Design and Implementation of a Medium Interaction Honeypot

Ayeni O.A
Department of Computer Science
Afe Babalola University, Ado Ekiti, Nigeria

Alese B.K
Department of Computer Science
Federal University of Technology Akure, Nigeria

Omotosho L.O.
Department of Computer Science
Afe Babalola University, Ado Ekiti, Nigeria

ABSTRACT

Security in computing world is a serious issue and must be handle with utmost care, hence the need to always protect and secure our networks as more and more business are been conducted through the internet. The expansion of the World Wide Web has given unlimited access to attackers to prey on ignorant administrator who lacks basic knowledge of network security. Vulnerabilities in common security components such as firewalls, security patches, access control and encryption are inevitable, so hackers take advantage of these loopholes to break into computer networks.This paper presents the result of a research that was carried out using a medium interaction honeypot, a virtual machine ware workstation, snort software and entropy-based model for capturing, analyzing and detection of malicious traffic targeted at the network. A ring topology network of three system was design using virtual machine work station, a Snort software was installed on all the three machine to capture traffic on the network and entropy-based mathematical analysis was conducted on the traffic to detect attack/malicious traffic. The entropy $H(x) = -\sum_{i=1}^{N}(P_i) * \log_2(P_i)$ where $P_i = N_i/S$. N is a set of positive integer that represent the total number of server on the network, n_i represent the size of the traffic in bytes and S represent the total length of the traffic that constitute the traffic. The result of the research work shows detection of malicious traffic and also limit the rate of denial of service targeted at the network.

Keywords

honeypot, security, Network, traffic & detection.

1. INTRODUCTION

In a physical structure, such as a building, strong materials for construction are used to provide the necessary security. By means of security, windows are located so that thieves can not access them easily; barriers are placed around the building and access is controlled on each entry. Besides these, systems of caution, alarms and cameras are placed to monitor the inside, in addition to properly equipped personnel, continuously, patrolling the installation. Similar to physical security, information security managers have utilized multiple technologies to keep their networks safe. Honeypots based on Anjali Sardana etal(2008) [4], a proactive detection mechanism, are machines that are not supposed to receive any legitimate traffic and, thus, any traffic destined to a honeypot is most probably an ongoing attack and can be analyzed to reveal vulnerabilities targeted by attackers. When they are connected to the Internet, any users other than their system administrator are unauthorized by definition and probably malicious. They yield much richer log and intrusion-detection data for attacks than is possible by monitoring ordinary computer systems or networks. Honeypots are often connected in networks called "honeynets" to see how attacks exploit networks.

There are mainly two reasons why information security continues to receive an increasing amount of attention. Firstly, new services providing critical services demand an increased level of security. Secondly, there is an ever growing increase in reported incidents and attempted attacks on computer systems. That is, not only are we more concerned with information security as a result of the impact a compromise would create on our lives, we also receive daily updates on discovered vulnerabilities and successful attacks.

Common strategies for providing security focuses on the prevention of attacks, through the means of firewalls, security patches, access control etc. The downside to this approach is that it assumes that "perfect" systems that are practically impenetrable can be built. History has however shown that this is relatively seldom the case, and even good systems suffer from the occasional fault. A better solution may be to assume that intrusions may occur and focus on handling them at that point in time. The challenge is to design systems that may tolerate intrusions and still being able to provide the given services.

2. HONEYPOT

A Honeypot can be characterized as a closely monitored network decoy serving several purposes. Honeypots can be set up to run any type of operating system and any number of services. The value of a Honeypot is directly proportional to the amount and type of information we can successfully obtain from it. Aside from information gathering, a Honeypot has the capabilities of distracting adversaries from more valuable machines on a network, and can provide early warning signs about a new type of attack or exploitation trends, and allows in-depth examination of adversaries during or after exploitation of a host. Another function that a Honeypot allows is the capturing the keystrokes typed by an adversary attempting to compromise the Honeypot – this provides particularly interesting insight if an intruder uses the compromised host as an IRC chat server.

