RUZANNA ARUTYUNYAN, 123RF

Remote einloggen mit Port Knocking

Klopf, klopf

SSH mit Passwörtern ist gegen Brute-Force-Angriffe nicht von Haus aus gesichert. Eine Alternative bietet Port Knocking, das geschlossene Ports nur bei einer Klopfsequenz öffnet. SPA überträgt das Prinzip auf ein speziell angefertigtes Datenpaket. Juliet Kemp

Mit SSH verschlüsselter Netzwerk-Traffic ist gegen Abhören gesichert und – wenn man die Identität der verwendeten Schlüssel immer verifiziert – auch gegen Man-in-the-Middle-Angriffe. Sehr selten gibt es einmal einen Fehler in einem SSH-Programm, das dann auch schnell geschlossen ist. Eine Bedrohung sind aber Brute-Force-Angriffe, bei denen ein Angreifer Variationen von Passwörtern mit einem Skript ausprobiert.

Wer nur wenige User-Accounts auf dem Server hat, dazu noch möglichst unbekannte und ungewöhnliche Benutzernamen und komplizierte Passwörter einsetzt, ist gegen Brute-Force-Angriffe gut gewappnet. Wenn die Benutzernamen aber nach einem Schema gebildet sind, zum Beispiel dem ersten Buchstaben des Vornamens gefolgt vom Nachnamen, wird es für Hacker schon einfacher. Über eine Mitarbeiterliste auf der Website können sie so mögliche Benutzernamen erschließen. Wenn die User dann noch selber ihre Passwörter ändern dürfen, ohne dass eine Policy deren Komplexität sicherstellt, erhöhen sich die Einbrecher-Chancen gewaltig.

Eine Lösung für das Problem stellt ein Passwort-Cracker wie John the Ripper [1] dar, den Sie lokal über die eigene Passwort-Datenbank laufen lassen können, bevor jemand anderes das Gleiche von außen mit schlechten Intentionen tut.

Gesperrt

Gegen Brute-Force-Angriffe auf SSH gibt es einige Pakete wie Fail2ban oder SSH-Guard, die Zugriffe von einer IP-Adresse mit Firewallregeln blocken, wenn von ihr aus in einem bestimmten Zeitraum zu viele gescheiterte Login-Versuche erfolgen. Je nach Konfiguration kann die Sperre permanent oder von einigen Mi-

nuten Dauer sein. Letzteres erfordert keinen Eingriff des Administrators zum Entsperren, erschwert Brute-Force-Angriffe aber trotzdem.

Eine weitere Alternative ist das so genannte Port Knocking, um das es im weitere Verlauf dieses Artikels gehen soll. In einer traditionellen Port-Knocking-Konfiguration sind alle Ports eines Servers geschlossen. Für einen Portscan sieht es so aus, als wäre kein Zugang zum Rechner möglich. Ein legitimer User initiiert dann nacheinander je eine Verbindung zu unterschiedlichen (geschlossenen) Ports auf dem Server. Diese schlagen fehl, aber eine spezielle Software auf dem Rechner registriert die Verbindungsversuche trotzdem und ändert die Firewall-Konfiguration so, dass der Benutzer sich schließlich einloggen kann.

