期末提纲 2

1. **黑客入侵一台主机后，会做哪些操作？**

   作为跳板攻击局域网中的其他主机：

   黑客可以利用被入侵的主机作为跳板，进一步攻击局域网内的其他设备，扩大攻击范围。

   用主机来挖矿：

   黑客可能会在被入侵的主机上安装加密货币挖矿软件，利用主机的计算资源进行挖矿，从而获取经济利益。

   留后门：

   黑客会在被入侵的主机上安装后门程序，以便在未来能够轻松地重新访问该主机，保持对系统的控制。

   爬取数据：

   黑客可能会窃取主机上的敏感数据，如用户账户信息、密码、财务数据等，进行非法利用或出售。

   下载监控程序监控用户行为：

   黑客可能会安装监控软件，记录用户的键盘输入、屏幕截图、网络活动等，获取更多的敏感信息。

   敲诈勒索：

   黑客可能会加密主机上的数据，并要求受害者支付赎金才能解密数据，这种攻击被称为勒索软件攻击。

2. **外挂（也称为“作弊工具”）的本质是一个进程干扰另一个进程的执行。以下是外挂的主要操作和技术：**

操作

1. **改数据**：修改目标进程中的数据，例如游戏中的分数、生命值等。
2. **改代码**：修改目标进程中的代码，尤其是 `if` 等判断语句，以改变程序的逻辑。

推广：DLL 注入

描述：DLL 注入是一种常见的外挂技术，即将一个 DLL 文件注入到目标进程中，从而干扰其执行。

过程：

1. **创建 DLL**：编写一个包含恶意代码的 DLL 文件。
2. **注入 DLL**：使用工具或代码将 DLL 文件注入到目标进程中。
3. **执行恶意代码**：注入的 DLL 文件在目标进程中执行，干扰其正常运行。

DLL 攻击思想

修改 EXE 文件：直接修改可执行文件（EXE）以包含恶意代码。

修改 DLL 文件：修改目标进程使用的 DLL 文件，以注入恶意代码。

示例

假设有一个游戏进程 `game.exe`，外挂可以通过以下步骤进行干扰：

1. **编写 DLL 文件**：创建一个包含作弊代码的 DLL 文件 `cheat.dll`。
2. **注入 DLL 文件**：使用注入工具将 `cheat.dll` 注入到 `game.exe` 进程中。
3. **执行作弊代码**：`cheat.dll` 在 `game.exe` 中执行，修改游戏数据或逻辑。

3. **二进制分析**

二进制分析是一项复杂且需要耐心和毅力的技术，尤其是在处理 exe 和 dll 文件时。以下是一些关键点和工具：

工具

1. **OllyDbg 和 x64dbg**：

   OllyDbg：适用于 x86 架构的调试工具，功能强大，界面友好。

   x64dbg：适用于 x64 架构的调试工具，支持现代操作系统和应用程序。

2. **IDA Pro**：

   IDA Pro：交互式反汇编工具，支持多种架构和平台，广泛用于逆向工程和漏洞分析。

知识

1. **X86 机器码和汇编**：

   理解 x86 架构的机器码和汇编语言是进行二进制分析的基础。

   需要掌握寄存器、指令集、调用约定等知识。

2. **外挂技术**：

   WPM（Write Process Memory）：通过修改目标进程的内存数据，实现数据篡改。

   DLL 注入：将恶意 DLL 文件注入目标进程，干扰其执行。

修改 EXE 文件：直接修改可执行文件，注入恶意代码。

实践

1. **调试和分析**：

   使用 OllyDbg 或 x64dbg 调试目标进程，设置断点，观察寄存器和内存变化。

   使用 IDA Pro 反汇编目标文件，分析代码逻辑和结构。

2. **注入和修改**：

   编写 DLL 文件，使用注入工具将其注入目标进程。

   修改 EXE 文件，插入恶意代码或修改现有代码逻辑。

需要的品质

耐心：二进制分析需要反复调试和验证，过程可能非常耗时。

毅力：面对复杂的代码和未知的挑战，需要坚持不懈地探索和解决问题。

运气：有时找到关键漏洞或成功注入代码也需要一些运气。

4. **防范 加壳**

加壳技术的原理

1. **代码加密**：对程序的代码和数据进行加密，只有在运行时才能解密和执行。
2. **代码混淆**：通过插入无用代码和重命名符号，使程序的逻辑更加难以理解。
3. **反调试保护**：插入防调试代码，检测并阻止调试器的使用。

常见的加壳工具

1. **UPX（Ultimate Packer for eXecutables）**：一种流行的开源加壳工具，支持多种操作系统和文件格式。
2. **Themida**：一种高级加壳工具，提供强大的保护机制，防止逆向工程和调试。

防范措施

1. **代码签名**：使用数字签名对加壳后的程序进行签名，确保其来源的可信度和完整性。
2. **环境监测**：检测运行环境，防止在虚拟机或调试器中运行，增加破解难度。
3. **多层加壳**：对程序进行多次加壳，增加逆向工程的复杂性。
4. **定期更新壳**：定期更换加壳工具和策略，防止壳被破解。

