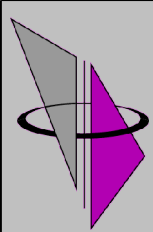


Was Wer Wie Wo tut? Intrusion Detection

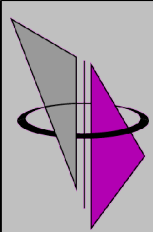
Bernhard Schneck,
GeNUA mbH

München, 2005-02-14



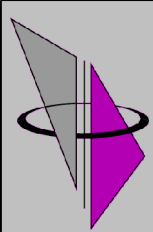
Themen

- Intrusion Detection Einführung
- Risiken
 - LAN und WLAN
- Abgreifstellen
 - LAN und WLAN
- GeNUDetect
 - Funktion, Struktur, weitere Planungen
- Beispiele



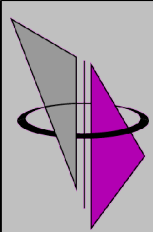
IDS Klassifizierung

- Herkunft der Daten
 - Host / Netz
- Art der Analyse
 - Muster / Anomalie
- topologische Abdeckung
 - Einzelsystem / Segment / Zentrale / Hierarchie



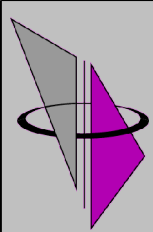
IDS Typen

- hostbasierte IDS
 - + geringerer Netzverkehr
 - + zusätzlich Zustand des Systems
 - - auf allen Rechnern verwalten
 - - sieht nicht alles
- netzbasierte IDS
 - + kann ganzes Segment überwachen
 - + sieht auch Scans auf unbenutzte Adressen
 - - hohe Bandbreite, Paketverlust?



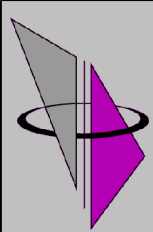
IDS Probleme

- hohe Datenmenge
 - Konzentration auf das Wesentliche
 - Datenreduktion möglichst nah an Quelle
 - Verlust von Informationen
- verteilte Datenquellen
 - mehrere Standorte
 - unterschiedliche Administratoren



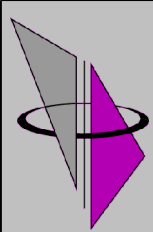
Risiken LAN

- Angreifer mit physikalischem Zugriff
 - geschützt durch Gebäude (hoffentlich!)
 - User oder Administratoren
 - sonstige Besucher?
- Angreifer mit logischem Zugriff
 - über Internet (nur ausserhalb Firewall / DMZ)
 - über andere bekannte Wege (z.B. Extranet)
 - über unbekannte Wege (OOPS!)



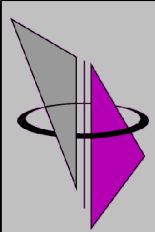
Risiken WLAN

- Zugriff auf WLAN Infrastruktur von Aussen
 - keine Kontrolle auf Layer 1 / 2
 - Abhörbar aus ≥ 10 km mit Richtantenne
- WLAN Verschlüsselung
 - WEP nicht „wire equivalent privacy“
 - LEAP geknackt (Exploit veröffentlicht in 04/2004)
 - aber besser wie garnix
 - **50%-90% ungesichert!**



Wireless IDS

- Wireless IDS ~ Wardriving Detection
- „normale“ Analyse auf Layer 3
 - braucht erst Assoziation mit AccessPoint
 - machen viele Scan-Tools nicht
- Analyse auf Layer 2
 - KISMET als Sensor-Typ
 - erkennt auch Netstumbler, void.11 & Co.
 - aber nicht rein passive Scanner (wie Kismet :-)
 - 802.11b sieht 802.11g, aber nicht 802.11a
 - Achtung, Chipset + Firmware muß stimmen!



