Proposed Methods of IP Spoofing Detection & Prevention

Sharmin Rashid[1], Subhra Prosun Paul[2]

[1, 2]World University of Bangladesh, Dhanmondi, Dhaka, Bangladesh

Abstract: ***In computer networking, the term IP address spoofing or IP spoofing refers to the creation of Internet Protocol (IP) packets with a forged source IP address, called spoofing, with the purpose of concealing the identity of the sender or impersonating another computing system. On January 22, 1995, in an article entitled, —New form of attack on computers linked to Internet is uncovered, John Markoff of the New York Times reported on the TCP/IP protocol suite's security weakness known as IP spoofing. The IP spoofing security weakness was published by S. M. Bellovin (1989). However, not much attention has been paid to the security weaknesses of the TCP/IP protocol by the general public. This is changing as more people and companies are connecting to the Internet to conduct business. This paper is on — "Proposed methods of IP Spoofing Detection & Prevention". This paper contains an overview of IP address and IP Spoofing and its background. It also shortly discusses various types of IP Spoofing, how they attack on communication system. This paper also describes some methods to detection and prevention methods of IP spoofing and also describes impacts on communication system by IP Spoofing. We think that our proposed methods will be very helpful to detect and stop IP spoofing and give a secured communication system.***

Keywords*:* IP address, IP Spoofing, TCP/IP, Compression, Cryptography

1. Introduction

The basic protocol for sending data over the Internet network and many other computer networks is the Internet Protocol ("IP"). The header of each IP packet contains, among other things, the numerical source and destination address of the packet. The source address is normally the address that the packet was sent from. By forging the header so it contains a different address, an attacker can make it appear that the packet was sent by a different machine. The machine that receives spoofed packets will send response back to the forged source address, which means that this technique is mainly used when the attacker does not care about the response or the attacker has some way of guessing the response.

Criminals have long employed the tactic of masking their true identity, from disguises to aliases to caller-id blocking. It should come as no surprise then, that criminals who conduct their nefarious activities on networks and computers should employ such techniques. IP spoofing is one of the most common forms of on-line camouflage. In IP spoofing, an attacker gains unauthorized access to a computer or a network by making it appear that a malicious message has come from a trusted machine by spoofing the IP address of that machine.

In certain cases, it might be possible for the attacker to see or redirect the response to his own machine. The most usual case is when the attacker is spoofing an address on the same LAN or WAN. Hence the attackers have an unauthorized access over computers. In this paper, we will examine the concepts of IP spoofing: why it is possible, how it works, how it can detect for and how to defend against it.

2. IP Spoofing

In computer networking, the term IP address spoofing or IP spoofing refers to the creation of Internet Protocol (IP) packets with a forged source IP address, called spoofing, with the purpose of concealing the identity of the sender or impersonating another computing system.

In a spoofing attack, the intruder sends messages to a computer indicating that the message has come from a trusted system. To be successful, the intruder must first determine the IP address of a trusted system, and then modify the packet headers to that it appears that the packets are coming from the trusted system. In essence, the attacker is fooling (spoofing) the distant computer into believing that they are a legitimate member of the network. The goal of the attack is to establish a connection that will allow the attacker to gain root access to the host, allowing the creation of a backdoor entry path into the target system.

Figure 1: Valid source IP address

Figure 1: Valid source IP address, illustrates a typical interaction between a workstation with a valid source IP address requesting web pages and the web server executing the requests. When the workstation requests a page from the web server the request contains both the workstation's IP address (i.e. source IP address 192.168.0.5) and the address of the web server executing the request (i.e. destination IP address 10.0.0.23). The web server returns the web page using the source IP address specified in the request as the destination IP address, 192.168.0.5 and its own IP address as the source IP address, 10.0.0.23.

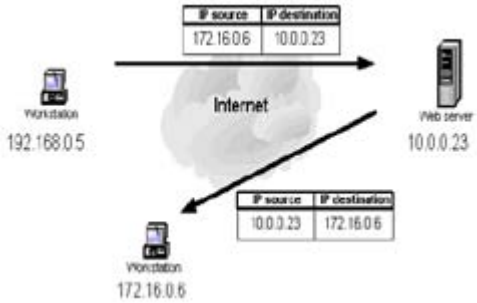

Figure 2: Spoofed source IP address

Figure 2: Spoofed source IP address, illustrates the interaction between a workstation requesting web pages using a spoofed source IP address and the web server executing the requests. If a spoofed source IP address (i.e. 172.16.0.6) is used by the workstation, the web server executing the web page request will attempt to execute the request by sending information to the IP address of what it believes to be the originating system (i.e. the workstation at 172.16.0.6). The system at the spoofed IP address will receive unsolicited connection attempts from the web server that it will simply discard.

