# 1. Systeembeheer 2 - Labo Linux server

## 1.1. Inleiding

Goed nieuws! Je hebt besloten een eigen hostingbedrijf op te richten FamilienaamHosting (naam veranderen). Je legt je toe op Linux hosting waarbij je wil evolueren naar een software stack van Apache, MySQL en PHP.

In labo Linux Server maken we gebruik van de CentOS distributie. Deze distro is eigenlijk een gratis versie van Red Hat, wat ons het voordeel geeft dat we dezelfde documentatie kunnen gebruiken en een grote community hebben als er moeilijkheden zijn.

De meest recente versie is CentOS 7. Deze maakt gebruik van het nieuwe standaard init systeem *systemd*. Met systemd kan je de meeste services beheren op een linux systeem. De syntax om services te beheren is: **systemctl <commando> <naam service>** Enkele voorbeelden:

systemctl start sshd #start de SSH-daemon

systemctl restart network #herstart de network service

Om een lijst te krijgen van alle mogelijkheden kan je zoals bij de meeste CLIs tab gebruiken om een deel van het commando aan te vullen. De hulppagina van de meeste commando’s kan je opvragen met **man <commando>** of **help <commando>**

Buiten systemctl heb je nog andere commando’s waarmee we het systeem kunnen configureren. Tijdens deze oefeningen ga je in aanraking komen met verschillende mogelijkheden om jouw server te beheren. Probeer steeds de systemd architectuur te gebruiken. Bij twijfel raadpleeg je best jouw docent.

## 1.2. Basisconfiguratie

We maken gebruik van VCloud om onze server te bouwen. Surf naar <https://vmcloud.ikdoeict.be/cloud/org/systeembeheer2/> en voeg een bestaande vApp toe (Add vApp from Catalog) en noem deze **2ICTX-L-VoornaamAchternaam** (met X de nummer van je klas). In de public catalog vind je de template **tpl labo linux server**. In deze template vind je een CentOS server en een Debian client. Laat alle opties tijdens de wizard op hun default waarde staan.

Op beiden staat een **gebruiker student** met default **wachtwoord student**. Het **root wachtwoord** is ook **student**. Het spreekt voor zich dat dit geen best practice is. Voordat we de setup doen van alle services gaan we onze server en client een basisconfiguratie geven.

### 1.2.1. Netwerkconfig

In CentOs 7 staan ethernet interfaces standaard niet geconfigureerd. In dit stuk gaan we ervoor zorgen dat onze server en client netwerkverbinding maken met het Odisee-netwerk en bij uitbreiding het internet door statische configuratie. Een lijst van alle interfaces kan je oproepen met de Network Manager CLI. Gebruik het commando **nmcli device**. Noteer hieronder de output:

|  |
| --- |
|  |

Bovenstaand commando maakt gebruik van de NetworkManager. In onze verdere setup gaan we die niet gebruiken, dus schakel die daemon uit. **systemctl stop NetworkManager**

De IP-configuratie van alle interfaces op je server vind je in **/etc/sysconfig/network-scripts** Let er op dat je de juiste interface kiest uit het lijstje hierboven.

Controleer op je server of de settings overeenkomen met deze vereisten:

* Zorg ervoor dat het MAC-adres van je machines overeenkomt met het MAC-adres in Vcloud (VM properties > tab hardware > NICs)
* Stel het IP-adres van je server statisch in op het adres dat je krijgt van VCloud
* De subnetmask is 255.255.248.0
* De default gateway is 10.129.39.254
* Als DNS-servers stel je 10.129.28.230 en 10.129.28.232 in
* Verander de naam van je server naar srv1.familienaam.ikdoeict. Maak hiervoor gebruik van **hostnamectl**

Configuratiefiles zoals die van de interfaces worden pas effectief van kracht als de bijhorende service herstart wordt. Herstart de netwerkservice voor je verder gaat. Test jouw opstelling door de volgende tests uit te voeren:

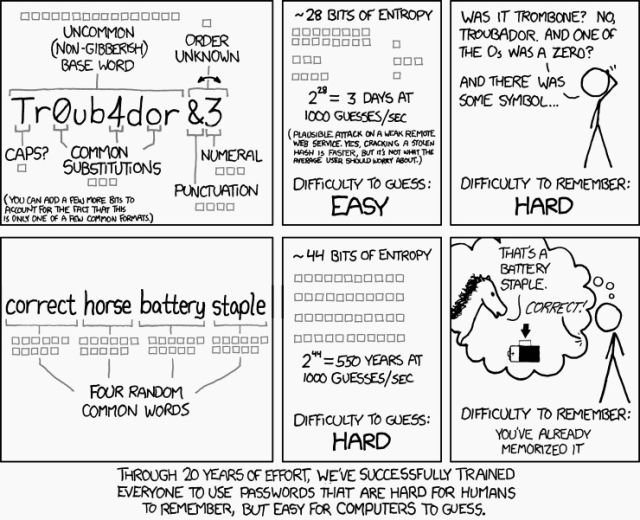
* ping naar de gateway: 10.129.39.254
* ping naar de DNS-server van google: 8.8.8.8
* ping naar een andere website

Pas als al deze testen lukken, ga je verder.

