



2 Benötige ich ein IDS oder ein IPS und eine Firewall?

2.1 Einordnung der IDS und IPS in eine Sicherheitsstrategie

Dies Kapitel ist mit der Frage überschrieben: *Benötige ich ein IDS oder ein IPS und eine Firewall?* Üblicherweise wird diese Frage anders herum gestellt: *Warum benötige ich ein IDS, wenn bereits eine Firewall installiert wurde?* Dieses Kapitel soll die Möglichkeiten und Grenzen einer Firewall und eines IDS aufzeigen und vergleichen. Am Ende werden Sie erkennen, dass die beiden Systeme sich recht gut ergänzen. Man sollte sich nie nur auf eine Firewall zur Sicherung des Netzwerkes verlassen, sondern immer versuchen, beide Systeme in Kombination einzusetzen.

1. Welchen Schutz bietet eine Firewall?
2. Welchen zusätzlichen Schutz bietet ein IDS?

2.2 Welchen Schutz bietet eine Firewall?

Wenn über die Sicherheit von Computernetzen gesprochen wird, fällt üblicherweise als Erstes der Begriff einer Firewall. Was ist nun eine Firewall? Sicherlich wird allen Lesern der Begriff vertraut sein, dennoch soll er hier noch einmal kurz betrachtet werden. Weiterführende Literatur findet sich im Anhang.

2.2.1 Was ist eine Firewall?

Der Begriff *Firewall* wird zum Beispiel in der Autoindustrie verwendet, um die Wand zu kennzeichnen, die den Motorraum von den Insassen trennt. Sie stellt einen trennenden Schutz vor einem möglichen Motorbrand dar und muss in der Lage sein, diesem zu widerstehen.

Übertragen auf Computernetze stellt eine Firewall ein trennendes Glied zwischen mindestens zwei Netzen dar. Sie unterbindet den ungehinderten Austausch von Informationen zwischen den Rechnern der beiden Netze. Lediglich bestimmte Informationen dürfen ausgetauscht werden. Hierbei sollte darauf geachtet werden, dass keine zweite Verbindung zwischen den beiden Netzen existiert, die eine Umgehung der Firewall erlaubt. Die Firewall muss die einzige Verbindung sein. Im Auto wäre es auch fatal, wenn ein Motorbrand über einen anderen Weg in den Innenraum gelangen könnte. Besteht zwischen den beiden Netzen eine weitere Verbindung parallel zur Firewall, könnte der Austausch auch über diese Verbindung ohne die überwachende Funktion der Firewall erfolgen.

Es gibt nun verschiedene Techniken, eine Firewall zu implementieren. Die beiden am weitesten verbreiteten Techniken sind der Paketfilter und der Filter auf der Schicht des Applikationsprotokolls, häufig auch als Proxy bezeichnet. In vielen Fällen setzen Firewall-Systeme beide Techniken ein. Beispiele für Open-Source-Produkte, die diese Techniken implementieren, sind *ipchains* und *iptables* (*netfilter*) als Paketfilter und *squid* und *httpf* als Proxy. Beide Ansätze unterscheiden sich stark in ihrer Performanz und in ihren Filtermöglichkeiten.

Der Paketfilter ist in der Lage (der Name ist Programm), Pakete zu filtern. Dazu betrachtet der Paketfilter die Header der IP-Pakete. Die meisten Paketfilter können den IP-Header und, wenn vorhanden, auch den TCP-, UDP- und ICMP-Header lesen und verarbeiten. Bei diesen Informationen handelt es sich um die IP-Adressen, das IP-Protokoll (TCP, UDP, ICMP, IGMP etc.), wenn vorhanden, die TCP- und UDP-Ports und den ICMP-Code (siehe auch Anhang A, »Netzwerkgrundlagen«). Weitere Informationen sind der Fragmentierungszustand, Länge des Paketes, TTL- und TOS-Werte etc. Damit können Regeln definiert werden, die nur Pakete zu einem Webserver durchlassen, wenn sie an den Port 80 gerichtet sind. Ein Paketfilter ist normalerweise nicht in der Lage, den Inhalt der Pakete zu betrachten. Er kann nicht feststellen, ob diese Pakete tatsächlich eine HTTP-Anfrage enthalten und ob das HTTP-Protokoll fehlerfrei verwendet wird. Der Paketfilter arbeitet meist im Kernel des Betriebssystems auf den Schichten 3 und 4 des OSI-Modells. Er hat normalerweise keinerlei Zugriff auf die Applikationsdaten. Die zu filternden Pakete müssen nicht an eine Applikation im Userspace weitergegeben werden. Dadurch kann der Paketfilter sehr schnell arbeiten.

Es existieren zwei verschiedene Varianten eines Paketfilters: einfache zustandslose Paketfilter und zustandsorientierte Paketfilter, so genannte Stateful Inspection Packetfilter.

Ein zustandsloser Paketfilter (z.B. *ipchains*¹) ist in der Lage, einzelne Pakete zu filtern. Er ist jedoch nicht in der Lage, einen Zusammenhang zwischen verschiedenen

¹ Der Paketfilter *ipchains* ist auch in der Lage, Pakete zu maskieren. Hierbei wird die Absender-IP-Adresse in den Paketen ausgetauscht. Damit die Antwortpakete später den korrekten Absendern zugeordnet werden können, muss *ipchains* eine Zustandstabelle pflegen und stellt in dem Moment eine Art zustandsorientierten Paketfilter dar. Dies trifft jedoch nur für die maskierten Verbindungen zu!

