Efficient Network Monitoring for Large Networks

Chia-Mei Chen* Chuan-Pi Wei

Department of Information Management
National Sun Yat-Sen University
Kaohsiung 804 Taiwan

Email:cchen@mail.nsysu.edu.tw

Received 16 November 2007; Revised 30 November 2007; Accepted 9 December 2007

Abstract. Denial-of-Service (DoS) attack has become a major threat to the Internet. Network anomaly may be a sign of a possible attack. Network administrators seek for an efficient, scaleable, and real time solution of monitoring a large and heavy traffic network and detecting network anomaly efficiently, or the network might not be able to operate properly. The collected data sometimes might be either too coarse to detect anomaly or too detail to finish processing in real time. SNMP based network monitoring collects coarse information not enough to detect the problem, while packet-sniffing based monitoring retains very detail contents and affects network performance, especially in large networks. Network flow is defined as a unidirectional sequence of packets between the given source and destination network endpoints. Flow information might be the balance between the above two approaches. We propose a network monitoring mechanism for large networks based on flow information which can manage a large network efficiently in real time manner. Based on simulation with the real network traffic, the proposed solution can efficiently monitor a large network and detect Denial of Service (DoS) attacks, port scans, and worm propagation. The results show that it is significantly helpful for network administrators for large networks.

Keywords: network monitoring, flow profiling, Denial of Service attack, worm propagation

1 Introduction

Nowadays the Internet has become an important part of people's daily life. People receive emails, surf the web sites, and chat with friends on the Internet everyday. Unfortunately the Internet is not secure as we more rely on it for personal or business purposes. Malicious probes, stealthy intrusions, and worms spreading happen on the Internet constantly. Undoubtedly security has become one of the most important and urgent issues on the Internet. It must be a common thought of all the Internet users that we need reliable and efficient network services with strong and confident network security.

Real-time network traffic monitoring provides the network administrators the status and the patterns of the network traffic and the signs of any possible abnormal traffic and possible potential problems within the administrative domain. During a development of an incident, the attacker might send packets to inquire or collect the detailed information of the target system. In the early stage of spreading, if there is one monitoring system to detect the irregular activities and to gather related information, the network administrators may identify the possible attack through the monitoring and thus can respond to the situation in time to prevent it from getting worse. The audit trail of the network activity can be collected for future investigation, such as what happened, how or why it happened and how to prevent in the future. These logs might be the only evidence of intrusion, in case the victim's logs have been compromised.

The network administrators typically use the SNMP-based tools to collect network traffic count from network service equipments like a switch or a router supporting SNMP protocol. These tools usually consist of two components. One, namely the collector, is to collect SNMP data, and the other, the grapher, is to generate HTML formatted output containing traffic loading image which provides a live and visual representation of the network traffic and traffic trends in time-series data of the network. These traditional SNMP-based traffic monitoring tools, such as MRTG [1] and Cricket [2] may provide abnormal warning that there is something strange--an unexpected increase in traffic may indicate that a security incident is being in progress--but SNMP-based tools only provide information about levels and changes in traffic volume. For security purposes, the volume-changing information is insufficient for network administrators to determine whether there is anything wrong with the network and to find out the trouble. Making a judgment needs more detailed data to be provided.

* Correspondence author

Packet sniffers and other packet sniffing related tools are prevalent recently and have been deployed rapidly at present. These tools capture the traffic packets, decode the packet header fields, and even dig into the packet content to provide much more detailed information. Although they can provide details on packet activity, they lack information on the network as a whole. They typically focus on the content of single network packets, not on global network activities, and thus lack high-level support to management activities.

Timely analysis and storing this usually tremendous volume of traffic data sometimes can be impractical in some environments, especially on large scale and busy networks. Increasing traffic bandwidth and throughput make data capture and analysis more problematic and unstable when the traffic is heavy. There may be a break-down when the traffic is too heavy to handle with. Furthermore, the purpose of these tools is usually and originally designed for detecting individual attacking event, not for monitoring overall network traffic condition. Accordingly using them for monitoring networks will be awkward and breakable. The most important is that we are unwilling at all to create an artificial bottleneck and uncertain factor in our network for network monitoring and network information gathering.

As we described above, the network administrators have realistic and pressing needs of getting more detailed information about the live network than the traditional SNMP-based tools can provide. Furthermore they don't either want to sacrifice performance and stability of the network for using packet sniffing tools. Moreover while worm spreading and DoS1 attacks have been the most critical problems that network administrators are facing at present time, they should have the capability to detect the puzzle as soon as possible. Consequently we desire to develop a new network monitoring method and even build a practical system for network administrators to help them to examine real time network utilization statistics, look at traffic patterns, and perform early detection of worm propagation and DoS attacks.

