2 What a Honeynet Is

HONEYPOTS

The concept of honeypots has been around for years. Simply put, honeypots are systems designed to be compromised by an attacker. Once compromised, they can be used for a variety of purposes, such as an alerting mechanism or deception. Honeypots were first discussed in a couple of very good papers by several computer security icons: Cliff Stoll's *Cukoo's Egg*[1], and Steve Bellovin and Bill Cheswick's "An Evening with Berferd."[2] Both instances used jail-type technology to capture an intruder's sessions and to monitor in detail what the intruder was up to. The term honeypot came later, but the same intent applies: setting up one or more systems that seem attractive to network intruders but are also capable of monitoring to a fine degree what is going on. By monitoring activity through a honeypot, you can identify the problem and be reasonably sure that you know how the intruder(s) got in and what they are doing on the compromised system. Traditionally, a honeypot has been a single system connected to an existing production network in order to lure attackers. Figure 2-1 shows a single physical

1. C. Stoll, *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage* (New York: Pocket Books, 1990).
2. *http://www.securityfocus.com/data/library/berferd.ps*

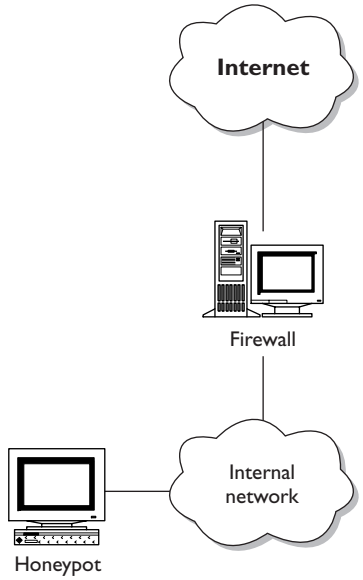

Figure 2-1 A traditional stand-alone honeypot

system placed in an internal network. This single system can then emulate various systems or vulnerabilities.

A variety of products or solutions allow you to create your own honeypot. Such options include:

- Fred Cohen's Deception Toolkit (*http://www.all.net/dtk/index.html*)
- Cybercop Sting (*http://www.pgp.com/products/cybercop-sting/default.asp*)
- Recourse Mantrap (*http://www.recourse.com/products/mantrap/trap.html*)

Each of these applications has its own interpretation of what a honeypot is and how it should be used.

For example, the Deception Toolkit, commonly called DTK, is a collection of scripts that emulate various known vulnerabilities. One such simulated vulnerability in DTK is an old Sendmail vulnerability that hands out a fake password file.

These scripts are then run on a host system. The attacker gets suckered into taking this fake password file and spending valuable time cracking passwords that are not real. The purpose of the toolkit is deception. This toolkit is also excellent for alerting and learning about known vulnerabilities.

Although such an approach is useful, keep in mind that one of the main goals of the Honeynet Project is to learn about *unknown* vulnerabilities. With the Deception Toolkit, you are limited to learning about what is already known.

Cybercop Sting is a honeypot that runs on NT emulating an entire network by replicating the IP (Internet Protocol) stacks of various operating systems. A blackhat could scan an entire network and find 15 systems available, each with a different IP address. However, all 15 virtual systems are contained within the one physical honeypot machine. Both the systems and the IP stacks are emulated. The advantage here is that you can quickly and easily replicate an entire network, allowing you to track trends. However, the problem is that you can emulate only limited functionality, such as a TELNET login or an SMTP (Simple Mail Transfer Protocol) banner. The blackhat community has no real operating system to access and interact with beyond that facade.

We wanted to learn everything possible, such as what happens once a system is compromised. We wanted the keystrokes and the system logs of a compromised system. In other words, we wanted our attackers to be able to fully exploit and take over their targets so we could zoom in afterward and learn as much as possible. Given their limited emulation capabilities, products like Cybercop Sting cannot provide that information.

Recourse Mantrap, a commercial product that comes close to the functionality of a Honeynet, does not replicate an operating system but instead runs an image of an operating system within another one. This so-called "jail" has a great advantage in that a real operating system is running. Unknown vulnerabilities can be learned, and the blackhat has a complete OS (operating system) to interact with once the system is compromised. However, you are limited to operating systems that the vendor can provide. For example, you may want to use HPUX or perhaps a network device, such as an Alteon switch. Also, you, the user, still must solve the problem of how to contain the blackhat once the system is compromised. The

Recourse Mantrap does not have the capability to limit blackhat activity. An attacker could use the compromised honeypot as a jumping-off point to attack additional systems. The product has excellent data-capture functionality but lacks the ability for detailed data control.

