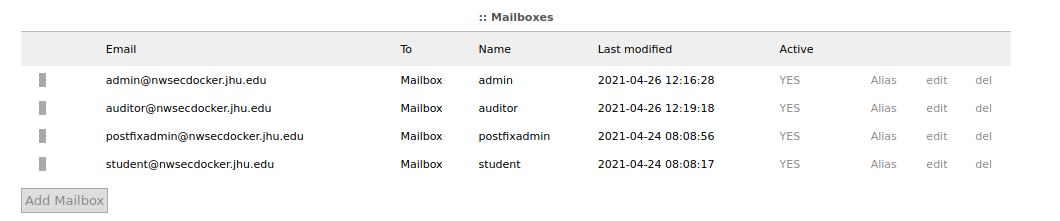
**Note**: Docker Container Used for the purpose of completing this assignment can be found at the following link: - https://github.com/jhu-information-security-institute/NwSec/tree/master/applications

**Part 1**

1. **Describe the sequence used.**

In this assignment, we have been assigned the task of configuring a Postfix Server and a Dovecot as our IMAP server.

After completing the setup instructions and configuring the postfix server, we proceed on to creating the Mailboxes.

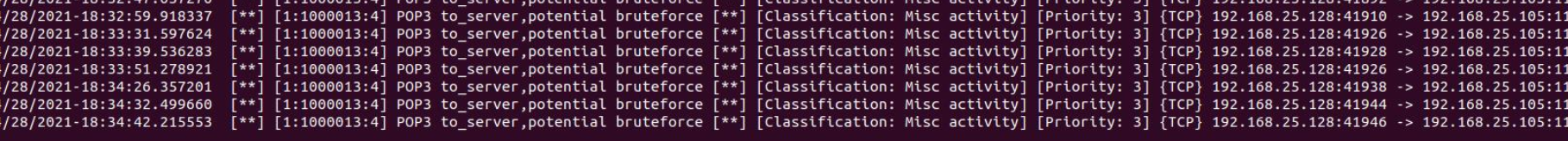


Now that we our done with our basic setup. We move onto the task of performing attacking the postfix server. We will be primarily carrying out: -

* RCPT TO attacks via ISMPT
* Open Relay Attack
* Spoofing a legitimate email
* Brute-force Dictionary attack against POP3 via hydra

After performing these attacks, we move on to improving the defences of our SMTP server. We implement SASL authentication mechanism, disable plaintext authentication, change the subnet parameters, imposing HELO Restrictions etc

Finally, we configure Suricata to detect these attacks.



1. **Provide a python script that executes your attack**

I did not create the scripts since these were just 1 liner commands. Please find the commands for executing their respective attacks below :-

1. RCPT TO attack via ISMTP: ismtp -h 192.168.25.105:25 -e email\_common\_roots.txt -l 2
2. Spoofing via ISMTP: ismtp -i infosecurity@ainsoft.com -s postfixadmin@nwsecdocker.jhu.edu -r student@nwsecdocker.jhu.edu -S postfixadmin -R student -h 192.168.25.105:25 -m
3. Bruteforce dictionary attacks against POP3 : hydra -L username.txt -P password.txt -t 6 pop3://192.168.25.105
4. **Provide \*. pcap network capture file from traffic between attack host and target running the application server.**

Provided in the zip folder.



**4. Describe the results of the attack.**

In this assignment we carried out 4 type of attacks against the Postfix Server running inside a docker container. These attacks were: -

1. RCPT TO attacks against the postfix server
2. Open Relay Attack to send some spam email
3. Spoof a valid email address
4. Brute force dictionary attack against POP3 via Nmap

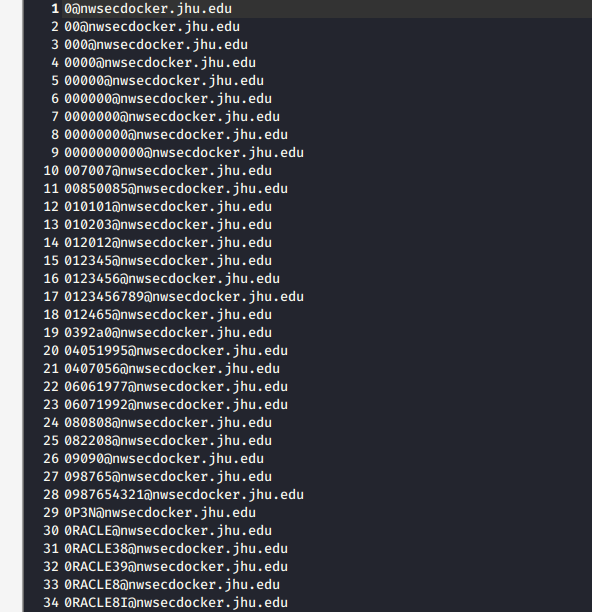
RCPT TO attacks: -   
Note: - a) I launched the first attack via ISMTP from my host VM instead of the Attack Docker Container as ISMTP and Thunderbird were not working  
b) For the rest of the attacks, I used old Attack Docker container

RCPT TO is command used to specify the recipient of the email. However, if the smtp server is not properly configured then it can be used as a social engineering attack for the purpose of enumerating Usernames.

