

# Scanning Techniques

Hendrik Spiegel

[hendrik.spiegel@rub.de](mailto:hendrik.spiegel@rub.de)

Tobias Füchtler

[fuchtlert@et.rub.de](mailto:fuchtlert@et.rub.de)

Ruhr-Universität Bochum

Lehrstuhl für Kommunikationssicherheit

Ahmad-Reza Sadeghi

Netzwerksicherheit

# Überblick

- Motivation
- Protokolltypen (kurze Wiederholung)
- Ping Sweeps
- Portscanning Varianten
- Portscanning Techniken
- Betriebssystemerkennung
- Scannen durch Firewalls
- Die Praxis: NMAP
- Fazit

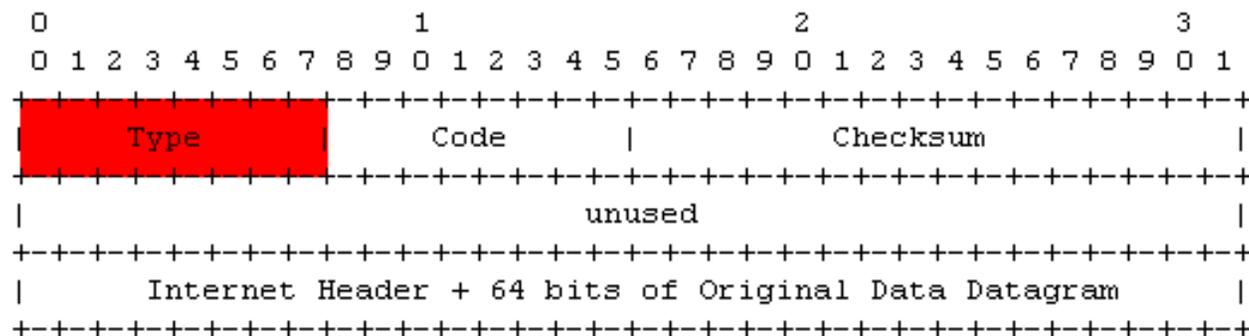
# Motivation

- Unter welchen IP Adressen eines Netzwerkes sind Rechner erreichbar?
- Welche Dienste laufen auf diesen Rechnern?
- Welche System Architektur (z.B. Sparc, Alpha, x86) liegt zugrunde?
- Welches Betriebssystem wird verwendet?

# Vorgehensweise

- Ermitteln der erreichbaren Hosts durch Pingen
- Portscans identifizieren:
  - Aktive Dienste
  - System Architektur
  - Betriebssystem
- Die gewonnenen Informationen liefern evtl. Schwachstellen im Betriebssystem

