Finding spam sending php scripts on multiple sites servers - Tracing and stopping spammer PHP scripts

Author : admin

Spam has become a severe issue for administrators, not only for *mail server admins* but also for *webshosting adms*. Even *the* ***most secure spam protected mail server*** **can get affected by spam** due to fact it is configured to relay mail from other servers acting as web hosting sites.

Webhosting companies *almost always suffer seriously from spam issues* and often their mail servers gets blocked (*enter spam blacklists*), because of their irresponsible clients uploading lets say *old vulnerable Joomla, Wordpress* without *Akismet* or *proper spam handling plugin*,a CMS which is not frequently supported / updated or *custom client insecure php code*.

What I mean is Shared *server* **A** is often configured to sent mail via *(mail) server* **B**. And often some of the *many websites / scripts hosted on server* ***A*** gets hacked and a spam form is uploaded and tons of spam start being shipped via mail server **B**.

Of course on mail server level it is possible to *configure delay between mail sent* and *adopt a couple of policies to reduce spam*, but the spam protection issue can't be completely solved thus admin of such server is forced to *periodically keep an eye on what mail is sent from hosting server to mail server*.

If you happen to be one of those *Linux (Unix) webhosting admins* **who find few thousand of spammer emails into mail server logs or your eMail server queue and you can't seem to find what is causing it**, *cause there* are *multiple websites shared hosting using mainly PHP + SQL* and you **can't identify what php script is spamming** by *reviewing Apache log / PHP files*. What you can do is get use of:

PHP mail.log directive

Precious tool in tracking spam issues is a PHP Mail.log parameter, *mail log paramater is available since*

PHP version >= 5.3.0 and above.
PHP Mail.log parameter records all calls to the PHP mail() function including *exact PHP headers, line numbers and path to script initiating mail* sent.

Here is how it is used:

1. Create empty PHP Mail.log file

touch /var/log/phpmail.log

File has to be writtable to same user with which Apache is running in case of *Apache with SuPHP* running file has to be writtable by all users.

On *Debian, Ubunut Linux*:

chown www:data:www-data /var/log/phpmail.log

On CentOS, RHEL, SuSE phpmail.log has to be owned by **httpd**:

chown httpd:httpd /var/log/phpmail.log

On some other distros it might be chown **nobody:nobody** etc. depending on the user with which Apache server is running.

2. Add to *php.ini* configuration following lines

mail.add_x_header = On
mail.log = /var/log/phpmail.log

PHP directive instructs PHP to *log complete outbund Mail header sent by mail() function*, containing the UID of the web server or PHP process and the name of the script that sent the email;

(*X-PHP-Originating-Script: 33:mailer.php*)

i.e. it will make php start logging to *phpmail.log* stuff like:

*mail() on [/var/www/pomoriemonasteryorg/components/com_xmap/**2ktdz2.php**:1]: To: info@globalremarketing.com.au -- Headers: From: "Priority Mail" X-Mailer: MailMagic2.0 Reply-To: "Priority Mail" Mime-Version: 1.0 Content-Type: multipart/alternative;boundary="------ ----13972215105347E886BADB5"*
mail() on [/var/www/pomoriemonasteryorg/components/com_xmap/2ktdz2.php:1]: To: demil7167@yahoo.com -- Headers: From: "One Day Shipping" X-Mailer: CSMTPConnectionv1.3 Reply-To: "One Day Shipping" Mime-Version: 1.0 Content-Type: multipart/alternative;boundary="--- -------13972215105347E886BD344"
mail() on [/var/www/pomoriemonasteryorg/components/com_xmap/2ktdz2.php:1]: To: domainmanager@nadenranshepovser.biz -- Headers: From: "Logistics Services" X-Mailer: TheBat!(v3.99.27)UNREG Reply-To: "Logistics Services" Mime-Version: 1.0 Content-Type: mult ipart/alternative;boundary="----------13972215105347E886BF43E"
mail() on [/var/www/pomoriemonasteryorg/components/com_xmap/2ktdz2.php:1]: To: bluesapphire89@yahoo.com -- Headers: From: "Priority Mail" X-Mailer: FastMailer/Webmail(versionSM/1.2.6) Reply-To: "Priority Mail" Mime-Version: 1.0 Content-Type: multipart/alternativ e;boundary="----------13972215105347E886C13F2"

On *Debian / Ubuntu Linux* to enable this logging, exec:

echo 'mail.add_x_header = On' >> /etc/php5/apache2/php.ini
echo 'mail.log = /var/log/phpmail.log' >> /etc/php5/apache2/php.ini

I find it useful to symlink */etc/php5/apache2/php.ini* to */etc/php.ini* its much easier to remember php location plus it is a standard location for many RPM based distros.

ln -sf /etc/php5/apache2/php.ini /etc/php.ini

Or another "Debian recommended way" to *enable mail.add_x_header logging* on Debian is via:

echo 'mail.add_x_header = On' >> /etc/php5/conf.d/mail.ini
echo 'mail.log = /var/log/phpmail.log' >> /etc/php5/conf.d/mail.ini

