

# Honeypots Deployed

Intrusion detection systems, commonly known as IDS are used in modern everyday networks. One of the most popular forms of IDS in the Honeypot. Honeypots are decoy systems basically, that are used to lure in possible attackers that threaten secure networks. Honeypots can be used on wired or wireless network technologies. In this paper I hope to inform of how some honeypots are deployed.

According to the Cyber and Homeland Security Research and Development study done at Dartmouth College on honeypots, one way of deployment of a honeypot is through deception services.

Deception Services are applications that are specifically designed to listen on an IP service port and respond to network requests like some other application. For example, a deception service could be configured to emulate Sendmail. When a perpetrator connects to port TCP/25 on the honeypot, they receive a banner that identifies the service as being some version of Sendmail. If the perpetrator is fooled by the deception and a Sendmail attack is part of their arsenal, they may attempt to gain access to the system through what they think is the Sendmail service. This allows the system administrator to log the particulars of the attack in an effort to safeguard their other systems that may actually be running Sendmail. This log may also be submitted to CERT, the vendor, or law enforcement for review in order to ensure that a proper fix can be created.

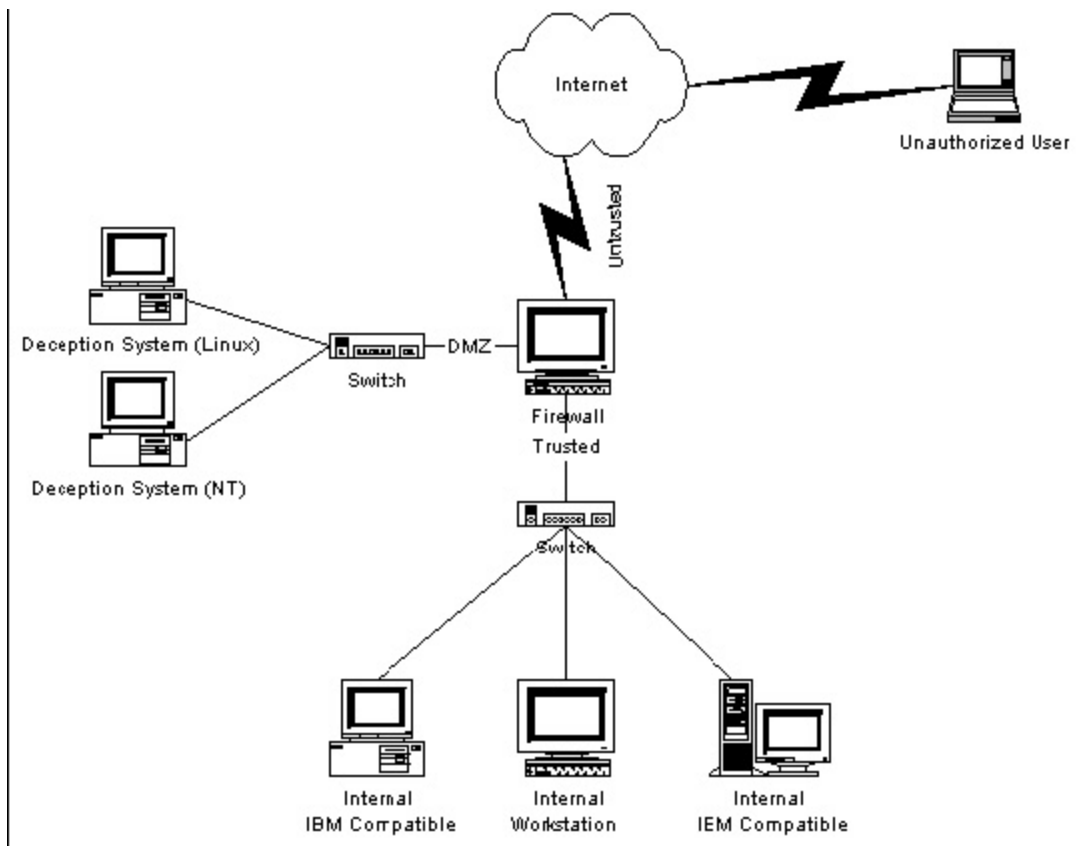
Deception services are probably the earliest form of honeypot deployment, Fred Cohen's Deception Toolkit being an excellent example.

