

2012 年全国大学生电子设计竞赛

—信息安全技术专题邀请赛作品设计报告

网络隐身系统 Cyber Stealth System



参赛学校： 西安电子科技大学

参赛队员： 张子兼 高小青 朱利军

指导老师： 张卫东

2012 年 8 月

2012 年全国大学生电子设计竞赛信息安全技术专题邀请赛

作品原创性声明

本人郑重声明：所呈交的参赛作品报告，是本人和队友独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果，不侵犯任何第三方的知识产权或其他权利。本人完全意识到本声明的法律结果由本人承担。

参赛队员签名

	年 月 日
	年 月 日
	年 月 日

指导教师签名

	年 月 日
--	-----------------

2012 年全国大学生电子设计竞赛信息安全技术专题邀请赛

作品简介

参赛学校	西安电子科技大学		
参赛队员			
指导教师			
作品题目	网络隐身系统 Cyber Stealth System		
作品简介	<p>网络隐身系统是一款让计算机正常通信的同时，能够安全隐身于网络中的防护软件。它与以往的安全防护软件相比，不再是被动滞后地应对威胁，而是采用新的安全模型。做到提前预判安全隐患，主动伪装自己并对攻击者进行混淆干扰，从而达到安全置身于网络。由于实现了先知先觉，防患未然，化被动防御为主动隐身，从而能够更好地为我们网络遨游保驾护航。</p> <p>网络隐身系统利用提前预判机制预防未知安全问题, 在 Linux 底层进行栈指纹混淆以干扰扫描软件认知，利用 IP 地址跳变策略，使攻击者的目标千变万化难以确定, 结合 libpcap 和 libnet 伪造网络流量，并在此基础上伪造多台主机与有漏洞虚假目标增加网络复杂度，从而增加攻击者确定目标的难度。</p> <p>网络隐身系统满足用户需求，用户可以灵活设置虚假模式，随时通过更新的日志动态跟踪隐身动向。</p> <p>关键词：网络隐身 主机隐身 反嗅探 反扫描</p> <p>Cyber Stealth System is to allow a computer to normal communication at the same time, can be safely hidden in the network protection software. It with previous security software, are no longer passive hysteresis responses to threats, instead of using the new security model. To predict in advance security risks, active camouflage themselves and the attackers were confounded interference, so as to achieve the safety in network. Due to the realization of having foresight, prevent trouble before it happens, change from passive to active defense stealth, which can better serve our network travel escort.</p> <p>Cyber Stealth System utilizing predict in advance the mechanism of prevention of unknown security problem, in the Linux bottom of stack fingerprint confounders to interference scanning software cognition, using IP address jump strategy, allow an attacker to target the myriads of changes difficult to determine, the combination of Libpcap and Libnet forgery network traffic, and on the basis of forged many hosts and vulnerability of false targets increase network complexity, thereby increasing the attacker to determine the difficulty of the target.</p> <p>Cyber Stealth System to meet the needs of users, the user can flexibly set the spurious modes, at any time through the update log dynamic tracking stealthy movements.</p> <p>Keywords: Cyber stealing Stealthy host Anti-sniffing Anti-scanning</p>		

网络隐身系统

摘要

网络隐身系统是一款让计算机正常通信的同时安全的隐身于网络的防护软件，它与以往的安全防护软件相比，不再是被动和滞后地应对威胁，而是采用新的安全模型，能够真正的做到化被动防御为主动防御，并且提前预判未知安全问题，在此基础上主动对攻击者进行干扰混淆，增加其确定攻击目标的难度。该系统在内核层采用 LKM 方式使用 Netfilter 框架实现实时数据包监控和网络行为预判，并且修改包内信息即提供假的操作系统栈指纹和端口进程软件的版本信息。同时在用户层采用拒绝回应或回应虚假信息模式，迷惑或欺骗攻击者而不影响正常的网络通信，另外利用 libnet 和 libpcap 伪造虚拟多台主机及流量，增加网络复杂度，干扰攻击者目标的确定，最后利用空闲 IP 地址动态跳变使攻击目标千变万化，进一步增加了攻击者确定目标的难度，从而实现防范未知安全隐患，真正隐身与网络，最终为我们遨游于网络保驾护航。

关键词：网络隐身 主机隐身 反嗅探 反扫描

Cyber Stealth System

Abstract

Cyber stealth system is a security system which makes it possible that host can't be detected while legal communication is available. Compared with traditional security software, it is applied with a new security model to predict and prevent the security problem instead of dealing with threats passively. Using LKM(loadable kernel module) and netfilter frame technology to oversee data traffic and habits of links, as well as change parts of replying nothing or dummy information to jam scanners. Therefore fabricating dummy data traffic with Libpcap and Libnet and change IP address of local host enhances the difficulty of sniffing or scanning local area network, so that we can prevent our local hosts from undefined security problem. All of above steals the real host on the internet.

Keywords: Cyber stealing Stealthy host Anti-sniffing Anti-scanning

目录

1	引言	6
1.1	研究背景	6
1.2	系统特色	6
2	方案设计与选择	7
3	系统性能与指标	9
4	系统原理与实现	9
4.1	地址跳变模块	10
4.2	单机网络化模块	12
4.3	动态协议栈模块	19
4.4	网络行为预判	22
5	系统测试	23
5.1	测试环境	23
5.2	测试项目	23
5.3	地址跳变模块功能测试	24
5.4	单机网络化功能测试	25
5.5	动态协议栈功能测试	30
5.6	网络隐身跟踪模块测试	33
6	结束语	33
7	参考文献	34

1 引言

1.1 研究背景

自人类步入了信息化的时代以来，计算机技术与计算机网络以难以想象的速度发展着。在提供网络便利的同时，信息的安全传输就显得尤为重要，但与此同时，层出不穷的网络安全问题却时刻困扰着我们。针对目前已有的防护措施：防火墙，杀毒软件，流量监控，入侵检测等都存在着一些共同的或单一的缺陷，防火墙并不能灵活地进行主动防御，并且依赖于管理员大量添加的过滤规则；杀毒软件并不能查杀变幻多端的新病毒，且只能在病毒入侵之后才能进行补救工作；流量监控或入侵检测收集大量的数据，增加了工作人员分析数据的负担，同时也只能在异常行为发生以后才进行报警或阻断。

可以看出现存的安全防护最大弱点是被动防御即都是在异常行为甚至安全事故发生后才进行修复，并不能满足实时保障用户安全的需求，而是有着比较大的滞后性，只有从根本上解决问题才能达到真正的安全防护。只有变被动为主动，提早在入侵之前就设法将安全隐患扼杀于摇篮，这正是我们网络隐身系统的特色功能。

当网络隐身系统开启之后，我们在正常享受互联网为我们带来的便利的同时，还能够安全隐身于网络，让入侵者“看不见”，无法确定攻击目标，而合法的用户又能与我们正常地进行通信。而对于未知的安全漏洞由于攻击者无法探测无法利用也就难以被发现利用。而对于恶意病毒因为隐身而找不到传播的宿主，也失去了效用。极大地增加了攻击者渗透或病毒侵染的难度。这样既能做到先知先觉，提前预判恶意网络行为，又能防患于未然，防范未知安全漏洞，做到百毒不侵，提前杜绝恶意访问。因此网络隐身系统及其策略有着很大的发展前景。

1.2 系统特色

本系统从原理上不同于其他安全保护软件，为了让保护行为时刻体现出一个“隐”字，即隐藏本机真实信息，实者虚之虚者实之，本系统具有以下特色：

(1). 自我隐身，实时运行：通过内核 **Nefilter** 预判并过滤阻断恶意网络行为，对各种探测保持沉默状态。

(2). 网络行为预判，规则动态添加：在内核添加网络行为预判准则，动态判断接收到的数据包的网络行为倾向，动态产生过滤规则。

(3). 动态协议栈机制，网络流分支处理：在原本协议栈基础上添加动态响应机制即

动态协议栈，通过预判提供分支处理依据，使用正常协议栈进行通信或是使用隐身系统协议栈模型进行通信。

(4). 地址随机跳变：动态变化本机 IP 地址，使攻击目标变化多端，让攻击者捉摸不定。

(5). 单机网络化，网络拓扑伪装：利用空闲 IP 地址虚假伪造多台主机，增加网络复杂度，伪装网络拓扑，安全隐身于网络。

2 方案设计与选择

网络隐身系统是以计算机为中心，它采取了各种方法与策略，让个人主机或者服务器系统在进行信息交互的同时尽可能减少泄露真实主机的有价值信息。

网络隐身系统灵感来源于军用隐形飞机以及跳频电台。借鉴隐形飞机的隐身原理使我们的计算机安全隐身于网络而不被恶意者探测；模仿跳频电台难以被窃听的特性让本地主机的某些信息也变得“飘忽不定”。

首先对隐形战斗机进行分析，我们可以将其“隐身”技术归结为主动隐身和被动隐身：

主动隐身技术包括两种，其一是隐形战机主动发射干扰电波让探测雷达接收到虚假信息，其二是在被红外制导导弹锁定的情况下散布红外诱饵，让红外制导导弹难辨真伪。

被动隐身技术包括三种，其一是在飞机表面添加涂层，采用非金属材料或者雷达吸波材料，直接吸收探测脉冲而不是反射来自雷达的探测波，其二减弱战机本身的红外辐射，减小被探测到的概率，其三是改变战机自身的外形结构，在外形上避免使用大而垂直的表面，这样可以有效散射探测脉冲绕过雷达的接收系统，从而实现隐身，

以上两方面策略总结起来如图 1 所示：

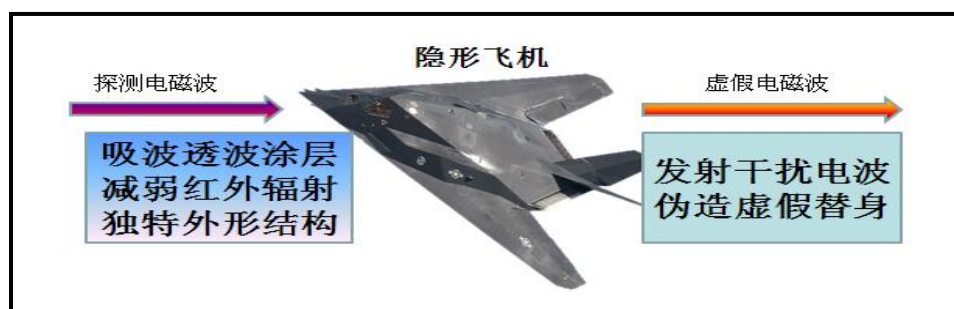


图 1 隐形战机隐形过程

网络隐身系统模拟战斗机的隐形原理，对探测的数据包也采取主动和被动两种处理策略。被动策略是指对出入本机的数据分组判断其合法性，对不合法的数据流量采取不接受、不回应、不交予正常协议栈、写入不良记录等处理方式；而主动策略分为两种方式，其一是在判断出探测行为正在进行的时候，回复虚假的信息迷惑探测者，增大其攻击难度，其二是伪造虚假数据流量与真实主机网络行为，避免嗅探。这样一来在攻击者眼中看来本地网络拓扑被大大地复杂化。

