

部分白盒黑盒技术经验分享

——web安全沙龙新浪站

张建伟

只有
开放共享
才能最终推动整个行业
达成安全目标



SPARK

SECSAY

Spark

P.Z

Overflow

团队Xsser安全问题

白盒安全

黑盒安全

任务管理

未来期望

XSSER

OVERFLOW

P.Z

secsay团队介绍

- Spark
 - 北京，生物技术专业，工作第四年
 - 早恋晚熟男
- PZ
 - 杭州，Flash、XSS 多个蠕虫作者，大三
 - 未婚 未婚 未婚 未婚~， 188cm
- Xsser
 - 杭州，90后，他不是黑客，高五
 - 一直被世俗打击打击打击打击~， 180cm
- Overflow
 - 纽约，WAP蠕虫缔造者，纽约大学
 - 俊男，俊男，俊男，纯情俊男！， 185cm

安全问题

拒绝服务

数据泄露

数据污染

团队介绍

白盒安全

黑盒安全

任务管理

未来期望

数据污染

数据泄露

拒绝服务

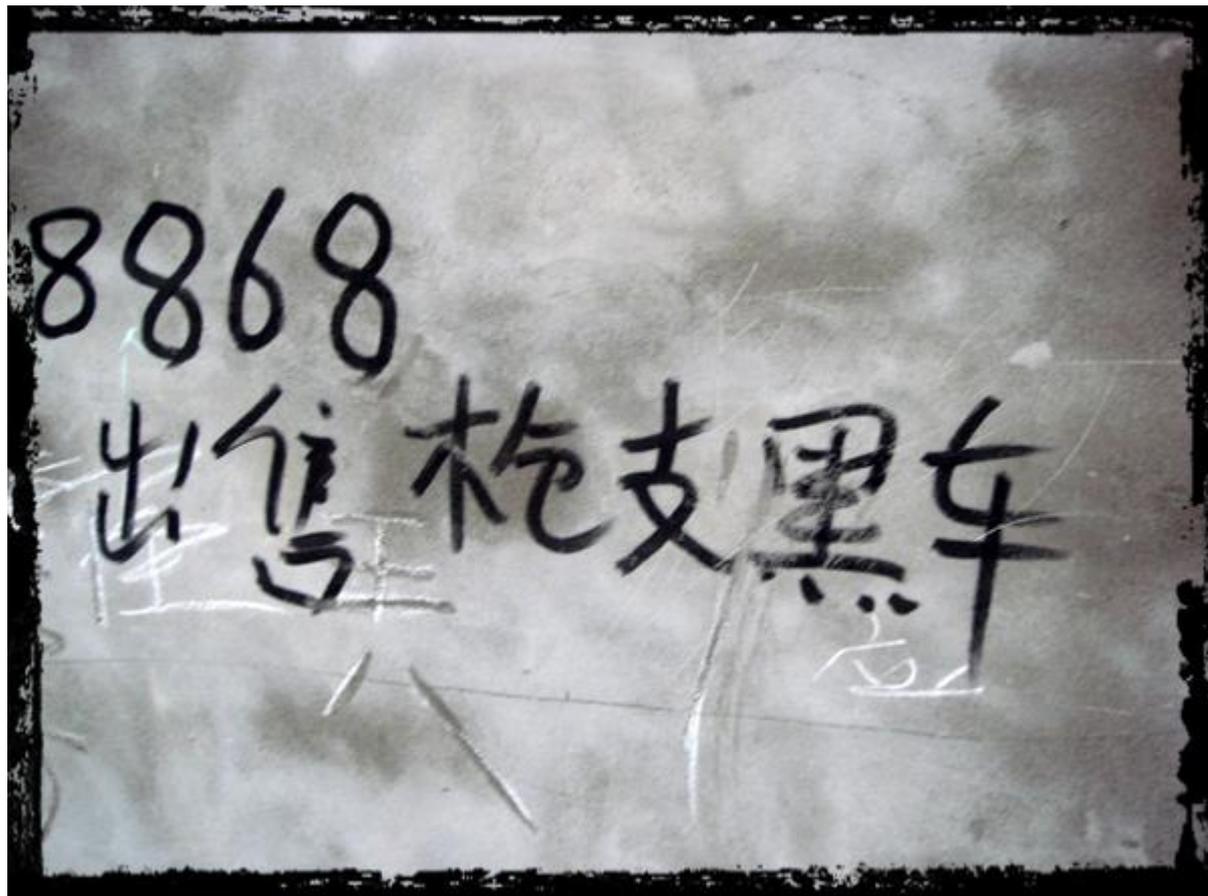
- 游戏玩家：
 - 黑客
 - 政府
 - 安全厂商
 - 企业自身
- 不中断服务是所有一切话题的前提

