# AWDphpspear

## 操作手册

## Version 0.1.7

目录

前言 1

file模块 2

request模块 3

shell模块 4

upload模块 5

attack模块 6

protect模块 6

后记 7

# 前言

AWD的脚本编写一直是众多CTFer的痛点。可能对于一些AWD熟手来说，编写合适的EXP易如反掌，但对于很多新手AWD初探，对于Python和PHP等语言的理解并不到位，对漏洞挖掘的经验也不足，导致其很难在AWD中有所作为。

awdphpspear这个Python库的开发宗旨，就是解决很多AWD选手在脚本编写上的困难。库整体难度不高，暂时由笔者一人开发完成，希望将来能有更多的同好一起加入，丰富该库的内容。

该库基于Python预置库以及第三方库requests开发，封装了基本的函数以及方法，覆盖了AWD中各种常用操作，希望能给各位CTFer带来一些良好的体验。

另外，在0.1.7版本的更新中，我制作了更加方便使用的集成版本，上传到pypi，可以通过简单的安装来开始使用。

最后，感谢BlackWatch中各位同好在开发中帮我解疑答惑，感谢米斯特安全团队对本库的支持与肯定，感谢正在阅读本手册的你，希望能给你带来一些帮助，即使它微不足道。

Time:2019年11月22日

Author:ZacharyZcR

# file模块

File模块是对本地文件进行操作和审计的一个模块，该模块提供了一些基础的文件操作，可以帮助你在AWD中对源码进行审计。

1. read\_array(target\_file):

该函数需要用户提供一个target\_file的字符串参数，用来确定你要读取的文件，按行读取后，返回一个target\_array，并输出该文件的行数。

样例:ip = read\_array('ip.txt')

1. read\_var(target\_file):

该函数需要用户提供一个target\_file的字符串参数，用来确定你要读取的文件，读取一行后，返回一个target\_variable，并输出该变量的内容。

样例:address = read\_var('address.txt')

1. write\_txt(target\_file,content):

该函数需要用户提供两个target\_file,content的字符串参数，用来确定你要写入的文件，以及写入的内容，写入后自带一个换行符。若读入内容为空，则抛出一个错误。

样例:write\_txt('flag.txt','flag')

1. dir\_tree(startpath):

该函数需要用户提供一个startpath的字符串参数，用来确定要建立目录树的位置，例如'./'，就是在当前目录建立目录树。

样例:dir\_tree('./')

1. get\_php\_list(startpath):

该函数需要用户提供一个startpath的字符串参数，用来确定要检测目录下php文件的位置，例如'./'，就是在当前目录开始检测PHP文件。检测完成后，会输出检测结果，并把结果返回到一个列表中。

样例:get\_php\_list('./')

# request模块

request模块是针对AWD中常用的操作对requests库再封装的一个模块，该模块全部内容都可以用requests库实现，该模块仅仅是提供一个完整性的选择。

1. get(address,data):

该函数需要用户提供两个address,data的字符串参数，其中address为URL，需要带上http://或https://，data则需要写成键值对的形式，例如{'a':'OK'}，该函数将会对address带data进行一次get访问。

样例:get('http://www.baidu.com',{'a':'OK'})

1. post(address,data):

该函数需要用户提供两个address,data的字符串参数，其中address为URL，需要带上http://或https://，data则需要写成键值对的形式，例如{'a':'OK'}，该函数将会对address带data进行一次post访问。

样例:post('http://www.baidu.com',{'a':'OK'})

1. post\_xwf(address,data):

该函数需要用户提供两个address,data的字符串参数，其中address为URL，需要带上http://或https://，data则需要写成键值对的形式，例如{'a':'OK'}，该函数将会对address带data，Content-Type设为application/x-www-form-urlencoded进行一次post访问。

样例:post\_xwf('http://www.baidu.com',{'a':'OK'})

1. post\_json(address,data):

该函数需要用户提供两个address,data的字符串参数，其中address为URL，需要带上http://或https://，data则需要写成键值对的形式，例如{'a':'OK'}，该函数将会对address带data，Content-Type设为application/json进行一次post访问。

样例:post\_json('http://www.baidu.com',{'a':'OK'})

1. test(address):

该函数需要用户提供一个address的字符串参数，进行一次无参数get访问，返回该次访问的状态码

样例:test('http://www.baidu.com')

# shell模块

shell模块是针对已经上传shell的一些操作，封装了一些常用的shell命令，方便CTFer在getshell以后迅速做出获取flag，破坏服务器等操作。

1. rce(address,password,method):

