UNIVERSITY OF OSLO
Department of Informatics

Computer Deception

Back to Basics

Aasmund Thuv
(akthuv@ifi.uio.no)

23rd May 2005

Abstract

In today's modern society, the increasing demands for connectivity and accessibility place computers in ever larger internetworks. As more and more computers become globally accessible, the number of threats from random and targeted attacks rise rapidly. To counter known and unknown threats, various technologies and concepts are employed as defensive measures. One concept that is in rising popularity is computer deception, the subject of this thesis.

The field of computer deception is characterized by fragmentation and is lacking unified definitions and methods. This thesis has reviewed five deception paradigms, in order to build a descriptive theory that is used for understanding the concept of computer deception. The border between human deception and computer deception is investigated.

The thesis concludes that computer deception for defense rarely can be seen as a field unrelated to human deception. When attacker tools are targeted for deception, they are only intermediary steps on the way to a human attacker. This makes the core issues of computer deception a matter of psychology, not technology. Computer specialists without knowledge of psychology do not have the expertise necessary for estimating the consequences of deceptions on human attackers.

Preface

Dear reader,

you are currently holding a copy of a master thesis in your hands. It has been written by Yours truly in the period of August 2003 to May 2005, the end of a five year study at the Department of Informatics, University of Oslo. The work has been carried out at the Norwegian Defence Research Establishment (FFI), Kjeller.

In the summer of 2003, the author and several researchers at FFI were discussing possible subjects for a master thesis. We decided on a theoretical and practical thesis in the subject of computer deception, the latter part consisting of a prototype emulating live systems by generating false network traffic. The former was intended to give a solid theoretical foundation for the prototype. Of course, we ended up somewhere completely different, and haven't implemented a thing. Go figure.

I would like to thank all of the researchers at FFI who have shown an interest in this thesis. Truly, the support have been above and beyond the call of duty. They trusted me enough to let me be in my own little world, reading and thinking for a year, before politely reminding me of the fact that it would be nice to have some sort of written result. When I desperately needed their input, they put aside much of their own work to help a lost student.

My thesis advisors, Lasse Øverlier at FFI and Pål Spilling at the University Graduate Center (UniK) deserves my gratitude. Lasse supported me throughout, while Pål asked pertinent questions of my objectives and desired results. Ane Daae Weng and Ronny Windvik, FFI, also read drafts and helped me sorting out my own thoughts.

A heartfelt thanks go out to Kjetil Mosesen and Kjell Olav Nystuen, both at FFI. Without their input the result would have been far less readable and coherent, besides lacking in content. Sometimes I wonder how they got anything else done, besides aiding me.

My fellow student and co-worker Torgeir Broen was always ready to discuss related and unrelated matters. Without a partner in crime, the last two years would have been far more boring. Finally, thanks to the room next door for the support in hacking LaTeX.

Aasmund Thuv
Kjeller, 23rd May 2005

Contents

List of Figures

List of Tables

Part I

Background

Chapter 1

Introduction

1.1 The Threat

> Note: Given the widespread use of automated attack tools, attacks against Internet-connected systems have become so commonplace that counts of the number of incidents reported provide little information with regard to assessing the scope and impact of attacks. Therefore, as of 2004, we will no longer publish the number of incidents reported.

(CERT Coordination Center 2005)

The proper way to start any paper or thesis dealing with security in any form is, of course, to stress the dangerous and hostile environment you are in. Preferably, you should not know about being on shaky ground beforehand: When the rug is pulled out from beneath your feet, the fear and paranoia - becoming so much greater from the unexpectedness of the attack - will lead you to grasp any solution being offered to you. Fear is a great motivator.

There is a slight problem with applying this method in our case. First of all, the shaky ground has been identified and a not insubstantial industry has grown up as a result. Secondly, there is a lack of grandeur inherent in our subject: We are not talking about black-ops, gun-toting ATF-agents shouting "go-go-go!" or even a stakeout with plain dressed cops, decaf or doughnuts. Our weapons are algorithms, code and packets, our agents usually undernourished youths without enough exercise (playing up to the stereotype here, sorry) and our subject, without fanfare, is computer network defense.

While there might be little glory in computer network defense, there is at least a real, acknowledged threat. The mere fact of a computer being connected to the Internet will very likely result in an unprovoked attack. Worms, automated scanners and attackers looking for resources rather than content do not care about the computer's reason for being present. All computers are "targets of opportunity" (Spitzner 2002), whether it is a fancy new server or an old home machine used for reading mail. If you are globally accessible, you are a potential target. Combined with the proliferation of automated tools over the Internet itself, any bored kid with point-and-click skills can also become an attacker. This is a volatile situation: Millions of potential targets, millions of bored kids, and an environment where the reach is global and anonymity the norm.

In addition to this, you have the advanced attackers to consider too. Less a threat for home users than companies and organizations holding some sort of interesting content, these attackers know what they want and where to get it. Skilled and sophisticated, more likely to be covering their tracks, not much is known about them. The computer arena is probably just another playground for the old goals, whether it is corporate espionage or other means for making money.

In addition to these two groups of attackers, who focus on "targets of opportunity" and "targets of choice", respectively (Spitzner 2002), there is a third category of attackers, namely the cyberterrorist. Both the definition of what constitutes a cyberterrorist, as well as the threat he represents is contested. The "cryptographeress"[1] Dorothy Denning defined and described cyberterrorism in a testimony to the Special Oversight Panel on Terrorism (Denning 2000):

> Cyberterrorism is the convergence of terrorism and cyberspace. It is generally understood to mean unlawful attacks and threats of attack against computers, networks, and the information stored therein when done to intimidate or coerce a government or its people in furtherance of political or social objectives. Further, to qualify as cyberterrorism, an attack should result in violence against persons or property, or at least cause enough harm to generate fear. Attacks that lead to death or bodily injury, explosions, plane crashes, water contamination, or severe economic loss would be examples. Serious attacks against critical infrastructures could be acts of cyberterrorism, depending on their impact. Attacks that disrupt nonessential services or that are mainly a costly nuisance would not.

While there is agreement on the fact that a cyberterrorist threat exists and targets are plenty, there are widespread differences in the belief of the severity and the extent of the threat. Five point of views can be summarized (Tan

[1] Bruce Sterling: The Hacker Crackdown.

2003): (1) The highly concerned who believes it is a matter of time (2) those who think it is above the technical capacity or desire of the traditional terrorist (3) those who see it as a new realm for the old terrorism and support normal computer security measures (4) a "cry-wolf" camp who only sees hype but no cyberterrorist acts (5) the "realist" camp who believes the cyberterrorist is in reality normal criminals committing cybercrimes.

It certainly looks like there are enough attackers to choose from, with a wide variety of reasons for attacking *you*. While it might be too dramatic to call it a state of war, it seems reasonable to assert that we are in a state of conflict with unknown assailants, with a multitude of motives. This is not a comfortable situation, and as a result security mechanisms and products have been developed to counter the threat. We have firewalls to block access, encryption to obfuscate content, integrity verification to detect unauthorized changes and intrusion detection systems both for the network and the host. This perpetual conflict drives the development of attacker and defender techniques and mechanisms, usually with the attacker initiating and the defender responding. Every tool that can aid us as defenders is desirable. Every technique which can bring us ahead should be investigated. In a roundabout way, we have finally arrived at the subject of this thesis: The use of deception for computer network defense.

1.2 The Field of Computer Deception

For most computer scientists, computer security is probably not the first that comes to mind when thinking about deception. The historically inclined will perhaps remember the wooden horse of Troy, Jacob being clothed in the raiments of his brother Esau or Machiavelli's words "you must be a great liar and hypocrite"[2]. There are plenty of examples closer to our own time, in politics, in diplomacy, or military actions. In fact, deception crops up in most professions in one form or another: "Diplomats, counter-espionage officers, politicians, businessmen, con artists, charlatans, hoaxers, practical jokers, poker players, gambling cheats, football quarterbacks, fencers, actors, artists, mystery story writers, or you or I in our everyday lives."(Whaley 1982). Such pervasiveness would suggest that deception should also have a place in the computer system.

It is therefore not a big surprise when we find, after some investigation, that it does. Several efforts are underway, and have been underway for some time, to use deception in the interest of computer defense.

However, if you try to gain a comprehensive overview of the field of computer deception, you will find that it is fragmented, lacking a commonly

[2] The Prince, chapter 18: "The way princes should keep their word".

accepted framework, methodology, or definition. In fact, this merely reflects the field of deception. Instead of a grand theory of deception, there are many small ones, tailored to specific situations or specialized fields. To maneuver in the field of computer deception, and to understand what is relevant or irrelevant to your own situation, is not an easy task.

Chapter 2

Thesis Overview

2.1 Motivation

Originally, our intention was to utilize computer deception actively by building a defensive capability for computers and computer networks. To do so, we needed a place to start, some sort of computer deception paradigm that could give us the foundation, the tools, and means such a project demands.

When we asked the question "what is computer deception", we got a lot of different answers. As diversity fosters development, this could be considered a good thing. From a practical standpoint, it was confusing. Faced with such a problem, one solution is to merely choose between the opposing views, either that which you find appearing more true, or that more usable. However, in this case the views appeared so different, that we found it hard to justify choosing one over the other without a more in-depth investigation. A defensive capability should not be built on a randomly chosen paradigm, nor without understanding the consequences of choosing.

2.2 Thesis Objectives

Our main objective was building a descriptive model or theory that we could use to understand the concept of computer deception. It became clear that the border between deception and computer deception was somewhat arbitrary, and we have tried to use our theory in finding a more precise placement of this border. This invariably means taking a stand on the definition of computer deception and its objectives, which we have done.

It is our intention that the theory can be used during deception planning to make clear what issues are at stake, as well as a tool for describing existing deployments of computer deceptions. As the theory is descriptive rather than normative, the result is not a set of techniques, nor guidelines for designing techniques, but a theory one can use to describe techniques.

A theory at this fundamental level must remain generic to be useful, and we have tried to avoid making assumptions of attacker and defender methodologies.

2.3 Working Process

We have used a strictly theoretical approach, ie a textual study. At this stage we saw no point in implementing any sort of prototype, when we did not know what we could do, what we should do, or how to do it. Which we still don't, but at least now we know why.

2.4 Limitations and Boundaries

Our search has led us out of the world of computers and into other domains. Following the notion of it being a good idea to know what deception is before we try figuring out what computer deception is, we sought texts that could have use for us in our situation. In such an endeavor, there are far too many texts to read them all, or even a significant fraction. All loose ends cannot be tied up, all facets cannot be investigated. We have done simplifications, and ignored theory worthy of more attention. And that's in addition to the countless mistakes you will undoubtedly find.

We chose mainly to investigate five different approaches to deception in more detail. Three of those were directly connected to computer deception, and as such should not need any justification for their inclusion. Of those the first is the honeypot, which is probably the best known computer deception concept. The second is a deception categorization of Dunnigan and Nofi much used by computer specialists, while the third is Cohen's "Framework for Deception", a work that looks at computer deception from a cognitive point of view.

The other two approaches belong to psychology and warfare. The first is an attempt a general deception theory with psychology as the focal point, by Barton Whaley. In his view, deception is psychology irrespective of the field it is applied to. The second consists of two texts from deception in warfare: "The Art of Deception in Warfare" by Michael Dewar, and ""Joint Publication 3-58: Joint Doctrine for Military Deception." The basic idea is that the situation between a computer attacker and defender resembles that of a military conflict, and that it can be worth looking into military doctrine[1].

There are many approaches not covered by this, for instance the use of artificial intelligence or semantic cases. Adding them would have been highly interesting, but both time and resources were, unfortunately, finite.

2.5 Thesis Structure and Composition

This thesis is divided into three parts. The first part is an introductory background, containing a chapter covering some of the threats we face as computer network users, and briefly describing the field of computer deception. The thesis overview (which you are now reading) ends the part by setting the stage for a few people whom we shall follow throughout the thesis, in the interludes. Besides being a minor comic relief, the interludes will try to show some aspects of each of the five texts we have chosen for our study and analysis of deception.

The second part contains the five texts. Our regular approach will be to describe the theory, largely as the authors themselves have done, before we analyze, and if necessary, relate the theory to computer deception.

The order of the texts is perhaps not the most intuitive, but it is a compromise between several factors. It was possible to group them by the focus on computer deception (honeypots, Dunnigan and Nofi, Cohen), or by warfare (Dunnigan and Nofi, Dewar, JP 3-58). In the end, we decided to take the two probably best known first (honeypots, Dunnigan and Nofi), then sneaking in Whaley's theory, before Cohen. Cohen's theory can, in some ways, be seen to logically follow Whaley.

The space we have spent on each text varies heavily, and does not necessarily mirror the texts themselves. For instance, the shortest chapter is the one on Dunnigan and Nofi, who wrote a book on nearly 400 pages. The paper of Barton Whaley, covering 14 pages, is given nearly three times the space. The theory of Whaley, however, goes much deeper than that of Dunnigan and Nofi, and therefore needs more space.

The third part begins with a discussion where we use the material we have read to define deception. We put a proper definition of computer deception on hold, until we have discussed the most important aspects covered by the texts. To anticipate events, we will cover strategic objectives, chains of assumptions, use the deception techniques to derive a set of deception characteristics, state a set of deception principles, and see how military deception planning and execution can shed some light on computer decep-

[1] This is not a new idea: Dunnigan and Nofi's categorization was originally used for categorizing historical military deceptions.

tions. Then follows a chapter where we build a model describing how a computer attacker and defender interact in very generic terms, which we will use to look closer at the border between deception and computer deception. We will also use the theory in a set of scenarios, where we try to avoid psychological assumptions; not an easy task. Finally we have the thesis conclusion and further work.

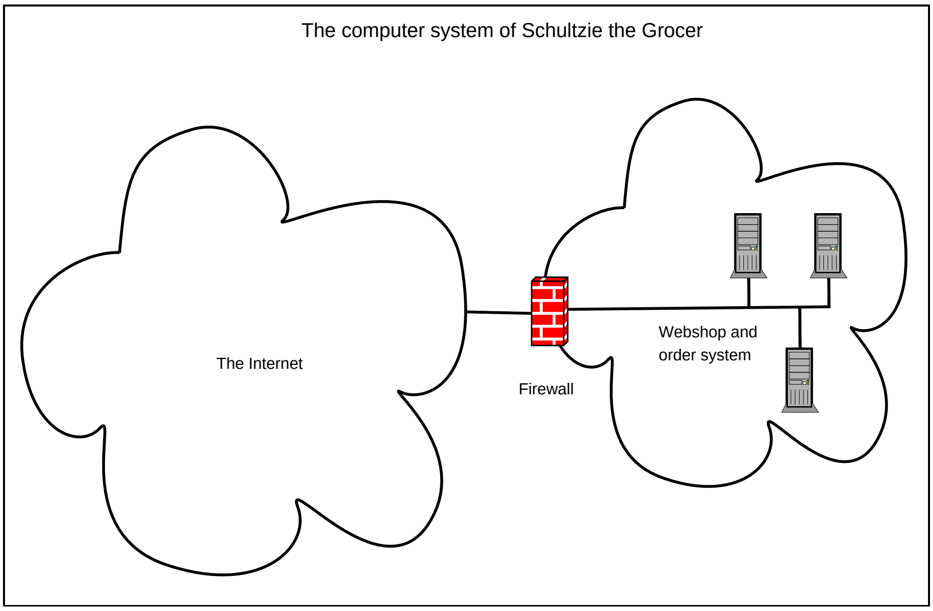

Figure 2.1: *Frank Schultz's computer system.*

2.6 An Interlude

First Act: Introduction of Our Heroes

Frank Schultz is a small-time grocer in a little known town in the country of Apathia. Having joined the Information Age Revolution, Frank has no public store, but runs a virtual grocer's shop on the Internet. He maintains a small stockroom in his cellar, and mainly serves a small group of local friends and acquaintances. Orders are received by the Internet or by phone, and deliveries are made thrice a day.

His computer system consists of a small network with three machines behind a firewall, directly connected to the Internet. Normally, the system runs itself without problems, but lately a worrying trend has become apparent. Every day, an increasing number of network probes are sent against the firewall. It is probably just a question of time before an attacker breaks through, and gains access to the whole system. As Frank depends on the system for his business, his anxiety increases proportionally with the attack frequency.

Luckily, amongst his friends there is a computer specialist who takes cares of his system, and whom he is just now awaiting. Having explained the situation and pleaded for help, the specialist, Bilibus (Bubba to friends) H. stands in the living room five minutes later.

"How are you, Frankie boy? I come prepared." Bubba takes off his backpack and

thrusts it towards Frank. "Catch!"

"Uh.. What's this?" Frank nearly topples forward due to the weight. "Why, the best defense: Offense! And preparation saves the cat." Frank, nonplussed, asks "Cat? What cat?"

"The one killed by curiosity, of course. Don't slow down on me now, Schultzie." Opening the backpack, Bubba takes out a thick pile of books and papers. "If we're going to do this, we must do it properly. You start on the top, and I'll start at the bottom."

"But Bubba, what is this?" Frank looks over the pile. "Honeypots: Tracking Hackers, Joint Doctrine for Military Deception, Toward a General Theory of Deception.. Bubba, explain? Please?" Bubba, exasperated, grabs Frank's shoulders and shakes him. "Frank! We talked about this on the phone, remember? The scans against your firewall? Yes?" Frank, slightly dazed, replies, "Yes?".

"And if we're going to take the firewall down, we must know what to expect, and how to counter it, yes? The possibilities, the means, the whole process."

"Take it down!?!? But that will let the attackers in!" Frank protests. "Of course they will get in - that's what we want! How can we deceive them if they are on the outside? Focus, Frankie, focus!"

"So you are saying we should let them in, in order to deceive them?" Frank slowly puts the pieces together. "Well, we cannot make it too easy, because then they'll know that we know they are attacking, and if they know that then they'll try to deceive us. Unless they don't know that we know that they know, but perhaps if we know that they know but they don't know that, that'll work too. Good thinking, Frank. Now you're on track."

"And we'll start by reading these papers on deception?"

"No time to loose. I'll get the pizza." Bubba moves towards the door. "Pizza? What pizza?" There is a knock at the door, and Bubba opens it to reveal a delivery boy. "Start reading, Frank. No food until I see some progress."

<eight hours, 2 pizzas and three beers later>

"I don't get this, Bubba. They're all doing different things, and sometimes they are not making any sense! It's just a big mess inside my head."

"Divide and conquer, Frank, divide and conquer. Rule number one when there's information overflow. Understand each of them by themselves, before you try to combine them. Look, we'll start at honeypots, move on to Dunnigan and Nofi, before we continue with..."

"My deliveries!" Frank jumps up from his chair. "I'm five hours late!"

"Frank! Where are your priorities! We have a mission here, you cannot leave now!" Bubba protests. "Send that son of yours, he's just hanging around the gas

station anyway."

"That's a time-honored way of spending time, Bubba. I'll see you in an hour."

"Fine, fine. Go then. I'll set up things here, and we'll check out honeypots when you're back."

Part II

Deception Paradigms

Chapter 3

Honeypots

A technology much hailed as active defense, honeypots are probably the best known attempt at deceptive measures in computer systems today. The traditional honeypot, used for detection, observation, or computer forensics, is described in "Honeypots: Tracking Hackers" by Lance Spitzner (Spitzner 2002), on which this chapter is based.

The chapter starts with a synopsis of our analysis, before presenting the theory and the analysis itself.

3.1 Paradigm Synopsis

In the honeypot paradigm, computer deception is mainly the masking of mechanisms and basic attempts at emulating services and the characteristics of hosts. Some measures are considered deceptive when specific service or protocol characteristics are exploited to achieve effects on a communicating partner.

Research honeypots are content with hiding the changes done to normal systems in order to learn what attackers do, while production honeypots per convention are used for detection and forensics. Newer honeypots go beyond this and target specific types of attackers.

By definition computer deception is not the same as deception, which is regarded as a psychological weapon used against humans, and is to be avoided. This view, however, is neither ubiquitous nor easy to defend when considering the aspirations of the unconventional honeypots. Nor is forensics as the only response in concordance with the idea of more active defense.

Guidelines for deception within this paradigm is not provided, beyond the idea of making anything false look as real as the truth.

3.2 The Theory

The theory begins with a short look at the origins of honeypots, before continuing with the definition of honeypots and two classification schemes, one by intent and one by level of interaction. Then follows a section on Honeynets, before the security paradigm is described.

3.2.1 The Origins

If we want to point to the beginning of the field of computer deception, surely we would not be too far off the mark if we claimed that it began properly in the late 1980's.

In 1986, Clifford Stoll, an astronomer doubling as a system manager at Lawrence Berkeley Lab (LBL), discovered a small 75-cent error in the accounting system (Stoll 1989). Not content with letting the error slide, Stoll started to examine the accounting programs written in a hodge-podge of Assembler, Cobol, and Fortran. The conclusion was that the error was not in the accounting system, but introduced by the presence of an unauthorized user. The accounting error soon showed itself to be the top of the iceberg, for the hacker used the LBL-machines as a platform for attacks into other US systems, including sensitive military systems. In the beginning Stoll recorded and logged the hacker's actions, while notifying administrators at the target systems. As the time went on, Stoll tried tracing the hacker while creating false content for holding the hacker's attention. In the end the FBI, NSA, and other alphabet groups were involved in an international chase. Stoll's novel reads like real cloak-and-dagger fiction, even having a quote from Tom Clancy ("A spy story for the '90s - and it's all true") on the back.

A few years later, in 1991, Bill Cheswick of AT&T Bell Laboratories modified their secure Internet gateway by adding false services and log scanning scripts (Cheswick 1992). When a hacker later attacked, Cheswick let the hacker believe he had gotten access, when the reality was that Cheswick emulated the compromised machine by hand. Later, Cheswick built a Jail, a limited controlled zone designed to contain the hacker without him noticing.

3.2.2 Definition

The closest descendant in the spirit of Stoll and Cheswick today, as well as probably the best known deceptive tool, would be the honeypot. Defined as "an information system resource whose value lies in unauthorized or illicit use of that resource" (Spitzner 2003)[1], honeypots can in principle be anything from a physical black box to a script. Most honeypots today either take the form of a program or a set of scripts running on a computer, or a larger framework where the computer itself, the operating system and other software combined constitutes the honeypot.

Considering the wide definition, there should be no surprise that there are multiple types of honeypots for a variety of uses. The definition, taken from Spitzner, is in fact given quite recently if you consider the time elapsed from Stoll and Cheswick's work in the early 1990's. As such, it is descriptive, seeking to encompass the entire field rather than being normative.

In his description of honeypot characteristics, Spitzner lists a set of advantages and disadvantages. These properties are largely defined, not derived, and embodies Spitzners vision of honeypots.

Advantages:

Data Value A honeypot is not directly involved in any production process, and should therefore not be contacted by any users. This decreases the chance of false positives (detected attacks that turn out not to be attacks), false negatives (undetected attacks), and lessens the noise of legitimate data. The traffic that is directed towards a honeypot is per definition suspicious.

Low probability of resource exhaustion Mechanisms like network intrusion detection systems should ideally monitor the entire flow of data in order to detect attacks. Due to limitations in hardware or software, this might be impossible to accomplish as the load increases, at least under normal budget constraints. The result is often that packets are dropped. Honeypots should not have this problem, as we already established that all traffic is suspicious.

Simplicity Again, the fact that all traffic is suspicious work to our advantage. The detection mechanism is in effect the existence of incoming traffic, in contrast to advanced algorithms, signature databases etc.

Return on investment With some technologies, it might be difficult to know if they are countering threats or are merely a drain of resources. Is encryption hindering attackers in reading our data, does the firewall

[1] Defined in (Spitzner 2002) as a "*security* resource [...]" (author's emphasis) but later refined in (Spitzner 2003) as given here.

keep attackers out? Honeypots are working as they should when you see that they are attacked, justifying their continuous employment.

Disadvantages:

Narrow field of view A honeypot is merely another unit or service on the network. An attacker targeting another unit or service will remain undetected. This means that honeypots cannot replace your existing security mechanisms.

Fingerprinting A honeypot pretending to be another service or unit might not succeed in portraying all the right characteristics, leading to its possible detection. What happens if a honeypot is recognized? Any detection mechanism known to the attackers might be used in the attackers favor as red herrings or otherwise.

Adding risk to the environment Depending on the complexity of the honeypot, additional risk might be introduced to the environment. A simple script honeypot is not likely to be any problem (unless exploitation of buffer overflows and the like is possible), while honeypots which in effect are complete machines with fully working operating systems are more vulnerable. If such a honeypot is taken over completely, it can be used for further attacks, not only against your own system, but against other third party bystanders.

3.2.3 Classification by Intent

Since the form of honeypots is arbitrary, a classification scheme exists which separates honeypots into two categories, depending on the reason for employing them. A *production* honeypot is used as a security mechanism, while *research* honeypots are used to study and gain information about the attackers. This categorization does not take the form of the implementation into consideration, which means that the implementation may actually be identical in the two cases. In reality different characteristics have shown themselves to be preferable for each type.

A production honeypot is used to increase the level of security. In this context we consider as security everything that lowers risk in a organization and its informational resources.

The typical use for a production honeypot is detection of attacks. One hopes that when real production machines are attacked, the honeypot will be included among those targeted. The honeypot should not, in normal circumstances, be in communication with anyone, and the presence of traffic will be an indication of attack.

Research honeypots have, in contrast with production honeypots, no direct security-increasing purpose. The purpose is to gather information about the attackers, their objectives, methods, and techniques. The reason for doing this may be a wish to learn about the attackers, in an anthropological spirit or to develop better security measures in the long run. New and unknown attacks may be identified and countered, and statistical modeling may be applied to make prediction about future attack patterns.

3.2.4 Classification by Level of Interaction

The definitions of production and research honeypots only consider the intent behind employing a honeypot. Another way of categorizing is by looking at the technology used when implementing the honeypot. From this point of view Spitzner describes three levels of interaction.

The low-level honeypot is usually a simple emulation script responsible for creating false services or units. This can be such a simple thing as opening a few ports like FTP or telnet, and recording the IPs of everyone connecting. An aspiring attacker would be able to connect, but nothing more; the honeypot would be able to detect the (assumed) attack but not be able to say anything about the intent of the initiator. By increasing the complexity of the emulation, one may try to portray a known service or unit type with a known vulnerability. An exploit directed at the vulnerability is definitely an attack attempt.

The interaction in these cases is between the attacker and the honeypot software. The attacker is never in contact with external services, software modules or the operating system unless other ports, outside the control of the honeypot software, is open. Often such solutions will be script based, piping all data from a port like FTP to an FTP-emulation script. You may then code the complexity you want yourself.

Raising the bar, medium-level honeypots offer more interaction possibilities for attackers. In one way, they can be regarded as complex low-level or constrained high-level honeypots. Examples of medium-level honeypots would be the emulation of a web-server with known vulnerabilities in order to capture payloads, or a restricting jail or chroot environment. Cheswick's controlled zone would be a real life example of this concept.

A high-level honeypot no longer uses emulated services. Now the attacker has access to real services on a real host. If the attacker connects to an insecure implementation of **sshd** and accomplishes a successful attack, he will get control over the honeypot. This being the intention, information is clandestinely gathered to know what is happening. The framework which is in place to do this is part of the honeypot.

Gathering information is not so easily done as you would expect, since an attacker often tries to hide his tracks. Usually you would log on different layers, from packets on the network to events on the computer. Catching the attacker's keystrokes and any uploaded software would be ideal, something which is possible by making changes to the operating system kernel. The danger of doing this is that the attacker might detect the changes.

If an attack at a high-level honeypot is successful a new problem arises. In many cases the attacker has so much control that he may attack other computers with the honeypot as the attacking platform. Even if the ethical questions around this is ignored, there may be legal issues. To block the honeypot from carrying out further attacks is an important part of the framework. This can be done by blocking outgoing traffic from the honeypot in a router or firewall. A knowledgeable attacker, however, might anticipate this and perform tests to ensure that he is not restrained in such a manner.

3.2.5 The Honeynet

An actor heavily involved in the development and use of honeypots is the Honeynet Project (The Honeynet Project 1999). A non-profit research organization of security professionals, the Honeynet Project is dedicated "To learn the tools, tactics, and motives involved in computer and network attacks, and share the lessons learned."[2] Their stated goals are to raise awareness of the threats that exist, to teach and inform about those threats, and to provide tools and methods for helping others learn more on their own.

The Honeynet Project has defined standards and requirements for a network of research honeypots: The Honeynet, which is a complete network environment under strict control set up to be compromised (The Honeynet Project 2004). Three areas are essential to the implementation of a Honeynet:

Data Control The attacker must be unable to use the Honeynet for further attacks. The activity must be contained in the Honeynet, and both automatic response and manual interaction should be provided. There should be at least two layers of protection against failure, with fallback to a safe state. Additional demands regarding data control are the tracking of connection states, real time local and remote administration, and automated alerting when a honeypot is compromised.

Data Capture The actions of an attacker must be monitored and logged. The objective is to capture as much relevant data as possible without

[2] Honeynet Project motto.

the attacker noticing. It is preferable with mechanisms on multiple layers, as packet sniffing on the network layer and keystroke capture in the kernel, in case the attacker is using encrypted connections like **ssh**.

Data must not be stored locally, since this increases the chance of detection or data pollution.

Data Collection Large organizations might employ several Honeynets. In this case the information should be gathered from all Honeynets and stored at central location in a standardized format.

