2003

A deception based framework for the application of deceptive countermeasures in 802.11b wireless networks

Suen Yek

USE OF THESIS

A Deception Based Framework for the Application of Deceptive Countermeasures in 802.11b Wireless Networks

A Thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Science Honours (Software Engineering)

School of Computer and Information Science
Edith Cowan University
Perth, Western Australia

Student:	Suen Yek
Student Number:	097 0588
Supervisor:	Dr. Craig Valli
Submission Date:	October 2003

ABSTRACT

The advance of 802.11b wireless networking has been beset by inherent and in-built security problems. Network security tools that are freely available may intercept network transmissions readily and stealthily, making organisations highly vulnerable to attack. Therefore, it is incumbent upon defending organisations to take initiative and implement proactive defences against common network attacks.

Deception is an essential element of effective security that has been widely used in networks to understand attack methods and intrusions. However, little thought has been given to the type and the effectiveness of the deception. Deceptions deployed in nature, the military and in cyberspace were investigated to provide an understanding of how deception may be used in network security. Deceptive network countermeasures and attacks may then be tested on a wireless honeypot as an investigation into the effectiveness of deceptions used in network security.

A structured framework, that describes the type of deception and its *modus operandi*, was utilised to deploy existing honeypot technologies for intrusion detection. Network countermeasures and attacks were mapped to deception types in the framework. This enabled the honeypot to appear as a realistic network and deceive targets in varying deceptive conditions. The investigation was to determine if particular deceptive countermeasures may reduce the effectiveness of particular attacks.

The effectiveness of deceptions was measured, and determined by the honeypot's ability to fool the attacking tools used. This was done using brute force network attacks on the wireless honeypot. The attack tools provided quantifiable forensic data from network sniffing, scans, and probes of the wireless honeypot. The aim was to deceive the attack tools into believing a wireless network existed, and contained vulnerabilities that may be further exploited by the naïve attacker.

The results indicated that the wireless honeypot was able to deceive wireless sniffing tools Kismet and Netstumbler (see section 2.12.1) into believing a wireless access point (AP) existed. However, network attacking tools NMAP and Nessus were not altogether deceived into believing a network of varying Operating System (OS) platforms existed within the wireless honeypot. The faked OS's on the wireless honeypot could not be guessed on all scans conducted by the attacking tools, which indicated that the deceptions deployed were not effective.

The implications of results demonstrated how deceptions might be used in network defence as a means to improve organisational network security against common network attacks. Additionally, the results indicated which areas of wireless network defence would need further investigation to determine a more effective use of deceptions.

DECLARATION

I certify that this thesis does not incorporate, without acknowledgement, any material previously submitted for a degree in any institution of higher education; and that to the best of my knowledge and belief, it does not contain any material previously written by another person except where due reference is made in the text.

Signature [redacted]

Date 12th January, 2004

ACKNOWLEDGEMENTS

I would like to take this opportunity to acknowledge and thank my supervisor, Doctor Craig Valli for his patience and valuable guidance throughout the year. I am also appreciative for the many research meetings conducted by Associate Professor William Hutchinson, who provided myself with constructive feedback and advice that facilitated the development of my research.

Furthermore, I would also like to thank the research candidates that participated in our research meetings for their assistance with suggestions, and enlightening me with their various research experiences.

Finally, I would like to express many thanks to my family. Especially to my Mother and Father, who gave me strength with words of encouragement and support, and who will continue to do so in my future endeavours.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1 CHAPTER 1 – INTRODUCTION

This research involves deployment of a wireless honeypot utilising a Honeyd honeypot and 802.11b wireless network technology. A honeypot is a security resource that is used for detecting and monitoring attacker behaviour in a network (Spitzner, 2002a). Honeyd is a type of honeypot that uses operating system (OS) and network emulation to appear real to an attacker. The Honeyd will be the primary source for testing network attacks in the honeypot for this research. The wireless capabilities that will be adopted for the wireless honeypot will be the IEEE 802.11b standard. This will allow the deployment of a wireless access point (AP) that will be used as the gateway entrance to the Honeyd virtual networks.

The intention of the wireless honeypot is to appear deceptively as a realistic network of wired and wireless integrated services for various OS platforms. Scanning of Internet Protocol (IP) address spaces and the probing of TCP/IP ports are popular methods of OS fingerprinting. Their aim is to identify the platform and version of the OS so that an attacker may discover a specific vulnerability to exploit.

The wireless honeypot is then an exploratory platform to investigate how deception may be utilised in wireless network defence. This encompasses observation of results from common attack tools used on the wireless honeypot. This will also allow the effectiveness of deceptions to be measured, and interpreted by the researcher.

A literature review of deceptive origins and implementation is presented to identify how deceptions may be deployed in network security. The various implementations of deceptions is researched from animals, the military, and from cyber exploit of deceptive capabilities. A framework for deception is developed to encompass the deceptive possibilities examined in the literature.

The framework for deception is used to map existing network countermeasures and attacks to a specific type of deception. This allows the researcher to determine how each deception may be deployed as a defence on the honeypot; and as a network attack method. A devised matrix for each countermeasure and attack aids in the identification of the conditions for deploying the deceptions in the form of deception in depth.

Deception in depth (DiD) encompasses rings of varying deceptive strength, deployed on the wireless honeypot. Each ring, moving outwards from the central core, abates in deceptive strength. The focal point of the DiD is the wireless honeypot. The second ring utilises Honeyd to create a virtual network topology containing numerous OS and web server platforms. The third ring employs FakeAP to generate false 802.11b network packets, and provide a subsequent entry-point to the Honeyd virtual networks.

Additionally, a Central Logging Structure (CLS) that encompasses Honeyd log-files and SNORT Intrusion Detection (IDS) logs records the network activity on Honeyd. The CLS runs concurrently with any attacks performed on the wireless honeypot and is thus part of the deceptive implementation.

Each ring in the DiD may be attacked using common network brute force tools. These tools include Kismet and Netstumbler that stealthily detect wireless access points. As well as Network Mapper (NMAP), that performs stealth TCP/IP port scanning; and Nessus, that forcibly probes and identifies system vulnerabilities.

Several attacking tools are freely available for download from the World Wide Web (WWW). The attacking tools NMAP and Nessus are popular choices for 'script kiddie' attackers (Honeynet Project, 2000; Spitzner, 2003). Script kiddies are amateur computer hackers that typically use tools that are easy to operate and require little human interaction.

There is a diverse range of resources available from the WWW that a script kiddie may use. Some examples of other wireless sniffing tools include Dsniff, WaveLAN, and AirMagnet. They are able to sniff out 802.11b access point signals and configurations such as the IP, MAC, SSID (see section 2.12.1), signal strength and channel of the AP, using a Linux or Windows machine.

System Administrator Tool for Analyzing Networks (SATAN), WebTrends Security Analyzer, and Argus, primarily used on Linux systems, are other examples of network security tools used for detecting and exploiting vulnerabilities.

The effectiveness of deceptions used for defence on the wireless honeypot will be evaluated from results gathered from the outputs of the attack tools used. Kismet and Netstumbler will indicate if a bogus AP may be identified through wireless sniffing of 802.11b packets. NMAP will scan IP blocks to enumerate OS platforms and running services on ports. Nessus-generated reports of attacks will indicate if a security warning or vulnerability may be detected on the selected IP addresses scanned and probed.

The Honeyd log-files and IDS files, recorded during the same attack, is used to triangulate the results of the attack tool Nessus and verify if the results are accurate. A discussion of the results will then be used as a basis to explore possible explanation and implications of the results ascertained from the wireless honeypot experiment.

The research aims are to investigate how deceptive strategies may be used for wireless network defence. The deployment of a wireless honeypot provides an experimental basis for testing deceptive countermeasures and attacks. The result of which may be beneficial for organisational understanding and implementation of deceptions for network security.

1.1 Significance of Research

Current studies and statistics show companies do not implement adequate security measures for protection of wireless networked resources (Barnes et al., 2002; Nanda, 2002; Webb, 2002). Furthermore, the nature of existing 802.11b (see 2.10.1) protocols subject sensitive information in wireless networks to remote attacks that may not even be detectable.

Results from a conventional wired honeypot experiment revealed deception to be a successful countermeasure to network threats generated by brute force attacks utilising vulnerability scanners (Gupta, 2002; Yek & Valli, 2002). The combination of wired and wireless services introduces new security issues. These hybrid networks may utilise similar deceptive countermeasures to successfully defend against malicious attacks (Nanda, 2002).

The primary purpose of the research is to ascertain the effectiveness of using deceptions on the wireless honeypot to counter common brute force network attacks. The wireless honeypot deployment aims to appear and perform as a wireless network. The implementation of the honeypot is strengthened through the development and application of a deceptive framework to investigate and apply deceptive, defensive, or offensive mechanisms. The framework for deception is utilised to determine if specific deceptions are more effective against particular network threats.

Investigation of deceptive implementations on the wireless honeypot aims to identify effective methods of deceptive network defence. Understanding how deceptions may be deployed, and what outcomes may be achieved, may be significant for devising better organisational defences. However, any small office - home office (SOHO) user or organisations implementing a wireless local area network (WLAN), may find the research significant for developing a defence for 802.11b insecurities.

1.2 Research Questions

The significance of the study has outlined particular focus areas that will be investigated in this research. A methodological formulation of the research questions will provide clear goals for the researcher. The research questions for the wireless honeypot experiment are:

1. Can a framework for deception be applied to common network countermeasures to reduce the effectiveness of attacks?
2. How effective is deception in a wireless honeypot against brute force attacks?
3. Under what conditions, and do conditions vary by the type of attack?

2 CHAPTER 2 – LITERATURE REVIEW

2.1 A Definition of Deception

Rue (1994, p.v) provides a definition of deception that is wide in scope, area, and application: "[Deception is] the problematic distinction between appearance and reality". However, a single statement may not convey all the fundamental characteristics of deception. Therefore, the nature and properties of deception may be described as:

> [To occur] when the designs embedded in the morphology [meaning the form or structure] and/or behaviours of one entity defeat the designs embedded in the perceptual structures and/or strategies of another entity (*ibid*, 1994).

It is essential to investigate the origins of deception and the diverse definitions that have since transpired. Several descriptions and classifications of deception have been observed and theorised by various authors that will be examined in the following literature review.

2.1.1 Animal deception

Studies by Caras (1972) and Hutchins (1980) show plant and animal organisms to be the earliest practitioners of deception for both defensive, and offensive means of survival. According to Caras "Evolutionists would insist that their colors and patterns are no accident; that there is some survival value in being marked the way they are" (1972, p.4).

Bowyer (1982) suggests deceptive behaviour and physical distortion in various species are determined by physical structure and genetic makeup that has emerged from a process of selection over the course of evolution, or possibly a single moment of mutation. Furthermore, Bowyer states:

> Any mutation that saves even one animal in 10,000 encounters of its species with predators will be preserved and will firmly establish itself in the species. It is these often almost imperceptible genetic advantages that explain some of the bizarre hiding and showing mutations that have been preserved and elaborated in nature (p.48).

Subsequently, anecdotal research of biological case studies facilitates understanding and classification of deception throughout evolution. The biological evidence and illustrative cases will be researched to determine deceptive origins, implementation, and the impact of external factors in order to devise a framework for deceptive pathways that network countermeasures and attacks may employ.

2.1.2 Taxonomy of deception based on biological case study

Bowyer (1982) stipulates two levels of deception, which are dissimulation - hiding the real, and simulation - showing the false. The levels of deception apply to psychological deception employed mainly by humans, but may also be mapped to those physical deceptions demonstrated in animals.

Through the structure of deception in Figure 2.1, Bowyer defines deception as "the advantageous distortion of perceived reality" (1982, p.47). Bowyer describes all deceptions to involve hiding, and is divided into three categories: masking, repackaging and dazzling. This is the first level of deception. A second level of hiding the real is showing the false: which is mimicking, inventing and decoying. According to Bowyer, all deceptions fall into either of these categories.

THE STRUCTURE OF DECEPTION
(With process defined)

Deception
(Distorting reality)

DISSIMULATION (Hiding the real)	**Level 1**	**SIMULATION** (Showing the false)	**Level 2**
MASKING Conceals one's own Matches another's	Characteristic spectrum	**MIMICKING** Copies another's characteristic spectrum	
REPACKAGING Adds new Subtracts old	Characteristic spectrum	**INVENTING** Creates new characteristic spectrum	
DAZZLING Obscures old Adds alternative	Characteristic spectrum	**DECOYING** Creates alternative characteristic Spectrum	

(Bowyer, 1982, p.61)

FIGURE 2. 1 The structure of deception

2.1.3 Biological examples of dissimulation - hiding the real

Masking

Masking is a form of hiding by blending into surroundings or seeking invisibility. The advantage of a masked deception allows an animal to hunt prey with as little possibility of being observed, or to integrate with the environment to conceal from potential predators. An example of masking is demonstrated by a Polar Bear's white fur that blends with their habitat of snow (Polar Bears International, 2002). The masking aids their invisibility to hunt prey such as seals.

Repackaging

Hiding the real in the form of repackaging different or new attributes is the intention of distorting perceptions through appearing dangerous, harmless, unobvious, or simply conveying a particular attitude. Chameleons utilise changes in skin pigmentation to repackage their appearances. This is done to communicate their willingness to mate (PBS, n.d.), and to hide from predators such as large birds (Carthy, 1972).

Dazzling

Dazzling is a mode of deception that is often employed when unsuccessful in masking or repackaging, and when the predator has recognised its prey. Hence dazzling is often a contingency manoeuvre to escape a predatory strike. Octopi utilise dazzling deceptions by expelling a smokescreen of black ink that aims to impair the predator's orientation, while the animal escapes danger.

2.1.4 Biological examples of simulation - showing the false

Mimicking

Mimicry is an "offensive deception" (Bowyer, 1982, p.50) employed by animals that actively impersonate another life form through faked behaviour or appearances, while concealing what is really their own. The mimicry should have bearing or connection to the environment, or the potential host for the true advantageous effect of the deceived target's distorted perception. The Cuckoo bird employs this type of deception by mimicking the colour of other birds' eggs. It then abandons its own offspring in their nests, to be fledged by a foreign parent.

Inventing

Inventing an alternative reality is a deception used by animals when hiding is not possible, and there is no capability to mimic. An animal that utilises inventing deceptions accessorises its physical structure with an attribute that appears to change the entire reality of its appearance or actions for an intended advantage; such as to gather food. A bioluminescent organ that protrudes from the usually dark coloured Deep Sea Anglerfish invents a new reality because only the Anglerfish's organ may be seen in the dark. This deception is used as bait to lure small fish to the Anglerfish for food.

Decoying

A decoyed deception is often used to distract predators from the discovered real. Mother ducks use themselves as decoys by feigning injury to predators such as foxes. This captures the attention of the predator and aims to deter them from the usually nearby ducklings.

Bowyer suggests masking, repackaging and dazzling as deceptive methods employed to hide real characteristics. Alternatively, mimicking, inventing and decoying are methods to hide real characteristics from view and weaken the predator's ability to identify and consequently hunt prey. These deceptions may be deployed in networks with the similar objective of hiding real assets, while impairing the attacker's ability to compromise.

2.1.5 Major types of deceptive effects sought

Similar to Bowyer's structure of deception, Gerwehr & Glenn (2003) specify three deceptive effects which are sought from a deception: masking, misdirecting and confusing. The deceptive 'effect' refers to a "specific type of disadvantageous misperception the deceiver is seeking to produce in the mind of the target" (*ibid*, p.36). This is described in Table 2.1.

Deceptive Effect	Definition	Common Examples
Masking	Concealing signal	Camouflage Concealment Commingling with non-combatants Signature reduction Reducing signals, ideally to the point of undetectability
Misdirecting or Misleading	Transmitting clear and unambiguous *false* signal	Feint/demonstration Decoy/dummy Disguise Disinformation Divert attention, resources away from

		real assets/activities
Confusing	Raising the "noise" level to create uncertainty, paralysis, degrade perceptual capabilities	Generating additional commotion, traffic, movement, etc. Shoot-and-scoot to disorient foes Purposeful departure from established pattern (conditioning/exploit) Randomisation Create 'noise', over saturation unpredictability, or the need for haste

(Gerwehr & Glenn, 2003, p.37)

TABLE 2. 1 Major types of deceptive effects sought

2.2 Evaluation of deceptive effects sought and the structure of deception

Gerwehr & Glenn's' major types of deceptive effects sought show that masking, misdirecting, and confusing effects also encompass the deceptive outcomes defined by Bowyer. A summary of deceptive effects sought may therefore be expressed in Table 2.2.

Deceptive Effect Sought	Other Distinguishing forms of the Deceptive Effect	Description of the Deception
Mask	Camouflage, Repackage	Attempting to blend with the environment without attracting attention.
Mislead	Misdirect, Mimicry, Decoy	Luring attention by creating a different scenario.
Confuse	Dazzle, Invent	Obscuring presence with exaggerated actions.

TABLE 2. 2 Summary of deceptive effects sought

Upon defining various forms of deceptions from biological instances, the purpose of executing a deception may be identified and further applied to any deceptive source.

Consequently, humans may implement deceptive effects for network security that may result as a mask, mimicry, or confusion.

2.3 Levels of Deceptive Sophistication

US National Security Research and Development organisation RAND, aims to improve policy and decision making through research and analysis of several governmental issues that include military deception. One focus area of RAND's research in deception is military adaptation from plant and animal deceptive models and precedence (RAND, 2003).

According to RAND (2001), and researchers Gerwehr and Anderson (2000) deception may be classified into four exclusive groups: static, dynamic, adaptive and premeditative. Each group is representative of a level of sophistication that an organism may exercise over the deception, as described in Table 2.3.

Category of Deception	**Description**
Static	The deception method is in place irrespective of state, activity, or history of either deceiver or target.
Dynamic	The deception method is employed by the deceiver when circumstances trigger it.
Adaptive	The deception method is triggered as in DYNAMIC above, but the method or triggering event may be modified by feedback (i.e., trial-and-error). Blending with each unique environment or adapting to each situation.
Premeditative	Designed and implemented based on experience, knowledge of friendly capabilities and vulnerabilities, and moreover, observations about the target's sensors and search strategies.

(Gerwehr & Anderson, 2000, p.3; RAND, 2001)

TABLE 2. 3 Deceptive levels of sophistication

2.3.1 Biological examples of deceptive levels of sophistication

Static

The colours of a Viceroy butterfly are an imitation of the poisonous Monarch butterfly. The Viceroy's colours are static, of which the insect does not harness any control (Ivyhall, n.d.). Thus, the deception remains unchanged regardless of the circumstance.

Dynamic

A dynamic deception is displayed through the Walking Stick insect's ability to stiffen its body, and change colour to it's surroundings when triggered by fear (Caras, 1972).

Adaptive

The Lacewing larvae rob the wax from an aphid's body to transfer onto their own, as a protection and disguise from predators. This deception is employed upon interaction with the Lacewing larvae's predator, and surrounding environment (Gerwehr & Glenn, 2003).

Premeditative

A premeditative deception is the most advanced level of sophistication. It involves prior assessment of the circumstance, and results in a cognitive decision. Dolphins and humans are characteristic of this level of sophistication, and thus have the ability to implement a premeditative deception (*ibid*, 2003).

2.4 Defensive and Offensive Induced Deceptions

RAND Researchers Gerwehre & Glenn reason that a defensive or offensive deception is "among one of the best methods for both successfully preying and escaping predation" (2000, p.16). This assertion holds merit, given that predators often utilise offensive, deceptive resources to hunt prey. Additionally prey will seek defensive, deceptive measures to evade attack.

2.5 Summary of Biological Deceptions

The biological case studies discussed reinforce Bowyer's (1982) theory of the wide variety of deceptive applications designed to suit each organism's available resources. The classifications identified (Gerwehr & Anderson, 2000; Gerwehr & Glenn, 2003; RAND, 2001) support nature's almost infinite selection of deceptive applications. Whether fixed or flexible, physically dependent, or behaviourally induced, there are "many deceptive methods that may accomplish similar ends" (Gerwehr & Glenn, 2003, p.12) This may be for both defensive and offensive purposes.

Deceptions thus implicitly encompass a level of complexity that the deceptive effect may be exercised through. Humans however, are able to select which deceptions are appropriate to apply to the circumstance. Therefore, implementing a network based deception may require varying levels of sophistication to respond to the shifting intensity of attack capabilities; such as the behaviour of many modern day interactive attacking tools.

2.6 External Factors

Deceptive 'conditions' encompass a number of external factors that include the predator and prey (Bowyer, 1982). The implications of external factors such as the immediate environment, knowledge of the adversary, recent history, preconceptions, and warning also have significant influence upon any deception of the prey and predator (Gerwehr & Glenn, 2003).

In The Art of Darkness: Deception and Urban Operation, Gerwehr and Glenn (2000) argue that in-depth operational deception should adapt to the urban environment through synergistic strategies that are aimed to reduce opponent strength, and expose weakness. Additionally, the nature of the urban terrain, whether friendly or dangerous, may affect the forces and processes of the battle where "Something similar might be said of deception" (*ibid*, p.iii).

Gerwehr and Glenn (2000) infer that an effective, operational deception should consider the combination of social and cultural conditions, physical infrastructures, and all other unique characteristics of the surroundings. An integration of this knowledge facilitates informed judgements, and decisions about deceptive actions. Consequently, circumstance becomes a fundamental element of the deception. Further research (*ibid*, 2003) has shown effective use of deception in urban conflict to be valuable for offensive deception, defensive deception, and intelligence gathering purposes.

2.6.1 The adversary and perception

The adversary, who is the intended target of the deception, is able to utilise its own distinctive recognition and interpretation of sensory stimuli (eyes, ears, smell etc) that is employed in a strategy of perception. This may include inch-by-inch scrutiny, quick scans, random walks, or spiral searches (Gerwehr & Glenn, 2003; RAND, 2001).

The assessment or perception of the adversary is formulated by insight, intuition, or knowledge of the prey, which prompts an action. Thus, the deceiver must be aware of such capabilities to be able to implement a deception that contends with those adversarial tools of perception. Humans are akin to animals in making decisions that affect survival (Gerwehr & Glenn, 2003).

> Decision makers rely upon their assessment of other actors' interests, intentions, and capabilities, as well as an assessment of the environment or context within which the action takes place... [Hence,] It is incumbent upon decision makers to form accurate perceptions (*ibid*, 2000, p.17).

Implications of the adversary, referred to as the 'attacker' (see section 2.9), in network security are similar to biological cases. The attacker's ability is considered when deploying the deception. If the attacker is unskilled, such as a script kiddie (see section 2.9.1), then the victim needs to become aware of the attacking tools and methods typically adopted by a script kiddie. Therefore, this research will adapt deceptive countermeasures that are able to contend with this attacker's common choice of method and skill.

2.7 Military and Human Adaptation of Deception

Gerwehr and Glenn (2003) state:

> The unforgiving nature of natural selection, combined with a truly staggering prevalence of deception, strongly supports the argument that causing an adversary's perception to be inaccurate (i.e. degrading their situational awareness) is of enormous value in virtually any setting or type of conflict (p.12).

