Intrusion Detection und Response: Anforderungen, Analysemethoden und Systemunterschiede

Dr. Kai-Oliver Detken[1] · Dirk Göttsche[2]

[1]DECOIT GmbH, Fahrenheitstraße 1, D-28359 Bremen
detken@decoit.de

[2]DECOIT GmbH, Fahrenheitstraße 1, D-28359 Bremen
goettsche@decoit.de

Zusammenfassung

Wenn man heute über das Thema Security in den Unternehmen nachdenkt, diskutiert man schnell über Firewall-Lösungen, um den Schutz zum Internet gewährleisten zu können. Eine korrekt installierte und sinnvoll konfigurierte Firewall kann dabei zwar viele dieser Angriffe abwehren, aber gerade die wenigen, die vielleicht nicht bemerkt werden, bringen ein erhöhtes Gefahrenpotenzial mit sich. Hinzu kommt, dass eine Firewall nur die Gefahren von außen betrachtet und die Sicherheit des internen Netzes nicht beachtet. Die Gefährdung kann hierbei sowohl von bereits eingedrungen Hackern als auch von den regulär im Netz arbeitenden Personen verursacht werden. Dieser Fachbeitrag untersucht daher Intrusion Detection und Intrusion Response Systeme sowie deren Analysemethoden und Systemunterschiede. Diese Systeme, im Folgenden der Einfachheit halber IDS genannt, werden als eine Art „Alarmanlage“ innerhalb der Netze zur Überwachung, Alarmierung und zum Schutz eingesetzt. Allerdings ist für den Einsatz eine genaue Planung und Anforderungsanalyse notwendig, da sonst ein IDS-System von internen Meldungen überrollt wird. Auch sollte man die vorhandenen Schwachstellen vorab genau untersuchen und offen legen. Hinzu kommt, dass es unterschiedliche Analysemethoden gibt, die mehr oder weniger gut Angriffe lokalisieren und ggf. dagegen wirken können. Abschließend will dieser Beitrag kommerzielle Systeme mit Open-Source-Lösungen vergleichen. Hierfür wurden reale Tests durchgeführt und ausgewertet.

1 Schwachstellen heutiger Netzinfrastrukturen

Durch das ständige Entdecken und Veröffentlichen immer neuer Sicherheitslücken wird eine breite Grundlage für Angriffe im Netzwerk geschaffen. Dabei stellt sich oft heraus, dass schon lange bekannte Schwachstellen, für die es bereits Sicherheitslösungen gibt, an vielen Stellen im Netz immer noch für Angriffe erfolgreich genutzt werden können. Für viele programmiertechnische Sicherheitslücken stehen häufig kurze Zeit nach der Entdeckung so genannte Patches zur Verfügung, die den Fehler beseitigen (manchmal allerdings auch neue Fehler mit sich bringen). Das Schließen dieser Schwachstelle mit dem Patch muss in der Regel jedoch manuell durch den Sicherheitsverantwortlichen oder Administrator des betroffenen Netzes erfolgen. Sicherheitsprojekte zeigen jedoch immer wieder, dass in Punkto Sicherheit der Soll- und Ist-Zustand eines Netzwerkes stark von einander abweichen. Das heißt, dass bei

P. Horster (Hrsg.) · DACH Security · syssec (2005) pp.

der Betrachtung der Schwachstellen sowohl die technisch-konzeptionelle Seite als auch die personell-organisatorische Seite betrachtet werden müssen.

1.1 Technische und konzeptionelle Schwachstellen

In diesen Bereich fallen alle Schwachstellen die durch Betriebssysteme, Applikationen, Dienste, Protokolle und die eingesetzten Technologien verursacht werden. Durch die Schnelllebigkeit am Markt und verkürzte Produktlebenszyklen werden die Produkte oftmals nicht mit der nötigen Sorgfalt entwickelt, die ein Fokus auf Sicherheit mit sich bringen würde. Stattdessen werden durch den Konkurrenzdruck noch mehr Features in die Produkte implementiert, die im Nachhinein wieder neue Sicherheitslücken aufweisen.

Betrachtet man die Betriebssysteme und die Applikationen sind besonders im Bereich der Mail- und Web-Server sowie bei Web-Browsern und so genannten Messangerprogrammen immer wieder Sicherheitslücken aufgetreten, die hauptsächlich auf Programmierfehler des Herstellers zurückzuführen sind. Diese Sicherheitslücken reichen von Systemabstürzen über Fehlfunktionen bis hin zu unerlaubten Zugriffen auf das komplette System.

Ein weiterer Ansatzpunkt für Angriffe sind die vom Zielrechner zur Verfügung gestellten Dienste. Durch die Vielfalt der Netzdienste geht häufig der Überblick in Bezug auf die Sicherheit bei der Konfiguration der Systeme verloren. Zum einen werden für das jeweilige System überflüssige Programme und Dienste mit installiert und gestartet (z.B. unnötige Serverdienste auf einem Clientsystem). Zum anderen werden die Systeme oftmals aus Unwissenheit falsch konfiguriert. Dadurch schafft man zusätzliche Angriffspunkte, die zu Missbrauch und Einbruch führen können. Sind diese Sicherheitslücken erst einmal von einem potenziellen Angreifer entdeckt worden, dienen diese betroffenen Systeme meist als Basis für weitere verteilte Angriffe.

Grundsätzlich werden alle Informationen im Internet offen übertragen und können daher mit so genannten Netzwerk-Sniffern mitgelesen werden. Es gibt zwar inzwischen sichere Methoden wie Virtual Private Networks (VPN), Secure Shell (SSH) und Secure Socket Layer (SSL), deren Einsatz sich immer mehr durchsetzt. Trotzdem werden in vielen Bereichen noch unsichere E-Mail-, FTP-, Telnet- und andere Remote-Login-Dienste zur Datenübertragung und Fernadministration genutzt. Damit ist nicht nur der Inhalt einer E-Mail oder einer Datei offen sichtbar, sondern auch die Zugangsdaten wie Benutzername und Passwort können in falsche Hände gelangen.

Viele Protokolle der TCP/IP-Familie ermöglichen eine Authentisierung miteinander kommunizierender System nur über die IP-Adresse, die jedoch gefälscht werden kann. Die Synchronisation beim Aufbau einer TCP/IP-Verbindung erfolgt über Sequenznummern, die ebenfalls leicht erraten werden können. So ist man in der Lage IP-Pakete mit jeder beliebigen Absenderadresse zu verschicken. Wenn die Kommunikation im Internet vernetzter Computer über IP-Adressen authentisiert wird, dann können durch IP-Spoofing und DNS-Spoofing falsche IP-Adressen vorgetäuscht werden. Rechte, die nur auf Grund der IP-Adresse vergeben werden, sind somit einfach zu missbrauchen.

