

# Taller de Monitoreo

LACNIC 24 / LACNOC '15,  
Bogotá, 28/9 - 2/10 de 2015

Santiago Aggio

Universidad Tecnológica Nacional Bahía Blanca  
CONICET Bahía Blanca

## Objetivo

**Monitorear nuestro propio tráfico IPv6**

**Utilizando la tecnología Netflow/IPFIX**

## Consideraciones

### Ambiente IPv6-only

El Exportador, el Colector y el Analizador deben conectarse por IPv6

### Medir tráfico IPv6

Los componentes del sistema de monitorización deben soportar NetFlow versión 9.

### Identificar tráfico IPv6

Diferenciar el tráfico IPv6 del IPv4 que atraviesa una interfaz

# SNMP

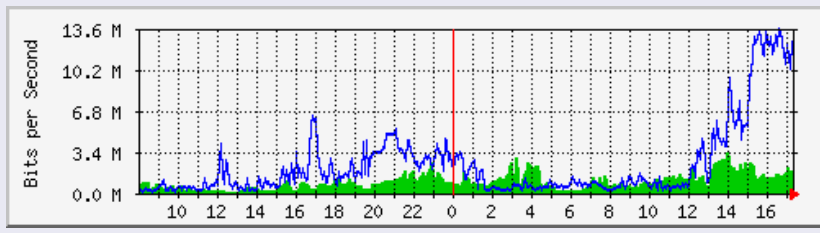
- Los operadores y administradores de red utilizan herramientas que se basan en el protocolo SNMP para obtener información de las interfaces de un dispositivo
- Estos datos son visibles mediante gráficos disponibles en páginas web
- Representan el ancho de banda que atraviesa dicha interfaz en ambos sentidos (in/out)
- Esta información es muy útil para la toma de decisiones que hacen al funcionamiento y la planificación a futuro, al observar por ejemplo la saturación de la capacidad de un enlace en diferentes momentos del día

# SNMP

## Herramientas basadas en consultas SNMP

- MRTG
- Cacti
- Zabbix
- Cricket
- Pandora

## Imagen generada

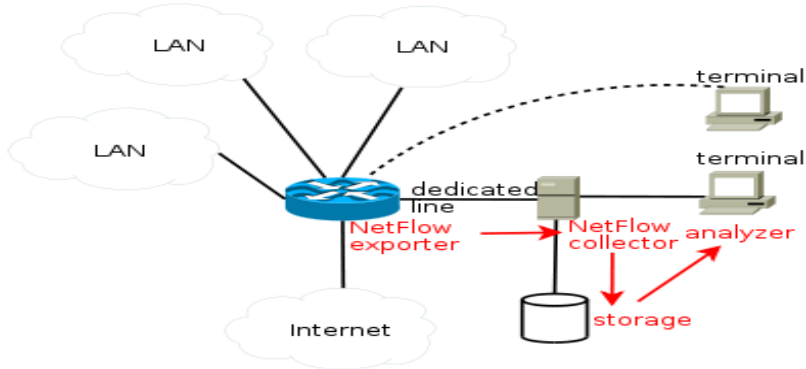


## SNMP... no alcanza

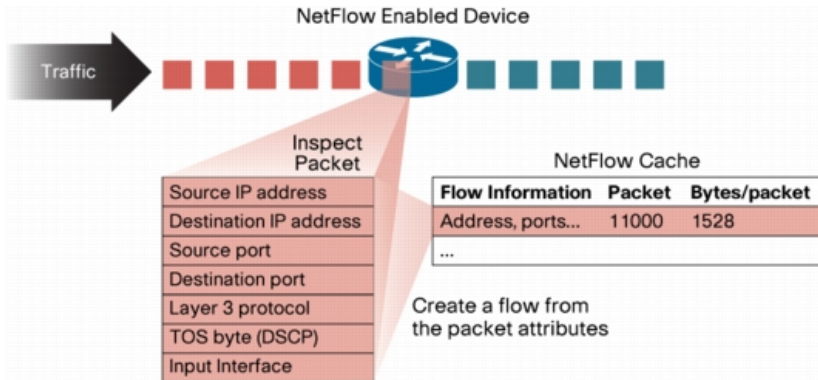
- SNMP es la forma tradicional de monitorear el ancho de banda
- Un conocimiento más detallado de cómo se está utilizando el ancho de banda es muy importante hoy en las redes IP
- Contadores de paquetes y bytes de interfaz son útiles pero.....

**.... conocer que direcciones IP son el origen y destino del tráfico, los protocolos que atraviesan los enlaces y que aplicaciones están generando el tráfico es muy valiosa**

