# Honeypot in Network Security: A Survey

Abhishek Mairh
Department of Computer Sc. Engg.
International Institute of Information and Technology, Bhubaneswar
Bhubaneswar, Odisha, India
iam.mairh@gmail.com

Debabrat Barik
Department of Computer Sc. Engg.
International Institute of Information and Technology, Bhubaneswar
Bhubaneswar, Odisha, India
davidbarik@gmail.com

Kanchan Verma
Department of Computer Sc. Engg.
International Institute of Information and Technology, Bhubaneswar
Bhubaneswar, Odisha, India
kanchan.2405@gmail.com

Debasish Jena
Department of Computer Sc. Engg.
International Institute of Information and Technology, Bhubaneswar
Bhubaneswar, Odisha, India
debasish@iiit-bh.ac.in

## ABSTRACT

In this paper we review the recent advances in honeypot. Some notable proposals and there analysis have been discussed. The aspects of using honeypot in education and in hybrid environment with IDS have been explained. In this paper we also defines the use of signature technique in honeypot for traffic analysis. In the end we summarizes all these aspects.

## Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General- *Security and protection.*

## General Terms

Security and Performance.

## Keywords

Honeypot, Honeycomb, Honeynet, IDS, Load balancer and Honeywall.

## 1. INTRODUCTION

Due to rapid growth of internet technology, people easily retrieve their information and quickly transfer messages. However, due to such a swift internet growth, if we don't concurrently attach value to basic network security, it will lead hackers to control the network by using some malicious code, system vulnerabilities and program weakness. Then the attack, devastation and stealing, tampering of information by the hackers may lead to great damages and loss of data. Traditionally we use IDS (Intrusion Detection System) and Firewall System in network to prevent our damages and to provide network defense against the intruders. But IDS and firewall cannot avail all the subsequent information to know the intruders attack and reduce loss caused by attacks.

Infected or malicious code: According to *Pei-Sheng Huang* et al [4] "Any malicious, unauthorized access, the program is not in line with expectations". Such as computer virus, worm, Trojan or backdoor software, dangerous program (Risk ware) threats are malicious and malevolent codes. An attacker might use them to employ illegal activities, invasive of privacy, and cause individuals or businesses major financial loss. By understanding these infected codes properly and knowing the target sites of attack in network, we can provide support to security officials to detect and analyze infected code to guarantee network security.

This information is collected by honeypot and offered to other gears that do not have this information. If we integrate this information together with an IDS and Firewall it may lead to reduction of false positive or false negative. We have various open source command line interface honeypots which shows various complications during the implementation. Those collected information has some restrictions at various levels and these information's are not complete. Now we are presenting various problem statements and flaws which can be used for further research in this area to provide better security. Thus security officials can understand the information and can perform deep analysis to realize the patterns of attacks and risks attached with it.

In 2002, Spitzner [1] defined honeypot as "a security resource whose value lies in being probed, attacked or compromised". Further, honeypots doesn't provide any solution to any problem, nor they "fix" anything, they are just a tool. It depends upon the user how and in which way they use this tool either for good or for bad.

A honeypot is a computer system which is placed to get compromised to get the information about the blackhats. A honeypot is like any other computer system which contains directories, drives in it as real computer systems but, its motive is

*ICCCS'11*, February 12–14, 2011, Rourkela, Odisha, India.

very specific and different. The use of real systems in this manner is famous among the white and blackhats only. One can never eliminate risk, but security helps reduce risk to an organization and protect its valuable resources [1].

The rest of the paper ia as follows. In the section 2, different types of honeypots have been discussed. Application and development of honeypots have been explained in section 3. Finally, section 4 describes the concluding remarks.

## 2. Types of Honeypot

In 2007, Marty Roesch suggested [1], there are the two types of honeypots are Research and Production. Further, according to Mokube I. and Adams M.[2] we can group honeypots according to their aims and level of interaction.

## 2.1 Research Honeypots

As the name suggest, these types of honeypots are solely used in the research areas. The main aim here is to get maximum information about the blackhats by giving them full access to penetrate the security system and infiltrate it. By allowing such an access to blackhats, it's easy to know about the tools used and other related information about them**.**

## 2.2 Production Honeypots

This type of honeypot is used to protect company from malicious activities done by blackhats. This honeypot is placed under the production network to increase the overall security of the company.

Spitzner L. and Bruce Schneier[1] model helps us to understand the honeypots. They divide the security issues into groups as: prevention, detection and response.

