# *网络安全数据可视化与可视分析案例*

# 基于堆叠流图的网络时序可视化

# 数据说明

## 数据1 VAST Challenge 2012 MC2 数据集介绍

Bank of Money公司是一家世界级的银行，它的营业网点和自动取款机有近百万个，公司的网络也异常复杂。公司的总部是整个网络运转的中心，有超过4000台主机和1000台服务器。近期，公司总部的网络出现了一些异常现象，怀疑网络已经被入侵，但海量的网络安全监控数据让管理人员无从下手，因此希望能够针对提供的网络安全监控数据使用可视分析方法，来帮助管理人员寻找数据中值得关注的异常事件，并试图解释这些异常事件发生的原因。

公司提供了20M的入侵检测系统日志，以CSV文本形式给出，时间跨度是两天。入侵检测系统依照一定的安全策略和智能算法对网络流量和运行状况数据进行分析，一旦发现可疑的主机行为和攻击企图将会发出警报并且记录下来形成入侵检测系统日志数据集，其数据结构如下表所示。

表1-1 入侵检测日志数据的主要数据项与相关说明

|  |  |
| --- | --- |
| **数据项** | **相关说明** |
| id | 日志数据的编号； |
| rule id | 规则编号； |
| rule descript | 规则描述； |
| priority | 警报优先级； |
| datetime | 记录时间； |
| source IP | 源IP； |
| destination IP | 目的IP； |
| source port | 源端口； |
| destination port | 目的端口； |
| remaining lines | 每条记录保留的其他信息。 |

## 数据2 VAST Challenge 2013 MC3 数据集介绍

Big Marketing公司是一家以广告与公关为主营业务的公司，于2013年3月15日收到自称为“Butterfly Warriors”组织的恐吓信，该组织宣称将发起网络攻击，破坏公司的日常运营。从4月1日到4月15日的两个星期，公司的网络管理人员与该恐怖组织展开了一场激烈的网络攻防对抗，且为公众提供了这两个星期内的网络安全监控数据，试图破解黑客组织的攻击手段与攻击规律。

公司提供的网络安全监控数据集包含两周的**NetFlow网络流量监控日志**、**Big Brother主机状态监控日志**和一周的**入侵防御系统日志**，原始数据为CSV格式，涉及内网主机和服务器约1900台，其中NetFlow日志约7000万行，Big Brother日志约550万行，IPS日志约1600万行，原始日志量近10G。

**（1）Netflow网络流量监控日志数据集：**

Netflow日志数据集清楚的记录了一次完整的流量行为，其数据结构如下表所示。该数据集共包含19项元素，分别表示一项网络活动中的时间、所通过的协议类型、源/目的地址、源/目的端口、持续时间、传输的源/目的包字节、源/目的包数目以及该次行为是否持续等。

表1-2 NetFlow流量监控数据的主要数据项与相关说明

|  |  |
| --- | --- |
| **数据项** | **相关说明** |
| TimeSeconds | 标准的UNIX时间值，指定为最后一个数据包流入的时间； |
| parsedDate | 解析后的时间，即更易于理解的时间格式； |
| dateTimeStr | 解析时间的字符串版本； |
| ipLayerProtocol | 流量所用协议的数值型编号，例如：TCP编号为6；UDP编号为17； |
| ipLayerProtocolCode | 流量所用协议的编号的文本型编号； |
| firstSeenSrcIp | 数据流中出现的第一个源地址； |
| firstSeenDestIp | 数据流中出现的第一个目的地址； |
| firstSeenSrcPort | 数据流中出现的第一个源端口； |
| firstSeenDestPort | 数据流中出现的第一个目的端口； |
| moreFragments | 标记该数据流是否是一个长数据流的一部分。当该字段不为0时说明该记录存在后续记录（持续发生），直到该字段的值为0为止； |
| contFragments | 标记该数据流是否是一个长数据流的一部分。当该字段不为0时说明该行数据不是从这里开始的； |
| durationSeconds | 该数据流传输时持续的时间，单位为秒； |
| firstSeenSrcPayloadBytes | TCP和UDP报文头所标识的有效载荷字节总和；TCP或UPD协议源地址发出的有效载荷字节数，该字段数据的计算方式为：表头长度（约55bt）\*包总数＋源流量＝源流量总数（加表头）； |
| firstSeenDestPayloadBytes | 协议目的地址接收的有效载荷字节数，该字段数据的计算方式为：表头长度（约55bt）\*包总数＋源流量＝源流量总数（加表头）； |
| firstSeenSrcTotalBytes | 源地址传输的数据流字节总数； |
| firstSeenDestTotalBytes | 目的地址接收到的数据流字节总数； |
| firstSeenSrcPacketCount | 从数据流中记录的第一个源地址传来的数据包数； |
| firstSeenDestPacketCount | 从数据流中记录的第一个目的地址接收的数据包数； |
| recordForceOut | 在传输有效时间截至前被存入数据文件的记录，通常用来表示程序关闭时未被正确传输的数据。 |

