隐身术——使用****Tor****匿名网络与****proxychains****构建****SOCKS****代理链****IP****跳板，在渗透测试与攻击入侵时销声匿迹

**\*\*\*\*\*免责声明\*\*\*\*\***

**此处介绍的技术仅作为研究，学习，交流用，严禁任何个人，机构，公司，组织，团体使用这里公布的技术对任何类似上述的第三方实体进行未经授权，非法的渗透测试与攻击入侵，同时本博文作者不承担由此造成目标的任何（包括但不限于）经济损失，用户数据丢失或泄漏，业务亏损等后果产生的相应法律责任**

用最简单的语言来描述，Tor 匿名网络是一个由位于世界各地的志愿者维护的各自的匿名网络组成的大型分布式匿名网络，类似于分布式的 VPN，但它是免费，开放的，而且其规模之大超乎想象

Tor 匿名网络使用的核心技术是美国海军研究室开发的第三代洋葱路由器系统，其初衷用于保护政府机关的数据通信隐私，现在它被广泛应用在任何民间企业，组织，机构，以及家庭，个人的安全数据传输等场合

为了切合主题以及限于篇幅，本博文不对 Tor 匿名网络的技术实现细节，它与 TCP/IP 之间的关系，以及与 IPSec，PPTP，L2TP，MPLS，等 VPN 技术之间的异同做分析对比，对 Tor 内幕感兴趣的，可以自行浏览其项目官方站点：

[https://www.torproject.org/](https://www.torproject.org/" \o "https://www.torproject.org/" \t "http://blog.csdn.net/yinchangxin/article/details/_blank)

Tor 匿名网络不仅可以保护你的网络通信不被“监听”，同时也可以在访问站点时，让对方无从判断你的真实物理地址，因为所有你访问目标站点的数据包，其 IP头部的源地址，都是某个 Tor 网络出口节点的 IP，它已经替换掉你的 ISP 给你分配的公网 IP，

这个机制类似于 windows 平台上被广泛使用的，用于“加速”网络游戏，以及“翻墙”的“非凡网络加速器”，后者实质上就是连接到某个非凡 VPN 节点的客户端程序，但它的 VPN 服务是收费的，尽管它也提供免费的测试账户连接到测试服务器，但相同的“原始” IP ，一天仅能连接3次，每次20分钟，超时即断开，

这对于要长期且稳定进行渗透测试计划而言，不是很方便，而且需要随时注意连接状态，否则真实 IP 就会曝露，尽管如此，对于有明确目标和丰富经验的攻击者而言，总共一小时的时间也足够作坏事了，关于非凡网络加速器的使用教程，以后有机会再发布

回到主题，测试环境为 BackTrack 5 Release 3  32-bits ，首先，将浏览器指向

[https://www.torproject.org/download/download.html.en](https://www.torproject.org/download/download.html.en" \o "https://www.torproject.org/download/download.html.en" \t "http://blog.csdn.net/yinchangxin/article/details/_blank)

下载 GNU/Linux ，BSD ，UNIX 通用的 “Tor Browser”软件包，

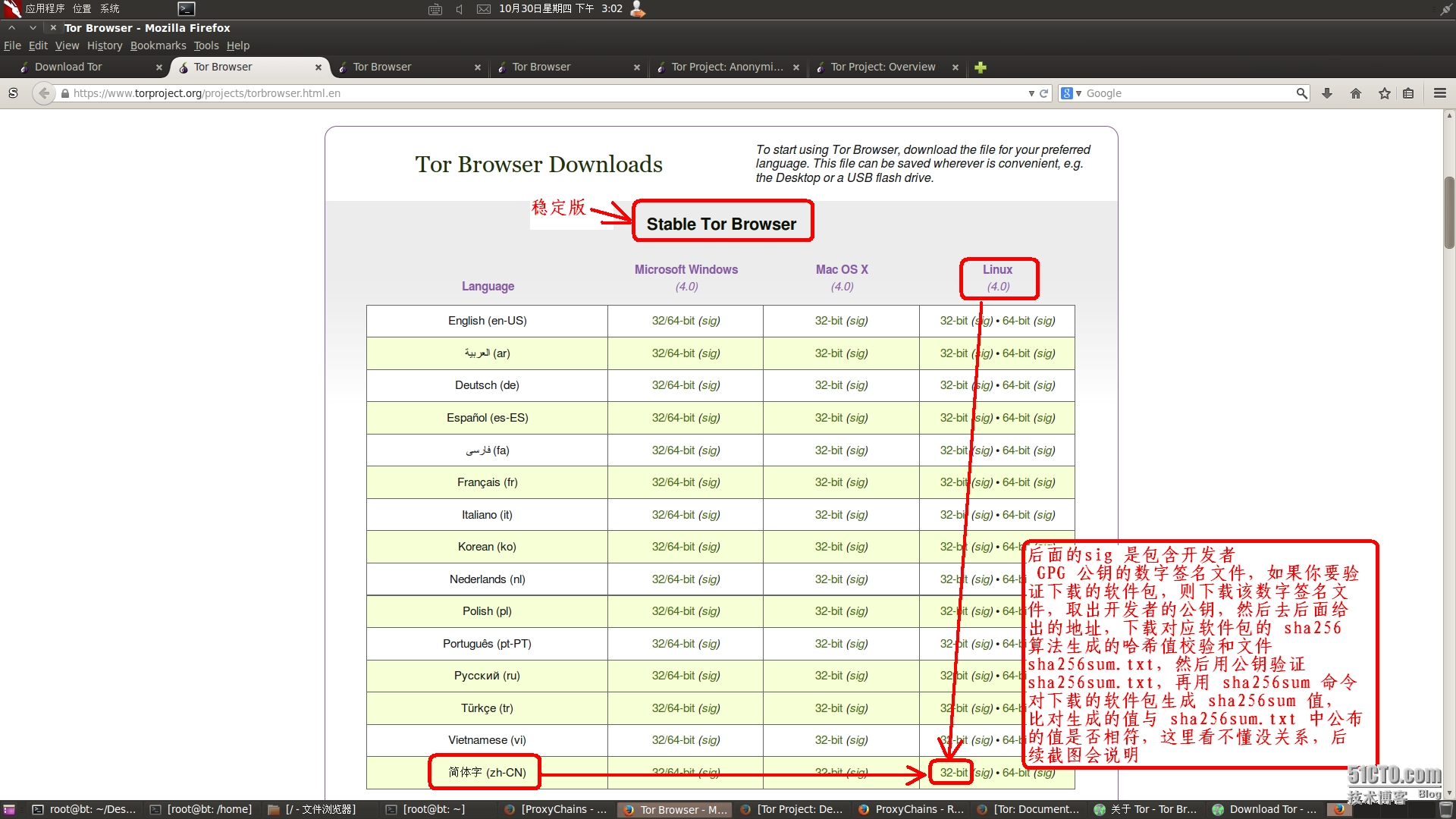
例如，我们下载的软件包名称为

**tor-browser-linux32-4.0\_zh-CN.tar.xz**

（也可以访问下面这个页面，根据你的处理器以及操作系统版本，体系结构，选择相应的版本下载：

[https://www.torproject.org/projects/torbrowser.html.en](https://www.torproject.org/projects/torbrowser.html.en" \o "https://www.torproject.org/projects/torbrowser.html.en" \t "http://blog.csdn.net/yinchangxin/article/details/_blank)