Most of these solutions share the problem of detectable signatures. It may be possible to identify these products based on signatures they leave, allowing moderate or advanced blackhats to realize the deception and move on to safer targets. All these solutions have excellent potential but only for specific requirements. None of them met all our requirements for the Honeynet Project. We wanted a flexible environment in which nothing was emulated, the systems were the same as those found on the Internet, and we could capture the activity of blackhats from beginning to end. Additionally, we did not want to endanger any other systems on the Internet, so we needed a solution that couldn't be used as a jumping-off point for an attack. We devised our own solution to meet all these requirements.

HONEYNETS

A Honeynet is different from the honeypot solutions we have discussed so far. The Honeynet is a tool for research; it is a network specifically designed for the purpose of being compromised by the blackhat community. Once compromised, the Honeynet can be used to learn the tools, tactics, and motives of the blackhat community. The two biggest differences between honeypots and our Honeynet solutions are as follows.

- A Honeynet is not a single system but a network. This network sits behind a firewall where all inbound and outbound data is contained, captured, and controlled. This captured information is then analyzed to gain intelligence about our adversary. Within this Honeynet, we can place any type of system to be used as a honeypot, such as Solaris, Linux, Windows NT, Cisco switch, and so on. This creates a network environment that has a more realistic "feel" to it for the intruder. Also, by having different systems with different services, such as a Linux DNS, a Windows NT Web server, or a Solaris FTP server, we can learn about different tools and tactics. Perhaps certain blackhats with specific techniques or motivations target specific systems or vulnerabilities. By having numerous systems, we are more likely to discover these differences.

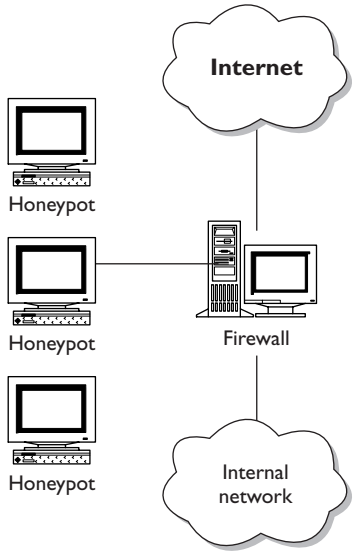

Figure 2-2 A Honeynet

- All systems placed within the Honeynet are standard production systems. These are real systems and applications, the same as you find on the Internet. Nothing is emulated. Nor is anything done to make the systems less secure. We can learn a great deal from using such systems. The risks and vulnerabilities discovered within a Honeynet are the same that exist in many organizations today. Additionally, a Honeynet can be as dynamic and flexible as your own organization.

The Honeynet's use of production systems makes it unique. Nothing is emulated, allowing you to use the same systems and applications found in your organization. Figure 2-2 shows a Honeynet. Each honeypot is a production system, mirroring the same builds that an organization would find on its internal network.

VALUE OF A HONEYNET

Traditionally, information security has been defensive. Firewalls, intrusion detection systems, encryption: All these mechanisms are used defensively to protect one's resources. The strategy is to defend one's organization as best as possible,

detect any failures in the defense, and then react to those failures. The problem with this approach is that it is *purely* defensive; the enemy is on the attack. Honeynets attempt to turn the tables, giving organizations the initiative. The primary purpose of a Honeynet is to gather intelligence about the enemy. By doing so, organizations can, potentially, stop an attack or a failure in defense before it happens. Information security has often been compared to the military, such as the defense of a castle or guerrilla warfare. Regardless of the analogy you choose, organizations can take the initiative by learning about the enemy before it strikes.

For example, one of the primary communication channels blackhats use is IRC (Internet relay chat). Blackhats tend to communicate freely among themselves, revealing their motives, goals, and actions. We have captured these conversations through the use of Honeynets, monitoring every word. We have even captured real-time video shots of blackhats involved in the attacks on our Honeynet. Once, we tracked blackhats compromising hundreds of systems for the sole purpose of attacking the infrastructure of a specific country. We then relayed this information to organizations that were compromised by these individuals. We also warned the country of the impending attack, thereby mitigating the effectiveness of the blackhat attacks. We were able to specify the attackers' exact tools and methodology, tipping off these organizations with specific information to better react to and defeat the threat. You can read more about this incident in Chapter 11.

Honeynets also provide an organization with intelligence on its own security risks and vulnerabilities. Honeynets can consist of the same systems and applications that the organization is using for its production environment. This allows you to identify the risks and vulnerabilities that may exist in *your* production environment. For example, if your organization depended on Microsoft NT IIS (Internet Information Server) with a database back end for its Web server application, you could build a Honeynet with those components, allowing you to identify any risks existing in that environment. You can also use systems that you want to test or are considering for deployment. Perhaps you are considering a new load balancer or switch and have concerns about possible risks. The Honeynet gives you an environment in which you can test those risks. Often, these same risks may be missed in your production environment, owing to data overload. The production network entails so much activity that it is difficult to determine what is malicious activity

and what is normal day-to-day network traffic. However, within the controlled environment of the Honeynet, these risks are easier to identify.

