Reading a TCP/IP for VSE trace
Session E18

Leo J Langevin
Leo@E-VSE.COM

RETURN TO INDEX

The commands – Part 1

```
DEFINE TRACE,ID=id,IP=x.x.x.x[,PORT=xxxx]
SEGMENT
DUMP TRACES
DELETE TRACE,ID=id
SEGMENT
```

The commands – Part 2

```
SUSPEND 6
REUSE OFF
DUMP GARBAGE
RESUME 6
REUSE ON
```

The commands – part 3 of 3

```
DEFINE MISDIRECT,ID=id,FROMIP=,TOIP=
DELETE MISDIRECT,ID=id
```

Typical email connection

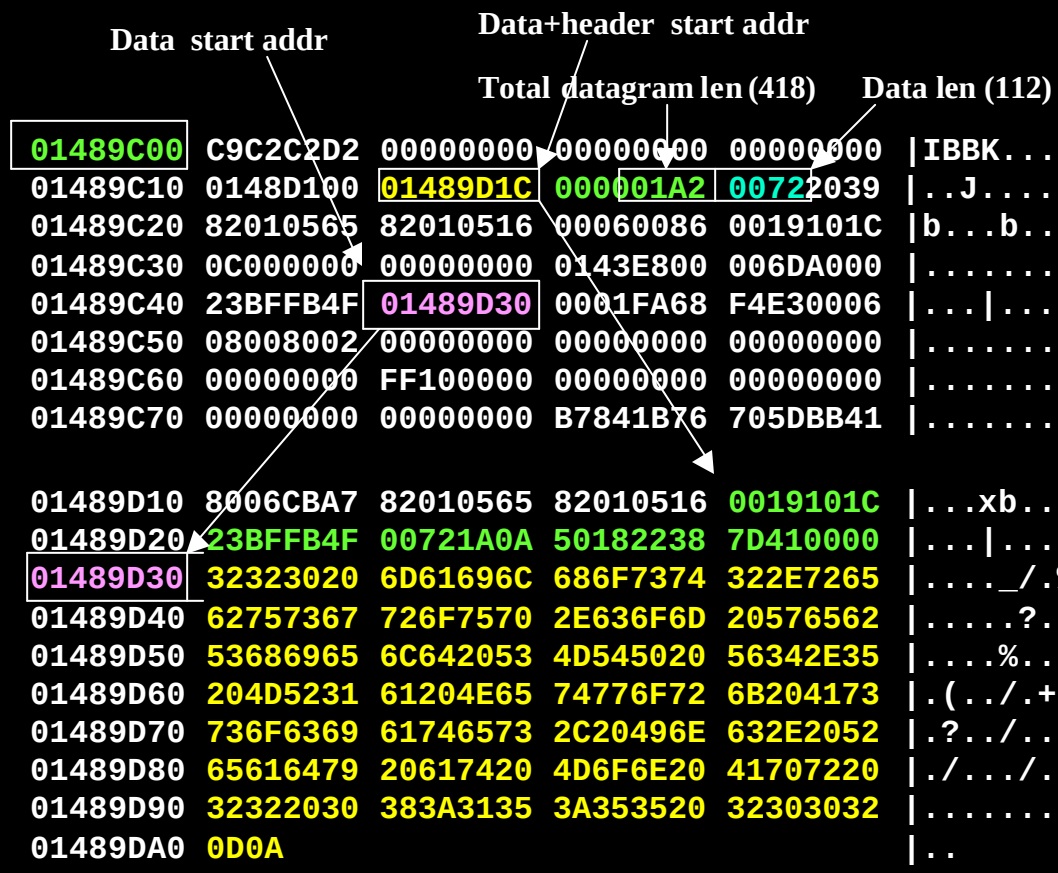
```
01489C00 C9C2C2D2 00000000 00000000 00000000 |IBBK............|................| 00342C00
01489C10 0148D100 01489D1C 000001A2 00722039 |..J........s....|.H...H.......r 9| 00342C10
01489C20 82010565 82010516 00060086 0019101C |b...b......f....|...e............| 00342C20
01489C30 0C000000 00000000 0143E800 006DA000 |.........Y._..|.........C...m..| 00342C30
01489C40 23BFFB4F 01489D30 0001FA68 F4E30006 |...|........4T..|#..O.H.0...h....| 00342C40
01489C50 08008002 00000000 00000000 00000000 |................|................| 00342C50
01489C60 00000000 FF100000 00000000 00000000 |................|................| 00342C60
01489C70 00000000 00000000 B7841B76 705DBB41 |.........d...)..|...........vp|.A| 00342C70
01489C80 00000000 00000000 B7841B76 705B1F41 |.........d...æ..|...........vp$.A| 00342C80
01489C90 01489D08 01489D1C 02004000 00317201 |................|.H...H....@..1r.| 00342C90
