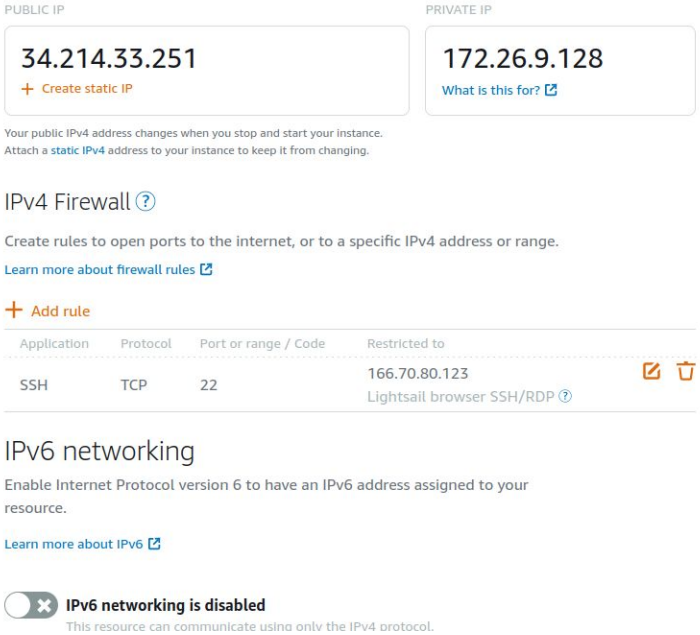
Configure - Lab

October 2022

PurpleVPN - Lightsail Configuration

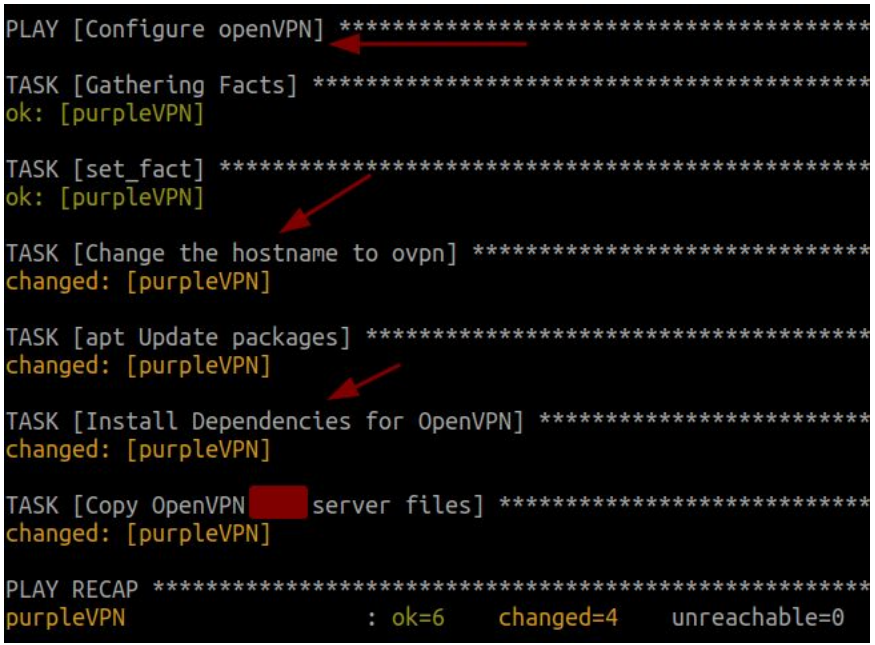


I have an instance for the PurpleVPN and it shows the Public and Private IP Addresses

I have set up the restriction to only allow SSH from my IP Address

Disabled IPv6 Networking

PurpleVPN - Ansible Execution



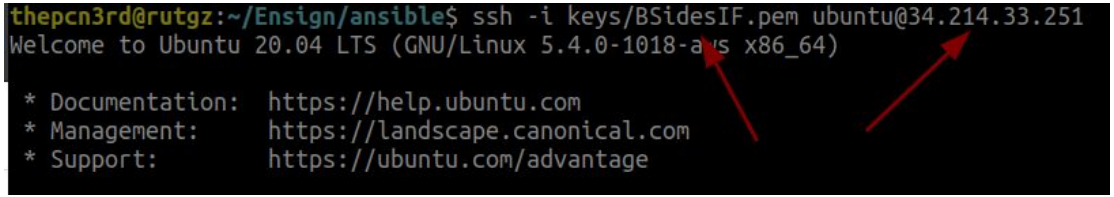
This is what it looks like for the execution of the playbook of configPurpleVPN.yml

