ANALYSIS OF ROUTE SPOOFING BASED ON OSPF ROUTING PROTOCOL

A Thesis Submitted to

Southeast University

For the Academic Degree of Master of Engineering

BY

XIA Yunfeng

Supervised by

A.Prof SONG Yubo

School of Information Science and Engineering

Southeast University

April 2014

摘要

路由协议是网络基础设施的核心，目前 OSPF 路由协议是一种使用广泛的内部网关路由协议。本文研究 OSPF 协议的安全性，在分析 OSPF 协议安全机制以及已有攻击方法的基础上，提出了四种新的攻击方法。这些攻击方法利用不同方式注入恶意的链路状态通告（LSA），修改路由器路由表，从而实现路由欺骗。本文同时设计实现了 OSPF 渗透测试系统，该系统可对 OSPF 网络进行安全性测试。全文主要工作如下：

1. 提出了一种邻接欺骗攻击。该攻击主要针对 OSPF 网络中边界路由器未设置为被动接口的场景，攻击者伪装成一台合法的路由器接入到 OSPF 网络中，注入恶意的 LSA。利用该攻击可实现网页欺骗、密码嗅探、中间人攻击、DNS 欺骗等效果。
2. 提出了一种双 LSA 远程多注入攻击。该攻击主要针对攻击者获得网络路由器拓扑及参数的场景，利用远程路由器的身份注入两个恶意的 LSA。与 Nakibly 等人提出的双 LSA 注入攻击相比，它可逃避自反击机制，同时增大了污染区域。这种攻击不仅可实现网页欺骗、密码嗅探等效果，还能控制流量的中间传输路径。
3. 提出了一种单路径注入攻击。在与攻击二相同的场景下，攻击者查找网络中满足单路径条件的路由器对，从中选出跳板路由器，并以它的身份注入一个恶意的 LSA。该攻击只需要注入一个 LSA 就能逃避自反击机制。利用该攻击可实现流量黑洞，从而造成部分区域的网络瘫痪。
4. 提出了一种远程邻接欺骗攻击。我们设计了一种探测远程路由器运行参数的方法，利用该方法可与远程路由器建立虚假的邻接关系，最后以幻影路由器的身份注入恶意的 LSA，可实现流量黑洞。该攻击与上述三种攻击相比其适用场景更广，可在未知网络路由器拓扑及参数的情况下实施。
5. 采用 GNS3 网络仿真软件、VMware 虚拟机以及真实物理计算机搭建了一个高仿真程度的网络模拟平台，并在此平台下验证了上述四种路由欺骗攻击的可行性及有效性。
6. 设计并实现了一个可对 OSPF 网络进行安全性检测的渗透测试系统，利用该系统可实现 OSPF 协议密钥认证机制的安全性评估以及 OSPF 网络渗透测试。经测试表明，上述四种攻击可在真实环境下达到预期的攻击效果，本文设计的系统可有效的发现 OSPF 网络的安全漏洞。

关键词：OSPF 安全；路由欺骗；链路状态通告；自反击机制

ABSTRACT

Routing protocol is the core of network infrastructure. OSPF routing protocol is a kind of interior gateway routing protocols, which is widely used at present. In this paper, the security of OSPF protocol is studied, and four new attack methods are proposed based on the analysis of OSPF protocol security mechanism and attack methods that exist. The method of these attacks injects malicious LSAs(link state advertisements)by different ways, modifies the routing tables of routers, finally realizes route spoofing. An OSPF penetration testing system is developed in this paper at the same time. The system can be used for security testing of OSPF networks. The work of this paper is as follows:

1. Adjacency spoofing attack is proposed. The attack is mainly aimed at the scenario that the border routers of OSPF networks are not set as passive interface. The attacker is access to OSPF networks by disguising as a legitimate router and injects malicious LSAs. Web spoofing、password sniffing、man-in-the-middle attack、DNS spoofing, etc are realized by the attack method.
2. Double LSA remote multi-injection attack is proposed.The attack is mainly aimed at the scenario that the attacker obtains router topology and parameters of networks, it injects two malicious LSAs by remote routers. Comparing with Double LSA injection attack proposed by Nakibly et al, it can evade “fight-back” mechanism, and increase contaminated area at the same time. The attack not only can realize web spoofing、password sniffing, etc, but also can control the middle of the transmission path of data traffic.
3. Single path injection attack is proposed. Under the scenario same as the second attack, the attacker finds a pair of routers that meet the condition of single path, chooses the springboard router, injects a malicious LSA by its identity. The attack just needs inject a LSA to evade “fight-back” mechanism. Traffic black-hole can be realized by the attack, causing parts of networks paralytic.
4. Remote adjacency spoofing attack is proposed. We design a method to detect running parameters of remote routers. False adjacency relation can be established by the method, finally malicious LSAs are injected by the identity of phantom routers to realize traffic black-hole. Comparing with above three attacks, the attack is used in more scenarios, it can be realized in the scenario that the attacker doesn’t obtain router topology and parameters of networks.

5. The feasbilites and effectivenesses of above four route spoofing attacks are verified on the network simulation platform constructed with GNS3 network simulation software、VMware virtual machine and a real physical computer.
6. An OSPF penetration testing system which can implement security testing of OSPF networks is designed and realized. It includes security assessment function of OSPF cryptographic authentication mechanism and penetration testing function on OSPF networks. Above four attacks can achieve desired effects in the real environment, and the system can effectively discover security vulnerabilities of OSPF networks.

Keywords: OSPF security; route spoofing; link state advertisement; fight-back mechanism

目录

摘要 ..... **II**
ABSTRACT ..... **IV**
目录 ..... **VI**
插图目录 ..... **X**
表格目录 ..... **XII**
第 1 章 绪论 **1**
1.1 研究背景 1
1.2 路由欺骗技术 2
1.3 课题研究现状 3
1.3.1 资源消耗攻击 3
1.3.2 路由欺骗攻击 3
1.4 论文具体工作 4
第 2 章 OSPF 路由协议 **5**
2.1 路由协议概述 5
2.2 OSPF 协议概述 6
2.2.1 区域 6
2.2.2 协议数据包 7
2.2.3 邻居协商过程 9
2.3 协议安全机制 13
2.3.1 协议包认证 13
2.3.2 自反击机制 14
2.3.3 过程化的检查与约束 14
2.3.4 其它安全机制 15
2.4 OSPF 协议安全性分析 16
2.4.1 安全威胁 16
2.4.2 现有攻击技术 16
2.4.2.1 JiNao Team 攻击 17
2.4.2.2 E.Jones 攻击 17
2.4.2.3 Gabi Nakibly 攻击 18
2.5 本章小结 19
第 3 章 路由欺骗攻击 **20**
3.1 邻接欺骗攻击 20
3.1.1 攻击原理 20
3.1.2 攻击应用分析 20

3.1.2.1 网络欺骗 21
3.1.2.2 邮箱密码嗅探 21
3.1.2.3 中间人攻击 22
3.1.2.4 DNS 欺骗 22
3.2 双 LSA 远程多注入攻击 23
3.2.1 Nakibly 双 LSA 注入攻击 23
3.2.2 双 LSA 远程多注入攻击 25
3.2.2.1 污染区域分析 25
3.2.2.2 双 LSA 注入攻击的改进 26
3.2.3 攻击应用分析 27
3.3 单路径注入攻击 27
3.3.1 攻击原理 27
3.3.2 攻击应用分析 30
3.4 远程邻接欺骗攻击 30
3.4.1 协议探测阶段 30
3.4.2 远程邻接欺骗阶段 32
3.4.3 恶意 LSA 注入阶段 33
3.4.4 攻击应用分析 33
3.5 仿真测试 34
3.5.1 仿真环境搭建 34
3.5.2 邻接欺骗攻击 35
3.5.3 双 LSA 远程多注入攻击 37
3.5.4 单路径注入攻击 40
3.5.5 远程邻接欺骗攻击 42
3.6 本章小结 44
第 4 章 OSPF 渗透测试系统的实现 46
4.1 系统总体架构 46
4.2 监测模块 46
4.2.1 OSPF 探测模块 47
4.2.2 协议分析模块 47
4.2.3 密钥认证破解模块 48
4.2.3.1 密钥认证破解原理 48
4.2.3.2 破解模块实现 49
4.2.3.3 密钥认证安全性测试 50
4.2.4 OSPF 协议模块 51
4.2.4.1 Quagga 开源路由软件 51

4.2.4.2 OSPF 协议模块与 Quagga 的交互 52
4.3 攻击测试模块 53
4.3.1 协议包构造模块 54
4.3.2 协议配置接口模块 55
4.3.3 最大序列号测试模块 56
4.3.4 LSA 覆盖测试模块 56
4.3.5 双 LSA 注入测试模块 57
4.4 本章小结 58
第 5 章 OSPF 安全性测试 60
5.1 测试环境 60
5.2 测试实现与结果 60
5.2.1 系统接入 61
5.2.2 配置注入 61
5.2.2.1 网络路由注入 62
5.2.2.2 DNS 路由注入 66
5.3 本章小结 68
第 6 章 总结与展望 70
6.1 全文总结 70
6.2 下一步工作展望 70
参考文献 72
致谢 74
攻读硕士学位期间的科研成果 76

插图目录

图 1-1 2013 上半年中国网民各种安全问题发生率 1
图 2-1 OSPF 协议路由表生成过程 6
图 2-2 OSPF 网络与区域 7
图 2-3 OSPF 协议包结构（LSU 类型） 7
图 2-4 OSPF 协议包头结构 8
图 2-5 LSA 头部结构 8
图 2-6 LSA 接收处理流程 9
图 2-7 邻居状态转换流程图 10
图 2-8 邻居状态转换与 OSPF 包的关系 12
图 2-9 MD5 散列值生成的过程 13
图 2-10 自反击机制示意图 14
图 2-11 协议包校验流程 15
图 3-1 网络攻击示意图 21
图 3-2 邮件客户端与 IMAP 服务器间的交互过程 22
图 3-3 抗自反击机制示意图 23
图 3-4 抗反击 LSA 结构图 25
图 3-5 污染区域示意图 26
图 3-6 链路状态数据库污染示意图 27
图 3-7 跳板路由器及源污染路由器定义示意图 28
图 3-8 几种 LSA 发送的不同情况 29
图 3-9 协议探测流程 31
图 3-10 协议探测包发送流程 31
图 3-11 远程邻接欺骗示意图 32
图 3-12 远程邻接欺骗攻击示意图 33
图 3-13 路由欺骗仿真拓扑图 35
图 3-14 以 ASBR 身份注入路由的配置 36
图 3-15 攻击后 R10 与 R2 的路由表 36
图 3-16 用户 C3 到 121.195.178.1 的传输路径 36
图 3-17 攻击前用户 C3、C4 间的传输路径 37
图 3-18 攻击后 R5 与 R8 上的 R8 路由器 LSA 39
图 3-19 攻击前后 R5 的路由表 39
图 3-20 攻击后用户 C3、C4 间的传输路径 40
图 3-21 攻击后 R6 与 R3 上的 R3 路由器 LSA 42
图 3-22 攻击后用户 C3 到 C2 的传输路径 42
图 3-23 远程邻接欺骗攻击示意图 42
图 3-24 远程邻接欺骗攻击包交互过程 44
图 3-25 远程邻接欺骗攻击前后 R8 的路由表 44
图 4-1 系统总体架构 46
图 4-2 监听程序运行流程 47
图 4-3 MD5 认证过程 49
图 4-4 破解模块结构图 49
图 4-5 quagga 开源路由软件的结构 52

图 4-6 OSPF 协议模块与 quagga 的交互 53
图 4-7 协议包构造程序流程..... 55
图 4-8 协议配置模块交互示意图..... 55
图 4-9 最大序列号测试流程..... 56
图 4-10 LSA 覆盖测试流程..... 57
图 5-1 真实 OSPF 网络拓扑图..... 60
图 5-2 协议分析结果..... 61
图 5-3 测试者与部分受害者位置..... 61
图 5-4 测试前网络的部分路由表..... 62
图 5-5 测试后网络的部分路由表..... 63
图 5-6 网络路由注入前测试结果..... 63
图 5-7 网络路由注入后测试结果..... 64
图 5-8 子网掩码划分示意图..... 65
图 5-9 IMAP 服务器嗅探结果..... 65
图 5-10 认证失败窗口..... 65
图 5-11 xxx.edu.cn 域的正向解析配置..... 66
图 5-12 DNS 路由注入前测试结果..... 67
图 5-13 DNS 路由注入后测试结果..... 67

表格目录

表 2.1 OSPF 协议包描述 7
表 2.2 各种 LSA 类型的描述 9
表 2.3 邻居状态转换事件 11
表 2.4 DD 包中位字段含义 13
表 3.1 抗反击 LSA 包的主要参数 38
表 3.2 单恶意 LSA 的主要参数 40
表 3.3 幻影路由器 LSA 的主要参数 43
表 4.1 Hello 包的关键字段默认值 48
表 4.2 密钥认证安全性测试环境参数 50
表 4.3 破解时间与字符长度的测试结果 50
表 4.4 破解时间与字符种类的测试结果 51
表 4.5 包构造主要的函数接口 54
表 4.6 不同类型 LSA 的矫正字段 57

第1章 绪论

1.1 研究背景

2013 年 6 月，前美国中央情报局职员斯诺登披露了美国国家安全局的“棱镜计划”。该计划监控各种用户的信息，包括即时消息、电子邮件、视频、照片、存储数据、语音聊天等等。许多知名的科技公司（如：微软、雅虎、谷歌等）被指加入了该计划，并且提供了用户的各种秘密信息[1]。此外，斯诺登还表示美国政府入侵中国网络至少有四年时间[2]，美国政府黑客攻击的目标达到上百个，其中还包括了学校。黑客的攻击方式主要通过入侵巨型的路由器，然后一举攻击成千上万台电脑，无需一一攻击个别电脑。

斯诺登事件让信息安全上升到前所未有的高度，进入了更多人的视野。人们开始关注自己的个人信息安全，许多国家和企业也加大了在信息安全领域的投入。比如，在今年的全国两会上，信息安全成了人们关注的一大“热词”[3]，多位全国人大代表就信息安全问题提出了建议。在美国，包括美国银行在内的六大银行，以及万事达卡均宣布将建设金融网络[1]，加强信息安全。

网络安全是信息安全的重要组成部分。随着互联网的不断发展，网络已经成为人们生活必不可少的一部分，然而网络所遭受的攻击也越来越多。常见的网络攻击方法[4]主要有口令攻击、木马程序攻击、网络欺骗、邮件攻击、网络监听、系统漏洞攻击、拒绝服务攻击、缓冲区溢出攻击、会话劫持攻击等。其中，网络欺骗是一种重要的攻击方式。图 1-1 显示了 2013 年上半年中国网民各种安全问题的发生率[5]，其中欺诈、诱骗信息、假冒网站等都可以用网络欺骗实现。

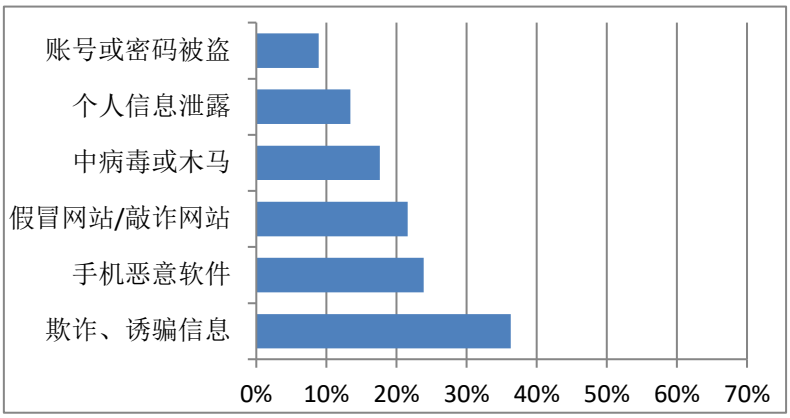

图 1-1 2013 上半年中国网民各种安全问题发生率

网络欺骗主要分为两大类，一类是通过技术手段进行欺骗，另一类利用非技术手段进行欺骗。非技术类欺骗主要通过社会工程学技术来实现，攻击者利用人类天性（如贪婪、好奇、容易接受他人说辞的倾向、心理弱点等）绕过安全防御系统。技术类手段主要分为协议欺骗技术、空间欺骗技术、蜜罐和密网技术、网络钓鱼技术等[6]。空间欺骗

技术是指通过增加搜索空间来增大攻击者的工作量，从而实现安全防护。蜜罐是用来被探测、被攻击甚至被攻陷的网络欺骗攻击。蜜网（也叫诱捕网络）是一类研究型的高交互蜜罐技术，它的目的是收集攻击者的信息。这两种欺骗技术更多的用于增强网络的安全性。网络钓鱼技术是指通过攻击者利用欺骗性的电子邮件或伪造的 Web 网站来进行网络诈骗活动，受害者常常会泄露自己的私人信息，如用户名、口令等等。

网络协议欺骗是指攻击者利用协议的安全漏洞实现攻击，通过伪造或修改信息来增强隐蔽性，提高攻击质量。网络协议欺骗的攻击方法有很多[7]，常见的有 ARP 欺骗、IP 欺骗攻击、路由欺骗、TCP 欺骗、DNS 欺骗等等。路由欺骗技术是指攻击者通过各种手段改变路由器或主机的路由表，从而可以影响流量的传输路径。路由欺骗对网络的危害很大，只需要攻击一台路由器就可以影响很多的用户。

1.2 路由欺骗技术

目前路由欺骗的方法不多，常见的有基于 ICMP 的路由欺骗技术、基于路由协议的欺骗技术等，下面将分别介绍这几种欺骗技术。

1. 基于 ICMP 的路由欺骗

基于 ICMP 的路由欺骗技术主要分为 ICMP 重定向欺骗以及 ICMP 路由通告欺骗[8]。ICMP 重定向欺骗主要利用 ICMP 重定向报文假冒路由器修改网络中主机的动态路由表，从而实现路由欺骗的目的。ICMP 路由通告欺骗是指攻击者伪造 ICMP 路由器通告报文，将子网主机的默认路由设置为自己的 IP 地址。基于 ICMP 的路由欺骗利用了协议自身的漏洞，具有较强的隐蔽性。此外，基于 ICMP 的路由欺骗技术可以与 ARP 欺骗等其它网络欺骗手段相结合，实现更强大的攻击。

2. 基于路由协议的路由欺骗

目前路由协议的种类很多，每种路由协议都存在着不同程度的缺陷。在某些场景下，利用这些缺陷可以实现路由欺骗。下面以内部网关路由协议中的 RIP 以及 IGRP 来介绍如何实现路由欺骗。

RIP 路由协议[9]是无连接的，以 UDP 的方式运行。因此，在缺少认证或者认证被绕过的情况下，攻击者很容易向 RIP 路由器发送伪造的协议数据包。路由器收到伪造的数据包后一般很难辨别它的真实性，这样路由器的路由表就会被恶意篡改，从而实现了路由欺骗。

IGRP 路由协议[9]由于没有认证机制，攻击者如果想要加入 IGRP 路由域，只需要知道自治系统号。如果攻击者位于本地，可以通过嗅探 IGRP 的更新包来获得此自治系统号，而 IGRP 每隔 90 秒会向 224.0.0.10 广播发送更新包。如果攻击者不在本地，可以使用 IRPAS 软件中的 ass 命令来破解此号码。在获得自治系统号后，攻击者可以向网络中注入恶意的路由，实现路由欺骗。

OSPF 路由协议具有多种安全机制，相比于 RIP、IGRP 等路由协议，它的安全性比较好，因此 OSPF 协议是目前使用最为广泛的内部网关协议，本文主要研究基于 OSPF 协议的路由欺骗。

1.3 课题研究现状

关于 OSPF 攻击主要分为两类，一类是路由欺骗攻击，另一类主要是消耗路由器的资源，下面将分别介绍它们的研究现状。

1.3.1 资源消耗攻击

1999 年 JiNao Team 描述了几种攻击[10]（除了最大序列号攻击），攻击者恶意的发送自治系统中其它路由器的 LSA（Link State Advertisement），这些描述的攻击都会触发受害路由器的自反击机制。攻击者不断的发送恶意的 LSA，就会不断的触发路由器的自反击机制，从而消耗它的资源。另一方面，这些攻击使得 AS（Autonomous System）中的路由处理变得不稳定，但是这也增加了攻击暴露的几率，使管理员容易发现它的位置。

2006 年 E. Jones 和 O. Moigne 测试了过去 OSPF 协议中所有不同的方式，同时也提出了一些新的攻击方式[11]，其中邻居表溢出攻击和链路状态数据库溢出攻击主要是通过消耗路由器的邻居表和链路状态数据库资源来实现拒绝服务攻击。

1.3.2 路由欺骗攻击

1997 年 Feiyi Wang、Brian vetter 等讨论了一种攻击[12]，区域内的路由器模仿 ASBR（Autonomous System Boundary Router），然后宣告 AS 外部路由信息。这种攻击使得到这些目的地的一部分或所有的流量都会被引到攻击者这边，导致流量黑洞、侦听或经过一个更长的路由。

1999 年 JiNao Team 提出了最大序列号攻击[10]，这种攻击利用了协议实现时不会冲掉最大序列号 LSA 的漏洞，实现了对其它路由器路由表的篡改。

2006 年 E. Jones 和 O. Moigne 提出了周期注入 LSA 以及邻接中断攻击[11]，周期注入 LSA 可以使自反击无效，影响路由器的路由表。而邻接中断攻击主要使路由器间邻接关系中断，从而使路由表在这段时间内不正常。

2011 年 Gabi Nakibly、A. Kirshon 等提出了使用两个 LSA 来改变其它路由器 LSA 的方法[13][14]。这种方法使得攻击者可以对网络拓扑进行控制，实现包括路由欺骗在内的各种攻击。此外，他们还提出了一种与远端路由器建立虚假邻接关系的攻击方法，篡改它们的路由表。

2013 年 Gabi Nakibly、Eitan Menahem 等发现了大多数路由器厂商在实现 OSPF 协议时的一个漏洞[15]。利用这个漏洞攻击者可以改变任何路由器自身的 LSA，而不会引起

自反击机制。这个漏洞对于网络的危害很大，攻击者可以任意的改变其它路由器的路由表，发起各种攻击。

1.4 论文具体工作

本文研究 OSPF 协议的安全性，在分析 OSPF 协议安全机制以及已有攻击方法的基础上，提出了四种新的攻击方法，分析了它们如何攻破协议的安全机制。通过搭建网络模拟平台，验证了上述四种路由欺骗攻击的可行性及有效性。接着设计实现了 OSPF 渗透测试系统，该系统可对 OSPF 网络进行安全性测试。最后，利用渗透测试系统在真实的网络环境中进行了渗透测试，并且实现了网络欺骗、密码嗅探以及 DNS 欺骗的效果。

本文的章节安排如下：

第一章首先阐述了课题的研究背景、典型的路由欺骗技术，然后介绍了课题目前的研究现状。

第二章首先介绍了几种常见的路由协议，分析它们的优缺点；接着从区域、协议数据包以及邻居协商过程三方面详细介绍了 OSPF 协议的原理及运行机制；然后介绍了协议的安全机制，了解协议如何抵抗攻击；最后分析了协议的安全性，了解协议的安全威胁以及现有的一些攻击技术。

第三章提出了邻接欺骗攻击、双 LSA 远程多注入攻击、单路径注入攻击、远程邻接欺骗攻击这四种路由欺骗攻击方法，分析了它们如何攻破协议的安全机制（主要是自反击机制），并对它们的应用场景进行分析。接着利用 GNS3 网络仿真系统、VMware 虚拟机以及真实物理计算机搭建了一个高仿真程度的网络模拟平台，验证了上述四种路由欺骗攻击的可行性及有效性

第四章设计了 OSPF 渗透测试系统的总体架构，对系统每一个模块的实现进行详细的介绍，并利用该系统对 OSPF 协议密钥认证机制的安全性进行了评估。

第五章利用 OSPF 渗透测试系统对真实的 OSPF 网络进行渗透测试，利用邻接欺骗攻击实现了网络欺骗、密码嗅探以及 DNS 欺骗的效果。

第六章总结了本文所做的工作，并指出了下一步的工作方向。

第2章 OSPF 路由协议

本章首先简单的介绍了各种路由协议，接着对 OSPF 路由协议的原理以及运行机制进行详细的介绍，最后对 OSPF 协议的安全机制以及安全性进行分析。

2.1 路由协议概述

路由协议一般运行于路由器之中，使路由器建立路由表，从而实现路径的选择以及数据包的转发。

路由协议分为两大类，一类是静态路由协议，它是由网络管理员手工配置生成，不随网络拓扑而变；另一类是动态路由协议，运行该类协议的路由器相互交换路由信息动态的建立路由表，并随网络拓扑而改变。动态路由协议按作用范围可以划分为内部网关协议（Interior Gateway Protocol，IGP）和外部网关协议（External Gateway Protocol，EGP）。IGP 作用于自治系统内部，而 EGP 作用于自治系统外部。在这里，自治系统（Autonomous System，AS）是指一个具有统一管理机构，统一路由策略的网络。常见的 IGP 有 RIP、RIPv2、IGRP、EIGRP、OSPF、ISIS 等[16]，常见的 EGP 有 BGP 等。根据路由算法，可以将路由协议分为距离矢量路由协议以及链路状态路由协议。距离矢量路由协议主要基于 Bellman-Ford 算法，有 RIP、RIPv2、IGRP、EIGRP、BGP 等；链路状态路由协议主要基于 Dijkstra 算法，有 OSPF、IS-IS 等。下面将对这几种路由协议进行简单的介绍和比较[17]。

- RIP、RIPv2 是开放的路由协议，适合于不同厂商路由器的互联，并且配置比较简单，但是它们对于链路带宽、CPU 以及内存资源的消耗很大，当路径较多时收敛速度很慢，所以只适合网络结构简单的小型网络。
- IGRP 是思科私有的协议，它减小了带宽的消耗，但是由于收敛的时间很长，不适合于大中型网络，所以已经被淘汰。为了解决 IGRP 的不足，思科公司开发了 EIGRP，它具有快速收敛、带宽消耗低、CPU 占有率低、适合于大中型网络等优点。由于 EIGRP 也是私有的协议，所以限制了它的使用范围。
- IS-IS 协议是一种开放的路由协议，它具有快速收敛、网络汇总、带宽消耗低、支持手工汇总、二级异构拓扑技术、适合于大中型网络等优点，但是 IS-IS 协议属于 OSI 体系，一般应用于运营商网络。
- BGP 协议是目前主要的外部网关路由协议[18]，一般应用于不同自治系统之间，而本文研究的主要是自治系统内的安全性，所以不对 BGP 协议进行讨论。
- OSPF 协议是一种开放的路由协议，它与 IS-IS 协议在性能以及质量上差别不大，但是它更适合于 IP 网络，比 IS-IS 协议更具有活力。

综上所述，相比于其它内部网关路由协议，OSPF 协议的性能以及质量更好，它已经成为目前内部网络中使用最为广泛的路由协议，下面将对 OSPF 协议进行详细的介绍。

2.2 OSPF 协议概述

开放最短路径优先（Open Shortest Path First，OSPF）协议[19]是由 Internet 工程任务组开发的开放性路由协议。它是一种链路状态路由协议，不同于距离矢量协议，路由器在交换路由信息前需要先建立邻接关系，同步各自的链路状态数据库，链路状态数据库反映了整个网络拓扑的情况，接着路由器利用最短路径优先算法生成最短路径树，并根据最短路径树构造路由表。图 2-1 描述了路由表生成的过程。

