Bescherm jezelf tegen hackers en scriptkiddies

MARTIJN DE MEULDER

MAAK EEN FORT VAN JE WEBSERVER

ALS JE EEN WEBSERVER BEHEERT DIE ONDERDEEL IS VAN HET INTERNET WEET JE ÉÉN DING ZEKER: JE WORDT AANGEVALLEN. VIERENTWINTIG UUR PER DAG, ZEVEN DAGEN IN DE WEEK PROBEREN HACKERS EN SCRIPTKIDDIES OP DE MEEST CREATIEVE MANIEREN JE SERVER BINNEN TE DRINGEN. HOE BESCHERM JE JEZELF DAARTEGEN? IN DEZE MASTERCLASS LEGGEN WE STAP VOOR STAP UIT HOE JE EEN GROOT DEEL VAN DE BEDREIGINGEN HET HOOFD KUNT BIEDEN.

Een van de wonderen van het moderne internet is de beschikbaarheid van server-infrastructuur. Betaalde je jezelf vroeger blauw aan een gedeeld hostingaccountje met een hele batterij beperkingen, tegenwoordig kun je bij de grotere hostingproviders al voor 15 euro per maand een virtual private server (VPS) huren. Met vier processorcores en acht gigabyte geheugen bieden deze genoeg kracht om een enorme hoeveelheid webservers en databaseapplicaties op te draaien. Als je weet hoe je ermee om kunt gaan, zijn de mogelijkheden eindeloos. Maar zoals zijn oom Ben al in de eerste Spiderman-film aan Peter Parker vertelt: “Met grote kracht komt ook grote verantwoordelijkheid.” Dat geldt ook voor jou als je zo’n server huurt. Want zo’n krachtige server kun je ook geweldig inzetten voor doelen die jij vermoedelijk helemaal niet voor ogen hebt, zoals voor het versturen van miljarden spamberichten, het hosten van dubieuze content of het aanvallen van andere servers op het internet. Criminelen, hackers en scriptkiddies zijn dag en nacht op zoek naar slecht beveiligde machines. Ook bij jouw VPS komen ze aankloppen, kijk je toegangslogs er maar eens op na. Je krijgt er koude rillingen van als je ziet hoe vaak allerlei volk uit de hele wereld langskomt. Meestal zijn het portscans of andere eenvoudige methodes om een indruk te krijgen van hoe je server eruitziet, vergelijkbaar met het rammelen aan de deur om te voelen of het slot het er goed op zit. Maar ze zijn er wel, en als jij even verslapt zullen ze binnendringen en toeslaan.