2 Related Studies

There are three types of monitoring tools: SNMP based, packet-sniffing, and flow based monitoring tools. The detail survey is described in the following subsections.

2.1 SNMP-based Monitoring Tools

MRTG (Multi Router Traffic Grapher) [1] is the most famous and most widely adopted SNMP-based traffic monitoring program. It is originally designed to monitor the network traffic loading. It generates HTML pages including traffic statistics images to provide a live visual representation of the network traffic.

MRTG keeps a log of all the data it has collected from the devices. Therefore in addition to provide a 5 minutes average daily view, MRTG also creates visual representations of the traffic history stored during the last seven days, the last five weeks, and the last twelve months. Because It is set up to contain all the relevant data seen over the last two years, logs does not grow unlimited over time. MRTG is not limited to monitoring network traffic, though. It is possible to monitor other SNMP variables in which we have interest. We can even use an external program to collect any statistic information which we plan to monitor via MRTG. People often use MRTG to monitor dynamic information such as system load, login sessions, CPU usage, or memory usage. Moreover, we can even accumulate two or more data sources into a single graph.

2.2 Packet-Sniffing Monitoring Tools

ntop

Ntop [4] is an open source, network traffic measurement and monitoring tool acting as both a traffic probe and analyzer. It is based on libpcap [5] to capture packets, and decodes the packets to show the network usage, similar to what popular Unix "top" command does. It can be used to track relevant network activities including traffic characterization, traffic patterns, network utilization, network protocol usage, and congestion detection. It can support various management activities, including traffic measurement and monitoring, network optimization, and network planning. The database support makes ntop suitable not only to debug network problems, but also to be taken for long-standing network monitoring and problem backtracking.

The traffic reports can be reported in two ways: web mode and interactive command line mode. In the web mode, the web page gives a summarized view of the current and past network activities (ntop acts as a web server), periodically refresh automatically or on user request. The use of a web interface and limited configura-

[1] DoS attack mentioned includes DoS and Distributed DoS (DDoS).

tion and administration via the web interface make ntop easy to use and suitable for monitoring various kinds of networks. In the interactive command line mode, traffic information, shown in a character-based terminal in a network shell based on the ntop engine, interacts with ntop and provides more powerful capabilities.

IPAudit

IPAudit [6] is an open source network monitoring program. It uses libpcap packet library to sniff packets. It records network activities on a network by host, protocol, and port. Hence, it listens to the network device in the promiscuous mode, and records every connection between two IP addresses. A unique connection is identified by the IP addresses of the two end machines, the protocol used between them, and the port numbers (if they are communicating via TCP or UDP). IPAudit is suitable for network monitoring, intrusion detection, bandwidth consumption, and DoS attacks. It is usually used with IPAudit-Web to provide web based network reports.

2.3 Netflow

A network flow is defined as a unidirectional sequence of packets between given source and destination network endpoints. Flow endpoints are identified both by IP addresses and the port numbers. Researchers and networking equipment vendors have developed the tools and techniques of flow profiling to characterize the network traffic. Flow data provides a detail information about the network traffic. Cisco's NetFlow services developed in 1996 became the most commonly available flow profiling system and the primary network accounting technology in the industry. It provides the measurement for the flow-based network analysis.

NetFlow uses the following header fields to uniquely identify a flow: source IP address, destination IP address, source port, destination port, layer 3 protocol type, Type of Service (ToS), and input logical interface (ifIndex). NetFlow log contains valuable information of a flow, including the source and destination IP addresses of the flow, the source and destination port numbers of the flow, the start and end time of the flow, the number of packets and octets in the flow. In addition, Netflow records the IP protocol (TCP, UDP, ICMP) and the router interface that the traffic was received on and sent to. In fact, the flow content varies slightly between NetFlow versions.

A flow is expired if one of the following conditions is met and the router exports the flow once it is expired.

- Flows which have been idle for a specified time are expired.
- Long-lived flows are expired. By default this is set at thirty minutes.
- The cache becomes full, and so heuristics are applied to age groups of flows to expire and export those flows.
- The TCP connection associated with the flow has reached its end (FIN) or has been reset (RST).

The router groups the records of the expired flow into the NetFlow export UDP datagrams and exports the datagrams to a collection station. The exported datagrams generally contain multiple flow records. The router exports them at least once per second, or when a full datagram is available. Flows are unidirectional, so a TCP connection usually consists of two flows: one from the initiating host to the destination host and the other from the destination back to the initiating host. A long TCP connection may be represented as multiple flow records due to the expiration rules for the flow cache mentioned above [7].