3. Why IP Spoofing is Easy

- Problem with the Routers. IP routing is hop by hop. Every IP packet is routed separately. The route of an IP packet is decided by all the routers the packet goes through.
- Routers look at Destination addresses only.
- Authentication based on Source addresses only.
- To change source address field in IP header field is easy [1]

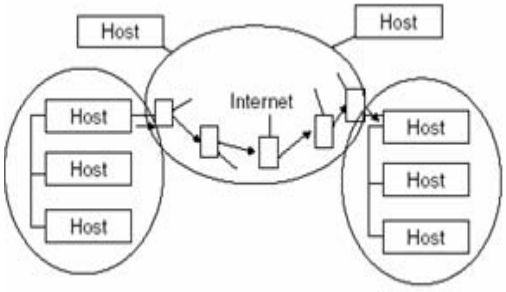

Figure 3: IP routing Mechanism

4. Spoofing Attacks

There are a few variations on the types of attacks that successfully employ IP spoofing. Although some are relatively dated, others are very pertinent to current security concerns.

4.1 Non-Blind Spoofing

This type of attack takes place when the attacker is on the same subnet as the victim. The sequence and acknowledgement numbers can be calculated, eliminating the potential difficulty of calculating them accurately. The biggest threat of spoofing in this instance would be session hijacking. This is accomplished by corrupting the DataStream of an established connection, then re-establishing it based on correct sequence and acknowledgement numbers with the attack machine. Using the spoofing, the attacker interferes with a connection that sends packets along the subnet.

4.2 Blind Spoofing

This attack may take place from outside where sequence and acknowledgement numbers are unreachable. Attackers usually send several packets to the target machine in order to sample sequence numbers, which is doable in older days.

Usually the attacker does not have access to the reply, abuse trust relationship between hosts. For example: Host C sends an IP datagram with the address of some other host (Host A) as the source address to Host B. Attacked host (B) replies to the legitimate host (A) shows in **Figure 4**.

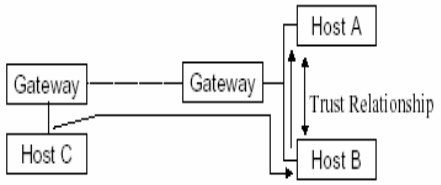

Figure 4: Blind Spoofing

4.3 Hijacking an Authorized Session

An attacker who can generate correct sequence numbers can send a reset message to one party in a session informing that party that the session has ended. After taking one of the parties' offline, the attacker can use the IP address of that party to connect to the party still online and perform a malicious act on it. The attacker can thus use a trusted communication link to exploit any system vulnerability. Keep in mind that the party that is still online will send the replies back to the legitimate host, which can send a reset to it indicating the invalid session, but by that time the attacker might have already performed the intended actions. Such actions can range from sniffing a packet to presenting a shell from the online host to the attacker's machine.

4.4 Man in the Middle Attack

Both types of spoofing are forms of a common security violation known as a man in the middle (MITM) attack. In these attacks, a malicious party intercepts a legitimate communication between two friendly parties. The malicious host then controls the flow of communication and can eliminate or alter the information sent by one of the original participants without the knowledge of either the original sender or the recipient. In this way, an attacker can fool a victim into disclosing confidential information by "spoofing" the identity of the original sender, who is presumably trusted by the recipient.

4.5 Denial of Service Attack

The connection setup phase in a TCP system consists of a three-way handshake. This handshake is done by using special bit combinations in the "flags" fields. If host A wants to establish a TCP connection with host B, it sends a packet with a SYN flag set. Host B replies with a packet that has SYN and ACK flags set in the TCP header. Host A sends back a packet with an ACK flag set, finishing the initial

handshake. Then hosts A and B can communicate with each other, as shown in Figure 5.

Figure 5: A Normal TCP Connection Request from A to B

The three-way handshake must be completed in order to establish a connection. Connections that have been initiated but not finished are called half-open connections. A finite-size data structure is used to store the state of the half-open connections. An attacking host can send an initial SYN packet with a spoofed IP address, and then the victim sends the SYN-ACK packet and waits for a final ACK to complete the handshake. If the spoofed address does not belong to a host, then this connection stays in the half-open state indefinitely, thus occupying the data structure. If there are enough half-open connections to fill the state data structure, then the host cannot accept further requests, thus denying service to the legitimate connections (Figure 6).