Their currently exist two types of Honeypots: a *physical Honeypot* which is a real machine with its own IP address, and a *virtual Honeypot* which is simulated by another machine that responds to network traffic. Physical Honeypots are often labeled as high-interaction because the system can be completely compromised and are resource expensive to install and maintain. For example - if you wanted to implement physical Honeypots for a certain range of IPs on your LAN you would have to build a separate instance of a Honeypot for each physical IP address. Virtual Honeypots are often labeled as low interaction because of the low implementation and maintenance costs. A virtual Honeypot

can simulate multiple Operating Systems, services and a separate TCP/IP stack for each instance of a Honeypot on that one machine. Virtual Honeypots are used more often than physical Honeypots because they require fewer computer systems, which in turn reduces maintenance costs, and also allows for a greater variety of hosts to be deployed and observed. Honeypots are important because they allow interaction between the attacker and the honeypot.

Honeypots based on Anjali Sardana etal(2008) [4], a proactive detection mechanism, are machines that are not supposed to receive any legitimate traffic and, thus, any traffic destined to a honeypot is most probably an ongoing attack and can be analyzed to reveal vulnerabilities targeted by attackers. When they are connected to the Internet, any users other than their system administrator are unauthorized by definition and probably malicious. They yield much richer log and intrusion-detection data for attacks than is possible by monitoring ordinary computer systems or networks.

Honeypots can be classified into three different levels; Low-Interaction Honeypots, Medium-Interaction Honeypots and High-Interaction Honeypots. In terms of installation, configuration, deployment and maintenance, the low-interaction honeypots are the easiest to implement. Basic services such as Telnet and FTP are emulated on low interaction honeypots. It limits the hacker to interact with only these few pre-configured services. For example, a honeypot could emulate a Windows 2000 server running with several services such as Telnet and FTP. A hacker could first telnet to the system to get a banner which would indicate what operating system the honeypot is running on. . The main objective of low-interaction honeypot is only to detect, such as unauthorized probes or login (http://www.securitydocs.com/library/2692[43].Low-interaction honeypots can be easily installed on the system and configured to any of the services specified above. This low-interaction honeypot is both easy to deploy and maintain. But to prevent the system from being fully exploited by hackers, there is a needs to ensure patch management on the host system and to conscientiously monitor the alert mechanisms. Low-interaction honeypots have the lowest level of risk. The honeypot cannot be used as a launch pad to attack other systems as there is no legitimate operating system for the hacker to interact with. The low interaction honeypot is only good at capturing known attack patterns, but is worthless at interacting or discovering unknown attack signatures. (http://www.securitydocs.com/library/2692) [43] Another type of honeypot is the Medium-Interaction Honeypots. In terms of interaction, this is a little more advanced than low-interaction honeypots, but a little less advanced than high-interaction honeypots. Medium-Interaction honeypots still do not have a real operating system, but the bogus services provided are more sophisticatedtechnically.(http://www.securitydocs.com/library/2692).[43] The final and most advanced of honeypots are the high-interaction honeypots. These kinds of honeypots are really time-consuming to design, manage and maintain. Among the three types of honeypots, this honeypot possess a huge risk. But, the information and evidence gathered for

3. Denial of Service attack (DoS).

A Denial of Service (DoS) attack is commonly characterized as an event in which a legitimate user or organization is deprived of certain services, like e-mail or network connectivity, that they would normally expect to have. The widespread need and ability to connect machines across has caused the network to be more vulnerable to intrusions and has facilitated break-ins of a variety of types. According to CERTStatistics (http://www.cert.org/stats/cert) [6], a mere 171 vulnerabilities were reported in 1995 which boomed to 8064 in the year of 2006. Apart from these, a large number of vulnerabilities go unreported each year. DoS attacks based on Garber (2000) and Houle et al(2001) [16], inject maliciously-designed packets into the network to deplete some or all of these resources. The attack power of a DoS attack is based on the massive number of attack sources instead of the vulnerabilities of one particular protocol. DoS attacks, which aim at overwhelming a target server with an immense volume of useless traffic from distributed and coordinated attack sources, are a major threat to the availability of resources and stability of the Internet. In many cases when a sustained high bandwidth attack reaches servers, it is not possible to contain the attack at border gateway as the offending packets have already consumed the finite bandwidth available, which makes resources unavailable to client. In this case, having a good relationship and clear communication channels with the servers are essential. High bandwidth attacks have a direct impact on the network. Diluted low rate attacks are critical component and remain undetected until the network functionality becomes unstable thus targeting Quality of Service (QoS). However, since Servers network are closer to the source of the attack they are in a better position to filter the offending traffic.

