Annual Academic Sessions 2012
ISSN 2012-9912; ©2012 Open University of Sri Lanka

STRATERGIES IN HARDENING AND PROTECTION MECHANISMS FOR A LINUX BASED E-MAIL SYSTEM

M. Punchimudiyanse[17]

[1]*Department of Mathematics and Computer Science, The Open University of Sri Lanka*

INTRODUCTION

An E-mail server of an organization is a one of the main threads that binds together communications with peers within the organization as well as the outside world. Providing a secure service which is protected from viruses and all kinds of malware, equipped with spam minimization methodologies and provide adequate facilities to the user community is a challenging task for an administrator of the in house e-mail server in an organization.

It is imperative to note that not all the users are technology savvy, not prone to mistakes or not become victims of socially engineered methods used by attackers. A weak password or user giving out his/her password to an attacker will lead to a compromise of an e-mail account. An e-mail server must prevent e-mails coming from a compromised host (eg. Virus affected host) from reaching the other users. System should enforce strong user passwords, protected from internal threats from the institutional network as well as threats coming from the Internet. Using e-mail server as an open relay, forging identity of other users, using the web mail interface for spamming, planting some files within web server of a mail server to act as a proxy to attack a different machine, try to send e-mails of a compromised server in your institution via e-mail server *etc* are common threats. E-mail server needs to interact with various servers throughout the world when sending and receiving e-mails. Worldwide e-mail filters may black list organizational e-mail server if suspicious behavior is detected.

Author's primary aim of this paper is to present a group of strategies that could be used to mitigate above mentioned problems when configuring a new Linux based email server.

METHODOLOGY

A basic working model of an e-mail server with a single mail transport agent (MTA) could be depicted in figure 1.

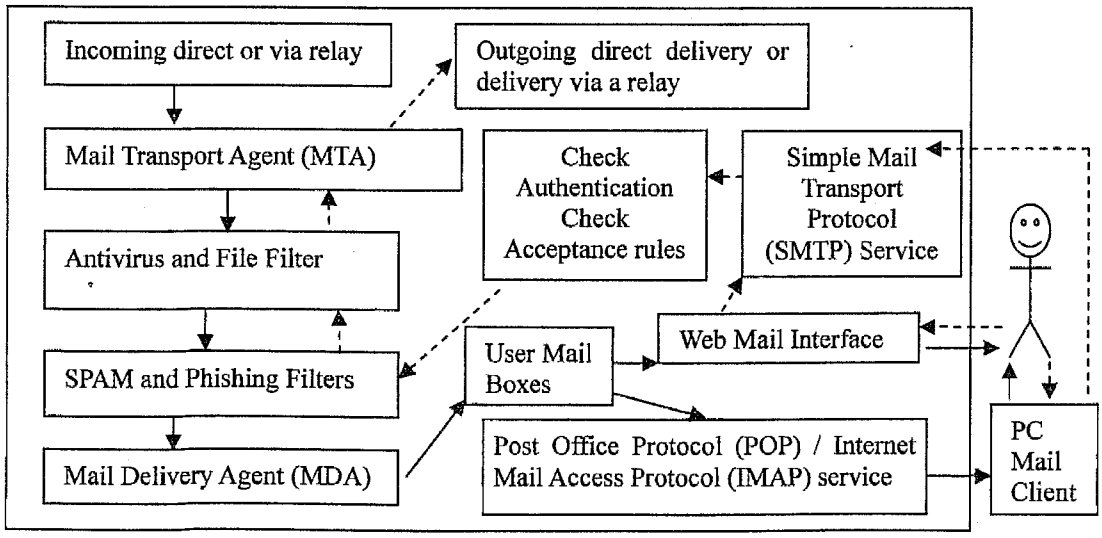

Figure 1 – Email server working model (---➤ outgoing mail, ⟶ incoming mail)

[7] Correspondences should be addressed to Mr. M. Punchimudiyanse, Department of Mathematics and Computer Science, The Open University of Sri Lanka (email: malinda@ou.ac.lk , Tel : 0112881098)

The modular working model of an e-mail server (Redhat Inc, 2007) paved the way to select the better software for each task. There could be different combinations of MTA, MDA, Antivirus, SPAM & Phishing filters, Web mail Interface, POP/IMAP/ SMTP servers based on the requirements of the institution and the capabilities of the systems administrator. Author suggests choosing software with a good documentation; rich set of features, less security exploits and easy upgradability, with self maintenance features *etc*.

