Linnæus University
School of Computer Science, Physics and Mathematics

Degree Project

Honeypots in Network Security

Deniz Akkaya – Fabien Thalgott

2010-06-29
Subject: Network Security
Level: Bachelor
Course code: 2DV00E

Abstract

Day by day, more and more people are using internet all over the world. It is becoming a part of everyone’s life. People are checking their e-mails, surfing over internet, purchasing goods, playing online games, paying bills on the internet etc. However, while performing all these things, how many people know about security? Do they know the risk of being attacked, infecting by malicious software? Even some of the malicious software are spreading over network to create more threats by users. How many users are aware of that their computer may be used as zombie computers to target other victim systems? As technology is growing rapidly, newer attacks are appearing. Security is a key point to get over all these problems. In this thesis, we will make a real life scenario, using honeypots.

Honeypot is a well designed system that attracts hackers into it. By luring the hacker into the system, it is possible to monitor the processes that are started and running on the system by hacker. In other words, honeypot is a trap machine which looks like a real system in order to attract the attacker. The aim of the honeypot is analyzing, understanding, watching and tracking hacker’s behaviours in order to create more secure systems. Honeypot is great way to improve network security administrators’ knowledge and learn how to get information from a victim system using forensic tools. Honeypot is also very useful for future threats to keep track of new technology attacks.

Keywords: Honeypot, hacking, security, forensic analysis of honeypots, network.

Acknowledgments

We would like to thank Ola Flygt to guide us and support our work, and supply the necessary equipment for our thesis process.

Table of content

1. INTRODUCTION

First of all, we would like to build a honeypot on a machine. One of us will try to find security flaws that exist on the system. After defining all those, we will try to attack the system. Once the hacker will be able to have access into the system, one of us will have the role of forensic examiner. Using useful forensic investigation tools, he will try to find out the changes that occurred on the victim system by looking at the tracks left behind the hacker. Furthermore, we will go deeper into the subject thinking about its problems bringing to the system. It will be helpful for network security administrators to create more and more secure systems and be aware of the threats.

1.1 Problem Description

As we are successful to make system that is interesting enough for hackers to attack, they will try to gain access by using security flaws on the system. By tracing the hacker, we are not sure if we will be the one who has the control. Therefore we do not know if honeypots are secure or not. Does the hacker know that it is a real system or a honeypot? Is he aware of how a great tool it is for investigators to acquire information about security flaws in the system? What does he gain from hacking it? It is a big problem if it is possible to reach other real systems using honeypot features and seize them, because the rest of the system will be compromised. We are not sure if the hacker will continue hacking even if he knows that it is a honeypot or not. Knowing all these issues does not make our investigation efficient. We will try to find answers and solutions to all these questions and think about what can be done to make honeypots more secure and make sure that the hacker will not be able to go further than hacking the honeypot. We will have two perspectives which are a forensic examiner and a hacker. We will use variety of hacking tools and forensic examiner tools to have very accurate results.

1.2 Motivation

First of all, we are very interested in this subject field of study. So, our motivation for this thesis is to understand how security systems are working and how an organization can be protected and being aware of the risks of security flaws in the system. We will learn how a system is working and how it can be developed. Once we have the results, we will examine the output with forensic science tools. While trying all these, we will come across some problems and we will try to solve it. At the same time we will have experience on creating and managing this kind of systems for the future. If we see similar problems in a network, we will be able to handle the system and recover the loss. Therefore, we will have a knowledge including both security problems examining and forensic science information gathering.

1.3 Goals

We will find answers to all the questions that we stated in problem description part. Are the honeypots secure? Does the hacker know that it is a trap system? If the hacker realizes that it is a trap system does he continue attacking to it? What does he gain from attacking it? Is it possible for the hacker to reach other systems and compromise them? Our perspective is to solve the problems related to security, how a honeypot can be deployed, and the amount of information that we can get. We will look into the restrictions honeypot implementation mainly in EU and USA including which laws exist, how far a network security administrator can go to obtain information and track the hacker. We will explain and come up with some discussions regarding what should be done and what should not be done with respect to the laws. We will have some opinions and suggestions based on our work. While we will be looking for answers for security problems, we will also evaluate and think about the limits of the experiment.

1.4 Restrictions

We will use variety of commercial and noncommercial software throughout the thesis process with appropriate hardware. We will not look into hardware and software properties deeply. We will try to have objective point of view with the comparison of other software.

1.5 Report Structure

In our thesis report, we will start with the chapter about honeypots and their aims which consist of the description of honeypots, their types, history, advantages and disadvantages, level of interactions and wi-fi honeypots. Next chapter will be looking at the security problems closer. We will state the legal issues and uses of honeypots as well as their risks. Forth chapter will be practical implementation which includes the work we accomplished. Chapter five will cover other honeypot products which exist in the market. Forensic part of the honeypots will be explained at chapter six. We will write the interview about honeypots that we did with lawyer Pehr Jern at chapter seven. We will show the results at chapter eight and finish with conclusion at chapter nine.

2. HONEYPOTS AND THEIR AIMS

In this chapter we will explain what a honeypot is and its purpose. We will also present its history and see its advantages and disadvantages. We will finish with wireless honeypots.

2.1 What is a honeypot ?

First of all, a honeypot is a computer system. There are files, directories in it just like a real computer. However, the aim of the computer is to attract hackers to fall into it to watch and follow their behavior. So we can define it as a fake system which looks like a real system. They are different than other security systems since they are not only finding one solution to a particular problem, but also they are eligible to apply variety of security problems and finding several approaches for them. For example, they can be used to log malicious activities in a compromised system, they can be also used to learn new threats for users and creating ideas how to get rid of those problems.

According to Mokube,I. & Adams M.(2007:p.322) we can divide honeypots according to their aims and level of interactions. If we look at the aims of the honeypots, we can see that there are two types of honeypots, which are research honeypots, and production honeypots.

2.2 Research honeypots

Research honeypots are mostly used by military, research and government organizations. They are capturing a huge amount of information. Their aim is to discover new threats and learn more about the Blackhat motives and techniques. The objective is to learn how to protect a system better, they do not bring any direct value to the security of an organization.

2.3 Production honeypots

Production honeypots are used to protect the company from attacks, they are implemented inside the production network to improve the overall security. They are capturing a limited amount of information, mostly low interaction honeypots are used. Thus, security administrator watches the hacker’s movements carefully and tries to lower the risks that may come from it towards the company. At this point, we will try to discuss and find out the risks of using production honeypots. Because while testing the security of the systems existing in an organization, unexpected actions may happen such as misusing other systems using honeypot features. If the network administrator is not aware of this problem, they put organization in a big trouble.

Spitzner L.(2002) claims that it is easier to break the honeypot phases into groups and refers that Bruce Schneier model is good for understanding the honeypots. He groups the security issues into several steps, which are prevention, detection and response.

2.3.1 Prevention

Prevention is the first thing to consider in our security model. As a definition, it means to prevent the hackers to hack the system. So, we will try not to allow them to access the system. There are many ways to do this in security. One can use firewall to control the network traffic and put some rules to block or allow it. Using authentication methods, digital certificates or having strong passwords are the most common and well-known security prevention techniques. There are also encryption algorithms that encrypt data. It is a good way to use it since it encrypts the messages and make them impossible to read.

The relation between using prevention and honeypot can be explained as following. If the hacker understands the company he is trying to hack is using honeypots and they are aware of today's security problems, it will make them think about it. It will be confusing and scary for a hacker. Even if a company uses the methods that we discussed in the first paragraph in order to stay secure, it is still good to have honeypot in an organization since security issues are concerned and handled professionally. As the security is very significant, it is always good to be conscious. There is no tolerance when there is a problem, it can give a lot of damage to any company. Because every company has private and important data, there is a need to protect the data from intruders.

2.3.2 Detection

Detection is the act of detecting any malicious activity in the system. We are assuming that prevention did not work so one way or another, a hacker compromised the system. There are some ways for detecting those attacks. The well-known detection solution is Network Intrusion Detection Systems. This technology will help users to know if the network is compromised, but it will not prevent hackers from attacking the system. For companies, such detection systems are expensive. At this point, honeypots are valuable to monitor the activity.

2.3.3 Response

Last component of Schneier's model is response. At this stage, we are sure that we had been attacked and we will have response to it. This is where our forensic investigation begins. When a hacker compromises the system, he leaves traces behind. With the appropriate tools, we can handle the data in a way that we can have some clues about what happened to the system. It is possible to watch log files and try to investigate what happened. More about forensic tools and how to get valuable information from it will be discussed later.

2.4 History of Honeypots

In this part, we will give the history of honeypots so far according to Lance Spitzner (2002):

1990-1991: It is the first time that honeypot studies released by Clifford Stoll (The Cuckoo's Egg) and Bill Cheswick (An Evening With Berferd).

1997: Deception Toolkit version 0.1 was introducted by Fred Cohen. After Clifford Stoll (The Cuckoo's Egg) and Bill Cheswick (An Evening With Berferd) , Deception Toolkit gave an idea of first honeypot structure.

1998: First commercial honeypot was released which is known as CyberCop Sting.

1998: BackOfficer Friendly honeypot was introduced. It was free and easy to configure. It is working under Windows operating system. Most of the people tried this software and the concept of honeypot became more and more known among people.

1999: After BackOfficer Friendly, people were more into this new technology. Honeynet project started at this year. Also, Know Your Enemy papers were also released. Thanks to these releases, people understood the aim of the honeypots more.

2000-2001: Honeypots started to be used for capturing malicious software from internet and being aware of new threats. Companies began to use honeypots in their systems to improve security and see the malicious traffic.
2002: Honeypot concept became popular and honeypots improved their functionalities, so they became more useful and interesting for both researchers and companies.

2.5 Advantages of honeypots

There are many security solutions available in the market. Anyone can browse the variety of choices through internet and find the most suitable solution for their needs. Here are the reasons why we should choose honeypots according to Mokube I. and Adams M. (2007):

Honeypots can capture attacks and give information about the attack type and if needed, thanks to the logs, it is possible to see additional information about the attack.

New attacks can be seen and new security solutions can be created by looking at them.

More examinations can be obtained by looking at the type of the malicious behaviors. It helps to understand more attacks that may happen.

Honeypots are not bulky in terms of capturing data. They are only dealing with the incoming malicious traffic. Therefore, the information that has been caught is not as much as the whole traffic. Focusing only on the malicious traffic makes the investigation far easier. Therefore, this makes honeypots very useful.

For the only malicious traffic, there is no need for huge data storage. There is no need for new technology to maintain. Any computer can be used as a honeypot system. Thus, it does not cost additional budget to create such a system.

They are simple to understand, to configure and to install. They do not have complex algorithms. There is no need for updating or changing some things.

As honeypots can capture anything malicious, it can also capture new tools for detecting attacks too. It gives more ideas and deepness of the subject proving that it is possible to discover different point of views and apply them for our security solutions.

2.6 Disadvantages of honeypots

As there are several important advantages of using honeypots, there are also some disadvantages of them as well. We are continuing with Mokube I. & Adams M. (2007)'s studies:

We can only capture data when the hacker is attacking the system actively. If he does not attack the system, it is not possible to catch information. If there is an attack occuring in another system, our honeypot will not be able to identify it. So, attacks not towards our honeypot system may damage other systems and cause big problems.

There is fingerprinting disadvantage of honeypots. It is easy for an experienced hacker to understand if he is attacking a honeypot system or a real system. Fingerprinting allows us to distinguish between these two. It is a not a wanted result of our experiment.

The honeypot may be used as a zombie to reach other systems and compromise them. This can be very dangerous.

2.7 Levels of interactions in honeypots

As we categorized honeypots according to their aims, now it is time to look into more details in levels of interactions way. Level of interaction stands for how much the hacker will be able to interact with the system. More amounts of data we would like to gather require more level of interaction. More level of interaction brings more risks into the network security as well. Based on the needs and the purpose of the experiment that one would like to examine, there are three categories of levels of interactions in honeypots. They are called low interaction, medium interaction and high interaction. Let us have a look at each of them and compare them one by one according to Mokube, I. & Adams M.(2007:p.322-323).

