Rechnernetze

Security Tools 2 TCP-Wrapper

Bernd Lehle / Oliver Reutter

Security Tools 2 TCP-Wrapper

Bernd Lehle / Oliver Reutter

In Analogie mit den Fantasy-Rollenspielen sehen die Hacker im Internet sich gerne als Nachfolger der mittelalterlichen Ritter und benutzen auch entsprechende Terminologie. In diesem Sinne wollen wir nach dem zweischneidigen Schwert SATAN nun einen recht wirksamen Schild vorstellen - den TCP-Wrapper.

Entwickelt hat dieses genial einfache Hilfsmittel der schon im letzten Artikel genannte Wietse Venema aus Holland. Es ist schon relativ alt und liegt momentan in der sehr stabilen Version 7.2 vor.

Das Prinzip ist denkbar einfach. Hauptproblem der Angriffe im Internet ist die Tatsache, daß man annähernd von jedem beliebigen Rechner aus fast jeden beliebigen anderen Rechner erreichen kann. Ein normaler Computer wird im Laufe seiner Online-Zeit vielleicht mit wenigen millionstel aller anderen Rechner Kontakt aufnehmen, obwohl er in der Lage ist, dies mit allen zu tun. Das Protokoll sieht hier keine Beschränkung vor. Wenn man seine Rechner vor Zugriff aus dem Internet wirksam schützen will, hilft meist nur ein aufwendiger Firewall, der sehr teuer wird, wenn er wirkungsvoll sein soll.

In diese Bresche springt der TCP-Wrapper: Eine Netzverbindung wird entweder direkt zu einem Unix-Daemon, der zu diesem Zweck an einem bestimmten Port lauscht, aufgebaut oder der Gast wendet sich an den Internet-Daemon (`inetd`) oder `portmapper` und wird von diesem an den entsprechenden Prozeß verwiesen. Der `inetd` hat ein simples Konfigurationsfile,das es erlaubt hier eine weitere Kontrollinstanz vorzuschieben.

Das Konfigurationsfile (`/etc/inetd.conf`) sieht typischerweise so aus:

```
telnet stream tcp nowait root /etc/telnetd telnetd -h
```

Hier wird dem `inetd` gesagt, daß der service `telnet` (er gehört zum port 23, was in `/etc/services` steht) folgendermaßen zu behandeln ist:

"Wenn eine Nachfrage nach port 23 kommt, dann baue einen tcp-stream auf, warte nicht, bis er beendet ist und verbinde ihn mit dem Programm `/etc/telnetd`, das mit der Option `-h` aufzurufen ist und unter der uid `root` läuft."

Wenn man nun aber kontrollieren will, wer da `telnet` machen will, bringt man folgende kleine Änderung an:

```
telnet stream tcp nowait root /etc/tcpd telnetd -h
```

`/etc/tcpd` ist der angesprochene TCP-Wrapper (tcp-Daemon). Er wird nun zuerst aufgerufen und erst, wenn er die Verbindung zuläßt, wird sie an den telnet-Daemon weitergegeben, der ein login veranlaßt. In dieser Form läßt sich der Zugang zu allen Diensten regulieren, die in `/etc/inetd.conf` stehen.

Wie entscheidet nun der TCP-Wrapper, wer eine Verbindung aufbauen darf? Wie fast alle Unix-Programme benötigt er dazu Konfigurationsfiles. In diesem Fall heißen sie `etc/hosts.allow` und `/etc/hosts.deny`

Die Regel heißt dann: "Verbinden darf der, der entweder explizit in `hosts.allow` zugelassen oder in `hosts.deny` nicht explizit verboten ist." Das Format dieser beiden Files ist ebenfalls denkbar einfach. Es besteht aus Dienst und Quellrechner oder -netz. Ein typisches `/etc/hosts.allow` - File könnte so aussehen:

```
telnetd :.rus.uni-stuttgart.de
ftpd    :.uni-stuttgart.de
fingerd : ALL
```

Hier wird der telnet-Zugang von allen Rechnern des RUS, der ftp-Zugang von allen Rechner der Universität Stuttgart erlaubt und finger von jedem Rechner. Ein sinnvolles `/etc/hosts.deny` sähe dann so aus:

```
ALL:ALL
```

Das heißt, daß alle außer den soeben zugelassenen Rechnern abgewehrt werden.

Will man die Sache etwas weiter spezifizieren kann man das EXCEPT-Konstrukt verwenden. Will man z.B. keine telnet-Verbindungen von öffentlich zugänglichen Studentenrechnern haben, ändert man die erste Zeile auf:

```
telnetd .rus.uni-stuttgart.de EXCEPT 129.69.21. 129.69.31.133
```

Hierdurch werden alle telnet-Versuche vom rpool, dem PC-Pool und dem studentischen Modemserver abgewehrt, während der Rest des RUS darf. Wenn man sich auf ganze Subnetze bezieht, läßt man einfach die entsprechenden Zahlen hinter dem Punkt weg (`129.69.`). Wenn man sich auf ganze Domains bezieht, läßt man einfach den Rechnernamen vor dem Punkt weg (`.uni-stuttgart.de`).

