Datensicherheit IV

Firewalls

Version 0.5

Sommersemester 2005

Technikum Wien

René Pfeiffer

pfeiffer at luchs.at

Systemadministrator GNU/Linux Manages!

Inhaltsverzeichnis

1 Bedrohungen **9**
1.1 Bedrohungen für Netzwerke 9
1.2 Bedrohungen für Server 10
1.3 Bedrohungen für aktive Netzwerkkomponenten 10

2 Sicherheitsmaßnahmen **11**
2.1 Einsatz von Proxy-Servern 11
2.2 Firewall-Architekturen 13
2.2.1 Screened Host Architektur 14
2.2.2 Screened Subnet Architektur 15
2.2.3 Mehrere interne Netzwerke 16
2.3 Einsatz und Aufbau von Bastion Hosts 17
2.3.1 Sichern der Maschine 19
2.3.2 Deaktivieren aller unnötigen Services 20
2.3.3 Installieren oder Ändern der benötigten Services 21
2.3.4 Rekonfigurieren der Maschine für Production Environment 22
2.4 Checklisten für ausgewählte Applikationen 23
2.4.1 Apache 23
2.4.2 BIND 23
2.4.3 FTP Server 24
2.4.4 Mail Transport Agents (MTAs) 24
2.4.5 PHP 24

3 Angriffe auf Netzwerke und Server **25**
3.1 Tools zur Aufklärung 26
3.1.1 Ethereal - Packet Analyzer 26
3.1.2 hping2 - Paketgenerator 27
3.1.3 isic - Zufallspaketgenerator 27
3.1.4 SING - ICMP Packetgenerator 28
3.1.5 nmap - Network Mapper 28
3.1.6 Sonstige Tools 29

4 Intrusion Detection Systeme **31**
4.1 Network Intrusion Detection mit Snort 32
4.2 Snort Filterregeln 33
4.3 Snort Flexible Response 33
4.4 Samhain / Yule - Client/Server HIDS 35

4.5 Logging - Aufspüren von Unregelmäßigkeiten . . . 36
4.6 Anomalien - Was sind Unregelmäßigkeiten? . . . 37
4.7 Kalibrierung und Planung . . . 38
4.8 Gezielte Fragen an Intrusion Detection Systeme . . . 39

5 IDS in komplexen Netzwerken **41**
5.1 Architektonische Überlegungen . . . 42
5.2 NIDS Zonen . . . 43
5.3 Methoden zur Netzwerküberwachung . . . 45
5.4 SPAN Port Implementation . . . 46
5.5 Network TAPs . . . 47
5.6 Attacken gegen NIDS Implementationen . . . 48
5.7 Überlegungen zur Performance . . . 49
5.8 Mehrstufige Architektur für Hochleistungs IDS . . . 50
5.9 Zentralisieren der Konfiguration . . . 51
5.10 Konsolidierung . . . 52
5.11 Performance-schonende Lösungen . . . 53
5.12 Hybrid-IDS mit Datenbankauswertung . . . 54
5.13 Effizientes Auswerten der Log Files . . . 55
5.13.1 Rechtsfragen beim Überwachen von Netzwerken . . . 56
5.13.2 Grundsätzliche Quellen für rechtliche Aspekte . . . 57
5.13.3 Auswerten der IDS Daten . . . 58
5.13.4 Automationsmöglichkeiten - Übersicht . . . 59
5.13.5 Der Einsatz von Data Mining Verfahren . . . 60
5.13.6 Beispiel für kontinuierliches Monitoring . . . 61
5.13.7 Möglichkeiten zur selbstständigen Anomaliebeschreibung . . . 63

Abbildungsverzeichnis

2.1 Einsatz eines Proxy-Servers 12
2.2 Einfache dual-homed Firewall 13
2.3 Screened Host Architektur mit Bastion Host 14
2.4 Screened Subnet Architektur mit DMZ und Bastion Host 15
2.5 Firewall mit mehreren internen Netzwerken 16

5.1 Schematischer Einsatz von Snort Sensoren 44
5.2 Schematische Ansicht eines Network TAPs 47
5.3 Zustandsgesteuerte Intrusion Detection in Hochleistungsnetzen 50
5.4 Zentralisieren der Konfiguration bei ausgedehnten IDS Netzen 51
5.5 Fluß von IDS und Logdaten durch eine Auswertung 54
5.6 Mehrstufige Auswertung von IDS Daten 59
5.7 Kontinuierliche Auswertung von IDS Daten 62

Tabellenverzeichnis

Kapitel 1

Bedrohungen

```
What to do in case of an alien attack:

    1) Hide beneath the seat of your plane and look away.
    2) Avoid eye contact.
    3) If there are no eyes, avoid all contact.

-- The Firesign Theatre, _Everything you know is Wrong_
```

Die Bedrohungen für Netzwerke und Server sind sehr vielfältig. Alleine durch die enorme Anzahl der Software, die im täglichen Einsatz ist, kann sich jeder einzelne Fehler im Code auch auf andere Bereiche auswirken. Neben Betriebssystemen und deren Komponenten sind individuelle und angepaßte Anwendungen in einem weit größeren Maß betroffen. Die folgende Aufstellung gibt einen kleinen Überblick.

1.1 Bedrohungen für Netzwerke

- **(Distributed) Denial of Service (DDoS)**
  stellt das vorsätzliche Deaktivieren eines Dienstes dar. Dies geschieht meist durch Fluten der Netzwerkanbindung mit Paketen.

- **DNS Cache Vergiftung**
  Bestimmte DNS Caches lassen sich gezielt mit Falschinformationen füttern und behalten diese für weitere Auskünfte. Damit kann man Authentifizierungen, die alleine aufgrund DNS Informationen passieren, umgehen. Weiterhin lassen sich damit Webseiten und Emails entführen.

- **Fragmentation Attacken**

- **FTP- und Proxy Bounce Attacken**

- **ICMP Attacken**
  ICMP Pakete können sowohl zur Aufklärung als auch zum Angreifen[1] verwendet werden. Der Angriff richtet sich dabei gegen Gateways, die der Angreifer mit ICMP Route Redirect und anderen Mitteln in seinen Besitz zu bringen versucht.

[1] http://rr.sans.org/threats/ICMP_attacks.php

- **Packet Spoofing**
  bezeichnet das Fälschen von Sessions bzw. Paketen. Betroffen sind davon oft IP, ARP, HTTP und DNS. Siehe diesen SANS Artikel[2] für weitere Informationen.

- **Paßwortattacken (Sniffing)**

- **Sondierungen & Proben**
  Die wenigsten wirklichen Attacken laufen ohne Vorbereitung ab. Oft wird im voraus das Ziel ausgekundschaftet, um möglichst viele Informationen zu erhalten.

1.2 Bedrohungen für Server

- **Buffer Overflow mit Code Injektion**
  Ein Buffer Overflow stellt das Einschleusen von Assembler Code in eine laufende Software dar. Anfällig für solche Attacken ist jeglicher Code, der keine Überprüfung von Pufferbereichen vornimmt[3] (z.B. C Funktionen wie `strcat()`, `strcpy()`, `sprintf()`, `vsprintf()`, `bcopy()`, `gets()` oder `scanf()`).

- **Denial of Service (DoS)**
  Durch Senden von ungültigen oder sehr vielen Daten wird ein bestimmter Dienst zum Absturz gebracht und steht dann nicht mehr zur Verfügung.

- **Mißkonfigurationen**
  Darunter fällt ein sehr großer Bereich, wie beispielsweise
  - Default Paßworte
  - offene Fileshares
  - Trust Relationships zwischen Systemen
  - versehentlich aktivierte Services

- **SQL Injektion**
  Durch Absetzen eines HTTP GET bzw. POST Requests mit abgeänderten Parametern bringt man ein CGI Skript dazu bestimmte SQL Befehle auszuführen, wodurch man Zugriff auf die hinter der Web Applikation liegende Datenbank bekommt. Es gibt zu diesen Attacken genauere Informationen bei Silksoft[4] oder `securiteam.com`[5].

1.3 Bedrohungen für aktive Netzwerkkomponenten

- **Managementkonsolen**
  Zugriff auf Management-Interfaces kann zu Problemen führen, wenn diese nicht richtig abgesichert sind.

