第 3 章　基于路由器 DDoS 检测的改进 CUSUM 算法

根据已有的技术，结合在路由器环境下的网络实际情况，本章在传统 CUSUM 算法的基础上设计、实现了矩阵式的多统计量 CUSUM 算法（M-CUSUM 算法）对网络端口输入输出流量的变化进行监控，以及时地检测出流量的异常。

利用改进的 M-CUSUM 算法对输入、输出端口流量的绝对差与和之比进行统计，实时地监控其均值的偏移情况，及时地检测出流量的异常，并进一步地利用第 4 章介绍的 AFCAA（Anomaly Traffic Character Aggregation Algorithm）算法对异常流量特征进行提取和过滤。通过在模拟路由环境下进行的实验，该改进的 M-CUSUM 算法取得了很好的效果。

3.1　DDoS 流量统计特征分析

传统的异常流量攻击，无论是 DoS 攻击还是 DDoS 攻击，都是用同一种方法，即用大量的垃圾数据包来堵塞网络，使得网络终端过于繁忙，超出了其正常运行的范围，使其不能完成正常的服务。

虽然 DDoS 攻击可以任意地伪造数据包，但是大多数攻击的流量还是有一定的特征可寻的，因为当今许多的 DDoS 攻击都不是有专业技能的黑客发起的，而是由一些人运用黑客编写的网络攻击程序而发起的。由于网络的发展和计算机的普及，使得黑客工具能够很容易在网上下载，使用方式也越来越简单。

基于非专业黑客运用黑客工具发起的 DoS/DDoS 攻击，其一般不会自己去修改黑客程序的代码，而是以默认的方式去攻击，更加使得攻击的特征有迹可寻。此外，由于 DoS/DDoS 攻击主要是运用大量的数据包来堵塞网络，因此为了达到攻击的效果，必须使发包的数据足够快，如果攻击方对每个攻击数据包都使用不同生成方法的话，必然会使发包的速度变慢，影响攻击的效果。

对于上面的结论，对 MIT 实验室所发布的 Darpa 数据集进行了分析，以 LLS_DDoS_2.0.2-inside.dump 为例来证明以上结论的正确性。

3.1.1　分析步骤

（1）先把 dump 数据集用 ethereal 读入，利用 ethereal 保存其所有详细信息，得到的保存文件大小为 828 636KB，文件很庞大。

（2）把保存得到的详细信息文件再加工，得到需要统计的字段，每个 IP 包转换为一条记录。

(3) 把记录追加到数据库中,按照各种需求进行统计分析。

数据表结构如下:

```
struct OneFrame{
/*
Frame Number: 1

    Arrival Time: Apr 17, 2000 02:45:01.903961000
    Time delta from previous packet: 0.000000000 seconds
    Time relative to first packet: 0.000000000 seconds
*/
int frame_no;
char frame_t1[64];
char frame_t2[16];
char frame_t3[16];
/*
Ethernet II

    Destination: 00:c0:4f:a3:57:db (falcon.eyrie.af.mil)
    Source: 00:10:5a:9c:b2:8e (3Com_9c:b2:8e)
    Type: IP (0x0800)
*/
char ethe_s[32];
char ethe_d[32];
char ethe_proto[16];
/*
Internet Protocol, Src Addr: www.a-be.org (194.7.248.153), Dst Addr: falcon.eyrie.af.mil
(172.16.112.194)
    Header length: 20 bytes
    Total Length: 41
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Protocol: TCP (0x06)
    Source: 172.16.112.20 (172.16.112.20)
    Destination: mill.eyrie.af.mil (172.16.115.20)
*/
int ip_len1;
int ip_len2;
char ip_s[20];
char ip_d[20];
char ip_flag[8];
char ip_proto[16];
/*
Transmission Control Protocol,Src Port: 63281 (63281),Dst Port: telnet (23),Seq: 2286481677,
Ack: 1493952196
    Source port: 63281 (63281)
    Destination port: telnet (23)
    Sequence number: 2286481677
    Next sequence number: 2286481678
    Acknowledgement number: 1493952196
    Header length: 20 bytes
```

```
    Flags: 0x0018 (PSH, ACK)
User Datagram Protocol, Src Port: 59155 (59155), Dst Port: domain (53)
    Source port: 59155 (59155)
    Destination port: domain (53)
    Length: 53
    Checksum: 0x2eef (correct)
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
*/
char port_s[16]; //for both tcp and udp
char port_d[16]; //for both tcp and udp
char seq_num[16]; //only for tcp
char ack_num[16]; //only for tcp
char tcp_flag[8]; //only for tcp
int icmp_type; //only for icmp
}
```

