IT-Sicherheit

- Sicherheit vernetzter Systeme -

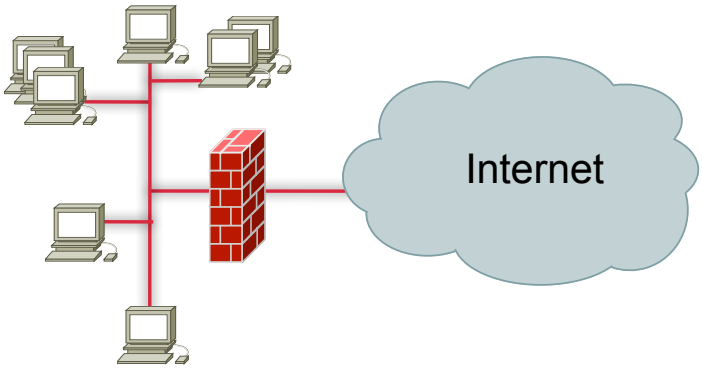
Kapitel 14: Firewalls und Intrusion Detection Systeme (IDS)

Inhalt

- Firewall Klassen
  - Paketfilter
  - Applikationsfilter
  - Verbindungs-Gateway
- Firewall Architekturen
  - Single Box
  - Screened Host
  - (Multiple) Screened Subnet(s)

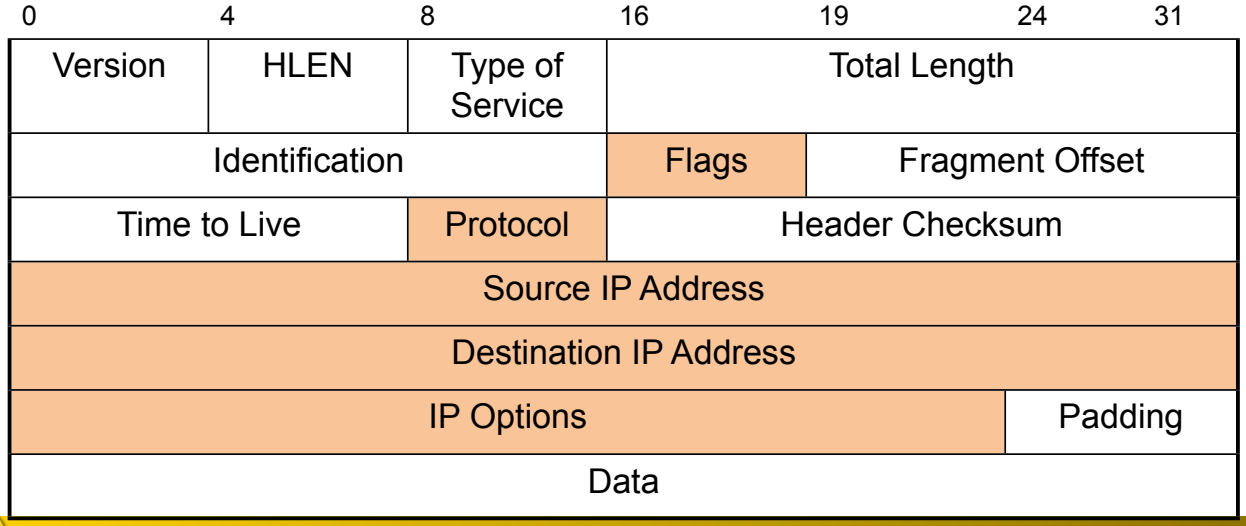
Firewall-Techniken

- Firewall:
  - Besteht aus einer oder mehreren Hard- und Softwarekomponenten
  - Koppelt zwei Netze als kontrollierter Netzübergang
  - Gesamter Verkehr zwischen den Netzen läuft über die Firewall (FW)
  - Realisiert Sicherheitspolicy bezüglich Zugriff, Authentisierung, Protokollierung, Auditing,....
  - Nur Packete die Policy genügen werden „durchgelassen“
- Klassen:
  - Paketfilter
  - Applikationsfilter (Application Level Gateway)
  - Verbindungs-Gateway (Circuit Level Gateway)
  - Kombinationen daraus



