# Low-End Embedded Linux Platform for Network Security Application – Smurf Based Attack Detection

N. Ahmed[1], Z. I. A. Khalib[2], R.B. Ahmad[3], Suhizaz Sudin[4], Salina Asi[5], Yacine Laalaoui[6]

School of Computer and Communication Engineering, Kompleks Pusat Pengajian, University Malaysia Perlis (UniMAP),
No. 12 & 14, Jalan Satu Taman Seberang Jaya, Fasa 3, 02000 Kuala Perlis, Malaysia.

## ABSTRACT

Embedded systems are becoming a main solution to many specific tasks because of this high stability, minimal power consumption, portability and numerous useful. Nowadays, many new applications are developed using embedded system. This paper presents the possible usage, design and implementation on embedded Linux platform system for Intrusion Detection (Smurf Attack Detect). By applying these methods the embedded system is able to identify Smurf attack and analyze ICMP traffic. The software is executed on a Linux based Single Board Computer (SBC) which run TS-Linux 2.4.23 kernel. Results show that the Embedded Security Scan Detector (ESSD) unit managed to identify possible attack besides running on relatively low-end embedded platform. It is significant that network security product develop on embedded Linux has a very high market potential. Our test of the new systems shows satisfactory results for monitor and analyzes ICMP traffic and Smurf Attack detecting activity under such hardware limitations.

. **Keywords—** Embedded System, Computer Security, DDoS Attack and Smurf Attack.

## I. Introduction

Embedded system is a system that is designed to serve specific tasks. Almost all embedded systems come in compact size, so users are able to use them as additional parts to other devices or to construct specific applications with them. Embedded systems have many advantages like high efficiency, long life usage, and economical energy consumption. Embedded systems have become ubiquitous as can be found in many new devices and systems such as cellular phones, PDAs and wireless networks. Older technologies also reap the benefits of embedded processing, for example a typical automobile now includes two–dozen microprocessors [1], Over 98% of all microprocessor are now deployed in embedded systems [2]. Unfortunately, security research targeting resource–constrained distributed embedded systems has not kept pace with the growing application of embedded systems. Distributed Denial of Service (DDoS) attacks continue to be a prominent threat to cyber infrastructure. A DDoS attack [3, 4] involves multiple DDoS agents configured to send attack traffic to a single victim computer to exhaust its resources. DDoS is a deliberate act that significantly degrades the quality and/or availability of services offered by a computer system by consuming its bandwidth and/or processing time. As a result, the legitimate users are unable to have full quality access to a web service or services. This may also include data structures such as open file handles, Transmission Control Blocks (TCBs), process slots etc. Because of packet flooding in a DDoS attack that typically strives to deplete available bandwidth and/or computing resources, the degree of resource depletion depends on the traffic type. DDoS attacks today are part of every internet user's life. The sole purpose of DDoS attacks is to disrupt the services offered by the victim. DDoS attacks can take several forms and can be categorized by several parameters, which can be classified based on how they affect a victim computer or based on how they are generated [5]. According to Computer Emergency Response Team Coordination Center (CERT/CC) [6], there has been an increase in use of Multiple Windows-based DDoS agents. There has been a significant shift from UNIX to Windows as an actively used host platform for DDoS agents. Furthermore, there has been increased targeting of windows end-users and servers. The CERT/CC published a tech tip entitled "Home Network Security" in July of 2001 [7] to raise awareness of such vulnerabilities. According to the CERT/CC [6], there is a perception that windows end-users are generally less security conscious, and less likely to be protected against or prepared to respond to attacks compared to professional industrial systems and network administrators. Furthermore, large populations of windows end-users of an Internet Service Provider are relatively easy to identify and hence the attackers or intruders are leveraging easily identifiable network blocks to selectively target and exploit windows end–user servers and computer systems.

The remainder of this paper is organized as follows Section II describes the Smurf Attack methods in literature. Section III describes Smurf Attack Diagram. Section IV describes the system Architecture. Section V discusses the test results discussion of the system and performance. Lastly Sections VI concludes the paper.