## Arquitectura de monitoreo NetFlow



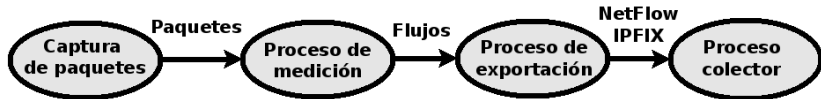
# NetFlow en Cisco



Fuente: <http://www.cisco.com>



## Procesos en la Arquitectura NetFlow/IPFIX



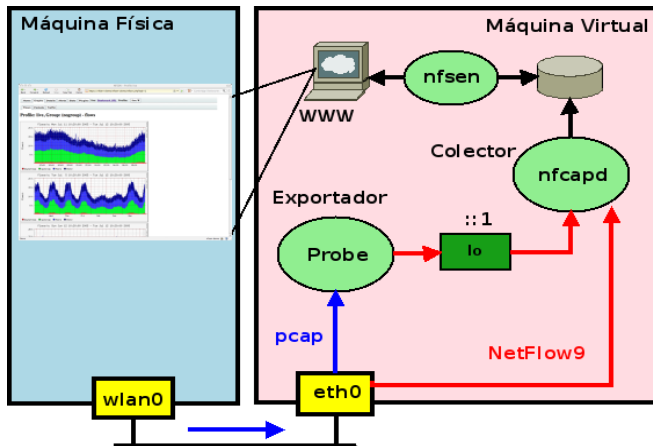
## Vagrant para crear MV

Repositorio GitHub con material para crear MV con Vagrant

<https://github.com/LACNIC/tutorial-netmon/tree/master/labs/lab-netflow-nfsen>

<https://github.com/sancolo/lab-netflow-nfsen.git>

## Escenario Taller: MV es el Router de MF



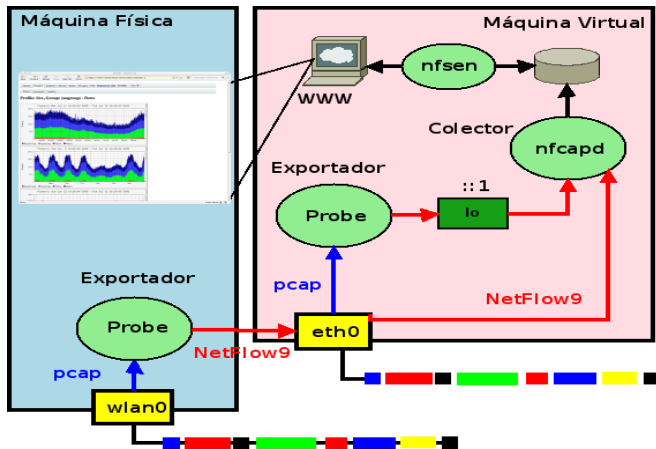
## Escenario Taller: MV es el Router de MF

- La MV actúa como router de la MF
- En la MF identificamos la dirección IPv4 del default gateway  
*ip route show | grep ^default*  
*route | grep UG*  
*netstat -nr | grep UG*
- En la MF borramos la ruta default gateway  
*ip route delete default via IPv4*  
*route delete default gw IPv4*
- Identificamos la IPv4\_MV y la asignamos en la MF como default GW  
*ip route add default via <IPv4\_MV>*  
*route add default gw <IPv4\_MV>*

## Escenario Taller: MV es el Router de MF en IPv6

- *ip -6 route show | grep ^default  
route [-A inet6 | -6] | grep UG  
netstat -6 -nr | grep UG*
- *ip -6 route del default via <ip6address>  
route -6 del default gw <ip6address>*
- *ip -6 route add default via <ipv6address>  
route -6 add default gw <ip6address>*

## Escenario Taller: MF + MV



# VirtualBox

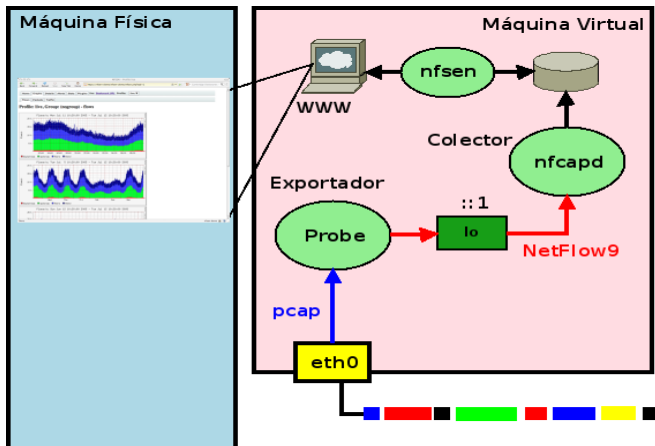
## 6.4. Bridged networking

Depending on your host operating system, the following limitations should be kept in mind:

- On Macintosh hosts, functionality is limited when using AirPort (the Mac's wireless networking) for bridged networking. Currently, VirtualBox supports only IPv4 over AirPort. For other protocols such as IPv6 and IPX, you must choose a wired interface.
- On Linux hosts, functionality is limited when using wireless interfaces for bridged networking. Currently, VirtualBox supports only IPv4 over wireless. For other protocols such as IPv6 and IPX, you must choose a wired interface.

[http://www.virtualbox.org/manual/ch06.html#network\\_bridged](http://www.virtualbox.org/manual/ch06.html#network_bridged)

## Escenario Taller: MV





## Software en la MV

- **Probe:** softflowd
- **Colector:** nfcapd
- **Analizador:** nfdump (modo texto)
- **Monitor:** nfsen (modo gráfico, acceso por página web)
- **Web server:** apache
- **Otros:** mtr, tcpdump, tshark, wget

# Probe

- Se basan en la librería pcap (<http://www.tcpdump.org>)
- Capturan tráfico sobre una interfaz en modo promiscuo (tcpdump, wireshark, tshark)
- Generan paquetes NetFlow/IPFIX que exportan a un colector

## Probe: paquetes disponibles para Unix

- **ipt-netflow:** módulo de Kernel basado en iptables, no soporta IPv6
- **fprobe:** basado en libpcap, no soporta IPv6
- **fprobe-ulong:** basado en libipulog, usado con iptables ULOG target, no soporta IPv6
- **pmacct:** utilizado en IXPs, Data Centers, IP Carriers, CDNs
- **nProbe:** aplicación del proyecto Ntop
- **softflowd:** simple, soporta IPv6

## Ejercicio 1: MV

- Verificar en VirtualBox que la red para la MV está configurada en modo bridge
- Iniciar la MV en VirtualBox
- Verificar la dirección IPv6 de la MV  
*ifconfig eth0*
- Verificar que softflowd está corriendo sobre la MV, exportando sobre la dirección ::1 y el port 9995  
*ps ax | grep softflowd*
- Verificar que softflowd abrió un socket en la dirección IPv6  
*lsof -i -n | grep 999*

## Ejercicio 1: MF

### Softflowd instalado en la MF (ver Requerimientos.pdf)

- Iniciar softflowd para que exporte paquetes a la IPv6 del colector sobre el puerto 9996  
*softflowd -i wlan0 -n IPv6\_MV:9996 -v 9 -6*
- Verificar que softflowd esta corriendo sobre la MF, exportando sobre la dirección IPv6\_MV y el port 9996  
*ps ax | grep softflowd*
- Verificar que softflowd abrió un socket en la dirección IPv6  
*lsnf -i -n | grep 999*