Paketfilter

- Filtern auf OSI Schichten 3 und 4
- Filter-Informationen aus den Protokollpaketen
- Im folgenden Beispielhaft TCP / IP
- IP Paket:



Packetfilter: TCP-Paketformat

- Bei Packetfilter-FW nur Regeln über Felder der Packet-Header möglich!

4	10						16	24	31
Source Port							Destination Port		
Sequence Number									
Acknowledge Number									
Data Offset	re-served	U R G	A C K	P S H	S Y N	Fl N	Window-Size		
Checksum							Urgent-Pointer		
Options							Padding		
Data									

FW für TCP/IP : Granularität der Filter

- Schicht 3: wesentliche Filter-Kriterien:
  - Quell-
  - Zieladresse
  - Protokoll der Transportschicht (z.B. TCP, ICMP, UDP,... Vgl. `/etc/protocols`)
- FW auf IP-Basis kann damit Endsysteme filtern (erlauben, verbieten)
- IP-Spoofing u.U. erkennbar, falls:
  - Packet mit interner Quell-Adresse
  - kommt an externem FW-Interface an
- Keine Filterung auf Ebene der Dienste möglich
- Schicht 4: wesentliches Filter-kriterium:
  - Quell- sowie
  - Zielport
  - Flags
- Über Port-Nr. werden Well-Known Services identifiziert; z.B. Port 23 = Telnet (vgl. `/etc/services`)
- Allerdings nur Konvention; OS setzt diese nicht automatisch durch
- Weitere Konventionen:
  - privilegierte Ports (Ports < = 1023) für Systemdienste
  - Ports > 1023 für jeden nutzbar
- Flags zum Verbindungsauf- und –abbau (vgl. Kap. 3 SYN Flooding)

Packetfilter: Filterregeln

- Grundsätzliche Ansätze:
  1. Alles was nicht explizit verboten ist, wird erlaubt.
  2. Alles was nicht explizit erlaubt ist, wird verboten.
- Reihenfolge der Regeln wichtig:
  - Regel die zuerst zutrifft wird ausgeführt („feuert“)
  - Daher im Fall 2. letzte Regel immer: alles verbieten
- Statische Paketfilterung
  - Zustandslos
  - Pakete werden unabhängig voneinander gefiltert
- Dynamische Paketfilterung (Statefull Inspection)
  - Zustandsabhängig
  - FW filtert abhängig vom Zustand des Protokoll-Automaten
- Grundsatz: KISS
  Keep it Small and Simple

Packetfilter-Regeln: Beispiele

- Statischer Paketfilter:
  - Ausgehender Telnet Verkehr erlaubt,
  - Eingehender Telnet Verkehr verboten

Regel	Source	Destina-tion	Proto-col	Source Port	Dest. Port	Flags	Action
1	<intern>	<extern>	TCP	>1023	23	Any	Accept, Log
2	<extern>	<intern>	TCP	23	>1023	!SYN	Accept
3	Any	Any	Any	Any	Any	Any	Drop, Log

- Dynamischer Packetfilter

Regel	Source	Destina-tion	Proto-col	Source Port	Dest. Port	Action
1	<intern>	<extern>	TCP	>1023	23	Accept, Log
2	Any	Any	Any	Any	Any	Drop, Log