Paketen herzustellen. Bei einem Paket, welches den Paketfilter von außen erreicht, ist er nicht in der Lage festzustellen, ob es sich um eine neue Netzwerkverbindung handelt. Das Paket könnte auch eine Antwort auf ein vorher von innen gesendetes Paket darstellen. Ein zustandsloser Paketfilter muss daher alle theoretisch möglichen Antwortpakete von außen erlauben, um eine reibungslose Kommunikation zu unterstützen.

Ein zustandsorientierter Paketfilter (z.B. *iptables*²) prüft bei jeder neuen Verbindung, ob sie entsprechend den Regeln erlaubt ist. Er erzeugt dann einen Eintrag in seiner Zustandstabelle. Anschließend können weitere Pakete dieser Verbindung automatisch zugelassen werden. Es müssen nicht mehr alle denkbar möglichen Antwortpakete erlaubt werden. Der Paketfilter erlaubt nur noch diejenigen Pakete, die zu vorher aufgebauten und entsprechend den Regeln authentifizierten Verbindungen gehören. Dies erhöht die Sicherheit des Paketfilters. Dies ist gewissermaßen ein Verbindungsfilter.

Viele dieser Paketfilter unterstützen die »Stateful Inspection«. Einige Protokolle weichen von dem üblichen Standard einer IP-Verbindung zwischen einem Client und einem Server ab. Normalerweise kontaktiert der Client von einem hohen Port (Port ≥ 1024) den Server auf einem privilegierten Port (Port < 1024). Über diese Verbindung werden **alle** Informationen ausgetauscht.

Der bekannteste Vertreter der Protokolle, die sich nicht an diesen Standard halten, ist FTP. Der Client verbindet sich von einem hohen Port auf den Port 21 (*ftp control port*) auf dem Server. Diese Verbindung wird verwendet, um die Informationen zur Anmeldung und die weiteren Befehle zu übertragen. Sobald der Server Daten auf den Client übertragen muss (Verzeichnisisinhalt oder Datei), öffnet der Server eine Verbindung von Port 20 (*ftp data port*) auf einen anderen hohen Port des Clients. Dies bezeichnet man als aktives FTP, da der Server eine aktive Rolle einnimmt. Der zu verwendende hohe Port wird zuvor vom Client an den Server in einem so genannten Port-Kommando übertragen. Stateful Inspection bedeutet, dass die Firewall in der Lage ist, das Port-Kommando zu erkennen und anschließend spezifisch die aktive FTP-Verbindung zu erlauben. Eine zustandslose Firewall kann diesen Zusammenhang nicht herstellen und muss daher grundsätzlich Pakete von jedem beliebigen Rechner und Port 20 auf jeden hohen Port eines Clients zulassen, um aktives FTP zu unterstützen.

Die Stateful Inspection stellt die einzige Ausnahme dar, bei der ein Paketfilter intelligent auf den Inhalt des Paketes zugreift. Dies kann auch für die Applikationsprotokolle Internet Relay Chat (IRC), Point to Point Tunneling Protocol (PPTP), H.323, ICMP und andere erfolgen. Ansonsten betrachtet jedoch ein Paketfilter nur die Header der Pakete. Er ist mehr oder weniger ein intelligenter Router!

² Der Paketfilter *iptables* ist nur dann ein zustandsorientierter Paketfilter, wenn das *ip_conntrack.o*-Modul geladen wurde. Dies erfolgt automatisch, wenn der Paketfilter ein Network Address Translation (NAT) durchführt. Zusätzlich müssen jedoch diese Funktionalitäten auch von den Regeln genutzt werden. Für die »Stateful Inspection« müssen ebenfalls weitere Module geladen werden.

Ein Paketfilter ist also in der Lage, den Aufbau der Verbindungen zwischen den Endpunkten Client und Server zu regeln. Es wird lediglich eine durchgehende Verbindung zwischen dem Client und dem Server aufgebaut.

Ein Filter auf den Schichten 5 bis 7 des OSI-Modells (Proxy) betrachtet die Pakete nicht. Ein Proxy arbeitet im Userspace und bekommt vom Betriebssystem die Pakete zu einem Datenstrom aufbereitet. Diesen Datenstrom kann nun der Proxy verarbeiten. Dabei ist er theoretisch in der Lage, auf sämtliche Informationen des Datenstroms zuzugreifen, diesen zu untersuchen und zu verändern.

Der Proxy fungiert hierbei als ein Mann in der Mitte (Man-in-the-Middle). Der Proxy nimmt an Stelle des Servers die Anfragen des Clients als Datenstrom entgegen. Er verarbeitet und filtert diese Anfragen und leitet sie anschließend als Client an den echten Server weiter. Dieser sendet seine Antwort an den Proxy, der erneut in der Lage ist, die Daten zu analysieren und zu filtern. Schließlich wird der Proxy die Daten dem echten Client zustellen.

Ein Proxy erlaubt nicht den Aufbau von Netzwerkverbindungen zwischen dem Client und dem Server. In Wirklichkeit werden zwei Netzwerkverbindungen aufgebaut: Client-Proxy und Proxy-Server. Es existiert kein Paketaustausch zwischen dem Client und dem Server!

Das größte Problem bei der Implementierung einer Firewall rein auf der Basis von Proxies stellen die Applikationsprotokolle selbst dar. Diese weisen keine gemeinsame Grundlage auf. Sie unterscheiden sich in ihren Befehlen, ihrer Syntax, Sprache und Funktionalität sehr stark. Daher ist es erforderlich, für jedes Applikationsprotokoll einen eigenen Proxy zu entwickeln, der in der Lage ist, dieses Protokoll zu verstehen, zu filtern und weiterzuleiten. So stellt das HTTP-Applikationsprotokoll andere Anforderungen an einen Proxy als das POP3-E-Mail-Protokoll.

