实操实验介绍

项目 2：第 2 部分
网络入侵检测系统 *(Snort)*

目录

类别：

CS-CNS:计算机网络安全

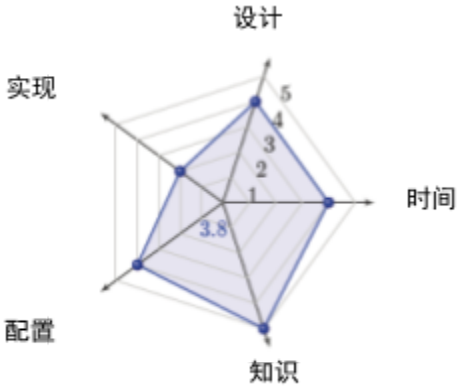
目标：

1. 设置网络入侵检测系统 (snort)，以检测恶意网络流量
2. 实施各种网络攻击，用以进行安全测试

预计时长：

1. 行家：180 分钟
2. 新手：2 天

难度示意图：



衡量指标	数值 (0-5)
时间	4
设计	4
实现	2
配置	4
知识	5
分数（平均）	3.8

难度表格

所需的操作系统：

Linux：Ubuntu 18.04 LTS (Bionic Beaver)

实验室运行环境：

3. 网关：Linux (Ubuntu 18.04 LTS)

4. 网络设置：
通过 Net 3 连接互联网：172.16.0.0/12
客户端 Net 1：192.168.0.0/24
服务器端 Net 2：10.0.0.0/8

实验室的准备工作：

1. 了解如何使用 Linux 操作系统

2. 有关计算机网络的基础知识（项目 1）

3. 了解如何使用 hping3 来部署网络测试和攻击（将提供额外资料）

实验室概述

在本实验室中，学生们将探索 Snort 入侵检测系统。学生们将学习 Snort IDS，这是一种基于特征的入侵检测系统，可用于检测网络攻击。Snort 还可用作简单的数据包记录器。对于这个实验室，学生们将把 snort 用作数据包嗅探器，并编写自己的 IDS 规则。

道德声明：这个实验室会为学生提供进入公共域的途径。此实验室中部署的攻击可能会对公共域造成破坏。因此，我们要求每个学生在进行本实验时同意以下要求：

1. 同意不故意部署针对任何公共域服务或计算机的网络攻击。
2. 删除网关节点上的默认网关，以阻止所有流向公共域的流量。为此，你可以在网关节点上发出以下命令：
   $ route -n % 识别默认网关 IP 地址，如 172.16.0.1
   $ sudo route del default gw 172.16.0.1 % 删除通往公共域的默认网关
   通过删除默认网关，你可以防止对外部公共服务/节点造成意外攻击。当你想从公共域下载必要的软件包时，可以恢复默认网关：
   $ sudo route add default gw 172.16.0.1 -i ens3 % 假设你的默认网关 IP 地址为172.16.0.1，该地址会分配给端口 ens3。
3. 部署的攻击是资源密集型的。因此，确保在完成测试后停止攻击。请不要离开或注销你的系统，并让已部署的 DoS 攻击继续运行。这将极大地减慢虚拟实验室的响应时间。

1. 任务 1 Snort 的基础知识和准备工作

Snort 是一个开源的网络入侵检测和预防系统。该系统可以分析网络中的实时流量和数据流，也可以检查协议，并检测不同类型的攻击。在 NIDS 中，snort 基本上是根据用户编写的规则检查数据包。Snort 规则可以用任何语言编写；规则不仅结构很好、易于阅读，也可以进行修改。在缓冲区溢出攻击中，snort 可以通过匹配之前的攻击模式来检测攻击，再采取适当的措施来防止攻击。在基于特征的 IDS 系统中，如果模式匹配，可以很容易地发现攻击，但当新的攻击出现时，系统会失败，但 snort 可以通过分析实时流量来克服这一限制。每当有数据包进入网络时，snort 都会检查网络的行为。如果网络性能下降，snort 就会停止数据包的处理，丢弃数据包，并将其详细信息存储在特征数据库中。

1.1 Snort 的组件

概括而言，Snort 是多个组件的组合。各个组件协同工作，以找到特定的攻击，再采取该攻击所需的对应操作。Snort 基本上由以下主要部件组成，如图 CS-CNS-00003.1 所示：

数据包来自互联网，会进入到数据包解码器并历经几个阶段；snort 会在每个阶段采取所需的操作，比如：如果检测引擎在数据包中发现任何杂项内容，则会丢弃该数据包，并在到达输出模块的途中登录数据包，或者生成警报。snort 组件的主要功能描述如下：

