Università degli Studi di Salerno

Progetto di Sicurezza

GPS Forensics
Un caso di studio: TomTom

Docente:

Prof. Alfredo De Santis

Candidati:

Armando Faggiano — *0522500043*

Ermanno Travaglino — *0522500042*

Settembre 2011

Sommario

Introduzione 3
1 GPS 4
1.1 Storia 4
1.2 Concetto di base 5
1.3 Calcolo della posizione 5
1.4 Struttura del sistema 8
2 L'azienda TomTom 10
2.1 Il dispositivo in possesso 10
3 Copia forense 12
3.1 Copia forense in ambiente Linux 13
3.1.1 Distribuzione CAINE 2.0 13
3.1.2 AIR 13
3.2 Metodologia di copia ripetibile 14
3.3 Copia forense della memoria flash 18
4 Analisi della copia forense 21
4.1 Busy Box 22
4.2 Punti di interesse & punto base 23
4.3 Label per l'indirizzo 27
4.4 Ultima posizione GPS rilevata 28
4.5 Triplog 31
4.6 File system proc & minor e major 41
5 Conclusioni e sviluppi futuri 44
6 Riferimenti 45

Introduzione

Le vendite di dispositivi portatili di navigazione sono cresciute molto nel tempo. Negli ultimi anni sono stati venduti, in tutto il mondo, milioni di dispositivi GPS come il TomTom oppure il Garmin. Questi dispositivi possono essere una buona sorgente di prove. Con l'ingresso sul mercato di dispositivi ibridi, adesso i dispositivi GPS contengono molte più informazioni sulla navigazione e potrebbero contenere dati comunemente presenti nei telefoni cellulari come audio, video e file di testo come documenti Word oppure PDF.

Le forze dell'ordine hanno assistito ad un drammatico incremento nell'uso dei dispositivi GPS come uno strumento utilizzato in un crimine oppure come un "dispositivo testimone" che, autonomamente, colleziona e registra dati sulla posizione mentre il crimine è in corso. TomTom e Garmin sono di gran lunga i dispositivi più popolari esaminati dalle forze dell'ordine. Il focus di questo lavoro sarà sui dispositivi TomTom (nello specifico il modello Start) e si vedrà che alcuni aspetti, di fondamentale importanza per un'analisi forense esaustiva, sono tutt'oggi un'incognita. In ogni caso il processo generale può essere esteso ad altri tipi di dispositivi.

Inizialmente si percorrerà la storia del sistema GPS, descrivendone la struttura e verrà mostrato come avviene il calcolo della posizione da parte di un ricevitore GPS. Sarà poi data una panoramica sull'azienda TomTom focalizzando l'attenzione sul modello preso in analisi.
Verrà illustrata una metodologia per compiere una copia forense ripetibile della memoria interna del dispositivo, descrivendo la distribuzione ed i tools utilizzati.
Successivamente verrà effettuata l'analisi dei dati contenuti nel dispositivo; tale procedimento si divide in tre fasi:

- La prima fase riguarda l'analisi del punto base e dei punti di interesse;
- La seconda fase descrive una metodologia per l'individuazione dell'ultima posizione GPS rilevata dal dispositivo;
- L'ultima fase contiene l'analisi dei viaggi effettuati con il dispositivo (file Triplog). Inizialmente è stata effettuata un'analisi che non alterasse il contenuto del dispositivo, facendo emergere solo la natura di questo file (cifrato). Sono stati quindi installati programmi addizionali per accedere alla root del dispositivo e tentare vari tipi di attacco a questo file, per comprenderne la struttura e il contenuto. Va precisato che finora non è stata trovata nessuna tecnica per decifrarlo.

1 GPS

Il **Global Positioning System** (abbreviato in **GPS**, a sua volta abbreviazione di **NAVSTAR GPS**, acronimo di **NAVigation Satellite Time And Ranging Global Positioning System**), è un sistema di posizionamento su base satellitare, a copertura globale e continua, gestito dal Dipartimento della Difesa degli Stati Uniti.

1.1 Storia

Il GPS è stato creato in sostituzione del precedente sistema, il Transit.

Transit è il nome di una flotta di satelliti statunitensi (chiamati anche Oscar o Nova) costruiti dall'U.S. Navy (marina militare degli Stati Uniti) per consentire a navi e sommergibili di determinare la propria posizione in mare in qualsiasi condizione meteorologica. Il sistema Transit fu reso obsoleto dall'introduzione del *Global Positioning System* e cessò il servizio nel 1996. Da allora i satelliti sono mantenuti in uso come 'mailbox' spaziali e per il Navy's Ionospheric Monitoring System.

Nel 1991 gli Stati Uniti d'America aprirono al mondo il servizio con il nome **SPS** (*Standard Positioning System*), con specifiche differenziate da quello militare denominato **PPS** (*Precision Positioning System*). In pratica veniva inserita la cosiddetta *Selective Availability* che introduceva errori intenzionali nei segnali satellitari allo scopo di ridurre l'accuratezza della rilevazione, consentendo precisioni solo nell'ordine di 100-150 metri. Tale degradazione del segnale è stata disabilitata dal mese di maggio del 2000, grazie a un decreto del Presidente degli Stati Uniti Bill Clinton, mettendo così a disposizione degli usi civili la precisione attuale di circa 10-20 metri. Nei modelli per uso civile devono essere presenti alcune limitazioni: massimo 18 chilometri per l'altitudine e 515 metri al secondo per la velocità, per impedirne il montaggio su missili.

L'Unione Europea ha in progetto il completamento di una propria rete di satelliti, il *Sistema di posizionamento Galileo*, per scopi civili. Questo progetto ha un'evidente valenza strategica in quanto la rete americana è proprietà dei soli Stati Uniti d'America ed è gestita da autorità militari, che, in particolari condizioni, potrebbero decidere discrezionalmente e unilateralmente di ridurre la precisione o bloccare selettivamente l'accesso al sistema: la condivisione dell'investimento e della proprietà da parte degli stati utilizzatori garantisce continuità, accessibilità e interoperabilità del servizio europeo. Nel dicembre del 2005 è stato lanciato il primo satellite denominato "GIOVE-A" dal nome attribuito al programma del sistema **Galileo**.

1.2 Concetto di base

Un ricevitore GPS calcola la propria posizione temporizzando con precisione i segnali inviati dai satelliti GPS presenti in orbita terrestre.

Ciascun satellite emette continuamente messaggi su due canali, L1 (frequenza 1.57542 GHz), l'unico fornito al servizio SPS (per uso civile), e L2 (frequenza 1.2276), usato come correzione agli errori causati dalle caratteristiche dielettriche della ionosfera e della troposfera per il servizio PPS (per uso militare).
A questi messaggi viene effettuata la modulazione numerica di tipo binario (0,1). Essi contengono:

- Tempo di trasmissione
- Effemèridi (precise informazioni orbitali relative al satellite)
- Almanacco (stato di salute e orbite di tutti i satelliti)

Il ricevitore utilizza questi messaggi per determinare il tempo di transito di ogni messaggio e calcola la distanza di ogni satellite. Queste distanze, conoscendo la precisa posizione dei satelliti, sono utilizzate insieme alla trilaterazione per calcolare la posizione del ricevitore. La posizione calcolata viene visualizzata su una mappa in movimento oppure in termini di latitudine e longitudine; possono essere aggiunte altre informazioni, come ad esempio l'altitudine. Molti dispositivi GPS mostrano le informazioni derivate, come direzione e velocità, calcolate dai cambiamenti di posizione.

Tre satelliti potrebbero sembrare sufficienti a determinare la posizione poiché lo spazio ha tre dimensioni e può essere assunta una posizione vicina alla superficie terrestre. Tuttavia, un minimo errore di tempo moltiplicato per la velocità della luce (velocità a cui si propagano i segnali satellitari) si traduce in un errore di posizione molto grande e quindi i ricevitori utilizzano quattro o più satelliti per determinare la propria posizione.

1.3 Calcolo della posizione

I messaggi, come detto nella sezione precedente, sono modulati con dei codici detti pseudo random noise (PRN), segnali noti e periodici. Tali segnali sono diversi per ogni satellite ed hanno l'importante proprietà di essere ortogonali tra loro permettendo la trasmissione simultanea di tutti i satelliti sulle stesse portanti L1 ed L2 senza problemi di interferenze.

Nel sistema GPS è fondamentale la misura delle distanze satellite-ricevitore; la misura di tali distanze avviene calcolando l'intervallo di tempo che impiega il segnale per arrivare dal satellite al ricevitore (tempo di propagazione).

Utilizzando i messaggi ricevuti da un minimo di tre satelliti visibili, un ricevitore GPS è in grado di determinare i tempi d'invio e poi le posizioni dei satelliti. Le componenti relative alla posizione, x, y e z, e il tempo d'invio t sono denotati da $[x_i, y_i, z_i, t_i]$, dove il pedice i rappresenta il numero del satellite e assume valore 1, 2, 3 o 4. Il messaggio generato dal satellite in un istante t_i arriva al ricevitore in un istante t_r, il tempo di propagazione è dato da $\Delta t = t_r - t_i$. Assumendo che il messaggio viaggi alla velocità della luce c, la pseudo-distanza è data da $\Delta t\, c$.

Non è detto che gli orologi di satellite e ricevitore siano sincronizzati tra loro, quindi la distanza determinata dalla soluzione delle equazioni non coincide con quella reale ed è detta pseudo-distanza. Indichiamo con b l'errore dell'orologio del ricevitore. Il ricevitore ha quattro incognite, le tre componenti della posizione del ricevitore GPS e l'errore dell'orologio $[x, y, z, b]$. Le equazioni delle superfici delle sfere sono date da:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = ([t_r - t_i + b]c)^2$$

La posizione del satellite e la pseudo-distanza definiscono una sfera, centrata sul satellite, con raggio pari alla pseudo-distanza. La posizione del ricevitore è, in un qualche punto, sulla superficie di questa sfera e quindi, con tre satelliti, la posizione del ricevitore GPS è pari o vicino all'intersezione delle superfici delle tre sfere. L'errore relativo allo sfasamento temporale degli orologi è corretto dal ricevitore ascoltando anche un quarto satellite. Se l'intersezione della quarta sfera non corrisponde a quella delle altre tre, il ricevitore percepisce che c'è un errore. Non è geometricamente possibile che le quattro sfere si intersechino nello stesso punto se l'orologio non è preciso. Quando questo non accade, il ricevitore dubita che l'orologio non sia sincronizzato, quindi esegue una piccola routine per correggere l'orologio, finché tutte e quattro le linee di posizione si intersecano nello stesso punto. Questo processo avviene ad ogni inizializzazione del ricevitore.

Se le superfici di due sfere s'intersecano in più punti, allora s'intersecano in una circonferenza. La condizione di trilaterazione mostra questo matematicamente. Come si vede nella figura seguente, due superfici sferiche s'intersecano in una circonferenza con diametro la distanza dei due punti.

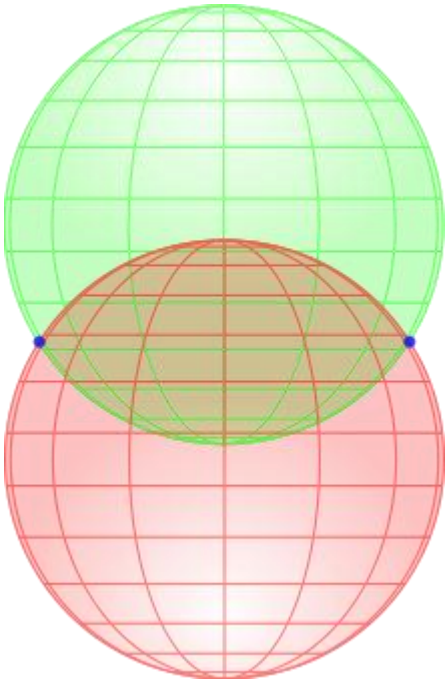


Figura 1

L'intersezione di una terza superficie sferica con la circonferenza costruita a partire dalle prime due sfere è mostrata nella figura 2:

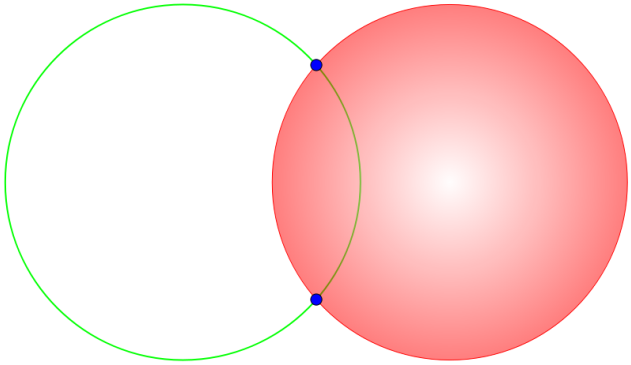


Figura 2

Nella maggior parte dei casi d'interesse pratico, questo significa che s'intersecano in due punti [1].

