Intrusion Detection Systeme
Intrusion Monitoring & Anomaly Detection

25. September 2002

PARADIGMA

René Pfeiffer

rene.pfeiffer@paradigma.net

Paradigma Unternehmensberatung GmbH

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Wichtiger Hinweis:

Die Logfileauszüge dieses Vortrags enthalten zum Teil „echte“ IP Adressen und Hostnamen, die von Providern oder anderen Personen in Verwendung sind. Ich bitte diesen Umstand nicht als Anschuldigung zu verstehen oder daraus Maßnahmen oder Empfehlungen abzuleiten. Die Beispiel-Logs wurden bereits ausgewertet und in Einzelfällen wurden Schritte unternommen bzw. wurde der entsprechende Vorfall in Abstimmung mit der zutreffenden Security Policy behandelt. Ich bitte das Erscheinen der IP Adressen in keinster Weise als Bewertung, Kritik oder Anschuldigung zu sehen. Weiterhin bitte ich darum, diese IP Adressen keinen speziellen Untersuchungen wie Portscans, Security Audits oder ähnlichem ohne Zustimmung des Eigentümers zu unterziehen.

1 Intrusion Detection Systeme als intelligente Firewall

```
BOUNDARY, n.
   In political geography, an imaginary line between two nations,
   separating the imaginary rights of one from the imaginary
   rights of the other.

 --- "The Devil's Dictionary", Ambrose Bierce
```

1.1 Begriffsdefinitionen

- **Client**
  Ein Client ist eine Maschine oder eine Applikation, die sich an einen Server wendet, um einen dort zur Verfügung gestellten Dienst / Service in Anspruch zu nehmen. *Faustregel: Ein Client ist eine Maschine, die niemand vermißt.*

- **Dienst / Service**
  Ein Dienst oder Service ist eine Applikation, die Informationen zur weiteren Verarbeitung entgegennimmt oder zur Verfügung stellt.

- **Firewall**
  Eine Firewall ist eine *Sammlung von Maßnahmen*, die zum Schutz eines oder mehrerer Netzwerke eingesetzt werden. Elemente einer Firewall trennen vertrauensunwürdige Netzwerke von Netzwerken mit höherem Sicherheitsbedarf.

- **Internet / das Internet**
  Das Internet stellt eine riesige Menge aus Netzwerken dar, die miteinander und untereinander verbunden sind (*inter-connected networks*) und über Internetprotokolle Daten austauschen.

- **internet / ein Internet**
  Ein Netzwerk aus 2 oder mehr Maschinen.

- **Protokoll**
  Als Protokoll bezeichnet man eine Sammlung von Konventionen und Regeln, die eine strukturierte Sprache zum Zweck der Kommunikation verschiedener Teilnehmer bilden. Dieser Mechanismus ist für das Austauschen von Daten zwischen zwei Punkten unerläßlich.

- **Router**
  Ein Router ist eine Maschine, die eine Verbindung zwischen verschiedenen Netzwerken herstellt.

- **Server**
  Ein Server ist eine Maschine oder eine Applikation, die bestimmte Dienste den Clients im Netzwerk zur Verfügung stellen.

1.2 Zusätzliche Aufgaben einer Firewall

- **Network Intrusion Detection Systems (NIDS)**
  beobachten Pakete, die über das Netzwerk gesendet und empfangen wurden
  - Nachvollziehbarkeit von Vorfällen
  - Archivierung von Netzwerkaktivität
  - erfordert unter Umständen ein eigenes Netzwerk von Sonden
  - Logging alleine reicht nicht
    $\longrightarrow$ *Auswertung und Monitoring ist kritischer Punkt*
- **Host Intrusion Detection Systems (HIDS)**
  - beobachten Pakete, die direkt an eine bestimme Maschine gehen.
  - extrahieren Angriffsspuren aus den Logs
- **System Integrity Verifiers (SIV)**
  überwachen kritische Bereiche eines laufenden Systems (Benutzerrechte, Binaries, Konfigurationen, etc.)
  - Systeme werden bei Installation mit Signatur versehen
  - „Fingerabdrücke“ von Binaries durch Checksummen
  - Vergleich mit archivierten Signaturen
- **Log File Monitore (LFM)**
  durchsuchen Logfiles in regelmäßigen Abständen nach Anomalien
- **Täuschen - Deceptions Systems**
  Vereiteln von Portscans, Ausgabe von Falschinformationen, Umschreiben der Mailheader

1.3 Sinn und Zweck zusätzlicher Aufgaben

- **Gegenprüfen der Paketfilter**
  - Portscanner und Packet Shaper als „Aggressoren“
  - „vergessene“ Ports
  - Logging erzeugt Protokoll, welches den Zustand wiedergibt
- **Aufspüren von Attacken durch Kanäle, die die Firewall passieren**
  $\longrightarrow$ *Anomalien im scheinbar normalen Betrieb*
- **Festhalten von fehlgeschlagenen Attacken**
  $\longrightarrow$ *Ermittlung von Trends bei den Eindringversuchen*
- **Sonden in interne Netzwerke**
  LANs sind nicht mehr als sichere Netzwerke zu betrachten!
  - skriptfähige Mail User Agents
  - skriptfähige Programme mit Schnittstellen zu anderer Software oder dem System
  - skriptfähige Web-Browser
  - eingeschleuste Trojaner & Viren
  - allerlei Instant Messaging Clients
  - ...
- **Qualitätskontrolle des Sicherheitskonzepts**

1.4 Vergleichen von Intrusion Detection Systemen

```
"The history of benchmarking and testing is rife with examples
 of deliberately and accidentally flawed testing methodologies."
   --- Marcus J. Ranum, NFR Security, Inc.
```

- **Welche Methode setzt das IDS ein?**
  $\longrightarrow$ *Pattern Matching*
  $\longrightarrow$ *statistische Analyse*
- **Was schaut sich das IDS an?**
  $\longrightarrow$ *Client oder Server Pakete*
  $\longrightarrow$ *Packet Reassembly Methode*
  - Defragmentierung von Paketen
  - Umordnen von Paketen
  - TCP Stream Reassembly
  - State Tracking
- **Problematik mit simuliertem Netzverkehr**
  - Zufallspakete sind für den Normalbetrieb nicht aussagekräftig
  - Paket Replays mit regulärem Verkehr sinnvoller
- **Bewertung von Tests bedarf Expertise**

Quelle: Experiences Benchmarking Intrusion Detection Systems[1]

[1] http://www.snort.org/docs/Benchmarking-IDS-NFR.pdf

Entwickler	Produkt	Beschreibung
Cisco Systems, Inc.	Cisco IDS (ehemals NetRanger)	Hard-/Software, NIDS, Web Server HIDS, Management Console
Internet Security Systems	RealSecure	Software, NIDS, Management Console
Intrusion Inc.	SecureNet Pro™	Software, NIDS
Intrusion Inc.	SecureNet Provider	Software, Management System
Intrusion Inc.	SecureNet Security Appliances	Hardware für Einsatz mit SecureNet Pro™
Samhain Labs	Samhain / Yule	Software, HIDS, Agent und Logserver
Snort / Sourcefire Inc.	Snort	Software, NIDS, Management Console, Analysetools
Symantec Corporation	Intruder Alert™	Software, HIDS mit Agents und Management Console, Policy Monitor
Symantec Corporation	NetProwler™	Software, NIDS mit Agents und Management Console (MS Windows)
Tripwire, Inc.	Tripwire für Server	Software, HIDS, Agent für MS Windows / UN*X
Tripwire, Inc.	Tripwire Manager	Software, HIDS Management Console für MS Windows / Solaris

Tabelle 1: Ein kurzer Einblick in Produkte auf dem IDS Markt. Ein detailierter Vergleich der Produkte ist nicht ohne überlegte Testkriterien und ohne Abstimmung mit dem Einsatzgebiet möglich.

1.5 Markteinblick IDS Hersteller

1.6 Tools of the Trade

- Ethereal[2] Packet Analyzer
  Protokollanalyse, Fehlersuche, Netzwerkdiagnose
- hping2[3] Paketgenerator
  Generieren gezielter Testpakete, TCP Ping auf abgesicherte Hosts, Testen von Sensoren
- isic[4] (**I**P **S**tack **I**ntegrity **C**hecker) Paketgenerator
  simpler Lasttest, Prüfen von Firewall Regeln (insbesondere ICMP Blocks)
- Nessus[5] Security Scanner
  Streßtest für Applikationen, automatisierte Checks mit stationären Nessus Servern in zu überwachenden Netzwerken (Nessus ist skriptfähig)
- nmap[6] Port Scanner
  schneller Überblick über verwendete Ports und Filter, Streßtest für Netzwerke
- Perl Skripte
  universeller Einsatz in der Systemadministration für vielfältige Aufgaben
- SING[7] (**S**end **I**CMP **N**asty **G**arbage) Paketgenerator

[2] http://www.ethereal.com/
[3] http://www.hping.org/
[4] http://www.packetfactory.net/Projects/ISIC/
[5] http://www.nessus.org/
[6] http://www.insecure.org/nmap
[7] http://sourceforge.net/projects/sing

1.7 Ethereal - Packet Analyzer

- **Abfangen und Darstellen von Paketen auf jedem Interface**
- **Darstellen von Paketen aus Dumps von anderen Tools**
  `tcpdump`, NAI Sniffer/Sniffer Pro, NetXray, LANalyzer, Shomiti, AIX `iptrace`, RADCOM WAN/LAN Analyzer, Lucent/Ascend Produkte, HP-UX `nettl`, Toshiba ISDN Router, ISDN4BSD `i4btrace`, Microsoft Network Monitor, Sun `snoop`
- **Abspeichern von Dumps in verschiedenen Formaten**
  libpcap (tcpdump), Sun `snoop`, Microsoft Network Monitor, NAI Sniffer
- **Filterkriterien & Suchfunktionen**
- **Protokolldekoder für eine Vielzahl von Protokollen**[8]
- **TCP Stream Verfolgung**
- **TCP Analyse**
  - Sequenznummeranalyse (Zeit-Sequenznummer Graph)
  - Datendurchsatz Graph
  - RTT Graph

[8] http://www.ethereal.com/docs/user-guide/appfiltfields.html

1.8 hping2 - Paketgenerator

`hping2`[9] ermöglicht es beliebige ICMP, TCP oder UDP Pakete an einen Host zu senden

- Testen von Packetfilterregeln
- Port Scannen (auch mit gefälschten Source-Adressen, „Spoofing“)
- Testen der Netzperformance durch Variation
  - der Packetgröße
  - des Protokolls
  - der Type of Service (TOS) Kennung
  - der Fragmentierung
- MTU Ermittlung
- Traceroute mit verschiedenen Protokollen
- File Transfer (auch durch Firewalls hindurch)
- Pakete mit ungültiger Checksumme

[9] `http://www.kyuzz.org/antirez/hping/`

1.9 isic - Zufallspaketgenerator

isic, *icmpsic*, *tcpsic*, *udpsic* und *esic* sind Tools zum Testen der Integrität eines IP Stacks

- „kontrolliertes“ Generieren von IP Paketen mit zufälligem Inhalt
- Quell- und Zieladresse werden festgelegt (nicht erforderlich)
- Parameter in den Headern werden zufällig generiert
- Pakete können fragmentiert werden
  Angabe eines Prozentsatzes steuert Anteile
- Streßtest für Firewalls und manche NIDS
  - Gauntlet Firewall Denial of Service Attack
    `http://www.securityfocus.com/bid/556`
  - Axent Raptor Denial of Service Vulnerability
    `http://www.securityfocus.com/bid/736`

1.10 SING - ICMP Packetgenerator

SING (**S**end **I**CMP **N**asty **G**arbage)

- ICMP[10] Handling gibt Aufschluß über OS
  $\longrightarrow$ `http://www.sys-security.com/archive/papers/ICMP_Scanning.pdf`
  $\longrightarrow$ `nmap` benutzt ähnliche Tests
- SING sendet
  - fragmentierte ICMP Pakete (Linux und BSD)
  - Pakete mit Längen > 65534 (Linux und BSD)
  - beliebige Pakete mit veränderten Parametern und Codes
    * Parameter Problem
    * Destination Unreachable
    * Time Exceeded, etc.
- Senden unter Verwendung von Source Routing Optionen

