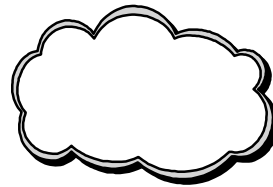


PUT 15 & PUT 16 TCP/IP Enhancements



Jamie Farmer

Rick Schoonmaker



Background

- PUT 11: TCP/IP native stack with IP over CDLC link layer
- PUT 12: Enhancements to increase the usability of TCP/IP Native Stack support
- PUT 13: OSA-Express support, Movable VIPA support and DNS server support
- PUT 14: Enhancements to increase the usability of TCP/IP native stack support



PUT 15 Enhancements

- SNMP agent support: PJ27932
 - ▶ Operator command to display MIB variables: PJ28168 (PUT 16)
- BSD *select* and associated functions: PJ28021
- *Listen* BACKLOG support for INETD: PJ28026
- Operator command to resolve remote host name or IP address: PJ28029
- OSA polling enhancements: PJ28064
- Allow *send/write* of data greater than 32K for TCP sockets: PJ28087
- Wildcard support for TPF host names defined to TPF DNS server: PJ28093



SNMP Agent Support

- Simple Network Management Protocol is an industry-standard protocol that provides a simple way to manage TCP/IP networks.
- Benefits
 - ▶ Centralizes the management of networks that contain a variety of different systems. Networks today contain many different systems, managing all these systems in a coherent framework becomes important.
 - ▶ Centralizes the gathering of performance and error data.
 - ▶ Helps with problem analysis.
 - ▶ Provides notification of important events.



SNMP Agent Support (cont.)

- TPF Support:
 - ▶ SNMP version I
 - ▶ MIB II
 - Standard MIB objects
 - User MIB objects
 - ▶ Traps
 - System Traps
 - User Traps
 - ▶ RFCs
 - RFC 1155 Structure and Identification of Management Information for TCP/IP-based internets
 - RFC 1157 A Simple Network Management Protocol (SNMP)
 - RFC 1213 Management Information Base for Network Management of TCP/IP-based internets: MIB-II
 - RFC 2233 The Interfaces Group MIB using SMIv2.



SNMP Agent Support (cont.)

- SNMP MIB Display Support

ZSNMP DISPLAY UDP

SNMP0018I 10.10.32 SNMP MIB RETRIEVAL DISPLAY

udpInDatagrams : 9436

udpNoPorts : 21

udpInErrors : 0

udpOutDatagrams : 6

udpLocalAddress.0.0.0.0.69 : 0.0.0.0

udpLocalAddress.0.0.0.0.520 : 0.0.0.0

udpLocalPort.0.0.0.0.69 : 69

udpLocalPort.0.0.0.0.520 : 520

END OF DISPLAY



BSD *select* () and Associated Functions

- Original TPF *select*() API was designed to handle a larger number of file descriptors than BSD *select*()
 - ▶ Integer list vs. Bitmap model
- Sockets: Kernel scope (TPF) vs. Process scope (other OS)
- BSD *select*, the related macros, and structures were needed to help ease porting of applications to TPF.



BSD *select* () and Associated Functions

- To port an application using BSD *select*, change all occurrences of the *select*() function to *tpf_select_bsd*()
 - ▶ No parameters need to change in order to support TPF's large socket descriptors or a large number of sockets.
 - ▶ An application should define its own `FD_SETSIZE` before the include of header `sys/time.h` if you want to specify more than 256 sockets on a single *tpf_bsd_select*() call.
- Related macros and declarations
 - ▶ `FD_SET`, `FD_CLR`, `FD_ISSET`, `FD_COPY`, `FD_ZERO` and structure `fd_set`



Listen BACKLOG Support for INETD

- Raised at the last TPF Users Group in several different meetings
- Backlog of 5 was a limitation of offload support
- BACKLOG parameter added to ZINET ADD/ALTER command for WAIT, NOWAIT, and LISTEN TCP server models
- *listen* backlog - number of connection requests that may be queued to wait to be accepted by the TCP server before connection requests are rejected.
 - ▶ Different per server application
 - ▶ Used for short-lived, high-volume connections
 - ▶ Network restart