Bevor man den Wrapper "scharf" macht, sollte man mit dem mitgelieferten Programm `tcpdmatch` testen, ob die Konfiguration auch den gewünschten Schutz bringt oder ob man sich nach dem Ausloggen selbst nicht mehr einloggen kann. Ebenso gibt es ein Programm `tcpdcheck`, das nach Syntaxfehlern in den Konfigurationsfiles sucht.

Es gibt hier noch weitere Möglichkeiten, den Zugang auf Netgroups oder ähnliche Sachen zu beschränken, deren genaue Ausführung hier allerdings zu weit führen würde. Es wird eine gute man-Page mitgeliefert: `host_access (5)`. Dort ist ebenfalls die Möglichkeit beschrieben, auf

bestimmte Connects mit Aktionen zu reagieren. Beispielsweise kann man einen Rechner, von dem eine abgelehnte Connection ausgeht, automatisch anfingern lassen, um die eingeloggten Benutzer herauszufinden.

Im Makefile kann man auch noch eine Option PARANOID spezifizieren, mit der der TCP-Wrapper alle Connects abweist, bei denen IP-Adresse und Hostname laut Nameserver nicht zusammenpassen, was begrenzt vor der Fälschung von Adressen oder Namen schützen kann. Ebenso kann man dem TCP-Wrapper beibringen, über den ident-Daemon beim Ausgangsrechner nach dem Username zu fragen, der die Verbindung aufbaut.

Eine wichtige Sache, die ebenfalls große Lücken in den verbreiteten Betriebssystemen schließt, ist die Logging-Fähigkeit des TCP-Wrappers. Während sich manche Betriebsysteme erst zu einer Meldung im syslog hinreißen lassen, wenn mehrfach failed logins kommen, protokolliert der TCP-Wrapper alle Verbindungen mit, über die er entschieden hat. Etwas unglücklich ist, daß er defaultmäßig in den gleichen Kanal wie sendmail loggt. Man kann dies jedoch im Makefile umbiegen und ein eigenes Logfile mit Hilfe von `/etc/syslog.conf` erzeugen, das dann etwa so aussehen würde:

```
Jan 19 09:10:01 4Q:visbl ftpd[10116]:refused connect from evil.com
Jan 19 09:10:01 4Q:visbl telnetd[10115]:refused connect from some.where.edu
Jan 19 09:10:01 4Q:visbl fingerd[10117]:refused connect from rpool2.rus
Jan 21 10:59:21 6Q:visbl rlogind[11738]:connect from friend.uni.edu
Jan 22 17:55:34 6Q:visbl rlogind[2139]:connect from friend.uni.edu
```

Man wundert sich manchmal, wer einen alles anfingert ... :-)

An zusätzlichen Features bietet das Paket dann noch einige nützliche Utilities, mit denen man z.B. suspekte Rechner sofort beim Verbindungsaufbau anfingern kann, wer eingeloggt ist oder eine Library, mit der man die Fähigkeit die Konfigurationsfiles zu lesen auch anderen Programmen beibringen kann.

Zum Artikelende noch kurz eine Zusammenfassung, was der TCP-Wrapper bietet, was er nicht kann und wo man ihn bekommt.

Was der TCP-Wrapper kann

Er kann wirksam den Zugriff auf Netzwerkdienste regulieren und protokollieren, die über `inetd` gestartet werden.

Was der TCP-Wrapper nicht kann

Er kann nicht den Zugriff auf Netzwerkdienste regeln, die standalone oder über den `portmapper` angesprochen werden. Dafür gibt es eigene Wrapper oder gepatchte portmapper. Speziellere Details stehen in der Dokumentation.

Wo man den TCP-Wrapper herbekommt

Hier zwei wichtige Adressen:

- ftp://ftp.uni-stuttgart.de/pub/unix/security
- ftp://ftp.cert.dfn.de/pub/tools

Warnung!

Dies ist keine plug-and-play-Anleitung, die zum wirksamen Schutz ausreicht. Jeder, der den TCP-Wrapper verwendet, sollte gründlich die beigefügte Dokumentation lesen und verstehen, was er tut.

Ansprechpartner bei sicherheitsrelevanten Fragen

An wen kann ich mich in so einem Fall wenden? Hier zwei wichtige Adressen und Ansprechpartner:

- sneakers@rus.uni-stuttgart.de
- dfncert-request@cert.dfn.de

Bernd Lehle, NA-5531
E-Mail: Lehle@rus.uni-stuttgart.de

Oliver Reutter, NA-4513
E-Mail: Oliver.Reutter@rus.uni-stuttgart.de