

# Wireshark Dissectors – Advanced

June 17, 2010

**Gerald Combs**

Lead Developer | Wireshark

**SHARKFEST '10**

Stanford University

June 14-17, 2010



SHARKFEST '10

# Protocol Preferences

- Uints, Booleans, Enums, Strings, Ranges
- General registration
  - Protocol + Callback
- Preference registration
  - Name
  - Data pointer (usually global)
- Stored in main prefs file
- See also: UATs

# Preferences Example

```
static guint g_xyzyy_tcp_port = TCP_PORT_XYZZY;  
  
proto_xyzyy = proto_register_protocol(...);  
  
xyzyy_module = prefs_register_protocol(proto_xyzyy,  
    proto_reg_handoff_xyzyy);  
  
prefs_register_uint_preference(  
    xyzyy_module, "tcp.port", "Xyzyy TCP Port",  
    "TCP port for xyzyy messages", 10, &g_xyzyy_tcp_port);
```

# Example

Gopher Preferences

# Keeping State

- Order not guaranteed
  - pinfo->fd->flags.visited
- Within your dissector
  - Normal C variables
- Up & down the stack
  - pinfo->private\_data
- Across calls
  - p\_add\_proto\_data
  - Conversations

# Protocol Data Example

```
per_packet_info = p_get_proto_data(pinfo->fd, proto_vnc);  
  
if(!per_packet_info) {  
    per_packet_info = se_alloc(sizeof(vnc_packet_t));  
  
    per_packet_info->state = per_conversation_info->vnc_next_state;  
    per_packet_info->preferred_encoding = -1;  
  
    p_add_proto_data(pinfo->fd, proto_vnc, per_packet_info);  
}  
  
/* Packet dissection follows */  
switch(per_packet_info->state) {
```

# Conversations

- Packets between address:port pairs
- Versatile creation:  
`find_conversation + conversation_new`
- Easy creation:  
`find_or_create_conversation`
- Adding / getting data  
`conversation_add_proto_data`  
`conversation_get_proto_data`

# Conversation State Example

```
/*  
 * Find or create the conversation for this.  
 */  
conversation = find_or_create_conversation(pinfo);  
  
/*  
 * Is there a request structure attached to this conversation?  
 */  
session_state = conversation_get_proto_data(conversation, proto_smtp);  
if (!session_state) {  
    /*  
     * No - create one and attach it.  
     */  
    session_state = se_alloc(sizeof(struct smtp_session_state));  
    session_state->smtp_state = SMTP_STATE_READING_CMDS;  
    session_state->crlf_seen = FALSE;  
    session_state->data_seen = FALSE;  
    session_state->msg_read_len = 0;  
    session_state->msg_tot_len = 0;  
    session_state->msg_last = TRUE;  
    session_state->last_nontls_frame = 0;  
  
    conversation_add_proto_data(conversation, proto_smtp, session_state);  
}
```



# TCP Reassembly

- TCP messages & tvbuffs have different boundaries
- `tcp_dissect_pdus()` to the rescue!
- `epan/dissectors/packet-tcp.h`
- What about other reassembly?

# Using tcp\_dissect\_pdus()

```
static void
dissect_dns_tcp_pdu(tvbuff_t *tvb, packet_info *pinfo, proto_tree *tree)
{
    col_set_str(pinfo->cinfo, COL_PROTOCOL, "DNS");

    dissect_dns_common(tvb, pinfo, tree, TRUE, FALSE, FALSE);
}

static void
dissect_dns_tcp(tvbuff_t *tvb, packet_info *pinfo, proto_tree *tree)
{
    tcp_dissect_pdus(tvb, pinfo, tree, dns_desegment, 2, get_dns_pdu_len,
        dissect_dns_tcp_pdu);
}

proto_reg_handoff_dns(void)
{
    dissector_handle_t dns_udp_handle;
    dissector_handle_t dns_tcp_handle;

    dissector_add("tcp.port", TCP_PORT_DNS, dns_tcp_handle);
}
```

