计算机和网络安全课程笔记 Week 6

Nite

2025-03-19T12:18:00+11:00

Diffie-Hellman 中间人攻击 (Diffie-Hellman MITM Attack)

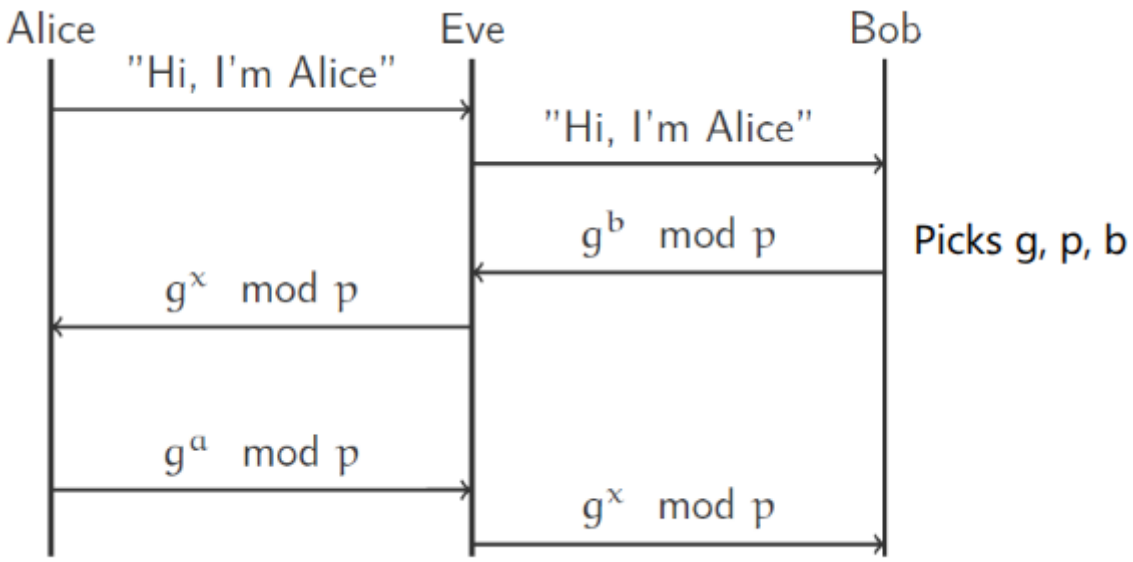

图 1: DH MITM attack diagram

- Eve 拦截 Alice 的 $g^a \pmod p$ 和 Bob 的 $g^b \pmod p$。
- Eve 与 Alice 进行 DH 交换，但使用自己的秘密 e。Alice 以为她与 Bob 交换，但实际上她与 Eve 交换，共享秘密为 $g^{ae} \pmod p$。
- Eve 与 Bob 进行 DH 交换，也使用自己的秘密 e。Bob 以为他与 Alice 交换，但实际上他与 Eve 交换，共享秘密为 $g^{be} \pmod p$。
- Eve 拥有了与 Alice 和 Bob 各自的共享秘密，她可以解密 Alice 发给 Bob 的消息，读取后用与 Bob 的共享秘密重新加密再发给 Bob，反之亦然。
- Alice 和 Bob 并不知道 Eve 的存在。

Encrypted Key Exchange (EKE)

- **原理:** 在进行 DH 密钥交换之前，通信双方（Alice 和 Bob）已经共享一个（可能低熵的）密钥 K。
- **工作方式:** 使用共享密钥 K 来增强最终的高熵共享秘密 $g^{ab} \pmod p$ 的安全性。
- **优点:**
  - 阻止窃听（eavesdropping）。
  - 阻止针对 DH 的主动 MITM 攻击。

  - 维护前向保密性（forward secrecy）。
- **问题:** 需要 Alice 和 Bob 在密钥交换之前已经拥有一个共享密钥 K。

完美前向保密 (Perfect Forward Secrecy, PFS)

- **定义:** 即使长期密钥（long-term keys）泄露，也不会危及过去的短期会话（session）。
- **实现:** 临时密钥（ephemeral keys），例如在 Diffie-Hellman 中生成的密钥，自动提供完美前向保密。
- **不具备 PFS 的情况:** 直接使用 RSA 公钥加密所有内容不具备前向保密性。
- **CA 与 PFS:** 在公钥管理中，CA 永远不会知道用户的私钥，这对于前向保密性很重要。不对称密钥系统（如 SSL/TLS, PGP）的 TTP（CA）通常离线，只在生成阶段使用，它们只知道公钥，这有助于 PFS。

