Attacks and Defenses on Email Servers based on Postfix and Dovecot

Shan Zhu, Nairong Zhang, Ziming Chen

***Abstract*—In this project, we will use penetration testing and intrusion detection tools on a server running email server applications to explore various email-based attacks and defenses. We first set up the email server Postfix and POP3 server Dovecot in a Docker container on Ubuntu Virtual Environment. Then we perform four SMTP attacks, including username enumeration attack, open relay attack, email spoofing attack and brute-force dictionary attack to the email servers with the default settings. Next, we explored the possibility to defend these SMTP attacks by modifying the setting in the Postfix servers and utilizing the spam filtering program SpamAssassin. Finally, we introduced Suricata, an intrusion detection tool to detect the SMTP attacks on the email server.**

I. Introduction

A. Postfix

In this project, we use Postfix as the email server, which is an open-source mail transfer agent(MTA) that routes and delivers electronic mail. As an SMTP server, Postfix implements a first layer of defense against spambots and malware. Administrators can combine Postfix with other software that provides spam/virus filtering, message-store access(e.g. Dovecot), or complex SMTP-level access-policies.

B. Dovecot

Dovecot is an open-source IMAP and POP3 server for Unix-like operating systems. The primary purpose of Dovecot is to act as a mail storage server. Mail is delivered to the server using some mail delivery agent (MDA) and stored for later access with an email client (mail user agent, or MUA). Dovecot can also act as mail proxy server, forwarding connection to another mail server, or act as a lightweight MUA in order to retrieve and manipulate mail on remote servers for e.g. mail migration.

C. SMTP Attacks

1) Username enumeration Attack: User enumeration is when a malicious actor can use brute-force techniques to either guess or confirm valid users in a system. For SMTP attack, penetrators can perform username enumeration via the EXPN and VRFY commands if these commands have not been disabled by the system administrator. The role of EXPN command is to reveal the actual address of users' aliases and lists of email and VRFY can confirm the existence of names of valid users. The SMTP enumeration can be performed manually through utilities like telnet and netcat or automatically via a variety of tools like metasploit and smtp-user-enum. In this project, we use ismpt command in Kali VM to perform the attack.

2) Open Relay Attack: Open Relay Attack is by using a third-party email server to relay the message to destination, and its role in this kind of attack is similar to "man-in-the-middle".

3) Spoofing: Spoofing is the act of disguising a communication from an unknown source as being from a known, trusted source.

4) Brute-force Dictionary Attack Against POP3: Mail servers often store very sensitive information, and penetration testers need to perform brute force password auditing against them to check for weak passwords. In this project, we used Nmap to launch dictionary attacks against POP3 servers.

D. SpamAssassin

To defend the attacks above, we integrate SpamAssassin to our email server. SpamAssasin is a computer program used for email spam filtering. It utilizes a variety of spam-detection techniques, including DNS and fuzzy checksum techniques, Bayesian filtering, external programs, blacklists and online databases. It is a highly configurable program that can be run by individual users on their own mailbox and integrates with several mail programs. It comes with a large set of rules which are applied to determine whether an email is spam or not. Most rules are based on regular expressions that are matched against the body or header fields of the message. A message will be given a global score by combining the test scores of the individual test. The higher the score, the higher the probability that the message is spam. The message will be marked as spam if it reaches the score threshold configured in the SpamAssassin.

E. Suricata

Suricata is an open source network threat detection engine that provides capabilities including intrusion detection (IDS), intrusion prevention (IPS) and network security monitoring. It does extremely well with deep packet inspection and pattern matching which makes it incredibly useful for threat and attack detection. It is a rule-based ID/PS engine that utilises externally developed rule sets to monitor network traffic and provide alerts to the system administrator when suspicious events occur. Designed to be compatible with existing network security components, Suricata features unified output functionality and pluggable library options to accept calls from other applications. In this project, we will explore the function of Suricate to detect the potential SMTP attacks.

II. Project Goal

This project is divided into three parts. First, we set up the SMTP server Postfix and the POP3 server Dovecot in an Ubuntu container. Then perform four SMTP attacks, i.e. username enumeration attack, open relay attack, spoofing, brute-force dictionary attack, with the default configuration of the email server. Second, we modify the Postfix server's setting to improve defenses against SMTP attacks and also install SpamAssassin with spam filters to reject spam. Then repeat the four attacks again in the first part. Third, we investigate the attack detection tool, Suricata, to detect the SMTP attacks on the email server. The project motivates us to learn the details of different kinds of SMTP attacks and how to improve email server's defenses corresponding to these attacks.