Listen BACKLOG Support for INETD

```
zinet alter s-ftp backlog-7
CSMP0097I 13.28.58 CPU-B SS-BSS  SSU-HPN  IS-01
INET0012I 13.28.58 SERVER FTP          ENTRY UPDATED
CSMP0097I 13.28.58 CPU-B SS-BSS  SSU-HPN  IS-01
SERVER - FTP          PROCID - B      ACTIVATION - AUTO
PGM    - CFTP        PARM    -
PROTOCOL - TCP      PORT          - 00021  MODEL - NOWAIT
SERVERERRORS - 00000  SERVETIME - 00000  USER  - root
MAXPROC  - 00000    TIMEOUT      - 00000  STATE - NORM
AORLENGTH - 00000  BACKLOG    - 00002
IP - ANY
```

ALTERED TO -

```
SERVER - FTP          PROCID - B      ACTIVATION - AUTO
PGM    - CFTP        PARM    -
PROTOCOL - TCP      PORT          - 00021  MODEL - NOWAIT
SERVERERRORS - 00000  SERVETIME - 00000  USER  - root
MAXPROC  - 00000    TIMEOUT      - 00000  STATE - NORM
AORLENGTH - 00000  BACKLOG    - 00007
IP - ANY
END OF DISPLAY
```



Operator Command to Resolve Remote Host Name or IP Address

■ ZDTCP NSLOOKUP *host*

- ▶ TPF command to resolve either remote host name or IP address

```
zdtcp nsl www.tpfug.com
```

```
CSMP0097I 13.09.04 CPU-B SS-BSS SSU-HPN IS-01
```

```
DTCP0007I 13.09.04 NAME SERVER RESOLUTION DISPLAY
```

```
NAME - www.tpfug.com
```

```
ADDRESSES - 12.42.18.9 12.42.19.9 12.42.16.9  
12.42.17.9
```

```
END OF DISPLAY
```

```
zdtcp nsl 12.42.18.9
```

```
CSMP0097I 13.09.31 CPU-B SS-BSS SSU-HPN IS-01
```

```
DTCP0007I 13.09.31 NAME SERVER RESOLUTION DISPLAY
```

```
NAME - www.tpfug.com
```

```
ADDRESSES - 12.42.18.9
```

```
END OF DISPLAY
```



OSA Polling Enhancements

- Enhancement to current OSA-Express polling code to increase performance and decrease the number of dropped packets under certain conditions:
 - ▶ Not getting kicked off enough in shared PR/SM environment
 - ▶ Overloaded VM system
 - ▶ Polling not getting called enough in single I-stream system with long-running ECBs that do not give up control.
- Enhancement helps in three ways:
 - ▶ Polling kicked off on a more consistent basis
 - ▶ Free up the buffers used for passing data between TPF and the OSA-Express card faster so that packets are not thrown away
 - ▶ New keypoint 2 parameter added to specify the number of buffers to allocate for the OSA-Express card to use to pass IP packets.
- Suggested OSA microcode level: 4.19 or higher



Allow *send/write* of Data Greater Than 32K for TCP Sockets

- 32K was a limitation of offload support
- No code or behavior changes to existing applications
- TPF now allows a *send/write* of up to 1G of data
- Available for TCP sockets only, not UDP or RAW
- Previously, we had a send all or nothing approach
- If *send/write* size is less than or equal to IP send buffer size
 - ▶ There is no change to existing processing
- If *send/write* size is greater than IP send buffer size
 - ▶ TPF will take as much data as possible and then return to the application the amount of data that was sent
 - ▶ Application must then issue more *sends/writes* for the remaining data, and the application must serialize the sends if the socket is being shared by multiple ECBs.



