快速通道

文档分类
- 红盟公告
- 新手上路
- 漏洞分析
- 工具介绍
- **安全配置 <=**
- 编程技术
- 安全技术
- 业界动态

本类热门
- WIN2000常用服务的列表
- 某大型企业局域网安全解决方案
- UNIX入侵检测方法
- IPSec打造FreeBSD下信息安全传输

IPSec打造FreeBSD下信息安全传输

提交文章： dahubaobao
提交日期： 2004-06-20
文章属性： 原创
文档类别： 安全配置
相关下载： **=>下载<=**
浏览次数： 今1次/总1420次

文：IPSec打造FreeBSD下信息安全传输
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/IPSec.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！
==========
IPSec在FreeBSD下应用实例
概述
在当今的网络安全中，由于很多服务都是明文通讯，安全性非常底，所以，采用通讯加密是非常有必要的，否则很可能会被嗅探工具简听。IPSec作为加密通讯中的佼佼者，我们没理由不利用。关于IPSec，我想大家或多或少的都有一些了解，黑客防线也刊登过多篇IPSec的相关文章。不过，这些文章都是基于Windows下的，并且都是在讲如何过滤数据包，其实，过滤数据包的任务完全可以交给防火墙来处理，而IPSec真正的强大地方是它的通讯加密。鉴于此，我觉得有必要说说如果使用IPSec进行通讯加密。所以，我通过一段时间的学习，写下了此文，并且以FreeBSD为例，讲解如何使用IPSec与Windows/Unix进行加密通讯，希望能对安全爱好者以及BSD爱好者一些帮助。

先介绍一下我测试的环境：
本地：FreeBSD 4.9 STABLE IP：192.168.0.20
目标：Wdinwos 2000 Advanced Server IP：192.168.0.1
目的：使用IPSec加密所有IP报文，除广播、多播协议外。
注意：这里不介绍如何使用IPSec过滤数据包，因为这些足可以交给IPFW来完成。

一，什么是IPSec
IPSec，即网际安全协议，它通过对IP报文进行加密，并将其封装在IP头中通过企业专用网或者公共网络发送。IPSec在网络层上保护TCP/IP 协议栈中所有 IP 和更高层的协议（例如 TCP、UDP、ICMP），应用程序可以享用经过IPSec处理的网络层的安全服务，而不必关心自身的安全机制。IPSec可在路由器、防火墙、主机上配置，实现端到端的安全通讯。
IPSec工作在网络层，它可以使用隧道模式封装数据包或是直接加密整个数据包。之后就可以将加密后的数据包封装在新的IP包头中，最后传输到目的地。在目标收到数据包后，去除IP头部分，然后对其进行解密（这时已经完成了客户端到服务器密钥交换、认证工作），最后得到原始数据包。如图1就是IPSec的工作流程图。关于IPSec就简单说到这里，如果想深入了解，Windows系统上的IPSec帮助就是很好的资料。

二，更新FreeBSD

FreeBSD默认并不支持IPSec，这是因为默认内核GENERIC中并没有相关选项，所以需要自己添加，并重新编译内核。
#cd /usr/src/sys/i386/conf
#cp GENERIC DAHUKERNEL
#vi DAHUKERNEL
//添加IPSec相关选项，可参见图2
options IPSEC
options IPSEC_ESP
options IPSEC_DEBUG
//保存退出，下面更新系统源代码以及Ports
#mkdir /root/ports_src_update
#cd /usr/share/examples/cvsup
#cp ports-supfile stable-supfile /root/ports_src_update/
#cd /;cd /root/ports_src_update
#vi ports-supfile
//修改“ default host= ”为你想使用的服务器，比如我是：cvsup.freebsdchina.org
#vi stable-supfile
//同样是修改“ default host= ”为你想使用的服务器
#cvsup －g －L 2 ports-supfile
//更新Ports，大概10-20分钟之间，视网络速度而定
#cvsup －g －L 2 stable-supfile
//更新系统源代码，同步到Stable。大概10-20分钟之间，视网络速度而定。在源代码更新完毕之后，就需要用新代码替换系统原有代码以及合并/etc下的配置文件，并且用新内核替换掉旧内核。当然，这步不是必须的，如果你不久之前更新过源代码（或者已经同步到Stable），就可以略过。但，你还是必须更新Ports和重新编译内核。
#cd /;cd /usr/src
#make buildworld //同步系统，这需要很长的时间，最少也要1个小时，也可以运行多个进程加快该过程：make －j4 buildworld
#make buildkernel GENERIC=DAHUKERNEL
#make installkernel GENERIC=DAHUKERNEL
//编译内核，并用新内核（DAHUKERNEL）替换掉旧内核（GENERIC）。编译完成之后重新启动即可。提示：如果你以前编译过内核，应该先删除/usr/obj下的内核组件文件，这样即可以加快make的速度，又可以避免系统在遇到不能覆盖的文件时发生冲突。/usr/obj目录下的文件比较特殊，它使用了schg（system immutable）标志，这是一个安全措施，为防止意外操作提供的一个保护方法，即使是root用户，也不能删除该文件。如果需要绕过这个限制，可以使用如下方法：
#cd /usr/obj
#chflags －R noschg *
#rm －rf *
如果这个时候系统的安全级别为1或者更高（根据rc.conf文件中的定义），那么将无法使用上边的命令删除schg标志。需要重新启动系统到单用户模式下方能完成该操作。