III. Project Execution Highlights

A. Email Server Setup in Ubuntu

We choose Postfix as our email server for this task as a Mail Transfer Agent, which accepts mail from the outside and from local resources, and routes it to its destination. We also choose Dovecot as the POP3 server, which delivers the mail to its authenticated users on the system. Both servers are installed in a Ubuntu Docker container on the Ubuntu VM.

B. SMTP Attacks with Default Settings

1) Username Enumeration Attack: User enumeration is when a malicious actor can use brute-force techniques to either guess or confirm valid users in a system. For the email server, we can perform the username enumeration attack via RCPT TO command which enumerates different usernames and sends emails with these enumerated addresses to test the validation of the addresses. In the first attack, we use ismtp RCPT command in Kali to test a list of email addresses from a file enumerating usernames from a dictionary file automatically.

2) Open Relay Attack: We used telnet to connect to the remote email server where our email receiver is rooted, and through the SMTP commands, we will be able to communicate with the remote server. We specified our relay domain and listed the sender address which is an arbitrary user guessed from the previous step in the remote server, and in our case, we used dummy@172.16.0.116 as the relay service. In addition, we specified the receiver address. And DATA command starting the transfer of the message contents (body text, attachments etc). After that the DATA command has been sent to the server from the client, the server will respond with a 354 reply code. Finally, the message contents can be transferred to the server. With an email ID shown below.

3) Email Spoofing: Here, we specified three addresses, internal receiver address, internal relay address and an external sender address. Different from what we did in the open relay attack, we did telnet to a server which is outside of organization (in order to simplify our configuration setups, instead of creating another outside email server, we were using the server running on the outside interface, particularly we have an internal server on tap0 and another server on ens33 which is the server outside of organization).In HELO command, the same as we did in the previous experiment, we stated our address as the internal relay address pretending the email is from an internal source. Later by sending the same commands as the open relay attack, we achieved spoofing in this experiment.

4) Brute-Force Dictionary Attack against POP3: We Created the sequence using Nmap, which is a network scanner and used to discover hosts and services on a computer network by sending packets and analyzing the responses. We fetched a large sequence of communication between host and destination using POP protocol. And We can see the sequences in the following figure. By using Nmap, our host kept trying to send requests for username and passwords to match if there exists one valid credentials on the destination side.

46	20:39:23.45047…	172.16.0.139	172.16.0.146	POP	81	C: USER webadmin
47	20:39:23.45066…	172.16.0.139	172.16.0.146	POP	81	C: USER sysadmin
48	20:39:23.47013…	172.16.0.146	172.16.0.139	POP	71	S: +OK
50	20:39:23.50131…	172.16.0.139	172.16.0.146	POP	77	C: PASS root
52	20:39:23.52337…	172.16.0.146	172.16.0.139	POP	71	S: +OK
55	20:39:23.52342…	172.16.0.146	172.16.0.139	POP	71	S: +OK
57	20:39:23.55077…	172.16.0.139	172.16.0.146	POP	78	C: PASS admin
58	20:39:23.55097…	172.16.0.139	172.16.0.146	POP	86	C: PASS administrator
60	20:39:23.57868…	172.16.0.146	172.16.0.139	POP	71	S: +OK
63	20:39:23.57886…	172.16.0.146	172.16.0.139	POP	71	S: +OK
66	20:39:23.60083…	172.16.0.139	172.16.0.146	POP	81	C: PASS webadmin
67	20:39:23.60086…	172.16.0.139	172.16.0.146	POP	81	C: PASS sysadmin

Fig. 1. Results of Brute-Force Dictionary Attack

C. SMTP Attacks with Improved Defenses

1) Improvement of Postfix Server Settings: To better defend the attacks mentioned above, we first improved the basic settings of Postfix by disabling the VRFY communication and adding constraints to its network interfaces. To prevent the unwanted email relaying, we then configure the related networks parameters for the email server. We also enhance the incoming and outgoing email configurations, including enable HELO command, configure authenticated relaying with the smarthost and enable SASL authentication. Besides, we have applied cryptography and encryption techniques with TLS logging enabled.