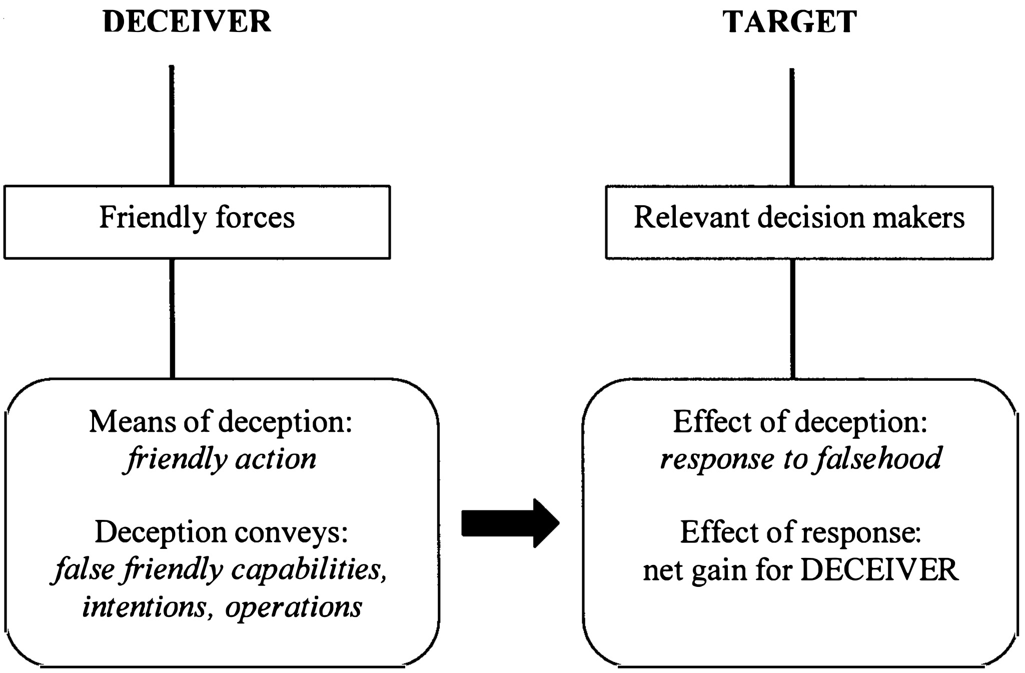
In this instance, the inaccuracy of perception is the result of deceptive capability. Defence and intelligence communities recognise the significance of biological deception throughout evolution. Subsequently, tactical planning of offensive and defensive strategies have been derived from animal deception (2000). RAND (2001) further supports mapping of animal biology to the military domain as highly useful for gaining analytical perspectives.

2.7.1 Military Definition of Deception

A definition of deception by the Joint Publication 3-58: Joint Doctrine for Military Deception cited in Gerwehr & Glenn (2003) states deception:

> [To be defined] as those actions executed to deliberately mislead relevant decision makers as to friendly military capabilities,

> intentions, and operations, thereby causing the relevant decision maker to take specific actions that will contribute to the accomplishment of the friendly mission (p.15).



(Gerwehr & Glenn, 2003, p.15)

FIGURE 2. 2 Deception: The joint definition

The military definition illustrated in Figure 2.2 is depicted as a relationship that involves the deceiver utilising a means of deception to convey a false message to the reciprocating target. According to Gerwehr & Glenn (2003), the response to falsehood as an effect of deception is a "deliberately induced misperception" (p.15) resulting in an outcome as an advantageous gain for the deceiver.

The College of Aerospace Doctrine, Research and Education (CADRE) publish the Air and Space Power Mentoring Guide, in which the essay Principles of War emphatically states "The principle of the offensive suggests that offensive action, or maintaining the initiative, is the most effective and decisive way to pursue and to attain a clearly defined goal" (1997, p.60).

The essay then further explains premeditated defence as the primary strategy in war. An offensive posture is empowered by resourceful and proactive conduct aimed to achieve results and to subdue the adversary. A defensive posture should only be a transient interval until an offensive posture may be restored. “No matter what the level of war, the side that retains the initiative through offensive action forces the enemy to react rather than to act” (*ibid*, 1997, p.60). The paper firmly maintains that an offensive stance is indispensable to ensure triumph over the adversary.

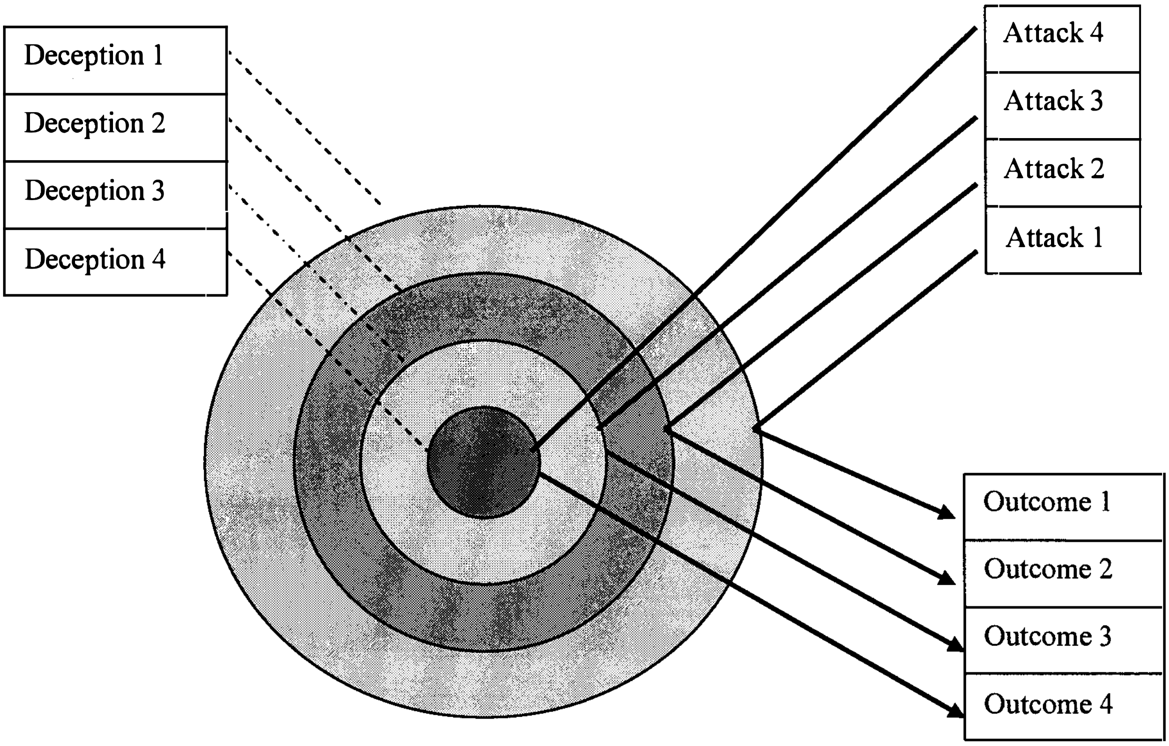
The paper also takes into consideration active and passive states within defence operations. This implies that defence may strategically utilise proactive execution, or yield submissive responses.

In a proactive situation, “deception is also a valuable mechanism of intelligence-gathering” (Gerwehr & Anderson, 2000, p.2). Despite complications in managing deception, “there is little doubt that when employed successfully, deception is among the most powerful instruments of conflict” (*ibid*, p.2).

> Moreover, deception is not a single tool: it is a diverse array of measures, which may be employed individually or in depth, alone or in concert with more traditional defensive measures, as simple schemes or complex ruses... There may thus be a synergistic effect in the use of deception alongside other defensive measures (*ibid*, p.2).

Figure 2.3 illustrates deception in depth utilising layered rings that express an array of deceptive measures that aim to strengthen the defence. Deception in depth demonstrates separate rings of deception and a reflected outcome for an attack on each ring. The central core embraces the most effective deception and is thus positioned closest to the protected asset. As the rings progress outward, the strength of the deception abates, where the peripheral is the most vulnerable of rings.

The rings of deception are implemented with careful consideration of the adversary. Different adversaries penetrate, and fall prey to deceptions in numerous ways depending upon their knowledge, experience, capabilities, determination, and resources (*ibid*, 2000).



(Gerwehr & Anderson, 2000, p.2)

FIGURE 2. 3 Deception in depth

2.8 Evolving Deception in Cyberspace

Humans are excellent wielders of deception, even within cyberspace. This is supported by the many deceptions implemented by humans including propaganda, spamming, spoofing, viruses, steganography, virtual reality, encryption, and lying (Hutchinson & Warren, 2002). Hence, the virtual world of anonymity yields almost unrecognisable, yet successful deceptions.

However, "as humans become almost totally dependent on digital data for their personal operational lives the consequences of deception increase exponentially" (*ibid*, 2002, p.5). Consequently, there imparts a necessity to protect one's own assets whether it is a person's identity, or data, in cyberspace. This may be achieved through employing deceptive strategies learned from the observation of deceptions used in the animal kingdom, and the military.

2.9 Identifying the Enemy in Network Security

In animal deception, the enemy of the prey is the adversary. In network deception and defence, the enemy has many names such as ‘hacker’ or ‘cracker’. Computer hackers may be defined as “an individual who experiments with the limitations of systems for intellectual curiosity or sheer pleasure” (Schneier, 2000, p.43). Ethical computer hackers aim to improve network security by breaking-in through network penetration testing (Hartley, 2003). Hackers, who penetrate computer systems for disruptive reasons, are branded as crackers. In this research, the computer cracker is the enemy, and will be referred to as the ‘attacker’.

Attackers use network penetration techniques to find vulnerable services running on particular TCP/IP ports of a computer, so that they may compromise the system. Any response from a TCP/IP port indicates that the particular port is active, and running a service that is potentially exploitable by an attacker. OS detection through TCP/IP fingerprinting determines the version number, and platform of the OS, so that a particular vulnerability of the service on the TCP/IP port may be identified for that version.

Network scanning of whole IP blocks or addresses, may be performed by automated tools such as NMAP, which stands for ‘Network Mapper’ (Fyodor, 2003b). NMAP determines OS fingerprints through a series of TCP/IP handshakes on TCP/IP ports that make the target respond in a particular way.

Automated tools such as Nessus (Deraison, 2003a) perform OS detection by probing specified targets through banner grabbing and port scanning. Nessus performs vulnerability testing by directing known attacks on a service that is found running on a TCP/IP port.

OS fingerprinting is “extremely valuable” (Insecure.org, 2003, ¶2) for exploiting computer vulnerabilities. This is because an attacker may then modify and tailor an attack to the particular type of OS the victim is using, and thus achieving a greater disruptive outcome. Therefore, understanding the methods used for remote OS detection are a necessity for testing and improving network defence (Insecure.org, 2003).

2.9.1 Script kiddies

The term ‘script kiddie’ is a derogatory term for the unsophisticated, though highly dangerous, attacker (Search Security, 2003; Spitzner, 2002a, 2002b). Script kiddies are typically immature attackers that use crude methods of compromise. They predominantly aim to penetrate a system and gain root (highest administrative level) access (Honeynet Project, 2000). A common technique is launching a buffer overflow attack that may be targeted to a discovered OS vulnerability, enumerated through port scanning.

A common *modus operandi* undertaken by a script kiddie is to scan random networks to find a target. Once found, the target is exploited using automated tools that require little understanding of their technical functionality (Honeynet Project, 2000; Spitzner, 2003). These automated tools typically include NMAP and Nessus, among other network scanning tools. Some script kiddies aim for the quantity of attacks achieved (Search Security, 2003). This type of behaviour may multiply the threat for damage that is caused by an unskilled attacker.

Although there are no definitive figures representing the number of script kiddie attacks, Chamales and Klinger from the IEEE (see section 2.10.1) communications society rationalise that as much as 95% of the attacking community are in the amateur script kiddie category (2003, p.3). Additionally, Spitzner (2002b), and Cohen (1999) also support that a vast number of attacks are executed by script kiddies. Consequently, the concern for potential damage from a script kiddie is emphasised, due to their numbers and their use of automated tools.

Therefore, the primary attack methods employed in this research will be based on tools typically used by a script kiddie. This includes the network security tools NMAP and Nessus. This will allow the researcher to investigate results of the attack tools used that would be relevant and meaningful to organisations defending against script kiddie attacks.

2.10 Comparing Wired and Wireless Networks

2.10.1 Wireless Network Protocols and Origin

Traditional wired networks have utilised cables and Ethernet connections for communication of physically linked devices. Typical entry to the World Wide Web (WWW) and private networks are via a discrete route. The arrival of wireless networks encompasses a new-network structure that replaces wired media with high frequency electromagnetic waves for data transmission across an air space.

In 1980 The Institute of Electrical and Electronic Engineers (IEEE) released the 802.11x suite of standards that specify the frequency range for wireless data communications, and is currently the most prevalent standard for implementing wireless local area networks (WLANs) (Montcalm, 2002).

The IEEE 802.11x suite currently comprises the 802.11a and 802.11b standards. However, standards for 802.11g and 802.11i products, with more security, are under development. The IEEE specifies the frequency range 2.4GHz in the Industrial, Scientific, and Medical (ISM) band for the use of 802.11b compliant devices. At present, 802.11b compliant products, also known as Wi-Fi (Wireless Fidelity) products, dominate the wireless market (Schoeneck, 2003).

The 2.4GHz band is shared by 802.11b (Wi-Fi) devices, microwave ovens, various cordless phones, and some fluorescent lights. However, one major disruptor of 802.11b are Bluetooth enabled devices. Bluetooth is the name given to a technology that uses small chips to connect short-range radio links that also utilises the 2.4GHz, ISM band. Several other wireless standards that may prevail in future are HiperLan-2 and 5-Unified Protocol (5-UP) (Nanda, 2002).

The deployment of WLANs differs largely from their wired counterpart's, mainly to accommodate user mobility of wireless communications. The principal devices required for wireless network communications are access points (AP), an antenna, and wireless enabled clients.

APs plug into a power supply and connect to the wired LAN to function as a root (central connecting) AP, or bridge connecting services to a wireless client. APs broadcast beacons with signal strength and data capabilities depending on the type, and strength of the antenna attached to the AP. Wireless clients such as laptops, or Personal Digital Assistants (PDAs) affixed with an antenna, and suitable network hardware, may then communicate with the AP through wireless packet exchange in the form of beacons.

Users may gain access to wireless networks provided the wireless client is within the coverage area surrounding the AP, and the signal is strong enough to support communication. Wireless connectivity generally ranges from 10-300m from the beaconing AP, and with connection of a antenna this range may extend to 24kms (Barnes et al., 2002). However, this is largely dependent on the signal strength of the antenna.

2.10.2 Future corporate wireless environment

According to Barnes et al. (2002), the corporate wireless environment will revolve around three predominant application solutions: mobile messaging, mobile office/corporate groupware and telepresence. These are described below with explanation of some possible security drawbacks.

Mobile messaging involves the extension of electronic mail to any wireless user located within the internal corporate (wireless) messaging network environment. Consequently, unintended wireless recipients may intercept confidential mail.

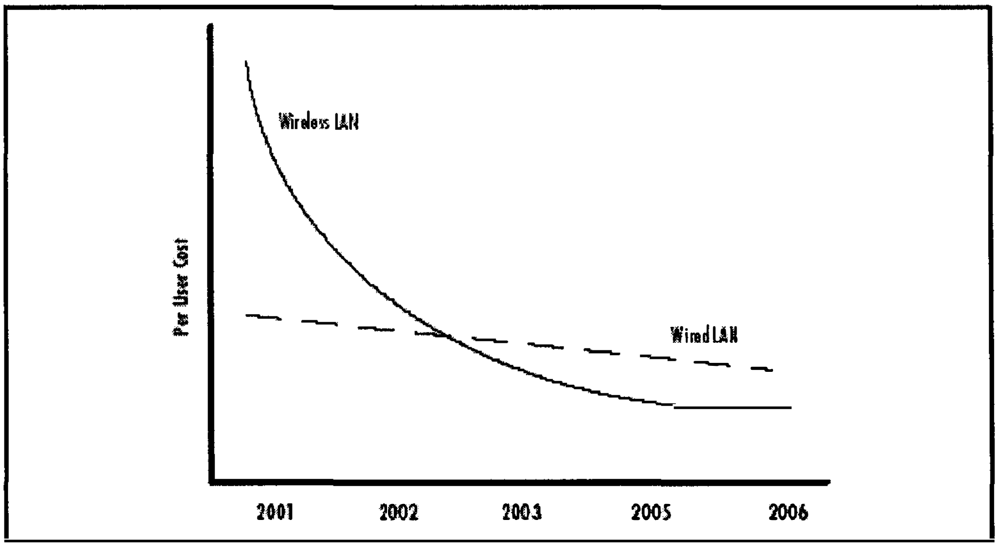
Outside wireless clients will also request equal access to the restricted corporate services. These include the database servers, application servers, information and news servers, directory services, travel and expense services, file synchronisations, Intranet server browsing, and file transfers services. Therefore, providing the remote user with the same localised access to the corporate wireless resources also opens doorways for unintended recipients to retrieve confidential data.

Further to the wireless mobile office, increasingly integrated wireless-networking technologies could unveil a revolution of interaction and communication between people and data stores (*ibid*, 2002). The potential for universally accepted and implicitly trusted wireless solutions, used in almost every context, is an optimism held by many. However, the studies conducted by the Gartner Group also highlight the growing concern of exploit of the corporate wireless environment:

- By the end of 2003, nearly 35% - 40% of cellular-based wireless traffic will be data
- By 2005, 50% of Fortune 100 companies will have deployed wireless LANs (0.7 probability)
- By 2010, the majority of Fortune 2000 companies will have deployed wireless LANs (0.6 probability)

(*ibid*, 2002, p.4)

Additionally, the anticipated future wireless cost trends, illustrated in Figure 2.4 indicate rapid decreases in the cost per user in wireless LANs, which is highly contrasted to marginal drops in wired LANs. Thus, the potential for wireless exploit may also increase with the growth of WLAN adoption.



(Barnes et al., 2002, p.23)

FIGURE 2. 4 Anticipated wireless cost trends

Though WLANs are not likely to completely substitute conventional wired LANs (Nanda, 2002), WLAN solutions provide a means for wireless clients to interface with wired LAN resources. The resulting hybrid networks allow interaction between wired and wireless clients, currently utilised effectively in several academic, corporate and home environments (Barnes et al., 2002; Nanda, 2002).

2.11 Reasons for Adoption of Wireless

“Wireless and mobile technologies are vital for the real-time enterprise” (Dulany, 2002, ¶3). Accompanied by exempt licence costs for bandwidth use in the wireless frequency bands, the exponential growth (Barnes et al., 2002; Trilling, 2003) of WLANs may be attributable to many factors (Barnes et al., 2002; Dulany, 2002; Nanda, 2002) that are described below.

With the absence of wires, WLANs are cheaper and often more convenient to implement. Limitations of fixed network APs are alleviated as wireless network expansion and upgrades are more easily accomplished than their wired equivalent.

The evolution of more powerful and compact wireless network components will see continued demand of more tightly integrated application support, and communication environments by consumers (Barnes et al., 2002, p.4). These will support greater access speeds, communication capabilities, and versatility of portable information appliances. Wireless information resources create greater portability through increased roaming capabilities, and facilitate increased work production with current throughput of 11Mbps and emergent standards such as 802.11g allowing 54Mbps.

"Today's wireless solutions offer flexibility, performance, and proven solutions that promise increased productivity and potential reductions of long-term capital and management costs associated with network deployments" (*ibid*, 2002, p.9).

Thus companies capitalising on the benefits of wireless mobility, in combination with increased development of wireless applications, assume significant risks pertaining to security and reliability, and the increased potential for malicious client activity (Barnes et al., 2002; Nanda, 2002; Webb, 2002; Wright, 2003).

2.12 Weakness and Threats to Wireless

A number of studies (Barnes et al., 2002; Liu, 2003; Nanda, 2002; Trilling, 2003; Webb, 2002; Wright, 2003) show that many companies lack properly implemented security over 802.11b wireless networks. Many physical aspects of wireless technology and the infrastructure used, in addition to exposed encryption, and authentication flaws further augment the threat to wireless networks (Schoeneck, 2003). The following is a description of various wireless security weaknesses that may be considered.

2.12.1 Discovering wireless network vulnerabilities

"Internet protocols are publicly posted for scrutiny by the entire Internet community" (Pfleeger & Pfleeger, 2003, p.403). Similarly Nanda (2002) states "of late there have been several articles in the press regarding weaknesses in 802.11 WLANs." Therefore, weaknesses may be rapidly discovered and published on the WWW and any person may utilise such knowledge to seek out, and infiltrate vulnerable networks. For every 1 million systems an attacker scans, 10 000 may be compromised (Spitzner, 2003). Randomness of attacking tools and the attacker's ability to relentlessly change and improve have become dangerously threatening to wired and wireless networks alike (*ibid*, 2003, p.29).

Spitzner claims declined network security is largely attributed to the availability, and growing trend of more powerful, fully automated attacking tools (2003). This is because script kiddies or unskilled attackers may easily operate many of these attacking tools. They may also be retrieved easily from the WWW and yield little understanding of how the software functions.

Wardriving

The term 'wardriving' is modern day network reconnaissance utilised by experts and novices alike to discover wireless networks (Montcalm, 2002; Nanda, 2002). Wardriving exposes WLANs usually by driving by metropolitan areas. This may be done with approximate speeds of up to 90km an hour, and by operating a wireless client such as a laptop or PDA.

Conventional wired network scanning is performed by enumerating TCP/IP ports and similarly, the same criteria may be applied to wardriving (Montcalm, 2002). Additionally, stealth scanning of wireless traffic may be achieved by using packet-sniffing software such as Kismet (Kershaw, 2003), Netstumbler (Milner, 2002) and Ethereal (Combs, 2003) to capture wireless traffic.

Nanda (2002) surmised that in 60% of cases, "security has proven to be absolutely none". In Manhattan, the Bay Area, and New England over 1000 WLANs have been exposed open to intruders (2002). Studies conducted by Webb (2002) revealed up to 50% of Perth city WLANs deployed were significantly insecure. It was also found that in many these cases, management were unaware of the security consequences (*ibid*, 2002).

Additionally, access to networks as far as 40kms away may be achieved by using powerful antennas such as yagi, which is not overly expensive (Nanda, 2002). This is indicative of the easy ability to detect distant WLANs and execute attacks remotely.

Compromising a WLAN may be achieved by disrupting signals through a 'man in the middle' (M-in-M) attack. This is where the attacker is positioned somewhere between the AP and transmitting range of intended wireless clients. Intercepted and disrupted signals from a M-in-M may relay false information, or devices may be burned out, or damaged. Additionally WLANs offering Dynamic Host Configuration Protocol (DHCP) allow use of the network connection by rogue wireless clients (Pfleeger & Pfleeger, 2003).

Access Point (AP) structure

One fundamental wireless security concern is the nature of the existing wireless setup. APs may be typically positioned in locations that are ideal for maximised data transmission, and user mobility. However, because APs may be probed by any client within reach of the network's electromagnetic radio frequency (RF) range, this creates a physically unbounded entry point that is attractive for launching attacks (Liu, 2003). Additionally, WLAN APs that are deployed behind conventional firewalls are more appealing for launching attacks on the internal network (*ibid*, 2003).

Radio Frequency (RF) deviation

Data communication is achieved by electrical signals transmitted over the network medium. Traditional wired networks bind data signals into the physical confines of copper or optic cables, as example. Thus, data interception on the cable requires access to the company router, and is restricted to the physical limitations of the cable.

Wireless data communications transmit electrical signals via RF waves, which freely traverse the open space to any node within a connectable distance from the transmitting AP. In contrast to controlled wired perimeters, RF signals are able to penetrate through many furnishings, floors, ceilings, walls, and even reach outside areas of the transmitting building. This may result in RF data transmission to unsolicited recipients in any location accessible by the RF waves (Nanda, 2002).

The Wired Equivalent Protocol (WEP)

Maintaining authenticity, integrity, and availability is another key security concern in WLANs. The IEEE 802.11b standard specifies two types of authentication methods required for participation in a WLAN: open key and shared key. Both may optionally utilise wired equivalent privacy (WEP) encryption. Each client is set to the corresponding authentication method of the AP it wants to associate with.

Open key authentication is the default method, which allows a client to associate with the AP without necessarily supplying the correct WEP key, and performs the entire authentication in clear text. Similarly, in shared key authentication the AP sends a clear text challenge passage that the client then returns encrypted with the correct WEP key. Both are insecure (Fennelly, 2001; Nanda, 2002) as an intruder may either attain the plain text challenge, and the encrypted text to decipher the key in the case of shared key; or illegitimately authenticate to an AP, sidestepping the need for a WEP key in the case of open key.

