Global Journal of Computer Science and Technology
Network, Web & Security
Volume 12 Issue 16 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Network Security in Organizations Using Intrusion Detection System Based on Honeypots

By Mukta Rao & Dr Nipur

Gurukul Kangri Vishwavidyalaya, Haridwar, India

Abstract - The role of the Internet is increasing and many technical, commercial and business transactions are conducted by a multitude of users that use a set of specialized / sophisticated network applications. Today we face threats of the network which cause enormous damage to the community day by day to the Internet. In this context, the task of network monitoring and surveillance is of utmost relevance and honeypots are promising tools for information and understanding of "areas of interest" of the attackers, and the possible relationship between blackhat teams. In this situation, people are increasingly trying to prevent their network security using traditional mechanisms, including firewalls, Intrusion Detection System, etc. Among them honeypot is a versatile tool for a practitioner security, of course, they are tools that are intended to be attacked or interacted with other information about the attackers, their motives and tools. In this paper, we describe a comparative analysis of various IDS and their usefulness on various aspects. Two major categories of HoneyPot viz. low interaction honeypot and high-interaction honeypot have also been discussed in detail. In this paper, low-interaction honeypot is used as a traffic filter. Activities such as port scanning can be effectively detected by the weak interaction honeypot and stop there. Traffic that cannot be processed by the weak interaction honeypot is delivered over high-interaction honeypot. In this case, the weak interaction honeypot is used as a proxy for high-interaction honeypot then offer optimal realism.

Keywords : *intrusion detection system, honeypot, network security, ip address mapping.*

GJCST-E Classification : *C.2.0*

Strictly as per the compliance and regulations of:

Network Security in Organizations Using Intrusion Detection System Based on Honeypots

Mukta Rao [α] & Dr Nipur [α]

Abstract - The role of the Internet is increasing and many technical, commercial and business transactions are conducted by a multitude of users that use a set of specialized / sophisticated network applications. Today we face threats of the network which cause enormous damage to the community day by day to the Internet. In this context, the task of network monitoring and surveillance is of utmost relevance and honeypots are promising tools for information and understanding of "areas of interest" of the attackers, and the possible relationship between blackhat teams. In this situation, people are increasingly trying to prevent their network security using traditional mechanisms, including firewalls, Intrusion Detection System, etc. Among them honeypot is a versatile tool for a practitioner security, of course, they are tools that are intended to be attacked or interacted with other information about the attackers, their motives and tools. In this paper, we describe a comparative analysis of various IDS and their usefulness on various aspects. Two major categories of HoneyPot viz. low interaction honeypot and high-interaction honeypot have also been discussed in detail. In this paper, low-interaction honeypot is used as a traffic filter. Activities such as port scanning can be effectively detected by the weak interaction honeypot and stop there. Traffic that cannot be processed by the weak interaction honeypot is delivered over high-interaction honeypot. In this case, the weak interaction honeypot is used as a proxy for high-interaction honeypot then offer optimal realism.

IndexTerms : intrusion detection system, honeypot, network security, ip address mapping.

I. Introduction

No matter how well defended your chicken coop is, a sly fox still finds the hole and carry off the most fat chicken. All the holes do not shut up ... But you can try to catch a fox in a trap by placing it towards a tempting bait, and then - Broads! - Shot at point-blank from a gun. With computers - the same story. The software is vulnerable and prone to all attacks. The timely installation of patches, cuts off only the most stupid of the hacker attacks, and is not accustomed to think of his head. Professional burglars also involved in an independent search for new holes, patches do not stop.

This tactic is commonly used for the detection of computer attacks. Vulnerable server is installed in a conspicuous place in the network, safely isolated from all other nodes. This server tracks unauthorized access attempts in real-time transmission of IP-address of the attacker in the FSB or similar bodies. Even if the hacker hides behind clever words (proxy), IDS will still find it and coming out of the trap of an IDS is not an easy task[1-3].

