



ERNW
providing security.

TODAY:

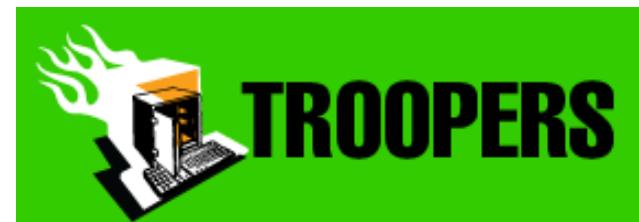
**Practical security
research on 3G and
4G mobile
telecommunications
networks**

Daniel Mende, Enno Rey

{dmende, erey}@ernw.de

Who we are

- **Old-school network geeks,
working as security researchers for**
- **Germany based ERNW GmbH**
 - Independent
 - Deep technical knowledge
 - Structured (assessment) approach
 - Business reasonable recommendations
 - We understand corporate
- **Blog: www.insinuator.net**
- **Conference: www.troopers.de**



Introductory notes

- **Yes, we changed the talk's title a little bit**
 - It's pretty much the same content though
(why did you change it then?!*)
 - ... but a different line of the story...
- **Given the sensitivity of the material some severe NDAs kick in.
And being responsible researchers we won't give details of
“innocent parties in the Internet” either.**
- **Still the main message of this presentation is – as in most of our
talks – to provide some “from theory to reality” perspective ;-)**
- **For the record: when we use terms like “sysadmin” or “security
officer”, these potentially designate men or women.**



Agenda

- **Intro & Basics**
- **Some notes on 3G security research**
- **Some notes on 4G security research**
- **Conclusions**



The setting

- **Imagine**
~~a bunch of hackers security researchers~~
wanting to get into \$SOME_TOPIC.
- Where this topic is “mobile telecommunications networks security”.
 - From an infrastructure perspective.
 - We do not (yet) understand very much of mobile terminals, at least not when it comes to the baseband controller stacks.
 - And quite some research has already been performed (TSTF et.al.).
 - Hi Philippe! Sorry for not being able to come to HES. Still, you’re cool, of course ;-)
- **How would you tackle the task?**



The traditional way

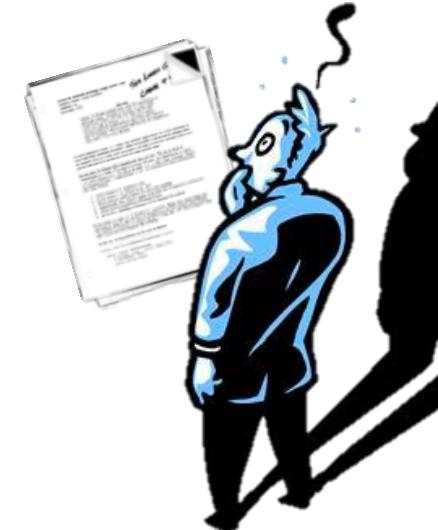
- **Read the specs & try to understand the big picture**

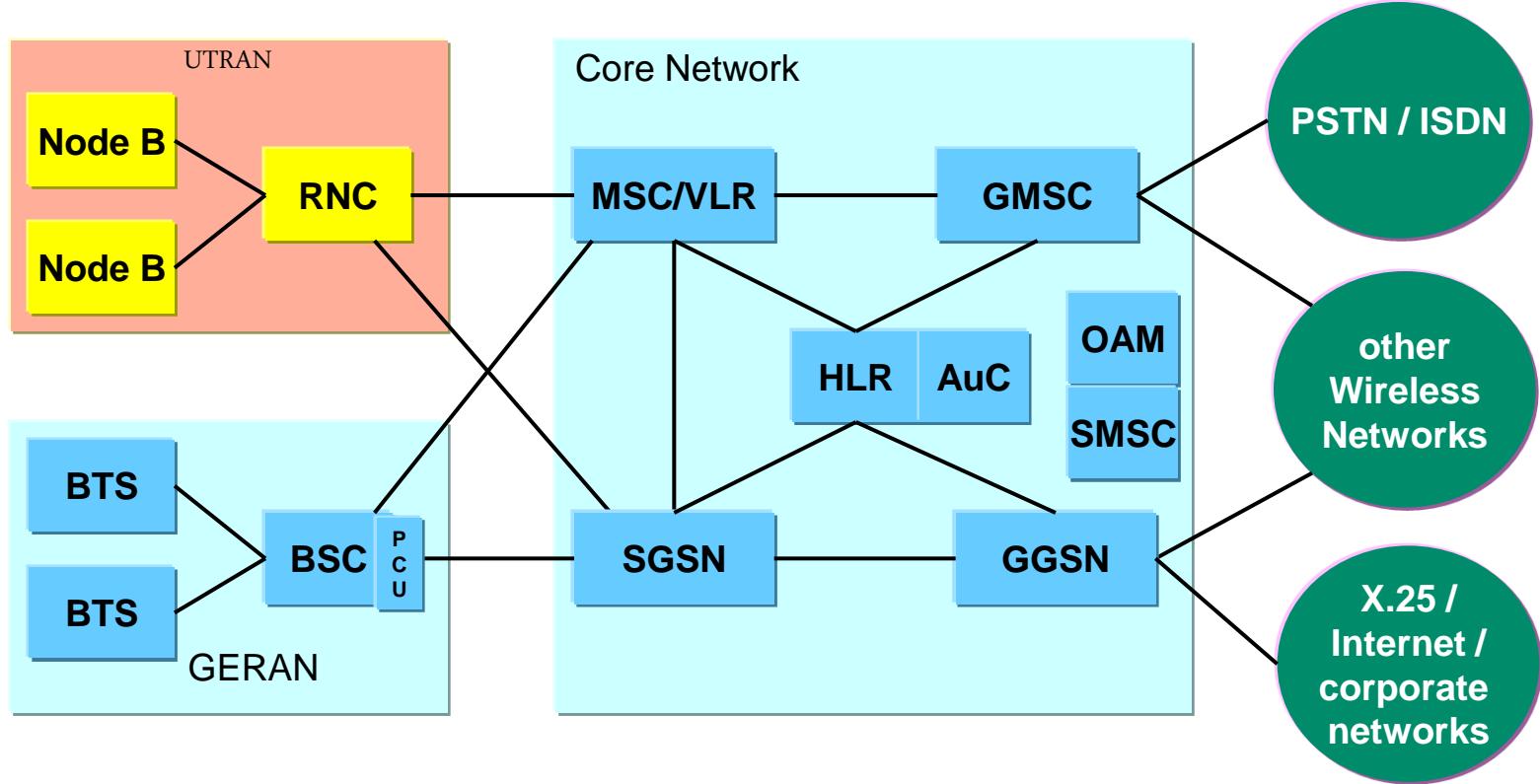


from "Hackers", 1995

Standards

- In mobile telco world everything standardized by **3GPP**
- **3GPP: collaboration between groups of telco standard orgs**
 - Which “type of documents” do you think these guys produce? ;-)
- **3GPP standards structured as/bundled in *releases***
 - 1992: *Phase 1* (GSM)
 - 2000: *Release 99* incl. first specification of 3G UMTS
 - 2008: Release 8 incl. first specification of LTE stuff
- **At times, 3GPP standards are a bit... bulky ;-)**





