Intrusion Detection Systeme

am Beispiel von SNORT

Poster erstellt im Rahmen der Vorlesung „Ausgewählte Kapitel der IT-Security"
bei Prof. Dittmann, Institut für Technische und Betriebliche Informationssysteme
(Multimedia and Security), Fakultät für Informatik, Otto-von-Guericke Universität Magdeburg
von Christian Graf und Andreas Mähnz
WS2003/04, Januar 2004

Einführung

Motivation

- Computer heute unersetzliches Werkzeug und Medium
- weltweite Datennetze zur Kommunikation überall präsent
- Computer und Netzwerke sind potentielles Angriffsziel für z.B. konkurrierende Mitbewerber

Grundlagen

Was ist 'intrusion' und was 'intrusion detection'?

- Englisch ‚intrusion': Eindringen in Computersysteme, die Umgehung der Sicherheitsmaßnahmen
- Sysadmins versuchen gegenzuwirken & Systemintegrität zu sichern
  → Analyse des Netzwerkverkehr anhand bestimmter Regeln
- Automatisierter, rechnergestützter und in Echtzeit ablaufende Prozeß
  → Angriffs-/Einbruchserkennung: Englisch ‚intrusion detection' (ID)

Was ist ein 'Intrusion Detection System' und was ist es nicht?

- Offene Netzstrukturen -> keine verläßlichen Sicherheitsstandards
- Datenaufkommen in Netzwerken groß -> ID ist aufwendig (vom Volumen wie auch Komplexität)
- alternative, zeitversetzte Analyse der Protokolldateien bietet keine unmittelbare Gefahrenerkennung
  → Regeln online durch Computer prüfen lassen!
  → 'Intrusion Detection' Systeme (IDS)
- IDS: Art Alarmanlage für einzelne Rechner oder ganze Rechnernetzwerke
- IDS mit Abwehrfunktionalität → Intrusion Response' Systemem (IRS)

Abgrenzung: Intrusion Detection vs. Intrusion Response

- IDS erkennt Angriff nur und kann ihn nicht verhindert
- Abhilhe: das Perl-Skript „Guardian" durchsucht die Logfiles von Snort nach Attacken
  → Sperrung der IP des Angreifers mittels IPCHAINS oder IPTABLES
  → Timeout für die Sperre bzw. Ignore-Listen z.B. für Nameserver möglich
- Ziel: so wenig Fehlalarme wie möglich damit keine „unschuldigen" Rechner geblockt werden

Abgrenzung: Intruison Detection vs. Firewall

- Firewall ist aktiver ‚Torwächter', ein IDS nur Beobachter
- keine Integritätssicherung der Dateien, nur Aufspüren der gefährlichen Verhaltens möglich
- Komplementärer Einsatz: IDS kann auch nach innen schützen & Effizienz der Firewall prüfen

Anwendungsgebiete

Generell:

- Frage: gibt es bestimmte Kategorien von Systemen mit hoher Anwendungsaffinität?
- ‚zentraler' Einsatz am wünschenswertesten: böte höchstes Durchflußvolumen -> beste Kontrolle
  1) ohne bestimmtes Nutzungsmodell -> keine definierte Sicherheitspolitik -> was wäre dann unerwünschter Datenverkehr? -> Filter könnten nicht sinnvoll definiert werden
  2) Datenentropie an zentraler Stelle zu groß -> Filter müßten generell gehalten werden -> Angriffe gehen unbemerkt durch
  3) Datenaufkommen an zentralen Knoten zu groß -> selbst bei minimalster Filterung noch Unmengen an Protokolldaten, Warnungen -> Auswertung fraglich

→ ID System nur für Subnetze mit klarer Sicherheitspolitik und Zuständigkeiten!

IDS „Snort"

- besonders für kleine und mittlere Netzwerke
- nicht für große Systeme: Performanz und Management komplexer Sicherheitsrichtlinien nicht adäquat
- auch Einzelrechner Angriffsziel: speziell wenn permanent mit Netzwerk verbunden (z.B. DSL)
- könnten ausspioniert werden (persönliche Paßwörter, Bankverbindung, Kreditkartennummern...)
- eventuell Mißbrauch als Mittäter in 'Distributed Denial of Service' (DDoS) Attacken

Funktionsweise eines IDS

Datensammlung

- Voraussetzung 1: Netzwerkkarte im "promiscuous mode" – kann dann ALLE Pakete mitlesen!
- Voraussetzung 2: zentral im Netz plazieren; wenn Switch, an dessen Monitorport anschließen!

