# 第四篇 网络安全

## 第8章 TCP/IP协议栈安全

### 引言

TCP/IP协议起源于60年代末美国政府资助的ARPANET网络。当前，随着协议栈的不断演进和发展，TCP/IP协议已经成为整个网络空间正常运行的基石和基本准则。为了保持良好的开放性和高效性，TCP/IP协议在设计之初并没有很好的考虑安全性和隐私性，这也导致了针对TCP/IP协议栈的网络攻击层出不穷。

本章就TCP/IP协议栈的安全问题进行分析讲解。本概要分四个部分：首先第一部分，介绍了协议栈安全的概念、背景及当前面临的主要安全问题；第二部分，剖析了导致协议栈安全问题，层出不穷的本质及共性原因是什么，协议栈中的哪些不当设计或缺陷，使得当前的网络系统易于受到攻击破坏；接着第三部分，针对协议栈安全问题的共性原因，给出了基本的网络安全防御原理和实践原则；最后第四部分，通过介绍典型的网络安全攻防实例，贯穿本章，讲解实际场景中网络安全事件如何发生、危害是什么、产生的原因及如何防御。表1整体呈现了本章的内容及知识结构。

表8.1 协议栈安全问题概览

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **协议栈** | **网络攻击** | **攻击共性特征** | **协议栈的缺陷** | **防御** |
| 链路层 | * 帧嗅探 * ARP污染 | (1)攻击者进行身份欺骗，伪装成原通信会话的合法参与方；  (2)攻击者伪造请求，并观测、判断目标对请求的响应，推理出受害通信会话的状态信息；  (3)攻击者依据推理出的会话状态信息，伪造可被原会话接受的数据包，注入源合法数据流中，进行恶意攻击。 | (1)网络地址可伪造；  (2)系统随机化程度不够。 | (1)真实地址与真实身份；  (2)网络系统随机化； |
| 网络层 | * IP地址欺骗 * IPID误用 * IP分片误用 * ICMP误用 * 路由劫持 |
| 传输层 | * TCP劫持 * TCP DoS |
| 应用层 | * Web安全 * DNS安全 |

### 8.1协议栈安全的背景及现状

本节，我们从三个方面讲解协议栈安全问题的背景及研究现状。首先简要给出协议栈安全的基本概念及内涵，然后就协议栈安全问题研究的范畴及研究背景进行描述，最后介绍协议栈安全的研究现状及协议栈中典型的安全问题。

#### 8.1.1 协议栈安全的基本概念

遵循分层模型，当前网络系统的协议都是以栈的形式实现和部署。如图8.1所示，处于最高层的用户进程，将应用层消息封装后向下传递给传输层，传输层封装后再逐层向下传递。在协议栈对消息进行处理的过程中，每一层都完成自己特定的语义或功能，包括寻址、路由、校验等。最终，实现了远端进程对用户消息的成功接收。

但在实际的运行过程中，由于整个协议族的复杂性，协议在设计和实现过程中不可避免的会存在漏洞或缺陷。这些漏洞或缺陷，如果被恶意的攻击者利用，那么协议栈的运行就会偏离正常的预期，受攻击者的控制或破坏，造成的结果是协议栈系统的3个安全属性，即机密性、完整性、可用性不能得到保证。

