





Parco Ex Caserma Cocco Pescara

NtopNg e il monitoraggio del traffico di rete (in high-speed network)

Giuseppe Augiero

<talk@augiero.it> - @GiuseppeAugiero

Luca Deri

<deri@ntop.org> - @lucaderi

(c) Ntop.org - 19 agosto 2016 - Moca 2016 - Pescara - Parco ex Caserma Cocco

Outlook

- What are the main activities of ntop.org ?
- Ntop's view on network monitoring.
- From ntop to ntopng.
- Ntopng architecture and design.
- Ntopng as a flow collector
- Exploring system activities using ntopng
- Using ntopng.
- Advanced monitoring with ntopng.
- Future roadmap items.



About ntop.org [1/2]

- Ntop develops of open source network traffic monitoring applications.
- Ntop (circa 1998) is the first app we released and it is a web-based network monitoring application.
- Today our products range from traffic monitoring, highspeed packet processing, deep-packet inspection, and IDS/IPS acceleration (snort and suricata).



About ntop.org [2/2]

Our software is powering many commercial products...



Integrated ASIC with JDSU technology







Parco Ex Caserma Cocco Pescara .

-

Ntop Goals

- Provide better, yet price effective, traffic monitoring solution by enabling users to have increased traffic visibility.
- Go beyond standard metrics and increase traffic visibility by analysing key protocols in detail.
- Provide users comprehensive and accurate traffic reports able to offer at a fraction of price what many commercial products do together.
- Promote open-source software, while protecting selected IPRs.



Ntop's Approach to Traffic Monitoring

- Ability to capture, process and (optionally) transmit traffic at line rate, any packet size.
- Leverage on modern multi-core/NUMA architectures in order to promote scalability.
- Use commodity hardware for producing affordable, longliving (no vendor lock), scalable (use new hardware by the time it is becoming available) monitoring solutions.
- •Use open-source to spread the software, and let the community test it on unchartered places.



Some History

- In 1998, the original ntop has been created.
- It was a C-based app embedding a web server able to capture traffic and analyse it.

About Summary Al	I Protoco	is IP Mer	da Utila Plugini	a Admin		(C) 1998-2005 - Luc	a Deri
Fratfic Unit: (Sytes) Par	skats]		1	Host Informa	tion		
Host	Domain	IP Address	MAC Address	Other Name(s)	Bandwidth	Nw Board Vendor	Hops
192.168.0.5 💣 🏲		192.168.0.5	00:0D:93:75:DB:C8	=		Apple Computer	
10.96.5.25		10.96.5.25		=			
cache netikos.com 👌 🏲	全	10.96.4.122	13	-	1		
pidc01.netikos.com 🔭	个	10.96.4.10	5				1
192.168.0.1 🖘		192.168.0.1	00:13:10:07 F1 AE				
 Olick here for more Bandwidth values a percentage point). The SENT bandwidth 	informati are the pe The total th is sho	on about host toentage of the of the values of an as an ar	and domain sorting to total bytes that relo wit WOT be 100% as id the RECEIVED ba	ap has seen on the in s local static will be o individith is shown as	erface. Hover the mouse to see the actu surfed TWICE (once as sent and again (al value (rounded to the nea as received).	rest fui
Report created on Sat (Generated by ntop v.3.2 to 1998-2005 by Luca De	Dot 22 14 (power) rl, built:	:25:55 2005 pc-apple-dar Oct 22 2005	[nlop uptime: 1:52 win8.2.0] : 11:55:17.	u.			

- Contrary to many tools available at that time, ntop used a web GUI to report traffic activities.
- It is available for Unix and Windows under GPL.



Ntop Architecture



19-20-21 agosto 2016 Parco Ex Caserma Cocco Pescara

Why was Ntop obsolete?

- •Its original LAN-oriented design prevented ntop from handling more than a few hundred Mbit.
- •The GUI was an old (no fancy HTML 5) monolithic piece written in C so changing/extending a page required a programmer.
- Ntop could not be used as web-less monitoring engine to be integrated with other apps.
- Many components were designed in 1998, and it was time to start over (spaghetti code).



Ntopng Design Goals

- Clean separation between the monitoring engine and the reporting facilities.
- Robust, crash-free engine (ntop was not really so).
- Platform scriptability for enabling extensions or changes at runtime without restart.
- Realtime: most monitoring tools aggregate data (5 mins usually) and present it when it's too late.
- Many new features including HTML 5-based dynamic GUI, categorisation, DPI.



Ntopng Architecture

•Three different and self-contained components, communicating with clean API calls.





Network Traffic & Probe

- Where to place the probe?
 - Near the router?
- And the Network Edge?
- Probe:
 - Passive (only analysis).
 - Active (analysis and block).



Network measures

- Quantitative:
 - Top Talkers.
 - Protocols or applications.
 - Destinations.
 - Host counters.
- Qualitative:
 - Traffic not allowed.
 - Errors.



Some problems

Security issues

- All the network traffic is captured and not just the one sent to the sniffing host.
- If there is a switched network it is captured only a part of traffic.
- Usability limited to those who have root capabilities.
- Performance
 - Sniffer implies also the cpu load because all the captured packets must be analysed by the program and not just those directed to the host.



Traffic mirror

- Hardware:
 - Hub (Copper Ethernet).
 - Optical Splitter (Optical Fibers).
 - Tap (Copper/Fiber).
- Software:
 - Switch Port Mirror (1:1, 1:N).
 - Switch VLAN Mirror (N:1).
 - Switch Traffic Filter/Mirroring (Packet Brokers).



Network Taps





Ntopng Monitoring Engine

- Coded in C++ and based the concept of flow (set of packets with the same 6-tuple).
- Flows are inspected with a home-grown DPI-library named nDPI aiming to discover the "real" application protocol (no ports are used).
- Information is clustered per:
 - (Capture) Network Device
 - Flow
 - Host



Local vs Remote Hosts [1/2]

- Ntopng keeps information in memory at different level of accuracy in order to save resources for hosts that are not "too relevant".
- For this reason at startup hosts are divided in:
 - Local hosts

The local host where ntopng is running as well the hosts belonging to some "privileged" IPv4/v6 networks. These hosts <u>are very relevant</u> and thus ntopng keep full statistics.

• Remote hosts

Non-local hosts for which we keep a minimum level of detail.



Local vs Remote Hosts [2/2]

- For local hosts (unless disabled via preferences) are kept all L7 protocol statistics, as well basic statistics (e.g. bytes/ packets in/out).
- No persistent statistics are saved on disk.
- A system host is the host where ntopng is running and it is automatically considered local as well the networks of its ethernet interfaces.

Loci I System 🗯
gistry of ccTLD it - IIT-CNR]
1 [192.12.193.11/32] [Pisa 🚺]
1

Information Lifecycle

- Ntopng keeps in memory live information such as flows and hosts statistics.
- As the memory cannot be infinite, periodically non-recent information is harvested.
- Users can specify preferences for data purge:





Packet Processing Journey

- 1.Packet capture: PF_RING (Linux) or libpcap.
- 2.Packet decoding: no IP traffic is accounted.
- 3.IPv4/v6 Traffic only:
 - 1. Map the packet to a 6-tuple flow and increment stats.
 - 2.Identify source/destination hosts and increment stats.
 - 3.Use nDPI to identify the flow application protocol
 - 1.UDP flows are identified in no more than 2 packets.
 - 2.TCP Flows can be identified in up to 15 packets in total, otherwise the flow is marked as "Unknown".
- 4. Move to the next packet.



PF_RING [1/2]

 In 2004 we realised the the Linux kernel was not efficient enough to fulfil our packet capture requirements and thus we have written a in-kernel circular buffer named PF_RING.



PF_RING [2/2]

- It creates a straight path for incoming packets accessed from user-space applications with memory mapping.
- No need to use custom network cards: any card is supported.
- Transparent to applications: legacy applications need to be recompiled in order to use it (pcap-over-PF_RING).
- Developers familiar with network applications can immediately take advantage of it without having to learn new APIs.
- Acceleration support for many popular open-source applications including Wireshark, Suricata and Snort.