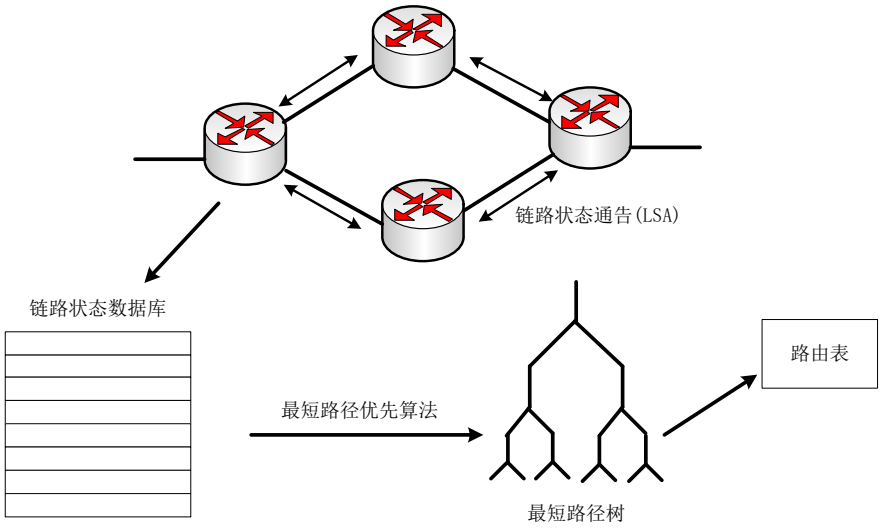

图 2-1 OSPF 协议路由表生成过程

2.2.1 区域

OSPF 协议是一个二层路由协议，它可以把一个大型的网络分割成许多小型的网络，而这些小型的网络就称为区域。图 2-2 是一种典型的 OSPF 网络，区域 1 和区域 2 需要通过区域 0 进行流量的传输。这里区域 0 就是骨干区域，区域 1 和区域 2 为常规区域，非骨干区域都要与骨干区域相连（虚链路除外）。除了骨干区域以及常规区域外，还有一些特殊区域，如存根区域、完全存根区域、NSSA 区域等，这里不作讨论。

根据路由器与区域的关系，可以将路由器分为四种类型[20]，分别为内部路由器、区域边界路由器、骨干路由器以及自主系统边界路由器。内部路由器是指所有的接口都在同一区域内的路由器，如 R5、R4。区域边界路由器（Area Border Router，ABR）是指连接一个或多个区域到骨干区域的路由器，如 R2、R3。骨干路由器是至少一个接口连接骨干区域的路由器，如 R1、R2、R5 等。自主系统边界路由器（Autonomous System Boundary Router，ASBR）是指连接外部网络与 OSPF 网络的网关路由器。

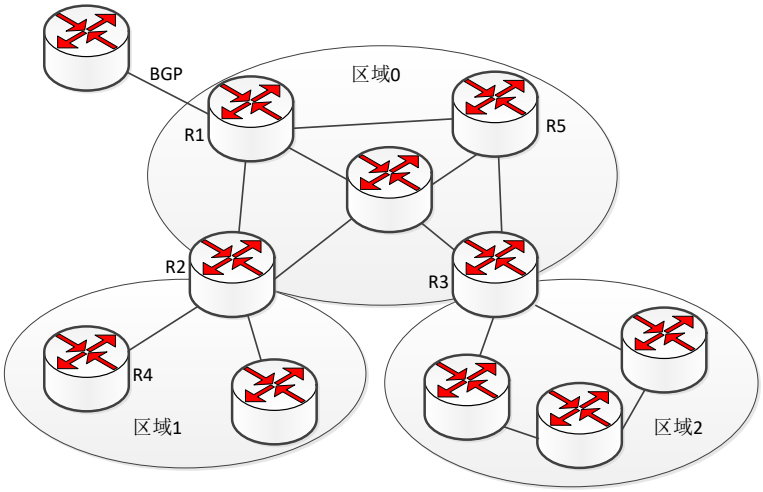

图 2-2 OSPF 网络与区域

2.2.2 协议数据包

在 OSPF 协议中，协议数据包是路由器之间交换路由信息的载体，同时它也是路由器发现邻居、建立邻接关系、相互协商的重要媒介。

OSPF 协议包直接封装在 IP 包中，它的协议号为 89。一个完整的协议包结构如图 2-3 所示（以 LSU 类型为例）。

图 2-3 OSPF 协议包结构（LSU 类型）

OSPF 协议包头部结构如图 2-4 所示，根据包类型字段取值的不同可以将 OSPF 协议包分为 5 种类型，分别为 Hello 包、数据库描述（DD）包、链路状态请求（LSR）包、链路状态更新（LSU）包以及链路状态确认（LSAck）包。表 2.1 描述了这些 OSPF 协议包。

表 2.1 OSPF 协议包描述

协议包类型	类型字段取值	作用
Hello 包	1	用于发现路由器的邻居和维持路由器间的邻居关系
DD 包	2	一方面用于确定邻居路由器间的主/从关系及初始数据包序列号，另一方面用于传送链路状态数据库的信息摘要
LSR 包	3	用来请求较新的链路状态通告
LSU 包	4	包含了具体的链路状态通告信息
LSAck 包	5	用于对收到的 LSU 包进行确认。

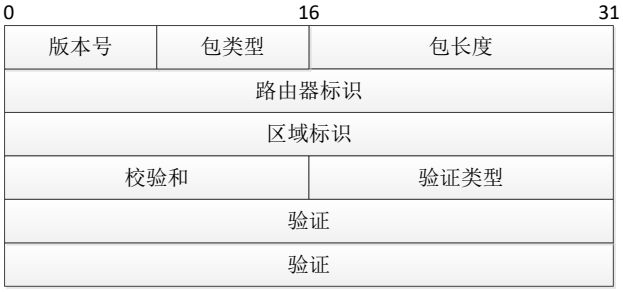

图 2-4 OSPF 协议包头结构

链路状态通告（Link State Advertisement，LSA）是描述 OSPF 网络拓扑的分组，协议进程会根据链路的情况自动生成相应的 LSA，并以 LSU 协议包为载体进行发送。所有有效的 LSA 都会被存放于路由器的链路状态数据库中，路由器就是根据这个链路状态数据库来计算得到路由表，所以 LSA 对于路由表的计算至关重要。

一个完整的 LSA 由 LSA 头部以及具体 LSA 内容组成。LSA 头部结构如图 2-5 所示，LS 类型、LS 标识以及宣告路由器可以唯一的确定 LSA。在同一时间内有时会存在 LSA 的多个实例，LS 时限、LS 序列号以及 LS 校验和可以确定出哪一个实例较新。

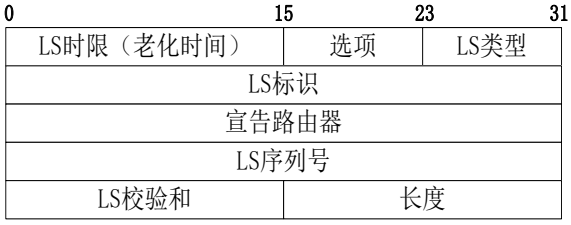

图 2-5 LSA 头部结构

为了保证 LSA 的真实性以及完整性，OSPF 协议进程会对收到的 LSA 进行严格的验证，具体的处理过程如图 2-6 所示。当收到一个 LSA 后，需要判断它是否已经存在于数据库中，这个主要根据 LS 类型、LS 标识以及宣告路由器这三个字段。如果不存在，把 LSA 添加到链路状态数据库中，然后完成发送 LSAck 包、泛洪 LSA 以及计算新路由表；如果存在，需要判断这个 LSA 与数据库中的 LSA 哪一个更新，这主要根据 LS 时限、LS 校验和以及 LS 序列号来判断。判断的规则如下：

1) LS 序列号大的 LSA 更新。
2) 如果序列号相同，则校验和较大的 LSA 更新。
3) 如果序列号以及校验和都相同，当 LS 时限为 MaxAge，则这个 LSA 较新。否则，当两个 LSA 实例的 LS 时限差异大于 15 分钟，较小时限的实例更新。如果差异小于 15 分钟，那么这两个实例就相同。

根据上面的规则，如果比数据库中的 LSA 更新，则跳转至 A 执行一系列动作。如果没有数据库中的 LSA 新，需要根据以上规则判断是否与本地 LSA 相同，相同的话就忽略接收的 LSA，不同的话，则需要发送一个携带更新 LSA 的 LSU 包给发送端。

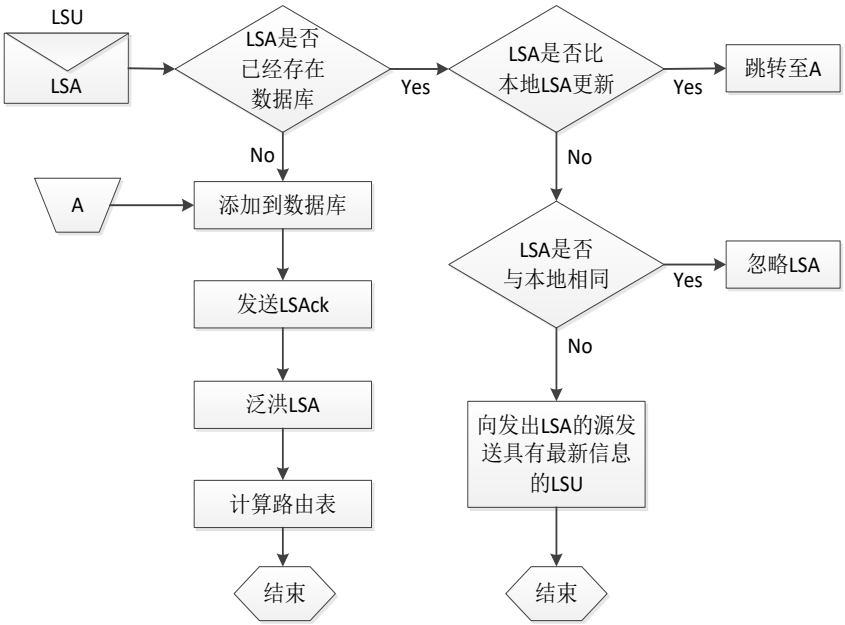

图 2-6 LSA 接收处理流程

LSA 有多种不同的类型，具体类型由 LS 类型字段决定，表 2.2 列出了几种主要的 LSA 并描述了它们的意义。

表 2.2 各种 LSA 类型的描述

LSA 类型	LS 类型值	描述
路由器 LSA	1	路由器对于所接入的区域都会产生一个路由器 LSA，描述了路由器接口、链路以及 Cost 值等信息，它只能在区域内泛洪。
网络 LSA	2	由指定路由器产生，描述了一个多接入网段中所有的路由器 ID，包含它自己，只能在区域内泛洪。
网络汇总 LSA	3	由 ABR 产生，将本区域内部的网络信息通告给其它的区域。
ASBR 汇总 LSA	4	由 ABR 产生，用于向其他区域通告一台 ASBR 路由器。
AS 外部 LSA	5	由 ASBR 产生，用于向自治系统内部通告外部网络的信息。

2.2.3 邻居协商过程

OSPF 协议适用于几种不同的网络类型，包括点到点网络、点到多点网络、广播型网络、非广播多路访问（NBMA）网络以及虚链路。它主要根据链路层的协议来区分不同的网络。

邻居协商过程与网络的类型有关，另外协议数据包的发送方式也与网络类型有关。由于在自治系统内部广播型网络的应用比较广泛，所以下面将以广播型网络为基本网络类型来进行分析。

邻居协商过程的目的在于发现邻居路由器，并且与邻居路由器建立邻接关系。只有建立了邻接关系的路由器，才能互相交互路由信息。邻接关系的建立过程主要分为 4 个阶段：邻居路由器发现阶段，双向通信阶段，数据库同步阶段以及完全邻接阶段。图 2-7 显示了邻居状态从失效状态到完全邻接状态的过程。

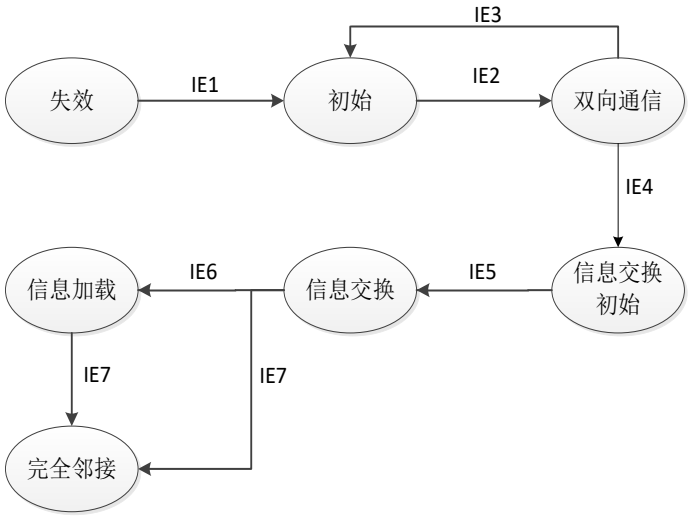

图 2-7 邻居状态转换流程图

失效状态是邻居状态转换中最初始的状态，表示在最近的路由死亡时间（默认 40 秒）内还没有收到邻居路由器发送的 Hello 包。

初始状态表示收到了从邻居发送的 Hello 包，但是路由器自身没有包含在 Hello 包的邻居列表中，即通信是单向的。此时，路由器会添加一个邻居数据结构来保存邻居路由器的相关信息。

双向通信状态表示邻居路由器建立了双向通信，这是建立邻接关系的基本条件。在广播型网络中，这个状态还会选举指定路由器以及备份指定路由器。在多接入的网络中，OSPF 协议会选举一个指定路由器（Designated Router，DR）负责收集和分发 LSA，这样可以减少网络的协议流量；还会选举一个备份指定路由器（Backup Designated Router，BDR），以防止 DR 发生故障后导致网络中断；选举完后，其它路由器都会变成 DROther。

信息交换初始状态、信息交换状态以及信息加载状态属于数据库同步阶段，信息初始化状态主要确定本地路由器和邻居路由器的主/从关系，并确定数据库描述包的初始序列号，为数据库描述包交换做准备。信息交换状态下路由器将描述整个链路状态数据库的数据库描述包发送给邻居路由器，同时路由器也会发送 LSR 包请求最新的链路状态通告。信息加载状态下主要发送 LSR 包请求一些还没有收到的 LSA。

完全邻接状态是整个状态转换的最终状态，当达到这个状态时说明邻居路由器间已同步了链路状态数据库。这个状态需要用 Hello 包（默认每 10 秒发送一次）来维持，否则有关邻居路由器的信息就会被清除，并且邻居状态变为无效。

表 2.3 列举了在邻居状态转换过程中几个主要的转换事件，一般邻居状态的转换通过协议包的收发来触发。

表 2.3 邻居状态转换事件

输入事件	描述
IE1	从邻居路由器收到了一个有效的 Hello 包
IE2	收到的 Hello 包中邻居字段包含自身路由器 ID 或收到数据库描述包
IE3	收到的 Hello 包中邻居字段中不再包含自身路由器 ID
IE4	与邻居路由器能形成邻接关系（如：DRother 间不能形成邻接关系）
IE5	路由器之间协商完成，确定了主/从关系以及初始的描述序列号
IE6	链路状态请求列表中存在请求的内容
IE7	链路状态请求列表为空

路由器邻居关系的建立主要基于 OSPF 协议包。图 2-8 描述了 OSPF 协议包与邻居状态之间的关系以及链路状态数据库同步的过程（这里假设路由器 RT2 比路由器 RT1 先运行 OSPF 协议，但是时间间隔小于 10s）。

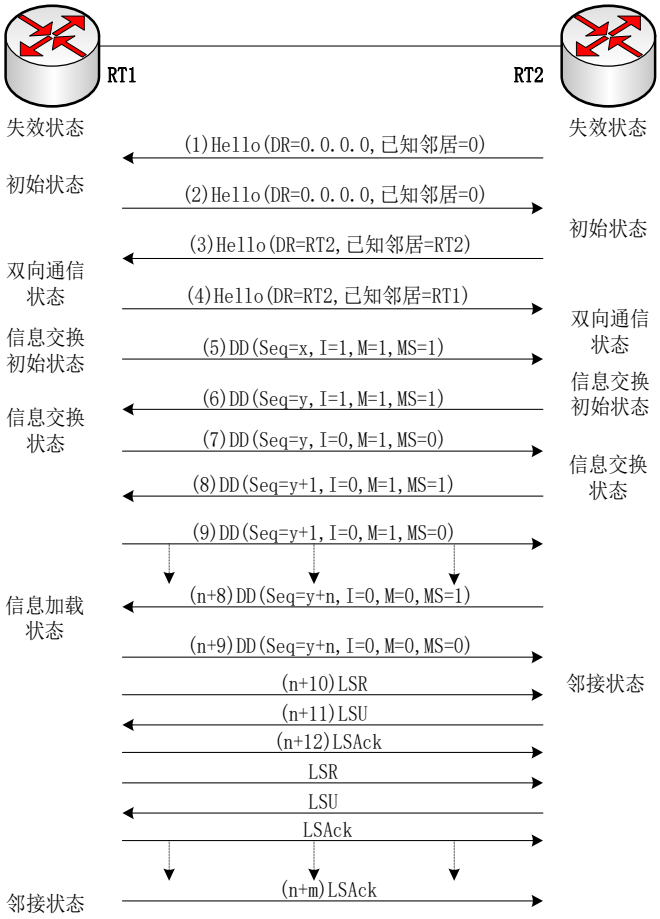

图 2-8 邻居状态转换与 OSPF 包的关系

从图 2-8 中可以看到 RT1 和 RT2 的邻居状态转换过程有一点区别，这与具体情况有关。当说 RT1 的邻居状态转换时，实际上它的意思是以 RT1 为基点来观察 RT2 的状态情况，所以在上面标出的状态实际上都是对方路由器在本地路由器上的状态。下面针对上述过程做一个简要的分析：

步骤 1~2：这个阶段为邻居发现阶段。由于还没有学习到邻居，可以看到 Hello 包的邻居字段为空，DR 字段为 0.0.0.0。在收到第一个 Hello 包后，路由器会创建邻居数据结构，并且把邻居路由器的状态设置为初始状态。

步骤 3~4：这个阶段为双向通信阶段。此时，路由器收到的 Hello 包中邻居字段不再为空，并且 DR 和 BDR 也已经选择完成，DR 为 RT2 的接口地址，所以 RT1 收到第 3 个包后邻居状态进入双向通信状态，RT2 收到第 4 个包后邻居状态进入双向通信状态。

步骤 5~n+m：这个阶段 RT1 和 RT2 的邻居状态转换有一些不同。因为 RT1 和 RT2 可以建立邻接关系，所以在进入双向通信阶段后，会进入信息加载阶段和完全邻接阶段。I 位、M 位以及 MS 位是 DD 包中最主要的三个字段，它们决定了状态之间的转换，具体的含义如表 2.4 所示。第 5 和第 6 个 DD 包中 I 位为 1，表明双方协商主/从关系和初始序列号。在第 7 到 n+9 个包之间，当 I 位和 M 位都为 1 时，表示双方处于信息交换状态；而当 I 位为 1、M 位为 0 时，邻居状态就会进入信息加载或完全邻接状态。由于

RT2 不需要向 RT1 请求新的链路状态，所以它的邻居状态直接进入了邻接状态；而 RT1 需要向 RT2 请求最新的链路状态，所以它的邻居状态进入了加载状态。RT1 在发送最后一个 LSAck 包后不再发送 LSR 包了，所以它的邻居状态进入了邻接状态。

需要注意的是上面的邻接过程是一种理想化的交互流程，目的是为了使邻接建立的过程更加容易理解。真实的邻接过程不会像图 2-8 描绘的那么整齐有序，但是它更加的高效，比如：当 RT1 的邻居状态还处于信息交换状态时，RT1 就已经开始发送 LSR 包。

表 2.4 DD 包中位字段含义

取值 / DD 包的位	0	1
I 位	表示 DD 包不是初始包，这时处于信息交换状态	表示 DD 包是初始包，这时处于信息交换初始状态
M 位	表示后面没有要发送的 DD 包	表示 DD 包还没有发送完
MS 位	表示自身为从路由器	表示自身为主路由器

2.3 协议安全机制

OSPF 路由协议相比 RIP、EIGRP 协议更加安全，它有非常完善的安全机制，可以抵挡一般的攻击。下面将介绍 OSPF 协议一些主要的安全机制，正是有了这些安全机制，攻击者很难对 OSPF 协议进行有效的攻击。

2.3.1 协议包认证

OSPF 协议规定了三种认证方式[19]，分别为空认证、明文认证以及密钥认证。空认证意味着 OSPF 协议包没有被验证，协议包头中 64 位的验证域字段不包含任何内容。明文认证是一种 64 位“明文”密钥的认证，协议包头中 64 位的验证域字段被设为所配置的值，当其它路由器收到协议包时会比较验证域的值与自身所配置的“明文”密钥是否相同。密钥认证采用了 MD5 算法，图 2-9 描述了 MD5 散列值生成的过程[21]。根据 MD5 算法的要求[22]，在 OSPF 包和发送端密钥后需要加上填充域及长度域，使整个待认证的数据包长度满足 512 位的倍数，生成的散列值加到 OSPF 包后一起发送。

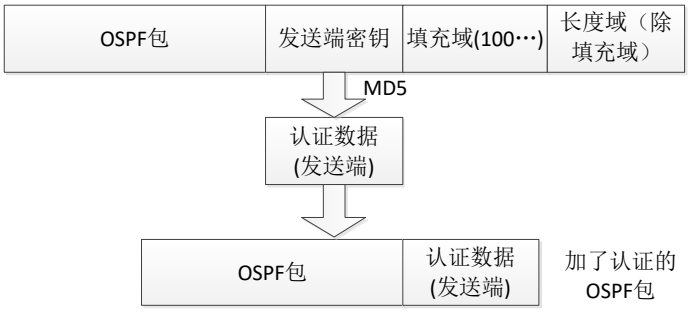

图 2-9 MD5 散列值生成的过程

上述三种认证方式中空验证以及明文验证安全性不高，一般攻击者通过嗅探就可以获取“明文”密钥，所以它们不能抵抗外部攻击以及内部攻击。而密钥认证可以抵抗外部攻击，但是不能抵抗内部攻击。当然密钥认证有可能会被破解，它的安全性与许多因素有关，比如说密钥长度、密钥字符种类、计算机处理能力等等。

明文认证和密钥认证的认证方式也分为两种，一种为单链路认证，另一种为区域认证。单链路认证是指每一条链路两端的路由器有相同的认证方式和密钥，而区域认证是整个区域里的路由器使用相同的认证方式和密钥。相比较而言，单链路认证更加的安全，但是由于缺少共享密钥分配机制，网络管理员需要对每一台路由器进行配置[23]，这就导致了今天很多自治系统中的密钥都相同，从而留下了安全隐患。

2.3.2 自反击机制

自反击机制[14]是 OSPF 协议安全机制中最主要的机制，它指一旦路由器收到一个它自己的 LSA 实例，而这个实例比路由器中对应的实例更新，那么路由器就会立即宣告一个比接收到的 LSA 更新的实例。结合泛洪机制，它就保证了在其他路由器中保存的 LSA，都是从产生该 LSA 的路由器中发出，攻击者冒充真实路由器发送的恶意 LSA 会立刻被更新的 LSA 所覆盖。正是有了自反击机制，攻击者很难改变其它路由器中的链路状态数据库，也就无法影响其它路由器的路由表。

如图 2-10 所示，假设攻击者伪造了一个最新的 Rc 路由器的 LSA，然后发向网络中。网络中的路由器（除 Rc）收到这个 LSA 时，会将它存放在链路状态数据库中，因此它们的链路状态数据库受到了污染。不过，当 Rc 收到这个 LSA 后，它会发现这个 LSA 比自己发出的 LSA 更新，所以它会重新发送最新的 LSA，这样网络中的路由器就会存放真实的 LSA。

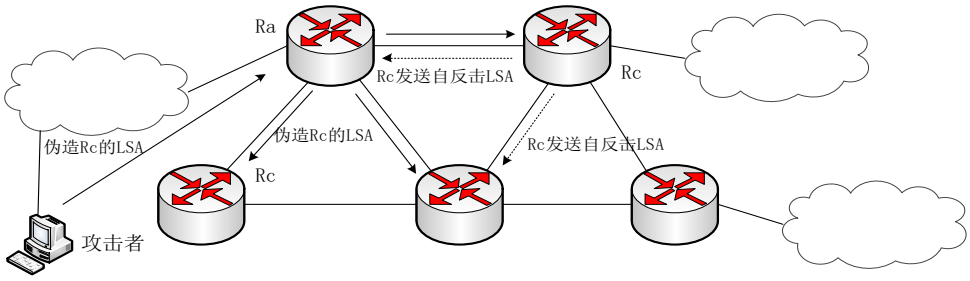

图 2-10 自反击机制示意图

2.3.3 过程化的检查与约束

OSPF 协议在接收到一个协议包后会进行严格的校验，如图 2-11 所示，一般分为三个阶段[24]。第一个阶段主要针对 IP 头进行校验，如 IP 源地址不能是路由器自己的接口地址，IP 校验和必须正确等；第二个阶段主要针对 OSPF 包头校验，如区域号要相同，认证方式及数据要匹配等；最后根据不同类型的包做相应的校验。这些校验步骤一方面

可以保证协议正确的运行，另一方面它增加了攻击的难度，简单伪造的数据包很难被路由器所接收。

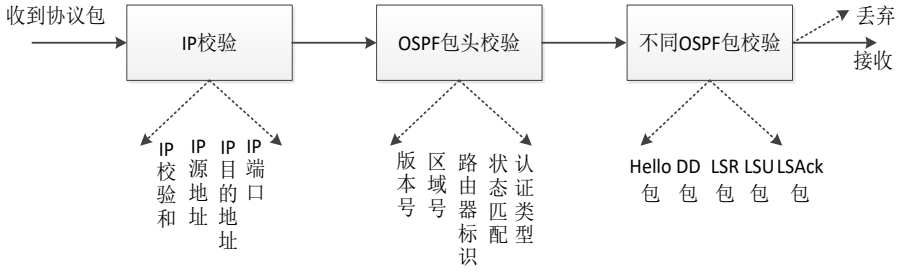

图 2-11 协议包校验流程

2.3.4 其它安全机制

除了上面一些主要的安全机制外，OSPF 协议还有一些其它的安全机制[14]。这些安全机制对于 OSPF 协议的安全也很重要。

1. 泛洪机制

泛洪机制是路由器收到 LSA 后转发的规则。可靠的泛洪机制可以保证同一区域内的路由器具有相同的链路状态数据库，只要有一条不经过恶意路由器的路径，那么恶意路由器就不能阻止 LSA 到达其他路由器。

2. 双向链路机制

在路由表生成过程中，只有那些被两端路由器都宣告的链路才会被加入计算中。如果攻击者宣告一个不存在的链路到另一个路由器，而另一个路由器没有宣告这条链路，那么这条链路对于路由表不会有任何影响。

