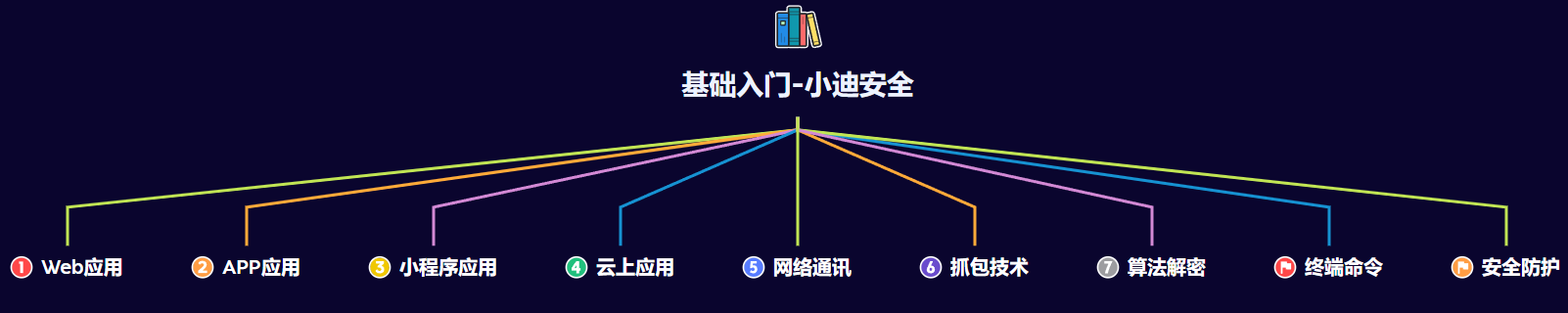
基础入门-Web应用&蜜罐系统&堡垒机运维&API内外接口&第三方拓展架构&部署影响



#知识点：  
1、基础入门-Web应用-域名上的技术要点  
2、基础入门-Web应用-源码上的技术要点  
3、基础入门-Web应用-数据上的技术要点  
4、基础入门-Web应用-解析上的技术要点  
-----------------------------------------------------  
1、基础入门-Web应用-搭建架构上的技术要点  
2、基础入门-Web应用-源码类别上的技术要点  
-----------------------------------------------------  
1、基础入门-Web应用-防护产品-WAF保护  
2、基础入门-Web应用-加速服务-CDN节点  
3、基础入门-Web应用-文件托管-OSS存储  
4、基础入门-Web应用-通讯服务-反向代理  
5、基础入门-Web应用-运维安全-负载均衡  
-----------------------------------------------------  
1、基础入门-Web应用-蜜罐系统  
2、基础入门-Web应用-堡垒机运维  
3、基础入门-Web应用-内外API接口  
4、基础入门-Web应用-第三方拓展架构  
  
#章节点：(待补充)  
Web架构，App架构，小程序架构，前后端分离，容器技术，云产品服务，  
数据加解密算法，数据包抓取，数据包解析，正反向网络通讯，内外网通讯，  
防火墙出入站，Windows&Linux渗透命令，WAF产品，负载均衡，加壳保护等  
  
#具体点：(待补充)  
架构：WEB,APP,小程序,前后端,容器化等  
服务：OSS存储,CDN加速,云数据库,负载均衡等  
网络：不回显,反向代理,防火墙出入站,内外网,正反向连接等  
算法：MD5,Base64,AES,DES,Salt,自定义,代码加密算法等  
命令：Windows&Linux,文件下载,网络查看,反弹权限,用户等  
防护：WAF防护,蜜罐系统,CDN加速,权限设置,加壳加密反调试等

演示案例：

* Web-拓展应用-蜜罐-钓鱼诱使
* Web-拓展应用-堡垒机-突破口
* Web-拓展应用-API接口-突破口
* Web-拓展应用-其他架构-突破口

#课程需明白：  
1、有那几种Web域名上差异  
2、有哪几种源码语言框架差异  
3、网站数据存储有那几个方式  
4、URL访问对应正确和错误原因  
由此我们需要结合安全测试思考：  
1、域名上架构了解后的思路意义  
2、源码上架构了解后的思路意义  
3、数据上架构了解后的思路意义  
4、解析上架构了解后的思路意义  
  