La figura seguente dà una visione realistica della spiegazione fornita in precedenza. A partire dai satelliti $x_{1,}x_2\ e\ x_3$, si tracciano le sfere con raggio rispettivamente $R_{1,}R_2\ e\ R_3$, e si ottengono due intersezioni che sono contrassegnate da due punti.

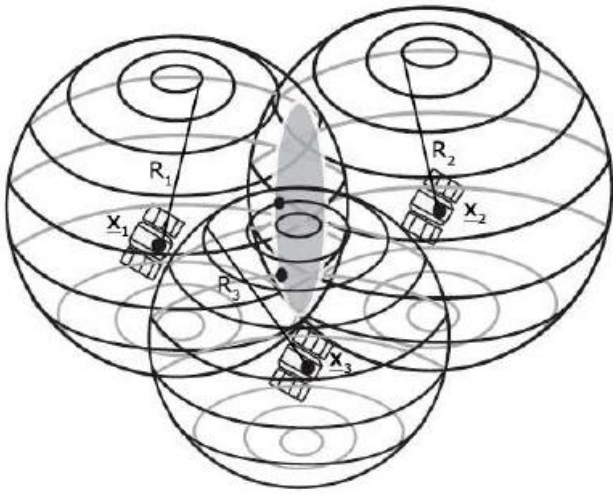


Figura 3

Per le automobili o altri veicoli vicini alla Terra, la posizione corretta del ricevitore GPS è l'intersezione più vicina alla superficie terrestre. Per i veicoli nello spazio la posizione corretta potrebbe essere l'intersezione più lontana dalla Terra.

1.4 Struttura del sistema

Il sistema GPS è composto da tre segmenti principali: segmento spaziale, segmento di controllo e segmento utente. L'aviazione degli Stati Uniti d'America (U.S. Air Force) gestisce sia il segmento spaziale sia il segmento di controllo. I satelliti GPS inviano in broadcast i segnali dallo spazio e ogni ricevitore GPS li utilizza per calcolare la posizione nello spazio tridimensionale (latitudine, longitudine e altitudine) e l'orario corrente.

Il segmento spaziale è composto da 32 satelliti in orbita MEO[1] disposti su 6 differenti piani orbitali inclinati di 55° rispetto al piano equatoriale e di 60° tra di loro.

Ciascun piano orbitale ha almeno 4 satelliti, e i piani sono disposti in modo tale che ogni utilizzatore sulla terra possa ricevere i segnali di almeno 5 satelliti. Ogni satellite trasmette l'almanacco (parametri orbitali approssimati) dell'intera costellazione, ma esclusivamente le effemèridi relative a se stesso.

Il segmento di controllo è responsabile del monitoraggio dello stato e della salute del segmento spaziale ed è costituito da 2 stazioni di controllo principale (MCS), di cui una è di backup, da 6 stazioni di monitoraggio (MS) e 4 antenne (GA) dislocate in diversi punti della superficie terrestre.

Le stazioni di monitoraggio seguono, in modo passivo, i satelliti raccogliendo informazioni sulla loro posizione, i dati sono poi mandati alla stazione di controllo principale, che li utilizza per

[1] Regione dello spazio intorno alla Terra fra la Low Earth Orbit (2000 chilometri di altitudine) e la Geostationary Orbit

determinare le orbite dei satelliti e le correzioni da apportare agli orologi atomici montati a bordo degli stessi. Le informazioni sulle orbite e sulle correzioni degli orologi vengono poi, tramite le antenne, trasmesse ai satelliti per aggiornare il messaggio di navigazione. Questo è in seguito trasmesso insieme al segnale GPS agli utenti che lo utilizzeranno per la determinazione della posizione.

Il segmento utente è costituito da tutti ricevitori GPS che decodificano e processano il segnale trasmesso dai satelliti per determinare: posizione; velocità; e precise informazioni temporali. Siccome i ricevitori operano passivamente (non trasmettono alcun segnale) il segmento spaziale può fornire servizio ad un illimitato numero di utenze.

2 L’azienda TomTom

TomTom International BV è una società olandese, con sede ad Amsterdam, nata nel 1991 (anno in cui gli Stati Uniti hanno aperto al mondo il servizio di posizionamento) che produce sistemi di navigazione satellitare per automobili e motoveicoli. E' disponibile, oltre che sui dispositivi proprietari, anche come applicazione per smartphone (inizialmente era prodotto anche per piattaforme Windows Mobile, adesso solo per IPhone e IPad). È il principale fornitore di sistemi di navigazione in Europa.

Grazie a ricevitori GPS (integrati o esterni) ed all'utilizzo, per la quasi totalità delle zone coperte dal servizio, di mappe digitali Tele Atlas (Whereis per l'Australia e GeoSmart per la Nuova Zelanda) viene calcolato il percorso ideale da intraprendere verso una destinazione. Durante il tragitto una voce annuncia il percorso da intraprendere che viene mostrato in tempo reale sullo schermo.

TomTom offre una vasta gamma di dispositivi per la navigazione, alcuni di questi dispongono di Bluetooth e permettono, così, di utilizzare il dispositivo come vivavoce per telefoni cellulari.

E' inoltre possibile acquistare servizi ulteriori, come lo stato di congestione del traffico o la presenza nella zona di transito di probabili autovelox [2].

2.1 Il dispositivo in possesso

Il dispositivo analizzato è il TomTom Start con le mappe dell'Italia. Tale dispositivo possiede una memoria centrale e non necessita di scheda di memoria SD esterna.

Il nocciolo è comunque l'analisi della memoria "interna" perché la memoria SD esterna, essendo estraibile dal dispositivo, può essere trattata come una qualunque memoria di massa secondo le regole della Computer Forensics.

Questo dispositivo, inoltre, permette di memorizzare contatti, indirizzi, itinerari costruiti, il punto base (Home location) e i punti di interesse (POI).

Si avvale al suo interno di un computer con processore della serie ARM 920 (ARM926EJ in versione5l) creato da Samsung, che gestisce, attraverso un sistema Linux, il funzionamento del software presente, nel nostro caso, nella memoria interna.

Nel file *init* sono contenute le informazioni di avvio del dispositivo, che analizzeremo in seguito, verrà ricercato il file *ttn* nella memoria interna e i dati della mappa in uso. Esso poi verrà trasferito nella memoria RAM, da 32 MB, e si avvierà il programma di navigazione.

La struttura generale di un navigatore TomTom è mostrata nella figura seguente [3].

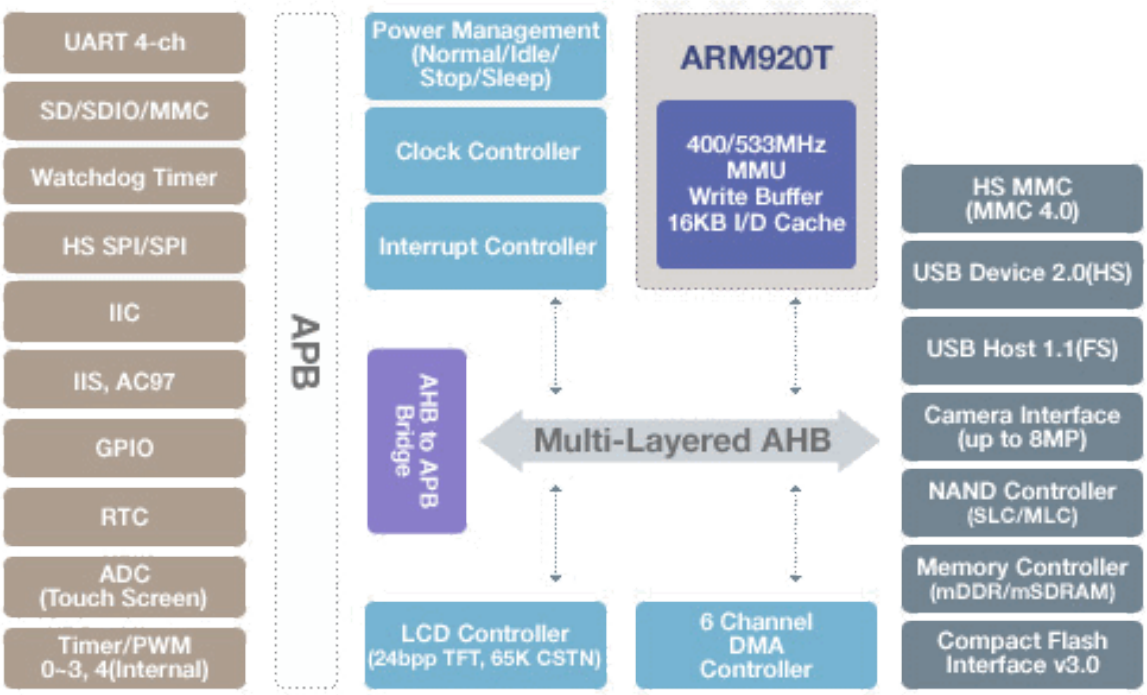


Figura 4

3 Copia forense

Quando si esegue un'indagine forense, per prima cosa bisogna compiere una copia fisica dei dati. Per immagine forense s'intende il risultato di una particolare procedura di copia, detta "bit a bit", che va a leggere la superficie fisica del supporto di memoria, un bit dopo l'altro, e ne produce un "clone", cioè una copia identica, su un supporto di destinazione, il cui contenuto sarà sottoposto ad analisi.
Quando possibile, infatti, l'analisi forense non è effettuata sul dispositivo originale, ma su un suo clone, o immagine forense (la cosiddetta "bitstream image"), allo scopo di preservare l'integrità del reperto originale per eventuali analisi future.

Ad oggi non esiste una legislazione chiara che regolamenti la procedura di acquisizione dei dati. Esistono in tal senso molteplici linee guida (best practices) che illustrano un possibile svolgimento [4]:

- L'orologio di sistema deve essere sincronizzato con un server NTP[2] noto (relativo alla macchina su cui verrà effettuata la copia dei dati presenti nel dispositivo GPS). Questo viene effettuato per avere riferibilità ad un momento fissato del tempo;

- Il dispositivo GPS deve essere isolato dalle onde radio per evitare aggiornamenti di file all'interno dello stesso, come l'ultima posizione GPS rilevata (può essere effettuato con una gabbia di Faraday[3] oppure con un jammer[4]);

- Il processo di acquisizione dei dati non deve alterare il contenuto del dispositivo GPS (montandolo in sola lettura);

- Si deve verificare la conformità della copia effettuata con i dati originali (confrontando il risultato ottenuto dalle funzioni hash[5]).

Uno degli obiettivi della sperimentazione è la ricerca di una procedura ***ripetibile*** per eseguire una copia forense della memoria interna del dispositivo, e cioè che consenta di ottenere, dal medesimo dispositivo, un'identica immagine forense nel tempo, per eventuali analisi

[2] Network Time Protocol: è un protocollo per sincronizzare gli orologi dei computer all'interno di una rete a commutazione di pacchetto, quindi con tempi di latenza variabili ed inaffidabili. L'NTP è un protocollo client-server appartenente al livello applicativo e in ascolto sulla porta 123.

[3] Con gabbia di Faraday si intende qualunque sistema costituito da un contenitore in materiale elettricamente conduttore (o conduttore cavo) in grado di isolare l'ambiente interno da un qualunque campo elettrostatico presente al suo esterno, per quanto intenso questo possa essere.

