IPv6 Security Lab - Netflow

Exploring Netflow

Netflow identifies anomalous and security-related network activity by tracking network flows. NetFlow data can be viewed and analysed via the command line interface (CLI), or the data can be exported to a commercial or freeware NetFlow collector for aggregation and analysis. NetFlow collectors, through long-term trending, can provide network behaviour and usage analysis. NetFlow functions by performing analysis on specific attributes within IP packets and creating flows. Version 5 is the most commonly used version of NetFlow, however, version 9 is more extensible and is required to support IPv6. NetFlow flows can be created using sampled traffic data in high-volume environments. Cisco Express Forwarding (CEF) is a prerequisite to enabling NetFlow.

NetFlow can be configured on routers and switches. In older releases of Cisco IOS software, the command to enable NetFlow on an interface was:

```
ip route-cache flow
```

In newer releases of Cisco IOS (12.4 onwards), the command has been replaced by:

```
ip flow {ingress | egress}
```

The following configuration illustrates the basic configuration of this feature.

```
ip flow-export destination <ip-address> <udp-port>
ip flow-export version <version>
!
interface fastethernet 0/0
 ip flow ingress
 ip flow egress
!
```

The following is an example of NetFlow output from the router command line interface. The `SrcIf` attribute can aid in traceback.

```
router#show ip cache flow
IP packet size distribution (26662860 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448
480
   .741 .124 .047 .006 .005 .005 .002 .008 .000 .000 .003 .000 .001 .000
.000

    512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .001 .007   .039   .000   .000  .000  .000  .000   .000

IP Flow Switching Cache, 4456704 bytes
  55 active, 65481 inactive, 1014683 added
  41000680 ager polls, 0 flow alloc failures
```

```
  Active flows timeout in 2 minutes
  Inactive flows timeout in 60 seconds
IP Sub Flow Cache, 336520 bytes
  110 active, 16274 inactive, 2029366 added, 1014683 added to flow
  0 alloc failures, 0 force free
  1 chunk, 15 chunks added
  last clearing of statistics never

Protocol        Total   Flows  Packets  Bytes  Packets Active(Sec) Idle(Sec)
--------        Flows   /Sec    /Flow   /Pkt     /Sec     /Flow     /Flow
TCP-Telnet      11512    0.0      15     42      0.2      33.8      44.8
TCP-FTP          5606    0.0       3     45      0.0      59.5      47.1
TCP-FTPD         1075    0.0      13     52      0.0       1.2      61.1
TCP-WWW         77155    0.0      11    530      1.0      13.9      31.5
TCP-SMTP         8913    0.0       2     43      0.0      74.2      44.4
TCP-X             351    0.0       2     40      0.0       0.0      60.8
TCP-BGP           114    0.0       1     40      0.0       0.0      62.4
TCP-NNTP          120    0.0       1     42      0.0       0.7      61.4
TCP-other      556070    0.6       8    318      6.0       8.2      38.3
UDP-DNS        130909    0.1       2     55      0.3      24.0      53.1
UDP-NTP        116213    0.1       1     75      0.1       5.0      58.6
UDP-TFTP          169    0.0       3     51      0.0      15.3      64.2
UDP-Frag            1    0.0       1   1405      0.0       0.0      86.8
UDP-other       86247    0.1     226     29     24.0      31.4      54.3
ICMP            19989    0.0      37     33      0.9      26.0      53.9
IP-other          193    0.0       1     22      0.0       3.0      78.2
Total:        1014637    1.2      26     99     32.8      13.8      43.9

SrcIf    SrcIPaddress      DstIf    DstIPaddress     Pr    SrcP     DstP
Pkts
Gi0/1    192.168.128.21     Local   192.168.128.20   11    CB2B
07A       3
Gi0/1    192.168.150.60     Gi0/0   10.89.17.146     06    0016     101F
55
Gi0/0    10.89.17.146       Gi0/1   192.168.150.60   06    101F
0016      9
Gi0/1    192.168.150.60     Local   192.168.206.20   01    0000     0303
11
Gi0/0    10.89.17.146       Gi0/1   192.168.150.60   06    07F1
0016      1
```

Netflow for IPv4

To get some practice, we will first turn on Netflow for IPv4. The IPv4 command set uses Cisco's original Netflow configuration. For IPv6 flow information, we can only use Flexible Netflow, and we will try that out in the next section.

