

Tutorial Sicurezza

Gestione quotidiana della sicurezza

Roberto Cecchini, Simona Venuti

GARR WS14 | Roma, 2 e 4 dicembre 2014



Agenda

- wireshark
- NfSen
- argus



wireshark



wireshark perche'?

Nella gestione quotidiana della sicurezza wireshark risponde a dubbi e quesiti di questo tipo:

- Come mai la rete e' lenta/ferma/congestionata?
- Come funziona il routing della mia rete?
- Da cosa dipende il malfunzionamento della subnet X?
- E perche' i PC del terzo piano vanno in rete pianissimo?
- Mi pare/Mi hanno segnalato di avere un virus ma l'antivirus dice di no

In reti complesse, oppure dove girano applicativi complessi e' oneroso risalire alla sorgente reale di un problema, o al malfunzionamento di un apparato di rete (magari uno stupido HUB nel sottoscala del laboratorio X), o alla rilevazione di un attacco informatico

Wireshark puo' essere utile

Elementi packet capturing

Vengono catturati pacchetti di livello 2 (frames)

Modalita' promiscua sull'interfaccia

- Richiede permessi di root per accedere all'interfaccia
- L'interfaccia riceve il traffico di tutto il segmento di rete
- Si comporta come bridge trasparente

In reti switched si configura una SPAN (o mirror) port sullo switch per avere tutto il traffico di tutti i segmenti

Wireshark cenni generali

Cattura pacchetti e frames, li ricomponi e li rende umanamente leggibili. Una volta convertiti il software permette la completa riclassificazione, ordinamento, raggruppamento

- Analizzatore in TEMPO REALE di tutti i protocolli
- Visualizzazione immediata dei problemi
- Potente interfaccia grafica
- Potente interfaccia a riga di comando (tshark)
- Possibilita' di scomporre il traffico in qualsiasi modo
- Esportazione di parti di traffico su criteri arbitrati (editcap)

wireshark requisiti

Wireshark si puo' utilizzare con profitto anche su semplici PC desktop
E' fondamentale capire in quale semento di rete mettere wireshark

- Una buona CPU per l'elaborazione in tempo reale
- Almeno 256Mb di RAM
- Almeno 100Mb di disco (dipende da quanti dati si vuole catturare)
- Una scheda di rete che supporti il modo promiscuo
- Permessi di root/administrator
- Librerie pcap

Wireshark funzionamento

Installazione:

- Linux:

`apt-get install wireshark`

- Windows:

Da <http://www.wireshark.org> scaricare l'eseguibile

Lanciare l'eseguibile si installazione

Eseguire wireshark:

- Linux: eseguirlo da root
- Windows: lanciarlo da Administrator

Wireshark cattura

The screenshot shows the Wireshark Network Analyzer interface. The main window title is "The Wireshark Network Analyzer [Wireshark 1.12.2 (v1.12.2-0-g898fa22 from master-1.12)]". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, and Help. The toolbar contains various icons for file operations, search, and capture control. The filter bar is empty. The main area is divided into three tabs: Capture, Files, and Online. The "Capture" tab is active, showing the "Interface List" on the left and the "Wireshark: Capture Options" dialog box in the center.

Interface List
Live list of the capture interfaces (counts incoming packets)

Start
Choose one or more interfaces to capture

- Connessione alla rete locale (LAN)* 22
- VirtualBox Host-Only Network
- Connessione alla rete locale (LAN)
- Connessione alla rete locale (LAN) 2
- Connessione rete wireless

Capture Options
Start a capture with detailed options

Wireshark: Capture Options

Capture

Capture	Interface	Link-layer header	Prom. Mode	Snaplen [B]	Buffer [MiB]	Capture Filter
<input type="checkbox"/>	10.0.0.40					
<input checked="" type="checkbox"/>	Connessione alla rete locale (LAN) 10.0.0.47	Ethernet	enabled	262144	2	

Capture on all interfaces
 Use promiscuous mode on all interfaces

Capture Filter: Compile selected BPFs

Capture Files

File: Browse...

Use multiple files Use pcap-ng format

Next file every 1 megabyte(s)

Next file every 1 minute(s)

Ring buffer with 2 files

Stop Capture Automatically After...

1 packet(s) 1 megabyte(s)

1 file(s) 1 minute(s)

Display Options

- Update list of packets in real time
- Automatically scroll during live capture
- Hide capture info dialog

Name Resolution

- Resolve MAC addresses
- Resolve network-layer names
- Resolve transport-layer name
- Use external network name resolver

Start Close

Schermata di traffico

The screenshot shows the Wireshark interface with a packet list and a packet details pane. The packet list shows a sequence of TCP and TLSv1 packets. Packet 989 is selected, and its details are shown in the pane below.

No.	Time	Source	Destination	Protocol	Length	Info
980	241.7272370	10.0.0.47	23.50.146.8	TCP	66	60505-80 [SYN] Seq=0 win=8192 Len=0 MSS=1260 WS=4 SACK_PERM=1
981	241.7698030	23.50.146.8	10.0.0.47	TCP	66	80-60505 [SYN, ACK] Seq=0 Ack=1 win=14600 Len=0 MSS=1452 SACK_PERM=1 WS=32
982	241.7698980	10.0.0.47	23.50.146.8	TCP	54	60505-80 [ACK] Seq=1 Ack=1 win=66780 Len=0
983	242.6149980	10.0.0.47	23.50.155.27	TCP	54	60450-80 [FIN, ACK] Seq=505 Ack=2085 win=66780 Len=0
984	242.6564330	23.50.155.27	10.0.0.47	TCP	60	80-60450 [FIN, ACK] Seq=2085 Ack=506 win=15680 Len=0
985	242.6565020	10.0.0.47	23.50.155.27	TCP	54	60450-80 [ACK] Seq=506 Ack=2086 win=66780 Len=0
986	243.8012300	192.84.145.9	10.0.0.47	TLSv1	107	Application Data
987	243.8018430	192.84.145.9	10.0.0.47	TLSv1	107	Application Data
988	243.8022190	10.0.0.47	192.84.145.9	TLSv1	128	Application Data, Application Data
989	243.8024920	10.0.0.47	192.84.145.9	TLSv1	128	Application Data, Application Data
990	243.8037740	192.84.145.9	10.0.0.47	TLSv1	107	Application Data
991	243.8039290	192.84.145.9	10.0.0.47	TLSv1	107	Application Data
992	243.8048370	10.0.0.47	192.84.145.9	TLSv1	128	Application Data, Application Data
993	243.8061110	10.0.0.47	192.84.145.9	TLSv1	128	Application Data, Application Data
994	243.8553040	192.84.145.9	10.0.0.47	TLSv1	107	Application Data
995	243.8557850	10.0.0.47	192.84.145.9	TLSv1	128	Application Data, Application Data
996	243.8632930	192.84.145.9	10.0.0.47	TLSv1	107	Application Data
997	243.8637280	10.0.0.47	192.84.145.9	TLSv1	128	Application Data, Application Data
998	243.8652290	192.84.145.9	10.0.0.47	TLSv1	107	Application Data
999	243.8655130	10.0.0.47	192.84.145.9	TLSv1	128	Application Data, Application Data
1000	243.8718000	192.84.145.9	10.0.0.47	TLSv1	107	Application Data