Kommerzielle Firewall-Lösungen auf der Basis eines Proxys wie auch Open-Source-Lösungen sind daher nicht in der Lage, sämtliche Protokolle nativ zu unterstützen. In solchen Fällen kommen häufig weitere generische Proxies zum Einsatz, die lediglich die Verbindung auf einem Port entgegennehmen und eine neue Verbindung aufbauen. Hierbei ist aber keine Analyse oder Filterung des Datenstroms möglich.

Ein Proxy hat durch seine Sicht auf den Datenstrom wesentlich mehr Möglichkeiten als ein einfacher Paketfilter. Dies soll am Beispiel eines HTTP-Proxys für den Internetzugriff beschrieben werden.

- Der Proxy kann in Abhängigkeit von der URL filtern. Ein Paketfilter sieht lediglich die IP-Adressen der Kommunikationspartner. Heute werden häufig viele verschiedene Websites auf einem Rechner gehostet. Ein Paketfilter ist nicht in der Lage, zwischen diesen Sites oder verschiedenen Bereichen einer Site zu unterscheiden.
- Der Proxy kann in Abhängigkeit vom Inhalt der Datei filtern. Ein Proxy erkennt den Beginn und das Ende der Dateien. Dadurch kann er den Dateityp erkennen und überprüfen und den Inhalt auf bestimmte Eigenschaften oder Viren testen.

Bei einem Bild kann zum Beispiel geprüft werden, ob es sich tatsächlich um ein Bild handelt, oder ob es doch eine ausführbare Datei ist.

- Ein Proxy kann den Datei-Inhalt verändern. Dies ist zum Beispiel sinnvoll bei aktiven Inhalten von Webseiten. Ein Proxy kann JavaScript-Inhalte filtern und so modifizieren, dass sie vom Client nicht ausgeführt werden.

Dies sind Fähigkeiten, die ein normaler Paketfilter nicht zur Verfügung stellen kann. Der Proxy benötigt jedoch aufgrund der fortgeschrittenen Möglichkeiten wesentlich mehr Ressourcen als ein Paketfilter. Speziell ein Virenskan ist sehr zeitaufwändig.

Diese Fähigkeiten stehen jedoch nicht bei allen Proxies zur Verfügung. Besonders der generische Proxy ist nicht in der Lage, derartige Filterfunktionen zur Verfügung zu stellen. In vielen Umgebungen ist die Implementierung fortgeschrittener Filterfunktionen durch einen Proxy nicht möglich, da die Anforderungen an die Bandbreite der Netzwerkverbindung nur von einem Paketfilter erfüllt werden können.

Eine Firewall ist also in der Lage, die Kommunikation einzuschränken und nur in einer bestimmten Richtung bestimmte Inhalte zu erlauben. Dennoch kann eine Firewall nur im Rahmen der Richtlinien ihre Filterfunktionen wahrnehmen. Erlaubt eine Firewall den Zugriff auf JavaScript-Inhalte einzuschränken, so besteht meist nicht die Möglichkeit, zwischen gutartigem und bösartigem JavaScript zu unterscheiden. Ähnliche Einschränkungen gelten für Java und andere aktive Inhalte.

Ein weiteres Problem bei einer Sicherheitsstruktur, die lediglich eine Firewall einsetzt, taucht auf, wenn diese Firewall den Zugriff auf den Webserver erlaubt und überwacht. Wird nun eine neue Sicherheitslücke in der Software entdeckt, die als Webserver eingesetzt wird, so besteht die Gefahr, dass die Firewall diesen Angriff nicht erkennt, sondern zulässt. Als Beispiel mag der *Directory Traversal*-Angriff auf den Microsoft Internet Information Server 3.0, 4.0 und 5.0 dienen (<http://www.kb.cert.org/vuls/id/111677>). Hier bestand die Möglichkeit, durch die Verwendung von Unicode-Zeichen auf Bereiche zuzugreifen, die üblicherweise gesperrt sind. So bestand auch die Möglichkeit, auf dem Webserver beliebige Programme auszuführen. Ein Paketfilter kann diesen Angriff nicht von einem normalen Zugriff unterscheiden. Auch ein Proxy wird den Angriff nicht erkennen, da ein Unicode-Zugriff grundsätzlich erlaubt ist. Weitere Beispiele für Sicherheitslücken, die von einer Firewall meist nicht erkannt werden können, stellen die Bufferoverflows dar. Hierbei wird ein Programmierfehler durch den Angreifer ausgenutzt. So wurde zum Beispiel von der Firma UssrBack ein Bufferoverflow in Microsoft Outlook festgestellt (<http://www.ussrback.com/labs50.html>). Hierbei ist es möglich, bei einem zu langen Datumsfeld in der E-Mail den Outlook E-Mail-Client zum Absturz zu bringen. Eine Firewall wird diese E-Mail meist passieren lassen. Selbst ein Viruscheck der E-Mail durch das Firewall-System wird nicht den Bufferoverflow erkennen. Eine Erklärung der Funktionsweise des Bufferoverflows erfolgt im Exkurs »Was ist ein Bufferoverflow?« auf S. 285.

