SSH Absicherung in 20 Minuten

SSH Absicherung in 20 Minuten so gehts

allo miteinander,

heute möchte ich mich etwas näher mit dem Thema SSH und dessen Absicherung beschäftigen. Viele Leute nehmen das Ganze leider viel zu wenig ernst, was dazu führen kann, dass Server geknackt und zweckentfremdet werden können. Dabei ist es so einfach, den Server richtig abzusichern und dafür Sorge zu tragen, dass das knacken des Server nicht mehr – bzw. nur noch extrem schwer – möglich ist.

Was benötigen wir?

1. Einen auf Linux basierten Server
2. Ein bisschen Feingefühl im Umgang mit der Shell
3. IPTables
4. Open-SSH Server
5. Einen dynamischen DNS Hostnamen (DynDNS Services gibt es ja hier und da – keine Werbung)
6. 20 Minuten Zeit

Vorwort:

20 Minuten brauchen wir nur einmal initial zum Aufsetzen des Ganzen Konstrukts. Sofern ihr 2 oder mehr Server hab, geht das Ganze “ratz fatz”. Wenn Ihr noch im Anfängerstadium im Umgang mit einem Linux Server seid, dann nehmt euch die Zeit und lest alles durch. Ich gebe an der ein oder anderen Stelle wirklich ein paar Wertvolle Tipps, die ihr im Alltag im Umgang mit einem Linux-Server benötigt!

Los geht’s!

Zunächst loggen wir uns auf dem Linux Server ein, auf dem wir den SSH Service absichern wollen. Bitte werdet zum “root”:

> Zitat
>
> sudo su –
> oder
> su

sofern ihr das gemacht habt, legen wir nun eine Ordnerstruktur an, damit wir das Ganze nachher über die “Crontab” des Roots aufrufen können. Die Crontab eines nutzers ist ein Modul des Cron. Crons oder Cronjobs sind “Aufgaben” die zu einer bestimmten Zeit beliebig ausgeführt werden. Auf englisch nennt man dies einen Task Sheduler (Windows bietet auch einen solchen Task Sheduler an – nur als Info)

Ordnerstruktur:

> Zitat
>
> mkdir -p /home/security/iptables.d

Der Parameter “-p” legt ALLE Verzeichnisse an, die es bis dahin noch nicht gibt (Rekursion). In eurem Fall wird dies vermutlich der Ordner “security” im “/home” Verzeichnis sein, sowie der Ordner “iptables” im “/home/security” Ordner.
Im nächsten Schritt legen wir 2 Dateien an, die wir später mit Inhalt füllen.

> Zitat
>
> touch /home/security/iptables
> touch /home/security/iptables.d/ssh.secure

Diese beiden Dateien benötigen wir im weiteren Verlauf, um den SSh Port abzusichern. Doch zunächst prüfen wir, welchen Port wir derzeit als SSH-Port benutzen. Das finden wir ganz schnell raus indem wir uns diesen aus der aktuellen Konfiguration “greppen”:

> Zitat

grep ‘Port ‘ /etc/ssh/sshd_config

Die Ausgabe könnte dann so aussehen:

Zitat

Port 22

Nun wissen wir, dass wir den Port “22” benutzen. Nun können wir unsere beiden Dateien, die wir zuvor angelegt haben, mit Inhalt füllen. Dies tuen wir wie folgt:

Zitat

vi /home/security/iptables

In diese Datei fügren wir folgenden Inhalt ein:

Zitat

#!/bin/bash

First of all flush iptables
/sbin/iptables –flush

include security files
source

Kurze Erläuterung:
“#!/bin/bash” – Gibt den Interpreter des Skripts an – in diesem Fall ist dies die BASH
“/sbin/iptables –flush” – leert die IPTables. Würden wir das nicht tun, entstehen Fehler, die dazu führen, dass wir eventuell keinen Zugriff mehr auf den Server haben”source /home/security/iptables.d/ssh.secure” – bindet die Datei an, die hinter “source” steht