[2] `http://rr.sans.org/threats/spoofing.php`
[3] http://www.networkmagazine.com/article/NMG20000511S0015
[4] http://www.silksoft.co.za/data/sqlinjectionattack.htm
[5] http://www.securiteam.com/securityreviews/5UP010A6AA.html

Kapitel 2

Sicherheitsmaßnahmen

```
"Probably the only place where a man can feel really
 secure is in a maximum security prison, except for
 the imminent threat of release."

   --- Germaine Greer
```

2.1 Einsatz von Proxy-Servern

Vorteile

- Zugriff auf Internetdienste erfolgt direkt
- Proxy-Server können effizient Logfiles über Zugriffe führen
- ACL auf Benutzerebene möglich
- Trennung von Netzwerken bei ausschließlichem Einsatz von Proxies
- Möglichkeit des Content Filtering
  $\longrightarrow$ Anti-Viren Systeme (SMTP Proxy)
  $\longrightarrow$ Downloadbeschränkung (HTTP Proxy)

Nachteile

- Entwicklung von Proxy-Software zu langsam für neue Protokolle
- Proxy-Server wird pro Service benötigt (z.B. SMTP, HTTP, Telnet)
- Einsatz erfordert oft spezielle Konfigurationen an Clients
- nicht alle Protokolle lassen einen Proxy zu
- Proxy-Server schützen nicht vor gefährlichem Inhalt (Data Driven Attack)

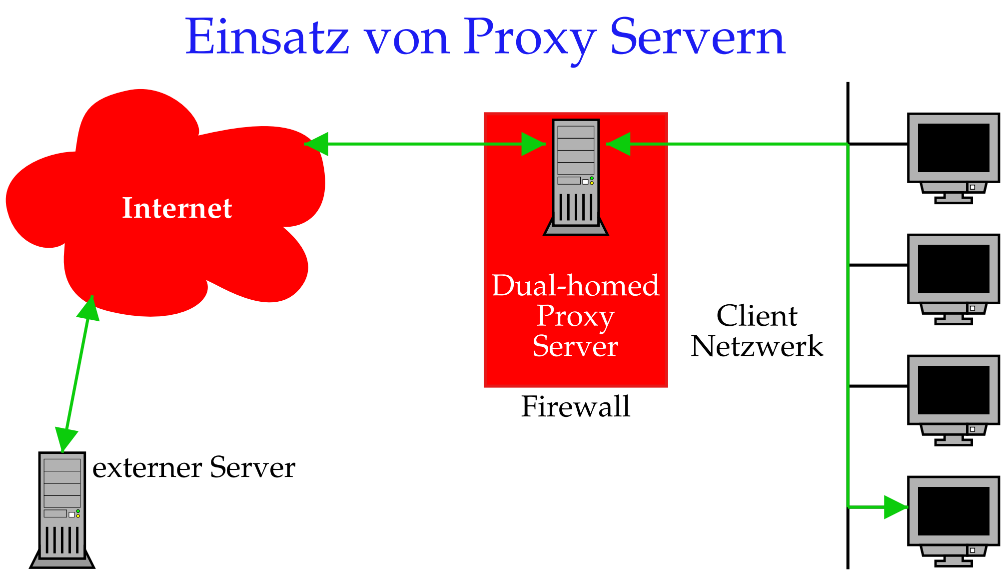

Abbildung 2.1: Einsatz eines Proxy-Servers zur Kontrolle des Netzwerkverkehrs auf Applikationsebene.

2.2 Firewall-Architekturen

Dual-Home Host Architektur

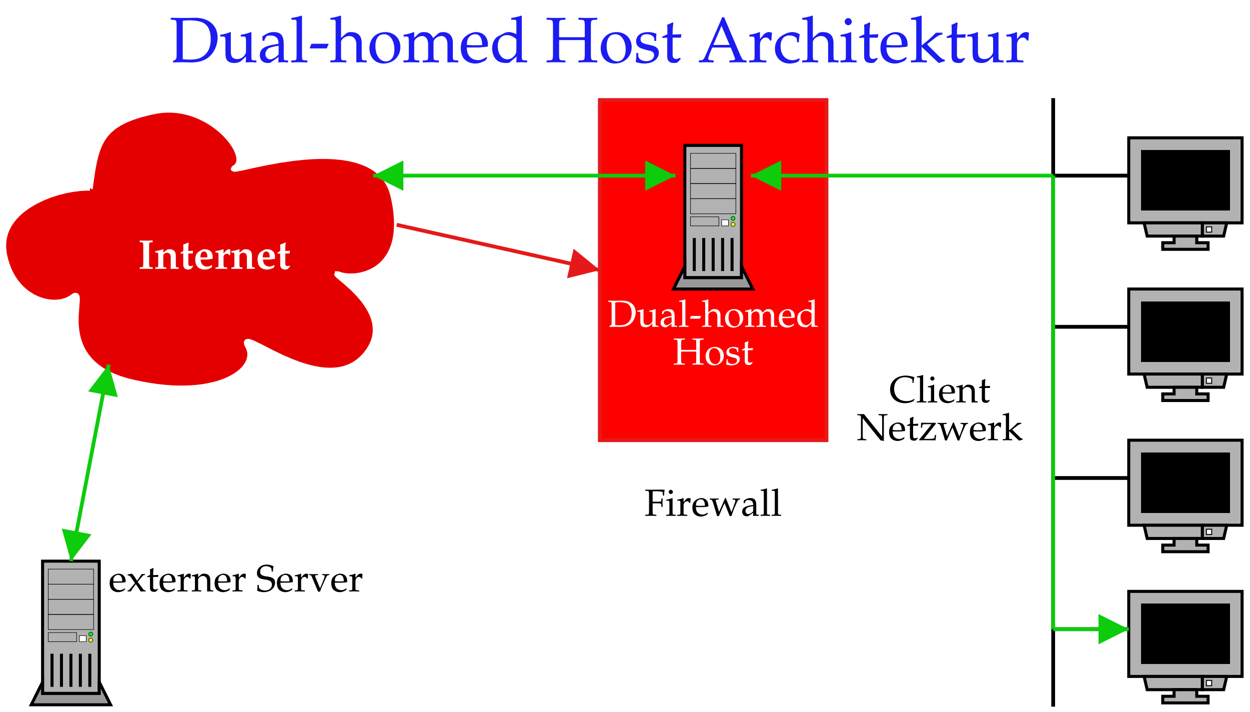

Abbildung 2.2: Einfache Firewall - ein dual-homed Host, der Verbindungen zu beiden Netzwerken hat.

Vorteile

- sehr sicher, wenn man ausschließlich Proxy-Services benutzt
- leicht aufzubauen und zu konfigurieren
- strikte Trennung zwischem internem und externem Datenverkehr[1]

Nachteile

- Proxy-Services sind nicht immer angebracht
- möglicherweise muß IP Masquerading oder NAT eingesetzt werden
  ⟶ größere Durchlässigkeit, geringere Kontrolle

[1] Es ist aber nicht ausgeschlossen, daß sich unter Umständen externe Pakete ins Innere verlieren.