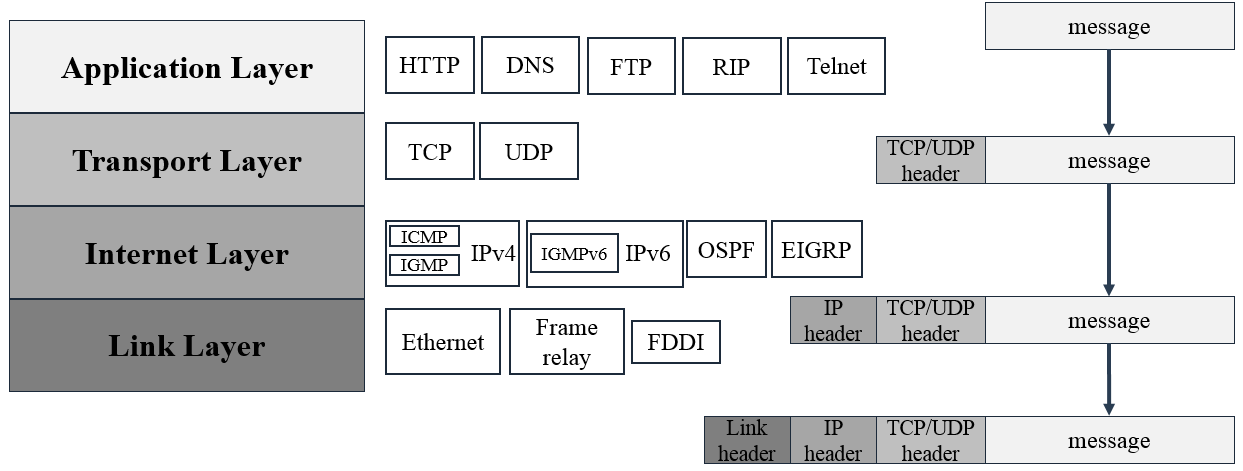


图8.1基于栈结构的协议数据处理

#### 8.1.2 协议栈安全的背景及研究范畴

从信息安全的3个基本属性出发，如果协议栈的运行破坏了信息安全的这3个属性，那么就可以判定当前的网络系统出现了安全事件。我们所讨论的安全事件，其起因是由攻击者利用了协议栈的安全漏洞或缺陷所引起的，因此本书所界定的协议栈安全问题研究，主要针对这一类问题。反之，协议栈的可靠性问题引起的安全事件，如掉电等随机故障，则不在我们讨论的范畴内。

#### 8.1.3 协议栈安全问题现状

以TCP/IP协议栈为例，协议栈在设计之初，并没有很好的考虑安全问题，因此伴随着协议栈的产生，各种各样的安全问题层出不穷。这些安全问题的出现有的是因为协议的设计缺陷所致，也有的是因为协议的不当实现或部署引入的。

典型的协议栈安全问题包括：共享信道中的帧嗅探[1]，ARP污染[2]、ICMP误用[3][4]、IP地址伪造[5][6]、IP分片误用[7]、路由劫持[8][9]、TCP劫持[10][11]等。

### 8.2协议栈安全问题的本质及原因

本节，我们从三个方面阐述分析，协议栈安全问题出现的本质和共性原因是什么。首先，结合协议栈层次模型，我们梳理归纳了不同网络层次面临的典型安全威胁。然后我们抽象总结了这些网络安全攻击的共性特征，即攻击成功发生所需要哪些基本条件和假设。最后我们分析了这些基本条件和假设，利用了网络协议栈中的哪些设计缺陷或不当实现。

#### 8.2.1 多样化的网络攻击

针对协议栈的网络攻击层出不穷，但归纳之后可以发现，当前的网络攻击基本都是基于分层模型、通过利用不同层协议的安全漏洞，对网络系统进行破坏。我们依据网络攻击所利用的安全漏洞或缺陷，对网络攻击进行分层讨论。

如图8.2所示，在TCP/IP协议栈的不同层，存在着大量的恶意攻击，可以被用来攻击破坏目标网络系统。

典型的安全威胁包括：

**1、链路层**

* 共享信道中的帧嗅探（Sniffing）：攻击者可以通过设置自己的网卡为混杂模式或监听模式，嗅探到共享信道中其他主机的数据帧[1]；
* ARP poisoning：在一个局域网中，攻击者伪造ARP应答数据包，将数据包的链路层地址填充为自己的地址，而IP地址填充为目标受害主机地址，将伪造的应答包发送到网络中，污染其他主机或网关的ARP表[2]。