RAN: Radio Access Network

UTRAN: UMTS RAN BTS: Base Transceiver Station

GERAN: GSM Enhanced RAN

PCU: Paket Control Unit

RNC: Radio Network Controller

BSC:

Base Station Controller

HLR: Home Location Register

MSC: Mobile Switching Center

VLR: Visitor Location Register

GMSC: Gateway MSC

SMSC: Short Message Service Center

GSN: GPRS Support Node

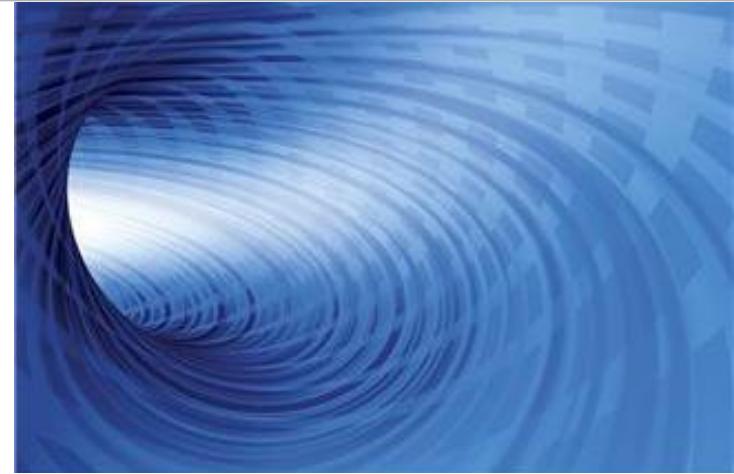
AuC: Authentication Center

OAM: Operation Administration & Maintenance

S/GGSN: Serving/Gateway GSN

Source: 3GPP

- **GPRS Tunneling Protocol**
- **IP-based protocol initially used to carry GPRS within GSM and UMTS networks.**
 - Plays major role in 4G networks as well.
- **Three variants**
 - GTP-C used for control plane (signaling)
 - GTP-U used for user data
 - GTP' used for charging data



■ **GTP-C**

- Control section of the GTP standard
- In 3G used for signaling between SGSN and GGSN
- Activates and deactivates GTP sessions
- In roaming scenarios this happens between different operators.



■ **GTP-U**

- Used for data transport between the RAN and the core network
- Can tunnel packets in several formats: IPv4, IPv6, PPP etc. ...

■ **GTP'**

- Used in 3G for transmitting charging data from the CDF to the CGF.

GTP Header

■ The GTP Header

- GTPv1

Bit 0-2	3	4	5	6	7	8-15	16-23	24-31
Version	Protocol Type	Reserved	Extension Header Flag	Sequence Number Flag	N-PDU Number Flag	Message Type	Total length	
TEID								
Sequence number					N-PDU number		Next extension header type	

- GTPv2

Bit 0-2	3	4	5-7	8-15	16-23	24-31		
Version	Piggybacking flag (P)	TEID flag (T)	Spare	Message Type	Total length			
TEID (only present if T=1)								
Sequence number				Spare				

Some GTP message types

■ **GTP-C provides messages for**

- Echo
- Create/Update/Delete/Initiate PDP Context
- PDU Notification
- Send Routing Information
- Failure Report
- Note MS/MS info
- Identification
- SGSN Context
- Forward Relocation
- Forward SRNS Context
- RAN information
- MBMS Notification/Context/(De-)Registration/Session



GTP from a security perspective

- **Unauthenticated protocol**
- **No inherent security properties**
- **Trusted environment assumed**

- **Is used to perform “quite some functions”**
 - Session establishment (“activate PDP context”)
 - Forwarding of packets
 - Charging related stuff

- **All these functions rely on certain protocol fields**
 - Presumably only known to valid peers... which are isolated anyway...



- Read the specs & try to understand the big picture 
- Build a lab



Build a lab

- **This was/is not an easy task.**
 - Even for an organisation like us disposing of quite_some.hardware and being populated by guys with an addiction for fancy_devices.
- **Actually we initially wasted quite some money on ebay.**
- ***OpenGGSN* was not regarded as a feasible option either.**
 - Does not support many features/functions.
- **After some digging around & cycles, we found out...**



GTP on 7200VXR

- **7200 is capable of serving as GGSN in a 3G net**
 - **Special image needed**
- 
- **service gprs ggsn config command**
 - **Once activated, device opens up udp/2123 and udp/2152**
 - **gtp-echo-requests (gtp-v1) are answered on both ports.**
 - **gtp-create-PDPcontext-requests (gtp-v1) are answered on udp/2123 (gtp-c) if a valid/configured APN is given in the request.**

Further down the road...

- **Read the specs & try to understand the big picture** 

- **Build a lab** 

- **And now?** 

- Read more specs ;-)
- How to handle protocols you've never touched before?
 - Right: fuzz them, if nothing else ;-)
 - Same approach will turn out to be helpful later...
 - Btw: <http://www.insinuator.net/2011/05/update-for-your-fuzzing-toolkit/>



GTP on 7200VXR – DoS

- **Sending out _a.lot_ of gtp-echo-requests will stress the 7200er CPU to 100%, so that**
 - No ICMP pings answered anymore.
 - No remote mgmt (ssh/telnet) possible (refuses connections on tcp/22).
 - No further GTP requests processed.
- **Sending out _a.lot_ of gtp-create-PDPcontext-requests will also stress the device, so that only ~30% of all (valid and bogus) requests are answered.**
- **However a valid APN is needed**
 - We'll get back to this ☺



Well, DoS is lame, isn't it?

- **Once some relevant parameters are known, we can send hand-crafted GTP packets to the GGSN.**
 - Remember: no authentication properties [and, for that matter, no integrity protection either]
- **We've not yet figured which exact attack scenarios can be implemented.**
- **Certainly all types of “session interference“ might be possible**
 - Incl. sending data traffic billed on \$SOME_OTHER_SUBSCRIBER??

DEMO

Into the great wide open

- **Read the specs & try to understand the big picture** 

- **Build a lab** 

- **Ok, got some results in the lab.** 

- **What's next?** 

- Right: any other GTP speakers out there? ;-)
 - Yes, “out there“ means “the Internet“...





DON'T WORRY SIR,



Notes from the field

TS_132216v1062009.pdf (SECURED) - Adobe Reader

Bookmarks

- Intellectual Property Rights
- Foreword
- Forward
- Introduction
- 1 Scope
- 2 References
- 3 Definitions, symbols and abbreviations
 - 3.1 Definitions
 - 3.2 Symbols
 - 3.3 Abbreviations
- 4 Overview over network domain security for IP based protocols
 - 4.1 Introduction
 - 4.2 Protection at the network layer
 - 4.3 Security for native IP based protocols
 - 4.4 Security domains
 - 4.5 Security Gateways (SEGs)
- 5 Key management and distribution architecture for NDS/IP
 - 5.1 Security services afforded to the protocols
 - 5.2 Security Associations (SAs)
 - 5.3 Profiling of IPsec
 - 5.4 Operations and

B.1 The need for security protection

The GPRS Tunnelling Protocol (GTP) is defined in 3GPP TS 29.060 [6]. The GTP protocol includes both the GTP control plane signalling (GTP-C) and user plane data transfer (GTP-U) procedures. GTP is defined for Gn interface, i.e. the interface between GSNs within a PLMN, and for the Gp interface between GSNs in different PLMNs.