## II. Smurf Attack

Smurf Attack is a type of well known DDoS attack where an attacker exploits packets unprotected computers on Internet to direct a flood of ICMP echo-reply messages towards the victim computer. Primarily Smurf Attack exploits the ICMP messages that are among the most commonly used diagnostics tools frequently used to troubleshoot problems in a network [8]. A computer system that receives an ICMP echo request message is to respond by sending an ICMP echo reply message back to the sender. The packet format used by the ICMP echo request and echo reply shown in Fig. 1 By the value of the type field the ICMP echo request and echo reply messages are identified. The echo request has the TYPE filed value = 8 where as the echo reply has the TYPE field value = 0. The OPTIONAL DATA field holds data that are returned to the sender by the receiver of the ping messages. The IDENTIFIRE and the SEQUENCE NUMBER fields are used to match the request and reply messages.

0 7 8 15 16 31

| | | |
|---|---|---|
| TYPE | CODE (0) | CHECKSUM |
| IDENTIFIER | | SEQUENCE NUMBER |
| OPTIONAL DATA | | |
| ………………………………… | | |

**Figure 1 ICMP Echo Request/Reply Message Format**

Both ICMP echo request and ICMP echo reply messages are used in Smurf Attack. A perpetrator sends a large amount of ICMP echo (ping) traffic to the IP broadcast addresses, all of it having a spoofed source address of a victim. If the routing device delivering traffic to those broadcast addresses perform the IP broadcast to layer 2 broadcast functions most host on that IP network will take the ICMP echo request and reply to it with an echo reply each, multiplying the traffic by the number of hosts responding. If the broadcast domain has N number of computers then for each echo request message sent to the broadcast domain, N number of echo reply messages are generated and sent not to the original sender but to the victim's computer (due to the spoofed source address in the ICMP echo request messages). In effect, the broadcast domain helps amplify and direct the DDoS attack traffic towards a victim computer. If more than one broadcast domains are involved then such DDoS attack traffic can be amplified even further and the victim computer is flooded with a large number of ICMP echo reply messages resulting in bandwidth exhaustion and also the resource exhaustion of the victim computer.

## III. Smurf Attack Diagram

Smurf Attack is a nasty type of DDoS attack. The attacker sends a large amount of ICMP packet to a broadcast address and uses a victim IP address as the source IP so the replies from all the devices that respond to the broadcast address will flood the victim. The attacker can use low-bandwidth connection to kill high-bandwidth connections. Fig 2 shows the diagram of Smurf attack.

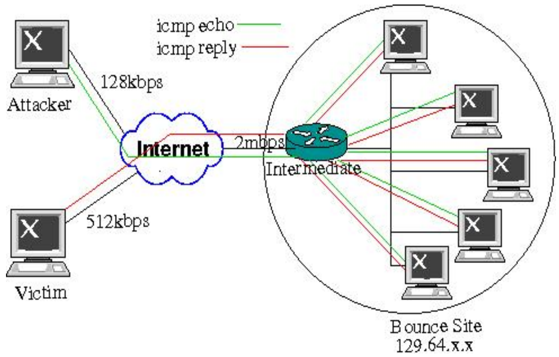

**Figure 2. Smurf Attack Diagram**

The above diagram shows a structure of Smurf Attack. The attacker sends a stream ICMP echo packets to the router at 128kbps. The attacker modifies the packets by changing the source IP address to be that of the victim's computer so replies to the echo packets will be sent to the address. The destination address of the packets is a broadcast address of the so-called bounce site.

## IV. System Architecture

### A. The Hardware Platform

Considering the focus of this paper, which is to evaluate the practicality of a low-end Embedded Linux Platform for a relatively average speed computer network application, we thus opted for the TS 5500 Single Board Computer. The board comes with TS-Linux 3.07 (2.4.23 kernel) operating system. Network supports is one important feature for this 32 bit embedded PC technology. TS5500 has one RJ45 port and support standard network by using Telnet and file transfer protocol (FTP). But it does not support Secure Shell (SSH) function. Furthermore, the Secure Copy (SCP) is allowed by this model by activating the dropbear functions provide by TS Linux. Fig 3 shows the embedded system Single Board computer (SBC) that we used. The efficiency of size, weight, cost, interchangeability, and consistency are the major factors [8] which lead to the selection of TS5500

Single Board Computer (SBC) as the hardware platform for the system.

**Figure3. Single Board Computer (SBC)**

The board comes with an AMD Elan 520 (x86 compatible) processor that runs at 133MHz and it has 64 MB of RAM. It also has a Type 1 Compact Flash card reader, USB, PCMCIA a 10/100Base-T Ethernet interface and an alphanumeric LCD and keypad interface.

