Global Information Assurance Certification Paper

The Use of Honeypots and Packet Sniffers for Intrusion Detection

Michael Sink, P.E.
GIAC Security Essentials version 1.2b
April 15, 2001

Introduction

Within the realm of computer security, a honeypot is a computer system designed to capture all traffic and activity directed to the system. While honeypots can be set up to perform simple network services in conjunction with capturing network traffic, most are designed strictly as a "lure" for would-be attackers. Honeypots differ from regular network systems in that considerably greater emphasis is placed on logging all activity to the site, either by the honeypot itself or through the use of a network/packet sniffer.

The use of honeypots, especially in conjunction with packet sniffers is at the center of an increasingly divisive debate over the use of such tactics and subsequent information collection procedures. On one side of the debate, it is held that honeypots are at best, a mechanism to provide information on attack protocols that are already widely known and unsophisticated (e.g. script kiddies), and at worst, are a form of electronic wiretapping and entrapment. As such, it is believed the concept is unethical and potentially illegal. Moreover, the use of data collected in such a fashion may or may not be admissible in court, rendering the use of honeypots impractical for the security specialist.

On the other side of the debate, it is believed that honeypots are a valuable source of information on attacks and specifically the levels of sophistication used by attackers. Moreover, it is held that since entrapment involves coercing someone to commit a crime they would otherwise not have committed, the typical honeypot cannot be considered entrapment.

It is important to acknowledge the political and legal realities of using a honeypot and to keep the framework of this debate in mind when considering its use. Having said that, this paper centers on the technical aspects of such a system. This paper contains a discussion on definition of, objectives for, and the use of honeypots, the advantages and disadvantages of such a system, the various types of honeypots, and the use of a honeypot and sniffer detection system.

Definition: What is a Honeypot?

A honeypot is designed to look like something an intruder can attack to gain access to a given system (i.e., bait). Typical examples of honeypot systems are:

- Installing a machine on a network with the express purpose of logging all access attempts.

- Installing an older unpatched operating system (such as Win NT 4 with IIS 4.0). A standard intrusion detection system (IDS) or sniffer is then used to log hack attempts directed against the machine, and possibly track what the plans the attacker has once the system is compromised.

- Install special software for the purpose of tracking attackers moves.

Honeypot Objectives

- The honeypot system should appear as generic as possible. This is one reason sniffer based systems are used in conjunction with honeypots as discussed below.

- You must design the honeypot such that an attacker cannot easily use the honeypot as a launch point for further attacks against networks.

- The honeypot should appear to contain genuine and interesting information, to entice attackers to stay on the site while you track their moves.

- Honeypots may be set up in front of a firewall, in the DMZ, or behind a firewall. In general, it is best to set up a honeypot behind a firewall for the advantages as discussed below. In addition, the closer the honeypot is to actual servers (which are likely behind a firewall), the more likely it is to tempt intruders.

Advantages of a Honeypot

- A distinct advantage of a honeypot system over a standard IDS is that a honeypot, unlike an IDS, will trip only upon perceived hostile activity. This is particularly true for isolated honeypots since they are systems that should not normally be accessed. This means that all traffic to a honeypot should be automatically suspect. This is a fundamental difference between a honeypot and a standard IDS, which typically has problems with "false positives" (i.e., the system trips over legitimate traffic). Note that false positives can still be generated in a honeypot system via network management tools and vulnerability assessment tools, but the number of false positives is significantly reduced.

- A honeypot can be used as a hostile intent assessment system. A common practice of attackers is to scan the Internet doing "banner checks". Any numbers of text-based protocols issue text banners when you connect to the service. Such a protocol can serve to fingerprint the operating service, since many banners reveal the exact version of the operating system. If an attacker knows the system version, they may be

able to exploit know weaknesses in the system. Common examples of such systems are RedHat 5.0 and Windows NT 4 IIS 4.0.

- The honeypot can be setup to look like a typical banner system, and trip when an attacker intrudes into the system. For example, several versions of well-known packages have "buffer overflow" holes. Buffer overflow holes allow attackers to take advantage of system weaknesses in data storage management, and at worst allow for system access via data overflow. In a typical scenario, an attacker connects to port 110 (a port used for POP3 e-mail services), uses buffer overflow to grab the version of the operating system from the banner, and looks for known weaknesses to exploit the system.

- A honeypot can teach you about incidence response. Setting up systems that attackers break into will teach you how to detect such attacks and how to "clean up" after them. A honeypot can also teach you about attack techniques and the various levels of sophistication used.

- Setting up a honeypot server may alert you to hostile activity long before the real systems are attacked. Generally, attackers try simple techniques before moving on to more sophisticated levels of attack. Thus, setting up a "non-hardened" system (i.e., a weakened system from a security standpoint) may alert you to hostile intents.

- A potential advantage of a honeypot is deterrence. Simply knowing that honeypot systems exist may inhibit some attackers from trying to hack into a system. Of course, the flip side of this is that any such system is also considered a challenge by some attackers and may invite attacks.

Disadvantages of a Honeypot

- If a honeypot system is successfully attacked, it can be used as a "hop" to further compromise the network or other networks. This is perhaps the biggest danger in setting up such a system. In particular, legal considerations may arise when a system you control attacks a third party. One common way to minimize this problem is to place the honeypot behind a firewall. While this may make it more difficult for a potential attacker to reach the honeypot, it has two advantages: it minimizes outbound traffic (if set up properly), and placing the honeypot behind a firewall makes the system seem more legitimate, particularly if the system is not hardened.

