

Improving the Effectiveness of Deceptive Honeynets through an Empirical Learning Approach

Nirbhay Gupta

*School of Computer and Information Science
Edith Cowan University, Australia
E-mail: nirbhaygupta@yahoo.com*

ABSTRACT

Over the last few years, network based intrusions have increased rapidly, due to the increase and popularity of various attack tools easily available today. Due to this increase in intrusions, the concept of network Honeypots are being developed, which can be used to trap and decode the attack methods of the malicious attackers. This paper will review the current state of honeypot technology as well as will describe the framework of how to improve the effectiveness of Deceptive Honeynets through the use of deception.

Keywords: Honeynet, deception, intrusion detection, attack intelligence

INTRODUCTION

In the last three years, the networking revolution has finally come of age. The Internet is purported to provide limitless possibilities and opportunities but on the other side it also increases the risks of malicious intrusions.

It is very important to understand and design some form of security mechanism to prevent or detect unauthorised access by users. These intrusions can be detected. Due to the increased attempts of intrusion into systems, the concept of 'honeypot' systems was developed. Honeypots are used to trap and decode attack methods used by the black hat community [Brenton, n.d; Klug, 2000; Spitzner, 2002]. Honeypots can provide a deceptive defence mechanism in which the attackers are deceived into believing they are intruding into a real production system. The correct deployment, monitoring and analysis of these systems helps in increasing our understanding of attackers' modes of operations and tools in details.

Deception can be referred as the state of being deceived or misled [Anonymous, 1998]. It can be considered as creation of false environment to fool or deceive people. Therefore, honeypot systems are meant for creating a false computing environment in order to keep away the attackers from the real network environment and entrap them in a false system.

This paper will examine the theory and background to the research into producing a deception based honeypot system. It is based on an ongoing Masters research project that has three main objectives these are

- 1.To improve the level of Deception presented to attackers in a Honeypot design.
- 2.To harden a deceptive honeypot and test its effectiveness using an empirical learning approach.
- 3.To improve the ability of a deceptive honeypot to gather attacks Intelligence.

HONEYPOTS AND HONEYNET CONCEPTS

Honeypot systems are systems whose main purpose is to be probed and exploited for vulnerabilities by hackers or black hat community. These systems are used to learn the attacker's moves and tools used to compromise systems. Honeynets are collection of multiple honeypot systems connected in a network to appear as a functioning network.

This network sits behind the firewall where all inbound and outbound traffic is captured and controlled. In a honeypot, we can have any type of system, such as Windows NT or Linux, to be used as honeypot. This creates a network environment that has more realistic approach to it for the attacker. All systems placed within the Honeynet are the mirror images of the standard systems placed within the internal network of any organisation. [Anonymous, 2002]

The success of honeypots is based on their simplicity. These systems are intended to be compromised by would be attackers. Any traffic, inbound or outbound, to honeypot is considered to be suspicious by nature. Because of its being simplistic in nature, honeypots have certain advantages and disadvantages [Anonymous, 2000; Sink, 2001; Spitzner, 2002]:

Advantages

- Data Collection: Honeypots collect normally a very high value of data. They provide a lot of useful information on attacker's movement within the system and their (attackers) interaction with the system in a quick and easy to understand format.
- Allows the "white hat" community to study exactly what attackers are doing without exposing systems and networks to additional risk that results from compromised systems.
- Honeypots provide deterrence.

Disadvantages

- They are worthless if no one attacks them.
- Honeypots can be used as a launching platform to attack other machines on different network.

A highly controlled network is created. Within this network honeypots are placed and then the activities of attackers are monitored, captured and analysed. Any traffic to honeypots is suspicious in nature. Such traffic to honeypots could be port scan or probe of the network. In case if there is an outbound traffic from the honeypots, then the system has been compromised.

"Creating and maintaining a successful Honeynet depends on two critical elements: data control and data capture." [Anonymous, 2002, p20] Data Control is the inbound and outbound control of data. Once the honeypot within the Honeynet is compromised, the administrators have a responsibility to ensure that the honeypots are not used to compromise other systems within the network or other outside networks. A firewall can act as an access control device for the data control. All inbound and outbound data should flow through the firewall. All inbound and outbound traffic must be controlled in an automated fashion, without the attacker getting suspicious. Therefore, a transparent firewall is used with three main rules:

1. Anyone can initiate a connection from the Internet to the Honeynet. This will allow the attackers to scan, probe and exploit systems on the Honeynet.