[10] RFC 792, 1122, 1256, 1822

1.11 nmap - Network Mapper

- **Ursprung: OS Identifikation anhand des TCP/IP „Fingerabdrucks“**
- **Scan-Arten umfassen**
  - IP Protokoll Scan (nmap V3.00 erkennt 144 IP Protokolle)
  - TCP connect() Scan
  - TCP SYN, FIN, Xmas oder NULL Scan
  - TCP Scanning per FTP Proxy (Bounce Attack)
  - SYN/FIN Scannen mit IP Fragmenten
  - TCP ACK und Window Scannen
  - UDP Scannen (ICMP Port Unreachable)
  - ICMP Scannen (Ping Sweep, Timestamp, Netmask)
  - direktes RPC Scannen (nicht über `rpcinfo`)
    nmap V3.00 erkennt 451 SunRPC Dienste
  - OS Identifikation
  - Ident[11] Scannen bei TCP connect()
- **Patch erlaubt Versions-Identifikation der Dämonen**
- **aktiv unterstützte OS Fingerabdruck Datenbank**
  nmap V3.00 erkennt über 700 Systeme
- **Angabe von Wildcards beim Ziel-Host**
  z.B. `192.168.0.*`, `212.17.1-230.54-168`
- **XML Output zur Verwendung in anderen Tools**
- **Vielfalt an Optionen bezüglich Timing für bessere Performance**

[11] Identification Protocol, RFC1413, `http://dungeon.luchs.at/RFC/rfc1413.txt`

1.12 Network Intrusion Detection mit Snort

Die Fähigkeiten von Snort[12] umfassen

- **Real-Time Traffic Analyse**
- **Analyse von aufgezeichnetem Netzwerkverkehr**
- **Schreiben von Log-Daten in externe SQL Datenbank**
  normaler Betriebsmodus arbeitet mit Logs im Dateisystem
- **frei programmierbar durch Rule Sets**
  $\longrightarrow$ *beliebige Definition von Alerts*
  $\longrightarrow$ *Konfigurieren von automatischen Benachrichtigungen*
- **Flexible Response Option**
  Snort kann auf bestimmte Pakete mit einer Reihe von Antworten reagieren
- **Schnelligkeit**
  - Entkoppeln von Detektieren und Auswerten
  - Erfassen von 100 Mbit/s Link mit 80 Mbit/s

  Es gibt noch weitere Anstrengungen die Rate zu verbessern und schnellere Netzwerke zu beobachten.
- **Stealth Modus**
  Snort benötigt keinen TCP/IP Stack auf dem System

[12] http://www.snort.org/

1.13 Snort Filterregeln

- **Grundfunktionen** `alert / log / pass`
- unterstützt derzeit TCP, UDP & ICMP
  In Zukunft geplant: ARP, IGRP, GRE, OSPF, RIP, IPX
- **Snort kann die folgenden Paketinformationen testen**
  - `TTL` - Wert des IP Pakets
  - `ID` - IP Header Fragment ID
  - `DSIZE` - Datenlänge des IP Pakets
  - `Content` - bestimmter Inhalt in den Daten des IP Pakets (Pattern Matching)
  - `Flags` - TCP Flags
  - `SEQ` - TCP Sequenznummern
  - `ACK` - TCP ACK-Nummer
  - `Session` - beobachtet einzelne Sessions (Telnet, rlogin, FTP, HTTP)
  - `IType` - ICMP Typ
  - `ICode` - ICMP Code
  - `ICMP_Id` - ICMP Echo ID
  - `ICMP_Seq` - ICMP Echo Sequenznummer
  - `IPoption` - IP Options
    - rr - Record Route
    - eol - End of list
    - nop - No op
    - ts - Time Stamp
    - sec - IP security option
    - lsrr - Loose source routing
    - ssrr - Strict source routing
    - satid - Stream identifier
  - `RPC` - RPC Service/Applikations Aufrufe
- **Flexible Response -** `resp`
  Snort kann eine sich aufbauende Verbindung trennen

1.14 Snort Flexible Response

- **Senden von TCP-RST** an Empfänger, Sender oder beide
- **Senden von ICMP Nachrichten an den Sender**
  - ICMP Network Unreachable
  - ICMP Host Unreachable
  - ICMP Port Unreachable
- **Gegenmaßnahmen mit Checkpoint Firewall-1 über Snortsam**[13]
  - Liste von IPs, die nicht geblockt werden sollen
  - Kontrolle über Zeitintervalle
  - Rollback Support zur Aufhebung von Blocks
  - verschlüsselte Two-Fish Kommunikation zwischen Snortsam und Firewall-1
  - Plugin-Möglichkeit für andere Firewalls

Damit ist es möglich auf bestimmte Kriterien und Datenpakete zu reagieren.

[13] http://www.snortsam.net/

1.15 Samhain / Yule - Client/Server HIDS

Samhain[14] ist ein HIDS Client mit den folgenden Eigenschaften:

- **Integritätscheck des Filesystems**
  - kryptografische Checksummen von Dateien
  - Prüfung auf SUID Programme
  - Erkennung von Linux Kernel Modul Root Kits
- **Selbstschutz des Clients**
  - signierte Konfigurationsdateien und Datenbankeinträge
  - signierte Logfileeinträge und Alarmmeldungen (EMail)
  - Client kann sich am Host „verstecken“
- **zentrales Monitoring**
  - verschlüsselte & authentifizierte Client/Server Verbindung
  - Checksummen und Clientkonfigurationen sind am Server abgelegt
  - HTML Statusseiten für jeden Client
  - unlimitierte Anzahl von Clients
- **Programme laufen als Dämon Prozesse**
- **Login/Logout Monitoring**

[14] http://www.la-samhna.de/samhain/index.html

1.16 Logging - Aufspüren von Unregelmäßigkeiten

- **Kombinieren von mehreren Logfiles verschiedener Maschinen**
  - Telefonlogs
  - `utmp` und `wtmp` Logs für Login-Zeiten
  - Prozeß Accounting Logs
  - Shell History Logs
  - `syslog`, NT Ereignisse, MTA Logs, etc.
  - Logs von Intrusion Detection Tools
  - Logs von anderen Sniffern & Sonden
- **Betriebsdaten via SNMP**
- **zeitliche Korrelation ist wichtig (Zeitserver)**
- **möglichst vielschichtig loggen, damit man die Integrität testen kann**
  - IP-Adresse **und** Hostname
    Forward und reverse DNS; Logging von IP-Adresse „fälschungssicherer“
  - MAC- und IP-Adresse
    Monitoring Tools für ARP Aktivität

1.17 Anomalien - Was sind Unregelmäßigkeiten?

- **Performanceveränderungen**
  $\longrightarrow$ *CPU Last, E/A, Plattenplatz*
- **Zustandsänderungen („Phasenübergänge“) von Systemen**
- **Interaktionen - Logins**
  - zu ungewöhnlichen Zeiten
  - von bestimmten Benutzern
  - von bestimmten Maschinen
- **ausbleibende Status Reports von Subsystemen**

Probleme:

- *Welche Log Informationen sind Indikatoren? Welche nicht?*
- *Wie sieht der „normale“ Betriebszustand aus?*
- *Wie schauen schlechte Nachrichten* wirklich *aus?*

1.18 Kalibrierung und Planung

- **Wie schauen die 10 häufigsten Logmeldungen im Normalbetrieb aus?**

```
cat /var/log/{messages,maillog} \
| sed -e "s/^... .. ........ $HOSTNAME //" -e "s/\[[0-9]*\]:/:/" \
| sort | uniq -c | sort -nr > /tmp/uniq.sorted
```

- **Wie schauen die 10 häufigsten Logmeldungen im „Ernstfall“ aus?**
- **Was bedeuten diese Meldungen?**
- **Gibt es Beispieldaten, die man für eine Analyse heranziehen kann?**
- **Kann man den Betrieb eines Servers/Netzwerks in Statistiken abbilden?**
  - Mails pro Zeiteinheit
  - Transaktionen bzw. E/A Zugriffe pro Zeiteinheit
  - Pakethistogramme, Datendurchsatz
- **Wie werden Daten abgelegt?**
  $\longrightarrow$ *XML, SQL, Rohform*

1.19 Gezielte Fragen an Intrusion Detection Systeme

- **Welche Applikationen sollen überwacht werden?**
- **Welche Teile des Netzwerks sollen überwacht werden?**
- **Gibt es genug Expertise in house für die Betreuung?**
  $\longrightarrow$ *IDS müssen betreut werden*
  $\longrightarrow$ *Administratorn müssen eingebunden werden*
- **Wie groß darf das Backlog werden?**
  $\longrightarrow$ *Festlegung eines Zeitfensters*
  $\longrightarrow$ *Wahl geeigneter Auswertemethoden*
- **Wie schnell muß aus den Daten ein Alarm generiert werden?**
  $\longrightarrow$ *Anforderungen an die Auswertung*
- **Möchte man NIDS Daten als Sonden für Trends einsetzen?**
  $\longrightarrow$ *Warnungen vor bevorstehenden Attacken*

1.20 Zustandsgesteuertes Paketfiltern unter Linux

- **Linux Netfilter ist zustandsgesteuert**
  verfügbar im Kern der 2.4.x Serie[15]
- **besseres ICMP Filtern**
  Filter kann ICMP Pakete zu bestehenden Verbindungen zuordnen
- **Connection Tracking**
  $\longrightarrow$ *Filtern von FTP und ähnlichen Protokollen*
  $\longrightarrow$ *Handhabung von Packetfragmenten*
  $\longrightarrow$ *Zuordnen von geblockten Paketen zum ursprünglichen Paket*
- **Transparentes Ändern des IP Headers**
  $\longrightarrow$ *Markieren von Paketen*
- **Quality of Service (QoS)**
- **Rate Limiting**
  - Einstellen von Paketraten zum Schutz vor Paketfluten
- **modularer Aufbau mit Erweiterungen**
  - Filtern nach MAC Adresse
  - MSS Clamping
  - Filterregeln per User Account
  - Sanity Checks mit verschiedenen Kriterien
  - Schnittstelle zu Paketfilter im User Space

[15] http://netfilter.samba.org/

1.21 Einsatzgebiet von Linux Paketfiltern

- **Router mit einem Subset von Filterregeln**
  - Netzwerkbereiche stellen verschiedene Anforderungen an Filter
  - Aufteilen der Filterfunktionen
  - begünstigt mehrstufige Verteidigung
- **interner Paketfilter zwischen LANs**
- **Paketfilter zwischen Standorten**
  - Einsatz von VPN Technologien (z.B. CIPE[16], IPsec[17], SSL, SSH)
  - On Demand Router (ADSL, ISDN, GSM/GPRS/UMTS)
  - IPV6 Support
- **Server Firewall**
  - Regeln können pro Benutzer Account eingestellt werden
  - netzwerkseitiges Beschränken von Applikationen
- **Embedded Systems**
- **Kommerzielle Firewalls sind vorhanden**
  - Astaro Security Linux[18]
  - Mandrake Single Network Firewall 7.2[19]
  - SuSE Linux Firewall on CD[20]

[16] http://sites.inka.de/~W1011/devel/cipe.html
[17] http://www.freeswan.org/
[18] http://www.astaro.com/
[19] http://www.mandrakesoft.com/products/snf
[20] http://www.suse.de/de/products/suse_business/firewall/index.html

1.22 Einsatzgebiet von Linux als Content Filter

- **Squid[21] ist ein Web & Reverse Proxy**
  - Web Proxy für Browser als Clients
  - Reverse Proxy vor Web Servern
    - Load Balancer
    - Entlastung des Web Servers bei statischem Content
- **Koppelung mehrerer Squids als Parent/Child Cluster**
- **Benutzung von externen Programmen als URL Filter**
  - URL Request wird vom Squid nach Prüfung der ACLs angenommen
  - URL wird an ein externes Programm weitergegeben
  - externes Programm kann URL beliebig modifizieren
  - Squid holt tatsächlich die URL, die der Filter zurückgibt

