Setup Postfix
Mail Transfer Agent (MTA)

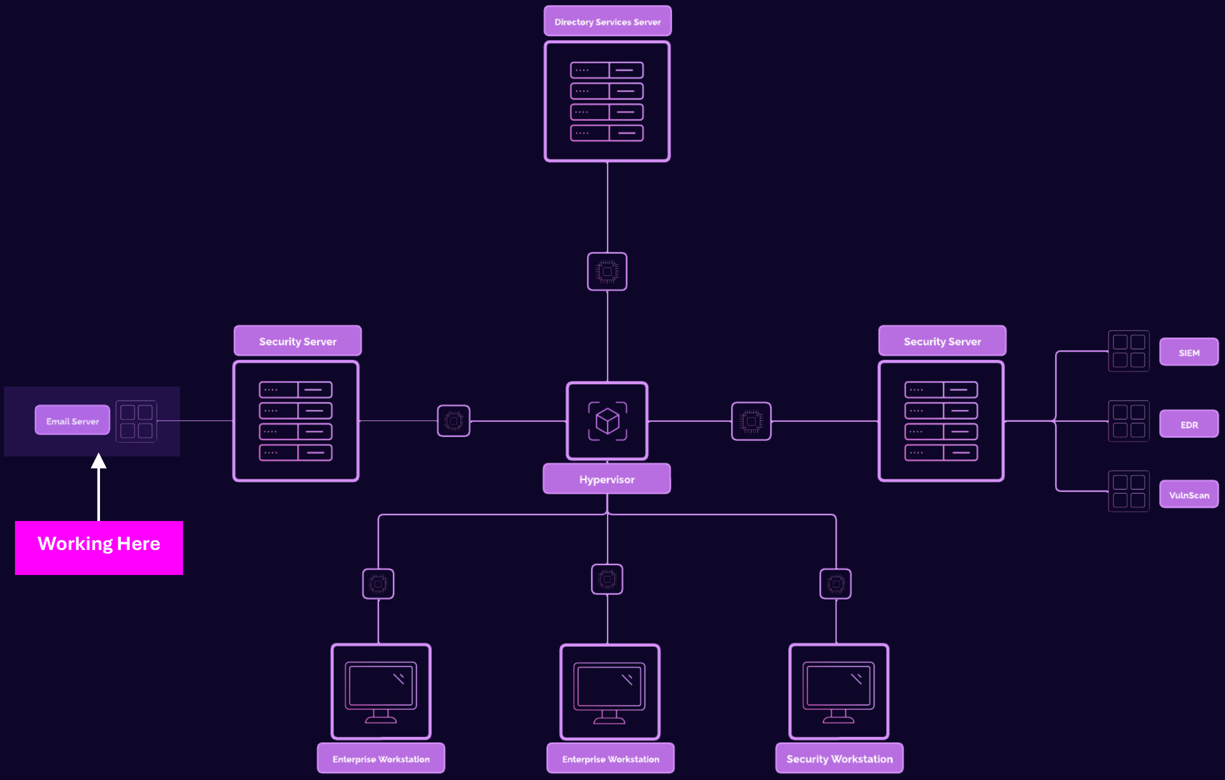
>projectsecurity_

Table of Contents

Prerequisites

1. Virtualbox installed.
2. Virtual Machine with Ubuntu 22.04 ISO Server has been provisioned, configured, and fully setup.
3. Windows Server 2025 with Active Directory Domain Services (ADDS) configured.

Network Topology



Postfix Overview

Overview

Postfix is a widely used open-source mail transfer agent (MTA) that routes and delivers email messages. Postfix is commonly used on Unix-like systems, including Linux distributions, as a server-side solution for managing email.

Postfix acts as an intermediary in the email delivery process. It receives email from a user or an application, processes it, and delivers it to the intended recipient. It supports various protocols and standards such as Simple Mail Transfer Protocol (SMTP).