Frame 989: 128 bytes on wire (1024 bits), 128 bytes captured (1024 bits) on interface 0
Ethernet II, Src: CompalIn_9a:87:0f (70:5a:b6:9a:87:0f), Dst: AsustekC_e8:53:72 (20:cf:30:e8:53:72)
Internet Protocol Version 4, Src: 10.0.0.47 (10.0.0.47), Dst: 192.84.145.9 (192.84.145.9)
Transmission Control Protocol, Src Port: 49175 (49175), Dst Port: 993 (993), Seq: 869, Ack: 978, Len: 74
Secure Sockets Layer

```
0000 20 cf 30 e8 53 72 70 5a b6 9a 87 0f 08 00 45 00 .0.SrpZ .....E.
0010 00 72 1d 57 40 00 80 06 81 a2 0a 00 00 2f c0 54 ..r.W@... ..../.T
0020 91 09 c0 17 03 e1 e6 78 a3 ca c5 ca f4 74 50 18 .....x .....TP.
0030 41 07 33 12 00 00 17 03 01 00 20 e1 af 80 3d c8 A.3..... =.
0040 53 79 bd 15 2f ce 08 ea 34 fc dc c3 64 e1 c1 09 Sy./... 4...d...
0050 f5 63 79 3b b5 ab 8b 2d 59 f2 55 17 03 01 00 20 .cy;... Y.U....
0060 b0 06 56 0a 40 fa 29 f8 fa e9 95 90 c0 7c 8c 61 ..V.@.)...|.a
0070 c5 1a 0c 9a 29 1b b0 89 7a 94 ea 8a 5a 3f 15 89 .....)...z...Z?..
```

Wireshark analisi traffico

The screenshot displays the Wireshark interface with several key panels:

- Protocol Hierarchy Statistics:** A table showing the distribution of traffic across various protocols. The 'Transmission Control Protocol' (TCP) is the most prominent, accounting for 97.46% of packets and 99.41% of bytes.
- Wireshark IO Graphs:** A line graph showing network activity over time. The X-axis represents time in seconds (40s to 140s), and the Y-axis represents the number of packets per tick. Multiple lines represent different protocols, with TCP and UDP being the most active.
- Conversations:** A table listing UDP conversations between different IP addresses and ports. The most frequent conversation is between 192.168.42.194 and 255.255.255.255.
- Wireshark: 755 Expert Infos:** A panel showing network errors and warnings. One warning is visible: 'Malformed TCP' with the summary 'New fragment overlaps old data (retransmission?)' and a count of 2.

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00 %	5084	100.00 %	3436774	0.185	0	0	0.000
Ethernet	100.00 %	5084	100.00 %	3436774	0.185	0	0	0.000
Internet Protocol Version 4	99.47 %	5057	99.94 %	3434863	0.185	0	0	0.000
User Datagram Protocol	2.01 %	102	0.54 %	18453	0.001	0	0	0.000
NetBIOS Datagram Service	0.04 %	2	0.01 %	459	0.000	0	0	0.000
SMB (Server Message Block Protocol)	0.04 %	2	0.01 %	459	0.000	0	0	0.000
SMB MailSlot Protocol	0.04 %	2	0.01 %	459	0.000	0	0	0.000
Microsoft Windows Browser Protocol	0.04 %	2	0.01 %	459	0.000	2	459	0.000
NetBIOS Name Service	0.12 %	6	0.02 %	552	0.000	6	552	0.000
Domain Name Service	1.55 %	79	0.43 %	14817	0.001	79	14817	0.001
Hypertext Transfer Protocol	0.30 %	15	0.08 %	2625	0.000	15	2625	0.000
Transmission Control Protocol	97.46 %	4955	99.41 %	3416410	0.184	4589	3218979	0.174
Secure Sockets Layer	1.67 %	85	1.00 %	34436	0.002	83	32593	0.002

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets B-A	Bytes B-A
192.168.42.194	60543	255.255.255.255	138	2	459	2	459	0	0
192.168.42.194	60544	255.255.255.255	137	6	552	6	552	0	0
ff02::1:2	547	fe80::70ca:232f:4bb:d3ab	546	7	1071	0	0	7	1071
192.168.42.194	60545	202.56.240.5	53	33	5620	17	1277	16	16
192.168.42.194	60546	202.56.240.5	53	10	1753	5	389	5	5
192.168.42.194	60547	202.56.240.5	53	6	916	3	231	3	3
192.168.42.194	60545	202.56.230.6	53	12	2923	6	472	6	6
192.168.42.194	60546	202.56.230.6	53	10	1936	5	381	5	5
192.168.42.194	60547	202.56.230.6	53	8	1669	4	311	4	4
192.168.42.194	52826	239.255.255.250	1900	15	2625	15	2625	0	0

Group	Protocol	Summary	Count
Malformed TCP		New fragment overlaps old data (retransmission?)	2

tshark

```
tshark [ -2 ] [ -a <capture autostop condition> ] ... [ -  
b <capture ring buffer option> ] ... [ -B <capture buffer size> ] [ -  
c <capture packet count> ] [ -C <configuration profile> ] [ -  
d <layer type>==<selector>,<decode-as protocol> ] [ -D ] [ -e <field> ] [ -  
E <field print option> ] [ -f <capture filter> ] [ -F <file format> ] [ -g ] [ -h ]  
[ -H <input hosts file> ] [ -i <capture interface>|- ] [ -I ] [ -K <keytab> ] [ -l ]  
[ -L ] [ -n ] [ -N <name resolving flags> ] [ -o <preference setting> ] ... [ -  
O <protocols> ] [ -p ] [ -P ] [ -q ] [ -Q ] [ -r <infile> ] [ -R <Read filter> ] [ -  
s <capture snaplen> ] [ -S <separator> ] [ -  
t a|ad|adot|d|dd|e|r|u|ud|udot ] [ -T fields|pdml|ps|psml|text ] [ -  
u <seconds type> ] [ -v ] [ -V ] [ -w <outfile>|- ] [ -W <file format option> ]  
[ -x ] [ -X <eXtension option> ] [ -y <capture link type> ] [ -  
Y <display filter> ] [ -z <statistics> ] [ --capture-comment <comment> ]  
[ <capture filter> ]
```

```
tshark -G [column-  
formats|currentprefs|decodes|defaultprefs|fields|ftypes|heuristic-  
decodes|plugins|protocols|values]
```