该函数需要用户提供三个address,password,method的字符串参数，其中，address为一句话木马的地址，password为一句话木马的传参名称，method为get或post。该函数读取参数后建立与木马的交互界面，输入shell命令来执行。

样例:rce('http://127.0.0.1/shell.php','a','post')

1. shell\_gen()：

该函数提供给用户一套已经制作好的木马选项，包括普通一句话木马，不死马，内存反弹shell马，通过简单的数字操作即可在本地目录生成。

1. batch\_rce(address,password,method,command)：

该函数需要用户提供四个address,password,method,command的字符串参数，其中，address为一句话木马的地址，password为一句话木马的传参名称，method为get或post，command为需要执行的命令。

样例:batch\_rce('http://127.0.0.1/shell.php','a','post','ls')

# upload模块

upload模块是基于一句话木马的存在而操作的模块，该模块主要提供植入不死马，植入文件等有限的权限维持操作。

1. trojan\_implant(address,webshell,trojan,password):

该函数需要用户提供四个address,webshell,trojan,password,的字符串参数，其中，address为一句话木马的地址上一层，webshell为连接的一句话木马名称，trojan为植入不死马的名称，password为一句话木马的传参名称。该函数读取参数后迅速发送请求植入不死马，并自动触发不死马。

样例:trojan\_implant('http://127.0.0.1/','shell.php','user.php','a')

1. file\_implant(address,webshell,name,data,password):

该函数需要用户提供五个address,webshell,name,data,password,的字符串参数，其中，address为一句话木马的地址上一层，webshell为连接的一句话木马名称，name为植入文件的名称，data为植入文件的内容，password为一句话木马的传参名称。该函数读取参数后迅速发送请求植入文件，并回显文件访问内容。

样例:file\_implant('http://127.0.0.1/','shell.php','user.php','a','a')

1. check(address):

该函数需要用户提供一个address的字符串参数，address为访问的文件地址，该函数发送一次无参数的get访问，主要用来检测木马是否存在。

样例:check('http://127.0.0.1/','shell.php')

1. trojan\_implant\_memory(address,webshell,trojan,ip,port,password):

该函数需要用户提供六个address,webshell,trojan,ip,port,password,的字符串参数，其中， address为一句话木马的地址上一层，webshell为连接的一句话木马名称，trojan为植入内存马的名称，ip为反弹shell的ip，port为反弹shell的port，password为一句话木马的传参名称，该函数读取参数后发送请求植入内存反弹shell马，落地一次后迅速删除自身，并回显反弹shell链接地址。

attack模块

attack模块是一个概念性的模块，也是本库的精髓所在，鄙人搜集和原创的一些AWD花式操作会在该模块中集合，可能它不会对每一场比赛尽皆适用，但其中的思路和想法也是值得各位选手借鉴的。

1. confuse(ip = [],php\_list = []):

该函数需要用户提供两个ip,php\_list的字符串列表参数，其中，ip为目标的ip地址，php\_list推荐用file模块的get\_php\_list获取，该函数的作用为对多个目标的全体php文件发起敏感请求，将真实攻击流量混入混淆流量中，防止被抓取流量。

样例:confuse(ip,php)

# protect模块

该模块是之前的Defence\_Framework改造而来，继承了之前防御框架的所有功能， 这里不做赘述，需要查看的朋友可以到项目github地址找之前的readme。

# 后记

整个库来源于我上一个AWD自动化工具的编写，我一直致力于编写一个可以通杀各比赛AWD的框架，后来我发现我错了，因为网络安全的变化是本质性的，没有永远的漏洞，也不存在可以通杀一切的工具，即使是0day也只有一次爆发可言。该库的开发可谓是仓促无比，代码也没有经过完整的优化，我会将源码完全开源到github上，对该项目有任何意见和建议欢迎反馈。

0.0.4版本是一个简陋的demo版本，它仅仅提供了一些简陋的进攻手段，其他的一些包括断数据库，fork炸弹，反代，感染马等精彩的骚操作已经有了初步的构想，可能会在未来几个版本尽快加入。另外防御框架方面的一些文件守护，日志包含等基本操作也会在不久后上线。

0.1.7版本的更新支持了一套集成化的工具框架，使用者可以方便的进行使用，也可以采用库中的各种封装函数编写自己的工具。

还望各位支持。

项目Github地址:https://github.com/ZacharyZcR/AWD

项目Pypi地址：<https://pypi.org/project/awdphpspear/>

作者联系方式：

作者：QQ：2903735704

米斯特安全团队负责人：QQ：627437686