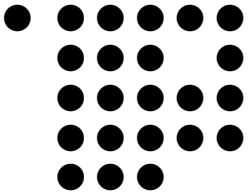


Fachhochschule Köln
09 Fachbereich Nachrichtentechnik

Institut für Datennetze

STUDIENARBEIT

Thema: Portscanner und Fingerprinting

WS 2001/2002

Studenten: Gruppe **M**

Victor Rodrigues (Matr.Nr.: 11001363)
Lutz Moppert (Matr.Nr.: 11011071)

Abgabetermin: 28. Januar 2002

Inhaltsverzeichnis

1 Übersicht

Ein Portscann dient dazu, Aufschluss über unbekannte Netzwerke und den beteiligten Maschinen zu geben. Mit seiner Hilfe kann man herausfinden, welche Aufgaben die einzelnen Rechner im Netzwerk haben, welche Dienste auf ihnen laufen und was für ein Betriebssystem benutzt wird. Man kann feststellen, ob man unter einer bestimmten IP-Adresse einen Drucker, einen normalen PC, einen Server oder sonstige Geräte findet.

Portscanner wurden zunächst vor allem von Hackern / Crackern benutzt und entwickelt, um die Schwachstellen eines Netzes aufzuspüren. Mit diesen Informationen kann man dann Lücke im Sicherheitssystem finden und mit Ihrer Hilfe Zugang zu fremden Netzen erlangen. Mitlerweile gehört die Verwendung eines Portscanners jedoch auch zum Standart Werkzeug eines jeden System Administrators, um das eigene Netzwerk auf offene Ports und damit auf mögliche Sicherheitslücken zu testen.

1.1 Vorgehensweise

Zunächst haben wir uns einen Überblick über die verbreitetsten Portscanner verschafft, um ein paar, für unsere Zwecke geeignete, herauszufinden. Die Suche beschränkte sich vornehmlich auf das Internet. Da Portscanner weit verbreitet sind, hatten wir keine Schwierigkeiten einige herunter zu laden.

Während der eigentlichen Testphase haben wir dann die in Frage kommenden Scanner installiert und getestet (da die Installation in allen Fällen problemlos verlief, werden wir innerhalb dieser Studienarbeit nicht näher auf die Details eingehen). Als Ziel-Netzwerk diente uns vor allem das LAN der FH Köln, da ein umfangreicher Scan eine recht hohe Netzlast produziert und von vielen nicht gern gesehen wird!

Tabelle 1: Portscanner

Scanner	**Preis**	**Art**	**Beschreibung**
nmapNT	Free	CMD	portiert von UNIX, basiert auf nesus und hat sehr mächtige Parametersteuerung
IP Scan	Free	CMD	Scannt nur jeweils einen Port!
wnuke4	Free	CMD	Scannt Ports sehr flexibel und bietet verschiedene Angriffstools ⇒ eher für Cracker geeignet
SuperScan	Free	GUI	Scannner mit sehr flexiblen Konfigurations-Möglichkeiten
7th Sphere	Free	GUI	Toolsammlung mit einem einfachen Portscanner, ein Client aber verschiedene Ports
IP Ultra Scan	Free	GUI	Scant nur jeweils 10 Ports und einen Client, umständliche Bedienung
Fast Scan	Free	GUI	Simpler Portscanner, scant nur einen Client
WS Ping Pro Pack	Share	GUI	Toolsammlung mit einem einfachem aber variablem Portscanner
DinglerScan	Share	GUI	Reiner Portscanner, scant nur einen Client aber variable Ports
Network Port Scan	Share	GUI	Portscanner mit weiteren Testtools
NetScan Tools	Share	GUI	Sehr umfangreiche Toolsammlung mit variablen Scanparametern

Zu letzt haben wir dann einen ausführlichen Scan auf das Subnetz 139.6.17.xxx gemacht. Das

Ergebniss dieses Scans, ist im Anhang zu sehen. Wir sind dabei auf viele verschiedene Geräte gestoßen (siehe Remote OS guess). Neben diversen Windows, Solaris und Linux Rechnern gab es einen Switch, einen Router, verschiedenen Drucker und sogar einen Windows For Workgroups 3.11 - Client.

