

UFW Firewall

Einrichten einer Firewall mit UFW unter Ubuntu 20.04

Einführung

UFW (oder Uncomplicated Firewall) ist eine vereinfachte Firewall-Verwaltungsschnittstelle, die die Komplexität von Paketfilterungstechnologie auf niedriger Ebene wie `iptables` und `nftables` versteckt. Wenn Sie mit dem Sichern Ihres Netzwerks beginnen möchten und Sie nicht sicher sind, welches Tool Sie verwenden sollen, könnte UFW die richtige Wahl für Sie sein.

In diesem Tutorial erfahren Sie, wie Sie eine Firewall mit UFW unter Ubuntu 20.04 einrichten.

Voraussetzungen

Um dieser Anleitung zu folgen, benötigen Sie:

- Einen Ubuntu 20.04-Server mit einem sudo Nicht-root User, den Sie über das [Tutorial Ersteinrichtung des Servers unter Ubuntu 20.04 einrichten können](#).

UFW ist unter Ubuntu standardmäßig installiert. Wenn UFW aus einem bestimmten Grund deinstalliert wurde, können Sie UFW mit `sudo apt install ufw` installieren.

Schritt 1 — Verwenden von IPv6

mit UFW (optional)

Dieses Tutorial wurde für IPv4 verfasst, funktioniert aber auch für IPv6, solange es aktiviert ist. Wenn auf Ihrem Ubuntu-Server IPv6 aktiviert ist, muss UFW so konfiguriert sein, dass IPv6 unterstützt wird, um Firewall-Regeln nicht für IPv4, sondern auch für IPv6 zu verwalten. Öffnen Sie dazu die UFW-Konfiguration mit `nano` oder Ihrem bevorzugten Editor.

```
sudo nano /etc/default/ufw
```

Stellen Sie dann sicher, dass der Wert von `IPV6` `yes` lautet. Das sollte wie folgt aussehen:

```
IPV6=yes
```

Speichern und schließen Sie die Datei. Wenn UFW aktiviert ist, wird es so konfiguriert, dass sowohl IPv4- als auch IPv6-Firewall-Regeln geschrieben werden. Bevor wir UFW aktivieren, wollen wir jedoch überprüfen, ob Ihre Firewall so konfiguriert ist, dass Verbindungen über SSH möglich sind. Beginnen wir mit der Einstellung der Standardrichtlinien.

Schritt 2 — Einrichten von Standardrichtlinien

Wenn Sie gerade mit der Verwendung Ihrer Firewall begonnen haben, sind Ihre Standardrichtlinien die ersten Regeln, die Sie definieren sollten. Diese Regeln steuern die Handhabung von Daten, die nicht ausdrücklich von anderen Regeln abgedeckt werden. Standardmäßig ist UFW so konfiguriert, dass alle eingehenden Verbindungen abgelehnt und alle ausgehenden Verbindungen zugelassen werden. So kann niemand, der versucht, Ihren Server zu erreichen, eine Verbindung herstellen, während jede Anwendung innerhalb des Servers nach außen kommunizieren kann.

Lassen Sie uns Ihre UFW-Regeln zurück auf die Standardeinstellungen setzen, um sicherzugehen, dass Sie diesem Tutorial folgen können. Um die von UFW verwendeten Standardeinstellungen auszuwählen, verwenden Sie diese Befehle: ▣

```
sudo ufw default deny incoming  
sudo ufw default allow outgoing
```

Diese Befehle legen die Standardeinstellungen fest: eingehende Verbindungen werden abgelehnt und ausgehende Verbindungen zugelassen. Die Standardeinstellungen der Firewall allein können für einen PC ausreichen, Server müssen aber normalerweise auf eingehende Anfragen von externen Benutzern reagieren. Das sehen wir uns als Nächstes an.

Schritt 3 — Zulassen von SSH-

Verbindungen

Wenn wir unsere UFW-Firewall jetzt aktivieren würden, würde sie alle eingehenden Verbindungen ablehnen. Das bedeutet, dass wir Regeln erstellen müssen, die legitime eingehende Verbindungen (z. B. SSH- oder HTTP-Verbindungen) ausdrücklich zulassen, wenn unser Server auf diese Art von Anforderungen reagieren soll. Wenn Sie einen Cloud-Server verwenden, werden Sie wahrscheinlich eingehende SSH-Verbindungen zulassen wollen, damit Sie sich mit Ihrem Server verbinden und den Server verwalten können.

