Systemsicherheit

Zusammenfassung

Wintersemester 2010/2011

v1.5

Inhaltsverzeichnis

1 Was bedeutet Systemsicherheit

1.1 Definition

Die Systemsicherheit gibt darüber Aufschluss wie leicht ein System missbraucht werden kann.

Von was hängt sie ab?

- **Kommunikationsysteme:**
  Schwächen von Protokollen, Überlisten von Netz Komponenten.
- **Betriebsysteme:**
  Schwachstellen => Updates fehlen.
- **Server- Dienste:**
  Konfigurationsfehler, Programmierfehler
- **Anwendungen:**
  Konfigurationsfehler, Programmfehler
- **Verhalten der Nutzer:**
  Surfverhalten, schwache Passwörter, Sicherheitseinstellungen des Browsers, E-Mail anhänge**.**

1.2 Personengruppen mit Einfluss auf die Systemsicherheit

1.2.1 Administratoren

Kann in der Regel nur Sicherheits- Patches einspielen und die Komponenten konfigurieren für das er selber Verantwortlich ist.
Beispiele:

- Rechte des Dateisystems
- Webserver Öffentlich (Zugriff auch aus dem Intranet) bedeutet Sicherheitsrisiko
  Server konfiguration (dynamische Dienste: PHP, JSP: Java Server Pages)
  Auswahl der Server (Verbindungsgrad, Open Source)
  Test Server unabhängig von Produktivserver (größere Restriktionen)(jeweils testen)
- Updates /Patches immer auf dem neusten Stand halten.

1.2.2 Verhalten von Programmierern

- Verantwortlich für die Sicherheit ihrer Programme
- Geeignete Vergabe von Zugriffsrechten für das erstellte Programm machen

- Abfangen von ungültigen Eingaben z.B. schutz gegen SQL- Injektion.

1.2.3Anwenderverhalten

- Öffnen von infizierten E-Mail anhängen.
- Herunterladen von infizierten ausführbaren Dateien
- Schutz für den Anwender
    - Antivirus
    - Browsercheck, E-Mail check (Beachten Dateianhänge .txt)
    - Krypto Kamapagne (Verschlüsseln und Signieren von Mails & Dokumenten)
    - Netzwerkcheck (überprüfen auf offene Ports, Schwachstellen von Webservern, Host mit “Host Based Intrusion Detection System” konfigurieren, Scan Tools, Clickjacking (durchsichtige Buttons) = Darstellung einer Website wird maipuliert über einen Button wird ein transparenter gelegt => um Formdaten an Angreifer zu senden.
    - Tools
    - Update- Chek

2 Grundlagen

2.1Klassifikation der Angreifer

2.1.1White Hat

Sucht nach Schwachstellen in einem System und meldet sie dem Betreuer und benutzt sein Wissen nicht für Kriminelle Tätigkeiten.

2.1.2Hacker

Computer Experte: Perfekte Systemspezialisten
realisieren häufig Open Source Projekte
Missbrauchen ihre Kenntnisse nicht
Folgen einer Hacker Ethik

2.1.3Cracker

Legal: Sport: Cracken von Schutzmechanismen
(*CrackMe: Eine Software bei der z.B. das Passwort, oder Serienummer gefunden werden soll.)*

Illegal: Das gleiche kann auch für illegale zwecke eingesetzt werden.

2.1.4Phreaker

(Phone & Freak) Manipuliert Telefonnetze z.B. für kostenlose nutzung von Telefonleitungen, Telefonkonferenzen schalten, Rückverfolgung erschweren druch spezielle Signaltöne.

2.1.5Spammer

Sendet unerwünschte Nachrichten => Spam oder Junk

2.1.6Phisher

Mit gefälschten Webseiten will man Daten von Anwendern sammeln/ergreifen.

2.1.7Script Kiddy

Die Hacker Scene Kopieren, Kaum Programmierkenntnisse, Meißt pupertierende Jungs, Manipulieren, Zerstören aus Langeweile ohne Hirn und Verstand.

2.2 Computerkriminalität

- Straftaten bei denen ein Computer als Tatmittel eingesetzt wird.
- Oder selbst Gegenstand der strafbaren Handlung ist
- **Strafbereiche:** Wirtschaftsdelikte
  Datenschutzdelikte
  Sexual Delikte
- **Straftaten:** Computer betrug
  Computer Sabotage
  Softwarepiraterie

2.2.1 Strafbestände

2.2.1.1 Ausspähen nach §202a StGB

- Unbefugt sich oder einen andern Zugang zu Daten zu verschaffen die nicht für ihn bestimmt sind, begeht eine Straftat => bis zu 3 Jahren Haft

2.2.1.2 Abfangen von Daten - §202b StGB (Wireshark)

- Ist das Bondant zu Straftaten (SIP, VOIP) ,Abhören von Telefongesprächen.
- Gilt nur für Daten die sich in der Übermittlung befinden.

2.2.1.3 Vorbereiten des Ausspähens und Abfangen von Daten §263a StGB

- Sonderform des Betrugs
- Beeinflussung des Ergenisses eines Datenverarbeitungsvorgans, durch:
  unrichtige Gestaltung eine Programms
  Verwendung von unvollständigen oder Falschen Daten
  Unbefugte Verwendung der Daten
  unbefugtes Einwirken auf den Ablauf
- Absichtlich sich oder eines Drittten einen rechtswiedrigen Vermögensvorteil zu beschaffen.

2.2.1.4 Datenveränderung §303a StGB Abs3

- Wer rechtswiedrig Daten (§202a Abs.2 StGB)
  Löscht; unbrauchbar macht; unterdrückt: oder verändert
  => Bis zu 2 Jahren Haft oder Geldstrafe
- Der Tatbestand kann durch folgende Handlungen erfüllt werden:
  Löschen, Unterdrücken (“Spams nicht weiterleiten”), Unbrauchbar machen, Verändern
- Vorbereitung der Datenveränderung ist ebenso Starfbar

2.2.1.5 Computersabotage §303b StGB

Störung von Datenvereinbarung die für andere wesentlich von Bedeutung ist
=> 3 Jahre Haft oder Geldstrafe

2.3 Angriffsaktionen

2.3.1 Einschleusen von

Viren: Werden durch ein infiziertes Programm verbreitet, Ausbreitung ist Passiv.

Trojaner: Nützliche Anwendung die im Hintergrund weitere Aktionen durchführt um z.B Daten zu erspähen.

2.3.2 Denial of Service

Lahmlegen von Diensten: SYN- Flooding oder Smurf- Attacke (große Anzahl von Pongs)

2.3.3 Verteilen von Bots

Botnet: Reihe von “übernommen” Rechnern (Bot) wird über das Internet Ferngesteuert, z.B. für Spam oder DoS- Angriffen.

2.3.4 System-Penetrations-Test

Versuch in Rechner einzudringen
=> Angriff auf RechnerNetz im Auftrag des Betreibers.

2.3.4.1 Reconnaissance (Auskundschaften)

Internetauftritt => who is => Daten zum Nameserver, responsible Person etc.
(Who is = Social Engineering)
=> Informationen über das Netz herausfinden
Google => Index of /etc/
DNS =>AXFR Zonentransfer => IP- Adresse der Rechner

2.3.4.2 Enumeration (Finden von Angriffsmöglichkeiten)

Portscan
Verwundbare Versionen von Distributionen/Betriebsystemen
Konfigurationsfehler finden
Installierte SOftware auf Fehler überprüfen
SNMP ausnutzen => Std. Community Strings benutzen (pubic(lesen) : private (schreiben))

2.3.4.3 Exploitation (Sicherheitslücke ausnützen)

Was kann durch Ausnutzung der Sicherheitslücke erreicht werde => z.B. root Rechte erhalten

2.3.5 Sniffing

Sniffer untersuchen und analysieren den Netzverkehr, um Schwachstellen und Sicherheitslücken zu erkennen.
=> Gut für Administratoren
=> Gut für den Angreifer

2.3.6Aktive Sniffer

Senden- Probe- Request- Pakete an den AccessPoint, und bekommen Probe- Response- Pakete zurück.

2.3.7Passiv Sniffer

z.B. Kismet

- Sendet *keine Pakete,* Monitored nur welche Pakete ankommen.
- Passiv scanner können nicht ausgemacht, gefunden, erkannt werden
- Passiv Scanner erkennen auch exotische Wlans, die nicht auf Probe- Requests Antworten, oder andere Protokolle verwenden (Straßenbahn, oder ESSID verborgen etc.)

2.3.8Footprinting

1 Phase eine Angriffs zur Informationsbeschaffung ohne direkten Zugriff auf das Zielsystem.

Informationen über/aus: who is (Datenbank Denic.de)
DNS- Einträge (“nslookup”)
Firmen Webseiten
Zonentransfer

2.3.9Firewalking

Pfad durch Firewall finden, dazu bedient man sich bekannten und unbekanten Fehlkonfiguration. Suche nach offenen Port. => “Firewalk” - entdeckt offen Ports
Firewall- Ports, Fehlkonfiguration, Mapping des Zielnetzes

2.3.10 Session Highjacking (Übernahme von Sitzungen)

1. Passives Sniffing & erforerliche Informationen Sammeln, dann
2. Übernahme von der TCP- Sitzung um die Rechte eines Benutzers zu bekommen.

2.3.11 Übernahme von Web- Sitzungen

Http ist Zustandslos
Verwaltung von Benutzereingaben bei Webformularen werden z.B. über Cookies (Session Cookies) gespeichert => Entführung von Session Cookies

2.3.12 IDS (Intrusion Detection Systeme)

System das Angriffe erkennen. Reagieren auf Angriffe nur mit generierung von Events, also nur Meldungen.
Benutzt auf einem switch, z.B. Snort (Tool for IDS).

2.3.13 IPS (Intrusion Prevention Systeme)

Reagieren und blockieren z.B. (auf) die Pakete des Angreifers. Also IDSysteme die nicht nur Angriffe erkennen und Melden, sondern auch gegenmaßnahmen einleiten.
Liegt im Datenstrom und erkennt und reagiert Angriffe, erweiterung von IDS.

2.4Zusätzliche Begriffe (SS2010)

2.4.1Incident Response

Reaktion auf ein Vorfall/ Notfallplan

2.4.2Open Source Tools

- Ettercap - Man in the Middle Angriffe, Sniffed auf IP u ARP Basis
- NETCORtools (TCP Trace basierend)
- Tcpdump
- Wireshark (früher Ethereal)
- NetworkMiner

2.4.3Ettercap

Tool für Man- in- the- Middle- Angriff

2.4.4Promiscous Mode:

NIC verwerfen in der Regel die Datenpakete die nicht an Sie gerichtet oder an eine Broadcast- Multicastadresse gesendet worden ist. Bei diesem Modus verarbeitet die NIC jegliche Pakete die die NIC durch das Medium “mithören” kann.

2.4.5Website defacament

Ein Angreifer ändert Webseite eines Dritten
Zugriff erreicht er durch z.B.
Ausnutzen von Schwachstellen:
des Betriebsystems, des Webservers, Webskripte, Konfigurationsfehler des Webservers.

2.4.6Schutzmaßnahmen (BSI Grundschutz Handbuch)

- Server in abgeschlossene Räume
- Security updates, Patches einspielen
- Web Application Firewall
- Regelmäßiges Testen des Servers auf Schwachstellen (Schwachstellentest => Nexus)
- Testskripte auf dem Server deaktivieren

2.4.7Exploiting the DNS Server of an Organization

DNS- Spoofing: DNS- Name zeigt auf falsche IP-Adresse:

2.4.8Sabotage

Löschen und verändern von Daten
SQL- Injection

2.5Häufige Ursachen für Sicherheitslücken

- Veraltete Software, insbesondere Software ohne automatische Onlineupdates
- Keine Updates mehr für die Software vom Hersteller/ Programmierer
- Schwaches Passwort
- Unsichere Konfiguration
- Administratoren wird oft nicht genug Zeit gelassen um alles sicher zu Konfigurieren.
- ?Nur an den “Außenrändern” (Gateways ins Internet) des Netzes Firewalls, IDS, IPS, etc?
- Jeder aktivierte Dienst birgt Sicherheitsrisiko => JEDEN unnötigen Dienst deaktivieren. (z.B Chargen, echo, fingon etc)
- Windows Freigabe im Intranet nur mit READ und EXECUTE Rechten
- Unsichere Wlans

3 Sicherheit von Netzkomponenten

3.1Arbeitsweise von Hubs

Was ein Hub an einem seiner Interfaces empfängt, leitet er an alle anderen Interfaces weiter. Der Datenrahmen wird also an die gesamte Kollisiondomäne verteilt.
Switches und Router begrenzen eine Kollisiondomäne, die durch Hubs gebildet werden. In dieser Domäne kann von jedem Netzteilnehmer der gesamte Datenverkehr abgehört werden.
Senden 2 PCs (Netzkomponenten) gleichzeitig Daten einen angeschlossenen Hub, so kommt es im Hub zu einer internen Kollision.
Der Hub sendet in diesem Fall JAM-Datensendungen an alle seine Interfaces, Dieses JAM-Signal leitet der Hub an alle seine Interfaces mit Ausnahme des Empfangsinterfaces weiter. Die JAM Signale kollidieren mit den Sendungen von den PCs. Jeder PC erkennt die Kollision.

