LFR-FSW - Bug #649

CCSDS_TC_PKT_MAX_SIZE wrong: detection of a TC with wrong length

08/03/2016 08:27 AM - Veronique bouzid

Status:	Closed	**Start date:**	08/03/2016
Priority:	High	**Due date:**	
Assignee:	Veronique bouzid	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
revision:	r0		

Description

Le requirement suivant est passé de design a test.
Command packets (DPU Software to RPW analyzer flight software)
SSS-IF-DPS-EQ-180
Command packets (DPU SW to RPW analyzer SW)
The Analyzer Flight Software shall receive commands from the DPU Software as Telecommand
Source Packets according to [ES1] with a maximum length given below for each analyzer:
- LFR: SY_LFR_TC_MAX_LEN = **228 bytes** (i.e. 216 bytes of application data)

J'ai regardé dans ton code et toi tu testes si la longueur de la TC est < CCSDS_TC_PKT_MAX_SIZE qui vaut 255.

fsw_spacewire.c
--> lecture du buffer
estimatePacketslength = len -CCSDS_TC_TM_PACKET_OFFSET -3

-CCSDS_TC_TM_PACKET_OFFSET = 7
ensuite appel de la fonction tc_parser

cette fonction est dans le fichier tc_acceptance.c

*elle va verifier la longueur de la TC avec la valeur CCSDS_TC_PKT_MAX_SIZE = 255 et cela ce n est pas bon
WRONG_LEN_PKT *

ensuite on verifie que la lg est compatible avec le subtype (tc_check_length)

ensuite il va envoyer uen send_tm_lfr_tc_exe_corrupted
il y a un champ currentTC_LEN_RCV = lg du buffer lu rangé dans un 16 bits.

Dans les logs voici ce que l on obtient:

15:56:39.56912, * **32** , TM_LFR_TC_EXE_CORRUPTED, CCSDS_VERSION_NUMBER = 0, PACKET_TYPE: TM_PACKET = 0, DATA_FIELD_HEADER_FLAG: WITH_HEADER = 1, PROCESS_ID: RPW_PID_2 = 76, PACKET_CATEGORY: ACKNOWLEDGE = 1, (PACKET_ID=0xcc1), SEGMENTATION_GROUPING_FLAG: STANDALONE_PACKET = 3, SEQUENCE_CNT=24, (PACKET_SEQUENCE_CONTROL=0xc018), PACKET_LENGTH=25, SPARE_1=0, PUS_VERSION = 1, SPARE_2=0, SERVICE_TYPE: TELECOMMAND_VERIFICATION = 1, SERVICE_SUBTYPE: TC_EXECUTION_COMPLETION_FAILURE = 8, DESTINATION_ID: MISSION_TIMELINE = 110, TIME=0x8000001f5a68, PA_RPW_TELECOMMAND_PKT_ID=0x1ccc, PA_RPW_PKT_SEQ_CONTROL=0xccf3, PA_RPW_TC_FAILURE_CODE: CORRUPTED = 42005, PA_RPW_TC_SERVICE=181, PA_RPW_TC_SUBTYPE=93, *PA_RPW_PKT_LEN_RCV_VALUE=231, PA_RPW_PKT_DATFIELDSIZE_CNT=231, PA_RPW_RCV_CRC=0x7b4, PA_RPW_COMPUTED_CRC=0x7b4

Les fichiers de test sont dans /home/validation/data/R3/3.0.0.22/1.1.89/SVS-0003

Contexte du test

FSW 3.0.0.22
VHDL 1.1.89
EM sans Timegen
SocExplorerEngine.getSocExplorer: Version = 0.6.2, Branch = default, Changeset = 819d0376d481
StarDundee

Related issues:

Related to Task #655: Mise à jour SRS et SVS / TC > 228 bytes	**Closed**	**09/03/2016**

History

#1 - 08/03/2016 09:21 AM - Veronique bouzid

Voici des précisions:
dans mon exemple, la commande TM_LFR_TC_EXE_CORRUPTED est envoyée par la fonction tc_check_length qui verifie la coherence entre le subtype de la TC et la long attendue (definie dans ICD).

DONC JE CONFIRME QUE LA DETECTION D'UNE TC AVEC UNE LG > 228 octets (216 en application) ne fonctionne pas.

