Femtocells: a Poisonous Needle in the Operator's Hay Stack

Ravishankar Borgaonkar, Nico Golde, Kévin Redon

Technische Universität Berlin, Security in Telecommunications
femtocell@sec.t-labs.tu-berlin.de

Black Hat 2011, Las Vegas, 3rd August 2011

Agenda

- mobile telecommunication
- end-user attacks
- network attacks

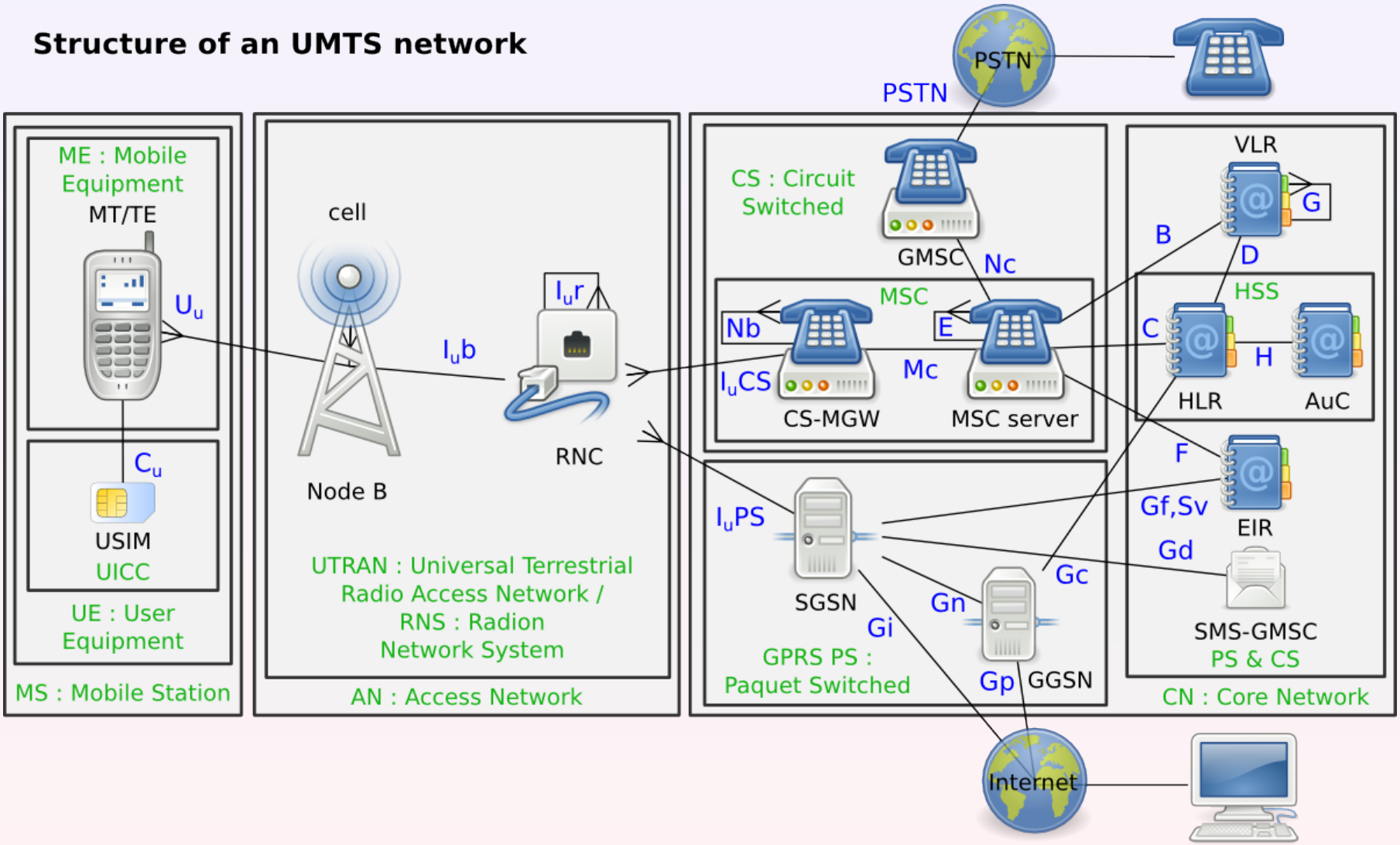
Agenda

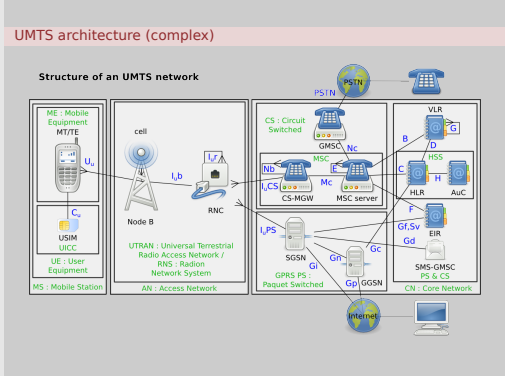
Agenda

- mobile telecommunication
- end-user attacks
- network attacks

1. UMTS architecture, femtocell definition, femtocell architecture
2. taking control over the device, reconfigure as IMSI-catcher, MitM 1: call interception, MitM 2: alter communication, MitM 3: nject traffic
3. collect information about the others, reconfigure other femtocells, taking control over other femtocells, playing with the perator network

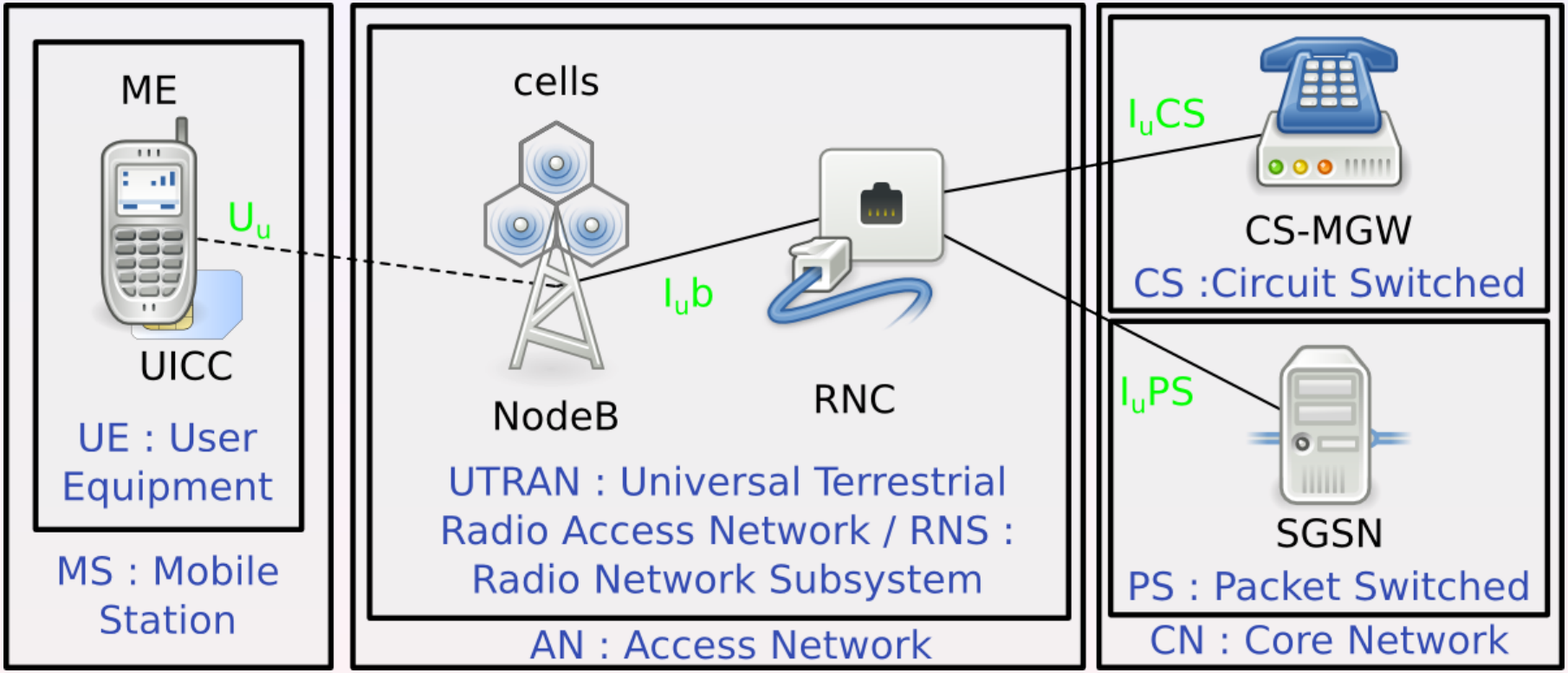
UMTS architecture (complex)