数据泄露

- 帐号密码
 - 盗号
 - 拖库
 - 诈骗
 - 其他
- 个人隐私
 - 资料
 - 消费
 - 情感
 - 交易

数据污染

- 垃圾注册
 - 验证码破解
- 群发广告
- 牛皮癣
- 敏感词
- 色情图片
- 侵权内容
- 各种多媒体格式



白盒安全

常用手段
语义分析
程序实现

团队介绍

安全问题

黑盒安全

任务管理

协作响应

词法、语法和语义

- 词法：词的构成方法
 - 哪些是词，哪些是名词哪些是动词
 - 哪些是空格，标识符，加减乘除符号等等
- 语法：语言的规则或法则
 - 定义主谓宾结构，定义复句等
 - 定义语句块，函数调用的写法，参数写法
- 语义：想要表达的是什么
 - 将一个用户输入的字符串当作代码来解释执行
 - Eval(\$_POST[cmd])

各种代码分析对比

- 关键字查找
 - 查找特定单词
 - 属于词法维度上的查找
- 高级正则表达式
 - 有一定上下文关系
 - 能到达特定语法查找的纬度
- 语义分析
 - 查找某种模式的代码
 - 语义纬度

PHPScan方法论

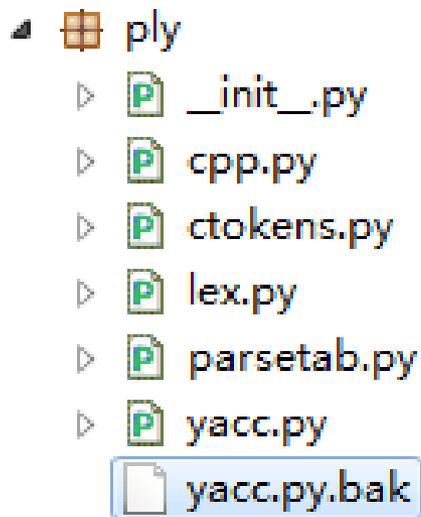
- 基本原理：
 - 基于语义分析
 - 黑名单方式找出危险语义
- 黑名单语义定义
 - 默认认为用户输入为“有害”
 - 有害信息不许成为主动调用方
 - 不能做函数
 - 不能做对象
 - 有害信息不能当作危险调用方式的参数
 - 不能成为eval参数
 - 不能成为system参数

Lex 和 Yacc

- Lexical Analyzar
 - 是一种生成扫描器的工具。
 - 扫描器是一种识别文本中的词汇模式的程序。
- Yet Another Compiler Compiler
 - 输入是巴科斯范式（BNF）表达的语法规则以及语法规约的处理代码
 - 输出的是基于表驱动的编译器，包含输入的语法规约的处理代码部分。

PLY: Lex&Yacc的Python实现

- Easy install 或者直接下载源码安装
 - 可以作为系统的库
 - 可以打包到程序目录



PHPar: PHP Parse

- `phplex.py`
 - 定义词法
- `Phpparse.py`
 - 定义语法
 - 解析语法
- `Phpast.py`
 - 生成抽象语法树

- ▾  `phpar`
 -  `ply`
 -  `__init__.py`
 -  `colortools.py`
 -  `phpast.py`
 -  `phplex.py`
 -  `phpparse.py`