本机网段的虚假是指在真实主机所在的网段内充分利用该网段下的未使用的 IP 地址构造数据流量，模仿使用这些地址的主机可能出现的网络行为；其中虚假是指模拟某些服务器行为，为探测锁定目标增加难度，如图 2 所示。

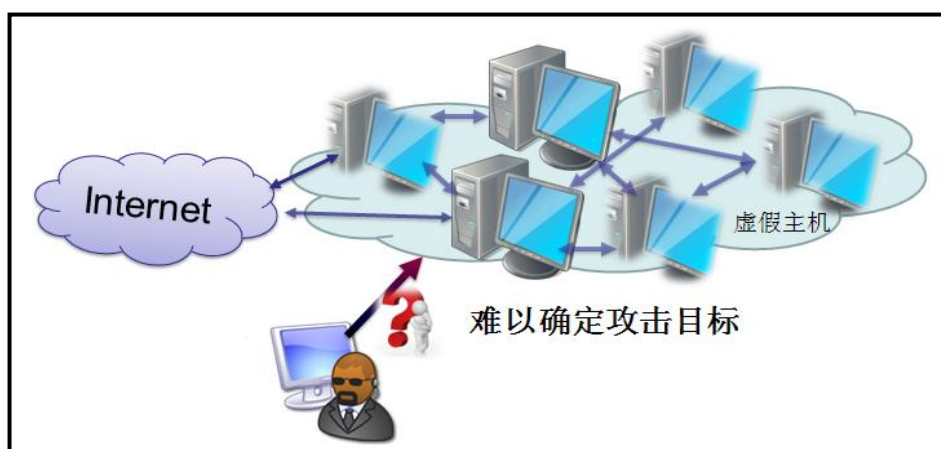


图 2 网络隐身过程

其次对跳频电台进行分析，跳频电台实现了载波频率随机或伪随机跳变，虽然通信双方能够根据事先约定好的频率跳变顺序更改调制与解调时用到的载波频率，但对这一约定一无所知的第三方监听者则很难进行监听。

如果把载波频率对应为计算机通信里面的 IP 地址的话，频率的跳变则能够联想到本机 IP 地址的跳变，同上面的虚拟网络流量一样，我们先在本网段内找到无人使用的 IP 地址，再从中随机挑选一个作为本机下次使用的 IP，经过一段时间之后，IP 地址更新，连接重置，攻击者可能掌握的原有 IP 的信息也都随即“消失”了。同时为了方便用户对本机安全与通信质量之间进行折中考虑，IP 地址跳变的周期是可以由用户自行设定的。

经过了上述过程，网络隐身能够很好地保证计算机安全，提前阻断未知安全问题，在源头就阻断恶意网络行为的发生，真正的做到提前防护，由于隐身的特点，该系统能够真正防患于未然。

3 系统性能与指标

基于上述设计思想，通过下面的应用场景对网络隐身系统做进一步描述：

功能 1：内核动态过滤。主机需要在数据包到达协议栈拆包之前做出判断，是恶意扫描或探测的数据包直接过滤或者提交由隐身协议栈处理，回复虚假信息迷惑探测者。

功能 2：内核数据指纹修改。内核对发送出去的数据的指纹特征进行修改，混淆恶意者的判断，避免恶意者从数据包的特征指纹获取有利用价值信息。

功能 3：动态地址池构建与维护。隐身的主机需要动态变化其 IP 地址和伪造多台虚假主机。首先需要获取局域网内未被使用的 IP 地址，其次需要动态更新 IP 地址状态，最后需要随机构建该网段的 IP 地址和 MAC 地址。

功能 4：真实主机 IP 地址动态跳变。为了影响恶意者的判断需要使真实主机的 IP 地址动态跳变，由用户层来操作跳变过程，通过延时来保证正常通信。

功能 5：伪造真实主机的网络通信。在伪造的计算机之间及外部网络之间主动构建数据流，避免嗅探，增加恶意者确定目标的难度。

功能 6：实时回复针对虚假主机的探测。在伪造主机后需要对探测的数据包予以回应，证明伪造的主机是活动的，并且需要回复扫描的探测数据包。

4 系统原理与实现

网络隐身系统按工作层次分为用户态和内核态两大部分。用户态按功能可分为网络虚拟化和隐身跟踪模块，在内核层为反侦查中心。其中主页是默认配置的隐身控制，在网络虚拟化可灵活设置虚假网络参数，利用 libnet、libpcap 和 honeyed 构建动态地址池，伪造多台虚假主机，其虚假主机的特性要求为：一能对主动探测做出虚假回复，其二能对嗅探查看做出混淆。而内核态的驱动程序为用户态的这些模块提供支持和保护，另外对数据包的拦截与修改、在混杂模式下对网络流量的嗅探也都应属于内核态。系统总体的框架图如图 3 所示。

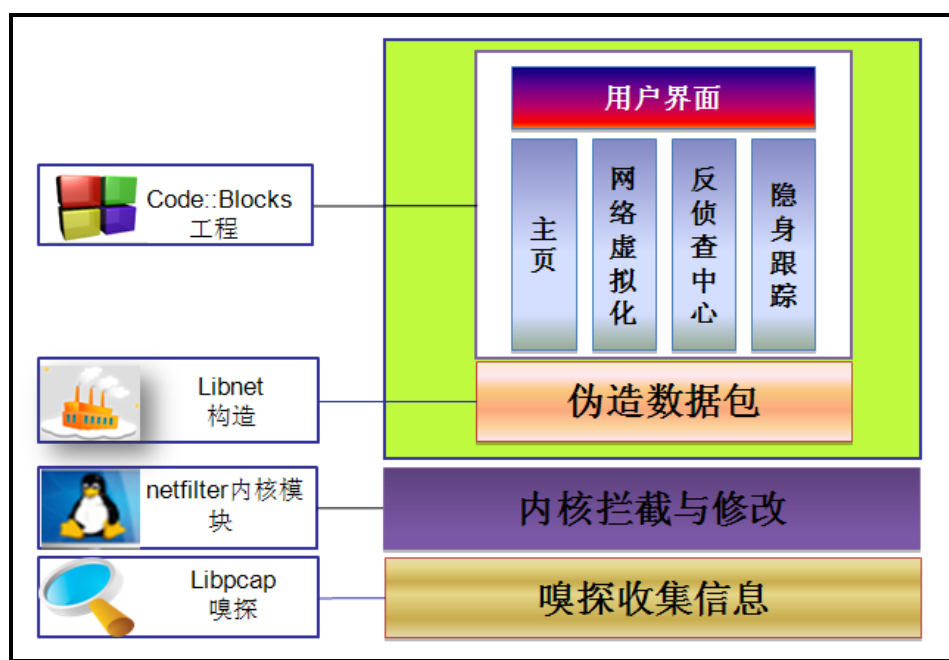


图 3 网络隐身系统框架图

以下是对各个模块具体实现过程的描述。

4.1 地址跳变模块

计算机之间的通信是以地址来区分彼此的，这里的地址不是单一的 IP 地址。其包括 MAC 地址，IP 地址和端口号。而在网络隐身系统中需要建立动态地址池为地址的跳变提供地址信息，并且实时跳变真实主机的 IP 地址。

4.1.1 物理地址的动态化

首先利用 ARP 询问探测出活动的 IP 地址，筛选出未被使用的 IP 地址添加到地址池。ARP 数据包结构如图 4

硬件类型 (16 位)		协议类型 (16 位)
硬件地址长度 (6 位)	协议地址长度 (6 位)	操作代码 (16 位)
发送方硬件地址 (以太网为 6 字节)		
发送方协议地址 (IP 地址为 4 字节)		
目标方硬件地址 (以太网为 6 字节)		
目标方协议地址 (IP 地址为 4 字节)		

图 4 ARP 结构图

把 ARP 的操作码置为 1，发送目的 MAC 地址为广播的 ARP 询问包，若询问的主机存活者回复操作码为 2 且携带 IP 地址对应的 MAC 地址。利用 libpcap 嗅探抓取 IP 地址为真实主机 IP 地址的 ARP 操作码为 2 数据包从而判断出该回复的主机是活动的，从而将 IP 状态标识置为“已被占用”。流程图如图 5

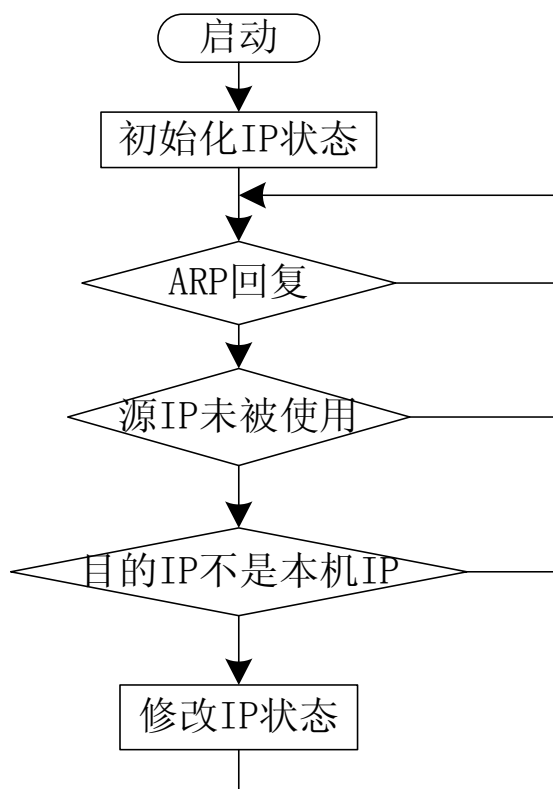


图 5 ARP 探测流程

该模块启动后首先初始化一个网段的 IP 地址，检查该网段中所有 IP 地址的使用状态，标识出哪些 IP 地址正在被使用，哪些 IP 地址是空闲的。这样做的目的是为了不影响那些正在使用的该地址的主机通信。

接下来创建一个线程，负责监听 ARP 请求。接受所有的 ARP 请求，通过 IP 地址管理平台检查该 ARP 请求的 IP 地址是不是伪装主机伪装的虚拟主机，如果是就发送一个 ARP 应答(应答包包含 IP 地址和对应虚假出来的硬件地址)，否则不回复 ARP。一般硬件地址前三个字节是生产商，后三个字节是生产商随时分配的，为了到达非常逼真的效果，我们前三字节是选择经常出现的生产商的标识，后三个字节是随机分配的。

最后还需要维护 IP 地址的状态，即创建一个线程监听回复 ARP 的 IP 地址，若目的 IP 地址不是本机地址并且源 IP 地址不是我们虚假的 IP 地址，那么可以判断出该 IP 地址已被占用。另外定时更新 IP 状态，即一段时间没有收到该 IP 地址的 ARP 回复这该 IP

地址的状态置为“空闲”。动态更新 IP 地址状态如图 6.