Betrachtet man die eingesetzten Technologien, so erkennt man nicht nur bei neuen Techniken wie der Funktechnologie Wireless LAN (WLAN) und Massenspeicher mit USB-Anschluss, sondern auch bei altbewährten Ausstattungen wie Diskettenlaufwerken und CD-Brennern diverse Schwachstellen. WLAN-Komponenten haben standardmäßig zwei Sicherheitslösungen

anzubieten. Die Festlegung zugriffsberechtigter MAC-Adressen und die Aktivierung der WEP-Verschlüsselung (Wired Equivalent Privacy). Beide Methoden bieten jedoch nur in gewissem Maße Schutz vor Angreifern, da sie mit ein wenig Aufwand überlistet werden können. Die physikalische Nummer eines Netzwerkadapters, die MAC-Adresse, stellt zwar jede für sich eigentlich ein weltweites Unikat dar; es gibt aber inzwischen Tools mit denen die Adresse manipuliert werden kann. Beim Einsatz des Address Resolution Protocol (ARP) kann problemlos die MAC-Adresse ausgelesen werden und dann mit gefälschter Adresse ungehinderter Zugriff ins Netz erlangt werden.

Die eingesetzte WEP-Verschlüsselung ist inzwischen selbst in der stärksten Version mit einem 128-Bit-Schlüssel geknackt worden. So kann durch den Einsatz von Netzwerk-Sniffern über einen von der Auslastung des WLANs abhängigen Zeitraum, der Netzwerkverkehr mitgelesen und zur Berechnung des eingesetzten Schlüssels missbraucht werden. Da das Funksignal unkontrolliert in einem gewissen Radius ausgestrahlt wird, muss sich der Angreifer im schlechtesten Fall nicht einmal innerhalb des Gebäudes oder des Grundstückes befinden.

Die Ausstattung heutiger Rechner wird immer umfangreicher und birgt somit weitere Gefahrenquellen im Bereich des Datendiebstahls durch Speichermedien (ZIP, CD-R/RW) und externe Festplatten und Laufwerke mit USB- oder FireWire-Anschluss. Aber auch schlechte BIOS-Einstellungen, die nach der Systeminstallation weiterhin den Bootvorgang von CD ermöglichen, sind als gefährlich einzustufen.

Mit so genannten Live-CDs aus dem Linux-Umfeld (z.B. Knoppix oder SuSE) kann von CD aus ein vollfunktionsfähiges Linux-System auf jedem beliebigen Rechner gestartet werden. Linux ist in der Lage unter Windows geschützte Bereiche zu lesen und kann somit auf gesicherte Daten des jeweiligen Systems zugreifen.

Besonders deutlich sind die Sicherheitslücken überall dort zu erkennen, wo immer gleich die neueste Technik bedenkenlos eingesetzt wird und bestehende Infrastruktur vom Sicherheitsaspekt aus nach der Inbetriebnahme nicht mehr gewartet oder kontrolliert wird.

1.2 Personelle und organisatorische Schwachstellen

Zahlreiche erfolgreiche Angriffsversuche nutzen Fehler und Schwachstellen aus, die schon seit Längerem bekannt sind, sodass ein zentraler Punkt einer Sicherheitspolitik Schulung und Information sein muss. Hauptgrund für Sicherheitsvorfälle in Datennetzen sind mangelhafte Systemkonfigurationen sowie teilweise oder ganz fehlende Zugangsbeschränkungen für Anschlüsse zu öffentlichen Datennetzen.

Ein nicht zu unterschätzender Anteil am Gefahrenpotenzial wird unter dem Begriff des „Social Engineering“ zusammengefasst. Dabei versucht der Angreifer im Vorfeld wichtige Informationen über Art und Aufbau des Netzes in Erfahrung zu bringen. Diese geschieht meistens per Telefon indem der Angreifer sich z.B. als Administrator ausgibt und versucht den Anwendern Passwörter oder andere relevante Information zu entlocken. Aber auch direkt Vorort getarnt als Service-Mitarbeiter oder Kunde kann der Angreifer versuchen Zugang zu Rechnern oder Daten zu erhalten.

Gerade im sicherheitsrelevanten Bereich des Netzwerkes gibt es oftmals keine klaren Regelungen bezüglich des Zugangs oder die Bereiche sind einfach für jeden offen. Die genannten Sicherheitslücken können nicht einfach durch die Integration eines IDS geschlossen werden, aber sie verdeutlichen das Intrusion Detection nicht nur technisch betrachtet werden darf.

2 Anforderungen an ein IDS/IRS

Um das für den jeweiligen Einsatzbereich richtige IDS zu finden, müssen im Voraus alle relevanten Anforderungen bekannt sein. Nur so ist es möglich bei der Vielzahl der verfügbaren Systeme, die optimale Variante zu ermitteln oder gegebenenfalls mehrere Lösungen zu kombinieren. Allgemein muss ein IDS in der Lage sein Angriffe und Einbrüche auf die Systeme des zu schützenden Netzes zu erkennen und durch entsprechende Reaktionen die Verteidigung dieser Systeme zu gewährleisten.

Die wesentlichen Ziele eines IDS sollten wie folgt definiert sein:

- Integrität der zu schützenden Daten
- Verfügbarkeit dieser Daten
- Verfügbarkeit der gewünschten Dienste
- Automatische Reaktionen auf Angriffe und Einbrüche
- Justierbare Fehlertoleranz zur Vermeidung von Fehlalarmen
- Betrieb mit geringem administrativen Aufwand
- Minimale Belastung des Netzverkehrs
- Geringe Belastung der zu schützenden Systeme
- Einfache Implementierung in die bestehende Struktur
- Zuverlässiger und sicherer Betrieb

Im speziellen Fall sollte eine genaue Anforderungsliste die gewünschten Optionen konkret beschreiben.