So, in our scenario the tool we are using for enumerating the usernames is iSMTP.

With the help of professor’s python script, I generated mail dictionary.





Next, we will be running the below iSMTP command to enumerate the usernames: -

ismtp -h 192.168.25.105:25 -e email\_common\_roots.txt -l 2

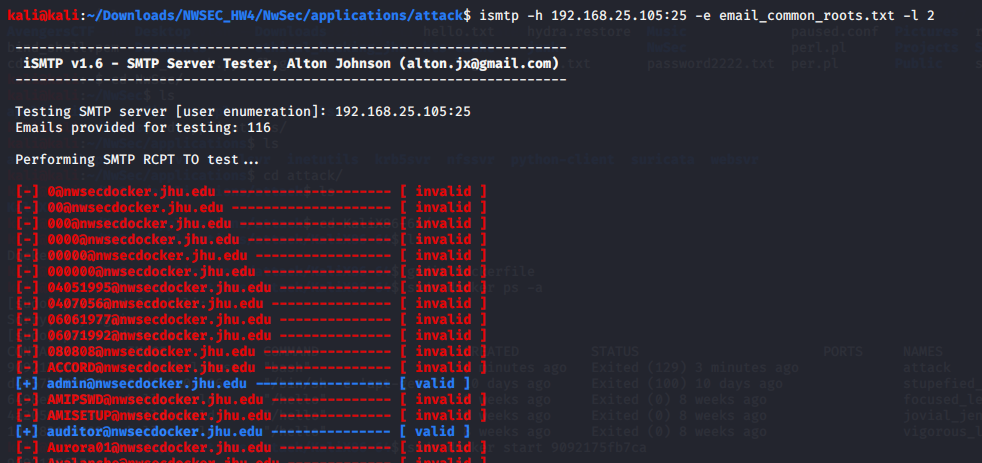
-h -> This option is used to specify the target host and port

-e -> This option enables SMTP enumeration and imports email list.

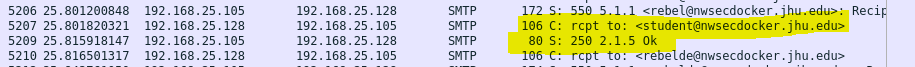
-l -> This option enables enumeration type and in this case we are using option 2 to carry out user enumeration via RCPT TO.



As we can infer from the below image, ismtp has successfully enumerated the email ids.



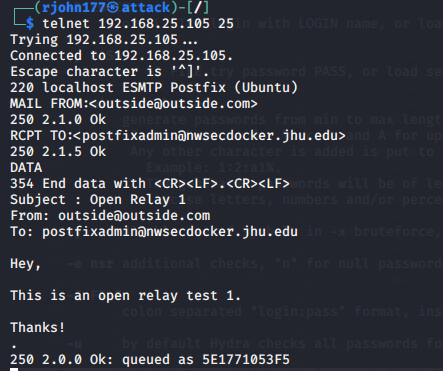
And please find the screenshot of packets capture via wireshark as well.

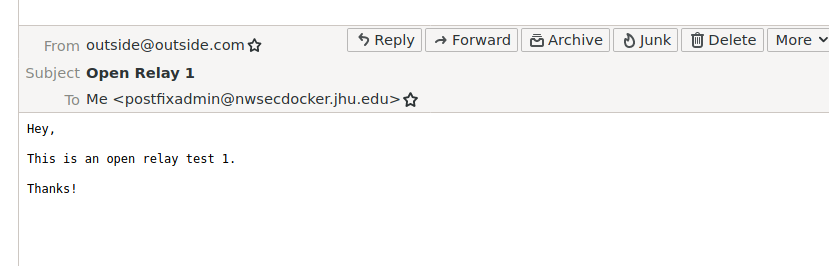


1. Open Relay Attack to send some spam email

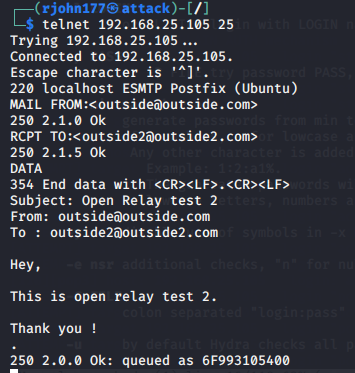
For open relay attack, I will demonstrate 3 scenarios: -

1. sent an email from an outside non-existing email ID to a known local email address