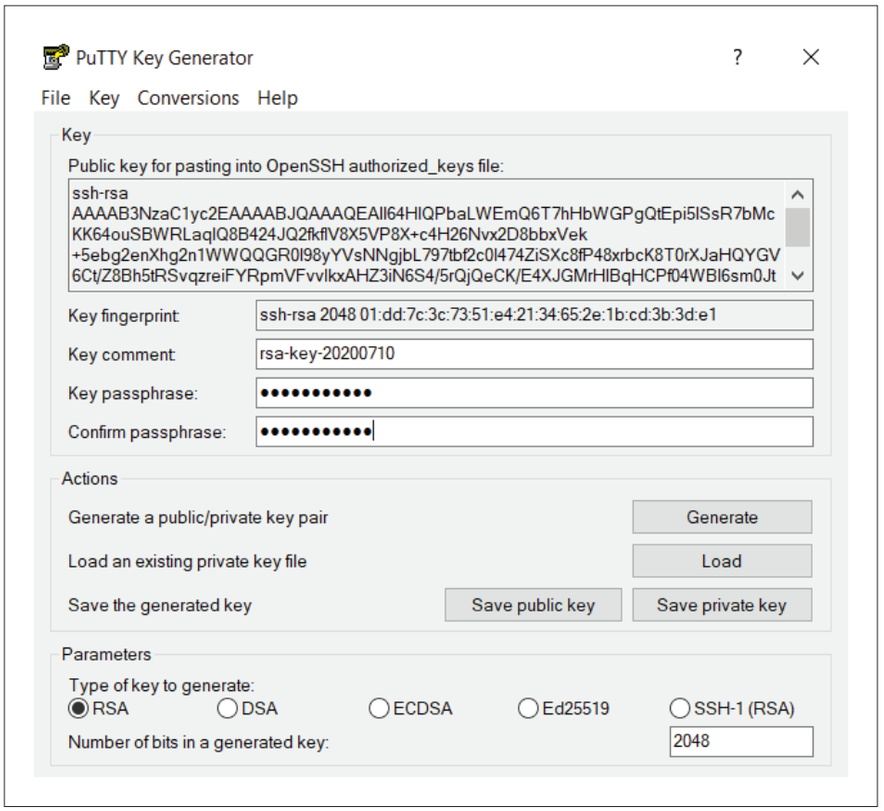
ZERO-DAY-EXPLOITS

Om dat te voorkomen, is het zinvol je te verdiepen in het beveiligen van je systeem. We benadrukken dat een waterdichte methode niet bestaat. Altijd kan er ergens diep in je software een zero-day-exploit zitten (een nog ongepubliceerde kwetsbaarheid) die alleen door de makers van de software kan worden gepatcht. Maar met de maatregelen in deze masterclass maak je van je server in ieder geval een moeilijk te nemen vesting, om zo de meeste aanvallers buiten de deur houden. In deze masterclass richten we ons op het beveiligen van een Linux-webserver met een zogenoemde LAMP-configuratie: met Linux, Apache, MySQL en PHP – de meestgebruikte webservercombinatie ter wereld. Wij gebruiken CentOS als besturingssysteem, maar de kennis kun je ook toepassen op bijvoorbeeld Ubuntu of een van de andere Linux-varianten. Het belangrijkste verschil voor de interpretatie van deze masterclass is vooral de gebruikte packagemanager, het systeem voor softwarepakketbeheer. Op CentOS is dat **yum**, terwijl Ubuntu (gebaseerd op Debian) **apt-get** gebruikt. Ook kan een enkel configuratiebestand op een andere plek staan, maar als je zoekt op internet op

Succes! Yum-cron draait en zorgt ervoor dat je server automatisch up-to-date blijft.

```
[pcm_tst@pcmweb.nl ~] $ sudo systemctl status yum-cron.service
● yum-cron.service - Run automatic yum updates as a cron job
   Loaded: loaded (/usr/lib/systemd/system/yum-cron.service; enabled; vendor preset: disabled)
   Active: active (exited) since vr 2020-07-10 10:18:44 CEST; 4s ago
  Process: 20073 ExecStart=/bin/touch /var/lock/subsys/yum-cron (code=exited, status=0/SUCCESS)
 Main PID: 20073 (code=exited, status=0/SUCCESS)
```

de naam van het bestand in combinatie met je besturingssysteem zul je snel de locatie weten. De configuratiebestanden zelf zullen in alle gevallen precies hetzelfde zijn.
Ook gaan we ervan uit dat je de webserver met bovenstaande software al draaiende hebt.
We gebruiken de editor nano in de voorbeelden, heb je deze nog niet geïnstalleerd, dan kun je dat doen met het commando `yum install nano`. Of als je Ubuntu/Debian gebruikt: `sudo apt-get install nano`.
Wanneer je fan van vi of vim bent, kun je natuurlijk ook deze editors gebruiken.



Met PuTTYgen kun je het sleutelpaar genereren waarmee je ssh beveiligt.

HOUD JE SERVER ACTUEEL

Zorg ervoor dat de software op je server altijd up-to-date is om oude lekken geen kans te geven. Gebruik dit commando om te zien of er updates zijn:

```
sudo yum check-update
```

Zit er wat voor je bij? Tik dan `yum update` in, waarna de pakketten worden bijgewerkt.
Deze manier van updaten is in de praktijk natuurlijk vrij lastig omdat je zelf alles moet doen. Je kunt het pakketbeheer ook automatiseren met yum-cron. Om deze service te installeren en activeren, gebruik je het volgende commando:

```
sudo yum install yum-cron
```

Kijk het configuratiebestand na en pas deze eventueel aan je wensen aan met:

```
sudo nano /etc/yum/yum-cron.conf
```

Bij alle opties zul je voldoende uitleg vinden om er de juiste beslissingen bij te maken. Let wel op de optie **apply_updates**, deze staat standaard op **no**. Maar om automatische updates uit te laten voeren – en dat wil je – moet je deze veranderen in **yes**.
Start nu de service en kijk de status na met deze commando's:

```
sudo systemctl enable yum-cron.service
sudo systemctl start yum-cron.service
sudo systemctl status yum-cron.service
```

Om nu deze service altijd te laten starten, gebruik je de volgende twee commando's:

```
sudo chkconfig yum-cron on
sudo service yum-cron start
```

UBUNTU EN DEBIAN

Ook op Ubuntu is er een dergelijke service, 'unattended upgrades' genoemd. Deze kun je op een vergelijkbare manier configureren en installeren. Installeer de service met de opdracht:

```
apt-get install unattended-upgrades
```