#cp －Rp /etc /etc.old
#cd /etc
#mergemaster －cv
//同步配置文件，-v表示限制详细过程，-c表示将配置文件进行比较（diff）而不是合并，如果打算合并配置文件，可以使用-i参数。最后使用uname －a查看结果。

现在，你的系统已经是最新的了，如果你是新安装的系统，请务必按照上边的方法更新系统，这样即可以补上一些安全漏洞，又可以为系统带来更好的性能，和乐而不为呢？！

三，应用IPSec
1，FreeBSD
经过上边的设置，系统已经是最新的，但现在还不能使用IPSec，还需要安装一个为IKE (ISAKMP/Oakley) Key Management Daemon的软件，好在FreeBSD已经将其收录在Ports中，这样就不用在下载了。
#cd /usr/ports/security/racoon
#make install clean
完成之后，我们需要对其进行相应的配置。它被安装到了/usr/local/etc/racoon/目录中，我们需要对psk.txt和racoon.conf进行编辑，以符合我们的需要。提示：在FreeBSD的世界里，凡是安装的软件（不管是通过Package还是Ports），都将安装到/usr/local目录下，配置文件放到/usr/local/etc下，库文件放到/usr/local/lib、/usr/local/libexec、/usr/local/libdata目录中，二进制文件放到/usr/local/bin、/usr/local/sbin目录中。这样做的好处是显而易见的，因为可以在不修改系统的情况下，将/usr/local目录下的子目录和所有文件直接复制到另一个新系统，而安装过的软件大部分都可以运行。
#cd ../../../local/etc/racoon
#cp psk.txt.dist psk.txt
#chmod 0600 psk.txt
//psk.txt权限必须修改为0600，并且为root所有，否则racoon无法运行
#vi psk.txt

//见图3，删除所有内容，只添加：
目标IP 预共享密钥
192.168.0.1 dahubaobao
#cp racoon.conf.dist racoon.conf
#chmod 0600 racoon.conf
//两次chmod后，权限和文件所属应如图4所示。
#vi racoon..conf
//编辑配置文件，注释掉一些不需要的内容。这里需要分开来说，一个是使用CA认证，一个是不使用。如果不使用CA认证，我们就需要把关于CA认证的部分注释掉；如果需要使用，就使用OpenSSL创建CA证书，并在配置文件中填写相关信息（后面会说到）。具体如下。
path certificate "/usr/local/etc/cert" ;
//如果需要CA认证，这里不做任何改动；如果不需要，注释掉就可以了。这里就是放CA证书的目录。默认没有注释。

log debug; //去掉注释，如果你需要日志的话。

isakmp 192.168.0.20 [500];
//去掉注释，添加本地IP。

#my_identifier address;
#my_identifier user_fqdn "sakane@kame.net";
#peers_identifier user_fqdn "sakane@kame.net";
#certificate_type x509 "mycert" "mypriv";
//这里是关于CA认证的地方，如果使用了CA认证，就需要根据使用OpenSSL创建CA证书时所填写的信息来修改这里；如果不使用，注释掉即可。

#my_identifier user_fqdn "sakane@kame.net";
#peers_identifier user_fqdn "sakane@kame.net";
#certificate_type x509 "mycert" "mypriv";
//同上

下面来看看如何使用OpenSSL来创建证书。
#openssl genrsa -des3 -out ca.key 1024
#openssl req -new -x509 -days 365 -key ca.key -out ca.crt

#openssl genrsa -out win2000.key 1024
#openssl req -new -key win2000.key -out win2000.csr
#openssl x509 -req -days 166 -in win2000.csr -CA ca.crt -CAkey ca.key -CAcreateserial －out
win2000.crt
#openssl pkcs12 -export -inkey win2000.key -certfile ca.crt -in win2000.crt -out win2000.p12
//以上是为Windows 2000（目标）创建的证书，由于Windows 2000证书需要Pkcs12格式，所以使用上边这条命令进行转换。下面来创建FreeBSD（本地）系统上的证书。
#openssl genrsa -out freebsd.key 1024
#openssl req -new -key freebsd.key -out freebsd.csr
#openssl x509 -req -days 166 -in freebsd.csr -CA ca.crt -CAkey ca.key -CAcreateserial －out
freebsd.crt
//这样就创建了需要的证书文件，所有的相关文件都会在当前目录下。创建证书的过程可以参见图5、6、7。

