Vulnerabilities of Electronics Communication: solution mechanism through script

[1] Arun Kumar Singh
[1,3,4]Department of Computer Science and Engineering Motilal Nehru National Institute of Technology, Allahabad, Uttar Pradesh, 211004 India,

[2]Pooja Tewari
Computer Science and Engineering, I.M.S. Engineering College, Ghaziabad,

[3]Shefalika Ghosh Samaddar

[4]Arun K. Misra

Abstract

World trade and related business ventures are more or less dependent on communication. Information content of communication is to be protected as mis-communication or incorrect information may ruin any business prospect. Communication using Internet or any other electronic communication is having various kinds of threat and vulnerability. Information should be packaged for communication in such a way that these vulnerabilities are reduced to a minimum. With the increased use of networked computers for critical systems, network security is attracting increasing attention. This paper focuses on the most common attacks to paralyze computer and network resources, in order to stop essential communication services. The paper provides methods, ways and means for obtaining network traces of malicious traffic and strategies for providing countermeasures. Analysis of packet captured in a network traffic is a common method of deletion of countermeasure of communication based vulnerabilities. Analysis of http based network traffic allows to intercept sensitive information such as the user's name and password. The ideal approach for secured communication is to remove all security flaws from individual hosts. A tradeoff between overheads (computational and business) and efficiency of securing mechanism of communication may be achieved by using the script based solutions. This paper presents the communication based vulnerabilities and their script based solution.

Keywords: Computer Security, Network Security, Internet Security, Cryptography, Vulnerability, Firewalls, Attackers, Network Attacks

1. Introduction

With the advent of more and more open systems, intranets, and the Internet, information systems and need to assess and manage potential security risks on their network users are becoming increasingly aware of the networks and systems. Vulnerability assessment is the process of measuring and prioritizing these risks associated with network, host based systems and devices. A rational planning of technologies and activities will be able to manage business risk to a considerable extent. These tools allow customization of security measures, automated analysis of vulnerabilities, and creation of reports that effectively communicate security vulnerability. Detailed corrective actions to all levels of an organization may be automated.

The primary sources of information for vulnerable systems are network log data and system activity. Network-based systems look for specific patterns in a network traffic and host-based systems look for those patterns in log generated files. In general, network-based vulnerability can detect attacks that host-based systems can miss because they examine packet headers and the content of the payload, looking for commands or syntax used in specific attacks.

1.1 Vulnerability Assessment

Vulnerability assessment in a communication aims at identifying weaknesses and vulnerabilities in a system's design, implementation, or operation and management, which could be exploited to violate the system's security. The overall scope of vulnerability assessment is to improve information and system security by assessing the risks associated. Vulnerability assessment will set the guidelines to stop or mitigate any risk.

This paper focuses on a technical vulnerability assessment methodology, giving an exposure of the threats and vulnerabilities. Major Internet-based security issues and network threats are covered. Threats and their management requires performing assessment exercise.

1.1.1 Host Based Vulnerability Assessment

[1,3,4]The first, second and third Authors are thankful to Information Security Education & Awareness Project (ISEA) of MCIT department of Information Technology, Govt. of India for the partial support to the research conducted.

Vulnerability Assessment is to identify what systems are "alive" within the network ranges for host based threats and what services they offer. Identifying the location of the establishment and cataloging its services are the two main elements of Vulnerability assessment Assessment of vulnerability may lead to the deletion of a number of viruses, worms and Trojan horses.

A virus is a package of code that attaches itself to a host program and propagates when the infected program is executed in a indirect mode along with some other essential programs. Attracting a virus to system programs or commands is an easy way of propagating of the viruses. Thus, a virus is self-replicating and self-executing. Viruses are transmitted when included as part of files downloaded from the Internet or as e-mail attachments Worms are independent programs that replicate by copying themselves from one system to another, usually over a network or through e-mail attachments. Many modern worms also contain virus code that can damage data or consume system resources that they render the operating system unusable.

A Trojan horse program (also known as a "back door" program) acts as a stealth server that allows intruders to take control of a remote computer without the owner's knowledge. Greek mythical Trojan horses are analogous in attributes which these digital Trojan horses posses. These programs typically masquerade as benign programs and rely on gullible users to install them. Computers that have been taken over by a Trojan horse program are sometimes referred to as zombies. Armies of these zombies can be used to launch crippling attacks against Web sites.

Communication based vulnerability are a real time threats to computer's security. Those may take the form of physical attacks, pilfered passwords, nosy network neighbors and viruses, worms, and other hostile programs. A number of manifestations of such vulnerability are seen these days e.g. Denial of service (DoS) attacks.

A denial-of-service (DoS) attack hogs or overwhelms a system's resources so that it cannot respond to service requests. A DoS attack can be effected by flooding a server with so many simultaneous connection requests that it cannot respond. Another approach would be to transfer huge files to a system's hard drive, exhausting all its storage space. A related attack is the distributed denial-of-service (DDoS) [1].

The Security Threat and the Response attack, is also an attack on a network's resources. It is launched from a large number of other host machines. Attack software is installed on these host computers, unbeknownst to their owners, and then activated simultaneously to launch communications to the target machine of a magnitude as to overwhelm the target machine.

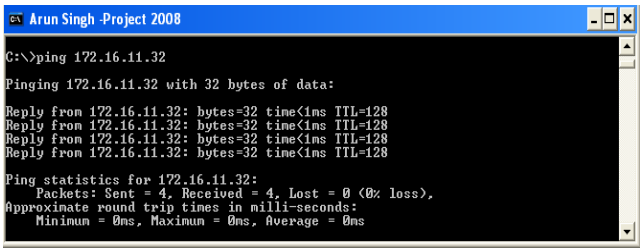

Figure –1 Ping command to check system is alive or not

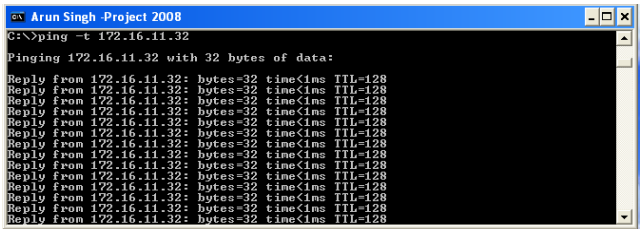

Figure –2 DoS Attack

Ping of Death is another flavour (Figure-1, Figure-2) of DDoS. Smurf Attack involves using IP spoofing and the ICMP to saturate a target network with traffic. It is then equivalent to launching a DoS attack. It consists of three elements: the source site, the bounce site, and the target site. The attacker (the source site) sends a spoofed ping packet to the broadcast address of a large network (the bounce site). This packet modified by the intruder contains the address of the target site. This causes the bounce site to broadcast the misinformation to all of the devices on its local network. All of these devices now respond with a reply to the target system, which is then saturated with those replies.

