渗透测试步骤

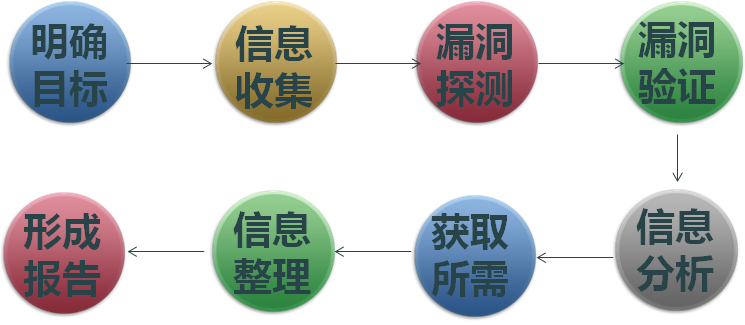
      渗透测试这个事情不是随便拿个工具就可以做了， 要了解业务还需要给出解决方案 。

**渗透测试流程**

**渗透测试与入侵的最大区别**

* 渗透测试：出于保护系统的目的，更全面地找出测试对象的安全隐患。
* 入侵：不择手段地（甚至是具有破坏性的）拿到系统权限。

**一般渗透测试流程**



流程并非万能，只是一个工具。思考与流程并用，结合自己经验。

**2.1 明确目标**

* 确定范围：测试目标的范围，ip，域名，内外网。
* 确定规则：能渗透到什么程度，时间？能否修改上传？能否提权等。
* 确定需求：web应用的漏洞(新上线程序)？业务逻辑漏洞（针对业务的）？人员权限管理漏洞（针对人员、权限）？等等。（立体全方位）

根据需求和自己技术能力来确定能不能做，能做多少。

**2.2 信息收集**

方式：主动扫描，开放搜索等

开放搜索：利用搜索引擎获得，后台，未授权页面，敏感url等。

* 基础信息：IP，网段，域名，端口
* 系统信息：操作系统版本
* 应用信息：各端口的应用，例如web应用，邮件应用等等
* 版本信息：所有这些探测到的东西的版本。
* 服务信息
* 人员信息：域名注册人员信息，web应用中网站发帖人的id，管理员姓名等。
* 防护信息：试着看能否探测到防护设备

**2.3 漏洞探索**

利用上一步中列出的各种系统，应用等使用相应的漏洞。

方法：

* 1.漏扫，awvs，IBM appscan等。
* 2.结合漏洞去exploit-db等位置找利用。
* 3.在网上寻找验证poc。

**内容：**

* 系统漏洞：系统没有及时打补丁
* Websever漏洞：Websever配置问题
* Web应用漏洞：Web应用开发问题
* 其它端口服务漏洞：各种21/8080(st2)/7001/22/3389
* 通信安全：明文传输，token在cookie中传送等。

**2.4 漏洞验证**

将上一步中发现的有可能可以成功利用的全部漏洞都验证一遍。结合实际情况，搭建模拟环境进行试验。成功后再应用于目标中。

* 自动化验证：结合自动化扫描工具提供的结果
* 手工验证，根据公开资源进行验证
* 试验验证：自己搭建模拟环境进行验证
* 登陆猜解：有时可以尝试猜解一下登陆口的账号密码等信息
* 业务漏洞验证：如发现业务漏洞，要进行验证
* 公开资源的利用

-exploit-db/wooyun/

-google hacking

-渗透代码网站

-通用、缺省口令

-厂商的漏洞警告等等。

**2.5 信息分析**

为下一步实施渗透做准备。

* 精准打击：准备好上一步探测到的漏洞的exp，用来精准打击
* 绕过防御机制：是否有防火墙等设备，如何绕过
* 定制攻击路径：最佳工具路径，根据薄弱入口，高内网权限位置，最终目标
* 绕过检测机制：是否有检测机制，流量监控，杀毒软件，恶意代码检测等（免杀）
* 攻击代码：经过试验得来的代码，包括不限于xss代码，sql注入语句等

**2.6 获取所需**

* 实施攻击：根据前几步的结果，进行攻击
* 获取内部信息：基础设施（网络连接，vpn，路由，拓扑等）
* 进一步渗透：内网入侵，敏感目标
* 持续性存在：一般我们对客户做渗透不需要。rookit，后门，添加管理账号，驻扎手法等
* 清理痕迹：清理相关日志（访问，操作），上传文件等