2.2.1 Screened Host Architektur

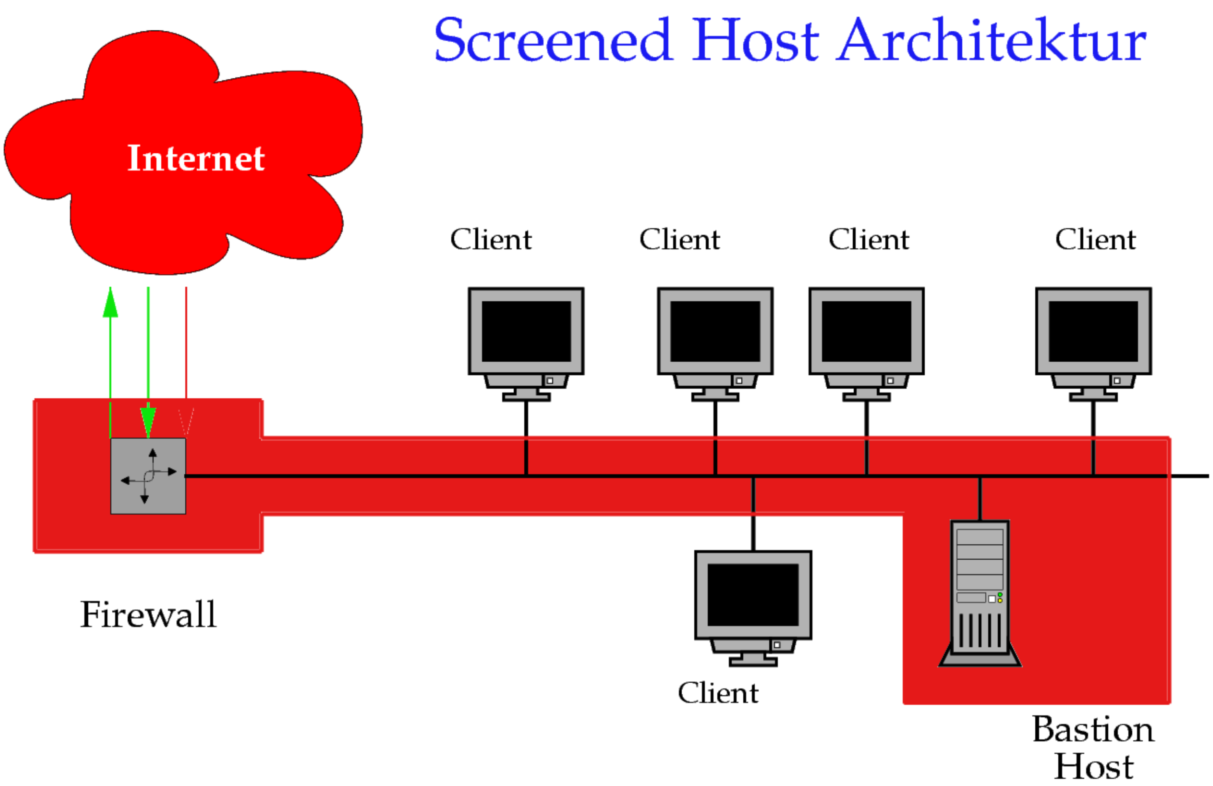

Abbildung 2.3: Hier sitzt ein Bastion Host im internen Netz, geschützt durch einen Router mit Packet Filtering. Nur der Bastion Host kann Verbindungen ins Internet öffnen. Alle Clients wenden sich an ihn.

- primäre Sicherheit durch Packet Filtering am Router
- nur Bastion Host kann Verbindungen ins Internet öffnen
- Bastion Host muß einen hohen Grad an Host Security aufweisen
- Bastion Host kann Proxy Services anbieten
- interner & externer Netzwerkverkehr mischt sich (durch Bastion Host im Inneren)
- **Point of Attack:** Router
- **Point of Attack:** Bastion Host

2.2.2 Screened Subnet Architektur

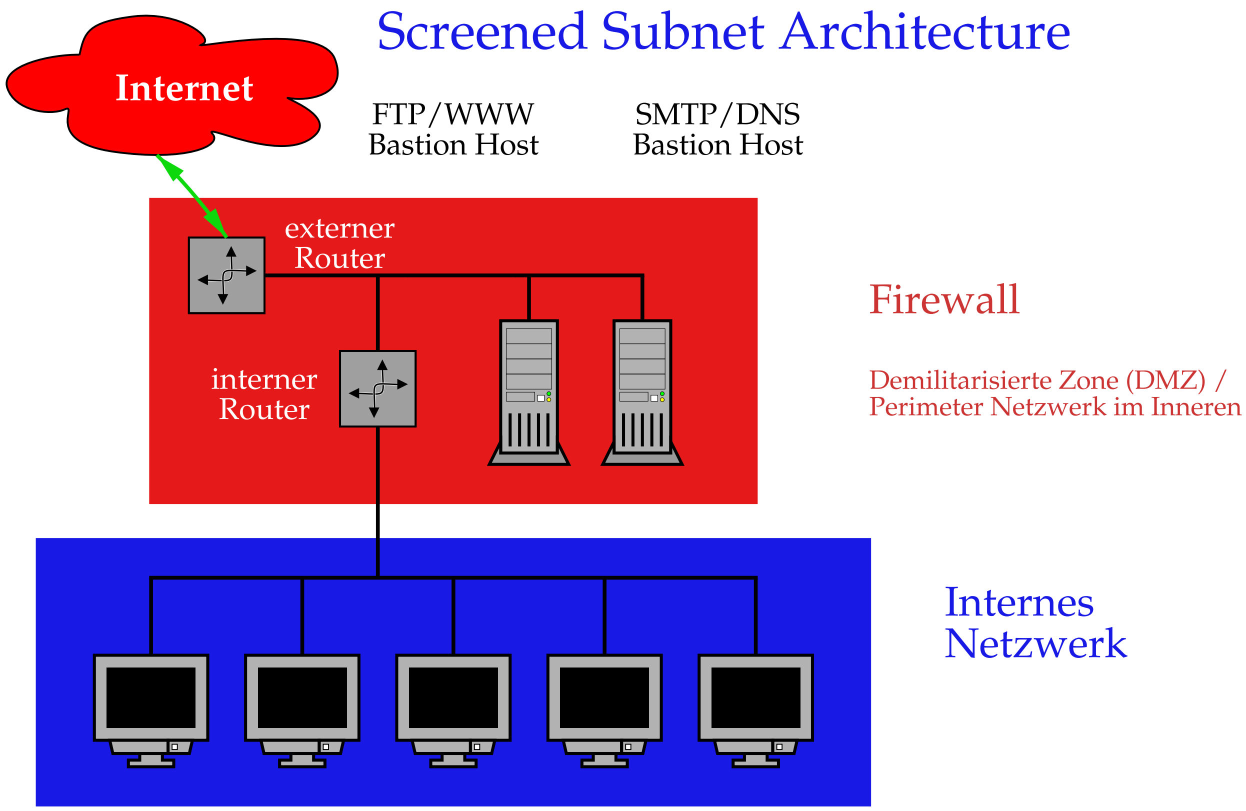

Abbildung 2.4: Zwischen Internet und internem Netz befindet sich nun eine Sicherheitszone, die den Bastion Host von internem Netzwerkverkehr isoliert.

- mehrschichtiger Verteidigungsaufbau (zwei Router bis zum internen Netzwerk)
- Perimeter Netzwerk
  isoliert internes Netzwerk
- externer Router
  stellt die Verbindung nach draußen her, möglicher Einsatz als erster Paketfilter
- interner Router
  dient als zweite Schutzmaßnahme vor dem internen Netzwerk
- Bastion Host
  dient als Hauptkontaktpunkt mit der Außenwelt

2.2.3 Mehrere interne Netzwerke

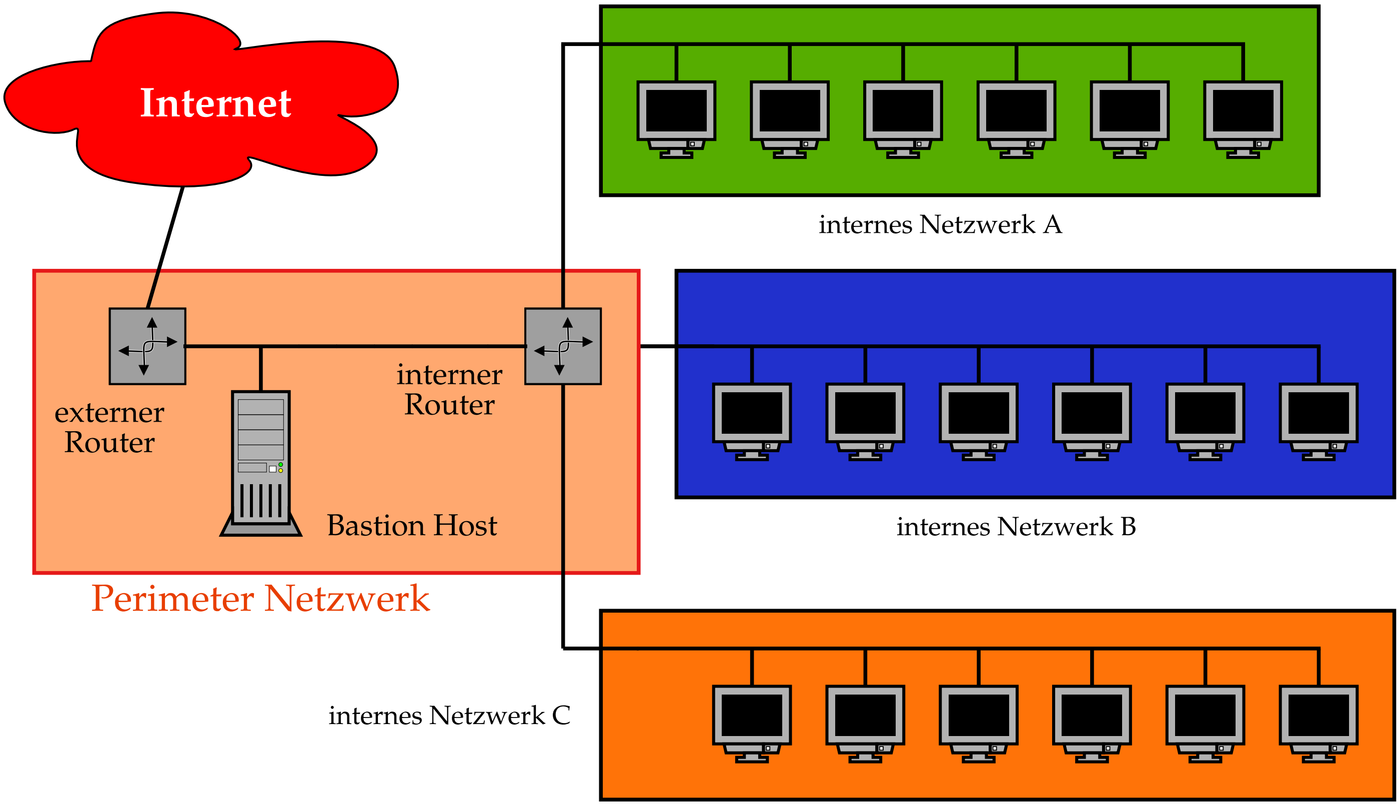

Abbildung 2.5: Anbindung mehrerer interner Netzwerke an eine Firewall. Die einzelnen Netzwerke können durch den internen Router zusätzlich auch noch abgesichert werden.

