PiMail: Affordable, Lightweight and Energy-Efficient Private Email Infrastructure

Sufian Hameed, Muhammad Arsal Asif, Farhan Kamal Khan
IT Security Labs, National University of Computer and Emerging Sciences (FAST-NUCES), Pakistan
sufian.hameed@nu.edu.pk

Abstract—Third-party Email services, whether a free webmail or corporate service hosted somewhere over the Internet, requires sacrifice of control and flexibility over personal communication. This make emails vulnerable to privacy leaks due to unauthorized access and inspection. In this paper, we propose PiMail, an affordable, lightweight and energy-efficient private email infrastructure, for individuals and small enterprises, based on Raspberry Pi. We also deployed a testbed implementation of PiMail using Raspbian OS, Postfix mail transfer agent (MTA), ClamAV antivirus and SpamAssassin. Based on our in-depth performance evaluation we conclude that PiMail is fully capable of providing email services to individuals and SMEs (small and medium-sized enterprises). To the best of our knowledge, this is the first extensive study to evaluate and benchmark the efficacy of using Raspberry Pi as low-cost and portable mail server.

Index Terms—IoT, Private Mail Server, Raspberry Pi, Email Privacy

I. INTRODUCTION

Electronic mail or Email is no doubt one of the biggest services being used over the internet today. Email has been around for more than 44 years and its transmission is based on Simple Mail Transfer Protocol (SMTP) [13]. With the widespread global usage of emails, the concept of email privacy is also under rapid threat. Email privacy is a very complicated issue that has to deal with unauthorized access and inspection of electronic mail. This unauthorized access can happen while an email is stored on un-trusted email servers of the service providers (like Gmail, Yahoo, Hotmail, etc.) or due to broader government surveillance programs such as PRISM [5].

Third-party mail service is easier to use, but they require sacrifice of control and flexibility. Running a private mail server can be an ideal solution to protect email privacy against unauthorized storage access. Private mail server allows full control over both the server and the emails, complete access to mail server’s logs, and access to raw email files in a user’s mail directory. One of the greatest benefit is the possibility to troubleshoot and fix things on your own, instead of requesting a third-party mail service provider. Nevertheless, this control and flexibility comes with an added cost. Running a private email server for individuals or small scale enterprises can cost from $7 to $15 monthly or more, depending on the configurations [2].

In this paper, we propose PiMail, an affordable, lightweight and energy-efficient private email system based on Raspberry Pi [6]. Raspberry Pi is low-cost, low-power and highly portable single board computer. Raspberry Pi is one of the smallest, credit-card sized, single board computer available in the market that has the highest performance to cost ratio. Raspberry Pi makes it possible to create an affordable, energy-efficient and portable miniature private mail server according to the need of individual users or small enterprise. In short, PiMail fulfils the following promises.

- A low cost infrastructure that would cost a one time investment of $35 to purchase Raspberry Pi 2.
- Personalized email address like MrX@mydomain.com with an annual recurring cost of domain registration with a registrar like namecheap.com.
- Low electricity consumption with an email server that can run 24/7/365 for under $5 of electricity per year.
- The ability to connect from anywhere, and read & send email, using a secure IMAP connection on your phone, tablet or computer.
- Complete control over your personal communication. Emails are stored over PiMail server, and nobody scan them to sell adverts.
- Smart spam filtering with SpamAssassin [9].
- Efficient virus scanning with ClamAV [1].

We developed a testbed implementation of PiMail using Raspbian OS, Postfix [4] mail transfer agent (MTA), ClamAV antivirus and SpamAssassin. We used different experiments to evaluate email processing latency, throughput and CPU/memory utilization of PiMail. The small size, low power consumption and performance benchmarks make Raspberry Pi an ideal candidate for personal email server in home and small organizations.

The rest of the paper is organized as follows. §II summarize the basic properties of Raspberry Pi computer. §III describes the software stack architecture of PiMail. In §IV we discuss the testbed and demonstrate the performance of the proposed system under different experiments. Finally we conclude the paper in §VI.