IDS im LAN

- Positionierung der Sensoren

1. externes Netz

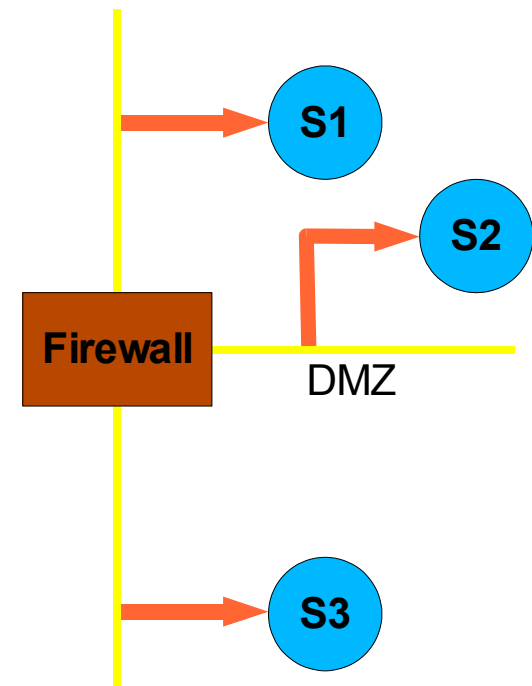
- viele wilde Alarme
- Firewall schützt meistens (?)

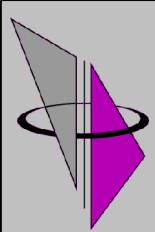
2. DMZ

- nur noch definierte Daten
- gute Qualität der Alarme

3. internes Netz

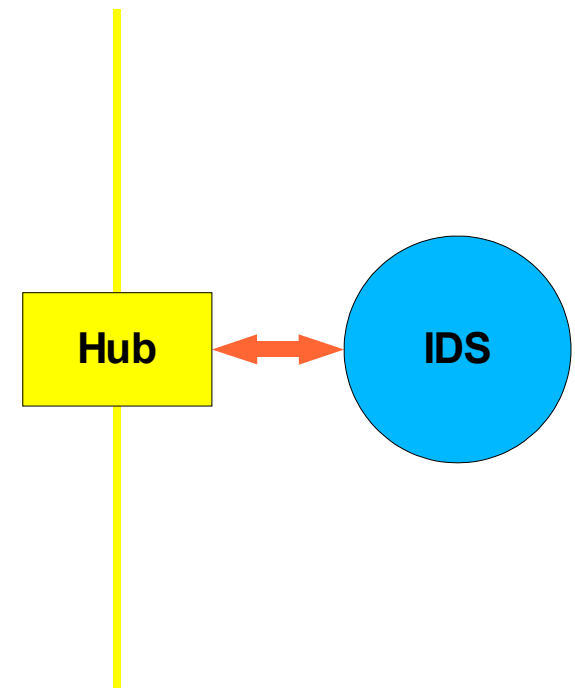
- geschwätzige LAN Protokolle
- hohes Risiko bei echtem Angriff





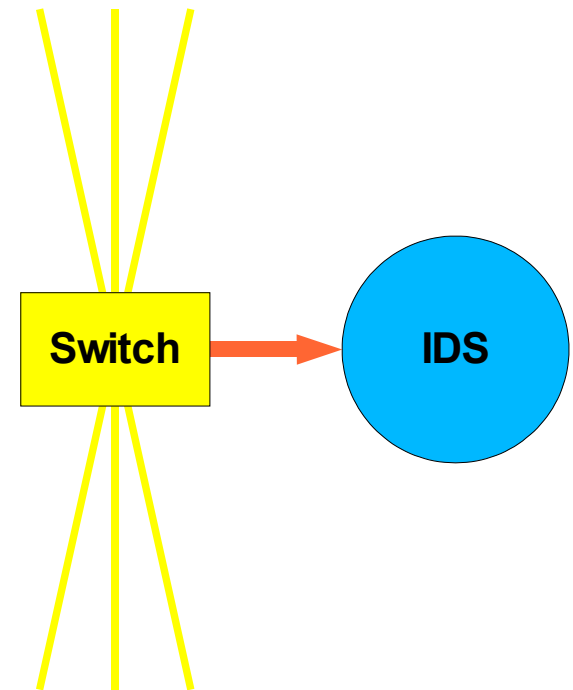
IDS Abgreifpunkte: Hub

- gibt's heut so gut wie nicht mehr
- TX-Leitung offen
- full duplex: nein



IDS Abgreifpunkte: Switch

- nur bei managebaren Switches
- full duplex: Puffer im Switch
- read only (normalerweise)