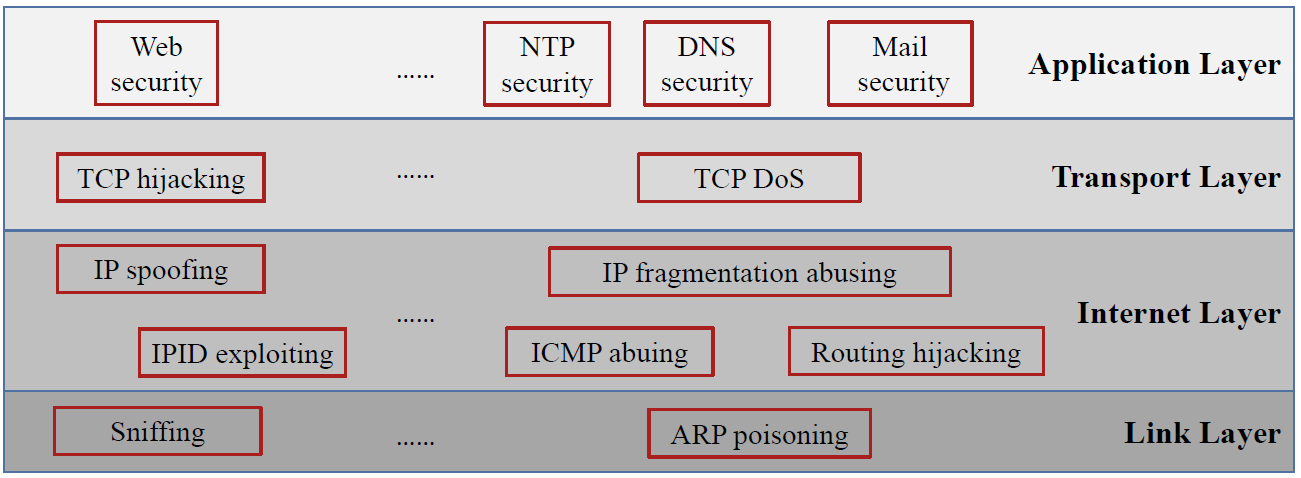


图8.2 TCP/IP协议栈中的网络攻击

**2、网络层**

* IP spoofing：攻击者可以篡改或伪造IP分组的源IP地址，冒充成其他主机发送数据包，欺骗接收方[5][6]。
* IPID exploiting：IPID是IP分组头部中的一个16位字段值，表征分组的唯一性。如果IPID分配算法不够健壮，攻击者可以通过观测受害主机的IPID，推理分析主机的流量[12][13]。
* IP fragmentation abusing：IP分片是网络层的一种机制，用于解决IP分组在不同MTU大小的网络中传输问题。但在有些场景下，攻击者可以伪造恶意的IP分片，注入到原始分片流中，污染原始流量[7]。
* ICMP abusing：攻击者可以伪装成中间路由器或其他主机，发送伪造的ICMP控制消息给目标主机，进而干扰或破坏目标主机上的原始网络数据流[3][4]。
* Routing hijacking：攻击者可以发送虚假的或伪造的路由通告消息到网络中，污染网络中其他路由器的路由表，进而实现对网络流量的路由篡改或劫持[8][9]。

**3、传输层**

* TCP hijacking：攻击者可以利用协议栈中的安全漏洞，例如随机化程度不高、静态可观测等，猜测出一个目标TCP连接的状态信息，进而伪造出可被该TCP连接接受的恶意报文，攻击破坏该连接[10][11][14]。
* TCP DoS：攻击者通过洪泛发送大量的连接请求报文，耗尽目标主机的资源[15]，或者通过欺骗TCP协议的拥塞控制机制，恶意降低目标主机的吞吐量，从而实现对目标主机的DoS攻击[16]。

**4、应用层**

应用层的攻击通常都与特定的应用程序相关，在这里我们列举2个典型应用相关的安全问题。

* Web安全：HTTP协议或者Web应用在设计实现的时候，如果存在输入审查不够等问题，可以被攻击者利用，攻击破坏Web程序的安全性，如跨站脚本攻击XSS[17]，SQL注入[18]，跨站请求伪造[19]等。
* DNS安全：DNS协议是基于UDP协议工作的，如果DNS协议在实现过程中，存在随机化程度不高，其端口号和事务号容易被攻击者猜测，那么攻击者就可以伪装成一个上游DNS服务器，发送伪造的应答报文，污染下游DNS服务器的缓存[20][21]。

#### 8.2.2 网络攻击的共性特征

通过梳理分析上述的针对网络协议栈的攻击，可以发现这些攻击基本都具备2个共性特征。一是攻击者可以进行身份欺骗，伪装成网络通信的一端，二是攻击者可以进行推理猜测，成功构造出可被通信对端接受的数据报文。这二者也构成了整个网络攻击的关键2步，前者为后者提供基础攻击能力和条件，后者在前者的基础之上，注入数据、实现攻击目标。

**1、攻击者可以进行身份欺骗**

在进行网络攻击时，一个恶意的攻击者要想攻击破坏原始的网络数据流，首先需要伪装成通信的一端，欺骗对端。例如在ARP污染、DNS劫持、TCP劫持、IP分片污染等攻击中，攻击者都需要伪装成原始网络会话的参与方，进行身份欺骗，只有这样，攻击者伪造的报文才有可能被接受。