II. THE RASPBERRY PI COMPUTER

In early 2006, Eben Upton, a British engineer, brought together a group of teachers, academics and IT professional to address the lack of programmable hardware to teach computer science at school level. This lead to the official formation of Raspberry Pi foundation, a charitable organization, in 2009
In 2011, the Raspberry Pi foundation developed a credit-card sized, single-board computer called Raspberry Pi. The first version of Pi i.e. Raspberry Pi 1 (model A) officially went on sale from 29th February 2012. The foundation has since then released couple of updates to Pi 1 and in February 2015 the latest Raspberry Pi 2 (model B) was introduced as the second generation of Raspberry Pi.

### System of Chip

<table>
<thead>
<tr>
<th>System of Chip</th>
<th>Broadcom BCM2836</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>900 MHz quad-core ARM Cortex-A7</td>
</tr>
<tr>
<td>GPU</td>
<td>Broadcom VideoCore IV</td>
</tr>
<tr>
<td>Memory</td>
<td>1 GB SDRAM (shared with GPU)</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Onboard 10/100 Ethernet RJ45 jack</td>
</tr>
<tr>
<td>USB</td>
<td>4 USB 2.0 ports</td>
</tr>
<tr>
<td>Video Output</td>
<td>HDMI (rev 1.3 &amp; 1.4)</td>
</tr>
<tr>
<td>Audio Output</td>
<td>3.5 mm jack, HDMI</td>
</tr>
<tr>
<td>Storage</td>
<td>MicroSD slot</td>
</tr>
</tbody>
</table>

**TABLE I**

**HARDWARE SPECIFICATION OF RASPBERRY PI 2 (MODEL B)**

A. System Specification for Raspberry Pi 2

We have used Raspberry Pi 2 (see Fig. 1) for PiMail, which is based on an integrated circuit (system on chip) combining a quad-core 900 MHz ARM Cortex-A7 processor (CPU), a Broadcom VideoCore graphics processor (GPU), and 1 GB RAM on a single chip [7]. Moreover, there is a microSD card slot for storage and I/O units such as USB, Ethernet, audio, RCA video, and HDMI. Power is provided via a 5 V micro-USB connector. Table I summarizes the hardware specification of Raspberry Pi 2 (model B).

B. System Software

In terms of system software, the Raspberry Foundation has done a tiring effort in optimizing softwares to get the best out of Raspberry Pi 2 (Broadcom BCM2836 application processor and its 900MHz quad-core ARM Cortex-A7 CPU). The recommended OS for Raspberry Pi 2 is Raspbian [8], which is a port of the well-known Linux distribution, Debian. Raspbian is optimized for the ARMv6 and ARMv7 instruction set with hardware floating point support. Raspbian is optimized around 35,000 pre-built packages, for easy installation on Raspberry Pi. This includes WebKit, LibreOffice, Scratch, Pixman, XBMC/Kodi, libav and PyPy. With the introduction of ARMv7 core, Raspberry Pi 2 can also run Ubuntu and Pi 2 compatible version of Windows 10 [10].

### III. SYSTEM DESIGN

The PiMail software stack for an individual Raspberry Pi 2 is shown in Fig. 2. PiMail runs Raspbian [8] (a distribution of Debian optimized for the Raspberry Pi hardware) from a 16 GB microSD card storage. The PiMail server is placed on top of the Raspbian operating system. We have used Postfix, SpamAssassin, ClamAV and Davelcot tools (briefly discussed below) to setup the mail server.

A. Postfix

Postfix [4] is a fast, easy to administer and secure Mail Transfer Agent (MTA) that can support LDAP, SMTP and AUTH (SASL). It was developed by Wietse Venema in 1997 as an alternative to SendMail.

Postfix performs a number of steps to deliver an email to any particular Inbox. Postfix receives an email via the Simple Mail Transfer Protocol Daemon (SMTPD) server. It then strips away the SMTP encapsulation, perform some sanity checks and alerts the cleanup server with the details on sender, recipients and content. The cleanup server inserts the details into the inbound mail queue and informs the queue manager that a new mail has arrived. This mail queue acts as a temporary buffer to the mail directory. As soon as the mail is inserted into the inbound queue a unique queue id is assigned to it. During our experiments we used this unique queue id to keep track of the mail.