Häufig befindet sich der »Angreifer« bereits im Netzwerk. Es kann sich dabei um einen Wartungstechniker handeln, der prüfen möchte, ob Rechnersysteme der Konkurrenz eingesetzt werden. Es kann jedoch auch der eigene Angestellte sein, der in der

aktuellen Ausgabe einer Computer-Zeitschrift von einem neuen Hackerwerkzeug gelesen hat und dies direkt ausprobieren möchte. Eine Firewall ist nur in der Lage, den Verkehr zu analysieren und einzuschränken, der über sie ausgetauscht wird. Sind die Angriffe des Angestellten gegen eigene Rechner gerichtet, so sieht die Firewall den Angriff nicht und kann ihn auch nicht erkennen. Sind die Angriffe nach außen gerichtet, so bestehen meist auf den Firewall-Systemen Regelsätze, die Verbindungen von innen nach außen grundsätzlich zulassen und den Angriff ermöglichen.

Hier sind zusätzliche Maßnahmen erforderlich. Diese sollten ein Intrusion-Detection-System implementieren. Weitere Maßnahmen, die teilweise in anderen Kapiteln in diesem Buch beschrieben werden, sind die Erstellung von Verhaltensrichtlinien in Form von Benutzerordnungen (Kapitel 17, »Datenschutz-Aspekte in einem Unternehmensnetzwerk«) und die Installation von Virenschnüchern.

Diese Probleme werden häufig noch deutlicher, wenn zusätzlich virtuelle private Netze geschaffen werden, die einen Zugriff von außen an der Firewall vorbei erlauben. Dies kann zum Beispiel der Zugriff eines Abteilungsleiters von seinem Heimbüro am Wochenende sein. Diese Zugriffe werden häufig so konfiguriert, dass der Anwender einen fast uneingeschränkten Zugriff auf die internen Strukturen erhält. Er soll von zu Hause genauso arbeiten können wie im Betrieb. Ist dieser Rechner gleichzeitig der Rechner, mit dem die böse Tochter (meist ist es jedoch der Sohn ; -)) im Internet surft und chattet, so besteht die Gefahr, dass über diese Verbindungen Viren und Trojaner den Zutritt in das interne Netz erhalten. Möglicherweise verwendet der Abteilungsleiter auch ein kleines Netzwerk zu Hause und der Sohn surft gleichzeitig im Netz. Dann besteht vielleicht sogar aus dem Internet in dem Moment eine Verbindung über das VPN in das Unternehmensnetzwerk! Diese Zugriffe werden von der Firewall bewusst nicht gefiltert! Hier soll ein ungehinderter Austausch möglich sein!

2.3 Welchen Schutz bietet darüber hinaus ein IDS?

Nachdem die verschiedenen Firewall-Technologien beleuchtet und ihre Möglichkeiten und Grenzen beim Schutz eines Netzwerkes erläutert wurden, sollen nun die Intrusion-Detection-Systeme, ihre Funktionen und Grenzen betrachtet werden. Am Ende dieses Kapitels soll deutlich geworden sein, wie ein IDS eine Sicherheitsstruktur erweitern und verbessern kann.

Es sollte bisher bereits klar geworden sein, dass eine Firewall alleine nicht in der Lage ist, eine ausreichende Sicherheitsstruktur zur Verfügung zu stellen. Hier ist ein Intrusion-Detection-System in der Lage, eine Firewall zu unterstützen.

Im Allgemeinen teilt man die Intrusion-Detection-Systeme in zwei Gruppen ein. Es werden die rechnerbasierten Intrusion-Detection-Systeme von den netzwerkbasierten Systemen unterschieden. Die rechnerbasierten Systeme bezeichnet man auch als Host Intrusion Detection System (HIDS) im Gegensatz zum Network Intrusion Detection System (NIDS).

Diese Systeme setzen nun unterschiedliche Methoden ein, um eine Intrusion oder Misuse zu erkennen. Die verschiedenen Methoden und ihre Ergebnisse sollen nun für HIDS und NIDS getrennt untersucht werden.

2.3.1 Host Intrusion Detection System

Als Host Intrusion Detection System (HIDS) werden Systeme bezeichnet, die Daten analysieren, welche auf einem Rechner zur Verfügung stehen. Hierbei handelt es sich um Daten, die vom Betriebssystem oder von den Anwendungen erzeugt werden. Dies können Protokolle sein, die vom Betriebssystem oder den Anwendungen bereits zur Verfügung gestellt werden; oder das Intrusion-Detection-System erzeugt zusätzliche Ereignisprotokolle oder wertet Daten des Betriebssystems direkt aus. Eine besondere Form des HIDS stellen die Systeme dar, welche die Integrität des Systems überprüfen. Diese Systeme werden auch als System Integrity Verifier (SIV) oder File Integrity Assessment (FIA) bezeichnet.

Host-Intrusion-Detection-Systeme sind insbesondere in der Lage, eine missbräuchliche Verwendung der Systeme durch Insider zu erkennen. Hier ist eine Firewall meist überfordert. Da das HIDS die Systeme selbst überwacht, können hier möglicherweise unerlaubte Tätigkeiten erkannt werden.

Der Nachteil einer HIDS-Implementation ist die recht umständliche und komplizierte Überwachung einer großen Anzahl von Rechnern. Das HIDS muss in irgendeiner Form auf jedem zu überwachenden Rechner installiert werden. Dies kann in einer verteilten Struktur aus Agenten und einer Management-Zentrale erfolgen. Dennoch ist die Verwaltung und Administration immer noch recht aufwändig.

HIDS: Erkennbare Angriffe

Ein HIDS ist nun in der Lage, insbesondere Angriffe von innen zu erkennen. Natürlich werden auch Angriffe von außen von einem HIDS erkannt. Einige klassische Beispiele sollen nun gegeben werden, um zu verdeutlichen, um welche Angriffe es sich hierbei handeln kann.

