渗透测试工程师技术实战课

其他漏洞的原理以及分析

验证码安全

- 验证码的原理：区分计算机和人的区别的公共自动程序。
  - 多为：防止暴力破解、刷票、论坛灌水、爬虫等行为。
- 验证码分类
  - 操作验证码
    - 用户暴力破解
    - 高频次的接口访问
    - 二次确认
  - 身份验证码
    - 验证操作人身份
    - 密码修改
    - 账户变更
    - 其他重要操作

操作验证码

- 验证码可重用 （特定账户暴力破解、CSRF）
- 验证码可识别 （特定账户暴力破解）
- 验证码在客户端生成、显示、校验 （特定账户暴力破解、CSRF）
- 空验证码绕过 （特定账户暴力破解、CSRF）
- 验证码数量有限 （特定账户暴力破解）
- 是否校验可控
- 超过次数才开启验证码（撞库）

身份验证码

- 验证码返回给客户端
- 业务流程缺陷
- 验证码无时间间隔限制
- 验证码可爆破
- 验证码在客户端生成

命令执行漏洞

- 命令执行漏洞是指应用有时需要调用一些执行系统命令的函数
  - system()
  - exec()
  - shell_exec()
  - eval()
  - passthru()
  - 代码未对用户可控参数做过滤，当用户能控制这些函数中的参数时，就可以将恶意系统命令拼接到正常命令中，从而造成命令执行攻击。

命令执行漏洞

- 应用调用执行系统命令的函数
- 将用户输入作为系统命令的参数拼接到了命令行中
- 没有对用户输入进行过滤或过滤不严

- 漏洞的分类
  - 代码层过滤不严
  - 系统的漏洞造成命令注入
  - 用的第三方组件存在代码执行漏洞

命令执行漏洞

- 常见连接符
  - A;B　　先执行A，再执行B
  - A&B　　简单拼接，A B之间无制约关系
  - A|B　　显示B的执行结果
  - A&&B　A执行成功，然后才会执行B
  - A||B　　A执行失败，然后才会执行B

中间件漏洞

- Struts2命令执行
- Tomcat漏洞在渗透测试
- 其余
  - IIS漏洞
  - Apache漏洞

漏洞修复

- 尽量少用执行命令的函数或者直接禁用
- 参数值尽量使用引号包括
- 在使用动态函数之前，确保使用的函数是指定的函数之一
- 在进入执行命令的函数/方法之前，对参数进行过滤，对敏感字符进行转义
- 使用新版本

谢谢观赏