第二次 习题

周玉川

2017221302006

1、 请描述远程主动攻击模式和远程被动攻击模式之间的区别，并列出目前典型的漏洞 CVE 编号和攻击实例。

答:

模式	特点
远程主动攻击	若目标主机上的某个网络程序存在漏洞，则攻击者可能通过利用该漏洞获得目标主机的访问或控制权。也就是说，攻击者发起攻击，主动寻找找漏洞。例如常见的 ddos 攻击。
远程被动攻击	当一个用户访问网络上的一台恶意主机，就可能遭到目标主机发动的针对性恶意攻击。也就是说，由于 victim 主动接触到恶意代码，从而遭受到攻击。例如常见的勒索病毒。
它们两者的区别	很显然，他们的区别在于主动和被动上，也很好区分。远程主动攻击是攻击事件的发起者具有主动性，而远程被动攻击具有被动性，由于受害主机主动接触到恶意程序，才给攻击者可乘之机。

典型漏洞

CVE 编号	攻击实例
CVE-2018-0171	2018 年 4 月 7 号，一个名为“JHT”的黑客组织利用 Cisco（思科） CVE-2018-0171 （远程代码执行漏洞）攻击了许多国家的网络基础设施。全球超过 20 万台路由器受到了攻击影响，其中俄罗斯和伊朗的损失最大。

2、 请查阅并列举微软公布的其对 Windows 安全漏洞的分类和处理方式。

答:

Windows 将安全漏洞分为三大类分别是安全边界、安全功能以及纵深防御（defense-in-depth）安全功能。

安全边界:

安全边界	**安全目标**
网络边界	未经授权的网络端点无法访问或篡改客户设备上的代码和数据。
内核边界	非管理用户模式进程无法访问或篡改内核代码和数据。管理员到内核不是安全边界。
工艺边界	未经授权的用户模式进程无法访问或篡改另一个进程的代码和数据。
AppContainer 沙箱边界	基于容器功能的基于 AppContainer 的沙箱进程无法访问或篡改沙箱外部的代码和数据
用户边界	未经授权，用户无法访问或篡改其他用户的代码和数据。
会话边界	未经授权，用户登录会话无法访问或篡改另一个用户登录会话。
Web 浏览器边界	未经授权的网站不能违反同源政策，也不能访问或篡改 Microsoft Edge Web 浏览器沙箱的本机代码和数据。
虚拟机边界	未经授权的 Hyper-V 来宾虚拟机无法访问或篡改其他来宾虚拟机的代码和数据。这包括 Hyper-V 隔离容器。

虚拟安全模式边界	在 VSM Trustlet 或 Enclave 中执行的代码无法访问或篡改 VSM Trustlet 或 Enclave 中的数据和代码。

安全功能：

类别	安全功能	安全目标
设备安全性	BitLocker	关闭设备电源后，将无法获取磁盘上加密的数据。
设备安全性	安全启动	根据 UEFI 固件策略的定义，只有授权的代码才能在 OS 之前的版本中运行，包括 OS 加载程序。
平台安全性	Windows Defender 系统防护（WDSG）	签名不正确的二进制文件无法根据系统的应用程序控制策略执行或加载。该策略允许的绕过杠杆应用程序不在范围之内。
应用安全	Windows Defender 应用程序控制（WDAC）	只有符合设备策略的可执行代码（包括由开明的 Windows 脚本主机运行的脚本）才能运行。该策略允许的绕过杠杆应用程序不在范围之内。要求管理权限的旁路不在范围内。
身份和访问控制	Windows Hello /生物识别	攻击者无法通过欺骗，网络钓鱼或破坏 NGC（下一代凭据）来冒充用户。
身份和访问控制	Windows 资源访问控制	身份（用户，组）不能访问或篡改资源（文件，命名管道等），除非得到明确授权
密码学 API：下一代（CNG）	平台密码学	算法是按照规范（例如 NIST）实现的，不会泄漏敏感数据。
健康证明	主机监护人服务（HGS）	评估发出或保留下游加密操作所需的健康声明的呼叫者的身份和健康状况。
认证协议	认证协议	协议是按照规范实施的，攻击者无法篡改，泄露敏感数据或冒充获得较高特权的用户。