抽象语法树

- 类型化
- 查询
- 遍历

```
def query(self, q="*", filter=None):
    '''查询的语法如下 [type|*] (>[type|*])
    eg. fdef 类型为fdef 的子结点
        fdef>vdecl 类型为fdef的子结点下面的类型为vdecl的子结点
        *>exp 第二层的exp结点
        **>exp 所有类型为exp 的结点。(不管层次)
        **>?所有叶结点
        ? 表示叶结点
    filter:过滤器函数
    '''
    if filter == None:
        filter=self.filter
    else:
        filter = filter
    ret = []
    qs = q.split(">")
    if self.is_type_match(qs[0]) or qs[0] == '*':
        r = apply(filter, (self,))
        ret.append(self)
        print 'append',self
    if qs[0] == '**' and self.is_type_match(qs[1]):
        ret.append(self)
```

例子:对反引号的解析支持

- <http://blog.renren.com/blog/bp/Qm3aTzRcce>

```
BackHole Type:eval_b Desc:eval a variable  
-->Code: ` $x ` ; line:27  
in file:d:\office\test6.php
```

- 见附件《phpar中增加反引号的解析.pdf》

变量检查：安全扫描

```
def varcheck(self, item):  
    if hasattr(item, 'query'):  
        vars = item.query('**>Variable')  
        for v in vars:  
            if v.name in self.dangevars:  
                v.safe = False  
            if '$_' in str(v.name):  
                self.dangeit(v)  
  
        ass = item.query('**>Assignment') #Variable  
        for a in ass:  
            if isinstance(a.node, Variable):  
                if not a.node.safe:  
                    self.dangeit(a.node)  
                if hasattr(a.expr, 'query'):  
                    vs = a.expr.query('**>Variable')  
                    for v in vs:  
                        if not v.safe:  
                            self.dangeit(a.node)
```

特定语义查找：插件A的实现

```
def cb_func_a(node):
    if node.name in dangefuncs:
        for param in node.paramees:
            print 'parm:',param
            if hasattr(param, 'query'):
                result = param.query('**>Variable')
                safe = True
                for v in result:
                    if hasattr(v, 'safe') and not v.safe:
                        safe = False
                if not safe:
                    return node
    if hasattr(node.name, 'query'):
        result = node.name.query('**>Variable')
        for v in result:
            if not v.safe:
                return node

queryString = r'**>FunctionCall'
filter = cb_func_a
desc = """userinput func name or dange func with dange var params"""
```

各种代码分析对比

- 词法维度：关键字查找
 - 硬编码的查找
 - 简单正则表达式
- 语法纬度：高级正则表达式
 - 有一定上下文关系
 - 例子
- 语义纬度：
 - 可以对特定模式和意义的代码处理
 - 更加高级的代码分析工具

各种代码分析对比

```
match = re.search(r' (?P<output>(request\.setAttribute|addModel))\s*\'  
+' ((?P<params>.*?) (?P<keyparam>(getParameter|request\.)))', file_contents,  
if match:↓  
    params = match.group('params')↓  
    if 'antispam' in str(params).lower() or 'safe' in str(params).lower():  
        iname += 1↓  
        continue↓  
    output = match.group('output')↓  
    keyparam = match.group('keyparam')↓  
    if iname == 0:↓  
        info = '\n[%s] :\n' % (file)↓  
    else:↓  
        info = ''↓  
    info += '\t|-- output:%s with parameters [%s] in line [%d] \n' % (output,  
    info += '\t\tcode:%s\n' % file_contents.strip()[:100]↓
```

SPARK

能干什么？

- 日常后门扫描中的辅助
 - 公开的版本功能不全面
 - 关键字查找后门的工作量其实已经可以接受
 - 静态后门检测永无止境
- 高级代码分析
 - XSS、SQL注入等漏洞检测
 - 高级代码搜索
- 这是个重复造出来的轮子
 - 目前轮子转的还不好
 - 但是这个轮子结构非常简单

