

Mail Server

Andrea Gussoni
andrea at gussoni.ovh

P.O.u.L.

12 Aprile 2017



POLITECNICO OPEN
unix LABS
Come hack with us.

Motivations

- Why bother to configure and manage our own mail server?:
- Can't I just use **THE CLOUD™**?

Motivations

Cambridge Analytica used data from Facebook and Politico to help Trump

Speech by company executive contradicts denial by Trump campaign that claimed the company used its own data and Facebook data to help the campaign



▲ Cambridge Analytica's CEO, Alexander Nix, at the company's office in New York City on 24 October 2016.
Photograph: The Washington Post/Getty Images

⁰<https://www.theguardian.com/technology/2017/oct/26/cambridge-analytica-used-data-from-facebook-and-politico-to-help-trump>

Advantages

- We have control on our data
- We can provide a service to a third party
- We do not depend on a Big Corp IncTM.
- We have full control on the domains we own and we can do nice things like having a *catchall* domain

Disadvantages

- We are responsible for our data (i.e. encrypt (and backup) all the things!)
- We are responsible for the uptime of the service (especially important if we provide this service to a third party)
- We have to actually understand how things work under the hood and we need to spend time configuring a server (but this is actually an **advantage**, isn't it?)

How does email work?

The principal actors:

- Mail User Agent (**MUA**): is the email client
- Mail Transfer Agent (**MTA**): is the component that transfers the email from one computer to another
- Mail Delivery Agent (**MDA**): is the component that delivers the email to the user inbox

You can find the detailed description of this architecture [here](#)

Protocols

Actor	Operation	Protocol	Port
MTA ↔ MTA	Forward	SMTP ¹	25, 587
MUA ↔ MTA	Send		
MUA ↔ MDA	Receive	POP3 ²	110, 995
		IMAP ³	143, 993

¹Simple Mail Transfer Protocol (RFC 5321)

²Post Office Protocol 3 (RFC 1939)

³Internet Message Access Protocol (RFC 3501)

Example

What does happen when we send an email message?

Let's suppose we want to send an email from the account **sender@mrobot.ovh** to **receiver@poul.org**

Example

1. The client connects to the SMTP server and wants to send an email to an address belonging to the **@poul.org** domain



smtp.mrobot.ovh

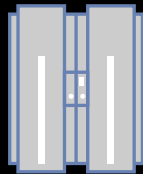
Example

2. The SMTP server asks for the MX record of **poul.org**



smtp.mrobot.ovh

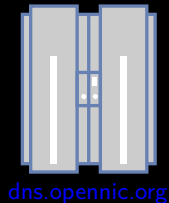
MX for poul.org?



dns.opennic.org

Example

3. The SMTP server obtains the **MX** record (in our case we obtain **smtp.poul.org**)



Example

4. Our SMTP server connects to the SMTP server of the receiver and delivers the email



smtp.mrobot.ovh

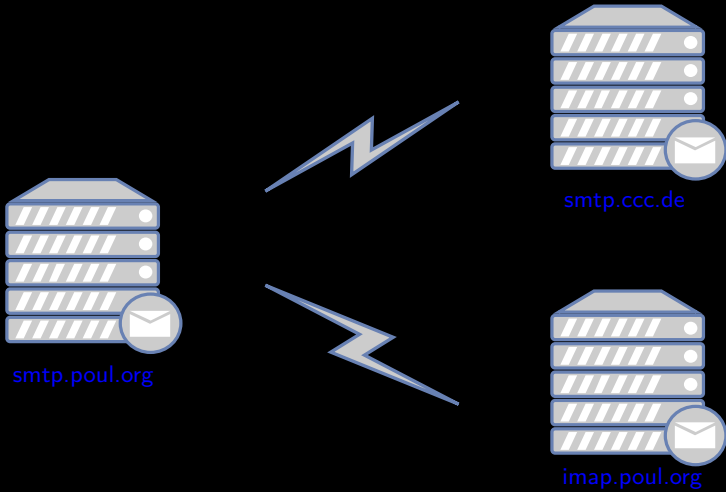


smtp.poul.org

Example

5. At this point the receiver SMTP server can:

- Act as a forwarder and forward the email to another MTA
- Forward the email to a MDA that will deliver it to the final receiver



Our Mailserver

Nice, but how can I actually build my email server? We will need:

1. A email client as our MUA: Thunderbird, Mutt...
2. A MTA: Postfix (which we will use), Exim
3. A MDA: Dovecot

Postfix

Some details on Postfix:

- Released under the IBM Public License 1.0 free software license
- The SMTP server, the main component in our architecture
- Made up of many components
- Highly configurable

Domains and users

We can have:

- A single domain, and different system account
- Multiple domains with multiple accounts

During the demo we will use a single domain and system account for simplicity

Installation and configuration

- We can install Postfix with a simple `sudo apt-get install mailutils` (or equivalent)
- We have to edit some files:
 - `/etc/mailname`
 - `/etc/postfix/main.cf`
 - `/etc/postfix/master.cf`
- The configurations files by default contain a lot of commented directives that we can enable at our pleasure, so reading carefully the comments above a certain option is always a good thing to do
- If we want a localhost only SMTP server (useful to enable services to send email or to experiment a bit) we can modify the line `inet_interfaces = all` to `inet_interfaces = localhost`

Other important configurations

- Set the MX record on your domain provider control panel (otherwise you will not receive any email)
- Obtain an SSL certificate and configure it (we will see more on this later)
- Specify an additional method to authenticate users (we will use SASL with Dovecot)

Demo time



Dovecot

- Our MUA
- We will use it also for authenticating the users with Postfix
- We will configure it for using the IMAP protocol to log into our mailbox from the client

Installation and configuration

- We can install it with `sudo apt-get install dovecot-imapd`
- The main configuration file is located in `/etc/dovecot/dovecot.conf`
- In `/etc/dovecot/conf.d` we can find additional configurations files

Basic configuration

/etc/dovecot/dovecot.conf

```
# Protocols we want to be serving.  
protocols = imap
```

```
# A comma separated list of IPs or hosts  
# where to listen in for connections.  
listen = $machineip
```

Basic configuration

/etc/dovecot/conf.d/10-master.conf

```
service imap-login {  
    inet_listener imap {  
        port = 143  
    }  
    inet_listener imaps {  
        port = 993  
        ssl = yes  
    }  
}
```

Basic configuration

/etc/dovecot/conf.d/10-auth.conf

```
# Enables the PLAIN auth
disable_plaintext_auth = no

# Enables the authentication using a file
# for storing credentials
!include auth-passwdfile.conf.ext
```


Basic configuration

/etc/dovecot/conf.d/auth-passwdfile.conf.ext

```
passdb {  
  driver = passwd-file  
  args = scheme=CRYPT username_format=%u /etc/dovecot/  
        users  
}  
  
userdb {  
  driver = passwd-file  
  args = username_format=%u /etc/dovecot/users  
}
```

File based auth

- Each line of the file contains the account name and the hash of the password
- Remember to use strong passwords and hash functions (e.g. MD5 without salt is not a great choice)

/etc/dovecot/users

```
andrea:{SHA512-CRYPT}$6$G...:1000:1000::/home/andrea::
```

Postfix auth

We now need to configure Postfix in order to authenticate the users using the SASL mechanism provided by Dovecot⁴:

/etc/dovecot/conf.d/10-master.conf

```
service auth {
  ...
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    # Assuming the default Postfix user and group
    user = postfix
    group = postfix
  }
  ...
}

auth_mechanisms = plain login
```

⁴<https://wiki2.dovecot.org/HowTo/PostfixAndDovecotSASL>

Postfix auth

/etc/postfix/main.cf

```
smtpd_sasl_type = dovecot
```

```
# Can be an absolute path, or relative to
```

```
# $queue_directory
```

```
smtpd_sasl_path = private/auth
```

```
# and the common settings to enable SASL:
```

```
smtpd_sasl_auth_enable = yes
```

TLS

- Obtaining and configuring a certificate for our server is really important to avoid credentials sniffing and Man-in-the-middle-attacks.
- We can either generate a self-signed certificate (useful only for testing or experimental purposes) or obtain a valid certificate for free using Let's Encrypt⁵
- Remember that Postfix by default uses STARTTLS, if you incur in client side problems with STARTTLS you can also enable strict TLS on a dedicated port⁶

⁵<https://letsencrypt.org/getting-started/>

⁶http://www.postfix.org/TLS_README.html

TLS

To configure Postfix for using our new certificates we need two files:

- The complete chain of certificates (usually stored in the file *fullchain.pem*)
- Private key (usually stored in the file *privkey.pem*)

