Port evolution: a software to find the shady IP profiles in Netflow. Or how to reduce Netflow records efficiently.

Gerard Wagener*

41, avenue de la Gare L-1611 Luxembourg Grand-Duchy of Luxembourg

*Electronic address: info@circl.lu; URL: www.circl.lu

Contents

I. SCOPE

Netflow records[1] are frequently used for accounting purposes in large Networks. Most router are capable of exporting Netflow data. In CIRCL's Netflow research program we collaborate with partners having large networks willing to exploit Netflow data for monitoring their infrastructures regarding information security incidents. The following objectives are addressed:

Validate Received Information Incident response teams or abuse handling teams receive information about incidents such as compromised hosts in their networks. Incident related information is sometimes volatile and quickly outdated. Therefore, it is essential to quickly validate received information. Netflow data can be used to validate this kind of information.

Measure the Impact Incident response teams or abuse handling teams including the technical support teams have often limited resources and some tasks must be prioritized. In addition,

abuse handling teams have multiple choices of actions. A harsh action for instance is to null route a host in a network. In order to facilities the choices, Netflow data often helps if it is presented in a proper way.

Identify Victims Netflow data can frequently be used to identify victims that were targeted by compromised hosts in the monitored network.

Detect Incidents Compromised systems are frequently reported by third parties. However, Netflow data can also be used to detect them in a proactive way.

In CIRCL's Netflow research program CIRCL technically support their partners to develop practical and customized solutions with the interest to improve the overall information security. **IP addresses were randomized with Cryptopan**[2].

II. INTRODUCTION

The system *Port Evolution* is a practical outcome of research and development activities on Netflow records. *Port Evolution* emerged after numerous failures of off-the-shelf solutions in order to index and process Netflow data. Due to the high volume of Netflow records, relational databases including numerous NoSQL solutions failed to handle the Netflow data in near-real time. Therefore, *Port Evolution* is a home grown solution to handle the Netflow data having a delay of 10 minutes. This delay is due to the partition of Netflow data in 5 minutes blocks, transfer, import and processing times. This delay has been observed for an average of 35000 flows per seconds on commodity hardware. The delay can be reduced on special and dedicated hardware.

III. PORT EVOLUTION MODEL

Port Evolution is operated in the Autonomous System (AS) having the ASN (AS Number) A. In this AS the set $ALL^* \subseteq \{0, 1, ..., 65535\}$ contains the ports that are monitored by *Port Evolution*. The following **features** are directly extracted from Netflow records:

bytes Number of bytes transferred.

durations Durations of the flows.

packets Number of packets transferred.

src as The AS of the source IP in a Netflow record.

dst port The destination port of a Netflow record.

The **feature** num_peers is calculated by *Port Evolution*. The num_peers corresponds the unique number of IP addresses in a given block, the IP address contacted. A Netflow block contains all the encountered Netflow records within a period of 5 minutes. The *feature* $flow\_id$ is created by *Port Evolution*. It is a linear counter for each block denoted B. When the word feature references a feature listed above including the term num_peers, the word feature is highlighted in **bold**. A netflow record is modeled as a tuple which is included in a block. Hence, B=$\{(flow\_id, bytes, durations, packets, src\ as, dst\ port, ...), ...\}$. Netflow records may also include more information. The minimal set of **features** F is $\{bytes$, $durations$, $packets$, $src\ as$ and $dst\ as$, $num\_peers\}$. Each block is processed and all the netflow records are preprocessed. During preprocessing the flows are matched according the definition 1. The matched flows are included in the set $\hat{B} \subseteq B$. A more informal definition is: *An IP address which belongs to the AS* A *which communicates with an IP address on a monitored destination port is matched.*

$$\forall(..., src\ as, dst\ port, ...) \in \hat{B} \Leftrightarrow src\ as = A \wedge dst\ port \in ALL^* \tag{1}$$

On each **feature** the following **operations** are performed. When those are references the word operations is highlighted in **bold**.

sum The sum of a given **feature** per day.

max The maximal value of a **feature** encountered in a 5 minutes block per day.

min The minimal value of a **feature** encountered in a 5 minutes block per day.

