# "Topological Aspects in References to Network Security"

**Kamlesh[1]* Anju Dhull[2]**

[1]Research Scholar, CMJ University, Shillong, Meghalaya

[2]Research Scholar, CMJ University, Shillong, Meghalaya

***Abstract – Security is optimized by lack of access; connectivity is optimized by complete access. Internet enabled organizations; wireless connectivity and roaming clientage have made network peripheries relatively transparent. Communication has become network savvy. People are collaborating with peers in the real time, using tools for convenience rather than security. Data has started to flow iii and outside the organization through wireless media and many users request a roaming profile, so that they can access parent network even from faraway places. Enterprises continue to invest heavily in perimeter security i.e. to bring security around the network, but not realizing the fact that security has to be within the network, i.e. in the network fabric itself not only at the periphery.***

----------♦----------

## 1. INTRODUCTION

In a computer network, technological aspects are often the strongest point of defense from the outside attacks. But most attackers know that it is difficult to penetrate the periphery, so they look for easier prey. In the quest might he roaming users accessing the network and social and /or engineering methodologies to break—i.e. threat not only lies at the periphery hut might be deep rooted into the network itself. The type of threat and the means by which it gains entry to the protected assets constitute a threat vector. As per [Ins04] the total percentage of internal threats is quoted many times higher i.e., these many computer crimes, attacks and violations originate from trusted employees as shown in Figure 1.2.

**Figure 1.2: Crime Loss Statistics**

A Firewall or IDS can do nothing to protect against inside attacks. rather a firewall can provide a false sense of security, because it is common assumption that firewalls block all unwanted access, which is not completely true - firewalls allows many types of traffic to pass, some of which may he malicious. Fragmented packets or ICMP messages are tunnel through otherwise working firewall, allowing all attacker to directly access the protected resources.

Dial-up modems that accept connections also contribute to internal threat vectors. E.g. Internal network is behind firewall, an ids and proxy etc. And users are not allowed the access to voice chat. These users knowingly or otherwise connect to internet for voice chats etc. Using dial up connections which completely bypasses the network security realm of an organization as shown in figure 1.3.

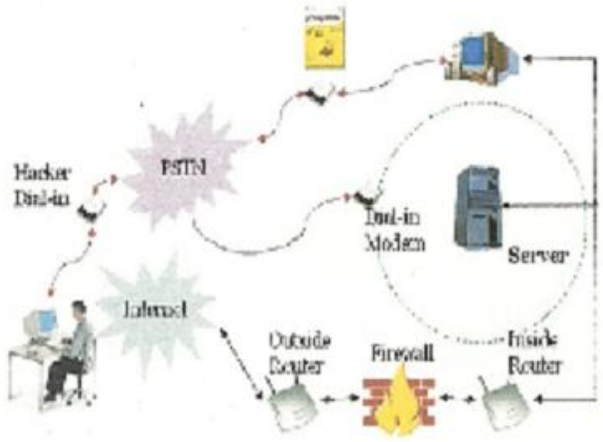


**Figure 1.3: Dial-up Connections bypassing Network Periphery Security**

Dual—homed systems often configured by administration for their case to access internal as well as external network also pose a great threat to the networks. Another potential problem is use of girlfriend programs. It refers to a program handed to an employee on a floppy or CD by a trusted friend, that actually contains a Trojan program (designed to open connection on the employee's machine. They can be difficult detect and eliminate. Inside threats, although they create some of the most hazardous and ubiquitous risks to networks, are often overlooked by security strategies [RB04].

Outside threat vector's most command and universal threat is the script kiddie. The script kiddie is someone looking for the easy kill. They are not out for specific information or targeting a specific company. Their goal is to gain super-user access in the easiest way possible. They do this by focusing on a small number of exploits, and then searching the entire Internet for that exploit. Sooner or later they find someone vulnerable [Pro00].

The script kiddie methodology is a simple one. It scans the Internet for a specific weakness, when they find it, they exploit it. Most of the tools used are automated, requiring little interaction. Some of them are advanced users who develop their own tools and leave behind sophisticated backdoors. Others have no idea what they are doing and only know how to type "go" at the command prompt. Regardless of their skill level, they all share a common strategy; randomly search for a specific weakness, then exploit that weakness [Pro00].

