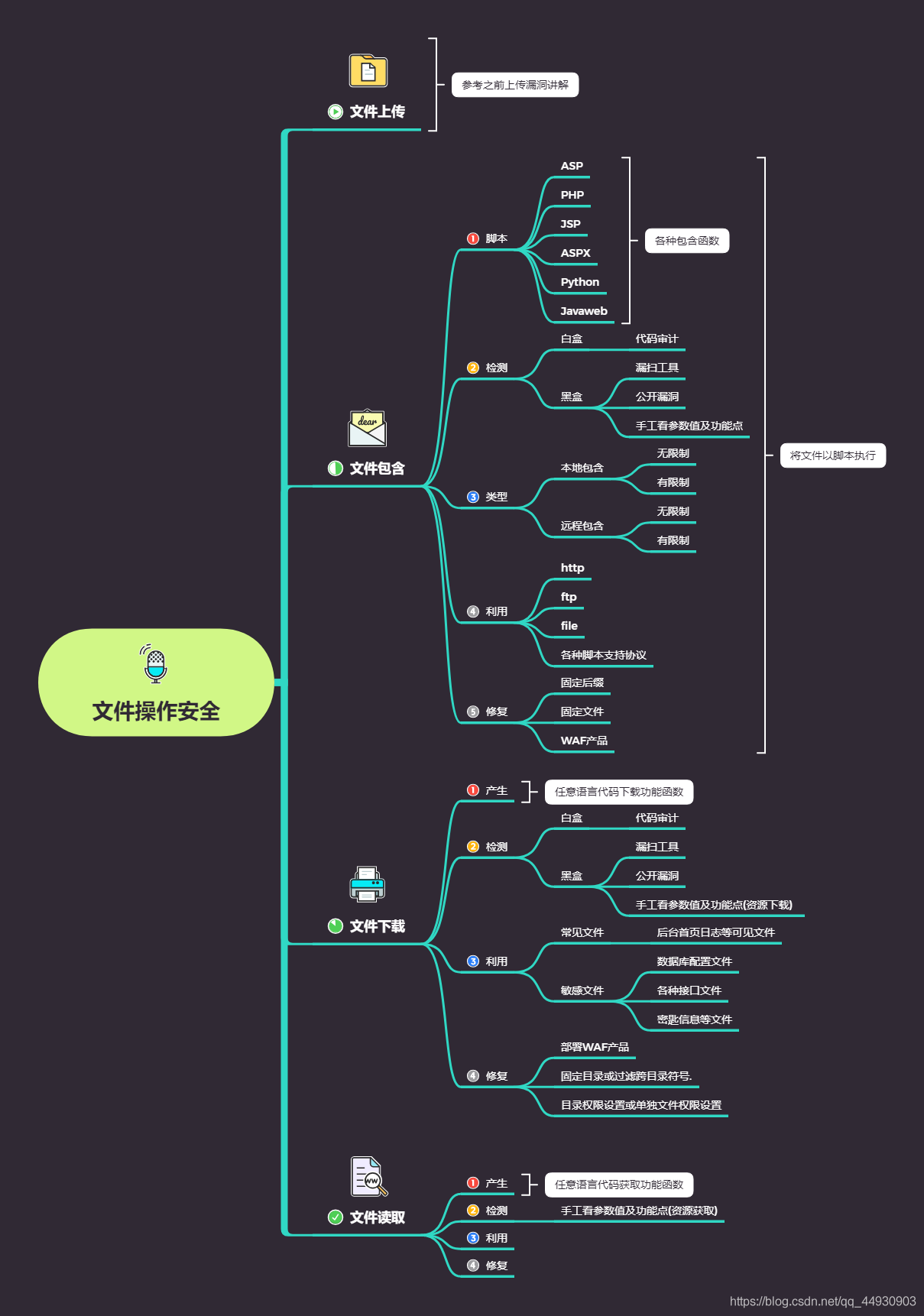
# 32 WEB漏洞-文件操作之文件下载读取全解



#文件下载，读取

原理，检测，利用，修复等

#利用

数据库配置文件下载或读取后续

接口密匙信息文件下载或读取后续

#文件名，参数值，目录符号

read.xxx?filename=

down.xxx?filename=

readfile.xxx?file=

downfile.xxx?file=

…/ …\ .\ ./等

%00 ? %23 %20 .等

&readpath=、&filepath=、&path=、&inputfile=、&url=、&data=、&readfile=、

&menu=、META-INF= 、WEB-INF

1.文件被解析，则是文件包含漏洞

2.显示源代码，则是文件读取漏洞

3.提示文件下载，则是文件下载漏洞

下载或文件读取漏洞：

对应文件：配置文件（数据库，平台，各种等）

#各种协议调用配合

#Javaweb文件下载代码

https://blog.csdn.net/Cheng\_May/article/details/78600833

<%@ page language="java" import="java.util.\*" pageEncoding="UTF-8"%>

<%

String path = request.getContextPath();

String basePath = request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+path+"/";

%>

<a href="/download/DownloadServlet?filename=1.jpg">文件下载</a>

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setCharacterEncoding("UTF-8");

//设置ContentType字段值

response.setContentType("text/html;charset=utf-8");

//获取所要下载的文件名称

String filename = request.getParameter("filename");

//下载文件所在目录

String folder = "/filename/";

//通知浏览器以下载的方式打开

response.addHeader("Content-type", "appllication/octet-stream");

response.addHeader("Content-Disposition", "attachment;filename="+filename);

//通知文件流读取文件

InputStream in = getServletContext().getResourceAsStream(folder+filename);

//获取response对象的输出流

OutputStream out = response.getOutputStream();

byte[] buffer = new byte[1024];

int len;

//循环取出流中的数据

while((len = in.read(buffer)) != -1){

out.write(buffer,0,len);

}

}

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

www.zdns.com

下在漏洞在哪里测？下载漏洞怎么判断存在

拿到下载文件的url，一般是base64加密，解密就好，再观察路径规律；然后再拿到你想下载的文件的ur，

base64加密在通过url下载

➢ 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

爬虫扫描地址-分析参数名参数值-文件操作安全-对应脚本

修改提交方式测试-读取WEB配置文件WEB-INF/web.xml

访问读取对应地址-访问读取flag对应class文件-

（WEB-INF/classes/com/wm/ctf/FlagController.class）

涉及资源：

https://www.seebug.org/vuldb/ssvid-98122

https://www.ichunqiu.com/battalion?t=1&r=57475

https://blog.csdn.net/Cheng\_May/article/details/78600833

https://buuoj.cn/challenges#%5BRoarCTF%202019%5DEasy%20Java