2.1 Echtzeitfähigkeit

Ein IDS muss in der Lage sein, die Erkennung, Alarmierung und Reaktion auf Angriffe in Echtzeit durchzuführen. Dazu sollte die Leistungsfähigkeit des IDS an das Aufkommen von Netzverkehr und Daten angepasst sein, damit bei der Datensammlung und Analyse keine Informationen verloren gehen. Die Sammlung und Analyse der Daten darf die zu schützenden Systeme nicht zu stark belasten. Gerade der Einsatz eines Hostbasierten IDS muss die Belastung des zu überwachenden Systems mit berücksichtigen. Echtzeitfähigkeit kann durch die Kombination verschiedener Analysemethoden erreicht werden, da die reine Analyse anhand von umfangreichen Signaturdatenbanken nicht für jeden Anwendungsfall, die effektivste Methode ist.

2.2 Sicherheit

Das IDS selbst muss vor möglichen Angriffen besonders geschützt sein. Nicht nur der kontinuierliche, fehlerfreie Betrieb sollte sichergestellt sein, sondern auch die Konfigurationsdateien, Signaturdatenbanken und Protokolldateien des IDS müssen vor Angriffen und Manipulation geschützt werden. Der Einsatz eines IDS darf keine neuen Schwachstellen und Sicherheitslücken in das Netzwerk einbringen. Die optimale Sicherheit kann durch Trennung von produktiven und überwachenden Komponenten erfolgen. Bei Netzbasierten Sensoren kann diese Trennung über gespiegelte Switch-Ports erfolgen. Hostbasierte Sensoren lassen sich jedoch nicht vom Produktivsystem trennen. In diesem Fall muss die Implementierung des Host-

sensors besonders sicher sein. Des Weiteren muss die Übertragung der gesammelten Daten zur Analyseeinheit sicher sein und notfalls verschlüsselt erfolgen. Bei Ausfall von IDS-Komponenten muss der Sicherheitsverantwortliche automatisch alarmiert werden. Der Ausfall einzelner Komponenten darf den sicheren Gesamtbetrieb des IDS nicht beeinflussen.

Folgende Ebenen müssen für die Angriffserkennung einbezogen werden können von einem IDS-System, wobei man Unterschiede bei der Architektur berücksichtigen muss:

1. Erkennung von Angriffen auf Netzwerkebene
2. Erkennung von Angriffen, die sich gegen einen Host richten
3. Erkennung von Anomalien bzw. Angriffsmethoden

2.3 Datenquellen

Ein IDS sollte in der Lage sein, Daten über verschiedene Sensoren zu sammeln und auszuwerten. Beim Einsatz unterschiedlicher Sensoren (Host- und Netzbasierte) sollte gewährleistet sein, dass die gesammelten Daten in einem einheitlichen Datenformat an die Analyseeinheit übermittelt werden.

2.4 Flexibilität

Durch den starken Anstieg neuer Angriffstechniken und Sicherheitslücken muss ein IDS schnell und zuverlässig an neue Signaturen oder neu erkannte Anomalien angepasst werden können. Daher ist es besonders wichtig, dass Updates der Signaturdatenbank nicht nur durch den Hersteller zur Verfügung gestellt werden, sondern auch der Betreiber des IDS selbst eigene Signaturen einpflegt oder bestehende verändern kann. Ein IDS sollte sich an das Sicherheitskonzept nach den individuellen Bedürfnissen anpassen lassen.

2.5 Adaptivität

Bei Änderungen im Nutzerverhalten sollte ein IDS in angemessener Zeit selbstständig darauf reagieren können. Die Justierung des IDS ist in diesem Bereich besonders schwierig, da Veränderung zuerst immer als Angriff gewertet werden müssen.

2.6 Betriebssystemvielfalt

Der Einsatz und die Auswahl eines IDS sollten sich an der schon vorhandenen Infrastruktur des Netzes orientieren. Ein IDS, das unter einem bereits verwendeten Betriebssystem implementiert wird, nutzt vorhandenes Wissen und verkürzt dadurch die Einarbeitungs- und Administrationszeit. Dieses Argument trifft natürlich nur auf Softwarebasierte IDS-Lösungen zu. Werden so genannte „Appliances“ eingesetzt, ist man unter Umständen von proprietären Herstellerlösungen abhängig.

2.7 Bediendung und Komfort

Die Bedienung und Konfiguration eines IDS sollte möglichst einfach implementiert sein. So sollten eine umfangreiche Signaturdatenbank und vorkonfigurierte Sicherheitsrichtlinien zur Grundeinstellung des IDS gehören. Das System sollte über eine benutzerfreundliche und in-

tuitive Oberfläche konfiguriert werden können. Auch die Alarmmeldungen und Protokolldateien müssen durch entsprechende Reportingtools für den Sicherheitsverantwortlichen leicht und verständlich einsehbar sein.

2.8 Beeinflussung der Netzperformance

Der Einfluss des IDS auf die Performance des zu überwachenden Netzes muss so gering wie möglich sein. Dazu können die gesammelten Daten z.B. über ein parallel zum produktiven Netz verlaufendes Netz geschickt werden, um die Netzlast nicht unnötig zu erhöhen. Bei Überwachung einzelner Hosts darf das IDS nicht die eigentliche Arbeit des Systems stören. Deshalb sollten die zu überwachenden Systeme leistungsfähig genug sein, um beide Prozesse zuverlässig zu ermöglichen.

2.9 Alarmierung

Ein IDS sollte verschiedene Alarmierungsmöglichkeiten unterstützen, da zu sollten alle modernen Kommunikationswege wie z. B. E-Mail, Pager oder SMS, aber auch Popup-Konsolenmeldung. Von Alarmierungen, die nur über E-Mail laufen, ist abzusehen, da der Mailserver immer ein potenzielles Ziel für den Angreifer darstellt, z.B. durch DoS-Attacken. Die Alarmmeldungen des IDS müssen nach Prioritäten definiert werden können, sodass auch unterschiedliche Reaktionen des IDS einzustellen sind. Die Alarmierung und die dadurch ausgelöste Reaktion müssen umgehend nach dem bei der Analyse erkannten Angriff erfolgen. Alle Alarmmeldungen können nach verschiedenen Prioritäten (High, Medium, Low) zugeordnet und angezeigt werden.

2.10 Authentizität

Ein ausgelöster Alarm muss mit hundertprozentiger Sicherheit vom IDS stammen. Die Authentizität der gesamten IDS-Daten muss durch eine fälschungssichere Zertifizierung gewährleistet werden. Nur dann können die Daten im Falle rechtlicher Konsequenzen gegen den Angreifer verwendet werden. Das IDS muss sicherstellen, dass alle von gesammelten Daten aus vertrauenswürdigen Quellen stammen. Daher sollte sowohl die Authentizität der Quelle als auch die Integrität der erhaltenen Daten überprüft werden.