**（2）Big Brother主机状态监控日志数据集：**

Big Brother 是一个主机状态监控软件，通过运行该客户端软件，可以向服务端发送和记录重要服务器和主机的状态。数据中包括了对主机与服务器的网络连接能力、CPU和磁盘的使用率、内存和虚拟内存的使用情况以及SMTP邮件服务器的状态信息，如表1-3所示。表1-4列出了本文使用的Big Brother状态监控数据的主要数据项与相关说明。

表1-3 Big Brother状态监控数据的状态监控代码与相关说明表

|  |  |  |
| --- | --- | --- |
| **状态监控代码** | **状态监控代码说明** | **监控对象** |
| conn | 监测系统的网络连接情况 | 服务器和主机 |
| cpu | 检测系统的cpu使用率 | 服务器 |
| disk | 监测系统的硬盘使用率 | 服务器和主机 |
| mem | 监测系统内存的使用情况 | 服务器和主机 |
| pagefile | 监测系统的页面文件使用情况 | 服务器 |
| smtp | 监测SMTP邮件服务器的状态 | 邮件服务器 |

表1-4 Big Brother状态监控数据的主要数据项与相关说明表

|  |  |
| --- | --- |
| **数据项** | **相关说明** |
| ID | 数据的编号； |
| Hostname | 被监测主机的名称； |
| Servicename | 被监测主机的服务类型； |
| Currenttime | 日志数据的监测时间，时间为标准的UNIX时间； |
| statusVal | 被监测主机的状态值 |
| Bbcontent | 被监测主机的详细信息。根据不同的服务类型，记录的详细信息也不同。 |
| Receivedfrom | 被监测主机的IP地址。 |
| distUsagePercent | 当服务类型（Servicename）为‘disk’时，表示被监控主机硬盘的使用率。 |
| pageFileUsagePercent | 当服务类型（Servicename）为‘pagefile’时，表示被监控服务器的页面文件使用情况。 |
| numProcs | 当服务类型（Servicename）为‘cpu’时，表示被监控服务器的处理器数目。 |
| loadAveragePercent | 当服务类型（Servicename）为‘cpu’时，表示被监控服务器的平均装载率。 |
| physicalMemoryUsagePercent | 当服务类型（Servicename）为‘cpu’时，表示被监控服务器物理内存使用率。 |
| connMade | 当服务类型（Servicename）为‘conn’时，表示被监控的服务器和主机是否连接成功。 |
| parsedDate | 日志数据的监测时间，精确到秒 |

**（3）入侵防御系统日志数据集：**

入侵防御系统IPS与入侵检测系统IDS都通过预先设定的安全策略，以网络状况获取的数据为基础，通过规则匹配尽可能发现各种攻击行为，以保护网络安全。所不同的是，IPS可以主动的执行某些操作，比如对恶意报文进行丢弃，对恶意流量进行限流等等，并且通常安装在网络边际，与防火墙的位置类似，而IDS以检测与报警为主要功能。本文使用的IPS日志数据的主要数据项说明如下表所示。

表1-5 IPS日志数据的主要数据项与相关说明

|  |  |
| --- | --- |
| **字段** | **说明** |
| Date/time | 记录的产生时间 |
| Priority | 优先级，比如Info、Warning、Error、Alert等等 |
| Operation | 操作类型，有Built、Teardown、Deny等 |
| Message code | 相应的操作对应的消息码 |
| Protocol | 协议类型，有TCP、UDP、ICMP等 |
| Source IP | 源IP |
| Destination IP | 目的IP |
| Source port | 源端口 |
| Destination port | 目的端口 |
| Destination service | 目的端口所提供的服务 |
| Direction | 通常有inbound和outbound，表示连接方向 |
| Flags | 标记位信息 |

