CTF线下赛AWD攻防准备

转载

Qwzf 于 2019-07-16 17:41:56 发布 4126 收藏 70

分类专栏： CTF AWD 文章标签： AWD CTF


CTF 同时被 3 个专栏收录

30 篇文章 6 订阅

订阅专栏


线下赛

1 篇文章 0 订阅

订阅专栏


AWD

1 篇文章 0 订阅

订阅专栏

CTF线下赛AWD攻防准备

最近发现一篇博客。感觉对CTF线下赛-AWD模式，总结比较好。于是学习了解了一下，为了方便寻找，把这篇博客复制了过来，并补充了点。。。

1、赛制流程：攻防模式（AWD）常见于线下攻防

一般比赛的具体环境会在开赛前半个小时由比赛主办方给出，前半个小时应熟悉配置环境。准备网线、网线转接口。

最好的防御就是攻击，不做好安全加固就会被吊打。

2、 赛前准备：

常用工具：(整理适合自己的)

- Burpsuite
- sqlmap
- nmap、masscan
- nc
- D盾
- Xshell、Xftp
- 菜刀或蚁剑
- Chrome、Firefox各类插件

一句话木马：

- php
- asp
- aspx
- jsp
- 内存马

py库、脚本：

- pwntools
- requests
- 软waf
- 日志分析
- Exp

SSH客户端：

- PuTTY
- XShell

编辑器：

- Sublime
- VS Code
- Notepad++
- Vim

个人知识库

常见应用源码库

Writeup集合

基础知识：

- 语言运用：编写自动化脚本等……
- WEB安全：熟悉常见漏洞类型、常见框架……
- pwn型：需要较好的底层基础、懂汇编等，需要理解各种堆栈溢出的原理、基础密码学……
- 中间件：apache、nginx、tomcat、jboss、weblogic
- 语言基础：php、java、python
- 常见web应用：phpmyadmin、dedecms、phpcms、帝国cms、Discuz
- linux命令：`netstat -tulpn` 、`ps -ef`

Gamebox:

系统：ubuntu、centos

中间件、版本：

- apache \ nginx
- php \ php-fpm
- tomcat \ jboss \ weblogic

web程序

数据库：

- MySQL \ MariaDB \ Oracle
- Redis \ MongoDB

3、常见加固方式：

加固流程：

修改网站管理员密码

备份网站源码

1. `tar -zcf /tmp/name.tar.gz /path/web`
2. `tar -zcf /tmp/name.tar.gz /var/www/html`

备份数据库

1. `mysqldump -u 用户名 -p 数据库名 > 导出的文件名`
2. `mysqldump -u user -p database > /tmp/database.sql`

修改ssh密码（即修改当前用户密码）

修改MySQL密码

1. `set password for 用户名@localhost = password('新密码');`
2. `set password for user@localhost = password('123');`

修改MongoDB密码（27017端口）

修改Redis密码（6379端口）

修改网站源码中的数据库连接配置

部署waf（视情况而定）

准备一个软waf

如何使用phpwaf.php

找到CMS/框架通用配置文件进行包含：

1. PHPCMS V9：`\phpcms\base.php`
2. PHPWIND8.7：`\data\sql_config.php`
3. DEDECMS5.7：`\data\common.inc.php`

3. DEDECMS5.7：`\data\common.inc.php`
4. DiscuzX2：`\config\config_global.php`
5. WordPress：`\wp-config.php`
6. Metinfo：`\include\head.php`

修改php.ini文件后重启（高权限）：

禁用敏感函数：

disable_functions = system,exec,shell_exec,passthru,proc_open,proc_close,proc_get_status,checkdnsrr,getmxrr,getservbyname,getservbyport, syslog,popen,show_source,highlight_file,dl,socket_listen,socket_create,socket_bind,socket_accept, socket_connect, stream_socket_server, stream_socket_accept,stream_socket_client,ftp_connect, ftp_login,ftp_pasv,ftp_get,sys_getloadavg,disk_total_space, disk_free_space,posix_ctermid,posix_get_last_error,posix_getcwd, posix_getegid,posix_geteuid,posix_getgid, posix_getgrgid,posix_getgrnam,posix_getgroups,posix_getlogin,posix_getpgid,posix_getpgrp,posix_getpid,posix_getppid,posix_Tupungato,posix_getpwuid, posix_getrlimit, posix_getsid,posix_getuid,posix_isatty, posix_kill,posix_mkfifo,posix_setegid,posix_seteuid,posix_setgid, posix_setpgid,posix_setsid,posix_setuid,posix_strerror,posix_times,posix_ttyname,posix_uname

