

Design and Implementation of Content-based Filter System on Embedded Linux Home Gateway

Li-Der Chou^{1*}, Zheng He¹, David Chunhu Li¹, Hui-Fan Chen¹, Jun-Jie Su¹, Ching-Yung Chen², Hsu-Chuan Wei² and Chia-Jen Li²

¹Department of Computer Science and Information Engineering, National Central University, Zhongli City, Taoyuan County, Taiwan

²Proscend Communications Inc, Hsinchu, Taiwan

*cld@csie.ncu.edu.tw

Abstract—With rapid development of the Internet, Internet surfing has been an important daily activity for peoples. There are many websites and web pages containing obscene and malicious information such as phishing, spam and pornography that peoples can easily obtain. In order to improve network security service, we developed and implemented a content-based filter system on an embedded Linux home gateway. We analyzed and compared two feasible approaches that the content-based filter was respectively embedded into user space and kernel space of Linux home gateway. The content-based filter system comprises URL (Universal Resource Locator) blocker and keyword filter in the kernel space. The Linux home gateway embedded with our new content-filter was evaluated by extensive experiments. The experiment results showed that the content filter method that based modification of kernel space outperforms around 35% improvements than that of based on user space.

Keywords— Content-based Filtering, Embedded Linux, Home Gateway, URL Blocker, Keyword Filter.

I. INTRODUCTION

With rapid development of scientific technologies, the Internet has increasingly made a great impact on the world. The network technology experienced a tremendous evolution from dial-up, ADSL (Asymmetric Digital Subscriber Line), CATV (Cable Television) to FTTx (Fiber to the x) and 3G (3rd Generation Mobile Telecommunication), which make the Internet indispensable to human life. According to statistics in [1], the population of Internet users in Taiwan ranked fourth in Asia, just behind Korea, Japan and Singapore. Most of Internet users are the youth including elementary school, junior and senior high school students. Network security problem is gradually brought the people's attention with the increase of the youth Internet users [2]. On the one hand, there are many websites and web pages containing obscene and malicious information such as phishing, spam and pornography that the peoples can easily obtain. It is very difficult for parents and teachers to watch their children and the teenage surf the Internet all the time. Furthermore, the parents and teachers concern about the content of website that the teenage browsing whether is appropriate or not. To manage the network security, the technique of content-based

filtering will be a feasible solution. It is possible to reduce the loss and even avoid negative effects by filtering inappropriate information such as phishing, spam, and pornography.

In the personal home network, a home gateway [3] is mainly responsible for delivering and interchanging data packets between the Internet and home network users. Moreover, with the use of home gateway, the internal network users can share a physical IP via NAT (Network Address Translation) [4]. A common home gateway based personal network is depicted as Figure 1. Even though many home gateway products are designed with various network services, network security management has been a challenge issue. Therefore, in this paper, we are motivated to design and implement a content-based filter system on embedded Linux home gateway to protect users from obscene and malicious information with benefits of cost-effective and improvement of network security management. Two different content filtering methods based on modifications of Linux home gateway are proposed and implemented. Moreover, we conducted comprehensive system experiments to evaluate network security performance of our new content-filter based Linux home gateway.

The rest of this paper is organized as follows: the related work of content filtering service is discussed in section II. In section III, we analyze feasible filtering approaches in the Linux Embedded System. System architecture and our two content-based filter modules are presented in section IV. Section V describes how to implement content filtering service in the Linux Embedded System. System evaluation and results are discussed in section VI. We conclude our work in section VII.

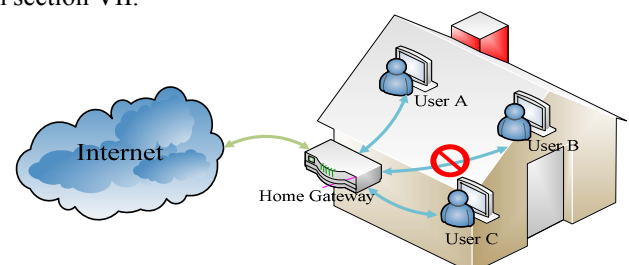


Figure 1. Home gateway

II. RELATED WORK

Network security has been an important research topic in academia and content-based filter is one of major solutions to solve network security problem. There are many researches related to content-filter services in literatures have been conducted and various approaches have different strategies. Liu and Wang [5] pointed out internet content rating is important to detect, filter and block illegal and inappropriate Internet content. Furthermore, they analyzed the state of the art of the content rating technology standards.

