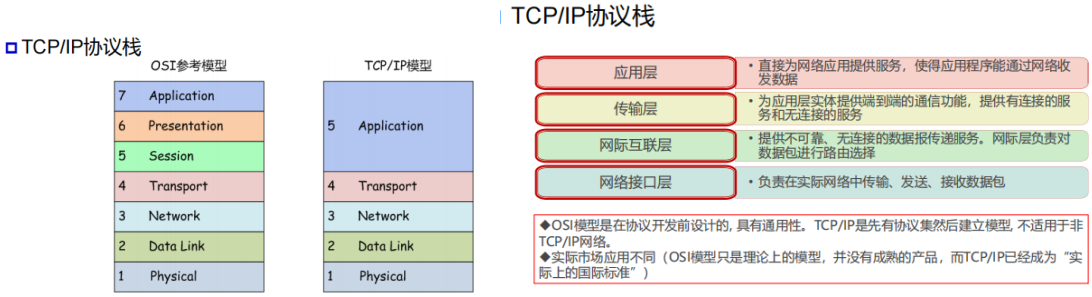
1. **网络攻防的基本概念** 1.1 **网络安全的属性** **网络空间安全**(Cybersecurity)： 研究网络空间中的安全威胁和防护问题。包括网络空间基础设施的安全和可信，以及网络空间信息的保密性、完整性、可用性、真实性和可控性的相关理论与技术。 **核心属性**（CIA)： **保密性**(Confidentiality)：确保隐私或者秘密信息不向非授权者泄漏，也不被非授权者使用，即：防止数据的未授权访问。 **完整性**（Integrity)：确保信息只能以特定和授权的方式进行改变，比如：确保接收者收到的消息就是发送者发送的消息。 **数据完整性**：确保信息和程序仅以授权方式进行变动 **系统完整性**：确保系统以非损害方式完成设计的功能，免受有意或者无意对系统的非授权操作。 **可用性**（Availability)：合法用户在需要使用网络资源的时候，能够获得正常的服务。 **可控性**：限制对网络资源（软件和硬件）和数据（存储和通信的数据）的访问，其目标是防止未授权使用资源、未授权公开或者修改数据。通过访问控制实现。 **不可否认性**（Non-reputation)：通信实体不能对自己做过的事情抵赖，包括两层含义，一方面发送者不能否认自己发送数据的行为；另一方面，接收者不能否认自己接收过数据。 **真实性**（authenticity)：一个实体是其所声称实体的这种特性。 **可靠性**（reliability)：与预期行为和结果一致的特性。 1.2 **典型网络安全事件：Morris worm（蠕虫)、Mirai botnet（DDos)、Stuxnet（震网蠕虫病毒)、WannaCry Ransomware（勒索病毒)、FireEye Redteam tools（泄露事件)、 Solarwinds（供应链攻击事件）** **病毒、蠕虫、木马**：病毒、蠕虫和木马是可导致计算机和计算机上的信息损坏的恶意程序。这三种东西都是人为编制出的恶意代码，都会对用户造成危害。 **计算机病毒**(Computer Virus)，根据《中华人民共和国计算机信息系统安全保护条例》，病毒的明确定义是“指编制或者在计算机程序中插入的破坏计算机功能或者破坏数据，影响计算机使用并且能够自我复制的一组计算机指令或者程序代码”。病毒往往具有很强的感染性、潜伏性、特定的触发性、破坏性。 **病毒必须满足两个条件：自行执行**：它通常将自己的代码置于另一个程序的执行路径中。**自我复制**：例如，它可能用受病毒感染的文件副本替换其他可执行文件。病毒既可以感染个人计算机也可以感染网络服务器。 **蠕虫(worm)病毒**是一种常见的计算机病毒，它利用网络（网络、电子邮件等）进行复制和传播。蠕虫病毒是自包含的程序（或是一套程序)，它能传播自身功能的拷贝或自身的某些部分到其他的计算机系统中（通常是经过网络连接)。(典型蠕虫病毒：震网病毒、勒索病毒） **普通病毒与蠕虫病毒的区别**：复制方式上，普通病毒需要传播受感染的驻留文件来进行复制，而蠕虫不使用驻留文件即可在系统之间进行自我复制；传染目标上，普通病毒的传染能力主要是针对计算机内的文件系统而言，而蠕虫病毒的传染目标是互联网内的所有计算机。 **木马(Trojan Horse)，**是指那些表面上是有用的软件、实际目的却是危害计算机安全并导致严重破坏的计算机程序。(是一种用于远程控制的黑客工具、具有隐蔽性、具有非授权性） **隐蔽性**是指木马的设计者为了防止木马被发现，会采用多种手段隐藏木马，这样服务端即使发现感染了木马，也难以确定其具体位置。**非授权性**是指一旦控制端与服务端连接后，控制端将窃取到服务端的很多操作权限，如修改文件、修改注册表、控制鼠标、键盘、窃取信息等等。 **木马与病毒的区别**：木马不具传染性，它并不能像病毒那样复制自身，也并不“刻意”地去感染其他文件，它主要通过将自身伪装起来，吸引用户下载执行。木马一般主要以窃取用户相关信息或隐蔽性控制为主要目的，相对病毒而言，可以简单地说，病毒破坏你的信息，而木马窥视你。 **软件漏洞 网络系统的脆弱性**：指系统的硬件资源、通信资源、软件及信息资源等存在的弱点和缺陷。包括硬件系统的弱点和缺陷、软件系统的弱点和缺陷、网络和通信协议的弱点和缺陷、使用者的弱点 **软件安全漏洞**：一台计算机是由硬件以及软件两个部分组成，最基本的软件就是操作系统。 计算机软件是由计算机程序员开发出来的。不同程序员的编程水平不一样，就会造成软件存在这样或者那样的问题，这些问题可能会造成软件崩溃不能运行，我们称这些问题为**软件缺陷（Bug）** 软件中存在的一些问题可以在某种情况下被利用来对用户造成恶意攻击，如给用户计算机上安装木马病毒，或者直接盗取用户计算机上的秘密信息，等等。这个时候，软件的这些问题就不再只是 Bug，而是一个软件安全漏洞，简称**“软件漏洞”** 电脑肉鸡：也就是受别人控制的远程电脑。肉鸡可以是各种系统,如 windows,linux,unix 等；更可以是一家公司、企业、学校甚至是政府军队的服务器。如果服务器软件存在安全漏洞，攻击者可以发起“主动”进攻，植入木马，将该服务器变为一个任人宰割的“肉鸡” **漏洞产生的原因** 1. 小作坊式的软件开发：质量参差不齐 2. 赶进度带来的弊端：投机取巧或者省工省料的办法来开发软件 3. 被轻视的软件安全测试：

