第 1 章 软件安全概论

1) 软件安全是一门对抗性学科
    a. 对抗双方是攻击者与防御者
    b. 武器是软件系统中存在的安全问题 - 漏洞或缺陷
2) 熟悉了解一些国际安全事件，如 SolarWinds 软件供应链安全攻击，WannaCry 勒索软件，log4j 漏洞
3) 牢记任何软件都是不安全的
    a. 为什么软件测试无法保证软件的安全性
    b. 在测试前尽量多地解决安全问题
    c. 业界公认事实：几乎所有的软件都是带着安全隐患投入运行
4) 软件不安全性的外在表现
    a. 运行不稳定，导致软件崩溃或非正常退出；
    b. 恶意攻击，达到信息窃取、系统破坏等目的；
5) 软件安全问题产生的原因
    a. 软件缺陷和错误
    b. 从软件开发者的角度看软件不安全的原因
        i. 软件开发没有严格遵守软件工程流程
        ii. 大多数软件，结构都相当大并且复杂，不易开发和维护
        iii. 开发者没有采用科学的编码方法
        iv. 测试不到位
    c. 我们要尽量减少安全问题，并通过防护手段尽可能减少安全事件的危害
    d. 熟悉 PPT 中展示的漏洞代码
6) 软件安全防护手段 （至少要熟悉三种）
    a. 安全设计与开发
    b. 保障运行环境
    c. 加强软件自身行为认证
    d. 恶意软件检测与查杀
    e. 黑客攻击防护
    f. 系统还原
    g. 虚拟隔离
    h. 内存安全语言
    i. 形式化验证

第 2 章 软件安全基础知识

1) 恶意软件与系统引导联系
    a. 熟悉并掌握操作系统引导的过程
    b. 恶意软件可以在哪些阶段重新获取控制权
2) Linux 系统虚拟地址空间
    a. 程序在内存中的映像，即程序地址空间分布（画图展示）

b. 虚拟地址 vs 物理地址

3) Linux 系统权限管理
   a. Linux 系统文件权限（drwxrwxrwx 含义解释）
   b. Linux 系统进程权限
      i. 当用户运行可执行文件时，所启动的进程必须携带发起当前用户的身份信息才能够进行合法的操作
      ii. SetUID 含义
4) ELF 文件的结构
   a. 链接视图 vs 执行视图（画图）
   b. 程序头表与节头表的区别
   c. 常见的 ELF 节及其含义（至少掌握三种）
   d. ELF 文件与内存之间的映射
      i. 掌握文件偏移地址，虚拟内存地址，长度等概念
   e. 动态链接的优缺点（为何动态链接在运行时完成只需一次）

第 3 章 软件缺陷与漏洞机理基础

1) 软件漏洞定义及其三要素
2) 软件漏洞的基本信息（至少要掌握 5 种）
3) CVE & CNVD
4) 漏洞分类方法
   a. 漏洞成因分类以及 CWE
5) 软件漏洞生命周期及漏洞利用
6) 漏洞利用对软件系统的威胁 （至少要掌握 3 种）
7) 典型漏洞类型
   a. 内存安全漏洞（至少掌握三种）
      iii. 越界写/读 Out-of-Bound Write/Read
      iv. 缓冲区溢出 Buffer Overflow
      v. 整数溢出 Integer Overflow or Wraparound
      vi. 释放后使用 Use After Free
      vii. 空指针 Null Pointer Dereference
      viii. 条件竞争 Race Condition
      ix. 失控的资源消耗 Resource Consumption
   b. 网络安全漏洞
      i. 跨站脚本攻击 XSS
      ii. 注入类漏洞 Injection
      iii. 路径穿越 Path Traversal
      iv. CSRF & SSRF

第 4 & 4.1 章 缓冲区溢出

1) 缓冲区溢出（掌握三者区别）

a. 栈缓冲区溢出
b. 堆缓冲区溢出
c. 全局数据缓冲区溢出

2) 栈区域的概念
   a. 栈区的内容（参数，返回地址，局部变量等，注意 x86_32 与 x86_64 调用约定的区别）
   b. 栈操作
   c. 函数调用过程与栈分布图
   d. 栈溢出的根本原因
3) 栈溢出的利用方式
   a. 覆盖局部变量
   b. 覆盖返回地址
   c. 覆盖 SEH 中的 handler
   d. Shellcode 的概念与常见功能

第 4.2 章 堆缓冲区溢出

1) 堆区域的概念
   a. 堆的数据结构和管理（堆表+堆块）
      i. 堆表：空表和块表
      ii. 堆块：块首和块身
         1. 占用块 vs 空闲块
2) 堆溢出
3) 堆溢出的利用方式
   a. what→where 操作或 Dword Shoot

第 4.3 & 4.4 & 4.5 章 整数溢出及其他漏洞类型

1) 整数溢出
   a. 整数溢出的概念
   b. 整数溢出的典型表现形式（三种全部掌握）
   c. 整数溢出和缓冲区溢出的经典结合方式 （至少掌握第一种）
      i. malloc 的参数计算中出现整数溢出，导致后续的堆区域访问出现问题
      ii. 内存拷贝的 size 参数传递出现符号溢出，导致后续的缓冲区访问出现问题
   d. 整数溢出修复方法（除法）
   e. 整数防护手段（安全意识，避免隐患操作等）
2) 释放后使用及双重释放漏洞
   a. 释放后使用的产生步骤以及悬挂指针的产生
   b. 释放后使用的危害与利用方式
3) 格式化字符串
   a. 格式化字符串函数（printf）的栈结构图

b. 格式化字符串的危害
    i. 越界读操作 ---> 数据泄露 （Information leak）
    ii. 越界写操作 ---> 数据破坏（Data Corruption），注意**“%n”**
c. 格式化字符串防御

第 5.1 & 5.2 章

1) 漏洞利用
    a. 漏洞发现 （0-day vs 1-day vs N-day）
    b. Payload vs Shellcode （Payload 与漏洞关联，Shellcode 独立于漏洞）
    c. 漏洞利用的目标（至少列出其中三种）
    d. 漏洞利用的整体过程
    e. Shellcode 的设计与编写
2) Return-to-libc (ret2libc)
    a. system(“/bin/sh”) + exit() 的栈结构设计 （重点考察，注意 x86_32 和 x86_64 之间的区别）
    b. ASCII Armoring & Ret2PLT
3) Return-Oriented-Programming（ROP）
    a. ROP 与 ret2libc 之间的区别
    b. 理解并掌握二进制代码重用示例 （重点考察）
4) 源代码安全审计
    a. 熟悉程序切片、符号执行、污点分析等概念
5) 静态分析技术 vs 动态分析技术
    a. 动静态之间区别
    b. 模糊测试 Fuzz Testing 概念

第 6 章 安全防护

1) 栈溢出保护 – Stack Guard / Stack Canary （重点考察）
    a. 防御机理及不足
    b. 绕过思路与 byte-by-byte 攻击
2) Fortify Source
    a. 防御机理及不足
    b. 绕过思路
3) 数据执行保护 – DEP / NX
    a. 防御机理及不足
    b. 绕过思路
4) 地址空间分布随机化 – ASLR
    a. 防御机理及不足
    b. 绕过思路