# ICMP (RFC 972)



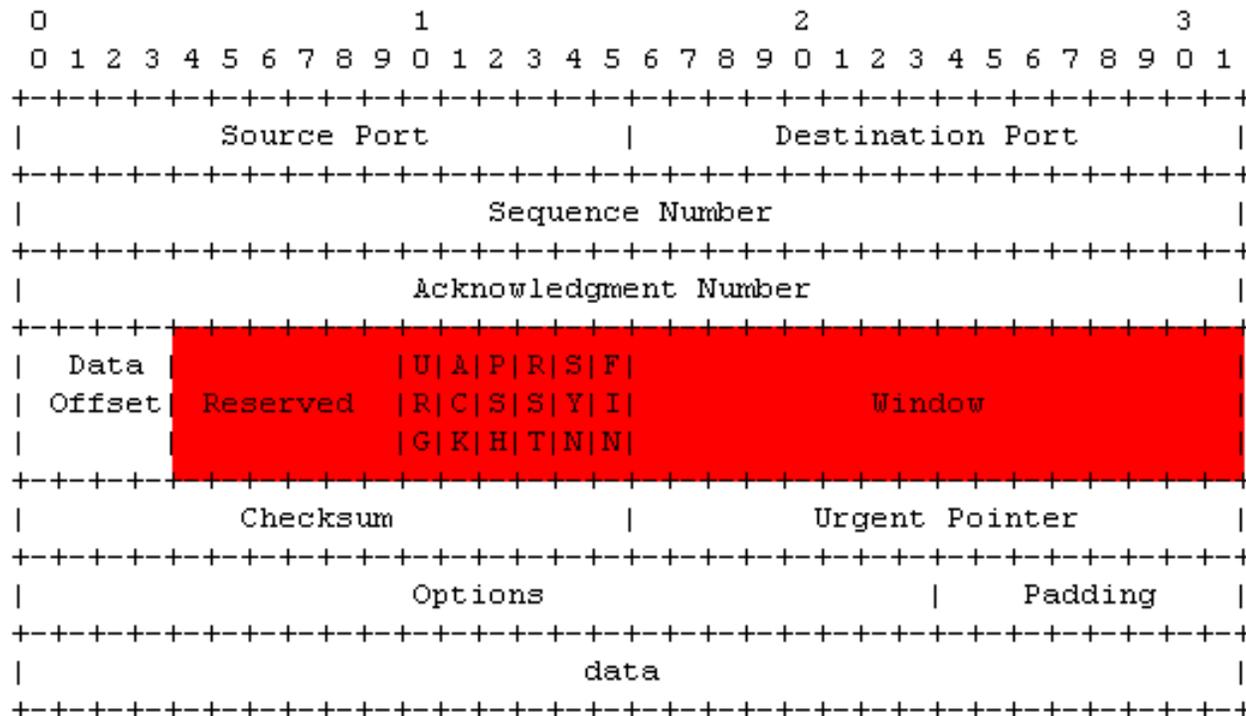
Verschiedene Typen z.B.

ICMP ECHO request (Typ 8)

ICMP ECHO reply (Typ 0)

# TCP (RFC 973)

- Flags



TCP Header Format

# TCP (RFC 973)

- “three-way handshake”

TCP A		TCP B
1. CLOSED		LISTEN
2. SYN-SENT	--> <SEQ=100><CTL=SYN>	--> SYN-RECEIVED
3. ESTABLISHED	<-- <SEQ=300><ACK=101><CTL=SYN, ACK>	<-- SYN-RECEIVED
4. ESTABLISHED	--> <SEQ=101><ACK=301><CTL=ACK>	--> ESTABLISHED
5. ESTABLISHED	--> <SEQ=101><ACK=301><CTL=ACK><DATA>	--> ESTABLISHED

Basic 3-Way Handshake for Connection Synchronization

# Ping Sweeps

- ICMP Sweeps (ICMP ECHO requests)
  - Testet ob Ziel IP erreichbar
- Broadcast ICMP
  - ICMP ECHO request an Broadcastadresse
  - Testet ganze Netze
- Non-ECHO ICMP
  - ICMP timestamp um aktuelle Zeit zu erfahren
  - ICMP address mask um Subnetmask zu erfahren

# Ping Sweeps

- UDP Sweeps (UDP Scans)
  - Basiert auf ICMP PORT UNREACHABLE
  - Nicht zuverlässig, weil:
    - Firewalls und Router oft UDP Pakete verwerfen
- Falls ICMP Pakete gefiltert werden, muß ein Angreifer alle IP Adressen eines Netzes Portscannen, was sehr zeitaufwändig ist

# Portscanning Übersicht

- Offenes TCP Scannen
  - TCP connect()
- Stealth TCP Scannen
  - Halb-offenes SYN Flag Scannen
  - Inverses TCP Flag Scannen
  - ACK Flag Probe Scannen
  - TCP Fragmentation Scannen