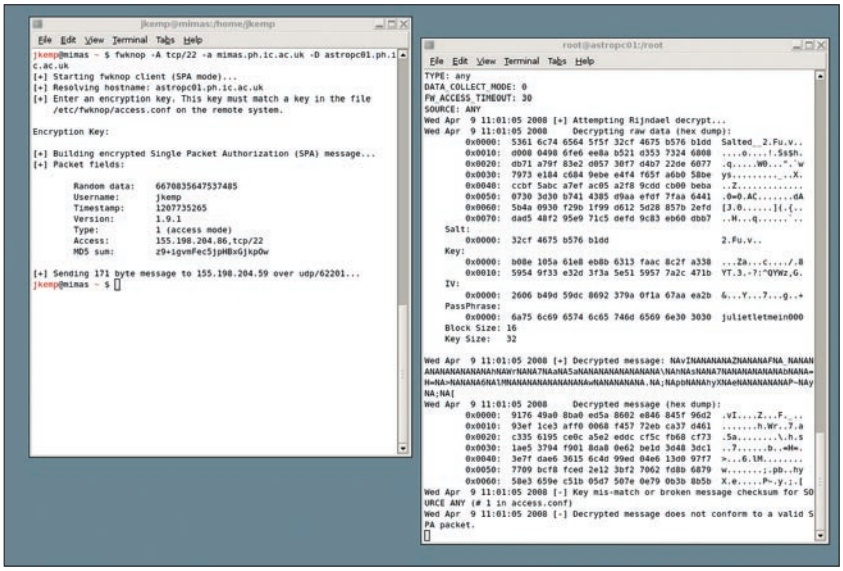

Abbildung 1: Port Knocking verwendet IPtables, um den Zugang zum Server zu regeln.

Öffnungszeiten

Ein Secure-Shell-Port wird also nur unter gewissen Umständen geöffnet, statt die ganze Zeit auf Verbindungen aus aller Herren Länder zu warten. Damit kommt zur Authentifizierung per SSH-Passwort noch eine weitere Schutzhülle hinzu. Dennoch bringt das traditionelle Port Knocking auch einige Nachteile mit sich. Zum Beispiel braucht man für den Anklopf-Vorgang eine spezielle Software. Je nachdem, wie technisch versiert die Anwender sind oder wie gut Sie selbst deren Clientrechner konfigurieren können oder wollen, kann das ein großes Problem sein. Auf fremden Rechnern wie etwa im Internetcafé kann man diese Option sowieso vergessen.

Wenn bei der Port-Knocking-Konfiguration auf dem Server etwas schiefgeht, sind Sie möglicherweise komplett ausgesperrt und müssen sich dann lokal anmelden. Außerdem ist Port Knocking nicht völlig immun gegen Angriffe, denn wenn ein Hacker das Klopfmuster mitschneidet, kann er es anschließend selbst verwenden.

Nur ein Paket

Eine fortgeschrittene Variante der gleichen Idee ist die so genannte Single-Packet-Autorisierung (SPA). Im Gegensatz zum traditionellen Port Knocking dient dabei keine Klopfsequenz, sondern ein einziges verschlüsseltes Paket als Eintrittskarte. Eine SSH-Sitzung kann nur dann aufgebaut werden, wenn der Server ein gültiges verschlüsseltes Paket empfangen hat, andernfalls ist von außen kein SSH-Dienst zu erkennen.

Der größte Vorteil von SPA ist, dass ein Angreifer mitgeschnittenen Traffic nicht für eigene Zwecke missbrauchen kann. Außerdem ist SPA schneller und weniger leicht zu entdecken, weil es nur ein Paket verwendet.

Das beste Paket für die Single-Packet-Autorisierung ist Fwknop, das auf der Cipherdyne-Website zum Download bereitsteht [2], die der Security-Spezialist Michael Rash betreibt. Fwknop beherrscht zwar auch das traditionelle Port Knocking, Rash empfiehlt jedoch dringend den SPA-Ansatz.

Um Fwknop zu installieren, benötigen Sie das Entwicklungspaket der Packet-Capture-Bibliothek Libpcap, die zum Beispiel unter Ubuntu/Debian »libpcap-dev« heißt. Zusätzlich setzt Fwknop einen Mailserver wie Sendmail oder Postfix voraus. Laden Sie sich dann das neueste Tar-Paket von [3] herunter und entpacken Sie es. Derzeit gibt es einen Release-Kandidaten von Fwknop 2, der in der Programmiersprache C geschrieben ist. Die stabile Version 1.x, auf die sich dieser Artikel konzentriert, benutzt die Skriptsprache Perl.