Every network security implementation is based on some model, which could he either specified or assumed. Mostly perimeter security models based on Firewalls and/or IDS are in uses which are reactive in nature. This model obviously with above mentioned risks lacks the robustness and provides false sense of security infrastructure. With tremendous complexity and hacking ease looming around; challenge is to build security into the network itself. This will lead to self-healing and self-defending network infrastructure. To achieve this, security has to be proactive i.e. should he part of the switching fabric that carries all the traffic: begin and malicious. There is compelling need to combine reactive and proactive security measures in order to have an integrated approach to the security across the information value chain.

## RESULTS & DISCUSSION

Every network security implementation is based on some model, which could be either specified or assumed. Based on the literature survey it is apparent that mostly perimeter security model based on firewalls and IDS, is in use: which is reactive in nature. Reactive approach, obviously with above mentioned risks lacks the robustness and provides false sense of security infrastructure. With tremendous complexity and hacking ease looming around; challenge is to build security into the network itself. This will lead to self-healing and self-defending network infrastructure. To achieve this security has to be proactive i.e. should be part of the switching fabric that carries all the traffic: benign and malicious. There is compelling need to combine reactive and proactive security measures in order to have an integrated approach to the security across the information value chain.

Keeping this ill view, it is proposed to design and develop, A Proactive Network Surveillance Framework. Proposed Framework aims to provide learning vision to the network attacks thus exhibiting ability to react intelligently. Proactive network security framework will be based on a "military Doctrine" which would address and eradicate major shortcomings of existing security system Research Work will e he Defense depth sometimes also called elastic defense concept for implementation purposes. Defense in depth seeks to delay rather than prevent the advance of an attacker, buying time by yielding space. The idea of defense in depth is now widely used to describe nonmilitary strategies like network security. Successive layers of defense may use different technologies or tactics. The inner layers of defense can support the outer layer and an attacker must breach each line of defense in turn.

This gives an engineering solution which emphasizes redundancy - a system that keeps working even when a single component fails e.g. an aircraft with four engines will be less likely to suffer total engine failure than a single-engine aircraft: no matter how much effort goes into making the single engine reliable. Different security vectors within the network, helps to prevent a shortfall in

any one defense leading to total system failure.

Subsequent chapters will elaborate upon framework design, implementation, deployment and testing.

**CASE-I**

The whole framework was tested against various threat vectors. Setup of the test bed is shown in Figure 1.4

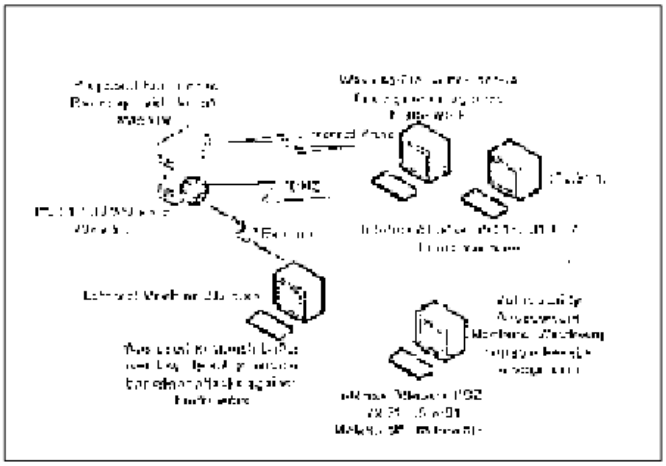

**Figure 1.4: Test Bed for Testing the Proposed Framework**

Firstly Core security layer was tested, Linux Red hat 9, Windows 2000 and proposed framework was installed on three different machines and allowed physical access to the systems. In security community it is said, once attacker has a physical access to the system, system no more belongs to the owner. Proposed framework is strengthened by making file system level changes, which are not recognized by standard utilities. Test cases were successfully able to mount the Linux, Windows partitions on other system and also it was tested that once hard drive is removed from the system and configured to work as slave, whole data on the chive was accessible.