Stel nu op je **client** de IP-gegevens statisch in zodat die overeenkomen met wat in VCloud staat. Let wel op: dit moet op een andere manier gebeuren als bij CentOS. Controleer ook op je client of je kan pingen en surfen.

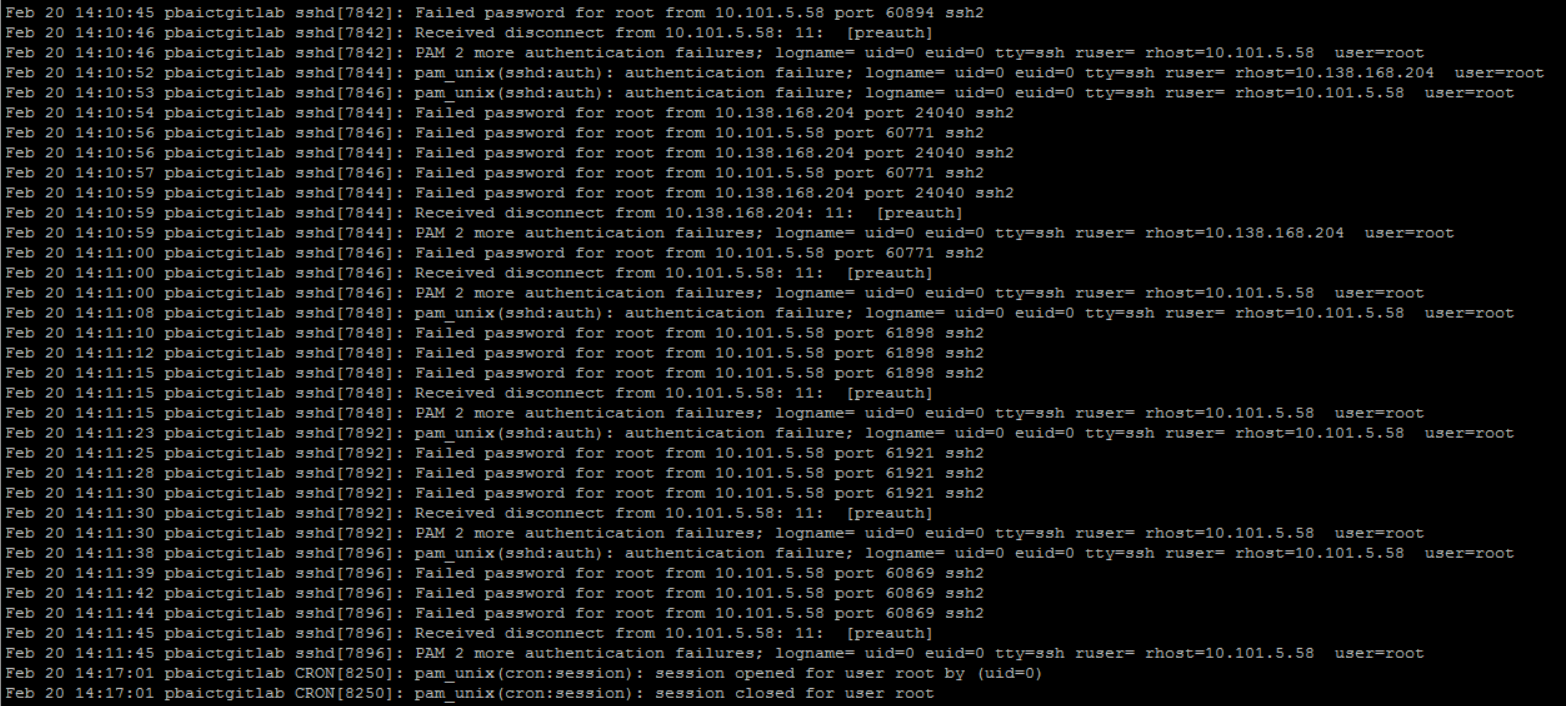
### 1.2.2. Remote toegang

Om je server te beheren maken we onder Linux gebruik van SSH. Out-of-the-box is de SSH daemon reeds gestart en kan je met de student-gebruiker inloggen (probeer het maar eens uit met bijvoorbeeld putty of mRemoteNG op jouw labopc of vanaf je linux client). Bij deze manier van authenticatie ben je sterk afhankelijk van hoe complex een gebruiker zijn wachtwoord is om de integriteit van jouw server te bewaken.