1.2 Verschiedene Portscanner

Tabelle 1 zeigt eine Auflistung, der von uns getesteten Portscanner. Die Spalte *Preis* zeigt, ob es sich um *Free*ware oder um *Share*ware Produkte handelt. Die Spalte *Art* beschreibt, ob die Scanner eine grafische Benutzerschnittstelle besitzen (GUI) oder ob es Kommandozeilen Programme sind (CMD).

Eine sehr gute Möglichkeit die eigene Angreifbarkeit zu testen, bieten die verschiedenen Online Portscanner (siehe Tabelle 2). Bei manchen ist jedoch eine vorherige Anmeldung erforderlich.

Tabelle 2: Online Portscanner

Scanner	**URL**	**Anmeldung**
Hacker Whacker	http://www.hackerwhacker.com/	Ja
Sygate Online Service	http://scan.sygatetech.com/	Nein
GRC Online Scan	http://grc.com/x/ne.dll?bh0bkyd2	Nein
DSL Scan	http://www.dslreports.com/secureme	Ja

1.3 Auswahlkriterien

Um die lange Liste an Portscanner kürzen zu können, haben wir folgende Kriterien angesetzt, die ein Portscanner in unseren Augen erfüllen sollte:

- Es sollte sich nach Möglichkeit um ein Freeware-Tool handeln. Zwar haben die Shareware Scanner meist einen größeren Funktionsumfang, jedoch ist die Testzeit zu kurz, um sich in die damit verbundene komplexere Bedienung einzuarbeiten.
- Der Portscanner sollte in der Lage sein, mehrere Clients bzw. ein ganzes Subnet zu scannen.
- Die Auswahl der zu scannenden Ports sollte möglichst komfortabel sein (Scanner, die nur jeweils eine begrenzte Anzahl Port scannen, scheiden aus).
- Eine große Auswahl weiterer Funktionen, die über das Scannen hinaus gehen, machen einen Scanner in unseren Augen eher unübersichtlich und erschweren oft das Verständnis und die Bedienung.

Der universellste aller getesteten Scanner ist sicherlich nmapNT. Trotz, oder gerade wegen der Bedienung via Kommandozeilen, ist er unser Favorit geworden. Unter den „grafischen“ Scanner hat uns SuperScan am besten gefallen, weil er die komfortabelsten Konfigurationsmöglichkeiten bietet.

1.4 Die wichtigsten Ports

Um eintreffende Daten an den richtigen Dienst der Anwendungsschicht weiterleiten zu können, benutzen sowohl TCP wie auch UDP eine Port-Adresse. Aus der Definition des

Tabelle 3: Auswahl bekannter Ports

Port	Dienst	Port	Dienst
5	RJE (Remote Job Entry)	119	NNTP Newsgroup
7	ECHO	137	NetBIOS Name Service
20	FTP-Data	139	NetBIOS Datagram Service
21	FTP	143	IMAP (Interim Mail Access P.)
23	TelNet	150	NetBIOS Session Service
25	SMTPb	156	SQL Server
37	Time	161	SNMP
42	Host Name Server	179	BGP (Border Gateway P.)
43	WhoIs	190	GACP (Gateway Access Control P.)
49	Login Host Protocol	194	IRC (Internet Relay Chat)
53	DNS	197	DLS Directory Location Service
69	TFTP (Trivial FTP)	389	LDAP
70	Gopher Services	396	Novell Netware over IP
79	Finger	443	HTTPS
80	HTTP (WWW)	444	SNPP (Simple Network Paging P.)
103	X.400 standard	458	Apple Quicktime
108	SNA (Gateway Access Server)	546	DHCP Client
109	POP2 (Post Office P. 2)	547	DHCP Server
110	POP3 (Post Office P. 3)	563	SNEWS
115	SFTP (Simple FTP)	565	whoami
118	SQLserver	569	MSN

TCP- bzw. UDP-Headers, wo 16 Bit für die Portnummer reserviert werden, ergeben sich 65536 adressierbare Ports (siehe auch RFC 793 und RFC 768), von denen aber i.A. nur ein paar wenige wirklich interessant sind. Die Ports teilen sich in drei Bereiche:

1. Die „Well Known Ports“ sind die von 0 bis 1023
2. Die „Registered Ports“ sind die von 1024 bis 49151
3. Die „Dynamic/Private Ports“ sind die von 49152 bis 65535

Diese Ports können entweder über TCP oder über UDP angesprochen werden, wobei das scannen von UDP Ports nicht immer funktioniert (siehe auch Kapitel 2.1). Tabelle 3 zeigt die wichtigsten Ports. Eine vollständige Liste mit den allgemein üblichen Belegungen der einzelnen Ports gibt es auf:

http://www.isi.edu/in-notes/iana/assignments/port-numbers

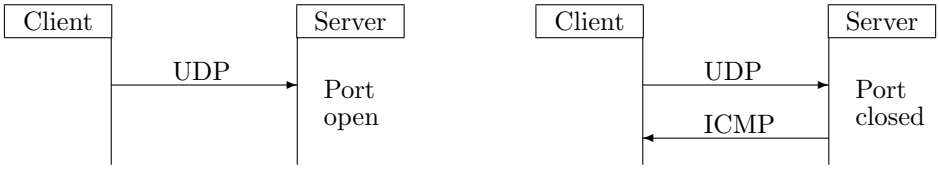
2 Die Techniken der Portscanner

Im Folgenden werden die verschiedenen Techniken zum Scannen von Ports beschrieben und mit kurzen (skizzierten und daher unvollständigen) Strichdiagrammen erläutert. Ausserdem geben wir zu jeder Art von Portscan die entsprechende Option für nmapNT an, da dieser Scanner (als Einziger) alle angegebenen Methoden beherrscht.

2.1 UDP Portscan

Bei einem UDP-Portscan (nmapNT Option „-sU“) wird ein UDP-Paket zum Ziel-Port übertragen. Wenn der Ziel-Port mit der Nachricht „ICMP Port unreachable“ den Erhalt

quittiert, ist der Port inaktiv und somit geschlossen. Falls diese ICMP-Nachricht nicht an den scannenden Clienten zurückgeschickt wird, kann eventuell davon ausgegangen werden, dass der Port offen ist.



Das Scannen von UDP Ports ist im Allgemeinen sehr schwierig und langsam. Da das UDP-Protokoll ein verbindungsloses Protokoll ist und es keine Fehlerkontrolle gibt, kann man nicht direkt feststellen, ob ein Port offen ist oder nicht. Die meisten Hosts senden jedoch eine „ICMP Port unreachable“ Fehlermeldung zurück, wenn man versucht ein Paket an einen inaktiven Port zu senden. Mit diesem Wissen lassen sich offene Ports also indirekt bestimmen. Wenn man keine Fehlermeldung bekommt, kann dies die verschiedensten Gründe haben:

1. Der Host antwortet generell nicht mit einer Fehlermeldung.
2. Die maximale Anzahl ICMP-Meldungen wurde erreicht.
3. Das UDP-Paket ist unterwegs verloren gegangen.
4. Die ICMP Antwort ist unterwegs verloren gegangen.
5. Der entsprechende Port ist offen.

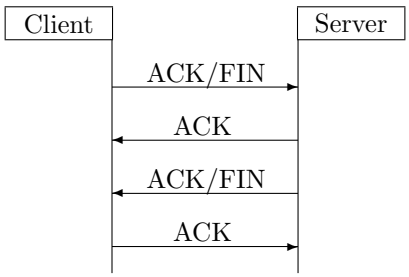
Den Fall 1. kann man von vornherein ausschließen, wenn man zunächst das Betriebssystem des Hosts verifiziert (geeignete Methoden werden in Abschnitt 3.1 beschrieben). Betriebssysteme mit korrekter UDP Implementation übermitteln die Fehlermeldung, allerdings ist deren Anzahl oftmals begrenzt. Ein Linux Kernel beispielsweise, begrenzt die Zahl der ICMP-Antworten auf 80 pro vier Sekunden (Fall 2.) und wartet anschliessend eine viertel Sekunde, bevor wieder Fehlermeldungen gesendet werden. Dies führt dann zu einem verfälschten Ergebnis. Einige Scanner versuchen dieses Verhalten durch geeignete Scannverfahren zu umgehen.