  Einsatz zum Schutz von Web Servern

[21] http://www.squid-cache.org/

1.23 Squid als Reverse Proxy

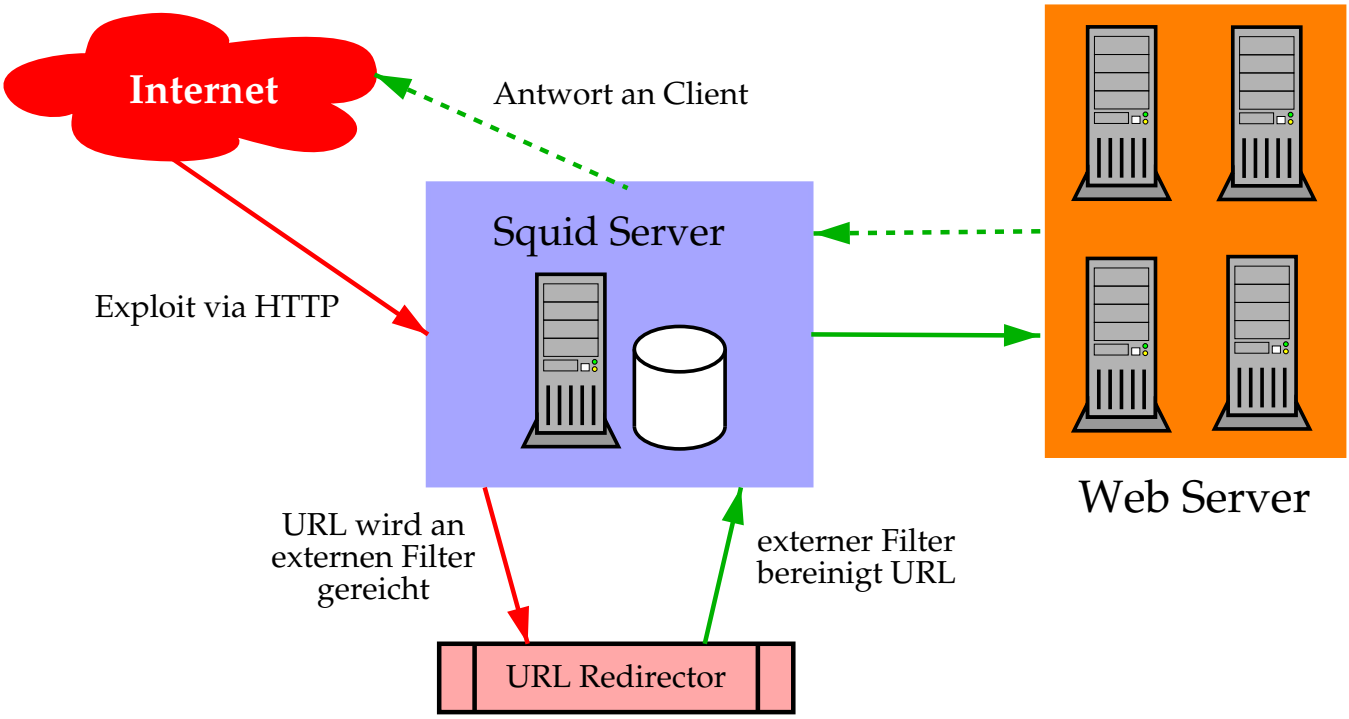

Abbildung 1: Ein Squid als Reverse Proxy mit URL Filtern vor einer Reihe von Web Servern. Das externe Filterprogramm am Squid prüft URLs auf Attacken und leitet diese um bzw. blockt sie.

1.24 Snort als Firewall mit Paketinspizierung

- **Hogwash[22] ist modifizierter Snort Engine**
- **Hogwash filtert Pakete nach IDS Signaturbeschreibungen**
  - `pass` - Paket darf passieren
  - `drop` - Paket wird blockiert und Alarm wird verzeichnet
  - `sdrop` - Paket wird blockiert
  - `alert` - Paket darf passieren und Alarm wird verzeichnet
  - `log` - Paket wird aufgezeichnet
- **Filter schützt Applikationen vor Attacken**

```
drop tcp $EXTERNAL_NET any -> $HOME_NET 23
(msg:"BACKDOOR Telnet Freebsd 3x4x"; flags: A+; content:
"LoUSUCKS"; reference: arachnids,519;)
```

- **Hogwash bedarf kein System mit IP Stack**

[22] http://hogwash.sourceforge.net/

1.25 Funktionalität versus Komplexität - Gleichgewicht der Kräfte

- **technische Möglichkeiten sind extrem vielfältig**
  $\longrightarrow$ *Gefahr von übermäßig komplexen Maßnahmen*
  $\longrightarrow$ *Propagierung von Universalmedikamenten*
- **„Domino Theorie“**
  - Firewall- und IDS-Implementationen können eine Kette von Abhängigkeiten bilden
  - Ausfälle in Teilbereichen können IDS daher unbrauchbar machen

  - Ausfallsicherheit kann Aufwand für Systemadministration erhöhen
- **Mächtigkeit von Paketfiltern bringt neue Risiken**
  - Paketfilter besteht aus Software, die Bugs haben kann
  - Paketfilter sollten robust implementiert sein
  - „Feature Freeze“ ist auch für Systemadministration sinnvoll

2 IDS in komplexen Netzwerken

CERBERUS, n.
   The watch-dog of Hades, whose duty it was to guard the
   entrance -- against whom or what does not clearly appear;
   everybody, sooner or later, had to go there, and nobody wanted
   to carry off the entrance. Cerberus is known to have had
   three heads, and some of the poets have credited him with as
   many as a hundred. Professor Graybill, whose clerky erudition
   and profound knowledge of Greek give his opinion great weight,
   has averaged all the estimates, and makes the number
   twenty-seven -- a judgment that would be entirely conclusive
   if Professor Graybill had known (a) something about dogs, and
   (b) something about arithmetic.

--- "The Devil's Dictionary", Ambrose Bierce

2.1 Architektonische Überlegungen

- **Ausdehnung des Netzwerks**
  - Standorte und Standleitungen
    $\longrightarrow$ *insbesondere VPNs*
  - Zuständigkeitsbereiche
- **Einteilung der Systeme in Gruppen bzw. Sicherheitsstufen**
  - lokale Netze
  - Perimeternetze / DMZ
  - Funktionen der Server
    $\longrightarrow$ *Sichten der aktiven Applikationen*
- **Festlegen von Intrusion Detection Zonen**
  $\longrightarrow$ *Isolieren der Zugänge*
- **Planung des Logdatentransports**
  $\longrightarrow$ *zentrales Log Repository*
  $\longrightarrow$ *Absicherung der Transportkanäle zum Repository*

2.2 NIDS Zonen

- **unsichere Netzwerke**
  - hohe Rate von Alarmen und Fehlalarmen
  - High Risk Zone
  - Sensitivität von NIDS muß gering sein
- **Perimeternetzwerk**
  - mittlere Rate von Alarmen und Fehlalarmen
  - Netzwerkverkehr ist vorgefiltert
  - Sensitivität von NIDS muß mit Umgebung abgestimmt sein
    $\longrightarrow$ *Verwenden von Signaturen, die zu Applikationen passen*
- **lokale/vertrauenswürdige Netzwerke**
  - geringe bis mittlere Rate von Alarmen und Fehlalarmen
  - in der Regel keine Paketfilter zwischen den Clients
  - NIDS kann nach eingeschleusten Risiken schauen
    $\longrightarrow$ *Überwachung der Security Policy*
  - höchster Anspruch an Performance

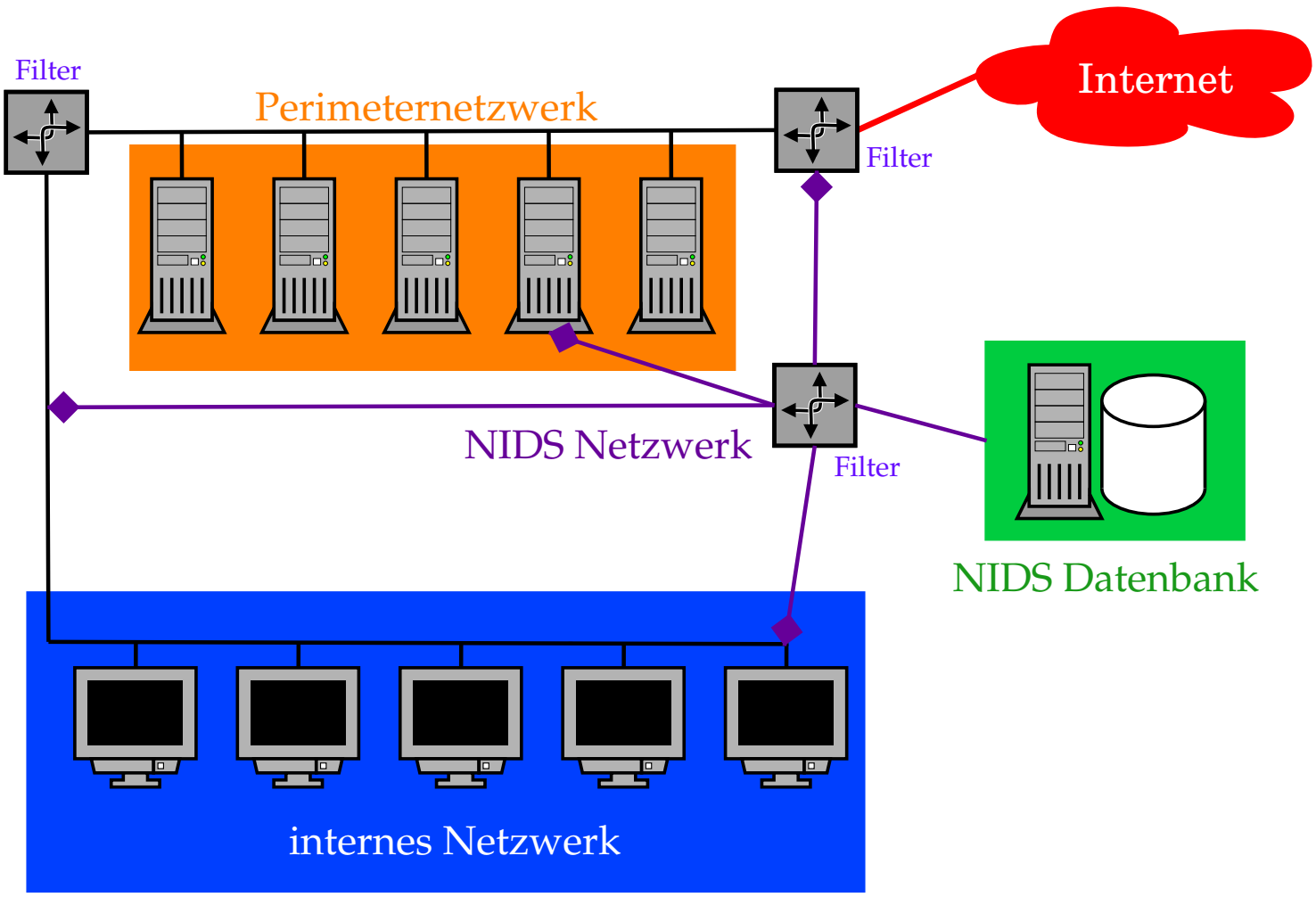

Abbildung 2: Schematischer Einsatz von Snort Sensoren in einem Netzwerk. Man kann Sensoren auf Paketfiltern, Servern oder auf speziellen Sondenservern unterbringen. Der sinnvollste Einsatz geschieht auf einer eigenen Maschine, die ausgewählten Netzwerkverkehr über einen dafür abgestellten Port an einem Switch erhält.