#域名差异-主站&分站&端口站&子站  
1、主站  
www.xiaodi8.com  
2、分站  
blog.xiaodi8.com  
3、端口站  
www.xiaodi8.com:88  
4、目录站  
www.xiaodi8.com/bbs/  
5、子站  
123.blog.xiaodi8.com  
  
#源码差异-结构&语言&框架&闭源&加密  
1、源码目录结构对应  
后台目录，文件目录，逻辑目录，前端目录，数据目录，配置文件等  
  
2、源码开发语言类型  
ASP，ASPX，PHP，Java，Python，Go，Javascript等  
  
3、语言开发框架组件  
PHP：Thinkphp Laravel YII CodeIgniter CakePHP Zend等  
JAVA：Spring MyBatis Hibernate Struts2 Springboot等  
Python：Django Flask Bottle Turbobars Tornado Web2py等  
Javascript：Vue.js Node.js Bootstrap JQuery Angular等  
  
4、开源闭源加密类型  
开源-如Zblog  
闭源-如内部开发  
加密-如通达OA  
  
#数据差异-本地数据&分离数据&云数据库  
1、数据库类型：  
Access、MYSQL、SqlServer、Oracle、  
Redis、DB2、Postgresql、MongoDB等  
2、本地数据库：本地服务器搭建  
3、分离数据库：另外的服务器搭建  
4、云数据库：RDS等  
  
#平台差异-中间件类型&系统类型&容器类型  
1、系统：Windows、Linux、MacOS等  
2、容器：Docker、K8s、Vmware、VirtualBox等  
3、中间件：Apache、Nginx、IIS、lighttpd、Tomcat、Jboos、Weblogic、Websphere、Jetty等  
  
#解析差异-URL路由&绝对相对路径&格式权限  
1、URL路由：URL访问对应文件，MVC模型等  
2、相对绝对：相对当前目录，完整的目录路径  
3、格式权限：后门解析格式，代码正常执行

#Web架构展示：，脚本执行权限等  
1、套用模版型  
csdn / cnblog / github / 建站系统等  
安全测试思路上的不同：  
一般以模版套用，基本模版无漏洞，大部分都采用测试用户管理权限为主  
  
2、前后端分离  
例子：https://www.rxthink.cn/  
思路：https://mp.weixin.qq.com/s/HtLU\_EBXWcbq-lt10oPYwA  
安全测试思路上的不同：前端以JS（Vue,NodeJS等）安全问题，主要以API接口测试，前端漏洞（如XSS）为主，后端隐蔽难度加大。  
  
3、集成软件包  
宝塔，Phpstudy，xamp等  
安全测试思路上的不同：主要是防护体系，权限差异为主  
  
4、自主环境镜像  
云镜像打包，自行一个个搭建  
安全测试思路上的不同：主要是防护体系，权限差异为主  
  
5、容器拉取镜像  
Docker  
安全测试思路上的不同：虚拟化技术，在后期测试要进程逃逸提权  
  
6、纯静态页面  
纯HTML+CSS+JS的设计  
安全测试思路上的不同：无后期讲到的Web漏洞  
找线索：找资产，域名，客户端等  
  
#Web源码形式：  
旨在了解源码差异，后期代码审计和测试中对源码真实性的判断  
1、单纯简易源码  
2、MVC框架源码  
3、编译调用源码  
如：NET-DLL封装 Java-Jar打包  
4、前后端分离源码  
https://segmentfault.com/a/1190000045026063  
5、加密型源码

#Web+WAF  
原理：Web应用防火墙，旨在提供保护  
影响：常规Web安全测试手段会受到拦截  
演示：雷池社区版 https://waf-ce.chaitin.cn/  
Ubuntu20.04 + 雷池社区版  
安装雷池：  
https://docs.waf-ce.chaitin.cn/  
bash -c "$(curl -fsSLk https://waf-ce.chaitin.cn/release/latest/setup.sh)"  
搭建网站：  
https://zhuanlan.zhihu.com/p/705196831  
docker run --name webgoat -d -p 8080:8080 -p 9090:9090 registry.cn-shanghai.aliyuncs.com/kubesec/webgoat:v2023.8  
配置雷池：  
添加站点-设置域名-上游地址-没真实域名修改本地host解析  
  
#Web+CDN  
原理：内容分发服务，旨在提高访问速度  
影响：隐藏真实源IP，导致对目标IP测试错误  
演示：阿里云备案域名全局CDN加速服务  
Windows2016 + BT宝塔面板 + CDN服务  
  