功能为上，测试为下 4. 淡薄的安全思想：缺乏安全开发的意识和经验 5. 不完善的安全维护：不重视安全维护，不重视漏洞修复 **漏洞分类：**一个漏洞从被攻击者发现并利用，到被厂商截获并发布补丁，再到补丁被大多数用户安装导致漏洞失去了利用价值，一般都要经历一个完整的生命周期。按照漏洞生命周 期的阶段进行分类的方法包括三种： **0-day 漏洞：**指还处于未公开状态的漏洞。这类漏洞只在攻击者个人或者小范围黑客团体内使用，网络用户和厂商都不知情，因此没有任何防范手段，危害非常大。(0-day 漏洞也是当前网络战中的核武器） **1-day 漏洞：**原义是指补丁发布在 1 天内的漏洞，不过通常指发布补丁时间不长的漏洞。由于了解此漏洞并且安装补丁的人还不多，这种漏洞仍然存在一定的危害。 **n-day 漏洞/已公开漏洞：**已公开漏洞是指厂商已经发布补丁或修补方法，大多数用户都已打过补丁的漏洞。这类漏洞从技术上因为已经有防范手段，并且大部分用户已经进行了修补，危害比较小。 电脑肉鸡是被别人控制的远程电脑。将大量服务器沦为肉鸡，主要依赖于**软件漏洞、木马**。 **漏洞库：**大量软件漏洞需要一个统一的命名和管理规范，以便开展针对软件漏洞的研究，提升漏洞的检测水平，并为软件使用者和厂商提供有关软件漏洞的确切信息。多个机构和相关国家建立了漏洞数据库，这些数据库分为公开的和某些组织机构私有的不公开数据库。公开的数据库包括 CVE、NVD、BugTraq、CNNVD、CNVD 等。 通过漏洞信息数据库，可以找到操作系统和应用程序的特定版本所包含的漏洞信息，甚至针对某些漏洞的专家建议、修复办法和专门的补丁程序。极少的漏洞库还提供检测、测试漏洞的 POC。POC（proof-of-concepts，为观点提供证据）: 样本验证代码。 目前，许多国家建立了针对漏洞的应急响应机构，例如美国计算机应急反应小组 US-CERT，中国的国家互联网应急中心 CNCERT/CC。他们是软件漏洞数据的主要提供者或者漏洞库的主要维护者，并且提供了高风险的漏洞警报和专家建议。 **美国国家漏洞数据库 NVD**（National Vulnerabilities Database）同时收录三个漏洞数据库的信息，CVE 漏洞公告、US-CERT 漏洞公告、USCERT 安全警告，也自己发布漏洞公告和安全警告，是目前世界上数据量最大，条目最多的漏洞数据库之一。 **（美国）通用漏洞列表 CVE(Common Vulnerabilities andExposures)相当于软件漏洞的一个行业标准。**它**实现了安全漏洞命名机制的规范化和标准化**，为每个漏洞确定了唯一的名称和标准化的描述，为不同漏洞库之间的信息录入及数据交换提供了统一的标识，使不同的漏洞库和安全工具更容易共享数据，成为评价相应入侵检测和漏洞扫描等工具和数据库的基准。**中国国家信息安全漏洞库 CNNVD**（China National Vulnerability Database of Information Security）隶属于中国信息安全测评中心，是中国信息安全测评中心为切实履行漏洞分析和风险评估的职能，负责建设运维的国家级信息安全漏洞库，为我国信息安全保障提供基础服务。 **国家信息安全漏洞共享平台 CNVD**（China National VulnerabilityDatabase）是由 CNCERT/CC（国家计算机网络应急技术处理协调中心）联合国内重要信息系统单位、基础电信运营商、网络安全厂商建立的信息安全漏洞信息共享知识库，致力于建立国家统一的信息安全漏洞收集、发布、验证、分析等应急处理体系。 **其他**：EDB 漏洞库、微软安全公告板和微软安全建议、绿盟科技的中文安全漏洞库、启明星辰的中文安全公告库 **术语解释**：YARA：一款 VirusTotal 开发的用于恶意软件识别和分类的规则匹配工具。; Snort：一款历史悠久的开源网络入侵检测系统。; IOC：失陷指标（Indicator Of Compromise），即攻击者控制被害主机所使用的远程命令及控制服务器的相关情报。IOC 往往是域名、IP、URL 等，这种 IOC 可部署于安全设备（如：IPS（入侵防御）、SIEM（安全、信息和事件管理）等）进行检测发现甚至实时阻截。; ClamAV：Linux 平台上的开源病毒扫描程序，主要应用于邮件服务器，采用多线程后台操作，可以自动升级病毒库。 **安全威胁与安全攻击 安全威胁**：威胁是指对安全可能的侵害，这种侵害并不意味着实际发生。正因为这种侵害可能发生，因此需要进行防范。如果这种侵害发生了，则称为攻击，而执行攻击行为的主体则称为攻击者。威胁的主要类型： 信息泄露 、完整性破坏、服务拒绝、未授权访问 **威胁来源**：内部操作不当、内部管理漏洞、外部安全威胁。 **安全攻击**：任何危及到信息安全的行为（安全攻击往往要利用一个或多个威胁） 安全攻击的类型（IATF 定义）：被动攻击、主动攻击、物理临近攻击、内部人员攻击、配装分发攻击。 **被动攻击**：攻击者只是窃听消息，不对消息做任何形式的修改。攻击者的目标是获取传输的信息，以便进行利用。 被动攻击的后果：信息内容泄漏、流量模式泄漏 对策：流量加密、流量混淆 常用软件 Wireshark、 Sniffer Pro、TCP Dump、 Snort 典型被动攻击：国家级监控：“棱镜”计划（PRISM） **主动攻击**：避开或破坏安全部件， 引入恶意代码，破

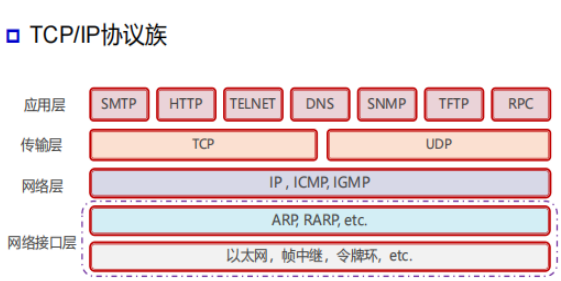
坏数据或系统完整性。 如：假冒某个实体主动发送消息、重放旧消息（re-play）、修改传输中的消息、删除选中的消息、各种 MiTM 攻击。 对策：增强区域边界保护、基于网络管理交互身份认证的访问控制、受保护远程访问、质量安全管理、自动病毒检测工具、审计和入侵检测。 主动攻击特点--可以检测：由于构成系统的物理通信设施、软件和网络协议等存在各种潜在的弱点，因此主动攻击难以绝对防御，但是可以检测。因此，针对主动攻击，重点在于检测并从破坏中恢复过来。 **物理临近攻击** ：一个未授权的个人近距离物理接触网络、系统或设备，以修改、收集信息或者拒绝对信息的访问。这种接近可以通过秘密进入、公开访问或者两者结合。 对策：配置环境监控体系，提供设备物理安全保护 **内部人员攻击**：由在信息安全处理系统物理边界内的合法人员或者能够直接访问信息安全处理系统的人员发起的攻击。 对策：安全意识和训练；审计和入侵检测；安全策略和增强安全性；关键数据、服务和局域网的特殊的访问控制；加强身份识别与认证能力等。 **配装分发攻击**：硬件或软件在生产与安装过程中，或者在运输过程中，被恶意地修改。 对策：可以通过加强处理配置控制将这类威胁降低到最低。通过使用受控分发，或使用由最终用户检验的签名软件和存取控制可以降低分发威胁。 1.3 **网络攻击的策略** APT 攻击（Advanced Persistent Threat）是指一种隐秘而持久的网络攻击，攻击组织通常由一个国家资助，具有政治、军事以及经济动机，通过网络攻击手段获得高价值目标网络的访问权限，并在目标系统中维持较长时间。 APT 行动者的目的是获得并渗出高度机密的信息，如专有技术的信息，如 F22 的源代码等；破坏目标系统的资源完整性，如 stuxnet 中的 PLC 等。 **Lockheed Martin：Cyber Kill chain（攻击链）：** 侦察（Reconnaissance, Recon）：攻击者选择目标，研究目标，试图识别目标网络的漏洞。 武器研制（Weaponization）：攻击者针对目标情况研制远控恶意软件武器，如：病毒、蠕虫，或者恶意 PDF 文档、恶意 office 文档等。 投递（Delivery）：攻击者传输攻击武器到目标，比如：通过电子邮件附件，网站，或者 u 盘 利用（Exploitation）：恶意软件被触发，造成目标系统的漏洞被利用。 安装（Installation）：恶意软件安装可以被攻击者使用的访问点（如：后门）以获得持久访问。 命令与控制（Command & Control）：恶意软件使得攻击者能够持久访问目标网络。目标行动（Action on Objective）：攻击者采取行动来达到其目标，如：数据外泄，数据破坏，加密勒索，入侵其它目标 **Mitre：ATT&CK 框架** Mitre 提出了 ATT&CK 框架，全称为敌手策略、技术与通用知识 Adversarial Tactics, Techniques, and Common Knowledge），是一个基于对真实世界观察而总结出的敌手策略与技术知识库。 策略（Tactic）是指敌手的技术目标（the adversary’s technical goal） 技术（Technique）是指敌手如何实现技术目标 过程（Procedure）是指技术的具体实现 ATT&CK 以矩阵形式呈现，目前分为三类，分别为：Enterprise、Mobile 和 ICS，每个矩阵还可以进一步细化为更具体的多个矩阵。在此基础上，还增加了一个 PRE-ATT&CK 矩阵，用以完善整个知识库。 **企业矩阵** 表示敌手针对企业信息系统的攻击策略与技术，主要包含下述平台的信息：Windows, macOS, Linux, AWS, GCP, Azure, Azure AD, Office 365, SaaS. 该矩阵的部分截图如下，共包含 12 项策略，每项策略有多种实施技术，每项技术又对应多种具体实现。 **PRE-ATT&CK 矩阵** 企业矩阵起始于初始访问，但是无论从攻击还是防御来说都是不完善的。如果按照企业矩阵进行防御，则企业可以采用周界防御措施（如防火墙）根据 IOC（Indicator of compromise）（如：IP 地址、域名、恶意软件哈希值等等）制定的黑名单进行封堵，然而这种防御方式的效果往往是有限的。比如 Verizon 曾经报道过 99%恶意软件的哈希值的可见时间仅小于或等于 58 秒，显然仅仅采用周界防御措施是有局限性的。 仅用企业矩阵是不足以制定完善的防御规划，防御方不仅需要监视和理解敌手在企业周界之内的活动，而且也需要将这些操作扩展到企业之外。PRE-ATT&CK 用于描述敌手发起攻击之前的活动。 1.4 **构建安全系统的原则 最小权限原则（least priviledge）、默认故障安全原则（fail-safe defaults）、安全机制的经济性原则（economy of mechanism）、完全仲裁原则（complete mediation）、开放设计原则（open design）、权限分离原则（separation of priviledge）、最少共用机制（least common mechanism）、心理可接受原则（psychonogical acceptability）、纵深防御原则（defense in depth） 最小权限原则**：主体（用户、程序等）应该仅被授予完成任务所需的访问权限。主体拥有的权限越少，在发生安全问题（如用户登录凭证泄露）时造成的损失越小。（例如：一个操作系统，所有用户都拥有全部权限，那么任何一个用户的登录凭证泄露都会造成整个系统沦陷。反之，如果按照最小权限原则来分配权限，则某个用户的登录凭证泄露所带来的风险仅限于该用户的权限范围） **默认故障安全原则**：如果一个主体没有被明确授权访问一个对象，该主体应该被拒绝访问该对象。默认 故障安全原则要求对一个对象的默认访问是无权限。任何时候，只要安全相关的属性（如访问权限等）没有明确授予，则应被拒绝。而且，如果主体不能在对象上完成其操作，则主体应该在终止之前取消对系统安全状态的改变。该原则用于限制在主体或者对象创建时，如何初始化权限。（例如：如果邮件服务器由于配额等原因不能向指定目录写入邮件消息，则该邮件服务器也不能向其它地方写入邮件数据，而应该关闭网络连接并报告错误。如果该邮件服务器能够

