Einbruchserkennung in Netzwerke

mit Intrusion Detection Systemen und Honeypots

Chemnitzer Linux-Tage 2005

Stefan Schumacher, <stefan@net-tex.de>, PGP Key <0xB3FBAE33>

http://www.net-tex.de/net/ids.html

$Header: /home/daten/cvs/lectures/ids/folien.tex,v 1.13 2005/03/05 11:14:56 stefan Exp $

IDS?

- System zur Einbruchserkennung
- sammelt, analysiert, präsentiert Daten
- fängt Ereignisse ab die Einbrüche kennzeichnen
- **Teil** eines Sicherheitskonzepts
- **ergänzt** Zugangsschutz und andere Sicherungen
- Klassifizierung nach:

  Ort: Host, Netz, Hybrid

  Zeit: proaktiv, reaktiv, Post Mortem

Komponenten eines IDS

- Datensammlung und -aufbereitung
- Datenanalyse
- Datenpräsentation

Analyse

- Integritätsprüfung
- Signaturerkennung
  erkennt $bekannte$ Angriffsmuster
  Signaturen/Vorgangsabläufe, Pattern Matching
- Anomalieerkennung
  kann unbekannte Angriffe mittels Heuristiken erkennen
  statistische oder logische Analyse

Plazierung

- Firewall setzt Policy im Netz um
- IDS möglichst hinter Firewall, z. B. in DMZ
- Monitoring Ports auf Switches
- Kryptotunnel auf Hosts umgehen
- Vergleich vor/hinter Firewall möglich

Angriffe gegen IDS

- **Die üblichen Verdächtigen**
  Buffer Overflow, Race Condition, Privilege Escalation ...

- **Insertion**
  IDS akzeptiert Pakete, die das Zielsystem nicht annimmt
  erkennt Angriff nicht
  Zielsystem dropt Pakete und stirbt qualvollen Tod

Angriffe gegen IDS

- **Evasion**
  IDS dropt Pakete, die das Zielsystem annimmt
  erkennt Angriff nicht
  Zielsystem akzeptiert Pakete und stirbt qualvollen Tod

  Beide Attacken erfordern die Möglichkeit Daten in den Netzwerkverkehr einzuschleusen

Multilayer IDS

- “Data Warehouse” für Administratoren
- mIDS aggregiert Daten aus unterschiedlichen Quellen
- hat eigene Syntax und Semantik
- vereinfacht Analyse
- erzeugt überblickbare Datenbasis

mIDS Komponenten

- **Datensammlung:**

  Zusammführen aller Daten aus verteilten Logs
- **Datenaufbereitung:**

  verlustfreie Konvertierung in das mIDS-Datenformat
- **Analyse der aufbereiteten Daten:**

  Aggregation der Informationen um Beziehungen aufzudecken
- **Alarmierung / Reaktion:**

  Analyse der Daten bezgl. Angriffe

Anwendung: aide

- reaktives HIDS / Dateisystemintegritätsprüfer
- Datenbank mit Dateistatus & Prüfsummen
- ermöglicht Vergleich des aktuellen Systems
- deckt Manipulationen an Dateien auf (Forensik)

Anwendung: `verifiedexec(4)`

- Ab NetBSD 2.0 (1.6-current)
- proaktives HIDS
- Prüfsummenvergleich im `exec()` Pfad des Kernel
- Abhängig von Kernel Securitylevel
- verweigert Ausführung bei unpassender Prüfsumme

Anwendung: `snort`

- **Das** NIDS
- “Sniffer on steroids”
- Monitore im Netz sniffen Traffic
- untersucht Traffic gegen Angriffssignaturen
- Rules geben Reaktion vor
- Signaturdatenbank ist Herzstück

Honeypot

- täuscht Rechner/Netze vor, wird komplett überwacht

Ziele:

- Anlocken und Binden des Angreifers
- Erkennung und Analyse der Angriffe
- Schutz der Produktivsysteme als “Roter Hering”
- juristische Beweissicherung

Typen & Analysepunkte

- Tarpits/Traps
- physisches Sytem
- logisches System

- Angriffsquelle(n)
- Ausrüstung des Angreifers
- well known attacker?
- sequentielle/parallele Angriffe?

Angriffe

- Überwachung “echter” Systeme
  normale Systeme lassen sich schlecht überwachen und schützen
- Emulation der Realität
  logische Systeme müssen Reale ideal abbilden
- Determinismus von Tarpits
  vorhersagbares Verhalten

Honeypot Programme

LaBrea (Tarpit)
erzeugt logische Hosts mit IP-Adresse
verzögert TCP/IP Verbindung → bindet Scanner
honeyd
erzeugt logische Hosts/Netze
Verhalten bei Verbindung festlegbar
bspw: Tarpit, Skripte, REJECT

Fazit

- IDS und Honeypots sind wirksame Verteidigungsmittel
- umfassende Netz-/Protokollkenntnisse notwendig
- erfordern ausgiebige Wartung und Vorbereitung