自动包含waf：
`auto_prepend_file = safe.php路径;`

分析日志文件

文件监控

1. 准备一个脚本，监控并删除所有新增文件。
2. 发现内存马，直接重启php。
3. 若监控脚本无法使用，使用命令定期查看新增与修改文件。
4. find web路径 -ctime -1 （查看最近一日新增的文件，是否可疑）

修改目录权限：(可能会违规)

1. `chmod -R 644 www`

waf脚本

```
<?php
 //部署waf可能会导致服务不可用，需要谨慎部署。
    error_reporting(0);
    define('LOG_FILENAME','log.txt');
    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"];

        //rewirte shell which uploaded by others, you can do more
        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']);//fix a bug
        $input = array("Get"=>$get, "Post"=>$post, "Cookie"=>$cookie, "File"=>$files, "Header"=>$header);
        //deal with
        $pattern = "select|insert|update|delete|and|or|\'|\/\*|\*|\.\.\/|\.\/|union|into|load_file|outfile|dumpf
ile|sub|hex";
        $pattern .= "|file_put_contents|fwrite|curl|system|eval|assert";
        $pattern .="|passthru|exec|system|chroot|scandir|chgrp|chown|shell_exec|proc_open|proc_get_status|popen|
ini_alter|ini_restore";
        $pattern .="|`|dl|openlog|syslog|readlink|symlink|popepassthru|stream_socket_server|assert|pcntl_exec";
        $vpattern = explode("|",$pattern);
        $bool = false;
        foreach ($input as $k => $v) {
            foreach($vpattern as $value){
                foreach ($v as $kk => $vv) {
                    if (preg_match( "/$value/i", $vv )){
                        $bool = true;
                        logging($input);
                        break;
                    }
                }
                if($bool) break;
            }
            if($bool) break;
        }
    }

    function logging($var){
        file_put_contents(LOG_FILENAME, "\r\n".time()."\r\n".print_r($var, true), FILE_APPEND);
        // die() or unset($_GET) or unset($_POST) or unset($_COOKIE);
    }
    waf();
?>
```

文件监控脚本

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

import os
import hashlib
import shutil
import ntpath
import time
```

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


# 特殊文件路径字符串
Special_path_str = 'drops_JWI96TY7ZKNMQPDRUOSG0FLH41A3C5EXVB82'
bakstring = 'bak_EAR1IBM0JT9HZ75WU4Y3Q8KLPCX26NDFOGVS'
logstring = 'log_WMY4RVTLAJFB28960SC3KZX7EUP1IHOQN5GD'
webshellstring = 'webshell_WMY4RVTLAJFB28960SC3KZX7EUP1IHOQN5GD'
difffile = 'diff_UMTGPJO17F82K35Z0LEDA6QB9WH4IYRXVSCN'

Special_string = 'drops_log'  # 免死金牌
UNICODE_ENCODING = "utf-8"
INVALID_UNICODE_CHAR_FORMAT = r"\?%02x"

# 文件路径字典
spec_base_path = os.path.realpath(os.path.join(CWD, Special_path_str))
Special_path = {
    'bak' : os.path.realpath(os.path.join(spec_base_path, bakstring)),
    'log' : os.path.realpath(os.path.join(spec_base_path, logstring)),
    'webshell' : os.path.realpath(os.path.join(spec_base_path, webshellstring)),
    'difffile' : os.path.realpath(os.path.join(spec_base_path, difffile)),
}


def isListLike(value):
    return isinstance(value, (list, tuple, set))


# 获取Unicode编码
def getUnicode(value, encoding=None, noneToNull=False):

    if noneToNull and value is None:
        return NULL

    if isListLike(value):
        value = list(getUnicode(_, encoding, noneToNull) for _ in value)
        return value

    if isinstance(value, unicode):
        return value
    elif isinstance(value, basestring):
        while True:
            try:
                return unicode(value, encoding or UNICODE_ENCODING)
            except UnicodeDecodeError, ex:
                try:
                    return unicode(value, UNICODE_ENCODING)
                except:
                    value = value[:ex.start] + "".join(INVALID_UNICODE_CHAR_FORMAT % ord(_) for _ in value[ex.st
art:ex.end]) + value[ex.end:]
    else:
        try:
            return unicode(value)
        except UnicodeDecodeError:
            return unicode(str(value), errors="ignore")
```

```
# 目录创建
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'])


if __name__ == '__main__':
```

```
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)
                    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())
```

