Ubuntu

- User Administration
  - Einen neuen User erstellen
  - User mit sudo Berechtigung
- Webserver mit Apache
  - Installieren von Apache
  - Virtual Server einrichten
  - Sichern gegen DDOS
- Sicherheit
  - Einleitung
  - ufw Firewall installieren
  - fail2ban installieren

User Administration

Einen neuen User erstellen

Mit dem root User solltest du gleich als erstes einen neuen User erstellen, der sudo Rechte auf dem System hat. Aus Sicherheitsgründen solltest du nicht mit dem User root arbeiten. In diesem Beispiel lege ich einen User mit dem Namen testuser an.

```
adduser testuser
```

Es werden Ihnen einige Fragen gestellt, angefangen mit dem Passwort für das neue Konto. Es reicht wenn man den Full Name ausfüllt. Die anderen Optionen kann man mit Return einfach bestätigen.

```
Adding user `testuser' ...
Adding new group `testuser' (1000) ...
Adding new user `testuser' (1000) with group `testuser' ...
Creating home directory `/home/testuser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for testuser
Enter the new value, or press ENTER for the default
        Full Name []: testuser
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

https://www.youtube-nocookie.com/embed/keUbZC3V2Ik

User mit sudo Berechtigung

Mit dem Befehl sudo, kannst du normale User Administrationsberechtigungen erteilen, die normalerweise nur mit root möglich sind.

Bevor du einem User sudo Berechtigungen geben kannst, muss dieser angelegt werden.
Wenn du nicht weißt wie man einen Benutzer anlegst, schaue auf dieser Seite vorbei.

Mit dem Befehlt usermod fügst du einen Benutzer der Gruppe sudo hinzu. In diesem Beispiel der User testuser.

```
usermod -aG sudo testuser
```

Um zu überprüfen, ob alles richtig funktioniert hat, wechselst du von dem User root, zum User testuser.

```
su - testuser
```

Jetzt werden wir mal das Verzeichnis des Users root uns anzeigen lassen.

```
sudo ls -la /root
```

Jetzt wirst du aufgefordert das Passwort des Users testuser einzugeben.

```
[sudo] password for testuser:
```

Danach wird der Inhalt des Verzeichnis root angezeigt. Hier ein Beispiel:

```
total 52
drwx------  5 root root  4096 Jun 23 19:50 .
drwxr-xr-x 24 root root  4096 Jun 22 19:56 ..
-rw-------  1 root root  2508 Jun 22 21:35 .bash_history
-rw-r--r--  1 root root  3106 Aug 14  2019 .bashrc
drwx------  2 root root  4096 Jun 23 19:50 .cache
-rw-r--r--  1 root root   161 Aug 17  2020 .profile
-rw-r--r--  1 root root    20 Jun 22 11:02 .screenrc
drwx------  3 root root  4096 Jun 22 19:58 snap
drwxr-xr-x  2 root root  4096 Jun 23 19:50 .ssh
```

```
-rw-------  1 root root 12779 Jun 22 20:05 .viminfo
```

https://www.youtube-nocookie.com/embed/keUbZC3V2Ik

Webserver mit Apache

Installieren von Apache

Zuerst sollte auf dem System ein Update durchgeführt werden, damit vor der Installation das System auf einem aktuellen Stand ist.

```
sudo apt update && upgrade -y
```

Jetzt installierst du das Paket apache2.

```
sudo apt install apache2
```

Sobald der Installationsprozess durchgelaufen ist, startet Ubuntu Apache. Mit dem folgenden Befehl kannst du überprüfen, ob der Dienst gestartet wurde:

```
sudo systemctl status apache2
```

So sollte das Ergebnis des Befehles aussehen:

```
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2022-08-28 12:38:10 CEST; 9s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 1831 (apache2)
      Tasks: 55 (limit: 60)
     Memory: 9.2M
     CGroup: /system.slice/apache2.service
             ├─1831 /usr/sbin/apache2 -k start
             ├─1833 /usr/sbin/apache2 -k start
             └─1834 /usr/sbin/apache2 -k start
```

In einem Browser gibst du jetzt die IP Adresse deines Servers ein:

```
http://deine-ip-adresse
```

Danach solltest du diese Seite sehen. Sie enthält einige Informationen über Verzeichnispositionen und Dateien.
Somit ist dein Apache Server installiert.

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers **`a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`,** and **`a2enconf`, `a2disconf`** . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

https://www.youtube.com/embed/2hZ1E4LgQy8

Virtual Server einrichten

Wenn Apache installiert wird, kannst du erst einmal núr eine Domäne dort einstellen die erreichbar ist. Was ist aber wenn du mehr als eine Domäne hast und diese auf einen Server hosten möchtest. Dazu musst du einige Veränderungen vornehmen.
Als erstes legen wir für jede Domäne ein Verzeichnis an und ändern die Berechtigungen.

```
sudo chown -R $USER:$USER /var/www/techdoctest.de
sudo chown -R $USER:$USER /var/www/techdoctest2.de
sudo chmod -R 755 /var/www/techdoctest.de
sudo chmod -R 755 /var/www/techdoctest2.de
```

Jetzt legen wir in jedem Verzeichnis eine index.html Datei an.

```
sudo vi /var/www/techdoctest.de/index.html
```

```
<html>
    <head>
        <title>Welcome to Techdoctest.de</title>
    </head>
    <body>
        <h1>Success!  The techdoctest.de virtual host is working!</h1>
    </body>
</html>
```

```
sudo vi /var/www/techdoctest2.de/index.html
```

```
<html>
    <head>
        <title>Welcome to techdoctest2.de</title>
    </head>
    <body>
        <h1>Success!  The techdoctest2.de virtual host is working!</h1>
    </body>
</html>
```

Jetzt muss noch für jeder der Domänen eine Konfigurationsdatei angelegt werden.

```
cd /etc/apache2/sites-available/
```

```
sudo vi techdoctest.de.conf
```

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName techdoctest.de
    ServerAlias www.techdoctest.de
    DocumentRoot /var/www/techdoctest.de
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
sudo vi techdoctest2.de.conf
```

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName techdoctest2.de
    ServerAlias www.techdoctest2.de
    DocumentRoot /var/www/techdoctest2.de
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Jetzt müssen diese Konfigurationsdateien aktiviert werden.

```
sudo a2ensite techdoctest.de.conf
sudo a2ensite techdoctest2.de.conf
```

Damit die default Datei nicht mehr aktiv ist, musst du diese deaktivieren.

```
sudo a2dissite 000-default.conf
```

Bevor du jetzt einen Restart des Apache Prozess machst, überprüfst du erst mal ob deine Konfigurationsdateien mit den Parametern in Ordnung sind.

```
sudo apache2ctl configtest
```

Ist alles in Ordnung, bekommst du diese Meldung zurück.

```
Syntax OK
```

Danach machst du einen Restart auf dem Apache Server.

```
sudo systemctl restart apache2
```

Jetzt kannst du beide Domänen erreichen.

https://www.youtube-nocookie.com/embed/86f02Dz3wuo

Sichern gegen DDOS

Zuerst sollte auf dem System ein Update durchgeführt werden, damit vor der Installation das System auf einem aktuellen Stand ist.

```
sudo apt update && apt upgrade -y
```

Jetzt installierst du das Paket libapache2-mod-evasive.

```
sudo apt install libapache2-mod-evasive
```

Jetzt musst du überprüfen ob der Mod installiert und aktiviert wurde.

```
sudo apachectl -M | grep evasive
```

Die Antwort die zurück kommen sollte sieht so aus

```
$ evasive20_module (shared)
```

In der Konfigurationsdatei evasive.conf kannst du jetzt einige EInstellungen vornehmen. Die default Einstellungen reichen aber hier aus.

```
sudo vi /etc/apache2/mods-enabled/evasive.conf
```

Hier die default Einstellung der Datei:

```
<IfModule mod_evasive20.c>
     DOSHashTableSize 3097
     DOSPageCount 2
     DOSSiteCount 50
     DOSPageInterval 1
     DOSSiteInterval 1
     DOSBlockingPeriod 10
     DOSEmailNotify info@example.com
     DOSSystemCommand "su - someuser -c '/sbin/... %s ...'"
     DOSLogDir "/var/log/mod_evasive"
 </IfModule>
```

