UNIVERSITÄT LEIPZIG

Intrusion Detection Systems

Problemseminar Mobilität und Sicherheit im Internet

Autor: Arne Brutschy

Prof. Dr. K. Irmscher

15. Juli 2003

This document was written as an exam at the University of Leipzig, Germany. It is intended to be used inside the university only. Nevertheless, I do not restrict the distribution of this document.

Because of the nature of this document, I am giving NO WARRANTY for any information in this document.

Altering of this document (retrieved, printed, or stored) and the distribution of this altered document in gernal is not permitted.

Universtät Leipzig
Abteilung für Rechnernetze
und Verteilte Systeme Postfach 100920
04009 Leipzig
Germany

Printed at Xylon Computersystems

Zusammenfassung

Dieses Dokument gibt eine umfassende Einführung in Aufbau, Konzeption und Einsatz von Intrusion Detection Systems. Außerdem werden sicherheitsrelevante Probleme beleuchtet und ein kurzer Einblick in die derzeitige Produktpalette gewagt.

Für Kommentare, Fragen oder Anregungen stehe ich gerne zur Verfügung.

Leipzig, 15. Juli 2003

Arne Brutschy[1]

[1] abrutschy@xylon.de

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einführung

1.1 Was ist ein „Intrusion Detection System“

Intrusion Detection Systeme (IDS) im Allgemeinen sind Hard- oder Software-Systeme, die der Erkennung eines Missbrauchs von Computersystemen dienen. Ein Missbrauch wird dabei meistens als Angriff bezeichnet; ein solcher, der erfolgreich verlaufen ist, als Einbruch (Intrusion). Die Aufgabe eines IDS ist dabei nicht nur die Erkennung eines erfolgreichen Angriffs, sondern schon die Entdeckung, Meldung und eventuell die automatische Vereitelung eines Angriffversuchs.

1.2 Arten von IDS

Ich stelle hier verschiedene Arten von IDS vor, wobei dies nicht unbedingt eine komplette Liste ist. Die Grenzen zwischen den Systemtypen sind manchmal nicht eindeutig zu ziehen.

1.2.1 Network Intrusion Detection Systems (NIDS)

Netzwerkbasierte IDS analysieren den Verkehr auf dem anliegenden Netzwerksegment durch passives Abhören des Netzwerkverkehrs. Alle Pakete einer Verbindung werden dabei gesammelt und auf Angriffsversuche hin untersucht. Ein typisches Beispiel ist ein System, das Portscans entdeckt, indem es auf auffällige Häufungen von TCP-Verbindungsversuchen achtet.

Im Allgemeinen werden mit NIDS Systeme bezeichnet, die den Netzwerkverkehr kompletter Netze überwachen, es gibt aber auch Systeme, die den auflaufenden Verkehr an einer einzigen Maschine überwachen (Hostbasierte IDS, seltener auch als HIDS bezeichnet).

1.2.2 System Integrity Verifiers (SIV)

System Integrity Verifiers überwachen die Integrität wichtiger Systemdateien (wie zum Beispiel `/etc/shadow` oder wichtige Registry-Einträge). Dies geschieht meistens mittels Bildung von Checksummen über alle vom Systemadministrator spezifizierten Dateien. Zusätzlich könne solche Systeme zum Beispiel überwachen, ob ein normaler User root-Privilegien erreicht.

Im Normalfall generiert ein solches System Warnmeldungen, einige beenden auch verdächtige Prozesse (zum Beispiel eine root-Shell).

Konzeptbedingt arbeiten diese Systeme fast immer hostbasiert.

Beispiel: Tripwire [21]

1.2.3 Log File Monitors (LFM)

Log File Monitor Systeme überwachen die Logdateien von relevanten Systemdiensten. Eine typische Anwendung wäre ein Parser, der die Logdateien des HTTP-Servers nach bekannten Attacken durchsucht.

Diese Systeme arbeiten meistens hostbasiert, können durch zentralisiertes Logging aber auch verteilte Systeme überwachen.

Beispiel: Swatch [22]

1.2.4 Deception Systems

Deception Systeme (a.k.a. Decoys, Lures, Traps oder Honeypots) enthalten oder emulierten sind Sicherheitslücken. Die Intention dabei ist, durch starke Überwachung dieser Systeme Angreifer zu fangen. Dies dient dazu neue Angriffstechniken kennen zu lernen und sich gegen diese absichern zu können.

Es kann sich dabei um einfache Programme handeln, die einen bestimmten Dienst simulieren, oder um ganze Netzwerke von präparierten Servern.

Allerdings ist die Rechtsfrage ein Problem bei solchen Systemen, da sie als Angriffsplattform auf fremde Netzwerke dienen können.

Ich werde in den folgenden Kapiteln nur auf NIDS weiter eingehen, da SIV- und LFM-Systeme im Kontext des Seminars nicht weiter von Interesse sind. Deception Systeme sind zwar hochinteressant, gehen aber weit über die Grenzen dieses Dokuments hinaus[2].

Außerdem sei anzumerken, dass der Begriff der hostbasierten IDS oft unterschiedlich interpretiert wird. So zählen auch SIVs und LFMs zu den hostbasierten IDS; Systeme die auf einem Host nur den lokal auflaufenden Verkehr analysieren werden mal den NIDS zugeordnet, mal den HIDS. In diesem Dokument ordne ich nur lokale Pakete analysierende IDS den hostbasierten IDS zu; trotz der Ausrichtung auf NIDS werde ich in Abschnitt 3.1.2 noch kurz auf Sinn und Aufbau von verteilten HIDS eingehen.

[2] Für weitergehende Informationen empfehle ich [7].

