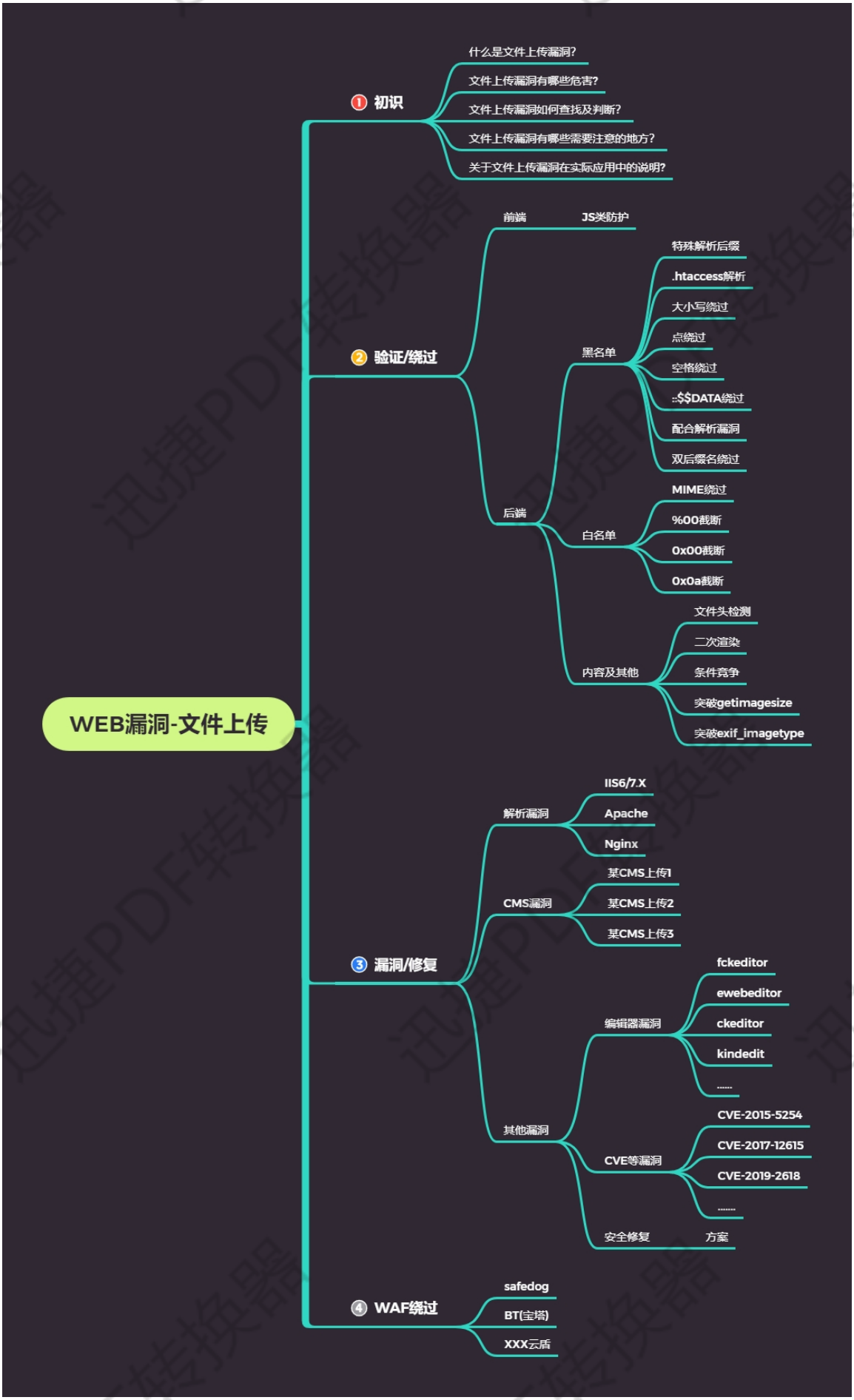
WEB 漏洞-文件上传之 WAF 绕过及安全修复


WEB漏洞-文件上传
① 初识
什么是文件上传漏洞?
文件上传漏洞有哪些危害?
文件上传漏洞如何查找及判断?
文件上传漏洞有哪些需要注意的地方?
关于文件上传漏洞在实际应用中的说明?
② 验证/绕过
前端
JS类防护
后端
黑名单
特殊解析后缀
.htaccess解析
大小写绕过
点绕过
空格绕过
::$$DATA绕过
配合解析漏洞
双后缀名绕过
白名单
MIME绕过
%00截断
0x00截断
0x0a截断
内容及其他
文件头检测
二次渲染
条件竞争
突破getimagesize
突破exif_imagetype
③ 漏洞/修复
解析漏洞
IIS6/7.X
Apache
Nginx
CMS漏洞
某CMS上传1
某CMS上传2
某CMS上传3
其他漏洞
编辑器漏洞
fckeditor
ewebeditor
ckeditor
kindedit
.....
CVE等漏洞
CVE-2015-5254
CVE-2017-12615
CVE-2019-2618
.....
安全修复
方案
④ WAF绕过
safedog
BT(宝塔)
XXX云盾

#上传参数名解析：明确哪些东西能修改？
Content-Disposition：一般可更改
name：表单参数值，不能更改
filename：文件名，可以更改
Content-Type：文件 MIME，视情况更改

#常见绕过方法：
数据溢出-防匹配(xxx...)
符号变异-防匹配（' " ;）
数据截断-防匹配(%00 ; 换行)
重复数据-防匹配(参数多次)

#Payload:
大量垃圾数据缓冲溢出(Content-Disposition,filename 等)
filename=x.php
filename="x.php
filename='x.php
filename="a.jpg;.php";
filename="a.php%00.jpg"
filename="Content-Disposition: form-data; name="upload_file";x.php"
filename="x.jpg";filename="x.jpg";.....filename="x.php";
filename="xxx/x.jpg"
filename=
"
x
.
p
h
p
"

#文件上传安全修复方案
后端验证：采用服务端验证模式
后缀检测：基于黑名单，白名单过滤
MIME 检测：基于上传自带类型检测
内容检测：文件头，完整性检测

自带函数过滤：参考 uploadlabs 函数
自定义函数过滤：function check_file(){}
WAF 防护产品：宝塔，云盾，安全公司产品等

语言	可解析后缀
asp/aspx	asp,aspx,asa,asax,ascx,ashx,asmx,cer,aSp,aSpx,aSa,aSax,aScx,aShx,aSmx,cEr
php	php,php5,php4,php3,php2,pHp,pHp5,pHp4,pHp3,pHp2,html,htm,phtml,pht,Html,Htm,pHtml
jsp	jsp,jspa,jspx,jsw,jsv,jspf,jtml,jSp,jSpx,jSpa,jSw,jSv,jSpf,jHtml

- Windows下文件名不区分大小写，Linux下文件名区分大写欧西
- Windows下ADS流特性，导致上传文件xxx.php::$DATA = xxx.php
- Windows下文件名结尾加入`.`,`空格`,`<`,`>`,`>>>`,`0x81-0xff`等字符，最终生成的文件均被windows忽略。

演示案例：

✧ 上传数据包参数对应修改测试

✧ Safedog+云服务器+uploadlabs 测试

✧ Safedog+云服务器+uploadlabs_fuzz 测试

✧ 文件上传安全修复方案-函数自定义及 WAF

涉及资源：

https://github.com/fuzzdb-project/fuzzdb

https://github.com/TheKingOfDuck/fuzzDicts