The server, acting as a decoy, the hacker jargon is called a ***"honey pot,*** a network of such servers, respectively, ***honeynet.*** A more practical, but more restrictive, definition is given by pcmag.com: "A server that is configured to detect an intruder reflecting an actual production system. It appears as a ordinary server doing a job, but all data and transactions are false. Located inside or outside the firewall, the honeypot is used to learn about techniques intruders, and to identify vulnerabilities in the real system "[4]. The etymology of this name goes back to the English belief that if you leave a pot of honey, the bees will fly to it (the hackers). Honeypots [3-9] can be useful for two main purposes. The first relates to an important possible assistance in finding rootkits, Trojans and potential risks of the network. The second objective relates to the chances of obtaining information and understanding of "areas of interest 'of attackers and the possible relationship between “blackhat” teams. Despite the relevance of the problem, only a limited number of works devoted to illustrate the results obtained by inspection of the network are present in the literature [10]. The honeypot logs all actions and interactions with users. Since honeypots do not provide legitimate services, any activity is prohibited (and possibly malicious). In practice working of honeypots is being analogous to the use of wet cement to detect human intruders [11].

The value of a Honeypot is directly proportional to the quantity and type of information that we can achieve with success from it. In addition to the collection of information, a honeypot has the ability to distract opponents from the most important machines on a network, and can provide warning signs of a new type of attack and exploitation trends, and provides a thorough examination of adversaries during and after exploitation of a host. Another function that allows the capture of Honeypot is key entered by an opponent attempting to compromise the Honeypot - this provides a particularly interesting if an attacker uses the compromised host as an IRC chat server. Two levels of interaction Honeypots are described as low and high interaction.

The honeypot was the first publicly available as Deception ToolKit by Fred Cohen in 1998 which was "intended to reveal to attackers as if the system works

Author α : Gurukul Kangri Vishwavidyalaya, Haridwar, India.

DTK a large number of known vulnerabilities" [12]. More honeypots became both publicly and commercially available throughout the nineties. As began to proliferate from 2000, honeypots proved imperative to capture and analyze the worms. In 2004, virtual honeypots were introduced that allow you to run multiple honeypots on a single server. The paper laid the groundwork for the honeyd project and describes building virtual honeypots which meet help honeypots meet the need to monitor a large network address space [13]. A detailed history of honeypots can be found in [14] and [15].

To resist honeypot is extremely difficult. Externally, they are no different from the normal servers, but in reality are well-disguised Trap[16]. One false step and the hacker has nothing to help.

II. Inside the Pot

A typical honeypot is a huge hardware-software complex consisting of the following components: a node-bait, a network of sensor and the reservoir (storage media)[17].

There are currently two types of honeypots: a physical honeypot is a real machine with its IP address, and a virtual honeypot is simulated by some another machine that reads network traffic. Physical honeypots are often termed as high-interaction, since the system can be totally compromised and are expensive to install. For instance- if someone wants to implement physical honeypot for a given/specified range of IPs on the LAN, he should create a separate physical honeypot for each IP address. Virtual Honeypots are often labeled as low interaction due to the implementation of low cost maintenance features.