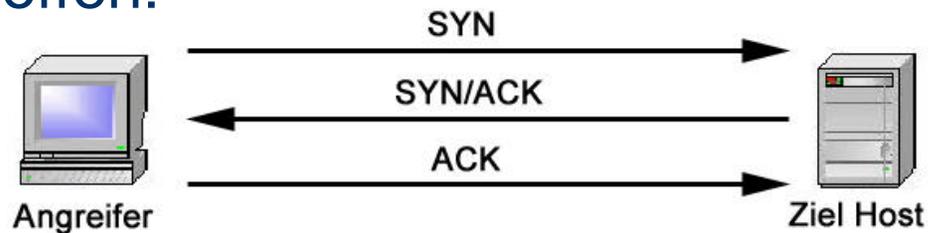
# Portscanning Übersicht

- Third-party und spoofed TCP Scannen
  - FTP bounce Scannen
  - Proxy bounce Scannen
  - Sniffer-based spoofed Scannen
  - IP ID Header scannen

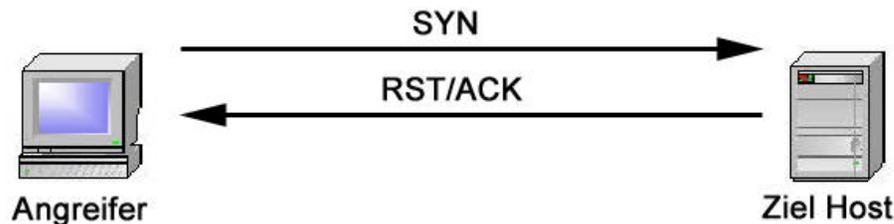
# Offenes TCP Scannen

- TCP connect()

Port offen:



Port geschlossen:

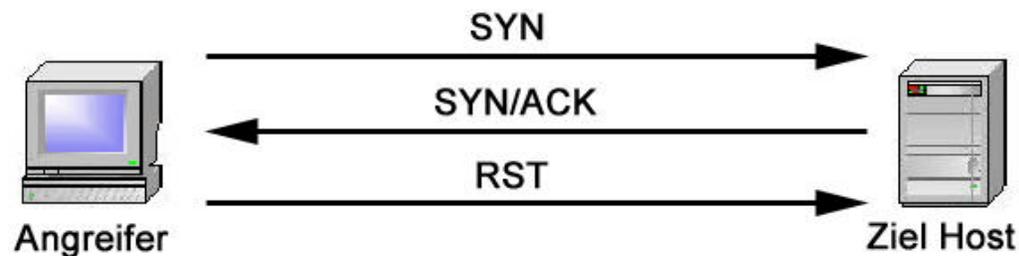


- Vorteil: Sehr genaue Ergebnisse
- Nachteil: Wird durch Logging erkannt

# Stealth TCP Scannen

- Halb-offenes SYN Flag Scannen

Port offen:



- Vorteil: geringere Gefahr durch Logging
- Nachteil: Nur als Super-User möglich, da RST Pakete „erstellt“ werden müssen

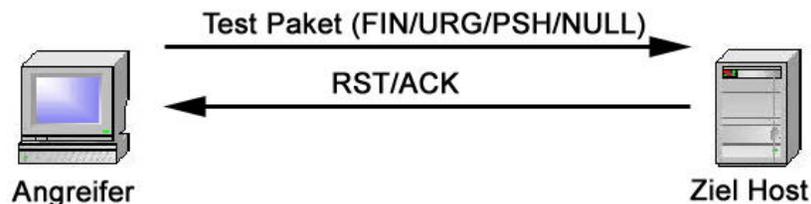
# Stealth TCP Scannen

- Inverses TCP Flag Scannen

Port offen:



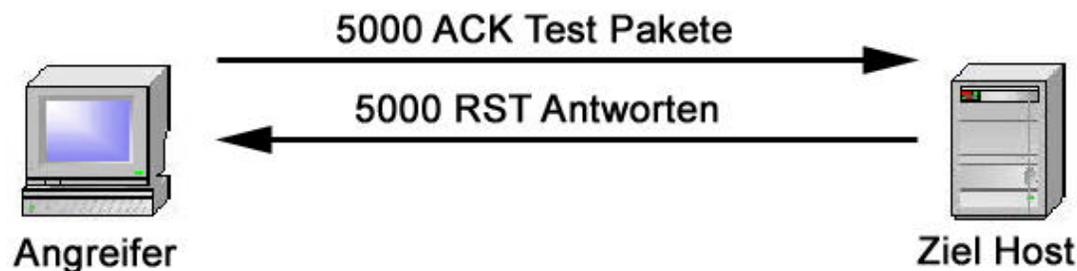
Port geschlossen:



- Vorteil: Schwer aufzuspüren
- Nachteil: Nicht möglich bei Windows Rechnern

# Stealth TCP Scannen

- ACK Flag Probe Scannen



- Analyse des TTL oder Window Feldes, der Pakete
- Die Werte unterscheiden sich bei offenem/geschlossenem Port
- Vorteil: Schwer aufzuspüren
- Nachteil: Zeitaufwändige Analyse und basiert auf Implementierungsfehler

# Port Scanning Techniken

- Motivation
- Exkurs: Intrusion Detection Systeme
- Verfahren
  - Slow Scan
  - Fragmentation Scan
  - Decoy
  - Coordinated Scans
- Gegenmaßnahmen

# Motivation

- Erkannte Scans können als Vorbereitung eines Angriffs erkannt werden
- Abwehrmaßnahmen und Maßnahmen gegen Angreifer sind nicht erwünscht
- Konsequenz: Scans bzw. Urheber müssen verschleiert werden

# Exkurs: Intrusion Detection Systeme

- Intrusion Detection Systeme (IDS) suchen in Echtzeit oder in Logfiles nach Mustern eines Angriffs
- Muster heißt z.B. auffällige Häufung von Anfragen in einem begrenzten Zeitraum

# Verfahren

- Slow Scan
  - nur wenige Pakete in langem Zeitraum ⇒ taucht nicht in Statistiken auf bzw. nicht in auswertbaren Umfängen
- Fragmentation Scanning
  - IP Pakete werden fragmentiert
  - Erstes Fragment enthält nur Source und Destination, keine Flags oder Payload
  - Paketfilter leiten ohne Überprüfung zum Ziel weiter

# Verfahren

- Decoy
  - Falsche Adressen werden übermittelt, zufällige TTL
  - IDS kann einen Angriff feststellen, nicht aber den Verursacher
- Coordinated Scans
  - Gruppe von Angreifern koordiniert Scan
  - In Kombination mit „Slow Scan“ sehr schwer erkennbar, da keine auswertbaren Muster auftreten.

# Gegenmaßnahmen

- Verfahren werden verwendet um Gegenmaßnahmen auszuhebeln
- Möglichkeit: Filter feiner einstellen
- Gefahr von „false positives“ steigt mit feineren Filtern

# Betriebssystemerkennung

- Motivation
- Verfahren
- Gegenmaßnahmen

# Motivation

- Viele Exploits funktionieren nur auf bestimmten Betriebssystemen oder auf verschiedenen Serverversionen
- IP Adressen werden auf „Vorrat“ gescannt für spätere Exploits

# Verfahren

- Banner Grabbing

- Server geben Betriebssystem in „Bannern“ an
  - ~> telnet ftp.rub.de
  - Trying 134.147.32.70...
  - Connected to sun218.rz.ruhr-uni-bochum.de
  - Escape character is '^]'.  
  
SunOS 5.8

# Verfahren

- Stack fingerprinting – der OS TCP/IP-Stack wird getestet
  - FIN Probe – FIN Paket wird an offenen Port gesandt, nach Standard darf nicht geantwortet werden
  - BOGUS Flag Test – undefinierter TCP-Flag im TCP Header eines SYN-Paketes

# Verfahren

- Sammeln von TCP Sequenznummern – verschiedene TCP/IP-Stacks verwenden verschiedene Verfahren
- Don't Fragment bit gesetzt oder nicht
- TCP Initial Window – überprüfen der „window size“ von Antwortpaketen