已知密钥攻击 (Known-Key Attack)

- **定义:** 如果过去的会话密钥（session keys）泄露，攻击者可以危及未来的会话密钥（包括主动冒充他人）。

Needham–Schroeder Protocol

- **目的:** 使用可信第三方 (Trusted Third Party, TTP)，即 Trent，协助 Alice 和 Bob 建立一个会话密钥 (session key)。
- **前提:** 所有参与者（Alice, Bob）都与 Trent 共享一个长期密钥。
- **流程:**
  1. Alice 向 Trent 发送请求与 Bob 对话的消息：(A, B, r_A)，其中 r_A 是一个一次性随机数(nonce)。
  2. Trent 向 Alice 发送会话密钥 k_{AB} 和一个给 Bob 的“票据”：$E_{k_{AT}}(r_A, B, k_{AB}, E_{k_{BT}}(k_{AB}, A))$，其中 k_{AT} 是 Trent 与 Alice 的共享密钥，k_{BT} 是 Trent 与 Bob 的共享密钥。
  3. Alice 将票据发送给 Bob：$(E_{k_{BT}}(k_{AB}, A), E_{k_{AB}}(s_A))$。
  4. Bob 挑战 Alice：$E_{k_{AB}}(s_A - 1, r_B)$，其中 r_B 是一个 nonce。
  5. Alice 回应 Bob 的挑战：$E_{k_{AB}}(r_B - 1)$。
- **问题:**
  - Bob 无法保证会话密钥 k_{AB} 是“新鲜的”（fresh），旧的会话密钥不会过期且有价值。
  - 如果 Eve 获取了 Alice 的长期密钥 k_{AT}，她可以阅读 Alice 的所有消息并冒充 Alice 与其他人通信。
  - Alice 需要撤销她的密钥，但这只能由 Trent 完成，密钥撤销是一个主要问题。
  - 该协议假设系统中的所有用户都是“好人”，目标是阻止“坏人”进入，即“蛋壳模型”(eggshell model)。

Kerberos

- **基础:** 建立在 Needham-Schroeder 协议之上。
- **目的:** 允许不安全网络中的节点相互证明身份。
- **特性:**
  - 提供用户和服务器的**双向认证** (mutual authentication)。
  - 主要适用于客户端-服务器模型 (client-server model)。
  - 协议本身受到保护，防御窃听和重放攻击 (replay attacks)。

公钥基础设施 (Public Key Infrastructure, PKI)

公钥管理 (Public Key Management)

- **使用认证机构 (Certification Authority, CA):** CA 是一个可信的第三方 (trusted third party)，用于绑定公钥与其所有者。
- **证书 (Certificate):** 将公钥与其所有者信息绑定起来的数字凭证。
  - 包含信息：X.500 格式的名称、组织、地址、公钥、有效期等，并由 CA 签名。
  - 格式示例：`signCA[X.500: name, org, address, pub. key, expires, ...]`。
- **CA 的角色:**
  - Alice 将公钥发送给 CA (Trent)。
  - CA 为 Alice 生成证书。
  - Alice 将公钥和证书发送给 Bob。
  - Bob 使用 CA 的公钥验证证书的有效性。
  - Bob 使用 Alice 的公钥加密消息发送给 Alice。
- **CA 的信任:**
  - 每个人都必须能够验证 CA 的公钥。
  - CA 的根证书（Root CA certificates）通常预装在操作系统、浏览器等软件中。
  - CA 被认为是可信的，有能力为任何人签发证书。

公钥证书生成 (Public Key Certificate Generation)

- **流程:**
  1. Alice 生成一对公钥/私钥 (public/private keypair)。
  2. Alice 将公钥发送给 CA。
  3. CA 向 Alice 发起挑战，验证她是否拥有对应的私钥。
  4. CA 生成证书并发送给 Alice。
- **重要注意事项:**
  - CA 永远不会知道 Alice 的私钥。
  - 这对于前向保密性 (forward secrecy) 很重要。
  - CA 的泄露或被攻破仍然可能导致有人冒充 Alice。
- **获取证书:**
  - **域名验证 (Domain Validation, DV):** CA 通过检查邮件（如发送到 admin@）或 DNS TXT 记录等方式验证你是否控制该域名。
    - **问题:** 这些方法并不能真正证明你拥有该域名。
  - **扩展验证 (Extended Validation, EV):** CA 验证你通过了一系列身份验证标准。
    - 限制证书有效期以确保信息是最新的。
    - 比普通证书成本更高。
- **证书签名请求 (Certificate Signing Request, CSR):** 包含域名、公钥及其他相关细节的文件，用于发送给 CA 以申请证书。
- **证书链 (Certificate Chaining):** 根 CA 可以授权其他 CA 签署 SSL 证书，形成信任链。