2) Integration of SpamAssassin: Apache SpamAssassin is an intelligent software application for filtering unsolicited emails from telemarketers and hackers. The utility runs on top of a Mail Transfer Agent (MTA) like Postfix. After installing it in the Ubuntu container, we then need to add users which the SpamAssassin is going to protect and configure the SpamAssassin so that it can be successfully used by the Postfix. We have mentioned before that SpamAssassin will run tests on the emails and give each email a global score to determine if the email is spam or not. Thus, we then need to add rules based on our usage for the running of the tests. After we have configured SpamAssassin, the final step is to integrate it into Postfix by changing its configuration file. Now, we can use SpamAssassin with Postfix to detect the potential spams.

3) Username Enumeration Attack: After improving the Postfix Server Settings and integrating Spam Assassin program, we repeat the username enumeration attack again, and the result is shown below. We can notice that there are more email addresses marked as "invalid" that were marked as "valid" before the improvements. This indicates that the ability

for the Postfix server to defend against username enumeration attacks has been strengthened.

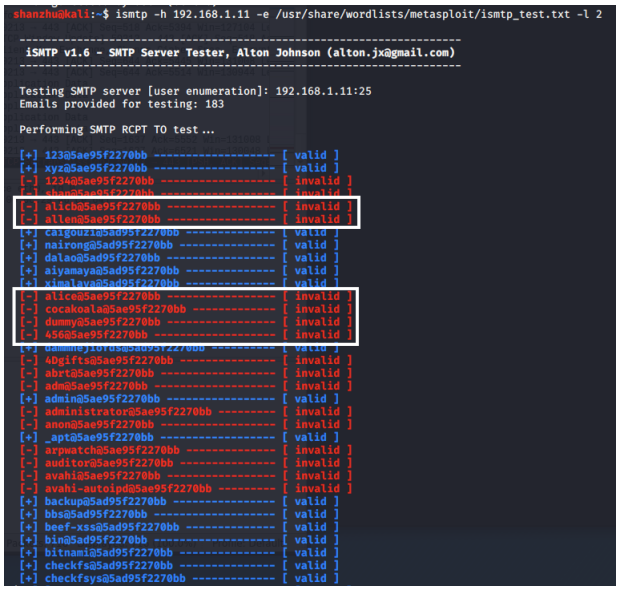

Fig. 2. Results of Attack-1 with Enhanced Settings

4) Open Relay Attack: By following the same SMTP commands we used before as well as improving Postfix settings and installing Spamassassin, We noticed several headers were added to the email when we received, and those headers matched the parameters we set in the configuration file. For example, we have headers that indicate the score of spam. In our case, a persuasive proof for a spam should include spurious links or keywords related to customer service. So for this test case, the score is calculated as 4.5, which is lower than the threshold for tagging as "Spam". We also noticed there is an "unknown address" in the header, which is different from what is displayed in the sender address, this difference indicates we used an internal relay service with an address of 172.16.0.116 and the actual sender address is 172.16.0.139. Through receiving this email, we justify our usage for the open relay attack and the usage of SpamAssassin to detect the potential spam email.

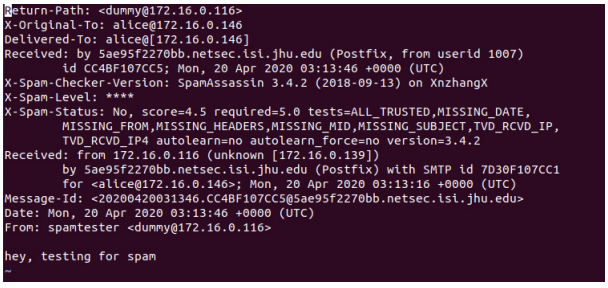

Fig. 3. Results of Attack-2 with Enhanced Settings

5) Email Spoofing: In this experiment, since we were simulating the situation where a suspicious external address is trying to send email to an internal address and using an internal relay service. Here, we changed the score of spam from default 5.0 to 3.0, and since this email scored as 4.5, the "Spam-Flag" was then set to "YES", in addition, the email subject was also changed to alerted spam. From the figure shown below, we proved by using SpamAssassin, we can efficiently detect the potential incoming spam and filter it out.