tshark

E' l'interfaccia a linea di comando di wireshark

Funziona come tcpdump, ma ha la superiore potenza di wireshark

E' molto utile da utilizzare in script di shell o programmi

Lista delle interfacce disponibili: **thark -D**

Cattura solo 1000 pacchetti: **thark -i #interfaccia -w file_out -c 1000**

Legge da file: **thark -r file.pcap**

Applicazione di filtri: **tshark -i #i -f FILTRO_sintassi_tcpdump**

Applicazione filtro complesso su file: **tshark -i #i -R filtro_file**

tshark statistiche

E' il comando che fa tutto quello che si puo' fare da interfaccia grafica

tshark -z conv,type[.filter]

Restituisce tutte le conversazioni di un certo tipo per un filtro

tshark -z expert,note.tcp

Restituisce, dal menu «expert», tutti i frames TCP con severity >= note

tshark -z follow.prot.mode.filter[.range]

Restituisce lo streaming TCP o UDP fra due nodi

Esempio:

```
tshark -z "follow,tcp,ascii,200.57.7.197:32891,200.57.7.198:2906"
```

Altre statistiche

`tshark -z hosts[,ipv4][,ipv6]`

Restituisce qualsiasi pacchetto IPv4 o IPv6 in formato host

`tshark -z http,stat, oppure -z http,tree oppure -z http_req,tree oppure
-z http_srv,tree`

Restituisce statistiche sull'utilizzo del protocollo HTTP

`tshark -z icmp,srt[,filter]`

Restituisce statistiche ICMP SRT

`tshark -z smb,sids oppure -z smb,srt[,filter]`

Restituisce statistiche sull'utilizzo del protocollo SMB

`tshark -z sip,stat[,filter]`

Restituisce statistiche sul protocollo SIP

etcetcetc

altre utility a linea di comando

rawshark

Legge lo stream dei pacchetti da un file

Esempio: **rawshark -d proto:http -r capture_file**

editcap

Compie ulteriori operazioni sui file binari pcap

Esempio: **editcap -d file.pcap file1.pcap 1 100**

Legge 100 pacchetti di file.pcap, -d droppa i pacchetti duplicati (DUP), scrive il risultato in file1.pcap1

mergcap

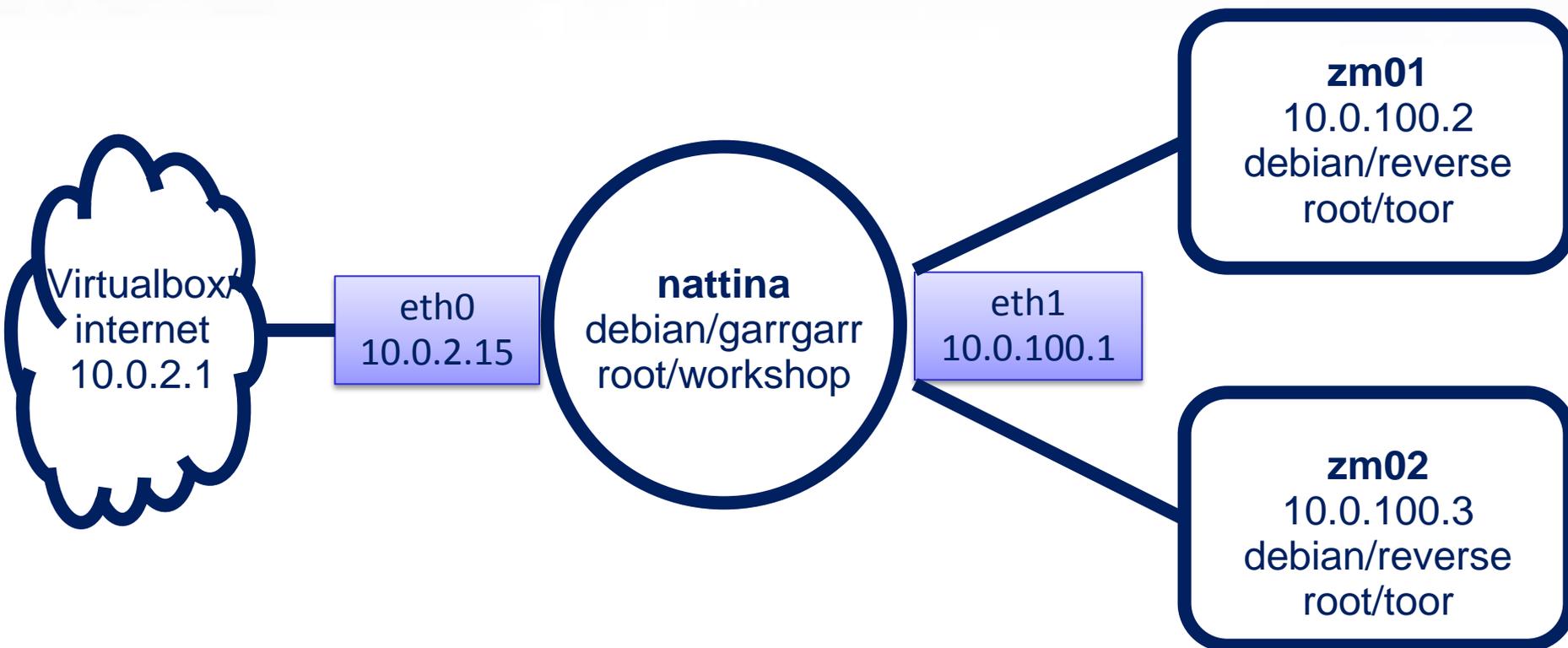
Mescola piu' file .pcap in un unico file, troncando o concatenando

Esempio: **mergcap -a -w newcapture.pcap capture.pcap capture1.pcap**

text2pcap legge ASCII hex dump file e li trasforma in formato .pcap

Esercitazione

Scenario



Esercitazione

Il GARR-CERT vi scrive che una vostra macchina X ha un virus non meglio specificato. Dai log inviati dal CERT non riuscite a trovare riscontro

Lo scopo e' trovare il malware installato sulla macchina, tramite wireshark.