- 数据包解码器：负责形成供其他组件使用的数据包。其任务是确定数据包中使用了哪些底层协议，并确定位置和

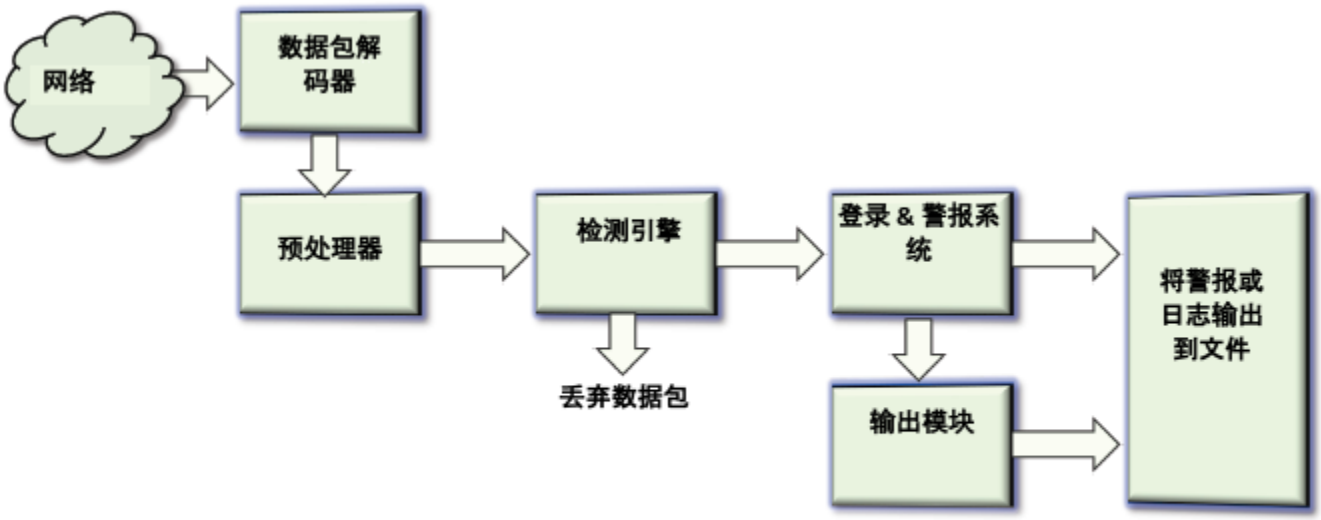


图 **CS-CNS-00003.1**
Snort 的组件

数据包数据的大小，这会在后面的组件中用到。需要注意的是，解码器也会查找报头中的异常（例如无效大小），这可能会导致解码器生成警报。

- 预处理器：用于验证针对 HTTP 或 FTP 等服务的异常。最终目的是尽量使其更难欺骗检测引擎。Snort 会使组件发挥插件的作用，这些插件预处理器可以整理或修改数据包数据。因此，每个服务都可以有相应的预处理器。使用预处理器的例子有：对 URI 进行解码，以处理碎片整理数据包，其中数据包的碎片可用于欺骗检测引擎、检测端口扫描以及检测 ARP 数据包中的异常，如 ARP 欺骗等。
- 检测引擎：负责“检测数据包中是否存在某种入侵活动”。通过将包含这些规则的配置文件中指定的规则集链接在一起，并将其应用于每个数据包来完成检测。如果数据包与某个规则匹配，则执行该规则的指定操作，或丢弃该数据包。如果 snort 根据网络负载实时执行此操作，可能会出现延迟，最坏的情况下，会导致数据包一起被丢弃。
- 日志记录和警报系统：如果数据包与某个规则匹配，则日志记录和警报系统将记录日志和/或生成警报。当然，该组件生成的消息和内容可通过配置文件进行配置。如果一个数据包触发多个规则，最高的警报级别就是该组件实际生成的警报级别。
- 输出模块：负责控制生成的输出类型，使用插件系统为用户提供灵活性，并且高度可配置。这可能包括简单地记录或登录到数据库，发送 SNMP 陷阱，生成 XML 报告，亦或是通过 UNIX 套接字发送警报，（例如）使其可以动态修改网络配置（防火墙或路由器）。

1.2 实验室环境的准备工作

注意：仅在用于网络入侵检测实践的情况下，才可运行本实验室。

第一步是检查 snort 是否正确安装在正在运行的虚拟机上。

```
$ snort -- v
```

如已安装 snort，可能会显示以下输出：

```
root@ubuntu:~# snort -- v

,,-        -*> Snort! <*-
o" )~      版本 2.9.7.0 GRE (Build 149)
''''       Martin Roesch 和 Snort Team 合著：http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco 及其附属公司。版权所有。
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           libpcap 版本：1.8.1
           PCRE 版本：8.39 2016-06-14
           ZLIB 版本：1.2.11
```

要检查 snort 运行状态，你可以运行以下命令：

```
service snort status
```

如果运行正常，可能会显示以下内容：

```
*    snort.service - LSB:Lightweight network intrusion detection system

Loaded: loaded (/etc/init.d/snort; generated)
Active: active (running) since Fri 2020-10-02 16:25:11 UTC; 1min 15s ago
Docs: man:systemd-sysv-generator(8)
Process:25269 ExecStart=/etc/init.d/snort start (code=exited, status=0/SUCCESS)
Tasks:2 (limit:2317)
CGroup: /system.slice/snort.service
--25319 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort -g snort -c /etc/snort/snort.conf -S HOME_NET=[192.168.0.0/24] -i ens4
Oct 02 16:25:05 ubuntu systemd[1]:Starting LSB:Lightweight network intrusion detection system...
Oct 02 16:25:05 ubuntu snort[25269]: * Starting Network Intrusion Detection System snort
Oct 02 16:25:11 ubuntu snort[25269]: ...done.
Oct 02 16:25:11 ubuntu systemd[1]:Started LSB:Lightweight network intrusion detection system.
```

snort 的配置文件位于 */etc/snort* 文件夹中。

如果系统中没有 snort，则需要安装。由于 Linux 发行版和 snort 版本的不同，安装过程也会有所不同。一般做法是：

```
$ sudo apt-get update -y % -y 标志的意思是假设你的回答为“是”并静默安装，在大多数情况下不会向你询问。
$ sudo apt-get install -y snort*
```