Bron: https://xkcd.com

Linux-servers die toegankelijk zijn van op het internet worden constant aangevallen om toegang te verkrijgen tot die machine. Uit een logfile op git.ikdoeict.be (Debian) kan je dit heel duidelijk zien:



Mislukte aanmeldpogingen voor de root-gebruiker op git.ikdoeict.be

Waar vind je de logfile op je server met aanmeldpogingen via SSH? Hoe wordt een mislukte poging gelogd?

|  |
| --- |
|  |

Je hebt een aantal manieren om jouw machine te beschermen. Een populair open source verweermiddel is fail2ban. Op welke manier werkt deze tool?

|  |
| --- |
|  |

Installeer fail2ban op jouw server. De twee belangrijkste configuratiebestanden zijn fail2ban.conf en jail.conf. Bekijk de inhoud van beide bestanden om volgende vragen te beantwoorden..

Na hoeveel mislukte aanmeldpogingen wordt jouw IP-adres geblokkeerd met de standaardsettings? En voor hoelang is dat dan?

|  |
| --- |
|  |

Hernoem jail.conf naar jail.local en pas de volgende zaken aan:

* zorg ervoor dat requests van 10.129.32.0/21 niet gecontroleerd worden. Op die manier zal onze client nooit uitgesloten worden.
* Beperk het aantal foutieve aanmeldpogingen tot 3
* Verwijder alle jails die geen betrekking hebben tot ssh
* enable het default profiel om aan te geven dat je deze settings wilt gebruiken

Start nu fail2ban. Als de service niet wil starten, ga zelf op zoek naar het probleem in /var/log/messages of /var/log/fail2ban.log

Probeer meer dan 3 keer foutief aan te melden vanaf je labopc (met bijvoorbeeld putty). Welke melding krijg je?

|  |
| --- |
|  |

Je kan de firewall regel die jouw IP blokkeert zien met het commando **iptables -L**.IPtables is de standaardfirewall op linux.

Noteer hieronder de firewall regel die jouw verkeer blokkeert

|  |
| --- |
|  |

Vanaf je Linux client zou fail2ban geen effect mogen hebben.

### 1.2.3. SSH key-based authentication

In deze setup gaan we gebruik maken van key-based authentication. Deze manier van authenticatie is veel minder gevoelig voor brute force aanvallen en moeilijker te kraken als je een voldoende lange key gebruikt. De werkwijze is als volgt:

* Genereer een public en private key van **4096 bits** op je Linux client.
* Transfereer jouw public key naar de server en zorg ervoor dat die gekoppeld wordt aan de student gebruiker

Tip: Als je vanuit Windows dit systeem wil gebruiken, kan je hiervoor de Git Bash gebruiken. Deze emuleert Linux-commando’s op Windows.

Noteer hieronder de commando’s die je gebruikte:

|  |
| --- |
|  |

Probeer nu eens een SSH-verbinding te maken met jouw server vanop de client. Als alles goed gegaan is, heb je een verbinding zonder dat je een gebruikersnaam of wachtwoord moet opgeven.

Opgelet: Als je jouw commando’s in Git Bash op de labopc uitgevoerd hebt, zal dit momenteel enkel van daaruit lukken. Om dit ook vanuit putty of mremoteng te kunnen, zal je met behulp van PuTTY, PuTTYgen en Pageant een key-pair moeten genereren en importeren in jouw server.

Op welke locatie (bestand) in je server wordt bijgehouden welke public keys aanvaard worden voor de ‘student’-gebruiker?

|  |
| --- |
|  |

## 1.3. Linux Filesystem Hierarchy Standard (FHS)

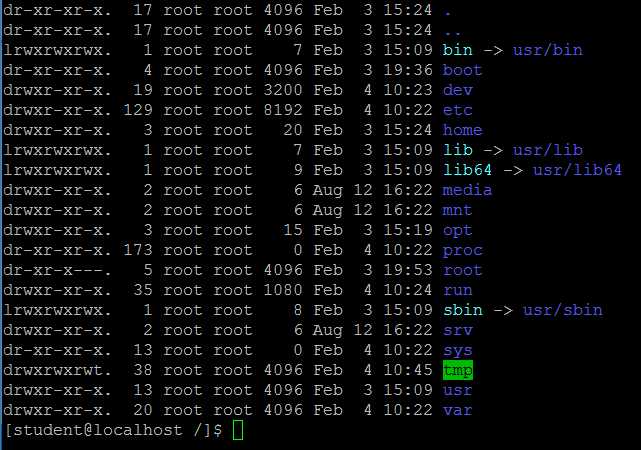
Om vlot met een Linux-systeem te kunnen werken, is het zinvol om even stil te staan bij de organisatie van alle bestanden en mappen. Om de organisatie van bestanden te standaardiseren werd FHS ontwikkeld. Je zal dus deze structuur terugvinden bij de meeste Linux-distributies. De bedoeling van FHS is om elk bestand onder te verdelen zodat die in onderstaande matrix passen:

|  |  |  |
| --- | --- | --- |
|  | Shareable | Unshareable |
| Static | /usr  /opt | /etc  /boot |
| Variable | /home  /var/mail | /var/run  /var/lock |

Unshareable bestanden zijn bestanden die eigen zijn aan het systeem (zoals programma’s, configuratiebestanden, …).

