TCP Wrappers Release Notes

HP-UX 11i v1

Manufacturing Part Number: 5991-4837
December 2005

U.S.A.

Legal Notices

The information contained herein is subject to change without notice.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the United States

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Trademark Notices

Intel® and Itanium® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX® is a registered trademark of The Open Group.

Contents

Contents

1 What's in This Version

The TCP Wrappers product suite provides an enhanced security mechanism for various services spawned by the Internet Services daemon (`inetd`). TCP Wrappers is available on HP-UX 11i v1 operating system as a Web upgrade.

TCP Wrapper Features

The following are the features of TCP Wrappers:

TCP Wrapper Overview

The Internet Super Server, `inetd` allows a single process to be waiting for multiple services instead of one process for each service. Whenever a connection is established with `inetd` for a service, inetd runs the appropriate server program specified in /etc/inetd.conf and waits for other connections.

`inetd` runs the wrapper program `tcpd` instead of running the server program directly.

When inetd invokes tcpd for a service, it will read the /etc/tcpd.conf file and configure itself to effect its behaviour for different features at runtime.

The tcpd program offers the following features to enforce access control check for a service:

- Access Control

  TCP Wrappers uses /etc/hosts.allow and /etc/hosts.deny files as Access Control Lists (ACLs). These access control files are used to match the client and server entries with the request for a service. These files are based on pattern matching and can be extended via optional extensions like: allowing the spawning of a shell command.

  Each access control file consists of a set of access control rules for different services, which use tcpd.

  An access control rule is of the following form:

  ```
  daemon_list:client_list:option:option:..
  ```

  Where daemon_list contains the list of daemons and client_list contains the list of clients for which this rule is applicable. Each list is a set of items separated by a space. The options are separated by a colon.

  The access control module reads the /etc/hosts.allow and /etc.hosts.deny files before granting or denying access to any service. The files are searched in the following order:

— The /etc/hosts.allow file is checked first. If a daemon-client pair matches an entry in this file, access will be granted.

— The /etc/hosts.deny file is checked. If a daemon-client pair matches an entry in this file, access will be denied.

— If no daemon-client pair match was found in either of the access control files, access will be granted.

Where “daemon” is the name of a network daemon process and “client” is the name and/or address of a host requesting the service.

Access control can be turned off by not providing an access control file.

Examples:

1. To grant access to ftp service to ‘all’, specify the following in /etc/hosts.allow file:

   ```
   ftpd:ALL
   ```

2. To deny access to the host “some.host.name” and all hosts in the domain, “some.domain” to all the services, specify the following entry in the /etc/hosts.deny file:

   ```
   ALL:some.host.name, .some.domain
   ```

3. To grant access to all hosts in the domain “xyz.com” except the host “abc.xyz.com” for telnet service, specify the following entry in the /etc.hosts.allow file:

   ```
   telnetd:.xyz.com EXCEPT abc.xyz.com
   ```

NOTE Refer to hosts_access(5) and hosts_options(5) man pages for more information on the access control language used and the various options provided by ACLs.

- Hostname/Address Spoofing

  The wrapper program offers protection against hosts which pretend as some other host. If any discrepancies are noticed in the client address or name, the wrapper program denies access by logging the information. tcpd also disables the source-routing socket options on every connection that it deals with. UDP services benefit from this protection.

- Client Username Lookups

  tcpd provides information about the owner of the client-side TCP connection using the RFC931 protocol. By default, client username lookup is disabled. If it is enabled through the configuration file /etc/tcpd.conf, tcpd assumes that the client requesting the service supports a RFC931-compliant daemon (like IDENT), running on it.

- Setting Traps

  This feature is the ability to trigger actions on the host which are based on attempted connections. For example, the following rule in /etc/hosts.deny not only rejects the attempt, but also notifies the system administrator whenever a remote site attempts to access your TFTP server:

  ```
  tftpd:ALL:spawn (/usr/bin/safe_finger -1 @%h2>&1| \
  mailx -s "remote tftp attempt" root)
  ```

- Banner Messages

  This feature provides a mechanism to send some message, when an ACL rule is fired. For example, the following rule in /etc/hosts.deny file sends the message present in telnetd file in /tmp/banner directory and denies access whenever a request comes from any host whose address starts with '192.5.2.'.

  ```
  telnetd:192.5.2.:banners/tmp/banner
  ```

  The Banner option does not add any service-specific characters while sending the text to the client as specified in the service protocol. To use the banner option successfully, the banners file must contain the necessary protocol parameters in addition to the actual text.