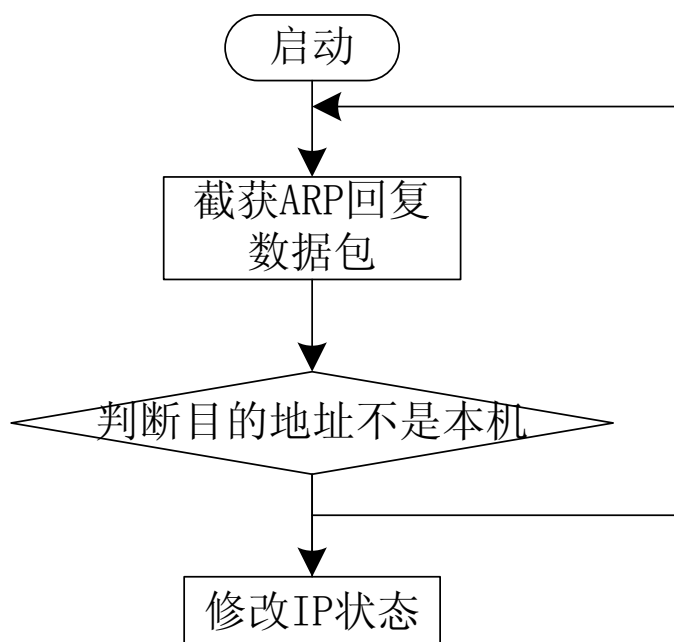


图 6 动态更新 IP 状态

4.1.2 IP 地址的动态化

IP 地址的动态变化主要包括两个方面：真实主机的 IP 地址的动态变化和虚拟主机的 IP 地址的动态变化。

真实主机的 IP 地址动态变化是指在建立地址池后随机选出一定数目的可用的 IP 地址作为真实主机通信时可能用到的 IP 地址，由于地址池在初始化时是顺序填入每个 IP 地址的因此在选取时只需产生一个 0-255 的随机数，作为 IP 地址的最后一位，提取出 IP 地址，在提取该 IP 地址的状态信息，并判断其是否可用，若可用则将其加入变化 IP 列表，若不可用则重新产生随机数并且进行选择，直到选出用户要求个数的 IP 地址为止。

虚假主机的 IP 地址动态变化是指在虚假主机之间和外部网络与真实主机之间的通信需要动态选出可用的 IP 地址作为通信的地址。在真实主机网段内 IP 选择方法与真实主机的 IP 地址选择相似。

4.2 单机网络化模块

战斗机在发现自己被红外制导锁定时，会放出红外引诱弹对红外制导系统进行干

扰，把自己隐藏在众多假目标之中，让敌方难辨真伪。单机网络化则模仿这一反制措施，由一台真实主机模拟出多台虚拟主机，让真实主机在攻击者看来置身于一个计算机集群中，即利用软件模拟真实情况下依赖硬件才能发生的网络行为，扰乱攻击者的认知。

攻击者判断目标主机是否活动的方法无外乎两种：其一是主动探测，即 Ping 查看是否连通和扫描工具扫描能够扫到该主机；其二是被动探测，即利用嗅探软件抓取数据包，从数据包的地址信息中可以查看出哪些 IP 地址是活动的哪些 IP 地址是没有网络流量的。

以下是单机网络化前后的对比，图 7 是低复杂度网络，即开启单机网络化隐身模块之前的情景。图 8 是集群化计算机网络，即单机网络化隐身策略的示意图。

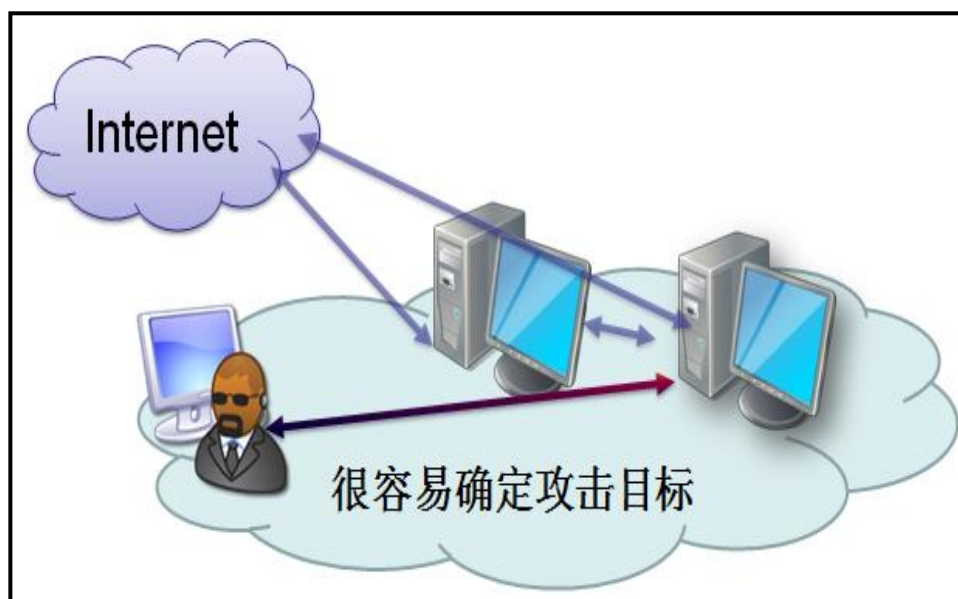


图 7 低复杂度网络

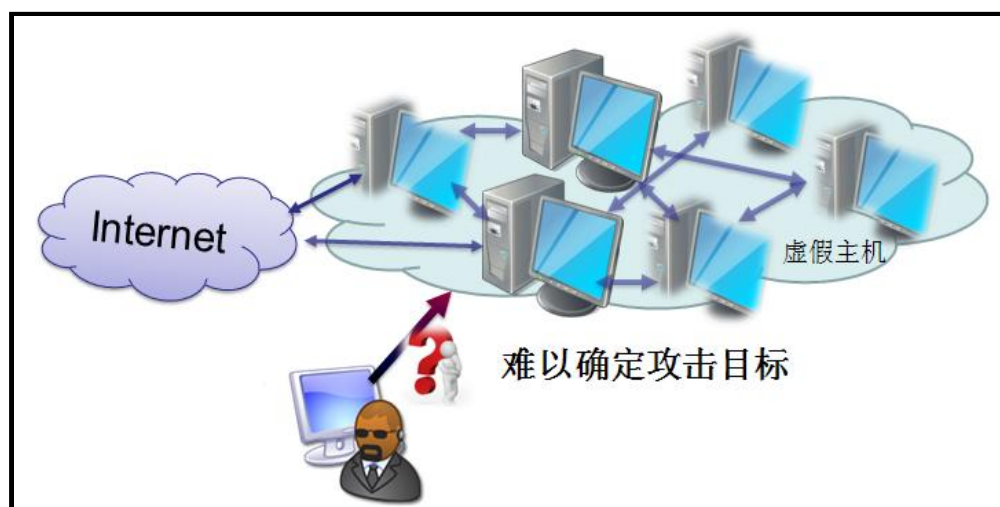


图 8 集群化计算机网络

4.2.1 分支双协议栈

在数据到达本机时系统底层驱动先判断其地址是否是该主机的 MAC 地址，若是则将数据交由内核处理，内核处理结束再交由协议栈一层一层拆解数据包的报头，最后交由应用层程序处理，如图是分支双协议栈数据流处理过程。

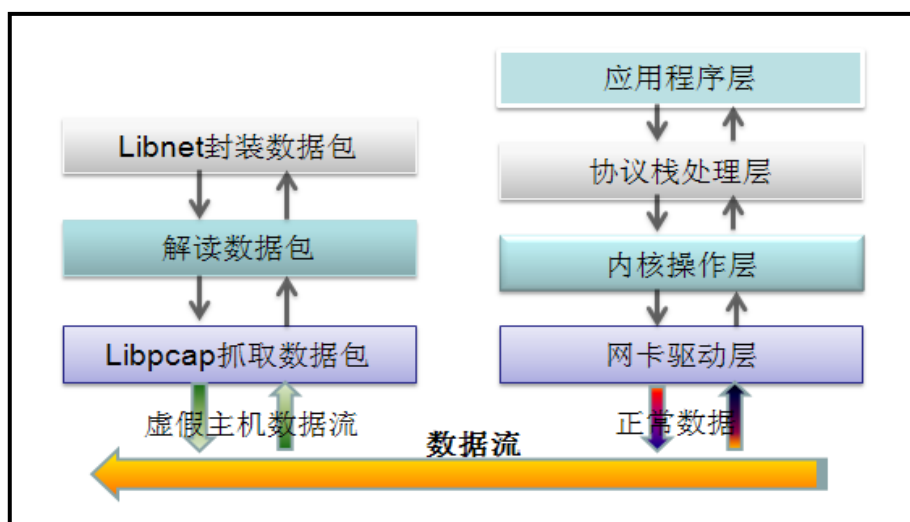


图 9 分支双协议栈数据流处理流程

为了能够实现虚假的主机的网络行为，因此也需要协议栈来处理虚假主机的数据包。本系统在数据进入协议栈之前就获取信息并作出响应，即在数据到达网卡驱动时就将其截获并交予模拟协议栈拆包获取数据包信息并对应回复其信息。

该项技术的实现主要依赖于来 libpcap 和 libnet。在 Linux 下 libpcap 和 libnet 都是开源的，其中 libpcap 主要用来嗅探，即在数据包到达网卡驱动层时复制一份数据交予用户层处理，而不影响正常的通信过程。

（一）libpcap 原理介绍

libpcap 主要用于网络统计软件，入侵检测系统，网络调试，数据包过滤，支持 BPF 过滤机制。

基于 libpcap 的嗅探器程序的总体架构，其流程如下：

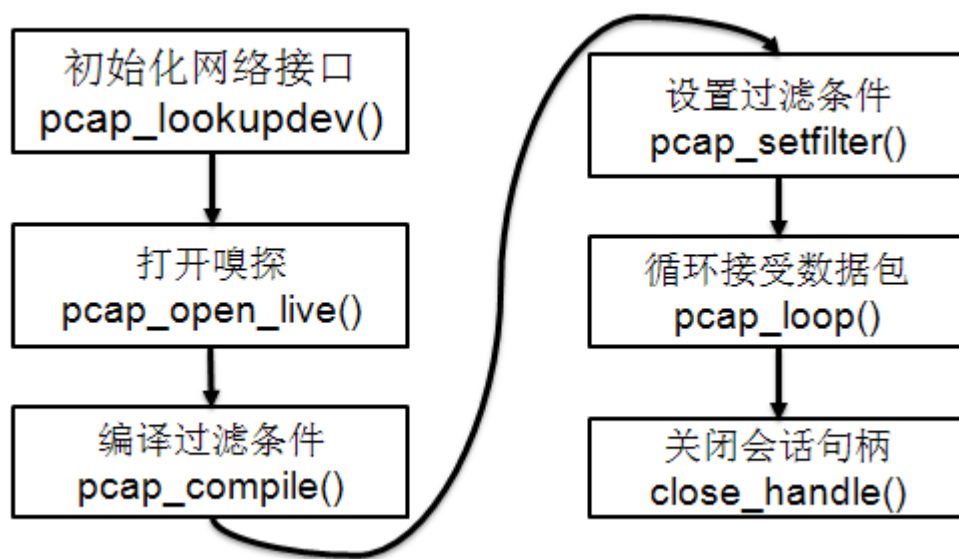


图 10 libpcap 开发流程

(1) 初始化网络接口。在 Linux 中，这可能是 eth0，而在 BSD 系统中则可能是 xl1 等等。我们也可以用一串字符串来定义这个设备，或者采用 pcap 提供的接口名来工作。