GTP-C is used for traffic that is sensitive in various ways including traffic that is:

- crucial with respect to both the integrity, safety and consistency of the network;
- essential in order to provide the user with the required services;
- crucial in order to protect the user data in the access network and that might compromise the security of the user data should it be revealed.

Amongst the data that clearly can be considered sensitive are the mobility management messages, the authentication data and MM context data. Therefore, it is necessary to apply security protection to GTP signalling messages (GTP-C).

Network domain security is not intended to cover protection of user plane data and hence GTP-U is not protected by NDS/IP mechanisms.

Table 1 presents a list of GTP interfaces that shall be considered by NDS/IP.

Table 1: GTP Interfaces that are affected by NDS/IP

Interface	Description	Affected protocol
Gn	Interface between GSNs within the same network	GTP
Gp	Interface between GSNs in different PLMNs.	GTP

So, in theory...

- ... no device speaking GTP-C should ever been reachable from the Internet (at least not [on] the Gp interface).
- Well, in theory.



GTP in the wild



```
greif@loki ~/scans $ wc -l v1_*
```

[...]

```
2954772 total
```



```
greif@loki ~/scans $ wc -l v2_*
```

[...]

```
2951685 total
```

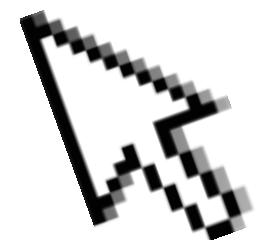
An updated version of gtp-scan will be released after HITB. Pls check www.insinuator.net for updates...

To give you an idea of the script

```
#gtp-scan.py -w2 192.168.85.0/24  
starting scan of 192.168.85.0/24
```

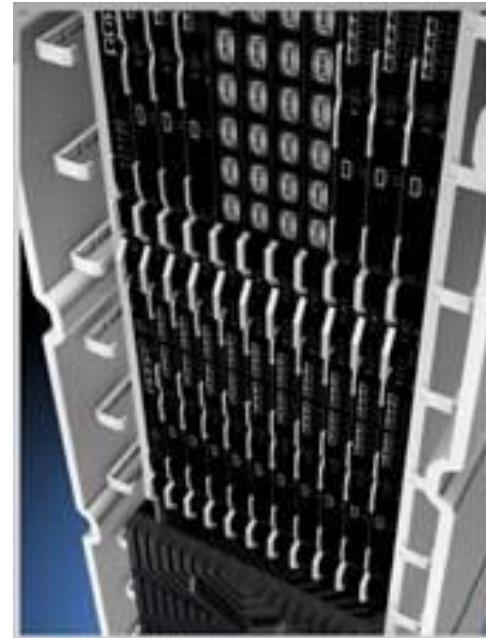
```
### 192.168.85.30 up, from udp/2123(gtp-c)  
sent 3202000600000000c3d00000edf  
*** VALID LEN IN GTP: version = 1 flags =  
xxx10010 type = 2 len = 6 data = 000000  
*** ECHO RESPONSE
```

cooling down for 2 sec...
done



Devices found

- **ZXUN xGW**

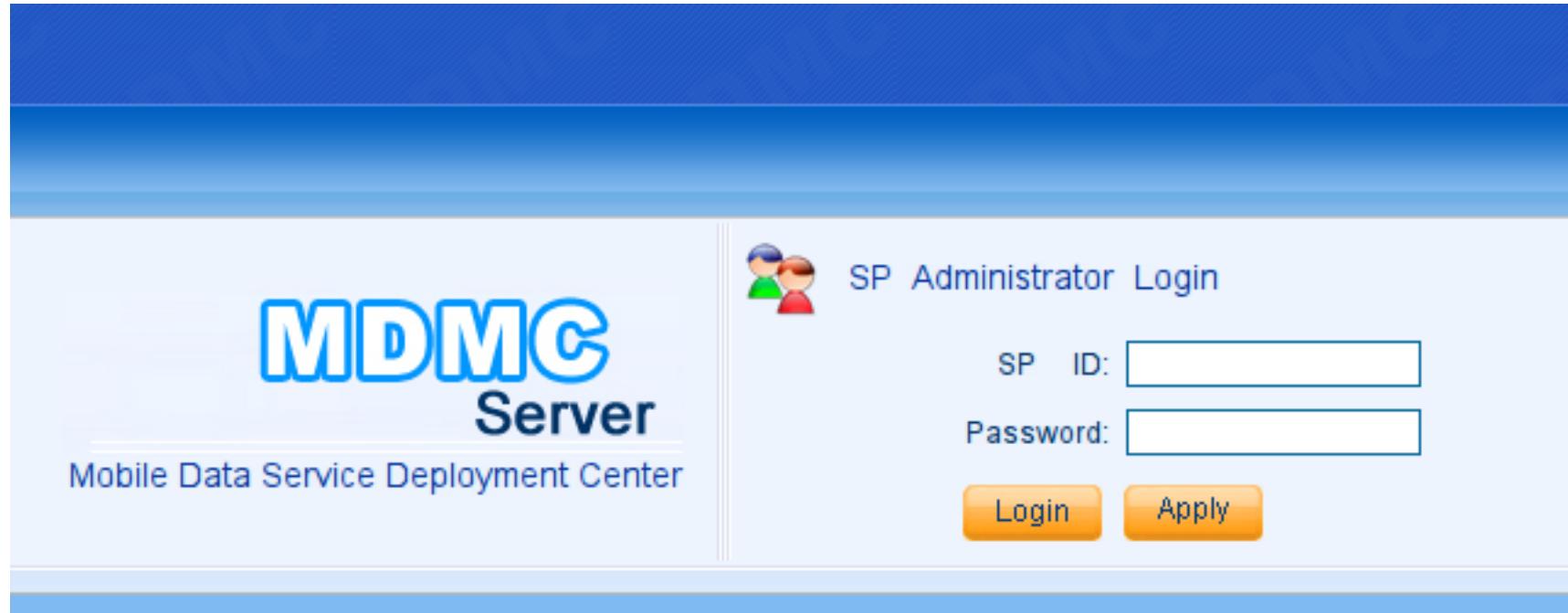


Devices found (2)

- SUSE LINUX Enterprise Server 9 (i586) - Kernel 2.6.5-7.201-bigsmp
- SunOS comptelC 5.10 Generic_118833-23 sun4u
- Cisco IOS Software, 7200 Software (C7200-G6IS-M) , Version 12.3(8)T3



Devices Found (3)



MDMC
Server

Mobile Data Service Deployment Center

SP Administrator Login

SP ID:

Password:

Login **Apply**

Some statistics (GTP-C)