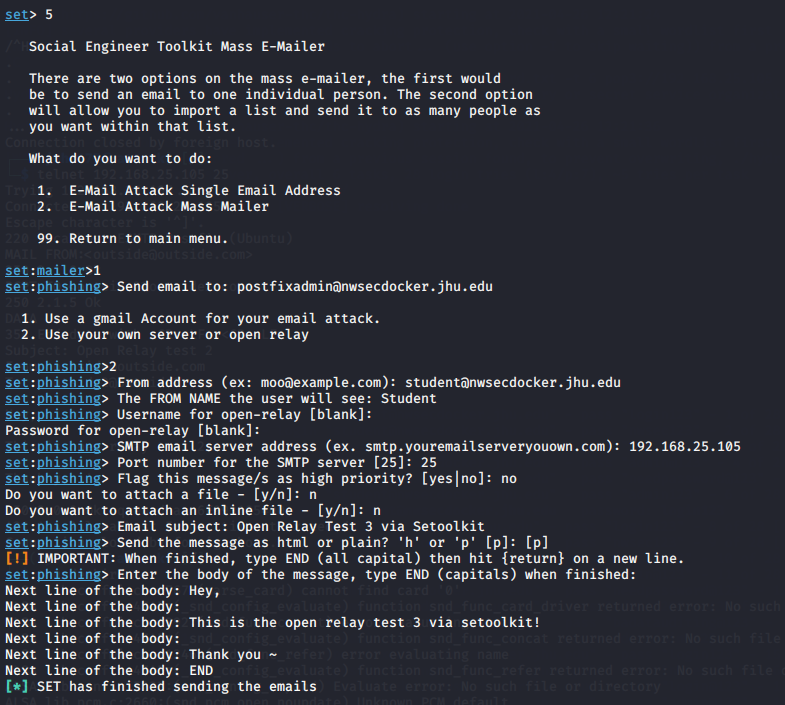


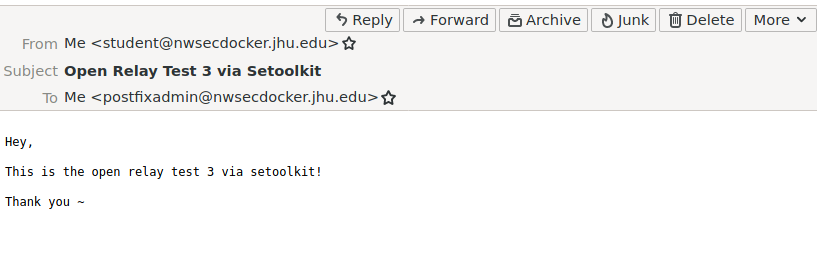


1. sent an email from a non-existing mail id to another non-existing mail id



1. Set an email from local email ID to another local email ID via setoolkit

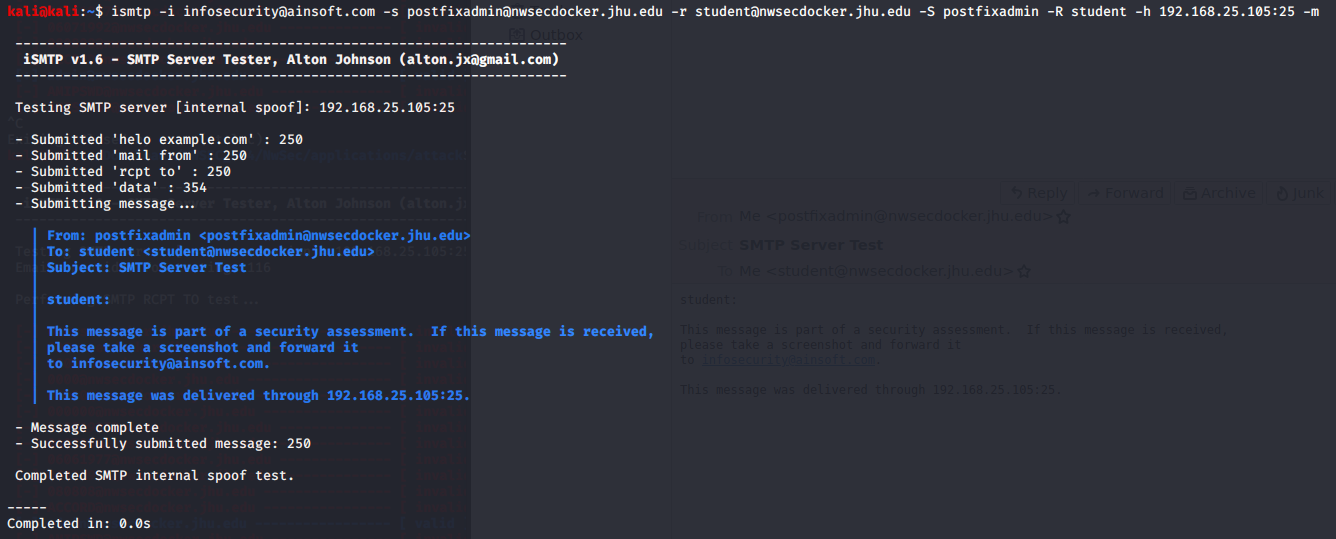




As we can see, if we don’t properly configure our mail server then it will accept email from any sender and deliver it to any recipient. As a result, it can be easily exploited by spammers and malicious users.

1. Spoof a valid email address

For spoofing a valid email address, I used ismtp tool.