3. LSA 不相关性

LSA 之间相互独立，若一个 LSA 受到攻击，它不会影响其它的 LSA。由于一个 LSA 只表示网络拓扑中的小部分，所以攻击者若要影响整个网络拓扑结构，必须对多个 LSA 进行攻击。

4. 层次路由与信息隐藏

引入区域后，路由分为了域内路由以及域间路由两种，协议会优先考虑域内的路由，所以区域内的路由不受其他区域的路由不稳定或错误配置影响。当一个区域受到攻击后，一般不会导致其他区域受到攻击。此外，由于区域的拓扑对其他区域是透明的，可以避免网络拓扑的整体暴露。

2.4 OSPF 协议安全性分析

2.4.1 安全威胁

虽然 OSPF 协议提供了协议包认证、自反击机制、过程化的检查与约束等安全机制，但是在某些情况下仍然存在以下威胁[25]。

1. 监听

OSPF 协议包中的内容是以明文的形式存在，所以监听会对路由信息的机密性产生威胁。

2. 消息注入

如果 OSPF 协议使用密钥认证机制，那么它不会受到外部攻击者的消息注入影响。然而在很多时候攻击者获得了密钥或者本身是内部路由器，那么消息注入就会对网络产生很大的安全威胁。

3. 消息删除

OSPF 协议本身提供了一定的保护机制来防止消息删除，接收路由器本身无法检测消息是否被删除，但是发送路由器可以检测到消息是否被删除。当攻击者没有收到 LSAck 包时，它就会重新发送 LSU 包。对于 Hello 包，OSPF 协议并没有确认机制，所以攻击者可以删除 Hello 包，从而导致邻接关系的断开。

4. 消息修改

配置了密钥认证的 OSPF 网络可以抵抗消息修改，但是如果攻击者获得了密钥或者本身是内部路由器，那么消息修改还是可以实现。

5. 中间人攻击

配置了密钥认证的 OSPF 网络可以抵抗中间人攻击，但是如果攻击者获得了密钥或者本身是内部路由器，那么中间人攻击还是可能实现，例如利用 ARP 欺骗、虚链路等。

6. 拒绝服务攻击

攻击者可以发送伪造数据包来实现拒绝服务攻击。比如说，攻击者可以发送大量恶意的 LSA，使受害路由器的链路状态数据库溢出。对于没有配置密钥认证或内部攻击来说，会受到拒绝服务攻击的威胁。

2.4.2 现有攻击技术

在 OSPF 网络中，攻击者对于 OSPF 协议的攻击分为两种类型[26]，分别为外部攻击以及内部攻击。外部攻击是指攻击者并没有参与到路由协议的处理进程中，即它与网络中的路由器并没有建立邻接关系。内部攻击是指攻击者参与到了路由协议的处理进程中，它与网络中的路由器建立了邻接关系。由于外部攻击对于网络的威胁不大，这里主要讨论的是内部攻击。下面介绍一些过去主要的攻击方式。

2.4.2.1 JiNao Team 攻击

JiNao Team 发现并实现了四种 OSPF 攻击[10]，它们都实现了拒绝服务攻击，假如改变包的内容，也会有其它的应用。

1. 最大年龄攻击

攻击者发送目标路由器的 LSA，并且设置 LSA 的年龄字段为 0xFF。目标路由器接收到这个数据包后，就会被引起自反击机制，从而发送更新的 LSA。如果攻击者不断的发送类似的数据包，就可能会耗尽目标路由器的资源，使路由器运行不正常。

2. 序列号增量攻击

攻击者向目标路由器发送 LSA，并且设置序列号的值比实际值更大。当目标路由器收到这个数据包后，就会发送更新的 LSA。当攻击者不断的发送类似的数据包后，就可能会导致和最大年龄攻击相似的结果。

3. 最大序列号攻击

攻击者向 OSPF 网络发送序列号为 0x7FFFFFFF 的 LSA 数据包，当其它路由器收到这个 LSA 时就会认为此 LSA 为最新的信息，然后更新链路状态数据库。一旦此 LSA 的真正发起路由器收到此恶意 LSA 后，就会发送一个修正 LSA，它的序列号为 0x80000001。不过很多 OSPF 协议实现都存在缺陷，路由器在发送正确的 LSA 之前不会冲掉此前恶意的 LSA，所以正确的 LSA 会被当作过期的 LSA 而丢弃。恶意的 LSA 会存活一个小时，利用协议实现的这个缺陷可以发起各种攻击。

4. 伪造 LSA 攻击

利用了 UNIX gated 守护进程实现上的缺陷，发送特定的 LSA 就会导致守护进程奔溃，这样就需要重新启动所有的进程来冲掉这个恶意的 LSA，因此这就造成了拒绝服务攻击。

2.4.2.2 E.Jones 攻击

E.Jones、O.Le Moigne 等提出了几种新奇的攻击方式[11]，下面将分别介绍这几种攻击方式。

1. 周期注入 LSA

攻击者周期性的注入恶意 LSA（每 5 秒 1 个包），这个攻击能使自反击机制无效。OSPF 标准规定不允许路由器在 MinLSInterval 时间间隔内发送同样 LSA 的两个实例（默认为 5 秒）。路由器处理完恶意的 LSA 后自反击机制才会被触发，这就意味着当受害的路由器每 5 秒内收到恶意的 LSA，它将不能发送自反击的 LSA。这个攻击的影响是持久的，但是需要很大的开销，攻击者必须以很高的速率泛洪恶意的 LSA。

2. 邻居表溢出攻击

攻击者产生大量的 Hello 包，这些 Hello 包包含了许多虚假的邻居标识。每一个这样的 Hello 包都会使受害路由器在邻居表中产生许多表项，通过使邻居表溢出，攻击者能保证受害路由器不能处理新的 Hello 包。

3. 链路状态数据库溢出攻击

攻击者利用恶意的 LSA 来压垮受害路由器，每一个 LSA 一般会保存在链路状态数据库中直到过期（需要 1 个小时）。通过使数据库溢出使受害路由器不能处理新的 LSA，导致路由器无法计算出正确的路由表。

4. 邻接中断攻击

攻击者发送恶意的 Hello 包，改变其中的指定路由器字段或者使其他路由器与指定路由器建立邻接关系，这两种情况下，网络中的路由器必须重新建立邻接关系，这需要消耗几十秒。在这段时间中，网络被宣告为残余网络，这会导致路由器不能正确的转发数据。

2.4.2.3 Gabi Nakibly 攻击

Gabi Nakibly、Alex Kirshon 等提出了几种对 OSPF 网络威胁很大的攻击方式[14]，下面将介绍这几种攻击方式。

1. 双 LSA 注入攻击

双 LSA 注入攻击可以改变其他路由器生成的 LSA，而且能逃避自反击机制。这种攻击对网络的威胁很大，它可以实现各种攻击，如拒绝服务攻击、路由欺骗等。具体实现方法见 3.2.1 节。

2. 远程虚假邻接

远程虚假邻接是指攻击者在远端与网络中的路由器建立虚假的邻接关系，具体的实现方法见 3.4.2 节，这种攻击方法容易实现流量黑洞。

3. LSA 覆盖攻击

这种攻击方法利用了 OSPF 协议实现的缺陷。目前 Cisco IOS 绝大多数版本在处理 LSA 头部时，没有对链路状态标识与宣告路由器的一致性进行检查，这个缺陷的后果很严重。假如攻击者发送一个其它路由器生成的 LSA，并且将它的内容篡改，LSA 头部的宣告路由器字段改为一个不存在的标识，链路状态标识不变。这个恶意的 LSA 到达真实产生该 LSA 的路由器时，协议进程发现宣告路由器不是自己，所以不会引起自反击机制，并且将该恶意的 LSA 存在链路状态数据库中，覆盖了真实的 LSA。这就导致了整个网络中的链路状态数据库都会被修改，包括产生该 LSA 的真实路由器。这个攻击比双 LSA 注入攻击的威胁更大。

2.5 本章小结

本章主要介绍了 OSPF 协议的原理以及运行机制，从区域、协议数据包、邻居协商过程这个三个方面进行分析；接着对 OSPF 协议的安全机制进行介绍，这些安全机制是保护协议安全的重要保障，尤其是自反击机制与密钥认证机制；最后对 OSPF 协议的安全性进行了分析，介绍了 OSPF 协议的安全威胁以及现有的主要攻击技术。

第3章 路由欺骗攻击

OSPF 协议有多种安全机制，其中自反击机制是最重要的机制，正是有了这个安全机制才使攻击者无法修改其他路由器生成的 LSA，保证了链路状态数据库的真实性和完整性，确保了路由表的正确性。本章提出了四种攻破自反击机制的攻击方法，这些方法利用了协议自身的弱点，所以适用于不同厂商的路由器。采用这些攻击方法可以恶意的篡改链路状态数据库，修改它们的路由表，实现路由欺骗。

3.1 邻接欺骗攻击

现实环境中很多 OSPF 网络的边界路由器接口并不会设置为被动接口，这就给网络的安全带来了隐患。这里针对这种场景，提出了一种邻接欺骗攻击，利用该攻击可以实现多种攻击效果。

3.1.1 攻击原理

在 OSPF 网络中，设置为被动接口的路由器接口不会发送和接收协议数据包，它一般用在末节网络中，主要为了节约资源。然而当边界路由器接口未设成被动接口时，攻击者可以利用邻接欺骗攻击对 OSPF 网络进行路由欺骗攻击。

邻接欺骗攻击主要分成两个步骤，分别为系统接入以及配置注入：

1. 系统接入是指攻击者使攻击主机与网络中邻居路由器建立邻接关系。当边界路由器未设为被动接口时，它会每隔一定时间（默认为 10 秒）发出 Hello 包，攻击者收到 Hello 包后就获得了邻居路由器运行的参数。影响邻接关系建立的参数主要有区域标识、验证类型和数据、发送时间间隔、死亡时间间隔等，攻击者调整本机的参数使它们的运行参数一致，接着攻击主机就会与邻居路由器建立邻接关系。
2. 在系统接入之后，攻击者就可以进行配置注入。配置注入是指攻击者配置 OSPF 协议，配置的改变会触发生成相应恶意的 LSA，这些 LSA 被其它路由器接收后，会导致它们路由表发生相应的改变。配置注入通过改变自身生成的 LSA 来欺骗其他路由器，这时 OSPF 网络中所有路由器的链路状态数据库都相同，因此不会存在自反击的现象。

3.1.2 攻击应用分析

邻接欺骗攻击是路由欺骗攻击的主要方式，利用这种攻击可以实现多种效果，比如密码嗅探、网络欺骗、中间人攻击、DNS 欺骗、拒绝服务等等。为了实现这些效果，攻击者可以在配置注入阶段注入网络路由或 DNS 路由（特殊的网络路由）。

网络路由注入是指攻击者注入一些恶意的网络路由，比如 126 邮箱登录界面的 IP 地址 121.195.178.58、建行网上银行登录界面的 IP 地址 106.120.109.130 等等，注入后网络中所有去往这些目标的流量将会受到影响。

DNS 路由注入是指攻击者注入恶意的 DNS 服务器路由，这些路由会篡改路由器的路由表，这样所有的 DNS 请求就会发送到恶意的 DNS 服务器中。

下面利用网络路由注入和 DNS 路由注入实现网络欺骗、邮箱密码嗅探、中间人攻击以及 DNS 欺骗。DNS 路由注入相比网络路由注入可以实现更多的效果，这里前三种效果采用网络路由注入，DNS 欺骗采用 DNS 路由注入。

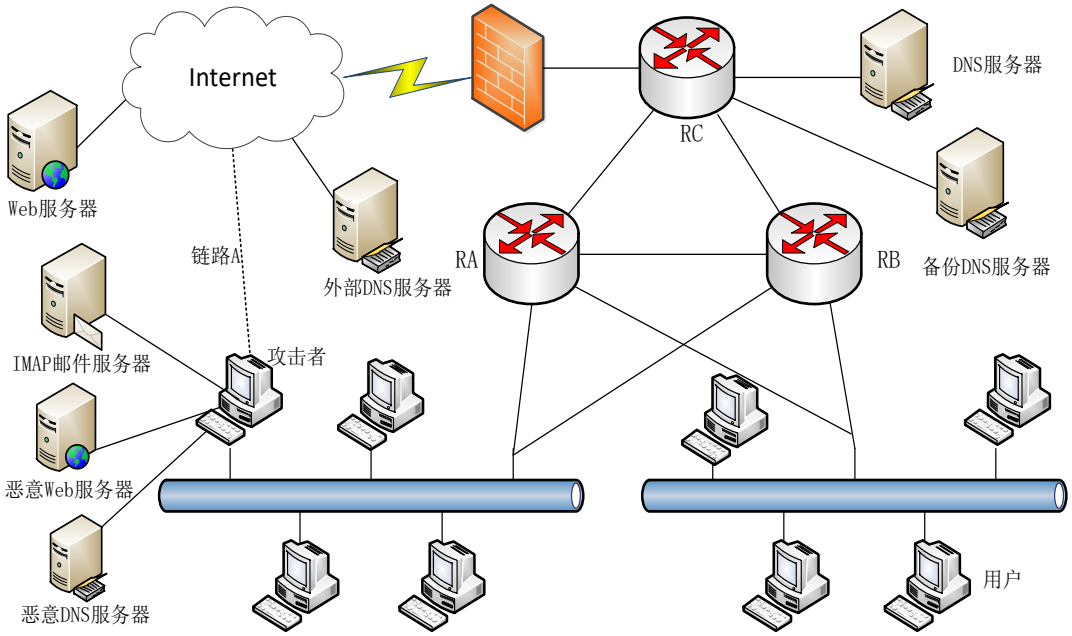

图 3-1 网络攻击示意图

3.1.2.1 网络欺骗

如图 3-1 所示，假设这个内部网络运行了 OSPF 协议，攻击者在本地注入一条网上银行登录界面的路由，这时所有的路由器中都会增加这条路由，并且这条路由的最终目的地为恶意 Web 服务器（假设恶意 Web 服务器的接口地址设为网上银行登录界面的 IP 地址）。攻击者可以制作虚假的网上银行登录界面，这时当用户登录网上银行时，他的用户名和密码就会被发送到恶意 Web 服务器中，被攻击者获得。这种攻击不仅实现了网络欺骗，而且网页经过特殊设计后，还可以实现钓鱼攻击。此外，攻击者可以注入不同的网络路由进行欺骗，从而得到不同的信息。

3.1.2.2 邮箱密码嗅探

现在很多人使用邮件客户端（如 Outlook、Foxmail 等）进行邮件的收发，而一般情况下，邮件客户端都是采用明文进行身份的验证。图 3-2 显示了邮件客户端与 IMAP 邮

件服务器的交互过程[27]，可以看到当邮件客户端与邮件服务器建立好 TCP 连接之后，双方进入认证阶段，此时邮件客户端会发送账号以及密码，认证通过后会进行邮件的接收。从上述过程可以看到，攻击者只需要与邮件客户端完成 TCP 三次握手，就可以收到账号和密码。

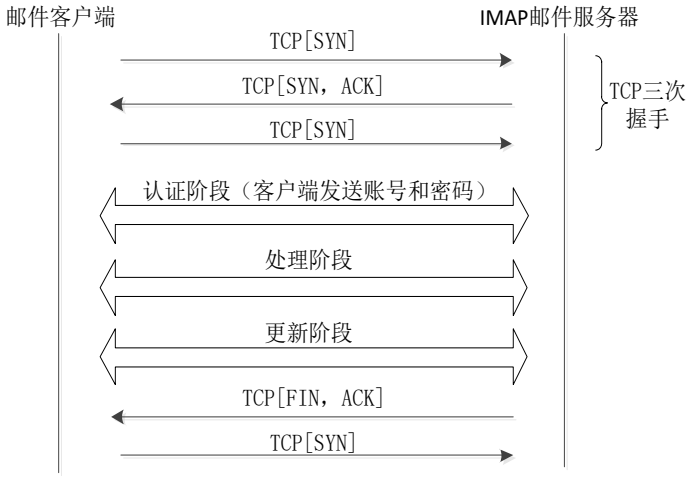

图 3-2 邮件客户端与 IMAP 服务器间的交互过程

攻击者将本地的 IMAP 邮件服务器的 IP 地址设为某个真实邮件服务器的地址，当用户利用邮件客户端接收邮件时，就会将邮箱的账号以及密码发往恶意的邮件服务器中，这时攻击者就可以嗅探到这些账号以及密码。

3.1.2.3 中间人攻击

中间人攻击是一种间接攻击的模式，它通过各种技术手段将攻击者控制的计算机虚拟的放置在网络连接中两台通信计算机之间，从而可以监听到它们之间的交互数据[28]。

上面介绍的几种攻击中，用户发送的数据都没有到达正确的目标。假如攻击者拥有另一条通往 Internet 的链路，那么他就可以实现中间人攻击。首先，攻击者注入恶意的网络路由，这样内网用户的部分数据会发往攻击者，接着攻击者将数据通过另一条链路发往正确的目标。由于攻击者只是针对某些网段进行中间人攻击，比如说只注入邮箱登录界面的 IP 地址，所以它的负担不会很大，用户也得到了相应的服务。

3.1.2.4 DNS 欺骗

攻击者注入一条恶意 DNS 服务器的路由后，内网中所有 DNS 请求都会发送到恶意 DNS 服务器中，攻击者可以对 DNS 服务器进行恶意的配置，实现 DNS 欺骗。利用 DNS 欺骗可以对网络造成多种影响，比如攻击者可以将网上银行登录界面的域名配置为恶意 Web 服务器的 IP 地址，这时用户就会把数据发送到恶意服务器中，攻击者也可以把域名配置为 Internet 上其它服务器的地址，这样用户就会把数据发送到外部的服务器等等。为了不影响对内网用户提供的域名解析服务，可以将在本地无法解析的域名请求，转发

到内网中的备份 DNS 服务器（如果存在），或者可以发往外部的 DNS 服务器，这样攻击就不会对网络产生很大的影响，从而不容易被察觉。

综上所述，网络路由注入的影响主要集中在自治系统内，而 DNS 路由注入相比于网络路由注入威胁更大，它的影响不局限于自治系统内，对 Internet 上的服务器也会产生影响。它们对于网络的正常运行影响不大，所以不容易被察觉。此外，攻击者可以控制网络路由存活的时间，一旦攻击者完成自己的目的后，只需要关闭运行的 OSPF 协议，几十秒后注入的网络路由就会从所有的路由器中消失。

3.2 双 LSA 远程多注入攻击

邻接欺骗攻击改变的是自身的 LSA，所以它不会引起自反击机制。然而当改变其它路由器的 LSA 时就会引发自反击机制，一直以来改变其它路由器的 LSA 都很难被攻破，Gabi Nakibly 提出了一种攻破自反击机制的方法[14]，并且可以改变其它路由器的 LSA。下面首先分析这种方法的原理，接着对该方法的缺点进行分析，并提出改进的方法。

3.2.1 Nakibly 双 LSA 注入攻击

根据第二章 OSPF 协议判断 LSA 新旧的规则可知，当两个 LSA 的序列号以及校验和相同，老化时间差小于 15 分钟时，路由器就会认为两个 LSA 相同。基于这一弱点，可以逃避自反击机制。

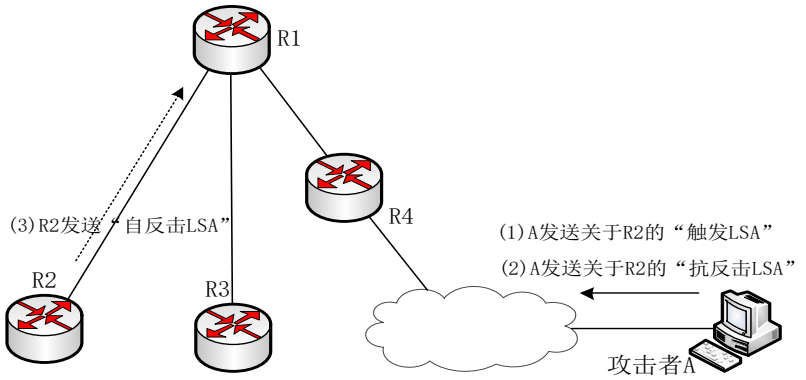

图 3-3 抗自反击机制示意图

首先定义 3 个重要的 LSA 数据包，图 3-3 中攻击者 A 发送关于 R2 的第一个恶意的 LSA，称为“触发 LSA”，它的序列号一般比当前 LSA 的序列号大。A 发送关于 R2 的精心构造的 LSA，称为“抗反击 LSA”，它与“自反击 LSA”具有相同的序列号和校验和，老化时间差小于 15 分钟，但是内容一般会被恶意篡改。R2 因自反击机制被触发后发送的 LSA，称为“自反击 LSA”，通常这个 LSA 的序列号会比触发 LSA 的序列号大 1。这里“抗反击”表示的是当 R2 收到抗反击 LSA 时不会触发自反击机制。如果 A 直接发送关于 R2 的当前序号的抗反击 LSA，那么当其它路由器收这个 LSA 后，会认为与

自己链路状态数据库中的 LSA 是同一个版本，而直接丢弃。虽然这没有触发自反击机制，但是其它路由器的链路状态数据库并没有改变。下面对这个方法进行改进，先触发 R2 发出一个新序号 LSA，然后再发送新序号的抗反击 LSA，并尽量使抗反击 LSA 先于反击 LSA 到达其余路由器。具体的攻击过程如下：

1) 攻击者 A 首先发送关于 R2 的触发 LSA，这个 LSA 的序列号要大于当前 R2 真实的 LSA 序列号，否则 R4 会向攻击者 A 发送自己所保存的关于 R2 的 LSA，而不会引起自反击机制。
2) 触发 LSA 发送后，立即发送抗反击 LSA。它与自反击 LSA 必须有相同的序列号、校验和，同时老化时间小于 15 分钟。这三个值都是可以预测的，下面将会讨论。
3) R2 发送自反击 LSA，这个 LSA 会被 R1 接收。由于 R1 已经收到了抗反击 LSA，它被认为与自反击 LSA 相同，所以不会更新自己的数据库以及泛洪出去。

抗反击 LSA 会被泛洪到整个区域中，当它到达 R2 时，由于它被认为是自反击 LSA 的一个副本，所以 R2 不会再次发出新的自反击 LSA，并且此时区域中所有路由器（除 R2）的链路状态数据库已经加载了抗自反击 LSA，所以它们的链路状态数据库会与真实的网络拓扑不同。需要注意的是，抗反击 LSA 与自反击 LSA 都会向区域中的路由器发送，先到达的 LSA 会被装入数据库中，后到的将被丢弃，这时就有一个时间竞争的问题。因为前者发送的更早，所以它有更大的概率被区域中的路由器接收。

路由器每隔 30 分钟（默认时间间隔）会更新自己的 LSA，所以为了维持攻击的效果，在 30 分钟到达之前要发送新的触发 LSA 和抗反击 LSA。

抗反击 LSA 中需要预先确定序列号、校验和以及老化时间这三个字段的值，其中序列号以及老化时间比较容易确定。自反击 LSA 序列号的值比触发 LSA 序列号的值大 1，而老化时间一般都是从 0 开始计时，所以抗反击 LSA 序列号的值为触发 LSA 序列号的值加 1，而老化时间可以设为 0。链路状态校验和为 fletcher 校验和，计算时包含除了 LSA 时域外的整个 LSA，所以自反击 LSA 校验和的值也可以提前确定。下面以路由器 LSA 为例来分析，当抗反击 LSA 与反击 LSA 的内容不同时，如何保证两者具有相同的校验和。从图 3-4 中可以知道校验和的长度为 16 位，而整个 LSA 的长度至少 288 位，所以校验和与整个 LSA 不是一对一的关系，有许多的值对应同一个校验和。通常抗反击 LSA 与反击 LSA 的内容不同，为了保证两者的校验和相同，需要在抗反击 LSA 后增加一个字段，称为“矫正字段”，这里矫正字段为链路标识、链路数据、类型、TOS 以及距离值的组合。由于反击 LSA 的校验和已知，可以对矫正字段的值一一列举并求校验和，直到满足校验和相同，这个穷举的过程所需要的时间很短。通过以上分析，可以知道抗反击 LSA 的构造比较容易。

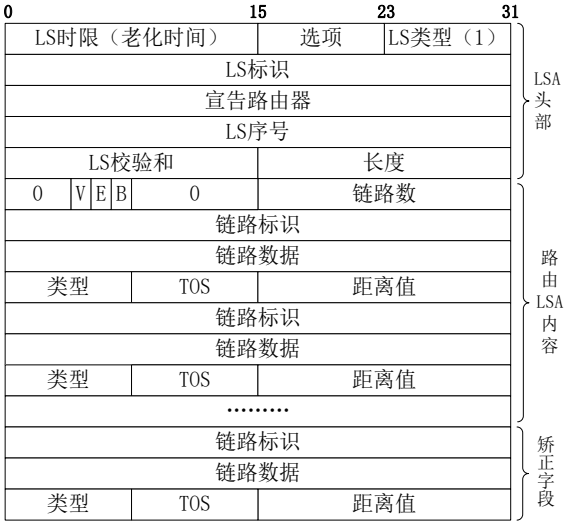

图 3-4 抗反击 LSA 结构图

3.2.2 双 LSA 远程多注入攻击

从双 LSA 注入的原理中可知抗反击 LSA 与自反击 LSA 有一个时间的竞争，先到达的 LSA 会被存放于数据链路状态数据库中，后到的会被抛弃，这是双 LSA 注入攻击方法最主要的不足。关于污染区域，Gabi Nakibly 只是提到攻击者发送完触发 LSA 后立即发送抗反击 LSA 比接收到自反击 LSA 后发送抗反击 LSA 更加有效，污染区域的范围更大[7]。这里首先分析影响污染区域的因素，最后对双 LSA 注入攻击进行改进，提出一种增大污染区域的方法。

3.2.2.1 污染区域分析

双LSA注入攻击中，影响污染区域的主要因素是LSA发送时间间隔以及网络拓扑。下面将对它们进行分析。

1. LSA 发送时间间隔

LSA接收间隔指的是对于特定的LSA，协议进程接收LSA新实例之间的时间间隔，以更高的频率到达的 LSA 实例将会被忽略，默认的时间间隔为 1 秒。所以在发送完触发 LSA 后，并不能立即发送抗反击 LSA，它们之间的时间间隔需大于 1 秒，这样抗反击 LSA 才会被自治系统中的路由器接收。

LSA 生成间隔指的是协议进程构造一个新 LSA，并发出的最小间隔。一般默认的时间间隔为 5 秒，通过配置命令“ip ospf retransmit-interval seconds”可以进行更改。当受害路由器收到触发 LSA 后，需要过 5 秒才能发送出自反击 LSA。当忽略数据库的更新和泛洪时，抗反击 LSA 与触发 LSA 的发送时间间隔只要在 1 秒到 5 秒之间，就可以使区域内的路由器被污染。如果考虑数据库的更新与泛洪时，发送间隔时间大于 5 秒时也有可能使部分路由器被污染，但是间隔时间越接近 1 秒，被污染的区域就越大。