2.11 Intrusion Response

Bei einem erkannten Angriff und daraufhin ausgelöster Reaktion müssen alle dazugehörigen Ereignisse und weitere Protokollinformationen aufgezeichnet (beispielsweise in einer Protokolldatei) oder mit Hilfe eines Druckers protokolliert werden. Das IDS sollte in der Lage sein die Verbindung über die der Angriff erfolgt zu beenden. Außerdem muss der Aufruf benutzerdefinierter Programme oder Skripte möglich sein, um weitere Informationen und Beweise über den Angreifer zu sammeln. Bei Erkennung von Angriffen können zusammenfassend verschiedene Maßnahmen eingeleitet werden. Diese Maßnahmen müssen in der Konfiguration des IDS vorab berücksichtigt worden sein. Mögliche Maßnahmen sind u. a. das Senden von Alarmen per E-Mail bzw. per SMS an den Systemadministrator, Unterbrechung der Verbindung, Erweiterung der Protokollfunktionen.

3 Funktionsweise und Analysemethoden

An dieser Stelle soll kurz auf die Funktionsweise von IDS eingegangen werden. Dies beinhaltet die Komponenten und das standardisierte Rahmenwerk, welches die Arbeitsweise verdeutlicht.

3.1 IDS-Komponenten

IDS-Systeme setzen sich aus den nachfolgend aufgelisteten Komponenten zusammen, wobei der Einsatz der Sensoren, je nachdem was überwacht werden soll, variieren kann:

- **Netzsensoren:** Die Netzsensoren dienen dazu den Netzverkehr eines Rechners bzw. eines Teilnetzwerkes auf Ereignisse zu überwachen, die in Bezug auf die Sicherheitspolitik des Unternehmens als verdächtig erscheinen. Für den Einsatz eines Netzsensors wird entweder ein separater Rechner eingesetzt oder der Hersteller des Sensors liefert diesen als Appliance.
- **Hostsensoren:** Hostsensoren werden auf den Rechnern installiert, die auf verdächtige Ereignisse überwacht werden sollen. Sie werden eingesetzt, um Angriffe zu erkennen die sich gegen Anwendungen bzw. das Betriebssystem des jeweiligen Rechners richten.
- **Datenbankkomponente:** Alle Daten werden in der Datenbank abgelegt, die für das Monitoring und die Auswertung relevant sind. Hier sind auch Signaturen vorhanden, um die Rechtevergabe ablegen zu können.
- **Auswertungsstation:** Damit die Ereignisdaten, die während der Überwachung anfallen, zu einem späteren Zeitpunkt genauer analysiert werden können, ist es erforderlich die Daten in geeigneter Form z. B. einer Firewall zu erweitern.
- **Managementstation:** Damit ein IDS-System sinnvoll eingesetzt werden kann, ist es zwingend erforderlich, es an die Anforderungen des Unternehmens anzupassen. Sämtliche Einstellungen, die an dem IDS vorgenommen werden, erfolgen über die Managementstation.

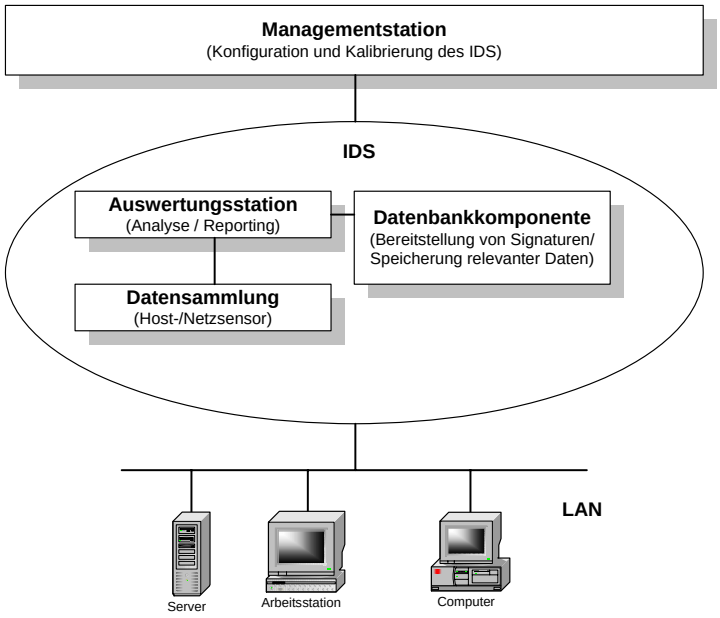

Abb. 1: Schematisch Darstellung eines IDS-Systems

Die Abb. 1 zeigt eine schematische Darstellung für den Aufbau eines IDS und wie ein solches System z. B. in einem Netzwerk eingesetzt wird. Außerhalb steht die Managementstation. Sie dient zur Konfiguration und Kalibrierung des IDS. Der Kern des IDS wird durch die Komponenten im Oval bestimmt. Hier befinden sich die Komponenten zur Datensammlung und Datenanalyse. Die Analysekomponente greift auf die Datenbankkomponente zu, welche die Signaturen bereitstellt und gesammelte Daten speichert. Die Sensoren, innerhalb dieser Abbildung ein Netzsensor, greifen auf das LAN zu und filtern die passierenden Pakete.

Die Analysemethoden von Intrusion Detection Systemen unterteilen sich hauptsächlich in zwei Bereiche. Zum einen die Erkennung von Angriffen anhand von Signaturen und zum anderen die Erkennung von Abweichungen vom definierten Normalbetrieb eines Systems oder des Netzverkehrs durch Anomalieanalyse. Heutige Systeme bevorzugen das Erkennen von Angriffen anhand von Signaturen und sind daher auch nur so gut, wie die Signaturen dies zulassen.

