Study on Auto Detecting Defence Mechanisms against Application Layer Ddos Attacks in SIP Server

Muhammad Morshed Alam
Department of Electrical and Electronic Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh
Email: morshed380@yahoo.com

Muhammad Yeasir Arafat, Feroz Ahmed
Department of Electrical and Electronic Engineering, School of Engineering and Computer Science, Independent University, Bangladesh
Email: yeasir08@yahoo.com, fahmediub@gmail.com

Abstract—Denial of Service (DoS) or Distributed Denial of Service (DDoS) is a powerful attack which prevents the system from providing services to its legitimate users. Several approaches exist to filter network-level attacks, but application-level attacks are harder to detect at the firewall. Filtering at application level can be computationally expensive and difficult to scale, while still creating bogus positives that block legitimate users. In this paper, authors show application layer DoS attack for SIP server using some open source DoS attack tools and also suggest a mechanism that can protect a given SIP server from application-level DoS attacks especially the attacks targeting the resources including CPU, sockets, memory of the victim server. In this paper author’s attempt to illustrate application layer distributed denial of Service (DDoS) attack on SIP Server such as SIP flooding attack, real time transport (RTP) flooding attack using open source DDoS attack tools. We propose a new DDoS defence mechanism that protects SIP servers from application-level DDoS attacks based on the two methodologies: IPtables and fail2ban detection. The attack flow detection mechanism detects attack flows based on the symptom or stress at the server, since it is getting more difficult to identify bad flows only based on the incoming traffic patterns. A popular software known as Wireshark which is a network protocol analyzer is used to capture the packets during DoS attack from the victim server Ethernet interface to detect the attacking host IP address and analysis the types of attack. We evaluate the performance of the proposed scheme via experiment.

Index Terms—Application Layer DDoS; VoIP; SSH; SIP; SDP; UDP; TCP; RTP; ICMP; Fail2ban; Iptables

I. INTRODUCTION

DoS attack is a malicious attempt to disrupt the service provided by networks or servers. The power of a DoS attack is amplified by incorporating over thousands of zombie machines through bonnets [1] and mounting a DDoS attack. Leveraging botnets and high-speed network technologies, modern DoS attacks exceed the scale of 300 Gbps becoming a major threat on the Internet [2]. Being one of the oldest type of attacks on the Internet, DoS attacks are known for their disruptiveness and ability to deplete the computing resources and/or bandwidth of their victims in a matter of minutes. Although many defense mechanisms have been proposed to counter DDoS attacks, this remains a difficult issue, especially because the attack traffic tends to mimic normal traffic recently.

VoIP may be the fastest growing technology that includes routing voice conversations over IP-based network. The flexibleness of the VoIP system convergence of voice and data networks brings with it additional security requirements. SIP [3] servers are vital network elements that endow SIP endpoints to switch messages, register user position, and faultlessly move between networks. SIP servers permit network operators to determine routing and security principles, authenticate clients and control user locations. SIP server applications might take many forms, however the SIP standard defines three general kinds of server functionality that applies to all or any proxy, redirect and registrar servers. SIP server submissions may take numerous forms, but the SIP benchmark defines three general kinds of server functionality that applies to all proxy, redirect and registrar servers. Since SIP is an open source code with a simple structure and high expansibility, SIP servers are more vulnerable to attack by SIP messages malformed in order to stunt the server, or by a flood of SIP messages causing server congestion or shutdown. One of the very most common attacks is denial of service (DoS). After a session has been established with SIP the particular media transfer is transmitted with the true time transport protocol (RTP). In transport layer SIP uses user datagram protocol (UDP). Today it becomes challenge to provide extreme level security for the VoIP server [3].

Today, the most common SIP attacks take the form of flooding attacks, in which a large number of SIP Invite/Cancel/Bye messages are sent to the SIP server, intended to slow or even shut the server down. Based on the facts that UAs must frequently register with a server
in order to update their location, and that the server should respond with an OK message. Chen et al. [4] established a white-list to prevent attacks from unregistered users. The authors in [5] used finite state machines to simulate the various states of the SIP protocol; when repetitions of a certain state are beyond the set threshold, this could indicate a flooding attack.

Zhou et al. [6] focused on protecting SIP register servers from CPU-exhausting DoS attacks, because register servers must perform identity confirmation and decryption/encryption computation. When connection requests increase considerably, the proposed mechanism will check the record of previous legitimate calls. If the suspect UAC is not in the list of legitimate calls, it will be regarded as an attacker, and the services will be blocked. The authors in [6] applied the Bloom filter algorithm [7] to record and check users’ IP addresses. The Bloom filter is a lightweight algorithm that consumes very little CPU resources.

