Scan-Techniken
Ein Überblick

Klaus Möller
DFN-CERT GmbH
Februar 2002

DFN CERT

Agenda

- Was sind Scans ?
- Scan-Techniken
  - ICMP Scans
  - TCP Scans
  - UDP Scans
  - Weitere Scan-Techniken
- Umgang mit Scans

DFN CERT

Was sind Scans ?

- Einbruchszyklus:
  - Informationssammlung
    - Öffentliche Informationen, z. B. Webseiten
    - Passive Verfahren: Sniffing, Traffic Analysis
    - Aktive Verfahren: Scans
  - Einbruch: Exploits
  - Verstecken und Festsetzen: Rootkits, Backdoors, ...
  - Mißbrauch: (D)DoS, Warez, Scans, weitere Einbrüche

DFN CERT

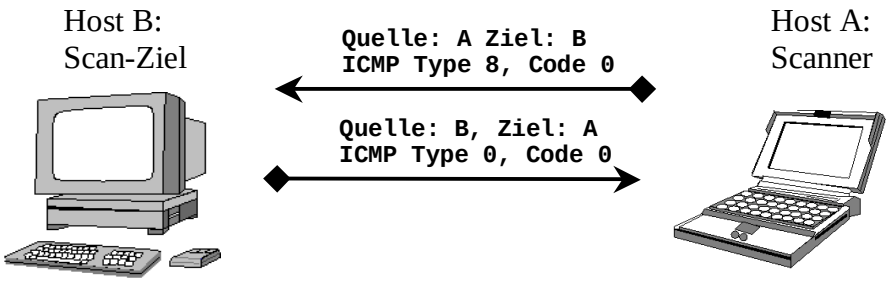
Ziele von Scans

- Vorhandene Systeme bestimmen:
  - ICMP Scans (`ping, hping, X`)
- Aktive Dienste bestimmen:
  - Portscans: TCP und UDP (`nmap`)
- Schwachstellensuche:
  - Vulnerability Scans (`nessus, ISS, NetRanger`)

DFN CERT

ICMP Scans

- Echo Reply Scan



DFN CERT

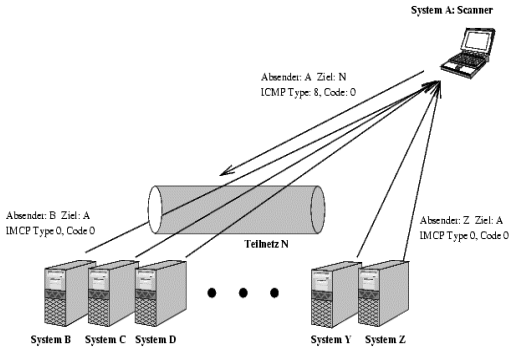
Weitere ICMP Scans

- Timestamp Request: Type 13 / 14
- Information Request: Type 15 / 16
  - Antwort nur von Konfigurationsserver
  - Request nicht über Subnetz Grenzen
- Address Mask Request: Type 17 / 18
  - Antwort nur von *Authorative Agent* (Router)
  - Request nicht über Subnetz Grenzen

DFN CERT

ICMP Broadcast Scans

- Zieladdresse Broadcast Adresse des Subnetzes
- Response von allen Hosts im Subnetz
- Default heute: Keine Weiterleitung von Directed Broadcasts !
- Bei Erfolg: Kenntnis über Smurf Amplifier



DFN CERT

ICMP Host Fingerprinting

- Methoden:
  - ICMP Response Rate
  - Datenmenge in ICMP Error Messages
  - Änderungen im IP Header
  - Fehler im IP Header von ICMP Error Messages
  - TOS Wert in ICMP Unreachable
  - Antwort auf ICMP Broadcasts
  - Defaults für IP ID, IP TTL

DFN CERT

ICMP Host Fingerprinting (Forts.)

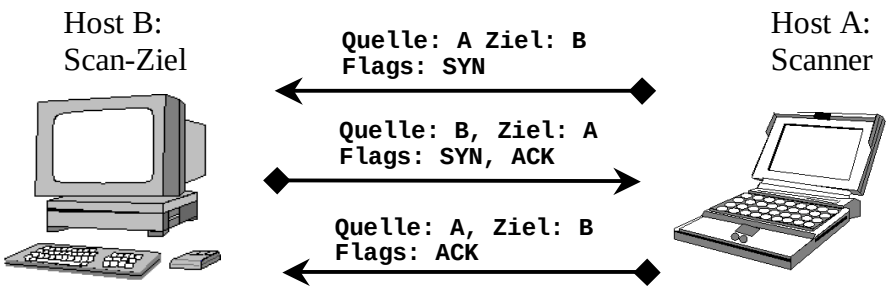
- Beispiel: xprobe 0.0.2 vs. Linux 2.4.10

```
host # ./xprobe -v -i lo localhost

LOG: Target: 127.0.0.1
LOG: Netmask: 255.255.255.255
LOG: probing: 127.0.0.1
LOG: [send]-> UDP to 127.0.0.1:32132
LOG: [98 bytes] sent, waiting for response.
TREE: Cisco IOS 11.x-12.x! Extreme Network Switches.Linux
2.0.x!2.2.x!2.4.x.
TREE: Linux kernel 2.0.x!2.2.x!2.4.x! Based.
TREE: Linux kernel 2.2.x!2.4.x! Based.
LOG: [send]-> ICMP echo request to 127.0.0.1
LOG: [68 bytes] sent, waiting for response.
TREE: ICMP echo/echo reply are not filtered
FINAL:[ Linux 2.2.x/2.4.5+ kernel ]
```