IV. MAIN PAGE

Once a user connects to the port evolution system, a menu is shown in the left side and various tables are shown in the center. The menu is called `Host Query Interface` and is shown in figure 2. Each table on the right side contains the top 10 IP addresses that have the highest score regarding a **feature** and a given **operation** for a given day. Another day can be selected

by using the date picker by clicking in the date field which is under the text `Specify other date`, shown in figure 3. The day selected is written above the tables. The score is computed according the selected **operation** which is also shown above the tables.

The top 10 scores for a given **feature** are grouped together regarding a port. Next to the score, sometimes a red filled circle emerges. This circle contains a number. This number corresponds to the number of meta-data[3] available related the IP address next to it. The meta data can be displayed by putting the mouse over the red circle.

An example is shown in figure 1. The presented top 10 scores are related to the day 2014-01-28. The *sum* is the used operation as it is written in the second line. The first large row describes the top 10 addresses communicating on port 25 with machines in other AS. The row contains 4 nested tables. The first nested table has three captions called `bytes`, `score`, `@`. The first column having the title `bytes` contains the top 10 IP addresses having exchanged the highest volume of traffic on port 25 with machines in other AS. The volume is expressed in bytes and written in the column entitled `scores`. The last column includes the number of available metadata. The version of *Port Evolution* described in this document was fed with anonymized data and no meta data was available for the randomized IP addresses. Similar nested tables for the monitored ports enumerated in table I can be observed in the next large rows.

V. HOST QUERY INTERFACE

On the main page only the top 10 IP addresses are shown. However, sometimes it is necessary to investigate an IP address that does not emerge in the top 10 list. This IP address can be entered in the `Host Query Interface` shown in figure 2. The IP address is entered in the field denoted *Host*. A port can selected in the *Port* combo box and an operation can be selected in the *Operation* combo box. In case the port *ALL** is selected, the ports enumerated in table I are considered. The service is assumed by the port number because no payloads are included in Netflow data. Above the *Host Query Interface* is a link called *Overview*. When this link is accessed, the main page is displayed.