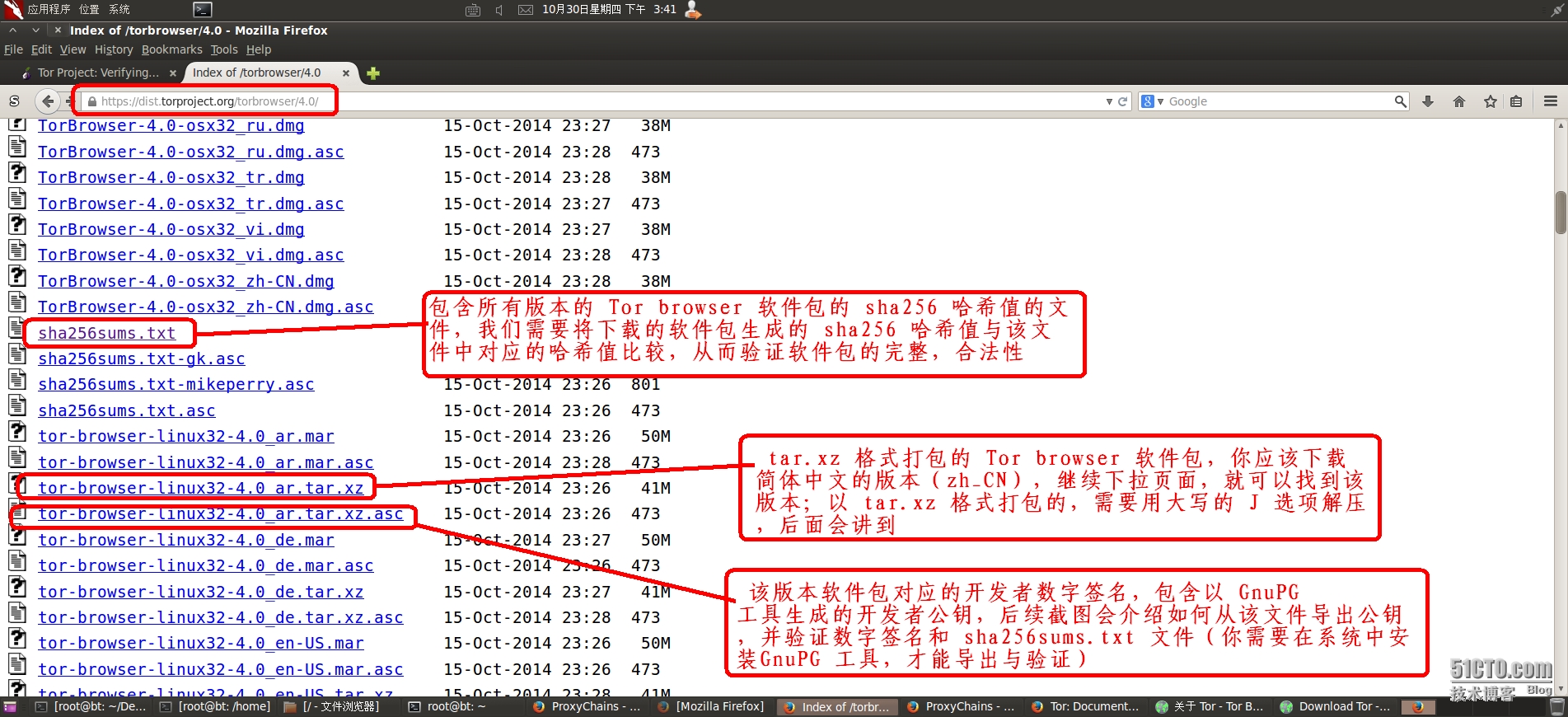
）



**（或者，从下面这个地址，一次性下载适用于你系统的 Tor browser 版本，对应的 sha256sums.txt  ，以及开发者数字签名这三个文件（如果你要验证软件包来源的合法性）**

[https://dist.torproject.org/torbrowser/4.0/](https://dist.torproject.org/torbrowser/4.0/" \o "https://dist.torproject.org/torbrowser/4.0/" \t "http://blog.csdn.net/yinchangxin/article/details/_blank)

**）**

[](http://s3.51cto.com/wyfs02/M02/4D/7B/wKioL1RR8Yfh-0EcABBymdlD6TM047.jpg)

一，进入下载目录：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | root@bt:~/Desktop# pwd  /root/Desktop    root@bt:~/Desktop# ls -alit  1451874 -rw-r--r--  1 root     root          473 2014-10-29 23:40 tor-browser-linux32-4.0\_zh-CN.tar.xz.asc  1451855 -rw-r--r--  1 root     root        12074 2014-10-29 23:39 sha256sums.txt  1451865 -rw-r--r--  1 root     root     43209248 2014-10-29 23:20 tor-browser-linux32-4.0\_zh-CN.tar.xz |

二，从 mozilla 的 公钥服务器上，下载软件包开发者的公钥：

|  |  |
| --- | --- |
| 1  2  3  4  5 | root@bt:~/Desktop# gpg --keyserver keys.mozilla.org --recv-keys 0x29846B3C683686CC  gpg: 下载密钥‘683686CC’，从 hkp 服务器 keys.mozilla.org  gpg: 密钥 683686CC：“Mike Perry (Regular use key) <mikeperry@fscked.org>”未改变  gpg: 合计被处理的数量：1  gpg:              未改变：1 |

三，以该公钥验证数字签名文件 ****tor-browser-linux32-4.0\_zh-CN.tar.xz.asc**** 与 软件包哈希值文件 ****sha256sums.txt**** ：

|  |  |
| --- | --- |
| 1 | root@bt:~/Desktop# gpg --verify tor-browser-linux32-4.0\_zh-CN.tar.xz.asc sha256sums.txt |

四，验证通过后，对 ****tor-browser-linux32-4.0\_zh-CN.tar.xz**** 生成 sha256 校验和，并且将其与 ****sha256sums.txt**** 中相应的值进行比对：

|  |  |
| --- | --- |
| 1  2  3  4  5 | root@bt:~/Desktop# sha256sum ./tor-browser-linux32-4.0\_zh-CN.tar.xz  e41fb3dfe6464d45be8a8a40d6f80ab0806336161e3dfefeefa42437d870c932  ./tor-browser-linux32-4.0\_zh-CN.tar.xz    root@bt:~/Desktop# vim sha256sums.txt  e41fb3dfe6464d45be8a8a40d6f80ab0806336161e3dfefeefa42437d870c932  tor-browser-linux32-4.0\_zh-CN.tar.xz |

