogen befindet sich im Anhang.

Die häufigste Antwort der Umfrage lautete, dass IDS nicht eingesetzt werden. Es liegen zum Teil zwar Konzepte bezüglich des Einsatzes vor, aber eine konkrete Umsetzung ist noch nicht im Gange.

An erster Stelle der eingesetzten Programme liegen laut Umfrage Open Source Produkte, allen voran *Snort*[1]. Bei Snort handelt es sich um ein "lightweight" Intrusion Detection für Netzwerke (NIDS), wobei sich "leightweight" nach Aussage des Entwicklers zum einen auf die geringe Last bezieht, die das Programm während des Einsatzes verursacht, zum anderen liegt dies im Ursprung des Programms, da es aus einem Netzwerksniffer entstanden ist, dem durch zusätzliche Regeln und Module ein erweiterter Aufgabenbereich zu Teil wurde.

Aus dem kommerziellen Bereich der IDS, wurden ISS Real Secure und Dragon von Enterasys genannt.

Wie aus einigen Kommentaren der Umfrage hervorging, werden IDS lediglich als zusätzliche Informationsquelle bzw. verbesserter Filter eingesetzt. Das Vertrauen in die Software ist gering und somit auch die Bereitschaft hier viel Geld zu investieren. Ein Zitat:"[...]Da uns keine Sicherheitslücken bekannt sind, die unsere Server kompromittieren könnten *und* von einem IDS erkannt werden können, ist dieses (Einstellung einer Vollzeitkraft zur Verfolgung von Angriffen) aber auch nicht notwendig."

Bei der Frage, wie hoch der administrative Aufwand sei, hat sich ergeben, dass das Open Source Produkt Snort im Durchschnitt am wenigsten Aufwand bereitet hat. Dies deckt sich mit einem Kommentar von Armin Barnitzke [barn02], in dem bemängelt wird, dass kommerzielle Security-Hersteller ihre Produkte derart komplex gestalten, dass sie ohne halbjährige Schulungen nicht mehr sinnvoll eingesetzt werden können. Den Untersuchungsergebnissen zu Folge ist diese Aussage abzuschwächen, dennoch gibt es hier einen softwareergonomischen Handlungsbedarf. In [jallen00] wird dieses Thema ebenfalls behandelt, ein Zitat lautet hier:"[...] Als Reaktion auf die Unfähigkeit der Konsumenten ein ID System vollständig zu verstehen und zu benutzen, wird seitens der Hersteller der Versuch unternommen, eine weitreichende Lösung des Sicherheitsproblems zu integrieren. Die Evaluation von ID Systemen ist nicht-trivial und es besteht ein Mangel an glaubwürdigen und umfassenden Evaluationsinformationen zu den Produkten."

In Bezug auf den Einsatz von grafischen Benutzungsoberflächen ergibt sich für Snort ein uneinheitliches Bild. Die Aussagen reichen von dem "nicht vorhanden sein" eines GUI bis hin zur fast ausschließlichen Nutzung eines solchen Interfaces. Die in dieser Untersuchung aufgeführten kommerziellen IDS werden vollständig über eine grafische Benutzungsoberfläche bedient.

Die Frage zur Echtzeitdarstellung des Gefahrenpotenzials, Ziel und Art des Angriffes spiegeln den Erfahrungswert der Administratoren wider, d.h. das Bild war unabhängig vom verwendeten IDS völlig uneinheitlich. Dies legt den Schluss nahe, dass eine detaillierte Darstellung des Ablaufs eines Angriffes notwendig ist, um einen ge-

[1] www.snort.org

ringeren Erfahrungsschatz auszugleichen. Die Verbesserungsvorschläge, die Seitens der Teilnehmer gemacht wurden sind:

- Einbindung anderer Werkzeuge ermöglichen
- Verwaltung von Adressen potentieller Angreifer[2]
- Bündelung von Erfahrungen, auf die im Falle eines Angriffes schnell zugegriffen werden kann
- Darstellung zusätzlicher Informationen, z.B.
  - Aus welchem Netzwerk erfolgt der Angriff
  - Über welchen Provider
  - Hat die Quelle eine feste IP oder Dialin-IP
- Angaben über den Erfolg eines Angriffes

2.2 Vorschläge anderer Untersuchungen

In “State of the Practice of Intrusion Detection Technologies” [jallen00] wird bereits darauf hingewiesen, das ID ein sich rasch entwickelndes Themengebiet ist. Der Schwerpunkt der Forschung liegt dabei im Bereich der Angriffserkennung. Neben der Signaturanalyse werden auch Verfahren der Anomaliedetektion und das Honigtopfprinzip verfeinert. Betrachtet man die Ergebnisse einer Diplomarbeit zum Thema “IDS - Einführung und Implementierung eines Angriffs” [eckho02], zeigt sich, das auf diesem Gebiet dringend Erfolge erzielt werden müssen, da den Angreifern bereits Werkzeuge zur Verfügung stehen, die mittels Verschlüsselung und Mutation alle gängigen netzwerkbasierten IDS aushebeln können.