	Version 1	Version 2
AfriNIC	26 (31)	11 (26)
APNIC	81 (131)	97 (90)
ARIN	52 (29)	45 (51)
LACNIC	22 (14)	10 (18)
RIPE	129 (97)	94 (435)
UP	310 (302)	257 (620)

[Values in brackets are the results from our last scan, some months ago]

Some statistics (GTP-U)



	Version 1	Version 2
AfriNIC	13809	13761
APNIC	585733	584156
ARIN	18348	18235
LACNIC	907736	907618
RIPE	1428574	1427899
UP	2954200	2951669

Other findings

- **Some GTP speaker, also listening to SNMP public**
 - ⇒ Internal addresses
 - ⇒ Internal routing Table
 - ⇒ Open ports
 - ⇒ Running processes
 - ⇒ Installed software (including install date ;)
 - ⇒ Whatever is in the MIB



whois



organisation:

org-name:

org-type:

country: GH

address:

address:

address: Accra



whois



organisation:

org-name: xy_org (Cote d'Ivoire)

org-type:

country: CI

address:

address:



status:

owner:

ownerid:

responsible:

address:

address: Asunción (Paraguay)

country: PY



Organisation:

org-name:

org-type:

Country:

Address:

Address:

address: Cairo



- This box had FTP and SSH open. Any guesses re the password?
For the correct answer I might spend a beer, or two.

Back on track for serious research



- **Read the specs & try to understand the big picture** ✓
- **Build a lab** ✓
- **Ok, got some results in the lab.** ✓

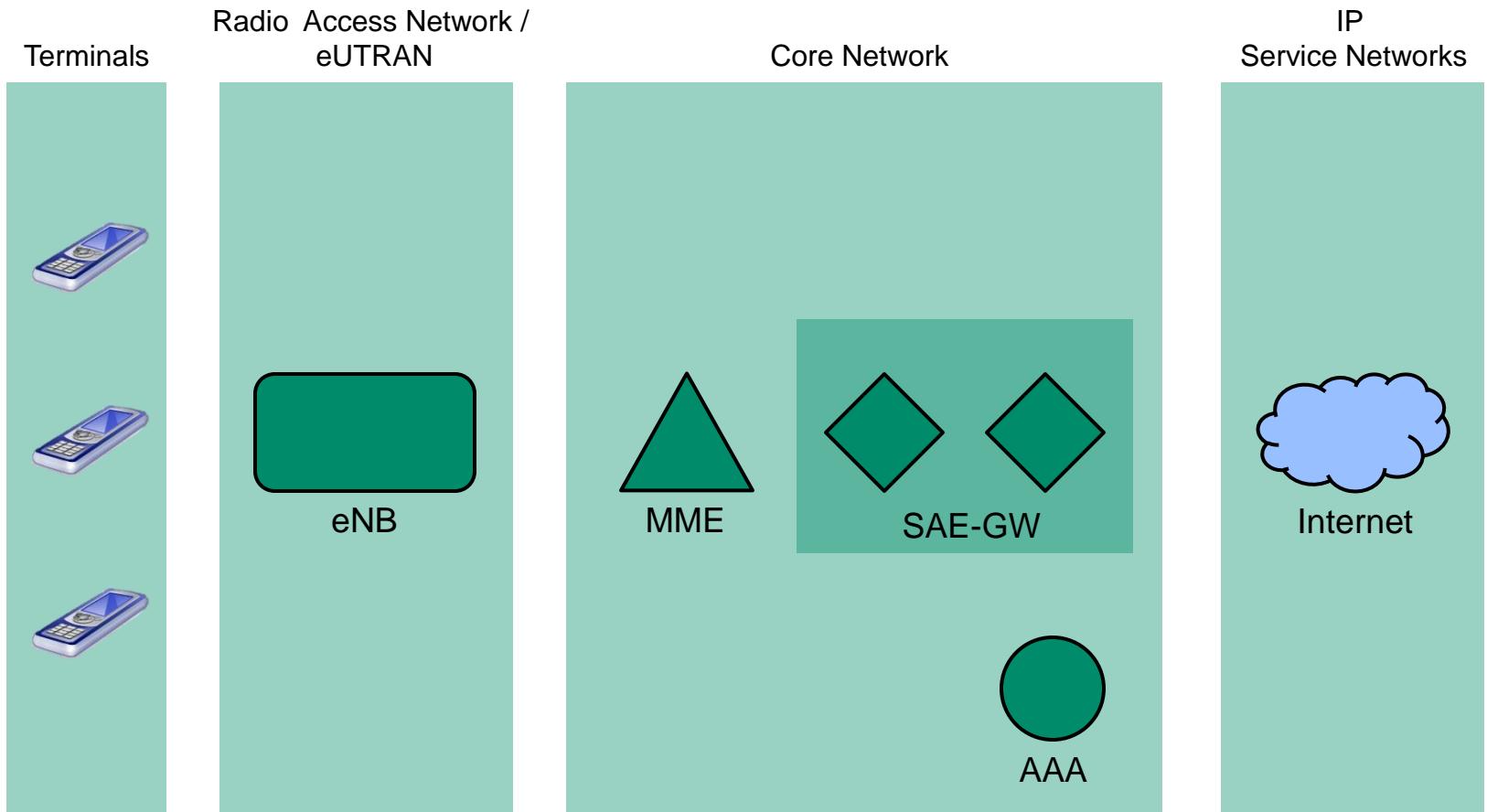
- **What else?**
 - GTP speakers in the Internet. ✓

- **The security researcher's dream option**
 - Find a telco with a lab, engaging the sec_evaluator.