Chen and Wang [6] reported that numerous web pages contain contents such as fraud, web spam, malware, child pornography and terrorism should be filtered. Vijay [7] proposed three common techniques that can block access to Internet sites. They were IP-level packet filtering (blocking or dropping), Domain Name System (DNS) poisoning, and content filtering at higher protocol layers, such as URL blocking. Content filtering mechanism is possible to implement on multiple levels in the Internet. For instance, on the level of ISP (Internet Service Provider) and on the end network user computer or network nodes such as the home gateway, the concentrator in central office and the edge router.

Textual content analysis is a popular approach to classify various webpages into categories. In [8], Kim et al proposed a semantic filtering mechanism to classify malicious contents. A lexical named entity recognition system was designed in [9] to improve the effectiveness of text classification in the framework of intelligent Web content filtering. The impact of content filter requirement on network security was studied in [10] and a fully polynomial time approximation scheme was developed to maximize traffic pass through the content filter nodes. In [11], Noll and Meinel demonstrated a collaborative web filter – TaggyBear to bind in the browser as an add-on component. With the help of the TaggyBear, Internet users can rank websites, share blacklists and download them from the Internet.

In recent years, many countries have established a series of laws, rules and regulations to supervise Internet contents. On the one hand, government authorities actively conducted various researches, developments and applications of Internet content filtering.

On the other hand, in the academia and industry, many academic organizations and commercial companies conducted lots of researches and manufactured content-filer based productions. HiNet [12], the biggest Internet service provider (ISP) in Taiwan, has provided URL filtering service by setting up transparent proxies and examining all HTTP requests sent from users. The users thus can have high level network security service without installing any particular anti-virus software in their computers. Another well-known filtering system is Green Dam Youth Escort [13] designed by Chinese government in which all retail personal computers in China were asked to install the filter software to filter networking content. The Internet users in Malaysia have increased gradually. Meanwhile network security threats that detected by various organizations also increased. In [14], a collaborative content filter system model was proposed by

Malaysia Communication and Multimedia Communication (MCMC) which is the regulator and law enforcer. The major purpose of collaboration system model was to make the system for filtering content in Internet more efficient and faster.

III. THE ANALYSIS OF FILTERING APPROACH

To improve existing content-based filter solution in network security with the benefit of cost-effective on the Linux embedded system, in this paper, we are motivated to design and implement a content-based filter system on embedded Linux home broadband gateway with Netfilter framework supported. The Netfilter is a subsystem presiding over the packets in Linux, such as packet mangling, packet filtering, connection tracking and network address translating (NAT). The Netfilter also provides hooking mechanisms to allow programmers alter and augment Netfilter behaviors. These hook positions are summarized in Table 1. We analyze three feasible content-based filter approaches that can implement on a Linux home gateway.

TABLE 1. NETFILTER HOOKS

Hook	Description
PRE_ROUTING	Incoming packets pass this hook before they are processed by the routing code.
LOCAL_IN	All incoming packets addressed to the local computer pass this hook.
FORWARD	All incoming packets not addressed to the local computer pass this hook, packets to be forwarded and leaving the computer over a different network interface.
LOCAL_OUT	All outgoing packets created in the local computer pass this hook.
POST_ROUTING	the last chance to access all outgoing (forwarded or locally created) packets before they leave the computer over a network device.

A. Modifying the code of Linux NAT

In order to prevent system resources from over occupied by processes, modifying kernel codes of Linux home gateway is a straightforward way to design our content filter. In this approach, we can add filtering mechanism in NAT (Network Address Translation) functions. Linux NAT is a kernel procedure in network layer. The Linux-based home gateway will have a content filter function if the filter mechanism is embedded in the Linux kernel. However, it is not easy to maintain the patch due to frequent Linux kernel source variation, for instance, the NAT functions which implemented in `ip_nat_*.c` changed to `nf_nat_*.c` when the Linux version 2.6.22 was released.

