实验三 入侵检测系统实验

1. 实验目的与实验要求

1. 通过实验掌握轻量级入侵检测系统Snort的安装。

2. 掌握Snort的配置与使用方法，理解基于误用检测的入侵检测系统工作原理。

1. 实验原理

**1. 基于特征的入侵检测技术**

从技术上，入侵检测也可分为两类：一种基于特征（误用检测）（signature-based），另一种基于异常情况(anomaly-based)。

对于基于标识的检测技术来说，首先要定义违背安全策略的事件的特征，如网络数据包的某些头信息。检测主要判别这类特征是否在所收集到的数据中出现。此方法非常类似杀毒软件。

而基于异常的检测技术则是先定义一组系统“正常”情况的数值，如CPU利用率、内存利用率、文件校验和等（这类数据可以人为定义，也可以通过观察系统、并用统计的办法得出），然后将系统运行时的数值与所定义的“正常”情况比较，得出是否有被攻击的迹象。这种检测方式的核心在于如何定义所谓的“正常”情况。

SNORT是一个强大的轻量级的网络入侵检测系统,它具有实时数据流量分析和日志IP网络数据包的能力，能够进行协议分析，对内容搜索或者匹配。它是一个基于特征检测的入侵检测系统。

**2.** **Snort简介**

snort有三种工作模式：嗅探器、数据包记录器、网络入侵检测系统。嗅探器模式仅仅是从网络上读取数据包并作为连续不断的流显示在终端上。数据包记录器模式把数据包记录到硬盘上。网路入侵检测模式是最复杂的，而且是可配置的。我们可以让snort分析网络数据流以匹配用户定义的一些规则，并根据检测结果采取一定的动作。

**嗅探器**

所谓的嗅探器模式就是snort从网络上读出数据包然后显示在你的控制台上。首先，我们从最基本的用法入手。如果你只要把TCP/IP包头信息打印在屏幕上，只需要输入下面的命令：

　　./snort -v

使用这个命令将使snort只输出IP和TCP/UDP/ICMP的包头信息。如果你要看到应用层的数据，可以使用：

　　./snort -vd

这条命令使snort在输出包头信息的同时显示包的数据信息。如果你还要显示数据链路层的信息，就使用下面的命令：

　　./snort -vde

注意这些选项开关还可以分开写或者任意结合在一块。例如：下面的命令就和上面最后的一条命令等价：

　　./snort -d -v –e

**数据包记录器**

如果要把所有的包记录到硬盘上，你需要指定一个日志目录，snort就会自动记录数据包：

　　./snort -dev -l ./log

当然，./log目录必须存在，否则snort就会报告错误信息并退出。当snort在这种模式下运行，它会记录所有看到的包将其放到一个目录中，这个目录以数据包目的主机的IP地址命名，例如：192.168.10.1

　　如果你只指定了-l命令开关，而没有设置目录名，snort有时会使用远程主机的IP地址作为目录，有时会使用本地主机IP地址作为目录名。为了只对本地网络进行日志，你需要给出本地网络：

　　./snort -dev -l ./log -h 192.168.1.0/24

这个命令告诉snort把进入C类网络192.168.1的所有包的数据链路、TCP/IP以及应用层的数据记录到目录./log中。

如果你的网络速度很快，或者你想使日志更加紧凑以便以后的分析，那么应该使用二进制的日志文件格式。所谓的二进制制日志文件格式就是tcpdump程序使用的格式。使用下面的命令可以把所有的包记录到一个单一的二进文件中：

　　./snort -l ./log -b

注意此处的命令行和上面的有很大的不同。我们勿需指定本地网络，因为所有的东西都被记录到一个单一的文件。你也不必冗余模式或者使用-d、-e功能选项，因为数据包中的所有内容都会被记录到日志文件中。

你可以使用任何支持tcpdump二进制格式的嗅探器程序从这个文件中读出数据包，例如：tcpdump或者Ethereal。使用-r功能开关，也能使snort读出包的数据。snort在所有运行模式下都能够处理tcpdump格式的文件。例如：如果你想在嗅探器模式下把一个tcpdump格式的二进制文件中的包打印到屏幕上，可以输入下面的命令：

　　./snort -dv -r packet.log

在日志包和入侵检测模式下，通过BPF(BSD Packet Filter)接口，你可以使用许多方式维护日志文件中的数据。例如，你只想从日志文件中提取ICMP包，只需要输入下面的命令行：