Spam is another malicious formulation in the arena of cyber crime. Responses to spam may lead to huge financial and material loss. Spam has the format of a e-mail message that are pushed to e-mail clients without their solicitation.

2.0 Related Work

Vulnerability assessment process is comprised of four phases, namely discovery, detection, exploitation, and analysis/recommendations [2]. Figure 3 identifies the relationships among the four phases, and the flow of information into the final report.

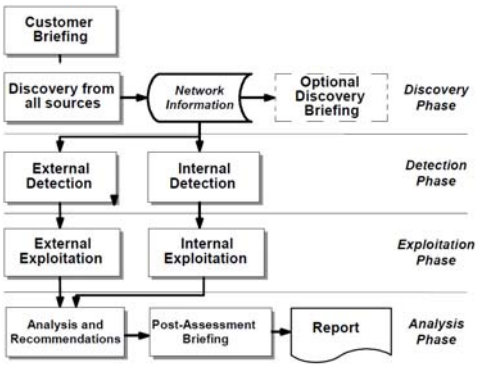

Figure-3 Vulnerability Assessment Process
[source: http://www.oissg.org/wiki/images/4/4b/Image001.png]

Protocol based attack/Packet based attack has been studied from the very beginning of the study of security and related vulnerabilities. With rapid growth in both the number and sophistication of cyber attacks, it has become imperative that cyber defenders be equipped with highly effective tools that identify security Vulnerabilities before they are exploited [3]. Vulnerability can be defined as a set of conditions which if true, can leave a system open for intrusion, unauthorized access, denied availability of services running on the system or in any way violate the security policies of the system set earlier.

A breach of security occurs when a stated organizational policy or legal requirement regarding information security, has been contravened. However, every incident which suggests that the confidentiality, integrity and availability of the information has been inappropriately changed, can be considered a security vulnerability. Every security breach is always initiated via security vulnerability, only if confirmed does it become a security breach [4].

A denial of service (DoS) attack is a malicious attempt by one or many users to limit or completely disable the availability of a service. They cost businesses millions of pounds each year and are a serious threat to any system or network. These costs are related to system downtime, lost revenues, and the labour involved in identifying and reacting to such attacks [5]. DoS attacks were theorized years ago, before the mass adoption of current Internet protocols [6].

DoS is still a major problem today and the Internet remains a fragile place [6]. A large number of known vulnerabilities in network software and protocols exist; relating DoS. Sending enough data to consume all available network bandwidth (Bandwidth Consumption) is a DoS attack. Sending data in such a way as to consume a resource needed by the service (Resource Starvation) is another DoS attack. Exercising a software.bug. causing the software running the service to fail (Programming Flaws) is the other type of the attack. Malicious use of the Domain Name Service (DNS) and Internet routing protocols leads to DoS. Many DoS attacks exploit inherent weaknesses in core Internet protocols. This makes them practically impossible to prevent, since the protocols are embedded in the underlying network technology and adopted as standards worldwide. Today, even the best countermeasure software can only provide a limiting effect on the severity of an attack [7]. An ideal solution to DoS will require changes in the security and authentication of these protocols [6].

In order to launch some DoS attacks, the programmer must be able to form raw packets. Using raw packets, the header information and data can be manipulated to form any kind of packet sequence. Hence techniques such as IP Spoofing and malformed ICMP Ping requests can be used [18]. This report will investigate the mechanism of DoS attacks and their countermeasures. Distributed denial of service attacks will also be investigated. A distributed DoS generally has the same effect as a single attack, with the disruption amplified by many systems acting together. These other systems are often compromised machines remotely controlled by the hacker [8].

With the rapid development of more complex systems, the chance of introduction of errors, faults and failures increases in many stages of software development life-cycle [9]. This class of system failures is commonly termed as software vulnerabilities. These security vulnerabilities violate security policies and can cause the system to be compromised leading to loss of information . Vulnerabilities can be introduced in a host system in different ways; via errors in the code of installed software, mis-configurations of the software settings that leave systems less secure than they should be (improperly secured accounts, running of necessary services, etc)

Network based vulnerability assessment gathers information of the system and services attached to the network and identities weakness and vulnerabilities exploitable in the network. These vulnerabilities could be related to services, such as HTTP, FTP and SMTP protocol, running on the given network. A network-based scanning assessment may also detect extremely critical vulnerabilities such as mis-configured firewalls or vulnerable web servers in a De-Militarized Zone (DMZ), which could provide a security hole to an intruder, allowing them to compromise an organizations security [10]. Network assessment tools gather information and may also have network mapping and port scanning abilities [2]. The tools use for such purpose are Nmap etc. [2].

3.0 Design of the solution

Host-based vulnerability analysis has been taken up for design of solution along with a lot of potential for further research and development in many other fields including the field of vulnerability analysis. Plugging of the vulnerability is ensured by designing script based and command based codes sniffing a HTTP packet is shown in figure 4. Capturing a HTTP based e-mail password is shown is figure 5.

Sniffing HTTP packet and its result in figure 4 are roles worthy .Capturing a HTTP based mail Password in figure 5 is equally important from the point of view of vulnearibuilities. The packet list pane shows that the HTTP protocol packets are being transmitted from source IP 172.31.132.59 to destination IP 172.31.100.29. The packets are being captured while transmitting from one mode to other. This particular packet gives the information that HTTP mail of this website *http:// mail.mnnit.ac.in* has been logged in by the source IP and its corresponding username and password are also captured under the heading of line-based text data in packet detail pane (figure-5).

