Einsatz des *flypaper-Dämons* zur effektiven Abwehr von Portscan-Angriffen

Johannes Franken
<jfranken@jfranken.de>

Abbildung: Klebefalle (engl.: flypaper)

Auf dieser Seite beschreibe ich, wie man Linux-Firewalls so erweitert, dass Hacker nach dem ersten Fehltritt keine zweite Chance erhalten.

Inhalt

Einleitung

Sobald Sie Dienste im Internet anbieten (d.h. Ports auf Ihrer Firewall öffnen), müssen Sie davon ausgehen, dass diese von Hackern angegriffen werden.

Wie verläuft ein Angriff?

Die meisten Angriffe laufen in drei Schritten ab:

1. Zunächst werden die Rechner eines Netzsegments und deren offene Ports mit Portscannern ermittelt. Hierbei sendet der angreifende Rechner eine Reihe speziell vorbereiteter TCP-Pakete oder UDP-Datagramme und beobachtet die Reaktion des Empfängers. Mögliche Reaktionen sind:
    1. **Annahme** der Verbindung: im Fall von TCP ein SYN-ACK-Paket, bei UDP ein Datagramm an die anfragende Portnummer.
    2. **Abwehr** der Verbindung: im Fall von TCP ein RST- oder FIN-Paket, bei UDP eine icmp-Nachricht
    3. **keine Reaktion**: im Fall von TCP wiederspricht dies den RFC und führt zu Retransmits. Bei UDP ist die Reaktion der anrufenden Anwendung überlassen.
2. Dann werden die Dienste ermittelt, die auf diesen Ports lauschen.
3. Anschließend werden die gefundenen Dienste gezielt unter Beschuss genommen, indem Passwortlisten und bekannte Implementierungsschwächen an ihnen ausprobiert werden.

Warum sollte gerade *ich* angegriffen werden?

Kurz gesagt: weil die Wahrscheinlichkeit dafür jede Sekunde steigt.

Die Zahl der Hacker steigt streng monoton. Da die Hacker die gehackten Rechner wiederum als Ausgangspunkt für weitere Angriffe nutzen, steigt die Zahl der angreifenden Rechner quadratisch. Im gleichen Maß (und somit ab einem Zeitpunkt schneller als die Zahl der Internet-Rechner) steigt auch die Zahl der weltweit erfolgenden Angriffe pro Sekunde. Dann steigt für *jeden* Rechner stetig die Wahrscheinlichkeit, Ziel eines Angriffs zu werden.

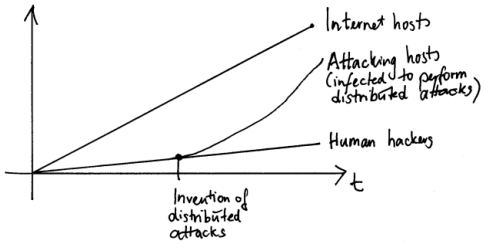

Abbildung: Hacker-Potential

Eine Auswertung mit dem *flypaper-Dämon* im September 2006 hat ergeben, dass ein Einwahlrechner beim Internetprovider t-online.de täglich durchschnittlich von 397 verschiedenen Rechnern angegriffen wird.

Was bewirkt der flypaper-Dämon?

Der *flypaper-Dämon* schützt vor Portscans (Schritt 1 des Angriffs), indem er **keine Reaktion** auf Pakete aller Rechner zeigt, die einmal einen geschlossenen Port angesprochen haben. Wenn Sie zusätzlich Ihre Dienste auf unübliche Portnummern verschieben (z.B. den ssh-Dämon auf den TCP-Port 43974), ist es ziemlich unwahrscheinlich, dass ein angreifender Rechner Ihre Dienste beim ersten Versuch findet, was jedoch die Voraussetzung für den eigentlichen Angriff (Schritt 3) wäre. Mit *traditionellen* Portscannern, welche die Portnummern in der Reihenfolge von 0 bis 65535 ansprechen, ist es sogar absolut ausgeschlossen, einen offenen Port zu finden.