4. SYSTEM DESIGN

The requirement for the design of the medium-level dynamic honeypot system include: An operating system such as Windows 2003 Professional, with a 1GHz processor, 512 Mb of RAM, with a 10/100 network card already and a CDROM or DVD/RW drive. Windows 2003 Professional was the best choice to since it can be secured the most from the operating systems, other operating system that can be use include: Windows XP, Windows 2000 Server and Windows 2003 Professional. A program called Snort was installed on the system. This program is an open source network intrusion prevention and detection system utilizing a rule-driven language, which combines the benefits of signature, protocol and anomaly based inspection methods. Snort is the most widely deployed intrusion detection and prevention technology worldwide and has become the de facto standard for the industry. (http://www.snort.org/) [41]. A system designed to redirect network traffic flow is positioned at the network gateway as point of presence (POP). All the traffic flows arriving at the Point of Presence (POP) of a destination network server to be protected from DoS attack are tagged as either legitimate or attack. Whenever a packet belonging to suspicious flow arrives at the POP, instead of sending that packet to the active FTP server or dropping it, it is redirected to honeypot server. This provides a proactive approach to mitigation against the attack because the FTP server is isolated from attack traffic and bandwidth of the links with FTP server will not be exhausted by the voluminous attack traffic.

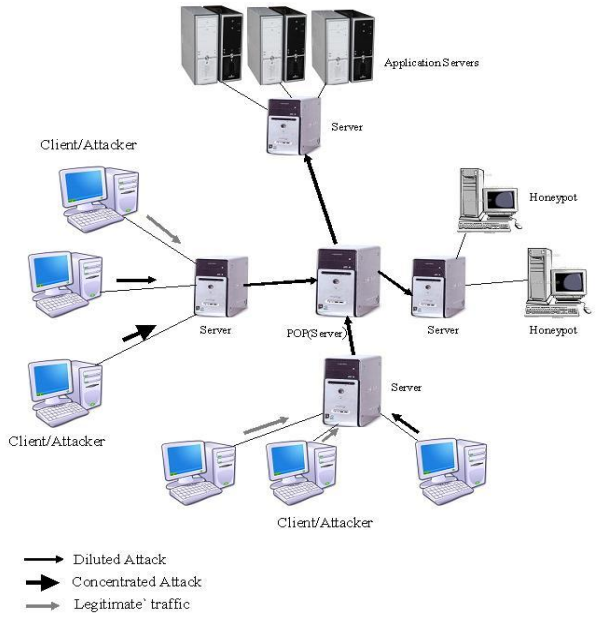

Fig. 1. The designed system.

The system detect and characterize attacks treats DoS anomalies as events that disturb the distribution of traffic features. For example, a DoS attack, regardless of its volume, will cause the distribution of destination address to be concentrated on the victim address or server. As proposed by Sardana et al (2008) [4], we use entropy to capture the degree of dispersal or concentration of a distribution flow. The sample entropy $H(X)$ is

$$H(X) = -\sum_{i=1}^{N} (P_i) \text{ x } log_2(P_i)$$

Where $P_i = n_i/S$, N is a set of positive integer that represents total number of system (server) on the network, n_i represents a flow of traffic at i. The value of sample entropy lies in the range $0 \; through \; log_2 N$. The metric takes on the value 0 when the distribution is maximally concentrated, *i.e.*, all observations are the same. Sample entropy takes on the value $log_2 N$ when the distribution is maximally dispersed, *i.e.* $n_1 = n_2 = \ldots . n_n$

Figure 1 shows the request that are coming from client/attacker which have to go through entropy test at the point of presence (pop) to determine if the flow is legitimate or malicious.