Balancing Traffic with PF_RING

- At high speed on modern multi-core systems, it is a good idea to improve the overall system performance by balancing traffic across cores.
- PF_RING shares ingress packets across multiple consumer applications (e.g. ntopng) by hashing them (tunnels are supported) so that they are balanced applications via virtual PF_RING



so that they are balanced to multiple consumer applications via virtual PF_RING network interfaces.



Moving towards 10 Gbit and above [1/2]

- The original PF_RING is a good solution up to 3/5 Gbit but not above as the cost of packet copy into the ring is overkilling.
- PF_RING ZC (Zero Copy) is an extension that allows packets to be received/transmitted in zero copy similar to what FPGA-accelerated cards (e.g. Napatech) do in hardware.





Moving towards 10 Gbit and above [2/2]

- In ZC a packet is put by the ingress NIC into a shared memory buffer, and it hop across applications (and VMs) by exchanging the buffer pointer (packets don't move).
- Thanks to this solution it is possible to create arbitrary packet processing topologies at multi-10 Gbit line rate using commodity hardware x86 servers and adapters (ZC natively supports Intel ethernet adapters).



PF_RING ZC Network Topologies [1/2]

Use Case:



PF_RING ZC Network Topologies [2/2]



Use Case:

Application pipeline or run multiple apps (e.g. ntopng) in VMs to insulate them.

Pescara

PF_RING (ZC) and ntopng

- Using PF_RING (ZC) with ntopng has several benefits:
 - Ntopng can scale to 10 Gbit and above by spawning several ntopng instances each bound to a (few) core(s).
 - It is possible to send the same packet to multiple apps. For instance it is possible to send the same packet to ntopng (for accounting purposes) and n2disk (ntop's application for dumping packet-to-disk at multi-10G) and/or and IDS (e.g. Suricata and snort).



The need for DPI in Monitoring [1/2]

- Limit traffic analysis at packet header level it is no longer enough (nor cool).
- Network administrators want to know the real protocol without relying on the port being used.
- Selected protocols can be "precisely dissected" (e.g. HTTP) in order to extract information, but on the rest of the traffic it is necessary to tell network administrators what is the protocol flowing in their network.



The need for DPI in Monitoring [2/2]

- DPI (Deep Packet Inspection) is a technique for inspecting the packet payload for the purpose of extracting metadata (e.g. protocol).
- There are many DPI toolkits available but they are not what we looked for as:
 - They are proprietary (you need to sign an NDA to use them), and costly for both purchase and maintenance.
 - Adding a new protocol requires vendor support (i.e. it has a high cost and might need time until the vendor supports it) = you're locked-in.
- On a nutshell DPI is a requirement but the market does not offer an alternative for open-source.



Say hello to nDPI

• Ntop has decided to develop its own GPL DPI toolkit in order to build an open DPI layer for ntop and third party applications.

rco Ex Caserma Cocco

- Supported protocols (> 220) include:
 - P2P (Skype, BitTorrent)
 - Messaging (Viber, Whatsapp, MSN, The Facebook)
 - Multimedia (YouTube, Last.gm, iTunes)
 - Conferencing (Webex, CitrixOnLine)
 - Streaming (Zattoo, Icecast, Shoutcast, Netflix)
 - Business (VNC, RDP, Citrix, *SQL)



nDPI Overview

- Portable C library (Win and Unix, 32/64 bit).
- Designed for user and kernel space
 - Linux ndpi-netfilter implements L7 kernel filters
- Used by many non-ntop projects (eg. xplico.org) and part of Linux distributions (e.g. Debian).
- Able to operate on both plain ethernet traffic and encapsulated (e.g. GTP, GRE...).
- Ability to specify at runtime custom protocols (port or hostname dns, http, https -based).



nDPI API

• The nDPI API is pretty simple

- ndpi_init_detection_module()
 ndpi_exit_detection_module()
 Init/term the nDPI library.
- ndpi_load_protocols()
 Load custom protocol definitions.
 - ndpi_detection_process_packet()
 Process the packet in nDPI and return the L7 protocol or NDPI_UNKNOWN (too early or detection failed).
 - ndpi_guess_protocol()
 Guess a L7 protocols when DPI fails.



nDPI on ntopng

- In ntopng all flows are analysed through nDPI to associate an application protocol to them.
- L7 statistics are available per flow, host, and interface (from which monitoring data is received).
- For network interfaces and local hosts, nDPI statistics are saved persistently to disk (in RRD format).



nDPI on ntopng: Interface Report [1/2]


nDPI on ntopng: Interface Report [2/2]



Left-click on any chart element to zoom in, right-click to zoom out.

Graph Info	Total: 35.69 MB	95th Percentile: 6.37 Mbit	Min: 0 bps @ 12/04/2016 22:40:10	Max: 6.71 Mbit @ 12/04/2016 22:35:49
------------	-----------------	----------------------------	----------------------------------	--------------------------------------



Ntopng and Redis [1/2]

- Redis is an open source key-value in-memory database.
- Ntopng uses it to cache data such as:
 - Configuration and user preferences information.
 - DNS name resolution (numeric to symbolic).
 - •Volatile monitoring data (e.g. hosts JSON representation).
- •Some information is persistent (e.g. preferences) and some is volatile: ntopng can tell redis how long a given value must be kept in cache.



Ntopng and Redis [2/2]

- Redis is also used as a (persistent) queue for requests towards external applications.
 - If configured (-F command line option), periodically flow status is saved onto a redis queue, requests are packed, and send to a remote BigData system.
- In essence Redis is used by ntopng to store information that might take too much memory (if kept on ntopng memory space), or to pile up list of things that are executed periodically or that require interaction with remote applications that might be slow or temporary unavailable.



Lua-based Ntopng Scriptability [1/3]

- A design principle of Ntopng has been the clean separation of the GUI from engine (in ntop it was all mixed).
- This means that ntopng can (also) be used (via HTTP) to feed data into third party apps such as Nagios or OpenNMS.
- All data export from the engine happens via Lua.
- •Lua methods invoke the ntopng C++ API in order to interact with the monitoring engine.



Lua-based ntopng Scriptability [2/3]

- /scripts/callback/ scripts are executed periodically to perform specific actions.
- /scripts/lua/ scripts are executed only by the web GUI.

	Name	*	Date Modified	Size	
7	📄 callbacks		Sep 30, 2013 2:15 PM		e.,
	🐑 daily.lua		Apr 17, 2013 1:55 PM	29 bytes	
	🐑 hourly.lua		Apr 17, 2013 1:55 PM	29 bytes	
	🐒 minute.lua		Sep 30, 2013 2:15 PM	5 KB	1
	nprobe-collector.lua		Sep 30, 2013 2:15 PM	4 KB	1
	🐑 second.lua		Sep 30, 2013 2:15 PM	2 KB	6
7	iii lua		Today 3:58 PM		0
	🐒 about.lua		Jun 30, 2013 10:27 PM	2 KB	E.
	🕨 🚞 admin		Jun 26, 2013 11:24 PM		-
	🐒 aggregated_host_details.lua		Sep 30, 2013 2:15 PM	6 KB	1
	🐑 aggregated_host_stats.lua		Aug 15, 2013 4:37 PM	442 bytes	1
	🐒 aggregated_hosts_stats.lua		Sep 30, 2013 2:15 PM	1 KB	6
	🐑 db.lua		Aug 12, 2013 7:48 PM	320 bytes	1
	🐑 do_export_data.lua		Sep 30, 2013 2:15 PM	765 bytes	6
	🐑 export_data.lua		Sep 4, 2013 7:49 PM	1 KB	1
	🐒 find_host.lua		Sep 4, 2013 7:49 PM	2 KB	1
	🐑 flow_details.lua		Sep 30, 2013 2:15 PM	7 KB	1
	🐑 flow_stats.lua		Aug 15, 2013 4:37 PM	1 KB	
	🐒 flows_stats.lua		Aug 15, 2013 4:37 PM	2 K8	
	🐒 get_aggregated_host_info.lua		Aug 15, 2013 4:37 PM	857 bytes	
	🐒 get_flows_data.lua		Sep 4, 2013 7:49 PM	6 KB	1
	🐒 get_geo_hosts.lua		Sep 4, 2013 7:49 PM	2 KB	1
	🐑 get_host_activitymap.lua		Sep 30, 2013 2:15 PM	505 bytes	
	🐒 get_host_traffic.lua		Sep 4, 2013 7:49 PM	399 bytes	6
	🐒 get_hosts_data.lua		Sep 30, 2013 2:15 PM	6 KB	1
	get_hosts_interaction.lua		Sep 30, 2013 2:15 PM	2 KB	E.