Je viens d'écrire un script qui va envoyer une TC avec une longueur > 255 pour valider le code écrit par Paul et voir la réaction du soft.
il se nomme /opt/VALIDATION_R3/lfrverif/LFR_SVS/SVS-0003/send_bad_tc1.py

J envoie une TC correspondant à packet_lg = 263 et avec le type = 181 et subtype = 63.
Cette TC est correctement formattée excepté sa taille.
Attention cette valeur correspond au nombre d'octets -1 de PACKET DATA FIELD .
Cela correspond donc à une TC de longueur 264+6=270 octets
6 pour PACKET HEADER

Voici la réponse du soft
le fichier 2016_03_09-12_41_06-Detail.txt montre

12:41:01.418377, /!\TC too long with lg is 274 --> On a 4 octets en plus liés au Spacewire

12:41:01.451066, TM_LFR_TC_EXE_CORRUPTED, CCSDS_VERSION_NUMBER = 0, PACKET_TYPE: TM_PACKET = 0, DATA_FIELD_HEADER_FLAG: WITH_HEADER = 1, PROCESS_ID: RPW_PID_2 = 76, PACKET_CATEGORY: ACKNOWLEDGE = 1, (PACKET_ID=0xcc1), SEGMENTATION_GROUPING_FLAG: STANDALONE_PACKET = 3, SEQUENCE_CNT=0, (PACKET_SEQUENCE_CONTROL=0xc000), PACKET_LENGTH=25, SPARE_1=0, PUS_VERSION = 1, SPARE_2=0, SERVICE_TYPE: TELECOMMAND_VERIFICATION = 1, SERVICE_SUBTYPE: TC_EXECUTION_COMPLETION_FAILURE = 8, DESTINATION_ID: MISSION_TIMELINE = 110, TIME=0x8000000622c8, PA_RPW_TELECOMMAND_PKT_ID=0x1ccc, PA_RPW_PKT_SEQ_CONTROL=0xec3f, PA_RPW_TC_FAILURE_CODE: CORRUPTED = 42005, PA_RPW_TC_SERVICE=181, PA_RPW_TC_SUBTYPE=93, **PA_RPW_PKT_LEN_RCV_VALUE=263, PA_RPW_PKT_DATFIELDSIZE_CNT=246,** PA_RPW_RCV_CRC=0x3f, PA_RPW_COMPUTED_CRC=0x84c9

12:41:01.463547, TM_LFR_TC_EXE_CORRUPTED, CCSDS_VERSION_NUMBER = 0, PACKET_TYPE: TM_PACKET = 0, DATA_FIELD_HEADER_FLAG: WITH_HEADER = 1, PROCESS_ID: RPW_PID_2 = 76, PACKET_CATEGORY: ACKNOWLEDGE = 1, (PACKET_ID=0xcc1), SEGMENTATION_GROUPING_FLAG: STANDALONE_PACKET = 3, SEQUENCE_CNT=0, (PACKET_SEQUENCE_CONTROL=0xc000), PACKET_LENGTH=25, SPARE_1=0, PUS_VERSION = 1, SPARE_2=0, SERVICE_TYPE: TELECOMMAND_VERIFICATION = 1, SERVICE_SUBTYPE: TC_EXECUTION_COMPLETION_FAILURE = 8, /!\DESTINATION_ID: 128, TIME=0x8000000625b4, PA_RPW_TELECOMMAND_PKT_ID=0x3f80, PA_RPW_PKT_SEQ_CONTROL=0x0, PA_RPW_TC_FAILURE_CODE: CORRUPTED = 42005, PA_RPW_TC_SERVICE=0, PA_RPW_TC_SUBTYPE=63, **PA_RPW_PKT_LEN_RCV_VALUE=16256, PA_RPW_PKT_DATFIELDSIZE_CNT=7**, PA_RPW_RCV_CRC=0xf26d, PA_RPW_COMPUTED_CRC=0xb0fa

On a donc un probleme car on recoit 2 TM_LFR_EXE_CORRUPTED.

Ce comportement est donc à corriger.

#2 - 09/03/2016 01:32 PM - Veronique bouzid

- File send_bad_tc1.py added

Les fichiers de test ont été rangés dans le repertoire /home/validation/data/R3/3.0.0.22/TESTS-UNITAIRES/bad-tc/tc_sup_255

Ici le script permettant de rejouer le test. Il est dans /opt/VALIDATIOn_R3/send_bad_tc1.py.

#3 - 09/03/2016 01:42 PM - Veronique bouzid

- File send_bad_tc.py added

Le script qui permet de valider le requirement de la SSS SSS-IF-DPS-EQ-180 se trouve dans /opt/VALIADATION_R3/lfrverif/LFR-SVS/SVS-0003 et se nommesend_bad_tc.py.