Attackers must be unable to detect the control and observation mechanisms. If it is impossible to create outbound connections or large amounts of unknown traffic is seen on the network, the attacker would probably become suspicious. The Honeynet Project has handled this by allowing a small number of outbound connections combined with a Network Intrusion Prevention System (NIPS). The NIPS system can block known attacks (The Honeynet Project 2003). To gather information a tool called Sebek has been developed, which is a kernel module that hides itself and the traffic it generates.

3.2.6 The Security Paradigm

When considering what a honeypot can accomplish in order to strengthen security, Spitzner relies on the categorization scheme used by Bruce Schneier in "Secrets and Lies" (Schneier 2000). This scheme places countermeasures into the classes of detection, prevention and response.

Prevention, says Spitzner, is to keep the attackers out[3]. Other measures are more useful than honeypots in this instance. Removing unnecessary and insecure services, patching services and proper authentication mechanisms are more important than production honeypots. A research honeypot might decrease the security level by introducing vulnerabilities.

Detection is the category of which production honeypots contributes the most. In a network with large amounts of traffic there are three common detection problems: False positives, false negatives, and data aggregation (the collection and analysis of data).

Honeypots handle all of these problems, largely because all traffic to (or, in some cases, from) a honeypot is suspicious. This decreases the number of

[3] Spitzner uses the metaphor of putting locks on the doors and a fence around the house. What the attackers should be kept out of in a computer system is less clear. In some cases they have acquired access to the internal network, but are blocked from attacking production systems or attaining status as authorized users.

false positives and negatives, and makes the size of the logs to be analyzed smaller.

When an attack has happened, a honeypot provides good information about the attack. In a normal production system users might have continued to use the system without knowing about the attack. The more the system is used, the more the level of noise is increased and it becomes harder to extract information. It is not trivial to separate actions done by authorized users from the actions done by attackers when the amount of data becomes large. On a honeypot there should, in principle, only be information about unauthorized actions, and as opposed to production systems, it can be taken down for a thorough analysis. A server that is heavily used and essential for an organization is unlikely to be taken down quickly enough to keep the data uncorrupted, if it can be taken down at all.

3.3 Analysis

We begin with the honeypot characteristics, the security paradigm given by Schneier and its later use by Spitzner. Then we will continue with a look at unconventional honeypots, the detection of honeypots, before we search for the deceptive elements.

3.3.1 The Honeypot Characteristics

Three advantages of honeypots, namely data value, low probability of resource exhaustion, and simplicity, rests on a single definition: No legal traffic should interact with the honeypot, consequently, the traffic that interacts is suspicious. This equates to the notion of "presence equals attack".

If we try to forget that this is a definition, let us consider if it always will be the case. Some cursory thinking reveals several sources of traffic that are not attacks. For instance, there are broadcasts intended for all hosts, protocols seeking out certain types of hosts, misconfigurations and spelling errors.

Think of the production honeypot principle applied to an emulated application server. Due to a mistake, a secretary connects to the honeypot and begins her normal workday activities. Five hours later, the computer security specialist knocks on her door, and informs her that all of her work is lost, since she is not connected to a real server. On the other hand, if the honeypot uses a real application server, her work will be retained, but on a unsafe machine open to other attackers.

Even if you discard the above examples as unlikely, this problem suddenly rises again if you think of the most conspicuous characteristic separating honeypots from real hosts. For all intents and purposes, it is completely dead, and any attacker above the average will find the lack of traffic to and from the honeypot as highly suspicious. This is acknowledged as a characteristic making honeypots unlikely to be used for more sophisticated deceptions (Rowe and Rothstein 2003).

This was in fact the problem we originally set out to tackle: The generation of false traffic to draw attackers in, by having honeypots look more real and interesting. Real systems have load and interaction.

But what happens if we begin generating traffic? Suddenly, the honeypot is expected to react to our false traffic, and that requires it to separate our traffic from that of an attacker. The "presence equals attack" rule is broken.

3.3.2 The Security Paradigm

The security paradigm, as used by Spitzner, is not quite equal to the way Schneier defined it. In order to understand this, we must go to the source (Schneier 2000).

To be effective, countermeasures must cover three areas: Protection, detection and reaction. These are in a way subsequent stages. We prefer that attackers do not enter our system (protection), if this fails we want to know that they are inside (detection), and in the end we must take some sort of action after detection (reaction). To illustrate this Schneier gives a scenario, in which a military office with classified documents are to suffer a break-in.

Protection is the creation of barriers. In this scenario, an example could be a locked safe. Detection is alarms that triggers when attacks are underway, for instance a mechanism connected to the safe's door. Reactions are actions done due to the detection of attacks, as when guards come running after being notified.

The combined value of security is not a simple case of adding the individual contributions together. It is the unity of countermeasures that matters. For instance, if the reaction mechanisms are missing, there is no point in detecting attacks.

Spitzner regards response as a reason to have honeypots on the network. With response he means computer forensics, analysis and reconstruction of the course of events after the attack is over. Response is something done when the attack is over and we want to know what happened.

Schneier, on the other hand, is concerned about the composition of security mechanisms, and emphasizes that all security systems must have protection, detection and reaction. The 'reaction' of Schneier is an action that is performed as a consequence of the detection. Without reaction, there is no point in security mechanisms - nothing is going to happen anyway.

A more proper way of interpreting Spitzners response in comparison with Schneier's reaction is by regarding information gathering as the reaction. When a honeypot is attacked, the attack in itself will lead to a change in the internal state of the honeypot, leading to the activation of logging mechanisms. This would be a form of reaction that is instigated after detection.

3.3.3 Unconventional Honeypots

Two examples of honeypots that break with the detection and forensics mindset are LaBrea (LaBrea 2005) and Bubblegum Proxypot (Bubblegum 2005).

LaBrea is a honeypot that performs tar-pitting, the intentional slowing down of communication. The initial TCP handshake is completed, but by tweaking TCP options the communication crawls to a halt. This is useful for making the attacker spend time and resources, which would have been used for scanning other systems.

Bubblegum Proxypot is an emulated open proxy, intended to stop and catch spammers. Every request except the dangerous ones, ie sending spam, is performed as usual. The average spammer that takes the responses from Bubblegum at face value will believe that he has successfully sent spam, while in reality the spam is blocked and the activity logged. At the Bubblegum website proof is given of the success of proxypots, citing cases where evidence from proxypots have been used to prosecute spammers.

3.3.4 Honeypot Detection

Honeypots based on the common paradigm are vulnerable of detection. For production honeypots, this is not so problematic since a detection mechanism works even if an attacker knows he has been detected. The exception is that a returning attacker might know what to avoid and remain undetected. Moving the honeypot around might solve this.

For research honeypots, the situation is different. It is of paramount importance that the attacker believes he is in a real production environment, and that he is unobserved. If the attacker becomes suspicious, he will very likely change his behavior, feeding the observers with false data.

All honeypots that use emulated services or hosts will have a set of characteristics that is incompatible with the original. This is unavoidable, unless you implement the emulation exactly as the original, which, of course, is pointless.

This is known to be a serious problem, perhaps *the* problem, with honeypots. Especially when considering honeypots that have specific objectives besides detection, like proxy emulation or tar-pitting: These are intentionally performing differently than the real services, and this is detectable by a knowledgeable attacker (Oudot and Holz 2004a). Restricted environments like jails and chroot are also detectable (Holz and Raynal 2005a).

Research honeypots do not use emulated services, but they are also altered compared to normal systems. The additional logging and action constraint mechanisms can be induced to have a visible impact, for instance by making the logging mechanisms work excessively (Oudot and Holz 2004b), or by performing actions that a defender would be likely to block (Oudot and Holz 2004a).

Common to all is a characteristic following from the definition, a lack of traffic and activity. This is easily detectable if an attacker can capture traffic. Script kiddies and others that are merely interested in stealing resources might not be bothered to do this, but a sophisticated attacker would surely like to know what is going on.

Perhaps what is important is not the theoretical possibility of an attacker detecting honeypots, but if he is likely to do so. If we want to capture the run-of-the-mill, perfect emulation of a real system might be unnecessary, while we must improve the quality if we want to fool advanced attackers. However, it is deemed likely that automated honeypot fingerprinting tools will be used more and more by novice attackers (Holz and Raynal 2005b).

3.3.5 Where Is The Deception

While honeypots might be the best known example of a deceptive tool for computer defense, they actually use very few deceptive techniques.

A research honeypot portrays mainly three deceptions: (1) This is a normal machine (2) there is no (additional) logging going on here (3) there are no constraints on the attacker's action above the normal. The principal deception is the first one, which is essential when the main overall objective is learning what attackers do at normal systems. Hiding the additional logging and action constraint mechanisms is necessary to support this deception.

A production honeypot portrays the same principal deception, in that of being a normal system. Additional deceptions, on the other hand, can be presented when we interact with the attacker. We can tell lies and obscure the truth by the means of the honeypot software. Per convention, it is used mostly for detection purposes and some information gathering, but this restriction is artificial. The limits of a production honeypot lies in the scripts which, in effect, *are* the honeypot. The use of honeypots for tar-pitting and fighting spam breaks with conventional honeypot use and demonstrates this fact.

Can we say anything about other deceptive techniques, methods, purposes? Spitzner discusses this when considering the use of honeypots for prevention. "The deception concept is to have attackers waste time and resources attacking honeypots, as opposed to attacking production systems". Lumping it together with deterrence, in which attackers are scared off, he considers both concepts as "psychological weapons used to mess with and confuse a human attacker." In his opinion, this will not work on the most common attack type, targets of opportunity. There is seldom any human attacker to confuse, since the attackers use scripts or automated tools. You cannot fool someone who not paying attention. And against worms and auto-rooters there are no humans involved whatsoever. In the end, spending resources on patching vulnerabilities and securing systems in the regular way is better.

An exception is made for targets of choice, of which little is known. At least, there we have a human attacker present that might be deceived. If the attacker is after specific information, false content could be provided, as Stoll did. Large companies with sensitive research, like governmental or military organizations, would profit from such deceptions.

3.4 An Interlude

Second Act: Honeypots

Returning from his deliveries, Frank pulls up in the driveway, exits the car, and enters his house. Bubba is sitting in the living-room, looking at a computer screen. "Ah, there you are. I have new intelligence" he says when Frank arrives. "I have analyzed the firewall logs, and there are several things going on. As far as I can tell, there's a worm that is incessantly trying to get in, while someone else is scanning and testing different exploits regularly. So far, nothing has gotten in since I patch the firewall routinely."

"Well, what do we do about it?" Frank asks with a concerned frown. "Set up a honeypot?"

"I can hear Spitzner talking" Bubba says approvingly. "Good show. What are our options then?" Frank thinks. "Improving security or learning about the attacker? Production versus research?"

"Correct. But we face an unknown threat, would it not be safer to know what we are dealing with? So we should in fact combine the two?" Bubba points out.

Frank hesitates. "But how do we figure out what the attacker want, without letting him do it?"

"Simple. We combine the two concepts by using production honeypots as a detection mechanism, and then shuffle the attacker into a Honeynet where we can see what he wants. I want the whole cow."

"How are you going to move an attacker into a Honeynet without the attacker noticing?"

"I've got the whole thing worked out. First, I've written a script that emulates closed TCP and UDP ports, which we're going to run on your real hosts. If they are scanned, well, presence equals attack, so the rerouting mechanism (I'm coming to that) will trigger. This is the beauty of having a strict security policy that frowns on scans."

Frank looks mystified. "I have a security policy?"

"You do now. I've also saturated you IP address space with silent honeypots doing the same thing."

"And the reroute thingy?"

Bubba holds up a black box and declares, "This is our savior. Bow before it."

"Uh..."

"When my script is detecting a scan, it sends a signal over a secure channel to this little chap here, who sits between the firewall and the internal network. If I could

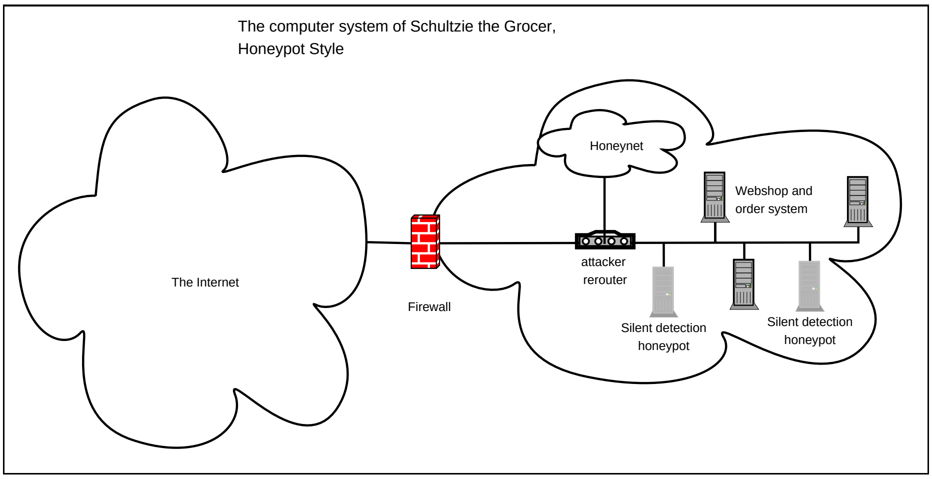

Figure 3.1: *Frank Schultz's computer system with attacker rerouting.*

convince you to take a look over there.." Bubba points to underneath a small table, "you will see three machines running a Honeynet. I've configured it to be exactly like your real system."

"So he won't understand that we're doing things?"

"No, no. I'm following the book. Now we just have to wait for the party crashers to arrive."

Chapter 4

Dunnigan and Nofi

In 1998, Fred Cohen published the paper “A Note on the Role of Deception in Information Protection” (Cohen 1998). In the paper he discusses his honeypot software, the Deception Toolkit (DTK), both with regards to theoretical issues and results from practical use. What is most interesting is the inclusion of a large section of deception theory, on which DTK is based. This is in stark contrast to regular honeypots, where deception is not the main issue.

The deception theory in question was a classification scheme by James Dunnigan and Albert Nofi. The scheme had originally nothing to do with computers, but it has become popular amongst several computer scientists working on computer deception. It was presented in their book on historical deceptions in warfare (Dunnigan and Nofi 1995). The book describes more than 120 situations where deception was used, ranging from the military campaign of the Egyptian pharaoh Ramses II in 1288 BC, to the Strategic Defense Initiative (SDI) and the Gulf War.

The chapter starts with a synopsis of our analysis, before presenting the theory and the analysis itself.

4.1 Paradigm Synopsis

Dunnigan and Nofi’s categorization scheme consists of nine categories, each representing a “traditional deception technique”: Concealment, camouflage, false and planted information, ruses, displays, demonstrations, feints, lies, and insight. No explicit definition of deception is given beyond the categories.

As a preliminary attempt at organizing the techniques, we grouped them by the different aspects they highlight: (1) detection avoidance: Concealment, camouflage (2) how to send information: False and planted information, lies (3) what to show and accomplish: Ruses, displays, demonstrations, and feints (this last group being a catch-all group). The category of insight remained the odd man out, and was not placed in any group.

As an aid in showing the common features of historical deception examples, the scheme functions as intended. But it was not designed for use as a taxonomy, and this makes it unsuitable as the theoretical foundation of a deception theory. When used for in-depth analysis, the results soon diverge and conflict with each other due to the lack of precision, to the lack of structure, and to the inconsistency of application to the computer domain.

4.2 Theory

We will describe the categorization scheme, as given by Dunnigan and Nofi, and see how it holds up with regards to taxonomic requirements. Then we will study the categories in more detail, before we see some examples of how it has been used in computer deception.

4.2.1 A Deception Categorization Scheme

Dunnigan and Nofi's categorization scheme consists of nine categories. Each category represents a "traditional deception technique", and there are few relationships given between them. No explicit definition of deception was given beyond their categorization scheme[1]:

Concealment Hiding by moving behind a natural obstacle (out of sight). *Examples:* Moving a force behind a hill or ships into fog. *Strategic concealment* is hiding one's movements on a grand scale, *political concealment* is hiding one's political objectives.

Camouflage Hiding by artificial means, usually by techniques making the object be indistinguishable from the background. *Examples:* Animals having different color on their fur in the summer and winter, covering tents and equipment with shrubbery and nets, patterns and coloring of uniforms.

False and planted information Letting the enemy see information (often false) on purpose. *Examples:* Invented and written orders or reports.

[1] Examples and their descriptions taken from (Dunnigan and Nofi 1995).

Ruses Make the enemy believe your troops are his, by using enemy equipment and procedures. This is a type of display. *Examples:* Use of enemy uniforms, fake electronic signatures.

Displays Make the enemy see what is not there, by making something appear other than what it is. *Examples:* Light more fires to fake a larger force, create dummies (of vehicles, aircrafts), more radio traffic to imply larger units.

Demonstration A show of military power, in order to confuse the enemy about your motives (imply actions, but do not follow through). *Examples:* Sending a naval force to a coast or a vehicle to a place where it is likely to be spotted by enemy reconnaissance (since the point is to be seen).

Feints A follow-up to demonstrations by actually attacking, but with the intention of distracting while the main assault is elsewhere. *Examples:* Allied forces at Pas de Calais, strengthening Hitler's belief of the main attack being there and not in Normandy.

Lies Lying, directly to the enemy. *Examples:* Diplomats lying about progress, commanders lying about the strength of their forces when parleying (often overstating their strength).

Insight The ability to deceive the opponent by out-thinking him. *Examples:* Using what you know of the enemy against him, by reinforcing his existing beliefs.

Beyond these categories, the book is mostly filled with examples.

4.3 Analysis

This categorization scheme was not developed with computer deception in mind. The viability of the scheme for the use of computer deception is therefore dependent on at least three qualifications: (1) Its internal consistency or correctness (2) its applicability to our specific use (3) its correct application when used.

We have already asserted that as defenders we are in a state of conflict with unknown assailants. That a scheme developed for categorizing deceptions in war should be applicable to computer deception is in this respect plausible.

What remains is to ensure that the categorization have the characteristics of a good classification, and that it is applied correctly to the computer domain.

4.3.1 Taxonomic Requirements

The science which deals with the laws and principles of classification is the science of systematics, also called the science of taxonomies. A taxonomy is a division of a domain into an ordered system of groups or classes, and should have certain characteristics (Howard 1997):

(1) Mutually exclusive There should be no overlap between the classes of the domain.

(2) Exhaustive The domain or space which the classes cover should be totally mapped (exhausted) when the classes are added together.

(3) Unambiguous There should be no doubt as to the correct class of an object in the domain.

(4) Repeatable The classification should be constant over time.

(5) Accepted It should be logical and intuitive, generally approvable.

(6) Useful Further insight be should be gained by using the taxonomy.

Dunnigan and Nofi's categorization scheme is useful when reading the compiled list of examples, as it makes the common themes among the different deceptions clearly visible. It shows that a moderate number of categories (nine) is sufficient to capture the main principles even when the time frame spans millennia. However, these categories are not taxonomic classes (indeed, this has not been claimed by the authors).

It is probably impossible to create taxonomies which excels in all of these requirements, as we seldom manage to acquire a complete overview of the complex reality that exists. We are creating descriptive models that captures the properties we decide are important, while both our detection of properties and understanding of them probably are doomed to be faulty in some ways.

Preferably, the taxonomy would use a single property for classification. If the taxonomy takes the form of a tree, it is not problematic that a different property is used at different levels. But when multiple properties are used at a single level you will run into problems when attempting to classify an object with a new combination of properties: Either you must place it into several classes (ambiguously), or extend your taxonomy. Such an augmentation shows an incomplete and non-exhaustive partitioning of the domain, and is an indication of improper construction of the taxonomy. Extension of the taxonomy is not limited to when multiple properties are used; the same may happen with a single property.

Dunnigan and Nofi's categorization scheme does not have non-overlapping classes of an exhausted domain. In fact, there is not a single domain (unless you count a more abstract "deception" domain) which encompasses all categories.

4.3.2 Technique Classification

While there are few stated relationship between the classes, we can place some of them closer together by the function they perform.

(1) Detection avoidance Concealment, camouflage.

(2) How to send information False and planted information, lies.

(3) What to show and accomplish Ruse, display, demonstration, feint.

Some of these groupings might be considered domains in their own right: (1) The domain of methods for avoiding detection (2) the domain of ways of sending information (3) the domain of contents and intentions (this third group is really a catch-all group for the remaining techniques). Whether or not the categories of Dunnigan and Nofi separates these domains in a good way, or what other elements should be added, is another matter.

The odd man out, not fitting in any domain, is the last category of insight. This category is the one that differs the most from its companions. In a way it is insight that drives the use of the other categories. We use our insight and knowledge to decide what to hide, what to show, and how to do it.

4.3.3 Scheme Application

When the categorization is far from being a taxonomy, it not surprising that when you try to use it as one, you will probably not get optimal results. When different persons try to use it, they will not get the same results. If there is no agreement on what we are talking about, how can we expect consistency?

When applying the categories to computer deception you must try to map the concepts belonging to the computer domain over to the categories. It is not always obvious what should go where, and it is possible to argue for different alternatives. This problem arises when the categories in themselves are not mutually exclusive or defined with precision.

For demonstration, take the category of camouflage. One paper describes it as "hiding your troops and movements from the enemy by artificial means"[2]

[2] Which is a direct quote from Dunnigan and Nofi.

(Rowe and Rothstein 2003). What, exactly, are your troops and movements in the computer domain? Here we are led to the camouflage of key resources, with renaming of key commands as example.

A second paper describes it as "hiding with the use of artificial means." (Tan 2003). Examples are hiding malicious software under innocent looking filenames and code camouflaged as a corrupted packet within a Network File Server. Easter eggs are also considered camouflaged.

A third paper is Cohen's "A Note on the Role of Deception in Information Protection" (Cohen 1998). Camouflage is "based on the creation of an artificial cover that makes it appear as if one thing is in fact another for the purpose of making it harder to find or identify". Examples: A server with information about the US government dressed up like a pornography server, Unix command line prompts looking like DOS prompts and corporate users creating fictitious *.edu* domains for anonymous surfing on the Internet.

Who is right? Well, depending on your own views, all are wrong, all are right, or a mixture somewhere in between. There are no agreements, and therefore no definitions to break.

Another difficulty with using the categorization of Dunnigan and Nofi is that you sometimes get problems due to the inherent nature of the computer domain. For instance, the lines between the categories of false and planted information ("letting the enemy get his hand on information that will hurt him and help you") and of lies ("flat-out lying when communicating with the enemy") are blurred in the computer domain. The description of Dunnigan and Nofi stresses the difference of *form*, not of *content*. This might be clear-cut when dealing with physical objects in contrast to person-to-person communication. But the form of communication in the computer domain is not that distinct.

Examples from the categories in (Rowe and Rothstein 2003) are "[...] files giving addresses of honeypots" (false and planted information) and "false error messages and false file-directory information" (lies). An argument can be made that when the system directly addresses the user (as in an error message) it is a lie, and when the user seeks out the information that is generated by a human being, it is false and planted information. But surely this is a difference of degree and not of kind.

The difference disappears when you consider that while some information might be system-originated (the error-message) or human-generated (the file with false information), in the end the information is treated equally. It is the system that transfers and presents all information. A human being with the proper access can manipulate everything: The origin is irrelevant.

It is claimed that people believe system-generated lies and not information planted by a human being (Rowe and Rothstein 2003). An astute defender realizes that from the view of possible manipulations, it is the same thing - and if the defender can realize this, certainly a sophisticated attacker can do the same, even if the average user does not.

4.4 An Interlude

Third Act: Dunnigan and Nofi

"The jury is still out, but some results are trickling in." Bubba swaps looks between the firewall logs and the Sebek command-line overview. "There's a bunch of people messing around in the Honeynet. Let's see, one, two, three.. I'm dubbing them Alpha, Bravo and Charlie." He writes the IP addresses of each in a small log. "And I think it's time to consider responses."

Frank, meanwhile, is reading. "I don't get this" he complains. "Why are there only nine deception techniques? There is no stated relationship between them - how do you know you have gotten them all? Although I can see that some are related. Somehow. And I feel slightly tricked."

Bubba halts his work and looks up. "Tricked?"

"Two of the best deception techniques for information protection are concealment of intentions and insight. What a surprise. One, don't tell what you are doing, two, be smart. That comforting to know. Please hold all attacks while I get smarter."

"Ah. I see what you are getting at. But think of it this way: One, take control of all information altered by the deception visible to the attacker, two, know the enemy, his methods and objectives."

"Well. Okay. I can buy that." He reads on. "And they say that lying is good. By the computer system, that is. Faking files is of the good."

Bubba nods. "Bingo. It seems one of our new friends here is trying to read your customer database, but he doesn't have access. I think we should update with a new one, and forget some access flags."

"But that isn't false files - that's false content."

"Same argument as before. There's no point in only lying about the system, unless it is a target of opportunity attack. Here we clearly have a target of choice-attack, and must create content, in the spirit of Stoll. To work, Frankie."

Frank begins the task of creating a false customer data file, which he saves on a Honeynet computer. After a while, Alpha notices the new file and downloads it.

Bubba gets a feral look in his eyes. "Ka-ching!"

Chapter 5

The General Theory of Barton Whaley

> Perception: The action, faculty, or product of perceiving.
>
> > I. From the literal sense of L. percipere, to take, receive.
> >
> > II. From the secondary or metaphorical sense of L. percipere, to be or become cognizant of.

(Oxford English Dictionary)

> I will go further and assert that deception is the same regardless of whatever field it appears in. It is not a function of technology. All deceptions are applied psychology - the psychology of misperception - and hence are connected by more than one or two analogous points. Consequently, along psychological lines it must be logically possible to develop a general theory of deception.

(Whaley 1982, page 179)

In 1982 Barton Whaley published a paper called “Toward a General Theory of Deception” (Whaley 1982). The paper sought to use perception as the basis for a complete theory of deception, irrespective of the field it was applied to. Whaley presented a taxonomy, several structural properties of deceptions, and a basic planning and execution process.

Whaley saw deception as the result of erroneous perception. We perceive the world and build a reflection of that world inside our brain. When we mistakenly believe something to be true when it is not, or vice versa, we have suffered a misperception: We have not seen the world correctly. Accordingly, knowledge of the perceptual process is essential for understanding deception.

The chapter starts with a synopsis of our analysis, before presenting the theory and the analysis itself.

5.1 Paradigm Synopsis

Barton Whaley has created an extensive theory of deception, with a perception taxonomy, structural properties of deception, and a planning and execution process.

However, there exists a set of flaws in the theory, mainly that it mixes conscious and unconscious perception, and makes large assumptions about how the brain works. These assumptions further influence Whaley's theory of the structure of deception; if we want to avoid those assumptions we must modify the structural theory.

In summary, three areas stand out from Whaley's theory.

(1) On the Nature of Deception

We have learned a fundamental truth about deception: It consists of simulation and dissimulation. And by this we do not think of the linguistic trick of inverting what is happening (hiding an item becomes showing the absence), but of the basic truth that things can be shown or hidden, to be or not to be.

(2) On the Nature of the Human Mind

The two fundamental operations can be combined in various ways to create effects to be shown. Whaley suggested six different effects in his table (5.1 on page 44), we grouped them into 4 (hiding, confusing, misidentification, attention diversion or attraction).

(3) Deception Planning and Execution

Whaley's 9-step planning and execution chain began with 6 planning steps and 4 execution steps; by excluding his hypothesis-generating theory we can shorten it to 5 planning steps and 3 execution steps: (1) decide on a strategic objective (2) decide deception objective in terms of target action (3) what should the target think (4) what shall we portray (5) how do we portray (6) execute (7) indicators are shown to the target (8) the target accepts. See figure 5.3 on page 51.

While executing, the target's responses should be monitored (feedback) to assure that they are as desired, and to make adjustments if needed.

5.2 Theory

The theory of Barton Whaley is an attempt at an encompassing theory, and includes several aspects of deception. We start with a taxonomy of perception, and see how deception is a consequence of the workings of our perceptual system. Then follows a closer look at the structure of deception, and a planning and execution process.

5.2.1 A Taxonomy of Perception

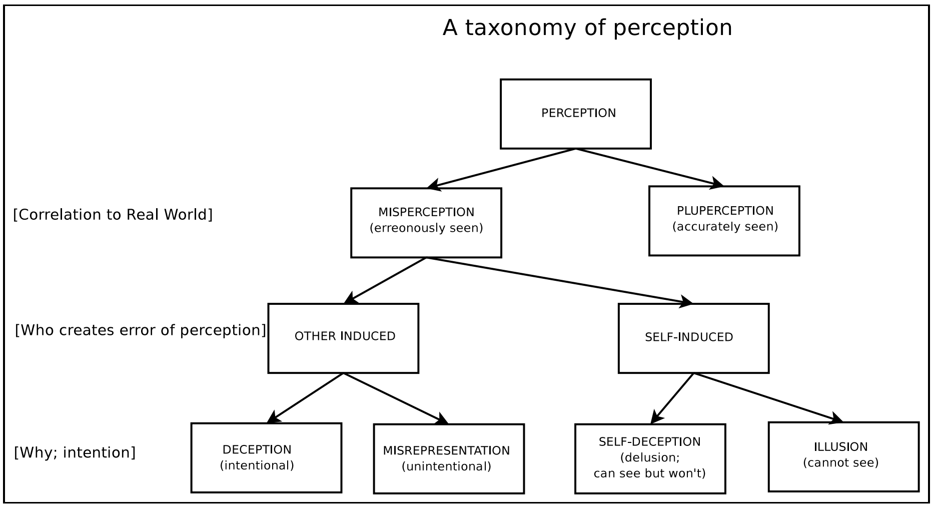
A taxonomy of perception
PERCEPTION
[Correlation to Real World]
MISPERCEPTION (erreonously seen)
PLUPERCEPTION (accurately seen)
[Who creates error of perception]
OTHER INDUCED
SELF-INDUCED
[Why; intention]
DECEPTION (intentional)
MISREPRESENTATION (unintentional)
SELF-DECEPTION (delusion; can see but won't)
ILLUSION (cannot see)

Figure 5.1: *Whaley's taxonomy of perception.*

The taxonomy is a four-level classification of the perception domain. For each level (excepting the first) the class from the previous level is partitioned into two classes by a single property. The property is different from each level; see figure 5.1 and the description below.