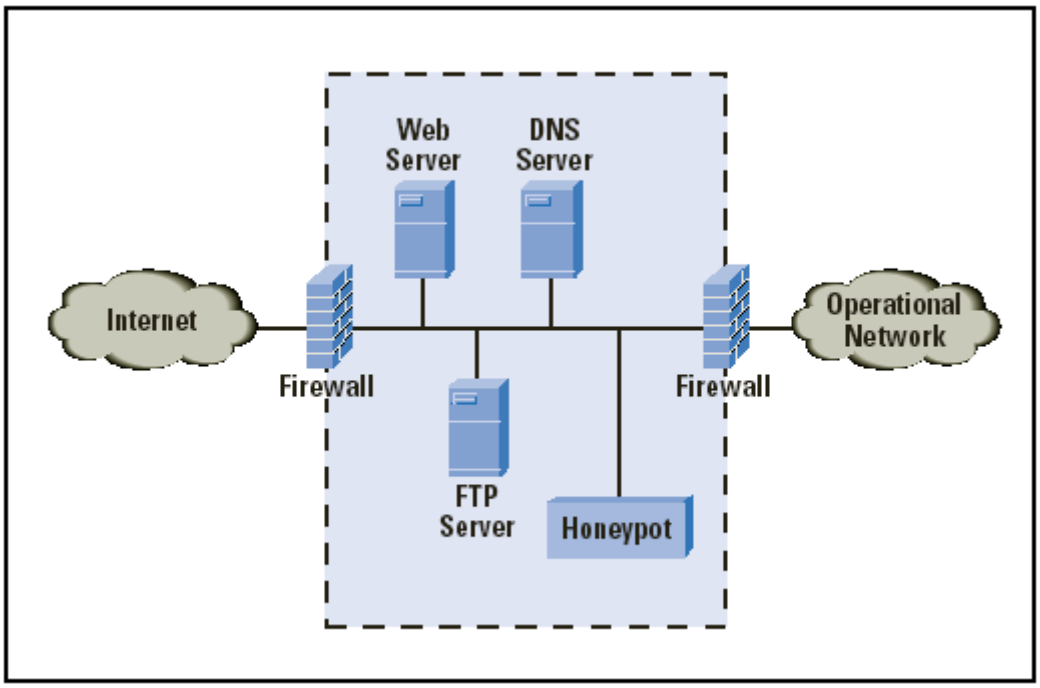

Figure 1 : Common Strategy for placement of Honeypot in Network

The other variety of HoneyPot i.e. virtual honeypot is able to simulate services of multiple operating systems together, and maintain a separate TCP / IP stack for each instance of a Honeypot on that one machine. An example of a virtual honeypot service is Honeyd, which simulates almost all TCP / IP interactions of target multiple operating systems, in order to fool TCP / IP stack fingerprinting tools like Nmap from xProbe. These Virtual honeypots are used more frequently than physical honeypots because they are low in cost as lesser computer systems are required, which eventually reduces maintenance costs. The other advantage is that they provide a greater variety of hosts to be observed.

Sensor network is most often realized on the basis of a UNIX-like operating systems, and for monitoring the information tcpdump utility is used or its analogs. Depending on network configuration, the sensor can be found as one of the nodes in this segment of the LAN and a router, located in front of the honeypot[18]. Sometimes the sensor network is combined directly with the bait as shown in figure 3. This greatly simplifies and reduces the cost of the system of honeypot, but it weakens the immune system (taking control of the lure, the attacker will quickly discover the sensor and make him safe). Placing the sensor in the broadcast segment gives it the greatest secrecy[19]. Network interface sensor may not have its own IP-

address, listening to traffic on Stealth-mode, which is achieved by physically cutting the wires transmitting the NIC[17].

Dumps collected from tcpdump and others are processed by different analyzers (eg, intrusion detection systems) in the first place that recognize the fact of the attack, and secondly, determining the IP-address of the offender. Intrusion related information ends up in the reservoir, which is the heart of the database. This is the most vulnerable spot of honeypot. Administrator must choose in advance a clear set of criteria to uniquely identify what actions are normal and what are not. Otherwise, the administrator will either be constantly jerking, shivering from each port scanning, or skip lightly modified version of the well-known attacks.

There is another problem. If the bait has no other traffic, except for the hacker (which is easy to determine the nature of change in the ID field in the IP-packet headers, details of which were described in an article by wagner [20]. Then the attacker shall immediately recognize the trap and will not attack it. If the lure is serving the users outside the network then direct analysis of the traffic dump becomes impossible and the attacker does not cost anything. Very effective bait is a database with credit card numbers or other confidential information (of course, spurious.) Any attempt to access this file, as well as the use of stolen information on the usages of debit cards is a clear indication of cracking. There are other ways to catch offenders, but they are somehow reduced to rigid patterns and, hence, in principle, unable to detect hackers with non-trivial way of thinking.