注意，snort 是建构在 Linux 发行版的几个软件包上。通常，你可能会遇到所需的依赖包。Snort 是一个基于 libpcap 的数据包嗅探器/记录器，用作轻型网络入侵检测系统。Snort 具有基于

规则的日志记录功能，并且除了检测各种其他攻击和探测（如缓冲区溢出、隐形端口扫描、CGI 攻击、SMB 探测等）外，还可以执行内容搜索/匹配。Snort 具有实时警报功能，警报被发送到 syslog，一个单独的“警报”文件，甚至可以通过 Samba 发送给 Windows 计算机。

在 snort 的安装过程中，系统会询问一些设置问题。其中一些配置是在 */etc/snort/snort.debian.conf* 文件中设置的，如端口和家庭网络等。你可以手动修改配置文件，也可以运行以下命令重新配置设置：

```
$ sudo dpkg-reconfigure snort
```

2. 任务 2：运行 Snort

在本任务中，我们将介绍有关如何设置和运行 snort 的基础知识。

2.1 网络入侵检测系统设置

通常，要使 snort 成为网络 IDS，你可以按以下格式发出一个通用命令：

```
$ sudo snort -[options] -i [sniffing interface] -l [logfile] -h [home-net] -c [configuration file]
```

snort 命令后面给出的每个选项都是可选的。下面给出了一个示例：

```
$ sudo snort -dev -i ens5 -l /var/log/snort -h 10.0.0.0/8 -c /etc/snort/snort.conf
```

首先，我们需要了解 snort 网络基础设施的设置。下面的子节中有对其他选项的说明。本实验室的网络基础设施如图 CS-CNS-00003.2 所示。

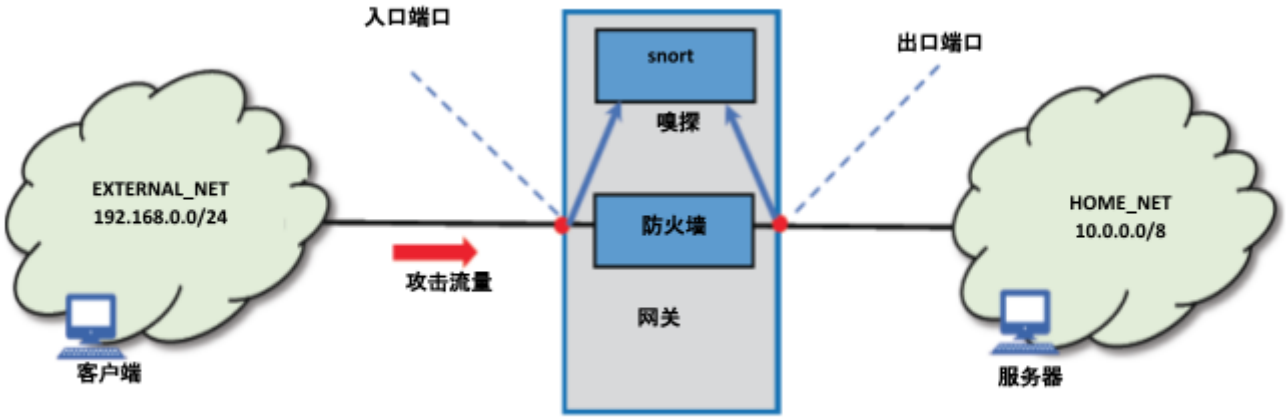


图 **CS-CNS-00003.2**
Snort 运行环境

注意：选项“-h”表示的是你要保护的网络。有了这个变量集，所有解码的数据包日志记录都是相对于家庭网络地址空间进行的。在我们的实验设置中，我们认为家庭网络是 10.0.0.0/8，连接的是服务器虚拟机，而网络 192.168.0.0/24 是外部网络，攻击者可能会在这里出现。在下面的示例和图解中，我们假设 IP 地址的设置如下：

- 客户端 IP：192.168.0.7
- 网关的入口 IP:192.168.0.3
- 网关的出口 IP:10.0.0.4
- 服务器 IP：10.0.0.6（运行 ssh、web、ftp、dns 服务等 ）

在网络入侵检测系统中，snort 通常被设置为嗅探模式。只捕获通过系统的流量，而不是将其放入“内联”以拦截流量。至于把 snort 放在何处来嗅探流量要取决于系统的安全设置和要求。

在图 CS-CNS-00003.2 中，snort 可以嗅探两个端口中的其中一个，即入口端口或出口端口。嗅探入口端口时，snort 会捕获发送到防火墙的所有流量。要检测针对防火墙的攻击，这项设置是合适的。但为了检测针对内部网络系统的攻击，snort 可能会消耗大量计算资源来执行检测，这并没有充分利用防火墙过滤功能的优势。因此，对于通过的流量，snort 通常被设置为嗅探防火墙已检查过的流量。