It is estimated that 25% of public email servers run Postfix.

Postfix can be used in various scenarios, including as an SMTP server, mail forwarder, spam filter, relay server (SMTP Relay), and more.

We will be configuring Postfix as a very simple SMTP server to send and deliver email to our simulated workstations (…`-win-client` & …`linux-client`).

Security Implications

Running an email server introduces several security considerations:

Common Threats

1. **Open Relay Exploitation**: If improperly configured, your server can be used by spammers to send large volumes of email, damaging your IP reputation.
2. **Brute Force Attacks**: Attackers often attempt to compromise accounts via brute force or credential stuffing. Leveraging the reputation of the email server, they can send large volumes of email.
3. **Spam and Phishing**: Attackers may spoof your domain or use your server for phishing campaigns.
4. **Data Breaches**: Poorly secured servers can expose sensitive emails, proprietary data, and critical business information.
5. **Malware Delivery**: Your server could inadvertently become a vehicle for spreading malware if attachments are not scanned.

Setup Postfix

This guide is inspired by DigitalOcean's How to Install and Configure Postfix, feel free to read more about setup and configuration options.

Step 1

Perform an update on local apt package cache:

```
sudo apt update
```

Install the postfix package, passing the environmental variable allows the installation wizard to prompt for some additional options.

```
sudo DEBIAN_PRIORITY=low apt install postfix
```

The installation process will raise several interactive prompts. Use the following settings below:

- **General type of mail configuration?:** Internet Site.
- **System mail name:** email-svr.
- **Root and postmaster mail recipient:** email-svr
- **Other destinations to accept mail for:** Default settings.
- **Force synchronous updates on mail queue:** No.
- **Local Networks:** Default.
- **Mailbox size limit:** 0 (disables any size restriction).
- **Local address extension character:** +.
- **Internet protocols to use:** All.

When prompted to restart services, accept the defaults, choose “OK”.

Step 2

Many of Postfix’s configuration settings are defined in `/etc/postfix/main.cf`. Postfix provides the `postconf` command to query and set configuration settings for this file.

Set the location for the Ubuntu user’s mailbox.

```
sudo postconf -e ‘home_mailbox= Maildir/’
```

Set the location of the virtual_alias_maps table, which maps arbitrary email accounts to Linux system accounts.

```
sudo postconf -e 'virtual_alias_maps= hash:/etc/postfix/virtual'
```

Next, create the virtual file, then we can begin mapping email accounts to user accounts to Linux system.

```
sudo nano /etc/postfix/virtual
```

Enter any email address we would like accept:

```
email-svr@corp.project-x-dc.com email-svr
```

👉 Here we are routing any email that comes from the email-svr address to local email-svr account.

Save and close with CTRL+X, Y, then ENTER.

Apply the mapping:

```
sudo postmap /etc/postfix/virtual

sudo systemctl restart postfix
```

Allow connections to the service Postfix with UFW:

```
sudo ufw allow Postfix
```

Postfix is configured and ready to accept external connections. We need to make a few changes to Ubuntu server's set up before sending email.

Step 3

To interact with mail being delivered, we will use the s-nail package. s-nail will look for a variable called MAIL to find mail for your user. Let's ensure the MAIL variable is set regardless of how the account is accessed:

```
echo 'export MAIL=~/Maildir' | sudo tee -a /etc/bash.bashrc |
sudo tee -a /etc/profile.d/mail.sh
```

Supply variable into the current session with:

```
source /etc/profile.d/mail.sh
```

Install s-nail email client:

```
sudo apt install s-nail
```

Adjust a few settings in the `/etc/s-nail.rc` at the bottom of the file:

```
sudo nano /etc/s-nail.rc

set emptystart

set folder=Maildir

set record=+sent
```

Save and close the file.

Step 4

Change the hostname to smtp.corp.project-x-dc.com in /etc/hostname.

```
sudo nano /etc/hostname
```

Delete all contents in the file, then add:

```
smtp.corp.project-x-dc.com
```

Save and close with CTRL+X, Y, then ENTER.

