

Using Honeynets to Protect Large Enterprise Networks

Computer networks currently connected to the Internet are vulnerable to a variety of exploits that can compromise their intended operations. They're subject to denial-of-service attacks, for example, which prevent other computers from connecting to

report was the first news of an infected system. Let's look at two typical case studies from our success stories.

A system with a compromised password

Our honeynet helped us identify a system an attacker compromised by obtaining a username and password, something that is difficult to detect with traditional methods. The system the attacker connected to on the honeynet was running Microsoft NT 4 Workstation software. Several days earlier, the same attacker compromised this same system by using a Microsoft Internet Information Server (IIS) exploit and setting the system up as a warez server. The attacker also set up a backdoor for later use; several days after the initial compromise, the attacker connected to this backdoor from another computer within the Georgia Tech Enterprise Network. We immediately notified network security personnel of this other potentially compromised computer.

After completing an offline analysis, network security personnel found no indication that this production machine had in fact been compromised, yet this machine's owner wasn't the person who connected to the warez server on our honeynet. The network security personnel speculated that the attacker obtained this machine's password by using a brute-force attack. The team instructed the user to change his password by selecting a more secure one and to discontinue using his production machine's password when establishing accounts on other Web sites. They concluded

JOHN G. LEVINE, JULIAN B. GRIZZARD, AND HENRY L. OWEN
Georgia Institute of Technology

them and vice versa. They're also subject to attacks that could cause them to cease operations temporarily or permanently. An attacker could compromise the system and gain root access—that is, gain the ability to control the system as if the attacker were the system administrator.

Network administrators use several methods to protect their network. Firewalls, for example, control the flow of traffic between the local network and the Internet. Based on the characteristics of the network traffic such as requested services, source and destination addresses, and individual users, a firewall can decide whether to let traffic pass through the network.¹ Firewalls can also be used on end-user systems.

Another method employed to protect networks is the use of an intrusion-detection system (IDS). The administrator can place IDS sensors at various points throughout the network, including the interfaces between the local network and the Internet, critical points within the local network, or on individual systems. An IDS is usually signature based, meaning it looks for predefined signatures of bad events; these signatures typically reside in a database associated with the IDS. An IDS can also perform statistical and anomaly analysis of network traffic to detect

malicious intrusions. When it detects malicious activity, it can then notify the network administrator.²

Installing a honeynet within large enterprise networks provides an additional security tool. Honeynets complement the use of firewalls and IDSs and help overcome some of the shortcomings inherent in those systems. In addition, honeynets can also serve as platforms for conducting computer security research and education.

Georgia Tech's honeynet

Initially, we established the Georgia Tech honeynet during the summer of 2002 as a single computer, but we've since expanded it to include several different machines running various operating systems. Figure 1 shows a configuration of the Georgia Tech honeynet.

The Georgia Tech honeynet has helped us find, on average, approximately 60 compromised machines per month. These compromises include worm exploits as well as individual systems targeted and compromised by attackers. Whenever the honeynet detects a compromised system, it sends a report to Georgia Tech network security personnel. In some cases, network security personnel were already aware of the compromised machines, but sometimes our

that detecting this compromised system would have been very difficult if they had just used their existing security measures instead of the honeynet.

Characterization of detected exploits

Any research effort requiring a copy of an entire exploit session is well suited to the use of a honeynet. Part of our own research effort involves using a honeynet to collect new rootkit exploits. A rootkit is a set of tools an attacker uses to retain access to a system after it's compromised. Rootkits install a backdoor on a system and usually have some functionality to enable the attacker to hide activities and files.

Rootkits are available via download from the Internet, but the honeynet gives us an opportunity to collect the ones that might not have been previously seen or those not publicly disclosed.

At 10:34 a.m. UTC on 1 June 2003, an exploit was launched against a honeynet system on port 21 (ftp daemon) in an attempt to gain root-level access. The ftp server running on the Red Hat Linux 6.2 machine was the `wu-ftp2.6.0(1)` ftp daemon, or the default ftp server. Exploits that grant an attacker root-level access for this particular service are available on the Internet.

After successfully gaining access, the attacker was able to install a rootkit called `r.tgz` on the target system. We don't believe this particular rootkit had been publicly analyzed previous to this attack; we knew of an ssh rootkit called `r.tgz`, but its characteristics, such as file size, differed from the rootkit installed on the target system (see www.pack-etu.org/hpa.html).

The attacker extracted the exploit code within the `r.tgz` file and then ran the exploit on the target system. Figure 2 shows the actual honeynet logs from the attacker's session. The `r.tgz` rootkit deletes all traces of itself on the target system