**2、攻击者可以推理构造出目标数据**

要成功实施网络攻击，仅仅实现身份欺骗通常是不够的。在身份欺骗之后，还需具备另一个共性特征，即攻击者能够推理、猜测出目标数据，构造出合法的报文，通过检查、被原始数据流的通信对端所接受。这一能力特征是在前一特征基础之上，更高一级的共性能力特征。比如在IP分片攻击中，攻击者在伪装成网络会话一端之后，还要能推理出分片的偏移、ID、校验和等取值，才能成功实施分片注入攻击。同理在针对TCP、DNS等的劫持攻击中，攻击者在身份欺骗成功后，还需能猜测、推理出报文的端口号、序列号等取值。攻击者只有成功猜测、推理出原始数据流的这些取值，才能构造出可被接受的报文数据，成功注入到通信对端，攻击网络系统。

#### 8.2.3 协议栈中的不当设计和实现

协议栈的网络攻击之所以能成功，其2个共性特征映射到实际的网络系统中，本质上是利用了当前协议栈中的2个基础安全缺陷。一是网络地址缺乏足够的真实性验证，可以被恶意伪造；二是网络系统在实现和部署过程中，随机化程度不高，致使网络的状态信息可被恶意攻击者预测推理。

**（1）网络地址可伪造，缺乏合法性验证**

当前的TCP/IP协议栈，在设计之初，并没有对网络地址、尤其是IP地址的真实性进行有效保护，也就是说如果网络系统没有部署新的安全防御机制，那么网络中的主机在发送IP分组的时候，分组的源IP地址是可以任意指定的。这就导致了攻击者可以伪装成其他主机，冒充该主机发送报文，实现身份欺骗。IP地址伪造，是当前网络协议栈各种安全问题出现的根本原因之一，梳理可以发现，8.2.1节列举的各个网络攻击，大部分都利用了协议栈中的这一基础缺陷。

**（2）网络状态信息可预测推理，缺乏足够的随机化**

在网络系统中，理论上网络的状态信息，如一个网络会话的ID、分组序号、流量大小、时间戳等信息，对于会话参与方以外的实体（如一个攻击者），应该是不可预知的，即攻击者不能借助某些资源或条件，猜测、推理出一个非自己所属的网络会话的状态信息。而当前的协议栈实现，缺乏足够的随机化，例如socket端口号的顺序分配、TCP协议中共享变量的使用、IP协议中分组ID的顺序分配、分组大小的恒定、报文填充字段的生成算法模式固定等。这些静态可猜测的特征，导致网络系统的随机化程度不高，从而允许攻击者可以成功的推理、构造出一个恶意报文，顺利通过原始报文流的合法性检查，被接收端所接受，最终攻击破坏网络系统。

### 8.3协议栈安全的基本防御原理

本节，针对协议栈安全问题的本质及共性原因，我们给出了协议栈安全防御的基本原理和实践规范。首先针对地址哄骗这一共性网络攻击条件，我们给出真实地址/真实身份这一网络安全防御的基本准则。然后，我们提出，增强和提高协议栈的随机化属性，避免网络状态信息被恶意攻击者观测利用，攻击破坏网络系统。最后我们简要介绍了已有的协议栈安全防御标准和技术手段。

#### 8.3.1 基于真实地址的网络安全防御

基于真实地址的网络安全防御，旨在将网络地址和网络会话参与方的真实身份相耦合，避免攻击者通过假冒源地址，伪装成其他主机，进行恶意欺骗。在具体实现层面，当前已经有一些标准规范推出[22][23]， 约束主机只能使用预分配给自己的IP地址发送数据，进行网络通信。如果主机（恶意攻击者）篡改自己出口数据包的源IP地址，通常网关会进行过滤和丢弃。这种安全机制，极大的缓解了当前网络空间中的各种安全问题，因为如我们前面所述，攻击者不能伪装成通信会话的参与方，因此攻击者伪造的恶意报文，也就无法成功的“参与”到原始网络会话数据流中去。

#### 8.3.2 增强协议栈随机化属性

除了通过部署真实地址机制，阻止攻击者进行身份欺骗、挫败网络攻击之外，另一方面也可以通过增强网络协议栈的随机化属性，提升攻击者猜测、推理出目标网络状态信息的难度，迫使攻击者不能在有效的时间窗口内，构造出可被接受的报文，导致攻击失败。当前已有一些工作，在这一防御理念之下开展研究，代表性的包括移动目标防御（Moving Target Defense，MTD）[24][25]，拟态防御等[26]。其核心思想是在保持网络语义功能完整性的前提下，通过随机、动态的变换网络特征，致使网络系统呈现给攻击者的是一个不断动态变化的攻击面，提升系统的攻击猜测难度。