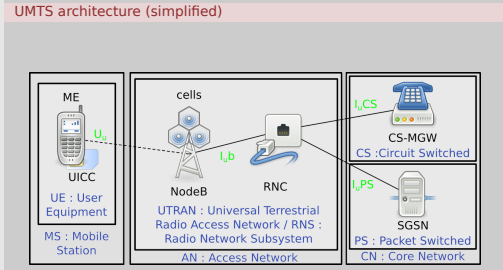




1. UMTS is the 3G technology used in Europe (mainly), equivalent to CDMA2000 in USA
2. UMTS and CDMA2000 both 3G, UMTS made by 3GPP, CDMA2000 by 3GPP2
3. UMTS architecture is quite complex, with a lot of one lettered elements and interfaces
4. diagramm should scare the audience
5. UML link multiplexing used in diagramm
6. the hay are all these elements (on letter), forming the haystack (operator network)

UMTS architecture (simplified)





1. the three main components to keep in mind are:
2. MS (mobile station) $\Leftrightarrow$ end-user equipment: the mobile phone
3. AN (access network) $\Leftrightarrow$ link between MS and CN
4. CN (core network) $\Leftrightarrow$ back-end for communication routing. critical infrastructure
5. CN is further divided into CS (Circuit Switched) for voice and PS (Packet Switched) for data traffic

technology - femtocell context?!

What is a femtocell?

- a small access point
- connects the mobile phone to the 3G/UMTS network
- compatible with every UMTS enabled mobile phone
- small cell, with a coverage of less than 50m
- low power device
- easy to install: you only have to provide power and Internet access
- technical name in 3G: Home Node B (HNB)

What is a femtocell?

- a small access point
- connects the mobile phone to the 3G/UMTS network
- compatible with every UMTS enabled mobile phone
- small cell, with a coverage of less than 50m
- low power device
- easy to install: you only have to provide power and Internet access
- technical name in 3G: Home Node B (HNB)

definition and use of femtocell technology

1. coverage area depends on exact model, operator and residential/business/...
2. sometimes called FAP (Femtocell Access Point)

customer advantages

advantages provided to users:

- can be installed at home to improve 3G coverage
- high bandwidth, and high voice quality
- location based services

advantages provided to users:
- can be installed at home to improve 3G coverage
- high bandwidth, and high voice quality
- location based services

1. femtocell is an personal base station, not shared with the rest of the public
2. location service example: kids arriving at home

operator advantages

advantages for mobile operators:

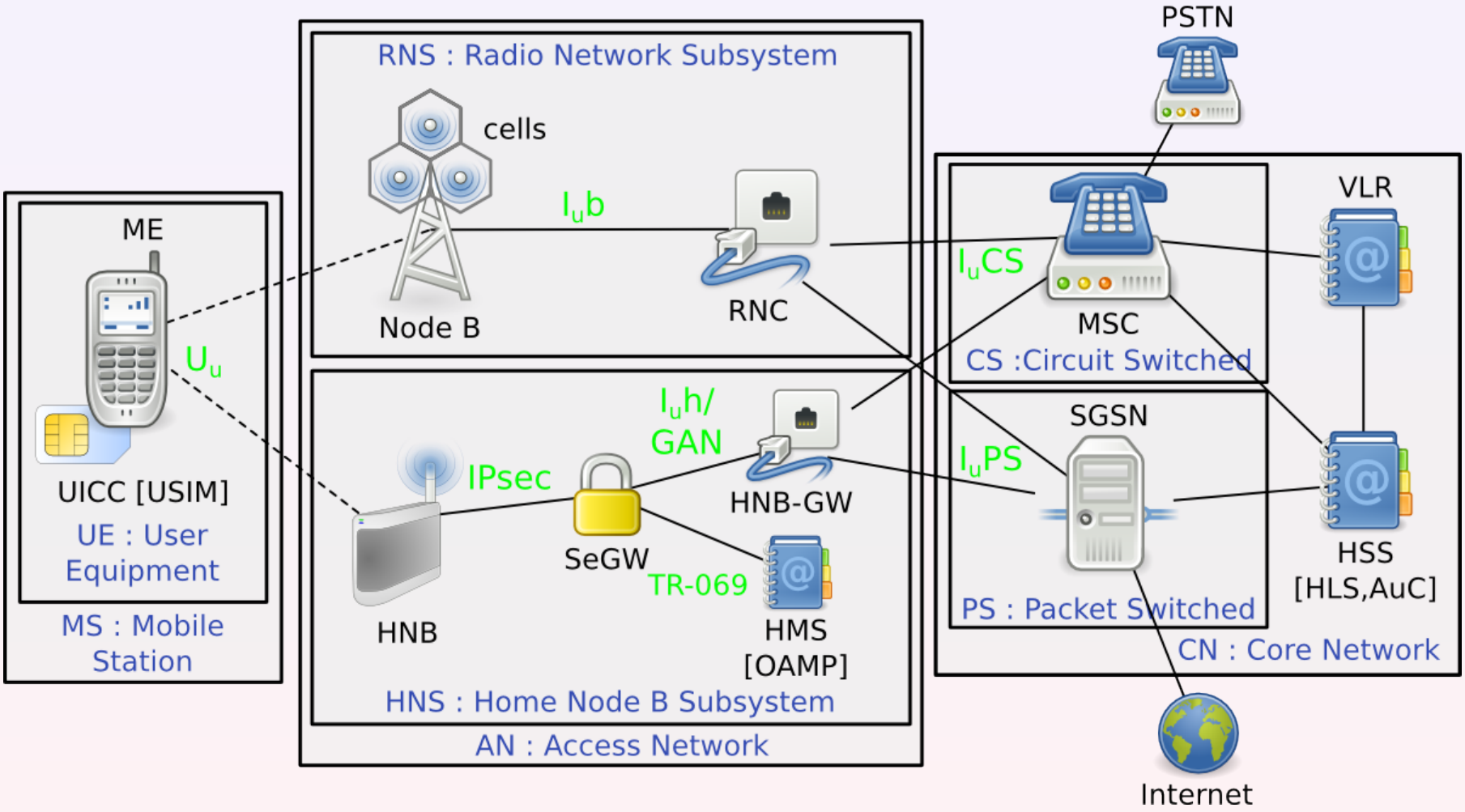
- traffic offload from public operator infrastructure ⇒ reduce expenditure
- cheap hardware compared to expensive 3G equipment
- no installation and maintenance cost
- IP connectivity

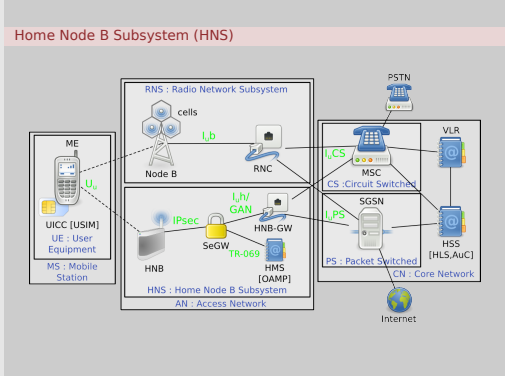
advantages for mobile operators:

- traffic offload from public operator infrastructure ⇒ reduce expenditure
- cheap hardware compared to expensive 3G equipment
- no installation and maintenance cost
- IP connectivity

1. the user has to buy the equipment and provide power/network
2. location-based services and high dedicated bandwidth offer new revenue possibilities
3. TCP/IP is well known, easy and cheap. The equipment tends to use this protocol
4. femtocells are a great opportunity for the operators
5. but now a part of their infrastructure is in the user's hand

Home Node B Subsystem (HNS)





1. HNS is equivalent to the RNS in the AN
2. HNB ⇔ Node-B (entry node for phone connection), HNB-GW ⇔ RNC (connected to the CN)
3. unlike the Node-B, a HNB is physically accessible by the user
4. the SeGW is required to provide privacy due to the use of the internet connection. it provides access control and encryption for communication
5. the HNB is the needle (in the haystack), we will make it poisonous

small cells


macrocells
cell
10^0
mini
10^{-3}
micro
10^{-6}
nano
10^{-9}
pico
10^{-12}
femto
10^{-15}
atto
10^{-18}