(2) 初始化 pcap。在这需要通知 pcap 对指定设备进行嗅探。假如需要，还可以嗅探多个设备。

(3) 设置过滤条件。我们必须创建一个规则集合，编译并且使用它。这个过程分为三个相互紧密关联的阶段。规则集合被置于一个字符串内，并且被转换成能被 pcap 读的格式(因此编译它)。编译实际上就是在我们的程序里调用一个不被外部程序使用的函数。接下来我们要告诉 pcap 使用它来过滤出我们想要的那一个会话。

(4) libpcap 进入它的主体执行循环。在这个阶段内 pcap 一直工作到它接收了所有我们想要的包为止。每当它收到一个包就调用另一个已经定义好的函数，这个函数可以做我们想要的任何工作，它可以剖析所部获得的包并给用户打印出结果，它可以将结果保存为一个文件，或者什么也不作。

(5) 在嗅探到所需的数据后，关闭会话并结束。

(二) libnet 原理介绍

Libnet 主要是用于构建各种各样的数据包，提供了在 IP 层和链路层构造数据包的功能，广泛应用于：ettercap, firewalk, snort 且支持很多种操作系统。可以高效的封装各种各样的数据包并发送到网络上。libnet 提供的接口函数按其作用可分为四类：

- 1) 内存管理(分配和释放)函数
- 2) 地址解析函数

- 3) 数据包构造函数
- 4) 数据包发送函数

libnet 作用：

- 1) 构造任意的数据内容
- 2) 构造各种不同协议的数据包
- 3) 支持从链路层到 IP 层数据包的构造
- 4) 支持跨平台、自动计算检验和
- 5) 发送数据包

下图为 libnet 开发流程：

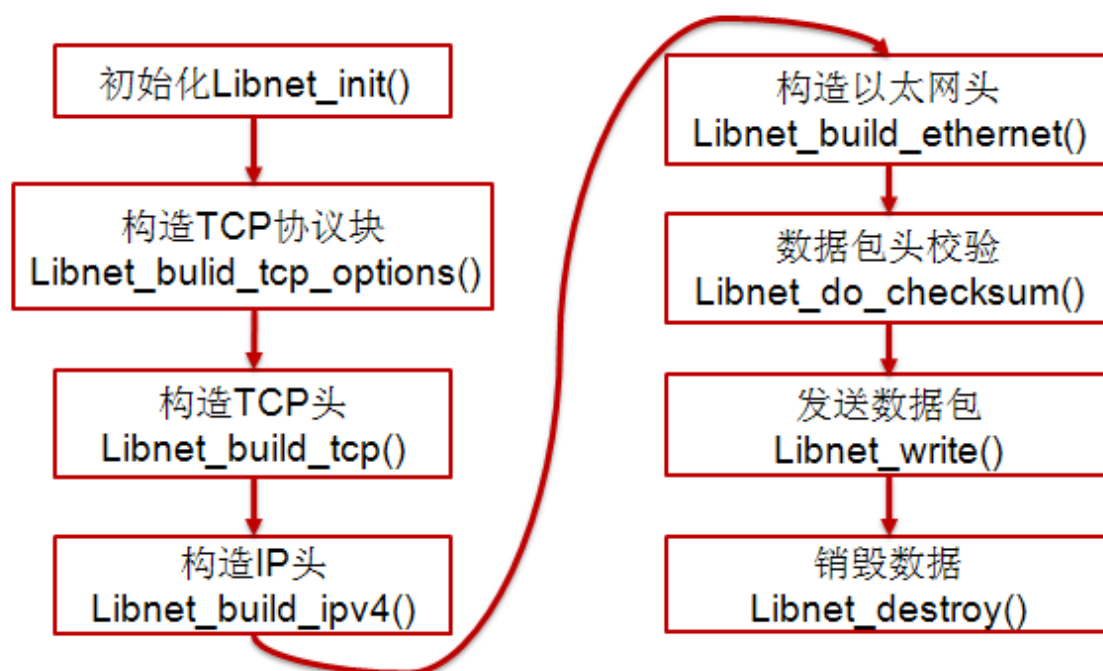


图 11 libnet 开发流程

4.2.2 伪造通信地址

伪造的通信地址按伪造对象来分包括虚假 IP 地址、MAC 地址和端口号，虚假的 IP 地址均为本网段无人使用的 IP 地址。

首先从地址池中获取相关地址信息，其次从中随机筛选出一定数目的可用的地址族作为虚假主机的地址，接着通过收集真实主机的网络行为的端口信息和协议类型信息，从而模拟真实主机的网络行为，在地址选好后需要利用 libnet 来构造各种各样的网络数据包，并发送到网络上。

网络通信的双方主要包括虚假主机与虚假主机之间的通信，虚假主机与外部网络的

通信，虚假主机与真实主机之间的通信。

针对地址的动态变化策略：

- 1) 通信源 MAC 地址:构建虚假 IP 时伪造的对应前三位真实生产商的 MAC 虚假物理地址。
- 2) 通信目的 MAC 地址: 虚假主机 IP 时伪造的对应虚假目的 IP 地址的 MAC 地址。
- 3) 通信源 IP 地址:虚假主机的地址，即真实主机网段未被使用的 IP 地址。
- 4) 通信目的 IP 地址:本机网段未被使用的 IP 地址，外部网络常见站点的 IP 地址，真实主机的 IP 地址。
- 5) 源端口地址: 源端口地址可以随机生成(端口>1024 的可被使用)。
- 6) 目的端口地址: 根据虚假数据包类型的协议信息来确定端口号。

4.2.3 IP 微路由技术

该技术实现 ARP 地址解析功能，但在 ARP 请求伪装 IP 地址时，微路由在伪装地址池中找到与伪装 IP 地址对应的主机硬件地址，给予回应，当攻击主机扫描不存在虚假主机 P1 时，微路由在伪装池中查找，把 P1 的硬件地址(虚假 P2 的硬件地址)回应攻击主机，攻击主机就可以利用 P2 的硬件地址与 P2 通信了，而 P1 起到了路由的作用，过程如图。

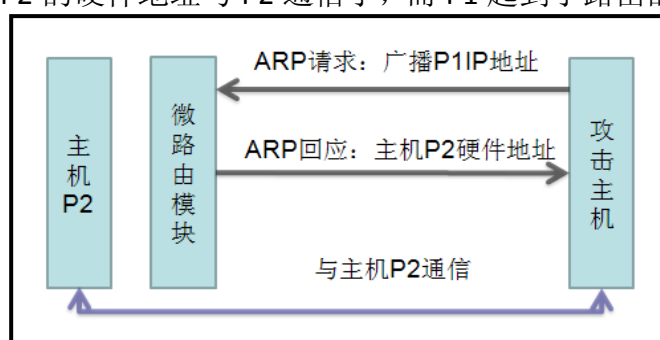


图 12 微路由实现过程

4.2.4 伪造主动数据流量

伪造数据流量的重点是构造各种类型的数据包，并且需要到达与真实主机的网络行为大体相似。因此获取真实主机的数据包类型尤为重要，我们提前嗅探收集相关主机的网络，获取特征数据样例，当伪造网络数据时利用这些样例数据作为种子，利用 libnet 构造数据包并发送出去。数据包类型信息包含带有 TCP 协议负责连接控制的数据包，带载荷的 UDP 数据包、带有应用层载荷的 TCP 数据包等等。

- 1) TCP 连接控制包主要模拟 TCP 三次握手然后携带数据包发送的过程如图，图 14TCP 连接过程。

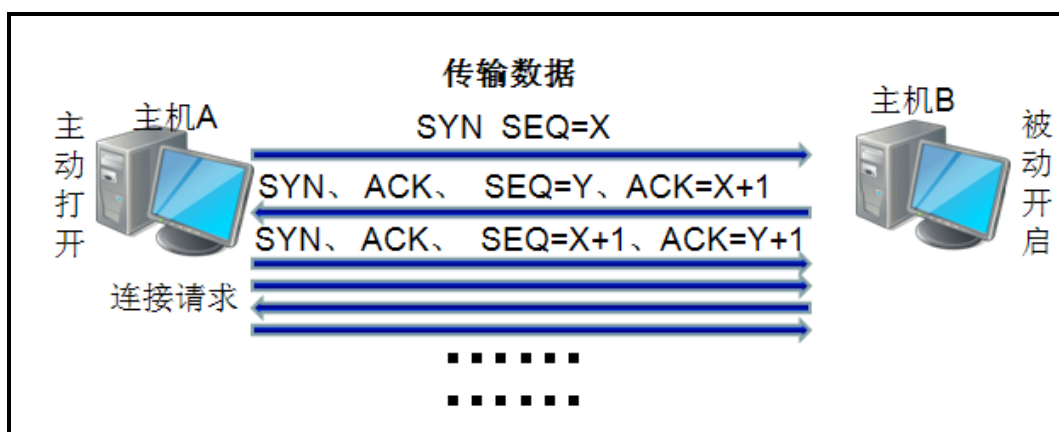


图 13 TCP 连接过程

2) UDP 带载荷数据包主要模拟 QQ 等使用 UDP 协议的应用层软件生成的数据包。

利用 libnet 构造 UDP 数据包，一部分数据包携带的数据信息是 QQ 软件所产生的数据信息，另外一部分是数据段是利用随机序列填充的数据包，让其看起来像是经过加密的数据流量而更凸显伪装的真实性和真实性。

3) 应用层主要模拟 HTTP 的网络过程。

由在真实局域网内进行抽查数据流量并进行统计得到的数据得知，一般个人主机使用频率最高的应用层行为为网页浏览，对应着 HTTP 协议，因此伪造 HTTP 流量很有代表性。由于前三层数据头已经封装好，在数据段加载 HTTP 相关的数据信息。接着组合这几部分为一体，封装成为网页浏览数据包。

4.2.5 伪造被动数据流量

伪造被动数据流量是指为了增加虚拟主机真实性，面向针对虚拟主机的扫描作出响应的行为。为达到此目的，为保证真实主机通信的原网络协议栈以及为支持虚拟主机而架设的虚拟协议栈，用真实协议栈响应真实数据流，用虚拟协议栈响应虚拟主机接收到的探测数据包。由于扫描服从少数判断原则，即只需要查到一部分数据包回复即可判断该主机可达，因此可以建立多个线程来响应某些探测包。探测回复的数据包主要是 ARP 协议回复、ICMP 回复，SYN 扫描的回复以及 ACK 扫描的回复。