Sniffer:

- liest kontinuierlichen Strom an Paketen vom Netzwerk
- werden mit eingebauten Methoden selbst dekodiert
- Snort: kann TCP, IP, UDP und ICMP Pakete verarbeiten
- Daten werden nicht gespeichert, Ausgabe nur über Bildschirm - die Arbeitsweise ist transient
- sofortige Erkenntnisse aus dem aktuellen Datenstrom -> **reaktionäres ID:** Angriffserkennung während sie stattfinden
- Aber: externe Logik zur Bewertung des Datenstroms nötig, der Sniffer beinhaltet diese nicht!

Logger:

- Pakete sammeln und auf Platte speichern – Arbeitsweise auf Permanenz ausgelegt
- zeichnet zusätzliche Metainformationen (z.B. Zeitstempel des Durchlaufs) auf
- aufgezeichnete Daten nur in zeitlichem Zusammenhang, nicht in einem inhaltlichen!
- kontunierliches Aufzeichnen allen Datenverkehrs ist nur in bestimmten Fällen zweckmäßig
- erfordert enormen Speicherbedarf & Zeitaufwand zum Analysieren
- Verarbeiten von Logdateien im Nachhinein -> **präventives ID**: Maßnahmen für einen etwaigen nächsten Angriff treffen
- Zusätzlich auf Applikationsebene: Analyse der Vergabe von Betriebsmitteln an laufende Prozesse (belegter Speicher, Auslastung der CPU, Netzwerkverbindungen usw.)

Zur exemplarischen Anbindung von SNORT als ein Beispiel-IDS siehe Abbildung 1

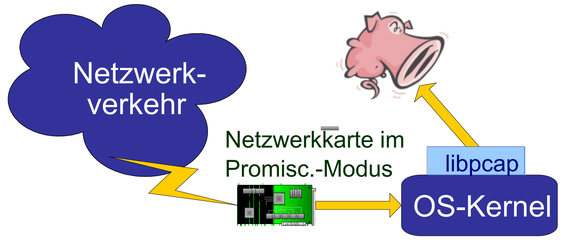

Abb.1: Datenfluß bei Analyse mittels SNORT

Datenanalyse

IDS mit zwei verschiedenen Zielen: Mißbrauchserkennung & Anomalieerkennung

Mißbrauchserkennung

- anhand vordefinierter Muster versucht das IDS Einbrüche zu erkennen
- mittels vordefinierter Signaturen und Pattern-Matching-Algorithmen werden einzelne Netzwerkpakete untersucht
- um einen Angriff erfolgreich identifizieren zu können, muß eine entsprechende Signatur vorliegen
- keine Signatur? -> ‚False Negatives': Pakete mit Angriffen werden unbehelligt durchgelassen
- Internet bietet eine genügend große Anzahl Angriffssignaturen -> Admin muß regelmäßig Updates einspielen
- nur Absicherung des Systems gegen bekannte Angriffe möglich
- nach diesem Paradigma arbeitet das Basissystem von Snort

Anomalieerkennung

- weiterhin problematisch: Angriffe, deren Angriffslogik bisher unbekannt war (also auch keine Signaturen)
- Techniken zur Identifizierung möglicher Attacken: Anomalieerkennung
- Def. Anomalie: jede Abweichung vom ‚normalen' Netzverkehr
- nicht jede Anomalie muß sofort eine Gefahr darstellen!
- nicht jeder Angriff wird sich als Anomalie darstellen! (z.B. unerwünschte Logins mit ausgehorchten Passwörtern)
- die Lernphase
  - Erlernen des ‚normalen' Systemsverhaltens -> längeren Einlaufzeit des System mit möglichst durchschnittlichen Kenngrößen
- der Betrieb:
  - statistischer Ansatz: Alarmauslösung bei Verlassen der definierten/erlernten Akzeptanzschwellen der überwachten Parameter
  - logischen Ansatz: Alarmauslösung bei Nichteinhaltung bestimmter zeitlicher Abfolge von Ereignissen
- Problem: schwierige Abwägungsfrage nach der Einlaufzeit und des ‚Lernmaterials'
  - zu kurz: keine ausgewogenen Daten des typischen Datenstroms -> Regeln zu strikt -> ‚False Positives': unbegründete Alarmmeldungen
  - zu lang: viele unterschiedliche Pakete mit wenigen Gemeinsamkeiten -> Regeln zu lax -> ‚False Negatives': Pakete mit Angriffen werden durchgelassen
- Systemadministrator muß hier also Fingerspitzengefühl aufweisen
- Anomalieerkennung hat Potential, nicht bekannte Angriffe aufzuzeichnen
- Präprozessor "spade" macht Snort fähig, Anomalien zu erkennen -> siehe nächstes Kapitel

