Global Information Assurance Certification Paper

TCP WRAPPERS—What are they??

Stacy M. Arruda

02/16/01

Introduction:

TCP Wrappers acts much like a soldier at a checkpoint, verifying a host's clearance prior to entry. Simply put TCP Wrappers capitalizes on the client/server relationship necessary for most TCP/IP applications. TCP Wrappers inserts itself into the middle of the relationship and acts as the server until the client/host is authenticated. TCP Wrappers utilizes its access control feature to authenticate hosts. TCP Wrappers does all of this with no overhead to the system and best of all it is free. (TCP Wrappers is available at ftp.cert.org/pub/tools/tcp_wrappers_7.6.tar.gz and ftp.porcupine.org/pub/security/).

History:

In 1990, a Dutch hacker obtained root level access to several computer systems at Eindhoven University of Technology, in Eindhoven, Netherlands. The University was under heavy attack and the hacker caused a considerable amount of damage. The majority of the damage was caused through the execution of the **rm –rf /** command. This command is extremely destructive, in that it has the same effect as the **format** command in a MS-DOS environment. Fortunately for the University an individual named Wietse Venema was employed in the Mathematics and Computer Science Department. He, in response to the attacks, developed a program that could control host access. The program could also track and log intruders. TCP Wrappers, as we know it today, was created.

Operational Breakdown

Prior to TCP Wrappers, systems typically used, one daemon, inetd, as a conduit between hosts and processes. This daemon would sit dormant and wait for network a connection. When the connection was established the daemon would run the appropriate server program. After the host and service were connected the daemon would go back to its dormant state and wait for the next incoming request.

TCP Wrappers requires that the network-server programs are moved to another directory and the TCP Wrapper programs are installed in their place.

TCP Wrappers operates by intercepting and filtering incoming requests for the network services; the popular services are SYSTAT, FINGER, FTP, TELNET, RLOGIN, RSH, EXEC, TFTP and TALK. As mentioned earlier, TCP Wrappers relies on the client/server relationship necessary for most TCP/IP protocols. TCP Wrappers operates by replacing the network service that is being called upon with the TCP daemon (tcpd). After it replaces the network service, it moves the service to another file until the client

can be authenticated. The following is an example of a client/user attempting to FTP to an unprotected server and a server employing TCP Wrappers:

Unwrapped server

client/user -> FTP client -> listening inetd host -> ftp -> transfer files

TCP Wrapped server

client /user -> FTP client -> tcpd -> ftp -> transfer files

TCP Wrappers acts as a blanket around the network services that are specified. Keep in mind though, the wrappers do not have any interaction with the client/user or the server. This is important for two reasons, first the wrappers are application-independent, the same program can protect many kinds of network services and secondly no interaction also means that the wrappers are invisible from the outside (at least to authorized users).

Once the tcp daemon has been activated, it searches for matches against the host/client's IP address and service requested. Next tcpd employs its Access Control function; by searching the host allow and host deny files. The host.allow file determines who is allowed to execute what functions/processes. Conversely, the host.deny file filters out unwanted users/hosts to specified servers/processes. Each file contains, or should contain, a set of rules that specifically allow or deny users, hostnames, networks or services. Tcpd will search the host.allow file first, if there is no rule preventing the host's connection, the request will be turned over to the requested process. If a host is allowed in the *.allow* file and denied in the *.deny* file, the host will be permitted access because tcpd stops at the first match. Access can be controlled per host, per service, or in combinations thereof. TCP Wrapper also logs the client's descriptive data, to include hostname, IP address, type of request and time of request, to the syslog configuration file. The distribution of the logs is determined by the syslog configuration file, (/etc/syslog.conf). The messages generated from the syslog file can be written/sent to files, to the console, forwarded to a @loghost, to a pager or to a printer.

TCP Wrappers is written with built in "hooks" to allow the execution of shell commands when access control rules are triggered. This feature allows the setting of *Booby Traps*. A good example to illustrate this would be the firing of a specific rule and the response to conduct a *reverse finger*, to identify the host attempting to access the service. Below is the language necessary, in the hosts.allow and hosts.deny, to trigger the *reverse finger* on outside tftp connections:

/etc/hosts.allow:

in.tftpd: Local, (your network)

/etc/hosts.deny:

in.tftpd: ALL: /usr/ucb/finger -1 @%h 2>&1 | (file to write results to)

The entry in the host.allow file tells TCP Wrappers to allow tftp connections from hosts within its own domain. The first part of the entry in the host.deny file tells TCP Wrappers to execute a *reverse finger* on all other host requests and the second part of the entry tells the wrapper where to send the results of the *reverse finger*.

Access control can also be set up to connect clients to specific services. These rules could depend on the requested service, the origin of the request and/or what host address the client connects to. Access control is enabled by default.

Other Useful Features

TCP Wrappers has many useful features that have to be enabled, these include banner messages, protection against host name spoofing, protection against sequence number guessing and protection against host address spoofing.

TCP Wrappers has the ability to display a pre-login banner from the wrapped connection-oriented services. The banners can also be configured to appear, with an explanation, before a connection is dropped. All systems, regardless of system content, should have a log on banner to all connection-oriented ports. The banner should include a statement that unauthorized access is prohibited and users are subject to monitoring. The language extensions have to be enabled for TCP Wrappers to support banner messages.