举例数据集首尾记录内容如下：

```
1  Apr 17, 2000 02:45:01.9039610000.0000000000.000000000 00:c0:4f:a3:57:db  00:10:5a:9c:
b2:8e  IP 20  41(194.7.248.153)(172.16.112.194)0x04  TCP 63281  telnet  0x0018 - 1
…
347987 Apr 17, 2000 04:27:48.357079000  0.018801000  6166.453118000 08:00:20:89:a5:9f
00:c0:4f:a3:57:db  IP 20  40(172.16.112.194)(172.16.112.50)0x04  TCP 28167  telnet
0x0010 - 1
```

3.1.2　结果分析

(1) 整个数据集的时间跨度：

Apr 17,2000 02:45:01.903961000——Apr 17,2000 04:27:48.357079000

(2) 整个数据集中各种协议包数量的比例：

整个数据集记录数为 347 987

在 ETHE 层分析：

协　　议	数　　目	比例(%)
0	728	0.21
ARP	587	0.17
IP	346 672	99.62

ETHE 层协议 0 是 LLC(Logical-Link Control)包。

在 IP 层分析：

协　　议	数　　目	比例(%)
0	1315	0.38
ICMP	11	0.00
TCP	279 890	80.43
UDP	66 771	19.19

IP 层协议"0"是非 IP 包(LLC+_ARP+...)。

(3) 分析整个数据集中的流量异常。

根据目的地址取 TOP10,如表 3-1 所示。

表 3-1 各领域应用光学成像举例

名次	目的 IP	包个数
1	(131.84.1.31)	52 080
2	(172.16.115.20)	44 323
3	(172.16.112.50)	34 200
4	(172.16.112.100)	18 081
5	(172.16.113.168)	18 052
6	(172.16.112.194)	13 856
7	(194.7.248.153)	11 057
8	(172.16.113.148)	10 078
9	(172.16.113.207)	9669
10	(172.16.116.194)	9641

对这些地址逐个进行流量时间的分布分析(按分钟间隔)。

只有第一个目的地址异常,52 080 个包中的 51 556 个集中在一分钟内,比例高达 98.99%。

第二、三个目的地址一分钟内最大的包数(一分钟内包数/总包数)分别为:2228/44 323=5%,1140/34 200=3.33%。从 DoS 攻击的本质看,必须在一定时间内发送大量包使目的机器超负荷,而对于第二、三和其他的目标地址,很明显在最高峰时也没有受到集中攻击,所以只有第一个地址可能被攻击。

(4) 对目的地址(131.84.1.31)的攻击行为的进一步分析。

对该地址流量的时间分布:

02:59 51

03:25 66

03:27 102

03:29 19

03:46 51

03:47 19

03:48 39

04:03 67

04:04 49

04:06 51556

04:07 42

04:08 19

可以看到,实际攻击在时间"04 点 06 分"发生,并且在"04 点 07 分"就结束了。

04 点 06 分内对目的地址(131.84.1.31)发包的源 IP 有 51 556 个(A 类网址),MAC 地址只有一个(00:10:5a:9c:b2:8e)。

攻击开始时间：Apr 17,2000 04:06:15.757497000

攻击结束时间：Apr 17,2000 04:06:23.407973000

```
MAC0    00:10:5a:9c:b2:8e(攻击者)
MAC1    08:00:20:89:ba:28(攻击目标)
```

在 Apr 17,2000 04:07:01.075327000 (131.84.1.31)的 MAC 地址从 MAC1 变为 MAC2,估计是攻击生效,目标备份启动(物理设备变化了)：

```
MAC 200:c0:4f:a3:57:db (攻击目标)
```

该 DDoS 攻击产生的异常流量区别于正常流量的特征：

- 总是发 ACK 包,为洪水攻击(TCP flag=0x0010)。
- 顺序生成源端口。
- 应答序列号一致(Seq=0)。
- 随机生成目的端口。
- 所有攻击 TCP 包长度固定(40,即不带数据)。