Change IP address to a static IP address. Create a new file with a priority of 01 (highest priority).

```
sudo nano /etc/netplan/01-netcfg.yaml
```

Add the following statements (note the white spacing and indentation).

```
network:
    ethernets:
            enp0s3:
                dhcp4: false
                addresses:
                  - 10.0.0.8/24
                gateway4: 10.0.0.1
                nameservers:
                addresses:
                  - 10.0.0.5
                  - 8.8.8.8
  version: 2
```

Save and close with CTRL+X, Y, then ENTER.

Apply the changes with (you may see some errors related to the outdated gateway4 statement, this is okay):

```
sudo netplan apply
```

Reboot the machine with:

```
reboot
```

Step 5

Navigate to the /postfix/main.cf configuration file:

```
sudo nano /etc/postfix/main.cf
```

Add the following (highlighted):

```
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination

myhostname = smtp.corp.project-x-dc.com

mydomain = corp.project-x-dc.com

alias_maps = hash:/etc/aliases

alias_database = hash:/etc/aliases

mydestination = $myhostname, localhost.$mydomain, localhost

relayhost =

mynetworks = 127.0.0.0/8 10.0.0.0/24 [::ffff:127.0.0.0]/104 [::1]/128

mailbox_size_limit = 0

recipient_delimiter = +

inet_interfaces = all

inet_protocols = all

home_mailbox = Maildir/

virtual_alias_maps = hash:/etc/postfix/virtual.
```

Apply the configurations changes with:

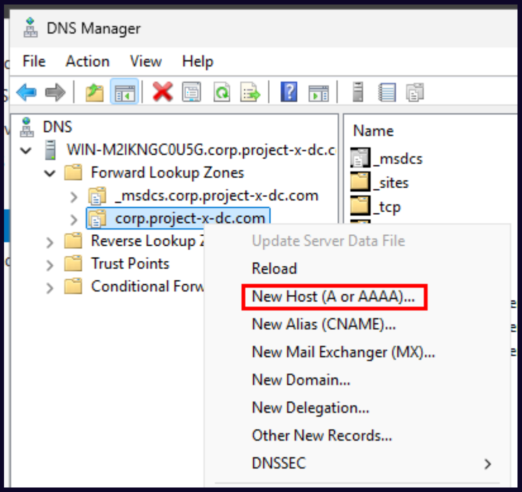
```
sudo systemctl restart postfix
```

Step 6

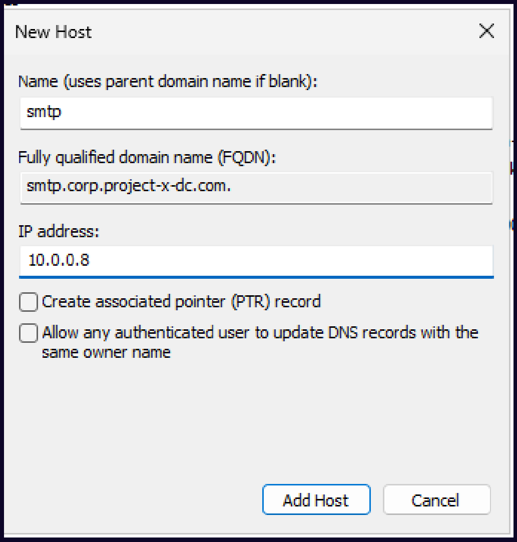
Go to `[project-x-dc]` domain controller.

Navigate to Tools → “DNS”.

Go to “Forward Lookup Zones” → Right-click “corp.project-x-dc.com” → “New Host”.



Add the following configuration. “Add Host”.



New Host

Name (uses parent domain name if blank):

smtp

Fully qualified domain name (FQDN):

smtp.corp.project-x-dc.com.