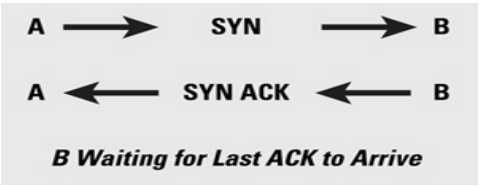

Figure 6: Half-Open TCP Connection

IP spoofing is almost always used in what is currently one of the most difficult attacks to defend against – denial of service attacks, or DoS. Since crackers are concerned only with consuming bandwidth and resources, they need not worry about properly completing handshakes and transactions. Rather, they wish to flood the victim with as many packets as possible in a short amount of time. In order to prolong the effectiveness of the attack, they spoof source IP addresses to make tracing and stopping the DoS as difficult as possible. When multiple compromised hosts are participating in the attack, all sending spoofed traffic; it is very challenging to quickly block traffic.

4.6 Attacks Concerning the Routing Protocols

A host can send spoofed RIP packets in order to "inject" routes into a host. This is easy to implement, it only requires IP/UDP spoofing. On a LAN with RIPv2 passwords have to be used for updating routes, but plaintext passwords are used. The plaintext passwords can be sniffed.

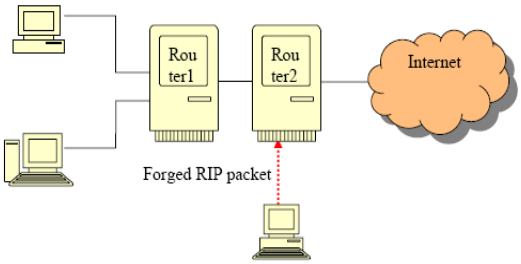

Figure 7: Link state before RIP attack

Attacker sends a forged RIP packet router 2 (Figure 7) and says it has the shortest path to the network that router1 connects. Then all the packets to that network will be routed to attacker (Figure 8). The attacker can sniff the traffic.

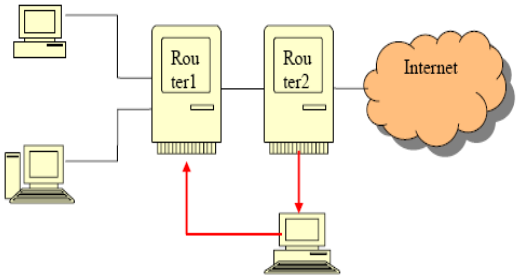

Figure 8: Link state after RIP attack

5. Spoofed Packet Detection

Packets sent using the IP protocol [2] include the IP address of the sending host. The recipient directs replies to the sender using this source address. However, the correctness of this address is not verified by the protocol. The IP protocol specifies no method for validating the authenticity of the packet's source. This implies that an attacker could forge the source address to be any he desires. This is a well-known problem and has been well described [3][4][5]. Detection methods can be classified as those requiring router support, active host-based methods, passive host based methods, and administrative methods. Administrative methods are the most commonly used methods today. When an attack is observed, security personnel at the attacked site contact the security personnel at the supposed attack site and ask for corroboration. This is extremely inefficient and generally fruitless. An automated method of determining whether packets are likely to have been spoofed is clearly needed. This section describes a number of such methods.

5.1 Routing Methods

Because routers (or IP level switches) can know which IP addresses originate with which network interface, it is possible for them to identify packets that should not have been received by a particular interface. For example, a border router or gateway will know whether addresses are internal to the network or external. If the router receives IP packets with external IP addresses on an internal interface, or it receives IP packets with an internal IP address on an external interface, the packet source is most likely spoofed. In the wake of recent denial-of-service attacks involving spoofed attack packets, ISPs and other network operators have been urged to filter packets using the above-described method. Filtering inbound packets, known as ingress filtering, protects the organization from outside attacks. Similarly, filtering outbound packets prevents internal computers from being involved in spoofing attacks. Such filtering is known as egress filtering. It is interesting to note that if all routers were configured to use ingress and/or egress filtering, attacks would be limited to those staged within an organization or require an attacker to subvert a router.

Internal routers with a strong notion of inside/outside can also detect spoofed packets. However, certain network topologies may contain redundant routes making this

distinction unclear. In these cases, host based methods can be used at the router. A number of IP addresses are reserved by the IANA for special purposes. These are listed in table 1. The addresses in the first group are private addresses and should not be routed beyond a local network. Seeing these on an outside interface may indicate spoofed packets. Depending on the particular site, seeing these on an internal address would also be suspicious. The other addresses in table 1 are special purpose, local only addresses and should never be seen on an outer interface.