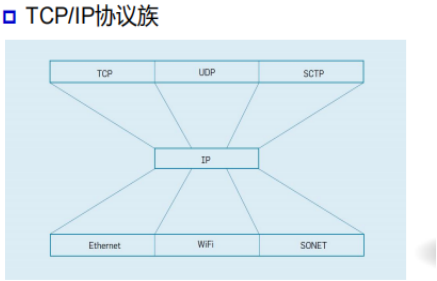
向其它地方写入邮件数据，则可能被攻击者利用，通过发送大量邮件来填满其它存储空间，可能导致整个系统崩溃）。 **安全机制的经济性原则：**安全机制的经济性原则要求“安全机制应该尽可能简单”如果设计比较简单，则机制的构件比较少，从而实现容易，所需的测试用例相对较少，安全机制出错的概率大大降低。**完全仲裁原则：**对资源的所有访问均需要审核。任何时候，当一个主体试图读取一个对象，操作系统都应该审核该行为。首先，操作系统应该确定主体是否被许可读取该对象；如果是，那么允许当前的读操作。之后，如果主体再次试图读取该对象，操作系统应该再次审核该操作是否被允许。 **开放设计原则：**和 Kerckhoff **原则一致，密码系**统应该在就算攻击者知道所有系统内部细节的情况下也保持安全。对于密码系统来说，密钥应该是唯一需要保密的，系统应该被设计为容易更换密钥，因此在密钥泄露的情况下，通过更换密钥，系统仍然能够保持安全性。典型实例：IKE、IPSec、TLS、WPA、RSA **权限分离原则：** 系统不能基于单一条件来授予访问许可，或者说系统需要同时满足多个条件时，才授予某个主体对某个资源的访问许可。（例如：在 BSD unix 中，普通用户可以使用 su 命令切换到 root 用户，但是需要两个条件，该用户知道 root 的口令，用户的组 ID 为 0。双因素身份认证，如银行卡取钱，需要提供银行卡和 PIN；某些重要网站的登录需要提供账号，移动认证码） **最少共用机制：**互不信任的用户之间访问资源的共用机制应该最小化，包括：共用子系统、共享资源、共享代码等等。因为共用机制提供了一种在攻击者和被攻击者之间的潜在通道，攻击者可能利用这种通道进行攻击。（例如：虚拟机和容器，虚拟机之间共享的资源少于同一平台上容器之间共用的资源，而事实上也证明，在隔离效果上虚拟机比容器更安全。DNS 缓存毒化攻击，其能够成功的原因也在于用户之间共用了 DNS 缓存。） **心理可接受原则：**安全机制不应该增加访问资源的难度，安全机制应该易于使用。在引入安全机制时，需要考虑如何屏蔽安全机制的复杂性，达到易于安装、配置和使用。如果做不到心理可接受性，用户可能会采取措施导致安全机制失效。（例如：安全管理员要求员工的口令必须是随机生成且包含大小写字母、数字、特殊符号，且不低于 16 个字符，且每周更新。这种要求通常会导致员工把口令写到便利贴上，贴到容易看到的地方，这种方式本质上导致了安全机制失效。） **纵深防御：**指在一个信息系统（包括：单机系统、网络系统等）中部署多层次的安全控制（防御）措施，其目的是提供冗余的安全保护，同时不同防御措施之间还存在互补性，以免一个防御措施失效，整个系统沦陷。（比如，在我们的个人电脑上通常都部署了防火墙和防病毒软件，而高价值的服务器还会部署 HIDS 等。在企业网络环境，还存在更复杂的各种安全措施） 2. **互联网协议的安全性分析**

1.TCP/IP 协议安全性概述

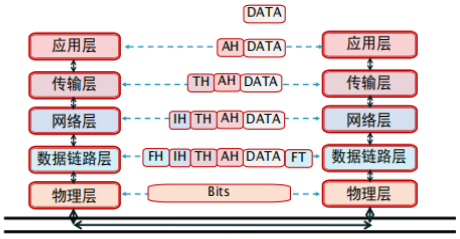


物理层（Physical Layer）： 负责在物理媒介上传输数据比特流，包括传输介质、电压等物理特性。
数据链路层（Data Link Layer）： 提供可靠的点对点数据传输，负责数据帧的传输、错误检测和纠正，以及物理地址寻址。
网络层（Network Layer）： 负责在网络上选择路由并传输数据包，处理逻辑地址（IP 地址），实现数据分组的路由和转发。
传输层（Transport Layer）： 提供端到端的数据传输服务，确保可靠的数据传输，包括数据分段、错误校验、流量控制等。
会话层（Session Layer）： 管理不同计算机之间的会话，包括建立、管理和终止通信会话。
表示层（Presentation Layer）： 负责数据格式的转换和加密解密，确保不同系统之间的数据格式兼容性。
应用层（Application Layer）： 提供用户接口和应用程序之间的通信服务，包括识别通信伙伴、数据交换和网络服务。

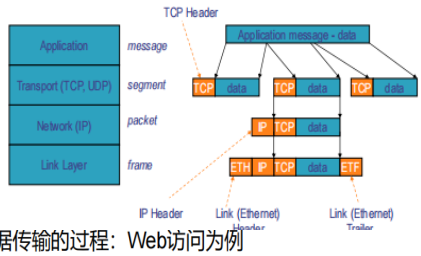




TCP/IP协议数据封装

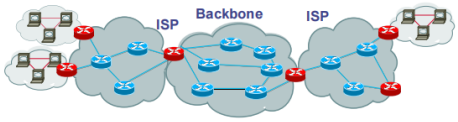


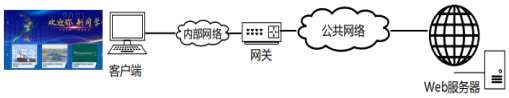
数据封装：TCP



数据传输的过程：Web访问为例

数据包从源到目的经过多跳

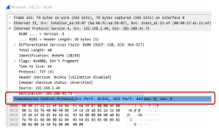




正确的MAC帧头 | 正确的IP包头 | 正确的TCP报头 | GET https://www.uestc.edu.cn

正确的TCP报头

- 要访问https://www.uestc.edu.cn首先要和www.uestc.edu.cn服务器建立连接，建立连接所使用的端口号由应用层协议指定。

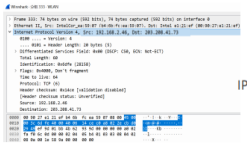


TCP包头部分包括源端口、目的端口、TCP标志、TCP选项等内容。

正确的MAC帧头 | 正确的IP包头 | 正确的TCP报头 | GET https://www.uestc.edu.cn

正确的IP包头

- 目的IP地址通过本地缓存的host文件或通过DNS解析确定。

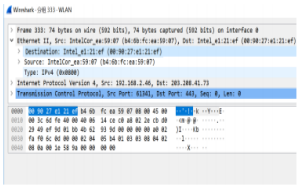


IP包头部分包括源IP地址、目的IP地址、IP标志等内容

正确的MAC帧头 | 正确的IP包头 | 正确的TCP报头 | GET https://www.uestc.edu.cn

正确的MAC帧头

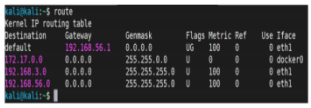
- 由路由表信息和ARP缓存信息，可以确定帧头的目的MAC地址。



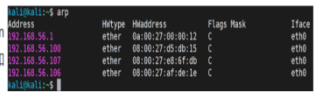
MAC帧头部分包括前导码、帧开始符、目的MAC地址等内容。

正确的MAC帧头 | 正确的IP包头 | 正确的TCP报头 | GET https://www.uestc.edu.cn

主机路由表与ARP缓存



- route 显示当前主机的路由表信息
- DestIP & Netmask == Destination Network ，选择相应的Interface和Gateway发送数据



- arp命令显示了本机的arp缓存
- arp缓存记录了IP地址与MAC地址的映射

安全问题的来源：互联网设计之初的使用目的是用于科学研究，其基本假设就是节点的诚实性；由于计算机网络的广泛使用，这种假设在今天已经无法成立，因此可能导致各种各样的攻击。

安全性问题分类：

设计缺陷导致的安全性问题：协议设计的缺陷，这类安全性问题会一直存在，直至该协议更新

实现缺陷导致的安全性问题：协议实现的缺陷，这类安全性问题会随着软件的更新而消除

信息泄露：TCP/IP 协议在设计时没有考虑保密性服务，所有消息均通过明文方式传输，导致消息在传输过程中存在信息泄露的安全威胁。

消息伪造：TCP/IP 协议在设计时没有考虑身份认证和完整性服务，导致消息容易被伪造。

拒绝服务：TCP/IP 协议在设计时没有考虑可用性服务，导致拒绝服务攻击。

安全威胁分布：　应用层• DNS 欺骗，邮件炸弹　　传输层• SYN Flood 攻击，会话挟持

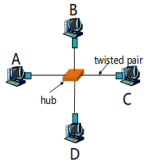
网络层• ICMP 重定向攻击，IP 分片攻击　网络接口层• 嗅探，ARP 欺骗，交换机毒化