On the other hand, proposed framework was able to restrict remote access thus not allowing to get mounted. Also when configured as slave machine local mount utilities were not able to recognize the file system type.

Next step was to lest the framework against active fingerprinting tools like nmap. Nmap was executed against the framework and following results were observed: [rootcns1 /1# nmap -v —sS —o 172.31.1.6 Starting nmap V. 3.00 (www.insecure.org/nmap/) Host (172.31.1.6) appears to be down, skipping it.

Note: Host seems down. If it is really up, but blocking our ping probes, try —P0 Nmap run completed -- 1 IP address (0 hosts up) scanned. Following results shows nmap fingerprinting fails to detect the operating environment when deny all firewall rule is fired: [rootcns1 /]# nmap -v —sS —0 —P0 172.31.1.6 Starting nmap V. 3.00 (www.insecure.org/nmap/) Host (172.31.1.6) appears to be up ... good. Initiating SYN Stealth Scan against (172.31.1.3) The SYN Stealth Scan took 1722 seconds to scan 1601 ports.

All 1601 scanned ports on (172.31.1.6) are: filtered too many fingerprints match this host for me to give an accurate OS guess TCP/IP fingerprint:

SInfo(V=3.00%p=i1386-redhat-linux gnu%d=7/19%time=44BDE628%0=-1%C=-1) T5(Resp=N) T6(Resp=N) T7(Resp=N) PU(RespN) Nmap run completed -- 1 IP address (1 host up) scanned in 1942 seconds.

After opening access for port number 22 (SSH) nmap was able to fingerprint it as Linux-2.4.7 as shown:

[rootns1 /1# nmap —v —sS —O —P0 172.31.1.6 Starting nmap V. 3.00 (www.inse.cure.org/nmap/) Host (172.31.1.6) appears to be up ... good. Initiating SYN Stealth Scan against (172.31.1.6)

Adding open port 22/tcp The SYN Stealth Scan took 750 seconds to scan 1601 ports. For osscan assuming that port 22 is open and port 32473 is closed and neither are firewalled Interesting ports on (172.31.1.6): (The 1600 ports scanned but not shown below are in state: filtered) Port State Service 22/tcp open ssh Remote operating system guess: Linux 2.4.7 (X86) TCP Sequence Prediction: Class=random positive increments Difficulty=1129696 (Good luck!) IPID Sequence Generation: All zeros Nmap run completed -- 1 IP address (1 host up) scanned in 755 seconds.

Next, Tenable Nessus was executed to find the vulnerabilities in the proposed framework. A new policy with many backdoors enabled options was used to lunch attacks against the framework. Tenable Nessus showed no vulnerability found in the framework. The network traffic was captured using tcpdump as #tcpdump —s 1600 —w /logs/tcpdump.log

Captured file was taken for analysis and analyzed using Wireshark network protocol analyzer. Flow Graphs showing three-way handshake sequence as launched by two attacker machines, the protocol hierarchy summary and TO graphs thus obtained are given below: Most of the attacks use TCP traffic. Figure 6.2 shows 99.33% of the traffic is TCP and Figure 6.3 shows Flow Graphs emphasizing three-way handshake sequences launched

by attacker machines on various ports of the framework.