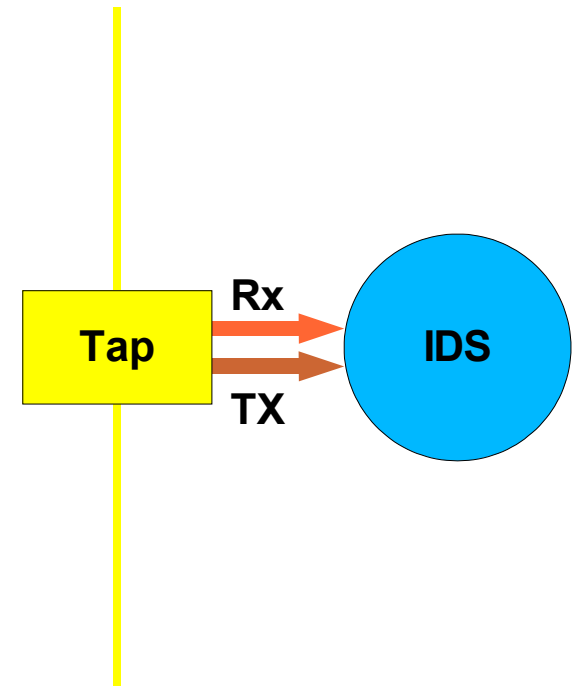


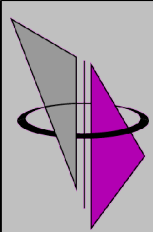
Beispiel Cisco:

```
monitor session 1 source  
  interface fa0/1 - fa0/24  
mon sess 1 dest int gi0/1
```

IDS Abgreifpunkte: Tap

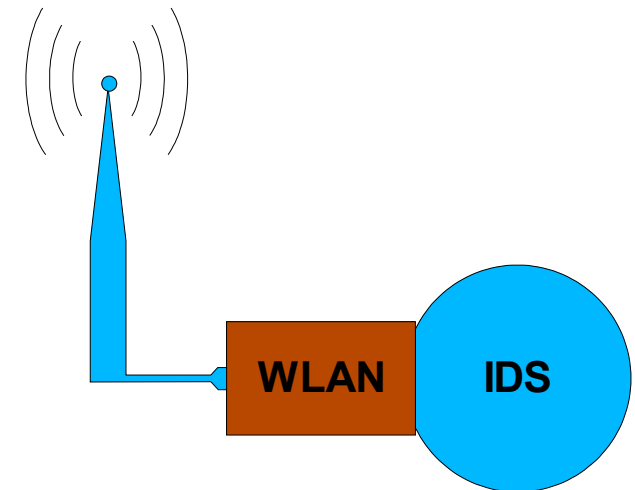
- read only
- full duplex
- 2 getrennte Interfaces am Sensor für RX/TX





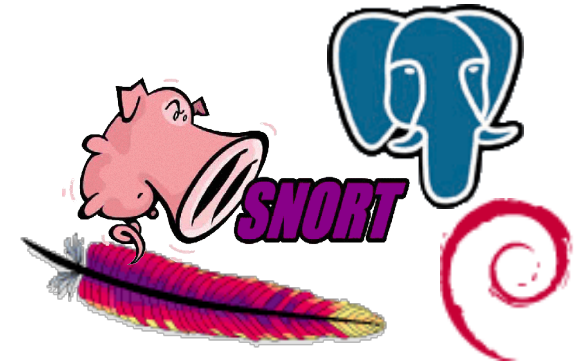
IDS Abgreifpunkte: WLAN

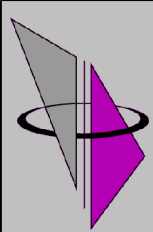
- Hardware
 - PCI oder PC-Card
 - Antennenanschluß
 - USB?
- Channel-Hopping
 - mehrere Interfaces?
- Chipset
 - PRISM/Hermes (802.11b)
 - Atheros (802.11a/b/g)



GeNUDetect

- Network Intrusion Detection System
- OpenSource
 - Debian/GNU Linux
 - Snort, Apache, Postgres
- GeNUA Software
 - System Management
 - Alert-Logik
 - GUI