With low interaction honeypots , one can get the least amount of data compared to other honeypot systems. They are limited, so the risk that was taken from intruder is not big either proportionally. First of all, there is no operating system to deal with. They can be used to identify new worms or viruses and analyzing the traffic that is going on through network. Low level of interaction honeypots are easy to configure and understand. In our thesis, firstly we will understand the logic of this category and test how efficient they are. Therefore, we will start our experiment using the most common low level of interaction honeypot, which is Honeyd. Its last version (1.5c) has been released on 2007 and we will use it. Details of Honeyd 1.5c will be explained under practical implementation chapter.

Medium interaction honeypots are more advanced than low interaction honeypots. Still, operating system does not exist. But this time, more information and more complicated attacks from the hacker can be obtained. As it is more advanced, it has more security holes so that hacker can access the system. Mwcollect, honeytrap and Nepenthes are some of the medium interaction honeypots that are used today.

High interaction honeypots are the most advanced honeypots. Unlike low interaction and medium interaction honeypots, there is an operating system. As a consequence, the hacker can perform anything. Proportionally, more data can be captured from the hacker's activities. However, it is the most risky one when it comes to security as it provides such an access to the hacker that he does not have any restrictions. These kind of honeypots are very time consuming and difficult to maintain. Honeywall is a good example of a high interaction honeypot. We will come back to these security issues covering all these kind of honeypots and discuss and state the exact security problems and come up with some ideas to improve security thanks to our laboratory work.

2.8 Wireless Honeypots

In this part, we looked into a different kind of honeypot systems which are wireless honeypots. The goal of deploying wireless honeypots is to capture behaviors of our system in a wireless area and obtain some information and statistics. IEEE 802.11 technology is covered, and also other technologies are possible such as bluetooth. We used Maggi F. & Zanero S., (2008) and Siles R., (2007) ideas on this part.

2.8.1 Why Wi-Fi Honeypots ?

This Wi-Fi structure can be obtained with some access points, wired network and some open-to-attack computers. Wi-Fi honeypots are used to capture unauthorized traffic, and tries to answer questions if it is possible to catch wardriving and hackers which are trying to compromise wireless networks.

2.8.2 Wireless Honeypot History

First idea of wireless honeypots was released by Kevin Poulsen in 2002. During his experiments, he realized that networks are not secure and protected. Intruders are trying to monitor your system, eavesdropping, hacking your system through your wireless network. Therefore, The Wireless Information Security Experiment started the work in 2002 in Washington USA. After that, the leader of this foundation Rob Lee continued the experiments and tried to answer questions related to wireless hacking, and understand the hackers' ideas and tools, especially the logic behind it.

Late 2002, Tenebris organization in Canada did the monitoring for malicious activities, and understood that there was a huge malicious traffic going on through the network. They did the experiment using wireless honeypot. After that other experiments followed this idea in 2003, 2004 and 2005.All the experiments proved that there had been always threats on wireless networks at that time. Moreover, those kinds of threats still exist today.

In 2004, Laurent Oudot published "Wireless Honeypot Countermeasures" article about wireless honeypots. This article explains the wireless honeypots in detail, its aim and restrictions.

In 2006, a new project was born named MAP Project.MAP was symbolizing the triple suggestions for wireless honeypots : Measure, Analyze and Protect. In this project, hacker was allowed to compromise the system and after that the project members were capturing the malicious activities on the wireless honeypot. However, this project was not improved and it could not answer further questions about wireless honeypots and intruders.

In 2007, "The Hive" project started to answer questions at University of Florida in USA. Project members tried to discover the attacks' identity, and get additional information about it.

2.8.3 Honeyspot

Honeyspot is the well known wireless honeypot project supported by Spanish Honeynet Project. The term comes from honeypot and hotspot. Basically, honeyspot was created to watch the hacker and his attacks towards the wireless network. Thus, the traffic that is through the honeyspot is considered as malicious. However, like any other honeypot structures, professional hackers may understand that it is not a real system. So, honeyspot should look as real as possible for the best results. Honeyspot team would like to know the attack type, intruder's ideas, tools, logic, and his approaches. It is very beneficial to get as many information as possible to identify the attack and help to understand any other further attacks in the future. From all these information, honeyspot can answer the questions about the security flaws in WEP wireless connections and attacks targeted to it. IP address spoofing, web session hacking, MAC address spoofing can be identified. It can also answer the special approaches to hack wireless clients. Thanks to all these information, more secure systems can be created.

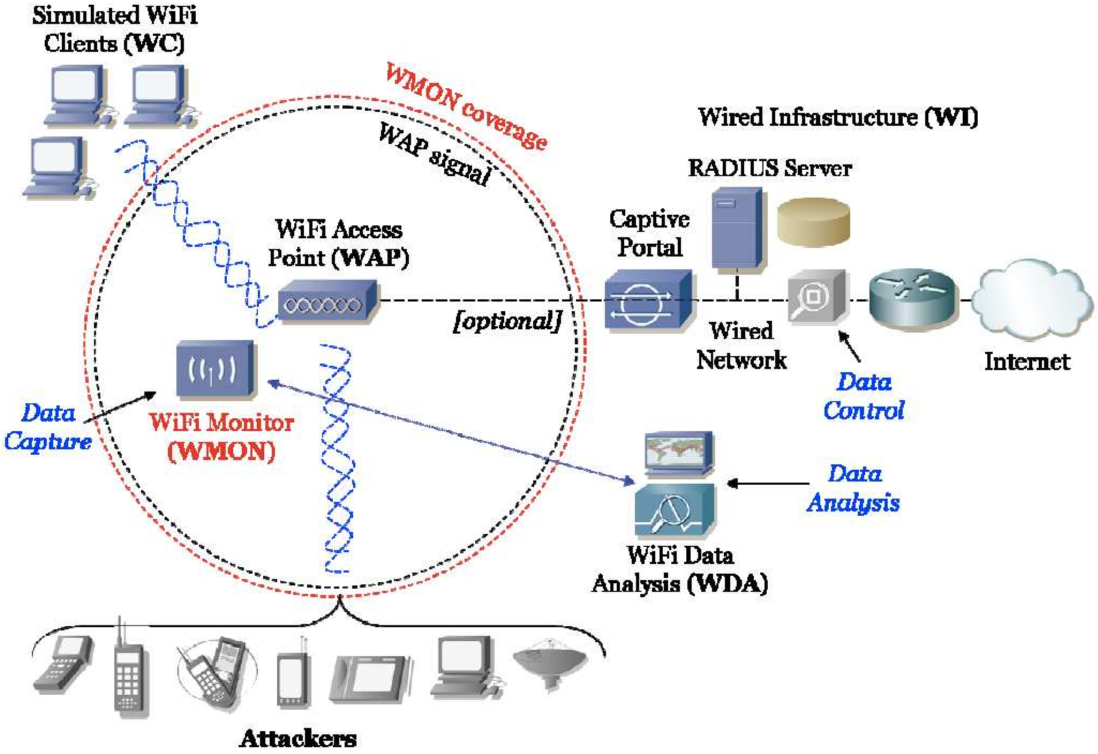


Figure 2.1. Honeyspot Architecture from Siles R., (2007)

The figure 2.1 is showing the architecture used in honeyspot.WAP which is in the middle is the wireless access point. It gives the wireless networks to the users for internet connection. Attacker can connect to it.

WC (Wireless Client) are the devices that are able to connect to the honeyspot network. The purpose of this is to create a traffic that is flooding through the network. It is to show the attacker that there is a traffic. The real traffic makes sense for the attacker as it looks like a real system. Furthermore, attacker can attack on this stage by using his monitoring tools.

WMON is wireless monitor module. This module captures the traffic in order to have the network traffic information. It helps to understand the attacks, so this module is quite significant at this point.

WDA is wireless data analysis module. This module works with WMON as a team. As WMON is supposed to capture the traffic, there must be a module which is responsible for examining it. Therefore, WMON has the records and saves them in order to send them to WDA for obtaining the information.

WI module is wired structure and it is up to the person to put it in the structure or not. If you wish to create a wired network in your structure, it is also possible. So, WI module gives you a different aspect to your structure.

3. LOOKING AT THE SECURITY PROBLEMS CLOSER

At this chapter, we will cover the security problems in honeypots and related issues. We will state the today's situations and emphasize the solutions based on our experiments throughout the thesis.

3.1 Legal issues with honeypots

3.1.1 Using honeypots are illegal or not?

While deploying and start using a honeypot, there are some legal issues that a person should know about. Every country has different laws regarding to honeypot usage and information capturing. These regulations are related to data security, collection of data and finally how to use honeypots. All these different laws are based on the quality of the data that a honeypot can capture and a person who is deploying it. In here, the type of the data and its contents are significant. It is not easy to say that if using honeypots are illegal or not. As we stated before, it depends on the intention and the usage of the information that has been collected. Therefore, there are several steps to think about before doing this job.

There are also several questions and approaches that you should ask yourself during the experiment. If it is for a company that you are deploying a honeypot rather than a homemade honeypot to use it at home, then as a network administrator you have other responsibilities as well. First thing is to think about the country laws regarding to these, and then company laws. Maybe the country is allowing you to experiment some things but what happens if there is a restriction on it in your company? Before taking any serious action, you should ask those questions to the responsible people to make sure that you are doing something without violating the laws of your country. We will look into three main legal issues now, which are privacy, entrapment and civil liability.

3.1.1.1 Privacy

Let us start with privacy issue. As the type of data we are gathering is important, privacy and data leads us to confidentiality term in network security. Our example is being a network administrator in a company. Does he have a right to collect information from other employees in the company? Accordingly, it is the same logic with the hacker. Does the hacker have a right to do so? If we combine both of these situations, then we come up with these: Does honeypot have a right to collect information from the hacker and his/her friends?

Privacy is relative here. As there are several levels of interactions honeypots, the information that is gained is also relative. Higher level of interaction means more security risks but more data we can capture. The question is how much data can we take from the honeypot while not breaking the laws at the same time?

Lance Spitzner (2001) is referring some useful points that is from the Department of Justice, mostly Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations. Here they are:

· "The people breaking into these systems are NOT AUTHORIZED to use them, and if they place any files on them (when they have no legitimate accounts or use privileges), they have given up their privacy rights to that data by placing it on the honeypot.

· By using honeypots for communication, they have given up their right to privacy in that communication. Honeypots generally do not provide public accounts; therefore, they are not a service provider and are not bound by privacy requirements designed for service providers.

· Most organizations are not law enforcement, nor do they act under the control of law enforcement, so they are not bound by the evidence collection restrictions otherwise placed on law enforcement and their agents. Think about it, a honeypot is collecting the same information, in the same technical manner, as many of your other security devices are (system logs, IDS sensors, etc.)."

Based on US constitutions, there are four main laws considering privacy on data communications. These four will be discussed in details under different laws in different countries US regulations part.

3.1.1.2 Entrapment

The definition of entrapment is "a law-enforcement officer" or government agent's inducement of a person to commit a crime, by means of fraud or undue persuasion, in an attempt to later bring a criminal prosecution against that person."(Spitzner L.(2002) taken from Campbell H.B.) Therefore, honeypot can be entrapment issue.

This issue is debatable as the concept of honeypots are new, there are not certain issues decided yet. There are also other aspects concerning entrapment issue of honeypots. According to Lance Spitzner's The Value of Honeypots, Part Two: Honeypot Solutions and Legal Issues article, honeypots cannot be entrapment issue. Here are his three reasons why entrapment is not an issue for honeypots:

1) "Honeypots do not induce or persuade anyone, they are most often production systems, or emulate production systems.

2) Attackers find and attack honeypots based on their own initiative.

3) Most administrators are not law enforcement. They are not using honeypots to collect evidence and prosecute. Normally they are used as a means to detect, and possibly learn about, attacks."

3.1.1.3 Civil liability

Civil liability is another legal problem in honeypots. The explanation can be defined with an example considering a hacked system. When a system is hacked, it can be used to hack and misuse other systems. Misused honeypot may bring problems as it is being used by hacker to reach other systems to hack as well. It should be noted that there is nothing to do with federal or law in this issue. When that kind of problem occurs, you should consult state which means you should talk about this problem with legal counsel.

