CTF训练

原创

两个白杯子 于 2019-01-11 22:26:12 发布 905 收藏 10

国外 we challenge———最经典
http://www.wechall.net/

西普学院
http://www.simplexue.com/
有在线视频教程

http://ctf.idf.cn/
IDF实验室 CTF训练营
有ctf资料，工具，知识

一个在线的ctf game————————值得马上去做~
http://fun.coolshell.cn/first.html
网上有writeup http://drops.wooyun.org/tips/2730

hackyou2014,有writeup
http://hackyou.ctf.su/
http://drops.wooyun.org/tips/870

wargames
http://overthewire.org/wargames

exploit linux的漏洞利用 通关版，有在线的资源，有虚拟机镜像，在网盘里，有writeup
https://exploit-exercises.com/protostar/
可以通过它学习提权,漏洞分析,漏洞开发,逆向工程等
相关介绍
http://www.myhack58.com/Article/html/3/8/2012/36180.htm

国外的一个在线训练平台
http://securityoverride.org/news.php

菜鸟级别的ctf
http://www.try2hack.nl/levels/
发表在 CTF | 一条评论
WAF防御能力评测及工具
发表于 2015 年 3 月 7 日 由 YouMeng
转自：http://danqingdani.blog.163.com/blog/static/18609419520141014627234 70/

本篇文章介绍如何从常规攻击的防御能力来评测一款WAF。一共覆盖了十六种攻击类型，每种类型均从利用场景（攻击操作的目的），注入点（漏洞产生的地方，比如说大多数WAF都会较全面地覆盖来自GET请求的攻击，有选择地覆盖来自POST请求的攻击而忽略来自请求头的攻击）和绕过方式来评测，最后附上评测代码。

一、SQL注入（注入）

1. 利用场景
   从攻击者进行SQL注入的阶段来看，一般分为探测与攻击两个阶段（p.s.攻击阶段是WAF的重点防御阶段）
   （1）探测阶段
   1) 探测是否存在SQL注入：基于SQL错误回显(e.g.extractvalue) 或时间响应(Benchmark,sleep)来探测目标网站是否存在SQL注入点
   2) 识别数据库类型：根据数据库的slang来判断目标网站采用的哪种数据库及数据库的版本等基本信息，例如@@version,user()
   （2）利用阶段
   i. select型SQLi
   1) 读取元数据库：通过读取数据库管理系统元数据库（e.g. MySQL的information_schema, SQL Server的sysobjects）来探测数据存储数据库，表，列等基本信息
   2) 提取数据：使用union查询或盲注中的逐位探测(e.g.length,substr)或者条件探测(Select if(1=1,'a','b');)来提取数据库管理系统中的数据，其中经常会用到concat(),group_concat()等函数,
   3) 读取系统文件：读取数据库管理系统所在操作系统中的重要系统文件(eg. MySQL的load_file)
   4) 写入系统文件：向数据库管理系统所在操作系统写入后门文件(e.g.MySQL的select into outfile)
   5) 执行系统命令：以数据库管理系统为跳板执行系统命令(e.g.SQL Server的exec master…xp_cmdshell)
   ii. update型SQLi
   iii. insert型SQLi

2.注入点
记住：任何输入都是有害的
（1）GET QueryString
（2）POST
（3）Referer
（4）Cookie
（5）X_Forwarded_For
（6）UserAgent
（7）Basic-Authorization

3.绕过方式
这里的绕过主要是针对采取模式匹配来识别攻击的WAF检测方法（其实大多数WAF的基本方法都是这个，引申到IDS,IPS,anti-virus等安全产品都是采取的这种方法）

采取模式匹配的检测方法会绕过的原因无外乎以下几种：
1）HTTP协议解析漏洞：WAF进行模式匹配的时候都是对HTTP协议变量进行匹配，攻击者构造异常的HTTP数据包导致不能正常提取变量，都不能进入到模式匹配阶段，自然而然就绕过了

2）模式匹配的先天不良：字符串匹配，无论是简单的正则匹配或者会配合一定的逻辑的匹配（比如对不同关键字进行加权操作，前后依赖关系判断）反正都逃不开模式两个字，而模式是固定的，就导致了各种侧漏。

对于第二点，在云WAF上的问题最为严重，我们知道云WAF的用户类型是多样化的（不同的搭建技术PHP/ASP/JSP，运行环境Windows/Linux，访问方式PC/Mobile），理想状态下，应该按站点类型精准投放防御规则，但是..基于站点自动建模（安全人员中缺乏数据分析师）是个"前沿"的技术活，而免费模式下是产生不了多大动力来投入精力的，所以现实是倾向于选择更通用的方式（放弃少数人）按危害优先级来定制规则。