Allow *send/write* of Data Greater Than 32K for TCP Sockets

- If *send/write* size is less than or equal to IP send buffer size
 - ▶ Buffer size - 50K Send - 10K
 - TPF will send out all 10K
- If *send/write* size is greater than IP send buffer size
 - ▶ Buffer size - 50K Send - 100K
 - TPF will send 50K and return to the application that it sent 50K
 - Application must then do another send for the remaining 50K



Wildcard Support for TPF Host Names Defined to TPF DNS Server

- Update to TPF's DNS server support that shipped on PUT 13 to be able to support wildcards (*) in the host file
- Customer requirement that allows users to define many host names with a single entry
- Example of host file:

rschoon.tpf.com 9.17.1.3

***.tpf.com 9.17.1.1 9.17.1.2**



PUT 16 Enhancements

- TCP fast retransmit support: PJ28344
- TCP/IP packet filtering support: PJ28213
 - ▶ Packet filtering firewall function
 - ▶ Includes Enhanced Diagnostics using IP trace
 - ▶ Provides Intrusion Detection Services (IDS)
- Network Services Database support: PJ28195
 - ▶ Provides standard *getservbyname()* & *getservbyport()* APIs
 - ▶ Ability to define the network priority for outbound messages on a per application basis using differentiated services
 - Ability to define a default priority: PJ28034 (PUT 15)
 - ▶ Ability to collect messages for data collection by TCP/IP application
 - Ability to collect input and output messages
 - Ability to count messages in data collection by TCP/IP application



TCP Fast Retransmit Support

- Enhancement to the original TCP architecture
- Detects lost packets in the network faster
- Defined in RFC 2001
- TPF implemented RFC 2001 and also other Fast Retransmit algorithms.



Enhanced Diagnostics and Packet Filtering Firewall Support

PJ28213



Enhanced Diagnostics

- TPF's IP trace facility has been updated to supply a reason code when an exception condition is associated with the packet
- Includes system-wide IP trace, individual IP trace, and offline IP trace facility
- Offline trace supplies a new parameter to search for specific reason codes
- Ability to display data in the offline trace in ASCII



Example with Enhanced Diagnostics

RWI-02 IPCCW-D1 SOURCE IP-9.117.241.12 DEST IP-9.117.241.11 LEN-48
TOD-B76A278B027F5A42 PROTOCOL-06 (TCP) SOURCE PORT-1024 **DEST PORT-7777**
SEQ-2024780421 WINDOW-65535 URGENT OFFSET-0
TCP FLAG BYTE-02 (SYN)

REASON CODE - SERVER NOT ACTIVE

IP HEADER 45000030 27390000 3C06628D 0975F10C 0975F10B
TCP HEADER 04001E61 78AFB285 00000000 7002FFFF 41630000 020405D4 01030304
RWI-01 IPCCW-D1 SOURCE IP-9.117.241.11 DEST IP-9.117.241.12 LEN-40
TOD-B76A278B08C45A01 PROTOCOL-06 (TCP) SOURCE PORT-7777 DEST PORT-1024
SEQ-0 ACK-2024780422 WINDOW-0 URGENT OFFSET-0
TCP FLAG BYTE-14 (ACK, RST)

REASON CODE - SERVER NOT ACTIVE

IP HEADER 450A0028 7FFC0000 3C0609C8 0975F10B 0975F10C
TCP HEADER 1E610400 00000000 78AFB286 50140000 6D370000
RWI-02 IPCCW-D1 SOURCE IP-9.117.241.12 DEST IP-9.117.241.11 LEN-48
TOD-B76A279FD3C1CF2F PROTOCOL-06 (TCP) SOURCE PORT-1025 **DEST PORT-7777**
SEQ-2046603721 WINDOW-65535 URGENT OFFSET-0
TCP FLAG BYTE-02 (SYN)

REASON CODE - SERVER NOT ACTIVE

IP HEADER 45000030 273C0000 3C06628A 0975F10C 0975F10B
TCP HEADER 04011E61 79FCB1C9 00000000 7002FFFF 40D10000 020405D4 01030304



Packet Filtering Support

- Provides standard packet filtering firewall function
- Added security for TCP/IP applications
- Compares Source and Destination against user-defined filter rules
- Rules determine whether to process the message, discard the message, or reject the message
 - ▶ Rejecting the message includes sending either a TCP RST or an ICMP error message.