3.1.2 Different laws in different countries

3.1.2.1 US regulations

According to Lance Spitzner (2002) there are four different laws in United States. They are U.S Constitution (Fourth Amendment), The Federal Wiretap Act (can be referred as Title III), The Pen Register / Trap and Trace Statue (Pen/Trap), The Electronic Communication Privacy Act (ECPA). The last three ones are taking care of privacy of communication.

First of all, fourth amendment is meaningful with the following situations according to Lance Spitzner (2002). "Where the honeypot is run by government, the search and seizure provisions of the Fourth Amendment may apply." And the second situation is "where a private actor is assisting a government actor, that private actor may be deemed an extension, or agent of the government actor for Fourth Amendment purposes."

Wiretap Act and Pen/Trap are working together to investigate the traffic that is happening at that time. They are obtaining the data from the source such as the traffic information, IP address and packets that has been sent through the network, port numbers and so on.

3.1.2.2 EU regulations

In EU regulations, it is not very easy to find laws that exist like US. Lorf Steyn is explaining with these words how it is working in United Kingdom according to the case of R versus Latif.

"The court has the discretion: it has to perform a balancing exercise.....the judge must weigh in the balance the public interest in insuring that those that are charged of grave crimes should be tried and the competing public interest in not conveying the impression that court will adopt the approach that the end justifies the means."

Lakhani A.D states that in European Union and United Kingdom, problems with privacy are handled with two major laws. They are Directive 97/66/EC and Regulation of Investigatory powers act, 2000. However second law only exists in United Kingdom.

Directive 97/66/EC is a law which handles the confidentiality of communication including the network and other telecommunications for public systems.

Regulation of Investigatory powers act, 2000 is the version of USA's federal wiretap statue. They both handle privacy of communication interceptions.

3.1.2.3 French regulations

In France there are no clear laws about honeypots. However Barel M., (2004) wrote an article discussing their legal issues. This article is only the opinion of the writer so it should not be taken as a law. This paper offers guidelines to ensure we are not entering in a legal grey area while implementing a honeypot. The first thing to do when the decision to use a honeypot has been taken is to clearly identify its aim. Will the honeypot be used to observe, defend or prosecute? Each of these goals lead to its own legal issues but the common factor is privacy. In France, an independent administrative authority, the CNIL (Commission nationale de l'informatique et des libertés), has the mission to ensure that data privacy law is applied to the collection, storage, and use of personal data. This organism has to be contacted when implementing a honeypot to ensure that the administrator is not breaking any law.

3.2 Security risks

As we started our experiment with low interaction honeypot Honeyd, we discovered its security risks. It is somewhat easy to detect Honeyd as a trap system. Without configuring our own honeypot with our settings, it is even easier to detect Honeyd. It is because Honeyd is dropping the connections until it cannot deal with them anymore as Maggi F. and Zanero S. (2008) stating. Honeyd is terminating the connection when SYN package is not good also. Using this information, any tool which can help to check connections through honeypot can help hackers to understand it is a honeypot system. Intruder will just look at the output of the tool and see dropped connections.

We explained that low interaction honeypots are emulating services of an operating system. Thus, as a hacker, you can come up with some conclusions by using this very basic information. As the services are emulated, low interaction honeypot cannot handle complicated services inside. Trying to break the system using this technique may work efficiently. The key point is to look for information through network, because in low level of interaction honeypots we are dealing with the network stack. We also know that low interaction honeypots are using the system's resources that they are on it. If we remove the resources that low interaction honeypot is interacting with, we can come up with this solution: Latency. We can do simple ping test and the reply will be longer than before. System will be heavier and will hardly return answers to us. So using these approaches we can detect the honeypots that we experimented which are Honeyd and Nepenthes. (Provos N., Holz T. (2007))

Other interesting findings are claimed from Alata E., Nicomette V., Kaâniche M., Dacier M., Herrb M., (2006).When we leave our low interaction honeypots for several days open, we can come up with some conclusions. First of all, we should be careful about the requests that are coming to our honeypot. Our honeypot should generate believable answers as replies in order to fool the attacker. But, in a low level of interaction honeypot, SSH server is up and running and from port 22, there are no replies that are sent to the attacker. This makes our situation weird since we know that if there are no appropriate answers, our honeypot does not look like a real system. Thus, it is not secure.

3.3 Things to consider while configuring a honeypot

We have to consider some points before experimenting with honeypots. There is a checklist that Amit D.Lakhani prepared for this purpose as follows.

First of all, we should experiment with honeypots in safe place closeby to the production systems. It is the best solution for entrapment, because you can configure and show banners both of the computers. To make everything clear and secure in your mind, you can consult law enforcement officer and ask him to make the investigation instead of you. As he is more experienced in that field, it will be very helpful throughout the process. As privacy issues are always important, so it is good to ask and learn from local solicitor before creating a serious honeypot that you will use for an organization. According to where you live, laws may change, so it is good to be aware of the laws and deploying honeypot according to them without any problem. Finally, for any kind of problems that may occur, it is always good to have some evidences. Therefore, make sure that you are logging everything that is happening on the system and record and save them in a safe environment. It is also very significant to know which users are using which systems and make sure that who are logging into specific systems. Protected data should be hidden from unauthorized users. Honeypot should be protected; it should not be open and easily accessible as it may create some security problems. Some malicious people use it for bad purposes such as installing unauthorized programs, cracks and they may download illegal information, documents, software or some hacking tools.

4. PRACTICAL IMPLEMENTATION

In this chapter we are presenting our practical work. We are starting with low interaction honeypot and then continue on a middle level of interaction to finally conclude with a high level of interaction.

4.1 Starting to honeypots

In chapter 4 practical implementation, we will speak about our laboratory experiments and why we chose to deploy those specific products throughout the thesis. We will explain how it is working and come up with some results related to our findings. We started with Honeyd as low level interaction honeypot and then we will move on medium level interaction honeypots. Every honeypot has specific and different attitudes. We will explain them one by one.

4.2 Starting with low level interaction honeypots : Honeyd

Low interaction honeypots are emulating the services of a real operating system. We started with deploying Honeyd. It is the most well known low level interaction honeypot. We thought it is a good starting point; it is easy to configure and understand its logic. More explanation can be found as the following.

We started our work with low level interaction honeypots and then we planned to move to more complicated and advanced level honeypots later as we are gaining experience step by step. Therefore, as a starting point, we worked on Honeyd.

Honeyd is developed by Niels Provos from University of Michigan and used mainly as a production honeypot. Honeyd is an open source solution and designed for Unix systems. Like the other low level interaction honeypots, there is no operating system installed in Honeyd. They are just some services running on it. It is configurable, so anyone can create their own services and decide which ports to open and listen as well. As hacker will not find any real computer with real operating system, the key point here is to configure a virtualized network stack. Honeyd basically captures TCP traffic that hacker is generating. On Honeyd , we configured a template which looks like a real system with Windows XP operating system, and IP address. Thus, when the hacker establishes the connection with Honeyd, Honeyd generates fake messages and return them to the hacker to fool the hacker. Honeyd is able to create many fake IP addresses and simultaneously run them for hackers trying to attack the machine. Unlike other low interaction honeypots, Honeyd can also handle several different operating systems at the same time. There are two other major advantages to use Honeyd. First of all, it can capture the connection on any port. This utility makes detection of the network traffic easier and better. Second advantage of it is that being able to change services. Thanks to these different and interesting sides of Honeyd, we decided to work on it. Following figure clearly explains the process of honeypot. As it is seen processes are designed to create some returns which are created by personality engine to make it look good and logical according to our template for Honeyd.

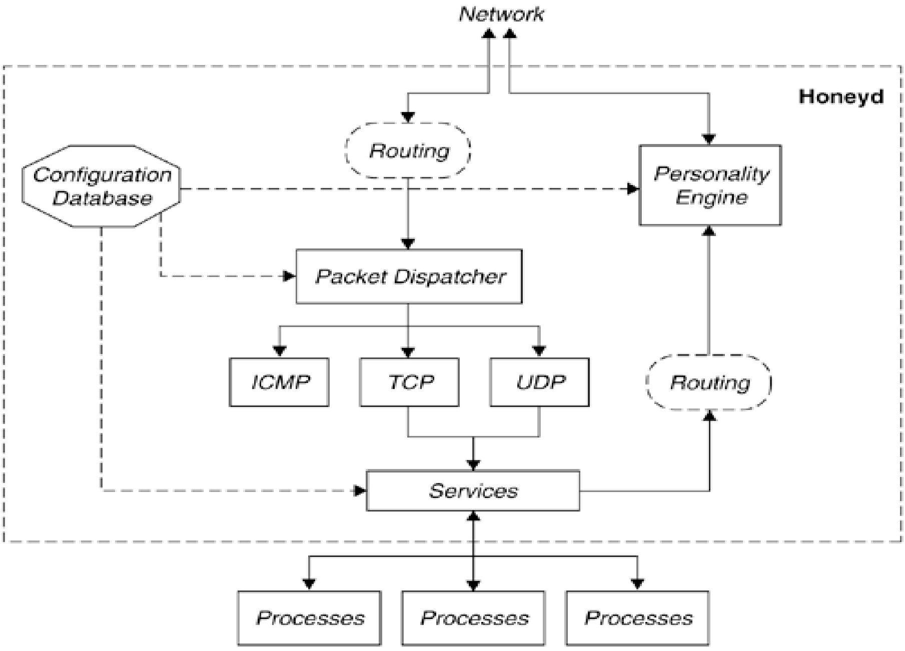


Figure 4.1 Honeyd structure from *Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Provos* N., Holz T. (2007)

Honeyd does not bind sockets, it is virtualizing the network stack. It is advantageous since it is possible to keep an eye on any different big address space. The entire network stack is being implemented by Honeyd which name is personality engine. This personality engine is receiving packets and changes them in a way that it looks like real implementation of a real system on TCP/IP stack. Honeyd copies the behaviors of real hosts. Figure 4.1 is explaining how Honeyd system is working. The time that the packet arrives to the system, it is being sent to packet dispatcher. The packet dispatcher is sending it to the services that are related to your template which is your Honeyd configuration. Personality engine and configuration engine are working together to decide the protocol for transferring still according to the configuration. Personality engine is a kind of emulator for transport and link layer. For application protocols for emulating it there are three options that Honeyd is allowing which are Honeyd service scripts, python services and subsystems. With Honeyd service scripts, one can create an executable for the configuration of Honeyd, and can adapt it to the system on a specific port. Script will work when the connection arrives. Python service is available in Honeyd. It gives chance to the network administrator to create python modules. It works through the emulation process of protocol interaction. And lastly, a subsystem means that one can use external Unix applications inside of Honeyd (Maggi F. and Zanero S. (2008)).

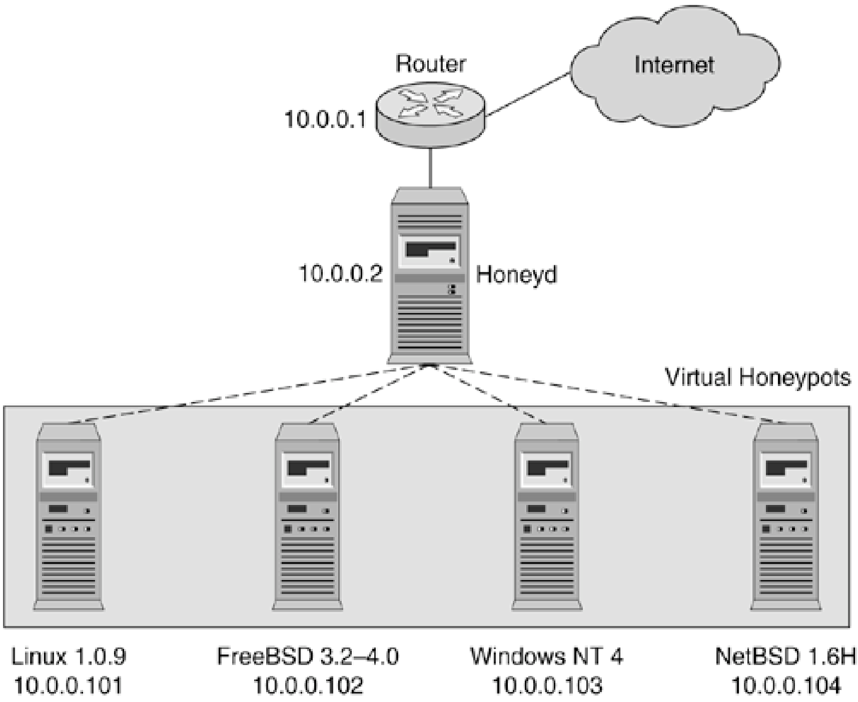


Figure 4.2 Honeyd virtual honeypots from *Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Provos* N., Holz T. (2007)

As said before, Honeyd is simulating operating systems and services as it is shown from the figure 4.2. Honeyd itself is just a daemon running on a linux machine. The most important configuration file is situated in /etc/honeypots/Honeyd.conf. It is in this file we create the virtual network and we configure the templates on which we want to assign an IP address. A template is a virtual machine, we can set which port are open, which operating system is running, its uptime and more. Each port can be set to be open with a script running on it to simulate the service. Once the template as been set up, it is possible to bind it to several IP addresses, this way we can create a full network which would look like real for the attacker.

