

## **Vulnerability Assessment/Audit Report**

**Asset Tested:** <https://discounters.pk/>

**Author:** Ali Akber (Security Researcher & Application Security Engineer)

**Motive:** My intent in identifying these vulnerabilities is purely ethical and aimed at improving the security of your system. I have no intention of causing any harm to your company or its users. As a responsible and concerned individual, my decision to report these vulnerabilities immediately upon discovery reflects my commitment to your organization's security and the well-being of your users. I sincerely believe that open and transparent communication is the best way to address these issues and prevent any potential harm. During testing I tried my best to not affect any performance of your system or change any state or to abuse any data.

**Summary:** I am writing to report several vulnerabilities that I have identified in your web application, their impact on security & performance, how to mitigate them.

# Vulnerabilities

## 1. Title: Potential Backdoor by “blnmrpb” plugin

### Vulnerability details

The "blnmrpb" plugin may contain a potential backdoor that could be exploited by malicious actors to gain unauthorized access and maintain control over the system. **The exact nature of this backdoor requires further investigation as I was testing from the user end perspective so we can't confirm whether this plugin actually exist in your side (backend) or not.**

### Impact

If the potential backdoor is exploited, it could lead to unauthorized access, control, and maintenance of the system. This poses a significant risk to the security and integrity of the website, potentially resulting in data breaches or other malicious activities.

### Proof of Concept (PoC)

1. Go to this URL <https://discounters.pk/wp-content/plugins/blnmrpb/>
2. You may receive 403 Response “Not Authorized” but this backdoor can be allowed to certain IP Address that’s linked to Attackers so only Attackers can access it.

### Mitigation

1. Login via “Administrator” User & check the presence of this “Plugin” from plugin menu in WordPress dashboard if you find this “**blnmrpb**” plugin installed then immediately deactivate & delete it from the application.

### External Reference/Resources:

(You can check these external resources to learn more about this vulnerability or to measure its impact from recent attacks and how malicious attackers/hackers can take advantage of this vulnerability)

1. <https://labs.sucuri.net/webshell-in-fake-plugin-blnmrpb-directory/>

## 2. Title: Active Account's Username Leak

### Vulnerability details

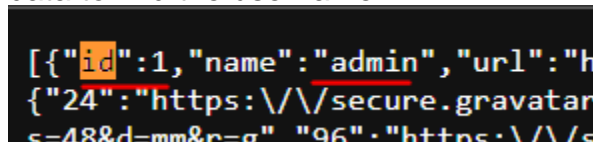
Valid WordPress user information, like in your website username **"admin"** and **"admin-sales"** are exposed and accessible.

### Impact

The exposure of valid WordPress user information poses a security risk, as malicious actors can leverage this data for launching other attacks such as "Guessing Password of that user" or "Brute-Forcing Password of users" to gain access to your website.

### Proof of Concept (PoC)

1. Visit the URL <https://discounters.pk/wp-json/wp/v2/users> and Search for string **"id"** or **"name"** (press f3 in keyboard to search for string) in the data to find the username.



```
[{"id":1,"name":"admin","url":"https://discounters.pk/wp-content/uploads/2023/12/placeholder.png"}, {"id":24,"name":"admin-sales","url":"https://discounters.pk/wp-content/uploads/2023/12/placeholder.png"}]
```

### Mitigation

1. To address this issue, measures should be taken to restrict access to the user information endpoint.

Use this code will hide the users list and give 404 as the result, while rest of the api calls keep running as they were.

```
add_filter( 'rest_endpoints', function( $endpoints ){
    if ( isset( $endpoints['/wp/v2/users'] ) ) {
        unset( $endpoints['/wp/v2/users'] );
    }
    if ( isset( $endpoints['/wp/v2/users/(?P<id>[\d]+)'] ) ) {
        unset( $endpoints['/wp/v2/users/(?P<id>[\d]+)'] );
    }
    return $endpoints;
});
```

### External Reference/Resources:

(You can check these external resources to learn more about this vulnerability or to measure its impact from recent attacks and how malicious attackers/hackers can take advantage of this vulnerability)

1. <https://hackerone.com/reports/356047>
2. <https://hackerone.com/reports/1784999>

## 3. Title: Debug file Disclosure Sensitive Information.

### Vulnerability details

Sensitive information, including the full path of the application on the server, local user details (**I found this user "discou56"**) and extensive information about plugins, themes, their versions, and local files, is disclosed.