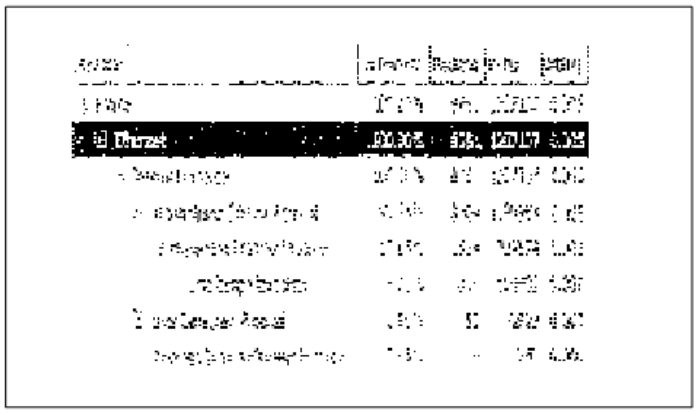

**Figure 1.5: Protocol Hierarchy Statistics**

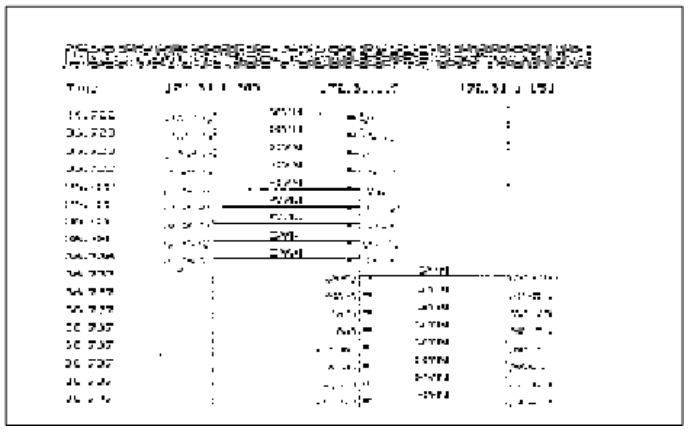

**Figure 1.6 10 Flow Graphs**

A low interaction Genl Honeypot with two virtual linux and two Window hosts is configured at Layer 5. Before configuring and running Honeyd, is was ensured that the Honeyd host responds to arp request, for the IPs of the honeypots virtually hosted. This was achieved by using the arpd software to spoof arp responses on behalf of the honeypots. #./arpd 172.31.1.0/24

Given below is the test configuration file to set up virtual hosts with user specified services running on it.

create Linux

set Linux personality “Linux 2.2.19”

add Linux tcp port 23 “sh scripts/telnet’sh”

add Linux tcp port 22 open

set Linux default tcp action reset

set Linux udp action reset

bind 172.31.1.110 linux

bind 172.31.1.112 linux

create windows

set windows personality “Windows NT 4.0 Server

SP5—SP6”

set windows default tcp action reset

set windows default udp action reset

add windows tcp port 80 “perl scripts/iisemulator—

0. 95/iisemul8. p1”

add windows tcp port 139 open

add windows tcp port 137 open

add windows udp port 137 open

add windows udp port 135 open

set windows uptime 42002

bind 172.31.1.101 windows

bind 172.31.1.102 windows

The above line creates two templates called ‘linux” and ‘windows” and bind the honeypot IP addresses to the templates. The linux template tells honeyd to present itself as a “Linux 2.2.19” when any machine tries to fingerprint it with NMap or XProbe. Ports 22 and 23 are opened on both linux virtual machines. Script telnet.sh will emulate the default behavior at port 23. In case of windows machines template present itself as a Windows NT 4.0 SP5-SP6. Five ports are open on the honeypot, 80/tcp, 139/tcp, 137/tcp, 137/udp and and 135/udp. When a machine connects to port 80 of the hoiieypot, the honeypot will engage the client with an ITS emulator pen script.

For ports that are closed, the configuration specifies that a RST be sent in the case of TCP. And an ICMP Port Unreachable message is sent for UDP.

Framework evaluation shows that low interaction honeynet is effective in creating virtual hosts across the network and successfully deceiving fingerprinting tools. This layer can be helpful in various areas of system security specifically, detecting active fingerprinting scans, flooding traffic analysis, creating operating system personalities and more

importantly detecting the unknown.

## METHODOLOGY

The proposed network framework has been implemented using “onion model of defense. This layered system operating at five layers provides better security scenarios as depicted by the analysis and results. The experimental results for the proposed framework demonstrate efficiency and effectiveness of the framework for solving network security problem. Further, it is observed from the various network traffic clumps and graphs that network traffic visualization helps to (I(Xhl1(C formal results in a relatively small amount of time. The output of the whole research work is bundled into a new Linux distribution with the capabilities to get installed with minimum deployment efforts.

## REFERENCES

[AH03] C Papadopoulos A Hussain, J Heidemann, A framework for classifying denial of service attack, In proceedings of sigcomm, 2003.