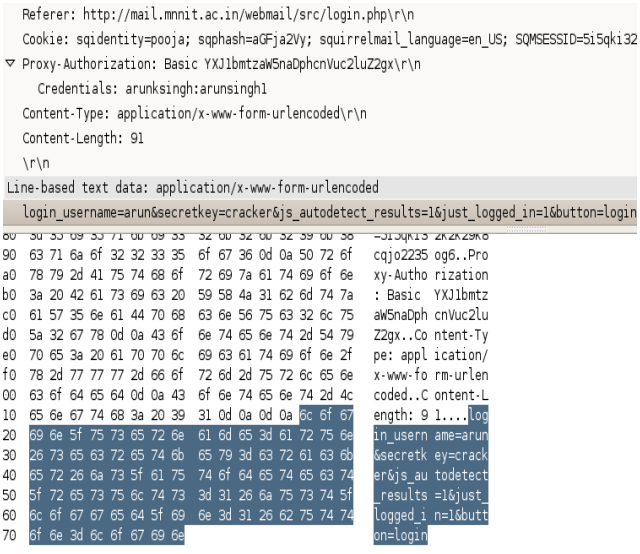

Figure-4 Capturing a HTTP based mail Password

Figure 6 shows that the username is *arun* and password is *cracker* which is given next to secret key. This is also shown in packet bytes pane in the right hand side of HEX numbers (Figure 4). Sometimes, when the password of a user contains some special characters, they are written using special character that appears in the pane.

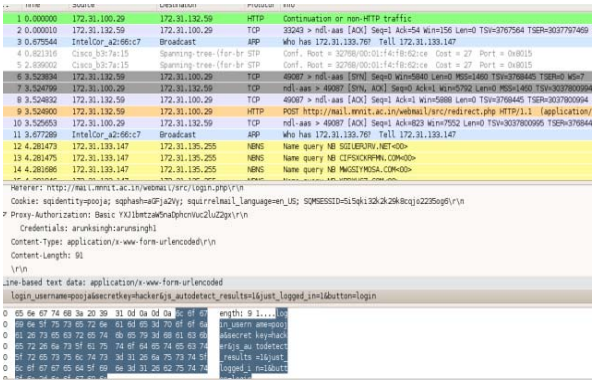

Figure-5 Proxy Authorization

There are a number of tools available for such purpose. Wireshark is able enables to sniff the proxy password as illustrated in figure-5. This is done in the same way as capturing of username and password of a mailuser as shown in figure 6. Proxy password is also obtained in packets detail pane under the Proxy- authorisaton. In this figure, proxy username is *'arunksingh* 'and password is *'arunsingh1'* which is shown next to Credentials. This is how sniffing is being done over HTTP connection in LAN.

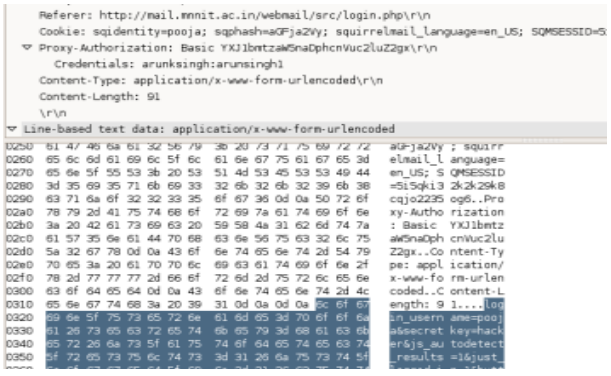

Figure-6 Capturing the Content of Message sites

Wireshark is able to capture the username and password of mail user in the same way it does for message websites like www.160by2.com or www.way2sms.com. Figure-6 shows the capturing of a message packet being sent from the message website www.160by2.com as shown in figure-6. This figure shows that the user whose IP address

is 172.31.132.59 when logins the message website, the packet is sent to the destination IP address 172.31.100.29 which capture the HTTP packet and the corresponding information to this is given in info 'POST' as http://www.160by2.com/logincheck.

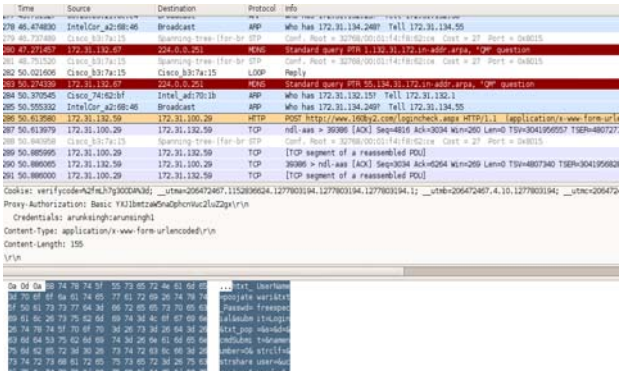

Figure-7 Massage Captured by Wireshark

Sending a secret message to anyone by these types of message sites has it own liabilities because Wireshark can easily capture his message. Example of this sent message is as shown in figure-8. This captured packet is analyzed by TCP stream.

```
POST http://www.160by2.com/postview_latest.aspx HTTP/1.1
Host: www.160by2.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.5) Gecko/2008121621 Ubun
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Proxy-Connection: keep-alive
Referer: http://www.160by2.com/compose_sms.aspx
Cookie: __utma=206472467.1152836624.1277803194.1277874178.1277881429.4; __utmz=206472
verifycode=5V2XF4JOBrw%3d; __utmc=206472467; __utmb=206472467.3.10.1277881429; id=nlr
2bZ2wAflmS9ehGNNv488rzdVOQhNvrRI9Oh0qKQxA%3d%3d
Proxy-Authorization: Basic YXJ1bmtzaW5naDphcnVuc2luZ2gx
Content-Type: application/x-www-form-urlencoded
Content-Length: 101

useremails=saumya+%3C9453642251%3E&txt_msg=hi+dear+hw+r+u%3F&hf_msg=&islang=&act_mnos
Date: Wed, 30 Jun 2010 06:59:38 GMT
```

Figure-8 Content of the message

It shows the content of a message is seen clearly and also the contact number of the person to whom it has been sent. The message content written next to the text message heading is *hi+ dear+ hw+ r+ u.* This is the original message content as *hi dear hw r u* was being sent from this website.