对于此次攻击,只要利用以上的一个或多个特点,就可以对其进行过滤。同样地,对 UDP 和 ICMP 的 DDoS 攻击也做了相似的分析,得到了下面的结论：

UDP 攻击可能有的特征：

- 攻击数据包的长度固定。
- 随机生成目的端口。
- 随机生成源端口。

ICMP 攻击可能有的特征：

- 攻击数据包的长度固定。
- 攻击包的类型固定(ICMP Flag 固定)。

虽然对于不同的攻击,可能特征不同,但是总可以找到某种特征来过滤攻击,除非是经心设计的攻击方法,才能使攻击流和正常流不可区分。所以此方法对于一般性的攻击都是有效的。

需要特别提出的是,与基于特征的入侵检测判断手段不同的是,这里指的特征值都是可变的,是基于统计的,由一定的方法从攻击流量中动态生成的,而不是从单个攻击包中寻找到的。

可以看出,DoS/DDoS 攻击都会使网络的流量发生大的变化,从而使路由器端口的输入与输出比在统计特性上发生异常。常见的 DoS/DDoS 攻击,在攻击前后某端口流量统计特征值的变化如图 3-1 所示。

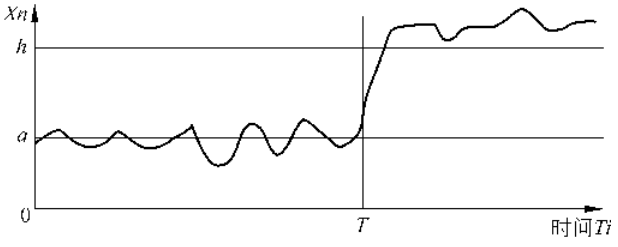

图 3-1　异常流量攻击过程中某端口统计特征值变化图

图 3-1 对发生异常流量攻击前后端口输入输出统计特征变化进行了描述。从图中可以明显地看出，在 T 时刻该统计特征值发生了明显的变化，由原来维持在 a 上下波动上升到了 h，使该端口的流量趋于饱和，并且攻击过程中将持续保持这个水平。其统计特征与流量的自相似性明显地发生了变化。

基于此统计特性的变化规律，本章利用 CUSUM 算法来发现这种特征，从而准确地检测网络的异常流量。利用本章的算法，观察 CUSUM 的累计统计值变化规律如图 3-2 所示。

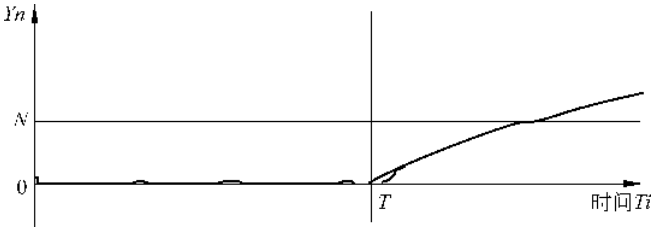

图 3-2 CUSUM 累计值攻击过程中的变化图

基于累积的 CUSUM 算法能够对任意所关心序列进行累计，本章使用非沉积的 CUSUM 算法，使其稳定状态下的累积值保持在 0 左右的一个域值内，而当流量异常攻击发生时，累计值将会发生明显的变化，并且在有限个统计周期的时间内到达系统设定的门限 N，产生异常流量攻击报警。

结合上述对其的简单应用描述，3.2 节将对 CUSUM 算法进行详细描述，并且结合特定的网络环境，提出适用于路由器端检测异常流量的改进算法——M-CUSUM。

3.2 CUSUM 算法描述

CUSUM 算法是工业异常监控常用的算法[82,84,85]，它可以检测到一个统计过程均值的变化。CUSUM 算法基于这一事实：如果有变化发生，随机序列的概率分布也会改变。