Changes the hostname

Installs the package dependencies

Copies the openvpn-install.sh script to /home/ubuntu

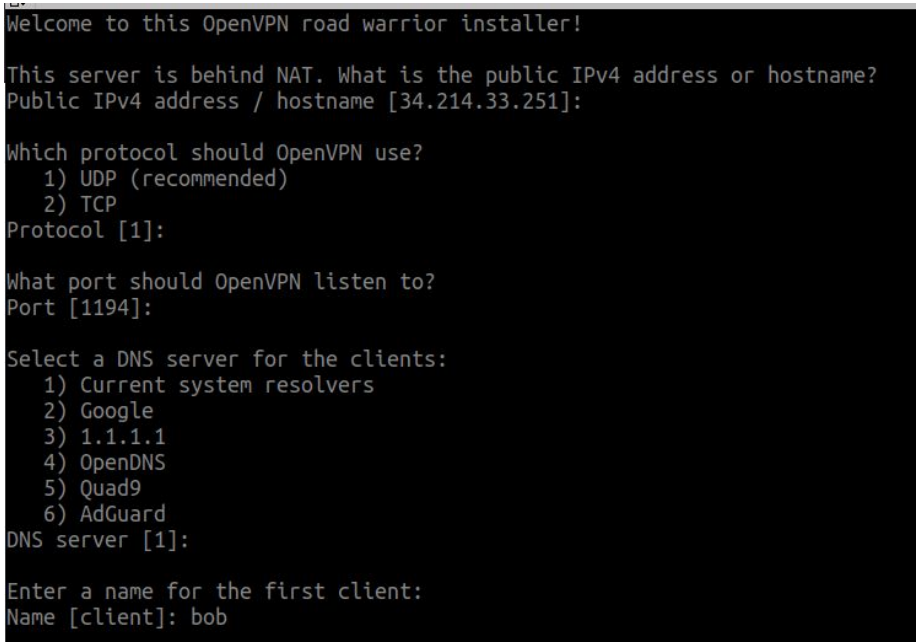
PurpleVPN - Install OpenVPN



Login with your SSH Key and IP Address for the PurpleVPN Server

Execute with sudo the openvpn-install.sh

PurpleVPN - Configure OpenVPN



Should populate your external IP of the LightSail Server

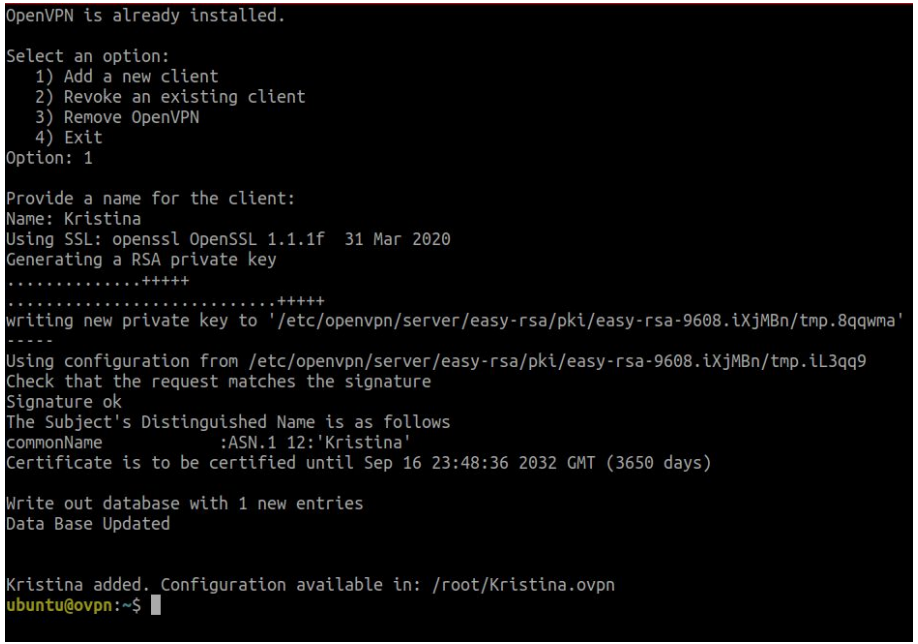
UDP is Recommended

Port 1194 - Needs to be setup in LightSail firewall for access

Choose the DNS servers your clients will utilize

Each client needs a uniquely created ovpn file (You cannot reuse)

PurpleVPN - Configure OpenVPN



Rerun the installer for openvpn then select “Add a new client”

Each additional ovpn file is created under /root/<name>.ovpn

This file is provided to the person connecting to your OVPN Server remotely.

Note anyone connected to the OVPN server can see other clients connected.

PurpleVPN - Why OpenVPN?

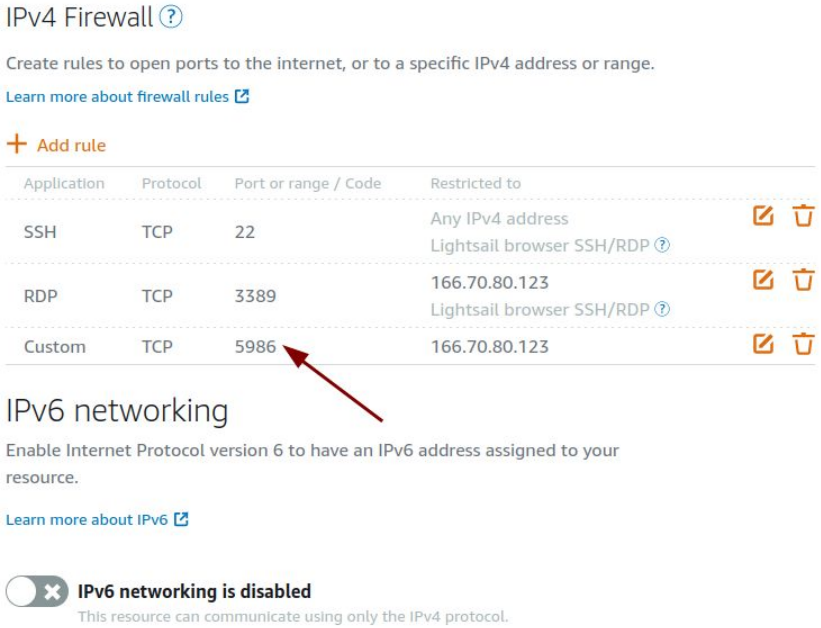
Why setup an OpenVPN Server?