3.2Hubs und Netzwerksicherheit

Hubs senden Datenpackete, die sie auf irgendeinam ihrere Interfaces empfangen, an alle anderen Interfaces weiter. Daher kann ein rechner, der an einem Hub, angeschlossen ist, den Datenverkehr von allen Systemen, die zu einer Kollisionsdomöne gehören, abhören. Zum Abhören werden Snifferprogramme wie z.B. **WireShark** verwendet.

3.3ARP-Spoofing in einer HUB-Umgebung

Beim Spoofing (Manipulation, Verschleierung) gibt ein Rechner vor ein anderer Rechner zu sein. Beim ARP-Spoofing gibt sich der angreifende Rechner die MAC-Adress eines anderen oder eines nicht vorhanden Rechners.
Unter WINDOWS XP kann man seine MAC-Adresse unter Eigenschaften von LAN-Verbindung anzeigen lassen.

3.4Arbeitsweise von Switches

Wurde eingesetzt um die Bandbreite zu erhöhen. Ein weiterer Effekt bei der Verwendung von Switches ist, dass der Datenverkehr, der an einem Switch angeschlossenen Komponenten, nicht mehr so einfach abgehört werden kann. Beim Einschalten der Switches sind die Porttabellen (Switching-Tables) leer. Switches lernen selber welche Netzkomponenten an ihre Interfaces angeschlossen sind. Sie identifizieren Rechner und Router anhand der Ethernetadressen dieser Komponenten. Die Adressen werden erst in die Porttabelle eingetragen wenn mind. 1 Packet verschickt wurde. Source-Address wird eingetragen

Sendet **PC1** nun einen Datenrahmen an **PC2**, so trägt er in den Enthernetdatenrahmen die folgenden Ethernetadressen ein:

Destination-Address: Ethernetadresse von **PC2**
Source-Address: Ethernetadresse von **PC1**

Switch empfängt Datenrahmen über sein Interface und wertet ihn aus. Da seine Porttabellen noch leer sind (wurde gerade eingeschaltet) , trägt er die Adresse von PC1 in seine Porttabelle ein.

Weiterleiten eines Datenrahmens an alle anderen Interfaces bei unbekannter Zieladresse nennt sich **flooding**.
Der erste Datenrahmen kann abgehört werden.
Wenn ein Daterahmen von einem Switch gezielt weitergleitet wird, bezeichnet man dieses Verhalten als **forwarding**.

Damit PC1 dem Rechner PC2 einen IP-Datenrahmen senden kann, benötigt er zusätzlich zur IP-Adresse von PC5 diessen MAC-Adresse. Diese erfragt er mit Hilfe eines ARP-Request-Datenrahmenes.

DST-Address: MAC-Broadcast : FF-FF-FF-FF-FF-FF
SRC-Address: MAC-Adresse von PC1
Type: 0x806: ARP

ARP-Request werden an die MAC-Broadcast-Adresse gesendet. Daterahmen, die an die Broadcast-Adresse gesendet werden, werden von einem Switch immer geflutet, d.h. diese Datenrahmen können an jedem Switch-Interface abgehört werden.

3.5Arbeitsweise von Routern

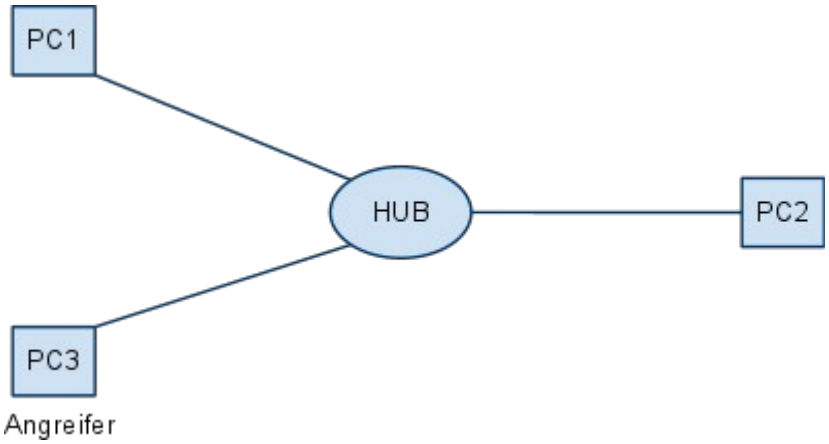
Hubs arbeiten in Schicht 1 des OSI-Modells. Sie werten die Datenrahmen, die sie weiterleiten nicht aus.
Switches verwenden Schicht 2 (MAC-Adresse, Ethernetadressen) um ihre Switching-Entscheidungen zu treffen.
Router arbeiten in der Schicht 3 des OSI-Modells. Sie werten daher die Adressen aus, die in dieser Schricht definiert sind. Bei Verwendung von IP sind es die IP-Adressen

Switches und Hubs sind für die Rechner transparent. D.h er merkt nichts von dem Vorhandensein dieser Komponenten im Netz.
Router sind nicht transparent, sie müssen explizit mit dem Weiterleiten von Datenpacketen beauftragt werden.
- Standard-Gateway ist meistens der Router.
- Router leiten MAC-Broadcasts niemals weiter. Daraus folgt, dass ARP-Requests am nächsten Router enden. Jedoch leitet der Router Pakete die ihn „beauftragen“ weiter wenn er eine Route zur Zielnetzadresse in seiner Routing Table hat. Datenverkehr, der sich “hinter” einem Router abspielt, kann nicht abgehört werden.

Beispiel ARP: 2 Host in versch. Netzen/Subnetzen, verbunden durch 1 Router
Alle Arp- Tables sind leer, da frisch gebootet ;-)
HostA will Paket an HostB schicken, kennt nur Ipv4 Adresse. HostA schaut zuerst in seiner Routingtable nach einem Match und findet nur z.B. das Default Gateway als Match => ARP mit Mac- Broadcast (Layer2) macht keinen Sinn da Mac's nur für ein lokales Subnetz/VLAN/Segment als Paket „zustellung“ gedacht sind.
Da aber auch die Mac- Adresse des Routers nicht in seinem Arp-Table vorhanden ist, wird ein ARP- Request auf die IP des Routers geschickt, dieser Antwortet mit einem ARP- Response und damit mit seiner Interface Mac- Adresse. HostA kann nun Router mit routing von Paket für HostB „beauftragen“ indem er die Mac- Adr. des Routers benutzt und die IP des HostB's, sein Ziel.

Router Conf: „proxa arp is enabled“:
Bedeutet der Router antwortet auf ARP- Requests für Rechner die in einem anderen Subnetz/VLAN/Netz liegen. Der Client wird so „vorgegaukelt“ sein Ziel liege im selben Segment.
Auch Beudeutet diese Funktion das keine ICMP redirects auf diesem Port möglich sind und Std.mäßig disabled!

3.6Spoof-Szenario



- PC2 erzeugt einen ARP-Request: PC2 sucht die MAC-Adresse von PC1
- PC3 generiert dazu einen gefälschten ARP-Response auf einem ARP-Request von PC2 mit folgendem Inhalt:

  DST-ether: 00:00:00:00:00:0b (PC2)
  SRC-ether: 00:00:00:00:00:0d (gibt es in diesem Netz nicht)
  sender-HW-Address: 00:00:00:00:00:0d (im Netz unbekannt)
  sender-ip: 192.168.0.1 (PC1)
  target-HW-Address: 00:00:00:00:00:0b (PC2)
  target-ip: 192.168.0.2 (PC2)

- 3.) PC2 trägt darauf hin in seinen ARP-Cache die falsche Zuordnung IP-Adresse zu MAC-Adresse ein:

PC1 sendet aufgrund dieses Eintarges in Folge Datenpakete für die IP Adresse 192.168.0.1 an die MAC-Adresse 00:00:00:00:00:0d. PC1 empfängt zwar diesen Datenrahmen. Seine Netzwerkkarte verwirft jedoch dieses Datenpaket, weil die MAC-DST-Adresse nicht mit der eigenen MAC-Adresse übereinstimmt.

PC3 (Angreifer) hat seine Netzwerkarte in den promiscuous-Mode gesetzt (damit empfängt die Karten den gesamten Datenverkehr). Er empfängt also auch den Datenverkehr, der an die Adresse 00:00:00:00:00:0d gesendet wird. Er kann diesen Datenverkehr abhören, verändern und an das eigentliche Ziel (PC 1) weiterleiten.
Wie kann der gefälschte ARP-Response dem PC2 untergeschoben werden?
PC3 bekommt den gesamten Datenverkehr mit, damit auch den ARP-Request von PC2. Er muss auf diesen schneller Antworten als PC1 und gegebenfalls PC1 durch eine DoS Attacke lahmlegen.

3.7ARP-Spoofing Variante 1

Der Angreifer gibt sich die MAC-Adresse des anzugreifenden Rechners. Der Datenverkehr kommt daraufhin bei ihm an. Diesen Verkehr hört er ab. Danach leitet er den Datenverkehr an den eigentlichen Bestimmungsort weiter.

3.8ARP-Spoofing Variante 2

Bei dieser Variante wird die MAC-Adresse des Angriffsopers mit einer nicht existierenden MAC-Adresse gespoofed. Ein Paket, welches an das Opfer gesendet wird, kommt bei allen Systemen im geswitchten Netzsegment an, da ein Switch eine unbekannte Adresse niemals lernen kann. Damit ist diese Adresse auch keinem Switch bekannt. Das Packet wird von allen Nettwerkkarten mit Ausnahme der Netzwerkkarte des Angreifers (diese befinden sich im Promiscuous-Mode) verworfen. Der Angreifer modifiziert das Packet und sendet es an den eigentliche Ziel weiter.

3.9ArpWatch

ArpWatch ist ein Tool, welches die ARP-Pakete überwacht. Auf diese Weise können neu angeschlossene Rechner erkannt werden und es können ebenfalls Situationen erkannt werden, in denen IP-Adressen doppelt vergeben werden. Diese Ergebnisse werden dem zuständigen Administrator automatisch per E-Mail mitgeteilt, so dass dieser zeitnah eingereifen kann.

3.9.1ArpWatch in einem HUB-Netz

In einer Netzwerkumgebung, die aus Hubs besteht, kann ArpWatch alle ARP-Requests und alle ARP-Responses aufzeichnen. Um alle Responses aufzeichnen zu können, setzt ArpWatch die Netzwerkkarte des Rechners, auf dem ArpWatch arbeitet, in den Promiscuous-Modus.

3.9.2ArpWatch in einem Switch-Netz

Überwacht man die ARP-Requests mit ArpWatch in einer geswitchten Umgebung, so kann man folgende ARP-Datenrahmen aufzeichnen:ARP-Requests, weil diese an die MAC-Broadcast-Adresse gesendet werden. Durch die Auswertung der ARP-Protokollfelder (SENDER HW-ADDRESS, SENDER-IP) ArpWatch zählt nicht zur vollwertigen IDS Programmen. ArpWatch ist in der Lage Angriffe, die durch eine Manipulation des ARP Protokolls durchgeführt werden. Zu solchen Angriffen zählt das ARP-Spoofing. Das Tool wertet den gesamten Broadcast-Verkehr aus und schreibt alle MAC und die dazugehörige IP-Adressen in eine Textdatei. Wenn ein neues System oder eine neue MAC-IP Paarung entdeckt wird, löst ArpWatch einen Alarm aus und protokolliert die Änderungen im Syslog

3.9.3Gratuitous ARP

“unaufgeforderte ARP” bezeichnet eine spezielle Verwendung von ARP. Dabei wird von einem Host ein ARP-Anforderungs-Broadcast gesendet, bei der er seine eigene IP-Adresse als Quell- und Ziel-Adresse einträgt. Damit teilt er seine ggf. neue MAC-Adresse unaufgefordert mit. Das kann mehreren Zwecken dienen:

1. Normalerweise darf keine Antwort kommen, denn eine IP-Adresse muss in einem Netz eindeutig sein. Bekommt er trotzdem eine Antwort

   => ein Host ist nicht richtig konfiguriert.