The command used in this case was: -

ismtp -i infosecurity@ainsoft.com -s postfixadmin@nwsecdocker.jhu.edu -r student@nwsecdocker.jhu.edu -S postfixadmin -R student -h 192.168.25.105:25 -m

-i -> Consultant’s email address

-s -> sender’s email address

-r -> receiver’s email address

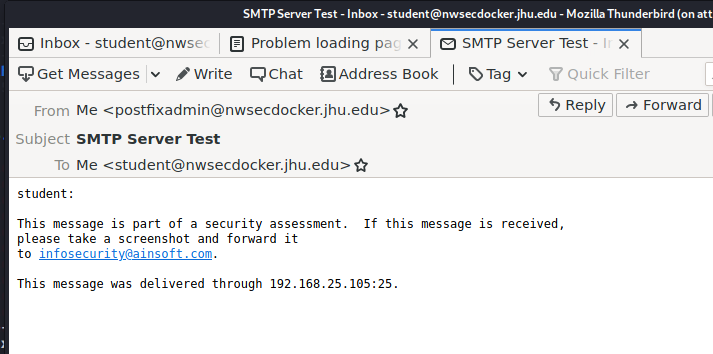
-S -> Sender’s name

-R -> Receiver’s name

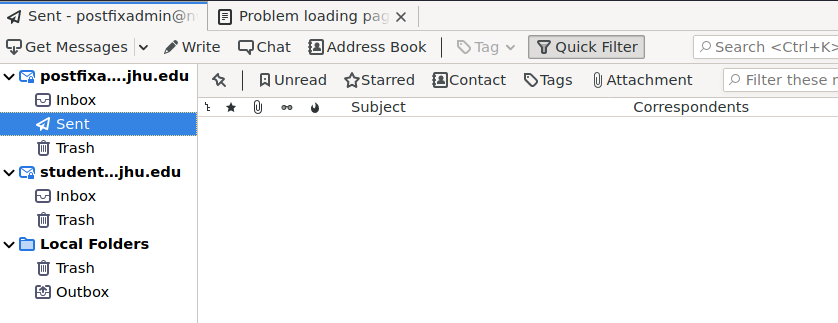
-h -> Target host

-m -> Enables SMTP spoof

On the victim’s (receiver) mailbox, we can see successful delivery of the email on behalf of spoofed email which in this case is [postfixadmin@nwsecdocker.jhu.edu](mailto:postfixadmin@nwsecdocker.jhu.edu)



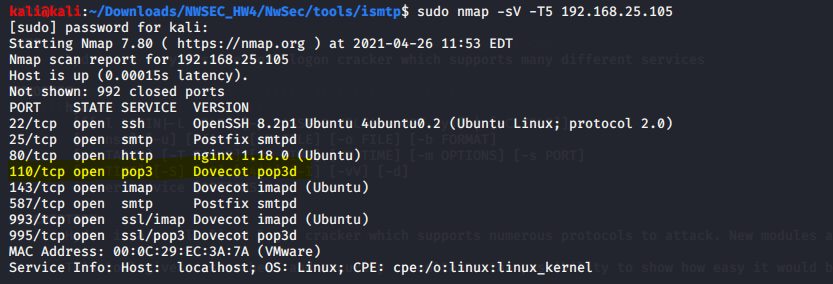
However, as we look into the sent box in postfixadmin mail box, we find that no mail was sent by [postfixadmin@nwsecdocker.jhu.edu](mailto:postfixadmin@nwsecdocker.jhu.edu)



Thus, we have successfully spoofed a valid email address.

1. Brute force dictionary attack against POP3 via Nmap

Note: I used hydra for launching the brute force attack in this case.

Ran nmap against the target ip to find which ports were open and what services were running on them. 

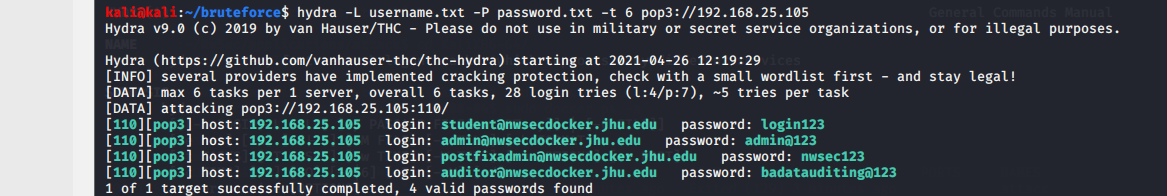
We will be bruteforce the passwords of usernames identified from the previous attacks against pop3.

I created two .txt files containing usernames and password respectively.



Next used hydra to launch the bruteforce dictionary attack and was successful at guessing the username password combination at the culmination of the attack.

hydra -L username.txt -P password.txt -t 6 pop3://192.168.25.105



This attack highlights the importance of complex passwords, and also what could happen if there is no security policy for multiple-attempts for authentication failure.

**Part 2**

1. **Provide the details for updated Postfix, other application, and OS settings.**

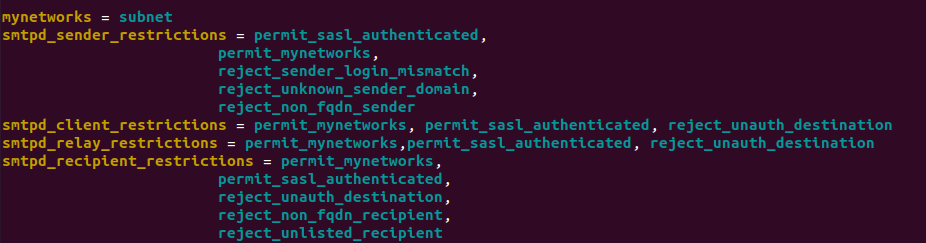
* I made changes to the following and also configured Spamassassin
  + /etc/postfix/main.cf
  + /etc/postfix/master.cf
  + /etc/dovecot/conf.d/10-master.conf
  + /etc/dovecot/conf.d/10-auth.conf

in order to prevent open relay attack, RCPT TO attacks, Spoofing attack and brute-force dictionary attack.

