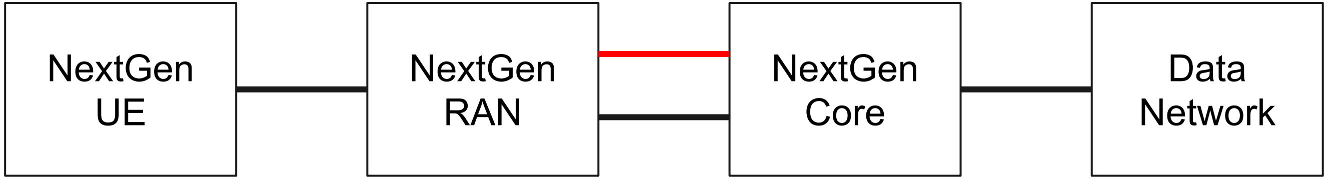
Free5GC 軟體測試

0756029 邱肇珩
0756086 郎宇傑

Outline

- 5G架構
    - 軟體架構
    - Attach
- Test

5G 架構


NextGen UE

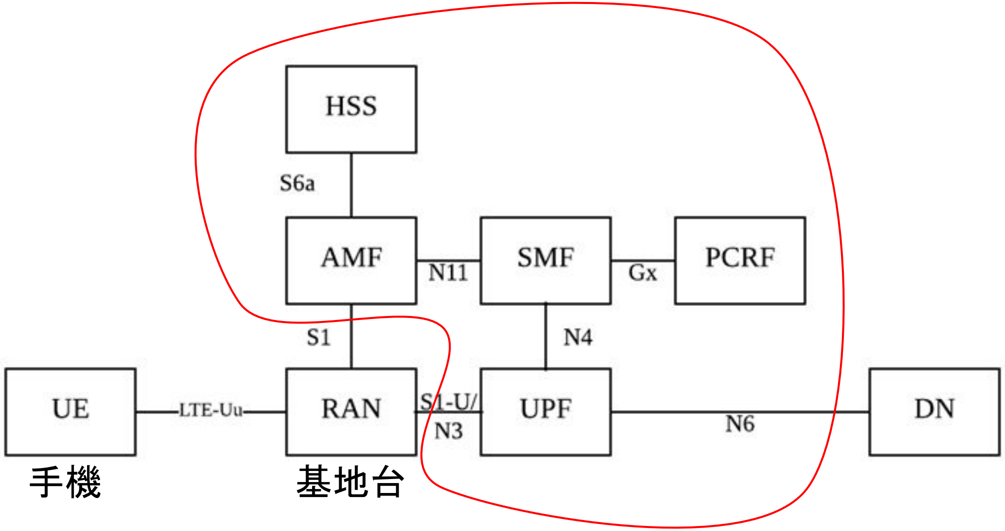
NextGen RAN

NextGen Core

Data Network

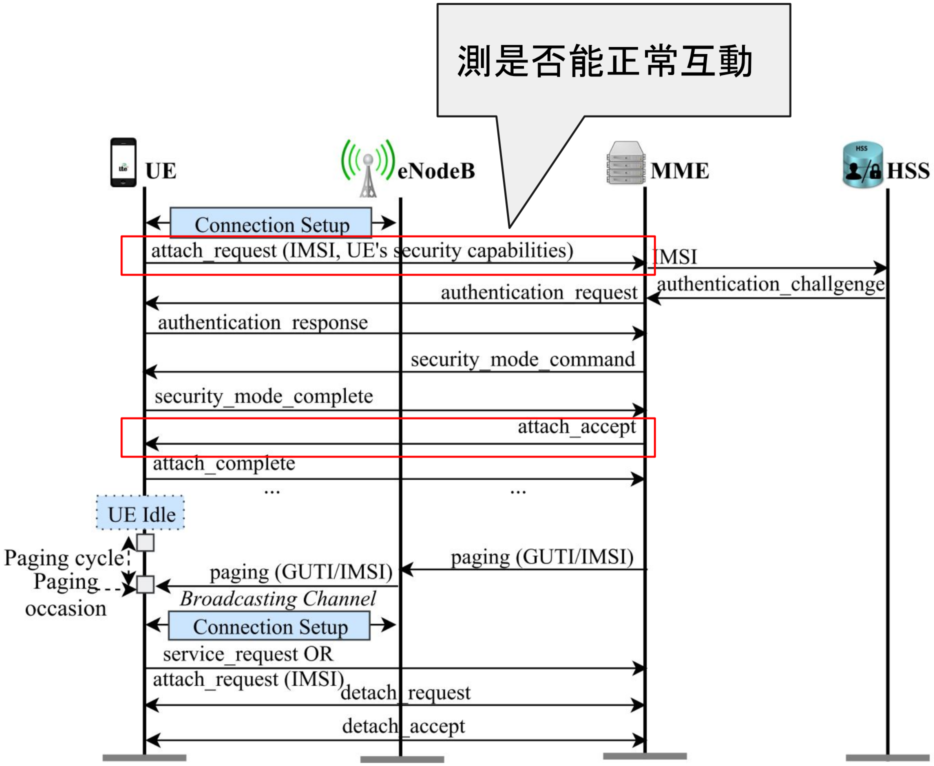
軟體架構

- Home Subscriber Server（HSS)
- Policy and Charging Rules Function（PCRF)
- Access and Mobility Management Function（AMF)
- Session Management Function（SMF)
- User Plane Function（UPF)



不是正常的 5G 架構，整體還是偏向 4G