*Précisions importantes
La valeur de 228 octets correspond à la longueur de la telemetry source packets.
Elle comprend donc
- packet header (6 octets)
- packet_data_fields (section variable en fonction de la TC)

Voir également le LFR TC Packet summary au début de l'ICD.*

les fichiers de test (2016_03_09-12_34_53-*) sont rangés dans /home/validation/R3/3.0.0.22/TESTS-UNITAIRES/bad-tc.

#4 - 09/03/2016 02:53 PM - Veronique bouzid

- Related to Task #655: Mise à jour SRS et SVS / TC > 228 bytes added

#5 - 24/05/2016 10:10 PM - paul leroy

Un bug intattendu. Le driver SpaceWire permet de configurer la taille maximale d'un paquet reçu par le hardware (actuellement 252 octets, c'est définit dans le fichier fsw_config.c des sources), je n'avais jamais essayé d'envoyer un paquet trop grand.

```
// GRSPW0 resources
struct drvmgr_key grlib_grspw_0n1_res[] = {
    {"txBdCnt", KEY_TYPE_INT, {(unsigned int)50}}, // 7 SWF_F0, 7 SWF_F1, 7 SWF_F2, 7 CWF_F3, 7 CWF_F1 ou 7 CW
F_F2
        {"rxBdCnt", KEY_TYPE_INT, {(unsigned int)10}},
        {"txDataSize", KEY_TYPE_INT, {(unsigned int)4096}},
        {"txHdrSize", KEY_TYPE_INT, {(unsigned int)34}},
        {"rxPktSize", KEY_TYPE_INT, {(unsigned int)248+4}},
        KEY_EMPTY
};
```

J'ai l'impression que le paquet est découpé en deux parties. Ce qui expliquerait pourquoi il y a deux paquets d'erreur, et que le deuxième comporte des valeurs farfelues pour les type, subtype etc..

Concrètement, si tu envoies une TC d'une taille immense, elle est découpée en tronçons qui valent la longueur configurée dans le driver, et non mon paramètre CCSDS_TC_PKT_MAX_SIZE.

Je propose donc de configurer le driver SpaceWire pour qu'il soit capable de gérer des TC de taille 228. Si tu envoies une TC de taille supérieure, elle est découpée en tronçons (comme dans des transmissions TCP), et tu obtiens plusieurs paquets d'erreur (autant que de tronçons). Et donc de mettre:

```
{"rxPktSize", KEY_TYPE_INT, {(unsigned int)228+4}},
```

où 4 est là pour les octets ajoutés par le protocole (le hardware enlève le premier octet il me semble, celui de l'adresse).

En conséquence, je peux aussi supprimer le test

```
if ( packetLength >= CCSDS_TC_PKT_MAX_SIZE ) {
            status = WRONG_LEN_PKT;
        }
```

qui n'a pas de sens puisque le driver ne transmettra que des tronçons de 228

#6 - 24/05/2016 10:17 PM - paul leroy

- Status changed from New to Feedback

- Assignee changed from paul leroy to Veronique bouzid

#7 - 28/06/2016 01:11 PM - Veronique bouzid

- Assignee changed from Veronique bouzid to paul leroy

Hello,
Es tu sure que tu as bien modifié la valeur du buffer du spacewire ({"rxPktSize", KEY_TYPE_INT, {(unsigned int)228+4}},)
car j ai refait des tests et le comportement est le meme
- lg de la tc < 270 on envoie une seule CORRUPTED
- lg de la tc = 271 on envoie 2 TM_LFR_EXE_CORRUPTED

--> cela signifie qu il y a un autre parametre

Avec Alexis, on a vérifié à quel niveau le packet etait découpé, ce n est pas coté banc de test car on voit bien qu'une commande trop longue est envoyée en un coup (interface spacewire de SOC), ce qui signifie que c'est bien coté carte LFR que ce découpage est fait.

Peux-tu m'expliquer la signification des champs
PA_RPW_PKT_LEN_RCV_VALUE --> longueur totale du buffer fourni par le spacewire (header + CCSDS TC)??
PA_RPW_PKT_DATFIELDSIZE_CNT --> champ PACKET_LENGTH extrait de CCSDS TC (= lg de packet_data_field -1)
de TM_LFR_EXE_CORRUPTED.

