RIPS源码精读(一):逻辑流程及lib文件夹大致说明

梅子酒m3i / 2018-07-30 21:01:43 / 浏览数 4477 技术文章 技术文章 顶(0) 踩(0)

很早就有深入分析学习一款源代码审计工具的想法，在查找rips源码分析相关资料时，发现相关的学习分析资料较少，于是选择rips作为该系列文章的分析对象，因为没有
因为我是第一次将具体的分析写下来，并且本身的技术能力问题，在某些场景下的用语或者技术细节描述可能存在偏差，请师傅们包涵。

引言

RIPS是一个源代码分析工具，它使用了静态分析技术，能够自动化地挖掘PHP源代码潜在的安全漏洞

本篇内容

作为本系列文章的开始，只介绍rips的逻辑流程以及lib文件夹下各文件大致内容分析，不具体分析代码审计的细节，相关细节在之后的文章中分析

整体结构

RIPS工具的整体架构如下:

```
+-- CHANGELOG [file]
+-- config [dir]
|   +-- general.php
|   +-- help.php
|   +-- info.php
|   +-- securing.php
|   +-- sinks.php
|   +-- sources.php
|   +-- tokens.php
+-- css [dir]
|   +-- ayti.css
|   +-- barf.css
|   +-- code-dark.css
|   +-- espresso.css
|   +-- notepad++.css
|   +-- phps.css
|   +-- print.css
|   +-- rips.css
|   +-- rips.png
|   +-- scanning.gif
|   +-- term.css
|   +-- twlight.css
+-- index.php [file]
+-- js [dir]
|   +-- exploit.js
|   +-- hotpatch.js
|   +-- netron.js
|   +-- script.js
+-- lib [dir]
|   +-- analyzer.php
|   +-- constructer.php
|   +-- filer.php
|   +-- printer.php
|   +-- scanner.php
|   +-- searcher.php
|   +-- tokenizer.php
+-- LICENSE [file]
+-- main.php [file]
+-- README.md [file]
+-- windows [dir]
|   +-- code.php
|   +-- exploit.php
|   +-- function.php
|   +-- help.php
|   +-- hotpatch.php
|   +-- leakscan.php
```

config■■:■■■■■■■■

css■■:■■css■■■■

js■■:■■js■■■■

lib■■:rips■■■■■■■■

window:rips■■■■■■

lib文件夹说明

lib文件夹存放rips运行的核心文件,定义了大量函数以及类用以完成rips完整的代码分析功能

1. analyzer.php

   仅定义了Analyzer类,并在类中定义了三个函数，分别是get_tokens_value,get_var_value,getBraceEnd,Analyzer类主要用以根据token信息分析文件信息

2. constructer.php

   本文件定义了五个类,分别为VarDeclare、VulnBlock、VulnTreeNode、InfoTreeNode、FunctionDeclare,分别用以■■■■、■■■■、■■■■■■■■、■■■

3. filer.php

   仅定义了函数read_recursiv,用以遍历文件夹下的文件信息

4. printer.php

   定义大量函数,基本都是用于将分析得到的结果输出至前端页面

5. scanner.php

   仅定义了scanner类,类中包含大量方法,本文件为rips分析操作的核心文件，包括token处理、字段处理等功能

6. searcher.php

   仅定义了searchFile函数,主要用于根据token信息分析漏洞情况，并使用VulnTreeNode类加以实例化

7. tokenizer.php

   仅定义了Tokenizer类,类中定义大量函数,其余文件中所用到的token信息均来源于此文件

index.php 分析

index.php是rips项目的入口文件，因此我放在了第一个分析。

代码构成主要是前端文件，功能方面主要将目标路径、扫描类型等参数发送至分析模块。

在index.php的112行附近，触发点击事件，进入main.php

main.php分析

main.php是整个rips代码分析的开始部分

配置文件引入:

```
<?php
   include('config/general.php');          // ■■■■■■■■■■■■■
   include('config/sources.php');          // ■■■■■■■■■■■■■■■■■■■■■$_GET■file_get_contents()
   include('config/tokens.php');           // ■■■■■■■■■■token■■■■■■■■■
   include('config/securing.php');         // ■■■■■■■■■■■■■■■■■■■■ htmlspecialchars ■■■■■■■ $F_SECURING_XSS
   include('config/sinks.php');            // ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
   include('config/info.php');             // ■■■■■■■■■■■■■■■■sqlite_open()=>'using DBMS SQLite'
```