# Flujo

## 5 Atributos que identifican un Flujo

- Dirección Fuente
- Dirección Destino
- Puerto Fuente
- Puerto Destino
- Protocolo de transporte

## Cisco Agrega

- Byte de TOS (DSCP)
- Interface de entrada

## Flujo Unidireccional

- Coincidencia de los 5/7 atributos → actualizar flujo
- Diferencia de 1 atributo → nuevo flujo

## ¿Cuándo un flujo es exportado?

- El flujo es terminado  
Conexión TCP termina debido a un FIN o RST
- El flujo permanece ocioso por un período de tiempo (timeout)  
Cisco establece 15 seg
- El flujo alcanza un máximo tiempo de vida permitido (active timeout)  
Lo valores varían. Cisco establece 1800 seg. ¿Y Softflowd?
- Se fuerza el descarte del flujo  
La cache esta llena y un nuevo flujo debe ser alojado

# Softflowd

```
sudo /usr/local/sbin/softflowctl help
```

Valid control words are:

```
debug+ debug- delete-all dump-flows exit  
expire-all shutdown start-gather statistics  
stop-gather timeouts send-template
```



## Ejercicio 2: softflowd

- Generar tráfico sobre la MF o MV
- Verificar los flujos activos  
*softflowctl dump-flows*
- Ver los tiempos de expiración  
*softflowctl timeouts*
- Ver la estadística de flujos activos y exportados  
*softflowctl statistics*

## Paquete de exportación NetFlow 9

Packet Header
Template FlowSet
Data FlowSet
Data FlowSet
. . . .
Template FlowSet
Data FlowSet
. . . .

## Header de NetFlow 9

bit 0-7	bit 8-15	bit 16-23	bit 24-31
Version Number		Count	
sysUpTime			
UNIX Secs			
Sequence Number			
Source ID			

## Captura de paquetes NetFlow 9

```
Version: 9
Count: 12
SysUptime: 263802007
Timestamp: Sep 17, 2014 15:46:01.000000000 EDT
    CurrentSecs: 1379447161
FlowSequence: 23995
SourceId: 0
FlowSet 1
    FlowSet Id: (Data) (1024)
    FlowSet Length: 472
    Data (468 bytes), no template found
```

## Template FlowSet

bit 0-15	bit 16-31
FlowSet ID = 0	Length
Template ID	Field Count
Field 1 Type	Field 1 Length
Field 2 Type	Field 2 Length
...	...
Field N Type	Field N Length
Template ID	Field Count
Field 1 Type	Field 1 Length
Field 2 Type	Field 2 Length
...	...
Field N Type	Field N Length

# Templates

- Expiran si no son refrescados periódicamente
- Se prevén dos formas de refresco del template:
  - El template puede ser reenviado cada N números de paquetes exportados
  - El template puede ser refrescado cada N minutos (timer)

## Template FlowSet

Tipo de Campo	Valor	Long	Descripción
IPV6_SRC_ADDR	27	16	IPv6 Source Address
IPV6_DST_ADDR	28	16	IPv6 Destination Address
IPV6_SRC_MASK	29	1	Length of the IPv6 source mask in contiguous bits
IPV6_DST_MASK	30	1	Length of the IPv6 destination mask in contiguous bits
IPV6_FLOW_LABEL	31	3	IPv6 flow label as per RFC 2460 definition

<http://www.iana.org/assignments/ipfix>

## Template Flowset

Tipo de Campo	V	L	Descripción
SAMPLING_INTERVAL	34	4	The rate at which packets are sampled. A value of 100 indicates that one of every 100 packets is sampled
SAMPLING_ALGORITHM	35	1	The type of algorithm used for sampled NetFlow: 0x01 Deterministic Sampling ,0x02 Random Sampling
FLOW_ACTIVE_TIMEOUT	36	2	Timeout value (in seconds) for active flow entries in the NetFlow cache
FLOW_INACTIVE_TIMEOUT	37	2	Timeout value (in seconds) for inactive flow entries in the NetFlow cache



# Captura de paquetes Template FlowSet

## FlowSet 1

FlowSet Id: Data Template (V9) (0)

FlowSet Length: 60

Template (Id = 1024, Count = 13)

Template Id: 1024

Field Count: 13

Field (1/13): IP\_SRC\_ADDR | Type: IP\_SRC\_ADDR (8) | Length: 4

Field (2/13): IP\_DST\_ADDR | Type: IP\_DST\_ADDR (12) | Length: 4

Field (3/13): LAST\_SWITCHED | Type: LAST\_SWITCHED (21) | Length: 4

Field (4/13): FIRST\_SWITCHED | Type: FIRST\_SWITCHED (22) | Length: 4

Field (5/13): BYTES | Type: BYTES (1) | Length: 4

Field (6/13): PKTS | Type: PKTS (2) | Length: 4

Field (7/13): INPUT\_SNMP | Type: INPUT\_SNMP (10) | Length: 4

Field (8/13): OUTPUT\_SNMP | Type: OUTPUT\_SNMP (14) | Length: 4

Field (9/13): L4\_SRC\_PORT | Type: L4\_SRC\_PORT (7) | Length: 2

Field (10/13): L4\_DST\_PORT | Type: L4\_DST\_PORT (11) | Length: 2

Field (11/13): PROTOCOL | Type: PROTOCOL (4) | Length: 1

Field (12/13): TCP\_FLAGS | Type: TCP\_FLAGS (6) | Length: 1

Field (13/13): IP\_PROTOCOL\_VERSION | Type: IP\_PROTOCOL\_VERSION (60) | L

## Captura de paquetes Template Flowset IPv6

### FlowSet 2

FlowSet Id: Data Template (V9) (0)

FlowSet Length: 60

Template (Id = 2048, Count = 13)