TShark is a network protocol analyzer and a command-line version of Wireshark, which captures the live packet data from a live network, or read packets from a previously saved capture file. By default, tshark prints the summary line information to the screen. This is the same information contained in the top pane of the Wireshark GUI. The default tshark output is shown below in figure-9.

```
0.367032 172.31.201.155 -> 172.31.255.251 ICMP Echo (ping) request
0.367073 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.155 is at 00:23:5a:47:cc:25
0.377262 172.31.201.153 -> 172.31.255.251 ICMP Echo (ping) request
0.377320 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.153 is at 00:23:5a:47:cc:25
0.387435 172.31.201.152 -> 172.31.255.251 ICMP Echo (ping) request
0.387477 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.152 is at 00:23:5a:47:cc:25
0.397579 172.31.201.151 -> 172.31.255.251 ICMP Echo (ping) request
0.397623 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.151 is at 00:23:5a:47:cc:25
0.407801 172.31.201.150 -> 172.31.255.251 ICMP Echo (ping) request
0.407876 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.150 is at 00:23:5a:47:cc:25
0.417976 172.31.201.149 -> 172.31.255.251 ICMP Echo (ping) request
0.418020 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.149 is at 00:23:5a:47:cc:25
0.428226 172.31.201.148 -> 172.31.255.251 ICMP Echo (ping) request
0.428266 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.148 is at 00:23:5a:47:cc:25
0.438376 172.31.201.147 -> 172.31.255.251 ICMP Echo (ping) request
0.438426 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.147 is at 00:23:5a:47:cc:25
0.448547 172.31.201.145 -> 172.31.255.251 ICMP Echo (ping) request
0.448599 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.145 is at 00:23:5a:47:cc:25
0.458705 172.31.201.143 -> 172.31.255.251 ICMP Echo (ping) request
0.458760 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.143 is at 00:23:5a:47:cc:25
0.468941 172.31.201.141 -> 172.31.255.251 ICMP Echo (ping) request
0.468982 00:23:5a:47:cc:25 -> Cisco_74:62:bf ARP 172.31.201.141 is at 00:23:5a:47:cc:25
```

Figure-9 Capturing Password by Tshark

This paper is focused on the data communication over the HTTP connections in LAN, which are not secure and important information maybe sniffed in the form of packet when passing through multiple stations to a destined one. In figure-9, it is illustrated that when the user logins the message website, then his password can be sniffed as shown in the right hand side of the column in the last 9th line. The username is *'poojatewari'* and password is *'passhacked'* when the user logins the message website (figure 10).

```
0370  55 6c 67 56 52 72 4b 53 6f 32 41 42 54 4d 69 44   UlgVRrKSo2ABTMiD
0380  31 78 74 44 74 68 35 74 46 50 56 4b 75 63 67 25   1xtDth5tFPVKucg%
0390  33 64 25 33 64 0d 0a 50 72 6f 78 79 2d 41 75 74   3d%3d..Proxy-Aut
03a0  68 6f 72 69 7a 61 74 69 6f 6e 3a 20 42 61 73 69   horization: Basi
03b0  63 20 59 58 4a 31 62 6d 74 7a 61 57 35 6e 61 44   c YXJ1bmtzaW5naD
03c0  70 68 63 6e 56 75 63 32 6c 75 5a 32 67 78 0d 0a   phcnVuc2luZ2gx..
03d0  43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 61 70   Content-Type: ap
03e0  70 6c 69 63 61 74 69 6f 6e 2f 78 2d 77 77 77 2d   plication/x-www-
03f0  66 6f 72 6d 2d 75 72 6c 65 6e 63 6f 64 65 64 0d   form-urlencoded.
0400  0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a   .Content-Length:
0410  20 31 35 34 0d 0a 0d 0a 68 74 78 74 5f 55 73 65    154....htxt_Use
0420  72 4e 61 6d 65 3d 70 6f 6f 6a 61 74 65 77 61 72   rName=poojatewar
0430  69 26 74 78 74 5f 50 61 73 73 77 64 3d 70 61 73   i&txt_Passwd=pas
0440  73 68 61 63 6b 65 64 26 73 75 62 6d 69 74 3d 4c   shacked&submit=L
0450  6f 67 69 6e 26 74 78 74 5f 70 6f 70 3d 26 73 3d   ogin&txt_pop=&s=
0460  26 64 3d 26 63 6d 64 53 75 62 6d 69 74 3d 26 6e   &d=&cmdSubmit=&n
0470  61 6d 65 6e 75 6d 62 65 72 3d 30 26 73 74 72 63   amenumber=0&strc
0480  6c 66 3d 26 73 74 72 73 68 61 72 65 75 73 65 72   lf=&strshareuser
```

Figure-10 Capturing the data of a Message Website

Tshark can also capture the sent message from a message website like www.160by2.com or www.way2sms.com. Capturing of the sent message from www.160by2.com is illustrated in figure 11.

When the source IP address 172.31.132.59 sents a packet containing the data content to the destination IP address 172.31.100.14, it can be sniffed as shown in the figure. It

shows the contact number and the message sent to that contact. Here, the captured content is *hello++ hwz+ u* next to text message heading as in Wireshark. This original message content sent is *“ hello hwz u”* sent from this website.

```
30 6a 44 6c 4e 31 39 32 77 45 72 78 4d 65 4f 4f   OjDlN192wErxMeOO
6f 68 31 47 51 74 35 61 56 42 47 5a 64 54 25 32   oh1GQt5aVBGZdT%2
62 77 25 33 64 25 33 64 0d 0a 50 72 6f 78 79 2d   bw%3d%3d..Proxy-
41 75 74 68 6f 72 69 7a 61 74 69 6f 6e 3a 20 42   Authorization: B
61 73 69 63 20 59 58 4a 31 62 6d 74 7a 61 57 35   asic YXJ1bmtzaW5
6e 61 44 70 68 63 6e 56 75 63 32 6c 75 5a 32 67   naDphcnVuc2luZ2g
78 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a   x..Content-Type:
20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 2d 77    application/x-w
77 77 2d 66 6f 72 6d 2d 75 72 6c 65 6e 63 6f 64   ww-form-urlencod
65 64 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67   ed..Content-Leng
74 68 3a 20 31 31 32 0d 0a 0d 0a 75 73 65 72 65   th: 112....usere
6d 61 69 6c 73 3d 76 69 73 68 75 2b 25 33 43 39   mails=vishu+%3C9
34 35 33 31 39 32 32 36 37 25 33 45 26 74 78 74   453192267%3E&txt
5f 6d 73 67 3d 68 65 6c 6c 6f 2e 2e 2e 2e 2e 2e   _msg=hello......
2e 2b 2b 68 77 7a 2b 75 2b 25 33 46 25 33 46 25   .++hwz+u+%3F%3F%
33 46 26 68 66 5f 6d 73 67 3d 26 69 73 6c 61 6e   3F&hf_msg=&islan
67 3d 26 61 63 74 5f 6d 6e 6f 73 3d 39 31 39 34   g=&act_mnos=9194
35 33 31 39 32 32 36 37 25 32 43                  53192267%2C

869674 172.31.132.59 -> 172.31.100.14 HTTP GET http://www.160by2.com/css/innerpage
```

