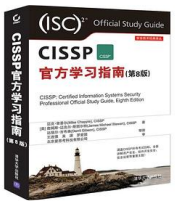


笔记—【OSG】15章：安全评估与测试

全书结构

本书的组织结构

本书涵盖 CISSP 通用知识体系的 8 个域，其深度足以让你清晰掌握相关资料。本书的主体由 21 章构成。域和各章的关系说明如下。

第 1~4 章：安全与风险管理。

第 5 章：资产安全。

第 6~10 章：安全架构和工程。

第 11 章和第 12 章：通信与网络安全。

第 13 章和第 14 章：身份和访问管理(IAM)。

第 15 章：安全评估与测试。

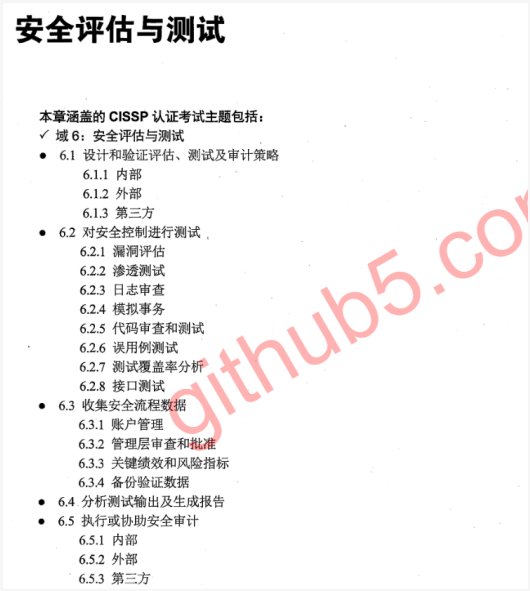
第 16~19 章：安全运营。

第 20 章和第 21 章：软件开发安全。

每章包含的元素可帮助你归纳学习重点和检验你掌握的知识。有关每章所涵盖域主题的详情，请见本书目录和各章介绍。

笔记—【OSG】15章：安全评估与测试

本章覆盖CBK内容



安全评估与测试

本章涵盖的 CISSP 认证考试主题包括：

✓ 域 6：安全评估与测试

- 6.1 设计和验证评估、测试及审计策略
  - 6.1.1 内部
  - 6.1.2 外部
  - 6.1.3 第三方
- 6.2 对安全控制进行测试
  - 6.2.1 漏洞评估
  - 6.2.2 渗透测试
  - 6.2.3 日志审查
  - 6.2.4 模拟事务
  - 6.2.5 代码审查和测试
  - 6.2.6 误用例测试
  - 6.2.7 测试覆盖率分析
  - 6.2.8 接口测试
- 6.3 收集安全流程数据
  - 6.3.1 账户管理
  - 6.3.2 管理层审查和批准
  - 6.3.3 关键绩效和风险指标
  - 6.3.4 备份验证数据
- 6.4 分析测试输出及生成报告
- 6.5 执行或协助安全审计
  - 6.5.1 内部
  - 6.5.2 外部
  - 6.5.3 第三方

💡 [1-4] 构建安全评估和测试方案

1.

安全评估和测试方案 (Program) 是信息安全团队的基础维护活动。

> 包括：**安全测试、安全评估、安全审计**

定期验证组织是否已采取**足够的安全控制**，及这些安全控制**是否正常运行并有效地保护信息资产**。

2.

安全测试是验证某项控制措施是否正常运行。需要定期执行。

> 包括：**自动化扫描、工具辅助的渗透测试、试图破坏安全的手动测试**

信息安全管理者（information security manager）需要考虑的一些因素：

> - 安全测试资源的可用性
> - 待测控制措施所保护系统及应用程序的重要性(Criticality)
> - 待测系统及应用程序所含信息的敏感性
> - 执行控制措施的机制出现技术故障的可能性
> - 关乎安全的控制措施出现错误配置的可能性
> - 系统可能遭受攻击的风险
> - 控制措施配置变更的频率
> - 技术环境下可能影响控制措施性能的其他变更
> - 执行控制措施测试的难度及时间
> - 测试对正常业务操作造成的影响

【业务场景考点】

信用卡处理系统，安全测试可以每年开展一次。（开展频率为考点）

使用新工具的想法固然不错，但是应该仔细设计安全测试方案，采用**"优先考虑风险"的方法对系统进行严格的、例行的测试**。

3.