Lösung

Konzept

Der *flypaper-Dämon* nimmt die IP-Adressen aller Absender von Paketen, die nicht *explizit* (d.h. mit einer ACCEPT-Regel) erlaubt sind, automatisch in eine Blacklist auf. Bei der Blacklist handelt es sich um eine netfilter-Kette, welche automatisch bei jedem eintreffenden Paket abgearbeitet wird.

Aufbau

Netfilter

Die im Linuxkernel 2.4 und 2.6 eingebaute Firewall heißt "netfilter". Sie ist insbesondere über die Schnittstelle "iptables" konfigurierbar.
Anatomie:

- Es gibt verschiedene "Tafeln" ("tables"): **`filter`**, **`nat`** und **`mangle`**.
- Die Tafeln enthalten verschiedene "Ketten" ("chains"). Im Fall der **`filter`**-Tafel sind das zunächst die Ketten **`INPUT`**, **`OUTPUT`** und **`FORWARD`**.
- Jede Kette setzt sich zusammen aus verschiedenen "Regeln" ("rules") und einer "Standard-Aktion" ("default policy"), welche beim Erreichen des Kettenendes ausgeführt wird.
- Jede Regel benennt verschiedene "Bedingungen" ("specifications") und eine "Aktion" ("target"), die bei Zutreffen aller Bedingungen ausgeführt werden soll. Die möglichen Bedingungen und Aktionen sind von der Kernelkonfiguration abhängig.
- Es gibt insbesondere folgende Bedingungen:
  - die IP-Adresse des Absenders
  - die angesprochene Netzschnittstelle
  - die angesprochene Portnummer

  Weitere Bedingungen finden Sie in der **`iptables(1)`**-Manpage.
- Es gibt insbesondere folgende Aktionen:
  - **`ACCEPT`**: beendet die Verarbeitung der Ketten. Das Paket wird an die zuständige Anwendung übergeben.
  - **`DROP`**: verwirft das Paket und beendet die Verabeitung der Ketten.
  - **`QUEUE`**: übergibt das Paket zur weiteren Inspektion an einen Userspace-Prozess (hier: an den flypaper-Dämon)
  - *Kettenname*: springt in eine andere Kette (derselben Tafel)
  - **`RETURN`**: springt zurück in die aufrufende Kette.

  Weitere Aktionen finden Sie in der **`iptables(1)`**-Manpage.

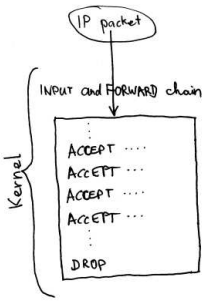

Abbildung: iptables (ohne flypaper)

flypaper-Kette

Beim Einsatz des *flypaper daemon* werden alle Pakete zuerst von der INPUT- oder FORWARD-Kette durch die "flypaper"-Kette umgeleitet, welche alle bisher erkannten Hacker DROPt (Blacklist). Anschließend erfolgt der Rücksprung in die ursprüngliche Kette.

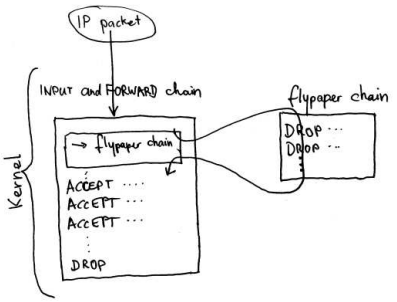

Abbildung: iptables mit flypaper

Wenn keine der ACCEPT-Regeln zutrifft, würde normalerweise die Standardaktion DROP am Kettenende greifen. Beim Einsatz des *flypaper daemon* wird jedoch das DROP dem flypaperd-Prozess überlassen, welcher außerdem den Absender in die "flypaper"-Kette aufnimmt.