验证通过，至此，准备工作完成

五，解压 tor-browser-linux32-4.0\_zh-CN.tar.xz 软件包，这会在当前目录下生成一个叫做 ****tor-browser\_zh-CN**** 的子目录，默认的目录所有者，所属组均为****postgres****，

必须指出，Tor 项目官网的安装指南中，建议不要以 root 用户解压该软件包，但是，如果以其它低权限普通用户解压，则 tar 命令提示报错，找不到目录，因此，我们以 root 用户解压该软件包，****然后将解包后的目录的所有者，所属主改成你系统上已经存在的任意普通用户即可****，当然，如果你系统上的 postgres 账户可用，那么也可以按照默认设置

假设我们的软件包（以及数字签名，哈希值文件）存放在  ****/download****  目录：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | root@bt:~# pwd  /root    root@bt:~# cd /download/  root@bt:/download# ls -alit  1451874 -rw-r--r--  1 root     root          473 2014-10-29 23:40 tor-browser-linux32-4.0\_zh-CN.tar.xz.asc  1451855 -rw-r--r--  1 root     root        12074 2014-10-29 23:39 sha256sums.txt  1451865 -rw-r--r--  1 root     root     43209248 2014-10-29 23:20 tor-browser-linux32-4.0\_zh-CN.tar.xz    root@bt:/download# tar -xvJf tor-browser-linux32-4.0\_zh-CN.tar.xz  root@bt:/download# ls -ld tor-browser\_zh-CN/  drwx------ 3 postgres postgres 4096 2000-01-01 08:00 tor-browser\_zh-CN/ |

注意，上面给 tar 命令传递大写的 J 参数，用于解压****.zx****格式的包

由于我的 BackTrack5 系统上默认没有任何普通用户账户，因此，先创建一个普通用户 ****shayi2**** ，并且指定其“家”目录，这一点很重要，如果没有指定家目录，后面在以该普通用户运行，启动 Tor 的时候会报错（ Tor 官方建议不要以 root 用户身份运行 Tor），因为 Tor 需要在启动它的用户的家目录中生成包含配置和统计信息的文件：

|  |  |
| --- | --- |
| 1  2  3  4  5 | root@bt:/download# useradd shayi2 -d /home/shayi2  root@bt:/home# passwd shayi2  输入新的 UNIX 密码：  重新输入新的 UNIX 密码：  passwd：已成功更新密码 |

在 ubuntu 系统上，有时 useradd 命令的 -d 参数不能成功创建用户家目录，因此，我们以 mkdir 命令，手动创建该目录，并将该目录的所有者，所属主递归改为

shayi2

|  |  |
| --- | --- |
| 1  2  3  4 | root@bt:/download# mkdir /home/shayi2  root@bt:/download# chown -R shayi2:shayi2 /home/shayi2  root@bt:/download# ls -ld /home/shayi2  drwxr-xr-x 2 shayi2 shayi2 4096 2014-10-30 22:51 /home/shayi2 |

将 /download/tor-browser\_zh-CN  目录的所有者，所属主，递归改成 shayi2 用户：

|  |  |
| --- | --- |
| 1  2  3 | root@bt:/download# chown -R shayi2:shayi2 tor-browser\_zh-CN/  root@bt:/download# ls -ld  tor-browser\_zh-CN/  drwx------ 3 shayi2 shayi2 4096 2000-01-01 08:00 tor-browser\_zh-CN/ |

六，由于我们需要切换到 shayi2 用户运行 Tor ，但是当前的 X-Server 是以 root 用户身份运行的，切换后将无法以 shayi2 用户的身份运行基于 X-Server 环境的 Tor，如果你硬要这么做，X-Server 会给你返回如下的错误提示：

**No protocol specified  
Error: cannot open display: :0.0  
Tor Browser exited abnormally.  Exit code: 1**  
  
于是，我们需要先以 root 用户禁用 X-Server 的访问控制功能，再切换到 shayi2 用户启动 Tor ：

|  |  |
| --- | --- |
| 1  2 | root@bt:/download# xhost +  access control disabled, clients can connect from any host |

命令 ****xhost +****用于关闭 ****X-Server**** 的访问控制；****xhost -**** 用于开启访问控制

前面我们做了这么多的准备工作：创建普通用户及其家目录，改变属主属组，禁用访问控制等等，都是出于安全的考虑，由于 Tor 本身在开发阶段（包括任何网络服务程序）潜在的不安全编码和逻辑缺陷，可能被挖掘而成为 0 day 漏洞，因此，以最小权限运行 Tor 可以保证在漏洞被成功利用时，将攻击者造成的破坏降到最低；

如果你尝试以 root 身份运行 Tor 的启动脚本，将会得到类似下面的错误信息：

**root@bt:/download/tor-browser\_zh-CN# ./start-tor-browser**  **The Tor Browser Bundle should not be run as root.  Exiting.**

如果你完成前面的准备工作，先别急着运行 Tor 来展现你渗透测试方面的才能，

我们先来看看，****tor-browser\_zh-CN**** 这个目录下都有哪些东西：

**tor-browser\_zh-CN/start-tor-browser**

这是一个到  **tor-browser\_zh-CN/Browser/start-tor-browser** 脚本的

符号链接，通过该链接或后者，都可以运行 Tor 启动脚本来启动 **Tor Browser**

**Tor Browser** 是特别定制的 ****Mozilla FireFox**** 浏览器，它已经配置成使用 Tor 客户端监听的 ****socks4****或 ****socks5  127.0.0.1:9150**** 地址，因此就可以使用 Tor 的代理服务，连接到某个 Tor 匿名网络，通过该网络的出口节点的公网 IP，访问因特网

**Tor Browser** 还包括一个叫做 ****Torbutton**** 的“已安装”FireFox 插件，通过该插件在 FireFox 主界面左上方提供的按钮，用户可以更改当前的 Tor 代理设置，支持 HTTP，FTP，SSL，SOCKS 等代理，以及配置一些其它的 Tor 功能，

Tor Browser 自带的另一个插件为 ****NoScript****，这是一个阻止恶意站点的 XSS （跨站脚本）攻击的强大 FireFox 安全插件，关于 ****NoScript**** 的使用介绍，请参考这篇博文：

[http://shayi1983.blog.51cto.com/4681835/1553095](http://shayi1983.blog.51cto.com/4681835/1553095" \o "http://shayi1983.blog.51cto.com/4681835/1553095" \t "http://blog.csdn.net/yinchangxin/article/details/_blank)