#8 - 28/06/2016 02:47 PM - paul leroy

Tu utilisais quelle version du logiciel de vol? J'ai regardé dans mes sources, normalement la modification devrait être effective à partir de la version 3.1.0.1 (changement visible à la révision 290 (135d1e42c4e1) du dépôt DVE_PLE). Tu ne vois aucune différence de comportement?

Pour la signification des champs, l'ICD dit:

PA_RPW_PKT_LEN_RCV_VALUE TC - - - Value of the field Packet Length of the corrupted TC.
Il s'agit donc de l'information reçue dans la TC.

PA_RPW_PKT_DATFIELDSIZE_CNT TC - - - Number of bytes of the Packet Data Field from corrupted TC minus 1.
Il s'agit donc de la taille du data field effectivement constatée, moins 1.

#9 - 28/06/2016 02:52 PM - paul leroy

- Assignee changed from paul leroy to Veronique bouzid

#10 - 29/06/2016 08:45 AM - Veronique bouzid

- Assignee changed from Veronique bouzid to paul leroy

la version est la dernière 3.1.0.2.

Tu dis que le spacewire coupe la TC > 228, comment peut on vérifier ce point? As tu dans ton code la possibilité de le faire.

#11 - 29/06/2016 09:18 AM - paul leroy

Je lis dans le document rtems-gaisler-drivers-1.2.11.0.pdf que j'ai dans le package rtems fourni par Gaisler:

The driver can be configured using driver resources as described in the driver manager chapter. Below is a description of configurable driver parameters. The driver parameters is unique per GRSPW device. The parameters are all optional, the parameters only overrides the default values.

Puis dans le tableau qui suit dans le document:

```
rxPktSize INT Maximum packet size of received packets.
```

Dans les sources grspw.c, on peut voir une valeur par défaut. Je ne comprends pas d'où peut venir le 270.

```
#define SPACEWIRE_RXPCK_SIZE 1024
```

#12 - 30/06/2016 07:34 AM - paul leroy

Je confirme que les paramètres modifiés que j'utilise pour configurer le driver SpaceWire ne sont pas pris en compte. Ce sont les paramètres par défaut qui s'appliquent.

Dans grspw.c:

```
/* initialize the code with some resonable values,
     * actual initialization is done later using ioctl(fd)
     * on the opened device */
    pDev->config.rxmaxlen = SPACEWIRE_RXPCK_SIZE;
    pDev->txdbufsize = SPACEWIRE_TXD_SIZE;
    pDev->txhbufsize = SPACEWIRE_TXH_SIZE;
    pDev->rxbufsize = SPACEWIRE_RXPCK_SIZE;
    pDev->txbufcnt = SPACEWIRE_TXBUFS_NR;
    pDev->rxbufcnt = SPACEWIRE_RXBUFS_NR;
```

Et aussi:

```
#define SPACEWIRE_INIT_TIMEOUT 10
#define SPACEWIRE_BDTABLE_SIZE 0x400
#define SPACEWIRE_TXD_SIZE 1024
#define SPACEWIRE_TXH_SIZE 64
#define SPACEWIRE_RXPCK_SIZE 1024
#define SPACEWIRE_TXBUFS_NR 64
#define SPACEWIRE_RXBUFS_NR 128
```

Pour ne pas changer trop la configuration, je propose de vivre avec les valeurs par défaut, qui sont utilisées actuellement, mais de corriger le traitement des télécommandes trop grandes.

#13 - 30/06/2016 07:36 AM - paul leroy

- Assignee changed from paul leroy to Veronique bouzid

#14 - 28/12/2016 07:49 AM - Veronique bouzid

- Assignee changed from Veronique bouzid to paul leroy

Dans quelle version de soft penses-tu nous livrer la correction de ce bug?.

#15 - 05/01/2017 05:28 PM - paul leroy

- File CHANGELOG.RCC added

- File grspw_1_2_13.c added

- File grspw_1_2_20.c added

Il y a du nouveau de ce côté.

En réinstallant le banc de test et de compilation sur une machine, j'ai découvert dans le changelog de la nouvelle version de RCC (1.2.20) des lignes qui parlent du paramètre rxPktSize (au niveau de la révision 1.2.18). Sachant que j'ai utilisé la version 1.2.13 pour compiler toutes les dernières versions du soft de vol jusqu'à la 3.1.0.4. On devrait donc voir une différence de comportement.

Alexis était plutôt pour qu'on utilise la dernière version de RCC, 1.2.20, pour avoir les corrections de bugs je suis assez d'accord avec ça. Ca méritera d'être tracé dans la gestion de conf.