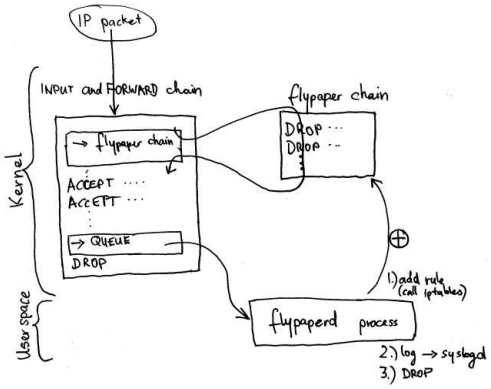

Abbildung: iptables mit flypaper

Bedienung

Starten und Statusabfrage

```
$ /etc/init.d/flypaperd start
Starting portscan protection: flypaperd.

...

$ /etc/init.d/flypaperd status
Kernel ip_queue support:                          OK
flypaper chain existence:                         OK (390 flies)
flypaper chain integration to INPUT chain:        OK
flypaper chain integration to FORWARD chain:      OK
flypaperd process:                                OK
flypaperd integration (QUEUE target):             OK
```

Liste der Flies

```
$ /etc/init.d/flypaperd list
Chain flypaper (2 references)
target     prot opt source               destination
```

```
DROP       all  --  84.178.96.11         0.0.0.0/0
DROP       all  --  204.16.208.161       0.0.0.0/0
DROP       all  --  84.178.93.112        0.0.0.0/0
[...]
```

```
$ grep flypaper /var/log/auth.log |tail
Aug 13 20:48:07 gate flypaperd[14183]: lockout 69.0.231.79, who tried TCP-port 20000
Aug 13 20:48:25 gate flypaperd[14183]: lockout 84.171.107.239, who tried TCP-port 135
Aug 13 20:50:15 gate flypaperd[14183]: lockout 84.173.231.94, who tried TCP-port 135
Aug 13 20:50:47 gate flypaperd[14183]: lockout 84.178.99.132, who tried TCP-port 135
Aug 13 20:53:20 gate flypaperd[14183]: lockout 59.191.61.67, who tried UDP-port 1434
Aug 13 21:00:00 gate flypaperd[14183]: lockout 84.176.230.110, who tried TCP-port 135
Aug 13 21:01:01 gate flypaperd[14183]: lockout 24.70.95.190, who tried UDP-port 1026
Aug 13 21:06:07 gate flypaperd[14183]: lockout 84.178.93.112, who tried TCP-port 445
Aug 13 21:08:23 gate flypaperd[14183]: lockout 204.16.208.161, who tried UDP-port 1027
Aug 13 21:08:57 gate flypaperd[14183]: lockout 84.178.96.11, who tried TCP-port 135
```

Befreien von Flies

```
$ iptables -D flypaper -s www.jfranken.de -j DROP
```

...

```
$ /etc/init.d/flypaperd flush
```

Stoppen

```
$ /etc/init.d/flypaperd stop
Stopping portscan protection: flypaperd.
```

...

```
$ /etc/init.d/flypaperd status
Kernel ip_queue support:                        OK
flypaper chain existence:                       OK (2 flies)
flypaper chain integration to INPUT chain:      Missing
flypaper chain integration to FORWARD chain:    Missing
flypaperd process:                              Missing
flypaperd integration (QUEUE target):           OK
```

Diskussion

Führt das nicht zu DOS-Attacken?

Nicht unbedingt. Denn der Firewall-Administrator kann die "Aggressivität" des *flypaper daemon* über die Bedingungen der QUEUE-Regel selbst steuern.

Downloads

- **/etc/init.d/flypaperd** (das Startscript) [3 kB]
- **/usr/local/bin/flypaperd** (der Dämon) [3 kB]

Installation (Debian):

```
$ update-rc.d flypaperd defaults
```