核心代码引入:

```
include('lib/constructer.php');         // ■■■
   include('lib/filer.php');               // ■■■■■■■■■■■■■■■■■■■■■■■■
```

```
   include('lib/tokenizer.php');           // prepare and fix token list
   include('lib/analyzer.php');            // string analyzers
   include('lib/scanner.php');             // provides class for scan
   include('lib/printer.php');             // output scan result
   include('lib/searcher.php');            // search functions
```

结束引用部分，进入main.php文件的逻辑处理部分

文件路径传入

对传入的路径参数进行处理，如果传入参数为目录，则递归目录的文件信息，并赋值入变量，如果为单个文件，则只记录该文件

```
if(!empty($_POST['loc']))
   {
       $location = realpath($_POST['loc']);

       if(is_dir($location))
       {
           $scan_subdirs = isset($_POST['subdirs']) ? $_POST['subdirs'] : false;
           $files = read_recursiv($location, $scan_subdirs);

           if(count($files) > WARNFILES && !isset($_POST['ignore_warning']))
               die('warning:'.count($files));
       }
       else if(is_file($location) && in_array(substr($location, strrpos($location, '.')), $FILETYPES))
       {
           $files[0] = $location;
       }
       else
       {
           $files = array();
       }
```

初始化扫描功能

首先对各种参数进行初始化赋值，如各类计数变量等，随后根据传来的verbosity变量，即"漏洞类型"参数，对scan_functions数组进行赋值

```
if(empty($_POST['search']))
       {
           $user_functions = array();
           $user_functions_offset = array();
           $user_input = array();

           $file_sinks_count = array();
           $count_xss=$count_sqli=$count_fr=$count_fa=$count_fi=$count_exec=$count_code=$count_eval=$count_xpath=$count_ldap=$

           $verbosity = isset($_POST['verbosity']) ? $_POST['verbosity'] : 1;
           $scan_functions = array();
           $info_functions = Info::$F_INTEREST;
           if($verbosity != 5)
           {
               switch($_POST['vector'])  //■■■■■■■
               {
                   //XSS
                   case 'xss':         $scan_functions = $F_XSS;           break;
                   //header
                   case 'httpheader':  $scan_functions = $F_HTTP_HEADER;   break;
                   //session
                   case 'fixation':    $scan_functions = $F_SESSION_FIXATION;  break;
                   //■■■■■
                   case 'code':        $scan_functions = $F_CODE;          break;
                   //■■
                   case 'ri':          $scan_functions = $F_REFLECTION;    break;
                   //■■■■
                   case 'file_read':   $scan_functions = $F_FILE_READ;     break;
                   //■■■■■■■■
                   case 'file_affect': $scan_functions = $F_FILE_AFFECT;   break;
                   //■■■■
                   case 'file_include':$scan_functions = $F_FILE_INCLUDE;  break;
                   //■■■■
```

```
        case 'exec':        $scan_functions = $F_EXEC;          break;
        //■■SQL
        case 'database':    $scan_functions = $F_DATABASE;      break;
        //XPATH■■
        case 'xpath':       $scan_functions = $F_XPATH;         break;
        //LDAP■■
        case 'ldap':        $scan_functions = $F_LDAP;          break;
        //■■■■
        case 'connect':     $scan_functions = $F_CONNECT;       break;
        //■■■■■■■■■■
        case 'other':       $scan_functions = $F_OTHER;         break;
        //POP■
        case 'unserialize': {
                            $scan_functions = $F_POP;
                            $info_functions = Info::$F_INTEREST_POP;
                            $source_functions = array('unserialize');
                            $verbosity = 2;
                            }
                            break;
        //■■■
        case 'client':
            $scan_functions = array_merge(
                $F_XSS,
                $F_HTTP_HEADER,
                $F_SESSION_FIXATION
            );
            break;
        //■■■
        case 'server':
            $scan_functions = array_merge(
                $F_CODE,
                $F_REFLECTION,
                $F_FILE_READ,
                $F_FILE_AFFECT,
                $F_FILE_INCLUDE,
                $F_EXEC,
                $F_DATABASE,
                $F_XPATH,
                $F_LDAP,
                $F_CONNECT,
                $F_POP,
                $F_OTHER
            ); break;
        //■■■■
        case 'all':
        default:
            $scan_functions = array_merge(
                $F_XSS,
                $F_HTTP_HEADER,
                $F_SESSION_FIXATION,
                $F_CODE,
                $F_REFLECTION,
                $F_FILE_READ,
                $F_FILE_AFFECT,
                $F_FILE_INCLUDE,
                $F_EXEC,
                $F_DATABASE,
                $F_XPATH,
                $F_LDAP,
                $F_CONNECT,
                $F_POP,
                $F_OTHER
            ); break;
    }
}
if($_POST['vector'] !== 'unserialize')
{
    $source_functions = Sources::$F_OTHER_INPUT;
    // add file and database functions as tainting functions
```

```
            if( $verbosity > 1 && $verbosity < 5 )
            {
                $source_functions = array_merge(Sources::$F_OTHER_INPUT, Sources::$F_FILE_INPUT, Sources::$F_DATABASE_INPUT
            }
        }
```