3.2 IDS-Verhalten

IDS-Systeme können mit unterschiedlichen Verhaltensweisen auf einen erkannten Angriff reagieren. Hierbei sind aktives und passives Verhalten zu unterscheiden:

1. Bei einem IDS mit passivem Verhalten reagiert das Überwachungssystem lediglich mit Warnnachrichten an andere Systeme bzw. Administratoren. Diese Nachrichten werden vorzugsweise mit dem SNMP-Protokoll oder per E-Mail übertragen, eine andere Möglichkeit ist die Darstellung als Popup-Fenster auf einem Systembildschirm. Es werden keine Versuche unternommen den Angriff in irgendeiner Form zu beenden (Trennen der Verbindung, Herunterfahren des Rechners).
2. Ein aktives System reagiert mit einem konfigurierten Verhalten auf den erkannten Angriff. Es kann versuchen den Angriff zu verhindern, indem das IDS z. B. die Netzverbindung trennt, den entsprechenden Port schließt oder den Rechner herunterfährt. Es besteht aber auch die Möglichkeit die protokollierenden Aktivitäten zu verstärken, um so mehr Informationen über diesen Angriff zu erhalten. Die gesammelten Informationen können dann sowohl zur Überarbeitung des Regelwerks der Firewall als auch für die technische und rechtliche Verfolgung des Angreifers eingesetzt werden.

3.3 IDS-Rahmenwerk

Die Arbeitsgruppe des CIDF (Common Intrusion Detection Framework) hat sich das Ziel gesetzt, eine allgemeingültige Architektur für ID-Systeme zu definieren. Sie soll eine Integration und Kompatibilität von Lösungen verschiedener Hersteller ermöglichen. Nach dieser Definition besteht ein IDS aus vier Komponenten, die auch als Boxen bezeichnet werden. Im Folgenden sind das die Event-Box, Analysis-Box, Countermeasure-Box und Storage/Data-Box.

Die Abb. 2 zeigt die Beziehung der vier Komponenten untereinander. Diese Struktur ist unabhängig von der Art des IDS und gilt gleichermaßen für Hostbasierte (HIDS) und Netzwerkbasierte Intrusion Detection Systeme (NIDS). Die Event-Box ist für die Sammlung der Daten zuständig, als NIDS-Sensor liest sie den Netzwerkverkehr für den ausgesuchten Bereich mit und als HIDS-Sensor soll sie die relevanten Logdateien des zu schützenden Rech-

ners auslesen. Die gesammelten Daten werden von der Event-Box in ein einheitliches Format gebracht, das von der CIDF unter der Bezeichnung GIDO (General Intrusion Detection Object) definiert wurde. In diesem Format werden die Daten an die Analysis-Box und die Data Box, übertragen. Bei der Umwandlung in das GIDO-Format sollen redundante Informationen entfernt werden, um das Datenvolumen zu reduzieren und die Netzlast bei Übertragung zu senken.

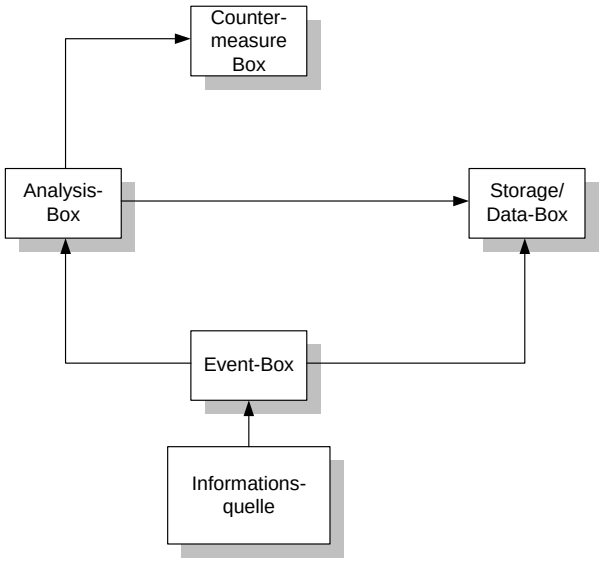

Abb. 2: Informationsfluss eines IDS

Die Analysis-Box ist das Herz des IDS. Hier werden die Daten von der Event-Box auf Angriffsmuster, Anomalitäten und nicht eingehaltene Sicherheitsvorgaben überprüft. Ergibt sich aus der Analyse ein Ergebnis für das eine Aktion definiert wurde, wird diese Aktion durch die Countermeasure-Box, oder auch Intrusion-Response-Einheit genannt, ausgeführt. Das Analyseergebnis wird außerdem an die Storage/Data-Box zur Speicherung weitergeleitet. Die Countermeasure-Box führt die Reaktionen auf die von der Analysis-Box gemeldeten Ereignisse durch. Die Reaktionen werden in die Bereiche aktiv und passiv unterteilt.

Passive Reaktionen können folgende Schritte beinhalten:

- Alarmierung der Sicherheitsverantwortlichen über verschiedene Kommunikationswege (E-Mail, Pager, SMS, Popup etc.)
- Anzeigen der gesammelten Daten
- Volumen der Datensammlung erhöhen
- Abschalten kritischer Systeme und Dienste
- Regelwerk der Firewall anpassen
- Verantwortlichen des angreifenden Systems informieren

Die Schritte der aktiven Reaktion könnten wie folgt aussehen:

- Informationen über das angreifende System sammeln, z.B. eingeloggte Benutzer (fingerd, rusersd oder identd), angebotene Dienste (Portscanning) und das verwendete Betriebssystem (OS Fingerprinting)
- Gesammelte Informationen für Gegenangriff nutzen

- Denial-of-Service (DoS) Attacken auf das angreifende System

4 IDS-Architekturen

IDS-Systeme können aufgrund ihres Einsatzortes und den zu überwachenden Daten/Systemen folgendermaßen kategorisiert werden:

- Host Intrusion Detection System (HIDS)
- Network Intrusion Detection System (NIDS)

4.1 Host Intrusion Detection Systeme (HIDS)

Ein HIDS analysiert Daten, die auf einem Rechner durch das Betriebssystem oder Applikationen erzeugt werden oder das HIDS erzeugt selbst so genannte Ereignisprotokolle. Dazu wird ein HIDS-Sensor auf dem zu überwachenden System installiert und liefert die erfassten Daten an die Analysis-Box. Zur Überwachung mehrerer Systeme können viele HIDS in einer verteilten Struktur aus so genannten Agenten und einer zentralen Managementeinheit aufgebaut werden.