你可以在本实验室的入口端口或出口端口上设置嗅探端口。保证嗅探端口和 HOME_NET 的正确设置，以确保 snort 能够检测和识别数据包模式。

2.2 Snort 规则的图解

Snort 让用户可以自己编写规则，以生成传入/传出网络数据包的日志。用户只需要遵循 snort 规则的格式，其中数据包必须满足阈值条件。记住，snort 规则的编写包括两个主要部分：“报头”段和“选项”段。这些规则一般是由已知的入侵特征系统创建的。规则可以分为两个部分：

```
规则报头（规则选项）
```

snort 规则的格式如图 CS-CNS-00003.3 所示。

图 **CS-CNS-00003.3**
Snort 规则格式

报头部分包含的信息有：操作、协议、源 IP 和端口、网络数据包指向目标 IP 和端口等。其余内容将作为“选项”部分。

```
action protocol source port -> destination port (options)
```

在报头字段中：

- 操作：当发现与规则描述匹配的数据包时，它会通知 Snort 执行什么类型的操作。Snort 中现有 5 个默认的作业操作：警报、日志、传递、激活和动态，这些是用于定义规则操作的关键词。你也可以选用其他选项，比如 drop 和 reject。

- 协议：在选定规则中的操作选项后，你需要描述适用于该规则的特定协议（IP、TCP、UDP 或 ICMP）。
- 源 IP：报头的这一部分描述的是流量源头的发送方网络端口。
- 源端口：报头的这一部分描述的是流量来源的源端口。
- 指向运算符(“->”, “<>”)：表示发送方和接收方网络之间的信息流流向。
- 目标 IP：报头的这一部分描述的是用于建立连接的流量源的目标网络端口。
- 目标端口：报头的这一部分描述的是用于建立连接的流量源的目标端口。

因此，检测/匹配操作的逻辑考量分为两个层次：

- 在规则内：组成规则的所有元素必须为 true，才能执行指定的规则操作。当这些元素结合在一起时，可视为这些元素构成了一个 AND 逻辑语句。
- 在多条规则中：同时，可将 Snort 规则库文件中的各种规则视为构成一个大的 OR 逻辑语句。

2.3 规则选项 (Metadata)

规则选项的主体通常写在圆括号“（ ）”之间，其中包含带实参的关键词，并用分号“；”与另一个关键词隔开。注意，任何规则都不需要具体的规则选项部分；这些部分只是为了更严格地定义要收集或警告(或丢弃)的数据包。Metadata 是可选规则的一部分，该规则一般会包含有关 snort 规则的附加信息，该规则是在一些关键词及其参数详细信息的帮助下编写的。规则选项共有四大类别：

- 通用选项：这些选项包含提供有关流量信息的元数据。
- 有效载荷：这些选项都适用于数据包有效载荷所包含的数据，并且可以相互关联。
- 非有效载荷：这些选项适用于非有效载荷的数据。
- 后检测：这些选项是触发规则后发生的关联触发器。

snort 通用规则选项 (metadata) 如表 CS-CNS-00003.1 所示。

表 **CS-CNS-00003.1**

通用规则选项 (Metadata)

关键词	描述
msg	msg 关键词表示的是“Message”(消息)，通知 snort 在分析数据包时，应在日志中打印书面参数。
reference	“reference”关键词让规则可以引用互联网上其他可用系统（如 CVE）上的信息。
gid	“gid”关键词表示的是“Generator ID”，在启动特定规则时，用于标识 Snort 中具体创建该事件的部分。
sid	“sid”关键词表示的是“Snort ID”，用于标识不同的 Snort 规则。
rev	“rev”关键词表示的是“Revision”，用于表示 Snort 规则的不同修订。
classtype	“classtype”关键词用于将分类和优先级编号分配给群集，并将其区分为用于检测更一般的攻击类别的规则。 句法：配置分类：名称、描述、优先级编号。
priority	“priority”关键词是指为规则分配严重性等级。

注意：尽管大多数选项都是可选的，但 sid (Snort ID) 是必选的，并且不应与另一个规则的 sid 相冲突。sid 是每个规则的唯一标识符。Snort 中的 SID 值在 0-1000000 之间。

2.4 数据包嗅探器

snort 最简单的形式就是数据包嗅探器。换言之，这是开始探索 snort 的最简单的方式。

```
$ sudo snort -options
```

其中“options”（选项）可以是：

```
-v 将 Snort 置于数据包嗅探模式（仅限 TCP 报头）
-d 包括所有网络层报头（TCP、UDP 和 ICMP）
-e 包括数据链路层报头
-p 在不将端口置于混杂模式的情况下进行捕获
```

注意：选项“-d”和“-e”都会消耗系统资源，应避免在生产系统中使用这两个选项。

2.5 数据包记录器

Snort 具有内置的数据包记录机制，你可以利用这些机制将数据收集为文件，按目录分类，或将数据存储为二进制文件。

```
$ sudo snort -l {logging-directory} -h {home-subnet-slash-notation}
```