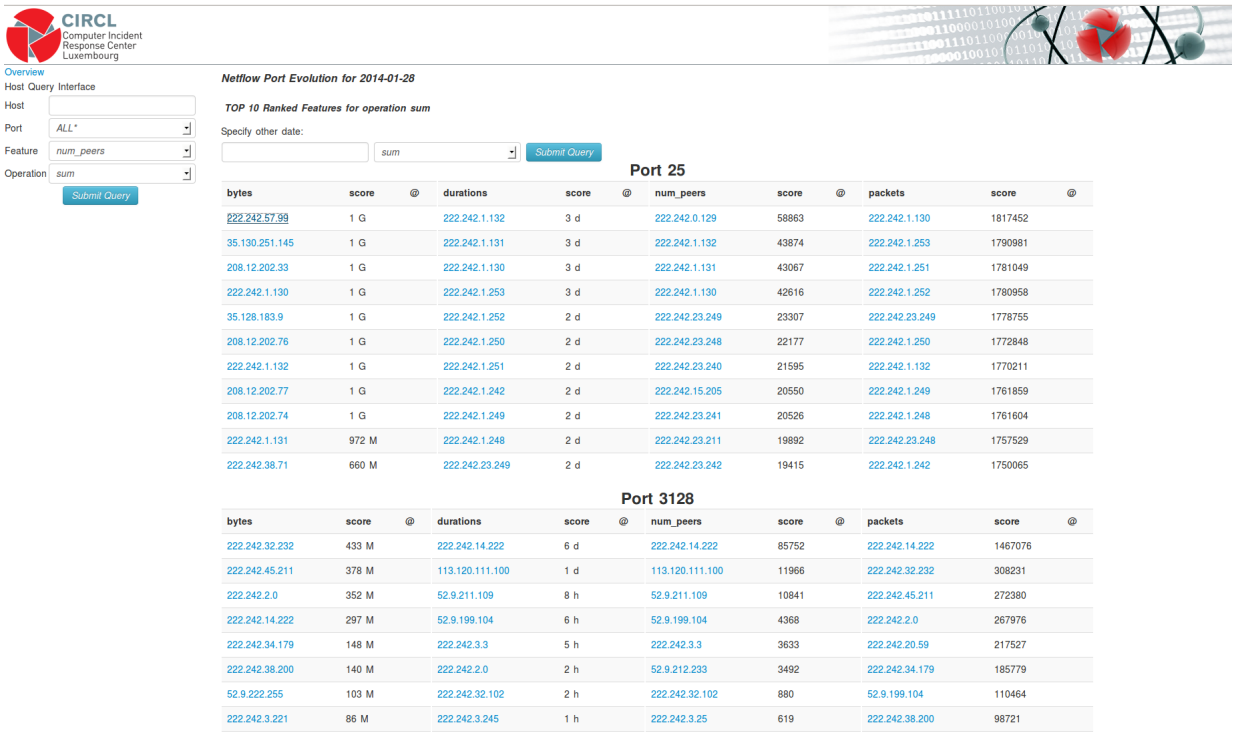

FIG. 1: *Port Evolution* - Overview

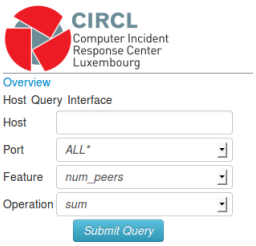

FIG. 2: Host Query Interface

VI. PORT EVOLUTION GRAPH

The Port Evolution graph can be assessed in two ways. Each IP addresses listed on the main page are links. When these links are clicked, the port evolution graph is shown. The alternative to see the graph is to pass by the `Host Query Interface`. An example of a Port Evolution graph is shown in 4. The selected IP address, the selected **feature** and the selected **port** is shown in the title above graph. The used operation is **sum** and the selected **feature** is **bytes**. On the x-axis, the time is presented in days. The time scale can be interactively changed by dragging the selected area with the mouse. The y-axis describes score of the **feature**. A logarithmic scale

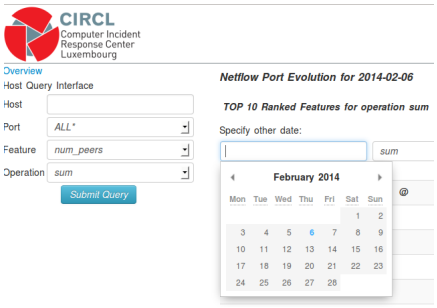

FIG. 3: Select an older date to get the top 10 addresses regarding this date.

Port Number	Assumed service
21	FTP
22	SSH
25	SMTP
80	Web
445	Netbios
1194	OpenVPN
3128	Squid proxy
8080	Webproxy
8081	Webproxy
8118	Privoxy
8123	Polipo web proxy
9050	Tor

TABLE I: Monitored Ports

is used on the y-axis and the unit is **feature** dependent. In the shown example the **feature** is the number of bytes.

If a top 10 address link is followed in the main page, then the `Host Query Interface` is automatically updated with the IP address. However, the option **ALL*** ports is preselected. In practice this is usually the next step which is done. For instance, if an IP addresses appeared in the top 10 list related to port 25, then just a click on the `Submit Query` button must be made to get to the results. However, the disadvantage of this **feature** is that the pre-filled values of the `Host Query Interface` does not necessarily match the graph which is shown next to it.