- Allows a connected client to connect to ALL of your private IP Addresses on Lightsail
- Access to insecure services to be hosted on the private IP Addresses
- Usage of forward connecting tools like crackmapexec, evil-winrm and many others work
- Webshells, shells and other connections work
- Simulates an internal compromised device
- Works great for a student environment as long as everyone plays nice!

Reverse Shells

- The openvpn and routing is not setup for reverse shells, however in the lab we will use SSH and work around the limitation

PurpleDC and PurpleMBR - Lightsail Configuration



Setup the firewall to restrict based on IP Address, for whatever reason it does not let you filter by IP on SSH (Bug?!?)

Allow Port TCP/5986 and restrict by IP Address so ansible can authenticate OR connect to the OVPN server and use the private IP Address

purpleDC and purpleMBR

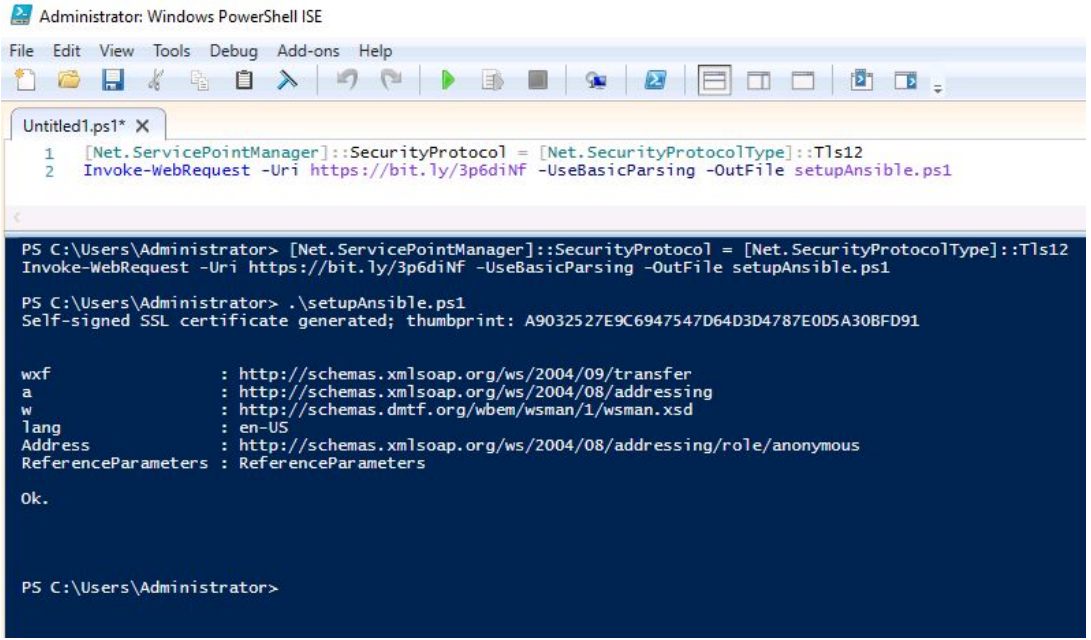
```
vars:
  ansible_python_interpreter: "/usr/bin/python3"
  ansibleDirectory: "/home/thepcn3rd/Ensign/ansible"
  domainName: "13lives.local"
  netbiosName: "13lives"
  # Needs to be updated to the internal IP of the Domain Controller
  dnsInternalServer: "172.26.6.32"
  # Needs to be updated to the administrator password of the domain controller
  domainPass: ""
  commonUser: "thepcn3rd"
  commonPass: "#Blast2022New!Year"
  lightsailUser: "ubuntu"
  lightsailPem: "/home/thepcn3rd/Ensign/ansible/keys/BSidesIF.pem"
  className: "BSides IF 2022"
```

Update the ansible_password for the respective host in the inventory.yml

Update the domainPass in the inventory.yml to be the administrator password for purpleDC