## 数据使用

本案例数据集使用情况如下：

|  |  |  |
| --- | --- | --- |
| **数据集来源** | **子数据集** | **是否使用** |
| VAST Challenge 2012 MC2 | **（IDS）入侵检测系统日志** | **√** |
| VAST Challenge 2013 MC3 | **（NetFlow）网络流量监控日志** | **√** |
| （Big Brother）主机状态监控日志 |  |
| **（IPS）入侵防御系统日志** | **√** |

# 可视分析任务

本案例希望通过多种类型的时序数据对比分析，寻找网络中可能发生的异常事件。具体包括如下几个分析需求：

**R1：从多粒度和多对象查看网络安全时序数据。**用户可以选择不同的时序统计粒度，主要包括：5分钟、30分钟、1小时；用户可以选择不同的统计对象，主要包括：不同流量、不同协议、不同IDS报警类型、不同IPS连接方向、不同IPS操作类型、不同主机。

**R2：同时展示同一种类不同对象的网络安全时序数据。**例如，用户可以查看同一时间段内IPS日志（同一种类）中不同操作（不同对象）的记录情况，即同时展示日志中建立连接（Built）、撤销连接（Teardown）和拒绝连接（Deny）三种操作量情况。

**R3：同时展示两类对象的网络安全时序数据。**例如：用户想同时查看IPS日志记录数和Netflow日志记录数（两类对象）；又例如：用户想同时查看IPS日志中inbound和outbound两个网络访问方向的记录数，并同时查看Netflow日志中源IP总流量和目的IP总流量（两类，4个对象）

**R4：注重多时序的对比分析。**网络安全数据种类繁多、流量类型多样、数据庞大，单纯比较一对流量数据难以得到预期分析结果，需结合多组流量时序数据进行比较分析，找出网络异常。

# 数据处理

针对R1需求，本案例需要对网络安全数据进行相应的处理，主要是数据对象和对象类别的定义与不同粒度的时序数据提取。

对于数据1，其数据类别和相应对象如下所示：

|  |  |
| --- | --- |
| **数据类别** | **相应数据对象** |
| 源IP | 分为不同的源IP |
| 目的IP | 分为不同的目的IP |
| 源端口 | 分为不同的源端口 |
| 目的端口 | 分为不同的目的端口 |
| 报警 | 分为IRC authorization、IPC Unicode share access等报警类型 |

对于数据2，其数据类别和相应对象如下所示：

|  |  |
| --- | --- |
| **数据类别** | **相应数据对象** |
| 源IP | 分为不同的源IP |
| 目的IP | 分为不同的目的IP |
| 源端口 | 分为不同的源端口 |
| 目的端口 | 分为不同的目的端口 |
| 连接方向 | 分为inbound和outbound |
| 优先级 | 分为Info、Warning、Error、Alert等 |
| 操作类型 | 分为Built、Teardown、Deny等 |
| 协议类型 | 分为TCP、UDP、ICMP等 |
| 流量汇总 | 分为源IP流量汇总和目的IP流量汇总 |

由于网络安全数据具有时序性，在将不同类别数据的不同对象进行整理之后，还需对数据的时序统计粒度进行定义，主要包括：5分钟、30分钟、1小时，即需要分别以5分钟、30分钟、1小时的时间间隔对各个数据类别及其相应对象的记录数（或流量）进行统计。以源IP数据为例，以5分钟为统计粒度得到的数据形式如下所示：