1. femtocells are very small cells (as the scale shows)
2. a cell is defined by the antenna and the area covered by its signal
3. attocells have be presented at the World Mobile Congress (`http://ubiquisys.com/femtocell-blog/what-is-an-attocell-new-personal-femtocell-technology/`)

femtocell threats (as defined by 3GPP)

HNB threats listed by the 3GPP

group	#	threat	impact
Compromise of H(e)NB Credentials	1	Compromise of H(e)NB authentication token by a brute force attack via a weak authentication algorithm	harmful
	2	Compromise of H(e)NB authentication token by local physical intrusion	harmful
	4	User cloning the H(e)NB authentication Token. User cloning the H(e)NB authentication Token	very harmful
Physical attacks on a H(e)NB	3	Inserting valid authentication token into a manipulated H(e)NB	harmful
	6	Booting H(e)NB with fraudulent software (“re-flashing”)	up to disastrous
	8	Physical tampering with H(e)NB	harmful
	26	Environmental/side channel attacks against H(e)NB	harmful
Attacks on Radio resources and management	21	Radio resource management tampering	harmful
Protocol attacks on a H(e)NB	5	Man-in-the-middle attacks on H(e)NB first network access	very harmful
	15	Denial of service attacks against H(e)NB	annoying
	17	Compromise of an H(e)NB by exploiting weaknesses of active network services	extremely harmful
	25	Manipulation of external time source	harmful
	27	Attack on OAM and its traffic	very harmful
	28	Threat of H(e)NB network access	harmful

group	#	threat	impact
Attacks on the core network, including H(e)NB location-based attacks	11	Changing of the H(e)NB location without reporting	harmful
	12	Software simulation of H(e)NB	very harmful
	13	Traffic tunnelling between H(e)NBs	very harmful
	14	Misconfiguration of the firewall in the modem/router	annoying
	16	Denial of service attacks against core network	annoying
	24	H(e)NB announcing incorrect location to the network	harmful
User Data and identity privacy attacks	9	Eavesdropping of the other user’s UTRAN or E-UTRAN user data	very harmful
	10	Masquerade as other users	very harmful
	18	User’s network ID revealed to Home (e)NodeB owner	breaking users privacy
	22	Masquerade as a valid H(e)NB	very harmful
	23	Provide radio access service over a CSG	very harmful
Configuration attacks on a H(e)NB	7	Fraudulent software update / configuration changes	extremely harmful
	19	Mis-configuration of H(e)NB	irritating to harmful
	20	Mis-configuration of access control list (ACL) or compromise of the access control list	irritating to harmful

HNB threats listed by the 3GPP

group	#	threat	impact
Compromise of HNB Credentials	1	Compromise of HNB authentication token by a brute force attack via a weak authentication algorithm	harmful
	2	Compromise of HNB authentication token by local eavesdropping	harmful
	3	User cloning the HNB authentication Token, User storing the HNB authentication Token	very harmful
Physical attacks on a HNB	4	Inserting valid authentication token into a manipulated HNB	harmful
	5	Hosting HNB with fraudulent software ("re-flashing")	critical
	6	Physical tampering with HNB	harmful
	26	Environmental/side channel attacks against HNB	harmful
Attacks on Radio resources and management	11	Radio resource management tampering	harmful
Protocol attacks on a HNB	7	Man-in-the-middle attacks on HNB first network access	very harmful
	13	Denial of service attacks against HNB	annoying
	25	Compromise of an HNB by exploiting weaknesses of active network services	critical
	27	Manipulation of external time source	harmful
	17	Attack on GAN and its traffic	very harmful
	28	Threat of HNB network access	harmful

group	#	threat	impact
Attacks on the core network, including HNB/HB location based attacks	11	Changing of the HNB location without reporting	harmful
	12	Software simulation of HNB	very harmful
	13	Traffic tunnelling between HNBs	very harmful
	14	Misconfiguration of the firewall in the modem/router	annoying
	15	Denial of service attacks against core network	annoying
	24	HNB announcing incorrect location to the network	harmful
User Data and identity privacy attacks	9	Eavesdropping of the other user's UTRAN or E-UTRAN user data	very harmful
	16	Masquerade as other users	very harmful
	18	User's network ID revealed to Home (e)NodeB owner	breaking user privacy
	22	Masquerade as a valid HNB	very harmful
	23	Provide radio access service over a CSG	very harmful
Configuration attacks on a HNB	3	Fraudulent software update / configuration changes	critical, disruptive
	10	Mis-configuration of HNB	disabling or limiting its utility
	20	Mis-configuration of access control list (ACL) or compromise of the access control list	disabling or limiting its utility

1. the 3GPP and operators are aware of the threats generated by femtocells
2. the threats are briefly described, with their effects and mechanism to prevent them
3. it is not a howto for attacks though (too vague), but a general overview of dangers

SFR femtocell

- sold by SFR (2nd biggest operator in France)
- cost: 99€ + mobile phone subscription
- hardware: ARM9 + FPGA for signal processing
- OS: embedded Linux kernel + proprietary services
- built by external vendors (in our case Ubiquisys), configured by operator

✆ mobile telecommunication

rogue femtocell

SFR femtocell

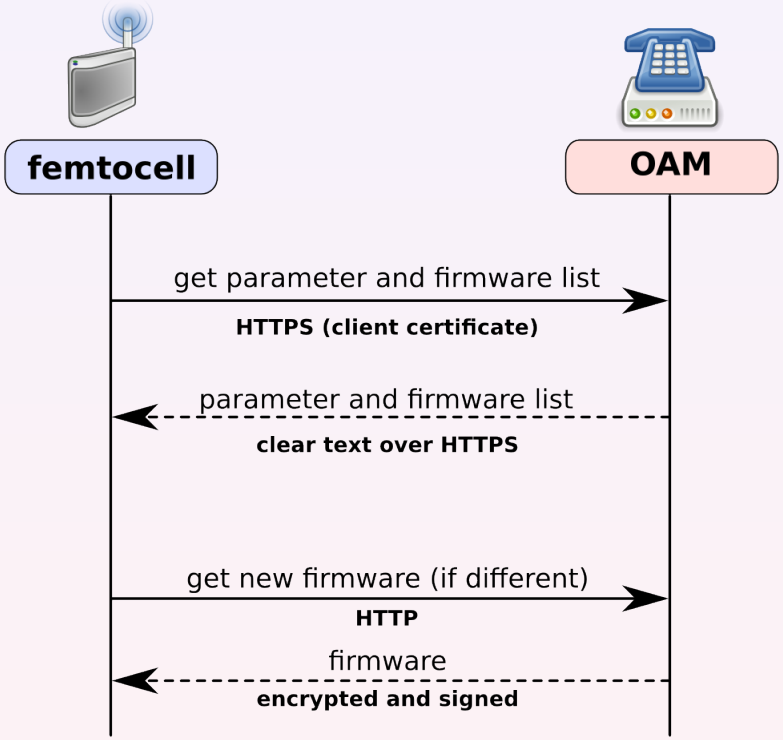
SFR femtocell

- sold by SFR (2nd biggest operator in France)
- cost: 99€ + mobile phone subscription
- hardware: ARM9 + FPGA for signal processing
- OS: embedded Linux kernel + proprietary services
- built by external vendors (in our case Ubiquisys), configured by operator

1. a brief description of our femtocell
2. all attacks have been performed with this model from SFR
3. however, the attack concepts apply to all femtocells, only the implementation varies
4. as hardware + software comes from the vendor and configuration is done by the operator, all fuckups are shared among them ;)

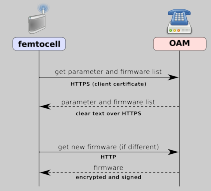
recovery procedure

- femtocells provide a recovery procedure
- similar to a factory reset
- new firmware is flashed, and settings are cleared
- used to "repair" the device without any manual intervention



recovery procedure

- femtocells provide a recovery procedure
- similar to a factory reset
- new firmware is flashed, and settings are cleared
- used to "repair" the device without any manual intervention



1. remember: keep it cheap
2. operators do not want to send a technical team to repair the femtocell
3. users are responsible for the femtocell
4. the diagram shows a simplified procedure. the complete procedure has already been presented in other confs.