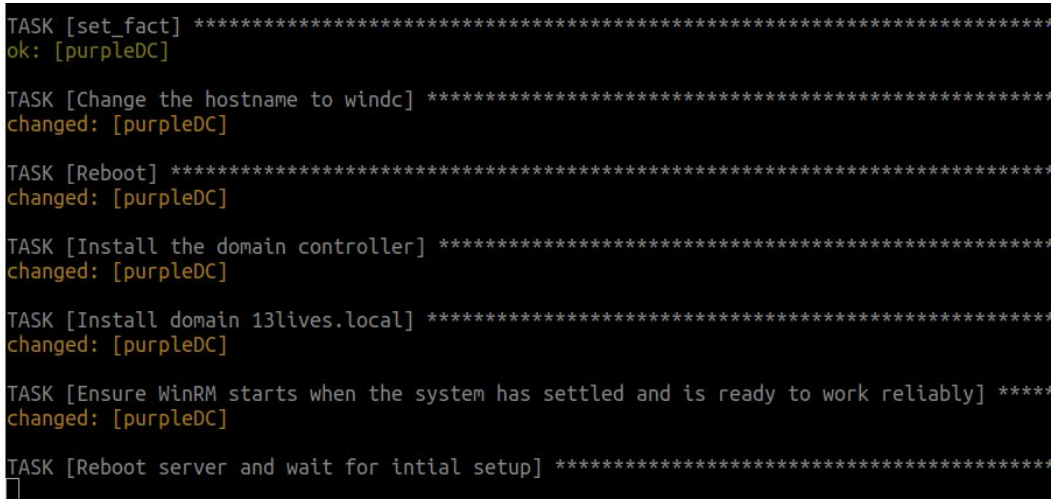
Update the dnsInternalServer to the private IP Address of the purpleDC

purpleDC and purpleMBR - Setup Ansible



Remember to setup ansible on these servers. I have provided a bit.ly link for convenience.

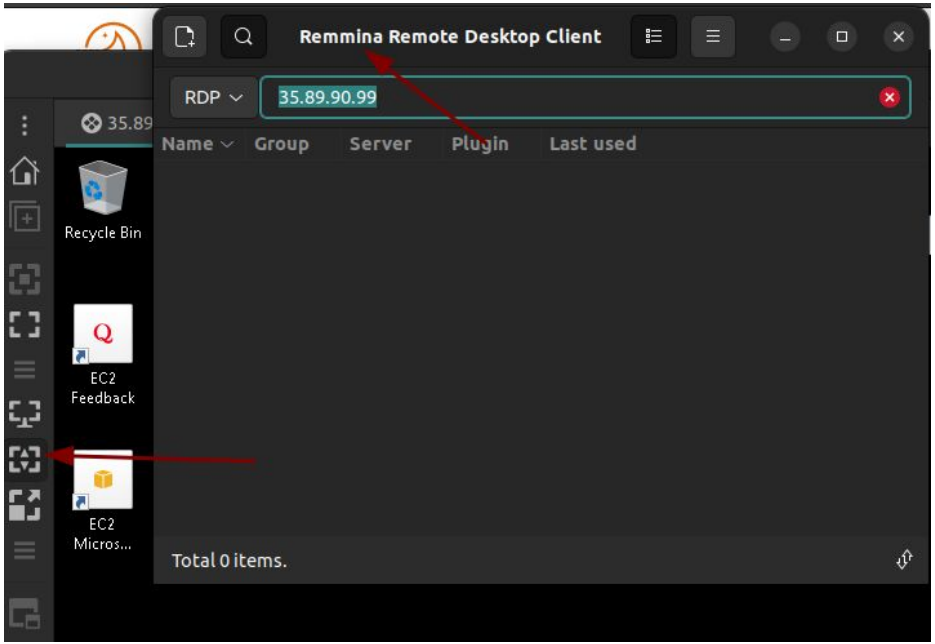
purpleDC and purpleMbr - ExecuteConfig



While running ansible if ansible is setup, the port in Lightsail is configured for TCP/5986, passwords updated in the inventory.yml file no issues should occur

Rerun the executeConfig script and comment out the playbooks of systems that worked in the configure.yml

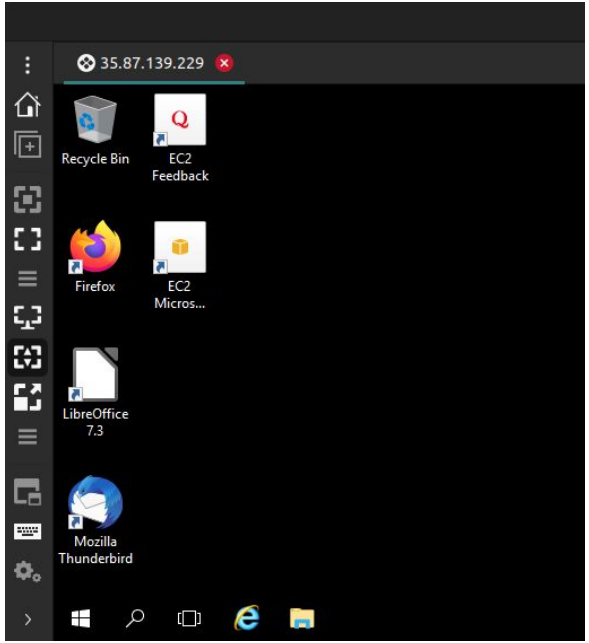
purpleDC - Remote Desktop



After installing the domain, I have had inconsistent connections using the “Connect to Remote Desktop” inside of Lightsail. I will use a tool called Remmina on Ubuntu to connect.