2 Aufgaben

Angriffs- und Einbruchserkennung (Intrusion Detection) ist eine zusätzliche Sicherheitsmaßnahme und ist für sich alleine nicht sehr sinnvoll. Da IDS meistens passive Sicherheit bieten, helfen sie bei Erkennung, Meldung und Rückverfolgung von Angriffen. Es gibt zwar aktive IDS, diese bergen aber immer ein potentielles Risiko, auf das später genauer eingegangen wird. Für ein umfassendes Sicherheitskonzept ist der Einsatz von weiteren Hard- und Softwarelösungen unerläßlich, zum Beispiel:

- Firewalls

  Firewalls sind meistens die erste Abwehrlinie eines Netzwerks. Simple Paketfilter betrachten nur Quelle und Ziel eines Pakets, das heißt solange diese legitim sind wird eine Verbindung erlaubt. Hier greifen IDS, die diese vermeintlich legitimen Verbindungen genauer untersuchen und eventuelle Sicherheitsgefährdungen im Paket-Payload aufspüren. Firewalls werden manchmal als aktive Komponente eines IDS eingesetzt, mehr dazu in 3.3.2.

- Authentisierung

  Die Authentisierung von Nutzern und Hosts ist eine viel gebrauchte Sicherheitsmaßnahme. Strikte Richtlinien bezüglich Passwörtern, Netzwerkzugriffen etc. erhöhen die Sicherheit zusätzlich.

- Verschlüsselung

  Unsichere Protokolle zu verschlüsseln ist inzwischen unerlässlich, um Angreifern die Möglichkeit zu nehmen, detaillierte Kentnisse über Aufbau und Art des Netzwerkverkehrs zu bekommen. Authentisierte, aber kompromittierte (da abgehörte) Nutzerkonten sind von ID Systemen nicht mehr vom legitimen Verkehr zu unterscheiden.
  VPNs fallen ebenfalls in diese Kategorie.

- Virenscanner

  Regelmäßige Viren- und Trojanersuche (auf allen internen Hosts) ist immens wichtig für eine sinnvolles Sicherheitskonzept. Durch Trojaner initiierte, verschlüsselte Verbindungen zu einem Angreifer sind weder durch Firewalls noch IDS erkenn- und verhinderbar.

Ich werde im folgenden weiter auf die einzelnen Aufgaben eines IDS eingehen.

2.1 Vermeidung von Problemen durch Abschreckung

Kontinuierliche Überwachung des Netzwerkverkehrs schreckt viele Angreifer schon von vornherein ab. Interne Nutzer werden damit vom Verletzen der Sicherheitsrichtlinien abgehalten. Leider trifft das aber nur auf „Funhackers“ oder „Scriptkiddies“ zu, ernstzunehmende Angreifer werden sich wohl im seltensten Fall vom Vorhandensein eines IDS beeindrucken lassen.

2.2 Erkennung von Problemen, die von anderen Sicherheitsmaßnahmen nicht erkannt wurden

Wie schon erwähnt, werden manche Arten von Angriffen von bestimmten Sicherheitsmaßnahmen nicht erkannt. Setzt man zum Beispiel zusätzlich zu einem Paket-Filter eine Application Firewall zum Filtern des HTTP-Verkehrs ein, ermöglicht diese zwar das Filtern von Verkehr der nicht dem HTTP-Protokoll entspricht, auf diesem Protokoll basierende Angriffe werden aber nicht erkannt.

Zusätzlich gibt es viele Systeme, die aus unterschiedlichsten Gründen (zu alt, konzeptionelle Probleme etc.) nicht gegen bekannte Sicherheitslücken abgesichert werden können. Die Anzahl der Systeme, die trotz verfügbarer Gegenmaßnahmen (Patches, Hotfixes, Workarounds) anfällig sind, sei es aus Zeit-, Ressourcen- oder Kompetenzmangel, ist nicht zu unterschätzen.

Ein IDS kann in diesen Fällen feststellen, wenn ein Angreifer nicht korrigierte oder nicht korrigierbare Sicherheitslücken ausnutzt. Durch aktive Gegenmaßnahmen kann dies teilweise auch verhindert werden.

2.3 Erkennung von Vorzeichen eines Angriffs

Angriffe auf Computersysteme laufen meistens in vorhersagbaren Stadien ab. Vor dem eigentlichen Angriff erfolgt zum Beispiel meistens eine genaue Sondierung des Netzwerkes. Dies dient zum Erkennen bestmöglicher Ansatzpunkte für einen Angriff. In Fällen solcher Sondierungen hilft die Früherkennung durch eine IDS dabei rechtzeitig auf den zu erwartenen Angriff zu reagieren.

2.4 Dokumentation einer Bedrohung

Es ist absolut notwendig, dass man über eine laufende oder bereits beendete Bedrohung so viele Daten wie möglich sammelt, um einerseits bestehende Sicherheitslücken zu schließen und andererseits die Verfolgung des Angreifers zu ermöglichen. Deshalb die eine der Hauptaufgaben eines IDS die Dokumentation und Registrierung aller im Netzwerk auftretenden Anomalien.

2.5 Aufdeckung von Sicherheitslücken

Da komplexe Systeme oft nicht mehr überschaubar sind, helfen IDS bei dem Versuch, ein lückenloses Sicherheitskonzept umzusetzen. So können sie zum Beispiel nicht bekannten oder dokumentierten Netzwerkverkehr aufdecken, der zwar legitim ist, aber dennoch eine Sicherheitslücke darstellt.

3 Architektur

Im folgenden Abschnitt beschreibe ich den Aufbau von IDS. Da es sich bei den meisten IDS um kommerzielle Produkte handelt, sind Informationen über die Architektur dieser Systeme rar und Rückschlüsse schwer zu ziehen. Ich beschreibe daher den allgemeinen Aufbau dieser Systeme. Diesem folgen die meisten IDS, auch freie Implementierungen wie zum Beispiel Snort [20].

3.1 Allgemeiner Aufbau

Aufgrund der vielen verschiedenen ID Systeme hilft es, ein Modell zu haben, welches den allgemeinen Aufbau aller Implementierungen beschreibt. Das „Common Intrusion Detection Framework“ (CIDF) [1] definiert einen Satz von Komponenten, die zusammen ein Intrusion Detection System bilden.