  For example, in an `ftpd` service, each line in the banners file is not automatically prefixed by the status code (`220-`) as defined in RFC 959 (FILE TRANSFER PROTOCOL). To send the following text to the FTP client:

  ```
  This is a Welcome text to demonstrate the banners
  option in tcpd(1M).
  ```

  HP recommends that you add the protocol-specific response code to the text as follows:

  ```
  220-This is a Welcome text to demonstrate the banners
  220- option in tcpd(1M).
  ```

For the `rlogind` service, a null character (`\0`) must be placed at the beginning of the `rlogind` banner file, as specified in the following example:

```
# echo "\0 Text to demonstrate the banners" > rlogind
# echo "    option in tcpd(1M)." >> rlogind
```

Configuration

TCP Wrappers on HP-UX uses the /etc/tcpd.conf configuration file. This file can be used to set time-out on client username lookups, log level, and action to be taken in case of reverse lookup failure.

What's in This Release

The following are the binaries distributed with this release of TCP Wrappers:

- tcpd

  The tcpd program will log the client hostname address and the remote username who owns the connection, if applicable and perform some additional access control checks. When it is through with all the checks, the wrapper executes the desired server program and exits.

  The following are the methods to use the tcpd program:

  1. Edit the /etc/inetd.conf file as follows:

     ```
     telnet stream tcp nowait root /usr/lbin/telnetd telnetd
     ```

     becomes:

     ```
     telnet stream tcp nowait root /usr/lbin/tcpd \
     /usr/lbin/telnetd telnetd
     ```

     Only the last component of the pathname /usr/lbin/telnetd, telnetd will be used for access control and logging.

     Similar changes will have to be made for other services, which need to be covered by tcpd. You also need to re-configure inetd(1M) process by running the command '`inetd -c`' on the command line to reflect the changes.

NOTE

If the above entry is specified without the absolute path of `telnetd` (/usr/lbin/telnetd), `tcpd` looks for the `telnetd` binary in `/usr/lbin/wrapper` directory.

2. In order to monitor the access to a service, move the original service daemon to the /usr/lbin/wrapper directory and move tcpd in place of the original service daemon. No changes are required to the /etc/inetd.conf file.

   The following commands run in the command line, describe how to enable the ftp service with tcpd:

   ```
   # mkdir /usr/lbin/wrapper
   # mv /usr/lbin/ftpd /usr/lbin/wrapper
   # cp tcpd /usr/lbin/ftpd
   ```

   Whenever any ftp service request comes in, inetd will spawn /usr/lbin/ftpd which is actually the tcpd.Then `tcpd` performs some access control checks before invoking the `ftpd` binary in `/usr/lbin/wrapper` directory.

NOTE

You can perform either of these steps to use `tcpd`.

Refer to tcpd(1M) and tcpd.conf(4) man pages for more information on configuration.

- libwrap.a

  The libwrap.a library provides a set of APIs for stand-alone applications to enforce host access control based on /etc/hosts.allow and /etc/hosts.deny files. The APIs implement a rule-based access control language with optional shell commands that are executed when a rule fires.

  In order to enforce the host access control in a stand-alone daemon, it has to include the tcpd.h header file and link with libwrap.a. This library includes the following APIs:

  — request_init()

This API initializes the request_info structure with information about the client request.

— request_set()

This API updates an initialized request_info structure.

Both request_init() and request_set() APIs take the `request_info` structure and a variable-length list of key-value pairs as input parameters and return the first argument which is the request_info structure defined in the tcpd.h header file. The argument lists are terminated with a zero key value.

— hosts_access()

This API reads the ACLs and returns either '1' or '0' indicating the access granted or denied respectively.

— hosts_ctl()

This API is a wrapper around request_init() and hosts_access() routines. It takes the daemon name, client's hostname, client's address and username as input parameters. The client hostname, address and username arguments should contain valid data or STRING_UNKNOWN defined in the tcpd.h file. The hosts_ctl() API returns zero if access should be denied.

