- 身份认证：客户端调用服务器提供的 API 接口需要进行身份认证，客户端判断是否是合法、可被准许的 API 请求。
- 授权：服务器通过客户端的身份认证请求后，需要进一步地进行身份授权鉴别，判断客户端能够访问哪些功能，允许调用哪些 API 服务。
- 访问控制：已经完成身份认证和授权的客户端请求 API 时，服务器还需要进一步验证客户端是否具备 API 的访问权限，避免出现越权问题。
- 日志审计：详细审计并记录 API 接口调用的关键信息，尤其是违反安全策略导致的错误日志相关信息。日志可以对接监控平台，便于通过日志及时发现 API 恶意调用引起的安全问题。同时，日志应支持对恶意行为向上溯源，找到接入点。
- 资产保护：对资产的保护包括两方面，API 自身的安全防护和传输数据信息的防护。
  - API 自身的防护：比如限制 API 的调用频率，限制 API 接口的文件上传大小和数量，设置 API 接口的有效荷载，避免 DDoS 等情况的发生。
  - 传输数据的保护：数据传输过程需要加密。而且，对于 API 的响应信息，需要规范格式，不能依赖客户端做数据过滤，也不能返回超出权限的资源对象属性信息。

2. API 纵深防御原则

“纵深防御”一词来源于军事领域，是指在前线和后方之间构建多层防线，以达到整体防御的目的。

API 安全纵深防御，是指开发人员需要根据 API 的业务属性来设置不同的安全级别，当已经认证过的客户端/用户在调用安全级别更高的 API 时，需要重新进行认证。

5.2.3.2　API 安全测试

在 API 全生命周期中，API 安全测试是一项很重要的工作，主要是通过渗透测试的方式发起对 API 的模拟攻击行为，发现潜在的漏洞和可被利用的风险、不安全的配置，在上线前即可完成对 API 的风险修复工作，减小上线后 API 被黑客攻击、利用的攻击面。