### *2.2.1 Prevention*

In this type, as company's point of view they are solely concerned about their security and not much interested to know about blackhats. So, they put firewall, use strong passwords, even try encryption techniques, digital signatures, digital certificates and provide well known security services. They do these just to keep away blackhats from their valuable resources.

### *2.2.2 Detection*

Considering that the prevention doesn't work well, the other solution to overcome attacks is Intrusion Detection System. This technology will help us know whether the system has been compromised or not, but, it will not prevent hackers from attacking the system.

### *2.2.3 Response*

We are unable to prevent the blackhats to infiltrate our system by the above two approaches. As our system has been compromised, in order to take down the attackers we have to backtrack them by the use of log files. Every system makes a log file, keeps information about everything happening in the system in it. By studying and analyzing the log file we are able to find information about blackhats, the IP address they used, their network address from which they accessed and the available ports from which they accessed our system. This technique is known as forensic investigation.

Based upon the level of interaction that we provide to the blackhats to access our systems, we can categorize honeypots as: low interaction and high interaction honeypots.

### *2.2.4 Low Interaction Honeypots*

In the low interaction honeypot, the interaction of the blackhats with the system is limited and is for small amount of time thus the blackhats can not intrude the system. This type of honeypot is made keeping in my mind that we are securing ourselves from the intruders. But we get very little information about blackhats. So, this approach is widely used in companies where they are concerned about protecting their system from the outer world.

### *2.2.5 High interaction honeypots*

In high interaction honeypot, the main emphasis is to get the maximum information about the blackhats allowing them to access the whole system or even tamper it. This is solely research oriented, for those who want to discover new techniques used by the blackhats.

## 2.3 Advantages

According to Mokube I. and Adams M.[2] some of the advantages of honeypot are :

1. Honeypots are placed just to get information about the attacks as they are been recorded in the log files.
2. People who target the honeypot are the blackhats as they only know about it not the common people.
3. Honeypots are not bulky as they are placed just to capture a specific pattern of data i.e. malicious traffic.
4. Honeypots provide us the information about the newly generated attacks, newly defined technologies.
5. Honeypots are simple and easy to configure. They do not have complex algorithms.
6. As honeypots captures the malicious traffic, they also capture the new tools used by the blackhats.
7. Honeypot detects few false positive and false negative data also.

A honeypot can be placed in a network, with firewall, before firewall and after firewall. We are considering these places because these are the most frequent places from where the blackhats accesses the system and we can trap them to get maximum information about them. Our aim is to get maximum information about them by compromising our research data, so that they may not infiltrate the data again in the future. We are here to know the tools used by the attackers, their technologies so that we can update our network security against these tools.

## 3. APPLICATIONS AND DEPLOYMENT OF HONEYPOT

This section discusses the application domains of honeypot. Here we discusses its application in educational areas, internet, with IDS and its implementation.

## 3.1 Honeypots in Educational Resource

Jeremiah K. Jones & Gordon W. Romney [3] discussed the aspects of using the honeynets in educational areas. A lab has been established at Brigham Young University for network security purpose for undergraduate and graduate students [6] called ITSecLab. They use this lab for tracing the malicious traffic in the network. This lab was designed solely for the purpose of experiments on network security by students. In addition to this lab they have implemented a honeypot in their lab to get in touch with blackhats and explore its uses as an educational tool. The lab is designed as an isolated "Sandbox" [7] in order to keep away the malicious activities from lab. The honeypot is implemented at Brigham Young University keeping in mind the certain benefits such as it notifies about the new threats, securing the lab at higher level, learning the network and security basics and closely identifies the flaws. One more aspect comes into play when implementing the honeypot, the legal issues that are most important part in implementation because if the honeypot gets compromised and is used as zombie then the owner has to suffer the loss.