01489CA0 00000000 00000000 23BFFBC1 82010516 |...........Ab...|........#.......| 00342CA0
01489CB0 01489D30 00000072 00000000 009027DC |................|.H.0...r......'.| 00342CB0
01489CC0 00008000 01408380 00000000 00000000 |......c.........|.....@..........| 00342CC0
01489CD0 00000200 00000000 00000000 00000000 |................|................| 00342CD0
01489D00 AAAA0300 00000800 4500009A 20394000 |................|........E... 9@.| 00342D00
01489D10 8006CBA7 82010565 82010516 0019101C |...xb...b.......|.......e........| 00342D10
01489D20 23BFFB4F 00721A0A 50182238 7D410000 |...|....&...'...|#..O.r..P."8}A..| 00342D20
01489D30 32323020 6D61696C 686F7374 322E7265 |...._/.%.?......|220 mailhost2.re| 00342D30
01489D40 62757367 726F7570 2E636F6D 20576562 |.....?....?_....|busgroup.com Web| 00342D40
01489D50 53686965 6C642053 4D545020 56342E35 |....%...(.&.....|Shield SMTP V4.5| 00342D50
01489D60 204D5231 61204E65 74776F72 6B204173 |.(../.+...?.,...| MR1a Network As| 00342D60
01489D70 736F6369 61746573 2C20496E 632E2052 |.?../......>....|sociates, Inc. R| 00342D70
01489D80 65616479 20617420 4D6F6E20 41707220 |./.../..(?>.....|eady at Mon Apr | 00342D80
01489D90 32322030 383A3135 3A353520 32303032 |................|22 08:15:55 2002| 00342D90
01489DA0 0D0A                               |..              |..              | 00342DA0
```

