Telnet & UNIX Basic Commands

Dieser Text ist für Anfänger gedacht, die noch nicht mit telnet gearbeitet haben. Es soll eine Einstiegshilfe sein, damit man sich erstmal auf einem anderen Rechner einloggen kann und ihn dann etwas zu erforschen...
Hinzu kommen noch die wichtigsten UNIX-Befehle, damit man mit dem terminal auch etwas anfangen kann, sonst wärs ja ziemlich sinnlos :))

Ach ja, dieses Dokument ist mit Sicherheit nicht 100% fehlerfrei (zurückzuführen auf den geistigen Zustand des Authors 8)), falls ihr also welche findet, Fragen habt, Beleidigungen loswerden wollt, Anregungen für andere Tuts habt, mailt mir unter Razor99@GMX.DE

Inhalt

Benutzung von Telnet

Ihr startet telnet mit:

```
telnet <hostname> <port>
```

Als <hostname> gebt ihr einfach den Rechnernamen oder die IP des Zielrechners ein.
Der <port> hängt, von dem Service ab, den ihr benutzen wollt (SMTP=25;FINGER=79;...)
Dazu ist am Ende des Textes eine Auflistung der known-ports, die einigermassen festgelegt sind, und hinter denen auf den meisten Rechnern auch diesselben Programme stehen.
Ist der angegebene Port auf dem Zielrechner offen, werdet ihr mit ihm verbunden, und habt je nach gewähltem Port verschiedene Möglichkeiten.

Es hat zwar nicht viel Sinn, mit Telnet Mails zu verschicken (ausser für Fake-Mails), aber um mal ein Gefühl für Telnet zu bekommen :)) ist es schon mal ganz hilfreich. Ihr verbindet euch mit dem Mailserver über Port 25.

Z.B. : telnet smtp.mail.net 25

Ihr solltet daraufhin ein Terminal erhalten, das euch mit einer entsprechenden Meldung begrüßt. Meist wird dort der Name und die Version des mailservers angegeben. Ihr müsst euch nun mit einem 'helo' anmelden. Dabei müsst ihr einen existierenden Host angeben. Ist jetzt eigentlich ziemlich uninteressant geworden, da die meisten Mailserver heutzutage automatisch die IP des Benutzers als Domain benutzen, egal welche Domain man angegeben hat.

z.B. : HELO AOL.COM

Dann müsst ihr angeben von wem die Mail ist. Dies geschieht mit 'MAIL FROM:'

z.B. : MAIL FROM:RAZOR99@GMX.DE

Hierbei könnt ihr auch irgendeine beliebe E-Mail-Adresse angeben, wenn ihr den Empfänger glauben machen will, das sie von jemand anderem stammt. Das Problem ist jedoch, das die eigene IP-Adresse auf jeden Fall mitgesendet wird (ausser auf alten Servern), aber um kleine Fake-Mails zu schreiben, ist es mal ganz nützlich.

Nun noch an wen die Mail ist.

z.B. : RCPT TO:RAZOR99@GMX.DE (:))

Um nun den Inhalt der Mail zu schreiben gebt ihr

DATA

ein. Nun könnt ihr den Text schreiben, der in die Mail kommen soll. Wenn ihr fertig damit seid, macht ihr in einer neuen Zeile einen Punkt (Nur Einen), und die Mail wird abgeschickt.

Andere Dienste (FTP, Finger, ...)

FTP:

Über Port 21 könnt ihr auch, falls vorhanden, auf einen FTP-Server zugreifen. Bei manchen kann man sich als anonymous anmelden. Dann muss man als Passwort seine e-mail-adresse eingeben. Mit HELP kann man sich jederzeit Hilfe zu den einzelnen Befehlen ausgeben lassen, daher werde ich jetzt nicht auf alle eingehen (keine Zeit :))

USER <Username> (z.B. USER anonymous / USER root)

PASS <Passwort> (z.B. PASS razor99@gmx.de / PASS root ;)

ECHO:

Über Port 7 kommt ihr an ECHO. Hier bekommt ihr einfach jede Taste die ihr drückt, als ECHO zurückgesendet (zum Testen der Verbindung)

CHARGEN:

Auf diesem Port werden zufällige Zeichen gesendet (auch zum Testen)

FINGER:

Über Port 79 könnt ihr den Service Finger benutzen. Ist dieser auf dem anderen Rechner aktiv, bekommt ihr die derzeitig angemeldeten User ausgegeben...