2.3 Einsatz und Aufbau von Bastion Hosts

Für den Aufbau von exponierten Servern („Bastion Hosts“) gibt es zahlreichen Rezepte und Checklisten. Sie dienen dazu, daß man möglichst wenig übersieht. Das IT Security Cookbook[2], das Firewall Buch von O'Reilly oder ähnliche Quellen haben Resourcen zu ausführlicheren Listen.
Wichtiger Grundsatz: *Keine Benutzer-Accounts am Bastion Host!*
Gründe für dieses Vorgehensweise:

- Schwachstellen der Accounts selber (z.B. Paßworte)
- Schwächen im Support dieser Accounts
- verringerte Stabilität und Verläßlichkeit des Hosts
- unabsichtliches Untergraben der Hosts Security durch User
- erhöhte Schwierigkeit Attacken zu entdecken
- verbesserte Sicherheit durch fehlende Benutzsoftware

Jeder zusätzliche Benutzer auf der Maschine vergrößert den Wartungsaufwand und damit das Sicherheitsrisiko.

[2] http://www.boran.com/security/index.html

Aufbau eines Bastion Hosts

1. Sichern der Maschine
2. Deaktivieren aller unnötigen Services
3. Installieren oder Ändern der benötigten Services
4. Rekonfigurieren der Maschine von Development in den Einsatzzustand
5. Testen und Prüfen der Sicherheit (Auditing)
6. *Zum Schluß:* Verbinden des Hosts mit dem Einsatznetzwerk

Diese Liste teilt die Installation in mehrere Abschnitte ein, die nun im Detail besprochen werden. Das CERT[3] hat eine UNIX Security Checklist v2.0[4], die ausführlicher ist. Für MS Windows Systeme gibt es ähnliche Resourcen (siehe beispielsweise die Windows 2000 Security Checklist[5]).

[3] http://www.cert.org/
[4] http://www.cert.org/tech_tips/usc20_full.html
[5] http://labmice.techtarget.com/articles/securingwin2000.htm

2.3.1 Sichern der Maschine

- minimale Installation des Systems
  $\longrightarrow$ **jede** gesparte Software ist eine Sorge weniger
- Beheben aller bekannten Systemfehler
  $\longrightarrow$ Verwenden der Vendor Updates
- Verwenden bzw. Aufstellen einer Checkliste, angepaßt je nach Verwendungszweck
- Sichern der System-Logs
- Testen der Maschine mit Scannern
  $\longrightarrow$ Portscans
  $\longrightarrow$ Nessus

2.3.2 Deaktivieren aller unnötigen Services

- Säubern der Init-Skripts
  ⟶ die Maschine muß nach dem Booten richtig eingestellt sein

- Säubern der `inetd` Services
  ⟶ Empfehlung: Einsatz von `xinetd`

- unnötige Services am Bastion Host sind beispielsweise

  - NFS und Verwandte
    nfsd, mountd, statd, lockd, automount, rquotad, amd
  - andere RPC Prozesse
    ypserv, ypbind, ypupdated, rexd, walld
  - Bootprotokolle
    tftpd, bootd, bootpd
  - BSD „r“ Kommandos
    rshd, rlogind, rexecd (alle anderen „r“ Kommandos laufen ohne diese nicht)
  - routed
    in der Regel wird man am Bastion Host nicht dynamisch routen
  - fingerd
  - ftpd
    anonyme FTP Server sollten nur mit besonders dafür geeigneter Software betrieben werden
  - uucp, rwhod, lpd (Printing)
  - IP Routing und Forwarding

  Insbesondere sollte man mit `netstat -nl` bzw. `netstat -anp -A inet | grep -i LISTEN` nach Netzwerkdiensten Ausschau halten.

2.3.3 Installieren oder Ändern der benötigten Services

- Ersetzen von Standardsoftware mit spezialisierter Software
  - Austausch von `inetd` durch `xinetd`
  - Einsatz der Secure Shell statt Telnet
  - Einsatz von `djbdns` statt BIND sofern möglich
  - Einsatz eines "sicheren" FTP Servers
    ⟶ NcFTPD[6]
    ⟶ ProFTPD[7]
    ⟶ vsftpd[8]
    ⟶ Dan Bernsteins Software[9]
- Einsatz des TCP Wrappers
  ⟶ Authentifizierung durch IP Adressen; einfacher, aber nützlicher Kontrollmechanismus
  Vorsicht! Access Rules nach IP Adressen stehen und fallen mit der Sicherheit des DNS (nicht nur der eigenen Server!)
- Installieren von Überwachungssoftware
  evtl. alle Aufgaben des Promiscuous Logging an einen seperaten Guardian Host weitergeben

[6]http://www.ncftpd.com/
[7]http://www.proftpd.net/
[8]http://vsftpd.beasts.org/
[9]http://cr.yp.to/

2.3.4 Rekonfigurieren der Maschine für Production Environment

- Konfigurieren des Kernels
  - keine Verwendung von Kernel Modulen
    ⟶ es existieren trojanische Module
  - keine Netzwerkfilesysteme (NFS, CODA, SMB/NetBIOS, AppleTalk)
  - keine Sniffing Tools (libpcap, Berkeley Packet Filter)
  - keine experimentellen Treiber
    bei Einsatz von β-Software diese beobachten
  - nicht zuviele Kernel Images auf der Maschine halten
- Entfernen aller unwichtigen Programme
  - Bastion Host ist *keine* Arbeitsumgebung
  - Entfernen der Entwicklungsumgebung
    Compiler, Header Files, Build Tools
  - Software für Graphical User Interfaces
  - Programme mit *setguid/setuid* Fähigkeiten
    ⟶ `find / -type f -perm +6000`

  Alternativ: Ersetzen dieser Programme mit Alarmierungstools, die nach Aufruf eine Notiz an die Administratoren senden
- Filesysteme als Read Only konfigurieren
  ⟶ bei BSD basierten Systemem das *immutable* Flag setzen
  ⟶ Linux Ext2 Filesystem hat einige Optionen zur Verbesserung der Sicherheit
- reine Datenfilesysteme mit `noexec` konfigurieren
- Backup des ganzen Systems
- Bilden und Archivieren von Checksummen aller statischen Files

2.4 Checklisten für ausgewählte Applikationen

Die folgenden Listen sollen einen kurzen Einblick in das Absichern von bestimmten Applikationen geben. Man sollte, gleichgültig um welche Software es sich handelt,

- einen eigenen Benutzer und eine eigene Gruppe dafür anlegen,
- die Zugriffsrechte auf das Filesystem sorgfältig wählen,
- Quoten bzw. Limits auf Resourcen festsetzen, um Plattenplatz, CPU-Zeit, Anzahl offener Dateien oder ähnliche Größen zu begrenzen,
- die Applikation in eine `chroot` Umgebung einsperren, sofern dies möglich ist.

2.4.1 Apache

- Rechte am Serverroot so wählen, daß Benutzer des Webservers die Binaries nicht ändern kann
- Server Side Includes (SSI) deaktivieren bzw. nur für spezielle Extensions (z.B. `.shtml`) konfigurieren
- CGIs nur von vertrauenswürdigen Benutzern ausführen lassen
- CGIs wenn möglich mit der suEXEC Option konfigurieren (jeweils ein CGI User pro Virtual Host)
- Nutzung von `.htaccess` Dateien möglichst einschränken (`AllowOverride None`)
- Verzeichnisse in `/` gegebenenfalls vor Auslesen durch den Webserver schützen (siehe `<Directory>`)
- Include-Pfad für PHP außerhalb des Document Roots legen
- Wichtige Dateien und Verzeichnisse durch ein `robots.txt` vor Suchmaschinen schützen

2.4.2 BIND

- Betrieb des Servers in einer `chroot` Umgebung
- Einschränken der Cache-Funktion auf ausgewählte Netzwerke
  ```
  acl validnets { 127.0.0.0/8; 10.0.0.0/24; 62.116.64.105/32; }
  allow-recursion { validnets; };
  allow-query { validnets; };
  ```
- Einschränken des Zonentransfers auf vertrauenswürdige Maschinen
  ```
  allow-transfer { 10.0.0.0/24; }
  ```
- Ändern des Versionsstrings
  ```
  version Name's BIND...James BIND;
  ```

2.4.3 FTP Server

- Betrieb des Servers in einer `chroot` Umgebung (möglicherweise per FTP User)
- kein Betrieb eines anonymen FTP Servers
- Einschränken der Server Capabilities (z.B. Permissionvergabe, Site Exec, etc.)