Attach



測是否能正常互動

UE

eNodeB

MME

HSS

Connection Setup

attach_request (IMSI, UE's security capabilities)

IMSI

authentication_challgenge

authentication_request

authentication_response

security_mode_command

security_mode_complete

attach_accept

attach_complete

...

...

UE Idle

Paging cycle

Paging

occasion

paging (GUTI/IMSI)

paging (GUTI/IMSI)

Broadcasting Channel

Connection Setup

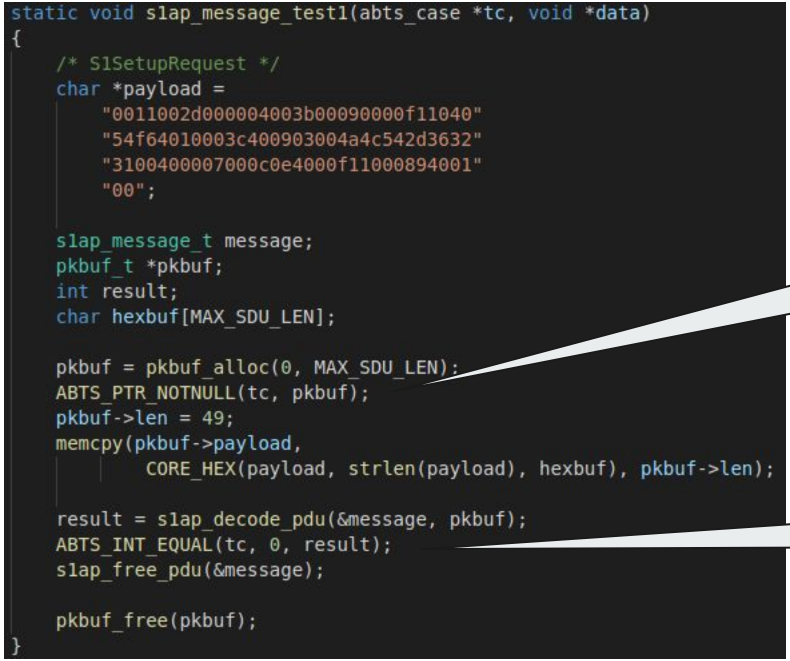
service_request OR

attach_request (IMSI)