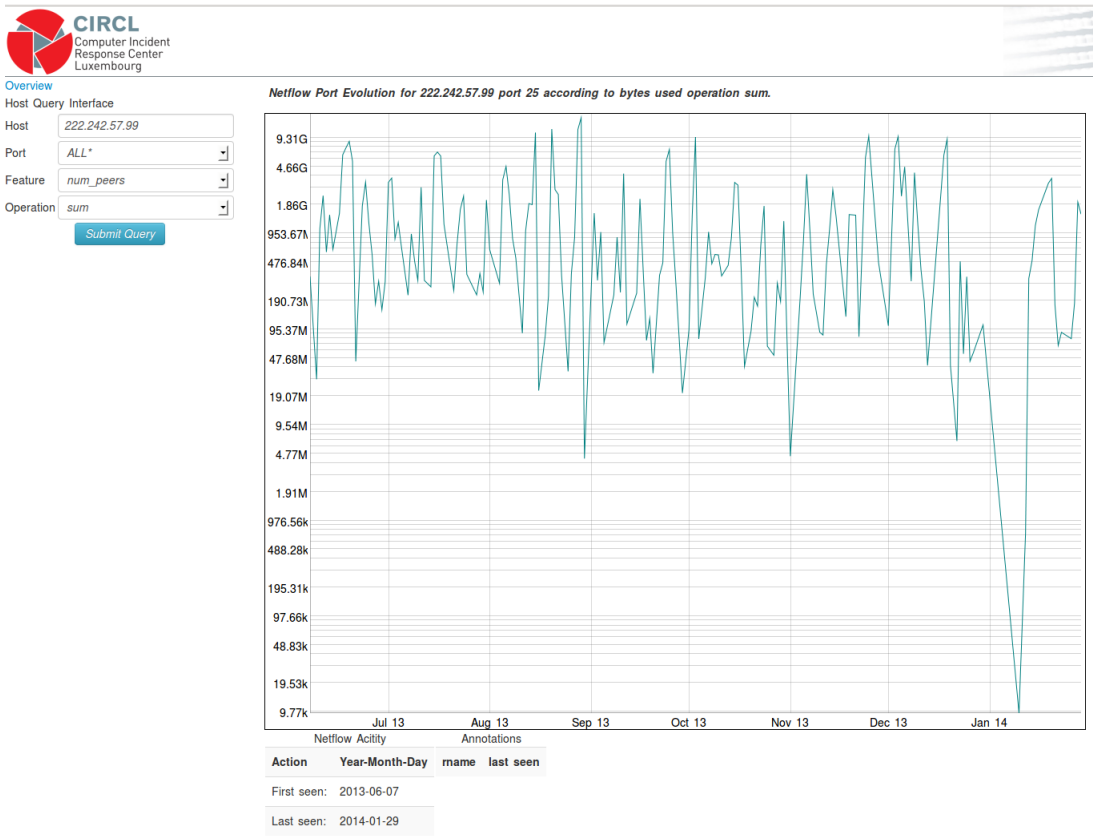

FIG. 4: Example of a Port Evolution Graph

Therefore, the title of the graph must be carefully read.

Under the graph, two tables are shown. This first table is entitled `Netflow Activity`. This table contains the two rows. The first row shows a date where an IP address first appeared in the Netflow records. The second row shows when the IP address latest appeared in the Netflow records. The current *Port Evolution* data structures can hold approximately 6 month of data. In this case 10^6 IP addresses are monitored which corresponds to 10^6 ranked sets holding the top ten scores. The data structure is bound to 32 GB in the current installation but could be increased. The old data is periodically removed by the *Port Evolution* based on a configuration setting. Therefore, it might be that an IP address has been known before the date shown in the `First seen` field.

The second table is entitled `Annotations`. This table contains additional Information about a given IP address. Currently, rnames from passive DNS are included. Annotations data might be frequently outdated. Due to the IPv4 exhaustion customer changes regarding IP blocks might

Host	222.242.51.194
Port	25
Feature	bytes
Operation	sum

TABLE II: Example of the Host Query Interface

change frequently. Therefore, the annotation data also includes a last seen field.

VII. EXAMPLES

- How many bytes the IP address *2*22.242.51.194 exchanged on port 25 over the last months?

  The data shown in table II is entered in the `Host Query Interface` and the `Submit Query` button must be pressed. The result is shown in figure 5. The outgoing traffic from the host *222.242.51.194* on port 25 started on the 22th December 2013 and ended the 15th January 2014.

  A higher version than 2.0 of *Port Evolution* is required to extend the machine profiles on non monitored ports distributions. The currently documented *Port Evolution* version only tracks a limited set of outgoing ports.

- Which other ports the IP address *222.242.51.194* contacted?

  Port Evolution remembers the last entered IP address. Therefore, just the port denoted `ALL*` should be selected and the `Submit` button should be pressed.

- Looking at the figure 1, the IP address *222.242.57.99* is at the top position of hosts sending out the most of SMTP traffic in the AS. Was it always like that? Or was the machine abused?

  A click on the IP address gives the evolution of outgoing SMTP traffic for this host for the last months.