Statische bestanden wijzigen bijna nooit, tenzij door een sysadmin. Variabele bestanden zijn gebruikersbestanden, logs, … Bestanden die aangepast worden door gebruikers, programma’s of scripts.

De hoofdmap waar elk Linux filesystem mee begint is / (uitgesproken als de root). In de root van CentOS vind je de volgende mappen terug:



De FHS definieert het gebruik van deze folders als volgt (beperkte selectie, voor een uitgebreidere lijst: zie theorie):

**/bin** bevat kritieke uitvoerbare bestanden eigen aan het besturingssysteem. Voorbeeld zijn ls, cp, mount. Deze commando’s zijn toegankelijk voor alle gebruikers. Op CentOS is dit eigenlijk een link naar /usr/bin/

**/boot** bevat bestanden noodzakelijk voor het opstarten

**/dev** Omdat alles in Linux te herleiden valt tot een bestand, moeten alle hardware devices een overeenkomstig bestand hebben. Je kan dit beschouwen als de map met drivers, al klopt dat niet helemaal.

**/etc** deze folder bevat alle configuratiebestanden van de software of van het systeem. /etc/yum bevat bijvoorbeeld de default configuratie van RPM (RedHat Package Manager) waarmee je software kunt installeren.

**/home** bevat gebruikersdata. /home/student is waar je terechtkomt als je inlogt met de student-gebruiker

**/proc** is een speciaal geval. Deze folder is eigenlijk geen directory maar een virtual filesystem om hardware informatie te voorzien. cat /proc/cpuinfo zal je gedetailleerde informatie geven over de processor bijvoorbeeld

**/root** de home-directory van de root-gebruiker

**/sbin** lijkt op /bin maar bevat programma’s die normaal enkel de sysadmin uitvoert zoals fdisk. Op CentOs is dit een link naar /usr/sbin/

**/usr** bevat het grootste stuk van de software op Linux. De inhoud van de mappen /usr/bin en /usr/sbin komt overeen met wat je vind in de mappen /bin en /sbin op CentOS.

**/var** Zoals de naam doet vermoeden, is het gebruik van deze folder variabel. /var/log bevat de logfiles. In een vorig stuk gebruikten we bijvoorbeeld /var/log/audit/audit.log om te zien wat er geblokkeerd werd door SELinux

Met alles wat je nu weet over de opbouw van het file system, waar vind je de volgende zaken?

|  |  |
| --- | --- |
| **Item** | **Locatie (volledig pad!)** |
| Informatie over het geheugen |  |
| Configuratie van firewalld (niet de regels zelf) |  |
| Programma om ssh keys te genereren |  |
| Desktop van de student-gebruiker |  |

**Controlepunt 1: Laat jouw labo tot nu controleren door je docent**

## 1.4. DNS (Bind)

Je besluit om voor jouw hostingoplossing zelf DNS-resolving te gaan doen om domeinen, sub-domeinen en records aan toestellen of sites bereikbaar te maken. Om onze server deze rol te laten opnemen moeten we een DNS-package installeren. Voor Linux is de meest gebruikte DNS software Bind. Voer volgende stappen uit:

* Installeer Bind en maak onderstaande configuratie

Waar vind je het configuratiebestand?

|  |
| --- |
| /etc/named.conf |

* Configureer de server in eerste instantie als forwarding server zodat:
  + enkel requests van 10.129.32.0/21 mogen verwerkt worden
  + dns-request voor sites die we zelf niet hosten geforward worden naar 10.129.28.232 en 10.129.28.230 (DNS-servers van de opleiding)
  + Schakel DNSsec uit. Dit komt in een later vak aan bod.
  + Verander op je client én op de server de DNS-settings zodat jouw server gebruikt wordt als primaire DNS-server.
  + Test jouw opstelling door te pingen naar [www.google.be](http://www.google.be). Als dit niet werkt, troubleshoot jouw oplossing.
* De volgende stap is dat we een bestand aanmaken voor de zone familienaam.ikdoeict. Deze DNS-server is de master.
  + Maak een nieuw bestand **familienaam.ikdoeict** aan (of kopieer een bestaand) in /var/named/
  + koppel de zone familienaam.ikdoeict aan dit bestand en maak jouw server master voor deze zone.
  + In het bestand familienaam.ikdoeict maak je nu een nieuwe forward zone aan familienaamhosting.ikdoeict
  + Maak binnen de zone de volgende records aan:
    - SOA-record
    - ns-record
    - a-record voor de server (waarde: srv1)
    - alias-record voor de server (waarde: www) Dit zal later van pas komen bij de configuratie van onze webserver.
* Test jouw opstelling door vanaf je client te pingen naar srv1.familienaam.ikdoeict. Controleer of het correcte IP-adres geresolved wordt.
* Zorg er tenslotte voor dat de service automatisch gestart wordt als de server opstart. Hoe heb je dit gedaan?