2. 网络拓扑

污染区域与网络拓扑有着密切的联系，有一种拓扑很特殊，即使忽略数据库的更新和泛洪时间，并且双 LSA 注入攻击的时间间隔在 1 秒和 5 秒内，也无法对其中的路由器进行污染。图 3-3 中假如发送关于 R1 的触发 LSA 以及自反击 LSA，则污染的路由器只有 R4，而 R2 和 R3 这种只有通过 R1 才能与其它路由器相连的部分无法被污染。关于 R2 以及 R3 这种路由器的污染，可以使用改进的双 LSA 注入攻击来进行污染。除了 R2 和 R3 这种特殊路由器，其他的路由器都可以被污染。网络拓扑中只有那些距离攻击者很远，而距离受害路由器很近的路由器才可能不被污染，如图 3-5 中，如果 R2 与 R3 之间的网络很复杂，那么关于 R1 的抗反击 LSA 一般不会比自反击 LSA 先到达 R3，这时 R3 就没有被污染。

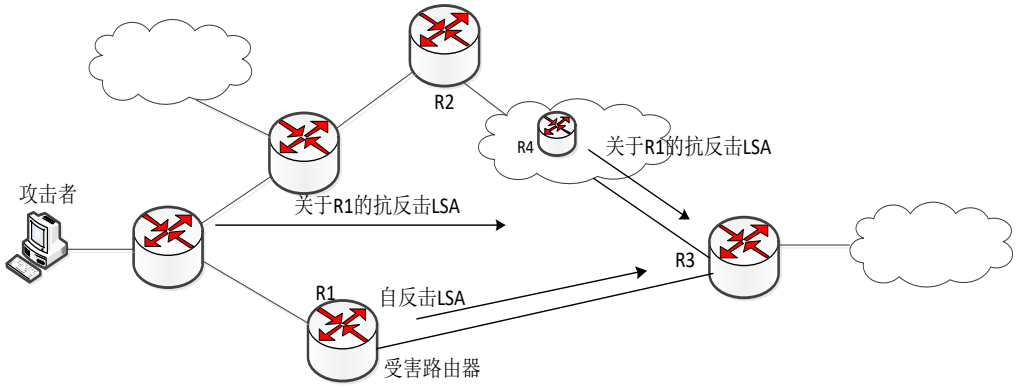

图 3-5 污染区域示意图

3.2.2.2 双 LSA 注入攻击的改进

针对图 3-3 与图 3-5 中部分路由器无法被污染的情况，这里提出了双 LSA 远程多注入攻击。双 LSA 远程多注入攻击是指攻击者注入触发 LSA 与抗反击 LSA 的位置不一定要在本地，可以将恶意 LSA 发出的位置分布在网络各个路由器中，这样可以同时发出多个双 LSA，从而增大了污染区域。

双 LSA 远程多注入攻击主要利用了源欺骗来实现远程注入。根据协议标准，路由器接收到协议数据包后会对其判断是不是从邻居发送过来，一般路由器主要根据自身的邻居数据结构表来进行验证。在邻居数据结构表中记录了邻居路由器的标识以及邻居接口的 IP 地址，这两个字段用来判断协议包来源的真实性。攻击者只需要对 IP 包的源地址以及 OSPF 包头中的路由器标识字段进行恶意设置，就能使路由器相信协议包是从邻居发送而来，从而实现了“源欺骗”。

现在利用改进的攻击方法对图 3-3 及图 3-5 中路由器进行污染。在图 3-3 中，攻击者可以将 OSPF 包的邻居标识设为 R1 的标识，IP 源地址设为 R1 相应的接口地址、目标地址分别设为 R2 和 R3 的接口地址、增大 TTL 值，这样触发 LSA 与抗反击 LSA 相当于从 R1 注入，这时 R2 和 R3 就会被污染。

图 3-5 中攻击者只需要将 OSPF 包中的邻居路由器标识设为 R4 的标识，IP 源地址设为 R4 的接口地址、目标地址设为 R3 的接口地址、增大 TTL 值，这样关于 R1 的触发 LSA 与抗反击 LSA 就可以从 R4 中发出，这时抗反击 LSA 将比自反击 LSA 先到达 R3。在这里只是伪装了一台路由器的身份发送恶意 LSA，在复杂的网络之中，可以伪装多台路由器的身份来发送多个恶意 LSA，这样污染的区域就会大大的增加。

3.2.3 攻击应用分析

双 LSA 远程多注入攻击与邻接欺骗攻击一样，它也可以通过注入网络路由或 DNS 路由实现路由欺骗。虽然它们的攻击效果类似，但是双 LSA 远程多注入攻击不要求攻击者与恶意服务器在相同的位置。

由于双 LSA 远程多注入攻击能改变其它路由器的 LSA，所以它可以对流量的中间传输路径进行控制，如图 3-6 攻击者可以控制流量从下面的路径传输。此外，它还能对网络的拓扑进行很大的改变，如 Ra 看到的网络拓扑与实际的拓扑相差很大。

双 LSA 远程多注入攻击适合于攻击者获得了网络路由器拓扑及参数的场景。当然，如果攻击者与网络中的路由器建立了邻接关系，那么他可以通过链路状态数据库获得网络路由器拓扑及参数。

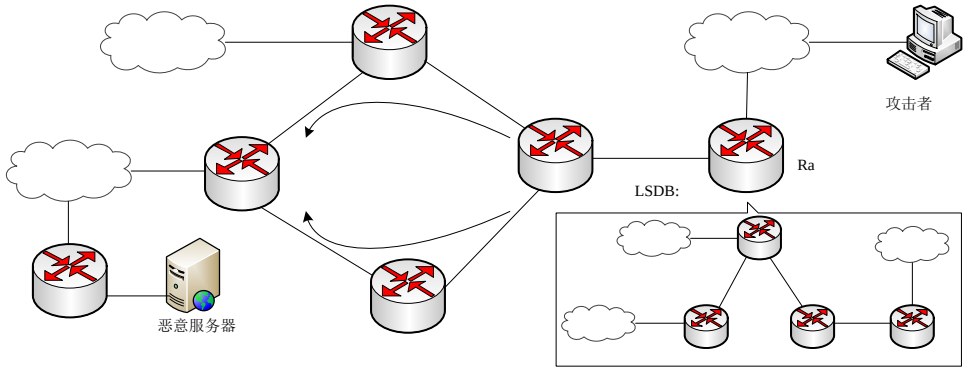

图 3-6 链路状态数据库污染示意图

3.3 单路径注入攻击

3.3.1 攻击原理

Gabi Nakibly 提出利用触发 LSA 以及抗反击 LSA 两个协议包来逃避自反击机制，这里提出了一种更加简单的攻击方法，只需要发送一个恶意的 LSA 就可以改变其它路由器的 LSA，而且不会引发自反击机制。

如图 3-7 所示，利用源欺骗的方法，攻击者可以冒充 R6 的身份发送恶意的 LSA 给 R8，为了使 R8 相信协议包是从 R6 发送过来，需要将包的源地址设为 f1/0 的地址，OSPF 包头的路由器标识设为 R6 的标识。伪造的协议包要能发送到 R8，需要将 IP 包中目的

地址设置为 R8 接口的地址，同时需要将 IP 包中的 TTL 字段设为较大的值，以满足到达 R8 的生存时间需要。这里把 R6 这样的路由器称为“跳板路由器”，因为攻击者可以利用它来发送关于其他路由器的 LSA（包括 R6 的 LSA），而把 R8 这样的路由器称为“源污染路由器”，因为它是第一个在自己链路状态数据库中存放恶意 LSA 的路由器，并且它将会把这些恶意的 LSA 继续泛洪出去，使更多的路由器被污染。R8 在收到恶意的 LSA 后，会发回一个 LSAck 包，虽然 LSAck 中含有的恶意 LSA 与 R6 中含有的真实 LSA 不一样，但是这不会引起自反击机制，这主要利用了路由器在处理 LSAck 包时的弱点，下面将描述路由器如何处理 LSAck 包以及为何不会引起自反击机制。

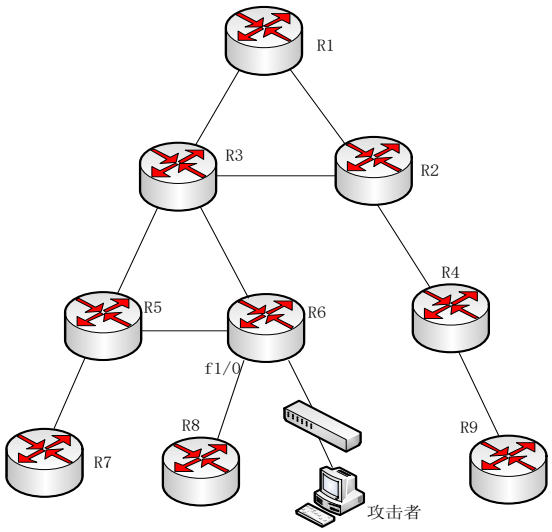

图 3-7 跳板路由器及源污染路由器定义示意图

OSPF 协议规定路由器在接收到 LSAck 包后会对包中所携带的 LSA 进行确认，如果发现某个 LSA 没有存在于链路状态重传列表中，则忽略这个 LSA，并继续确认下一个 LSA，如果发现它存在于链路状态重传列表中，则删掉列表中该 LSA。

图 3-8 中显示了 LSA 发送的三种情况，其中前两种显示了邻接建立阶段以及泛洪阶段的 LSA 交互过程，第三种则是发送恶意 LSA 的交互过程。从前两种正常的交互过程中，可以看到路由器发出 LSA 后，会在本地链路状态重传列表中会加入相应的 LSA，等待 LSAck 包，若一定时间内未收到回复，则会重发。当收到 LSAck 包后，会确认此 LSA，并将其从链路状态重传列表中删除，这样就表示链路状态数据库同步完成。在发送恶意 LSA 的攻击阶段，由于 LSA 是攻击者发出，而不是协议进程发出，所以不会在链路状态重传列表中增加 LSA，当收到 LSAck 包后，根据上面协议处理 LSAck 包的过程，它会直接丢弃，所以此时链路状态数据库并没有同步。利用这个弱点，虽然路由器收到的恶意 LSA 与本地链路状态数据库中真实的 LSA 不相同，也不会触发自反击机制，因为此时协议进程根本就不会检查 LSA，从而实现了对其它路由器链路状态数据库的污染。

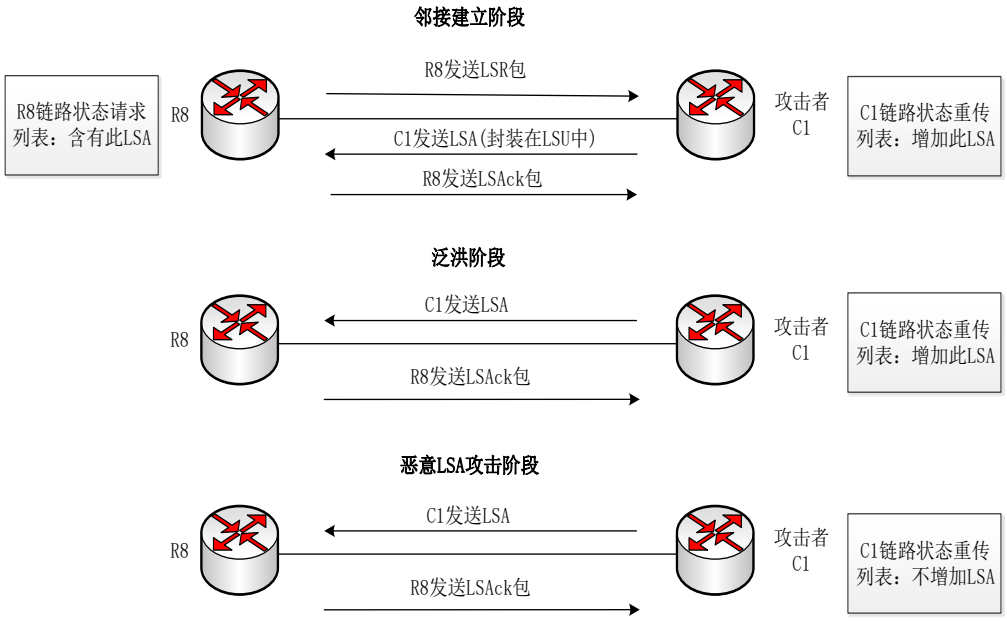

图 3-8 几种 LSA 发送的不同情况

不过这种方法也有局限性，并不是所有的情况都不会引起自反击机制，只有跳板路由器与源污染路由器之间满足“单路径”的条件才不会引发自反击机制。“单路径”是指跳板路由器与源污染路由器之间不存在任何环路，它们之间的路径只有一条。根据这个概念以及跳板路由器与源污染路由器之间必须是邻居关系，可以判断出图 3-7 中 R7 与 R5、R8 与 R6、R9 与 R4、R4 与 R2 之满足单路径条件，接着需要从单路径路由器对中选择哪台路由器是跳板路由器以及源污染路由器。跳板路由器相较于源污染路由器离攻击者经过的路由器个数更少，根据这个原则可以对以上的路由器对进行选择。例如 R5 与 R7 中，R7 与攻击者间相距两台路由器，而 R5 与攻击者间相距一台路由器，所以 R5 会被选择为跳板路由器，而 R7 会被选择为源污染路由器。对于像 R1 与 R3、R3 与 R5、R5 与 R6 等路由器对，它们不满足单路径条件（例如 R1 可以直接到 R3，也可以经过 R1 到 R3），无法作为单路径路由器对。

攻击者在选择完跳板路由器与源污染路由器之后，就可以选择伪造哪个路由器生成的 LSA。由于跳板路由器与源污染路由器之间单路径的特性，可以根据它们把整个拓扑网络划分为两个区域。跳板路由器各接口（除了与源污染路由器相连接口）上所相连的网络称为“被伪造区”，这个区域也包含了跳板路由器，攻击者可以从中任意选择路由器，来伪造它们生成的 LSA。而跳板路由器与源污染路由器相连接口所连接的网络称为“污染区”，因为每当攻击者完成伪造攻击后，这个区域内的路由器都会被污染。以图 3-7 为例，选择 R2 为跳板路由器，R4 为源污染路由器，那么 R4 与 R9 组成了污染区，而 R2、R1、R3、R5、R6、R7、R8、R9 组成了被伪造区。

根据以上描述可以把攻击者发起一次攻击分为以下几个步骤：

1) 从网络拓扑中选择满足单路径的路由器对。

2) 根据与攻击者之间经过的路由器个数，从路由器对中选择跳板路由器与源污染路由器。
3) 根据跳板路由器与源污染路由器，将网络拓扑划分为被伪造区以及污染区
4) 从被伪造区的路由器中选择要伪造的对象
5) 攻击者冒充跳板路由器，并发送伪造对象生成的 LSA。

3.3.2 攻击应用分析

单路径注入攻击可以选择网络中特定的路由器进行路由欺骗，它不像双 LSA 远程多注入攻击的影响是全局的，它可以针对小范围进行欺骗攻击。例如图 3-7，攻击者可以只针对 R7、R8 或 R9 进行路由欺骗，比较容易实现流量黑洞。

与双 LSA 远程多注入攻击的适用场景一样，攻击者需要知道网络路由器拓扑以及参数。

3.4 远程邻接欺骗攻击

上面介绍的方法以及过去关于 OSPF 协议的攻击几乎都是基于攻击者的邻居路由器运行了 OSPF 协议且未设置为被动接口，或者攻击者获得了网络路由器拓扑及参数。当攻击者接入到 OSPF 网络，但是相邻接口没有运行 OSPF 协议或设为被动接口，这时攻击者就无法与邻居路由器建立邻接关系，并且对于非网络管理员一般很难获得网络路由器拓扑及参数。这里提出了一种即使攻击者无法满足这些条件时，仍然可能注入恶意 LSA 的远程攻击方法。

远程邻接欺骗攻击是指攻击者利用幻影路由器与网络中某些路由器建立虚假的邻接关系，然后以幻影路由器的身份注入恶意的 LSA。远程邻接欺骗攻击分为三个阶段，分别为协议探测阶段、远程邻接欺骗阶段以及恶意 LSA 注入阶段。

3.4.1 协议探测阶段

协议探测阶段主要用于判断哪些远程路由器接口运行了 OSPF 协议，并且确定协议运行的基本参数。根据 OSPF 协议规定，在广播网络上，Hello 包的目标地址始终为 224.0.0.5。传统的协议探测方法是攻击者在本地监听是否有 Hello 包，然后读取其中的相关参数。然而当相邻的路由器接口没有运行 OSPF 协议或设为被动接口时，它们不会发出 Hello 包，这时这种方法就不能判断网络中哪些接口运行了 OSPF 协议以及确定它们的基本参数。这里提出了一种探测远程路由器接口是否运行 OSPF 协议以及确定基本参数的方法，图 3-9 描述了该方法的流程。

收集远程路由器接口地址是攻击者必须完成的第一步，内部网络中的用户访问外部网络、内部网络的用户或服务器一般会经过几个路由器，而经过的路由器接口地址可以

用命令 tracert（windows 环境下）或 traceroute（linux 环境下）来获得。为了使搜集到的路由器接口地址更全，攻击者需要不断的尝试。

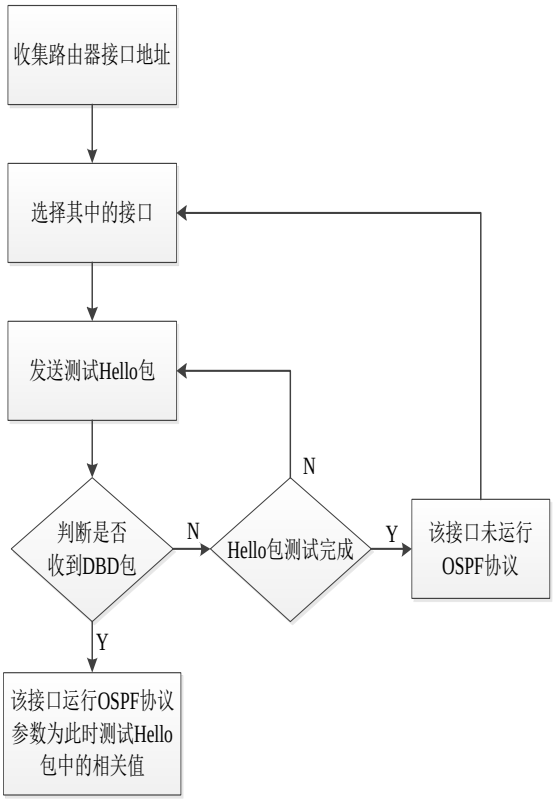

图 3-9 协议探测流程

为了判断远程路由器接口是否运行 OSPF 协议以及确定它的基本参数，攻击者需要不断的发送测试 Hello 包。如图 3-10 所示，只有当攻击者收到 DBD 包后，才能确定目标路由器运行了 OSPF 协议以及它运行的基本参数。

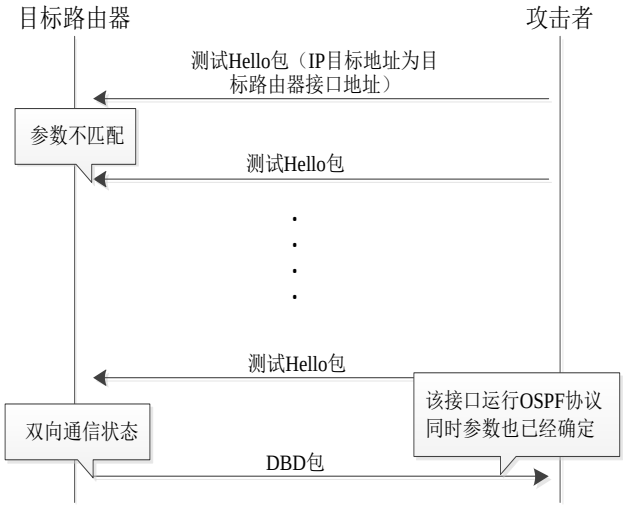

图 3-10 协议探测包发送流程

DBD 包只有在建立双向通信状态后才会发出。当目标路由器接收到 Hello 包后，它会比较 Hello 包中的参数与自己的运行参数是否一致，并且检查邻居字段是否包含自己的标识，一旦这些条件都满足后，它就会进入双向通信状态，同时发出 DBD 包。这时说明测试 Hello 包的参数与目标路由器的运行参数相同，攻击者就能确定目标路由器的基本参数。

目标路由器对 Hello 包中每个字段进行验证，任何字段不匹配都会导致无法进入双向通信状态。这些字段中部分字段是不确定的，比如区域标识、验证类型和数据、网络掩码、Hello 发送间隔、路由器死亡间隔、邻居标识。一般内部网络中会有骨干区域，所以区域标识基本上都会设为 0。验证类型和数据比较难以预测，需要进行穷举，可能需要花很长时间，但是为了网络的性能，很多网络管理员并不会配置认证。网络掩码取值的范围很小，例如当目标路由器的接口地址为 B 类地址，那么网络掩码的可能性也就 16 种。包发送间隔以及路由器死亡时间一般分别设置为 10 秒和 40 秒。路由器标识是一个 32 位的值，它一般选择接口 IP 地址值中最大的那个值，所以要收集路由器完整的接口地址。根据已知的目标路由器的接口地址，可以减小列举的范围，例如已知接口地址为 B 类 IP 地址，那么只需要对 B 类以及 C 类的部分地址范围进行穷举。

根据 OSPF 协议规定，与 Hello 包不同，在广播网络上，DBD 包的目标地址为单播地址，正是由于这个原因，攻击者可以判断远程路由器接口是否运行 OSPF 协议以及确定协议运行参数。

3.4.2 远程邻接欺骗阶段

协议探测阶段发现某一接口运行 OSPF 协议以及确定运行参数后，攻击者就可以发送恶意的协议包，以幻影路由器身份与之建立邻接关系。Gabi Nakibly 曾经提出了一种与远程路由器建立虚假邻接关系的方法，这里使用这种方法来实现远程邻接的欺骗。

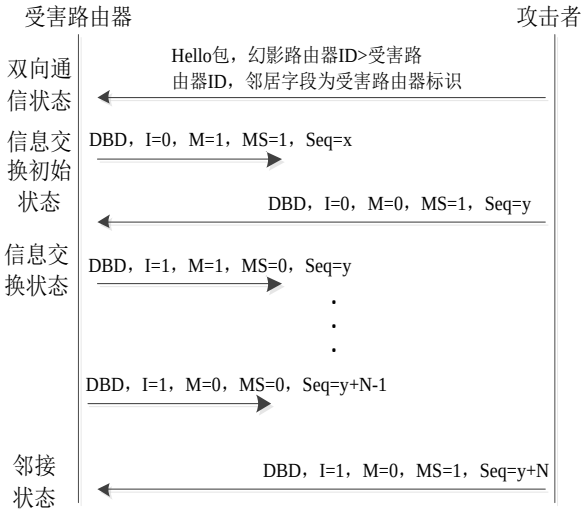

图 3-11 远程邻接欺骗示意图

攻击者以一个不存在的幻影路由器在受害路由器所在的网段上与它建立虚假的邻接，图 3-11 描述了整个远程邻接欺骗的过程。首先攻击者发送一个恶意的 Hello 包，邻居字段设为受害路由器标识，并且幻影路由器标识的值大于受害路由器标识的值。当受害路由器接收到这个 Hello 包，就进入了双向通信状态。接着受害路由器发送 DBD 包进行主从关系的选举，攻击者收到这个包后，发送 DBD 包并宣称幻影路由器为主路由器，由于幻影路由器标识大于受害路由器标识，所以受害路由器为从路由器。接下来受害路由器会不断的发送 DBD 包，而攻击者每次只需要将回复序号加一的 DBD 包，当受害路由器全部发送完 DBD 包，这时双方就进入了邻接的状态。这个邻接建立过程中没有信息加载过程，因为攻击者发送的 DBD 包都是空的，并没有携带 LSA。虽然受害路由器发送了 DBD 包，但是攻击者并没有发送 LSR 包来请求 LSA，所以当 DBD 包都发送完后，幻影路由器以及受害路由器都进入了邻接状态。

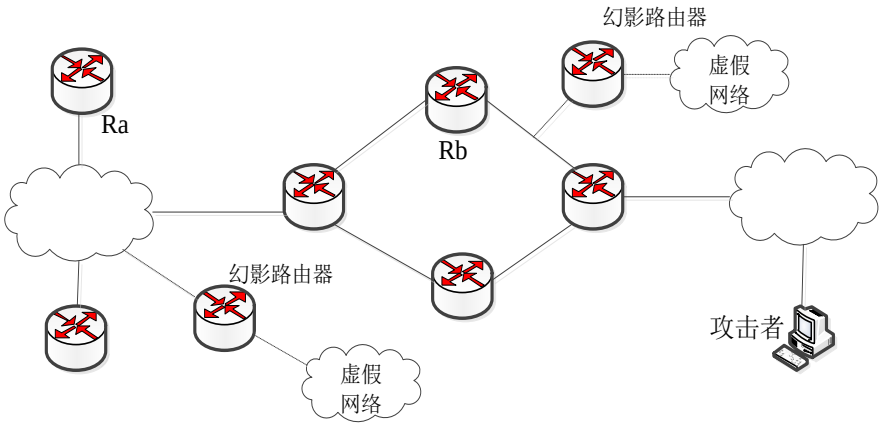

图 3-12 远程邻接欺骗攻击示意图

利用上述远程邻接欺骗方法，攻击者可以与网络中的多台路由器建立远程邻接关系，如图 3-12，攻击者利用幻影路由器与 Ra、Rb 建立了远程邻接关系。

3.4.3 恶意 LSA 注入阶段

当虚假的远程邻接关系建立之后，攻击者就可以以幻影路由器的身份注入 LSA，这些 LSA 会被发送区域中的路由器中来改变它们的路由表，实现路由欺骗。由于幻影路由器实际上并不存在，所以这些 LSA 不会引发幻影路由器的自反击机制。

3.4.4 攻击应用分析

如图 3-12 所示，攻击者可以在网络上多个位置添加幻影路由器以及虚假的网络，任何去往该网络的流量都会被转发到幻影路由器，由于幻影路由器与网络实际上并不存在，它们只是为了将用户的流量都欺骗过来，这样就形成了流量黑洞。此外，由于很多的网络流量会转发到幻影路由器中，这样可能会造成前一跳的路由器处理负担或链路带宽消耗增大。

3.5 仿真测试

3.5.1 仿真环境搭建

1. GNS3 网络仿真软件

GNS3 是一款非常优秀的图形网络仿真软件，可以应用于不同的平台之中，如 Windows、Linux、MacOS 等等[29]。相比于 Packet tracer、Boson NetSim 等仿真软件，它的功能更加强大。下面列举了它的一些特点：

1) 网络拓扑的搭建以及拓扑的动态改变非常方便，可扩展性强。
2) 提供了各种网络设备，如各种系列的 Cisco 路由器、交换机、防火墙等等。
3) 设备可以运行不同的镜像文件，如 Cisco 路由器可以运行不同版本的 IOS 系统。
4) 网络设备提供可视化的终端接口，便于设备的配置以及查看设备内部数据。
5) 整个网络环境运行的协议是真实的，而不像 Boson NetSim 那样是基于命令行的模拟。
6) 结合了 Wireshark 网络分析软件，可以用它来查看接口上包的具体信息。
7) 模拟的网络环境可以与外部真实的网络设备进行桥接，从来将内部网络与外部网络连接在一起，形成一个整体。

GNS3 中 Cisco 路由器在使用之前需要配置系统软件，默认情况下所有仿真实验使用的 Cisco IOS 版本都是 c3640。

2. 网络模拟平台

利用 GNS3 网络仿真软件、VMware 虚拟机以及真实物理计算机搭建一个高仿真程序的网络模拟平台[30]。GNS3 与虚拟机、计算机通过桥接的方式相连，这样仿真网络就能与外部网络相互通信。