## B. System Overview

The system is called Embedded Security Scan Detector (ESSD) and its task is to ensure security through incorporation of Smurf Attack Detection. Figure 4 shows a possible deployment of the Embedded Security Scan Detector. Assuming the router and firewall permit ICMP echo requests and echo replies out of the network, and ESSD is connected with configured monitor switch port from where this new system can detect abnormal behaviors and also the other systems are connected to the switch. The system is user programmable, meaning the user has the flexibility of choosing the ports that he/she would like to peep into looking for any possible malicious attack activity. The SBC which comply with the embedded PC standard, a commonly-used robotic development platform [9, 10], has a main board of approximately 4 by 4 inches that houses a processor, memory and the basic chipset needed to function as a standalone embedded computer capable of functioning with only a separate power supply and whatever outside input or output devices the application calls for. The embedded PC allows the use of an 802.11b (Wi-Fi) and wired Ethernet that provide high-speed two way communications link between the system and PC Database Server.

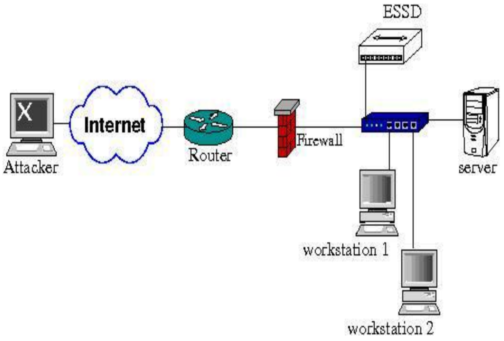

**Figure 4. Embedded Security Scan Detector Possible Deployment**

Utilizing Linux based embedded PC allows us to manipulate the availability of open source resources such as libraries, kernels and drivers in developing and implementing this system. Integration of TCP/IP network protocol within the Linux kernel running on board allows network centric application to be easily developed and implemented. The only concern is the processing speed of the embedded platform, which is generally a constraint for network application. Thus the focus of the project is to realize the possible usage of low-end embedded Linux platform for a medium speed hungry network application like Smurf Attack detection.

## C. Experimental Setup

We designed experiments to simulate attack involving real computer systems. In these experiments, a Smurf-attack was generated in a controlled environment. A Linux Ubuntu-based computer was used as the victim computer of the Smurf-attack. Table 1 shows the detail experimental setup information.

**Table 1 Desktop Experimental Setup**

| **Processor** | **Intel (R) core (TM)2 Duo** |
|---|---|
| Clock Frequency | 2.20 GHz |
| Operating System | Ubuntu 2.6.20-16-generic |
| L1 I-Cache | 32k |
| L1 D-Cache | 32k |
| L2 Cache | 2048k |
| Main memory size | 2 075772k |
| FSB (Front side bus) | 365.56 |
| Memory Bus | 609.26 |

## V. Result and Discussion

Embedded Security Scan Detector (ESSD) has been implemented on Linux 2.4.23 Single Board Computer

(SBC) and programmed in C. Developing as a low-end new ESSD for to have the benefit that the system modules are natively more secure with substantially good system performance. In addition, a lot of legacy C library code can be easily ported. The entire test was conducted on the Single Board Computer (SBC). At first, we monitor and analyze ICMP traffic in the LAN because we wanted to know what ICMP messages go through the entire network interface, whether there is much more echo reply than echo request and also whether the reply message arrive within the short period of time or not. Then we wanted to know the overall picture of our lab LAN traffic information. So we run a web based Embedded Network Monitor System which has been developed in our lab for 24 hours in order to get traffic information. Figure 4 shows the detail statistical results about network traffic information.

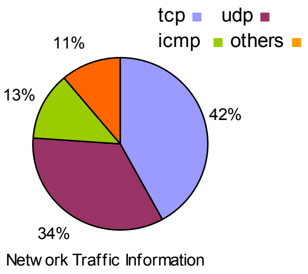

**Figure 4. Traffic Information**

It is well known that the Smurf Attack comes from ICMP protocol (echo request and echo reply). The Embedded Security Scan Detector can be used to scan all the classes of IP addresses (A, B, C). The new systems successfully detect Smurf attack from switch monitor port. For the experimental test we deployed Smurf Attack from the same gateway segment by Linux Based desktop computer. At the end, the system will send all the detected information into a file. Thus, the new Embedded Security Scan Detector system is considered to be a security scanner. Table 2 shows the new system detection information.