|  |
| --- |
| Systemctl enable named |

## 1.5. Firewall

Met de systemd architectuur wordt ook een firewall meegeleverd, firewalld. We gaan de ingebouwde firewall zo configureren dat enkel het verkeer toegelaten wordt dat we echt nodig hebben.

Controleer eerst of de firewall wel actief is. Gebruik hiervoor opnieuw systemctl. Noteer hieronder jouw commando:

|  |
| --- |
| Systemctl status firewalld |

Firewalld is een zone-based firewall. In een zone zitten interfaces. Tot welke zone hoort de interface die we eerder configureerden? Maak gebruik van het commando firewall-cmd.

|  |
| --- |
| [root@srv1 ~]# firewall-cmd --get-active-zones  public  interfaces: eno16780032 |

Welke zones bestaan er nog allemaal?

|  |
| --- |
| [root@srv1 ~]# firewall-cmd --get-zones  block dmz drop external home internal public trusted work |

We gaan ons momenteel beperken tot het veranderen van de default zone. Aangezien we van deze server een web server willen maken, moeten we alvast een regel toevoegen die HTTP-verkeer toelaat. Zorg ervoor dat deze regel permanent opgeslagen wordt. Jouw commando is:

|  |
| --- |
| [root@srv1 ~]# firewall-cmd --zone=public --permanent --add-service=http  success  [root@srv1 ~]# firewall-cmd --zone=public --permanent --add-service=dns  success |

Herstart de firewall-service om de regel actief te maken en controleer of jouw regel toegepast wordt. Vraag een lijst op van alle actieve, permanente regels binnen de default zone. Noteer je commando en de output:

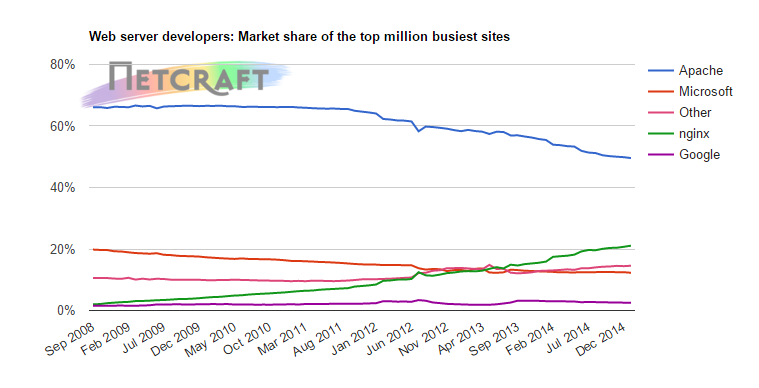
|  |
| --- |
| [root@srv1 ~]# firewall-cmd --zone=public --permanent --list-services  dhcpv6-client dns http ssh |

**Controlepunt 2: Laat jouw labo tot nu controleren door je docent**

## 1.6. Webserver (Apache)

### 1.6.1. Installatie

Apache is één van de meest gebruikte webservers ter wereld door zijn hoge configureerbaarheid en uitbreidbaarheid. Andere webservers zoals Nginx en IIS worden ook vaak gebruikt, al blijft Apache voorlopig dominant. Ter illustratie deze grafiek met de situatie tot Januari 2015:



Bron: Netcraft.com

Installeer nu Apache op je server en zorg ervoor dat die service automatisch opstart als je server reboot. Test jouw installatie door te surfen naar het IP-adres van jouw server vanop de labopc, of naar <http://srv1.familienaam.ikdoeict> vanaf je client. Als je de welkomstpagina te zien krijgt, is de installatie geslaagd.

### 1.6.2. Configuratie

Zorg er eerst voor dat de standaardpagina verdwijnt. Volg hiervoor de instructies op die pagina zelf.

Stel vervolgens de volgende basissettings van jouw webserver in:

* Stel jouw Odisee-emailadres in als serveradmin
* Stel de naam van jouw webserver in op [www.familienaam.ikdoeict](http://www.familienaam.ikdoeict) en luister op poort 80. Hiervoor hebben we al een DNS-record aangemaakt in een eerdere opdracht.
* Schakel de mogelijkheid uit om directory’s op te lijsten uit in de “DocumentRoot”
* Door het bovenstaande correct uit te voeren krijg je 403. Zorg voor een aangepaste foutmelding met de melding om jou, de beheerder, te contacteren.