#Web+OSS  
原理：云存储服务，旨在提高访问速度  
演示：https://cloudreve.org/  
Windows2016 + cloudreve + 阿里云OSS  
https://github.com/cloudreve/Cloudreve/releases/tag/3.7.1  
1、启动应用  
2、登录管理  
3、配置存储信息  
4、更改用户组存储属性  
  
阿里云OSS:  
开OSS  
2、新建Bucket  
3、配置Bucket属性  
4、配置Access访问  
  
原理：  
为什么要使用第三方存储？  
1）静态文件会占用大量带宽  
2）加载速度  
3）存储空间  
影响：  
上传的文件或解析的文件均来自于OSS资源，无法解析，单独存储  
1、修复上传安全  
2、文件解析不一样  
3、但存在AK/SK隐患  
  
#Web+反向代理  
正代理为客户端服务,客户端主动建立代理访问目标（不代理不可达）  
反向代理为服务端服务,服务端主动转发数据给可访问地址（不主动不可达）  
原理：通过网络反向代理转发真实服务达到访问目的  
影响：访问目标只是一个代理，非真实应用服务器  
注意：正向代理和反向代理都是解决访问不可达的问题，但由于反向代理中多出一个可以重定向解析的功能操作，导致反代理出的站点指向和真实应用毫无关系！  
演示：Nginx反向代理配置  
Windows2016 + BT宝塔面板 + Nginx  
  
#Web+负载均衡  
原理：分摊到多个操作单元上进行执行，共同完成工作任务  
影响：有多个服务器加载服务，测试过程中存在多个目标情况  
演示：Nginx负载均衡配置  
Windows2016 + BT宝塔面板 + Nginx  
#定义负载设置  
upstream fzjh{  
 server 120.26.70.72:80 weight=1;  
 server 47.75.212.155:80 weight=2;  
}  
  
#定义访问路径 访问策略  
location / {  
 proxy\_pass http://fzjh/;  
}

#蜜罐：https://hfish.net/  
测试系统：Ubuntu 20.04  
一键安装：bash <(curl -sS -L https://hfish.net/webinstall.sh)  
有害影响：用来钓鱼或诱惑测试人员的防护系统  
  
#堡垒机：https://www.jumpserver.org/  
测试系统：Ubuntu 20.04  
一键安装：curl -sSL https://resource.fit2cloud.com/jumpserver/jumpserver/releases/latest/download/quick\_start.sh | bash  
有利影响：Web应用或其他应用提供给测试人员一个能获取到价值信息的系统  
  
#API接口：是一个允许不同软件应用程序之间进行通信和数据交换的接口。API定义了一组规则和协议，软件开发者可以使用这些规则和协议来访问操作系统、库、服务或其他应用程序的功能。  
1、Web API:  
通过HTTP协议进行通信的API，常用于Web服务和应用程序。  
例如，RESTful API、GraphQL API。  
  
2、库和框架API:  
提供特定编程语言或框架功能的API，供开发者在应用程序中使用

。  
例如，Java API、Python标准库。  
  
3、操作系统API:  
提供操作系统功能访问的API。  
例如，Windows API、POSIX API。  
  
4、远程API:  
允许在网络上远程访问服务的API。  
例如，SOAP API、XML-RPC API。  
  
例子：  
内部API：比如我自己开发了一个收银系统，使用API接口可以查询到顾客数据，收入支付，销售提成等  
外部API：比如我自己搭建了一个网站应用，功能需求有要借助到外部的资源，如地图，归属地，短信收发等  
  
有利影响：  
内部API：Web应用提供给测试人员一个能获取到价值信息的接口  
外部API：可以借助提供的API获取到当前网站不想让你获取的信息  
分析API的目录结构、接口命名规则、参数命名规则、功能和业务逻辑等，  
根据这些信息可以进行接口枚举和参数枚举，进而可以进行相关的漏洞测试。  
  
#拓展应用：防火墙 消息队列 分布式等  
ActiveMQ Redis Memcache Jenkins等漏洞  
https://mp.weixin.qq.com/s/SEjxrUgiIIK2bveSBz6mTg  
https://www.yuque.com/u25571586/dyaqbugs?# 密码：xnzx  
有利影响：搭建越多应用即方便了运维也提供给测试人员更多机会

涉及资源：[资源下载地址](https://docs.qq.com/doc/DQ3Z6RkNpaUtMcEFr)