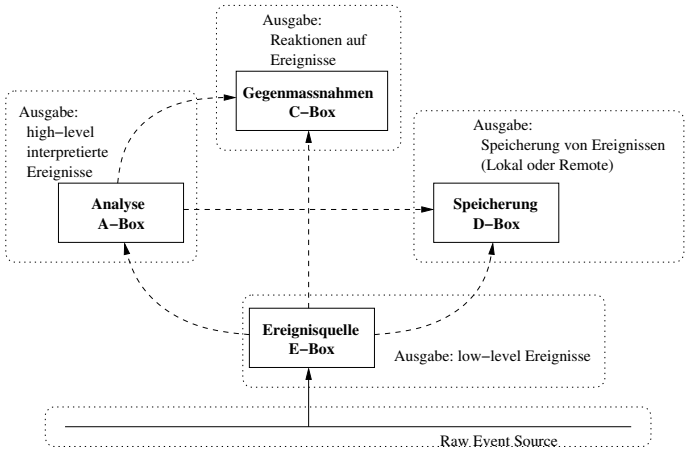

Abbildung 1: CIDF Komponenten

Es handelt sich dabei um folgende Komponenten:

- Ereignisquelle (E-Box)
  Die Ereignisquelle sammelt „Ereignisse“ aus verschiedenen Schichten des Systems, wobei Netzwerk-, Host- und Applikationsüberwachung die häufigsten Quellen sind.
- Analyseeinheit (A-Box)
  Die A-Box analysiert die von der Ereignisquelle gelieferten Daten. Sie korreliert Daten aus verschiedenen Quellen und versucht die relevanten Daten herauszufiltern. Häufige Analysemethoden sind die Erkennung von *Angriffssignaturen* und *Anomalien*.
- Speichermechanismen (D-Box)
  Die Ereignisquelle und Analyseeinheit können große Datenmengen produzieren. Diese Informationen müssen schnell und zuverlässig gespeichert werden, damit sie von Nutzen sind (zum Beispiel für eine forensische Analyse eines Angriffs)
- Gegenmaßnahmen (C-Box)
  Obwohl ID Systeme aus Sicherheitsgründen oft nur als passive Alarmsysteme konzipiert sind, enthalten viele kommerzielle IDS Gegenmaßnahmen. Das erlaubt einem System Angriffe zu verhindern oder zumindest einzuschränken.

Abbildung 1 zeigt die Beziehung zwischen den einzelnen Komponenten eines CIDF.

3.1.1 Aufbau von NIDS

Netzwerkbasierte Intrusion Detection Systeme sind im Allgemeinen nach dem CID-Framework aufgebaut. Allerdings erfordern die immer schnelleren Netzwerke zunehmend eine verteilte Bearbeitung der Ereignisse. Bei Geschwindigkeiten von einem Gigabit und mehr reicht ein einzelnes System nicht mehr aus, alle

anfallenden Daten zu verarbeiten. Um trotzdem solche Hochgeschwindigkeitsnetze überwachen zu können, setzt man Cluster von Analyserechnern ein [5]. Dabei verteilt ein sogenannter „Scatterer“ die Daten an eine Reihe „Slicer“, die diese ein managebare Teilmengen aufteilen und über einen Switch an einen von mehrern „Reassemblern“ verschicken. Dieser sorgt dafür, dass die Pakete einer Verbindung richtig zusammengesetzt werden. Danach schickt er die Pakete an einen Intrusion Detection Sensor. Diese Sensoren (von denen idealerweise ein ganzes Netz existiert) sind voneinander unabhängig und bearbeiten nur jeweils nur eine bestimmte Teilaufgabe (zum Beispiel die Analyse von HTTP-Verkehr). Somit ist es möglich, auch bei sehr hoher Netzlast keinen Paketverlust durch Überlastung zu haben. Abbildung 2 zeigt den Aufbau eines solchen Systems.

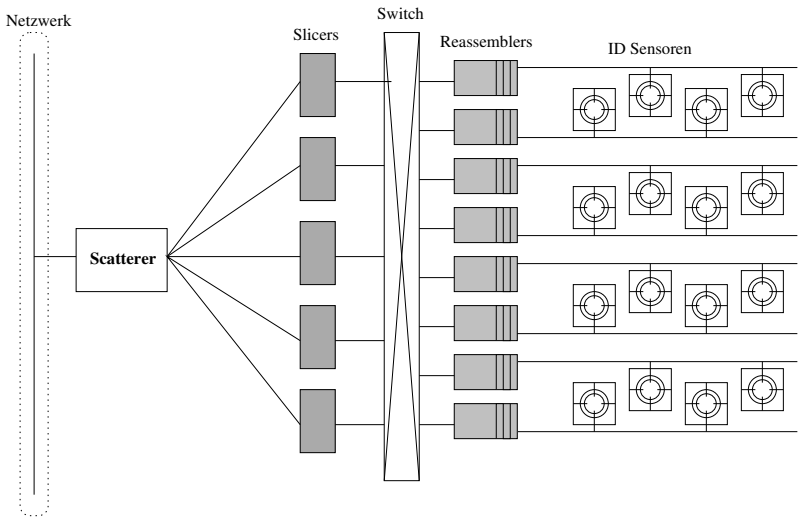

Abbildung 2: Aufbau eines hochgeschwindigkeits IDS

3.1.2 Aufbau von verteilten HIDS

Einer der großen Nachteile von NIDS ist, dass sie nicht wissen, was auf den einzelnen Systemen im Netzwerk passiert (siehe Kapitel 5). Dies wird durch den Einsatz von verteilten, hostbasieren IDS gelöst. Dabei wird auf möglichst jedem Host im Netzwerk ein ID System installiert, welches lokal die Log- und Systemdateien überwacht, Prozesse und Nutzer kontrolliert. Im Gegensatz zu reinen hostbasierten IDS gibt es bei den verteilten HIDS einen gemeinsamen Managementknoten, der die Informationen aller einzelnen Hosts sammelt und miteinander korreliert. Dadurch wird einer der Hauptnachteile von HIDS, das Nichterkennen von verteilten Angriffen, kompensiert. Ein Nachteil dieser Lösung ist, dass für jedes Betriebssystem im Netzwerk ein eigener Client verfügbar sein muß. Dies ist bei vielen älteren Systemen nicht der Fall oder schlichtweg nicht mehr möglich. Ein weiterer Nachteil ist die hohe Belastung des zentralen Knotens bei einer hohen Anzahl von Clients. Abbildung 3 zeigt den allgemeinen Aufbau eines solchen verteilten HIDS.

