Analyzing Outbound Network Traffic

Mirosław Skrzewski

Politechnika Śląska, Instytut Informatyki, Akademicka 16, 44-100 Gliwice, Polska
miroslaw.skrzewski@polsl.pl

Abstract. Conventional security solutions monitor network communication without paying much attention to outgoing traffic, due to high processing cost of packet level network traffic analysis. Outgoing network communication, originating from typical system's application has common properties, which can be used for traffic selection in security related analysis. The paper presents the concept of outbound network traffic classification based on temporal characteristics of network flows and shows the results of experiments identifying traffic patterns of common user's application and values of classification parameters.

Keywords: network flow patterns, web flow, application traffic analysis.

1 Introduction

Conventional security solutions attempts to monitor network traffic, trying to identify and block any signs of malicious intended communication. Typically these systems analyzes mostly incoming traffic as potentially dangerous, paying less attention to outgoing, user initiated communication, assumed as secure from definition. This paradigm of security operation changes in last years due to arrival of stealth malware programs (trojans, bots) capable of exporting user's data over network and so the properties of outbound network traffic became the subject of research interest.

Most of the present client-server applications utilize HTTP protocol for request – result delivery, due to security imposed network communication constrains. Typical LAN access router/firewall configuration allows for incoming communication on ports 80 and 443 and blocks nearly all other lower TCP ports. Therefore many application use web browser as their user interface (mail, messengers), and others (voice, stream) use http ports for communication initialization.

The HTTP traffic, due to origin from typical applications, has some common properties on TCP connection or flow levels, so there should be possible to derive patterns of typical applications traffic, and use them to separate user initiated communication from remaining network flows, potentially requesting more attention from security monitoring systems. The paper presents results of analysis of outbound network communication for some typical applications and resulting models of network flow traffic, aimed on selection of user initiated packet flows in network outbound communication.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2011, CCIS 160, pp. 204–213, 2011.

2 Related Work

Since the mid-1990s a lot of works analyses properties of HTTP network traffic. Its authors analyze the content of downloaded web pages, numbers of requested objects, amount of transferred data and on this basis attempt to derive statistical models of web related traffic and describe requirements on network throughput. Most of this works [1,2,3,4,5] were done on the network edges, on the campus external connections, and attempt to describe network traffic properties on the basis of relatively small traffic samples (of up to few hours duration) recorded on network access links.

Recorded traces of web communication were analyzed applying different models of web interaction, based on *Page-request* (single web page transfer) or *Web-request* (one or multi-page transfer in response to the user action) models. A set of parameters describing details of analyzed data exchange were defined for simulation models of web traffic and best fit probability distributions for model parameters were presented. The values of parameters were based on the analysis of communication on HTTP or TCP level, and in [4] also on Web-flow level.

The levels of analysis were described as follows: HTTP conversation represent single HTTP request and response transaction. In early (HTTP 1.0) times request and response pair use single TCP connection, from HTTP 1.1 due to persistent connection more then one pair of request and response may be transported in TCP connection (network flow). HTTP communication may use many concurrent TCP flows on the link, initiated by the same or different hosts.

A Web-flow was defined [4] as a group ot TCP flows with the same source IP address with parameters *TCP connection count* (number of TCP connections in Web flow) and *Interval of TCP connections* (interval between the starts of two consecutive TCP connection in a Web flow).

The notion of network flow is often used in monitoring of network usage, where it simple means IP data stream. There are two versions of network flow definition, one originating from Cisco and others vendors (NetFlow) says [6]: "a network flow is a sequence of packets between a given source and destination in one direction only" , the other used in *argus* program [7] ommits the words "in one direction only" so the flow is described by 5-tuple: IP source and destination addresses, source and destination ports and IP protocol number. This basic data are generally suplemented with timing (origin, closing time) and volume (number of transmitted packet, transmitted bytes) information.

3 System Network Activity Monitoring

Most of works devoted to HTTP traffic monitoring is based on data recorded at the network edges and due to amount of processing they are not well suited for real-time operation. To obtain the solution capable of real-time operation one must move the analysis near the source of the outgoing traffic – to user system, and base it on locally available traffic.

User systems generate a lot of outgoing network traffic induced by daily users activity, due to interaction with diferent types of web sites (informational, social, commercial etc.) or services (email, messengers, voip) and not containing any hazardous contents. It's full real time analysis on packet level may be very expensive in terms of computer power and processing time.