Table #1: Special IP Addresses

Private Networks (RFC 1918) --	
10.0.0.0/8	
172.16.0.0/12	
192.168.0.0/16	
Special / IANA Reserved --	
0.0.0.0/8	- Historical Broadcast
127.0.0.0/8	- Loopback
169.254.0.0/16	- Link Local Networks
192.0.2.0/24	- TEST-NET
240.0.0.0/5	- Class E Reserved
248.0.0.0/5	- Unallocated
255.255.255.255/32	- Broadcast

Table 1: Special IP addresses

Many firewalls look for the packets described in this section. Typically they are dropped when received. Because firewalls have been a popular security product, research into routing methods has been active. Most all research has been in this area. Routers can also take a more active role in detecting spoofed packets. A number of advanced router projects have dealt with this and spoofed packet trace back [6]. We have proposed a number of proactive methods that can be used to detect and prevent spoofed packets.

One limitation of routing methods is that they are effective only when packets pass through them. An attacker on the same subnet as the target could still spoof packets. When the attacker is on the same Ethernet subnet as the target, both the source IP address and the Ethernet MAC would be spoofed. If the spoofed source address was an external address, the MAC would be that of the router. This implies that other techniques are required.

5.2. Non-routing methods

Computers receiving a packet can determine if the packet is spoofed by a number of active and passive ways. We use the term active to mean the host must perform some network action to verify that the packet was sent from the claimed source. Passive methods require no such action; however an active method may be used to validate cases where the passive method indicates the packet was spoofed. There are some other methods for detecting spoofed packets:

1. If we monitor packets using network-monitoring software such as netlog, look for a packet on our external interface that has both its source and destination IP addresses in your local domain. If we find one, you are currently under attack.
2. Another way to detect IP spoofing is to compare the process accounting logs between systems on our internal network. If the IP spoofing attack has succeeded on one of our systems, we may get a log entry on the victim machine showing a remote access; on the apparent source machine, there will be no corresponding entry for initiating that remote access.

There are some configuration and services that are vulnerable to IP spoofing:

- RPC (Remote Procedure Call services)
- Any service that uses IP address authentication
- The X Window System
- The R services suite (rlogin, rsh, etc.)

Some Softwares that are caused for IP Spoofing:

- Mac Spoofing
- Macaroni Screen Saver Bundle
- SpoofMAC
- sTerm
- MAC Change

6. IP Spoofing Prevention Methods

6.1 Compression

Basically compression classified into two types-

6.1.1 Lossy Compression

In Computer terminology, lossy compression is a data encryption method which eliminates some of the data, in order to achieve its goal, with the result that decompressing the data yields content that is different from the original, though similar enough to be useful in some way. Lossy compression is most commonly used to compress multimedia data, audio, video, image, etc. lossless compression is required for text and data files, such as bank records, text articles, etc. In many cases it is advantageous to make a master lossless file which can then be used to produce compressed files for different purposes.

We can compress many formats of digital data through that we can minimize the size of a computer file needed to store it. According to the networks the effective utilization of bandwidth needed to stream it, with no loss of the full information contained in the original file. A picture is converted to a digital file by considering it to be an array of dots, and specifying the color and brightness of each dot. If the picture contains an area of the same color, it can be compressed without loss by saying 200 red dots instead of red dot, red dot, etc red dot.

The original contains a certain amount of information; there is a lower limit to the size of file that can carry all the information. For example, most people know that WinRar produce the compressed ZIP file is smaller than the original file; but repeatedly compressing the file will not reduce the size to nothing, and will in fact usually increase the size.
Lossy compression formats suffer from generation loss: repeatedly compressing and decompressing the file will cause it to progressively lose quality. This is in contrast with lossless data compression. Information-theoretical foundations for lossy data compression are provided by rate distortion theory. Much like the use of probability in optimal coding theory, rate distortion theory heavily draws on

Bayesian estimation and decision theory in order to model perceptual distortion and even aesthetic judgment.

6.1.2 Lossless Compression

Lossless data compression is a kind of data compression algorithms that allows the exact original data to be fetched from the compressed ZIP data. The term lossless is in contrast to lossy data compression, which only allows an approximation of the original data to be re fetched, in exchange for better compression rates. Lossless data compression is used in many applications. For example, it is used in the popular ZIP file format and in the kernel OS UNIX tool gzip. It is also often used as a component within lossy data compression technologies.