recovery to fail

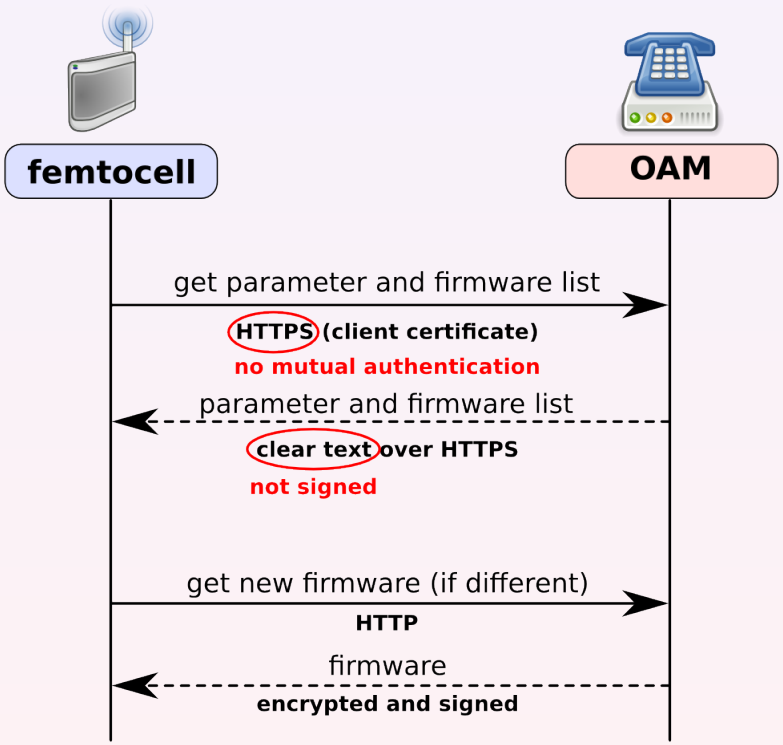
- firmware server is not authenticated

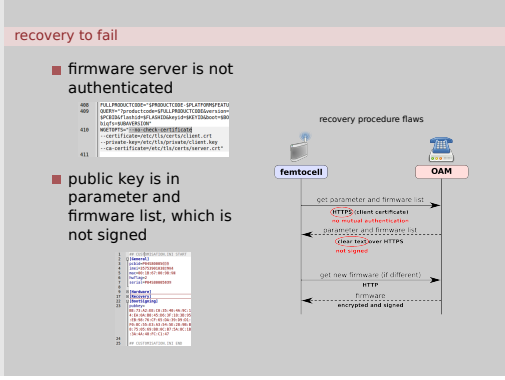
```
408 FULLPRODUCTCODE="$PRODUCTCODE-$PLATFORM$FEATU
409 QUERY="?productcode=$FULLPRODUCTCODE&version=
    $PCBID&flashid=$FLASHID&keyid=$KEYID&boot=$BO
    biqfs=$UBAVERSION"
410 WGETOPTS="--no-check-certificate
    --certificate=/etc/tls/certs/client.crt
    --private-key=/etc/tls/private/client.key
    --ca-certificate=/etc/tls/certs/server.crt"
411
```

- public key is in parameter and firmware list, which is not signed

```
1  ## CUSTOMISATION.INI START
2  [General]
3  pcbid=P04S80005[illegible]
4  imei=35753901038[illegible]
5  mac=00:1B:67:00:98:[illegible]
6  hwflag=2
7  serial=P04S80005[illegible]
8
9  [Hardware]
17 [Recovery]
22 [BootSigning]
23 pubkey=
   BE:73:A2:EE:C0:35:40:4A:9C:1
   4:EA:0A:B8:45:D6:3F:18:3B:95
   :EB:98:76:CF:65:DA:39:D9:D1:
   F0:8C:55:E3:A3:54:5E:28:9B:B
   8:75:05:69:B8:0C:B7:5A:8C:1B
   :3A:4A:48:FC:C1:47
24
25 ## CUSTOMISATION.INI END
```

recovery procedure flaws





1. the recovery procedure has a security flaw: it does not authenticate the image server
2. attacker can push his own configuration and firmwares
3. the images are signed, but the public key can be provided in the configuration file (which is not signed)
4. devices can be cloned (except for the SIM)
5. we were able to analyze the procedure because an unencrypted recovery image could be retrieved. this has been fixed, but we now have the tools to decrypt them
6. however, there are still other ways to get unencrypted images ;)

debug traces

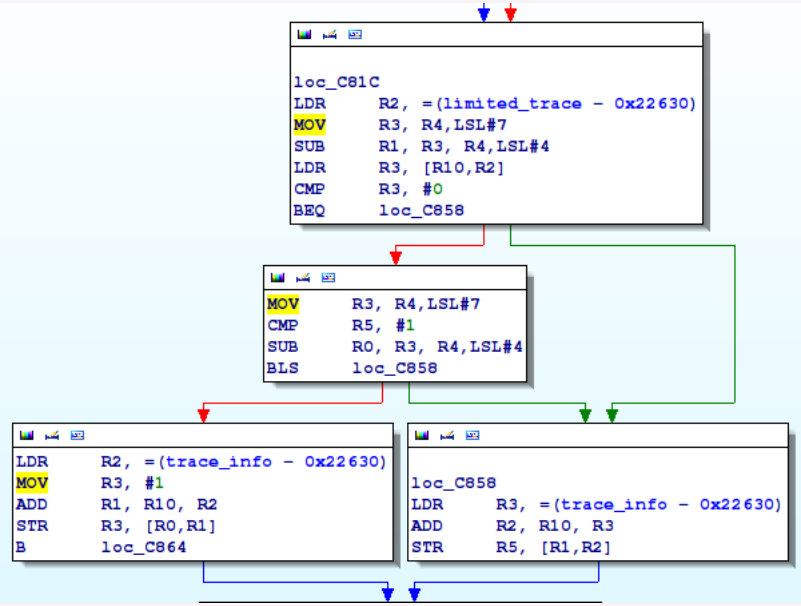
- based on local_trace_config.txt
- heavy use of dbg_trace (libosal.so)
- LD_PRELOAD db_trace to export traces
- still not very verbose (see next slide)

local trace conf

```
#TRACE_SIP
33 1
#TRACE_PKTC
34 1
#TRACE_GAC
35 4
#TRACE_GANIF
36 3
```

disabling limited trace

- all trace levels set to 1
- limited trace option compiled in libosal.so (needs patching)



final traces

```
Reply: 60 00 11 00 00 00 00 54 6F 40 04 00 11 00 44 01 02 01 1C 90 00 D5
Command: 60 00 05 00 A0 B2 01 04 1C 6E
Reply: 60 00 1E 00 4D 6F 6E 20 4E 6F 20 53 52 52 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 9
0 00 D0
usim_send_sc_id_response: IMSI: [redacted]
usim_send_sc_id_response: MSISD[redacted]
CPU Load Total=0%
SC_Main: message received, id = 0x10, if = 0x10e
Mem (KB): 62808 tot, 40432 used, 22376 free, 8244 buff, 17856 cache
CPU Load Total=0%
gac_tu3903_callback called
gac_thread_proc, Message received in state 1, Interface ID 24095, Message ID 34.
gac_thread_proc: Message received in state 1, Interface ID 24095, Message ID 34.
gac_idle called
Message 34 on IF_ID_SELF interface, state GAC_IDLE
TU3903 message received.
gac_request_ipsec called
Performing DNS lookup of fqdn unc1-ch1.fr.sfr.com
gac_dns_lookup called
Calling gethostbyname with fqdn unc1-ch1.fr.sfr.com.
gac_tu3903_callback exiting
Retrieved IPV4 address [redacted]
Calling gac_ipsec_build_send_tunnel_req with IP address       0 for fqdn unc1-ch1.fr.sfr.com
gac_ipsec_build_send_tunnel_req called
Sending tunnel_req with fqdn unc1-ch1.fr.sfr.com
Sending tunnel_req with IP [redacted]
gac_ipsec_build_send_tunnel_req: starting timer with value 120
gac_ipsec_build_send_tunnel_req exiting
gac_request_ipsec exiting
Changing state to GAC_CONNECTING_IPSEC_INITIAL
gac_idle exiting returning 2
SC_Main: message received, id = 0x215, if = 0x5e1f
SC <- SC: SC_HLR_PERIODIC_METRIC_TIMER_EXPIRED
ScHlr::handlePeriodicMetricTimerExpire:
u32 ScHlr::getUEsCampedOn() const: 0 UEs camped-on
SC -> APM: APM_PERFORMANCE_IND[52]= 0, avg= 1
ScMsgqManager::getPrMqHandle:   state error interface 0x115
build_and_send_apm_performance_ind: getPrMqHandle() returned null pointer
SC_Main: message received, id = 0x210, if = 0x5e1f
:
```

any attacks hmm?