SSH:

Dies ist eine Secure Shell, ähnlich einem Login per Telnet, jedoch mit einer sichereren (was für ein Wort) Überprüfung, wer sich da einloggen will. Hierfür sollte man sich am besten mal die Manpages durchlesen, oder einen anderen Text im I-Net suchen :))

Über den Telnetport auf einem anderen Rechner einloggen

Mithilfe von telnet ist es möglich, sich bei anderen Rechnern einzuloggen. Auf den offenen Ports kann man meist bestimmte Services benutzen. Bei vielen Unix-Rechnern kann man z.B. über den SMTP-Port (25) dann Mails verschicken. Die Ports unter 1024 sind die well known-ports, und für bestimmte programme zugeteilt. Das heißt natürlich nicht, dass hinter dem jeweiligen Port auch immer dasselbe Programm stehen muss, aber in den meisten Fällen kann man davon ausgehen. Am Ende des Textes sind die bekanntesten Ports aufgeführt.

Es ist meist ratsamer erstmal die wichtigsten Ports per telnet auszuprobieren, als direkt einen Scan durchzuführen, der weitaus auffälliger ist als eine Telnet-Anfrage !!!

Am interessantesten ist es natürlich, wenn der Zielrechner einen offenen Telnet-Port besitzt. Dann ist es möglich sich auf dem jeweiligen Rechner einzuloggen, und mithilfe eines terminals auf diesem Rechner Befehle auszuführen. Wenn man entsprechende Rechte hat, kann man Dateien hoch- oder runterladen, Programme compilieren und ausführen und Dateien verändern. Interessant wird das natürlich, wenn man vorhat, in das System einzudringen. Oft kann man sich mit einem Guest-Account einloggen. Man wird dann aber meist nicht viele Rechte haben, und auch nicht allzuviel ausrichten können. Man kann daher einige Standard-Accounts und Passwörter ausprobieren, aber die wenigsten Rechner werden die noch drin haben, ausser der Admin wäre ein totaler Ignorant...

Unter Windows und Linux könnt ihr Telnet mit 'telnet <hostname> <port>' starten. Der Hostname ist der Name oder die IP des Remote-Rechners (der auf den ihr wollt :)), z.B. 192.168.1.12 . Als Port gebt ihr nun den Port :)) an, auf den ihr wollt.
In unserem Fall würden wir Port 23 (telnet) angeben, um ein terminal zu erhalten.

Hier bekommt ihr die Versionsnummer der jeweiligen UNIX-Version, die auf dem Rechner installiert ist. Ausserdem müsst
ihr euch jetzt einloggen. Ihr braucht dazu einen gültigen Usernamen und das dazugehörende Passwort. Bei manchen Rechnern
wird man nach 3 Fehlversuchen rausgeworfen, bei anderen hat man soviele Versuche wie man nur will. Um ein Passwort zu bekommen, könnte man es mit einem Brute-Force-Programm versuchen, was aber viel zuviel Aufsehen erregen würde, und daher meistens nicht in Frage kommt. Eine weitere Möglichkeit wäre ein Ausprobieren der Standardpasswörter. Kommt man mithilfe eines Accounts in das System, könnte man versuchen die passwd.dat (???) runterzuladen, und sie mit einem Password-Cracker zu bearbeiten, um einen Account mit mehr Rechten zu erlangen. Da die meisten Systeme heutzutage aber Password-Shadowing benutzen, sind die Passwörter dort nicht mehr in der passwd.dat gespeichert.
Auch kann man versuchen, mithilfe von exploits die Sicherheitslücken einiger programme auf dem rechner auszunutzen, um dadurch root-Rechte zu erlangen (für alle, die mit UNIX GAR NICHTS zu tun haben >> root ist der User, der alle Rechte besitzt, und vollen Zugriff auf den Rechner hat. Also, das was jeder gern wär...). Aber darauf will ich jetzt erstmal nicht eingehen.
Versucht erstmal einen Rechner zu finden, der einen Guest-Account mit beschränktem Zugriff hat. Davon gibt es eigentlich eine ganze Menge. Versuchts vor allem mal bei Universitäten oder anderen öffentlich zugänglichen Servern.
Seid ihr nun am Prompt, das bei Unix meist aus <Username>@<Rechnername>> besteht (also z.B. root@kileak>). Nun braucht ihr erstmal ein paar grundlegende Befehle.

Dos-Kommando	Unix-Befehel	Auswirkung
DIR	LS	Inhalte der Verzeichnisse anzeigen
CD	CD	Verzeichnis wechseln
MD/MKDIR	MKDIR	ein Verzeichnis anlegen
RD/RMDIR	RM	ein Verzeichnis oder
DEL	RM	eine Datei löschen
MORE < FILE	LESS FILE	eine Datei seitenweise ausgeben lassen

Mit diesen Befehlen kann man sich erstmal auf dem Remote-Rechner umsehen, was für Programme installiert sind, welche Versionsnummern sie besitzen. Einige Sachen, die man später z.B. mithilfe Exploits ausnutzen könnte.