ARP 回复即创建一个线程实时监听是否有主机询问虚假主机 IP 对应的 MAC 地址，若探测到 ARP 询问则回复该 IP 对应的 MAC 地址。此功能主要利用 libpcap 和 libnet 实现。

ICMP 回显即实时监听是否有 ICMP 的探测包，若有则回复标识为 reply 的数据包。ICMP 头信息结构如下：

Type(8 位)	Code(8 位)	检验和(16 位)
ID(16 位)		Seq(16 位)
数据部分		

图 14 ICMP 报头信息

SYN 回复是指当监听到有 SYN 请求时回复对方 ACK 或者 RST 数据包，若是伪造开放端口，则回复 ACK，这可看做是 TCP 协议建立连接三次握手的第二步，告诉扫描者虚拟主机正在等待连接的建立；若是伪造关闭端口，则会回复一个 RST 数据包。若是虚假主机之间的通信则可利用这一机制继续连接操作和后续通信。而判断是扫描还是虚假主机之间的连接的方法是查找 SYN 的源 IP 地址，若 IP 地址不是正在用来虚假通信的，则可以判断出该源 IP 地址的主机正在进行扫描，可以通知真实主机屏蔽该 IP 地址的请求。而虚假的主机按照正常协议流程回复 ACK 数据包，流程如下。

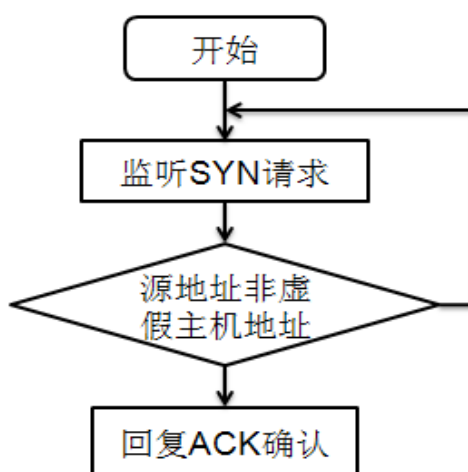
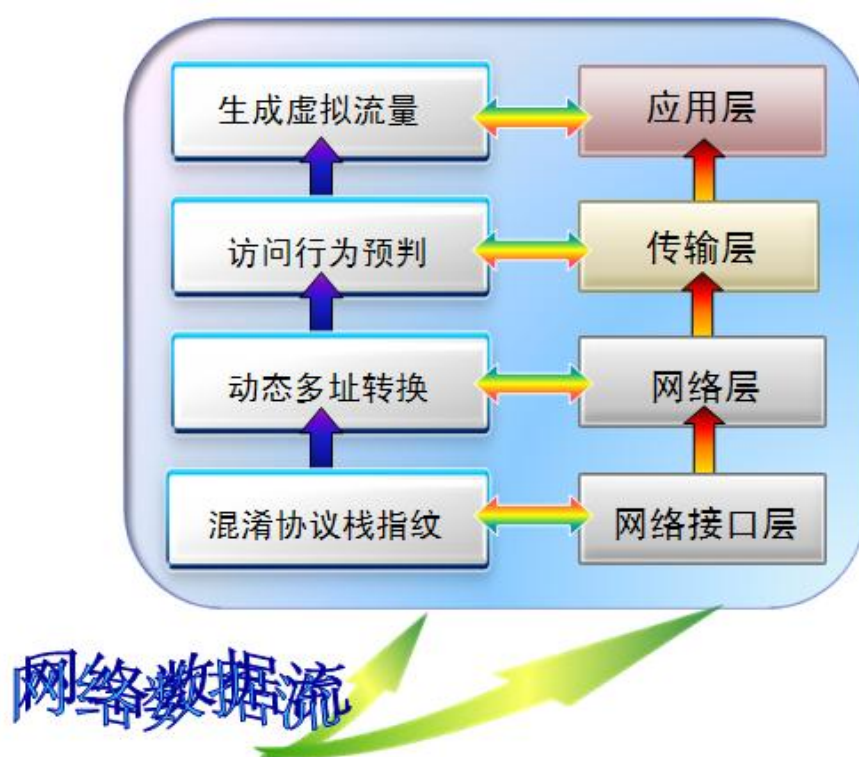


图 15 SYN 扫描回复

4.3 动态协议栈模块

动态协议栈包括两个关键点，其一是对于主机接收到的数据进行分支利用双协议栈模型处理。其二动态修改协议栈指纹特征混淆恶意者的判断。动态协议栈模型既能高效的处理属于真实主机的数据包与虚假主机的数据包，对其采取分支处理，从而不影响真实主机的正常通信。由于黑客可能利用协议栈特征来判断目标的各种信息，因此修改协议栈特征并使其动态变化尤为重要。

其中我们模拟 TCP/IP 四层模型结构，在不影响原有通信的基础上构造出了我们自己的网络隐身模型结构，其结构如下图所示。



该模型结构负责两方面的处理，其一负责处理真实主机的数据流，因此不影响正常通信，其构造出了伪装虚假协议栈，该协议栈也包括四层，其中混淆协议栈指纹层主要是截获出入真实主机的数据包，修改协议栈指纹特征；动态多地址转换主要完成单机网络化功能，扩展地址域，增加网络复杂度；访问行为预判主要是在内核层通过数据包的端口特征做出行为倾向预判，并动态添加预判规则，实时监控阻断攻击行为；生成虚假流量主要为了加深隐身效果，伪造真实主机的流量混淆嗅探。

通过该模型结构能够很好的使我们的主机安全的隐身于网络。

4.3.1 动态协议栈指纹

动态协议栈指纹是指动态变更因不同操作系统或软件版本带来的差别特征，从而影响入侵者的判断。协议栈指纹探测主要分为操作系统协议栈特征指纹探测和应用程序特征指纹探测。

（一）操作系统类型混淆

基于网络协议栈的探测又称基于操作系统指纹的探测。根据探测方式的不同分为两大类，一类是主动探测，即通过发送特定或特殊构造的数据报给攻击目标，进而分析攻击目标返回的数据报网络特征，得到攻击目标操作系统的信息。比较典型的主动探测工具有 Nmap, XProbev, Ring 等。另一类是被动探测，即被动嗅探，通过“嗅探”并分析攻击目标进行正常网络会话数据报的网络特征值，进而得出攻击目标的操作系统信息。

下面简单介绍几种操作系统类型探测工具的实现技术。

(1) Nmap 是一个功能全面的扫描软件，能对攻击目标的操作系统类型进行主动探测。其中 Nmap 探测时所利用的网络特征值主要包括：TCP 初始化序列 ISN 模式、不分段标识 DF、ACK 确认序列号模式、接收窗口的大小、TCP 报头标识中 flags 内容及选

项内容、不可达 UDP 数据报响应数据报内容。Nmap 构造并发送七种 TCP 探测数据包（称为 Tx）和一种 UDP 探测数据包（称为 PU）。通过对每种探测数据分组的回应进行综合分析并已有的协议栈特征值对比进而判断目标主机的操作系统类型。

(2) Xprobev 通过发送 UDP 数据报到攻击目标的关闭端口来促发 ICMP 端口不可达消息，并对 ICMP 消息中网络特征值基于模糊算法和逻辑树进行分析判断。其网络特征值主要包括：IP 总长度，IPID，IP 头检验和准确与否，UDP 包头校验和正确与否，优先权子段位（TOS 的值），DF 位响应，IP TTL 的值，ICMP 错误信息应用（Quoting）大小，ICMP 错误信息应答完整性，ICMP 时间戳请求，ICMP 信息请求，ICMP 地址掩码请求。

(3) RING 使用了一个建立在常规、无危险 TCP 传输上的新的操作系统探测技术。它模拟网络拥塞，不给目标主机及时发送响应数据报，通过分析目标主机在各次数据报之间的延时来判断目标主机的操作系统类型。

(二) 操作系统类型混淆

为得知扫描器如何根据收发特定数据包来判断数据包，小组使用 nmap 扫描器进行了针对 Windows 和 Linux 系统扫描实验，仔细观察对比针对两者的扫描返回结果，加上对 nmap 扫描器源码与协议栈特征库的解读确定了一些对操作系统类型的探测方法，由此得出针对这些方法的应对措施：

(1)针对 SYN 扫描探测，一般 windows 系统对非明确规定开启或关闭的端口采取不予回应的策略，而 Linux 系统则是“有求必应”，把所有端口默认为关闭状态，即对 SYN 包均回复 RST，为混淆扫描器判断以及尽量少地泄露信息，我们可以在 linux 内核层对非明确指定开启或者关闭的端口上所回复的 RST 包进行过滤。

(2)Windows 系统常常对 3389 远程桌面等一些高风险端口默认设置为关闭状态，即回复 RST 包，故在 Linux 系统上也将计就计，对这些端口的 SYN 探测回复的 RST 不予拦截，但对其中的 IP 包头 ID 字段与 Flag 字段进行修改伪造，再重新进行校验和计算，混淆操作系统信息。

(3)扫描中还发现一般的 Windows 系统经常有 139、445、912 等数个默认开放的端口，为混淆操作系统判断，我们把针对这些端口的 SYN 扫描所回复的 RST 分组强制进行 TCP Flag 改编，使其变成表示着端口开放的 SYN ACK 包，同时对 IP 头的 ID 字段、TCP 头的 sequence number 字段、window 字段进行模仿 Windows 系统协议栈行为的修改，再重新计算 IP 头与 TCP 头校验和，发送伪造数据包。

(4)针对 ICMP 数据包的探测，我们将回复包的 IP 头的 flags 字段修改为 Windows

系统网络协议栈常用的 0x02 这一数值，ICMP 头的回复数据包中，code 字段也设置为了 Windows 系统协议栈用的 0x00 这一数值。

(5)针对 UDP 数据包探测，Windows 系统的常见做法是对所有的 UDP 数据包采取静默处理，不回复 ICMP Destination Unreachable 信息，但是 Linux 系统网络协议栈则常常回复这一信息，故侦测到 UDP 数据包发送到非开放的 UDP 端口，并触发了协议栈回复 ICMP Destination Unreachable 信息，则把此 ICMP 数据包过滤掉。

(6)针对 FIN+PUSH+URG 探测，对一个未建立连接的端口发送 TCP Flag 位设为 FIN+PUSH+URG 的数据包，Windows 系统的常见做法是不予进行回复，而 Linux 系统的网络协议栈则常常回复一个包含了许多系统协议栈信息的 RST 包，故当侦测到发向未建立连接的 TCP 端口的 FIN+PUSH+URG 包我们可以将其直接过滤。

(8) TTL 探测：即数据包的“存活时间”，表示一个数据包在丢弃之前可以通过的多少个跳跃节点，不同操作系统类型的缺省 TTL 是有差异的。如 Windows 9x/NT/2000 是 128，Linux 2.2.6xIntel 是 64，Netware 4.11Intel 是 128，AIX 4.3.xIBM 是 60，Solaris 8 intel 是 64，可以把所有回复数据包的 TTL 字段全部设为 windows 协议栈常用的 128 这一数字，可对扫描器起到一定的迷惑作用。