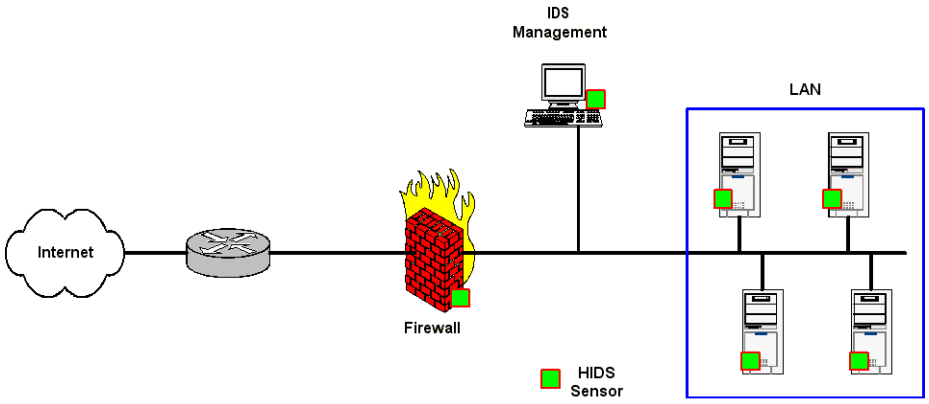

Abb. 3: Aufbau eines HIDS

Besondere Formen von HIDS, welche die Integrität des zu schützenden Systems überwachen, werden als System Integrity Verifier (SIV) oder File Integrity Assessment (FIA) bezeichnet. Für die hierbei eingesetzten Integritätstests wird nach der Installation und Konfiguration des zu überwachenden Systems eine Momentaufnahme aller relevanten Systemdatei gemacht. In der Regel werden die Dateieigenschaften und die Prüfsummen gespeichert. Das HIDS vergleicht dann in regelmäßigen Abständen den IST-Zustand dieser Parameter mit den gespeicherten Werten der Momentaufnahme. Stimmen die Werte nicht überein wird ein Alarm ausgelöst, der die Änderung mit Zeitpunkt und Nutzer ausgibt. Die Überprüfung der Rechtmäßigkeit der Änderung muss dann der Sicherheitsverantwortliche selbst auswerten.

Die aufwändigste Variante von HIDS betreibt eine Echtzeitanalyse von allen System- und Dateizugriffen. Dazu muss das HIDS auf Betriebssystemebene eingreifen. Bei Linux-Systemen wird das HIDS als Kernelmodul implementiert und kann dadurch jede Art von Zugriff überwachen. In dieser Variante kann das HIDS Zugriffsrechte und Dateien kontrollieren und die Aktion gegebenenfalls auch unterbinden. So können z.B. Regeländerungen einer Firewall oder häufig wiederholte Anmeldeversuche vom IDS in Echtzeit verhindert werden.

Die letzte Variante der HIDS ist die Analyse von Protokoll- oder Logdateien die vom Betriebssystems oder einer Applikation zur Verfügung gestellt werden. Im Regelfall werden die Argumente dieser Dateien mit einer so genannten Positivliste verglichen. Kommt es zu Übereinstimmungen, wird ein Alarm ausgelöst. Bei der Definition dieser Liste könnten jedoch wichtige Ereignisse übersehen werden, sodass es sinnvoller wäre eine Negativliste zu definieren und alle Übereinstimmungen ignoriert werden. Dann würden die Alarme nur durch unbekannte Ereignisse ausgelöst. Diese Funktionalität wird aber nicht immer vom HIDS unterstützt.

4.2 Network Intrusion Detection Systeme (NIDS)

Ein NIDS liest die benötigten Daten aus dem Netzwerkverkehr heraus. Dazu wird ein Netzwerk-Sniffer verwendet. Der Netzwerksensor wird dabei üblicherweise als separater Rechner oder „Appliance“ installiert, damit die produktiven Systeme nicht in ihrer Arbeit beeinflusst werden. Die Event-Box eines NIDS liest die Daten, die über das zu überwachende Netzwerksegment transportiert werden mit und übermittelt sie dann an die Analysis-Box. Auf Grund der Header-Informationen, wie Flags und Attribute, können DoS-Attacken oder Scans erkannt werden. Die Daten der Payload, die vom TCP/IP-Stack an die Applikationsebene weitergereicht werden, können durch die Analysis-Box auf Angriffsmuster überprüft werden. Oftmals sind die Sensoren auch in die Router oder Switches integriert, um dort die Protokollspezifikationen zu überwachen.

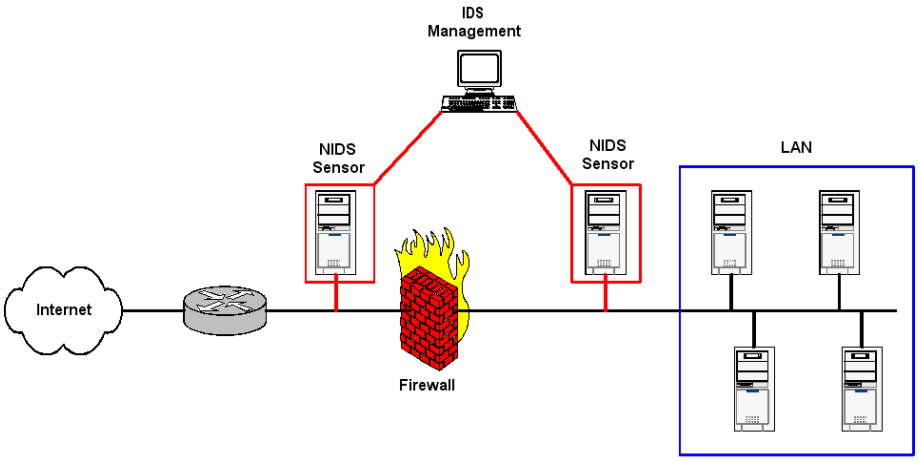

Abb. 4: Aufbau eines NIDS

Die Sensoren des NIDS werden meistens von einer zentralen Managementeinheit verwaltet. Aus Sicherheitsaspekten ist es sinnvoll, dass die Sensoren kein Bestandteil des aktiven Netzes sind, sondern parallel dazu über ein extra Netz kommunizieren. Dadurch können die Sensoren im so genannten „Stealth Mode“ betrieben werden und sind für den potenziellen Angreifer nicht erreichbar. Bei der Implementierung von NIDS werden unterschiedliche Technologien verwendet. Die heutigen freien und kommerziellen Systeme unterstützen meist mehrere Technologien auf einmal. Die Signaturerkennung wird von den meisten NIDS verwendet. Dafür werden die Signaturen oder auch Fingerprints aller bekannten Netzwerkangriffe in einer Datenbank gespeichert. Das NIDS vergleicht dann jedes Paket mit dieser Datenbank und löst bei Übereinstimmung einen Alarm aus.