detach_request

detach_accept

S1 setup request 測試

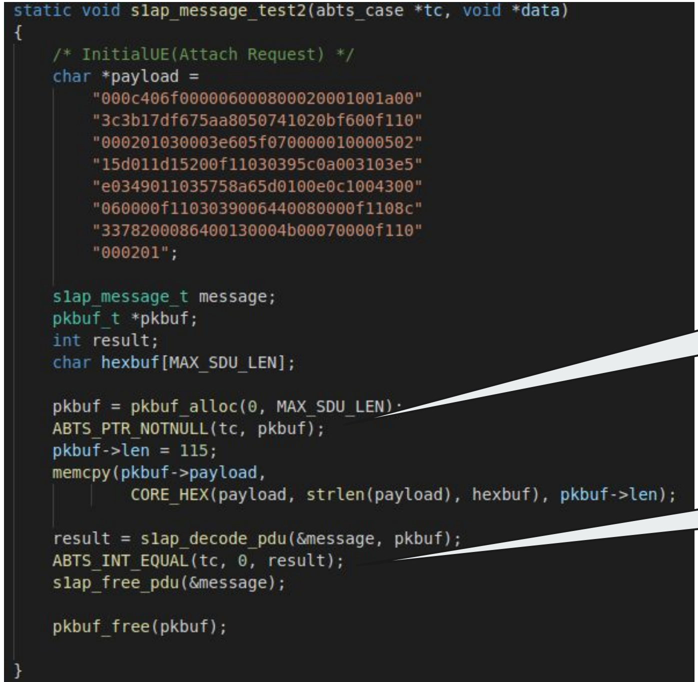


```c
static void s1ap_message_test1(abts_case *tc, void *data)
{
    /* S1SetupRequest */
    char *payload =
        "0011002d000004003b00090000f11040"
        "54f64010003c400903004a4c542d3632"
        "3100400007000c0e4000f11000894001"
        "00";

    s1ap_message_t message;
    pkbuf_t *pkbuf;
    int result;
    char hexbuf[MAX_SDU_LEN];

    pkbuf = pkbuf_alloc(0, MAX_SDU_LEN);
    ABTS_PTR_NOTNULL(tc, pkbuf);
    pkbuf->len = 49;
    memcpy(pkbuf->payload,
            CORE_HEX(payload, strlen(payload), hexbuf), pkbuf->len);

    result = s1ap_decode_pdu(&message, pkbuf);
    ABTS_INT_EQUAL(tc, 0, result);
    s1ap_free_pdu(&message);

    pkbuf_free(pkbuf);
}
```

測 pkbuf 是否為空

測是否能正常 decode

Attach Requset 測試

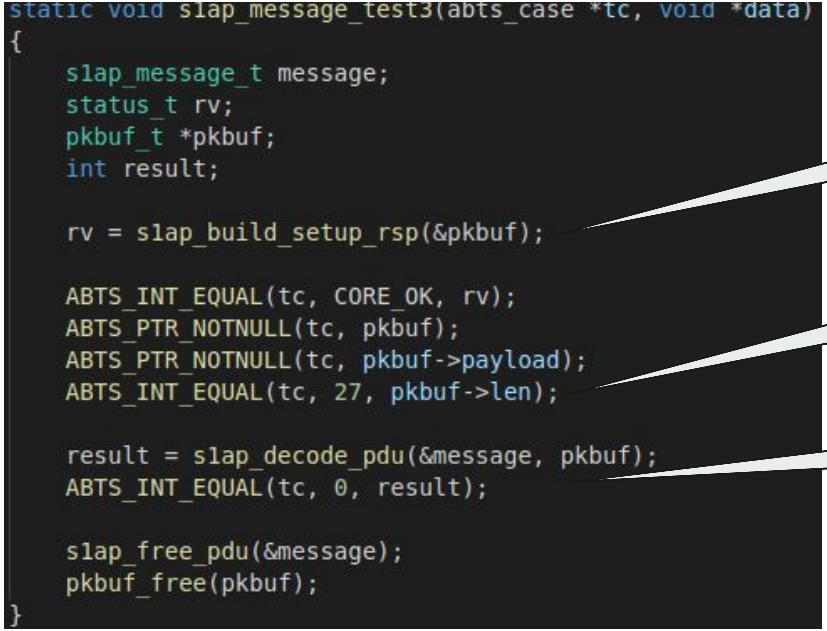


```c
static void slap_message_test2(abts_case *tc, void *data)
{
    /* InitialUE(Attach Request) */
    char *payload =
        "000c406f00000600080002000l001a00"
        "3c3b17df675aa8050741020bf600f110"
        "000201030003e605f070000010000502"
        "15d011d15200f11030395c0a003103e5"
        "e0349011035758a65d0100e0c1004300"
        "060000f1103039006440080000f1108c"
        "33782000086400130004b00070000f110"
        "000201";

    slap_message_t message;
    pkbuf_t *pkbuf;
    int result;
    char hexbuf[MAX_SDU_LEN];

    pkbuf = pkbuf_alloc(0, MAX_SDU_LEN);
    ABTS_PTR_NOTNULL(tc, pkbuf);
    pkbuf->len = 115;
    memcpy(pkbuf->payload,
            CORE_HEX(payload, strlen(payload), hexbuf), pkbuf->len);

    result = s1ap_decode_pdu(&message, pkbuf);
    ABTS_INT_EQUAL(tc, 0, result);
    s1ap_free_pdu(&message);

    pkbuf_free(pkbuf);

}
```