WHAT NOW?

requirements

- classical approach in GSM: IMSI-Catcher
  - fake operator BTS (MCC/MNC)
  - acts as MitM between operator and victim
  - phone usually can't detect
  - usually used to track and intercept communication
- UMTS standard requires mutual authentication
  $\Rightarrow$ GSM approach not working [1]
- no devices acting as UMTS base station + code is available

[1] some attacks by using protocol downgrades are known

- classical approach in GSM: IMSI-Catcher
  - fake operator BTS (MCC/MNC)
  - acts as MitM between operator and victim
  - phone usually can't detect
  - usually used to track and intercept communication
- UMTS standard requires mutual authentication
  $\Rightarrow$ GSM approach not working [1]
- no devices acting as UMTS base station + code is available

[1] some attacks by using protocol downgrades are known

1. MCC: Mobile Country Code, MNC: Mobile Network Code. It's like the SSID in WLAN
2. no openBTS or openBSC project for UMTS exists
3. USRP is capable of doing it, but no implementation exists

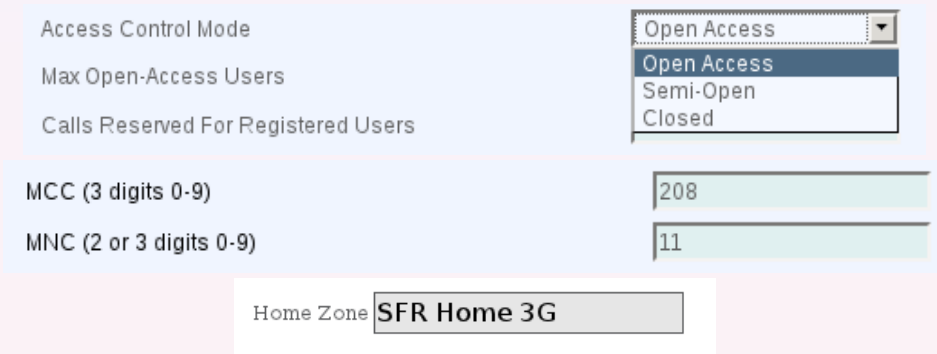
mutual authentication in the femtocell ecosystem

- in case of femtocell: mutual authentication also provided
  $\Rightarrow$ but it's useless ☺
- mutual authentication is done with the **home operator**
- NOT with the actual cell
  $\Rightarrow$ the femtocell forwards the authentication tokens
  $\Rightarrow$ mutual authentication is performed even with a rogue device

getting the fish into the octopus' tentacles

Howto build a 3G IMSI-Catcher:

- cell configuration is kindly provided as a feature of femtocells
- local cell settings stored in a proprietary database format
- some comfort provided ⇒ web interface



- we can catch any phone user of **any** operator into using our box
- roaming subscribers are allowed by SFR

⇒ the femtocell is turned into a full 3G IMSI-Catcher

getting the fish into the octopus' tentacles

Howto build a 3G IMSI-Catcher:

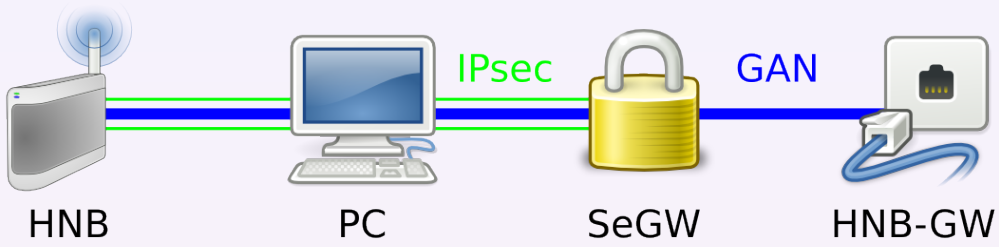
- cell configuration is kindly provided as a feature of femtocells
- local cell settings stored in a proprietary database format
- some comfort provided ⇒ web interface

- we can catch any phone user of **any** operator into using our box
- roaming subscribers are allowed by SFR

⇒ the femtocell is turned into a full 3G IMSI-Catcher

1. there is an operator web interface (main) and a vendor web interface (hidden)
2. they are password protected, but easily accessible (just get a valid cookie to override the auth)
3. roaming might be allowed because the HNB-GW is only forwarding the traffic, without filtering
4. users are handled the same way as in a real operator network
5. collecting IMSI (even without call interception) is already a privacy threat
6. roaming notification can be dropped on the way (shown later)

intercepting traffic



- proprietary IPsec client + kernel module (xpressVPN)
- multiple ways to decrypt IPsec traffic: NETLINK, ip xfrm state (not available on SFR box)
- we decided to hijack/parse ISAKMP messages passed via sendto(2) glibc wrapper
- voice data encapsulated in unencrypted RTP stream (AMR codec, stream format)

intercepting traffic

- proprietary IPsec client + kernel module (xpressVPN)
- multiple ways to decrypt IPsec traffic: NETLINK, ip xfrm state (not available on SFR box)
- we decided to hijack/parse ISAKMP messages passed via sendto(2) glibc wrapper
- voice data encapsulated in unencrypted RTP stream (AMR codec, stream format)

1. there are several ways to get decrypted traffic. the easiest is probably netlink
2. we don't need the decrypted traffic on the box, so we just extract the keys before they are passed to the PF_KEY2 kernel interface and decrypt traffic on our gateway
3. details of GAN will be presented later

extracting voice

- LD_PRELOAD ipsec user-space program to hijack sendto() and extract keys
- pass key material to host running tcpdump
- decrypt ESP packets
- extract RTP stream (rtpbreak)
- opencore-based (nb) utility to extract AMR and dump to WAV

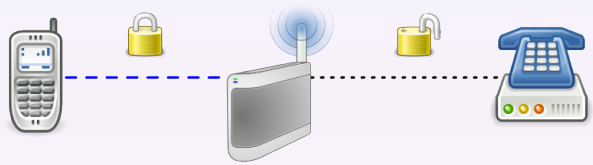
demo time

DEMONSTRATION

interception

but what about over-the-air encryption?

- only the phone ⇔ femtocell OTA traffic is encrypted
  ⇒ encryption/decryption happens on the box



- femtocell acts as a combination of RNC and Node-B: receives cipher key and integrity key from the operator for OTA encryption

Protocol	Info
UMA	GA-CSR UPLINK DIRECT TRANSFER(DTAP) (MM) Authentication Resp
UMA	Unknown URR (144)

- reversing tells us: message is **SECURITY MODE COMMAND** (unspecified RANAP derivate), which includes the keys

⚔ end-user attacks

intercepting communication

but what about over-the-air encryption?

but what about over-the-air encryption?

- only the phone ⇔ femtocell OTA traffic is encrypted ⇒ encryption/decryption happens on the box

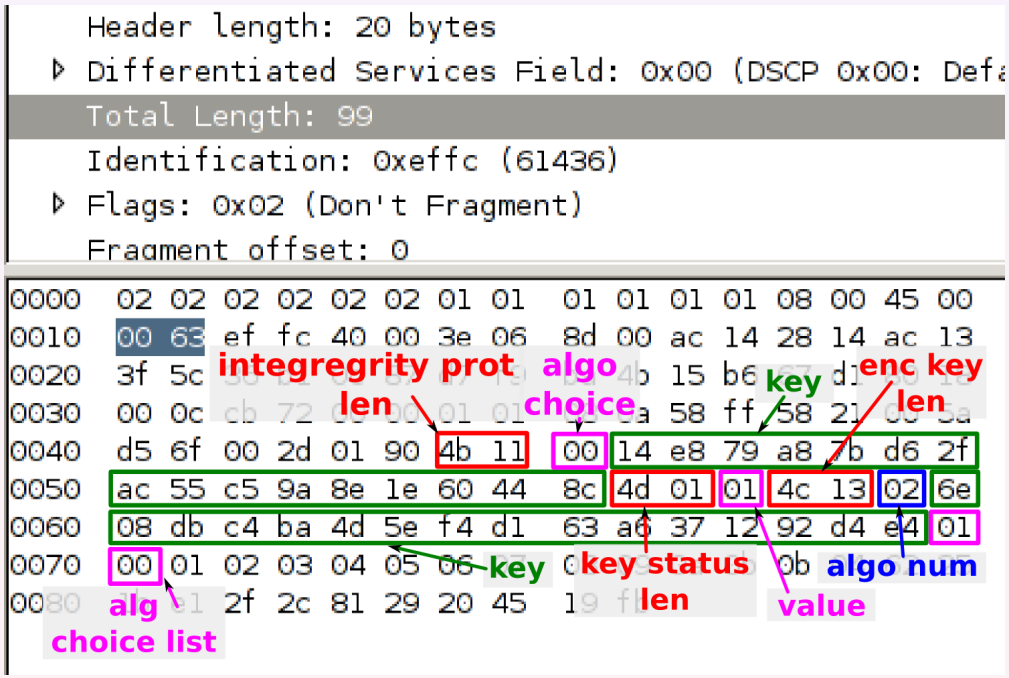
- femtocell acts as a combination of RNC and Node-B: receives cipher key and integrity key from the operator for OTA encryption