Bij de directory statements zie je vaak “Require all granted” staan. Wat betekent dit?

|  |
| --- |
| Alle domeinen worden behandeld, kan aangepast worden naar specifieke domeinen |

Hoe zou je het exact omgekeerde resultaat kunnen bereiken?

|  |
| --- |
| Require all denied |

Met het directory statement kan je de toegang en verwerking van mappen door de webserver op de lokale schijf beheren. Als webhosting bedrijf is het onze bedoeling om voor meerdere klanten, elk met hun eigen domein, websites te hosten. Om dit te kunnen configureren gaan we gebruik maken van virtualhosts. Wat is het voordeel van virtualhosts?

|  |
| --- |
| Via virtualhosts kunnen meerdere websites op 1 server worden gehost. Dit kan op 2 manieren: Ip based of name based. Ip based wil zeggen dat elke website op de server een ander ip adres krijgt, name based betekent dat een ip adres op de server meerdere names krijgt. |

Om modulair te werken plaats je een nieuw bestand in /etcc/httpd/conf.d/ en noem dat vhosts.conf. Om wat voeling te krijgen met deze manier van werken gaan we de fictieve website van ons bedrijf zelf uitwerken. Maak binnen het vhosts.conf bestand een virtualhost met de volgende eigenschappen:

* Luister op poort 80
* De serveradmin is jouw emailadres
* De bestanden staan in cfamilienaam.ikdoeict. Maak hierin een index.html pagina met een duidelijke beschrijving van jouw bedrijf.
* Zorg voor een eigen error logfile voor deze virtual host.

## 1.7. Jouw eerste klant

Om onze eerste klant te simuleren, download je een site template. Bijvoorbeeld van één van deze sites:

* <http://www.oswd.org/>
* <http://solucija.com/templates/free>

Met alles wat je tot nu weet, maak de volgende configuratie:

* de website moet bereikbaar zijn vanaf je client via het adres [www.klant1.ikdoeict](http://www.client1.ikdoeict)
* de code plaats je in een subdirectory onder /var/www/html/vhosts/klant1.ikdoeict
* Configureer de webserver en maak gebruik van een virtualhost zoals we hierboven deden.
* Configureer DNS zodat je kan surfen van je client naar deze website. Je zal hiervoor een nieuwe zone moeten configureren.

**Controlepunt 3: Laat jouw labo tot nu controleren door je docent**

## 1.8. Services

Je bedrijf groeit en besluit je scala aan services uit te breiden om ook dynamische websites te hosten. Hiervoor gaan we de dynamische scriptingtaal PHP gebruiken en MySQL opzetten als databankserver.

### 1.8.1. MySQL

MySQL werd een tijd terug opgekocht door Oracle waardoor dit stuk software strikt gezien niet meer open source is. CentOS heeft daarom besloten om MySQL te vervangen door MariaDB, een fork van de originele MySQL.

Wij willen toch MySQL gebruiken, maar dan de Community Edition die vrijgegeven werd onder de GPL licentie. Doordat we geen repository hebben met daarin MySQL, gaan we dit pakket via de Red Hat Package Manager (RPM) installeren.

Voer de volgende stappen uit:

* Download de RPM-file van MySQL op je server
* Installeer de RPM
* Installeer de laatste versie van de MySQL-community-server
* Start tenslotte MySQL service

Noteer hieronder de commando’s die je gebruikte:

|  |
| --- |
| wget <https://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm>  sudo rpm -ivh mysql57-community-release-el7-8.noarch.rpm  sudo yum install mysql-community-server  sudo systemctl enable mysqld |

Bij de eerste start van de MySQL-service wordt er een tijdelijk wachtwoord gegenereerd voor de root gebruiker. Dit vind je terug in /var/log/mysqld.log

Log in met deze account en verander het wachtwoord van de root gebruiker. Noteer hieronder jouw gekozen wachtwoord ter referentie (gebruik geen bestaand wachtwoord ;-)):

|  |
| --- |
| QSDFjklm123 |

Installeer MySQL en voer nadien de secure installation uit. Wat doet deze laatste stap?

|  |
| --- |
| overloopt enkele veiligheidsstappen. |

**Automatisatie**

Om snel nieuwe klanten te kunnen voorzien gaan we de creatie van een databank, de bijhorende user en de rechten automatiseren via een Python script.

Python staat standaard geïnstalleerd op jouw server. Om de MySQL databank te kunnen manipuleren, installeer je het “MySQL-python” pakket.