3. 仿真拓扑结构

路由欺骗仿真拓扑图如图 3-13 所示，整个 OSPF 网络分为了三个区域，分别为骨干区域 Area0、常规区域 Area1 和 Area2。网络中所有网段均标注在路由器链路旁，而路由器接口 IP 地址为网络号加上路由器的标号，比如 R9 的两个接口地址为 30.129.21.9 与 30.129.22.9。路由器标识基于路由器的标号，比如 R4 的路由器标识为 4.4.4.4。用户 C2 与 C3 是两台内部主机，它们通过路由器关闭路由功能来模拟。

攻击者 A2 是区域 2 中一台桥接的真实物理计算机，它的 IP 地址为 192.168.80.200。R3 是与 A2 相邻的网络路由器，接口 f1/0 运行了 OSPF 协议，但是设置为被动接口，它与 A2 未建立邻接关系。

攻击者 A1 是区域 1 网段 10.129.23.0/24 中的一台主机，IP 地址为 10.129.23.100，标识为 100.100.100.100。A1 是外部的虚拟机，它通过桥接的方式连接到仿真网络之中。

R10 是与 A1 相邻的网络路由器，接口 f3/0 运行了 OSPF 协议，并且未设置为被动接口，它与 A1 建立了邻接关系。

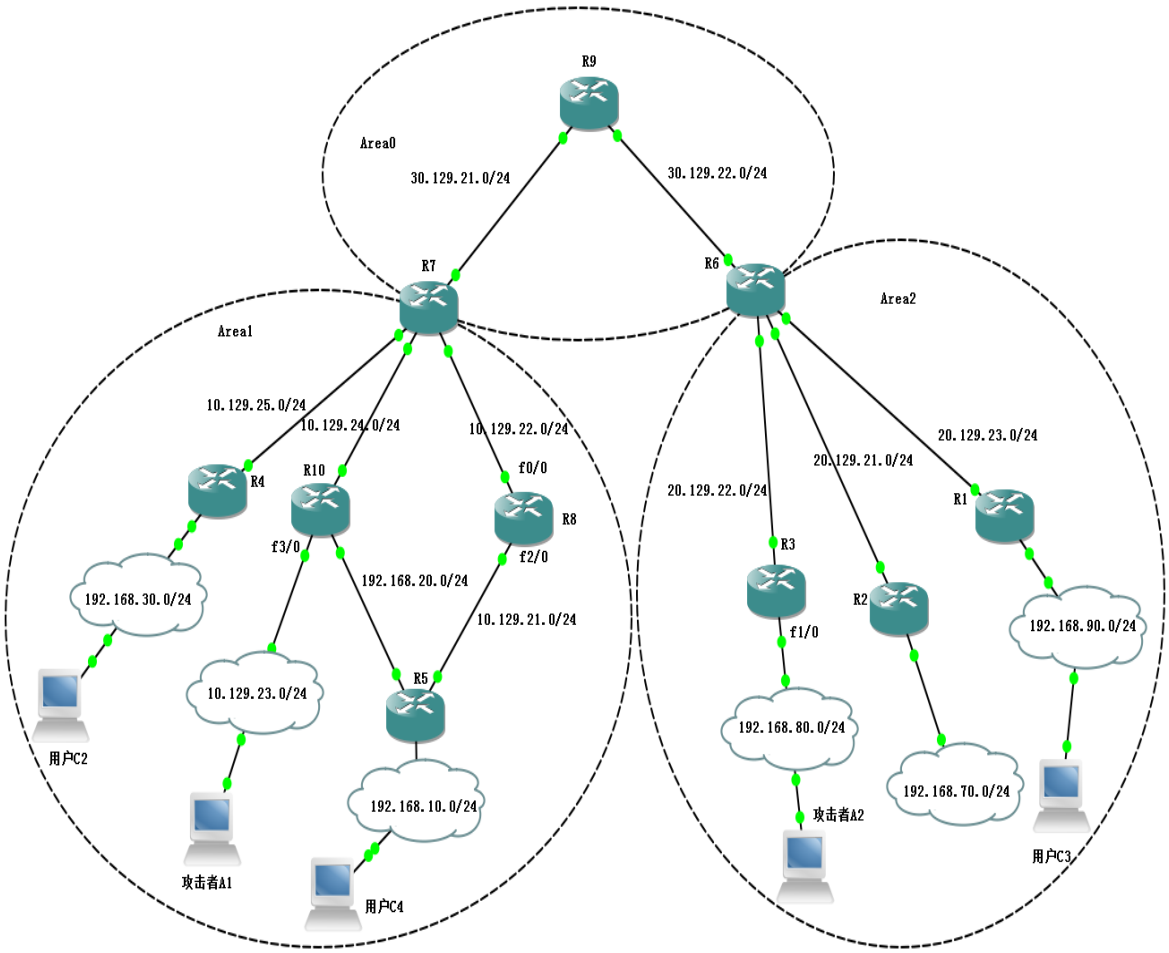

图 3-13 路由欺骗仿真拓扑图

3.5.2 邻接欺骗攻击

1. 欺骗攻击场景

R10 接口 f3/0 运行了 OSPF 协议，且未设为被动接口。这时攻击者 A1 可以与 R10 建立邻接关系，通过向自治系统注入了一条新的 121.195.178.0/24 路由，使得去往该网段的流量全部被欺骗到攻击者这里。

2. 欺骗攻击实现

邻接欺骗攻击分为系统接入与配置注入两步。由于 R10 的协议运行参数都采用默认值，攻击者 A1 只需配置命令 network 10.129.23.0/24 area 0.0.0.1，就可以与 R10 建立邻接关系。系统接入之后，攻击者可以以内部路由器身份或 ASBR 身份注入恶意的 LSA，这里以 ASBR 身份进行实现。图 3-14 为注入时的配置[31]，由于主机 A1 只有一个接口 eth1，这里通过配置虚拟接口 eth1:2 来增加接口。在 zebra 进程中设置一条或多条静态路由，然后在 OSPF 进程里重发布这些静态路由。

```
root@ubuntu:~# ifconfig eth1:2 15.15.15.3 netmask 255.255.255.0 up
root@ubuntu:~#
zebra# conf t
zebra(config)# ip route 121.195.178.0/24 15.15.15.3   配置静态路由
zebra(config)#
ospfd# conf t
ospfd(config)# router ospf
ospfd(config-router)# redistribute static          重发布静态路由
ospfd(config-router)# end
ospfd#
```

图 3-14 以 ASBR 身份注入路由的配置

3. 欺骗攻击结果

当欺骗攻击完成后，整个自治系统中路由器的路由表都会被篡改，图 3-15 列举了区域 1 中 R10 以及区域 2 中 R2 的路由表，可以看到多了一条“O E2”的路由，“O E2”表示这条路由是从自治系统的外部学习到的。流量经过的路径可以使用 traceroute 命令来观察，它显示的是传输路径上经过的路由器接口地址。图 3-16 显示了用户 C3 到 121.195.178.1 的传输路径，对应的路由器依次为 R1、R6、R9、R7、R10，由于网络中并不存在 121.195.178.1 这台主机，所以它最终无法达到，但是所有的流量都已经转移到攻击者 A1。这里只是初步的使用户流量转移到攻击者这里，攻击者利用其他技术可以进一步的实现中间人攻击、密码嗅探、钓鱼攻击等等。

```
O IA 192.168.90.0/24 [110/5] via 10.129.24.7, 00
O    192.168.30.0/24 [110/3] via 10.129.24.7, 00
     20.0.0.0/24 is subnetted, 3 subnets
O IA    20.129.21.0 [110/4] via 10.129.24.7, 00:
O IA    20.129.23.0 [110/4] via 10.129.24.7, 00:
O IA    20.129.22.0 [110/4] via 10.129.24.7, 00:
O    192.168.10.0/24 [110/2] via 192.168.20.5, 0
O IA 192.168.80.0/24 [110/5] via 10.129.24.7, 00
C    192.168.20.0/24 is directly connected, Fast
     10.0.0.0/24 is subnetted, 5 subnets
O       10.129.25.0 [110/2] via 10.129.24.7, 00:
C       10.129.24.0 is directly connected, FastE
C       10.129.23.0 is directly connected, FastE
O       10.129.22.0 [110/2] via 10.129.24.7, 00:
O       10.129.21.0 [110/2] via 192.168.20.5, 00
     121.0.0.0/24 is subnetted, 1 subnets
O E2    121.195.178.0 [110/20] via 10.129.23.100
O IA 192.168.70.0/24 [110/5] via 10.129.24.7, 00
     30.0.0.0/24 is subnetted, 2 subnets
O IA    30.129.22.0 [110/3] via 10.129.24.7, 00:
O IA    30.129.21.0 [110/2] via 10.129.24.7, 00:
R10#
```

```
O    192.168.90.0/24 [110/3] via 20.129.21.6, 00:14:57, Fas
O IA 192.168.30.0/24 [110/5] via 20.129.21.6, 00:14:57, Fas
     20.0.0.0/24 is subnetted, 3 subnets
C       20.129.21.0 is directly connected, FastEthernet0/0
O       20.129.23.0 [110/2] via 20.129.21.6, 00:14:57, Fast
O       20.129.22.0 [110/2] via 20.129.21.6, 00:14:57, Fast
O IA 192.168.10.0/24 [110/6] via 20.129.21.6, 00:14:57, Fas
O    192.168.80.0/24 [110/3] via 20.129.21.6, 00:14:59, Fas
O IA 192.168.20.0/24 [110/5] via 20.129.21.6, 00:14:59, Fas
     10.0.0.0/24 is subnetted, 5 subnets
O IA    10.129.25.0 [110/4] via 20.129.21.6, 00:14:59, Fast
O IA    10.129.24.0 [110/4] via 20.129.21.6, 00:14:59, Fast
O IA    10.129.23.0 [110/5] via 20.129.21.6, 00:14:59, Fast
O IA    10.129.22.0 [110/4] via 20.129.21.6, 00:14:59, Fast
O IA    10.129.21.0 [110/5] via 20.129.21.6, 00:14:59, Fast
     121.0.0.0/24 is subnetted, 1 subnets
O E2    121.195.178.0 [110/20] via 20.129.21.6, 00:00:38, F
C    192.168.70.0/24 is directly connected, FastEthernet1/0
     30.0.0.0/24 is subnetted, 2 subnets
O IA    30.129.22.0 [110/2] via 20.129.21.6, 00:14:59, Fast
O IA    30.129.21.0 [110/3] via 20.129.21.6, 00:14:59, Fast
R2#
```

图 3-15 攻击后 R10 与 R2 的路由表

```
PC-C3#traceroute 121.195.178.1

Type escape sequence to abort.
Tracing the route to 121.195.178.1

  1 192.168.90.1 1020 msec 12 msec 8 msec
  2 20.129.23.6 20 msec 36 msec 20 msec
  3 30.129.22.9 20 msec 40 msec 24 msec
  4 30.129.21.7 68 msec 32 msec 40 msec
  5 10.129.24.10 80 msec 48 msec 40 msec
  6  *  *  *
```

图 3-16 用户 C3 到 121.195.178.1 的传输路径

3.5.3 双 LSA 远程多注入攻击

LSA 有多种不同的类型，针对路由器 LSA 攻击造成的影响更大，所以这里以伪造图 3-13 中 R8 的路由器 LSA 为例来讨论利用双 LSA 远程多注入攻击如何实现路由欺骗。

1. 欺骗攻击场景

与 3.5.2 的情形不同，这里将 R10 的接口 f3/0 设为被动接口，并假设攻击者 A1 获得了网络路由器拓扑及参数。正常情况下用户 C3 与用户 C4 的通信流量不一定经过 R10，也有可能经过 R8，利用双 LSA 远程多注入攻击使它们之间的传输路径都经过 R10。

图 3-17 显示了攻击前用户 C3 与用户 C4 之间流量传输的路径，可以看到用户 C3 到 C4 的路径为 C3、R1、R2、R6、R9、R7、R10、R5、C4，而用户 C4 到 C3 的路径为 C4、R5、R8、R7、R9、R6、R2、R1、C3，C4 到 C3 的路径没有经过 R10。默认情况下路由器接口的距离值都是为 1，现在通过改变 R8 的 f0/0 以及 f2/0 的接口距离值来改变 C3 与 C4 间的传输路径，确保它们的来回路径都经过 R10。

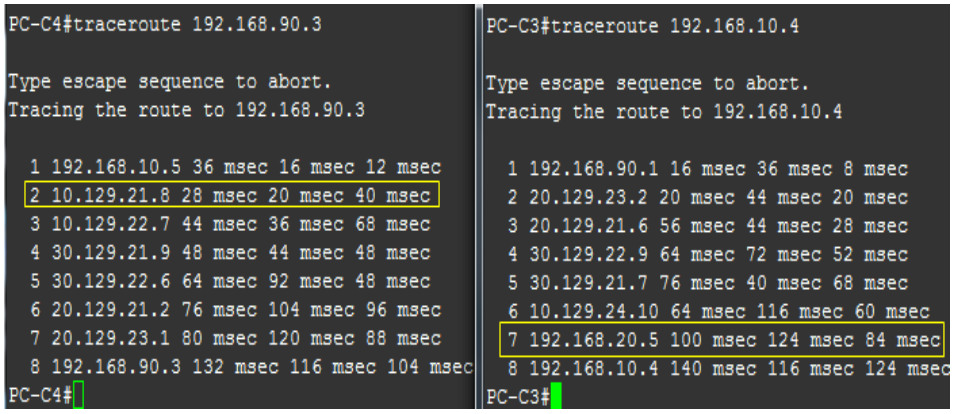

图 3-17 攻击前用户 C3、C4 间的传输路径

2. 欺骗攻击实现

为了保证 R5 和 R7 被污染，利用双 LSA 远程多注入攻击，从 R10 这里分别向 R5 和 R7 注入关于 R8 的触发 LSA 以及抗反击 LSA。两种情况的触发 LSA 及抗反击 LSA 参数类似，这里以向 R7 注入的情况为例。触发 LSA 需要满足序列号大于真实 LSA 的序列号，它的 IP 头部以及 OSPF 头部见表 3.1。抗反击 LSA 的参数设计需要用到矫正字段，它主要的参数设置如表 3.1 所示。

3. 欺骗攻击结果

攻击者 A1 注入关于 R8 的触发 LSA 以及抗反击 LSA 之后，区域 1 中大部分路由器（除 R8）保存的是抗反击 LSA，它们所看到的网络拓扑将与网络的实际拓扑不相同。图 3-18 显示了在 R5 以及 R8 上观察到的关于 R8 的路由器 LSA，可以看到 R5 链路状态数据库中存放的是抗反击 LSA，R8 接口 f0/0 的距离值为 32，接口 f2/0 的距离值为 16，而 LSA 头部却显示与真实的 LSA 是同一实例。

表 3.1 抗反击 LSA 包的主要参数

位置	字段	值及描述
IP 头	源地址	10.129.24.10（R10 的发送接口地址）
	目标地址	10.129.24.7
	协议号	89（OSPF 协议）
	TTL	10
OSPF 头部	类型	4（代表 LSU 包）
	路由器标识	10.10.10.10（生成此包的路由器标识）
	区域标识	0.0.0.1（区域号）
LSU 头部	LSA 数量	3（表示链路状态的数量）
路由器 LSA	链路标识	10.129.21.8（指定路由器的接口地址）
	链路数据	10.129.21.8（本地路由器的接口地址）
	链路类型/TOS	2/0
	距离值	16（R8 接口 f2/0 的伪造距离值）
	链路标识	10.129.22.8
	链路数据	10.129.22.8
	链路类型/TOS	2/0
	距离值	32（R8 接口 f0/0 的伪造距离值）
矫正字段	链路标识	0x08080808（随机设定的值，固定不变）
	链路数据	0x08080808（随机设定的值，固定不变）
	链路类型/TOS	2/0（随机设定的值，固定不变）
	距离值	0x6a27（通过穷举距离值来保持校验和的一致）

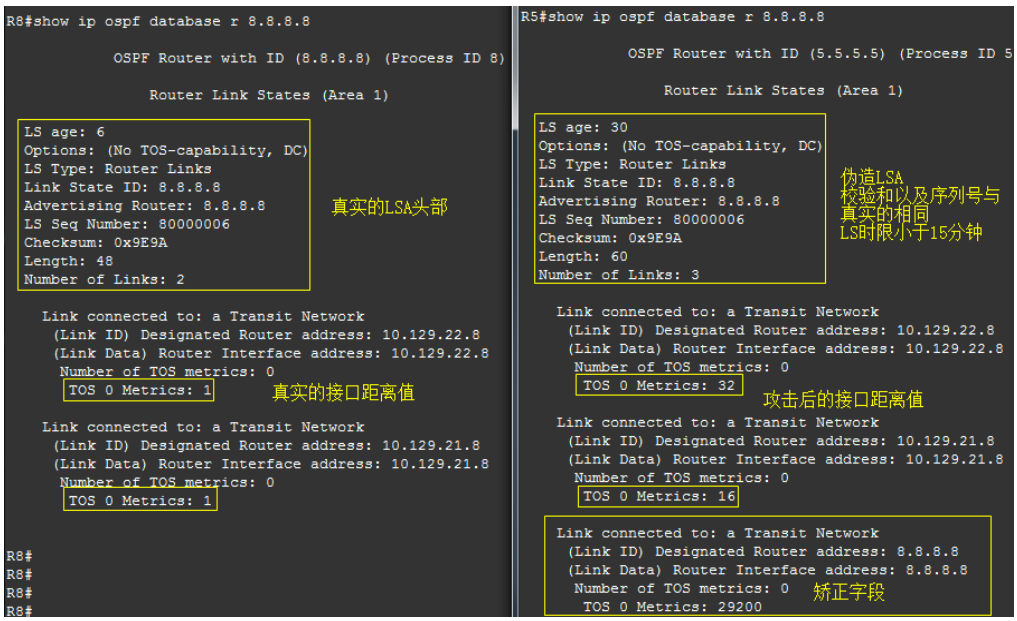

图 3-18 攻击后 R5 与 R8 上的 R8 路由器 LSA

图 3-19 显示了攻击前后路由器 R5 的路由表，可见在攻击后 R5 的路由表被篡改。现在去往 192.168.90.0/3 网络的流量将经过“距离更近”的 R10，不会经过 R8。图 3-20 显示了攻击后 C3 与 C4 之间流量传输的路径，可以看到它们都经过了 192.168.20.0/24 这个网段。通过这个实验表明，双 LSA 远程多注入攻击实现了路由欺骗，并且可以控制流量的中间传输路径。

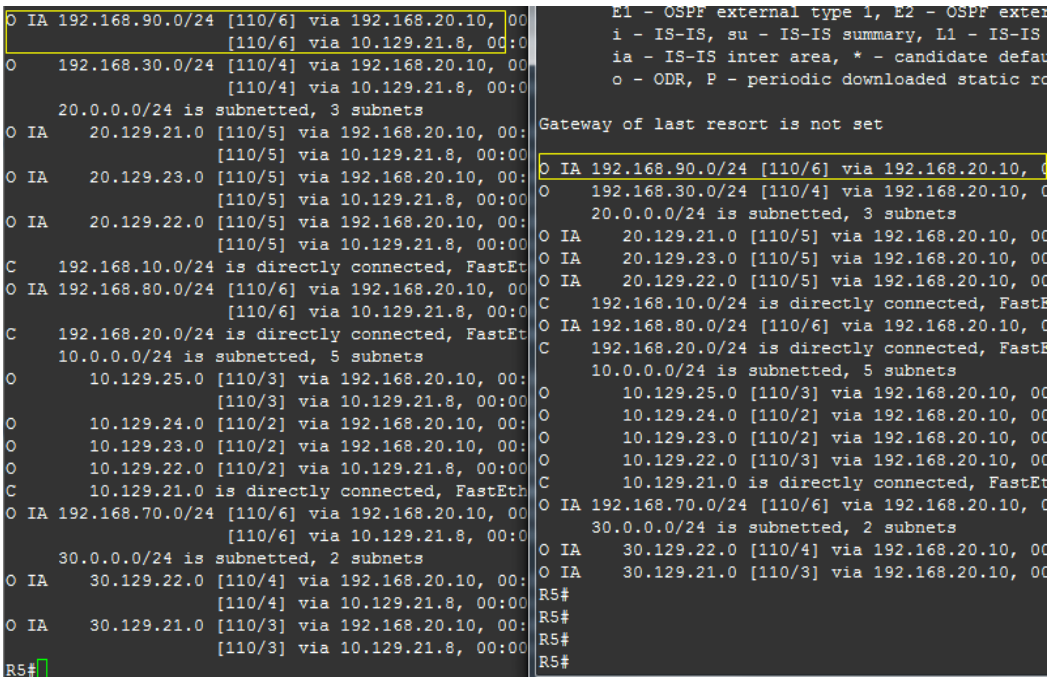

图 3-19 攻击前后 R5 的路由表

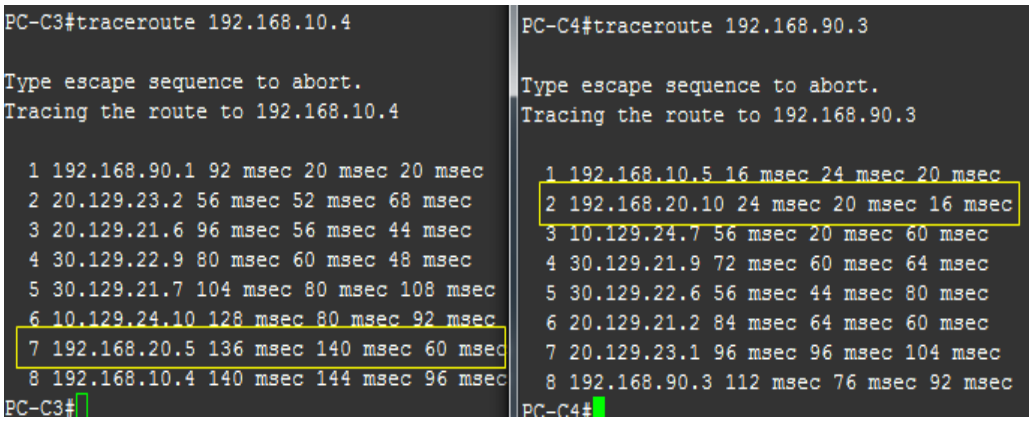

图 3-20 攻击后用户 C3、C4 间的传输路径

3.5.4 单路径注入攻击

1. 欺骗攻击场景

正常情况下用户 C3 到用户 C2 的流量不会经过 R3，假设攻击者 A2 获得了网络路由器拓扑及参数，利用单路径注入攻击使用户 C3 发送给 C2 的流量全部转移到 R3。

2. 欺骗攻击实现

根据上面描述的攻击步骤，首先选择单路径路由器对，选择图 3-13 中 R3 与 R6 为单路径对，由于 R3 比 R6 离攻击者 A2 更近，所以选择 R3 为跳板路由器，R6 为源污染路由器。这时网络拓扑分为两个部分，R3、攻击者 A2 组成了被伪造区，而区域 2 中其余路由器组成了污染区。为了使流量能够发送到 R3，攻击者 A2 可以选择 R3、攻击者 A2 自身作为伪造对象，这里选择 R3 作为伪造对象，并冒充跳板路由器 R3 的身份发送伪造的 R3 路由器 LSA 给 R6。表 3.2 描述了伪造协议包的主要参数，可以看到源地址为 20.129.22.3，路由器标识为 3.3.3.3，这就使 R6 相信这个协议包是 R3 生成并发出的。目标地址设为 20.129.22.6，TTL 设为 3 保证了数据包能发送到 R6。宣告路由器设为 3.3.3.3 表示这是 R3 生成的路由器 LSA。链路标识设为 192.168.30.0，链路数据设为 255.255.255.0，这是攻击者伪造的网络。

表 3.2 单恶意 LSA 的主要参数

位置	字段	值以及描述
IP 头	源地址	20.129.22.3（跳板路由器的接口地址）
	目标地址	20.129.22.6（源污染路由器的接口地址）
	协议号	89（OSPF 协议）
	TTL	3（保证协议数据包能到达 R6）
OSPF 头部	类型	1（代表 LSU 包）
	路由器标识	3.3.3.3（生成此包的路由器标识，冒充 R3 身份发送）
	区域标识	0.0.0.2（区域号）

LSU 头部	LSA 数量	1（表示链路状态的数量）
LSA 头部	链路状态类型	1（代表路由器 LSA）
	链路状态标识	3.3.3.3（与链路状态类型有关）
	宣告路由器	3.3.3.3（此路由器标识的 LSA 被伪造）
	序列号	80000010（保证大于现有 LSA 的序列号）
路由器 LSA	链路标识	192.168.30.0（伪造的网络号）
	链路数据	255.255.255.0（掩码地址）
	链路类型	3（表示的是残余网络）
	接口距离值	1
	链路标识	20.129.22.6（与真实值一样，保证了 R3 与 R6 互通）
	链路数据	20.129.22.3（与真实值一样，保证了 R3 与 R6 互通）
	链路类型	2（表示的是传输网络）
	接口距离值	1

3. 欺骗攻击结果

攻击者 A2 发出这个单恶意包后，图 3-21 显示了攻击后 R6 与 R3 中关于 R3 的路由器 LSA 信息，可以看到 R6 上显示的网络号为 192.168.30.0/24，与真实的网络拓扑不再一样。除了 R6，R2 与 R1 因为在污染区内，所以它们的链路状态数据库也会被修改。由于 R6 是 ABR，所以它也会把这个网络泛洪至其他的区域，对其他区域也会有影响。图 3-22 显示了攻击后用户 C3 到用户 C2 的传输路径，可以看到流量被欺骗到 R3 中，由于 R3 收到数据后会丢弃，导致了流量黑洞。

```
R3#show ip ospf database r 3.3.3.3

           OSPF Router with ID (3.3.3.3) (Process ID 3)

               Router Link States (Area 2)

  LS age: 20
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 3.3.3.3
  Advertising Router: 3.3.3.3
  LS Seq Number: 80000002
  Checksum: 0x3CB2
  Length: 48
  Number of Links: 2

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 192.168.80.0
     (Link Data) Network Mask: 255.255.255.0
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 20.129.22.6
     (Link Data) Router Interface address: 20.129.22.3
      Number of TOS metrics: 0
       TOS 0 Metrics: 1
```

```
R6#show ip ospf da router 3.3.3.3

           OSPF Router with ID (6.6.6.6) (Process ID 6

               Router Link States (Area 2)

  LS age: 15
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 3.3.3.3
  Advertising Router: 3.3.3.3
  LS Seq Number: 80000010
  Checksum: 0x5CB6
  Length: 48
  Number of Links: 2

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 192.168.30.0
     (Link Data) Network Mask: 255.255.255.0
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 20.129.22.6
     (Link Data) Router Interface address: 20.129.22.3
      Number of TOS metrics: 0
       TOS 0 Metrics: 1
```

图 3-21 攻击后 R6 与 R3 上的 R3 路由器 LSA

```
PC-C3#traceroute 192.168.30.2

Type escape sequence to abort.
Tracing the route to 192.168.30.2

  1 192.168.90.1 1020 msec 20 msec 8 msec
  2 20.129.23.6 32 msec 20 msec 24 msec
  3 20.129.22.3 48 msec 32 msec 16 msec
  4 20.129.22.3 !H  !H  !H
```