Jetzt speichern wir die Datei ab und geben dieser noch die benötigten Rechte. Dies tut man mittels des Kommandos:

Zitat

chmod 755 /home/security/iptables

– 755 bedeutet, dass der Benutzer, dem die Datei gehört, diese Lesen, Schreiben und ausführen darf. Die Gruppe, die der Datei angehört und alle anderen Benutzer auf dem System, dürfen die Datei lesen und ausführen. Wenden wir uns nun unserer zweiten Datei zu:

Zitat

vi /home/security/iptables.d/ssh.secure

Dort fügren wir folgenden Inhalt ein:

Jetzt legen wir noch einen kleinen Crontab Eintrag an. Dieser sieht wie folgt aus:

Zitat

allow ssh from
zuhause
/sbin/iptables -A INPUT -p tcp -m tcp -s meinname.dynamischerdns.tld –dport 22 -j ACCEPT

deny all other
/sbin/iptables -A INPUT -p tcp -m tcp -s 0.0.0.0/0 –dport 22 -j DROP

WICHTIG: Bitte kommentiert immer eure Eintragungen wie z. B. “# zuhause”, damit ihr später wisst, wieso dieser Eintrag vorhanden ist!
Erläuterung:
“/sbin/iptables -A INPUT -p tcp -m tcp -s meinname.dynamischerdns.tld –dport 22 -j ACCEPT” – Akzeptiert jeglichen reinkommenden Traffic auf Port 22 von der IP/ dem Host meinname.dynamischerdns.tld
”iptables -A INPUT -p tcp -m tcp -s 0.0.0.0/0 –dport 22 -j DROP” – Lässt alles andere von jeder IP aus dem Internet “fallen”, der nicht mit "Accept" zugelassen wurde.

Alles anzeigen

Zitat

*/2 * * * * /sbin/iptables --flush

Das stellt sicher, dass sollten wir einen Fehler eingebaut haben, die IPTables wieder gelöscht werden, damit wir nach 2 Minuten wieder Zugriff auf den Server erhalten!

Zum Schluss rufen wir das Skript einmal manuell aus:

Zitat

/home/security/iptables

Wenn nun keine Fehler entstanden sind - und ihr nicht ausgesperrt wurdet, könnt ihr mit dem nächsten Schritt beginnen. Sollten Fehler auftauchen, dann meldet euch bitte bei mir, oder lest euch NOCHEINMAL AUFMERKSAM das Tutorial durch. Für alle die, bei denen es geklappt hat, prüfen wir nun mit dem folgenden Kommando unsere iptables:

> Zitat
>
> /sbin/iptables -L

Nun solltet ihr eine Ausgabe bekommen, die in etwa so aussehen sollte:

> Zitat
>
> Zitat:
>
> Chain INPUT (policy ACCEPT)
> target prot opt source destination
> ACCEPT tcp — meinname.dynamischerdns.tld anywhere tcp dpt:22
> DROP tcp — anywhere anywhere tcp dpt:22

Sollte eure Ausgabe so aussehen, dann habt ihr alles soweit richtig gemacht. Nun können wir noch unseren Cronjob installieren und initialisieren:

crontab -e

Und dort definieren wir, dass das Ganze alle 5 Mnuten ausgeführt werden soll

Zitat:

> Zitat
>
> */5 * * * * /home/security/iptables

Die Zeile

Zitat:

> Zitat
>
> */2 * * * * /sbin/iptables --flush

löschen wir nun wieder!

abspeichern – fertig!
Nun könnt ihr euren dynmischen DNS beliebig mit eurer dynamischen IP zuhause updaten und auf den Server zugreifen. Natürlich könnt ihr noch weitere IPs und Hostnamen freigeben. Dies definiert ihr einfach unter den Bereich “# zuhause” mit einer weiteren iptables Zeile.