2. Nützlich wenn z.B. Netzwerkkarten ausgetauscht werden und die anderen Hosts über die neue MAC-Adresse informiert werden sollen.Gratuitous ARP geschieht normalerweiße beim Booten eines Computers.

3. 2 Server teilen sich eine IP (Server und Ersatzserver), fällt ein Server aus muss dem Netzwerk die neue MAC-/IP Adress-Zuordnung bekannt gemacht werden. Sonnst bekommt niemand den Wechsel mit.

4. **I**n einem Mobile IP-Szenario sendet der Home Agent einen Gratuitous ARP, wenn sich der Mobile Host aus dem Heimatnetz entfernt, um die Pakete stellvertretend für diesen zu empfangen.

3.9.3.1 Example Traffic:

Ethernet II- Frame:

Dst. Mac:	FF:FF:FF:FF:FF:FF
Src. Mac:	HostMac (Vom Sender des Grat.ARP)
Type:	0x0806 => ARP

IP- Frame:

Protocol Type:	0x0800 => IP
HW- size:	6
Protocol- size:	4 => ARP
Sender Mac:	HostMac
Sender IP:	HostIP (Vom Sender des Grat.ARP)
Target Mac:	flooded BC
Target IP:	HostIP (Vom Sender des Grat.ARP)

4 Schwächen von Protokollen ausnutzen

4.1Angriffe über ICMP

4.1.1ICMP Redirection

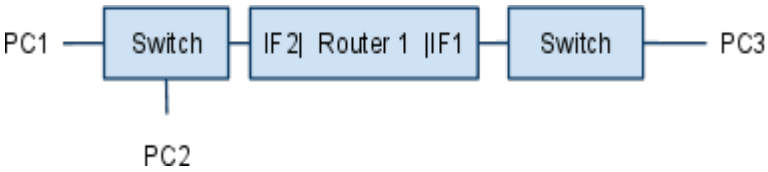
Diese Nachricht wird gesendet wenn der Router eine kürzere Route zum Ziel kennt.
Router erkennt Redirection *nur wenn das Weiterleitungspaket über das Empfangsinterface wieder verschickt werde muss.*
Aufgabe ICMP (Seite 43 - Schwächen von Protokollen Ausnützen) = redir.pkt

4.1.2ICMP Destination Unreachable

Type 3: Code *

- Code 0 - net unreachable Zielnetzwerk nicht erreichbar weil Interface z.B. down.
- Code 1 - host unreachable Wenn Zielhost (z.B. PC) nicht auf eine *ARP-Request* reagiert vom Router reagiert.
- Code 3 - port unreachable Dienst nicht verfügbar, “läuft nicht auf Server”

Angriff:



IF1 von Router fällt aus, Router sendet *host unreachable*.
Angriff (DoS) PC2 sendet *NET/Host/Port unreachable* an PC1, PC2 sendet an PC1 TTL exceeded (DoS)

4.1.3ICMP Router Discovery

4.1.3.1ICMP Router Advertisement Msg

Type 9, Code 0
Router werden Prioritätsgewichtet in der Message gelistet mit *Router Adress + Preference Level* (wird in Metrik umgerechnet vom Client).
Lifetime field: Gibt Dauer des Eintrags in der Routingtabelle an.
Advertisement Messages werden von Router nur verrschickt wenn sie “*Router Soliciation”* anktiviert haben.
Angriff
Rechner im selben Subnetz kündigt sich als besten Router an.

4.1.3.2 ICMP Router Soliciting Msg

Type 10, Code 0
Nachricht fordert Advertisement Message an, üblicherweise verschickt von einem Client nach boot.

4.2 TCP/IP Verbindungen Manipulieren

-> Will man eine TCP Verbindung übernehmen, muss man die richtige Sequenznummer verwenden!

4.2.1 Regulärer TCP Protokollablauf - 3 Way Handshake

1. Client (SYN sent) ------> SYN = 1, ACK#ungültig, SEG#(beliebig gewählt)
2. Server (SYN/ACK sent) <------ SYN = 1, ACK = 1, SEG#(belibig gewählt),
   Daten0 ACK = Client SEG# + 1
3. Client (ACK sent) ------> ACK = 1, SEG# +1
   Daten0 ACK = Server SEG# + 1

ACK# ist nur ungültig oder Wert 0 beim ersten Paket eines Verbindungsaufbaus, sonst immer 1!
Windows Size: *Gibt anzahl Bytes an die in pipelining gesendet werden können ohne auf ACK zu warten.*
pipelining: *Senden mehrerer Pakete ohne Quittierung abwarten zu müssen.*
Data Offset: Länge des TCP- Headers in 32Bit Blöcken - OHNE NUTZDATEN (Payload). => Zeigt Startadresse von Payload Daten an.

4.2.2 Regulärer TCP Datenaustausch

Kommunikationspartnern quittieren jeweils die SEG# des Partners durch *inkrementieren* dieser mit *Data Offset.*
Beispiele Client:

- Aus Server Paket: SEQ#542, Data 3 => ACK#545
- Aus Server Paket: SEQ#545, Data 28 => ACK#573

4.2.3 Regulärer TCP Protokollablauf - 4 Way Handshake (Verbindungsabbau)

Beide Verbindungsrichtungen müssen über ein FIN = 1 , ACK = 1 und der dazugehörigen quittierten Antwort abbgebaut werden.

4.3Gestörte TCP - Protokollabläufe

4.3.1TCP - Hijacking

Entführen einer TCP Verbindung.

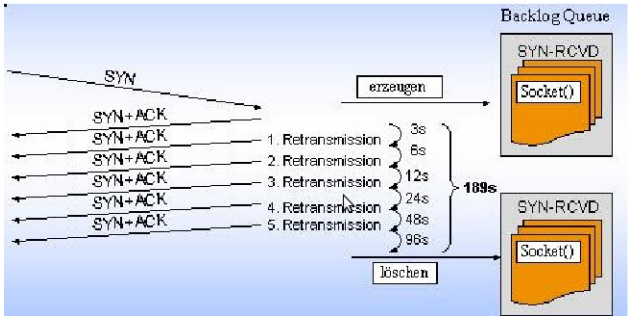
4.3.2TCP ACK- Storm

Es ensteht eine Schleife die erst endet wenn eines der Pakete verloren geht.

1. Angreifer sendet in bestehende TCP Verbindung eine *falsche SEG# an Server (injected paket).*
2. Server sendet an Client *ACK für das injected paket.*
3. Client sendet *letzte gültige ACK#*

4.3.3SYN- Flooding (Angriff)

Kann verwendet werden um (Distributed) DoS (Denial of Service) Attacken durchzuführen. Für *DDoS- Attacken verwenden Bot- Rechner* für ihre Angriffe.
Beim SYN- Flooding versucht der Angreifer die *Systemresourcen des Servers zu blockieren.*
Dafür sendet er *wiederholte TCP Verbindungsaufbau Pakete (SYN = 1), unterschlägt aber das 3.Paket eines 3 Way Handshakes*. Server belegt so wiederholt Systemresourcen/Datenstrukturen, da angenommen wird das 2.Paket wäre verloren gegangen. Jede Verbindungsanfrage wird im Server beim dazugehörigen Dienst in die *Backlog Queue* abgelegt. Ist dieser Puffer voll kann keine weiter Verbindung zu diesem Dienst mehr aufgenommen werden. Ein Eintrag in der Queue halten sich i.d.r. mehrere Minuten. Zwischen den Retransmission des 2.Pakets vergehen überlicherweise 3 Sekunden, dies wird auf einem Linux System 5 mal wiederholt. Ein Std. Linux- System hat Platz für 256 Backlog- Queue Einträge.Angreifer hat genügend Zeit (Heise Security).



In der Regel verwendet der Angreifer gefälschte Absenderadressen, da die Antworten des Servers nicht interressieren. Durch geschicktes verteilen der Adressen kann das Filtern der Angriffspakete erschwert werden. Wenn möglich werden Adressen benutzt die nicht belegt sind, da es sonst vorkommen kann das ein Rechner fälschlicherweise das SYN/ACK Paket des Server erhält und mit einen RESET den Verbindungseintrag, auf dem Server, löschen läßt. Dadurch würde der Server nicht die volle Warteprozedur durchlaufen (Summe: 189sekunden).

4.3.3.1 Gegenmaßnahme: SYN- Cookies (Informationen in SEQ# ablegen)

Server senden die Antworten zur Verbindung als Cookie an den Client zurück und speichern die Information nicht in einem Logfile. Die TCP- Sequenznummer dient als Träger für die Cookies.

Der Cookie enthält: Einen erstellten MD5- Hashwert aus folgenen Werten:
Src. Port, Dst. Port, zugehörigen IPs, ein Geheimnis.

Kommt das 3.Paket eines Verbindungsaufbaus - ein ACK- Paket vom Client zurück, so enthält es das zuvor vom Server gesendete Cookie in der Acknowledgement- Nummer. Server bildet erneut Hashwert aus empfangenem Paket und vergleicht diesen mit dem Wert der ACK- Nummer. Falls es stimmt stellt der Server die Verbindung her.

4.3.3.2 WIKI

Das Transmission Control Protocol (TCP) macht keine Vorgaben zum initialen Wert der Sequenznummer der SYN/ACK-Pakete. Also kann der Server sie nutzen, um Informationen zu kodieren, die er sonst in einer Tabelle halboffener TCP-Verbindungen speichern müsste. Da es keine solche Tabelle gibt, kann sie auch nicht überlaufen, womit ein SYN-Flood-Angriff nicht zu einem Denial of Service (DOS) führen kann.
Da der Client die Sequenznummer des TCP SYN/ACK um 1 hochgezählt, enthält sein TCP ACK-Paket die vom Server generierte initiale Sequenznummer um 1 inkrementiert. Der Server dekrementiert diese also wieder um 1 und vergleicht sie anschließend mit dem Hashwert des Pakets. Stimmen die beiden Hashes nicht überein, muss die Verbindung neu aufgebaut werden, wozu der Server dem Client ein TCP RST-Paket sendet.
Weil die Überprüfung des Verbindungsaufbaus auf dem Server passiert, kann die Hashfunktion der Implementierung grundsätzlich beliebig definiert sein; sie sollte jedoch möglichst zufällig sein, um Sicherheitsrisiken zu vermeiden. Dieses Verfahren läuft für den Client transparent ab, weswegen Verbindungen zwischen Gegenstellen unabhängig davon aufgebaut werden können, ob sie SYN-Cookies verwenden.

4.3.3.3 Gegenmaßnahme: TCP- Parameter ändern

Möglich ist auch die größe der Backlog- Queue zu erhöhen. Außerdem kann man die Anzahl der Retransmissions herabsetzen. Bei Std. Linux ergeben diese Timeouts ca. 3Minuten.

4.4Angriffe auf IP

4.4.1IP Fragmentierung

Verwendet ein Router wenn ein Datenpaket über eine Leitung verschickt werden muss das ein
MTU Wert hat < 1500.

4.4.1.1Beispiel

472 IP- Bytes (452 IP Payload, 20 Ip Header). Nutzbytes werden durchnummeriert von 0 bis 451. Der Fragment Offset gibt den durch 8 dividierten Offset des 1. Byte des Fragments, in Bezug auf das nicht fragmentierte Datenpaket an.
IP Header Feld: Flg=1 zeigt an es gibt weitere Fragmente
Flg=0 ziegt an das es das letzte oder einzige Fragment ist.
Fragment Offset: IP Daten Offset / 8

4.4.1.2Angriff

Angreifer versucht 1 Datenpaket einer Verbindung an der Firewall vorbei zu schleusen, da Firewalls standardmäßig Verbindungen aus dem Intranet ins Internet beliebig gestatten. Da ein Verbindungsaufbau immer am gesetzten SYN FLag erkannt werden kann, fragmentiert der Angreifer das Verbindungsaufbau Pakete so klein das das SYN Flag (TCP Flags) nicht erkannt wird (Version 1).
Version 2 des Angriffstyp überschneiden sich die Fragmente im TCP Flag bereich. Fragment 1 enthält ACK=1, Fragment 2 ACK=0. Das 2 Fragment überschneidet das 1Fragment.
Fragmente werden erst beim Empfänger zusammengbaut.