測 pkbuf 是否為空

測是否能正常 decode

S1AP setup response 測試



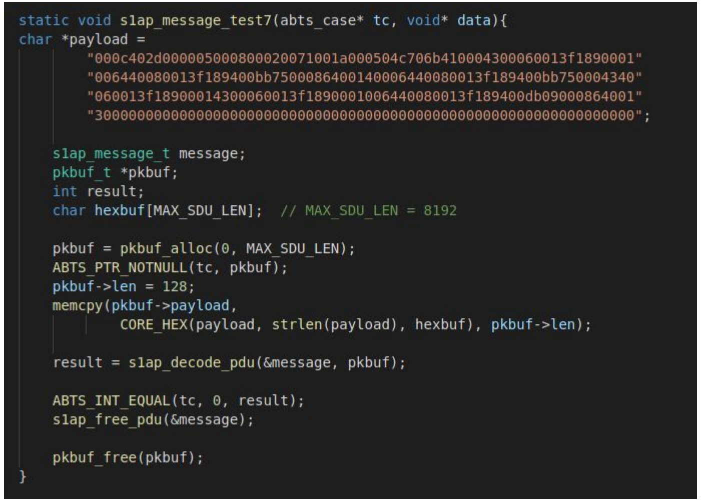
```
static void s1ap_message_test3(abts_case *tc, void *data)
{
    s1ap_message_t message;
    status_t rv;
    pkbuf_t *pkbuf;
    int result;

    rv = s1ap_build_setup_rsp(&pkbuf);

    ABTS_INT_EQUAL(tc, CORE_OK, rv);
    ABTS_PTR_NOTNULL(tc, pkbuf);
    ABTS_PTR_NOTNULL(tc, pkbuf->payload);
    ABTS_INT_EQUAL(tc, 27, pkbuf->len);

    result = s1ap_decode_pdu(&message, pkbuf);
    ABTS_INT_EQUAL(tc, 0, result);

    s1ap_free_pdu(&message);
    pkbuf_free(pkbuf);
}
```

呼叫原有的 function

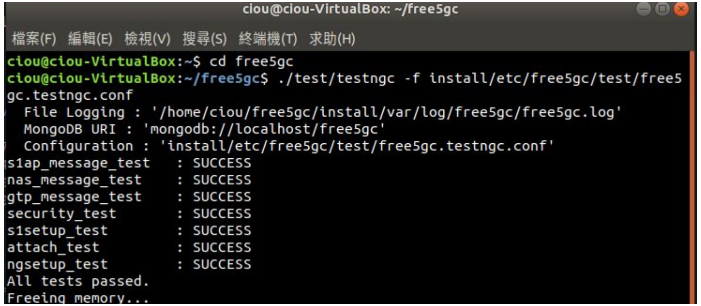
檢查是否回傳正常
並確認 pkbuf 長度正常

確認可以正常解碼

Service Request 測試



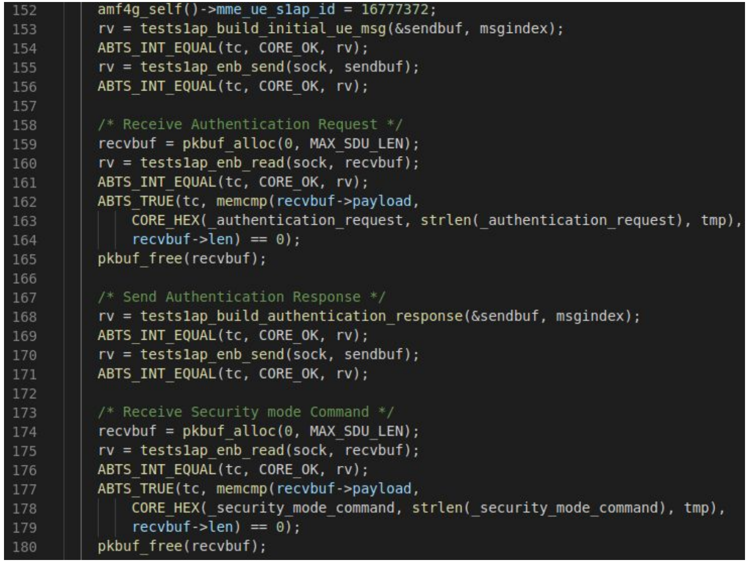
```
static void s1ap_message_test7(abts_case* tc, void* data){
char *payload =
            "000c402d000005000800020071001a000504c706b410004300060013f1890001"
            "00644008 0013f189400bb75000864001400064400800 13f189400bb750004340"
            "060013f18900014300060013f18900010064400800 13f189400db09000864001"
            "3000000000000000000000000000000000000000000000000000000000000000";

    s1ap_message_t message;
    pkbuf_t *pkbuf;
    int result;
    char hexbuf[MAX_SDU_LEN];  // MAX_SDU_LEN = 8192

    pkbuf = pkbuf_alloc(0, MAX_SDU_LEN);
    ABTS_PTR_NOTNULL(tc, pkbuf);
    pkbuf->len = 128;
    memcpy(pkbuf->payload,
            CORE_HEX(payload, strlen(payload), hexbuf), pkbuf->len);

    result = s1ap_decode_pdu(&message, pkbuf);

    ABTS_INT_EQUAL(tc, 0, result);
    s1ap_free_pdu(&message);

    pkbuf_free(pkbuf);
}
```