En bewerk daarna het configuratiebestand met het commando:

```
sudo nano /etc/apt/apt.conf.d/50unattended-upgrades
```

Haal hier de beide slashes weg voor de regel met het woord **updates** en open het volgende bestand met de opdracht:

```
sudo nano /etc/apt/apt.conf.d/20auto-upgrades
```

Plaats daar de volgende regels in om iedere twee dagen te controleren op updates:

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::AutocleanInterval "2";
APT::Periodic::Unattended-Upgrade "1";
```

Test ten slotte of het werkt:

```
sudo unattended-upgrades --dry-run -debug
```

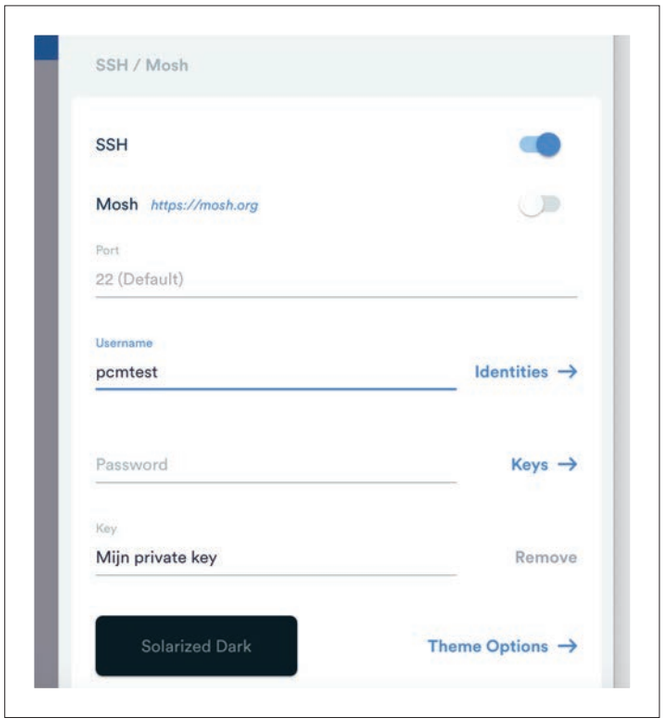
Alles in orde? Gefeliciteerd, je server houdt zichzelf nu bij de tijd.

BEVEILIG SSH

Ssh (Secure shell) is een onmisbaar gereedschap voor het onderhoud van je

ZO'N KRACHTIGE SERVER IS OOK GESCHIKT VOOR **DOELEN DIE JIJ HELEMAAL NIET VOOR OGEN HEBT**

server, het zorgt ervoor dat je door een versleuteld kanaal verbinding maakt met je machine en de opdrachtregel kunt gebruiken. Ssh werkt in een standaardinstallatie echter met een gebruikersnaam en een wachtwoord, en dat is een gapend gat in de beveiliging. Wachtwoorden kunnen worden geraden met zogenoemde brute-force-aanvallen. Het beste kun je de toegang tot je server met inlognaam en wachtwoord helemaal verwijderen. Ssh biedt daarvoor ondersteuning, je kunt met een sleutelpaar toegang krijgen: de publieke sleutel staat op je server, de private sleutel staat op de machine waarmee je toegang wilt hebben tot de server. Heeft iemand deze sleutel niet (iedereen buiten jezelf), dan wordt deze meteen geweigerd door de server. Installeren doe je als volgt. Als eerste moet je de sleutels genereren. Gebruik je een Windows-pc, download daarvoor dan het programma PuTTYgen vanaf **www.puttygen.com**. Start PuTTYgen na installatie en druk op de knop **Generate**. Na het genereren zie je de publieke sleutel in het venster verschijnen, kopieer deze naar het klembord. Geef in dit venster ook het wachtwoord op, en bewaar de publieke en private sleutel in een bestand door op de daarvoor aangegeven knoppen te drukken.



De configuratie van Termius: geef hier de private sleutel op die je zojuist hebt gegenereerd.

LAATSTE KEER MET WACHTWOORD

Nu je dat hebt gedaan, kun je inloggen op je server (met ssh natuurlijk, maar voor de laatste keer nog met inlognaam en wachtwoord). Kopieer nu de publieke sleutel uit het PuTTYgen-venster en open het bestand **~/.ssh/authorized_keys** met het commando:

```
sudo nano ~/.ssh/authorized_keys
```

Plak daarin de publieke sleutel die je zojuist hebt gekopieerd. Bestaat de directory **.ssh** in je home-directory nog niet – dat kan goed als je een nieuwe server configureert – maak deze dan eerst aan met `mkdir .ssh`. Zet daarna de permissies in een veilige stand met:

```
sudo chmod 700 ~/.ssh
sudo chmod 600 ~/.ssh/authorized_keys
restorecon -Rv ~/.ssh
```

Als je met een Linux-client werkt of met macOS, dan heb je PuTTYgen en bovenstaande instructie niet nodig, je kunt je sleutelpaar genereren én meteen naar **~/.ssh** kopiëren met de volgende serie commando's (aangenomen dat je Homebrew hebt geïnstalleerd op je systeem):

```
sudo brew install ssh-copy-id
ssh-keygen -t rsa
ssh-copy-id -i [hier het pad naar de zojuist aangemaakte sleutel] inlognaam@serveradres
```

Vervang de aanduiding **[hier het pad naar de zojuist aangemaakte sleutel]** door het volledige pad inclusief bestandsnaam.

TESTEN

Je kunt nu de installatie testen door je ssh-client opnieuw te configureren. Hoe je dat doet, hangt natuurlijk af van de software die je gebruikt. Wij gebruiken zelf Termius (**www.termius.com**). Druk daar op de configuratie van je server en scrol naar onderen. Verwijder het wachtwoord, maar laat je gebruikersnaam staan. Druk op **Key** en in het venster dat volgt, op de groene knop **+ key**. In het venster dat je ziet, kun je de inhoud van het bestand met de private sleutel kopiëren dat je zojuist met PuTTYgen hebt opgeslagen. Geef daar ook het wachtwoord op dat je hebt ingegeven in PuTTYgen en sla de gegevens op. Nu kun je verbinding maken met de server. Lukt dat? Gefeliciteerd, je kunt nu overgaan tot de laatste stap: het verbieden van toegang met inlognaam en wachtwoord.

ALLEEN NOG MET SLEUTELS

Het verbieden van het gebruik van wachtwoorden doe je door op je server het sshd-configuratiebestand te bewerken met het commando:

```
sudo nano /etc/ssh/sshd_config
```

Zoek naar de regel **PasswordAuthentication** en verander **yes** in **no**. Sluit nano af en herstart de ssh-service met:

```
sudo systemctl restart sshd
```

Test wel even vanaf je lokale

SSH WERKT STANDAARD MET EEN INLOGNAAM EN EEN WACHTWOORD: EEN GAPEND BEVEILIGINGSGAT

```
Status for the jail: sshd
|- Filter
|  |- Currently failed: 4
|  |- Total failed:     1877961
|  `- Journal matches:  _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`- Actions
   |- Currently banned: 1
   |- Total banned:     13594
   `- Banned IP list:   209.141.37.175
```

De inhoud van onze ssh-jail op een server die al een paar jaar draait.

```
/var/log/fail2ban.log-20200705:2020-07-03 00:46:12,944 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 207.180.241.187
/var/log/fail2ban.log-20200705:2020-07-03 02:31:06,153 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 71.90.79.184
/var/log/fail2ban.log-20200705:2020-07-03 04:42:03,869 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 24.85.206.238
/var/log/fail2ban.log-20200705:2020-07-03 04:44:55,296 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 135.0.24.9
/var/log/fail2ban.log-20200705:2020-07-03 04:50:29,938 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 71.81.53.233
/var/log/fail2ban.log-20200705:2020-07-03 05:01:37,951 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 98.197.24.128
/var/log/fail2ban.log-20200705:2020-07-03 05:22:42,671 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 96.67.174.97
/var/log/fail2ban.log-20200705:2020-07-03 05:43:52,635 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 151.61.155.20
/var/log/fail2ban.log-20200705:2020-07-03 09:06:05,968 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 72.134.168.109
/var/log/fail2ban.log-20200705:2020-07-03 13:58:24,977 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 45.148.10.221
/var/log/fail2ban.log-20200705:2020-07-03 15:06:14,382 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 104.158.178.37
/var/log/fail2ban.log-20200705:2020-07-03 15:13:05,087 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 221.14.189.127
/var/log/fail2ban.log-20200705:2020-07-03 17:09:51,055 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 98.165.232.86
/var/log/fail2ban.log-20200705:2020-07-03 19:39:55,308 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 93.228.136.106
/var/log/fail2ban.log-20200705:2020-07-03 19:59:50,927 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 96.73.97.219
/var/log/fail2ban.log-20200705:2020-07-03 21:13:48,994 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 45.148.10.222
/var/log/fail2ban.log-20200705:2020-07-04 04:42:52,228 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 116.98.171.215
/var/log/fail2ban.log-20200705:2020-07-04 09:27:51,676 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 75.139.184.5
/var/log/fail2ban.log-20200705:2020-07-05 01:01:51,222 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 45.148.10.222
/var/log/fail2ban.log-20200705:2020-07-05 02:22:44,464 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 98.127.187.121
/var/log/fail2ban.log-20200705:2020-07-05 02:44:53,099 fail2ban.actions        [14252]: NOTICE  [sshd] Ban 205.185.116.157
```

Een paar dagen Fail2ban levert deze indrukwekkende lijst met dubieus volk op.

computer of je echt niet meer kunt aanmelden met inlognaam en wachtwoord. Als dat inderdaad mislukt, heb je een enorme stap voorwaarts gedaan in het beveiligen van je server.

ACTIEVE BEWAKING MET FAIL2BAN

Je hoeft niet ál het beveiligingswerk zelf te doen, je kunt met Fail2ban ook een automatische bewakingsrobot op je server installeren. Fail2ban scant de logbestanden van de serversoftware die je hebt draaien, zoals ssh, Apache en ftp, op zoek naar kwaadaardige activiteit, zoals inlogpogingen en zoektochten naar bekende zwakheden. Ontdekt het een poging tot inbraak, dan past Fail2ban zelf de regels van je firewall aan om de ip-adressen die verantwoordelijk zijn voor het ongewenste gedrag alle toegang tot je systeem te ontzeggen. In Fail2ban-terminologie: het ip-adres wordt in de 'jail' geplaatst. Fail2ban zit niet in de standaard CentOS-distributie, maar in de Extra Packages for Enterprise Linux (EPEL) van Red Hat. Installeer de package met de commando's:

```
sudo yum install epel-release
sudo yum install fail2ban
```

Voor Ubuntu hoef je EPEL niet te installeren, dit commando is genoeg:

```
sudo apt-get install fail2ban
```

Kopieer dan het standaard configuratiebestand en open het in nano:

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
sudo nano /etc/fail2ban/jail.local
```

JAILS

Lees in dit bestand rustig de opties door die op jouw machine van toepassing zijn, vooral ook de instellingen onder het kopje **JAILS**, waar de gevangenissen voor aanvallers van de verschillende diensten op je server worden ingesteld. De meeste opties hoef je niet te wijzigen, let echter op de optie **enabled = true** in de ssh-jail-sectie, die staat nog niet aan. Haal het hekje (#) voor deze regel weg. Ben je tevreden met de instellingen? Sluit dan de editor af, voeg Fail2ban toe aan de standaard op te starten diensten van je server en start de service met:

```
sudo systemctl enable fail2ban
sudo systemctl start fail2ban
```

Wil je weten of Fail2ban draait? Dan kun je de status ervan controleren met:

```
sudo fail2ban-client status
```

Zien welke inbrekers een mislukte poging op je ssh-dienst hebben gedaan? Gebruik dan de opdracht:

```
sudo fail2ban-client status sshd
```

Of weten welke ip-adressen zijn verbannen? Al is deze opdracht meestal pas na een paar uur of een paar dagen zinvol:

```
sudo zgrep 'Ban' /var/log/fail2ban.log*
```

Of wil je een live log zien van malicieuze pogingen?

```
sudo tail -F /var/log/fail2ban.log
```

FTP-TOEGANG BEVEILIGEN

Als je een webserver draait, is de kans dat je ook een ftp-server hebt natuurlijk bijzonder groot. Ftp is een geweldig protocol dat ook heel makkelijk werkt. Het grote nadeel ervan is dat bestanden, wachtwoorden en inlognamen bij een standaardinstallatie onversleuteld worden verstuurd. Dat is niet verstandig, het maakt je kwetsbaar voor allerlei 'man-in-the-middle'-aanvallen. Niet alleen op internet, maar bijvoorbeeld ook bij het