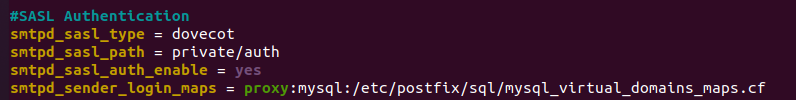
**/etc/postfix/main.cf**

I made changes to “mynetworks” parameter to reject emails from outside domains; implemented restrictions on smpt clients, senders, recipients and relays by assigning values to below parameters in order to reject authorized access.

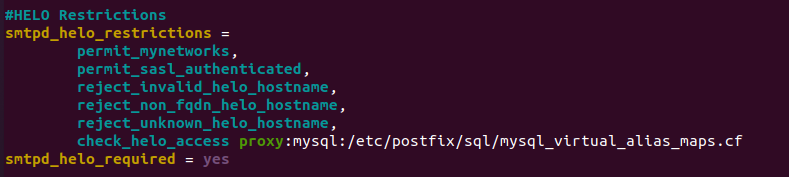
Below are the configuration changes I made: -



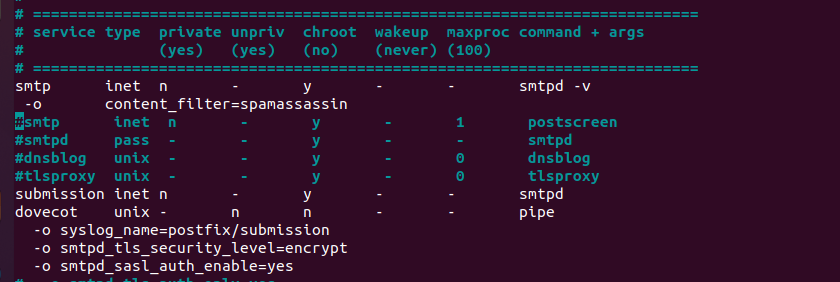
I also Implemented SASL authentication mechanism



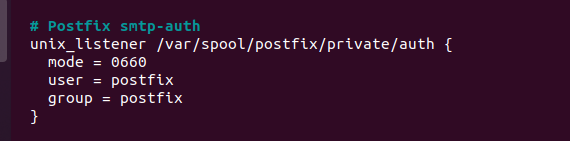
As well as, HELO Restrictions



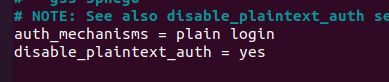
**/etc/postfix/master.cf**



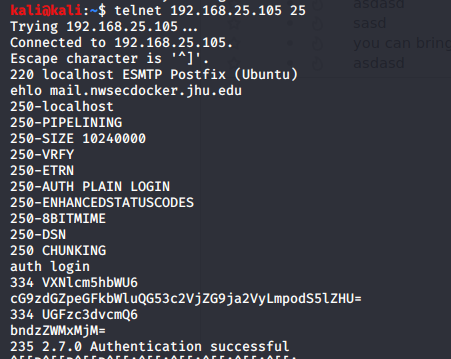
**/etc/dovecot/conf.d/10-master.conf**



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



Now if try to telnet, we will have to authenticate ourselves.



**Spamassasssin Configuration**: -

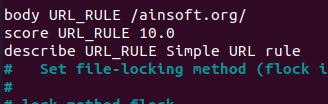
1) To install Spamassassin we use the below command: - apt-get install spamassassin spamc

2) Added SpamAssassin System User: - adduser spamd –disabled-login

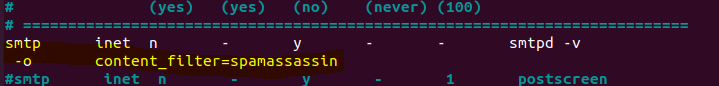
3) Edited the default configuration file /etc/default/spamassassin and made the following changes:- a) ENABLED = 1 b) SAHOME="/var/log/spamassassin/" c) OPTIONS="--create-prefs --max-children 5 --username spamd --helper-home-dir /home/spamd/ -s /home/spamd/spamd.log" d) CRON =1

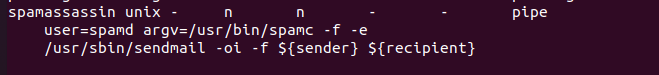
4) Adding SpamAssassin Spam Rules by editing /etc/spamassassin/local.cf

1. rewrite\_header Subject [\*\*\*\*\* SPAM \_SCORE\_ \*\*\*\*\*]
2. required\_score 5.0
3. use\_bayes 1
4. bayes\_auto\_learn 1
5. body URL\_RULE /ainsoft.org/
6. score URL\_RULE 10.0
7. describe URL\_RULE Simple URL rule



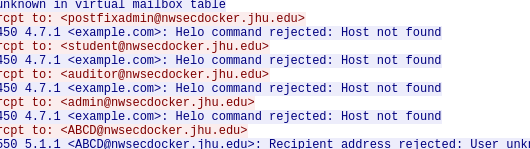
5) Next, we edit the /etc/postfix/master.cf files





