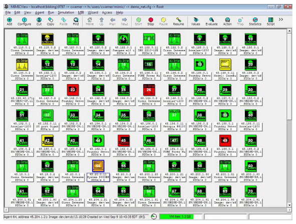
GAMBIT COMMUNICATIONS Application Brief

Port Scanners and Sweepers Testing

A port scanner is a software application designed to probe a network host for open ports. This is often used by administrators to verify security policies of their networks and by attackers to identify running services on a host with the view to compromise it. Typically, a port scanner scans many hosts on the network for listening ports. This is also used to search for a specific service, for example, an SQL-based computer worm may scan ports looking for hosts listening on TCP port 1433.

Many scanners discover hosts and services on a computer network and create a "map" of the network. They send specially crafted packets to the target host and then analyze the responses. They not only send packets at some predefined constant rate, but also account for the network conditions (latency fluctuations, network congestion, and the target interference with the scan) during the run. Many extend their discovery capabilities beyond basic host being up/down or port being open/close to being able to determine operating system of the target, names and versions of the listening services, estimate uptime, the type of device, and presence of the firewall.



The result of a scan on a port is usually generalized into one of three categories:

- Open or Accepted: A service is listening on the port.
- Closed or Denied or Not Listening: Connections will be denied to the port.
- Filtered, Dropped or Blocked: There was no reply from the host.

Open ports present two vulnerabilities of which administrators must be wary:

1. Security and stability concerns associated with the program responsible for delivering the service.
2. Security and stability concerns associated with the operating system that is running on the host.

Closed ports present only the latter of the two vulnerabilities that open ports do. Blocked ports do not present any reasonable vulnerability. There is also the possibility that there are no known vulnerabilities in either the software or the operating system at this given time.

Testing these scanners thoroughly is very important. The test setup needs a large network with thousands of hosts connected on different subnets. Each of them needs to have many ports with various open/close/blocked conditions. Having this type of setup in a QA lab is very expensive and also hard to maintain. Testing it on a production network is not a recommended way. But without the hardware available for development and testing, it is hard to assure the customers that the scanner will run in their environment and will find all the vulnerabilities.

A fast, easy way of containing capital expenditures for port scanning and network management software testing is by using the **MIMIC Server Simulator**. MIMIC Server Simulator creates a large network with variety of devices with different IP addresses. Each of them can be setup to support many services running on different ports. MIMIC makes it easier to switch ports on/off at run-time. It is also very easy to add latency or create a faulty links by dropping the packets. Since MIMIC's services are proxied, **even the advanced port probing software can be fooled into thinking that the actual service is there.**

MIMIC saves time for setting up large sized labs, creating complex scenarios and storing those for future use. By simulating thousands of expensive devices, QA groups can reduce a significant amount of capital budget.

With MIMIC, testing proceeds faster, companies can have higher confidence in the level of software testing, and software products can be released to customers more quickly. And **win the time to market battle with the competition.**

For more information:
Please contact Gambit Communications: info@gambitcomm.com

www.gambitcomm.com