**Second part of the
talk: notes from
the lab**

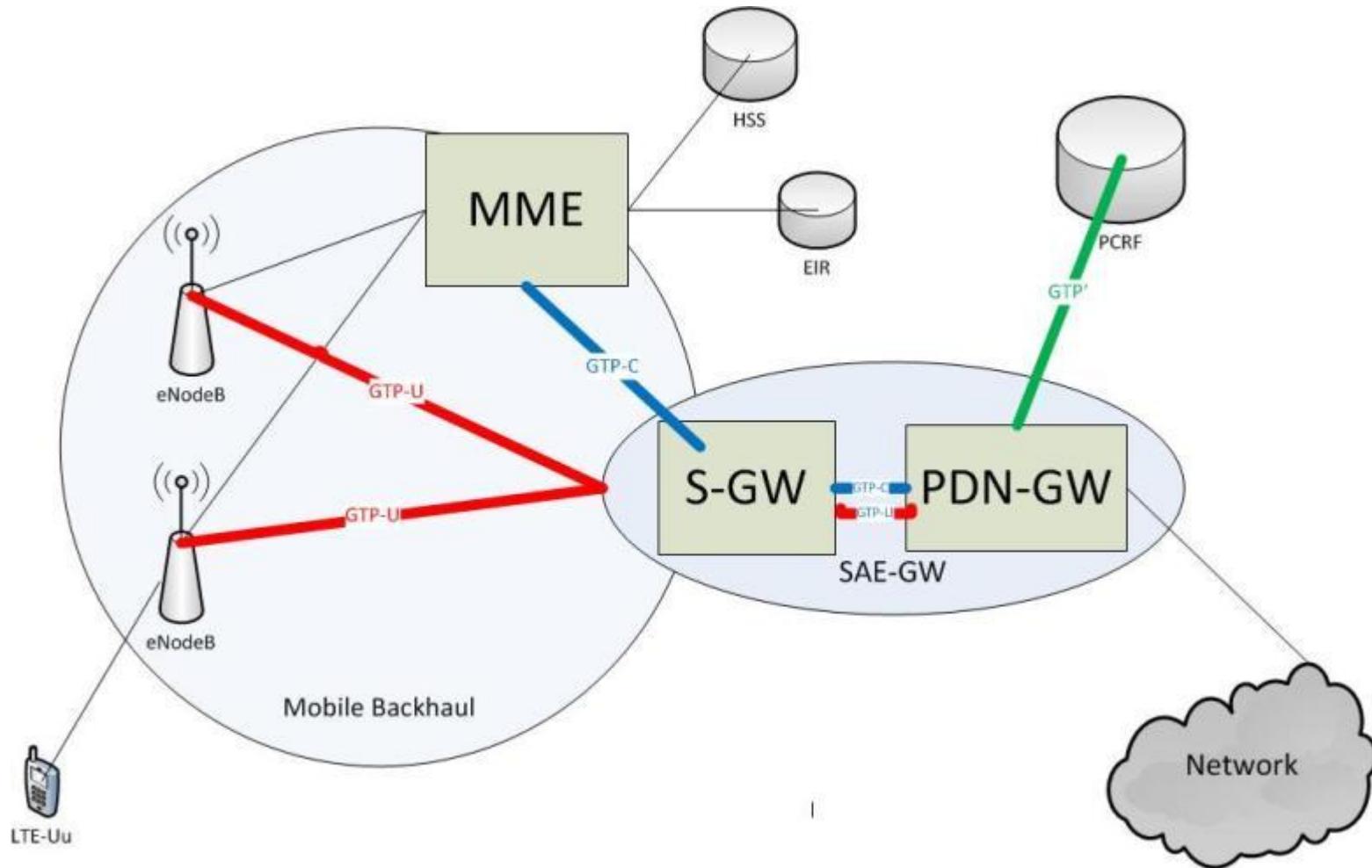


Main Protocols (4G)

- **Transport Layer: (mostly) UDP or SCTP**
- **Generic Packet Tunneling: GTP**
- **All types of signaling:**
 - S1AP, X2AP, GTP-C
- **Authentication**
 - DIAMETER
- **Others**
 - L2TP, DSMIPv6 etc. → Lots of “areas for future research” ;-)



GTP in 4G



SCTP - Overview

■ SCTP

- Stream Control Transmission Protocol
- Specified by IETF, maintained IETF Transport Area (TSVWG) WG

■ Specs:

- RFC 3286 (Introduction)
- RFC 2960 (2000)
- RFC 3309
- RFC 4960 (2007)
- RFC 5062

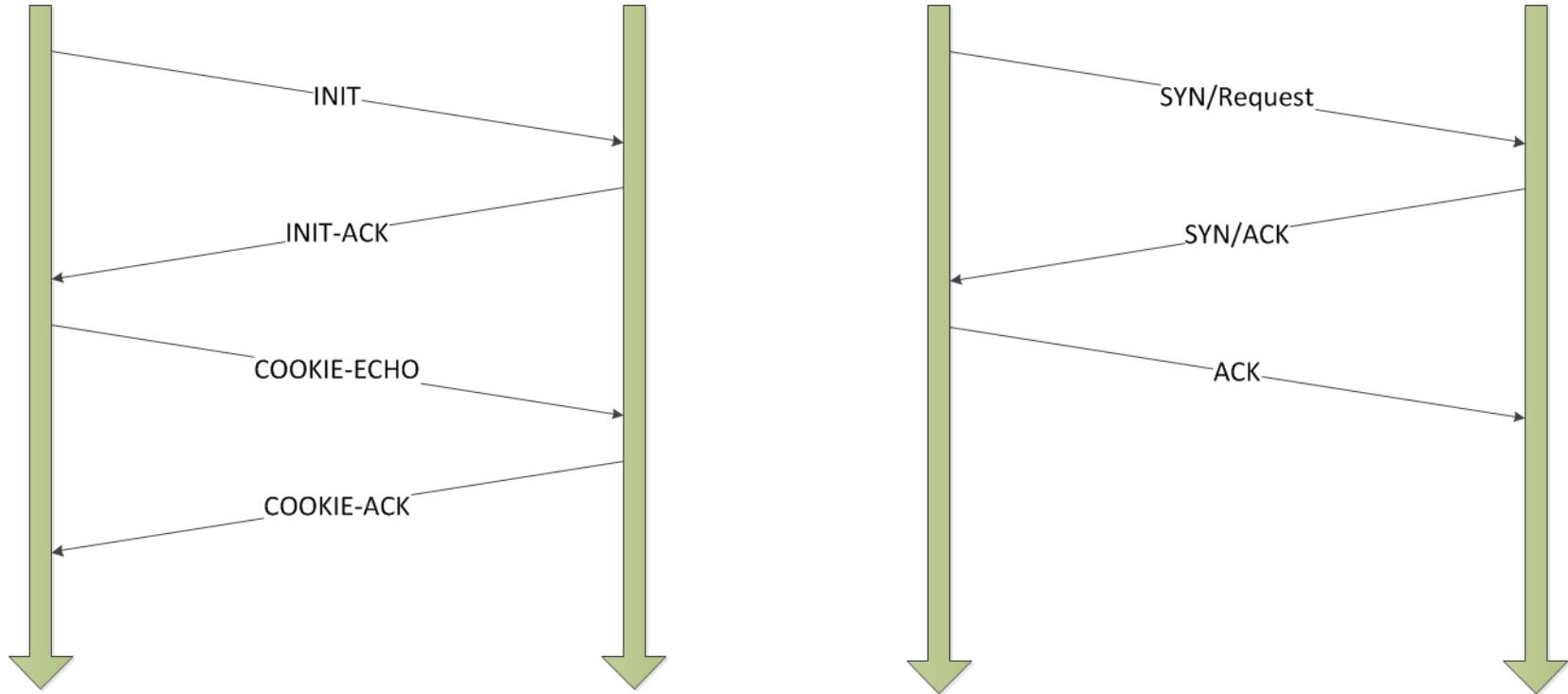


SCTP – 4 way handshake

SCTP

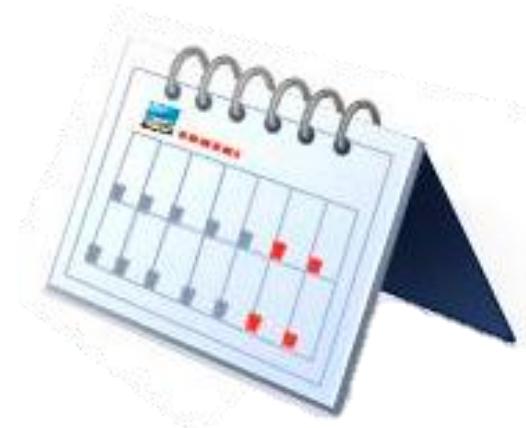
vs.