如果你想把数据记录到目录 /var/snort/logs 中，可以使用以下命令：

```
$ sudo snort -l /var/log/snort
```

如果你想指示 snot 在终端发出警报，则可以使用选项“-A console”：

```
$ sudo snort -A console -c /etc/snort/snort.conf
```

对于二进制格式的日志记录，你无需用到所有选项。二进制格式使 Snort 的数据包收集速度更快，因为 Snort 不必再立即将数据转换为人类可读的格式。

```
$ sudo snort -l {log-file} -b
```

选项“-b”表示二进制模式。二进制模式将 tcpdump 格式的数据包记录到日志记录目录中的单个二进制文件中。

要读取日志文件，你可以使用以下 snort 命令：

```
$ sudo snort [-d|e] -r {log-file} [tcp|udp|icmp]
```

命令行的最后一项是选用的，适用于你想根据 tcp、udp 或 icmp 等数据包类型过滤数据包的情况。

注意，snort 可以输出两种输出文件格式，具体取决于该文件的 snort 输出插件选项：

- tcpdump pcap 或
- snort’s unified2

为了解具体的生成格式，如 log file /var/log/snort/snort.log，你可以用 file 命令来检查日志文件：

```
$ sudo file /var/log/snort/snort.log
```

如果你被告知“tcpdump 捕获文件”，可用 wireshark、tshark-r、tcpdump-r 或 snort-r 来读取该文件。如果你被告知“数据是 unified2 格式”，这是一种“本机”snort 格式。你可以用 snort 中包含的

u2spewfoo <file> 来读取，亦或是用 u2boat 将其转换为 pcap。在本实验室中，只需添加“-b”选项即可创建 “tcpdump 捕获文件”格式，并且可以用 snort-r 来读取日志文件。

你还可以添加各种其他选项来调整 snort 的行为。例如，“-A Fast”可以启用较短的警报日志格式，这可以缩减磁盘访问和日志文件大小。无论你是否使用此选项，Snort 都会创建一个日志目录树，与你把 Snort 用作数据包嗅探器时其所创建的一样。但区别在于“Fast”日志文件包含的数据更少；仅记录符合规则的数据包。此外，还有一个名为“alert”的文件，该文件包含 snort 检测到的所有可疑活动的摘要。

要向 syslog 发送警报，你需要使用“-s”交换机。syslog 警报机制的默认功能是 LOG_AUTHPRIV 和 LOG_ALERT。要想为 syslog 输出配置其他功能，你要用 snort.conf 中的输出插件指令。

例如，用以下命令行将日志记录到默认（解码 ASCII）功能，并向 syslog 发送警报：

```
$ sudo snort -c /etc/snort/snort.conf -l /var/log/snort -h 10.0.0.0/8 -s
```

3. 任务 3：创建并测试 Snort 规则

要检查 Snort 是否如前所述记录了任何警报，你可以在“local.rules 文件”中的 IP 数据包上添加检测规则警报。要测试你新创建的 snort 规则，你可以禁用 snort 中显示的除 local.rules 外的所有 snort 规则文件。先备份 snort.conf：

```
$ cp snort.conf snort.backup.conf
```

再编辑 snort.conf 文件：

```
$ vim /etc/snort/snort.conf
```

通过在每个规则文件之前添加“#”符号，你可以注释掉所有 snort 规则文件，例如：

```
# site specific rules
include $RULE_PATH/local.rules % 不注释 local.rules
...
#include $RULE_PATH/app-detect.rules % 注释所有其他规则文件
#...
```

为此，你可以运行以下命令注释掉所有 include 语句，然后再取消注释包含 local.rules 的命令行：

```
$ sudo sed -i 's/include \$RULE_PATH/#include \$RULE_PATH/' /etc/snort/snort.conf
```

3.1 任务 3.1 创建并测试 icmp 规则

现在编辑 */etc/snort/rules/local.rules* 并添加下面的 icmp 规则：

```
alert icmp any any <> 192.168.0.3 any (msg:"ICMP Packet found"; sid:10000001; )
```

在该示例中，你可以将 snort 设置为嗅探 IP 地址为 192.168.0.3 的入口端口，如 ens4。“<>”选项意味着你想要捕获双向的 ping 消息。

要测试 icmp 规则，你可以运行以下 snort 命令：

```
$ sudo snort -A console -c /etc/snort/snort.conf -q -i ens4
```

对于测试，你可以在不使用日志记录选项的情况下发布 snort。在该命令中，选项“-q”用于抑制初始的 snort 信息消息。选项“-i”指定嗅探网络端口为 ens4，在本示例中配置为 IP 地址 192.168.0.3。接下来，你可以通过 ping 到另一个 IP 地址进行测试（例如，在我们的示例中，192.168.0.7是客户端的 IP 地址）；或是从 192.168.0.7，你也可以对 snort 嗅探端口 IP 地址 192.168.0.3 执行 ping。snort 检测结果会打印在控制台上，如图 CS-CNS-00003.4 所示。

```
root@ubuntu:/etc/snort# snort -A console -c /etc/snort/snort.conf -h 192.168.0.3
 -q -i ens4
10/04-07:34:53.203175  [**] [1:10000001:0] ICMP packet found [**] [Priority: 0]
{ICMP} 192.168.0.7 -> 192.168.0.3
10/04-07:34:53.203232  [**] [1:10000001:0] ICMP packet found [**] [Priority: 0]
{ICMP} 192.168.0.3 -> 192.168.0.7
10/04-07:34:54.204432  [**] [1:10000001:0] ICMP packet found [**] [Priority: 0]
{ICMP} 192.168.0.7 -> 192.168.0.3
10/04-07:34:54.204497  [**] [1:10000001:0] ICMP packet found [**] [Priority: 0]
{ICMP} 192.168.0.3 -> 192.168.0.7
```