• Example:

http://ntopng:3000/lua/flow_stats.lua



Lua-based ntopng Scriptability [3/3]

- ntopng defines (in C++) two Lua classes:
 - interface
 - Hook to objects that describe flows and hosts.
 - Access to live monitoring data.
 - •ntop
 - General functions used to interact with ntopng configuration.
- Lua objects are usually in "read-only" mode
 - C++ sets their data, Lua reads data (e.g. host.name).
 - Some Lua methods (e.g. interface.restoreHost()) can however modify the information stored in the engine.



Ntopng as a NetFlow/sFlow Collector [1/3]

• The "old" ntop included a NetFlow/sFlow collector. Considered the effort required to support all the various NetFlow dialects (e.g. Cisco ASA flows are not "really" flows), in Ntopng we have made a different design choice.





Ntopng as a NetFlow/sFlow Collector [2/3]

- nProbe (a home-grown NetFlow/sFlow collector/probe) is responsible for collecting/generating flows and convert them to JSON so that ntopng can understand it.
- The communication ntopng <-> nProbe is over ØMQ a simple/fast messaging system that allows the two peers to be decoupled while:
 - Avoiding "fat" communication protocols such as HTTP.
 - Relying on a system that works per message (no per packet) and handles automatic reconnection if necessary.



Ntopng as a NetFlow/sFlow Collector [3/3]

Flows are sent in the following format (gzip+encryption)

- {"8":"192.12.193.11","12":"192.168.1.92","15":"0.0.0.0","10":0,"14":0,"2":5,"1": 406,"22":1412183096,"21":1412183096,"7":3000,"11":55174,"6":27,"4":6,"5":0,"16": 2597,"17":0,"9":0,"13":0,"42":4}
- Where:
 - "<Element ID>": <value> (example 8 = IPV4_SRC_ADDR)
- Contrary to what happens in NetFlow/sFlow ntopng (collector) connects to nProbe (probe) and fetches the emitted flows. Multiple collectors can connect to the same probe. No traffic is created when no collector is attached to the probe.



Flow Collection Setup: an Example

Flow collection/generation (nProbe)

•Probe mode

nprobe --zmq "tcp://*:5556" -i eth1 -n none

•sFlow/NetFlow collector mode

nprobe --zmq "tcp://*:5556" -i none -n none --collector-port 2055

Data Collector (ntopng) •ntopng -i <u>tcp://127.0.0.1:5556</u>



Flow Collection: Pull vs Poll Mode

Poll Mode

- host X> ntopng -i "tcp://Y:1234" --zmq-encrypt-pwd myencryptionkey
- host Y> nprobe -n none --zmq "tcp://*:1234" --zmqencrypt-pwd myencryptionkey

Push Mode

- host X> ntopng -i "tcp://Y:1234" --zmq-collector-mode -zmq-encrypt-pwd myencryptionkey
- host Y> nprobe -n none --zmq "tcp://*:1234" --zmqprobe-mode --zmq-encrypt-pwd myencryptionkey



Creating ntopng Clusters [1/3]

- Ntopng is not only a flow collector, but it can export flows in the same JSON format used in the received flows.
- This allows complex clusters to be created:



Creating ntopng Clusters [2/3]

- In many companies, there are many satellite offices and a few central aggregation points.
- Using ØMQ (both ntopng and nProbe flows are in the same format) it is possible to create a hierarchy of instances.
- Each node aggregates the traffic for the instances "below" it, so that at each tree layer you have a summarised view of the network activities.





Creating ntopng Clusters [3/3]

Example

Start the remote nProbe instances as follows

- [host1] nprobe --zmq "tcp://*:5556" -i ethX
- [host2] nprobe --zmq "tcp://*:5556" -i ethX
- [host3] nprobe --zmq "tcp://*:5556" -i ethX
- [host4] nprobe --zmq "tcp://*:5556" -i ethX
- If you want to merge all nProbe traffic into a single ntopng interface do:
 - ntopng -i tcp://host1:5556,tcp://host2:5556,tcp://
 host3:5556,tcp://host4:5556

If you want to keep each nProbe traffic into a separate ntopng interface do:
 ntopng -i tcp://host1:5556 -i tcp://host2:5556 -i tcp://host3:5556 -i tcp://host4:5556



Managing Alerts [1/2]

- In many situations it is fundamental to set alerts that can signal anomalous conditions
- Ntopng handles host/interface/network alerts hooked to multiple metrics
- Metrics include bytes/packets received/generated
- User-submitted alerts are continuously monitored in the background



Managing Alerts [2/2]

Every Minute	Every 5 Minutes	Houriy Daily			U	ptime: 1 h,	16 min, 12 sec	>	
Alert Function		Threshold			4	1 Alert	86 Hosts 148 F	lows	
oytes		> 📀 25000000							
		Bytes delta (sent + received)							
dns		> 📀 DNS traffic delta bytes (sent + rece	Queu	ed Alerts	6				
02p			Action	Date	Severity	Туре	Description		
packets		Peer-to-peer traffic delta bytes (ser	B Showing 1	Mon Apr 11 18:36:01 2016	Warning	Cross	Threshold bytes of 192.168.2.130 [11	crossed by [68 > 25]	host
			€ Purge	All Alerts					
Rearm minutes		3 0 The rearm is the dead time betwee	n one alert	generation and t	he potentia	al generation (of the next alert o	of the same	e kind.
		s	ave Conf	iguration [Delete	All Host Co	onfigured Alerts	s]	



Sending ntopng Alerts to Nagios [1/2]

- Nagios is the de-facto standard in infrastructure monitoring
- Ntopng features alert propagation to Nagios

Nagios Alerts		
Alerts To Nagios Enable sending ntopng alerts to Nagios NSCA (Nagios Service Check Acceptor).	On	Off
Nagios NSCA Host Address of the host where the Nagios NSCA daemon is running. Default: localhost.	192.168.1.10	Save
Nagios NSCA Port Port where the Nagios daemon's NSCA is listening. Default: 5667.	5667 5	Save



Sending ntopng Alerts to Nagios [2/2]

- Alerts are sent to Nagios via NSCA
- Nagios will intercept all alerts that are explicitly declared as passive services
- Passive service description format is:
 - •NtopngAlert_<host/network/interface>_<timespan>_<metric>

ntopng- host	NtopngAlert	?	ок	12-23-2015 15:25:50	0d 17h 27m 59s	1/1	Alert for host Y!
	NtopngAlert_192.168.1.15_min_bytes	?	OK	12-23-2015 09:13:22	0d 6h 47m 34s	1/1	OK, alarm deactivated
	NtopngAlert_192.168.2.0/24	?	OK	12-23-2015 11:02:34	0d 4h 33m 4s	1/1	OK, alarm deactivated
	NtopngAlert_192.168.70.0/24_min_egress	?	WARNING	12-23-2015 15:33:01	0d 0h 6m 5s	1/1	Threshold egress crossed by network 192.168.70.0/24 [1180 > 10]
	NtopngAlert_192.168.70.0/24_min_ingress	?	WARNING	12-23-2015 15:33:01	0d 0h 2m 5s	1/1	Threshold ingress crossed by network 192.168.70.0/24 [11241211 > 10]



System+Network Monitoring Operating System Hardware

stap

dtrace

perf

- Historically on Unix there are many tools for system monitoring.
- iostat I/O Bridge tcpdump ip nicstat dtrace iotop Like when we started the blktrace I/O Controller Network Controller dtrace dstat development of ntop, all Disk Disk Port ping these tools are nice per-se, but are not integrated with the rest of the environment.
- Ntopng/nProbe monitor network activities, but have no visibility of the processes that are originating the observed network activities.