TCP



SCTP – Timeline

- RFC 2960 (2000): initial spec
- RFC 4960 (2007): “major rewrite”
- RFC 5062 (2007) *Security Attacks Found Against the Stream Control Transmission Protocol (SCTP) and Current Countermeasures*
- So, over time SCTP has changed a bit...



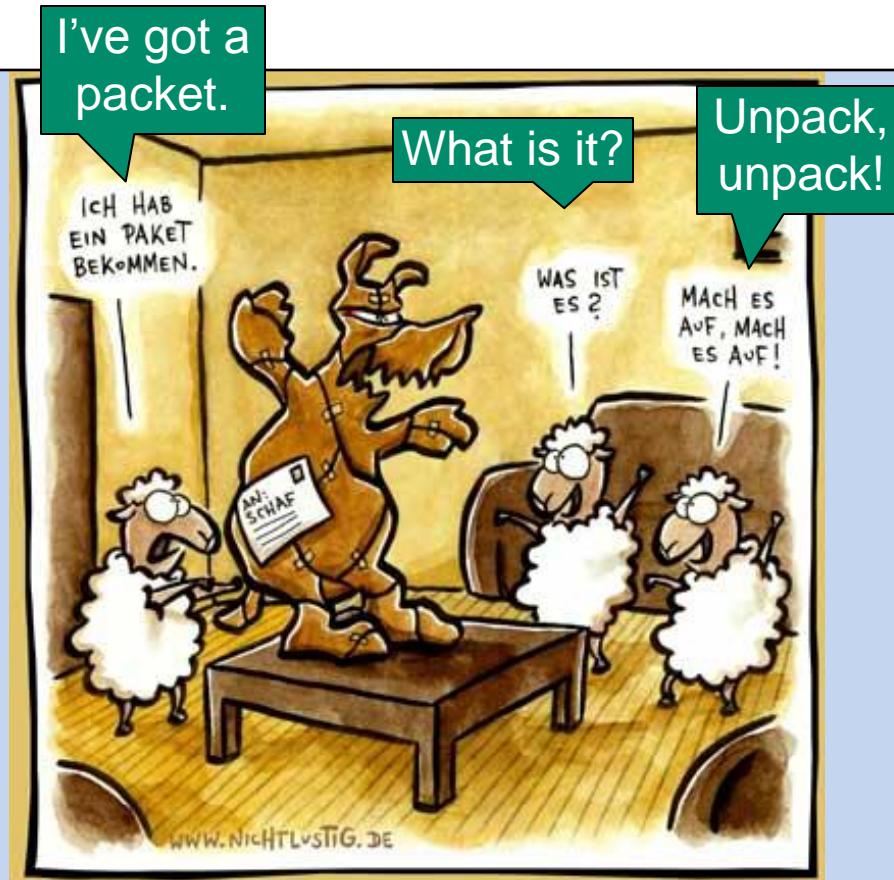
- **Current tools... do not work very well**
 - Probably due to stack rewrites based on RFC 5206 and 4960
- **nmap SCTP does not work “in a satisfactory manner”**
 - -sZ does give results
 - -sY (“half-open handshake”) didn’t show anything useful
 - But we knew the ports were there...
- **Philippe Langlois’ *SCTPscan* didn’t work either.**
- **Daniel wrote quick+dirty “simple SCTP port scanner”.**



SCTP hacked scanner ;)



```
s = socket.socket(socket.AF_INET, socket.SOCK_SEQPACKET)
for i in ip:
    for j in xrange(sys.argv[2], sys.argv[3]):
        time.sleep(0.01)
        try:
            s.connect((j, i))
        except Exception, e:
            print "Port %d closed on %s: %s" % (i, j, e)
        else:
            print "Port %d open on %s" % (i, j)
            s.close()
```



**Some discussion
on attacks from
within telco network**

Attacks from within telco network

■ Attacks from backhaul networks

- Might be geographically dispersed
- Can be protected internally with firewalls, (IPsec) security gateways etc. ... or not...

■ Attacks from core networks

■ Attacks from management networks

- We'll not cover those here as "this is standard stuff"
- Still it should be noted that the operators we know (EU/US space) have quite good operational security practice with regard to these devices.



Backhaul networks – Definition

■ In communication services

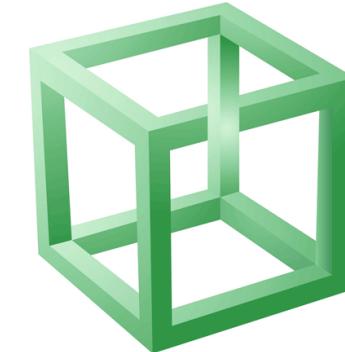
- Used to transport information from one network node to another

■ In mobile communication

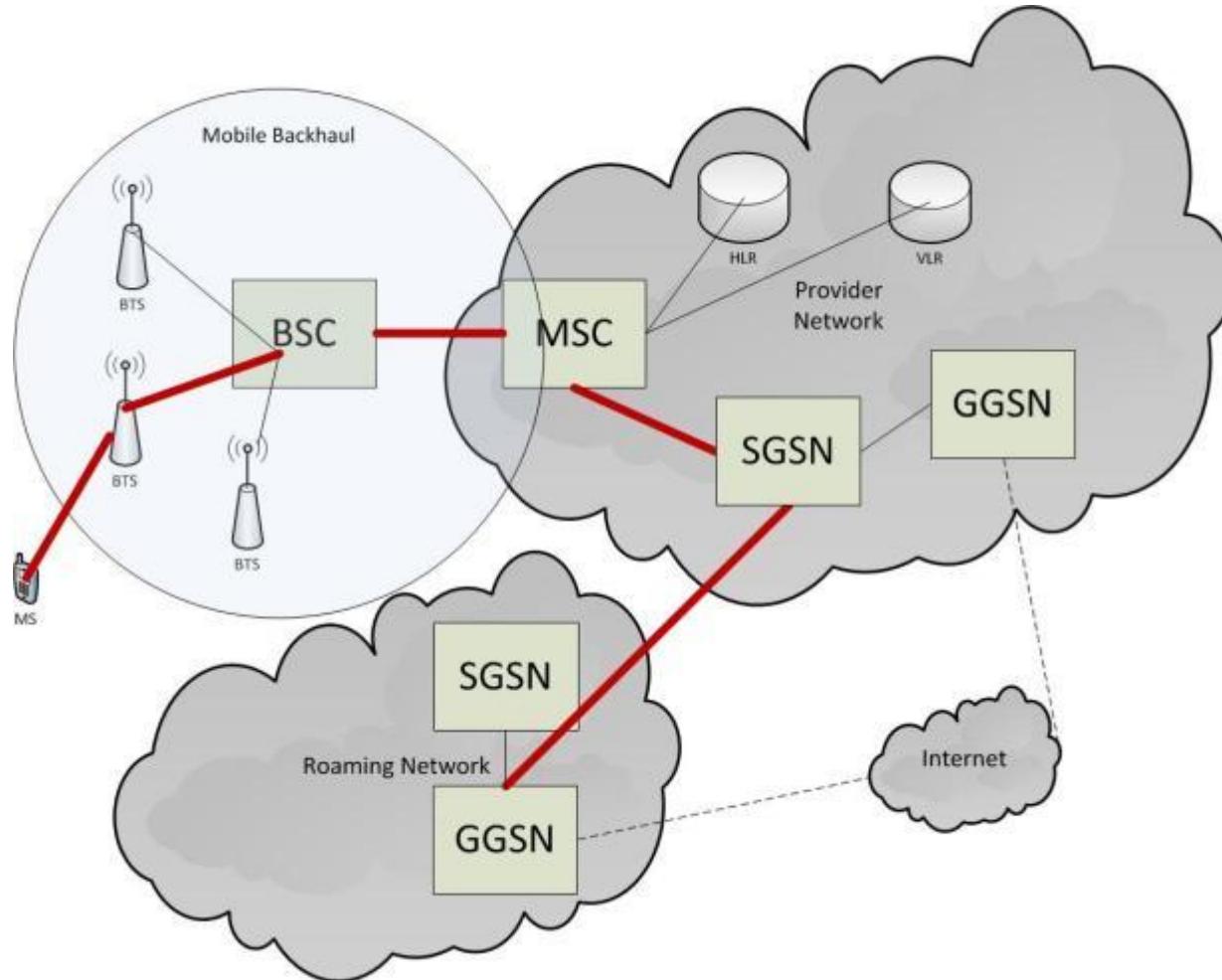
- *Mobile Backhaul*
- Carries data from the RAN to the management network and back.