Furthermore, Honeynets can help an organization develop its incident-response capabilities. Over the past two years, the Honeynet Project has vastly improved our abilities to detect, react to, recover, and analyze systems that have been compromised. After numerous system compromises, we have perfected a variety of techniques. You can read more on these techniques in Chapter 6, Analyzing a Compromised System, and Chapter 8, Forensic Challenge. Traditionally, when you analyze a compromised system, you have no idea whether your analysis is correct; you can make only a best guess. The advantage one has in analyzing Honeynet compromised systems is that you already have most of the answers, as you captured every packet and keystroke sent to the system. You can then treat a compromised system as a "challenge," testing your abilities to determine what happened by using various forensic techniques. You can then compare these results to the data captured from within the Honeynet. This information can also be used to determine whether any other systems in your production network have been compromised. Once you have identified the signatures of the blackhat and the attacks, you can then review your production environment for the same signatures, identifying compromised systems you did not know about.

Over the years, we have discovered another advantage of Honeynets: They teach us a lot about not only the blackhat community but also ourselves and our security capabilities. A Honeynet is nothing more than a highly controlled lab that you put out on your network or on the Internet. You learn when blackhats compromise systems on the Honeynet. However, you also learn a great deal just setting one up and maintaining it. While working with Honeynets, we have learned extensively about logging, IDSs, forensics, network traffic analysis, system hardening, kernel modules, and a variety of other techniques.

THE HONEYPOTS IN THE HONEYNET

To learn more about the blackhat community, our honeypot systems were usually default installations of commonly used systems. We did nothing to secure these systems, but we did nothing to make them more insecure, either. Our goal was to use systems commonly found on the Internet. Many organizations feel

that they are not at risk and do little to protect or to secure their systems. It was these very organizations that we hoped to prove wrong. By demonstrating the tools, tactics, and motives of the blackhat community, we hoped not only to learn but also to raise awareness. Many organizations also feel they have nothing of value to be compromised. As you will soon learn, these are the very organizations that many blackhats target.

The honeypots we have used are default installations of Red Hat Linux, Windows 98 desktop, Windows NT server, and Solaris server. We then proceeded to build these systems, using default parameters and keeping customization to a minimum. During the entire build and installation process, we selected default parameters. Nothing was done to make the systems more secure. Many security professionals would consider these systems insecure, and they are correct. Most default installations of an operating system are highly insecure, especially if no measures are taken to harden them. Unfortunately, these very same default installations are a high percentage of systems connected to the Internet. Many organizations take no measures to secure their systems, believing that they are secure or not realizing their exposure to risk. It is these very organizations that the Honeynet Project has tried to replicate. For organizations that do secure their systems, the lessons learned here still apply. As you will soon learn, regardless of who you are and where you are located, the blackhats will find you. All it takes is one mistake or an unknown vulnerability, and your organization can be compromised.

Some people have questioned whether this technique is entrapment. Systems purposely intended to be compromised could be considered an attempt to entrap the blackhat community. However, we firmly believe that a Honeynet is not a form of entrapment, for the following reasons.

- The intent of the Honeynet is not to catch bad guys but only to learn from them. Activity within the Honeynet is captured and analyzed and is not used to prosecute. At certain times, members of the law enforcement community have been informed of our findings. However, this information is not used to prosecute individuals.
- Systems in the Honeynet do not differ from those in many production environments. The only difference is that the data entering and leaving the Honeynet is

more closely studied. If the Honeynet is considered a form of entrapment, then so too would many production networks found on the Internet.

- The Honeynet Project does not do anything to attract the blackhat community to our machines. We do not actively advertise their existence or lure people into accessing them. Blackhats actively find and compromise these systems on their own initiative. You will be amazed at how aggressive blackhats can be.

Honeynets do have their limitations. They are primarily a tool to learn, to be used for research and intelligence gathering. They are not the ultimate solution to all your security problems. We, the Honeynet Project, highly recommend that you first focus on securing your existing environment, using security best practices, such as applying patches, eliminating unneeded services, and reviewing your system logs. It is these day-to-day mundane but critical procedures that are vital to any organization's security. Once such standards have been met and are part of your everyday procedures, a Honeynet can add value to an organization. Meanwhile, the Honeynet Project hopes to continue its research and to share its lessons learned.

SUMMARY

Honeynets are a tool to learn—specifically, the tools, tactics, and motives of the blackhat community. What makes a Honeynet unique is the fact that nothing is emulated. Instead, a highly controlled network is made up of machines running operating systems and applications identical to production systems. Once compromised, the systems not only teach us how the blackhat community operates but also identify risks and vulnerabilities that exist in our own environment. This is the primary value of the Honeynet: learning. Now, let's discuss how a Honeynet works.