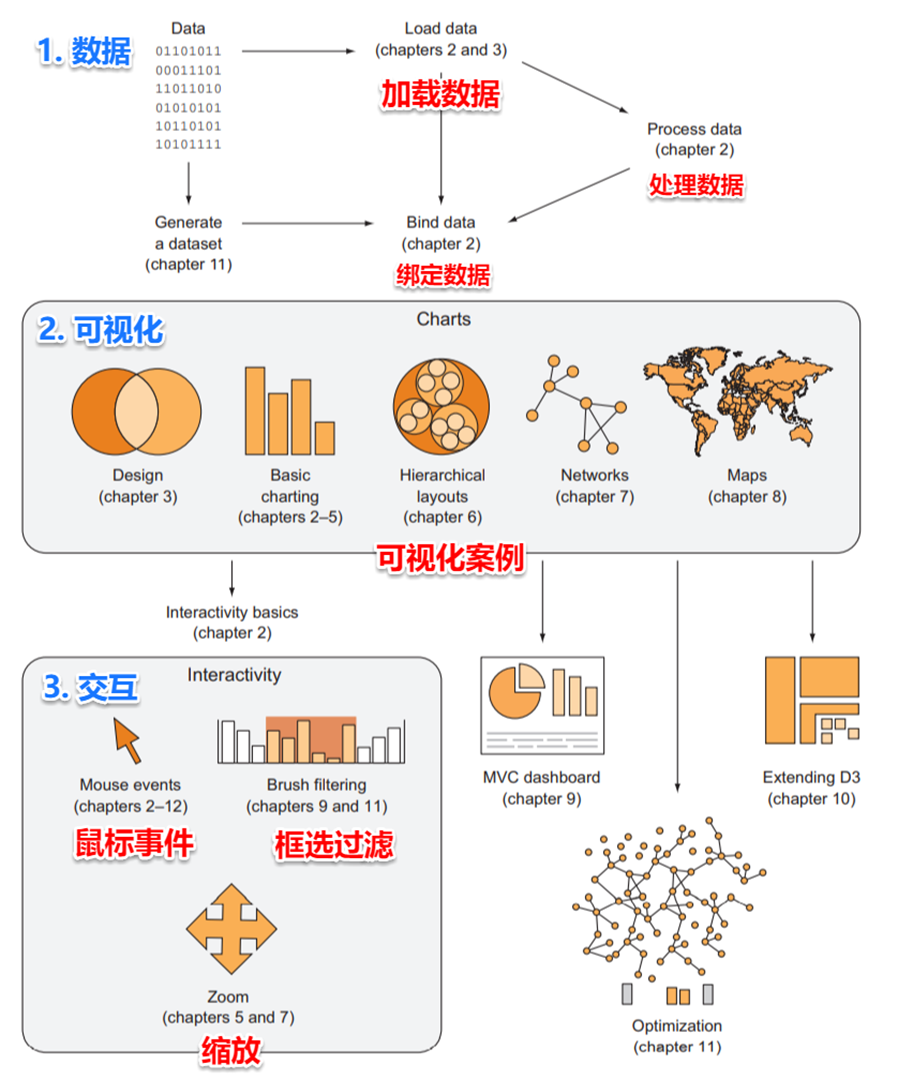
|  |  |
| --- | --- |
| **字段类型** | **取值说明** |
| 时间节点 | Ti（表示统计的开始时间节点） |
| 不同源IP | IPx（表示待统计的源IP名称） |
| 记录数 | N（表示从Ti到Ti+5min时间段内源IPx在数据源中出现的总记录次数） |

# 可视化设计

## 时序图

针对R1需求，时序图可将一组时序数据转为对应的面积图。

作为第二个案例，我们D3.js可视化图形库为例，大致讲解最简单的面积图的绘制流程，具体流程如下图所示，主要从**数据、可视化、交互**三个方面进行阐述。为方面初学者快速上手开发，我们通过使用d3.area()方法，快速生成对应图形生成器，对输入的一组数据渲染出对应的面积图；此外，我们还提供对应的**参考文档和参考案例链接**。

****

D3数据可视化基本流程

**Step1：数据：**

首先，D3提供了方法可以对不同的数据类型进行加载，比如d3.text(), d3.xml(), d3.json(), d3.csv(), 和d3.html()，分别使用对应格式的方法可将预处理的数据进行导入，且单独为时序列数据定义时间解析函数，如：

*var parseTime = d3.timeParse("%y/%m/%d %H:%M:%S");*

其次，为载入数据定义恰当的比例尺，方便将数据进行屏幕大小的映射，对应的D3方法为d3.scaleLinear() 线性比例尺和d3.scaleTime() 时间比例尺；

然后，D3还提供select("父标签").data(数据集) .enter().append("元素标签")一组链式语法，可以实现数据与对应元素的动态绑定。

参考文档：

[https://github.com/d3/**d3-fetch**/tree/v3.0.1#readme](https://github.com/d3/d3-fetch/tree/v3.0.1#readme)

[https://github.com/d3/**d3-dsv**#readme](https://github.com/d3/d3-dsv#readme)

[https://github.com/d3/**d3-time-format**/tree/v3.0.0#readme](https://github.com/d3/d3-time-format/tree/v3.0.0#readme)

[https://github.com/d3/**d3-scale**/blob/main/README.md](https://github.com/d3/d3-scale/blob/main/README.md)

**Step2：可视化：**

可视化部分即对整个图表进行可视化，包含图表骨干和图表内容两大部分。

图表骨干体现为图表坐标轴和标签。D3提供的d3.axisLeft()、d3.axisBottom()、d3.axisTop()、d3.axisRight()方法和事先定义的横轴纵轴比例尺相结合可绘制出对应的坐标轴，ticks()方法能定义坐标轴的刻度线。