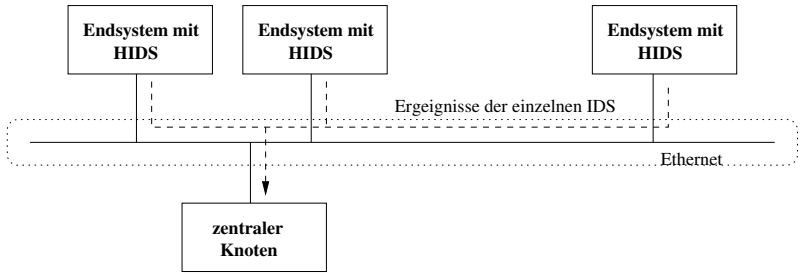

Abbildung 3: Aufbau eines verteilten HIDS

3.1.3 Hybride IDS

Hybride IDS sind die logische Konsequenz aus NIDS und verteilten HIDS. Der zentrale Managementknoten, der alle Informationen von den einzelnen Hosts sammelt, ist zusätzlich mit dem NIDS verbunden (oder ist selbst ein NIDS). Der zentrale Knoten kann dann alle auftretenen Netzwerkereignisse mit denen der einzelnen Hosts korrelieren. Dadurch werden einige der Hauptnachteile der beiden Arten von IDS gelöst. Diese Art der Systeme sind allerdings zur Zeit (noch) nicht weit verbreitet, da sie sehr komplex sind und extrem hohe Leistungsanforderungen haben. Bei heutigen Netzwerken wird ein solches System nur durch massives Clustering sinnvoll einsetzbar.

Beispiel: Prelude Hybrid IDS [23]

3.2 Passive Analyse

Network Intrusion Detection Systeme basieren auf der passiven Analyse des Verkehrs des anliegenden Netzwerksegments. Sie versuchen Angriffe anhand bekannter Muster oder statistischer Anomalien zu erkennen. Dazu werden alle Pakete im sogenannten „promiscuous mode“ mitgeschnitten. Das IDS versucht dann die einzelnen Pakete einer Verbindung wieder richtig zusammenzusetzen. Auf Basis der erhaltenen Daten fällt es seine Entscheidungen über die Legitimität und Plausibilität der Verbindung.

Der große Vorteil der passiven Analyse ist, dass der einzelne Rechner im Netz nicht von der Existenz des IDS wissen muss (und es normalerweise auch nicht tut). Diese Art von Überwachung ist extrem schwer zu umgehen, da sie auf der untersten Schicht eines Netzwerkes ansetzt.

Es existieren dennoch einige Möglichkeiten, ein NIDS zu umgehen oder zumindest zu täuschen. Einige davon werden in Kapitel 5 behandelt.

3.2.1 Signaturerkennung

Die Frage, welche Art von Information für ein IDS interessant ist, hängt davon ab, was es versucht zu erkennen. Für ein System, das DNS Verkehr überwacht, sind die Hostnamen der abgefragten Rechner interessant, für ein System, das versucht Angriffe gegen FTP Server zu erkennen, ist der Inhalt aller TCP Verbindungen zum Port 21 interessant.
Manche Angriffe können durch das einfache Parsen von IP-Paketen erkannt werden, andere brauchen einen Kontext, mit dem sie in Verbindung gebracht werden müssen (eine DNS-Anfrage mag nur dann relevant sein, wenn es sich um einen bestimmten Host handelt).
Die meisten IDS indentifizieren Angriffe mittels einer Technik, die „Signaturerkennung“ (oder manchmal „Missbrauchserkennung“) genannt wird: das System untersucht die Pakete einer Verbindung nach Mustern von bekannten Angriffen („Pattern Matching“).
Zum Beispiel werden die Payloads aller HTTP-Verbindungen nach dem String „`phf`“ durchsucht. Wenn ein solcher String gefunden wird (zum Beispiel in der Anfrage „`GET /cgi-bin/phf?`“), erkennt das IDS einen Angriff und leitet entsprechende Maßnahmen ein.

3.2.2 Statistische Analyse

Eine weitere Analysemethode, die viele Systeme einsetzen, ist die statistische Analyse des Netzwerkverkehrs. Das System versucht damit Anomalien im Netzwerkverkehr aufzudecken. So ist es zum Beispiel auffällig, wenn sich Sonntagnacht viele Workstations auf den Servern einloggen und anfangen Daten zu kopieren. Die Möglichkeiten sind vielfältig, allerdings ist die Gefahr von Fehlalarmen („false positives“) sehr hoch. Im Allgemeinen muß ein solches System sehr gut an den Alltag eines Netzwerkes angepasst werden.

3.2.3 Weitere Methoden

Weitere diskutierte (aber seltener eingesetze) Methoden umfassen lernfähige Systeme, die sich an den üblichen Verkehr eines Netzwerkes „gewöhnen“ und darauf basierend versuchen, Anomalien zu entdecken. Desweiteren sind biometrische und sogar semi-intelligente Systeme im Gespräch.

3.3 Reaktionsarten

Wenn ein IDS einen Angriff erkennt, leitet es eine oder mehrere der folgenden Reaktionen ein. Diese Reaktionen lassen sich in zwei Kategorien einordnen.

3.3.1 Passive Reaktionen

Passive Reaktionen werden von jedem IDS beim Auftreten eines sicherheitsrelevanten Ereignisses ausgeführt.

- Alarm und Benachrichtigung
  Alarme und Benachrichtigungen werden vom System generiert um auf das Ereignis aufmerksam zu machen. Dies umfasst visuelle Alarme wie Popup-Fenster, das Senden von E-Mails oder Nachrichten an Pager und Handys sowie Benachrichtigung anderer Computersysteme über SNMP-Traps.
- Logging
  Die auftretenen Ereignisse werden detailiert und möglichst sicher gespeichert um eine spätere Analyse eines Angriffs zu ermöglichen.

3.3.2 Aktive Reaktionen (Gegenmaßnahmen)

Manche IDS bieten aktive Reaktionen auf Ereignisse an. Diese bergen allerdings ein hohes Risiko, da Absenderadressen beim IP-Protokoll nicht verlässlich sind (das heißt gefälscht werden können).