安全评估是对系统、应用程序或其他待测环境的安全性进行全面审查。

> 不限于自动化扫描和手动渗透测试，**还包括对威胁环境、当前和未来的风险、目标环境价值的细致审查**。
>
> 主要工作成果通常是**向管理层提交的评估报告**。非技术语言描述、包含提高安全性的建议。
>
> 安全评估可以由**内部团队**或**有经验的第三方评估团队**执行。

实施隐私和安全评估的**最佳实践**：美国国家标准与技术研究院 (National Institute of Standards and Technology，NIST），**'NIST SP 800-53A'：联邦信息系统中的安全控制评价指南**。

> 根据 NIST 800-53A，评估包括 4 个组成部分：
> - 规范(Specification)是与待审计系统有关的文档。规范通常包括政策、规程、要求、详细及设计。
> - 机制是信息系统中用于满足规范的控制措施。机制可以基于硬件、软件或固件。
> - 活动是在信息系统中人员所采取的行动。这些行动可能包括执行备份，导出日志文件或审查账户历史记录。
> - 人员是指执行规范、机制及活动的人员。
>
> 在进行评估时，评估人员可检查此处列出的四个组件中的任意一个。他们也可以采访个体并进行直接测试来确定控制措施的有效性。

4.

安全审计 vs. 安全测试与评估

> **安全审计**是为了**向第三方证明控制措施有效性**而进行的评估。
>
> **安全测试与评估**仅供**内部使用**，旨在**评估控制措施**，并**发现改进空间**。

审计员 (Auditor) 为组织的安全控制状态提供一种客观中立的视角。他们撰写的报告与安全评估报告非常相似，但**适用千不同的受众，可能包括组织的董事会、政府监管机构和其他第三方**。

审计有三种主要类型：**内部审计、外部审计和第三方审计**。

> **内部审计**是由组织内部审计人员执行，通常**适用于组织内部（管理用途）。**审计负责人直接向类似总裁、CEO、董事会汇报。
>
> **外部审计**通常由外部审计公司执行。由于审计员与组织没有利益冲突，所以外部审计具有**很高的公信力**。
>
> > 四大审计公司：安永 (Ernst & Young)、德勤 (Deloitte & Touche）、普华永道（PricewaterhouseCoopers）、毕马威（KPMG）。
>
> **第三方审计**是由另一个组织（如：**监管机构**），或以另一个组织的名义进行的审计。**向其他组织提供服务的组织**通常要求进行第三方审计。
>
> > **美国注册会计师协会** (American Institute of Certified Public Accountants, **AICPA**) 发布一个减轻这类负担的标准。**第 16 号认证业务标准声明** (The Statement on Standards for Attestation Engagements document 16, **SSAE 16**) 提供一个**通用标准**。

审计标准描述了需要满足的控制目标，审计或评估的目的就是确保组织正确实施控制措施来实现这些目标。

> **信息和相关技术控制目标** (Control Objectives for Information and related Technology, **COBIT**) 是一个**开展审计和评估的通用框架**。 COBIT 描述了组织围绕其**信息系统所应具备的通用要求**。
>
> **国际标准化组织 (ISO)**： **ISO 27001** 描述了建立信息安全管理系统的标准方法，而 **ISO 27002** 则详细介绍了信息安全控制的细节。

💡 **[5-7] 开展漏洞评估**

5

漏洞评估是信息安全专业人员手中最重要的测试工具之一，包括：**漏洞扫描**和**渗透测试**。

CISSP语境下，漏洞评估是安全测试的工具（手段）之一，而不是安全评估的工具。另一个安全测试技术/工具是**软件测试**。

NIST 为安全社区提供**安全内容自动化协议** (Security Content Automation Protocol, **SCAP**) 以统一**漏洞描述**标准。