The IBBK header

```
01489C00 C9C2C2D2 00000000 00000000 00000000 |IBBK............|................| 00342C00
01489C10 0148D100 01489D1C 000001A2 00722039 |..J.......s....|.H...H.......r 9| 00342C10
01489C20 82010565 82010516 00060086 0019101C |b...b......f....|...e............| 00342C20
01489C30 0C000000 00000000 0143E800 006DA000 |..........Y._..|.........C...m..| 00342C30
01489C40 23BFFB4F 01489D30 0001FA68 F4E30006 |...|........4T..|#..O.H.0...h....| 00342C40
01489C50 08008002 00000000 00000000 00000000 |................|................| 00342C50
01489C60 00000000 FF100000 00000000 00000000 |................|................| 00342C60
01489C70 00000000 00000000 B7841B76 705DBB41 |.........d...)..|...........vp|.A| 00342C70
```

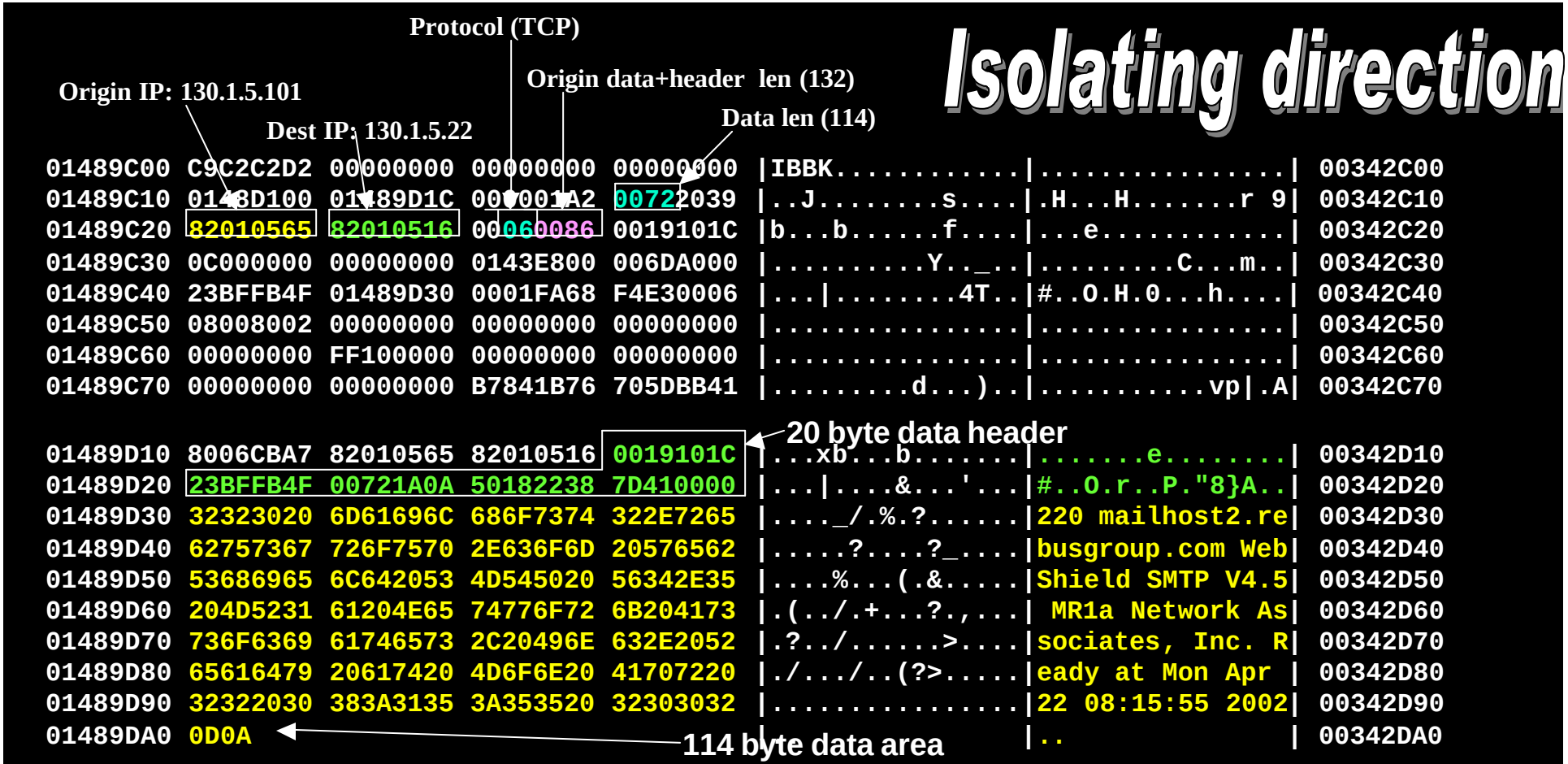
Each entry is known as an internal bus block of data, or an IBBLOCK. Each one is dumped in LIFO sequence when you issue the “DUMP TRACES” command. This, and the next 4 fullwords, (Next block in the chain, next sub-block in the chain, next retransmit in the chain, and the pointer to the garbage chain) are of no real interest.