Figure-11 Captured content of the message

Before doing arpspoofing, IP forwarding is enabled so that all the traffic passes through the attacker’s system. The attacker deter-mines whether the IP forwarding is enabled in the system or not by the command ‘cat /proc/sys/net/ipv4/ip forward’ If the IP forwarding is disabled in the system then the output is 0 else the output is 1. When the system has its IP forwarding disabled then it is enabled by the following command as given in figure-12.

echo 1 > /proc/sys/net/ipv4/ip forward

```
ip_dynaddr          ip_local_port_range  ip_no_pmtu_disc
root@pooja-laptop:/home/pooja# cat /proc/sys/net/ipv4/ip_forward
1
root@pooja-laptop:/home/pooja#
```

Figure-12 IP forwarding

The communication between the host and a gateway is achieved in a defined manner Computer A whose IP address is 172.31.132.42 and MAC address is 00:24:be:b5:a6:73 wants to communicate with gateway whose IP address is 172.31.132.1 and MAC ad-dress is 00:1b:d4:74:62:bf to access Internet. Computer A sends out ARP request to gateway requesting MAC address. Switch receives request (which is broadcasted) and passes this request along to every connected computer. Switch also up-dates its internal MAC address to port table. Gateway receives ARP request from Computer A, and replies with MAC address. Gateway updates internal ARP table with MAC address and IP address of Computer A. Switch receives ARP reply to Computer A, checks its table, and finds Computer A’s MAC address listed at port 1. It passes this information to port 1 and then updates MAC table with MAC address from gateway. Computer A receives ARP information from gateway, and it updates it ARP table with this information. Computer A sends information out to gateway using updated MAC address information, and communication channel is established. ARP spoofing is now done after the IP forwarding is enabled to sniff all the packets going between a host IP 172.31.132.49 and gateway IP 172.31.132.1, which is being sent to the internet as illustrated in figure 13.

arpspoof -t 172.31.132.42 172.31.132.1 & > /dev/null

Figure 13 illustrates that all the packets that were destined to 172.31.132.1 are rerouted to the system running this command. The system whose IP address is 172.31.132.42 and MAC address 0:24:be:b5:a6:73 is being spoofed by the attacker’s system whose IP address is 172.31.132.59 and MAC address is :23:5a:47:cc:21. The system running ARP spoof whose MAC address is 0:23:5a:47:cc:21 broadcasts the ARP reply that it has the IP address 172.31.132.42. The victim’s MAC address is spoofed by the attacker’s MAC address.

```
0:23:5a:47:cc:21 ff:ff:ff:ff:ff:ff 0806 42: arp reply 172.31.132.42 is-at 0:23:5a:47:cc:21
0:23:5a:47:cc:21 ff:ff:ff:ff:ff:ff 0806 42: arp reply 172.31.132.42 is-at 0:23:5a:47:cc:21
0:23:5a:47:cc:21 ff:ff:ff:ff:ff:ff 0806 42: arp reply 172.31.132.42 is-at 0:23:5a:47:cc:21
0:23:5a:47:cc:21 ff:ff:ff:ff:ff:ff 0806 42: arp reply 172.31.132.42 is-at 0:23:5a:47:cc:21
0:23:5a:47:cc:21 ff:ff:ff:ff:ff:ff 0806 42: arp reply 172.31.132.42 is-at 0:23:5a:47:cc:21
0:23:5a:47:cc:21 ff:ff:ff:ff:ff:ff 0806 42: arp reply 172.31.132.42 is-at 0:23:5a:47:cc:21
0:23:5a:47:cc:21 ff:ff:ff:ff:ff:ff 0806 42: arp reply 172.31.132.42 is-at 0:23:5a:47:cc:21
0:23:5a:47:cc:21 ff:ff:ff:ff:ff:ff 0806 42: arp reply 172.31.132.42 is-at 0:23:5a:47:cc:21
0:23:5a:47:cc:21 ff:ff:ff:ff:ff:ff 0806 42: arp reply 172.31.132.42 is-at 0:23:5a:47:cc:21
```

Figur-13- ARP request

3.1 Capturing of WebPages Visited

Dsniff is a tool that extracts information about the webpages visited by the victim. Let us consider the following case study as conducted is the Information Security Laboratory. The victim’s MAC address 00:24:be:b5:a6:73 has been spoofed by the attacker’s MAC address 0:23:5a;47:cc:21. In figure-14, victim’s IP 172.31.132.42 has been spoofed and IP for- warding has already been enabled to get the whole traffic between the victim and the gateway IP. It shows all the webpages

which has been visited by victim in the system who is running the dsniff tool. Here, the system whose IP address is 172.31.132.59 and MAC address is 0:23:5a:47:cc:21 dsniffs all the webpages visited by the victim's system. Hacker-Arun first connects to the web site *www.google.com* on date 07-07-10 at the time 15:13:05 and then to the mail.mnnit.ac.in after 2 minutes 15:15:53 on the same day.

```
----------------
07/07/10 15:13:05 tcp Hacker-Arun.local.43924 -> 172.31.100.14.3128 (http)
CONNECT www.google.com:443 HTTP/1.1
Host: www.google.com
Proxy-Authorization: Basic YXJ1bmtzaW5naDphcnVuc2luZ2gx [arunksingh:arunsingh1]

----------------
07/07/10 15:15:53 tcp Hacker-Arun.local.60382 -> 172.31.100.14.3128 (http)
GET http://mail.mnnit.ac.in/webmail/images/draft.png HTTP/1.1
Host: mail.mnnit.ac.in
Proxy-Authorization: Basic YXJ1bmtzaW5naDphcnVuc2luZ2gx [arunksingh:arunsingh1]

GET http://mail.mnnit.ac.in/webmail/images/senti.png HTTP/1.1
Host: mail.mnnit.ac.in
Proxy-Authorization: Basic YXJ1bmtzaW5naDphcnVuc2luZ2gx [arunksingh:arunsingh1]
```

Figure-14 Capturing of Webpage Visited

3.2 Denial Of Services

In a denial-of-service (DoS) attack, an attacker attempts to pre-vent legitimate users from accessing information or services. It is an action or set of actions that prevent any part of a system from functioning as it should. This includes the actions that causes unauthorized destruction, modification, or delay of service. DoS results in the loss of a service in a particular network or temporary loss of services in all the network services. It does not usually used to sniff the data and information passing through the network traffic over the HTTP connection in LAN. By targeting victim's computer and its network connection, an attacker may be able to prevent him from accessing email, websites, online accounts (banking, etc.) or other services that rely on the affected computer. When a person connects to a website into the browser, he is sending a request to that site's computer server to view the page. There is a limit to the number of the requests which can be accessed at a given time. So, the attacker overloads the server with requests, which in turn can not process the victim's request.