2.3 Methoden zur Netzwerküberwachung

- **Einsetzen von HUBs in Segmente**
  - *leicht zu implementieren, keine besondere Konfiguration*
  - *Paketkollisionen steigen, Performanceverlust*
- **Switch Port Analyzer (SPAN) Port**
  - *keine weitere Hardware nötig, Infrastruktur bleibt unverändert*
  - *begrenzte Anzahl von Ports pro Switch, NIDS nur passiv*
  - *Paketverlust bei großem Netzwerkverkehr*
- **Network Test Access Ports (TAPs)**
  - *kein Performanceverlust, keine Störung am Netzwerk*
  - *NIDS muß im Stealth Mode arbeiten*
  - *zusätzliche Kosten durch TAP Hardware*
- **Abfrage von Kenndaten der Applikation selbst**

2.4 SPAN Port Implementation

- **meist gibt es nur einen SPAN Port pro Switch**
- **Überwachen mehrerer Ports problematisch**
  - mehrere Ports meist nur duch Erstellung von VLANs möglich
  - Überlastung des SPAN Ports möglich
    insbesondere im full-duplex Modus
- **Performanceprobleme bei hoher Last am Switch**
- **manche SPAN Ports sind nicht bidirektional**
  $\longrightarrow$ *aktives Terminieren von Sessions nicht möglich*
- **Verlust von fehlerbehafteten Paketen**
  - überlange oder zu kurze Pakete
  - Pakete mit Checksummenfehler

  Switchlogik kann daher durch Korrekturen Bild des Netzwerks verfälschen

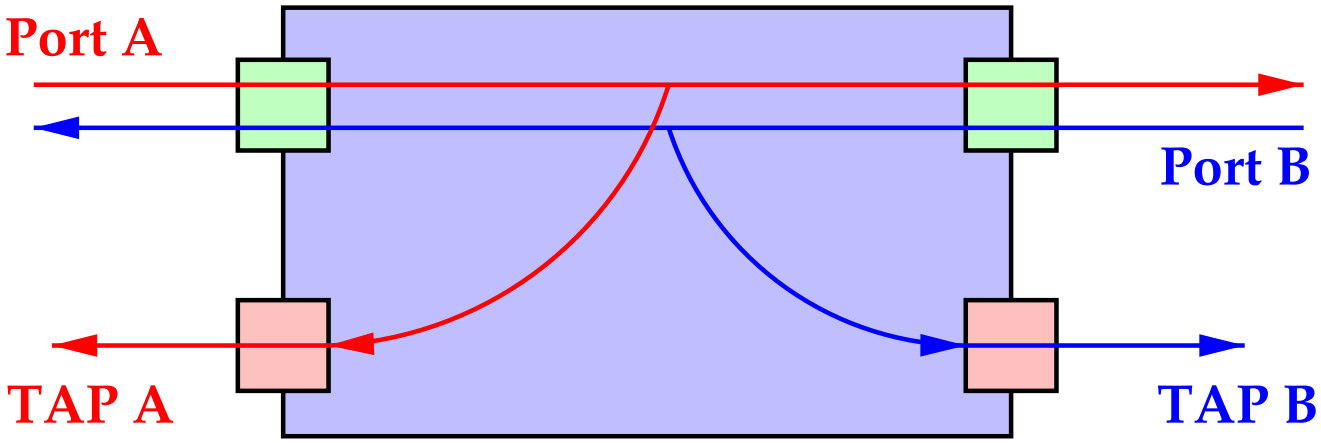

Abbildung 3: Das Diagramm zeigt die schematische Ansicht eines Network TAPs. TAP A fängt Pakete von an Port A angeschlossenen Geräten ab, TAP B welche von an Port B angeschlossenen Geräten. Eine Kombination von TAP A und TAP B ergibt den gesamten Netzwerkverkehr. (Quelle: Internet Security Systems)

- **TAP Elektronik wird transparent, wenn Strom ausfällt**
  $\longrightarrow$ *verringerte Fehleranfälligkeit*
- **Verfügbar für 10/100/1000 Mbit/s Netzwerke**
- **keine Beeinflussung des Netzverkehrs**
  $\longrightarrow$ *kein Performanceverlust*
- **IDS kann Paketanomalien untersuchen**
  $\longrightarrow$ *überlange oder zu kurze Pakete*
  $\longrightarrow$ *Checksummenfehler*

2.6 Attacken gegen NIDS Implementationen

Sichtbare IDS Sensoren sind angreifbar:

- **Sensor mit Paketflut blenden**
  - IP Pakete mit gefälschten Quelladressen
  - fragmentierte Pakete
  - Versuch den Sensor durch Überlastung auszuschalten (CPU, Storage, Alarmrate)
  - Mischen von Attacken und Paketflut zur Täuschung
- **langsame Proben und Portscans**
  - Proben können sich über Tage und Wochen hinziehen
  - Detektierung durch manuelle Auswertung unmöglich
- **koordinierte Attacken von verschiedenen Ausgangspunkten**
- **wechselnde Muster bei Proben und Attacken**
  $\longrightarrow$ *manche Attackvektoren erlauben Variationen*
  $\longrightarrow$ *gute Definition und Wartung von Signaturen notwendig*
- **Verwenden von nicht-standard Ports**
  oft wird ein Protokoll nur nach Portnummer identifiziert

Quelle: `http://www.robertgraham.com/pubs/network-intrusion-detection.html`

2.7 Überlegungen zur Performance

- **Abstimmen der Signaturen**
  $\longrightarrow$ *je weniger Checks, desto schneller der Sensor*
  $\longrightarrow$ *NIDS Regeln unbedingt auf Einsatzort abstimmen*
- **Verteilung der Sensoren auf das Netzwerk**
- **Verteilung der Aufgaben auf mehrere Sensoren**
  $\longrightarrow$ *Aufteilung von Protokollen (Load Balancer, Layer 7 Switch)*
- **Ändern der Default Timeouts für verschiedene Protokolle**
  $\longrightarrow$ *geringeres Timeout für High Volume Protokolle (z.B. HTTP)*
  $\longrightarrow$ *Anpassen der Session Timeouts für TCP*
  $\longrightarrow$ *sauberes Terminieren von Verbindungen (TCP RST, ICMP)*
- **Filtern von unerwünschtem Netzwerkverkehr**
  $\longrightarrow$ *an Routern (ACLs)*
  $\longrightarrow$ *an Sensoren und Logservern*

2.8 Mehrstufige Architektur für Hochleistungs IDS

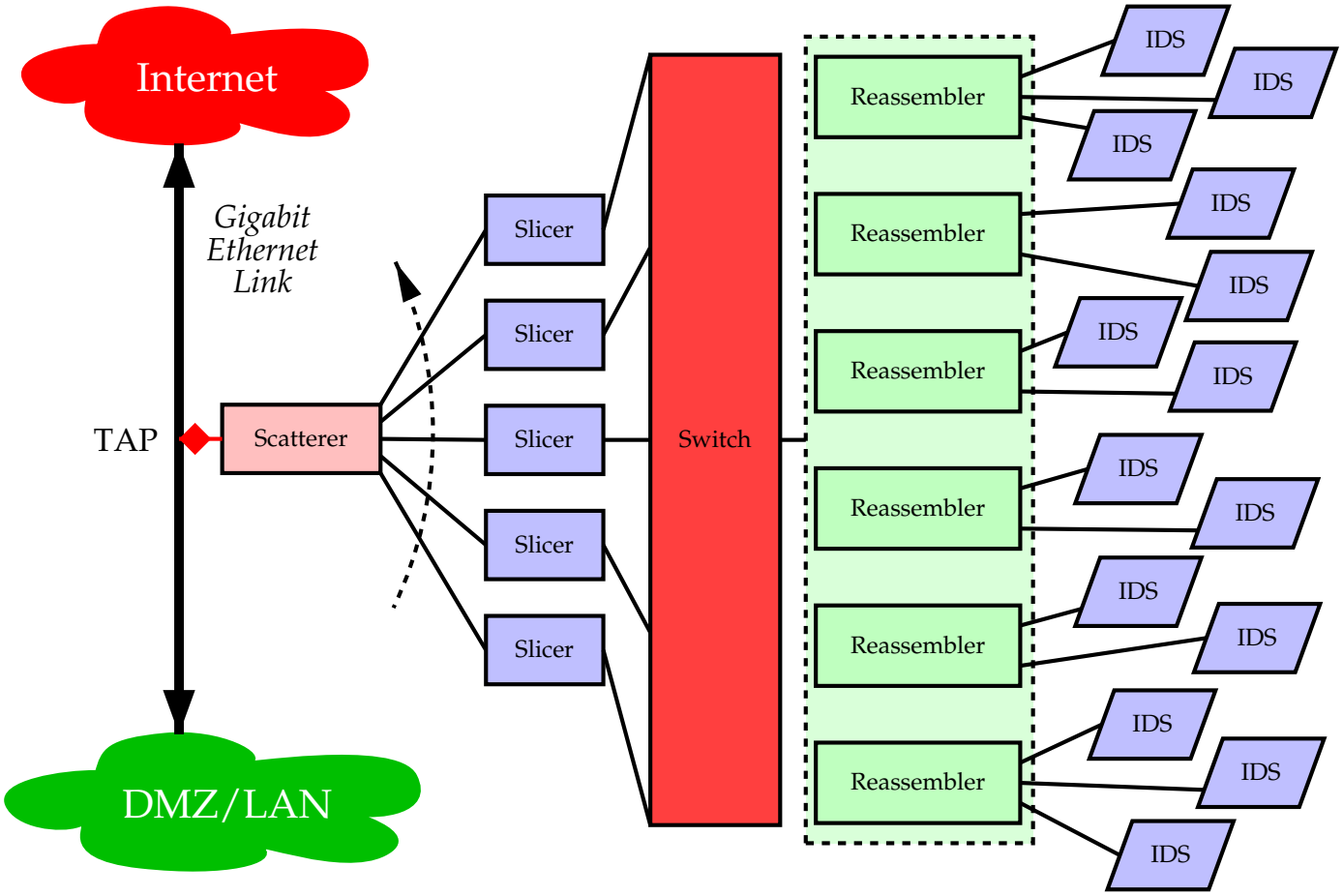

Abbildung 4: Schema eines verteilten Intrusion Detection Systems für Hochleistungsnetze. Ein Scatterer greift alle Daten eines Zeitabschnitts ab und verteilt sie nach einem Splitting Algorithmus auf verschiedne Slicer. Jeder Slicer sortiert nun die einzelnen Ethernet Frames an einen oder mehrere Sensorkanäle. Die hinter dem Switch liegenden Reassembler sorgen für die richtige Reihenfolge der einzelnen Frames. Jeder Sensor bzw. jede Sensorgruppe analysiert nur bestimmte Angriffszenarien, um die Last zu verteilen. (*Quelle:* Stateful Intrusion Detection for High-Speed, University of California, `http://www.computer.org/proceedings/sp/1543/15430285abs.htm`)