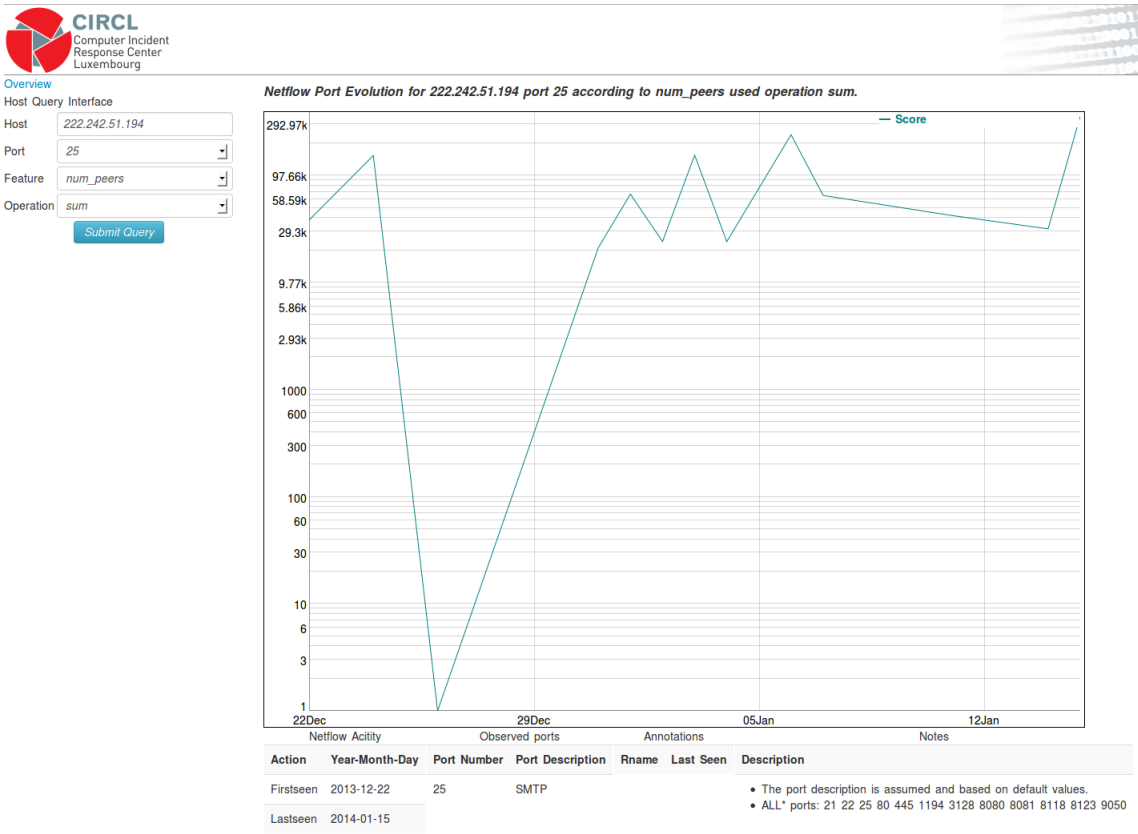

FIG. 5: Pattern of a compromised server being abused to send SPAM

VIII. USE CASES

A. Spammers

The initial goal of *Port Evolution* was do detect hosts that were abused to send SPAM. The assumption was made that SPAM is sent from hosts within the monitored AS, to hosts in other ASNs on port 25 using the protocol TCP. An example is shown in figure 5. The x-axis shows the time and the y-axis the unit of the selected **feature**. After the 26th December 2013 the number of peers increased and reached the peak of 355839 the 15th January 2014. Such a high number of peers within a day is a good indicator to identify spammers as their goal is to send a maximum of mails within a short time.