以上绕过原因衍生了以下的通用绕过方式
（1）参数污染
（2）URL重写

http://localhost/uyg/id/123+or+1=1/tp/456

（3) 加密payload
MD5、SHA-1、自定义加密
（4）缓冲区溢出
（5）编码绕过
（6）特殊字符插入（%00）
（7）异常HTTP请求包（e.g.超级大，不符合HTTP规范但被server容错的数据包）
（8）数据包分块传输方式Transfer-Encoding: chunked

SQL注入一般存在以下绕过方式（其实也就是模式匹配的先天不良）
1) 编码绕过：通过对SQL注入攻击Payload进行Unicode编码，十六进制编码，双URL编码来绕过检测规则
2) 注释语句绕过：通过在SQL注入攻击Payload中插入注释语句（内联注释)来绕过检测规则
3)大小写绕过：通过变化SQL注入攻击Payload的大小写来绕过检测规则
4) 类型转换绕过：使用hex, ascii, ord, char，chr，cast，convert等类型转换函数来变化特定字符来绕过检测规则，除了类型转换函数还有隐性类型转换的特征 http://danqingdani.blog.163.com/blog/static/18609419520133185493818 2/
5) 生僻的SQL关键字绕过
6)特殊的sql语法（e.g. mysql . ~ ! + – 符号）
7)关键字拆分
8)请求方式转换（将GET转变为POST，因为误报的问题POST的规则要远远比GET规则松）
（参考了Seay大神总结的绕过方式）

二、文件包含（文件操作）
攻击的核心目标之一是信息操纵，而信息的载体就是文件（数据），对文件的非法读、写、删除等操作就成为防御的核心。

1.利用场景
（1）包含本地文件
本地文件包含的出发点一般分为两种

a. 读取系统文件获取敏感信息，例如配置文件
除了读取同目录下的文件外，一般会配合目录遍历

b. 实施代码执行
（1）包含（执行）存在后门的文件（写入后门的方法有很多，例如SQLI写马/文件上传写马/代码注入）
（2）包含（执行）系统可执行文件
（2）使用php://input 协议将文件包含漏洞变成代码执行漏洞
http://danqingdani.blog.163.com/blog/static/1860941952013993810261/
（2）包含远程文件

2.注入点
记住：任何输入都是有害的
（1）GET QueryString
（2）POST

3.绕过方式
绕过目录遍历检测（其实目录遍历因该单独列出来，在命令执行等地方都会用到）
1) 编码绕过：
b. %c0%af Apache，Tomcat UTF-8编码错误
c. %25%5c Unicode漏洞
d. %c0%af
e. %c1%9c Unicode漏洞
f. %fc%80%80%80%80%af Unicode漏洞
2) 截断: %00

读取系统文件时的绕过检测方法
1) 使用php://filter协议以base64编码方式读取指定文件内容
2)使用data:// URI schema协议执行系统命令

三、文件上传/下载（文件操作）

1.利用场景
（1）直接上传webshell文件
一般的文件上传模块，都会配置文件上传白名单（e.g.只允许上传图片文件)，所以这种攻击方式的一般看是否有白名单以及如何绕过白名单

webshell的类型如下：
1> asp shell
2> php shell
3> jsp shell
4> python shell
5> pl-cgi shell
6> sh shell
7> c shell
8> cfm shell
9> exe shell

（2）图片写马上传
在文件名无法做文章的情况下，一般会配合服务器解析漏洞或者文件包含漏洞来利用

（3）下载任意文件
处理用户请求的时候允许用户提交文件路径，攻击者通过变化目录或文件地址来达到下载系统敏感文件的目的

2.注入点
文件上传表单

3.绕过
文件名白名单绕过
（1）利用上传文件请求的解析漏洞，e.g.不能正常提取文件名
（2）配合服务器解析漏洞，构造奇怪的文件名绕过白名单

服务器解析漏洞
（1）Apache解析漏洞
xxx.php.jpg
xxx.php.rar
xxx.php.x1.x2.x3
xxx.php. (windows下点和空格都会被自动消除）