- reversing tells us: message is **SECURITY MODE COMMAND** (unspecified RANAP derivate), which includes the keys

1. OTA encryption only is by design: there is no end-to-end encryption in telco network. every elements performs a task. the communication is unencrypted in the AN/CN
2. the operator network is usually closed and trusted
3. we didn't find any standard, but seems to be a 3xTLV: (algorithm, integrity key), (key status), (algorithm list, encryption key)

SECURITY MODE COMMAND

- derived from RANAP, but spec unknown



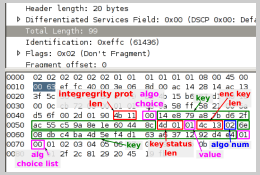
⚔ end-user attacks

intercepting communication

SECURITY MODE COMMAND

SECURITY MODE COMMAND

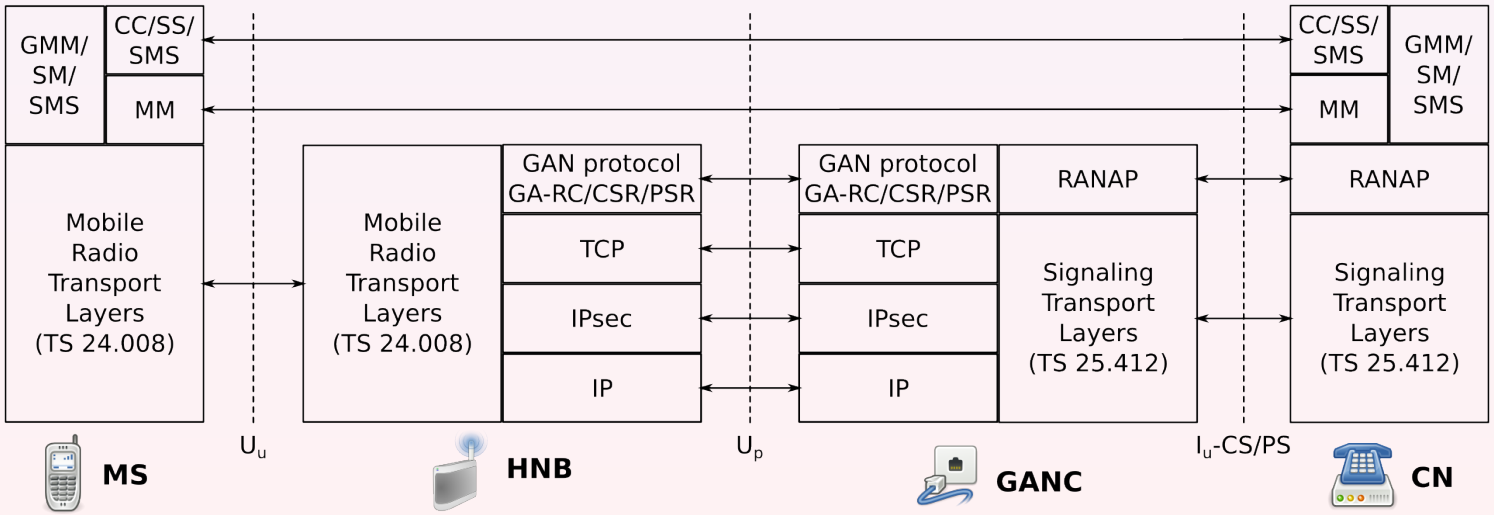
- derived from RANAP, but spec unknown



1. thanks Dieter Spaar for help on this one!

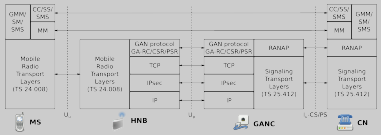
femtocell operator communication: the GAN protocol

- device is communicating with operator via GAN protocol (UMA)
  - TCP/IP mapped radio signaling
  - encapsulates radio Layer3 messages (MM/CC) in GAN protocol
  - one TCP connection per subscriber
  - radio signaling maps to GAN messages are sent over this connection
- GAN usage is transparent for the phone



femtocell operator communication: the GAN protocol

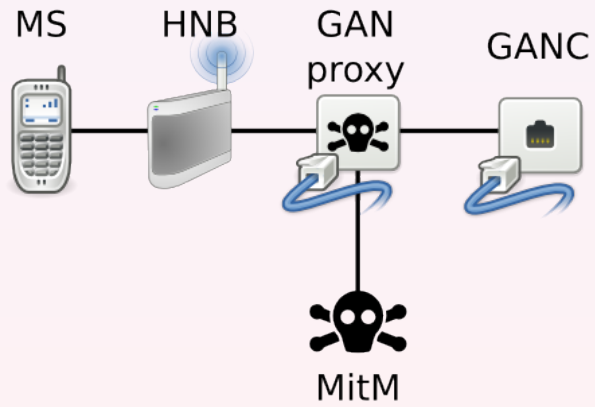
- device is communicating with operator via GAN protocol (UMA)
  - TCP/IP mapped radio signaling
  - encapsulates radio Layer3 messages (MM/CC) in GAN protocol
  - one TCP connection per subscriber
  - radio signaling maps to GAN messages are sent over this connection
- GAN usage is transparent for the phone



1. GAN is not the only I_uh solution. I_ub and IMS are the other alternatives
2. GAN is the standardized term for UMA
3. GAN is defined in 3GPP TS43.318 and TS44.318
4. GAN was designed to be used between MS and GANC over WLAN

GAN proxy/client

- proxies all GAN connections/messages
- reconfigure femtocell to connect to our proxy instead of real GANC
- proxy differs between GAN message types
- attack client controls GAN proxy over extended GAN protocol

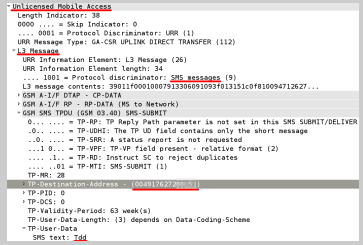


more mitm pls? sms...

- SMS message filtered by GAN proxy
- modified by client
- transfered to real GANC

```
▾ Unlicensed Mobile Access
    Length Indicator: 38
    0000 .... = Skip Indicator: 0
    .... 0001 = Protocol Discriminator: URR (1)
    URR Message Type: GA-CSR UPLINK DIRECT TRANSFER (112)
  ▾ L3 Message
      URR Information Element: L3 Message (26)
      URR Information Element length: 34
      .... 1001 = Protocol discriminator: SMS messages (9)
      L3 message contents: 39011f000100079133060910 93f013151c0f810094712627...
    ▸ GSM A-I/F DTAP - CP-DATA
    ▸ GSM A-I/F RP - RP-DATA (MS to Network)
    ▾ GSM SMS TPDU (GSM 03.40) SMS-SUBMIT
        0... .... = TP-RP: TP Reply Path parameter is not set in this SMS SUBMIT/DELIVER
        .0.. .... = TP-UDHI: The TP UD field contains only the short message
        ..0. .... = TP-SRR: A status report is not requested
        ...1 0... = TP-VPF: TP-VP field present - relative format (2)
        .... .1.. = TP-RD: Instruct SC to reject duplicates
        .... ..01 = TP-MTI: SMS-SUBMIT (1)
        TP-MR: 28
      ▸ TP-Destination-Address - (0049176272[redacted])
      ▸ TP-PID: 0
      ▸ TP-DCS: 0
        TP-Validity-Period: 63 week(s)
        TP-User-Data-Length: (3) depends on Data-Coding-Scheme
      ▾ TP-User-Data
          SMS text: Tdd
```

- SMS message filtered by GAN proxy
- modified by client
- transfered to real GANC



1. client indicates it is waiting for a SMS
2. proxy identifies uplink direct transfer, L3 body + type SMS
3. forwards message to attacking client
4. client decodes TPDU and destination number
5. adjusts according to our needs and re-injects into the proxy
6. proxy transfers it to the GANC