Defining the IP Packet Filtering Rules Table

- User will create packet filtering rules file, */etc/iprules.txt*, on the TPF file system.
- The */etc/iprules.txt* file can be refreshed into core storage and used immediately using the ZFILT REFRESH command.
- The file is also refreshed during cycle-up to 1052 state.



Sample */etc/iprules.txt* File

```
ACTION-ALLOW FROM-9.117.249.0/24 # Customer network
ACTION-ALLOW FROM-9.117.236.0/24 PORT-350
ACTION-ALLOW FROM-9.117.241.32/32 PORT-1414
# Denies all packets to an HTTP server
ACTION-DENY PORT-80
# Denies all ICMP PING request packets
ACTION-DENY PROTO-ICMP ICMPATYPE-8
DEFAULT-REJECT #Default is to reject packets
```



Order of Rules

- The order of the rules coded in the file is important.
- When a packet arrives, the rules in the table will be searched sequentially.
- In the following example the second rule will never apply to an inbound packet.
 - ▶ ACTION-ALLOW FROM-9.117.249.0/24
 - ▶ ACTION-REJECT FROM-9.117.249.50/32 PORT-1414



Performance Considerations

- Minimum overhead for TCP traffic
- TCP messages, other than connection requests, bypass the packet filtering code.
 - ▶ This is done because remote was already verified when connection started.
- For non-TCP traffic, the more rules you define the more overhead there will be.
 - ▶ Most commonly applied rules should be at the top of the rules table.



Displaying the IP Rules Table

- New ZFILT DISPLAY command to display table

ZFILT DISPLAY

CSMP0097I 13.25.05 CPU-B SS-BSS SSU-HPN IS-01

FILT0001I 13.25.05 DISPLAY PACKET FILTERING RULES

RULE	ACTION	REMOTE NETWORK	PORT	PROTO	ICMPTYPE	PACKETS
1	ALLOW	9.117.249.0/24				2435
2	ALLOW	9.117.236.0/24	350			457852
3	ALLOW	9.117.241.32/32	1414			80
4	DENY		80			12
5	DENY			ICMP	8	8
DEF	REJECT					467

END OF DISPLAY



Intrusion Detection

- Determine if the firewall has been rejecting packets using ZFILT DISPLAY
- Search the offline trace for all packets that have been "DISCARDED BY FIREWALL" or "REJECTED BY FIREWALL"
- Using the offline trace search, you can determine which remote users are attempting to access TPF applications they are not authorized to.



Sample IPTPRT Output

RWI-02 IPCCW-D1 **SOURCE IP-9.117.211.52** DEST IP-9.117.241.12 LEN-48
TOD-B6FA5951E20BF04E PROTOCOL-06 (TCP) SOURCE PORT-1029 DEST PORT-80
SEQ-2491461275 WINDOW-65535 URGENT OFFSET-0
TCP FLAG BYTE-02 (SYN)

REASON CODE - DISCARDED BY FIREWALL

IP HEADER 45000030 A70E0000 3906DD8E 0975F934 0975F10C
TCP HEADER 04050586 9480AE9B 00000000 7002FFFF 30FD0000 02040F00 01030304

RWI-02 IPCCW-D1 **SOURCE IP-9.117.211.52** DEST IP-9.117.241.12 LEN-48
TOD-B6FA5954A6842C69 PROTOCOL-06 (TCP) SOURCE PORT-1029 DEST PORT-80
SEQ-2491461275 WINDOW-65535 URGENT OFFSET-0
TCP FLAG BYTE-02 (SYN)

REASON CODE - DISCARDED BY FIREWALL

IP HEADER 45000030 A70E0000 3906DD8E 0975F934 0975F10C
TCP HEADER 04050586 9480AE9B 00000000 7002FFFF 30FD0000 02040F00 01030304

RWI-02 IPCCW-D1 **SOURCE IP-9.117.211.52** DEST IP-9.117.241.12 LEN-48
TOD-B6FA595C98D4E321 PROTOCOL-06 (TCP) SOURCE PORT-1029 DEST PORT-80
SEQ-2491461275 WINDOW-65535 URGENT OFFSET-0
TCP FLAG BYTE-02 (SYN)