Geneiatakis et al. [8, 9] also applied the Bloom filter algorithm to determine whether an SIP server is under a DoS attack. They used the Bloom filter algorithm to count the number of Invite messages, and corresponding responses. If the Invite messages and the response messages differ significantly in number, it is regarded as a flooding attack.

The remainder of this paper is organized as follows. We first discuss application layer DDoS attack in Section 2. Section 3, we showed DoS attack on SIP. In Section 4, we showed DoS attack on SIP and also showed effect of DoS attack on SIP server. In Section 5, we propose a mitigation technique based on Iptables and Fail2ban detection. We also showed how to save SIP server using Linux Iptables rules called fail2ban and showed how to find the BOT from log file. Finally, conclusions are presented in Section 6.

II. APPLICATION LAYER DDoS ATTACK

DoS or DDoS is an endeavor to make a server or network resource unavailable to its legitimate users by flooding the server request queue with lots of false requests. The power of DoS attack is enlarged by creating attack using thousands of zombie machines which known as DDoS attack. DoS attack also responsible for seize bandwidth, high memory and Central Processing Unit (CPU) utilization processing of victim server. These fake requests can be created by using some open source DoS attack tools or scripts. At present DoS attack becomes one of the most powerful and common attacks in cyber world due to availability of free open source DoS attack tools like Low Orbit Ion Canon (LOIC), Slowloris [4], Open Web Application Security Project (OWASP) DoS HTTP Pos, DDOSIM, are-u-dead-yet (rudy) etc. If DoS attack is launched at the application layer, then the attack can be effective with a small number of packets. For example, DoS attack can be effective even with a low traffic rate after the appearance of low-rate attack tools such as slowloris. Application Layer DoS attack can be achieved many ways such as Application crashing. Data destruction and Resource depletion. Application crashing is a common way of performing DoS attack where certain types of inputs may defer an error in the application which it did not expect and will cause victim server to crash

Buffer Overflows
Malformed data – causing parser exception
Terminating with error
SQL Injection

Resource Depletion simply utilizes very large amounts of victim server resources which includes-Memory, CPU and Disk Space. In the Web application world, a DoS attack aims to down the site in order to make it unreachable to its users. When a server system is being flooded from fake requests which are coming from numerous sources, it calls a DDoS attack. Main Purpose of DoS attack is to temporarily or forever interrupt or suspend services of a host connected to the Internet. Methods to bring out this attack may differ as given bellow [10]

Saturating the victim with external communications fake requests which can be generated using DoS attack tools such that it cannot reply to legitimate traffic or simply creating server overload.

May include malware to max out target resources such as CPU, trigger errors, or crash the operating system.

The procedure of sending and receiving data from one host to another host, data encapsulation is possible due to the existence of a layer 7 protocol like HTTP, SIP, SMTP, DNS, SSH, FTP etc. as given in the application layer of the above OSI model. Application layer based DoS attacks are expected more dangerous due to their network-based counterparts they take benefit of application layer protocol behaviors but unlike their network-based counterparts it requires far fewer clients to overwhelm a host due to this reason application layer based DoS attacks are so tough to detect. This paper authors will discuss application layer DoS attack like HTTP Get request using different DoS attack tools such as slowloris and OWASP HTTP slow post and slow header, SIP [11] Invite request flooding tool with RTP flooding using some built in tools of the Operating System Back Track, Web application attacks with its effects on CPU and memory resources of victim server. There are different types of attacks likes volume based attacks includes UDP floods, ICMP floods and other spoofed-packet floods the attack’s goal is to saturate the bandwidth of the attacked site Magnitude is measured in Bits per Seconds (Bps). Protocol based attacks includes SYN floods, fragmented packet attacks, Ping of Death This type of attack consumes actual server resources Measured in Packets per Second (Pps). Application Layer Attacks includes Slowloris, Zero-day DDoS Attack, DDoS attack that target Apache, Windows or OpenBSD vulnerabilities and more Compromised of seemingly legitimate and innocent requests, the goal of these attacks is to crash the web server Magnitude is measured in Request per Second. Over the past few years, the size and frequency of DDoS attacks have grown dramatically as attackers take advantage of botnets and other high-speed...
Internet access technologies to overwhelm their target’s network infrastructure. In fact, according to Arbor’s sixth annual Worldwide Infrastructure Security Report, the largest-recorded DDoS attack has grown ten times in size from 2005 (10 Gbps) to 2013(300 Gbps). To make matters worse, in Fig.1 the report also highlights a growing new trend with DDoS attacks. Not only are DDoS attacks getting larger and more frequent, but they are also becoming more sophisticated as they pinpoint specific applications (e.g., DNS, HTTP or VoIP) with smaller, stealthier attacks [2].