Die Installation startet, wenn Sie »./install.pl« ausführen. Wählen Sie bei der Konfiguration »server« für den lokalen Ausführungsmodus aus und »pcap« als Methode zum Datensammeln. Geben Sie das richtige Netzwerk-Device (zum Beispiel »eth0«) an und eine E-Mail-Adresse, an die Fwknop Benachrichtigungen schickt. Am Ende können Sie auswählen, ob Fwknop beim Booten automatisch startet.

Die Konfiguration editieren Sie in »/etc/fwknop/access.conf«, die Datei »/etc/fwknop/fwknop.conf« können Sie im Regelfall so belassen. Alle Zugriffsregeln befinden sich in »access.conf«, per Default erlaubt Fwknop damit den SSH-Zugriff 30 Sekunden lang, nachdem es den zugelassenen Schlüssel erhalten hat. Normalerweise müssen Sie nur die eine Zeile mit dem »KEY«-Wert verändern und dort einen Schlüssel mit mindestens acht Zeichen vergeben.

IPtables

Bevor Sie Fwknop starten, prüfen Sie noch, ob in der IPtables-Konfiguration bestehende und Loopback-Verbindungen erlaubt sind, aber alle anderen Pakete verworfen werden. Nehmen Sie sich zur IPtables-Konfiguration ruhig etwas Zeit, denn nur allzu schnell sperren Sie sich bei einem Fehler aus (Abbildung 1).

Listing 1: Input-Chain

```
01 iptables -A INPUT -d 1.2.3.4 -m state --state
   RELATED,ESTABLISHED -j ACCEPT
02 iptables -A INPUT -i lo -j ACCEPT
03 iptables -P INPUT DROP
```

Wenn auf dem Server schon IPtables läuft, geben Sie »iptables -F« ein, um das Regelwerk zu leeren. Die Kommandos aus Listing 1 geben die Regeln für die Eingabekette an. Die IP-Adresse »1.2.3.4« ersetzen Sie dabei durch die Adresse Ihres Servers.

Diese Konfiguration belässt bestehende Verbindungen (Zeile 1) und Verbindungen über die Loopback-Schnittstelle (Zeile 2), aber verwirft alles andere (Zeile 3). Die Output- und Forward-Chains von IPtables interessieren in diesem Zusammenhang nicht weiter. Starten Sie dann Fwknop mit »/etc/init.d/fwknop start«.

Erster Test

Zum Testen brauchen Sie neben dem Server auch einen Client. Installieren Sie dazu wie oben beschrieben die Libpcap-Dev und das Fwknop-Paket, wählen Sie aber dieses Mal »client« aus. Versuchen Sie zuerst, sich auf dem Server per SSH einzuloggen. Dank der oben erstellten Firewallregeln sollte der Versuch fehlschlagen. Erscheint dennoch ein Login-Prompt, müssen Sie Ihre Firewall-Konfiguration überarbeiten.

Zum Anklopfen geben Sie auf der Konsole ein:

```
fwknop -A tcp/22 -a 192.168.1.136 -D ➥
192.168.111.20
```

Hinter »-A« folgt der freizuschaltende Server, hinter »-a« der Hostname oder die IP-Adresse des Client-Rechners und hinter »-D« schließlich den Server.

Fwknop fragt Sie nach dem Passwort, das Sie auf der Konsole eintippen, baut dann die SPA-Nachricht zusammen und schickt sie an den Server. Im Erfolgsfall haben Sie 30 Sekunden Zeit, um sich mit SSH einzuloggen. Diesen Zeitraum können Sie auf dem Server in »access.conf« ändern. Dass Fwknop noch einmal die Clientadresse erwartet, soll verhindern, dass Angreifer gefälschte Pakete verschicken. Dazu dupliziert Fwknop die Adresse noch einmal im anschließend verschlüsselten Paket.