The second arrow points to dynamically adjust the resolution based on the size of the window. I like this setting, sometimes it distorts the view

purpleMbr - Build Vulnerable User Profile



After the installation of PurpleMBR is complete, setup is required for the scenario to work. Login as prayut.c and setup his profile. (Information is in playbooks/configPurpleDC.yml)

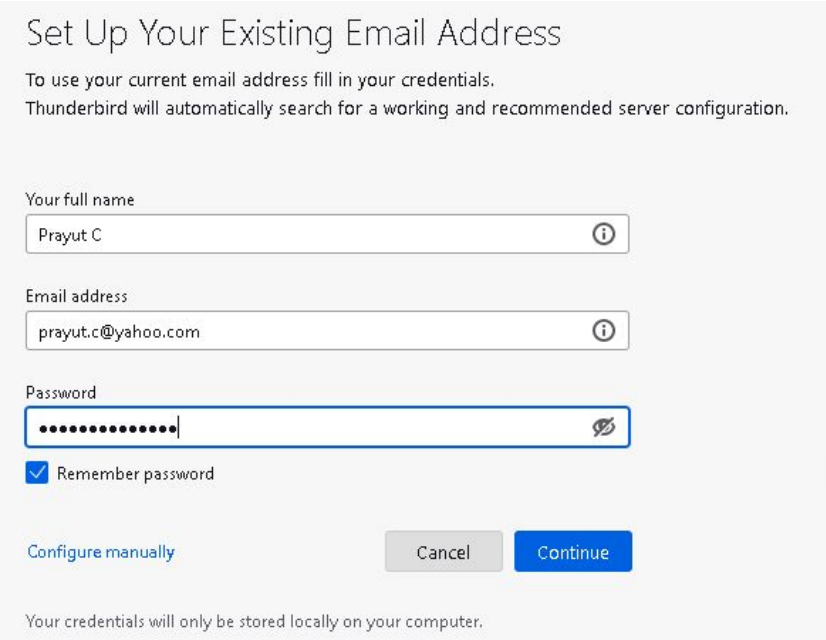
Login as prayut.c

Password: ExpRRQvwn24SJv<xxxx>

This builds the profile for the user that needs some configuration

Observe Firefox, Thunderbird, LibreOffice and the JDK should be installed. We need to configure them

purpleMbr - Create Yahoo Account and Setup Thunderbird



Setup Thunderbird with a yahoo account that you create (Personal Opinion: Yahoo is lacking controls that end-users need in-place)

Configure the mailbox as IMAP

Authenticate

Set as Default

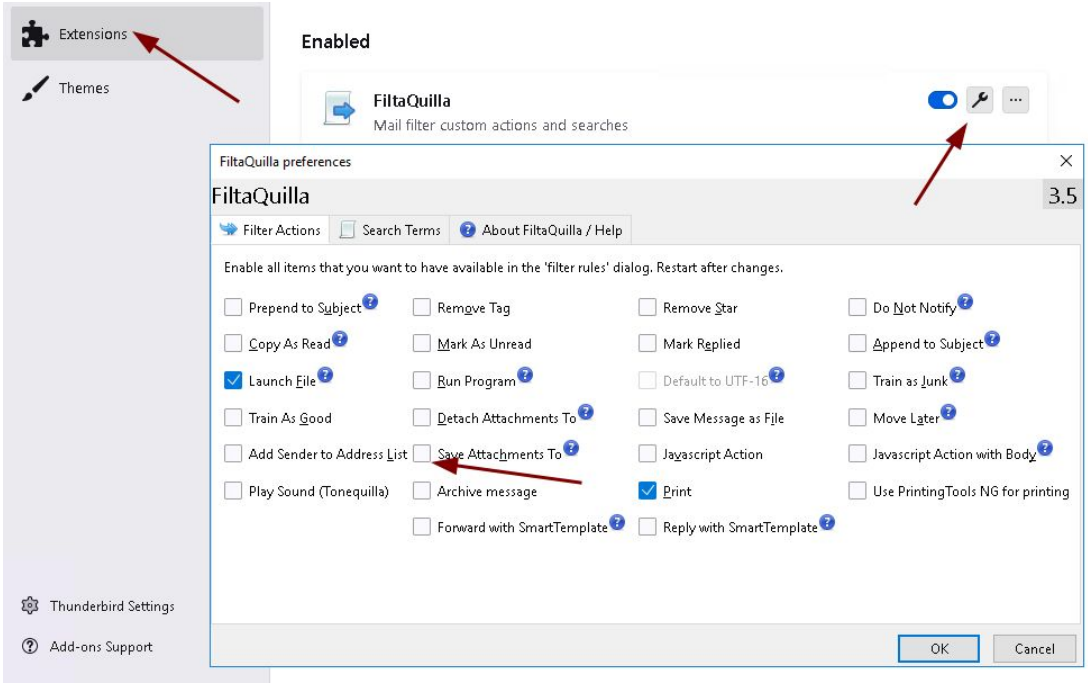
You should see a couple of emails in your inbox from spam