- Wartungstechniker oder externe Berater erhalten häufig für die Dauer ihrer Tätigkeit einen Zugang zum Rechnersystem, auf dem sie arbeiten müssen. Dieser Zugang ist meist privilegiert, damit sämtliche erforderlichen Änderungen durchgeführt werden können. Unter Linux wird ihnen daher häufig das *root*-Recht übertragen. Hiermit sind sie in der Lage, den gesamten Rechner zu administrieren. Während dieser Tätigkeit besteht für sie die Möglichkeit, eine Hintertür für einen späteren Zugang auf den Rechner zu öffnen. Viel zu häufig wird auch versäumt, den Zugang des externen Beraters nach der Tätigkeit wieder zu schließen, so dass dieser in der Lage ist, noch nach Monaten auf das System zuzugreifen.
- Die Konten ehemaliger Angestellter und Administratoren werden häufig nicht gelöscht oder deaktiviert. Diese Personen besitzen meist noch ihre alten Anmeldeinformationen und können sich noch Wochen und Monate später an den Systeme

men anmelden. Dies ist häufig umso prekärer, wenn diese Personen inzwischen bei der Konkurrenz arbeiten.

- Einige Administratoren und Angestellte versuchen, auch für den speziellen Fall ihres Wechsels zu einem Konkurrenzunternehmen, sich eine Hintertür für einen späteren Zugang zu öffnen. Über diesen Zugang besteht die Möglichkeit, Unternehmensgeheimnisse zu erhalten.
- Angestellte modifizieren kritische Daten in Berichten oder in Personalakten. Hierdurch wird die Integrität sämtlicher Daten in Frage gestellt, da eine Erkennung dieser Modifikationen meist erst sehr spät erfolgt.
- Häufig versuchen Angestellte Zugriff auf die Personalakten oder andere vertrauliche Informationen über ihre direkten Vorgesetzten oder Konkurrenten in der eigenen Firma zu erhalten. Diese Informationen werden dann eingesetzt, um diese Personen zu mobben oder zu übervorteilen.
- Modifikationen der Website (ein so genanntes Defacement) sind ein allseits beliebtes Mittel, mit dem eine Firma oder eine Person in Misskredit gebracht werden soll.
- Die Ausführung von Administrationsbefehlen durch eine unbekannte Person stellt sicherlich auch einen möglichen Angriff dar, der von einem HIDS erkannt werden kann.
- Die Installation von Trojanischen Pferden oder Rootkits (siehe Anhang C, »Rootkits«) stellt ebenfalls eine Tatsache dar, die vom HIDS erkannt und gemeldet werden sollte.

Sämtliche aufgelisteten Angriffe sollen von einem HIDS erkannt werden. Anschließend soll das HIDS die zuständigen Personen alarmieren oder sogar selbst eine Gegenmaßnahme ergreifen. So besteht die Möglichkeit, bei einem erkannten Website-Defacement den Webserver so zu konfigurieren, dass keine Seiten mehr angeboten werden. Getreu dem Motto: besser keine Seite, als eine mit pornographischem Inhalt. Erkennt das HIDS die Ausführung eines Administrationsbefehls zum Beispiel zur Modifikation von Firewall-Regeln durch einen unbekanntem Benutzer, so kann es direkt diesen Benutzer aussperren und abmelden.

Im Folgenden sollen die unterschiedlichen Technologien betrachtet werden.

HIDS: Technologien

Die unterschiedlichen verfügbaren Host-Intrusion-Detection-Systeme verfolgen unterschiedliche Ziele und setzen dazu auch unterschiedliche Technologien ein. Diese Technologien sollen kurz vorgestellt und ihre Vor- und Nachteile aufgezählt werden. Viele kommerziell verfügbaren HIDS setzen eine Kombination dieser Techniken ein. Open-Source-Lösungen konzentrieren sich meist auf eine Technologie. Hier ist es erforderlich, eine Kombination der verfügbaren Lösungen einzusetzen.

- **Protokollanalyse.** Die Protokollanalyse stellt die einfachste und ursprüngliche Form der Intrusion Detection dar. Hierbei überwacht das IDS die vom Betriebs-

system und den Anwendungen zur Verfügung gestellten Protokolldateien. Viele HIDS erlauben die Definition einer Positiv-Liste. Sämtliche Meldungen, die mit dieser Liste übereinstimmen, lösen eine Alarmierung aus. Dabei können jedoch leicht wichtige Meldungen übersehen werden. Sinnvoller ist daher die Definition einer Negativ-Liste. Sämtliche Meldungen, die mit dieser Liste übereinstimmen, werden ignoriert. Alle weiteren, unbekannteren Meldungen führen zu einer Alarmierung. Intelligente Protokollanalytoren sind sogar in der Lage, diese Meldungen weiter aufzuarbeiten und zusammenzufassen. Damit braucht der Administrator nicht mehrere hundert oder tausend Zeilen Protokollmeldungen zu lesen, sondern erhält die Meldungen in übersichtlichen Berichten aufgearbeitet.