III. Preparation for the Attack

To start the hacker will need a reliable channel of communication from the authorities that could not trace him. Strictly speaking, all channels are monitored, however, the degree of security of each of them different. If you are in a broadcast network, the successful cloning of masking can restrict someone else's IP and MAC-address (of course, cloned vehicle at the time of the attack must be inactive). Provided that the network does not impose any additional equipment to determine the perpetrators, to identify the attacker is practically impossible, although there is a "but." If the machine is vulnerable to hackers, honeypot can quietly throw his computer "bug" with all the ensuing consequences. Many novice attackers are caught in the cookie, passed through a browser.

a) Tearing the veil of darkness

Before, to rush into battle, you must carefully examine his opponent: to reconstruct the network topology, determine the place of greatest congestion of opposing forces and, of course, to try to identify all the honeypots. The main weapon at this stage, the hacker will attack the port scanner that runs through "dumb" node and therefore concealing reliable IP-attacker. Clearly vulnerable server is better to discard, where a high probability of being caught with them are present.

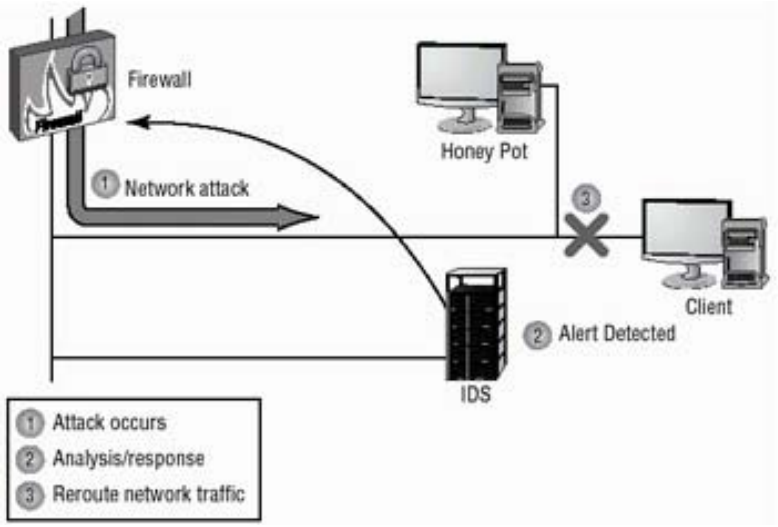

Figure 2 : Network Attack being defended

It is safest to attack workstations, corporate networks, bred for the firewall (if it really is). The probability of running into a honeypot is minimal. Unfortunately for the attacker, workstations contain a lot less holes than a server-based applications, and therefore to attack here, and nothing in particular.

IV. Attack on the Honeypot

Being by nature common network node, honeypot subject to various DoS-attacks [21]. The most vulnerable network sensor is obliged to listen to all the passing traffic. If an attacker can take it out of

the game, the fact that the invasion of the system at some time go unnoticed. Naturally, the attacked site should stay alive, otherwise no one will attack. We assume that the sensor to take all the packages, then sending a packet to a nonexistent node, or addressed to any other unnecessary node.