证书吊销 (Certificate Revocation)

- **原因:**
  - Alice 的私钥被盗或泄露。

  - 她换了工作（或其可信度发生变化）。
- **问题:** 证书吊销是 CA 系统的一个主要问题。
- **方法:**
  - 设置**过期日期** (expiration date)。
  - 使用**证书吊销列表 (Certificate Revocation List, CRL)**，像“坏信用卡”列表一样分发。
  - 使用**在线证书状态协议 (Online Certificate Status Protocol, OCSP)**。
    - **工作方式:** 浏览器想要验证证书时，会向 CA 签发证书中嵌入的 OCSP URL 发出请求。OCSP 服务器发送一个带签名的响应，表明证书是否仍然有效。
    - **OCSP 的攻击问题:** OCSP 响应上的签名只覆盖部分数据，不包括响应状态。一个可能的 MITM 攻击者可以拦截 OCSP 请求并发送 `tryLater` 的响应状态。由于大多数浏览器会默默忽略这种状态，通信会继续而不发出警告。

SSL/TLS 协议

概述

- **HTTP 的不足:** 裸 HTTP 连接中的所有内容都以明文形式传输，容易受到消息篡改、窃听和 MITM 攻击。底层 TCP/IP 协议也存在许多缺陷。
- **SSL/TLS:** Transport Layer Security (TLS) 是 Secure Socket Layer (SSL) 的后继者，是一组用于安全通信的加密协议。
- **安全机制:**
  - 使用**非对称加密** (Asymmetric cryptography) 进行认证 (authentication)。
  - 使用**对称加密** (Symmetric encryption) 保证消息的机密性 (confidentiality)。
  - 使用**消息认证码** (MACs) 保证消息的完整性 (integrity)。
- **HTTPS:** HTTP 协议运行在 SSL/TLS 之上形成 HTTPS (HTTP Secure)。
- **可见信息:** 在 HTTPS 连接中，唯一可见的信息是连接的 IP 地址和 TCP 端口，以及发送和接收消息的大小。
- **应用范围:** 以前仅用于“安全操作”，现在建议默认在所有地方使用。

TLS 协议流程 (简化)

1. **客户端 → 服务器:** 发送支持的 SSL 版本、可用的密码套件（ciphers）及其他信息。
2. **服务器 → 客户端:** 返回选定的 SSL 版本、密码套件以及服务器证书。
3. **客户端验证服务器:** 客户端使用 CA 的公钥验证服务器证书的有效性，确保消息由该证书签名。
4. **生成和交换密钥:** 客户端（可能与服务器协同）生成会话的预主密钥（pre-master secret），使用服务器证书中的公钥加密后发送给服务器。
5. **随机数:** 交换的消息中包含随机比特字符串以防止重放攻击 (replay attacks)。
6. **可选的客户端认证:** 服务器可以使用客户端证书对客户端进行认证（可选步骤）。
7. **加密通信:** 从此时起，所有消息都被加密。

CA 的问题 (针对 SSL/TLS)

- **信任问题:** CA 的安全性实践往往不佳，且愿意与政府合作。
- **根 CA 数量少:** 少量的根 CA 允许其他 CA 代表它们签名，增加了风险。
- **欺诈证书:** 2011 年，Comodo 被攻击者获取了 Google、Yahoo! Mail 等 9 个域名的欺诈证书，可能被用于 MITM 攻击。

- **成本问题:** 128 位和 256 位证书成本明显不同，但额外工作很少。
- **虚假安全感:** “此网站由 SSL 保护”容易给人虚假的安全感，骗子很容易为自己拥有的域名获取 SSL 证书。

证书吊销 (在 SSL/TLS 中的 OCSP 问题)

- **OCSP 问题:** 浏览器使用 OCSP 检查证书有效性时，攻击者可以拦截 OCSP 请求并发送 `tryLater` 响应。由于多数浏览器默认忽略此响应，连接仍会继续，不会引发警告。