**Table 2 new system scan information**

| Type of IP Network | Detect Information | Time |
|---|---|---|
| Class A | 10.172.1.255 169 | 32 min |
| Class B | 10.172.1.255 301 | 46 min |
| Class C | 10.172.1.255 397 | 57 min |

**Table 3 desktop-based scan information**

| Type of Network | Detect Information | Time |
|---|---|---|
| Class A | 10.172.1.255 196 | 32 |
| Class B | 10.172.1.255 356 | 46 |
| Class C | 10.172.1.255 426 | 57 |

Table 2 and 3 shows the detail attack detection results. Table 2 shows low-end Embedded Security Scan Detector results and the new system are capable to detect malicious activities. We compare our new system with desktop pc and we consider detect time. Because of low speed Embedded System can not run fast but can detect attacks as like high speed desktop. The present new system results evaluate fairly.

The experiments present the performance of the new system ESSD. The performance of the new system is evaluated by comparing the CPU status and memory usage before and during execution of the program. The total memory of the new system is 62684k by default the system has 22 packages runs where it first start for boot the PC, and using 16900k memory. The rest 45784k memory was free. It shows the average CPU utilization before and at the time of program execution. For performance test of the new system we considered three working days with and without new software and also we compared the new system with Linux-based Ubuntu system in the same manner. The "top" General Linux command was used on Single Board Computer (SBC) and a desktop to extract the actual status of the CPU. Fig 5 (a, b and c) shows CPU utilization without any other program except the system packages.

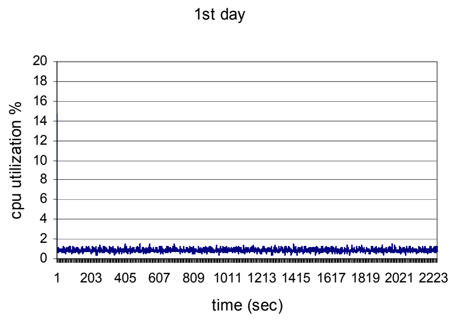

**Figure 5 (a)**

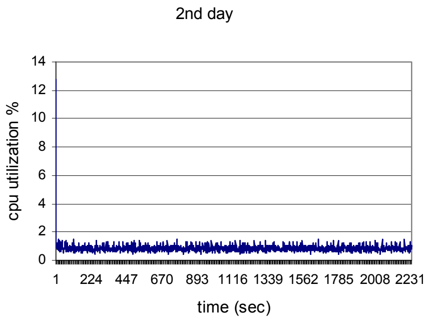

**Figure 5 (b)**

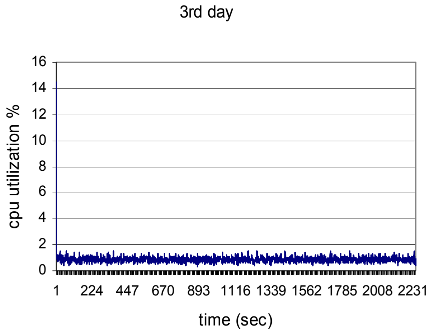

**Figure 5(c)**

Figure 5 (a, b, and c) presents Single Board Computer (SBC) averages CPU utilization. The maximum and minimum CPU utilization is 1.5% and .1% respectively when the system boot. The stability of the system is good. Fig6 (a, b and c) shows the new system Embedded Security Scan Detector (ESSD) CPU utilization at the time of execution of ICMP network monitor program. As we mention that our new hardware platform is TS-Linux 2.4 kernel and it has many limitations. Libraries is one of the big limitation because of that at first, the program dot C file has been executed on 2.6 kernel Ubuntu Linux desktop platform in the chroot environment after that the object file has been exported to the TS – Linux 2.4 kernel using general Linux “scp” command. The source code object file total length is 20.k and it does not allocate much memory.