Install only the minimum required services and software to run an e-mail server at the installation. Author suggests following configuration strategies for a server.

User creation / password complexity strategy – Do not allow e-mail users to login to a server via secure shells or interactively. E.g. use */sbin/nologin* as login shell. Enforce a complex password acceptance mechanism in e-mail server. Password change policy of the most Linux systems could be altered to accept complex passwords even when super user tries to change.

Webmail interface password change strategy – Some web mail interfaces only allow to have minimum password length. Therefore if a user tries a simple password it will be accepted by the system because password change mechanisms are programmed to run using super user permissions which will override complexity rules. To prevent this alter the code of password change pages of respective web mail interfaces. Eg. In OpenWebmail interface (T. Chung 2005) a strong password could be enforced by altering the code of openwebmail-prefs.pl file procedure "change password". Original code only checks for upper and lower case combinations [A-Za-z]. Author changed the source code to include numbers by specifying [A-Za-z0-9].

Outgoing mail acceptance strategy – configure e-mail server to accept outgoing mails only from internal network unless somebody sends e-mail using PC clients from home. Alternate strategy is to use web mail at home and use PC clients at office. Turn on the authentication and SASL or TLS for all the SMTP clients. Limit the number of e-mails that could be sent per user per hour. Treat all the hosts are equal and configure the mail filtering rules in SPAM filters (P. Koetter, 2007).This will prevent any compromised host in internal network from sending mail out. Do not trust the peer servers especially web servers as they might have unprotected web forms which could be susceptible for attacks. Put a rule to attach your domain name at the back of un-resolvable domain names (Do a reverse DNS check) as well as behind all usernames of your domain(eg. samana.lk) to prevent a user bill@samana.lk from sending e-mails such as bill@microsoft.com using e-mail server. This rule will prevent phishing attacks coming from a trusted internal host or a server.

To prevent users from forging from address, enforce a mapping list for username and canonical name and prevent user from changing from address after the authentication. Eg. user = palithu should not be allowed to send e-mail as coming from user = samanp. (F. A. Solana, 2006). Alternate strategy is using sender policy framework (SPF) or use domain keys identified mail (DKIM) (E. Allman, 2009) technology with your with Domain Name Service (DNS) provider to sign your mail.

Incoming mail acceptance strategy – all internal/external incoming mail should be checked for viruses and malware to prevent viruses spreading via e-mails. Antivirus program must detect file type by content and check the compressed files about minimum of three levels to detect extension changed files. Prevent users from e-mailing executable files or scripts. Some of the antivirus programs such as clamav are designed to spawn children based on the load of the e-mail server as well as do automatic updates and does kill and restart of scanning processes to prevent memory leaks. Use SPF and/or DKIM to prevent incoming spam.

Tweak the spam filter to learn about spam by giving it a sample of common spam messages that commonly encounter. It also could be configured to check with online spam lists. If a score based spam filter is used it is good to have a high score to start with and gradually

decrease it to a level where you have a lower number of false positives (i.e. good messages are classified as spam) Encourage users to submit spam samples to a particular e-mail address and then you can train your spam filter to avoid mails like them in future.

Web interface permissions – It is a must to use recommended permissions in the folders of the webmail interface and the rest of the folders in the web server root folder and subfolders of the e-mail server. Do not issue 777 permissions on web root folder because attacker could plant a script file and run it with elevated permissions and use your server for his/her unauthorized activities. Always turn on SSL in your web interfaces, if you cannot obtain a commercial SSL certificate from a provider, configure and generate one for your server.

Security Enhanced Linux (SE Linux) and Firewall – Change the secure shell (SSH) port from port 22 to a different value and limit the number of unsuccessful login attempts. Open only the relevant ports that are need to run mail services in the IP tables firewall. Change the SE Linux mode to *enforcing* after all configuration changes to attach security labels to all the configuration files and relevant executable scripts. This will ensure that nobody could run file which are not having a proper security label attached. It will prevent attacker from running a script even if they succeed in planting it inside e-mail server.

RESULTS AND DISCUSSION

Comparison of threat mitigation in old e-mail server without the strategies mentioned above and a new server with above strategies is presented in Table 1. Results obtained via log file analysis. Both servers operated outside perimeter firewall with the highest attack surface.