Instead we propose the concept of system network activity monitoring, based on analysis of network flows – network conversation between given pair of source and destination IP addresses/ports treated as a unit of communication. This approach is loosely modeled on bidirectional argus [7] flows. The flows, recorded locally on monitored system, with the additional information about the process and account responsible for communication and flow timing information are passed through origin analysis process, filtering out all users' application initiated traffic.

3.1 Data Collection

To capture network activity data we use custom program, recording communication events on Windows system's TDI (transport driver interface) kernel API (*TDIlog*). Acquired event's data recorded in text file delivers continuous picture of all system network activity – opening and closing of ports by active processes, reception and transmission of data on active UDP and TCP ports, source and destination IP addresses, with the amount and the direction of transmitted via given connection and port data. Each operation on TDI interface is recorded with suitable timestamp, so it is possible to compute diffrent statistics describing network communication of given application or system processes.

Our idea is to move with the flows recording near to the source of the outgoing traffic – to user system, and to use additional, locally available, information to classify outgoing traffic on user (aplication) intended communication, composing most of computer network activity, probably safe from security point of view, and on remaining, smaller part, requesting more detailed analysis.

We focused our attention on outbound communication, mostly user initiated outside of computer system. We don't want to analyze traffic contents nor traffic destinations, as this properties are constantly changing with user activity. We are searching for properties connected with communication algorithms embedded in application design, related to temporal relationship between application generated flows.

3.2 Developing Network Activity Patterns

To capture necessary flow data we record with *TDIlog* network activity of the test system with installed selected application. Because most of user's application operates in client-server model, with the usage of HTTP protocol, we focused our attention on the operation of web browsers, as the basic user tools. In our experiments we record flows generated by selected web browsers (Internet Explorer, FireFox, Opera and Chrome) opening the same, selected url's.

To have other variables of experiment constant, we installed all browsers on the same system, so the network operation background (operating system network activity) was identical in all tests. For comparison we also recorded operation of messenger communication (gadu-gadu messenger, skype), web mail applications (gmail) and the system's background network activity. We analyzed also recorded operation of database (PostgreSQL) client and CAD systems.

Selection of tested web pages may influence the test results, especially with limited page list, so we attempt to select diverse types of websites, situated in different distance from our system. The list of tested sites and their approximated localization are presented in Table 1. The test were run in a way similar to Webflow analysis model [4].

Selected url was activated in chosen browser with one mouse click and all network activity connected with presenting the content of the web page were recorded. After some pause (1–3 minutes) the test was repeated with the other browser or the next url. We start all browsers before tests and don't close them during the test because of various network activity of the starting browsers (contacting search engines, company web site, restoring visited pages etc.) adding to recorded test traffic.

Recorded data were inserted into database and a set of queries were run to derive the TCP flows. Exemplary picture of user's network activity is portrayed on Fig. 1 as a plot of consecutive net flows in time. A group of net flows with small flow start to flow start span compose *Web-flow*, a unit of network activity

Table 1. Tested websites and their localization

Website url	Localization	Protocol	Comments
`cn.polsl.pl`	on campus	http	
`cordis.europa.eu`	Brussels	http	portal
`www.techrepublic.com`	San Francisco, California	http	portal
`gmail.com`	Mountain View, California	https	webmail

