Chapter 19

CONSTRUCTING YOUR BASTIONS

INTRODUCTION

The largest security team at the largest enterprise is still many orders of magnitude smaller than the population attempting to break into the enterprise's network. Why? The Internet [1]!

In most cases, would-be intruders use a number of small programs to probe your host computer for weaknesses. When malicious hackers find a hole, they use other tools to exploit the vulnerability and gain access to your network. To protect yourself from these black hats, it's important to see your own system from the perspective of a person with the necessary tools. Only then can you effectively fortify your network against probing and potential attacks [1].

HOST PROBING

When probing a host machine, the goal is to find out as much information as possible about the machine. Using that information, black hats can further probe and even attack the remote machine. The most critical pieces of information found during a probe include the host's operating system and the available applications on the computer. With that information, the crackers can discern which vulnerabilities to exploit [1].

With each probing tool and its respective countermeasures, the machine becomes more secure, as it is more difficult for probing software to see it. The less information someone can get about your machine, the better [1].

PORT SCANNING

To learn which applications are potentially exploitable on a machine without having to log into the machine, an intruder can use a port scanner. The port scanner steps through all the network ports on a remote machine, attempting to establish a connection to each one, producing a brute-force report that lists which services are running and accessible to the machine doing the probing [1].

While there are many port scanning tools, you may want to use one that is free. It could be a simple Unix program that gives feedback on which ports it was able to connect to and, if possible, which protocols are related to those ports. The protocols often indicate that a potentially exploitable application is connected to that port. For example, Web servers (which use the HTTP protocol) are connected to port 80 on most computers. If the scanning tool can establish a connection to port 80, you can generally assume that the HTTP protocol is related to that port on that machine [1].

OS FINGERPRINTING

The second step is to learn which operating system the remote machine is running, using a technique called OS fingerprinting. It was once the case that, to determine the OS running on a remote machine, one could use a primitive tool like telnet. Connecting to another computer with this utility would yield basic information regarding the remote system's available services. From that information, you could infer, with some degree of accuracy, which OS was running on it [1].

To take the guesswork out of fingerprinting, developers invented a new generation of tools that use creatively formed IP packets to test for behavioral patterns in a remote machine's IP stack. The software sends packets with particular characteristics that cause only certain operating systems to handle them in predictable ways. Each packet/response pair further limits which operating systems the remote machine could be, until the tool has no further ability to narrow the OS options and it reports back one or more choices [1].

The best tool for this type of OS fingerprinting is one that incorporates multiple techniques used in the area of packet-based OS detection and now looks for particular signatures associated with operating systems that can be used to classify types of remote machines. If the software responds with multiple OS choices, it will probably be the same base OS and different version numbers [1].

The last step in securing your system from host probing, after port scanning and OS fingerprinting, is to learn which exploits exist for those combinations and which countermeasures can defend against them. Many applications on a given OS have known vulnerabilities [1].

For example, you can further explore port 25, the standard port for the Simple Mail Transport Protocol (SMTP), using the relaycheck.pl utility. By doing so, you'll find out whether the system uses the sendmail Mail Delivery Agent (MDA), and whether that agent is misconfigured. You can also determine if the agent can be used as an SMTP relay, and