Applications

System Call Interface

Device Drivers

ZFS

VFS

Block Device Interface

perf

ext3/.

LVM

System Librari

Sockets

TCP/UDP

IP

Ethernet

Scheduler

Virtual

Memory

DBs, all server types,

top

CPI

DRAM

perf

Various:

sar

/proc

pidstat

slabtop

Port

dstat

free

mpstat dstat

System+Network Monitoring [2/3]

How most system management tools work on Linux:





System+Network Monitoring [3/3]

 Using Ntopng/nProbe you can see the flows that are are being exchanged across systems but it is not possible to know more than that.





System+Network Monitoring [3/3]

- It would be desirable to know exactly what is the process originating the traffic observed and what resources the process is using while generating such traffic.
- In essence we would like to see this picture:



Welcome to Sysdig

- Sysdig is a Linux framework developed by Draios Inc for capturing system calls.
- The kernel module intercepts the calls.
- The user-space libs receive and interpret the received calls.





Why Sysdig?

- Contrary to all other tools available for system monitoring, sysdig implements the "packet paradigm" applied to system events:
 - Events are received in a way similar to what happens with packet capture.
 - It is possible to store events on pcap-like files and reply them later on.
- To simplify things, instead of using the sysdig API, we added native sysdig support in PF_RING so that all apps (e.g. Ntopng) can use it.



Integrating sysdig in nProbe [1/2]

- Instead of complicating the design of ntopng with sysdig support, we have decided to extend nProbe with system visibility.
- nProbe monitors both the network interfaces and the system events via PF_RING.
- Network and system information is then combined and exported in standard network flows over NetFlow v9/ IPFIX and in JSON to ntopng for data visualization.



Integrating sysdig in nProbe [2/2]

- Using sysdig, nProbe is able to bind a (local) process to a network flow, and to monitor its I/O activities, CPU and memory utilisation.
- This way we know for sure what network activities are performed by processes, including those activities performed by trojans and malware that start up, send the packet-of-death and then disappear.
- Thanks to the PID/father-PID hierarchy it is possible to know an any time the exact activation chain.



Ntopng+nProbe+sysdig

- When ntopng receives flow enriched with system information, it interprets it, and depicts:
 - The process-to-flow association.
 - For flows whose peers are hosts monitored by nProbe instances, it "glues" the flows together.
 - The process call father/process hierarchy is depicted.
 - The overall system process view including the process relationships.



Process Network Communications



Flow/Process Drill-down [1/2]



Flow/Process Drill-down [2/2]

Client Process Information	
User Name	deri
Process PID/Name	13058/ntopng [son of 11235/tcsh]
Average CPU Load	0.71 %
I/O Wait Time Percentage	0 %
Memory Actual / Peak	1.4 MB / 1.46 MB [95.7%]
VM Page Faults	0
Server Process Information	
User Name	redis
Process PID/Name	1769/redis-server [son of 1/init]
Average CPU Load	0.12 %
I/O Wait Time Percentage	0 %
Memory Actual / Peak	344.13 KB / 344.13 KB [100%]
VM Page Faults	0

2016

Flow-to-Process binding

Dynamically Updated

Flow-to-Process binding

Dynamically Updated



Active Process Network Connections

Info	Application	L4 Proto	Client Process	Client Peer	Server Process	Server Peer	Duration	Breakdown	Total Bytes
Info	HTTP	TCP	chrome	17)=:50540	java	17 10:9200	1 h, 39 min, 2 sec	Server	37.88 MB
Info	HTTP	TCP	chrome	i7 14: 45671	java	17 10:9200	1 h, 11 min, 2 sec	Server	24.03 MB
info	HTTP	TCP	ntopng	i7 **: 48526	java	i7 🍽:9200	1 sec	Client S	7.62 KB
info	HTTP	TCP	ntopng	17 10:48528	java	i7 m :9200	1 sec	Client S	7.27 KB
Info	HTTP	TCP	ntopng	17 1=:48529	java	17 10:9200	1 sec	Client	6.71 KB
Info	HTTP	TCP	ntopng	17 14:48527	java	17 10:9200	1 sec	Client S	6.69 KB
Info	HTTP	TCP	ntopng	17 10:48525	java	17 10:9200	1 sec	Client S	6.48 KB
Info	HTTP	TCP	chrome	17 148461	java	17 12:9200	1 sec	Server	5.2 KB

Showing 1 to 8 of 8 rows



Process Network Traffic

Active Processes

10 -

Name	Flows Count	Active Since	Traffic Sent	Traffic Rcvd
sshd	1 -	19 min, 42 sec	39.99 КВ —	20.87 KB -
redis-server	1-	1 day, 6 h, 19 min, 50 sec	1.96 GB 🛧	1.86 GB 🛧
ntopng	40 🛧	1 day, 6 h, 19 min, 50 sec	2.05 GB 🛧	3.89 GB 🛧
nprobe	2 -	1 day, 6 h, 19 min, 55 sec	2.72 GB 🛧	198.96 MB 🛧
java	13 🔶	1 h, 41 min, 32 sec	63.57 MB 🛧	554.87 KB 🛧
dropbox	1-	20 min, 27 sec	28.79 KB -	6.36 KB -
chrome	17 🛧	14 h, 18 min, 2 sec	11.58 MB 🛧	790.09 MB 🛧

Showing 1 to 7 of 7 rows



Process Protocols Drill-Down





Processes Timeline





User Flows and Processes

🚨 elasticsearch - 📕 i7

Applications Protocols

Flow

Active Flows

10 - Applications-

Info	Application	L4 Proto	Client Process	Client Peer	Server Process	Server Peer	Duration	Breakdown	Total Bytes
Info	HTTP	TCP	chrome	i7 🎮:50540	java	i7 🍽:9200	1 h, 54 min, 2 sec	Server	43.81 MB
Info	HTTP	TCP	chrome	i7 14: 45671	java	i7 🎮:9200	1 h, 26 min, 2 sec	Server	26.99 MB
Info	HTTP	TCP	chrome	i7 🎮:48461	java	i7 🍽:9200	15 min, 1 sec	Server	3.82 MB
Info	HTTP	TCP	ntopng	i7 17: 33419	java	i7 🍽:9200	1 sec	Client	8.13 KB
Info	HTTP	TCP	ntopng	i7 🎮:33418	java	i7 🍽:9200	1 sec	Client S	6.72 KB
Info	HTTP	TCP	ntopng	i7 🛤:33417	java	i7 🍽:9200	1 sec	Client S	6.71 KB



Historical Flow Navigation [1/2]

- Ntopng can send (-F) network flows to MySQL
 - •A built-in database explorer retrieves such flows and allows them to be navigated and searched

From:		To:		Client/Server Hos	t: Protocol:	Port:	Info:	Application Protoco
11/04/201	6 🗰	11/04/2016	m		Any ‡			Any
Duration: 1	h							Search Flows
Summary	IPv4 Fi	ows IPv6 Fic	ws Ta	ilkers Protocols				
Summary Search I	IPv4 Fie	ows IPv6 Fic	ws Ta	ilkers Protocols				
Summary Search I	IPv4 Fi Results Total Flow	ows IPv6 Fic	ws Ta Traffic Volu	ume	Total Packets		Traffic Rate	Packet Rate
Summary Search I IPv6	IPv4 Fit Results Total Flow 65 Flows	ows IPv6 Fic	ws Ta Traffic Volu 9.64 KB	ume	Total Packets 87 Pkts		Traffic Rate 21.92 bps	Packet Rate 0.02 pps


Historical Flow Navigation [2/2]