Perception Level 1: The overall class of perception; the domain.

Misperception and pluperception Level 2: Partitioning of perception into two classes, based on correctness. There exists only one Real World, and this world can be correctly or incorrectly perceived.

Other induced and self-induced Level 3: Partitioning of misperception into two classes, based on the inducer of the error. The reason for the misperception is either ourselves, or another entity which has manipulated our perception process.

Deception and misrepresentation Level 4: Partitioning of both other-induced and self-induced into two classes, based on intention. A deliberate inducement is deception, while unintentional inducement is misrepresentation.

By using this taxonomy we get a definition of deception as *the intentional distortion of perceived reality*. Implicit in Whaley's use of the taxonomy is the existence of an entity which perceives (the target) and an entity (the deceiver) which purposely distorts the reality perceived by the target.

5.2.2 Finding Deception in Perception

If deception is a type of perception, then we must understand how perception works in order to manipulate the process to our advantage. Whaley based his theory on the work of a neuropsychologist, Professor R. L. Gregory (Gregory 1973).

Gregory had looked at causes and mechanisms behind visual illusions. Clearly the process of perception was to blame, but in order to describe that process he had to choose an initial definition to work with. He saw that the definition of perception had a crucial impact on the possibility of explaining illusions. Gregory abandoned the Kantian view of perception as intuitive truth, since illusions could not be explained in any satisfactory way (false intuitive knowledge?). Instead he continued with the work of a German physicist, Hermann von Helmholtz, who regarded perceptions as conclusions to unconscious inferences. From this Gregory saw two possible causes for illusions:

Our physiological perceptual mechanisms malfunction A physiological mechanism does not perform its function correctly.

Our cognitive hypothesis-generating strategies are inappropriate The functions carried out are not suitable to the problem at hand.

Gregory provided examples of illusions caused by both types of failures, and described various characteristics inherent in the perceptual system. Whaley later tried to make explicit the process of perception, and described a five-step process:

1. The environment continuously transmits a spectrum of discrete data bits.

2. Our sensors detect certain portions of some of these spectra.

3. The received data are transmitted with delay and distortion to the brain.
4. The brain discards most of these data, processes some of it and stores it in memory.
5. The brain develops hypotheses about the environment by drawing inferences from new and stored data.

Based on this process, Whaley regarded *thinking* as the cognitive process of testing hypotheses with data. *Learning* would then be the accumulation of more and more interrelated hypotheses.

Since a major cause of the possibility of deceptions arises from the way our hypothesis-generating brain functions (step 4 and 5), Whaley gave a theory of how this generation occurs. He separated the generation process into three levels of aggregation.

Categories The brain interprets the discrete bits as categories, which are hypotheses about their likeness or difference. The categories might be seen as Plato's 'ideal types'. The hypotheses are then stored in memory.

Characteristics Sets of related bits are combined into characteristics (charcs). As the previous level, these sets are hypotheses, but of the relation between categories. They are also stored in memory. New sensory data input are compared to the existing characteristics, accepted and incorporated if in concordance or abandoned if incongruent.

Patterns The categories are combined into patterns, which are hypotheses about the relation between characteristics. As before, these are stored in memory, and the same comparison process with acceptance or abandonment occurs with patterns.

It is in the nature of nature, so to speak, that a given set of sensory data may support more than one hypothesis. The brain has its own biases towards which hypotheses it will be open to accept, due to styles of thinking and cultural bias.

5.2.3 The Structure of Deception

All acts of deception, in nature or by man, are built using two basic concepts:

Dissimulation *Hiding the real*, concealing or obscuring the truth.

Simulation *Showing the false*, presenting or portraying a lie.

Using his hypothesis-generating theory of humans, Whaley described *operational* dissimulation as done by hiding one or more characteristics that make up a pattern of something real, and simulation as done by showing one or more characteristics of a pattern that is false. You should always hide the actions you take when executing your deception, while you show the effect the targets should perceive.

Simulation and dissimulation can also be considered as two sides of the same coin. A simulation can become a dissimulation and vice versa, if only implicitly by the point of view taken: Hiding an item, for instance, implies showing the absence of the item in question.

Whaley arranged six deception categories in a 2x3 table, three dissimulation techniques that stands in a logical relation to three simulation techniques. See table 5.1.

Dissimulation	Simulation
Masking Hiding the real by making it invisible	**Mimicking** Showing the false by imitation
Repackaging Hiding the real by disguising it	**Inventing** Showing the false by creating something new
Dazzling Hiding the real by confusing the target	**Decoying** Showing the false by diverting attention

Table 5.1: *Whaley's six simulations and dissimulations.*

Not only are the items in opposition to each other (*masking - mimicking, repackaging - inventing, dazzling - decoying*), but there is a natural progression from technique one to three, including a descending order of effectiveness. The idea is, preferably *mask* your pattern, if that is impossible, *repackage* it, if that is impossible, use *dazzling* as a last attempt at confusion. The same can be done with the simulation techniques: First *mimic* another pattern, if that is impossible *invent* a new pattern, and try *decoying* if this too is impossible. Whaley gives operational descriptions of all of these techniques using his categories / characteristics / pattern theory.

5.2.4 Planning and Execution

Whaley gives an overview of a general deception planning and execution process which the deception planner must go through. The process consists of ten steps, of which the first seven belongs to the planning phase and the last three belongs to the execution phase. Not every step is necessarily done consciously. The last two steps are not so much actions to be performed, as things to watch out for. Whaley uses examples of war and magic, as he has done deception research in these two areas.

1. **Know the strategic goal** This is the objective of the whole enterprise, and is usually an outside parameter given to the planner. A magician seeks to amuse and entertain, while a military commander at the highest level supports political strategy.
2. **How should the target react to the deceptive measures taken** A magician usually wants the attention of his audience to be diverted at the crucial moment, a military commander wants specific reactions from the enemy ranging from simple to complex.
3. **What should the target perceive** After deciding what the target should do, one imagines what the target must think in order to perform those actions.
4. **What should be hidden and shown of the forthcoming events** The impending events must be modified to show what we want the target to perceive.
5. **Analyze the pattern to be hidden** Identify the characteristics, and give another pattern by masking, repackaging or dazzling.
6. **Analyze the pattern to be shown** Identify the characteristics, and give another pattern by mimicking, inventing or decoying.
7. **How to present the effect to the target** The combination of what you hide with what you show is an effect, which is portrayed by a method that must remain hidden to the target. The means to portray this effect might demand resources unavailable to the deception planner, which forces the planner to reconsider his decisions.
8. **The operational phase** The effect must be 'sold' to the target. A magician is possibly both the deception planner and executor, while a military planning staff may pass orders to an operational unit.
9. **Open channels are needed** The effect must be presented to the target's sensor. A deception which does not reach the target will not cause any change in his behavior.
10. **The target must accept the deception** Several circumstances exist which can result in failure: The target does not see the effect, decides it is unimportant, misunderstands it, or possibly detects the method used to portray it. To be successful, the target must notice the effect, find it interesting enough to hold his attention, generate the intended hypothesis and fail to detect the method.

In addition feedback from the target should be observed to ensure his attention, perception and reaction remain as desired and the deception undetected.

Whaley does not use much space discussing counterdeception, except to say that it will always be possible in theory. It is impossible to completely dissimulate or simulate an event or object, which means there will be indicators available revealing the existence of the deception.

5.3 Analysis

Whaley's paper is possibly the first attempt at a general theory of deception. That is a highly challenging and laudable endeavor. We will attempt an analysis with the following questions in mind: Does the theory conflict with any known facts and is it logically consistent.

We begin by looking at the taxonomy and how it conforms to taxonomic standards. After this we will see how the use of the word perception leads to an ambiguity. We continue with what the perception process assumes with regard to the perceptual system, and an analysis of Whaley's hypothesis-generation strategy as well as the structure of deception. Then we will look at the planning and execution process, before we end with some ruminations on what role the computer could play in this paradigm.

5.3.1 The Taxonomy

Whaley's taxonomy conforms without problems to the demands required of a good taxonomy. For each level the classes are mutually exclusive, exhaustive, and unambiguous. The fact that there is only one property used for classification at each level contributes to this. The taxonomy is presumably constant over time and seems logical. As for usefulness, it forms a context which provides a setting for the rest of Whaley's theory.

The strength of the taxonomy is seen when compared to the categorization scheme of Dunnigan and Nofi, which lacks many taxonomic characteristics. Although this might be unfair to the latter, which does not claim to be a taxonomy, and whose categories actually are within Whaley's class of deception. It is much harder to partition the deception class than the perception class, which we see if we try to lay taxonomic requirements on Whaley's six element structure of simulations and dissimulations.

While the various classes are easy to separate from each other in the taxonomy, there exists an ambiguity which Whaley either has missed or ignored. It results from the use of the word perception.

5.3.2 What Does Perception Entail

We have seen perception defined as unconscious hypotheses about the environment. Now, Gregory loses the word "unconscious" when he is not quoting Helmholtz directly, but the constraint still holds. Perception is always regarded as an unconscious process done by the perceptual system. The selection of strategy is not done by conscious choice, but by a process we have neither control over nor are aware of. This is what makes perception active and not passive. Every example by Gregory maintain this distinction, even if it is unvoiced (Gregory 1973).

However, this use of the word is not universal, not even when used by other works on deception. Consider a RAND report on deception and urban military operations (Gerwehr and Glenn 2000):

> Humans, like animals, must make decisions in order to survive. Decisionmakers rely upon their assessments of other actors' interests, intentions, and capabilities, as well as an assessment of the environment or context within which the action takes place. These assessments - or perceptions - engender policy preferences and galvanize action.

Richard Heuer, on strategic deception and counterdeception (Heuer 1981):

> Perceptions are quick to form but resistant to change. Once an impression has been formed about an object, event, or situation, one is biased toward continuing to perceive it in the same way.

These two sources use a different definition of perception. Here, perception is the sum of all opinions and meanings, not unconscious hypotheses. Consequently, the word perception covers more than the process done by the perceptual system, it also covers human thinking, reflections, and assessments.

All of Gregory's work is related to the unconscious, but active, perceptual system. Whaley does not acknowledge this distinction, and it is sometimes difficult to see if we are dealing with human thinking or unconscious perception.

An indication of this is to be seen in the definitions of learning and of thinking: " [...] 'thinking' is the cognitive process of testing hypotheses about incoming and stored data [...] 'learning' is the accumulation of more and more interrelated hypotheses."

If the unconscious perceptual system can learn, which Gregory incidentally indicates, then the definition given by Whaley can be taken in the spirit of

Gregory. But the perceptual system is unlikely to think, in the normal use of the word; and it looks like Whaley here states that all human thinking is basically the testing of hypotheses. This seems too simplistic to be true, Ockham to the contrary.

5.3.3 The Perception Process

Whaley's five-step perception process is based on the work of Gregory, and is assumed to describe the entire process, although crudely, as Whaley states. It contains as few specific assumptions as possible, which elegantly minimizes the problem of becoming outdated when science advances. Some assumptions, however, are always unavoidable, and what Whaley has done is to model the human perceptual system as a sensor system: We have *sensors* (our senses), a *processing center* (the brain) and *data flow* (by the nervous system) in between.

We are now saddled with possible perceptual errors (ie deceptions) from two different sources. Gregory gives us two classes of perceptual errors (mechanical and strategic) with examples from actual experiments, while an analysis of Whaley's perception process as a sensor system might give rise to another set of possible errors.

What we really want to know is if the model of Whaley fits the experimental examples given by Gregory. And it looks like that a pure sensor-based model is insufficient to explain all experimental data. For instance, "prior knowledge of objects affects apparently primary sensations and perception" (Gregory 1973). We see this when we mistakenly judge larger objects to be heavier, or when we try to walk on a moving staircase which stands still. There is also the fact that human sensors do not work independently (NewScientist 2004) and that emotions affect how we interpret perceptions (Ekman 2004).

5.3.4 Whaley's Hypothesis-Generation Strategy

It is difficult to know how to interpret the hypothesis-generation theory of Whaley. Is this a conscious or unconscious process? In both cases there is a lack of empirical proof indicating the use of categories, characteristics, and patterns. How do we know there are three, not four or five levels, or that there are any such levels at all? Is comparison the basic method by which the brain functions? In fact, the theory of "Toward a General Theory of Deception" (as stated by Whaley himself) is a refinement of previous work published by J. Bowyer Bell and Whaley. In "Cheating and Deception" (Bell and Whaley 1982), there is no mention of categories, only characteristics and patterns. Was there a scientific basis for adding another level, besides making the model more smooth?

It is hard to find any hard, conclusive evidence to support this hypothesis-generating process. That the brain must recognize objects seems clear, the method less so.

5.3.5 The Structure of Deception

The six-element table of simulation and dissimulation techniques is a logical construct of techniques supposedly in opposition to each other (see table 5.1 on page 44).

The techniques do not seem to be real mirror-image antonyms. The antonym to masking is displaying or exhibiting, not mimicking. The connection between repackaging and inventing, dazzling and decoying is not obvious. If you wish to organize techniques into blocks of hiding and showing, it would be more logical to have masking as the sole element of hiding and the rest in showing.

In the table the techniques are linked to the categories / characteristics / patterns theory of Whaley. If we ignore that theory and instead look at the techniques solely for themselves, we get a set of effects which can be sought by deception. We can mask an object, avoiding its detection. We can mimic or repackage it so that it is detected as something else that is known. We can invent a new object, and have our old one detected as that. We can dazzle the target by modifying or creating an object that makes no sense, and we can try to attract the targets attention by decoying.

What we more or less are doing here, is deceiving a pattern matching device. If we assume a pattern matching sensor system is operating in an area, these are some of the actions we could take to fool it. See figure 5.2 on the following page for a mapping of Whaley's categories to a pattern matching model.

The two fundamental operations can be combined in various ways to create effects to be shown. Whaley suggests six different effects in his table; let us see if we can make place them into some sort of groups.

(1) **Hiding.** The category of masking. Both concealment (being outside the sensor's point of view) and camouflage (being seen, but not detected by blending with the environment) are covered.

(2) **Confusing.** The category of dazzling. The sensor (or interpreting center) does not understand the data that is received.

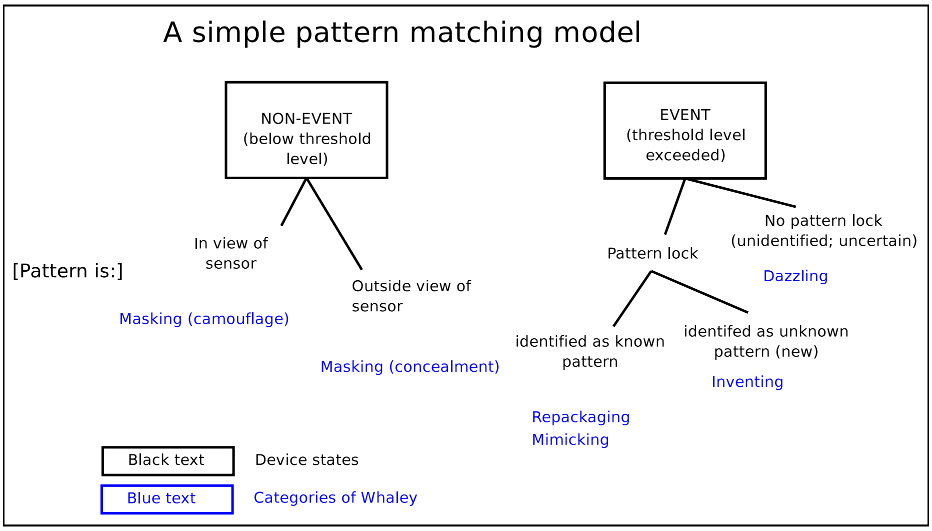
A simple pattern matching model
NON-EVENT (below threshold level)
EVENT (threshold level exceeded)
[Pattern is:]
In view of sensor
Outside view of sensor
Masking (camouflage)
Masking (concealment)
Pattern lock
No pattern lock (unidentified; uncertain)
Dazzling
identified as known pattern
identifed as unknown pattern (new)
Inventing
Repackaging
Mimicking
Black text
Device states
Blue text
Categories of Whaley

Figure 5.2: *Model of pattern matching. Note: Whaley's category of decoying is absent, since sensor focus is not depicted.*

(3) Misidentification. The categories of repackaging, mimicking and inventing. An object or event is detected, but not identified as what it really is.

(4) Attention diversion or attraction. The category of decoying. The sensor's focus is drawn or repelled.

Note that as humans, we have several types of foci. The "attention" of the unconscious perceptual system can be drawn by unanticipated noises or movements, while our conscious focus can be drawn or repelled by something we find interesting or uninteresting.

5.3.6 Planning and Execution

While Whaley's planning and execution process is colored by his categories / characteristics / pattern theory, the major insights are not directly dependent on it. We see a lot of the same thoughts in the military deception planning and execution process we will look at later (Chairman Joint Chiefs of Staff 1996).

First of all, deception is a supportive tactic or strategy. There is no inherent value in deceiving someone for the sake of deception. The reason for employing deception is to help achieving an external strategic objective.

We want a specific reaction from the target, the assumption being that the reaction will support our strategic objective. The reaction can be considered the deception objective.

The reaction must be provoked by inducing certain beliefs in the target, the assumption being that the beliefs will make him take the desired actions. Further, we make assumptions about what the target must perceive[1] in order to get these beliefs.

When we know what the target should perceive, we must use our available means, technological and otherwise, to present the deception to the target. The deception is presented through channels to the target's sensors. Channels that are closed to the target are useless for portrayal. The success of the deception depends on the target's acceptance of the deception, and on the actions he takes as a result. See figure 5.3.

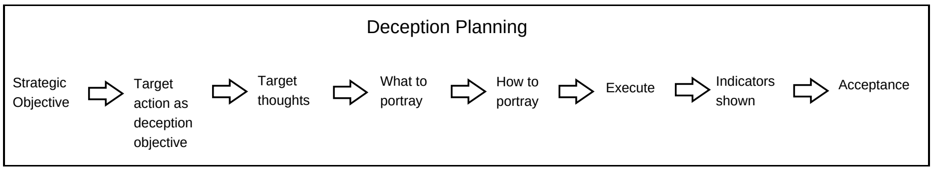

Figure 5.3: *Whaley's planning and execution process.*

5.3.7 Computer Deception

We have, in this chapter, rushed headlong into a field outside our purview. By using common sense we have tried to maneuver in unknown waters, looking at deception from the angle of psychology. If we try to capture the essence of Whaley's theory in a few sentences, an adequate example could be as follows:

(1) When deceiving we create effects to be show by simulation and dissimulation to support a strategic objective (2) In order to be successful, we should plan from the strategic objective to the portrayal, ensure open channels, and verify deception acceptance by feedback.

It is easy to forget that we are in fact not investigating deception, but *computer* deception. Since Whaley attempted a general theory, it is natural to ask, where do the computers fit? Whaley nowhere mentions computer deception or computers, which, considering the time of publication (1982), is not so strange. The closest we get is when he mentions radars, taken to be a type of external sensor feeding information to the human eyes.

A radar is, in many respects, a type of computer. Conversely, can we con-

[1] In the combined conscious/unconscious sense.

sider the computer to be a type of external sensor? A computer also feeds information to the human operator, who process the information further. But after some reflection, the notion proves to be too simplistic: Computers (and radars, to some extent) do more than show information, they also process, manipulate and refine information.

In the paradigm of Whaley, computers have no special or acknowledged place. Since deception is entirely about deceiving human beings, everything else can be considered as aids in this regard. This is what a computer is - an intermediary between deceiver and target. As to how the computer can be best used to deceive humans, the theory does not say.

5.4 An Interlude

Fourth Act: Whaley

"Well, this doesn't tell me anything about computer deception at all." Frank waves a paper in the air. "It claims that deception is only about deceiving humans. Aren't we deceiving computers?"

Bubba shakes his head. "Think about it. It's not the computer that is making decisions - it's the human operator. The computer system is merely the environment where we manipulate."

"So we're simulating and dissimulating in the computer system, in order to dazzle, mislead, mask or whatever?"

"Yup. It's all about the human being."

"Huh." Frank continues to read. "But Bubba? What is our strategic objective?"

"Our whatnow?"

"Strategic objective, you know, why are we doing the things we are doing? Planting false files?"

"To make the attacker believe he got what he wanted."

"Which helps us how?"

"Well, I, uh.." Bubba stops to think. "It seemed like a good idea? No, I mean, it's to keep him occupied."

"But all he has to do is to see where I drive when delivering groceries, and then he'll know that we have given him a false file."

"Uh-oh. Verification against external facts. That's not good. This demands a rescue operation." Bubba picks up a cell phone. "We can at least cover our base on the phone numbers. They are false, right?"

"Didn't want the possibility of anyone intruding on my customers personal space."

"Good. I'll be right back." Bubba vanishes out of the room. Five minutes later, he is back with a triumphant grin. "Do not ask what you can do for the country."

"What?"

"Nothing. I've got the phone numbers covered. If anyone calls them, they'll be answered by some friends of mine." He throws the cell phone on the table.

"That's neat. How did you manage that?" Frank asks.

"They, ah, work at the phone company."

"But why would anyone call my customers?"

"That's what we'll find out if anyone calls. Attacker Alpha has the list, so we'll see

what he does. The waiting game again."

Chapter 6

Cohen's Framework for Deception

> While our study will seek general understanding, our ultimate focus is on deception for information protection and is further focused on information technology and systems that depend on it. At the same time, in order for these deceptions to be effective, we have to, at least potentially, be successful at deception against computers used in attack, people who operate and program those computers, and ultimately, organizations that task those people and computers. Therefore, we must understand deception that targets people and organizations, not just computers.
>
> (Cohen, Lambert, Preston, Berry, Stewart, and Thomas 2001)

"A Framework for Deception" is a comprehensive paper seeking to map out the essentials of deception for information protection. The authors have investigated several areas they deemed relevant, including but not limited to military deception, cognitive deceptions as physiological deceptions, negotiation tactics and intelligence analysis. Some existing computer deception efforts are also reviewed.

It is impossible in the allotted space to fully describe or analyze every aspect of this paper, nor do we possess the required expertise to attempt such an endeavor. We are forced to focus on selected parts of the paper, which we find the most relevant.

The chapter starts with a synopsis of our analysis, before presenting the theory and the analysis itself.

6.1 Paradigm Synopsis

Cohen et al take the view that computer deception is inseperable from human deception. A computer can be deceived only so far as you define its goals and objectives by considering the humans that design, program, and operate it.

Deceptions are induced or inhibited effects on cognitive structures, which are possible due to limitations and inflexibilities in the structures. A human three-level cognitive structure was given, based on previous experimental and theoretical work from psychology and cognition. Using the same approach of computers, a seven-level model of computer cognition was presented.

To describe the nature of deception, sixteen dimensions or properties were given, most concerned with the target. A thorough investigation of all of these was beyond our timeframe, but we grouped them into (1) fundamental deception properties (2) target properties (3) target and deceiver properties (4) execution properties (5) societal constraints.

A framework of human deception in three parts was briefly presented. It consisted of a set of primitive techniques, the properties of those techniques (ie the dimensions), and a syntax and semantics for applying those techniques.

6.2 The Theory

This section largely mirrors the structure of the paper, with some parts ignored. We start with the nature of deception before entering the area of cognition. Two cognitive models of human beings are described, as well as one for computers. Then follows a brief description of planning and execution guidelines based on the human cognitive model, and a quick look at the proposed framework.

6.2.1 The Nature of Deception

After giving an overview of efforts in several deception related fields, the authors begin straight away at tackling the nature of deception. Acknowledging the difficulty of defining deception accurately and with precision, they remain content with asserting:

> Deception is a set of acts that seek to increase the chances that a set of targets will behave in a desired fashion when they would

> be less likely to behave in that fashion if they knew of those acts.

The focus does not remain on the definition, but is quickly moved to a key insight: Computer deception is not restricted to deception of computers, but involves the human beings that operate and program the computers, and even the organizations that task the human beings. We will largely ignore the theory concerning deception of human beings in groups and of organizations, although we acknowledge that they can be of high importance, depending on the deception objective.

Working toward building a framework, the authors follow with no less than sixteen deception dimensions that attempt to capture the vital characteristics of the nature of deception. Instead of explicitly stating them all, we will give a summary of the most important points.

The possibility of deception results from a set of inherent characteristics of nature and the targets of interest. When a target interacts with the world, events are reflected by observables that the target can see or sense. All deceptions are mixtures of simulations and concealments[1] of observables seen by the target.

Most targets have limited resources and memory, limited abilities to process received information, and inflexibilities in their cognitive structure. These limitations contribute to a predictability that can be exploited, both directly by actions induced by false information, as well as indirectly by controlling the target's focus of attention.

We, the deceivers, use our knowledge and feedback from the target to plan and adjust the deceptions, but we cannot know the internal state of the target. This means that modeling deceptions is tricky business, always fraught with danger and uncertainty. We are vulnerable to counterdeception since we are limited by what we can see and not see in the same way as the target. The recursive nature of deceptions makes it possible go back and forth: Counterdeception, countercounterdeception, countercountercounterdeception and so on. Operational security is a necessity to keep information that can reveal the existence of the deception from reaching the target.

Deceptions can be simple one-step actions, or involve complex sequences. Depending on the complexity and the desired effect, timing requirements can range from the trivial to demands of high precision and accuracy. Some deceptions rely on physical reflexes and are nearly instantaneous, while others require strategic planning and months or years of execution time, with constant monitoring of feedback and adjustment of means. Changes wrought by deceptions do not necessarily have to be large, in order to have

[1] In Whaley's parlance, simulations and dissimulations.

large consequences. An entire organization can be affected by targeting the right person.

Unintended consequences can be hard to avoid if the changed observables affect others than the target, or as an indirect consequence of changes in the target's behavior. Legality can be an issue regarding the deception directly or due to unforeseen consequences.

6.2.2 Lambert's Cognition Model

In "A Cognitive Model For Exposition of Human Deception and Counterdeception" (Lambert 1987), Dave Lambert presented a model of human cognition based on states and processes. This model was later used in "A Framework of Deception" by Fred Cohen, Lambert himself and others when looking at deception for information protection.

The model used components like sensors and affectors, managers and controllers, executives and buffer memories to account for brain functions. Cognition processes were depicted by showing the flow of information from the senses and up through the cognitive structure, gradually increasing the aggregation of information from feature perception, to form perception, to basic comprehension and association to the top of the "self". When decisions were taken, the execution followed the same structure downwards. See figure 6.1.

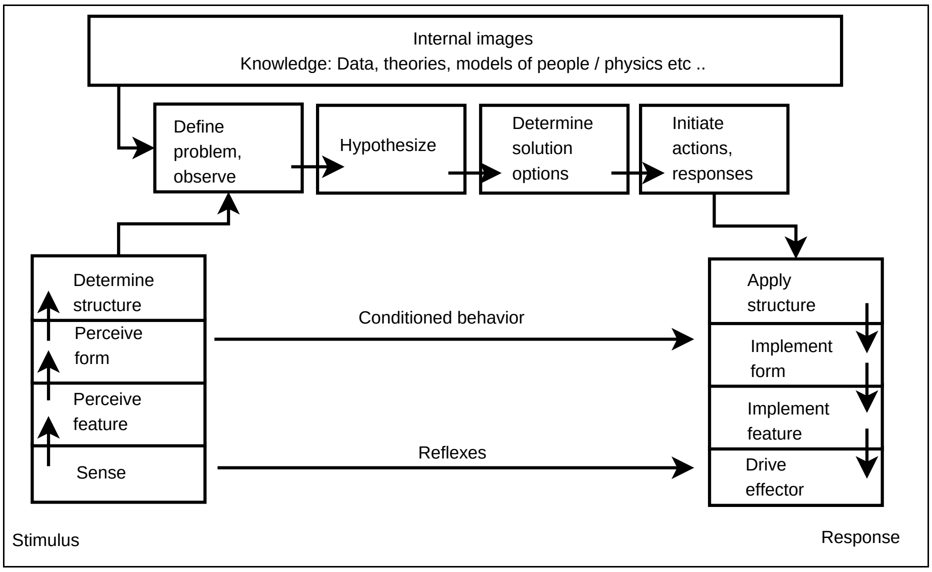

Figure 6.1: *A simplified depiction of Lambert's cognition processes.*

We receive continuously impressions from the world, which our senses interpret according to the state we, or rather our cognition system, are in. By controlling the impressions it is possible to induce states in the cognitive system. This is done by inhibiting (hiding or blocking) or inducing (portraying) impressions, and as a result, effects within the cognitive structure.

Low level effects in the structure are typically predictable, like visual and audiovisual deceptions which are not within conscious control. The loud noise or flashing light will draw our attention before we realize it has happened. Conditioned reflexes are below full consciousness, and can also be exploited. However, for longer term or higher level deceptions that involve reason, assessments, and reflections, the model was deemed inadequate.