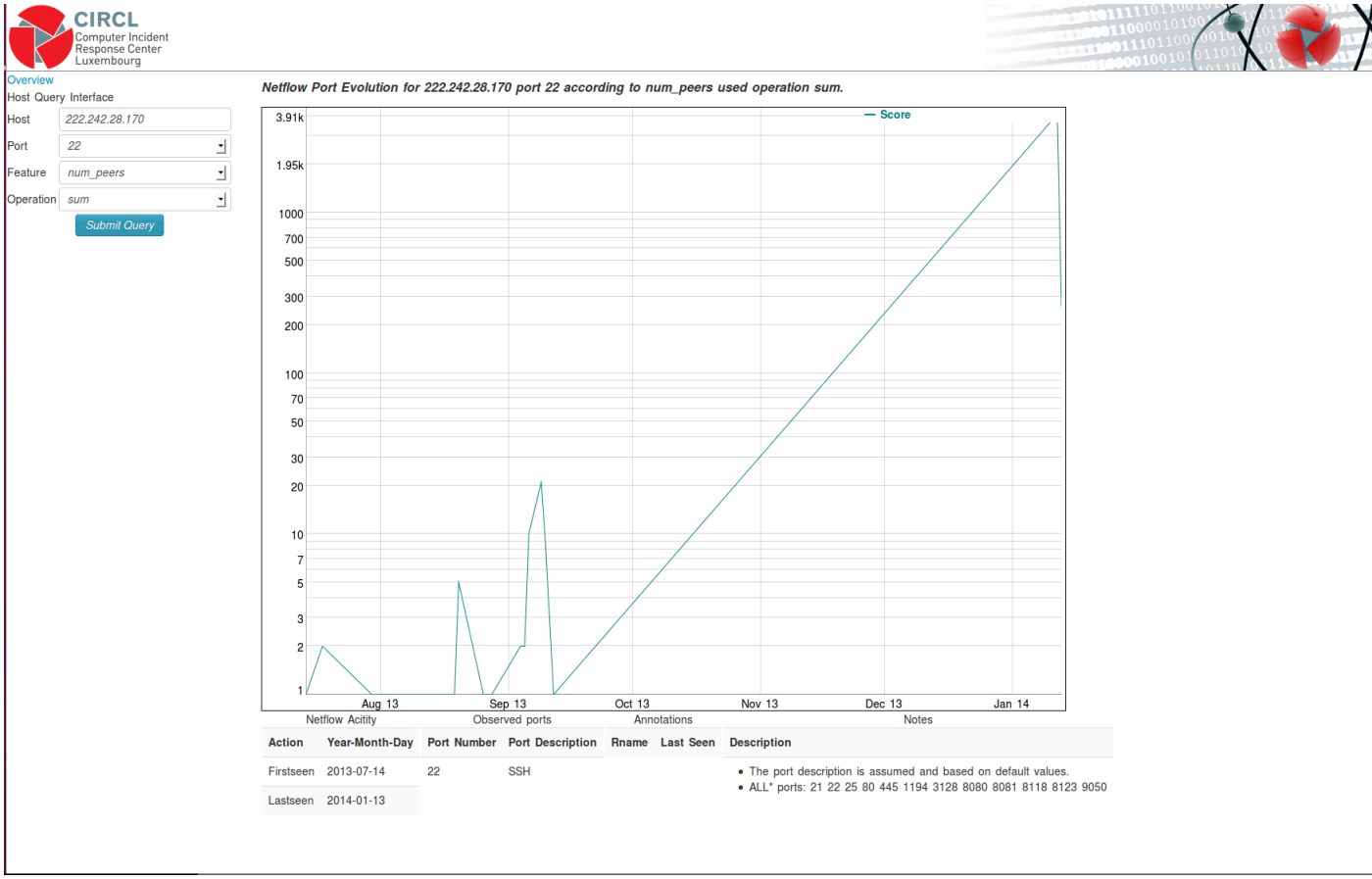

FIG. 6: Host scanning Networks for discovering SSH servers

B. Aggressive Scanning

The port evolution of the host *222.242.28.170* is shown in figure 6. On the x-axis is shown the time expressed in days and on the y-axis is shown the score. The score represents the number of unique IP address encountered in a Netflow block of 5 minutes that were contacted by the IP address *222.242.28.170* on port 22. The host *222.242.28.170* first appeared the 18 July 2013 because it contacted another peer on port 22. The outgoing SSH traffic slightly increased until the 9th September 2013. On September 12 the host contacted at least 4096 peers. The peer data structure for an IP address can hold at maximum 4096 peers, so its it not known if the host contacted exactly 4096 hosts or more. However, it seems to be sufficient to detect scanning activities within a day. The increase is shown by a straight line due to the logarithmic scale on the y-axis.

