COVER PAGE

PLANNING, DESIGN OF HONEYPOT AT CUTM BHUBANESHWAR CAMPUS

A PROJECT REPORT

Submitted by

CHIRANJIB PARIDA

In partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

CSE-CTIS ENGINEERING

CENTURION INSTITUTE FOR TECHNOLOGY & MANAGEMENT

BHUBANESHWAR

DECEMBER 2018 / MAY 2019

CERTIFICATE

DEPARTMENT OF CSE-CTIS ENGINEERING

CENTURION INSTITUTE FOR TECHNOLOGY & MANAGEMENT

BHUBANESHWAR -752050

BONAFIDE CERTIFICATE

Certified that this project report ***Planning, Design of HONEYPOT at CUTM BHUBANESHWAR Campus*** is the bonafide work of "HONEYPOT" **NAME OF THE CANDIDATE(S) "CHIRANJIB PARIDA (160301200008)"** who carried out the project work under my supervision. This is to further certify to the best of my knowledge that this project has not been carried out earlier in this institute and the university.

Certified that the above mentioned project has been duly carried out as per the norms of the college and statutes of the university

ACKNOWLEDGEMENTS

I wish to express my profound and sincere gratitude to Dr. DEBASIS MOHANTY, Department of CSE-CTIS Engineering, CUTM BHUBANESHWAR, who guided me into the intricacies of this project non-chalantly with matchless magnanimity.

I thank Prof. DEBASIS MOHANTY, Head of the Dept. of CSE-CTIS Engineering, CUTM BHUBANESHWAR and Prof. PRASHANTA MOHANTY, DEAN, CUTM for extending their support during Course of this investigation.

I would be failing in my duty if I don't acknowledge the co-operation rendered during various stages of image interpretation by DEBASIS MOHANTY

I am highly grateful to DEBASIS MOHANTY who evinced keen interest and invaluable support in the progress and successful completion of my project work.

I am indebted to Prof. DEBASIS MOHANTY for their constant encouragement, co-operation and help. Words of gratitude are not enough to describe the accommodation and fortitude which they have shown throughout my endeavor.

CHIRANJIB PARIDA

TABLE OF CONTENTS

1. INTRODUCTION

Scale of Internet technology is very large and it is still growing every day. The security of network is required for improvement of the industries which are dependent on the internet to enhance the business and providing services on the network. So security of network is primary concern of the industries for securing the critical information. Big sums of attacks are noticed in recent years on these kinds of industries. Intrusion detection system (IDS) is used for monitoring the processes on a system or a network for examining the threats and alert the administrator. IDS and firewalls are used for protecting the system and network from attacks, but after so many efforts for security still the network is not fully secured so different types of solutions are proposed by the experts. The small scale industries using LAN have to keep high their own security level as the database, server and clients are all handled by themselves. Since threat from internal network is Always the big challenge for the administrators, so a solution is required for small scale network to secure their internal network. This report provides the solution for the same using honeypot.

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.

2. HONEYPOTS AND THEIR AIMS

We will explain what a honeypot is and its purpose. We will also present its history and see its advantages.

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.

2.3. 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 introduced 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.

3. KEYWORD

The keyword is defining, those are using in this project.

3.1. Security

Security, Information Security (IT), is the digital information and IT assets against internal and external, malicious and accidental threats. This defense includes detection, prevention and response to threats through the use of security polices, software tools and IT services.

3.2. Mmh

Files which are given the .MMH extension are known as Media Manager Helper DLL files, however other file types may also use this extension. If you are aware of any additional file formats that use the MMH extension, please let us know.

The best way to open an MMH file is to simply double-click it and let the default associated application open the file. If you are unable to open the file this way, it may be because you do not have the correct application associated with the extension to view or edit the MMH file.

3.3. VMWare Workstations

VMware Workstation is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems (an x86 version of earlier releases was available); it enables users to set up virtual machines (VMs) on a single physical machine, and use them simultaneously along with the actual machine. Each virtual machine can execute its own operating system, including versions of Microsoft Windows, Linux, BSD, and MS-DOS. VMware Workstation is developed and sold by VMware, Inc., a division of Dell Technologies. There is a free-of-charge version, VMware Workstation Player, for non-commercial use. An operating systems license is needed to use proprietary ones such as Windows. Ready-made Linux VMs set up for different purposes are available from several sources.

VMware Workstation supports bridging existing host network adapters and sharing physical disk drives and USB devices with a virtual machine. It can simulate disk drives; an ISO image file can be mounted as a virtual optical disc drive, and virtual hard disk drives are implemented as .vmdk files.

VMware Workstation Pro can save the state of a virtual machine (a "snapshot") at any instant. These snapshots can later be restored, effectively returning the

virtual machine to the saved state, it was and free from any post-snapshot damage to the VM.