Figure 1. Application-layer attacks are on the rise, according to Arbor’s sixth annual Worldwide Infrastructure Security report

III. DDoS Attack on SIP

Session Initiation Protocol (SIP), an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with a number of applicants. The VoIP service basically contains user agents (UA) and servers. UAs refer to end user devices; they can be software applications or hardware, like SIP telephones. User agent clients (UAC) are the parties that initialize the connection, namely the caller, while user agent servers (UAS) are the callees. Every UA must register with an SIP server via a Register message. If a UA is authorized, the server will respond with a 200-OK message. In general, a server may start a validation procedure by responding with a 401-Unauthorized message with a nonce value in the header field of the message. The UA uses this nonce value to generate an encrypted digest and put it in the header field, and then re-transmits the Register message. Upon receiving the message, the server will compare the encrypted digest with the original digest and send the 200-OK response to the UA if no error is found. The registration procedure is shown in Fig. 2.

An SIP connection is established as shown in Fig. 2. The UAC sends an Invite message to the SIP server, which passes the request to the UAS. Upon reception, the UAS begins a ring notification, while at the same time responding to the UAC through the SIP server with a 180-Ringing message. As the UAS answers the request, it sends a 200-OK message to the UAC, and the UAC then returns an Ack (acknowledgement) to the UAS. All SIP messages pass through an SIP server. Once the SIP connection is built, subsequent Real-Time Transport Protocol (RTP) connection establishment and voice packet transmissions will not be made through the SIP server. When either the UAC or the UAS hangs up the phone, they will send a Bye message to the SIP server to indicate that the connection has been terminated. The SIP server will transmit the Bye message to the other party, and then forward a responding 200-OK message [11].

A Denial of Service (DoS) attack on voice over Internet protocol (VoIP) services can make it ineffective by causing damage to the victim network and VoIP systems availability with damaging its quality of service (QoS). From the Fig. 2, usually hacker will send lots of fake SIP INVITE requests within a very short period to the victim SIP server to consume all its resources and also in order to reduce its bandwidth too. Due to receiving lots of fake SIP INVITE authentication requests within a second victim SIP server cannot negotiate to its registered SIP user agents and also those legitimate users who want to register to the SIP server with original authentication due to increasing service asterisk processes with high CPU and memory utilization in victim SIP server and also to reduce its bandwidth [11].

Figure 2. Victim SIP server condition during invite flood DoS attack

Every SIP message contains a command line, a header and a body. The command line is used to confirm the type of message. There are six types of requests, and six types of responses. The six types of request message include Invite, Bye, Ack, Options, Cancel and Register, while the six types of response message are usually represented by codes including 1XX (notification response, for example 180-Ring), 2XX (success response, such as 200-OK), 3XX (call forward response), 4XX (call failed, such as 401-Unauthorized), 5XX (server failure) and 6XX (global failure). The Invite message also contains information for performing Session Description Protocol (SDP) [12], which includes the caller’s media format, address, port, etc. The callee can accept or reject the connection request.

IV. Common Parameters in Quality of Service (QoS) in IP Telephony

Here authors used SIP invite flood tool which is a built tool in Back Track Operating System that can send lots of fake authentication invite request to the victim SIP server with a few seconds. This tool exist in Back Track in the directory given below
This tool invite flood have a format of command, command syntax is given below:

```
root@bt:~# ./inviteflood [eth0] [target_extension] [Source_IP] [target_SIP server_ip] [number_of_packets]
```

Effect of this attack included, as long the tool keeps flooding the SIP server will not be able to dial phone calls. Attacker can flood the SIP server with an inexistant extension number due to this reason it generates a 404 not found just to keep victim server busy [13].

Now for the research purpose authors use Back Track OS with IP address 202.4.96.199 to use it’s built in tool invite flood to flooding victim SIP server with fake invite authentication requests which IP address is 202.4.96.198. To do this need to follow above command format and the command is given below:

```
root@bt:~# ./inviteflood eth1 100 202.4.96.199 202.4.96.198 50000000000
```

The output of above command is given below in Fig. 3.