WEP uses Ron's Code 4, known as the RC4 symmetric stream cipher that supports a variable length key of 64 bits. The RC4 algorithm was invented by Ron Rivest of RSA (Rivest, Shamir and Adleman) Security Inc, and is used as the standard 802.11b WLAN encryption protocol. Symmetric keys use the same key and algorithm for both the encryption and decryption of data. The original design goals of WEP were:

- To prevent unauthorised users lacking the correct WEP key from gaining access control to the network
- To protect WLAN data streams by encrypting them and allowing decryption only by users with the correct WEP key

(Nanda, 2002, p.2)

The WEP protocol is widely known to be insecure, and is publicly posted on the WWW to be "riddled with architectural flaws" (*ibid*, 2002, p.4). Researchers Fluhrer, Mantin, and Shamir published the paper Weaknesses in the Key Scheduling Algorithm of RC4 in which the researchers confirm RC4 is "completely insecure in a common mode of operation which is used in the widely deployed [WEP] protocol" (Fluher, Mantin, & Shamir, 2001, ¶1).

WEP utilises an initialisation vector (IV) that is generated every time a packet is sent or received via the Wi-Fi client. The IV has a maximum of 2 to the 24 bits long, starting from 0, and incrementing by a value of 1 each time. Thus when the maximum of 2 to the 24 IVs is reached, it will have to restart at 0 (Fluher et al., 2001).

Referred to as the FMS attack, the paper explains how knowledge of the IV and the first output byte reveal information about the key bytes. Subsequently, decryption of captured packets became much easier after the paper was released (Nanda, 2002).

Furthermore, Pfleeger and Pfleeger (2003) state that the likely cause for deficient use of the WEP protocol is largely due to administrative difficulties in the configuration and management of encryption. Additionally, surveys reveal that WEP has been disabled in up to 85% of wireless installations. Pfleeger and Pfleeger also state that "even when encryption is used, the design of the encryption solution sometimes makes it easy to crack" (2003, p.401).

Service Set Identifier (SSID)

All packets sent by APs and WLAN clients contain the Service Set Identifier (SSID), which is a rudimentary naming scheme that functions to logically segment networks, and manage access control. SSIDs are not typically used as a network securing mechanism, and should not be as APs are, by default, set to broadcast their SSID in all beacons. An SSID may be guessed easily because they are often unassigned or set to manufacturer default values (Nanda, 2002). An intruder could therefore ascertain company SSIDs via social engineering means, or simply sniffing packets and identifying the SSID in the packet payload, as it is often not encrypted.

Internet Protocol (IP) address spoofing

Similar to wired networks, wireless networks are also vulnerable to IP address spoofing attacks where an attacker sends packets to the destination from an arbitrary source IP address. Response packets are sent to the spoofed IP address and the identity of the real provoking IP is never disclosed. This method of attack may be exploited in cases where the perpetrator wishes to send numerous probing packets but does not wish to raise suspicion by disclosing the solitary perpetrating IP address. Therefore, the victim's intrusion detection or packet sniffing tools may fail to detect a trend in port scans as malicious client activity.

Media Access Control (MAC) address spoofing

The Media Access Control (MAC) address is a physical, network identifier number allocated to hardware vendors for installation onto wired and wireless Network Interface Cards (NIC). All genuine MAC addresses are globally unique for each LAN based device and may be used for authentication for granting users various levels of network and system privileges. 802.11b wireless networks also utilise MAC addresses for client tracking and authentication.

In nearly all 802.11b wireless NICs, the MAC address value may be modified to a random number using vendor-supplied drivers, open-source drivers or various application programming frameworks (Wright, 2003).

Amongst several publicly posted articles on the WWW, Wright (2003) demonstrates the ease of changing a MAC address using the *ifconfig* command, or by executing a short C program using Linux open source drivers. Alternatively, the applet in the network control panel of a Windows OS may also permit changes to the MAC address properties.

Depending on their skill level, an attacker may spoof a MAC address to masquerade or hide their presence on the network. Or, they may falsely appear to be a valid MAC address that is authorised by the network and AP, and consequently circumvent access control lists or escalate network privileges (*ibid*, 2003).

Obfuscating network presence may be utilised to launch brute force attacks on a system by generating random MAC addresses for malicious packets. Though this is often used to evade intrusion detection systems, a honeypot is able to overcome the failed IDS detection, as all packets are rendered suspicious.

An attacker may bypass access control lists by obtaining a registered network MAC address simply by passively monitoring network traffic. In addition to gathering a valid list of MAC addresses from packet headers, which are broadcast in the open when wireless clients communicate with APs.

Because MAC addresses are constantly broadcast in plain text in the header of wireless packets, the crude acquisition and manipulation of MAC data is hence far more common than on regular wired networks (Nanda, 2002; Wright, 2003). Successful MAC address spoofs may grant the intruder unauthorised access to control mechanisms that provide launching points for Denial of Service (DoS) attacks, and advertise fallacious services to wireless clients (Wright, 2003). Thus, the propensity for significant WLAN network destruction through MAC address spoofing is high.

2.12.2 Penetration testing of networks and OS fingerprinting

OS fingerprinting is the technique for distinguishing the operating system of a host through its network stack (layer 3 of the OSI model). Typical OS fingerprinting tools probe for the known differing characteristics among OS's through identifying features found in the probes of open TCP/IP ports (Beck, 2001; Fyodor, 2003a).

OS fingerprinting tools such as NMAP and Xprobe (Yarochkin & Arkin, 2003) are designed to connect with the network layer, layer 3 of the Open Systems Interconnect model (OSI), and therefore communicate using a sequence of TCP/IP handshakes each time a connection is attempted on a port. Each OS's TCP/IP stack responds to a handshake in its own unique way, which is how NMAP uses OS fingerprinting to identify a particular OS.

OS fingerprinting is an effective technique for enumerating a network as it gives insight into the specific OS platform and version number. Once the OS architecture is identified, a vulnerability scanner such as Nessus may be used to exploit any OS weaknesses. The Nessus network vulnerability scanner will forcibly probe each TCP/IP port on an OS for any known security weakness, and report findings to the Nessus client (a Graphical User Interface). An unsophisticated attacker would then investigate on the WWW a method to exploit the OS weaknesses found by Nessus, with buffer overflow attacks.

2.13 Deception as a Network Countermeasure

For the scope of this research, networks aim to defend against script kiddies, as they are the primary attackers. Countering network tools that these attackers use in the electronic environment requires a deceptive system that is able to mimic actual systems and networks. Thus, a network that acts real, aims to distort appearances by hiding the real assets, and showing false values.

One of the first publications of organisational, electronic adaptation of deception is Bill Cheswick's An Evening with Berferd: In Which a Cracker is Lured, Endured, and Studied (1992). Cheswick described how deploying deceptive strategies in a faked networked environment was valuable in learning the tactics and location of the attacker, and eventually reported the attacker to authorities.

2.13.1 The Deception ToolKit (DTK)

In 1997, Fred Cohen released the first open source honeypot solution known as the Deception ToolKit (DTK) on the WWW. Cohen's DTK is an effective tool for countering attacks by enabling customisable PERL script files to simulate behaviours of existing OS's. The system appears populated with known vulnerabilities that may be exploitable by attackers.

The DTK gives the victim the advantage of early warning of an intrusion or attack, while the attacker consumes time and effort to penetrate the deceptive OS's. The gathered evidence of all attacker activity is recorded in the DTK's logfiles. Therefore, tracking the attacker's activities allows the victim to identify the vulnerability the attacker is targeting, and the tools that are being used. The victim may then respond to the attacks with necessary actions. These actions may include disabling or patching vulnerable services, and notifying the appropriate authorities.

Cohen (1999) rationalises that increased organisational use and acceptance of the DTK is likely to separate many of the less sophisticated attackers commonly known as script kiddies, from the more advanced attackers. This is because of the efforts and resources taken to compromise such deceptive systems. Consequently, the deceptive ability of the DTK distracts attackers from the real assets and exhausts the attackers' resources on the faked system.

2.13.2 Honeypot solutions

BackOfficer friendly

In 1998 the 'Cult of the Dead Cow' (cDc) community designed BackOfficer Friendly (BOF) to combat the Back Orifice Trojan on UDP port 31337. BOF is a honeypot that also generates faked replies when a connection is made by an attacker to a specific port on the computer, running the services Telnet, FTP, SMTP, POP3, or IMAP2. BOF pretends to open the connection, while it logs the activity on the port, generates an alert to the victim, and then closes the connection on that port.

Specter

Specter is a commercially supported, production honeypot (see section 2.13.3) that encompasses greater functionality and capabilities than BOF. It requires low interaction, is easy to deploy, simple to maintain, and is low in risk.

Specter is able to emulate 1of 13 different OS vulnerabilities at the application level by providing application banners, and has extensive alerting and logging capabilities. Small modifications on the Specter honeypot solution allow it to appear more realistic and hence it is slightly more interactive than BOF. An alerting function also allows a system administrator to be contacted in real time. The information gathered by Specter is limited; however it is ideal for confusing or wasting time for an attacker (*ibid*, 2003).

Honeyd

The Honeyd incorporates the use of "Blackhole monitoring" (*ibid*, 2003, p.144), which is the technique of monitoring and collecting data from entire network blocks for analysis. Honeyd can successfully emulate hundreds of OS's at the application and TCP/IP network stack level. Honeyd is also able to detect, capture, alert, and monitor networks of millions of systems through real-time interaction with the attacker using customised services.

Honeyd can actively simulate a whole network and sub network topologies. This simulation may be achieved by instructing a daemon to route packets to nodes, decrementing the Time to Live (TTL), showing attributes of packet loss, latency, and Internet Control Messaging Protocol (ICMP) replies, thus performing as real network packets traversing a network. Furthermore, Honeyd utilises a 'personality engine' (Provos, 2003) to process network packet content such as stack behaviour of fingerprinting formats of the virtual OS.

Address Resolution Protocol in Honeyd (ARPD), is a service that runs in combination with Honeyd. When a connection request is made on an IP address, Honeyd searches for the OS bound to that IP space in its configuration file. If there is no assigned OS to the requested IP, then ARPD assigns the default OS. The default OS is any OS that the researcher wishes to bind to all unassigned IP addresses within Honeyd's network.

Primarily designed for UNIX, Honeyd is relatively easy to install and configure and is ideal for research. It may gather Internet trends of worm activity, exploit tools, and automated attacks (Spitzner, 2003). Honeyd is thus an appropriate honeypot for investigating script kiddie attacks and attacking methods.

ManTrap

ManTrap is another commercial honeypot that does not emulate single services but entire OS's and can create up to 4 virtual OS's. ManTrap incorporates extensive administrative control, data capturing capabilities, and can simulate production applications such as a DNS, web server, or database. A master OS monitors and controls the attacker through mirrored partitions residing in cage-like environments, where attackers are not able to exit and attack the host OS.

Honeynets

The honeynet is often the most difficult to deploy and maintain because it is a true production system placed and monitored from behind a firewall, primarily deployed for research into attacking tools and tactics. The extreme high interaction is due to complete OS's in multiple honeypots deployed within a highly controlled network. Hence, the honeynet is able to capture all activity, and decreases risk by containing the attacker's activities. One major benefit of a honeynet is that newly discovered risks may be addressed before the technologies are deployed in real production environments (*ibid*, 2003).

2.13.3 Honeypot technology for intrusion detection

"A honeypot is a security resource whose value is in being probed, attacked, or compromised" (Spitzner, 2003, p.3). A honeypot's prime stratagem encompasses the use of deception to either mitigate risks through detecting attacks in the form of production honeypots; or gain knowledge of the hacking communities' tools and tactics in the form of research honeypots.

Lance Spitzner, founder of the Honeynet project vigorously advocates for the use of honeypot technology as it gives victims control and greater understanding of hacker activity (2003). A greater ability to identify, detect or capture the attacker is decided by the type of honeypot constructed and its deployment (*ibid*, 2003). This is analogous to the deceptive lessons of animals as the effective implementation of the honeypot deception may be produced from clear deceptive goals, knowledge of the adversary, and the environmental circumstances.

Results from a recent study (Yek & Valli, 2002) showed a research honeypot deployed in a wired network environment effectively deceived popular brute force attacking tools Nessus and NMAP. The honeypot in this study utilised Cohen's DTK as the deceptive tool. The honeypot was attacked and, buffer overflows were manually counted and cross-evaluated with Nessus reports to provide evidence of effective deception against a would-be attacker.

Benefits of production honeypots

Production honeypots are useful when deployed to detect and report on abnormal network activity. Network traffic is not usually configured to be directed through a honeypot. This configuration may reduce the problem of false positives and consequently, the only packets sent to the honeypot have no purposeful function except harm.

The "valuable information" (Spitzner, 2003, p.59) collected and aggregated may identify a scan, probe or attack. The information may then be used to establish trend analysis and statistical modelling of targeted ports, services and protocols used. This aids in detecting and researching attacks and attack methods. More importantly, this identifies an organisation's exposed vulnerabilities that an attacker may be targeting. Thus a return on investment through definitive results and minimal cost provide incentive for honeypot use in deceptive environments to protect assets (*ibid*, 2003).

Benefits of research honeypots

Many lessons may be learnt from the deployment of research honeypots, which do not aid to reduce risk to organisations. Rather, research honeypots gather information that may be applied to improve prevention, detection, and reaction to attacks (2003). Research honeypots are able to capture extensive forensic data of attacks and the attack method used by the enemy (Honeypots Net, 2003). This is essential as "the greatest challenges the security community faces is lack of information of the enemy" (Spitzner & Roesch, 2001, ¶23).

The intelligence gathering function of a research honeypot aids to uncover vital information that may be used to improve network security for organisations (*ibid*, 2001). The information gathered includes whom the threat is, and thus identifies if the attacker is a script kiddie, an activist group of hackers, or a single highly skilled hacker. Knowing who the attacker is may also help the victim determine why they are attacking. A script kiddie is typically a "bored [and] lonely teenager" (Search Security, 2003, ¶2) intending to compromise as many systems as possible using simple to operate, automated tools (Honeynet Project, 2000). An activist group or skilled hacker may wish to use complex and strategic methods to perform a specific purpose, such as a political message, on a single organisation (Spitzner, 2002b).

Depending on the skill level of the attacker and their intentions, many methods or tools may be adopted to execute an attack. A research honeypot is an "excellent tool for capturing automated attacks" (Spitzner & Roesch, 2001, ¶24). As automated attacks target whole network blocks or blocks of IP addresses, the honeypot will capture all the attacks and identify evident trails of an automated tool. This may then be examined to discover how the automated tool was used for exploit.

Attack intelligence gathered through research honeypots, are regarded as a "critical asset" (*ibid*, 2001, ¶23). Spitzner and Roesch assert that the ability to identify and understand an attack is the best method to defeat the attack (2001). Furthermore, research honeypots provide the ability to discover and investigate diverse attacks and attack methods. This information then becomes vital to the understanding and improvement of security measures used in organisations.

Risks of honeypots

Risks however may arise if the honeypot is programmed incorrectly often due to human error. These errors then allow an attacker to compromise the honeypot itself. The resulting danger is the attacker may gain entry into and damage the protected network, or use the compromised honeypot to conduct third party attacks. Therefore, the honeypot should aim to control the attacker within the monitored environment of the honeypot (Spitzner, 2003).

While a honeypot gains value when it is exploited, if attackers intentionally or unintentionally circumvent the honeypot, then a compromise in the real system may not be detected or recorded. Alternatively, attackers may recognise the DTK or other honeypot signatures through techniques such as OS fingerprinting. Managing system updates and checking for weaknesses, helps prevent compromise of the honeypot.

"Threats are always adapting and changing – and so will honeypots." (*ibid*, 2003, p.111). The revolution of the WWW and technology has fashioned changes in the new-networked environment. Operational WLANs are growing and hence a transition from wired deceptive honeypots to wireless deceptive honeypots is widely anticipated.

2.13.4 Wireless honeypots

The need to deploy wireless honeypots has become apparent due to the recent popularity of wireless networks (Lemos, 2002). In the article Catching wireless hackers in the act, Spitzner states "It is important to see how the bad guys are breaking into systems using not just wired networks, but wireless networks as well" (2002, ¶2). On June 15, 2001, US research and engineering organisation Science Applications International Corp (SAIC) implemented an operational wireless research honeypot designed to investigate wireless attack methods through observation.

Additionally the Wireless Information Security Experiment (WISE) has deployed an 802.11b wireless network in an undisclosed location in Washington DC, entirely for research purposes. The WISE wireless research honeypot employs five Cisco APs, a small number of computers running vulnerable services for added appeal to hackers, with two high-gain, omni-directional antennas, for widespread coverage. Network packets are passively logged on a customised Intrusion Detection System (IDS), in addition to a back end logging host for traffic generated to and from APs (Poulsen, 2002).

The SAIC wireless honeypot has not revealed any nefarious activity other than single ping sweeps and unsuccessful attempts to surf the WWW. However, the WISE wireless honeypot is expected to have Internet connectivity in the near future that will present a consent-to-monitor banner to allow legal observation of Internet utilisation via the wireless honeypotted network. As there is no real productive use of either wireless honeypots other than to research emerging wireless tools and tactics, all network activity is closely examined (*ibid*, 2002).

In addition to the same motivations for deploying a research honeypot, a wireless honeypot will thus enable the security community to investigate the wireless attacking tools and techniques that are being used by attackers (Schoeneck, 2003).

2.13.5 Legal issues pertaining to honeypots

The arising legal challenges of honeypots could restrain the effective use of these deceptive defence mechanisms owing to strict regulations of the country the honeypot is being deployed in (Gerwehr & Anderson, 2000; Spitzner, 2003). Spitzner (2003) specifies three possible legal issues arising from the deployed use of honeypots: privacy and entrapment of the attacker, and civil liability of the victim. The legality of actions may be subject to the nature of the information that is collected and what is intended to be done with it, “similarly, what intruders [or attackers] do while on your honeypot may expose you to certain legal troubles” (*ibid*, 2003, p.368).

Therefore, certain information about the activities of the unauthorised attacker may not be captured rightfully by the victim’s honeypot and thus unlawful handling or dissemination of that information may result in an invasion of privacy on the attacker’s part. Alternatively, an attacker may argue the honeypot to be entrapment, designed to persuade the attacker to carry out a criminal activity that the attacker otherwise would not have committed given the honeypot was not deployed.

Equally significant is the concern of the victim’s civil liability, should the attacker launch third party attacks from the victim’s compromised honeypot. Similarly, the compromised honeypot may be used to store contraband, such as stolen credit card numbers, or pilfered or prohibited software, which may be difficult if not impossible to defend against in court.

However, each country’s own legal statutes, regulations, and case laws independently state the legalities of deployed honeypots. Additionally, organisational policies within regulated industries or governments should provide individual guidelines and procedures for honeypot deployment whereby violations of internal policies or breaches of contracts may be handled in isolation.

2.14 Other Integrated Security Mechanisms within Network Security

Honeypots are security solutions that operate as deceptive defence systems. Network Intrusion Detection Systems (NIDS) and packet sniffers are additional network security mechanisms. For the purpose of this research, an NIDS will function as a logging tool that will produce forensic evidence of intrusions and attacks on the deceptive wireless honeypot.

Intrusions may be defined as “an unauthorised usage of or misuse of a computer system” (Ptacek & Newsham, 1998, ¶2). A network intrusion system (NIDS or IDS) is a security technology that passively monitors network activity and attempts to identify and isolate ‘intrusions’ against computer systems and alert unauthorised activity (Ptacek & Newsham, 1998; Spitzner, 2003).

Figure 2.5 illustrates a wired network topology in which the attacker typically utilises the Internet as a means to reach the corporate router and access the internal Ethernet connection where a corporate end system may be compromised. However, residing transparently on the corporate Ethernet is also an NIDS network monitor. The NIDS may be set in a promiscuous mode to collect all packets and is thus passively ‘sniffing’ packets

(Ptacek & Newsham, 1998)

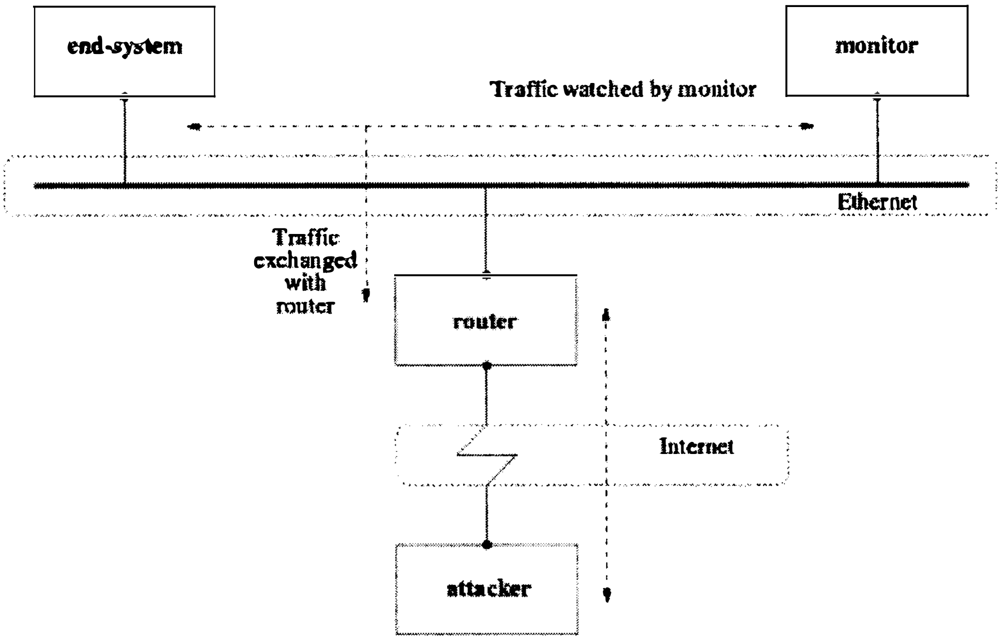


FIGURE 2. 5 Example network topology using a passive monitor

The NIDS operates unobtrusively on the network causing no disruption or degradation of network performance. Thus, an NIDS is difficult to evade as all packets traversing the network media are monitored transparently (Ptacek & Newsham, 1998).

NIDS's are able to gather forensic verification of network activity that may identify the origins of attacks, and may render attackers accountable, or deter them (*ibid*, 1998). Identified attacks may be examined at the network packet level through a process of analysis and verification of protocols, and extracting the relevant information. An NIDS will also detect known signatures that may signify erroneous activity or suspicious packet payloads.

Drawbacks of a NIDS are that frequent updated signatures by the system administrator, are necessary to enable the NIDS to detect malicious packets. However, new attacks and evasion methods to circumvent NIDS detection, that contain unidentified signatures, are constantly being developed (Spitzner, 2003). Additionally, the data collection may appear voluminous. However, this may be managed through a comprehensive system of data mining.

2.14.1 Packet sniffers

Packet sniffers passively collect and analyse network packets on a wired or wireless medium. Packet sniffers are transparent to the network, and they do not have alerting functionality. Many packet sniffers are open source, and therefore differ in capabilities. SNORT (Caswell & Roesch, 2002) and AirSNORT (Hegerle & Bruestle, 2002) are common packet sniffers that passively capture network traffic at the TCP/IP level. This data traffic identifies information such as the source and destination IP address, MAC address and port number, the time and date, in addition to the specific protocol used. This information is useful in determining what attacks are being executed and what they are targeting.

2.15 Review of the Literature on Deception, Honeypots, and 802.11b.

Deception may be expressed as a deliberate and/or fortuitous distortion of a perceived reality. Evidence suggests that deception forms a major part of biological existence and thus survival. This was found from the literature of deceptions used by animals (Bowyer, 1982; Gerwehr & Glenn, 2000; RAND, 2001).

The characteristics of deception may be combined into a framework that demonstrates a decomposition of deceptive possibilities, encompassing a defensive or offensive posture. Both defences and offences may implement an actively or passively executed deception. Additionally, a static, dynamic, adaptive, or premeditative level of sophistication may implement a masked, misleading, or confusing effect.