There are a number of problems with using deception services for honeypot deployment. To start, it can be extremely difficult to emulate a service to enough of a level that a perpetrator will be fooled. For example, the perpetrator may try a variety of e-mail addresses and check for expected responses, as well as try a number of control commands. Unless the deception service is capable of passing this level of advanced testing, the perpetrator may become wise to the trap and never actually launch the attack.

Another problem is that a deception service is only capable of collecting a limited amount of information. You see the initial attack, which attempts to gain root access to the machine, but you see nothing else. It could be useful to see what the perpetrator would do if the attack were actually successful. A successful attack could yield additional information, such as other systems compromised by the perpetrator, clues to the perpetrator's identity, or even some of their tools. Since the deception service should not provide the perpetrator access to the machine, additional forensic information cannot be collected.

Another method of deployment of a decoy system refers to placing the system on a Demilitarized zone which is described vividly by Kellep A. Charles, CISSP, in the article *Decoy Systems*.

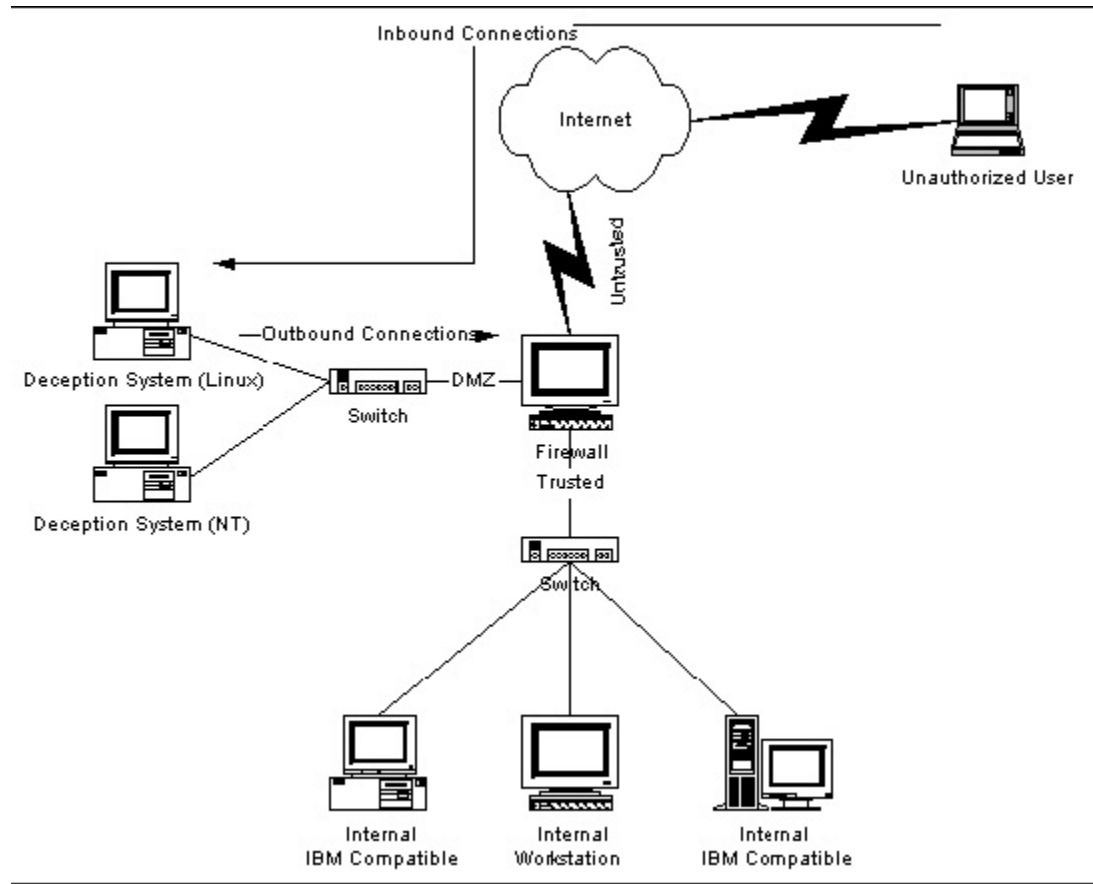
Decoy systems placed on a DMZ to lure attackers away from the internal trusted network assets provide many benefits, as illustrated in Figure 1. An access control rule-set on the system firewall can be less stringent on the DMZ network where the decoy systems reside. When the unauthorized user performs scans to locate system vulnerabilities, the decoy systems on the network would reply and move all focus away from the trusted network resources.