Ein falsches Ergebnis kann auch durch hohe Netzlasten zustande kommen (Fälle 3. und 4.) und diese sind vom Scanner nur sehr schwer abzufangen. Gerade ein UDP-scan via Internet ist deshalb recht schwierig.

Um sicher zu sein, dass Fall 5. eingetreten ist, muss ein Portscanner die Pakete u.U. wiederholt schicken und gelegentlich eine Pause einlegen. Je zuverlässiger ein Scan werden soll, desto mehr Zeit muß man einkalkulieren.

2.2 TCP Portscan

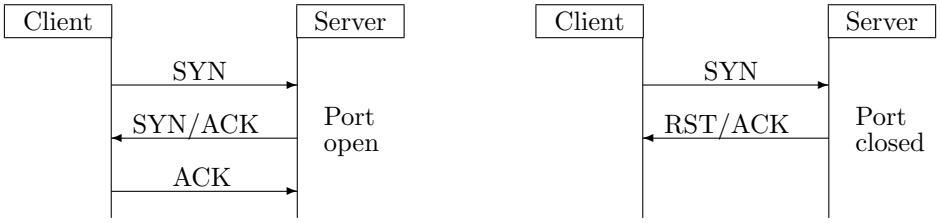
Das TCP Protokoll ist im Gegensatz zum UDP Protokoll verbindungsorientiert. Vor der eigentlichen Datenübertragung findet deshalb ein drei Wege Handshake statt (siehe auch Strichdiagramm weiter unten). Erst nach der erfolgreichen Verbindung werden Daten übertragen und abschließend wird durch vier weitere Meldungen die Verbindung wieder getrennt:



TCP bietet ausserdem Fehlerkontroll-Mechanismen, die einem das Scannen erleichtern und unauffälligere Scanmethoden erlauben. Die nun folgenden Kapitel erläutern die verschiedenen Scanmethoden im einzelnen.

2.2.1 TCP Connect Scanning

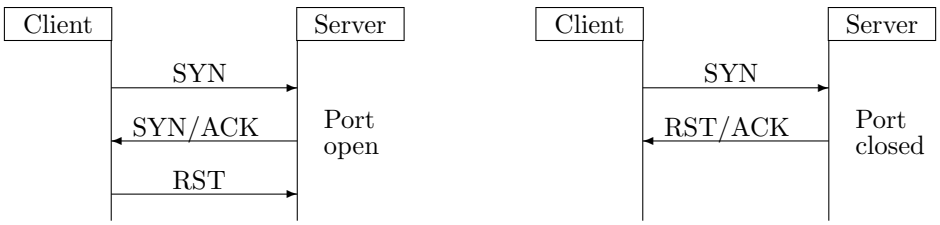
Die einfachste Art des Portscannens ist der vollständige TCP Connect Scan (nmapNT Option „-sT“, default Einstellung), bei dem eine echte Verbindung zum Ziel-Port zustande kommt. Die Prozedur des 3-Wege-Handshakes von TCP-Verbindungen wird dabei komplett abgearbeitet. Der Client überträgt also zunächst ein SYN-Paket an den Server, dieser quittiert den Empfang mit einem SYN-/ACK-Paket und wartet dann auf eine weitere Bestätigung von Seiten des Clients durch ein ACK-Paket:



Diese Scanmethode ist die auffälligste und langsamste, kann jedoch ohne Administrationsrechte durchgeführt werden. Auf der Serverseite kann ein solcher Scan leicht festgestellt und protokolliert werden.

2.2.2 SYN Scanning (half open scan)

Bei einem „halb offenen“ Scan (nmapNT Option „-sS“) wird keine vollständige Verbindung mit dem Server hergestellt. Es wird zunächst wieder ein SYN Paket zum Server geschickt, wenn dieser mit SYN/ACK antwortet handelt es sich i.A. um einen offenen Port und die noch nicht vollständige Verbindung wird vom Client sofort durch senden eines RST/ACK-Paketes unterbrochen. Sendet der Server ein RST/ACK zurück, so ist der Port geschlossen und die Verbindung ebenfalls beendet:



Diese Art des Scannens wird nicht von allen Systemen protokolliert und ist daher etwas unauffälliger.