Unter diesen Gesichtspunkten ist es verständlich, das der Bereich der Benutzungsoberflächen etwas zurücksteht. Ein mathematischer Ansatz wurde bereits von Greg Vert et al. [vert98] vorgeschlagen, unter softwareergonomischen Gesichtspunkten gibt es bisher keine öffentlichen Ergebnisse. Vorschläge zu diesem Thema aus [jallen00] und [bace99] sind z.B.

- Eine grafische Oberfläche, die über die Darstellung von Listen, Zuständen und Trends der Rohdaten hinausgeht
- Eine Übergeordnete Instanz, die unabhängig von der vorhandenen Infrastruktur mit Teils verschiedenen Technologien und Sicherheitspolicen, eine einheitliche Präsentation aller notwendigen Daten gestattet
- Einbeziehung der menschlichen Analysefähigkeiten in Bezug auf die Ereignisdiagnose

Zudem sollten folgende Informationen zur Verfügung gestellt werden

- Was ist passiert?

[2] Nach einem Beispiel von http://www.tu-bs.de/rz/tubsnet/nullroute.html

- Wer ist betroffen?
- In welcher Form ist das System betroffen? (Konsequenzen)
- Wer ist der Angreifer?
- Wo ist die Quelle des Angriffs?
- Wann und wie passierte der Angriff?
- Warum geschah der Angriff?

Der letzte Punkte dürfte von heutigen IDS kaum beantwortet werden können; dennoch ist diese Frage gestattet, da sie mögliche Vorgehensweisen und Folgeangriffe vorhersagbar macht.

3 Das Konzept von ID Management

Wie aus dem vorigen Abschnitt zu entnehmen ist, ist eine übergeordnete Instanz notwendig, um verschiedene Strukturen und Quellen einheitlich darstellen und verwalten zu können. Hieraus leitet sich der Begriff des Intrusion Detection Managements ab. Ähnlich wie beim Netzwerkmanagement, wo heterogene Strukturen zentral verwaltet werden, ist es das Ziel des IDM ein Netzwerk vollständig abzubilden und alle sicherheitsrelevanten Ressourcen zur Überwachung zu nutzen. Dies sind im Idealfall:

- Logdateien der jeweiligen Rechner
- Logdateien von Paketfiltern
- Informationen die von gegebenenfalls verschiedenen IDS stammen (HIDS, NIDS, an spezielle Betriebssysteme angepasste IDS)
- Ergebnisse von Analyseprogrammen, wie Passwortbrecher oder Scanner

Die zur Zeit größte Schwierigkeit ist dabei eine geeignete Schnittstelle zu den kommerziellen IDS zu schaffen. Wie bereits in [jallen00] erwähnt wurde, ist dieses Gebiet ein hart umkämpfter Markt und Releases von IDS folgen in kurzen Abständen aufeinander.

3.1 Sammeln der Informationen

Je mehr Informationen zur Verfügung stehen, desto exakter kann der Zustand des Netzwerkes wiedergegeben werden. Zur Sammlung der Informationen wird auf Agenten zurückgegriffen, die auf die oben beschriebenen Quellen zugreifen. Während das Auslesen von Logdateien unproblematisch ist, stellt die Schnittstelle zu kommerziellen IDS eine gewisse Schwierigkeit dar. Da diese Programme meist proprietär sind, und ein Zugriff auf gesammelte Daten schwer möglich ist, ist zunächst nur ein Agent für Snort vorgesehen.

Für die Verwendung externer Programme muss jeweils ein eigener Agent entwickelt werden.

Die Aufgabe der Agenten beschränkt sich auf das Sammeln der Daten, die vom IDM explizit benötigt werden. Um den Overhead so gering wie möglich zu halten, erfüllen sie keine anderen Aufgaben, wie beispielsweise eine Signaturanalyse. Ist diese Funktionalität gewünscht, so muss sie von einem Programm, z.B. einem IDS, zur Verfügung gestellt werden.