C. Tor Nodes

Deployed Tor exit nodes are generating by definition a lot of outgoing traffic. The outgoing traffic is usually controlled by the Tor exit policy [4] that is configured by the local Tor operator. For instance, if the exit policy only allows outgoing port 80 TCP and outgoing port 443, then the network profile of this exit nodes looks very similar to an open proxy. Tor Exit Nodes usually participate in the Tor Network and thus often communicate with other Tor nodes on port 9050 TCP.

In figure 7, is shown an onion router that is participating in the Tor Network. The node was continuously active from the 20th October 2013 to the 26th January 2014. In order to see which kind of traffic this Tor node relays, it is sufficient to click on the *Submit* button in the *Host Query Interface*. The result is shown in figure 8. The graph looks quite overloaded because it contains 12 curves. The host operating the Tor exit node, communicates with other hosts on the ports {21, 22, 25, 80, 455, 1194, 3128, 8080, 8081, 8118, 8123}. This means that either the exit policy allows these protocols and that these flows were originated from the Tor exit node or the system administrator installed other services on the same host or routed them through the same host. A *Port Evolution* higher than 2.0 is needed to compare incoming traffic on port 9050 and outgoing traffic. In this case incoming traffic distribution on port 9050 can be compared with the others outgoing ports.

---

[1] http://www.ietf.org/rfc/rfc3954.txt

[2] http://www.cc.gatech.edu/computing/Networking/projects/cryptopan/

[3] e.g. passive dns data, reverse ptr data

[4] https://trac.torproject.org/projects/tor/wiki/doc/ReducedExitPolicy

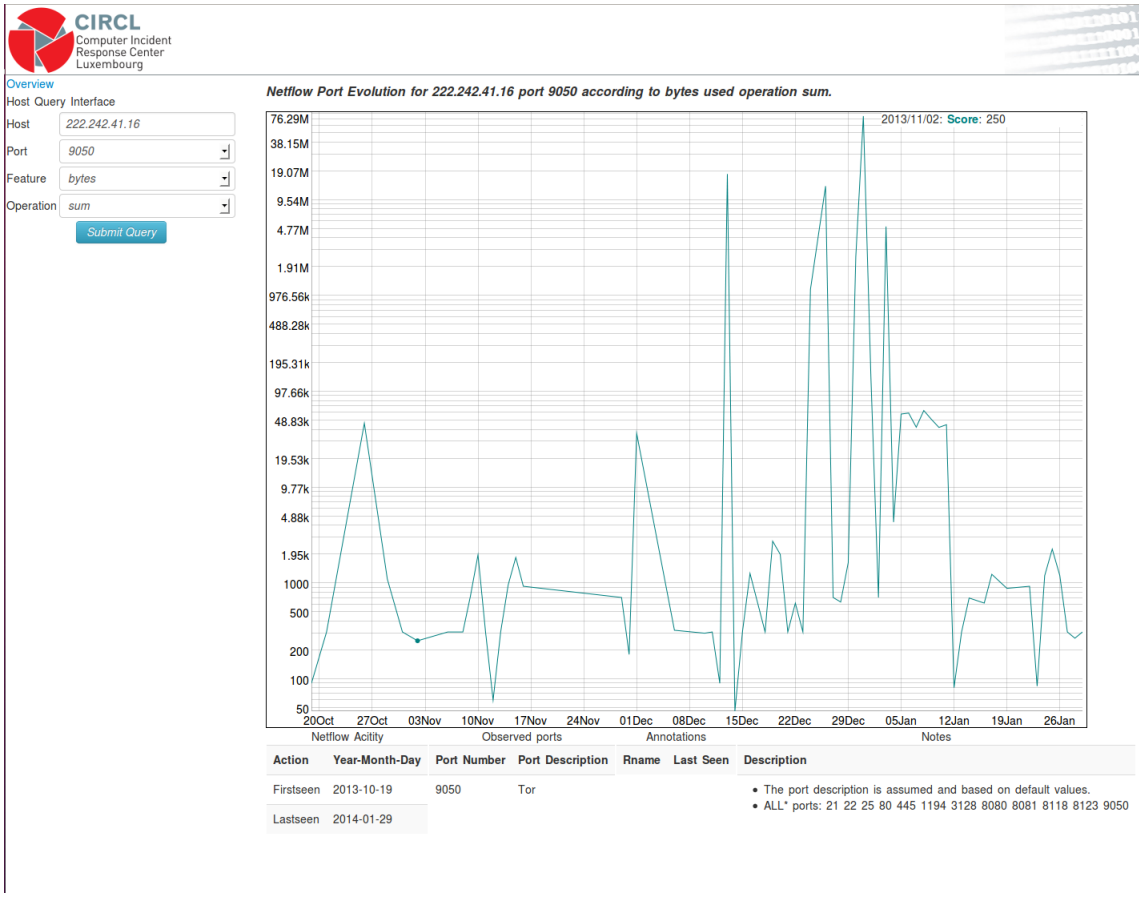

FIG. 7: Onion Router Traffic

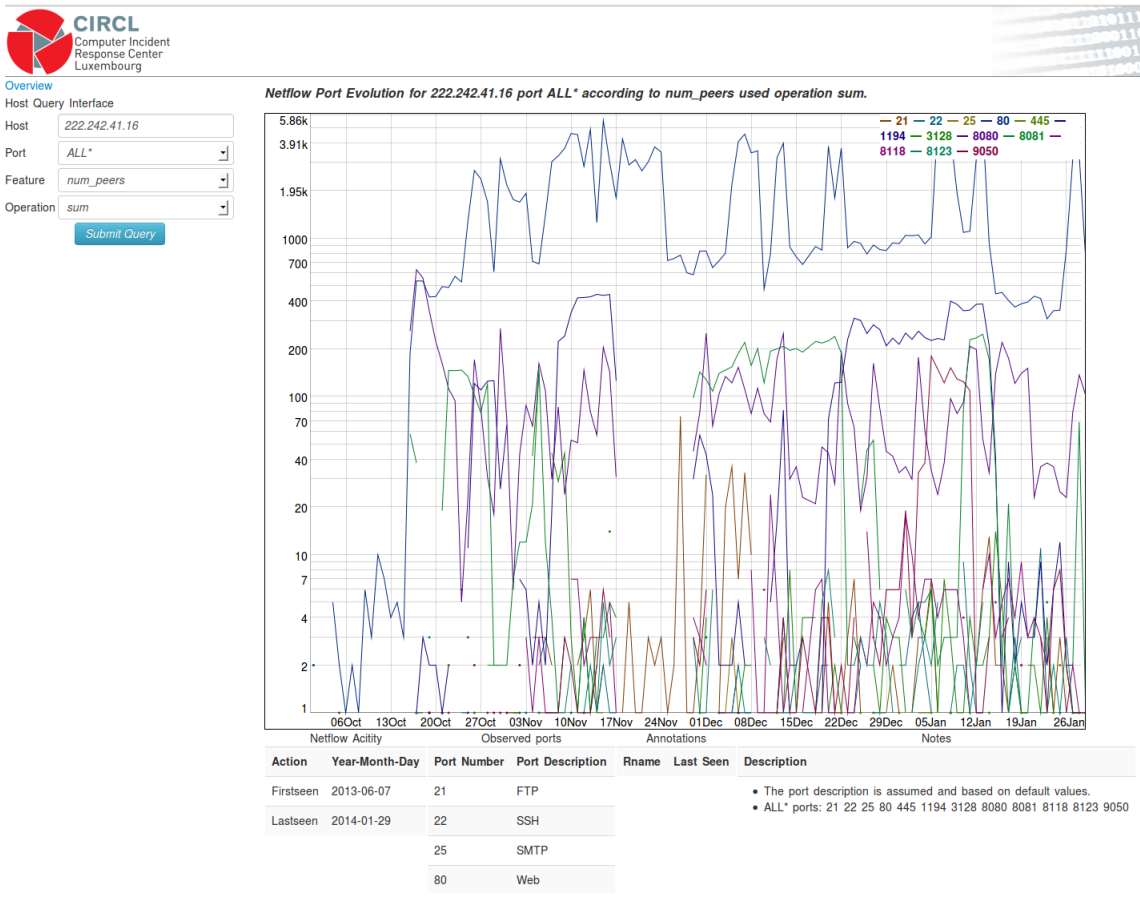

FIG. 8: Tor Exit Node Traffic