****tor-browser\_zh-CN/Browser****

这个目录中包含运行 ****Tor Browser**** 所需的核心动态共享库文件（提供连接到 Tor 匿名网络的功能），配置文件，以及一些插件，例如前面提到的 ****Torbutton**** ，以及****HTTPSEveryWhere****，

**限于篇幅，这里不介绍文件各自的功能以及实现机制（有机会以后再发文章），让我们专注于如何通过使用 Tor 提供的匿名服务，来保护自身的真实 IP 信息**

通过前面的介绍粗略的了解 Tor 以及  ****Tor Browser**** 的工作机制后，我们切换到

shayi2 用户启动 Tor 与 ****Tor Browser****，这是通过运行 ****start-tor-browser****

这个 shell 脚本实现的：

|  |  |
| --- | --- |
| 1  2  3 | root@bt:/download/tor-browser\_zh-CN# su - shayi2  shayi2@bt:~$ cd /download/tor-browser\_zh-CN/  shayi2@bt:/download/tor-browser\_zh-CN$ ./start-tor-browser |

下面是启动过程中，输出到终端的调试信息，我们稍微分析一下这些信息：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | Launching Tor Browser for Linux in /download/tor-browser\_zh-CN/Browser...  Oct 31 00:44:27.311 [notice] Tor v0.2.5.8-rc (git-cc2477cc002ac542) running on Linux with Libevent 2.0.21-stable, OpenSSL 1.0.1i and Zlib 1.2.3.3.  Oct 31 00:44:27.311 [notice] Tor can't help you if you use it wrong! Learn how to be safe at https://www.torproject.org/download/download#warning  Oct 31 00:44:27.311 [notice] Read configuration file "/download/tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/torrc-defaults".  Oct 31 00:44:27.311 [notice] Read configuration file "/download/tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/torrc".  Oct 31 00:44:27.313 [notice] Opening Socks listener on 127.0.0.1:9150  Oct 31 00:44:27.313 [notice] Opening Control listener on 127.0.0.1:9151  Oct 31 00:44:27.000 [notice] Parsing GEOIP IPv4 file /download/tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/geoip.  Oct 31 00:44:27.000 [notice] Parsing GEOIP IPv6 file /download/tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/geoip6.  Oct 31 00:44:27.000 [notice] Bootstrapped 0%: Starting  Oct 31 00:44:27.000 [notice] Bootstrapped 5%: Connecting to directory server  Oct 31 00:44:27.000 [notice] We now have enough directory information to build circuits.  Oct 31 00:44:27.000 [notice] Bootstrapped 80%: Connecting to the Tor network  Oct 31 00:44:27.000 [notice] New control connection opened from 127.0.0.1.  Oct 31 00:44:27.000 [notice] New control connection opened from 127.0.0.1.  Oct 31 00:44:28.000 [notice] Bootstrapped 85%: Finishing handshake with first hop  Oct 31 00:44:29.000 [notice] Bootstrapped 90%: Establishing a Tor circuit  Oct 31 00:44:32.000 [notice] Tor has successfully opened a circuit. Looks like client functionality is working.  Oct 31 00:44:32.000 [notice] Bootstrapped 100%: Done  Oct 31 00:44:33.000 [notice] New control connection opened from 127.0.0.1. |

可以看到，启动时将会读取 tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/torrc-defaults 以及 torrc 这两个配置文件中的信息；并且以  socks 代理的工作模式，监听在地址 127.0.0.1 的 9150 端口；与控制通信相关的模式则监听在地址 127.0.0.1 的 9151 端口；并且通过 /tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/geoip 与 geoip6 这两个文件来解析与 GEOIP 相关的信息；

首先会连接到一个目录服务器，推测其中应该有分布在世界各地的所有 Tor 匿名网络的信息，然后会选择其中一个来建立虚拟链路，当完成与链路中第一个节点的握手后，一个从客户端到某 Tor 匿名网络的虚拟链路就建立完成

查看一下 Tor 以及 ****Tor Browser**** 建立的网络连接：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | root@bt:~# netstat -antupeo  激活Internet连接 (服务器和已建立连接的)  Proto Recv-Q Send-Q Local Address           Foreign Address         State       User       Inode       PID/Program name Timer  tcp        0      0 127.0.0.1:7337          0.0.0.0:\*               LISTEN      1000       6973        1106/postgres.bin 关闭 (0.00/0/0)  tcp        0      0 127.0.0.1:9150          0.0.0.0:\*               LISTEN      1002       60606       9982/tor         关闭 (0.00/0/0)  tcp        0      0 127.0.0.1:9151          0.0.0.0:\*               LISTEN      1002       60607       9982/tor         关闭 (0.00/0/0)  tcp        0      0 127.0.0.1:9151          127.0.0.1:36640         ESTABLISHED 1002       60611       9982/tor         关闭 (0.00/0/0)  tcp        0      0 127.0.0.1:36638         127.0.0.1:9151          ESTABLISHED 1002       59104       9959/firefox     关闭 (0.00/0/0)  tcp        0      0 127.0.0.1:36640         127.0.0.1:9151          ESTABLISHED 1002       61507       9959/firefox     关闭 (0.00/0/0)  tcp        0      0 192.168.1.25:50031      178.254.44.234:9001     ESTABLISHED 1002       60608       9982/tor         关闭 (0.00/0/0)  tcp        0      0 127.0.0.1:9151          127.0.0.1:36638         ESTABLISHED 1002       60610       9982/tor         关闭 (0.00/0/0)  tcp6       0      0 ::1:7337                :::\*                    LISTEN      1000       6972        1106/postgres.bin 关闭 (0.00/0/0)  udp6       0      0 ::1:34587               ::1:34587               ESTABLISHED 1000       6986        1106/postgres.bin 关闭 (0.00/0/0) |

可以看到，启动了一个包含 5 个线程的 tor 进程，其中几个监听在本地 9151 端口的 tor 线程建立了控制通信连接，对应于前面调试信息中的 New control connection opened from 127.0.0.1

其中一个 tor 线程建立了到 178.254.44.234:9001 的 TCP 连接，该连接起到的作用至关重要，类似于到 VPN 服务器端的 Tor 匿名网络虚拟链路

启动了一个包含 2 个线程的 firefox 进程，****这个 firefox 实际上就是经过特殊配置，使用 tor 的代理服务功能，并且集成了 Torbutton ，Noscript 插件的 Tor**Browser**