2.9 Zentralisieren der Konfiguration

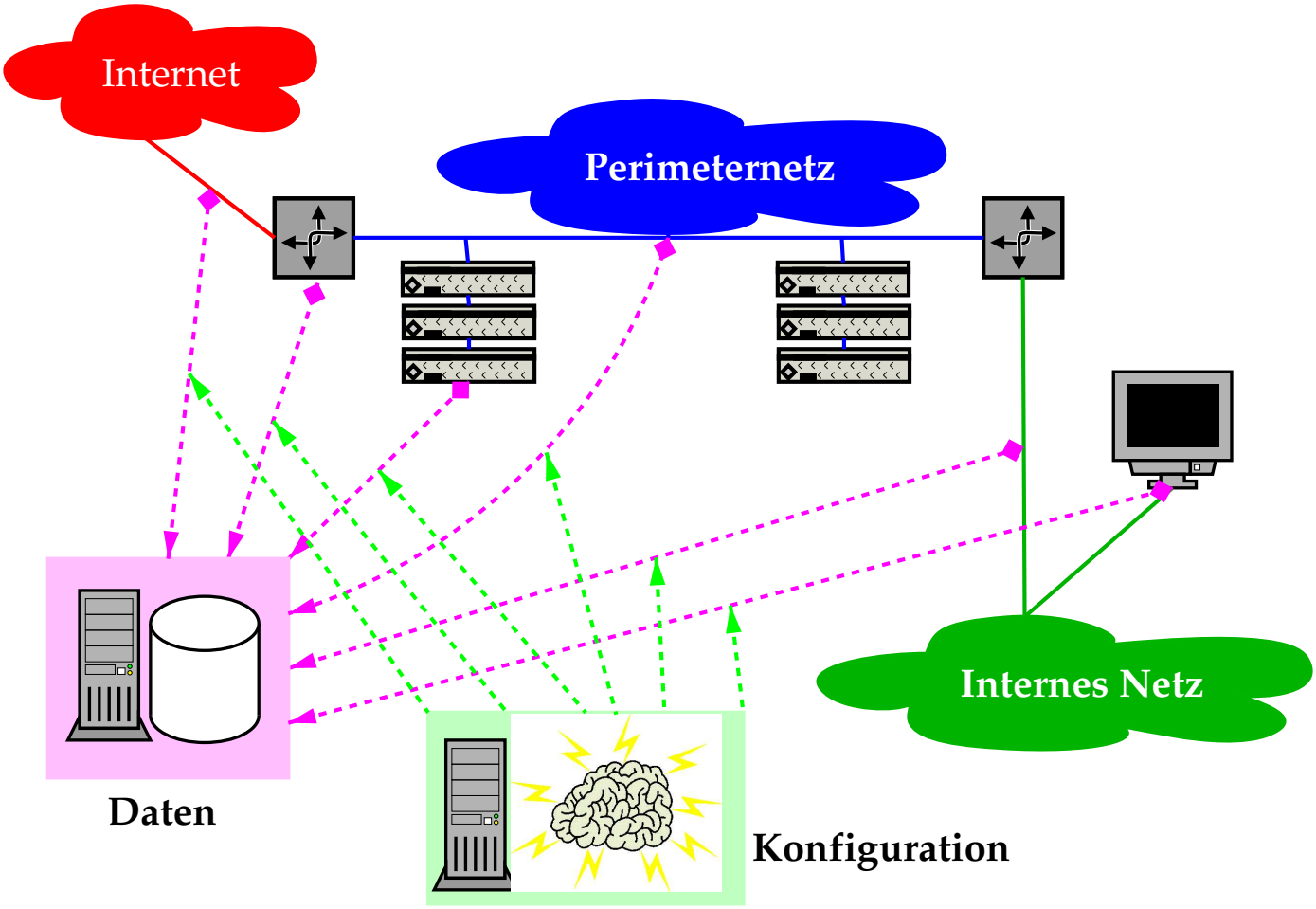

Abbildung 5: Ausgedehnter Einsatz von IDS Maßnahmen erfordert ein gewisses Maß an zentralisiertem Sammeln der Daten. Dies gilt ebenso für die Verteilung der Konfigurationen an die verschiedenen Sensoren.

2.10 Konsolidierung

- **IDS werden in bestehenden Netzen meist nachgerüstet**
  $\longrightarrow$ *Nutzung bestehender Log Infrastruktur*
  $\longrightarrow$ *Platzierung von HIDS auf exponierte Systeme*
  $\longrightarrow$ *NIDS hinter Paketfiltern*
- **Wahl der eingesetzten Systeme gut planen**
  $\longrightarrow$ *Interoperabilität zwischen Herstellern*
  $\longrightarrow$ *Kompatibilität der generierten Alarme & Daten*
- **Verzicht auf High-Speed IDS**
  - ähnlicher Ansatz wie bei Paketfiltern
  - High-Speed IDS Infrastruktur derzeit komplex
- **Wahl von Lösungen, die plattformunabhängig sind**
  $\longrightarrow$ *leichtere Migration auf vereinheitlichte Hardware*
  $\longrightarrow$ *Beibehaltung der Konfiguration*

2.11 Performance-schonende Lösungen

- **Real Time Auswertung stellt höchste Ansprüche an IDS**
  $\longrightarrow$ *Pattern Matching beansprucht CPU*
  $\longrightarrow$ *Abgreifen der Pakete muß zuverlässig sein*
- **Einsatz von TAPs und performanten Sensoren**
  $\longrightarrow$ *gezielt an „Hot Spots“*
- **Teilen der zu überwachenden Ereignisse**
  - Festlegen von kritischen Überwachungsbereichen
  - Sammlung der anderen Daten in größeren periodischen Abständen
- **Data Warehouse Ansatz - bestehende Informationen nutzen**
  - oft existieren schon Informationsresourcen
  - statistische Analyse und Zusammenführen der Daten

2.12 Hybrid-IDS mit Datenbankauswertung

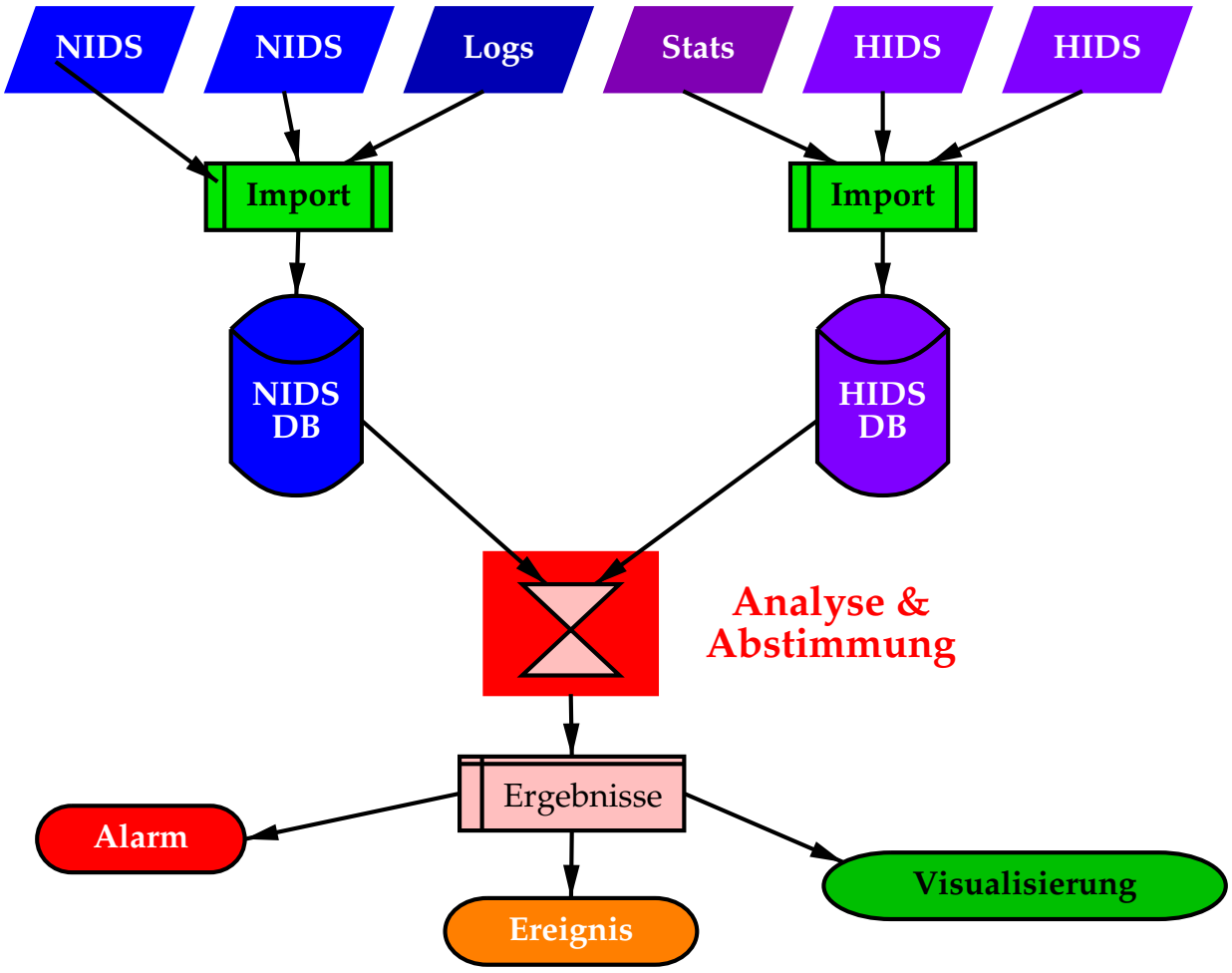

Abbildung 6: Ausgedehnter Einsatz von IDS Maßnahmen erfordert ein gewisses Maß an zentralisiertem Sammeln der Daten. Dies gilt ebenso für die Verteilung der Konfigurationen an die verschiedenen Sensoren. Die Sammlung der Daten führt in eine statistische Auswertung. Das Befüllen kann in größeren periodischen Abständen (z.B. Tagen) erfolgen, um Performance-Engpässe auszugleichen.

3 Effizientes Auswerten der Log Files

```
UNDERSTANDING, n.
   A cerebral secretion that enables one having it to know a
   house from a horse by the roof on the house. Its nature and
   laws have been exhaustively expounded by Locke, who rode a
   house, and Kant, who lived in a horse.

 --- "The Devil's Dictionary", Ambrose Bierce
```

3.1 Rechtsfragen beim Überwachen von Netzwerken

Darf man überhaupt effizient Auswerten?

- **Begriff „Sniffen“**
  umfaßt das Aufzeichnen und Auswerten von Netzwerkverkehr
- **Sniffen in fremden Netzwerken ist definitiv rechtlich problematisch**
- **firmeninterne Netzwerke sind rechtlich besser „handhabbar“**
  $\longrightarrow$ *Betriebsvereinbarungen*
  $\longrightarrow$ *interne Security Policy*
  sind jedoch **keine** Freibriefe auf wilden Informationsraubbau
- *Datenschutz spielt wesentliche Rolle*
  - Sammlung personenbezogener Daten kritisch
  - Auswertungen können für Personenprofile miß-/gebraucht werden
  - Aufbewahrungsgrenzen für Daten

3.2 Grundsätzliche Quellen für rechtliche Aspekte

- **EU Datenschutzrichtlinie**[23]
- **DSG 2000**[24]
  - Recht auf Geheimhaltung (§§ 1ff DSG 2000, Verfassungsbestimmung)
  - Informationsrecht (§ 24 DSG 2000)
  - Recht auf Auskunft (§ 26 DSG 2000)
  - Recht auf Berichtigung und Löschung (§ 27 DSG 2000)
  - Recht auf Widerspruch (§ 28 DSG 2000)
  - Recht auf Widerruf (§§ 8, 9 DSG 2000)
  - Recht auf Information über logischen Ablauf bei automatisierten Einzelentscheidungen (§ 49 Abs. 3 DSG 2000)
- **Strafrecht**
  - §126a Datenbeschädigung
  - §148a Betrügerischer Datenverarbeitungsmißbrauch
  - §202a deutsches StGB Ausspähung von Daten
- **Zivilrecht**
  - Vertragsrecht (insbesondere ISP, Arbeitgeber, etc.)
  - Schadenersatz

ARGE Daten[25] hat Praxisbeispiele in ihrem Newsletter.