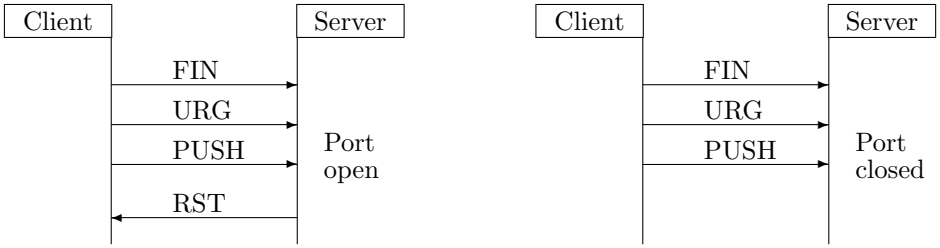
2.2.3 FIN Scanning (stealth scan)

Beim Stealth FIN Scan (nmapNT Option „-sF“) wird nur ein FIN Paket an den zu scannenden Port gesendet, nach RFC 793 (TCP) müsste der Server ein RST senden, wenn es sich um einen geschlossenen Port handelt, ansonsten sollte das Paket ignoriert werden.

Diese Scanmethode, für die man Administrationsrechte benötigt, wird sehr selten protokolliert und nennt sich daher auch „stealth scan“ also „heimlicher Scan“. Sie funktioniert jedoch nicht immer, da sich nicht alle TCP Implementationen an die Forderungen des RFC's halten (siehe Kapitel 4).

2.2.4 Xmas-Portscan

Der Xmas Scan (nmapNT Option „-sX“) funktioniert analog zum FIN Scan, jedoch wird hierbei ein FIN-, ein URG- und ein PUSH-Paket zum Ziel-Port geschickt. Nach den Empfehlungen von RFC 792 (ICMP) müsste der Server ein RST für alle geschlossenen Ports zurückgeben. Auch hier hängt die korrekte Antwort wieder von der richtigen Implementierung auf dem Server ab (siehe Kapitel 4).



2.2.5 TCP-Null-Portscan

Beim TCP-Null-Scan (nmapNT Option „-sN“) werden alle Flags (URG, ACK, PSH, RST, SYN und FIN) auf Null gesetzt und laut RFC 793 (TCP) müsste das Ziel-System einen RST für alle geschlossenen Ports zurückschicken. Es gelten die gleichen Einschränkungen wie bei den beiden zuvor genannten Methoden.

2.3 Weitere Portscans

Es gibt noch ein paar weitere Typen von Portscans, die jedoch unter Windows 2000 nicht unterstüzt werden und die wir daher auch nicht getestet haben. Dazu gehört z.B. der IP

Protocol Scan. Er wird verwendet, um herauszufinden, welche IP Protokolle auf einem Host verwendet werden. Dabei werden „nakte“ IP Pakete (also Pakete, die zwar ein Protokoll addressieren aber keine weiteren Daten für dieses Protokoll enthalten) an den Host gesendet. Wenn dieser mit einer ICMP - Port unreachable Message antwortet, ist der Port geschlossen, sonst ist er womöglich offen. Es ergeben sich bei dieser Technik die gleichen Probleme wie beim UDP Port Scan (siehe Kapitel 2.1). Weitere Scans, wie der ACK Scan oder der Window Scan, helfen einem, vorhandene Firewalls zu erkennen und deren Einstellung näher zu bestimmen.

2.4 Erweiterte Scanoptionen

Die erweiterten Scanoption sollen das Entdecken eines Portscans zusätzlich erschweren. Da der Zeitaufwand für einen eingehenden Test dieser Funktionen zu hoch wäre, werden diese hier nur kurz erwähnt. Eine ausführliche Beschreibung enthält die Man-Page zu nmap (englisch). In dieser werden auch die bisher beschriebenen Scanoptionen nocheinmal genau erläutert.

2.4.1 Fragmentation Scanning

Um einen Paket Filter zu umgehen, kann es manchmal sinnvoll sein, die Pakete beim Scannen in sehr kleine IP Pakete packen zu lassen (nmapNT Option „-f“), so wird sogar der TCP-Header auf mehrere Pakete verteilt. Ein Paket Filter kann einen solchen Scan nur entdecken, wenn er die Pakete zwischenspeichert, auspackt und zusammensetzt.