Summary	IPv4 F

Info

Info

Info

Info

IPv6 Flows lows

client

Talkers Protocols

IPv6 Top Flows [11/04/2016 17:56:35 - 11/04/2016 18:56:35]

L4 Client Application Proto Server Begin End Bytes Thpt UDP simones-macbook-pro.loca...:mdns ff02::fb:mdns 11/04/2016 11/04/2016 811 B 3.24 ? Unknown 18:22:02 18:22:03 ? Unknown UDP simones-macbook-pro.loca...:mdns ff02::fb:mdns 11/04/2016 11/04/2016 811 B 3.24 18:22:03 18:22:02 ? Unknown UDP fe80::3e15:c2ff:feb7:720...:mdns ff02::fb:mdns 11/04/2016 11/04/2016 613 B 4.9 Kbit 18:39:30 18:39:30 UDP 11/04/2016 11/04/2016 324 B ? Unknown fe80::b675:eff:fe92:8917...:dhcpv6ff02::1:2:dhcpv6-18:50:43 client 18:50:40 server ? Unknown UDP fe80::b675:eff:fe92:8917...:dhcpv6-11/04/2016 11/04/2016 324 B ff02::1:2:dhcpv6-

server

Showing 1 to 5 of 65 rows

Download flows: IPv4 IPv6

Extract pcap: * Bulk download and full

18:41:55

pcap extraction options

19-20-21 agosto

Parco Ex Caserma Cocco Pescara

18:41:58

3

5

>

4

5-

Avg

Kbit

Kbit

648

bps

648

bos

Historical Talkers [1/2]

Top Talkers can be automatically extracted from flows.

• Every top talker can be clicked to inspect its peers.

• Every peer can be clicked to inspect L7 application protocols.



Historical Talkers [2/2]

IPv6 Flows Ta	alkers Protoco	Is								
							50 -			
IP Address	Total Traffic	Total Packets	Ingress Traffic	Ingress Packets	Egress Traffic	Egress Packets	Flows			
192.168.2.130	18.27 MB	119,364	9.02 MB	86,911	9.25 MB	32,453	2,320			
									K	
Su	ummary IPv	4 Flows IPv6	Flows Talkers	Protoco	ols					
Int	terface en4 / Tal	kers with 172.217	7.16.5 ♡							
										50 -
Ho	st Name	IP Address	Total Traffic*	Total Pa	ackets T	raffic Sent	Packets Sent	Traffic Received	Packets Received	Flows
192	2.168.2.130 ≓	192.168.2.130	1.68 ME		3,317	0 B	0	1.68 MB	3,317	12
IPv6 Flows	Talkers Protoc	cols								
172.217.16.5 / App	plications between	172,217,16.5 and 1	192.168.2.130 ♡							
							50 -			
		Т	raffic Volume		Packe	ts	Flows			
			1.68 MB		3,31	17	12			
	IPv6 Flows 1 IPv6 Flows 1 IPv6 Flows 72.217.16.5 / Ap	IPv6 Flows Talkers Protocol IP Address Total 192.168.2.130 18.27 MB Summary IPv Interface en4 / Tal Host Name 192.168.2.130 = IPv6 Flows Talkers Protocol 72.217.16.5 / Applications between	IPv6 Flows Talkers Protocols IP Address Total Traffic Total Packets 192.168.2.130 18.27 MB 119,364 Summary IPv4 Flows IPv6 Interface en4 / Talkers with 172.213 Host Name IP Address 192.168.2.130 192.168.2.130 IPv6 Flows Talkers Protocols Talkers Flows Talkers Protocols Talkers	IPv6 Flows Talkers Protocols IP Address Total Traffic Total Packets Ingress Traffic Ingress 192.168.2.130 18.27 MB 119,364 9.02 MB Ingress Summary IPv4 Flows IPv6 Flows Talkers Interface en4 / Talkers with 172.217.16.5 ♥ Interface Host Name IP Address Total Traffic 192.168.2.130 192.168.2.130 1.68 MB IPv6 Flows Talkers Protocols 72.217.16.5 / Applications between 172.217.16.5 and 192.168.2.130 ♥ Interface Volume 1.68 MB 1.68 MB Interface Volume	IPv6 Flows Talkers Protocols IP Address Total Traffic Total Packets Ingress Traffic Ingress Packets 192.168.2.130 18.27 MB 119,364 9.02 MB 86,911 Summary IPv4 Flows IPv6 Flows Talkers Protocols Interface en4 / Talkers with 172.217.16.5 ♀ Total Traffic ♥ Total Packets 192.168.2.130 IPv6 Flows Total Traffic ♥ Total Packets 192.168.2.130 IPv6 Flows Total Traffic ♥ Total Packets 192.168.2.130 IPv6 Flows Talkers Protocols Traffic Protocols Traffic Volume IPv6 Flows Talkers Protocols Traffic Volume IPv6 Flows Talkers Protocols Traffic Volume Interface Interface 1.68 MB Interface	IPv6 Flows Talkers Protocols IP Address Total Traffic Total Packets Ingress Traffic Packets 192.168.2.130 18.27 MB 119,364 9.02 MB 86,911 9.25 MB Summary IPv4 Flows IPv6 Flows Talkers Protocols Interface en4 / Talkers with 172.217.16.5 CV Total Traffic Y Total Packets T IPv6 Flows Talkers Protocols T IPv6 Flows Talkers P A A	Talkers Protocols IP Address Total Traffic Total Packets Ingress Traffic Egress Packets Egress Traffic 192.168.2.130 18.27 MB 119,364 9.02 MB 86,911 9.25 MB 32,453 Summary IPv4 Flows IPv6 Flows Talkers Protocols Interface IP Address Total Traffic Total Packets Traffic Sent Ipv6 Flows Talkers Protocols Traffic Sent Ingress Traffic Sent Ingress Ipv6 Flows Talkers Protocols Traffic Sent Ingress Ingress Ingress Ingress Ingress Ingress Ingres Ingres Ingres Ing	Talkers Protocols 1P Address Total Total Ingress Egress Egress Flows 192.168.2.130 18.27 MB 119,364 9.02 MB 86,911 9.25 MB 32,453 2,320 Summary IPv4 Flows IPv6 Flows Talkers Protocols Interface end / Talkers with 172.217.16.5 CO Total Traffic Volume Total Packets Traffic Sent Packets Sent 192.168.2.130 192.168.2.130 1.68 MB 3,317 0 B 0	Prof Flows Talkers Protocols IP Address Total Total Ingress Ingress Egress Flows 192.168.2.130 18.27 MB 119,364 9.02 MB 86,911 9.25 MB 32,453 2,320 Summary IPv4 Flows IPv6 Flows Talkers Protocols Traffic Packets Traffic Received 192.168.2.130 192.168.2.130 1.68 MB 3,317 0 B 0 1.68 MB 192.165.2 Talkers Protocols Traffic Volume Packets Traffic Sent Packets Sent Traffic Received 192.165.2 192.168.2.130 1.68 MB 3,317 0 B 0 1.68 MB 192.165.2 Tatlers Protocols Traffic Volume So - So - 192.165.2 Traffic Volume Packets Traffic Sent Packets So - 192.165.2 Traffic Volume Packets Flows So - So - So - 10.6 1.68 MB 3,317 12 So - <td>PR6 Flows Talkers Pretocols IP Address Total Ingress Packets Flows Summary IP Address Ingress Tartic Packets Flows Iso <td< td=""></td<></td>	PR6 Flows Talkers Pretocols IP Address Total Ingress Packets Flows Summary IP Address Ingress Tartic Packets Flows Iso Iso <td< td=""></td<>



Historical Applications [1/2]

- Top Applications can be automatically extracted from flows as well.
- Every top application can be clicked to inspect hosts that have used it.
- Every host can be clicked to inspect peers that have used a given application to communicate with the host.