#### 8.3.3 安全防御实践及规范

在实践中，网络安全防御的整体性原则是要求在网络发生被攻击、破坏的情况下，必须尽可能地快速恢复网络的服务，减少损失。同时在网络系统各个点上部署安全防御措施，避免出现安全的木桶效应。因此，网络安全防御系统的设计与实现，通常遵循的原则包括：最小权限原则、纵深防御原则、防御多样性原则、防御整体性原则、安全性与代价平衡原则、网络资源的等级性原则。

在构建安全防御体系时，应考虑构建安全防护机制、安全检测机制和安全响应机制。安全防护机制是根据具体系统存在的各种安全威胁采取的相应的防护措施，避免非法攻击的进行。安全检测机制是检测系统的运行情况，及时发现和制止对系统进行的各种攻击。安全响应机制是在安全防护机制失效的情况下，进行应急处理。

### 8.4典型案例分析

本节，我们通过三个实际的网络安全案例，贯穿本章内容，讲解了在典型的网络场景中，攻击者如何利用协议栈中的安全缺陷，攻击破坏网络系统，同时也分析了攻击能够成功的本质原因是什么，以及如何进行安全防御。

#### 8.4.1 误用IP分片机制污染UDP协议

IP分片是互联网最基本的一种报文转换处理机制，但这种机制却被经常误用，进行恶意攻击。攻击者可以通过地址伪造，伪装成数据流的源端，构造恶意的分片，注入到数据流中，迫使接收端进行错误重组、污染攻击接收端。

IP分片攻击目前主要应用在基于UDP的通信场景下，主要原因在于UDP不具备路径MTU发现能力（Path Maximum Transmission Unit Discovery，PMTUD）[27]，因此当传输路径上出现MTU不一致时，UDP报文不可避免的会被分片。攻击者就可以利用IP分片，攻击UDP协议。最典型的一种攻击就是，攻击者伪装成权威域名服务器，利用IP分片，污染攻击递归域名服务器的DNS缓存，如图8.3所示。攻击者伪装成权威服务器，伪造一个恶意分片，发送给递归服务器，该恶意分片和合法的权威服务器分片，在接收端递归服务器处、错误重组，实现了DNS缓存污染。

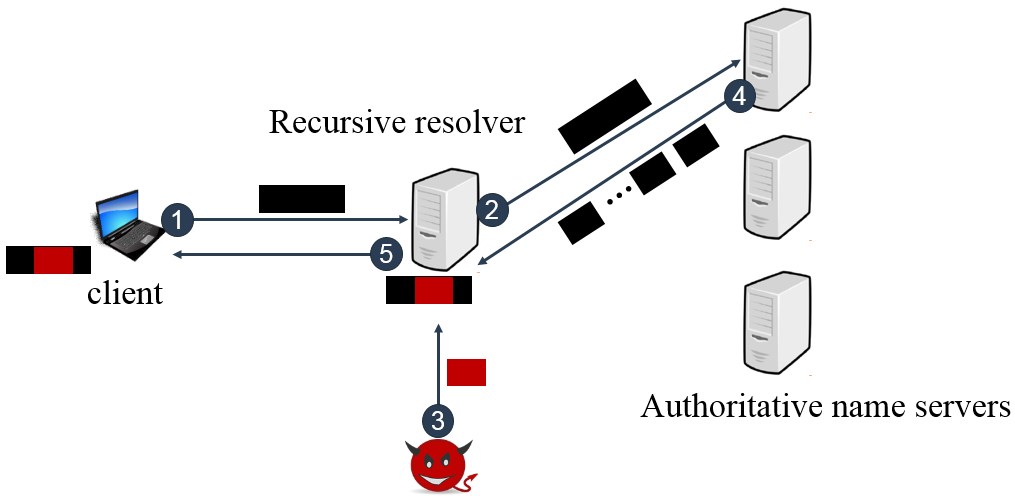


图8.3 利用IP分片实现DNS缓存污染攻击

分析上述攻击过程可以发现，攻击之所以能成功的2个必要条件如下，一是攻击者可以进行地址伪造，伪装成权威服务器，欺骗递归服务器；二是攻击者可以推理、猜测出网络状态信息，包括伪造的分片的IPID，校验和、偏移等。相应的，IP分片攻击的防御也可以从这两方面入手。