A generalised framework that embraces all the identified deceptive possibilities may then be applied to any object seeking to implement any of the deceptive effects (see section 3.2 for framework). Network defences may utilise the same deceptive characteristics to create a systematic method for deploying a deceptive defence.

Investigation of deceptions may be tested on a wireless honeypot. Honeypots have demonstrated themselves to be valuable research tools for discovering and understanding attacking methods. A honeypots primarily uses deception to appear and perform as a real network. It may emulate OS platforms, services on TCP/IP ports, application level banners, and whole network topologies, depending on its configuration.

Contemporary networks may now utilise wireless resources to expand data communication capabilities. These include the deployment of access points, antennas, and the use of wireless clients. The common 802.11b standard used however, has been demonstrated to have several architectural flaws that may allow compromise of devices utilising the standard.

Therefore, a wireless honeypot may be deployed to investigate the effectiveness of deceptive network countermeasures against common network attacks.

3 CHAPTER 3 - RESEARCH METHODOLOGY

The research focus questions formulated from the literature review require an experimental approach to both support propositions, and demonstrate intended or expected responses. Figure 3.1 demonstrates a hierarchy of research functions that express a modelled sequence of experimental processes based on Sarantakos' (1998) and Davis' (1997a) steps of experimental research.

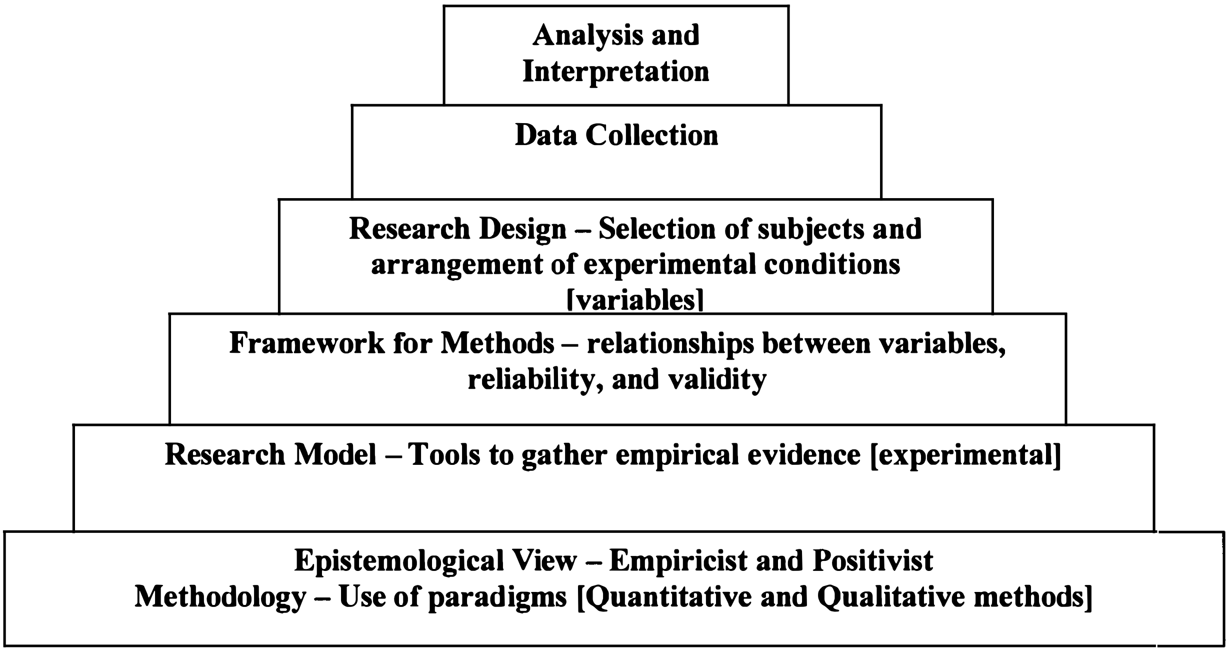

Analysis and Interpretation
Data Collection
Research Design – Selection of subjects and arrangement of experimental conditions [variables]
Framework for Methods – relationships between variables, reliability, and validity
Research Model – Tools to gather empirical evidence [experimental]
Epistemological View – Empiricist and Positivist
Methodology – Use of paradigms [Quantitative and Qualitative methods]

FIGURE 3. 1 Sequence of experimental processes

3.1.1 Epistemology

At the base of the pyramid is the epistemological stance which are the "views about one's own knowledge and learning [or] views about the nature of discovery and knowledge in the scientific community" (Elby & Lising, n.d., ¶3). According to Dolhenty (2003), individuals utilise senses and perception to fashion concepts and ideas that form a reality; and knowledge is attained by the affirmation or denial of an interpretive judgement about reality. Furthermore Pollock, (cited in Chesnevar, Maguitman, & Loui, 2000) maintains that epistemology involves the acquisition of reasons for supporting arguments.

Knowledge may be defined as the accumulation of a body of facts (Clarke, 2001), and a comparatively abstract description of truth is characterised by judgements about reality (Dolhenty, 2003). Thus, the formulation of knowledge and truth develop the philosophical and conceptual foundations for observing and interpreting reality, and accordingly characterises the research methodology.

The epistemological views of the researcher agree with the preceding depiction of reality, and thus consider knowledge and truth to be observable phenomenon that may be used to draw conclusions. Therefore, this experimental research will be based on observable outcomes that will answer the research questions:

1. Can a framework for deception be applied to common network countermeasures to reduce the effectiveness of attacks?
2. How effective is deception in a wireless honeypot against brute force attacks?
3. Under what conditions, and do conditions vary by the type of attack?

3.1.2 Paradigms

There are many epistemological views that draw parallels from several schools of thought. The epistemic views of empiricism and positivism reinforce the experimental nature of the research questions. The collected propositions in each epistemic view including beliefs, values, and techniques (Kuhn, 1970 cited in Sarantakos, 1998) form paradigms of established explanations of how the world is perceived (Sarantakos, 1998). Thus the epistemological view determines the set of paradigms employed, and the paradigm in turn should be in context of the methodology (*ibid*, 1998).

3.1.3 Empiricism and positivism

Empirical epistemology relies on the principle that knowledge is derived from observed or experimental observations (Philosophical Society, n.d.; Trochim, 2002). A positivist view is found on rules and procedures for the observation and measurement of data (Sarantakos, 1998). The positive paradigm originates from scientific laws that are typically deductive through a process of abstraction to concretisation, and are explained through universal causal laws (*ibid*, 1998).

Though there are several branches of positivism, which include logical positivism, neopositivism, and methodological positivism, a comprehensive depiction of the positivist paradigm distinguishes reality as:

> Everything that can be perceived through the senses… is objective, rests on order, is governed by strict, natural and unchangeable laws, and can be realised through experience (*ibid*, 1998, p.36).

Thus, empiricist and positive epistemic views of knowledge and truth advocate observed and measured actions as a medium for researching scientific hypothesis in experimental conditions. Furthermore, the positivist paradigm applies formally defined methods incorporating concepts of measurement, validity, threats to validity (external, internal, construct, statistical conclusion), and reliability (Thomsen, n.d.) that are used in a methodological process in the research to establish the cause-and-effect relationships (Davis, 1997b).

Other epistemological views embracing similar philosophies and ideologies from Philosophical Society (n.d.) include rationalism where reason and intuition are independent of experience; pragmatism where truth is the subject of experimentation; and conversely, realism views knowledge and truth as attainable attributes from experiencing the actual 'form' of the subject in query.

Alternatively, paradigms that deviate from a positivist theoretical direction mainly involve interpretive or naturalistic values. The interpretivist approach argues that various factors that are difficult to isolate and control create flawed assumptions, which result in prejudiced observations. Consequently multiple interpretations of the same phenomena emerge and truth becomes unattainable (Sarantakos, 1998) Branches of the interpretivist discipline include phenomenology, ethnography, hermeneutics, psychoanalysis, and sociolinguistics and are often adapted to social research involving human behaviour.

3.1.4 Views adopted by the researcher

The positivist paradigm defends investigation of experimental outcomes as truthful, and may thus be objectively analysed to derive deductive inferences. Therefore, positivism will be adopted as the primary view for the experimental research. Interpretive views will be adopted subsequent to the data collection to allow the researcher to rationalise conclusions that may be interpreted from the data.

3.1.5 Quantitative and qualitative paradigms

Experimental outcomes of empirical observation and measurement give rise to further principles pertaining to the data collection and analysis in the form of the quantitative paradigm. The quantitative paradigm is based on positivist philosophy where the natural world is governed by fixed laws that are empirically observed (*ibid*, 1998). Quantitative research aims to determine and quantify relationships between variables through descriptive or experimental methods (Hopkins, 2000).

As the research has been determined as experimental, the quantitative methods employed will aim to ascertain the relationship between an independent variable (IV) and dependent variable (DV). The research questions hypothesis that direct manipulation of the IV – the honeypot, causes the changes in the DV’s – the resulting deceptions; and not other erroneous variables (Davis, 1997b). Furthermore, the experimental research will aim to eliminate alternative variables (*ibid*, 1997b).

Studying the relationship between variables involves taking a quantitative measurement, performing some changes, and taking the measurement again. This process of iterative intervention is known as repeated measures and will be used in the experiment to determine the causality of the relationship between the IV and DV’ s (*ibid*, 2000). Tools that may be used for interpretation of quantitative, statistical data are usually mean, median, mode, frequencies and regression analysis. Quantitative research then leads to deductive theories.

Upon experimental execution of the IV and DV's, deductions from the outcomes may then be interpreted. Hence, an associated qualitative paradigm will be fostered as a supplement to quantitative research. Qualitative analysis involves the discovery of themes and patterns within the data and is typically exploratory and descriptive. The qualitative paradigm may also be based on positivistic views that the data is true given the explicitly stated experimental conditions and limitations.

3.1.6 Methodology

The methodology encompasses the design process and the use of methods which will both be determined and justified by the principles of empiricism and positivism, and established through the doctrines of the researcher's favoured epistemology (Crotty, 1998). Therefore, the methodology translates guidelines for the research practice based on paradigmatic assertions about reality.

An empirical investigation is the primary data gathering tool arrived through observation and has been thought arguably to constitute the epistemology for understanding experience (Willemsen, 1974). Moreover the research strategy for empirical observation typically employs correlational, field-descriptive (applying correlation methods), or experimental techniques summarised in brief from (Huck & Cormier, 1996; Willemsen, 1974).

Correlational investigation

Correlational investigation measures variables against pre-existing traits in order to ascertain relationships between variables, drawing its distinction from measuring variables that are caused by manipulation of the researcher. Correlational studies determine the existence of a relationship and the nature of that relationship by examining both the variables simultaneously in addition to the strength of the relationship. Correlations may be established as a high-high, low-low case; a high-low, low-high case; or with little systematic tendency.

Field studies

Field studies utilise settings where "behaviours of interest naturally occur" (Willemsen, 1974, p.34), given a laboratory environment may be too artificial and stifle 'normal', or 'typical' activity. A field study researcher would elect a complementary landscape for the study and consider a narrative description and interpretation of the setting and circumstances.

Experimental studies

Using experimental strategies the researcher isolates, manipulates, and controls variables relating to behaviours or experience pertaining to the researched phenomena.

3.1.7 Research method used for the conduct of research

Empirical observation may engage a combination of the above research strategies to optimise results for conducting the research, and as a process of operationalising the research questions.

Thus the research methodology for this research will predominantly encompass experimental strategies as the principal implementation of empirical and positivist epistemology, and through the observation and measurement of regulated variables. Therefore the independent variable (IV) controlled by the researcher will be tested with dependent variables (DV), that are products of the direct manipulation of the IV.

A correlational investigation will ascertain the presence of relationships between variables so that a cause-effect association may be tested to verify results. A field study on the other hand will offset the artificial composition of the experiment by allowing the researcher to assimilate an archetypal environment to execute the experiment. This is because a completely contained wireless network environment would be unachievable by the researcher.

An underlying positivist paradigm will then direct the structure and process of the framework of methods within the experimental implementation.

3.1.8 Framework for methods

Research methods are the tools or instruments of data generation and analysis that are employed to accumulate empirical evidence, and are invoked from the underpinning precepts of the major elements of the methodology (Crotty, 1998; Sarantakos, 1998). Hence empirical and positivist, experimental methods are designed in a specifically defined and detailed framework.

3.2 Framework for Deception

The empirical nature of the research will utilise the anecdotal investigations of deceptive origins and implementation as the experimental variables to conceive a framework for the research. Thus, the conceptual framework for methods will be the foundation of the experiment, and applying the established methodologies formerly described will answer the research focus questions:

1. Can a framework for deception be applied to common network countermeasures to reduce the effectiveness of attacks?
2. How effective is deception in a wireless honeypot against brute force attacks?
3. Under what conditions, and do conditions vary by the type of attack?

Therefore, collating the deceptive characteristics suggested in the literature, a framework for deception may be constructed to encompass all the previous mentioned categories of deception examined in the form of an attack tree (Schneier, 2000).

The framework for deception is the conceptual basis from which to implement the research design of the testable experimental conditions. The framework will also be used to identify the mapping of network countermeasures and attacks to deceptions that will be deployed on the wireless honeypot.

The research design (see chapter 4) will depict how each experimental variable will be mapped to a type of deception from the framework. The experimental implementation will identify the type of deceptions (DV's) to be deployed as a network countermeasure or attack; and the outcomes of deceptions on the wireless honeypot (IV). Therefore the following arrangement and execution of the experimental variables will guide the execution of the experiment (Sarantakos, 1998).

Figure 3.2 characterises the framework for deception from an offensive or defensive stance for a single deception. Both account for active and passive states that may implement a static, dynamic, adaptive, or premeditative level of sophistication and resulting in a masked, misleading, or confusing effect. A single deception may assimilate one or many deceptive pathways.

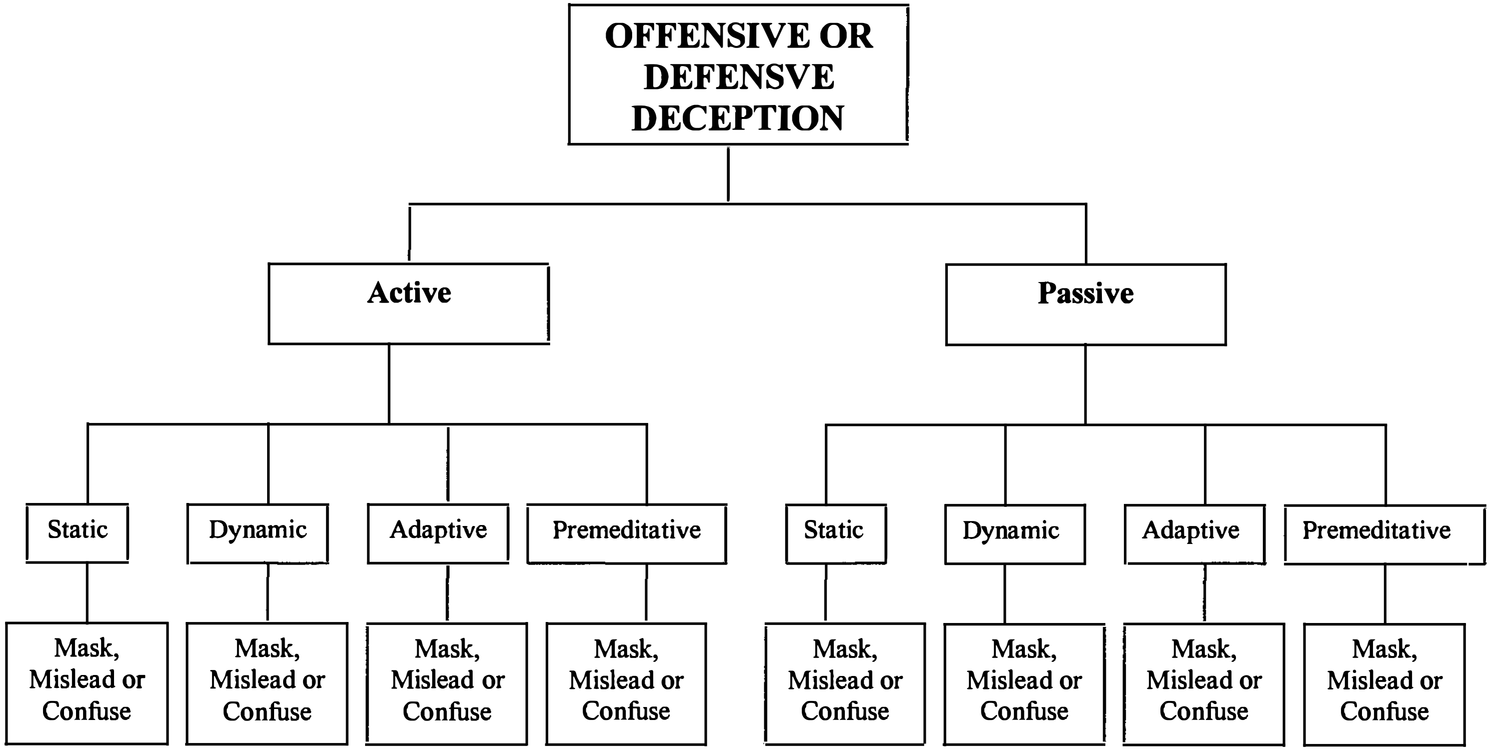


FIGURE 3. 2 Framework for deception

4 CHAPTER 4 - RESEARCH DESIGN

The aim of the research is to measure the outcomes of deceptive capabilities against brute force attacks so that results may be examined using a systematic approach that either supports or nullifies arguments pertaining to the research questions:

1. Can a framework for deception be applied to common network countermeasures to reduce the effectiveness of attacks?
2. How effective is deception in a wireless honeypot against brute force attacks?
3. Under what conditions, and do conditions vary by the type of attack?

Upon definition of the underlying methodology for the research questions, a research design may be devised. The research design will encompass two major phases. The first will describe the logical and technical implementations of the hardware and software used in the experiment. The second phase will define each experimental variable in varying deceptive conditions thus describing how they may be deployed in the wireless honeypot.

This research involves constructing and deploying an integrated wired and wireless honeypot utilising deceptive mechanisms that encompasses a fake access point, Honeyd and the Intrusion Detection System (IDS). The deceptive wireless honeypot will be deployed in the form of deception in depth.

4.1 Logical and Technical Implementation

4.1.1 Logical structure of the deployed deceptive wireless honeypot

Figure 4.1 illustrates the logical deployment of the deceptive wireless honeypot through the rings of deception in depth. A fake access point utilising FakeAP is situated on the peripheral as ring 3 as it is the most static and transparent of the deceptions. The Honeyd virtual networks encircle the honeypot asset within the second ring of the deception given that Honeyd implements variable network level deception that will sustain the majority of the deceptive network attacks. The honeypot asset is located centrally (ring 1) where only strategically targeted deceptive attacks will be able to penetrate.

The Honeyd logs and SNORT (Caswell & Roesch, 2002) Intrusion Detection System (IDS) will form the central logging structure (CLS). The aim of the CLS will be to log and record all wireless traffic to the honeypot to verify the results of the attack tools used. This will detect network activity such as scans and probes from the attacking machine; however, from the victim's perspective.

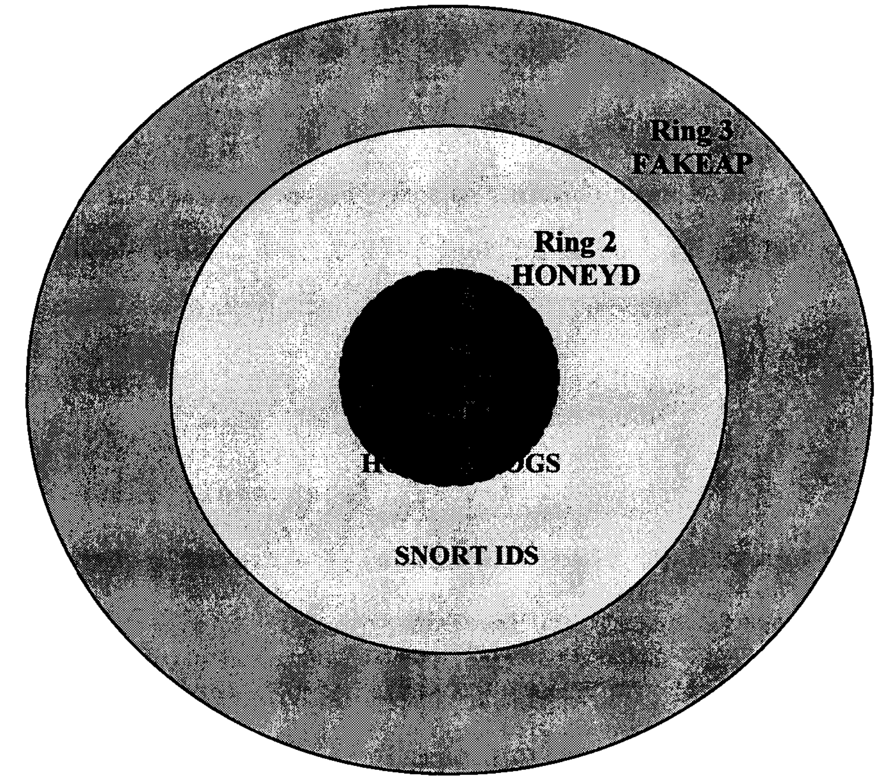


FIGURE 4. 1 Applying the wireless honeypot to deception in depth

4.1.2 Ring 3 FakeAP

Access points are used as gateways to bridge to the private wireless network. There may be several configurations for an access point depending on the role of the AP gateway within the network. FakeAP is software that may be used to simulate many access points.

APs transmit 802.11b data packets containing the IP of the access point gateway, configurations, and possible information about the connecting wireless network. Thus, an AP is often sniffed by an attacker to gain further access to exploit network resources situated behind the AP (Spitzner & Roesch, 2001).

FakeAP changes configurable parameters through the command *iwconfig*. The behaviour, signal strength, and data contained in beaconed packets are dependent on the configuration of *iwconfig* parameters. Hence, the way in which FakeAP is set up will determine the strength of the ring 3 deception.

The synopsis for the *iwconfig* interface in Linux appears below (Tourrilhes, 1996). Table 4.1 identifies and describes the function of each of the *iwconfig* parameters that will be utilised in the deceptive wireless honeypot.

```
SYNOPSIS

    iwconfig [interface]

    iwconfig interface [essid X] [nwid N] [freq F]

                        [channel C] [sens S] [mode M]

                        [ap A] [nick N] [rate R] [rts RT]

                        [frag FT] [txpower T] [enc E]

                        [key K] [power P] [retry R]

                        [commit]
```

Name of parameter	Description	Configuration for experiment
essid	Domain identifier or network name assigned as the name of the AP, which also specifies the cells that are part of the same virtual network	Name of AP - "WinNT4 Web AP"
nwid/domain	Creates logical wireless networks that are used to differentiate and identify nodes that belong to the same cell	"WinNT4 Web AP"
freq/channel	Frequency is set in GHz and the channel is a number. Regulations control the number of channels available and usable frequencies.	802.11b wireless default is 2.4GHz. The channel will switch between numbers 1-11
sens	Sensitivity threshold for the lowest signal level to attempt packet reception and used to avoid background noise measured in dBm	The lowest threshold for which the channel is not considered busy and the handover threshold maintaining association with the access point - 80dBm
mode	Operating mode of the device depending on network topology: • Ad-hoc – 1 cell & 1 AP • Managed – many cells roam with many APs • Master – node acts as AP • Repeater – node forwards packets	Managed mode will reflect the virtual networks created by honeyd
ap	Forces the card to pre-register with the AP. when connection is too low wireless cards will attempt to connect to the strongest signals beaconed from an AP	Will be set to "true"
rate/bit	The speed at which bits are transmitted over the medium used for cards that support multiple bit rates	11Mb

TABLE 4. 1 FakeAP parameters used for the wireless honeypot

Proposed attacks on ring 3

The role of the ring 3 deception is to produce an access point gateway for attackers to discover ingress to the private wireless network via the IP of the FakeAP. Wireless sniffing tools such as Kismet (Kershaw, 2003), and Netstumbler (Milner, 2002) identify the presence of APs, and any AP related parameters that are instructed by *iwconfig*. Subsequently, the attacker advances to the next level of deception, which are the virtual wired and wireless networks within Honeyd.