Security through obscurity

Het belangrijkste wat je moet doen om hackers buiten de deur van je server te houden, is zorgen dat je software up-to-date is en dat je configuratie op orde is. Maar er is nog een stroming binnen het beveiligingsdenken die het verdient vermeld te worden: 'security through obscurity', zoals dit in de digitale wandelgangen wordt genoemd. Of zoals de Nederlandse overheid in zijn anti-inbraak-campagne roept: "Maak het ze niet te makkelijk." Zorg dat je een goed slot op je deur hebt, maar het helpt al helemaal als je ervoor zorgt dat die deur niet te vinden is. Veel software gebruikt immers standaardmethoden voor zijn installatie en locatie, wijk daar dus vanaf. Een mooi voorbeeld is de database-administratiesoftware phpMyAdmin. Dat is een heel gemakkelijke webinterface voor MySQL die zichzelf standaard installeert op **www.domein.nl/phpmyadmin**. Dat is natuurlijk heerlijk voor hackers. Ze weten daardoor dat je deze software hebt draaien (en kunnen uitzoeken of er misschien een exploit voor is) én dat je server op MySQL draait. Ze kunnen ook proberen het wachtwoord te kraken. Het is in de configuratie van phpMyAdmin makkelijk om die locatie veranderen. Het is namelijk geen directory, maar een alias (zie de paragraaf 'MySQL en php'). Dus verander dat echt! Weinig hackers zullen de locatie **www.domein.nl/pcm_masterclass_toegang** kunnen raden. Dat is een kleine maar belangrijke stap op weg naar meer veiligheid. Het is verstandig dat voor alle software met standaardlocaties te doen, denk bijvoorbeeld ook aan WordPress: **www.domein.nl/wp-login** is een klassieker in de poortscans. En let ook op alles onder de **/vendor**-directory die door Composer wordt geïnstalleerd.

gebruiken van een openbare hotspot (zonder vpn). Daarom gaan we de bestandsoverdracht beveiligen met een sleutel. We richten ons Vsftpd, op dit moment de meest gebruikte Linux-ftp-server, en we gaan ervan uit dat je deze inmiddels draaiende hebt.

Het idee is dat je een sleutel en een certificaat genereert, en deze op je server bewaart. Maak daartoe eerst een directory aan:

```
sudo mkdir /etc/ssl/private
```

Genereer je certificaat-sleutelpaar met het volgende commando:

```
openssl req -x509 -nodes -days 3650
-newkey rsa:1024 -keyout /etc/ssl/
private/ftp_certificaat.pem -out
/etc/ssl/private/ftp_certificaat.pem
```

Beantwoord de vragen die aan je worden gesteld. Het belangrijkst is die van de hostnaam: deze moet overeenkomen met de hostnaam van je server.

CONFIGURATIEBESTAND INSTELLEN

Stel nu het ftp-serverconfiguratiebestand in met:

```
sudo nano /etc/vsftpd/vsftpd.conf
```

En voeg de volgende regels toe aan het einde van het bestand:

```
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
rsa_cert_file=/etc/ssl/private/ftp_
certificaat.pem
rsa_private_key_file=/etc/ssl/
private/ftp_certificaat.pem
```

Herstart je server:

```
systemctl restart vsftpd
```

MET FAIL2BAN KUN JE EEN AUTOMATISCHE **BEWAKINGSROBOT OP JE SERVER INSTALLEREN**

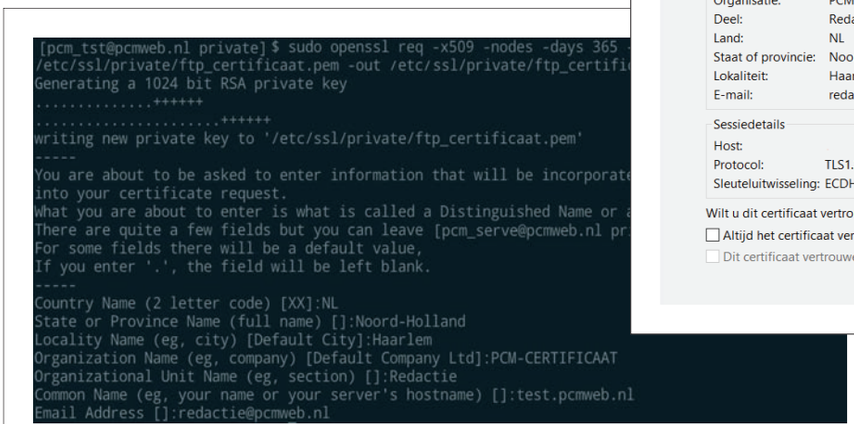
Het is nu niet meer mogelijk om zonder versleutelde verbinding contact te maken met je ftp-server, precies zoals we dat wilden. Stel sftp in als de standaard verbindingsmethode van je client en maak verbinding. Je ftp-client – wij gebruiken FileZilla – zal een overzicht van de door jou gebruikte serversleutel laten zien. Accepteer deze, daarna zal de client verbinding maken met je server.