Supponiamo che abbiate catturato una sessione di traffico della macchina X e che sia sul desktop di nattina

Su nattina entrare come utente debian/garrgarr

Lanciare wireshark

Aprire il file .pcap sul desktop

Trovare il malware

Suggerimento

Il malware solitamente viene
preso/installato
navigando su pagine web:
Io controllerei traffico strano
E pacchetti HTTP

NfSen



NfSen perche'?

Ogni APM quotidianamente si trova ad affrontare domande di questo tipo:

- Cosa ha causato questo picco nelle statistiche di traffico?
- Quali sono i top talkers/ le top subnet della mia sede?
- Quali sono le applicazioni piú utilizzate dai miei utenti?
- Qual e' il traffico relativo a questo incidente avvenuto il tale giorno?
- Si puo' analizzare questo DoS? Da quali indirizzi IP parte?

Gli strumenti tradizionali di monitoraggio basati sul protocollo SNMP non sono in grado di rispondere ...

NfSen e' una possibile semplice risposta

NfSen vs SNMP

SNMP	NfSen
Sorgente: i contatori dei router (numero dei pacchetti e ottetti)	Sorgente: qualsiasi apparato di rete router, switch, NAT server, server
Informazioni solo su interfacce fisiche Contengono solo il traffico aggregato	Informazioni relative a qualsiasi tipo di flusso, sessione, interfaccia
Traffico raw Servono altri strumenti per l'analisi	Traffico analizzabile per protocollo, ip, rete, porta etc. Analisi GRAFICA Analisi a linea di comando (nfdump)

NfSen vs IDS

IDS	NfSen
Pesantezza a livello di CPU, carico di rete, occupazione spazio disco	Leggero, flessibile, installabile ovunque, “zippabile”
Invasivo sulla privacy: ispezione del contenuto dei pacchetti	Rispettoso della privacy: ispezione solo degli header di pacchetto
Complessita' di configurazione di allarmi e individuazione traffico malevolo	GRAFICI! Sistema semplicissimo di allarmistica e potente nella personalizzazione

Elementi NetFlow

- Inizialmente sviluppato da Cisco
- Diventato uno standard “de facto”
- Implementato tutti i costruttori di hardware
- Esempi:
 - Juniper: c/jflowd
 - Alcatel:
 - Huawei: NetStream

NetFlow versioni

- V5, supportata da quasi tutti i vendor e ancora la più utilizzata
- V7, per gli switch catalyst della serie 5000
- V8, come la 7 con in più la possibilità di esportare flussi aggregati
- V9, più recente e flessibile (RFC 3954)
 - possibilità di definire template personalizzati
 - Trasporto di informazioni di livello2, IPV6, MPLS, BGP, protocol next_hop, etc
- IPFIX (Internet Protocol Flow Information eXport) standardizzazione IETF (RFC 5101 e 5102) di NetFlow v9

NetFlow versione 5

- Supporta solamente IPv4, il formato dei record NetFlow 5:
 - IP sorgente e destinazione
 - porte sorgente e destinazione
 - interfaccia d'ingresso e di uscita
 - AS number sorgente e destinazione
 - TCP flags
 - ToS (DSCP)
 - Contatori di ottetti e pacchetti

NetFlow versione 9

- Supporta IPv4, IPv6 ed MPLS, le informazioni trasportate:
 - Indirizzi IP sorgente e destinazione
 - porte sorgente e destinazione
 - interfaccia d'ingresso e di uscita
 - AS number sorgente e destinazione
 - Indirizzo IP "Next-Hop"
 - BGP "Next-hop"
 - TCP flags
 - ToS (DSCP)
 - Contatori di ottetti e pacchetti
 - Direzione del flusso
 - Indirizzo MAC sorgente e destinazione in ingresso
 - Indirizzo MAC sorgente e destinazione in uscita
 - Label VLAN
 - Label MPLS

NfSen – Cenni generali 1/2

NfSen puo' svolgere diversi compiti, sia a livello di analisi delle performance della rete che per la sicurezza della stessa

Analisi performance:

- gestione del traffico di rete
- l'ottimizzazione del routing
- la rilevazione di problemi
- Matrici di traffico
- Applicazioni e/o protocolli e/o porte piu' utilizzati
- Occupazione della banda

NfSen – Cenni generali 2/2

Per quanto riguarda la sicurezza di una rete:

- Analisi picchi di rete (picchi di flussi o di # di pacchetti o di traffico)
- Analisi incidenti di sicurezza
- Analisi e rilevamento BotNet e/o zombie
- Anomalie di traffico dovute a sfruttamenti di nuove vulnerabilita'
- Profilatura di macchine infette o compromesse
- Rilevazione scanport/probe
- Rilevazione server non autorizzati (DNS, NTP, WWW)

NfSen tutorial

Oggi invece vogliamo utilizzare questo strumento per aiutare gli APM ad individuare quale macchina dietro un NAT fa traffico malevolo