B. SpamAssassin

SpamAssassin [9], which is one of the most widely used content-based filter, uses a variety of defense mechanisms to filter spam before it reaches the mailbox. The defense mechanisms mainly includes header tests, body phase tests, Bayesian filtering, automatic address whitelister/blacklist, automatic sender reputation system, manual address whitelister/blacklist,
collaborative spam identification database, DNS blacklist and character sets. Due to these thorough tests it is difficult for the spammers to identify simple work around while creating spam. During the evaluations, we used SpamAssassin with its default settings to identify spam emails and separate them for legitimate ones. After completion of any single test, SpamAssassin assign cumulative score to the mail and check against the user defined threshold. If the calculated score is higher than the user defined threshold, the mail is marked as spam and sent to the spam box.

C. ClamAV

Clam AntiVirus [1] is an open source antivirus toolkit designed for the scanning of emails at the mail gateways. ClamAV analyze the mail from the inbound queue using shared libraries of the anti-virus engine. If the scan results in a positive ID the file/mail is moved to a quarantine folder, else the mail is queued back into the mailing inbound folder.

D. Dovecot

Dovecot [3] is a secure IMAP server that provide IMAP functionality to fetch the mail from the mail directory. It is also used to provide simple authentication and security layer (SASL) to validate the identity of a user before he can send or receive an email.

IV. TESTBED AND EVALUATIONS

For evaluating the system performance of email processing with PiMail, we augmented Raspberry Pi 2 with Postfix MTA, SpamAssassin content filter, ClamAV antivirus and Dovecot IMAP server, and deployed it over the LAN. For all the experiments, we used a desktop machine connected via LAN to send mails with different size and frequencies to PiMail server. We conducted experiments in four different scenarios as follows.

1) S1: In scenario 1 (S1), the SMTP server runs postfix without any spam filter.
2) S2: In scenario 2 (S2) ClamAV is used as an anti-virus with Postfix.
3) S3: In scenario 3 (S3) SpamAssassin is used as a content-based filter with Postfix.
4) S4: In scenario 4 (S4) SpamAssassin is used as a content-based filter and ClamAV is used as an anti-virus with Postfix.

We run different experiments using these four scenarios to study the impact of message size, processing delays, end-to-end throughput, CPU and memory utilizations.

A. Processing Delays

We used two different modes to measure the email processing latency for the four scenarios listed above. First we bombard (as rapidly as possible) the PiMail server with 50 messages to saturate the mail server. After that we added a delay of 1 sec between any two messages. For all the experiments we used fix message size of 8 KB, under the assumption of being the average size of email message [12]. Fig. 4 shows the average email processing delay of all the four scenarios. In the burst mode S1 was able to process an email with an average delay of 0.74 seconds. This increased to 3.2, 54 and 72 seconds in S2, S3 and S4 respectively. After introducing a delay of 1 second between two consecutive messages the average email processing delay was observed to be 1.1, 2.6, 9.8 and 24.5 seconds in S1, S2, S3 and S4 respectively. Spam filtering with SpamAssassin is involving task on PiMail server and with the burst of incoming messages it creates a bottle neck. Delay of 1 second smooth out the delays by 66% in worst case scenario (S4).

B. Throughput

In order to measure the end-to-end throughput of the PiMail server, we used the same settings (sending of 50 email messages of 8 KB each in burst and with 1 second delay) discussed before. Fig. 5 shows the throughput of all the four scenarios. Even with the bombardment of messages S1 was able to receive approximately 81 messages per minute in burst mode. This reduced to 18.75, 1.11 and 0.83 messages per minute S2, S3 and S4 respectively. After introducing a
delay of 1 second between two consecutive messages the average end-to-end throughput came out to be 54.55, 23.08, 6.12 and 2.45 messages per minute in S1, S2, S3 and S4 respectively. PiMail can effectively handle simple email processing in burst mode. Virus scanning and Spam filtering introduce processing delays which directly effects the overall system throughput.