在该算法中，令 $x_1, x_2, \cdots, x_t$ 为独立的 $N(0,1)$同分布，$x_{t+1}, x_{t+2}, x_{t+3}, \cdots, x_{t+n}$为独立的 $N(\delta,1)$同分布，其中 t 为未知变点，对于给定的观察序列 $x_1, x_2, x_3, \cdots, x_n$，假设 $t=v(v<n)$ 对于原假设 $t=\infty$的似然比统计量为(以 $\phi(\cdot)$表示标准正态分布 $N(0,1)$的分布密度函数)：

$$L_{n,v}=\frac{\prod_{i=1}^{v}\phi(x_i)\prod_{i=v+1}^{n}\phi(x_i-\delta)}{\prod_{i=1}^{n}\phi(x_i)}=\frac{\prod_{i=v+1}^{n}\phi(x_i-\delta)}{\prod_{i=v+1}^{n}\phi(x_i)}=\exp\left\{\delta\sum_{i=v+1}^{n}\left(x_i-\frac{\delta}{2}\right)\right\} \quad (3.1)$$

由于 $\prod_{i=n+1}^{n}\phi(x_i)=1$，$\sum_{i=n+1}^{n}x_i=0$，对数化为 $A_{n,v}=\ln L_{n,v}=\delta\sum_{i=v+1}^{n}\left(x_i-\frac{\delta}{2}\right)$。

假设变量 $x_1, x_2, \cdots, x_t$ 与 $x_{t+1}, x_{t+2}, x_{t+3}, \cdots, x_{t+n}$有偏移，那么其对数似然统计量为

$$\Lambda_n=\max_{1\leqslant v<n}\Lambda_{n,v}=\max\left\{\delta\sum_{i=v+1}^{n}\left(x_i-\frac{\delta}{2}\right)\right\} \quad (3.2)$$

假设检测的为向上偏移，即 $\delta>0$，则上述的对数似然统计量等价于下面的统计量

$$Z_n = \max_{1\leqslant v<n} \sum_{i=v+1}^{n} \left(x_i - \frac{\delta}{2}\right) \tag{3.3}$$

定义 1：设 $n-1$ 个观测值没有均值偏移，即 $Z_i \leqslant h, i=1,2,\cdots,n-1$（$h$ 为门限）。如果在时刻 n，满足 $x_n-\delta/2>h$，或 $x_n+x_{n-1}-\delta>h$，或 $x_n+x_{n-1}+x_{n-2}-3\delta/2>h$，…，或 $x_n+x_{n-1}+\cdots+x_1-n\delta/2>h$，则这个过程发生了均值偏移。

记 $\tilde{x}_i = x_i - \frac{\delta}{2}$，且 $\tilde{x}_0 = 0, \tilde{S}_k = \sum_{i=0}^{k} \tilde{x}, \tilde{S}_0 = 0$，于是可得

$$Z_n - Z_{n-1} = \tilde{x}_n - \min\{0, \tilde{S}_n - \min_{0\leqslant v\leqslant n-1} \tilde{S}_v\} = \max\{\tilde{x}_n, \tilde{x}_n - \tilde{S}_n + \min_{0\leqslant v\leqslant n-1} \tilde{S}_v\}$$

$$= \max\{\tilde{x}_n, \min_{1\leqslant v\leqslant n-1} \tilde{S}_v - \tilde{S}_{n-1}\} = \max\{\tilde{x}_n, -Z_{n-1}\} \tag{3.4}$$

用不定参数 k 代替 $\delta/2$，就得到了 Z_n 的递推公式：

$$Z_n = \max\{0, Z_{n-1} + x_n - k\}, \quad n = 1,2,\cdots \tag{3.5}$$

这就是 CUSUM 算法。若设定报警门限为 $h>0$，如果在第 n 个观察点，$Z_n>h$（$Z_i \leqslant h$，$i=1,2,\cdots,n-1$），则报警，确定在过程 n 以前的统计量发生了均值偏移。

3.3　基于路由器的改进 CUSUM 算法（M-CUSUM）

通常，CUSUM 需要随机序列的参数模型，以便可以用概率密度函数来监控序列。但因特网是一个非常动态而复杂的实体，因特网业务模型的理论结构是一个复杂的问题，因而一个主要的难题是如何模拟随机序列（X_n），而非参数方法不是具体的模型，它更适合于分析因特网。非参数 CUSUM 算法的主要思想是累积明显比正常运行情况下平均水平高的 X_n 值。算法的优点之一是它能以连续方式监控输入的随机变量，从而达到实时检测。

基于路由器的特殊网络环境，本章对 CUSUM 算法做一些改进措施。由于每个流经路由器的流量都是从路由器的某个端口进来，再从某个端口转发出去，基于其多端口环境，因此本章对多个端口的不同目的 IP 流同时进行检测，运用矩阵的方法对每个路由特定端口的特定目的 IP 流设定统计数值。