4.4.1.3Beispiel

Fragment 1: Fragment Offset= 0; Fragment 2: Fragment Offset=1 (8 Byte)

4.4.1.4Overlapping Fragments

TCP Header in 2 Fragmente aufgeteilt:
1 Fragment: Fragment Offset = 0 , ACK = 1 , SYN = 0
2 Fragment: Fragment Offset = 1 , ACK = 0 . SYN = 0
Firewall läßt beide Fragmente durch, da alle Fragmente gleichbehandelt werden und die Flags des 1 Fragments werde durch das des 2. überschrieben.
Firewalls Defragmentieren!

4.4.2IP Spoofing

Angreifer verwendet die IP- Adresse eines anderen Rechners. Ziel ist entweder Herkunft des Angriffs zu verschleiern, oder durch fremde IP Zugriffsschranken auf Zielrechner zu umgehen.
Linux TCP Wrapper, Windows IIS (Internet Informationen Server) verwenden z.B. Access Lists!
(basieren auf IP- Adressen)

4.4.2.1 Blind Spoofing

Antwortpakete erreichen aufgrund des Routings nicht Angreifer. Angreifer gibt die IP-Adresse des Rechners an als Source IP der auf dem Opfer Zugriff hat.

- Verwendung von IP- Adressen für Zugriffskontrolle ist nicht sicher (Spoofing gefahr)!
  => Daher wird SSH verwendet
- SNMPv1; SNMPv2 verwendet noch IP Zugriffskontrollen. (SNMPv3 = SSH)
- Im Netzwerkbereich, Intranet, nur Zugriff auf Kompontenten über Management VLAN1.

Hinweis: *Sequnez number attacks sind unwahrscheinlicher geworden weil OS- Hersteler die Verfahren geändert haben mit denen Sequenze numbers erzeugt werden. Die überholte orgehensweise addiert einen konstanten Wert, um die nächste Sequenz number einer neuen Verbindung zu erhalten.*

Wndows: Addiert Sequence Number um 1 oder 2
Linux: Wählt einen Zufallswert.

4.4.2.2 Bewertung von IP Spoofing Attacken

- Source Routing und maipulieren von Routingtabellen => Schwer für Angriffe
- Blind Spoofing benutzt Zugriffskontrollen die auf IP's basieren als Sicherheitslücke. Dies waren die meisten R- Dienste: rsh, rlogin, rcp, usw. (Zumeißt von SSH abgelöst)

4.4.3 Schutzmassnahmen basierend auf IP-Adressen

4.4.3.1 Xinetd/inetd

Schonen Ressourcen von Servern, da sie nur Dienste starten, dynamische, wenn sie gebraucht werden also wenn ein Verbindungswunsch vorliegt. Xinetd/ Inetd nimmt die die Verbindungswünsche entgegen.

4.4.3.2 Xinetd erweitert Inetd um folgendes

- Access Control
- Erweitertes Logging
- Zeitbeschränkung für Dienste
- Dienste auf bestimmte Interfaces binden.
- Verwendung von RAM, CPU Zeit beschränkbar.

4.4.4TCP- Wrapper (tcpd)

Schützt Dienste vor unbefugten Zugriffen durch IP Access List (bestimmte IP's oder Adressbereiche) oder schreibt nur Logeinträge bei Verwendung von Diensten.
Dienste die vom TCP- Wrapper Deamon (tcpd) geschützt werden sollen müssen mit den IP- Adresse oder Adressen in der Datei /etc/hosts.allow und /etc/hosts.deny definiert werden.
Existieren die beiden Dateien nicht werden sie om Deamon als *leer interpretiert*.
Mit *tcpdmatch* können die Access Control Regeln *überprüft* werden,
z.B. /etc/tpcdmatch in.telnetd 141.69.1.1

4.4.4.1Beispiel

(**Format:** deamon_list : client_list [: shell_command] # A Deamon liste kann aus Prozessnamen oder Protnummer oder Wildcards bestehen)

hosts.allow
#
ftpd,telnetd: 141.69.200.2
ALL: 141.69.100.
ALL: 127.0.0.1

4.4.4.2Access Control

Beschränken Zugriff auf Dienste vom Server durch IP- Adressen.
Unix Systeme haben für die Zugriffskontrolle über Access Control 2 Möglichkeiten:
1. Über den TCP- Wrapper,2. über Xinetd/Inetd.
Cisco Router besitzen hierfür die Access Control Lists, um Dienst zugriffe zu defnieren.

4.4.5Verwendung von Source Route Option

Loose Source and Record Route: Type 131
ping -j 1.IP 2.IP ... Destination (max 9 IP's)
Strict Source and Record Route: Type 137
Die erste IP- Adresse muss local erreichbar sein.
ping -k 1.IP 2.IP ... Destination (max 9 IP's)

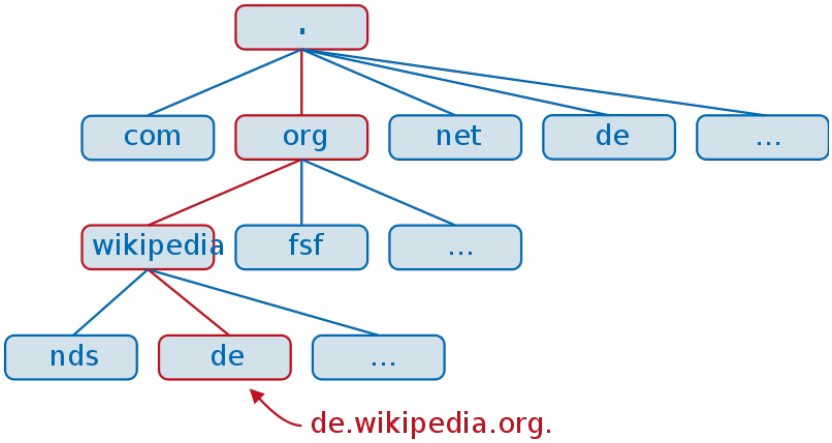
Hinweis: *Die meisten Firewalls verwerfen IP- Pakete die aus dem Internet, die als IP-Quelladresse eine Internetadresse verwenden, um IP Spoofing zumindest aus dem Internet zu verhindern.*

4.4.6Routingtabellen manipulieren

Um die Antwortpakete zu erhalten, kann ein Angreifer versuchen durch gefälschte Pakete eine Routingprotokolls die Routingtabllen eine Routers zu manipulieren.

4.4.7DNS

Ist ein verteilter hierarchischer Verzeichnisdienst, der in Zonen unterteilt ist. Hauptsächlich wird DNS dazu verwendet DomanNamen in IP- Adressen umsetzen (“forward lookup”), IP- Adresse zu DomainName (“reverse lookup”). Eine DomainName darf maximal 255 Zeichen lang sein mit alleń Punkten, die die Zonen voneinander trennen.



Fully Qualified Domain- Name (FQDN): www.hs-weingarten.de.
DomainNames werden von rechts nach link delegiert, d.h. der Punkt ganz rechts trennt “root” von der 1 Ebene = “Top Level Domain (TLD)”.
Die DNS- Objekte, wie z.B. Rechnernamen, werden als Satz con “Resource Records” in einer Zonendatei gehalten.Für jede Zone ist mindestens ein Nameserver zuständig, dessen Informationen über die Zonen gelten als gesichert auch als “autoritativer Nameserver” bezeichnet. Weil die Administration der Zonen delegiert wird.
Ein “nicht autoritativer Nameserver” bezieht seine Informationen über andere Zonen durch “Nachfragen” bei anderen Nameservern die nicht nur Anfagen zu “Ihren” Zonen zulassen, dies wird als “nicht authorisierte Antwort” bezeichnet.

4.4.8Zusammenarbeit der Nameserver

4.4.8.1Delegierung

Teile des Namens werden meißtens ins *Subdomains* unterteilt, die mit eigens zuständigen Nameservern verwaltet werden. Ein Nameserver einer Domäne kennt die zuständigen Nameserver seiner Subdomains, an die die Anfrage(“Query”) delegiert wird.

4.4.8.2Weiterleitung(forwarding)

Falls der angefragte Namensraum außerhalb der eigenen Domäne liegt, wird die Anfrage an einen fest konfigurierten Nameserver weitergeleitet.

4.4.8.3 Auslösung über Root- Server

Falls kein Weiterleitungsserver konfiguriert wurde oder dieser nicht antwortet, werden die Root-Server befragt. Die Adressen dieser sind in einer statischen Datei hinterlegt. Diese beantworten dann die Anfrage ausschließlich *“iterativ”* (Server antwortet mit einem Verweis auf anderen Nameserver)

4.4.9 Resolver

Sind Software Module die Clients ermöglichen informationen von Nameservern anzuzeigen und abzufragen. Dieser ergänzt die Query zu einem FQDN und fragt beim fest zugeordneten Nameserver nach. Sie arbeiten entweder *iterativ oder recursiv.*

4.4.9.1 rekursive Query

Nameserver fragt selbständige weiter/zuständige Nameserver nach der gesuchten Domäne, falls sie nicht in seinem Cache liegt und er nicht für sie zuständig ist.

4.4.9.2 iterative Query

Resolver bekommt entweder *Resource Record* als Antwort oder den *Verweis* auf den nächsten Nameserver.
Client Resolver arbeiten üblicherweise ausschließlich rekursiv, daher werden sie auch **“stub- Resolver”** genannt.

DNS verwendet i.d.r UDP als Protokoll, aber auch TCP kann verwendet werden wenn Extended DNS verwendet wird. Die maximale zulässige größe eine DNS- UDP Paket beträgt 512 Bytes.
Für Anfragen(Query’s) und für Zonentransfer wird der Port 53 verwendet.
Programme: nslookup, dig, host

4.4.10 Zonentransfer

AXFR = Asynchronous Full Transfer Zone (Engl. asynchronous Xfer full range) übertragung von Zonen auf andere Nameserver. AXFR überträgt alle Einträge, die etwas neuere Spezifikation IXFR überträgt ausschließlich die geänderten Einträge

4.4.11 Resource Records (Zonendatei)

- Mit dem SOA Resource Record werden Parameter der Zone, wie z. B. Gültigkeitsdauer oder Seriennummer, festgelegt.
- Mit dem NS Resource Record werden die Verknüpfungen (*Delegierungen*) der Server untereinander realisiert.
- Mit folgenden Record-Typen werden die eigentlichen Daten definiert:
    - Ein A Resource Record weist einem Namen eine IPv4-Adresse zu.
    - Ein AAAA Resource Record weist einem Namen eine IPv6-Adresse zu.
    - Ein CNAME Resource Record verweist von einem Namen auf einen anderen Namen.
    - Ein MX Resource Record weist einem Namen einen Mailserver zu, er stellt aus historischen Gründen eine Besonderheit dar, da er sich auf einen speziellen Dienst im Internet, nämlich die E-Mailzustellung mittels SMTP bezieht. Alle anderen Dienste nutzen *CNAME*, *A* und *AAAA Resource Records* für die Namensauflösung.
    - Ein PTR Resource Record weist einer IP-Adresse einen Namen zu (*Reverse Lookup*) und wird für IPv4 und IPv6 gleichermaßen benutzt, nur für IPv4 unterhalb der Domain „*IN-ADDR.ARPA.*“ und für IPv6 unterhalb von „*IP6.ARPA.*“.

