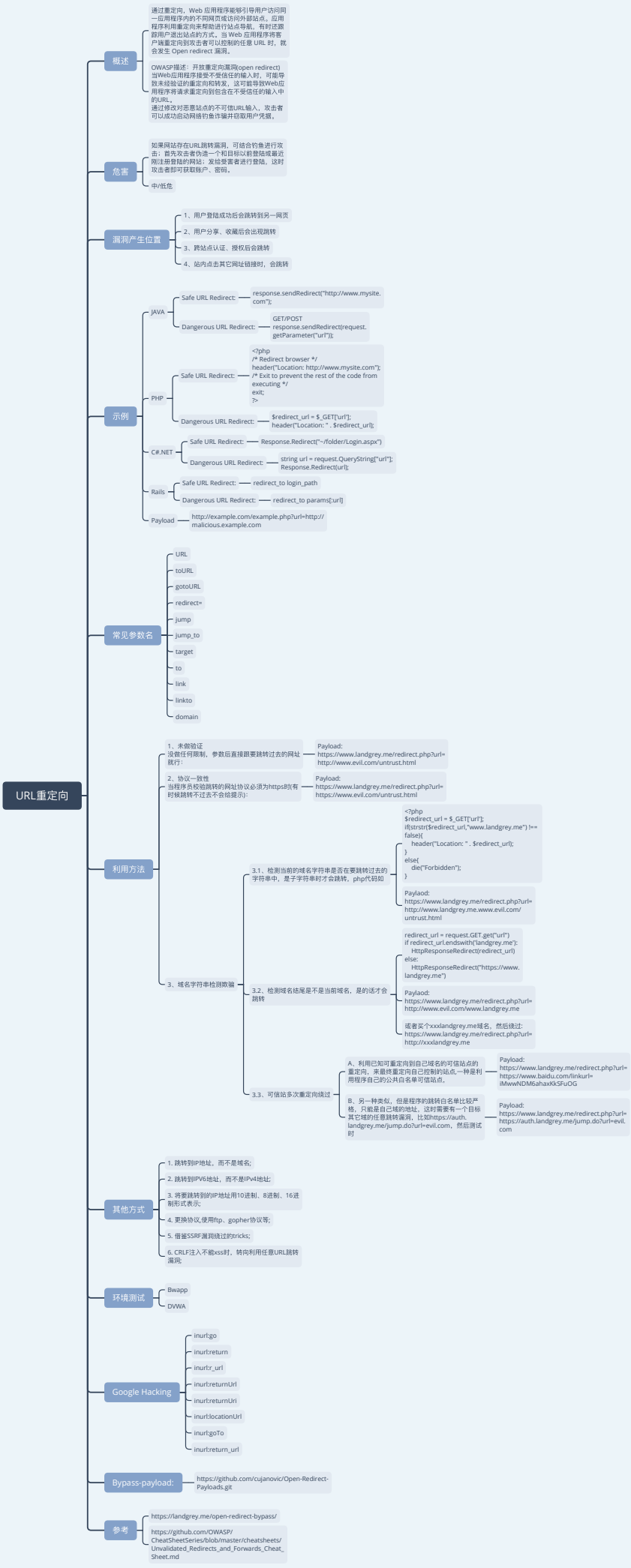
URL重定向
概述
通过重定向，Web 应用程序能够引导用户访问同一应用程序内的不同网页或访问外部站点。应用程序利用重定向来帮助进行站点导航，有时还跟踪用户退出站点的方式。当 Web 应用程序将客户端重定向到攻击者可以控制的任意 URL 时，就会发生 Open redirect 漏洞。
OWASP描述：开放重定向漏洞(open redirect)
当Web应用程序接受不受信任的输入时，可能导致未经验证的重定向和转发，这可能导致Web应用程序将请求重定向到包含在不受信任的输入中的URL。
通过修改对恶意站点的不可信URL输入，攻击者可以成功启动网络钓鱼诈骗并窃取用户凭据。
危害
如果网站存在URL跳转漏洞，可结合钓鱼进行攻击：首先攻击者伪造一个和目标以前登陆或最近刚注册登陆的网站；发给受害者进行登陆，这时攻击者即可获取账户、密码。
中/低危
漏洞产生位置
1、用户登陆成功后会跳转到另一网页
2、用户分享、收藏后会出现跳转
3、跨站点认证、授权后会跳转
4、站内点击其它网址链接时，会跳转
示例
JAVA
Safe URL Redirect:
response.sendRedirect("http://www.mysite.com");
Dangerous URL Redirect:
GET/POST
response.sendRedirect(request.getParameter("url"));
PHP
Safe URL Redirect:
<?php
/* Redirect browser */
header("Location: http://www.mysite.com");
/* Exit to prevent the rest of the code from executing */
exit;
?>
Dangerous URL Redirect:
$redirect_url = $_GET['url'];
header("Location: " . $redirect_url);
C#.NET
Safe URL Redirect:
Response.Redirect("~/folder/Login.aspx")
Dangerous URL Redirect:
string url = request.QueryString["url"];
Response.Redirect(url);
Rails
Safe URL Redirect:
redirect_to login_path
Dangerous URL Redirect:
redirect_to params[:url]
Payload
http://example.com/example.php?url=http://malicious.example.com
常见参数名
URL
toURL
gotoURL
redirect=
jump
jump_to
target
to
link
linkto
domain
利用方法
1、未做验证
没做任何限制，参数后直接跟要跳转过去的网址就行：
Payload:
https://www.landgrey.me/redirect.php?url=http://www.evil.com/untrust.html
2、协议一致性
当程序员校验跳转的网址协议必须为https时(有时候跳转不过去不会给提示)：
Payload:
https://www.landgrey.me/redirect.php?url=https://www.evil.com/untrust.html
