workaround.org

literacy and subtleties for open-source bigots and other weirdos

[2015-10-06]

Creating a TLS encryption key and certificate

(If you are unfamiliar with the abbreviation "TLS": it is the successor to SSL.)

The internet is the best invention since sliced bread but it has become an evil place more than ever. We are dealing with script kiddies, out-of-control secret services and organized crime on a large scale. It is completely out of the question to run any unencrypted services over the internet. I consider any password immediately wasted that went through the internet

ISPmail guide for Debian Jessie

What's new in the Jessie guide?

Migrating from a Wheezy to a Jessie mail server

The big picture

Types of email domains

Install Debian Jessie on your server

Install the software packages

unencryptedly. So let us create an encryption key and a certificate for Postfix (SMTP), Dovecot (IMAP/POP3) and Roundcube (Webmail/HTTPS).

Caveat: do not use different certificates for Postfix and Dovecot. At least the MacOS mail client will refuse your connection with confusing error messages.

There are two factors that make an encryption certificate secure:

(a) – Cryptographical security

The **mathematical** idea of one-way functions that make it easy to decrypt data if you know the key – but to make it virtually impossible trying to break the decryption without that knowledge. For a couple of years the SHA1 algorithm was famous but thanks to Edward Snowden's disclosures we assume that it can be broken too easily. So everyone is advised to use at least SHA256 (which is part of the SHA2 algorithm). Also RSA signatures should not be 1024 bits long any more but rather 4096 bits. Creating secure keys and certificates is relatively easy – we have tools for that.

(b) – Trust

The idea of **trusting** a certificate. What's the point if the certificate is mathematically good but you don't know who you are communicating with. You may communicate very securely with the wrong counterpart

Creating a TLS encryption key and certificate

Setting up the Apache web server

Preparing the database

Making Postfix get its information from the MySQL database

Setting up Dovecot

Making Postfix send emails to Dovecot

Setting up Roundcube webmail

Testing email delivery

Relaying with SMTP authentication

Filtering out spam with SpamAssassin

DNS MX records

Managing users, aliases and domains

Success stories

Deploying a mail server with Ansible

Troubleshooting your mail server

(called man-in-the-middle). That’s not what you want. (See also my article on creating your own certificate authority.)

So you have to decide whether to buy a certificate or to create one yourself. In comparison:

Type	Advantages	Disadvantages	Application
Self-signed	No costs. Quickly created.	Users will get a warning message about an untrusted certificate. They can accept the certificate manually but you should tell them the certificate’s fingerprint so they can verify it.	Private mail server for yourself and friends who don’t mind ignoring the warning message about an unverified identity.

Readers’ comments

Misha 2018-10-16 at 09:59 on Relaying with SMTP authentication
Hi Christoph, I've been using Postfix/Dovecot setup for years now thanks also to your tutorial. I found out by experience

Jblack 2018-10-14 at 03:28 on Creating a TLS encryption key and certificate
Please provide clarification on which file I edit for the automated Let's Encrypt to work. You advised, "In this case

fleg 2018-10-11 at 06:43 on Managing users, aliases and domains
Very good article, but I`d like suggest to make version with Postfixadmin sql scheme because Postfixadmin has many options howto

Type	Advantages	Disadvantages	Application
Self-signed with own PKI	No costs. You can create further certificates that your users will consider trusted if they have your root certificate installed. The *easy-rsa* package is a good start.	Takes 15 minutes longer for you. Requires that all users install your root certificate as trusted on their computers. This may make sense only if you can automatically distribute the certificate in a corporate environment.	Private mail server for yourself and friends who don’t mind installing your root certificate once. Or a corporate mail server where you can automatically deploy your root certificate on all client workstations.