- **Integritätstest.** Eine große Anzahl von HIDS versucht eine Intrusion zu erkennen, indem sie konstant das System und seine Dateien auf ihre Integrität prüfen. Hierzu wurde nach der Installation und Konfiguration vom System ein Schnappschuss geschossen. Im Weiteren vergleicht das HIDS in regelmäßigen Abständen den Zustand des Systems mit diesem Schnappschuss. Hierbei werden häufig nur die Eigenschaften der Datei und ihre Prüfsummen verglichen, um Speicherplatz für den Schnappschuss zu sparen. Diese Intrusion-Detection-Systeme werden daher auch System Integrity Verifier (SIV) oder File Integrity Assessment (FIA) genannt.
- **Echtzeitanalyse von Systemaufrufen und Dateizugriffen.** Eine letzte Gruppe von HIDS führt eine Echtzeitanalyse sämtlicher System- und Dateizugriffe durch. Dies ist die aufwändigste Variante eines HIDS. Das HIDS greift hierzu meist auf Betriebssystemebene ein. Unter Linux wird das HIDS meist als Kernel-Modul implementiert, welches anschließend jeden Zugriff protokolliert und überwachen kann. Es besteht dann die Möglichkeit, den Zugriff auf bestimmte Privilegien und Dateien zu überwachen und auch zu verweigern. So kann ein derartiges System erkennen, wenn die IP-Adresse des Rechners oder die Regeln einer Firewall modifiziert werden sollen. Diese Modifikationen können vom System in Echtzeit gemeldet werden und das System ist sogar in der Lage, die Modifikation zu unterbinden.

Ein großes Problem bei all diesen Technologien sind Falschmeldungen durch das HIDS. Hierbei gibt es zwei grundsätzliche Möglichkeiten einer Falschmeldung:

- **Falsch-positiv:** So wird eine Alarmierung durch das IDS bezeichnet, die in Wirklichkeit keine Meldung darstellt. Es handelt sich um einen Fehlalarm. Dies ist zunächst recht harmlos, wenn es ein seltenes Ereignis darstellt. Kommen diese Fehlalarme jedoch häufig vor, ohne dass sie abgestellt werden, so führt dies über kurz oder lang zu einem Desinteresse und ein echter Alarm wird nicht ernst genommen.
- **Falsch-negativ:** So wird eine fehlende Alarmierung durch das IDS bezeichnet. Das IDS hätte einen Alarm auslösen sollen, da ein Einbruch stattgefunden hat. Diese fehlenden Meldungen stellen eine große Gefahr dar, wenn das IDS nicht richtig konfiguriert wurde. Der Administrator verlässt sich zu einem gewissen

Maß auf die korrekte Funktion des IDS. In diesem Fall versagt jedoch das IDS. Leider erfährt dies zunächst niemand.

Falschmeldungen können nur durch sorgfältige Konfiguration und aufmerksames Studium des IDS vermieden werden. Dieses Buch versucht das notwendige Wissen zu vermitteln, um dies zu erreichen.

2.3.2 Network-Intrusion-Detection-System

Ein Network-Intrusion-Detection-System bezieht seine Daten aus dem Netzwerk. Hierbei untersucht das System die Netzwerkpakete. Üblicherweise erhält es die Netzwerkpakete durch einen Netzwerk-»Sniffer«. Einige Systeme überwachen jedoch nur die Protokolle von Routern und Switches. Diese Systeme sind nicht so mächtig, wie NIDS, die einen eigenen Sniffer enthalten.

Network-Intrusion-Detection-Systeme sind insbesondere geeignet, um Angriffe und Einbrüche von außen zu erkennen. Sie können aber natürlich auch Insider, die weitere Rechner anzugreifen versuchen, erkennen. Hier kann eine Firewall nur wenig Schutz bieten. Ein NIDS ist in der Lage, diese Angriffe und Einbrüche dennoch zu erkennen. Häufig ist ein NIDS auch in der Lage zu erkennen, dass ein Angreifer von außen sich bereits im Netzwerk befindet und dort weitere Rechner anzugreifen versucht.

Die Installation und Administration von Network-Intrusion-Detection-Systemen ist meist nicht so aufwändig wie die Administration von HIDS. Es genügt oft die Installation eines NIDS-Sensors pro überwachtem Netzwerk. Netzwerke können meist zentral überwacht und verwaltet werden.

NIDS: Erkennbare Angriffe

- **Denial-of-Service-Angriffe mit Bufferoverflows.** Diese Bufferoverflow-Angriffe verwenden meist Pakete mit sehr charakteristischen Eigenschaften. Hierbei handelt es sich zum Beispiel um den NOP Sled (siehe Exkurs »Was ist ein Bufferoverflow?« auf S. 285).
- **Ungültige Pakete.** Angriffe wie der TearDrop, Bonk oder Ping of Death verwenden ungültige Pakete. Diese Pakete dürfen entsprechend der Protokollspezifikation nicht existieren. Sie sind grundsätzlich künstlich, zum Zweck des Angriffes, erzeugt worden.
- **Angriffe auf Applikationsprotokoll-Schicht.** Viele Applikationen sind lediglich in der Lage, bei einer korrekten Anwendung des Applikationsprotokolls vorhersehbar zu reagieren. Angreifer nutzen dieses aus und verändern das Applikationsprotokoll, damit unerwünschte Funktionen genutzt werden können (siehe z.B. IIS Data Stream-Zugriff, siehe Abschnitt »Data Stream-Zugriff« auf S. 283).
- **Zugriff auf vertrauliche Daten.** Ziel vieler Angriffe ist es, vertrauliche Informationen, die den Zugang zu weiteren Daten ermöglichen, zu erhalten. Das häufigs-

te Ziel unter UNIX sind die Kennwortdatenbanken `/etc/passwd` und `/etc/shadow`. Ein NIDS kann sämtliche Pakete auf diese Zeichenketten hin untersuchen.