purpleMbr - Install Add-on FiltaQuilla

Top-right of thunderbird click the options menu and Add-ons and Themes

Search for and “Add to Thunderbird” FiltaQuilla

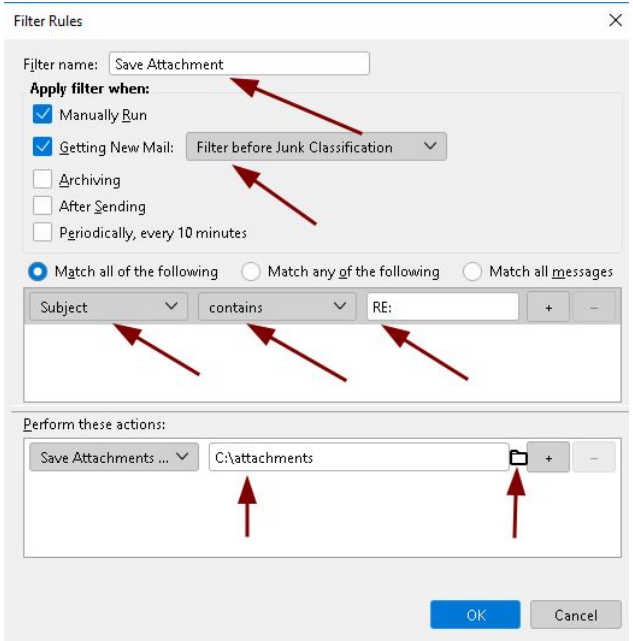
purpleMbr - Configure FiltaQuilla



After installation click on extensions, then the configure wrench icon and select Save Attachments To.

Then go back to the options menu, select Tools and Message Filters

purpleMbr - Setup Message Filter



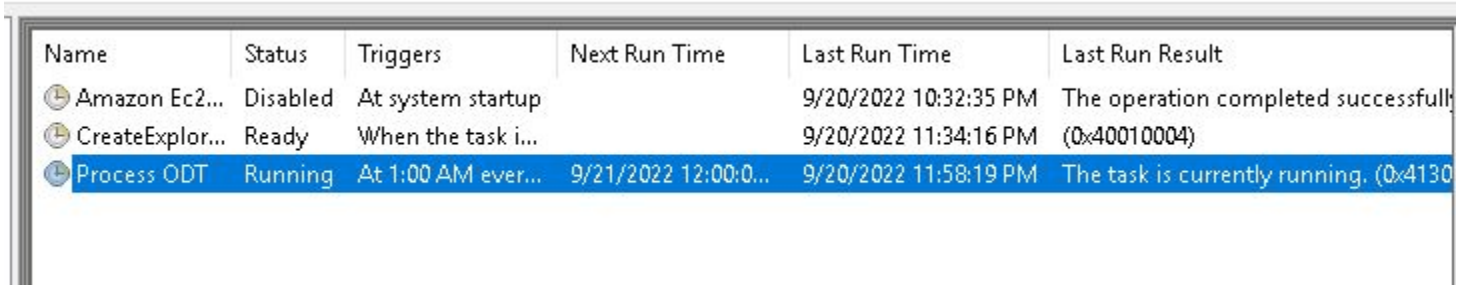
Name the filter you are creating

I setup the Subject contains a keyword like “RE:” so I can control which emails this executes against

Then configure the attachments of a received email to save to c:\attachments

The c:\attachments folder is where the powershell script triggered by a scheduled task will run the macros that exist inside of an odt file.

purpleMbr - Check Scheduled Task



Name	Status	Triggers	Next Run Time	Last Run Time	Last Run Result
Amazon Ec2...	Disabled	At system startup		9/20/2022 10:32:35 PM	The operation completed successfull
CreateExplor...	Ready	When the task i...		9/20/2022 11:34:16 PM	(0x40010004)
Process ODT	Running	At 1:00 AM ever...	9/21/2022 12:00:0...	9/20/2022 11:58:19 PM	The task is currently running. (0x4130

Verify the scheduled task of “Process ODT” is executing. You may have to stop and restart the task. It should say in the “Last Run Result” the task is currently running.

