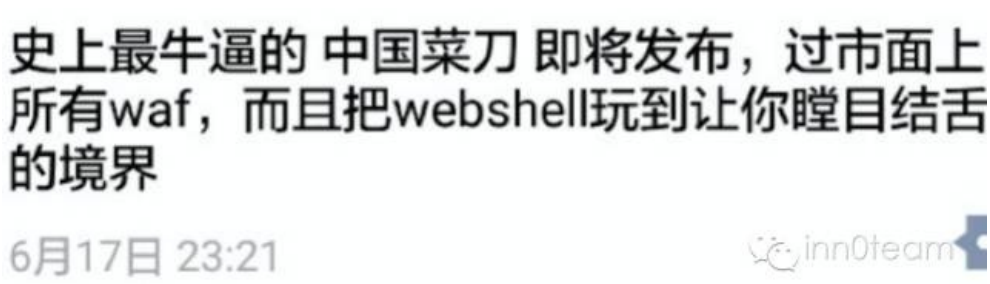
中国菜刀20160620初体验

原创 2016-06-20 药丸君 inn0team

6月17号，某牛在朋友圈发了消息:



> 史上最牛逼的中国菜刀即将发布，过市面上所有的 waf，而且把 webshell 玩到让你瞠目结舌的境界

当时有消息称 6 月底将会发布新版菜刀。

果不其然，在 6 月 20 日，原本已经关闭的 maicaidao.com 又开放了，而且下载量瞬间就到了 660 +。

话不多说，赶紧去下载一个体验一波到底有多牛逼。

> 文章发布时站点又关闭了，具体原因不得而知。

注意：菜刀的唯一官网是： `www.maicaidao.com`

下载后解压，密码是 `www.maicaidao.com`

具体的信息来源参考解压后的 readme.txt。

文件说明

caidao.exe 菜刀程序

db.mdb 菜刀的主数据库

caidao.conf 配置文件（重要，千万别删除）

cache.tmp 菜刀的缓存数据库(可删除)

readme.txt 说明文档(可删除)

ip.dat 一个IP库，用于IP地址识别(可删除)

CCC 菜刀的自写脚本目录(可删除)

Customize 模式的服务端(可删除)

我们可以发现，新版本中主要是多了 caidao.conf，这个配置文件，这个文件作者自己的评价是“重要，千万别删除”。而药丸君觉得这个看似小小的一个改动，甚至可以让中国菜刀起死回生，秒过 waf。

我们知道，菜刀的源代码是不公开的，自从停止更新以后，各种 waf 都对其数据包进行了里三层外三层的分析。在那个年代里，想做到过 waf，通常有下面这几种做法：

1. 过 waf 一句话。通过变形，混淆一句话 webshell 逃过检查。(然后就出了对请求进行分析的 waf)

2. 反编译修改发包特征过 waf。这种方法成本相对比较高，你要会逆向呀，然后还有个尴尬的问题就是你能改的地方实在是有限。

3. 中转脚本修改数据包过 waf。这种方法应该是用菜刀过 waf 用的最多而且方便的了。但是你得要开一个中转的 web 服务，我的尴尬症又犯了。

正是因为有诸多不便，加上菜刀本身不能跨平台，近些年开源的 webshell 管理工具层出不穷，典型的有药丸朋友蚁逅开发的 AntSword(中国蚁剑), 使用 nodejs 与 ES6 开发，跨平台，并且开源。以及 Altman，使用 C++ 开发，支持跨平台，源代码也已经在 github 开源。还有Java 开发的开源 webshell 管理软件 Cknife。诸如此类软件相当之多了。

好吧，来跟药丸君一起看看如何玩转这个 caidao.conf 吧。

---

caidao.conf

对，我得事先解释一下，菜刀能过 waf 的前提，是你需要会改这些配置，不然，你手里的只能是生锈了的菜刀。

1. <FLAG>

返回的内容分隔符，只限三个字符。我们知道原来的分割符是 ->| 与 |<-, 有些 waf 会拦截这个字符，所以，我们可以用生辟点的字符，比如说 ~>$。

2. <UA>

定制 HTTP 请求中的 User-Agent 部分，我们看到默认的 caidao.conf 中将 UA 改成了百度蜘蛛的 UA, 当然你可以改成 Google 的嘛。有些时候需要改改，大部分情况下这个参数可以不用改。

3. <K1> 与 <K2>

POST的参数名称，默认情况下, K1 是 z1，K2 是 z2。一般情况下，waf 是不会拦这个，所以可以不用改。

4. <PHP_BASE>, <ASP_BASE>, <ASPX_BASE>, <PHP_BASE.加密示例>

这4个是 webshell 的基础代码部分，所有的功能代码都会发到这里进行组装，然后发送到服务端去。

如果你服务端的 webshell 做了一层加解密操作，比如服务端代码是这样的
<?php @eval(base64_decode($_POST['caidao']));?>
那么，你就可以在这里修改配置为
<PHP_BASE.加密示例>eval(base64_decode($_POST[id]));&id=%s</PHP_BASE.加密示例>。