### Impact

The exposure of sensitive information through the debug log file presents a serious security concern, allowing potential attackers to gather valuable information about the application, server, and local users this information can be used to launch further advanced attacks.

### Proof of Concept (PoC)

1. Visit the URL <https://discounters.pk/wp-content/debug.log> to access the debug log file containing sensitive information.

### Mitigation

1. To address the vulnerability associated with sensitive information disclosure, immediate action is required to disable debug logging in the production environment and prevent public/internet access to the debug log file.
2. Remove the debug.log file from the server.

#### **4. Title: Vulnerabilities in “Flatsome” theme.**

##### **Vulnerability details**

The Flatsome theme version 3.15.5 (currently running in your website) has been identified as susceptible to two critical vulnerabilities: Unauthenticated PHP Object Injection (versions < 3.17.6) and Reflected XSS (versions < 3.17.0).

##### **Impact**

Exploitation of these vulnerabilities could lead to remote access of your system and injection of malicious JavaScript code into your website, potentially compromising user credentials.

##### **Proof of Concept (PoC)**

1. You can check this theme version from your WordPress dashboard.

##### **Mitigation**

1. Upgrade the Flatsome theme to a version equal to or higher than 3.17.6

## 5. Title: DDOS & Internal Network Scanning Vulnerability.

### Vulnerability details

The "**xmlrpc.php**" file on your website is exposed, and it contains a method named "**pingback.ping**" that can be exploited by attackers. This vulnerability poses two significant risks:

- DDoS (Distributed Denial of Service): Attackers can use the "pingback.ping" method to send traffic from other sites to your website, causing your web application to slow down or go down completely.
- Internal Network Scanning: The same "pingback.ping" method can be exploited by attackers to scan your internal network and ports, potentially exposing sensitive information.

### Impact

The exposure of the "xmlrpc.php" file and the "pingback.ping" method poses a serious threat to the availability and security of your website. Taking immediate action to prevent public access to this file is essential to mitigate the risk of DDoS attacks and internal network scanning.

### Proof of Concept (PoC)

1. Visit the URL <https://discounters.pk/xmlrpc.php> to access the exposed "xmlrpc.php" file.

```
XML-RPC server accepts POST requests only.
```

Using following code attacker can send POST Request interact with your xmlrpc.php methods remotely.

```
POST /xmlrpc.php HTTP/1.1
Host: discounters.pk
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0)
Gecko/20100101 Firefox/70.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
```

```
Upgrade-Insecure-Requests: 1
Content-Length: 135
<methodCall>
<methodName>pingback.ping</methodName>
<params>
```

```
<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>
```

## Mitigation

1. To mitigate the risks associated with this vulnerability, it is crucial to prevent public user access to the "xmlrpc.php" file.

### Disabling XML-RPC in WordPress (Manually)

1. Open up your .htaccess file. You may have to turn on the 'show hidden files' within the file manager.
2. Paste the following code anywhere & save it to block all incoming xmlrpc.php requests.

```
<Files xmlrpc.php>
order deny,allow
deny from all
</Files>
```

### External Reference/Resources:

1. <https://hackerone.com/reports/325040>
2. <https://hackerone.com/reports/1619536>

## 6. Title: DDOS & Brute Force Vulnerability

### Vulnerability details

The "xmlrpc.php" file on your website is exposing methods such as "**system.multicall**," "**wp.getUserBlogs**," or "**metaWeblog.getUsersBlogs**" that can be exploited for brute force attacks on WordPress credentials. These methods enable attackers to send thousands of requests per second without rate-limiting or lockout mechanisms.

### Impact

This vulnerability poses a significant risk, allowing attackers to gain unauthorized access to WordPress user accounts (**Remember Attacker already have access of your WordPress username**) or potentially slow down your web application by flooding it with requests without any rate limit.

### Proof of Concept (PoC)

1. Visit the URL <https://discounters.pk/xmlrpc.php> to access the exposed "xmlrpc.php" file containing the vulnerable methods.

### Mitigation

1. To mitigate the risks associated with this vulnerability, it is crucial to prevent public user access to the "xmlrpc.php" file.  
**Use previous mitigation technique to mitigate this vulnerability**

### External Reference/Resources:

1. <https://hackerone.com/reports/1147449>