2 网络接口层协议安全分析

网络嗅探

共享环境下的网络嗅探

- 以太网采用星型拓扑结构，使用集线器（hub）或者交换机（switch）连接网络节点。
- 集线器本质上是物理层的中继器：
  - 处理的基本单位是比特
  - 信号放大，延长网络距离
  - 收到的比特发送给所有其它连接节点
  - 多个端口使用相同的传输速率，没有帧缓存
  - 没有CSMA/CD：由计算机的网卡检测冲突

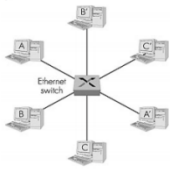


共享环境下的网络嗅探

工作原理：网络嗅探器是通过对网卡的编程来实现的。对网卡的编程是使用原始套接字方式来进行，Windows环境下通过创建原始套接字s=socket(AF_INET,SOCK_RAW,IPPROTO_RAW);设置为对IP头进行编写操作setsockopt(s,IPPROTO_IP,IP_HDRINCL,(char*)&bFlag,sizeof(bFlag));设置SOCK_RAW为SIO_RCVALL，以便接收所有的IP包ioctlsocket(s, SIO_RCVALL, &dwValue)，可以将网卡设置为混杂模式，来获取网络接口上侦听到的所有的数据包。

交换环境下的网络嗅探

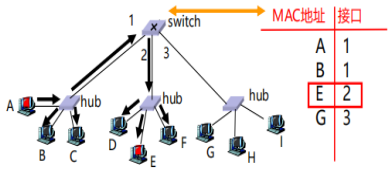
- 交换机采用接口转发方式实现主机间的通信。
  - 交换机工作在链路层
  - 基于帧转发，实现MAC地址过滤
  - 物理上和逻辑上都是星型结构
  - 交换：A-to-A'和 B-to-B'同时工作，不冲突



交换环境下的网络嗅探（需要通过交换机毒化或ARP欺骗）

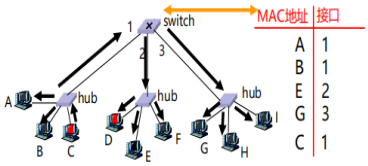
- 交换机内部保存一个源地址表又称为交换表
  - 交换表的表项: (MAC地址, 接口, 时间)
  - 交换表中过期的表项将被删除 (TTL 可以是60分钟)
  - 交换机学习哪一个主机可以通过哪一个接口到达。交换机当接收一个数据帧时,交换机“学习”发送者的位置：即数据进入交换机的LAN网段与接口之间的对应关系，在交换表中记录发送者/位置对应关系
- 上述机制称之为交换机的“自学习”

交换机工作过程（交换表有记录）



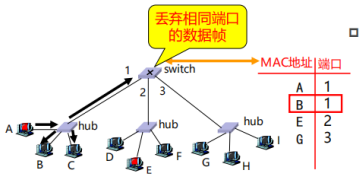
- 假设A发送数据帧到E
  - 交换机接收来自A的数据帧
    - 注意在交换表中A在交换机的接口1上，E在交换机的接口2上
    - 交换机将转发数据帧到接口
  - 数据帧被E接收

交换机工作过程（交换表无记录）



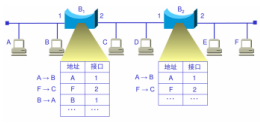
- 假设C发送数据帧到D
  - 交换机接收来自C的数据帧
    - 记录C所对应的接口号1
    - 因为D不在交换表中,交换机将转发数据帧到接口2和3
  - 数据帧被D接收

交换机工作过程（接口相同情况）



- 假设A发送数据帧到B
  - 交换机接收来自A的数据帧
    - 注意在交换表中A在交换机的接口1上,B也在交换机的接口1上
  - 数据帧将被交换机丢弃

交换机毒化攻击（嗅探）



- 交换表的空间是有限的，新的“MAC地址——接口”映射对的到达会替换旧的表项。
- 如果攻击者发送大量的具有不同伪造源MAC地址的帧，由于交换机的自学习功能，这些新的“MAC地址—接口”映射对会填充整个交换机表，而这些表项都是无效的，结果交换机完全退化为广播模式，攻击者达到窃听数据的目的。

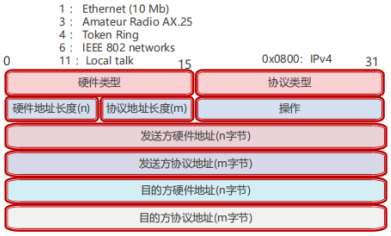
ARP 欺骗

ARP协议的功能

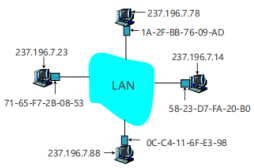
- 在互联网上是使用IP地址来定位主机，而但在交换机上是通过MAC—接口映射来实现主机间数据帧的发送，因此需要使用协议完成IP地址和MAC地址的转换。
  - ARP协议：IP地址→MAC地址
  - RARP协议：MAC地址→IP地址

ARP协议位于网络层与数据链路层之间

ARP头部结构



ARP协议工作原理



- 每个在局域网上的IP节点 (Host, Router)都有ARP表
- ARP表:局域网上（部分）节点的IP/MAC地址映射

<IP address; MAC address; TTL>

- TTL (Time To Live): 映射地址的失效时间 (典型为20分钟)

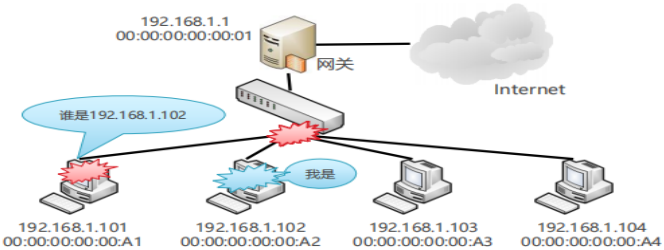
ARP协议工作原理：ARP是即插即用的，无需网络管理员干预，节点就能创建ARP表。

- A想发送分组给 B，A知道 B的IP地址，假设B的MAC地址不在A的ARP表中

ARP协议工作过程

192.168.1.101主机的本地ARP缓存

192.168.1.1	00:00:00:00:00:01
192.168.1.101	00:00:00:00:00:A1
192.168.1.102	00:00:00:00:00:A2
192.168.1.103	00:00:00:00:00:A3
192.168.1.104	00:00:00:00:00:A4



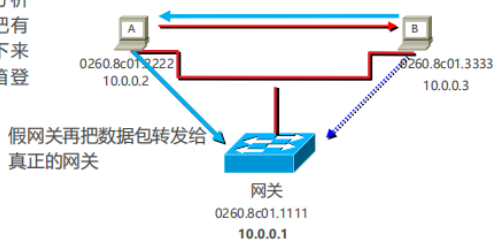
ARP 协议的特殊设计（改进效率）：响应 ARP 请求的主机将请求者的 IP－MAC 映射缓存；主动的 ARP 应答会被视为有效信息接受

ARP 协议的缺陷：ARP 协议设计之初没有考虑认证问题，所以任何计算机都可以发送虚假的 ARP 数据包；ARP 协议的无状态性。响应数据包和请求数据包之间没有什么关系，如果主机收到一个 ARP 响应却无法知道是否真的发送过对应的 ARP 请求；ARP 缓存需要定时更新，给攻击者以可乘之机