黑盒安全

扫描器

安全测试工具

XCapture
XBrowser

团队介绍

安全问题

白盒安全

任务管理

未来期望

XBL0M26L

XC9bfn16

系统测试工具

扫描器 VS 安全测试工具

- 扫描器
 - 功能强大且全面
 - 使用复杂
 - 价格不菲
 - 定制性低
 - 不适合在SDL流程中使用
- 安全测试工具
 - 功能相对单一
 - 使用简单
 - 自动化控制方便
 - 适合在SDL流程中使用

XNMD. XCapture特点

- 基于代理
 - Fiddler core
 - Inception方式
 - 输出格式方便在Fiddler中分析
- XSS检测
 - 替换HTTP中的参数
 - 暴力重放
 - 发送特定关键字
 - 检测返回值
- 用户体验
 - 面向非安全人员
 - 面向非技术人员

XNMD. XCapture实现

- 现场调试代码

XNMD. XBrowser

- 对XCapture的补充
 - 解耦合
- 爬虫
 - 这个没什么好讲的
- 自动表单提交
 - 分析页面表单
 - 填写表单提交
 - 等待XCapture捕获后重放测试

团队介绍

安全问题

白盒安全

黑盒安全

任务管理

Crontab

CI系统

Hudson

未来期望

Hudson

CI系统

任务管理

- Crontab
 - `*/20 6-12 * 12 * /usr/bin/backup`
 - 12 月内, 每天早上 6 点到 12 点中, 每隔 20 分钟执行一次 `/usr/bin/backup`
- 其他系统
 - 这个大家分享一下
- CI系统

CI系统：持续集成系统

- 代码持续集成
 - 定时从代码库更新代码
 - 定时或者从其他任务触发build任务
 - 定制化Report
 - 其他插件
- 一些CI系统
 - CruiseControl
 - LuntBuild
 - TeamCity
 - AntHill Pro
 - Hudson

Hudson

- Java开发，跨平台
- 支持分布式任务节点
- 安装和操作都傻瓜化
- 插件丰富

The screenshot shows the Hudson web interface for a project named 'scan-code'. The page has a blue header with the 'Hudson' logo. Below the header, the breadcrumb 'Hudson » scan-code' is visible. A vertical menu on the left contains several icons and links: a green arrow for '返回' (Return), a magnifying glass for '状态' (Status), a notepad for '变更集' (Changeset), a folder for '工作区' (Workspace), a play button for '立即构建' (Build Now), a red prohibition sign for '删除Project' (Delete Project), and a wrench for '设置' (Settings). Below this menu is a 'Build History' section with a gear icon and a '(趋势图)' (Trend Chart) link. It lists three builds: #11 on 2011-1-19 13:07:09, #10 on 2011-1-18 13:06:29, and #9 on 2011-1-17 13:08:29. On the right side, the title 'Project scan-code' is displayed above the text '代码扫描' (Code Scan). Below this are two icons: a folder for '工作区' (Workspace) and a notepad for '最近变更集' (Recent Changeset). At the bottom right, under the heading '上游项目' (Upstream Project), there is a blue circle icon and the link 'code-ugc-update'.

Hudson

- 对白盒的意义
 - 内部集成SVN CVS等代码库管理工具
 - Update完代码后可以触发其他任务
- 支持邮件、IM报警
 - 成功失败等等定制化很任意
- 对黑盒的意义
 - Web化的Crontab
 - 支持bat，甚至python等众多脚本

团队介绍

安全问题

白盒安全

黑盒安全

任务管理

协作响应

未来期望

协作打击

开放共享

开放共享

协作打击

未来期望

- 对黑色、灰色产业链的协作打击
 - 包括但不限于钓鱼、木马、入侵等恶势力
 - 对地下经济进行曝光来推动法制健全
 - 提高企业对安全认识程度
- 开放共享推动技术体系发展和促进整个行业发展
 - 形成行业通用标准和通用技术平台
 - 在恶意IP恶意URL方面更加深入的合作

谢 谢 ！