Activating Netflow for IPv4

Each Group should turn on Netflow on the border router of their AS. To do this, simply go to the border interface and do something similar to this:

```
interface fastethernet 0/0
 ip flow ingress
 ip flow egress
!
```

Once this has been running for a few minutes, commands like “show ip cache flow” will display output similar to that from the introduction above. To create traffic for Netflow to see, try some ICMPs, traceroutes, and even telnet or ssh to other routers in the lab. This will generate traffic, and the info will persist in Netflow’s cache for a few minutes.

Top talkers in Netflow

Each team should also configure a set of top-talkers, to see what the busiest source and destinations are. Try this configuration:

```
ip flow-top-talkers
 top 20
 sort-by bytes
```

This displays the top 20 talkers, sorting them in descending order of bytes transferred.

Try some of the other CLI options available under the ip flow-top-talkers configurations. There are many match options:

```
gw(config-flow-top-talkers)# match ?
  byte-range        Match a range of bytes
  class-map         Match a class
  destination       Match destination criteria
  direction         Match direction
  flow-sampler      Match a flow sampler
  input-interface   Match input interface
  nexthop-address   Match next hop
  output-interface  Match output interface
  packet-range      Match a range of packets
  protocol          Match protocol
  source            Match source criteria
  tos               Match TOS
```

Try some of these and see what happens to the output.

Netflow for IPv6

Cisco IOS used to support IPv6 with standard Netflow. But this was only briefly the case in IOS 12.3 and 12.4. From 12.4T onwards, IPv6 support in Netflow was replaced by Flexible Netflow for IPv6 (it is also available for IPv4).

Activating Netflow for IPv6

The configuration syntax for Flexible Netflow is somewhat different and a lot more sophisticated. First off we need to create Flow Monitors for our incoming and outgoing Netflow captures. Here is an example

```
flow monitor FLOW-MONITOR-V6-IN
 cache timeout active 300
 record netflow ipv6 original-input
!
flow monitor FLOW-MONITOR-V6-OUT
 cache timeout active 300
 record netflow ipv6 original-output
!
```

And then we apply these flow monitors to the interface we want to monitor:

```
interface FastEthernet0/0
 ipv6 flow monitor FLOW-MONITOR-V6-IN input
 ipv6 flow monitor FLOW-MONITOR-V6-OUT output
!
```

Top talkers in Flexible Netflow

The top talkers in the Flexible Netflow configuration is somewhat different – there is no need to create a specific stanza to set up the top talkers as the router can simply display the top talkers from the command line. Here is an example

```
show flow monitor FLOW-MONITOR-V6-OUT cache aggregate \
ipv6 source address ipv6 destination address sort counter \
bytes top 20
```

This is all one command line and displays the top 20 talkers for outbound traffic, sorting them in descending order of bytes transferred. The command above can be modified to look at the inbound traffic also, by using the inbound flow monitor.

Summary

While this exercise has shown how to set up Netflow for both IPv4 and IPv6, it has a more serious aspect. It is possible for a network operator to very simply see what traffic is traversing their network. If it very easy to spot malicious activity, scanning, etc, simply by looking at the flow data and searching for particular signatures (tcp or udp ports, addresses, etc). This makes Netflow a valuable security tool for all network operators, whether they are running an IPv4-only network, or are dual stack IPv4 and IPv6.

Try some of the other CLI options available under the `show flow monitor` command.