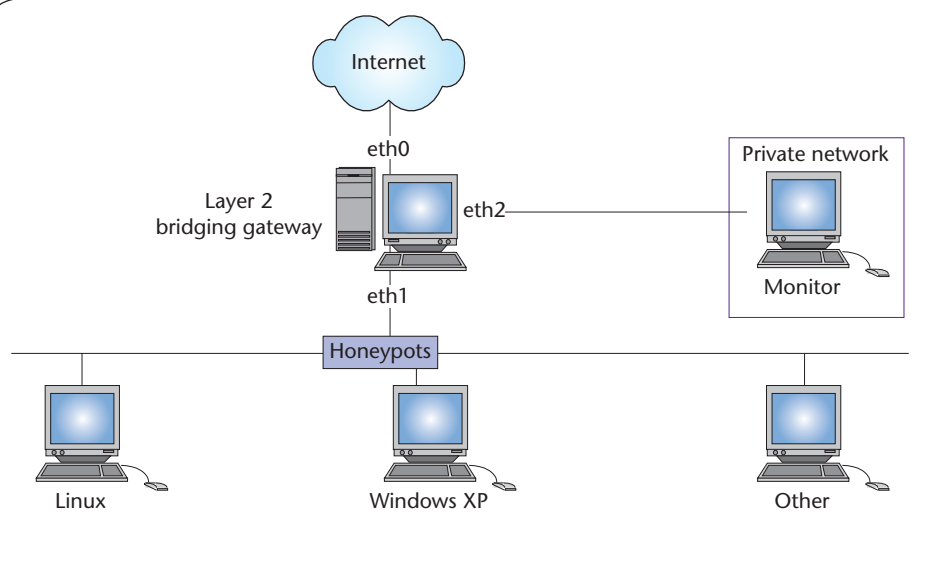


Figure 1. Georgia Tech honeynet. The honeypots are behind the layer 2 bridging gateway, which bridges traffic between interfaces eth0 and eth1. All traffic is logged on the bridge; interface eth2 can be used to monitor the traffic.

after installation, but we were able to reconstruct what the attacker accomplished by using the honeynet logs from this exploit session.

Using the elements of a methodology for detecting unique rootkit string signatures described in more detail elsewhere,³ we detected some unique string signatures in the binary files that the `r.tgz` rootkit replaced.

We examined the captured rootkit and discovered that it redirected the system call table to an entirely new system call table.⁴ Based on additional analysis, we were finally able to uninstall and then reinstall this rootkit on the target system. We concluded that `r.tgz` is a blended rootkit that contains elements of the INKIT kernel rootkit and the `hax.tgz` binary rootkit. The INKIT rootkit is based on SuckIT; the `hax.tgz` rootkit is based on `bigwar.tgz` rootkit (see www.honeylux.org/lu/project/honeyluxR1/result/sub01/report/hax.htm).

The Georgia Tech honeynet remains active and continues to help secure the Georgia Tech network. We have ongoing projects in-

```

Stream Content
Follow TCP stream
[1:36m Starting Installation Of Rootkit ... [0m
[1:33m Step [1:35m: [1:37m1 ... [0m
[1:32m[-] [1:36mStarting Creating Our Local Directory And Moving Programs ... [1:32m[-] [0m
[0:32m[-] [1:36m Moving & Copying Utils Programs On Our Directory ... [0:32m[-] [0m
[0:32m [-] [1:33mcuratate.[0m
[0:32m | \0m
[0:32m | [-] [1:36mdone moving [1:33mcuratate.[0m
[0:32m [-] [1:33mcl & X.[0m
[0:32m | \0m
[0:32m | [-] [1:36mdone copying [1:33mcl & X.[0m
[0:32m [-] [1:33mfirewall status & socklist.[0m
[0:32m | [-] [1:36mdone copying [1:33mfirewall , status & socklist.[0m
[0:32m [-] [1:33mread & write.[0m
[0:32m | \0m
[0:32m | [-] [1:36mdone copying [1:33mread & write.[0m
[0:32m [-] [0m
[1:33m Step [1:35m: [1:37m2 ... [0m
[1:32m[-] [1:36mStarting FireWall Rules ... [1:32m[-] [0m
[1:33m Step [1:35m: [1:37m3 ... [0m
[1:32m[-] [1:36mRemoving Other Rootkits [1:33m (If They Exist Here) [1:36m... [1:32m[-] [0m
[1:33m Step [1:35m: [1:37m4 ... [0m
[1:32m[-] [1:36mReplacing Some Files [1:33m (If They Exist Here) [1:36m... [1:32m[-] [0m
type port inode uid pid fd name
tcp 23 23141 0 13001 2 in.telnetd
tcp 21 23065 0 13201 1 libss
tcp 21 11901 0 001 4 r.tgz

```

Figure 2. Honeynet logs. An attacker installed the `r.tgz` rootkit on 1 June 2003, and these logs from that session show the installation process. We were able to reconstruct the entire attack using the honeynet logs.

cluding initiatives to help attract more advanced threats. We're also looking at methods to recover from compromises and will use the honeynet as a testbed for this research. □

References

1. E. Skoudis, *Counter Hack*, Prentice Hall, 2002, p. 47.
2. S. Northcut et al., *Inside Network Perimeter Security*, New Riders, 2003, p.5.
3. J. Levine, H. Owen, and B. Culver, "A Methodology for Detecting New Binary Rootkit Exploits," *Proc. SoutheastCon*, IEEE Press, 2003.
4. J. Levine, J. Grizzard, and H. Owen, "A Methodology to Characterize Kernel Level Rootkit Exploits Involving Redirection of the System Call Table," *Proc. 2nd Int'l Information Assurance Workshop*, IEEE Press, 2004, pp. 107–125.

John G. Levine is a professor at the United States Military Academy. His research interests include network security. Levine recently received a PhD in electrical and computer engineering from the Georgia Institute of Technology. Contact him at levine@ece.gatech.edu.

Julian B. Grizzard is a PhD student in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. His research interests include network and operating system security. Contact him at grizzard@ece.gatech.edu.

Henry L. Owen is a professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. His research interests include network security. Contact him at owen@ece.gatech.edu.