图 **CS-CNS-00003.4**
Snort ICMP 警报

3.2 任务 3.2 用系统变量来创建和测试 snort 规则

为了更有效地管理 snort，你应该采用的是系统参数，而不是硬编码的数值。例如，你可以在规则设置中使用系统变量 EXTERNAL_NET 和 HOME_NET。这两个系统级变量可以在 /etc/snort/snort.conf 文件中进行设置。例如，你可以编辑以下内容：

```
ipvar HOME_NET 10.0.0.0/8 % 将服务器端网络设置为 HOME_NET

ipvar EXTERNAL_NET !$HOME_NET
```

在本示例中，受保护的 HOME_NET 被设置为网络 10.0.0.0/24。我们可以用“！”（即，“NOT”）来指定除 HOME_NET 以外的网络为“EXTERNAL_NET”。

从命令行启动 snort 时，你可以用“-h”选项覆盖 snort.conf 文件中给出的 HOME_NET 。例如，

```
$ sudo snort -A console -h 10.0.0.0/24 -c /etc/snort/snort.conf -q -i ens4
```

HOME_NET 可以是单个 IP 地址，也可以是多个 IP 地址，既可以是一个网络，也可以是混合网络。这里，我们列举了几个例子：

```
var HOME_NET [10.0.0.6,10.0.0.6,192.168.0.3] % 列出各个 IP 地址
ipvar HOME_NET 10.0.0.0/8 % 将服务器端网络设置为 HOME_NET
var HOME_NET [192.168.0.3,10.0.0.0/8] % 将服务器端网络和入口 IP 设置为
HOME_NET
```

接下来，你可以在 /etc/snort/rules/local.rules 中附加 TCP 规则，如下所示：

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"SSH attempt";flags:S;
    classtype:attempted-recon; sid:10000002; rev:0;)
```

此规则告知系统，当其检测到来自 $EXTER-NAL_NET 上的计算机的流量（指向 $HOME_NET（本地）计算机上的端口 22）时，会发出警报消息。括号内的信息包括：日志记录信息和分类代码。flags 的关键字是 TCP flags“S”，表明该规则仅匹配设置了 SYN flag 的数据包。在本示例中，Snort 定义的攻击分类位于 classification.config 文件中。关键字和值“classtype: attepmpted-recon”表示试图进行的信息泄漏攻击。在很多情况下，你还可以找到有关数据包内容（由内容关键字表示）或其他数据包功能的信息。

在你对 snort 规则集感到满意后，你就可以启动 snort。通常，你会让 snort 记录其输出，供你稍后细读。你可以用以下命令启动 snort，使其发挥 NIDS 的功能：

```
$ sudo snort -c /etc/snort/snort.conf -l /var/log/snort -b -q -i ens5 % 在本示例中，
ens5 是防火墙后面的出口端口
```

```
root@ubuntu:/etc/snort# sudo snort -A console -l /var/log/snort/ -c /etc/snort/
snort.conf -qb -i ens5
10/05-02:31:57.383548  [**] [1:10000002:0] SSH attempt'' [**] [Classification: A
ttempted Information Leak] [Priority: 2] {TCP} 192.168.0.7:41414 -> 10.0.0.6:22
```

图 CS-CNS-00003.5
Snort 捕获 ssh 连接。

图 CS-CNS-00003.5 中，显示了在出口端口（内部网络）捕获的 ssh 连接。

3.3 任务 3.3 用载荷规则选项来创建和测试 snort 规则

如前面的规则选项小节所述，有效载荷选项显示的是在数据包内容中捕获了什么数据。如果与参数数据字符串匹配的数据包含在数据包有效载荷内的任何位置，则测试成功，并执行其余的规则选项测试。默认情况下，匹配会区分大小写。例如，以下规则会匹配 HTTP 以获取消息。

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Http Get
request";content:"GET";classtype:web-application-activity;sid:10000003;rev:0;)
```

我们可以混合和匹配文本和二进制数据。我们将二进制数据的十六进制表示形式括在管道符（“|”）中，用以指定二进制数据。如果我们想指定一个以上的二进制字符，我们会用空格（“”）来分隔二进制字符。具体示例如下：

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (content:"|5c
00|P|00|I|00|P|00|E|00 5c|";)
```

现在，你可以通过从客户端向服务器发出 http 访问来测试上述指定的 http 获得检测规则：

```
$ curl 10.0.0.6  % 你可以直接访问 IP 地址并运行：curl
http://10.0.0.6
```

捕获的数据包如图 CS-CNS-00003.6 所示