6.2.3 A High Level Cognitive Model

By building on the work of Lambert, a new compatible model was created. This high level model tries to account for how humans act, interact and affect each other. Simulation and concealment of target observables are still the fundamental deception operations.

The model has three levels of human cognition, dubbed the low, middle and high levels of cognition. See figure 6.2 on the next page.

(1) Low-level cognition This level consists of the two lower levels of Lambert's model, and covers unconscious perception and reflexive responses. Visual and audiovisual deceptions function at this level.

(2) Mid-level cognition This level covers learned conditioned responses, (nearly) automated capabilities based on pattern matching, and instinctual responses. More conscious thinking are involved than automatic reflexes.

(3) High-level cognition This level covers logical reasoning. We are still influenced by factors like authority, emotions, charisma etc. Negotiation tactics are referred to that show many techniques working at this level.

By using different strategies it is possible to shift the level of cognition of a target. Rush, stress, distraction and fatigue leads to more automatic responses, which is useful if we want to slip slight variations in expected impressions by the target. In the opposite case, the level can be shifted up if expectations are not met, inconsistencies are great, and the resources needed for more thoughtful reasoning are present. If we need to make great changes that will be easily visible, we must anticipate high level cognition being used, and construct the deception accordingly.

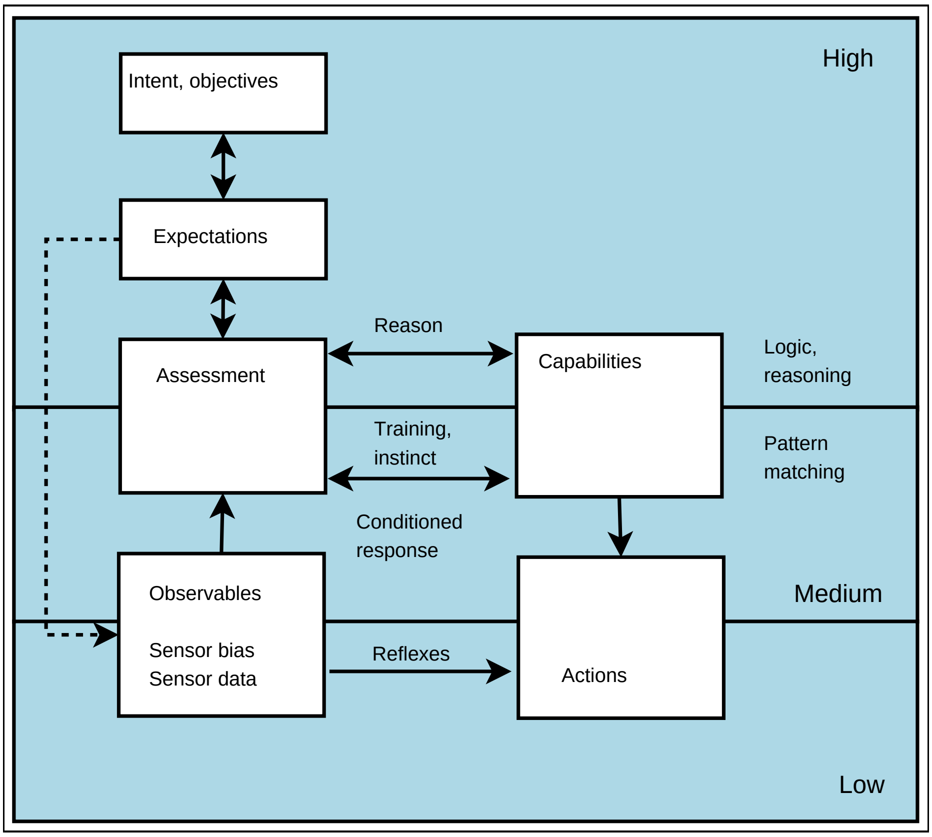

Figure 6.2: *A simplified depiction of Cohen's three-level cognitive structure.*

6.2.4 Computer Deception

A computer is not an intelligent being, but an automaton. This means the computer cannot be aware of anything, in the same sense as a human being. Deceptions employed against a computer are in reality employed against the humans that design, program, and use the computer.

Cohen built a model of computer deception by using the same approach as towards human deception. Just as a human being, the computer has a cognitive structure where signals can be induced or inhibited at different levels. See figure 6.3 on the facing page.

Information flows from the bottom of the structure, the hardware layer, and upwards through drivers, protocols, the operating system, libraries, and applications. Inside the applications there might be embedded languages, which can support languages defined in themselves, leading to recursivity.

While the computer as an automaton is unconscious, it has expectations

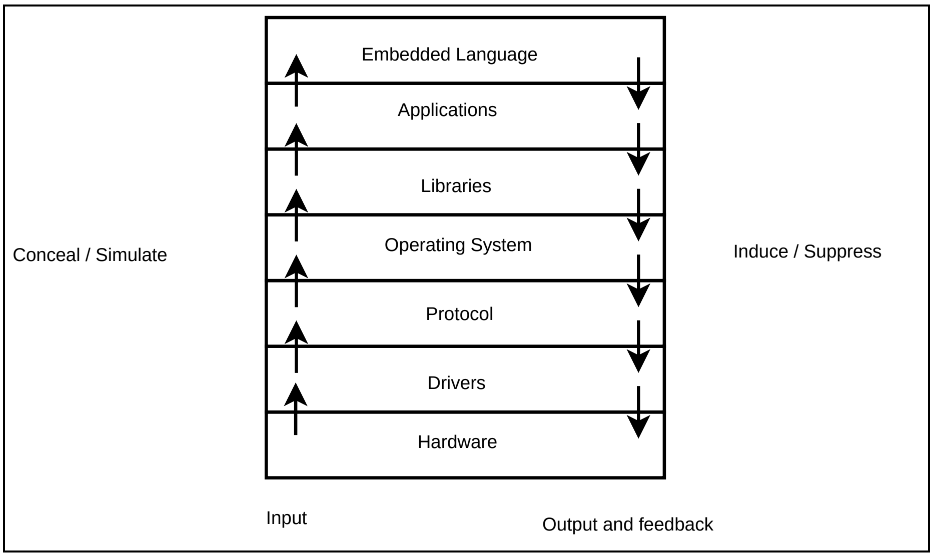

Figure 6.3: *A simplified depiction of Cohen's cognitive structure of a computer.*

and sometimes objectives, even if these objectives are derived from the intent of humans. In one way, this means a computer can be deceived. Cohen suggests three approaches: (1) creating a deception environment which has as high a fidelity as possible to a real environment (2) defeating specific tools by exploiting knowledge of their inner workings (3) modifying functions to comply with your needs.

When it comes to computer network deceptions, they always involve people unless you count misinformation passing between computers, as with cascades failures in larger systems like the power grid or the telephone systems.

Most of the deceptions of humans over networks are extensions of old deception types into the new domain. Examples are false personas, identity theft, fraudulent billing, false postings to affect the stock market, and the transfer of computer viruses that relies on tricking users into running infected programs.

While there is no explicit theory presented of how the computer's cognitive structure affects a human being, four different scenarios are mentioned of how an attacker can use the computer in attack, and how they can be countered:

(1) Computer Only A fully automated attack. In such a situation we are fighting with a computer, not a human operator. Analysis of the tools used or simple automated responses can be effective in maintaining enemy computer expectations.

(2) People Only Manual attack with no automated tools. If rerouting is achieved without the human attacker noticing, three options are available: (1) Maintain expectations to waste attacker time and resources (2) slowly change attacker expectations to our advantage in small unnoticeable increments, like slowing response time down (3) induce cognitive dissonance to create uncertainty

(3) People With Poorly Integrated Computers Automated tools with bursts of human activity. This is the common attack form. Deceptions can be employed to deceive through all attack phases: (1) Intelligence (2) system entry (3) privilege expansion (4) capability planting (5) exploitation. Results of the tools in any phase will be believed if they are congruous with normal expectations.

(4) People With Well Integrated Computers Intelligence enhancement by the use of computers. This has not been observed.

6.2.5 Planning and Execution

Cohen covers two approaches to deception planning and execution. One refers to military sources and principles, while the second is based on the model that is presented in the paper. We will look closer at the first approach in the chapter on warfare, and briefly present the second here.

By considering the characteristics of the human cognitive structure, one can get some general guidelines for achieving effective and successful deceptions.

By the nature of the model, certainty decreases as one moves upwards. Therefore, deceptions should be carried out at the lowest possible level, and care should be taken to avoid dissonance and uncertainty that force higher level cognition. The best is to hide objects so that they are not sensed by the target's sensors, or to present a perfect simulation of a desired false situation.

If low level deceptions are impossible, try working at mid-level by remaining within normal expectations or using techniques like time pressure, stress, training exploitation etc. Feedback is increasingly important since the deceptions become more tailored to the specific target, and there is a higher need for adjustments while deceiving.

At the highest level of cognition, there is the greatest uncertainty of effects and success. Try reinforcing existing beliefs, and avoid changes that create dissonance unless confusion is desired. Knowledge of target expectations is very important, as well as sufficient feedback to measure effects and react to dynamic and changing situations.

6.2.6 A Possible Framework

A problem with designing deceptions is that there is a mismatch between techniques, which focus on what has been seen to work, and objectives, which focus on what should be portrayed to the target. Cohen et al start from the former, with a focus on what is possible to achieve, and suggest building deception programs.

The authors describe the framework as follows:

> In essence, our framework starts with a programming language for human deception by finding a set of primitives and creating a syntax and semantics for applying these primitives to targets. We can then associate metrics with the elements of the programming language and analyze or create deceptions that optimize against those metrics.

The framework consists of three parts: A set of primitive techniques, properties of those techniques, and the syntax and semantics for applying and optimizing the properties. The primitive techniques would be associated with one or more of Observables, Actions, Assessments, Capabilities, Expectations and Intent. The properties would include, but not be limited to, those previously described (in 6.2.1). As an example, Cohen gives a technique, audit suppression, codified according to these properties. An example of a simple script in the (probabilistic) programming language is also presented.

6.2.7 Call for Experiments

Cohen et al is quite clear on the fact that experiments with the use of social science methodologies are lacking from the field of computer deception, and considers it a critical area for further work. Of the experiments that have been done, few have been controlled experiments designed to understand the attack and defense processes. The lack of empirical data makes it hard to reach scientific conclusions.

6.3 Analysis

There is a lot of theory in this paper that we are ill-equipped to judge, for instance the correctness of the human cognitive model. We will look closer at the definition of deception, see if we can organize the sixteen dimensions somewhat, and comment on the computer cognition model.

6.3.1 The Definition of Deception

Let us look closer at the definition of deception:

> Deception is a set of acts that seek to increase the chances that a set of targets will behave in a desired fashion when they would be less likely to behave in that fashion if they knew of those acts.

The above definition is not particularly precise; this is stated and acknowledged by the authors. If we look at other definitions we will find that they are either narrow and precise, or, as in this definition, wide-ranging and broad. It seems to be a constant problem, unlikely to go away anytime soon.

By reflecting on the given definition quite a few points can be extracted. (1) There is a deceiver acting with intent (2) it is the intent that defines the actions as deceptive actions (3) there is a set of targets having awareness (4) which the deceiver seeks certain actions from (5) the targets would act different if they knew of the deceptive acts (presumably identifying them as such).

The definition excludes quite a few deceptions: (3) non-aware entities (4) deceptions resulting in changed states of mind, but not of actions (5) deceptions having the same effects even when the targets know of them.

Let us consider (3) non-aware entities further. This excludes computers as targets since they lack awareness, and are therefore merely intermediate stepping stones for human deception. However, we have seen that computers have cognitive structures, can have built-in expectations, and sometimes seek goals. This resulted in the possibility of deceiving computers. In some way a definition should be expanded to cover these circumstances.

6.3.2 The Nature of Deception

The dimensions or properties stated by Cohen et al range a wide spectrum of areas. Instead of delving deeply into the aspects of each, we have tried to group them according to their main subject.

(1) Fundamental deception properties (2) All deception is a composition of concealments and simulations.

(2) Target properties (1) Limited resources lead to controlled focus of attention (3) Memory and cognitive structure force uncertainty, predictability, and novelty (5) Observables limit deception (6) Operational security is a requirement [2] (7) Cybernetics and system resource limitations (9) Large systems are affected by small changes (10) Even simple deceptions are quite complex (11) Simple deceptions are combined to form complex deceptions (12) Knowledge of the target (14) Modeling problems (16) Counterdeception.

(3) Target and deceiver properties (4) Time, timing, and sequence are critical, (8) The recursive nature of deception.

(4) Execution properties (15) Unintended consequences.

(5) Societal constraints (13) Legality.

While this brief attempt at grouping the dimensions in no way should be taken for an attempt at a taxonomy, there is a strong indication of the target's characteristics being the reason for many complications when deceiving.

6.3.3 Computer Cognition

The computer cognitive structure mirrors the same principles as the human cognitive structure, in that it is layer based and one can perform the two fundamental operations (concealment and simulation) on each layer, and consequently achieve induced or inhibited responses.

What the model does not explain is what the effect of this cognitive structure is on the human structure, ie how they are connected. There is no overarching architecture in which one can track the flow of information or effects.

There is also the question, do the given layers properly capture all of the computer aspects? Where does middleware object fit in, for instance? One can always use the existing layers to define objects and entities crossing several layers, but there is no proof of these building blocks being the most suitable.

[2] Due to altered target actions if the deception is known.

6.4 An Interlude

Fifth Act: Cohen

"What are we waiting for again?" asks Frank after a long period of calm.

"For the storm, Frankie, for the storm."

"Sorry?"

Bubba sighs. "Frankie, Frankie. You are not going to get any medals if you continue like this. But since I like you, I'll do a recap." He rises. "In the recent times, the number of attacks against your computer system has gone up. As a response, we have decided to implement deceptive measures. Yes?"

"That's the Honeynet."

"Exactly. We've combined production and research honeypots as well as some nifty hardware to implement an attacker rerouter mechanism. Every attacker we detect is shuffled into the Honeynet. As per now, we have three attackers roaming about, that is, Alpha, Bravo, and Charlie. Our last event was Alpha downloading a false customer file which we had prepared."

"And we are now waiting for the attacker to make contact, courtesy of your friends in the phone company."

"Exactly."

"But I don't get how a false customer file will lead the attacker to doing that. This is cognitive deceptions, working at the highest level. There is no causality inducing the attacker to make contact."

"Oooh, now we're getting the lingo treatment. Nice. But you have to turn the argument around. We are not trying to get all attackers that download the file to make contact, we are trying to get the attackers that have ulterior motives beyond the basic look-around. There's a limit to how many things an attacker can do with a customer database. We're hoping that making contact will be one of them, and..." Bubba breaks off, and looks at his laptop. "Whatnow?"

Frank comes closer. "What is happening?"

"Attacker Bravo just saved a file called 'read.me' in root. Strange. Let's check it out." Bubba opens the file in an editor.

"2: T3h suX0rz 4dMinz. 0 tr4ffiX => f4k3n3zz!!!1111"

"Why I never.. and now there's another file." Bubba opens this next.

"And if this wasn't a set-up, would someone find a new file in root after five seconds? You are not the only one who can bait. B out."

Frank raises a hand. "Er.. Question. How come Bravo is calling himself B?"

"The plot thickens, Frankie. Thickens." Bubba's cell phone starts to vibrate, and Bubba answers distractedly, "Eych. What? Contact?" He listens. "That's good. Can you take it further? Brilliant." He hangs up. "One of our fake customers just got contacted by Sam's Snazzy Foodstuff, who was offering the exact same orders as in the false database at 10 percent discount. Someone is homing in on your business, Schultz."

"Well, I'll be."

"Precisely."

Chapter 7

Warfare

The books and papers on deception in warfare cover a small part of the works on warfare in general. This is relative; for the purpose of this master thesis there are far too many works to tackle them all, or a majority.

We have chosen two works for further study: "The Art of Deception in Warfare" by Michael Dewar (Dewar 1989), and "Joint Publication 3-58: Joint Doctrine for Military Deception", published under the direction of the Chairman of the Joint Chiefs of Staff (Chairman Joint Chiefs of Staff 1996). Together they show a mixture of examples, doctrine, and the theory of the military branch.

The chapter starts with a synopsis of our analysis, before presenting the theory and the analysis itself.

7.1 Paradigm Synopsis

Computer deception is not treated explicitly in Dewar or JP 3-58: The focus is on human deception.

Dewar's is mainly a historical work, but with an emphasis on deception theory. The theoretical foundation is largely built on the theory of Barton Whaley[1]. A set of human tendencies that can be exploited for deception purposes, as well as a set of techniques, are given. We grouped the techniques by the different aspects they highlight: (1) Detection avoidance: Concealment and camouflage (2) information release: The piece of bad luck, the unintentional mistake (3) what to show and accomplish: Reinforcement of probable action, the ruse, the double bluff, the repetitive process, the lure, the substitution.

[1] We describe Whaley's theory in detail in chapter 5.

JP 3-58 is primarily concerned with giving guidance for planning and execution. The planning and execution process of JP 3-58 is quite structured and detailed: The most important steps are (1) receive the externally defined strategic objective (2) decide on a deception objective in terms of target actions that support the strategic objective (3) gather extensive information about the adversary and target to guide and understand the possible effects of deception (4) develop a full deception plan. The development of the deception plan places demands on knowledge of the workings of both friendly and enemy forces. We will return to this process in the discussion in the next chapter.

Both works also presents deception principles which mostly, but not completely, overlap. The main concern of the principles is the deception target.

7.2 The Theory of Dewar: "Art of Deception in Warfare"

Michael Dewar covers a lot of ground with both theory and examples of deception in warfare. A few examples are historical and from preindustrial times, but the focus is mainly on the World Wars and later when technology began having serious effects on warfare. Dewar discusses electronic deception, psychological operations and propaganda, and deception used in counter revolutionary warfare.

The deception theory of Dewar covers different areas. Much of his theory is influenced by Barton Whaley and his attempt at a general theory (Whaley 1982), so we can focus on what Dewar adds to the subject. We will go through a set of human tendencies that opens up the possibility of deception, principles of planning and execution, a set of deception techniques, and finally the relationship between security and deception as Dewar sees it.

7.2.1 Human Tendencies

Dewar begins with describing "several tendencies [identified by psychologists] which make the human mind peculiarly susceptible to being deceived." (Dewar 1989, page 9)

Association of wisdom with seniority or age This is illustrated by the (unwritten?) doctrine that more weight is given to the opinion of a higher ranking officer than a lower ranking officer, something that is problematic when novel questions need minds unaffected by long routine[2].

Preference of evidence which support our point of view We like to be right and prefer evidence which confirms our preconceived ideas; conversely, lower priority is attached to ideas which oppose our existing notions. This is strengthened by subordinates telling their superiors what they want to hear.

Jumping to conclusions We dislike uncertainty and confusing or ambiguous situations, and try to resolve them as quickly as possible. Especially in stressful situations (like war), the temptation is to jump to conclusions.

Limited means of processing powers The mind is limited in how much information it can cope with simultaneously. It must prioritize its use of resources and filter information.

Focus of attention We have a tendency to focus on new and interesting things while ignoring the old and known; this means the mind has a focus which can be directed.

Regularity and routine lulls the mind Repetitive events are given less priority than rare events, gradual changes are harder to spot than big differences.

The result of being taken in by these tendencies is often surprise, and Dewar describes further the link between surprise and deception:

> There is only one way of achieving surprise, and that is by concealing one's intention. There is only one way of concealing one's intention, and that is by some form of deception.

(Dewar 1989, page 14)

7.2.2 The Deception Techniques

In order to achieve surprise or an effect on the adversary, some sort of deception technique must be employed. Dewar is quite clear on the what the aim of deception is: "Deception aims to mislead the enemy into a predictable course of action or inaction which can be exploited." (Dewar 1989, page 15). Lack of information or false information which leads to confusion might end in unpredictable reactions; in Dewar's opinion this is not proper deception.

[2] Dewar quotes Winston Churchill in "The World Crisis" on this issue.

Dewar presents eight deception techniques:

Reinforcement of probable action Make the target believe the most probable plan will be used, while in reality use an alternative plan.

The Lure Present the target with an opportunity, which is a trap.

The Repetitive Process Lull the target into a false sense of security, by showing the same over and over.

The Double Bluff Reveal the truth when falsehood is expected.

The Unintentional Mistake Let the target acquire information, seemingly through a breach of security or by negligence.

The Piece of Bad Luck Let the target acquire information, seemingly through circumstances which his adversary had no control over.

The Substitution When the target has identified something as false, substitute it (covertly) with the real, and vice versa.

The Ruse Disguise one's own forces as the enemy.

7.2.3 The Principles of Deception

If you have decided on the technique you want to employ, how should it be planned and executed? Dewar points out six principles that should be considered during any deception operation.

(1) Centralized control and coordination Without coordination friendly troops are as likely to be deceived by the deception as the enemy.

(2) Sound and thorough preparation Knowledge of the target and its procedures is necessary, including calculation of the target's reaction to each phase of the deception.

(3) The deception must not be incongruous or illogical The deception should accord with patterns of events which the enemy has reason to expect. When this is not possible, false information should let the enemy derive the desired conclusions by use of his own intelligence apparatus.

(4) Present false indicators through as many sources as possible Be careful to avoid giving so much information that the target becomes suspicious. In case of failure, be prepared to modify or abandon the deception plan without revealing the original intent.

(5) Timing is crucial The target must have time to react to the false information, but not enough time to understand that a deception is taking place.

(6) Operational security Absence of normal security precautions must not arouse suspicion, and knowledge of the deception must not be widespread (note that friends also can be unwittingly deceived).

7.2.4 The Relationship between Security and Deception

An interesting comparison is done by Dewar regarding the relationship between security and deception. Security is the implementation of negative measures, where you deny information to the enemy:

- Where you are and/or where he is.
- What weapons and forces you have at your disposal.
- What you intend to do.
- Where you intend to do it.
- When you intend to do it.
- How you intend to do it.
- Your knowledge of the enemy's intentions and techniques.
- How successful his operations are.

In short, the *location* (of you or the enemy), your *intentions* (what, where, when, how) your *possible means/forces*, your *knowledge* of the enemy and *feedback* of his own operations.

Each element has a positive counterpart in deception, where you try to convince the enemy:

- You are somewhere else and/or he is somewhere else.
- Your weapons and forces are different from what they are.
- You intend to do something else.
- You intend to do it elsewhere.
- You intend to do it at a different time.
- You intend to do it in a different manner.

- Your knowledge of the enemy is greater/less than it actually is.
- His operations are more/less successful than they actually are.

Security, then, is preventing the enemy from seeing indicators, while deception is providing alternative indicators.

7.3 Analysis of Dewar

We will comment on most of the areas covered by Dewar: The human tendencies, the deception techniques, the deception principles and the connection between deception and security. Last we see how computer deception fits into this paradigm.

7.3.1 Human Tendencies

The tendencies Dewar describes fit easily within Cohen's human cognitive structure. Some tendencies are applicable to several levels in the three-level structure, for instance focus of attention, which exists both at a conscious and unconscious level. Limited means of processing powers is a property of the whole mind, while the rest seem to be effects of the highest level.

Deception is acknowledged as the method for achieving surprise by concealing one's intention, but is surprise always a part of the desired result? It is the result when the adversary sees through the deception and realizes he's been had, or when he ends up in an unanticipated situation. We can see this happening tactically, when a concealed force moves out from an unexpected direction. But some deceptions may never be revealed, or the objectives of the supported mission might be reached before the deception is unmasked. Surprise does not seem to be a temporary objective or the final result in these cases.

7.3.2 Deception Techniques

Dewar considers only misleading as deception since it (hopefully) leads to predictable courses of actions. Confusion leads to unpredictable actions, and is not proper deception. This view, however, is not ubiquitous within the military establishment. For instance, Field Manual 90-2, "Battlefield Deception" (Department of Army 1988) acknowledges the distinction by classifying deceptions into A-deceptions (ambiguity deceptions) and M-deceptions (misdirection deceptions).

Dewar presents the deceptions techniques as a simple list, but we will try to organize them into groups.

(1) Detection avoidance Concealment and camouflage[3].

(2) Information release The piece of bad luck, the unintentional mistake.

(3) What to show and accomplish Reinforcement of probable action, the ruse, the double bluff, the repetitive process, the lure, the substitution.

Some of these techniques are directly connected to a human effect they exploit; from a theoretical standpoint it could be useful to see if more techniques could be taken into account by exploring such relationships. This is, however, not within the bounds of this thesis.

7.3.3 Deception Principles

The six principles are a mixture of planning and execution principles. Most are in some ways connected to the target and its characteristics; this becomes apparent if we restate them:

Restated Principles (1) unified planning and execution to avoid friendly confusion (2) know the target (3) have high enough quality to convince the target (4) use the sources the target reads in sufficient quantity (5) time for deception to enter, be processed, and acted on by the target (6) mask indicators revealing the deception to the target.

Alone stands the first principle, centralized control and coordination, in order to avoid self-deception and friendly force deception. The rest is concerned with the target.

That most of the principles should be concerned with the target is not so surprising, considering the complexity of the human mind. As we remember, Cohen's sixteen dimensions were also skewed heavily in this direction.

7.3.4 Deception and Security

Since we are mainly concerned with deception for furthering computer security, it is of interest when Dewar places deception and security so close together. In essence, his definition of security was hiding information from the enemy, while deception was providing alternative information.

[3] Not listed under techniques, but heavily described and demonstrated in other parts.

We are lying, then, about what *is*. How shall we choose what to lie about? We know that deception is a supportive tactic, with value only in its support of an external strategic objective. It would seem reasonable that we must know the strategic objective in order to select what to show.

7.3.5 Computer Deception

Since most of Dewar's theoretical foundation is the theory of Barton Whaley, it is not surprising that computers are not mentioned. Electronic warfare, radars and surveillance are covered, not computers.

Mostly, we have gained more high-level characteristics of the human mind, and techniques that exploit them in various ways. We merely state, as we did with Whaley, that a computer can be used as an aid in deception of human beings. We have the same problem of not knowing how to use it.

7.4 The Theory of Joint Publication 3-58

Joint Publication 3-58 is a military doctrine document written for a military audience. As such, deception is sometimes seen in relation to more specialized concepts like Command and Control Warfare (C2W) and the C2W tools (deception, psychological operations, electronic warfare, operations security and physical destruction). However, deception is employed in a conflict against an adversary, just as we are in conflict with computer attackers.

We will look at the definition of deception before we focus on the planning and execution process, which is given in detail. A set of principles is also covered.

7.4.1 Definition and Categorization

In the Department of Defense Dictionary of Military and Associated Terms (J-7 2001), we find the definitions and categorizations of deception as used in Joint Publication 3-58.

> **Deception:** (DOD) Those measures designed to mislead the enemy by manipulation, distortion, or falsification of evidence to induce the enemy to react in a manner prejudicial to the enemy's interests.

(J-7 2001)

> **Military deception:** Actions executed to deliberately mislead adversary military decision makers as to friendly military capabilities, intentions, and operations, thereby causing the adversary to take specific actions (or inactions) that will contribute to the accomplishment of the friendly mission. [...]

(J-7 2001)

The definitions are clearly affected by the military context, with its focus on adversary decision makers and desired (in)actions. A categorization of military deceptions into five is also given: (1) strategic military deception (2) operational military deception (3) tactical military deception (4) service military deception (5) military deception in support of operational security.

The categorization is divided by looking at *by whom* the deception was employed and *why* (strategically speaking). As a means for characterizing differences among deceptions it is less useful.

7.4.2 The Planning and Execution Process

The main planning process is a six-phased process, with several of the phases further broken down into subphases. While this implies linearity, where we simply proceed from stage to stage, this is an oversimplification, and a dangerous one if followed blindly. The process is in truth much more an iterative process than a linear one, and requires continuous monitoring of objectives, targets, and means. A situation can be highly dynamic and one must be prepared to respond to unforeseen changes or consequences.

Singled out for added emphasis is risk. What are the possible benefits of the deception, and do they outweigh the costs of failure or detection? Basing the success of an operation on the success of a deception involves a high level of risk.

The reasons for possible deception failures are many. The deception may fail to reach the target, be doubted, the target may hesitate, be unable to act or act differently than expected. Means or feedback channels may be compromised and unintended effects on or by other actors may have unforeseen consequences.

7.4.2.1 The Main Planning Process

All of the six phases have explicit demands that must be met in relation to other military processes and doctrines; we will ignore these and focus on the parts that are relevant in a general situation.

Deception Mission Analysis How can deception support *the mission*, an objective defined outside the deception operation. Information about the mission and the operational area is studied.

Deception Planning Guidance Guidance from the commander to the planning staff; the deception goal is stated.

Staff Deception Estimate Gathering and analysis of information relating to the adversary; development of several deception course of actions (COAs)[4].

Commander's Deception Estimate Selection of an operational COA and supporting deception COA.

Deception Plan Development A five-step process for developing the selected deception in detail.

Deception Plan Review and Approval The commander reviews and approves.

With regard to processes the focus has hitherto been on the development of a specific deception[5]. The Joint Publication places this planning within a larger process which ultimately supports the operational objective. This makes perfect sense, since a successful deception that does not support the strategic objective is useless.

7.4.2.2 Need for Information

The need for information is great throughout the process. Phase three (Staff Deception Estimate) specifies the information needed about the adversary before the detailed deception plan is developed:

- Profiling of key decision makers.
- The C2 system and the decision making process.
- The intelligence collection and analysis capabilities.
- Preconceptions the adversary may have about friendly intelligence and capabilities.