#### 8.4.2 伪造源IP地址进行DDoS攻击

网络中另一种常见的攻击是分布式拒绝服务攻击（Distributed denial of service，DDoS）[28]。如图8.4所示，在这种攻击中，攻击者通过控制大量僵尸节点，向一个选定的目标服务器发送大量请求，消耗目标服务器的资源，包括带宽、CPU、存储等，迫使服务器不能向正常的网络用户提供服务，导致出现拒绝服务的效果。

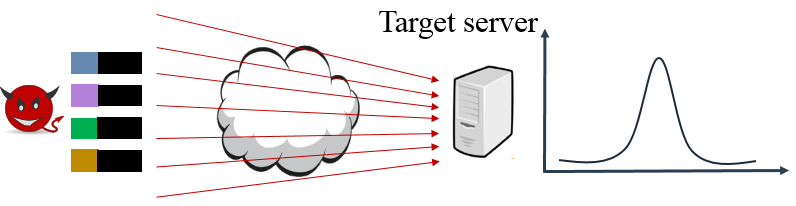


图8.4 基于伪造源IP地址的DDoS攻击

在这种攻击中，攻击者为了隐藏自己的攻击行为，通常会篡改或伪造发往目标服务器的数据包源IP地址，避免自己真正的节点主机被服务器端的安全机制追踪或审计到。当前已有一些针对DDoS攻击的防御手段，但普遍而言，DDoS攻击仍然是当前互联网上最严重的威胁之一。

#### 8.4.3 TCP连接劫持攻击

TCP连接是由一个四元组标识的，即[源IP地址，源端口号，目的IP地址，目的端口号]，如果一个恶意的攻击者能够识别出这样的一个四元组，那么就表示攻击者检测到一个活TCP连接。

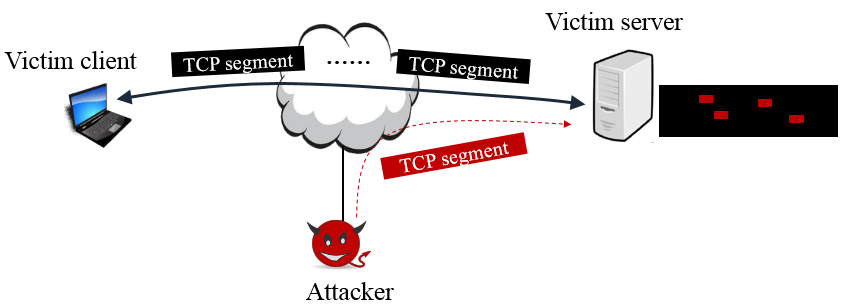


图8.5 TCP连接劫持

如图8.5所示，如果攻击者在检测出一个TCP连接后，还能够猜出该连接上的序列号（Sequence number）和确认号（Acknowledgement number），那么攻击者就可以伪装成连接的一端（比如客户端），伪造一个TCP报文，里面嵌入自己猜出的连接信息，将报文发送给服务器，欺骗服务器误以为该报文是由客户端发来的，错误接收该报文，最终受到攻击。

通常一个路径外的攻击者，要成功实施上述攻击，也是需要具备两个能力，一是攻击者可以伪装身份，通过伪造源IP地址，伪装成TCP连接的一端，然后攻击另一端；二是攻击者需要推理猜出TCP连接的状态信息，如序列号等，注入伪造报文被原始TCP流接受。由于当前网络系统的随机化程度不高，攻击者可以找到很多可利用的条件，如网络系统的侧信道[29][30]，来观测推理目标连接状态信息，进行攻击。对应的，对TCP攻击的防御，也可以从上述两个方面着手。

### 8.5总结

本章我们对TCP/IP协议栈的安全问题进行了阐述分析。首先我们介绍了协议栈安全的基本概念、背景及当前面临的主要安全问题；然后我们剖析归纳了协议栈安全问题的本质及共性原因，我们从三个方面进行了阐述，包括列举多样化的网络攻击、抽象提取网络攻击的共性特征、以及分析协议栈中的不当设计；然后针对协议栈安全的本质原因，我们给出了基本的防御原理；最后我们通过三个典型攻击场景和案例，具体讲解了在实际场景中，网络安全事件如何发生及防御。

未来，随着多样化网络场景和应用的不断涌现，基础的TCP/IP协议将面临更多的挑战，不仅局限在与性能相关的高效传输方面，与安全性相关的威胁挑战也值得更进一步的深入研究，尤其是在多应用场景下，不同层协议之间复杂交互过程中，引入的安全问题，将是未来协议栈安全研究的主要方向之一。