At the server node *i* (pop), a traffic flow is the aggregate of several legitimate and possibly of some attack, where $l = (l1, l2, \ldots, lj, \ldots, l(N_{fl}))$ are legitimate traffic flow and $a = (a1, a2, \ldots, ai, a(N_{fa}))$ are attack or malicious flow. The set of flow *FL* contains *a* and *l*. The total traffic rate $_i$ arriving at pop is composed of two part.

$$\partial = \sum_n \partial_i^n, \text{n} + \sum_d \partial_i^d, \text{d}$$

Where ∂_i^n, n is the legitimate incoming traffic rate which belongs to normal flow *n*, and ∂_i^d, d is the arrival rate of attack packets belonging to flow d. Any traffic characterized as attack, its destination address changed to an honeypot system address and redirected at one of the randomly selected honeypot system.

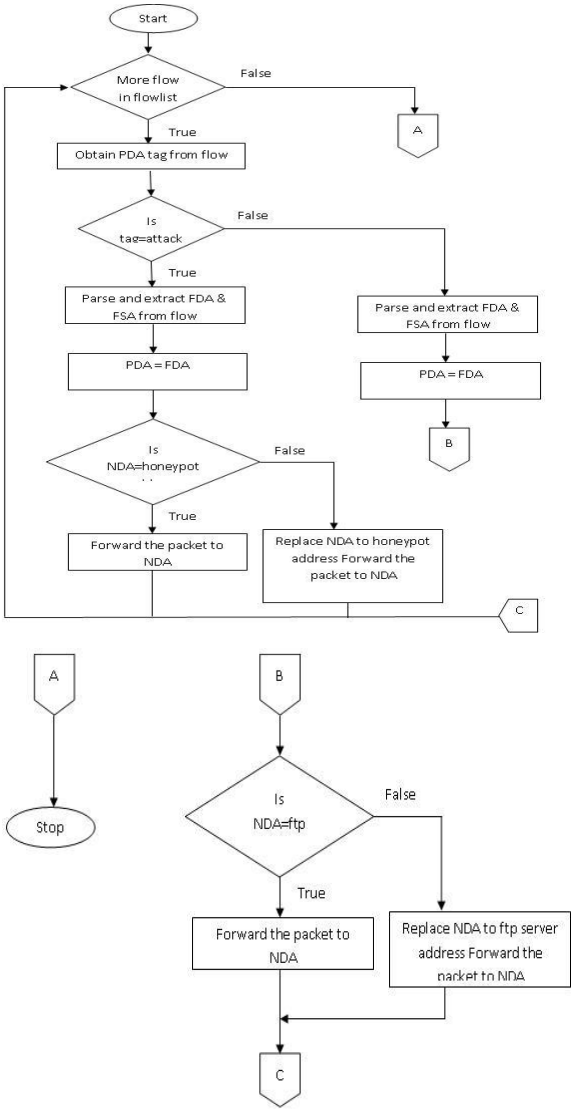

Fig. 2 Flowchart for redirection Algorithm

4.1 Detection of attack

A random process $\{X(t), t = j\Delta, j \in N\}$, where d a constant time interval is called time window, *N* is the set of positive integers that represents total number of servers, and for each $X(t)$,t is a random variable. Here $X(t)$ represents the number of packet arrivals for a flow in $\{t - \Delta, t\}$. . $X(t)$. It is found in Sardana et al (2008) [4] that simulation without attack that Entropy $H(X)$ value varies within very narrow limits after slow start phase is over. This variation becomes narrower if d is increased i.e. monitoring period. The average of $H(X)$ is taken and designated as normal Entropy $H_n(X)$. To detect the attack, the entropy $H_c(X)$ is calculated in shorter time window d □ continuously, whenever there is appreciable deviation from $H_n(X)$, attack is said to be detected. It is assume that the system is under attack at time t_a, which means that all attacking sources start emitting packets from this time: the network is in normal state for time $t_a < t$ and turns into attacked state in time t_a . a denote our estimate on t_a At time t_d following event triggers