[4] 1. Any sequence of activities that an individual or unit may follow. 2. A possible plan open to an individual or commander that would accomplish, or is related to the accomplishment of the mission. 3. The scheme adopted to accomplish a job or mission. 4. A line of conduct in an engagement. 5. A product of the Joint Operation Planning and Execution System concept development phase. Also called COA. (J-7 2001)

[5] See, for instance, Whaley's ten step list in section 5.2.4.

- Identification of possible adversary courses of action and the probability of usage if possible.
- Estimates of likely reactions to the deception.
- Explanation of how the adversary processes, filters and uses information.

Using this information, several deception COAs are developed. Such a COA will contain the deception objective, the target, the desired perception, outline of a deception story[6] and the possible means to use. It is not until each deception COA has been analyzed for feasibility, impact on operations and security, and the results handed over to the commander (phase four), that the deception COA is selected.

7.4.2.3 Completing the Deception Plan

Development of the complete deception plan (phase 5) is divided into five major actions: (1) Complete the deception story (2) identify the means (3) develop the event schedule (4) identify feedback channels (5) develop the termination concept. We will describe each of these in greater detail.

Action 1: Complete the Deception Story

The deception story must identify all the actions the adversary intelligence system would expect to see if the deception was real. The quality the story must uphold is closely connected to issues of timing. Five specific areas are pointed out:

Time of Maximum Disadvantage When is the adversary's (in)action required? How much time is available for our planning and execution?

The Deception Target The psychological nature of the target affects the portrayal of the story. How much time does the target use for decision making? Is the target bold, requiring little evidence before taking action, or cautious, seeking reconfirmation through multiple sources?

Opposing Force Execution After the desired adversary decision has been made, time is needed for issuing orders. The order must propagate to its executors and the execution itself takes time.

Intelligence Processing The adversary's intelligence system uses time to detect, collect, analyze and inform the deception target.

[6] A scenario that outlines the friendly actions that will be portrayed to cause the deception target to adopt the desired perception. (J-7 2001)

Execution of Deception Tasks When and for how long should deception actions be observable by the adversary?

Action 2: Identify the Deception Means

The deception story must somehow be presented to the adversary. This means that knowledge of friendly force operations and the adversary intelligence system is essential.

Determine adversary's detection and collection capabilities We must know what the adversary can see, and if possible, what the deception target relies most heavily upon.

Identify indicators When we know what the adversary can see, we must find out which indicators would reflect the activity portrayed by the story, if it was real. This requires detailed knowledge of friendly operations and forces, including operational patterns of normal activities.

Compare capabilities to indicators The intersection of indicators available to the adversary and indicators that would be changed by the story being real, must be manipulated to be in concordance with the story. Indicators which would be changed by the deception, but is unseen by the adversary, can remain unchanged.

Select means Based on the previous steps, select the means to portray the story. Operational Security (OPSEC) and deception both adjust the indicators seen by the adversary and must work closely together. Coordination with the other C2W tools is also necessary since they may target the adversary's intelligence system.

Action 3: Develop the Deception Event Schedule

As part of the deception plan, event schedules for the execution are developed. These schedules identify the *what, when, where, who*. Timing and sequences are of high importance.

Action 4: Identify the Deception Feedback Channels

There are two types of feedback: *Operational* feedback, which identifies what information is reaching the target, and *analytical* feedback, which tells what actions the target is taking due to the received information. Both of these feedback types are generally acquired by observing the adversary. His intelligence efforts may give information about the former, and changes (or no changes) is his dispositions or actions about the latter.

Action 5: Develop the Termination Concept

Information about the deception and its existence must be released in a controlled manner. Often the existence itself should be kept secret. Plans for unforeseen compromise and early termination should be included.

7.4.3 The Six Principles of Military Deception

These principles are to provide guidance during deception planning and execution, just as Dewar also intended. Indeed, we see a lot of the same.

Focus The target of the deception must be the adversary deception maker that can take the desired action(s).

Objective The objective is to get the target to take or not take specific actions, as opposed to having a specific belief.

Centralized Control In order to avoid confusion, a single element must control the deception. Note, however, that the execution may be decentralized as long as a single plan is followed.

Security Knowledge of the deception plan and its existence must be kept secret through OPSEC security and a need-to-know basis.

Timeliness Several events must be timed carefully and be given sufficient time: The deception must be portrayed over time, the adversary's intelligence system must have time to collect, analyze and react to the deception, the adversary decision maker (the target) must be given time to react, and the friendly intelligence system must have time to detect the adversary's reaction.

Integration The deception must be fully integrated with the operation it is supporting. The deception planning should occur simultaneously with operation planning.

7.5 Analysis of JP 3-58

The Joint Publication does not bring to light any new inherent properties of deception. Considering the intention of doctrine, this is not so strange: It is not a forum for academic speculations, but a resource which should be useful and applicable in real life when the situation so demands. Both the principles and the given process are exclusively concerned about planning and execution.

There is not much to say about the planning process per se, as we do not have any especial expertise in this area. We will try drawing parallels to computer deception in our later discussion. But we have gotten a new set of principles that look a lot like another set of principles we have seen, namely those of Dewar. Let us restate them in the same manner:

Original (1) Focus (2) Objective (3) Centralized Control (4) Security (5) Timeliness (6) Integration

Restated (1) know whom to target (2) seek target actions (3) unified planning and execution to avoid friendly confusion (4) mask indicators revealing the deception (5) time for deception to enter, be processed, and acted on, and time for the effect to support the strategic objective (6) integration.

It is interesting that the lists of Dewar and Joint Publication 3-58, while very much alike, are not completely equal. Perhaps this arbitrariness exists because the 'principles' are what one think are the most important characteristics of deception, and various factors influence this opinion. There are no obvious inherent differences between the principles and other deception characteristics.

7.6 An Interlude

Sixth Act: Warfare

"I want to know about feedback. And indicators." Frank is reading JP 3-58. "What are we getting?"

"Well, we are not getting that much feedback per se... Are you thinking of anything specific?"

"Joint doctrine calls for operational and analytical feedback."

"Uhm. We saw that Alpha found the file, and downloaded it as a result. The contact can be taken as a proof of deception success."

"Of the deception objective success, yes. He took the bait. But not the strategic objective, since we wanted to know more about the attacker."

Bubba disagrees. "But we do know more. Its the Snazzy people, homing in on your business. Motive and objective."

"How can we know that its not a counterdeception scheme?"

"Oh, come off it. Who'd want to do anything like that? That's just not probable. Anyhow, Bravo was a sneaky dude, and obviously understood the game."

Frank nods. "We forgot a set of indicators, traffic. But how could we have realized that Bravo had detected the deception, if he hadn't given himself away? He could just have bounced around, and we wouldn't have been any wiser."

Bubba reluctantly agrees. "True, true. Seems like we have to think of something better for the smart ones. And then there's Charlie, who has done nothing." He checks his logs. "No, wait, he uploaded some files a couple of hours ago."

Suddenly, Frank's phone rings, and Frank picks it up. "Hello? Yes, that's me. What?" he listens with a puzzled frown. "I think you'll have to talk to the computer specialist" he hands the phone over to Bubba. "There's something about spamming. They're threatening to shut down my Internet account."

"Have no fear." Into the phone, "This is Bilibus (Bubba to friends) H." He narrows his eyebrows. "Really. Uhuh. Well, preferably not. We're running a small experiment." His begin to roll his eyes. "Where's your sense of adventure, man. No, I.." A loud click signals the end of the conversation. "I'll say, they cannot call me Bubba."

"I didn't get what they talked about."

"Well, apparently Charlie is using our Honeynet as a proxy relay for sending spam, so the ISP is blocking our account. In fact, we should lose the link right about now."

Bubba and Frank both turn to look at the ADSL-modem, which loses its link LED after a few seconds.

From a room further into the back of the house a voice sounds, "Doh!"

Part III

Theory

Chapter 8

Discussion

Throughout the deception paradigms, we have seen that there are many ways of approaching the subject of deception, and as a consequence, computer deception. The aspects that have been covered range from techniques, to principles, to planning and execution processes, to the human mind, to the nature and structure of deception itself.

Our thesis objective was to build a descriptive theory that could help us understand the concept of computer deception, both to aid us when designing new deceptions and for analyzing existing deceptions. For such a theory to be complete, we must touch upon all of the above aspects[1]. As we have seen, the various views differ in their take on computer deception, which largely stems from difference of opinions of what deception is. If we want to get anywhere, we must first agree on a definition of deception.

This chapter begins with a short summary of the five paradigms we have been through. Then we will find our own definition of deception, before we look closer at the problem of defining computer deception. We continue with looking at strategic objectives, indicator properties derived from the techniques, principles, and some lessons from the planning and execution process.

8.1 Paradigm Summaries

We have looked at five different deception paradigms. The first, honeypots, clearly distinguished between deception and computer deception. The latter was mainly the masking of mechanisms, and basic attempts at emulating services and the characteristics of hosts. Some measures were

[1] It is limited, however, how deeply one can go in a master thesis, and some aspects will be more fleeting than others.

considered deceptive when specific service or protocol characteristics were exploited to achieve effects on a communicating partner. Human deception, on the other hand, was considered to be a psychological weapon, and something to be avoided. This view, however, was not ubiquitous, and unconventional honeypots explicitly tried to fool human beings.

The second, Dunnigan and Nofi, gave a categorization of deception techniques that were applied to the computer domain, but there was internal inconsistencies in the categorization and differences in the application. The resulting computer deception techniques were neither obvious nor commonly agreed upon.

The third, Whaley, attempted to explain deception purely from the field of psychology. Basically, deceptions were effects shown to a human being, as part of a larger process in which a strategic objective was to be accomplished. There were some inconsistencies in the theory, resulting from the lack of separation between conscious and unconscious deceptions. Computer deception was not mentioned, and probably had no place besides being an aid for deceiving humans.

The fourth, Cohen, in some ways continued the notion of deception as psychology, but extended it to all of cognition. Instead of being effects shown to a human being, deceptions were induced effects on cognitive structures. By defining the cognitive structure of computers, the same basic approach could explain both human deception and computer deception. Still, the intent and objectives of a computer was directly connected to the human operator or designer.

The fifth paradigm was the combined military theory of Dewar and JP 3-58. The former was largely based on the theory of Barton Whaley, but gave additional principles and techniques. The latter did not say much on the nature of deception, but contained an extensive planning and execution process, as well as six principles. Nothing explicit on computer deception was presented.

8.2 Definition of Deception

As we have worked through the five paradigms, we have been exposed to different ideas of what deception is. The honeypot paradigm says little except a 'psychological weapon', which does not help. The military paradigm does not bring to light any new theory on the definition of deception which we do not have elsewhere. We are then left with Dunnigan and Nofi, Whaley, and Cohen, who can say something about the nature of deception. However, Dunnigan and Nofi is clearly constrained, in that all deceptions are reduced to 9 categories, which we have seen are neither taxonomic nor consistent. Our best bet for a definition lies in the direction of Whaley and Cohen.

Whaley thinks of deception as falsehood, where there is a mismatch between what a deception target believes to be true, and what really is true. If we intentionally seek to create this mismatch in the mind of another, we are deceiving.

But we have also seen examples of deceptions that are not covered by a definition hinging on beliefs and truths. Visual and audiovisual deceptions can be effective even if we know of them, as well as other reflexes that belong to the lowest layer of the human cognitive structure. Deception of animals, computers or organizations does not fit easily into a deception framework requiring consciousness or beliefs.

A solution to this problem was presented by Cohen, where a computer was considered deceivable by giving it a cognitive structure. By saying that deceptions go against the intent of the operator or designer, computer deception became possible. If we build further on this notion we can say that the presence of a cognitive structure, and some idea of correct operation, is the requirement for deceivability. By seeking effects on cognitive structures, contrary to normal operation or intent, we are deceiving.

Unfortunately, Cohen did not give us any precise definition of deception, beyond a belief versus truth assertion. We have already stated that this assertion can be taken to be incompatible with deception of non-aware entities, and of deceptions that do not rely on beliefs.

To get a step further, we continue with the notion of cognitive structures. The computer was deceivable largely in part because it had one. That leads us to the question, who is handing out these cognitive structures? What is the requirement for having one? Can we deceive, say, a toaster? What about computer worms or viruses?

We believe the crucial distinction is *information processing*. If an entity, software or hardware, mechanical or biological, processes information, it stands to reason that it must have input, possibly an internal state, and possibly some sort of output. If we interpret this to be a cognitive structure, and that it per definition has a correct or normal operational pattern, we have grounds for saying the entity is deceivable.

We embody these ideas in the following definitions that we will use in the latter part of this thesis:

(1) All entities that process information have a cognitive structure By this we mean (1) anything that process information is an entity (2) the entity's cognitive structure is defined by how it processes information. This means that humans, animals, technological hardware and software all have cognitive structures.

(2) Some entities have internal states Again, humans, animals, technological hardware and software have states. The exception would be something that does not analyze nor react to input in any way.

(3) Deception is the exploitation of a cognitive structure by the inducement or inhibition of state change, contrary to the intent of the design, operation, or conscious will of the entity This definition seeks to encompass:

(1) Non-aware entities, logical or physical, that process information. Deception of these entities is against the intended design or operation.

(2) Aware entities without full high-level consciousness like animals. Deceptions are against the intent of the entity, or exploitations of the cognitive structure outside full entity control.

(3) Full conscious entities like human beings. Deceptions are against the intent of the entity, or exploitations of the cognitive structure outside full entity control.

(4) A cognitive structure that might change state as a result of processing information, is deceivable This is merely a corollary of the above.

Before we start running around and using this definition, let us be clear on what we know of cognitive structures.

In the section on the nature of deception, Cohen et al stated sixteen dimensions covering several aspects of deceptions. Many of those were concerned with the target and its properties. The most general properties we retain for our template entities: That they have limited resources, limited abilities to process available data, and some predictability as a consequence of the inflexibility of the cognitive structure. Certain types of entities may possess memory, which makes them able to learn from previous deceptions.

The fundamental operations on cognitive structures are simulation and dissimulation, which affects the information that is processed by the entity. As a simple model of effects that can be accomplished on any cognitive structure, we use the grouping we created of Whaley's effects: Hiding, confusion, misidentification, and attention attraction or diversion[2]. In more complex entities, these basic effects can be combined to achieve more advanced

[2] Note that we do not think of them as effects *shown*, but effects *achieved on* a cognitive structure.

effects. A human being is the prime example of this, where Dewar's human tendencies can be regarded as advanced effects made possible by the use of the basic four effects.

8.3 Computer Deception?

It is very tempting to use our newfound definition in a subsequent definition of computer deception as "deception of computer entities", or something equivalent. This would, of course, fit neatly with the honeypot paradigm: 'Deception' works against humans, and should be avoided, and 'computer deception' works against computer entities and their normal operation, which is what we are all about. From this point of view, computer deception would be deception *of* computers.

But this is not the only possible definition. More in line with Whaley and Cohen, is the view that deception of computer entities is only an intermediary step to deceiving the human beings using the computer. This would define computer deception as "deception *with* computers".

We can also conceive of a middle course, as a combination of the previous two. There we would be deceiving computer entities, but also anticipating a set of guaranteed or highly probable effects on the human attackers. This view would only be valid if we found a chain showing causality from effect X on the computer entity to an effect Y on the human attacker.

Clearly the choice one makes here has a lot to say about means, methods, and possible objectives of deception. Are we dealing with entities of program code, or are we deceiving human beings? In our eyes, to make an arbitrary choice here would be meaningless. In the next chapter we will build a model that can shed some light on this issue. Until then, we will continue with our investigation into the other aspects of deception, which in most instances are valid irrespective of where the border between deception and computer deception lies.

8.4 The Reason for Deception

Per our definition, deception is the inducement or inhibition of an effect on a cognitive structure. However, we included the word "exploitation". The reason for this was to imply something more than just a state change: We are hoping the state change will accomplish something. This *something* is the strategic objective, and in principle it is not related to the deception at all. Deception is a means, and the strategic objective the goal. This is what makes deception a supportive tactic rather than an objective in its own right.

Unfortunately, strategic objectives are neither clear cut nor obvious. For example, take a computer network defense objective like "keep attackers out". Why do we want this? One answer can be that we want to increase security. Why? To make authorized users able to perform their tasks. Why? Because they are employees in an organization, and it is the work they do. Why? Because they want to support their livelihoods. But this is not what the shareholders want; they want maximum profit.

The point here is that strategic objectives can be seen as existing in chains, where the lower objective support the higher. A deception supporting a lower strategic objective is often worthless if the higher remains unsupported. Human beings, however, rarely have explicit chains. The ultimate top would be the meaning of life, which is either non-existent, unknown, self-defined or 42, take your pick. A chain is in reality an entanglement of conflicting desires, and different persons do not see situations from the same point of view, resulting in divergent strategic objective chains. When stating the strategic objective of a deception, thoughts should be given to this issue. Is the objective really stating what we want it to say, and does it imply what we believe it to imply?

If we start with a clearly defined strategic objective, like "keep attackers out of the computer system[3]", we find another problem. This objective is so coarse, that there are no obvious deception objectives supporting it directly. Rather, one must derive a lower strategic objective, and support that one. An example would be "hide key resources" or "divert attackers to false systems." But in doing this, one specializes, and the mere act of specialization removes opportunities.

This fact means we cannot be exhaustive in deriving lower strategic objectives. Consequently, we will not be able to exhaustively enumerate all possible deceptions that support a strategic objective. This is unrelated to actual implementations or situation contexts. If you find yourself in a situation where you see no possibility for deception to support the strategic objective, it might be because you are not deriving the correct subobjective.

8.5 The Assumption Chain

When we have been given a strategic objective, we begin thinking of how deception can support that objective. Usually it is by having a deception target taking an action or inaction. All deceptions end up with there being

[3] For the moment, ignoring the issue of where the borders between inside and outside are.

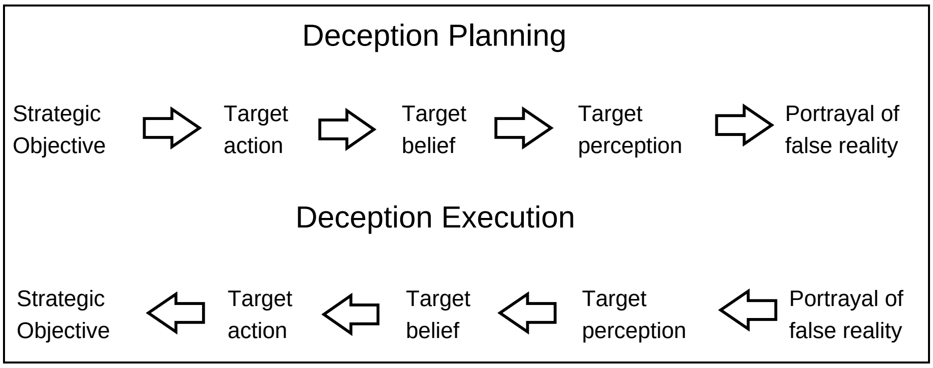

Figure 8.1: *The Assumption Chain.*

a set of actions that we execute in order to show something to the target, which we expect will make the target change behavior, which in turn support our strategic objective. What we show is the *portrayal*.

When we consider our actions from portrayal to the desired strategic objective, we get a chain of assumptions, or, as we have dubbed it, the Assumption Chain. Basically, we assume (1) we portray the deception as intended (2) the target perceives it (3) the target acquire a certain belief (4) the target takes an action (5) this supports our strategic objective. See figure 8.1.

It is crucial to be aware of the types of assumptions we make to get from one step to another, including both those external and internal to the target. If our portrayal does not reach the target as intended, or the desired target actions do not support our strategic objective, we fail. If the target perceives as intended, but get other beliefs or acts differently than expected, we also fail.

8.6 The Deception Techniques

The deception techniques we have seen, ie those of Dewar and those of Dunnigan and Nofi, are not bound to any explicit definition of deception. By only considering our definition, we would probably expect deception techniques to be various ways of achieving effects on a target in support of a strategic objective. With this in mind, what do the existing techniques accomplish? Those we saw were the following:

Dunnigan and Nofi (1) concealment (2) camouflage (3) false and planted information (4) ruses (5) displays (6) demonstrations (7) feints (8) lies (9) insight.

Dewar (1) reinforcement of probable action (2) the lure (3) the repetitive process (4) the double bluff (5) the unintentional mistake (6) the piece of bad luck (7) the substitution (8) the ruse.

You might remember that we tried to create some coarse groups into which we could place the techniques:

(1) Detection avoidance How to remain hidden. (Dunnigan and Nofi: Concealment and camouflage. Dewar: Mentioned, but not listed as a technique.)

(2) How to send information How information reaches the target. (Dunnigan and Nofi: false and planted information, lies.)

(3) Methods for disclosing indicators How information *apparently* is made available to the enemy (Dewar: the piece of bad luck, the unintentional mistake.)

(4) What to show and accomplish Catch-all group containing the deception techniques intended to accomplish a specific result (Dunnigan and Nofi: Ruses, displays, demonstrations, and feints. Dewar: Reinforcement of probable action, the ruse, the double bluff, the lure, the substitution).

The one odd man out was Dunnigan and Nofi's insight, which was taken to be the driving force making use of the other classes.

We believe a good way to read the above list is by considering the groups to be an initial attempt at capturing different aspects common to many deception techniques, rather than classes of techniques. Let us take a closer look at each group.

Group 1: Detection avoidance It is tempting to separate the set of deception techniques into two kinds, those that hide the real and those that show the false. But this type of categorization is nearly worthless, since there would be only one technique in the hiding the real (total concealment), and the rest would be in showing the false.

Consider the question closely: When is something completely hidden without showing anything false? This must be if the thing is outside the target's sensors in every respect. But if you begin altering your surroundings to support the impression of absence, ie removing tracks and other traces, you are also showing the false. And then you cannot claim to be totally undetected, because some parts of what you try to hide has caused state change in the target.

What we have, then, is that you often try to hide certain details, but are ultimately showing something false.

Group 2: How to send information If, instead of total concealment, we want to show the target a false reality, we must somehow make that reality reach the target. This is described by Whaley in step nine of his planning process, that the portrayal must travel over a *channel* to reach the target. The two classes of Dunnigan and Nofi (false and planted information, lies) embody this concept, since they speak of how the target acquires the false reality.

But is this not necessary for all deceptions that are intended to be detected by the target? Somehow, the target must be exposed to the deception. This is not just the case if explicit false material is planted somewhere or lies spoken, but it will always be the case.

Let us call the concept of sending a false reality to the target over a channel for *conveyance*.

Group 3: Information release Dewar speaks of two ways that information can be implied to be made available to the enemy, using the cover of bad luck or a mistake resulting in unintended information release.

But again, all deceptions must, in some way, be released. In fact, we can rephrase by saying, "all deceptions must be placed on a channel". The two above deception techniques embodies two possible ways of implying specific information release events to the target, and by using those implied events to affect the target's assessment of the information.

Group 4: What to show and accomplish During our investigation we took the simple route, and lumped the remaining classes into this group. By the above discussions, we have tried to argue for the fact that all deception techniques have properties related to detection or non-detection, conveyance and release. The same argument can be applied here: All deceptions have some sort of content, and all deceptions are meant to accomplish something. Depending on what part of the Assumption Chain you focus on, the accomplishment can be the target belief, action, or the strategic objective.

Indicators

Above we defined the channel, by which we said information or deceptions travel from the deceiver to the target. Let us consider more closely what, exactly, is doing the traveling.

When Whaley gave operational descriptions of how to accomplish one of his six effects, he used his categories / characteristics / pattern theory to manipulate the false reality on different levels. By hiding some categories, he could show other characteristics, which in turn (for example) could make the pattern blend, be misidentified or dazzle.

There is some truth here. We manipulate details in order to show a different whole. Creating false details on an object can make it be misidentified, changing the interpretation of a situation.

In order to capture this concept of levels, we institute *the indicator*. The concept of indicators, as JP 3-58 has shown, is not new. There it is defined as[4] "data derived from open sources or from detectable actions that adversaries can piece together or interpret to reach personal conclusions or official estimates concerning friendly intentions, capabilities, or activities. [...]".

We will define it in a more general way: *An indicator is the smallest unit of meaning.* Indicators can be combined to form aggregated indicators with new meanings, and "information" is the meaning conveyed by indicators. A channel is the conduit by which indicators are visible.

The fundamental operations of simulation and dissimulation, or showing and hiding, works on indicators. By applying them on different aggregation levels one can achieve the effects on cognitive structures that interpret or react to them.

By using the concept of indicators, we can restate the set of aspects that most deception techniques have:

(1) Indicator Masking Detection or non-detection.

(2) Indicator Conveyance How is the indicator transmitted (ie by what channel).

(3) Indicator Release How is it implied that the indicator is placed on the channel.

(4) Indicator Contents What does the indicator show.

(5) Indicator Intention What is the result intended by showing this indicator.

Note that you can apply these aspects to indicators, aggregated indicators or the overall meaning.

[4] In truth, the definition is not given there but in (J-7 2001). It is, however, used in JP 3-58 in this capacity.

8.7 The Deception Principles

As we have seen deception techniques, we have also seen deception principles. The two warfare texts presented each a set of principles, which we restated:

Dewar (1) unified planning and execution to avoid friendly confusion (2) know the target (3) have high enough quality to convice the target (4) use the sources the target reads in sufficient quantity (5) time for deception to enter, be processed, and acted on by the target (6) mask indicators revealing the deception.

JP 3-58 (1) know whom to target (2) seek target actions (3) unified planning and execution to avoid friendly confusion (4) mask indicators revealing the deception (5) time for deception to enter, be processed, and acted on by the target, and time for the effect to support the strategic objective (6) integration.

Let us group them according to the subject matter[5]:

(1) Make sure you target the entity capable of producing the effects and consequences you seek. Is your target the one that can change things to your advantage? (Principle JP 3-58: 1)

(2) Estimate the effect of the deception on friendly forces. Will your own forces or those of your allies become entangled in the deception? (Principles Dewar:1 and JP 3-58: 3)

(3) Plan the deception quality and extent by considering the target it is to affect, and the effect it is to achieve. This means making the deception to be in concordance with how the target processes input from various sources, makes decisions, and acts. (Principles Dewar: 2,3,4,5,6 and JP 3-58: 2, 4, partly 5)

(4) Time is needed for effects outside the target. Remember to estimate the time needed and time available for the entire Assumption Chain to be fulfilled, not just those related to the target. (Principle JP 3-58: partly 5)

As we can see, if we combine all principles concerning the target into one principle, we have four remaining. To put it plainly, (1) find the correct

[5] Note: Principle JP 3-58: 'Integration with friendly forces' can either be placed under (1) to avoid friendly force confusion or (3) to avoid revealing the deception to the target. Original intention is uncertain.

target (2) consider the consequences of the deception on yourself (3) plan according to the target's characteristics (4) time is needed for effects to percolate.

It is tempting to take a look at these principles from the perspective of computer deception. We will attempt this, but keep in mind that we will suffer from some of the same faults that the applications of Dunningan and Nofi's categorization scheme do, in that there is no common agreement on the method of application. There are also other texts that apply sets of warfare principles to computer deception (Rowe and Rothstein 2004, for instance), but we have not investigated this aspect further.

The first principle calls for targeting the correct entity. There are two issues here: (1) Is the entity captured by our deception friend or foe (2) if foe, is the entity the correct entity with respect to the effect we want to achieve. The former is the old problem of recognizing attackers in computer systems, which we do not take upon us to solve here. The latter can be more insidious. In one way, we have no choice in selecting a target - the attacker is the attacker, which we want to deceive. With more complex strategic objectives, however, the person executing the attack might not be the one we want to affect. Maybe we want to induce actions in a decision maker, who uses input from multiple sources for making decisions, including information from the operator who attacks our system.

The second principle asks you to consider the effect of the deception on friendly forces. Again, we have the problem of recognizing attackers. Some deceptions may not have the capability of separating between friends and foes. For instance, if we create network level deceptions, these deceptions would be visible to administrators doing legitimate work. Should the administrators know of the deceptions, contrary to OPSEC principles? Then we run the risk of informing too many people, and we lose the possibility of deceiving insiders.

The third principle says to tailor the deception to the target. This is the crux of the matter; easy to say, very difficult to achieve. Both acquiring enough intelligence to inform your deception, as well as portraying it in a correct manner, is hard.

The fourth principle says that effects need time to flow. There is not much we have control over beyond the portrayal, rather it is important to factor these things into our deliberations when designing deceptions. Especially if we desire specific actions as opposed to inactions must we estimate time for consequences to happen.

8.8 The Planning and Execution Process

We have seen two examples of a planning and execution process: Whaley's ten-step list and the more elaborate doctrine of JP 3-58. Unfortunately, we do no know enough to construct a planning and execution process. It would be slightly premature to construct one before we agree on a definition of computer deception.

We can still say something of a more general character. It should have become clear that we always start with an external strategic objective, and that we seek the fulfillment of an Assumption Chain. We should make the chain explicit, so that we avoid the pitfall of implicit assumptions. Generally, one plans from strategic objective to portrayal, and executes in reverse (Gerwehr and Glenn 2000).

We have already talked about Whaley's planning and execution process in the analysis, and will focus on different aspects of the JP 3-58 process here. While we cannot apply the whole process to computer deception, there are many issues mentioned that are worthy of further attention.

To summarize, the process was as follows: (1) An external strategic objective is given (2) a deception objective in terms of target (in)actions are decided upon (3) an information gathering phase to determine links between target beliefs and target actions is performed, and several tentative deception chains are developed in parallel (4) after selecting an operational COA, a supporting deception is selected (5) the development of a full deception plan.