Christoph Haas 2018-10-10 at 12:06 on Relaying with SMTP authentication
I'm a bit confused. No, mail server are not supposed to change the sender address. But as long as the

Lars 2018-10-10 at 11:14 on Updating the BIOS on Lenovo laptops from Linux using a USB flash stick
I've tried exactly this process, and while it does result in a bootable image, trying to start the update process

Mark Petersen 2018-10-09 at 14:00 on Relaying with SMTP authentication
Christoph, Thanks for the reply. - I'm not sure what your mean "Did I use authenticated SMTP?" - My mail

Type	Advantages	Disadvantages	Application
No-cost certificate from LetsEncrypt	No costs. Users will likely not get a warning because most software trusts LetsEncrypt.	Certificates are only valid for 90 days. Depending on your network it may be a hassle and cause a downtime to renew a certificate.	Public mail server that does not need sugar coating called “extended validation”.
Paid certificate	Users will not get a warning.	Expensive (50€-200€ per year). Probably a lengthy registration process. You support the certificate mafia. You don’t improve the communication security in any way.	For anyone who has too much money.

Note: Whichever method you use – always create the key on your own server. Never trust a key that has been created by anyone else. It appears

Christoph Haas 2018-10-09 at 08:20 on Relaying with SMTP authentication
I wonder why the log reads "Sender address rejected: not logged in". Did you use authenticated SMTP? smtpd_sender_login is pretty

Mark Petersen 2018-10-08 at 21:48 on Relaying with SMTP authentication
Christoph, Thanks for your posts on how to configure and run a mail server. - Thanks to you, I've been

Lee Bosch 2018-10-05 at 21:45 on Success stories
You might want to remove the delete option when the current folder is Trash. I tried write protecting the Trash

Christoph Haas 2018-10-04 at 08:27 on Preparing the database simpler because you can omit one step. But the other party now knows your secret key and could in theory intercept your encrypted traffic. I will describe how to do that properly in any of the following sections. Just choose your option and follow the instructions.

Option 1: Self-signed

The simplest option. Just run this command on your server and you have a valid all-purpose certificate that is valid for the next ten years:

```
openssl req -newkey rsa:4096 -nodes -sha512 -x509 -days 3650 -nodes -out /etc/ssl/certs/mailserver.pem -keyout /etc/ssl/private/mailserver.pem
```

You will be asked for several pieces of information. Enter whatever you like. The only important field is the "*Common Name*" that must contain the fully-qualified host name that you want your server to be known on the internet. Fully-qualified means host + domain.

Make sure that the secret key is only accessible by the 'root' user:

```
chmod go= /etc/ssl/private/mailserver.pem
```

Option 2: Self-signed with own PKI

You seem to have done something unsual. There is no PHP7.2 in Debian's stable distribution. If you start upgrading packages

This option is a bit better than using a simple self-signed certificate. But your users or customers need to install your root certificate manually to establish a trust relationship to your PKI in both their browsers and their email programs.

I suggest you install the “easy-rsa” package and read the documentation:

```
zless /usr/share/doc/easy-rsa/README-2.0.gz
```

If there is enough interest in this topic I will elaborate on managing your own certificate authority using easy-rsa.

Option 3: No-cost certificate from StartSSL

This is almost always the best option. It doesn’t cost you money and will give you a basic trustworthy certificate that most browsers and email clients will accept.

Create a 4096 bit key file:

```
openssl genrsa -out /etc/ssl/private/mailserver.pem 4096
```

Create a certificate signing request (CSR) file from that key:

```
openssl req -new -key /etc/ssl/private/mailserver.pem -out /etc/ssl/certs/mailserver.csr
```

You will be asked for several pieces of information. Enter whatever you like. The only important field is the "*Common Name*" that must contain the fully-qualified host name that you want your server to be known on the internet.

Now go to StartSSL and sign up for an account. First use the "*Validation Wizard*" to validate your email address first. Then use the validation wizard again to validate the domain you want to use for your email server.

Next use the "*Certificates Wizard*" to create a "Web Server SSL/TLS certificate". Choose your email domain, paste the contents of the CSR file (/etc/ssl/certs/mailserver.csr) you created previously and receive your certificate file. Move that file to /etc/ssl/certs/mailserver.pem and set its permissions properly:

```
chmod go= /etc/ssl/private/mailserver.pem
```

In addition to the Apache configuration described below please ensure to install the chaining certificate, too. A sample configuration section is available in the StartSSL documentation.

Option 4: Paid certificate

Are you sure you want to do it? Then just search the web for "SSL certificate" and throw your money at any certificate authority.

The key and the certificate request are created as described above in Option 3.