加壳技术的局限性

尽管加壳技术可以有效地增加逆向工程和破解的难度，但它并不是万能的。有经验的攻击者仍然可以通过以下手段进行破解：

1. **静态分析**：使用反汇编工具（如 IDA Pro）分析加壳程序的逻辑。
2. **动态调试**：使用调试器（如 OllyDbg、x64dbg）在运行时分析和修改程序。
3. **脱壳工具**：使用专门的脱壳工具，自动去除加壳保护。

5. **HTTPS 和 SSH 的概念**

HTTPS（HyperText Transfer Protocol Secure）

定义：HTTPS 是一种在 HTTP 基础上添加了 SSL/TLS 加密协议的安全通信协议，用于保护数据在网络上的传输安全。

- **传输层**：HTTPS 工作在传输层，通过加密传输层的数据，确保数据的机密性、完整性和真实性。
- **公钥密码体制**：HTTPS 使用公钥密码体制（如 RSA）进行密钥交换，通过证书验证服务器身份，建立安全的加密通道。

SSH（Secure Shell）

定义：SSH 是一种用于安全登录和传输数据的网络协议，主要用于远程登录和命令执行。

- **应用层**：SSH 工作在应用层，提供加密的通信通道，保护数据在传输过程中的安全。
- **公钥密码体制**：SSH 使用公钥密码体制（如 RSA 或 DSA）进行身份验证和密钥交换，确保通信的安全性。

共同点和区别

共同点：

- **加密通信**：HTTPS 和 SSH 都提供加密通信，保护数据在传输过程中的安全。
- **公钥密码体制**：两者都使用公钥密码体制进行身份验证和密钥交换。

区别：

- **层级**：HTTPS 工作在传输层，而 SSH 工作在应用层。
- **用途**：HTTPS 主要用于保护网页传输的安全，SSH 主要用于安全远程登录和数据传输。

4. **内存安全问题**

1. **缓冲区溢出（Buffer Overflow）**：

   描述：当将数据写入缓冲区时，超过了缓冲区的长度，导致覆盖相邻内存区域，从而可能导致任意代码执行或崩溃。

   常见原因：

使用不安全的函数，如 `gets`、`scanf` 等，这些函数不检查输入数据的长度。

旧函数设计缺陷

1. **老的 C 语言函数**：

   `gets`：不检查输入的长度，可能导致缓冲区溢出。

   `scanf`：在使用 `%s` 格式符时，不限制输入长度，可能导致缓冲区溢出。

函数无法修改的原因

1. **历史原因**：许多旧的代码库和系统依赖这些函数，直接修改它们会导致兼容性问题。
2. **重要性**：这些函数在早期的许多系统中被广泛使用，修改它们需要大量的测试和验证工作。

开发人员出错实例

1. **Heartbleed 漏洞**：

   背景：Heartbleed 是一个著名的漏洞，发生在 OpenSSL 库中，开发人员没有正确检查数据与其声明的长度是否一致。

   - **影响**：攻击者可以读取内存中的任意数据，包括敏感信息，如密码和密钥。

解决方案和防范措施

1. **使用安全函数**：

   替代函数：使用更安全的替代函数，如 `fgets` 代替 `gets`，`snprintf` 代替 `sprintf`。

   - **长度检查**：在使用输入函数时，明确指定输入的最大长度。

2. **静态和动态分析工具**：

   - **静态分析**：使用工具（如 `lint`、`cppcheck`）检查代码中的潜在问题和不安全的函数调用。
   - **动态分析**：运行时使用工具（如 `Valgrind`）检查内存问题和溢出。

3. **代码审查和测试**：

   - **代码审查**：定期进行代码审查，确保没有使用不安全的函数和存在的潜在漏洞。
   - **单元测试**：编写单元测试，特别是边界条件测试，确保输入处理的安全性。