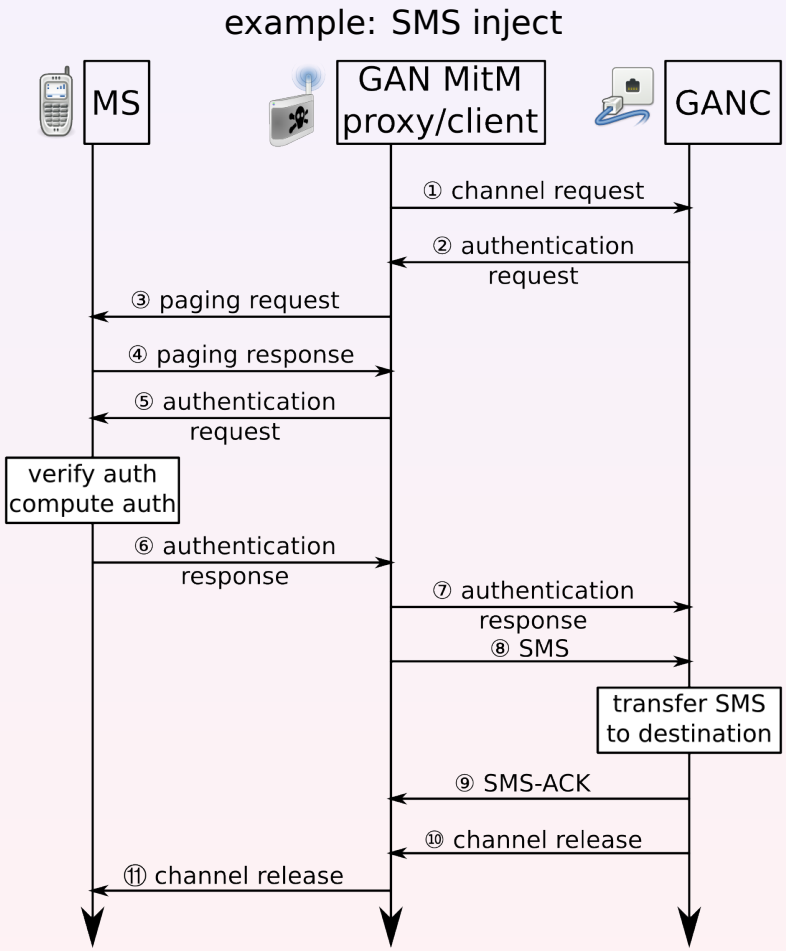
demo time

DEMONSTRATION

SMS modification

how about impersonating subscribers?

- lets use services for free, billed to a victim
- client requires subscriber information
- proxy additionally caches subscriber info (TMSI/IMSI) for each MS-GANC connection
- phone needed for authentication
- applies to any traffic (SMS,voice,data)
- victim is impersonated



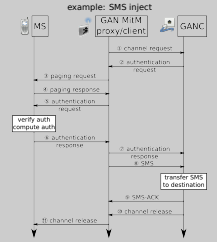
⚔ end-user attacks

playing with traffic

how about impersonating subscribers?

how about impersonating subscribers?

- lets use services for free, billed to a victim
- client requires subscriber information
- proxy additionally caches subscriber info (TMSI/IMSI) for each MS-GANC connection
- phone needed for authentication
- applies to any traffic (SMS,voice,data)
- victim is impersonated



1. client requests subscriber information from the proxy (IMSI/TMSI)
2. issues a service request (call, data, sms, ...) with subscriber information
3. network asks for authentication
4. attack client can't answer this because the secret stored on the victims USIM is required to compute response
5. proxy pages victim and forwards the AUTH request
6. victim assumes a service is coming in, answers AUTH request
7. proxy relays response to the operator and notifies client about the new state
8. client continues injecting messages on behalf of the victim, free for the attacker, billed to the user
9. injection can also work the other way round, to attack phones

demo time

DEMONSTRATION

SMS injection

return of the IMSI detach

- IMSI detach DoS discovered by Sylvaint Munaut in 2010 [2]
  $\Rightarrow$ results in discontinued delivery of MT services (call, sms,...)
  $\Rightarrow$ network assumes subscriber went offline
- detach message is unauthenticated
- however, this is limited to a geographical area (served by a specific VLR)
- user can not receive calls

[2] http://security.osmocom.org/trac/ticket/2

return of the IMSI detach

- IMSI detach DoS discovered by Sylvaint Munaut in 2010 [2]
  ⇒ results in discontinued delivery of MT services (call, sms,...)
  ⇒ network assumes subscriber went offline
- detach message is unauthenticated
- however, this is limited to a geographical area (served by a specific VLR)
- user can not receive calls

[2] http://security.osmocom.org/trac/ticket/2

1. an attacker can send an IMSI detach message to cause an interruption of mobile terminated services
2. MSC forwards detach message to VLR and marks the subscriber as detached
3. VLR notifies HLR of the detach via Location Cancel Request
4. as a result the network assumes the subscriber is not available anymore
5. this is limited to a geographical area
6. if you fake an IMSI detach with subscriber information unknown to your current VLR, the message will be ignored
7. so the attack works only against victims in the same VLR

imsi detach in femtocell ecosystem

- proximity constraint not existent in femtocell network
- devices reside in various geographical areas
- but all subscribers meet in one back-end system ⇒ and they are all handled by one femtocell VLR (at least for SFR) ☺

- we can send IMSI detach payloads via L3 msg in GAN
  ⇒ we can detach any femtocell subscriber, no proximity needed!

demo time

DEMONSTRATION

IMSI detach

attacking other femtocells

- attack surface limited:
  - network protocols: NTP, DNS spoofing (not tested)
  - services: webserver, TR-069 provisioning (feasible)
- both HTTP. TR-069 is additionally powered by SOAP and XML
- lots of potential parsing fail
- all services run as root

attacking other femtocells

- attack surface limited:
  - network protocols: NTP, DNS spoofing (not tested)
  - services: webserver, TR-069 provisioning (feasible)
- both HTTP. TR-069 is additionally powered by SOAP and XML
- lots of potential parsing fail
- all services run as root

1. the attack surface of the femtocell from a network attackers perspective is rather limited
2. all devices make heavy used of NTP and DNS, besides IPsec
3. NTP functionality is based on ntpdate. used as a reliable clock source for frequency stability
4. DNS is done by libc functionality. used to identify operator services
5. both based on UDP, thus spoof'able (NTP also not using authentication headers
6. mentioned web services are accessible from within the network
7. and TR-069 is open so that the femtocell operator can push updates
8. way more potential to find bugs by reversing the software

femtocell remote root (0day)

- we went for the web service (wsal)
- based on shttpd [3]/mongoose [4]/yassl embedded webserver
- we found a stack-based buffer overflow in the processing of HTTP PUT requests
- direct communication between femtocells is not filtered by SFR
- exploit allows us to root **any** femtocell within the network
- `www.sec.t-labs.tu-berlin.de/~nico/wsal_root.py`

[3] http://docs.huihoo.com/shttpd/
[4] http://code.google.com/p/mongoose/

- we went for the web service (wsal)
- based on shttpd [3]/mongoose [4]/yassl embedded webserver
- we found a stack-based buffer overflow in the processing of HTTP PUT requests
- direct communication between femtocells is not filtered by SFR
- exploit allows us to root **any** femtocell within the network
- `www.sec.t-labs.tu-berlin.de/~nico/wsal_root.py`

[3] http://docs.huihoo.com/shttpd/
[4] http://code.google.com/p/mongoose/

1. we decided to audit the web service in more detail, both because of the good knowledge about involved protocols and as we later found out the service is based on an open source project
2. we discovered a buffer overflow in the PUT processing
3. PUT itself is not much of value because the web server directory is read-only and directly traversal is handled by the web service
4. however the buffer overflow itself allows us to reliably root other devices
5. this is extremely serious because most of the previous threat now leverage from a local problem to a **global** problem