图表内容部分即数据可视化的核心部分。D3提供d3.area()方法来定义面积生成器，从而绘制面积图。通常一个区域由两条边界线进行定义，可以是直线、折现或曲线。面积图通常由一条基线和曲线包围而成。

参考文档：

[https://github.com/d3/**d3-axis**/blob/main/README.md](https://github.com/d3/d3-axis/blob/main/README.md)

[https://github.com/d3/d3-shape/blob/v3.0.1/README.md#**areas**](https://github.com/d3/d3-shape/blob/v3.0.1/README.md#areas)

**Step3：交互：**

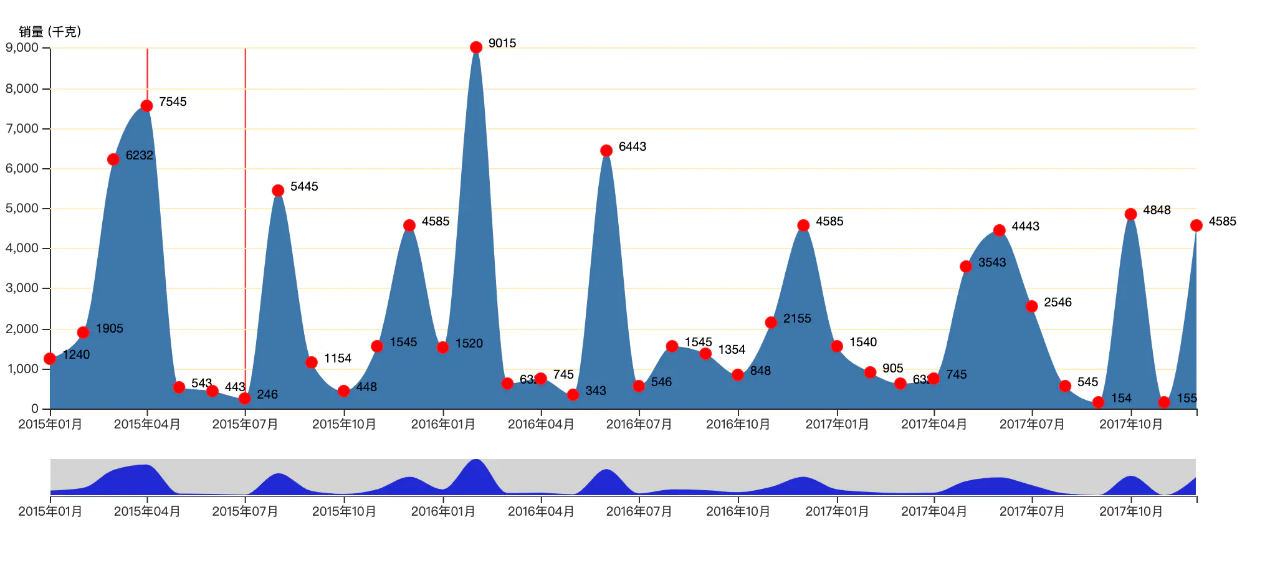
交互行为的添加能增强图表的实用性，针对面积图，通常使用的有“Zoom”和“Brush”交互，分别表示对面积图添加局部图像缩放操作和缩略图拖拽查看操作。其中D3提供的d3.zoom()和d3.brush()方法分别对应着图表的Zoom和Brush操作。

参考文档：

[https://github.com/d3/d3-**zoom**#readme](https://github.com/d3/d3-zoom#readme)

[https://github.com/d3/d3-**brush**#readme](https://github.com/d3/d3-brush#readme)

参考案例：<https://www.jianshu.com/p/7dcf4486950d#comments>



## 堆叠时序图

针对R2需求，堆叠时序图可以同时展示相同数据类别下不同的数据对象的时序图像。通过使用d3.stack()工具，能为具有相同时序索引的不同对象数据列生成相应的堆叠时序图，即堆叠面积图。由于Step1和Step3中的操作与时序图的操作相似，主要区别在于Step2可视化上，故主要介绍Step2。