VERSTERK JE MYSQL-DATABASE

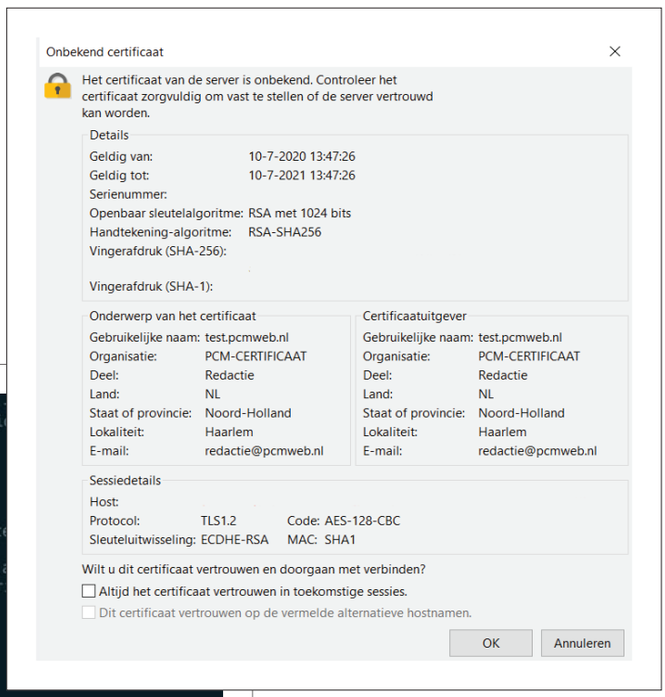
Geen moderne website kan zonder goede database en MySQL is daarvoor doorgaans de standaardkeuze. Met het dichttimmeren van de MySQL-installatie kom je al een heel eind door het mysql-hardening-script op je systeem te draaien dat je met de installatie van de software krijgt meegeleverd. Daarmee kun je in één beweging wachtwoorden beheren, anoniem gebruik tegengaan, externe verbindingen naar je database onmogelijk maken en verwijder je testdata:

```
sudo mysql_secure_
installation
```

Volg de instructies zorgvuldig en je bent een behoorlijk eind op weg. Verder geldt dat het goed is om je gebruikers te beperken in hun bewegingsvrijheid: geef ze alleen toegang tot de hoogstnoodzakelijke databases. En mocht je toch netwerkverbindingen willen toestaan, versleutel ze. Hoe je dat instelt, gaat voorbij de essentie van dit verhaal, maar je kunt in het diepe duiken aan de hand van dit artikel op de website Medium.com: **www.tiny.cc/5mysql**.



Bij het genereren van het certificaat word je gevraagd wat gegevens in te vullen.



Gelukt! We krijgen het certificaatoverzicht in Filezilla te zien na de eerste keer dat we contact maken.

MYSQL EN PHP

Besteed wat extra aandacht aan de manier waarop je MySQL via scripts als php gebruikt. De klassieke methode is die met het gebruik van de mysqli-driver. Deze is inherent onveilig als je parameters uit de url accepteert en in database-query's opneemt. Je kunt dan het slachtoffer worden van MySQL-injection-aanvallen. Een veel modernere en stabielere methode vind je in de vorm van een database-abstractielaag als PDO (PHP Data Objects), die je helpt je query's op relatief eenvoudige wijze te beveiligen. Start met lezen op **www.phpdelusions.net/pdo**.

Ook is er PHPMyAdmin, deze php-front-end voor MySQL is niet voor niets enorm populair en maakt het beheer van je databases veel eenvoudiger. Zorg dat de toegangspoort ertoe niet makkelijk te vinden is (zie het kader 'Security through obscurity') door het configuratiebestand te openen:

```
sudo nano /etc/httpd/conf.d/phpMyAdmin.conf
```

Onder Ubuntu vind je dit bestand op **/etc/apache2/conf.d/phpmyadmin.conf**. Verander daar deze regel in iets creatievers:

```
Alias /phpMyAdmin /usr/share/phpMyAdmin
```

Bijvoorbeeld:

```
Alias /pcm_masterclass_toegang /usr/share/phpMyAdmin
```

Daarna open je PHPMyAdmin door naar **https://servernaam.nl/pcm_masterclass_toegang** te gaan.

En nu je toch in dit bestand bezig bent, kun je meteen de toegang tot PHPMyAdmin beperken tot alleen het (vaste) ip-adres van je client-pc, door direct onder de start van **<Directory /usr/share/phpMyAdmin/>** het volgende toe te voegen:

```
Order Deny,Allow
Deny from All
Allow from 1.2.3.4
```

Vervang **1.2.3.4** met jouw eigen ip-adres. Zo sluit je heel de wereld buiten, behalve jezelf. Sluit de editor af en herstart apache met:

```
sudo systemctl restart apache
```

En controleer of het werkt.