Um Ihren Server so zu konfigurieren, dass eingehende SSH-Verbindungen zugelassen werden, können Sie diesen Befehl verwenden: ▣

```
sudo ufw allow ssh
```

Dadurch werden Firewall-Regeln erstellt, die alle Verbindungen an Port `22` zulassen; das ist der Port, an dem der SSH-Daemon standardmäßig lauscht. UFW weiß, was Port `allow ssh` bedeutet, da dies in der Datei `/etc/services` als Dienst aufgeführt wird.

Wir können die äquivalente Regel jedoch auch schreiben, indem wir den Port anstelle des Dienstnamens angeben. Dieser Befehl funktioniert zum Beispiel genauso wie oben: ▣

```
sudo ufw allow 22
```

Wenn Sie Ihren SSH-Daemon so konfiguriert haben, dass er einen anderen Port verwendet, müssen Sie den entsprechenden Port angeben. Wenn Ihr SSH-Server beispielsweise an Port `2222` lauscht, können Sie diesen Befehl verwenden, um Verbindungen an diesem Port zuzulassen:

```
sudo ufw allow 2222
```

Nachdem Ihre Firewall nun so konfiguriert ist, dass eingehende SSH-Verbindungen zugelassen werden, können wir sie aktivieren.

Schritt 4 — Aktivieren von UFW

Um UFW zu aktivieren, verwenden Sie diesen Befehl: ▣

```
sudo ufw enable
```

Sie erhalten eine Warnung, die besagt, dass der Befehl bestehende SSH-Verbindungen stören kann. Wir haben bereits eine Firewall-Regel eingerichtet, die SSH-Verbindungen zulässt. Daher sollte es in Ordnung sein, fortzufahren. Beantworten Sie die Eingabeaufforderung mit `y` und drücken Sie `ENTER`.

Die Firewall ist jetzt aktiv. Führen Sie den Befehl `sudo ufw status verbose` aus, um die festgelegten Regeln anzuzeigen. Im Rest des Tutorials wird die Verwendung von UFW im Detail behandelt, wie das Zulassen oder Ablehnen verschiedener Verbindungen.

Schritt 5 — Zulassen anderer Verbindungen

Jetzt sollten Sie alle anderen Verbindungen zulassen, auf die Ihr Server reagieren soll. Die Verbindungen, die Sie zulassen sollten, sind von Ihren spezifischen Bedürfnissen abhängig. Glücklicherweise wissen Sie bereits, wie Sie Regeln schreiben, die Verbindungen anhand eines Dienstnamens oder Ports zulassen. Das haben wir bereits für SSH an Port `22` getan. Sie können es auch tun für:

- HTTP an Port 80, was nicht verschlüsselte Webserver verwenden; mit `sudo ufw allow http` oder `sudo ufw allow 80`
- HTTPS an Port 443, was verschlüsselte Webserver verwenden; mit `sudo ufw allow https` oder `sudo ufw allow 443`

Es gibt weitere Möglichkeiten, um andere Verbindungen zuzulassen, abgesehen von der Angabe eines Ports oder bekannten Dienstes.

Spezifische Portbereiche

Sie können mit UFW spezifische Portbereiche angeben. Einige Anwendungen verwenden mehrere Ports anstelle eines einzelnen Ports.

Um zum Beispiel X11-Verbindungen zuzulassen, die Ports `6000` - `6007` verwenden, nutzen Sie diese Befehle: ▣

```
sudo ufw allow 6000:6007/tcp
sudo ufw allow 6000:6007/udp
```

Wenn Sie mit UFW Portbereiche angeben, müssen Sie das Protokoll (`tcp` oder `udp`) angeben, für das die Regeln gelten sollen. Wir haben das vorher nicht erwähnt, da wir ohne Angabe des Protokolls automatisch beide Protokolle zulassen, was in den meisten Fällen in Ordnung ist.