3.2 Gestaltung der Oberfläche

Die im folgenden dargestellten Skizzen dienen als Ansatzpunkt und enthalten bereits einige der aufgeführten Ideen. Welche Ansätze sich als brauchbar erweisen und welche Eigenschaften noch hinzugefügt werden müssen, wird in einer späteren Untersuchung ermittelt, die auf dem derzeitig in Entwicklung befindlichen Prototypen aufbaut.

Abbildung 4 zeigt den Aufbau eines fiktiven lokalen Netzwerkes, welches hier als *LAN 1* bezeichnet werden soll. Zu sehen sind zwei Server, zwei Desktop-Rechner, wobei ein Rechner zusätzlich über einen Modemanschluss verfügt, ein Drucker und ein Laptop. Das Laptop kann jeder Zeit vom Netz getrennt werden.

Popup-Menüs, die für jedes Symbol existieren und detailliertere Information, z.B. über die Quelle eines Angriffs oder geeignete Maßnahmen zur Verfügung stellen, sind für die Entwicklung eingeplant, werden aber in der folgenden Modellierung nicht betrachtet.

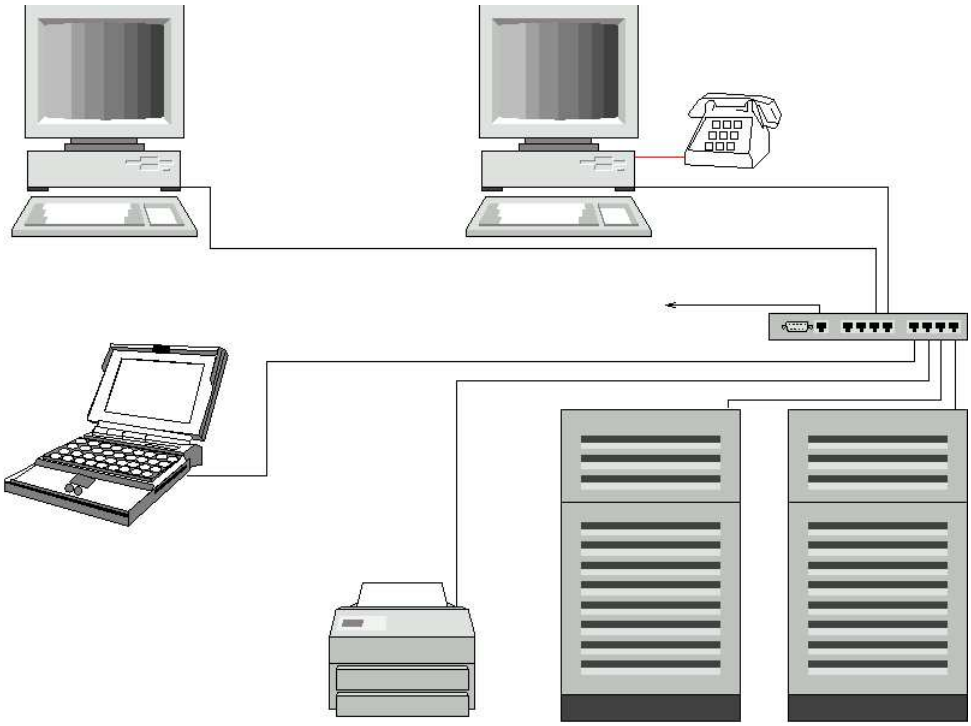

Abbildung 4: Aufbau eines LANs

3.2.1 Modellierung des Netzwerkes

Eine mögliche Modellierung des Netzes LAN 1 ist in Abbildung 5 zu sehen. Zunächst werden die Geräte den jeweiligen Räumen zugeordnet, in denen sie sich befinden. Dies ist eine optionale Maßnahme, um den Standort eines Rechners im Falle eines Angriffs schnell ausfindig machen zu können, falls ein physisches Eingreifen notwendig wird. Jedem Gerät wird dabei ein Gerätesymbol in Form eines blauen Rechtecks zugeordnet, welches zusätzliche Elemente enthält. Einige Elemente können individuell hinzugefügt werden. Zum Beispiel wurde dem Server *SV 1* in Abbildung 5 eine Auslastungsanzeige für Speicher- und CPU-Last hinzugefügt. Die genaue Bedeutung der Elemente wird in Abschnitt 3.2.2 erläutert.