IDS richtig plazieren! aka Wo das Schwein parken?

vor der Firewall: • überprüft hereinkommenden Verkehr auf etwaige Angriffe
• kann allerdings nicht sagen ob der Angriff erfolgreich war
hinter der Firewall: • erkennt Angriffe die von der Firewall nicht geblockt wurden
im internen Netz: • erkennt Angriffe im Inneren des Netzwerkes
• verringert die Gefahr des unberechtigten Zugriffs durch Mitarbeiter, Fremde, Würmer oder Trojaner

→ der dreifache Einsatz eines ISD ist sehr sinnvoll
→ strenge Sicherheitsrichtlinien sollten zwischen IDS und Firewall abgestimmt sein

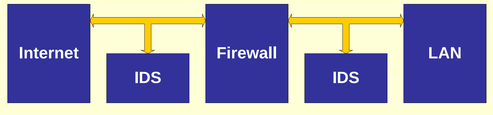

Abb.2: empfohlene Plazierung eines IDS

Zur Arbeitsweise von "SNORT"

Grundfunktionalität

- Decodieren der Netzwerkpakete & Prüfen derselben gegen bestimmte Regeln
- kein monolithisches System, kann durch Plugins (Präprozessoren & Module) erweitert werden

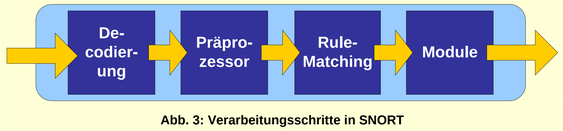

Abb. 3: Verarbeitungsschritte in SNORT

- vor dem eigentlichen Prüfen:
  - verschiedene sog. Präprozessoren einsetzbar -> der Datenstrom wird zusätzlich analysiert
  - mit Metainformationen angereichert -> Verwendung dieser in späteren Verarbeitungsschritten

frag2	setzt Paketfragmente zuammen, ideal für erfolgreichen Mustererkennung für zerstückelte Angriffsmuster
stream4	setzt TCP-Stream wieder zusammen und erlaubt so die Analyse eines vorher fragmentierten Angriffs → Invasion und Evasion
portscan	entdeckt Ausspähversuche nach offenen Ports
spade	Statistical Packet Anomaly Detection Engine - ist ein selbstlernendes Modul, das nach einer Lernphase anormalen Netzwerkverkehr feststellen kann
rpc_decode	defragmentiert RPC-Pakete
telnet_decode	trennt Telnet-Kontrollzeichen von den Sessiondaten
flow	Tracking um eindeutigen IPv4-Stream sicherzustellen
Performance Monitor	mißt die Echtzeit- und theoretische Maximumperformance
HTTP Inspect	grundlegende Analyse der HTTP-Pakete (benötigt ein vorgeschaltetes Zusammensetzungs-Modul)

Tabelle 1: Liste der bei SNORT mitgelieferten Präprozessoren

Das Prüfen

- Regelaufbau: aus Kopf (für grobe Filterung) und Option (für Feinjustierung)
- im Kopf: Regelaktion, Protokoll, Quell- und Ziel-IP-Adresse, Netzmaske, Quell und Zielport, Datenflußrichtung

alert	generiert einen Alarm und loggt das Paket
pass	ignoriert das Paket
activate	generiert einen Alarm und wendet weitere Regeln an
dynamic	loggt das Paket nur wenn durch „activate" aufgerufen
log	loggt das Paket