2.4.4 Mail Transport Agents (MTAs)

- Korrektes Einbinden des Servers in den DNS (reverse & forward DNS)
- Konfigurieren des MTAs gegen unerlaubtes Relaying
- Benutzung geeigneter Schutzmaßnahmen, falls Skripte vom MTA aufgerufen werden müssen (z.B. Mailing List Manager)

2.4.5 PHP

- Abfangen aller gängigen PHP Extensions durch den Interpreter (`.php`, `.php3`, `.phtml`, `.inc`, `.php4`, etc.)
- Aktivieren des PHP Safe Mode[10]
- Aktivieren der *register globals* Option (Default bei neueren PHP-Versionen)
- Prüfen aller Limits in der `php.ini` Datei
- Prüfen aller Permissions, damit PHP-Skripte nicht im Dateisystem schrankenlos Zugriff haben

[10] http://www.php.net/features.safe-mode

Kapitel 3

Angriffe auf Netzwerke und Server

```
"All we know is the phenomenon: we spend our time sending messages to each
 other, talking and trying to listen at the same time, exchanging information.
 This seems to be our most urgent biological function; it is what we do with
 our lives."
                -- Lewis Thomas, "The Lives of a Cell"
```

3.1 Tools zur Aufklärung

Ohne Werkzeug geht es nicht, das gilt sowohl für Angreifer als auch für Verteidiger. Sie erleichtern die Arbeit und ermöglichen es in kurzer Zeit bequem[1] Tests durchzuführen. Es gibt zusätzlich noch ganze Attack Frameworks und Test Suites zu ausgewählten Applikationen und Protokollen.

3.1.1 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**[2]
- **TCP Stream Verfolgung**
- **TCP Analyse**
  - Sequenznummeranalyse (Zeit-Sequenznummer Graph)
  - Datendurchsatz Graph
  - RTT Graph

Ethereal besitzt auch ein Tool namens `tethereal`, welches sich ähnlich `tcpdump` von der Console einsetzen läßt.

[1] Die Definition von „bequem“ hängt stark vom individuellen Wissensstand ab.
[2] http://www.ethereal.com/docs/user-guide/appfiltfields.html

3.1.2 hping2 - Paketgenerator

`hping2`[3] 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

3.1.3 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`

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

3.1.4 SING - ICMP Packetgenerator

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

- ICMP[4] Handling gibt Aufschluß über OS
  ⟶ `http://www.sys-security.com/archive/papers/ICMP_Scanning.pdf` ⟶ `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

3.1.5 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[5] Scannen bei TCP connect()
- **Patch erlaubt Versions-Identifikation der Dämonen**
- **aktiv unterstützte OS Fingerabdruck Datenbank**
  nmap V3.00 erkennt über 700 Systeme

[4] RFC 792, 1122, 1256, 1822
[5] Identification Protocol, RFC1413, `http://dungeon.luchs.at/RFC/rfc1413.txt`

- **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**

3.1.6 Sonstige Tools

- airsnort[6] - WLAN Tool, um verlorene Encryption Keys „wiederzufinden\.

- dsniff[7] - Sniffer für Protokolle, die Verschlüsselung verwenden

- kismet[8] - 802.11 Layer2 Wireless Network Detektor, Sniffer und Intrusion Detection System

- netcat / cryptcat - Tools zum Transferieren von Daten zwischen zwei Ports. cryptcat überträgt mit Blow Fish Encryption.

- ping - Standard ICMP Tool

- traceroute

- tracepath

- mtr

- Auf der lokalen Maschine

  - ip oder ifconfig

[6]http://airsnort.shmoo.com/
[7]http://www.monkey.org/ dugsong/dsniff/
[8]http://www.kismetwireless.net/

Kapitel 4

Intrusion Detection Systeme

```
"The possession of knowledge does not kill the sense of wonder and
 mystery. There is always more mystery."
  --- Anais Nin ---
```

4.1 Network Intrusion Detection mit Snort

Die Fähigkeiten von Snort[1] 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.
- **Stateful Inspection**
  Streams können verfolgt und analysiert werden
- **Stealth Modus**
  Snort benötigt keinen TCP/IP Stack auf dem System

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

4.2 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

4.3 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**[2]
  - 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.