Listing 2: »access.conf« für GPG-Schlüssel

```
01 SOURCE: ANY;
02 OPEN_PORTS: tcp/22;
03 DATA_COLLECT_MODE: PCAP;
04 FW_ACCESS_TIMEOUT: 30;
05 GPG_HOME_DIR: /root/.gnupg;
06 GPG_DECRYPT_ID: AAAAAAAA;
07 GPG_DECRYPT_PW: myGpgPassword;
08 GPG_REMOTE_ID: CCCCCCCC;
```

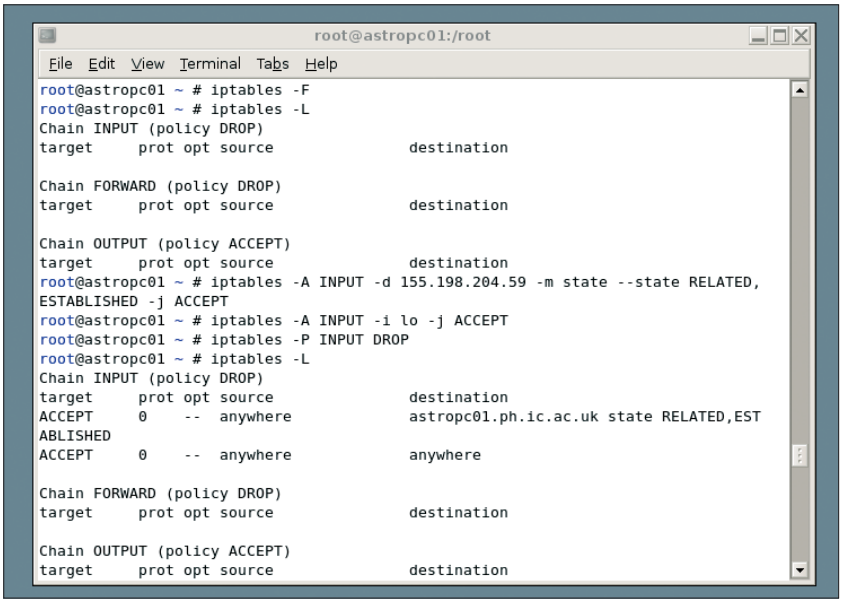

Abbildung 2: Fwknop im Einsatz mit einem einfachen Passwort, rechts die Analyse mit TCPDump.

Wen sich zwischen Client und Server eine Firewall befindet, kommt das Klopf-Paket vielleicht nicht durch. Ist das der Fall, editieren Sie den Wert für »PCAP_FILTER« in »/etc/fwknop/fwknop.conf« und geben einen erlaubten Port an. Am Client verwenden Sie »--Server-port«, um die Portnummer anzugeben (Abbildung 2):

```
fwknop -A tcp/22 --Server-port 8080 ➥
-a 192.168.1.136 -D 192.168.111.20
```

Um einem Fehler in der Konfiguration auf die Spur zu kommen, starten Sie den Fwknop-Daemon im Debugging- und Verbose-Modus: »fwknop -d -v«. Damit zeigt Fwknop alle Verbindungsversuche und die Firewallaktionen an. Die IPtables-Chain »FWKNOP_INPUT« legt das Tool allerdings erst beim ersten Verbindungsversuch an. Wenn sie vorher fehlt, ist das also kein Fehler. Um die Veränderungen an der Firewall-Konfiguration zu beobachten, geben Sie in einem Terminalfenster »watch -n1 iptables -L -n« ein.

Mit GPG-Schlüssel

Statt eines Plaintext-Schlüssels kann Fwknop auch GPG-Schlüssel benutzen. Allerdings dürfen Sie nicht Ihren privaten GPG-Key dafür verwenden, denn das Passwort zum Entschlüsseln muss in der »/etc/fwknop/access.conf« stehen. Doch können Sie auf dem Client einen existierenden Key verwenden, wenn Sie einen haben. Falls nicht, erfahren Sie jetzt, wie man einen neuen anlegt. Geben Sie am Server dazu folgende Befehle ein:

```
gpg --gen-key
gpg --list-keys
```