2. The firewall controls how the honeypots can initiate connections to the Internet. If no outbound traffic is allowed from the compromised Honeypot to the Internet, the attacker will become suspicious and may leave the network. Therefore, some limited outbound connections should be allowed. The Honeynet Project [Anonymous, 2002] found that five to ten outbound connections within a 24-hour period should be sufficient to deceive the attackers.
3. The Honeynet and the internal network should not have any direct communication. The internal network is used for critical data collection for all data generated by the attacker's activity. If the Honeynet is compromised, it should prevent the ability to communicate with the internal network and modify or delete any data collected.

Therefore, to control and contain data flow to and from the Honeynet, access control must be wisely used to separate the Honeynet from other networks.

Data capture is the collecting of all activity that occurs within the Honeynet, including both the network and system levels. Data capture is an essential component for the success of a Honeynet system. According to the Honeynet project, it is ideal and useful to capture data in layers. By using a layered approach the multiple layers of information can be used together to aid developing a richer understanding of the attacker's tools, tactics and motives.

It is very important to remember that none of the captured data should be stored on the honeypot locally as it is easily compromised. All data captured on the honeypot should be transferred to a secure location on the network where no other communication is allowed with the Honeynet.

DECEPTION USED FOR DEFENCE

The honeypot used for this research will vary greatly in that it will be a single host server that will masquerade as multiple servers leveraging and building upon 2 key pieces of deception based software namely honeyd [Spitzner, 2002] and DTK [Anonymous, n.d]. This allows the researchers to use a singular RedHat Linux box to effectively mimic a Honeynet using these tools.

Deception Services are specially designed to listen on an IP service port and respond to network requests. They can be used to emulate services like for example, Sendmail. When an attacker connects to the honeypot, he or she may receive a banner that identifies the service as being some version of Sendmail. If the attacker is fooled by the deception, he or she may try to gain access to the system. Such actions will provide the administrator the logs of attackers' movement. The best example for Deception Service could be Fred Cohan's Deception Tool Kit.

Deception Services too have some problems. Firstly, it is difficult to emulate a service that can fool the attacker. Secondly, deception services can only collect limited amount of information. It records the initial attempt of attack, but nothing more than that. Finally, on a theoretical basis, deception service may have some form of vulnerability that could give attacker unexpected access to the system.

According to Cohen (2000), the deception for network defence is the model of an enemy who believes that information systems are vulnerable and has finite resources with which to attack them. Attacker would intelligently try to explore the vulnerabilities and will try to exploit

them. Even though there can be well defences in place but the attacker always believe that the imperfection always exist somewhere in the system. Therefore, we can make use of deception that induces the belief of attacker that vulnerabilities exist and can be exploited. Although such weaknesses do exist in every system so deception should be used to try to defeat the attackers attack process.

An attacker can be technically competent and if can properly identify the vulnerabilities, can lead in launching a successful attack. Thus, the defensive deception process is essential and must be oriented in defeating the attacker's intelligence process.

The ideal defensive deception allows an attacker to process to proceed in a manner that the attacker's intelligence effort appears to meet expectations without being able to recognise the deception in place. The results of attacker's actions should be that of adding to a defender's advantage.

IMPROVING HONEYPOT DESIGN

Earlier honeypot systems were based on the idea of placing a small number of attractive targets in locations where they are likely to be found and would draw attacker's attention towards them. However this defence mechanism offered a little as it only consumed a small portion of the overall intelligence space and has little effect on altering the characteristics of the typical intelligence probe.

The original Deception Tool Kit (DTK) by Cohen provided the low probability of detecting the deception and the extreme localisation of deception under previous honey pot systems. Under DTK, deceptions are spread among the normal systems in a network in such a way that unused services on those systems are consumed with deceptions. This leads to two effects:

- 1.It spreads the deceptions over a large portion of the IP/port address space
- 2.It increases the percentage of deceptions in the environment, thus increasing the likelihood of an intelligence probe encountering a deception rather than a vulnerability.

Increasing the size of the Search Space and the Sparsity of Real Vulnerabilities

Cohen (2000) proposed that in order to improve the situation for the defender, increase the intelligence workload by increasing the size of the search space. Configuring one Ethernet card as the host for numerous IP addresses can do this. Each of these can optionally have their own MAC address as well. This technique can be applied for deception by filling a large address space that would normally be sparsely populated to dazzle the attacker. Combined the use of the honeypot system developed by Provos it is possible to further enhance the deception by providing bogus TCP/IP fingerprints to the attacker. This aids in the deception by making the system appear at a TCP/IP stack level to also correspond to the deceptive system.