B. Processing queueing packet through the Netlink to User Space

Except straight modifying kernel codes of Linux, building a Netlink channel between the Linux kernel space and user space is another approach. With the Netlink channel, message communication and data exchange are achieved between kernel space and user space. When the packets arrive at the

Linux kernel, they were mirrored and queued in the Netlink channel with the FIFO (First In First Out) queuing discipline. The Linux kernel then notifies the user programs to dequeue the packets and process the packets with filtering, blocking and other treatments in the user space. After processing the packets, the user programs notice the Linux kernel module whether to allow the packet directly pass, or replace the packets filtered by the user programs and sent the modified packets.

C. Hooking in Netfilter chains

In the third approach, we seek to the solution by designing and building our filter as a kernel module. In addition, we alter procedures of PRE_ROUTING and POST_ROUTING (As depicted in Table I). We make our new kernel module to be a callback function of these hooks. As a result, when the packets arrive in the kernel space of Linux, the filter procedure is called to process the data packets. However, the kernel program still has some limitations such as strict requirements about dynamic memory allocation and accessibility to file systems and libraries.

According to the analysis of three aforementioned feasible approaches, we develop and implement content-based filter as a Linux kernel module by hooking in Netfilter chains and using the Netlink processing as a control group.

IV. SYSTEM DESIGN

To assist readers understand our new content-based filter system embedded on Linux home gateway, we describe our system architecture and present details of system design in this section.

A. System Architecture

As illustrated in Figure 2, our system architecture consists of two parts: modification in user space and kernel space of Linux home gateway. We built the filtering module in the kernel space. In user space modification, we designed various web-based management interfaces to assist system users.

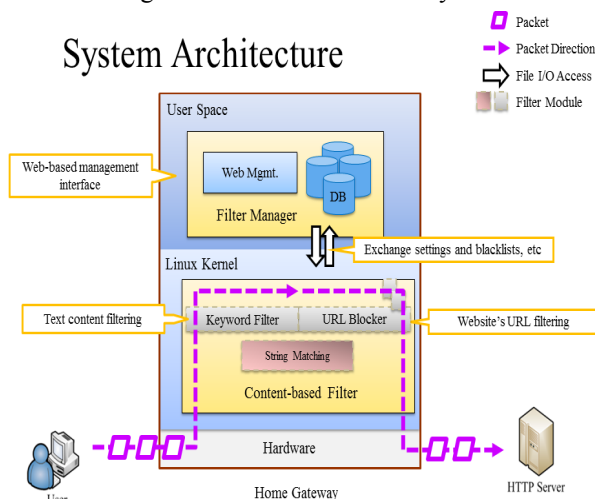


Figure 2. System architecture

We also created management functions such as data storage, system configuration, black lists and white lists in the user space of Linux home gateway. On the other hand, in the kernel space modification, two different content-based filter strategies URL Blocker and Keyword Filter were designed and implemented. The detail descriptions of two major filter strategies are described as follows:

1) URL Blocker: By extracting HTTP requests, the URL Blocker-based content filter can retrieve the URLs that system users are connecting. In addition, the requested URLs are compared with black lists and white lists. As a result, the content filter will decide whether to directly drop the request or accept the connection without doing any remaining checks. If the requested URLs are filtered and blocked, the content filter will alarm and inform the users with reasons that the requested URL connection is refused and then disconnect the connection.

2) Keyword Filter: This function modifies requests sent from network users' web browsers. Furthermore, the keyword filter removes three fields: Accept-Encoding, If-None-Match, If-Modified-Since to ensure that the responses are in plaintext so that the filter can perform continuous text filtering. The keyword filter also reads keywords blacklist and performs text matching when data packets arrive. If any keywords in black lists are matched, the filter will replace the text into messy codes such as "****". In addition, recalculation mechanism including IP and TCP checksums are performed and the keyword filter sends the modified web content back to the users.

B. System Flow

To help readers understand how our content filter based Linux home gateway works, we present system flowchart as illustrated in Figure 3. The system flow begins from the start state.