[4] Il jammer è un disturbatore di frequenze, utilizzato per impedire ai telefoni cellulari, ai dispositivi GPS e ai dispositivi affini, di ricevere o trasmettere segnali.

[5] Una funzione hash è in grado di trasformare un dato in input (di misura variabile) in un dato di output di misura fissata (hash). Le funzioni hash utilizzate sono resistenti alle collisioni: è computazionalmente intrattabile la ricerca di una coppia distinta di input che dia come output lo stesso hash.

successive o di controparte.

3.1 Copia forense in ambiente Linux

La copia forense è stata eseguita in ambiente Linux su un PC notebook (DELL Studio 1558) con sistema operativo Caine 2.0.
Prima di illustrare la metodologia di copia forense utilizzata, sarà fornita una breve panoramica su questa distribuzione Linux e sui tools che sono stati utilizzati.

3.1.1 Distribuzione CAINE 2.0

CAINE (Computer Aided Investigative Environment) è una distribuzione live orientata alla Computer Forensics (informatica forense), ideata da Giancarlo Giustini all'interno di un progetto di Digital Forensics del Centro di Ricerca Interdipartimentale per la Sicurezza (CRIS) dell'Università di Modena e Reggio Emilia.
CAINE 2.0 è basata su Ubuntu Linux 10.04 LTS, nella sua versione più aggiornata. La distribuzione originale è stata modificata per venire incontro agli standard forensi di affidabilità e sicurezza. Per questo sono stati installati solo i software essenziali per condurre al meglio l'indagine e sono state modificate alcune caratteristiche base del sistema operativo.

3.1.2 AIR

AIR (Automated Image and Restore) è una GUI front-end per *dd*[6]/*dc3dd*[7] progettato per creare facilmente immagini digitali forensi. Questo tool permette:

- Il rilevamento automatico di dischi IDE o SCSI, CD-ROM e unità a nastro;
- La scelta di utilizzare dd o dc3dd;
- La verifica dell'identicità dell'immagine originale e della copia tramite il calcolo e il confronto dell'hash MD5 o SHA1/256/384/512;
- La compressione e la decompressione della copia tramite gzip/bzip2;
- La creazione dell'immagine tramite una connessione di rete;
- Il wiping[8] di partizioni e dischi;
- La suddivisione dell'immagine in più parti (splitting);
- Il logging dettagliato di tutte le operazioni effettuate;

[6] E' un comando dei sistemi operativi Unix e Unix-like che copia i dati in blocchi, opzionalmente effettuando conversioni
[7] E' una patch del comando *dd* con delle funzionalità aggiuntive per la computer forensics. Per approfondimenti: http://www.forensicswiki.org/wiki/Dc3dd
[8] Processo di sovrascrittura del contenuto di un file in modo da impedirne il recupero.

3.2 Metodologia di copia ripetibile

Il primo passo da compiere, prima di eseguire la copia, è quello di collegare il dispositivo al PC, tramite il cavo miniUSB.
Fatto ciò, bisogna accendere necessariamente il dispositivo perché altrimenti la memoria interna non sarebbe vista e riconosciuta dal sistema operativo. E' stato verificato che tale operazione non modifica alcun file.

Dopo l'accensione, il TomTom mostrerà una schermata per confermare, o meno, la connessione del dispositivo al computer.

Figura 5

Selezionando SI viene avviata la procedura di connessione del dispositivo al computer, quindi vengono installati i driver USB.

Figura 6

Una volta terminata questa procedura, sul TomTom appare una schermata di avvenuta connessione.

Figura 7

Dopo aver eseguito la procedura di connessione al PC, si esegue la copia forense della memoria interna del TomTom, tramite il tool AIR.
Viene selezionato il device sorgente sulla parte sinistra dell'interfaccia, mentre sulla destra il device di destinazione.
Si è scelto di non eseguire alcuna compressione dell'immagine creata, perché comunque la dimensione della memoria interna è inferiore ad 1 GB.
Viene selezionato il tipo di hash da utilizzare per verificare l'identità tra i dati letti e la copia risultante.
Si è utilizzato il comando *dc3dd* al posto di *dd*. Questo comando permette l'hashing on the fly, cioè il calcolo del codice hash durante il processo di copia.
Si è scelto di non splittare l'immagine e di non cifrare il file.
Infine si è specificata l'opzione *noerror* al parametro *conv*, che fa proseguire l'operazione di creazione dell'immagine anche in caso di errori in fase di lettura.
Prima di premere il pulsante *Start* e dare inizio al processo di copia, si è aperta la finestra di stato, cliccando sul bottone *Show Status Window,* che mostra l'andamento dell'operazione.
Si è selezionato il dispositivo, montato in */dev/sdc1*. Si è selezionato il device di destinazione, in questo caso un file: */home/caine/Desktop/TomTomStart*.
Si è selezionato il block size sia per la sorgente sia per la destinazione, pari a 8192.
Sono stati scelti come algoritmi di hash sia MD5 sia SHA-1.
La figura seguente mostra la configurazione selezionata (interfaccia del tool AIR).

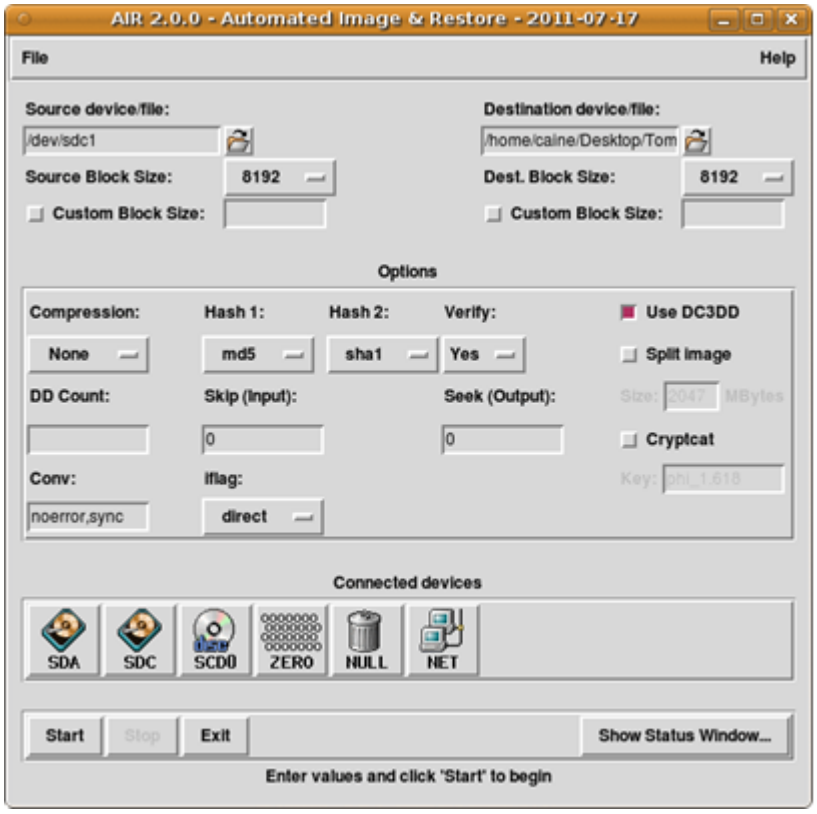


Figura 8

Dopo aver cliccando sul bottone *Show Status Window* è mostrato l'avanzamento della procedura di copia, tramite un session log e un'analisi del throughput.

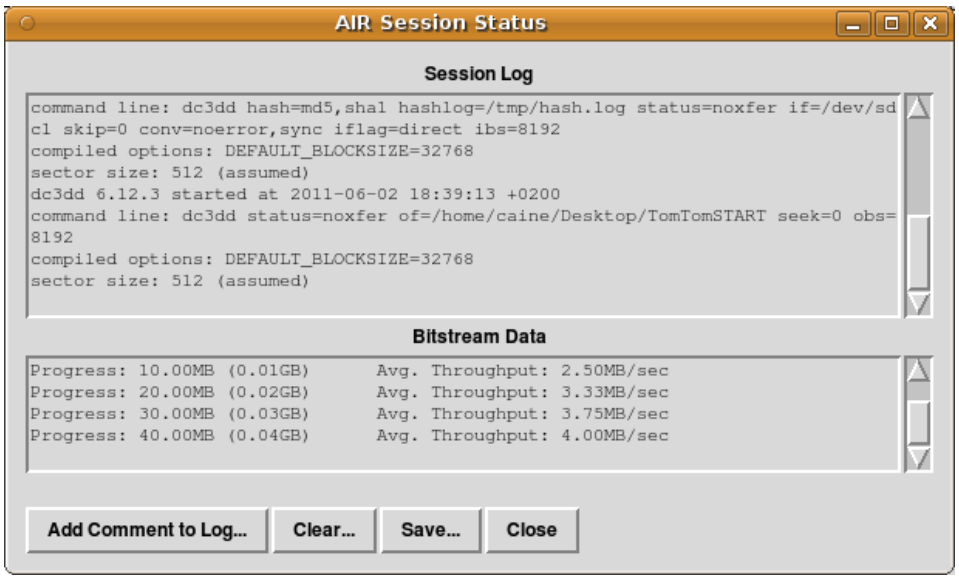


Figura 9

Una volta terminata la copia, AIR mostra l'hash MD5 e SHA-1 sia dell'originale sia della copia, ne fa il matching e comunica all'utente il risultato.

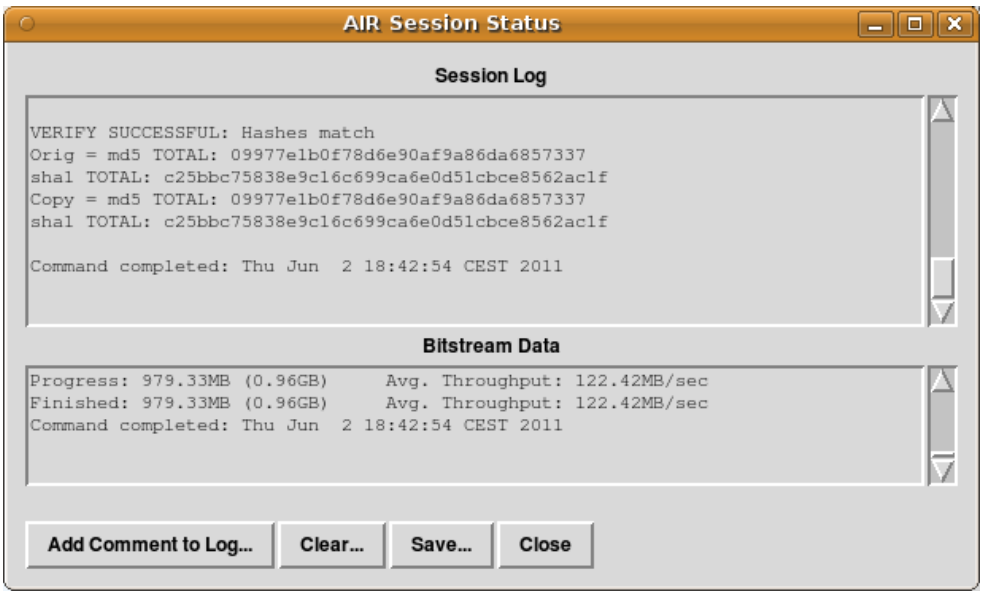


Figura 10

Eseguendo più copie forensi dello stesso dispositivo, in modo sequenziale e senza compiere alcun cambiamento nei dati, si è notato che gli hash di tali copie erano diversi fra loro. Alcuni file sul dispositivo cambiano quando lo stesso viene disconnesso dal PC. Infatti, quando viene rimosso il cavo USB, la schermata del dispositivo cambia, mostrando una barra di caricamento che indica l'aggiornamento di alcuni file (passaggio alla modalità utente).
I file che vengono modificati sono:

- *CurrentMap.dat*: contiene la mappa in uso corrente.
- *ttgo.bif*: contiene le informazioni relative al dispositivo tra cui modello, numero di serie, lingua, mappa corrente, base corrente, voce ecc..
- */itn/temporary.iti*: contiene gli itinerari non memorizzati con un nome file (itinerari di default).
- *settings.dat*: contiene il nome ed il mac address del telefono eventualmente collegato, la configurazione del provider, i dati del telefono e dell'utente se immessi (solo per modelli TomTom GO).
- *UserPatch.dat*: contiene le eventuali modifiche effettuate dall'utente su determinati punti stradali.