# Verfahren

- ICMP Error Message Quenching – wie wird die Wiederholungsrate der Error Messages begrenzt
- TCP Optionen – freiwillige Implementation, Anfrage mit bestimmten Argumenten  $\Rightarrow$  unterschiedliche Stacks reagieren verschieden

# Gegenmaßnahmen

- Banner abstellen oder fälschen
- Keine „verräterischen“ Host-Namen verwenden
- Application Proxies verwenden
- Langfristig: TCP/IP-Stacks stärker standardisieren
- Kurzfristig: Wenn möglich selbst Parameter des Stacks ändern

# Scannen durch eine Firewall

- Non-Echo-Request ICMP Pakete für Anfragen verwenden, diese werden von Firewalls oft nicht geblockt
- Ports von Firewalls mit verschiedenen Protokollen testen, um Regelsatz der Firewall nachzuvollziehen
- Anschließend kann man mit verschiedenen Verfahren die Struktur hinter der Firewall testen (Firewalking)

# NMAP - Quelle

- Nmap - DER Portscanner schlechthin, bietet eine Vielzahl von Optionen zum Scannen und eine recht zuverlässige Betriebssystemerkennung - [www.insecure.org](http://www.insecure.org) (unixoide Plattformen), [www.eeye.com/html/Research/Tools/nmapNT.html](http://www.eeye.com/html/Research/Tools/nmapNT.html) (Windows Plattformen)

# NMAP – Scan Typen

- NMAP unterstützt alle oben genannten Scanmodi; z.B. führt:
  - Nmap `-sU Ziel_IP` einen UDP Scan durch
  - Nmap `-sT Ziel_IP` einen TCP connect() aus
  - Nmap `-sS Ziel_IP` einen TCP SYN Scan aus
  - Nmap `-sP Ziel_IP` einen ICMP Sweep durch
  - Nmap `-sO Ziel_IP` einen IP Protocol Scan aus

# NMAP – Scan Typen

- Weitere unterstützte Scan Typen sind:
  - SYN Sweep
  - FIN/ACK Sweep
  - XMAS/NULL Flag Scanning
  - FTP Bouncing
  - Reverse Ident Scanning
  - Fragmentation Scanning
  - Decoy Scanning

# NMAP – Timing Optionen

- NMAP bietet verschiedene Timing Optionen, um:
  - Intrusion Detection Systeme zu hintergehen (Slow Scan)
  - Scanvorgänge in Bezug auf die vorliegende Netzwerk Infrastruktur zu optimieren (Bandbreite)
- Folgende Optionen sind möglich:
  - `-host_timeout`: Dauer, für die ein Host gescannt wird

# NMAP – Timing Optionen

- `-max_/min_/initial_rtt_timeout`: maximale/minimale/anfangs Zeit, die auf eine Antwort gewartet werden soll, bevor ein Timeout gemeldet wird
- `-max_parallelism`: maximale Anzahl paralleler Scans
- `-scan_delay`: Pause zwischen zwei Testpaketen
- `-T`: legt eine vordefinierte Timing Policy fest, die alle o.g. Parameter beinhaltet

# NMAP für den Administrator

- NMAP hilft nicht nur Angreifern
- Durch scannen des „eigenen“ Netzes kann ein Admin die Hosts finden, die von „außen“ sichtbar sind
- Die Konfiguration bzw. das Regelwerk einer Firewall können mit NMAP Scans überprüft werden
- Die Anfälligkeit gegenüber DoS Attacken kann geprüft werden