DFN CERT

TCP Connect Scans

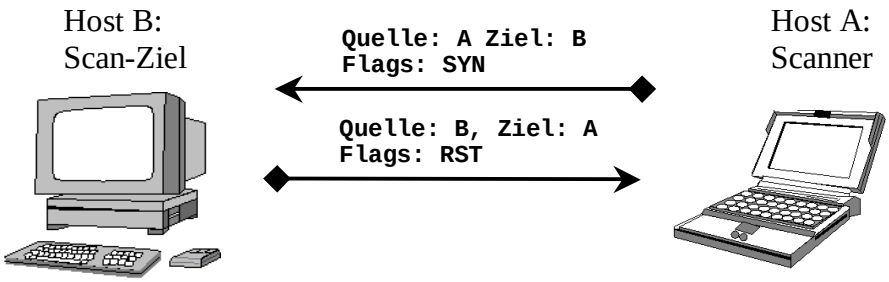
- Connect Scan auf offenen Port



DFN CERT

TCP Connect Scans (Forts.)

- Connect Scan auf geschlossenen Port



DFN CERT

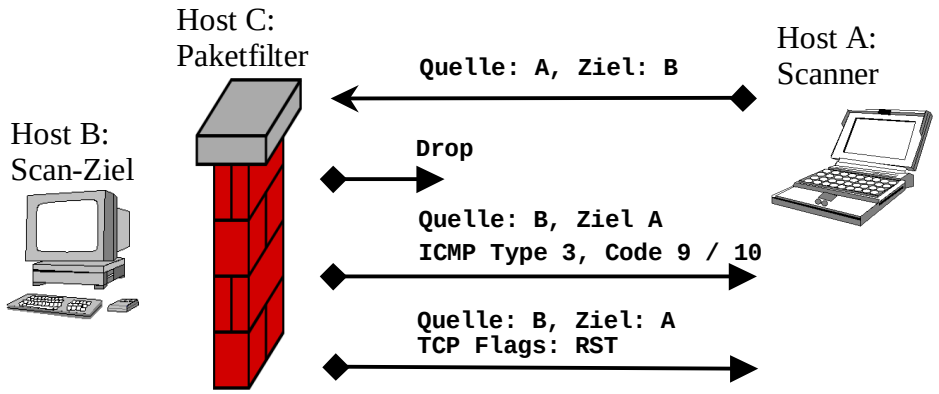
TCP Connect Scans (Forts. 2)

- Implementierung einfach
  - Std. Calls: `socket(), connect(), ...`
- Keine Root Rechte
- Zuverlässig:
  - False Negative nur bei Paketfiltern
  - Keine False Positive
- Schnell

- Leicht zu entdecken
  - TCP-Wrapper, IDS Connection Logger, Paketfilter, ...
  - Aber: Nicht einfach von normalen Verkehr zu unterscheiden