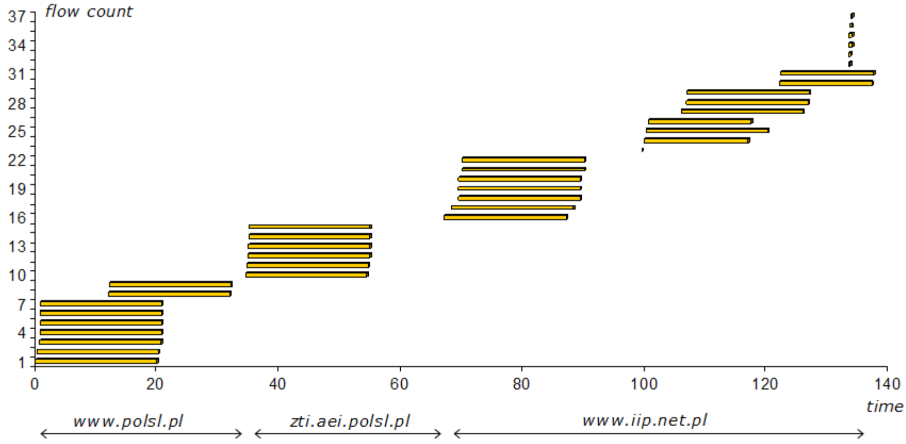


Fig. 1. Example of browser web-flow activity

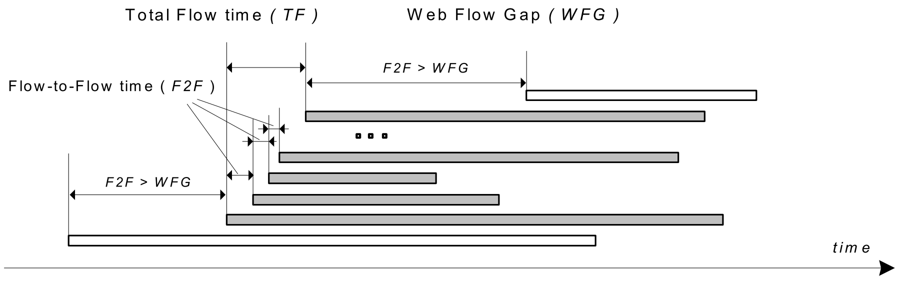


Fig. 2. Web-flow and net flows relationships

responding to user action. Detailed definition of parameters we use for temporal flows characterization is presented on Fig. 2

TCP net flows were represented on the figure as the long narrow rectangles. Flows temporal characteristics depends on the values of interval between two next flow starts (*Flow-to-Flow* or *F2F*) times. A group of consecutive net flows with small F2F times constitute the *Web-flow*. Adjacent Web-flows are separated by F2F time greater then *Web Flow Gap* time (WFG). Duration of Web-flow or *Total Flow* time is equivalent to the sum of F2F times of N belonging net flows (grayed on Fig. 2).

The aim of our test was to find the values of F2F and TF times constituting Web-flows for given web sites and their dependency on the type of used browsers. From our test procedure the inter Web-flow interval (WFG time) will be forced to be greater then 10 to 30 seconds.

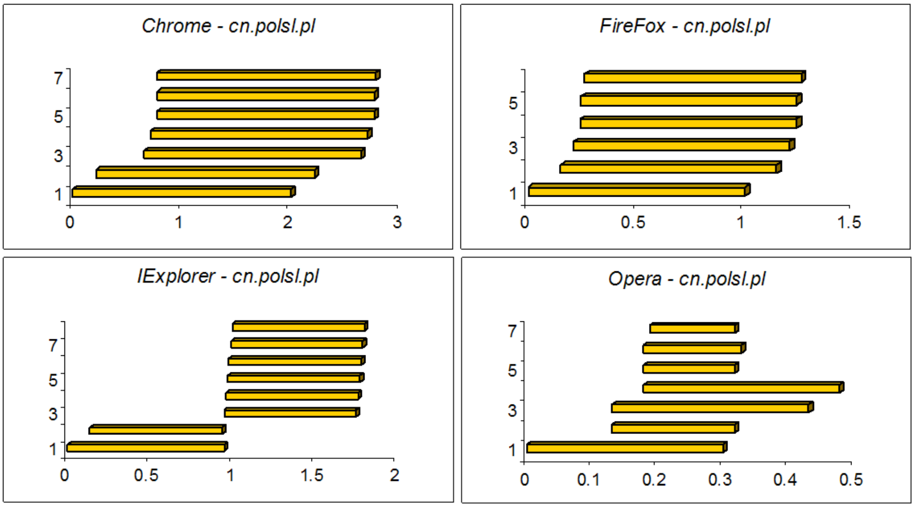
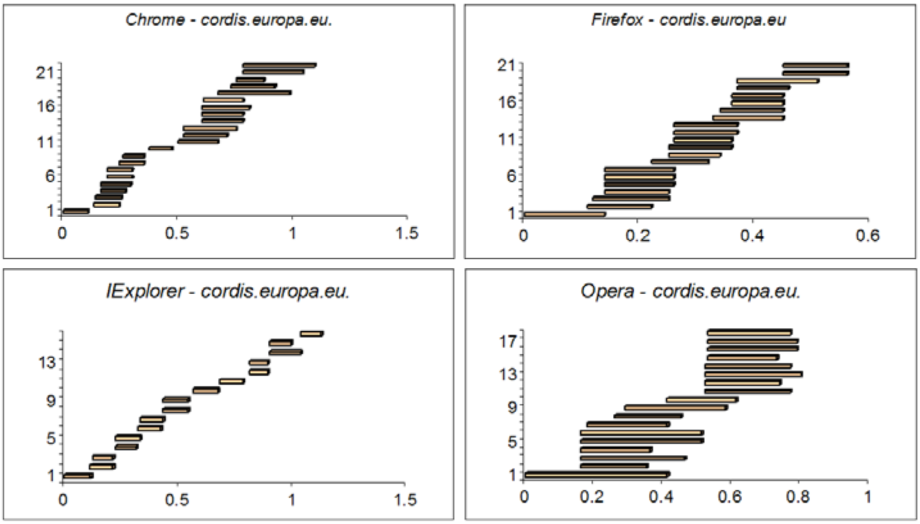