DOS includes sending oversized ICMP echo packets which increases the payload and results in Denial of Services for the client.

4.0 Countermeasures for Network Attacks

Static ARP table is a one way to prevent the ARPspoofing. The ARP table is generated using the command arp -s IPaddress MAC address This will add static entries to the table i.e. unchanging entries which prevents attacker from adding spoofed ARP entries as illustrated in figure-17. This detects if a new Ethernet device is added to an existing network, but it has no method of predefining an acceptable IP address. In this figure, a static entry to the ARP table is added by arp -s 172.31.152.45 00:1B:D4:74:62:BF.

4.1 Static ARP Table

The table will record this IP address and MAC address. As a result no ARP spoofing can be done. Whenever there is any data communication in between the hosts over the HTTP connection in LAN, it will check whether the table has the particular IP address or not before broadcasting the ARP request to each hosts on the network. So, no ARP broadcasts request is sent which prevents the ARP spoofing. ARP table shows IP address, MAC address, interface and flag

```
root@pooja-laptop:/home/pooja# arp -e
Address                  HWtype  HWaddress           Flags Mask            Iface
172.31.100.14            ether   00:1B:D4:74:62:BF   CM                    eth0
root@pooja-laptop:/home/pooja# arp -s 172.31.152.45 00:1B:D4:74:62:BF
root@pooja-laptop:/home/pooja#
root@pooja-laptop:/home/pooja# arp -e
Address                  HWtype  HWaddress           Flags Mask            Iface
172.31.152.45            ether   00:1B:D4:74:62:BF   CM                    eth0
172.31.100.14            ether   00:1B:D4:74:62:BF   CM                    eth0
```

Figure-15- Adding Static entry to the ARP table

Mask in figure 15. If any static entry is added to the ARP table, then the corresponding IP/MAC address is marked and remains unchanged until the system shuts down.

4.2 ARPwatch

ARPwatch is a program which works by monitoring an interface in promiscuous mode and recording MAC and IP address pairings over a period of time. When it sees anomalous behavior in case of change to one of the MAC and IP address pairs that it has received, it will send an alert in the form of a warning to the user. ARPwatch runs by selecting one of the inter- face from multiple interfaces on the command line. It runs and records the IP and MAC address by arpwatch -d and gives the information about hostname, host IP address, interface, Ethernet address and time when it is recorded as illustrated in figure-16. The system running the ARPwatch gets the details of MAC and the corresponding IP addresses. In the presented simulation, the system pooja-laptop is running the ARPwatch and gets the information about the unknown host name whose IP address is 172.31.134.126, interface is eth0 and has an ethernet address 0:13:20:b1:3d:8. It again records that the host name *'Hacker-Arun'* whose IP address is 172.31.132.42 , interface is eth0 and has its corresponding MAC address 0:24:be:b5:a6:73. A file

arp.dat is created so as to record the MAC/IP address of the system in that network.

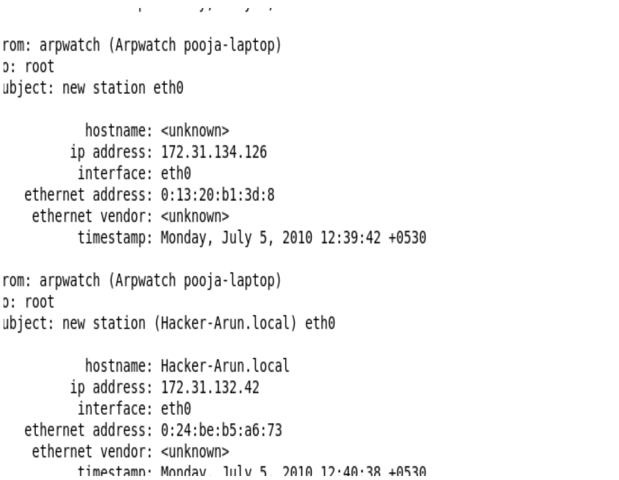

Figure 16- Record of MAC and IP addresses made by ARPwatch

This file is reloaded every time a new pair of MAC and IP address becomes known. Whenever there is any change found in MAC and IP address, then ARPwatch alerts the person that ARPspoofing of a particular MAC is done as shown in figure-17. The system executing this program as this simulated attack is that pooja-laptop gets to know that the host-name *'Hacker-Arun'* whose IP address is 172.31.132.42, interface eth0 and has now changed its MAC address from 0:24:be:b5:a6:73 to 0:30:65:24:21:36. Detection of ARP spoofing ARPwatch by first finding all of the current ARP entries by the command arp –a sends an alert. Then, one among them is selected for ARPspoofing which spoofs the victim's MAC address by the attacker's MAC address. This is detected by ARPwatch and it shows the alert by showing the old ethernet address and current ethernet address as illustrated in figure-18. arp -a command finds the current ARP entry which has the IP address 172.31.132.49 and MAC address 00:16:35:ae:56:14 which is shown in the right hand side of the figure 18.

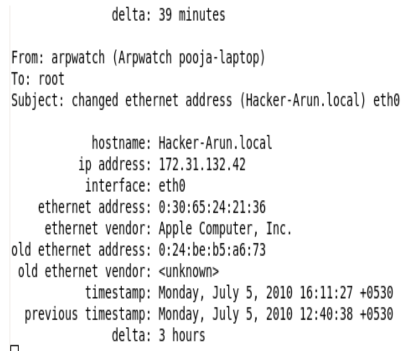

Figure- 17 Alert when change in IP and MAC address

Then, the ARP spoofing is done which is illustrated in the above side of the figure. The system whose MAC address 0:23:5a:47:cc:21, is running the ARPspoof broadcasts an ARP reply that the system having IP address 172.31.132.49 is at 0:23:5a:47:cc:21. This ARP spoofing is detected by this tool ARPwatch which is shown in the left hand side of the figure 20 i.e. hostname 'niraj-desktop' is having IP address 172.31.132.49, interface eth0, whose old ethernet address was 00:16:35:ae:56:14, is now changed to 0:23:5a:47:cc:21, the attacker's MAC address running the ARPspoof.

