phpinfo中值得注意的信息

作者: seaii 时间: October 25, 2017 分类: WEB安全

在我们平时的渗透、ctf的过程中，或多或少会碰到 `phpinfo` 页面。但是这个页面包含的信息太多，常常感觉无从下手，在这里总结一下，可能没有那么全面。

2019.2.28 时代在发展，技术在进步。将近两年的时间已经有了好多新姿势，中间有好几次想完善一下这篇blog，一直没做，这次终于拔掉了这个flag，23333～

2019.3.4 修改了一些错误，写了一个抓取重要信息的小工具:https://github.com/proudwind/phpinfo_scanner

基本信息

php版本

这是最基本的，php更新速度非常快，各版本都有一些小特性。

http://www.cnblogs.com/iamstudy/articles/study_from_php_update_log.html

php7的一些特性：

1. 移除不支持SQL预编译的Mysql扩展：mysql
2. 移除preg_replace中容易导致代码执行漏洞的正则模式：e
3. assert从一个函数变成一个语法结构（类似eval，无法再动态调用。至此，大量PHP一句话木马将失效），7.2中废弃字符串形式的参数
4. hex字符串（如0xf4c3b00c）不再被作为数字，is_numeric也不再认可，可见 https://3v4l.org/ORuc7
5. 7.2中废弃可以动态执行字符串的 create_function
6. 7.2中废弃容易导致变量覆盖的无第二个参数的parse_str
7. 移除<script language="php">和<%，这两种另类的PHP标签
8. 移除dl函数

图片内容来自p师傅小密圈。

system info

System	Linux ebs-8457 2.6.32-642.6.2.el6.x86_64 #1 SMP Wed Oct 26 06:52:09 UTC 2016 x86_64

详细的操作系统信息，为提权做准备

server api

Server API	FPM/FastCGI

php解释器与应用层的桥梁。

1. FPM/FastCGI 多用于和nginx通信，当然也可用于其他web中间件。

   Fastcgi协议分析 && PHP-FPM未授权访问漏洞 && Exp编写

   https://gist.github.com/phith0n/9615e2420f31048f7e30f3937356cf75

2. Apache 2.0 Handler php为apache提供的专用SAPI
3. Command Line Interface php命令行
4. CGI/FastCGI 碰见的几次都是用于iis

配置文件位置

Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d

某些情况下可以加载自己的扩展。

Registered PHP Streams and filters

Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip

常见的就不说了。

1. phar

   利用phar/zip协议绕过有后缀的文件包含：include zip:///var/www/html/upload/1.gif#1.php

   phar反序列化 https://paper.seebug.org/680/

2. gopher

   利用 Gopher 协议拓展攻击面

   https://github.com/tarunkant/Gopherus

3. dict

探测为主

Registered Stream Filters	zlib.*, bzip2.*, mcrypt.*, mdecrypt.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

谈一谈php://filter的妙用

核心配置

extension_dir

extension_dir	/usr/php5.2/modules	/usr/php5.2/modules

php扩展的路径

allow_url_include

远程文件包含，但是一般不会开启

asp_tags

asp_tags	Off	Off

php标签有4种形式，如果这个选项不开启的话，使用asp的标签是不会解析的。

~~这里有一篇 .user.ini + asp_tags 绕过的文章 针对内容(php tags)检测的一种绕过思路~~

原来的文章已经删了，说说大体意思。当 `<?php ?>` 标签被过滤时，可以通过.user.ini来覆盖php.ini中的配置。.user.ini在nginx等其他web中间件中也是有效的，应用范围比.htaccess广。

注意： 在PHP 7已经完全移除了这种标签

short_open_tag

short_open_tag	On	On

还是标签的问题，允许 `<??>` 这种形式，并且 `<?=` 等价于 `<? echo`

disable_functions

disable_functions	passthru,exec,system,shell_exec,proc_open,popen	passthru,exec,system,shell_exec,proc_open,popen