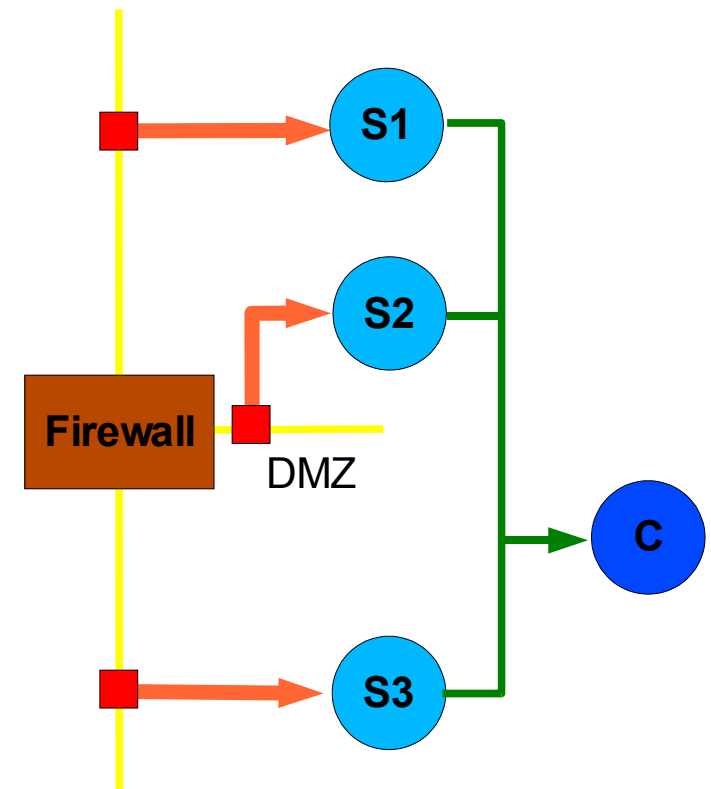


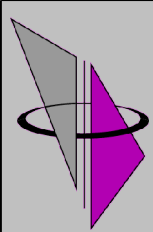
GeNUDetect Administration

- Installation
 - Central: Basissystem (Sensoren via PXE)
 - Definition der relevanten Signaturen
- Pflege
 - Updates der Signaturen
 - neue Angriffe
 - geänderte Umgebung
- Überwachung
 - false positives ignorieren
 - true positives verfolgen

GeNUDetect Netz

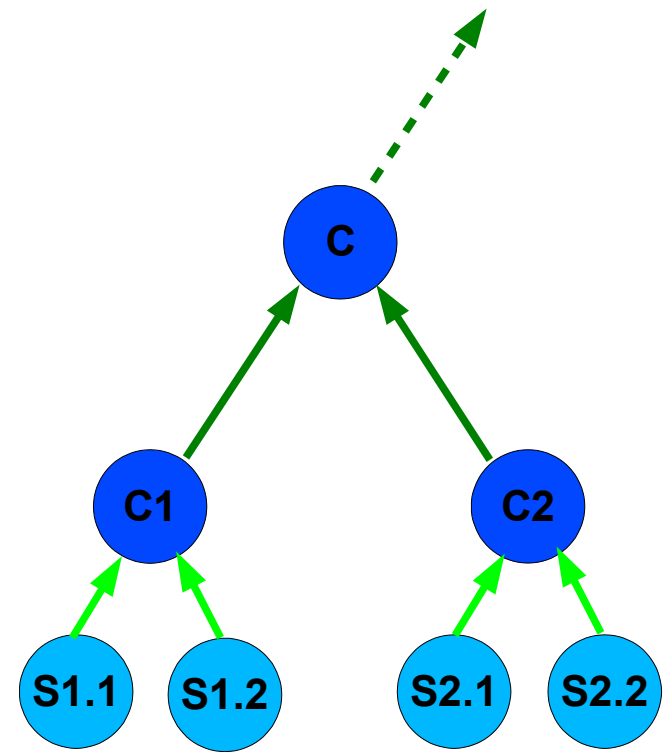
- mehrere Sensoren
 - verschiedene Netze
 - verschiedene Policies
- eine Central
 - Datenbank
 - Auswertung
 - Datenreduktion
 - Verwaltung
- eigenes IDS-Netz
 - umgeht Firewall (!)