本系统基于 Linux 2.6 内核中的 Netfilter 框架对数据包进行实时修改实现协议栈数据特征混淆，让扫描器难以判断操作系统网络协议栈指纹信息，也让看到扫描结果的攻击者难以根据扫描收集到的信息判断目标主机的操作系统究竟为何。

4.4 网络行为预判

网络行为预判主要是在内核操作，它将正常的网络行为与异常的网络行为分别建立起一个状态链，对网络行为是否合法进行提前判断。

网络行为判断需要区分服务器和个人计算机两个模块，在个人计算机上，因为其不为外界提供服务，故不会接受从外界发来的主动 SYN 请求，这样看来凡是接收到 SYN 的源 IP 地址都是可疑的，除此之外，如果本机和某个远程主机未曾建立连接，对方却发来 RST 包或 ACK+PUSH+URG 包等情况也很有可能是扫描行为的先兆；而服务器由于需要向外提供服务，因此端口与地址不但不能更改并且还必须告知客户，故需要对受到的数据包的状态链进行判断。以下从个人系统和服务器角度具体阐述网络行为预判准则：

在个人系统中出现可疑行为即可初次判断该主机是可疑的

- 1) 出现端口误撞症状(请求的数据包请求了主机未开放的端口)则扔掉数据包，并将该源地址信息添加可以地址列表。

- 2) 个人系统不会接收 SYN 数据包，凡是接收到 SYN 的地址列为可疑对象。

3) 添加延时记录几个数据包的源 IP 地址和端口信息，若出现同一 IP 地址出现不同端口的数据包者可以扔掉该数据包，在协议栈响应后回复的数据包直接不向外发送

4) 对于 ACK 数据包采取基于连接机制，记录前后建立连接的 IP 地址信息，若接收到 ACK 数据包，但在此之前并未建连接则该主机属于异常。

5) 对于出现 TCP 标识位异常即标识为 NULL 或 FIN 与除了 ACK 的其他标识的任何搭配的数据包，则该主机肯能在扫描，可将其加入黑名单。

6) 对于 UDP 的扫描数据包不回复 ICMP 的 port 状态数据包（由于 UDP 是看 ICMP 回复来判断端口状态的，若端口关闭则回复 port unreachable，若开放则没反应，也可混淆攻击者）。

在服务器中出现一下行为视为可疑对象

1) 出现端口误撞，即请求了服务器没有提供服务的端口视为可疑。

2) 出现高频率的请求行为，比如多个 SYN 请求视为可疑。

3) 短时间一个源地址请多个端口，视为可疑。

4) 接收到数据包 TCP 标识位异常出现 NULL 或 FIN 与除了 ACK 的搭配的标识信息视为可疑。

5) 接收到未建立连接的源地址的 ACK 数据包视源地址为可疑对象。

在众多规则中我们采用少数确定原则来判断，即只要在建立的规则中符合一条即可判断出该主机可能正在被扫描或探测，即可把该 IP 地址加入黑名单，禁止该地址的任何一切访问与请求。还可以根据满足规则的数目来动态为其设置权限。符合规则越少的权限越高，若符合 4 条以上则直接拉入黑名单，根据每个地址的权限来判断该 IP 地址的可疑来判断是否完全截断该 IP 地址对真实系统的通信。

5 系统测试

5.1 测试环境

两台计算机，其中一台安装有安装网络隐身系统，另一台先后充当相对应的合法服务器、合法用户、攻击者等角色，一台交换机，系统运行于 Linux2.6.32 操作系统。

5.2 测试项目

测试项目表如下：

表 1

系统模块	测试项目
地址跳变模块	1. 真实主机的 IP 地址跳变
单机网络化模块	2. Ping 探测虚假主机状态 3. 扫描虚拟化网络 4. 扫描单台虚假主机详细信息 5. 伪造虚假主机流量 6. 伪造 HTTP, FTP, telnet 服务信息
动态协议栈模块	8. 预判并阻断扫描行为 9. 操作系统类型混淆
网络隐身跟踪模块	10. 隐身日志和跳变日志查看

5.3 地址跳变模块功能测试

测试项目如下：

1. 真实主机的 IP 地址跳变

5.3.1 真实主机的 IP 地址跳变

测试项目：测试系统是否使真实主机的 IP 地址发生跳变。

测试指标：真实主机 IP 地址间隔跳变且还能继续保证通信。

测试过程：运行系统设置动态跳变参数，测试时设置跳变时间为 1 分钟每次，查看列表 IP 地址变化状况和真实计算机的 IP 地址情况。

测试结果及分析：



图 16 一次跳变截图



图 17 接下来一次跳变

结论：从上图可以看出本机真实 IP 地址发生动态跳变，本测试成功

5.4 单机网络化功能测试

测试项目如下：

1. Ping 探测虚假主机状态
2. 扫描虚假主机
3. 抓取虚假主机数据
4. 扫描单台虚假主机详细信息
5. 伪造 HTTP, FTP, telnet 服务

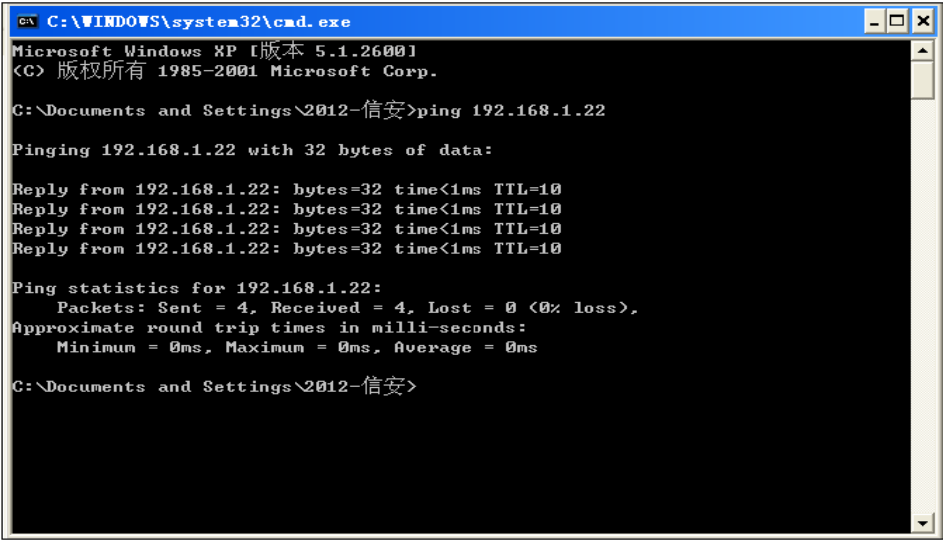
5.4.1 Ping 探测虚假主机状态

测试项目：测试虚假主机是否存活。

测试指标：测试时对于 Ping 有回复信息

测试过程：运行虚假主机虚假模块，在另外计算机上 Ping 伪装主机，查看状态

测试结果及分析：



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\2012-信安>ping 192.168.1.22

Pinging 192.168.1.22 with 32 bytes of data:

Reply from 192.168.1.22: bytes=32 time<1ms TTL=10
Reply from 192.168.1.22: bytes=32 time<1ms TTL=10
Reply from 192.168.1.22: bytes=32 time<1ms TTL=10
Reply from 192.168.1.22: bytes=32 time<1ms TTL=10