On Redhats *(RHEL, CentOS, SuSE) Linux* issue:

echo 'mail.add_x_header = On' >> /etc/php.ini
echo 'mail.log = /var/log/phpmail.log' >> /etc/php.ini

3. Restart Apache

On Debian / Ubuntu based linuces:

/etc/init.d/apache2 restart

P.S. Normally to restart Apache without interrupting client connections **graceful** option can be used, i.e. instead of restarting do:

/etc/init.d/apache2 graceful

On *RPM baed CentOS, Fedora* etc.:

/sbin/service httpd restart

or

apachectl graceful

4. Reading the log

To review in real time exact PHP scripts sending tons of spam tail it:

tail -f /var/log/phpmail.log

mail() on [/var/www/remote-admin/wp-includes/class-phpmailer.php:489]: To: theosfp813@hotmail.com -- Headers: Date: Mon, 14 Apr 2014 03:27:23 +0000 Return-Path: wordpress@remotesystemadministration.com From: WordPress Message-ID: X-Priority: 3 X-Mailer: PHPMailer (phpmailer.sourceforge.net) [version 2.0.4] MIME-Version: 1.0 Content-Transfer-Encoding: 8bit Content-Type: text/plain; charset="UTF-8"
*mail() on [/var/www/pomoriemonasteryorg/media/rsinstall_4de38d919da01/admin/js/tiny_mce/plugins/inlinepopups/skins/**.3a1a1c.php**:1]: To: 2070ccrabb@kiakom.net -- Headers: From: "Manager Elijah Castillo" X-Mailer: Mozilla/5.0 (Windows; U; Windows NT 5.0; es-ES; rv:1.9.1.7) Gecko/20100111 Thunderbird/3.0.1 Reply-To: "Manager Elijah Castillo" Mime-Version: 1.0 Content-Type: multipart/alternative;boundary="----------1397463670534B9A76017CC"*
*mail() on [/var/www/pomoriemonasteryorg/media/rsinstall_4de38d919da01/admin/js/tiny_mce/plugins/inlinepopups/skins/**.3a1a1c.php**:1]: To: 20wmwebinfo@schools.bedfordshire.gov.uk --*

*Headers: From: "Manager Justin Murphy" X-Mailer: Opera Mail/10.62 (Win32) Reply-To: "Manager Justin Murphy" Mime-Version: 1.0 Content-Type: multipart/alternative;boundary="----------1397463670534B9A7603ED6" mail() on [/var/www/pomoriemonasteryorg/media/rsinstall_4de38d919da01/admin/js/tiny_mce/plugins/inlinepopups/skins/.3a1a1c.php:1]: To: tynyrilak@yahoo.com -- Headers: From: "Manager Elijah Castillo" X-Mailer: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; pl; rv:1.9.1.9) Gecko/20100317 Thunderbird/3.0.4 Reply-To: "Manager Elijah Castillo" Mime-Version: 1.0 Content-Type: multipart/alternative;boundary="----------1397463670534B9A7606308" mail() on [**/var/www/pomoriemonasteryorg/media/rsinstall_4de38d919da01/admin/js/tiny_mce/plugins/inlinepopups/skins/.3a1a1c.php**:1]: To: 2112macdo1@armymail.mod.uk -- Headers: From: "Manager Justin Murphy" X-Mailer: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.5; pl; rv:1.9.1.9) Gecko/20100317 Thunderbird/3.0.4 Reply-To: "Manager Justin Murphy" Mime-Version: 1.0 Content-Type: multipart/alternative;boundary="----------1397463670534B9A76086D1"*

As you can see there is a junky spam mails sent via some spammer script uploaded under name ***.3a1a1c.php***, so to stop the dirty bastard, deleted the script:

rm -f ***/var/www/pomoriemonasteryorg/media/rsinstall_4de38d919da01/admin/js/tiny_mce/plugins/inlinepopups/skins/.3a1a1c.php***

It is generally useful to also *check (search) for all hidden .php files inside directoring storing multiple virtualhost websites*, as often a *weirdly named hidden .php* is sure indicator of either a *PHP Shell* script kiddie tool or a *spammer form*.

Here is **how to Find all Hidden Perl / PHP scripts inside /var/www**:

find . -iname '.*.php'
./blog/wp-content/plugins/fckeditor-for-wordpress-plugin/ckeditor/plugins/selection/.0b1910.php
./blog/wp-content/plugins/fckeditor-for-wordpress-plugin/filemanager/browser/default/.497a0c.php
./blog/wp-content/plugins/__MACOSX/feedburner_feedsmith_plugin_2.3/._FeedBurner_FeedSmith_Plugin.php

find . -iname '.*.pl*'

....

Reviewing complete list of all hidden files is also often useful to determine *shitty cracker stuff*

find . -iname ".*"

Debugging via /var/log/phpmail.log enablement is useful but is more recommended on development and staging (QA) environments. Having it enable on productive server with high amounts of mail sent via PHP scripts or just on **dedicated shared site server** could cause both performance issues, hard disk could quickly get and most importantly could be a severe security hole as information from PHP scripts could be potentially exposed to external parties.