(Mit freundlicher Unterstützung von deiner favorisierten Wikipedia Seite)

```
$ORIGIN example.com.     ; designates the start of this zone file in the name space
$TTL 1h                  ; default expiration time of all resource records without their own TTL value
example.com.  IN  SOA  ns.example.com. username.example.com. (
              2007120710 ; serial number of this zone file
              1d         ; slave refresh (1 day)
              2h         ; slave retry time in case of a problem (2 hours)
              4w         ; slave expiration time (4 weeks)
              1h         ; minimum caching time in case of failed lookups (1 hour)
              )
example.com.  NS    ns                    ; ns.example.com is a nameserver for example.com
example.com.  NS    ns.somewhere.example.     ; ns.somewhere.example is a backup nameserver for example.com
example.com.  MX    10 mail.example.com.  ; mail.example.com is the mailserver for example.com
@             MX    20 mail2.example.com. ; equivalent to above line, "@" represents zone origin
@             MX    50 mail3              ; equivalent to above line, but using a relative host name
example.com.  A     10.0.0.1              ; IPv4 address for example.com
              AAAA  2001:db8:10::1        ; IPv6 address for example.com
ns            A     10.0.0.2              ; IPv4 address for ns.example.com
              AAAA  2001:db8:10::2        ; IPv6 address for ns.example.com
www           CNAME example.com.          ; www.example.com is an alias for example.com
wwwtest       CNAME www                   ; wwwtest.example.com is another alias for www.example.com
mail          A     10.0.0.3              ; IPv4 address for mail.example.com,
                                          ;  any MX record host must be an address record
                                          ; as explained in RFC 2181 (section 10.3)
mail2         A     10.0.0.4              ; IPv4 address for mail2.example.com
mail3         A     10.0.0.5              ; IPv4 address for mail3.example.com
```

4.4.12 Reguläre DNS- Anfragen

Ein *Domain Name Server (DNS)* erhält eine Anfrage von einem *Name Resolver Process,* ober der *gesuchte Name in der Domain liegt,* für die er zuständig ist.

- Falls Ja, Name wird in Adresse übersetzt und zurückgeschickt an Resolver.
- Falls Nein, Welche Art der Interaktion wurde angefragt ?
    - Vollständige Übersetzung (recursive resolution):
      Der angefragte Nameserver nimmt Verbindung auf mit "richtigen" Nameserver auf und "Fragt weiter".
    - (nonrecursive resolution):
      DNS kann den Namen nicht übersetzen, schickt Fehlermeldung zurück.

DNS Server (geschlossene !) beantwortet Anfragen aus:

- **Internet** Nur Fragen zur EIGENEN DOMAINE. Auch keine Reverse Anfragen!
- **Intranet** JEDE Anfrage!

Konfigurationsdateien enthalten die Zuordnung der eigenen reservierten Domaine und zum eigenen IP Adressbereich, und umgekehrt.
Beispiel: hs-weingarten 69.141.in-addr.arpa CLASS B: 141.69.0.0/16

4.4.13 DNS- Anfrage/ Antwort über UDP

Damit eine Antwort Aktzeptiert wird muss gelten:
==> ID Passt; IP's Passen; Ports stimmenDNS ID- Fälschung
Die ID Kennzeichnet eine Anfrage eindeutig, dieses Feld wird auch bei einer "Vollständigen Überetzung" vom DNS verwendet, zur Identifizierung zischen den Domain Name Server.

4.4.14 Ziel eines DNS- Angriffs

Im Cache eines DNS- Servers soll zu einem DomainName eine falsche IP- Adresse abgelegt sein (Phishing). Jeder Eintrag bleibt für die Dauer des TTL Eintrags im Server erhalten, danach wird er gelöscht. Solange liefert der DNS die gefälschte IP an anfragende Resolver aus.

Angreifer muss ID raten, oder hochzählend erraten, oder viele Antworten mit variierten ID's senden.

4.4.15 Ablauf eines DNS- Angriffs (Cache Poisoning)

Der Opfer DomainName Server wird nach einer Domaine,z.B. www.fhwgt.de, gefragt:

- Ist die Adresse im Cache, wird sie an den Resolver ausgeliefert.
- Ist die Adresse nicht im Cache, muss der Nameserver den zuständigen DNS der Domäne fhwgt.de nach dem Eintrag fragen.
  ==> Der Angreifer sniffed die übertragene ID der rekursiven Anfrage und versucht dem DNS seine gefälschte Antwort als gültige Antwort unterzuschieben.

War sind ID,IP's und Ports richtig Aktzeptiert der Nameserver die gefälschte Antwort und trägt die falsche IP in seinen Cache ein => "Der DNS Cacher wurde vergiftet".

4.4.16 Zeitbedarf bei klassischem DNS- Cache poisoning

Die Wahrscheinlichkekit für eine rictig geratene ID beträgt:

$$P_{CS} = 1 - (1 - \frac{D * R * W}{N * P * I})^{(\frac{T}{TTL})}$$

I: Number Distinct ID's available (max 65536)
P: Number of Ports used (max. around 64000, mostly 1 ;-))
N: Number of authoritative nameservers for a domain
R: Number of packets sent per second by attacker (z.B. 10Mbit/s)
W: Windows opertunity, in sec. Bounded by the respoonse time of the authoritative servers (often 0.1s).
D: Average nuber of identical outstanding questions of a resolver (typically 1)
T: Time period in which the attacks occurs.
TTL: Time To Live of the legitimate resource recor. (Std. 604800s = 7 Tage)

$$Framerate\ in\ s = \frac{1}{iFg + \frac{(FS + 64)}{NDR}}$$

IFG: Interframe Gap, min 9,6 µs
FS: Framsize
NDR: Datenübertragunsrate

4.4.17 DNS Einträge fälschen in

/etc/hosts | C:\WINDOWS\system32\drivers\etc\hosts

Die Einträge in dieser Datei haben Vorrang gegenüber dem DNS Cache!
Standard Eintrag ist: 127.0.0.1 localhost

4.5SNMP

Simple Network Management Protocol wird für das Verwalten von Netzkomponenten verwendet.
Der Zugriff kann über Access Lists (IP's) u VLAN's beschränkt werden. Als Passwörter für die Komponenten werden Community Strings verwendet, es gibt 2 Arten
(1 read | 1 write) => Default Community strings (Pwd): (public | private)
Zugriffe am besten nur über Management VLAN zulassen.

4.6 FTP

Verwendet einen Commando Port 21 und einen Daten Port 20

4.6.1active FTP

Der FTP- Server baut die Datenverbindung auf. Über den Cmd Port (21) teilt der Client mit, auf welchem Port er Verbindungswünsche für Daten entgegennimmt.
Das bedeutet nach erfolgreicher Verbdindung bestehen 2 Ports:

1. Commando Kanal: Zwischen FTP Client, zufälliger PORT und FTP- Server, well known PORT 21.
2. Datenkanal: Zwischen FTP Client, bestimmter PORT und FTP- Server, selbstgewählter PORT.

Wireshark:
=> "FTP Request: PORT 192,168,0,108,19,137"
==> FTP Datenport vom Client (IP:192.168.0.108) Port: 19 * 256 + 137 = 5001

Nachteil:
Eingehende Verbindungen aus dem Internet muss Zuggelassen werden.

4.6.2passive FTP

Der FTP Client baut die Datenverbindung auf. Über den Cmd- Port (21) teilt der Server mit auf welchem Port er Verbindungwünsche für die Daten entgegenimmt.
Vorteil:
Client baut die Verbindung aus dem Intranet ins Internet auf, keine Anpassung an der Firewall nötig.

4.6.3FTP- Bounce Back

Ziel des Angreifers ist es, den FTP Server als "Portscan" auf den Opfer- HTTP- Server umzulenken. Dies kann er nur bei *active FTP*, da hier der client den Port dem Server übermittelt und durch gefälscht IP auch das Opfer definiert.
Beispiel:

1. HTTP GET Anfrage als Datei auf dem FTP- Server ablegen.
2. Cmd Port (21) auf Webserver zeigen lassen.
3. Datei an Webserver senden.

5 Denial-of-Service bei Netzwerken

Diese Angriffe führen dazu, dass ein Dienst oder mehrere Dienste nicht mehr verfügbar sind. Diese Standardvorgehensweise eines Angreifers überlastet den anzugreifenden Rechner.

Werden mehrere Rechner eingesetzt, um einen Angriff auf ein Opfer simultan durchzuführen, so spricht man von einem verteilten DoS-Angriff (Distributed Denial-of-Service, kurz DDos). Dazu werden häufig Bot-Netze eingesetzt.

5.1 Klassifikation von Denial-of-Service Angriffen:

5.1.1 Networkdevices

- HW Ressourcen erschöpfen
- viele Pakete senden, die Logs erzeugen

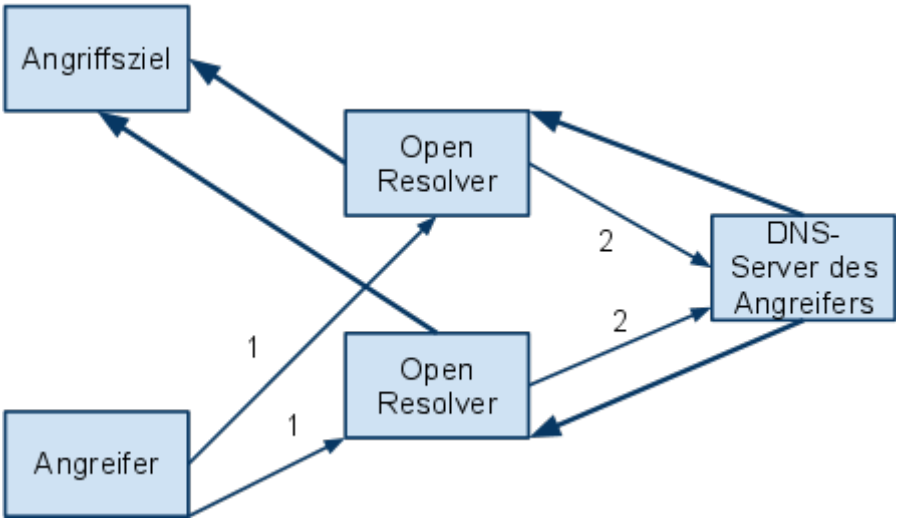
5.1.2 OS

- Schwächen der Implementierung des Protokollstack ausnutzen
- Ping of Death [Paket > 65535]
- Teardrop [Fragmentierung] (Fragment im Fragment)

5.2DNS DDos Amplification Attacks

Verwendet zum Angriff

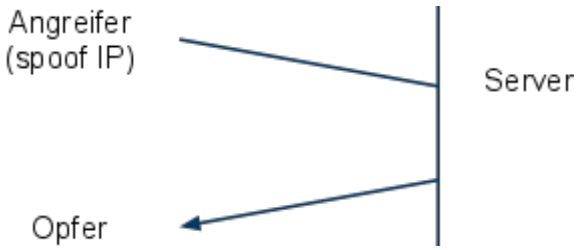
- die Vergrößerung der ursprüngliche gesendeten Datenmenge (amplification)
- Fälschung der Absenderadresse
- Open DNS Resolver
- UDP Eigenschaften (keine Flußkontrolle)
- Maximale Größe von DNS Antwort Paketen wurde von 512 Byte hochgesetzt auf 4000 Byte.
  Anfrage: 60 Byte
  Antwort: 4000 Byte - NAPTR (RFC 2918)



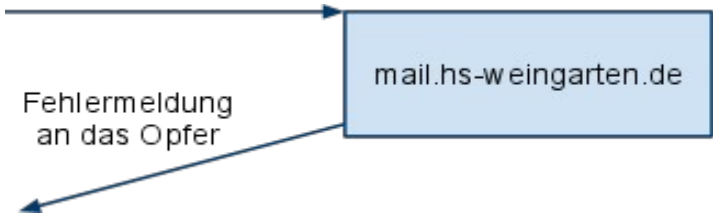
Anfrage an den Open Resolver:
Gib mir einen großen Record für IP des Angriffsziels
2) rekursive Anfrage für den großen Record (60 Byte)
3) Antworten auf die rekursive Anfrage (4000 Byte)
4) Open Resolver senden “große” Antworten an das Angriffsziel

5.3Distributed Reflected Denial of Service

Eine DRDos-Attacke ist eine weitere Vorgehensweise dieser Angriffsart
Angreifer sendet Datenpaket nicht an Opfer, sondern an einen Internetdienst: SRC-Adress: OPFER IP. Ist der Angreifer z.B. nicht berechtigt, sendet der Internetdienst die Fehlernachricht an das Opfer.