./snort -dvr packet.log icmp

**网络入侵检测系统**

snort最重要的用途还是作为网络入侵检测系统(NIDS)，使用下面命令行可以启动这种模式：

　　./snort -dev -l ./log -h 192.168.1.0/24 -c snort.conf

snort.conf是规则集文件。snort会对每个包和规则集进行匹配，发现这样的包就采取相应的行动。如果你不指定输出目录，snort就输出到/var/log/snort目录。

注意：如果你想长期使用snort作为自己的入侵检测系统，最好不要使用-v选项。因为使用这个选项，使snort向屏幕上输出一些信息，会大大降低snort的处理速度，从而在向显示器输出的过程中丢弃一些包。

此外，在绝大多数情况下，也没有必要记录数据链路层的包头，所以-e选项也可以不用：

　　./snort -d -h 192.168.1.0/24 -l ./log -c snort.conf

这是使用snort作为网络入侵检测系统最基本的形式，日志符合规则的包，以ASCII形式保存在有层次的目录结构中。

网络入侵检测模式下的输出选项

在NIDS模式下，有很多的方式来配置snort的输出。在默认情况下，snort以ASCII格式记录日志，使用full报警机制。如果使用full报警机制，snort会在包头之后打印报警消息。如果你不需

要日志包，可以使用-N选项。

snort有6种报警机制：full、fast、socket、syslog、smb(winpopup)和none。其中有4个可以在命令行状态下使用-A选项设置。这4个是：

-A fast：报警信息包括：一个时间戳(timestamp)、报警消息、源/目的IP地址和端口。

-A full：是默认的报警模式。

-A unsock：把报警发送到一个UNIX套接字，需要有一个程序进行监听，这样可以实现实时报警。

-A none：关闭报警机制。

使用-s选项可以使snort把报警消息发送到syslog，默认的设备是LOG\_AUTHPRIV和LOG\_ALERT。可以修改snort.conf文件修改其配置。

snort还可以使用SMB报警机制，通过SAMBA把报警消息发送到Windows主机。为了使用这个报警机制，在运行./configure脚本时，必须使用--enable-smbalerts选项。

下面是一些输出配置的例子：

使用默认的日志方式(以解码的ASCII格式)并且把报警发给syslog：

./snort -c snort.conf -l ./log -s -h 192.168.1.0/24

使用二进制日志格式和SMB报警机制：

./snort -c snort.conf -b -M WORKSTATIONS

**3.** **编写snort 规则**

3.1基础

snort使用一种简单的，轻量级的规则描述语言，这种语言灵活而强大。在开发snort规则时要记住几个简单的原则。

第一，大多数snort规则都写在一个单行上，或者在多行之间的行尾用/分隔。Snort规则被分成两个逻辑部分：规则头和规则选项。规则头包含规则的动作，协议，源和目标ip地址与网络掩码，以及源和目标端口信息；规则选项部分包含报警消息内容和要检查的包的具体部分。

下面是一个规则范例：

alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg: "mountd access";)

第一个括号前的部分是规则头（rule header），包含的括号内的部分是规则选项（rule options）。规则选项部分中冒号前的单词称为选项关键字（option keywords）。注意，不是所有规则都必须包含规则选项部分，选项部分只是为了使对要收集或报警，或丢弃的包的定义更加严格。组成一个规则的所有元素对于指定的要采取的行动都必须是真的。当多个元素放在一起时，可以认为它们组成了一个逻辑与（AND）语句。同时，snort规则库文件中的不同规则可以认为组成了一个大的逻辑或（OR）语句。

3.2规则高级概念

**Includes:**

include允许由命令行指定的规则文件包含其他的规则文件。

格式：

include:

注意在该行结尾处没有分号。被包含的文件会把任何预先定义的变量值替换为自己的变量引用。参见变量（Variables）一节以获取关于在SNORT规则文件中定义和使用变量的更多信息。

**Variables :**

变量可能在snort中定义。

格式：

var:

例子：

var MY\_NET 192.168.1.0/24

alert tcp any any -> $MY\_NET any (flags: S; msg: "SYN packet";)