3 The Proposed Solution

We propose a network monitoring mechanism based on NetFlow log as shown in Fig. 1. First, the collecting module collects flow logs exported from NetFlow-supported network devices, writes them into the disk periodically. The statistic module then reads from the disk and performs flow aggregation and produces visualized statistic report on-line. The system alerts when a suspicious activity is found. Administrator can watch the current traffic pattern in real time based on the statistic report. Flows can be examined later more thoroughly and comprehensively if there is a possible on-going event. Each module will be described in detail below.

Collecting Module

The router groups the records of the expired flows into NetFlow exported UDP datagrams and an exported datagram generally contains several flow records. Hence, the capturing daemon takes charge of capturing UDP packets, storing NetFlow records in their internal format, and rotating the records into the disk for further analysis. A fixed amount of disk space is allocated for such storage. Hence, the records are kept and flushed away periodically once the analysis is done.

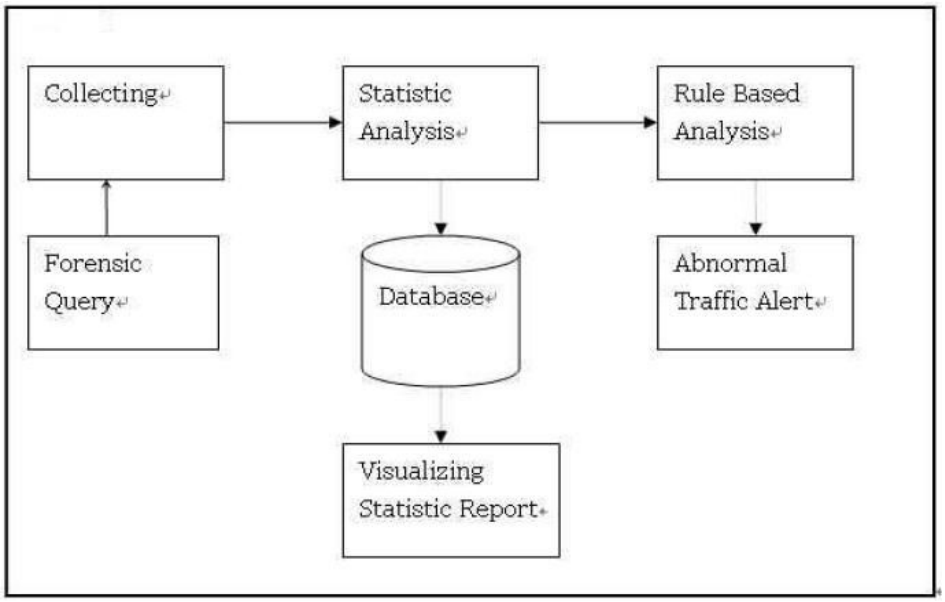

Fig. 1. The proposed system

The number of the records might occupy extremely large space, such as tens even hundreds of megabytes in several minutes in a high traffic network. Hence, the length of a period of the analysis will be affected by the allocated disk space and the maximum traffic load of the network. Therefore, the disk size should be carefully chosen. To accelerate the analysis, a RAM disk can be selected to store flow files temporarily. Access time of a RAM disk is much faster than that of a real disk.

Statistic Analysis Module

The traffic analysis module examines each flow, maintains the counts of the attribute values, and summarizes and stores the statistics into the database. The statistic information during a period of time will be shown in a visual form in web pages, including the number (count) of flows in each protocol (TCP, UDP, ICMP), the count of application ports, the bandwidth consumption per host, and so on. The network administrator can monitor the network with such visual graphs and hence can understand the network performance, bandwidth utilization, application usage distribution and the big talkers.

We observe that monitoring the traffic by flows can identify anomalies more efficiently than that by bytes, since some anomaly might not cause a large variation in traffic volume, but it could have an impact on flow counts, such as small packet-size DoS attacks. It is also important to watch the TopN service ports and hosts as we could discover which application or host consumes the most bandwidth. A quiet port or host becoming a big talker might be a sign of anomaly. We also found that the aggregated flow information of the selected service ports should be plotted into separate graphs and anomaly could then be identified more easily. Fig. 2 shows a single graph aggregating all the selected services, while Fig. 3 shows the same situation with a separate graph for each service. The former graph suppresses the variation of the counts and might not be able to spot an unusual event through such graph. Hence, the proposed system provides a separate graph for each monitored service.

Rule Based Analysis Module

Statistic traffic information helps network administrator shorten the management time, but some attacks may need further analysis and reaction in real time. Hence, the proposed system establishes rules to alerts the attacks, such as DoS, port scan, network scan, and worm. DoS attacks often have the pattern of a distributed or a single source IP address sending out similar small packets targeted at a particular host. Hence, the system will collect abnormal amount of the flows with this pattern in a short time. A threshold is set to find the abnormal traffic load.