ARP欺骗攻击过程

攻击者在局域网段发送捏造的IP/MAC对应信息，篡改攻击目标的arp缓存中关于网关的表项，使自己成为假网关

假网关（攻击者）分析接收到的数据包，把有价值的数据包记录下来（比如QQ以及邮箱登录数据包）



ARP 欺骗攻击的特点

危害：嗅探、中间人攻击、拒绝服务攻击

局限性：ARP 欺骗只能被用于局域网（攻击者必须已经获得局域网中某台机器的访问权）

3 IP 协议安全性分析

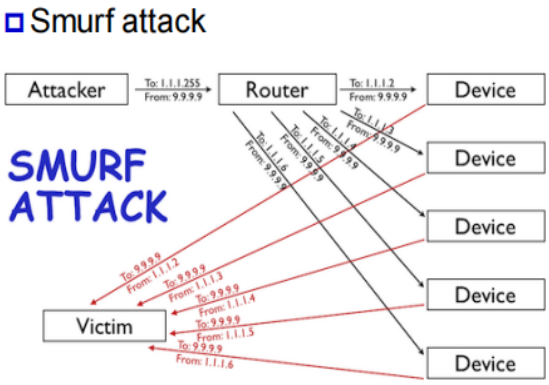
IP 假冒攻击

攻击原理：IP 协议本身没有验证源 IP 地址真实性的机制

攻击类型：

拒绝服务--避免被追踪而受到惩罚，构造针对同一目的 IP 地址的 IP 分组，而源 IP 地址为随机的 IP 地址。

基于 IP 地址认证的网络服务欺骗--假冒可信的 IP 地址而非法访问计算机资源，X-window、rlogin、rsh 等。



步骤：

1、攻击者构造 ICMP 报文，源 IP 地址为受害者的 IP 地址，目的 IP 地址为一个网络的广播地址。然后，发送出去。

2、目标网络的主机收到 ICMP 报文后，以受害者的 IP 地址为目的 IP 地址进行响应。

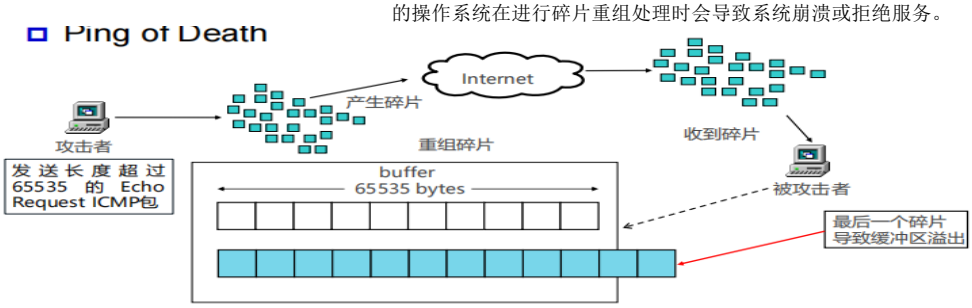
3、结果，受害者被 DDos。

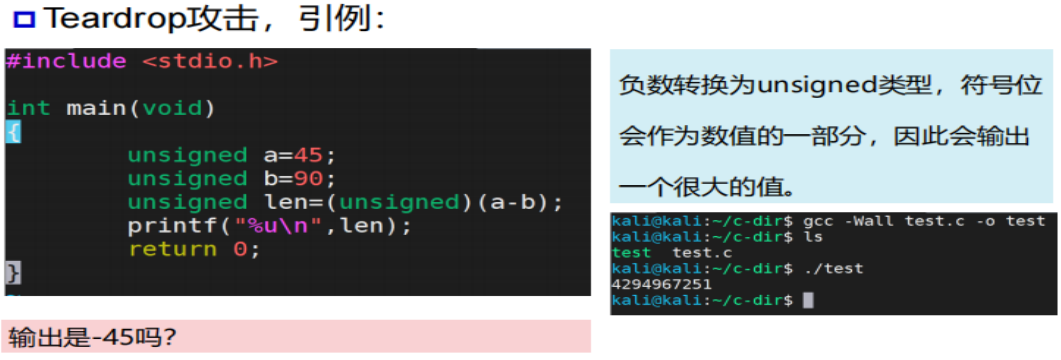
IP 碎片攻击

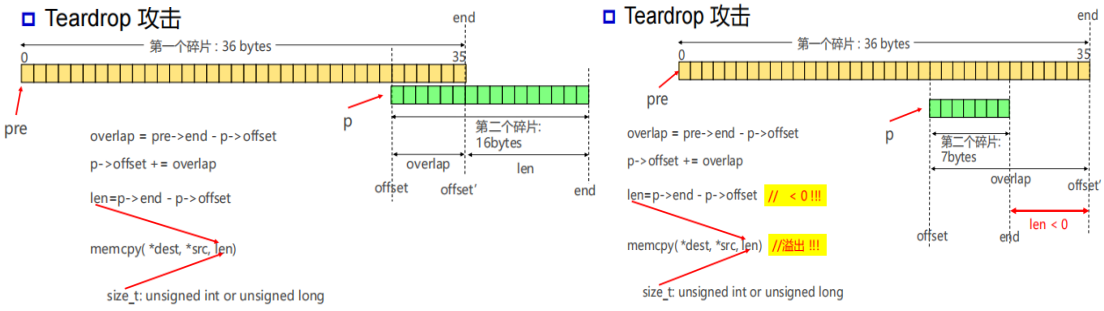
攻击原理

链路层具有最大传输单元 MTU 这个特性，它限制了数据帧的最大长度，不同的网络类型都有一个上限值，以太网的 MTU 是 1500。如果 IP 层有数据包要传，而且数据包的长度超过了 MTU，那么 IP 层就要对数据包进行分片(fragmentation)操作，使每一片的长度都小于或等于 MTU。

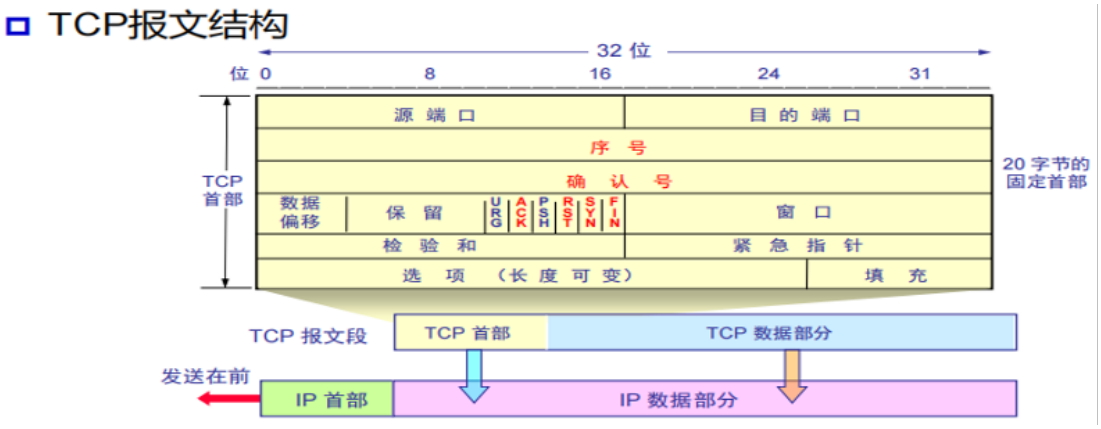
IP 首部有两个字节表示整个 IP 数据包的长度，所以 IP 数据包最长只能为 0xFFFF，就是 65535 字节。如果有意发送总长度超过 65535 的 IP 报文，或构造畸形的 IP 碎片，部分老的操作系统在进行碎片重组处理时会导致系统崩溃或拒绝服务。