C. Effect of Message Size

We measure how the size of the message effects the time required to process it and the end-to-end throughput. For this we sent 50 messages with varying sizes (8 KB to 64 KB) in burst and with 1 second delay. Fig. 6 shows the average processing delays of all the four scenarios in burst mode. The processing delays of 64KB message for S3 and S4 spikes exponentially to 236 and 366 seconds respectively. These delays smooth out to 159 and 304 seconds for S3 and S4 respectively (see Fig. 7), when an interval of one second was introduce between two messages. Message size also have a direct impact on end-to-end throughput. Fig. 8 & 9 shows that with the increase in message size the throughput can dip to 0.16 message per minute. SpamAssassin utilize content based filtering, which is directly related to the size of the message. Stand alone mail server deployment or addition of ClamAV have negligible effect on the size of the message.

D. CPU and Memory Utilization

To evaluate the CPU and Memory utilization of PiMail in all four scenarios, we continuously sent email message of 8 KB every 0.6 seconds for a total time of 8 minutes or 480 seconds. These experiment settings are based on email statistics from a large university discussed in [12].
The average CPU utilization results (Fig. 11) show that S3 and S4 are expensive in terms of CPU usage (70%-75%). This is because content-based filtering has to apply different rules and filters. In contrast, the CPU usage of S1 and S2 remained as low as 13%-29% on average.

As shown in Fig. 13, the memory usage in S2 and S4 remained constant to 53% and 63.52% respectively. Memory utilization is highest in virus filtering as compared to other scenarios. Surprisingly, there was not much difference in memory utilization when SpamAssassin (S3) was added to simple mail server (S1).

E. Processing Delays with Low Email Volume

In the end we measure the effect of low volume (without burst) of emails on the PiMail server. We sent 50 email messages of 8 KB each with an interval of 1 minute between any two messages. Fig. 14 shows the average email processing delays for all the four scenarios. In S1, the PiMail server was able to process an email with an average delay of 1.06 seconds. This increased to 3.2, 10.6 and 18.6 seconds in S2, S3 and S4 respectively.

This indicates that PiMail server can handle low volume (without burst) of emails without creating any backlogs and
it can fully cater for the needs of individual users and SMEs (small and medium-sized enterprises).

V. DISCUSSION

A. Longer Interval Delays

The experiments in §IV were mainly designed to stress test the performance of PiMail under different settings. In §IV-E we explored longer interval delay time of 1 minute between average sized emails. Fig. 14 shows that with longer interval delays, PiMail server processed the emails well inside the interval delays without creating any backlogs. At this point we consider it unnecessary to explore any longer interval delay between emails.

B. Email Attachments

Reasonable proportion of emails today (10-20%) comes with an attachment i.e. graphics, spreadsheets, pdf, Word documents, etc. Emails with and without attachment(s) follow the same workflow in PiMail. The email processing delays and throughput mainly depends on the size of attachment/message. In §IV-C we measured the effect of message size on the processing delays and throughput. Based on the results we observed that stand alone mail server deployment or addition of ClamAV have negligible effect on the size of the message. On the other hand SpamAssassin utilize content based filtering, which is directly related to the size of the message. Thus, large attachment size will affect the PiMail performance in scenario S3 and S4.

C. Target Users

The main target users of PiMail infrastructure are individuals or SMEs with 10-250 users and aggregate email volume of 2500-4000 emails per day.

VI. CONCLUSION

In this paper, we propose PiMail, an affordable, lightweight and energy-efficient private email infrastructure based on Raspberry Pi 2. To the best of our knowledge, we performed the first extensive study that benchmarks the performance of Raspberry Pi used as a portable and private mail server.

The experiments were designed to stress test the performance of PiMail under different configurations. Based on the results, we observed that content-based spam filtering with SpamAssassin is the most resource hungry process. With high volume of emails, PiMail experienced performance bottleneck when configured to perform content filtering.

Having said that, if we are to focus on providing email services to individuals and SMEs, it would be unrealistic to have back to back emails or even with a short interval of just 1 second. With an interval of 20-30 second, even the most decorated configuration of PiMail (S4) will not exhaust the resources and there will not be any backlogs. In the end, we can conclude that PiMail is capable handling a volume of 4000 emails with frequency 3 emails per minute, which is more than enough for individuals and SMEs.

REFERENCES