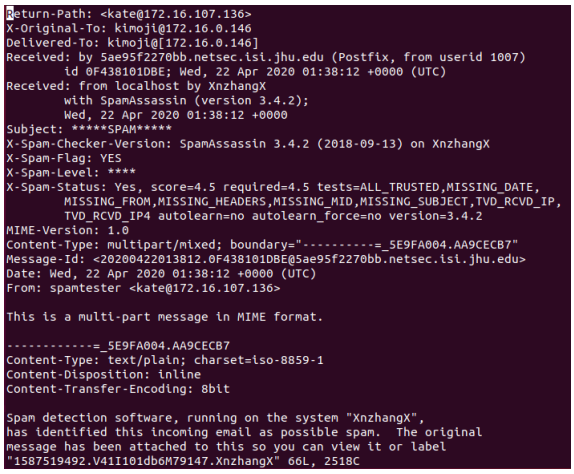

Fig. 4. Results of Attack-3 with Enhanced Settings

6) Brute-Force Dictionary Attack with Enhanced Settings: The result is shown below. From the following figure, we can justify our successful guess of the credentials on our email server, which matches the user created on the email server.

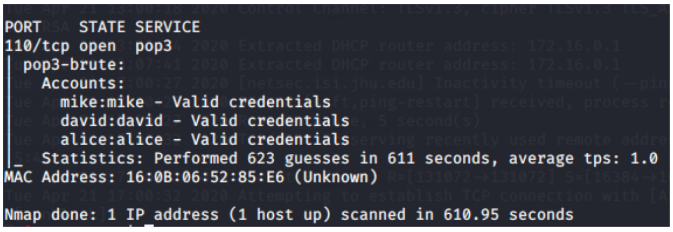

Fig. 5. Results of Attack-4 with Enhanced Settings

D. SMTP Attack Detection on Suricata

To better detect the onset of the SMTP attack, we have performed the experiment with Suricata on Docker. We initially built a docker image with Suricata installed. When we ran the container, we obtained a default setting of Suricata, and configured the network setting to always run on the target interface. As mentioned above, Suricata is a rule-based IDS, thus we created a rule when there are more than 1 attempted connections within 1 second to the local network on port 25, the Suricata will alert a possible SMTP attack. Then we configured Suricata to include this rule in its rule-files section. After the setup of Suricata, we then performed our experiment with it in PCAP live mode and fetched the log files later on.

1) Username Enumeration Attack: We repeated the Attack 1 again with Suricata, and the corresponding log files are shown below. The file indicates that the Suricate could successfully capture the attempted username enumeration attack

and could also report the IP address and ports number of the attacker who is trying to attack the email server.

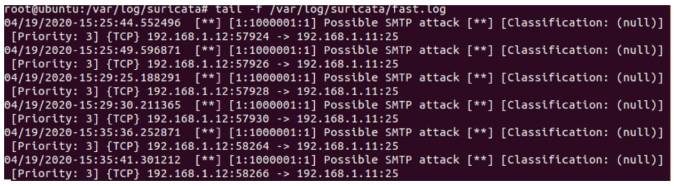

Fig. 6. Records of Attack-1 in Suricata

2) Open Relay Attack: We repeated Attack 2 under the default condition of Postfix, and the results are shown below. This log files records the SMTP communication between host and destination.

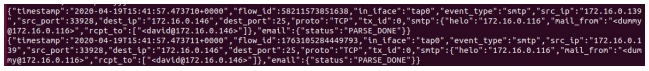

Fig. 7. Records of Attack-2 in Suricata

3) Email Spoofing: We repeated Attack 3 and in order to bring about any contrast after we improved Postfix setting and installed SpamAssassin, we performed email spoofing attack under the improved configuration. From the results shown above, the email was tagged as spam, and we searched through fast.log and eve.json, there are no such detection records, which in other words proved the effective usage of SpamAssassin who filters out any suspicious email.

4) Brute-Force Dictionary Attack against POP3: This result is captured when the team was performing a dictionary attack using Nmap. As stated above, we scanned port 110 on the remote server and kept enumerating usernames and passwords.