Tabelle2: mögliche Regelaktionen in SNORT

- als Option: Alarmmitteilungen und Information welcher Teil des Pakets noch weiter untersucht werden soll:
  - **meta-data:** Metainformation über die Regel - kein Effekt während der Analyse (z.B. Nachricht ausgeben)
  - **payload:** Optionen beziehen sich auf Paket selbst (z.B. Inhalt)
  - **non-payload: O**ptionen beziehen sich auf Metadaten des Pakets (z.B. Größe)
  - **post-detection:** Regel spezifischer Auslöser (z.B. Verkehr aufzeichnen)
- Vielzahl fertiger Regeln im Netz

Die Ergebnisweiterverarbeitung

Module

- sorgen für Weiterverarbeitung der Analyseergebnisse (z.B. Visualisierung, Adminalarm etc.)
- verschiedene Module hintereinander anwendbar
- Regelsystem ähnlich dem für Präprozessoren

Alert_xxxx	Warnhinweis in wahlweise Logdatei oder zu einem Port in wahlweise kurzer oder ausführlicher Form schreiben
Log_tcpdump Log_null	Pakete im tcpdump-Format in eine Datei schreiben nur Warnhinweise ausgeben, kein Paketmitschnitt
Database	Daten direkt in angeschlossene Datenbank schreiben
CVS	Warndaten in einfach importierbares Datenbankformat schreiben
Unified	Daten getrennt in Warndatei und Paketedatei schreiben (schnellste Ausgabemöglichkeit)

Tabelle 3: SNORT Aktionen bei vermutetem Angriff

Einige Bespielemodule

- **libnet**: ermöglicht Flexible Response, d.h. ermöglicht Reaktionen auf Angriffe wie z.B. Beendigung der TCP-Verbindung über RST-Paket oder ICMP („Net unreachable", „Host unreachable", „Port unreachable" oder alle 3 auf einmal) [http://www.packetfactory.net/libnet/]
- **oinkmaster**: ist ein Perl-Programm welches die Snort-Regeln auf dem neuesten Stand hält [http://oinkmaster.sourceforge.net/]
- **guardian**: ermöglicht Sperrung der Angreifer-IP-Adresse (auch zeitweise) [http://www.chaotic.org/guardian/]
- **acid**: PHP-Skripte zur grafischen Aufbereitung der Logfiles [http://acidlab.sourceforge.net/]
- **snortsnarf**: Perl-Skripte zur grafischen Aufbereitung der Logfiles [http://www.silicondefense.com/software/snortsnarf/]
- **Snort Webmin Interface**: ein Plugin für Webmin zur Konfiguration von Snort [http://msbnetworks.net/snort/]

Visualisierung der Ergebnisse

- nach einem erfolgten Angriff ist es elementar, die gesammelten Daten zu sichten, um sie für etwaige Maßnahmen auszuwerten
- Zusatzprogramme um die Daten grafisch aufzubereiten wie z.B.
  ACID: basiert auf PHP, greift auf Daten zurück die Snort in eine Datenbank loggt
  SnortSnarf: besteht aus Perl-Skripten die HTML-Output erzeugen (aus Logfiles oder einer Datenbank)

Praktischer Einsatz

Gefahren einer Fehlkonfiguration

- es ist aufwändig das System in das lokale Netzwerk einzuarbeiten (Standardverkehr des Systems erkennen, Definition der Akzeptanzschwellen)
- Anomalieerkennung erfordert eine relativ lange Eingewöhnungsphase (auch seltener Netzwerktraffic wie z.B. das monatliche Backup müssen beachtet werden)

Fehlalarme (False Positives):

- kann die Folge einer nicht ausreichenden Konfiguration sein, oder aber der Administrator hat sich auf an andere Netzwerkumstände angepaßte Signaturdatenbanken aus dem Internet verlassen
- kann generell durch Hinzufügen einer entsprechenden Signatur oder dem Anpassen der Anomaliedaten behoben werden

Angriffe die das IDS nicht erkennt (False Negatives):

- kann auftreten wenn dem IDS eine entsprechende Regel fehlt (ev. auch weil die Angriffsart zum Zeitpunkt der Einrichtung des IDS noch nicht bekannt war)
- der Administrator muß sich ständig über neue Angriffstechniken informieren

→ das IDS muß genauso wie eine Firewall ständig gewartet werden!