- Beenden der Verbindung
  Das IDS kann, wenn ein Angriffsversuch erkannt wird, die betreffende Verbindung beenden. Dies ist die schwächste und auch verbreitetste Form der Gegenmaßnahme.
- Aktivieren von Firewall-Regeln
  Das IDS kann die Absenderadresse des Angriffs mittels Regeln auf dem Eingangsrouter komplett sperren. Hier sieht man auch deutlich die Gefahr bei aktiven Reaktionen: ein Angriff mit gefälschter Absenderadresse kann zur Sperrung von legitimen Verkehr führen (und damit wiederum eine Sicherheitslücke öffnen).
- Aktionen gegen den Angreifer
  Diese Art von Reaktionen bewegen sich in der rechtlichen Grauzone und werden deshalb kaum eingesetzt. Sie sind außerdem nur schwer automatisierbar.

4 Einsatz

Die Einsatzmöglichkeiten eines IDS sind vielfältig. Sie reichen von dedizierter Überwachung eines Dienstes auf einem bestimmten Server bis zu genereller Überwachung von interkontinentalen WAN-Strecken.

4.1 Einsatz von netzwerkbasierten IDS

Je nach Einsatzort und -art muß ein NIDS unterschiedliche Regeln tragen und unterschiedlich auf Ereignisse reagieren. So ist heutzutage ein Portscan im Internet (leider) nichts ungewöhnliches und sollte damit keine hohe Alarmpriorität erhalten, sobald ein solcher aber im internen Netz auftritt, muß das zuständige NIDS sofort reagieren.

Im Allgemeinen wird ein NIDS wie in Abbildung 4 eingesetzt.

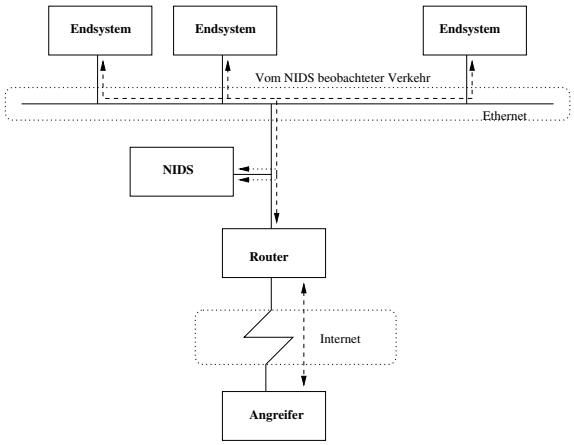

Abbildung 4: Allgemein üblicher Einsatz eines NIDS

4.2 Unterschiedlicher Einsatz in gößeren Installationen

In größeren Netzen (zum Beispiel Campus- oder Unternehmensnetzwerke) gibt es mehrere sinnvolle Einsatzmöglichkeiten. Dabei ist es wichtig, dass die einzelnen IDS detailliert an den jeweiligen Standort angepasst sind, um Fehlalarme („false positives“) und unerkannte Angriffe („false negatives“) zu minimieren.

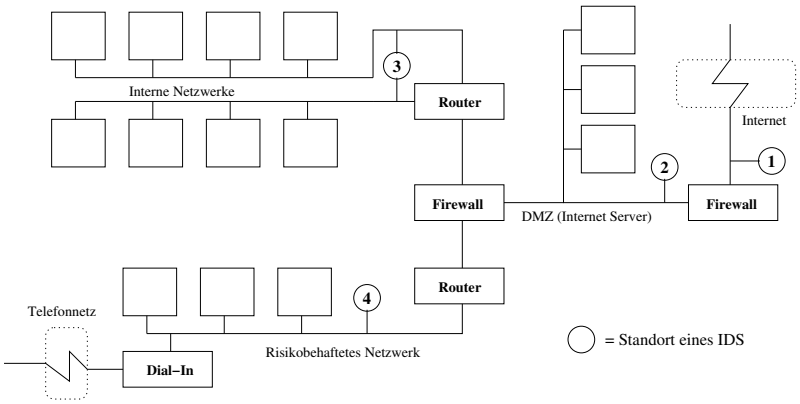

Abbildung 5: Allgemein üblicher Einsatz eines NIDS

4.2.1 Außerhalb der externen Firewall

(Abbildung 5, Standort 1)

Vorteile:

- Dokumentiert die Art und Anzahl der Angriffsversuche aus dem Internet.
- Liefert detaillierte Statistiken über die (angebliche) Herkunft der Angriffsversuche.

4.2.2 Innerhalb der DMZ

(Abbildung 5, Standort 2)

Vorteile:

- Erkennt Angriffe, die von ausserhalb gegen die Grenzfirewall des Netzwerkes gehen.
- Zeigt Probleme der äußeren Firewall auf.
- Sieht Angriffe gegen die Server in der DMZ[3].
- Selbst wenn eingehender Verkehr nicht als Angriff erkannt wurde, kann das IDS evtl. den ausgehenden Verkehr eines kompromittierten Servers erkennen.

4.2.3 An wichtigen Backbones

(Abbildung 5, Standort 3)

Vorteile:

- Überwacht eine Hauptteil des internen Verkehrs und erhöht somit die Wahrscheinlichkeit einen Angriff zu entdecken.
- Erkennt Angriffe von innen, verletzungen der Sicherheitsrichtlinie durch Nutzer sowie Viren und Trojaner.

4.2.4 Innerhalb eines risikobehafteten Netzwerkes

(Abbildung 5, Standort 4)

Vorteile:

- Erkennt Angriffe auf kritische Systeme.
- Erlaubt es, IDS aus Kostengründen auf die wertvollsten und/oder schützenswertesten Systeme zu beschränken.

4.3 Einsatz von hostbasierten IDS

Hostbasierte IDS werden, je nachdem ob es sich um ein verteiltes System oder individuelle Installationen handelt, auf den betreffenden Rechnern installiert. Individuelle Installationen werden hauptsächlich auf kritischen Systemen wie Internetservern aufgesetzt. Eine zu große Menge von Einzelinstallationen macht nicht viel Sinn, da das System dadurch nicht mehr effektiv verwaltbar ist. Für größere Netze ist ein verteiltes HIDS nötig, dessen Einsatz bereits in Abschnitt 3.1.2 genauer beschrieben wurde.