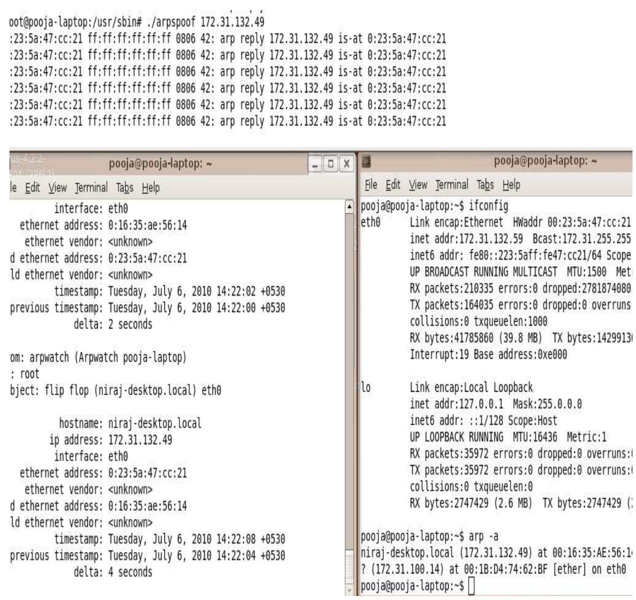

Figure-18 ARP spoofing Detected by ARPwatch

The Security Threat and the Response attack, is also an attack on a network's resources, but is launched from a large number of other host machines. This is a type of DOS attack. Attacking software is installed on these host computers, unbeknownst to their owners, and then activated simultaneously to launch communications to the target

machine of such magnitude as to overwhelm the target machine (figure 19, figure 20).

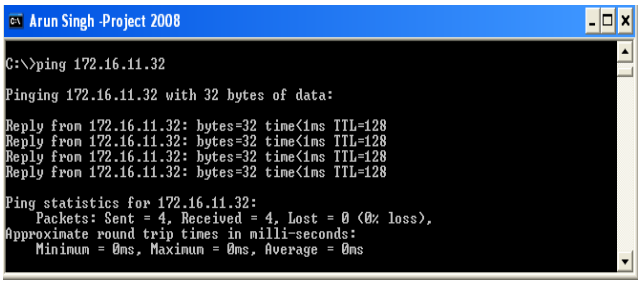

Figure- 19Ping command to check system is alive or not

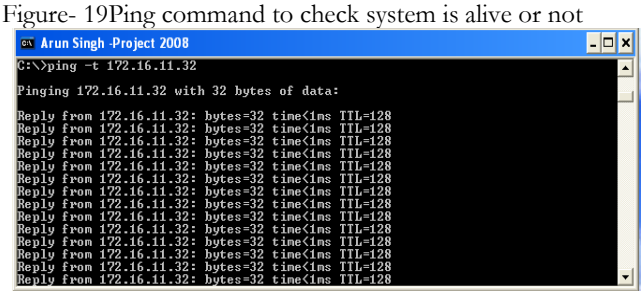

Figure -20 Dos Attack

The proposed solution which is given for ARP poisoning is to have control of the user over the ping reply i.e. if the user wants to reply the ping then only he or she can reply else not. Control of the user can be of two types either he or she ignores all the ICMP echo packets or accepts all. In the first one, the user will ignore all the ICMP echo packets i.e. the other system user will not be able to detect whether the system is host or not even if the system is actually hosting up . In the second one, the user will accept all the ICMP echo packets i.e. if any other system pings the user's system, it will reply the number of times it is asked to do so. This will increase the payload on the user's system. which may lead to crash. The proposed solution gives a way to have control on this payload which in turns, benefits the user to reply to the system once when it is pinged by another system and then stops for some time and then continue again. This pattern may be repetitive. Such repetitive pattern may be indicative of a network attack or vulnerabilities. This will reduce the payload to a very great extent which was the disadvantage of accepting all ICMP echo packets and will also inform the other users that the host is up. By this way, the proposed solution will overcome both the problems arising earlier. The solution is designed using shell script. If the user is busy and does not want to reply then it will ignore all the ICMP echo packets and continue doing his or her work even if the other system pings the user's system. But, if the user is not busy and wants to reply the trusted system so that no ARP poisoning could take place, then he/she may choose to reply to the system requesting.

```
#! /bin/bash
echo "enter the ip adress"
read i
while [ 1 ]

do
echo "0" > /proc/sys/net/ipv4/icmp_echo_ignore_all
ping $1 -w1
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
sleep 20
done
```

Figure-21 Shell script preventing DoS

```
#! bin/bash

char=" "
echo "whether u want to rply or not"
read char
echo "enter the ip address"
read i
echo "how many times you want the reply when replies"
read p
if [ $char = 'y' ]
then
echo "0" > /proc/sys/net/ipv4/icmp_echo_ignore_all
ping $1 -w $p
elif [ $char = 'n' ]
then
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
ping $1
fi
```

Figure- 22 Shell script to prevent payload

5.0 Conclusion and Future Direction of Work

Security threats and breaches in an organization's network infrastructure can cause critical disruption of business processes and lead to information and capital losses. A potent security system is imperative for an enterprise networks and vulnerability assessment is an important element for the same.

A host-based vulnerability scanning system informs about the vulnerabilities that the respective host carries. This paper provides a review of the current research related to host-based vulnerability assessment followed by avenues for further research. It is important to make a distinction between penetration testing and network security assessments. Some of the simulators have strong resemblance with the penetration testing but these differ for their purpose. The purpose has carefully been taken care to simulate the attacked and its successful solution by writing scripts for these attacks. A network security or vulnerability assessment may be useful to a degree, but do not always reflect the extent to which hackers will go to exploit a vulnerability. Penetration tests attempt to emulate a 'real world' attack to a certain degree. The penetration testers will generally compromise a system with vulnerabilities that they successfully exploited.

If the penetration tester finds several holes in a system to get in this does not mean that hackers or external

intruder will not be able to find more than the holes deleted earlier. Hackers and intruders need to find only one hole to exploit whereas penetration testers need to possibly find all if not as many as possible holes that exist. This is a daunting task as penetration tests are normally done within a certain time frame.