代码审计及结果输出

Scanner类在171行附近进行实例化，并进行词法分析，输出结果至前端

```
$scan = new Scanner($file_scanning, $scan_functions, $info_functions, $source_functions);
            $scan->parse();
            $scanned_files[$file_scanning] = $scan->inc_map;
```

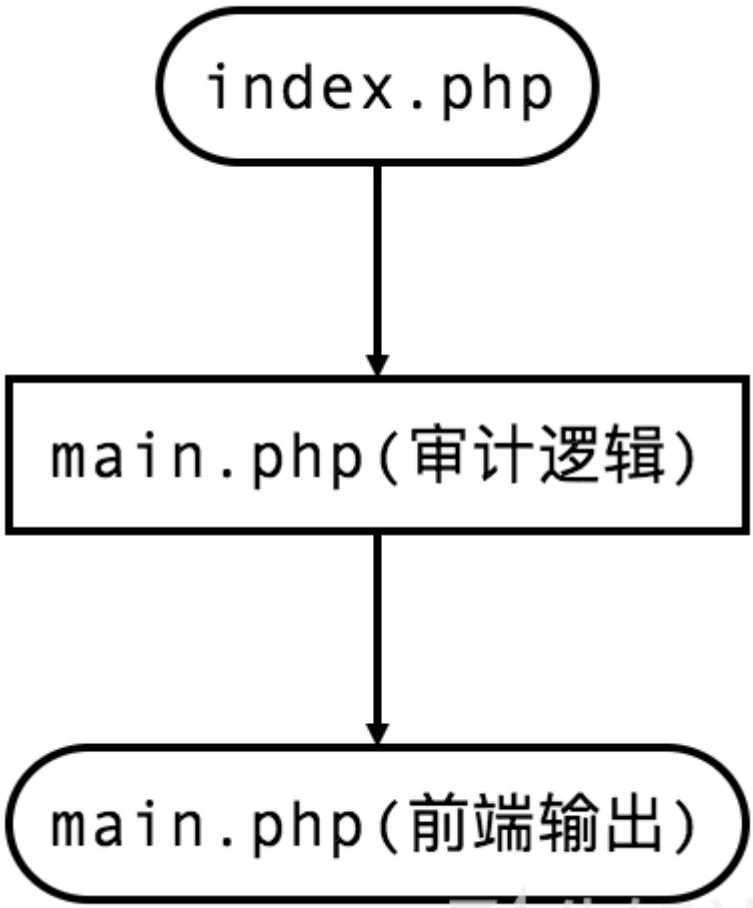
总结

流程部分总结

```
st=>start: index.php
op=>operation: main.php(■■■■)
e=>end: main.php(■■■■)



st->op->e
```



梅子酒m3i 2018-07-30 21:02:46

尴尬...最后的流程图部分，先知的markdown貌似并不支持解析

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板