The following are the methods to implement access control checks in a daemon program:

1. Fill the `request_info` structure's variable elements using `request_init ( )` and `request_set ( )` routines and call the `hosts_access( )` routine to verify these elements with ACLs.
2. Call `hosts_ctl ( )` function with appropriate input parameters to check with ACLs.

NOTE Refer to hosts_access(3) man page for more information on these APIs.

- tcpdchk

  tcpdchk is a tool that can be used to examine the validity of entries in the /etc/inetd.conf file and ACLs.

`tcpdchk` examines the tcp wrapper configuration and reports all potential and real problems it can encounter. The program examines the tcpd access control files (by default, these are /etc/hosts.allow and /etc/hosts.deny), and compares the entries in these files against the entries in the /etc/inetd.conf file.

This is executed on the command line as:

```
/usr/bin/tcpdchk [-a] [-d] [-i inet_conf] [-v]
```

Where

"-a" option is used to report access control rules that grant access without an explicit ALLOW keyword.

"-d" option is used to examine the hosts.allow and hosts.deny files in the current directory instead of the default ones.

"-i inet_conf" option is used, if you want to specify a different path for the inetd configuration file instead of the default one, i.e. /etc/inetd.conf.

"-v" option is used to display the contents of each access control rule. The daemon lists, client lists, shell commands, and options are shown in a printable format; this helps you spot any discrepancies between what you want and what this program understands.

NOTE Refer to the tcpdchk(1) man page for more information.

- tcpdmatch

  tcpdmatch is a tool that can be used to simulate the Wrappers daemon program, tcpd's behaviour for a particular host and a particular service.

  tcpdmatch predicts how the tcp wrapper daemon would handle a specific service request.The program examines the tcpd access control tables (default /etc/hosts.allow and /etc/hosts.deny) and prints its conclusion. For maximum accuracy, it extracts additional information from the /etc/inetd.conf file.

  This is executed on the command line as:

  ```
  /usr/bin/tcpdmatch [-d] [-i inet_conf] daemon client
  ```

```
/usr/bin/tcpdmatch [-d] [-i inet_conf] daemon@[server]
[user@]client
```

The second syntax can be used when your server has more than one address or name.

Where

"daemon" is a daemon process name.

"client" is a host name or network address, or one of the 'unknown' or 'paranoid' wildcard patterns.

Optional information specified with the "daemon@server" and "user@client" forms:

"server" is a host name or network address, or one of the 'unknown' or 'paranoid' wildcard patterns. The default server name is 'unknown'.

"user" is a client user identifier. Typically, it is a login name or a numeric user id. The default user name is 'unknown'.

The following example illustrates how tcpd would handle a ftp request from a local system:

```
tcpdmatch ftpd localhost
```

Pretending that the hostname lookup fails, the same request would be handled by tcpd as follows:

```
tcpdmatch ftpd 127.0.0.1
```

To predict what tcpd would do when the client name does not match the client address:

```
tcpdmatch ftpd paranoid
```

NOTE

Refer to tcpdmatch(1) man page for more information.

Refer to hosts_access(5) man page for more information on wildcard patterns.

- try-from

try-from is a utility program that can be used to identify the end-point details related to a connection. This program must be called via a remote shell command as given below to find out if the hostname and the address are properly recognized, and also if the username lookup works.

This is executed on the command line as:

```
# remsh host /usr/bin/try-from
```

try-from prints the following output on its inception:

"client address (%a):"

"client hostname (%n):"

"client username (%u):"

"client info (%c):"

"server address (%A):"

"server hostname (%N):"

"server process (%d):"

"server info (%s):"

NOTE Refer to hosts_access(5) man page for more information on %<letter> expressions.

The client-related information describes how the remote host recognizes the client in terms of address, name, and user name whereas the server-related information describes the remote host.

- safe_finger

  safe_finger is a wrapper program to finger(1) client which offers protection against possible damages to the data sent by the remote finger server. This command accepts all the options supported by finger(1).

NOTE Refer to finger(1) man page for more information.

safe_finger command is executed on the command line as:

```
# /usr/bin/safe_finger -l @xyz.abc.def.com
```

This command prints the user information on the remote host "xyz.abc.def.com".

HP recommends using this program in the implementation of Traps in the access control language of /etc/hosts.allow and /etc/hosts.deny.

NOTE Refer to hosts_access(5) man page for more information on setting Traps.