Spezifische IP-Adressen

Beim Arbeiten mit UFW können Sie auch IP-Adressen spezifizieren. Wenn Sie zum Beispiel Verbindungen von einer bestimmten IP-Adresse zulassen möchten, wie einer Arbeits- oder privaten IP-Adresse unter `203.0.113.4`, müssen Sie `from` und dann die IP-Adresse angeben: ▣

```
sudo ufw allow from 203.0.113.4
```

Sie können auch einen bestimmten Port angeben, mit dem die IP-Adresse eine Verbindung herstellen darf, indem Sie `to any port` (zu jedem Port) gefolgt von der Portnummer hinzufügen. Wenn Sie zum Beispiel `203.0.113.4` erlauben möchten, sich mit Port `22` (SSH) zu verbinden, verwenden Sie diesen Befehl: ▣

```
sudo ufw allow from 203.0.113.4 to any port 22
```

Subnetze

Wenn Sie ein Subnetz von IP-Adressen zulassen möchten, können Sie CIDR-Notation verwenden, um eine Netzmaske anzugeben. Wenn Sie zum Beispiel alle IP-Adressen im Bereich von `203.0.113.1` bis `203.0.113.254` zulassen möchten, können Sie diesen Befehl verwenden: ▣

```
sudo ufw allow from 203.0.113.0/24
```

Außerdem können Sie auch den Zielport angeben, mit dem das Subnetz `203.0.113.0/24` eine Verbindung herstellen darf. Auch hier verwenden wir Port `22` (SSH) als Beispiel: ▣

```
sudo ufw allow from 203.0.113.0/24 to any port 22
```

Verbindungen zu einer spezifischen Netzwerkschnittstelle

Wenn Sie eine Firewall-Regel erstellen möchten, die nur für eine bestimmte Netzwerkschnittstelle gilt, können Sie dazu „allow in on“ gefolgt vom Namen der Netzwerkschnittstelle angeben.

Sie möchten möglicherweise Ihre Netzwerkschnittstellen überprüfen, bevor Sie fortfahren. Dazu verwenden Sie diesen Befehl:

```
ip addr
```

Output Excerpt

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
...
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default
...
```

Die hervorgehobene Ausgabe gibt die Namen der Netzwerkschnittstellen an. Sie haben typischerweise Namen wie `eth0` oder `enp3s2`.

Wenn Ihr Server eine öffentliche Netzwerkschnittstelle namens `eth0` hat, könnten Sie HTTP-Verkehr (Port `80`) dorthin mit diesem Befehl zulassen: ▣

```
sudo ufw allow in on eth0 to any port 80
```

Dadurch würden Sie zulassen, dass Ihr Server HTTP-Anfragen aus dem öffentlichen Internet empfängt.

Oder wenn Sie möchten, dass Ihr MySQL-Datenbankserver (Port `3306`) an der privaten Netzwerkschnittstelle `eth1` nach Verbindungen lauschen soll, können Sie diesen Befehl verwenden: ▣

```
sudo ufw allow in on eth1 to any port 3306
```

Dadurch dürften andere Server in Ihrem privaten Netzwerk eine Verbindung mit Ihrer MySQL-Datenbank herstellen.

Schritt 6 — Ablehnen von

Verbindungen

Wenn Sie die Standardrichtlinie für eingehende Verbindungen nicht geändert haben, ist UFW so konfiguriert, dass alle eingehenden Verbindungen abgelehnt werden. Das vereinfacht im Allgemeinen das Erstellen einer sicheren Firewall-Richtlinie, da Sie Regeln erstellen müssen, die bestimmte Ports und IP-Adressen explizit zulassen.

Manchmal werden Sie jedoch einzelne Verbindungen auf Grundlage der Quell-IP-Adresse oder des Subnetzes ablehnen wollen, vielleicht weil Sie wissen, dass Ihr Server von dort angegriffen wird. Wenn Sie Ihre Richtlinie für eingehenden Datenverkehr in **allow** ändern möchten (was nicht empfohlen wird), müssten Sie für alle Dienste oder IP-Adressen, bei denen Sie keine Verbindung zulassen wollen, **deny**-Regeln erstellen.

Um **deny**-Regeln zu schreiben, können Sie die oben beschriebenen Befehle verwenden und **allow** durch **deny** ersetzen.

Um zum Beispiel HTTP-Verbindungen abzulehnen, können Sie diesen Befehl verwenden: ▣

```
sudo ufw deny http
```

Oder wenn Sie alle Verbindungen von `203.0.113.4` ablehnen möchten, können Sie diesen Befehl verwenden: ▣