规则变量名可以用多种方法修改。可以在"$"操作符之后定义变量。"?" 和 "-"可用于变量修改操作符。

$var - 定义变量。

$(var) - 用变量"var"的值替换。

$(var:-default) - 用变量"var"的值替换，如果"var"没有定义用"default"替换。

$(var:?message) - 用变量"var"的值替换或打印出错误消息"message"然后出。

例子：

var MY\_NET $(MY\_NET:-192.168.1.0/24)

log tcp any any -> $(MY\_NET:?MY\_NET is undefined!) 23

**规则头**

规则动作：

规则的头包含了定义一个包的who，where和what信息，以及当满足规则定义的所有属性的包出现时要采取的行动。规则的第一项是"规则动作"（rule action），"规则动作"告诉snort在发现匹配规则的包时要干什么。在snort中有五种动作：alert、log、pass、activate和dynamic.

1、Alert-使用选择的报警方法生成一个警报，然后记录（log）这个包。

2、Log-记录这个包。

3、Pass-丢弃（忽略）这个包。

4、activate-报警并且激活另一条dynamic规则。

5、dynamic-保持空闲直到被一条activate规则激活，被激活后就作为一条log规则执行。

你可以定义你自己的规则类型并且附加一条或者更多的输出模块给它，然后你就可以使用这些规则类型作为snort规则的一个动作。

下面这个例子创建一条规则，记录到tcpdump。

ruletype suspicious

{

type log output

log\_tcpdump: suspicious.log

}

下面这个例子创建一条规则，记录到系统日志和MySQL数据库

ruletype redalert

{

type alert output

alert\_syslog: LOG\_AUTH LOG\_ALERT

output database: log, mysql, user=snort dbname=snort host=localhost

}

**协议**

规则的下一部分是协议。Snort当前分析可疑包的ip协议有四种：tcp 、udp、icmp和ip。将来可能会更多，例如ARP、IGRP、GRE、OSPF、RIP、IPX等。

Ip地址

规则头的下一个部分处理一个给定规则的ip地址和端口号信息。关键字"any"可以被用来定义任何地址。Snort没有提供根据ip地址查询域名的机制。地址就是由直接的数字型ip地址和一个cidr块组成的。Cidr块指示作用在规则地址和需要检查的进入的任何包的网络掩码。/24表示c类网络，/16表示b类网络，/32表示一个特定的机器的地址。例如，192.168.1.0/24代表从192.168.1.1到192.168.1.255的地址块。在这个地址范围的任何地址都匹配使用这个192.168.1.0/24标志的规则。这种记法给我们提供了一个很好的方法来表示一个很大的地址空间。

有一个操作符可以应用在ip地址上，它是否定运算符（negation operator）。这个操作符告诉snort匹配除了列出的ip地址以外的所有ip地址。否定操作符用"！"表示。下面这条规则对任何来自本地网络以外的流都进行报警。

alert tcp !192.168.1.0/24 any -> 192.168.1.0/24 111 (content: "|00 01 86 a5|"; msg: "external mountd access";)

这个规则的ip地址代表"任何源ip地址不是来自内部网络而目标地址是内部网络的tcp包"。

也可以指定ip地址列表，一个ip地址列表由逗号分割的ip地址和CIDR块组成，并且要放在方括号内“[”，“]”。此时，ip列表可以不包含空格在ip地址之间。下面是一个包含ip地址列表的规则的例子。

alert tcp ![192.168.1.0/24,10.1.1.0/24] any -> [192.168.1.0/24,10.1.1.0/24] 111 (content: "|00 01 86 a5|"; msg: "external mountd access";)

端口号

端口号可以用几种方法表示，包括"any"端口、静态端口定义、范围、以及通过否定操作符。"any"端口是一个通配符，表示任何端口。静态端口定义表示一个单个端口号，例如111表示portmapper，23表示telnet，80表示http等等。端口范围用范围操作符"："表示。范围操作符可以有数种使用方法，如下所示：

log udp any any -> 192.168.1.0/24 1:1024

记录来自任何端口的，目标端口范围在1到1024的udp流

log tcp any any -> 192.168.1.0/24 :6000

记录来自任何端口，目标端口小于等于6000的tcp流

log tcp any :1024 -> 192.168.1.0/24 500:

记录来自任何小于等于1024的特权端口，目标端口大于等于500的tcp流

端口否定操作符用"！"表示。它可以用于任何规则类型（除了any，这表示没有，呵呵）。例如，由于某个古怪的原因你需要记录除x windows端口以外的所有一切，你可以使用类似下面的规则：

log tcp any any -> 192.168.1.0/24 !6000:6010

**方向操作符**

方向操作符"->"表示规则所施加的流的方向。方向操作符左边的ip地址和端口号被认为是流来自的源主机，方向操作符右边的ip地址和端口信息是目标主机，还有一个双向操作符"<>"。它告诉snort把地址/端口号对既作为源，又作为目标来考虑。这对于记录/分析双向对话很方便，例如telnet或者pop3会话。用来记录一个telnet会话的两侧的流的范例如下：

log !192.168.1.0/24 any <> 192.168.1.0/24 23

Activate 和 dynamic 规则：

注：Activate 和 dynamic 规则将被tagging 所代替。在snort的将来版本，Activate 和 dynamic 规则将完全被功能增强的tagging所代替。

Activate 和 dynamic 规则对给了snort更强大的能力。你现在可以用一条规则来激活另一条规则，当这条规则适用于一些数据包时。在一些情况下这是非常有用的，例如你想设置一条规则：当一条规则结束后来完成记录。Activate规则除了包含一个选择域：activates外就和一条alert规则一样。Dynamic规则除了包含一个不同的选择域：activated\_by 外就和log规则一样，dynamic规则还包含一个count域。

Actevate规则除了类似一条alert规则外，当一个特定的网络事件发生时还能告诉snort加载一条规则。Dynamic规则和log规则类似，但它是当一个activate规则发生后被动态加载的。把他们放在一起如下图所示：

activate tcp !$HOME\_NET any -> $HOME\_NET 143 (flags: PA; content: "|E8C0FFFFFF|/bin"; activates: 1; msg: "IMAP buffer overflow!";)

dynamic tcp !$HOME\_NET any -> $HOME\_NET 143 (activated\_by: 1; count: 50;)

1. 实验环境

安装Windows操作系统并连接网络的一台PC机。

1. 实验内容与步骤

**任务一：在Windows 环境下安装和使用Snort**

* + 1. 首先从<http://www.winpcap.org/install/default.htm>或实验室ftp下载winpcap并安装，这一步基本可以忽略，实验室的电脑上一般都安装了winpcap
    2. 从<http://www.snort.org> 或用老师转发的snort并安装
    3. 从<http://www.snort.org> 或用老师转发的snort规则，解压后，将规则文件（.rules）复制到Snort安装目录的rules/目录下
    4. 从“命令提示符”进入Snort安装目录，找到\bin目录并运行snort.exe。例如：

C:\snort\bin>snort

具体使用方法，可以参考“Snort使用手册”或者从互联网上寻找更多资料。

注意：使用-i选项，以选择正确的网卡。使用-l选项，选择正确的日志记录目录。

* + 1. 修改并使用Snort的默认配置文件(snort.conf)运行Snort，注意Snort的语法。配置文件可以参考老师给出的示例。使用时，将其复制到Snort的/etc/目录，使用命令：

C:\snort\bin>snort –i 2 –c ../etc/snort.conf –l ../log/

注：-i 2 表示系统里面的第2块网卡，根据自己的电脑网卡情况决定，需要使用 snort –W显示网卡接口。不要复制上面命令，手动输入，字符存在差异。

如需停止snort，使用“ctrl+c”

**任务二：验证Snort的检测性能**

为了验证Snort的工作状态与检测性能，使用另外一台电脑发起一次端口扫描攻击。

1. 下载扫描工具（如nmap等），对安装Snort的电脑进行一次端口扫描。
2. 查看Snort的检测结果，并记录在实验报告中。如果没有检测到端口扫描，对Snort的配置及规则库进行检查与调整，使其能检测到端口扫描攻击。

**任务三：查看Snort的代码结构**

为了能理解Snort的工作原理，需要对Snort的代码结构进行分析。可以打开snort.conf文件，了解其插件机制，并可以打开snort目录找到相应的源代码。

1. 提交的实验报告

完成实验内容，在实验报告里面保存截图记录，并思考相关问题，统一由各班学习委员收齐实验报告（每位同学的实验报告文件名：学号-姓名-实验3.docx）后压缩打包通过QQ发给吴礼发老师。