Template Id: 2048

Field Count: 13

Field (1/13): IPV6\_SRC\_ADDR | Type: IPV6\_SRC\_ADDR (27) | Length: 16

Field (2/13): IPV6\_DST\_ADDR | Type: IPV6\_DST\_ADDR (28) | Length: 16

Field (3/13): LAST\_SWITCHED | Type: LAST\_SWITCHED (21) | Length: 4

Field (4/13): FIRST\_SWITCHED | Type: FIRST\_SWITCHED (22) | Length: 4

Field (5/13): BYTES | Type: BYTES (1) | Length: 4

Field (6/13): PKTS | Type: PKTS (2) | Length: 4

Field (7/13): INPUT\_SNMP | Type: INPUT\_SNMP (10) | Length: 4

Field (8/13): OUTPUT\_SNMP | Type: OUTPUT\_SNMP (14) | Length: 4

Field (9/13): L4\_SRC\_PORT | Type: L4\_SRC\_PORT (7) | Length: 2

Field (10/13): L4\_DST\_PORT | Type: L4\_DST\_PORT (11) | Length: 2

Field (11/13): PROTOCOL | Type: PROTOCOL (4) | Length: 1

Field (12/13): TCP\_FLAGS | Type: TCP\_FLAGS (6) | Length: 1

Field (13/13): IP\_PROTOCOL\_VERSION | Type: IP\_PROTOCOL\_VERSION (60) | L

## Data FlowSet

bit 0-15
flowset_id = template_id (>255)
length
record_1-field_1_value
record_1-field_2_value
...
record_1-field_M_value
record_2-field_1_value
record_2-field_2_value
...
record_2-field_M_value
...
record_N-field_M_value
padding

## Captura de paquetes Data FlowSet

### FlowSet 3

FlowSet Id: (Data) (1024)

FlowSet Length: 316

#### Flow 1

- (1) SrcAddr: 192.168.1.103 (192.168.1.103)
- (2) DstAddr: 192.168.13.109 (192.168.13.109)  
[Duration: 29.6640000000 seconds]
- (3) StartTime: 263892.5370000000 seconds
- (4) EndTime: 263922.2010000000 seconds
- (5) Octets: 998
- (6) Packets: 6
- (7) InputInt: 0
- (8) OutputInt: 0
- (9) SrcPort: 55073
- (10) DstPort: 80
- (11) Protocol: 6
- (12) TCP Flags: 0x1b
- (13) IPVersion: 04

## Captura de paquetes Data FlowSet

### FlowSet 1

FlowSet Id: (Data) (2048)

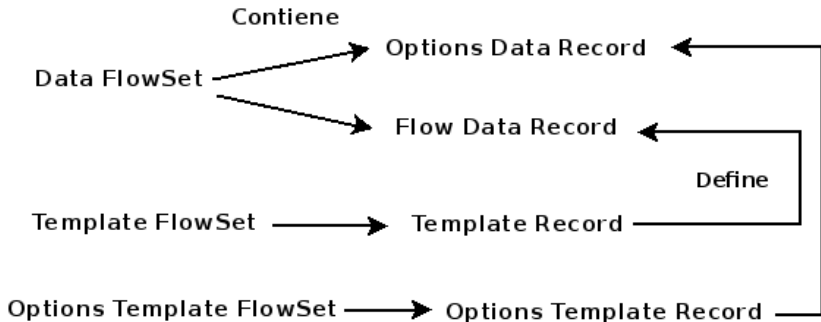
FlowSet Length: 132

. . . . .

### Flow 2

- (1) SrcAddr: 2001:db8:90:192::30 (2001:db8:90:192::30)
- (2) DstAddr: 2001:db8:90:192::16 (2001:db8:90:192::16)  
[Duration: 1.2990000000 seconds]
- (3) StartTime: 1204388.3360000000 seconds
- (4) EndTime: 1204389.6350000000 seconds
- (5) Octets: 2484
- (6) Packets: 21
- (7) InputInt: 0
- (8) OutputInt: 0
- (9) SrcPort: 35849
- (10) DstPort: 995
- (11) Protocol: 6
- (12) TCP Flags: 0x1b
- (13) IPVersion: 06

## Options Template Flowset y Options Data Record



Fuente: RFC7011 (09/13) / RFC5101

## Ejercicio 3: tshark

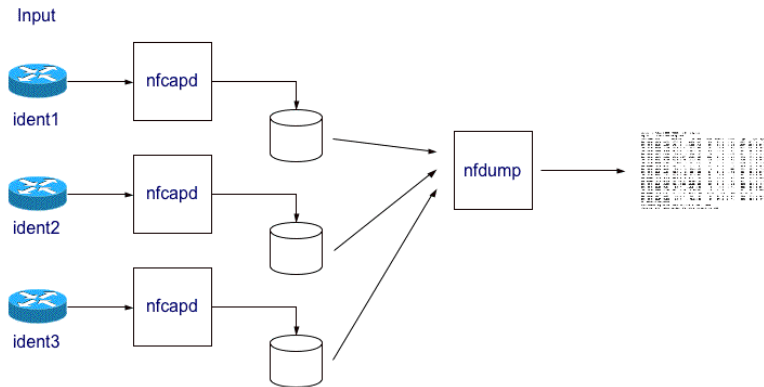
- Ejecutar tshark en una consola sobre la MV  
*tshark -ni eth0 -d udp.port==9996,cflow -f 'udp dst port 9996' -V*
- Generar tráfico sobre la MF o MV
- Volver a la consola para ver los paquetes NetFlow capturados con tshark
- Forzar el envío del template desde softflowctl y ver la captura

# Nfdump

- Colecta los paquetes NetFlow y los almacena en archivos generados en intervalos de tiempo (5 minutos)
- Filtrado basado en la sintaxis de la librería PCAP
- Rápido en procesar, Eficiente en el uso de la CPU, Flexible en la agregación de flujos.