1. **Provide \*. pcap network capture file from traffic between attack host and target running the application server**

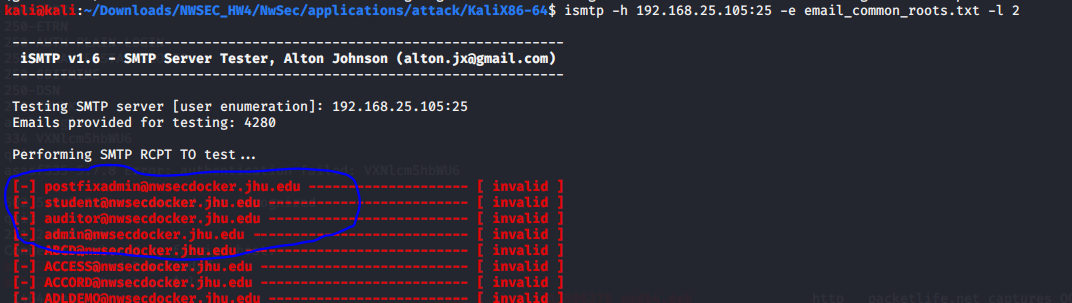
Provided in Zip folder



1. **Describe the results of the attack.**
2. Results of RCPT TO attacks

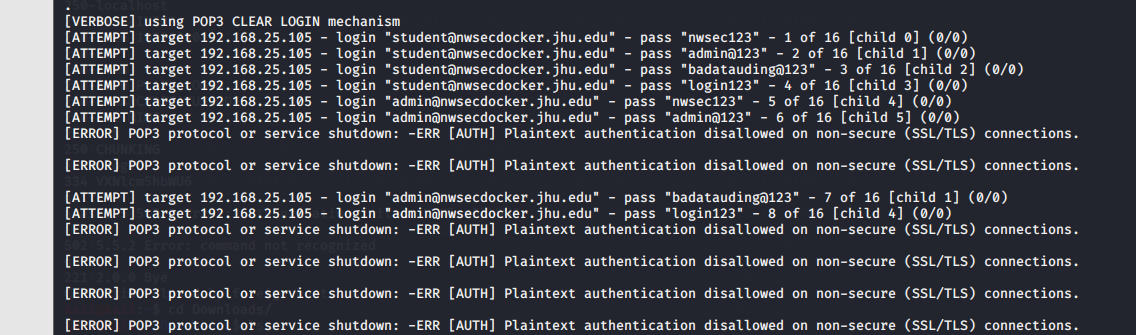
For the ease of showing the results, I added legitimate emails at the start of .txt file.

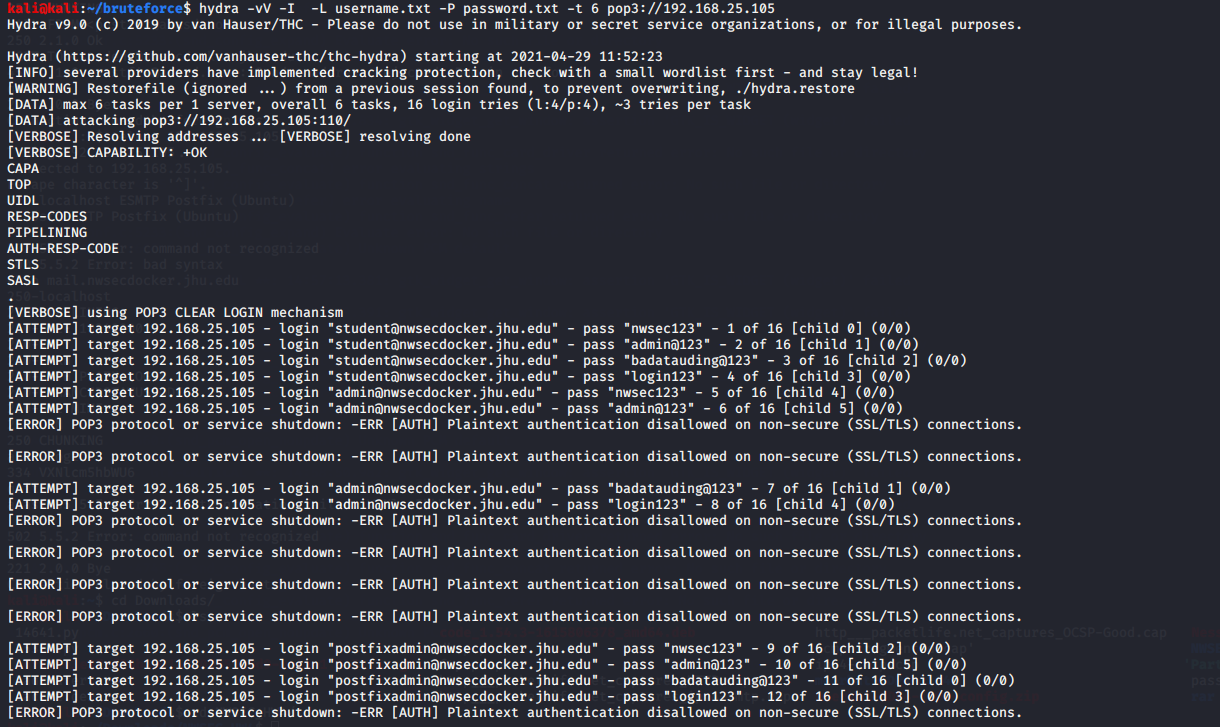
As you can see below, ISMTP was unsuccessful in RCPT TO attacks.