纵深防御（defense-in-depth）安全功能：

类别	安全功能	安全目标
用户安全	用户帐户控制（UAC）	未经管理员同意，防止不必要的系统范围内的更改（文件，注册表等）
用户安全	AppLocker	防止未经授权的应用程序执行
用户安全	受控文件夹访问	保护来自恶意应用的访问和对受控文件夹的修改
用户安全	网络标记（MOTW）	防止在本地查看时从网上下载活动内容提升特权
利用缓解措施	数据执行保护（DEP）	攻击者无法从不可执行的内存（例如堆和堆栈）中执行代码
利用缓解措施	地址空间布局随机化（ASLR）	攻击者无法预测进程虚拟地址空间的布局（在 64 位上）
利用缓解措施	内核地址空间布局随机化（KASLR）	攻击者无法预测内核虚拟地址空间的布局（在 64 位上）
利用缓解措施	任意代码守卫（ACG）	启用了 ACG 的进程无法修改代码页或分配新的私有代码页

利用缓解措施	代码完整性保护（CIG）	启用 CIG 的进程无法直接加载签名不正确的可执行映像（DLL）
利用缓解措施	控制流防护（CFG）	CFG 保护的代码只能对有效的间接调用目标进行间接调用
利用缓解措施	子进程限制	启用此限制后，无法创建子进程
利用缓解措施	安全 SEH / SEHOP	异常处理程序链的完整性不能被颠覆
利用缓解措施	堆随机化和元数据保护	不能破坏堆元数据的完整性，并且攻击者无法预测堆分配的布局
利用缓解措施	Windows Defender 漏洞利用防护（WDEG）	允许应用启用其他深度防御防御缓解功能，从而使利用漏洞更加困难
平台锁定	保护过程灯（PPL）	通过开放流程功能防止非管理性非 PPL 流程访问或篡改 PPL 流程中的代码和数据
平台锁定	屏蔽虚拟机	有助于保护 VM 的机密及其数据免遭恶意结构管理员或主机上运行的恶意软件的攻击，从而免受运行时和脱机攻击

3、 请从分配方式、使用情况以及释放方式等方面来对比分析堆和栈的区别。

答：堆和栈的区别

一、在分配方式上：

栈由编译器自动分配，而堆由程序员分配。

二、使用情况上：

栈是系统提供的功能，特点是快速高效，缺点是有限制，数据不灵活；而堆是函数库提供的功能，特点是灵活方便，数据适应面广泛，但是效率有一定降低。

栈是系统数据结构，对于进程/线程是唯一的；堆是函数库内部数据结构，不一定唯一。不同堆分配的内存无法互相操作。

三、释放方式上：

栈由编译器释放，而堆一般由程序员释放，若程序员不释放，在程序结束时可能系统会释放。

四、内存增长方向上

堆由低地址到高地址，栈由高地址到低地址

五、所处位置

堆变化范围一般很大，栈变化范围比较小一般在 0x0010xxxx

4、 什么是 SQL 注入，其本质原因是什么，如何防范？

答：SQL 注入的概念：通过在输入中插入一段数据库查询代码，根据程序返回的结果，获得某些他想得知的数据。

SQL 注入的本质原因是没有对用户的输入进行合法性判断，给了用户意料之外的权力。

防范方法有以下几个：

a) 参数化查询

将数据与命令分开，访问数据库时切勿直接用字符串方式查询数据库，不要运行参数中的指令。

b) 过滤转换

过滤掉含有常用 SQL 关键字的输入参数

c) 服务器与数据库安全设置

给应用程序最低的权限，删除不必要的用户，对数据库进行升级和补丁。

5、反射型 XSS 和基于 DOM 的 XSS 的区别是什么？比较 XSS 和 CSRF 的异同点。简述什么是漏洞利用及实现一次成功的漏洞利用所需要具备的条件。

答:

一，反射型 XSS 和基于 DOM 的 XSS 的区别

基于 DOM 的 XSS 漏洞，由服务器响应到客户端的页面中并没有直接包含攻击的恶意代码，而是由客户端在运行时动态生成了最终执行的恶意脚本代码。

二，XSS 和 CSRF

不同点：XSS 利用站点内的信任用户，CSRF 通过伪装来自受信任用户的请求来利用受信任的网站。

相同点：都可以利用用户的 cookie 进行攻击。

三，漏洞利用及实现一次成功的漏洞利用所需要具备的条件

漏洞利用：是黑客针对已有的漏洞，根据漏洞的类型和特点而采取相应的技术方案，进行尝试性或者实质性的攻击。常见的攻击类型有简单的命令，具体操作，恶意软件。

实现一次成功的漏洞利用必要条件：必须事先已知安全漏洞或者当场分析出安全漏洞，而且该种漏洞未被打上补丁。

6、简述常见的漏洞利用技术以及每种技术分别适用的场合。

答：常见的漏洞利用技术有

1) 修改内存变量：适用于需要改变内存变量的情景。
2) 修改代码逻辑：适用于破解软件，获得额外权限等场景。
3) 修改函数返回地址：适用于利用函数返回值的场景。
4) 修改函数指针：执行自己定义的代码等场景。
5) 攻击异常处理机制：使程序失效，意外停止或者跳转到预料中的位置执行。
6) 修改 PEB 中线程同步函数入口：执行事先定义好的代码，跳转到另一个进程。

一般漏洞利用技术会相结合着使用，才能达到目的。

7、简要说明在 Shellcode 中进行 API 自搜索和代码重定位的作用和过程。

一、地址重定位

a) 作用：指令使用相对于基址的相对地址作为偏移地址，可以通过加上基址获得指令的真实地址，这样就更加灵活便捷，而且使用偏移地址字节较少，速度快，占用内存少，若移动程序，只需要改变基址的位置，十分灵活方便。
b) 过程：以 call NEXT 为例，这条指令会把下一条指令的偏移地址入栈，若获得下一条指令的真实地址，只需要加上基址就可以获得真实地址。

二、API 自搜索技术

a) 作用：由于不同操作系统的动态链接库的加载地址不同，需要去遍历全部地址，获得目标 api 地址。
b) 过程
   i. 通过 FS 取得 TEB 的地址，记为 TEB_ptr。
   ii. PEB 的地址 PEB_ptr = [TEB_ptr+0xoc]即存放在 TEB 偏 ox0c 的地方。
   iii. PEB_ptr 偏 0x1c 存放着 InInitializationOrderModulelist 的地址记为 IOM_ptr = [PEB_ptr+0x1c]
   iv. Kernel32.dll 的地址，也就是基地址，存放在 InInitializationOrderModulelist 偏 8 的地方，即 base_ptr = [IOM_ptr+8]
   v. base_ptr 偏 0x3c 即位 PE 的头的偏移地址，从这一步开始利用到地址重定位技术，要加上基地址，PE_ptr = [base_ptr+0x3c] + base_ptr
   vi. PE 偏移 0x78 即位函数导出表的偏移地址。table_ptr = [PE_ptr+0x78] +

base_ptr

vii. 函数偏移地址（4 字节）的列表的偏移地址存放在 table 偏 0x1c 处

viii. 函数名(2 字节)列表的偏移地址存放在 table 偏 0x20 处

ix. 顺序遍历函数名，可找到该函数偏移地址，加上基址即可得到实际地址。

8、 简述漏洞挖掘方法中白盒测试、黑盒测试和逆向分析的异同。

答:

种类	相同	不同
白盒测试	包含静态测试和动态测试	检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。测试数据较为庞大。
黑盒测试		不关注软件的功能和逻辑，只在乎程序的健壮性。
逆向分析	既可以通过程序代码也可以根据程序结果进行逆向分析，同时结合了白盒和黑盒的一些方法，也包含静态分析，和动态分析	需要对代码进行反编译

9、 如果 Shellcode 中不允许出现特定字符，应当如何处理？请给出一种通用方法。

答：经过查阅资料，在这里给出一个较为简单的方法，但是该方法有一定局限性，他的优化的修改暂且不谈。

shellcode 的编码是简单的 ASCII 码，8 字节 0xXX。举例常见的 Shellcode 中不允许出现的特定字符，00 出现（0x00 表示字符串结尾，出现 00 会被截断），对于简单的编码可以采用异或 or 大法，将 00 与 0x66 异或相当于编码就变成了非零，解码时再与 0x66 异或可得到原程序，通过这种变换方法可以避免出现特定字符。