[A1e96] Michael Alexander, The underground guide to computer security, Addison-Wesley Publishing Company, 1996.

[All03a] Honeynet Research Alliance, Sebek- keylogging tool, Http://www.honevnet.org/tools/sebek/, 2003.

[All03b] Honeynet Research Alliance, Snort-inline, snort augmentation tool. Http: //snort-inline.sourceforge.Net/, 2003.

[All03c] Honeynet Research Alliance, Virtual honeynets, Http://www.honeynet.org/papers/virtual/, 2003.

[Amo94] Edward G Amoroso, Fundamentals of computer security technology, Prentice-Hall PTR, Upper Saddle River, N.J, 1994.

[Bar89] R Barden, RFC 1122 - Requirements for Internet Hosts-Communication Layers, Internet Engineering Task Force, Http://www.faqs.org/rfcs/rfc1122.htm1, 1989.

[Bee06] Gerard Beekmans, Linux From Scratch, Version 6.2, Linux from scratch: Project, http://www.linuxfromscratch.org, 2006.

[Bel89] S M Bellovin, Security problems in the TCP/IP protocol suite, ACM computer communications review, vol. 19, No. 2, 1989.

[Bil91] Cheswick Bill, An Evening with Berferd in Which a Craker is Lured, Endured and Studied, Find out from Net, 1991.

[Biso3] Matt Bishop, What is Computer Security, IEEE Security and Privacy, 2003.

[CA-01] CERT Advisory CA-2001- 19, CodeRed Worm Exploiting Buffer Overflow In IIS Indexing Service DLL. http://www.cert.org/advisories/CA-2001- 19.html, 2001.

[CA-02a] CERT Advisory CA-2001-31, Buffer Overflow in CDE Subprocess Control Service. http://www.cert.org/advisories/CA-200 1-31 html, 2002.

[CA-02b] CERT Advisory CA-2002-01, Exploitation of Vulnerability in CDE Sub- process Control Service, http://www.cert.org/advisories/CA-2002-01 html, 2002.

[CAI01] CAIDA, CodeRed Worms a Global Threat, http://www.caida.org/ analysis/security/code-red/, 2001.

[Car96] Lorrie Faith Carnor, Internet Security a Public Concern, ACM, SIGSOFT, 1996.

[Cas96] Manuel Castells. The rise of the network society, Blackwell Publishers, 1996.

[CC04] CERT CC, Denial of Service Attacks, http://www.cert.org/tech tips/denial of service.html, 2004.

[CD95] Zwicky E Chapman D, Internet Security Firewalls, O” Reilly, 1995.

[Cer93] Vinton G Cerf, Core Protocols in Internet System Handbook, Addison-Wesley Publishing Company, Inc., Greenwich, CT, 1993.

[CER95] CERT, IP spoofing attacks and hijacked termrnai connections, CERT Advisory CA, 1995.

[CER96] CERT, TCP SYN flooding and ip spoofinq attacks, CERT Advisory CA, 1996.

[CER03] CERT, Incident and Vuinerohiiity Trends, CERT Coordination Center Soft-Ware Engineering Institute Carnegie Mellon University Pittsburgh, PA. 2003.

[CH96] Karanjit Sijan Chris Hare, Internet Firewails and Network Security, New Riders Publishing, Indianapolis, 1996.

[Che99] Bill Cheswick, Map of the Internet, http://www.cheswick.com/ches/map/gallery/isp-ss.gif. 1999.

[Coh97] Fred Cohen, Deception Toolkit, http://www.all.net/dtk, 1997.

[Co103] Eric Cole, Hackers Beware, New Riders, Indianapolis, 2003.

[Con06] Internet Systems Consortium, Internet Domain Survey Host Count, http: //www.isc.org/index.pl?/ops/ds/, 2006.

[CSI04] CSIFBI, Computer Crime and Security Survey, CSI FBI, 2004.

---

**Corresponding Author**

**Kamlesh***

Research Scholar, CMJ University, Shillong, Meghalaya

**E-Mail – anju_jind@rediffmail.com**