现在把win2000.p12复制到Windows 2000，然后运行mmc，“ 控制台---添加删除管理单元---添加---证书---添加---服务账户---下一步---本地计算机---下一步---IPSEC Policy Agent---完成 ”。最后如图8将证书导入即可。
#mkdir /usr/local/etc/racoon/cert
#cp * /usr/local/etc/racoon/cert/
#chmod 0600 /usr/local/etc/racoon/cert/*
//把证书文件复制到/usr/local/etc/racoon/cert目录中，然后修改配置文件，如下。
my_identifier address;
my_identifier user_fqdn "freebsd@ringz.com";
peers_identifier user_fqdn "win2000@ringz.comt";
certificate_type x509 "freebsd.crt" "freebsd.key";
peers_certfile "win2000.crt";
//在下面还有一处需要修改，跟这里完全一样，就不在多说了。

现在就可以运行racoon，并指定一个放日志的地方
#racoon -f /usr/local/etc/racoon/racoon.conf -l /var/log/racoon.log
然后就可以加入安全策略了。加入安全策略需要使用Setkey命令，它是用来添加，更新，删除，显示内核中的安全策略数据库。Setkey可以执行标准输入操作和从文件的输入操作。图9是它的语法，下面解释一下参数的含义：

-c //执行标准输入操作
-f filename //执行从文件的输入操作
-D　//打印出SAD。如果有选项-P，打印出SPD。
-F　//清除SAD。如果有选项-P，则清除SPD。
-a　//显示死了的SAD。
-d　//允许打印调试信息。通常不会使用。
-x　//循环显示发送到Pf_key接口的所有消息
-h　//加入16进制显示
Setkey还有很多属性，具体可以通过man setkey查看，这里只需用到Spdadd属性，使用它可以很简单的添加安全策略。
#setkey -c <<EOF
spdadd 192.168.0.20 192.168.0.1 any -P out ipsec
esp/transport/192.168.0.20-192.168.0.1/require;
spdadd 192.168.0.1 192.168.0.20 any -P in ipsec
esp/transport/192.168.0.1-192.168.0.20/require;
EOF
这里使用的是传输模式，如果打算使用隧道模式，只需把上边的安全策略中的Transport修改为Tunnel即可。注意，Transport和Tunnel是有区别的，同样，目标主机也需要适当的修改以符合FreeBSD上的安全策略。也就是说，目标主机是使用传输模式还是隧道模式。（本文中目标主机是Windows 2000，稍后会做详细解释。）

2，Windows 2000
FreeBSD系统上的相关设置终于完成了。现在就可以使用IP安全策略添加安全规则。首先，点击开始---程序---管理工具---本地安全策略；然后右键单击IP安全策略，在本地机器，选择创建IP安全策略；弹出IP安全策略向导---下一步，在名称中填上加密所有IP通讯，描述随意填写；下一步，将激活默认响应规则前边的对勾去掉---下一步；这里就需要注意一下，如果使用CA认证，则选择“ 使用由此证书颁发机构（CA）颁发的证书 ”，然后点浏览选择之前创建的证书（win2000.p12）；如果使用预共享密钥，就选择“ 此字串用来保护密钥交换 ”，然后填写预共享密钥（psk.txt）---下一步(如图10)---完成。

好，继续来添加IP安全规则。点击添加---下一步，来到隧道终结点对话框，如果在FreeBSD系统上的安全策略是传输模式，则选择“ 此规则不指定隧道 ”；如果是隧道模式，就选择“ 隧道终结点由此IP地址指定 ”，填写上FreeBSD的IP地址---下一步---局域网（LAN）---下一步；身份验证方法选择预共享密钥，填写dahubaobao---下一步；IP筛选器列表选择“ 所有IP通讯 ”---下一步；在筛选器操作中选择要求安全设置。当然，你也可以自己创建一个筛选器操作，方法是点击添加---下一步，填写名称，下一步，选择协商安全---下一步，选择回到不安全的通讯---下一步，来到IP通信安全设施，推荐默认设置，如果对默认设置不满意，也可以选择自定义（如图11），下一步---完成。选中刚才添加的筛选器操作---下一步---完成。确定---关闭，最后指派一下就可以了。

到此，两个系统之间的IP通讯都使用IPSec进行加密了，安全性得到了绝对的保证。如果你的计算机是两台Unix，添加规则的时候注意一下IP地址的顺序就可以了，过程比Windows简单很多，再此就不多说废话了。

总结
IPSec的功能还不只如此，本文只是介绍了它的部分内容。IPSec的最大特点是系统自带的，不关是Unix/Linux还是Windows，都支持IPSec，我们确实应该好好利用。本文的实践性很强，大家回去一定要动手试试，否则一切都是纸上谈兵！鉴于本人的水平有限，文章难免有不足之处，还请多多请教。如有任何问题，欢迎来我那人气可怜的网站（www.ringz.org）与我交流。

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