$(H_c(x) > (H_n(x) + a \text{ x } d)) \text{ U } (H_c(x) < (H_n(x) - a \text{ x } d))$
attack = true

5. SYSTEM IMPLEMENTATION.

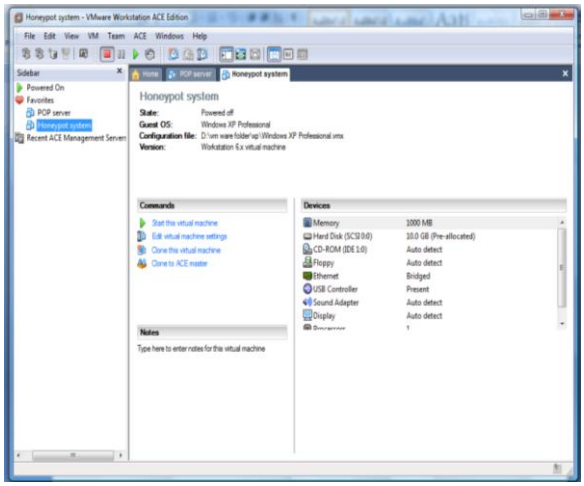

Fig.3 VM ware honeypot system.

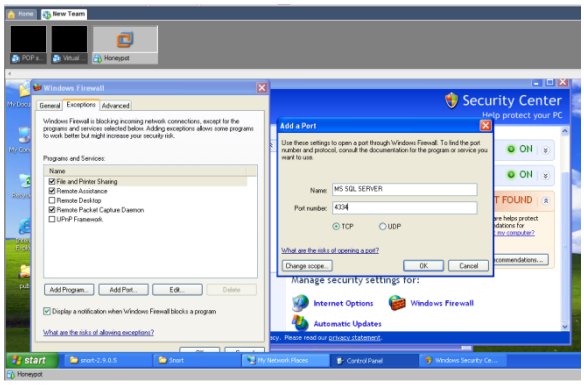

Fig.4 creating port for application.

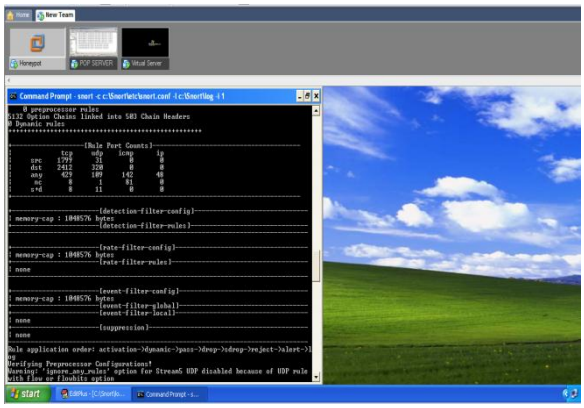

Fig.5 honeypot system.

5. RESULTS.

The data below shows the sample result in log file from the network packet at POP server:

08/21-04:55:40.702060 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 192.168.0.130:100 -> 192.168.0.132:100

08/21-04:55:41.830478 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 209.85.143.104:80 -> 192.168.0.130:1048

08/21-04:55:41.626241 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 192.168.0.130:1049 -> 192.168.0.132:80

08/21-04:55:42.141453 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 209.85.143.99:80 -> 192.168.0.132:1049

08/21-04:55:42.410615 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 209.85.143.99:80 -> 192.168.0.132:1049

08/21-04:55:42.830478 [**] [1:1000002:1] extracting packet data from network connection norm[**] [Priority: 0] {TCP} 192.168.0.130:1049 -> 192.168.0.132:80

08/21-04:55:42.255600 [**] [1:1000002:1] extracting packet data from network connection norm[**] [Priority: 0] {UDP} 192.168.0.130:1050 -> 192.168.0.132:80

The data below shows the result from the network packet from honeypot system

08/21-02:55:40.702060 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 192.168.0.130:1048 -> 192.168.0.129:1048

08/21-02:55:41.203766 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 209.85.143.104:80 -> 192.168.0.130:1048

08/21-02:55:41.626241 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 192.168.0.130:1049 -> 192.168.0.132:80

08/21-02:55:42.141453 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {UDP} 209.85.143.99:80 -> 192.168.0.130:1049.