A penetration test alone provides no improvement in the security of a computer or network. Action taken to address these vulnerabilities that is found as a result of conducting the penetration test are not the part of penetration listing. Security is an ever-changing arena. Hackers are constantly adapting and exploring new avenues of attack. The technology is constantly changing with new versions of operating systems and applications. The result of all this change is an increased risk to the typical workstation based on popular operating system. Increased upgrades and patches are a result of the need to propagate fixes to security vulnerabilities. The quick fixes of vulnerabilities presented in this paper provide a readymade solution.

This paper also provides an overview of Network Security Monitoring (NSM) which involves network analysis through NMAP, sniffing of the packets across the traffic over the HTTP connection in LAN by Wireshark, Tshark, Dsniff. Analysis and detection of ARP poisoning are discussed briefly to highlight the vulnerabilities in the data communication over the HTTP connection in LAN. The proposed solution given in this paper to stop the MITM attack in LAN is simple to understand and provides the user to have a control over the ping reply given by its system. It allows the user to defend from the attack of ARP poisoning. The future work can extend this bash shell script to block the particular IP address if it pings the system many times and does not allow any system to send the packet with a greater size than that has been sent the first time. Analysis of data communication over HTTPS connections in LAN and se- cure routing of the network data communication over HTTP and HTTPS in LAN or Wi-Fi are the other area having applicability of the present research.

6.0 Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. The research reported here is fully supported by the ISEA Project, DIT, MCIT, and Government of India.

References

[1]. Cryptography and Network Security Principles and Practices, Fourth Edition, By William Stallings, Prentice Hall publication, 2006

[2]. Arpspoof a arp poisoning tool availble at, http://monkey.org/ dugsong/dsniff/ [Accessed on May 20, 2010].

[3]. Ettercap a arp poisoning tool, http://ettercap.sourceforge.net/ [Accessedon May 20, 2010].

[4]. HTTPS sniffing through sslsnif, http://thoughtcrime.org [Accessed onMay 20, 2010].

[5]. http://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf [Accessed on May 20, 2010].

[6]. tshark command manual at http://www.wireshark.org/docs/manpages/tshark.html [Accessed on May 20, 2010].

[7]. B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell, StrongerPassword Authentication Using Browser Extensions, Proceedings of the14th Usenix Security Symposium, 2005

[8]. Nmap Security Scanner For Network Exploration & Security http://nmap.org/

[9]. Wireshark. [Online document] Available: http://www.wireshark.org/

[10]. US-CERT Technical Cyber Securi Alert http://www.us-cert.gov/cas/tips/ST04-015.html [Last Accessed on 5th July, 2010]

[11]. Wireshark,http://www.wireshark.org/docs/man-pages/tshark.html, [Last Accessed on 8th July, 2010]

[12]. Douglas E. Comer, Internetworking with TCP/IP Principles, Protocols and Architecture, Fifth Edition,

[13]. Pearson Prentice Hall Publications, 2006Angela Orebaugh, Wireshark & Ethereal Network

[14]. Protocol Analyzer Toolkit, Syngress Publication, 2007

[15]. Chris Sanders, Practical Packet Analysis using Wireshark to solve real world Network Problems, William Pollock Publications, 2007

[16]. David Slee, Common Denial of Service Attacks, July 10, 2007.

[17]. Renaud Bidou, Denial of Service AttacksJoe Habraken, Absolute Beginner's Guide to Networking, Fourth Edition, Que Publication, 2003.

[18]. Arun Kumar Singh, Lokendra Kumar Tiwari, Shefalika Ghosh Samaddar and C.K Dwivedi, Security Policy & Its Scope in Research Area, accepted in International Conference on Strategy and Organization, ICSO 2010 on 14 & 15 May-2010, Institute of Management Technology, Ghaziabad, Uttar Pradesh, India.

[19]. Lokendra Kumar Tiwari, Arun Kumar Singh, Shefalika Ghosh Samaddar and C.K Dwivedi , Recovery Evidentiary files using Encase Ver 6.0, accepted and presented in National conference & Workshop on High Performance & Applications, 08-10 February, Banaras Hindu University, Varanasi, Utter Pradesh, India, pp-8.

[20]. Lokendra Kumar Tiwari, Arun Kumar Singh, Shefalika Ghosh Samaddar and C.K Dwivedi, Evidentiary Usage of E-mail Forensics: Real Life Design of a Case, First International Conference on Intelligent Interactive Technologies and Multimedia (IITM-2010) page 219-223, on Dec 28-30, 2010, Indian Institute of Information Technology Allahabad, Uttar Pradesh, India..

[21]. Arun Kumar Singh, Pooja Tewari, Shefalika Ghosh Samaddar and A.K.Misra , Communication Based Vulnerabilities and Script based Solvabilities, International Conference on Communication, Computing & Security (Proceedings by ACM with ISBN-978-1-4503-0464-1) on 12-14 Feb-2011, National Institute of Technology Rourkela Orissa, India .

[22]. Arun Kumar Singh, Pooja Tewari and Shefalika Ghosh Samaddar, A. K. Misra , Vulnerabilities of Electronics Communication: solution mechanism through script, International Journal of Computer Science Issues (IJCSI), Volume 8, Issue 3, 2011 (IN Press).

[23]. Arun Kumar Singh, Lokendra Tiwari , Vulnerability Assessment and penetration Testing, National Conference on Information & Communication Technology (NCICT–2011), ISBN: 978-93-80697-77-2, 5th–6th March, 2011, Centre for Computer Sciences Ewing Christian College Allahabad–211003 Utter Pradesh, India.

[24]. Lokendra Kumar Tiwari, Arun Kumar Singh, Shefalika Ghosh Samaddar and C.K Dwivedi, An Examination into computer forensic tools, accepted and to be presented in 1st International Conference on Management of Technologies and Information Security (ICIMS 2010) page 175-183, on 21-24 of January 2010, Indian Institute of Information Technology Allahabad, Uttar Pradesh, India. (http:// icmis.iiita.ac.in/TOOL_FORENSIC.ppt.

Corresponding Author: Arun Kumar Singh received his B.Tech in Electronics and Communication from SRMCEM College, Lucknow, Uttar Pradesh , India in 2005. He received his MS degree in Information Security from Indian Institutes of Information Technology, Allahabad, Uttar Pradesh, India in 2008. Currently, he is pursuing the Ph.D. degree in Computer Sciences and Engineering at the Motilal Nehru National Institute of Technology (MNNIT), Uttar Pradesh, India. He also is working as a Research Associate at the MNNIT. His research interests include network security, network protocol design and verification, in network security, Cryptography and Computer Forensic fields.