# General Reassembly

- Collect fragments: `fragment_add_XXX`
- Create tvb: `tvb_new_XXX`
- Create detail tab: `add_new_data_source`
- Dissect the child data: `dissect_XXX`

# IP Defragmentation

```
/* If ip_defragment is on, this is a fragment, we have all the data
 * in the fragment, and the header checksum is valid, then just add
 * the fragment to the hashtable.
 */
save_fragmented = pinfo->fragmented;
if (ip_defragment && (iph->ip_off & (IP_MF|IP_OFFSET)) &&
    tvb_bytes_exist(tvb, offset, pinfo->iplen - pinfo->iphdrln) &&
    ipsum == 0) {
    ipfd_head = fragment_add_check(tvb, offset, pinfo,
        iph->ip_p ^ iph->ip_id ^ src32 ^ dst32,
        ip_fragment_table,
        ip_reassembled_table,
        (iph->ip_off & IP_OFFSET)*8,
        pinfo->iplen - pinfo->iphdrln,
        iph->ip_off & IP_MF);

    next_tvb = process_reassembled_data(tvb, offset, pinfo, "Reassembled IPv4",
        ipfd_head, &ip_frag_items, &update_col_info, ip_tree);
} else {
```

# Exceptions

- Automatic  
offset = 234567890;  
uid = tvb\_get\_ntohs(tvb, offset);
- Manual  
THROW(ReportedBoundsError);  
DISSECTOR\_ASSERT(offset < 300);  
REPORT\_DISSECTOR\_BUG("That wasn't cheese..");

# Error Reporting

- Bad:

```
g_assert(len <= MAX_LEN);
```

- Sort-of-OK:

```
fprintf(stderr, "Oops.");  
proto_tree_add_debug_text(...);
```

- Better: Expert Info

# Expert Info

- Adds to expert windows
- Similar to syslog
- epan/expert.h, epan/expert.c

```
expert_add_info_format(pinfo, ti, PI_MALFORMED, PI_ERROR,  
                      "Corrupted data segment");  
expert_add_info_format(pinfo, ti, PI_SEQUENCE, PI_NOTE,  
                      "Less horseradish next time");
```

# Portability Tips

- We run on Windows (32 & 64), Linux, Solaris, OS X, FreeBSD, NetBSD, OpenBSD, AIX, HP-UX, ...
- GLib types
- Old compilers (Visual C++ 6.0)
  - No C++ comments
  - No C99



# Portability tips 2

- No malloc, sprintf, strcpy, open...
- sizeof and strlen returns a size\_t
- Use ep\_ and se\_ allocated memory
- `#ifdef _WIN32 /* not WIN32 */`

# Crashing Wireshark

- Dereference a NULL pointer
- Overrun a buffer
- Pass a NULL string to a printf-style function
- Global pointer to ep\_allocated memory

# Check Your Inputs

```
elem_desc_len = tvb_get_ntohs(...);  
  
while (desc_bytecnt != 0) {  
    elem_bytecnt = elem_desc_len;  
  
    if (elem_bytecnt > desc_bytecnt)  
        elem_bytecnt = desc_bytecnt;  
  
    dissect_something_or_other(...);  
  
    offset += elem_bytecnt;  
    desc_bytecnt -= elem_bytecnt;  
}
```

# Speaking of Loops...

```
guint8 pdu_len, el_len;
int offset;
pdu_len = tvb_get_guint8(tvb, offset);
offset++;
while (pdu_len > 0) {
    el_len = tvb_get_guint8(tvb, offset);
    dissect_our_pdu(...);
    offset += el_len;
    pdu_len -= el_len;
}
```

# What's the Difference?

```
some_string = tvb_get_string(tvb, 0, 20);  
col_add_fstr(pinfo->cinfo, COL_INFO, some_string);  
col_set_str(pinfo->cinfo, COL_INFO, some_string);
```