Da heutzutage die verwendeten Protokolle unterschiedliche Codierungen unterstützen, muss ein NIDS in der Lage sein mittels Protokolldecodierung die Darstellung des Inhalts der gesammelten Daten in eine Normalform zu bringen. Andernfalls wäre es dem Angreifer möglich durch Umcodierung die Signaturerkennung zu umgehen. Durch die Protokolldecodierung können auch Angriffe auf Webserver mit Codierungen wie ASCII und Unicode erfolgreich erkannt werden. Durch statistische Anomalieanalyse können auch Angriffe und Portscans, die nicht mit der Signaturerkennung zu entdecken sind, erkannt werden. Portscans besitzen oftmals keine Signatur und können dann nur über die Häufigkeit ihres Auftretens analysiert werden.

5 Vergleich von IDS-Systemansätzen

In diesem Kapitel sollen die unterschiedlichen Systemansätze von Open-Source- und kommerziellen Systemen anhand von Tests veranschaulicht werden. Das heißt, es werden hier nicht Signaturansätze mit selbst lernenden Ansätzen miteinander verglichen, sondern zwei führende IDS-Systeme aus unterschiedlichen Lagern, um die Effektivität heutiger Systeme veranschaulichen zu können. Neben den Beschreibungen der Testumgebungen, sollte eine Kategorisierung möglicher Angriffe vorgenommen werden, um daraus gezielt die interessanten Angriffe für die Testdurchführung auszuwählen. Hierbei soll im Anschluss ausgewertet werden, ob die getesteten Produkte die Angriffe erkannt haben und welche Reaktionen der Systeme möglich sind.

5.1 Darstellung der Testumgebungen

Bei den Tests wurden ein kommerzielles IDS und ein Open-Source-Produkt gegenübergestellt. Als kommerzielles IDS wird das Dragon System von Enterasys verwendet, da es eindeutig zu den Marktführern gehört und die größte Funktionalität und Flexibilität bietet. Aus dem Open Source Bereich kommt Snort zum Einsatz, da es zu den am häufigsten genutzten Systemen zählt und durch die offene Entwicklung auch die Signaturen in großer Anzahl und hoher Aktualität frei verfügbar sind.

5.1.1 Dragon

Das IDS Dragon von Enterasys ist ein hybrides System, welches netz- und host-basierte Sensoren einsetzen kann. Der Hersteller hat für den Test ein ISO-Images zur Verfügung gestellt, dass in der Form auch auf den Appliances zum Einsatz kommt und beide Sensorarten integriert. Das System basiert auf Slackware Linux 8.0 und beinhaltet 1.788 Signaturen ausgestattet. Die derzeit aktuelle Signaturdatenbank enthält über 3.000 Einträge. Die Signaturen werden in verschiedenen Bibliotheken (.lib) zusammengefasst.

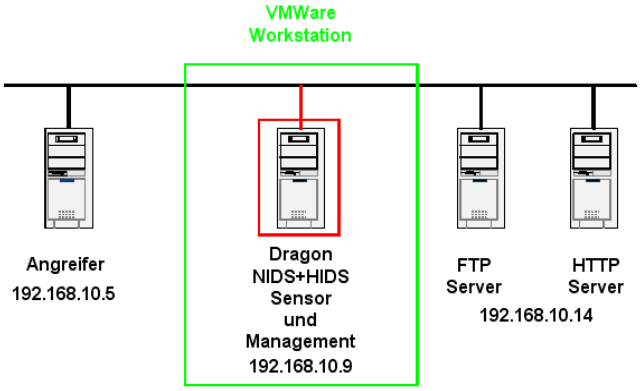

Abb. 5: Testaufbau Dragon

Die Testumgebung wurde wie in Abb. 5 aufgebaut. Der Angriffsrechner wird durch eine Knoppix Linux Version 3.2 integriert. Bei dem HTTP-Server handelt es sich um einen Apache Web-Server Version 1.3.23 und der FTP-Server ist ein Cerberus FTP Server Version 2.1. Diese beiden Server sind auf einer Windows XP Professional Plattform aufgesetzt und dienen als Angriffsziele.

Das Dragon System wurde in der Version 6.0.2 verwendet. Die Administration des IDS erfolgt über ein Webinterface per https. Hierbei werden Funktion wie Realtime Monitoring und Forensische Analyse unterstützt, die über vor eingestellte oder frei definierbare Filter, die gesammelten Daten des IDS aus der MySQL-Datenbank in unterschiedlichen Arten darstellen können. Die Signaturbibliotheken können einzeln angewählt werden und zeigen dann die zugehörigen Signaturen. Hier besteht die Möglichkeit eigene Signaturen zu definieren oder bestehende anzupassen.

5.1.2 Snort

Snort ist ein frei verfügbares NIDS, das hier in der Version 1.9.1 verwendet wurde. Es ist als Paket in der Suse 8.2 Distribution enthalten und wurde bei der Installation mit ausgewählt. Die Testumgebung ist identisch mit der Abb. 5.

Die Signaturen werden in Regeldateien (.rules) abgelegt. Die Dateien sind mit den Bibliotheken beim Dragon System vergleichbar und enthalten genauso mehrere Signaturen. Diese können nach Bedarf angepasst oder erweitert werden. Zurzeit sind im Internet ca. 1.800 Signaturen verfügbar.

Alle Konfigurationseinstellungen von Snort werden in der Datei etc/snort/snort.conf vorgenommen. Im Anhang ist die „snort.conf“ aus dem Test abgebildet. Hier werden die zu nutzenden Signaturen und die Ausgabedateien festgelegt.

Da Snort von sich aus nicht über eine grafische Benutzeroberfläche verfügt, müssen alle Einstellungen per Kommandozeile oder Editor geändert werden. Snort wird über das Kommando „/etc/init.d/snort restart“ nach den Änderungen neu gestartet.

5.1.3 Planung und Durchführung der Angriffe

Angriffe laufen unabhängig von der Kategorie immer nach dem gleichen Muster ab. Als Erstes beginnt der Angreifer allgemeine Informationen über das Opfersystem zusammeln. Die

Auswahl des Angriffsziels erfolgt je nach Absicht des Angreifers gezielt (bestimmte Firma oder Institution) oder durch Scannen eines beliebigen Adressbereiches mit anschließender Auswertung des Scans nach potentiellen Opfern. Die technischen Methoden der Informationssammlung unterteilen sich in TCP und UDP Portscans, sowie die Ausnutzung von Systemdiensten (z.B. finger, netstat, identd).