**Figure 1 – Decoy Systems on a Separate Network**

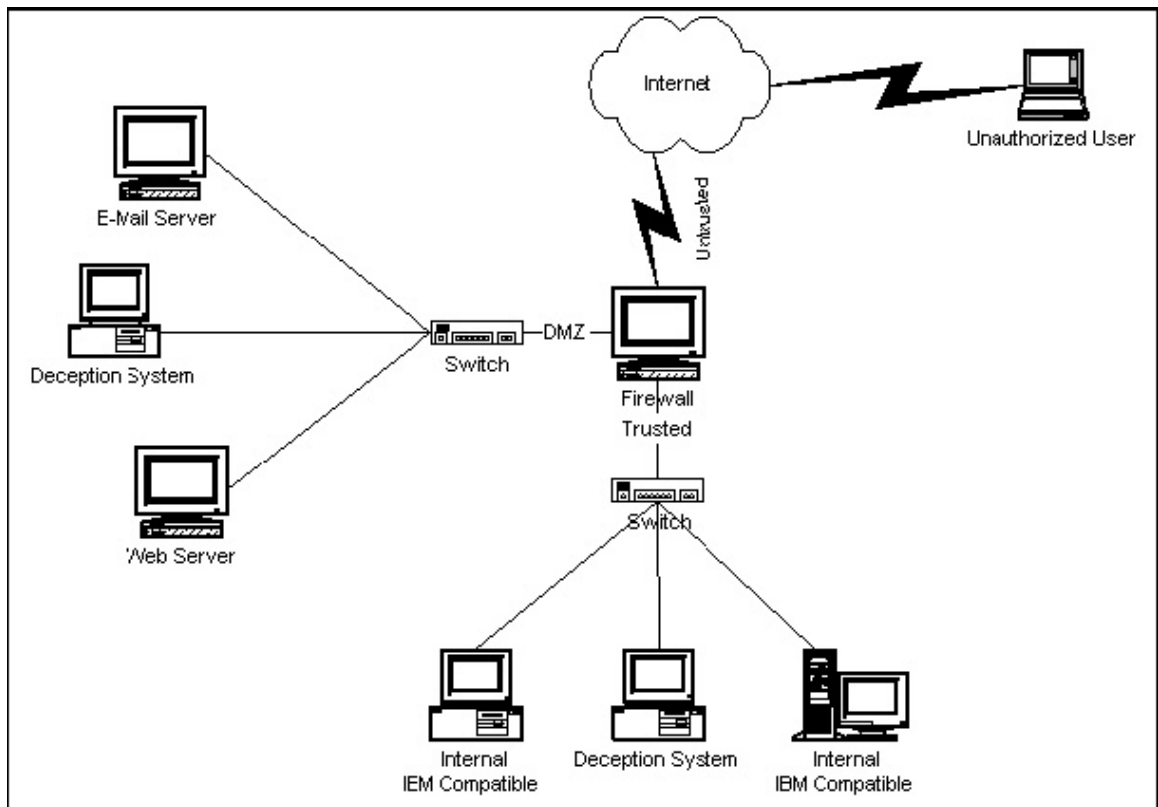
Once an unauthorized user compromises the systems on the DMZ, special data control mechanisms are put in place to prevent further harm to other information systems. The access control rule-set on the firewall allows data to enter the DMZ,

but restricts certain data to depart from the DMZ. This prevents the unauthorized user from launching further attacks to other information systems. Figure 2 depicts an example of data control flow on a network infrastructure, using a DMZ concept to deploy decoy systems.



**Figure 2 – Data Control of Decoy Systems on a Separate Network**

The minefield principle of deploying decoy systems involves placing decoy systems with other production information systems on a trusted network and trusted DMZ network. This is depicted in Figure 3. Often the decoy systems will have an appealing server with names such as "Primary Mail Server" and "HR File Server" and a lower IP address for quicker vulnerability scan detection.



**Figure 3 – Minefield Approach to Decoy Systems**

The final deployment is actually a special one. It involves creating a honeypot for a wireless network. It is the method of creating a fake access point to lure potential attackers from the network. The deployment is described in the Security Focus article, *Wireless Honeypot Countermeasures*.