Historical Applications [2/2]

Chart	IPv4 Flows	IPv6 Flows	Talkers	Protocols			
Interface	en4						
• protoco	xls	D of	· host pee	ars by protocol		E all	
Select sa	ved	\$	Select save	id		\$	
							50 -
Applicati	on				Traffic Volume	Packets	Flows
🔹 Applei	Tunes 🕼				471 B	2	1
IGMP 🔇					600 B	10	10
NTP 😂					1.05 KB	12	6

Chart IPv4 Flows	IPv6 Flows	Talkers Protocols			
Interface en4 / Apple	iTunes talkers ♡	2			
♥ protocols	🖬 all	 host peers by protocol 	al 🗉		
Select saved	\$	Select saved	•		
					50 -
Host Name		Address	Traffic Volume*	Packets	Flows
192.168.2.130 ==		192.168.2.130	471 B	2	1

Chart IPv4 Flows	IPv6 Flows	Talkers	Protocols			
Interface en4 / Applei	Tunes talkers	AppleiTunes t	talkers with 192.168.2.130 🎔			
P protocols	0.4	 host pee 	ers by protocol	0.0		
Select saved	\$	Select save	ed	\$		
						50
Host Name	Addre	155		Traffic Volume *	Packets	Flow
jake.unipi.it	131.1	14.18.19		471 B	2	



Ntopng and Big Data

- Using SQLite to save flows persistently is good when flows are not too many and the system that runs ntopng has storage.
- For large deployments or disk-less systems (e.g. ARMbased PCs) it is desirable to upload flows on remote, cloud-based, systems able to scale with the number of flows.
- In essence ntopng has been opened to what is currently defined as "big data" systems that can scale with data in volume and speed.



Integrating Ntopng with ElasticSearch [1/2]

- An emerging Big Data system is ElasticSearch that is used by a large community because of its flexibility and user interface (Kibana) that allow visual applications to be developed in minutes.
- Although we do not want to bind ntopng only with ES, we believe that its integration is a good starting point for:
 - Opening ntopng to the Big Data world.
 - Allowing people to use ntopng as data source and let them use ES for long-term data storage and develop custom dashboards using Kibana.



Integrating Ntopng with ElasticSearch [2/2]

- Ntopng dumps exported flows in JSON format onto a Redis queue enriched with some specified ES attributes (e.g. @timestamp that specifies the time such flow has been exported).
- As soon as there is a minimum number of flows in queue, a ntopng thread packs them together and sends them to ES using the ES bulk API.
- ES indexes the received flows and make them available to external applications such as the Kibana dashboard.



Ntopng Process Dashboard in Kibana [1/2]



Ntopng Process Dashboard in Kibana [2/2]

 The GUI refreshes automatically as new data arrive and users can drill down data or visualise raw flows.

 FV4_DST_ADDR FV4_SRC_ADD 	Q Ø Micro Analysis of IPV4_SRC	_ADDR (string)	X Count (400
🗆 json			Count / 400
C L4_DST_PORT	Value	Action	events
C L4_SRC_PORT	1. 192.12.193.11	Q Ø	297
C L7_PROTO	2. ::1	90	102
C LAST_SWITCHED	3. 127.0.0.1	0.0	1
OUT_BYTES			
OUT_PKTS			
C PROTOCOL	@timestamp (100%),@version (100%),	DST_VLAN (100%), FIRST_SWIT	TCHED
SRC_FATHER_PRO	(100%),IN_BYTES (100%),IN_PKTS (1	00%),IPV4_DST_ADDR (100%),	L4_DST_PORT
SRC_FATHER_PRO	(100%),L4_SRC_PORT (100%),LAST_	SWITCHED (100%), More >	
SRC_PROC_ACTU/	ita Terms ◄		
SRC_PROC_AVERA			

View: Table / JSON / Raw		
Field	Action	Value
Øtimestamp	0,⊘Ⅲ	2014-10-01T20:00:25.021Z
Oversion	0,⊘⊞	1
DST_VLAN	0,⊘⊞	0
FIRST_SWITCHED	0.⊘ Ⅲ	1412193584
IN_BYTES	0,⊘Ⅲ	40
IN_PKTS	0, ⊘ Ⅲ	1
IPV4_DST_ADDR	0.⊘ Ⅲ	192.12.192.104
IPV4_SRC_ADDR	0,⊘⊞	192.12.193.11
L4_DST_PORT	0, ⊘ Ⅲ	1234
L4_SRC_PORT	0 ⊘ ⊞	55451
LAST_SWITCHED	0, Ø Ⅲ	1412193584
OUT_BYTES	0 ⊘ Ⅲ	60
OUT_PKTS	0,⊘⊞	1
PROTOCOL	0, ⊘ Ⅲ	6
SRC_FATHER_PROC_NAME	0, ⊘ Ⅲ	init
SRC_FATHER_PROC_PID	0.011	1
SRC_PROC_ACTUAL_MEMORY	0, ⊘ ⊞	1467872
SRC_PROC_AVERAGE_CPU_LOAD	0, ⊘ Ⅲ	0
SRC_PROC_NAME	0, ⊘ Ⅲ	ntopng
SRC_PROC_NUM_PAGE_FAULTS	0,⊘⊞	0
SRC_PROC_PEAK_MEMORY	0.0 Ⅲ	1533796
SRC_PROC_PID	0, ⊘ Ⅲ	13058
SRC_PROC_USER_NAME	Q Ø III	deri

What's Next on Big Data and Ntopng

- We believe that the big data world is still very liquid and it is not clear what the emerging technology will be.
- We believe ntopng should be just a data source without being tightly integrated with any external tool (ntopng speaks JSON and HTTP so we can cover most of them pretty easily).
- We are experimenting with other big data technologies (e.g. druid.io) and we plan to open it to all the emerging technologies available.



Ntopng on Virtual Environments

- Ntopng has been packaged for major Linux distributions such as Debian/Ubuntu, CentOS/RedHat and also FreeBSD and OSX (brew): installation couldn't be simpler.
- However the current trend is going towards virtualised environments (not just VMs such as VMware) and laaS (Infrastructure as a Service) and thus we need to support them.







Embedding Ntopng [1/4]

- Historically we have started our first embed attempt in 2003 with the Cyclades TS100.
- The nBox was used to analyse traffic then sent to ntop for representation.
- After 10 years we have tried again with ntopng.



op.org	mBox metermingament conside			
atten	nBox Configuration			
Collector(s) IP:	192 108 103 1 2055 Specify the list of collector(s) that will revenue the NetFlow Rows entitled by ribus. The formal of the field a segarated by contrasts if multiple addresses are specified (or this case ribos will send hows in round rotats). Reampler 193 198 100 2 2 2015 109 108 100 2 2016			
Max Flows Lifetime:	Ad sec Plant sider that the specified likenes will be experied by Hick.			
Max Flower Idle Time:	nd wec Plant inactive longer of the specified line will be experied by ribes.			
Expired Flows Scart:	60 kec. Expired hows will be periodically checked according to the specified value.			
Flow Export Policy	Confector manual rubon (default) If multiple collectors are defined. It is possible to specify the flaw expertability. Round rubon means, that whenever a flow needs to be exported it is sent sequentially to all collectors in mound rubon, so that each collector receives a publicit of all flows. Reflector means that all the sufficience receives at the flows. If unity one collector has been defined. If is option has its effect.			
Flows Export Delay:	200 mm Some collectors converties and index of the second of the two reasons it regard he reconstancy to store shown flow expect rule by adding a short delay (< 1000 mm) after a flow expect. Leave this field set to 0 (sected) for max expect speed.			
BPF Filter:	It is possible to evel flows only for a subset of the possel tarffic. It is possible to specify a RPF filter p to the same formal used by the popular topdump tool; for selecting taffic. Leave the field angly if you work to process all the received taffic.			
and the second second	\$1671502845102C84731656689546250			



Embedding Ntopng [2/4]

 It is a while that we are working towards a cheap platform for everyone...



BeagleBoard Black



Embedding Ntopng [3/4]

- Main issue with boards like BeagleBoard/Raspberry: only one ethernet interface built-in (extra ports via USB).
- Boxes like Ubiquity Networks EdgeRouter are also an option but we're basically jeopardising a box designed for other tasks (issues with hardware guarantee, GUI etc.).
- Open issues: how to monitor traffic? Port mirror or tap?