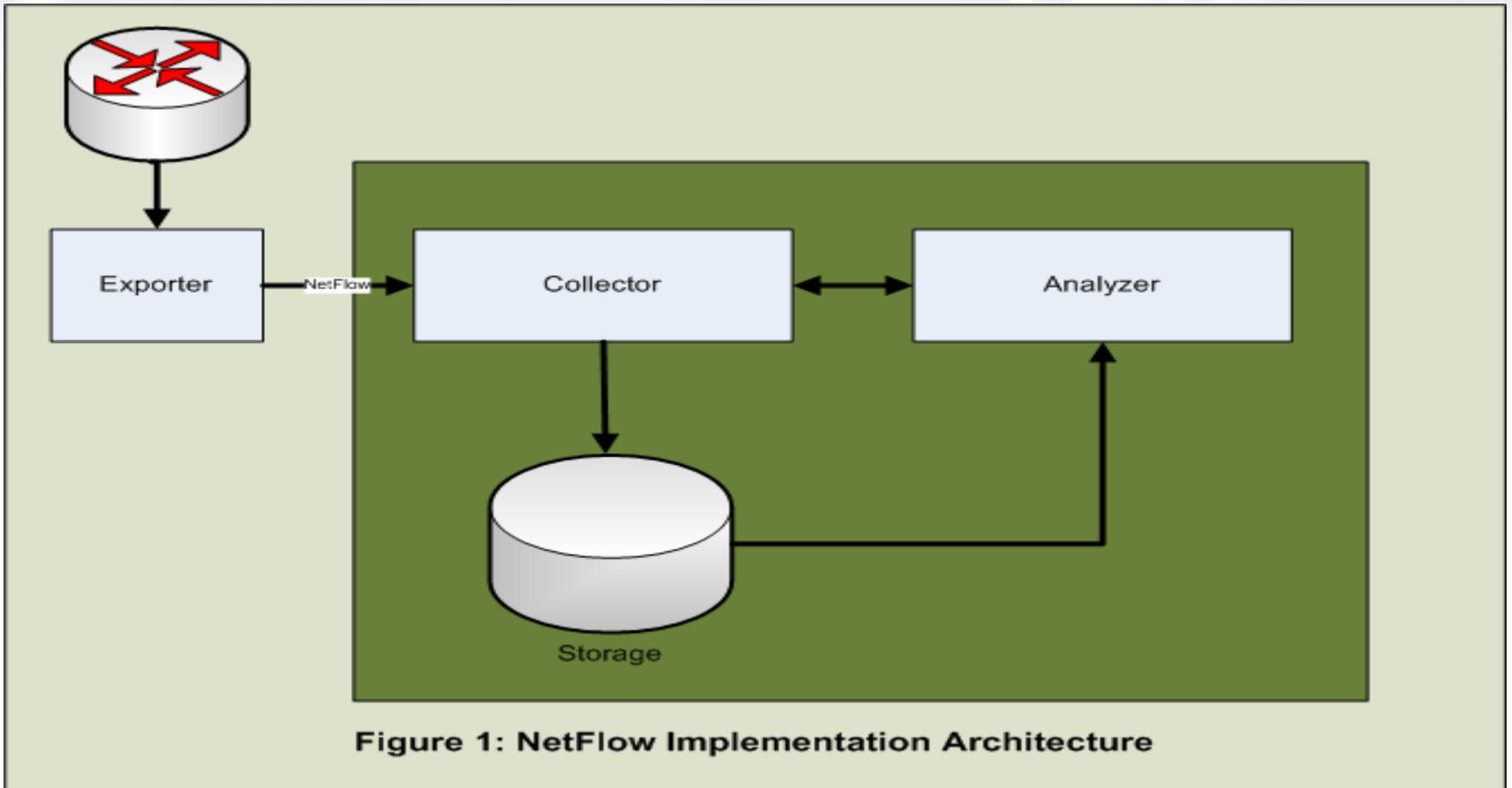
Scenario: Il GARR-CERT apre un incidente per una macchina della vostra rete e vi da' pochissime informazioni: data, IP, porta sorgente. L'IP in questione e' un NAT server che da' la rete a un elevato numero di client. Senza adeguati strumenti e' difficoltoso risalire alla macchina interna infetta e/o compromessa.

NB: il GARR nelle AUP chiede espressamente che in caso di incidente o violazione, una struttura deve sempre essere in grado di risalire al responsabile della macchina incriminata

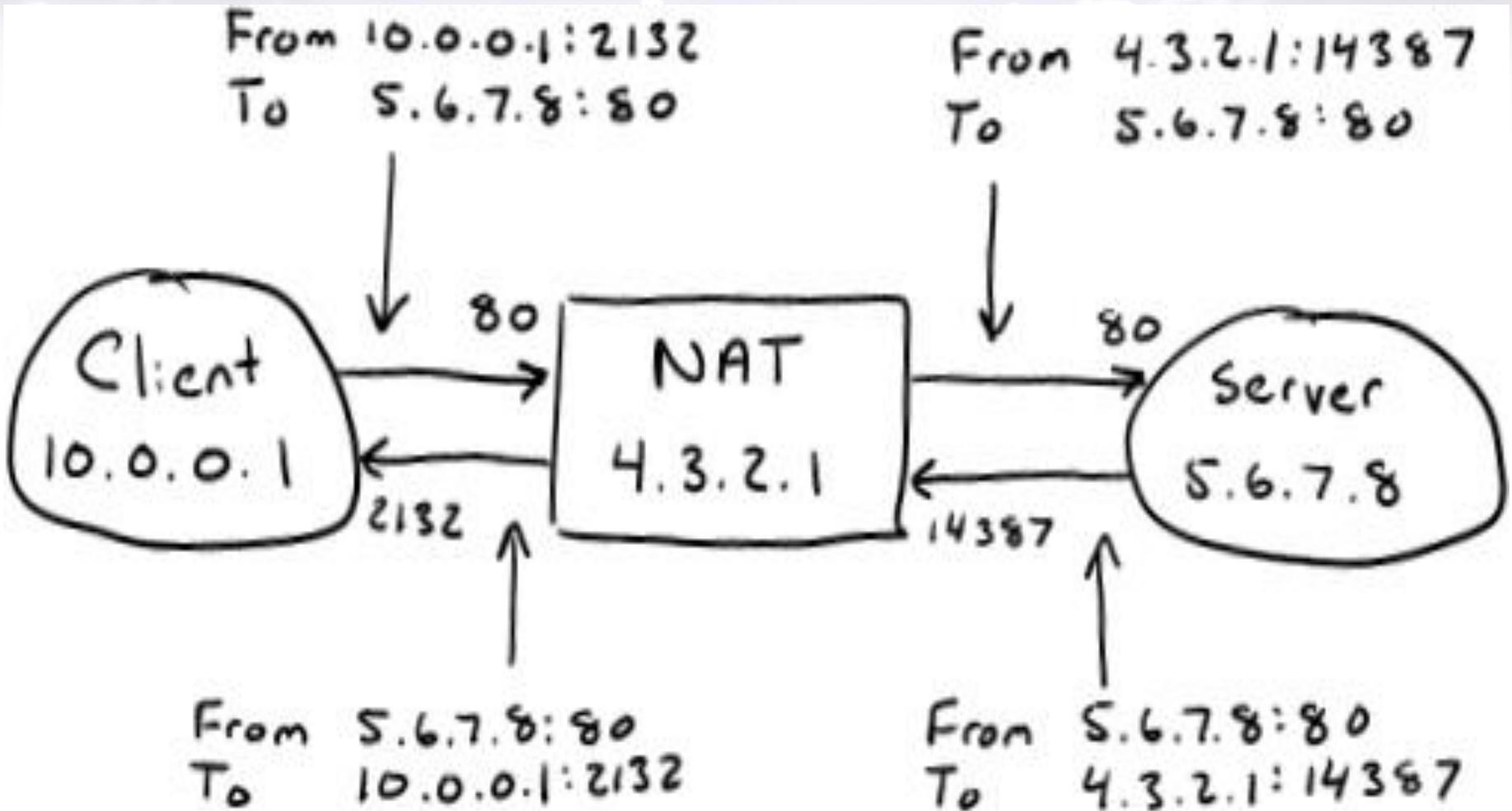
NfSen tutorial

Tramite l'utilizzo di NfSen
vogliamo evidenziare
quanto sia semplice
ricavare l'ip interno al NAT della macchina incriminata
con quei pochi dati a disposizione
senza dover passare l'antivirus
o leggere i log
su tutti i client di quella rete