Per i primi due file viene modificato soltanto il timestamp, mentre per gli altri tre file viene modificato anche il contenuto.
Questi cambiamenti hanno posto come obiettivo quello di trovare una metodologia per

compiere una copia forense ripetibile della memoria del dispositivo.
Dopo aver compiuto la copia forense seguendo i passi descritti in precedenza, come la connessione del dispositivo, il montaggio del file system INTERNAL e l'utilizzo di AIR, si devono effettuare le seguenti operazioni:

- **Fare l'unmount del file system INTERNAL**
- **Spegnere il dispositivo, tramite l'apposito pulsante fisico**
- **Scollegare il dispositivo dal PC**

Questi ultimi tre passi sono **cruciali** per evitare modifiche dei file sopraelencati, sia nei dati sia nei metadati (timestamps), infatti se anziché eseguire l'unmount del file system, si utilizza l'opzione "EJECT" fornita da CAINE, il sistema operativo smonterà il file system e il dispositivo passerà in modalità utente, quindi seguirà una fase di start-up del dispositivo in cui verrà modificato il contenuto della memoria interna.
Si propone un video dimostrativo con il quale si intende chiarire i passi descritti dall'inizio di questo capitolo, al fine di rendere ripetibile la copia della memoria del dispositivo analizzato.

3.3 Copia forense della memoria flash

Il dispositivo in esame possiede un supporto di memoria flash interno. E' possibile accedere alla memoria flash NOR utilizzando il sottosistema Linux chiamato Memory Technology Device (MTD), attraverso il device */dev/mtd0*. Utilizzando il comando *dd* è stato possibile copiare il contenuto della memoria flash nella memoria interna. Per permette un facile esame dei file memorizzati nel file system NGFFS[9], che è presente sul device, questi file sono stati copiati anche separatamente. Il contenuto di questa memoria è mostrato nell'immagine sottostante (directory in rosso e file in blu).

[9] Next Generation Flash File System: è un driver del file system per kernel Linux. Ulteriori informazioni sono disponibili al sito di riferimento http://ngffs.sourceforge.net/

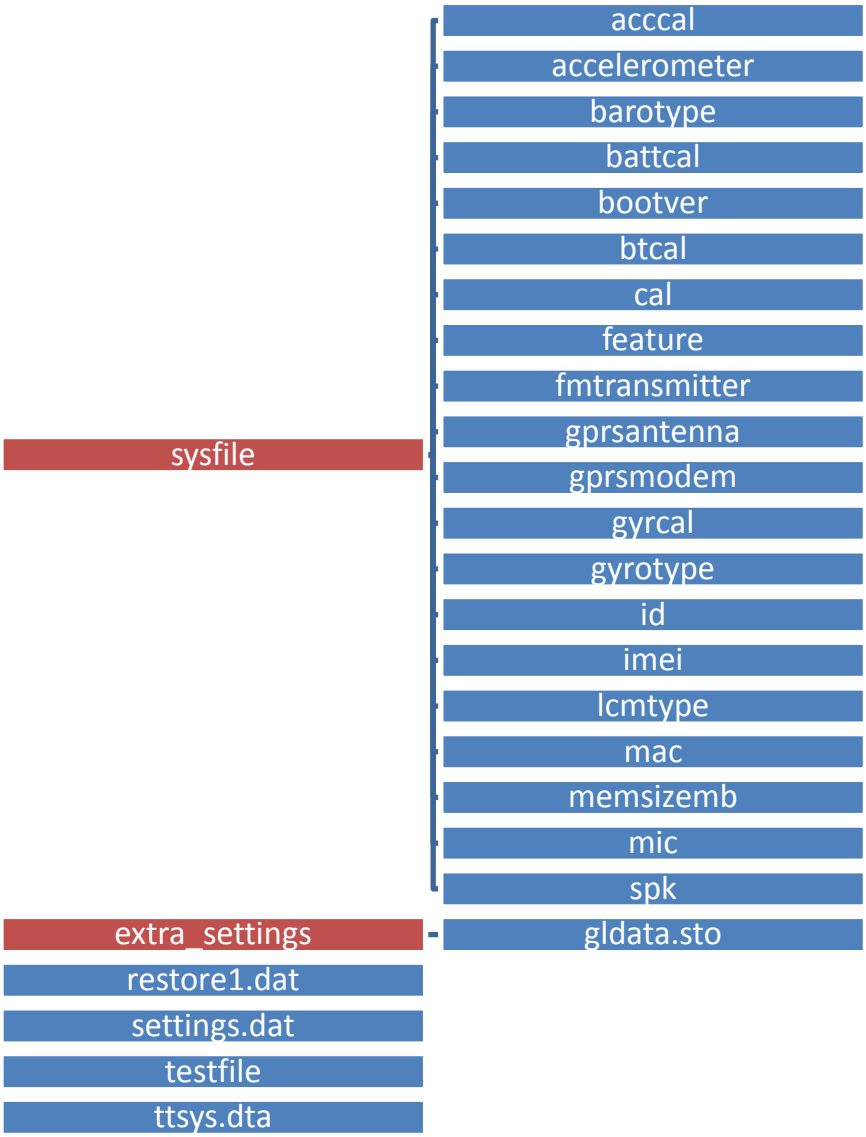


Figura 11

Lo scopo di alcuni file sulla partizione NGFFS è noto. Il file "*bootver*" memorizza la versione del bootloader del dispositivo, "*cal*" memorizza i dati di calibrazione del touch screen, "*id*" memorizza l'id del dispositivo o il numero di serie, "*mac*" memorizza l'indirizzo mac del bluetooth. Anche se non è stato verificato, si pensa che il file "*imei*" contenga l'International Mobile Equipment Identity (IMEI) di un dispositivo con funzionalità GSM e che il file "*btcal*" contenga i dati di calibrazione del Bluetooth [8]. Su alcuni dispositivi, come quello in questione, i dati memorizzati nella directory "*sysfile*" possono essere visualizzati sul display. Occorre tenere premuto il pulsante di accensione, quando il dispositivo è spento, finché non appare una schermata simile a quella che segue.


TomTom START (TM) (c) TomTom 2003-2010 Build -N/A-
Hardware type: TomTom START
SubID (feature) 0 1: 00 (0000000000000000)
Bootloader version: S5.5274 clist#540743 RVCT
Checksum: 1DF5
Memory: 32 MB, Type:MDDR
CPU ID: 32450003
Clocks: 264.0 132.0 66.0
GPS type: GL2 BCM4750 FFFF
TFT Type: Samsung LMS350GF20 16
Internal FLASH capacity: 1003520 KB
MMC NAND version (PRV): 36 48.0 MHz 4 8 Samsung
Startup mode: Sleep
Device ID: XC6250K02432
Touchscreen calib. data: 223 827 284 714 3300mV
Sector protection: Yes
Battery boot voltage: 4207 mV 0x036F 0x0300
RTC: 18:23:42 11/09/2011

Figura 12

4 Analisi della copia forense

Dopo aver effettuato la copia del dispositivo TomTom si è passati all'analisi dei dati contenuti.

A seconda del modello è possibile acquisire diversi tipi d'informazione, compresi i dati che si trovano comunemente sui telefoni cellulari. Tutti i modelli hanno una memoria interna, quindi consentono ad un utente di memorizzare immagini, file audio e video, documenti, etc.

Dei tipi di file specifici per i dispositivi TomTom possono essere:

- *Locations*: tutti i modelli di TomTom hanno un file delle location che contiene l'indirizzo Home, una lista delle destinazioni recenti e talvolta l'ultimo viaggio effettuato.
- *Informazioni sul dispositivo*: numero di serie, numero del modello, versione del software di navigazione e altre informazioni generali sul dispositivo.
- *Chiamate ricevute*
- *Chiamate effettuate*
- *SMS ricevuti*
- *SMS inviati*
- *Contatti*
- *Bluetooth & MAC ID*
- *Informazioni utente*

Benché l'esame dei dispositivi TomTom può fornire tutte queste informazioni, si è focalizzata l'attenzione sull'analisi dei location record. I dispositivi memorizzano i record delle destinazioni recenti, i preferiti, il punto base e l'inizio e la fine dell'ultimo viaggio calcolato. Tutte queste informazioni sono contenute nel file *MapSettings.cfg*. La locazione di questo file dipende dalla versione del dispositivo e dalla mappa utilizzata, più precisamente sarà */Italia-Map/Italia-Map.cfg* per versioni di TomTom precedenti alla sei, che utilizzano una mappa chiamata Italia-Map, sarà invece in */Italia/MapSettings.cfg* per le versioni successive che utilizzano la mappa italiana.

I file interessanti da un punto di vista forense sono:

- **.cfg*: contiene le location memorizzate.
- *ttgo.bif*: contiene informazioni generali sul dispositivo, numero del modello, numero seriale, password dell'utente (cifrata).
- *settings.dat*: contiene il nome e l'indirizzo MAC di un telefono cellulare associato (se è stato collegato), i dati sulle impostazioni wireless e sul provider (se è stato configurato) e

informazioni sul numero telefonico di casa e sul proprietario del dispositivo.

- */contacts*[10]*/called.txt*: file che contiene i numeri chiamati dal cellulare associato al TomTom.
- */contacts/callers.txt*: file che contiene i numeri che hanno chiamato il telefono cellulare associato al TomTom.
- */contacts/contacts.txt*: file che contiene i dettagli dei numeri nella rubrica del telefono cellulare associato al TomTom.
- */contacts/inbox.txt*: contiene gli sms in arrivo.
- */contacts/outbox.txt*: contiene gli sms in uscita.
- */itn/*: questa directory contiene gli itinerari, se sono stati attivati (*temporary.iti* è l'itinerario correntemente attivo).

La letteratura in materia è davvero povera e quindi ci sono poche informazioni sull'analisi dei file *.cfg* di TomTom.
Esiste una comunità attiva open-source (OpenTom) che si occupa di sviluppare software addizionali o alternativi per la piattaforma Linux-embedded presente sui dispositivi TomTom [5]. In questo caso l'analisi è stata focalizzata sul bootloader, poiché quest'ultimo permette la riscrittura di software alternativo per essere eseguito. Naturalmente tali operazioni modificano il contenuto della memoria interna del dispositivo.

Ci sono due pubblicazioni che descrivono la struttura dei file *.cfg*: la prima, di Paul Weall[11], presentata al First Forensic Forum, e la seconda di Simon Siezenga.

4.1 Busy Box

I dispositivi TomTom possiedono una BusyBox. BusyBox è un software libero, rilasciato sotto la GNU General Public License, che combina diverse applicazioni standard di Unix in un piccolo eseguibile. BusyBox può fornire la maggior parte delle utility menzionate nel Single Unix Specification ed in aggiunta altre che un utente si aspetterebbe di trovare su un sistema GNU/Linux. Viene definito come "il coltellino svizzero del Linux embedded".

La versione presente sul dispositivo in esame è la 1.00, che non possiede comandi utili per un'analisi forense, come ad esempio `dd`.

[10] I file in /contacts/ non sono presenti sul dispositivo TomTom Start perché non possiede funzionalità di connessione a cellulari.
[11] Insieme a Andy Sayers hanno creato il software di Forensic Navigation Tomtology.

Per superare questa limitazione si è cercato di compilare la BusyBox per il processore ARM presente nel dispositivo.

Seguendo la guida ufficiale[12] dal sito TomTom, per prima cosa si è scaricato il compilatore gcc per il processore ARM. Poi si sono scaricati i sorgenti di BusyBox e, tramite uno script shell, è stata fatta la compilazione dei file. Dopo la compilazione, si è testato l'esito della procedura tramite l'esecuzione di un semplice comando quale `./BusyBox –help`. Purtroppo questo comando non ha sortito gli effetti desiderati, in quanto, non ha prodotto nessun output. Sono state effettuate altre prove, modificando alcuni parametri in ingresso al compilatore, ma l'esito non è variato.