可以负责任的讲，免杀，过 waf 大部分都是在这里作文章的，后面基本都是一些功能函数，waf 一般是不会杀的，他要是敢杀，就证明他的业务中不需要那类操作。

这里就有意思了，你想，以前的菜刀只有一个 base64_decode 那现在，你就可以随便改了，比如说，你用 hex, 再比如你

可以用凯撒密码，又或者是栅栏密码，再过分点，你可以用 AES, DES，最过分的就是自己写一个加密解密的算法。那样 waf 想拦就很尴尬了。（作者说的过 waf 大概就是这层意思吧）

5. <GETBASEINFO>

获取基础信息的功能代码。比如你添加完 shell, 直接去虚拟终端，一进去就会显示当前路径，当前用户，系统类型什么的这些。

6. <SHOWFOLDER>

文件目录查看功能代码。这里主要就是用户传进来一个路径，然后把这个路径下的目录，文件全都显示出来，包括一些属性，比如文件时间，权限，类型等。

7. <SHOWTXTFILE>

查看文件内容，编辑文件也首先调用的就是这里。

8. <SAVETXTFILE>

保存文件。

9. <DELETEFILE>

删除文件。

10. <DOWNFILE>

下载服务器文件到本地。

11. <UPLOADFILE>

上传文件到服务端。

12. <PASTEFILE>

粘贴文件。

13. <NEWFOLDER>

新建目录。

14. <WGET>

从远程下载一个文件到服务器。

15. <SHELL>

执行系统命令，这里比较尴尬的地方，就是相关函数被干掉的话，基本是没戏。

其它的就是一些数据库的功能代码了，我实在是不想写了，就那几个单词，百度翻译都知道是什么意思，我就不翻译了

哈。

如何过 WAF

虽然我上面已经提到了一些过 WAF 的点，不过为了凑字数，我还是在这系统总结一下。

这里就不得不提到一些 WAF 的特性了。

1. 基于正则类 waf

这一类的 waf，太好绕了，可以说，你基本都不用改菜刀的配置就可以做到，比如你把一句话 base64一下，再 base64 一下, 这就绕过去了。如果他查你的 Request, 那你就把 <PHP_BASE> 这里随便改改就能像过大街一样过 waf 了。这类 WAF 比如 xxx 狗。

2.基于hook机制的 waf

比如 PHP 中，诸如 eval, create_function, preg_replace, assert 等能够将 string 以脚本代码执行的函数都会经过脚本层的 API **compile_string**。

这类的 waf 从底层 hook 了这个 API, 如果有脚本调用了该 API，**直接干，往死里干，不解释。**一般情况下正常的网站都不会用这些个功能，如果真的有，那就走白名单就行了。**所以一句话要怎么过这种的 WAF 呢？想啥呢，还想用一句话？**除非... 你能挖一个 PHP 的本地代码执行漏洞，那你直接上远控就行了还用一句话?

所以这类 waf 要怎么用菜刀连？ CUSTOM 类型呀。CUSTOM 类型的webshell,你完全可以理解成一个大马。只是输入输出用菜刀去解析了而已。这时一些功能代码都是直接在服务端的，所以不用经过 **compile_string** 这个 API，所以可以正常使用。但是要想执行 system 命令的话，那就是另一回事了。ahahaha，再提一句，这个功能是旧版菜刀支持的。这类的 WAF 典型的比如 D 盾。

总结一下，所以 WAF 分为两类，一类是随便过的，一类是过不了的，：）。

有关菜刀后门

菜刀自从被曝有后门之后，大部分的人都不再相信爱情了。太可怕了，黑吃黑。这也是不少人放弃使用中国菜刀转投自己开发自己的 webshell 管理工具的原因。

经过某牛检测过的，最近几个历史版本中无后门的菜刀 hash 如下：

> 文件: caidao-20100928.exe
> 大小: 200192 字节
> MD5: C05D44DBE353525F492208D891B53875
>
> 文件: caidao-20111116.exe
> 大小: 220672 字节
> MD5: 5001EF50C7E869253A7C152A638EAB8A
>
> 文件: caidao-20141213.exe
> 大小: 220160 字节
> MD5: 4B4A956B9C7DC734F339FA05E4C2A990

这一次菜刀将所有发包功能使用配置文件形式，在很大程度上是解决了后门的问题。

没错，以后你想要求过xxx的菜刀的时候，不需要再去拷整个文件了，直接求一个过xxx的配置文件就行了。

> 那么问题来了，这样真的就没后门了吗？
>
> 显然不是

可以杜绝菜刀主程序没后门，但是，**配置文件里面可以做点手脚呀**。

我举个例子哈，比如我给 **<wget>** 中加入一段后门代码，当你执行 wget 的时候，会把 shell 信息发送到我后门服务器。

你会说，谁有那么傻逼呀，一看就看出来了。

> 真的是这样吗？
>
> 讲道理，安全从业者大部分的人都不会写代码，那些求过xxx刀的，连简单的正则过 waf 什么的都不会，你还指望他能看懂多少行代码。再加上编码混淆一下什么的，画面太美我不敢想。
>
> 以前的菜刀，为了加后门，得逆向加，而且受长度限制，现在使用配置文件了，加后门什么的，更方便了，也更难查了。

最后悄悄打个广告，蚁剑 2.0 会在近期发布，为极客定制，如果你到现在都不懂 Webshell 的原理的话，还是建议你花点时间学一学基础，不会有害的～