Troubleshooting

TCP Wrappers daemon logs the information related to a connection and problems encountered, before invoking the original daemon in syslog (default /var/adm/syslog/syslog.log). The logging level parameter can be specified as either 'normal' or 'extended' in the /etc/tcpd.conf file. A value of "extended" will cause the TCP Wrappers daemon to log the ACLs information like: the entry with which the client request is matched and its related options. By default, the value for this entry is "normal", in which case tcpd will log the connection details about refusal or acceptance of the connection. TCP Wrappers suite also provides tools tcpdchk and tcpdmatch to validate the inetd.conf, hosts.allow and hosts.deny entries in the configuration file and to predict how tcpd would handle a specific service request respectively.

Refer to tcpdchk and tcpdmatch sections on page 13 for more information.

2 Installation Information

Read this chapter before installing the TCP Wrappers web upgrade on your system.

System Requirements

The following are the system requirements to install TCP Wrappers:

- Hewlett-Packard 9000 computer
- HP-UX operating system version 11i

Installing TCP Wrappers

TCP Wrappers is available as a web release on HP-UX 11i platform from HP's software depot at www.software.hp.com. After downloading the software package, use the `swinstall` command to install the package on your system. Detailed information on how to configure and use TCP Wrappers can be found in the respective man pages.

Install the web upgrade as per the following steps:

1. Run the following command on the command line.

   ```
   swinstall -s <destination path>
   ```

   Where <destination path> is the absolute path where you downloaded the TCP Wrappers web upgrade depot to.

   A GUI screen appears.

2. Select the TCP Wrappers product in the GUI screen.

3. Go to `Action` menu and choose `Install` option.

   TCP Wrappers on HP-UX 11i is now available for your use.

This version of TCP Wrappers uses a configuration file, `/etc/tcpd.conf` to configure the `tcpd` wrappers daemon. It also uses the access control files, `/etc/hosts.allow` and `/etc/hosts.deny` to enforce access control in the `tcpd` daemon. A sample of these files are provided in the `/usr/newconfig/etc` directory. You can edit these files as per your requirements and copy them to the `/etc` directory.

3 Documentation

The following product documentation is available with this release of TCP Wrappers.

TCP Wrappers Information

The following man pages are distributed with the TCP Wrappers depot available at www.software.hp.com:

1. tcpd(1M)

   This man page describes the tcpd program used to enforce access control for services spawned by inetd.

2. tcpdmatch(1)

   This man page discusses the functioning of the tcpdmatch utility program.

3. tcpdchk(1)

   This man page discusses the tcpdchk program, which is used to check the tcp wrapper configuration.

4. hosts_access(3)

   This man page describes the routines included in the libwrap.a library.

5. hosts_access(5)

   This man page discusses the format of the host access control files.

6. hosts_options(5)

   This man page discusses the optional extensions to the host control language.

7. try-from(1) and safe-finger(1)

   This man page describes the try-from and safe_finger utility programs included in the TCP Wrappers product suite.

8. tcpd.conf(4)

   This man page discusses the tcpd configuration file.

The following RFCs have been implemented in TCP Wrappers:

- RFC 931

4 Known Problems and Limitations

This chapter discusses the known problems and limitations in this release of TCP Wrappers.

Known Problems

There are no known problems in this release of TCP Wrappers.

Limitations

The following are the limitations in this release of TCP Wrappers:

- When a new request comes in, the UDP (rpc/udp) daemons linger around for a while after servicing the request.In the /etc/inetd.conf file, these daemons are registered with the ‘wait’ option. The nowait option is not supported. Only the request which started such a daemon will be seen by the wrappers.
- The wrappers do not work with RPC services over TCP. These services are registered as rpc/tcp in the /etc/inetd.conf file. The only non-trivial service that is affected by this limitation is rexd, which is used by the `on(1)` command.
- Some RPC requests like rwall, rup, rusers et al appear to come from the server host. The client broadcasts its request to all portmap daemons on its network; each portmap daemon in turn forwards the request to a daemon on its own system. However, daemon like rwall assumes that the request is coming from the local host.
- The user name lookup feature of TCP Wrappers uses identd to identify the username of the remote host. By default, this feature is disabled, as identd may appear hung when there are large number of TCP connections. To enable the username lookup, perform the steps as described in the tcpd.conf(4)man page.