More concretely, when a hacker will try to scan a range of IP addresses, Honeyd will reply for the IP on which a template is bound, for the empty IP addresses, no reply will be sent.

Honeyd can sometimes interfere with a DHCP server if one is running on the same network. The reason is that Honeyd is simulating machines using real addresses.

Honeyd is the most popular low interaction honeypot but its problem is its age. The project is opensource but part of it is outdated and nobody seems to upgrade it. On the other hand hacker tools are evolving so identifying this honeypot is not hard. Honeyd is using an old version on Nmap fingerprint to create fake virtual operating systems so by using a newer version of Nmap, the fake operating systems will not be recognized and Nmap will detect that there is a problem.

Another limitation of Honeyd is the scripts binded to the different ports. With a basic scan it is possible to find which ports are open but as soon as the attacker tries to actually connect on a port, he will realize the service is fake. For example the script used for a Web server, by connecting it using telnet, the server should send back replies but nothing is happening. As a result the hacker can understand quickly that there is something wrong with the target and will abort his attack.

4.3 Continuing with medium level of interaction honeypots : Nepenthes

After deploying Honeyd, we understood how it is working and examined its problems. Now, we are moving on to medium level of interaction honeypots. Medium level of interaction honeypots are mostly used on learning new threats for the users that is on internet such as worms and new viruses and being aware of them. Thus, these kinds of honeypots are used to detect those malware and botnet. Their simulation algorithm is based on virtualizing logical responses for incoming requests. They are not virtualizing the whole operating system needs and they are not simulating application protocols in detail. When the request arrives to the medium interaction honeypot, that message is watched and examined, and fake responses are created. The difference between low level and medium level of interaction honeypots is medium level of interaction honeypots are not working on network stack and do the management on it. They bind on sockets and management is done thanks to the operating sysem itself. We will explain the most well known medium level of interaction honeypots as following.

As we stated it above, Nepenthes is developed with Mwcollectd. According to Maggi F. and Zanero S. (2008), Nepenthes is working on five modules which are vulnerability, shellcode parsing, fetching, logging and submission modules. Vulnerability function allows us to create vulnerable services. Shellcode parsing takes the payload and examine on it and get information about the extracted data. If any important data is found to examine, then fetch functionality gets the malware and submits to the center part. You can log the information that you have by using logging function of Nepenthes. Nepenthes is used for mostly malicious software that are spreading over internet automatically. Figure 4.3 is explaining Nepenthes architecture.

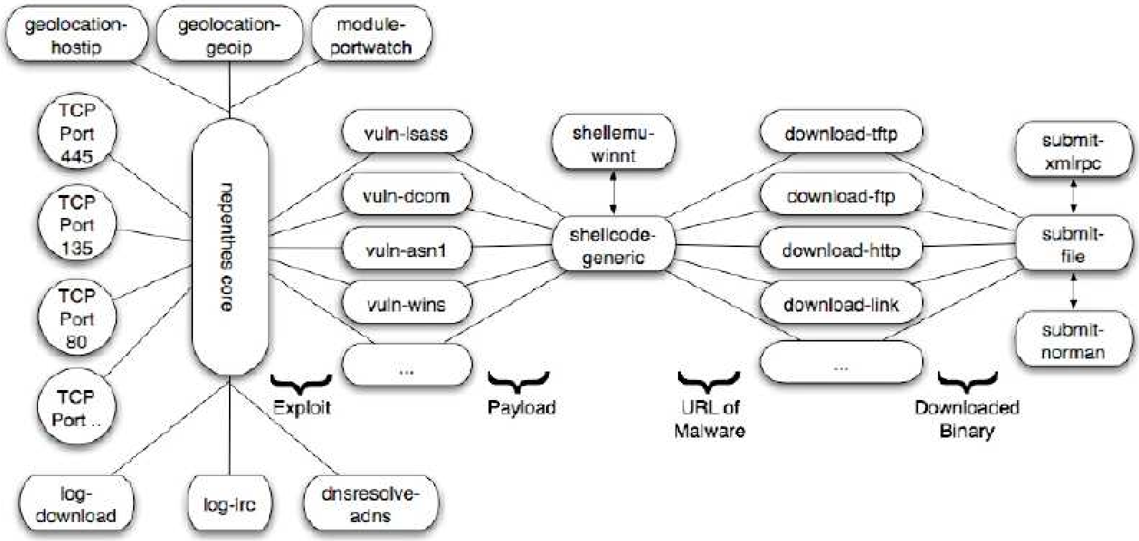


Figure 4.3 : Nepenthes architecture from Maggi F. and Zanero S. (2008)

We decided to install this software on Ubuntu operating system. The installation itself is simple as Nepenthes is present on the ubuntu repositories. To install the software, we used the command *apt-get install Nepenthes.* Once the process finished, we had to customize the configuration files. All of them are included in the folder /etc/Nepenthes/. The first one to check is Nepenthes.conf, it includes all the basic configuration on the software. The other ones are:

- submit-file.conf in which it is possible to set in which directory the downloaded malware will be stored.
- submit-norman.conf in which we set our email address. Norman Sandbox is an automated malware analyser. When Nepenthes will download a new malware, it will automatically be submitted to norman sandbox and the report will be send to our email address.
- log-download.conf in which we set the path of the logs for downloaded malware and malware submissions.

One of the strength of Nepenthes is that it emulates FTP and TFTP servers so the bot/attacker can upload the malicious software to the honeypot which allows the forensic party to analyze the threat.

After finishing the configuration, the last step is to put the honeypot on a DMZ and wait for the results.

An experience lead by Jean-Michel Phillipe in 2007 for 192 days using Nepenthes showed the following results:

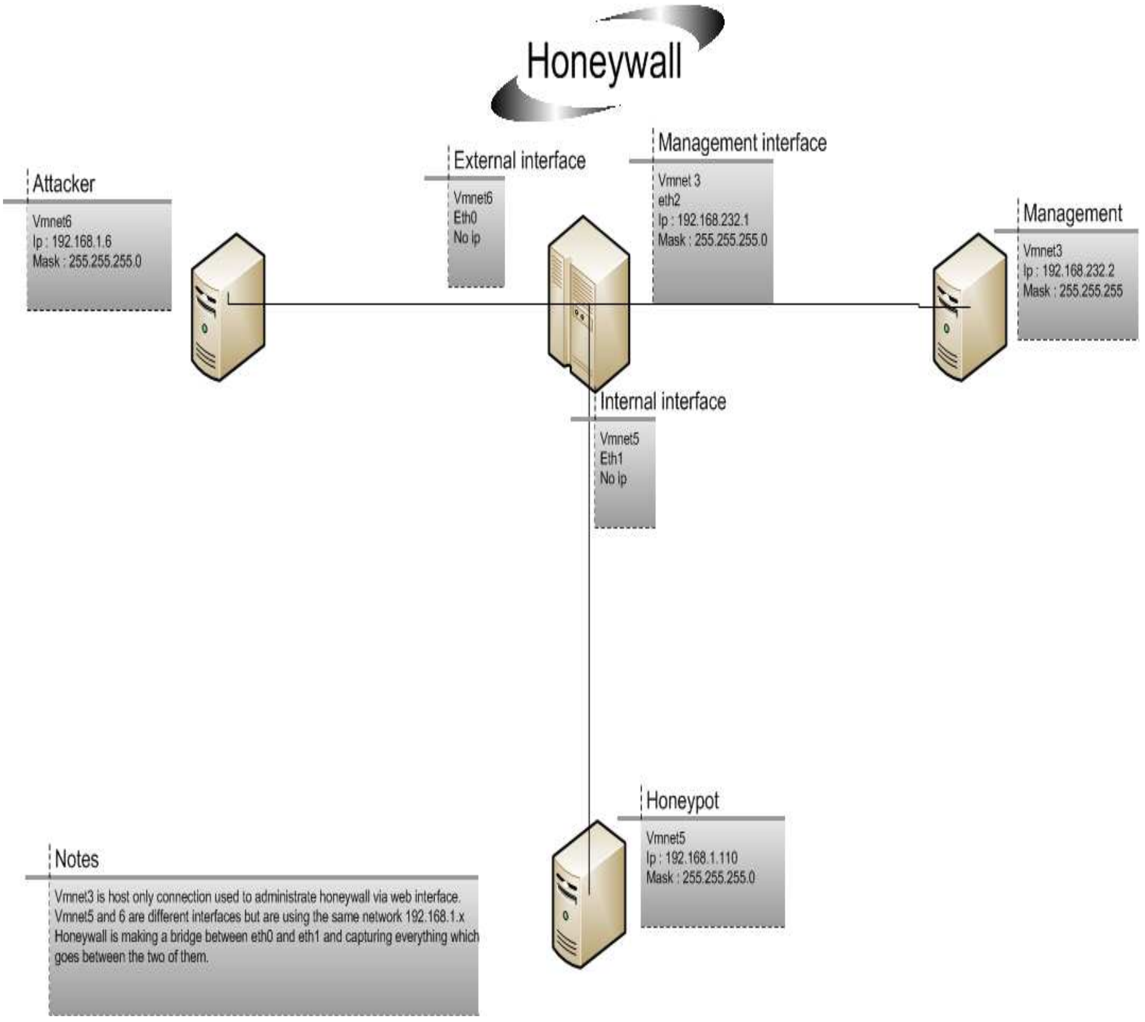
- One malware downloaded every 17 seconds.
- More or less 10 new malware per day and only a few detected as malicious.
- Almost only malware targeting Windows operating systems.

According to the researches of Baecher P., Koetter M., Holz T., Dornseif M., Freiling F., (2006), Nepenthes is giving us a great way of expressiveness. We can also create some vulnerability modules in it. That is what makes Nepenthes more interesting to work on. Thanks to this unique module, we can define so many vulnerabilities. This key point is also not possible in high level of interaction honeypots. Thus, in our experiment, it was very good to see and discover different sides of different level of interaction honeypots. Nepenthes has scalability and flexibility features. We can compare it with several honeypots. For example, we deployed Honeyd as a low interaction honeypot before. And, Honeyd cannot virtualize complicated protocols. Even high interaction honeypot Gen III is not very good at scalability. There are limitations of creating several honeypots. However, we learned that Nepenthes is covering the blanks that several honeypot do not have. We can create many honeypots in our system and capture data easily.

4.4 High level of interaction honeypots : Honeywall

Our last experiment will be based on high level of interaction honeypots. As we examined two types of interaction honeypots, we will move on further on implementation. Both low level and medium level of interaction honeypots offer more or less the same things. Services are emulated and you have restrictions. Thanks to several network monitoring tools, it is easy to understand what is going on throughout the traffic and understand that they are honeypots. Now, with high level of interaction honeypots, we will discover more on honeypots and with real operating system we will be able to catch more useful and interesting findings. Hackers will be freer with a real system without restrictions. Implementation will be time consuming and complicated. Our aim is to investigate if it is difficult to hack it and understand its structure, and detecting possible problems related to it and finding appropriate solutions or actions. Now, we will get to know available high level of interaction honeypot products currently exist in the market.

In high level of interaction honeypots part, we experimented on Honeywall. Our implementation is shown on the figure 4.4 below.