![Figure 3. Generating fake SIP invite authentication using Back Tack built in tool invite flood](image)

While the invite flood attack is running victim server with IP address 202.4.96.198 receives all the fake SIP invite authentication requests which captured using wireshark from the victim server Ethernet interface as like below Figure 4.

![Figure 4. Fake SIP invite flood request captured form the victim server Ethernet interface](image)

Above figure gives 41281 invite requests which were generated by invite flood tool within a few seconds which can severely damaged victim server performance as due to handle lots of fake request, victim server cannot negotiate with legitimate users. To detect the attacking host IP address form the victim SIP server administrator see the asterisk log files from the Linux kernel by using the below command:

```
[root@elastix ~]# tail -f /var/log/asterisk/full
```

In Fig. 5 shown that there are lots of fake request are coming from attacker.

![Figure 5. Asterisk log file form the victim SIP server](image)

From Fig. 5 administrator can determine the fake authentications are coming from the IP address 202.4.96.199 which actually the IP address of Back Track operating system with running DoS attack using invite flood tool form the VoIP portion. As victim SIP server receives lots of fake SIP invite authentication requests, sever service asterisk process increases during attacking period with high CPU utilization [14]. Due to this reason server performance severely damaged specially attacking period all the SIP user agents that all ready registered to the victim SIP server also get poor performance form the server as well as voice quality too. Some of them also get unregistered from the victim SIP server though they send original authentication to the SIP server that happens due to overload to the victim SIP server flooding SIP invite request. To monitor the server performance with all the processes that running in a server administrator can use the command given below:

```
[root@elastix ~]# top -H
```

The output of above command has shown in Fig. 6. Where during DoS attack period which increase the high CPU utilization percentage 98.3%

RTP provides a uniform packet format for transmitting end-to-end real-time data such as audio and video over IP network. Usually RTP uses user datagram protocol (UDP) an unreliable transport layer protocol as its transport protocol since data is delivered faster or real time. RTP flood attack is used to flood a target SIP user agent or SIP server with a UDP packet contains a RTP data in order to initiate a successful attack using RTP flood tool attacker need to know the RTP listening port on the remote device.
For example- For VoIP [15], SIP uses default RTP port which is 10000 to 20000 as given as below Fig. 7.

**Figure 6.** High CPU utilization percentage for service asterisk during invite flood DoS attack

**Figure 7.** Default RTP port for asterisk

During a successful RTP flooding attack, end-to-end SIP user agents cannot hear each other their voice sessions will be interrupted with noise sometimes victims sessions will be disconnected that actually depends on the strength of the flooding attack with observing correct listening port number. In Fig. 8 for research purpose authors use Back Tack operating system with IP address 202.4.96.197 which has a built in tool rtpflood for flooding victim SIP server with IP address 202.4.96.19. This tool generates fake RTP data packets within a very short period of time as a DoS attack. It exists in Back Track in the directory given below

```
root@bt:~# cd /pentest/voip/rtpflood/
```

This tool rtpflood use a format of command, the syntax of the command is given below

```
root@bt:~# ./rtpflood source_IP_Address destination_IP_address srcport destport numpackets seqno timestamp SSID
```

According to above command format the actual command given below

```
root@bt:~# ./rtpflood 202.4.96.199 202.4.96.198 21264 18102 500000000000000000000000000081
```

While the RTP flood attack is running victim server with IP address 202.4.96.198 receives the entire fake RTP data using UDP protocol [16] which was captured using wireshark from the victim server Ethernet interface as like Fig. 9.

**Figure 8.** RTP flooding using RTP flood tool

**Figure 9.** Fake RTP data packets using RTP flood tool form the victim server interface

```
root@bt:~# ./rtpflood 202.4.96.199 202.4.96.198 21264 18102 500000000000000000000000000081
```

While the attack is running, from the victim SIP server administrator’s login into the asterisk CLI mode to see RTP packets flow and also to detect the attacking host IP address which is 202.4.96.197 using the command given below

```
[root@elastix ~]# asterisk -r
Verbosity is at least 3
elastix*CLI> rtp set debug on
```

From asterisk CLI victim can detect IP address of RTP flood attacking host and also with the IP address of SIP user agent which listening port or RTP voice sessions will be affected by this attack as given below Fig. 10.