Use the CSR file to request a certificate from the SSL authority. Move that certificate file to /etc/ssl/certs/mailserver.pem and set its permissions properly:

```
chmod go= /etc/ssl/private/mailserver.pem
```

Increasing TLS security

By the way – you should forbid communication using potentially insecure encryption methods. There is a way to attack the encryption by pretending that you do not support modern encryption techniques and thus rather want to use SSL version 2 or 3. This is called the POODLE attack. So you better run

```
postconf 'smtpd_tls_mandatory_protocols=!SSLv2,!SSLv3'
```

to forbid communication using pre-Snowden protocols that may be abused to attack you. (Thanks for the hint, Guillaume.)

Views: 67,009

Next page: Setting up the Apache web server

47 thoughts on “Creating a TLS encryption key and certificate”

Ace 2015-11-03 at 12:15 Permalink

You have wrong paths to mailserver.pem in your last two commands.
You should NOT use this:

```
chmod go= /etc/ssl/private/mailserver.pem
```

You should use this:

```
chmod go= /etc/ssl/certs/mailserver.pem
```

Reply

Christoph Haas Post author 2015-11-03 at 12:58 Permalink

Why? The certificate is the “public key” so to say. It does not need to be protected. It contains the data that is sent first when a secure connection is opened anyway. The private key however (located in /etc/ssl/private) must not be read by anyone except the mail server. Anyone who would have the private key could decrypt your otherwise secure traffic.

Reply

Marnix 2016-04-27 at 10:24 Permalink

I think the confusion is caused by the sentences above those two commands. They state “Move that certificate file to /etc/ssl/certs/mailserver.pem and set its permissions properly:”. The command that follows it then refers to the keyfile in /etc/ssl/private, instead of the certificate file that you just moved to /etc/ssl/certs. So the command is correct, but the sentence above it is confusing.

Great guide by the way!

Reply

Udo 2015-11-15 at 11:37 Permalink

Hi,

maybe someone need it also, here is explained how to install the chain certificat for postfix and dovecot:

https://gerritbeine.com/2014/02/postfix-und-dovecot-mit-startssl-zertifikaten/ [GERMAN!]

Greets Udo

Reply

Rulas 2015-11-16 at 09:56 Permalink

Thanks for the great tutorial, I was wondering, if I wanted to host mail for several domains, ie: example1.com, example2.org etc would a single ssl cert for example1.com work for example2.org and others?

Also, how would email be handled at a domain level? mx.example1.com and mx.example2.org point to the same ip?

Reply

Christoph Haas Post author 2015-11-16 at 10:05 Permalink

You could in theory create a certificate for several domains. But I haven’t seen a certificate authority that would sign such a certificate. When dealing with web servers you can get several certificates and thanks to modern SNI (server name indication) the web server can figure out which certificate it should use. But that won’t work for secure IMAP, POP3 and SMTP communication.

If several MX servers point to the same IP that’s no problem. If one server sends email to another then nowadays it will accept any certificate and initiate an encrypted connection.

Reply

Guillaume 2015-11-17 at 20:22 Permalink

By the way, you should add smtpd_tls_mandatory_protocols=!SSLv2,!SSLv3 in your main.cf to avoir the Poodle SSLv3 vulnerability

Reply

Christoph Haas Post author 2016-06-04 at 18:29 Permalink

Sorry for taking so long. I have added your hint to https://workaround.org/ispmail/jessie/create-certificate. Thanks.

Reply

cCred 2017-01-25 at 04:22 Permalink

Hi Christoph,
in your (awesome) guide Postfix has an opportunistic use of TLS (in the chapter about relaying you set “smtpd_tls_security_level” to “may”).
So the correct parameter is
smtpd_tls_protocols = !SSLv2, !SSLv3 # Postfix as SMPT Server
smtp_tls_protocols = !SSLv2, !SSLv3 # Postfix as SMTP client
But don’t be afraid, this is the default value for all Postfix releases after the middle of 2015. 😉

Reply

Will 2015-11-26 at 18:42 Permalink

Hi,

Just to mention that letsencrypt.org is issuing free certs that are validated, a good alternative to the commercial things and to the self signed.

W.