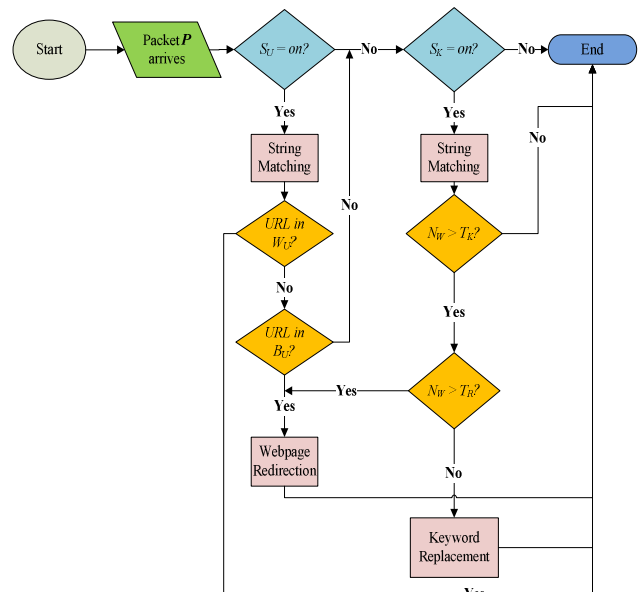


Figure 3. System flow

When a packet P arrive the filter system, the filter system check the S_U is on or off. The S_U is the URL filter function switch. If the S_U is set on, the URL Filtering function catches the HTTP requested URL from data packet P content and performs the string matching with URL black list. The URL filtering function uses string matching to find out whether the URL from packet P is in the black list. If a matched URL is found in the blacklist, the filter will identify this URL containing obscene or malicious web pages and perform webpage redirection. In addition, the filter will inject redirection JavaScript code in the packet P and recalculate packet P checksum to notify the user that the website connection is terminated with warning webpage configured by manager, otherwise the system flow is continue.

After the URL filtering, the system checks the S_K is set on or off. The S_K is keyword filtering function switch. If the S_K is set on, the keyword filtering function will perform the keyword matching between the packet P content and the keyword obscene database. After the keyword matched, a number N_W is obtained.

Moreover, the keyword replacement with messy codes will be conducted when the value of N_W is between the number T_K and T_R . The number T_R and T_K represent the high and low threshold value. The number T_R and T_K are set the 10% and 60% keywords represent in the webpage now. In the keyword replacement, the filter finds the obscene keyword in packet P content and replaces the obscene keyword with the string “***” repeatedly until no obscene keyword in packet P content is left. After the replacement, the filter recalculates packet P checksum. If the N_W is greater or equal to the T_R the filter then acts the webpage redirection. Otherwise, if the N_W is less than the T_K , the filter will identify the packet P content in the webpage containing no obscene and malicious information and end content filter service.

V. SYSTEM IMPLEMENTATION

To implement content-based filter system on the embedded linux home gateway, we build the testing environment on the VMware Server ESXi which is mainly using the Debian Linux 5.04 operating system. The kernel version is 2.6.26.2 and the cross-compiler is a PowerPC version. Moreover, we chose a Linux embedded development board to compile and transplant source code. In addition, we built a NAT environment on the ESXi virtual machine to simulate a home network environment, in which the users of ESXi virtual machine can surf the Internet through the NAT.

The test environment is used to implement the content-based filter in the kernel module. After completing all of filter module, we transplanted and integrated the content-based filter module to the Linux embedded development board. We then conducted actual test and performance evaluation. The equipment for system development is listed in Table 2 and Table 3.

VI. SYSTEM EVALUATION

We examined the pornographic content filtering service in our Linux home gateway. The URL blocking module and

keywords filtering module indeed can effectively filter webpage containing pornographic words and pictures. Figure 4 shows the snapshot of web browser when URL blocker module is off. As illustrated in Figure 5, when the URL blocker module is enabled, the webpage that containing pornographic information was filtered and replaced with warning messages to inform the user that the obscene information has been blocked by filtering services.

Figure 6 shows snapshot of web browser when the keyword filter is enabled. There are many filthy words that circled with red color. Once the keyword filtering module is enabled, the filtering service replaced those pornographic key words with messy codes “***” as shown in Figure 7.