Port scan may produce small packets of SYN, ACK, or UDP from the same source host to the same destination host, while network scan targets different hosts on the same network, instead of different ports on a specific host. The system needs to know worm behaviors a priori to discover worm activities. Worms usually scans the specific port, such as TCP 80 port, on random targets. The attack packets of such flow are usually the same. For example, Slammer worm generates 404 byte packets, destined at UDP 1434 port to randomly generated hosts. We can use these known characteristics on the destined port, packets, and bytes and establish filtering rules to discover worm propagation.

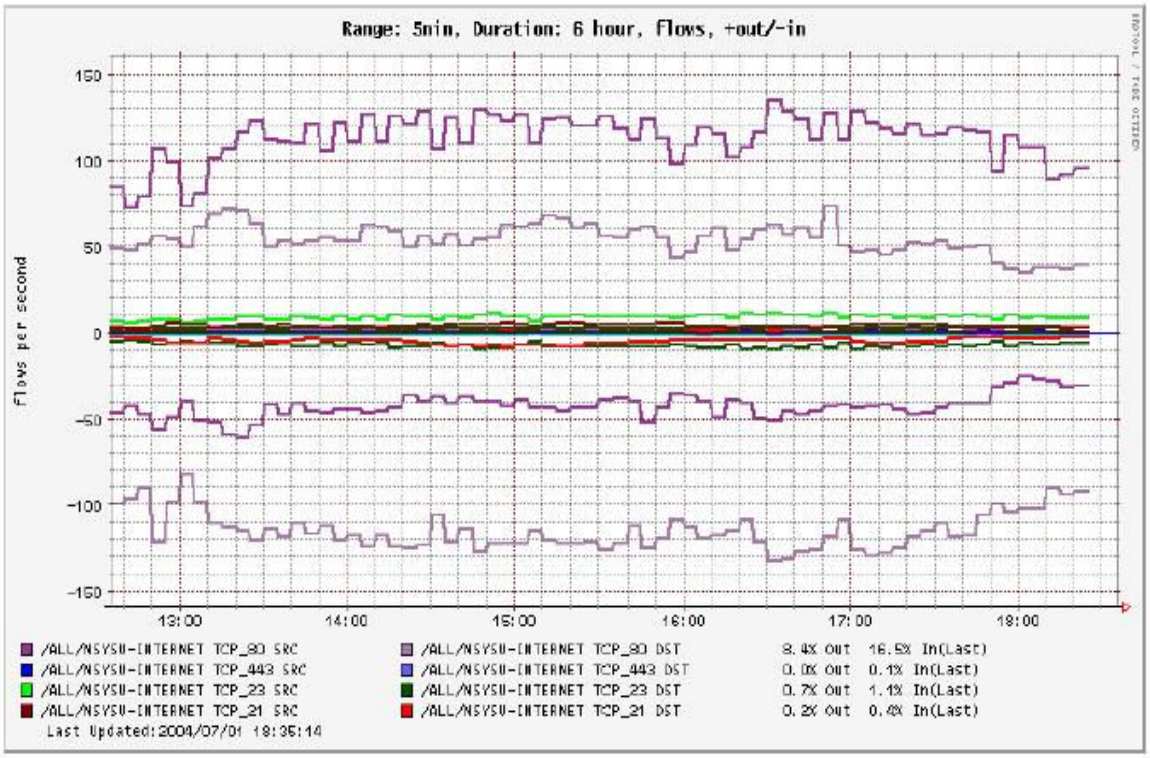

Fig. 2. Graph with flow aggregation

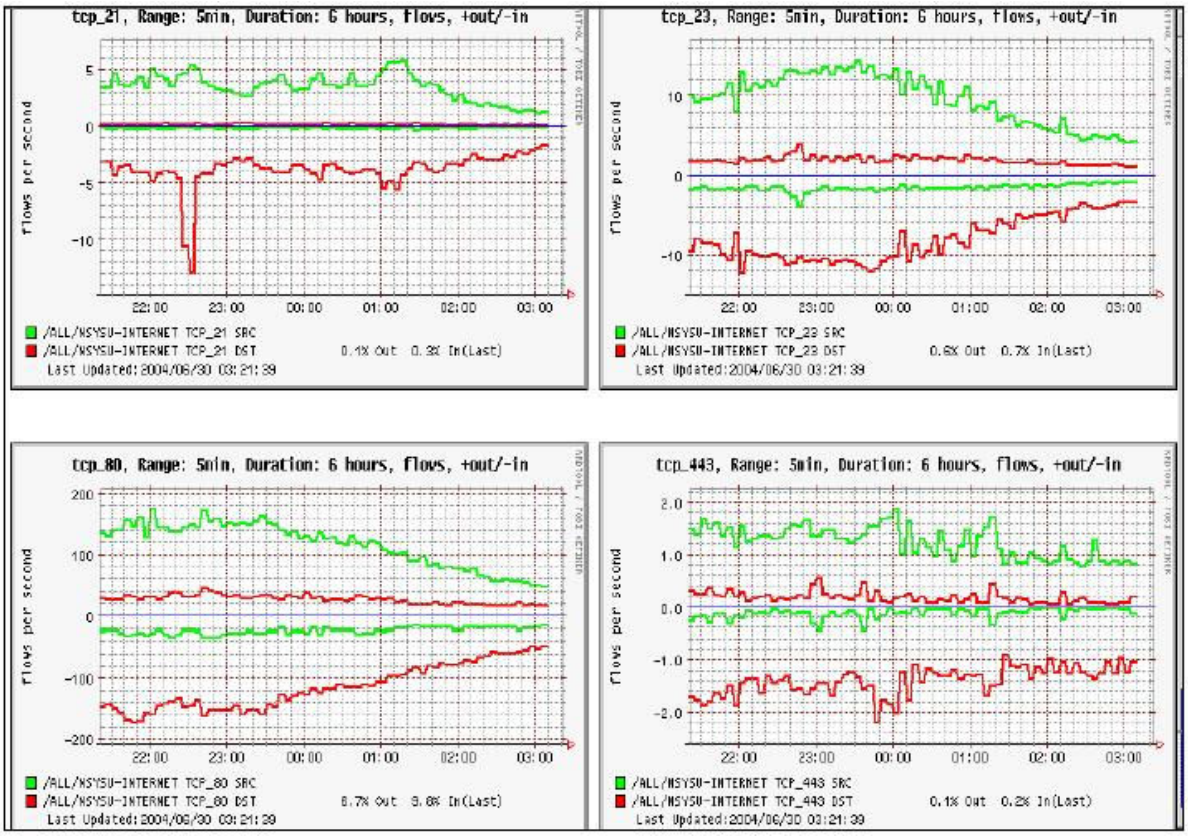

Fig. 3. Graphs without flow aggregation

4 Experimental Results

The real network, the university network of the National Sun Yat-Sen University where southern area network center is located, is used to evaluate the proposed system. The experimental results are shown as follows.

Results on Traffic Monitoring

Network administrator used to aggregate different data on the same graph and to present it simultaneously and comparatively, but the diversity of traffic volume makes the presentation obfuscating. As we can see in Fig. 4, we almost see no information except TCP protocol traffic variation. Hence, It is practical to show different protocols on separate graphs. Note that as what we have emphasized before, it is beneficial to monitor flow change instead of byte change.

The peaks of the flow graph shown in Fig. 5 illustrate that the ICMP connection burst indicates a suspicious on-going event. Network administrator can then query which TopN hosts during that period of time which cause the ICMP traffic and then identifies the attack host. The proposed system provides a visualized graph showing abnormal traffic and a query system for further investigation and attack origin identification.