This approach of offering a wider range of systems will increase the workload of the intelligence effort in determining which of these systems are legitimate and which are not. Cohen has used DTK to populate more than 40,000 IP addresses with false services. This was highly effective in increasing the intelligence workload, in increasing the time to attack and decreasing the odds of certain classes of intelligence probes going undetected. The effect should be that of an increased attack detection window for the honeypot system.

ENHANCING THE QUALITY OF DECEPTION

The author examined a few deceptive protection techniques through using Dunnigan and Nofi's classification scheme [Dunnigan, 1995] (i.e. concealment, camouflage, false and planted information, ruses, displays, demonstrations, feints, lies and insight) and how this may apply to the study and aid us in develop a more deceptive deception.

Camouflage

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. For example, use of scripts like those in the DTK that make services such as SMTP belong to a different operating system. There are various difficulties associated with camouflage. One of them is the use of automation to gather information about all of the available information on the Internet. These search engine or crawlers provide such a thorough search of the environment that camouflage is nearly impossible to maintain. Even the links between different identities may be derived from analysis of linguistic characteristics and other patterns in postings. Camouflage may be successful in preventing attackers with limited skills and resources but it is very difficult to maintain against skilled or well-resourced attackers. One of the key things this study will do is look at ways of improving the baseline scripts provided with DTK to provide a better camouflage for the deception system.

False and planted information

False and planted information is based on the concept that attackers depend for their success on gaining information about their victims and that by providing inaccurate information, successful and undetected attack becomes far more difficult. This involves understanding the intelligence capacities of the attacker and finding ways to cause their intelligence operations to go away from the desired path. The use of the honeyd system will help in that it will provide an appropriate TCP/IP fingerprint back to the attacker for the modelled system.

Displays

The main purpose of displays is to make attacker see what is not there. For example, creation of a fictional computer security organisation, a set of policies, procedures and other protective techniques, which appear to be in place but in fact, are not. The deception systems modelled in the study will where possible provide bogus documents to provide this sort of deception to an advanced attacker.

PROBLEMS WITH DTK AND HONEYD

The Deception Tool Kit (DTK) is a publicly available deception system if enough people start using DTK, a lot of sophisticated attackers may be eliminated as they will test for its existence on a system. One of the indicators of DTK is IP port 365 - a deception port. Port 365 indicates whether the machine you are trying to connect is running any deception defence. Normally attackers will first try to probe this port, and if the deceptive defence is in place it will help in keeping most of the attackers away from the network.

The study hopes to overcome one of the major drawbacks of using DTK and honeyd in that competently deployed systems these days use a small range of services per server. This reduces the ability to dazzle the attacker with a multitude of ports because this system automatically becomes an evident deception due to the ports that are displayed. The study will focus on improving the richness of deception for SMTP, SSH, Telnet, POP3 and FTP services. Then allowing a system to appear with a reduced number of services that are highly deceptive in nature and hopefully will prove effective at increasing attacker workloads.

PROPOSED RESEARCH FRAMEWORK

The framework for this research will involve the use of empirical learning as the main guiding principle. The research will provide a range of quantitative data that can be used to improve deception services on the system.