We will look closer at three areas that especially stand out in relation to computer deception, namely intelligence, indicators, and feedback.

8.8.1 Intelligence

After deciding on the target of the deception, JP 3-58 called for an extensive information gathering phase. This phase was used to acquire large amounts of information regarding the adversary, including but not limited to information for the profiling of key decision makers, the C2 system and the decision making system, the intelligence apparatus, adversary preconceptions of friendly capabilities, possible courses of actions, estimates of likely reactions to the deception, and explanations of how the adversary processes, filters and uses information.

Compare this with what we know about a computer attacker: Nothing. Well, okay, we have an IP address that is probably spoofed or anonymized. There can be several people working behind a single IP or IP range, like an attack team or even unrelated persons sitting at an Internet Café. Likewise, the same person can utilize multiple IP addresses, unknown to the defender, and have different probes classified as belonging to separate attackers, or have some of them remain undetected.

Of course, this depends on how we detect and identify attackers. If we only work at the network level, this is what we have. If we further assume that we do not have any external means of gathering information, our only source of information will be the actions the attacker performs in our system and which are observable to us.

From the point of view of military doctrine, this is nothing less than a disaster. We want to know how the target thinks, acts, interprets intelligence, what he relies on, what his objectives are, why he is here. Clearly, we have far too little information to fulfill these demands.

What should the consequences of this be? Should we try to alleviate the problem with observations or other means, or are we forced to avoid or accept a large number of assumptions, on shaky grounds? This is an issue that does not have any easy solutions.

8.8.2 Indicators

In phase five, first action (complete the deception plan), there was a set of issues related to indicators. Under the header of "identify the deception means" the following four-step chain was given: (1) determine the adversary's detection and collection capabilities (2) identify indicators [that would be affected by "the real thing"] (3) compare capabilities to indicators (4) select means.

Let us restate the first three steps: (1) What can the attacker see (2) how would things look if the deception is real (3) what is the intersection of one and two. This problem can be considered in two different ways, whereas the first focuses on what is theoretically possible, and the second on what is the case with a specific attacker.

We are unaware of any study explicitly mapping out possible indicators. Some cursory thinking makes us suggest at least seven types:

(1) System topology: The logical layout of hosts, network components and higher level services.

(2) System characteristics: Properties derived from the system topology like route latencies.

(3) System contents: Information stored in the system, like the contents of data files.

(4) Traffic patterns: The flow of traffic, both human generated and automatic protocols.

(5) Traffic contents: The information found in the packets that traverse the network, as opposed to that which is stored in memory or on a hard drive.

(6) Usage patterns: How humans and automated processes use the system.

(7) Internal application environment: The characteristics of the environment presented by an application. If the application is complex, one can speak of internal topology, contents, and patterns.

To build any type of indicators, there are at least two strategies that can be employed. The first method seeks to recreate the real event as correctly as possible within a reasonable cost, while the second anticipates what channels the target uses, and therefore limits itself to manipulating indicators observable to the attacker by those channels. The former has a higher hope of being successful than the latter if the target has access to unknown channels.

A problem inherent in the computer system seems to be the fact that new indicators can be created in unforeseen ways. This is somewhat related to the problem of covert channels. If the attacker can combine or process information we have falsified to get meanings outside the scope of our considerations, we are in trouble. For a real example, take the detection of Sebek. By using a program like **dd**, an attacker can exploit the reflexive response of Sebek, which is to send packets with information about the read calls. The packets themselves may be hidden, but the effect on network throughput is not, and this the attacker can measure (Oudot and Holz 2004b). Our error lies in the failure of considering ways in which Sebek is visible through other indicators, in certain circumstances.

8.8.3 Feedback

The matter of feedback is closely connected to that of intelligence. JP 3-58 calls for two types of feedback: Operational feedback, showing us what information is reaching the target, and analytical feedback, showing us the actions the target is taking because of that information. Do we have access to this type of information in the computer system? If we do not have any external means of getting intelligence, we are restricted to observation. Maybe we can see the attacker capturing packets, downloading files, or attempting to use passwords. What exactly we can derive of the attacker's intention and information processing methods from this is highly variable.

This is reflected in the problem of *verification*. Whaley stated that feedback should be used to ascertain that the target has noticed the portrayal, found it interesting, formed the intented hypothesis, and failed to detect the deception as such. It is especially the latter that is of interest to us. Is it possible to separate between a deceived attacker and one going for counterdeception, with any certainty? This is of importance if we make any further decisions based on deception success.

If we consider the computer system as a closed sphere, separation of deception and counterdeception is very difficult. This is due to the fact that no actions, from the attacker's point of view, have any *cost*. He is, after all, inside our system. Traditionally, if someone was deceived in warfare, he would execute some actions that had cost, as the movement of troops, bombing position X instead of Y et cetera. A counterdeception scheme in such a situation demanded that the target would have to bear those costs, at least as far as the original deceiver could verify. But there are few actions that have any sort of cost directly associated with them in a computer system, from the point of view of an attacker. Perhaps it is possible to create some sort of burned bridge scenarios, where an attacker must sacrifice a set of future advantages. However, to have any real cost on the attacker's part the deceiver must probably make the deception have consequences for the target outside the computer system.

Also note that the opposite is the case for the attacker! The attacker can use all public information to correlate what he learns within the computer system with the external world. If we seek to deceive the attacker with information having ramifications outside the computer system, we must consider all of the ways he can reach information related to the deception. This is perhaps more often the case than we believe. For instance, if we try creating a subnet representing a false department of an organization, have we considered all the paperwork and traces that reflects a real department in sources available to the public? What about false employees, produced material, payments, and so on. Stoll (Stoll 1989) "simulated" a small office by himself, creating documents for a false secretary. This was viable in 1989, but it is far easier today to verify this kind of information by other means (social engineering, google?) Clearly the level of sophistication and the resources available to the attacker have a large impact on what is needed.

8.9 An Interlude

Final Act: Revelation

The silence is loud while Bubba tries to put his thoughts into words. "Frank?"

"Yes?"

"Why do I hear 'Doh!' from that room over there?"

"Oh, that's just John, my tenant. He's renting a room."

Bubba looks skeptically at Frank. "Really. Let's take a look." He crosses the living room and quickly opens the door. Inside, John the tenant is facing four computer screens, tagged with Alpha, Bravo, Charlie, and Delta.

"Frank! You have hostiles inside you perimeter!" He looks accusingly at Frank. "What happened to physical security?"

John the tenant leaps up from his chair, and begins a revealing monologue in the spirit of Agatha Christie. "Ha! The computer specialist, playing at deception. Little did you know that I heard every plan, and adjusted my actions to fit **your** *preconceptions. As if a simpleton couldn't have seen through the rerouting system. The Delta probe was in the real system, while you played with my diversions."*

Frank asks John the tenant "but why? What could you have to gain from this? Are you working for Snazzy Foodstuff?"

"Pah. That was just a cover. I'm doing exercises for a computer security course at the University. And I rolled over you like a Sherman! Pathetic."

Bubba snaps "thats it! You're going down, buddy! I'm sicking my telco team at you, unless you're out of here in 10 seconds. And leave the gear, that's the price you pay for messing with Bubba (for friends, not you)."

"Whatever!" John stalks out and disappears into the night.

"And you!" Bubba turns to Frank. "We're going to talk PHYSEC, OPSEC and every other SEC in the book. Cannot have things like this going on at my watch."

"But..." Frank protests. "I have deliveries!"

"Your son is promoted to acting driver. Back to school. Lesson 1: Secure the area. Lesson 2: ..."

– The End –

Chapter 9

Model

It is time to continue with the main question we raised in our discussion. Where does the border between deception and computer deception lie? To answer this we must know more of what happens during attack and defense. In this chapter we will try to construct a model, an understanding that can aid us in finding an answer.

Our intention is to describe the connection between the attacker, the defender, and the computer system, as generically as possible. By this we hope to derive some results that are valid for a large set of attacks and defenses, as opposed to for a specialized instance of an attack or defense type.

We start with the basic partitioning of reality given in the Domain Model (Alberts, Garstka, Hayes, and Signori 2001). Then we will place two human beings and a computer system in the model, and see how they interact on a purely generic level. After that we will reintroduce the concept of "courses of actions" as the basis for a simple tool - the COA map - that helps us understand the relationship between effects and consequences, inside and outside the computer system. Then it is time to specialize somewhat, by sketching out a generic attacker planning and execution process within this framework, and a corresponding defender process. We will then look closer at deception, and how the use of multiple COA maps can make clearer what happens when deceiving. Finally, we will consider how an attacker relates to the computer system, how he uses his tools, and the consequences of this for the separation of deception and computer deception.

9.1 The Domain Model

In order to understand the interplay between attacker and defender through the computer system, we needed a model that could account for cognitive models and the flow of information between them. Such a model, the Domain Model, is given in (Alberts, Garstka, Hayes, and Signori 2001) as the foundation of Network Centric Warfare concepts. The model is also used by the Norwegian equivalent "nettverksbasert forsvar" (Forsvarssjefen 2003).

9.1.1 The Three Domains

The Domain model is an attempt at showing the fundamental relationships between the physical world, the information that it carries and our understanding of that information. The model splits reality into three domains, one for each of these concepts: The physical domain, the information domain and the cognitive domain. See figure 9.1.

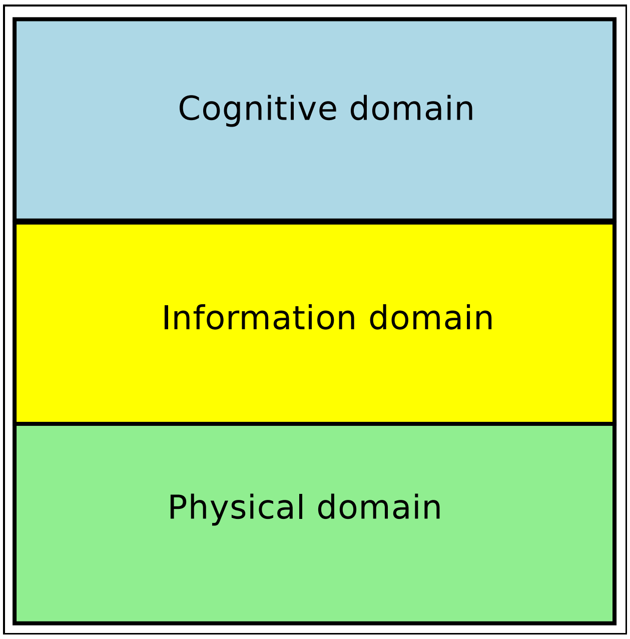

Figure 9.1: *The Domain Model.*

The physical domain is the world where physical objects, animals and human beings exist. Entities in this domain are tangible and measurable. Traditionally, this is the domain where wars have been fought.

The information domain is the sum of information external to ourselves. A book has content, beyond the physical marks on its pages, and sensors produce data that has informational meaning. This information can be correct or incorrect, altered, removed, stolen, inserted, deleted. We interact with this domain when we get information that is not directly sensible to our native senses.

The cognitive domain is the sum of all minds. Information is perceived and mirrored in our heads, where we have an understanding of the real world. Each person has a separate sphere of this domain, and sees the physical domain and the information domain through a personal filter built up by personal experiences, expectations, training, and capabilities. This filter, or mindset, is mutable, but not removable.

Many terms and concepts have been defined for use in this model. Currently, we have only used the basic partitioning of the domains, and modified it to suit our purpose.

9.1.2 Defining Subdomains

What place does the computer system have in this model? Clearly there are physical components like the computers themselves, network equipment, cables and so on. The computer system is therefore partly within the physical domain.

There also exists information about the computer system, as well as related or unrelated information *inside* the system. This means the computer system also is partly within the information domain.

Last, if we use the expanded definition of cognitive structures, we also must acknowledge that the computer system is partly within the cognitive domain. Whether you regard the computer as one entity with a rather complex cognitive structure, or as a composition of several smaller structures, is irrelevant for this point.

As a consequence, we can define the *computer domain* to be a subdomain of the three regular domains, as depicted in figure 9.2 on the next page. Note that the drawing is generic; in theory we can use the same arguments to define other subdomains.

9.1.3 Human Actors

As we placed the computer system in the domain model, so we can place human beings. We are physical beings in a physical world, while our minds contain separate partitions of the cognitive domain. Figure 9.3 on the following page shows two human beings plotted in a slightly modified do-

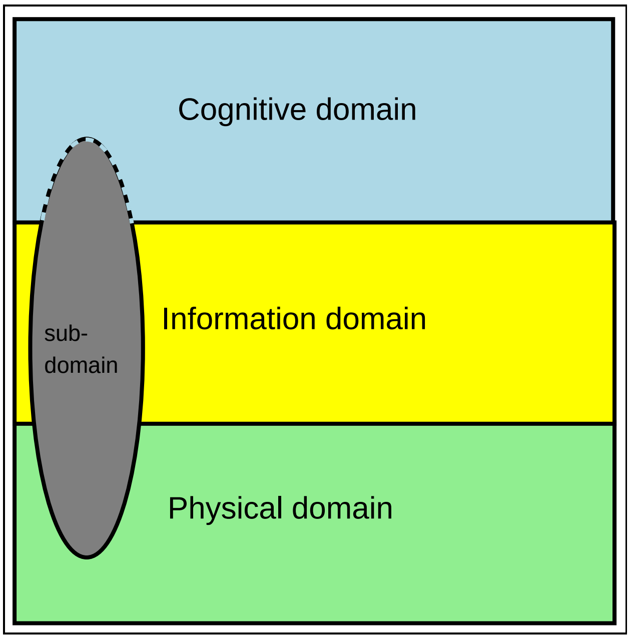

Figure 9.2: *The computer domain as a subdomain of the physical domain, the information domain and the cognitive domain.*

main model, with Cohen's three-level structure in the cognitive partition of each human being. Note that the cognitive partition is not drawn to scale, and the smaller blue vertical stripes outside the human partitions symbolize other human beings and entities with cognitive structures.

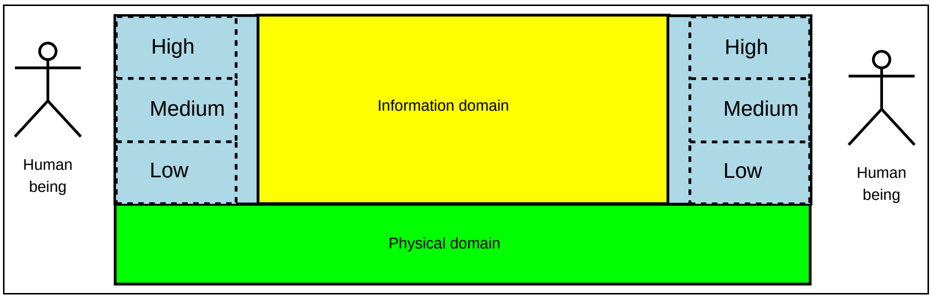

Figure 9.3: *Modified domain model including two persons with cognitive structures.*

We are also users of the computer domain, interacting with the domain itself, and communicating with other users. See figure 9.4 on the next page. Here we see that the computer domain is a subdomain of all the three domains.

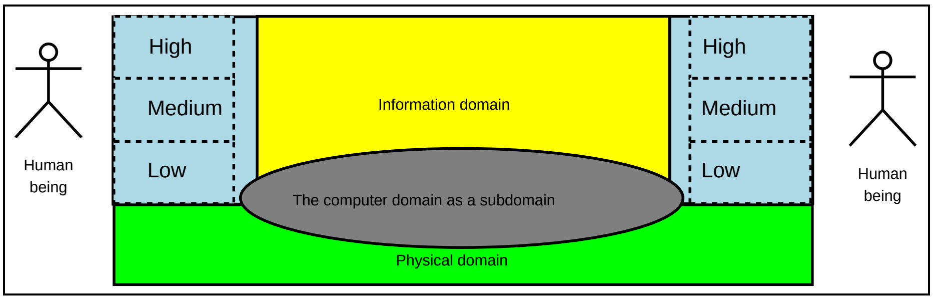

Figure 9.4: *Computer domain mapped onto merged domain and cognitive structure model.*

9.2 The Nature of Computer Domain Interaction

For every action a human being performs, we have changes in the physical domain, the information domain and the cognitive domain. The extent of changes is, of course, dependent on the action that is carried out.

Actions not only have consequences in all three domains, but also have consequences in multiple subdomains. That is, an action in one subdomain can cross into another subdomain. By thinking about it for a second, it becomes clear that this is a necessity in several ways. First, the lines between subdomains are drawn up arbitrary by human considerations. What forces could enforce uncrossability? Second, by reductio ad absurdum: By assuming that consequences cannot cross, it would be impossible to use or interact with any subdomain.

Consider human computer interaction. When a human being interacts with the computer domain, information flows from the "self", down through the cognitive structure, through affectors (usually fingers) and into the computer domain. If we are communicating with another human being, the information ripples through the computer domain and up into the other human being's cognitive structure, usually through the eyes. See figure 9.5 on the following page.

Let us go through a human computer interaction example in more detail. A human computer user, Alice, is working on a budget proposal for her employer. After editing the document locally on her computer, she is about to update the old version which is stored on a shared fileserver.

(1) Alice decides to transfer the file. Alice makes a decision to act. This changes her physiological state as nerve signals transmit the decision, which flows from the top of her cognitive structure downwards. Finally it is executed by her hands which type at a keyboard. Anyone

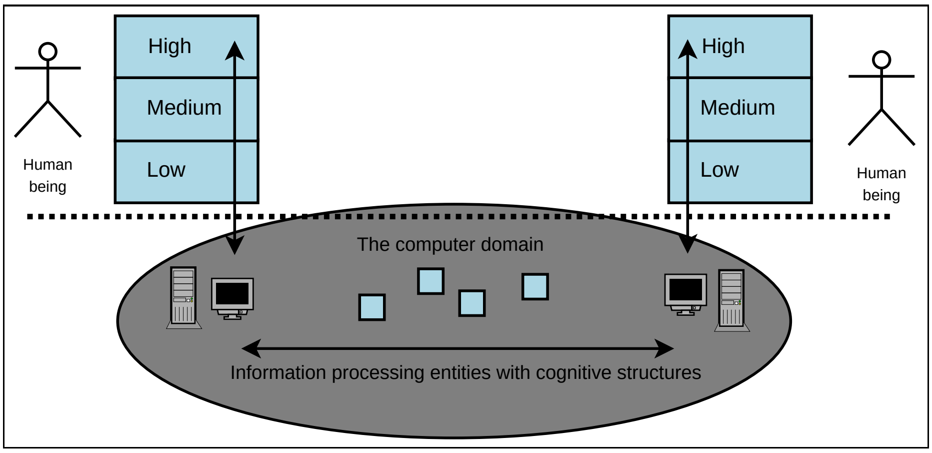

Figure 9.5: *Flow of information through the computer domain.*

in her surroundings, either human beings or other entities (cameras?), can detect and be affected by her decision.

(2) Actions are taken which cross into the computer domain. The computer detects keystrokes by the interrupt mechanism, interprets them and executes Alice's commands. The computer equipment has changed state, and makes changes in the information domain and the cognitive domain: Internal representation of information is changed as well as the state of affected information processing entities. As a consequence, the file is sent flowing through the computer domain.

(3) Consequences percolate through the computer domain. Changes wrought by Alice's commands ripples through the computer domain, where some effects are intended and others unforeseen. The file is transferred by various computer equipment along the way. Generally, the workload of the equipment will be slightly increased without any great consequences.

(4) Consequences cross out of the computer domain. After updating a file, colleagues of Alice read the altered file and change their actions accordingly, affecting other people in turn. There are also consequences on Alice's side, when she sees that the file has been successfully updated.

Note that unintended effects almost always happen in all domains, and other subdomains, continuously. As step number three is underway, so is step number four. Alice thinks of step three and four as sequential, since the consequences she awaits will follow after certain consequences in the computer domain. The consequences that cross out while step three is underway, are usually considered ignorable, but this is not necessarily the case.

For example, Alice's file is split up into network packets and sent on its way, increasing the use of bandwidth. Maybe one of those packets is the final drop, making the network congested and affecting one, five, ten other users whose connections suddenly are lost. Perhaps one of those packets unintentionally contains a virus signature, making the company IDS overreact and remove the file, which means that Alice misses the so-important deadline, the company loses the bid for a contract, and hundreds of people lose their jobs. Perhaps Darth, employed by the competitor, is playing man-in-the-middle, and alters Alice's packets in transit, intentionally triggering the whole event.

9.3 COA Maps

We have spoken of actions and consequences crossing domain boundaries without discussing the nature of actions and consequences. Let us take a closer look at the difference between them, how they relate to the term "course of action" or COA, and how we might visualize them in a simple way.

When Alice, or any human being, interacts with the computer domain, she has a set of possible actions that she can execute, leading to a set of consequences. The same is true in every subdomain, or real life in general. Sometimes there are nearly no discernable differences between the execution and its consequences. For example, you want a chair moved two meters to the left, so you move the chair two meters to the left. Total overlap between execution and consequence. But the execution is the *action*, while the consequence is the *result*. Moving chair. Chair moved.

Sometimes the desired consequence is less tightly bound to the initial action, as we often see in the computer domain. An attacker might perform a technological action, which he intends to have a certain consequence. For instance, a lot of generated packets usually suggests an attempt at bandwidth exhaustion, ie a denial of service attack. There is more uncertainty of the consequence here, for instance the attacker might succeed in generating a lot of packets, but not in achieving bandwidth exhaustion.

Usually, the reason for lower certainty is that one relies on a chain of consequences following from the inital action, and some of these consequences have weaker causality. To capture the fact that we often speak of such chains, we will use the term course of action, COA, that can cover an arbitrary level or depth of application. Depending on the situation one may take a macroscopic view of events (COA: attack, consequence: penetration) or a microscopic view (COA 1: launch intelligence probe, consequence 1: initial data, COA 2: etc ...).

If we think back to our example with Alice, we see that we have three sequential sets of COAs and consequences during computer domain interaction. After Alice's decision to act we had a set outside the computer domain (step 2), ripple effects through the computer domain (step 3), and consequences crossing out of the computer domain afterwards (step 4). As a visualization tool, if we think of COAs as links, and consequences as nodes in a map, the COA map, these three steps can be illustrated as in figure 9.6.

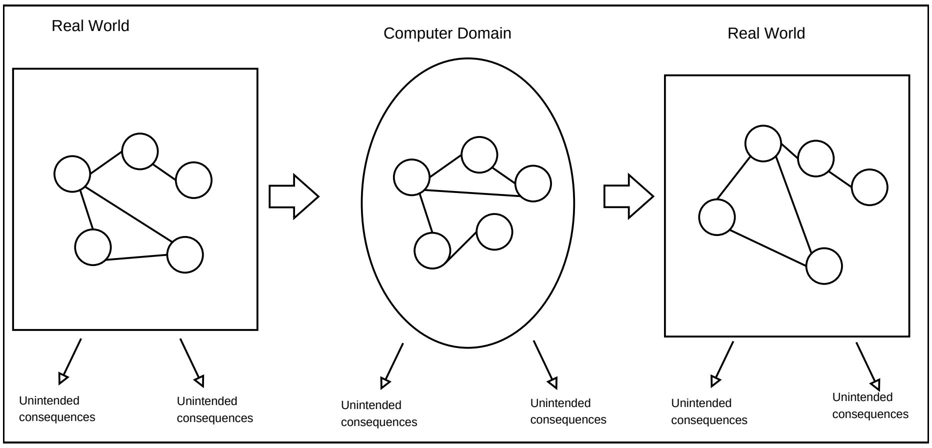

Figure 9.6: *Actions going from real world, into the computer domain, and affecting consequences both in the computer domain and outside. The real world, in this instance, means everything not inside the computer domain, including part of the physical domain, the information domain and the cognitive domain.*

While the prime worth of the COA map is not its direct visualization aspect in a specific situation, it can be used to sketch out possible decisions and events. For an example, se figure 9.7 on the facing page. This is a very basic drawing of COAs in the computer domain that an attacker thinks he has when planning an attack.

The attacker begins without information in state A. He considers the initial scan to be a necessity, and does not think it likely that he will be detected after the scan.

The scan results in a basic network topology overview, state B. From this he can choose between a more targeted scan resulting in detailed host information (state C) or go for an exploit attempt at once. He acknowledges the fact that if he immediately attempts exploitation he can be successful or fail, but does not believe that an exploit based on detailed host information

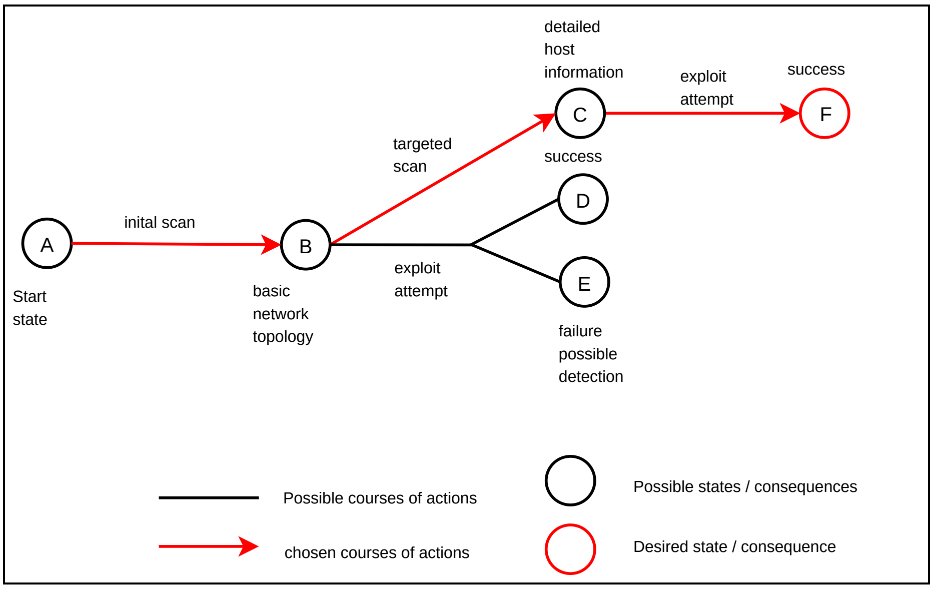

Figure 9.7: *Attacker COA map.*

can fail. Ultimately, he chooses the COA with the extra scan, and achieves state F.

9.4 The Attack

We can draw up an attacker's planning and execution process within the framework of this model. Just as a deceiver has a strategic objective that is supported by a deception objective, we can consider the attacker to have a strategic objective supported by an attack objective.

Every action the attacker wants to carry out by his computer will launch a chain of actions and consequences from the real world, into and through the computer domain, and consequences crossing out to the real world again. See figure 9.8 on the next page. In theory, any of these actions or consequences can be the event the attacker has defined as the attack objective, symbolized by the thin arrows.

However, we argue that the attack objective is always a consequence sought in the real world. This can be everything from satisfaction brought by a successful attack, to money gained by selling stolen information, to increased reputation amongst blackhats. Ultimately, the computer domain is a means. As always, there is a chaotic entanglement of strategic objectives fighting and supporting each other in a human being.

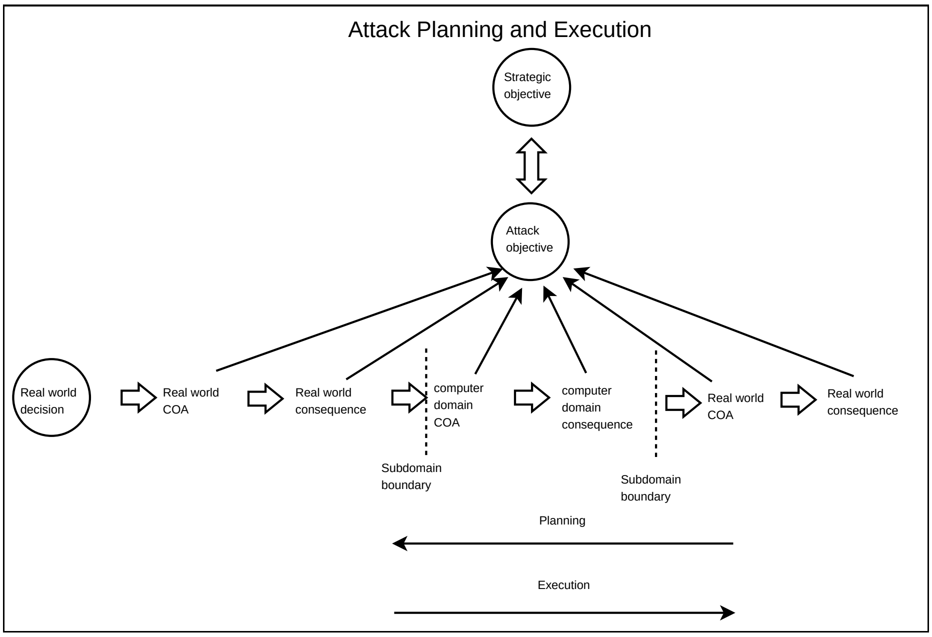

Figure 9.8: *An attacker's planning and execution process.*

Note that the chain of thick arrows leading from COAs to consequences and across domain boundaries in reality is a selected path through a large COA map.

9.5 The Defense

As defenders, we have the choice between trying to counter the attacker's objectives, or to support a strategic objective unaffected by what the attacker attempts to do. Irrespective of what we choose, we have no choice but to respond to the detection of the attacker[1]. With our knowledge of the attacker planning process (figure 9.8), we have to choose which event we should respond to. Additionally, we have the choice between responding to the event as we detect it, or as we believe the attacker intended it to happen. See figure 9.9 on the next page.