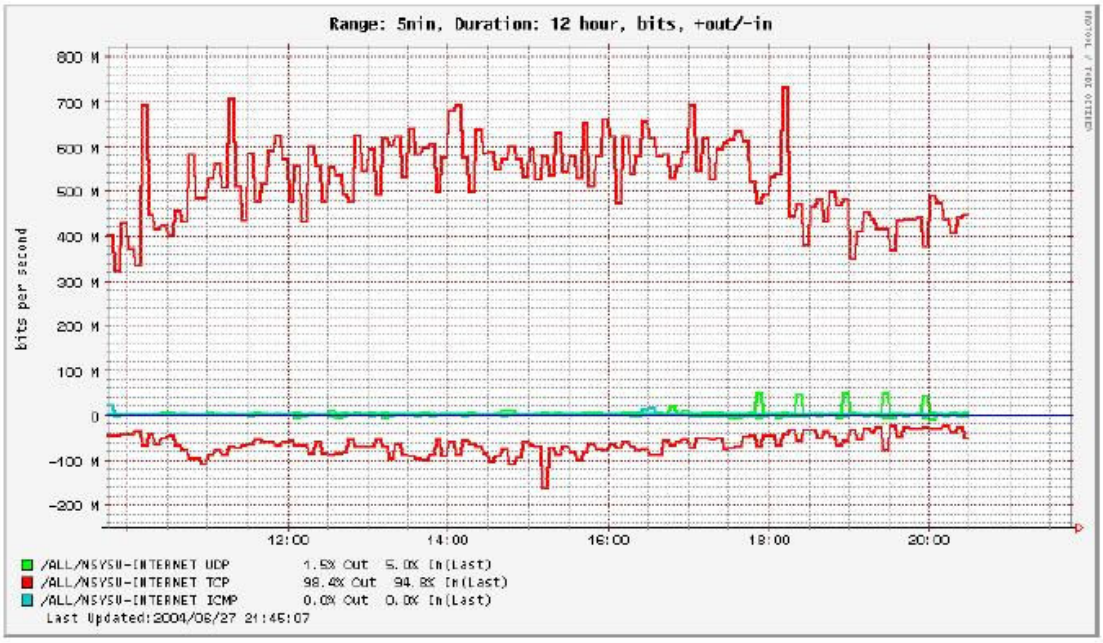

Fig. 4. Traffic volume in IP protocol distribution

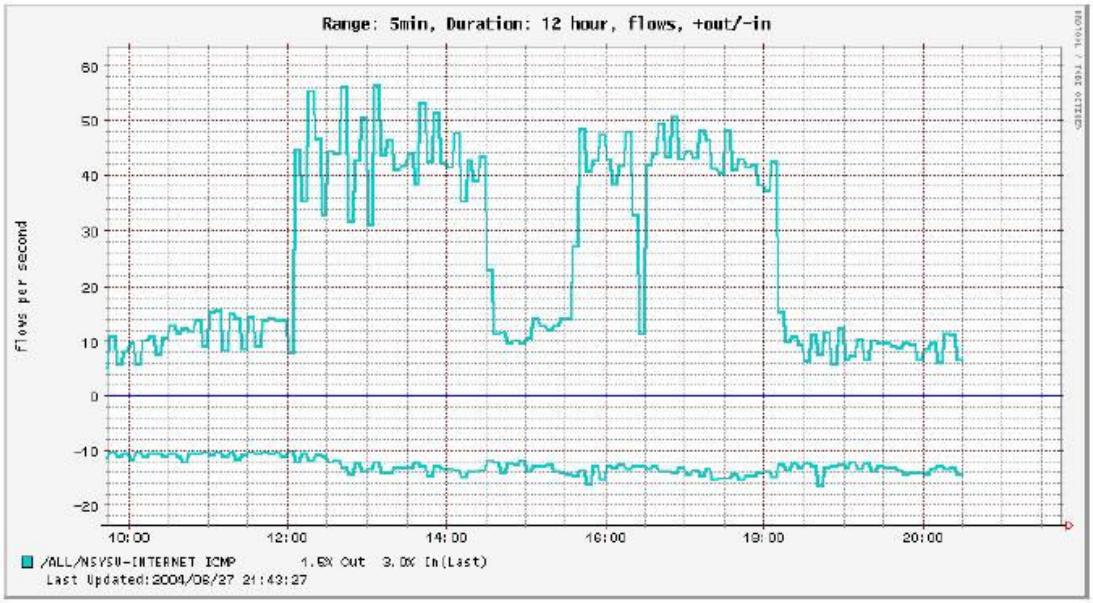

Fig. 5. Flow graph with diverged ICMP traffic

Results on DoS Attack Detection

During the period of experiments, monitoring the network with a separate flow graph for each well-known service, we observe that a discrepancy flow volume between the incoming and outgoing TCP port 22 (SSH service). A normal connection consists of two unidirectional flows and the number of the flows in both directions in a network normally will be the same or approximately equivalent. An unbalanced flow volume between the two directions may signal an on-going event.

Fig. 6 shows the traffic volume (in flow graph), in different time intervals, 5 minutes, half hour, two hours, and one day. All of the flow graphs indicate that the outbound flows are exceptionally higher than the inbound flows. Therefore, we explored the raw flow files or query Ntop hosts to see why there were so many unusual flows. We found that several hosts on our network connected to TCP port 22 of a particular host outside continually and repeated. Such traffic pattern indicates a possible DoS attack on the network. Further investigation or query is needed to spot the attack site.