2.4.2 Reverse Ident Scannning

Der Ident-Dienst liefert eine zusätzliche Informationsebene über aktive Ports, nämlich den Benutzernamen. Dieser ist für manche Angriffe wichtig, da er verrät, welche Zugriffsrechte der Server-Dienst auf dem jeweiligen Port besitzt. Dieser Dienst läßt sich jedoch nur bei einem Scan mit vollständiger Verbindung durchführen, also z.B. dem TCP Connect Scan und funktioniert nur, wenn der Ident Dienst auf dem Zielsystem läuft.

2.4.3 FTP Bounce Attack

Hier wird die eingebaute Proxy-Funktion von RFC-konformen FTP-Servern benutzt, um die eigene Identität zu verstecken. Unter Ausnutzung dieser Sicherheitslücke denkt ein gescanntes System, der übernommene FTP-Server würde diese Scans durchführen. Verweigert die Firewall danach Pakete dieses Servers, so kann der Angreifer sein Glück einfach über einen anderen FTP-Server versuchen.

2.4.4 RPC Scan

Mit diesem Scan kann man die gefundenen offenen Ports auf RPC Funktionen testen. Ein Scanner testet hierbei, ob der offene Port auf eine RPC Anfrage antwortet und versucht herauszufinden, was für ein Dienst zu erreichen ist.

3 Anwendungsbeispiele von Portscannern

Unbekanntes Netzwerk als Black Box (am Beispiel der FH-Köln). Anhand der laufenden Dienste kann man die Funktion eines Rechners innerhalb eines Netzwerks bestimmen.

3.1 Betriebssysteme erkennen

Die nmapNT Option „-O“ veranlasst nmap, einen Tip abzugeben, welches Betriebssystem auf einer Maschine verwendet wird. Wie bereits erwähnt haben wir diese Option auf das FH-Netz 139.6.17.* getestet und viele verschiedene Betriebssysteme gefunden.

Nmap sucht hierfür beim Scannen nach eigenheiten, die auf ein bestimmtes Betriebssystem deuten. Beispiele hierfür finden sich unter anderem im Kapitel 4. Eine sehr genaue Beschreibung, welche Methoden nmapNT benutzt, ist in einer Dokumentation des nmap Autors zu finden. Diese wurde unter anderem auch ins deutsche übersetzt und liegt im Internet zum Download bereit:

http://www.insecure.org/nmap/nmap-fingerprinting-article-de.html

3.2 Verwendung der Ergebnisse

Hat man erst einmal die Funktion der einzelnen Maschinen, deren Betriebssysteme und die angebotenen Dienste bestimmt, so kann man versuchen, diese auf bekannte Sicherheitslücken zu überprüfen.

4 Windows spezifische Probleme

- Wenn Windows 9x-Systeme gescant werden, liefert das SYN Scanning nur ein stark verfälschtes Bild der Ports! Leider stand uns im Labor Netz kein solches System zur Verfügung und wir müssen uns auf die Angaben in der Literatur verlassen.
- Bei Windows funktionieren generell keine FIN, Xmas und Null Scans! Dies ist auf eine falsche Implementierung von TCP zurückzuführen. Entgegen den Empfehlungen von RFC 793 sollte ein offener Port mit einem RST Antworten, während ein geschlossener Port das Paket verwerfen und gar nicht antworten sollte. In Windows Systemen (und noch ein paar anderen) ist diese Empfehlung nicht aufgenommen worden. Ein Windows Port schickt *immer* ein RST zurück, egal ob er offen oder geschlossen ist.

5 Anhänge

Als sehr nützliche Informationsquellen haben sie die im Anhang befindlichen Dokumentationen erwiesen. Die Texte stehen in der Version, die wir beim schreiben dieser Arbeit verwendet haben, zur Ansicht zur Verfügung. Es sind im einzelnen:

- `nmapNT-man.txt`: Die „Man-Page“ zu nmapT vom Entwickler von nmap.
- `OSFingerprinting.txt`: Das vom Entwickler von nmap geschriebene und von Stefan Maly ins Deutsche übersetzte „Manual“ zum Fingerprinting.
- `port-numbers.txt`: Eine Liste der häufigsten Portbelegungen.