图 3-22 攻击后用户 C3 到 C2 的传输路径

3.5.5 远程邻接欺骗攻击

1. 欺骗攻击场景

如图 3-23 所示，攻击者 A1 尝试以幻影路由器与 R8 建立虚假的邻接关系，并以幻影路由器的身份注入 121.195.178.0/24 这条路由，这样区域中所有去往该网段的流量都会被欺骗到幻影路由器这里。

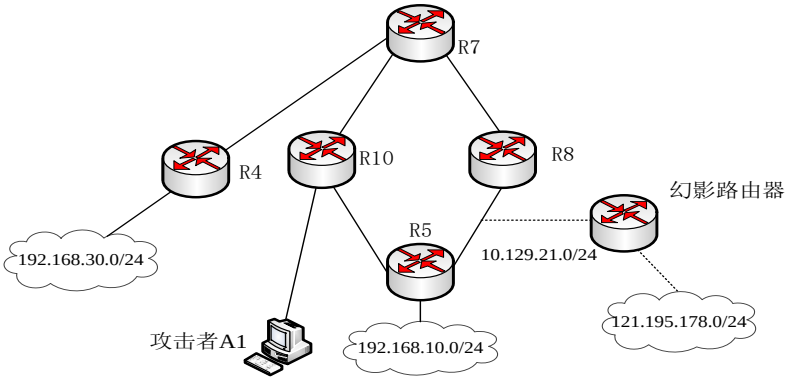

图 3-23 远程邻接欺骗攻击示意图

2. 欺骗攻击实现

远程邻接欺骗攻击分为三个阶段，每个阶段需要构造不同的协议包，分别为测试 Hello 包、DBD 包以及 LSU 包。由于在仿真环境中已经知道了 R8 运行的基本参数（真实场景下需不断探测），所以测试 Hello 包和 DBD 包的构造就很容易。下面主要列出 LSU 包的参数，如表 3.3 所示。

表 3.3 幻影路由器 LSA 的主要参数

位置	字段	值以及描述
IP 头	源地址	10.129.21.2（幻影路由器的发送接口地址）
	目标地址	10.129.21.8
	协议号	89（OSPF 协议）
OSPF 头部	类型	4（代表 LSU 包）
	路由器标识	30.30.30.30（标识值要大于 R8 的标识值）
	区域标识	0.0.0.1（区域号）
LSU 头部	LSA 数量	1（表示链路状态的数量）
LSA 头部	链路状态类型	1（代表路由器 LSA）
	链路状态标识	30.30.30.30
	宣告路由器	30.30.30.30 表示该 LSA 是幻影路由器的 LSA）
	序列号	80000002
路由器 LSA	链路类型	2（表示传输网络）
	链路标识	10.129.21.8（指定路由器的接口地址）
	链路数据	10.129.21.2（本机路由器的接口地址）
	距离值	1
	链路类型	3（表示残余网络）
	链路标识	121.195.178.0（表示伪造的网络）
	链路数据	255.255.255.0
	距离值	1

3. 欺骗攻击结果

图 3-24 显示了对路由器 R8 进行远程邻接欺骗时，在 R8 接口 f2/0 捕获到的协议数据包。由于这里已经知道了 R8 运行的基本参数，所以直接构造正确的测试 Hello 包，当发送测试 Hello 包后，R8 发回一个 DBD 包。此后，双方交互一系列的 DBD 包，主要用于 R8 发送自己链路状态数据库中信息，而幻影路由器发出的 DBD 包是没有 LSA 内容，所以不会有 LSR 包出现。当虚假的邻接关系建立之后，攻击者以幻影路由器的身份发送一个恶意的 LSA，这个 LSA 中包含了网络 121.195.178.0/24，R8 将这个恶意的 LSA 保存在链路状态数据库中，并发回了 LSAck 进行了确认。

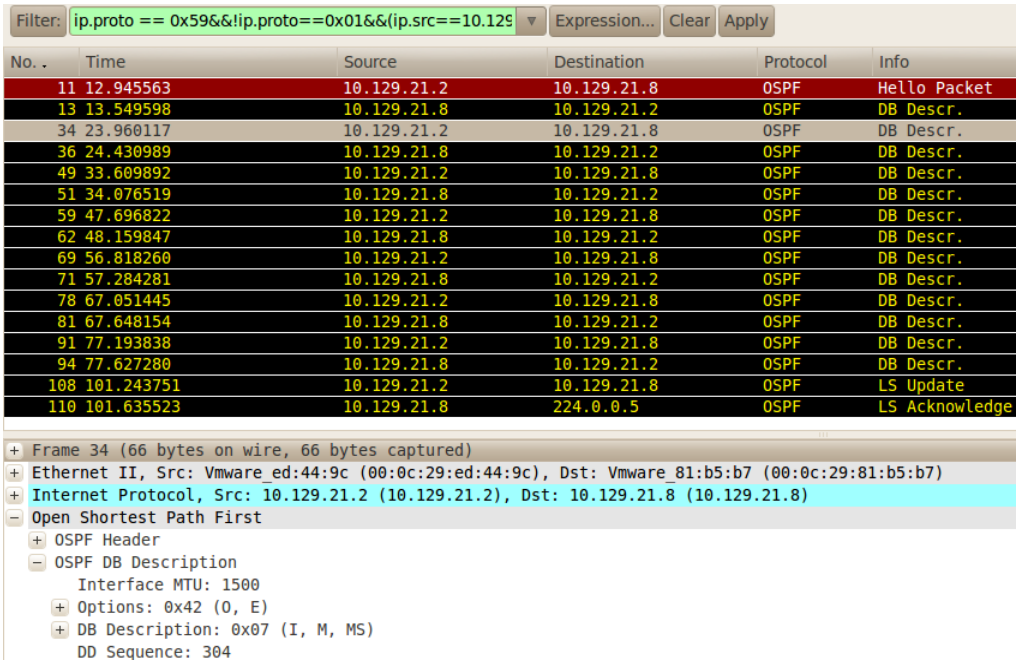

图 3-24 远程邻接欺骗攻击包交互过程

图 3-25 显示了攻击前后 R8 的路由表，可见它的路由表被恶意篡改。除了 R8，区域 1 中其它路由器的路由器也会被篡改。所有去往 121.195.178.0/24 的流量都会被欺骗到幻影路由器这里，从而形成了流量黑洞。

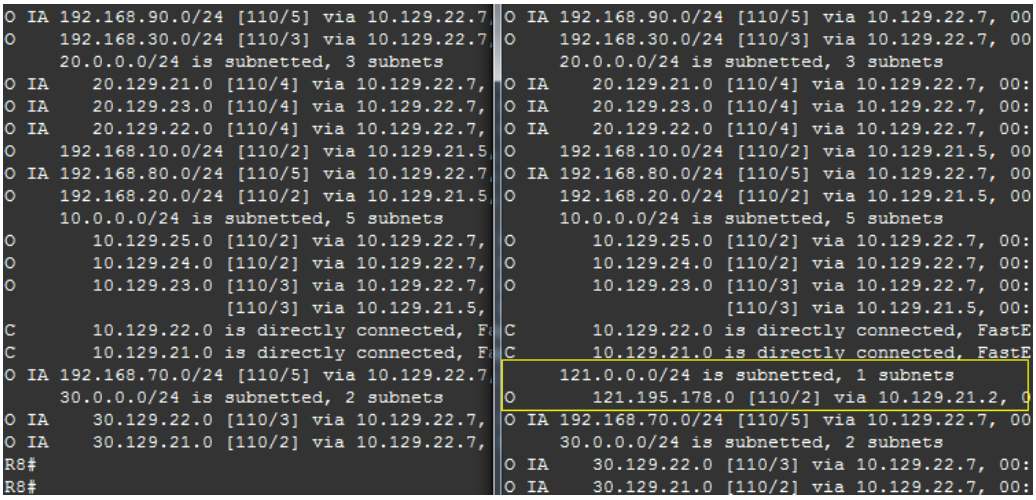

图 3-25 远程邻接欺骗攻击前后 R8 的路由表

3.6 本章小结

本章提出了邻接欺骗攻击、双 LSA 远程多注入攻击、单路径注入攻击、以及远程邻接欺骗攻击这四种路由欺骗攻击，分析了它们的攻击原理以及应用场景。接着基于路由欺骗攻击的理论基础，利用搭建的网络模拟平台对这四种路由欺骗攻击进行仿真。通过仿真实验，验证了它们的可行性以及有效性。

第4章 OSPF 渗透测试系统的实现

本章首先介绍 OSPF 渗透测试系统的总体架构，渗透测试系统主要分为监测模块以及攻击测试模块，这两个模块又由许多子模块组成。接着对各个功能子模块的实现进行详细介绍。

4.1 系统总体架构

渗透测试系统的总体架构如图 4-1 所示，主要分为两个大模块，其中监测模块由 OSPF 探测模块、协议分析模块、密钥认证破解模块、OSPF 协议模块这些子模块组成。而攻击测试模块由协议包构造模块、协议配置接口模块、最大序列号测试模块、LSA 覆盖测试模块以及双 LSA 注入测试模块组成。

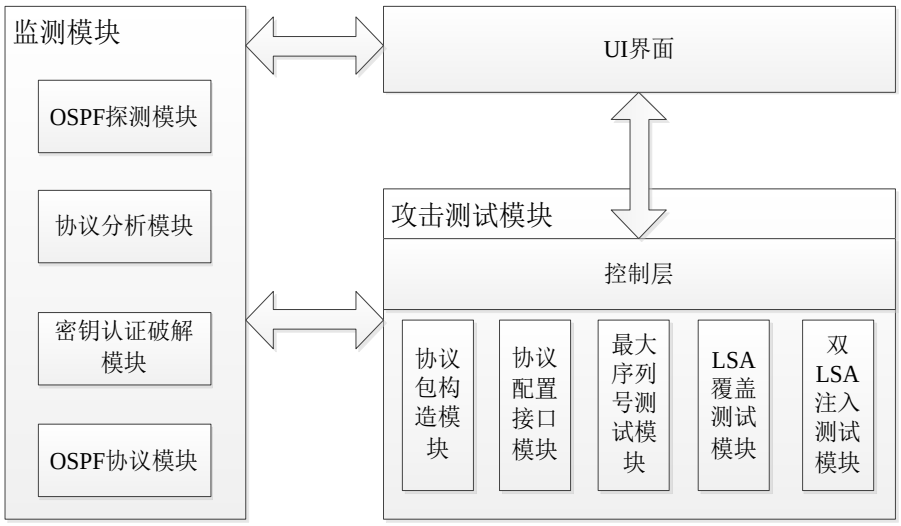

图 4-1 系统总体架构

渗透测试系统提供图形化的交互界面，用户利用图形接口对监测模块以及攻击测试模块进行控制。监测模块主要负责与网络中的路由器建立邻接关系，而攻击测试模块主要用于对 OSPF 网络进行安全性测试。攻击测试模块依赖于监测模块，只有当监测模块监测到协议并且建立邻接关系后，攻击测试模块才能进行各种安全测试分析。

本系统选择 ubuntu 操作系统作为实现的平台，利用 Qt 作为渗透测试系统开发工具。下面将详细介绍这些子模块的实现。

4.2 监测模块

监测模块主要用于监测网络是否运行 OSPF 协议，如果发现运行了 OSPF 协议，渗透测试系统会对接收的协议包进行分析。如果发现协议采用了密钥认证机制，就利用密钥认证破解模块进行破解。最后，OSPF 协议模块会根据得到的参数对运行参数进行调整，从而与邻居路由器建立邻接关系。

4.2.1 OSPF 探测模块

OSPF 探测模块的作用是探测网络是否运行 OSPF 协议，由于运行 OSPF 协议的路由器会发送出 Hello 包（除被动接口），所以本模块会监听接收到的数据包，并判断是否有 Hello 包。

Libpcap 是 unix/linux 平台下的函数包[32]，提供了一系列用户级的接口，可以实现对于网络底层数据包的访问。本模块的实现基于 libpcap 函数库，libpcap 主要由网络分接头以及数据过滤器两部分组成。网络分接头用来从网络设备驱动程序中接受数据，而过滤器主要是对接收到的链路层的数据进行过滤。利用 libpcap 提供的接口可以很方便的开发监听程序。

图 4-2 显示了监听程序运行的流程，下面将对各个步骤的实现进行介绍。

1) 选择并开启网络接口调用了 pcap_open_live()函数，并设置监听的网络接口开启混杂模式。
2) 设置过滤条件调用 pcap_compile()和 pcap_setfilter()函数，过滤表达式基于 BSD Packet Filter(BPF)结构，这时表达式为“IP[9]==89”（即 OSPF 协议）
3) 接受数据包调用 pcap_loop()函数，使系统一直监听数据，当接收到 OSPF 包后发送到处理程序。
4) 数据包处理程序需要从 OSPF 包中过滤出 Hello 包，过滤表达式设置为“ospf_hdr->ospf_type==0x01”。如果发现接收到了 Hello 包，则将整个 Hello 包保存下来；如果没有接收到 Hello，则继续接收数据包。

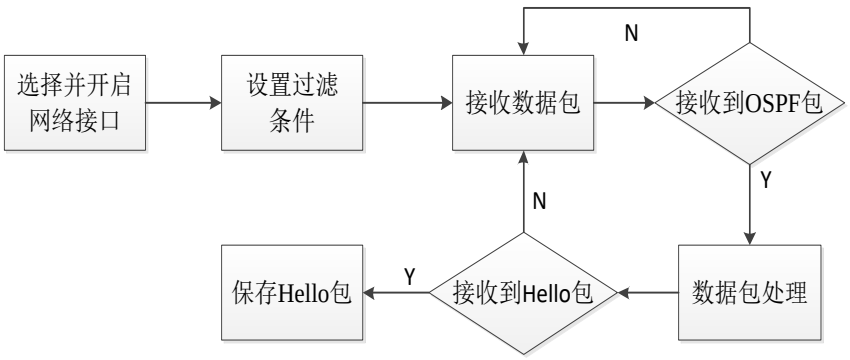

图 4-2 监听程序运行流程

4.2.2 协议分析模块

协议分析模块主要对接收的 Hello 包进行分析，并且提取出 Hello 包中一些关键的参数。

当监听到 Hello 包后，协议分析程序会将接收到的关键字段值与本地默认值进行比较。表 4.1 列出了本地关键字段的默认值，如果发现接收到的值与它们不一致，那么就将相应的接收值保存起来。验证类型有三种类型，当发现验证类型为明文认证（值为 1）

时，协议分析程序将保存 64 位的明文密钥；而当发现验证类型为密钥认证（值为 2）时，协议分析程序将会把保存的 Hello 包留给密钥认证破解模块处理。此外，协议分析程序还会把网络掩码的值、路由器标识以及发送端 IP 地址保存起来。这些保存起来的参数，主要用于密钥认证破解模块以及 OSPF 协议模块。

表 4.1 Hello 包的关键字段默认值

关键协议字段	默认值
区域标识	0
验证类型	0（默认不验证）
发送时间间隔	10 秒
死亡时间间隔	40 秒
选项	0x02

4.2.3 密钥认证破解模块

密钥认证破解模块主要用于破解 OSPF 协议的密钥认证机制，下面首先介绍密钥认证破解的原理以及密钥认证破解模块的实现，最后利用密钥认证破解模块对密钥认证机制的安全性进行测试分析。

4.2.3.1 密钥认证破解原理

OSPF 密钥认证机制采用了 MD5 算法，由于 MD5 算法的不可逆性，所以无法根据 128 位的输出结果推出原始信息。关于 MD5 攻击主要分为 3 种[33]，第一种是原像攻击，给定了 H 值，找到明文 M，使得 H=hash(M)；第二种是次原像攻击，给定了明文 M1，找到另一个不同于 M1 的明文 M2，使它们满足 hash(M2)=hash(M1)；第三种是碰撞攻击，找到两个不同明文 M1 和 M2，使得 hash(M1)=hash(M2)。这三种攻击中，第一种和第三种攻击破解的难度较易，第二种攻击目前尚未有突破性进展。

图 4-3 描述了 MD5 认证的过程，根据这个过程可知密钥认证破解属于原像攻击，它满足 H=hash(Mo+Ms)这条公式，其中 Mo 为收到的明文 OSPF 包，H 为 MD5 算法生成的散列值，而 Ms 就是需要破解的密钥。

原像攻击的方法一般分为 3 种，分别为字典法、暴力破解法以及查询 MD5 破解网站。查询 MD5 破解网站主要是根据散列值到网站上的数据库中查找相应的明文，这种方式对于破解密钥认证不一定成功，因为数据库中不一定存在相应的散列值。字典法是将常见的密钥存在一个文本中，破解程序逐一测试，这种方法需要一个好的字典集，否则就无法破解密钥。暴力破解就是对所有可能的情况进行逐一的测试，设置的密钥范围越大，破解的可能性就越大，但是可能需要花很长时间。OSPF 协议密钥的最大长度为 16 个字节，而密钥的设置由管理员手工设置，一般输入的都是常见字符，所以穷举的范围会大大减小。

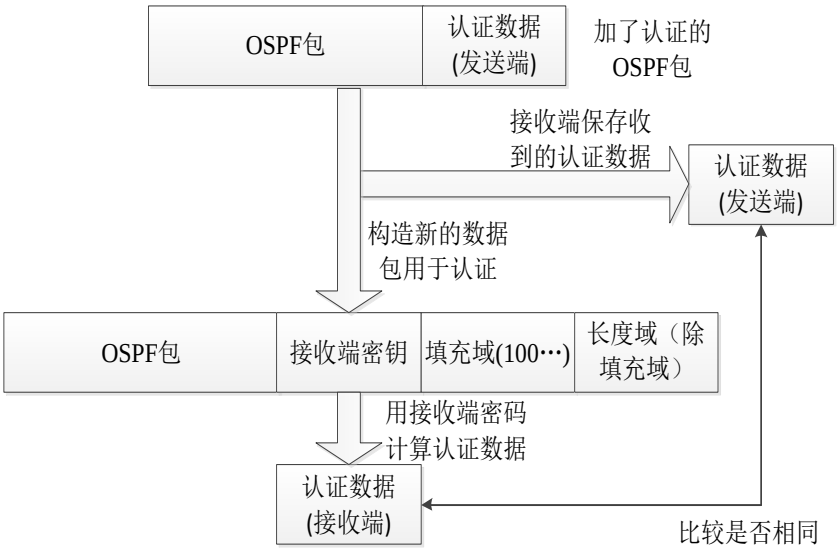

图 4-3 MD5 认证过程

4.2.3.2 破解模块实现

破解模块的结构设计如图 4-4 所示，密钥认证破解程序通过界面接口获得字符集和文件的位置，从而将必要的信息全部加载到破解程序中。该破解模块实现了两种破解方式，分别为暴力破解以及字典破解。MD5 计算进程会根据 OSPF 包对测试密钥一一计算，直到密钥空间的密钥全部穷举完或找到了正确密钥为止。

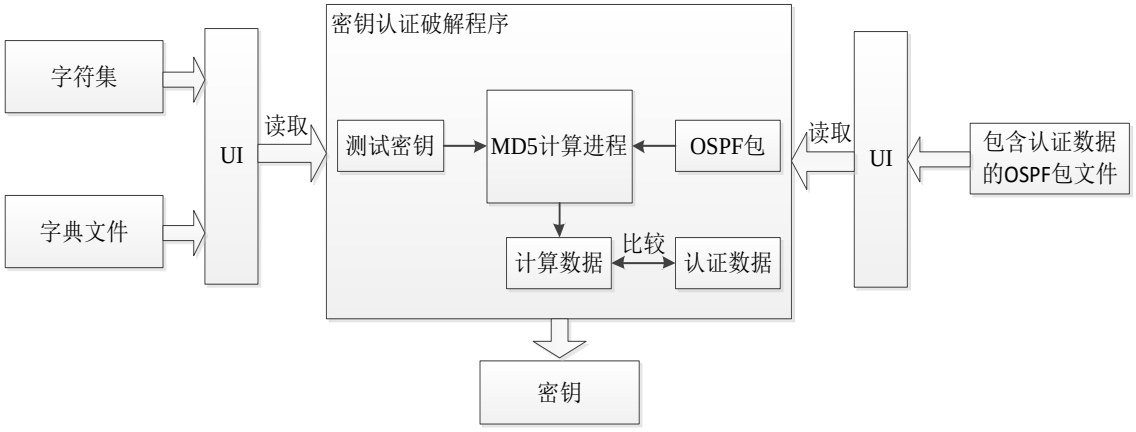

图 4-4 破解模块结构图

字典文件以及包文件的读取主要利用 QFile 类来实现，而对于字符集的读取，需要通过穷举的方式输入所有情况，这里采用递归的算法来实现，实现代码如下：

```
void Crack(QString strKey, int len) //len 表示穷举字符串的长度，strKey 测试的密钥
{
    if (len == 0)
     {
        if(MD5Crack(strKey,ospfPacket,authData)==1)
        {                          //MD5Crack()为 MD5 认证破解函数
            finalKey=strKey;    //ospfPacket 为接收的 OSPF 包
```

```
            }                              //authData 为认证数据
            else
            {
                return;
            }
        }
        else
        {
            for (int i = 0; i < sLength; i++)   //sLength 为字符集的长度
            {
                strKey [len - 1] = CharSource[i]; //CharSource 存储着字符集
                Crack(strKey, len - 1);
            }
        }
}
```

4.2.3.3 密钥认证安全性测试

下面基于密钥认证破解模块以及暴力破解法来测试密钥认证的安全性，一般攻击者很难知道密钥中的字符种类，这里以最坏的情况来进行测试，假设攻击者已经知道了字符的种类。根据这个条件，下面将分析破解时间与密钥的长度、字符的种类的关系。

1. 测试环境

密钥认证安全性测试的环境参数如表 4.2 所示，其中密钥穷举的速度是通过密钥认证破解模块对主机进行测试得到。

表 4.2 密钥认证安全性测试环境参数

测试参数	配置描述
操作系统	64 位 Windows 7
处理器	AMD Athlon Ⅱ X4 Quad-Core Processor
主频	2.80GHz
内存	4G
密钥穷举的速度	平均 1350000Pass/Sec

2. 测试结果

1) 破解时间与单一类字符长度的关系

假设密钥是同一类的字符，比如说纯数字（0～9）、纯小写字母（a～z）或纯大写字母（A～Z）等，以纯小写字母为例来说明。

从表 4.3 中可以看到随着密钥长度的增大，破解所需的时间也会增大。当密钥长度小于 9 个字符时，破解密钥还是比较容易；当密钥长度大于 9 个字符时，破解密钥所需要的时间会很长；而当密钥长度超过 11 个字符时，密钥就很能破解。

表 4.3 破解时间与字符长度的测试结果

密钥	5	6	7	8	9	10	11	12

长度								
密钥空间	1.12 E7	3.09 E8	8.03 E9	2.01 E11	5.43 E12	1.41 E14	3.67 E15	9.54 E16
平均时间	4.4 (sec)	114.4 (sec)	49.6 (min)	21.4 (hour)	23.2 (day)	605.1 (day)	43.1 (year)	1120.8 (year)

2) 破解时间与字符种类的关系

在这里假设密钥的长度为 8 个字符，比较纯数字、纯小写字母、数字和小写字母、大小写字母以及数字和大小写字母的破解时间。

从表 4.4 中可以看到随着字符范围的增大，破解所需的时间也会增大。当密钥以数字加大写字母或小写字母时，破解密钥还是比较容易；当密钥为大小写字母或三者结合，则需要很长时间。除了上面的字符，密钥中还可以包括@、#、%、!等等这些比较特殊的字符，这样破解的难度就会大大的提高。

表 4.4 破解时间与字符种类的测试结果

字符种类	0～9	a～z	0～9&a～z	a～z &A～Z	0～9&a～z &A～Z
密钥空间	1E8	2.01 E11	2.82 E12	5.34 E13	2.18 E14
平均时间	37.04 (sec)	21.48 (hour)	12.09 (day)	229.2 (day)	2.56 (year)

综上所述，OSPF 密钥认证的安全性与密钥长度以及字符种类有着密切的关系，密钥长度越长，字符种类越多（特别是特殊字符），那么安全性就越强。除了字符的长度以及种类外，计算的速度也是一个影响密钥认证安全的重要因素，由于测试主机是普通的 PC 机，速度还不是很快，如果换成计算速度更快的机器，那么破解的时间也会大大缩短。近些年，随着云计算技术的发展、计算机处理能力的不断增强，破解密钥的难度也会不断降低。此外，改进破解的算法对于降低破解时间会有很大的帮助。

然而，在现实生活中，很多网络管理员缺乏安全意识或是为了密钥便于记忆，配置的密钥不会很复杂，这样暴力破解法就能很容易破解密钥，所以 MD5 密钥认证保护也并非一定安全。

4.2.4 OSPF 协议模块

OSPF 协议模块主要是对 OSPF 协议进行初始配置，使它的运行参数与邻居路由器的参数一致，接着启动 OSPF 协议，从而与邻居路由器建立邻接关系。它的实现主要基于对 Quagga 路由软件配置文件的操作。

4.2.4.1 Quagga 开源路由软件

OSPF 渗透测试系统的实现依赖路由软件，开源路由软件的种类有很多，比较常用的有 Quagga、XORP、Vyatta 等。这里选择 Quagga 作为系统实现的一部分。Quagga 是

一个 Unix 平台下的开源路由软件套件[34]，实现 OSPF、RIP、IS-IS 以及 BGP 这些路由协议。它不仅支持 IPv4 路由协议，还支持 IPv6 路由协议。

传统的路由软件是一个单进程，运行着各种不同的路由协议。如图 4-5 所示 Quagga 路由软件的实现基于模块化，这样软件的健壮性以及扩展性更强。它的核心进程为 zebra，是 Unix 底层核心的抽象层，表示为一些 Zserv API 或 Quagga 客户端的 TCP 流。Quagga 的客户端包括了 RIPd、BGPd、OSPFd、IS-ISd、Vtysh 等等，它们实现了不同的动态路由协议，Vtysh 是一个虚拟的终端接口，用来与这些进程进行交互。

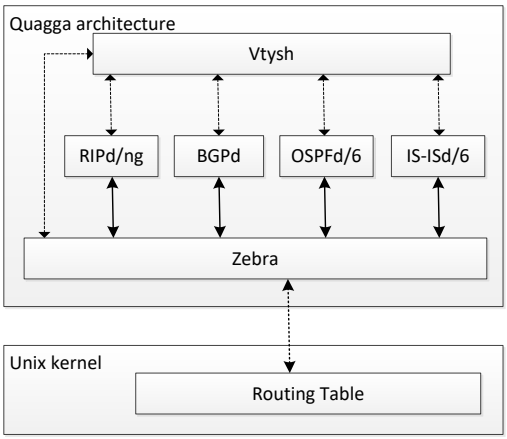

图 4-5 quagga 开源路由软件的结构

4.2.4.2 OSPF 协议模块与 Quagga 的交互

OSPF 协议模块在获得建立邻接关系的基本参数后，就可以对 Quagga 软件的配置文件进行修改。图 4-6 显示了 OSPF 协议模块、Quagga 以及终端的交互过程，OSPF 协议模块需要对 Quagga 中的 daemons、zebra.conf 以及 ospfd.conf 进行配置，同时它通过终端命令来控制 Quagga 的运行与停止。