Reply

Keith Vella 2015-12-03 at 11:43 Permalink

Hi,

First of all thanks for the amazing tutorial. I am stuck trying to install the chain certificate. On StartSSL it is not so clear and I don’t have any experience setting this up.

Keith

Reply

Christoph Haas Post author 2015-12-07 at 10:21 Permalink

HI Keith. When StartSSL offers you to download (copy/paste) the actual certificate then there are links to the CA and the Intermedia certificates. Download both. And then put these two plus your own certificate all into one “mailserver.crt” file. That will do it.

Reply

Jon Singer 2016-08-18 at 18:00 Permalink

Hi Christoph, Thank you for this wonderful guide. I am confused slightly because I believe the StartSSL website has changed. I am trying to properly install certificates. When I receive my certificates from StartSSL I get a downloaded file called “mydomain.com.zip” inside that file are three more .zip files: ApacheServer.zip, IISServer.zip, NginxServer.zip and OtherServer.zip. Inside each of those four zip files are: ApacheServer.zip: 1_root_bundle.crt, 2_mydomain.com.crt, IISServer.zip: 1_Intermediate.crt, 2_mydomain.com.crt, NginxServer.zip: 1_mydomain.com_bundle.crt and OtherServer.zip: 1_Intermediate.crt, 2_mydomain.com.crt, root.crt.

What do I put where? I currently have /etc/ssl/certs/mailserver.csr and /etc/ssl/private/mailserver.pem. I am confused about your instructions as well as the comments as they seem to conflict. I would also like to know how and where I should put the intermediate certificate.
Thank you for your tutorial!

Reply

Correy 2015-12-11 at 22:12 Permalink

Hey Christoph,

I've read through the entire tutorial and I've got to say, this is astounding! I do want to leave a bit of extra info on installing the chaining cert though. I feel like this will be a massive roadblock for a lot of readers. So hopefully this saves some time…

First,

# cd /etc/ssl/private

Next, we're going to obtain the Class 1 Intermediate Server CA by typing,

```
# wget https://www.startssl.com/certs/sub.class1.server.ca.pem
```

To combine your certificate with the newly obtained "sub.class1.server.ca.pem" certificate, you're going to use the cat command as follows (assuming YOUR certificate is still in /etc/ssl/certs)

```
# cat /etc/ssl/certs/mailserver.pem sub.class1.server.ca.pem > ssl-chain-mail-yourdomain.pem
```

This should place the new chaining certificate in your /etc/ssl/private directory.

Let me know if you disagree with this.

Cheers

Reply

Peter Gutwein 2015-12-15 at 15:41 Permalink

A very good alternative way is to use a letsencrypt certificate. It's free of cost, trusted and very easy to install on a modern Debian Server (automatic Installation via cert Client). In the meantime the letsencrypt Project is in Status Open Beta. It worked fine for me. Have a look at https://letsencrypt.org/ if u are interested.
Regards Peter

Reply

Said El Mazghari 2015-12-23 at 10:11 Permalink

Hey could anyobody please tell me what/where to find the “certificate wizard” is?
thanx

Reply

Paul 2015-12-23 at 13:43 Permalink

StartSSL.com have launched a new website. The old certificate wizard is no longer there.

You need to click on “Start Now for Free SSL Certificate” and then “Sign Up” to validate who you are.

But be warned, the site at the moment is at a crawl.

Reply

Hadi 2016-01-01 at 18:49 Permalink

For certificates: you can get a totally legitimate free SSL certificate from the newly started project (funded by EFF and Mozilla) called Let’s Encrypt: https://letsencrypt.org/.

Reply

Robert 2016-01-10 at 14:47 Permalink

+1

Works like a charm over here! Worth mentioning.

Reply

abnquet 2016-03-04 at 11:13 Permalink

I think this is worth including in this or the next tutorial. The letsencrypt packages are now in debian-backports, so the setup is actually quite easy.

Reply

Peter Thylander 2016-02-03 at 15:43 Permalink

I Went for option three, but not ran into problems.
§ Dns records need to be set.
§ A running mailserver needs to be online on the domain you need to validate (accordning to godaddy).