**2.7 信息整理**

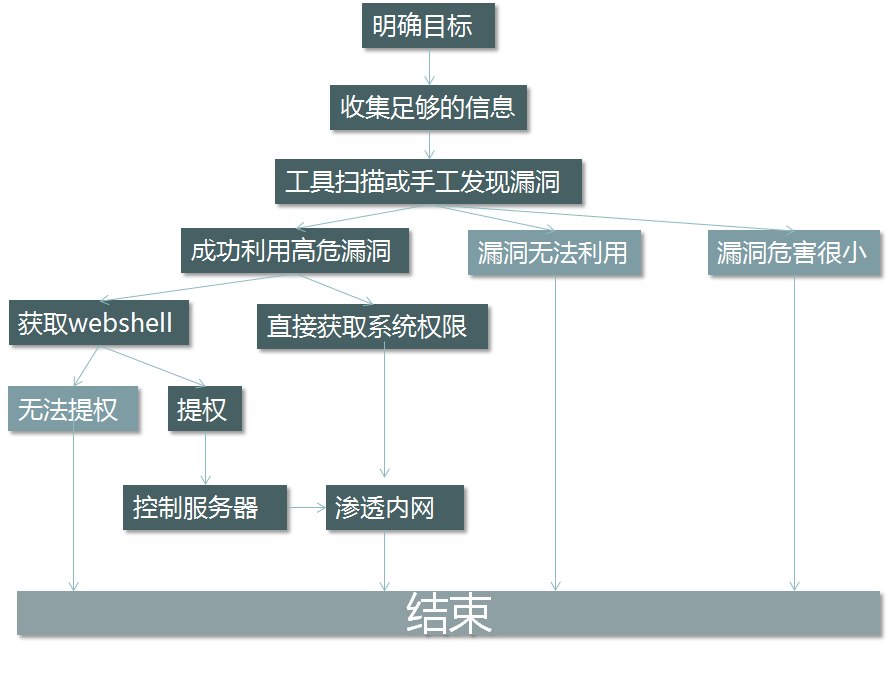
* 整理渗透工具：整理渗透过程中用到的代码，poc，exp等
* 整理收集信息：整理渗透过程中收集到的一切信息
* 整理漏洞信息：整理渗透过程中遇到的各种漏洞，各种脆弱位置信息

目的：为了最后形成报告，形成测试结果使用。

**2.8 形成报告**

* 按需整理：按照之前第一步跟客户确定好的范围，需求来整理资料，并将资料形成报告
* 补充介绍：要对漏洞成因，验证过程和带来危害进行分析
* 修补建议：当然要对所有产生的问题提出合理高效安全的解决办法

**2.9 流程总结**



**渗透测试相关名词解析**

**1.1 一些前置知识（包含但不限于）**

* 脚本（asp、php、jsp）
* html（css、js、html）
* HTTP协议
* CMS（B/S）

**1.2 肉鸡**

被黑客入侵并被长期驻扎的计算机或服务器。可以随意控制，可以是任意系统的设备，对象可以是企业，个人，政府等等所有单位。

**1.3 抓鸡**

利用使用量大的程序的漏洞，使用自动化方式获取肉鸡的行为。

**1.4 Webshell**

通过Web入侵的一种脚本工具，可以据此对网站服务进行一定程度的控制。

**1.5 漏洞**

硬件、软件、协议等等的可利用安全缺陷，可能被攻击者利用，对数据进行篡改，控制等。

**1.6 木马**

通过向服务端提交一句简短的代码，配合本地客户端实现webshell功能的木马。

<%eval request(“pass”)%>

<%execute(request(“pass”))%>

request(“pass”)接收客户端提交的数据，pass为执行命令的参数值。

eval/execute    函数执行客户端命令的内容

**1.7 提权**

操作系统低权限的账户将自己提升为管理员权限使用的方法。

**1.8 后门**

黑客为了对主机进行长期的控制，在机器上种植的一段程序或留下的一个“入口”。

**1.9 跳板**

使用肉鸡IP来实施攻击其他目标，以便更好的隐藏自己的身份信息。

**1.10 旁站入侵**

即同服务器下的网站入侵，入侵之后可以通过提权跨目录等手段拿到目标网站的权限。常见的旁站查询工具有：WebRobot、御剑、明小子和web在线查询等

**1.11 C段入侵**

即同C段下服务器入侵。如目标ip为192.168.180.253 入侵192.168.180.\*的任意一台机器，然后利用一些黑客工具嗅探获取在网络上传输的各种信息。常用的工具有：在windows下有Cain，在UNIX环境下有Sniffit, Snoop, Tcpdump, Dsniff 等。

**1.12 黑盒测试**

在未授权的情况下，模拟黑客的攻击方法和思维方式，来评估计算机网络系统可能存在的安全风险。

黑盒测试不同于黑客入侵，并不等于黑站。黑盒测试考验的是综合的能力（OS、Datebase、Script、code、思路、社工）。思路与经验积累往往决定成败。

**1.13 白盒测试**

相对黑盒测试，白盒测试基本是从内部发起。白盒测试与黑盒测试恰恰相反，测试者可以通过正常渠道向被测单位取得各种资料，包括网络拓扑、员工资料甚至网站或其它程序的代码片断，也能够与单位的其它员工（销售、程序员、管理者……）进行面对面的沟通。

**1.13 黑白盒的另一种说法**

知道源代码和不知道源代码的渗透测试。这时，黑盒测试还是传统的渗透测试，而白盒测试就偏向于代码审计。

**1.14 APT攻击**

Advanced Persistent Threat，高级可持续性攻击，是指组织(特别是政府)或者小团体利用先进的攻击手段对特定目标进行长期持续性网络攻击的攻击形式。

* 1.极强的隐蔽性
* 2.潜伏期长，持续性强
* 3.目标性强