Testmöglichkeiten

- mit einem Paketgenerator (z.B. IP-Packet, kann der Regelsatz von Snort getestet werden)
- der Flaschenhals des IDS kann ermittelt werden (z.B. stick)
- aus Regelsätzen können Alarme erzeugt werden (z.B. mit snot)
- die gesammelten Daten müssen vom Administrator ausgewertet werden, umfangreiche Logfiles können u.U. nicht mehr vollständig überblickt werden

IDS als Schwachpunkt

- ID System soll möglichst unauffällig sein, sonst: selbst Angriffsziel!
- zwei Techniken, um IDS mit falsch erscheinenden Pakete zu verwirren
- beide Methoden problematisch vor allem bei regelbasierten IDS

Insertion

- IDS analysiert Pakete, die vom Zielhost nicht angenommen werden
- Angreifer 'adressiert' so Pakete extra für das IDS
- Beispiel: einzelne Pakete, die zusammengesetzt im IDS einen anderen String ergeben als im Zielsystem
- IDS analysiert: 'A''T''T''A''X''C''K'
- keine Erkennung vom IDS, da dies nur nach "ATTACK" sucht
- Paket "X" ist aber manipuliert -> Zielhost nimmt es nicht an
- "ATTACK" ist erfolgreich beim Zielhost angekommen!

Evasion

- IDS lehnt Pakete ab, die der Zielhost akzeptieren würde
- ähnliches Prinzip wie bei Insertion: Angreifer sendet 'A''T''T''A''C''K'
- Paket 'C' ist manipuliert -> IDS lehnt es ab
- IDS analysiert: 'A''T''T''A''K'
- keine Erkennung - kein Alarm!
- "ATTACK" ist erfolgreich beim Zielhost angekommen!

DoS Attacken

- Ressourcenerschöpfung – engl.: 'Resource Exhaustion'
- Angreifer verursacht eine Bedingung, die das IDS komplett beansprucht
- Angreifbare Ressourcen: CPU-Zeit, Speicher, Plattenplatz, Netzwerkkapazität
- z.B. provozierte Protokollüberläufe -> automatische Deaktivierung des IDS
- Mißbrauch Reaktiver ID Systeme
  - Angreifer bringt IDS dazu 'überzureagieren'
  - legitimer Verkehr passiert nicht mehr

'Normale' Angriffe

- Konfigurationsfehler
- der Mensch läßt sich täuschen → Social Engineering
- Passwortattacken
  - Ausspähung eines gültigen Benutzers und dessen Passwort
  - regulärer Login: IDS kann dies nicht erkennen! :(
  - danach möglicherweise Eindringverhalten -> durch IDS möglicherweise erkennbar
- daher: hohe allgemeine Vorsicht und Sicherheitsstandards besonders auf ID Rechner

Gegenmaßnahmen

- gängige Sicherheitsrichtlinien sollten eingehalten werden (sichere Paßwörter, Zugriffsrechte, etc.)
- Integrität der Konfigurationsdateien muß gesichert werden
- im Extremfall: separaten IDS-Rechner!

nach einem Angriff stellt sich die Frage was man mit den gesammelten Daten macht, z.B.

- zum Gegenangriff nutzen: man zieht die Aufmerksamkeit des Angreifers auf das IDS
- vorrübergehende oder dauerhafte Sperrung der IP-Adresse des Angriffsursprungs: es kann nicht garantiert werden daß der Angriff wirklich von der verwendeten IP-Adresse stammte (u.a. IP-Spoofing oder dynamische vergebene IP-Adressen kann Identifizierung des Angreifers stark erschweren)
- Sperrung des entsprechenden UDP/TCP-Ports
- Terminierung des betroffenen Programmes

→ Intrusion Response Systeme

- Einleitung rechtlicher Schritte, dazu siehe nächster Absatz

Rechtliche Probleme

- da die gesammelten Auditdaten nachträglich modifiziert und manipuliert werden können, stellen sie nach §416 der Zivilprozeßordnung kein rechtsverbindliches Beweismittel dar (wie z.B. ein notariell beglaubigtes Schriftstück)
- Daten unterliegen der freien Beweisführung, daher gleicher gerichtlicher Status wie eine Zeugenaussage (Beurteilung liegt im Ermessen des Gerichts)
- Lösung des Problems: das IDS muß die Daten mittels einer vertrauenswürdigen digitalen Signatur signieren, der entsprechende private Schlüssel muß sicher aufbewahrt werden