[23] http://www.ad.or.at/office/recht/eu.htm
[24] http://www.ad.or.at/office/recht/dsg2000.htm
[25] http://www.ad.or.at/

3.3 Auswerten der IDS Daten

- **IDS generieren sehr viele Daten**
- **datenbankgestützte Auswertung für ernsthafte Analyse erforderlich**
  - Einsatz von Data Mining Methoden
  - Erkennen von Mustern und Anomalien
  - Suche nach Korrelationen
- **Korrelation der IDS Daten mit anderen Logs**
  - Systemlogs (z.B. *syslogd*)
  - Maillogs
  - Web- und FTP-Server Logs
  - Logindaten (z.B. SSH, RADIUS, VPN Connects)
  - Disk I/O, Netzwerklast, CPU Last

  Korrelation erfordert Zeitsynchronisation aller Sonden!
- **Human Intelligence ist gleichermaßen erforderlich**
  $\longrightarrow$ *auch visualisierte Daten müssen interpretiert werden*

3.4 Automationsmöglichkeiten - Übersicht

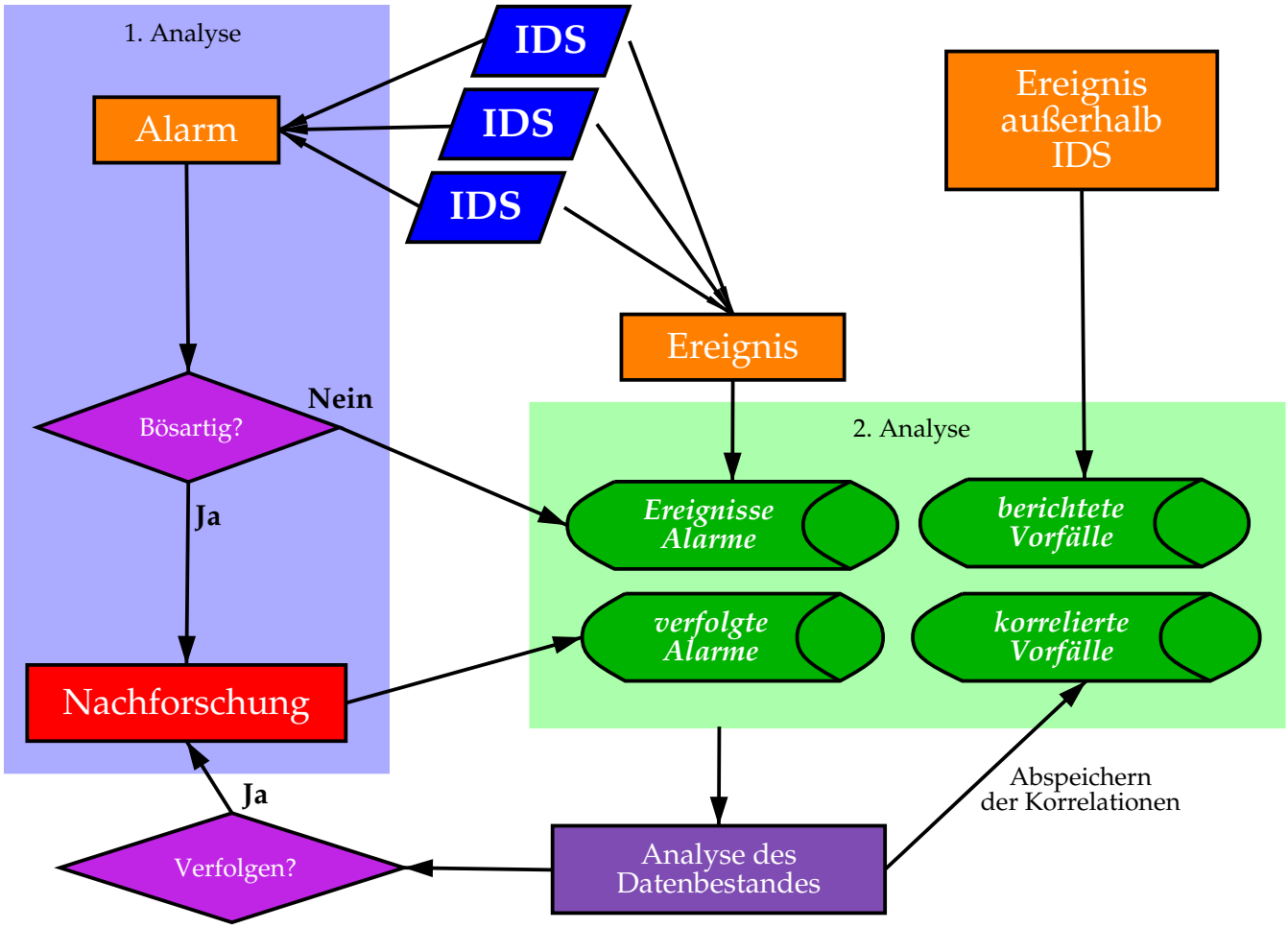

Abbildung 7: Die Abbildung zeigt die mehrstufige Auswertung von IDS Daten. In der ersten Stufe wird unmittelbar auf einen Alarm reagiert. Die Daten werden in Datenbanken gesammelt und durch weitere Prozesse analysiert. Dadurch können sich weitere Alarme ergeben, die behandelt werden müssen. (`http://www.securityfocus.com/infocus/1201`)

3.5 Der Einsatz von Data Mining Verfahren

- **Data Mining ist kein Real Time Verfahren**
  $\longrightarrow$ *Daten stehen nicht sofort im richtigen Format bereit*
  $\longrightarrow$ *Verzögerung durch Aufbereitung der Daten*
- **Etablierung einer Baseline**
  - Grundzustand der überwachten Systeme muß bekannt sein
  - Simulation von Angriffen bzw. Ausnahmesituationen muß möglich sein
  - ausreichend Zeit für Kalibrierung muß vorhanden sein
- **Organisation des Datenflusses muß geplant sein**
  $\longrightarrow$ *Weg der Daten & Delta Load muß feststehen*
- **aufbereitete Daten können zur Visualisierung dienen**
  - Online Analytical Processing (OLAP)
  - direkte Aufbereitung der Datenbank in Graphen

  sehr nützliches Hilfsmittel für Systemadministration
- **Infrastruktur ermöglicht leichteres Erstellen von Statusberichten**

3.6 Beispiel für kontinuierliches Monitoring

- **Überwachung mehrerer Maschinen durch Samhain Sensoren**
  $\longrightarrow$ *Meldung aller Ereignisse an Yule Prozeß am Logserver*
- **DMZ und LAN wird durch Snort Server beobachtet**
  $\longrightarrow$ *je eine Netzwerkkarte pro Snort Sensor Prozeß*
  $\longrightarrow$ *lokale Packet Capture Logs*
  $\longrightarrow$ *Ereignisse gehen an Logserver*
- **periodisches Wandeln von System- und Maillogs in SQL**
  $\longrightarrow$ *Perl Skripte konvertieren syslodg & Sendmail Logs*
- **zentrales Logging an Logserver**
  $\longrightarrow$ *PostgreSQL Datenbank*
  $\longrightarrow$ *direkte Generierung von Email Alerts*
- **Aufbereiten der Daten zur Analyse**
  $\longrightarrow$ *Import der Daten in IBM DB2 zur OLAP Analyse*
  $\longrightarrow$ *weitere statistische Analyse auf PostgreSQL und DB2*

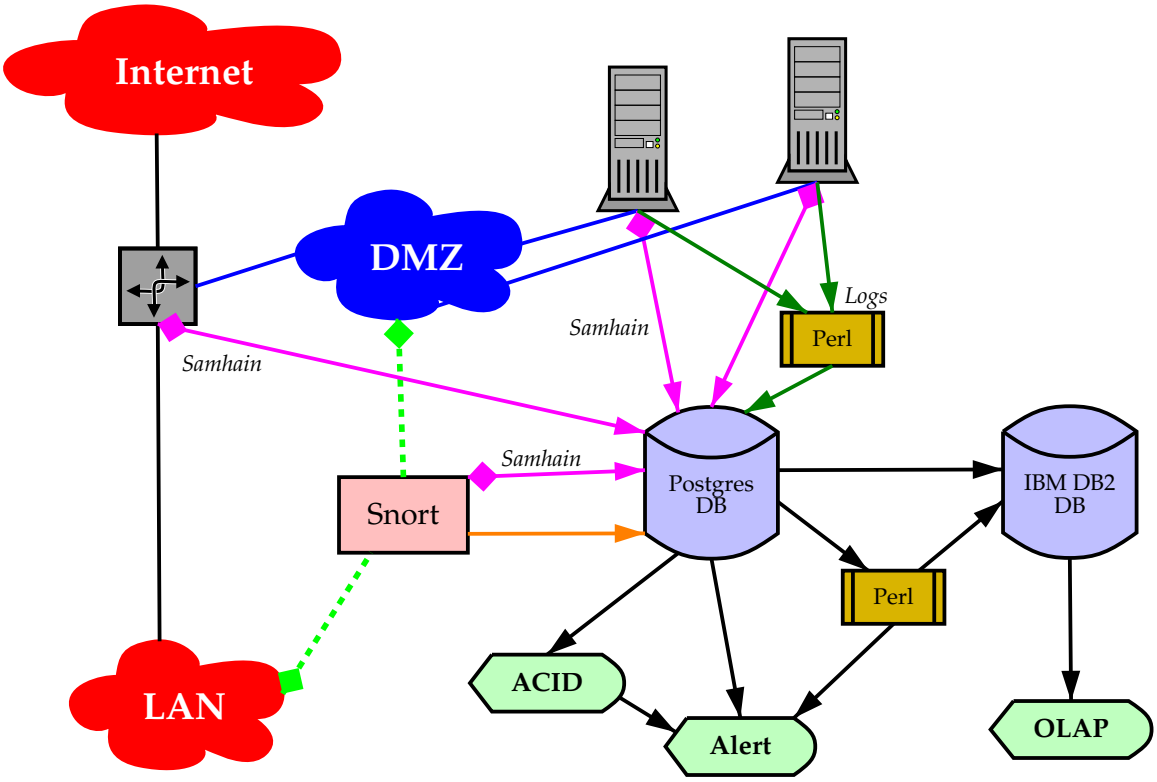

Abbildung 8: Die Abbildung zeigt die kontinuierliche Auswertung von IDS Daten. Es spielen hier NIDS und HIDS Komponenten zusammen, die verschiedene Applikationen, Server und Netzwerke überwachen. Zentraler Sammelpunkt für die Rohdaten ist die PostgresSQL Datenbank. Dort findet die erste Stufe der Auswertung statt, wo auf bestimmte Ereignisse sofort reagiert werden kann. Weitere Auswertungen geschehen auf der DB2 Datenbank.

3.7 Möglichkeiten zur selbstständigen Anomaliebeschreibung

- **Definition eines Feature Sets für Gruppen von Systemen**
  Sammlung von Daten
  - im Normalzustand
  - mit simulierten Attacken

  Beispiele für kategorisierte Paketdumps in den Daten zum Knowledge Discovery in Databases (KDD) Cup 1999[26]

- **Verarbeitung der Feature Sets durch Data Mining**
  - IBM Intelligent Miner[27]
    kann direkt mit IBM DB2 verwendet werden
  - Torch[28]
    freie C++ Bibliothek mit Implementation verschiedener Machine Learning Methoden
  - WEKA[29]
    freie Data Mining Software geschrieben in Java

- **Mining Analyse kann Anomalien und unbekannte Attacken aufspüren**

[26] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[27] http://www-3.ibm.com/software/data/iminer/
[28] http://www.torch.ch/
[29] http://www.cs.waikato.ac.nz/~ ml/weka/

A hping2 Beispiele

A.1 hping2 als Standard ICMP Ping (mit Dump des Replies)

```
hping2 --icmp --count 4 --dump --verbose cvs.einhost.xy

HPING cvs.einhost.xy (eth1 z.y.x.v): icmp mode set, 28 headers + 0 data bytes
46 bytes from z.y.x.v: icmp_seq=0 ttl=248 id=12166 rtt=39.2 ms
                0060 9711 d902 0030 7b6c 2860 0800 4500
                001c 2f86 0000 f801 c7d3 c3e6 2ac4 d4ba
                0822 0000 24e4 db1b 0000 0000 0000 0000
                0000 0000 0000 0000 0000 0000

46 bytes from z.y.x.v: icmp_seq=1 ttl=248 id=12167 rtt=13.8 ms
                0060 9711 d902 0030 7b6c 2860 0800 4500
                001c 2f87 0000 f801 c7d2 c3e6 2ac4 d4ba
                0822 0000 23e4 db1b 0100 0000 0000 0000
                0000 0000 0000 0000 0000 0000

46 bytes from z.y.x.v: icmp_seq=2 ttl=248 id=12168 rtt=18.5 ms
                0060 9711 d902 0030 7b6c 2860 0800 4500
                001c 2f88 0000 f801 c7d1 c3e6 2ac4 d4ba
                0822 0000 22e4 db1b 0200 0000 0000 0000
                0000 0000 0000 0000 0000 0000

46 bytes from z.y.x.v: icmp_seq=3 ttl=248 id=12180 rtt=25.5 ms
                0060 9711 d902 0030 7b6c 2860 0800 4500
                001c 2f94 0000 f801 c7c5 c3e6 2ac4 d4ba
                0822 0000 21e4 db1b 0300 0000 0000 0000
                0000 0000 0000 0000 0000 0000


--- cvs.einhost.xy hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 13.8/24.2/39.2 ms
```