Bewertung Packetfilter

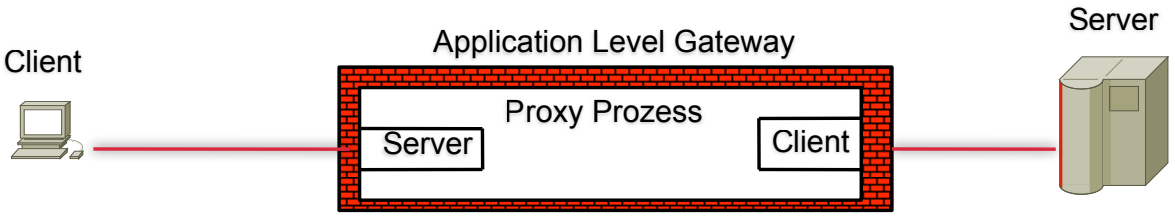
- Einfach und preiswert
- Effizient
- Gut mit Router-Funktionalität kombinierbar (Screening Router)
  - Integrität der Header Informationen nicht gesichert; alle Felder können relativ einfach gefälscht werden
  - Grobgranulare Filterung
  - Keine inhaltliche Analyse bei freigegebenen Diensten
  - Statische Strategie: Damit Problem bei Diensten, die Ports dynamisch aushandeln (z.B. Videokonferenz-Dienst)
  - Abbildungsproblematik: Policy auf konkrete FW-Regeln
  - Erstellung einer Filtertabelle nicht triviale Aufgabe
    - Korrektheit ?
    - Vollständigkeit ?
    - Konsistenz ?

Weitere Firewall-Techniken auf Schicht 3/4

- Network Adress Translation (NAT)
  - Intern werden andere Adressen („private IP-Adr.“) oder Ports als extern verwendet
  - FW macht Adress/Port-Umsetzung
  - Statisch oder dynamisch
- Masquerading
  - Alle ausgehenden Pakete erhalten Adresse der FW
  - Gesamtes internes Netz wird verborgen
- Anti-Spoofing
  - Binden von FW-Regeln an bestimmte Interfaces (ingress, egress)
  - Wenn an externem Interface ein Packet mit interner Quell-Adresse ankommt muss dieses gefälscht sein

Applikationsfilter (Application Level Gateway)

- Filtern auf Schicht 7 (Anwendungsschicht)
- Analyse des Anwendungsschichtprotokolls u. d. Protokolldaten
- Für **jeden** Dienst, jedes Protokoll eigener Filter Prozess (auch als **Proxy** bezeichnet) erforderlich
- Interner Client muss sich i.d.R. am Proxy authentisieren
- Proxy trennt Verbindung zwischen Client und Server
- Nach außen erscheint immer nur die Adresse des Application Level Gateways; völlige Entkoppelung von internem und externem Netz
- Kann Zustandsinformationen halten und nutzen



Proxies

- Für viele Standarddienste verfügbar (z.B. HTTP, Telnet, FTP,..)
- Problematisch für „proprietäre“ Dienste (SAP R3, Lotus Notes,....)
- Beispiel eines HTTP Proxies: Squid
  - Umfangreiche Access Conrol Listen (ACL) möglich:
    - Quell- / Zieladresse
    - Domains
    - Ports
    - Protokolle
    - Protokoll-Primitive (z.B. FTP `put`, HTTP `POST`)
    - Benutzernamen
  - Benutzerauthentisierung am Proxy
  - Zusätzlich Caching-Funktionalität
  - Beispiel:
    - `acl SSL_PORT port 443` (Definition von SSL Ports)
      `acl AUTH proxy_auth REQUIRED` (Benutzerauthentisierung)
      `http_access deny CONNECT !SSL_PORT` (Alle Verbindungen zu einem anderen Port außer SSL verbieten)

Bewertung Applikationsfilter

- Feingranulare, dienstspezifische Filterung
- Umfangreiche Logging Möglichkeit und damit Accounting
- Zustandsbehaftet
- Inhaltsanalyse (damit z.B. Filterung aktiver Inhalte möglich)
- Benutzerauthentisierung und benutzerabhängige Filterung
- Entkopplung von internem und externem Netz
- Möglichkeit der Erstellung von Nutzungsprofilen