Um Programme zu starten kann man unter Unix nicht einfach nur den Dateinamen, wie unter Dos angeben. Gibt man in Unix nur den Programmnamen ein, sucht Unix dieses Programm nur in den in der PATH-Variable angegebenen Verzeichnissen, nicht aber im derzeitigen Verzeichnis (ausser es steht natürlich im PATH). Man muss also, um ein Programm im derzeitigen Verzeichnis zu starten ein './' vorsetzen und z.B. mit './myscript' starten. Abhilfe kann man da schaffen wenn man '.' in die PATH-Variable mit einbaut, aber das ist jetzt mal nicht so wichtig. Man sollte es einfach mal ausprobieren (wodurch man meiner Meinung nach am besten lernt).

Die wichtigsten UNIX-Befehle

Ich hatte schon mal eine kleine Auflistung aller UNIX-Befehle aufgeschrieben. Da ich jetzt zu faul bin, um das alles nochmal neu zu schreiben, hab ich die Text-Datei zu diesem Tutorial beigelegt. Da es eigentlich auf einer englischen Seite veröffentlicht werden sollte, hatte ich es auf englisch geschrieben. Falls

jemand nicht der englischen Sprache mächtig ist, aber diesen Text gerne verstehen würde, mailt mir einfach. Falls genug Interesse besteht, setz ich mich vielleicht ja doch noch mal hin und schreibe den Text nochmal auf Deutsch (Lazy as an ass :))

Compilieren eigener Programme

Habt ihr auf dem Rechner ein Verzeichnis, in dem ihr Dateien schreiben und den Compiler benutzen könnt, könnt ihr dort entweder Programme schreiben, oder sie runterladen und dann dort compilieren. Dazu benutzt ihr entweder

```
gcc -o <ausgabe-datei> <programm>
```

oder

```
cc -o <ausgabe-datei> <programm>
```

als <ausgabe-datei> gebt ihr den namen an, den das fertig compilierte Programm haben soll ;). <programm> ist der name, der datei, die zu compilieren ist. Gebt ihr nicht den schalter '-o <ausgabe-datei>' an, wird das Programm als 'a.out' gespeichert.
Nach dem Compilieren, könnt ihr es dann wie immer mit './<ausgabe-datei>' starten.

Die Well-Known Ports

Die Well-Known Ports sind im RFC 1700 aufgelistet. Hier ist eine kleine Auswahl davon (hab ich von irgendeiner Website (weiß es grad nicht mehr), leicht überarbeitet...

```
      1     TCPMUX TCP Port Service Multiplexer
      5     RJE Remote Job Entry
      7     ECHO
      20    FTP-Data
      21    FTP
      23    TelNet
      25    SMTP Simple Mail Transfer Protocol
      37    Time

      42    Nameserv Host Name Server
      43    WhoIs

      49    Login Login Host Protocol
      53    DNS  Domain Name System
      69    TFTP Trivial File Transfer Protocol
      70    Gopher Gopher Services
      79    Finger
      80    HTTP (WWW)
      103   X400 X.400 standard
      108   SNA Gateway Access Server
      109   POP2
```

```
110   POP3 Post Office Protocol version 3
115   SFTP Simple File Transfer Protocol
118   SQLserver
119   NNTP Newsgroup
137   NetBIOS-NS NetBIOS Name Service
139   NetBIOS-DG NetBIOS Datagram Service
143   IMAP Interim Mail Access Protocol

150   NetBIOS-SS NetBIOS Session Service
156   SQLSRV SQL Server
161   SNMP
179   BGP Border Gateway Protocol
190   gacp Gateway Access Control Protocol
194   IRC Internet Relay Chat-->6667

197   DLS Directory Location Service
389   LDAP
396   netware-ip Novell Netware over IP
443   HTTPS
444   SNPP Simple Network Paging Protocol
458   appleqtc apple quick time
546   dhcp-client DHCP Client
547   dhcp-server DHCP Server

563   SNEWS
565   whoami whoami
569   MSN

- 1080 -    Socks
- 1477 -    ms-sna-server

- 1755 -  MS NetShow
- 4000 -  ICQ
- 5190 -  AOL
- 6667 -  IRC
- 6801 -  Net2Phone
- 6500 -  Net2Phone registration

- 7000 -  VDOLive
- 7070 -  Real Audio
- 7075 -  Real Audio
- 12468 - VXtreme
```

Die letzten paar gehören zwar nicht mehr zu den Well-Known-Ports (<1024), aber trotzdem nützlich zu wissen.

Abschluss

Nun ja, das Tutorial ist vielleicht etwas unzusammenhängend geworden, und es sind auch einige Sachen weggefallen, die ich noch anmerken wollte, aber ich hoffe, das es ein paar Leuten weiter hilft.

Ich muß mich jetzt mal wieder anderen Sachen widmen, es kann also sein, das noch einige Fehler drin sind, hatte keine Zeit mehr es nochmal durchzusehen, tut mir leid :))

cu Razor