- **Informationsdiebstahl.** Ein weiteres Ziel eines Angriffes ist häufig der Diebstahl von Informationen. Werden diese Informationen unverschlüsselt übertragen, so kann ein NIDS diese Informationen an Schlüsselwörtern erkennen.
- **Angriffe gegen die Firewall oder das NIDS.** Viele Angriffe sind auch gegen die Firewall und das NIDS selbst gerichtet. Diese Angriffe können ebenfalls von einem fortgeschrittenen NIDS erkannt werden. Hierzu muss das NIDS in der Lage sein, Angriffe mit fragmentierten Paketen zu erkennen. Weitere Angriffe auf die Zustandsüberwachung des NIDS sind mit modifizierten TCP-Paketen möglich. Auch die sollte ein NIDS erkennen.
- **Distributed Denial of Service (dDoS).** NIDS sind sicherlich kein Schutz vor einem dDoS, dennoch sind sie in der Lage, diesen häufig zu erkennen und den Verdacht, dass ein dDoS vorliegt, zu bestätigen.
- **Spoofing-Angriffe.** Häufig fälschen Angreifer ihre Herkunft auf MAC-, IP- oder DNS-Ebene. Ein NIDS sollte diese Spoofing-Angriffe so weit wie möglich erkennen können.
- **Portscans.** Ein Portscan sollte vom NIDS erkannt werden. Dies ist besonders der Fall, wenn es sich um intelligente langsame Portscans handelt, die pro Stunde oder gar pro Tag nur einen oder wenige Ports prüfen.

Alle aufgeführten Angriffe sollten von einem NIDS erkannt werden. Diese Angriffe müssen zu einer Alarmierung der verantwortlichen Personen führen wie auch möglicherweise eine direkte Reaktion des NIDS hervorrufen.

Die Konfiguration eines NIDS ist meist sehr aufwändig. Die Pflege erfordert mehr Zeit als die Pflege eines einzelnen HIDS. Auf einem NIDS-System müssen stets die neuesten Angriffe eingepflegt werden, so dass das NIDS auch in der Lage ist, diese zu erkennen.

Eine automatische Reaktion durch das NIDS ist in vielen Fällen zunächst als problematisch einzustufen, da ein Angreifer diese Funktionalität unter Umständen nutzen kann. Wenn das NIDS jegliche Kommunikation zu den vermeintlichen Angreifern unterbricht, kann der Angreifer gezielt einen Angriff von einem wichtigen DNS-Server (Root-DNS-Server) oder E-Mail-Server vortäuschen. Das NIDS führt dann möglicherweise den Denial-of-Service-(DoS-)Angriff aus.

NIDS: Technologien

NIDS-Systeme verwenden eine Vielzahl von Technologien. Die wichtigsten Technologien werden kurz vorgestellt.

- **Signatur-Erkennung.** Die meisten NIDS arbeiten zunächst mit einer einfachen Signatur-Erkennung. Hierzu besitzen sie eine große Datenbank, in der die Signaturen oder Fingerabdrücke sämtlicher bekannter Netzwerkangriffe gespeichert sind. Nun vergleicht das NIDS jedes Paket mit dieser Datenbank und prüft, ob ei-

ne der Signaturen auf das Paket zutrifft. Ist dies der Fall, so wird das NIDS einen Alarm auslösen.

- **Zustandsorientierte Signatur-Erkennung.** Die ersten einfachen NIDS wiesen lediglich eine einfache Signatur-Erkennung auf. Ein Angreifer kann bei Kenntnis der Datenbank spezifisch Pakete erzeugen, die einen Alarm auslösen. Erzeugt er diese Pakete in genügender Anzahl, führt dies meist zu einer Überlastung des NIDS, da jedes Paket vom NIDS als Angriff gewertet wird. In Wirklichkeit handelt es sich jedoch nicht um einen Angriff. Alle zusätzlichen Pakete, die für den Verbindungsaufbau erforderlich sind, fehlen. Das Angriffsziel wird dieses Paket verwerfen. Die zustandsorientierte Signatur-Erkennung betrachtet nur Pakete, die tatsächlich in einer aufgebauten Verbindung übertragen werden. Alle weiteren Pakete werden vom NIDS genau so verworfen, wie sie vom Zielsystem verworfen werden würden. Das NIDS pflegt hierzu eine Tabelle mit sämtlichen bekannten Verbindungen.
- **Protokolldekodierung.** Eine reine Angriffserkennung auf Signaturbasis ist meist zum Scheitern verurteilt. Moderne Applikationsprotokolle erlauben häufig verschiedenste Kodierungen derselben Anfrage. Ein Angreifer kann daher eine Signatur-Erkennung leicht unterlaufen, wenn er denselben Angriff anders kodiert. Ein fortgeschrittenes NIDS ist daher in der Lage, bevor der Vergleich der Signaturen durchgeführt wird, den Inhalt des Datenstroms zu normalisieren. Dies ermöglicht zum Beispiel die Erkennung von Angriffen auf Webservern mit unterschiedlichen Kodierungen wie Base64, ASCII und Unicode.
- **Statistische Anomalie-Analysen.** Viele Portscans und Angriffe können nicht mit einer Signatur-Analyse erkannt werden. Ein Portscan weist meist keine besondere Signatur auf, sondern ist nur durch die Häufigkeit zu erkennen. Neue bisher unbekannte Angriffe können nicht erkannt werden, da bisher noch keine Signatur in der Datenbank für diese Angriffe vorhanden ist. Eine statistische Analyse des Netzwerkverkehrs ist unter Umständen in der Lage, diese Vorgänge zu erkennen, wenn es sich um äußerst ungewöhnliche Pakete handelt.
- **Heuristische Analysen.** Ein einfacher 1:1-Vergleich der Signaturen mit den zu analysierenden Paketen ist meist sehr langsam. Heuristische Methoden können diese Analyse stark beschleunigen, ohne dass die Genauigkeit und Trefferrhäufigkeit leidet. Ein Algorithmus, der derartige Heuristics verwenden kann, ist Bayer-Moore (http://philby.ucsd.edu/~cse291_IDVA/papers/coit,staniford,mcalerney.-towards_faster_string_matching_for_intrusion_detection.pdf).
- **Reaktion.** Ein Netzwerk-Intrusion-Detection-System kann unterschiedlich auf einen erkannten Angriff reagieren. Die Reaktionen reichen von einem einfachen Protokolleintrag bis hin zur kompletten Sperrung aller weiteren Verbindungen des angreifenden Rechners durch eine Modifikation der Firewall. Dazwischen befinden sich meist die Möglichkeiten, spezifische Alarmmeldungen über SNMP, Pager oder SMS absetzen zu können, oder genau die Verbindung, die den Angriff durchführt, durch gespoofte Reset-Pakete abzubrechen.