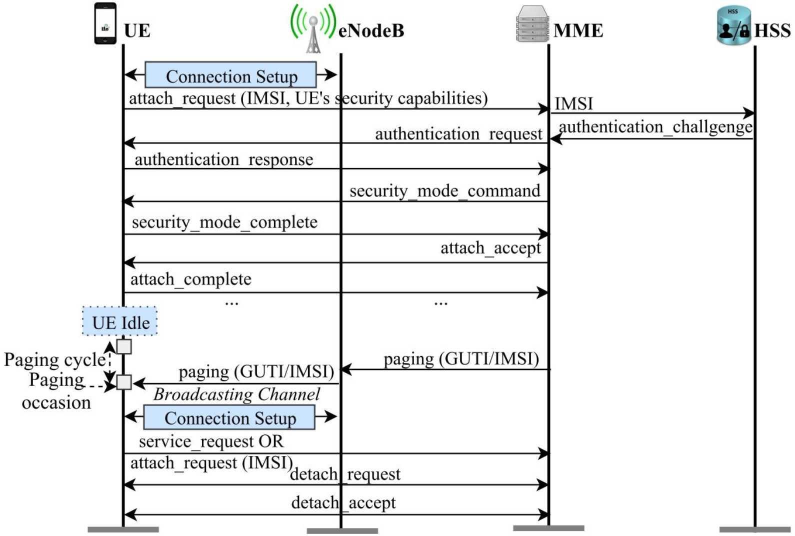
ciou@ciou-VirtualBox: ~/free5gc
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
ciou@ciou-VirtualBox:~$ cd free5gc
ciou@ciou-VirtualBox:~/free5gc$ ./test/testngc -f install/etc/free5gc/test/free5
gc.testngc.conf
  File Logging : '/home/ciou/free5gc/install/var/log/free5gc/free5gc.log'
  MongoDB URI : 'mongodb://localhost/free5gc'
  Configuration : 'install/etc/free5gc/test/free5gc.testngc.conf'
s1ap_message_test   : SUCCESS
nas_message_test    : SUCCESS
gtp_message_test    : SUCCESS
security_test       : SUCCESS
s1setup_test        : SUCCESS
attach_test         : SUCCESS
ngsetup_test        : SUCCESS
All tests passed.
Freeing memory...
```

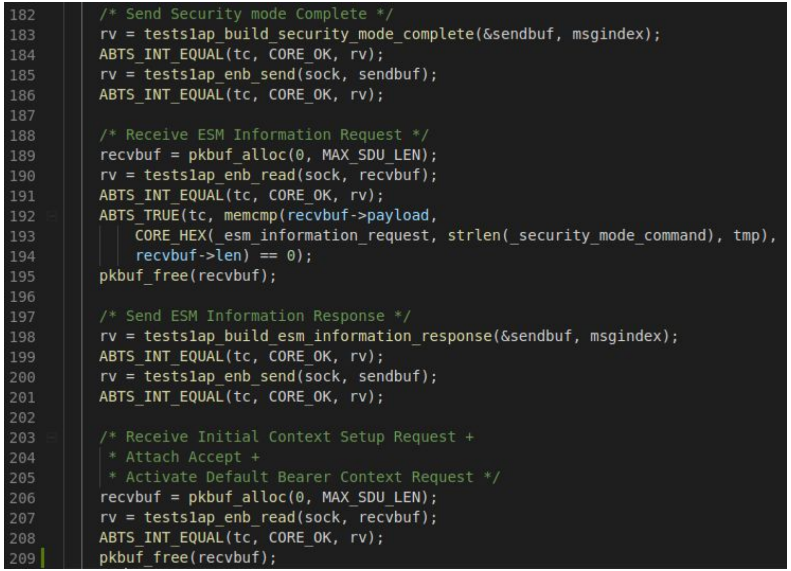
Test Attach (1 / 2)



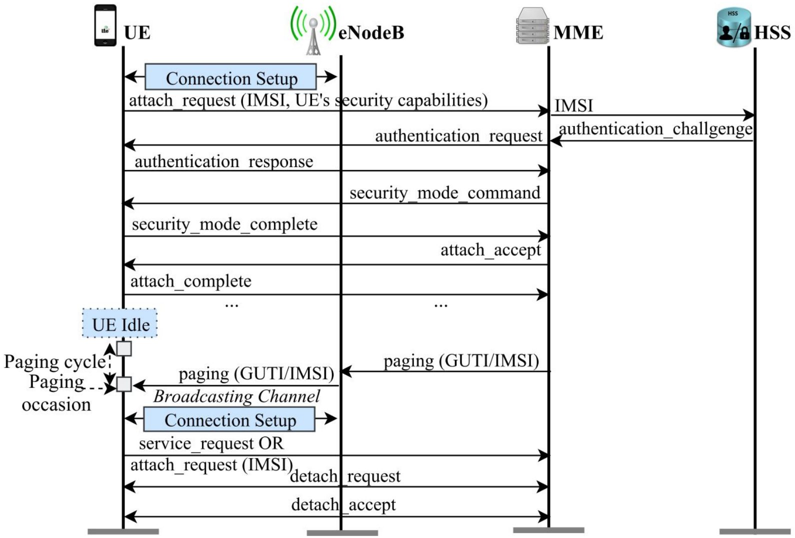
```c
    amf4g_self()->mme_ue_s1ap_id = 16777372;
    rv = tests1ap_build_initial_ue_msg(&sendbuf, msgindex);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);
    rv = tests1ap_enb_send(sock, sendbuf);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);

