# Systemsicherheit: Teil ohne Unterlagen

## Penetrations-Test

- Versuch in einen Rechner einzudringen
  - Angriff im Auftrag des Betreibers
- Phasen
  - Reconnaissance (Auskundschaftung)
  - Enumeration (Finden von Angriffsmöglichkeiten), z.B.
    - Port Scanning
    - Verwundbare Versionen von Diensten / Betriebssystemen durch OS-Fingerprinting identifizieren
    - Konfigurationsfehler finden
    - SNMP ausnutzen (Community string)
  - Exploitation (Möglichkeiten durch Ausnutzen der Sicherheitslücke identifizieren)
  - Documentation (Abschlussbericht)
    - Schwachstelle
    - Details
    - Risikoeinstufung
    - Lösungvorschläge
    - Managementkurzbericht
    - ToDo-Liste: Was kann sofort gemacht warden?

## Enumeration

s.o.

## OS-Fingerprinting

- die Erkennung von Betriebssystemen durch die Beobachtung diverser Reaktionsweisen der Betriebssysteme im Netzwerk aus der Ferne
- Ziel: verwundbare Versionen von Diensten / Betriebssystemen identifizieren

## Remote-Forensic- Software

- Software für die Online-Durchsuchung für den unbemerkten Zugriff von Strafverfolgungsbehörden auf informationstechnische Systeme

## Keylogger

- eine Hard- oder Software, die dazu verwendet wird, die Eingaben des Benutzers an einem Computer mitzuprotokollieren und dadurch zu überwachen oder zu rekonstruieren
- Keylogger werden beispielsweise von Hackern verwendet, um an vertrauliche Daten – etwa Kennworte oder PIN – zu gelangen

## Entstehung TCP-ACK-Storm

- Def. ACK−Storm: Die Kommunikationspartner werfen sich gegenseitig falsche Sequenznummern vor. Sie droppen eingegangene Pakete und fordern mit ACK die »richtigen« an. (Angreifer sendet eine falsche SEQ#)
- Um einen ACK−Storm auszulösen muss man die Sequenz−Nummer so manipulieren, dass gilt:
  - CLT_SEQ != SRV_ACK
  - SRV_SEQ != CLT_ACK

- Ist eine solche Situation gegeben, so spricht man vom desynchronized state (da die Sequenz−Nummern nicht mehr synchron sind).
- Versucht nun der Server Daten zum Client zu senden (oder umgekehrt), so kommt es zum blanken Chaos:
- der Client empfängt das Packet und wundert sich über die merkwürdigen Sequenz−Nummern des Servers. Also sendet der Client ein Packet mit (aus seiner Sicht) richtigen Sequenz−Nummern an den Server zurück.
- Nun empfängt der Server dieses Packet und wundert sich ebenfalls über diese merkwürdigen Sequenz−Nummern, also sendet er ein Packet mit (aus seiner Sicht) richtigen Sequenz−Nummern an den Client. Und jetzt fängt der ganze Spaß also wieder von vorne an

## Funktionweise SYN-Flood-Angriff

- böswilliger Client unterschlägt das ACK-Datenpaket
- Server speichert diese halboffene Verbindung und belegt Ressourcen
- viele halboffene Verbindungen -> Buffer voll

## Maßnahmen gegen SYN-Flood-Angriff

- SYN-Cookies-Mechanismus
- TCP-Paramter ändern -> Größe der Backlog-Queue erhöhen, Anzahl der Versuche SYN/ACK zu senden herabsetzen
- Bsp. Windows: Schwellwert für halboffene Verbindungen

## Botnet

- fernsteuerbares Netzwerk (im Internet) von PCs, das aus untereinander kommunizierende Bots besteht
- Kontrolle durch Würmer, bzw. Trojanische Pferde, die den Computer infizieren
- Netzwerke werden z.B. für Spam-Verbreitung, DoS-Attacken verwendet, ohne das die betroffenen PC-Nutzer etwas darüber erfahren