## 3.2 Virtual Honeynet in Teaching and Research

Another way of implementing the honeypot in educational areas can be done by implementing real or virtual honeynet for better understanding the flaws in network security. Depending upon the use and its advantages, real or virtual or both can be used in educational institutions. Research on honeynet in a Brigham Young University IT security curriculum [3] states the advantages of implementing the real and virtual honeynets. In order to predict which scheme is better either the real honeynet or the virtual honeynet, comparisons have been done taking under consideration setup, deployment, maintenance, data collection, and data analysis defined and considered as in [4]. In the real honeynets, all the connections have to be considered very generously in order to remove any possibility of fault. If any connection is made wrong, then the whole system gets thrashed. Whereas in the case of virtual honeynet, it is implemented in a virtual environment using VMware in which each network is divided into subnets [4]. It should be considered properly that different ports must be assigned to different virtual networks. To establish honeynet so that all the traffic directly comes to honeypot first, three steps must be considered in order to make our host machine safe from intruders. First the host machine must be fully secured and all the safety measures must be taken into account. Then, the connection establishment between virtual and host machine must be such that any debase that may occur in the honeypot must not affect the host machine. Finally, all the traffic must be directed towards honeypot not the host machine. If the intruder connects with the host machine then the information cannot be obtained. The techniques done to implement these three steps are discussed in [3].The machines used in it for testing the possibilities contains Honeywall based on Roo 1.89 and Debian based Linux honey pot[5]. In order to compare the effectiveness of two different systems, consider the case of data analysis. After running both systems for same time the Real honeynet has 33MB dump file whereas the Virtual honeynet has 64MB dump file but, after filtering the Real honeynet has 7.39MB file and Virtual honeynet has only 4.71MB file[3]. Both the systems are effective in collecting information analysis. While performing analysis the clocks of the two systems are set same, but after hours of data analysis it is found that virtual honeynet system has clock time left behind as compared with the Real honeynet's clock. In order to make them same again, the clock must be set. Considering the attacks on these systems 21 were found on Virtual honeynet and 13 were found on Real honeynet [3]. In maintenance issues, the real honeynet needs the equipments to be purchased and becomes costlier whereas the virtual honeynet has no such issues. Another important issue in this is to keep a good honeypot on the system. If the virtual honeynet is compromised then it can be recovered by just reset and running whereas in the real honeynet, two steps can be taken first, re-image each machine as needed and another, complete hard drive backup must be taken. If this approach doesn't seem to work in real honeynet then, the administrator has to physically access the honeynet and remove its hard drive. The Real Honey Net is a valuable teaching tool if the goal is simply to set up a Honey Net for the experience of building one, and taking it down shortly thereafter due to the relative simplicity of the setup process. However, if the goal is to set up a long term and ongoing research on Honey Net, then Virtual Honey Net would be the better choice due to the ease of maintenance [3].

## 3.3 Honeypot with IDS

An Intrusion Detection System (IDS) [7] discriminates between the traffic coming from various clients and from the attackers, in an effort to simultaneously ease the problems of throughput, latency and security of the network. After that we can present the results of a sequence of load and their response time in the terms of performance and scalability tests, and suggest various types of potential uses for such a system.

In IDS we may use two common type detection level known as *Misuse detection* and *Anomaly detection* [8]. In misuse detection the IDS analyzes all the various kinds of information that have collected and match it to large database of attack signatures. In anomaly detection the administrator makes a baseline, or we may say a normal network traffic load, collapse, protocol and packet size. It monitors network [9] and compares it to those baseline.

IDS can be further categorizes into *Network based* and *Host based* [8]. In network based IDS, the individual packets are analyzed whereas in host based IDS all the activities of the host are monitored. Honeypots can either be host and/or network based, but generally they are not network based as all interface operations are typically performed over a network connection. Its key utility is that it simplifies the Intrusion Detection problem of separating "anomalous" from "normal". Thus any activity on a Honeypot can be immediately defined as abnormal.

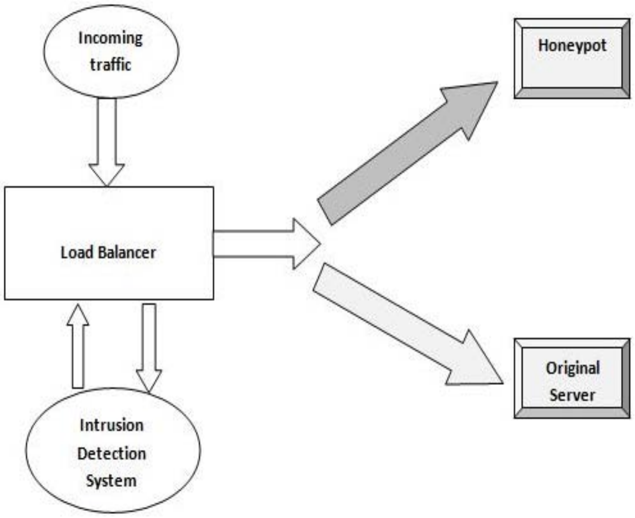

**Figure 1 Flow of packets through IDS in Honeypot[8]**

From the above diagram each components play a specific role in implementation of honeypot with IDS within a network. Initially load balancer **[8] [11]** receives the virtual IP address, and checks whether the packet containing the request has been fragmented, and then it is reassembled. Then load balancer opens a TCP connection to the IDS Process, and sends the content of the packet (less the headers) over that connection.IDS checks the content of packets against its database and returns the Boolean value of that to load balancer through the same TCP connection. After receiving the result, the load balancer closes the TCP connection. If the result from the IDS was "true" (Indicating an attack) the packet is forwarded to the Honeypot. Otherwise, a server is selected from the active server pool in a round-robin fashion and the packet is forwarded to the server.