Fig. 8. Records of Attack-4 in Suricata

IV. Goal Accomplishment

The project overall accomplished our goal settings, and enhanced our understanding of how SMTP attack is happening and how we can mitigate the attack by applying different techniques. We were introduced and practiced to launch different kinds of SMTP attacks(username enumeration attack, open relay attack, spoofing and brute-force dictionary attack against POP3) with the default configuration of the Postfix server and Dovecot server. In order to fight against these SMTP attacks, we modified default settings of Postfix to mitigate the attack, and also integrate the SpamAssassin program to filter the potential spams. The results suggest that these improvements can successfully enhance the security level of our email server. Besides, through the usage of Suricata, we can achieve the detection of these SMTP attacks by monitoring the log files in the Suricata.

V. What was learned

From this project, we got familiar with the widely-used email server, Postfix and the IMAP/POP3 server Dovecot and how to install them in a Docker container on Ubuntu Environment. We also learned how to perform different kinds of SMTP attacks on the Postfix server both internally and externally. Facing these attacks, we investigated different methods to improve the defenses of the Postfix server by modifying its settings and integrating other defending programs like SpamAssassin. Besides, we also explored the usage of another useful tool, Suricata, to detect possible SMTP attacks on the server and report potential threatening client IP addresses. In general, we got a profound understanding of penetration testing and intrusion detection on the Postfix server through the interactions with SMTP attacks and defenses.

References

[1] "Postfix (software)", En.wikipedia.org, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Postfix-(software). [Accessed: 20- Apr- 2020].

[2] "Dovecot (software)", En.wikipedia.org, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Dovecot-(software). [Accessed: 20- Apr- 2020].

[3] "SMTP User Enumeration", Penetration Testing Lab, 2020. [Online]. Available: https://pentestlab.blog/2012/11/20/smtp-user-enumeration/. [Accessed: 20- Apr- 2020].

[4] "How to Secure Postfix with SpamAssassin on an Ubuntu 18.04 VPS or Dedicated Server — HostAdvice", HostAdvice, 2020. [Online]. Available:https://hostadvice.com/how-to/how-to-secure-postfix-with-spamassassin-on-an-ubuntu-18-04-vps-or-dedicated-server/. [Accessed: 21- Apr- 2020].

[5] "iSMTP", Tools.kali.org, 2020. [Online]. Available: https://tools.kali.org/information-gathering/ismtp. [Accessed: 21- Apr- 2020].

[6] Thexplorion.com, 2020. [Online]. Available: https://thexplorion.com/7-common-email-security-protocols-explained/. [Accessed: 22- Apr- 2020].

[7] "SMTP Hacks and How to Guard Against Them - dummies", dummies, 2020. [Online]. Available: https://www.dummies.com/programming/networking/smtp-hacks-and-how-to-guard-against-them/. [Accessed: 22- Apr- 2020].

[8] "SecurityTrails — The Social Engineering Toolkit", Securitytrails.com, 2020. [Online]. Available: https://securitytrails.com/blog/the-social-engineering-toolkit. [Accessed: 22- Apr- 2020].

[9] "Send Fake Mail using SETOOLKIT [Kali Linux] - Yeah Hub", Yeah Hub, 2020. [Online]. Available: https://www.yeahhub.com/send-fake-mail-setoolkit-kali-linux/. [Accessed: 22- Apr- 2020].

[10] "IBM Knowledge Center", Ibm.com, 2020. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSLTBW-2.1.0/com.ibm.zos.v2r1.halu001/smtvrfy.htm. [Accessed: 22- Apr- 2020].

[11] S. block, "Suricata custom rule alerts but won't block", Netgate Forum, 2020. [Online]. Available: https://forum.netgate.com/topic/146708/suricata-custom-rule-alerts-but-won-t-block/50. [Accessed: 22- Apr- 2020].

[12] ”Open Relay Test — exchange.sembee.info”, Exchange.sembee.info, 2020. [Online]. Available: http://exchange.sembee.info/network/openrelaytest.asp. [Accessed: 22- Apr- 2020].
[13] ”Brute force and dictionary attacks: A cheat sheet”, TechRepublic, 2020. [Online]. Available: https://www.techrepublic.com/article/brute-force-and-dictionary-attacks-a-cheat-sheet/. [Accessed: 22- Apr- 2020].