Vmnet3 is host only connection used to administrate honeywall via web interface.
Vmnet5 and 6 are different interfaces but are using the same network 192.168.1.x
Honeywall is making a bridge between eth0 and eth1 and capturing everything which goes between the two of them.

Figure 4.4 Our Honeywall implementation

For our experimentation, we decided to use virtual machines. It allows us to create our network without much physical equipment. We created three virtual machines:

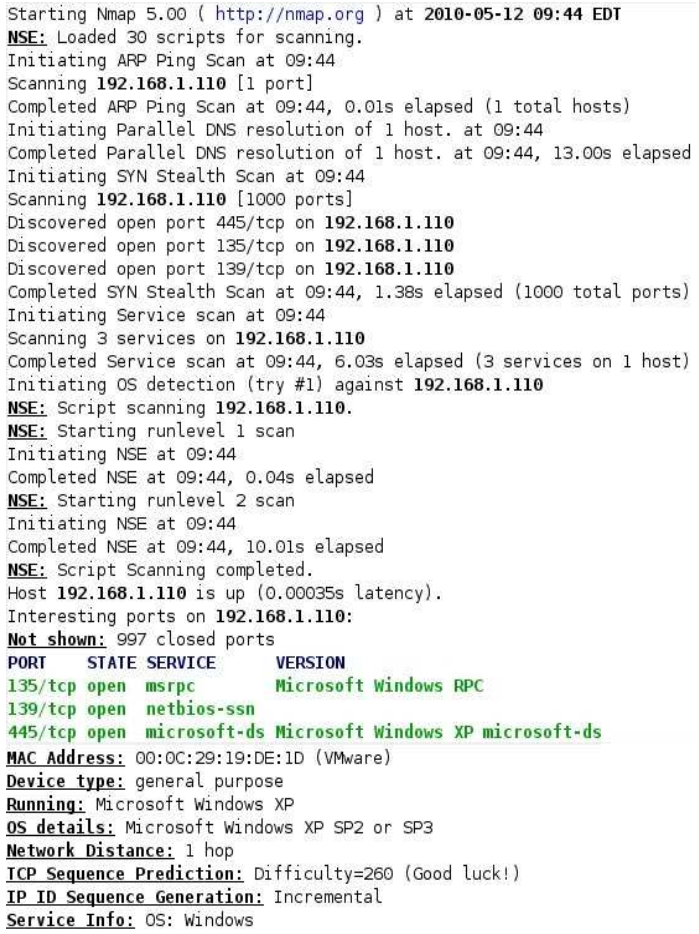
-One hosting Honeywall

-One hosting the attacker machine, we installed backtrack4 on it

-One hosting our honeypot, an unpatched Windows XP sp3

The Honeywall has three virtual network interfaces. eth0 is bridged to vmnet6, it is the attacker side. Eth1 is bridged on vmnet5, it is the honeypot side. Finally, eth2 is bridged to vmnet3, it is the management administration, and it allows remote administration of Honeywall. We did not create a virtual machine for the management part, we used a host only connection with the computer hosting all the virtual machines, this way we did not need another virtual machine just to administrate the Honeywall. Eth0 and eth1 are making a bridge, thus none of these interfaces have a network address making these two interfaces invisible. Honeywall does not give a choice on that part, but it is the best way to keep it undetected.

Once we managed to install and run all the virtual machines properly, we used the attacker machine in order to hack the honeypot. The first step is to detect any security flow that we could exploit. In order to do that, we used two tools very known: Nmap and Nessus. Nmap is a port scanner offering a lot of options (type of scan, level of detail about the target, etc…). The result of our scan is the figure 4.5 below:



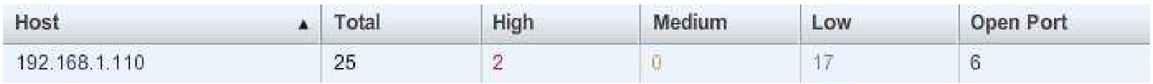
```
Starting Nmap 5.00 ( http://nmap.org ) at 2010-05-12 09:44 EDT
NSE: Loaded 30 scripts for scanning.
Initiating ARP Ping Scan at 09:44
Scanning 192.168.1.110 [1 port]
Completed ARP Ping Scan at 09:44, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:44
Completed Parallel DNS resolution of 1 host. at 09:44, 13.00s elapsed
Initiating SYN Stealth Scan at 09:44
Scanning 192.168.1.110 [1000 ports]
Discovered open port 445/tcp on 192.168.1.110
Discovered open port 135/tcp on 192.168.1.110
Discovered open port 139/tcp on 192.168.1.110
Completed SYN Stealth Scan at 09:44, 1.38s elapsed (1000 total ports)
Initiating Service scan at 09:44
Scanning 3 services on 192.168.1.110
Completed Service scan at 09:44, 6.03s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against 192.168.1.110
NSE: Script scanning 192.168.1.110.
NSE: Starting runlevel 1 scan
Initiating NSE at 09:44
Completed NSE at 09:44, 0.04s elapsed
NSE: Starting runlevel 2 scan
Initiating NSE at 09:44
Completed NSE at 09:44, 10.01s elapsed
NSE: Script Scanning completed.
Host 192.168.1.110 is up (0.00035s latency).
Interesting ports on 192.168.1.110:
Not shown: 997 closed ports
PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows XP microsoft-ds
MAC Address: 00:0C:29:19:DE:1D (VMware)
Device type: general purpose
Running: Microsoft Windows XP
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows
```

Figure 4.5 Our Nmap result

Many precious information have been gathered, thanks to the scan we could identify the operating system running on the target (Windows XP sp2 or sp3) and which port are open (135, 139 and 445). We also obtained some information about the network card used, we used the default value for the MAC address of the machine, so Nmap detected it as a

Vmware network card. We could have changed this address in order to avoid the virtual machine detection.

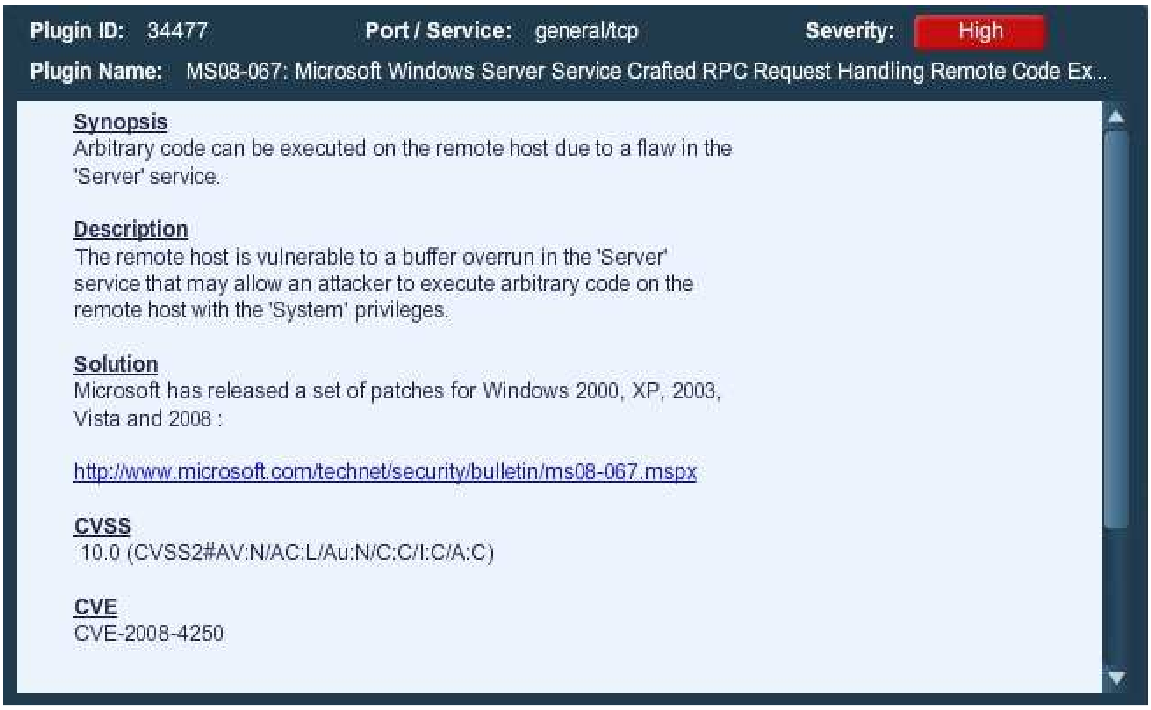
Windows XP is known for its security holes, especially if not patched regularly. In order to check that, we used the security scanner Nessus. This tool is free to use on its home version, the professional version is charged. Nessus is able to detect and report any security problem for the target it is scanning. On the report page, a link is provided to install the right patch and protect the system. On the hacker point of view, it just highlights which exploits he can use. After the scan, we obtained the following result shown in the figure 4.6:



Host ▲	Total	High	Medium	Low	Open Port
192.168.1.110	25	2	0	17	6

Figure 4.6 Nessus result report

Nessus detected two critical problems, by opening the pane to take a closer look at it, we saw that the machine had the vulnerability ms08-067. Figure 4.7 is showing this vulnerability.



Plugin ID: 34477 **Port / Service:** general/tcp **Severity:** High

Plugin Name: MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling Remote Code Ex...

Synopsis
Arbitrary code can be executed on the remote host due to a flaw in the 'Server' service.

Description
The remote host is vulnerable to a buffer overrun in the 'Server' service that may allow an attacker to execute arbitrary code on the remote host with the 'System' privileges.

Solution
Microsoft has released a set of patches for Windows 2000, XP, 2003, Vista and 2008 :

http://www.microsoft.com/technet/security/bulletin/ms08-067.mspx

CVSS
10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

CVE
CVE-2008-4250

Figure 4.7 Nessus vulnerability report MS08-067

As we can see on the screenshot, this vulnerability allows a hacker to execute code with System privileges. Now that we know what to exploit, we updated metasploit framework 3. This tool contains a huge library of exploits and payloads and can very easily hack into unprotected machines. Metasploit can be used via command line, or a user interface, or a web interface. We chose to use the web interface for its conviviality. In the research field for exploits, we wrote ms08-067 and after a few seconds the exploit was displayed. We selected it, and chose automated targeting (if not, we had to chose the operating system we were attacking). Then we chose the payload to use, in our case we wanted to create a new user on the machine who will have administrator privileges. The final step is to fill the ip address of the target and click on exploit. A command shell appeared to tell us that the exploit was successful. We checked on the honeypot the list of users, and the new user was here. This showed us how easy it is to hack an unprotected Windows XP. We used a simple payload to create a user, but we could have used a remote shell or a VNC session to fully control the computer.

If the attacker realizes what he is up against, he may want to hack into the Honeywall itself. Thanks to its architecture, the system is really hard to get into. The only way to take control over it is to have a physical access to the machine or to find a way into the management interface. A prudent administrator would most likely isolate that interface from the rest of the network to ensure its security. With this setup, Honeywall should be perfectly secure. However, if the administrator wishes to have a less restricted access to the remote administration of the system, it would also give an attacker the chance to enter the management as well. For that reason the design of Honeywall has to be very carefully studied before its implementation.

5. OTHER PRODUCTS

This chapter covers honeypot products we did not implement but which still are interesting to know about.

5.1 Medium interaction

5.1.1 Mwcollectd

Mwcollect was published by Georg Wicherski as a open source medium level of interaction honeypot for the first time. Later Nepenthes and Mwcollectd merged together to form better functionalities. Mwcollectd is working under Unix systems.

5.1.2 Multipot

Multipot is a medium level of interaction honeypots to be used under Windows operating systems. It has the same structure as Nepenthes has. There are six functions of multipot. When the request is arriving to multipot, five of the shellcode functions handle creating responses to it. As far as the location of the malicious software is found, it is downloaded for examining.

5.2 High interaction

5.2.1 Argos

Argos is mostly used for capturing new threats over internet such as worms and viruses or any other malicious action. It was developed by Vrije Universiteit in Amsterdam. It both supports Linux and Windows systems. Argos arose from the emulator called Qemu according to Maggi F. & Zanero S. (2008). Quemu emulator is following the traffic over network using instructions, jump targets and function addresses. After that, to be able to recognize those actions, Argos is using dynamic taint analysis which is also referred as memory tainting.