Per avere una BusyBox funzionante, si è scaricata una versione precompilata per il processore ARM. E' stato trovato un repository di applicazioni precompilate per l'architettura ARM[13]. Da questo è stato scaricato il binario di BusyBox, è stato copiato nella directory */bin* della root del dispositivo e si è testato il corretto funzionamento dell'applicazione tramite uno script shell.

Grazie a questa BusyBox è stata possibile l'esecuzione di comandi utili durante la fase di copia forense della memoria flash e durante tutta la fase di analisi.

4.2 Punti di interesse & punto base

I file *.cfg* sono organizzati in record. Ogni record contiene una coppia di coordinate. La longitudine e la latitudine sono rappresentate in WGS84 datum. Facciamo una piccola digressione su questo punto. WGS84 è l'acronimo di World Geodetic System 1984 e definisce il sistema come geodetico[14], mondiale, riferito al 1984. Esso costituisce un modello matematico della Terra da un punto di vista geometrico, geodetico e gravitazionale, costruito sulla base delle conoscenze del 1984.

Un *datum* è un sistema geodetico di riferimento da cui le misure sono effettuate.
I *datum* della geodesia classica, possono essere definiti locali o regionali, approssimano bene il geoide solo in un intorno del punto di emanazione (che è il punto di tangenza del geoide con l'ellissoide di proiezione), mentre il datum globale WGS84 utilizza lo standard EGM96 (modello

[12] Disponibile al sito http://www.tomtom.com/page.php?Page=gpl

[13] Link http://www.netwinder.org/allrpms.html

[14] Sistema particolare per indicare la posizione di un oggetto sulla superficie terrestre.

geo-potenziale della terra), che approssima il geoide nel suo complesso ed è valido per tutto il mondo. Dal 2000 è obbligatorio l'utilizzo del WGS84 come standard per la navigazione aerea.

Facciamo un'altra precisazione, su latitudine e longitudine, che sarà utile nel prosieguo dell'analisi.
Le coordinate geografiche, che ci permettono di risalire ad un punto esatto sulla superficie terrestre (ma non solo), sono una coppia di numeri che esprimono latitudine e longitudine.

La latitudine esprime la distanza angolare dall'equatore (l'equatore ha una latitudine di 0°). Il valore della latitudine aumenta man mano che ci avviciniamo ai poli , i quali hanno una latitudine di 90° (positiva per il polo Nord: +90°, negativa per il polo Sud: -90°). I paralleli, ovvero le linee parallele all'equatore che troviamo generalmente disegnate sulle cartine geografiche, sui mappamondi etc, non sono altro che punti aventi la stessa latitudine.

La longitudine, invece, esprime la distanza angolare dall'equatore al meridiano di Greenwich, il quale ha una longitudine di 0°. La longitudine aumenta da 0° a 180° muovendosi dal meridiano di Greenwich al meridiano opposto. Il valore di longitudine è positivo verso est e negativo verso ovest.
Ci sono due modi per esprimere questa coppia di coordinate:
solitamente latitudine e longitudine sono valori in base 60, ovvero utilizzano la notazione sessagesimale, ma possono anche esser espressi in base 10, ovvero in decimale.
La conversione dalle due basi è abbastanza semplice.

- Da sessagesimale a decimale
  Consideriamo il valore 82°33'24''.
  Prendiamo i gradi (82): rimarranno invariati.
  Prendiamo i minuti (33) e li dividiamo per 60 : 33/60 = 0,55.
  Prendiamo i secondi (24) e li dividiamo per 3600 : 24/3600 = 0,0066.
  Sommiamo le tre quantità e otteniamo il valore di gradi espresso in decimale : 82 + 0,55 + 0,006667 = 82,556667.
- Da decimale a sessagesimale
  Consideriamo il valore 82,556667.
  Prendiamo la parte intera (82) : questo è il valore dei gradi.
  Prendiamo la parte decimale (0,556667) e la moltiplichiamo per 60 : 0,556667 * 60 = 33,40002
  Di quest'ultimo valore prendiamo la parte intera (33) : questi sono i minuti
  Prendiamo la restante parte decimale del valore precedente (0,40002) e moltiplichiamola per 60 : 0,396 * 60 = 24,0012 : questo è il valore dei secondi.

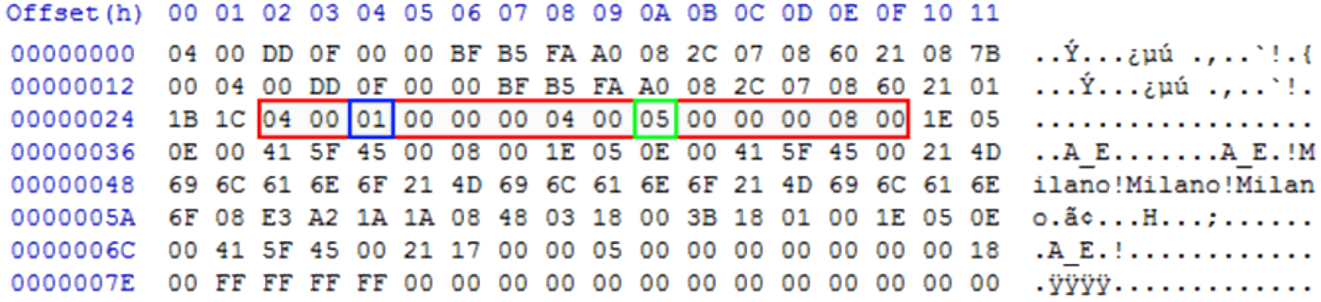
Ovviamente la precisione dipende dal numero di decimali che ci trasciniamo dietro durante le operazioni.

Oltre alle coppia di coordinate ci sono altre informazioni presenti nel record: una label che rappresenta l'indirizzo e altri due insiemi di coordinate.
L'ipotesi di Weall, menzionato sopra, è che il primo set di coordinate rappresenti la posizione attuale mentre gli altri tre rappresentino ulteriori caratteristiche, come ad esempio una strada vicina oppure dei raccordi stradali. Anche Siezenga ha analizzato la struttura dei file *.cfg*. Ha notato che il primo record nel file è l'Home location, se è stata inserita, e che gli ultimi due record riguardano l'inizio dell'ultima rotta calcolata e l'ultima destinazione inserita. Con queste informazioni si comprende meglio la struttura di un file *.cfg [6]*.

Il file *.cfg* si trova nella directory della mappa attiva. Questo file contiene, oltre alle informazioni citate in precedenza, anche l'ultima posizione GPS rilevata, illustrata in seguito. Il file contiene molti record come le destinazioni recenti e include gli indirizzi inseriti e poi salvati come preferiti, gli indirizzi parzialmente inseriti (per esempio, quelli in cui un utente ha iniziato a immettere un indirizzo tramite codice postale, ma annullato prima del completamento), e le ricerche per Point of Interest (POI). Queste posizioni sono utilizzate quando viene immesso un indirizzo per predire in modo intelligente dove l'utente potrebbe desiderare di andare. Ad esempio, quando s'inserisce un indirizzo di una determinata città, i nomi delle strade di quella città inseriti in precedenza nel file *.cfg* appariranno per primi nella lista.

All'interno del file *.cfg* sono memorizzati un numero variabile di location record.
Sarà focalizzata l'attenzione sul formato dei singoli record contenuti nel file *.cfg*, piuttosto che sulla struttura e il formato del file stesso.

Si è utilizzato un editor esadecimale per aprire questo file e per studiarne la struttura dei record. Questi record hanno lunghezza variabile.
Un location record si identifica all'intero del file *.cfg* dalla sequenza 04 00 XX 00 00 00 04 00 XX 00 00 00 08 00 come mostrato nella figura seguente:



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	
00000000	04	00	DD	0F	00	00	BF	B5	FA	A0	08	2C	07	08	60	21	08	7B	..Ý...¿µú .,..`!.{
00000012	00	04	00	DD	0F	00	00	BF	B5	FA	A0	08	2C	07	08	60	21	01	...Ý...¿µú .,..`!.
00000024	1B	1C	04	00	01	00	00	00	04	00	05	00	00	00	08	00	1E	05	.....
00000036	0E	00	41	5F	45	00	08	00	1E	05	0E	00	41	5F	45	00	21	4D	..A_E.....A_E.!M
00000048	69	6C	61	6E	6F	21	4D	69	6C	61	6E	6F	21	4D	69	6C	61	6E	ilano!Milano!Milan
0000005A	6F	08	E3	A2	1A	1A	08	48	03	18	00	3B	18	01	00	1E	05	0E	o.ã¢...H...;.....
0000006C	00	41	5F	45	00	21	17	00	00	05	00	00	00	00	00	00	00	18	.A_E.!.....
0000007E	00	FF	FF	FF	FF	00	00	00	00	00	00	00	00	00	00	00	00	00	.ÿÿÿÿ.....

Figura 13

Il byte in blu (XX) identifica la precisione con cui è stata inserita una location (ad esempio una location del centro città oppure un indirizzo con numero civico), mentre il byte in verde (YY) identifica il tipo di location.
Di seguito sono indicate le tabelle che specificano il significato dei due identificatori a seconda del valore.

Valori del byte XX (precisione della location)	
Byte XX	Precisione
01	Centro città
02	Incrocio specifico
03	Numero civico o edificio
04	Ovunque su una strada

Tabella 1

Valori del byte YY (tipo della location)	
Byte YY	Tipo del record
01	Inserito tramite mappa o codice postale
03	Preferito
04	Home location
05	Inserito tramite l'indirizzo
06	Inserito tramite POI
07	Inizio dell'ultima rotta calcolata

Tabella 2

Nella figura successiva si può vedere che dopo l'identificativo del location record c'è il primo set di coordinate espresso in formato WGS84 datum. Dopo il primo set c'è un separatore *08 00* e poi il secondo set di coordinate che rappresenta la strada più vicina alla location.
Le coordinate sono memorizzate come interi con segno, in formato little-endian, in 24 bit.
Per poter calcolare il valore corrispondente alle coordinate bisogna prima trasformare il numero in big-endian, convertilo in decimale e poi dividerlo per 100˙000. Compiendo quest'operazione si ottengono la **longitudine** e la **latitudine** della location espresse in decimale.

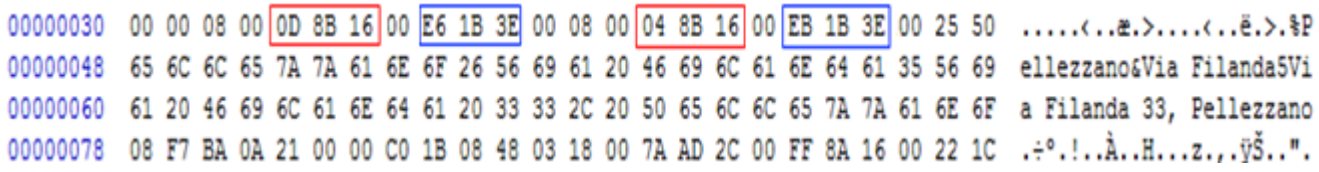


Figura 14

Calcolo del valore delle coordinate :

Longitudine 0D 8B 16 → 16 8B 0D → 1477389 →14,77389

Latitudine E6 1B 3E → 3E 1B E6 → 4070374 → 40,70374

Va precisato che, essendo le coordinate espresse in 24 bit, la precisione dei numeri interi dopo la virgola è di almeno 5 cifre. Questa precisione rientra nella tolleranza dell'errore del

dispositivo, infatti l'errore generato dalla sesta cifra decimale dopo la virgola corrisponde a poco più di 1 metro (1,11 m).