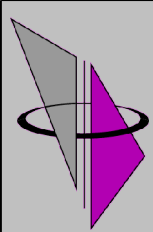




GeNUDetect Hierarchie

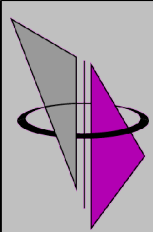
- mehrere Standorte
 - lokale IDS Netze
- Central pro Standort
 - Verwaltung Sensoren
 - Datenbank
 - evtl. weiterleiten an:
- Übergeordnete Central
 - (GOTO 2)





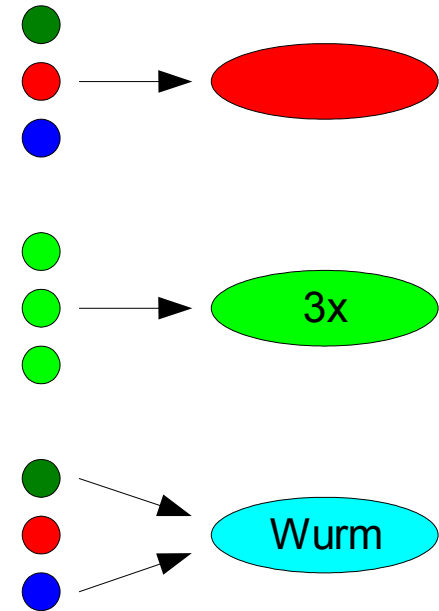
Events vs. Alerts

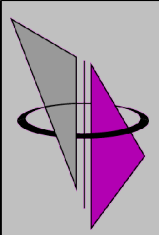
- Sensor meldet Event an Central
- Datenreduktion
 - abstrahierter Alert
 - enthält Referenzen auf Events/Alerts
 - Filter, Konsolidierung, (Korrelation)
- Central meldet (nur) Alert weiter
 - dort evtl. weitere Datenreduktion