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

4.4 Samhain / Yule - Client/Server HIDS

Samhain[3] 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**

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

4.5 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

4.6 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?*

4.7 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*

4.8 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*

Kapitel 5

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
```

5.1 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*

5.2 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

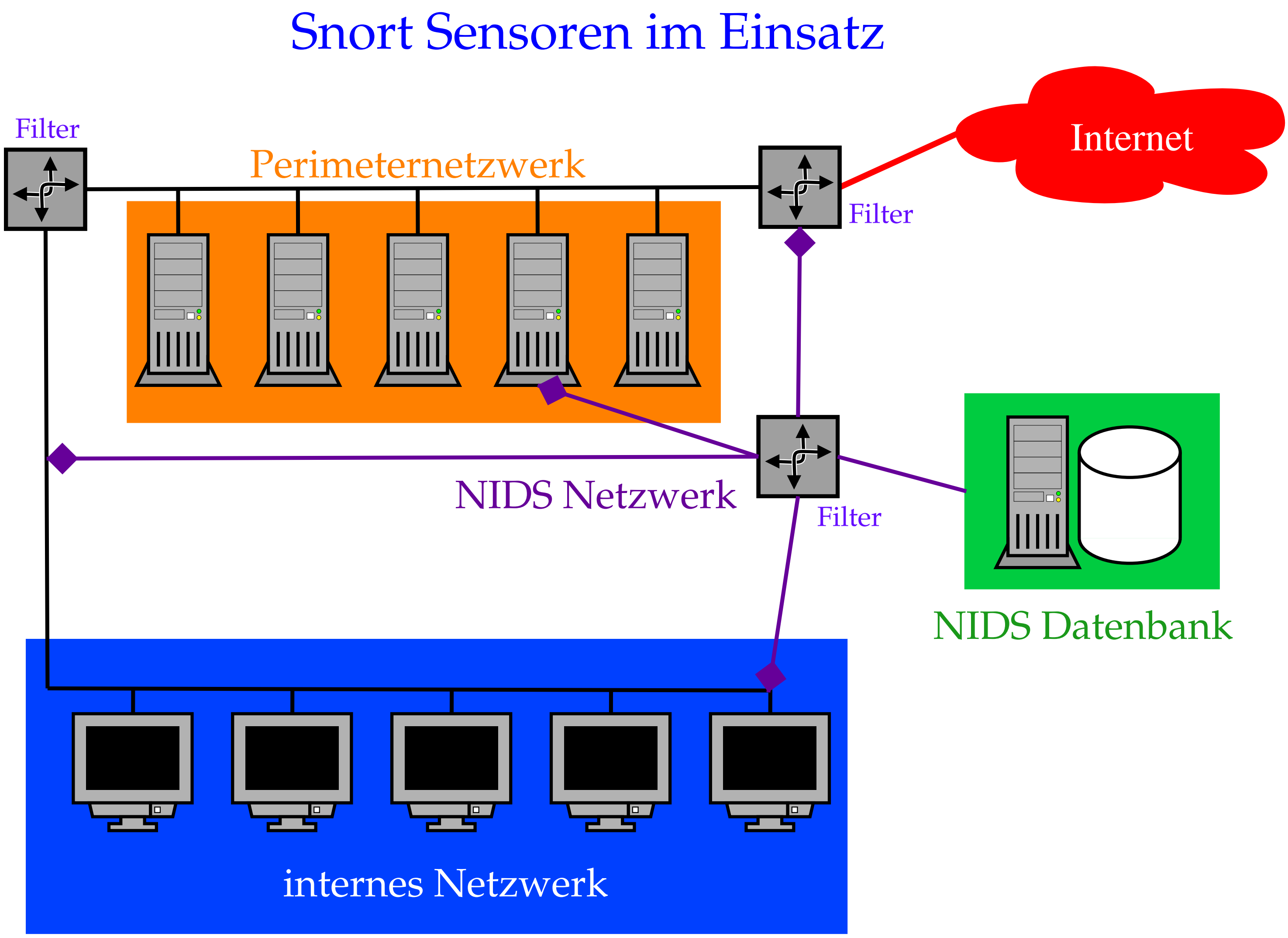

Abbildung 5.1: Schematischer Einsatz von Snort Sensoren in einem Netzwerk. Man kann Sensoren auf Paketfiltern, Servern oder auf speziellen Sondenservern unterbringen. Der sinnvollste Einsatz geschieht auf einer eigenen Maschine, die ausgewählten Netzwerkverkehr über einen dafür abgestellten Port an einem Switch erhält.

5.3 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**

5.4 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

5.5 Network TAPs

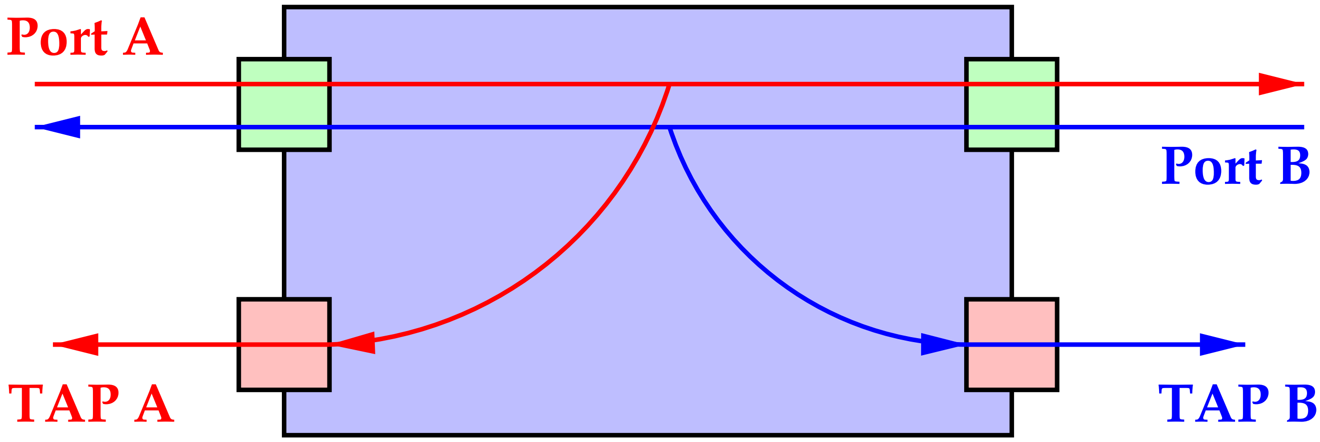

Abbildung 5.2: Das Diagramm zeigt die schematische Ansicht eines Network TAPs. TAP A fängt Pakete von an Port A angeschlossenen Geräten ab, TAP B welche von an Port B angeschlossenen Geräten. Eine Kombination von TAP A und TAP B ergibt den gesamten Netzwerkverkehr. (Quelle: Internet Security Systems)

- **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*

5.6 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`

5.7 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*

5.8 Mehrstufige Architektur für Hochleistungs IDS