> **通用漏洞披露** (Common Vulnerabilities and Exposures, **CVE**)： 提供一个描述安全漏洞的命名系统。
>
> **通用漏洞评分系统** (Common Vulnerability Scoring System, **CVSS**)： 提供一个描述安全漏洞严重性的标准化评分系统。
>
> **通用配置枚举** (Common Configuration Enumeration, **CCE**)： 提供一个系统配置问题的命名系统。
>
> **通用平台枚举** (Common Platform Enumeration, **CPE**)： 提供一个操作系统、应用程序及设备的命名系统。
>
> **可扩展配置检查表描述格式** (Extensible Configuration Checklist Description Format, **XCCDF**)： 提供一种描述安全检查表的语言。
>
> **开放漏洞评估语言** (Open Vulnerability and Assessment Language, **OVAL**)： 提供一种描述安全测试过程的语言。

6

漏洞扫描可自动探测系统、应用程序及网络，探测可能被攻击者利用的漏洞。无需人工干预即可执行复杂的扫描测试任务。主要分为4类：

> **(1)网络发现扫描，**运用多种技术扫描一段IP，探测存在开放网络端口的系统。通常不探测系统漏洞。
>
> > **TCP SYN 扫描**，发送 SYN 标志位的数据包，请求创建一个新 TCP 连接，观察是否收到SYN/ACK回复。也称为“**半开放 ”(half-open) 扫描**。
> >
> > **TCP Connect 扫描**，远程系统的某个端口创建**TCP全连接**。
> >
> > **TCP ACK 扫描**，发送 ACK 标志位的数据包，表明它属于某个开放连接。以尝试确定防火墙规则或防火墙方法。

> **Xmas 扫描**，发送设置 FIN 、 PSH 及 URG 标志位的数据包。
>
> ---
>
> 常见工具：**Nmap**
>
> > 端口状态 **开放（open）**：目标端口开发，存在监听该端口的应用并且接受连接请求。
> >
> > **关闭（closed）**：目标端口被防火墙允许，但系统无监听该端口的应用。
> >
> > **过滤（filter）**：存在防火墙干扰，无法确认确切状态。
> >
> > ---
> >
> > 常见端口：80/HTTP，22/SSH，443/HTTPS，**1433/SQL Server**，3389/RDP，3306/MySQL等

(2)网络漏洞扫描，不止步于探测开放端口，会继续探测目标系统或网络，发现是否存在已知漏洞。

> 使用漏洞测试用例数据库，可能存在**误报 (false positive report)** 、**漏报 (false negative report)** 。
>
> 扫描工具默认使用未经过身份认证的扫描，会限制发现漏洞的能力；**执行经过身份认证的扫描可有效减少误报和漏报**。
>
> ---
>
> 常见工具：**Nessus**、商业扫描器有 Qualys 公司的 **QualysGuard** 和 Rapid7 公司的 **NeXpose**、开源扫描器 **OpenVAS**；**Aircrack**是一种常见的无线网络安全评估工具。

(3)Web应用漏洞扫描

> Web应用程序特点：需要向互联网用户提供服务、防火墙及其他安全设备通常允许Web流量通过、通常具有访问底层数据库的权限。
>
> Web应用漏洞扫描器通常**使用自动化技术控制输入及其他参数来识别 Web 应用漏洞**。
>
> ---
>
> Web漏洞扫描良好实践：
>
> > - 首次扫描时，扫描所有应用程序。可以检测遗留应用程序的问题。
> > - 任何新应用程序首次移植到生产环境前，必须进行扫描。
> > - 代码变更引入生产环境前，必须扫描任何修改过的应用程序。
> > - 定期扫描所有应用程序。
>
> ---
>
> **【业务场景考点】**
>
> **支付卡行业数据安全标准** (Payment Card Industry Data Security Standard, **PCI DSS**) 要求企业**至少每年执行一次 Web 应用漏洞扫描**，或**安装专业的 Web 应用防火墙**，增加针对 Web 漏洞的额外防护层。
>
> ---
>
> 虽然许多网络漏洞扫描器可以执行基本的 Web 应用漏洞扫描任务，但是深度的 Web 应用漏洞扫描仍需要定制的、专用的 Web 应用漏洞扫描器。**Nessus 是一种可执行两种扫描的复合工具**。
>
> 其它常见工具：商业扫描器 **Acunetix** 、开源扫描器 **Nikto** 和 **Wapiti**, 以及代理工具 **Burp Suite**。