Lossless compression algorithms and their implementations are routinely tested in head-to-head existing methods. There are a number of better-known compression existing methods. Some existing methods cover only the compression ratio, so winners in this benchmark may be unsuitable for everyday use due to the slow speed of the top performers. Another drawback of some existing methods is that their data files are known, so some program writers may optimize their programs for best performance on a particular data set. The winners on these existing methods often come from the class of context-mixing compression software.

6.2 Cryptography

Cryptography has been used as a way to send secret messages between warring nations, between users, between organizations etc; as such, it became an important issue in national security and laws. With the increasing need for secure transactions for data traversing computer networks for medical, financial, and other critical applications, cryptography is now becoming a necessity for nongovernmental, nonmilitary applications. All over the globe, the laws and regulations concerning cryptography are undergoing a vast change. Legal restrictions on the import and export of cryptographic products are being debated and modified.

Cryptography has some major issues:

- ***Key length*:** The combination of the algorithm and the key length are factors of cryptographic strength. The algorithm is usually well known. The longer key is the stronger the cryptographic strength of a given algorithm. Some countries have export laws that limit the key length of a given cryptographic algorithm.
- ***Key recovery***: In recent years, export laws have been modified if the cryptographic algorithm includes the capability of incorporating key recovery methods. These modified laws enable governments to wire-tap for encrypted electronic data if they deem it necessary to do so.
- ***Cryptography use***: A distinction is sometimes made about whether cryptography is used for authentication and integrity purposes or for confidentiality purposes. When used for confidentiality, the export laws are typically much more stringent.

In this chapter, cryptography uses to enhance the security in IP compression technique.

The main objective of IP compression is to avoid the overhead, which provides the bandwidth utilization. The IP header compression work initiated ten years ago but still there is some drawback and problem persists. For handling the packet transformation in effective manner we are moving to IPv6 but the header size will increase in IPv6. To increase the bandwidth utilizations, avoid the network traffic, congestion, collision, we go for compression technique. Basically compression used for minimize the size of file into half. For example if the original file size is 100mb after compression it will reduced into 50mb. While decompress your file we have to get original information without loose anything. Basic idea behind in this is remove the unwanted data's or information's.

In our work we incorporate the compression technique into TCP/IP packets. While data transfer two end systems will make the communication between these two end points the session will allocated for temporarily. Both systems has an unique IP address for identifying the system in network, using this IP address only communication will established. After establishing the end to end point connection the corresponding application will take charge to transactions. Application will identified using the port number. While continues data transfer some information will repeatedly send to the receiving end namely IP address of sender and receiver, port address of sender and receiver. To avoid this kind of information we go for compression technique. Most of the data compression algorithms have been developed and programmed in the traditional way. None of the previous algorithms has been evolved. The use of Evolutionary Computation has not been thoroughly investigated thus far. Researchers in the compression field tend to develop algorithms that work with specific types of data, taking the advantage of any available knowledge about the data. It is difficult to find a universal compression algorithm that performs well on any data type

6.3 Algorithm

- Split the packet header with data
- Applied the GRS compression algorithm
- Apply the cryptography technique
- Transmit the data
- Decryption
- Decompression
- Original information.

First take the original packet then split the packet header with the data. Whenever the data transmission happen that time 4tuple information are common for through out the data transfer. If we compress these things we can minimize the many space due to that we can utilize bandwidth in optimized manner.

The next step is applying the GRS algorithm which is the novel algorithm what we designed for our implementation. The concept behind in this is group of IP address considered as a single no which is taken as host identification no likewise we have to interchange into 4tuple's. For example 192.168.30.2 this is a one host IP address. This will

converted into like this. 2. We have to remember one thing after establishing the connection only the stream of packet will change into like this.

The next step is applying the cryptography technique. There are variety of techniques and complex methods available but in this scenario we couldn't use the complex technique because we going to apply in packet header. If we use complex technique, for encryption and decryption will take too much time. We have to use simple functions; in our implementation we used transformation function as method. It just modify the one value into another form using add or multiply that value into original no. for example the previous 2 will converted onto 6 adding 4 with 2 . The final thing is we have to send the key value for decryption. Key value will add into encrypted value for easy identification similar to the format of IP address 6.4 is the final value that will send to the destination machine likewise all 4tuple's. Again the decryption will happen in reverse manner.