Launch LibreOffice to create some necessary files

purpleMbr - pwshProcessODT.ps1

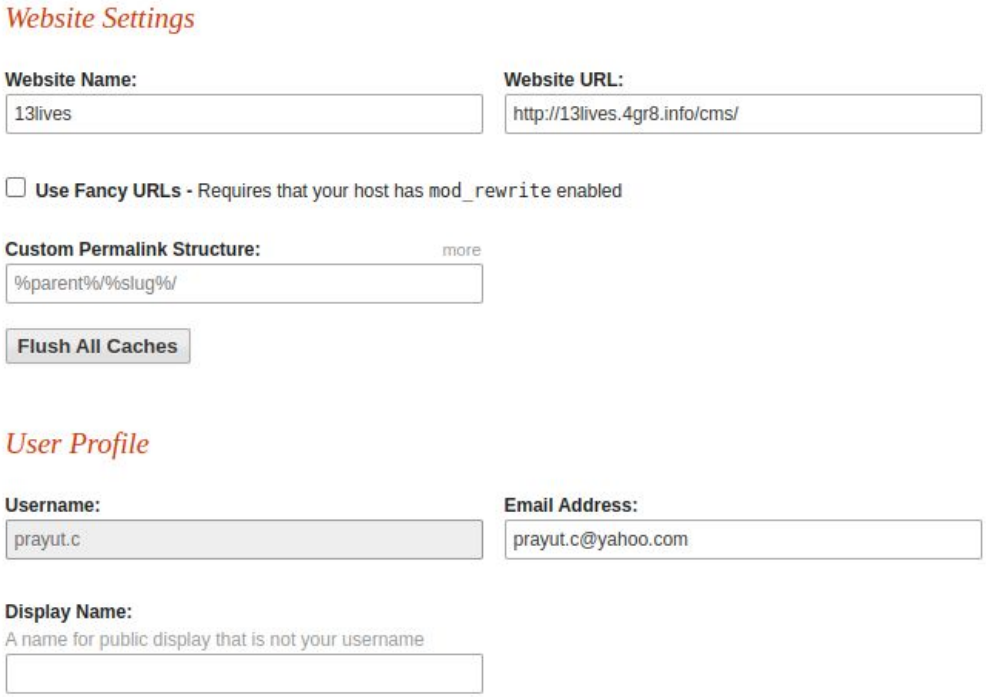
pwshProcessODT.ps1

```
$process_dir = "C:\attachments"

while($true) {

        # If any odt files in attachments, load the file and then archive:
        $files = ls $process_dir\*.odt
        if ( $files.length -gt 0) {
        # Copy the registrymodifications.xcu file back in-place to remove the document recovery
        Copy-Item -Force -Path "C:\files\registrymodifications.xcu" -Destination "C:\users\administrator\appdata\roaming\libreoffice\4\
        Start-Sleep -s 15

                # launch odt files
                Invoke-Item "$($process_dir)\*.odt"
                Start-Sleep -s 60

                # kill libre office, sleep
                Stop-Process -Name soffice*
                Start-Sleep -s 60
        Remove-Item -Recurse -force -Path $process_dir\*
        Remove-Item -Force -Path "c:\users\administrator\appdata\roaming\libreoffice\4\user\registrymodifications.xcu"
        }
        Start-Sleep -s 60
}
```

This script executes and pauses for up to 195 seconds, note that in sending in your ODT files as attachments

This also recreates the registrymodifications.xcu file, this is to reset the settings in OpenOffice for relaxed Macro security and it removes an issue of LibreOffice trying to recover the last document that crashed (Interesting Vulnerability)

purpleLin - Initial Setup



Previously setup GetSimpleCMS with a registered DNS that I own.

I setup the yahoo account for BSidesIF 2022

The website password is not the same as the email password

The password for prayut.c is provided only to simulate an administrator logging into the site for the Persistent XSS vulnerability to work

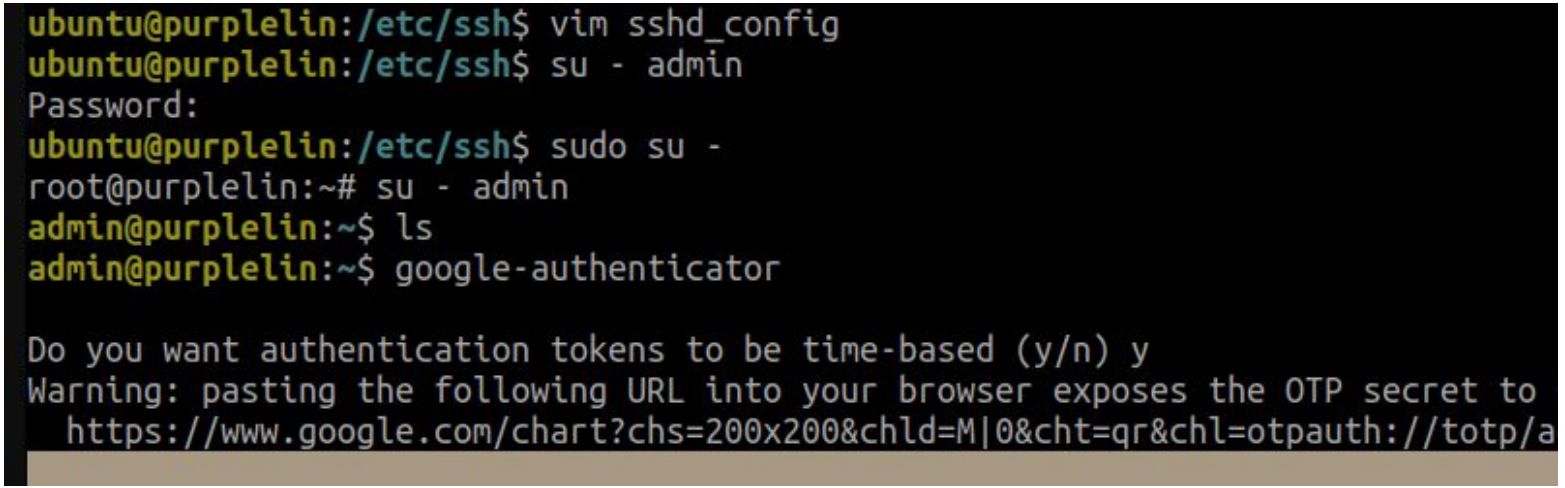
purpleLin - Google Auth Setup

Setup google authenticator following the below page:

https://www.digitalocean.com/community/tutorials/how-to-set-up-multi-factor-authentication-for-ssh-on-ubuntu-18-04

Run google-authenticator for the admin user. Only change the token for the admin user for the purposes of the Lab. The admin user is not a privileged account, this account is used for purposes of the lab.