# NMAP Scan: Windows XP SP 1

```
xterm
mars# nmap -vv -s5 -O 192.168.2.16

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-06-05 17:50 CEST
Host 192.168.2.16 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.2.16 at 17:50
Adding open port 5800/tcp
Adding open port 139/tcp
Adding open port 445/tcp
Adding open port 5900/tcp
Adding open port 135/tcp
The SYN Stealth Scan took 0 seconds to scan 1623 ports.
For OSScan assuming that port 135 is open and port 1 is closed and neither are firewalld
Interesting ports on 192.168.2.16:
(The 1618 ports scanned but not shown below are in state: closed)
Port      State      Service
135/tcp   open       loc-srv
139/tcp   open       netbios-ssn
445/tcp   open       microsoft-ds
5800/tcp  open       vnc-http
5900/tcp  open       vnc
Remote operating system guess: Windows XP Pro SP1 or Windows 2000 SP3
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=A0F%IPID=I%TS=0)
T1(Resp=Y%DF=Y%M=FFF0%ACK=S++%Flags=AS%0ps=NNNNNT)
T2(Resp=Y%DF=N%M=0%ACK=S%Flags=AR%0ps=)
T3(Resp=Y%DF=Y%M=FFF0%ACK=S++%Flags=AS%0ps=NNNNNT)
T4(Resp=Y%DF=N%M=0%ACK=0%Flags=R%0ps=)
T5(Resp=Y%DF=N%M=0%ACK=S++%Flags=AR%0ps=)
T6(Resp=Y%DF=N%M=0%ACK=0%Flags=R%0ps=)
T7(Resp=Y%DF=N%M=0%ACK=S++%Flags=AR%0ps=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E)

TCP Sequence Prediction: Class=random positive increments
                        Difficulty=2575 (Medium)
TCP ISN Seq. Numbers: 79CA4C51 79CC7EA7 79CEC608 79D10DFD
IPID Sequence Generation: Incremental

Nmap run completed -- 1 IP address (1 host up) scanned in 3.244 seconds
mars#
```

# NMAP Scan: SMC Barricade DSL Router

```
xterm
nars# nmap -vvv -sS -o 192.168.2.1

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-06-05 18:15 CEST
Host (192.168.2.1) appears to be up ... good.
Initiating SYN Stealth Scan against (192.168.2.1) at 18:15
Adding open port 80/tcp
Adding open port 1900/tcp
Adding open port 1723/tcp
The SYN Stealth Scan took 1 second to scan 1623 ports.
For OSScan assuming that port 80 is open and port 1 is closed and neither are firewalled
Interesting ports on (192.168.2.1):
<The 1620 ports scanned but not shown below are in state: closed>
Port      State  Service
80/tcp    open   http
1723/tcp  open   pptp
1900/tcp  open   UPnP
Remote operating system guess: SMC Barricade DSL Router/Modem/Wireless AP
OS fingerprint:
TSeq(Class=TD%gcd=2A%SI=0%IPID=I%TS=U)
T1(Resp=Y%DF=Y%M=1770%ACK=S++%Flags=AS%Ops=ME)
T2(Resp=N)
T3(Resp=Y%DF=Y%M=1770%ACK=S++%Flags=AS%Ops=ME)
T4(Resp=Y%DF=N%M=0%ACK=0%Flags=R%Ops=)
T5(Resp=Y%DF=N%M=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%M=0%ACK=0%Flags=R%Ops=)
T7(Resp=Y%DF=N%M=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=E%ULEN=134%DAT=E)

TCP Sequence Prediction: Class=trivial time dependency
                        Difficulty=0 (Trivial joke)
TCP ISN Seq. Numbers: 4BC2 4BEC 4C16 4C40
IPID Sequence Generation: Incremental

Nmap run completed -- 1 IP address (1 host up) scanned in 7.483 seconds
nars#
```