- Honeypots take effort to set-up and properly maintain. In general, additional effort and complexity for any security system is undesirable, since it can lead to additional ways to exploit the system. In fact, since

honeypots require continuous effort, some security administrators may simply turn them off after a while.

- It is a commonly held belief that that since honeypots lure attackers, legal rights to prosecute are reduced. In fact, since honeypots are not "active" lures, legal rights of attacked systems are not reduced. Of course, prosecution is another matter.

Types of Honeypots

Several different types of honeypots are employed in the security industry today. These range from the simplest (port monitors) to the most complex (full systems plus network IDS) and are briefly discussed below.

Port Monitors

These are the simplest type of honeypot. A port monitor is simply a "sockets" based program that opens up to a listening port. A "socket" is defined as the minimum amount of information necessary for communication on the network, and originated from TCP/IP. A socket contains the source/destination IP address, the source/destination port, and the transport protocol (UDP or TCP).

A port monitor will listen for traffic on ports typically scanned by attackers. However, this will alert an attacker that the port is monitored, because the port monitoring system will first accept, and then drop the connection. When a connection is suddenly dropped, it alerts the attacker of the possibility of an IDS running on the port.

Deception Systems

A port monitor is simply a passive listening device that may actually alert attackers, as discussed above. The next step up in complexity is a deception system that interacts with the attacker. In contrast to a port monitoring system, a deception system will respond to port intrusions as if it is an actual server. For instance, such a system usually comes with banner headlines to use as a lure for potential attackers.

Multi-Protocol Deception Systems

A multi-protocol system is simply a deception system having multi-protocols and banners to emulate packages for different operating systems. Examples of such a system include commercially available systems like the Deception Toolkit (DTK) and SPECTER. Both of these packages simulate multiple operating systems and network services.

Full Systems (with and without IDS)

A full system goes beyond a honeypot (which is implemented strictly for deception). A full system is fully functional and operational, and is usually set to

alert on exceptional conditions. A full system with IDS includes a full intrusion detection system to supplement the internal logging of the full system.

Sniffers

A packet sniffer is a wiretap device that plugs into computer networks and eavesdrops on network traffic. Since network traffic consists of binary data, sniffers come with "protocol analysis" which decodes the binary traffic. Most sniffers are capable of decoding common TCP packets like SNMP, Telnet, and HTTP.

A packet sniffer consists of "hardware" (usually standard network adapters), a "capture driver", which is the heart of a sniffer and actually captures the network traffic, filters it and stores the data in a buffer, a "buffer" which allows the captured data to be stored for analysis, a "decoder" which displays the contents of network traffic with descriptive text, and "packet editing/transmission" which allows the user to edit the network packets and re-transmit them over the network.

Some security consultants do not use sniffers in conjunction with honeypots and simply log the attackers moves from the honeypot. While this is certainly technically possible (especially with commercial honeypot systems), it is not advisable to log information on the honeypot itself. First, the simpler you can make the honeypot the better (the more complex a honeypot is, the more likely it is to arouse suspicions for an attacker), and secondly, you may lose the information since the attacker will have root in the honeypot. As such, a talented attacker may be able to erase any tracking logs contained within the honeypot.

The advantage of using a sniffer in conjunction with a honeypot is a sniffer will pick up all keystrokes and screen captures. In this way, you can see exactly what the attacker sees.

A disadvantage is that an attacker can hide his moves with encryption, and that talented attackers can spoof a sniffer. A "spoof" is a simply a method whereby a user impersonates another user without permission.

Conclusions

Honeypots are a valuable tool that can allow you to learn about attackers. When implemented and maintained especially with sniffers, they provide a window into potential attack methods.

However, honeypots cannot be viewed as a security panacea. They are but one facet of a successful IDS, which may also include firewalls and logs, network based detection systems, host based detections systems, virus scanners, response planning, and other countermeasures such as

authentication/encryption and Virtual Private Networks. The security specialist must remember that the key to detecting and tracking attacks is security layers.

References

1. Graham, Robert. Network Intrusion Detection Systems. 2000. URL: http://www.robertgraham.com/pubs/network-intrusion-detection.html.

2. Graham, Robert. Sniffing FAQ. September 2000. URL: http://www.robertgraham.com/pubs/sniffing-faq.html.

3. Spitzner, Lance. Intrusion Detection. 2000. URL: http://www.enteract.com/lspitz/ids.html.

4. Spitzner, Lance. Know Your Enemy: I, II and III. 2000. URL: http://www.project.honeynet.org/papers.

5. Conover, Matt. Heap and Buffer Overflows. January 1999. URL: http://packetstorm.security.com/docs/infosec/buffer-overflows.txt.

6. Klug, David. Honeypots and Intrusion Detection. September 13, 2000. URL: http://www.sans.org./infosecFAQ/intrusion/honeypots.htm.

7. Winkler, Robert. Intrusion Detection Systems. December 9, 2000. URL: http://www.sans.org/infosecFAQ/intrusion/systems.htm.

8. Northcutt, Stephen. Network Intrusion Detection-An Analysts' Handbook. SANS GIAC. New Riders Publishing. 2000.

9. Even. Loris. What is a Honeypot? July 12, 2000.URL: http://www.sans.org./nwelook/resources/IDFAQ/honeypot3.htm.

10. Manderschid, Scott. An Intrusion Detection Process: Defense in Depth. February 9, 2001. URL: http://www.sans.org/infosecFAQ/intrusion/process.htm.

11. Schneier, Bruce. Secrets and Lies. John Wiley and Sons, Inc. 2000.