4.1.3 Ring 2 Honeyd

The Honeyd is configured to appear as a wired network containing web servers and client workstations that bridge added wireless services. Figure 4.2 illustrates the course of logical network routes deployed through the Honeyd virtual networks. A router typically interfaces outside Internet connections with the internal network that is segmented to form various operating environments

Honeyd aims to ascertain the attacker's objectives and resources by mimicking legitimate services via direct manipulation of the network stack of the designated operating system (OS). Therefore, the configurations for Honeyd involve reproducing the TCP/IP handshake sequence for OS matches, packet latency, packet loss and traceroute functions with the intention of appearing realistic. Table 4.2 outlines the technical implementation of each of the selected Honeyd operating platforms for the deceptive wireless honeypot.

Proposed attacks on ring 2

When an intruder ascertains the IP address of the AP gateway, further probing will identify network IP addresses that are really the Honeyd virtual networks. Each wired and wireless network topology emulates an operating system "personality" (Provos, 2003) that is allocated through the researcher's preference. However, each selected OS personality must be a precise match of an NMAP or Xprobe prescribed OS signature for the reason that Honeyd functions by simulating connections made at the network level (TCP, IP, UDP, and ICMP).

Honeyd was designed to only provide network level interaction due to combat popular network security scanning tools such as NMAP (Conry-Murray, 2001; Fratto, 2003; Noordergraaf, 2002) that utilises TCP/IP handshakes at the network level, to administer connections. NMAP enumerates network information such as the OS type and version, opened or blocked TCP/IP ports, in addition to active services (Fyodor, 2003a). Thus, relayed packets from the network level give substantive information that denotes consequent OS vulnerabilities and possible exploitive opportunities for the attacker.

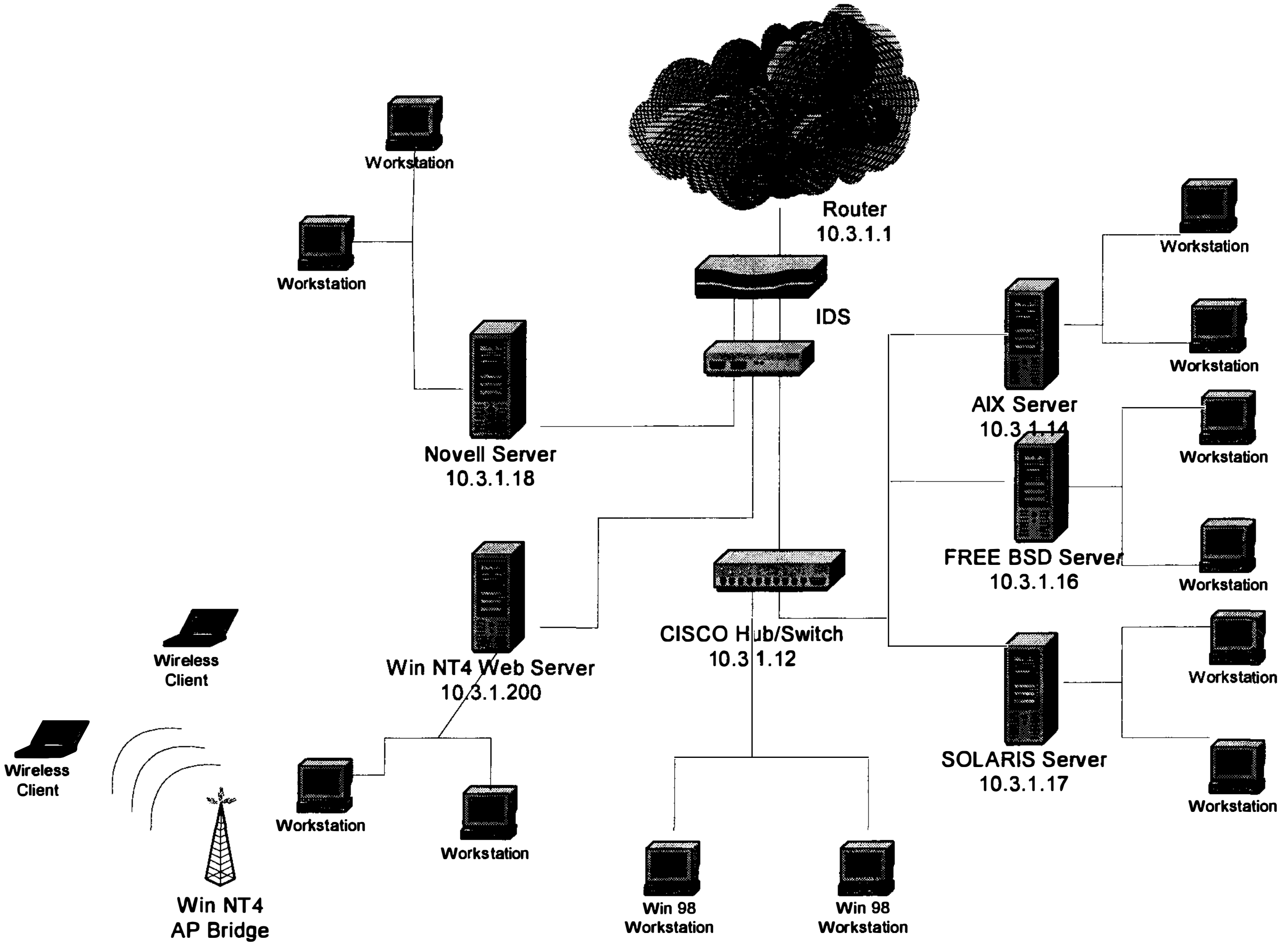


FIGURE 4. 2 Logical configuration of the Honeyd virtual networks

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Cisco 3com Router	3com Office Connect Router 810	10.3.1.1.1	TCP 80 – HTTP TCP 139 – NetBIOS TCP 137 – NetBIOS-ns UDP 137 – NetBIOS-ns UDP 135 – MS Exchange	Telnet	Set default TCP action reset Set default UDP action reset Set router uid 32767 gid 32767
Cisco Hub/Switch	Cisco Router/Switch with IOS 11.2	10.3.1.12	No ports open	Telnet	
Default to FreeBSD	FreeBSD 2.2.1-STABLE	10.3.1.13	No ports open		Set default TCP action reset
AIX Server	AIX v4.2	10.3.1.14	TCP 25 – SMTP TCP 80 – HTTP TCP 21 – FTP	Web FTP	Set default TCP action reset
FreeBSD	FreeBSD 2.2.1-STABLE	10.3.1.15	TCP 80 – HTTP	Web	Set default TCP action reset
OpenBSD Server	FreeBSD 2.7/SPARC or NFR IDS Appliance	10.3.1.16	TCP 80 – HTTP	Web	Set default TCP action reset
Solaris Server	Solaris 2.6 – 2.7	10.3.1.17	TCP 80 – HTTP	Web	Set default TCP action reset

Novell Server	Novell Netware 3.12 – 5.00	10.3.1.18	TCP 80 – HTTP	Web	Set default TCP action reset
FreeBSD	FreeBSD 2.2.1-STABLE	10.3.1.19	TCP 80 –HTTP	Web	Set default TCP action reset
Default to FreeBSD	FreeBSD 2.2.1-STABLE	10.3.1.19	TCP 80 – HTTP	Web	Set default TCP action reset
Cisco Hub	Cisco Router/Switch with IOS 11.2	10.3.1.20	TCP 23 - TELNET	Telnet	Set default UDP action reset
Cisco Aironet AP	Cisco Aironet 3500 Wireless Bridge V5.0J	10.3.1.200	No ports open		MAC 00:40:96:31:81:cf

TABLE 4. 2 Honeyd technical configuration

4.1.4 Ring 1 honeypot asset

The honeypot itself is a Linux Mandrake 9.0 machine that runs all the deceptive services that are required for the rings of deception in depth, in the wireless honeypot. The honeypot is the most protected ring, as the outer rings of deception cannot function without the operational core. Therefore, the honeypot deception encompasses the deceptions deployed through rings 2 and 3.

Proposed attacks on ring 1

Attacking the honeypot itself requires targeted buffer overflows on OS flaws, or vulnerabilities revealed from the penetration testing on ring 2 of the deception. Buffer overflows or "smashing the stack" (McClure, Scambray, & Kurtz, 2001, p.161) refers to an attack that attempts to overwhelm the virtual memory with more input than the buffer stack may contain.

Buffer overflow attacks are an increasing danger (Aleph One, n.d.; Cowan, Wagle, & Pu, 1999; Grover, 2003; McClure et al., 2001), and "represent one of the most serious classes [of] security threats" (Cowan et al., 1999, p.1) in relation to network penetration. A successfully executed buffer overflow can cause system crashes and core dumps that may consequently allow the intruder to inject attack or malicious instructions into the system and network (*ibid*, 1999).

4.1.5 Central logging structure

The central logging structure encompasses an IDS namely SNORT (Caswell & Roesch, 2002) packet sniffer, that will serve in cooperation with the Honeyd logs to passively record all system traffic. The network data collected will confirm network penetration and buffer overflow success through captured and dissected data that include the source and destination: IP address, MAC address, TCP/IP ports, and the protocols used, as well as any buffer outputs.

4.2 Experimental Variables

Each ring in the deception in depth draws on deceptive countermeasures, and each ring is targeted by anticipated deceptive network attacks. This describes the testable conditions for investigating deceptions used on the wireless honeypot. The arrangement of the experimental conditions thus guides the execution of the experiment (Sarantakos, 1998).

The following is a description of the experimental variables. The variables will be implemented to provide quantitative measurement of a causal or correlated relationship between the IV and DV's.

The independent variable

The independent variable (IV) is the deployed deceptive wireless honeypot and has conditions that are manipulated by the researcher as the causal object within the correlated relationship (Davis, 1997b). Quantitative measurements will be used for initial testing of the IV. The researcher will then manipulate the IV and perform a second round of quantitative measurements. This will also be used to identify any changes that may occur.

The dependent variable

The dependent variable (DV) is the effect, or the result of the manipulated IV (*ibid*, 1997b) which will be the different types of deceptions deployed. This will determine and quantify the strength of the correlated relationship, if any. The DV's will be defined in the research design as a matrix (see sections 5.2.1 and 5.3.1).

Alternate variables

Alternate variables which may effect the outcomes of the experiment should be controlled or eliminated in order to support argument that the true cause of the measured outcomes are by result of the IV and not by any deviating variables that are not considered as part of the experiment (*ibid*, 1997a). Alternate variables will be considered in the research resign in descriptive tables of how the DV's are deployed on the IV (see sections 5.2.1 and 5.3.1).

Control variables involve settings, configurations, and limitations of the tools and software used within the experiment that may alter the outcomes of the DV and involve deceptions that are not part of the intended condition of the IV. Control variables should be held constant by the researcher so they do not influence the results (*ibid*, 1997b). However, due to the often-volatile nature of software, some control variables such as a one-off glitch, will be unavoidable. Therefore, they will be considered as a limitation of the research and will be addressed in the results (see chapters 6 and 7) and discussion (see chapter 8).

Random variables

Random variables are other variables that may be potential causes (*ibid*, 1997b) and not the IV. This may include weather, interfering noise, inaccurate machine responses, or bugs in software. Eliminating such random variables involves anticipation and timely preparation. Repeated testing to compare the consistency of results of the experimental testing will aid reliability of results.

Confounding variables

Confounding variables are variables that may alter the IV and may result as the cause instead of the intended IV (*ibid*, 1997b). Therefore, confounding variables of the deceptive wireless honeypot are unintended fluctuations of conditions on the IV, which may be a result in software errors or inaccurate configurations by the experimenter. Repeated tests should identify such errors for elimination. This should be identified during the experimentation, and addressed in the results section (see chapters 6 and 7).

5 CHAPTER 5 - DECEPTIVE COUNTERMEASURES AND ATTACK IMPLEMENTATION

The chief goal of the experiment is to implement a composition of deceptions that simulate a real 802.11b wireless integrated network to test the deceptive countermeasures against common network attacks. Utilising the identified technical and logical configurations of FakeAP and Honeyd, a coherent network using deceptive mechanisms may be deployed as a series of rings in deception in depth.

A constructed matrix of deceptive countermeasures and attacks expresses a diagrammatic implementation of the experimental conditions for testing the nature of the relationship between deceptive defences of the wireless honeypot, and the potential outcomes of deceptive attacks.

Thus, the framework for deception illustrates the theoretical dissection of deceptive categories and characteristics gathered from biological and military case study. The matrices also demonstrate implementation of those conceptual deceptions through testable conditions. The wireless honeypot will be the independent variable (IV) to be manipulated in the experiment, and the deceptive conditions identified in the matrices are the dependent variables (DV's).

5.1 Applying the Framework for Deception to a Matrix

The network countermeasures and attacks are categorised through the same levels of deception identified in the framework. Deceptive defences and deceptive offences are described independently as two matrices for each countermeasure and attack. The remaining deceptive categorisations are maintained in each matrix. This includes an active or passive state, followed by a static, dynamic, adaptive, or premeditative approach to implementing a masked, misleading, or confusing deceptive effect.

5.2 Deceptive Network Countermeasures

Table 5.1 is the matrix of deceptive Defence Network Countermeasures (DNC) to be deployed on the wireless honeypot. The DNC encompasses deceptions employed on the Honeyd virtual networks and the CLS (Honeyd logs and IDS). Each defence strategy is reactive, and thus only executed after the attacking machine has initiated a probe or scan on the targeted wireless honeypot.

Table 5.2 is the matrix of deceptive Offence Network Countermeasures (ONC). FakeAP beacons are the only ONC that will be deployed on the wireless honeypot. It is an offensive deception as it is activated regardless of an attacker executing any probes or scans.

5.2.1 Application of the matrices of deceptive countermeasures

The deceptive DNC and ONC in Tables 5.1 and 5.2 are security strategies that aim to strengthen the wireless honeypot by employing the deceptive characteristics that were mapped from the framework for deception.

Table 5.3 presents a detailed description of the conditions for deploying each deceptive countermeasure against the wireless honeypot. This will also entail explanations of the dependent variables (DV) of the experiment; with consideration of any random variables. The experimental conditions involve:

- Identifying the ring of the deception in depth the countermeasure applies to
- The deceptive effect sought as denoted on the matrix
- An explained function of how the deception is performed
- Random variables that may interfere or alter the intended experimental conditions
- The method to overcome the random variables

Network Countermeasures	Deceptive Defence																							
	Active												Passive											
	Static			Dynamic			Adaptive			Premed			Static			Dynamic			Adaptive			Premed		
	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?
Faked OS personality														✓										
OS platform vulnerability																				✓				
Faked network topology														✓										
Faked TCP/IP stack					✓			✓																
TCP/IP fingerspoofing					✓			✓																
Faked buffer overflow					✓																			
Faked services					✓																			
Faked banner scripts					✓																			
Central logging syslog-ng																✓								
IDS logging																✓								
IDS alerting																			✓					

TABLE 5. 1 Matrix of deceptive DNC

Network Countermeasure	Deceptive Offence																							
	Active												Passive											
	Static			Dynamic			Adaptive			Premed			Static			Dynamic			Adaptive			Premed		
	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?
Faked AP beacons														✓										

TABLE 5. 2 Matrix of deceptive ONC

👽 = mask, camouflage, repackage; ✥ = mimic, mislead, decoy; ? = confuse, dazzle, invent;

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Faked OS personality	2	Mimic	OS appears as a real platform on an OS fingerprint	OS TCP/IP fingerprint does not match an OS platform	OS personality must match a prescribed NMAP or Xprobe signature
OS platform vulnerability	2	Mimic	Appears as a true OS flaw that may be exploited	The flaw cannot be exploitable through a buffer overflow	Test each OS with Nessus for vulnerabilities, if no vulnerabilities occur, then change the OS as it is not deceptive
OS platform vulnerability	2	Mislead	The intruder believes an exploitable vulnerability exists	The vulnerability does not match the OS	Test the vulnerabilities by directing a targeted buffer overflow to that OS flaw
Faked network topology	2	Mislead	Appears as a network of systems and not as a standalone computer	Incompatible platforms on the same network	Conduct Nessus tests to check accurate configuration of networks in honeyd
Faked network topology	2	Mimic	Routes packets through the networks	Packet latency does not match the network topology	Test packet latency through NMAP scans
Faked TCP/IP stack	2	Mislead	Intruder believes the TCP/IP stack is matches the TCP/IP stack of the OS	NMAP TCP/IP stack cannot perform handshakes with the TCP/IP stack of the scanned machine	Test NMAP scans and change the OS fingerprint until NMAP picks up the correct OS match.
Faked TCP/IP stack	2	Mimic	The stack acts the same as a real TCP/IP stack	Possible errors in software	Conduct several NMAP tests to ensure reliability

TCP/IP fingerspoofing	2	Mislead	TCP/IP fingerprint gives the guessed OS of the machine	TCP/IP fingerprint gives an incorrect guess of the OS	Test NMAP scans and change the OS fingerprint until NMAP picks up the correct OS match
Faked buffer overflow	2	Mislead	Intruder believes the machine has been crashed and the system is down	OS may still be probed after an alleged buffer overflow hit	Test buffer overflow attacks against Honeyd and change the OS if it may still be probed after an attack
Faked buffer overflow	2	Mimic	Machine responds with a buffer overflow upon attack	The buffer overflow does not crash or core dump the system as it should	Change the OS fingerprint where the buffer overflow response is accurate.
Faked services	2	Mimic	Services appear available for each OS	Mismatched services for OS platforms	Nessus and NMAP tests will identify services available, if they are not intended; they may be eliminated or changed to match the OS
Faked banner scripts	2	Mimic	Scripts appear as a typical banner upon scanning of the OS.	The banner does not match the OS or the service.	Test Nessus and NMAP scans to view all banner scripts that appear and if they are accurately matched to the OS and service.
Faked banner scripts	2	Mislead	Intruder believes the banner distinguishes the OS type and version.	The OS banner message is inappropriate or does not match the OS.	Conduct Nessus tests to ensure all banners are appropriate and correct and eliminate those that are inapt.
Honeyd logs	1	Camouflage	Passively gathers the honeyd logs without knowledge of the	Intruder is able to compromise the honeyd and gain access to the	Encrypt log files an save on a separate machine

			intruder.	logs.	
IDS	1	Camouflage	Passively logs all wireless network activity without knowledge of the intruder.	Intruder is able to compromise the IDS and gain access to the logs.	Save IDS logs on a separate machine.
FakeAP beacons	3	Mislead	Beaconing a faked SSID, MAC and IP.	Other legitimate APs that beacon the same SSID, MAC or IP.	Test other APs in the coverage area for their SSID, MAC and IPs, and ensure the FakeAP is different.
FakeAP beacons	3	Mimic	Appears as an existing AP.	Beacons incorrect MAC address.	Ensure the MAC address matches the intended Cisco vendor's first 6 hex digits.
FakeAP beacons	3	Decoy	Captures attention from existing APs.	AP beacons are not picked up by wireless sniffers.	Test AP beacons and strength at various lengths.
FakeAP beacons	3	Dazzle	Set the AP to flick through different MAC addresses and appear as many APs.	The IP of the flicking APs does not change.	This will limit the deception depending on the sniffing tool used, as some will only give the MAC and others also give the IP.

TABLE 5. 3 Experimental conditions for deceptive DNC and ONC

In Table 5.3 varying experimental conditions are described that will be deployed on the wireless honeypot. This will allow investigation of what conditions and how conditions vary by the type of attacks that will be launched against the deceptive network countermeasures.

5.3 Deceptive Network Attacks

Table 5.4 is the matrix of deceptive Network Defence Attacks (NDA) that will be carried out against the wireless honeypot. Address Resolution Protocol (ARP) poisoning, IP spoofing and MAC spoofing are methods of attack that deceptively conceal the real identity of the attacker with a faked identity. These defence attacks are used as a preliminary technique for ingress to the network (which in this case is the Honeyd virtual networks) to then execute offensive deceptive attacks.

Table 5.5 is the matrix of deceptive Network Offensive Attacks (NOA) that will be launched against the wireless honeypot. Offensive attacks may be brute forced such as continuous hits resulting in buffer overflows and other variants of Denial of Service (DoS) attacks. Other attacks such as NMAP and Nessus scans seek to interact with the victim to collect useful OS configuration and vulnerability information. Brute force attacks are frequently active because they are aimed to take out the victim.

A stealth attack such as a man-in-the-middle that initiates a single masked hit triggering a DoS, or breaking a wireless connection, is also active. A passive attack such as packet sniffing aims to inconspicuously intercept and capture data that is retained by the attacker. Packets will show data that include the network configurations that the attacker will use to tailor an active attack.

Network Attacks	Deceptive Defence																							
	Active												Passive											
	Static			Dynamic			Adaptive			Premed			Static			Dynamic			Adaptive			Premed		
	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?
ARP poisoning							✓																	
IP spoofing	✓																							
MAC spoofing	✓																							

TABLE 5. 4 Matrix of deceptive DNA

Network Attacks	Deceptive Offence																							
	Active												Passive											
	Static			Dynamic			Adaptive			Premed			Static			Dynamic			Adaptive			Premed		
	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?	👽	✥	?
Brute force buffer overflow			✓																					
Man-in-middle				✓																				
ICMPflood (DoS)			✓																					
Nessus scans									✓															
Nmap scans							✓																	
Stealth DoS	✓																							
Packet sniffing																✓								

TABLE 5. 5 Matrix of deceptive ONA

👽 = mask, camouflage, repackage; ✥ = mimic, mislead, decoy; **?** = confuse, dazzle, invent;

5.3.1 Application of the matrices of deceptive attacks

The matrices of deceptive DNA and ONA identified in Tables 5.4 and 5.5 are attack strategies that may be launched against the deceptive countermeasures on the wireless honeypot. Consequently, experimental conditions for deploying brute force attacks may be coordinated to test the effectiveness of deceptions used to defend the wireless honeypot.

Table 5.6 presents the experimental conditions for potential attacks derived from the matrices of deceptive attacks (Tables 5.4 and 5.5) that may be performed against the wireless honeypot. The experimental conditions that may be tested are demonstrated through:

- The deception in depth ring given for each attack, identifying the ring of the defence the attack is targeted to
- The distinguished brute force network attacks from stealth attacks
- Description of the function of the attacking deception in order to achieve the deceptive effect sought

[illegible] the framework	Ring	Brute	Stealth	[illegible] Effect Sought	[illegible] the Deception	How [illegible] accomplished
ARP spoofing	2	✓		Camouflage	Deceive ARP to believe the attacker has a legitimate MAC.	Gain access to honeypot's ARP table and assume one of the MAC addresses or insert a false MAC.
IP spoofing	2	✓		Camouflage	Appearing as a legitimate IP address.	Use downloadable tools to spoof the IP of the attacker.
MAC spoofing to connect to AP	2	✓		Camouflage	Appearing as a legitimate MAC address that may connect to the AP.	Ascertain permissible MAC addresses from an ARP table, or obtain MAC address from packet headers travelling the private network.
Brute force buffer overflow	1	✓		Dazzle	Overwhelm the memory stack of the target.	Exploiting a discovered flaw in the OS and targeting an excessive number of packets to dazzle and crash the OS.
Man-in-middle	2	✓		Camouflage	Intercept the connection without being detected.	Knock out the AP connection in a single concealed attack.
ICMP flood	2	✓		Dazzle	Send overwhelming numbers of ICMP packets.	Directing continuos ICMP packets or ICMP type floods to specific OS to cause a crash.
Stealth DoS	1		✓	Dazzle	Hit the OS with an unexpected solitary strike.	Send a single hit to cause the OS to crash.
Packet sniffer	2		✓	Mask	Inconspicuously sniff AP broadcast beacons.	Sniff packets to identify the AP MAC, IP and other configurations to tailor attacks.