REASON CODE - DISCARDED BY FIREWALL

IP HEADER 45000030 A70E0000 3906DD8E 0975F934 0975F10C
TCP HEADER 04050586 9480AE9B 00000000 7002FFFF 30FD0000 02040F00 01030304



Network Services Database Support

PJ28195



Network Services Database

- Database of TCP/IP server application names and their associated port number and protocol
 - ▶ Ability to use standard *getservbyname()* and *getservbyport()* APIs
- Defined in the */etc/services* file
- TPF extended the network services database
 - ▶ Ability to define the network priority for outbound messages on a per application basis using differentiated services
 - ▶ Ability to count messages in data collection by TCP/IP application



Differentiated Services Support

- Ability to use a Quality of Service (QoS) differentiated services codepoint for each application
 - ▶ RFC 2474 and RFC 2475
 - ▶ Done by coding the TOS parameter in the applications entry in the /etc/services file
- If application is not defined in the NSD or the TOS parameter not coded, TPF will use default value
 - ▶ Default value defined by the IPTOS parameter in keypoint 2.
 - ▶ Default TOS for outbound packets was shipped on PUT 15 (PJ28034).



Data Collection/Reduction

- Before PUT 16:
 - ▶ Incremented the high-speed message counter for every socket read type API
- With PUT 16:
 - ▶ High-speed message counter is no longer updated
 - ▶ New counter for TCP/IP input messages
- The following is collected per TCP/IP application defined in */etc/services* file:
 - ▶ Input and Output packets per second
 - ▶ Input and Output bytes per second
 - ▶ Input and Output messages per second



TCP/IP Message Counting

- Unclear what a message is in TCP/IP
 - ▶ Only the application knows what a message is
- System applications (MQ, MATIP, etc) have been updated to increment appropriate data collection counters
- User applications should be updated to update data collection counters
 - ▶ Application can use new C/C++ function `tpf_tcpip_message_cnt` to update the data collection message counters



Weighting of Messages

- User can specify the weight of each message
 - ▶ User must code `tpf_tcpip_message_cnt` if weight is specified
- If the weight parameter is not coded in the `/etc/services` file for the application:
 - ▶ Every socket send type API is counted as an output message
 - ▶ Every socket read type API is counted as an input message
- Specifying `WEIGHT-100` means each message counts as one weighted message
 - ▶ Specifying `WEIGHT-200` means each message counts as two weighted messages
 - ▶ Specifying `WEIGHT-50` means each message counts as a half of a weighted message



Defining the Network Services Database */etc/services* File

- The */etc/services* file can be refreshed into core storage and used immediately using the ZIPDB REFRESH command.
- The file is also refreshed during cycle-up to 1052 state.
- If applications were defined in the old */etc/services* file on refresh, the counters will be carried over.



Sample */etc/services* File

```
dns          53/udp      tos-20 weight-100    #DNS
tftp         69/udp      weight-100     #Trivial File Transfer
http        80/tcp      weight-100     #World Wide Web
snmp        161/udp     weight-100     #SNMP
snmp-trap   162/udp     weight-100     #SNMP trap
matipa      350/tcp     weight-100     #MATIP Type A
matipb      351/tcp     weight-100     #MATIP Type B
https       443/tcp     weight-100     #Secure HTTP
rip         520/udp     weight-50      #RIP
mq          1414/tcp    tos-20 weight-500     #MQ
res0        5001/tcp    #User Reservation Appl
```



Summary/Questions

- Increases Performance
- Provides added security
- Provides network management
- Socket API extensions
- Enhanced Diagnostics
- Enhanced operator commands



PUT 15/16 Maintenance APARs

- PJ28143 (PUT 15): SNMP CTL-1
- PJ27968 (PUT 15): Duplicate File Descriptors
- PJ28161 (PUT 16): AOA CTL-3
- PJ28197 (PUT 16): CTL-C socket inactivation
- PJ28233 (PUT 16): CTL-4 from socket sweeper
- PJ28237 (PUT 16): CTL-1 IPMT corruption
- PJ28360 (PUT 16): AOR/Read timing problem
- PJ28460 (PUT 16): AOR CTL-3
- PJ28479 (PUT 17): OSA Polling fix