## 3.4 Honeypot in Internet

The honeypot project measures the actual computer attacks on the Internet [10] [13].According to their most recent results, a random computer is scanned dozens of times a day. A honeypot is a program that takes the form of attractive services, an entire OS, or even an entire network, but is in actual a tightly scaled compartment built to attract an attacker, effectively shunting an intruder safely from production systems for convert analysis. Here honeypot monitors each logs files and every action of an attackers.

If any kind of attack comes, a honeypot provides two things: first the information needed about an attack to develop quick and suitable response in real-time and second is the time required to implement that response. During analysis of an attack after it has occurred, a honeypot can be used in analyzing an attacker's activity (basic information) to develop long-term strategic line of action, including tagging which counter measures/patches should be implemented. For other response operations, honeypots can be useful for detecting insider misuse [8] where it **is** known exactly what to do to attract the "attackers".

Honeypots can be implemented to show its effectiveness by means of three different ways: *to deceive*, *to intimidate*, or *to provide reconnaissance* [12].To deceive, a honeypot should provide various reasonable responses such that intruder does not suspect it is a trap. To intimidate, a honeypot increases the intruder level of risk trough advertisement in unauthorized deception port. For reconnaissance, a honeypot allows vital attack signature information to other security tools such as IDS [7] and Firewalls to decrease the number of false alarms.

There are various societal issues attached with honeypot, such as the issue of entrapment [12], if an attacker is intentionally lured to a honeypot, there must be no implicit permission to access the system. Viewing files and intercepting communication such as chat or Email on a honeypot is related to privacy law. The attacker files are not protected since there is no legitimate account or privileges. While there is case law about the loss of the right of privacy in storing files on a stolen computer or files on a compromised computer without owner's authorization, there is little or no case law on interception of communications relayed through a compromised computer [14].

## 3.5 Network Security through "Hybrid Honeypot"

According to Lance Spitner, author of Honeypots, Tracking Hackers [16],"*A honeypot is security resource whose value lies in being probed, attacked, or compromised*"

A honeypot is a system that is made and set up in order to be hacked. It can be used in a different scenario as intrusion detection facility [15], defense or reaction mechanism. Moreover, it can be deployed in order to consume the resources of the intruders or distract them from the precious targets and slow them down that wastes their time on the honeypot instead of attacking production systems or servers.

Here again we divide the honeypots into two categories according to their level of interaction, *low level interaction* and *high level interaction* [15]. The level of interaction can be defined as the maximum range of attack possibilities that a honeypot allows an intruder to have. In high-level interaction honeypot, attacker interact with real operating systems, all the services and programs and this type of interaction can be used to observe the attackers performance, their tools, motivation and explored vulnerabilities. This type of high-level interaction honeypot can be deploying inside a virtual machine using various virtualization software such as VMware, Qemu and Xen. Example of high-level interaction honeypot is honeynet. It is a network of multiple systems. Honeynet [17] can collect deep information about intruders, such as their keystrokes when they compromise with a system, their chatting sessions with fellow blackhats, or the various tools they use to explore and develop susceptible systems. On low-level interaction honeypot [18], there is no operating system that an intruder can operate on. All the tools are installed in order to emulate OS and other services. And they all work together with the intruders and infected code. This will reduce the risk radically. This type of honeypot has a few chance of being compromised. These are production honeypots. Typical use of low-level interaction honeypot includes; port scans identification, generation of attack signatures, trend analysis and malware collection.

