Ports, Protocols & Services (PPS) Overview

Imagine a massive building so enormous that looking at it from the outside there are 1,023 doors and 64,512 windows! The first floor is entirely doors, numbered from 1 to 1,023. When you look up, you can see each of the windows has a number painted on it as well. For such a large building, you can see activity in only some of the windows and doors – only where it is open and you can see the lights are on.

Now imagine that you were just hired as a security guard and it is your job to ensure the safety of the building's occupants. You know there is a mailroom in room 25, the information desk is in room 53, and it looks like someone is taking a smoke break and has the window in room 1,337 open. There are a few other doors and windows open, but you are not sure what departments those belong to or who is authorized entry.

What is the first thing you would do? Make sure that the doors and windows are closed and locked when not in use? Try to monitor the people that come and go from the building? Does that delivery guy really work for FedEx?

In reality, this explanation of a building with 65,535 doors and windows (and the uphill battle of securing them all) is a simplistic example of a computer Operating System (OS). Welcome to the world of Ports, Protocols & Services (PPS)!!

Communications Breakdown

Application communication is based on established protocols, just like common courtesies and protocols people use to properly greet each other and conduct business. These protocols were originally designed for legitimate use, but hackers have found ways to exploit them. However, hackers also have to follow some basic principles of how OSs communicate, so it is possible to also identify misuse by PPS configurations. The ability to know what services are running, what ports are open, and being able to identify the two parties involved in the communications exchange serve as a basic foundation for Intrusion Detection Systems (IDS) and vulnerability assessment tools.

Regardless if you are Windows, Unix, Linux, or Mac, your OS communicates via 65,535 ports and their associated protocols. The port numbers are divided into three ranges:

- Well Known Ports (0 - 1,023)
- Registered Ports (1,024 – 49,151)
- Dynamic and/or Private Ports (49,152 – 65,535)

When two networked computers communicate with one another, they send data to each other in a mutually agreed upon manner. A protocol, much like that in human communication, is comprised of a precise and specific definition of how communication should start, continue and end.

For example, the communication between a mail server and a mail client is very different than the communication between a web server and a web browser. Mail clients need to know how to read a message, delete a message, download more mail headers, etc and communicate using the POP or IMAP protocols. Web clients need to know how to fetch web pages, process web page contents, upload form data, etc, and speak the HTTP protocol. Typically a given client application is designed to communicate with a particular class of server applications using a single protocol. Like anything predictable, it can be exploited.

Socket To Me

A server application normally listens to a specific port waiting for connection requests from a client. When a connection request arrives, the client and the server establish a dedicated connection over which they can communicate. During the connection process, the client is assigned a local port number, and binds a *socket* to it.

A socket is a network communications end point - the analogy is to a power cord being plugged into an electrical socket. The client talks to the server by writing to the socket and gets information from the server by reading from it.

A socket is an end-to-end connection between a client and a server identifying:

- IP address (client and server)
- protocol (TCP or UDP)
- port (client and server)

Sockets come in two primary flavors: active and passive.

- An active socket is connected to a remote active socket via an open data connection. Closing the connection destroys the active sockets at each end point.
- A passive socket is not connected, but rather awaits an incoming connection, which will spawn a new active socket.

A socket is not a port, though there is a close relationship between them. A socket is associated with a port, though this is a many-to-one relationship. Each port can have a single passive socket, awaiting incoming connections. However, it can have multiple active sockets, each corresponding to an open connection on the port. Normally, a computer runs a specific application and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

Information security analysts and hackers can identify the function of servers from the listening ports. This can identify mail servers, DNS servers, SQL servers, the type of backup software used, remote admin tools, and more. This process of identifying a target is called footprinting.

Fire Me Up!

Returning to the idea of the building with the 65,535 doors and windows, the process of securing or "hardening" a computer is simply turning off unnecessary services and closing ports that are not needed. While turning off services is generally done through the OS, closing ports is the role of the firewall.

Firewalls are applications which control network communication. A firewall constrains which external computers can connect to the host computer (IP filtering) and to which ports connections can be made (port filtering). Firewalls control which remote computers can connect to given ports. While some ports are typically meant for general public use (e.g. http/80), communications to others might need to be tightly controlled (e.g. FTP/21, POP3/110, proprietary applications, etc.).

An example of this is the following rule set:

allow connections from everywhere to ports ftp/21, snmp/25, http/80
allow connections from IP1,IP2,IP3 to ports POP/110 (+log all connection attempts)
disallow connections to port mysql/3306 (+log all connection attempts)
disallow connections to all other ports

.

By properly documenting PPS and enforcing "least privileges" by running only necessary services, it goes a long way to keeping systems from becoming infected in the first place. It is also easier to spot anomalous behavior, when traffic appears where it is not expected. This is further enforced by firewall rule sets to narrow the scope of connections allowed.