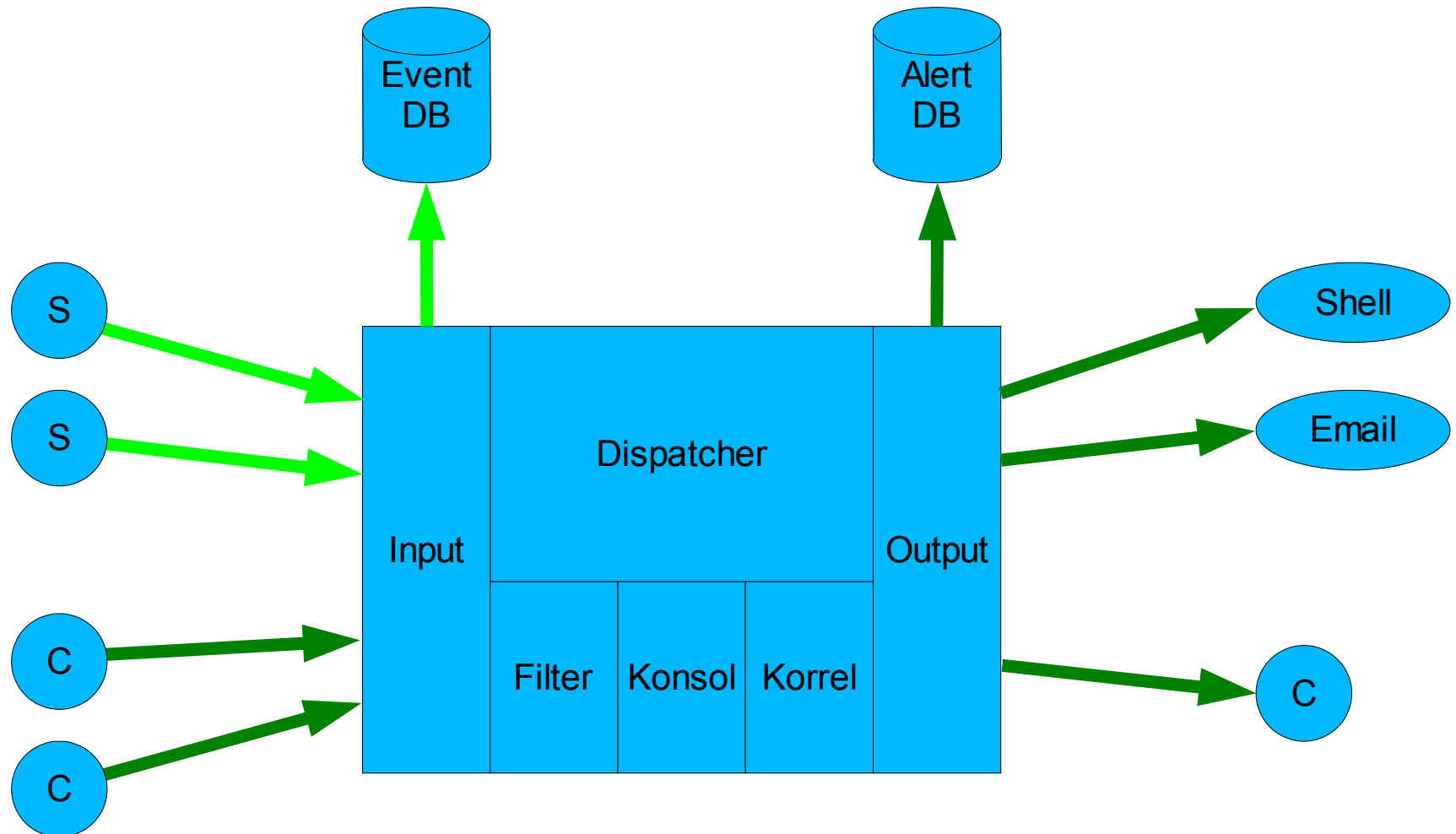
Datenreduktion

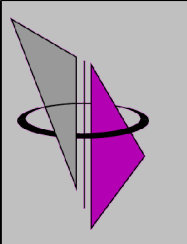
- Filterung
 - anhand Typen / Prioritäten
- Konsolidierung
 - mehrfache Events
- Korrelation
 - unterschiedliche Events
(noch nicht)





Dispatcher





Alert DB

- Home
- Search
- Alert Groups

Event DB

- Home
- Search
- Event Groups

Management

- Central
- Accounts
- Sensor & Policy
- Local Signatures
- Reports
- Status

Central Configuration

[Refresh](#)

- Parameters
- Tasks
- Hierarchy ?**
- Dispatcher
- Database

GeNUDetect Hierarchy

New Up Link

Central:	abcdef
Name:	Uplink 1
Prio:	1

||

Local Central (f31422)

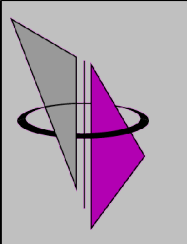
||

Central:	ce3e0a
Name:	genudetect2

||

Central:	123456
Name:	Downlink1

New Down Link



- Alert DB
 - Home
 - Search
 - Alert Groups

- Event DB
 - Home
 - Search
 - Event Groups

- Management
 - Central
 - Accounts
 - Sensor & Policy
 - Local Signatures
 - Reports
 - Status

Manage Alerts > Details

[Back](#)

Alert Detail

Delete Alert

Alert	
Message	EXPLOIT x86 Linux mountd overflow
Quantity	1
Occurance	From 2005-02-11 08:57:27 till 2005-02-11 08:57:27

Signature	
Message	EXPLOIT x86 Linux mountd overflow
Priority	12
Group	exploits
Info	bugtraq
	cve

References		
Events(s)	From local Sensor	EXPLOIT x86 Linux mountd overflow
	Quantity	1
	Duration	from 2005-02-11 08:57:27 till 2005-02-11 08:57:27

Back to alert list

Event Detail

Meta	
Signature	EXPLOIT ntpdx overflow attempt
Signature Group	Network Time Protocol (ntp)
References	bugtraq arachnids
Occurred at	2005-02-07 15:44:43
Priority	12
Generated from	Snort Engine
on sensor	ids1

[Configure Signature](#)

Ethernet

IP

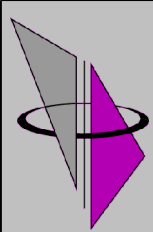
Version	Hdr Len	TOS	Total Length			
4	20 Bytes	0	157			
Identification			R0	DF	MF	Frag Offset
125 (0x007D)						0
TTL		Protocol		Checksum		
255		17 (UDP)		42169		
IP Source Address						
10.1.1.11						
IP Destination Address						
10.1.1.13						