Die Default-Optionen von GPG sind in Ordnung (DSA- und Elgamal-Schlüssel, 2048 Bit Länge und kein Ablaufdatum). Sie sollten keinen Schlüssel länger als 2048 Bit verwenden, denn er muss noch in ein Datenpaket passen. Geben Sie die passenden Servernamen und E-Mail-Adressen an und notieren die Passphrase. Die Ausgabe sieht in etwa so aus:

```
pub   1024D/AAAAAAAA 2008-03-07
uid   server.example.com fwknop <username@➥
example.com>
sub   2048g/BBBBBBBB 2008-03-07
```

Dann importieren Sie den Schlüssel im Ascii-Format:

```
$ gpg -a --export AAAAAAAA > server.asc
```

Erzeugen Sie auch auf dem Client einen Key und exportieren ihn:

```
$ gpg --gen-key
$ gpg --list-keys
pub   1024D/CCCCCCCC 2008-03-07
uid   test fwknop <username-test@example.➥
com>
sub   2048g/DDDDDDDD 2008-03-07
$ gpg -a --export CCCCCCCC > client.asc
```

Nun übertragen Sie die Schlüssel über einen sicheren Kanal jeweils auf den anderen Rechner. Wenn Fwknop schon läuft, müssen Sie – wie oben beschrieben – anklopfen. Nun importieren und signieren Sie beide Schlüssel. Zunächst auf dem Client:

```
$ gpg --import server.asc
$ gpg --edit-key fwknop
Command> sign
Command> save
```

Ersetzen Sie dabei »fwknop« durch den Namen, der zur hexadezimalen Key-ID gehört, im Beispiel AAAAAAAA. Auf dem Server verfahren Sie mit dem Client-Key analog. Die Fwknop-Konfiguration für die im Beispiel verwendeten Schlüssel sieht dann so aus wie in Listing 2.

Hierbei ist »_ _GPG_DECRYPT_ID_ _« der Serverschlüssel, »_ _GPG_DECRYPT_PW_ _« das Passwort dafür. »_ _GPG_REMOTE_ID_ _« ist die GPG-Key-ID des Clients. Starten Sie dann den Fwknop-Server zum Testen neu. Der dafür passende Aufruf auf dem Client sieht folgendermaßen aus:

```
fwknop -A tcp/22 --gpg-recip AAAAAAAA --gpg-sign CCCCCCCC -w -D server.example.com
```

Der Serverschlüssel folgt hinter »--gpg-recip«, die Client-Key-ID nach »--gpg-sign«. Wenn Sie, wie verlangt, die GPG-Passphrase eingegeben haben, schickt Fwknop das Paket an den Server. Nun sollten Sie sich wieder während des eingestellten Zeitfensters per SSH einloggen können.

Kosten und Nutzen

Beim derzeitigen Stand der Konfiguration kann sich nur ein einziger Benutzer auf dem Server einloggen. Damit das auch andere tun können, fügen Sie weitere Zeilen mit »GPG_REMOTE_ID« der »access.conf« hinzu. Allerdings muss jeder Schlüssel wieder auf den Server importiert und signiert werden, genauso wie der jeweilige Benutzer den Server-Key importieren und signieren muss. Für eine normale Login-Umgebung ist das natürlich etwas viel Aufwand, aber der ist für besondere Maschinen mit einer kleinen Gruppe von Admins vielleicht noch vertretbar.

Um den Zugang außerdem auf einzelne Benutzernamen zu beschränken, verwenden Sie in »/etc/fwknop/access.conf« den Parameter »REQUIRE_USERNAME«. Analog dazu können Sie auch noch ein bestimmtes Betriebssystem oder eine Quelladresse verlangen. Mehr Informationen dazu finden Sie in der Manpage zu »fwknop«. *(ofr)* ■

Infos

[1] John the Ripper: [http://www.openwall.com/john/]

[2] Cipherdyne: [http://www.cipherdyne.org]

[3] Fwknop-Download: [http://www.cipherdyne.org/fwknop/download/]