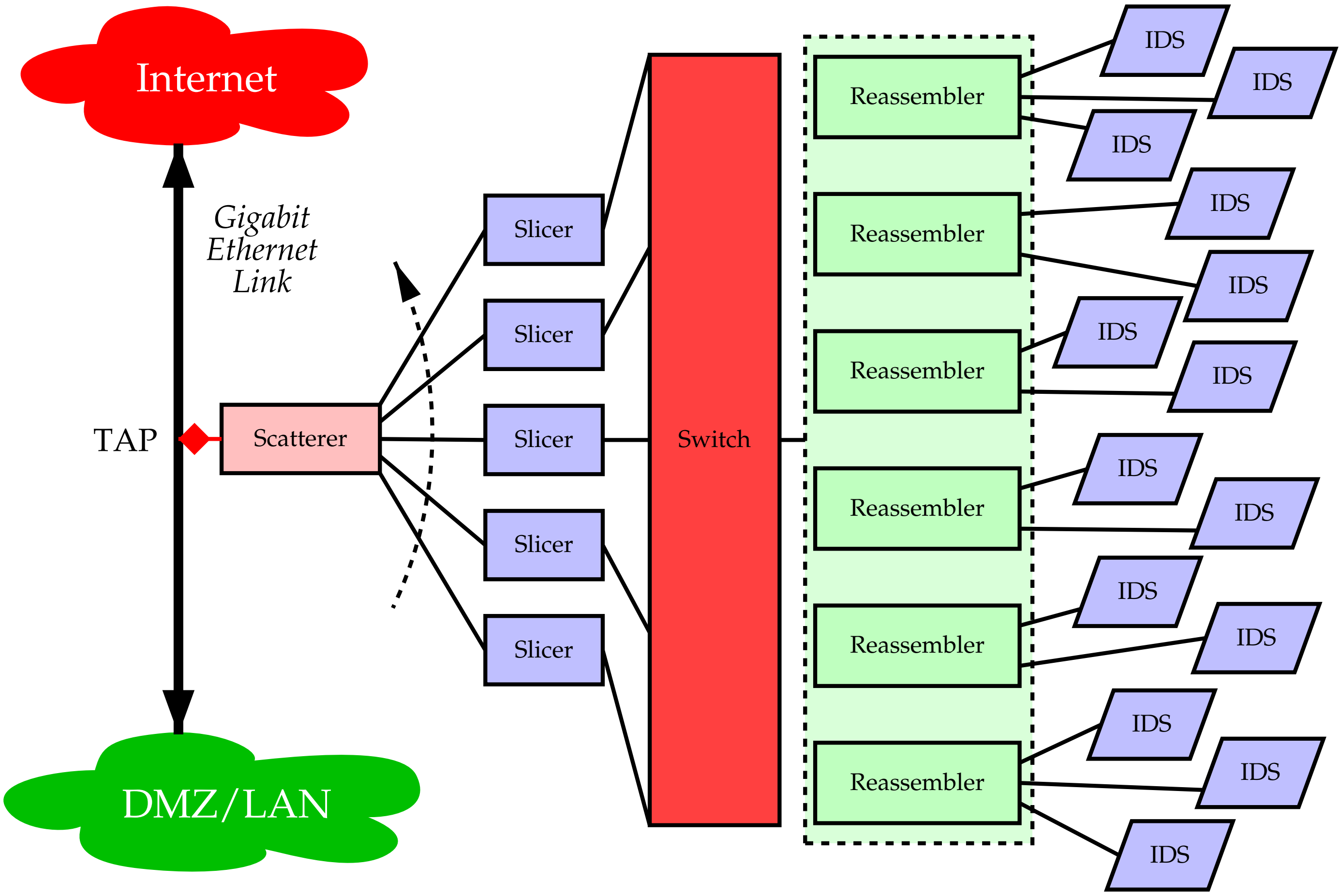

Abbildung 5.3: Schema eines verteilten Intrusion Detection Systems für Hochleistungsnetze. Ein Scatterer greift alle Daten eines Zeitabschnitts ab und verteilt sie nach einem Splitting Algorithmus auf verschiedne Slicer. Jeder Slicer sortiert nun die einzelnen Ethernet Frames an einen oder mehrere Sensorkanäle. 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`)

5.9 Zentralisieren der Konfiguration

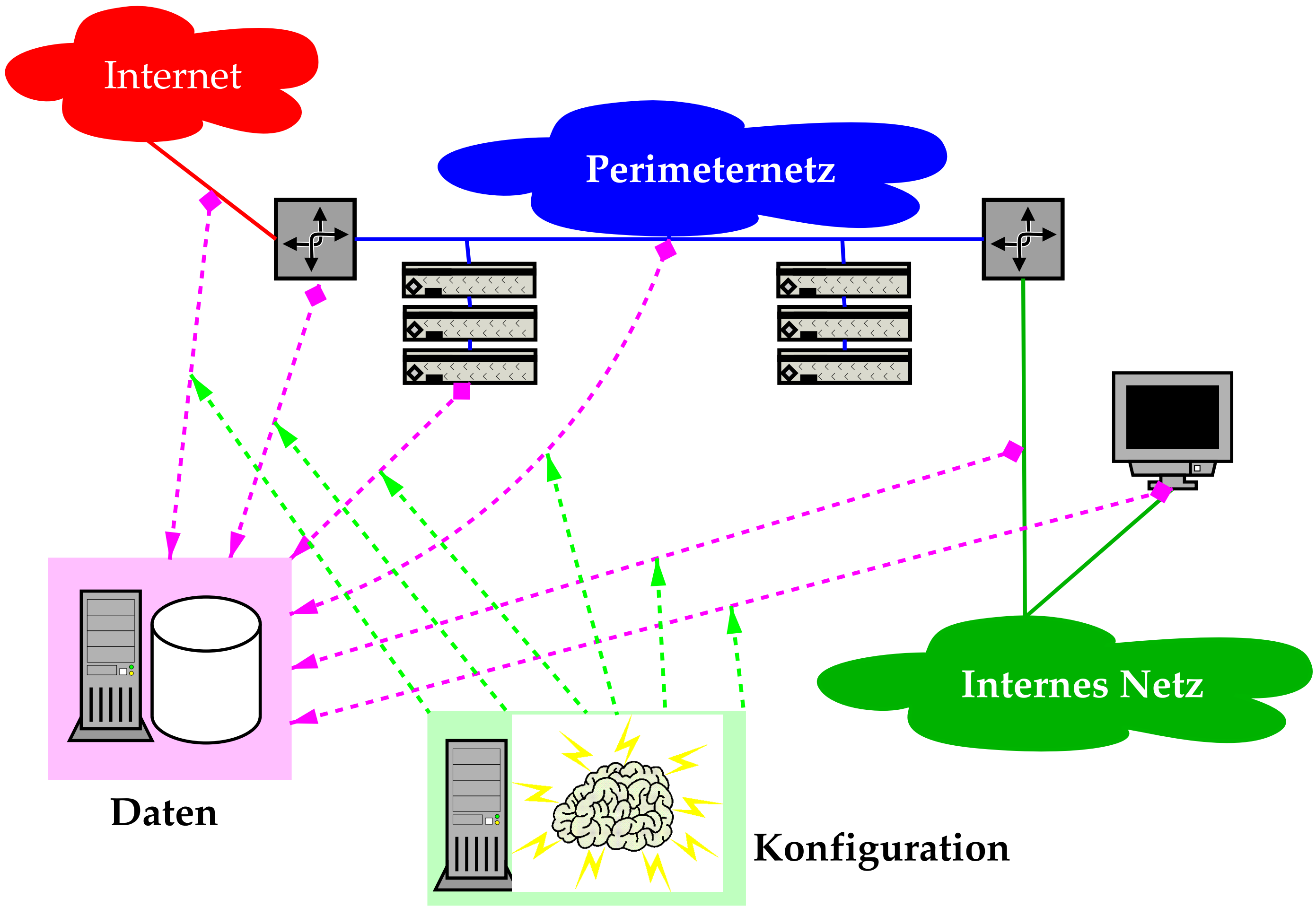

Abbildung 5.4: Ausgedehnter Einsatz von IDS Maßnahmen erfordert ein gewisses Maß an zentralisiertem Sammeln der Daten. Dies gilt ebenso für die Verteilung der Konfigurationen an die verschiedenen Sensoren.

5.10 Konsolidierung

- **IDS werden in bestehenden Netzen meist nachgerüstet**
  ⟶ *Nutzung bestehender Log Infrastruktur*
  ⟶ *Platzierung von HIDS auf exponierte Systeme*
  ⟶ *NIDS hinter Paketfiltern*

- **Wahl der eingesetzten Systeme gut planen**
  ⟶ *Interoperabilität zwischen Herstellern*
  ⟶ *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**
  ⟶ *leichtere Migration auf vereinheitlichte Hardware*
  ⟶ *Beibehaltung der Konfiguration*

5.11 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

5.12 Hybrid-IDS mit Datenbankauswertung

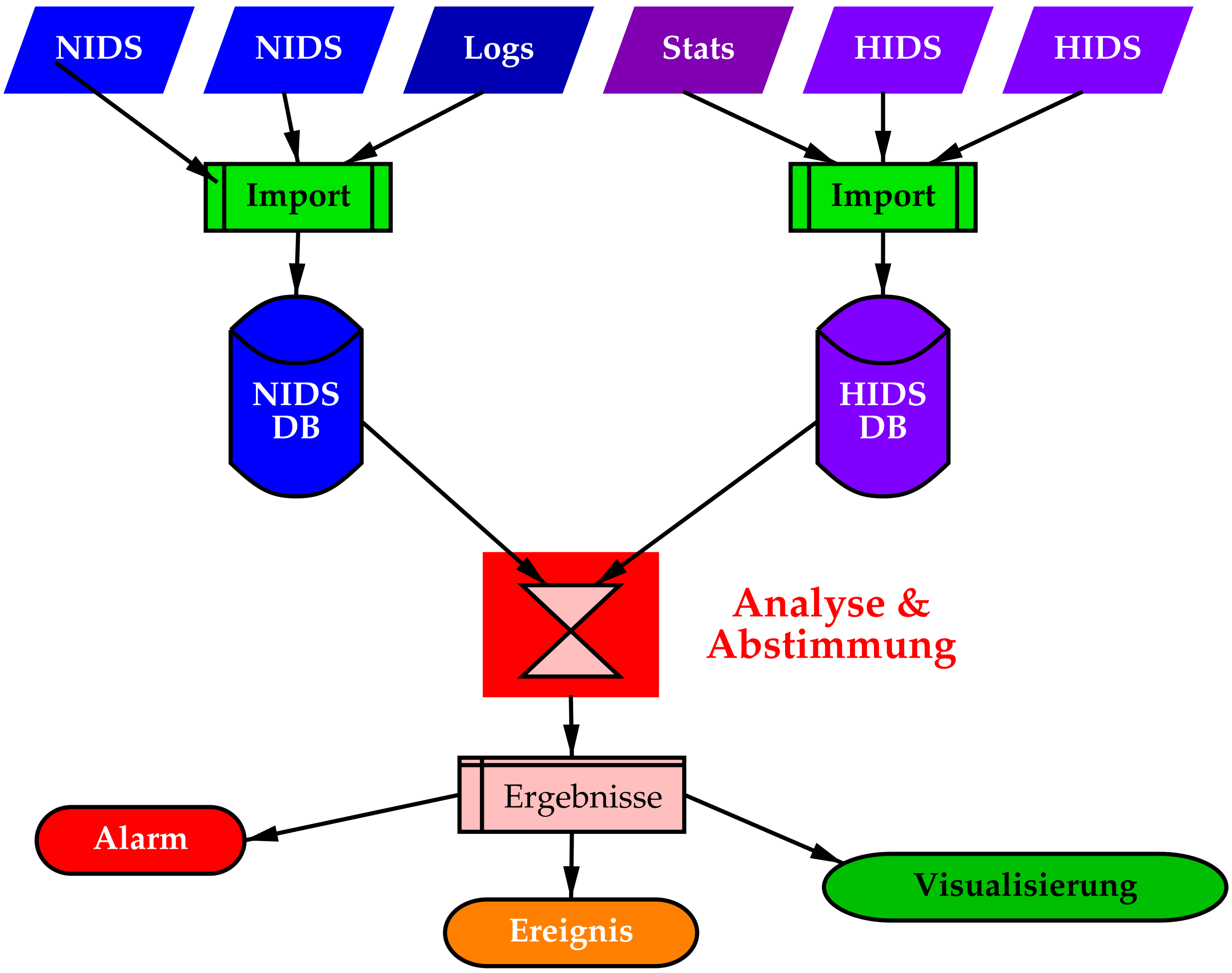

Abbildung 5.5: Ausgedehnter Einsatz von IDS Maßnahmen erfordert ein gewisses Maß an zentralisiertem Sammeln der Daten. Dies gilt ebenso für die Verteilung der Konfigurationen an die verschiedenen Sensoren. Die Sammlung der Daten führt in eine statistische Auswertung. Das Befüllen kann in größeren periodischen Abständen (z.B. Tagen) erfolgen, um Performance-Engpässe auszugleichen.

5.13 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
```