4. **现代编译器和安全特性**：

   堆栈保护：启用编译器的堆栈保护功能（如 `-fstack-protector`），防止缓冲区溢出攻击。

地址空间布局随机化（ASLR）：启用 ASLR，增加攻击者预测内存地址的难度。

5. **网络攻击中如何使用口令哈希值查询网站**

在网络攻击中，攻击者可能会利用口令哈希值查询网站来发现是否有泄露的密码或用户名。这通常涉及通过已知的哈希值反向查找原始密码。以下是攻击者可能使用的步骤和方法：

1. 获取口令哈希值

攻击者首先需要从目标系统或数据库中获取加密存储的口令哈希值。这可以通过漏洞利用、社会工程或数据泄露来实现。

2. 使用在线口令哈希查询网站

有许多在线服务和数据库可以帮助攻击者查询口令哈希值是否已知和泄露。以下是一些常见的方法：

使用彩虹表（Rainbow Table）

描述：彩虹表是一种预计算的哈希值反查表，通过存储大量哈希值与对应的明文密码，快速查找匹配项。

- **步骤**：
  1. 下载或访问在线彩虹表。
  2. 输入口令哈希值。
  3. 通过彩虹表快速查找并匹配明文密码。

使用暴力破解工具

描述：如果上述方法无法找到匹配，攻击者可能会使用暴力破解工具，通过生成大量的密码候选并计算其哈希值来匹配目标哈希值。

- **工具**：常见的暴力破解工具包括 John the Ripper、Hashcat 等。

保护措施

为了防范这些攻击，以下是一些安全建议：

1. **使用强哈希算法**：选择安全的哈希算法，如 bcrypt、scrypt 或 Argon2，它们具有抗暴力破解和抗彩虹表攻击的能力。
2. **添加盐值（Salt）**：在密码哈希前添加独特的随机盐值，可以防止使用彩虹表进行反查。
3. **加密传输**：确保敏感数据在网络传输中使用加密协议（如 HTTPS）。
4. **多因素认证**：使用多因素认证（MFA），增加额外的安全层。

6. **匿名网络通信的方法**

1. 代理通信（Proxy Communication）

VPN（虚拟专用网络）：

- **定义**：VPN 是一种通过在公用网络上创建加密的通信通道，为用户提供匿名性和安全性的技术。VPN 会隐藏用户的真实 IP 地址，用 VPN 服务器的 IP 地址替代。
- **特点**：
  - **IP 地址分配**：VPN 通常会分配一个新的 IP 地址（例如 10.xxx）给用户，用于所有的网络通信。
  - **加密通信**：提供全局的加密通信通道，保护所有网络流量。
- **应用**：访问受限网站、保护在公共 Wi-Fi 上的通信安全等。
- **示例**：OpenVPN 是一种开源的 VPN 软件，支持多种加密协议和身份验证机制。

Proxy（代理服务器）：

- **定义**：代理服务器充当客户端和目标服务器之间的中介，用户的请求首先发送到代理服务器，再由代理服务器转发到目标服务器。代理服务器会替换用户的 IP 地址。
- **特点**：
  - **端口通信**：代理服务器通过特定端口进行通信，代理服务器使用不同的端口来处理网络流量。
- **应用**：绕过地理位置限制、缓存加速访问等。
- **示例**：HTTP 代理、SOCKS 代理。

2. 端口转发（Port Relay）

描述：端口转发是一种网络通信技术，用于将一个网络端口上的流量重定向到另一个端口。通过端口转发，可以实现透明代理和数据中继，隐藏用户的真实 IP 地址。

- **应用**：远程访问、绕过防火墙、隐藏真实 IP。
- **示例**：SSH 隧道（`ssh -L`、`ssh -R`、`ssh -D`）用于创建加密的端口转发通道。

VPN 和 Proxy 的区别

- **IP 地址分配**：
  - **VPN**：VPN 通常会分配一个新的 IP 地址（如 10.xxx）给用户，所有的网络通信都会通过这个新的 IP 地址。
  - **Proxy**：代理服务器通过端口进行通信，用户的请求通过代理服务器的端口转发，不改变 IP 地址。
- **加密**：

VPN：提供全局的加密通信通道，保护所有网络流量，确保数据的机密性和完整性。

- **Proxy**：代理服务器通常只代理特定的网络流量，可能不提供加密。

综合比较

特点	VPN	Proxy
IP 地址	分配新的 IP 地址（如 10.xxx）	使用代理服务器的 IP 地址
通信范围	全局加密通信	仅代理特定流量
加密	提供全局加密	不一定提供加密
应用场景	访问受限网站，保护通信安全	绕过地理位置限制，加速访问

P2P文件共享

P2P文件共享

描述：P2P（点对点）文件共享是指通过 P2P 网络协议，在用户之间直接分享文件的方式，如 BT（BitTorrent）、eDonkey 等。

- **安全问题**：
  - **版权侵权**：非法共享受版权保护的文件可能导致法律问题。
  - **病毒和恶意软件传播**：P2P 文件共享可能成为恶意软件传播的途径。
  - **带宽占用**：P2P 共享通常占用大量带宽，影响网络性能。

IP限速

IP限速

描述：通过配置网络设备限制特定 IP 地址的带宽使用，防止某些设备占用过多带宽。

方法：

- **路由器或防火墙配置**：通过设置限速规则，确保公平的带宽分配。
- **动态限速策略**：根据网络使用情况，动态调整限速策略，优化网络资源利用。

工具：

- **Cisco、MikroTik 路由器**：支持高级限速和流量管理。

pfSense：开源防火墙和路由器软件，支持限速配置。

监控和管理工具

1. **Wireshark**

   功能：捕获和分析网络流量，查看网卡传输的数据。

   用途：分析具体会话，识别异常流量。

2. **ntopng**

   功能：宏观统计和监控网络流量，识别活跃 IP 和流量模式。

   用途：综合汇总网络流量，帮助管理员了解整体网络状况。