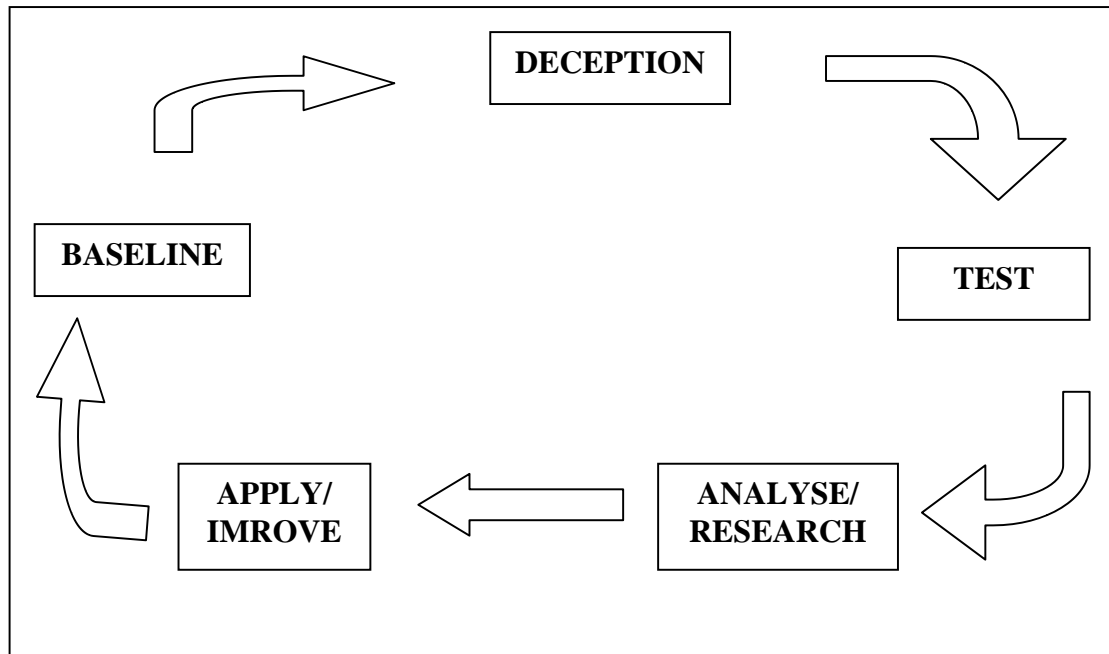


Figure 1: Conceptual Framework Diagram

The approach will be to make and test baseline deception system. The baseline system will be configured to present as a standard corporate server configured with SMTP, POP3, SSH, Telnet and FTP services through the use of DTK [Anonymous, n.d] and honeyd [Spitzner, 2002] deception daemons in default mode. The system will utilise a Linux RedHat 7.3 system as a base server. Nessus and other vulnerability scanners will then be used to examine how well the baseline system fools these systems. Then a group of pre-selected hackers would be required to attack the improved system. Selected attackers would be required to attack the system with DTK installed on it to test the effectiveness of the DTK. Most probably the hackers would be the computer security students present in the School of Computer and Information Science of Edith Cowan University, Perth. The reason to choose this group of students is because they will be readily available to approach. To select this group of student, a test of computer security literacy will be formed and conducted to determine their level of knowledge about operating systems and hacking tools and tricks.

The default deceptions will then be enhanced to better deceive these vulnerability tools.

Then the improved systems will then be probed via vulnerability scanners and the pre-selected hackers to try to determine if the deceptions have improved.

CONCLUSION

This research is attempting to provide richer deception through the use of an empirical learning approach to attacks and probes on systems. This research can be regarded as evolutionary: findings of one phase will be the focus and design of next phase. After testing the systems

with attacking tools and improving them based on the results obtained from pre-selected hackers attacks, systems will be further tested to determine if the level of deception has further improved. Deceptive honeypots coupled with appropriate intrusion detection systems and firewalls may provide a means for providing much need forward intelligence about attackers and give defenders an increased reaction and countermeasure time window.

REFERENCES

Anonymous (1998). Dictionary.com/deception. **2002**.

Anonymous (2000). Honeypot Effectiveness Study, Global Integrity Corporation.

Anonymous (2002). Honeynet Project, Available at URL: <http://project.honeynet.org> **2002**.

Anonymous (2002). Know Your Enemy, Addison Wesley.

Anonymous (n.d). Deception Tool Kit, Available at URL: <http://www.all.net/dtk> **2002**.

Brenton, C. (n.d). Honeynets, Dartmouth College Institute for Security Technology Studies (ISTS).

Cheswick, B. (n.d). An Evening with Berferd, AT & T Bell Laboratories.

Cohen, F. (1998). "A note on the role of deception in information protection." Computers & Security **17**(6): 483.

Cohen, F. (2000). "A mathematical structure of simple defensive network deceptions." Computers & Security **19**(6): 520.

Dunnigan, J. (1995). Victory and Deceit - Dirty Tricks at War, William Morrow and Co.

Holcroft, S. (2002). Design Of a Default RedHat Server 6.2 Honeypot. **2002**.

Klug, D. (2000). Honey Pots and Intrusion Detection. **2002**.

Sink, M. (2001). The use of Honeypots and Packet Sniffers for Intrusion Detection. **2002**.

Spitzner, L. (2002). Honeypots. **2002**.