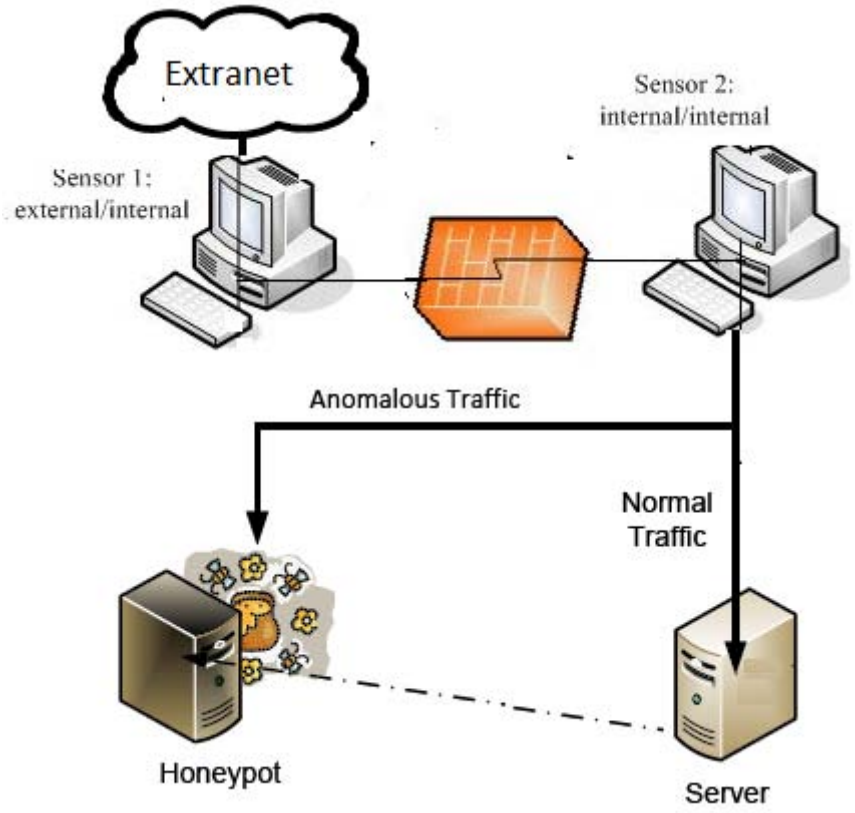

Figure 3 : Dual Sensor Based Arrangement For Anomaly Detection

Alternatively, one can flood the network of SYN-packets (look on the internet description of the SYN-attack) or call the ECHO - death (Storm ICMP packets directed at the victim with a few dozen high-end servers, which is achieved by spoofing IP-addresses - That is, sending echo requests from the victim's behalf)

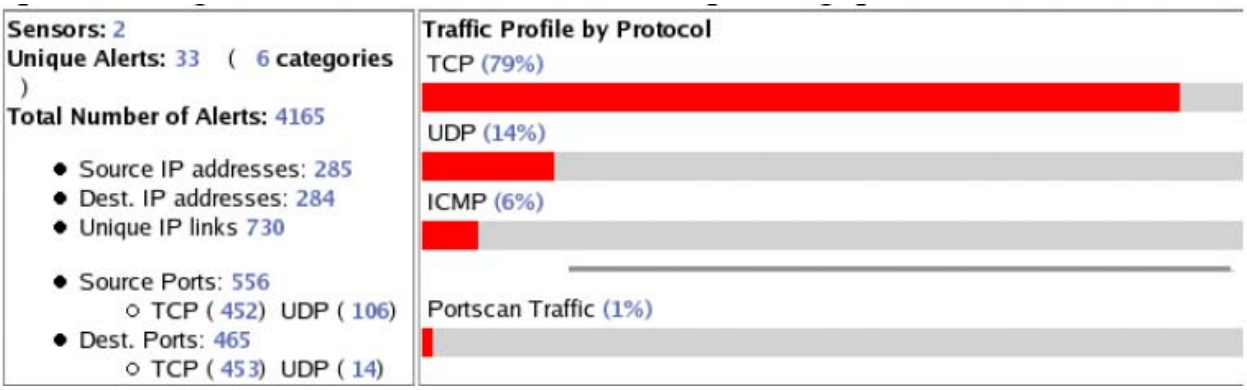

Figure 4 : Snap-Shot of IDS Capturing/Monitoring Network Interactions

The very same attack is best done over the protocols that are resistant to the interception of traffic, and support transparent encryption, blinding the sensor network. Most often used for this purpose SSH (Secure Shell), however, it limits the choice of attacking only the explicit support of its nodes, which negates the whole advantage of the encryption.

V. Drowned in Honey

If the attacked site had Honeypots installed, the attacker will not take any success (the vulnerable server silently "eats" an abandoned shell-code, continuing to work properly), or show empty resource does not contain almost anything interesting because Honeypots are dumb stations. In this situation the

main thing is not to panic and not to get confused. The first step is to get rid of compromising your Machine by disconnecting from network for some time. Next is to destroy everything related to the attack, software and related files, including temporary. Naturally, the above applies only to attacks on the really serious resources (government websites, banking institutions, etc.). Expect that after breaking someone's home page for you will take seriously, a bit naive. Reinstallation of web-pages will temporarily resolve the issue and then after the behavior/pattern of attack should be made learnt to the IDS. An example of such sequence of actions has been shown in Figure-5 below.