(4)数据库漏洞扫描

> 攻击者可以利用Web应用程序来攻击后端数据库。如SQL注入。
>
> 数据库漏洞扫描器允许安全专业人员**扫描数据库和 Web 应用程序**，寻找影响数据库安全的漏洞。

> 常见工具：sqlmap

标志提取（banner grabbing）技术，扫描器通常使用该技术来辅助完成服务版本指纹识别。

漏洞管理工作流程

检测：通常在扫描漏洞之后，第一次发行某个漏洞。

验证：一旦扫描器检测到一个漏洞，管理员应该确认此漏洞，判断它是否为误报。

修复：验证过的漏洞需要加以修复。修复措施包括，

> 打补丁、修改配置、执行规范方案、部署WAF等阻止漏洞的控制措施。

7

渗透测试，安全专业人员尝试突破安全控制措施，入侵目标系统或应用来验证漏洞。

渗透测试过程：

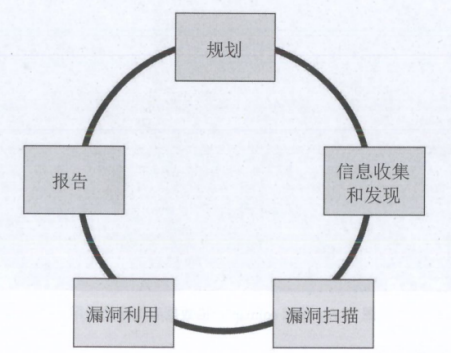


图 15.7　渗透测试过程

规划阶段包括测试范围和规则的协议。**测试团队和管理人员对测试性质达成共识，明确测试是经过授权的**。

信息收集和发现阶段结合人工和自动化工具来收集目标环境的信息。侦察、端口扫描等。

漏洞扫描阶段探测系统脆弱点，结合网络漏洞扫描、 Web 漏洞扫描和数据库漏洞扫描。

漏洞利用阶段试图**利用漏洞**尝试攻破系统安全防线。

报告阶段总结渗透测试结果，并提出改进系统安全的建议。

常见工具：**Metasploit** 使用脚本语言来实现常见攻击（漏洞利用）的自动化执行。

渗透测试类型:

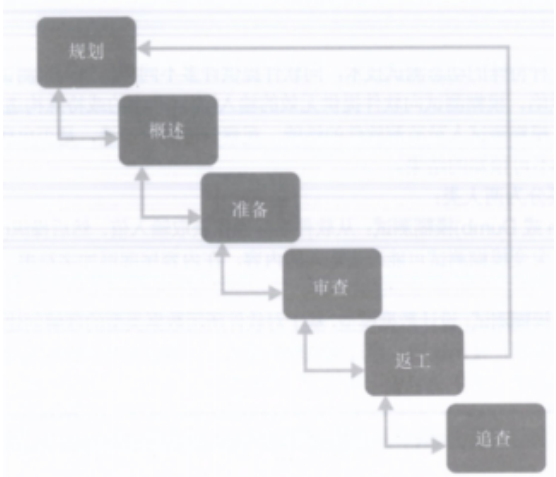
> **白盒渗透测试** (White Box Penetration Test)： 攻击者了解目标系统的详细信息。提升发现安全漏洞的可能性。
>
> **黑盒渗透测试** (Black Box Penetration Test)： 攻击之前不会向测试人员透露任何信息。用来模拟外部攻击者。
>
> **灰盒渗透测试** (Gray Box Penetration Test)： 也称为**部分知识测试**，平衡黑盒与白盒的优缺点。

> **行业标准**：OWASP **测试指南**、 OSSTMM 、 NIST 800-115 、 FedRAMP **渗透测试指南**、 PCI DSS 关于渗透测试的信息补充。