We have also proposed some prevention methods to stop IP spoofing. They are:

1. The best method of preventing the IP spoofing problem is to install a filtering router (Figure 9) that restricts the input to our external interface (known as an input filter) by not allowing a packet through if it has a source address from our internal network. In addition, we should filter outgoing packets that have a source address different from our internal network in order to prevent a source IP spoofing attack originating from our site.

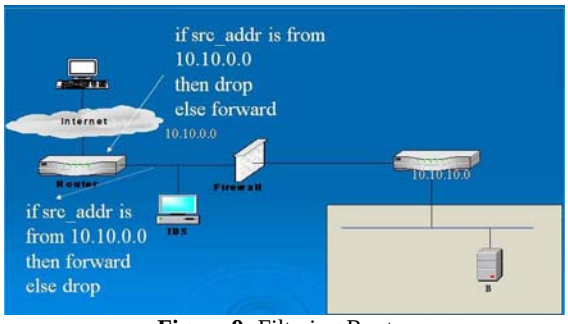

Figure 9: Filtering Router

2. Avoid using the source address authentication. Implement cryptographic authentication system-wide.
3. Configuring our network to reject packets from the Net that claim to originate from a local address.
4. If we allow outside connections from trusted hosts, enable encryption sessions.
5. Implement egress and ingress filtering. This will monitor all incoming and outgoing information and will block any unauthorized traffic.
6. Use encryption when sending any private information over the Internet. This will change any information you share into a code that hacker's will not be able to understand.
7. Use an ACL, or access control list, to block any unauthorized or private IP addresses.
8. Configure your router to reject unauthorized users that are claiming to be in your local network, when they are actually coming from outside your network.
9. Add an authenticated password-based key exchange to prevent IP spoofing. Two more users on the same network can use this key to access information. Without this, access is denied.

6.4 Software to Stop IP Spoofing

We can use some software's to stop IP Spoofing:

- StopCut
- Find Mac Address pro
- SecurityGateway for Exchange / SMTP
- PacketCreator
- Responder Pro

7. Conclusion

This paper describes the use of IP spoofing as a method of attacking a network in order to gain unauthorized access and some detection and prevention methods of IP spoofing. The goal of the attack is to establish a connection that will allow the attacker to gain root access to the host, allowing the creation of a backdoor entry path into the target system. We think that our proposed methods will be very helpful to detect and stop IP spoofing and give a secured communication system.

8. Acknowledgement

Special thanks to Md. Samsuzzaman, Assistant Professor, Department of Computer & Communication Engineering, Faculty of Computer science & Engineering, Patuakhali Science & Engineering for his helpful comments on IP Spoofing We also thank the anonymous reviewers for their helpful and constructive comments.

References

[1] Leila Fatmasari Rahman, Rui Zhou. ***IP Address Spoofing,*** (December 16, 1997). CERT Advisory CA-1997-28. IP Denial-of-Service Attacks. CERT/CC.
[2] Daemon9. IP Spoofing Demystified. Phrack Magazine Review, Vol 7, No. 48, June 1996, pp. 48-14.
[3] Computer Incident Advisory Committee (CIAC) (1995). Advisory Notice F-08 Internet Spoofing and Hijacked Session Attacks.
[4] Donkers, A. (1998, July). Are You really Who You Say You Are? System Administrator, Vol 7, No. 7, 69-71.
[5] D. Schnackenberg, K. Djahandari., and D. Sterne. Infrastructure for Intrusion Detection and Response. Proc. of the DARPA Information Survivability Conference and Exposition (DISCEX '00), 2000.
[6] S. Staniford-Chen and L. T. Heberlein. Holding Intruders Accountable on the Internet. Proc. of the 1995 IEEE, Symposium on Security and Privacy, , May 1995 Oakland, CA, pages 39-49.

Author Profile

Subhra Prosun Paul received the B.Sc.Engg. in Computer Science & Engineering from West Bengal University of Technology, Kolkata and M.S.in CSE from East West University, Dhaka in 2008 and 2011, respectively. He has researched on Linux, Networking, RDBMS and so on. He is now workind as Lecturer in department of Computer

Science & Engineering at World University of Bangladesh, Dhaka, Bangladesh.

Sharmin Rashid received the B.Sc. Engg. in Computer Science & Engineering a from Patuakhali Science & Technology University and M.S. degrees in Information Technology from Dhaka University in 2011 and 2013, respectively. She has researched on Networking, Semantic Web technology and so on. She is now working as Lecturer in department of Computer Science & Engineering at World University of Bangladesh, Dhaka, Bangladesh.