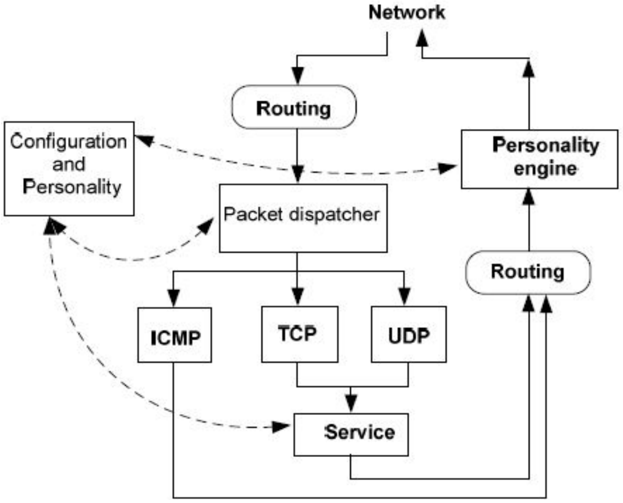

**Fig. 2: A simplified view of the honeyd architecture[24]**

Here we deploy the concept of hybrid honeypot, by using both low-level interaction honeypot and high-level interaction honeypot. Also low-level interaction honeypot is more secure than high-level interaction honeypot because of processing real service; it needs the skill to provide a good level of practicality [19]. However, high-level interaction honeypot provides the best possible level of realism but it has more risk. To offer an extensible infrastructure for honeypot installation and growth of detection mechanisms on top, good features of both types have to be combined. Here in this type of system, low-level interaction honeypot act as lightweight proxy [19]. As we require high-level interaction honeypot to process all traffic destined to block IP address space.

## 3.6 Deployment of Intrusion Detection Signatures using Honeycomb

This phenomenon generally deals with generation of signatures [20]. At present, generating signatures are tiresome work, manual process that needs detailed knowledge of each software function that is supposed to be detained. Simplistic signatures tend to generate large numbers of false positives, too specific ones cause false negatives. For the same reason the concept of *Honeycomb* [20] a system that generates signature for infected traffic automatically, is used. Here pattern-detection techniques and packet header are used for conformance tests on traffic captured by honeypots.

The purpose discussed about the attack signatures is to explain the characteristic elements of attacks. Right now we don't have any such standard for defining these signatures. As a consequence, different systems offer signature languages of varying expressiveness. A good signature must be *narrow* enough to confine precisely the characteristic aspects of exploit it attempts to address; at the same time, it should be *flexible* enough to capture variations of the attack. Failure in one way or the other leads to either large amounts of false positives or false negatives.

In this way system supports signatures only for the Bro [21] and Snort [22] NIDSs. Bro has a controlling signature language that permits the use of regular expressions, involvement of traffic going in directions, and encoding of attacks that comprise numerous stages. Snort's signature language is currently not as communicative as Bro's. So we include Snort here because of its present reputation and huge signature warehouse. System used here is an extension of honeyd [23] a well-liked low-level interaction open-source honeypot. Honeyd simulates hosts with personage networking personalities. It interrupts traffic sent to non-existent hosts and uses the imitated systems to respond to this traffic. Each host's personality can be personally configured in terms of OS type and running network services.

### *3.6.1 Architecture of Honeycomb*

Here two new concepts are added to honeyd: a plug-in infrastructure and event callback hooks[20].The plug-in infrastructure allows us to write extensions that stay logically separated from the honeyd code-base, during the event callback, hooks offers a mechanism to join together the plug-in into the activities inside the honeypot. Presently, hooks allow a plug-in to be notified when packets are received and sent, when data is passed to and received from the subsystems and to receive updates about honeyd's connection state. Honeycomb is implemented as a honeyd plug-in. As shown below.

### *3.6.2 Signature Algorithm*

The viewpoint behind this approach is to keep the system free of any knowledge specific to certain application layer protocols. Each received packet reasons Honeycomb to begin the same succession of activities:

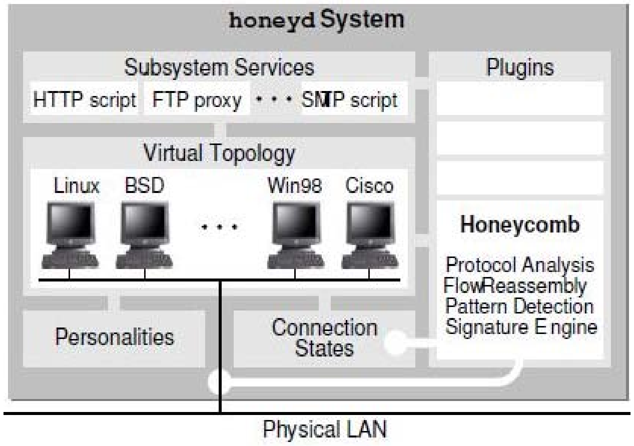

**Figure 3 Architecture of honeycomb with typical honeyd setup[20]**

- If there is any existing connection state for the new packet, that state is updated, otherwise new state is created.
- If the packet is outbound, processing stops here.
- Honeycomb performs protocol analysis at the network and transport layer.
- For each stored connection:

1. Honeycomb performs header comparison in order to detect matching IP networks, initial TCP sequence numbers, etc.
2. If the connections have the same destination port, Honeycomb attempts pattern detection on the exchanged messages.