## tiny-fragments-Angriff / Overlapping IP-Fragments-Angriff

- Firewalls erkennen Verbindungsaufbau am ersten Datenpaket einer TCP-Verbindung:
  - SYN-Flag = 1
  - ACK-Flag = 0
- bei allen weiteren Datenpaketen: ACK = 1
- Grundidee: wenn das erste Datenpaket einer Verbindung die Firewall passiert, lässt die Firewall dann alle weiteren Pakete dieser Verbindung ebenfalls durch
- Version 1 (**Tiny Fragments**):
  - erstes Fragment ist so klein, dass TCP-Flags nicht enthalten sind
  - zweites Fragment enthält Flags für Verbindungsaufbau (SYN=1,ACK=0)
- Version 2 (**Overlapping Fragments**):
  - ersten beide Fragmente überschneiden sich im Bereich der TCP-Flags
  - beim ersten Fragment ist SYN=0 und ACK=1 (nicht das erste Paket, wird von Firewall durchgelassen)
  - beim zweiten Fragment SYN=1 und ACK=0 (Verbindungsaufbau)
  - zweites Fragment überschreibt beim ersten Fragment den Flag-Bereich (beim Zusammenbau)
  - Fragmente werden erst beim Empfänger wieder zusammengebaut

## IP-Spoofing

- Angreifer verwendet IP-Adresse eines anderen Rechners

- mögliche Ziele:
    - Herkunft des Angriffs verschleiern
    - Zugriffsschranken auf den angegriffenen Rechnern umgehen (z.B. Access Lists: MS-IIS, Linux TCP-Wrapper)
- **Blind Spoofing**
    - Antwortpakete kommen auf Grund von Routing nicht beim Angreifer an
    - Angreifer nutzt daher source route IP-option (Datenverkehr zum Angreifer umleiten)
    - Ziel: Dienste, die als Autentifizierungsmechanismus IP-Adressen des Client nutzen z.B. rhosts-Datei manipulieren für rlogin, rsh, rcp
    - eventuell ist Sequence-number-attack zusätzlich notwendig um Sequenznummern der Antwortpakete zu erhalten

## Oscillation-Data-Flood-Angriff

## Zweck TCP-Wrapper

- Dienste schützen, indem Zugriffe nur von bestimmten IP-Adressen oder IP-Adressbereiche erfolgen können
- eingehende Dienstanforderungen überwachen (Log)

## Funktionsweise Tripwire

- Intrusion Detection System
- um zu erkennen ob und welche Dateien durch “Fremdeinflüsse” verändert werden zu installieren
- erstellt Prüfsummen aller zu schützenden Verzeichnisse und Dateien

## ARP-Spoofing(Funktionsweise, Tarnung, Alternative)

Funktionsweise:

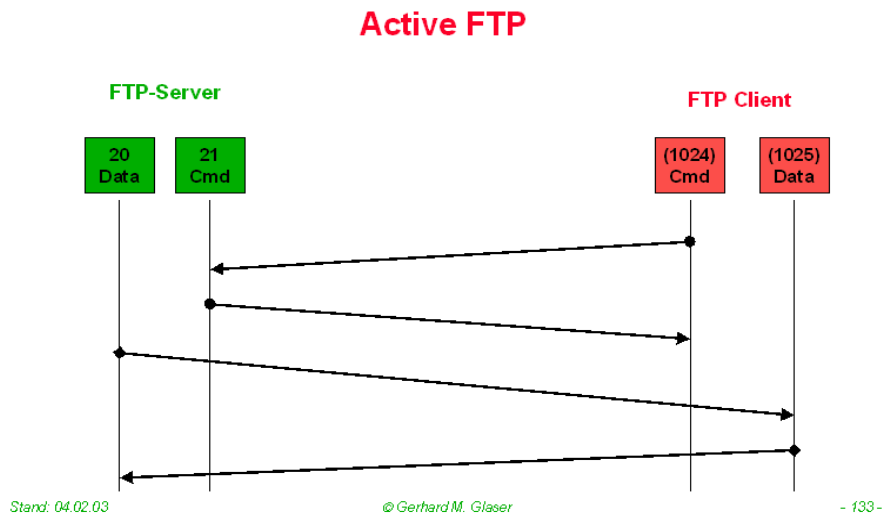
- Variante 1
    - Angreifer gibt sich die MAC-Adresse des anzugreifenden Rechners
    - Datenverkehr kommt nun beim Angreifer an und wird abgehört
    - danach Datenverkehr an den eigentlichen Bestimmungsort weiterleiten
        1. PC2 sucht MAC von PC1 -> Broadcast
        2. PC3 erhält ARP-Request, fälscht einen ARP-Reply an PC2 mit der Sender-IP von PC1
        3. PC2 trägt in ARP-Cache falsche Zuordnung ein, Datenpakete an PC1 gelange zu PC3
        4. PC3 leitet nach Auswertung Datenpakete an das eigentliche Ziel PC1 (Tarnung)
- Variante 2
    - PC2 sucht MAC von PC1 -> Broadcast
    - PC2 lernt MAC zur IP von PC1 (Arp-Cache)
    - PC3 sendet einen ARP-Reply mit einer anderen, nicht existenten Mac an PC2
    - PC2 aktualisiert seinen ARP-Cache und trägt die falsche MAC (von PC3) ein
    - Paket an PC1 kommt bei allen Systemen im geswitchten Netzsegment an, da Switch an alle Ports aussendet (kennt Adresse nicht)
    - Paket wird von allen Netzwerkkarten verworfen, außer beim Angreifer (promiscuous mode)
    - Angreifer modifiziert das Paket und sendet es an das eigentliche Ziel weiter (Tarnung)

## Probleme SNMP-Community-Strings

- SNMP für Verwaltung von Netzkomponenten
- Community Strings sind eine Art Passwort für Zugriffsrechte
- je nach SNMP-Version unverschlüsselter Versand über das Netzwerk
- Default-Werte:
    - public -> lesen
    - private -> schreiben

## Probleme Active-FTP






- active: Server Verbindung für den Datenkanal auf
- Port-Kommando vom Client: Port client-IP, Client-Data-Port
- Firewall auf Server-Seite
    - aus dem Internet alle IPs auf den FTP-Server zulassen -> Port 21
    - FTP-Server SRC-Port 21 an DST-Port > 1023 an alle Rechner zulassen
    - FTP-Server von/zu SRC-Port 20 an/von alle Clients DST-Port > 1023 zulassen (+Antwortpakete)
- Firewall auf Client-Seite bzgl. Port 21:
    - Verbindungen vom Intranet ins Internet immer zulassen (Client [SYN])
    - erzeugt Regel für Server-Antwortpakete (temporäre Regel)
    - für alle FTP-Clients von allen Servern im Internet: Port 20 -an Port >1023 zulassen
    - Problem: Aus Sicht seiner Firewall initiiert ein außen stehendes System die Verbindung zu einem internen Clientrechner (und das wird normalerweise blockiert)

## Funktionsweise stateful-firewall

- überprüft u.a. TCP-Verbindungszustand

## rekursive DNS-Anfrage

- der befragte Nameserver nimmt Verbindung mit dem Nameserver auf, der die Anfrage beantworten kann

## open-resolver

- DNS-Server der Anfragen von extern für externe Domains auflöst (rekursive Anfragen)

## Wieso komplexe TCP-Sequenznummern

- um das Erraten von Sequenznummern zu erschweren, z.B. bei Blind-Spoofing-Attacke verwendet

## IDS

- =Intrusion Detection System
- versucht Angriffe zu erkennen

## IPS

- =Intrusion Prevention System
- Reagieren und blockieren z.B. die Pakete des Angreifers

## Beschreibung & Informationsauswertung HIDS, NIDS

**HIDS**

- Host-based intrusion detection system
- muss auf jedem zu überwachenden System installiert werden
- besteht aus einem Agent der Eindringungen identifiziert durch:
    - Systemaufrufe analysieren
    - Anwendungs-Logs
    - Dateisystem-Veränderungen
    - sowie andere Host-Aktivitäten und -ZuständeProxy-Arp