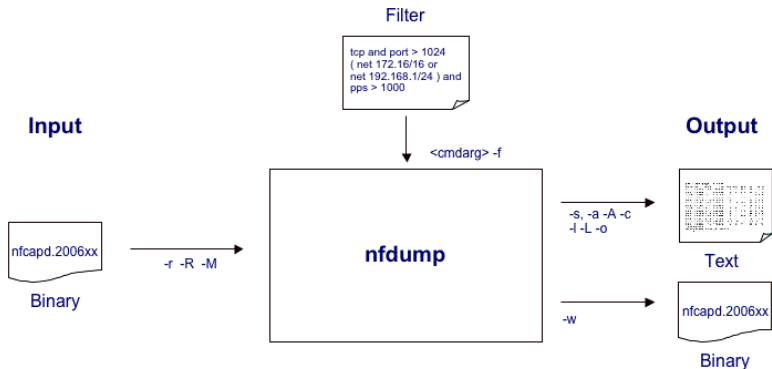


## Arquitectura de Nfdump



Fuente: <http://nfdump.sourceforge.net/>

## Análisis de información colectada



Fuente: <http://nfdump.sourceforge.net/>

## Componentes de nfdump

- nfcapd - netflow capture daemon
- nfdump - netflow dump
- nfprofile - netflow profiler (run by nfsen)
- nfreplay - netflow replay
- nfclean.pl - cleanup old data
- nfexpire - data expiry program (maxtime, maxsize, watermark)  
(nfcapd -e)
- ft2nfdump - Read and convert flow-tools data

# Nfsen

- Interfaz web para graficar y procesar los datos colectados
- Utiliza nfdump a bajo nivel para obtener la información estadística requerida
- Presenta gráficos de Flujos, Paquetes y Tráfico, diferenciando los protocolos TCP, UDP, ICMP y otros.
- Permite el análisis sobre ventanas de tiempo
- Alertas definidas en base a condiciones que determinan comportamientos anómalos del tráfico y los flujos activos
- Definición de Perfiles para seguimientos de subredes, máquinas, puertos, servicios, etc.
- Extensiones basadas en Plugins (Mod.Pperl y PHP)

## Implementación de Nfsen en la MV

- Directorio de instalación: **/data/nfsen**
- Archivo de configuración: **/data/nfsen/etc/nfsen.conf**
- Fuentes que generan paquetes NetFlow a coleccionar:

```
%sources = (  
    'mv' => { 'port' => '9995', 'col' => '#0000ff', 'type' => 'netflow' },  
    'mf' => { 'port' => '9996', 'col' => '#00ff00', 'type' => 'netflow' },  
);
```

## nfcapd

```
nfcapd -6 -w -D -p 9995 -u netflow -g www-data -B 200000 -S 1  
-P /data/nfsen/var/run/p9995.pid -z -l mv -l  
/data/nfsen/profiles-data/live/mv
```

### Opciones

- 6 listen on IPv6 only
- w Align file rotation
- D daemon mode
- p port
- u usuario
- g group
- B buflen
- l base\_directory
- S 1 %Y/ %m/ %d
- P pidfile
- z Compress flows

## Ejercicio 4: Nfsen

- Verificar los procesos nfcapd

```
ps ax | grep nfcapd
```

```
3278 ?          S          0:00 /usr/bin/nfcapd -6 -w -D -p 9995 -u netflow  
-g www-data -B 2000000 -S 1 -P /data/nfsen/var/run/p9995.pid -z -I mv  
-l /data/nfsen/profiles-data/live/mv
```

```
3284 ?          S          0:00 /usr/bin/nfcapd -6 -w -D -p 9996 -u netflow  
-g www-data -B 2000000 -S 1 -P /data/nfsen/var/run/p9996.pid -z -I mf  
-l /data/nfsen/profiles-data/live/mf
```

- Ver el tráfico colectado mediante nfsen ingresando a [http://\[ipv6\\_mv\]/nfsen/nfsen.php](http://[ipv6_mv]/nfsen/nfsen.php)



## Ejercicio 5: nfdump

- Verificar que los paquetes NetFlow son colectados y almacenados para cada fuente
- Identificar flujos IPv6 colectados aplicando filtros  
*nfdump -M /data/nfsen/profiles-data/live/mf/2014/10/27 -R . 'ipv6' -o long6*
- Aplicar filtros específicos para ver diferentes estadísticas  
*nfdump -M /data/nfsen/profiles-data/live/mf/2014/10/27 -R . -l -n 10 -s ip/bytes*

Referencia: <http://nfdump.sourceforge.net>



# Nfsen Profile

<b>Profile:</b>	<input type="text"/>	
<b>Group:</b>	(nogroup) 	
<b>Description:</b>	<input type="text"/>	
<b>Start:</b>	<input type="text"/> <b>Format: yyyy-mm-dd-HH-MM</b>	
<b>End:</b>	<input type="text"/> <b>Format: yyyy-mm-dd-HH-MM</b>	
<b>Max. Size:</b>	<input type="text" value="10G"/>	
<b>Expire:</b>	<input type="text" value="60 Days"/>	
<b>Channels:</b>	<input checked="" type="radio"/> <b>1:1 channels from profile live</b> <input type="radio"/> individual channels	
<b>Type:</b>	<input checked="" type="radio"/> <b>Real Profile</b> <input type="radio"/> Shadow Profile	
<b>Sources:</b>	<input type="text" value="mv"/> <input type="text" value="mf"/>	
<b>Filter:</b>	<input type="text"/>	
<input type="button" value="Cancel"/> <input type="button" value="Create Profile"/>		

## Ejercicio 6: Nfsen Profile para IPv6

- Obtener la dirección IPv6 y el prefijo de la red
- Identificar tráfico IPv6 entrante y saliente mediante 2 canales diferentes
- Crear el filtro a aplicar en el Profile para cada canal  
*inet6 and dst net ipv6/prefix*  
*inet6 and src net ipv6/prefix*