对 SSL/TLS 的攻击 (Attacks on SSL/TLS)

- **BEAST (Browser Exploit against SSL/TLS):**
  - **漏洞:** 利用了 TLS 1.0 中 CBC 模式的协议缺陷。
  - **影响:** 如果攻击者能快速读取和注入数据包，就能窃听使用 *-CBC（如 AES-CBC）的 SSL 连接，可在局域网（如 Wi-Fi）内实施。
  - **修复:** 在 TLS 1.1 (2006) 中修复，但未广泛采用。TLS 1.2 和 1.3 不受 BEAST 影响。
  - **缓解方法 (TLS 1.0):**
    * 使用 RC4 (流密码)，不依赖 CBC，速度更快。**但后来 RC4 被发现不安全**。
    * 更改分组密码工作模式 - TLS 1.0 只支持 CBC，不可行。
    * 插入空数据包消耗不安全的初始化向量 (initialisation vectors) - TLS 1.0 未规定，引起兼容性问题。
    * 不完整数据块用随机数据填充到块大小。
- **CRIME (Compression Ratio Info-leak Made Easy):**
  - **漏洞:** 利用了先压缩后加密的设计缺陷。
  - **原理:** 加密载荷的大小是可见的，压缩使得大小泄露了很多关于载荷的信息。
  - **攻击方式:** 当协议先压缩再加密时，攻击者可以通过控制消息的一部分（例如注入可控字符串 `a`）并观察压缩后加密消息的大小变化，来推断消息中未知部分（例如认证 token `C`，如 Cookie）的内容。如果 `C` 和 `a` 有共同部分，压缩效率会更高，导致加密后的大小变小。攻击者通过尝试不同的 `a` 来猜测 `C` 的内容。
  - **场景:** Cookie 常用于认证，随每个 HTTP 请求发送，攻击者可控制请求的一部分。
  - **示例:** 攻击者猜测 `sessionKey=j` 是否在 `C` 中，如果存在，压缩 `C || "sessionKey=j"` 的效果比 `C || "sessionKey=a"` 好，加密后大小会变小。
- **POODLE (Padding Oracle On Downgraded Legacy Encryption):**
  - **漏洞:** 针对 SSLv3 的填充填充神谕攻击 (padding oracle attack)，利用 CBC 模式的填充设计缺陷和 MAC-then-Encrypt 顺序。
  - **SSLv3 填充问题:** CBC 模式下需要填充字节，最后一个填充字节的值是已知的（取决于消息长度）。但 SSLv3 没有指定其他填充字节的内容，服务器也不检查它们。
  - **MAC-then-Encrypt:** SSLv3 使用先计算 MAC 再加密的顺序，而不是推荐的先加密再计算 MAC。
  - **攻击方式:** 攻击者可以诱导 SSLv3 连接发生降级，然后利用填充缺陷和 MAC 顺序，让服务器解密任意数据块。服务器只有在填充的最后一个字节正确时才会接受该消息。攻击者可以通过暴力猜测填充的最后一个字节（平均 256 次猜测）来恢复数据块中的一个字节，然后逐步恢复整个数据。
- **Heartbleed:**
  - **漏洞类型: 不是协议漏洞**，而是 OpenSSL 库中的**特定实现漏洞**。该库被广泛使用。
  - **原因:** 利用了 TLS 的 Heartbeat 扩展 (RFC 6520)，该扩展允许发送心跳消息来保持连接活跃。心

跳消息格式是（消息长度，消息）。
  - **实现错误:** OpenSSL 错误地信任了客户端报告的消息长度，没有进行边界检查。
  - **攻击方式:** 攻击者发送一个小的消息，但在心跳请求中声称消息长度很大，OpenSSL 会读取并返回比实际消息大得多的内存区域的内容，泄露服务器内存中的敏感信息（如私钥、用户会话数据等）。Heartbleed 是缺乏边界检查的经典错误。