Therefore, I suggest that you move at least option three and four until the mail server is online.
Regards

Reply

Robin Martens 2016-02-03 at 15:48 Permalink

People looking for an alternative might also want to look at Let's Encrypt.

Reply

Jan 2016-03-09 at 10:14 Permalink

I am trying to use option 3 (startssl), but it seems that something changed in their system (or i made a big mistake).
I didnt got a pem-file, i got a zip with many crt-files and now i dont understand how to install this (which file did i have to use for replace my /etc/ssl/private/mailserver.pem) ?

Thanks in advance
Jan

Reply

Jan 2016-03-10 at 10:17 Permalink

It seems i got it:

unzip File from StartSSL
unzip Apache file and got 2 files – used the second (2_myhost.mydomain.somewhere)

```
wget https://startssl.com/certs/sca.server1.crt # its PEM Version
cat 2_myhost.mydomain.somewhere sca.server1.crt >
/etc/ssl/certs/mailserver.pem
```

to test it u have to get the root-ca file from startssl

```
wget https://startssl.com/certs/ca.crt
mv ca.crt /etc/ssl/certs/ca.pem
openssl s_client -connect localhost:25 -starttls smtp -CAfile
/etc/ssl/certs/ca.pem
```

or use the existing one:
openssl s_client -connect localhost:25 -starttls smtp -CAfile /etc/ssl/certs/StartCom_Certification_Authority.pem

I hope thats the correct way – any advices?

Reply

Christoph Haas Post author 2017-01-01 at 22:05 Permalink

Thanks for the reminder and the pointers. As much as I liked StartCom I have replaced the recommendation by LetsEncrypt. Also I consider LetsEncrypt pretty much a mess of scripts and a hassle of downtimes every 90 days. 🙁

Reply

Joe M. 2017-02-03 at 12:22 Permalink

Hi Christoph – thanks so much for the tutorials, I've been using them for years – since sarge I think?

Just a couple of things – using certbot as instructed on the letsencrypt site works perfectly with your guide, is fully automated and extremely simple to install.

Also can I recommend BitDefender rather than spamassassin? Free, great WebGUI and does a much better job for me out of the box (with daily updates) than feeding SA 10,000 of good and bad mail.

Reply

Jan 2016-03-16 at 10:26 Permalink

Made another cert for my webserver. After i compared the files, i noticed: U can use both files from Apache folder (maybe one of the other files are ready to use, but this way you may better understand the cert things):

cat 2_myhost.mydomain.somewhere 1_root_bundle.crt > /etc/ssl/certs/mailserver.pem

still learning every day 🙂

Reply

Marnix 2016-04-27 at 13:56 Permalink

In case anyone else is confused about the meaning of the certificate files that you receive from startssl and why these can be concatenated: I found this page very informative: http://theheat.dk/blog/?p=534. A brief summary: it is advised that you not only provide your own certificate, but also the intermediate certificate. If you don’t provide the latter, most browsers will download it themselves, but it is not best practice.

In our scenario we need both the 1_root_bundle.crt and the 2_myhost.mydomain.somewhere and they can indeed be combined into a single PEM file as per instructions of Jan above here. But you can also keep them in two separate files. I have renamed the 1_root_bundle.crt file to mailserver_intermediate.pem. In this case you need to enable the following in the /etc/apache2/sites-enabled/default-ssl.conf file:

```
SSLCertificateFile /etc/ssl/certs/mailserver.pem
SSLCertificateKeyFile /etc/ssl/private/mailserver.pem
SSLCertificateChainFile /etc/ssl/certs/mailserver_intermediate.pem
```

In case some other component needs the concatenated pem file it can't hurt to keep this one as well. Mine is called mailserver_with_chain.pem.

Reply

Davide Marchi 2016-08-31 at 16:29 Permalink

Hey Christoph,
Thanks very much for your great tutorial, which is very special because it describes the procedures considering the person on the other side. Which is rather rare and denoting great sensitivity and a real educational spirit!

With regard to the question SSL instead, I would like to suggest this project: "https://github.com/lukas2511/letsencrypt.sh".

What do you think about this? Could be useful to integrate your tutorial with this resource?