4、攻防演练：

如何获得flag?

在实际比赛中，一般有两种方式获取flag，一种是先获取webshell权限，然后去读flag文件，另一种则是直接通过漏洞读取flag文件。

- Getshell：
  - 官方后门、文件上传
  - 文件写入、文件包含
  - 命令注入、反序列化
  - Redis写shell
  - Mysql写shell
- 直接读文件：
  - SSRF
  - 任意文件读取
  - XXE
  - 文件包含
  - Sqli

1. web后门：

在任意APP的某个文件的源码中加上一句话后门。

- @eval($_POST['XXX']);
- @assert($_POST['XXX']);
- system($_REQUEST['CMD']);

对于这种类型的漏洞，只要用正则遍历匹配就能找到
`grep -r "eval\(\$_"`
或者还有一些复杂变异的后门，这种情况就可以选择使用D盾Webshell查杀或者SafeDog之类的工具对源码进行扫描。只要删掉就可以解决。

2. 系统后门

- NC后门
- SSH后门
- suid后门

3. webshell：

- 内存马：不断生成shell文件
- Webshell密码：给Webshell增加密码，增加一个password参数MD5

4. 文件上传：

put方法：

Tomcat远程代码执行漏洞分析(CVE-2017-12615)

一般上传：

- 各种绕过方式一定要熟悉
- 常见改包、解析漏洞、图片渲染、逻辑文件（双文件上传）、条件竞争
- 防护方式——白名单、禁止上传目录执行权限、上传于Web目录外

5. 文件写入：

缓存：

- 存在缓存机制，后缀名为php，直接代码执行。

-配置文件：

- 单引号内：输入单引号，尝试逃逸。如`'+@phpinfo()+'`
- 双引号内：输入会被解析的符号。如`${@phpinfo()}`

模板文件：

- 模板被包含，getshell。
- 创建新文件时无校验后缀名。

日志：

- 日志以php后缀保存，X-Forwarded-For来伪造ip植入木马。

6. 命令注入/反序列化：

PHP中使用unserialize函数对数据进行反序列化，反序列化过程类的__wakeup方法与__destruct方法会被调用。

```php
<?php

class TestClass
{
    public $variable;
    public function __destruct()
    {
        print_r(shell_exec("ping ".$this->variable));
    }

}
unserialize($_GET['data']);
?>
```

7. 文件读取：

SSRF：存在服务器请求伪造漏洞时，可使用file协议读取本地文件。

```
http://127.0.0.1/read.php?url=file:///flag
```

SQL注入：目标存在SQL注入时，可尝试直接读取flag。

常规注入、盲注、二次注入、insert注入、http头注入

读取

```
select load_file()
```

写入

```
select outfile()
select dumpfile()
```

8. 困难漏洞：

有时候出题者会直接丢一个0day，现场审计。找不到漏洞没关系，上Waf保平安，时刻关注你的日志记录，NPC也会打出攻击流量。

找到别人写在自己服务器上的shell，一般其他服务器也会有，可以去留后门。

Collected by 此名如此彪悍

如有侵权，请敬请告知！！