# NMAP Scan: Mac OS X

```
xterm
mars# nmap -vv -sS -O 192.168.2.3

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-06-05 18:19 CEST
Host 192.168.2.3 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.2.3 at 18:19
Adding open port 427/tcp
Adding open port 548/tcp
Adding open port 6000/tcp
Bumping up senddelay by 10000 (to 10000), due to excessive drops
The SYN Stealth Scan took 30 seconds to scan 1623 ports.
For OSscan assuming that port 427 is open and port 1 is closed and neither are firew
lled
Interesting ports on 192.168.2.3:
(The 1620 ports scanned but not shown below are in state: closed)
Port      State      Service
427/tcp   open       svrloc
548/tcp   open       afpovertcp
6000/tcp  open       X11
Remote OS guesses: Mac OS X 10.1 - 10.1.4, Mac OS X 10.1.5-10.2.3
OS Fingerprint:
TSeq(Class=TR%IPID=I%TS=2H2)
T1(Resp=Y%DF=Y%M=807A%ACK=S++%Flags=AS%Ops=MNMNNT)
T2(Resp=N)
T3(Resp=Y%DF=Y%M=807A%ACK=S++%Flags=AS%Ops=MNMNNT)
T4(Resp=Y%DF=N%M=0%ACK=0%Flags=R%Ops=)
T5(Resp=Y%DF=N%M=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%M=0%ACK=0%Flags=R%Ops=)
T7(Resp=Y%DF=N%M=0%ACK=S%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=0%ULEN=134%DAT=E)

TCP Sequence Prediction: Class=truly random
                        Difficulty=9999999 (Good luck!)
TCP ISN Seq. Numbers: D5F6B081 E3E344C9 62230DD5 F4E31C69
IPID Sequence Generation: Incremental

Nmap run completed -- 1 IP address (1 host up) scanned in 36.197 seconds
mars# █
```

# NMAP Scan: FreeBSD 5.1

```
xterm
mars# nmap -vvv -sS -O 192.168.2.2

Starting nmap 3.27 ( www.insecure.org/nmap/ ) at 2003-06-05 18:16 CEST
Host mars.planet (192.168.2.2) appears to be up ... good.
Initiating SYN Stealth Scan against mars.planet (192.168.2.2) at 18:16
Adding open port 2049/tcp
Adding open port 22/tcp
Adding open port 901/tcp
Adding open port 1023/tcp
Adding open port 139/tcp
Adding open port 111/tcp
The SYN Stealth Scan took 8 seconds to scan 1623 ports.
For OSScan assuming that port 22 is open and port 1 is closed and neither are firewalled
Insufficient responses for TCP sequencing (3), OS detection may be less accurate
Interesting ports on mars.planet (192.168.2.2):
<The 1617 ports scanned but not shown below are in state: closed>
Port      State      Service
22/tcp    open      ssh
111/tcp   open      sunrpc
139/tcp   open      netbios-ssn
901/tcp   open      samba-swat
1023/tcp  open      netvenuechat
2049/tcp  open      nfs
Remote OS guesses: Mac OS X 10.1.4 (Darwin Kernel 5.4) on iMac, Mac OS X 10.1.5, FreeBSD
4.3 - 4.4PRERELEASE, FreeBSD 5.0-CURRENT (Jan 2003), FreeBSD 5.0-RELEASE (x86)
OS Fingerprint:
T1(Resp=Y%DF=Y%M=FFFF%ACK=S++%Flags=RS%Ops=MNNNNT)
T2(Resp=N)
T3(Resp=Y%DF=Y%M=FFFF%ACK=S++%Flags=RS%Ops=MNNNNT)
T4(Resp=Y%DF=N%M=0%ACK=0%Flags=R%Ops=)
T5(Resp=Y%DF=N%M=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%M=0%ACK=0%Flags=R%Ops=)
T7(Resp=Y%DF=N%M=0%ACK=S%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E%UCK=0%ULEN=134%DAT=E)

Uptime 0.111 days (since Thu Jun 5 15:36:38 2003)
IPID Sequence Generation: Incremental

Nmap run completed -- 1 IP address (1 host up) scanned in 14.550 seconds
mars#
```

# Fazit

- Scannen von Netzwerken kann verschiedenen Zwecken dienen: defensiv, zum Absichern des eigenen Systems, oder offensiv, zum Ausspähen eines Ziels für Cracker
- Scanner sind Alltag geworden, sowohl automatisierte, als auch manuelle Varianten
- Kenntnisse von Verfahren und Funktionen wichtig, um Sicherheit der eigenen Hosts zu gewährleisten