Thank you

Davide

Reply

Xandor Schiefer 2016-09-20 at 19:59 Permalink

Perhaps you could update the guide to include Let's Encrypt as an option? I've been using one of their certificates without issue (for mail).

Reply

Jeff 2017-01-01 at 16:42 Permalink

Did a little digging around and WoSign / StartSSL are "too good to be true"

In fact, they are now distrusted by Google, Mozilla, and Apple for a variety of reasons, including deceitful practices, as I read the releases.

https://security.googleblog.com/2016/10/distrusting-wosign-and-startcom.html

https://blog.mozilla.org/security/2016/10/24/distrusting-new-wosign-and-startcom-certificates/

https://support.apple.com/en-us/HT204132

Reply

L Jones 2017-01-05 at 21:18 Permalink

I was a big fan of StartSSL until Google stopped trusting them. For ubuntu 16.10 with apache2 working (http:), LetsEncrypt setup took 3 commands:

```
# Get the certification bot:
sudo apt-get install python-certbot-apache
# Run the bot, answer questions, it even configures apache:
certbot –apache
# Load the updated apache site config file:
sudo service apache2 restart
```

After that, browser refresh returns https:// – that's all it took!
For background, see https://letsencrypt.org/getting-started/
which should guide you to: https://certbot.eff.org/
Pick your configuration and go.

Reply

Leigh the noob 2017-01-23 at 21:44 Permalink

Great Tutorial here so far. I am a bit of a noob with linux and using it to host my own mail server as MS Exchange has become way too expensive now days. Kinda dumb question, but hostnames:

I have set the servers hostname as: ServerName.MyDomain.local
When setting up postfix, do i still use the .local FQDM or a .com to match my owned public domain?
and when it comes to the certificate FQDN, do i want it to be a .local or

ServerName.MyDomain.com or a DNS A Record that points to my server (eg mail.mydomain.com)?

Probably a really dumb question but always had a bit of confusion in this area if someone can clear it up for me please?

Thanks!

Reply

Ralf L. 2017-01-24 at 09:48 Permalink

Hi
I have finished the setup about 3 weeks ago. Also 'noobs level' – and on a different OS (openSUSE).
Took me about a month to do it.
I found best is to have a FQDN server name (thus in your case of the .com domain). Though it is also ok with a .local. But than chances are higher that you will end up on blacklists and/or spamlists.

I also use a PTR record in DNS (DNS reverse lookup) and also a SPF record to prevent ending up on those lists.
I also have added the hostname to my certificate.

Reply

Leigh the noob 2017-01-24 at 15:26 Permalink

Thanks for the reply Ralf.

So are you saying on the certificate you used the servers actual hostnamename (eg Servername.MyDomain.com) or a a record that points to that server (eg mail.mydomain.com)? or both? I guess putting both could not hurt…

I am using cloudflare for my DNS hosting (because its free) and cannot create a PTR record. is there any way I can get around this without switching to another DNS host that you are aware of? I am guessing adding the entry to my domain local DNS wont do the trick?

Reply

Jeff 2017-01-24 at 19:32 Permalink

You definitely want your SSL cert to read the mail server name. as in, when you setup thunderbird, if your incoming mail server is mail.mydomainisgreat.com then your SSL certificate should be for mail.mydomainisgreat.com because that’s what email clients will check against when they talk to your server, i used let’s encrypt and did mail.mydomain.com and mydomain.com both in the cert so the webpage mydomain.com is covered as well for my webmail users.

Another thing that stumped me and I think is confusing you as well I think the creator of these wonderful guides explains very well on the page about types of email domains. Read through that page again. He recommends a setting for ‘mydestination’ to be sure all incoming email is seen as virtual domains. The recommendation comes in the guide before you actually install the OS on your server so i hadn’t gone back to change that setting until much later. I think the explaination about types of domains and server names will help you better than i can explain. But the actual name you give your server locally shouldn’t cause problems as long as you have dns records that points emails to your server, a mx record and then a matching A record for that. I hope this helps.