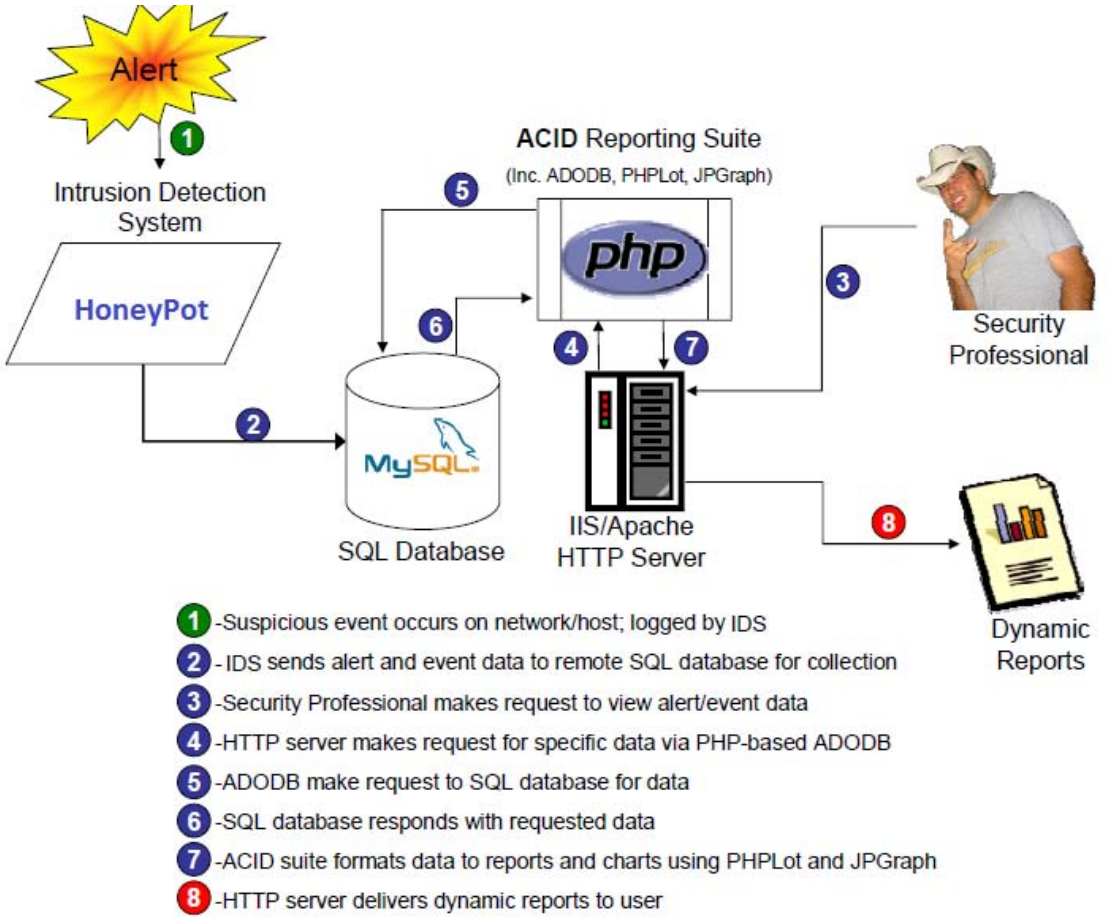

Figure 5 : The Attacker Thinks He Is Attacking The Vulnerable Service, In Fact, He Fells Into A Pot (With Honey)

VI. Conclusion

The strength of honeypot lies in their novelty and obscurity. Hackers are no adequate methods of confrontation with them, however one should not expect that such a balance of power will continue in the future. Architecture of honeypot is still ill-defined and vulnerable. Even today, nothing is impossible for experienced attacker (to bypass honeypots), tomorrow every teenager shall be capable of bypassing such IDS.

Honeypots are positioned to become an essential tool for defending the corporate enterprise from hacker attacks, it is a way to spy on your enemies, it might even be a form of concealment. Hackers could be misled into thinking they have achieved a corporate network, when in reality they are just kicking around a honey pot, while the real network remains safe and sound. Honeypots have gained increased prominence in the strategy to protect against intrusions overall business. Security experts do not recommend that these systems replace existing technologies for intrusion detection security, they see the honeypots as a complementary technology to protect against network and host intrusion.