- ✶ Möglichkeit der Erstellung von Nutzungsprofilen
- ✶ Jeder Dienst braucht eigenen Proxy
- ✶ Sicherheit der Proxy Implementierung??
- ✶ Problem von Protokollschwächen bleibt bestehen

Verbindungs-Gateway (Circuit Level Gateway)

- Filtert auf Schicht 7; spezielle Ausprägung des Application Level Gateway
- Circuit Level Gateway stellt generischen Proxy dar
- Nicht auf einzelne Dienste zugeschnitten, allgemeiner „Vermittler“ von TCP/IP Verbindungen
- Trennt Verbindung zwischen Client und Server
- Benutzerauthentisierung am Gateway möglich
- Bsp. `Socks`:
  - Socks-Server filtert den TCP/IP Verkehr
  - Alle Verbindungen der Clients müssen über Socks-Server laufen
  - Daher Modifikation der Clients erforderlich (SOCKSifing)
  - Filtert nach: Quelle, Ziel, Art des Verbindungsaufbaus (z.B. Initiierung oder Antwort), Protokoll, Benutzer

Bewertung Verbindungs-Gateway

- Anwendungsunabhängige Filterung
- Ein Proxy für alle Dienste
- Umfangreiche Logging Möglichkeit und damit Accounting
- Zustandsbehaftet
- Benutzerauthentisierung und benutzerabhängige Filterung
- Entkopplung von internem und externem Netz
- Möglichkeit der Erstellung von Nutzungsprofilen

- ✶ Möglichkeit der Erstellung von Nutzungsprofilen
- ✶ I.d.R. keine Filterung nach Dienstprimitiven möglich
- ✶ Sicherheit der Proxy Implementierung??
- ✶ I.d.R. Modifikation der Clients erforderlich

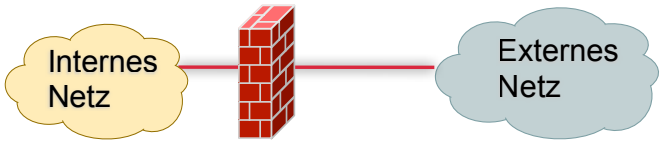
Firewall Architekturen

- Kombinationen von FW Komponenten und deren Anordnung wird als FW Architektur bezeichnet

- Single Box Architektur
  - Screening Router
  - Dual Homed Host
- Screened Host
- Screened Subnet
- Multiple Sceened Subnets

Single Box Architektur

- FW als einziger Übergang ins interne Netz
  - Router (Screening Router) übernimmt FW Funktionalität (i.d.R: Packetfilter)
  - „normaler“ Rechner mit 2 Netzwerk-Interfaces (Dual Homed Host)

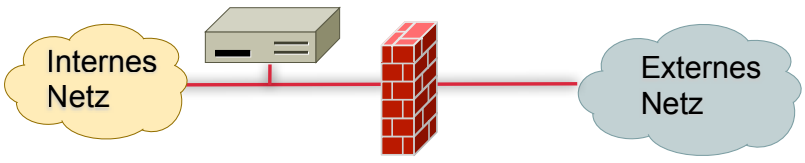


- Billige und einfache Lösung
- Single Point of Administration
- I.d.R. gute Performance (falls nur Packetfilter eingesetzt wird)

- ✶ Wenig flexibel
- ✶ Single Point of Failure

Screened Host

- FW (**Bastion Host**) liegt im internen Netz (nur 1 Interface)
- Verkehr von außen wird über Screening Router (vor-) gefiltert und i.d.R. zum Bastion Host geleitet
- Bastion Host kann Aplication Level Gateway oder Circuit Level Gateway realisieren