    /* Receive Authentication Request */
    recvbuf = pkbuf_alloc(0, MAX_SDU_LEN);
    rv = tests1ap_enb_read(sock, recvbuf);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);
    ABTS_TRUE(tc, memcmp(recvbuf->payload,
        CORE_HEX(_authentication_request, strlen(_authentication_request), tmp),
        recvbuf->len) == 0);
    pkbuf_free(recvbuf);

    /* Send Authentication Response */
    rv = tests1ap_build_authentication_response(&sendbuf, msgindex);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);
    rv = tests1ap_enb_send(sock, sendbuf);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);

    /* Receive Security mode Command */
    recvbuf = pkbuf_alloc(0, MAX_SDU_LEN);
    rv = tests1ap_enb_read(sock, recvbuf);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);
    ABTS_TRUE(tc, memcmp(recvbuf->payload,
        CORE_HEX(_security_mode_command, strlen(_security_mode_command), tmp),
        recvbuf->len) == 0);
    pkbuf_free(recvbuf);
```



Test Attach (2 / 2)



```
    /* Send Security mode Complete */
    rv = tests1ap_build_security_mode_complete(&sendbuf, msgindex);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);
    rv = tests1ap_enb_send(sock, sendbuf);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);

    /* Receive ESM Information Request */
    recvbuf = pkbuf_alloc(0, MAX_SDU_LEN);
    rv = tests1ap_enb_read(sock, recvbuf);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);
    ABTS_TRUE(tc, memcmp(recvbuf->payload,
        CORE_HEX(_esm_information_request, strlen(_security_mode_command), tmp),
        recvbuf->len) == 0);
    pkbuf_free(recvbuf);

    /* Send ESM Information Response */
    rv = tests1ap_build_esm_information_response(&sendbuf, msgindex);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);
    rv = tests1ap_enb_send(sock, sendbuf);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);