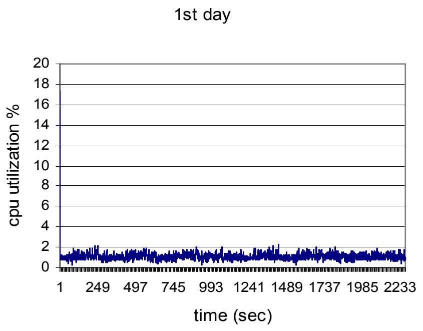

**Figure 6(a)**

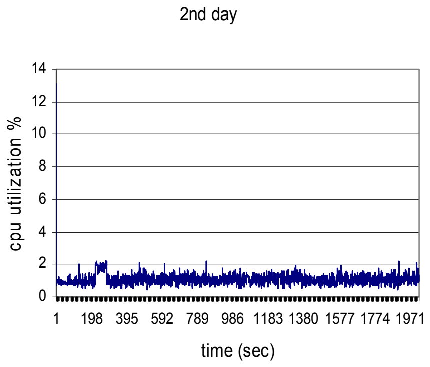

**Figure 6(b)**

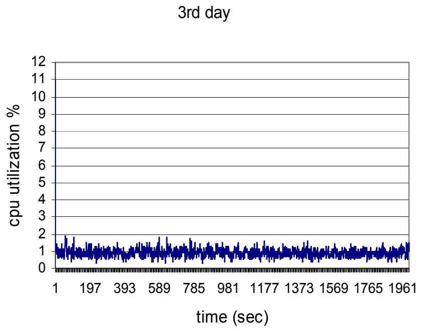

**Figure 6(c)**

The above graph 6 (a, b and c) shows the average CPU utilization when we executed our new ICMP network traffic monitor program. The maximum average CPU utilization was 2.3% and minimum 1.3%. The three days graph proves that the CPU utilization is not very high and the behaviors and performance of the new system is good which satisfy good system character.

In this section we present the Smurf Attack Detection program execution on Single Board Computer (SBC), shows the CPU utilization status. Figure 7 (a, b and c) shows new system Embedded Security Scan Detector (ESSD) CPU utilization at the time of Smurf-based Attack Detection.

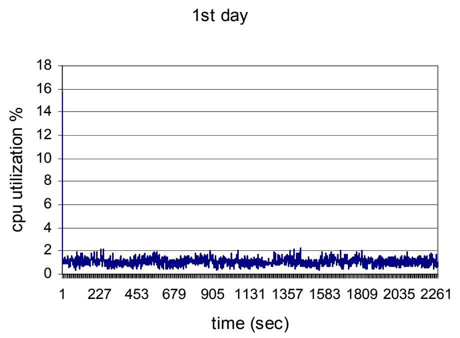

**Figure 7(a)**

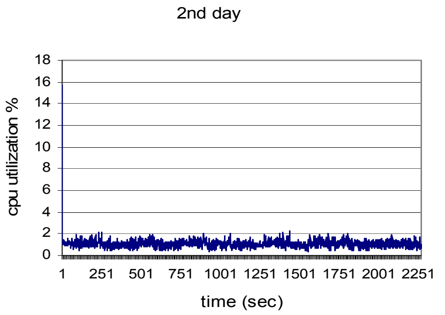

**Figure 7(b)**

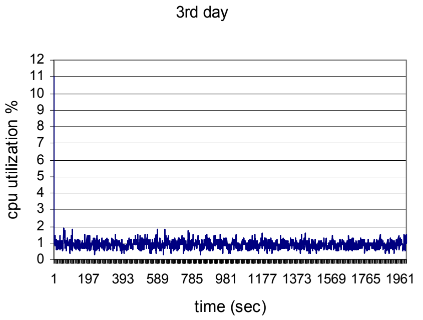

**Figure 7(c)**

At the time of program execution of the Smurf Attack detection the new system (ESSD) maximum CPU utilization is 2.0% and minimum 1.6.

The above Figure (5, 6 and 7) showed that the application does not keep the processor busy. We had compared the new system performance running on the SBC while the same applications have been executed on a PC with a Core Duo processor and 2GB RAM. Interesting enough, the new system does not fall far behind the other system and yet it managed to beat one of the systems. Total of 38 packages were running when we boot our experimental workstation. Usually, the total CPU utilization will be high. The detail comparison can be found in Figure 8 (a, b and c) with our new software running.