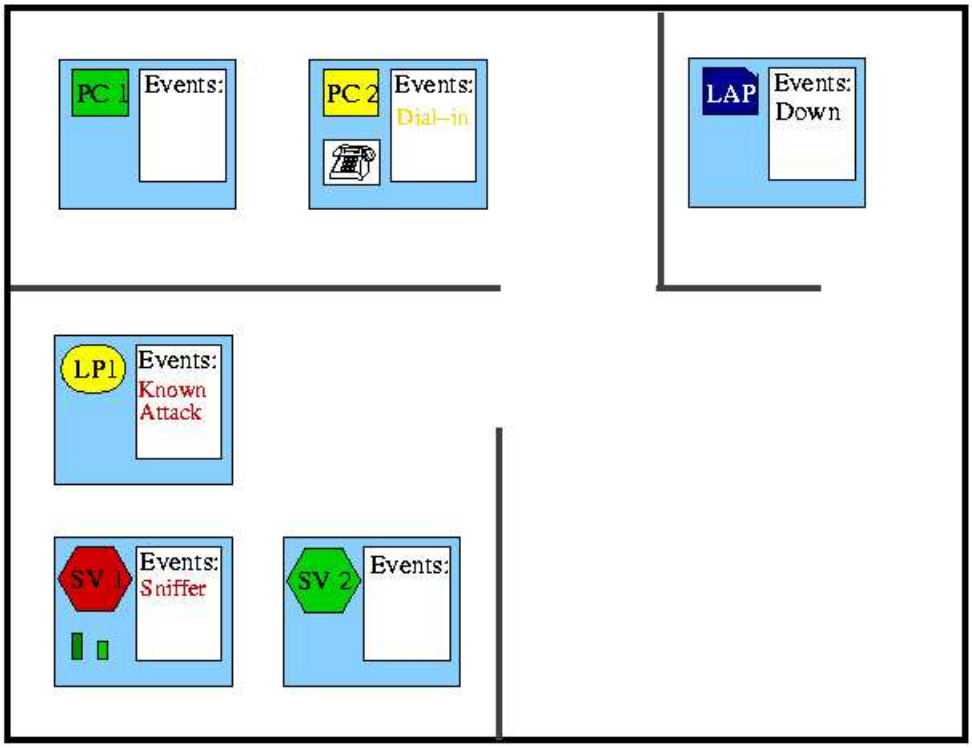

Abbildung 5: Modell des LANs

Um ein großes Netzwerk verwalten zu können, empfiehlt sich ein hierarchischer Ansatz. In Abbildung 6 ist eine höhere Ebene dargestellt. Verschiedene Subnetze oder Bereiche mit speziellen Aufgaben können in Containern zusammengefasst werden. Geräte mit sicherheitstechnisch wichtigen Aufgaben bekommen auf dieser Ebene eigene Symbole. Im Beispiel sind dies die Paketfilter. Neben den Geräten werden hier auch die Verbindungen modelliert. Eine zunehmende Linienstärke gibt eine steigende Verkehrsdichte an. Um die Kapazitätseigenschaften von Leitungen mit einfließen zu lassen, findet zusätzlich eine Einfärbung der Verbindungen statt.

Weitere Hierarchie-Ebenen sind möglich, werden an dieser Stelle aber nicht weiter behandelt.