IP address:

10.0.0.8

□ Create associated pointer (PTR) record

□ Allow any authenticated user to update DNS records with the same owner name

Step 7

Go to `/home/email-svr` and create a new empty directory called `Maildir`.

```
cd /home/email-svr && mkdir Maildir
```

Go back to [project-x-email-svr], Send a test email.

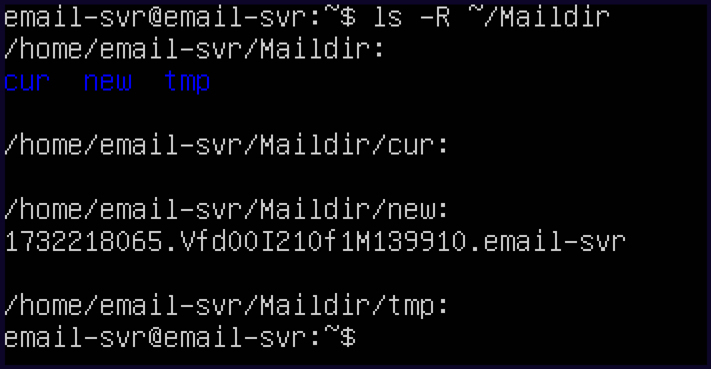
```
echo 'init' | s-nail -s 'init' -Snorecord email-svr
```

You may get a response such as `Can't canonicalize "/home/email-svr/Maildir".` This is okay.

Make sure the directory was created by looking at the `~/Maildir` directory:

```
ls -R ~/Maildir
```

A new directory structure will have been created (`cur`, `new`, `tmp`).

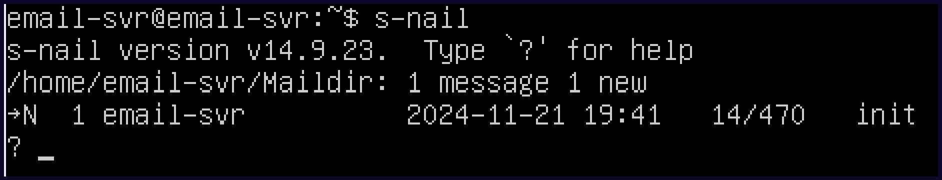


```
email-svr@email-svr:~$ ls -R ~/Maildir
/home/email-svr/Maildir:
cur  new  tmp

/home/email-svr/Maildir/cur:

/home/email-svr/Maildir/new:
1732218065.Vfd00I210f1M139910.email-svr

/home/email-svr/Maildir/tmp:
email-svr@email-svr:~$
```

Step 8

Open the client with `s-nail` command.

You should see an inbox with the init message.



```
email-svr@email-svr:~$ s-nail
s-nail version v14.9.23.  Type `?' for help
/home/email-svr/Maildir: 1 message 1 new
→N  1 email-svr          2024-11-21 19:41   14/470    init
? _
```

Press “Enter” to display the message.

Get back to the message list with:

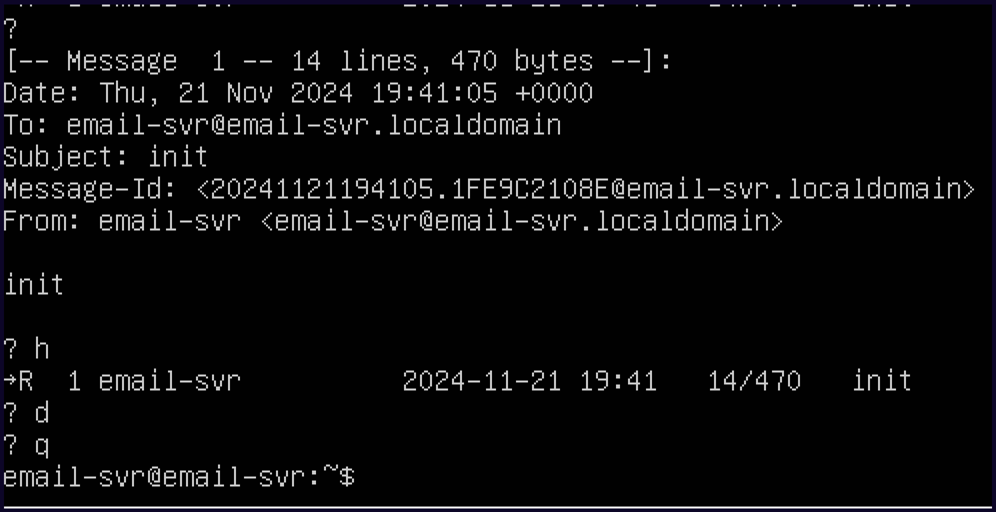
```
        ? h
```

Delete the message:

```
        ? d
```

Get back to the terminal:

```
        ? q
```



```
?
[-- Message  1 -- 14 lines, 470 bytes --]:
Date: Thu, 21 Nov 2024 19:41:05 +0000
To: email-svr@email-svr.localdomain
Subject: init
Message-Id: <20241121194105.1FE9C2108E@email-svr.localdomain>
From: email-svr <email-svr@email-svr.localdomain>

init

? h
→R  1 email-svr          2024-11-21 19:41   14/470   init
? d
? q
email-svr@email-svr:~$
```

Let's test whether s-nail can send email messages. Let's first create a test message in a test_message file.

```
nano ~/test_message
```

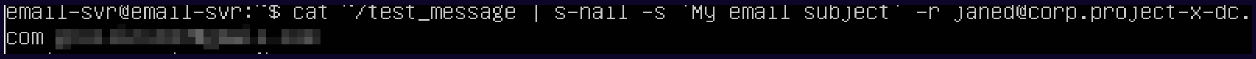
…

```
Hello! This is a test!
```

Save and close the file. We can then use the cat command, followed by s-nail process to send the file:

```
cat ~/test_message | s-nail -s 'Test email subject line' -r janed@corp.projext-x-dc.com email-svr@email-svr
```

The `-s` option supplies the Email Subject. The -r allows you to modify the "from" field.

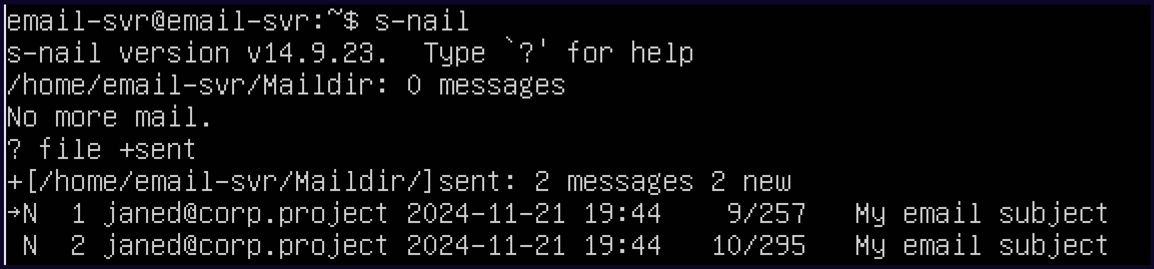


```
email-svr@email-svr:~$ cat ~/test_message | s-nail -s 'My email subject' -r janed@corp.project-x-dc.
com [redacted]
```

Navigate to the inbox with `s-nail`.

View your sent messages:

```
? file +sent
```



```
email-svr@email-svr:~$ s-nail
s-nail version v14.9.23.  Type `?' for help
/home/email-svr/Maildir: 0 messages
No more mail.
? file +sent
+[/home/email-svr/Maildir/]sent: 2 messages 2 new
→N  1 janed@corp.project 2024-11-21 19:44    9/257   My email subject
 N  2 janed@corp.project 2024-11-21 19:44   10/295   My email subject
```

Success! You can now send and receive mail on email-svr.