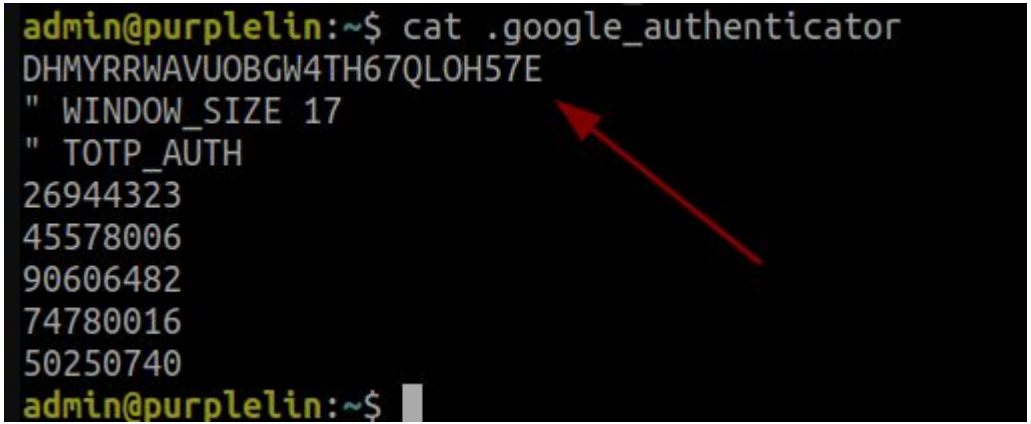
purpleLin - Google Authenticator Setup



From the SSH session of your Ubuntu user on PurpleLin, switch over to the admin account and execute “google-authenticator”

Setup the token to be time-based, allow multiple connections, and your choice on hardening from brute force.

purpleLin - Google Auth Token (Step 2)



Modify the token in /home/admin/.google-authenticator to have the same secret key as is contained in the web server file /var/www/html/floatinglogs/db/floatinglogs.db

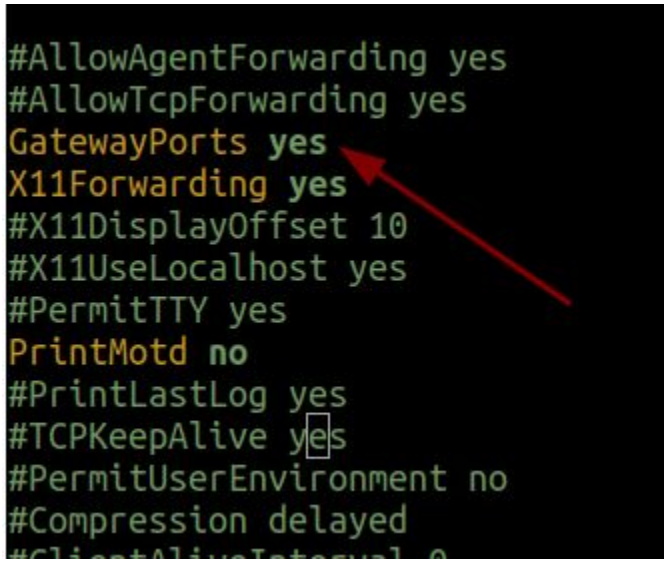
purpleLin - Setup SSH Authorized_Keys

```
admin@purplelin:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
admin@purplelin:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC0hFiWT+geyxWbkt5d+tgN55ZhKCTWY90rl7mglc5h
8ZWMwxIUEli7xL9Wk5URD+wPTcpt+CqZIOatw6iAMwZZmIx/bmT1ljowXVFWUcHLFbHyU8P5ncVbCek1
o6KzWg8R9sRT3ZNqz6cQbf1+7K6RmH0hs17e9w8Z6X2Y3KWKhMn0NVygTYCKVaRG1xgsjx0wApqAHGlz
VbTgqKM5aNl5VK63vtybnV6YUumuQ72ODDWW/9YAeyYCup3eI0egvjRU25uLttwyLWsuVFrwWmeDPuKX
HD2107zZXFTm917SJfI1VBV0ueVlK+ef53arrHEaJayDnmsWoOlY9jnwsZPb BSidesIF
admin@purplelin:~/.ssh$
```

To create the .ssh directory and appropriate files I execute “ssh-keygen” as the user (This can be done manually)

Copy or create the same authorized_keys file from /home/ubuntu/.ssh/authorized_keys for the admin user. This allows the BSidesIF SSH key to be utilized in conjunction with the google auth token.

purpleLin - Setup SSH GatewayPorts



Modify /etc/ssh/sshd_config and change GatewayPorts from being commented out and set to yes.

This is for convenience of allowing a SSH reverse tunnel to listen on the 172.16.x.x private IP Address. (Other tools could be used like socat)

This reverse tunnel is used from the phishing that we will conduct.