3、域名字符串检测欺骗
3.1、检测当前的域名字符串是否在要跳转过去的字符串中，是子字符串时才会跳转，php代码如
<?php
$redirect_url = $_GET['url'];
if(strstr($redirect_url,"www.landgrey.me") !== false){
header("Location: " . $redirect_url);
}
else{
die("Forbidden");
}
Paylaod:
https://www.landgrey.me/redirect.php?url=http://www.landgrey.me.www.evil.com/untrust.html
3.2、检测域名结尾是不是当前域名，是的话才会跳转
redirect_url = request.GET.get("url")
if redirect_url.endswith('landgrey.me'):
HttpResponseRedirect(redirect_url)
else:
HttpResponseRedirect("https://www.landgrey.me")
Paylaod:
https://www.landgrey.me/redirect.php?url=http://www.evil.com/www.landgrey.me
或者买个xxxlandgrey.me域名，然后绕过:
https://www.landgrey.me/redirect.php?url=http://xxxlandgrey.me
3.3、可信站多次重定向绕过
A、利用已知可重定向到自己域名的可信站点的重定向，来最终重定向自己控制的站点,一种是利用程序自己的公共白名单可信站点,
Payload:
https://www.landgrey.me/redirect.php?url=https://www.baidu.com/linkurl=lMwwNDM6ahaxKkSFuOG
B、另一种类似，但是程序的跳转白名单比较严格，只能是自己域的地址，这时需要有一个目标其它域的任意跳转漏洞，比如https://auth.landgrey.me/jump.do?url=evil.com，然后测试时
Payload:
https://www.landgrey.me/redirect.php?url=https://auth.landgrey.me/jump.do?url=evil.com
其他方式
1. 跳转到IP地址，而不是域名;
2. 跳转到IPV6地址，而不是IPv4地址;
3. 将要跳转到的IP地址用10进制、8进制、16进制形式表示;
4. 更换协议,使用ftp、gopher协议等;
5. 借鉴SSRF漏洞绕过的tricks;
6. CRLF注入不能xss时，转向利用任意URL跳转漏洞;
环境测试
Bwapp
DVWA
Google Hacking
inurl:go
inurl:return
inurl:r_url
inurl:returnUrl
inurl:returnUri
inurl:locationUrl
inurl:goTo
inurl:return_url
Bypass-payload:
https://github.com/cujanovic/Open-Redirect-Payloads.git
参考
https://landgrey.me/open-redirect-bypass/
https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.md