TLS

/etc/postfix/main.cf

```
smtpd_tls_cert_file=/etc/letsencrypt/live/mrobot.ovh/  
    fullchain.pem  
smtpd_tls_key_file=/etc/letsencrypt/live/mrobot.ovh/  
    privkey.pem  
smtpd_tls_security_level = may  
smtpd_use_tls=yes  
smtp_tls_security_level = may  
smtp_tls_loglevel = 1  
tls_ssl_options = NO_COMPRESSION  
tls_high_cipherlist=omissis  
smtpd_tls_protocols=!SSLv2,!SSLv3  
smtp_tls_protocols=!SSLv2,!SSLv3  
smtpd_tls_mandatory_ciphers = high  
smtpd_tls_mandatory_protocols = !SSLv2,!SSLv3  
smtpd_tls_exclude_ciphers = aNULL, LOW, EXP, MEDIUM,  
    ADH, AECDH, MD5, DSS, ECDSA  
    , CAMELLIA128, 3DES, CAMELLIA256, RSA+AES, eNULL
```

TLS

- To better understand what's going on here please refer to more detailed sources like the Postfix manual⁷
- Take into consideration that if you don't encrypt your connections, in addition to be vulnerable to attacks when authenticating to the server, your emails will flow unencrypted between different MTA (unless you encrypt the content with GPG⁸ or another mechanism)
- Crypto is hard, so I suggest you to spend some time trying to understand and configure consciously all these parameters.

⁷http://www.postfix.org/TLS_README.html

⁸<https://gnupg.org/>

TLS



If you can't understand what's going on here, remember that the default suggested configurations should be reasonable for a standard use, so unless you *Roll your own crypto* you should be fine

TLS

We also need to configure Dovecot to use the certificates we obtained:

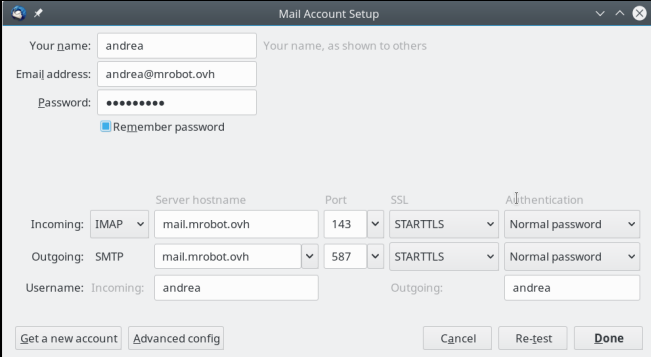
```
/etc/dovecot/conf.d/10-ssl.conf
```

```
# SSL/TLS support <doc/wiki/SSL.txt>
ssl = required

# PEM encoded X.509 SSL/TLS certificate and private
  key.
ssl_cert = </etc/letsencrypt/live/mrobot.ovh/fullchain
  .pem
ssl_key = </etc/letsencrypt/live/mrobot.ovh/privkey.
  pem
```

Thunderbird configuration

An example of a thunderbird configuration working with our setup:



The screenshot shows the "Mail Account Setup" window in Thunderbird. The window title is "Mail Account Setup". The fields are filled with the following information:

- Your name: andrea (Your name, as shown to others)
- Email address: andrea@mrobot.ovh
- Password: [masked]
- Remember password

The server configuration section is as follows:

	Server hostname	Port	SSL	Authentication
Incoming:	IMAP mail.mrobot.ovh	143	STARTTLS	Normal password
Outgoing:	SMTP mail.mrobot.ovh	587	STARTTLS	Normal password

Username configuration:

- Incoming: andrea
- Outgoing: andrea

Buttons at the bottom: Get a new account, Advanced config, Cancel, Re-test, Done.

Demo time



SPAM

- We have configured our new shiny mail server, but when we try to send an email to an existing account we are marked as SPAM
- This is normal, since there are some techniques and standard to respect in order to avoid being marked as SPAM and end up in some blacklists

SPAM

The main things we need to look at are:

- Reverse DNS and PTR record (rDNS)
- Sender Policy Framework (SPF)
- DomainKeys Identified Mail (DKIM)

rDNS

- This is easy, all we need to do is to configure the PTR record of our machine to point to the hostname of the server (e.g. *mail.mrobot.ovh*)
- Keep in mind that you need to do this on the control panel of your VPS/Dedicated Server/whatever, not in the DNS Zone records of your domain name provider
- We can verify the record with *host desired-ip-here*