Fig. 3. Web-flow for cn.polsl.pl

3.3 Web Flow Analysis

Results of our test were presented graphically as plots of the same Web-flows obtained with selected browsers, side by side, and resulted temporal parameters of the Web-flows are presented in tables. The Fig. 3 presents the process of opening conference website `cn.polsl.pl`. The initial web page requires 6–7 net flows for completion. Nearly all plotted net flows were strongly shortened from original 3–5 minutes to 1–2 seconds, for plot resolution clarity. The Table 2 shows the values of flow parameters.



Chrome - cordis.europa.eu.

Firefox - cordis.europa.eu

IExplorer - cordis.europa.eu.

Opera - cordis.europa.eu.

Fig. 4. Web-flow for cordis.europa.eu

Table 2. Web flow parameters for page cn.polsl.pl

Parameters	Chrome	Firefox	IExplorer	Opera
F2F mean	0.130	0.209	0.143	0.0270
F2F variance	0.028	0.115	0.089	0.0024
TF time	0.781	0.261	1.001	0.1880
N (flows number)	7	6	8	7

Next figure presents the Web-flows related to the opening the link of `cordis.europa.eu` site located in Belgium. The download of page require many TCP connections. The Figure 4 presents the plots of Web-flows, the Table 3 contains derived Web-flow parameters.

The next set of flow plots (Fig. 5) presents the operation of `techrepublic.com` site, located in California. Presumably due to longer transport delays the single click Web-flow splits in parts separated by inter flow gaps (*WFG*) bigger then 1-1.5 seconds. The Table 4 presents parameters of selected Web-flows. The web

Table 3. Web flow parameters for the page cordis.europa.eu

Parameters	Chrome	Firefox	IExplorer	Opera
F2F mean	0.036	0.022	0.067	0.032
F2F variance	0.043	0.0011	0.003	0.003
TF time	0.781	0.450	1.051	0.661
N (flows number)	21	20	17	21

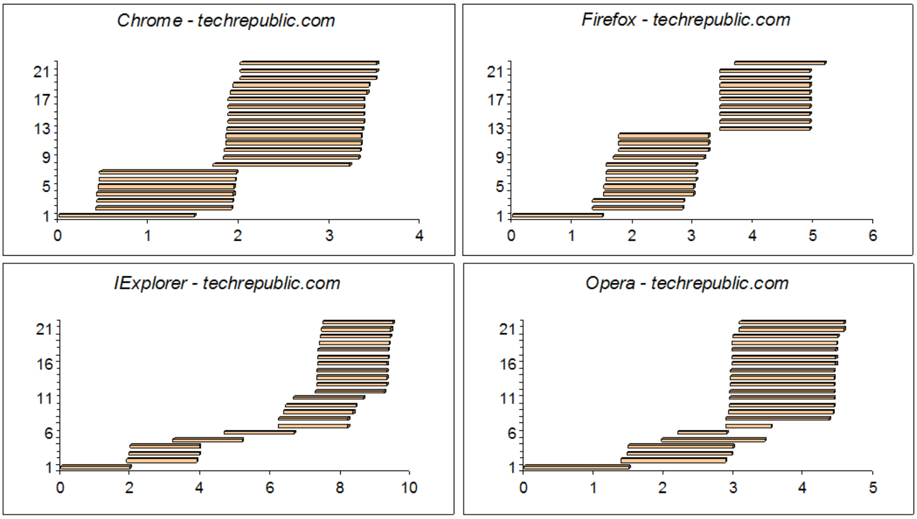


Fig. 5. Web-flows for `techrepublic.com`

Table 4. Parameters for the Web flow of `techrepublic.com`

Parameters	Chrome	Firefox	IExplorer	Opera
F2F mean	0.172	0.168	0.341	0.140
F2F variance	0.185	0.194	0.359	0.102
TF time	2.013	3.685	7.491	3.085
N (flows number)	22	22	22	22

mail applications make use of regular browsers, so their network activity looks very similar to previous cases. Figure 6 presents the flows related to user login to the `gmail.com` site, and derived flow parameters contain Table 5.