Reply

Leigh the noob 2017-01-25 at 16:28 Permalink

Thanks for the advice Jeff. Very helpful although still a little bit of confusion. From what I have gathered from your advice, my setup I presume should be as follows (might help some other noobs):

– hostname = MyServer.MyDomain.local

– FQDN for postfix = MyServer.MyDomain.local (you dont use MyServer.MyDomain.com becasue you would have to create system accounts for each user, it is impractical and postfix cant distinguish the local domains)

– mydestination = localhost (because we are using virtual domains)

– virtual_mailbox_domains = MyDomain.com, ClientDomain1.com, ClientDomain2.com etc… (defines the domains the server will receive mail to)

– My certificate will have the name: mail.MyDomain.com (becasue that is what email setup will use and it will match the cert) I am unsure how you said you put just MyDomain.com on a cert as they do not allow wildcards?? Did you mean a http://www.MyDomain.com for your website??

I gather this is the correct way to go about things? Confirmation would be great although trying again now using this config anyway.

Jeff 2017-01-24 at 19:35 Permalink

oh, also, as far as the PTR setup, the host of your DNS cannot do that, so switching DNS hosts wont help, the owner of your IP address has to do that. I.E. your internet provider, or if you’re renting a VPS, the company that’s coming from might let you. Godaddy for example doesn’t allow PTR’s for their servers, but they allow you to use their outgoing email server so you’d need to configure your mail server to relay through godaddys mail server instead of sending directly.

Reply

Ralf L. 2017-01-25 at 00:48 Permalink

– I use 'servername.mydomain.com' as my hostname
– from the register of my domain I point my domain to the DNS of my VPS provider
– I use my hostname and public IP from my VPS for a PTR record. I have a VPS hosted and the provider allows me to set this myself on their system.
– What helped me to test the server is this site:
http://mxtoolbox.com/

Hope you will be able to work everything out. Like I said. It took me about 4 weeks to understand, test and adapt to my situation (since I do not use Debian – though the differences are minor).

Reply

Leigh the noob 2017-01-25 at 16:34 Permalink

Thanks again Ralf. So are you only hosting emails for one domain? not multiple? From what was mentioned on the page about different domains I got the impression that using your servername.mydomain.com had 3 drawbacks.

Your PTR record I think might only be relevant in a VPS scenario where there are multiple domains and such sharing one IP. I dont think I should need it in my case as it is to be hosted from my home office (hopefully). So long as I have MX and possibly an autodiscover record (or is that just a microsoft exchange thing?) I should be ok I think.

Damn linux can be frustrating 😛 Problem is once you start you become determined to make the damn thing work!

Ralf L. 2017-01-26 at 02:38 Permalink

Hi Leigh

I use 3 domain. (lets say mydomain1.com, mydomain2.com, mydomain3.com)
See below a port of my main.cf for postfix:

```
myhostname = server.mydomain1.com
#mydomain =
mydestination = localhost
mynetworks = 127.0.0.0/8
inet_interfaces = all
virtual_mailbox_domains = mysql:/etc/postfix/mysql-virtual-mailbox-domains.cf
```

As of this manual the virtual mailbox domain will be looked up from the database.

This is what a received e-mail header looks like from my server:

```
Return-Path:
Delivered-To: me@mydomain2.com
Received: from server.mydomain1.com
by server.mydomain1.com (Dovecot) with LMTP id codyIx4BiViheQAAAMb7Tw
for ; Wed, 25 Jan 2017 20:48:46 +0100
Received: from localhost (localhost [127.0.0.1])
by server.mydomain1.com (Postfix) with ESMTP id 6BB3432013B
for ; Wed, 25 Jan 2017 20:48:46 +0100 (CET)
Authentication-Results: server.mydomain1.com;
dkim=pass (1024-bit key) header.d=email.vimeo.com
header.i=vimeo@email.vimeo.com header.b=ZuTgtk9W;
dkim-atps=neutral
X-Virus-Scanned: amavisd-new at mydomain1.com
Received: from server.mydomain1.com ([127.0.0.1])
by localhost (server.mydomain1.com [127.0.0.1]) (amavisd-new, port 10024)
with ESMTP id Jkolz7z9Ogn7 for ;
Wed, 25 Jan 2017 20:48:42 +0100 (CET)
```

The PTR: Even though I use 3 domains – it is 'server.mydomain1.com' which always sends my mail, therefor my public IP resolves to server.mydomain1.com in the PTR. If a receiving party can not resolve my public IP to server.mydomain1.com (lets say you set it to server.mydomain.local) it might be considered as spam and rejected (depending on how strict their server is setup to accept mail). So 'myhostname=' in postfix is the same as in your PTR record.