A.2 TCP Ping

```
hping2 --count 8 mail.einhost.xy

HPING mail.einhost.xy (eth1 a.b.c.d): NO FLAGS are set, 40 headers + 0 data bytes
46 bytes from a.b.c.d: flags=RA seq=0 ttl=248 id=57313 win=0 rtt=9.8 ms
46 bytes from a.b.c.d: flags=RA seq=1 ttl=248 id=57314 win=0 rtt=12.8 ms
46 bytes from a.b.c.d: flags=RA seq=2 ttl=248 id=57315 win=0 rtt=8.6 ms
46 bytes from a.b.c.d: flags=RA seq=3 ttl=248 id=57316 win=0 rtt=16.7 ms
46 bytes from a.b.c.d: flags=RA seq=4 ttl=248 id=57317 win=0 rtt=13.1 ms
46 bytes from a.b.c.d: flags=RA seq=5 ttl=248 id=57318 win=0 rtt=17.2 ms
46 bytes from a.b.c.d: flags=RA seq=6 ttl=248 id=57319 win=0 rtt=10.5 ms
46 bytes from a.b.c.d: flags=RA seq=7 ttl=248 id=57320 win=0 rtt=8.4 ms

--- mail.einhost.xy hping statistic ---
8 packets tramitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 8.4/12.1/17.2 ms
```

- Der Scan benutzt TCP Null-Flag Packets mit Ziel-Port 0.
- TCP Ping ist guter Ersatz falls ICMP Echo Request blockiert wird (viele Paketfilter blockieren ICMP Echo Request und Reply Pakete).

A.3 TCP Ping auf einen Host mit `iplog`

```
hping2 --count 8 cvs.einhost.xy

HPING cvs.einhost.xy (eth1 z.y.x.v): NO FLAGS are set, 40 headers + 0 data bytes
46 bytes from z.y.x.v: flags=RA seq=0 ttl=248 id=31148 win=0 rtt=24.4 ms
(DUP!) 46 bytes from z.y.x.v: flags=RSY seq=0 ttl=248 id=65280 win=0 rtt=27.7 ms
(DUP!) 46 bytes from z.y.x.v: flags=RSY seq=0 ttl=248 id=65280 win=0 rtt=59.3 ms
(DUP!) 46 bytes from z.y.x.v: flags=RSY seq=0 ttl=248 id=65280 win=0 rtt=79.0 ms
(DUP!) 46 bytes from z.y.x.v: flags=RSY seq=0 ttl=248 id=65280 win=0 rtt=92.8 ms
(DUP!) 46 bytes from z.y.x.v: flags=RSY seq=0 ttl=248 id=65280 win=0 rtt=134.0 ms
(DUP!) 46 bytes from z.y.x.v: flags=RSFY seq=0 ttl=248 id=65280 win=256 rtt=186.7 ms
(DUP!) 46 bytes from z.y.x.v: flags=RSFY seq=0 ttl=248 id=65280 win=256 rtt=223.7 ms

--- cvs.einhost.xy hping statistic ---
1 packets tramitted, 8 packets received, -700% packet loss
round-trip min/avg/max = 24.4/103.5/223.7 ms
```

Am Host läuft `iplog --ignore -V -y --fool-nmap=true -g iplog -u iplog` Detektiert wurde

```
Jun 17 23:26:06 cvs iplog[528]: TCP: port 0 connection
        attempt from hping2.einhost.xy:2805
```

Durch die Option `--fool-nmap` sendet `iplog` Decoy-Packets als Reply

A.4 Senden von TCP Null-Flag Packets an einen Webserver

```
hping2 --count 4 --destport 80 wayreth.luchs.at

HPING wayreth.luchs.at (eth1 195.230.42.195): NO FLAGS are set, 40 headers + 0 data bytes

--- webserver.hier.zz hping statistic ---
4 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Host antwortet nicht

Senden von TCP SYN-Packets

```
hping2 --count 4 --destport 80 -S wayreth.luchs.at

HPING wayreth.luchs.at (eth1 195.230.42.195): S set, 40 headers + 0 data bytes
46 bytes from 195.230.42.195: flags=SA seq=0 ttl=57 id=41687 win=31624 rtt=25.9 ms
46 bytes from 195.230.42.195: flags=SA seq=1 ttl=57 id=41688 win=31624 rtt=29.2 ms
46 bytes from 195.230.42.195: flags=SA seq=2 ttl=57 id=41689 win=31624 rtt=21.1 ms
46 bytes from 195.230.42.195: flags=SA seq=3 ttl=57 id=41704 win=31624 rtt=12.9 ms

--- wayreth.luchs.at hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 12.9/22.3/29.2 ms
```

Host meldet korrekt SYN+ACK

Senden von TCP ACK-Packets

```
hping2 --count 4 --destport 80 -A wayreth.luchs.at

HPING wayreth.luchs.at (eth1 195.230.42.195): A set, 40 headers + 0 data bytes
46 bytes from 195.230.42.195: flags=R seq=0 ttl=248 id=39818 win=0 rtt=21.1 ms
46 bytes from 195.230.42.195: flags=R seq=1 ttl=248 id=39819 win=0 rtt=39.7 ms
46 bytes from 195.230.42.195: flags=R seq=2 ttl=248 id=39831 win=0 rtt=18.9 ms
46 bytes from 195.230.42.195: flags=R seq=3 ttl=248 id=39832 win=0 rtt=17.4 ms

--- wayreth.luchs.at hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 17.4/24.3/39.7 ms
```

Host quittiert mit einem RST

A.5 TCP SYN auf firewalled Port (Policy DENY / DROP)

```
hping2 --count 4 --destport 3306 -S -r wayreth.luchs.at

eth1 default routing interface selected (according to /proc)
HPING wayreth.luchs.at (eth1 195.230.42.195): S set, 40 headers + 0 data byt

--- wayreth.luchs.at hping statistic ---
4 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Spuren im Logfile

```
Jun 08 20:54:56 wayreth kernel: Packet log:
        input DENY eth1 PROTO=6 192.168.0.72:1056
        195.230.42.195:3306 L=40 S=0x00 I=41685 F=0x0000 T=57 SYN (#6)
Jun 08 20:54:56 wayreth iplog[558]: TCP: mysql
        connection attempt from roaminglynx.luchs.at:1056
Jun 08 20:54:57 wayreth kernel: Packet log:
        input DENY eth1 PROTO=6 192.168.0.72:1057
        195.230.42.195:3306 L=40 S=0x00 I=51043 F=0x0000 T=57 SYN (#6)
Jun 08 20:54:57 wayreth iplog[558]: TCP: mysql
        connection attempt from roaminglynx.luchs.at:1057
Jun 08 20:54:58 wayreth kernel: Packet log:
        input DENY eth1 PROTO=6 192.168.0.72:1058
        195.230.42.195:3306 L=40 S=0x00 I=26586 F=0x0000 T=57 SYN (#6)
Jun 08 20:54:58 wayreth iplog[558]: TCP: mysql
        connection attempt from roaminglynx.luchs.at:1058
Jun 08 20:54:59 wayreth kernel: Packet log:
        input DENY eth1 PROTO=6 192.168.0.72:1059
        195.230.42.195:3306 L=40 S=0x00 I=51704 F=0x0000 T=57 SYN (#6)
Jun 08 20:54:59 wayreth iplog[558]: TCP: mysql
        connection attempt from roaminglynx.luchs.at:1059
```

A.6 TCP SYN auf firewalled Port (Policy REJECT)

```
hping2 --count 5 --destport 65535 -S -r wayreth.luchs.at
eth1 default routing interface selected (according to /proc)
HPING wayreth.luchs.at (eth1 195.230.42.195): S set, 40 headers + 0 data byt
ICMP Port Unreachable from 195.230.42.195  (wayreth.luchs.at)
ICMP Port Unreachable from 195.230.42.195  (wayreth.luchs.at)
ICMP Port Unreachable from 195.230.42.195  (wayreth.luchs.at)
ICMP Port Unreachable from 195.230.42.195  (wayreth.luchs.at)
ICMP Port Unreachable from 195.230.42.195  (wayreth.luchs.at)

--- wayreth.luchs.at hping statistic ---
5 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Spuren im Logfile

```
Jun 08 21:01:41 wayreth iplog[558]: TCP:
        port 65535 connection attempt from roaminglynx.luchs.at:2930
Jun 08 21:01:42 wayreth kernel: Packet log:
        input REJECT eth1 PROTO=6 192.168.0.72:2931
        195.230.42.195:65535 L=40 S=0x00 I=23591 F=0x0000 T=57 SYN (#1)
Jun 08 21:01:42 wayreth iplog[558]: TCP:
        port 65535 connection attempt from roaminglynx.luchs.at:2931
Jun 08 21:01:43 wayreth kernel: Packet log:
        input REJECT eth1 PROTO=6 192.168.0.72:2932
        195.230.42.195:65535 L=40 S=0x00 I=17784 F=0x0000 T=57 SYN (#1)
Jun 08 21:01:43 wayreth iplog[558]: TCP:
        port 65535 connection attempt from roaminglynx.luchs.at:2932
Jun 08 21:01:44 wayreth kernel: Packet log:
        input REJECT eth1 PROTO=6 192.168.0.72:2933
        195.230.42.195:65535 L=40 S=0x00 I=59363 F=0x0000 T=57 SYN (#1)
Jun 08 21:01:44 wayreth iplog[558]: TCP:
        port 65535 connection attempt from roaminglynx.luchs.at:2933
Jun 08 21:01:45 wayreth kernel: Packet log:
        input REJECT eth1 PROTO=6 192.168.0.72:2934
        195.230.42.195:65535 L=40 S=0x00 I=34878 F=0x0000 T=57 SYN (#1)
Jun 08 21:01:45 wayreth iplog[558]: TCP:
        port 65535 connection attempt from roaminglynx.luchs.at:2934
```

B nmap Beispiele

B.1 Standard TCP connect() Scan mit Version-ID-Patch

```
Nmap (V. nmap) scan initiated 2.53 as:
nmap -sT -sV -sR -r -O -I -oN /root/output_sT ein.host.xy
Interesting ports on some.host.xy (a.b.c.d):
(The 1512 ports scanned but not shown below are in state: closed)
Port       State       Service (RPC)       Owner       Protocol     Version
21/tcp     open        ftp                 0           FTP          ProFTPD 1.2.0pre1
22/tcp     open        ssh                 0           SSH          1.5-1.2.26
25/tcp     open        smtp                0           SMTP
53/tcp     open        domain              0
110/tcp    open        pop-3               0           POP3         QPOP 2.53
111/tcp    open        sunrpc (rpcbind V2) 1
113/tcp    open        auth                65535
515/tcp    open        printer             0
724/tcp    open        (bwnfsd V1)         0
767/tcp    open        phonebook (mountd V1-2)0
2049/tcp   open        nfs (nfs V2)        0

TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
Remote operating system guess: Linux 2.0.35-38

# Nmap run completed at Sun May 28 19:00:57 2000 --
# 1 IP address (1 host up) scanned in 116 seconds
```

- **Wichtig:** Die Service-Bezeichnung stammt aus dem `/etc/services` File! Nur nmap mit Versionspatch versucht herauszufinden, was auf dem entsprechenden Port läuft
- TCP connect() Scans sind sehr auffällig
  $\longrightarrow$ einfacher Test für Network Intrusion Detection Systeme
- Portsentry und Iplog erzeugen ca. 1 MB Logfileeinträge

B.2 Standard Scan