Embedding Ntopng [4/4]

- We're trying to find the third way...
 - Rely on a hardware company to build a cheap ARMbased box suitable for network monitoring (ntop is making software no hardware).
 - Two ethernet interfaces to be used as either a bumpin-the-wire or 2 x independent interfaces.
 - Built-in hardware tap with bypass.
 - Able to monitor xDSL/cable and up.
 - Power-over-Ethernet (POE).



Final Remarks

- Over the past 18 years ntop created a software framework for efficiently monitoring traffic.
- "We have a story to tell you, not just hacks".
- Commodity hardware, with adequate software, can now match the performance and flexibility that markets require. With the freedom of open source.
- Ntopng is available under GNU GPLv3 from http://www.ntop.org/.









Parco Ex Caserma Cocco Pescara

Using Ntopng

Logging into ntopng

Welcome to ntopng

admin

.....



If you find ntopng useful, please support us by making a small donation. Your funding will help to run and foster the development of this project. Thank you.

C ntop.org - ntopng is released under GPLv3.

Hint: the default user and password are admin



ntopng Dashboard

ashboard:	Talkers	Hosts	Applications	ASNs	Senders	
					Top Flow Talkers	
	212.69.173	3.191				
1	212.69.17	1.251				
	212.69.17	1.57				whois.nic.it
	212.69.17	3.181				
	93.190.236	6.154				
93.190.236.163					das.nic.it	
998-2013 - r	ntop.org	(+57.10)			26.08 Mbps [33,317 pps]	

 \bigcirc \bigcirc 2016

Available Menu Items



Interfaces - Admin	- Search Host
Available Interfaces eth4 eth5	
Switch Interfaces	eth4 eth5







Dynamic Web Interface



6.06 Mbps [4,857 pps] O Uptime: 1 day, 2 hours, 18 min, 27 sec 38,257 hosts 158,961 flows

Throughput	Total Bytes
8.09 Kbit 🕹	94.23 MB
5.59 Kbit 🛧	60.15 MB
5.16 Kbit 🕹	60.15 MB





Flows Monitoring [1/2]

Active Flows

fo	Application	L4 Proto	Client	Server	Duration	Breakdown	Throughpu	DHCPV6
fo	VRRP	VRRP	fe80::192:12:192:7	ff02::12	1 day, 2 hours, 4 min, 19 sec	Client	8.09 Kbit 🖌	HTTP
fo	VRRP	VRRP	192.12.192.7	224.0.0.18	1 day, 2 hours, 4 min, 19 sec	Client	5.59 Kbit 4	ICMP
fo	VRRP	VRRP	192.168.18.7	224.0.0.18	1 day, 2 hours, 4 min, 19 sec	Client	5.16 Kbit 🗸	ICMPV6
ifo	DHCP	UDP	0.0.0.0:68	255.255.255.255:67	1 day, 2 hours, 3 min, 57 sec	Client	0 bps -	IGMP
fo	OSPF	89	192.12.192.7	224.0.0.5	1 day, 2 hours, 4 min, 13 sec	Client	0 bps 🗸	OSPF
fo	OSPF	89	192.168.18.7	224.0.0.5	1 day, 2 hours, 4 min, 7 sec	Client	0 bps -	Unknown
fo	OSPF	89	192.168.18.9	224.0.0.5	1 day, 2 hours, 4 min, 14 sec	Client	359.83 bps 1	VRRP
ifo]	OSPF	89	192.12.192.9	224.0.0.5	1 day, 2 hours, 4 min, 16 sec	Client	359.83 bps 4	whois-DAS
ifo	OSPF	89	192.168.18.34	224.0.0.5	1 day, 2 hours, 4 min, 7 sec	Client	0 bps 🕹	1 MB
fo	OSPF	89	192.12.192.34	224.0.0.5	1 day, 2 hours, 4 min, 7 sec	Client	0 bps -	1 MB

> O O O 2016

Showing 1 to 10 of 151325 rows

← First Prev 1

2 3 4

5

Next

Last ->

✿ 10 - Applications

Parco Ex Caserma Cocco

Pescara

Flows Monitoring [2/2]

Flow: 192.12.192.237:53060 = whois	i.nic.it:5043 Overview								
Client	web-r1.nic.it:53060								
Server	whois.nic.it:5043								
Application Protocol	HTTP								
First Seen	11/10/2013 13:45:26 [6 min, 54 sec ago]								
Last Seen	11/10/2013 13:52:12 [8 sec ago]								
Total Traffic Volume	7.03 KB —								
Client vs Server Traffic Breakdown	192.12.192.237								
Client to Server Traffic	63 Pkts / 7.03 KB -								
Server to Client Traffic	0 Pkts / 0 Bytes -								
Actual Throughput	0 bps —								
TCP Flags	SYN PUSH ACK								

© 1998-2013 - ntop.org Generated by ntopng v.1.0.1 (r6749)

for user admin and interface eth5



193.98 Kbps [260 pps] O Uptime: 1 day, 2 hours, 4 min, 49 sec 1,272 hosts 153,747 flows

19-20-21 agosto
2016

Host Monitoring [1/3]

Hosts List

IP Address	Location	Symbolic Name	Seen Since	ASN	Breakdown	Throughput	Traffic			
192.12.192.230	Local	das.nic.it	1 day, 2 hours, 4 min, 49 sec	2597 🕑	Rovd	13.57 Kbit	bit 51.27 GB			
192.165.67.192	Remote	192.165.67.192	1 day, 2 hours, 4 min, 31 sec	34971 🗗	Sent	0 bps	9.62 GB			
192.165.67.166	Remote	192.165.67.166	1 day, 2 hours, 4 min, 31 sec	34971 🕑	Sent	659.95 bps	9.18 GB			
78.46.216.98	Remote	78.46.216.98 📟	1 day, 2 hours, 4 min, 48 sec	24940 🗗	Sent	219.98 bps	7.87 GB			
192.165.67.22	Remote	192.165.67.22	1 day, 2 hours, 4 min, 30 sec	34971 🗗	Sent	0 bps	s 7.81 GB			
78.47.50.132	Remote	78.47.50.132 🕅	1 day, 2 hours, 4 min, 48 sec	24940 🕑	Sent	879.93 bps	7.18 GB			
62.149.189.11	Remote	62.149.189.11	1 day, 2 hours, 4 min, 35 sec	31034 🕑	Sent	0 bps	1.44 GB			
192.12.192.242	Local	whois.nic.it	1 day, 2 hours, 4 min, 49 sec	2597 🕑	Rovd	84.86 Kbit	964.02 MB			
224.0.0.18	Remote	vrrp.mcast.net	1 day, 2 hours, 4 min, 49 sec		Rovd	8.81 Kbit	it 120.35 MB			
213.154.243.80	Remote	213.154.243.80 🚍	18 hours, 57 min, 57 sec	12859 🕑	Sent	4.51 Kbit	116.72 MB			



Parco Ex Caserma Cocco Pescara

۰

10 -

2

Host Monitoring [2/3]

ntop		Home - Flows Hosts - Interfaces - Admin - Search Host										
Host: 192.12.192.230	Overview Tra	ffic Packets Protocols Flows Talkers Geomap Contacts Historical										
(Router) MAC Address		00:1B:21:30:81:D0										
IP Address		192.12.192.230										
ASN		2597 C [Registry of ccTLD it - IIT-CNR]										
Name		das.nic.it 🕑 Local										
First Seen		10/10/2013 11:47:36 [1 day, 2 hours, 5 min, 14 sec ago]										
Last Seen		11/10/2013 13:52:48 [2 sec ago]										
Sent vs Received Traffic Bre	akdown	Rovd										
Traffic Sent		39,290 Pkts / 2.27 MB -										
Traffic Received		746,704,852 Pkts / 51.27 GB 🛧										
JSON		Download										
Activity Map												

Host Monitoring [3/3]



Activity Map

1 second resolution host and aggregation activity

Compressed bitmap

- > ls -l client14.dropbox.com
- 4 -rw-rw-rw- 1 nobody nogroup 24 Oct 11 02:31 client14.dropbox.com

• Saved persistently on disk (Local Hosts only)





Traffic Aggregations [1/2]

- nDPI extracts specific attributes from traffic that ntopng aggregates (if configured):
 - DNS/Whois responses
 - HTTP host names
 - Operating System (from HTTP headers)
- Aggregations can be enabled (they are off by default) and are handled just as flows and hosts.