查看这2个进程之间的关系：

|  |  |  |
| --- | --- | --- |
| 1  2  3 | root@bt:~# ps -ef |grep 9959  shayi2    9959  9942  0 00:44 pts/2    00:00:06 ./firefox --class Tor Browser -profile TorBrowser/Data/Browser/profile.default  shayi2    9982  9959  0 00:44 pts/2    00:00:02 /download/tor-browser\_zh-CN/Browser/TorBrowser/Tor/tor --defaults-torrc /download/tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/torrc-defaults -f /download/tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/torrc DataDirectory /download/tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor GeoIPFile /download/tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/geoip GeoIPv6File /download/tor-browser\_zh-CN/Browser/TorBrowser/Data/Tor/geoip6 HashedControlPassword 16:74ceffa486896c056023b5ea65d0bbfc9208e6903fdce290c7d2a56a37 \_\_OwningControllerProcess 9959 |  |

可以看到，firefox 进程先启动，带有特殊的参数“--class Tor Browser”

以及“-profile TorBrowser/Data/Browser/profile.default”

然后启动一个 tor 子进程，对应命令为 /download/tor-browser\_zh-CN/Browser/TorBrowser/Tor/tor

这是我们前面提到的动态连接库文件，它完成实际的连接 Tor 匿名网络的任务，

并且带有很多参数：

两个 tor 配置文件 torrc-defaults 与 torrc 的路径；

两个 GEOIP 数据文件的路径（前面的调试信息有输出）；

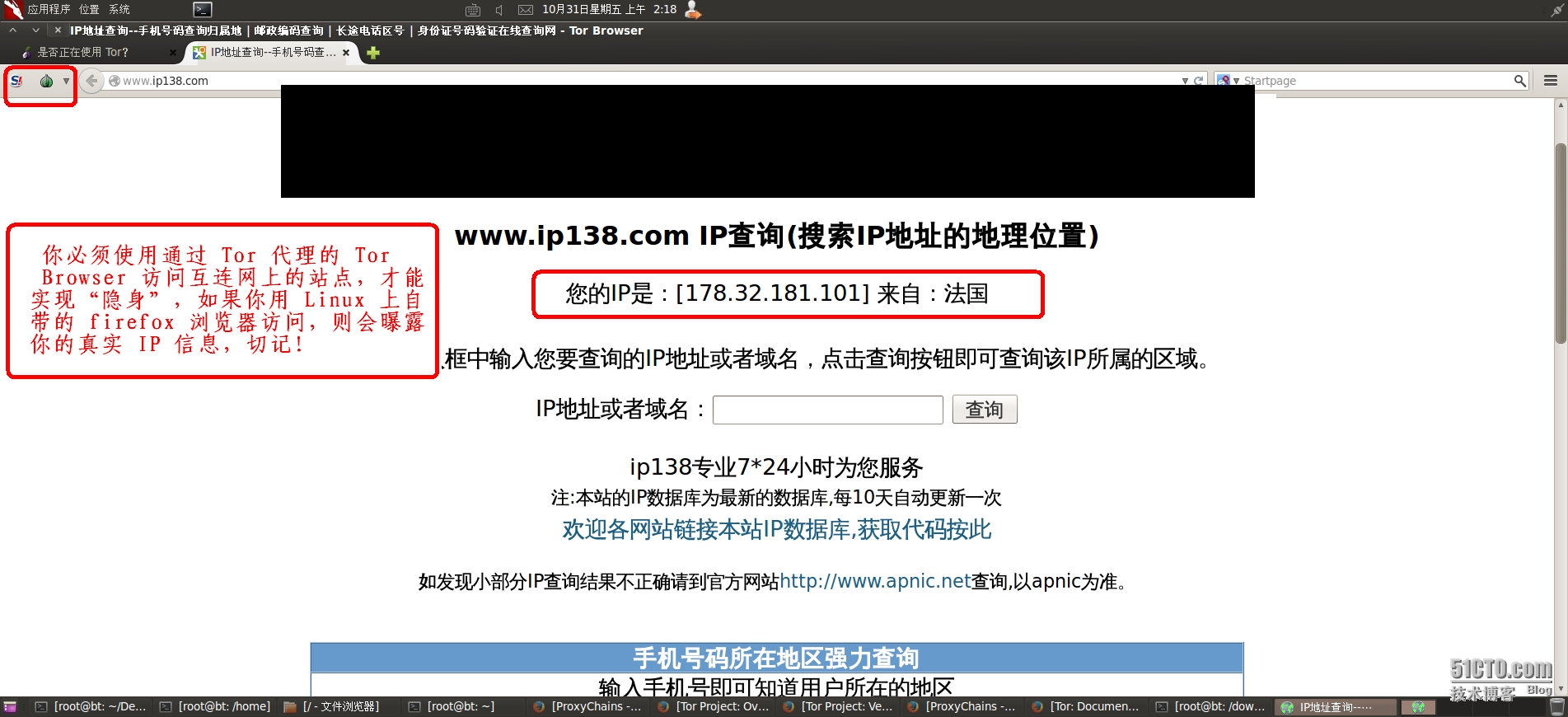
用于在控制连接通信中验证身份的哈希密码；

如前所述，这 2 个进程会创建各自的线程，然后打开网络连接

如果你的系统成功的启动了 ****Tor Browser**** ，并且连接到某个 Tor 网络，那么 ****Tor Browser**** 应该会给出类似下面的页面：

[](http://s3.51cto.com/wyfs02/M00/4D/84/wKiom1RSfjixXJjMAA2s_MNG4g8133.jpg)

[](http://s3.51cto.com/wyfs02/M01/4D/84/wKioL1RSgPOj_sv8AAibV1SI4OQ084.jpg)

[](http://s3.51cto.com/wyfs02/M01/4D/84/wKiom1RSglLgTKWfAAhzrqGmKqE969.jpg)

**前面我们提到，windows 平台上的非凡网络加速器，也可以实现隐身，缺点在于它的试用时间有限，但是它可以让所有的“流量”通过 VPN 转发出去，换言之，只要运行了其客户端软件，在试用时间内，你使用的所有网络应用程序，**

**包括 web 浏览器，QQ，在线电影播放器，网络游戏客户端，LOIC（低轨道离子加农炮，一种由 Anonymous 黑客组织开发并在早期使用的进行 DDOS 攻击的工具）**

**所有包括但不限于上面列出的网络应用程序，一概使用某个非凡 VPN 出口节点的 IP ，这就是非凡 VPN 会拥有很多付费用户的原因，国内付费用户可以连接到它的一个在国外的节点，实现“翻墙”访问 youtube，facebook 等站点，类似地，国外付费用户可以连接到它的一个在国内的节点，绕过百度视频，优库视频等站点对国外 IP 的访问限制，再也不会出现蛋疼的“因为版权原因，该视频只允许在内地播放。。。。”提示信息**

**这里不是在帮非凡 VPN 广告宣传，而只是为了强调，在 Linux 平台上，我们也可以实现类似的梦想，这就需要结合 Tor 与 proxychains 两个工具，构建“代理链”，然后再运行你的网络应用程序，就能实现任何“流量”通过 Tor 匿名网络“转发”（对于渗透与入侵任务至关重要），下面介绍 proxychains 从源码编译，安装，以及结合  Tor 的使用**