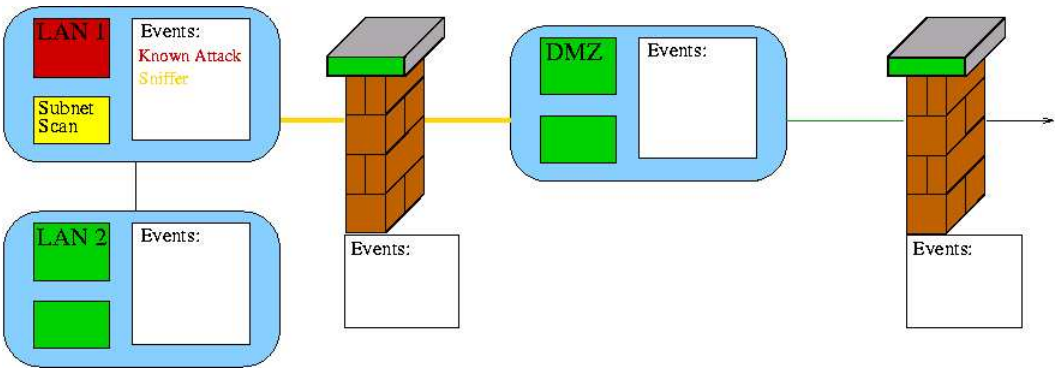

Abbildung 6: Modell eines Netzwerks

3.2.2 Bedeutung der Symbole

Abbildung 7 zeigt die im obigen Szenario verwendeten Symbole. Einige der in den Symbolen verwendeten Elemente wechseln je nach Status ihre Farbe, mögliche Farben dabei sind grün, gelb, orange und rot.

Das Containersymbol besteht aus einem blauen Rechteck, mit stark abgerundeten Ecken. Standartmäßig enthält der Container drei Elemente.

Das Eskalationsstatuselement ist ein farbig hinterlegtes Rechteck, welches eine kurze Bezeichnung des Containers beinhaltet. Im Beispiel gibt die Beschreibung an, dass hierdurch das LAN 1 repräsentiert wird. Die Farbe spiegelt die Situation der darunter liegenden Geräte wieder. Wird beispielsweise einer der Server angegriffen und erhält den Status rot, so wechselt der Eskalationsstatus ebenfalls auf rot.

Das LAN Statuselement bezieht sich nur auf den Bereich an sich. Wird z.B. dieser lokale Netzabschnitt gescannt, wechselt das Element gegebenenfalls seine Farbe, zusätzlich kann der Grund für diesen Alarm angezeigt werden. Wird stattdessen gezielt ein Rechner dieses Netzes angegriffen, findet keine Veränderung statt.

Das Eventfensterelement zeigt Ereignisse an, die den jeweiligen Netzabschnitt betreffen. Welche Ereignisse hier angezeigt werden, kann vorher definiert werden. In Abhängigkeit der Priorität lässt sich die Schrift entsprechend farbig ausgeben.

Das Gerätesymbol besteht aus einem blauen Rechteck und enthält standartmäßig zwei Elemente. In dem Beispiel wurde von der Funktion Gebrauch gemacht, zusätzliche Elemente zu platzieren. In diesem Fall handelt es sich um eine Balkenanzeige zur Darstellung der Speicher- und CPU-Auslastung des Servers.

Das Gerätebezeichnerelement enthält eine kurze Bezeichnung des Gerätes und gibt gleichzeitig den Sicherheitsstatus des Gerätes an. Im verwendeten Beispiel gibt es einen roten Alarm, da bei einer Routinekontrolle ein Sniffer

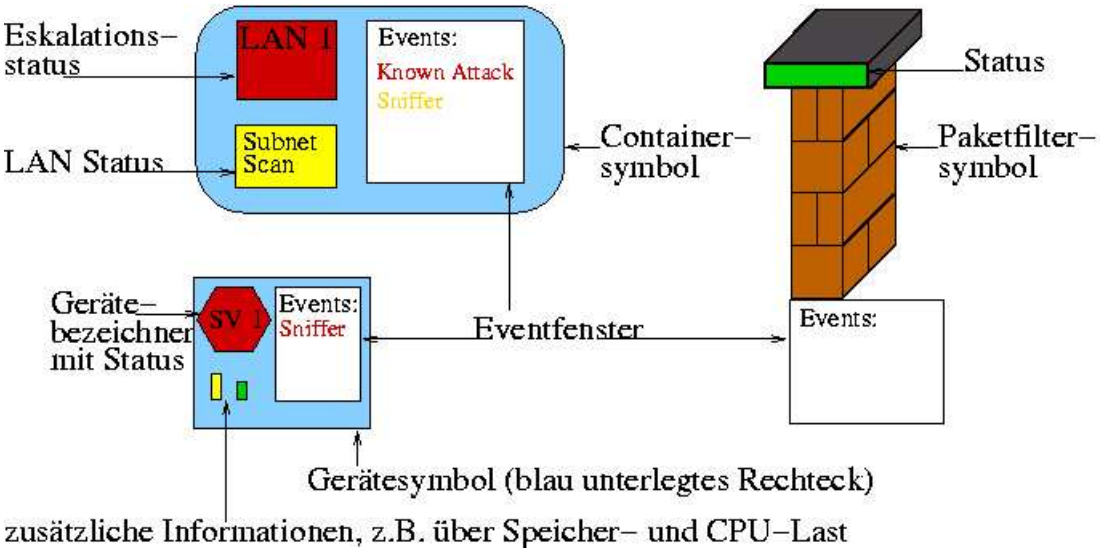

Abbildung 7: Symbollegende

auf dem Server entdeckt wurde. Die Form des Elements gibt darüber hinaus Auskunft über die Art des Gerätes, z.B. ein Sechseck für Server, ein Rechteck für Desktop-Rechner, ein Rechteck mit einer fehlender Ecke für mobile Geräte und ein Oval für fernwartbare Ausgabegeräte.