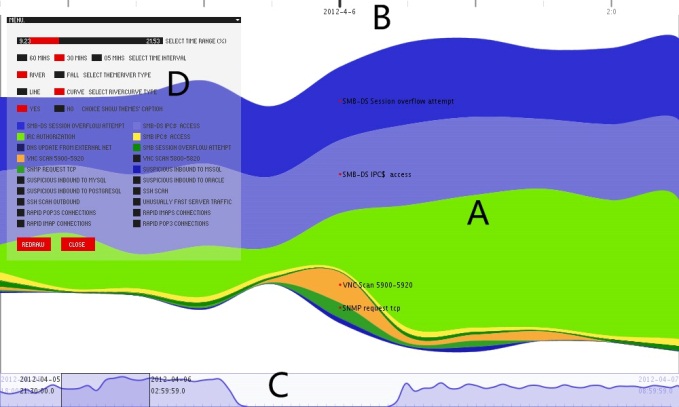
**Step2：可视化：**

堆叠时序图与时序图的区别在于，时序图仅表现为单列数据的面积图，而堆叠时序图则展示的是具有相同时序索引的多列数据的堆叠面积图。图表内容同时使用D3提供的d3.area()和d3.stack()方法来对多个区域的面积进行绘制，相互堆叠起来，从而形成堆叠面积图。

而河流图与堆叠面积图十分相似，两者都使用d3.stack()进行布局，两者区别在于河流图堆叠的基准线不再是X轴，可通过使用stack.offset([offset])方法进行快速定义。

参考文档：[https://github.com/d3/d3-shape/blob/v3.0.1/README.md#**stack**](https://github.com/d3/d3-shape/blob/v3.0.1/README.md#stack)

参考案例：<https://blog.csdn.net/zjw_python/article/details/98592543>

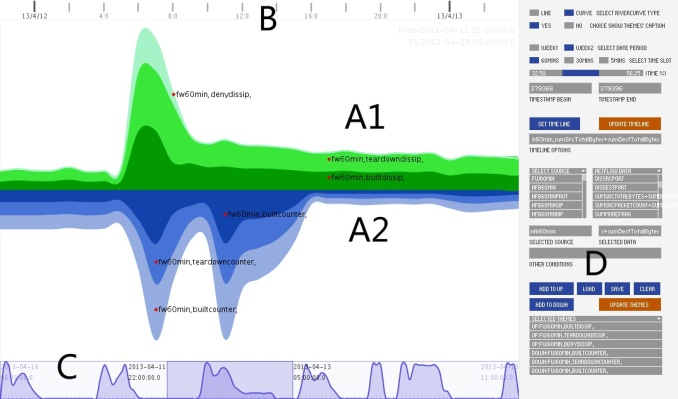


## 对比堆叠时序图

针对R3需求，通过将两组堆叠时序图以水平基准线为中心，分别以垂直向上和垂直向下的方式对堆叠进行渲染，可有效比较一组同一时间段内的不同类时序数据。

**Step2：可视化：**

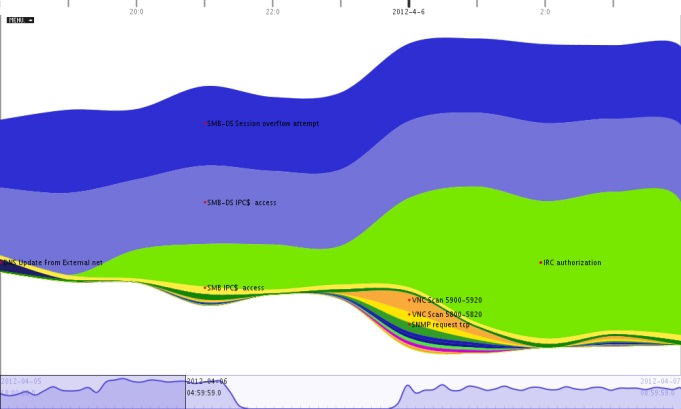
可发现，对比堆叠时序图由上下两部分堆叠面积图组合而成，故通过修改两组数据的比例尺映射方向就可生成对比堆叠时序图。