How to find the data area



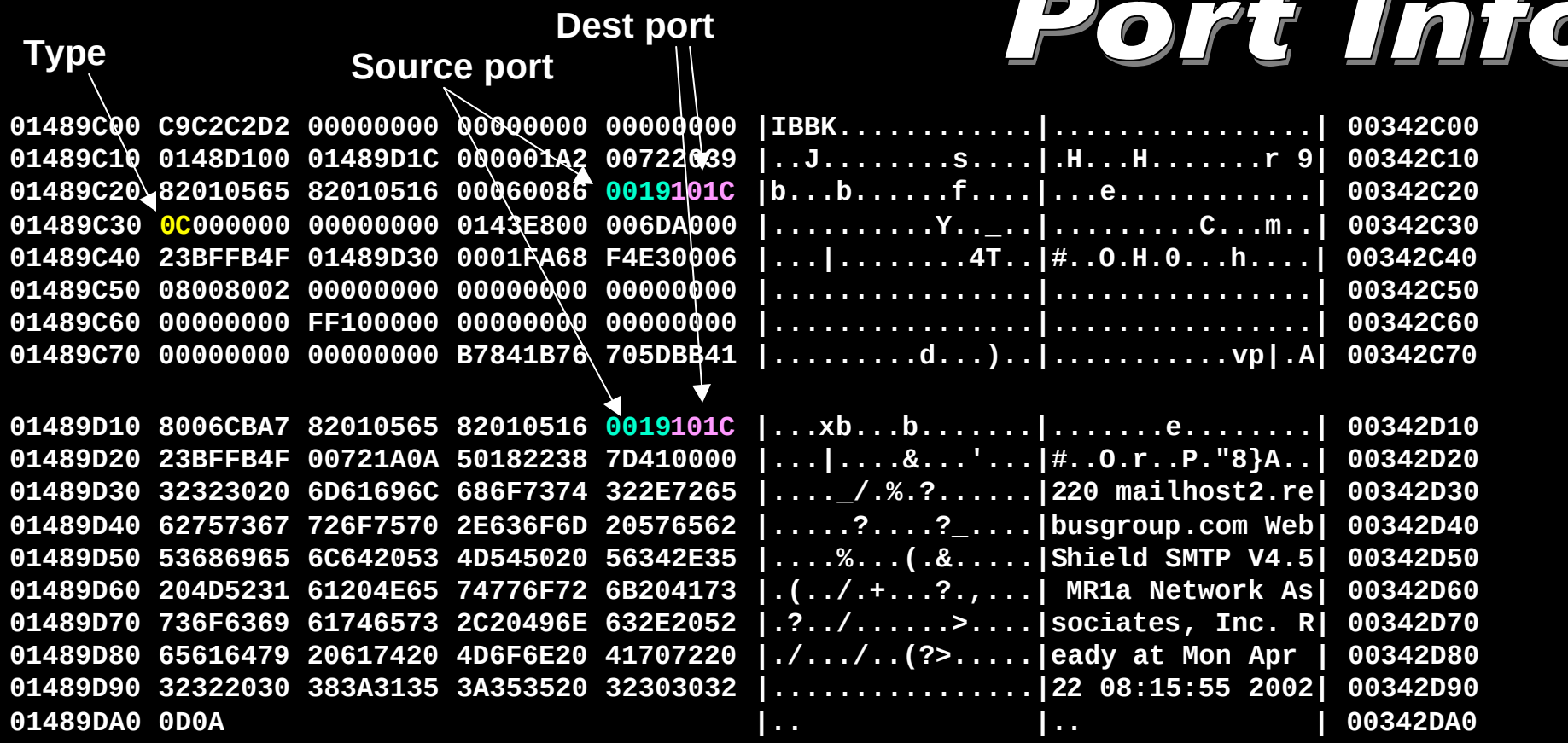
The next field points to the actual data being passed to the stack, which is several bytes of header information followed by the actual data. The Email client doesn't see the header information. That is for the stack to use. You can also see that the start address for this datagram is X'01489C00' and there is a total length of X'1A2', which tells us that the last address should be X'01489DA2'. We also see that the actual data is only X'72' long, so X'1489DA2' – X'72' = X'1489D30', which is where the actual data that is passed to the application begins. And as you can see, the datagram also tells use where the actual data begins.