记 $\{x_{n,m}\}$ 为在第 n 个时间段，第 m 个流入某端口的统计量，则有

$$\delta_{n,m} = (1-\beta) \times \delta_{n-1,m} + \beta \times x_{n,m}, \delta_{0,m} = x_{0,m} \tag{3.6}$$

$$Z_{n,m} = x_{n,m} - \delta_{n,m} - d \tag{3.7}$$

$$S_{n,m} = \sum_{i=0}^{n} Z_{i,m}, S_{0,m} = 0 \tag{3.8}$$

$$Y_{n,m} = S_{n,m} - \min_{1\leqslant k\leqslant n} S_{k,m} \tag{3.9}$$

其中 $\delta_{n,m}$ 为第 m 个端口统计序列 $\{x_{n,m}, n=1,2,3,\cdots\}$ 的均值，β 为 EWMA（Exponentially Weighted Moving Average）系数。通常情况下取 $\beta=0.01\sim0.03$，d 为使统计量 $E(Z_{n,m})$ 在正常情况下小于 0 的偏移。

定义 2：在 $n\times m$ 的随机矩阵中，对于第 m 列序列的统计量，如果在 $t-1$ 时间段内没有检测出异常，那么在 t 时刻检测出异常当且仅当（其中 h 为门限）

$$Y_{n,m} \leqslant h, \quad n = 1,2,3,\cdots,t-1$$
$$Y_{t,m} > h$$

此为改进的 CUSUM 算法。

由以上定义可得

$$Y_{n,m} - Y_{n-1,m} = Z_{n,m} - \min\left\{0, S_{n,m} - \min_{1\leqslant k\leqslant n-1} S_{k,m}\right\} = \max\left\{Z_{n,m}, Z_{n,m} - S_{n,m} + \min_{1\leqslant k\leqslant n-1} S_{k,m}\right\}$$
$$= \max\left\{Z_{n,m}, \min_{1\leqslant k\leqslant n-1} S_{k,m} - S_{n-1,m}\right\} = \max\{0, -Y_{n-1,m}\}$$

有下面的递推式：

$$Y_{n,m} = (Y_{n-1,m} + Z_{n,m})^+, \quad Y_{0,m} = 0 \tag{3.10}$$

其中 X^+ 定义为：

$$X^+ = \begin{cases} x & x > 0 \\ 0 & x \leqslant 0 \end{cases} \tag{3.11}$$

$Y_{n,m}$ 增长的越快，该端口遭到的攻击就越强。

3.4 M-CUSUM 算法检测路由器端网络异常流量

3.4.1 端口统计量分析

通过对网络中的流量进行实时监测，根据检测得到的数据，总结各种典型网络访问的规律，得知典型的路由器端口信息流量具有以下观测事实，核心路由器某个端口输入输出的流量存在一定的统计特征。

(1) 正常使用模式端口下入出流量基本持平：

$$C_{out}(n) \sim C_{in}(n) \quad 0 \leqslant E\left(\frac{|C_{in}(n) - C_{out}(n)|}{C_{out}(n) + C_{in}(n)}\right) < t < 1 \tag{3.12}$$

(2) 异常流量攻击下网络端口流量的变化：

$$C_{out}(n) \ll C_{in}(n) \text{ 或 } C_{in}(n) \ll C_{out}(n), \quad 1 > E\left(\frac{|C_{in}(n) - C_{out}(n)|}{C_{out}(n) + C_{in}(n)}\right) \gg t \tag{3.13}$$

式中 C_{in} 表示输入路由器某端口的流量，C_{out} 表示输出路由器某端口的流量，t 表示正常模式下统计量观测值上限，对于具体网络 t 的值也不同。可以在正常情况下做一定时期的观察获得，具体方法为：

$$t = \max\left\{E\left(\frac{|C_{in}(n) - C_{out}(n)|}{C_{out}(n) + C_{in}(n)}\right), \quad n = 1,2,3,\cdots \text{when attack is not happened}\right\} \tag{3.14}$$

3.4.2 算法分析

文献[82]已经证明，在一个时间序列中的值是独立的并且是同一参数模型的同一分布，那么 CUSUM 对于多变点检测问题是近似最佳的。将 CUSUM 应用于上述的随机序列 (X_n) 有两个必要条件：一是随机变量之间的依赖性随着时间增长而下降，二是随机变量的