TABLE 2. DEVELOPMENT HOST

Use	Content filtering system development and cross-compiler
Equipment	DELL OptiPlex 760 PC, Intel Quad 2.66G CPU, 8GB memory, 500GB hard disk
Operating System	Debian Linux 2.6.26.2 i686

TABLE 3. EMBEDDED LINUX DEVELOPMENT BOARD

Use	Home gateway with content-based filtering
Equipment	PowerPC 333 MHz CPU 、128MB Memory
Operating System	Linux Kernel 2.6.22.19 PowerPC

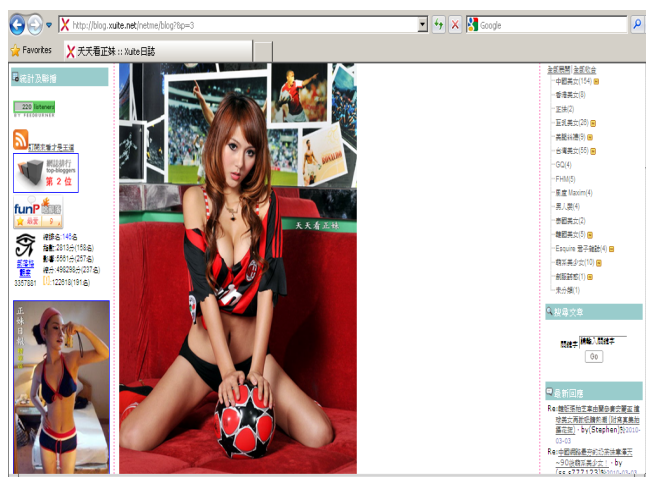


Figure 4. URL blocker module is unable

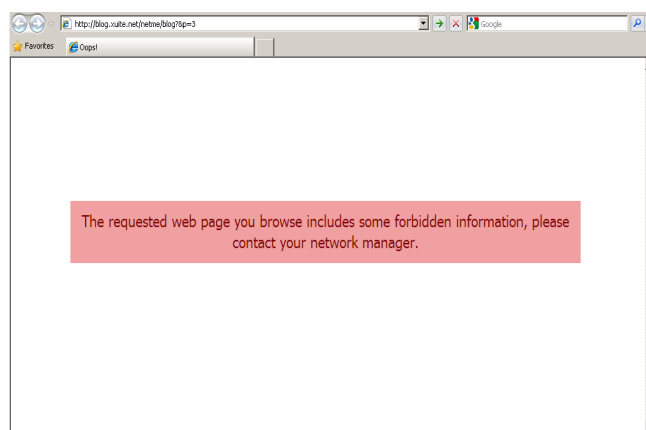


Figure 5. URL blocker module is enable



Figure 6. Keyword filter module is unable



Figure 7. Keyword filter module is enable

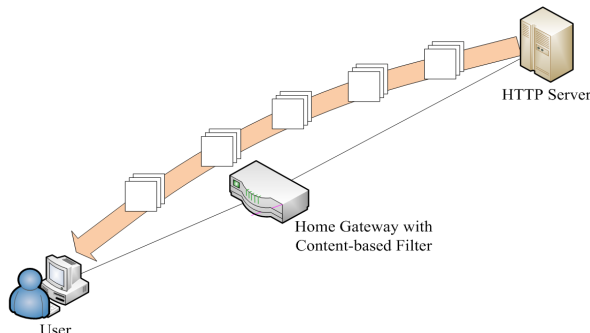


Figure 8. Evaluation environment

We conducted extensive experiments to evaluate our content-based Linux home gateway. The experiment environment is shown in Figure 8. When network users are surfing Internet via a home gateway with implemented our content-based filter system, the filter system will inspect the packet content and perform filtering action. The experiment equipment is summarized in Table 4. System experiments process and results are discussed as follows.

A. URL Blocker performance test

Filtering performance including processing rate of HTTP requests and CPU utilization over the number of URLs were studied.

The famous hash matching algorithm – Jenkins was respectively implemented on user space and kernel space of content-based filter. Figure 9 shows that running URL Blocker in kernel space has better processing rate than that of running in user space. Figure 10 shows the utilization of CPU using the string matching algorithm in the kernel space and user kernel when matching with different blacklist. When running URL Blocker in the user space, the utilization of CPU is quickly increased to 80%. The reasons are that when running URL Blocker in the user space, data packets need to go through from kernel space to user space and then kernel space from user space as the packet filtering is finished.