VMware Workstation includes the ability to group multiple virtual machines in an inventory folder. The machines in such a folder can then be powered on and powered off as a single object, useful for testing complex client-server environments.

3.3.1. Virtual Machine

A virtual machine is a computer file, typically called an image, which behaves like an actual computer. In other words, creating a computer within a computer. It runs in a window, much like any other programme, giving the end user the same experience on a virtual machine as they would have on the host operating system itself. The virtual machine is sandboxed from the rest of the system, meaning that the software inside a virtual machine cannot escape or tamper with the computer itself. This produces an ideal environment for testing other operating systems including beta releases, accessing virus-infected data, creating operating system backups and running software or applications on operating systems for which they were not originally intended.

Multiple virtual machines can run simultaneously on the same physical computer. For servers, the multiple operating systems run side-by-side with a piece of software called a hypervisor to manage them, while desktop computers typically employ one operating system to run the other operating systems within its programme windows. Each virtual machine provides its own virtual hardware, including CPUs, memory, hard drives, network interfaces and other devices. The virtual hardware is then mapped to the real hardware on the physical machine which saves costs by reducing the need for physical hardware systems along with the associated maintenance costs that go with it, plus reduces power and cooling demand.

3.4. Operating System

An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs. Time-sharing operating systems schedule tasks for efficient use of the system and

may also include accounting software for cost allocation of processor time, mass storage, printing, and other resources.

3.4.1. Linux

UNIX originated as a research project at AT&T Bell Labs in 1969 by Ken Thompson and Dennis Ritchie. The first multiuser and multitasking Operating System in the world. Developed in several different versions for various hardware platforms (Sun Sparc, Power PC, Motorola, HP RISC Processors).

Linux is the best-known and most-used open source operating system. As an operating system, Linux is software that sits underneath all of the other software on a computer, receiving requests from those programs and relaying these requests to the computer's hardware.

Slackware was one of the first Linux distributions.

3.4.1.1. Parrot

Parrot Linux is a Linux distribution based on Debian with a focus on computer security. It is designed for penetration testing, vulnerability assessment and mitigation, computer forensics and anonymous web browsing. It is developed by the Frozenbox team.

The operating system ships with the MATE desktop environment preinstalled and is available in several flavors to fit your needs.

3.4.1.1.1. History of Parrot

The first public release appeared on April 10th, 2013 as the result of the work of Lorenzo Faletra who continues to lead development.

Originally developed as part of Frozenbox, the effort has grown to include a community of open source developers, professional security experts, advocates of digital rights, and Linux enthusiasts from all around the globe.

The project is headquartered in Palermo, Italy and it is supported by an international team of experts and enthusiasts.

3.4.1.1.2. What is Debian?

"The Debian Project is an association of individuals who have made common cause to create a free operating system. This operating system that we have created is called Debian.

An operating system is the set of basic programs and utilities that make your computer run. At the core of an operating system is the kernel. The kernel is the most fundamental program on the computer and does all the basic housekeeping and lets you start other programs.

Debian systems currently use the Linux kernel or the FreeBSD kernel. Linux is a piece of software started by Linus Torvalds and supported by thousands of programmers worldwide. FreeBSD is an operating system including a kernel and other software.

However, work is in progress to provide Debian for other kernels, primarily for the Hurd. The Hurd is a collection of servers that run on top of a microkernel (such as Mach) to implement different features. The Hurd is free software produced by the GNU project.

A large part of the basic tools that fill out the operating system come from the GNU project; hence the names: GNU/Linux, GNU/kFreeBSD, and GNU/Hurd. These tools are also free.

Of course, the thing that people want is application software: programs to help them get what they want to do done, from editing documents to running a business to playing games to writing more software. Debian comes with over 51000 packages (precompiled software that is bundled up in a nice format for easy installation on your machine), a package manager (APT), and other utilities that make it possible to manage thousands of packages on thousands of computers as easily as installing a single application. All of it free.

It's a bit like a tower. At the base is the kernel. On top of that are all the basic tools. Next is all the software that you run on the computer. At the top of the

tower is Debian — carefully organizing and fitting everything so it all works together."

3.4.1.1.3. Parrot OS Features

Digital Forensics: supports “Forensic” boot option to shun boot automounts plus many more.

Anonymity: supports Anonsurf including anonymization of entire OS, TOR and I2P anonymous networks and beyond.

Cryptography: comes with custom built Anti Forensic tools, interfaces for GPG and cryptsetup. Additionally, it also supports encryption tools such as LUKS, Truecrypt and VeraCrypt.

Programming: braces FALCON (1.0) programming language, multiple compilers and debuggers and beyond.