## Nfsen Plugins

- Extienden la funcionalidad de Nfsen
- Plugin tiene dos componentes: backend y frontend

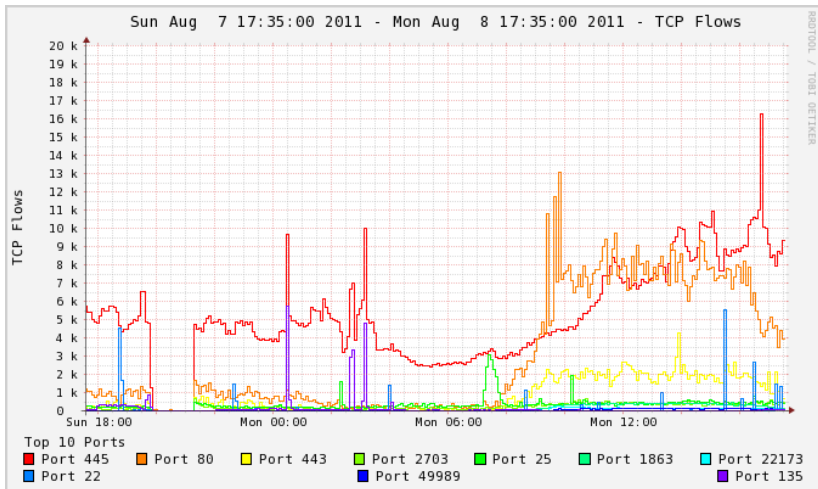
### Backend

- Nfsen procesa periódicamente el backend asociado
- Escritos en Perl

### Frontend

- Grafica los resultados del proceso backend asociado
- Escritos en PHP

# Nfsen Plugin: PortTracker



## Ejercicio 7: Nfsen PortTracker Plugin

Instalar el plugin PortTracker en la MV

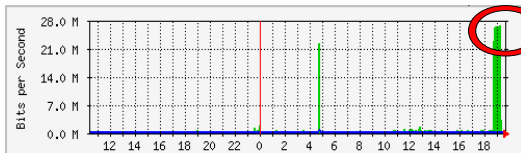
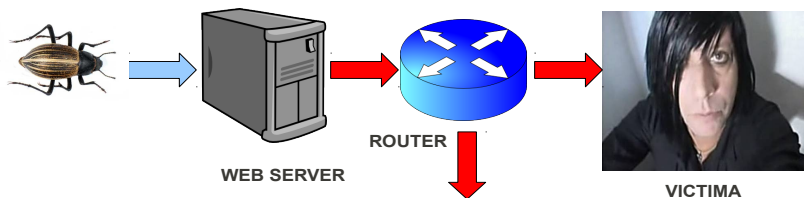
Referencia:

<http://sourceforge.net/apps/trac/nfsen-plugins/wiki/PortTracker>

Plugins disponibles para Nfsen

<http://sourceforge.net/apps/trac/nfsen-plugins/>

# UDP flood



ADMINISTRADOR

## UDP flood

- La red esta lenta, se cayo un enlace ?
- Mucho download o algún P2P
- Generalizemos ..... No anda Internet !!!!

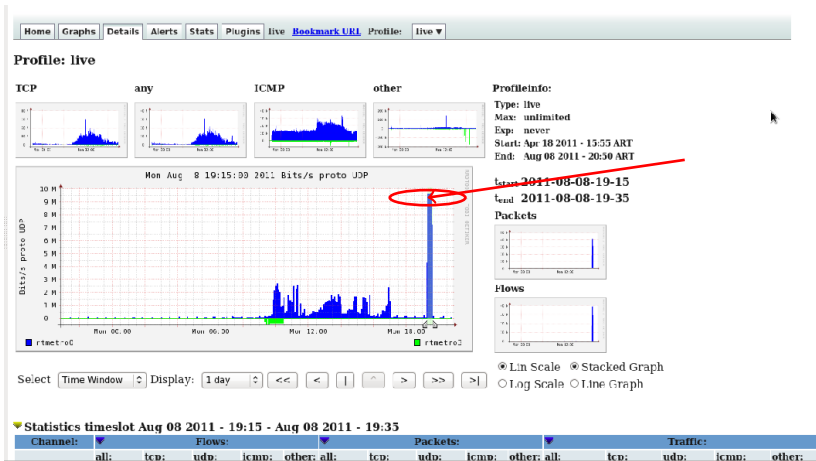
### Como verifico un comportamiento anómalo, si....

- Mi browser no responde !!!
- ¿Se cayo el enlace o ... es el DNS que no resuelve?
- Ping, traceroute, mtr, dig, hosts

**Empiezan a sonar los teléfonos  
y .....  
no es para invitarte a una  
fiesta!!!!**



# UDP flood





# UDP flood

### Netflow Processing

Source:   
  
 Filter:   
 Options:  List Flows  Stat Top N  
 Top:   
 Stat:  order by   
 Limit:  Packets   
 Output:  / IPv6 long  
 Clear Form process

```

** nfdump -M /var/nfsen/profiles-data/live/rtmetro0:rtmetro3 -T -R 2011/06/08/nfcpad.201106061915:2011/06/08/nfcpad.201106081935 -n 10 -s dstip/flows
nfcump filter:
proto UDP
Top 10 dst IP Addr ordered by flows:

```

Time	Secs	Source IP	Port	Dest IP	Count	Packets	Bytes	pps	bps	app
2011-08-08	18:57:22.775	1252.298	any	150.128.229.104	48.5 M(99.8)	51.8 M(99.8)	1.5 G(99.1)	41345	9.6 M	29
2011-08-08	18:49:42.752	1020.094	any	150.128.229.104	15736( 0.0)	24731( 0.0)	1.7 M( 0.1)	13	825	07
2011-08-08	18:54:18.443	1533.128	any	150.128.229.242	8862( 0.0)	8804( 0.0)	2.0 M( 0.1)	5	10649	231
2011-08-08	18:54:16.567	1544.616	any	1.0.1.4.192.2	7620( 0.0)	7007( 0.0)	1.3 M( 0.1)	4	6059	107
2011-08-08	18:54:16.939	1524.196	any	1.0.1.1.156.165	5467( 0.0)	5756( 0.0)	1.0 M( 0.1)	3	3866	177
2011-08-08	18:54:16.859	1544.490	any	150.128.208.2	2545( 0.0)	2618( 0.0)	385933( 0.6)	1	1969	147
2011-08-08	18:54:17.963	1540.132	any	150.128.204.2	2525( 0.0)	2477( 0.0)	299010( 0.6)	1	1566	117
2011-08-08	18:54:17.179	1543.428	any	150.128.128.2	2177( 0.0)	2331( 0.0)	263880( 0.6)	1	1367	113
2011-08-08	18:54:18.427	1497.176	any	150.128.202.2	1879( 0.0)	1879( 0.0)	298858( 0.6)	1	1566	159
2011-08-08	18:53:03.675	1355.544	any	150.128.112.249	758( 0.0)	1273( 0.0)	91644( 0.6)	0	540	71

Summary: total flows: 48651443, total bytes: 1.5 G, total packets: 51.9 M, avg bps: 6.6 M, avg pps: 28405, avg tpp: 29  
 Time window: 2011-08-08 18:49:34 - 2011-08-08 19:20:01  
 Total flows processed: 48943560, blocks skipped: 0, bytes read: 2545864500  
 Sys: 5.528s flows/second: 8853292.9 wall: 8.396s flows/second: 5828870.2

alfans 1.3.3

← Previous → Next Highlight all Match case Reached end of page, continued from top

6 paused downloads 150.124.208.21

## UDP flood

```
** nfdump -M /var/nfsen/profiles-data/live/rtmetro0:rtmetro3 -T -R
  2011/08/08/nfcapd.201108081915:2011/08/08/nfcapd.201108081935 -n 10 -s
  dstip/flows
```

nfdump filter:

proto UDP

Top 10 Dst IP Addr ordered by flows:

Date first seen	Duration	Proto	Dst IP Addr	Flows
(%)	Packets(%)	Bytes(%)	pps	bps
2011-08-08 18:57:22.775	1252.208	any	192.168.229.104	48.5 M
(99.8)	51.8 M(99.8)	1.5 G(99.1)	41345	9.6 M
2011-08-08 18:49:42.791	1618.604	any	192.168.198.68	19758
(0.0)	24745(0.0)	1.7 M(0.1)	15	8294
2011-08-08 18:54:18.443	1533.128	any	192.168.130.242	8802
(0.0)	8804(0.0)	2.0 M(0.1)	5	10649

**Summary:** total flows: 48661443, total bytes: 1.5 G, total packets: 51.9 M,  
 avg bps: 6.6 M, avg pps: 28405, avg bpp: 29

**Time window:** 2011-08-08 18:49:34 - 2011-08-08 19:20:01

**Total flows processed:** 48943560, Blocks skipped: 0, Bytes read: 2545094500

**Sys:** 5.528s flows/second: 8853202.9 Wall: 8.396s flows/second: 5828870.2 57/69

# UDP flood

## NetFlow Processing

Source:

Filter:

Options:

List Flows  Stat TopN

Top:

Stat:  order by

Limit:  Packets  Bytes

Output:  / IPv6 long

```
** nfcupd -M /var/nfsen/profiles-data/live/rtmetro0:rtmetro0 -T -R 2011/00/00/nfcupd.201100001935:2011/00/00/nfcupd.201100001935 -n 10 -s ip/flows
nfdump filter:
proto UDP
```

Top 10 IP Addr ordered by flows:

Date first seen	Duration	Proto	TP Addr	Flows (%)	Packets (%)	Bytes (%)	pps	bps	lpp
2011-00-00 18:57:22.775	1252.208	any	193.173.229.104	48.5 M(99.8)	51.8 M(99.8)	1.5 G(99.1)	41345	9.6 M	29
2011-00-00 18:57:22.775	1252.208	any	193.173.204.37	48.5 M(99.8)	51.8 M(99.8)	1.5 G(99.1)	41345	9.6 M	29
2011-00-00 18:49:54.515	1626.620	any	193.173.196.165	39531( 0.1)	49331( 0.1)	3.4 M( 0.2)	30	10619	68
2011-00-00 18:54:16.211	1533.360	any	193.173.130.242	24362( 0.0)	24307( 0.0)	3.2 M( 0.2)	15	16860	133
2011-00-00 18:54:16.827	1544.756	any	193.173.134.182.2	16785( 0.0)	16909( 0.0)	2.4 M( 0.2)	10	12395	141
2011-00-00 18:54:16.939	1543.694	any	193.173.196.165	8386( 0.0)	8917( 0.0)	1.3 M( 0.1)	5	6849	148
2011-00-00 18:54:16.859	1544.400	any	193.173.134.200.2	5516( 0.0)	5639( 0.0)	897371( 0.1)	3	4648	159
2011-00-00 18:54:17.943	1540.132	any	193.173.134.200.2	4962( 0.0)	5478( 0.0)	849539( 0.1)	3	4412	155
2011-00-00 18:54:16.835	1543.624	any	193.173.134.200.2	4949( 0.0)	4949( 0.0)	540335( 0.0)	3	2891	169
2011-00-00 18:49:57.395	1803.212	any	193.173.128.2	4535( 0.0)	5254( 0.0)	810168( 0.1)	2	3594	154

Summary: total flows: 48661463, total bytes: 1.5 G, total packets: 51.9 M, avg bps: 5.6 M, avg pps: 28/05, avg lpp: 29  
 Time window: 2011-00-00 18:49:34 - 2011-00-00 19:20:01  
 Total flows processed: 48943560, Blocks skipped: 0, Bytes read: 2545094599  
 Sys: 6.392s flows/second: 7656524.6 Wall: 9.293s flows/second: 5266173.7