Isolating direction



After the unique half-word identifier, we see three full-words. The first is the IP address where this datagram originated. The second is the destination IP address. We then have 1-byte of filler followed by the type of protocol being used. The value is: 1=ICMP, 2=IGMP, 6=TCP, or 17 (X'11')=UDP. As you can see in this example, we were doing a TCP connection. The next half-word is the length of actual data being sent to the application, which, in this case is X'86' or 134 bytes. If we subtract that value from the previous X'1A8' (114), which was the total of the data + the header, we get 20 bytes, or the 5-full-word IP header that is being passed with the actual data.

Port Info

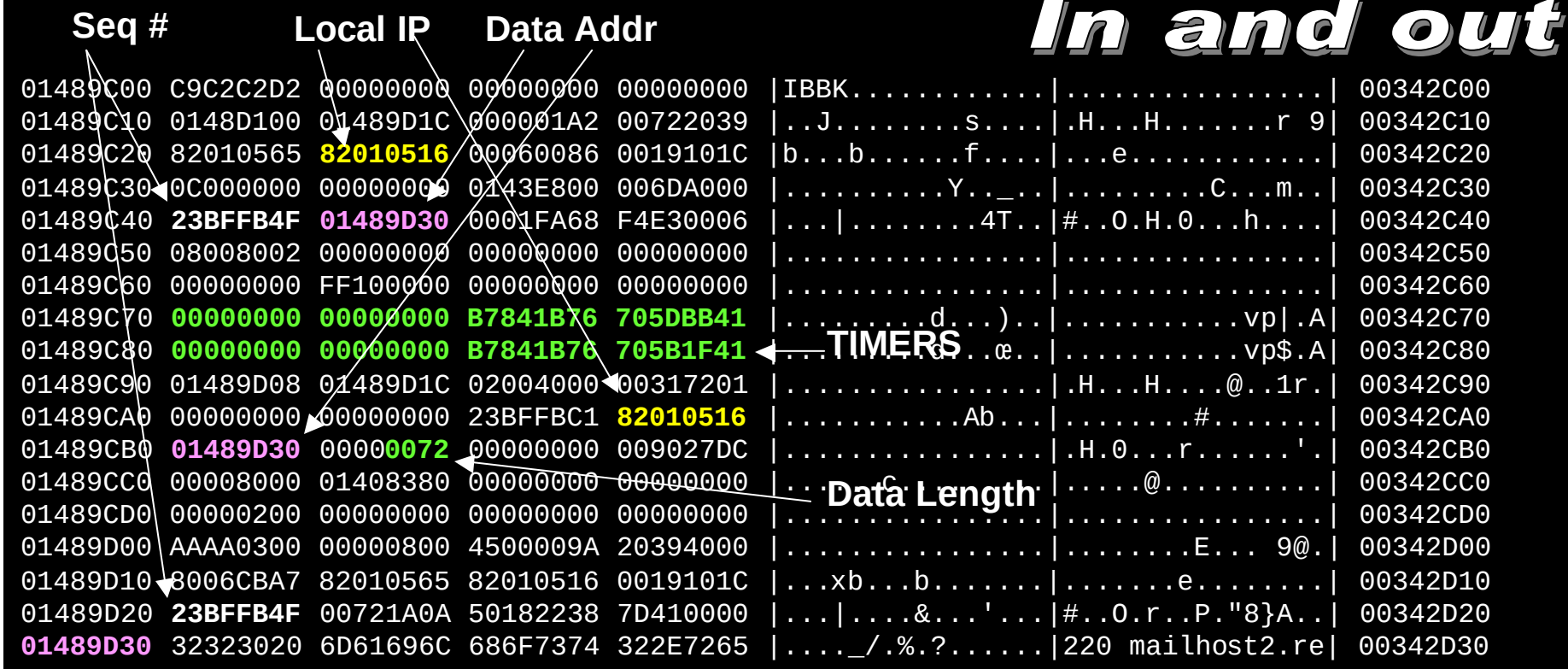


Next is a pair of half-words, giving the port number of the source and the destination. This information is also included in the data header. The data is coming from an SMTP server ad port X'19' or 25 (the standard SPTM port), while the VSE client is using port X'101C' or 4124. (This number is always very high and decrements). Now the next byte tells us what this datagram is doing. In our example, it is a 12, or X'0C'. (see next slide) This means that is is an inbound TCP datagram, coming from outside of VSE to VSE.

Datagram type list (decimal)

- 1=Raw1
- 2=Raw2
- 3=Prepare1
- 4=Prepare2
- 5=IPBlock (link)
- 6=IPBlock (retrans)
- 7=ARP-In
- 8=ARP-Out
- 10=IP-in
- 11=IP-out
- 12=TCP-in
- 13=TCP-out
- 14=UDP-in
- 15=UDP-out
- 16=ICMP-in
- 17=ICMP-out
- 18=IGMP-in
- 19=IGMP-out
- 20=IEEE 802.2 in
- 21=IEEE 802.2 out
- 30=Socket block
- 98=Retransmitted
- 99=Delete

In and out




```
01489C00 C9C2C2D2 00000000 00000000 00000000 |IBBK............|................| 00342C00
01489C10 0148D100 01489D1C 000001A2 00722039 |..J........s....|.H...H.......r 9| 00342C10
01489C20 82010565 82010516 00060086 0019101C |b...b......f....|...e............| 00342C20
01489C30 0C000000 00000000 0143E800 006DA000 |.........Y..._..|.........C...m..| 00342C30