Comparing the obtained values of F2F parameter from this exemplary cases of web browsers operation one can notice that mean value of F2F is smaller then 400 ms, (in most cases much smaller then 200 ms), and the variance of F2F is generally smaller then mean value. The values of total flow time (*TF*) and the number of net flows in Web-flow are sometimes difficult to define and was chosen somewhat arbitrary. If web page contains any animations, the download

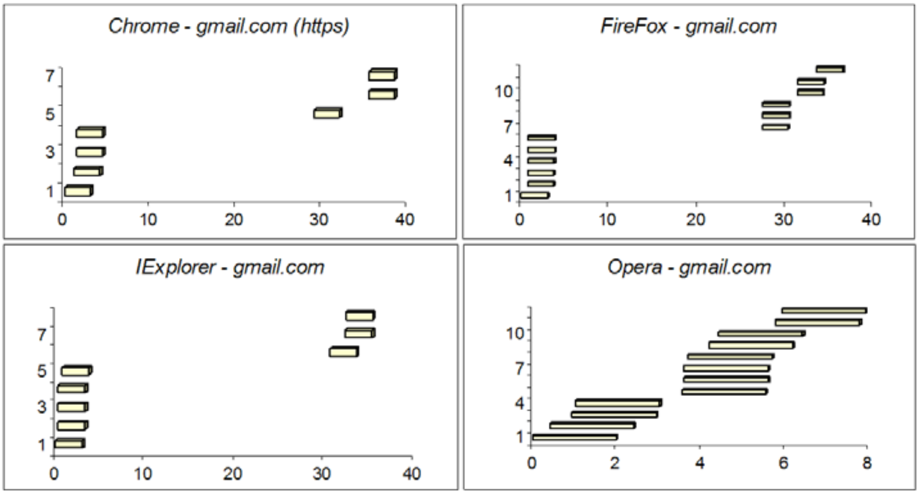


Fig. 6. Web-flows of login proces to `gmail.com`

Table 5. Parameters for the Web flow of `gmail.com`

Parameters	Chrome	Firefox	IExplorer	Opera
F2F mean	0.497	0.156	0.178	0.347
F2F variance	0.378	0.090	0.034	0.053
TF time	1.492	0.781	0.711	1.041
N (flows number)	4	6	5	4

continues through many TCP connections (30–50), without any clearly visible WFG gap.

There were many other then browser applications generating outbound network traffic. Two of them, communicators of the client-server type, *gadu-gadu*, and peer-to-peer type, *Skype* were also tested, and their behavior looks different. Figure 7 presents flow activity of both communicators. *Gadu-gadu* behaves like browser with web page containing animation, with F2F mean time of 0.078 s and F2F variance of 0.0022 s. The *Skype* HTTP communication is totally different. Single net flows arrives at random moments, and are probably connected with user textual communication. Similar activity is presented also on connections directed to port 443.

From other common system application similar to web browsers network activity manifests anti-virus programs, which continously periodically contacts their known sites. Figure 8 presents the operations of two such programs: *ESET NOD32 Antyvirus* and *COMODO AntiVirus*. The *F2F* time parameter has for NOD32 program values from 0.286 to 0.717 s.