{Hosting the server at your home on a DSL/cable connecting can have serious drawbacks. You should do some reseach before you go ahead. For example my porvider would not allow me to send mail from my home (smtp from my server). I would have to specify another smtp server to send mail from in my case. But what is the point than if I am only allowed to do half of the setup from my home?}

pbw 2017-03-21 at 01:30 Permalink

The letsencrypt process I find easiest and most reliable is getssl. https://github.com/srvrco/getssl/wiki

It's a bash shell script and it works a treat.

Reply

Yaniv 2017-05-07 at 20:48 Permalink

Should you be revising this post to note that startSSL should not be used?

Reply

Christoph Haas Post author 2017-05-07 at 21:22 Permalink

Absolutely. And I apologize that I haven’t yet managed to fix that. 🙁

Reply

David 2017-09-01 at 09:11 Permalink

Hello again,
Some more feedback:
I know you’re going to include details of how to use Letsencrypt (LE) certificates in your tutorial for Debian “stretch” so I’d just like to comment in case anyone’s interested in doing this in “jessie.” Let me say first—just to clear any lingering doubts any readers may have—that a LE certificate will work perfectly for providing LS – standard encryption.

I have to make do with only one dedicated server with one 1 IP address. In order to successfully use a Letsencrypt certificate for both the website and the first virtual domain I can confirm that the configuration given at https://skippy.org.uk/lets-encrypt-postfix-and-dovecot/ works. I used the configuration for postfix without any changes, but had to make a minor change in the path in dovecot from the path given in the referenced howto, otherwise I couldn’t log in to Roundcube. I also had to disable inward bound spf, configured as per https://words.bombast.net/postfix-with-spf-dkim-and-dmarc/

```
# path to the certificate file, should be root:root and 0444
ssl_cert = </etc/letsencrypt/live/example.com-0001/fullchain.pem
```

```
# path to the private key file, should be root:root and 0400
ssl_key = </etc/letsencrypt/live/example.com-0001/privkey.pem
```

So at present I can now send email, for for example to gmail without the Red (no encryption) padlock appearing informing the recipient that such mail is not secure.

So far so good; at least I've secured my most important email account with LE so now it's a question of getting the other accounts to work with the same certificate, assuming this is possible since —I assume—the other virtual domains included in the certificate are technically aliases of the base domain. They all get the green light when checking at http://www.checktls.com
At present I'm not sure how this may be done —if it can be done— for the devil is in the detail.

Regards,

Reply

David 2017-09-04 at 20:47 Permalink

To add to my last post where I said that I could only get my primary domain to work and hadn't figured out how to use ONE Letsencrypt (LE) certificate for multiple virtual domains.
Well, I 've now figured it out as I posted on the Debian forums at http://forums.debian.net/viewtopic.php?f=3&t=134383#p653666 where I tried, without success to date, to get a discussion going about the the subject of self-issued, paid and unpaid certificates. The way I did it is posted there.
All virtual hosts work sharing only one IP work with one LE Certificate, the primary domain and all the others. I have tested this thoroughly with several domains and so I can confirm it.

Reply

David 2017-09-06 at 05:53 Permalink

I left another reply but may have appended the wrong extension to my email address because so far it hasn't appeared here.
Anyway, I'd just like to add to my previous reply to say that I've got ONE Letsencrypt Certificate issued for my primary domain to work with multiple virtual hosts which all sharing only one IP address. No problems at all with the certificate. The same little dedicated server also doubles as a web server, again using the same certificate. I wrote about it recently on the Debian forums at: http://forums.debian.net/viewtopic.php?f=3&t=134383#p653666
But now that everything's working as it should and email passes all the

tests with the LE certificate, spf, dkim & dmarc—and the server is not blacklisted anywhere— yet, to my surprise, several emails sent to a hotmail account got blocked when they get delivered immediately to gmail accounts.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Post Comment