\*\*\*\*\*下面的操作是可选的，具体来说，如果是 ****BackTrack 5 Release 3****，则系统默认已经在  ****/bin/proxychains****  目录下自带了该工具；并且在****/etc/proxychains.conf**** 目录下提供了配置文件，只需要修改配置文件以及运行 proxychains 即可（****Release 2**** 以前的版本我没试过，你可以先在系统上搜索看看）

[](http://s3.51cto.com/wyfs02/M00/4D/8E/wKiom1RTMpiyUHn4AAw4dhnN4YA127.jpg)

如果你的系统是基于 ****RedHat**** 或者 ****Fedora**** 的发行版，例如 ****CentOS****，因为默认没有安装 proxychains ，因此必须进行如下操作：

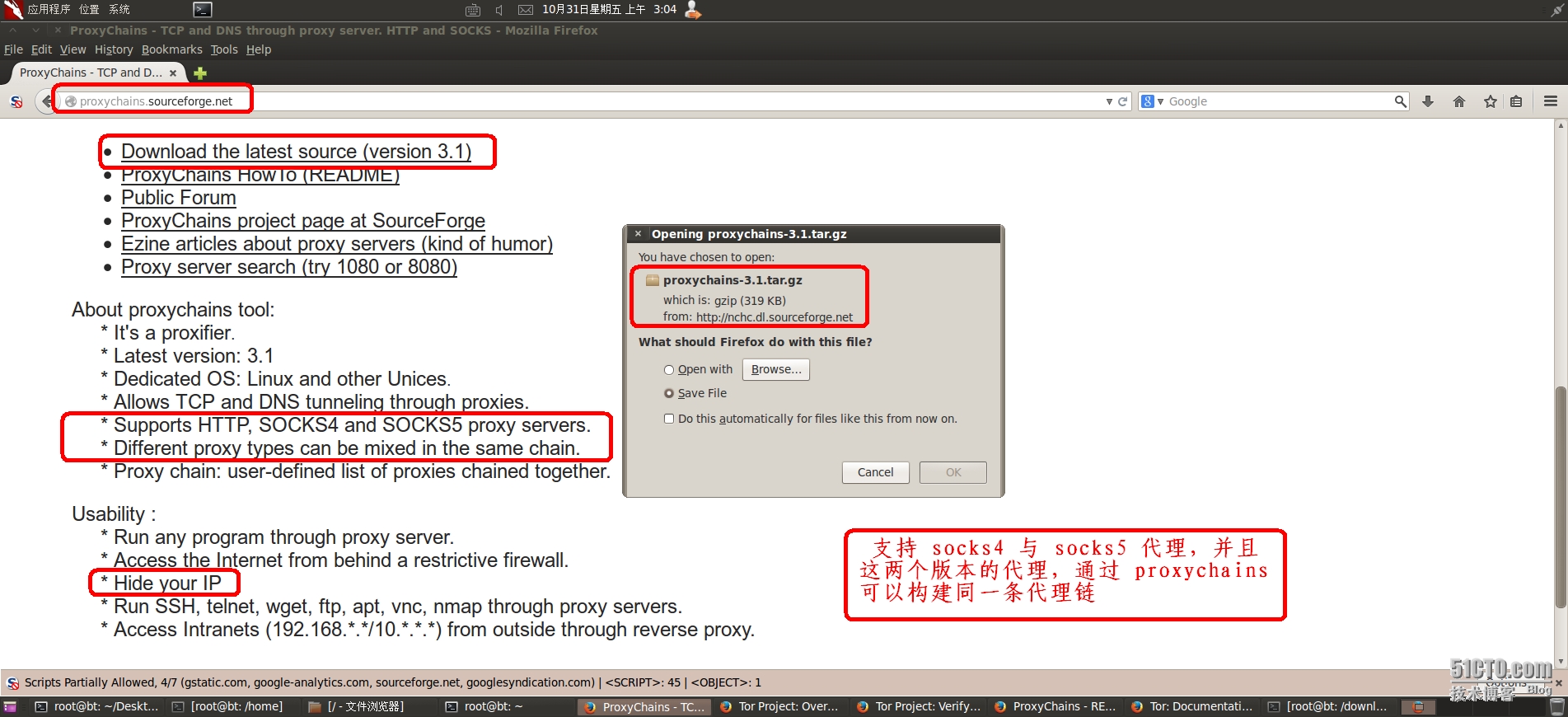
打开你的普通 FireFox 浏览器，将地址指向

[http://proxychains.sourceforge.net/](http://proxychains.sourceforge.net/" \o "http://proxychains.sourceforge.net/" \t "http://blog.csdn.net/yinchangxin/article/details/_blank)

（如果你安装了 Noscript 插件，可能需要暂时禁用 Noscript，或者临时允许该****域****的 javascript 脚本运行）

这是开源项目 proxychains 在 sourceforge.net 上的主页

下载它的当前最新版（3.1）的 .tar.gz 格式的源码包，你应该进行类似如下截图的操作：

[](http://s3.51cto.com/wyfs02/M02/4D/84/wKioL1RSjZ_hRAVfAAri8cbkVcA776.jpg)

一，切换回 root 用户，进入前面的下载目录 /download ，解压下载的 proxychains 源码包：

|  |  |
| --- | --- |
| 1  2 | root@bt:/download# tar -zxvf proxychains-3.1.tar.gz  root@bt:/download# cd ./proxychains-3.1 |

二，进入解包后生成的子目录，阅读其中的 INSTALL 文件：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | root@bt:/download/proxychains-3.1# vim INSTALL    ProxyChains ver 3.1 Installation  =======================    If you have installed other ver of proxychains  "make uninstall" before installing this distribution    unpack the .tar.gz , 'cd' to distribution directory and  run following commands (as root)  -------------------------------------  ./configure  make  make install  -------------------------------------  that's all |

再也没有比这里更简洁的编译安装步骤了，注意，如果此前安装了 proxychains 的旧版，需要先执行 make uninstall 命令来卸载

三，配置，编译，安装三部曲的一次到位：

先确认你的系统上是否安装了 gcc ，如果没有，可以通过 yum 来安装，不建议从源码安装 gcc ，因为过程太繁琐，而且容易出错，如果你执意要从源码编译安装 gcc ，这两篇博文提供了思路清晰的完整攻略：

[http://shayi1983.blog.51cto.com/4681835/1532646](http://shayi1983.blog.51cto.com/4681835/1532646" \o "http://shayi1983.blog.51cto.com/4681835/1532646" \t "http://blog.csdn.net/yinchangxin/article/details/_blank)