Die TCP Portscans gliedern sich in:

- Stealth Scan: Angreifer sendet ein FIN-Paket an den Zielport. Erhält er ein RST Paket zurück ist der Port inaktiv. Dieser Scan verwendet nicht den dreistufigen Verbindungsaufbau einer TCP-Verbindung und ist dadurch schwer zu erkennen.
- Half Open Scan: Hierbei werden nur die ersten zwei Stufen des TCP-Verbindungsaufbaus benutzt, um einen Dienst auf dem Port zu erkennen. Da der Verbindungsaufbau nach zwei Stufen nicht zustande kommt, wird er auch nicht mitprotokolliert und ist so schwer zu entdecken.
- Aktiver Portscan: Dieser Scan nutzt alle drei Stufen des TCP-Verbindungsaufbaus und kann leicht erkannt werden. Der Angreifer erhält eine Übersicht aller aktiven TCP-Ports.

Ein UDP Portscan bietet die Möglichkeit inaktive Ports zu erkennen, da der gescannte Rechner, wenn er auf dem angesprochenen Port keinen Dienst anbietet mit einer „ICMP Port Unreachable“-Meldung antwortet. Nun kann der Angreifer anhand der inaktiven Ports die aktiven ermitteln.

Sind genügend Informationen über das Zielsystem gesammelt, geht der Angriff in die nächste Phase. Hier hat der Angreifer die Möglichkeit über verschiedene Angriffsebenen und Schwachstellen in der Systemstruktur zum Ziel zu gelangen.

Die Angriffe auf die Testumgebung wurden mit Nessus (http://www.nessus.org) gemacht, da dieses Tool durch die Plugin-Technik, die größten und vollständigsten Angriffsarten bereitstellt. Zurzeit sind über 2.000 Plugins in der Nessus-Datenbank verfügbar, die sich auf 24 Obergruppen aufteilen.

Zur Testdurchführung wurden insgesamt 17 dieser Angriffsmuster exemplarisch ausgeführt. Die Einschränkung der Angriffe ist hier sinnvoll, da die Ergebnisdarstellung der Testsysteme von der Auswertbarkeit her sehr unterschiedlich ist. Das Ziel des Tests ist, die Unterschiede der Systeme in der Reaktion auf erkannte und nicht erkannte Angriffe darzustellen, wozu schon eine geringe Anzahl von Angriffen ausreicht.

Tab. 1: Ergebnisübersicht des Tests

Angriffsmuster	Dragon	Snort
	Erkennung	Erkennung
FTP CWD ~root	Ja	Nein
FTP site exec	Ja	Nein
FTP anonymous	Ja	Nein
Teardrop	Nein	Nein
SYN Scan	Nein	Ja
FTP Server type and version	Ja	Nein
SSH protocol versions supported	Nein	Nein
Apache /server-info accessible	Nein	Nein

Apache < 1.3.27	Nein	Nein
Pocsag password	Nein	Nein
BackOrifice	Ja	Ja
Portal of Doom	Ja	Nein
Systat	Nein	Nein
Netstat	Nein	Nein
Telnet	Nein	Nein
Check for Apache vulnerability	Nein	Nein
SMB Registry	Nein	Nein
Loginversuche per ssh	Ja	Nein

Die Erkennungsrate der Testsysteme war sehr unterschiedlich. Das lässt sich zum einen auf die Anzahl der Signaturen zurückführen, da nur Angriffe entdeckt werden konnten, die auch bereits bekannt waren. Zum anderen wurden aber auch Angriffe nicht erkannt, die in der Signaturdatenbank enthalten waren. Dies lässt darauf schließen, dass selbst eine korrekte Signatur nicht 100% vor einem Angriff schützen kann. Die für den Test verwendeten Angriffsmuster wurden willkürlich aus dem riesigen Portfolio ausgewählt, um keines der Systeme absichtlich stärker hervorzuheben.

6 Ausblick

Heutige IDS basieren überwiegend auf den Methoden der Signatur- und Anomalieerkennung, welche zurzeit einen hohen Konfigurationsaufwand und entsprechendes Know-how voraussetzen. Die weitere Entwicklung und Verfügbarkeit intelligenter IDS ist schwer abzuschätzen, da diese im universitären Bereich erforscht wird und noch in den Kinderschuhen steckt, da hier künstliche Intelligenz notwendig wird.

Die Tests zeigten jedoch wie unterschiedlich der Funktionsumfang von vorhandenen IDS-Produkten ist. Dabei ist das noch kein Negativkriterium, da es auch immer auf den gewünschten Einsatzumfang ankommt. Kleine IDS lassen sich mit Open-Source-Lösungen jedenfalls schneller und gezielter etablieren. Bei größeren Einsatzgebieten ist unabhängig vom Funktionsumfang ein Vorteil für kommerzielle Produkte zu sehen.

Bei der Nutzung von IDS ist die Entscheidung zwischen kommerziellen und Open Source Produkten vom gewünschten Einsatzumfang und den verfügbaren Ressourcen abhängig. Vor dem Einsatz sollte immer eine Evaluierung verschiedener Produkte stehen, um die größte Abdeckung der individuellen Vorgaben zu ermitteln.

Abschließend ist festzuhalten, dass Intrusion-Detection-Systeme nicht ein Gefühl von Sicherheit erzeugen sollen, sondern sie aktiv den Prozess der Sicherheit unterstützen.

Literatur

[ADEC02] A. Decker: Intrusion Detection Systeme; S&L GmbH; 2002

[DETK04] K.-O. Detken: Absicherung der internen Netzinfrastruktur durch IDS-Systeme: Schwachstellen, Angriffskategorien, IDS-Anforderungen und Funktionsweise; Deutscher Wirtschaftsdienst; 106. Ergänzungslieferung; Juni 2004; ISBN 3-87156-096-0; Köln 2004

[GOET03] Dirk Göttsche: Intrusion Detection und Response; Diplomarbeit an der Hochschule Bremen; Studiengang Medieninformatik; Bremen 2003

[KRIE03] K. Rieck: Analyse von neuronalen Netzen in host-basierten Intrusion Detection Systemen; Freie Universität Berlin; 2003

[MMIE03] M. Miettinen: Intrusion Detection Systeme; Ruhr-Universität Bochum; 2003

[MBLA02] M. Blatter: Seminar IT Sicherheit: Intrusion Detection Systeme (IDS); Universität Zürich; 2002

[SPEN03] R. Spenneberg: Intrusion Detection für Linux-Server; M+T Verlag; 2003