Argos is a different tool as it is possible to track a malicious incoming traffic without using signatures thanks to dynamic taint analysis. Dynamic taint analysis is performing the data that is arriving to the system and all the data that are not safe for the system are being tagged by it. All the traffic is tagged because we cannot trust the incoming traffic. After that those tagged data are handled and examined in more detail. When they are executed, it is possible to watch them. If the tainted data executes, alarm is launching and that action is being logged on the system. Therefore, it is possible to understand any malicious behavior that targets your system. Log is displaying the description of the attack mostly registers, network information and physical memory blocks.

Argos and Minos are both using virtual machine technology. This is useful since we can simulate many virtual machines up and running in a single machine. On the other hand, using one computer with one system is very limited when it comes to experiment honeypots. It is also useful to use virtual machines, because later the honeypot may be used for further attacks. The environment is safer when we are working with virtual machines.

Figure 5.1 is showing the flow of Argos and how it is working. We will explain the steps one by one that is shown on the figure according to Portokalidis G. & Slowinska A. & Bos H.'s studies. (2006)

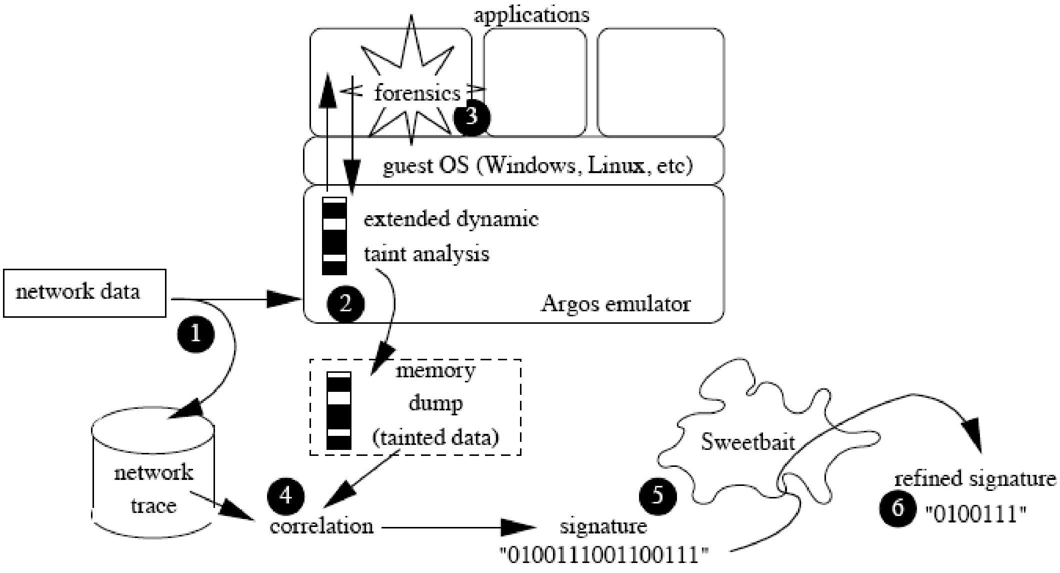


Figure 5.1 Flow of Argos from Portokalidis G., Slowinska A., Bos H.'s studies. (2006)

As it is seen from the figure 5.1 there are six steps. In step 1, the traffic is arriving to our host and being sent to the network trace that is collecting the incoming traffic and stores it to trace it. Emulator is doing the dynamic taint analysis as we mentioned before in step 2. There are three main approaches to understand the vulnerability of the data arrived:

1) Tagging the data when it is coming from a source which is not safe.
2) Following the tainted data when it executes.
3) Trying to block it from doing any malicious action.

In case of an emergency there is an alarm system that is informing user to show that there is a problem. At this point, alarm is forwarding us to the signature generation phase on step 3 – 6.

Step 3 is about forensics. When there is an attack, forensics step is trying to discover more about the attack finding its name its process identifier and so on. The code is injected to a system in order to perform forensic functions on the system.

Step 4 is trace database. At this step, memory dump is stored and detailed information is being reached by us when it is necessary. When there are TCP connections, we build flows. After that, the signature becomes ready to be filtered. As default, we would like to change the signature and optimize it. Therefore, Argos sends the signature to SweetBait system which is relating the signatures from several sites.

Last step which is step 6 is about use of signature. SweetBait has signatures that we can use for the incoming traffic. Our purpose may be blocking the traffic or keeping an eye on the traffic. SweetBait offers both of these functions.

5.2.2 Minos

According to Crandall J.R & Wu S.F & Chong F.T.'s studies Minos' emulations are very useful for honeypot architecture and it works under Linux and Windows systems. Minos is detecting attacks that results in damaging the data. So, there is control flow on Minos. The main objective of Minos is to check the data if it is fully not corrupted by anything. Using flow control, it is saving the data from unexpected problems. Flow control also introduces sending and receiving data.

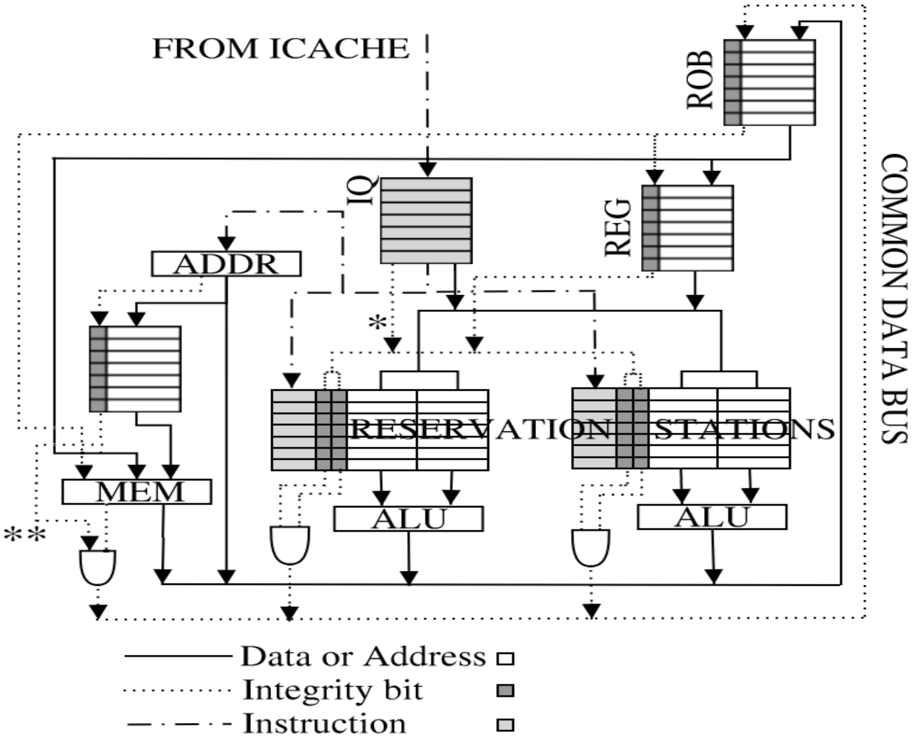
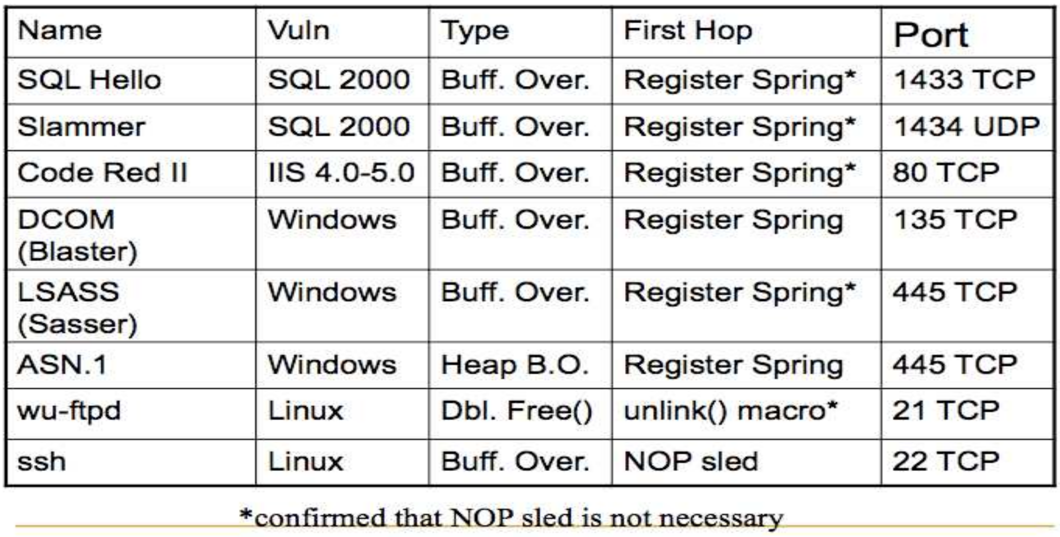


Figure 5.2 Structure of Minos from Crandall J.R & Wu S.F & Chong F.T. slides

Figure 5.2 is showing how the structure is working on Minos. When data is arriving to Minos, Minos is tagging them one by one. Then it is using Biba’s low water-mark integrity policy. When the data is travelling, Biba’s low water-mark integrity policy is being used on the devices. To be able to understand if there is an attack occurring on the system, we simply check integrity and if the integrity is low, then it means there is a problem. Figure 5.3 is showing the captured attacks by Minos.



Name	Vuln	Type	First Hop	Port
SQL Hello	SQL 2000	Buff. Over.	Register Spring*	1433 TCP
Slammer	SQL 2000	Buff. Over.	Register Spring*	1434 UDP
Code Red II	IIS 4.0-5.0	Buff. Over.	Register Spring*	80 TCP
DCOM (Blaster)	Windows	Buff. Over.	Register Spring	135 TCP
LSASS (Sasser)	Windows	Buff. Over.	Register Spring*	445 TCP
ASN.1	Windows	Heap B.O.	Register Spring	445 TCP
wu-ftpd	Linux	Dbl. Free()	unlink() macro*	21 TCP
ssh	Linux	Buff. Over.	NOP sled	22 TCP

*confirmed that NOP sled is not necessary

Figure 5.3 Captured attacks by Minos from Crandall J.R & Wu S.F & Chong F.T. slides

5.2.3 ManTrap

Mantrap is another high level of interaction honeypots which is commercial. It is sometimes called medium level of interaction honeypot as well. Therefore, it is providing the features that two of the honeypot types have individually. Like any other high level of interaction honeypot, there is an operating system working and there is no emulation of services. Lance Spitzner, (2002) is claiming that ManTrap is widely used on companies to test their security flaws. It is easy to configure and make it run smoothly. There is a great advantage of ManTrap. It is able to capture many necessary data that helps to understand the incoming attacks. Therefore, it makes experiments deeper and more interesting. It is detecting attacks and the connections which are not authorized. However, it is detecting more than that. It is able to find new types of attacks and threats for the users. Disadvantage of Mantrap is that only Solaris operating system can run it.