[3] Netzwerk, in dem die vom Internet aus verfügbaren Server stehen, oft „Demilitarisierte Zone“ genannt

5 Probleme

Es existierten zwei generelle Probleme beim Einsatz von NIDS: erstens, die Informationen auf dem Netzwerk sind unzureichend um zuverlässig zu rekonstruieren was bei komplexen Protokolltransaktionen abläuft, und zweitens, ID Systeme sind Aufgrund ihrer funktion anfällig für Denial of Service Angriffe.

5.1 Unzureichende Informationen im Netzwerk

Ein netzwerkbasiertes IDS liest alle Pakete vom anliegenen Netzwerk um herauszufinden, was auf dem Zielrechner geschieht. Ein Paket alleine ist nicht übermäßig relevant, viel mehr ist das Verhalten des Systems, welches das Paket empfängt, von Interesse.

Leider ist es für ein NIDS nicht komplett vorhersagbar, wie das Betriebssystem auf dem Zielrechner die Pakete verarbeitet. Verschiedene Inkonsistenzen in den unterschiedlichen Implementierungen der Protokoll-Stacks führen dazu, dass das IDS Pakete annimmt, die das Zielsystem verwirft oder anders herum.

Nehmen wir zum Beispiel ein IDS und ein Endsystem an, welche über eine lange oder langsame Strecke miteinander verbunden sind. Es kann sein, dass in einem solchen Fall das IDS Pakete verarbeitet, die vom Endsystem aufgrund der Verzögerung abgelehnt wurden.

Ein anderes Beispiel ist ein UDP Paket mit einer falschen Prüfsumme. Die meisten Betriebssysteme werden die Annahme des Paketes verweigern, einige ältere Systeme vielleicht nicht. Das IDS muß wissen, ob jedes System, das es überwacht, das Paket annimmt oder nicht; falls dies nicht der Fall ist, werden die Geschehnisse auf solchen Maschinen falsch rekonstruiert.

Selbst wenn das IDS weiss, um welche Art von Betriebssystem es sich handelt, kann es sein, dass es dem System immer noch nicht möglich ist vorherzusagen ob der Zielrechner das Paket annimmt oder nicht. Ein Rechner, der keinen freien Hauptspeicher mehr hat, nimmt zum Beispiel keine Pakete mehr an. CPU- und Netzwerkauslastung können den gleichen Effekt hervorrufen.

Als Resultat haben diese Probleme, dass es einem NIDS einfach nicht möglich ist, die genauen Abläufe auf einem Zielsystem anhand von Netzwerkpaketen zu rekonstruieren; es braucht dafür detaillierte Zusatzinformationen. Leider gibt es für ein NIDS keinen einfachen und verwaltbaren Weg diese Informationen zu bekommen.

5.2 Anfälligkeit für Denial of Service

Ein „Denial of Service“ (DoS) Angriff ist eine Attacke gegen die Verfügbarkeit eines Systems. Denial of Service-Angriffe können genutzt werden um Systeme zu umgehen oder auszuschalten. In Bezug auf DoS-Angriffen stellt sich die Frage, ob ein System „fail-open“ oder „fail-closed“ ist. Ein „fail-open“ System bietet im Fall eines Denial of Service-Angriffs den von ihm gesicherten Systemen keinen Schutz mehr. Ein „fail-closed“ System andererseits lässt das Netzwerk gesichert, falls es ausgeschaltet wird.

Die Begriffe „fail-open“ und „fail-closed“ werden meistens im Zusammenhang mit Firewalls gebraucht. Eine fail-open Firewall hört im Falle eines Absturzes auf, den Zugang zum Netzwerk zu kontrollieren. Gute Firewalls sind so entworfen, dass sie bei einem Crash das Netzwerk komplett blocken und damit zwar allen Verkehr verhindern, aber sichern.

Netzwerkbasierte IDS sind passiv. Da sie das Netzwerk in keinem Fall kontrollieren, sind sie zu „fail-open“ gezwungen. Ein Angreifer, der ein IDS abstürzen oder es seine Resourcen aufbrauchen lässt kann das Netzwerk angreifen, als wäre kein IDS vorhanden.

Leider kann man sich schwer gegen Denial of Service-Angriffe absichern. Vor allem das Problem der erschöpften Ressourcen (zum Beispiel durch Flooding) lässt sich nicht so einfach beheben.

6 Angriffe gegen NIDS

Im folgenden Kapitel lege ich Angriffe gegen NIDS dar. Diese sind alle allgemein bekannt, funktionieren aber angeblich noch gegen eine Vielzahl der heutzutage eingesetzten Systeme. Ich habe die Angriffe allerdings nicht selbst getestet.

6.1 Einfache Angriffe

Einfache IDS kann man auch durch einfache Mittel umgehen.

- Pattern Change Evasion

  In bekannten Angriffskripten kann man Strings ändern. Dies schaltet teilweise die Signaturerkennung eines IDS aus.

- Paketfragmentierung

  Viele einfache ID Systeme können mit Paketrfragmentierung nicht richtig umgehen. Das Aufteilen eines Angriffs auf mehrere Pakete reicht schon um solche Systeme zu umgehen.

- Koordinierte, langsame Angriffe

  Angriffe von verschiedenen Absenderabdressen über einen langen Zeitraum hinweg sind von IDS ebenfalls schwer fassbar.

- Langsame Scans

  Portscans kann man über Stunden und Tage hinweg verteilen und so die Erkennung durch ein IDS verhindern. Aus Ressourcenmangel ist es für ein IDS schwer Langzeitdaten zu korrelieren.

- Vermeidung von Standards

  Die Vermeidung von Standardparametern bei der Nutzung von Trojaner und Ähnlichem hilft ebenfalls dabei, die Signaturerkennung einfacher IDS zu umgehen. Zum Beispiel indentifiziert ein IDS eventuell den Trojaner „Back Orifice“ an dem benutzten Standard-Port 31337. Das Ändern des benutzten Ports würde solch ein IDS umgehen.