TABLE 5. 6 Experimental conditions for DNA and ONA

5.4 Research Design of the Framework and Matrices

The research design of the framework and the matrices indicated that the active countermeasures should be implemented before the passive countermeasures. This is because the wireless honeypot must be proactively secured before enabling the correct deceptive response. Passive attacks will be initiated before active attacks to identify to the attacker, how to implement deceptive attacks against the victim.

This presents a method for carrying out the experimental brute force and stealth attacks against the wireless honeypot. Additionally, the effectiveness of deceptions will be observed to discover if conditions vary by the type of attack and under what conditions.

Thus, the wireless honeypot may be deployed following the deceptive network countermeasures identified in the DNC and ONC. Investigating the effectiveness of the deceptions on the wireless honeypot will be done through executing the deceptive network attacks identified in the DNA and ONA. This will enable the researcher to evaluate if the framework for deception may be applied to reduce the effectiveness of attacks, and the effectiveness of the wireless honeypot against brute force attacks. Additionally, the conditions identified in Tables 5.3 and 5.6 will be deployed to identify what conditions may vary by the type of attack.

6 CHAPTER 6 - ROUND 1 RESULTS

6.1 Baseline Testing

Round 1 testing involved a baseline test to determine if the wireless honeypot was able to be scanned or attacked with the given honeypot configurations and architecture. This was also to confirm if FakeAP and Honeyd were able to run concurrently.

The attacking machine utilised Linux Mandrake 9.0 with dual boot Windows XP to allow the attacker to utilise various attacking tools for both OS platforms. The attack machine was also equipped with a wireless PCMCIA card. The following wireless security tools were installed:

- Kismet
- Netstumbler
- NMAP
- Nessus

An attacker would typically use such tools, as they are free to download from the World Wide Web (WWW) and require little understanding of the technical functionality. The tools used for attacks against the wireless honeypot may be installed and used on a Linux or Windows machine.

6.2 Ring 1 FakeAP

Round 1 of testing involved FakeAP at the outer ring of the deception in depth for the wireless honeypot. Attacks identified in the matrix of deceptive attacks included the wireless sniffing tools for determining the existence of APs and their related configurations.

6.2.1 Kismet

Kismet is a wireless sniffing tool that was installed on the attacking machine. Kismet was able to pick up the beaconed 802.11b packets from a bogus access point, and identified the following information:

- The SSID of the bogus AP as "WinNT AP"
- The channel of the WinNT AP used which was channel 4
- The IP address of the WinNT AP as 192.168.1.99 which consequently informs the attacker of the gateway IP address

6.2.2 Netstumbler

Netstumbler is a wireless sniffing tool for Windows platforms, and was also used to verify the information sniffed by Kismet. Netstumbler was able to enumerate the following information on the rogue AP:

- The SSID as of the rogue AP as "WinNT AP"
- The MAC address of WinNT AP and recognised the vendor as Cisco
- The channel of the WinNT AP as channel 4

6.3 Implications of FakeAP testing

Testing in Round 1 FakeAP demonstrated that attacking tools Kismet and Netstumbler were able to sniff out the rogue AP. The attacking machine was able to identify information about the AP including an IP address of the AP gateway, and the SSID of the wireless network. The attacker could then reconfigure and spoof the IP of the attacking machine so that the attacking machine's IP may be on the same network as the rogue AP's. Subsequently the attacking machine's IP was changed to 192.168.1.253 so that the attacking machine could then conduct further probing of network resources through the AP gateway of 192.168.1.99.

6.4 Ring 2 Honeyd

Testing ring 2 of the deception in depth followed from ring 1, FakeAP testing. Therefore, the next level of attack encompassed stealth and brute force scans and probes of the private network, which were the Honeyd virtual networks. The goal of the Round 1 testing on ring 2, was also a baseline test to investigate if attacking tools NMAP and Nessus would be deceived by the deceptions deployed on the wireless honeypot, through Honeyd.

Two network scanning tools typically used by a script kiddie are NMAP and Nessus (Conry-Murray, 2001; Fratto, 2003; Noordergraaf, 2002) and were subsequently the attacking tools selected for network attacks on ring 2 of the wireless honeypot. The testing incorporated assessment of the outputs and reports generated from the attack tools.

A naïve attacker such as a script kiddie would use NMAP stealth scans by selecting a block of IP addresses to scan the network. NMAP then return results informing the attacker of the OS and platform that it believes is running on the IP address, and any interesting ports that should be noted; such as TCP port 23 (telnet) – open. An attacker could then be well informed on what OS's are on selected IP's and enter those IP's into Nessus for brute force scanning and probing.

Nessus will forcefully scan each IP address selected by the attacker and probe every specified port. Nessus then generates a report that details the OS and security information including security warnings such as banners, and vulnerabilities or holes that identify a dangerous threat exposure of a specific service or a TCP/IP port.

The deceptive capability of ring 2 testing was determined by how well NMAP and Nessus could be fooled into believing an OS existed, and if any security vulnerabilities existed on the OS.

6.4.1 Round 1 NMAP testing

NMAP (Fyodor, 2003b) scans networks for live hosts and offered services through the known sequence of TCP/IP handshakes of various OS's. Therefore, NMAP uses a method of OS fingerprinting to identify the networks of wired and wireless services that were associated with Honeyd. Table 6.1 describes the functions of the different NMAP scans (Fyodor, 2001) that were used to perform the NMAP OS fingerprinting on the Honeyd.

Scan method	Scan type	Description
-sS	TCP SYN	NMAP sends a SYN packet to half open a TCP connection, if the response is a SYN/ACK a RST will be sent to abandon the connection. This type of SYN technique is rarely logged.
-sT	TCP connect	TCP connections are attempted on all ports of a target machine.
-sU	UDP port scan	Determines which UDP ports are open when there is no ICMP port unreachable message.
-sF	FIN	Advanced stealth scans
-sX	XMAS	Closed ports reply with a RST
-sN	NULLn	Open ports ignore the packet
		Will not work on a Windows machine – hence identifying that the target machine is most likely a Windows OS.
-O	OS detection	Uses TCP/IP handshakes to make an OS 'fingerprint' and checks it against the NMAP OS fingerprint file.
-F	Fragmented packets sent	Splits the TCP header over more smaller packets as an attempts to evade detection.

TABLE 6. 1 NMAP v.3 scan types

SYN scans

A SYN scan was conducted on each IP address to determine if the Honeyd would deceptively respond by detecting the half-open connections and discarding them.

SYN scans, in combination with OS detection (-O), and fragmented packets (-F) would typically be used by the attacker to determine the OS of the machine, ports that are listening, and services running while evading detection by the target machine. The outcome of the scan may reveals if the attacker is fooled by the NMAP result returning a successful OS match.

UDP scans

UDP scans with OS detection and fragment determine which UDP ports are open, and are potential ports for placing a Trojan, or backdoor such as bo2k to listen on. An attacker would believe an open UDP port could be vulnerable and exploit the possibility of injecting malicious software. Therefore, NMAP, UDP scans attempts to identify which OS's would potentially be vulnerable to these types of attacks.

FIN, XMAS, and NULL scans

FIN, XMAS, and NULL scans attempt a further clandestine approach to port scanning and OS detection. Due to the technical configurations of Windows machines, these scans usually do not work. However, a failed scan may deceptively be construed by the attacker as a possible Windows 95 or 98 machine.

The aim of the NMAP scans was to determine if Honeyd could deceptively mislead NMAP into believing there were ports and services belonging to an OS of a machine. Subsequently an attacker using NMAP may also be deceived. This information would then be used to tailor an attack to the specified OS.

Table 6.2 shows the results of Round 1 of the NMAP scans that were conducted as a baseline test to ensure the machine could be probed wirelessly. The researcher chose IP addresses with no significance to the logical network topology and assigned them one of nine NMAP prescribed OS signatures, only to establish if the Honeyd virtual networks could be reached via a wireless NMAP scan. 10.3.1.15, 10.3.1.19 and 10.3.1.20 were unallocated IP address spaces which the researcher also performed

NMAP scans on to observe if the default OS assigned by ARPD, would also be detected.

The default OS was FreeBSD and was randomly selected to test if ARPD was able to pick up which IP addresses did not have a prescribed OS bound to the IP, and would automatically allocate the default. This formed part of the baseline testing of Round 1.

Three scans were conducted for each NMAP scan type, and on each IP address to check the reliability of results on each scan. It was found there were some software bugs in NMAP and Honeyd which created some inconsistencies, thus a fourth scan was conducted for the SYN and XMAS scans. Honeyd crashed on the second test for XMAS scans for an unknown reason, most likely due to a one-off software malfunction. Therefore, Honeyd was restarted and an extra XMAS scan was performed.

SYN scans

SYN scans were the most effective at guessing the correct remote OS. The Cisco router and hub OS's, Novell, AIX, and OpenBSD could all be successfully detected on all four SYN scans conducted. NMAP however, could not detect Solaris on any SYN scans, indicating a possible error in the Honeyd configuration file, or an unmatching NMAP fingerprint in the NMAP fingerprint file. Further results showed the NMAP OS guess for Cisco Aironet AP defaulted to FreeBSD on two SYN scans, and then crashed for the succeeding two SYN scans.

SYN scans performed on the Default OS, FreeBSD (IP addresses 10.3.1.19 and 10.3.1.20) resulted in two successful OS guesses; however, a third scan crashed which indicated NMAP was not able to guess the OS at all. A fourth SYN scan was conducted to check if the results of the third test was a chance outcome, though the result showed a crash again.

UDP scans

Successful NMAP, UDP scans, that were able to guess the correct OS, were only on the Cisco router, AIX, Novell and OpenBSD. All other OS's could not be guessed. This indicated that the NMAP fingerprint file possibly did not have an OS fingerprint record for UDP ports on those unsuccessful OS's.

FIN, XMAS, and NULL scans

FIN scans were able to guess the OS reliably in most cases, except one instance with Novell Netware (IP 10.3.1.18), but was not able to identify any of the IP addresses that were assigned the default FreeBSD, and Solaris (10.3.1.17). FIN and NULL scans also produced the default FreeBSD OS guess for the Cisco Aironet (10.3.1.200). However, all other scans on 10.3.1.200 resulted in a crash and no OS guess at all. Therefore, NMAP was overall ineffective at guessing the Cisco Aironet OS.

NULL scans could successfully identify the Cisco router, FreeBSD, OpenBSD, AIX, and Novell. The XMAS scans resulted in a correct remote OS guess for the Cisco router, AIX, OpenBSD and Novell. All other NULL and XMAS scans could not find an OS match for the fingerprint. This indicated that opened and closed ports on those OS's did not respond in an expected RST, or an ignore response, so that NMAP could identify the OS. Consequently, those OS's could not be identified by an NMAP, UDP scan.

IP Addr	OS Profile	SYN Stealth			[illegible] Scan			XMAS			[illegible]			UDP Scan			[illegible]
		✓	✗	D	✓	✗	D	✓	✗	D	✓	✗	D	✓	✗	D	
10.3.1.11	3com Office Connect Router 810	4	0	T	3	0	T	3	1	F	3	0	T	3	0	T	4
10.3.1.12	Cisco Router/Switch with IOS 11.2	4	0	T	3	0	T	0	4	F	0	3	F	0	3	F	2
10.3.1.13	FreeBSD 2.2.1-STABLE	2	2	F	3	0	T	0	4	F	3	0	T	0	3	F	2
10.3.1.14	AIX v4.2	4	0	T	3	0	T	3	1	F	3	0	T	3	0	T	4
10.3.1.15	Default FreeBSD 2.2.1-STABLE	0	4	F	0	3	F	0	4	F	0	3	F	0	3	F	0
10.3.1.16	OpenBSD 2.7/SPARC or NFR IDS Appliance (12/10/00)	4	0	T	3	0	T	0	4	F	3	0	T	3	0	T	4
10.3.1.17	Solaris 2.6-7 X86	0	4	F	0	3	F	0	4	F	0	3	F	0	3	F	0
10.3.1.18	Novell Netware 3.12 – 5.00	4	0	T	3	0	T	3	1	F	3	0	T	3	0	T	4
10.3.1.19	Default FreeBSD 2.2.1-STABLE	2	2	F	3	0	T	0	4	F	3	0	T	0	3	F	2
10.3.1.20	Default FreeBSD 2.2.1-STABLE	2	2	F	3	0	T	0	4	F	3	0	T	0	3	F	2
10.3.1.200	Aironet AP4800E v8.07 – Aironet (Cisco?) 11 Mbps wireless access point	0	4	F	0	3	F	0	3	F	0	3	F	0	3	F	0
Total / 11				5			8			0			7			4	44

TABLE 6. 2 Results of Round 1 NMAP scans

✓= number of correct OS guesses; ✗ = number of incorrect OS guesses or no OS match; **D =** a Boolean value (**T**rue or **F**alse) indicating if the deception was achieved through a correct OS guess on each scan;

The Round 1 NMAP scans indicated that NMAP is was always reliable on a single scan attempt; several scan attempts were required to check the reliability of scan results. Additionally, not all scan types could successfully identify a correct OS match. This indicated that not all the NMAP OS fingerprints have the ability to detect the OS when attempting unconventional TCP connections.

6.4.2 Round 1 Nessus testing

Round 1 testing using Nessus aimed to determine if Nessus would be deceived by returning the OS platform with a list of the services available, as specified by the Honeyd configuration file. Nessus also utilises NMAP scanning however unlike NMAP, Nessus will not assume that a given service will operate on an expected TCP/IP port. Nessus will test the security of every port regardless of the version number of the service (Deraison, 2003b). This allows Nessus to generate information on the security warnings, holes and vulnerabilities which in turn provides significant OS exploit opportunities to a would be attacker.

For the purpose of the Round 1 Nessus baseline testing, it was only necessary for Nessus to report the matching remote OS guess and exploitable security information on ports/services set in the Honeyd configuration file for each OS.

Three Nessus scans were conducted on each IP to ensure validity. All three tests conducted on each of the IP's returned the same results indicating the Nessus scans gave consistent results. Each Nessus scan generated a report that listed the TCP/IP ports opened and any related security warnings, vulnerabilities (security holes), and the level of the security threat posed (indicated as a low, medium or high). Table 6.3 is a table of the results from the Round 1 Nessus scans.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
10.3.1.1	3com Office Connect Router 810	telnet 23/tcp	1	1				1
10.3.1.12	Cisco Router/Switch with IOS 11.2	telnet 23/tcp	1	1				
10.3.1.13	FreeBSD 2.2.1-STABLE	ssh 22/tcp http 80/tcp	1	1				
10.3.1.14	AIX 4.0 - 4.2	ftp 21/tcp http 80/tcp		1				1
10.3.1.15	Solaris 2.6 - 7 (SPARC)	http 80/tcp		1				
10.3.1.16	OpenBSD 2.6-2.8	http 80/tcp	1	1				
10.3.1.17	Solaris 2.6 - 7 (SPARC)	http 80/tcp		1				
10.3.1.18	Novell Netware 5.x	http 80/tcp		1				1
10.3.1.19	FreeBSD 2.2.1-STABLE	ssh 22/tcp http 80/tcp	1	1				
10.3.1.20	FreeBSD 2.2.1-STABLE	ssh 22/tcp http 80/tcp	1	1				
10.3.1.200	FreeBSD 2.2.1-STABLE	ssh 22/tcp http 80/tcp	1					

TABLE 6. 3 Results from Round 1 Nessus scans

Nessus was able to successfully guess all but two OS's and report security holes, warnings, and vulnerabilities that applied to services run on each OS. Nessus scans performed on the Cisco router and hub/switch (10.3.1.1 and 10.3.1.12) returned telnet service warnings and one security vulnerability/hole. Nessus was also able to pick up the default OS (FreeBSD), AIX, Solaris (10.3.1.17), OpenBSD and Novell with at least one warning on each OS.

Nessus indicated IP 10.3.1.15 as a Solaris OS when in fact the default OS (FreeBSD) would have been the correct OS guess. Additionally IP 10.3.1.200 came up as the default OS and not the Cisco Aironet AP. These errors indicated that the Honeyd configuration file most likely had an error with the Solaris and Cisco Aironet personality signatures that would need correcting for Round 2 testing.

6.5 Implications of Round 1 Testing

The overall results from Round 1 testing reported on attacks and intrusions that were conducted on ring 1 of the defence, FakeAP and ring 2, Honeyd. Results from ring 1 testing showed that the fake access point could be sniffed and identified by the wireless sniffing tools Kismet and Netstumbler. The information collected by the sniffing tools identified the existence of an AP by the SSID, IP address, MAC address, channel used, and a potential gateway to a private wireless network (Honeyd virtual networks).

Results from ring 2 testing utilised network scanning tools NMAP and Nessus to enumerate remote OS guesses of a private network (Honeyd virtual networks) and their matching IP's by using various stealth scans. Nessus was able to give further OS security warnings, vulnerabilities and holes on services running on the scanned OS through brute forced probes.

Consequently, Round 2 of the testing will involve further NMAP and Nessus probes on a reconfigured Honeyd network to investigate if the scanning tools may detect all the OS's.

7 CHAPTER 7 - ROUND 2 RESULTS

7.1 Reconfiguring Honeyd

For Round 2 testing, Honeyd was reconfigured to reflect a more realistic layout of a corporate wireless network. Figure 7.1 illustrates the revised Honeyd virtual networks as a logical configuration. The Linux Mandrake 9.0 machine was maintained as the honeypot installed with FakeAP and Honeyd. The gateway IP address of FakeAP, which is also the access point gateway to the Honeyd virtual networks, remained as 192.168.1.99. The network topology of routers, servers, and client machines are presented as a structured logical layout in Figure 7.1. This model was validated as a practicable network configuration by (Dawson, 2003; Valli, 2003a, personal communication).

The logical network topology of the Honeyd virtual networks encompasses three subnets separated by Routerone, a hub and Routertwo. The first network (network address 10.1.0.0) utilises Routerone with IP address 10.1.1.1 as the first route entry into the network. Only the Cisco Aironet AP shares the same network address as Routerone.

The second subnet has the network address of 10.2.1.0 and is intended to be seen as a corporate Demilitarised Zone (DMZ) containing all the network servers. Six OS platforms are allocated to six IP address spaces in this network. A second linked route entry using a hub (IP 10.2.1.1) separates the OS platforms Linux, AIX, OpenBSD, Solaris, FreeBSD and Novell. These OS's were chosen to represent a variety of platforms with testable vulnerabilities. Changes to the application level involved the appropriate Web service for each OS platform, which is identified as APACHE for the majority of OS's.

The third subnet uses the network IP address of 10.3.1.0 and a connecting route entry via Routertwo (IP 10.3.1.1). This subnet encompasses a network of all the unallocated IP spaces that will be assigned the default OS through ARPD. Therefore, all IP address spaces in network three are Windows 98 machines. The purpose of this network is to appear as an assembly of Windows 98 client machines that access the servers and services from the DMZ. Table 7.1 identifies the precise NMAP OS signature, and accompanying technical configurations for the revised Honeyd virtual networks.

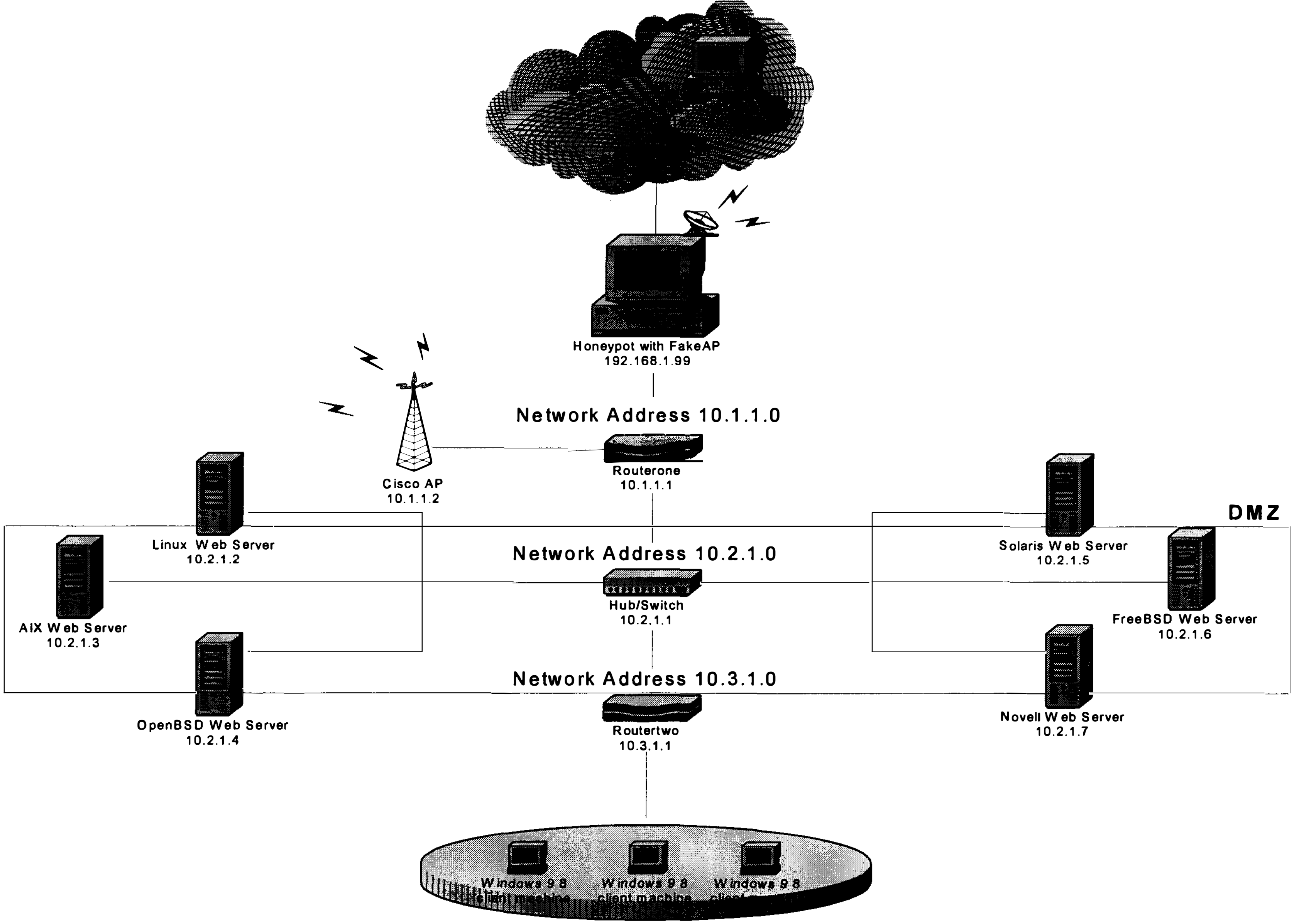


FIGURE 7. 1 Revised honeyd virtual networks - logical configuration

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Router one	3com Office Connect Router 810	10.1.1.0	10.1.1.1	TCP 23 - TELNET	Telnet banner
Hub	Cisco Router/Switch with IOS 11.2	10.2.1.0	10.2.1.1	TCP 23 - TELNET	Telnet banner
Router two	Cisco 726 Non-IOS Software release 4.1(2) or 766 ISDN router	10.2.1.0	10.3.1.1	TCP 23 - TELNET	Telnet banner
AP	Aironet AP4800E v8.07 – Aironet (Cisco?) 11 Mbps wireless access point	10.1.1.0	10.1.1.2		
Solaris	Solaris 2.6 – 7 X86	10.2.1.0	10.2.1.5	HTTP 80 - WEB	Apache 1.3.9
FreeBSD	FreeBSD 2.2.1-STABLE	10.2.1.0	10.2.1.6	HTTP 80 - WEB	Apache 1.3
Novell	Novell Netware 3.12 – 5.00	10.2.1.0	10.2.1.7	HTTP 80 - WEB	Apache 1.3.27
Linux	Linux Kernel 2.4.0 – 2.4.18 (X86)	10.2.1.0	10.2.1.2	HTTP 80 - WEB	Apache 1.3.26
AIX	AIX v4.2	10.2.1.0	10.2.1.3	HTTP 80 – WEB TCP 21 - FTP	Apache 1.3.9
OpenBSD	OpenBSD 2.7/SPARC or NFR IDS Appliance (12/10/00)	10.2.1.0	10.2.1.4	HTTP 80 - WEB	Apache 1.3.12
Default	Windows98 w/ Service Pack 1	10.3.1.0	Unassigned		

TABLE 7. 1 Revised Honeyd virtual networks - technical configuration

7.2 Round 2 Testing

7.2.1 Round 2 NMAP testing

The purpose of Round 2 NMAP testing was to investigate if the reconfigured Honeyd could deceive NMAP. This would be achieved through NMAP guessing the correct remote OS using the same scan techniques. Corrections were made to the Honeyd configuration file to remove errors and ensure that all the scripts were accurately matched to an NMAP prescribed signature. The same five NMAP scans, SYN, FIN, XMAS, NULL and UDP were conducted as in Round 1, however for Round 2, five scans were performed to validate the final NMAP scan/test results. Table 7.3 shows the findings from the Round 2 NMAP scans.