It also supports development frameworks for embedded systems and many other amazing features.

Free (as in freedom): Feel free to get the system, share with anyone, read the source code and change it as you want! This system is made to respect your freedom, and it ever will be.

Lightweight: We care about resources consumption, and the system has proven to be extremely lightweight and run surprisingly fast even on very old hardware or with very limited resources.

3.5. Snort Rule

What is Snort?

Snort is an open source network intrusion detection system (NIDS) created by Martin Roesch. Snort is a packet sniffer that monitors network traffic in real time,

Scrutinizing each packet closely to detect a dangerous payload or suspicious anomalies.

Allows for monitoring of:

Local machine

Machines on your local network

Snort Rule

In the sniffer mode of operation, Snort will read network packets and just display them on the console. In packet logger mode, it will be able to log the packets to the disk. In network intrusion detection mode, Snort will monitor
the network traffic and analyse it based on the rules defined by the user.

Basic usage

snort -i <interface> -c <config file>

3.6. Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural. It also has a comprehensive standard library.

4. PRACTICAL IMPLEMENTATION

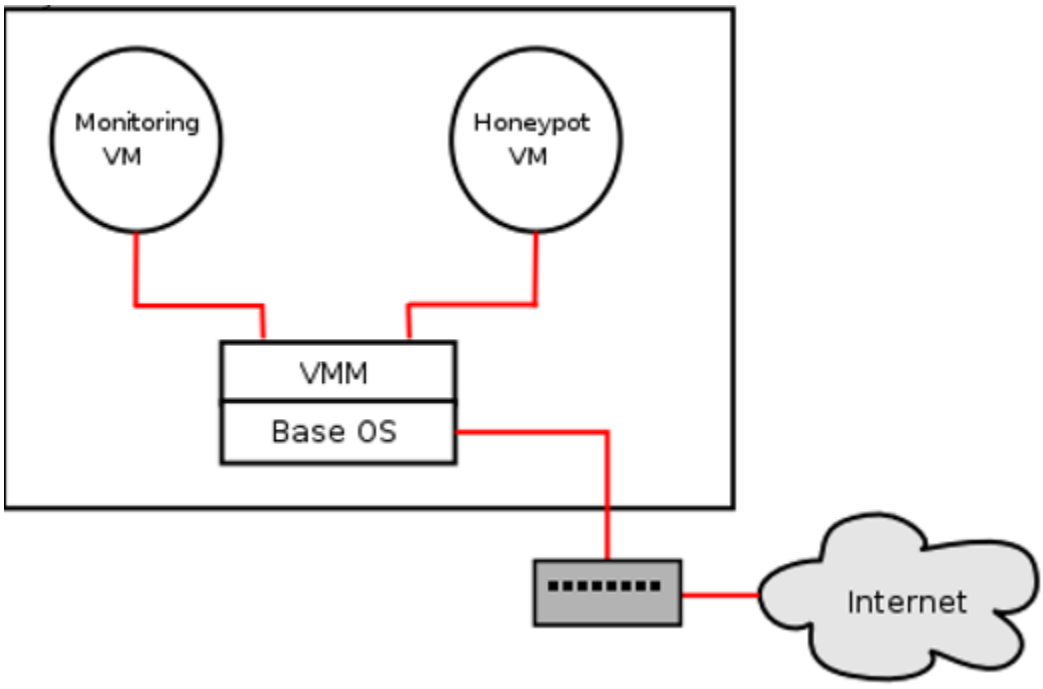
In practical implementation I will explain step by step to implement honeypot on your system. I will speak about our experiments and why we chose to deploy those specific products throughout the thesis. We will explain how it is working and come up with some results related to our findings.