Im folgenden Beispiel müsste man die E-Mail Absenderadresse der ursprünglichen E-Mail auf die des Angriffsopfers fälschen, um die Fehlermeldung auf das Opfer zu lenken. Der vermeintliche Angreifer ist in diesem Fall der Mailer-Daemon.
Mail an: gibtesnicht@hs-weingarten.de
Absender: armesau@sau.de



5.3.1Blacklisting

Jeder Mailserver, der Unzustellbarkeits Nachrichten (bounces) verschickt, läuft Gefahr, auf schwarze Listen (RBL), zu landen.
E-Mail-Blacklist + Liste von E-Mail Servern, die bei der E-Mail Kommunikation negativ aufgefallen sind.
E-Mail Server fragen u.U. die Listen ab. Sie verweigern die Annahme von E-Mail, die von Server aus der Liste gesendet werden. Kein offenes E-Mail-Relay betreiben.
ACHTUNG BEI BOUNCES.
Als **Joe-Job** bezeichnet man E-Mails mit gefälschtem Absender
Absender zeigt auf E-Mail-Adresse der Person die geschädigt werden soll
Inhalt: SPAM und Hetzschriften
Nicht so wichtig:
Namensgebend war Joe Doll. Der Ami wurde im März 1997 Opfer einer derartigen Rufschädigungskampagne. Aufgrund der großen Menge an Beschwerden, Unzustellbarkeitsnachrichten und anderen Angriffen war das System von Joe Doll für zehn Tage nicht erreichbar.

5.4SYN Flooding

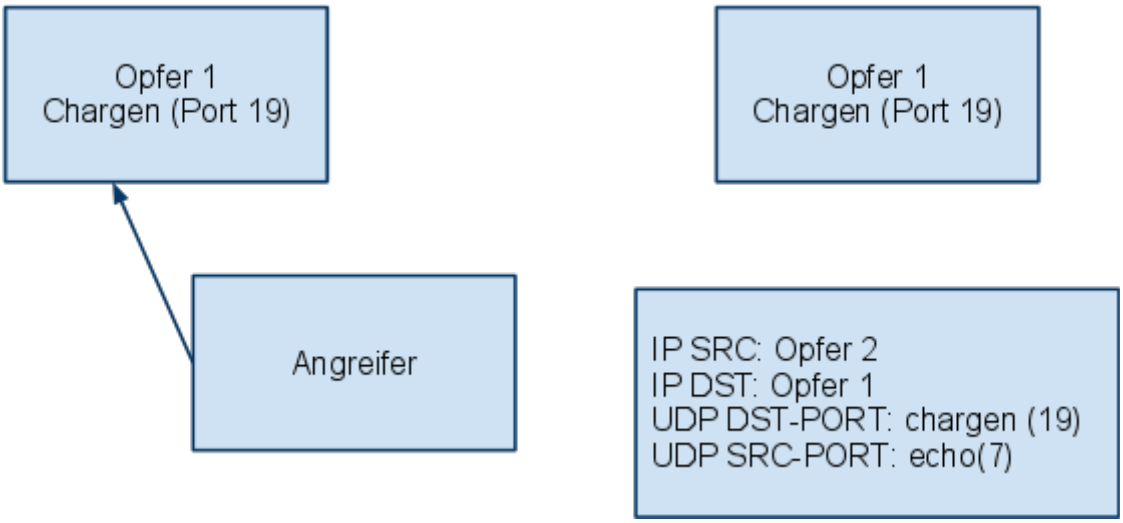
Eine Beschreibung erfolgte bereits im Kapitel “gestörte TCP-Protokollabläufe” ;)

5.5UDP Flood Attack

Dieser Dos-Angriff verwendet UDP. UDP verwendet anders als TCP keine Flusskontrolle. Bei TCP kann ein Sender, der zu viele Datenpakete pro Zeiteinheit sendet, angehalten werden (Window-Mechanismus). Ein UDP-Sender kann hingegen nicht gebremst werden. Daher kann ein UDP-Angriff sehr viel Bandbreite belegen.

Eine Vorehensweise bei dieser Attacke ist, dass der Angreifer
zufällige Zielportnummer verwender bei Packeten, die er an das Opfer sendet.
OPFER
- /etc/services nachschlagen (welche Anwendung gehört zu diesem Port ?)
- falls Dienst nicht verfügbar: ICMP Fehlernachricht

Ein bekanntes Beispiel einer UDP Flood Attack verwendet den Chargen Dienst. Der Angreifer sender ein UDP-Paket zum Port 19 (chargen) eines seiner Opfer. Er fälscht die IP-Adresse und gibt die Adresse eines zweiten Opfers an. Als Absenderport gibt er 7 (echo) an. Die beiden Maschinen legen sich gegenseitig lahm.



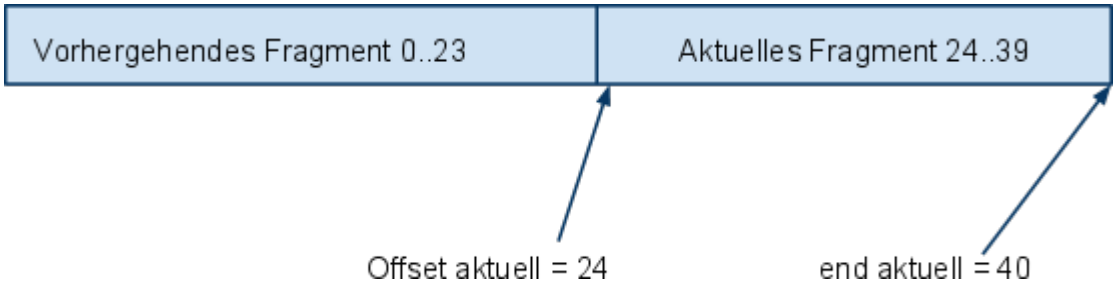
5.6Packet Fragments

Diese Dos-Angriffe nutzen Schwächen bei einigen TCP-Stacks, die das Wiederzusammensetzen von IP-Fragmenten betreffen.

5.6.1Teardrop (Fragment liegt vollständig im vorausgehende Fragment)

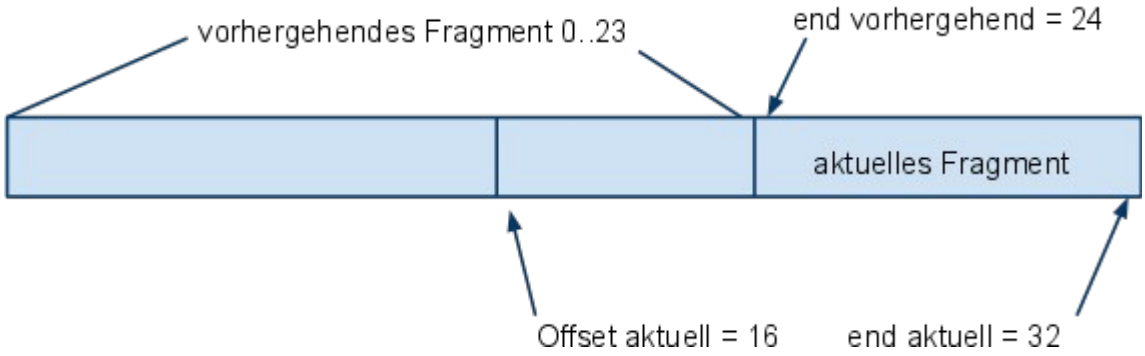
Bei der Teardrop Attacke werden falsche Fragment Offset Werte benutzt, um ein Fragment vollständig in das vorhergehende Fragment einzubetten. Im folgenden wird betrachtet, wie für das aktuelle empfangene Fragment der Speicherbedarf ermittelt wird:

5.6.1.1Normalfall



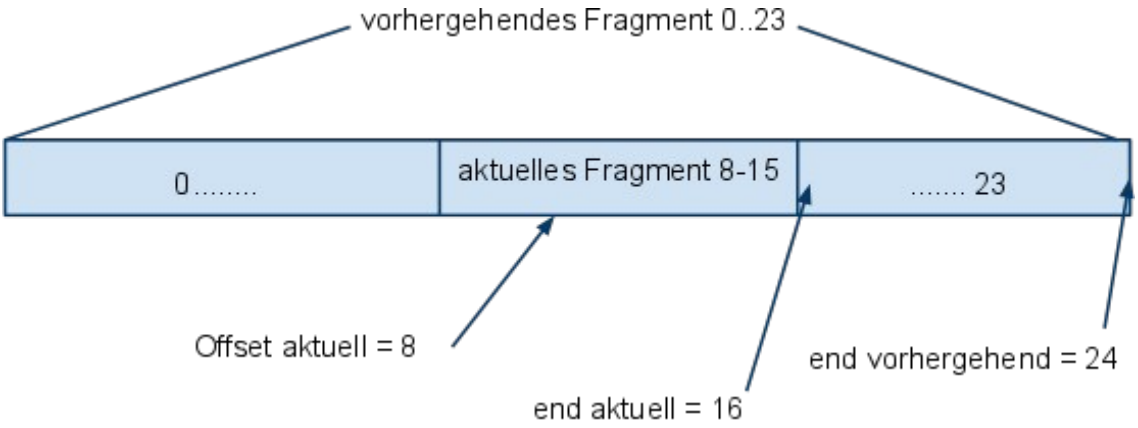
Der Offset des nächsten Bytes des Fragments wird ermittelt aus:
end(aktuell) = Total_Lenght(aktuell) - Header(aktuell) + Fragment Offset(aktuell) * 8
= 36 - 20 + 3*8 = 40
Daraus ergibt sich der für das aktuelle Fragment zu reservierende Speicherbedarf zu:
Speicherbedarf = end(aktuell) - Offset(aktuell) = 40 - 24 = 16

5.6.1.2Überlappende Fragmente



Der Offset des nächsten Byte des nächsten Fragments wird ermittelt aus:
end(aktuell) = Total_Lenght(IP) - Header(IP) + Fragment_Offset(IP) * 8
= 36 - 20 +2*8 = 32
Daraus gibt sich der für das aktuelle Fragment zu reservierende Speicherbedarf zu:
Speicherbedarf = end(aktuell) + end(vorhergehend) = 32 - 24 = 8

5.6.1.3Teardrop



Der Offset des letzten Bytes des aktuelle Fragments wird ermittelt aus:
end(aktuell) = Total_Length(IP) - Header(IP) + Fragment_Offset(IP) * 8
= 28 - 20 + 1*8 = 16
Daraus ergibt sich der für das Fragment zu reservierende Speicherbedarf zu:
Speicherbedarf = end(aktuell) - end(vorhergehend) = 16 - 24 = **-8**

Es ergibt sich ein negativer Wert für den Speicherbedarf! Konsequenz siehe folgenden Beitrag:
Seite 96 GANZ VIEL CODE!

5.6.2Bonk

Bonk sendet falsche Fragmentinformationen an sein Opfer
Bonk ist eine Variante von dem älteren Teardrop. Der Teardrop Exploit arbeitete auf vielen verschiedenen Systemen -> Ein Patch wurde veröffentlich welche die Systeme immun gegen diese Attacke machte. Bonk arbeitet speziell in dem “loophole in Microsofts Teardrop patch”.
Windows 95 und Windows NT sind davon betroffen.