值是有限的。

对于路由器的某个端口 m，定义在第 n 时间段的统计量为：

$$x_{n,m}=\frac{|C_{in}(n,m)-C_{out}(n,m)|}{C_{out}(n,m)+C_{in}(n,m)},\quad n=1,2,3,\cdots \tag{3.15}$$

由于 $Z_{n,m}$是由因特网业务衍生的，而因特网上长程依赖过程非常普遍，因此 $Z_{n,m}$抽样之间的依赖性会随着间隔的增长而衰减。又由于在正常情况下 $x_{n,m}\in(0,t)$，$t<1$，$Z_{n,m}=x_{n,m}-\delta_{n,m}-d$，其中 $\delta_{n,m}$和 d 是有限的常数，因此 $Z_{n,m}$的值也是有限的，检测变量 $Z_{n,m}$可以很容易地满足以上两个必要条件。

DoS/DDoS 攻击检测系统有两个关键指标：错误告警率和检测时间。然而，这两个参数是相互矛盾的，很难在缩短检测时间的同时降低错误告警率，因此必须在这两个指标之间进行折中。

根据前面的分析，检测时间 T_a 和算法攻击反应时间 ρ_n 可以定义如下：

$$T_a=\inf\{n:Y_n>h\} \tag{3.16}$$

$$\rho_n=T_n-T_a \tag{3.17}$$

其中 inf 为下确界，T_a 为攻击的开始时间。

从以上的论述可知，参数的设置和检测效率的关系如下：

算法通过选择最佳参数 d 和 h 来降低告警速率和缩短检测时间。设定的 d 越大，在 (Z_n)中出现正值的可能性就越小，因此测试统计量 Y_n 累积到一个较大的值来显示攻击的可能性就越小。h 是 Y_n 的攻击门限，h 越大，错误告警的速率就越低，但检测时间会越长。

参数设定方法：

根据一般情况下攻击对 $x_{n,m}$产生的抖动情况，可以设定：

$$d=\mu\times\delta_n\quad \mu\in(0.05,0.25) \tag{3.18}$$

$$h=\lambda\times\delta_n\quad \lambda\in(10,20) \tag{3.19}$$

其中 μ 为偏移比率，λ 为门限倍数，端口均值 $\delta_n=\sum_{i=1}^{m}\delta_{n,i}$。

由式(3.6)、式(3.7)和式(3.17)得：

$$\rho=\inf\left\{k:\sum_{i=0}^{k}(x_{n,m}-\delta_m-d)>h\right\} \tag{3.20}$$

偏移比率 μ 和门限倍数 λ 都为实验值。我们做了大量的攻击实验，一般的攻击都会使均值产生大于 50%的变化，在严重情况下甚至大于 250%，而变化率过小的攻击，产生的攻击效果也不好，对网络产生的危害不大。这样，设定 μ 和 λ 使系统能在不多于 10 个时隙内检测出高强度的攻击(对端口 m 的统计数据 $x_{n,m}$的震动幅度大于其均值 250%左右的攻击)，平均 30 个时隙内检测出低强度攻击(对端口 m 的统计数据 $x_{n,m}$的震动幅度为其均值 50%左右的攻击)。利用式(3.20)可以算出设定具体参数的 CUSUM 算法对攻击的反应时间，有利于参数的调整。由于网络中流量随着时间的变化而变化，因此需要系统能够根据参数 μ 和 λ 实时地得到适合当前网络流量的 h 和 d。

在系统正式运行时，首先设定数据采集模块的采集频率，然后要在正常的网络环境运行一段时间，使系统能够根据式(3.6)计算出较稳定的 δ_n，然后再利用控制模块设定的 $\mu\in$

(0.05,0.25)和 $\lambda \in (3.5,7.5)$；系统用式(3.18)和式(3.19)计算出所需要的门限 h 和零值偏移 d，然后系统才能进行检测工作。

3.5 本章小结

本章首先分析了 DDoS 流量统计特征，然后详细描述了 CUSUM 算法，并设计了一种基于路由器的改进 CUSUM 算法——M-CUSUM 算法。最后通过利用 M-CUSUM 算法检测路由端网络异常流量，并对该算法的优劣性进行了分析。