Packet size(byte)	Total length	Source IP	port	protocol	entropy	Remark
1389	1589	192.128.76.126	515	Tcp	0.170	attk(syn)
180	121	205.128.88.20	120	Tcp	-0.85	norm
1460	1923	192.168.99.95	401	Tcp	0.302	attk(syn)
335	552	192.68.99.95	580	Tcp	0.437	attk(fin)
856	1184	112.173.75.120	95	Tcp	0.34	attk(fin)
264	152	155.12.168.10	85	Tcp	-1.383	norm
845	1811	180.101.24.160	132	Tcp	0.513	attk(syn)
992	1312	175.112.55.10	400	Tcp	0.305	attk(syn)
1272	1802	198.171.112.80	250	Tcp	0.355	attk(syn)
792	1604	205.115.28.162	95	Tcp	0.503	attk(syn)

Table 1. Traffic characterization at pop.

The maximum value of entropy here is 0, so any value greater than 0 is an attack and any value lesser than 0 is categorize as normal traffic as shown from table 1 above.

6. CONCLUSION

A simple virtual network of three systems was designed to implement the concept of the system. This system was connected to the internet to allow interaction with the network The design and configuration of this honeypot was

implemented using a virtual machine(VM ware) workstation to detect attack or malicious traffic on a network. The Point of Presence (POP) server serve as a link to the two other systems; honeypot system and the application server with virtual application running in it to give the impression of the presence of useful resources.

7. REFERENCE

[1] A.Lakhina, M. Crovella, and C. Diot, "*Mining Anomalies Using Traffic Feature Distributions*," ACM SIGCOMM, 2005.

[2] A. Sardana and R. C. Joshi, "Simulation of Dynamic Honeypot Based Redirection to Counter Service level DDoS Attacks". In *Proceedings of ICISS 2007, Springer LNCS 4812*, pp. 259–262, 2007.

[3] Anjali Sardana, Krishan Kumar and R. C. Joshi, "Detection and Honeypot Based Redirection to Counter DDoS Attacks in ISP Domain" *In Proceedings of IEEE Third International Symposium on Information Assurance and Security*. Manchester, UK, pp. 191-196, *Aug* 2007

[4]. Anjali Sardana and R. C. Joshi1 *An Integrated Honeypot Framework for Proactive Detection, Characterization and Redirection of DDoS Attacks at ISP level*, 2008

[5] S. Bellovin (1992). "There be dragons". *Proceedings of the Third Usenix Security Symposium*, Baltimore MD.

[6]. B. Stephan, "Optimal filtering for denial of service mitigation," In *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, Vol. 2, pp. 1428 – 1433, Dec. 2002.CERT Statistics http: //www.cert.org/starts/cert

[7] B. Cheswick (1992). An evening with Berferd in which a cracker is lured, endured, and studied. Proceedings of the Winter *USENIX Conference*, San Francisco

[8] Christian Döring: Improving network security with Honeypots, pages B34-B36, Master's thesis University of Applied Sciences Darmstadt, Department of Informatics, 2005

[9] Clifford Stoll: The Coocoo's egg, Pocket Books 1990

[10] C.M. Cheng, H.T. Kung, and K.S. Tan, "Use of spectral analysis in defense against DoS attacks". In *Proceedings of IEEE GLOBECOM 2002*, pp. 2143-2148, 2002.

[11] Eric Peter, epeter(at)wustl(dot)edu and Todd Schiller, tschiller(at)acm(dot)org (A project report written under the guidance of Prof Raj Jain). A practical guide to honeypot.

[12] [ForeScout02] ForeScout Technologies. "Beyond Detection: Neutralizing Attacks Before They Reach the Firewall". Summer 2002

[13] [ForeScout04] ForeScout Technologies, Inc. January 2004. [http://www.forescout.com/]

[14] J. Mirkovic, J. Martin ,and P. Reiher, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms". Technical Report 020018, Computer Science Department, University of California, LosAngeles,2002.