💡 **[8-11] 测试软件**

8

代码审查与测试是软件测试方案最关键的组成部分。在应用上线前发现安全、性能、可靠性等方面的问题。

> **代码审查**，也被成为**同行评审（peer review）**，由编写代码外的其他开发人员审查代码是否存在缺陷。
>
> > **最正式的**代码审查过程称为**范根检查法（Fagan inspection）**：**规划（第一步）**、概述、准备、审查、返工、**追查（最后一步）**。该级别的审查通常只用于严格限制的研发环境中，这里代码缺陷将会造成灾难性的危害。
> >
> > 
> >
> >
> > **稍微宽松的审查流程**
> >
> > > 同行共同进行走查代码（walk through）；
> > >
> > > 高级开发人员执行手动代码审查，上生产前签署所有代码；
> > >
> > > 上生产前所有代码使用自动化代码审查工具，检测常见缺陷。

静态测试，在**不运行软件的清况下**，通过分析软件源代码或编译后的应用程序，评估软件的安全性。常使用自动化工具检测常见软件缺陷，如缓冲区溢出。

动态测试，在**软件运行环境下**检测软件的安全性，如果组织**部署他人开发的软件，这将是唯一选择**。常使用漏洞扫描工具，如Web应用漏洞扫描工具。

> 动态测试可能使用**模拟事务** (synthetic transactions) 在内的方法，从而验证系统的性能。通过自动化脚本执行模拟事务，并比对事务输出与预期结果。

模糊测试（Fuzz）,是一种**特殊的动态测试技术**，向软件提供许多不同类型输入来测试其限制，发现之前未检测到的缺陷。观察软件是否崩溃，是否出现缓冲区溢出或其他不可取和（或）不可预知的结果。

> **突变 (Mutation 或 Dumb) 模糊测试**：通过**操纵真实输入值来生成模糊输入**。
>
> > 比特反转（bit flipping），一种略微控制输入的过程，通常进变化数个bit。

> **预生成（智能）模糊测试**：设计数据模型，基于对软件所用数据类型的理解创建新的模糊输入。或者说，**根据预期输入模型来生成输入**。
>
> **局限性：**模糊测试**通常不能完全覆盖所有代码**，一般仅限于检测**不涉及复杂业务逻辑**的简单漏洞。
>
> ---
>
> 常见工具：**zzuf工具**可根据用户使用说明，通过操纵软件输入，实现突变模糊测试自动化。

9

接口测试依据接口设计规范评估模块的性能，以确保模块在所有开发工作完成时可以协同工作。

> 需要测试的接口分为3种类型：
>
> > **应用编程接口**（Application Programming Interface，**API**）:为代码模块之间交互提供统一方法，通常通过 Web 服务形式向外部公开。
> >
> > **用户界面 (User Interface, UI)**：包括图形用户界面 (Graphic User Interface, GUI) 和命令行界面。
> >
> > **物理接口**：存在千操作机械装置、逻辑控制器或其他物理设备的一些应用程序。

误用例测试，通过测试已收集的已知误用例，来发现系统是否存在已知风险相关的漏洞。

10

测试覆盖率分析，以评估对软件测试的程度。一个非常主观的计算公式：

$$\text{测试覆盖率} = \frac{\text{已测试用例的数量}}{\text{全部用例的数量}}$$

> 测试覆盖率分析公式适用于许多不同标准。下面是五个常见标准：
>
> > **分支覆盖率**：在所有 if 和 else 条件下，每个 if 语句是否已被执行？
> >
> > **条件覆盖率**：在所有输入集合下，代码中每个逻辑是否已被测试？
> >
> > **函数覆盖率**：代码中每个函数是否已被调用并返回结果？
> >
> > **循环覆盖率**：在导致代码执行多次、一次或零次的条件下，代码中每个循环已被执行？
> >
> > **语句覆盖率**：是否已经运行过每行代码？