### 参考文献

1. Ansari, Sabeel, S. G. Rajeev, and H. S. Chandrashekar. "Packet sniffing: a brief introduction." *IEEE potentials* 21.5 (2003): 17-19.
2. Tripathi, Nikhil, and B. M. Mehtre. "Analysis of various ARP poisoning mitigation techniques: A comparison." *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies* (*ICCICCT*). IEEE, 2014.
3. Gupta, Neha, et al. "DDoS attack algorithm using ICMP flood." *2016 3rd International Conference on Computing for Sustainable Global Development* (*INDIACom*). IEEE, 2016.
4. Gont, Fernando. *ICMP attacks against TCP*. RFC 5927, July, 2010.
5. Duan, Zhenhai, Xin Yuan, and Jaideep Chandrashekar. "Controlling IP spoofing through interdomain packet filters." *IEEE transactions on Dependable and Secure computing* 5.1 (2008): 22-36.
6. Sahu, Mridu, Rainey C. Lal, and Chhattisgarh Bhilai. "Controlling IP Spoofing Through Packet Filtering." *International Journal of Computer Technology and Applications* 3.1 (2012).
7. Herzberg, Amir, and Haya Shulman. "Fragmentation considered poisonous, or: One-domain-to-rule-them-all. org." *2013 IEEE Conference on Communications and Network Security* (*CNS*). IEEE, 2013.
8. Nakibly, Gabi, Adi Sosnovich, Eitan Menahem, Ariel Waizel, and Yuval Elovici. "OSPF vulnerability to persistent poisoning attacks: a systematic analysis." In *Proceedings of the 30th Annual Computer Security Applications Conference*, pp. 336-345. 2014.
9. Butler, K., Farley, T. R., McDaniel, P., & Rexford, J. (2009). A survey of BGP security issues and solutions. *Proceedings of the IEEE*, *98*(1), 100-122.
10. Feng, X., Fu, C., Li, Q., Sun, K., & Xu, K. (2020, October). Off-Path TCP Exploits of the Mixed IPID Assignment. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1323-1335).
11. Chen, W., & Qian, Z. (2018). Off-path TCP exploit: How wireless routers can jeopardize your secrets. In *27th USENIX Security Symposium* (*USENIX Security’18*) (pp. 1581-1598).
12. Knockel, J., & Crandall, J. R. (2014). Counting packets sent between arbitrary internet hosts. In *4th USENIX Workshop on Free and Open Communications on the Internet* (*FOCI* 14).
13. Pearce, P., Ensafi, R., Li, F., Feamster, N., & Paxson, V. (2017, May). Augur: Internet-wide detection of connectivity disruptions. In *2017 IEEE Symposium on Security and Privacy* (*SP*) (pp. 427-443). IEEE.
14. Cao, Y., Qian, Z., Wang, Z., Dao, T., Krishnamurthy, S.V. and Marvel, L.M., 2016. Off-Path TCP Exploits: Global Rate Limit Considered Dangerous. In *25th USENIX Security Symposium* (*USENIX Security 16*) (pp. 209-225).
15. Wang, H., Zhang, D., & Shin, K. G. (2002, June). Detecting SYN flooding attacks. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies* (Vol. 3, pp. 1530-1539). IEEE.
16. Kuzmanovic, Aleksandar, and Edward W. Knightly. "Low-rate TCP-targeted denial of service attacks: the shrew vs. the mice and elephants." In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 75-86. 2003.
17. Grossman J, Fogie S, Hansen R, Rager A, Petkov PD. XSS attacks: cross site scripting exploits and defense. Syngress; 2007 May 23.
18. Clarke-Salt J. SQL injection attacks and defense. Elsevier; 2009 Jun 16.
19. Kombade, Rupali D., and B. B. Meshram. "CSRF vulnerabilities and defensive techniques." *International Journal of Computer Network and Information Security* 4, no. 1 (2012): 31.
20. Man, Keyu, Zhiyun Qian, Zhongjie Wang, Xiaofeng Zheng, Youjun Huang, and Haixin Duan. "DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels." In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1337-1350. 2020.
21. Berger, Harel, Amit Z. Dvir, and Moti Geva. "A wrinkle in time: a case study in DNS poisoning." *International Journal of Information Security* (2020): 1-17.
22. Wu. J, Bi. J, X. Li, G. Ren, K. Xu, and M. Williams. " a source address validation architecture (SAVA) testbed and deployment experience." RFC5210. 2008 Jun.
23. Wu. J, Bi. J, Bagnulo M, Baker F, Vogt C. “Source address validation improvement (SAVI) framework.” RFC7039. 2013 Oct.
24. Jajodia, Sushil, Anup K. Ghosh, Vipin Swarup, Cliff Wang, and X. Sean Wang, eds. *Moving target defense: creating asymmetric uncertainty for cyber threats*. Vol. 54. Springer Science & Business Media, 2011.
25. Lei, C., Zhang, H. Q., Tan, J. L., Zhang, Y. C., & Liu, X. H. Moving target defense techniques: A survey. *Security and Communication Networks*, *2018*.
26. 邬江兴. "网络空间拟态防御研究." 信息安全学报 . 4 (2016): 1-10.
27. Mogul, J. C., and S. E. Deering. "RFC1191: Path MTU discovery." (1990).
28. Zargar, Saman Taghavi, James Joshi, and David Tipper. "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks." *IEEE communications surveys & tutorials* 15, no. 4 (2013): 2046-2069.
29. Feng, X., Fu, C., Li, Q., Sun, K., & Xu, K. (2020, October). Off-Path TCP Exploits of the Mixed IPID Assignment. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1323-1335).
30. Cao, Yue, Zhiyun Qian, Zhongjie Wang, Tuan Dao, Srikanth V. Krishnamurthy, and Lisa M. Marvel. "Off-path tcp exploits of the challenge ack global rate limit." *IEEE/ACM Transactions on Networking* 26, no. 2 (2018): 765-778.