知识点可能考法及示例考题

- **理解基本概念:**
  - **考点:** Diffie-Hellman 密钥交换的原理、步骤及其 MITM 攻击方式。完美前向保密（PFS）的定义、重要性以及哪些协议（或配置）提供/不提供 PFS。已知密钥攻击的定义。
  - **示例考题:**
    - 请解释 Diffie-Hellman 密钥交换的基本流程，并描述如何进行中间人攻击（MITM）。**参考答案:** 描述 DH 交换过程，并详细解释 Eve 如何拦截通信、与双方分别建立秘密共享，从而窃听和篡改消息。
    - 什么是完美前向保密（Perfect Forward Secrecy）？为什么 Diffie-Hellman 提供 PFS，而使用 RSA 公钥直接加密会话密钥则不提供？**参考答案:** 给出 PFS 定义，解释 DH 的临时密钥性质，说明 RSA 直接加密会话密钥时，长期私钥泄露会导致历史会话被解密。
- **协议细节及优缺点:**
  - **考点:** Needham-Schroeder 协议的流程、依赖的可信方以及它的主要问题（密钥新鲜度、密钥泄露影响、密钥吊销困难）。Kerberos 与 NS 协议的关系以及其提供的核心安全特性（双向认证、防御重放）。
  - **示例考题:**
    - 请详细描述 Needham-Schroeder 协议的主要步骤。该协议存在哪些安全问题？**参考答案:** 依次列出五步流程。说明其密钥新鲜度问题和依赖 TTP 进行密钥吊销的困难性，以及长期密钥泄露的严重后果。
    - Kerberos 协议解决了 Needham-Schroeder 协议中的哪些问题？它提供了哪些额外的安全特性？**参考答案:** 说明 Kerberos 在 NS 基础上提供双向认证，并防御窃听和重放攻击。
- **PKI 与证书管理:**
  - **考点:** CA 的角色和信任模型。证书的构成和生成流程。证书吊销的原因、重要性及主要方法（CRL, OCSP）。CA 系统的问题（信任、欺诈证书、OCSP 攻击）。不同类型的证书验证（DV, EV）。
  - **示例考题:**
    - 请解释公钥基础设施（PKI）中认证机构（CA）的作用。用户如何验证服务器证书的有效性？**参考答案:** 说明 CA 绑定公钥和身份，是可信第三方。描述 Bob 如何使用 CA 公钥验证 Alice 证书。说明浏览器预装 CA 证书并验证服务器证书的签名。
    - 证书吊销为何重要？请列举至少两种证书吊销的方法，并简述它们的原理。**参考答案:** 说明私钥泄露等原因导致吊销需求。解释 CRL 是列表，OCSP 是在线查询。可以提及 OCSP 的攻击问题。
    - 描述域名验证（DV）和扩展验证（EV）证书的区别。CA 系统存在哪些普遍问题？**参考答案:** 区分 DV 和 EV 的验证严格程度和成本。列举 CA 的信任、欺诈证书、成本等问题。
- **SSL/TLS 协议及其安全性:**
  - **考点:** HTTP 的不足。SSL/TLS 提供的安全服务（机密性、完整性、认证）及其使用的加密类型。HTTPS 与 HTTP 的区别。TLS 握手流程中的关键步骤（交换 cipher suite, 证书验证, 密钥交换）。常见的 SSL/TLS 攻击（BEAST, CRIME, POODLE, Heartbleed）的原理和影响，以及它们是协议漏洞还是实现漏洞。

- **示例考题:**
  - 解释 SSL/TLS 协议如何提供机密性、完整性和认证服务，并说明分别使用了哪类加密技术。HTTPS 相比 HTTP 有哪些安全优势？**参考答案:** 说明使用对称加密提供机密性、MAC 提供完整性、非对称加密进行认证。说明 HTTPS 在 SSL/TLS 层对 HTTP 流量进行加密，防止窃听和篡改。
  - 请描述 TLS 握手过程中的主要步骤，特别是服务器证书的验证和会话密钥的建立过程。**参考答案:** 描述客户端发送能力、服务器选择、服务器发送证书、客户端验证证书、客户端生成和发送预主密钥等过程。
  - 请简述 BEAST、CRIME 和 POODLE 攻击的原理，并说明它们是针对 SSL/TLS **协议**的漏洞还是**实现**的漏洞。**参考答案:** 说明 BEAST 利用 TLS 1.0 CBC 填充，CRIME 利用压缩 + 加密，POODLE 利用 SSLv3 填充 +MAC 顺序。这些是**协议**漏洞。
  - Heartbleed 漏洞是针对 SSL/TLS 协议本身的漏洞吗？请解释其原理和影响。**参考答案:** 说明 Heartbleed 是 OpenSSL 库的**实现**漏洞。解释其利用 Heartbeat 扩展中缺乏边界检查的弱点，导致内存泄露。