```
sudo ufw deny from 203.0.113.4
```

Jetzt werfen wir einen Blick auf das Löschen von Regeln.

Schritt 7 — Löschen von Regeln

Zu wissen, wie man Firewall-Regeln löscht, ist genauso wichtig wie zu wissen, wie man sie erstellt. Es gibt zwei Wege, um anzugeben, welche Regeln gelöscht werden sollen: anhand der Regelnummer oder der tatsächlichen Regel (ähnlich wie beim Angeben der Regeln im Rahmen der Erstellung). Wir beginnen mit der Methode **Löschen anhand von Regelnummer**, da sie einfacher ist.

Nach Regelnummer

Wenn Sie die Regelnummer verwenden, um Firewall-Regeln zu löschen, wird eine Liste Ihrer Firewall-Regeln angezeigt. Der UFW-Statusbefehl hat eine Option, um neben jeder Regel eine Nummer anzuzeigen, wie hier gezeigt: ▣

```
sudo ufw status numbered
```

Numbered Output:

Status: active

To	Action	From
--	-----	----
[1] 22	ALLOW IN	15.15.15.0/24
[2] 80	ALLOW IN	Anywhere

Wenn wir entscheiden, dass wir Regel 2, die Verbindungen an Port 80 (HTTP) zulässt, löschen möchten, können wir sie in einem UFW-Befehl wie diesem angeben: ▣

```
sudo ufw delete 2
```

Dadurch würde eine Bestätigungsaufforderung angezeigt und Regel 2, die HTTP-Verbindungen zulässt, dann gelöscht. Beachten Sie, dass Sie bei aktiviertem IPv6 wahrscheinlich auch die entsprechende IPv6-Regel löschen möchten.

Nach tatsächlicher Regel

Die Alternative zu Regelnummern besteht darin, die tatsächlich zu löschende Regel anzugeben. Wenn Sie zum Beispiel die Regel `allow http` entfernen möchten, können Sie das wie folgt schreiben: ▣

```
sudo ufw delete allow http
```

Sie könnten die Regel auch anhand von `allow 80` anstelle des Dienstnamens angeben:

```
sudo ufw delete allow 80
```


Diese Methode löscht sowohl IPv4- als auch IPv6-Regeln, falls vorhanden.

Schritt 8 — Prüfen von UFW-Status und -Regeln

Sie können den Status von UFW mit diesem Befehl jederzeit überprüfen: ▣

```
sudo ufw status verbose
```

Wenn UFW deaktiviert ist, was standardmäßig der Fall ist, sehen Sie in etwa Folgendes:

Output

```
Status: inactive
```

Wenn UFW aktiv ist, was der Fall sein sollte, wenn Sie Schritt 3 ausgeführt haben, teilt die Ausgabe mit, dass UFW aktiv ist; zudem werden alle festgelegten Regeln aufgelistet. Wenn Sie die Firewall beispielsweise so einrichten, dass SSH (Port `22`)-Verbindungen überall zugelassen werden, könnte die Ausgabe ungefähr wie folgt aussehen:

Output

```
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
```

To	Action	From
--	-----	----
22/tcp	ALLOW IN	Anywhere

Verwenden Sie den Befehl `status`, um zu prüfen, wie UFW die Firewall konfiguriert hat.

Schritt 9 — Aktivieren oder Zurücksetzen von UFW (optional)

Wenn Sie entscheiden, dass Sie UFW nicht mehr verwenden möchten, können Sie die Firewall mit diesem Befehl deaktivieren:

```
sudo ufw disable
```

Alle Regeln, die Sie mit UFW erstellt haben, sind dann nicht mehr aktiv. Sie können später jederzeit `sudo ufw enable` nutzen, um sie wieder zu aktivieren.

Wenn Sie bereits UFW-Regeln konfiguriert haben, aber lieber neu anfangen möchten, können Sie den Befehl `reset` verwenden: ■

```
sudo ufw reset
```

Dadurch wird UFW deaktiviert und alle Regeln, die zuvor definiert wurden, werden gelöscht. Beachten Sie, dass die Standardrichtlinien nicht zu ihren ursprünglichen Einstellungen zurückkehren, wenn Sie sie irgendwann geändert haben. Jetzt sollten Sie mit UFW neu anfangen können.

Ping verbieten

Ping kann in der Datei `/etc/ufw/before.rules` deaktiviert werden.

```
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP
```

Revision #3

Created 21 March 2024 20:37:29 by Hermann

Updated 19 April 2024 10:00:08 by Hermann