11

安全专业人员也经常参与**网站监测**，从事性能管理、故障排除、潜在安全问题识别等活动。

> **被动监测**：通过捕获发送到网站的实际网络流量，进行分析；帮助管理员深入理解网络上发生的事情。
>
> > **真实用户监控 (Real User Monitoring, RUM)** 是一种被动监测的变体，监测工具重组单个用户的活动，追踪其与网站的交互。

> 被动监测仅在真实用户出现问题之后才能监测到。被动监测擅长解决用户识别的问题，因为可捕获到与问题相关的流量。

综合监测（或主动监测）：执行伪造的事务活动，从而评估网站的性能。综合监测可能忽略真实用户遇到的问题，但可在真正发生之前检测到问题。

💡 **[12-16] 实施安全管理流程**

12

健全的信息安全计划不仅应包含执行**安全评估和测试**，还应包括各种管理流程，旨在监督信息安全计划的有效运行：**为安全评估提供关键反馈环路**；**对内部威胁产生威慑作用**。

> 包括：**日志审查**、**账户管理**、**备份验证**、**关键性能和风险指标**。

13

日志审查

> 借助**安全信息和事件管理**（Security Information and Event Management, **SIEM**）工具实现日志审查工作自动化。
>
> > SIEM通常使用**syslog**功能收集信息，一些设备（包括Windows系统）可能需要安装第三方客户端来支持syslog；
> >
> > 管理员可通过Windows组策略对象（Windows Group Policy Objects，GPO）来部署日志策略；
> >
> > 利用NTP协议实现发送端与SIEM接收端时钟同步，保证多种来源的信息有一致的时间轴。
>
> **信息安全管理者还应该定期进行日志审查**，特别是对于敏感功能（如管理员的活动），确保特权用户不会滥用其职权。
>
> 在调查安全事件时，**网络流（NetFlow）日志**特别有用。这些日志提供了系统连接和传输数据量的记录。

14

账户管理审查 (account management review) 确保**用户仅保留被授予的权限**，并不发生未授权的修改。通常**针对所有特权账户进行全面审查**。

> 管理人员从系统管理员处收集特权、特权用户列表；
>
> 再从特权审批机构处获取授权用例的列表及其分配的权限；
>
> 最后比对两个清单，确保只有授权的用户具备访问系统的权限，并且每个用户的权限均不超出其授权。
>
> ---
>
> 验证已解雇的用户不保留对系统的访问权限。
>
> 确保随机抽样的情况下，组织可以采取抽样方式进行账号审查，以节省时间成本。

> 组织还可自动执行账户审核流程的一部分工作。许多**身份和访问管理** (Identity and Access Management, **IAM**) 供应商可提供账户审查工作流程提示管理员审查、维护用户账户的文档
> 并提供说明审查已完成的审计踪迹 (audit trail)

15

备份验证

管理人员应**定期检查备份结果**，保流程可**有效地运行并满足组织的数据保护需求**。这个过程可能涉及审查日志、审查散列值或请求系统或文件的实际恢复。

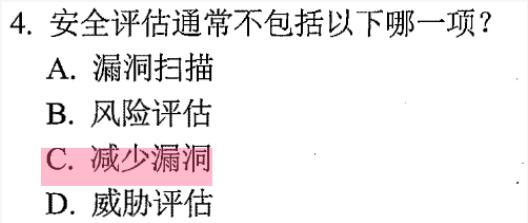
16

安全管理人员应**持续监测关键绩效和风险指标**。组织需要各自识别出自身需要跟踪的关键安全指标，为安全方案有效性提供高层次的视角。通常包括如下内容：

> 遗留漏洞的数量、修复漏洞的时间、漏洞/缺陷重现、被盗用账户的数量、在移植到生产环境前扫描过程中检测到的软件缺陷数量、重复审计的结果、尝试访问已知恶意站点的用户

17

💡 **书后习题易错分析**



安全评估包括识别漏洞的多种类型测试，而评估报告通常包含缓解建议。但是，评估不包括缓解这些漏洞危害的实际行动。此时A并不是安全测试语境下的“漏洞扫描”，安全测试场景下，漏洞发现后经过验证，即需要进行修复。而安全评估场景，通常包括发现并报告风险，以及风险缓解建议。