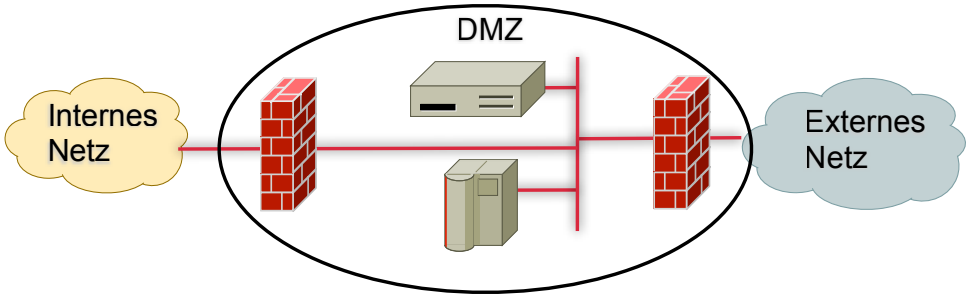


- Trennung von Packet- und Applikationsfilter
- Vorfilterung des externen Verkehrs
- Hohe Flexibilität

- ✶ Packete können immer noch direkt in internes Netz gelangen

Screened Subnet

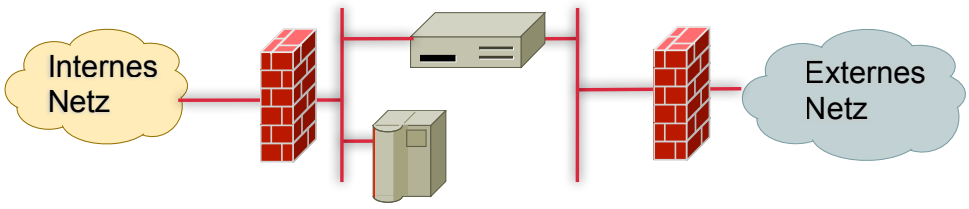
- FW Komponenten liegen in einem eigenen Subnetz (Perimeter Subnet) auch demilitarisierte Zone (DMZ) genannt
- Schutz der DMZ sowohl nach innen als nach außen durch Paketfilter
- Erweiterung der DMZ um dezidierte Server, z.B. WWW



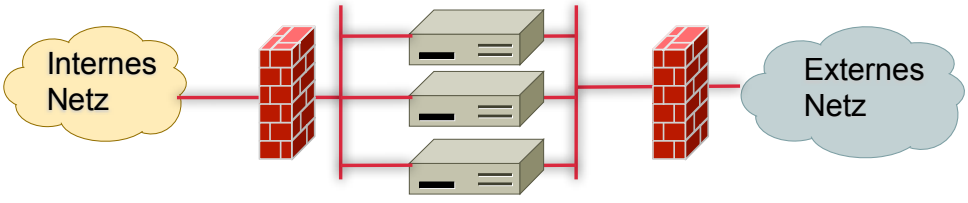
- Keine direkte Verbindung von extern nach intern mehr möglich
- Zusätzlicher Grad an Sicherheit
- Interner Router/FW schützt vor Internet und ggf. vor DMZ

Multiple Screened Subnet

- Verwendung zweier Perimeter Subnets getrennt durch Dual Homed Host



- Verwendung mehrerer Bastion Hosts (Redundanz)



Möglichkeiten und Grenzen von Firewall-Arch.

- Abgestufte Sicherheitskontrollen (vom Einfachen zum Komplexen)
- Möglichkeiten effizienter Protokollierung
- Möglichkeiten der Profilbildung

- Problem der Fehlkonfiguration
- Umfangreiche Kenntnisse erforderlich
- Trügerische Sicherheit
- Erheblicher Administrationsaufwand
- Tunnel-Problematik
  - Anwendungsprotokolle werden z.B. über HTTP getunnelt
  - FW kann dies nicht erkennen

Intrusion Detection Systeme (IDS)

- Intrusion Detection Systeme (IDS)
  - Ziel: „Angriffe“ automatisch erkennen (und ggf. blockieren)
- Intrusion Prevention Systeme (IPS)
  - Ziel: Auf „Angriffe“ reagieren (verhindern / Gegenmaßnahmen ergreifen Gegenangriff durchführen)