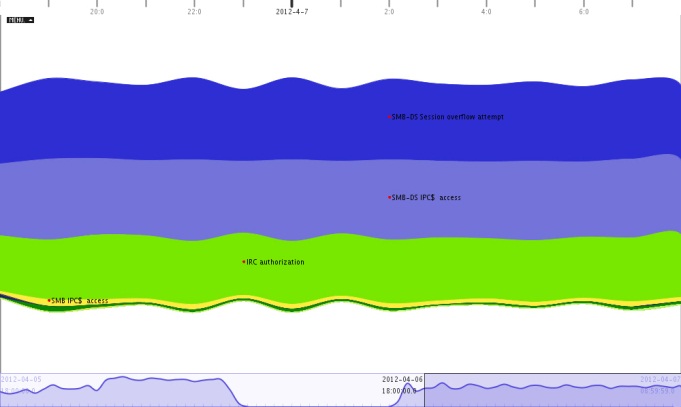
# 案例分析

## 基于堆叠流图的IDS数据实例分析

VAST Challenge 2012提供的IDS日志数据包含20个预先定义的报警类型，本案例以60分钟为统计间隔，以各报警类型出现次数为分析对象，采用传统堆叠流图进行可视分析。如图5-1所示，堆叠流中从上到下前2个蓝色的流是报警类型“IPC Unicode share access”和“NTMLSSP Unicode asn1 overflow”，它们从第一个时刻就出现开始，而且一直维持稳定，占据整个堆叠区域一半以上的面积。这说明这两种报警类型在整个日志中出现的次数较多，而且非常平稳，占据整个时段的大部分的报警比例，很有可能是规则参数设置不太合理，形成了大量的误报。



（a）IDS日志第一天的情况



（b）IDS日志第二天的情况

图5-1 基于堆叠流图的IDS日志可视分析

报警类型“IRC authorization”在图5-1（a）中比较引人注意，它在2012-4-5日19:00第一次出现，从2012-4-5 23:00开始趋势明显增长，并且在后面两天都保持了较高的频率。除了这三种报警次数较多的报警类型以外，其它十多种报警类型出现次数都不多，但是有明显的聚集效应，并且和“IRC authorization报警类型出现时间有一定的时间关联度，比如：在2012-4-5 12:00，IRC报警类型首次出现后不久，几种报警类型首次并且同时出现；在2012-4-6 23:00 IRC报警爆发的同时，近20种报警类型同时出现，特别是 “Inbound to MSSQL port 1433”、“Inbound to Oracle SQL port 1521”、“IMAP Connections Possible Brute Force Attack”等与数据库服务器和邮件服务器相关的报警类型值得高度关注。由于IRC协议经常被僵尸网络利用，大量的IRC报警以及其它报警的关联出现，说明很有可能网络已经被入侵，僵尸病毒控制了网络中的一些主机，而且数据库服务器的报警反映出有可能部分敏感数据已经被窃取了。

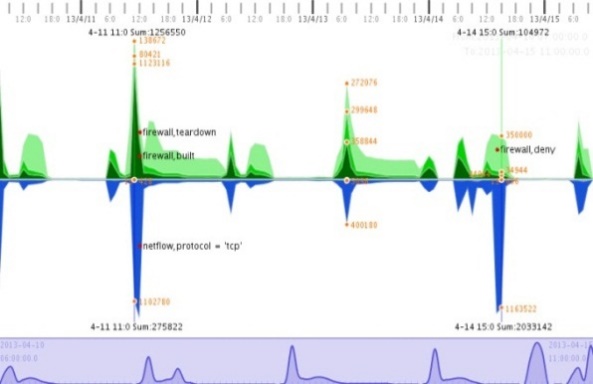
## 基于对比堆叠流图的多源数据实例分析

本案例结合VAST Challenge 2013提供的Netflow日志和IPS日志，采用对比堆叠流图进行可视分析。

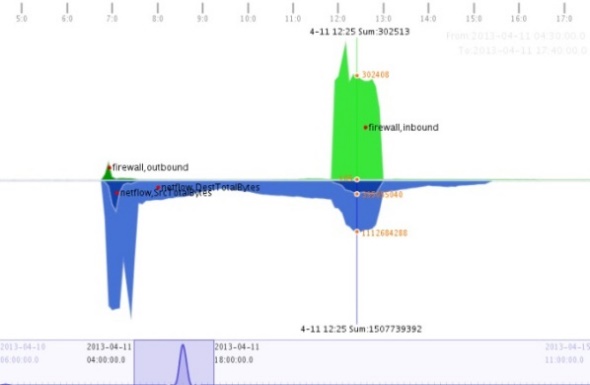
第一步，如图5-2（a）所示，底部的总时间线区域选取了2013-4-10到2013-4-15的**NetlFow日志中网络流量变化情况**，上方堆叠区域显示了同一时间段**IPS日志中建立连接、撤销连接和拒绝连接三种操作量情况**，下方堆叠区域显示了同一时段**NetFlow日志中TCP、UDP和其它协议记录数情况**，时间间隔为60分钟。图5-2（a）显示NeFflow日志中TCP协议记录数远远大于UDP和其它协议，并且在2013-4-11 12:00以及2013-4-15 15:00出现了两次峰值，特别是2013-4-11 12:00左右，IPS的记录数也出现了一个峰值，这说明这个时段存在网络异常的可能性非常高。