SPF

- It's a method for specifying the association between a domain and the SMTP servers allowed to send emails from that domain
- It's a TXT DNS record that you need to set on the DNS Zone records of your domain name provider
- You can find a detailed guide about the policy here⁹
- For a standard use the suggested policy is: `"v=spf1 mx -all"`
- Or with soft-fail: `"v=spf1 mx ~all"`

⁹<https://www.digitalocean.com/community/tutorials/how-to-use-an-spf-record-to-prevent-spoofing-improve-email-reliability>

DKIM

- DKIM is a digital signature mechanism that consists in apposing a signature to the mail when sending it, and verifying the signature, using a public key retrieved from an ad-hoc DNS record, at the other end.
- I suggest using OpenDKIM for this purpose
- Since the configuration is really long but not difficult or interesting per-se, I suggest you to follow this guide¹⁰ to setup DKIM

¹⁰<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-dkim-with-postfix-on-debian-wheezy>

DMARC

- DMARC stands for *Domain-based Message Authentication, Reporting & Conformance*, is an additional mechanism that combines the informations generated by SPF and DKIM for preventing and identifying SPAM
- You can have a look here¹¹ for more informations

¹¹<https://dmarc.org/>

Virtual Alias

- With Postfix we can have account virtual aliases (and also virtual domains)
- We need to configure them in the `/etc/postfix/virtual` file
- We can, for example, forward the email received at an address to another address:

```
info@mrobot.ovh andrea@mrobot.ovh
```

- Or even have a catch-all domain:

```
@mrobot.ovh andrea@mrobot.ovh
```

- Take a look at this reference¹² for more detailed informations about this and also about virtual domains

¹²http://www.postfix.org/VIRTUAL_README.html

Spamassassin

- Spamassassin is a powerful tool we can use for trying to identify and prevent SPAM on our side
- Once you will start using your server in the real world, and you will finish in some SPAM lists, you'll start receiving a lot of undesired emails
- Configuring spamassassin¹³ and tweaking it might be not so easy, but might be worth in the long run

¹³<https://www.digitalocean.com/community/tutorials/how-to-install-and-setup-spamassassin-on-ubuntu-12-04>

Sieve filters

- The Sieve plugin can be really helpful for automatically organizing the received emails in different folders
- It has a simple syntax, for example this configuration:

```
require ["fileinto", "reject"];

if address :contains ["From"] "spam@spam.com" {
    fileinto "INBOX.spam";
} else {
    keep;
}
```

Moves all the email received from *spam@spam.com* into a dedicated folder.

- You can find a manual here¹⁴
- Remember to configure Sieve using the `lmtp` protocol support in Dovecot

¹⁴<https://wiki.dovecot.org/Pigeonhole/Sieve>

Verifying our configuration

We can use these services to check how good/bad we have configured our mail server or if we are on some blacklist:

- `https://www.port25.com/authentication-checker/`
- `https://mxtoolbox.com/domain/`

Special Thanks

I want to thank the authors of the last editions of this talk, whose material I used as a starting point for preparing this talk. In order:

- Alessandro Di Federico¹⁵
- Emanuele Santoro¹⁶
- Massimo Maggi¹⁷

¹⁵[https:](https://www.poul.org/wp-content/uploads/2015/03/presentation.pdf)

[//www.poul.org/wp-content/uploads/2015/03/presentation.pdf](https://www.poul.org/wp-content/uploads/2015/03/presentation.pdf)

¹⁶[https:](https://www.poul.org/wp-content/uploads/2014/04/posta.pdf)

[//www.poul.org/wp-content/uploads/2014/04/posta.pdf](https://www.poul.org/wp-content/uploads/2014/04/posta.pdf)

¹⁷[http:](http://www.poul.org/wp-content/uploads/2012/06/postfix.pdf)

[//www.poul.org/wp-content/uploads/2012/06/postfix.pdf](http://www.poul.org/wp-content/uploads/2012/06/postfix.pdf)

References

- <http://www.postfix.org/documentation.html>
- <https://wiki.dovecot.org/>
- <https://www.digitalocean.com/community/tutorials/how-to-configure-a-mail-server-using-postfix-dovecot-mysql-and-spamassassin>
- <https://github.com/tomav/docker-mailserver>

License

Thank you!



These slides are published under a Creative Commons Attribution-ShareAlike 4.0 license.