    /* Receive Initial Context Setup Request +
     * Attach Accept +
     * Activate Default Bearer Context Request */
    recvbuf = pkbuf_alloc(0, MAX_SDU_LEN);
    rv = tests1ap_enb_read(sock, recvbuf);
    ABTS_INT_EQUAL(tc, CORE_OK, rv);
    pkbuf_free(recvbuf);
```

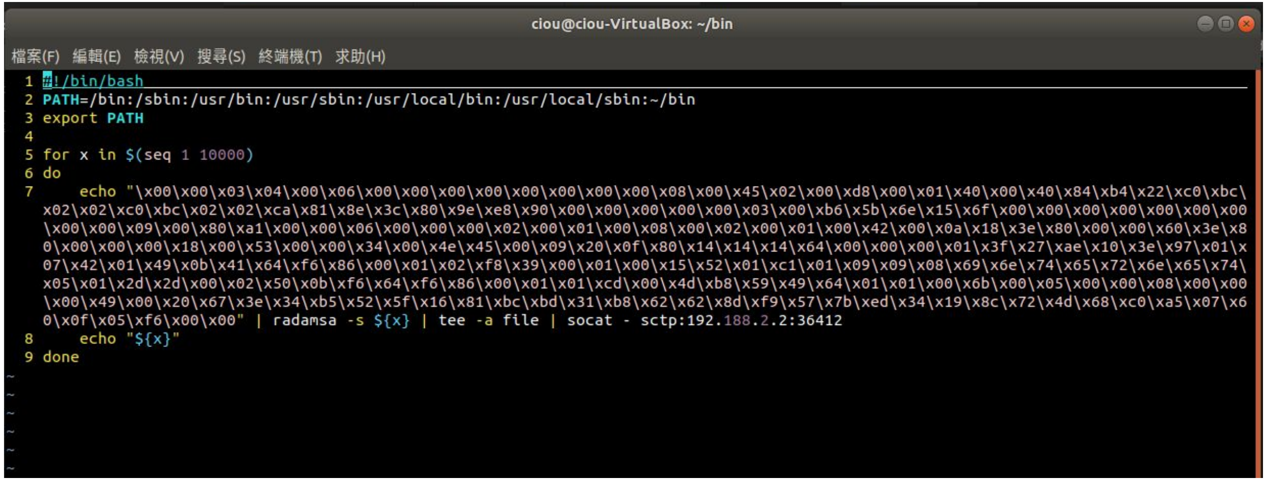


Fuzz Testing

產生一系列非法、非預期、隨機的輸入給目標程序

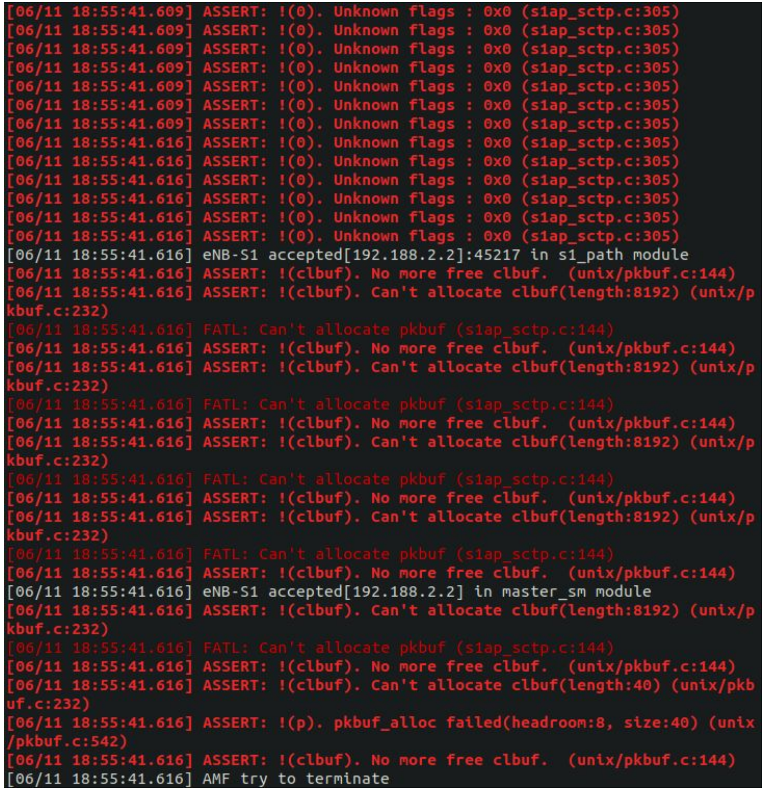
Fuzzer : radamsa

Input : Attach request


ciou@ciou-VirtualBox: ~/bin
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
#!/bin/bash
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:~/bin
export PATH

for x in $(seq 1 10000)
do
    echo "\x00\x00\x03\x04\x00\x06\x00\x00\x00\x00\x00\x00\x00\x00\x08\x00\x45\x02\x00\xd8\x00\x01\x40\x00\x40\x84\xb4\x22\xc0\xbc\
x02\x02\xc0\xbc\x02\x02\xca\x81\x8e\x3c\x80\x9e\xe8\x90\x00\x00\x00\x00\x00\x03\x00\xb6\x5b\x6e\x15\x6f\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x09\x00\x80\xa1\x00\x00\x06\x00\x00\x00\x02\x00\x01\x00\x08\x00\x02\x00\x01\x00\x42\x00\x0a\x18\x3e\x80\x00\x00\x60\x3e\x8
0\x00\x00\x00\x18\x00\x53\x00\x00\x34\x00\x4e\x45\x00\x09\x20\x0f\x80\x14\x14\x14\x64\x00\x00\x00\x01\x3f\x27\xae\x10\x3e\x97\x01\x
07\x42\x01\x49\x0b\x41\x64\xf6\x86\x00\x01\x02\xf8\x39\x00\x01\x00\x15\x52\x01\xc1\x01\x09\x09\x08\x69\x6e\x74\x65\x72\x6e\x65\x74\
x05\x01\x2d\x2d\x00\x02\x50\x0b\xf6\x64\xf6\x86\x00\x01\x01\xcd\x00\x4d\xb8\x59\x49\x64\x01\x01\x00\x6b\x00\x05\x00\x00\x08\x00\x00
\x00\x49\x00\x20\x67\x3e\x34\xb5\x52\x5f\x16\x81\xbc\xbd\x31\xb8\x62\x62\x8d\xf9\x57\x7b\xed\x34\x19\x8c\x72\x4d\x68\xc0\xa5\x07\x6
0\x0f\x05\xf6\x00\x00" | radamsa -s ${x} | tee -a file | socat - sctp:192.188.2.2:36412
    echo "${x}"
done
~
~
~
~
~
~
```