TABLE 4. EXPERIMENT EQUIPMENTS

Target	Freescall MPC8323E RDB	Asus M5200
CPU	PowerPC 333MHz	Intel P4M 1.5GHz
RAM	128MB	512MB
OS	Linux 2.6.22.19	Linux 2.6.32.5
Library	libgd-2.0.36 libcv-1.0.0	

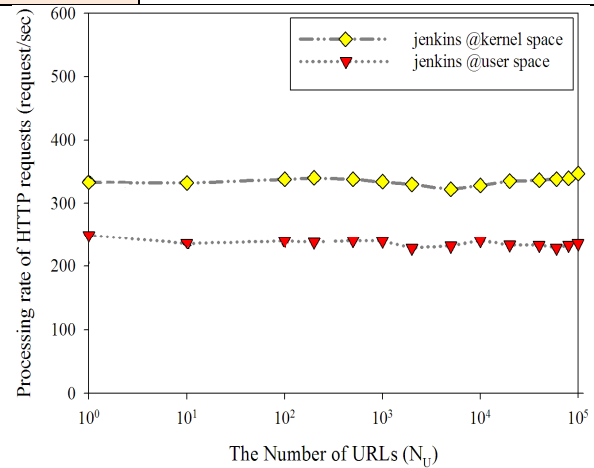


Figure 9. The evaluation result of URL Blocker about Processing rate of HTTP requests.

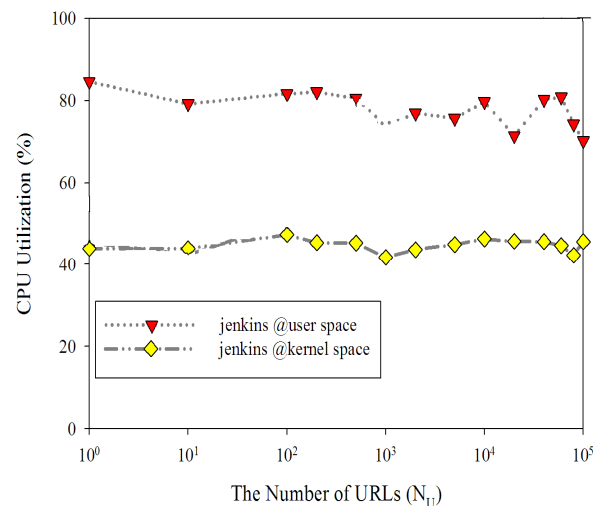


Figure 10. The evaluation result of URL Blocker about CPU utilization requests

This process consumes lots of times and system resources in the embedded system. Therefore the URL blocker in kernel space has high performance.

B. Keyword Filter

In the keyword filter experiment, two string matching algorithms Boyer–Moore (BM) and brute force matching - STRSTR were implemented in our filter system. The keyword blacklist contains 100 keywords to 2000 keywords. The experiment method is to fetch a 100 megabyte plain-text file from remote HTTP server and then evaluate the throughput and CPU usage.

Figure 11 shows that when the number of keywords is more than 250, as a string matching algorithm, the BM algorithm demonstrate a better performance than taking STRSTR when the number of keywords is more than 250. The BM algorithm's time complexity of BM algorithm is $O(n)$, compared without the time complexity of STRSTR algorithm's time complexity is $O(n^2)$. Figure 12 shows that the BM string matching algorithm has the lower utilization of CPU range (from 80% to 85%) than the STRSTR range (from 85% to 90%). The interesting phenomenon is reason why the utilization of CPU with the keywords which is less than 250 is higher than that of utilization of CPU with the keywords which is more than 250.

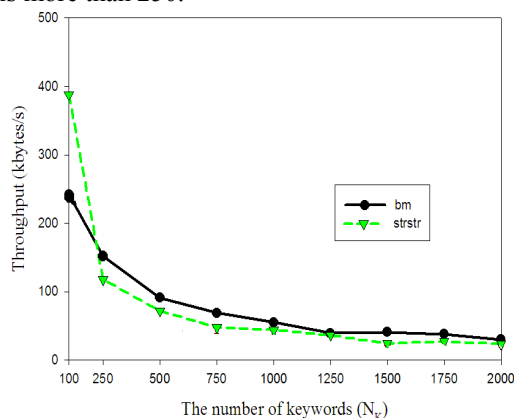


Figure 11. The evaluation result of Keyword Filter throughput.