IP Number	Platform	SYN Stealth			FIN Stealth			XMAS			Null			UDP			Total
		✓	×	D	✓	×	D	✓	×	D	✓	×	D	✓	×	D	
10.1.1.1	3com Office Connect Router 810	5	0	T	5	0	T	0	5	F	0	5	F	0	5	F	2
10.1.1.2	Aironet AP4800E v8.07 – Aironet (Cisco?) 11 Mbps wireless access point	5	0	T	5	0	T	5	0	T	5	0	T	5	0	T	5
10.1.1.3	Windows98 w/ Service Pack 1	4	1	F	0	5	F	0	5	F	0	5	F	0	5	F	0
10.1.1.0	Network Address	4	1	F	0	5	F	0	5	F	0	5	F	0	5	F	0
10.2.1.1	Cisco Router/Switch with IOS 11.2	5	0	T	5	0	T	0	5	F	0	5	F	0	5	F	2
10.2.1.2	Linux Kernel 2.4.0 – 2.4.18 (X86)	5	0	T	5	0	T	5	0	T	4	1	F	5	0	T	4
10.2.1.3	AIX v4.2	5	0	T	5	0	T	5	0	T	5	0	T	5	0	T	5
10.2.1.4	OpenBSD 2.7/SPARC or NFR IDS Appliance (12/10/00)	5	0	T	5	0	T	5	0	T	5	0	T	5	0	T	5
10.2.1.5	Solaris 2.6 – 7 X86	0	5	F	0	5	F	0	5	F	0	5	F	0	5	F	0
10.2.1.6	FreeBSD 2.2.1-STABLE	0	5	F	0	5	F	0	5	F	0	5	F	0	5	F	0
10.2.1.7	Novell Netware 3.12 – 5.00	5	0	T	5	0	T	5	0	T	5	0	T	5	0	T	5
10.2.1.8	Windows98 w/ Service Pack 1	3	2	F	0	5	F	0	5	F	0	5	F	0	5	F	0
10.2.1.0	Network Address	3	2	F	0	5	F	0	5	F	0	5	F	0	5	F	0
10.3.1.1	Cisco 726 Non-IOS Software release 4.1(2) or 766 ISDN router	5	0	T	5	0	T	0	5	F	0	5	F	0	5	F	2
10.3.1.2	Windows98 w/ Service Pack 1	3	2	F	0	5	F	0	5	F	0	5	F	0	5	F	0
10.3.1.3	Windows98 w/ Service Pack 1	3	2	F	0	5	F	0	5	F	0	5	F	0	5	F	0
Total /16				8			8			5			4			5	60

TABLE 7. 2 Results from Round 2 NMAP scans

✓= number of correct OS guesses; × = number of of incorrect OS guesses or no OS match; **D =** a Boolean value (**T**rue or **F**alse) indicating if the deception was achieved through a correct OS guess on each scan

SYN scans

The SYN scan was conducted first with results that differed from the Round 1 NMAP, SYN scans. The Cisco routers, AP, hub, Linux, AIX, OpenBSD and Novell could all be successfully fingerprinted. However, the Solaris and FreeBSD OS platforms could not be guessed by any NMAP, SYN scan. Additionally, the default Windows 98 IP's were successfully fingerprinted until half way through the 4th round of SYN scans, when NMAP continuously crashed on all IP's scanned with the default OS of Windows 98.

An interesting software anomaly found in the NMAP scans was that the network addresses were also scanned (on all scan types), although the network address was not specified to be scanned by the researcher. As the network addresses were not allocated any instructions in the Honeyd configuration file (such as a *request timed out* message), the default OS should have been assigned by ARPD.

For half the SYN scans conducted, each network address scanned returned the default OS of Windows 98. However, as the OS guesses on the default designated IP's began to crash half way through the NMAP SYN scans, returning an unsuccessful NMAP fingerprint and subsequently no OS guess, so did the network addresses.

FIN scans

Round 2 of FIN scans indicated the NMAP was able to successfully fingerprint and guess the correct remote OS for both Routers, the Cisco Aironet AP, hub, Linux, AIX, OpenBSD and Novell. Failed NMAP fingerprints using a FIN scan could not identify Solaris, FreeBSD and any of the default OS allocated IP addresses.

XMAS, NULL, and UDP stealth scans

Round 2 of the XMAS, NULL and UDP scans returned almost identical results, except the first NULL scan conducted on Linux; which NMAP could not fingerprint.

XMAS, NULL, and UDP scans were able to guess the correct OS for the AP, Linux, AIX, OpenBSD, and Novell. The Cisco Routers, hub, Solaris, FreeBSD, and all default IP addresses could not be fingerprinted.

Another NMAP abnormality that became apparent was in the XMAS scans where the network addresses were scanned in-between each selected scanned IP in the network. The number of scans conducted on each network address is given below:

- 10.1.1.0 – scanned 3 times
- 10.2.1.0 – scanned 8 times
- 10.3.1.0 – scanned 2 times

It was not known why the NMAP XMAS scans did this, although it may be attributable to a software error.

7.2.2 Round 2 Nessus testing

Round 2 of the Nessus scans was aimed to investigate if Nessus could be deceived by the revised Honeyd, and potentially appear realistic to an attacker scanning the network. Nessus was required to guess the remote OS platform and any security information that would be exploitable for a would-be attacker.

For the Round 2 testing, five scans were attempted on each OS allocated IP address in Honeyd. However, it was found that Nessus was not able to scan the IP 10.1.1.1 (Routerone) at all, and any default OS assigned IP addresses, such as in network 10.3.1.0 (excluding 10.3.1.1, which is Routertwo) would take an extensively long time and resulted in a Nessus Crash. The 3com Office Connect Router 810 on IP 10.1.1.1 would stop and close at every attempt made by the researcher to scan the IP. Consequently, the results shown in Table 7.4 exclude the OS platforms, and IP addresses that Nessus could not scan.

IP	Remote OS guess	Open Ports found	Number of Warnings			Number of Vulnerabilities		
			[illegible]	Yes	Unsure	[illegible]	[illegible]	[illegible]
10.1.1.2	No OS guess		1					
10.2.1.1	Cisco Router/Switch with IOS 11.2	telnet 23/tcp		1				
10.2.1.2	Linux Kernel 2.4.0 - 2.5.20	http 80/tcp		1				
10.2.1.3	AIX v4.2	ftp 21/tcp http 80/tcp		1				
10.2.1.4	OpenBSD 2.7/SPARC or NFR IDS Appliance (12/10/00)	http 80/tcp	1	1				
10.2.1.5	Solaris 2.6 - 7 X86	http 80/tcp	2	1				
10.2.1.6	No OS guess		1					
10.2.1.7	Novell NetWare 3.12 - 5.00	http 80/tcp		1				1
10.3.1.1	Cisco 762 Non-IOS Software release 4.1(2) or 766 ISDN router	telnet 23/tcp	1	1				1

TABLE 7. 3 Results from Round 2 Nessus scans

In the Round 2 Nessus tests, each Nessus scan generated a report that listed the ports opened and any related security information such as the warnings, vulnerabilities (security holes), and the level of the security threat posed. The results in Table 7.4 represent the outcomes from all five scans conducted on each of the IP addresses listed.

Each of the 5 scans performed on all the listed IP's produced the same findings except for the 4th Nessus scan conducted on IP 10.2.1.1 (Cisco Router/Switch with IOS 11.2), which produced two security warnings. The other four scans performed on 10.2.1.1 produced only one warning, as demonstrated in Table 7.4. In the Round 2 Nessus testing, this was the only irregular finding. This indicated that the Nessus software was subject to some differences when used continually to scan IP addresses.

However, Nessus was still able to guess the remote OS for the Cisco Routers, Linux, AIX, OpenBSD, Solaris, and Novell. Additionally, Nessus found at least one security vulnerability and/or vulnerability (hole) on each of the OS's for a potential attacker to investigate further and exploit.

Nessus was not able to guess the OS for IP 10.2.1.6 which was FreeBSD. It is probable that Nessus was unsure of the exact OS on the IP 10.2.1.6; however, a warning was still found from the Nessus probes performed on that OS. Below is the warning generated by the Nessus scan performed on 10.2.1.6:

Warning found on port general/tcp

```
The remote host uses non-random IP IDs, that is, it is
possible to predict the next value of the ip_id field of
the ip packets sent by this host.

An attacker may use this feature to determine if the remote
host sent a packet in reply to another request. This may be
used for portscanning and other things.
```

```
Solution : Contact your vendor for a patch
Risk factor : Low
Nessus ID : 10201
```

As the attacker usually cannot differentiate if this warning is a false-positive, it may not matter if the OS was undetectable. The deception is this instance is that the warning provides a misleading vulnerability of the potential OS. This may still be exploited by an attacker by searching the vulnerability on the Internet to discover how it may be exploited. Example sites are CERT, Bugtraq, CVE and SecuriTeam vulnerability search engines and databases.

It was not known why IP 10.1.1.1, Routerone with the OS of 3com Office Connect Router 810, could not be scanned at all. Nessus would attempt to initiate a connection but then drop the scan without producing any findings. Additionally, Nessus was ineffective at scanning a target machine with the default OS allocated to the IP. Nessus would complete the scan after an extensively long time. Each scan usually had duration of approximately 15minutes. The Windows 98 IP addresses lasted for up to 45 minutes, and Nessus would then crash without producing a generated report of results.

7.2.3 Results from the Central Logging Structure (CLS)

The intended purpose of the CLS was to provide a supplementary method of data collection to triangulate and verify attacks made on the wireless honeypot. The CLS comprised of the Honeyd log files, which demonstrated themselves to be lacking in the richness of information given. The Honeyd logs only indicated that TCP/IP ports had some activity, and did not give further information of the possible attacks being performed on the network level. Therefore, the researcher discovered that the Honeyd log files were not effective in aiding the deceptions.

The SNORT log files verified the attacks achieved by Nessus. This was useful for the researcher as it provided triangulated verification of the Nessus generated reports of scans. This also aided the deceptions through camouflaged that concealed the victim's knowledge of attacker activity.

7.3 Implications of Round 2 Testing

The overall results gathered from Round 2 testing focussed on NMAP and Nessus scans and probes. Honeyd was reconfigured to overcome suspected errors found in Round 1, NMAP and Nessus tests. However, for Round 2 testing, Honeyd did not demonstrate that it could deceive effectively all the NMAP and Nessus attacks.

Both NMAP and Nessus revealed software abnormalities through performing additional scans that were not requested by the researcher, and refusing to perform scans that were requested by the researcher. However, NMAP was able to detect three OS's across all the five scan types out of sixteen that were scanned, and they were AIX, OpenBSD, and the Cisco Aironet AP. This indicated that some of the Honeyd deceptions were successful in fooling NMAP. Nessus reported at least one security warning or vulnerability on all the OS's that it could perform a complete scan. Honeyd was therefore effective at deceiving Nessus for only nine out of the possible twelve scans.

8 CHAPTER 8 – DISCUSSION OF RESULTS

Results from Round 1 and Round 2 testing are combined in a discussion relating to the research questions:

1. Can a framework for deception be applied to common network countermeasures to reduce the effectiveness of attacks?
2. How effective is deception in a wireless honeypot against brute force attacks?
3. Under what conditions, and do conditions vary by the type of attack?

8.1 Can a Framework for Deception be applied to Common Network Countermeasures to Reduce the Effectiveness of Attacks?

8.1.1 Ring 1 - FakeAP

The deceptions for FakeAP resulted as highly effective countermeasures against the network attacks of the wireless sniffing tools Kismet and Netstumbler. The framework was used to determine how to implement the deceptions for FakeAP. The identified deceptions were mimicry, misleading, and decoying.

The forensic data that was collected from Kismet and Netstumbler indicated they were both misled into believing a bogus AP existed. This was done by FakeAP's mimicry of a real AP. It is also likely that the decoyed deception would be achieved, if a potential attacker were able to sniff the bogus AP. This is because the attacker consumes time and resources when sniffing the fake AP; and consequently, the attacker may become deterred from a real AP.

A script kiddie using Kismet or Netstumbler may be misled into thinking a real AP existed. Furthermore, the potential attacker could think there is a network behind the AP that may be scanned, probed, and exploited. Thus, the deceptions implemented on FakeAP, which were identified from the framework, were effective countermeasures on the wireless honeypot against the attacking tools Kismet and Netstumbler.

8.1.2 Ring 2 - Honeyd

The deceptions employed by the Honeyd virtual networks were not effective, on every occasion, in deceiving the attack tools used. The framework identified mimicry and misleading deceptions for the Honeyd countermeasures. This was to be demonstrated through faked: OS's, login-banners, services, and OS vulnerabilities.

The attacking tool NMAP was used first to determine if it would be able to guess the OS on each of the IP's scanned. Subsequently, this was used to determine if Honeyd was able to mimic all the OS's.

Evaluating the framework in reducing the effectiveness of NMAP attacks

NMAP was unable to detect all the OS's in both Round 1 and Round 2 testing. However, through the application of the framework, the researcher found a highly useful method for understanding how to deploy Honeyd deceptions. The researcher learned the value of OS mimicry as a significant network countermeasure against attacks. This was demonstrated by NMAP's ability to fingerprint three of the Honeyd OS's.

In Round 2 of the NMAP scans, the Cisco Aironet AP, AIX, and OpenBSD OS's indicated a highly effectively deception achieved, through OS mimicry. This was demonstrated through successful OS guesses from NMAP, for each scan that was conducted across all the five scan-types.

NMAP believed that the Cisco Aironet AP, AIX, and OpenBSD OS's were real. A script kiddie would typically rely on NMAP to distinguish what is real and what is not. Subsequently the attacker may begin probing those OS's for vulnerabilities using a tool like Nessus. However, as the OS's are not real, the attacker would be misled into wasting time and resources on the faked OS's.

Therefore, in applying the framework to the wireless honeypot, it was determined that OS mimicry was an effective deception used as a network countermeasure. This was demonstrated through the successful mimicry of three OS's using Honeyd, against the attacking tool NMAP. Additionally, effective OS mimicry would also achieve a misleading deception for the potential script kiddie.

Evaluating the framework in reducing the effectiveness of Nessus attacks

The framework was also used to identify deceptions including faked: OS vulnerabilities, security holes, login-banners, and services through mimicry and misleading deceptions. These Honeyd deceptions were tested against Nessus scans. The results showed that Nessus was not able to identify all the OS's, although, was able to find at least one OS security warning or vulnerability for each of the scanned OS's.

The mimicry of the faked login-banners on the Cisco routers appeared real, in addition to the faked security warnings, and service vulnerabilities that Nessus found on each of the OS's. These deceptions would typically fool a script kiddie. This is because they would not normally be able to distinguish the difference between a false-positive OS weakness, and a genuine OS weakness. The result would be that the attacker would think any OS weakness found by Nessus would be an opportunity for exploit.

The results from the Nessus reports were valuable in identifying the level of deception achieved by Honeyd. The results revealed which countermeasures would require a greater level of mimicry and misleading in order to fool the network attack tool Nessus. An example was the web service run on each of the servers, which Nessus identified as IIS v.5 in Round 1. However, most of the server platforms should have been using the APACHE web service. Therefore, this would was changed to facilitate an effiective deception of misleading the attacker for Round 2.

The framework was thus useful in identifying how each Honeyd countermeasure may be implemented to deceive Nessus attacks. Additionally, the researcher found the framework to be highly useful for determining which deceptions may be deployed on the wireless honeypot, against the attack tools NMAP, Kismet and Netstumbler.

It may also be assumed that the framework would be useful for determining how to implement deceptions to fool other network attacking tools. These include wireless sniffing tools such as Dsniff, WaveLAN, and AirMagnet. Network security tools that may also be used for attacking include SATAN, Whisker and WebTrends Security Analyzer. therefore according to the results gathered by the experiment, the researcher found that the framework for deception may be applied to common network countermeasures to reduce the effectiveness of attacks.

8.2 How effective is Deception in a Wireless Honeypot against Brute Force Attacks?

The brute force attacks that were tested against the wireless honeypot were primarily Nessus scans and probes. Nessus uses NMAP stealth scans to guess the remote OS and then forcibly probes each service with any known vulnerabilities. Nessus was able to detect at least one security warning or vulnerability on each of the OS's scanned. However, Nessus was not able to guess the correct OS on all scans, and in addition was not able to scan all the OS's selected by the researcher.

In Round one of the Nessus testing, Nessus did not guess the OS for the Cisco Aironet AP and guessed the default OS instead. It was discovered that the Honeyd configuration file had an error and this was changed for the Round two testing. However, after the reconfiguration of Honeyd for Round two testing, Nessus produced less successful results. Nessus was not able to scan all the OS's and again, was not able to guess the Cisco Aironet AP.

The researcher could not identify why the Nessus scans were less effective in Round 2 of the testing, than in Round 1. Therefore, the researcher evaluated the effectiveness of NMAP scans as an aid to understanding the possible causes for the irregularities found in the Nessus results.

It was also found that the NMAP scans were less successful at guessing the OS's in Round 2 of the testing, than in Round 1. Even after correcting the Honeyd configuration file, not all the NMAP scans were able to successfully fingerprint and guess the OS's. Moreover, the percentage of correct OS guesses achieved in the first Round of testing was higher at 43.6%, in comparison to the percentage of correct OS guesses achieved in the second Round of testing; which was 33.75%.

Furthermore, Figure 8.1 depicts a comparison of NMAP scan results for each scan type performed in Round 1 and Round 2 tests. The diagram identifies that the SYN scans conducted showed approximately 50% successful scans over the two Rounds of testing. FIN and NULL scans were significantly improved in Round 1, when compared to Round 2 results. UDP scans were reasonably more successful in Round 1, and XMAS scans produced some successful guesses in Round 2, compared to none in Round 1.

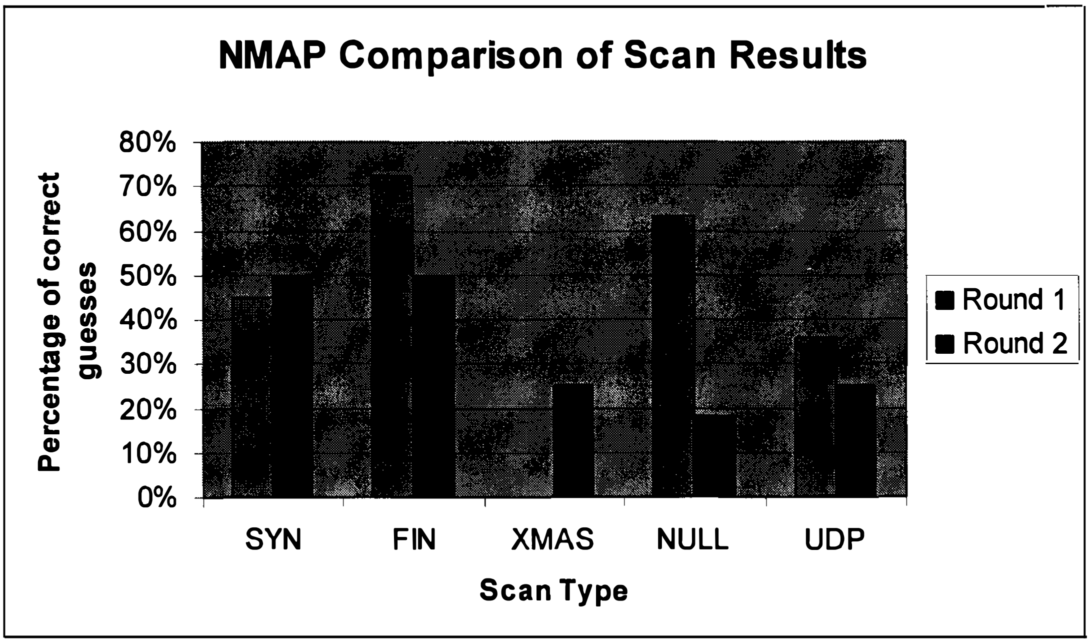


FIGURE 8. 1 A comparison of NMAP scans performed

The implications of these NMAP results indicated that in the experiment, NMAP was could not reliably fingerprint the Honeyd OS's across the five scan types. No correlated relationship between the IV and DV's of Honeyd deceptions may be drawn from these results.

The NMAP OS fingerprints selected for Honeyd were previously tested against all five scan types in a wired environment. Each OS fingerprint was able to be successfully detected by the NMAP SYN, FIN, XMAS, UPD and NULL scans (Valli, 2003b). However, the results showed that NMAP was not able to fingerprint successfully the OS's in a wireless environment using the same five scan types.

This may be attributable to a number of factors. The researcher's opinion is that the network packet sequence and exchange over the wireless medium most likely caused the irregularity in NMAP scan results. The open air allows a greater possibility of disordered packet sequencing and loss of packets, than over wired media.

Another possibility may be that NMAP (the software) begins to crash when it is used excessively for prolonged periods. Many of the NMAP scans were performed on up to 16 IP spaces at a time, and scan types were executed consecutively. Additionally, the ratio of correct OS guesses in the first Round of testing was higher than that of the second Round of testing; which was 34 more scans.

Thus, the increased amount of scanning performed may have initiated a software overload and produced erroneous packets. Therefore, NMAP's inability to guess correctly each of the OS's, may have been a carry-over effect on the Nessus scans. This is because Nessus also utilises NMAP OS fingerprinting to guess the remote OS in each scan, and then Nessus performs the brute force probing for vulnerabilities.

Nessus however, was able to guess correctly each of the OS platforms for all the scans it completed, for both Round 1 and Round 2 testing. Nessus also generated at least one security warning or vulnerability. Figure 8.2 outlines the scans results of both the Nessus scans conducted. The majority of Nessus scans produced the correct OS, with security warnings or vulnerabilities. There were no more than two incorrect guesses at a time and only in Round 2 of the Nessus scans, where there null/no attempts. This indicated that Honeyd was effective on most occasions, at deceiving Nessus; however, not in every case.