Result



```
[06/11 18:55:41.609] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.609] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.609] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.609] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.609] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.609] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.609] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.616] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.616] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.616] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.616] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.616] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.616] ASSERT: !(0). Unknown flags : 0x0 (s1ap_sctp.c:305)
[06/11 18:55:41.616] eNB-S1 accepted[192.188.2.2]:45217 in s1_path module
[06/11 18:55:41.616] ASSERT: !(clbuf). No more free clbuf.  (unix/pkbuf.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). Can't allocate clbuf(length:8192) (unix/p
kbuf.c:232)
[06/11 18:55:41.616] FATL: Can't allocate pkbuf (s1ap_sctp.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). No more free clbuf.  (unix/pkbuf.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). Can't allocate clbuf(length:8192) (unix/p
kbuf.c:232)
[06/11 18:55:41.616] FATL: Can't allocate pkbuf (s1ap_sctp.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). No more free clbuf.  (unix/pkbuf.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). Can't allocate clbuf(length:8192) (unix/p
kbuf.c:232)
[06/11 18:55:41.616] FATL: Can't allocate pkbuf (s1ap_sctp.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). No more free clbuf.  (unix/pkbuf.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). Can't allocate clbuf(length:8192) (unix/p
kbuf.c:232)
[06/11 18:55:41.616] FATL: Can't allocate pkbuf (s1ap_sctp.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). No more free clbuf.  (unix/pkbuf.c:144)
[06/11 18:55:41.616] eNB-S1 accepted[192.188.2.2] in master_sm module
[06/11 18:55:41.616] ASSERT: !(clbuf). Can't allocate clbuf(length:8192) (unix/p
kbuf.c:232)
[06/11 18:55:41.616] FATL: Can't allocate pkbuf (s1ap_sctp.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). No more free clbuf.  (unix/pkbuf.c:144)
[06/11 18:55:41.616] ASSERT: !(clbuf). Can't allocate clbuf(length:40) (unix/pkb
uf.c:232)
[06/11 18:55:41.616] ASSERT: !(p). pkbuf_alloc failed(headroom:8, size:40) (unix
/pkbuf.c:542)
[06/11 18:55:41.616] ASSERT: !(clbuf). No more free clbuf.  (unix/pkbuf.c:144)
[06/11 18:55:41.616] AMF try to terminate
```

THANK YOU