[http://shayi1983.blog.51cto.com/4681835/1531682](http://shayi1983.blog.51cto.com/4681835/1531682" \o "http://shayi1983.blog.51cto.com/4681835/1531682" \t "http://blog.csdn.net/yinchangxin/article/details/_blank)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | root@bt:/download/proxychains-3.1# gcc --version  gcc (Ubuntu 4.4.3-4ubuntu5.1) 4.4.3  Copyright (C) 2009 Free Software Foundation, Inc.  This is free software; see the source for copying conditions.  There is NO  warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.    root@bt:/download/proxychains-3.1# ./configure && make && make install    root@bt:/download/proxychains-3.1# whereis proxychains  proxychains: /bin/proxychains /etc/proxychains.conf |

四，配置 proxychains 使用 Tor 代理，构建代理链：

首先阅读前面解包生成子目录下的 README 文档，下面删除了无关紧要的部分内容：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44 | root@bt:/download/proxychains-3.1# vim README    This version (3.x) introduces support for DNS resolving through proxy  it supports SOCKS4, SOCKS5 and HTTP CONNECT proxy servers.  Auth-types: socks - "user/pass" , http - "basic".    Some cool features:    \* This program can mix different proxy types in the same chain          like: your\_host <-->socks5 <--> http <--> socks4 <--> target\_host  \* Different chaining options supported          random order from the list ( user defined length of chain ).          exact order  (as they appear in the list )          dynamic order (smart exclude dead proxies from chain)  \* You can use it with any TCP client application, even network scanners          yes, yes - you can make portscan via proxy (or chained proxies)          for example with Nmap scanner by fyodor (www.insecire.org/nmap).          proxychains nmap -sT -PO -p 80 -iR  (find some webservers through proxy)  \* You can use it with servers, like squid, sendmail, or whatever.  \* DNS resolving through proxy.    Configuration:  proxychains looks for config file in following order:  1)      ./proxychains.conf  2)      $(HOME)/.proxychains/proxychains.conf  3)      /etc/proxychains.conf  \*\*  \*\*see more in /etc/proxychains.conf    Usage Example:            bash$ proxychains telnet targethost.com    in this example it will run telnet through proxy(or chained proxies)  specified by proxychains.conf    Usage Example:            bash$ proxyresolv targethost.com    in this example it will resolve targethost.com through proxy(or chained proxies)  specified by proxychains.conf    NOTE:  to run suid/sgid programs(like ssh) through proxychains you have to be root |

总结一下上面的内容，

1. proxychains 可以连接使用两种不同代理版本（socks4 与 socks5）的代理服务器，来构建代理链；

2. 支持 用户名/密码 形式的代理验证，以及 http 协议规范中的 basic 验证；

3. 可以通过 proxychains 提供的代理中继服务，运行 NMAP 端口扫描工具；

4. 支持向网络服务程序，如 squid，sendmail 提供代理中继服务；

5. proxyresolv 命令，用于通过代理链解析域名，类似于 dig 与 nslookup 的代理版本，

（实际上，proxyresolv 是一个 shell 脚本，它通过内置的默认 DNS IP ：4.2.2.2 调用 dig 对其发起查询请求；）

proxychains 与 proxyresolv 根据配置文件中的信息，来构建代理链并工作，配置文件位于文件系统中三个地方，分别是安装目录（默认安装在 /download/proxychains-3.1/proxychains 目录下）下的 proxychains.conf

以及  ****/etc/proxychains.conf****

我们只关注这两个配置文件，第三个不重要，可以忽略

五，编辑  ****/etc/proxychains.conf**** ，将其中的 ****[ProxyList]**** 项的条目

socks4  127.0.0.1 9050  更改为  ****socks5 127.0.0.1 9150**** ，这是前面要你记录下的 Tor Browser 使用的代理配置信息，保存退出：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | root@bt:/download/proxychains-3.1# vim /etc/proxychains.conf    [ProxyList]  # add proxy here ...  # meanwile  # defaults set to "tor"  socks5  127.0.0.1 9150 |

如果不能通过 ****proxychains**** 连接 Tor 匿名网络，还需要配置**/download/proxychains-3.1/proxychains/proxychains.conf** 将其改成与****/etc/proxychains.conf****相同的代理服务器配置即可，

一般而言，执行 ****/bin/proxychains**** 与 ****/bin/proxyresolv**** ，会读取****/etc/proxychains.conf**** 这个全局配置文件中的信息

六，一切就绪后，切换到 shayi2 用户，确保已经通过执行 start-tor-browser 脚本，连接到某个 Tor 匿名网络，然后在一个新的终端

（start-tor-browser 脚本以前台形式运行，它会占用一个 shell 终端）

以 root 身份运行 proxychains ，后面接上你要隐藏 IP 的应用程序，例如 NMAP：

|  |  |
| --- | --- |
| 1 | root@bt:/# /bin/proxychains nmap 192.168.1.35 |

**\*\*\*\*\*在真实场景的扫描中，应该将这里的内网 IP ，替换成你的目标公网 IP ，在目标上的防火墙日志中，记录的 IP 会是你连接到地球上某个 Tor 匿名网络的出口节点的 IP，试想，除非目标由于这次渗透造成巨额经济损失因而报案申请进行计算机取证，找出入侵者的来源，**

**有经验的网络警察通过受害者提供的防火墙日志中的 IP 信息，应该会认为这个 IP 仅是跳板，侦查小组可能会将线索转向为受害者提供接入服务的 ISP ，要求 ISP 提供在入侵时间的路由信息，例如，假设入侵者的真实 IP 在国内，但防火墙日志的 IP 显示来自于法国，那么侦查小组根据发生入侵的时间，来查找国内所有 ISP 的路由信息，包括电信，联通，移动，教育网。。。（这会是一个费时费力的艰巨任务)**

**因为攻击者无论再如何伪装，二进制的 0 与 1 信号势必要通过国内的主干网络链路传输，才能达到匿名网络，并建立虚拟链路，**

**侦查小组可能会查找所有 ISP 路由器上面的日志记录，找出在入侵发生时刻，那些 IP 头部的目标地址是法国 IP 地址的数据包（或者目标地址是一些“出名”代理服务器 IP 的数据包），其源地址信息；通过 ISP 的那哪个路由器上的哪个端口进入并转发的；该“入口”连接到什么网段；AS 自治域号码是多少，通过类似的一步步回溯链路，最终有可能找出入侵者的国内真实 IP（及其物理位置），**

**但如前所述，这会是一个艰巨耗时的任务，为入侵者“转移阵地”提供了充足的缓冲时间，**

**更进一步，当前无线网络技术 802.11 a/b/g/n 广泛应用，WIFI 热点， 无线 AP 接入点遍地开花，能破解 WPA2-PSK 密钥的工具随手可得，如果入侵者仅仅是通过破解的某个无线网络作为跳板，这会极大的增加追踪真实 IP 与物理位置的难度，**