Erklärung der Parameter:

DOSHashTableSize	Die Grösse der HashTables pro child-prozess (und somit pro Verbindung) Grössere HashTables = hoehere Geschwindigkeit = mehr RAM-Verbrauch
DOSPageCount	Anzahl der Requests pro Seite innerhalb des mit DOSPageInterval angegebenen Zeitraumes, nach denen weitere Verbindungen unterbunden werden.
DOSSiteCount	Anzahl der Requests pro Website innerhalb des mit DOSSiteInterval angegebenen Zeitraumes, nach denen weitere Verbindungen unterbunden werden.
DOSPageInterval	Der Zeitraum (in Sekunden) in dem nicht mehr Requests pro Seite, als in DOSPageCount angegeben, erlaubt sind.
DOSSiteInterval	Der Zeitraum (in Sekunden) in dem nicht mehr Requests pro Website, als in DOSSiteCount angegeben, erlaubt sind.
DOSBlockingPeriod	Der Zeitraum (in Sekunden) für den eine Sperre aufrecht erhalten wird. Der Zähler wird bei erneuten Anfragen innerhalb dieses Zeitraumes zurückgesetzt, weshalb hier sehr niedrige Werte ausreichend sind.
DOSLogDir	Wo die HashTables abgelegt werden sollen. (Default ist "/tmp" was auf Systemen mit mehreren Shell-Usern ein Sicherheitsrisiko ist, da dieses Verzeichnis von jedermann gelesen werden kann.

Verständlich ausgedrückt bedeutet das in unserem Beispiel, dass jeder für 10 Sekunden geblockt würde, der öfter als zwei mal pro
Sekunde die gleiche Seite oder mehr als 50 mal pro Sekunde irgendeine Seite auf einer WebSite aufruft. Wenn er dann innerhalb dieser 10 Sekunden wieder versucht, eine Seite aufzurufen, bekommt er eine 403 Fehlermeldung präsentiert und der Zähler wird wieder zurückgesetzt.

Jetzt musst du noch einen Restart auf dem Apache Server machen

```
sudo systemctl restart apache2
```

Jetzt kannst du einen Test durchführen, ob alles funktioniert. Dazu machst du ein:

```
sudo perl /usr/share/doc/libapache2-mod-evasive/examples/test.pl
```

Jetzt bekommst du etliche Bad Request Antworten:

```
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
```

```
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
HTTP/1.1 400 Bad Request
.
.
.
```

Somit sollte dein Server jetzt gut gegen DDOS Attacken geschützt sein.

Sicherheit

Einleitung

Sobald du einen neuen Server installiert hast und dieser im Internet öffentlich ist, wirst du nach einigen tagen feststellen, das sich einige versuchen darauf zu verbinden. Hier einer meiner Server, wie das aussehen kann:

```
cd /var/log
```

Dort schaust du einfach in das Logfile auth.log und kannst solche Einträge finden:

```
Oct 16 00:06:24 linux-test sshd[31407]: pam_unix(sshd:auth): authentication failure; logname=
uid=0 euid=0 tty=ssh ruser= rhost=92.255.85.69  user=root
Oct 16 00:06:26 linux-test sshd[31407]: Failed password for root from 92.255.85.69 port 29672
ssh2
Oct 16 00:06:26 linux-test sshd[31407]: Received disconnect from 92.255.85.69 port 29672:11:
Bye Bye [preauth]
Oct 16 00:06:26 linux-test sshd[31407]: Disconnected from authenticating user root
92.255.85.69 port 29672 [preauth]
Oct 16 00:17:01 linux-test CRON[31412]: pam_unix(cron:session): session opened for user
root(uid=0) by (uid=0)
Oct 16 00:17:01 linux-test CRON[31412]: pam_unix(cron:session): session closed for user root
Oct 16 00:38:17 linux-test sshd[31424]: Invalid user nagios from 92.255.85.69 port 35922
Oct 16 00:38:17 linux-test sshd[31424]: pam_unix(sshd:auth): check pass; user unknown
```

Hier versucht jemand, das Passwort für den Root-Benutzer zu erzwingen. (Es spielt keine Rolle, ob die Passwortanmeldung für diesen Benutzer verboten ist. Er kann es trotzdem versuchen, nur wird er jedes Mal scheitern).