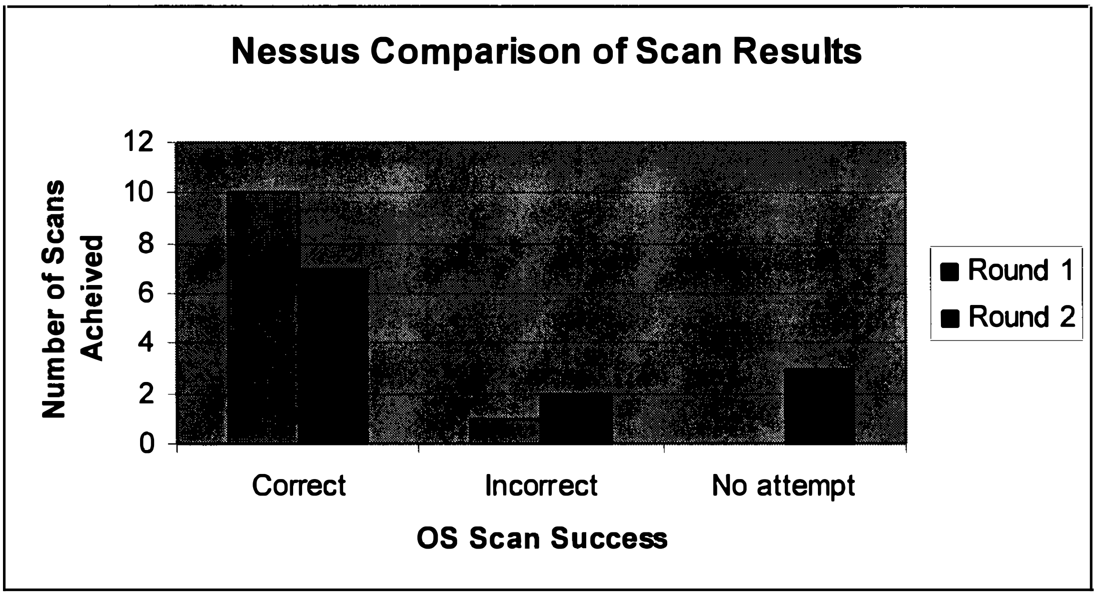


FIGURE 8. 2 A comparison of Nessus scans conducted

Therefore, the NMAP stealth scans and Nessus brute force attacks identified a possible anomaly in the way wireless network scanning may be conducted. Consequently, the results ascertained from the NMAP and Nessus tests indicated that the deceptions used in the wireless honeypot were not effective on all occasions, against the brute force attacks. This was demonstrated by the various discrepancies found in the scan results, and the unpredictability of the wireless experimental environment discussed above.

8.3 Under what Conditions and do Conditions vary by the type of Attack?

The researcher determined how each experimental condition for each type of attack was applied against the wireless honeypot. These were described in the research design through the rings of deception in depth, the way in which the deception would be deployed, and any random variables that may distort the results.

8.3.1 Conditions for FakeAP attacks

The first Round of testing utilised Kismet and Netstumbler wireless sniffing tools for attacking ring one of the deception, FakeAP. The experimental research design identified the conditions for attack. This included the configurations of the attacking machine, and the FakeAP configurations. These conditions were chosen to test if the deceptions on FakeAP would fool the wireless sniffing tools used. The results demonstrated that FakeAP was able to mimic a real AP and thus, was able to fool the attack tools used.

The results from FakeAP testing on the wireless honeypot indicated that wireless sniffing of an AP is an effective attack. This may be because the wireless environment and the nature of the wireless tools used were conducive for AP sniffing. Additionally, the researcher found that some conditions may vary by the attack, however, they are the configurable parameters used by FakeAP.

The conditions that may be changed on FakeAP include testing of varying: distances from the AP, signal strengths, and channels. These could be tested against different wireless sniffing tools such as Dsniff, WaveLAN, and AirMagnet. Therefore, if FakeAP is able to effectively mimic a real AP, other wireless sniffing-type attacks may be deceived in a similar way.

8.3.2 Conditions for NMAP and Nessus attacks

The conditions for Honeyd identified OS and network mimicry, OS vulnerabilities, and TCP/IP fingerspoofing as countermeasures for NMAP and Nessus wireless attacks. It was found that NMAP and Nessus attacks did always achieve successful results against Honeyd, within the wireless environment.

NMAP and Nessus are security tools that are also utilised for attacking purposes through TCP/IP port scanning and probing for vulnerabilities in an OS. This type of attack varies from the wireless sniffing tools that were used on FakeAP. FakeAP utilised the wireless medium effectively, to carry out the attacks against the wireless honeypot. However, NMAP and Nessus were originally designed for the wired network environment.

The researcher found the nature of network packets travelling through the air may be a disruption to the way NMAP and Nessus scans are performed. Therefore, the conditions for NMAP and Nessus attacks do vary in the wireless environment. The conditions may be that NMAP and Nessus require a fixed medium to execute effective scans that are able to guess the correct OS and vulnerabilities on every occasion. This is likely because NMAP and Nessus rely on the packet sequencing of TCP/IP handshakes to determine what the OS is. Therefore, if wireless packets are more frequently lost, and packets are not received in the correct order, the likelihood of failed OS guesses may be higher. The results from the NMAP and Nessus scans support this possible deduction.

Consequently, the conditions for attacks perpetrated on the wireless honeypot did vary by the type of attack. The researcher came to the conclusion that wireless sniffing tools utilise the wireless medium effectively because they are designed to enumerate static information on the physical layer (of the OSI), which is the open air, also used by FakeAP. However, network-attacking tools such as NMAP and Nessus require a more reliable medium to conduct interactive and dynamic packet exchange. Therefore, these attacks may require a stable network layer (of the OSI) of TCP/ IP connectivity in order to achieve effective Honeyd attacks.

This is usually the case with wired media. However, the wireless medium does not contain electromagnetic waves through rigid confines of a copper or optic cable, and instead, allows the data signals to disperse with abandon. Therefore, the conditions for NMAP and Nessus attacks varied significantly, on the stability of the software and the network environment.

9 CHAPTER 9 – CONCLUSION

This research entitled A Deception Based Framework for the Application of Deceptive Countermeasures in 802.11b Wireless Networks was experimental research aimed to improve understanding of how deceptive methods may be used in network defence. The researcher adopted an exploratory method for testing logical and technical concepts to investigate the effectiveness of deceptions deployed on a wireless honeypot. The results indicated how deceptions might be used as network countermeasures against common brute force attacks.

The countermeasures were deployed as deception in depth on a wireless honeypot. This included FakeAP and Honeyd as the primary honeypot. The attacks performed were wireless sniffing tools Kismet and Netstumbler on FakeAP, and NMAP and Nessus attacks on Honeyd. The deceptions deployed were mapped from a framework that was used to identify how, and what type of deceptions the countermeasures and the attacking tools may use.

The experimental approach and exploratory investigation allowed the researcher to test dependent variables - DV's (different types of deceptions), against an independent variable - IV (the wireless honeypot). The outcomes gave indications of cause and effect relationships between the IV and DV's. The researcher implemented a set of conditions to test the deceptions on the IV, tested the outcomes, and then reconfigured the IV, and tested again.

This was to ascertain experimental outcomes to answer the research questions:

1. Can a framework for deception be applied to network countermeasures to reduce the effectiveness of attacks?
2. How effective is deception in a wireless honeypot used against brute force attacks?
3. Under what conditions, and do conditions vary by the type of attack?

9.1.1 Evaluating the effectiveness of deceptions in a wireless honeypot

The researcher found from the results that deceptions used on FakeAP were highly effective against the wireless sniffing tools Kismet and Netstumbler. However, the deceptions utilised by Honeyd were not effective, in all instances, against the network attacking tools NMAP and Nessus.

It was determined from the results of all the attack tools used against the wireless honeypot, that FakeAP was able to deploy the most effective deceptions because FakeAP was designed for the wireless medium. The wireless sniffing tools Kismet and Netstumbler could be deceived because they too, were tools that utilise the wireless environment.

NMAP and Nessus however, have been primarily used in a wired environment utilising TCP/IP connectivity at the network level. Because the network level becomes blurred in a wireless environment, this may have been the reason why Honeyd could not effectively deceive the attack tools NMAP and Nessus. Network packets may have been lost, or disrupted packet sequencing may have caused Honeyd and the attacking tools to have errors in their network communication.

9.1.2 Evaluating the application of the framework for deception

The researcher found the application of the framework to be highly useful for mapping the network countermeasures and attacks to deceptions. It gave the researcher a greater understanding of how each deceptive effect of masking, mimicry and confusing may be implemented. Furthermore, different levels of deceptive sophistication, active and passive states, as well as a defensive or offensive stance were considered to heighten understanding of the deceptive effect.

The framework was therefore useful in determining which network countermeasures were more effective against the attacks. It was observed that depending on the type of deception, the countermeasure implemented, and the way it was deployed, affected the success of the deception on the attack.

9.1.3 Future research and investigation of deceptions

Deceptions may be investigated further by using FakeAP and Honeyd to deploy deceptions identified in the framework that are more sophisticated. FakeAP may incorporate wireless communication between more than one AP. This would be a dynamic and mimicked deception for wireless sniffing tools and packet sniffers to intercept.

Additionally, the deceptions deployed by Honeyd may incorporate more network and application level interaction, also through a dynamic and mimicked deception. This may be done by adding open ports, and running services to the Honeyd configuration file. Each service may run a script that produces a login-banner to inform the attacker of what service and OS is running. Additionally, a PERL script may be called for other front-end applications, such as a web page. This may give additional richness to the deception to a would-be attacker as a secondary step after using NMAP and Nessus.

The investigation may still be exploratory, although a field type study would be adopted instead. This would allow the researcher to deploy the wireless honeypot in an 'live' and 'open' space (not in a laboratory), to observe the attacks perpetrated in an uncontrolled environment. The results may give insight to how the effectiveness of similar deceptions deployed in this experiment, may result in a live and open environment.

9.1.4 Limitations in the research

There were several limitations within this research. The researcher had intended to conduct a third round of attacks. These would have incorporated single-hit buffer overflow attacks on the honeypot. This would yield useful results pertaining to the effectiveness of deceptions against stealth attacks on the wireless honeypot. However, the time constraint limited the researcher's ability to conduct testing beyond Round two.

Additionally the researcher found that the Honeyd log files were lacking in providing useful network information on extended network activity. Honeyd reported when packets were sent to a port, however, there was no additional information of the protocol activity and exchange that occurred between the attacking machine and the target wireless honeypot.

However, it was the researcher's main objective to measure the effectiveness of deceptions used according to the results found through NMAP and Nessus scans. The Central Logging Structure (CLS), which was the SNORT IDS and Honeyd logs, were intended as a supplementary form of data collection to triangulate and verify the results of NMAP and Nessus. It was anticipated that the CLS would provide additional findings at the protocol level. This may have identified the honeypot's interpretation of what was happening from the victim's perspective. Nonetheless, the scope of the research was to determine the level of deception achieved on tools that would typically be used by a script kiddie.

Further research may use exploratory investigation to view the results from the victim's point of view. This may determine the victim's perspective of the level of deception achieved, which may then be compared to the attacker's perspective.

9.1.5 Assessing the effectiveness of deceptions used for network defence

The observed levels of deceptions achieved were significant for understanding how wireless networks may improve security. The research honeypot gave insight into the way deception may be effective or ineffective in a wireless environment against common network attacks. FakeAP demonstrated itself to be a highly effective deceptive tool for countering wireless sniffing tools. The Honeyd deceptions demonstrated inconsistent results that indicated that network-attacking tools may not be effectively deceived when used in a wireless environment.

This identified to the researcher the difficulties that may arise when deploying network-based deceptions in a wireless environment. Based on the findings of this experiment, it is the researcher's deduction that the effectiveness of deceptions used in wireless network defence will need further investigation.

10 REFERENCES

Aleph One. (n.d.). *Smashing The Stack For Fun And Profit.* Retrieved 16 Sept, 2003, from http://destroy.net/machines/security/P49-14-Aleph-One

Barnes, C., Bautts, T., Lloyd, D., Ouellet, E., Posluns, J., & Zendzian, D. (2002). *Hack proofing your wireless network.* USA: Syngress Publishing Inc.

Beck, R. (2001). Passive-Aggressive Resistance: OS Fingerprint Evasion. *Linux Journal, n.a.*(89).

Bowyer, J. (1982). *Cheating: deception in war & magic, games & sports, sex & religion, business & con games, politics & espionage, art & science.* New York: St. Martin's Press.

Caras, R. (1972). *Protective coloration and mimicry: nature's camouflage.* New York: Westover Publishing Co.

Carthy, J. (1972). *Animal camouflage.* Great Britain: William Clowes & Sons Ltd.

Caswell, B., & Roesch, M. (2002). Snort (Version 2.0.2) [Wireless packet sniffer] [computer software].

Chamales, G., & Klingner, B. (2003). *Introduction to Network Security.* Retrieved 20 Oct, 2003, from http://ieee.ece.utexas.edu/comm/presentations/Spring03/netsec2-1.pdf

Chesnevar, C. I., Maguitman, A. G., & Loui, R. P. (2000). Logical models of argument. *ACM Computing Surveys., 32*(4), 337 - 387.

Cheswick, B. (1992). *An evening with Berferd: In which a cracker is lured, endured, and studied.* Paper presented at the Winter USENIX Conference, San Francisco.

Clarke, R. (2001). *Knowledge.* Retrieved 28 July, 2003, from http://www.anu.edu.au/people/Roger.Clarke/SOS/Know.html

Cohen, F. (1999). *The Deception Toolkit home page and mailing list: Introduction and basic idea.* Retrieved 19 March, 2003, from http://www.all.net/

College of Aerospace Doctrine Research and Education. (1997). *Air and space power mentoring guide (essays - volume 1): Principles of war.* Retrieved 1 March, 2003, from http://www.cadre.maxwell.af.mil/ar/MENTOR/vol1/SEC06.PDF,

Combs, G. (2003). Ethereal (Version 0.9.15).

Conry-Murray, A. (2001). Network security's not-so-secret ingredients. *Network Magazine, 16.*

Cowan, C., Wagle, P., & Pu, C. (1999). *Buffer Overflows: Attacks and Defences for the Vulnerability of the Decade*. Retrieved Sept 16, 2003, from http://www.immunix.org/StackGuard/discex00.pdf

Crotty, M. (1998). *The foundation of social research.* Australia: Allen & Unwin Pty Ltd.

Davis, J. (1997a). *Experimental research methods.* Retrieved May 12, 2003, from http://www.naropa.edu/faculty/johndavis/prm2/exper1.html#model

Davis, J. (1997b). *Experimental research methods part two: types of variables and validity.* Retrieved May 12, 2003, from http://www.naropa.edu/faculty/johndavis/prm2/exper2.html

Dawson, A. (2003) [personal communication].

Deraison, R. (2003a). Nessus (Version 2.0) [Network vulnerability scanner] [computer software].

Deraison, R. (2003b). *Nessus introduction.* Retrieved 12 Oct, 2003, from http://www.nessus.org/intro.html

Dolhenty, J. (2003). *The problem of knowledge: A brief introduction to epistemology.* Retrieved 28 July, 2003, from http://radicalacademy.com/epistom.htm

Dulany, K. (2002). *The wireless and mobile market starts to mature.* Retrieved 20 March, 2003, from http://www4.gartner.com/pages/story.php.id.3056.s.8.jsp

Elby, A., & Lising, L. (n.d.). *The importance of epistemological considerations in fostering conceptual development.* Retrieved 28 July, 2003, from http://www.physics.umd.edu/rgroups/ripe/perg/talks/RochesterAAPT/ElbyPERC.pdf

Fennelly, C. (2001). *Security in wireless: Let security hound you.* Retrieved 15 Oct, 2003, from http://www-106.ibm.com/developerworks/security/library/wi-sec.html?dwzone=security

Fluher, S., Mantin, I., & Shamir, A. (2001). *Penetration testing: weaknesses in the key scheduling algorithm of RC4.* Retrieved 31 March, 2003, from http://lists.insecure.org/lists/pen-test/2001/Aug/0012.html

Fratto, M. (2003). *NIP attacks in the bud.* Retrieved 20 Oct, 2003, from http://www.networkcomputing.com/1417/1417f26.html

Fyodor. (2001). Nmap exploration tool and security scanner (Version 3.48-1).

Fyodor. (2003a). *Nmap Remote OS Detection.* Retrieved 4 Sept, 2003, from http://www.insecure.org/nmap/nmap-fingerprinting-article.html

Fyodor. (2003b). Nmap: Network Mapper (Version 3.48) [Network exploration tool and security scanner] [computer software].

Gerwehr, S., & Anderson, R. (2000). *Employing deception in INFOSEC.* Retrieved 18 Feb, 2003, from http://www.cert.org/research/isw/isw2000/papers/26.pdf

Gerwehr, S., & Glenn, R. (2000). *The art of darkness: deception and urban operations.* Retrieved 20 Feb, 2003, from http://www.rand.org/publications/MR/MR1132/MR1132.chap3.pdf

Gerwehr, S., & Glenn, W. (2003). *Unweaving the web: deception and adaptation in future operations.* Santa Monica: RAND.

Grover, S. (2003). *Buffer Overflow Attacks and Their Countermeasures.* Retrieved Sept 16, 2003, from http://www.home.linuxjournal.com/article.php?sid=6701

Gupta, N. (2002, 28 & 29 November). *Improving the effectiveness of deceptive honeynets through an empirical learning approach.* Paper presented at the 3rd Australian Information Warfare & Security Conference, Perth, Western Australia.

Hartley, B. (2003). *Ethical Hacking: The value of controlled penetration testing.* Retrieved 20 Oct, 2003, from www.certconf.org/presentations/2003/Wed/WM4.pdf

Hegerle, B., & Bruestle, J. (2002). AirSnort (Version 0.2.1b) [computer software].

Honeynet Project. (2000). *Know Your Enemy: The tools and methodologies of the script kiddie.* Retrieved 15 Oct, 2003, from http://project.honeynet.org/papers/enemy/

Honeypots Net. (2003). *Intrusion Detection Systems*

Honeypots & Incident Response. Retrieved 17 Oct, 2003, from http://www.honeypots.net/

Hopkins, W. (2000). *Quantitative research design.* Retrieved 20 Oct, 2003, from http://www.sportsci.org/jour/0001/wghdesign.html

Huck, S., & Cormier, W. (1996). *Reading statistics and research.* New York: HarperCollins Publishers Inc.

Hutchins, E. (1980). *Nature invented it first.* New York: Dodd, Mead & Company.

Hutchinson, W., & Warren, M. (2002). *Deception in cyberspace*. Retrieved 12 March, 2003, from http://www.dlux.org.au/dataterra/deception_in_cyberspace.html

Insecure.org. (2003). *Nmap remote OS detection*. Retrieved 20 Oct, 2003, from http://www.insecure.org/nmap/nmap-fingerprinting-article.html

Ivyhall. (n.d.). *Viceroy*. Retrieved 15 March, 2003, from http://www.ivyhall.district96.k12.il.us/4th/kkhp/1insects/viceroy.html

Kershaw, M. (2003). Kismet (Version 3.0.1).

Lemos, R. (2002). *Catching wireless hackers in the act*. Retrieved 20 March, 2003, from http://asia.cnet.com/sg/0,39002190,39078380,00.htm

Liu, J. (2003). *How to build a secure WLAN*. Retrieved 10 March, 2003, from http://www.computerworld.com/mobiletopics/mobile/story/0,10801,78275,00.html?SKC=mobile-78275

McClure, S., Scambray, J., & Kurtz, G. (2001). *Hacking exposed: network security secrets and solutions* (3rd ed.). California: McGraw Hill.

Milner, M. (2002). Netstumbler (Version 0.3.30) [computer software].

Montcalm, E. (2002). *How to avoid ethical and legal issues in wireless network discovery*. Retrieved 10 Feb, 2003, from http://www.sans.org/rr/wireless/ethical.php

Nanda, S. (2002). *Wireless insecurity / how Johnny can hack your WEP protected 802.11b network!* Retrieved 15 Feb, 2003, from http://www.cs.dartmouth.edu/~snanda/presentations/Wireless_Soumendra_CS188.pdf

Noordergraaf, A. (2002). *How Hackers Do It: Tricks, Tools, and Techniques*. Retrieved 20 Oct, 2003, from http://www.sun.com/solutions/blueprints/0502/816-4816-10.pdf

PBS. (n.d.). *The living Edens Madagascar a world apart: a truly bizarre lizard*. Retrieved 28 March, 2003, from http://www.pbs.org/edens/madagascar/creature3.htm

Pfleeger, C., & Pfleeger, S. (2003). *Security in computing* (3rd ed.). New Jersey: Pearson Education Inc.

Philosophical Society. (n.d.). *Epistemology*. Retrieved 28 July, 2003, from http://www.philosophicalsociety.com/epistemology.htm

Polar Bears International. (2002). *Polar bear fur*. Retrieved 12 March, 2003, from http://www.polarbearsalive.org/facts3.htm#anchor768453

Poulsen, K. (2002). *Wi-Fi honeypots a new hacker trap*. Retrieved 20 March, 2003, from http://www.securityfocus.com/news/552

Provos, N. (2003). *Honeyd: a virtual honeypot daemon (extended abstract)*. Retrieved 28 March, 2003, from http://www.citi.umich.edu/u/provos/papers/honeyd-eabstract.pdf

Ptacek, H., & Newsham, T. (1998). *Insertion, evasion, and denial of service: eluding network intrusion detection*. Retrieved 15 Feb, 2003, from http://www.packetfactory.net/papers/NIDS-evasion/

RAND. (2001). *Lessons from animal and plant deception*. Retrieved 25 Feb, 2003, from http://www.rand.org/natsec_area/products/animal.html

RAND. (2003). *About RAND*. Retrieved 25 Feb, 2003, from http://www.rand.org/about/

Rue, L. (1994). *By the grace of guile: the role of deception in natural history and human affairs*. New York: Oxford University Press.

Sarantakos, S. (1998). *Social research*. Melbourne: Macmillan Education Australia PTY LTD.

Schneier, B. (2000). *Secrets and lies digital security in a networked world*. New York: John Wiley & Sons.

Schoeneck, R. (2003). *Wireless Honeypot*. Retrieved 18 Oct, 2003, from http://www.giac.org/practical/GSEC/Richard_Schoeneck_GSEC.pdf

Search Security. (2003). *Script kiddy*. Retrieved 15 Oct, 2003, from http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci550928,00.html

Spitzner, L. (2002a). *Honeypots: definitions and value of honeypots*. Retrieved 14 April, 2002, from http://www.enteract.com/~lspitz/honeypot.html

Spitzner, L. (2002b). *Know your enemy*. Indianapolis: Addison-Wesley.

Spitzner, L. (2003). *Honeypots - tracking hackers*. Boston: Pearson Education Inc.

Spitzner, L., & Roesch, M. (2001). *The Value of Honeypots, Part One: Definitions and Values of Honeypots*. Retrieved 17 Oct, 2003, from http://www.securityfocus.com/infocus/1492

Thomsen, C. (n.d.). *General research paradigm*. Retrieved 29 July, 2003, from http://faculty.njcu.edu/cthomsen/rm-grp-basic.html

Tourrilhes, J. (1996). Linux Programmer's Manual: iwconfig - configure a wireless network interface.

Trilling, S. (2003). *How to tighten loose security in wireless networks.* Retrieved 19 March, 2003, from http://www.computerworld.com/securitytopics/security/story/0,10801,78476,00.html?SKC=security-78476

Trochim, W. (2002). *Positivism & post-positivism.* Retrieved 13 May, 2003, from http://trochim.human.cornell.edu/kb/positvsm.htm

Valli, C. (2003a) [personal communication].

Valli, C. (2003b). *Honeyd - a fingerprint artifice.* Paper to be presented at the 1st Australian Computer, Information and Network Forensics Conference, Scarborough, Western Australia.

Webb, S. (2002, 28 & 29 November). *Wireless insecurity - current issues with securing WLANs utilising 802.11b technology.* Paper presented at the 3rd Australian Information Warfare & Security Conference, Perth, Western Australia.

Willemsen, E. W. (1974). *Understanding statistical reasoning.* San Francisco: W. H. Freeman and Company.

Wright, J. (2003). *Detecting wireless LAN MAC address spoofing.* Retrieved 23 Feb, 2003, from http://home.jwu.edu/jwright/papers/wlan-mac-spoof.pdf

Yarochkin, F., & Arkin, O. (2003). Xprobe (Version 2-0.2) [computer software].

Yek, S., & Valli, C. (2002, 28 & 29 November). *If you go down to the Internet today - deceptive honeypots.* Paper presented at the 3rd Australian Information Warfare & Security Conference, Perth, Western Australia.