5.13.1 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

5.13.2 Grundsätzliche Quellen für rechtliche Aspekte

- **EU Datenschutzrichtlinie**[1]
- **DSG 2000**[2]
  - 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[3] hat Praxisbeispiele in ihrem Newsletter.

[1] http://www.ad.or.at/office/recht/eu.htm
[2] http://www.ad.or.at/office/recht/dsg2000.htm
[3] http://www.ad.or.at/

5.13.3 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*

5.13.4 Automationsmöglichkeiten - Übersicht

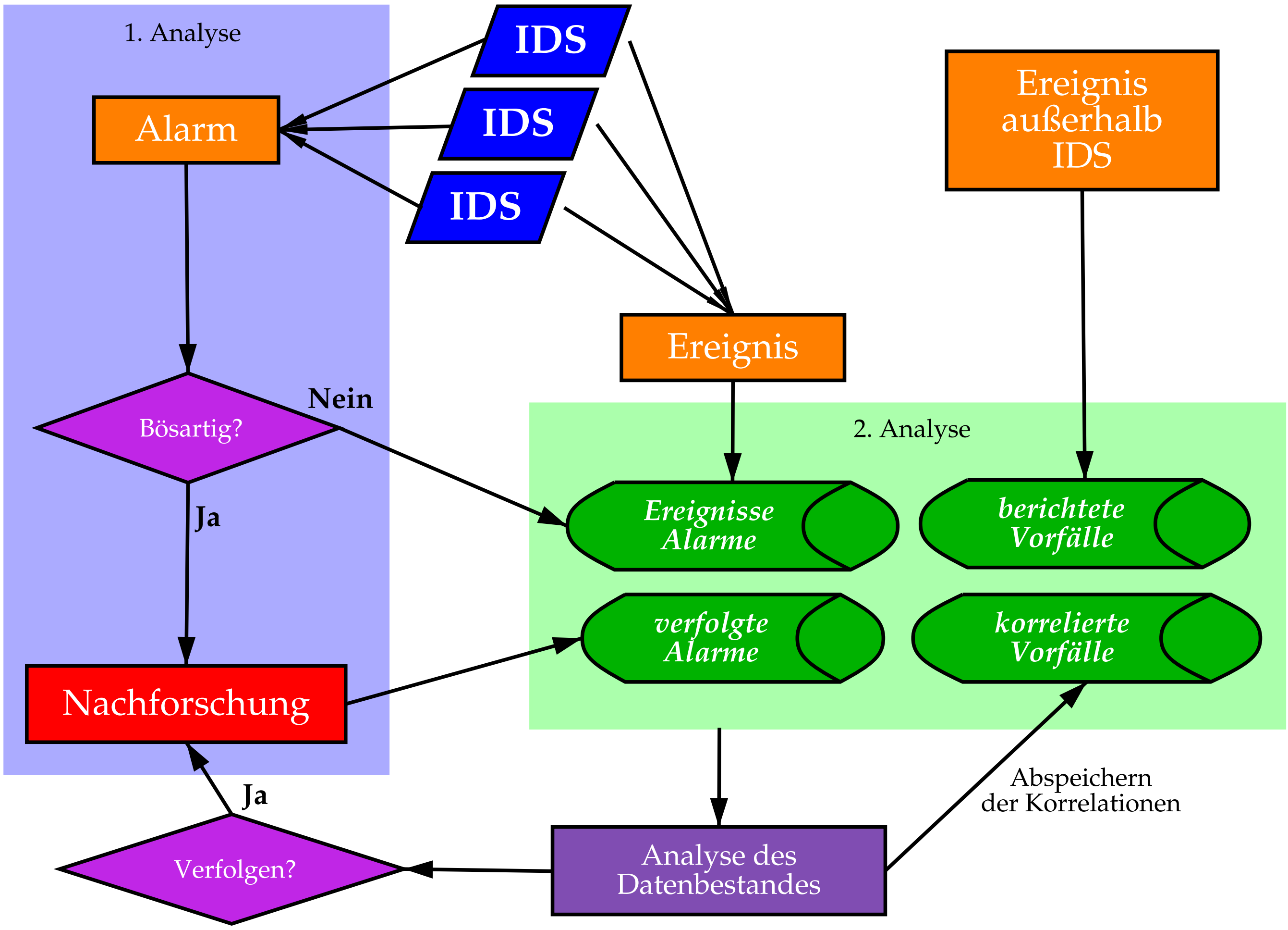

Abbildung 5.6: Die Abbildung zeigt die mehrstufige Auswertung von IDS Daten. In der ersten Stufe wird unmittelbar auf einen Alarm reagiert. Die Daten werden in Datenbanken gesammelt und durch weitere Prozesse analysiert. Dadurch können sich weitere Alarme ergeben, die behandelt werden müssen. (`http://www.securityfocus.com/infocus/1201`)

5.13.5 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**

5.13.6 Beispiel für kontinuierliches Monitoring

- **Überwachung mehrerer Maschinen durch Samhain Sensoren**
  ⟶ *Meldung aller Ereignisse an Yule Prozeß am Logserver*

- **DMZ und LAN wird durch Snort Server beobachtet**
  ⟶ *je eine Netzwerkkarte pro Snort Sensor Prozeß*
  ⟶ *lokale Packet Capture Logs*
  ⟶ *Ereignisse gehen an Logserver*

- **periodisches Wandeln von System- und Maillogs in SQL**
  ⟶ *Perl Skripte konvertieren syslodg & Sendmail Logs*

- **zentrales Logging an Logserver**
  ⟶ *PostgreSQL Datenbank*
  ⟶ *direkte Generierung von Email Alerts*

- **Aufbereiten der Daten zur Analyse**
  ⟶ *Import der Daten in IBM DB2 zur OLAP Analyse*
  ⟶ *weitere statistische Analyse auf PostgreSQL und DB2*

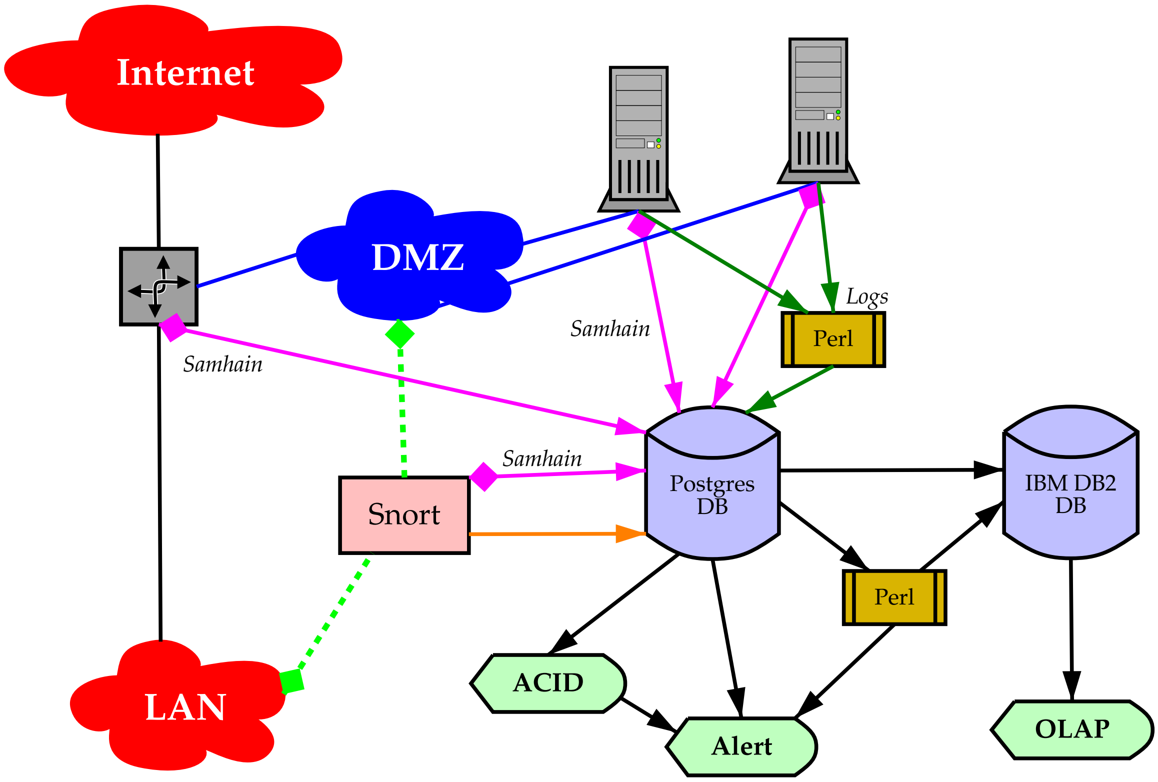

Abbildung 5.7: Die Abbildung zeigt die kontinuierliche Auswertung von IDS Daten. Es spielen hier NIDS und HIDS Komponenten zusammen, die verschiedene Applikationen, Server und Netzwerke überwachen. Zentraler Sammelpunkt für die Rohdaten ist die PostgresSQL Datenbank. Dort findet die erste Stufe der Auswertung statt, wo auf bestimmte Ereignisse sofort reagiert werden kann. Weitere Auswertungen geschehen auf der DB2 Datenbank.

5.13.7 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[4]
- **Verarbeitung der Feature Sets durch Data Mining**
  - IBM Intelligent Miner[5]
    kann direkt mit IBM DB2 verwendet werden
  - Torch[6]
    freie C++ Bibliothek mit Implementation verschiedener Machine Learning Methoden
  - WEKA[7]
    freie Data Mining Software geschrieben in Java
- **Mining Analyse kann Anomalien und unbekannte Attacken aufspüren**

[4] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[5] http://www-3.ibm.com/software/data/iminer/
[6] http://www.torch.ch/
[7] http://www.cs.waikato.ac.nz/˜ ml/weka/