6. FORENSICS POINT OF VIEW

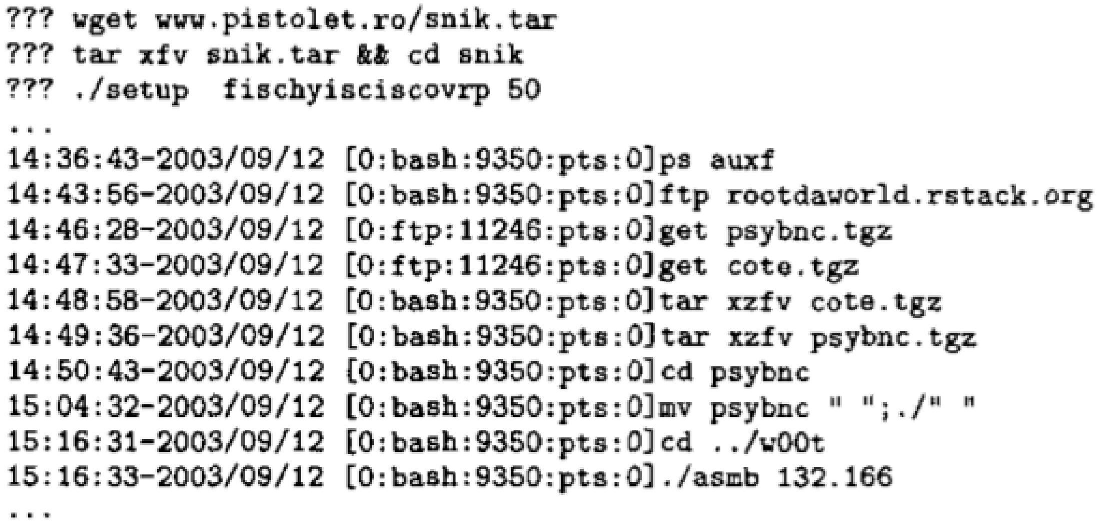
First of all, honeypot forensics is used to study and understand a hacker strategy and his tools but not to prosecute him. This science is very time consuming and according to honeynet project members, one hour of hacker activity can lead to more than 40h of forensic work. The suggested approach is to work on a copy of the original victim, that way the analysis process can be repeated from the beginning without losing any important data. Forensic in computer science require a perfect knowledge of hacker techniques as well as how different software works in general. Forensic science is to find evidences to make researches on it and trying to find some details and answers from it. The forensic science branch that we are interested in our thesis is computer forensics which is the same definition of forensic science but this time electronic devices are involved with our researches. The necessary data is obtained from the devices, and forensic investigators make deeper examination on them. There are several roles and responsibilities for forensic investigation. Forensic investigation is done with first responders, investigators, technicians, evidence custodians, forensic examiners and forensic analysts. (Kipper G., (2007)).

The different honeypots we studied offered us several log files that a forensic party can analyze. The most common file to study when we talk about network security is the .pcap file that most honeypots are generating. This file contains all the packets exchanged between the attacker and its target. It can be opened with Wireshark and allow the forensic to see what communication happened. This file can be huge in size but contains very important information. The difficulty here is to sort the relevant information. In the case of a honeypot, we assume that all traffic is suspicious thus any IP address not within our network must be analyzed. This make the sorting easier than on a production network where the attack is harder to detect.

Another part of the forensic work is called reverse engineering. When a hacker successfully compromises a system, he will most likely upload one or more malware. Reverse engineering take a closer look at these malware by decompiling it and trying to understand what are their purposes and how they work. Again this technique is very time consuming but can allow the forensics team to identify new threats.

Raynal F. & Berthier Y. & Biondi P. & Kaminsky D., (2004) highlight that there are four steps in honeypots forensic to be considered: Know your honeynet, know your network, know your system and know your enemy. Thus, first of all we should know the honeypot that we are using. First step shows us that we should be aware of what we are capturing. For example, connecting our honeypot to the internet will give us different results compared to connecting the honeypot to some other networks. Know your network part clearly states that we are dealing with the network, so our forensic part will be based on the experiments that we are performing on network traffic. Know your system mainly focuses on the activities of the hacker. He may download on the system some malicious software and so on. Last step which is know your enemy states that we did all the experiments and we have our evidences and we found the traces of the hacker in our system. We should analyze it and come up with the conclusion. On forensic part of our work, thanks to several forensic tools, we can have network traffic and logs from the hacker. Our analysis is very important and meaningful since the traffic inside and outside is critical. During our high interaction honeypot experiment we obtained pcap file and by looking at it we can address the intruder.

Next approach for forensic analysis according to them is the system part. They experienced something very interesting. After the attack happened, hacker tried to make whole analysis of the system. At this part Sebek is a very useful tool to show the captured information from the hacker's activities. The figure 6.1 below is a Sebek output example:



```
??? wget www.pistolet.ro/snik.tar
??? tar xfv snik.tar && cd snik
??? ./setup  fischyisciscovrp 50
...
14:36:43-2003/09/12 [0:bash:9350:pts:0]ps auxf
14:43:56-2003/09/12 [0:bash:9350:pts:0]ftp rootdaworld.rstack.org
14:46:28-2003/09/12 [0:ftp:11246:pts:0]get psybnc.tgz
14:47:33-2003/09/12 [0:ftp:11246:pts:0]get cote.tgz
14:48:58-2003/09/12 [0:bash:9350:pts:0]tar xzfv cote.tgz
14:49:36-2003/09/12 [0:bash:9350:pts:0]tar xzfv psybnc.tgz
14:50:43-2003/09/12 [0:bash:9350:pts:0]cd psybnc
15:04:32-2003/09/12 [0:bash:9350:pts:0]mv psybnc " ";./" "
15:16:31-2003/09/12 [0:bash:9350:pts:0]cd ../w00t
15:16:33-2003/09/12 [0:bash:9350:pts:0]./asmb 132.166
...
```

Figure 6.1 Sebek output example from Raynal F. & Berthier Y. & Biondi P. & Kaminsky D., (2004).

We should try to understand what this output means, and evaluate the results. First of all, any intruder looks for a target to hack and for this purpose he tries to obtain necessary information to do it. There are some commands to have information about the target such as ifconfig, netstat and so on. After he knows the system, he starts to do this job. According to his aim, he installs some software on the target. For example, if he wishes to connect to the system several times, he installs a backdoor to achieve his goal. Once he finishes those initial preparations for the target, he jumps to the next step. Next step is to completely use the system the way he wants it. To be able to keep up the things that hacker is doing to the system, we should keep track of him very closely. We should analyze his techniques deeply. We should definitely find a timeline and reconstruct it over time.

There are several ways for a hacker to intrude the system. He may use tools, scripts and so on. We know how to analyze the tools and gather data from them. Analyzing scripts are also important. By looking at the scripts, we may gain a lot of useful information. We should perform linear analyze which means checking every action that has happened. After the linear analyze we should move on to classification step. Classification is to look at those actions by their classes. There are different commands for the same aims, each aim correspond to a class. If there is no class associated, we should have a look at them too in unclassified session. After completing all these steps, we gain meaningful information at the end.

7. INTERVIEW ABOUT HONEYPOTS

For our thesis project,we contacted Combitech company in Sweden. With the great help of our supervisor Ola Flygt, we found lawyer Pehr Jern who is working in Combitech in Stockholm. Therefore, we made a small interview asking him about the legal aspects of honeypots.

Deniz: What are the laws in Sweden about deploying honeypots? Is it illegal ? What are the restrictions ?

*Pehr:*There is no such laws, it depends on the purpose. You should not advertise your system, the hacker should come by himself and not be invited in.

*Deniz:*What would you recommend to improve honeypot laws in Sweden ? What could be done ?

Pehr: It could be used in one good purpose to try to prove security branches. If you would want the regulation to be changed, you would actually set one up, it requires fundamental changes against crime. It is not necessarily a crime, it depends on the committing criminal, that would change the law. Number of people who writes text, they do not know about IT. Judges, they do not know it. It is difficult to make both sides work. I am very skeptical on the possibilities to change this. This sort of crime is not very well known, might be a problem. But it is not a subject in Sweden.

*Deniz:*According to you, what are the criterias to consider before deploying honeypot in any country ?

Pehr: You must be aware of the purpose before setting up one. If you do it on a commercial basis, and the purpose is to carry out missions for companies that you have contract with saying that you are allowed to do so , or to prove something. And to see how frequent people are authenticating to honeypots and not knowing that it might be something oddly. If you push the attacker to commit the crime, then setting up one could be a crime depending what the fullfil crime is suppose to be. I cannot say it in general terms if it would be illegal to set one up. It all depends on the crime that you are about to commit. I think you would not suppose that sort of crime that where honeypots used a tools probably some sort of con or something. It can be illegal to set one up , you should be very much aware of the purpose before you set one up. There may be good purpose like doing a survey that no one really ask and you show surveillance of that sort of communication. But it can be possible to have a crime depending on what you do in order to prove something. So you should be very careful before you plan to set one up. I wish I could be more specific.

*Deniz:*Did you come across any specific honeypot cases in Sweden ?

Pehr: I have not really heard of one. As i said before it is not very well known that you are busted by someone who sets up one. This is not very well known. The survey that is done for Swedish authority , it shows that people are not very much aware of the risks on honeypots. People authenticate and so at hotel and especially at airports. Its all kind of uses but many of them do it on business. It is not working information in IT security. It is not what we advise really. You should not use that sort of communication in public places where you authenticate wireless. Very little possibility to check where you have been logged into. But it is not very well known. That is not something in an issue on the media.

*Deniz:*If there would be some crimes about deploying honeypot and if you were able to capture some criminals that are hacking honeypots in Sweden and law could be improved, because laws are improving with the cases that you are dealing with. You could be able to set the structure of laws by looking at the cases , so it could have been improved like that , otherwise how do you set the structure ? Because there is no case , it is really new in network security. Am I right ?

Pehr: As I mentioned also a bit before , there is a problem with the people writing the law and problem of understand IT. So,many of the crimes that can be committed partially, it is not the crime in a way that could be understood in a good way of people works with the IT sector, and general problem is writing the laws works for IT. It is a problem of traditional people who is actually writing laws in Sweden, because that people generally in the beginning of their jurist career. So you work with for example as something within the police or you work as a judge, and then you work at the justice department for three to four years. This is a huge problem that people within the law area does not really understand IT. The bigger problem is you are going to case the law, it is very much in the hands of experts.

Deniz: Yes, because as you stated before, lawyers and it security administrators should merge together and work. So, it is a big problem between these two professions. What do you recommend to improve it ?

Pehr: I wish I could have a straight answer just a simple advice. But the problem is once again in Sweden it is traditional how you write a law. In Sweden we tend to be very dependent on the sub text of law. There is no english word. In European law, like if you bring a case to the European court of justice, you do not have a lot of sub texts that where you describe how the law should be interpreted, because it has a different approach where you have a specific purpose for the law. And you should follow that purpose. You should not try bring a case court as soon as possible to make court to decide what direction the law should go in and what interpretion you should follow. But in Sweden the tradition is the sub text to the law, and that sub text is written by people who are in the early career of being a judge or possibly something within the civil services in some authority in Sweden. So, Swedish law and European law collides, then it could be a change but several one, not the criminal crime, so it is problematic. Crime as we know it , legislation it is national law in European community. So, I wish I had a more optimistic answer.

*Deniz:*What is the role of network security administrators on honeypots capturing hackers?

*Pehr:*There is a huge problem when you come to a point when you actually need to prove that the identity of the user in a security log. Security logs are by default and not very good. And the bigger problem with security logs is that maybe it shows what has been done with IP and MAC address. Maybe it even shows what it has been done with smart card. But the weak point is identifying the user behind.

We thank Mr. Pehr Jern for his valuable time to answer our questions. We are sorry if there would be some english understanding mistakes during the interview.

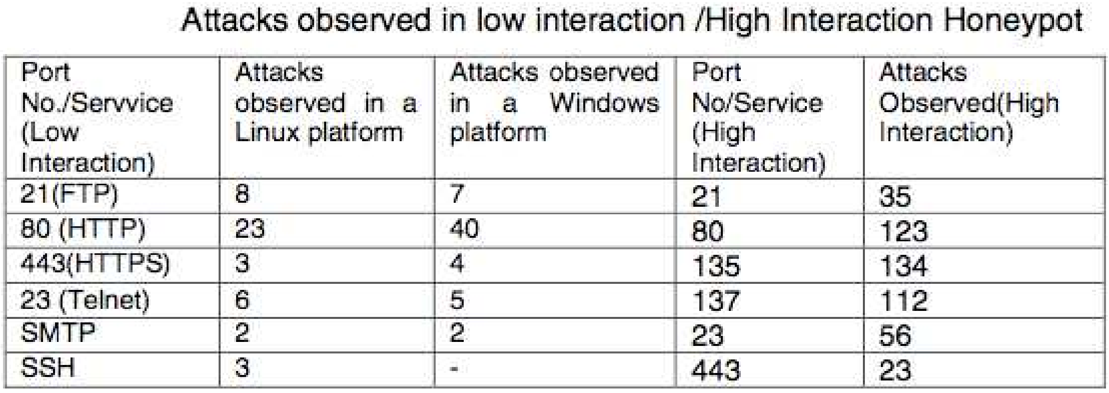
8. RESULTS

We studied all level of interaction honeypots and configured them. As first level of interaction honeypot, we deployed Honeyd. We explained the logic behind it and installed it correctly. Our findings about Honeyd are; Honeyd is the most popular low interaction honeypot but its problem is its age. The project is opensource but part of it is outdated and nobody seems to upgrade it. On the other hand hacker tools are evolving, so identifying this honeypot is not hard. Honeyd is using an old version on Nmap fingerprint to create fake virtual operating systems so by using a newer version of Nmap, the fake operating systems will not be recognized and Nmap will detect that there is a problem.