# Making Your Own Package

- Why?
- doc/README.packaging
- version.conf + make-version.pl



# Bonus Material



# Disk Requirements

- Sources (plain) 350 MB
- Sources (compiled) 850 MB
- Support libs 250 MB
- Cygwin .5 – 2.0 GB
- Python 50 MB

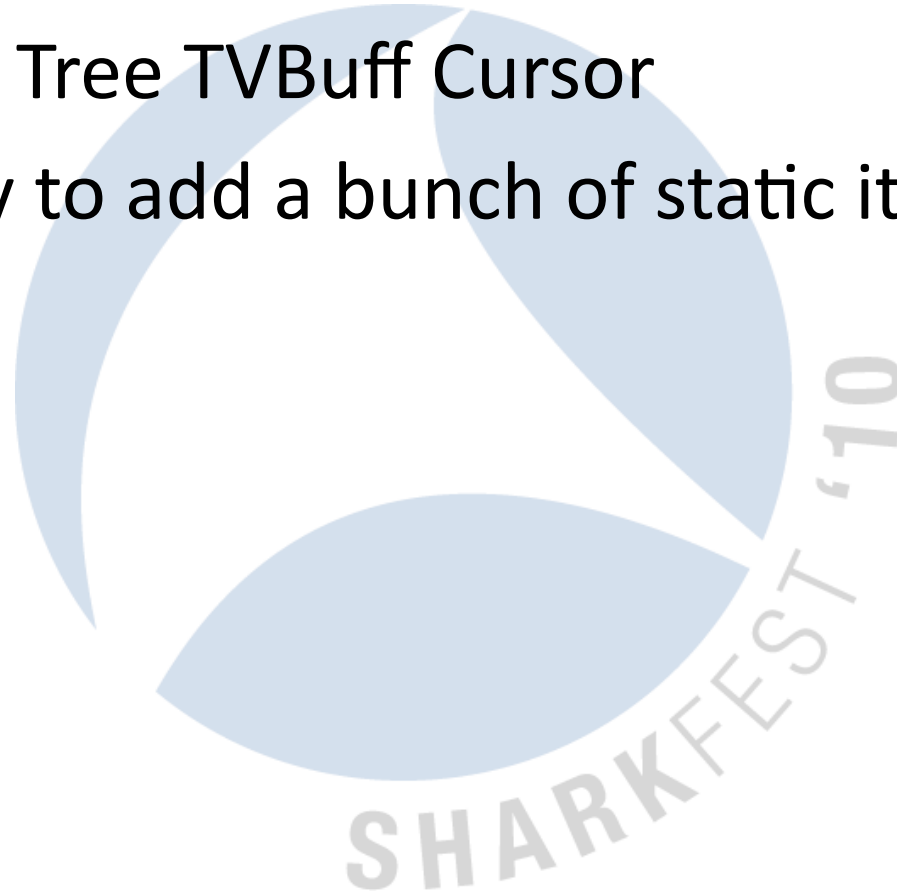


# Why won't you add my code?

- Is it well-written?
- Did you fuzz it?
- Did you send along a capture file?
- Should you ping someone?

# Ptvcursors

- Protocol Tree TVBuff Cursor
- Easy way to add a bunch of static items



# Ptvcursor Example

```
ptvcursor_t *cursor;
int offset = 0;

cursor = ptvcursor_new(tree, tvb, offset);
ptvcursor_add(cursor, hf_stream_addr, 1,
              FALSE);
/* more ptvcursor_add calls */
ptvcursor_add(cursor, hf_salmon_count, 4,
              FALSE);
offset = ptvcursor_current_offset(cursor);
ptvcursor_free(cursor);
return offset;
```

# Automatic Generation

- ASN.1
- CORBA IDL
- Samba PIDL
- Protomatics