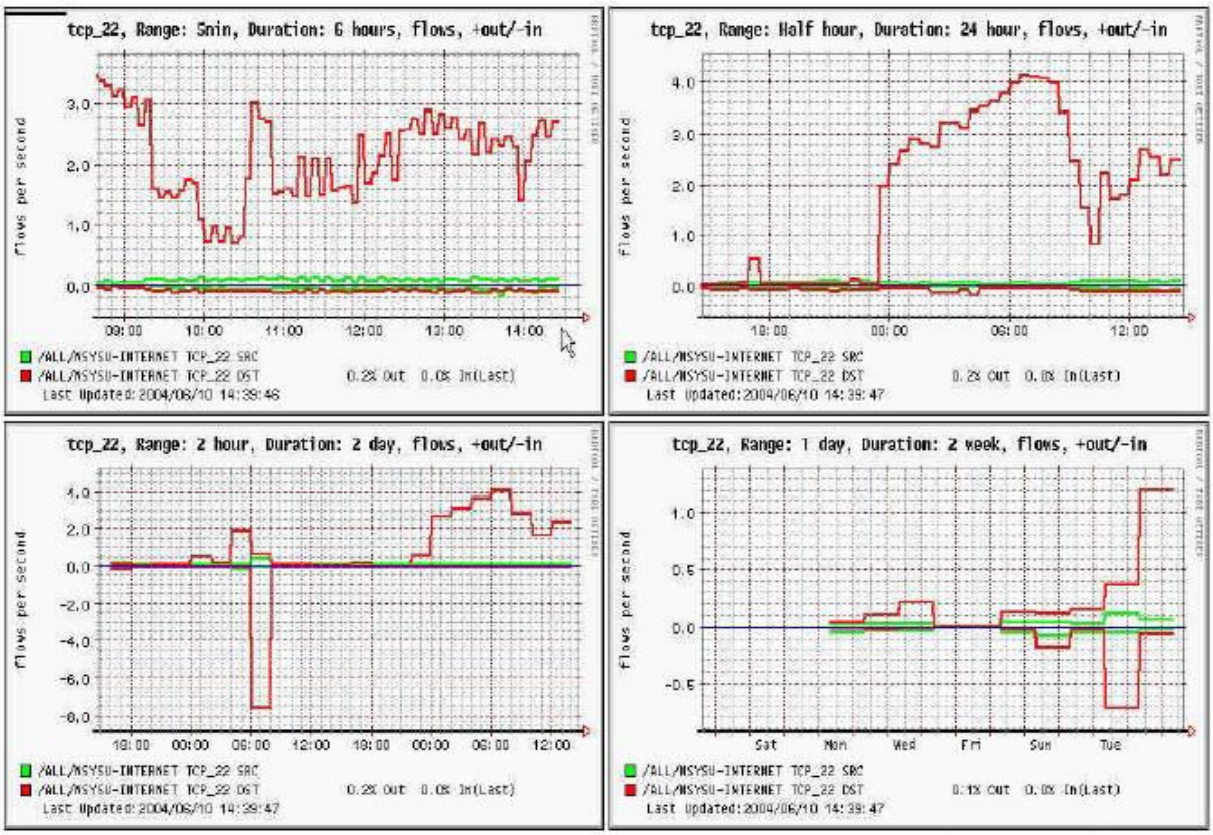

Fig. 6. TCP port 22 flow graphs

Worm attack usually goes through a specific port, for example a vulnerable service, sometimes, so the traffic of the specific port should be monitored. Therefore, we can detect the attacking instances or the worm propagation as soon as possible and can launch countermeasures promptly to defend against. When the Microsoft vulnerability, MS04-011 [8], is announced, the traffic through TCP port 443 (HTTPS service) needs to be monitored. During the experimental time, we observe the abnormal traffic from that port as shown in Fig. 7.

After exploring the flow records further, we found suspicious scans coming from a single source host, sending to multiple victims of a specific network, and having the same packets and bytes. Such abnormal flow may be generated by an attacking tool or a worm. The proposed system could find the abnormal traffic through the flow graphs showing the real-time traffic statistic information and the further analysis can find the attack site or identify the problem. The administrator can use the pattern to detect other suspicious activities afterwards. According to this collected pattern, then we do discover more and more questionable traffic of the same characteristic spreading on the network.