（2）Nginx解析漏洞
xxx.jpg%00.php
xxx.jpg/a.php

（3）IIS解析漏洞
xxx.asp;.jpg xxx.asa;.jpg xxx.cer;.jpg xxx.cdx;.jpg
xxx.asp:.jpg xxx.asa:.jpg xxx.cer:.jpg xxx.cdx:.jpg
xxx.asp/xx.jpg xxx.asa/xx.jpg xxx.cer/xx.jpg xxx.cdx/xx.jpg

参考 http://drops.wooyun.org/papers/539

四、命令执行 （注入）

1.利用场景
输入点接收并运行系统命令

2.注入点
（1）POST
（2）GET
（3）Cookie

五、代码执行 （注入）

1.利用场景
（1）输入点接收并运行PHP/JSP/ASP代码

2.注入点
（1）POST
（2）GET
（3）Cookie

六、webshell (这个分类有点纠结，主要是从已经被种马的情况下看能否拦截）
1.利用场景
配合文件上传、代码执行，SQLI写马等操作写入webshell后进行webshell连接

webshell按传递payload来分类的
（1）payload采用请求头提交，以cookie提交最多，其中采取自定义请求头更隐蔽
（2）payload采用POST提交

2.注入点
（1）GET QueryString
（2）POST
（3）Cookie
（4）其他请求头

3.绕过方式
webshell payload提交的时候一般都会加密

以下列出常见的webshell，可以探测一下这些基本的webshell WAF是否能拦截
caidao一句话连接客户端
Lanker微型php 后门客户 2.0正式版 一句话连接客户端
weevely php后门生成工具及客户端端
webacco php后门生成工具及客户端端
phpspy.php
b374k.php
80sec.php
90sec.php
r57.php
c99.php
b4che10r
X14ob-Sh3ll
aspxspy
server_sync.php (phpadmin后门文件）

七、XSS
1. 利用场景
从攻击者进行 XSS注的阶段来看，一般分为探测与攻击两个阶段。
探测阶段是指弹框测试 xss是否存在，常用alert(),prompt(),confirm()等函数（弹框只是一种，也有看 html标签是否能实际运行）；
攻击阶段是指在确认存在 XSS后，进行利用，例如盲打盗取 cookie(session hijacking)伪造用户身份登陆，蠕虫传播，keylogger,下载安装恶意软件，构造钓鱼页面
（xss我了解的太少，xss好复杂滴说，按成因划分有反射型xss，存储型xss，DOM型XSS，mXSS（突变型XSS），UXSS（通用型XSS），按平台划分还有flash xss，xss和sqli的攻击方式都能出成一本书了）

2.注入点
记住：任何输入都是有害的
（1）GET QueryString
（2）POST
（3）Cookie

3.绕过方式
1）编码绕过（url unicode编码，base64编码）
2）使用Data URI schema绕过
3）使用Javascript伪协议
4）基于事件函数绕过
5）类型转换绕过引号e.g.String.fromCharCode
6）xss payload在锚点后提交 https://www.securusglobal.com/community/2014/10/13/bypassing-wafs-with-svg/

八、CSRF
（其实CSRF更像一种攻击模式而不仅仅是漏洞类型）
1.攻击场景
由于http请求无状态，而服务器仅仅依赖于用户浏览器发送cookie信息进行身份合法性判断，而web浏览器会自动发送session id（cookie）的特性，攻击者诱使受害者访问恶意请求，以受害者的身份执行“构造”的请求

2.注入点
记住：任何输入都是有害的
（1）GET QueryString
（2）POST

3.绕过方式
csrf实际上是一种逻辑上的错误，常规csrf的防御其实就不好办（不能根据referer，有些csrf结合xss本来请求就是本域发起；而且还存在协议转换丢referer，mobile平台丢referer的情况）

九 、自动化工具攻击
自动化工具的攻击据统计占了总攻击的90%，能否准确的拦截这些自动化工具是非常考验WAF能力的评测项。
自动化工具分为离线工具与在线工具

在线工具（基本是综合性工具）：
各种CDN服务都附带网络安全在线检测功能（云端扫描器），这里就不提及了

离线工具分为以下几类：
a. 漏洞扫描器

（1）综合漏洞扫描器
这里列出比较常用的
scanner信息参考 http://www.sectoolmarket.com/price-and-feature-comparison-of-web-application-scanners-unified-list.html
AWVS
AppScan
WebInspect
NetSparker
Websecurify
WebCruiser
Nikto
wikto
w3af
vega
OWASP-ZAP
arachni-scanner
golismero
brakeman ruby on rails漏洞扫描器
grendel-scan