**Figure 10.** RTP debug at victim SIP server asterisk CLI
This DoS tool can generate huge amount of fake RTP [17] data packets within a few seconds using UDP as RTP uses it as a transport layer protocol for real time voice sessions. The entire packets that pass during RTP flood attack through the Ethernet interface of the victim SIP server 202.4.96.198 is captured by using wireshark as packet/ticks graph that gives huge amount of traffic which is enough for flooding a RTP voice session show in Fig. 11.

![Packet/Tick graph for Fake RTP data packets captured from victim SIP server interface](image)

**Figure 11.** Packet/Tick graph for Fake RTP data packets captured from victim SIP server interface

**V. PROPOSED MITIGATION TECHNIQUE**

In this paper, we approach some mechanism that can protect a given web server farm from application-level DoS attacks, especially, the attacks targeting the resources, including CPU, sockets, or memory, of the web server. The traffic rate of the source node including the SYN packet rate and the http request rate may not be effective any longer in discriminating the normal flows from the DoS attack flows, since the DoS attack can be effective, even with a low traffic rate after the emergence of low-rate attack tools. Instead, we focus on the symptoms at the server, rather than the attack traffic pattern itself. Since almost all the DoS attack tools intend to disable the server or degrade the performance of the server by offering excessive work to the server or holding the limited resource of the server, we attempt to detect the malicious node based on the amount of work given by each source node [18].

Fig. 12 shows the system architecture. Basically, the proposed system is able to drop malformed SIP messages and deter Invite/Cancel/Bye flooding. Any incoming SIP message must first pass the Malformed Message Detection module. Once a message is determined to be malformed, the proposed system will update the black-list in the SIP server to block further connections from that user.

**A. Approach 1: IPtables**

Conventional host based firewall using IPtables. In practice there may be thousands of nodes. Billions of packets can be directed at the victim, taking up all available bandwidth or causing DoS. The following Perl script has been developed to stop DoS attacks.

Enable socket reuse

```
[root@web ~]# echo 1 > /proc/sys/net/ipv4/tcp_tw_recycle
[root@web ~]# echo 1 > /proc/sys/net/ipv4/tcp_tw_reuse
```

Increase local port range

```
[root@web ~]# echo 1024 65535 > /proc/sys/net/ipv4/ip_local_port_range
```

As SIP server provides the graphical user interface, so administrator can allow a particular IP or IP address ranges for graphical usually admin login to the server by allowing or blocking the HTTPS protocol port number as given below [19]:

```
[root@l3ippx ~]# /sbin/iptables -A INPUT -p tcp -s 203.76.99.0/24 --dport 443 -j ACCEPT
[root@l3ippx ~]# /sbin/iptables -A INPUT -p tcp --dport 443 -j DROP
[root@l3ippx ~]# service iptables save
[root@l3ippx ~]# iptables -L
```

**B. Approach 2: Fail2ban**

Fail2ban operates by monitoring log files (e.g. /var/log/pwdfail, /var/log/auth.log, etc.) for selected entries and running scripts based on them. The working principle of fail2ban is given below Fig 13.

```
[root@13ipxx ~]# /sbin/iptables -A INPUT -p tcp --dport 443 -j DROP
```

Most commonly this is used to block selected IP addresses that may belong to hosts that are trying to break the system’s security. It can ban any host IP that makes too many login attempts or performs any other unwanted
action within a time frame defined by the administrator. Fail2ban can perform multiple actions whenever an abusive IP is detected: update iptables or PF firewall rules, TCP Wrapper’s hosts.deny table, to reject an abuser’s IP address; email notifications; or any user-defined action that can be carried out by a Python script. It works by scanning log files and then taking action based on the entries in those logs [20].

C. Configuring Fail2ban

“Enabled” defines whether or not a given section is enabled or not, it’s possible values are true or false. “filter” this is not used in the default section as it is used to tell fail2ban what it is looking for in the log file, its values could be among others likes apache-badbots, sshd, https, asterisk etc. Basically it is how the service is identified on the log file being parsed. “action” this option tells fail2ban what action to take once a rule is broken, could be specified a default action in the default section, and overwritten on each jail section may need to change the default value. “ignoreip” this option is used to set one or some IPs that should not be blocked, no matter how many times a users fail in login from those IPs. “maxretry” this option is used to set the limit of retries a user have before he gets blocked [20]. “bantime” this option is used to set the time (in seconds) an IP will be banned. It works by scanning log files and then taking action based on the entries in those logs. Fail2ban implemented with a configuration to be able to prevent SIP brute force attacks against our Asterisk PBXs. The following describes how to setup fail2ban to protect an Asterisk PBX from SIP brute force attempts and scans utilizing the iptables firewall. After install fail2band in SIP server, we need to create a configuration for Fail2Ban so that it can understand attacks against Asterisk. If having issues with system not banning properly when the "Registration from" section in your log file contains a quotation mark ("”) as in the Fig. 14.