6.2 Insertion

Ein IDS kann ein Paket akzeptieren, das ein Endsystem verwirft. Ein solches IDS macht den Fehler anzunehmen, dass das Endsystem das Paket ebenfalls angenommen und verarbeitet hat obwohl es das in Wirklichkeit nicht getan hat. Ein Angreifer kann dies ausnutzen indem er Pakete sendet, die das Zielsystem verwirft, das IDS aber annimmt. Auf diese Weise kann der Angreifer Pakete in das ID System einfügen (to insert) - kein anderes System in Netzwerk stört sich an den Paketen.

Diesen Angriff nennt man daher im Allgemeinen „Insertion“-Attacke [4]. Diese Art von Angriff nutzt eine der am meisten verbreiteten Schwächen von IDS. Diese erlaubt es einem Angreifer die Signaturanalyse eines IDS zu umgehen und somit Angriffe an dem System vorbei auf das Netz auszuführen.

Ein Beispiel:

Ein IDS versucht den bekannten PHF-Angriff zu erkennen, dafür sucht es in den Paketen von HTTP-GET Verbindungen nach dem String „phf“. Der Angreifer schickt nun drei Pakete los, das erste und das letzte werden von dem Endsystem erkannt und enthalten die Strings des PHF-Angriffs „GET /cgi-bin/p“ und „hf?“. Das mittlere Paket enthält zum Beispiel den String „XXX“. Da das Endsystem das mittlere Paket nicht annimmt, wird der String zu einem kompletten Angriff „GET /cgi-bin/phf?“ zusammengesetzt. Das IDS akzeptiert das mittlere Paket und setzt demzufolge die Pakete anders (falsch) zusammen und bildet den String „GET /cgi-bin/pXXXhf?“. Da dieser String „phf“ nicht enthält, wird kein Angriff erkannt.

Abbildung 6 verdeutlicht noch einmal diese Art von Angriff. Ein Angreifer konfrontiert das IDS mit einem Strom von 1-Buchstaben Paketen, bei dem die Pakete mit dem Buchstaben „X“ nur von dem ID System akzeptiert werden. Als Resultat rekonstruieren das Endsystem und das IDS verschiedene Strings.

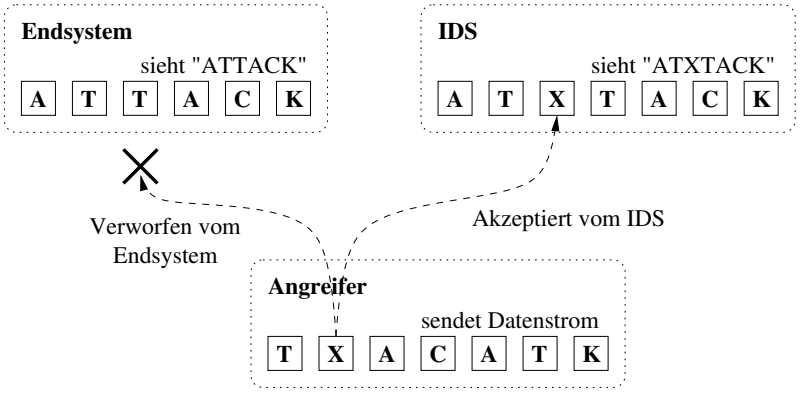

Abbildung 6: Beispiel für einen Insertion-Angriff

Im Allgemeinen treten Insertion-Angriffe auf, wenn das IDS beim Verarbeiten von Paketen weniger strikt ist als die Endsysteme. Eine einfache Lösung für das Problem scheint zu sein, das ID System so zu entwerfen, dass es beim Verarbeiten von Paketen so strikt wie möglich ist. Jedoch tritt, wenn man diesen Weg wählt, ein anderes schweres Sicherheitsproblem auf („Evasion“-Angriffe).

6.3 Evasion

Ein Endsystem kann Pakete akzeptieren, die ein IDS verwirft. Ein IDS das ein solches Paket fälschlicherweise verwirft, übersieht den Inhalt des Paketes komplett. Dieser Umstand kann ebenfalls ausgenutzt werden, dieses mal um Pakete an dem IDS vorbei zum Zielsystem zu schleusen. Da der Angreifer die IDS umgeht (to evade), wird dieser Angriff „Evasion“-Attacke genannt [4]. Diese Art des Angriffs kompromittiert die Genauigkeit des IDS besonders stark; komplette TCP-Sessions können auf diese Weise am System vorbei geleitet werden.

Evasion-Angriffe überwinden Signaturerkennung auf ganz ähnliche Weise wie Insertion-Angriffe. Wie vorher sendet der Angreifer einen Datenstrom, der auf dem Zielsystem und dem IDS unterschiedlich verarbeitet wird, dieses Mal jedoch sieht das Endsystem mehr als das ID System.

Beim vorigen Beispiel eines PHF-Angriffs würde das IDS den String „GET /cgi-bin/p?“ sehen, während das Endsystem die Daten richtig rekonstruiert.

Abbildung 7 verdeutlicht diese Art von Angriff.

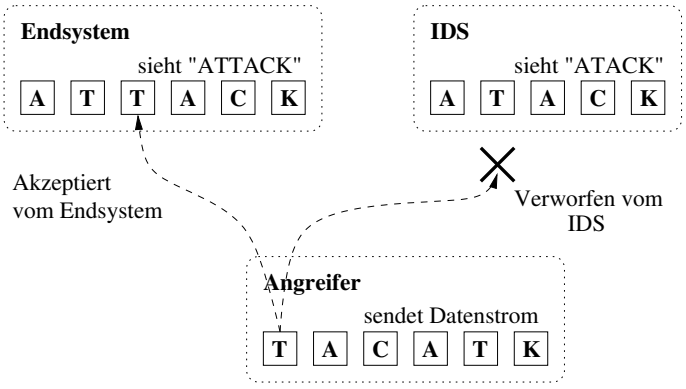

Abbildung 7: Beispiel für einen Evasion-Angriff

6.4 Insertion und Evasion Angriffe in der Realität

In der Realität sind Insertion- und Evasion-Angriffe nicht so einfach ausführbar. Ein Angreifer kann nicht nach Belieben Zeichen in den Datenstrom einfügen. Allerdings spielt diese Art von Angriffen schon eine große Rolle bevor überhaupt eine Signaturanalyse ausgeführt wird.