```
root@ubuntu:/etc/snort# sudo snort -A console -l /var/log/snort/ -c /etc/snort/
snort.conf -qb -i ens5
10/05-03:12:25.515184  [**] [1:10000003:0] Http Get request [**] [Classificatio
n: access to a potentially vulnerable web application] [Priority: 2] {TCP} 192.
168.0.7:60274 -> 10.0.0.6:80
```

图 **CS-CNS-00003.6**
Snort 捕获的 TCP Get 消息。

3.4 任务 3.4 用非载荷规则选项来创建和测试 snort 规则

如前面的规则选项小节所述，非有效载荷选项适用于非有效载荷数据，如数据包报头字段或协议相关选项。例如，你可以用 ttl 关键词来检查 IP 生存时间值。此选项关键词旨在用于检测路由跟踪尝试。该关键词的取值范围为 0-255。

你也可以用 icmp_seq 来标识收到的 icmp 消息的数量。例如：

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP echo request
message NO.7";classtype:icmp-event;icmp_seq:7;sid:10000004;rev:0;)
```

现在，你可以通过发出 ping 消息来测试 ping：

```
$ ping -c 8 192.168.0.7
```

然后，snort 数据包的检测结果会如图 CS-CNS-00003.7 所示。

```
root@ubuntu:/etc/snort# sudo snort -A console -l /var/log/snort/ -c /etc/snort/
snort.conf -qb -i ens4
10/05-03:38:23.319439  [**] [1:10000004:0] ICMP echo request message NO. 7 [**]
 [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.0.7 -> 192.1
68.0.3
```

图 **CS-CNS-00003.7**
Snort 捕获到序列号为 7 的 icmp 消息。

4. 任务 4：实验室要求

在前面的任务中，我们介绍了几个关于如何设置和运行 snort 的演练。要设置实验室运行环境，你需要在一个简单的客户端-网关-服务器网络系统中工作，如图 CS-CNS-00003.2 所示。在该环境中，你应该：

1. 启用网关上的数据包转发。注意，你有两个 snort 可以嗅探流量的端口。出于测试目的，你可以嗅探网络 IP（入口端口 IP）的客户端。但一些攻击场景（如 smurf 攻击）可能需要你设置防火墙（例如，设置实验室 CS-CNS-00001 数据包过滤防火墙中给出的系统），以模拟真实的防火墙系统，并对网络的服务器端（即出口端口 IP）实施嗅探。此外，你也可以在服务器上设置 Apache、ssh、FTP 和 DNS 服务。
2. 如果你在网关上建立了防火墙，如 iptables，出于测试目的，你可以禁用所有网络过滤规则，并启用黑名单（iptables-P chain name ACCEPT），以允许所有流量通过。在本实验室中，我们认为这种设置符合最低要求。
3. 在下面描述的所有要求中，你需要对 /etc/s- nort/snort.conf 中规定的所有规则进行注释。但 /etc/snort/rules/local.rules 规则文件除外，在此文件中，你要做的是实现和演示你的 snort 规则。

4.1 任务 4.1 Land 攻击部署和检测

当攻击主机发送伪造的 TCP SYN 数据包(连接启动)并将目标主机的 IP 地址和 TCP 端口作为源和目的地时，就会发生 Land 攻击。Land 攻击之所以有效，是因为它会让计算机不断地自我应答。也就是说，目标主机会通过向自身发送 SYN-ACK 数据包进行响应，创建一个空连接，该连接将持续至达到空闲超时值为止。用这种空连接淹没系统可能会让系统不堪重负，造成拒绝服务 (DoS) 的情况。

在该任务中，你需要：

1. 用 hping3(参见 LAB-CNS-10003 hping3 教程)部署从客户端到服务器的 Land 攻击。攻击要求：
   - TCP 报头：源端口 = 80，目的地端口 = 80, TCP SYN Flag=1。
   - IP 报头：源 IP = 目标主机的 IP 地址，目的地 IP = 目标主机的 IP 地址
2. 实现 snort 规则来检测 Land 攻击。

注意，最新的 Linux 操作系统通常附带地址欺骗攻击保护。如果此消息通过网关节点(例如 Linux)，它可能会停止转发伪造的数据包。若要在网关上禁用地址欺骗保护，你可以执行以下操作：

1. 在 */etc/sysctl.conf* 中设置以下配置：

```
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.ip_forward = 1
net.ipv4.conf.all.accept_redirects = 1
net.ipv4.conf.all.send_redirects = 1
```

2. 重新启动网关。

然后，网关将能够转发伪造的数据包。

4.2 任务 4.2 洪水攻击的部署和检测

当主机因发起不完整连接请求的 TCP SYN 数据包变得不堪重负，而无法再处理合法连接请求时，就会发生 SYN 洪水攻击。事实上，当客户端系统试图与提供服务的系统(服务器)建立 TCP 连接时，客户端和服务器会交换一系列消息，这被称为“三次握手”。客户端系统首先向服务器发送 SYN(同步)消息。然后，服务器通过向客户端发送 SYN-ACK(确认)消息来确认 SYN 消息。继而，客户端通过用 ACK 消息应答来完成连接的建立。

当使用伪造的源 IP 地址将 SYN 消息发送到受害服务器时，则会创建 TCP 不完整连接。因此，受害服务器将永远不会收到来自欺骗客户端的 ACK 消息，从而无法完成连接的建立。用这种伪造的 TCP SYN 流量淹没受害服务器可能会使服务器不堪重负，导致拒绝服务的情况。