- Im folgenden:
  - Host basierte IDS
  - Netz-basierte IDS
  - Angriffs- bzw. Mißbrauchserkennung
  - Beispiele

Host-basierte IDS (HIDS)

- Ausgeführt auf dem zu überwachenden System
  - Systemüberwachung
  - Applikationsüberwachung
  - Integritätsüberwachung (kryptographische Prüfsummen; Hashing)
- \+ Individuell an zu überwachendes System anpassbar
- \+ Sehr spezifische Aussagen über erkannte Angriffe möglich
- \- Benötigt Ressourcen des zu schützenden Systems
- \- HIDS selbst als Angriffsziel
- \- Anpassung an individuelles System erforderlich
- \- Fällt bei erfolgreichem Angriff mit angegriffenem System aus
- Bsp.:
  - Tripwire
  - AIDE (Advanced Intrusion Detection Environment)

Netz-basierte IDS (NIDS)

- Eigener Sensor (kein Gastsystem)
- Überwachen den Netzverkehr (Mithören):
  - Eines Rechners
  - Eines (Sub-) Netzes
  - Einer gesamten Domäne
- \+ Ein Sensor für das gesamte Netz
- \+ NIDS kann „unsichtbar“ installiert werden
- \+ NIDS kann ausfallsicher installiert werden
- \+ Verteilte Angriffe erkennbar
- \- Betrieb am Mirror Port von Netzgeräten; Problem: Packet-Drop
- \- Überlast des gesamten NIDS möglich
- \- Verschlüsselte Daten und Kanäle
- \- „nur“ Netzauffälligkeiten erkennbar

Angriffs- bzw. Mißbrauchserkennung

- Signatur-basiert:
  - Pattern-matching & Expertensysteme
  - „typische Angriffsmuster“
  - \+ Zuverlässigkeit
  - \- Nur bekannte Angriffe erkennbar (keine Zero Day Exploits)
  - Bsp.: Snort
- Anomalie-Erkennung:
  - Lernt „Normalverhalten“
  - Abweichung wird als Angriff gedeuted
  - \+ Flexibel bei neuen Angriffen
  - \- Adaptivität bei Netzänderungen
  - \- False Positives & False Negatives
- Integritätsprüfung
  - Berechnung kryptographischer Hashes
  - Speicherung auf WORM und Vergleich
  - \+ Schnell und Zuverlässig
  - \- Aufwand bei gewollter Datenänderung
  - Bsp.: Tripwire
- Kombination der Verfahren

Bsp.: Tripwire

- www.tripwire.org
- Host-basiertes IDS
- Überwacht anzugebende Dateien und Verzeichnisse
- Erstellt (verschlüsselte) Datenbank mit Prüfsummen:
  - MD5
  - SHA-1
  - HAVAL, ...
- Berechnet regelmäßig Hashes und vergleicht mit gespeicherten
- Probleme:
  - Auswahl schützenswerter Objekte
  - Befugte Änderungen
  - Initialstatus muss sicher sein

Bsp. Snort

- www.snort.org
- Netzbasiertes IDS
- Signatur-basierte Analyse
  - Regeln in Dateien definiert (*.rules)
  - Alle Regeln mit logischem ODER verknüpft
  - Netzverkehr wird gegen diesen Regelsatz geprüft
- Signatur Beispiel:

  alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg: "mountd access";)
  - alert: Alarm generieren; Packet loggen
  - Protokoll Addresse Port -> (Richtungsoperator) Adresse Port
  - content: (Regel Option) Suche nach Muster in der Payload
    - String und/oder
    - Binärdaten; Hex-Notation; gekennzeichnet durch | |
  - msg: Nachricht in Alarm und Log aufnehmen
  - mehrere Regel Optionen mit logischem UND verknüpft