Ping statistics for 192.168.1.22:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\2012-信安>
```

图 18 ping 查看虚假主机是否存活

结论：开启虚假主机模块，从界面获取随机虚假的 IP 地址是 192.168.1.22，在另外一台计算机上 Ping192.168.1.22 后发现该主机存活，TTL 为 10 是测试的区分标识，此测试成功

5.4.2 扫描虚假主机

测试项目：测试虚假主机是否成功骗过扫描器，通过扫描来查看是否有回复来判断

测试指标：利用 Nmap 扫描到真实主机网段出现多台虚假计算机信息

测试过程：运行网络隐身系统，两台计算机，一台运行网络隐身系统，一台计算机运行 Nmap, 扫描网段内的主机，查看是否有虚假主机存在。

测试结果及分析：

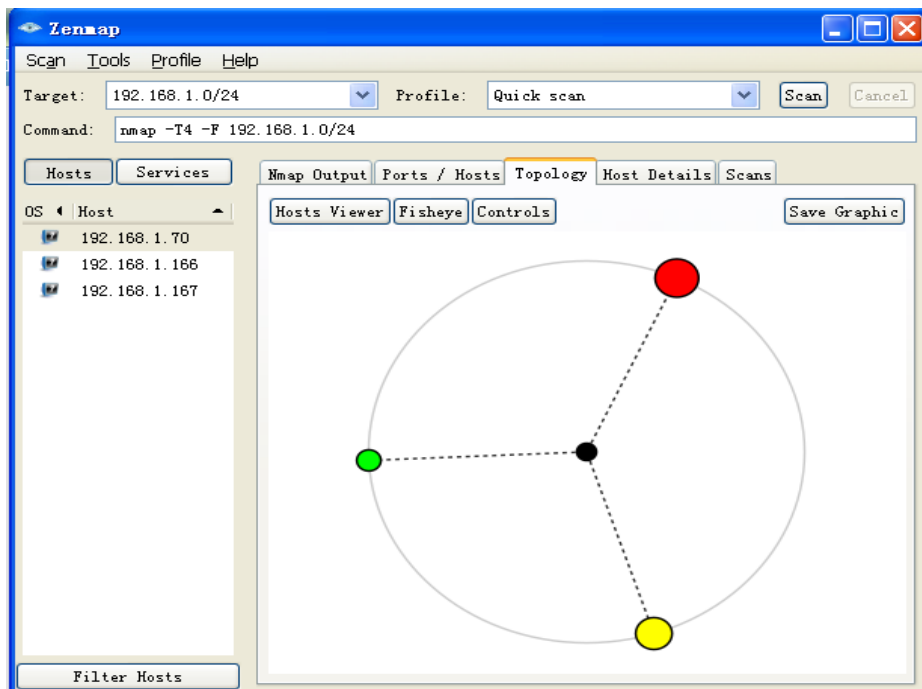


图 19 未开启虚拟网络扫描网段效果

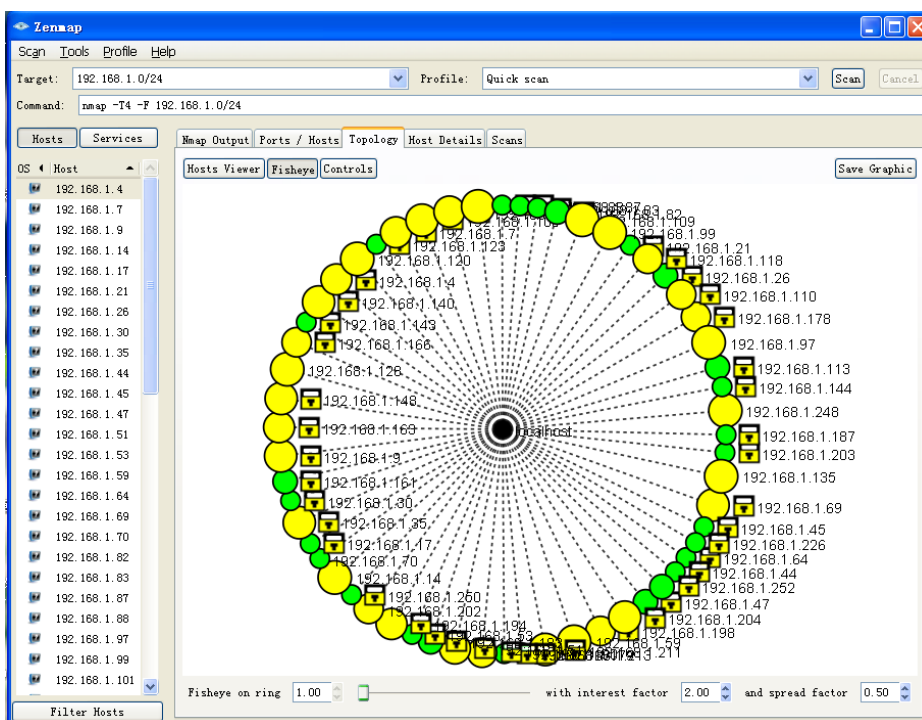


图 20 开启虚拟化网络扫描 192.168.1.0 网段主机

结论：扫描 192.168.1.0/24 网段可以看出我们虚假了多台主机，说明虚假主机模块的开启大大增加了网络复杂度，起到迷惑攻击者的目的，此测试成功。

5.4.3 抓取虚假主机数据包

测试项目：利用 wireshark 监听网络数据包查看数据信息

测试指标：数据信息显示有虚假主机的地址在通信

测试过程：运行网络隐身系统，另外一台计算机运行 **wireshark** 监听数据包并查看源地址和目的地址

测试结果及分析：

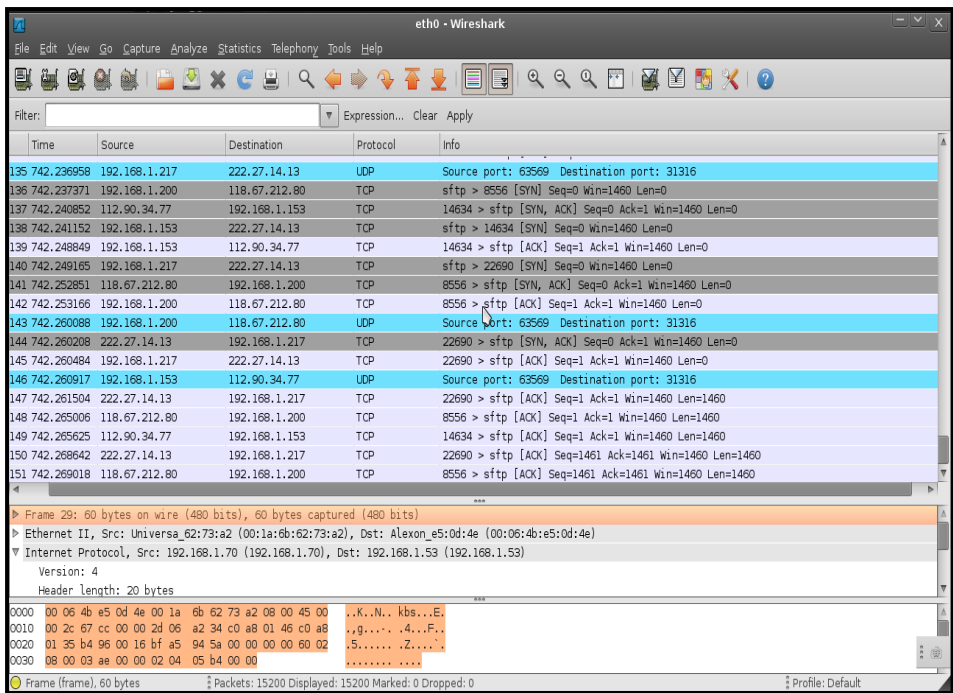


图 21 嗅探虚假主机所在网络的数据流量

结论：通过对 **wireshark** 监听的数据包进行分析可以看出，有多个 IP 地址正在通信，由于虚假主机模块伪造了各种类型的数据包，故相似度很接近真实主机的网络数据流，由于模拟了真实网络环境的网络行为，大大增加了入侵者嗅探查取目标信息的难度，由此图看出测试成功。

5.4.4 扫描单台虚假主机详细信息

测试项目：利用 nmap 扫描单台虚假主机

测试指标：可以扫描出特定虚假主机的端口信息和系统信息

测试过程：运行网络隐身系统，开启单机网络化模块，利用 **nmap** 扫描其中某台虚假主机，查看扫描结果信息

测试结果及分析：

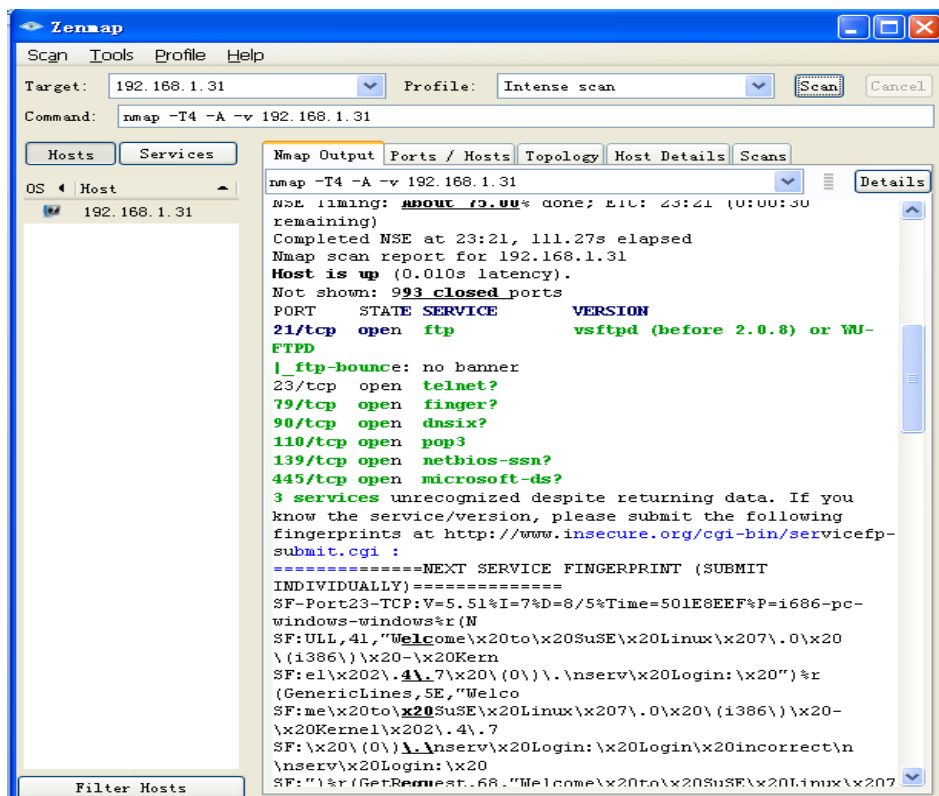


图 22 Nmap 扫描单台计算信息

结论：从扫描主机信息可以看出虚假主机的开放端口与系统信息，此测试成功

5.4.5 伪造模拟 HTTP,FTP 及 telnet 服务信息

测试项目：伪造 Web,FTP 及 telnet 等服务

测试指标：虚假主机出现 Web 服务, FTP 服务和 telnet 服务

测试过程：开启隐身模块, 选择虚假服务的 IP 地址, 在模拟攻击主机上利用 Web 浏览器查看 HTTP 服务和 FTP 服务, 并在 Dos 下连接 23 端口查看 telnet 服务

测试结果及分析:

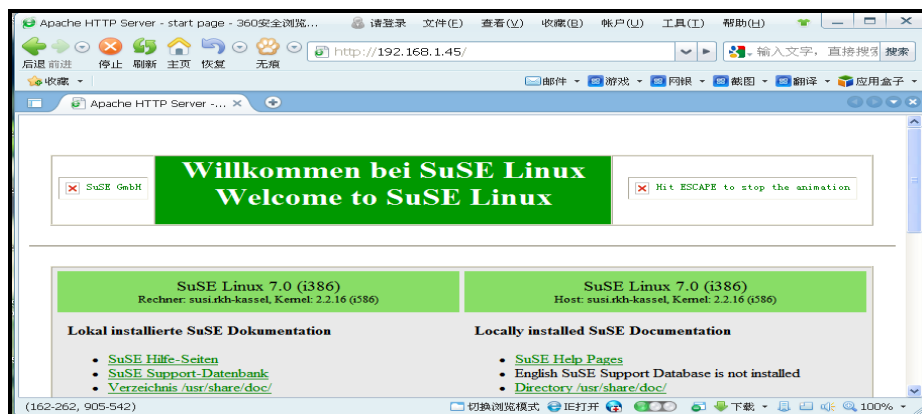


图 23 浏览虚假的 Web 服务

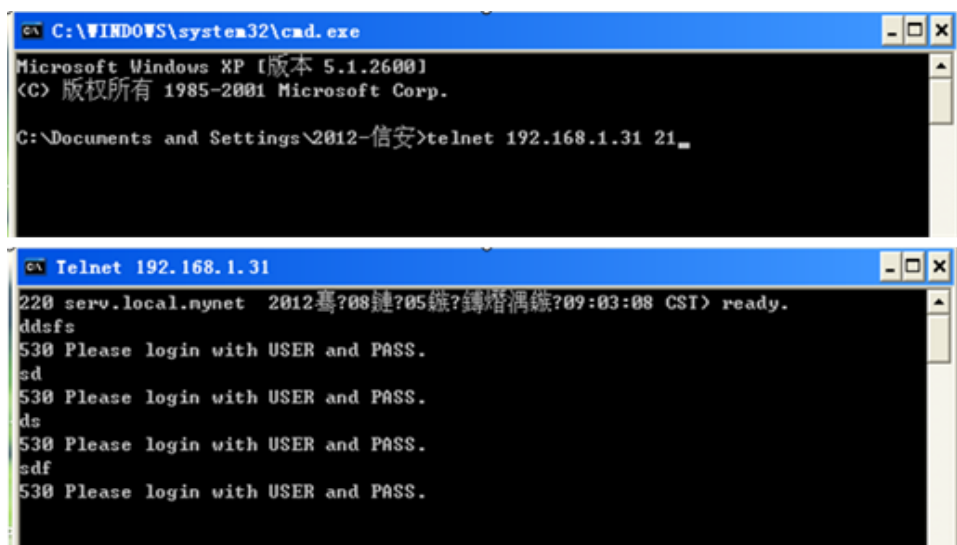


图 24 连接 FTP 服务

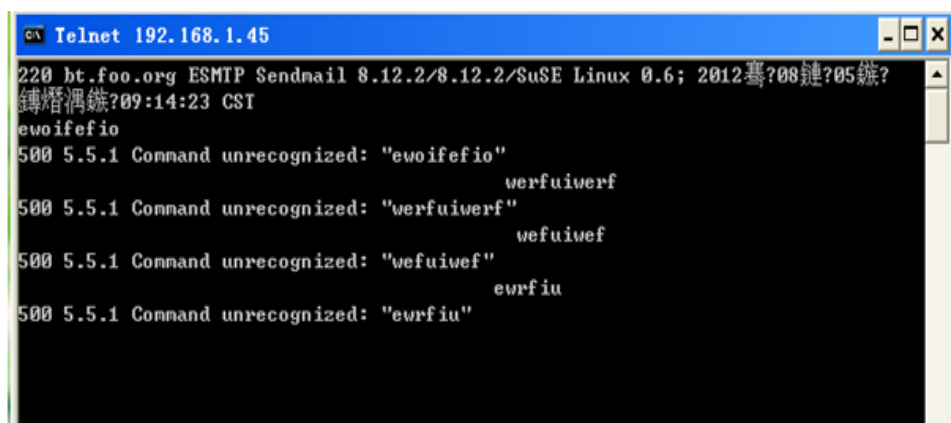


图 25 连接 telnet 服务