## UDP flood

```
# netstat -alunp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         PID/Program name
udp      0      0 0.0.0.0:38447           0.0.0.0:*               1897/avahi-daemon:
udp      0      0 0.0.0.0:5353           0.0.0.0:*               1897/avahi-daemon:
udp      0      0 0.0.0.0:746           0.0.0.0:*               1418/rpc.statd
udp      0      0 0.0.0.0:749           0.0.0.0:*               1418/rpc.statd
udp      0      0 0.0.0.0:111           0.0.0.0:*               1382/portmap
udp      0      0 0.0.0.0:51188         0.0.0.0:*               12237/perl
udp      0      0 0.0.0.0:631           0.0.0.0:*               1646/cupsd
udp      0      0 :::5353                :::*                    1897/avahi-daemon:
udp      0      0 :::47860                :::*                    1897/avahi-daemon:
```

```
# ps aux | grep perl
apache 12237 95.1 0.2 25356 2424 ? R 04:27 23:20 perl /tmp/U
192.168.229.104 0 0
```

## Detección de anomalías

- La inspección de cada paquete no siempre es viable en redes de alta velocidad
- Detecciones basadas en flujos IP es un complemento y una primera aproximación para detectar ataques

### Detección de Intrusos analizando Flujos IP

- Denial of Service
- Scans
- SPAM
- Botnets
- Worms

## Ejemplo: DNS & Feederbot

El canal C&C de una Botnet puede utilizar el puerto 53

- 1 Consultas de DNS a servidores propios, **es habitual**
- 2 Consultas de DNS a servidores públicos, **es probable**
- 3 Alto número de consultas a servidores públicos, **es raro**
- 4 Alto número de consultas de dominios de dudosa denominación, **estamos en problemas**
- 5 Incremento en las consultas DNS sobre TCP respecto de UDP, **seguimos en problemas**

**Este tráfico representa un porcentaje ínfimo del total y podremos inspeccionar, sin un alto costo, el payload del paquete usando futuras extensiones de IPFIX**

## Ejemplo: DNS & Feederbot

- Podemos crear un profile para ver consultas a otros DNS
- Filtro del profile:  
*dst port 53 and not (host ipv4\_dns1 or host ipv4\_dns2 or host ipv6\_dns1 or host ipv6\_dns2)*
- Diferenciamos TCP de UDP  
*proto tcp and dst port 53 and not (host pv4\_dns1 or host ipv4\_dns2 or host ipv6\_dns1 or host ipv6\_dns2)*

## Desempeño en redes de alta velocidad

### Sampling

- Determinístico: 1-de-N
- Random: n-de-N

### Consecuencia

- ↓ **Perdemos información !!!!**
- ↑ Menor uso de la CPU

### Agregación de flujos

- Disminuye el tamaño de memoria cache
- Disminuye el tráfico de paquetes NetFlow

### Colector

- Disminuye el número de paquetes a coleccionar
- Menor procesamiento para análisis de ventanas de tiempo



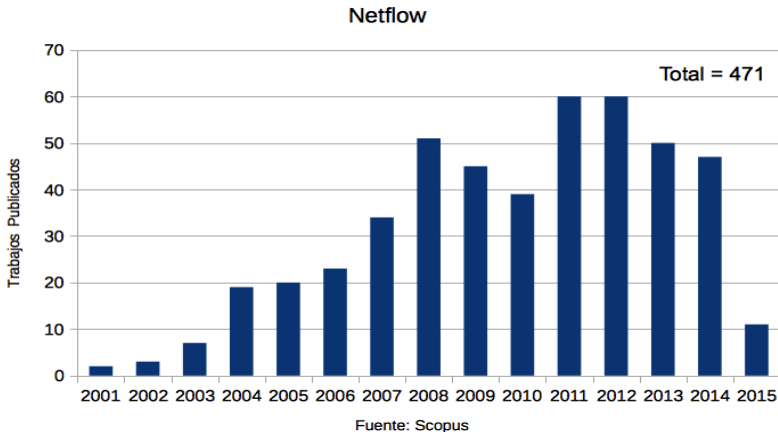
## Requerimiento de Almacenamiento

Valores Promedio					
AB	5 minutos	Diario	Semanal	Mensual	Anual
10 Mbps	500 KB	150 MB	1 GB	4 GB	50 GB
100 Mbps	5 MB	1.5 GB	10 GB	40 GB	500 GB
1 Gbps	50 MB	15 GB	100 GB	400 GB	5 TB
2 Gbps	100 MB	30 GB	200 GB	800 GB	10 TB
10 Gbps	500 MB	150 GB	1 TB	4TB	50 TB

Introducción  
Probe  
NetFlow 9  
Nfdump  
Nfsen  
Detección de anomalías  
Conclusión

UDP flood  
Detección de anomalías  
Detección de Intrusos analizando Flujos IP  
Desempeño en redes de alta velocidad  
Almacenamiento  
Producción Científica  
Tecnologías

## Producción Científica



## Tecnologías de mejor desempeño

### Sonda

TAP → Pasivo, no compromete al router

### Exportador

Hardware dedicado → FPGA (10Gbps)

### Colector

#### High Performance Computing (HPC)

- GPU → Indexado de flujos

#### Big Data

- Hadoop → Hadoop Distributed File System (HDFS)
- MapReduce → Task and Jobs

## Conclusión

Un sistema de monitoreo basado en NetFlow/IPFIX permite:

- Mejorar la visibilidad de la red en su conjunto
- Mayor granularidad en el análisis del tráfico IP
- Facilitar la gestión y la adopción de nuevas políticas y tecnologías
- **Observar el desempeño y calidad de la red**
- **Diagnosticar en menor tiempo diferentes tipos de anomalías en el tráfico**
- **Verificar el buen uso y la seguridad de la red**

# FIN

¿ Preguntas ?

Muchas gracias!!!

slaggio@criba.edu.ar

**Agradecimientos**

**LACNIC / LACNOG**