[15] Johansson, Karsten. "Offensive Operations Model". KSAJ Inc. August 2001. [http://www.penetrationtest.com/]

[16] K.J. Houle, G. M. Weaver, N. Long, and R. Thomas. "Trends in denial of service attack technology".Technical Report Version 1.0, CERT Coordination Center, Carnegie Mellon University, 2001.

[17] K.J. Ioannidis, and S. M. Bellovin, "Implementing Pushback: Router-Based Defense against DDoS Attacks". IEEE INFOCOMM, 2003.

[18] L. Spitzner: "Honeypots, Tracking Hackers", pages 239-240, Addison-Wesley 2002

[19] M. Roesch: Martin Roesch, Snort – Intrusion Detection and Prevention System, http://www.snort.org/, Sourcefire Inc.

[20] M. Roesch, "Snort—Lightweight Intrusion Detection for Networks*"*. In Proceedings of USENIX Systems Administration Conf. (LISA '99), Nov. 1999.

[21] National Institute of Standards and Technology (NIST)"Guidelines on firewalls and firewall policy" January 2002

[22] Nor Badrul Anuar, Omar Zakaria, and Chong Wei Yao University of Malaya, Kuala Lumpur MY Honeypot through Web: The emerging ofsecurity application integration.

[23]P. Dewan, P. Dasgupta, and V. Karamcheti. "Defending against Denial of Service attacks using Secure Name resolution", 2003

[24] [PPC97] Pfleeger, P. Charles. "Security in Computing". Prentice Hall PTR. Second Edition. p 3. 1997

[25]. Rafeeq Ur Rehman, 2003. Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID

[26] R. B. Blazek, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of denial of-service attacks via adaptive sequential and batch sequential change-point detection methods". In Proceedings of IEEE Systems, Man and Cybernetics Information Assurance Workshop, 2001

[27] Saleh Ibrahim Bakr Almotairi. Using honeypots to analyse Anomalous internet activities.

[28] Spitzner, Lance. Honeypots: Tracking Hackers. Addison-Wesley Professional,2002. An older book providing a comprehensive discussion of honeypots. Includes an in-depth treatment of 6 available honeypots

[29] Spitzner, L. Honeypots – Tracking Hackers. Addison-Wesley, 2003.ISBN 0-321-10895-7.

[30] Spitzner, L. Honeypots: Are they illegal? Security Focus, Infocus (June 2003). http://www.securityfocus.com/infocus/1703.

[31]Spitzner, L. Honeypots: Catching the insider threat. In proceedings of the 19th Annual Computer Security Applications Conference (ACSAC) (2003), pp.170–179.

[32] Stoll, C. (1998). Stalking the wiley hacker. Comunications of the ACM, 31(5), 484- 497

[33] The Honeynet Project: Research alliance, http://www.honeynet.org,non-profit Honeypot research organization, 1999

[34] The Honeynet Project. Know Your Enemy: Sebek - a kernel based data capture tool, Nov. 2003.http://www.honeynet.org/papers/ sebek.pdf

[35] Y. Xiang and W. Zhou. "Classifying DDoS packets in high-speed networks", In International Journal of Computer Science and Network Security, Vol. 6, No. 2B, February 2006

[36] [BaS].Bait and Switch Honeypot. http://baitnswitch. sourceforge.net/.

[37] [CVE] CVE – Common Vulnerabilities and Exposures. http://cve.mitre.org.

[38] [HNET] The HoneyNet Project. http://www.honeynet.org.

[39] [LOBSTER] LOBSTER-Large-scale Monitoring of Broadband Internet Infrastructures. http://www.ist-lobster.org/.

[40] [SEBEK] Sebek - A data capture tool. http://www.honeynet.org/tools/sebek.

[41] [SNORT] Snort - A network intrusion detection system. http://www.snort.org.

[42] A virtual Honeypot framework, Niels provos Google, Inc. niel@google.com