■ Three primary functions

- Transport
- Aggregation and grooming
- Switching/routing



Mobile Backhaul (3G)

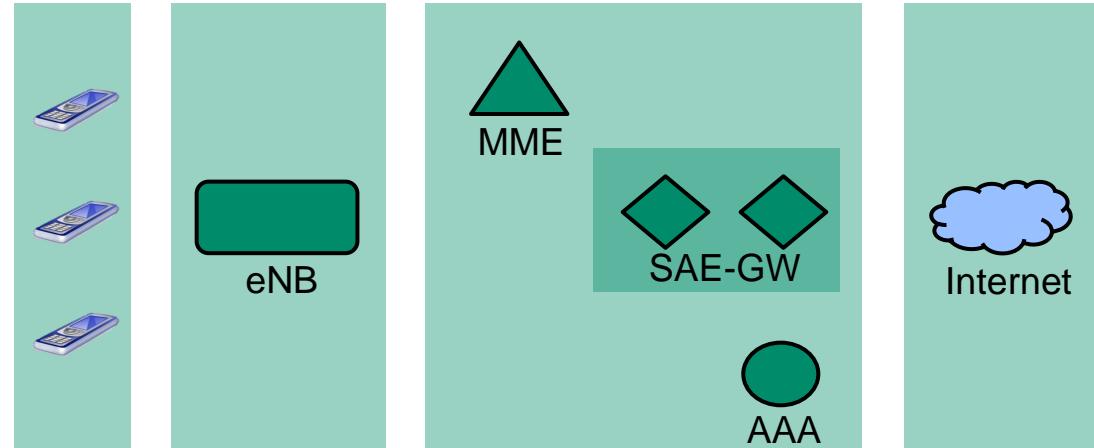


Backhaul networks in 4G

- **4G specific requirements laid out by 3GPP**

- **Includes**

- eNodeB
- MME
- SGW



- **Represents**

- The transport network between eNodeB and MME
- The transport network between eNodeB and SGW

Backhaul networks – Technologies

- **Mostly ATM in the early years (GSM)**
- **PDH/SDH over Microwave, T1/E1**
- **IP/MPLS**
- **“Hybrid Approach“ with data offloading to DSL**
- **Carrier Ethernet**



How to get into backhaul

- **Physical intrusion to some cage located “in the somewhere”**
- **Get access to “network segment”**
 - Microwave
 - DSL
 - Carrier Ethernet
- **4G aggregates “dumb” BTS and BSC/RNC functions on one device → eNB not “dumb” anymore.**



Once you're in (a backhaul network)



■ Attack components

- 3G: SGSN, RNC, NodeB
- 4G: MME, eNB, SAE-GW
- Routers/Switches

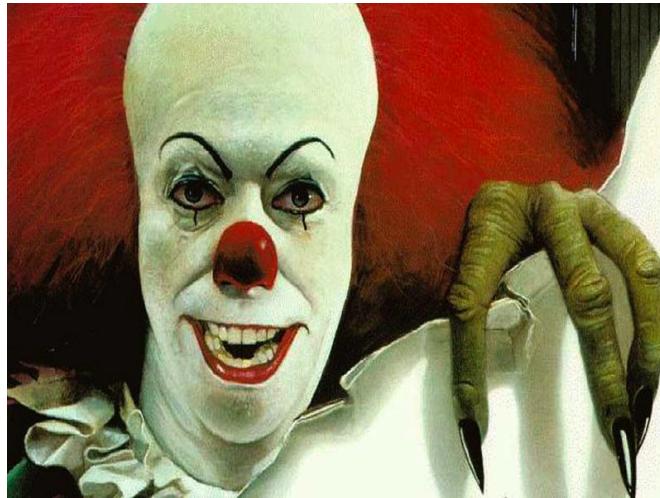
■ Eavesdropping

- Will get you some key material
 - But what would you need this for? Pretty much everything is unencrypt. here anyway.
- That's why 3GPP insists on using IPsec gateways.
- Subsequent question: do (which) operators implement this?
- In standard bodies \$SOME_BIG_COUNTRY (hint: in Asia) strongly opposed this recommendation.



Once you're in backhaul

- **ARP spoofing works smoothly**
 - In particular with all those latency-friendly/tolerating protocols ;-)
 - Apparently not on the (security) radar.



- **4G's All-IP approach comes in handy**
 - 2015 version of *Cain* might support some of these protocols ;-)

Let's get practical

- We were able to perform some testing in the LTE lab of \$SOME_BIG_TELCO_IN_SOME_PART_OF_THE_WORLD.
- In that lab there were no firewalls or (IPsec) security gateways.
 - → Our results might be misleading... or not ;-)



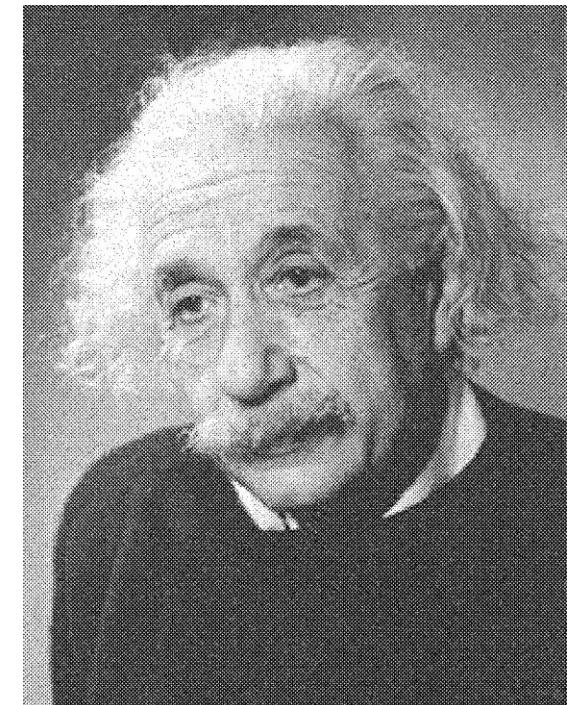
Misc observations: TEID (GTP)

- ***Tunnel Endpoint Identifier***
- **Do I need to explain that it serves to *identify endpoints of tunnels*? ;-)**
 - For each (user) data session.