**NIDS**

- Network intrusion detection system
- versuchen:
    - alle Pakete im Netzwerk aufzuzeichnen
    - zu analysieren
    - verdächtige Aktivitäten zu melden
    - außerdem, aus dem Netzwerkverkehr Angriffsmuster erkennen

## Wildcard-Mask bei Cisco-ACLs

- werden verwendet, um einen Adressbereich anzugeben
- besteht aus 32-Bit, in 4 Byte aufgeteilt

## Proxy-Arp bei Router

- Router beantwortet ARP-Anfragen für Hosts
- wird bei getrennten Subnetzen gemacht, wenn die Anfrage von einem in ein anderes Netz geht
- z.B. wenn Subnetzmaske falsch gesetzt ist und Rechner im gleichen Subnetz vermutet werden, obwohl sich diese in unterschiedlichen Subnetzen befinden; Der Router gibt dann beim Arp-Reply die MAC seines Interfaces an, über das der gesuchte Rechner erreicht werden kann

## NAPT-Router: Zuordnung von Datenpaketen ins Internet zu bestimmten Rechnern des Intranet

- Verfahren bei dem viele IP-Adressen auf eine IP-Adresse abgebildet werden

- zusätzlich zur Quell-IP-Adresse wird der Quell-TCP-/UDP-Port vom NAT(Network Address Translator) vom Stub-Router umgeschrieben, bevor das Paket zum Ziel weitergeleitet wird
- Stub-Router führt Stub-Netzwerk-IP, Quellport des Stub-Rechners, zugewiesener Quellport in einer internen Tabelle (Masquerading-Table)

## tcpd: welche Dateien berechtigte IP-Adressen

- /etc/host.allow
- /etc/host.deny

## source route option bei IP (Beschreibung & Grund für Blockierung durch Firewalls)

- Festlegen eines vorgegebenen Pfads für Antwortpakete
- wird blockiert, weil der Datenverkehr der Antwortpakete sonst zum Angreifer umgeleitet werden kann

## DNS: ID-Fälschung

- bei Anfragen und zugehörigen Antworten wird dieselbe ID verwendet
- ID fälschen (wird erraten) um falsche DNS-Antwort unterzuschieben

## DNS: Vergiften DNS-Namecache (welche Erleichterung gibt es)

im Skript nicht beantwortet

## DNS: Erkennungsmerkmal Antwortpaket aus Namecache

- Haeder-Flag: Auth.Answer nicht gesetzt -> Antwort ist aus Cache

## SQL-Injection

- bezeichnet das Ausnutzen einer Sicherheitslücke in Zusammenhang mit SQL-Datenbanken, die durch mangelnde Maskierung oder Überprüfung von Metazeichen in Benutzereingaben entsteht
- der Angreifer versucht dabei, über die Anwendung, die den Zugriff auf die Datenbank bereitstellt, eigene Datenbankbefehle einzuschleusen

## Footprinting

- 1.Phase des Angriffs
- Informationsbeschaffung ohne direkten Zugriff auf das Zielsystem (z.B: WHOIS-Datenbank, DNS-Einträge, Firmenwebseiten)

## Firewalking

- Pfad durch Firewalls finden (Suche nach offenen Ports, Fehlkonfiguration, Mapping des Zielnetzes)

## Sniffing

- Sniffer untersuchen und analysieren den Netzwerkverkehr

## Promiscuous Mode

- alle Datenpakete werden von der NIC aufgesammelt

## Session Highjacking

- Übernahme von Sitzungen (TCP, Web)

## Stub-Netz

- geht nur über eine Route raus (ins Internet)

## Arp-Watch

- überwacht ARP-Pakete
- Tool um ARP-Spoofing zu verhindern

## FTP-Bounce Attack

k.A.

## xinetd

- Superserver
- Dienstprogramme werden dadurch gestartet wenn ein Verbindungswunsch vorliegt -> ressourcenschonend

## Backlog-Queue

- Puffer für TCP-Verbindungsanfragen