01489C40 23BFFB4F 01489D30 0001FA68 F4E30006 |...|........4T..|#..O.H.0...h....| 00342C40
01489C50 08008002 00000000 00000000 00000000 |................|................| 00342C50
01489C60 00000000 FF100000 00000000 00000000 |................|................| 00342C60
01489C70 00000000 00000000 B7841B76 705DBB41 |.........d...)..|............vp|.A| 00342C70
01489C80 00000000 00000000 B7841B76 705B1F41 |.....!.!.d..œ..|............vp$.A| 00342C80
01489C90 01489D08 01489D1C 02004000 00317201 |................|.H...H....@..1r.| 00342C90
01489CA0 00000000 00000000 23BFFBC1 82010516 |............Ab...|........#.......| 00342CA0
01489CB0 01489D30 00000072 00000000 009027DC |................|.H.0...r......'.| 00342CB0
01489CC0 00008000 01408380 00000000 00000000 |.....C.....C....|.....@..........| 00342CC0
01489CD0 00000200 00000000 00000000 00000000 |................|................| 00342CD0
01489D00 AAAA0300 00000800 4500009A 20394000 |................|........E... 9@.| 00342D00
01489D10 8006CBA7 82010565 82010516 0019101C |...xb...b.......|.......e........| 00342D10
01489D20 23BFFB4F 00721A0A 50182238 7D410000 |...|....&...'...|#..O.r..P."8}A..| 00342D20
01489D30 32323020 6D61696C 686F7374 322E7265 |...._/.%.?......|220 mailhost2.re| 00342D30
```

We'll skip over the internal block pointers and get to the next field that we pass to each IP, which is the unique sequence number of this datagram. The next full-word is the address where the data that the application sees actually begins. The next pair of full-words is the MAC address of the device on VSE that is being used, and an assorted other hardware related bytes. The four double-words that are highlighted are the retransmission time, garbage time, departure time, and arrival time. This is in STCK format. As you can see in this example, this datagram arrived and moments later a copy of it was marked for the garbage collector, which handles time-outs and the like. This datagram arrived fine.

So what do we know so far?