5.7Land

Einige Fehler in der Implementierungen von TCP/IP können dazu führen das dass System Ausgelastet wird oder sogar abstürzt (Sogar 2005 noch Win XP SP2, Win 2003). Dies konnte bei diesem DoS Angriff passieren.(ein SYN Paket, bei dem die SRC-Adresse und der Port die gleichen sind wie bei der DST-Adresse, i-e spoofed)
Dieses Paket wird an einen offenen Port des Opfers geschickt, das Opfer Antwortet dann mit einem SYN/ACK Packet auf den eigenen Port und erzeugt selbst die Belastung.
SYN-Paket
SRC Adress = DST Adress
SRC Port = DST Port

5.8ICMP Angriffe

5.8.1Ping of Death

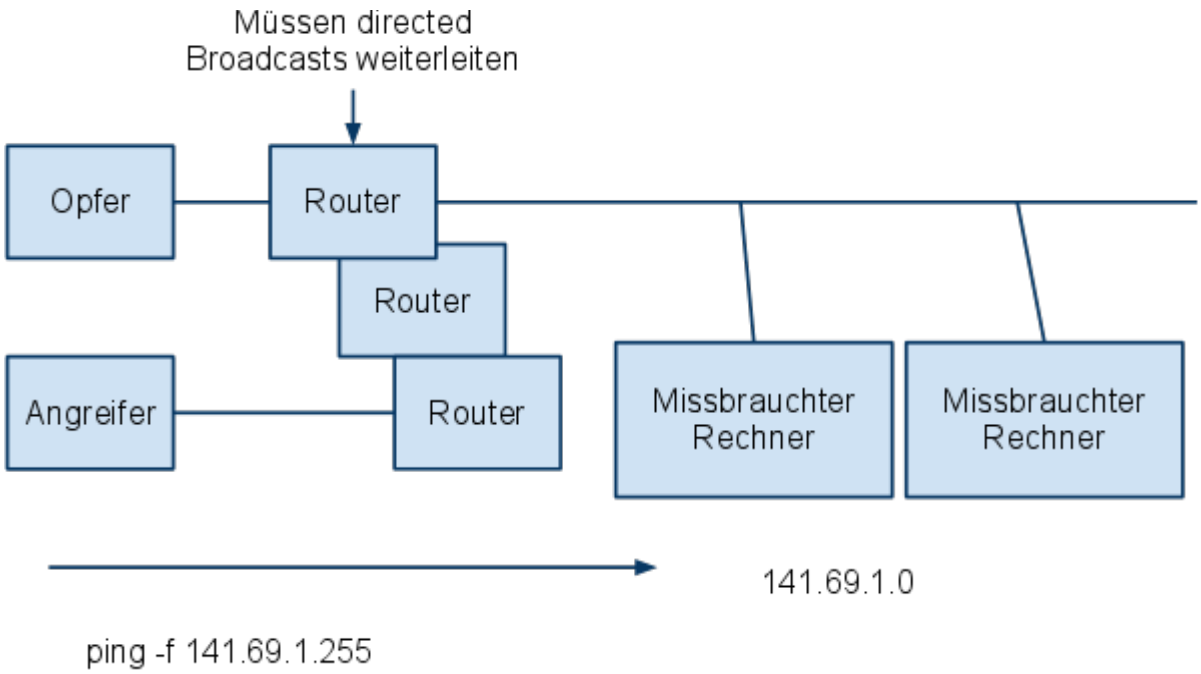
ICMP Paket > 65535
Bufferoverflow, weil nach RFC 791 (IP) IP Pakete nicht größer sein dürfen

5.8.2Ping Flood

Linux ping - f Zieladresse
Opfer wird von einer reihe von ICMP Pakete beschossen.
(Trabbi): Bei Flood- Ping sendet von einem oder mehreren Rechnern Ping Pakete an das Opfer, welches dann seine komplette Bandbreite mit Echo- Replys und eingehenden Echos verstopft. Voraussetzung dafür ist eine größere Bandbreite des Angreifer.

5.8.3Smurfing

Beim Smurfing werden ICMP-Echo-Pakete an eine **directed IP-Broadcast-Adresse** gesendet.
ICMP-Echo-Pakete an directed Broadcast-Adresse
Absende Adresse gefälscht auf die IP des Opfers.



5.8.4Fraggle

Fraggle ist eine Abwandlung von Smurf. **Fraggle verwendet UPD - Pakete**, die an die Dienste echo(Port7), chargen(port19), daytime(13) und qotd(17) der missbrauchten Rechner gerichtet werden. Benötigt aber ebenso ein direct Broadcasting forwarding.

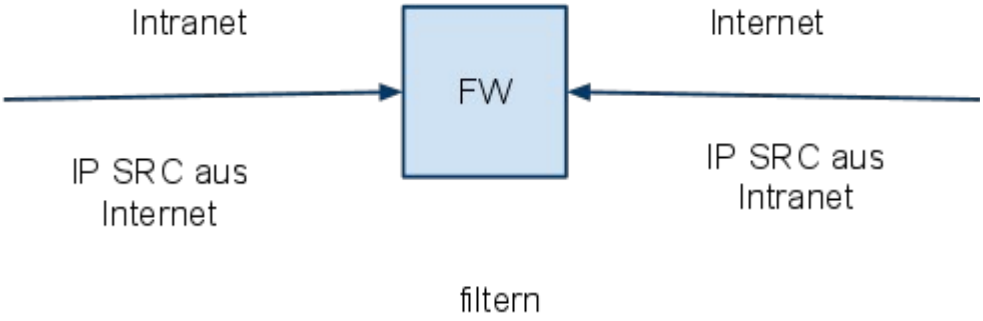
5.9WinNuke -> fertiges Exploid

Der Begriff WinNuke bezeichnet ein über Netzwerk ferngesteuerte DoS-Attacke, gegen die folgende Windoof OS anfällig sind / waren:
Windoof 95, NT, 3.1.x
Das Senden eines TCP Paketes mit gesetztem URG Flag auf den TCP - Port #139
(NetBIOS) hat einen Bluescreen zur Folge oder verursacht einen Neustart des Rechners.
ScriptKiddies haben diesen fertigen Exploid gerne benutzt.

5.10 Maßnahmen gegen Dos-Angriffe

5.10.1 Network device level

- patches and upgrades
- Packet filtering



OS level
- patches and upgrades
- protocol implementation changes
-> SYN Cookies
no smurf: Router leiten keine directed Broacasts mehr weiter.

5.10.2 OS-Level: Lazy receiver processing (LRP)

Standard Vorgehensweise beim IP-Paketempfang

- Der Empfang eines Datenpaketes löst ein HW-Interrupt aus
  - der Interrupt Handler legt das Paket in die IP-Queue ab und löst ein SW Interrupt aus.

- Innerhalb des SW Interrupt Handlers laufen die folgenden Aktionen im IP-Modul ab:
  - Eventuelle de-fragmentierung
  - UDP oder TCP Eingangsfunktionen aufrufen
  - Paket in der Socket-Queue der betreffenden Applikation ablegen.
- Immer wenn die jeweilige Applikation einen receive system call aufruft, werden die Daten in den Adressbereich der Applikation kopiert und der Speicher in der Socket Queue wird freigegeben

Probleme bei dieser Vorgehensweise

- Das Verwerfen der Pakete infolge einer Überlastung des Empfängers findet erst statt, nachdem schon Ressource für die Bearbeitung des verworfenen Paketes verbraucht wurden.
- Der Datenverkehr einer Verbindung kann andere Verbindungen beeinträchtigen (ein geteilter Puffer)
- Die CPU-Zeit, die in einer Interrupt Service Routine verbraucht wird, wird beim Prozess zugerechnet, der unterbrochen wurde (schlechtes Scheduling)

Vorgehensweise beim IP-Paketempfang mit Hilfe des LRP

Jede Applikation erhält eine eigene Paket-Queue
Die Netzwerkkarte (NI) legt Daten in der dem betreffenden Socket zugeordneten Warteschlange ab.
Pakete, die in einer vollen Queue abgelegt werden sollen, werden verworfen
Die Bearbeitung der Protokolle erfolgt im Kontext des jeweiligen Benutzerprozesses der den “receive system call” ausführt. Damit wird die CPU-Zeit dem verursachenden Prozess angerechnet.

5.10.3 Application Level

Load Balancing

-> Server Farm

IDS-Systeme = Intrasion Delection System
Erkennung von Angriffen im Netz

Wie IDS-Systeme (Beispiel SNORT) arbeiten

- für ein Netz (Analyse des Datenstroms)
- auf einem Host (Analyse Logs, Prüfsumme)
- zur Überwachung einer Applikation

Net based IDS (NIDS)

angewendete Methoden
- Signatur basierte Erkennung = Suche nach typischen Angriffsmustern in Daten
- IPS Intrusion Prevention Systems = Leiten Gegenmaßnahmen ein
- Firewall - Regel
- TCP - Verbindungen rücksetzen
- Anomalie basierte Erkennung = Verhalten des Angreifers weicht vom “normalen” Verhalten ab. **Bsp:** Portscan, Directory Traversal, Dos

Protocol Standard Level

itrace = Zurückverfolgen von Angriffen

6 Werkzeuge für Sicherheitstests

6.1Scan Methoden

6.1.1Portscan- Grundlagen

2 Gründe für Portsan's:

- Überprüfen eigener Sicherheitlöcher
- Fremde Netze Untersuchen auf Schwachstellen (illegal!),
  um einfache eindringen zu können.

Firewalls und **IDS-Systeme** versuchen Portscans **aufzuspüren** und zu **blockieren** bzw. (IDS) einen Admin zu benachrichtigen.
Portscan untersucht fremde Rechner auf offene Ports/Dienste die verfügbar sind. Dienste werden i.d.R unter well-known ports zur Verfügung gestellt.

6.1.2Scan Typen

6.1.2.1TCP Connect Scan

Die offenen Ports werden durch RFC- Konformen TCP Verbindungsaufbau (3- Way Handshake) geprüft. War er erfolgreich Beendet der Angreifer die Verbindung wieder mit einem 4Way Handshake. Auch als **"aktiver Port- Scan"** bezeichnet.
Dieser Port- Scan ist leicht zu entdecken weil ein **abgeschlossener Verbindungsaufbau** zustande kommt und damit leicht zu protokollieren ist.
Verschleierung durch "stealth Scans"!
Anormale TCP Pakete gesendet dabei können übliche Flags gesetzt oder nicht gesetzt sein.

6.1.2.2Halb offener Scan (TCP SYN Scan) = "SYN- Stealth"

Bei dieser Variante des **aktiven Portscans** bricht der Angreifer den Verbindungsaufbau ab, sobald er vom Opfer auf sein SYN- Paket ein SYN,ACK- Paket erhält. Der Angreifer unterschlägt damit das 3.Paket eines 3-Way Handshakes, so wie beim SYN- Flood Angriff.
Da bei diesem Scan **kein vollständige Verbindung aufgebaut wird, protokollieren** die **Systeme** diese Versuche in der Regel **nicht**.
=> **Ist schwieriger von IDS- Systemen zu erkenen als TCP Connect Scan!**
Stealth Scans sind solche die im Normalfall nicht von den IDS- Systemen protokolliert werden. Sowie die die keine RFC- Konformen Protokollablauf entsprechen verschicken.

Usadel:"Zählt diesen Scan nicht zu den Stealth Scans da bis zum 3.Paket der RFC- Konforme Protokollablauf eingehalten wird."

6.1.2.3FIN-, NULL-, XMAS-, YMAS- Tree Stealth Scan

Da moderne Firewalls und IDS- Systeme den SYN- Scann erkennen entstand dieser Scan Typ. **Bezeichnung leitet sich ab von den TCP Flags die im Paket gesetzt sind**

Grundlagen Reaktion eines Rechners nach RFC 793(TCP):

- Ist der **Port geschlossen** (Kein Dienst aktiv) muss auf das SYN- Paket ein RST-Paket gesendet werden und damit die halboffene Verbindung geschlossen werden.
- Ist der **Port** geöffnet (**“open” = Dienst aktiv**) sollte das **Paket verworfen** werden da ein Server bei einer n**euen Verbindung** auf das gesetzte **SYN Flag wartet.**

Ferne sind keine Verbindungen geöffnet und es wird ein SYN Paket gesendet, auf das die IDS- Systeme in der Zwischenzeit reagieren. Da aber kein Verbindungsversuch vorliegt wird er nicht protokolliert.

=> **Entdeckung schwierig**

Die zugrundeliegende Idee besteht darin, dass laut RFC 793, S.64 geschlossene Ports auf derartige Zugriffe mit einem RST - Datagramm antworten während ansprechbare Ports die Anfragen ignorieren.

- **FIN Scan:** FIN- Flag im TCP Paket gesetzt.
- **NULL Scan:** Keine Flags gesetzt im TCP Paket.
- **XMAS- Tree Scan:** Hat FIN, URG, PSH der TCP Flags gesetzt.
- **YMAS Scan:** vll nur URG Flag gesetzt.

6.1.2.4 ACK Scan

Mit dem ACK- Scan soll untersucht werden ob eine Firewall “Statefull” arbeitet. Dabei wird ein TCP Paket mit zufälliger ACK# (Number) geschickt, dies sollte eine Statefull Firewall mit einem RST- Paket quittieren.

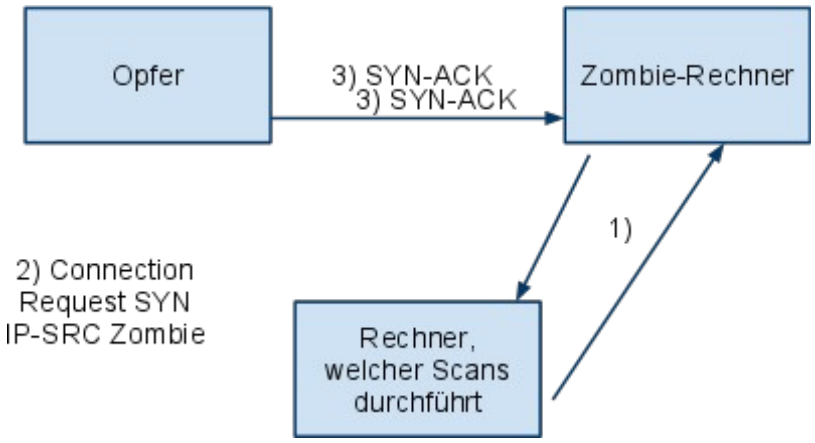
Falls die Firewall das Paket durchlässt, ist sie nicht statefull

6.1.2.5IDLE Scanning

Dieser Scantype verwendet eine fortgeschrittene Technik des stealth scans. Die Scan-Datenpakete stammen nicht vom Rechner, der den Scan ausführt. Für diese Zwecke wird ein sog. Zombie-Rechner eingesetzt.