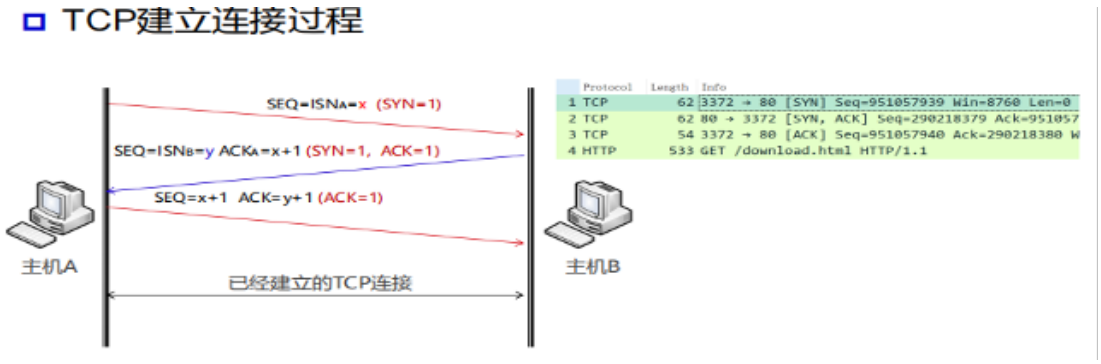






2.4 传输层协议安全性分析　TCP 协议安全威胁

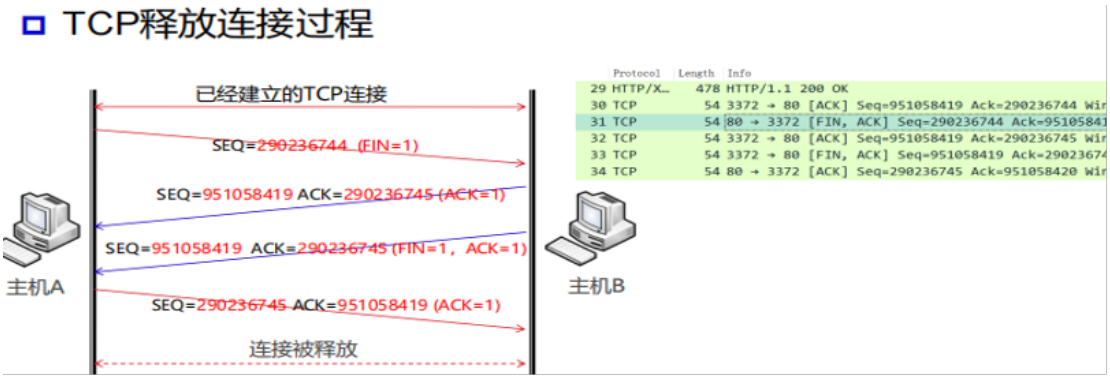




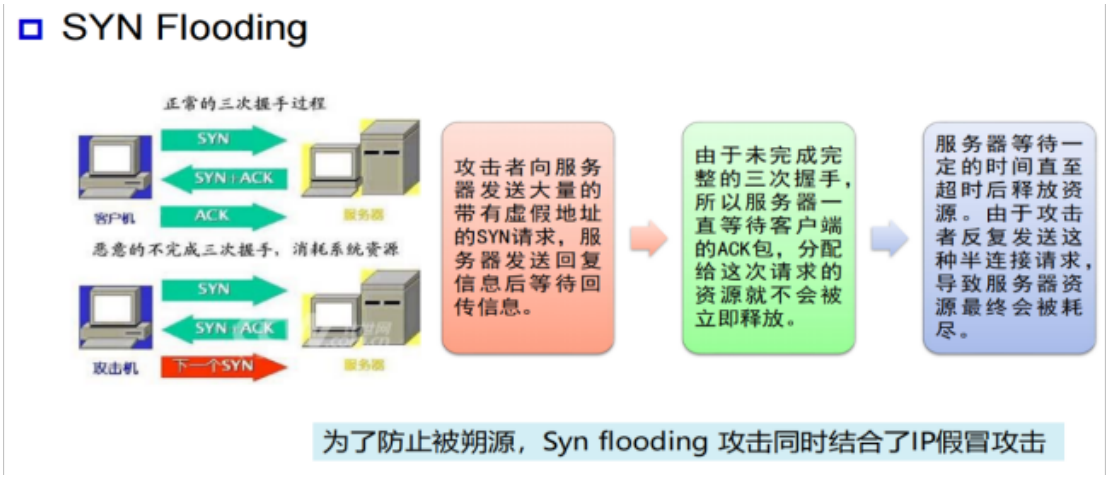
时间	145.254.160.237		65.208.228.223	注释
0.000000	3372	SYN	80	Seq = 951057939
0.911310	3372	SYN, ACK	80	Seq = 290218379 Ack = 951057940
0.911310	3372	ACK	80	Seq = 951057940 Ack = 290218380
0.911310	3372	PSH, ACK - Len: 479	80	Seq = 951057940 Ack = 290218380
1.472116	3372	ACK	80	Seq = 290218380 Ack = 951058419
1.682419	3372	ACK - Len: 1380	80	Seq = 290218380 Ack = 951058419
1.812606	3372	ACK	80	Seq = 951058419 Ack = 290219760
1.812606	3372	ACK - Len: 1380	80	Seq = 290219760 Ack = 951058419
2.012894	3372	ACK	80	Seq = 951058419 Ack = 290221140
2.443513	3372	ACK - Len: 1380	80	Seq = 290221140 Ack = 951058419
2.553672	3372	PSH, ACK - Len: 1380	80	Seq = 290222520 Ack = 951058419
2.553672	3372	ACK	80	Seq = 951058419 Ack = 290223900
2.633787	3372	ACK - Len: 1380	80	Seq = 290223900 Ack = 951058419
2.814046	3372	ACK	80	Seq = 951058419 Ack = 290225280
2.894161	3372	ACK - Len: 1380	80	Seq = 290225280 Ack = 951058419
3.014334	3372	ACK	80	Seq = 951058419 Ack = 290226660
3.374852	3372	ACK - Len: 1380	80	Seq = 290226660 Ack = 951058419
3.495025	3372	PSH, ACK - Len: 1380	80	Seq = 290228040 Ack = 951058419
3.495025	3372	ACK	80	Seq = 951058419 Ack = 290229420
3.635227	3372	ACK - Len: 1380	80	Seq = 290229420 Ack = 951058419

序列号和确认号（SYN 用于初始化连接，而 ACK 则用于确认收到数据或者确认连接的建立）**包 1**：客户端发起建立连接请求，SYN 标志为 1，序 号 为 随 机 生 成 的 整 数 951057939　**包 2**：服务端响应客户端的建立连接请求，SYN 和 ACK 标志均置为 1，序号为随机生成的整数 290218379，确认号为客户端的初始序号+1 即：951057939 + 1 = 951057940　需要注意的是，尽管客户端没有发送任何有效数据，确认号还是被加 1，这是因为接收的包中包含 SYN 或 FIN 标志位（并不会对有效数据的计数产生影响，因为含有 SYN 或 FIN 标志位的包并不携带有效数据）　**包 3**：客户端对服务端的连接请求进行确认，ACK 标志为 1，序号为包 2 的确认

号，即：951057940；而确认号为包 2 的序号加 1，即：290218379 + 1 = 290218380 **包 4**：这是 TCP 流中第一个携带有效数据的包（确切的说，是客户端发送的 HTTP 请求），序列号与前一个 ACK 报文一致，因为到上个包为止，还没有发送任何数据；确认号也与前一个 ACK 报文一致，因为客户端没有从服务端接收到任何数据。 包内携带发送给服务器的数据为 479 个字节。 **包 5**：当上层处理 HTTP 请求时，服务端发送该包来确认客户端在包 4 中发来的数据，需要注意的是，确认号的值增加了 479（479 是包 4 中有效数据长度），即：951057940 + 479 = 951058419。服务端以此来告知客户端端，目前为止，我总共收到了 479 字节的数据，服务端的序列号保持不变。 **包 6**：这个包标志着服务端返回 HTTP 响应的开始。因为服务端在该包之前返回的包中都不带有有效数据，所以序列号依然不变。该包带有 1380 字节的有效数据。 **包 7**：由于上个数据包的发送成功，TCP 客户端的序列号增长至 951058419，从服务端接收了 1380 字节的数据，客户端的确认号 290218380 + 1380 = 290219760。



TCP 协议的特点 全双工连接(full-duplex connection) 该连接的两端有两条彼此独立、方向相反的传输通道 面向连接(connection-oriented) 通信双方在开始传输数据前，必须通过“三次握手”的方式在双方之间建立一条逻辑上的链路（虚电路），用于传输数据 可靠性(reliable) 自动分片；保证传送给应用层的数据顺序是正确的 ；自动过滤重复的封包 ；确认-重传确保数据包可靠到达 面向字节流(byte-stream) 将应用程序和网络传输相分割，为流传输服务提供了一个一致的接口 **拒绝服务（Denial of Service）** DoS 攻击是指利用网络协议漏洞或其他系统以及应用软件的漏洞耗尽被攻击目标资源，使得被攻击的计算机或网络无法正常提供服务，直至系统停止响应甚至崩溃的攻击方式。即攻击者通过某种手段，导致目标机器或网络停止向合法用户提供正常的服务或资源访问。利用 TCP 面向连接的特点:三次握手过程需要存储连接状态，因此会产生系统开销。

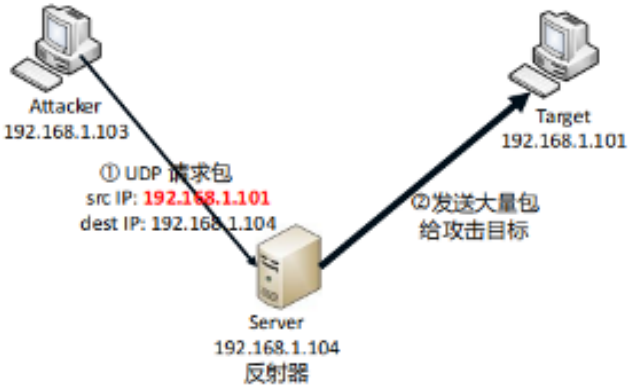


SYN Flooding 特点： 针对 TCP/IP 协议的设计缺陷进行攻击；发动攻击时，只要很少的数据流量就可以产生显著的效果；攻击来源无法定位；在服务端无法区分 TCP 连接请求是否合法。 **防御措施**（RFC4987）： 在防火墙上过滤来自同一主机的后续连接；采用 SYN Cookie（无法防范全连接拒绝服务） **其它针对 TCP 协议的攻击 ACK Flooding：**主机接收到带有 ACK 状态的数据包，需要检测数据包所包含的连接四元组是否存在，如存在需要检查数据包状态数据是否合法。（消耗资源） **序列号猜测攻击**（会话劫持）：攻击者通过猜测序列号，在 TCP 会话中插入自己构造的数据包。 **Land 攻击：**构造一个 SYN 包，其源地址和目标地址都被设置成某一个服务器地址；导致接收服务器向它自己的地址发送 SYN-ACK 消息，结果这个地址又发回 ACK 消息并创建一个空连接；每一个这样的连接都将保留直到超时。

UDP 协议安全威胁

UDP协议报头

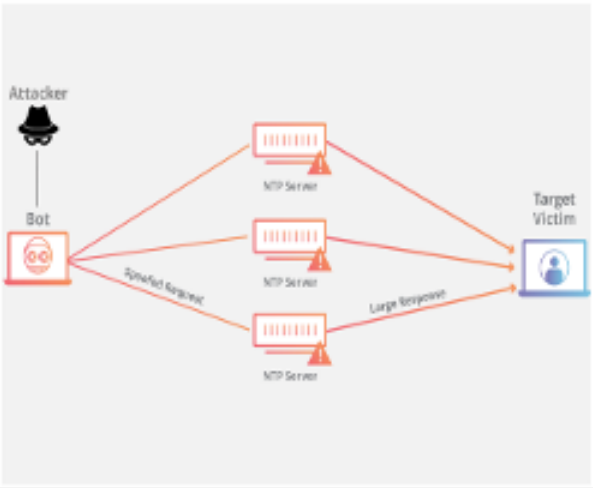
UDP反射攻击



① 攻击者构造以攻击目标IP地址为源IP地址的UDP请求包，发送给服务器。

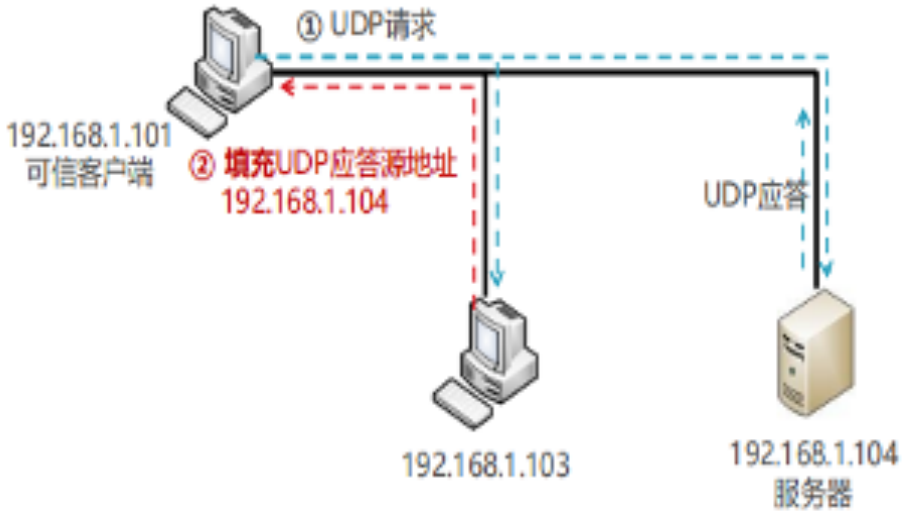
① 服务器发送UDP响应包给攻击目标，通常UDP响应流量是请求流量的几十倍，导致攻击目标被DoS （流量放大）