有时候我们上传了一个webshell却不能用，有很大可能是管理员做了配置，禁用了php执行系统命令的函数。

绕过的方式有这么几个：

1. 黑名单绕过

百密一疏，寻找黑名单中漏掉的函数，上图中禁用的函数算是比较全的了。

比如在编译php时如果加了 `--enable-pcntl` 选项，就可以使用 `pcntl_exec()` 来执行命令。

渗透技巧：利用pcntl_exec突破disable_functions

2. 利用扩展（如ImageMagick）绕过

   利用ImageMagick漏洞绕过disable_function

3. 利用环境变量LD_PRELOAD来绕过

   利用环境变量LD_PRELOAD来绕过php disable_function

   https://github.com/yangyangwithgnu/bypass_disablefunc_via_LD_PRELOAD

4. 利用扩展库绕过

   http://www.91ri.org/8700.html

一个综合：https://github.com/l3m0n/Bypass_Disable_functions_Shell

enable_dl

enable_dl	On	On

上面说的利用扩展库绕过disable_functions，需要使用 `dl()` 并且开启这个选项

magic_quotes_gpc

magic_quotes_gpc	Off	Off

这个就不用多说了吧

open_basedir

open_basedir	*no value*	*no value*

这个参数将用户可操作的文件限制在某目录下，但是这个限制是可以绕过的。

PHP绕过open_basedir列目录的研究

php5全版本绕过open_basedir读文件脚本

绕过open_basedir读文件脚本

PHP Variables

真实ip

iis用

cdn什么的都不存在的，找到真实ip，扫一扫旁站，没准就拿下几个站。

当网站使用了nginx反向代理时，如果反代服务器和web服务器在同一内网，这个值可能会是内网ip。

当网站在docker中运行时，这个值会是宿主机docker网卡上的ip。

web根目录

`$_SERVER['DOCUMENT_ROOT']` 可能会有偏差。

临时文件路径

向 `phpinfo()` 页面post一个shell（自己写一个上传页面），可以在 `_FILES["file1"]` 中看到上传的临时文件，如果有个lfi，便可以直接getshell了。

https://github.com/hxer/vulnapp/tree/master/lfi_phpinfo

扩展

imagick

前段时间影响比较大的漏洞，注意看版本。

漏洞影响ImageMagick 6.9.3-10之前的版本，包括ubuntu源中安装的ImageMagick。

ImageMagick 漏洞利用方式及分析

ImageMagick远程执行漏洞分析及利用

libxml

libxml 2.9以前的版本默认支持并开启了外部实体的引用，服务端解析用户提交的 xml 文件时未对 xml 文件引用的外部实体（含外部普通实体和外部参数实体）做合适的处理，会导致XXE。

memcache

Memcache未授权访问漏洞利用及修复

redis

redis也不用多说了吧

session

主要是序列化的一些问题

session.serialize_handler	php	php_serialize
session.upload_progress.cleanup	Off	Off
session.upload_progress.enabled	On	On

序列化处理器不一致导致对象注入

当一个上传在处理中，同时POST一个与INI中设置的session.upload_progress.name同名变量时，当PHP检测到这种POST请求时，它会在$_SESSION中添加一组数据。所以可以通过Session Upload Progress来设置session。

具体可以看我的另一篇文章php对象注入总结。

xdebug

xdebug命令执行

Xdebug: A Tiny Attack Surface

opcache

当开启了opcache并可以上传文件时，可以在本地生成一个与服务器文件名相同的文件，并生成缓存文件xx.php.bin。上传后恶意缓存文件会将服务器上的原文件覆盖，从而getshell。

需要将缓存文件的system_id和timestamp两个字段为服务器上文件的值。

system_id可以使用工具https://github.com/GoSecure/php7-opcache-override修改。

imap

https://github.com/vulhub/vulhub/blob/master/php/CVE-2018-19518/README.md

标签: ctf , phpinfo , pentest