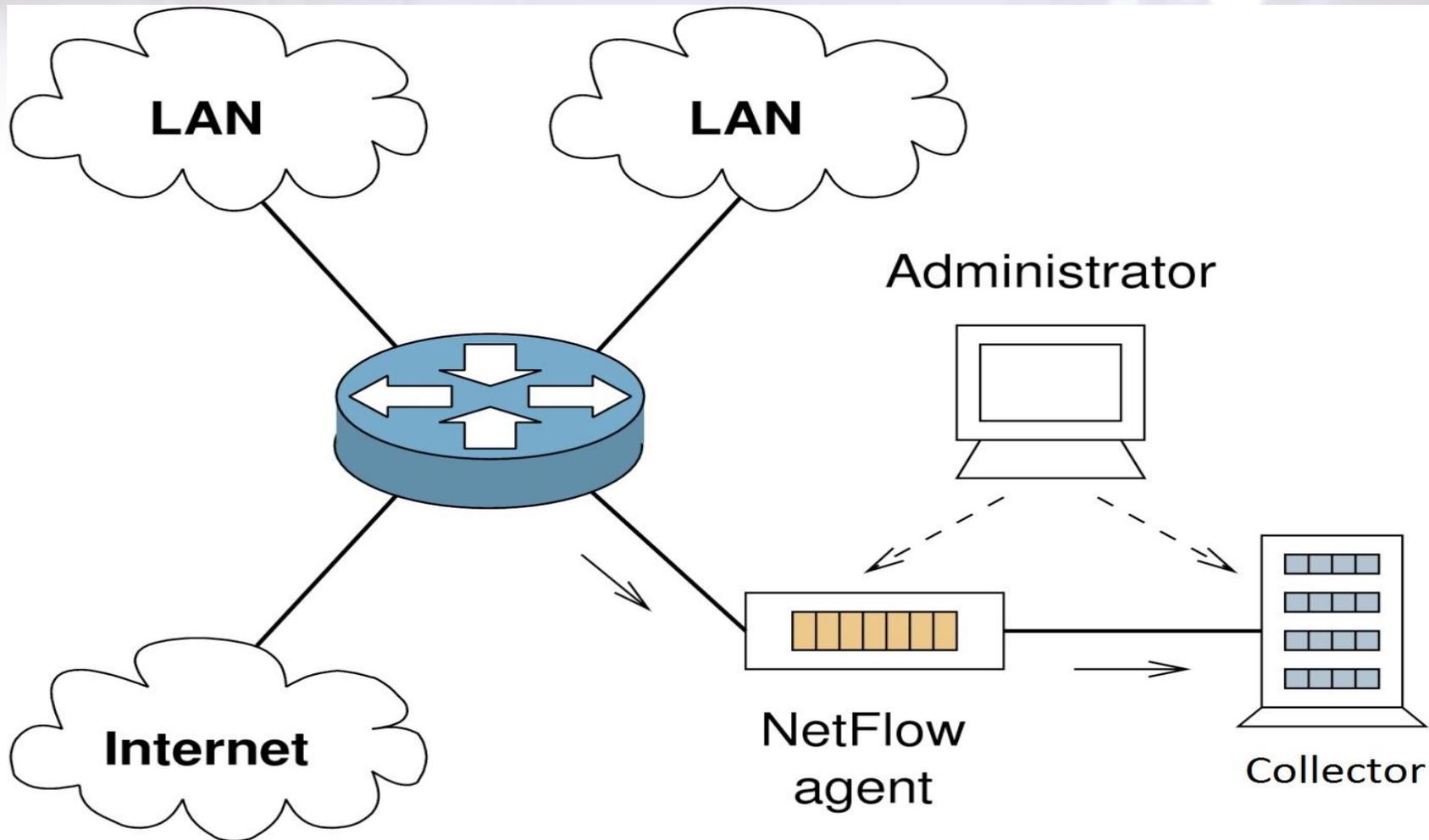
NfSen Architettura



NAT Architettura

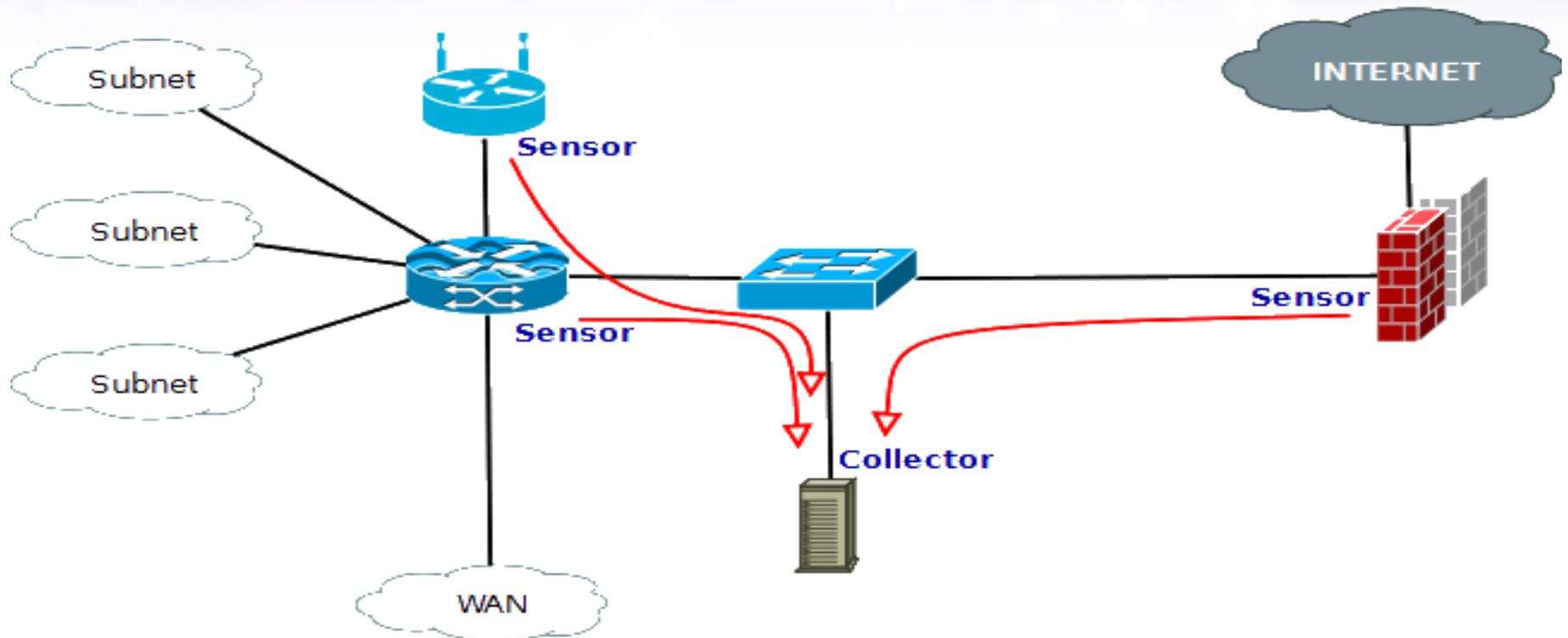


Architettura NfSen per NAT



NfSen ambiente misto

Netflow - Deployment Diagram



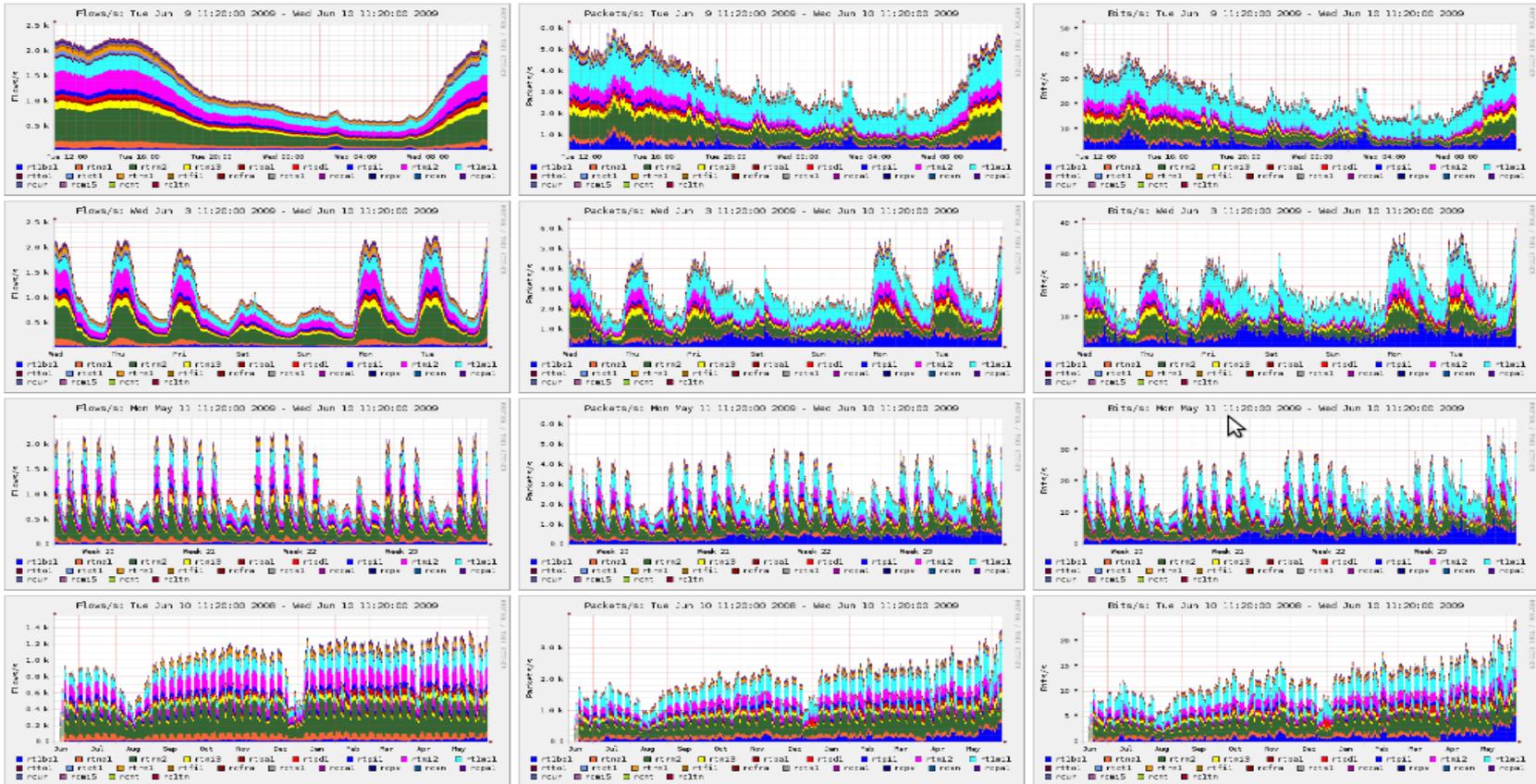
Legend

 netflow export from sensor to collector

Risultato

Home Graphs Details Alerts Stats Plugins live [Bookmark URL](#) Profile: live ▾

Overview Profile: live, Group: (nogroup)



Tool disponibili

Collector:

- flow-tools
- Stager
- Ntop
- Nerd
- nfdump/NfSen

Exporter:

- flowprobe
- NetFlow iptables module
- nProbe

Scelte operative

Collector: suite nfdump/nfsen

- Supporto NetFlow 9 (IPv6, MPLS)
- Tool da linea di comando per la collezione e l'analisi dei flussi (nfcapd, nfdump)
- Numerose utility per la gestione dei flussi (ri-esportazione dei flussi, cancellazione dei flussi più vecchi, conversione da altri formati, etc.)
- Possibilità di anonimizzare i flussi
- Sistema di plugin per l'estensione delle funzionalità
- Interfaccia grafica
- Software in continuo sviluppo (mailing list attiva!)
- Sviluppato da SWITCH.ch ; licenza BSD license

<http://nfsen.sourceforge.net/>

Scelte operative

Exporter di flussi dal NAT server: nProbe

- Affidabilita' altissima anche per carico pesante sulla rete
- Altissima stabilita'
- Altissime performance anche sotto stress
- sviluppato dal prof. Luca Deri, dell'Universita' di Pisa (ai tempi)
- OpenSource (per Universita' ed Enti di Ricerca)

<http://www.ntop.org/nProbe.html>

Requisiti minimi

CPU

- Minima: pentium 4 > 3Ghz
- Consigliata: bi o quadri multicore cpu > 2Ghz

RAM

- Minima: 2GB
- Consigliata: 4GB

Dischi di alta capacità e ad accesso veloce

- Da 40MB a 1.6G al giorno di dati per un router di trasporto

Dipende dal numero di client dietro il NAT!