Maak een nieuwe file aan /opt/hostingscripts/createdb.py en vertrek vanaf onderstaande code:

MYSQL

#!/usr/bin/python

import sys # package to read parameters

import MySQLdb # communication to database

#

# Script to automatically provision a database, user and user rights

# usage: ./createdb.py <databasename> <username> <password>

#

MYSQL\_HOST = 'localhost'

MYSQL\_USER = 'root'

MYSQL\_PASSWORD = 'QSDFjklm123'

if len(sys.argv) < 3:

print 'Please enter the database, user and password you want to create.'

sys.exit()

database = sys.argv[1]

user = sys.argv[2]

password = sys.argv[3]

response = raw\_input(

'Creating database {database} and user {user} with password {password} Is '

'this ok? [y/N]\n'.format(

database=database,

user=user,

password=password

)

)

if response.lower() not in ['y', 'yes']:

print 'Exiting...'

sys.exit()

print 'Connecting...'

connection = MySQLdb.connect(

host=MYSQL\_HOST,

user=MYSQL\_USER,

passwd=MYSQL\_PASSWORD

)

cursor = connection.cursor()

print 'Creating database...'

cursor.execute(

'CREATE DATABASE IF NOT EXISTS {database}'.format(

database=database

)

)

print 'Creating user...'

cursor.execute(

'CREATE USER "{user}"@"{host}"'

'IDENTIFIED BY "{password}"'.format(

user=user,

host=MYSQL\_HOST,

password=password

)

)

print 'Granting permissions...'

cursor.execute(

'GRANT ALL ON {database}.\* TO "{user}"@"{host}";'.format(

database=database,

user=user,

host=MYSQL\_HOST

)

)

print 'Flushing privileges...'

cursor.execute(

'FLUSH PRIVILEGES;'

)

print 'Done.'

Breid bovenstaand script uit met de volgende acties:

* Het script verwacht drie parameters: Database naam, user naam en wachtwoord. Je roept het bijvoorbeeld op met: ./createdb.py dbklantX klantX wachtwoordX. Je kan parameters gebruikers in python via de sys.argv array
* Connecteer naar de databank als root en je eigen gekozen wachtwoord
* Maak een SQL-gebruiker aan met de naam en het wachtwoord dat als parameter meegegeven werd
* Geef deze gebruiker alle rechten op de databank.

Vergeet niet om feedback te geven aan de gebruiker van het script (db aangemaakt, user gecreëerd, …)

### 1.8.2. PHP

Installeer nu de PHP package en maak in jouw familienaam.ikdoeict vhost een phpinfo.php bestand die de output toont van phpinfo(). Op basis van die output vul je onderstaande gegevens in:

|  |  |
| --- | --- |
| PHP Versie | 5.4.16 |
| Max grootte van 1 POST | 8M |
| Apache versie | Apache/2.4.6 (CentOS) PHP/5.4.16 |
| Locaties van de extensions | /usr/lib64/php/modules |
| Default timezone | UTC |

Deze informatie is handig voor jou als systeembeheerder, maar ook voor hackers die op zoek gaan naar kwetsbaarheden in jouw systeem. Als je jouw phpinfo pagina in een recente browser opent, rechtermuisklikt en ‘inspect’ (of iets gelijkaardigs) doet vind je daar een network tab. Deze informatie kan door iedereen gelezen worden.

Op die network tab kan je onder andere de HTTP response headers zien. Daarin krijg je de versienummers van Apache, PHP en het OS van jouw server te zien. Hoe minder een hacker te weten komt over jouw server hoe beter.

Met de default settings geef je dus veel informatie prijs over jouw server. Pas de configuratie van PHP en je webserver aan zodat je onderstaande security settings realiseert. Controleer telkens met de development tools (network tab) in je browser of in de output van je phpinfo-pagina wat het effect was. Noteer hieronder wat je gewijzigd hebt en wat het resultaat was:

|  |  |
| --- | --- |
| **Security Setting** | **Hoe ingesteld? (locatie + waarde noteren!)** |
| PHP versie verbergen | /etc/php.ini  expose\_php = Off |
| Apache versie verbergen | /etc/httpd/conf/httpd.conf  ServerTokens ProductOnly  ServerSignature Off |
| Schakel allow\_url\_fopen uit  [Meer info](http://phpsec.org/projects/phpsecinfo/tests/allow_url_fopen.html) | /etc/php.ini  allow\_url\_fopen = Off |

Noot: dit is zeker niet voldoende om een secure PHP webserver op te bouwen en dient enkel om je een idee te geven waar bepaalde settings staan.