UDP反射攻击实例：NTP DDoS攻击



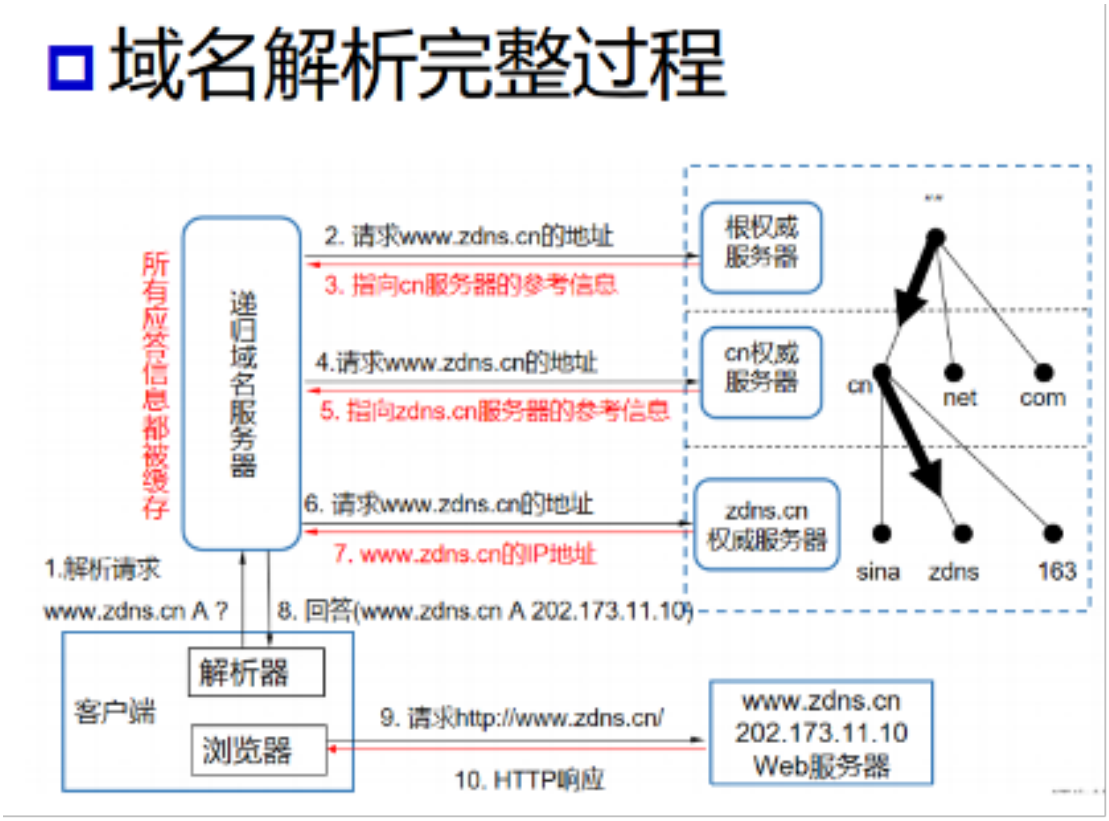
1. 攻击者发送**带有伪造 IP 地址（受害者的真实 IP 地址）**的 UDP 包发给启用了 monlist 命令的 NTP 服务器。
2. 每个 UDP 数据包使用其 monlist 命令向 NTP 服务器发出请求，NTP服务器将返回最近访问的600个客户端的IP地址，分6个IP地址一个包进行发送。（流量放大约200倍）
3. 攻击者查找多个符合要求的NTP服务器，同时发送伪造的NTP请求，导致目标主机/网络被大流量淹没，形成DDoS攻击。