Here is an example illustrating the differences between the various response possibilities.

1. A defense mechanism detects an event e_0. The mechanism might re-

[1] Unless we consider static defense mechanisms that perform no reactions, but merely are a part of the environment.

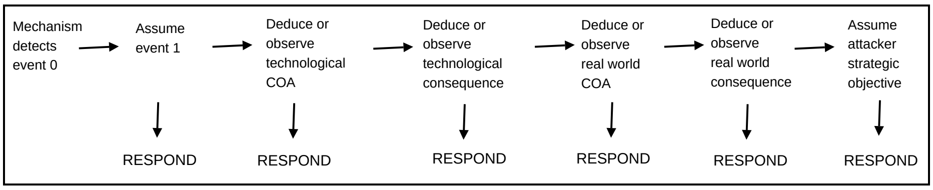
Mechanism detects event 0
Assume event 1
Deduce or observe technological COA
Deduce or observe technological consequence
Deduce or observe real world COA
Deduce or observe real world consequence
Assume attacker strategic objective
RESPOND
RESPOND
RESPOND
RESPOND
RESPOND
RESPOND

Figure 9.9: *Possible defense detection and response levels.*

spond in various ways; normally notification is given to another entity (a mechanism or human) which in turn takes action. There might be intermediate recipients before the last recipient is notified.
Example: A network interface is listening in promiscuous mode for inbound traffic, and triggers when the number of packets over a fixed interval of time is above a threshold, ie $p > h$ when $t_2 - t_1 < i$ for threshold h and interval of time i.

2. The receiving entity is notified of the event, but might not have the same idea of what that event is. Therefore, the entity assumes event e_1 has happened, which may or may not overlap with e_0. The entity may choose to respond.
Example: We are notified of an event and decides this means that many packets are being sent into our network. Possible response: Block inbound traffic.

3. We begin to reflect on the possible technological COA sought by the attacker.
Example: We assume the attacker's intent is to send a lot of packets into our network. Possible response: Block attacker packets.

4. We begin to reflect on the possible technological consequence sought by the attacker.
Example: By using all possible bandwidth, the attacker seeks to deny the normal operation of our network (denial of service). Possible response: We use additional links reserved for prioritized traffic, which remains unaffected by the attack.

5. We begin to reflect on the possible real world COA sought by the attacker.
Example: We believe the attacker wanted rumors of the successful attack to spread. Possible response: Make excuses to those affected, imply that something else was the cause. Get the responsible attacker so he cannot start rumors.

6. We begin to reflect on the possible real world consequence sought by the attacker.

Example: We believe the attacker wanted us to lose face by implying that we don't take security seriously. Possible response: Press statements saying we responded rapidly and effectively, we have tightened the security vulnerabilities used for the attack and have employed additional security experts to strengthen security even more.

7. We begin to reflect on the possible strategic objective of the attacker. *Example:* After an extensive investigation, we identify the attacker as one of the security analysts we just hired to strengthen the network. The analyst had previously been brought in for reporting on our security, but his proposals for changes was deemed unnecessary and he was let go. Being in financial troubles, the analyst decided to create job openings.

The defender's strategic objective is the one that decides if the countermeasures at any level are successful. If the strategic objective was to maintain a certain level of bandwidth available, response two, three, or four would have sufficed. If we were concerned with defeating the real world COAs and consequences intended by the attacker, response five and six would have sufficed. However, to pick out what real world changes the attacker desired would not have been easy. We can only guess at those we either see following, or those we suspect could have followed with a relatively high certainty, and perhaps fitting an attack profile (if we have one).

The last response, direct defeat of the attacker's strategic objective, is the hardest. Since we control the example, we let the company discover the attacker's motives. Accomplishing this with an attacker known only by his (anonymized) IP address, is hard.

9.6 COA Map Warfare

From our model of the attacker planning and execution process, we know that the attacker goes through two subdomain boundary crossings, into and out of the computer domain.

Since we do not know where the consequences that support the attacker's strategic objective are, it stands to reason that to defeat it we should break the chain as early as possible. This is also pertinent if we want to limit the number of changes wrought by an attack. The earlier we halt the ripple effect from spreading, the better.

Unless we have an intelligence and response branch that can act proactively, we are limited to acting in our part of the computer domain. We can draw COA maps of possible attacker methods, and use these as an aid

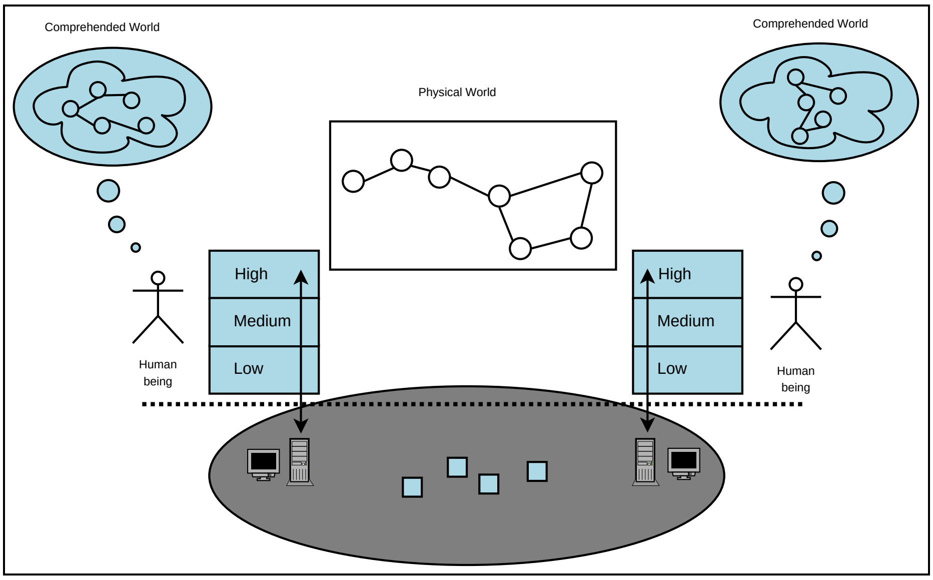

Figure 9.10: *Cognition COA maps of attacker and defender, and the actual physical COA map.*

in defense and deception planning. In a way, we are going to do this, but probably not as expected.

9.6.1 COA Map Bonanza

When we described the COA map in 9.7 on page 113, we did it from the point of view of the attacker, ie it was a simple map of how the attacker regarded the situation. We could just as easily have drawn a map of how the defender saw the situation. In fact, as Whaley concluded, we all perceive the world and build a reflection of it inside our brains. Whether or not the use of COA maps is the best way to model a part of that reflection, it is clear that we reflect on possible ways to act, make judgments on outcomes, desired or undesired, likely or unlikely, and decide on some actions while discarding others.

Since an attacker and defender have their own maps, we have two maps to deal with. In addition, we can consider a third map - the real actions and consequences that actually are possible, and in the end carried out. The first two live inside each participant's head, while the last is the objective map that is in the information and physical domains. See figure 9.10.

Deception, when considered against a conscious opponent, is about creating mismatches between the target's map and the real map. The deceiver

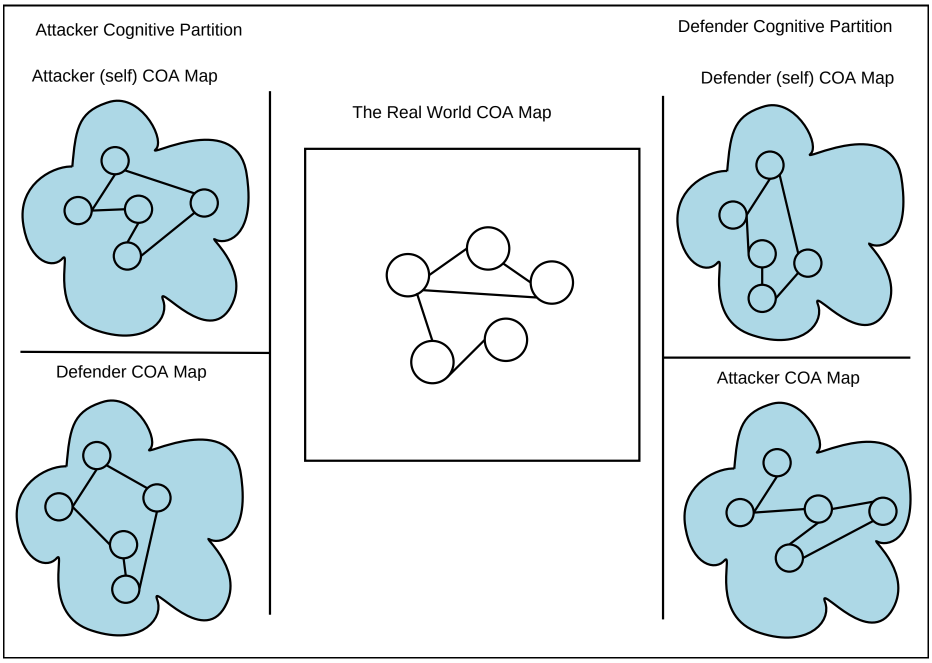

Figure 9.11: *Five COA Maps, one for the real world and two each for attacker and defender.*

uses his own map to decide on COAs leading to such mismatches. To achieve this, he must estimate the target's map. If the target knows that he is in opposition with a defender, or, as is our case, with a deceiver, he will also estimate the deceiver's map. That means we have no less than five maps to keep track of. See figure 9.11. The map in the middle is the actual COA map existing in the information domain and the physical domain. To the left is the attacker, with a map of his own choices and an estimated map of the defender. To the right is the defender, with a map of his own choices and an estimated map of the attacker.

9.6.2 Manipulation of COA Maps

What kind of operations can we perform on a target's COA map? In theory this would depend on the type of target one is attempting to deceive. We, however, use the four basic effects we claimed were possible on every cognitive structure: Hiding, confusion, misidentification, and attention diversion or attraction. These actions can be used to affect the existence and form of both COAs and consequences. As we already mentioned, more advanced effects like those of Dewar can also be accomplished by using these actions.

9.7 The Relationship Between Human and Computer System

We are interested in knowing more about the border between deception and computer deception. Our basic setting comprises of an attacker targeting a defender or his computer system by the use of computers, and a defender utilizing deception as a defense mechanism. It would be logical to assume that the way in which the attacker uses the computers has a lot to say about how deception can be used as a counter, and yet we want to avoid limiting ourselves to a specific attack methodology.

We will describe two roles the computer system can have, and, in generic terms, how the attacker can use tools for interaction. By looking closer at the flow of information we hope to see where deceptions ultimately seek effects, and to resolve the issue of whom, or what, we are deceiving.

9.7.1 The Role of the Computer System

You might recall from Dunnigan and Nofi the distinction between human-generated and system-generated information. In our view the difference was one of degrees rather than that of kinds, and yet the distinction seems to reflect an important aspect. We can regard the computer system as a *channel* of information, or as an *environment* with a multitude of characteristics. The two distinctions do not correlate one hundred percent, as the computer system can be a channel of computer-generated information, and the environment can be manipulated by humans.

If you plant files with false information regarding situations and events outside the computer system, you are using it as a channel to spread disinformation. If you create false hosts, you are manipulating the environment in which an attacker maneuver. The midway would be planting files with false content regarding the computer environment, say, a false password file.

Clearly, if the computer system is regarded as a channel of information for human being, we are dealing with human deception. If computer deception is to be separated from deception, it must be when it is considered an environment, or when the target for disinformation ultimately is a computer entity.

9.7.2 How Are Tools Used

We said we wanted to avoid a specific attack methodology, and yet we must have some sort of idea of how the attacker uses his tools to get anywhere. If we know nothing of this, how can we know what we are deceiving?

To resolve this issue we decided to use the OODA loop as a model of the different ways in which a tool could be used. The OODA loop is a model developed by Colonel John Boyd[2] that breaks the general decision making process down into four phases: Observation, orientation, decision, and action. Basically, we are observing and assessing what is happening, we make a decision to act, and we execute those actions. The OODA loop has been much used by military doctrine, although it has been pointed out to be insufficient to model the real human decision making process (Alberts, Garstka, Hayes, and Signori 2001).

If we incorporate the OODA loop into our model, we can think of both attacker and defender as being inside a separate loop. Taken a step further, we can also apply it to computer entities. In human beings, we consider the COA map being built in the observation and orientation phases, a route to execute is decided on in the decision phase, and the route is traversed in the action phase. For computer entities it is a question of resolution: We can apply the OODA loop at a microscopic level, as instructions are read and executed, or we can zoom out and think of larger entity as objects or tools. In the latter an object could be a password checking entity, which observes and analyzes input, decides on acceptance or denial, and executes the decision while informing the user.

If we consider a tool used by the attacker as an entity performing a function for the attacker at some stage in the OODA loop, we find that most tools are used to observe the computer domain, ie scanning tools and the like. In a way, these tools can be considered the equivalent of human eyes into the domain. The other phase we see covered by tools is action, as when exploits are launched. There does not seem to be much use of tools in direct analysis or decision making, as Cohen also indicated (section 6.2.4).

9.7.3 Flow of Information Through Tools

If we use the above notions of the attacker's tool as a deceivable entity, we have information flowing into the tool from the computer environment, the tool processing and acting upon the information, and in the end send-

[2] Unfortunately, Colonel Boyd never published his theory. Brief explanations of the model can be found in (Alberts, Garstka, Hayes, and Signori 2001) and (Forsvarets overkommando 2000). Further, a lot of content related to Colonel Boyd, including sets of slides from his lectures can be found at (Kettle Creek Corporation 2005).

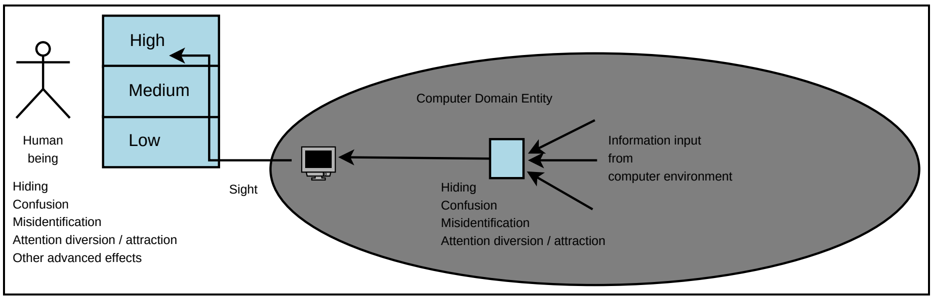

Figure 9.12: *Flow of information through a computer domain entity.*

ing output to the human attacker. On the tool we can achieve the basic effects of hiding, confusion, misidentification, or attention diversion or attraction, while on the human being we can achieve the same as well as more advanced effects. See figure 9.12.

We stated three possible definitions of computer deception in section 8.3: (1) deception of computers (2) deception with computers (3) a mixture of one and two where we deceive computer entities but expect guaranteed effects on the human attacker. Let us look closer at option one and two, while saving number three for the next chapter. To resolve the issue, we will track various effects through the tools and to the human attacker, and consider what effects are needed for success. For this we will use Nmap as an instance of a typical scanning tool. We assume that we employ deceptions that successfully (1) hide a host from Nmap (2) confuse Nmap (3) make Nmap misidentify the operating system of a host. The fourth effect, attention attraction or divertion seems to be inapplicable to Nmap at this level.

Case 1: If Nmap does not see a host, the attacker will not see it either. As such, we are successful. But what if the attacker uses another tool which find the host? Are we still successful? We believe the overall deception objective is most correctly stated as "hide host from attacker", not just "hide host from the specific tool Nmap". Both objectives are valid as deception objectives, but the value of the latter alone is small. The problem is, perhaps, that it is easy to think that the latter implies the former, which it unfortunately do not.

Case 2: We try confusing Nmap. This can be done by replying with invalid packets or invalid combinations of characteristics, making identification impossible. The consequence for the attacker would be a combination of hiding or confusion, depending on how Nmap reports the information. If Nmap reports nothing (invalid packet), it is hidden from the attacker. If Nmap reports "unknown" on all characteristics of a host (no appropriate profile), the attacker faces uncertainty. Still, we have the same issue as in case 1. If the attacker manages to circumvent the deception, are we successful? From the point of view of a defender, we believe not.

Case 3: We make Nmap misidentify a host, and this is reported to the human attacker. This is perhaps what is closest to existing technology today. But again, what is the point of deceiving Nmap if the attacker remains undeceived? If the attacker can reason along the lines of "since Nmap has shown Y, what truly is the case is X", nothing would be gained.

9.8 Chapter Conclusion

When we are deceiving tools that support the observation phase of a human OODA loop, we are ultimately interested in deceiving the human attacker that is utilizing the information[3]. This supports the view of deception *with* computers being the proper definition of computer deception.

Most attacks, beyond the most simple point-and-click scripts of an amateur, feed information to a human decision maker. If the attack was completely automated, we could speak of deception of computers, but most often there is human involvement somewhere in the attack loop. For it to be otherwise, there would have to be (1) no humans involved or (2) any human would have to be completely irrelevant, in that what they do have no impact on how we run our deception, nor on success or failure. In such a case we would be fighting against the human *designer* rather than the human *operator*.

The only scenarios fitting this, as far as we can tell after cursory reflection, are autonomous computer worms and attacks where a final launch has been set irrevocably in motion. That is, the human decision has been made, and we are left with using deception for limiting the consequences of the decision. An example of the latter could be a denial of service attack.

Interestingly, if we dare to step slightly beyond the scope of this thesis and consider deception for *attack*, the situation is different. Much of a defensive system run without querying human decision makers. A password checking system grants or denies access, while an ATM machine decides on whether or not to give out money. In such a case an attacker fights with the human designer, and deception solely of computer entities is a viable option. In fact, by our definition of deception, we would be hard pressed to find examples of computer attacks that are not deceptive in some way or another.

[3] Of course, three cases do not rigorously prove anything. But we fail to see any reason for the argument to be incorrect for any observation tool, which includes all scanning tools.

Chapter 10

Scenarios

Our conclusion in the previous chapter stated that computer deception is properly defined as deception with computers, unless we wish to restrain ourselves to a very limited set of attacks. This points to a need for psychology in order to understand the consequences of deceptions on the human attacker. We did, however, also mention a slim chance of an expanded definition of computer deception while retaining the basic notion of deception *of* computers, if the effects sought on the human being were guaranteed to work.

Initially, it does not look promising. The problem with this approach is that unless you exploit the physical properties of computer equipment to achieve physiological effects on a human being, most deceptions targeted against another computer user would initially leave the computer domain through a computer screen[1]. Unless we tweak the output of the screen in such a way that the unconscious perceptual system is affected[2], we would not be using visual illusions, rather the deception would be in the informational content of the data conveyed. This would mean that any effects would be at the medium and high levels of Cohen and Lambert's cognition model, of which the consequence is increased uncertainty. The only level where we have high certainty of the effects of deceptions, is the low physical level. But that level is not available to us when dealing with computer deception.

Nevertheless, we will go through a set of scenarios where we try to minimize the number of psychological assumptions we must accept in order for our deceptions to work. We have tried to use our COA map theory in find-

[1] The argument also works with false information portrayed by other means like printed paper, etc.

[2] Despite the similarity to the plotline of Neal Stephenson's Snowcrash, we consider this to be unlikely.

ing such deceptions, but there are, of course, no guarantees in there being no other deceptions with alternative deception objectives that would make fewer assumptions.

The scenarios have only been considered theoretically, ie no practical experiments have been performed. Which is kind of sad, yet more importantly, it is risky on a subject which has the type of uncertainty that deceptions have.

10.1 Visible Model Parameters

Our theory has several components that we can use in scenarios, for instance:

(1) COA Maps Our starting point will be the five COA maps, and our deceptions will try to accomplish changes in the contents of some of the maps.

(2) The Assumption Chain Every deception runs from portrayal to the support of a strategic objective. The assumptions we make can be described in terms of (1) demands on knowledge of target (2) consequence of the deception becoming known (3) resistance to counterdeception (4) need for feedback.

(3) Indicator Properties We can state the deception in terms of the aspects we covered earlier, ie detection or non-detection, conveyance, release, contents and intent.

(4) Detection Level On what level are we detecting an attack, and on what level are we trying to respond.

(5) Ripple Tracking It is possible to track the intended effects of both the deceptions and the attacks which they should counter. This can be done through tools and entities in the computer domain, and further to effects and consequences on the outside.

(6) Application of Principles We can analyze our methods with regard to the four principles we have established.

It will, however, be very cumbersome to cover all of these points in detail for every deception. Therefore, we will cover those we find the most illuminating.

10.2 Scenario Preparations

As we repeatedly have stated, the strategic objective is the alpha and omega that defines success. Since it is impossible to cover all strategic objectives, we have decided to split them into two types or classes: Those that desire inaction, and those that desire action.

Think of the situation from the defender's perspective. There is an attacker inside the defender's network, and in terms of COA maps the attacker is standing in an unknown state. Since we assume exploitation is possible, some of the COAs available to the attacker lead to system compromise. This is something the defender want to avoid. Therefore, his main objective is to make sure the attacker is not traversing down any of those COAs. Hence the focus on inaction.

We can also think of situations where specific actions are desired. Perhaps the defender wants to draw the attacker's attention to a specific host, away from the real hosts[3], or to make some false information that has been prepared fall into his hands. In these cases the defender want the attacker to follow a specific COA.

The computer system we will work with is the system of Frank Schultz the Grocer, introduced in 2.6. See figure 2.1 on page 11.

10.3 Scenario 1

A common way of thinking about computer network security is by using the fortress metaphor (Bishop and Frincke 2004). The computer system is our castle, which we want to keep uninfested by thieves, rats and freeloaders. By creating walls around it we try to stop attackers from entering. A strategic objective in accordance with this mindset would be "keep attackers out".

10.3.1 COA Map Considerations

Keeping attackers out is primarily an objective that demands inaction, ie stopping attackers from following COAs that lead to gaining access. There are probably countless ways of affecting an attacker in the choices he makes, but we want to use the approach having the highest certainty of success that is possible.

[3] Which actually means that the defender is desiring an action (attention on specific host) in order to ensure an inaction (no compromise of real systems)

By considering the five COA maps, logic dictates that the best would be to actually remove the COAs from the real map. That is, use deception in such a way that the possibility of system compromise is removed in reality, not just in the attackers comprehension of the situation.

If we think back to the Honeypot interlude (see 3.4), this was exactly what was done. Detected attackers were rerouted into a Honeynet, cutting off possible COAs that could lead to system compromise. In terms of COA maps, the real map was altered, while the attacker's map of his own situation was kept intact without changes.

10.3.2 The Deceptions

There are several deceptions going on in this example, and the number of deceptions depends on how you break the situation down into smaller parts.

The overall deception would seem to be, "nothing has happened". In order to pull this off, there are three supporting deceptions: (1) Masking of the detection event (2) masking of the rerouting event (3) portrayal of the old system's characteristics.

The overall deception supports the strategic objective of "move attackers inside the network out of the network without them noticing", which in turn support the overall strategic objective of "keep attackers out".

10.3.3 The Assumption Chain

(1) Detection and rerouting chain.

- **(1) Strategic objective:** Support main deception.
- **(2) Deception objective:** Keep the attacker from noticing (1) that he has been detected (2) that he has been rerouted.
- **(3) Target belief:** Maintain existing beliefs.
- **(4) Target perception:** Maintain existing perceptions.
- **(5) Portrayal:** Mask the detection event, mask the rerouting event.

(2) Portrayal of the false system

- **(1) Strategic objective:** Support main deception.
- **(2) Deception objective:** Keep the attacker from noticing that he no longer is in the normal system.
- **(3) Target belief:** Maintain existing beliefs.

- **(4) Target perception:** Maintain existing perceptions.
- **(5) Portrayal:** Make the new system equal to the old system with a resolution good enough to fool the rerouted attacker.

(3) The overall deception.

- **(0) Overall strategic objective:** Keep attackers out.
- **(1) Strategic objective:** Remove real COAs leading to compromise, by moving attackers inside the network out of the network without them noticing, and making them spend time and resources.
- **(2) Deception objective:** Keep the attacker from noticing that he no longer is in the normal system.
- **(3) Target belief:** Maintain existing beliefs.
- **(4) Target perception:** The system he believes he is in.
- **(5) Portrayal:** Maintaining and showing the old system characteristics during detection, rerouting, and afterwards.

10.3.4 Analysis

The danger with rerouting, as we saw in the interludes, is that it is possible to understand that you have been rerouted. There are at least three different ways of detecting this: (1) The rerouting event is not completely masked, for instance when existing connections are broken (2) the new system is not fully equal to the old one (3) the attacker has multiple probes, and not all of them are rerouted. This means that the attacker will see two realities, and will suspect that one, or both, are false.

All of these three points are valid when the attacker has no initial information about the system. If we open up for the possibility of preattack intelligence, it becomes even worse. Earlier probes or other methods that use information outside the system, like social engineering, make detection of rerouting even more likely[4].

The possibility of multiple probes places heavy constraints on tailoring deceptions to specific attackers. Any deception that is dynamic and develops as it interacts with the attacker, can be revealed if the attacker has another undetected probe. It can even be revealed if he has multiple detected probes, if the defender does not realize that it is the same attacker. And it is not easy to understand that two wildly different IP addresses belong to the same attacker.

[4] This supposes that the environment usually is static. If one routinely reconfigures the environment as a security measure, changes do not necessarily imply rerouting.

This implies that dynamic deceptions should be visible to all, ie there should be one common deception state across all connections.

10.3.4.1 Indicators

The detection and rerouting deceptions both rely on masking all indicators reflecting the detection and rerouting events. When portraying the false system we hope to replicate all indicators from the real system, as well as removing all indicators not in concordance with the real system. Differences in physical properties between the real and the false system must be masked, while all derivable indicators must be thought of and portrayed. Normal honeynets fail in doing this, when usage and traffic patterns are left unportrayed.

10.3.4.2 Detection and Response Level

In principle, there are no demands as to how the attacker is detected. In the Honeypot interlude, we saw detection on a purely technical level: Any probe hitting a honeypot made the reroute mechanism trigger. There is, however, nothing that stops you from classifying an user as an attacker after an arbitrary event, and then reroute.

The rerouting response is technical. We are not trying to counter the attacker's motive, nor any real world effects he might be trying to accomplish.

On the other hand, we are also aiming to mislead by emulating the real system. But we do not place any demands on the success of this endeavor. Had we done so, then the requirements for certain success would have been much more difficult to fulfill.

10.3.4.3 Demands on Knowledge of Target

The overall deception places no demands on knowledge of the target. Any attacker being detected will be rerouted. Intelligence on the target could, however, be used to adjust the quality of the deception.

10.3.4.4 Deception Revelation

The consequence of the deception being revealed depends on the interpretation of the strategic objective. If the attacker realizes he is in a false system and quits, are we still successful? If we just wanted the attacker to spend time and resources, we have achieved what we set out to do, as long as the attacker was in the system for a period of time. If the intention was to keep the existence of rerouting mechanisms and events a secret, we have failed.

10.3.4.5 Counterdeception and Feedback

In principle we are highly resistant to counterdeception, and have no need for feedback. However, if we begin observing the attacker's actions in the false system, and make decisions based on those observations, the game changes. Then we are very much open to counterdeception.

10.3.5 Conclusion

In theory, the best one can do is to actually remove COAs from the real map. In practice, however, this is hard to accomplish while remaining undetected, at least if attacker rerouting is performed. No other method has been thought through.

While the objections covered here are valid for sophisticated and more advanced attackers, rerouting is not useless for other attackers. For instance, attackers going for targets of opportunity might not bother to examine the computer system they have hijacked. As a first line of defense, rerouting can be viable.

10.4 Scenario 2

We continue with the strategic objective from the previous scenario: "Keep attackers out". Now we will expand the "what" we are keeping them out of.

10.4.1 COA Map Consideration

We have seen that rerouting is fraught with danger, if it is important that the attacker does not suspect the authenticity of the information he is receiving.

If we cannot modify the real map[5], then we must begin altering the attacker's understanding of the situation. The closest thing to removing real COAs is removing COAs from the attacker's map.

[5] Of course, by patching and tightening security we are removing exploitable COAs, but not by deception.

There are several ways of removing COAs from a cognitive map. We can mask the COA with concealment or camouflage, or confuse the attacker about the map's layout. We will run through the example of concealing COAs.

10.4.2 The Deception

This is one of those instances where what is good in theory is harder to pull off in real life. We have decided that we want to conceal COAs, the question is, COAs leading to what?

Let us think through four alternative versions of this scenario.

(1) Concealing the Network Entry Point We do this by keeping the existence of the entrance point unannounced.

(2) Concealing a Host We do this by removing all information pointing to the host, disabling all automatically generated traffic, and configuring the host to only answer when a special handshake is followed. By this we try to hide its existence.

(3) Concealing Functionality in a Service We do this by keeping the functionality undocumented and unsupported.

(4) Concealing Functionality in a Distributed Application We do this by removing all hints of the functionality from menus, options etc. It is almost like an Easter egg. By entering a special combination in a menu the functionality is triggered.

10.4.3 The Assumption Chain

(1) Strategic objective: Keep attackers out.

(2) Deception objective: Conceal COAs leading to further system penetration.

(3) Target belief: Maintain existing beliefs.

(4) Target perception: Maintain existing perceptions.

(5) Portrayal: Conceal the entrance point, host, service or application functionality.

10.4.4 Analysis

(1) Concealing the Network Entry Point The first example, concealed entry point, is clearly debunked by experience. This is equivalent to assuming that a computer connected to the Internet will not be attacked, since nobody knows that it exists. Target of opportunity-attackers frequently scan huge blocks of IP addresses, and will find the entrance point. If any of the users of the network are known, they can be placed under surveillance to see where they connected (ie packet capturing).