### 1.8.3. Dynamic hosting (Klant 2)

Nu we onze setup vervolledigd hebben, meldt zich een tweede klant aan. Voer de volgende acties uit:

* de website moet bereikbaar zijn vanaf je client via het adres [www.klant2.ikdoeict](http://www.client1.ikdoeict)
* De aangeleverde code vind je op Google Drive ([klant2.zip](https://drive.google.com/open?id=0B0YTezDATQOtU1JBTVRIUGZhTFE)). Je plaatst die in een subdirectory onder /var/www/html/vhosts/klant2.ikdoeict
* Configureer de webserver en maak gebruik van een aparte virtualhost zoals we eerder deden.
* Gebruik je MySQL script om de volgende zaken te doen:
  + Maak een nieuwe gebruiker “klant2” met als wachtwoord “Azerty123!”
  + Maak een databank “todo” waar klant2 alle rechten op krijgt
* Importeer todo.sql uit de aangeleverde code. Gebruik hiervoor jouw klant2 gebruiker.
* Pas het bestand Includes > config.php aan zodat de juiste MySQL-gebruiker verbinding maakt naar de databank.
* Configureer DNS zodat je kan surfen van je client naar deze website. Je zal hiervoor opnieuw een nieuwe zone moeten configureren.
* Je zal mogelijks de configuratie van PHP moeten aanpassen om dit project werkende te krijgen.

**Controlepunt 4: Laat jouw labo tot nu controleren door je docent**

## 1.9. Secure hosting

Klant2 werkt met gevoelige informatie (inloggegevens van klanten) en wil dit graag encrypteren met SSL. We gaan hiervoor een self signed certificate gebruiken. In een echt hostingbedrijf gebruik je uiteraard beter een publiek erkend certificaat om geen foutmelding in je browser te krijgen.

Genereer een self signed certificaat met bijhorende private key. Plaats beide files in /etc/httpd/ssl/klant2.ikdoeict Zorg dat je certificaat de volgende eigenschappen heeft:

* 4096 bits key lengte
* Geldigheid 1 jaar
* Vul zoveel mogelijk gegevens in over jouw site.

Noteer hieronder je commando’s.

|  |
| --- |
|  |

Pas nu je vhost configuratie aan zodat je <https://www.klant2.ikdoeict> kan bereiken via je client.

OpenSSL heeft de laatste jaren nogal veel security problemen ondervonden. Hieronder vind je enkele namen van die security bugs. Noteer in onderstaande tabel (in eigen woorden) wat het probleem was en hoe je kan oplossen. Doe de nodige configuratie op jouw server (indien nodig).

|  |  |  |
| --- | --- | --- |
| **Naam bug** | **Probleem** | **Oplossing** |
| POODLE (2014) | Bij de handshake kon er een probleem zijn waardoor een een 64kb memory leak ontstond. Hierdoor was een DDOS aanval gemakkelijker mogelijk. | Updaten nar versie 1.0.1j. |
| HEARTBLEED (2012) | Een memory leak werd veroorzaakt en kon bekeken worden. Alles in het geheugen kan is dan bijna publiek. | Updaten naar 1.0.1g of OpenSSL opnieuw compilen met `-DOPENSSL\_NO\_HEARTBEAT`. |
| CRIME (2012) | Het maakt het mogelijk authenticatie cookies te stelen en dus een geauthenticeerde sessie te hijacken. | Eén van de eindpunten (server/client) moet TLS 1.2 en SPDY compressie uitschakelen. |

Recente opensll bugs vind je onder andere op: <https://www.openssl.org/news/vulnerabilities.html>

## 1.10. Backup

Nu we steeds meer klanten aantrekken, moeten we ervoor zorgen dat hun data op een veilige locatie gebackupped wordt. Wij gaan dit simuleren door de inhoud van de vhosts te kopiëren naar een andere map op dezelfde schijf.

Er bestaan heel wat bestaande oplossingen om files te backuppen op Linux. SCP en Rsync zijn twee veelgebruikte commando’s. Wat is het verschil tussen deze twee commando’s?

|  |
| --- |
|  |

Wij gaan gebruik maken van rsync om een .tar file te bekomen via een nieuw Pythonscript.

Maak een nieuw bestand aan /opt/hostingscripts/vhostbackup.py en zorg voor de volgende acties:

* Log alle output naar /var/log/hostingscripts/backup.log. Om de complexiteit van dit script te beperken mag je dit bestand eerst aanmaken via de command line.
* Kopieer de inhoud van de vhosts naar /tmp via rsync. Hiervoor maak je best een nieuwe map in je script aan.
* Maak een database dump en sla die ook op in /tmp
* Voeg de data en de database dump toe aan een gecomprimeerd bestand met als naam TIMESTAP\_backup.tar
* Kopiëer het tar bestand naar /var/backup (controleer of die al bestaat, zoniet maak die aan)
* Verwijder de inhoud van de /tmp map (enkel jouw bestanden!)

Zorg er tenslotte voor dat dit script elke dag loopt, om 5u ‘s nachts. Hiervoor maak je gebruik van cron. Noteer hieronder hoe je dit deed:

|  |
| --- |
|  |

**Controlepunt 5: Laat jouw labo tot nu controleren door je docent**