Attempted Threats to the system	**Outcomes of the strategies (old server)**	**Outcomes of the strategies (new server)**
Gain unauthorized shell access	A daily list of intrusion attempts from IPs of various countries	Prevented because of change of ports and limitation of login attempts in secure access shell.
Obtain a User password using social engineering techniques and send mass mail	System did not prevent this problem.	10 attempts within a period of 6 months. SPAM filter has prevented the spamming from reaching the other users.
Attempts to use as open relay	System did not prevent this problem.	Prevented by tight mail acceptance strategy. Test mail servers installed inside laboratories was automatically denied.
Black listing by the other servers	Server was blacklisted by hotmail and yahoo domains from accepting mail.	Could not totally prevent this without application of SPF and Domain keys which is out of control of the Author's responsibility.
Attempting to plant files	Due to faulty permissions applied on a configuration change, one attempt was successful	Planting files were promptly denied by the access permissions. Prevented even an accidental modification because SE Linux was turned on.

Table 1 – Comparison of new and old server outcomes on threats and strategies applied

New server operated successfully for duration of six months without a single reboot. Configuration summary of an old e-mail server and new e-mail server is given below.

Old Server : OS-Redhat linux 9, MTA – Sendmail, Antivirus – clamav (a free virus scanner), SPAM filter – Email Scanner, Webmail – Openwebmail, Pop/IMAP/SMTP services –APOP.

New server : OS – CentOS, MTA – Postfix, Antivirus – clamav, SPAM filter – Amavisd + Spamasssin, Webmail – Openwebmail, Pop/IMAP/SMTP services – Dovecot with SASL.

CONCLUSIONS/RECOMMENDATIONS

The best approach to deploy a highly available and secure e-mail server is to plan from the installation of an operating system itself. A good operating system (OS) with at least two to three years availability of system updates and less number of known security exploits has to be chosen. One of the recommended configurations of disk system is to use in an e-mail server is (Raid 1) mirroring in the operating system partitions. Use Raid-5 based disk system for mail boxes storage partitions. This will ensure the robust e-mail server without disk/OS failure issues, if the hot swappable disks are used then there is zero down time changing a defective hard disk. Discussed configuration strategies under methodology are recommended for institutional e-mail server which runs only on a single domain. If more than one user is responsible for management of an e-mail server it is recommended to use accounts with sudoer permissions and keep the root user account only for special purposes.

Author has deployed a mechanism to get an incremental backup copy of email boxes to a different machine daily via a non blocking Network File System (NFS) mount and a CRON job. Incremental copy of inboxes could be used to recover inboxes of users when they have accidentally downloaded all mail to PC client via POP mechanism. Alternate strategy is to have two e-mail servers on Active-Active or Active – Standby setup which was not possible in Author's case. The mechanisms such as Digital signatures will also help prevent forging of user's identity but some users are not aware or not capable of using it. SPF and/or DKIM technologies are essential to prevent server blacklisting by global SPAM lists. The ideal location to place a production e-mail server is the DMZ (de-militarized zone).

REFERENCES

J. Mehnle. (April 2010). Sender Policy Framework Introduction. Retrieved from http://www.openspf.org/Introduction

E. Allman *et al.* (August 2009). DomainKeys Identified Mail (DKIM) Author Domain Signing Practices (ADSP). Retrieved from http://www.rfc-editor.org/rfc/rfc5617.txt

P. Koetter,M. Martinec. (June 2007) Integrating amavisd-new in Postfix. Retrieved from http://www.ijs.si/software/amavisd/README.postfix.html

Redhat Inc. (January 2007) Red Hat Enterprise Linux Deployment Guide - Email (chapter 23). Retrieved from http://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-email.html

F. A. Solana. (February 2006). Block sender address spoofing with SMPT AUTH. Retrieved from http://www.felipe-alfaro.org/blog/2006/02/19/block-sender-address-spoofing-with-smpt-auth/

T. Chung (2005). Open WebMail Project. Retrieved from http://www.openwebmail.org

ACKNOWLEDGEMENTS

Author wishes to acknowledge his teacher Dr. Shantha Fernando (Senior Lecturer, Department of Computer Science and Engineering, University of Moratuwa) who has made him an enthusiast in the field of computer and network systems security.