4.3 Label per l'indirizzo

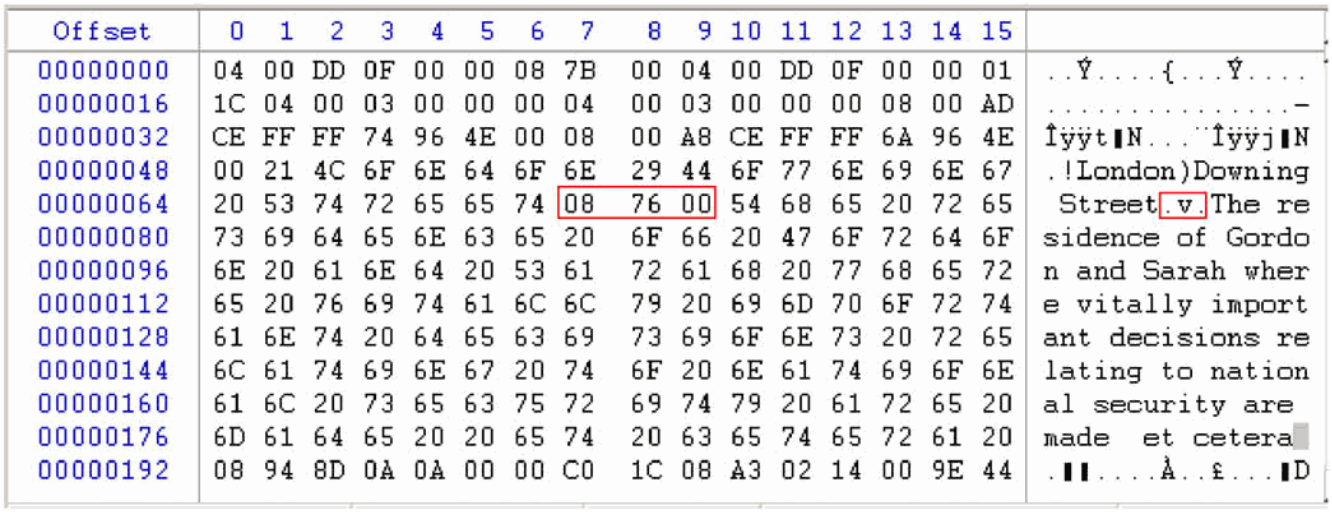
La label per l'indirizzo segue il secondo set di coordinate analizzate in precedenza. E' costituita da massimo tre parti. L'inizio di un percorso calcolato, spesso, ha una label di due parti. La terza parte è quella che può essere modificata quando si salva la location come un preferito oppure quando contiene il nome del POI, cioè quando viene effettuata una navigazione tramite punti di interesse (le label per le POI-location di solito possono essere identificate facilmente perché hanno l'iniziale maiuscola e gli altri caratteri in minuscolo).
Scendiamo nei dettagli analizzando il formato di questa label.
Il primo byte che segue il delimitatore "00" dopo il secondo set di coordinate determina la lunghezza della prima parte della label, come mostrato in figura 14 (in rosso).
Il byte immediatamente successivo alla prima parte della label determina la lunghezza della seconda parte allo stesso modo (lo vediamo nel riquadro blu) e così anche per la terza parte (in verde). La lunghezza vera e propria di ogni parte della label è il valore di questi byte sottraendo 0x1B.
Se convertiamo questi valori in decimale otteniamo:

- 10, che è la lunghezza della stringa "Pellezzano".
- 11, che è la lunghezza della stringa "Via Filanda".
- 26, che è la lunghezza della stringa "Via Filanda 33, Pellezzano".

```
16 00 EB 1B 3E 00 25 50 65 6C 6C 65 7A 7A 61 6E  ..ë.>.%Pellezzan
6F 26 56 69 61 20 46 69 6C 61 6E 64 61 35 56 69  o&Via Filanda5Vi
61 20 46 69 6C 61 6E 64 61 20 33 33 2C 20 50 65  a Filanda 33, Pe
6C 6C 65 7A 7A 61 6E 6F 08 F7 BA 0A 21 00 00 C0  llezzano.÷°.!..À
```

Figura 15

Un'eccezione per questa procedura si ha per label lunghe (oltre i 100 caratteri): in questo caso il byte che normalmente determina la lunghezza della label avrà valore 0x08 e i due byte successivi determineranno la lunghezza della label in formato little-endian come mostrato nella figura sottostante [6].



Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
00000000	04	00	DD	0F	00	00	08	7B	00	04	00	DD	0F	00	00	01	..Ý....{....Ý....
00000016	1C	04	00	03	00	00	00	04	00	03	00	00	00	08	00	AD	...............-
00000032	CE	FF	FF	74	96	4E	00	08	00	A8	CE	FF	FF	6A	96	4E	Îÿÿt▮N...¨Îÿÿj▮N
00000048	00	21	4C	6F	6E	64	6F	6E	29	44	6F	77	6E	69	6E	67	.!London)Downing
00000064	20	53	74	72	65	65	74	08	76	00	54	68	65	20	72	65	Street.v.The re
00000080	73	69	64	65	6E	63	65	20	6F	66	20	47	6F	72	64	6F	sidence of Gordo
00000096	6E	20	61	6E	64	20	53	61	72	61	68	20	77	68	65	72	n and Sarah wher
00000112	65	20	76	69	74	61	6C	6C	79	20	69	6D	70	6F	72	74	e vitally import
00000128	61	6E	74	20	64	65	63	69	73	69	6F	6E	73	20	72	65	ant decisions re
00000144	6C	61	74	69	6E	67	20	74	6F	20	6E	61	74	69	6F	6E	lating to nation
00000160	61	6C	20	73	65	63	75	72	69	74	79	20	61	72	65	20	al security are
00000176	6D	61	64	65	20	20	65	74	20	63	65	74	65	72	61	20	made et cetera
00000192	08	94	8D	0A	0A	00	00	C0	1C	08	A3	02	14	00	9E	44	.▮▮....À..£...▮D

Figura 16

Questa eccezione è stata osservata solo con label di preferiti selezionati dall'utente, non è stato riscontrato nessun indirizzo che abbia una label più lunga di 100 caratteri. Tuttavia è possibile che questa tecnica si applichi a tutte le parti della label. Se una delle parti della label è vuota, allora il byte precedente sarà 0x1B.

4.4 Ultima posizione GPS rilevata

Dopo varie ricerche non è emersa alcuna tecnica che permettesse di trovare all'interno del file *MapSettings.cfg* l'ultima posizione GPS rilevata. L'unica informazione emersa dalle fonti analizzate è che questa posizione è presente da qualche parte nel dispositivo.

L'idea di base è stata quella di compiere più rilevazioni di posizioni GPS prossime fra loro (è stato possibili grazie alle coordinate indicate dal display del dispositivo). Tale procedimento è formato dai seguenti passi:

1. Reset del dispositivo TomTom (per avere un device con le impostazioni di fabbrica).
2. Posizionamento fisico del device in un punto di coordinate noto.
3. Rilevamento dei dati GPS presenti sull'interfaccia grafica del device e salvataggio del file *MapSettings.cfg* su un computer.
4. Ripetizione dei passi precedenti.

Dopo aver effettuato questo procedimento un certo numero di volte, si è passato al confronto dei vari file *.cfg* generati. E' emerso che la differenze tra questi file erano minime, di pochi byte, e in posizioni "simili".

Questa analisi è stata fatta utilizzando il comando shell `cmp -l`. Il comando *cmp* restituisce la posizione, in decimale, delle differenze tra due file.

Nella figura seguente si può vedere il contenuto del file *MapSettings.cfg* interpretato con l'editor esadecimale GHex (messo a disposizione di default da CAINE).

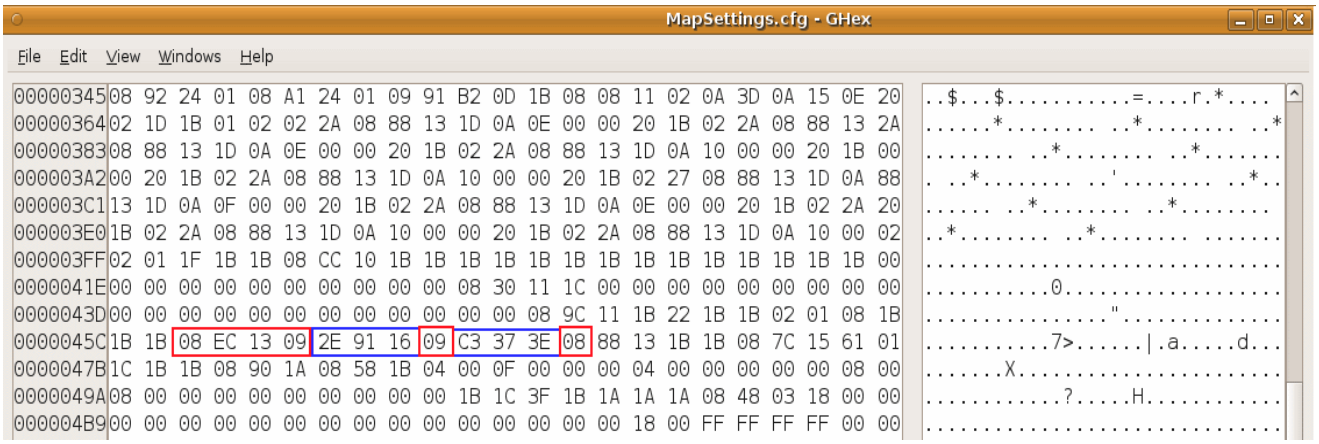


Figura 17

Dal risultato dei confronti è emerso che i byte, evidenziati in blu, cambiavano per ogni file di pochi bit, mentre quelli evidenziati in rosso rimanevano invariati.

La conclusione è stata che i byte in rosso risultano essere dei delimitatori per le coordinate dell'ultima posizione GPS rilevata dal dispositivo. Analizzando le coordinate evidenziate in blu, utilizzando il metodo di conversione descritto in precedenza, e confrontandole con le posizioni rilevate dall'interfaccia grafica del device (e con Google Maps), si è avuta la conferma che queste fossero proprio le ultime coordinate che il dispositivo ha registrato (Last GPS Fix).

Di seguito è illustrata una parte dell'esperimento condotto (in figura si può vedere una seconda rilevazione di coordinate, file MapSettings1.cfg interpretato con l'editor GHex).

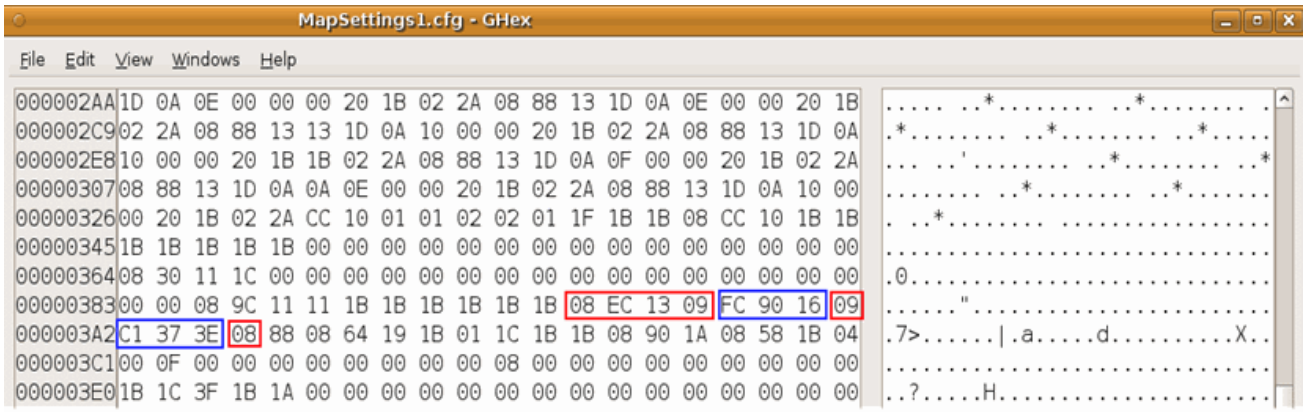


Figura 18

Si è effettuato il calcolo del valore delle coordinate presenti nel file MapSettings:

Longitudine 2E 91 16 → 16 91 2E → 1478958 → 14,78958 ;

Latitudine C3 37 3E → 3E 37 C3 → 4077507 → 40,77507 ;

e delle coordinate presenti nel file MapSettings1:

Longitudine FC 90 16 → 16 90 FC → 1478908 → 14,78908 ;

Latitudine C1 37 3E → 3E 37 C1 → 4077505 → 40,77505 ;

Ottenute queste due coppie di coordinate, sono state inserite su un sito di facile utilizzo, Google Maps, che mostra le posizioni del dispositivo, avendo come input latitudine e longitudine.