**道理非常简单，入侵者破解某个提供无线接入服务的机构，公司，组织的无线密钥，然后进入其无线网络，横向渗透到内部有线网络，并以其公网 IP 作为跳板再入侵，你只能查到这个跳板的 IP ，根本无法通过该 IP 对应的物理地址抓到人，这就是“战争驾驶”强大的原因**

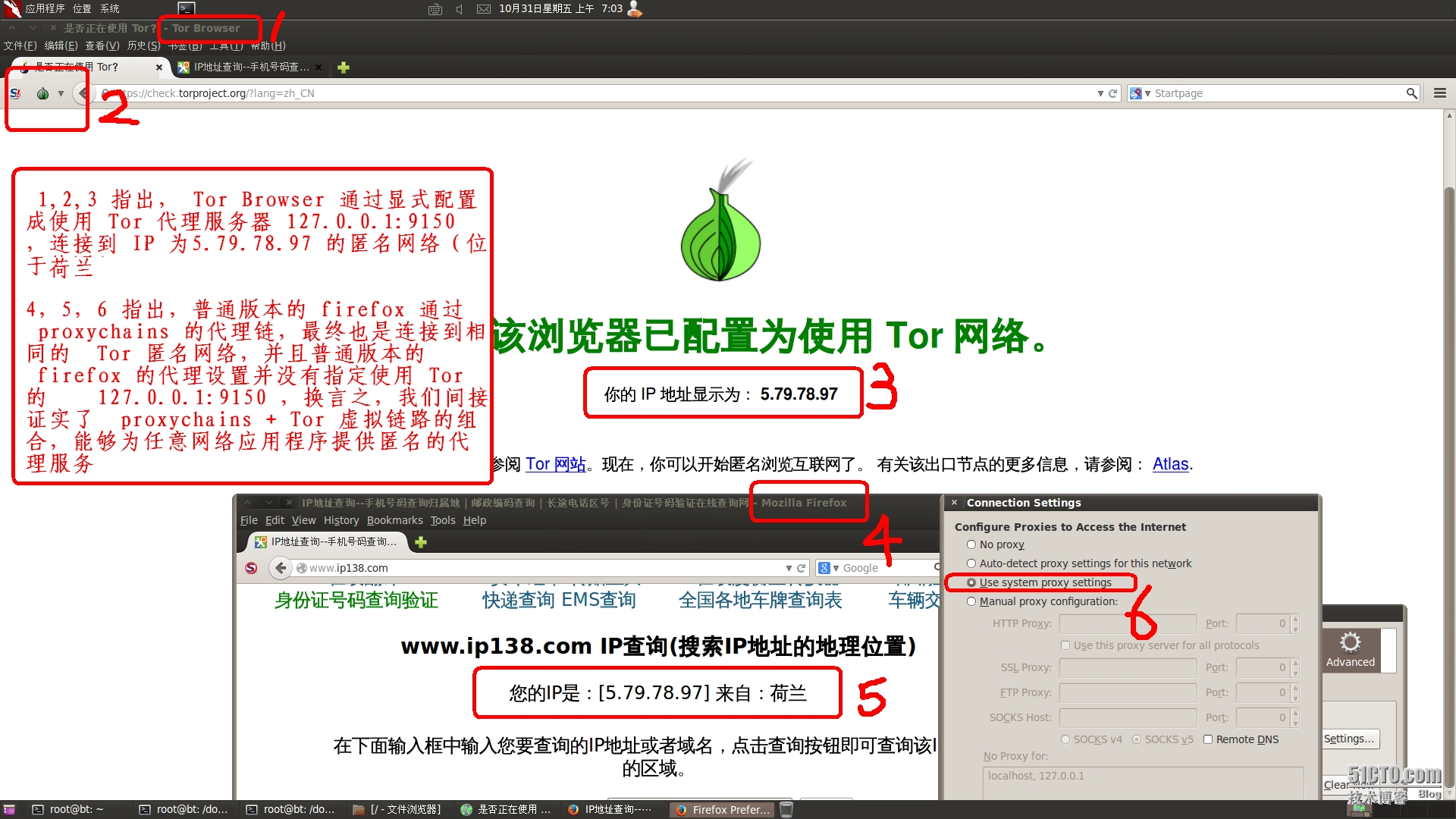
回到主题上来，前面使用连接到匿名网络的 NMAP 扫描，也许不能直观地验证 proxychains  究竟有没有向 NMAP 提供代理服务，因此我们通过直接以 proxychains 运行“普通版本的 firefox”，因为正常的 firefox 没有配置使用任何代理，如果它访问类似像 ****ip138**** 站点时，对方给出的 IP 信息与使用代理的 Tor Browser 中的 IP 一致，那就说明 proxychains 确实能通过代理链转发任何“流量”

顺便一提，有托管主机条件的童鞋，可以在你的远程  SSH 登录客户端上，以 proxychains 执行远程登录，然后查看 SSH 服务器上的防火墙日志，查看其登录 IP ，是否已经变更为你的客户端连接到的某个 Tor 匿名网络的 IP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | root@bt:/download/proxychains-3.1# /bin/proxychains firefox    ProxyChains-3.1 (http://proxychains.sf.net)  |S-chain|-<>-127.0.0.1:9150-<><>-127.0.0.1:16001-<--timeout  |DNS-request| www.backtrack-linux.org  |DNS-request| www.offensive-security.com  |S-chain|-<>-127.0.0.1:9150-|S-chain|-<>-127.0.0.1:9150-<><><>-4.2.2.2:53-<>-4.2.2.2:53-<><>-OK  <><>-OK  |DNS-response| www.backtrack-linux.org is 198.58.126.105  |DNS-response| www.offensive-security.com is 198.50.176.211  |DNS-request| localhost  |S-chain|-<>-127.0.0.1:9150-<><>-4.2.2.2:53-<><>-OK  |DNS-response| localhost is 127.0.0.1 |

需要以****绝对路径****的形式，来执行 proxychains ，这样它会读取****/etc/proxychains.conf****  这个配置文件中的代理链设置

如果产生与上面类似的调试输出信息，说明 proxychains 已经成功将普通的 firefox 添加到它的代理链中，下面以截图来验证：

[](http://s3.51cto.com/wyfs02/M00/4D/84/wKiom1RSyLmi230iAA199TIa290703.jpg)

至此，我们实现了“隐身术”，再次提醒，****慎用这里介绍的技术****，因为它不是万能的，假设为了打击网络犯罪，每个 Tor 匿名网络都积极配合各国的公安机关，警察局等单位，提供连接到它们链路的 IP 信息，那么找到“你”只是时间上的问题而已，此外，前面提到，应该关注“保持匿名小技巧”页面的内容，养成良好的使用 Tor 的习惯，避免在无意中曝露你的真实 IP

对于使用****wireshark****来揭露 Tor 的内部机制，以及对 Tor 官网中一些实用信息的翻译，将总结，整理在今后的另一篇博文