The advantages that honeypots provide to intrusion protection strategies are difficult to ignore. In time, as security officials understand the benefits, honeypots will become an essential ingredient in a operation of enterprise-class security. We believe that although honeypots have legal problems now, they do provide useful information regarding the security of a network. It is important that new legal policies be formulated to promote and support research in this area. This will help solve the current challenges and

make possible to use honeypots to benefit the Internet community at large.

References Références Referencias

1. Sobh TS. Wire and wireless intrusion detection system: classification, good characteristics and state of art. Computer Standard Interface 2006; 28(6):670–694.
2. Münz G, Li S, Carle G. Traffic anomaly detection using K-means clustering. Proceedings of GI-IGT Workshop, MMBnet, Hamburg, September 2007; 13–14.
3. Sourour M, Adel B, Tarek A. Ensuring security in depth based on heterogeneous networks security technologies. International Journal of Information Security 2009; 8(4):233–246.
4. honeypot Definition - PC Magazine. pcmag.com. 24 March 2009. http://www.pcmag.com/encyclopedia_term/ 0,2542,t=honeypot&i=44335,00.asp PC Magazine's encyclopedia entry for honeypot
5. Intrusion Detection, Honeypots, and incident Handling Resources. Available from: www.honeypots.net.
6. Spitzner L. Honeypots: Tracking Hackers. Addison Wesley: Boston, MA, 2002. ISBN 0-321-1095-7.
7. Development of the Honeyd Virtual Honeypot. Available from: www.honeyd.org.
8. Project Honey Pot. Available from: www.projecthoneypot.org.
9. The Honeynet Project. Available from: www.honeynet.org/project.
10. Verwoerd, Theuns and Ray Hunt. "Intrusion detection techniques and approaches." Computer Communications 15 September 2002: 1356-1365. Aavailable at http://sureserv.com/technic/datum_detail.php?id=468
11. Talabis, Ryan. "Honeypots 101: A Honeypot By Any Other Name." 2007. Available http://www.philippinehoneynet.org/index2.php?option=com_docman&task=doc_view&gid=1&Itemid=29
12. Cohen, Fred. "The Deception ToolKit." The Risks Digest 9 March 1998.The announcement can be found at http://catless.ncl.ac.uk/Risks/19.62. html#subj11
13. Provos, Niels. "A Virtual Honeypot Framework." In Proceedings of the 13th USENIX Security Symposium. 2004. 1-14. . The paper is located at http://www.citi.umich.edu/techreports/reports/citi-tr-03-1.pdf.
14. Spitzner, Lance. Honeypots: Tracking Hackers. Addison-Wesley Professional, 2002.
15. Talabis, Ryan. "Honeypots 101: A Brief History of Honeypots." 2007. Available at http://www.philippinehoneynet.org/index2.php?option=com_docman&task=doc_view&gid=2&Itemid=29
16. Anagnostakis, K. G., et al. "Detecting targeted attacks using shadow honeypots." Proceedings of the 14th conference on USENIX Security Symposium. ACM, 2005. 129-144.
17. He, Xing-YuLam, Kwok-Yan, et al. "Real-Time Emulation of Intrusion Victim in HoneyFarm." Content Computing. Springer Berlin / Heidelberg, 2004. 143-154.
18. Mukherjee, B., L. Heberlein and K. Levitt. "Network Intrusion Detection." IEEE Network May/Jun 1994: 26-41
19. honeynet Project. "Know Your Enemy: Honeynets." 24 March 2008. http://old.honeynet.org/papers/honeynet/
20. Wagner, David and Paolo Soto. "Mimicry Attacks on Host-Based Intrusion Detection Systems." Proceedings of the 9th ACM conference on Computer and communications security. ACM, 2002. 255 - 264. Available at http://www.scs.carleton.ca/~soma/id-2007w/readings/wagner-mimicry.pdf
21. Hussain, Alefiya, John Heidemann and Christos Papadopoulos. "A framework for classifying denial of service attacks." Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. ACM, 2003. 99 - 110.