### 附录

章节实验：《基于IPID侧信道的TCP连接恶意阻断》

#### 实验目的：

IPID是IP分组头部中的一个16 bit字段，IPID的分配，不同的操作系统有着不同的算法实现。但如果算法缺乏健壮性，不够安全，产生分配的IPID容易被攻击者猜测到的话，IPID很容易成为一个侧信道漏洞，被攻击者利用攻击破坏系统。比如，Windows 7及之前的Windows操作系统版本上，系统采用一个全局计数器为每一个出去的IP分组分配ID。这种分配方式是不安全的，攻击者可以观测目标系统的IPID，推理出系统的网络状态，比如猜测系统上TCP连接的序列号，恶意阻断TCP连接，进而破坏系统。

通过复现基于全局IPID计数器的TCP连接劫持攻击场景，使同学们学习、了解到网络协议栈中，常见的安全威胁及安全漏洞，并掌握漏洞利用过程及原理，从而加深对网络安全问题的认识，并了解相应的安全防御手段。

#### 实验环境设置：

本实验的环境设置如下：

（1） 一台装有Window 7操作系统的主机，充当TCP 服务器端；

（2） 一台客户端机器 Client，操作系统不限，请求TCP服务器，建立连接；

（3） 一台攻击者机器 Attacker （Kali 虚拟机）；

三台机器之间网络互通，但Attacker不在Server和Client的路径上，不能监听嗅探Server和Client之间的流量。Attacker的目标就是通过利用Server上的IPID侧信道，推理出Server和Client之间TCP连接的序列号，从而恶意阻断TCP连接（充当TCP连接的一端，向连接中注入RST报文）。

#### 实验步骤：

整个实验过程及步骤如下：

（1） Client向Server请求建立一条TCP连接，二者能正常通信；

（2） Attacker向Server发送ping请求，查看Server响应的reply报文的IPID；

（3） Attacker伪造源地址，冒充client向Server发送一个TCP报文，里面嵌入了自己猜测的连接序列号（假定源端口号已知）；

（4） 如果嵌入的信息正确，Server会发送响应TCP报文给Client，该TCP报文会“消耗”Server端的全局IPID计数器；如果猜测不正确，Server会丢弃报文，Server端的全局IPID计数器不会变化；

（5） 攻击者再次ping server，查看reply报文的IPID是否好上次连续，如果连续则说明刚才的猜测失败，如果不连续，则说明刚才的猜测成功，触发了Server产生了响应TCP报文；

（6） 重复步骤（2）-（5）直到猜出正确的序列号；

（7） 伪装成TCP连接的一端，向目标连接中注入伪造的RST报文，恶意阻断该TCP连接。

#### 预期实验结果

攻击成功后，Client到Server之间的TCP连接被恶意阻断，二者不能正常通信。