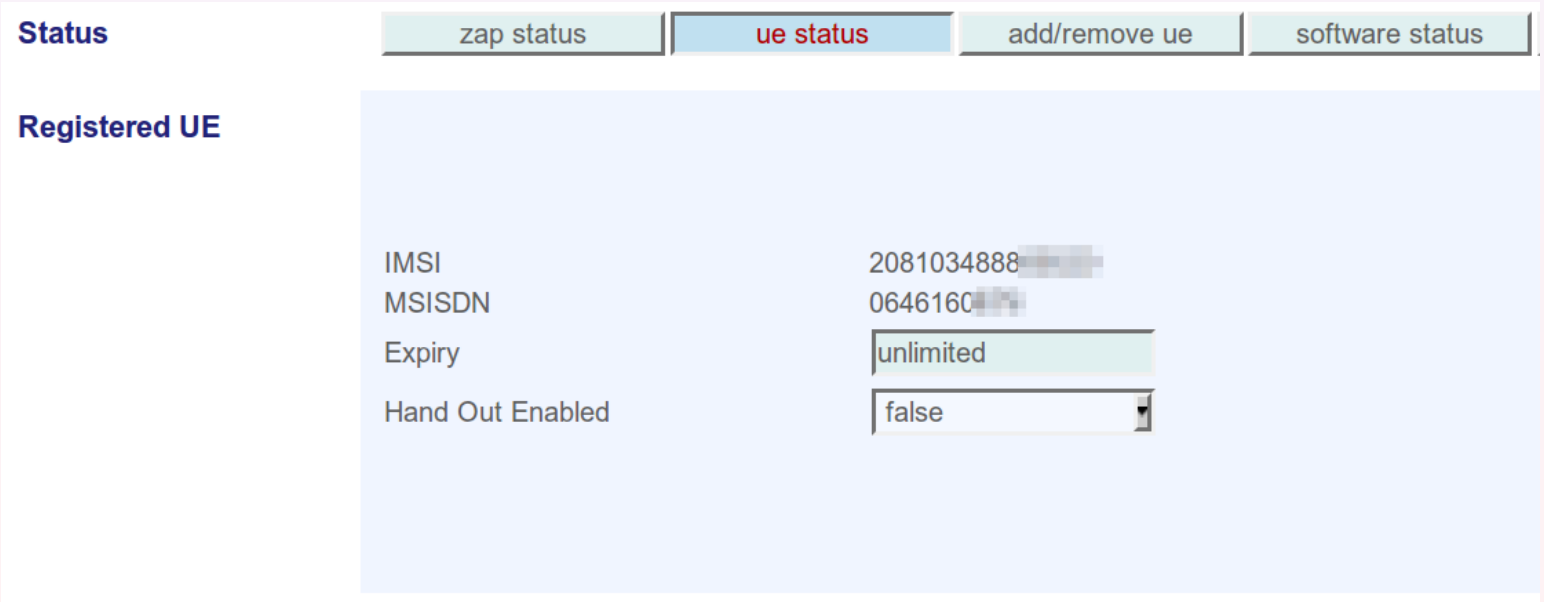
demo time

DEMONSTRATION

remote root

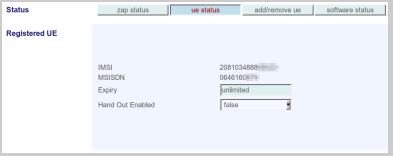
collecting subscribers

- other femtocell are accessible within the network
- website is also accessible
- leaks **phone number** and IMSI of registered subscriber
- **wink** IMSI detach ⇒ detach whole network



collecting subscribers

- other femtocell are accessible within the network
- website is also accessible
- leaks **phone number** and IMSI of registered subscriber
- **wink** IMSI detach ⇒ detach whole network



1. scraping can easily be done
2. there is a lot more info: access mode, software version, ...
3. the scraped IMSIs can be abused to build a database and detach all subscribers at once
4. would block incoming services for the whole network

locating subscribers

- location verification performed by OAM
- femtocell scan for neighbour cells

Engineering

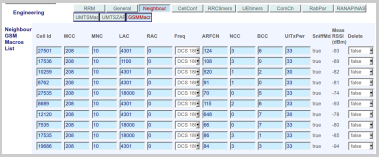
RRM | General | Neighbour | CellConf | RRCtimers | UEtimers | ComCh | RabPar | RANAP/NAS

UMTSMac | UMTSZAR | GSMMacr

Neighbour GSM Macros List

Cell Id	MCC	MNC	LAC	RAC	Freq	ARFCN	NCC	BCC	UlTxPwr	SniffMd	Meas RSSI (dBm)	Delete
27501	208	10	4301	0	DCS 18(	124	3	6	33	true	-93	false
17536	208	10	1100	0	DCS 18(	108	3	0	33	true	-89	false
10259	208	10	4301	0	DCS 18(	520	1	2	30	true	-82	false
8762	208	10	4301	0	DCS 18(	91	1	0	33	true	-81	false
27535	208	10	18000	0	DCS 18(	70	0	5	33	true	-74	false
8689	208	10	4301	0	DCS 18(	115	2	6	33	true	-93	false
12120	208	10	4301	0	DCS 18(	648	0	7	30	true	-78	false
7535	208	10	18000	0	DCS 18(	66	0	7	33	true	-80	false
17535	208	10	18000	0	DCS 18(	86	3	1	33	true	-85	false
19686	208	10	4301	0	DCS 18(	84	3	3	33	true	-94	false

- location verification performed by OAM
- femtocell scan for neighbour cells



1. location verification is a security aspect defined by the specification
2. used to enforce femtocell location, avoid roaming evasion, respect radio licenses, ...
3. other methods are: geoIP and GPS (if available on the board)

global control

- web-site/database is not read-only
- OAMP, image and GAN server can also be set
- or using root exploit
- traffic can be redirected to our femtocell (either settings or iptables)

$\Rightarrow$ any femtocell can be flashed
$\Rightarrow$ any femtocell subscriber communication can be intercepted, modified and impersonated

meeting the usual suspects

HNS servers run typical Open Source software, not especially secured, e.g:

- MySQL, SSH, NFS, Apache (with directory indexing), ... available
- FTP used to submit performance measurement reports, including femtocell identity and activity
- all devices share the same FTP account
- vsftpd users are system users, SSH is open :D

advanced access

- SeGW is required to access the network
- authentication is performed via the SIM (removable)
- how about configuring an IPsec client with this SIM?

⇒ no hardware and software limitation
⇒ no femtocell required anymore
⇒ femtocells don't act as a great wall to protect the operator network anymore :D

stairways to heaven

- attacks on operator network
- signaling attacks (not blocked)
- free HLR queries
- leveraging access to:
  - other Access Networks
  - Core Network
- ...

other femtocell research

- THC vodafone `http://wiki.thc.org/vodafone`, rooted in 2009, unfortunately bug fixed since 2 years
- Samsung femtocell
  `http://code.google.com/p/samsung-femtocell/`
- clearly shows that this is no single operator problem and might cause some pain
- femtocell architecture is defective by design, security wise

- THC vodafone `http://wiki.thc.org/vodafone`, rooted in 2009, unfortunately bug fixed since 2 years
- Samsung femtocell `http://code.google.com/p/samsung-femtocell/`
- clearly shows that this is no single operator problem and might cause some pain
- femtocell architecture is defective by design, security wise

1. operator infrastructure is trusted, weakly secured
2. femtocells are physically accessible by attackers
3. compromised devices endangers the mobile telecommunication network infrastructure

thanks (in no particular order)

- Jean-Pierre Seifert
- Collin Mulliner
- Benjamin Michéle
- Dieter Spaar
- K2

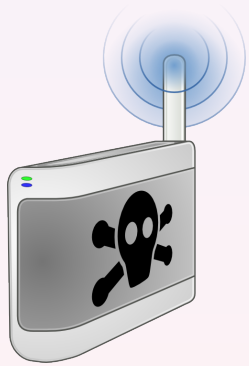
- Jean-Pierre Seifert
- Collin Mulliner
- Benjamin Michéle
- Dieter Spaar
- K2

1. hay collecting pictures: Basil & Tracy,
   `http://www.flickr.com/photos/basilb/`
2. hole in haystack pictures: funkypancake,
   `http://www.flickr.com/photos/funkypancake/`
3. hay eater: Seattle Roll,
   `http://www.flickr.com/photos/seattle_roll/`

the end

thank you for your attention

questions?



contact us

- Nico Golde <nico@sec.t-labs.tu-berlin.de> @iamnion
- Kévin Redon <kredon@sec.t-labs.tu-berlin.de>
- Ravi Borgaonkar <ravii@sec.t-labs.tu-berlin.de> @raviborgaonkar
- or just femtocell@sec.t-labs.tu-berlin.de
- all material from this talk (including tools) will be available one week after Black Hat at:

  `http://tinyurl.com/sectfemtocellhacks`

extended coverage

- femtocells have a small coverage (by definition, 25-50m)
- signal range can be increased using amplifier and external antenna

extended coverage

- femtocells have a small coverage (by definition, 25-50m)
- signal range can be increased using amplifier and external antenna

1. the board has an antenna connector
2. used to test the device while/after manufacturing, without emitting into the air