1. Brute Force Attack against POP3 via hydra

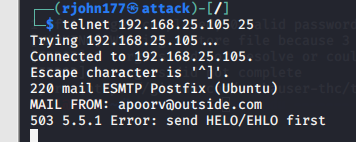
Prevented Brute Force Logins by disabling plaintext logins.





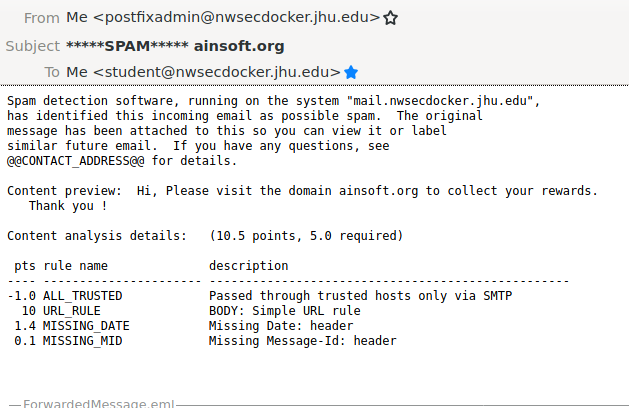
1. Open Relay Attacks

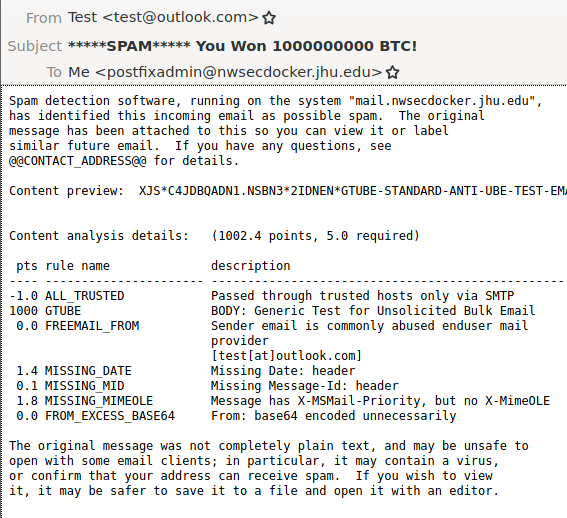
AS we can in below screenshot, we are open to prevent Open Relays through our SASL Authentication, HELO Restrictions in place, and as well other restrictions related to Sender, Recipient, clients which I have mentioned in previous question etc



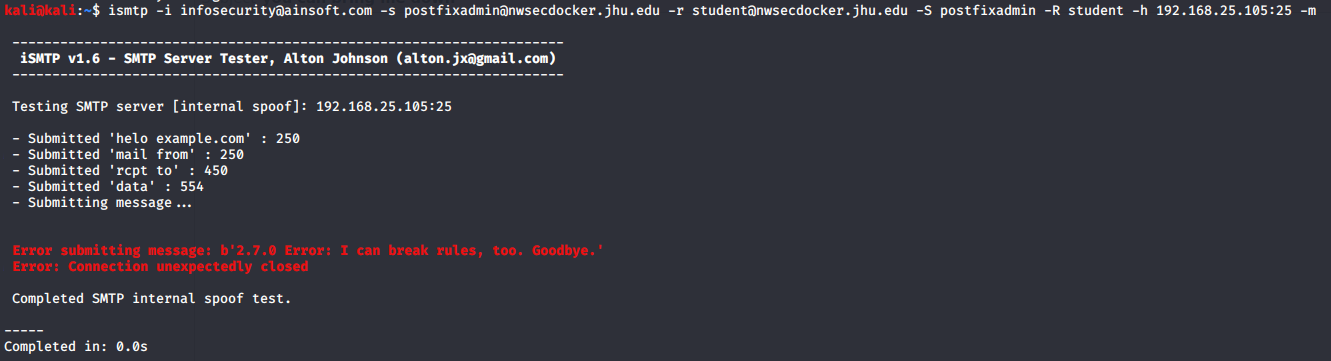


1. Spams & Spoofing attacks

Spamassassin marks the email “spam” for the rules being configured such as marking an email spam for particular URL (In this case ainsoft.org). Please find the screenshots below: - 



ISMTP fails to spoof due to our mechanisms in place as well.