L'immagine seguente è relativa al primo set di coordinate.

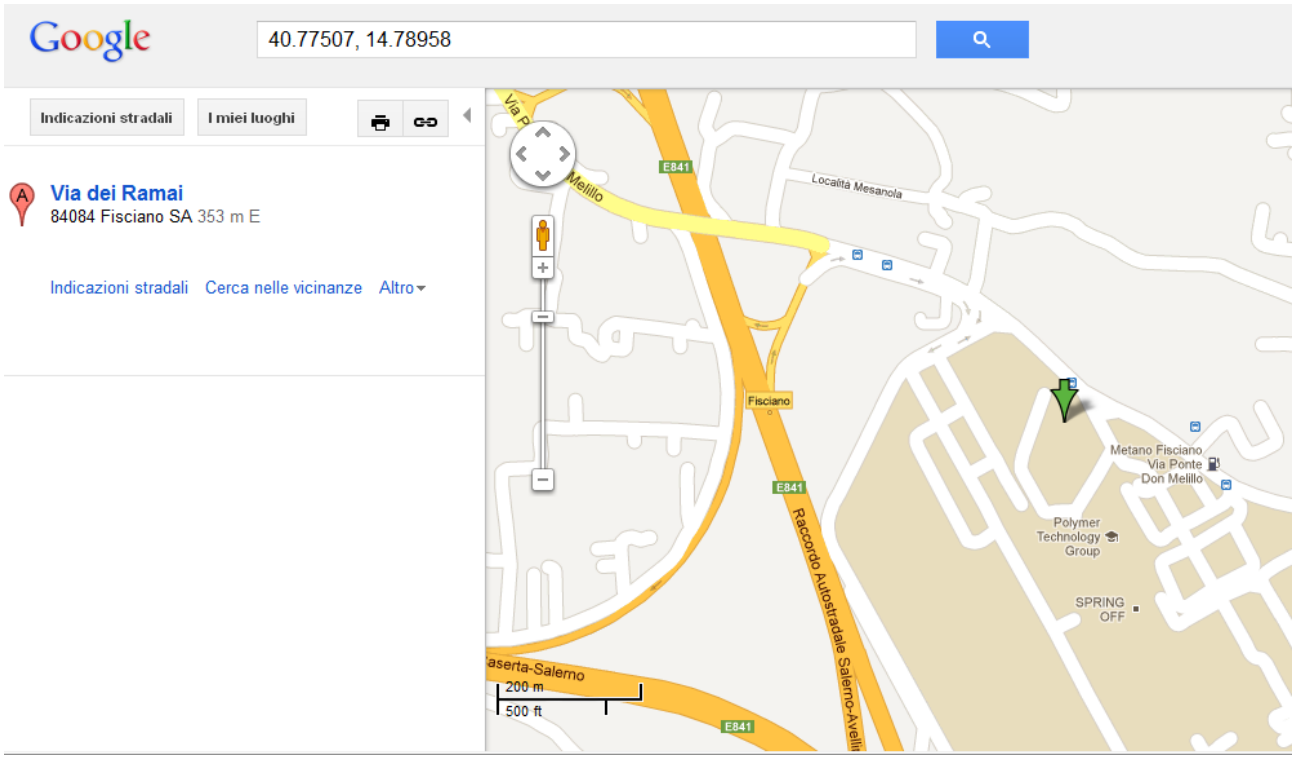


Figura 19

La successiva immagine, oltre che illustrare il secondo set di coordinate, mostra la proiezione del punto calcolato nell'immagine precedente (in verde chiaro).

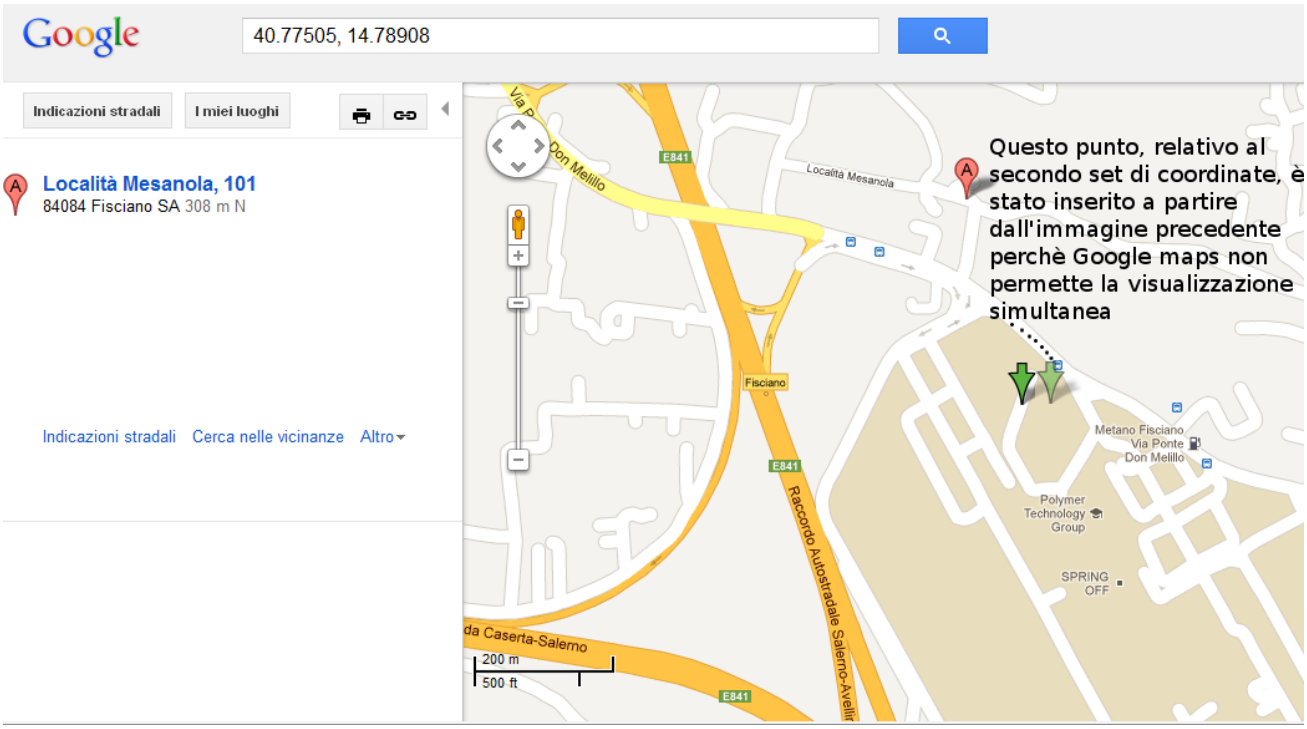


Figura 20

Come si evince dalle immagini le due posizioni rilevate sono molto prossime (c'è un po' di tolleranza nel segnale ottenuto) confermando la tesi proposta.

4.5 Triplog

Dopo aver analizzato i record contenuti nel file *MapSettings.cfg* si è passati all'analisi dei viaggi effettuati e salvati dal dispositivo TomTom.

I dispositivi TomTom raccolgono dati anonimi sull'utilizzo da parte degli utenti che lo consentono. Se un utente attiva questa funzionalità durante l'installazione del dispositivo parteciperà alla "raccolta di statistiche di utilizzo anonime" e nel file system del dispositivo, all'interno della directory */sdcard/statdata* verrà inserito un file, *allowtrip,* contenente il valore 32, e verranno salvati file del tipo "*triplog-yyyy-mm-dd.dat*". Il contenuto del file *allowtrip* è verificato alla partenza della modalità viaggio del dispositivo (esecuzione del file *ttn*) che riscriverà, qualora modificato, l'impostazione settata. In altri dispositivi dove l'analisi anonima non è consentita, è presente il file *disallowtrip*, nella medesima directory (*/sdcard/statdata/),* contenente il valore 33, che non permetterà il salvataggio dei dati. E' doveroso precisare che il salvataggio di tali dati è legato a meri fini dell'azienda TomTom per creare un nuovo servizio sempre più preciso che indica il reale tempo di percorrenza sulle strade; tale servizio offerto è

chiamato IQ Routes[15]. Secondo l'azienda TomTom IQ Routes è: "L'unico metodo basato sulla velocità media effettiva di percorrenza sulle strade e fornisce ad oggi il sistema più avanzato, al mondo, di calcolo del percorso ideale".

Come detto, questa tecnologia di navigazione intelligente si basa sulla raccolta anonima dello storico delle velocità reali relative a oltre 10 miliardi di chilometri di strade, segnalate, nel corso degli anni, da milioni di utenti TomTom. In questo modo vengono effettivamente considerati tutti i fattori che possono influenzare la durata di un percorso come l'eventuale presenza di semafori, rotatorie, strisce pedonali o dossi.

Proprio di recente (27 aprile 2011) la polizia olandese ha comprato i dati del traffico raccolti da TomTom per fornire aggiornamenti su code, incidenti e strade più congestionate [7]. Grazie alla moltitudine d'informazioni raccolte è possibile posizionare autovelox e pattuglie in modo più efficace. Chi acconsente alla raccolta di statistiche anonima, invia questi dati a TomTom quando collega il dispositivo al PC, o in tempo reale per chi ha un dispositivo connesso ad Internet.

Questi file sono criptati e, ad oggi, non si conosce nessuna tecnica per poterne leggere il contenuto.

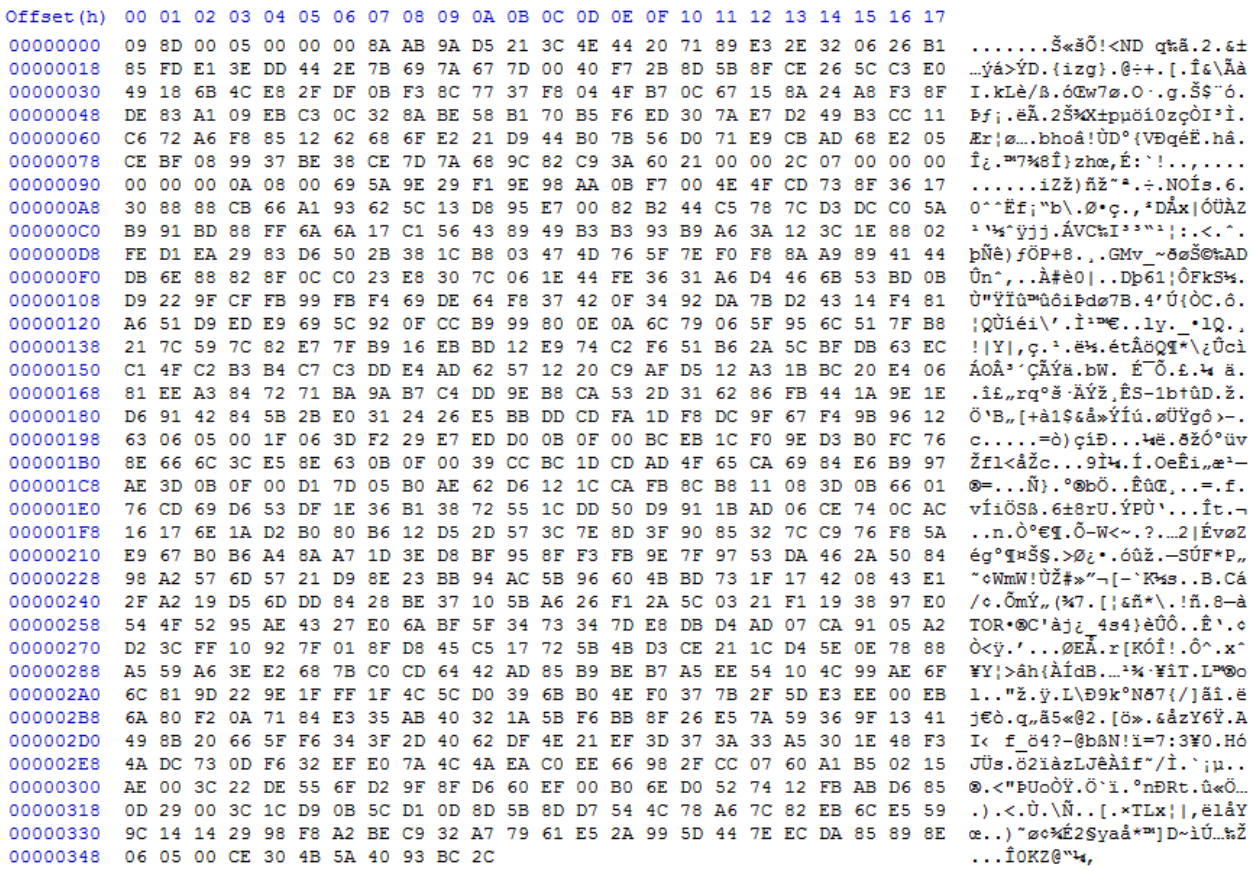
Di seguito verrà illustrata l'analisi compiuta su questi file. In una prima fase, si è tentato di capire la struttura di quest'ultimi. Per far ciò sono stati effettuati dei confronti fra più file triplog generati ad-hoc, inerenti a viaggi di breve distanza (pochi metri), per capire se ci fossero delle somiglianze fra essi.