Another limitation of Honeyd is the scripts bound to the different ports. With a basic scan it is possible to find which ports are open but as soon as the attacker tries to actually connect on a port, he will realize the service is fake. For example the script used for a Web server, by connecting it using telnet, thew server should send back replies but nothing is happening. Another problem is one cannot understand if there is an incoming attack to the system or not. Because there is no such alarm system that can make you understand that there is an attack. Information gathering is not very smart either. As a result the hacker can understand quickly that there is something wrong with the target and will abort his attack. Even unprofessional intruders can compromise the honeypot without spending too much time on it. Because it is very popular and easy to use well known techniques such as Nmap. There is no additional approach needed for it.

Our second step was to configure medium level interaction honeypot Nepenthes. We explained how it works and how we studied on it in implementation part. However, we found some problems with Nepenthes too. First of all, Nepenthes is for capturing malware over internet. It is mostly used for this aim. Thus, it must be implemented very rapidly since threats for users over internet are increasing dramatically day by day. Nepenthes could not keep up with new threats. As new threats are arriving and Nepenthes is not up to date, it will not be able to capture malware. Another problem comes from the shellcode. Shellcode manager should consider about shellcode and understand it. As new threats cannot be captured, new exploits cannot be captured either. Furthermore, as we are investigating the problems and security flaws in our experiment, there is an important security flaw in Nepenthes structure. Nepenthes do not have transport layer security. Transport layer security is a protocol that gives security for communications throughout the internet. We think it is a real problem for honeypot deployment. Some malware exist on port 445 that are being involved with each other which are “LSASS, PNP, DCOM, ASN1, ms06-070, ms08-067”. When this kind of interference happens, we are not sure about the replies either. It creates a big mess between modules. (Schloesser M., (2009)).

Figure 8.1 is showing the attacks observed according to Maheswari V. & Sankaranarayanan Dr.P.E., (2007).



Attacks observed in low interaction /High Interaction Honeypot

Port No./Servvice (Low Interaction)	Attacks observed in a Linux platform	Attacks observed in a Windows platform	Port No/Service (High Interaction)	Attacks Observed(High Interaction)
21(FTP)	8	7	21	35
80 (HTTP)	23	40	80	123
443(HTTPS)	3	4	135	134
23 (Telnet)	6	5	137	112
SMTP	2	2	23	56
SSH	3	-	443	23

Figure 8.1 Attacks observed from Maheswari V. & Sankaranarayanan Dr.P.E., (2007).

The last step was the implementation of a high interaction honeypot: Honeywall. We explained his structure and how to set it up. Its architecture is solid, and an intruder will not understand at the first glance that he is entering a honeypot system. Honeywall is making a bridge between the attacker and the honeypots interface making the link invisible. All the traffic going through that link can be saved and analyzed. The main difficulty for this system is to create a believable honeypot system. The servers should be properly configured and not giving any hint about their purpose. The honeypot system needs to be vulnerable so the attacker can come in but not too open either or it will be obvious that it is a trap. The only way to hack into the Honeywall system is to access the management interface. This could be done by attacking the routers themselves so it is entirely up to the hardware in place and how the network administrator has set up the system. One of the strength of Honeywall is that a good forensic team will be able to identify new threats. Once the honeypot has been compromised, the team will have to analyze the machine and hopefully find out which tools the hacker used and how he used them.

9. CONCLUSION

In our thesis, we explained honeypot systems in detail, and implemented low interaction, middle interaction and high interaction honeypots at laboratory. Our goal was to understand their strategy and how they are working in order to lure intruders towards the system. We discovered their security flaws in order to help researchers and organizations. Several companies are using honeypot systems to protect the whole organization's network security, and researchers are making academic experiments on them at schools. As we all know network security is very significant for all computer systems because any unprotected machine in a network can be compromised in any minute. One may lose all the secret and important data of a company, which can be a great loss, and it is also very dangerous that someone else knows your important personal information. Thus, we tried to find answers for honeypots' security using all interaction honeypots possible.

Our main goal for our thesis was to see if honeypots are easy to hack and check if they are really isolated from other networks like a organization's network. When a honeypot is compromised, is it possible to reach other systems and compromise them too ? After the system is compromised, is it possible to track the hacker by using necessary forensic science tools ? How efficient are they ? As we stated in results and analysis part,we easily hacked all the honeypots that we used for our thesis.

Especially, low interaction honeypot Honeyd can be hacked easily without too much effort. As we stated before, any amateur hacker can seize the system and also can see that it is a trap system. Therefore, Honeyd is not a good honeypot as its features are not efficient to fool the hacker. As Honeyd is a deamon, it is just simulating a operating system's services. So, it is not possible to a hacker to seize other systems using Honeyd. For the intruder, it will not take time to see that the system is not real, so he will not continue compromising it. He will leave the system. For forensic part, Honeyd's log was sufficient to see the actions of the hacker.

Next part was to try Nepenthes as medium interaction honeypots. The result was quite similar. Thus,we came up with this conclusion: Low interaction honeypots and medium interaction honeypots are just simulating the services of a real system, because of that it is not possible to capture significant data from intruders. They are slightly different from each other but the main idea is the same. As they are not real operating systems , it is not risky to build them. There is no need to mention about further attacks.

So, we moved on to the last level. After working low interaction and medium interaction honeypots, we decided to deploy high interaction honeypots. We studied on Honeywall. Even though it is time consuming and difficult, we managed to create a structure and worked on it. Our result were more interesting than before.High interaction honeypots are not virtualizing the system. They are real systems.So, it is very risky but the captured information is important. After deploying the implementation correctly, we successfully hacked the honeynet, but not Honeywall itself. It was the result we were looking for.

As we stated in this paper, honeypot systems are still very new but are a great tool to identify cyber threats. The problem nowadays is that a very good hacker will most likely be able to understand when he is attacking a honeypot. Low interaction honeypots will be able to identify mostly automated attack and will hardly be able to understand new hacker method. On the other hand, high interaction systems are here to entrap the hacker and make him give away his techniques and tools to the forensic team. The network administrator implementing this kind of honeypot should make sure that the system is completely isolated from the production network. This is the best defense if the hacker compromises the honeypot.

Network security is not a path many students are taking but we see it as one of the most important topics when we speak about computing. We were curious about this subject and decided to write a thesis on that field. This work taught us a lot about the black hat and white hat community. It also gave us an idea how huge and complex the forensic work is. New threats are discovered everyday and the best way to stay protected is to always stay up to date. By doing this simple task, most attacks will not have any effect on the system. The problem nowadays is that people using pirated version of an operating system are contributing to botnets. Their system does not support critical updates and they are more sensitive to automated attacks.

Nowadays, the implementation and development of honeypots are under control by network security expert. The weakness of this system is that it is not backed up by a clear legislation. Most of the work in the future should be about improving the laws about honeypots. The current laws about honeypots in most of the countries are not clear. There is a gap between the lawyers and the IT professionals. They should learn to cooperate with each other in order to clarify the legislation and give a clear answer about the legality of this technology. A lot of work should be done in the future to improve this situation. On a technical aspect, the main difficulty is to keep up with the new attacks. These days, it is not hard to detect a honeypot system, most of the work should focus on making this technology stealthier.

REFERENCES

Alata E. & Nicomette V. & Kaâniche M. & Dacier M. & Herrb M.,2006.Lessons learned from the deployment of a high-interaction honeypot, *EDCC'06.*

Baecher P. & Koetter M. & Holz T. & Dornseif M. & Freiling F.,2006.The Nepenthes Platform: An Efficient Approach to Collect Malware.

Barel M., 2004. Honeypot : un pot-pourri…juridique, *Actes du symposium SSTIC04.*

Bishop M. & Frincke D.A.,2006.Computer Security Education and Research Handle with Care , *IEEE Computer Society*.

Crandall J.R. &Wu S.F & Chong F.T., Experiences Using Minos as a Tool for Capturing and Analyzing Novel Worms for Unknown Vulnerabilities University of California, Davis University of California, Santa Barbara [Online] Available at : http://www.cs.unm.edu/~crandall/DIMVAMinos.ppt [Accessed 3 May 2010]

Glasvezel Networks Nederland Fiber to the Home Project (FTTH),2009.Honeypot Software,Honeypot Products,Deception Software [Online] (Updated 23 October 2009) Available at : http://www.honeypots.net/honeypots/products [Accessed 5 May 2010].

Kipper G.,2007.*Wireless Crime and Forensic Investigation.*USA:Auerbach Publications,Taylor & Francis Group.

Lakhani A.D., A dissertation on deception techniques using honeypots.Information Security Group Royal Holloway, University of London,UK.

Maggi F. & Zanero S.,2008.ICT-216026-WOMBAT Worldwide Observatory of Malicious Behaviors and Attack Threats,Analysis of the state-of-the-art.

Maheswari V. & Sankaranarayanan Dr.P.E.,2007.Honeypots:Deployment and Data Forensic Analysis,*International Conference on Computational Intelligence and Multimedia Applications 2007.*

McFarland B.,2005. Ethical Deception and Preemptive Deterrence in Network Security GCFW Practical Version 4.1 , *SANS Institute 2000-2005.*

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

Phillipe JM. 6 mois de capture de logiciels malveillants [Online](Updated 24 july 2007). Available at : http://philipjm.free.fr/blog/index.php?2007/07/24/113-6-mois-de-capture-de-logiciels-malveillants/ [Accessed 2 May 2010].

Portokalidis G.,2008. Argos An emulator for capturing zero-day attacks [Online] (Updated 16 April 2009**)** Available at **:** http://www.few.vu.nl/Argos/ [Accessed 2 May 2010].

Portokalidis G. & Slowinska A. & Bos H.,2006. Argos: an Emulator for Fingerprinting

ZeroDay Attacks for advertised honeypots with automatic signature generation, *EuroSys'06 April 18–21, 2006.*

Provos N.,2008.*Developments of the Honeyd Virtual Honeypot*. [Online] (Updated July 15 2008 06:48:04 PM) Available at : http://www.Honeyd.org [Accessed 20 March 2010].

Provos N. & Holz T., 2007. *Virtual Honeypots: From Botnet Tracking to Intrusion Detection* , Addison Wesley Professional.

Raynal F. & Berthier Y. & Biondi P. & Kaminsky D.,2004.Honeypot Forensics,*Proceedings of the 2004 IEEE Workshop on Information Assurance and Security.*

Sadasivam K. & Samudrala B. & Yang T.A.,2005. Design of network security projects using honeypots.p.282-291

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

Spitzner L.2001.The Value of Honeypots, Part Two: Honeypot Solutions and Legal Issues [Online] (Updated 23 October 2001) Available at : http://www.symantec.com/connect/articles/value-honeypots-part-two-honeypot-solutions-and-legal-issues [Accessed 13 March 2010].

Sutton Jr.,R.E DTEC 6873 Section 01:How to build and use a honeypot.

Shuja F.A.,2005. Pakistan Honeynet Project Virtual Honeynet: Deploying Honeywall using Vmware [Online](Updated November 2005) Available at: http://www.honeynet.pk/Honeywall/roo/index.htm [Accessed 7 May 2010].

Schloesser M.,2009.Low Interaction Server Honeypot Evolution,Giraffe honeynet project,FIRST Technical Colloquium,Kuala Lumpur.

Siles R.,2007.HoneySpot:The Wireless Honeypot the Attacker's Activities in Wireless Networks A design and architectural overview ,The Spanish Honeynet Project (SHP).

Talabis,R.,2005.Honeypots 101:Risks and Disadvantages.

Linnæus University
School of Computer Science, Physics and Mathematics

SE-351 95 Växjö / SE-391 82 Kalmar
Tel +46-772-28 80 00
dfm@lnu.se
Lnu.se