Beveilig je Apache webserver

Apache is van zichzelf al een behoorlijk veilige webserver. Maar het is goed om zelf nog het een en ander af te stellen. Daarom drie snelle tips voor een betere Apache:

STAP 1: VERSLEUTEL AL JE VERKEER

Dat doe je met een 'Let's Encrypt'-certificaat en de volgende commando's. We gaan ervan uit dat je al een domeinnaam aan de server hebt gekoppeld en deze als virtual host hebt geconfigureerd:

```
sudo yum install epel-release
```
(mocht je deze nog niet hebben geïnstalleerd)

```
sudo yum install certbot python2-certbot-apache mod_ssl
sudo certbot install
sudo certbot --apache
```

Volg de instructies (vergeet niet te selecteren dat al het gewone verkeer wordt doorgestuurd naar poort 443) en herstart je webserver weer met:

```
sudo systemctl restart apache
```

STAP 2: VERBERG JE SOFTWAREVERSIES

Je server verraadt meer over zichzelf dan nodig is, geef de url van server maar eens op bij **http://security.firewallmonitor.org**. Dan zie je dat zaken als de Apache-versie, je besturingssysteem en php-versie zichtbaar zijn in zowel de footer van lege pagina's als in de http-headers. Daar heeft helemaal niemand iets mee te maken. Je kunt ze gelukkig makkelijk verbergen, open het Apache-configuratiebestand met:

```
sudo nano /etc/httpd/conf/httpd.conf
```

Onder Ubuntu vind je dit bestand op **/etc/apache2/conf/httpd.conf** of **apache2.conf**. Onder aan dit bestand voeg je de volgende regels toe:

```
ServerSignature Off
ServerTokens Prod
```

Scherm ook de php-versie af door **php.ini** te bewerken:

```
sudo nano /etc/php.ini
```

Wijzig de regel `expose_php` in `expose_php = Off`.

Als je toch bezig bent, voorkom ook dat php meer informatie over fouten laat zien met:

```
display_errors = 0
display_startup_errors = 0
```

En maak uploads onmogelijk als dat kan, zo kunnen hackers in geen geval geen dubieuze scripts naar je server uploaden via het web:

```
file_uploads = 0
```

Sla **php.ini** op en herstart apache met:

```
sudo systemctl restart apache
```

STAP 3: VOORKOM BROWSEN DIRECTORY'S

Open het configuratiebestand van Apache met:

```
sudo nano /etc/httpd/conf/httpd.conf
```

Zoek naar de **content directive** en wijzig deze in:

```
<Directory /var/www/html/>
  Options -Indexes
  AllowOverride None
  Require all granted
</Directory>
```

Als een directory niet voorzien is van een **index.php** of **.html**, dan zal je server niet meer laten zien welke bestanden erin staan. En ook nu geldt: herstart natuurlijk de webserver met:

```
sudo systemctl restart apache
```

TOT SLOT

Je server is nu veel veiliger geworden. De meeste kwaadwillenden houd je nu buiten de deur. Maar vergeet niet dat veiligheid ook een werkhouding is, niet louter het instellen van software en hardware. Zorg dat je op de hoogte blijft van de ontwikkelingen op dit gebied, handel zelf op een veilige manier en maak je back-ups. Veilig ben je nooit honderd procent, maar als je het goed doet, dan kun je heel veel ellende voor zijn. En dat is waar het allemaal om is begonnen. «

Minder is beter

"De meeste zaken die we bezitten zijn niet nuttig. Sommige zijn overbodig, terwijl andere niet zoveel waard zijn. Maar we zien dit niet en zien ze als gratis, terwijl ze ons veel kosten." Tweeduizend jaar geleden gaf Seneca, Romeins schrijver en filosoof, al een van de grondregels van zinnig serverbeheer aan de mensheid. Hij bedoelde het natuurlijk voor het leven als geheel, maar zijn wijsheid geldt ook in de digitale achtertuin van je webserver: hoe minder software op je webserver aanwezig is, hoe beter. Want daardoor zijn er inherent minder kwetsbaarheden aanwezig en minder oude software waarvan je bent vergeten dat je het ooit hebt geïnstalleerd. Leer jezelf dus de principes aan van het minimalistisch pakketbeheer met yum of apt-get: hoe kun je na het experimenteren met software de boel weer opruimen? Of beter nog: doe je experimenten niet op een productiemachine, maar op een losse server. Maar wat je ook doet, doe zoals Seneca. Of wat je moeder vroeger vermoedelijk al riep: "Ruim je rotzooi op!"