为了保证系统每次运行时 Quagga 的配置文件 ospfd.conf 和 zebra.conf 都是原始的，系统需要在本地保存这两个配置文件的原始版本，当系统运行后 OSPF 协议模块调用终端命令来覆盖，具体的命令为：

```
system("cp /home/OSPFAttackSys/ospfd.conf /etc/quagga/ospfd.conf")
system("cp /home/OSPFAttackSys/zabra.conf /etc/quagga/zebra.conf")
```

而在本地保存的 deamons 需要对其进行修改，相应的修改项为：zebra=yes 以及 ospfd=yes，当系统运行后同样调用终端命令来覆盖 Quagga 中 deamons 文件。

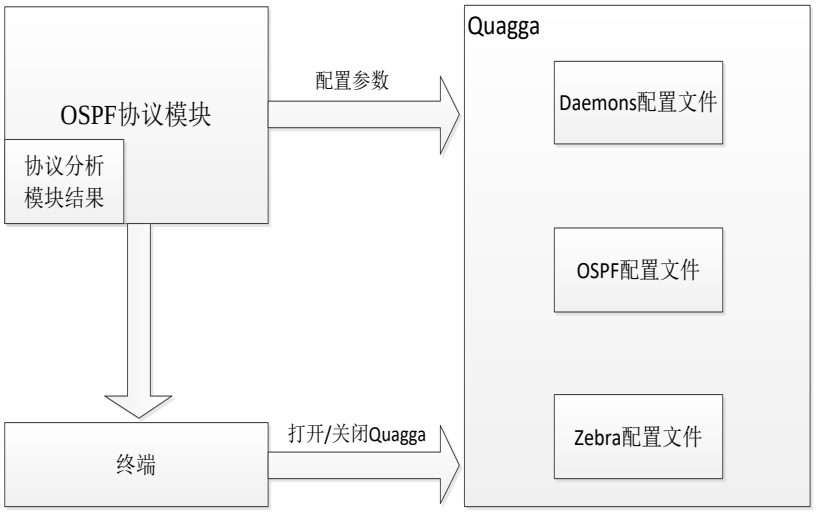

图 4-6 OSPF 协议模块与 quagga 的交互

根据协议分析模块的结果，OSPF 协议模块会自动对 ospfd.conf 进行配置，根据发送端 IP 地址、区域标识以及网络掩码来生成 network 命令；根据验证类型、64 位验证数据以及破解密钥（若使用了密钥认证机制）来生成明文认证或密钥认证配置命令；根据包发送间隔与路由器死亡时间来生成发送及死亡时间命令。在对配置文件进行修改前需要先找到"interface eth"以及"router ospf "，然后写入相应的配置。具体的代码如下：

```
while(! file.atEnd())                          //file 代表 ospfd.conf，并且以可读写的形式打开
{
    strLine=file.readLine();
    if(strLine.trim()=="router ospf \n")    //接口下配置时为"interface eth"
    {
        break;                                 //得到写入配置的位置
    }
}
QString ospf_conf_str= "area 0.0.0.0 authentication"; //配置命令的例子
file.write(ospf_conf_str);                     //写入配置命令
```

当所有的配置命令都写入后，就可以启动 zebra 以及 ospf 进程，具体的命令为：system("service quagga start")，这时邻接关系就开始准备建立。

4.3 攻击测试模块

攻击测试模块是渗透测试系统的核心模块，它提供了各种协议包构造的图形接口，此外它还集成了几个典型的测试子模块。利用这些测试子模块，可以对 OSPF 网络的安全性进行测试。

4.3.1 协议包构造模块

协议包构造模块主要用于构造各种 OSPF 协议包，包括 Hello 包、DD 包、LSR 包、LSU 包以及 LSAck 包。该模块提供了一个图形化的用户接口，使得协议包的构造更加的方便。

本模块的实现主要基于 libnet 函数库[35]，该函数库用于对低层网络数据包进行构造、处理以及发送，它提供了许多协议包构造的函数接口，例如：ICMP、RIP、ARP、IGRP、DNS、OSPF 等等。Libnet 函数库支持各种 OSPF 协议包的构造，如表 4.5 所示，它提供了非常完整的函数接口，所以本模块基于该函数库来进行开发。

表 4.5 包构造主要的函数接口

包构造函数接口	描述
libnet_build_ipv4	构造 IP 包头部
libnet_build_ospfv2	构造 ospf 包头部
libnet_build_ospfv2_hello	构造 Hello 包
libnet_build_ospfv2_lsu	构造 LSU 包
libnet_build_ospfv2_lsa	构造 LSA 的头部
libnet_build_ospfv2_lsa_rtr	构造路由器 LSA
libnet_build_ospfv2_lsa_net	构造网络 LSA
libnet_build_ospfv2_lsa_sum	构造网络汇总 LSA /ASBR 汇总 LSA
libnet_build_ospfv2_lsa_as	AS 外部 LSA
libnet_build_data	添加更多的 LSA 到 OSPF 包尾部

不同协议数据包构造的过程基本一样，以构造路由器 LSA 为例来介绍包构造程序的流程。图 4-7 显示了包构造程序运行的流程，下面将对各个步骤的实现进行介绍。

1) 选择网络接口、初始化 libnet 调用了函数 libnet_init()，初始化后会返回 libnet_t 类型句柄。表 4.5 中的函数都需要用到这个参数。
2) 协议数据包一般都是从上层往下层进行封装，因此首先构造路由器 LSA 和 LSA 头部。路由器 LSA 以及 LSA 头部的构造调用函数 libnet_build_ospfv2_lsa_rtr()和 libnet_build_ospfv2_lsa()。
3) 一般 OSPF 包有三个校验和需要计算，分别为 IP 校验和、OSPF 校验和以及 LSA 校验和，其中 IP 以及 OSPF 校验和在 libnet_build_ipv4()以及 libnet_build_ospfv2()函数中自动生成，而 LSA 的校验和在函数 libnet_build_ospfv2_lsa()不会自动生成，需要利用 fletcher 算法进行计算生成。
4) LSU 头部的构造调用函数 libnet_build_ospfv2_lsu()，需要设置链路状态的个数。
5) OSPF 头部以及 IP 包头的构造调用 libnet_build_ospfv2()和 libnet_build_ipv4()，需要对多个字段进行设置。

6) 协议数据包的发送调用函数 libnet_write()，它唯一的参数就是初始化时返回的句柄。

7) 为了降低内存的消耗，发送完后调用 libnet_destory()进行内存的释放。

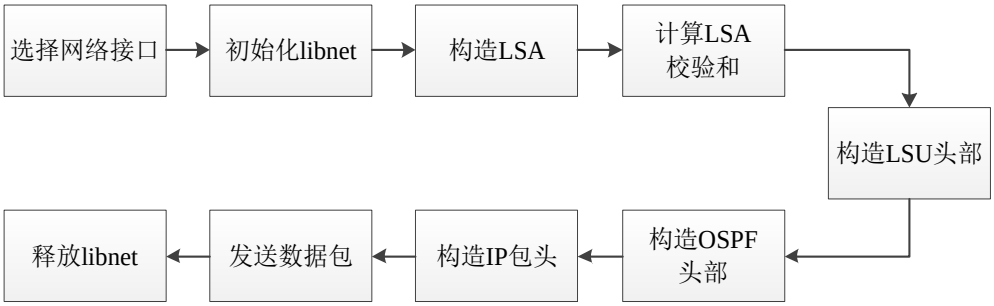

图 4-7 协议包构造程序流程

4.3.2 协议配置接口模块

协议配置接口模块提供了对 OSPF 协议配置的图形接口。利用配置接口模块，测试者可以方便的对 OSPF 协议进行配置，从而实现邻接欺骗攻击。图 4-8 显示了协议配置接口交互示意图，对 OSPF 协议的配置可以通过两种方式来实现，分别为终端方式以及配置文件方式。

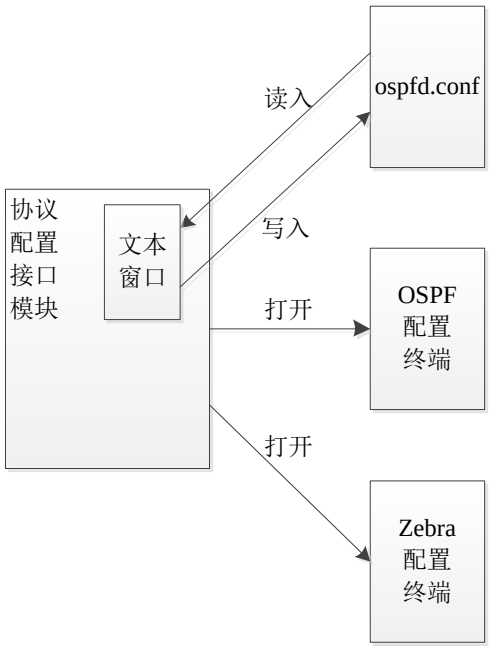

图 4-8 协议配置模块交互示意图

终端方式是指测试者打开协议配置的命令行终端，这种方式下命令一条一条输入到协议进程中。利用命令 system("telnet localhost 2604")和 system("telnet localhost 2604")可以分别打开 OSPF 配置终端以及 zebra 配置终端。配置文件方式是指测试者直接对协议的配置文件进行修改，这种方式下许多命令同时输入到协议进程中。配置文件方式首先需要读入 ospfd.conf 的内容到本文窗口，测试者直接在窗口中进行修改；接着配置接口

模块将配置信息写入到 ospfd.conf 中。为了使写入的配置信息生效，需要在写入后运行命令 system("service quagga restart")。

4.3.3 最大序列号测试模块

最大序列号测试模块主要用于测试 OSPF 协议实现能否抵抗最大序列号攻击。图 4-9 描述了最大序列号测试的流程，测试模块首先发送一个序列号为 0xFFFFFFFF 的被测路由器 LSA；然后测试模块发送一个 LSR 包请求被测路由器 LSA，这时测试模块会收到被测路由器 LSA。测试模块只需要判断 LSA 序列号是否为 0x80000001，如果序列号为 0x80000001，则可以抵抗最大序列号攻击。

图 4-9 最大序列号测试流程

被测路由器 LSA 不需要与真实的 LSA 完全相同，只需要使 LS 类型、LS 标识以及宣告路由器字段保持一致。协议分析模块中保存了邻居路由器的标识，利用这个标识可以构造出符合要求的被测路由器 LSA。

4.3.4 LSA 覆盖测试模块

LSA 覆盖测试模块主要用于测试 OSPF 协议实现能否抵抗 LSA 覆盖攻击。图 4-10 描述了覆盖测试的流程，测试模块首先发送一个伪造的被测路由器 LSA，其中 LS 标识设为被测路由器标识，宣告路由器设为某一不存在的标识；接着测试模块发送一个 LSR 包，包中 LS 标识和宣告路由器的值与发送时一样。测试模块只需要判断被测路由器是否回复一个对应的 LSA，如果被测路由器没有回复对应的 LSA，那么该路由器可以抵抗 LSA 覆盖攻击。

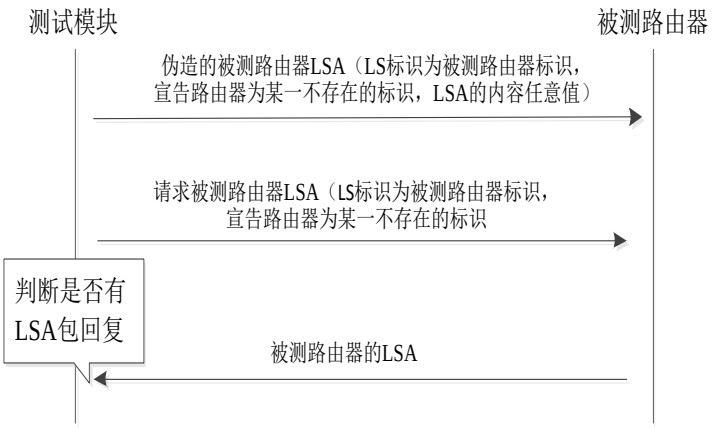

图 4-10 LSA 覆盖测试流程

覆盖测试过程中被测路由器 LSA 以及 LSR 包的构造主要针对 LSA 头部进行设置，协议分析模块提供了路由器标识，宣告路由器设为任意不存在的标识，其余字段只需设为符合协议标准的值即可。

4.3.5 双 LSA 注入测试模块

双 LSA 注入测试模块主要实现了触发 LSA 以及抗反击 LSA 的构造，并且能够控制它们之间发送的时间间隔，利用该模块可以测试网络受双 LSA 注入攻击的影响范围。

触发 LSA 以及抗反击 LSA 在注入之前首先需要构造相应的包，并且设定它们之间的时间间隔，否则无法进行注入。协议包构造的函数接口以及构造流程在 4.3.1 节中已经详细的介绍，时间间隔的控制调用了函数 Sleep()。

在介绍双 LSA 注入攻击方法时，提出了矫正字段的概念，有了矫正字段抗反击 LSA 与自反击 LSA 就可以设计为同一实例，从而欺骗其他路由器。本模块具有计算矫正字段的功能，从而保证两个内容不一样的 LSA 具有相同的头部（除 age 字段）。对于不同类型的 LSA，它们的矫正字段也不相同，表 4.6 列出了不同类型 LSA 的矫正字段。

表 4.6 不同类型 LSA 的矫正字段

LSA 类型	矫正字段
路由器 LSA	链路标识/链路数据/类型/TOS 数/距离值
网络 LSA	接入路由器标识
网络汇总/ASBR 汇总 LSA	0/距离值
AS 外部 LSA	转发地址/外部路径标识

不同类型 LSA 矫正字段的计算方法类似，这里以路由器 LSA 为例进行介绍。矫正字段的计算主要对字段进行穷举，直到使抗反击 LSA 与自反击 LSA 的校验和相同。为了实现的方便，固定其中的 4 个字段，对一个字段进行穷举。由于校验和长度为 16 位，穷举字段长度只需要大于等于 16 位即可。这里将链路标识、链路数据、类型以及 TOS 数设为固定值，穷举距离值字段，实现的代码如下：

```
m_ospf_LSU_rtr.m_rtr_correct.rtr_link_id=htonl(0x08080808); // m_rtr_correct 为矫正字段
m_ospf_LSU_rtr.m_rtr_ correct.rtr_link_data=htonl(0x08080808);
m_ospf_LSU_rtr.m_rtr_ correct.rtr_type=2;
m_ospf_LSU_rtr.m_rtr_ correct.rtr_tos_num=0;
m_ospf_LSU_rtr.m_rtr_ correct.rtr_metric=0;//矫正字段中固定前 4 个值，枚举距离值
for(;m_ospf_LSU_rtr.m_rtr_ correct.rtr_metric<0xffff;
m_ospf_LSU_rtr.m_rtr_ correct.rtr_metric++){
    if(ospf_lsa_checksum((struct libnet_lsa_hdr *)&m_ospf_LSU_rtr)==originchecksum)
    break;
}//originchecksum 为自反击 LSA 的校验值，可以提前确定
```

4.4 本章小结

本章首先设计了 OSPF 渗透测试系统的总体架构。整个系统可以划分为监测模块以及攻击测试模块，监测模块与攻击测试模块又可以划分为许多的子模块。接着详细的介绍了这些子模块的实现，并且利用 OSPF 渗透测试系统对 OSPF 协议的密钥认证机制的安全性进行了分析。

第5章 OSPF 安全性测试

本章利用 OSPF 渗透测试系统在某真实的 OSPF 网络中进行安全性测试，使用邻接欺骗攻击进行路由欺骗测试，分别测试了网页欺骗、邮箱密码嗅探以及 DNS 欺骗。

5.1 测试环境

图 5-1 显示了测试网络的实际拓扑结构，该网络由四个子网络组成。网络中的路由器都运行了 OSPF 协议，保证了整个网络的连通。图中标出了测试者所在的位置，他位于网络 1 的某一网段中。

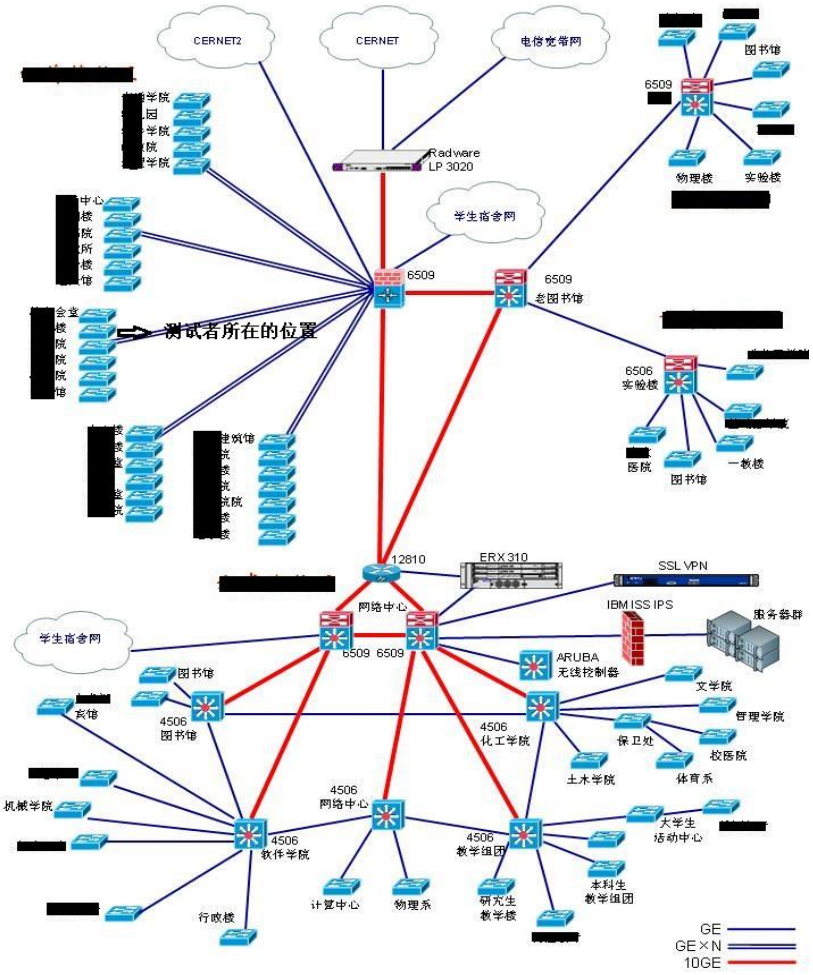

图 5-1 真实 OSPF 网络拓扑图

5.2 测试实现与结果

邻接欺骗攻击分为系统接入以及配置注入两步，下面分别介绍它们在真实环境下的实现。

5.2.1 系统接入

系统接入的实现分为两步，分别为网络接口配置以及运行 OSPF 探测模块。现在很多网络都采用了 DHCP 服务来分配 IP 地址，但是这里的网络环境没有 DHCP 服务，需要手动配置网络接口地址。网络的默认网关为 172.21.134.1，所以接口配置命令可以设为 ifconfig eth1 172.21.134.2 netmask 255.255.255.0 up。

网络接口配置完后，就可以运行监测模块。OSPF 探测模块发现图 5-3 中邻居路由器接口 172.21.134.1 运行了 OSPF 协议且未设为被动接口。图 5-2 显示了监测模块中协议分析子模块对于 Hello 包的解析，可以看到关键的参数值与表 4.1 所列出的默认值相同，因此 OSPF 协议模块只会向配置文件加入 network 172.21.134.0/24 area 0 这条命令。经过短暂的协商后，系统就可以接入到 OSPF 网络之中。

图 5-2 协议分析结果

5.2.2 配置注入

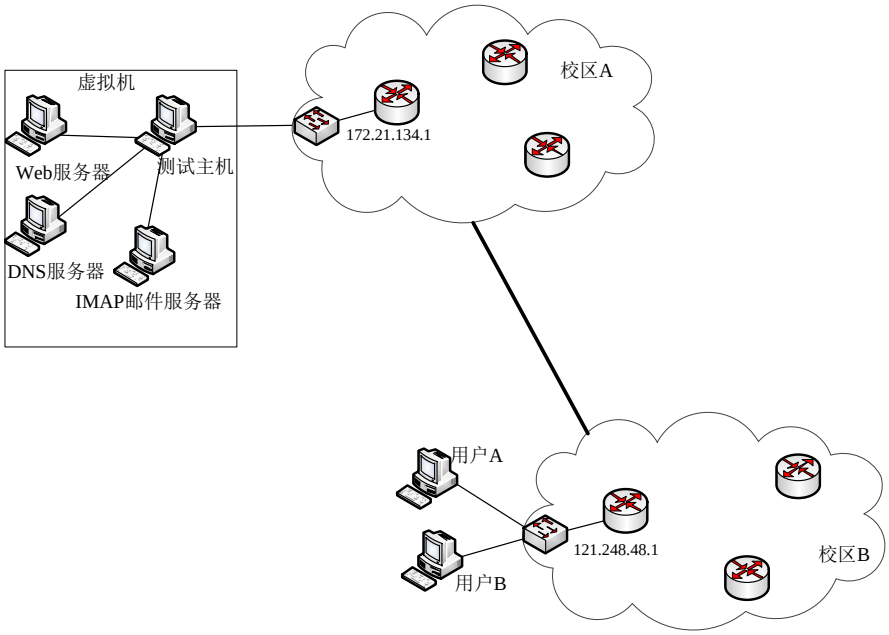

图 5-3 测试者与部分受害者位置

测试者与部分受害者位置如图 5-3 所示，测试者利用虚拟机创建了四个主机，测试主机与 Web 服务器、DNS 服务器以及邮件服务器以仅主机方式连接，而测试主机与真实的网络以桥接的方式相连，这样内部网络与外部网络就可以相互通信。下面分别介绍网络路由以及 DNS 路由注入的实现。

5.2.2.1 网络路由注入

3.1 节中介绍了网络路由注入对网络的攻击效果，并且以网页欺骗以及邮箱密码嗅探为例，介绍了它们实现的原理。为了验证网页欺骗以及邮箱密码嗅探的有效性，这里将对真实网络进行测试。

1. 网页欺骗

a) 测试场景

测试者注入网络路由，使整个网络中访问 infosec.xxx.edu.cn 的请求全部转发到测试者设置的 Web 服务器中。

b) 测试实现

测试者需要配置 Web 服务器的 IP 地址，IP 地址的值要与被测试网站的 IP 地址相同。利用 nslookup 命令可以查询 infosec.xxx.edu.cn 的 IP 地址，经查询可知 IP 地址为 202.xxx.5.32。接着，测试者需要查看此时网络中的路由表，图 5-4 显示了测试前的部分路由表，可以看到 202.xxx.5.32 这条路由的掩码为 24。当路由器发现有两条目标网络一样的路由时，它会根据最长匹配原则优先选择匹配更高的那条路由[36]，因此注入的网络路由掩码需要大于 24 位，这样路由器才会选择注恶意路由，而忽略真实的路由。根据子网掩码的划分规则，掩码的长度可以设为 25 位或 26 位，这里将 Web 服务器的接口以及对端测试主机的接口地址分别设为 202.xxx.5.32/26 以及 202.xxx.5.33/26。

为了使 202.xxx.5.0/26 这个网络影响其它路由器的路由表，需要将这个网络加入到 OSPF 协议进程中。利用协议配置接口模块，只需在 router ospf 下面添加命令 network 202.xxx.5.0/26 area 0.0.0.0。图 5-5 显示了测试后的部分路由表，可以看到多了一条恶意路由。

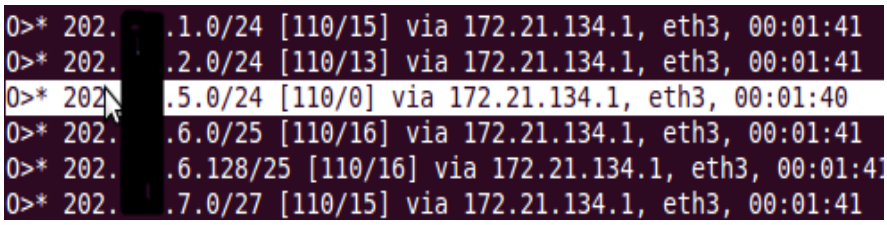

图 5-4 测试前网络的部分路由表

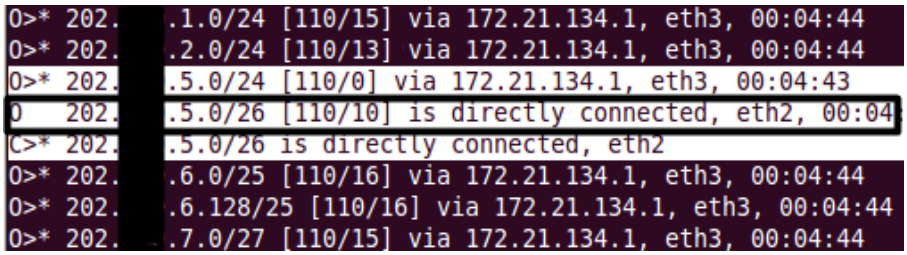

图 5-5 测试后网络的部分路由表

c) 测试结果

图 5-6 与图 5-7 显示了网络路由注入前后用户 A 测试的结果。测试后用户 A 输入域名 infosec.xxx.edu.cn，出现了非真实的页面，这里只使用了 apache2 的默认页面，测试者可以对页面进行设计，使用户相信这是一个真实的页面，这样就可以形成钓鱼攻击，获得用户的各种信息。

一般发现页面异常后，很多人使用 nslookup 或 ping 命令检查域名解析是否正确。图 5-6 与图 5-7 中的窗口 3 显示了对域名 infosec.xxx.edu.cn 解析的结果，可以看到测试前后解析的结果完全相同，此时大多数人可能认为服务器出现了异常，不会察觉到自己已被攻击。窗口 2 显示了到 infosec.xxx.edu.cn 所经历的路由器接口，从中可以看到正常情况下经历 3 跳就可以到达真实的服务器，而测试后经历了 6 跳到达恶意的服务器。测试后流量最终经过 172.21.134.2 到达恶意 Web 服务器，而 172.21.134.2 正是测试主机的接口地址。网络管理员只有了解网络的拓扑结构，并且运行了路由跟踪命令，才会发现网络受到了路由欺骗攻击。