Damit ein Rechner als Zombie-Rechner eingesetzt werden kann, muss er folgende Bedingungen erfüllen:

- Keine Kommunikation (Rechner schläft) mit anderen Rechnern -> IP -IDs ändern sich nur falls der Angreifer Pakete sendet.



(1) IP-ID vom Zombie ermitteln. Steckt in der Antwort des Zombie Rechners. IP-ID des Zombies merken!!

(2) RST - Port geschlossen (Dienst nicht vorhanden)

- sendet nicht: er hat keine Verbindung durch SYN geschlossen.

(3) IP-ID vom Zombie nun ermitteln. Antwort zu 5) erhöht Zombie IP-ID um

1 - falls Zombie an Opfer nichts gesendet hat

2 - falls Zombie RST an Opfer geendet hat - PORT OPEN!

6.1.3 Verbergen von Scans

6.1.3.1 Timing

Der Scanner Nmap passt seine Verzögerungszeiten zwischen Scan-Paketen automatisch an die Netzwerküberwachungsgeschwindigkeit und die Antwortzeiten des Opfers an.

Die Zeiten können jede Woche angepasst werden je nachdem ob man einen schnellen Scan oder einen verborgenen Scan durchführen will.

Timings sind:

Paranoid, Sneaky, Polite, Normal, Agressive und Insane!

Paranoid (-T0):

Wartezeit zwischen zwei Scan-Paketen: 5 Minuten

Insane (-T5):

Kleinste Zeitverzögerung zwischen 2 Scan-Paketen

6.1.3.2 Verwendung von Lockvögeln (decoy)

Scans von anderem System vortäuschen

- viele gespoofte Adressen verwenden

- eigene Scans einstreuen

6.1.3.3 FTP-Bounce

Angreifer -> FTP PROXY -> Scan --> Opfer

Man kann einen FTP Proxy verwenden oder einen FTP Server, bei dem im Port Kommando IP-Adresse frei gewählt werden können.

6.1.3.4 Scan ohne Ping (welche System befinden sich im Subnetz)

Keine ICMP-Ping weil einige Systeme nicht auf ICMP Pakete antworten

ACK – Scan

6.1.3.5 Fragmentierung

Pakete in kleine Pakete aufspalten

- sind dadurch schwer zu erkennen

Verwendung bei den Schaltern:

- sS (für SYN)

-sF (für FIN)

-sX (für XMAS)

-sN (für NULL)

6.1.3.6 IDLE Scanning

(siehe Kapitel Scan-Methoden)

6.2 Nmapfe (Scan-Werkzeug mit grafischer Oberfläche)

Nmpafe ist eine transparente graphische Benutzteroberfläche für nmap. Alle Einstellungen, die Sie in den Dialogen machen, werden umgesetzt auf den entsprechenende Kommandozeilen-Aufruf von Nmap. Man kann NmapFE als normaler Benutzer verwenden, muss also nicht Administrator sein. Aber die Möglichkeiten, die man besitzt, sind dann eingeschränkt.

6.2.1 OS-Fingerprinting

OS detection
Betriebsystem & Kernel Version soll erkannt werden. Zur Erkennung werden besondere Testpakete gesendet.

6.2.2 hping2

Hping ist ein Kommandozeilen orientierter TCP/IP-Paketgenerator und Analysator. Die Erzeugung von TCP-, UDP-, ICMP- und Raw-IP Paketen wird unterstützt. Ferner wird ein traceroute Modus zur Verfügung gestellt.

Einsatzmöglichkeiten:

- Testen der Firewall
- Erweiterter Port Scan
- Network testing, using diffrent protocols, TOS, fragmentation
- hping can also be useful to students that are learning tcp/ip

6.2.3 Beispiele:

6.2.3.1 Portscan

hping2 --scan 1-100,8888, known -S target.host.com

(alle Ports zwischen 1 und 100, Port 8888 und alle “well known Ports”, -S: setze SYN)

6.2.3.2 Ping

hping2 -1 -c 5 -j target.host.com

(-1: ICMP-Modus, -j: Antwortpaket als Hex-Dump anzeigen)

6.3Nessus (BSI)

Nessus ist ein System mit dem Rechner auf offene Ports und Sicherheitslücken getestet werden können. Nessus zählt zur Klasse der Vulnerability Scanner(Verwundbarkeitsprüfer).Diese Programme untersuchen das Zielsystem auf Sicherheitslücken die u.a. in folgenden Bereichen auftreten dürfen

- verfügbare Dienste
- Druck- und Laufwerksfreigaben
- Einhaltung der Richtlienien für die Wahl von Passwörter
- Offene Ports
- Fehlende Patches und Updates

Nessus basiert auf dem Client Server Prinzip.

6.3.1(BSI OSS Security Suite) - Version 2.0

Das spezielle Modul für das Lokalauditing heißt SLAD = Security Local Auditing Deamon.) SLAD is a tool for performing local security checks againgst GNU/Linux systems and is published entirely under the GPL. SLAD has been primarily developed the BOSS project to work together with Nessus to enhance its local scanning capabilities. For example, scanning for weak passwords with a tool like John-the-Ripper is something that simply cannot achieved by a network scan.
SLAD is implemented in Perl and provides an extendable plugin architecture allowing to use various GPL-based security scanners and auditing tools under one common framework. Currently, SLAD comes packaged with

- **John-the-Ripper**: Password Knacker
- **Chkrootkit**:Sucht nach Rootkits
- **LSOF**: List open files - zeigt offene Dateien an
- **ClamAV**: Virenscanner
- **Tripwire**: File Integrity Checker - überwacht Datein auf Veränderungen (Prüfsumme) HIDS / Host / Intension / Detecion / Systems
- **TIGER**: UNIX Security Scanner - Security audit and untrusion detection
- **Logwatch**: überwacht Logs / Loganalyser and Reporter
- **TrapWatch**: überwacht SNMP Traps
- **LM-Sensors**: Hardware Monitoring
- **snort**: Network IDS

6.4ntop: Netzwerk-Monitor

ntop ist eine web-basierte Anwendung für das Monitoring des Netzwerks. Zunächst wird der ntop-Dienst gestartet, danach kann diese Anwendung unter der URL http://localhost:3000 administriert werden.

6.5ngrep

Das Unix-Werkzeug grep dient zum Durchsuchen von Dateien nach bestimmten Inhalten. Analog dazu wurde ngrep entwickelt, um Netzwerkverkehr zu “durchsuchen”

ngrep kann dazu benutzt werden, um aus einer Flut von unverschlüsselten Daten bestimmte “interessante” Informationen herauszufischen. Natürlich besteht hier ein hohes Missbrauchspotential, aber es gibt auch legitime Einsatzgebiete:

- Aufklärung und Schulungen
- Überwachungen der Netzwerksicherheit
- Erkennung von unerlaubtem Daten-Zugriff
- Analyse von Netzwerk-Protokollen

6.6netcat

netcat ist ein Werkzeug um Daten im Netzwerk über das TCP- oder UDP PRotokoll zu lesen oder zu schreiben. Es wurde zur Verwendung in anderen Skripts oder Programmen konzipiert, bietet aber auch allein viele Möglichkeiten zur Analyse oder Beobachtung von Datenübertragungen

netcat kann als SERVER, CLIENT und als WEBBROWSER gestartet und verwendet werden.

7 ACL - Access Control List

7.1Einloggen, Configmode

>enable

#configure terminal

7.1.1ACL an Interface binden

(config)# interface ethernet 0

(config-if)# ip access-group 101 out

(config-if)# ip access-group meine_acl in

7.2ACL erstellen

7.2.1Standard ACL

(config)# access-list 1 permit 5.6.0.9 0.0.255.255

(config)# access-list 1 deny host 192.168.2.1

(config)# access-list 1 permit any log

#nur Source Address, keine Ports

7.2.2Extended ACL

(config)# access-list 101 deny tcp 192.168.2.0 0.0.0.255 host 192.168.2.5 eq 21

(config)# access-list 101 permit tcp any gt 1023 host 192.168.2.5 eq 21

(config)# access-list 101 permit ip host 1.2.3.4 any

(config)# access-list 101 permit tcp host 1.2.3.4 15.21.2.0 0.0.0.3 establishd

#Source IP, Wildcard, Port, Dest IP, Wildcard, Port, established/log

7.2.3Named ACL

(config)# ip access-list extended blub

(config-std-nacl)# deny ip any any log

(config-std-nacl)# end

(config)# ip access-list standard wow

(config-std-nacl)# permit host 192.168.1.2

(config-std-nacl)# end

7.2.4Complex ACLs

7.2.5Reflexive ACL

(config)# ip access-list extended mein_out

(config-std-nacl)# permit tcp 192.168.0.0 0.0.255.255 any reflect MEINZEUG

(config-std-nacl)# end

(config)# ip access-list extended mein_in

(config-std-nacl)# evaluate MEINZEUG

(config-std-nacl)# end

(config)# interface ethernet 0

(config-if)# ip access-group mein_out out

(config-if)# ip access-group mein_in in

#nur named extended IP-ACLs, beide an gleiches Interface binden

7.3ACLs überprüfen

#show ip interface # welche ACLs sind an Interfaces gebunden

#show access-lists # welche Regeln enthalten die ACLs?

7.4ACLs editieren (nur named ACLs)

#show access-lists

(config)# ip access-list standard meine_acl

(config-std-nacl)# 15 permit host x.x.x.x # Einfügen der ACL an 2. Stelle

(config-std-nacl)# end

7.5ACLs löschen

(config)# no access-list 100

7.6Aufgabe1

Router(config-if)#ip access-group 101 out

Router(config)#access-list 101 permit udp 192.168.1.0 0.0.0.255 host 192.168.2.2 eq domain

Router(config)#access-list 101 permit tcp 192.168.1.8 0.0.0.3 host 192.168.2.2 eq www

Router(config)#access-list 101 permit tcp host 192.168.1.4 host 192.168.2.3 eq telnet

Router(config)#access-list 101 permit icmp any any echo #in dieselbe Richtung

Router(config)#access-list 101 permit icmp any any echo-reply #"

access-list 101 permit tcp host 192.168.1.100 eq www host 192.168.2.4 gt 1023 #Antwortpakete

7.7Aufgabe2

Router0(config)#interface FastEthernet1/0

Router0(config-if)#ip access-group 100 out

Router0(config)#access-list 100 permit tcp 192.168.1.0 0.0.0.255 host 192.168.3.2 eq www

Router0(config)#access-list 100 permit udp 192.168.1.0 0.0.0.255 host 192.168.3.2 eq domain

Router0(config)#interface FastEthernet0/0

Router0(config-if)#ip access-group 101 out

Router0(config)#access-list 101 permit icmp host 192.168.2.4 192.168.1.8 0.0.0.3 echo

Router0(config)#access-list 101 deny icmp any any echo

Router0(config)#access-list 101 permit ip any any

Router0(config)#access-list 100 permit tcp host 192.168.2.3 eq telnet host 192.168.3.100 gt 1023 #Antwortpakete

7.8Aufgabe3

Router0(config)#ip access-list extended mein_reflex

Router0(config-ext-nacl)#permit udp 192.168.1.0 0.0.0.255 host 192.168.2.2 eq domain reflect dns_zeug

Router0(config-ext-nacl)#permit tcp 192.168.1.4 0.0.0.3 host 192.168.2.2 eq www reflect web_zeug

Router0(config)#ip access-list extended mein_reflex_antwort

Router0(config-ext-nacl)#evaluate dns_zeug

Router0(config-ext-nacl)#evaluate web_zeug

Router0(config)#interface FastEthernet1/0

Router0(config-if)#ip access-group mein_reflex out # muss im selben Interface sein

Router0(config-if)#ip access-group mein_reflex_antwort in