Architettura nel tutorial

nattina: macchina linux che fa da NAT ad una sottorete di due macchine: zm01 e zm02 che stanno dietro NAT

nattina:

eth0 → interfaccia verso l'esterno (10.0.2.15 DHCP di VirtualBox)

eth1 → interfaccia verso la rete interna (10.0.100.1)

zm01: linux 10.0.100.2

zm02: linux 10.0.100.3

nattina agisce come un router NAT (iptables MASQUERADE)

nattina esporta i pacchetti NetFlow delle sue due reti verso il collector (nProbe localhost:1000x)

nattina=il collector (conserva i file del traffico + interfaccia web NfSen)

Installazione NfSen

Nfdump su nattina:

Via distribuzione: es. `apt-get install nfdump`

Via sorgenti:

es. `./configure --enable-nfprofile && make && make install`

Nfsen su nattina:

- Prerequisiti: PHP, Perl, RRDtools, Nfdump con «`--enable-nfprofile`»
- Installazione da sorgenti: copiare `nfsen-dist.conf` in `nfsen.conf`
- Modificare `nfsen.conf` definire gli exporter in `%sources` e i path
- Lanciare `./install.pl </path/nfsen.conf>` **FINE!**

Configurazione NfSen 1/2

Nfsen.conf

\$BASEDIR # directory base della suite

\$BINDIR # directory dove si desiderano i binari

\$HTMLDIR #root del webserver

\$PROFILESTATDIR #dove devono essere salvati i dati

\$back-end_PLUINGDIR #directory che ospita i plugin di back-end

\$FRONTEND_PLUGINDIR #directory che ospita i front-end plugin

\$SUBDIRLAYOUT #importante per motivi prestazionali

0 default

1 anno/mese/giorno

2 anno/mese/giorno/ora

@plugins #array per l'associazione tra il profilo e il plugin installato

Configurazione NfSen 2/2

Definizione delle fonti da cui mi arrivano i dati NetFlow da collezionare e analizzare (sempre dentro nfsen.conf):

```
%sources = (  
    'ext' => { 'port' => '10000', 'col' => '#0000ff', 'type' => netflow },  
    'int' => { 'port' => '10001', 'col' => '#ff0000' },  
);
```

Rilanciare: `./install.pl ./etc/nfsen.conf`

nProbe

Installazione nProbe (solo su nattina!):

Scarico dei sorgenti da www.ntop.org

```
gzip -d nprobe-xxx.tgz  
tar xvf nprobe-xxx.tar  
./autogen.sh && make && make install
```

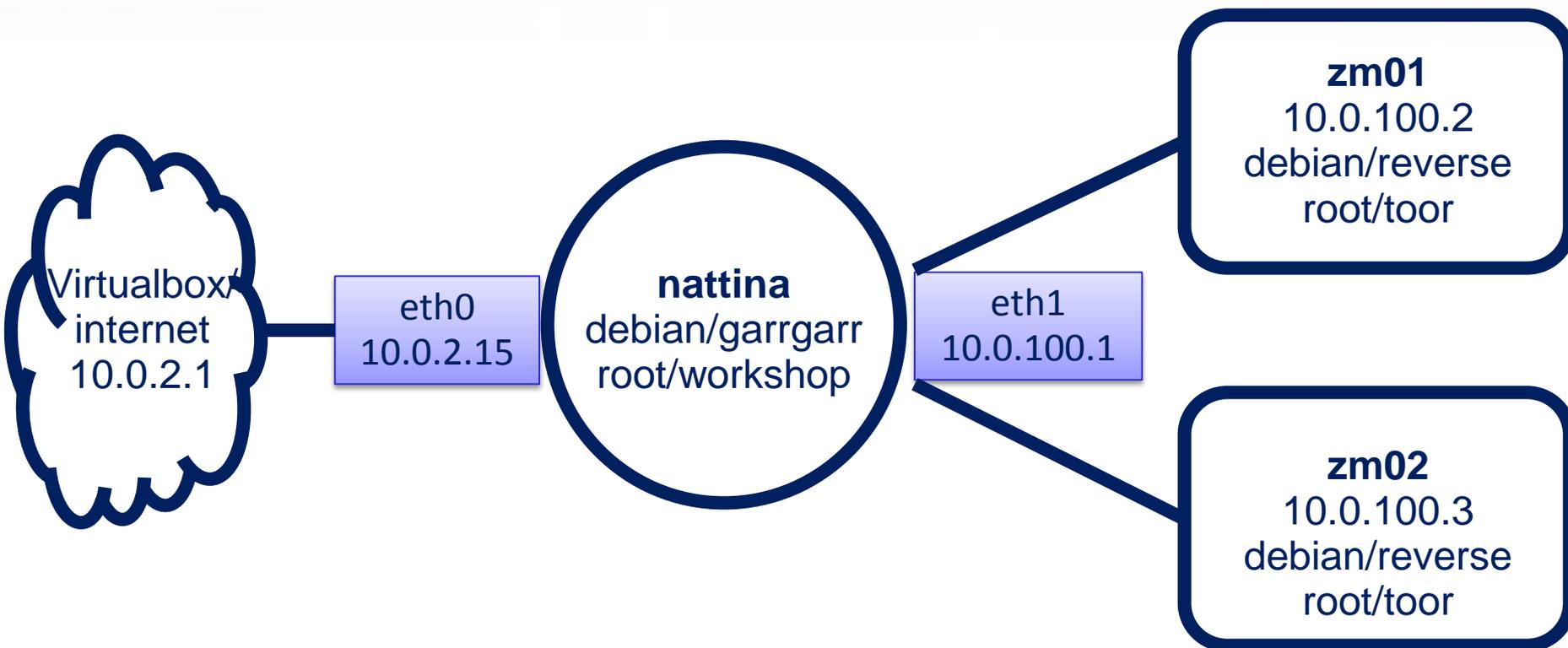
Comandi d'esportazione:

```
nprobe -G -i eth0 -n localhost:10000  
nprobe -G -i eth1 -n localhost:10001
```

Vanno fatti partire al boot della macchina (es. rc.local)

Esercitazione

Scenario



Esercitazione

Il GARR-CERT vi segnala TRE incidenti sull' IP 10.0.2.15 (l'ip «pubblico» del NAT server):

- Scanport
- ssh probe
- DoS

Le possibili vittime sono 192.84.145.55 o 192.84.145.67

L'esercizio consiste nel capire quale macchina dietro a NAT ha fatto cosa, in modo da curare e chiudere l'incidente

argus