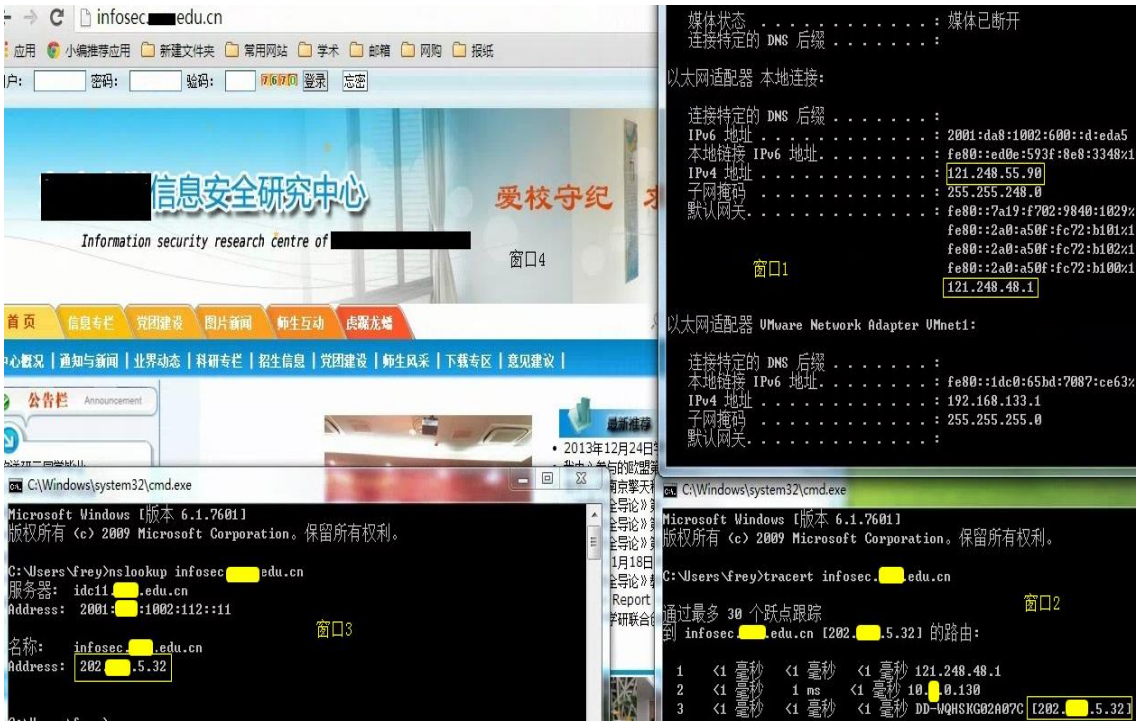

图 5-6 网络路由注入前测试结果

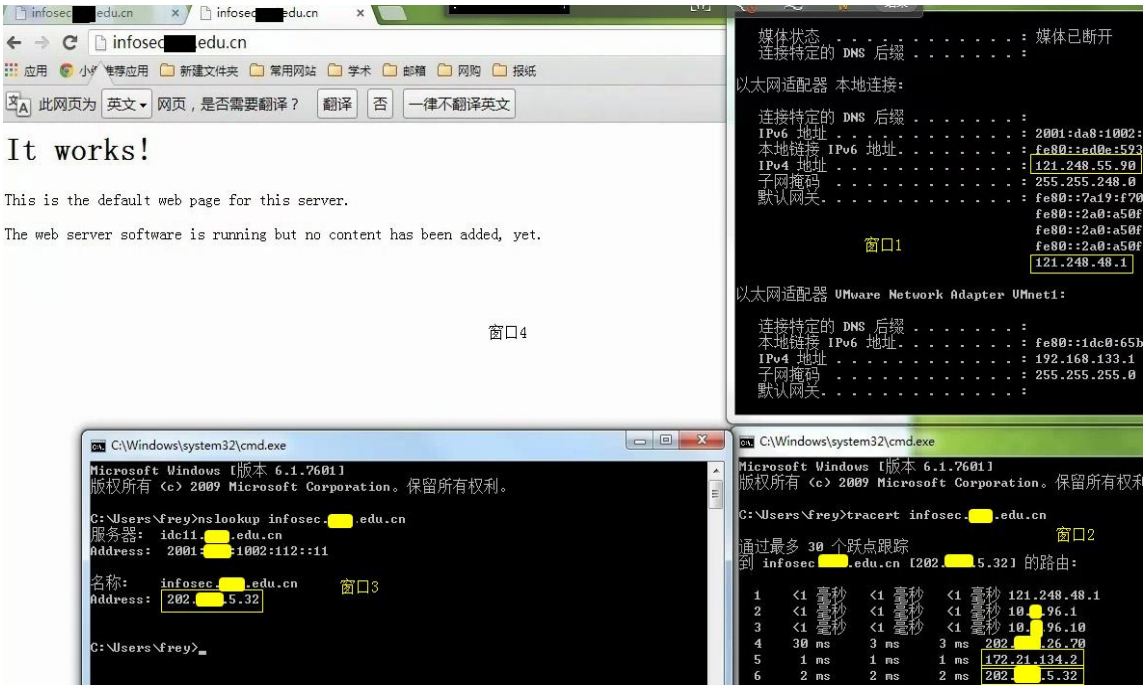

图 5-7 网络路由注入后测试结果

这种攻击方法影响的范围很大，它对整个内部网络都会产生影响，例如图 5-3 中用户 A 与测试者虽然不在同一校区内，但是他仍然受到了影响。不过，它的隐蔽性很好，由于只是针对某一注入的网段进行攻击，不会影响其它的网段，所以这种攻击不容易被察觉。

2. 邮箱密码嗅探

a) 测试场景

测试者在本地搭建了 126 的 IMAP 邮件服务器，用于嗅探整个内部网络内邮箱账号以及密码。

b) 测试实现

测试者需要配置 IMAP 邮件服务器的 IP 地址，而 IP 地址的值要与被测试服务器的 IP 地址相同。利用 nslookup 命令可以查询 imap.126.com 的 IP 地址，经查询可知 IP 地址为 121.195.178.54。由于 121.195.178.54 是外网地址，内网络由器中不会存在相应的路由项，所以对掩码的长度没有要求。如图 5-8，可以看到 126 邮箱相关域名 IP 地址最后一位的二进制表示，为了不影响用户使用 Web（mail.126.com）收发邮件，掩码长度需设为 29 位。IMAP 邮件服务器以及对端测试主机的接口地址分别设为 121.195.178.54/29 以及 121.195.178.55/29。最后，利用协议配置接口模块，在 router ospf 下面添加 network 121.195.178.48/29 area 0.0.0.0 命令，从而将网络路由宣告出去。

	128	64	32	16	8	4	2	1
121.195.178.53 (smtp.126.com)			1	1		1		1
121.195.178.54 (imap.126.com)			1	1		1	1	
121.195.178.57 (mail.126.com)			1	1	1			1
121.195.178.58			1	1	1		1	

图 5-8 子网掩码划分示意图

c) 测试结果

图 5-9 显示了测试者在 IMAP 邮件服务器上用 Wireshark 捕获数据包的结果，从中可以看到测试者获得了 172.22.138.10 主机以及 172.19.8.39 主机上用户的邮箱账号以及密码。邮件服务器不会对账号以及密码进行验证，最终会回复“Authentication failed”，这时用户会看到图 5-10 所示的窗口，认为账号与密码不匹配或服务器异常。由于测试者没有对 mail.126.com 造成影响，用户可以通过浏览器收发邮件，所以不会对网络产生很大的影响。

12.425789	172.22.138.10	121.195.178.54	TCP	netarx > imap [SYN] Seq=0 Win=64240 Len=0 MSS=1460
12.425821	121.195.178.54	172.22.138.10	TCP	imap > netarx [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
12.445837	172.22.138.10	121.195.178.54	TCP	netarx > imap [ACK] Seq=1 Ack=1 Win=64240 Len=0
12.446064	121.195.178.54	172.22.138.10	IMAP	Response: * OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID
12.465933	172.22.138.10	121.195.178.54	IMAP	Request: C1 LOGIN SunSpace[redacted]@126.com "SunSpace[redacted]04"
12.465974	121.195.178.54	172.22.138.10	TCP	imap > netarx [ACK] Seq=108 Ack=49 Win=5840 Len=0
15.536709	121.195.178.54	172.22.138.10	IMAP	Response: C1 NO [AUTHENTICATIONFAILED] Authentication failed.
15.656746	172.22.138.10	121.195.178.54	TCP	netarx > imap [ACK] Seq=49 Ack=161 Win=64080 Len=0
17.446964	172.22.138.10	121.195.178.54	TCP	netarx > imap [FIN, ACK] Seq=49 Ack=161 Win=64080 Len=0
17.485764	121.195.178.54	172.22.138.10	TCP	imap > netarx [ACK] Seq=161 Ack=50 Win=5840 Len=0
20.537722	121.195.178.54	172.22.138.10	TCP	imap > netarx [FIN, ACK] Seq=161 Ack=50 Win=5840 Len=0
20.558235	172.22.138.10	121.195.178.54	TCP	netarx > imap [ACK] Seq=50 Ack=162 Win=64080 Len=0
39.041838	121.195.178.54	172.19.8.39	IMAP	Response: * BYE Disconnected for inactivity.
39.042145	121.195.178.54	172.19.8.39	TCP	imap > hermes [FIN, ACK] Seq=37 Ack=1 Win=5840 Len=0
39.056527	172.19.8.39	121.195.178.54	TCP	hermes > imap [ACK] Seq=1 Ack=38 Win=64044 Len=0
124.065869	172.19.8.39	121.195.178.54	TCP	dssiapi > imap [SYN] Seq=0 Win=64240 Len=0 MSS=1460
124.065916	121.195.178.54	172.19.8.39	TCP	imap > dssiapi [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
124.086054	172.19.8.39	121.195.178.54	TCP	dssiapi > imap [ACK] Seq=1 Ack=1 Win=64240 Len=0
124.086343	121.195.178.54	172.19.8.39	IMAP	Response: * OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID
124.105952	172.19.8.39	121.195.178.54	IMAP	Request: C1 LOGIN xiahui[redacted]@126.com "xiahui[redacted]05"
124.105996	121.195.178.54	172.19.8.39	TCP	imap > dssiapi [ACK] Seq=108 Ack=51 Win=5840 Len=0
128.377981	121.195.178.54	172.19.8.39	IMAP	Response: C1 NO [AUTHENTICATIONFAILED] Authentication failed.
128.492408	172.19.8.39	121.195.178.54	TCP	dssiapi > imap [ACK] Seq=51 Ack=161 Win=64080 Len=0
130.373545	121.195.178.54	172.22.138.10	IMAP	Response: * BYE Disconnected for inactivity.
130.373729	121.195.178.54	172.22.138.10	TCP	imap > nsstp [FIN, ACK] Seq=37 Ack=1 Win=5840 Len=0
130.393144	172.22.138.10	121.195.178.54	TCP	nsstp > imap [ACK] Seq=1 Ack=38 Win=64044 Len=0

图 5-9 IMAP 服务器嗅探结果

图 5-10 认证失败窗口

5.2.2.2 DNS 路由注入

DNS 路由注入可以实现网络路由注入的攻击效果，它主要通过 DNS 欺骗来实现。下面介绍 DNS 欺骗在真实网络中的测试。

1. 测试场景

测试者注入 DNS 路由，用户打开 bbs.xxx.edu.cn 页面时，出现 www.xxx.edu.cn 的页面。

2. 测试实现

测试者需要配置 DNS 服务器的 IP 地址，而 IP 地址的值要与被测试 DNS 服务器的 IP 地址相同。查找相关的网络信息，可知真实的 DNS 服务器 IP 地址为 58.xxx.112.11。由于路由表中存在真实的 DNS 路由，掩码为 24 位，为了使路由器忽略真实的路由项，DNS 路由的掩码长度要大于 24 位。这里将 DNS 服务器的接口以及对端测试主机的接口地址分别设为 58.xxx.112.11/28 以及 58.xxx.112.12/28，掩码的长度为 28。利用协议配置接口模块，添加命令 network 58.xxx.112.0/28 area 0.0.0.0，将 58.xxx.112.0/28 这条路由宣告到网络中。

实现 DNS 欺骗之前，需要对 bind9 的配置文件进行设置[37]。图 5-11 显示了 xxx.edu.cn 域的正向解析配置，可以看到 www.xxx.edu.cn 与 bbs.xxx.edu.cn 具有相同的 IP 地址。

```
;
;BIND data file for [redacted].edu.cn
;
$TTL    604800
@       IN      SOA     [redacted].edu.cn. root.[redacted].edu.cn. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
        IN      NS      idc11
        IN      MX      0       dc209.[redacted].edu.cn.
@       IN      A       202.[redacted].24.12
idc11   IN      A       202.[redacted].24.12
www     IN      A       121.[redacted].63.50
bbs     IN      A       121.[redacted].63.50
h1c     IN      A       58.[redacted].112.28
my      IN      A       58.[redacted].119.39
dc209   IN      A       58.[redacted].119.209
pop3    IN      A       58.[redacted].119.209
smtp    IN      A       58.[redacted].112.17
infosec IN      A       121.[redacted].63.50
```

图 5-11 xxx.edu.cn 域的正向解析配置

3. 测试结果

图 5-12 与 5-13 显示了 DNS 路由注入前后用户 B 测试的结果。测试后用户 B 输入域名 bbs.xxx.edu.cn，却出现了 www.xxx.edu.cn 的页面。当然，可以修改 bind9 的配置，使用户看到测试者自己设计的页面，从而发起钓鱼攻击。

图 5-12 与 5-13 中窗口 2 显示了对于域名 bbs.xxx.edu.cn 的解析结果，可以看到测试前后域名的解析结果不同，发生了 DNS 欺骗。窗口 1 显示了测试前后到 DNS 服务器的路径，可以看到正常情况下经历 3 跳就可以到达真实服务器，而测试后经历了 6 跳到

达恶意服务器。一般网络管理员会认为 DNS 服务器收到了污染，然而真实的服务器并没有收到攻击，只有使用路由跟踪命令，才会发现网络受到了路由欺骗攻击。

上面的攻击只会对那些将 DNS 服务器设为 58.xxx.112.11 的用户产生影响。假如内部网络中有多个 DNS 服务器，测试者可以注入多条 DNS 路由，这样就会对所有的用户产生影响。为了使攻击不容易被发现，测试者需要将不能解析 DNS 请求转发到其它 DNS 服务器中。

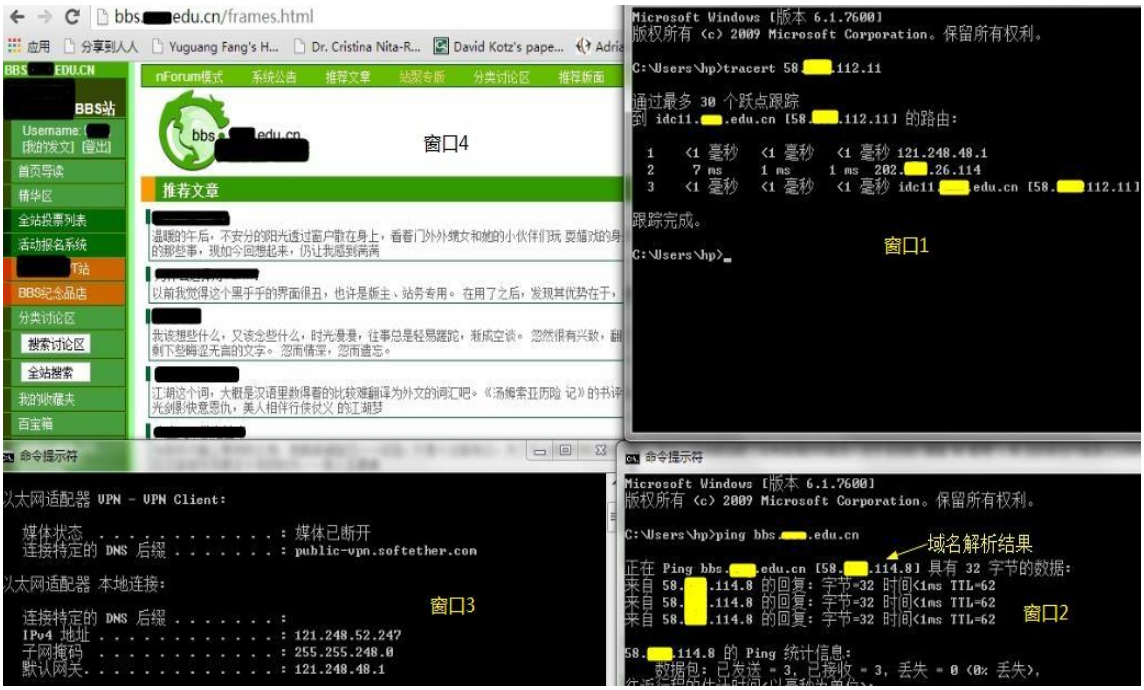

图 5-12 DNS 路由注入前测试结果

图 5-13 DNS 路由注入后测试结果

5.3 本章小结

本章利用 OSPF 渗透测试系统对真实 OSPF 网络环境进行渗透测试，检测网络的安全性。利用邻接欺骗攻击注入了网络路由和 DNS 路由，实现了网页欺骗、邮箱密码嗅探以及 DNS 欺骗这三种效果，通过测试结果可以看到这几种攻击效果对网络的威胁很大。

第6章 总结与展望

6.1 全文总结

本文主要研究了 OSPF 协议的安全性，在分析 OSPF 协议安全机制以及已有攻击方法的基础上，提出了四种新的路由欺骗攻击，并对它们的可行性及有效性进行了验证。本文同时设计实现了OSPF渗透测试系统,用该系统可以对OSPF网络进行安全性测试。

第二章中主要介绍了 OSPF 协议的原理以及运行机制，接着对 OSPF 协议的安全机制以及安全性进行了分析。

第三章中详细的介绍了这四种路由欺骗攻击。利用 GNS3 网络仿真软件、VMware 虚拟机以及真实计算机搭建网络模拟平台，并对这四种攻击进行验证。邻接欺骗攻击主要针对 OSPF 网络中边界路由器未设置为被动接口的场景，攻击者需要与邻居路由器建立邻接关系。接着介绍了如何利用该攻击实现网页欺骗、密码嗅探、中间人攻击以及 DNS 欺骗。双 LSA 远程多注入攻击主要针对攻击者已经获得了网络路由器的拓扑以及参数的场景，攻击者利用远程路由器的身份注入两个恶意的 LSA。这种攻击与 Nakibly 等人提出的双 LSA 注入攻击相比，它不仅能逃避自反击机制，而且能增大污染区域。该攻击可以实现网页欺骗、密码嗅探等效果，同时还能控制流量中间传输路径。单路径注入攻击针对的场景与双 LSA 远程多注入攻击一样，它只需要注入一条恶意的 LSA 就可以实现路由欺骗。它需要满足单路径的条件，同时可实现了流量黑洞。远程邻接欺骗攻击主要针对邻居路由器设为被动接口或攻击者不知道路由器拓扑及参数的场景，它主要利用幻影路由器与远程路由器建立邻接关系，从而注入恶意 LSA，可实现流量黑洞。

第四章设计实现了一个可对 OSPF 网络进行安全性检测的渗透测试系统，详细的介绍了系统中每一个子模块的实现。利用该渗透测试系统，对 OSPF 协议密钥认证机制的安全性进行了分析。

第五章利用 OSPF 渗透测试系统对真实的 OSPF 网络进行安全性测试，并发现网络容易受到网页欺骗、邮箱密码嗅探以及 DNS 欺骗等攻击。通过测试实验,可以看到 OSPF 网络的漏洞对网络造成的影响很大。

6.2 下一步工作展望

本文初步的分析了 OSPF 协议弱点，提出了四种的路由欺骗攻击，但是仍然有许多的不足需要改进。

1. OSPF 协议适用于不同的网络类型，本文主要研究的是广播型网络。不同网络类型下协议运行机制会有差别，在后续的工作中将对其他网络类型进行研究。
2. 本文设计的 OSPF 渗透测试系统还有许多不完善的地方，如没有实现测试报告模

块，使用户可以看到 OSPF 网络存在的问题；攻击测试模块中并没有把所有过去的攻击都集成进去等。

3. OSPF 协议只是内部网关路由协议的一种，在后续的工作中将研究更多的内部网关路由协议以及外部网关路由协议，并分析内部网络之间的安全性。

参考文献

[1] FREEBUF.COM. 网络安全威胁周报[EB/OL]. http://www.freebuf.com/news/29085.html, 2014-03-17.

[2] FreeBuf.COM, 斯诺登爆料称美国政府入侵中国网络多年[EB/OL]. http://www.freebuf.com/news/10445.html, 2013-06-13.

[3] 新华网, 信息安全成 2014 两会“热词”[EB/OL]. http://news.xinhuanet.com/fortune/2014-03/09/c_119678747.htm, 2014-03-09.

[4] 袁希群. 常见的网络攻击方法分析[J]. 福建电脑, 2011(11): 81–82.

[5] CNNIC. 2013 年中国网民信息安全状况研究报告[EB/OL]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/mtbg/201312/P020131219359905417826.pdf, 2013-09.

[6] 程佳, 王子纲. 网络欺骗技术研究[J]. 中国科技财富, 2008(06): 141+140.

[7] TECHTARGET 安全. 网络协议欺骗攻防小结[EB/OL]. http://www.searchsecurity.com.cn/showcontent_3757.htm, 2008-02-29.

[8] 黄文, 文春生, 欧红星. 基于 ICMP 的路由欺骗研究[J]. 微计算机信息, 2008(15): 98–99+143.

[9] 安德鲁. Cisco 网络黑客大曝光[M]. 2008.

[10] WU S F, CHANG H, JOU F 等人. JiNao: Design and implementation of a scalable intrusion detection system for the OSPF routing protocol[J]. Journal of Computer Networks and ISDN Systems, 1999.

[11] JONES E, MOIGNE O. OSPF security vulnerabilities analysis[J]. Work in Progress, 2006.

[12] WANG F, VETTER B, WU S. Secure routing protocols: Theory and practice[R]. Technical report, North Carolina State University, 1997.

[13] NAKIBLY G, KIRSHON A, GONIKMAN D 等人. Owning the Routing Table - New OSPF Attacks[R]. USA: Black Hat, 2011.

[14] NAKIBLY G, KIRSHON A, GONIKMAN D 等人. Persistent OSPF attacks[C]// Proceedings of the 19th Annual Network and Distributed System Security Symposium. 2012.

[15] NAKIBLY G, MENAHEM E, WAIZEL A 等人. Owning the Routing Table Part2[R]. USA: Black Hat, 2013.

[16] JEFF·多伊尔 D. Routing TCP/IP.: VolumeI[M]. 人民邮电出版社, 2003.

[17] 张海廷. 常用动态路由协议的分析及比较[J]. 电脑知识与技术, 2009(25): 7108–7109.

[18] DOYLE J, CARROLL J D. Routing TCP/IP.: TCP/IP 路由技术. 第二卷[M]. 人民邮电出版社, 2001.

[19] MOY J. OSPF Version 2, RFC 2328[J]. Internet Engineering Task Force, 1998.

[20] ALBERTH N, VON ESSEN R. Security in Internet Routing Protocols[J]. 2006.

[21] 杨晓东, 刘玉珍, 张焕国等人. OSPF 路由协议的认证分析[J]. 计算机工程与设计, 2005(01): 18–21.

[22] RIVEST R. The MD5 message-digest algorithm[J]. 1992.

[23] MANRAL V, BHATIA M, JAEGGLI J 等人. Issues with existing cryptographic protection methods for routing protocols[J]. Work in Progress, 2006.

[24] WANG F, WU S F. On the vulnerabilities and protection of OSPF routing protocol[C]//

Computer Communications and Networks, 1998. Proceedings. 7th International Conference on. IEEE, 1998: 148–152.
[25] JONES E, LE MOIGNE O. draft-ietf-rpsec-ospf-vuln-02 - OSPF Security Vulnerabilities Analysis[EB/OL]. https://tools.ietf.org/html/draft-ietf-rpsec-ospf-vuln-02, 2006-06-16.
[26] VETTER B, WANG F, WU S F. An experimental study of insider attacks for OSPF routing protocol[C]//IEEE, 1997: 293–300.
[27] MARK R. CRISPIN. INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1[EB/OL]. http://tools.ietf.org/html/rfc3501, 2003-03.
[28] 张恒伽. 基于中间人攻击的 HTTPS 协议安全性分析[D]. 上海交通大学, 2009.
[29] WANG Y, WANG J. Use GNS3 to Simulate Network Laboratory[J]. Computer Programming Skills & Maintenance, 2010, 12: 046.
[30] 顾春峰, 李伟斌, 兰秀风. 基于 VMware, GNS3 实现虚拟网络实验室[J]. 实验室研究与探索, 2012(1): 73–75.
[31] PARKHURST W R. Cisco OSPF Command and Configuration Handbook[M]. Cisco Press, 2002.
[32] BERKELEY L, NATIONAL NETWORK RESEARCH GROUP. TCPDUMP/LIBP CAP public repository[EB/OL]. http://www.tcpdump.org/, 2008.
[33] 钟锦敏. 基于中间相遇的哈希函数原像攻击[D]. 上海交通大学, 2011.
[34] JAKMA P, JARDIN V, LAMPARTER D 等人. Quagga Software Routing Suite[EB/OL] http://www.nongnu.org/quagga/, 2013.
[35] WANG P. Libnet home page[EB/OL]. http://libnet.sourceforge.net/, 2003.
[36] 崔北亮. CCNA 认证指南: 640-802[M]. 电子工业出版社, 2009.
[37] NATHAN S S, MOHAN S S, HARUDAS A R 等人. BERKELEY INTERNET NAME DOMAIN (BIND)[J]. International Journal on Cybernetics & Informatics(IJCI), 2012, 01(01).

致谢

时光如梭，转眼间两年半的研究生生活就要结束了，刚入学时的场景依然历历在目，心中非常不舍。回首在信息安全中心的生活，心中充满了无限的感激，在这里向所有关心我、帮助我的人表达最真挚的感谢。

首先，我非常感谢我的导师宋宇波老师。宋老师知识渊博、工作严谨、思维敏捷、为人随和，对待我们像朋友一样。在平时的学习和生活中，宋老师给予我非常多的帮助，使我有了克服困难的勇气。我的论文是在宋老师的悉心指导下完成的，无论从论文的选题、方案的制定以及论文的撰写，宋老师都倾注了大量的精力。当我的论文或项目遇到困难时，宋老师总会给我提出许多好的想法，帮助我拓宽思路，让我受益匪浅。真的非常幸运能够成为宋老师的学生，在此，对宋老师二年多来对我辛勤的培养表达我最诚挚的感谢和敬意！

其次，我要感谢我的学长蓝智灵，我的同门朱克龙、张志伟以及于晓文，我的师弟陈飞、浦希益，师妹顾荣荣等等，你们在学习和生活中都给予我很多的帮助；感谢 angry birds 组里的各位朋友们，你们让我的生活变成非富多彩，你们是我开心的源泉；感谢信息安全的各位朋友们，感谢橘 1A101 的各位朋友们，感谢橘 1D134 的各位舍友们，正是你们的陪伴，我的学习和生活才变得更加的充实、更加有意义。

感谢东南大学信息安全研究中心的各位老师以及在东大 7 年的任课老师们，正是有了你们的帮助，我才能不断进步。

特别感谢我的父母、爷爷奶奶以及我的亲戚，感谢我的父母二十多年来含辛茹苦的把我养育成人，你们为了让我有更好的学习环境，不辞辛苦的工作；感谢我的爷爷奶奶，从小到大你们对我非常疼爱；感谢我的亲戚姑姑、姑父、哥哥等等，你们对我的生活以及学习都很关心，对我们家也很关心。

最后，非常感谢各位评审老师百忙之中抽出时间评阅本论文！

攻读硕士学位期间的科研成果

发表的论文

1. A platform for GSM Um interface signalling observation，第一作者，2013，MINES 会议，EI 检索
2. 分布式 GSM 教学系统的设计与实现，第一作者，2013，东南大学校庆研究生学术报告会
3. Using Short Message Service (SMS)to deploy Android exploits，第三作者，2013，MINES 会议，EI 检索

软件著作权

1. GSM 教学系统教师管理平台软件，中国软件著作权，第二作者
2. GSM 教学系统学生实验平台软件，中国软件著作权，第二作者

参与的项目

1. 2012.04 – 2012.08，参与项目——分布式 GSM 教学实验平台（第一版）
2. 2012.11 – 2013.02，参与项目——分布式 GSM 教学实验平台（第二版）
3. 2013.04 – 2013.09，研究课题——路由协议安全性研究