第二步，将时间聚焦，显示内容更改为2013-4-10到2013-4-15的**NetFlow日志记录数的时序变化情况**，可以看到明显的高峰，如图5-2（b）所示。然后将堆叠区的时间段聚焦在2013-4-11 12:00左右，时间间隔选为5分钟。上方堆叠区选择了**IPS日志中inbound和outbound两种网络访问方向的记录数**时序变化情况，下方堆叠区选择了**NetFlow日志中源IP流量汇总和目的IP流量汇总的时序变化情况**。在2013-4-11 12:00到13:00，上方的防火墙日志inbound记录数比outbound多很多，说明这个时段绝大部分都是外网对内网的访问，而这时的流量并不高，甚至还不如2013-4-11 7:00的流量。可以初步判断，在2013-4-11 12:00到13:00有外网主机对内网主机进行了异常访问，访问记录数极高，而访问过程中产生的流量并不大，应该属于恶意攻击。

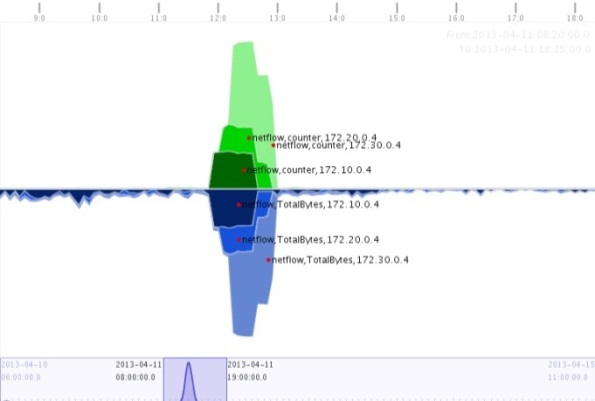
第三步，初步确定网络中存在异常现象后，为了获得更多的网络异常信息，**我们对内网服务器的连接和流量情况进行检查**，如图5-2（c）所示，我们选取了十台内网服务器，上方堆叠区显示它们的**NetFlow记录数**，下方堆叠区显示它们的**流量**，每台主机对应一条流，时间间隔5分钟。在2013-4-11 12:00到13:00，有三台服务器的流量和连接数都非常高，而这三台服务器都是Web服务器，所以，可以确定这三台服务器遭受了攻击。



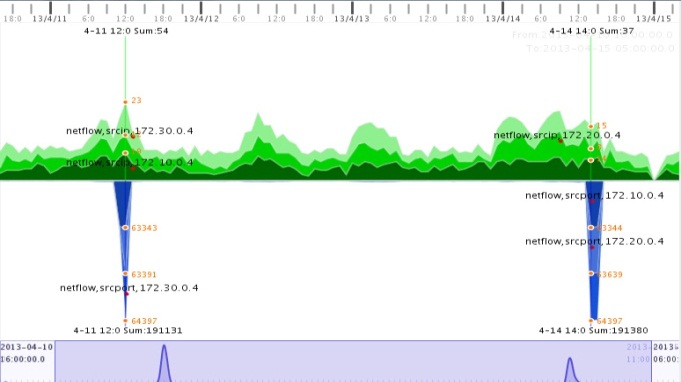
（a）第一步分析



（b）第二步分析



（c）第三步分析



（b）第四步分析

图5-2 基于对比堆叠流图的NetFlow日志和IPS日志协同分析

第四步，接下来进一步分析攻击手段。时间范围选择2013-4-10到2013-4-15，时间间隔改为60分钟，上方堆叠区选择**访问三台服务器的不同源IP数**作为分析对象，下方堆叠区选择**访问三台服务器的不同源端口数**作为分析对象。如图5-2（d）所示，在2013-4-11 12:00，访问三台受攻击Web服务器的不同源端口数达到了六万以上，而源IP数不超过25台，说明这次攻击是由少数外网主机通过大量的不同端口发起。

综上所述，可以判断在2013-4-11 12:00左右，三台企业内网Web服务器遭受到了一些外部主机发起的协同攻击，攻击的方法是通过60000多个不同的源端口对Web服务器的80号端口发起连接，反映出明显的DDoS攻击特征。