结论：从上述三张效果截图可以看出，成功的虚假伪造了 HTTP,FTP,telnet 三种协议类型的服务，虚假服务测试成功。

5.5 动态协议栈功能测试

测试项目如下：

- 1.操作系统类型混淆
2. 预判并阻断扫描行为

5.5.1 操作系统类型混淆

测试项目：测试操作系统栈指纹模块能否混淆真实主机的操作系统类型信息。

测试指标：把本地真实主机的系统混淆为其他类型的操作系统。

测试过程： 利用 Nmap 扫描探测操作系统（开启-O 选项），查看操作系统类型扫描结果。

测试结果及分析：

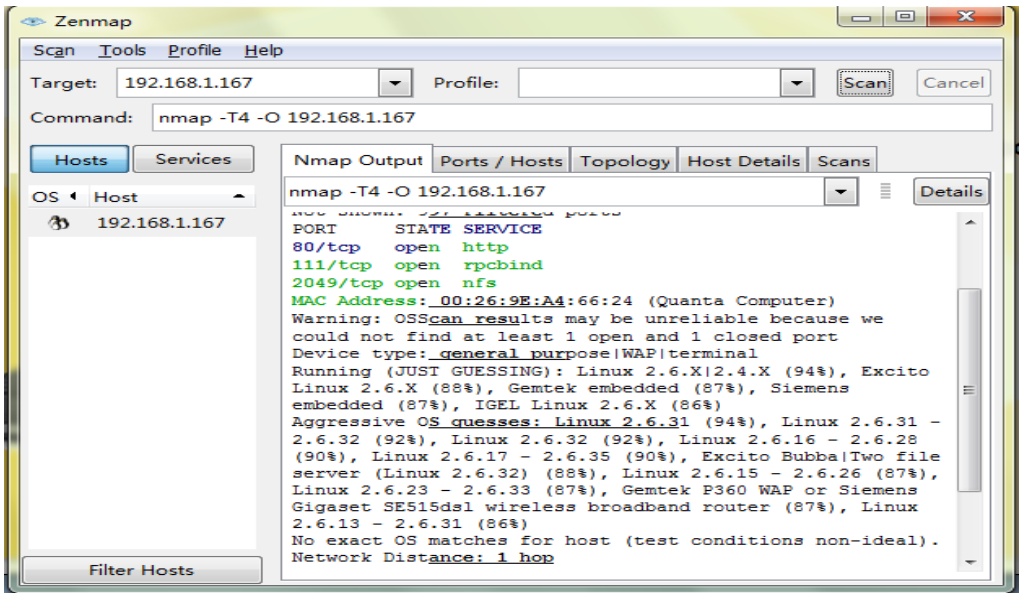


图 26 未开启操作系统混淆模块扫描信息

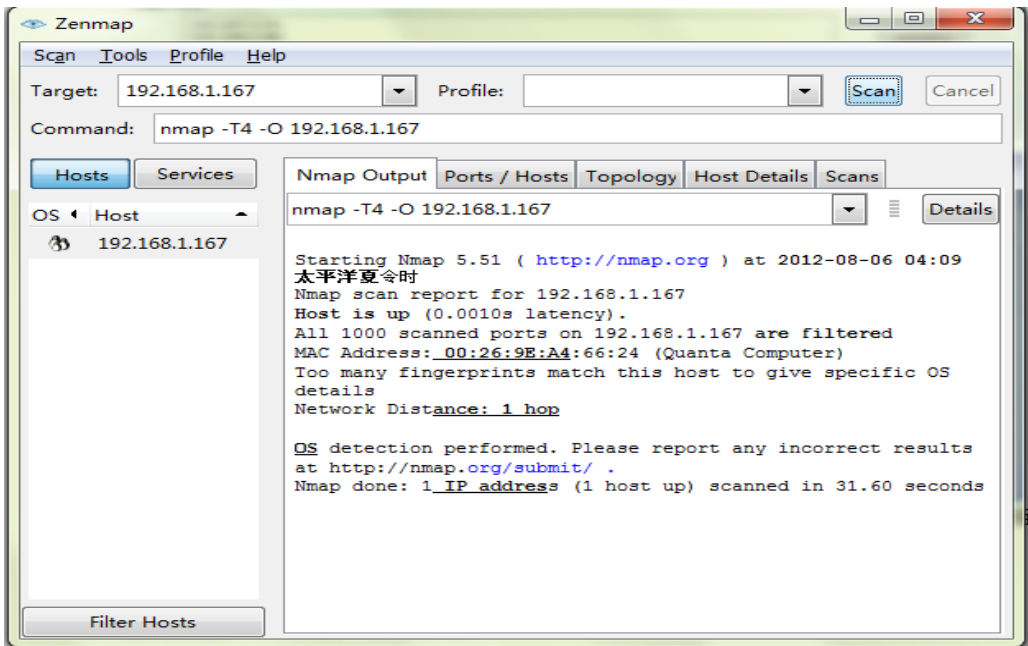


图 27 开启操作系统混淆模块扫描信息

结论：从扫描信息看出该系统伪装成虚假主机且隐藏了真实主机信息，而且判断不出此系统类型信息，此测试成功

5.5.2 预判并阻断扫描行为

测试项目：预判扫描行为并阻断扫描

测试指标：在界面得到扫描者的地址信息，并阻断扫描行为

测试过程：开启网络行为预判模块，利用另外一台计算机扫描被系统保护计算机，在保护主机查看预判信息，并查看扫描获取的信息。

测试结果及分析：



图 28 反侦查可疑地址列表

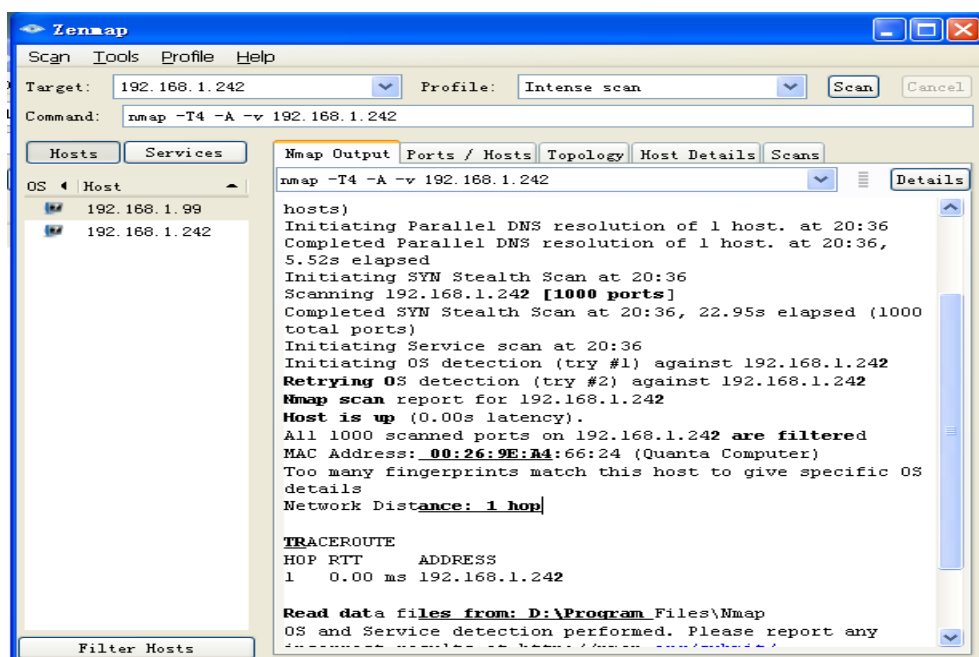


图 29 扫描开启预判模块的主机

结论：从上图中可以看出，在可以地址池中加入可疑地址后扫描的主机信息，成功阻断扫描行为

5.6 网络隐身跟踪模块测试

测试项目：网络隐身过程跟踪。

测试指标：查看界面获取日志，其中包括动态跳变的日志信息和虚假服务的日志信息。

测试过程：开启网络隐身模块，定时查看跳变的日志信息和虚假服务的日志信息。

测试结果及分析：



图 30 真实主机 IP 跳变日志和虚假服务日志

结论：从上图可以看出在隐身一段时间后出现隐身日志信息，其包括地址跳变信息和虚假 HTTP 服务的信息

6 结束语

当前防护软件种类繁多，有些软件防护过于被动，有些软件不易操作，有些软件只是专注于事后补救。因此提前防御、安全隐身的思想应运而生，而网络隐身系统能够很好地阻断入侵的第一步，防止入侵者获取相关信息而找到入侵渗透切入点，并且能够防

范未知安全问题，抵御变化多端的网络病毒，提前预判恶意网络行为，从根源消除安全隐患，伪装虚假迷惑攻击者。真正做到隐于无形，存于无形；防患于未然，隐身于网络。

综上所述，隐身的概念源于人类理论的创新，也将更好的服务于人，如果网络隐身系统与当前的防护软件相互协作，那么系统的安全性将会得到更大幅度的提升，这是本系统未来努力的方向。

7 参考文献

- [1] 李瑞民. SCAN 网络扫描技术揭秘[M]. 北京：电机械工业出版社，2011.
- [2] 谢进忠，高铁军.Linux kernel Module 及 TCP/IP 程序设计[M]. 北京：人民邮电出版社，2007.
- [3] Jasmin Blanchette &Mark Summerfiled C+ GUI Qt 4 编程（第二版）[M]. 北京：电子工业出版社，2008.
- [4] 宋敬彬 孙海滨著.Linux 网络编程[M]. 北京：清华大学出版社，20010.
- [5] Eric Code(英)，曹继军，林龙信，网络安全宝典（第二版）[M]. 北京：清华大学出版社，2010.
- [6] 谭献还.网络编程技术及应用[M]. 北京：清华大学出版社，2006, 23(9).
- [7] 刘文涛,网络安全开发包详解[M]. 北京：电子工业出版社，2010.
- [8] 杜华,Linux 编程技术详解 [M].北京：人民邮电出版社, 2007.