DFN CERT

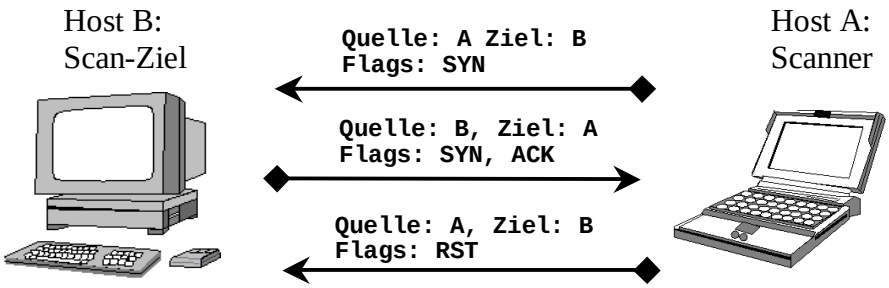
Einfluß von Paketfiltern



DFN CERT

TCP SYN (half-open) Scans

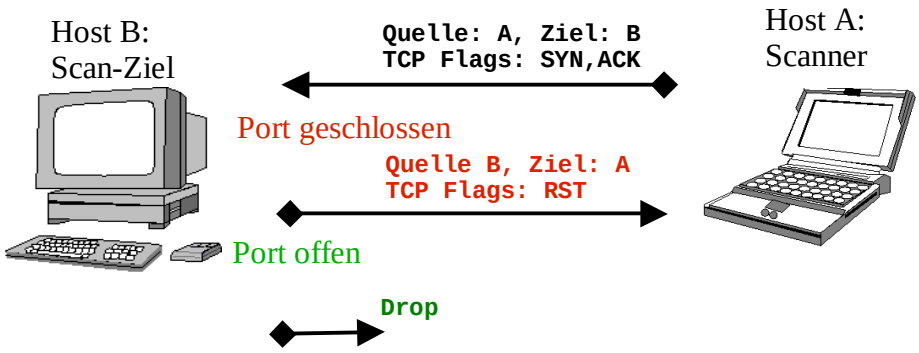
- SYN Scan auf offenen Port



DFN CERT

TCP SYN-ACK Scans

- SYN-ACK Scan



DFN CERT

TCP SYN-ACK Scans (Forts.)

- Stealthy: Kein formeller TCP Zustand
- Durchdringung statischer Paketfilter
  - Wenn bei einkommenden Paketen nur ACK Flag geprüft wird
- Mapping von Paketfilter-Regeln über ICMP Destination Unreachable
- Ergebnis-Invertierung: Anworten nur für geschlossene Ports
  - False Positive bei Paketverlust / Drop
  - False Negative bei RST von Paketfilter
- Einige OS antworten auch für offene Ports mit RST
  - Z. B. Windows, Cisco, ...

DFN CERT

TCP FIN, XMAS, Null Scans

- FIN Scan:
  - Wie SYN-ACK Scan, aber nur FIN Flag gesetzt
- XMAS Scan:
  - Wie FIN Scan, aber mit **allen** Flags gesetzt (mind. FIN, PSH, URG)
- Null Scan:
  - Dto., diesmal aber **keine** Flags gesetzt
- Fehlerpotential wie bei SYN-ACK Scans

DFN CERT

TCP RST Scans

- RST Scan:
  - Scan mit Paketen bei denen nur RST Flag gesetzt
  - Auf USENIX 99 zur Erklärung von RST/ RST-ACK Backscatter vorgeschlagen
  - Seither kein Tool aufgetaucht
- Potential nicht größer als FIN oder SYN-ACK Scan
  - Schlechter, da keine Antwort auf RST Pakete

DFN CERT

TCP Host Fingerprinting

- Methoden:
  - Antworten von offenen Ports auf FIN Pakete
  - RST Antwort auf RST Paket an geschlossenen Port
  - ECN Unterstützung in TCP
  - TCP Optionen: Reihenfolge, Unterstützung
  - Wert der Acknowledgement Number im ersten Paket
  - Muster bei der Generierung von ISNs
  - Wert der Window Size

DFN CERT

TCP Host Fingerprinting (Forts.)

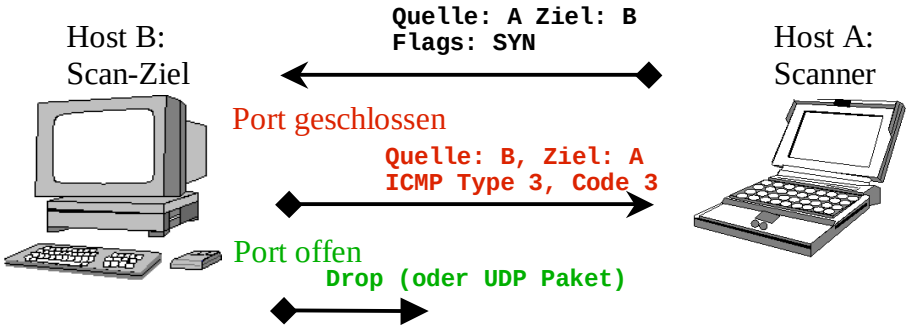
- Beispiel: nmap 2.54BETA30 und Linux 2.4.10

```
host# ./nmap -sT -vv -O localhost

No exact OS matches for host
TCP/IP fingerprint:

T1(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=7FFF%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=C0%IPLEN=164%RIPTL=148%RID=E%RIPCK=E%UCK=E%UL
EN=134%DAT=E)
```