- This is a TCP connection with an incoming datagram coming from port 25, which is the standard SMTP server port.
- It can from another computer with a different IP address, but within the same subnet.
- It is sending 112 bytes of data (which we can see in ASCII).
- Since it's an SMTP server, it is likely talking to an SMTP client.
- This client is running on VSE and is listening on port 4124.

Now let's look at the incoming datagram without the IBBLOCK headers, but with the application header and data only.

Looking at the data and header

Here is the header…

```
01489D10 8006CBA7 82010565 82010516 0019101C |...xb...b.......|.......e........| 00342D10
01489D20 23BFFB4F 00721A0A 50182238 7D410000 |...|....&...'...|#..O.r..P."8}A..| 00342D20
```

And here is the data…

```
01489D30 32323020 6D61696C 686F7374 322E7265 |...._/.%.?......|220 mailhost2.re| 00342D30
01489D40 62757367 726F7570 2E636F6D 20576562 |.....?....?_....|busgroup.com Web| 00342D40
01489D50 53686965 6C642053 4D545020 56342E35 |....%...(.&.....|Shield SMTP V4.5| 00342D50
01489D60 204D5231 61204E65 74776F72 6B204173 |.(../.+...?.,...| MR1a Network As| 00342D60
01489D70 736F6369 61746573 2C20496E 632E2052 |.?../......>....|sociates, Inc. R| 00342D70
01489D80 65616479 20617420 4D6F6E20 41707220 |./.../..(?>.....|eady at Mon Apr | 00342D80
01489D90 32322030 383A3135 3A353520 32303032 |................|22 08:15:55 2002| 00342D90
01489DA0 0D0A                                 |..              |..              | 00342DA0
```

The data obviously gives us some important clues concerning what is actually going on, telling us some server information as part of the handshaking, but let's focus on the 20 bytes of header information.

Taking the header apart – part 1

Header: X'0019101C23BFFB4F00721A0A501822387D410000'

Let's look at the layout:

```
Source port:        X'0019'     (25)
Destination port:   X'101C'     (4124)
Sequence number:    X'23BFFB4F' (Current block pos. in the overall msg)
Acknowledgment #:   X'00721A0A' (Expected next sequence #. Last # + 1)
```

There isn't too much that you can use there that you couldn't get elsewhere. But let's look at the next full-word.

Header: X'0019101C23BFFB4F00721A0A50182238 7D410000'

X'2238' is the window size, or the maximum amount of data to send at any one time.

Taking the header apart – part 2

Header: X'0019101C23BFFB4F00721A0A**5018**22387D410000'

X'5018' is B'**0101** 000000 **011000**'

Now, the first 4 bits (**0101**) indicates the number of full-words in the TCP header. (B'0101="5", and 5*4 = 20 bytes). The next 6 bits (000000) are reserved. The next 6 bits, however, contain actual protocol flags. In this case, it's B'**011000**'. These flags (in order of their position) are:

URGENT (Off) – Often ignored by most stacks
ACK (On) – This is in response to a previous request.
PUSH (On) – Send data right away. Don't buffer it. We want immediate acknowledgment.
RESET (OFF) – Used to break the connection.
SYNCHRONIZE NUMBERS (OFF) - Used when establishing connection
FINISH (OFF) – If on, it's the end of transmission.

The last pair of half-words contains a checksum value and an URGENT

Final things to note in a trace:

- Only look at datagrams that are relevant to your task. If the DEST/SOURCE IP addresses don't match, ignore them.
- If the port number is fairly low, it indicates a standard protocol, and if it's not what you expect, discard that too.
- Only trace what you need to trace.
- Compare the SOURCE/DEST IP address to the VSE IP address in order to determine direction of flow.
- Use EBCDIC/ASCII info to the right for a quick diagnosis.
- If a connection breaks, first check if it ever successfully connected, and check the FIN setting.
- Some traffic, such as "keep alive" will come through at regular intervals.
- If you see an eye catcher for "/MAILBOX", this is Microsoft message flooding, and it can be ignored.
- Utility to make it easier?

RETURN TO INDEX