If you remember the movie called War Games, the young adolescent was using a modem on the phone line to scan remote phone numbers and find open lines like BBSes. This activity was called wardialing, and by transposition in the wireless world, people talking about wireless scanners or wireless listeners as wardriving, or even warwalking. Wardrivers try to find open networks. A good first idea to delude those potential intruders would be to simulate as many fake networks as

possible for them to lose time and patience. Targeting one network is an easy task, whereas dealing with a cloud of targets could be more difficult.

This proof of concept was done with a tool called FakeAP [\[ref 5\]](#), free software distributed under GPL by the guys from Black Alchemy during the Defcon X. This tool can send specific wireless network traffic to fool basic attackers. As a wardriving countermeasure, it generates 802.11b beacon frames as fast as possible, by playing with fields like BSSID (MAC), ESSID, channel assignments, and so on. This trick is easily created by playing with the tools used to manage a wireless card (under Linux, that's like manually playing with: `iwconfig eth1 ESSID RandomSSID channel N...`). A remote, passive listener should then see thousands of fake access points! To quote the web site of the authors: "*If one access point is good, 53,000 must be better.*" The idea behind this simple tool was quite good when it was first released, and you could even detect NetStumbler users by looking at 802.11b probe requests/responses. Whereas now, most updated tools can advise the attacker that the detected access points are unusually strange, such as these cases where no traffic is generated on the found networks.

Figure 4, below, indicates a NetStumbler scan on one of these honeypots:

MAC	SSID	Name	Ch.	Vendor	Ty.	En.	SN	Sign.	No
0000CE992FA4	TrackingHackers		10		AP		30	-56	-86
0000CE1DDBD2	CanSecWest		10		AP			-54	-86
0000CE69BFB0	Berbus		10		AP			-54	-89
0000CB9C890	CanSecWest		10		AP			-56	-88
0000C193CF9	Moutane		10		AP			-54	-90
0000CE991A75	SSTIC		10		AP			-54	-87
0000CE3EC028	SSTIC		10		AP			-54	-91
0000C1BDF30	SSTIC		10		AP			-56	-85
0000CE43B002	Rsteck		10		AP			-53	-89
0000CBF3100	Moutane		10		AP			-54	-91
0000CE082274	Moutane		10		AP			-56	-86
0000C2CA061	MiscMag		10		AP			-55	-88
0000CEFB05A3	MiscMag		10		AP			-55	-91
0000CE2EBED5	Moutane		11		AP			-57	-89
0000C72DA69	Moutane		11		AP			-58	-87
0000CED52D36	Moutane		11		AP			-60	-87
0000CF5FCF18	Berbus		11		AP			-60	-87

**Figure 4: NetStumbler scan on a FakeAP honeypot**

There are many other forms of deployment of a honeypot or decoy system. Which ever method is used the system should always be a benefit for the network that is installed on. The administrator should have it set that it is able to be learned from for the methods of making the network more secure. According to *Defeating Honeypots*, a two part article from Security Focus, a honeypot should be strong enough to be hacked but not detected as a decoy system.

## References

- \*Brenton, Chris 2004. Dartmouth College *Institute for Security Technology Studies*. Cyber and Homeland Security Research and Development. Honeynets. [Electronic Version]. From <http://www.ists.dartmouth.edu/classroom/honeynets.php>
- \*Charles, Kellep A. Decoy Systems: A New Player In Network Security and Computer Incident Response. *International Journal of Digital Evidence*, Winter 2004, Volume 2, Issue 3 [Electronic Version]. [www.ijde.org](http://www.ijde.org)
- FakeAP tool. Black Alchemy retrieved April 1, 2006 from <http://www.blackalchemy.to/project/fakeap/> .
- Security Focus. (2005, March 23). Defeating Honeypots: System Issues, Part 1. Holz and Raynal. Retrieved April 1, 2006 from <http://www.securityfocus.com/infocus/1826> .
- Security Focus. (2005, April 6). Defeating Honeypots: System Issues, Part 2. Holz and Raynal. Retrieved April 1, 2006 from <http://www.securityfocus.com/infocus/1828> .
- Security Focus. (2004, February 13) Wireless Honeypot Countermeasures. Oudot, Laurent Retrieved April 1, 2006 from <http://www.securityfocus.com/infocus/1761> .