DFN CERT

UDP Scans

- UDP Scan



DFN CERT

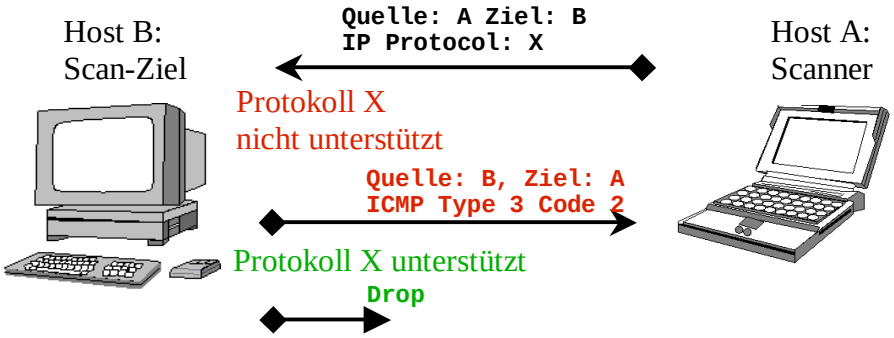
UDP Scans (Forts.)

- Hohes Potential für False Positive
  - Paketverlust, Drop durch Paketfilter
- Langsam wegen ICMP Rate Limiting auf Hosts
  - Im Vergleich mit TCP Scans
  - Zu schnell → False Positives wegen nicht versendeter ICMP Destination Unreachable / Port Unreachable

DFN CERT

IP Protocol Scans

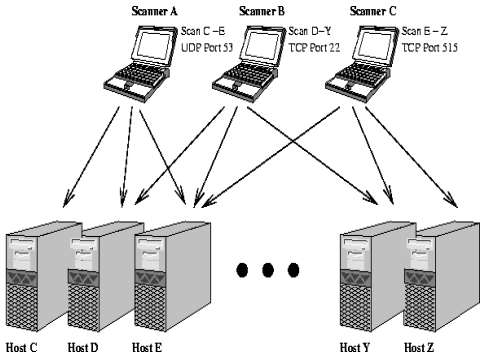
- IP Protocol Scan



DFN CERT

Koordinierte Scans

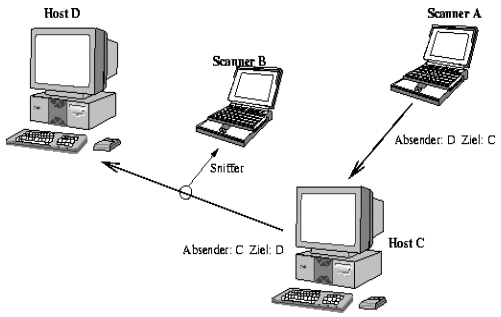
- Einzelscans bleiben unterhalb IDS Erfassungsgrenze
- Verteilung kompensiert Geschwindigkeitsein-buße
- Zusammenführung der Ergebnisse noch manuell



DFN CERT

Verteilte Scans

- Spoofing verschleiert Herkunft
- Der eigentliche Scan kann bemerkt werden



DFN CERT

Scans erkennen

- Heuristik: N Versuche in Zeitraum T gegen Ziel Z (von Quelle Q)
  - N: Anzahl IP Pakete
  - T: Sekunden, Minuten, Stunden
  - Z, Q: Menge von IP-Adressen, Portnummern
- Keine präzise Definition möglich
  - Balance zwischen False Positive und False Negative
  - Gruppieren bzgl. Arbeitserleichterung

DFN CERT

Automatische Reaktionen

- Blockieren gescannter Ports, scannender IP-Adressen
  - Leicht für Denial-of-Service zu mißbrauchen
    - Auch leicht von Angreifer zu erkennen
- Scan des Scanner
  - Ebenfalls leicht zu mißbrauchen
  - Wozu ?
- Resourcenverbrauch vs. Sicherheitsgewinn ?

DFN CERT

Warum auf Scans achten ?

- Erfahrung zeigt: Wo Rauch ist, ist auch Feuer
  - Scans folgt meistens ein Angriffsversuch
  - Auch dann, wenn **Sie** nichts bemerken !
- Zunahme von Scans auf bestimmte Dienste gutes Anzeichen für neue Schwachstellen
  - Informationsaustausch notwendig für Warnungen
- Oftmals Hinweis auf kompromittierte Hosts
  - Die meisten Techniken erfordern root-Rechte

DFN CERT

Welche Scan-Technik ist die Beste ?

- Zur Qualitätssicherung:
  - TCP Connect-Scan, UDP-Scan: Zuverlässig
  - Andere nur zum Test von IDS oder Paketfilter
- Für Angreifer:
  - Gezielt: Stealth: Koordinierte Scans, Verteilte Scans
  - Ungezielt: Beliebige Techniken, häufig Connect-Scan
    - Oft schlechte Implementierungsqualität
    - False Positives / Negatives akzeptabel für deren Zwecke

DFN CERT

Fragen

? ? ?

DFN CERT