Traffic Aggregations [2/2]

			-		
Name	Protocol	Seen Since	Last Seen	Qu	All
dnsmon.nic.it	HTTP	1 day, 46 min, 20 sec	4 sec		Operating System
ð Linux x86_64	Operating System	1 day, 46 min, 20 sec	4 sec		HTTP
daisy.ubuntu.com	DNS	1 day, 46 min, 16 sec	28 sec	1	13,613 -
7.ntop.org	HTTP	11 sec	1 sec		26 -
intel Mac OS X 10_8_5	Operating System	11 sec	1 sec		26 -
www.google.com	DNS	1 min, 30 sec	39 sec		15 -
prinptflomq.nic.it	DNS	39 sec	39 sec		2 -
takoxonuji.nic.it	DNS	40 sec	40 sec		2 -
lkomppxne.nio.lt	DNS	39 sec	39 sec		2 -
checkip.dyndns.com	DNS	40 sec	40 sec		2-

10

Showing 1 to 10 of 20 rows

← First Prev 1 2 Next Last →



 19-20-21 agosto
 Parco Ex Ca

 2016
 Pescara

Hosts and Aggregations Interaction



2016

Geolocation

G					
Host: 192.12.193.5	Overview	Traffic	Packets	Protocols	F
(Router) MAC Address			78:AC:0	C0:A7:0D:4C	
IP Address			192.12	193.5 [Pisa I]

Hosts GeoMap



NOTE

- 1. 9 Browser reported home map location [Latitude: 43.71949459086955, Longitude: 10.4219399273913]
- 2. In order to visualize maps you must:
 - 1. Have a working Internet connection.
 - 2. Have compiled ntopng with geolocation and started with it.
 - 3. Have active flows between peers with public IP addresses.

16

19-20-21 agosto

0

3. HTML browser geolocation is used to place on map hosts based on unknown locations.

Live Host Activities

		02.20	0	0	2:30	S		02:	35			02:	40		(02:4	5		(J2:5	0			02:5	5	
																										i7.ntop.org
																										dnsmon.nic
																										192.12.193
10	. Ellista dal	1140	i il i i i i	1444	AH 11	1116		hihi				11.1	1.1			MIN	11	64.4	dille	d id	111	Jill.				192.12.192.
11				Label			li hi		11.			U.I.		11.1		1111	1.1	MB	1.11.	hi	311.	1.4	all.		-11	192.12.193.
(-1	Andra I In 11	In Laker	11111	11	1111	41.1	lu- ill	.44 141	NIA	na.	dh N	a et l	BLM	141	h r	1971	hat 1	a siti	1.1.1	11	dit	1.1	114	1.12	11	192.12.193.
r. h.	A & I & B & B	6 5 8 5 F	A A A	V Male	la far fa	1 5 M	511	L L	1.1			al s	h h h	It do to	100		1.1.1	1.1	1.1.1	the late	1.18	1 A)	h, birli	1 - E		192.12.193.
1.1	The states and the states of t	a linita	a Malei Ar		Lah la	4141	L h il		24	LIN.	LL M		NU.	ta l	d II	h di		ii 1	A.L.	11.1	IL I	tu	1.11	1. 10		192.12.193.
11	J.H.H.H.H.H.	1111	1351	1 0 1 1	11.1	111	111	11	o La	11	4.4	11	1	1 1 1	11	1.1	1.1.1		U.U	[]]	1.1	11	U)			192.12.193.
0.66																	111.999.11							1.5	1.1	192.12.193.
1.1	11 11	1111	1111	111	1 = 1	111	1.1	111	11		11	111	11	Lef 1	11	11	11	111	1.11	1.1.1	1.1	11	1.1.1			192.12.193.
d g	i i Uddarhai ii a i	n kon del tur	i h kiliminin	naprati i	adigate in	manin	lanakorika	ddadh	duit	land.	nii nin	in the second	rikalaa	, dan qi sah	II GENIU	ni ultra	nıh (hi	distant	ndaria		i i ha	unund	nhruu		100	192.12.193.
11/ a	Neuli I.A., I	fi niellet	di la h	at at h	all a	14	164 B	(h).#	ar n	h i linit	1.1	6111	11.4	10.1	1	4		htth	l + h	•1 •	di at		nt ni	1.10	14	192.12.193.
	11111	1.1	1.1.1	1111	11	111	11	11.	1.1	11.		1.1.1	1.1.	1.1.1	111		1.1.1	1.1	113	11	1	1.1	1.1.1			192.12.193.
1	Real (Black)	11.11	200.4	11.1.1	in i	14.1 1	11	hatter	4 1	111		L II I	1.1	11 4	11.		1.1	1.8	111	11	111	14	1 15	-		192.12.193.
			医无水		1.1	Jad		1				1	1	1	1.11	in I	dù.		1				1			192.12.193.
	dilli ara ir	kak i	AN E		15 s	14	- 0	6 4	14	delle I	a i	aki .	1	1	1	1		li	1							192.12.193.
				1	ü	1	•						1		1		1		1	1	4	1				perseus.nic.
Ϋ́.	1 1	- 71	1.1		8	11	1	•					1		2	1 m			1			1				192.12.193.
L.		1				1	1	1					1 13				1									snom01.nic.
1		i l ra		0.10	CI II	Q. I	10	LL.	99	10			of 1	11	1.1.1	3.0	10	1.1	1961	1.1	1.1	1.1	003			192.12.193.
1.	Weburd Lin 7	AC ARE	alsh i a	1.1.1.1	th: 1	141 - 1°	4.1.1	Halti	11 34	if felb	14 13	11, 15, 1	1	1 1	u: 1		1.1.3	11.4	E E E	1.1	JL 11	1.1			• 41	192.12.193.
1	1111	1.1.1	1.1.1	1111	L. I. I.	111	1	-h	94	1.1.	l h l	111	11	1.1	111	1	111	1.1	1.1.1	11	-	1.1.1				192.12.193.

Historical Activities

- All relevant counters are saved on disk in RRD.
- Interface counters are saved with 1 second resolution. Hosts counters every 5 minutes.



Using Ntopng as a Live Data Source

- ntopng is a server able to serve data to third party applications via HTTP.
- Data is exported via JSON.
- This mechanism can be extended via Lua scripts.

Traffic Sent	744,856 Pkts / 97.54 MB 🛧			
Traffic Received	807,881 Pkts / 190.37 MB 🛧			
JSON	Ownload			
Activity Map		Export Dat	a	
		Host:	IP or MAC Address	tod
			Export JSON Data Reset Form	100
		19-20-21 agosto	Parco Ex Caserma Cocco	

Using Ntopng with NetFlow/sFlow

ntopng can handle flows (Net/sFlow) via nProbe.

- Data Collector (ntopng)
 - ntopng -i tcp://127.0.0.1:5556
- Probe (nProbe)



- nprobe --zmq "tcp://*:5556" -i eth1 -n none (probe mode)
- nprobe --zmq "tcp://*:5556" -i none -n none --collector-port 2055 (sFlow/NetFlow collector mode)








Parco Ex Caserma Cocco Pescara

NtopNg e il monitoraggio del traffico di rete (in high-speed network)

Giuseppe Augiero

<talk@augiero.it> - @GiuseppeAugiero

Luca Deri

<deri@ntop.org> - @lucaderi

(c) Ntop.org - 19 agosto 2016 - Moca 2016 - Pescara - Parco ex Caserma Cocco