Dans la version 3.1.0.5, je propose de repartir en considérant que le paramètre est bien pris en compte et de regarder ce qu'il se passe.

#16 - 06/01/2017 08:56 AM - paul leroy

- Assignee changed from paul leroy to Veronique bouzid

Je ne retrouve pas le test send_bad_tc.py sur le dépôt VALIDATION Rhodecode. Il n'est pas versionné?

#17 - 06/01/2017 10:04 AM - paul leroy

J'ai refait des tests avec le driver de la version 1.2.20 de RCC et en fixant les paramètres de tailles avec l'appel à

```
status = ioctl(fd, SPACEWIRE_IOCTRL_SET_PACKETSIZE, packetsize); // set rxsize, txdsize and txhsize
```

avec

```
spw_ioctl_packetsize packetsize;
```

```
    packetsize.rxsize = 228;
    packetsize.txdsize = 4096;
    packetsize.txhsize = 34;
```

Le comportement observé est le suivant. Pour une TC > 228 bytes, le driver ne remonte pas l'arrivée des données (pas de paquet d'erreur généré, la TC est complètement ignorée). Pour des TC <= 228, le paquet est traité et le message d'erreur attendu est envoyé.

Modifications effectives dans fsw >= 3.1.0.5.

#18 - 06/01/2017 11:25 AM - bruno katra

- Assignee changed from Veronique bouzid to paul leroy

paul leroy wrote:

> Je ne retrouve pas le test send_bad_tc.py sur le dépôt VALIDATION Rhodecode. Il n'est pas versionné?

Si il est versionnée mais pas dans VALIDATION (Description : VALIDATION SVS scripts : R1, R2 et R2+(EM+)), c'est dans VALIDATION_R3plus : https://hephaistos.lpp.polytechnique.fr/rhodecode/HG_REPOSITORIES/LPP/INSTRUMENTATION/SOLO_LFR/VALIDATION_R3plus

SINON :

Véro avait inclu le script dans la présente issue (voir attached files dans la description)

#19 - 09/01/2017 07:54 AM - paul leroy

- Assignee changed from paul leroy to Veronique bouzid

C'est bon, je l'ai trouvé, merci.

#20 - 17/01/2017 04:30 PM - Veronique bouzid

- Status changed from Feedback to Closed

Bug corrigé en 3.1.0.6.
Par contre il faut IMPERATIVEMENT DOCUMENTER LA SRS ET LA SVS-0003.

le script utilisé est le /opt/VALIDATION_R3plus/lfrverif/LFR_SVS/SVS-0003/send_bad_tc1.py.

Voila la trace mettant en évidence le comportement du soft.

1- La TC sera rejeté par le driver spacewire et donc pas d acquittement, cela se traduit par
13:21:00.500713, /!\No TM_LFR_TC_EXE_* after time-out (4s).

2- La detection d'une TC trop longue est décrite dans la structure spw_stats et ensuite reportée dans les HK. Les champs suivants seront mis à jour
HK_LFR_DPU_SPW_RX_TOO_BIG sera incrementé
HK_LFR_ME_CNT sera incrementé car HK_LFR_DPU_SPW_RX_TOO_BIG est une medium criticity

3- L'erreur de type medium est reportée dans les champs dédiés
HK_LFR_LAST_ER_RID: ME_LFR_DPU_SPW = 42338, HK_LFR_LAST_ER_CODE: RX_TOO_BIG = 9,
HK_LFR_LAST_ER_TIME=0x80000012218c

4- Ces champs ne seront pas mis à jour
HK_LFR_LAST_REJ_TC_ID, HK_LFR_LAST_REJ_TC_TYPE, HK_LFR_LAST_REJ_TC_SUBTYPE, HK_LFR_LAST_REJ_TC_TIME

Les fichiers (2017_01_17-13_21_05*) sont rangés dans /home/validation/R3+/3.1.0.6/3.1.91/TESTS-UNITAIRES/send_bad_tc1.

Files

send_bad_tc1.py	3.56 KB	09/03/2016	Veronique bouzid
send_bad_tc.py	3.56 KB	09/03/2016	Veronique bouzid
CHANGELOG.RCC	15.8 KB	05/01/2017	paul leroy
grspw_1_2_13.c	62.5 KB	05/01/2017	paul leroy
grspw_1_2_20.c	63.3 KB	05/01/2017	paul leroy