Das Eventfensterelement entspricht dem Eventfenster des Containers.

Das Firewallsymbol dargestellt durch eine schematische Mauer hat zwei Elemente.

Das Statuselement zeigt durch Farbwechsel an, wenn der Paketfilter angegriffen wird oder anderweitig in seiner Funktion beeinträchtigt ist.

Das Eventfensterelement entspricht dem Eventfenster des Containers.

4 Ausblick

Das hier vorgestellte Konzept beinhaltet die technische Grundlage für eine ID Management Plattform. Es wurde grob skizziert, wohin diese Entwicklung zielt. Ein wesentlicher Punkt ist dabei der ergonomische Ansatz. Im weiteren Verlauf ist daher neben der technischen Umsetzung der Schnittstelle eine eingehende Untersuchung vorgesehen, wie folgende Kriterien erfüllt werden können:

- einfache Installierbarkeit und Wartung
- individuelle Anpassungsmöglichkeiten in Bezug auf relevante Daten
- intuitive Bedienung, um eine schnelle Reaktion zu ermöglichen

Optional kann dem IDM eine Wiederholungsfunktion hinzugefügt werden, die nicht nur die Event eines bestimmten Zeitraumes anzeigt, sondern den Ablauf rekonstruiert und grafisch darstellt. Hierdurch liessen sich leichter Erkenntnisse über Art und Erfolg eines Angriffs gewinnen.

Literatur

[jallen00] State of the Practice of Intrusion Detection Technologies, Julia Allen et al., Carnegie Mellon, Pittsburgh PA 2000

[cz02] Security im Mittelstand ist lediglich Stückwerk, Computer Zeitung Nr. 29, Konradin Verlag, 15. Juli 2002

[eckho02] Diplomarbeit: IDS - Einführung und Implementiernug eines Angriffs, Carsten Eckholt, Universität Oldenburg, 2002

[bace99] An Introduction to Intrusion Detection Assessment, Rebecca Bace, Infidel Inc., 1999

[larsen99] Global Security Survey: Virus Attack, Amy K. Larsen, InformationWeek, 1999, (http://informationweek.com/743/security.htm)

[barn02] Die Anwender sind schlicht überfordert, Armin Barnitzke, Computer Zeitung Nr. 29, Konradin Verlag, 15. Juli 2002

[vert98] A Visual Mathematical Model for Intrusion Detection, G. Vert, D.A. Frincke, J.C. McConnel, University of Idaho, 1998

A Anhang

Fragebogen zur Evaluation von IDS Benutzungsoberflächen

Unter welchem Betriebssystem wird es eingesetzt?

Wie hoch ist bzw. war der administrative Aufwand?

- Bei der Installation
- Bei der Wartung
- Im Betrieb

(1=sehr gering - 5=sehr hoch)

In welchem Umfang wird

- eine grafische Oberfläche (GUI) benutzt?
- ein Kommandozeileninterpreter (CLI) verwendet?
- auf Skripte zurückgegriffen?

(0=wird vom IDS nicht bereit gestellt, 1=gar nicht - 5=sehr viel)

Wenn ein Angriff stattfindet, ist ausser einer rechtzeitigen Alarmierung auch die Bereitstellung von zusätzlichen Informationen wichtig (Gefahr des Angriff, welcher Bereich ist betroffen usw.)

Wie bewerten Sie die Echtzeitdarstellung von Angriffen Ihres Systems in Bezug auf

- Gefahrenpotential des Angriff?
- Ziel des Angriffs?
- Art des Angriff?
- potentielle Gegenmaßnahmen?

(1=sehr hilfreich - 5=kaum erkennbar)

Was könnte bei der Echtzeitdarstellung verbessert werden?

Welche zusätzlichen Informationen wären wünschenswert, um einem Angriff schnell begegnen zu können?

Sonstige Anmerkungen und Vorschläge