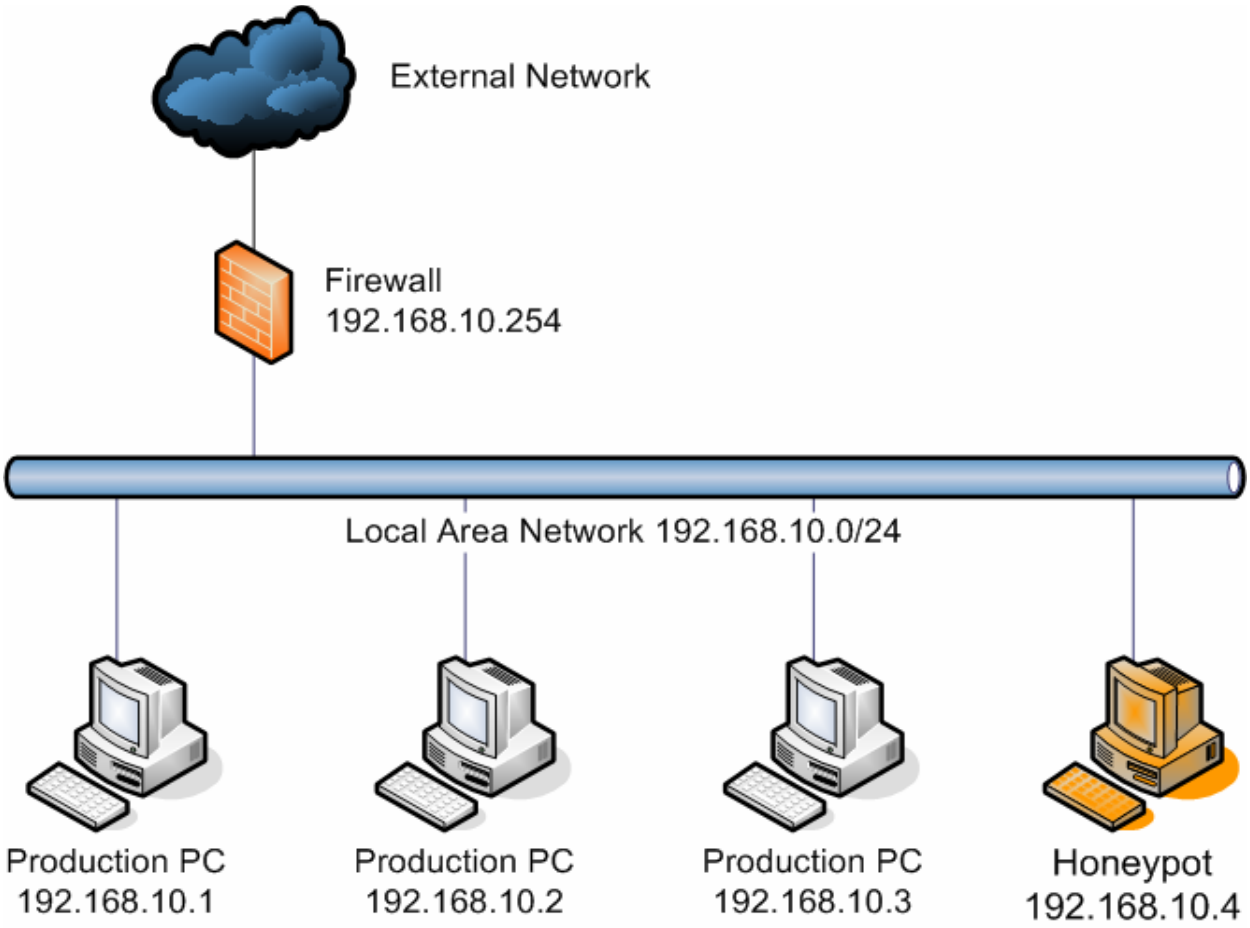
4.1. Starting to honeypots

In this project, I have installed honeypot in the local machine and the local machine will have connected to a particular network. Then I have assigned the local IP address to the honeypot and also assign a particular port number and given a MOTD after that honeypot will ready to run.

When the attacker click the IP address of the victim machine then the attacker system information will be coming to the hacker machine and save in the local machine .MMH format or save in secure manner.

In my Honeypot project, I have used snort rule for monitoring the attacker machine and collect the information of attacker machine.




External Network
Firewall
192.168.10.254
Local Area Network 192.168.10.0/24
Production PC
192.168.10.1
Production PC
192.168.10.2
Production PC
192.168.10.3
Honeypot
192.168.10.4

5. *CONCLUSION* & REFERENCE

CONCLUSION

Honeypot is not a solution to network security but a good tool supplements other security technologies to form an alternative active defense system for network security. Working with IDS and firewall, Honeypot provides new way to attacks prevention, detection and reaction. Honeypot can serve as a good deception tool for prevention of product system because of its ability of trapping attacker to a decoy system. Supplemented with IDS, honeypot reduces false positives and false negatives. This kinds of honeypot share the common technologies of data control and data capture. Honeypot easier to deploy and more difficult to detect.

The largest challenges facing the world today is to protecting the servers against the attackers, that is to provide the security to the network, this is done by honeypot indirectly .It provides the resources to gather information about the attacker, but it carries a lot of risk.

REFERENCE

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

Iyad Kuwatly, Malek Sraj, Zaid AI Masri, and Hassan Artail, “A Dynamic Honeypot Design for Intrusion Detection”, ©2004 IEEE.

Mueller Patrick and Shipley Greg, “Network Computing”, Aug 2001 http://www.networkcomputing.com/1217/1217f2.html.

Northcutt Stephen, “Network Intrusion Detection”: An Analysis Handbook. New Riders Publishing, 1999.