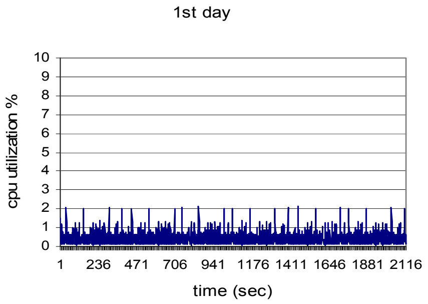

**Figure 8(a)**

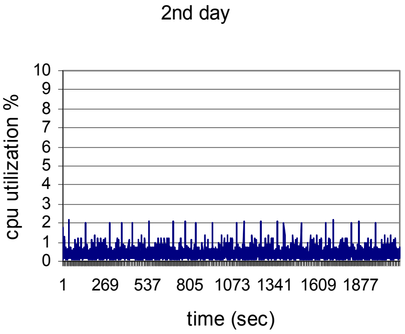

**Figure 8(b)**

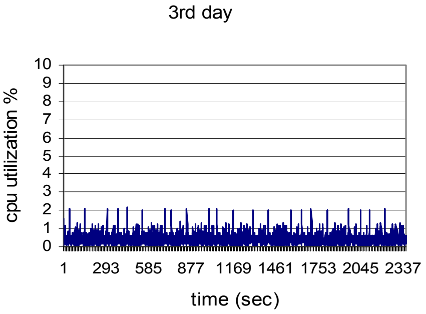

**Figure 8(c)**

The experiment shows that new system does not use much memory for processing, which a good candidate for embedded application which is known for having limitation in memory.

## VI. Conclusion

This paper presents Embedded Security Scan Detector (ESSD) for Smurf Attack Detection integrated into Low-end embedded Linux platform Single Board Computer (SBC). Based on testing performed, the developed ESSD is found to be performing at par with Ubuntu Linux Desktop which runs same application. Thus we can conclude that low-end embedded Linux platform which integrates open source TCP/IP network protocol is suitable for IPV4 application. Apart from that the inherited features of portability, low power, low cost and small size would make such product competitive.

### References:

[1] J. Turley. The Essential Guide to Semiconductors. Prentice hall, 2003, Professional technical Reference, Upper Saddle River, NJ 07458, www.phptr.com

[2] D. Tennenhouse. " Embedding the Internet: Proactive Computing," Comm. Of the ACM, May, 2000

[3]. Lee Gerber, "Denial of Service Attacks Rip the Internet," IEEE Computer, April 2000

[4]. "Smurf IP Denial-of-Service Attacks," CERT® Advisory CA-1998-01, March 2000.
http://www.cert.org/advisories/CA-1998-01.html

[5].Siliva Farraposo, Laurent Gallon, Phillippe Owezarski, "Network Security and DoS Attacks," Feb – 2005.
http://www.cert.org/reports/dist_workshop.pdf

[6] Kevin J. Houle and George M. Weaver, "Trends in Denial of Service Attack Technology," Computer Emergency Respons Team (CERT)® Coordination center, v1.0, October 2001

[7] Computer Emergency Response Team (CERT)® Advisory CA-2001-20, Home Network Security,
http://www.cert.org/tech_tips/home_netwoks.html

[8] J. Xu and W. Lee, "Sustaining Availability of Web Services under Distributed Denial of Service Attacks," IEEE Transactions on Computers, Vol. 52, Feb 2003

[9] M. D. Schiffman, "Biulding open Source Network Security Tools Components and Technique," Willy Publishing, Inc. ISBN 0-471-20544-3, pp 217-218.

[10] Fyodor. http://www.insecure.org/nmap

[11] TS-5500 PC/104 SBC with AMD 586 Processor. Citing Internet Source, URL http://www.embeddedarm.com/epc/ts5500-spec-h.html

**Nasim Ahmed** received Computer Science degree from University of Madras, Chennai, India in 2003. Currently, he is a graduate student at School of Computer and Communication Engineering, University Malaysia Perlis (UniMAP), Malaysia. His research interest is Embedded System Based on GNU/Linux for Network Security and Intrusion Detection.

**Associate Professor Dr. R. Badlishah Ahmad** is a Dean at School of Computer and Communication Engineering, University Malaysia Perlis (UniMAP). He received his degree in B. Eng (Hons) from University of Glasgow, Scotland in 1994. Master of Science (M.Sc) and PhD from University of Strathclyde, Glasgow, Scotland in 1995 and 1999 respectively. His current research interest includes Modeling & Simulation of Computer and Optical Network, Embedded System Based on GNU/Linux for Vision System, Data Acquisition and Network Security.