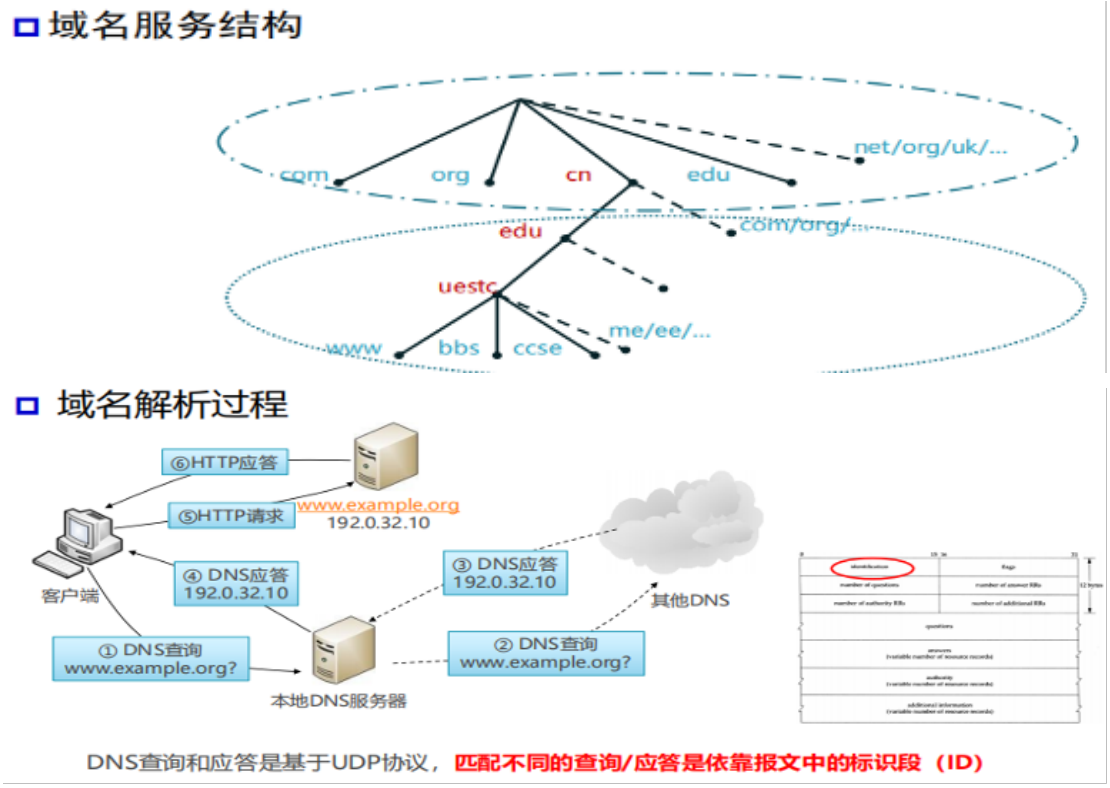
UDP劫持



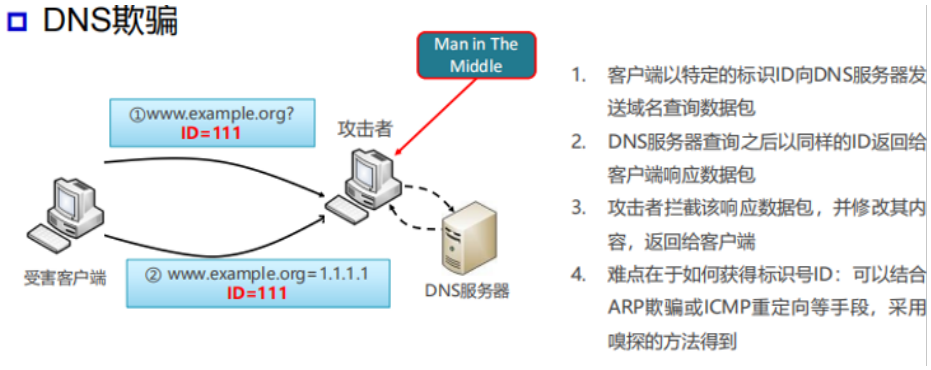
2.5 应用层安全协议分析　DNS 协议安全威胁

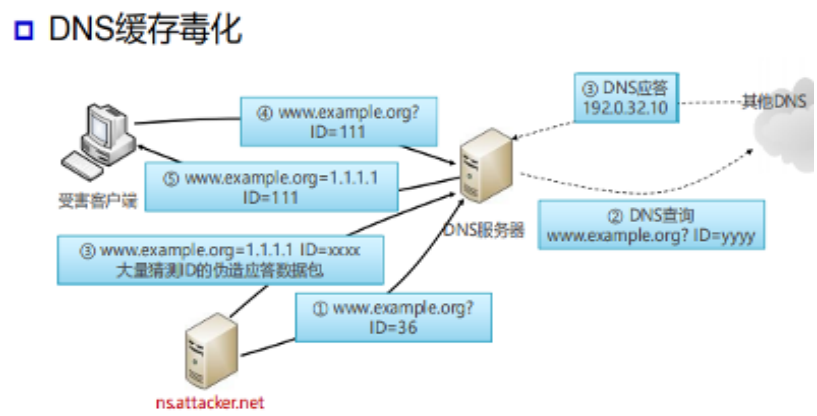
域名解析完整过程

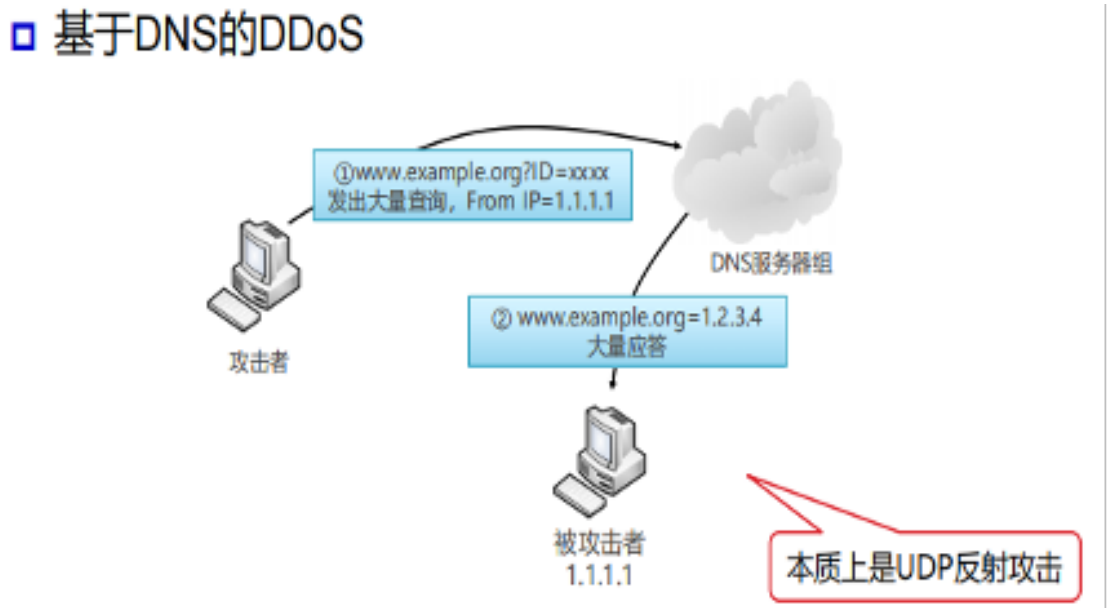




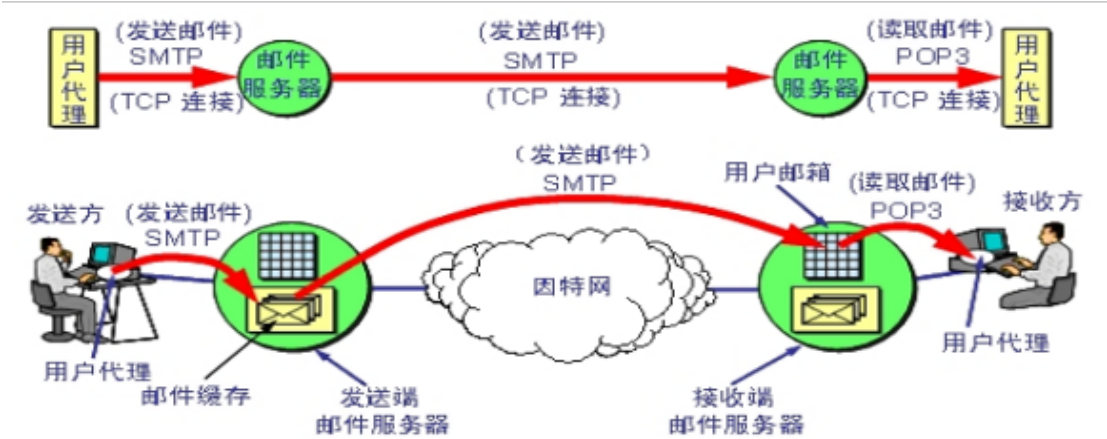
1. 客户端在打开浏览器，输入域名 www.zdns.cn。浏览器会发起一个 DNS 解析请求。
2. 如果本地 DNS 服务器中找不到结果，则首先向根服务器查询，根服务器里面记录了各个顶级域服务器的 IP 地址。
3. 当向根服务器请求 www.zdns.cn 的时，根服务器就会返回 cn 域名的解析结果。
4. 递归域名服务器向 cn 权威服务器发起 www.zdns.cn 的解析请求。
5. cn 权威服务器查找并返回 zdns.cn 的解析结果。
6. 递归域名服务器向 zdns.cn 的权威服务器发起域名解析请求。7. zdns.cn 权威服务器返回 www.zdns.cn 的解析结果。
7. 递归域名服务器缓存查询结果并向客户端返回查询结果 202.173.11.10。9. 客户端访问 web 服务器获取数据







电子邮件协议安全威胁



电子邮件协议 SMTP：主要负责底层的邮件系统如何将邮件从一台机器传至另外一台机器 POP：目前的版本为 POP3，POP3 是把邮件从电子邮箱中传输到本地计算机的协议 IMAP：目前的版本为 IMAP4，是 POP3 的一种替代协议，提供了邮件检索和邮件处理的新功能 S/MIME:支持邮件加密的传输协议 电子邮件协议的安全：传输安全、发送者身份确认、接收者已收到确认、邮箱炸弹攻击 **HTTP 协议安全威胁 HTTP 钓鱼攻击 跨站攻击**：浏览器对网页的展现是通过解析 HTML 代码实现的，如果传入的参数含有代码浏览器会解析它而不是原封不动的展示。跨站攻击类型： 持久性跨站（persistent XSS or stored XSS） - 攻击数据存放于服务器。当用户访问正常网页时，服务端会将恶意的指令夹杂在正常网页中传回给用户。 非持久性跨站（non-persistent XSS or reflected XSS）- 当服务端未能正确地过滤客户端发出的数据，并根据用户提交的恶意数据生成页面时，就有可能生成非持久性跨站攻击。 DOM 跨站（DOM-based XSS） - 如果客户端脚本（例如 JavaScript）动态生成 HTML 的时候，没有严格检查和过滤参数，则可以导致 DOM 跨站攻击。

如果主机 A 跳过与主机 B 建立 TCP 连接的前两个步骤，直接发送三次握手中最后一个带 ACK 标志的包，主机 B 会如何处理？ 它将无法正确地识别这个连接请求，因此会忽略这个单独的带有 ACK 标志的包

如果应用程序在释放连接的过程中，由于应用程序异常终止来不及通知 TCP 协议释放连接，试问在实际情况中应该如何处理这种异常？ 通常，在实际情况中，操作系统会针对这样的情况采取一些机制来处理：TCP 超时和重传： TCP 协议具有超时和重传机制，如果一端长时间没有收到另一端的响应，它会尝试重新发送数据或发送连接释放的请求。这有助于避免永久性的连接保持状态。操作系统资源管理： 操作系统会周期性地清理无效的或闲置的连接资源，以确保系统资源的有效利用。

UDP 协议安全威胁产生的根本原因是什么？请举例分析。 UDP 的设计目标是简单快速，但也因此存在一些安全威胁，其根本原因主要包括： 缺乏连接状态和可靠性： UDP 不像 TCP 那样有连接状态管理，因此缺乏对数据传输的控制和监视。攻击者可以轻易伪造数据包或进行数据包的欺骗性操作，因为 UDP 不验证数据包的来源或内容的完整性。 易于伪造 IP 地址： UDP 协议本身不提供对 IP 地址伪造的保护机制。攻击者可以轻

松伪造源 IP 地址发送 UDP 数据包，这种做法称为 IP 欺骗（IP Spoofing）。例如，DNS 放大攻击（DNS Amplification Attack）利用 UDP 的特性，伪造 IP 地址发送大量的 DNS 请求，向目标服务器发送大量响应，导致拒绝服务（DDoS）攻击。 易于进行 UDP 泛洪攻击： UDP 泛洪攻击（UDP Flood Attack）利用 UDP 协议的特性，向目标服务器发送大量的 UDP 数据包，消耗服务器的网络带宽和处理能力，导致服务不可用。

域名解析协议中主要存在哪些安全威胁？简要说明威胁过程和原理。 DNS 劫持： 攻击者篡改 DNS 响应，将用户的域名解析请求重定向到恶意网站。这种攻击可能通过在本地网络或路由器上植入恶意 DNS 服务器实现，或者在用户计算机中安装恶意软件进行 DNS 劫持。当用户输入一个域名时，他们将被重定向到攻击者控制的恶意网站，可能导致信息泄露或钓鱼攻击。

域名解析协议中主要存在哪些安全威胁？简要说明威胁过程和原理。 DNS 投毒： 攻击者在 DNS 缓存中插入虚假的映射信息，当其他用户向同一 DNS 服务器请求相同的域名时，他们会获得被篡改的结果。这种攻击可以利用 DNS 响应的缓存机制，将虚假信息注入到缓存中，影响其他用户的域名解析结果。 DNS DDos： 攻击者尝试大量的 DNS 查询请求，以耗尽服务器资源或者使得服务器无法响应合法的查询。对 DNS 服务器发动大规模的查询请求，导致其超载并拒绝正常用户的服务请求。 DNS 欺骗： 攻击者在传输过程中伪造 DNS 响应，使得客户端接收到的 IP 地址是攻击者所控制的恶意服务器 IP 地址，而不是正确的 IP 地址。这可能导致用户被重定向到恶意网站，进行钓鱼攻击或者窃取用户的敏感信息。