TCP Wrappers has the ability to protect against host name spoofing. In order to activate this option wrappers has to be complied with -DPARANOID turned on. This option attempts to match the hostname and IP address. This lookup via TCP Wrappers can provide additional information about the owner of the connection. TCP Wrappers attempt to verify the client hostname returned by the DNS server by asking for a second opinion. If any thing does not match, or the second DNS opinion is not available, wrappers assumes that one of the two name servers is lying and the connection is refused.. The –DPARANOID option tells wrappers to always lookup and double check the client hostname. If anything does not match in the two checks refuse the request. After the request has been determined to not be genuine, there are still a few options. First, access can be denied immediately and the client's username and host can be mailed to the system administrator. Secondly, access can be denied and a *safe-finger* command can be executed. (*Safe-Finger* is the stealth version of finger, it filters out the data sent by the remote host). The third option would be to allow the host in and monitor their activity.

TCP Wrappers has the ability to protect against sequence number guessing. Sequence number guessing occurs when an intruder/client attempts to guess the

SYN/ACK sequence number of a packet and impersonates a trusted host. TCP Wrappers can be configured to query a client's IDENT server to determine if the client actually sent the request. If the client's IDENT server verifies the request, the request will be processed. If the IDENT server does not verify the client, the request will be denied.

TCP Wrappers has the ability to protect against host address spoofing. The -DKILL_IP_OPTIONS refuse to accept TCP connections with IP source routing options. Editing the Makefile can enable this option.

Installation Options

TCP Wrappers has two installation options. The first is the easy installation and configuration method, which requires no changes to the existing software and configuration files. The second is the advanced installation method, which requires modifications of the inetd configuration file.

The easy installation method has the user move the daemons they want to protect to the REAL_DAEMON_DIR directory and replaces them with copies of the tcpd. Here is an example of how to reset the telnet daemon:

Mkdir REAL_DAEMON_DIR
Mv /sbin/in.telnetd REAL_DAEMON_DIR
Cp tcpd /sbin/in.telnetd

All directories, paths and files used by TCP Wrappers should have read- or read-and-execute-only access (modes 755 of 555); **they must not be writable.**

The advanced installation method does not modify the daemon executables, but the inetd configuration file, *file/etc/inetd.conf*, is modified. Tcpd should be executed in place of the original daemon for each service that you want to protect, passing the original daemon pathname as an argument to tcpd. Here is an example of the standard inetd.conf record for telenet:

telnet stream tcp nowait
root /sbin/in.telnetd /sbin/in.telnetd

Here is the same service modified to support TCP Wrappers:

telnet stream tcp nowait root /sbin/tcpd /sbin/in.telnetd

After modifying the files/services that you want to protect, remember to tell inetd to re-read the file with *kill (process id)-HUP*. As with the easy installation, all files, paths and directories utilized by TCP Wrappers should have read- or read- and execute-only access (modes 755 or 555) **they must not be writable.**

Problems with TCP Wrappers

In some instances UDP and rpc/udp daemons go into a "wait" mode after they have completed a connection. They are waiting just in case another request is initiated. These daemons are classified with the 'wait' option in the inetd configuration file. This classification is not recognized by TCP Wrappers. Also, TCP Wrappers do not work with RPC over TCP because they are registered as rpc/tcp in the inetd configuration file.

In January 1999, an intruder modified TCP Wrappers version 7.6 at its primary FTP site. The modified version of TCP Wrappers contained a Trojan horse that provided intruders, connecting from port 421, root access. Additionally, the trojanized version of TCP Wrappers sent an e-mail to an external address upon compilation. The e-mail identified the site and the account that complied the program.

The software has many bugs and the known system bugs are discussed in the Makefile.

Conclusion:

TCP Wrappers should be an essential tool in every security administrators box. It protects against numerous common UNIX compromises by allowing system administrators to offer vulnerable services to legitimate clients (as long as these clients have static IP addresses) without opening the service to the world. It is relatively easy to install and compile. The "out of the box" default version will work on most UNIX platforms. The program does not exchange data with the client/server process, it only acts as a pass through. It works with both TCP and UDP protocols (covered by inetd daemon process). The access control feature protects connection-oriented services. The banner feature warns potential intruders that they are subject to monitoring. This is essential to successfully pursue either criminal or civil legal actions against the intruder. All of these features add to TCP Wrappers impressive functionality as a security tool.

1. Paul Hahnke, "Trick Hackers with TCP Wrappers"
URL: http://www.unixreview.com/archives/articles/1997/September/9709f1bi.shtml (September 1997)

2. Wietse Venema, "TCP WRAPPER, Network monitoring, access control and booby traps"
URL: ftp://ftp.porcupine.org/pub/security/ (July 15, 1992)

3. Paul Dunne, "Enhancing System Security with TCP Wrappers"
URL: http://www.unixreview.com/administration/articles/9905of1.shtml (May 1999)

4. Index of /doc/tcp_wrappers-7.6
URL: http://etp1.ocs.lsu.edu/doc/tcp_wrappers-7.6 (March 21, 1997)

5. Nalneesh Gaur, "Top Open-Source Security Tools for Unix"
URL: http://www.unixreview.com/administration/articles/9907f1.shtml (July 1999)

6. SANS, "UNIX Auditing" SANS/GIAC Level One course materials (Feb 15, 2001)