- If no useful signature was created in the previous step, processing stops. Otherwise, the signature is used to augment the *signature pool.*

## 4. CONCLUSION

In this paper we present the concept of honeypots and its application. We have implemented and deployed honeypots in different technical ways in a network, to provide various security aspects. We have also discussed various types of honeypots and its use with different functionality aspects.

## 5. REFERENCES

[1] Spitzner, L. 2002. Honeypots: Tracking Hackers. 1st ed. Boston, MA, USA: Addison Wesley.

[2] Mokube, I. & Adams M., 2007. Honeypots: Concepts, Approaches, and Challenges. ACMSE 2007, March 23-24, 2007, Winston-Salem, North Carolina, USA, pp.321-325.

[3] Aaron Lanoy and Gordon W. Romney, Senior Member, IEEE [2006] A Virtual Honey Net as a Teaching Resource

[4] F. A. Shuja. (2005, November). Virtual Honeynet: Deploying Honeywall using VMware, Pakistan Honeynet Project [Online], Available: http://www.honeynet.org.pk/honeywall/roo/

[5] (2005, August). Know Your Enemy: Honeywall CDROM Roo 3rd Generation Technology, Honeynet Project & Research Alliance, [Online] Available: http://www.honeynet.org, Last Modified: 17 August, 2005.

[6] G. Romney, et al., "A Teaching Prototype for Educating IT Security Engineers in Emerging Environments," Presented at the IEEE ITHET 2004 Conference in Istanbul, Turkey, June 2, 2004. Published in IEEE Xplore.

[7] Cliaord Stoll. Stalking the Wily Hacker. Communications of the ACM. pp 484- 497. 1988.

[8] Ram Kumar Singh & Prof. T. Ramanujam. Intrusion Detection System Using Advanced Honeypots, 2009

[9] Martin Roesch, Snort- Lightweight Intrusion Detection for Networks, Proceedings of LISA'99: 13th System Administration Conference, Seattle, Washington USA, 2005

[10] The Honeynet Project. Know Your Enemy: Honeynets (May 2005) http://www.honeynet .org/papers/honeynet/.

[11] Honeynet Research Alliance. Project Honeynet Website. Retrieved May 16th 2003 from the World Wide Web: http://project.honey.org

[12] Brian Scottberg et-al. Internet Honeypot: Protection or Entrapment, 2002.

[13] The Honeynet Project, Know Your Enemy: Honeynets, April 2001.

[14] The Honeypot Project, Know Your Enemy: Revealing the Security tools, tactic, and motives of Blackhats community.2002.

[15] Hybrid Honeypot System for Network Security by Kyi Lin Lin Kyaw, 2008.

[16] Spitzer, Lance. Honeypots, Tracking Hackers. Pdf version. Addison Wesely, 2002.

[17] Honeynet project, The. (2007a). Know your enemy: Honeynets. Retrieved on 7 October 2007 from http;//www.Honeynet.org/papers/honeynet/index.html

[18] Research infrastructures action, Sixth framework programme, D1.1: Honeypot Node Architecture, page 7-24.

[19] Research infrastructures action, Sixth framework programme, D1.4: Architecture Integration, page 36.

[20] Honeycomb. Creating Intrusion Detection Signatures Using Honeypots Christian Kreibich, Jon Crowcroft.

[21] V. Paxson, .Bro: A System for Detecting Network Intruders in Real-Time. *Computer Networks (Amsterdam, Netherlands: 1999)*, vol. 31, no. 23-24, pp. 2435.2463, 1998. [Online]. Available: http://citeseer.nj.nec.com/article/paxson98bro.html

[22] M. Roesch, .Snort: Lightweight Intrusion Detection for Networks. In *Proceedings of the 13th Conference on Systems Administration*, 1999, pp. 229.238.

[23] N. Provos, .Honeyd - A Virtual Honeypot Daemon, in *10th DFN-CERT Workshop, Hamburg, Germany*, February 2003.

[24] Provos N. and Holz T. 2007, Virtual Honeypots : From Botnets Tracking to Instrusion Detection, Addision Wesley Professional