UDP

Packet

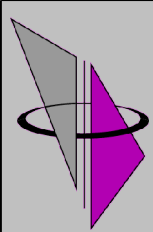
00:00	00D0 B7F0 7989 00D0 B7F0 78CD 0800 4500	...y.....E.
00:10	009D 007D 0000 FF11 A4B9 0A01 010B 0A01	...}.....
00:20	010D E5C6 007B 0089 30B4 D840 CD80 E8D9	...{..0..@...
00:30	FFFF FF2F 6269 6E2F 7368 0000 0000 0000	.../bin/sh.....
00:40	0000 0000 0000 0000 0000 0000 0000 0000
00:50	0000 0000 0000 0000 0000 0000 0000 0000
00:60	0000 0000 0000 0000 0000 0000 0000 0000
00:70	0000 0000 0000 0000 0000 0000 0000 0000
00:80	0000 0000 0000 0000 0000 0000 0000 0000
00:90	0000 0000 0000 0000 0000 0000 0000 0000
00:A0	0000 0000 0000 0000 0000 00 00

[Download TCPDump-File](#)



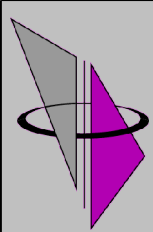
Erweiterungen

- **Auswertung**
 - **Event-GUI: Ersatz für ACID**
 - auch optimierte DB-Struktur
 - **Reports**
 - TOP-10 Auswertungen
 - regionale Herkunft (via AS/BGP Routing)
 - **Baselining (2.2)**
 - Unterdrückung von „unmöglichen“ Events
 - OS, Service, etc.



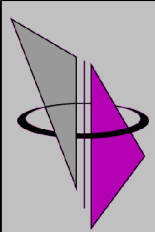
Erweiterungen (Sensoren)

- neue Typen
 - IPS Sensor (2.1, mit 802.1d ab 2.2)
 - Wireless Sensor (2.2)
- neue Varianten
 - Remote-Sensor (2.2)
 - mit eigener Festplatte und Netz-Verschlüsselung
 - Subsystem-Sensor (2.2?)
 - auf Linux-Host mit chroot/uml/vserver/xen
- Erweiterungen
 - „echtes“ GBit (>2.2)



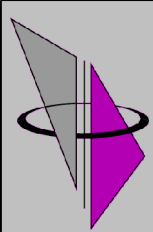
GeNUDetect: Beispiele

- **Beispiel 1: Großkunde ohne Firewall**
 - ca. 3000 Systeme am Internet
 - Sensoren an 2 Übergabepunkten zu Provider (jeweils 100MBit)
 - > 500000 Alerts pro Tag
 - Lösung: „Umkehrung“ der Regeln
 - nicht: wer greift uns an?
 - sondern: welche Systeme sind verseucht und greifen andere an?



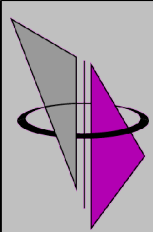
GeNUDetect: Beispiele

- **Beispiel 2: Großkunde mit Firewall**
 - ca. 20 Systeme direkt im Internet
 - ca. 100 Systeme in verschiedenen DMZs
 - 6 Sensoren, innen, aussen und DMZs
 - ca. 1PT pro Woche für Überwachung und Analyse



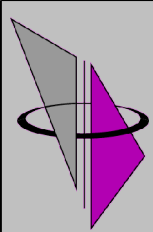
GeNUDetect: Beispiele

- **Beispiel 3: Tuning**
 - Testbed, ca. 100 Systeme aus Internet erreichbar
 - Begin: alle Rules enabled
 - ca. 100k Alerts / Tag
 - Anpassungen
 - Deaktivieren irrelevanter Rules (normaler PING)
 - Ignorieren falscher Angriffe (IIS Exploit auf Apache)
 - PASS Rules für zulässigen Verkehr
 - usw.
 - Ende: ca. 100 Alerts/Tag



GeNUDetect: Beispiele

- Kunde hat kommerzielle WWW-Applikation
 - Reverse Proxy in DMZ, Server intern
 - Anbieter hat offiziellen Weg für Fernwartung
 - ssh, normalerweise abgeklemmt, etc ...
- Applikation hat Backdoor
 - erst nach Login, aber ohne SSL
- Anbieter benutzt diese für Administration
 - GeNUDetect klingelt
 - Kunde was „not amused“



Links

- IDS
 - <http://www.snort.org/>
- WLAN Sniffing
 - <http://www.netstumbler.com/>
 - <http://www.kismetwireless.net/>
- Wireless IDS
 - <http://snort-wireless.org/>
- GeNUDetect
 - <http://www.genua.de/>

Vielen Dank



Mehr Infos
auf der
CeBIT 2005