Dies ergibt sich aus der Notwendigkeit, dass ein ID System erst einmal alle Pakete einer Verbindung richtig ordnet und die Daten rekonstruiert, bevor es sie analysieren kann. Bei einfachen Protokollen wie DNS genügt es ein einzelnes UDP-Paket zu analysieren, um dessen Zweck zu erfahren.

Komplexe Protokolle wie TCP sind allerdings wesentlich schwieriger zu handhaben. So sind durch Inkonsistenzen in den unterschiedlichen Implementierungen der Protokoll-Stacks Möglichkeiten gegeben, Pakete zu erzeugen die nur einige Systeme verwerfen und andere annehmen. Es ist zum Beispiel möglich, durch Evasion-Angriffe mittels falscher Options-Flags komplette Verbindungen am IDS vorbei zu schleusen. Ein weiteres Beispiel ist das Fluten eines IDS mittels Paketen, deren Sequenznummern völlig wahllos verteilt sind. Dadurch kann das System die Daten nicht mehr in die richtige Reihenfolge bringen und verliert eventuell die Verbindung. Ein solches vorgehen nennt man „desyncronisieren“ einer Verbindung.

Ein anderes Problem ist, dass auf TCP-Pakete fragmentiert werden können. Dadurch gibt es wiederum mehrere Angriffsmöglichkeiten. So ist zum Beispiel das Verhalten von Betriebssystemen bei überlappenden Segmenten unterschiedlich, wodurch Insertion-Angriffe möglich werden.

6.5 Denial of Service

Wie bereits erwähnt sind Denial of Service-Angriffe gegen ID Systeme schwerwiegend, da sie aufgrund ihrer passiven Konzeption „fail-open“ sind und im Falle eines Fehlers das gesamte Netzwerk unbeobachtet lassen.

Manche Denial of Service-Angriffen existieren aufgrund von fehlerhafter Software. Günstigerweise sind diese Art von Fehler einfach zu beseitigen, das Auffinden solcher Fehler erfordert aber extrem aufwendiges Software Auditing.

Angreifer können Denial of Service-Attacken gegen IDS fahren, indem sie besonders ressourcenintensive Vorgänge indentifizieren und ausnutzen. So kann zum Beispiel ein Angreifer, der von einem ID System weiss, das es TCP-TCBs (Transmission Control Blocks) schon auf ein einfaches SYN-Paket hin öffnet, mit tausenden sinnlosen Verbindungen fluten. Dies führt dazu, dass das System aus Hauptspeichermangel keine TCBs für neue Verbindungen anlegen kann; der Angreifer kann dann neue Verbindungen initiieren, ohne dass sie wahrgenommen werden.

Es ist im Allgemeinen unerheblich, ob es sich nun um Mangel an (Haupt-)Speicher, CPU-Leistung oder Netzwerkdurchsatz handelt, das Resultat ist immer das gleiche.

6.6 Missbrauch von aktiven Reaktionen

Wie schon erläutert wurde, kann ein Angreifer ein ID System durch gezielte Angriffe dazu bringen, legitime Verbindungen zu unterbrechen. Dies kann zu weiteren Sicherheitslücken führen.

Ein Beispiel hierfür ist ein Angriff, der die IDS veranlasst eine laufende IKE (Internet Key Exchange) Verbindung zu beenden. Sind die beteiligten VPN Partner im Fallback-Modus konfiguriert, wird die Verbindung unverschlüsselt aufgebaut. Dies ermöglicht es dem Angreifer, sensitive Daten abzuhören.

A Anhang

A.1 Quellenverzeichnis

Onlinequellen

[1] Stuart Staniford-Chen, Dan Schnackenberg, Brian Tung, *Common Intrusion Detection Framework*, Position paper accepted to the Information Survivability Workshop, Orlando FL, Oktober 1998. `http://www.isi.edu/gost/cidf/papers/cidf-isw.txt`

[2] Rebecca G. Bace, Peter Mell, *NIST Special Publication on Intrusion Detection Systems*, Februar 2001. `http://www.snort.org/docs/nist-ids.pdf`

[3] Robert Graham, *FAQ: Network Intrusion Detection Systems*, März 2000. `http://www.robertgraham.com/pubs/network-intrusion-detection.html`

[4] Thomas H. Ptacek, Timothy N. Newsham, *Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection*, Januar 1998. `http://www.insecure.org/stf/secnet_ids/secnet_ids.html`

[5] Christopher Krügel, Thomas Toth, Engin Kirda, *Stateful Intrusion Detection for High-Speed Networks*, In IEEE Symposium on Security and Privacy, IEEE Computer Society Press, USA, Mai 2002. `http://www.infosys.tuwien.ac.at/Staff/chris/doc/2002_04.ps`

[6] Christopher Krügel, Thomas Toth, Engin Kirda, *Service Specific Anomaly Detection for Network Intrusion Detection*, In Symposium on Applied Computing (SAC), ACM Digital Library, Spanien, März 2002. `http://www.infosys.tuwien.ac.at/Staff/chris/doc/2002_03.ps`

[7] Intrusion Detection Systems, Honeypots & Incident Handling, `http://www.honeypots.net`

Bücher

[10] Earl Carter, Rick Stiffler, *Cisco Secure Intrusion Detection System*, Cisco Press, 2001.

[11] Rebecca G. Bace, *Intrusion Detection*, Macmillan Technical Publishing, 2000.

[12] Edward G. Amoroso, *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response*, Intrusion.net, 1999.

[13] Stephen Northcutt, *Network Intrusion Detection: An Analyst's Handbook*, New Riders, 1999.

Produkte

[20] Snort™, The Open Source Network Intrusion Detection System, `http://www.snort.org`

[21] Tripwire, Inc., Tripwire™, `http://www.tripwire.com`

[22] Swatch: the active log file monitoring tool, `http://swatch.sourceforge.net`

[23] Prelude Hybrid IDS, `http://www.prelude-ids.org`

Index