Misc observations: TEID (II)

- **Apparently some discussion about it being random**
 - For obvious (?) security reasons.
 - Although we were not able to find spec prescribing this.
- **What we observed**
 - 0x00005c35
 - 0x00005c4d
 - 0x00005c65
 - 0x00005c7d
 - 0x00005c95
 - [...]
 - Does this look random to you ?



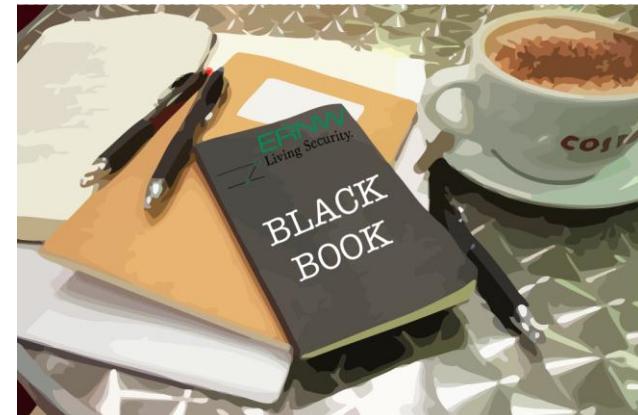
- **Attacker with access to S11 could attack the GTP signaling between MME and SAE-GW**
 - Would require (probably unlikely) network access to core. Or...
 - ... would require S11 to be accessible from the Internet.
- **In any case GTP here transported via UDP**
 - → No pain with spoofing or sth.
- **Potentially DoS of user sessions doable.**
- **In case of switched TEIDs “mixing sessions” possible?**
 - Remember TEID is the only separating element of user sessions.



Some results from practical testing



- “Standard attack approach” did not yield anything
 - At least nothing interesting
 - UDP Echo service was open on some devices and might be exploitable for “mutual amplification attack” between those (with spoofed source IP)
 - Still, this is a bit, well... lame ;-)
- SCTP scanning via *nmap* or *SCTPscan* showed nothing
 - See above as for general problems of current SCTP tool space.
- Using our own SCTP scan tool gave some open ports
 - Some of those “obscure signaling protocols”.



Scanning...

```
nmap scan report for 10.40.68.2
```

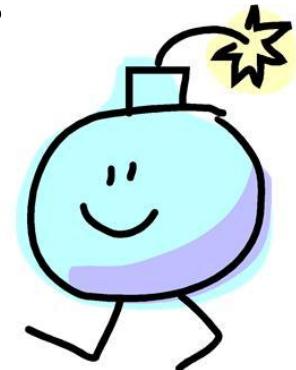
```
[...]
```

PROTOCOL	STATE	SERVICE
1	open	icmp
2	open filtered	igmp
4	open filtered	ip
6	open	tcp
17	open	udp
41	open filtered	ipv6
45	open filtered	idrp
132	open filtered	sctp



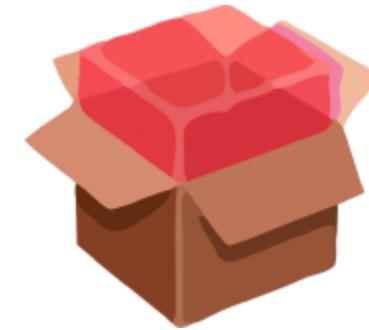
```
Port 36410 closed on 10.40.68.2: [Errno 111] Connection refused
Port 36411 closed on 10.40.68.2: [Errno 111] Connection refused
Port 36412 open on 10.40.68.2
Port 36413 closed on 10.40.68.2: [Errno 111] Connection refused
```

- Started fuzzer
- All of a sudden fuzzing script got slower
 - System sent SCTP ABORT messages instead of “valid responses”
 - Obviously sth has happened ;-)
 - Probably daemon had crashed
- At the same main function of device no more available
 - No further sessions could be established between entities
- Recovered after some minutes
- So we continued fuzzing and changed script
 - At the end of the day system went... down...
 - Ping still worked & mgmt interface. But main function not working.



Postmortem

- **First (!) field of packet payload responsible for “major crash”.**



- **Targeted code was running in kernel.**



- **All that glitters is not gold...**

More potential attacks

- **Theoretical at this point**
 - But we will continue testing,
either with own equipment or in that lab.
- **It seems some authentication info is sometimes cached.**
- **We assume some signaling protocols can be DoSed on
“the function level” once “we get the circumstances right”.**
- **There are some new “self organizing mechanisms” one
might be able to interfere with.**
- **Yes, vagueness abounds on this slide...**



Given we're seasoned speakers...



- ... we know you'd like to see some more practical stuff ;-)
- Remember us mentioning that APN needed?
- On his way to HITB Daniel coded another small tool ;-)



© 2007

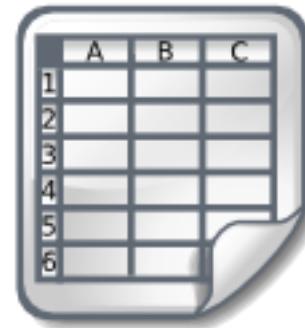
- Python script that brute forces the APN (Access Point Name) in GTPv1c.
- Uses gtp-create-PDPcontext-requests with the APN taken from a wordlist

```
greif@loki $ python apnbf.py -w apnlist 172.25.1.3
starting scan of 172.25.1.3
trying test.com
[...]
trying ernwtel.com
*** APN FOUND: ernwtel.com
done
```

DEMO

- **List of most used APNs in the Internet:**

- internet (12)
- INTERNET (10)
- Internet (10)
- wap (5)
- mms (5)
- airtelnet.es (4)
- online.telia.se (3)
- cmnet (3)



- **Some gtp speakers don't care about the APN at all...**
- **We might add it to the “HITB version of gtp-scan”
(→ see insinuator.net) ... as HITB is such a cool con ;-)**

Conclusions

- **We expect to see a number of attacks in 3G and 4G mobile telco networks in the next years, for some reasons**
 - *Walled (telco) gardens* are vanishing.
 - At the same time “terminals“ get more and more powerful.
 - In the future it’s all IP in those networks.
 - There’s a complex (IP based) protocol landscape.
And potentially ppl_outside_telcos are able to understand these prots.
As there are apparently people understanding Siemens PCS 7...
- **Theory ≠ reality**



There's never enough time...

THANK YOU...

...for yours!



Credits

- **To Simon for his helping hands.**
- **To our students Hendrik and Kai for digging restlessly through standards, preparing slides, etc. THANK YOU!**



Questions?