Und darum dreht es sich hier in diesen Kapiteln. Der Server soll sicherer gemacht werden, damit den Angreifern keine Möglichkeit für einen Erfolg gegeben werden kann.

Die Kapitel hier bringen sicher nicht den 100 % Schutz. Du kannst mit diesen Methoden die Möglichkeiten der Angriffe minimieren.

Sicherheit

ufw Firewall installieren

Du solltest immer eine Firewall auf deinem Linux System installieren. Hier empfiehlt sich die Uncomplicated Firewall (ufw).

```
sudo apt install ufw
```

Du solltest die Firewall auf keinem Fall einschalten, bevor du deinen ssh Port nicht frei geschalte

Ich gehe in diesem Beispiel auf dem Standard Port für ssh aus. Solltest du schon einen anderen port vergeben hast, musst du diesen eintragen.

```
sudo ufw allow ssh
```

Als Bestätigung erhältst du hier

```
Rules updated
Rules updated (v6)
```

Wenn du einen Apache Server installiert hast, musst du auch diesen Port frei geben für http und/oder https.

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw allow www
```

Damit die Einstellungen aktiv werden, machst du jetzt ein

```
sudo ufw enable
```

Jede Applikation die einen bestimmten Port braucht, musst du jetzt frei geben.

https://www.youtube-nocookie.com/embed/Axsh4a6FUNQ

Sicherheit

fail2ban installieren

Mit fail2ban kannst du sehr leicht und einfach Brute-Force-Angriffe auf deinem System abwehren.

```
sudo apt install fail2ban
```

jetzt wird die gezeigt, was alles mit dem Paket fail2ban installiert wird. Das bestätigst du mit J wenn du fail2ban installieren möchtest.

```
Paketlisten werden gelesen… Fertig
Abhängigkeitsbaum wird aufgebaut… Fertig
Statusinformationen werden eingelesen… Fertig
Die folgenden zusätzlichen Pakete werden installiert:
  python3-pyinotify whois
Vorgeschlagene Pakete:
  mailx monit sqlite3 python-pyinotify-doc
Die folgenden NEUEN Pakete werden installiert:
  fail2ban python3-pyinotify whois
0 aktualisiert, 3 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
Es müssen 473 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 2.486 kB Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren? [J/n] j
```

Wenn fail2ban installiert wurde, sieht das danach so aus.

```
Holen: 1 http://de.archive.ubuntu.com/ubuntu jammy/universe amd64 fail2ban all 0.11.2-6 [394
kB]
Holen: 2 http://de.archive.ubuntu.com/ubuntu jammy/main amd64 python3-pyinotify all 0.9.6-1.3
[24,8 kB]
Holen: 3 http://de.archive.ubuntu.com/ubuntu jammy/main amd64 whois amd64 5.5.13 [53,4 kB]
Es wurden 473 kB in 0 s geholt (2.986 kB/s).
Vormals nicht ausgewähltes Paket fail2ban wird gewählt.
(Lese Datenbank ... 108933 Dateien und Verzeichnisse sind derzeit installiert.)
Vorbereitung zum Entpacken von .../fail2ban_0.11.2-6_all.deb ...
Entpacken von fail2ban (0.11.2-6) ...
Vormals nicht ausgewähltes Paket python3-pyinotify wird gewählt.
```

```
Vorbereitung zum Entpacken von .../python3-pyinotify_0.9.6-1.3_all.deb ...
Entpacken von python3-pyinotify (0.9.6-1.3) ...
Vormals nicht ausgewähltes Paket whois wird gewählt.
Vorbereitung zum Entpacken von .../whois_5.5.13_amd64.deb ...
Entpacken von whois (5.5.13) ...
whois (5.5.13) wird eingerichtet ...
fail2ban (0.11.2-6) wird eingerichtet ...
python3-pyinotify (0.9.6-1.3) wird eingerichtet ...
Trigger für man-db (2.10.2-1) werden verarbeitet ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Somit ist fail2ban auf deinem System installiert.

https://www.youtube-nocookie.com/embed/0NGVntL0sMM