Need to add the following line will block these attempts. Next edit /etc/fail2ban/fail.conf to include the following section so that it uses the new filter. It is also recommend to turn on an iptables ban for ssh, httpd/apache, and ftp if they are running on the system [21]. Edit vi /etc/fail2ban/fail.conf file and add this section. Configuration snap shown in Fig. 15.

Now we need to create the filter. Configuration snap shown in Fig. 16. The following filter configuration files are stored in /etc/fail2ban/filter.d/:

After using 3 times wrong authentication attacker IP address will automatically block in web server. The block IP address shown in Fig. 17.

Figure 16. Filter Configuration in Fail2ban

Figure 17. Fail2ban banned log from the server

After applying fail2ban rules scenery of RTP packet show in Fig.18. In Fig. 18 we can see that average RTP
packets are around 500packets/20 seconds. Which are normal RTP packets in voice network [22].

VI. CONCLUSION

In this paper, we investigated a new two-stage mechanism that can protect SIP from low rate resource-consuming DoS attacks. The proposed mechanism is based on two key ideas. The first one is an IPtables-based admission control scheme in the first stage, which protects the servers from a sudden surge of attack flows. We also investigated the condition to detect the victim servers and freeze the whitelist based on the server response time in detail. The second key idea is to detect attack flows based on the concept of a whitelist-based admission control defined for each pair of client and server IP addresses in the second stage. The experiment results show that the fail2ban scheme protects the SIP at the initial stage of DDoS attack, and the whitelist-based admission control policies attack flow detection mechanism distinguishes attack flows from normal flows and effectively filters the IP addresses of the attackers from the band list. Although we focused on protecting SIP in this paper, the proposed approach will be extended to other types of SIP attack in future study.

ACKNOWLEDGMENT

The authors would like to thanks Department of Electrical and Electronic Engineering, School of Engineering and Computer Science, Independent University, Bangladesh.

REFERENCES


Muhammad Morshed Alam received the B.Sc. degree in Electrical and Electronic Engineering from Independent University, Bangladesh, in 2013. He is doing M.Sc. degree in Electrical and Electronic Engineering from Islamic University of Technology (IUT). He worked as a System Engineer at Link3 Technology, Dhaka. His research interests include Computer networks, Network architecture, Network planning, design and Network security, communication protocols: SIP, H.323, Linux and UNIX, Open source VoIP system, Asterisk, VoIP development, integration, VoIP Security, Quality of Service and Universal radio
peripheral (USRP) Software. He has published over 4 papers in different peer reviewed international journals.

**Muhammad Yeasir Arafat** received the B.Sc. degree in Electronic and Telecommunications engineering from Independent University, Bangladesh, in 2012, the M.Sc. degree in Computer Network and Communications engineering from Independent University, Bangladesh, in 2014. He worked as a System Engineer at Dhakacom Limited. His research interests include Computer networks, Network architecture, Network planning, design and Network security, communication protocols: SIP, H.323, Linux and UNIX, Open source VoIP system, Asterisk, VoIP development, integration, VoIP Security, Quality of Service and Universal radio peripheral (USRP) Software. He has published over 5 papers in different peer reviewed international journals and 5 papers on national and international conferences. Mr. Yeasir is a member of the Institute of Electrical and Electronic Engineers (IEEE).

**Feroz Ahmed** received the B.Sc. degree in electrical and electronic engineering from Rajshahi University of Engineering and Technology, Bangladesh, in 1995, the M.Sc. degree in electrical engineering from University Technology Malaysia, Malaysia, in 1998, and the Ph.D. degree in communication engineering from the University of Electro-Communications, Tokyo, Japan in 2002. From 2002 to 2005, he worked as a postdoctoral research fellow at Gunma University, Gunma, and the University of Electro-Communication, Tokyo, Japan. In 2005, he was appointed as an Assistant Professor in the school of engineering and computer science at the Independent University, Bangladesh and became an Associate Professor in 2011. His research interests include optical fiber communications, wireless communications, and network security. He has published over 60 papers in different peer reviewed international journals and conferences. Dr. Ahmed is a member of the Institute of Electrical and Electronic Engineers (IEEE).