(2) Concealing a Host The second, concealed hosts, reminds us of the honeypot mentality. But honeypots are found and attacked; that's the point. That does not bode well. Also, the host cannot be kept completely concealed, since it must communicate with its users. This communication will be visible for hidden attackers[6].

There was an extra security feature on the concealed host, that it should only reply to a special handshake. If the handshake is replay-resistant it gives another layer of security, but this is not deception.

(3) Concealing Functionality in a Service The third example was the concealed service functionality. If the service is not encrypted, it can be analyzed by hidden attackers. Let us assume that it is encrypted: It is impossible to see what users actually are doing with the service. From the outside it would seem to be a success.

Unfortunately, what happens when an attacker manages to get hold of the source code to the service? The concealed functionality becomes known, and there is no more deception.

(4) Concealing Functionality in a Distributed Application The fourth and last example suffers from the same problem. It is not necessary to get the source code either, if you have people skilled in reverse engineering.

To some degree, concealed functionality suffers from the security by obscurity syndrome. If it becomes known, the game is over. Interestingly, the same description (security by obscurity) has also been applied to passwords. The acknowledged difference between successful and flawed mechanisms that use this concept is the possibility of changing the 'obscured' object. A password can be changed without problems, and is therefore usable, while an algorithm that relies on the same cannot be modified.

[6] This, of course, assumes the attackers have achieved such a level of access that they can capture traffic.

This can be taken for an argument in support of concealed functionality if it is configurable, ie it is relocatable and the sequence of events that triggers it is changeable.

10.4.4.1 Indicators

We are hoping to conceal all indicators. Easy in theory, but not in practice.

10.4.4.2 Detection and Response Level

There is in fact neither detection nor response used with this deception. As such, it is completely static.

10.4.4.3 Demands on Knowledge of Target

Concealment is a universal effect working on all targets and places no demands on knowledge of the target, if the indicators are masked properly. In theory different targets can have different threshold levels for detection, but we disregard this.

10.4.4.4 Deception Revelation

If the existence of the deception is known, attackers might start looking harder. If the concrete nature of the deception is known, the game is up.

10.4.4.5 Counterdeception Resistance

Counterdeception, in this case, would be pretending to be ignorant of the concealed entity when its existence and whereabouts are known. If the attacker acts in a way consistent with him not knowing, there is no way of detecting this. Due to this, we can say that counterdeception, if attempted, is very likely to succeed. This can be dangerous if the attacker is merely waiting for the ideal time to strike.

10.4.4.6 Feedback

Feedback, in this case, is not necessary beyond keeping an eye on the information flowing to those who are in a position to get indicators revealing the concealed entity.

10.4.5 Conclusion

Concealing COAs is difficult in a world where exhaustive searches and hidden observation is possible for attackers.

10.5 Scenario 3

Still retaining the same strategic objective (keep attackers out), we will try another COA map action, that of confusion.

10.5.1 COA Map Consideration

We have tried removing COAs from the real map and concealing COAs in the attacker's map. The next action in line is that of confusion, making the attacker uncertain about what he can do, and the consequences of those actions. See figure 10.1.

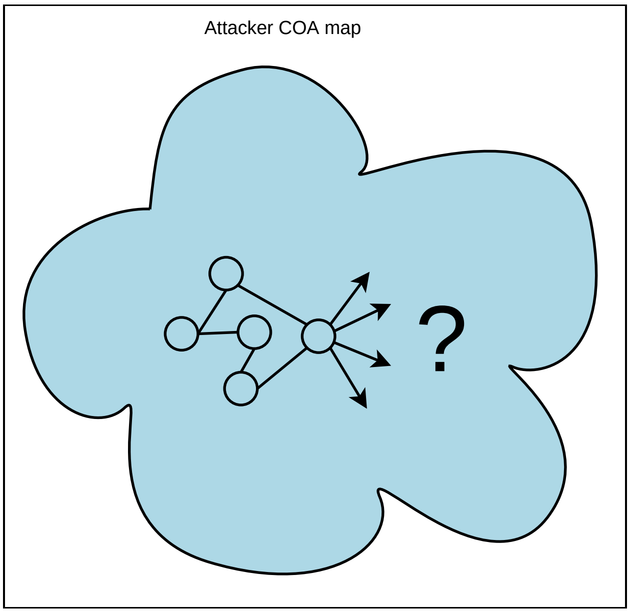

Figure 10.1: *Attacker uncertainty of possible options.*

10.5.2 The Deception

The deception objective is to make the attacker uncertain and confused about what he can do. We do this by misleading his tools to show a false picture of the computer environment.

We can think of at least three different sources for confusion: (1) Present ambiguous information (2) present logically inconsistent information (3) present a mixture of truth and lies, forcing the attacker to separate them. If we assume that an attacker has managed to penetrate the computer network, he will be able to see real traffic flowing between real hosts. Whatever we do, this truth will be visible at the same time as our deceptions.

Therefore we choose number three, presenting a mixture of truth and lies. We do this by saturating the computer environment with false hosts and false traffic. The attacker must be able to separate between them in order to advance.

10.5.3 The Assumption Chain

(1) Strategic objective: Keep attackers out.

(2) Deception objective: Remove COAs leading to further system penetration, by confusing the attacker.

(3) Target belief: Ambiguity; not knowing truth from lies.

(4) Target perception: A realistic environment of hosts and traffic.

(5) Portrayal: Create false hosts and traffic.

10.5.4 Analysis

In this instance we are misleading the tools of the attacker, in order to achieve confusion about the computer environment. In terms of assumptions, this is a different class than deceptions attempting to mask by concealment.

For this to work, we must know what characteristics the attacker looks at when building up an impression of the computer environment. These characteristics must be in concordance with the attacker's expectations of a real environment.

What does an attacker look at? We can draw on experience with regards to other attackers, put ourselves in the attacker's shoes, or actually observe what the attacker does. Unfortunately, the latter information comes too late, since there is no point in changing indicators that the attacker already has seen, except for improving the deception in the next round. There is also a question of what we are observing: We can perhaps see the contents of the packets going to and fro, without knowing how the attacker utilizes this information.

It is nearly impossible to ascertain if the attacker is deriving indicators that we do not know about. If he captures traffic for a long time, he can begin analyzing the types of communications with regard to sequences, typical usage patterns, replays and so on.

10.5.4.1 Indicators

The complexity of a deception increases the instant we do anything else but masking. Although we are aiming to confuse the attacker, we do so at a high cognitive level. There are no visual tricks that confuse, but false entities and characteristics in the computer environment that creates uncertainty of COAs and consequences.

In order to be successful, all indicators that an attacker can see or derive from false entities must be faithful to the real system. This means we must understand the tools he uses to maneuver in the computer environment.

10.5.4.2 Demands on Knowledge of Target

We assume the target interprets the information he receives from his tools in such a way that he will be unable to separate truth from lies, or at least spend time and resources solving the problem.

10.5.4.3 Deception Revelation

Knowledge of the existence of deception will not make the attacker's problem any easier; he must still separate between truth and lies. If he identifies specific deceptions, he will probably avoid them.

10.5.4.4 Counterdeception and Feedback

Is the attacker confused by our deceptions? It is tempting to assume that if he is spending time interacting with a deception, this is so.

But try separating between the two cases: (1) The attacker is fooled, therefore he is spending time on a deception (2) the attacker is fooling us, therefore he is spending time on a deception.

From the feedback of his actions alone, it is impossible to separate between them. The best we can accomplish is, "it does not look like he has detected the deception.” Assuming more leads to dangerous grounds, if we let this affect our own actions.

10.5.5 Conclusion

In the instant we moved beyond masking, we saw that the assumptions that were needed to come true, in order to be successful, placed greater demands on how the attacker behaves, and how he interprets information returned by his tools. The possibility for counterdeception is great.

10.6 Scenario 4

We have now arrived at the only scenario that tries to move the attacker down a COA: We desire a specific action from the attacker. This is an example of misleading. Our strategic objective is to gain more information about the attacker, to make identification and capture more likely.

10.6.1 COA Map Considerations

In theory we are less interested in making the attacker go down a specific COA, than triggering the resulting consequences. However, if the consequences are tightly bound to the COA we might not have any choice in what we must get the attacker to do.

Ideally, we would make the alternative COAs on the attacker’s map all result in the desired COA being executed. This seems to be a very difficult thing to do, except when very special strategic objectives are given. The closest example would be the rerouting scenario, where every action performed by the attacker implies the execution of the “spending time and resources” COA.

To make an attacker go down a specific COA, we must alter his impression of the consequences that follow. Based on our understanding of how the attacker makes decisions, and what consequences he considers important, we can try to convince him of the good consequences that follow if he goes down the COA, and the bad that follow if he does not. What we should choose depends on what affects the attacker the most: The carrot or the stick.

10.6.2 The Deception

This deception will have several components. One, we will use false and "interesting" traffic to draw the attacker to a specific account on a host, where false information has been prepared. The account name and password will be sent in the clear over the network. Two, the false information will contain false customer information. Three of the highest paying customers are false, and in on the game. We hope one of these "customers" will be contacted by the attacker.

10.6.3 The Assumption Chain

We break the situation down into two deceptions: First the attacker is lead to the false information, then he is to interpret it in a way that makes him react in a certain way.

(1) Lead attacker to false information.

- **(0) Overall strategic objective:** Gather information that can be used for identification and capture.
- **(1) Strategic objective:** Make attacker download false information.
- **(2) Deception objective:** Mislead the attacker to (1) try user/password on an account where false information is stored (2) make him find and download it.
- **(3) Target belief:** (1) This is a valid guest/password pair (2) we should use it (3) this information that we see is worth downloading.
- **(4) Target perception:** User/password pair sent by a protocol.
- **(5) Portrayal:** (1) False traffic giving away user/password (2) interesting document.

(2) Make attacker act predictably on false information.

- **(0) Overall strategic objective:** Gather information that can be used for identification and capture.
- **(1) Strategic objective:** Get information from contact with attacker.
- **(2) Deception objective:** The attacker contacts a false customer.
- **(3) Target belief:** Make the attacker believe it is in his interest to contact one of the top three customers.
- **(4) Target perception:** Sees false customer database.
- **(5) Portrayal:** Planted file with false information.

10.6.4 Analysis

This is the first scenario that deals explicitly with human-generated content. The computer domain is used both as an environment, when we lead the attacker to the false material, and as a channel of that material. What immediately stands out, is the increased number of assumptions that must be valid for us to succeed.

(1) Lead attacker to false information We assume that an attacker has the capability to pick out the username and password from all the other data he is gathering, and that he believes it to be authentic. He will use this information and log on to the computer. When he is there, he will look around, find the false customer database, and download it.

(2) Make attacker act predictably When the attacker has downloaded the information, he will read through it and, for some reason, contact one of the best customers. When doing this, he will impart some knowledge that makes it easier to identify and capture him.

On what grounds can we claim that any of these assumptions will be fulfilled? Rather than exploiting common human characteristics, we are making assumptions on how an attacker interprets a situation, and what COAs he will choose to follow. Recall the immense volume of information that was gathered before anything like this was attempted in JP 3-58.

Most of the first deception, "lead attacker to false information", rests on how we believe a computer attacker goes forth. This begs the question, how does an attacker commence? In what way does his methods depend on his motives? Does "interesting traffic" depend on what the attacker wants to achieve, or do we assume that all attackers seek full access, and therefore want any handle that can help?

If we pretend that the attacker in his desire for access has used the username and password we sent, he will be situated in the home directory of the false user. What now? If he just want the resources, he might ignore any data that is present. If he is curious, he might take a look around, say "what's this?", and download anything out of the ordinary. If he is targeting the data specifically, he will search for it. Is it possible to ascertain his motives by looking at his actions in this regard?

The customer database is found and downloaded. What does the attacker do with it? If he was only curious, who knows. The defender's main assumption is that anyone going to this length is a would-be competitor playing dirty, with the intention of stealing his most profitable customers. Is that a reasonable assumption?

10.6.4.1 Indicators

We are doing more complex things than masking, or replicating a real system. We are specifically planting falsehoods to (mis)lead the attacker down COAs of our choosing.

We create "interesting traffic" in order to make the attacker login on a specific account. An even more ambitious endeavor would be to make him focus solely on the traffic or host of our choosing, ignoring anything else. Such an attempt clearly demands intelligence on the attacker and his mission.

An aspect not relevant in the previous deceptions is now important: Indicator release. Why is the password and username sent in the clear, in the way it is? Is it normal procedure? Can the attacker become suspicious? It would seem to depend on how much it differs from real use of the system, and the analysis process of the attacker. This again demands more knowledge of the attacker than what is visible, if we want certainty.

10.6.4.2 Detection and Reaction Level

What makes the username and password be sent across the network? It can be risky to react automatically to the presence of the attacker. If the attacker, for the fifth time, sees the username and password fly by when he launches another probe, it looks a bit odd. Letting it happen periodically is also dangerous in case he monitors and analyzes the traffic. Nothing stands out as monotonous patterns.

Notice how the reaction level ultimately is on a much higher level than with the others: Now we are battling with the attacker's strategic objective. Do we deduce this from the attacker's actions, or do we assume that anyone penetrating this far belong to a specific class of attackers?

10.6.4.3 Demands on Knowledge of Target

Basic knowledge of how the attacker works in order make him see and use the username and password. Assumptions of the attacker's strategic objective in order to know what he will download, and how he will use it.

10.6.4.4 Deception Revelation

It if is revealed that a deception of this kind is going on, the attacker will probably become highly suspicious of everything, and the game is up.

10.6.4.5 Counterdeception and Feedback

There are two aspects of counterdeception here: Is it possible, and what is the consequence of it.

As always when operating in the computer domain, the possibility for counterdeception is high. The nature of feedback makes it impossible to separate between accepted and rejected deceptions. We see what, not why. This is not good enough to assume anything with a high certainty: The feedback is consistent with a multitude of theories. This tendency to take observations consistent with the prevalent or desired hypothesis as verification[7] is a fallacy, seen even in the intelligence community (Heuer 1999).

When it comes to the consequence of counterdeception, it depends on what is done if the deception is considered successful. If resources are put in motion, they would be wasted. If we are facing a sophisticated attacker, he could trigger reactions from us.

10.6.5 Conclusion

We can perhaps argue for the first deception (attractive traffic) to be within the purview of computer specialists, but the last deception, predictable actions after reading false material, has nothing to do with computers. This is within the purview of psychologists.

10.7 Chapter Conclusion

We have tried in these four scenarios to avoid making psychological assumptions about the attacker, but this became difficult very quickly. While other deceptions might avoid some of these assumptions, it seems there are few, if any, deceptions above the most simplistic that achieves guaranteed effects on the human attacker. This is to be expected, considering the cognitive level of computer deceptions. As such, we do not see option three being a viable alternative for the definition of computer deception.

Another lesson learned here, is the limited value of scenarios that do not rely on anything other than guesswork. Without attacker profiles, actual feedback or intelligence, we have no data that can aid us in selecting correct hypotheses with regards to attacker reactions or motives.

[7] As opposed to seeking refutation.

Chapter 11

Conclusion and Further Work

> **Focus.** The deception must **target the adversary decision maker** capable of taking the desired action(s). The **adversary's intelligence system is normally not the target.** It is only the primary conduit used by deceivers to get selected information to the decision maker.

(Chairman Joint Chiefs of Staff 1996, principle 1, emphasis not added)

Eons ago[1], we defined the objective of this thesis: To build a descriptive theory that could be used to understand the concept of computer deception. We looked to five different deception paradigms for aid in this, and found many deception aspects: Techniques, principles, planning and execution processes, the workings of the human mind, the nature and structure of deception itself. Based on this, we first sought a definition of deception that avoided the inconsistencies between the paradigms, and yet encompassed the vital concepts of each. Our result was:

Deception is the exploitation of a cognitive structure by the inducement or inhibition of state change, contrary to the intent of the design, operation, or conscious will of the entity.

The definition, however, did not tell the whole story. Equally important was the realization that deceptions are never executed for their own sakes, but for the fulfillment of strategic objectives. We emphasized this by using the Assumption Chain, a chain that shows how we believe consequences will flow from our portrayal, have effects on a target and support a strategic objective. This chain is why we are in business, our raison d'étre. We seek the achievement of the whole chain, and if there are any links in the chain that fail, the whole endeavor fails. This should be remembered in any type of deception.

[1] That is, section 2.2

The natural step, after defining deception, was to do the same when computers were added to the mix. But there we came up short. Instead of one logical definition revealing itself, we found three: Deception *of* computers, *with* computers, and a curious mixture where the former would imply the latter. To make a choice at this point would be completely arbitrary, so we set out to build a theory that could shed some light on this issue.

We started by taking the Domain Model as the backdrop for our own model of an attacker, a defender, and the computer system by which they interact. After defining an attack process and a defense process in this framework, we instituted COA maps and looked closer at the flow of information. By this we determined where the end effects that we really sought were induced or inhibitied.

We found that in solely targeting a computer entity, we are fighting with the human designer. In such a case we can ignore any human attacker. But for defense, it was far more common to decieve the computer entity in order to decieve the human operator. This implies that the correct starting point for computer deception is psychology. Any deviation from this by a computer specialist would place the burden of proof on him: Prove that the deception is only within the realm of computer science. Alas, we tried this in the scenarios, but had to admit defeat. It did not take long before we ended up with conjectures of the likes of "when the attacker sees X, he will do Y". Is this not a psychological assumption, are not attackers[2] part of the race of Man, which is imbued with free will?

Unfortunately, the average computer specialist refuses to acknowledge this. Blinded by the technological factor, he fails to realize that whether you are masking mechanisms or fooling Nmap, you are still targeting the human being running the show. He is turned from the Assumption Chain, which he does not know exists, and remains satisfied with a program that performs some automagics, confusing the technological effect with the strategic objective. When he first acknowledges the existence of human beings in his paradigm, he barges in unawares into the terrain of others, tossing psychological assumptions around as if they flow from a horn of plenty.

To be fair, we are intentionally polemical. And you probably would object, "hold on. Did not Stoll deceive the LBL[3]-attacker with flair and success, by anticipating his actions?" That is true. Some attackers are deceived by dead Honeynets without content or activity. Many, perhaps most, use predictable methods. Some will also buy the London Bridge if you offer a discount. Do not underestimate human stupidity.

Computer specialists do have a reason for being involved when designing

[2] With the exception of spammers.

[3] Lawrence Berkeley Lab.

computer deceptions targeting humans. To exclude them is skewing the problem in the opposite direction. But they should remain within their area of expertise. For instance, when the computer domain is used as a channel of false information having ramifications in the external world we are beyond 'regular' computer deception. The role of the computer specialist in such a case should be restricted to consider the plausibility of information release, conveyance and placement within the computer system. This is to maintain the impression of normal computer system operation. To properly understand the consequences of making the information available to the target, we need a psychologist. If the information is related to a specialized field, we might require a specialist on doctrinal and operational procedures.

When the computer domain is treated as an environment, clearly the computer specialist knows more than a psychologist of the attacker's tools, methods, and computer related expectations. But this does not mean that the computer specialist is competent to judge how the attacker will reason or how he will act. Deceptions on high cognitive levels have weak causality and high uncertainties. This is why professionals like the military forces, who like to be successful in their endeavors, have large preliminary and ongoing intelligence gathering and target profiling phases. If you desire high certainty of success against unknown and competent attackers with unknown objectives and methods, you have a problem.

What does this mean for the viability of computer deception? That all is lost? By no means. But there are clearly limitations to what one can achieve by theorizing about likely consequences of deception techniques. When the hidden assumptions are made explicit, this becomes apparent.

In one way, our theory has done its work in aiding us settling this issue, at least to our satisfaction. But it was our intention that it should be useful for more than this alone. As a descriptive tool it does not bring about any normative rules for deception design, but it does shed light on many important aspects. If we would dare go out on a limb and give any guidelines, it would be to start on that which is most important: The strategic objective. From there, you must construct an explicit Assumption Chain. Consider closely what hidden assumptions might lurk about; do not believe that everybody else will act like you want them to. Fate, Chance, free will, they all are fickle.

There is always more work that can be done, both in the field of computer deception and with regards to the theory presented in this thesis. The proper way of dealing with the entry of psychology is by running controlled experiments, rather than just hypothesizing about attacker intentions by the use of observational mechanisms. And in fact, such experiments have been performed by Cohen[4]. If there is one glaring omission in this thesis, this would be it. Unfortunately, neither time nor brain power were sufficient for the inclusion of those experiments here.

Studies of the computer domain as such, and its characteristics, would also be valuable. Is it possible to say anything conclusive on types or classes of indicators, their creation or conveyance? How should channels be defined, in a more precise way? This would surely be of interest when designing deceptions.

When it comes to our theory, two things stand out. The first is better integration with the Domain Model, of which we barely scratched the surface. It would be interesting to see if incorporating more theory from that model could contribute to the descriptive capabilities of our theory.

The second is an expansion of the COA maps, which we unfortunately did not have time to do. We only covered hiding, confusion, and a basic attempt at misleading. Our notion is that to expand, one can take each of the techniques that we have seen, and add a property to the model based on the manipulation performed by that technique. The intention was to do this with all of the techniques in the catch-all group of "what to show and accomplish".

For instance, Dewar's "reinforcement of probable action" means strengthening the probability of a COA in the attacker cognitive partition, in the part that reflects the attacker's estimate of the defender's actions. "The Lure" is adding a COA with an estimated desirable consequence, in the attacker's estimate of his own options. "The Double Bluff" presupposes a state in the target that has tagged the information with true or false before the information has been read. This would not be directly reflected on a COA map, so we would have to expand with a set of entity states. It is our belief that most, perhaps all, techniques can be explained by adding features in this way.

That, however, must remain for another time.

The End

[4] See, for instance, (Cohen, Marin, Sappington, Stewart, and Thomas 2001) and (Cohen and Koike 2002).

An Interlude

Epilogue

"So it was a student, huh. Strange." Frank and Bubba are walking in the neighborhood.

"Not so. They are everywhere, you know." Bubba answers. "Turn right here."

"Weird. And for a class exercise. Didn't see that one coming." Frank watches as Bubba looks around, sees nobody, and walks over to a large grey case marked 'Apathia Telco - Carrying Your Voice'. He opens it and inside the case Frank sees a lot of wires. Bubba disconnects a small unit from one of the wires and closes the case again. "No point in bugging your phone anymore now, is it?"

Frank gapes. "Excuse me?"

"Always a good thing to have incriminating evidence on tape."

"But.. how long have you bugged my phone?"

"Oh, only since we heard that the Big Tee was hiring a room at your house. Did you know that he has quite the business going, by hacking systems for a price? We've had him in our sights for a long time."

"Really." They move on. "Where was it that you worked again?"

"Ah.. need to know, and all that, you know?"

"Well, I'll be."

"Precisely."

Down the road, they pass a newly renovated building, where two men are putting up a sign for 'Dorian's Delightful Delicacies'. "This isn't the sign we were supposed to put up, originally, is it?" One of them says. "No, had to change name for some reason. Don't know why."

Frank and Bubba look at each other.

"Naahh."

– The End. No, really. There's no more coffee. There's no more thesis, either. –

References

Alberts, D. S., J. J. Garstka, R. E. Hayes, and D. A. Signori (2001). *Understanding Information Age Warfare*. CCRP.

Bell, J. B. and B. Whaley (1982). *Cheating and Deception*. Transactions Publishers. Reprint 1991.

Bishop, M. and D. Frincke (2004, May/June). Guarding the castle keep: Teaching with the fortress metaphor. *IEEE Security and Privacy*, 69–72.

Bubblegum (2005). Bubblegum Proxypot. Year is date of last access. `http://www.proxypot.org`.

CERT Coordination Center (2005). CERT/CC Statistics 1988-2004. Year is date of last access.
`http://www.cert.org/stats/`.

Chairman Joint Chiefs of Staff (1996, May). Joint Publication 3-58: Joint Doctrine for Military Deception.
`http://www.dtic.mil/doctrine/jel/new_pubs/jp3_58.pdf`.

Cheswick, B. (1992). An evening with Berferd, in which a hacker is lured, endured, and studied. Usenix Winter 1992 Conference.
`http://www.cheswick.com/ches/papers/berferd.ps`.

Cohen, F. (1998). A note on the role of deception in information protection. *Computers & Security 17*(6), 483–506.

Cohen, F. and D. Koike (2002, May). Leading attackers through attack graphs with deceptions.
`http://all.net/journal/deception/Agraph/Agraph.html`.

Cohen, F., D. Lambert, C. Preston, N. Berry, C. Stewart, and E. Thomas (2001). A Framework for Deception.
`http://all.net/journal/deception/Framework/index.html`.

Cohen, F., I. Marin, J. Sappington, C. Stewart, and E. Thomas (2001, November). Red teaming experiments with deception technologies.
`http://all.net/journal/deception/experiments/experiments.html`.

Denning, D. (2000, May). Testimony before the Special Oversight Panel on Terrorism, Committee on Armed Services, U.S. House of

Representatives.
`http://www.cs.georgetown.edu/d̃enning/infosec/cyberterror.html`.

Department of Army (1988, October). *Field Manual 90-2: Battlefield Deception*. Department of Defence.

Dewar, M. (1989). *The Art of Deception in Warfare*. Newton Abbot, Devon: David & Charles Publishers ; New York, NY: Distributed in the United States by Sterling Pub.

Dunnigan, J. F. and A. A. Nofi (1995). *Victory and deceit: dirty tricks at war* (1st ed.). New York: W. Morrow.

Ekman, P. (2004). *Emotions Revealed: Understanding Faces and Feelings*. Phoenix.

Forsvarets overkommando (2000). *Forsvarets fellesoperative doktrine: Grunnlag (A)*. Forsvarets overkommando.

Forsvarssjefen (2003). Konsept for nettverksbasert anvendelse av militærmakt.

Gerwehr, S. and R. W. Glenn (2000). *The Art of Darkness: Deception and Urban Operations*. RAND.

Gregory, R. L. (1973). *Illusion in nature and art*, Chapter 2: The Confounded Eye, pp. 49–95. Charles Scribner's Sons.

Heuer, R. (1981). Strategic deception and counterdeception. *International Studies Quarterly 25*(2), 294–327.

Heuer, R. J. (1999). *Psychology of Intelligence Analysis*. Center for the Study of Intelligence, CIA. Year is date of Internet publication.
`http://www.cia.gov/csi/books/19104/index.html`.

Holz, T. and F. Raynal (2005a, March). Defeating honeypots: System issues, part 1. Date of last modification.
`http://www.securityfocus.com/infocus/1826`.

Holz, T. and F. Raynal (2005b, March). Defeating honeypots: System issues, part 2. Date of last modification.
`http://www.securityfocus.com/infocus/1828`.

Howard, J. D. (1997). *An Analysis Of Security Incidents On The Internet*. Ph. D. thesis, Carnegie Mellon University.
`http://www.cert.org/research/JHThesis/Start.html`.

J-7, J. (2001, April). Joint Publication 1-02: DOD Dictionary of Military and Associated Terms. As amended through 30 November 2004.
`http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf`.

Kettle Creek Corporation (2005). War, Chaos, and Business. Year is date of last access.
`http://www.belisarius.com`.

LaBrea (2005). LaBrea the sticky honeypot/IDS. Year is date of last access.
`http://labrea.sourceforge.net`.

Lambert, D. R. (1987). A cognitive model for exposition of deception and counterdeception. Technical report, NOSC.
`http://www.spawar.navy.mil/sti/publications/pubs/tr/1076/tr1076.pdf`.

NewScientist (2004). Ears play visual tricks on us.

Oudot, L. and T. Holz (2004a, September). Defeating honeypots: Network issues, part 1. Date of last modification.
`http://www.securityfocus.com/infocus/1803`.

Oudot, L. and T. Holz (2004b, October). Defeating honeypots: Network issues, part 2. Date of last modification.
`http://www.securityfocus.com/infocus/1805`.

Rowe, N. C. and H. S. Rothstein (2003). Deception for defense of information systems: Analogies from conventional warfare.
`http://www.au.af.mil/au/awc/awcgate/nps/mildec.htm`.

Rowe, N. C. and H. S. Rothstein (2004). Two taxonomies of deception for attacks on information systems.

Schneier, B. (2000). *Secrets & Lies: Digital Security in a Networked World*. John Wiley & Sons, Inc.

Spitzner, L. (2002). *Honeypots: tracking hackers*. Boston: Addison-Wesley.

Spitzner, L. (2003, 29 May). Honeypots: Definitions and value of honeypots. Date of last update.
`http://www.tracking-hackers.com/papers/honeypots.html`.

Stoll, C. (1989). *The Cuckoo's egg: Tracking a spy through the maze of computer espionage*. Doubleday.

Tan, K. L. G. (2003, December). Confronting cyberterrorism with cyber deception. Master's thesis, Naval Postgraduate School. Thesis Advisor: Neil C. Rowe. Second Reader: Dorothy E. Denning.

The Honeynet Project (1999, October). Founding date.
`http://www.honeynet.org`.

The Honeynet Project (2003, November). Know your enemy: Gen II Honeynets. Date of last modification.
`http://www.honeynet.org/papers/gen2/index.html`.

The Honeynet Project (2004, October). Honeynet definitions, requirements, and standards. Date of last modification.
`http://project.honeynet.org/alliance/requirements.html`.

Whaley, B. (1982). Toward a general theory of deception. *Journal of Strategic Studies 5*, 178–192.