Die verschiedenen Technologien, die von einem NIDS eingesetzt werden können, ergänzen sich meist in ihrer Funktionalität. Moderne kommerzielle und freie Systeme unterstützen viele oder sogar fast alle diese Technologien. Die Unterschiede finden sich dann in der genauen Implementierung und Umsetzung. Diese Unterschiede sind jedoch sehr bedeutsam und können große Auswirkungen auf die Geschwindigkeit des NIDS haben. Hierbei soll nicht vergessen werden, dass die Sicherheit und die Geschwindigkeit eines NIDS nur in zweiter Linie vom genauen Produkt abhängen. In erster Linie ist eine durchdachte und konstante Wartung und Administration erforderlich. Dies kann durch geschultes Personal oder im Rahmen eines Wartungsvertrages erfolgen.

Auch ein NIDS leidet wie ein HIDS unter falsch-positiven und -negativen Meldungen. Hier kann nur eine sorgfältige Administration und Pflege die Anzahl dieser Falschmeldungen reduzieren.

2.4 Welchen Schutz bietet darüber hinaus ein IPS?

Auch Intrusion Prevention Systeme werden in zwei Gruppen eingeteilt. Die rechnerbasierten Intrusion Prevention Systeme werden als HIPS und die netzwerkbasieren Intrusion Prevention Systeme werden als NIPS bezeichnet.

2.4.1 Host Intrusion Prevention System

Bei den Host Intrusion Prevention Systemen für Linux handelt es sich üblicherweise um Systeme, die in der Lage sind, die Allmacht von `root` zu beschränken. Hierzu definieren und überwachen sie bestimmte Richtlinien, die die Tätigkeiten von `root` oder bestimmten privilegierten Prozessen beschreiben. Sobald eine Verletzung der Richtlinien erkannt wird, verhindern die Systeme die weitere Ausführung und können den betroffenen Benutzer abmelden oder den Prozess beenden.

In der Erkennung des Angriffes unterscheiden sie sich nicht von den entsprechenden HIDS.

2.4.2 Network Intrusion Prevention System

Die Network Intrusion Prevention Systeme arbeiten üblicherweise auf einem Router und analysieren dort den gesamten Netzwerkverkehr. Erkennen sie bösartige Pakete, so stoppen sie deren Weiterleitung. Daher werden sie häufig auch als Gateway-IDS oder Inline-IDS bezeichnet. Sie unterscheiden sich nicht von den klassischen NIDS in der Art und Weise, wie sie den Angriff erkennen und bieten lediglich zusätzlich zur Protokollierung des Angriffes die Möglichkeit, eine Weiterleitung des beanstandeten Paketes zu verhindern.

Die Möglichkeit, ein TCP-RST-Paket zum Abbruch der Netzwerkverbindung zu senden, wie es manche IDS-Systeme (auch Snort) anbieten, reicht nicht aus, um als NIPS

zu gelten. Hier kann der Angriff schon lange erfolgreich abgeschlossen worden sein, wenn das TCP-RST-Paket sein Zielsystem erreicht. Dies ist insbesondere der Fall, wenn ein einziges Paket für den Angriff ausreicht. Dies trifft zum Beispiel auf den SQL-Slammer zu.

2.4.3 Vor- und Nachteile eines IPS

Der Vorteil eines IPS liegt klar auf der Hand. Ein IDS erkennt lediglich den Angriff und alarmiert den Administrator. Dieser muss anschließend in mühsamer Arbeit überprüfen, ob der Angriff erfolgreich war, den Rechner für forensische Untersuchungen sichern und wiederherstellen. Wenn jedoch das IPS den Angriff direkt verhindert, entfallen diese aufwändigen Arbeiten.

Ist das IPS daher die Lösung für moderne Netzwerke? Kann man nun vielleicht sogar auf die Firewall verzichten?

Sicherlich nicht, denn das IPS leidet noch wesentlich mehr als ein IDS unter seiner Fehlerrate. Während bei einem IDS falsch-positive Meldungen lästig sind, sind sie bei einem IPS fatal. Bei jeder falsch-positiven Meldung unterbindet das IPS die damit verbundene Handlung. Tauchen diese falsch-positiven Meldungen selten aber regelmäßig im Zusammenhang mit unternehmenskritischen Prozessen auf, kann das IPS nicht eingesetzt werden oder die betreffenden Regeln müssen mindestens deaktiviert werden.

IPS werden daher üblicherweise lange nicht so »scharf« eingestellt wie die entsprechenden IDS.

Sobald ein IPS mit hundertprozentiger Erkennungsrate und null falsch-positiven Meldungen existiert, werden diese Systeme sicherlich Firewall und IDS ablösen. Aber das ist ferne Zukunft, wenn es denn jemals Realität werden sollte.