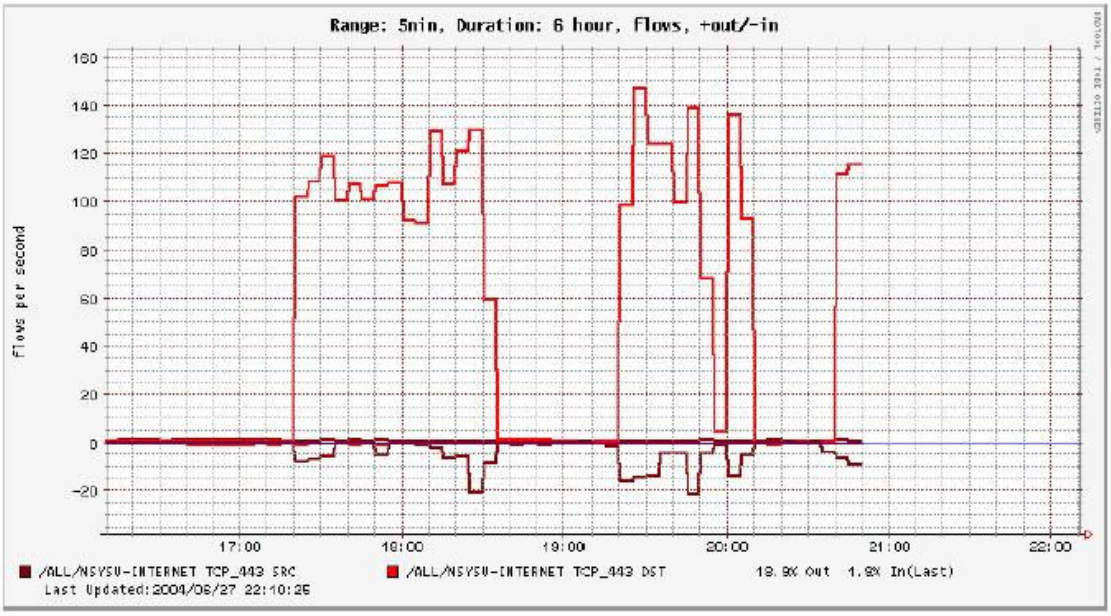

Fig. 7. Flow graph of TCP port 443

5 Conclusions

Many security events happened on the network nowadays. The administrators seek a solution to shorten the network management time in a large network and to find the malicious activities in progress as soon as possible in order to launch effective and efficient countermeasures to defend against. They often monitor the network by collecting the real time traffic data, but they might find it may be too little or too much information. Too less information might not be able to discover the on-going event, while too much information may require more analysis effort and could not be able to discover the anomaly promptly.

The proposed mechanism use flow information to monitor a large network in real-time. Based on the experimental data collected from the real network, campus network of a university, we found that the proposed mechanism based on flow log does help the network administrator monitor a large network in a timely manner and that the flow graphs and the flow records provide enough information for the network administrators to identify suspicious network abuse in a large network. We also discover that separate flow graph for each monitored service is easier to identify anomaly than aggregated flow graph of all monitored services, especially detecting DoS attacks and worm propagation.

The proposed mechanism use rule-based approach to filter well-known worm or DoS attack flows. Data mining or artificial intelligent approach may be used to discover the spreading of unknown worms or new DoS attacks from the flow records.

Acknowledgement

This work was supported partially by NSC research Grant NSC96-2218-E-110-005 and TWISC@NCKU, NSC, under Grant NSC96-2219-E-006-009.

References

[1] T. Oetiker, D. Rand, et al. "*Multi Router Traffic Grapher (MRTG),*" http://oss.oetiker.ch/mrtg/.

[2] J. R. Allen, *Cricket*, http://cricket.sourceforge.net/.

[3] Cisco, “Cisco IOS NetFlow,” http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.

[4] MRTG, http://www.mrtg.org.

[5] Ntop, http://www.ntop.org/ntop.html.

[6] Tcpdump/libpcap, http://www.tcpdump.org.

[7] IPaudit, http://ipaudit.sourceforge.net.

[8] Cisco, "NetFlow Services and Applications," http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.pdf.

[9] *Microsoft Security Bulletin MS04-011*, http://www.microsoft.com/technet/security/bulletin/ms04-011.mspx.

[10] Cisco, "NetFlow Services Solutions Guide," http://www.cisco.com/univercd/cc/td/doc/ciintwk/intsolns/netflsol/ nfwhite.htm.

[11] D. Plonka, "FlowScan: A Network Traffic Flow Reporting and Visualization Tool," http://net.doit.wisc.edu/~plonka/lisa/FlowScan/out.ps.gz.

[12] J. -P. Navarro, B. Nickless, and L. Winkler "Combining Cisco NetFlow Exports with Relational Database Technology for Usage Statistics, Intrusion Detection, and Network Forensics," *Proceedings of the 14th System Administration Conference*, pp.285-290, 2000.

[13] D. W. McRobb, "cflowd configuration," http://www.caida.org/tools/measurement/cflowd/configurtion/ configuration.html.

[14] Flow-capture, http://www.splintered.net/sw/flow-tools/docs/flow-capture.html.

[15] Flow-nfilter – filter flows, http://www.splintered.net/sw/flow-tools/docs/flow-nfilter.html.

[16] Cflow – find "interesting" flows in raw IP flow files, http://net.doit.wisc.edu/~plonka/Cflow/.