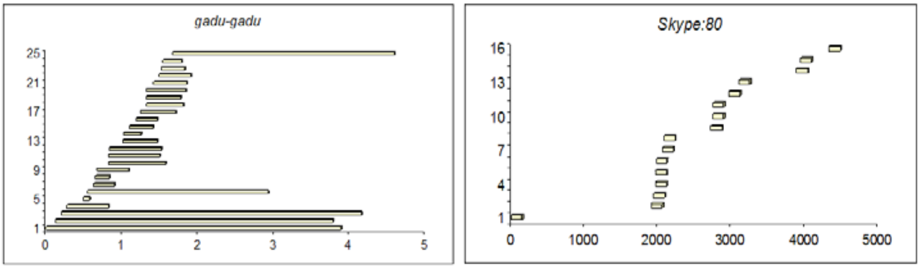


Fig. 7. Messengers net flows activity

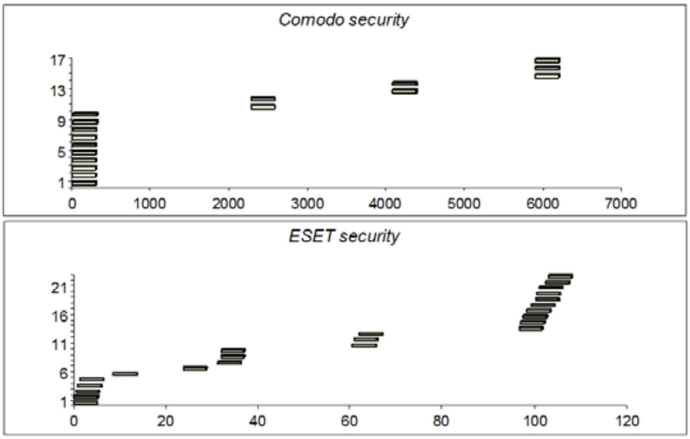


Fig. 8. Network flows of AntiVirus software

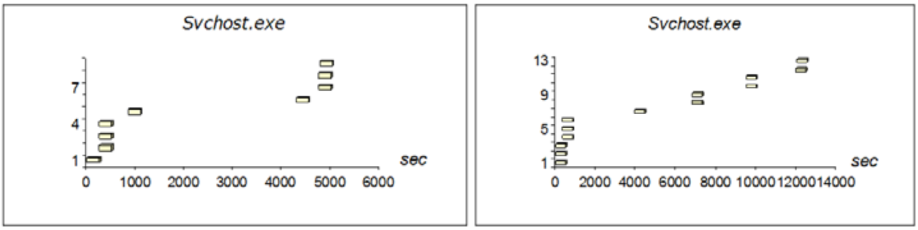


Fig. 9. Network HTTP activity of system services

The operating system network activity is also visible as HTTP outbound flows, however do not shows signs of regularity. On Fig. 9 there are presented network flows related to two different days of *svchost* service activity. The connections occures in unregular times, and their random, rather short (less then 500 ms) duration was, for visibility, extended on plot to about 500 seconds.

4 Conclusion

The concept of outgoing network traffic classification based on Web-flow model flows analysis was presented. Results of experiments confirm the ability of user initiated traffic selection based on the properties of outgoing network flows of typical user applications. Some initial values of flow-to-flow time (*F2F*) parameter intended as the basis for network traffic classification were identified. The tests results depend on application's embedded communication algorithms, so normal user activity has no influence on classification results (don't change minimal (*F2F*) times, chosen as the basis for application flows characterization). Examples of other types of system's network behavior were also presented.

In the future the examination of the effectiveness of Web-flow traffic classification model in the detection of malware generated network traffic is planned, for incorporation as additional way of improving the efficiency of N-IDS system in network threats detection.

References

1. Choi, H.K., Limb, J.O.: A Behavioral Model of Web Traffic. In: ICNP 1999 Proceedings of the Seventh International Conference on Network Protocols, pp. 327–334. IEEE Press, Washington (1999)
2. Hernandez-Campos, F., Jeffay, K., Smith, F.D.: Tracking the Evolution of Web Traffic: 1995-2003. In: Proceedings of the 11 th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Orlando, pp. 16–25 (2003)
3. Lee, J.J., Gupta, M.: A new traffic model for current user web browsing behavior, `http://blogs.intel.com/research/HTTP%20Traffic%20Model_v1%201%20white%20paper.pdf`
4. Shuai, L., Xie, G., Yang, J.: Characterization of HTTP behavior on access networks in Web 2.0. In: International Conference on Telecommunications, ICT, pp. 1–6 (2008)
5. Kim, M.S., Wona, Y.J., Hong, J.W.: Characteristic analysis of internet traffic from the perspective of flows, `http://dpnm.postech.ac.kr/papers/Comp-Communications/06/flow-based-traffic-analysis.pdf`
6. What is netflow? `http://www.caligare.com/netflow/netflow.php`
7. ARGUS – Auditing Network Activity, `http://www.qosient.com/argus`