**Part 3**

1. **Provide the details used to install and configure Suricata so that it could be replicated by someone else**

For this assignment, I made sure SMTP related rules were enabled such as :-





Next, I configured the rules to detect the RCPT TO attacks and Brakeforce Dictionary attacks against POP3 by configuring the below two rules: -

1. Rule for Bruteforce against POP3

alert tcp any any -> $HOME\_NET 110 (msg:"to\_server, potential bruteforce against POP3"; flow:to\_server; threshold: type both, track by\_dst, count 5, seconds 5; classtype:misc-activity; sid:1000013;)

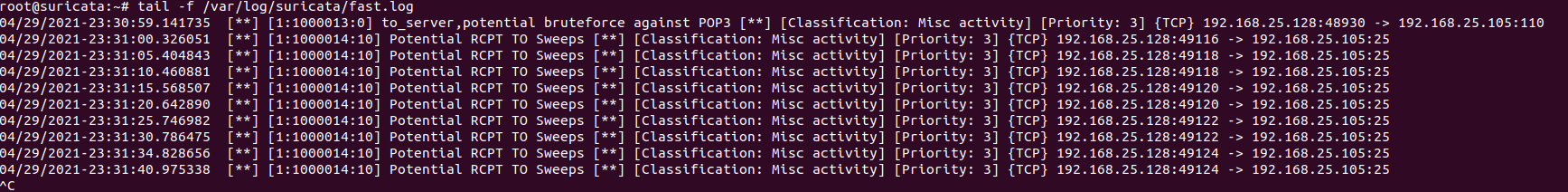
1. Rule for RCPT TO sweeps

alert smtp any any -> $HOME\_NET 25 (msg:"Potential RCPT TO Sweeps"; flow: to\_server; threshold: type both, track by\_dst, count 2, seconds 5; classtype:misc-activity; sid: 1000014; rev: 10;)

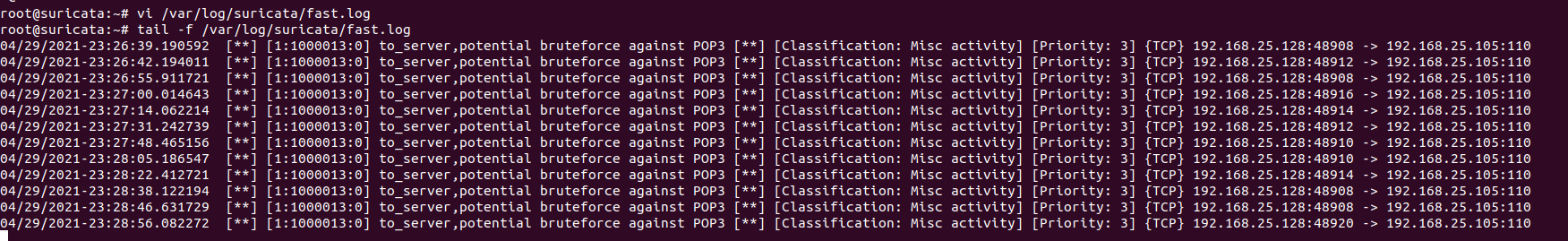
1. **Provide reporting details from Suricata that show it detected the attack.**

**Note: Log file is provided in the Zip Folder**

RCPT TO Attack being detected by Suricata: -



Bruteforce Attack being detected by Suricata:-



**Part 4**

1. **Discuss what else could be performed to defend against these attacks.**

* SMTP STARTTLS or SMTP over [Transport Layer Security](https://en.wikipedia.org/wiki/Transport_Layer_Security)/Secure Sockets Layer can be used to encrypt communications for a single mail hop between the SMTP client and the SMTP server. We can enable TLS Encryption when Postfix receives incoming emails and sends outgoing emails by configuring the /postfix/master.conf and /postfix/main.conf files.
* We will also have to disable insecure SSL Protocols suches SSLv3, TLSv1 and TLSv1.1
* **Configure Postfix** to reject spam. Thereby not having to rely solely on Spamassassin.
* We can also take measures against client to prevent RCPT TO and BruteForce attacks **suc**h **limiting the number of connections that an SMTP Client may make simultaneously or in the timer interval specified etc**
  + smtpd\_client\_connection\_rate\_limit = 20  
    smtpd\_error\_sleep\_time = 10s  
    smtpd\_soft\_error\_limit = 3  
    smtpd\_hard\_error\_limit = 5

smtpd\_client\_auth\_rate\_limit = 3

smtpd\_client\_new\_tls\_session\_rate\_limit = 2

* Configure Sender Policy Framework to specify which hosts or IP addresses are allowed to send emails on behalf of a domain and DomainKeys Identified Mail (DKIM) to thwart spoofing attacks
* We can rely on other tools such as Fail2Ban to protect our Servers from Brute Force Attacks

1. **SMTP is not a secure protocol. Discuss some of the secure alternatives available today.**
2. We can rely on Simple Mail Transfer Protocol Secure (**SMTPS**) for securing SMPT via Transport Layer Security (TLS). Thereby, enforcing the security tenets such as authentication, data integrity and confidentiality
3. **X.400 -** Message security capabilities provided by the X.400 standards are far superior to SMTP/MIME. X.400 is designed to address not only messaging, but also directory, security, and network management. It has strong security standards defined and ensures Robust message delivery and non-delivery schemes. However, However, there are very few large-scale implementations that take advantage of the numerous security-related features specified within the standards
4. [**GPG**](https://en.wikipedia.org/wiki/GNU_Privacy_Guard)**,**[**PGP**](https://en.wikipedia.org/wiki/Pretty_Good_Privacy)**, SMEmail,** or [**S/MIME**](https://en.wikipedia.org/wiki/S/MIME) can be used for [end-to-end](https://en.wikipedia.org/wiki/End-to-end_principle) message encryption