在该任务中，你需要：

1. 用 hping3(参见 LAB-CNS-10003 hping3 教程)部署从客户端到服务器的 SYN 洪水攻击。攻击要求：
   - TCP 报头：源端口 = 任意，目的地端口 = 公开 TCP 端口号，TCP SYN flag=1。
   - 源 IP = 伪造或随机 IP 地址(如 192.168.0.100)，目的地 IP = 目标主机的 IP 地址

2. 实现 snort 规则来检测 SYN 洪水攻击。特殊阈值检测要求：你需要在 60 秒的时间间隔内记录此攻击的每 20 个事件。

4.3 任务 4.3 Smurf 攻击的部署和检测

Smurf DoS 攻击包括向广播 IP 地址发送大量 ICMP 回应请求 (ping) 流量，所有这些流量都具有受攻击主机的伪造源 IP 地址。由该广播 IP 地址表示的网络上的每台主机将接受 ICMP 回应请求，并向受害主机发送 ICMP 回应回复，将通信量按响应主机的数量翻倍。

在该任务中，你需要：

1. 用 hping3（参见 LAB-CNS-10003 hping3 教程）部署从客户端到服务器的 Smurf 攻击。
   - ICMP 报头：type=8 (回应请求), code=0。
   - IP 报头：源 IP = 受害主机的 IP 地址，目的地 IP = 广播 IP 地址
2. 实现 snort 规则来检测 Smurf 攻击。具体要求如下：
   (a) 受害者是服务器（10.0.0.x），目标广播网络是 192.168.0.0/24，广播 IP 地址是 192.168.0.255。
   (b) 默认情况下，Ubuntu 不会响应广播 ping。要启用针对网关的广播 ping，你需要通过以下任一方法更改网关的设置：

```
$ sudo echo 0 >/proc/sys/net/ipv4/icmp_echo_ignore_broadcasts % 启用对广播请求的临时响应
$ sudo sysctl net.ipv4.icmp_echo_ignore_broadcasts 0
```

   或者，你也可以通过添加以下命令行来修改 /etc/sysctl：

```
net.ipv4.icmp_echo_ignore_broadcasts=0 % 你可能需要重新启动节点
```

   更改后，你可以向网关发送广播消息，例如 192.168.0.255，并且你会看到一条从网关发送到服务器 IP 地址（即伪造的 IP 地址）的回复消息。

   (c) 你需要创建一个 snort 规则来检测来自网关的 smurf 攻击（icmp 回复消息）。

4.4 任务 4.4 UDP 洪水攻击的部署和检测

UDP 洪水攻击包括用 UDP 数据包淹没受害系统上的目标 UDP 端口。如果有足够的 UDP 数据包被传送到目的地 UDP 端口，受害主机或 UDP 应用程序可能会减速或停机。

在该任务中，你需要：

1. 用 hping3（参见 LAB-CNS-10003 hping3 教程）部署从客户端到服务器的 UDP 洪水攻击。提示：在此攻击中，你需要将网关的 IP 地址锁定在客户端
   - UDP 报头：源端口 = 任意，目的地端口 = 公开 UDP 端口号。
   - IP 报头：源 IP = 伪造或随机 IP 地址，目的地 IP = 目标主机的 IP 地址。
2. 实现 snort 规则来检测 UDP 洪水攻击。特殊阈值检测要求：你需要在 60 秒的时间间隔内记录此攻击的每 10 个事件。

4.5 任务 4.5 端口扫描的部署与检测

端口扫描是一种确定网络上哪些端口处于打开状态的方法。由于计算机上的端口是发送和接收信息的地方，因此端口扫描类似于敲门看是否有人在家。

在该任务中，你需要：

1. 用 hping3（参见 LAB-CNS-10003 hping3 教程）部署从客户端到服务器的端口扫描。提示：在此攻击中，你需要将网关的 IP 地址锁定在客户端

   - TCP ACK 扫描
   - TCP FIN 扫描
   - TCP Xmas 扫描
   - TCP Null 扫描（你可能会使用 nmap）。
   - UDP 扫描

2. 实现 snort 规则以检测上述扫描。特殊阈值检测要求：你需要在 60 秒的时间间隔内记录此攻击的每 20 个事件。

5. 实验室评估（50 分）

完成任务 4 的实验室评估取决于以下完成情况：

1. （10 分）成功演示任务 4.1 中的攻击和检测
2. （10 分）成功演示任务 4.2 中的攻击和检测
3. （10 分）成功演示任务 4.3 中的攻击和检测
4. （10 分）成功演示任务 4.4 中的攻击和检测
5. （10 分）成功演示任务 4.5 中的攻击和检测

6. 相关信息和资源

Snort 用户手册 2.9.16：
http://manual-snort-org.s3-website-us-east-1.amazonaws.com/snort_manual.html
Snort IDS 工具：
http://books.gigatux.nl/mirror/snortids/0596006616/main.html
Snort 命令行选项：
http://books.gigatux.nl/mirror/snortids/0596006616/snortids-CHP-3-SECT-3.html
Snort
Snort 配置：
https://www.sbarjatiya.com/notes_wiki/index.php/Snort_configuration