（2）专项扫描器
a. SQLI/NoSQLI
Havij, SQLMap, Pangolin

b.XSS
X5S，XSScrapy

c.文件包含
fimap

d.开源框架漏洞扫描器
wpscan
joomscan

（3）密码破解工具
hydra
medusa
patator

（4）目录字典攻击工具
Pker
dirbuster

（5）其他
burpsuit
MSF

还有各种自己写的脚本工具 sh/perl/python(e.g. pycurl,python-urllib)/php/ruby/java

十、恶意爬虫/采集器/机器人
恶意爬虫／采集器／机器人会给以内容为王，带宽又不宽裕的小网站带来要命的伤害，所以也是评测WAF防御能力的重要方面，能否有效精准地将其与搜索引擎（搜索引擎是不会提供其完整的IP段的）与正常的API调用区分开来一直是WAF面临的难题。

1.攻击场景
（1）伪装成搜索引擎绕过检测，该场景适用于缺少准确的的搜索引擎判断方法
（2）自动发送垃圾评论机器人等

十一、信息泄露
1.攻击场景
(1) 系统文件
直接访问备份、隐藏等信息文件
(2) 错误回显
暴目录，暴路径
(3)列目录
(4)管理后台暴露：搜索引擎收录管理后台后如何处理

十二 、重定向
主要用于钓鱼
一般情况下WAF是不会防御这种漏洞

十三、基于会话的攻击
1.攻击场景
（1）采用固定的会话
（2）会话ID生成算法可猜测
（3）利用xss等其他漏洞劫持会话ID

十四、权限验证漏洞
1.攻击场景
垂直/水平提升权限
（1）请求参数来控制权限
（2）referer来控制访问权限
一般情况下WAF不会防御这种漏洞（基本逻辑漏洞云WAF都不会防御）

十五 、拒绝服务
1.攻击场景
（1）xml billion laughs （消耗内存）
（2）CC （消耗带宽）
（3）slow HTTP DoS（消耗带宽）

十六、其他漏洞
（1）HTTP响应拆分
（2）XML实体攻击 可以参考pnig0s大牛的《XML实体攻击－从内网探测到命令执行步步惊心》
（3）隐藏参数篡改
1）hidden表单值篡改
2）viewstate值篡改
（4）其他注入
1）XXE注入
2) XPath注入
3）LDAP注入
4）CRLF注入
（5）逻辑错误 云WAF不会防！！

上诉评测的方法都是从WAF防御覆盖度来考虑的，覆盖的越全当然越好。

其实，WAF防御能力的评测概括的说就是检出率 （漏报率 false negative）与准确率 （误报率false positive）。检出率是一般是WAF厂商对外宣传的核心，而实际准确率（不少云WAF对管理后台，API接口调用，富文本的误报就满严重的）更为重要（安全一定不能影响正常使用），后续另开一篇来讲误报的问题。

十七、WAF测试工具

WAF能力评测的方法，一句话来说，就是构造攻击场景，发送攻击包，看WAF的响应。

测试工具地址： https://github.com/tanjiti/WAFTest

源码介绍：
（1）攻击场景
https://github.com/tanjiti/WAFTest/tree/master/vulCode （持续添加中）
（2）发包工具
https://github.com/tanjiti/WAFTest/blob/master/HTTP.pl （参考：HTTP.pl——通过HTTP发包工具了解HTTP协议 ）
https://github.com/tanjiti/WAFTest/blob/master/HTTPFromFile.pl （从文件读取HTTP包内容，并发送）
（3）攻击包
https://github.com/tanjiti/WAFTest/tree/master/t （持续添加中）

测试示例：
git clone https://github.com/tanjiti/WAFTest.git
perl HTTPFromFile.pl -host www.tanjiti.com -dir t -code 403

WAF防御能力评测 – 碳基体 – 碳基体

选项说明如下
perl HTTPFromFile.pl –help

Usage: perl HTTPFromFile.pl [-code 403] [-uri 127.0.0.1] [-host example.com] [-port 80] -file request_file_path

-code: 指定拦截响应码，默认为403，不同的WAF会为拦截响应定制不同的响应码

-uri: 指定使用WAF的IP或域名,默认127.0.0.1

-host: 指定发送请求的Host头，如果uri未指定，则uri设置为host的值，默认localhost

-port: 指定使用WAF的端口号，默认80

-file: T文件的文件路径

-dir: 存放T文件的目录路径

再分享一下我老师大神的人工智能教程吧。零基础！通俗易懂！风趣幽默！还带黄段子！希望你也加入到我们人工智能的队伍中来！https://blog.csdn.net/jiangjunshow