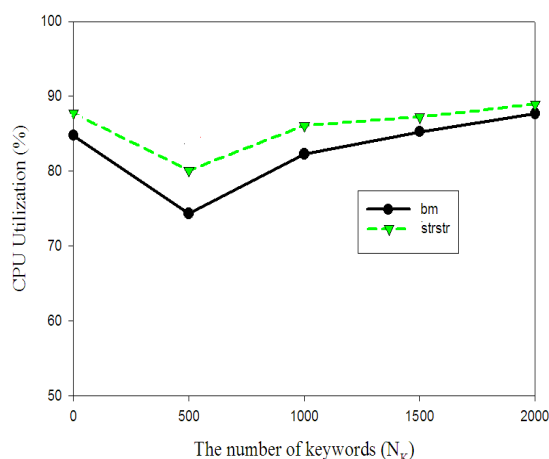


Figure 12. The evaluation result of Keyword Filter CPU utilization.

The reason is that few keywords matching will cause relative efficient filtering and high transmission rate. In such a situation, the interruption of network adapter is increasing.

VII. CONCLUSION

To sum up, in this paper, we designed and implemented a content-based filter system on embedded Linux home gateway with useful filtering functions such as URL Blocker and Keyword Filter. We extended personal home gateway with network security features such as Internet content filtering. We examined our new content based filter system with system experiments. According to the results of performance evaluation, running filter in kernel space can get extra 35% performance improvement than in user space. In the future, we may consider using the machine learning approach which is implemented in the semantic recognition system with the classifier to improve filtering efficiency and accuracy.

ACKNOWLEDGMENT

This paper is fund in part by Hsinchu Science Park, Taiwan under the grant number of 99RA06 and was supported by the National Science Council of Taiwan, under Contracts NSC 100-2218-E-008 -012 -MY3 and NSC 99-2221-E-008-041-MY3. Special thanks to research assistant Yu-yan Lin for her enlightened and helpful advices.

REFERENCES

- [1] (2011) T WNIC Statistics. [Online] Available: <http://statistics.twnic.net/tw/>
- [2] K. Rojanakul and L. Hu, "Initiatives and Challenges in Countering Inappropriate Internet Use," in *Proc. ICCTD '09*, 2009, vol.2, pp.516-520, 13-15, Nov. 2009.
- [3] W. Rong, "Intelligent network for digital home network gateway," in *Proc. ICTM '09*, 2009, vol.2, pp.363-366, 5-6 Dec. 2009.
- [4] L. P. eng, B. Zheng, L. Hong, L. Lin, "An experiment teaching of NAT application in firewall," in *Proc. ICCSN '11*, 2011, pp.126-128, 27-29 May 2011
- [5] Y. Liu and T.Wang, "Technical Standard System Development for the Internet Content Rating Service," in *Proc. ISA '11*, 2011, pp.1-5, 28-29 May 2011.
- [6] T.M. Chen, V.Wang, "Web Filtering and Censoring," *IEEE Computer Society*, Vol. 43, Issue 3, pp. 94-97, 11 Mar. 2010.
- [7] V. Varadharajan, "Internet Filtering Issues and Challenges," *IEEE Computer Society*, Vol. 8, Issue 4, pp. 62-65, 26 Jul. 2010.
- [8] S. Kim, H. Min, J. Jeon, Y. M. Ro and S. Han, "Malicious Content Filtering Based On Semantic Features," in *Proc. ICIS '09*, 2009, pp. 802-806, 2009.
- [9] J. M. G. Hidalgo, F. C. Garc'ia, and E. P. Sanz, "Named Entity Recognition for Web Content Filtering," in *Proc. NLDB '05*, 2005, vol. 3513, pp. 286-297, 15-17 Jun. 2005.
- [10] M. Kodialam, T. Lakshman and S. Sengupta, "Configuring networks with content filtering nodes with applications to network security," in *Proc. INFORCOM '05*, 2005, vol.4, pp. 2395-2404, 13-17 Mar. 2005.
- [11] M. G. Noll and C. Meinel, "Building a Scalable Collaborative Web Filter with Free and Open Source Software," in *Proc. SITIS '08*, 2008, pp. 563-571, Dec. 2008.
- [12] (2011) HiNet Hicare [Online] Available. <http://hicare.hinet.net/>
- [13] (2011) Green Dam Youth Escort [Online] Available, <http://www.lssw365.net/>
- [14] F.A. Maulana, S. Abdulmana, F. Alfariati, "Cyber Security Malaysia and Malaysia Department of Islamic Development (JAKIM) as the content advisor," in *Proc. URKE '11*, 2011, vol. 1, pp. 12-15, 4-7 Aug. 2011.