```
[root@anubis ~]# nmap -sTUR -I -O -oN /net/tmp/10.0.0.12_sTUR 10.0.0.12

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on tempest.elements.lan (10.0.0.12):
(The 3064 ports scanned but not shown below are in state: closed)
Port       State       Service (RPC)           Owner
22/tcp     open        ssh
25/tcp     open        smtp
80/tcp     open        http
3306/tcp   open        mysql
5432/tcp   open        postgres
Remote OS guesses: Linux Kernel 2.4.0 - 2.5.20, Linux 2.4.19-pre4 on
Alpha

Nmap run completed -- 1 IP address (1 host up) scanned in 1526 seconds
```

B.3 Standard Scan mit XML Protokoll

```
[root@kiri-jolith root]# nmap -sTUR -oX /tmp/10.0.0.141.xml -I -O 10.0.0.141

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on dummy0141.elements.lan (10.0.0.141):
(The 3058 ports scanned but not shown below are in state: closed)
Port       State       Service (RPC)           Owner
135/tcp    open        loc-srv
135/udp    open        loc-srv
137/udp    open        netbios-ns
138/udp    open        netbios-dgm
139/tcp    open        netbios-ssn
445/tcp    open        microsoft-ds
445/udp    open        microsoft-ds
500/udp    open        isakmp
1025/tcp   open        NFS-or-IIS
1433/tcp   open        ms-sql-s
1434/udp   open        ms-sql-m
Remote operating system guess: Windows Millennium Edition (Me), Win 2000,
                               or WinXP

Nmap run completed -- 1 IP address (1 host up) scanned in 17 seconds
```

C Samhain / Yule

C.1 Verteilte HIDS mit zentralem Logserver

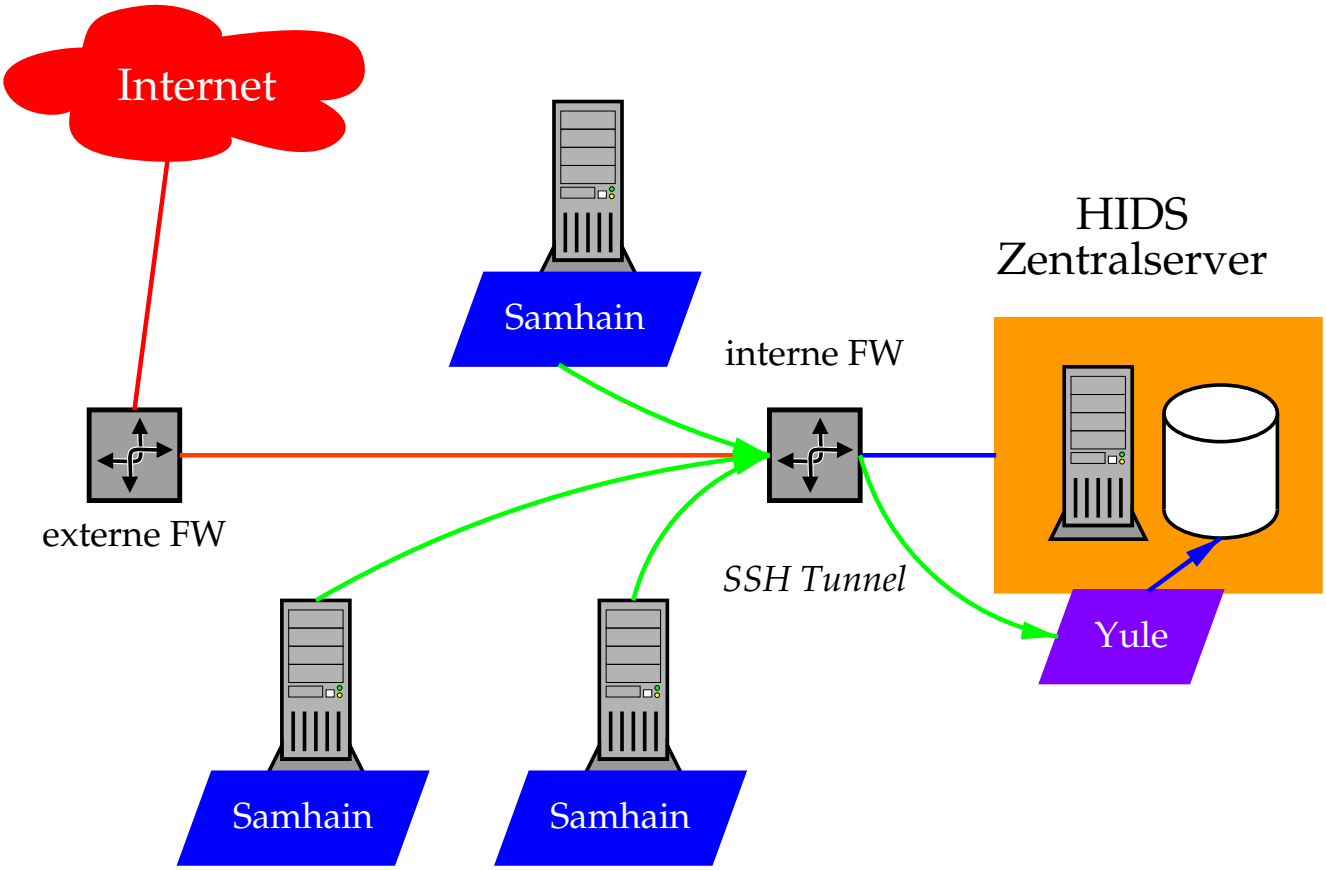

Abbildung 9: Schematischer Einsatz von Samhain Sensoren in einem Netzwerk. Samhain kann mit dem Yule Prozeß am Logserver sowohl über SSH Tunnel als auch über eine eigene Verschlüsselung (mittels AES Algorithm Rijndael) kommunizieren.

D ACID - Analysis Console for Intrusion Databases

D.1 Manuelle Auswertung von Snort Events

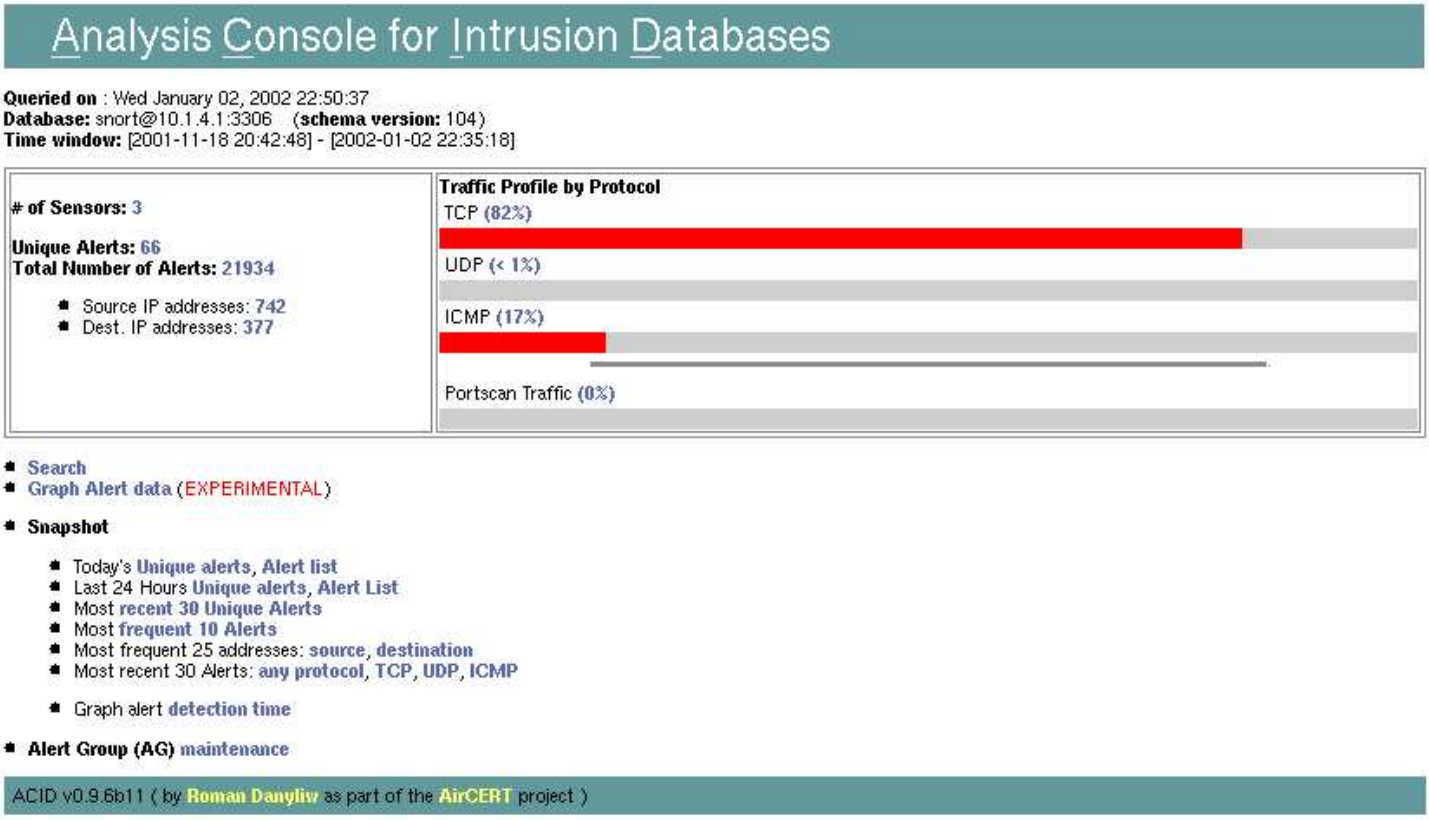

Abbildung 10: Die Abbildung zeigt die Analysis Console for Intrusion Databases (ACID). ACID besteht aus einer PHP4 Applikation, die Snort Ereignisse aus einer SQL Datenbank darstellen und bearbeiten kann.
(`http://www.cert.org/kb/aircert/`)

D.2 Manuelle Auswertung - Detailansicht

ACID **Alert Listing** Home | Search | AG Maintenance

Queried DB on : Wed January 02, 2002 22:52:05

Meta Criteria	time >= [01 / 01 / 2002] [22 : * : *]
IP Criteria	any
Layer 4 Criteria	none
Payload Criteria	any

Displaying alerts 1-13 of 13 total
(Aggregating 21934 events)

Queried DB on : Wed January 02, 2002 22:52:05

	‹ Signature ›	‹ Total # ›	# Sensors	Src. Addr	Dst. Addr	‹ First ›	Previous	‹ Last ›
☐	ICMP Destination Unreachable (Communication Administratively Prohibited)	3 (0%)	1	2	1	2002-01-01 23:18:40	2002-01-02 16:37:59	2002-01-02 16:38:03
☐	WEB-IIS CodeRed v2 root.exe access	16 (0%)	2	13	3	2002-01-01 22:41:21	2002-01-02 21:35:49	2002-01-02 22:22:10
☐	WEB-MISC 403 Forbidden	54 (0%)	1	2	13	2002-01-01 23:16:29	2002-01-02 22:22:13	2002-01-02 22:22:30
☐	WEB-IIS cmd.exe access	161 (1%)	2	11	3	2002-01-01 23:46:02	2002-01-02 22:22:47	2002-01-02 22:22:50
☐	WEB-FRONTPAGE /_vti_bin/ access	12 (0%)	2	10	3	2002-01-01 23:46:08	2002-01-02 21:35:54	2002-01-02 22:22:26
☐	[arachNIDS] MISC Large ICMP Packet	41 (0%)	2	9	4	2002-01-01 23:24:46	2002-01-02 19:48:09	2002-01-02 20:34:19
☐	[arachNIDS] DNS zone transfer	1 (0%)	1	1	1	2002-01-02 18:11:20	2002-01-02 18:11:20	2002-01-02 18:11:20
☐	INFO - Possible Squid Scan	30 (0%)	1	1	1	2002-01-02 12:08:51	2002-01-02 17:44:02	2002-01-02 17:44:50

Abbildung 11: Ereignisse der letzten 24 Stunden im Überblick.
(`http://www.cert.org/kb/aircert/`)