Tutti i file triplog generati hanno in comune un header di 7 byte così composto: *09 8D 00 05 00 00 00*.

Si è verificato che, sul dispositivo in esame, se si effettuano più viaggi nello stesso giorno viene sovrascritto di volta in volta lo stesso file *triplog* e, quindi, viene memorizzato soltanto l'ultimo viaggio effettuato. Va precisato che l'operazione di scrittura del file *triplog* viene eseguita durante il viaggio.

Fin dall'inizio della sperimentazione, questo file è parso di natura crittografica, come si può vedere dalla figura 19 (file *triplog* aperto con l'editor GHex).

[15] Nel marzo del 2008 a CeBIT – Hannover TomTom svela la rivoluzionaria tecnologia IQ Routes™ per la nuova linea TomTom GO.



```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17
00000000   09 8D 00 05 00 00 00 8A AB 9A D5 21 3C 4E 44 20 71 89 E3 2E 32 06 26 B1  .......Š«šÕ!<ND q‰ã.2.&±
00000018   85 FD E1 3E DD 44 2E 7B 69 7A 67 7D 00 40 F7 2B 8D 5B 8F CE 26 5C C3 E0  …ýá>ÝD.{izg}.@÷+.[.Î&\Ãà
00000030   49 18 6B 4C E8 2F DF 0B F3 8C 77 37 F8 04 4F B7 0C 67 15 8A 24 A8 F3 8F  I.kLè/ß.óŒw7ø.O·.g.Š$¨ó.
00000048   DE 83 A1 09 EB C3 0C 32 8A BE 58 B1 70 B5 F6 ED 30 7A E7 D2 49 B3 CC 11  Þƒ¡.ëÃ.2Š¾X±pµöí0zçÒI³Ì.
00000060   C6 72 A6 F8 85 12 62 68 6F E2 21 D9 44 B0 7B 56 D0 71 E9 CB AD 68 E2 05  Ær¦ø….bhoâ!ÙD°{VÐqéË.hâ.
00000078   CE BF 08 99 37 BE 38 CE 7D 7A 68 9C 82 C9 3A 60 21 00 00 2C 07 00 00 00  Î¿.™7¾8Î}zhœ‚É:`!..,....
00000090   00 00 00 0A 08 00 69 5A 9E 29 F1 9E 98 AA 0B F7 00 4E 4F CD 73 8F 36 17  ......iZž)ñž˜ª.÷.NOÍs.6.
000000A8   30 88 88 CB 66 A1 93 62 5C 13 D8 95 E7 00 82 B2 44 C5 78 7C D3 DC C0 5A  0ˆˆËf¡"b\.Ø•ç.‚²DÅx|ÓÜÀZ
000000C0   B9 91 BD 88 FF 6A 6A 17 C1 56 43 89 49 B3 B3 93 B9 A6 3A 12 3C 1E 88 02  ¹'½ˆÿjj.ÁVC‰I³³"¹¦:.<.ˆ.
000000D8   FE D1 EA 29 83 D6 50 2B 38 1C B8 03 47 4D 76 5F 7E F0 F8 8A A9 89 41 44  þÑê)ƒÖP+8.¸.GMv_~ðøŠ©‰AD
000000F0   DB 6E 88 82 8F 0C C0 23 E8 30 7C 06 1E 44 FE 36 31 A6 D4 46 6B 53 BD 0B  Ûnˆ‚..À#è0|..Dþ61¦ÔFkS½.
00000108   D9 22 9F CF FB 99 FB F4 69 DE 64 F8 37 42 0F 34 92 DA 7B D2 43 14 F4 81  Ù"ŸÏû™ûôiÞdø7B.4'Ú{ÒC.ô.
00000120   A6 51 D9 ED E9 69 5C 92 0F CC B9 99 80 0E 0A 6C 79 06 5F 95 6C 51 7F B8  ¦QÙíéi\'.Ì¹™€..ly._•lQ.¸
00000138   21 7C 59 7C 82 E7 7F B9 16 EB BD 12 E9 74 C2 F6 51 B6 2A 5C BF DB 63 EC  !|Y|‚ç.¹.ë½.étÂöQ¶*\¿Ûcì
00000150   C1 4F C2 B3 B4 C7 C3 DD E4 AD 62 57 12 20 C9 AF D5 12 A3 1B BC 20 E4 06  ÁOÂ³´ÇÃÝä.bW. É¯Õ.£.¼ ä.
00000168   81 EE A3 84 72 71 BA 9A B7 C4 DD 9E B8 CA 53 2D 31 62 86 FB 44 1A 9E 1E  .î£„rqº·ÄÝž¸ÊS-1b†ûD.ž.
00000180   D6 91 42 84 5B 2B E0 31 24 26 E5 BB DD CD FA 1D F8 DC 9F 67 F4 9B 96 12  Ö'B„[+à1$&å»ÝÍú.øÜŸgô›–.
00000198   63 06 05 00 1F 06 3D F2 29 E7 ED D0 0B 0F 00 BC EB 1C F0 9E D3 B0 FC 76  c.....=ò)çíÐ...¼ë.ðžÓ°üv
000001B0   8E 66 6C 3C E5 8E 63 0B 0F 00 39 CC BC 1D CD AD 4F 65 CA 69 84 E6 B9 97  Žfl<åŽc...9Ì¼.Í.OeÊi„æ¹—
000001C8   AE 3D 0B 0F 00 D1 7D 05 B0 AE 62 D6 12 1C CA FB 8C B8 11 08 3D 0B 66 01  ®=...Ñ}.°®bÖ..Êû Œ¸...=.f.
000001E0   76 CD 69 D6 53 DF 1E 36 B1 38 72 55 1C DD 50 D9 91 1B AD 06 CE 74 0C AC  vÍiÖSß.6±8rU.ÝPÙ'...Ît.¬
000001F8   16 17 6E 1A D2 B0 80 B6 12 D5 2D 57 3C 7E 8D 3F 90 85 32 7C C9 76 F8 5A  ..n.Ò°€¶.Õ-W<~.?.…2|Évø Z
00000210   E9 67 B0 B6 A4 8A A7 1D 3E D8 BF 95 8F F3 FB 9E 7F 97 53 DA 46 2A 50 84  ég°¶¤Š§.>Ø¿•.óûž.—SÚF*P„
00000228   98 A2 57 6D 57 21 D9 8E 23 BB 94 AC 5B 96 60 4B BD 73 1F 17 42 08 43 E1  ˜¢WmW!ÙŽ#»”¬[–`K½s..B.Cá
00000240   2F A2 19 D5 6D DD 84 28 BE 37 10 5B A6 26 F1 2A 5C 03 21 F1 19 38 97 E0  /¢.ÕmÝ„(¾7.[¦&ñ*\.!ñ.8—à
00000258   54 4F 52 95 AE 43 27 E0 6A BF 5F 34 73 34 7D E8 DB D4 AD 07 CA 91 05 A2  TOR•®C'àj¿_4s4}èÛÔ..Ê'.¢
00000270   D2 3C FF 10 92 7F 01 8F D8 45 C5 17 72 5B 4B D3 CE 21 1C D4 5E 0E 78 88  Ò<ÿ.'...ØEÅ.r[KÓÎ!.Ô^.xˆ
00000288   A5 59 A6 3E E2 68 7B C0 CD 64 42 AD 85 B9 BE B7 A5 EE 54 10 4C 99 AE 6F  ¥Y¦>âh{ÀÍdB.…¹¾·¥îT.L™®o
000002A0   6C 81 9D 22 9E 1F FF 1F 4C 5C D0 39 6B B0 4E F0 37 7B 2F 5D E3 EE 00 EB  l.."ž.ÿ.L\Ð9k°Nð7{/]ãî.ë
000002B8   6A 80 F2 0A 71 84 E3 35 AB 40 32 1A 5B F6 BB 8F 26 E5 7A 59 36 9F 13 41  j€ò.q„ã5«@2.[ö».&åzY6Ÿ.A
000002D0   49 8B 20 66 5F F6 34 3F 2D 40 62 DF 4E 21 EF 3D 37 3A 33 A5 30 1E 48 F3  I‹ f_ö4?-@bßN!ï=7:3¥0.Hó
000002E8   4A DC 73 0D F6 32 EF E0 7A 4C 4A EA C0 EE 66 98 2F CC 07 60 A1 B5 02 15  JÜs.ö2ïàzLJêÀîf˜/Ì.`¡µ..
00000300   AE 00 3C 22 DE 55 6F D2 9F 8F D6 60 EF 00 B0 6E D0 52 74 12 FB AB D6 85  ®.<"ÞUoÒŸ.Ö`ï.°nÐRt.û«Ö…
00000318   0D 29 00 3C 1C D9 0B 5C D1 0D 8D 5B 8D D7 54 4C 78 A6 7C 82 EB 6C E5 59  .).<.Ù.\Ñ..[.×TLx¦|‚ëlåY
00000330   9C 14 14 29 98 F8 A2 BE C9 32 A7 79 61 E5 2A 99 5D 44 7E EC DA 85 89 8E  œ..)˜ø¢¾É2§yaå*™]D~ìÚ…‰Ž
00000348   06 05 00 CE 30 4B 5A 40 93 BC 2C                                        ...Î0KZ@"¼,
```

Figura 21

Dopo vari contatti con esperti della materia quali Mattia Epifani, Luigi Ranzato, Clara Maria Colombini e dopo vari esperimenti, non si è trovata una tecnica per poter leggere il contenuto (in chiaro) di un file triplog.

In una seconda fase di analisi si è riuscito ad accedere alla root del dispositivo, così da tentare vari attacchi. Questo è stato possibile grazie al software messo a disposizione da OpenTom, una comunità dedita all'hacking del TomTom, che fornisce un software per utilizzare una keyboard virtuale dove eseguire comandi shell, chiamato *TTconsole*.

Per utilizzare questo software nel dispositivo basta scaricarlo da OpenTom e copiare le directory presenti nel file compresso nella memoria interna del dispositivo (in */mnt/sdcard*).

Fatto questo, apparirà un'icona sul dispositivo (*Impostazioni -> Avanzate, Text Console*), come si può vedere dalla figura 20, e selezionandola apparirà una mini tastiera ed una shell dove sarà possibile eseguire i comandi presenti in */bin/sh*, come ad esempio il comando *ll* (figura 21).

Figura 22

Figura 23

Essendo la tastiera molto piccola ed essendo il touch del display poco sensibile, su suggerimento del dott. Albano, sono stati ideati diversi script shell e sono stati eseguiti direttamente cliccando sull'icona. Per eseguire uno script basta modificare il contenuto del file */mnt/sdcard/bin/TTconsole-wrapper*, mostrato in seguito.

```
#!/bin/sh
```

```
# Wrapper script for TTconsole (c) Markus Hoffmann 2008-2011
#
# This file is part of TTconsole, the TomTom virtual Console
# ==========================================================================
# TTconsole is free software and comes with NO WARRANTY - read the file
# COPYING for details
#
# Change these values if you like
# You can add some commandline options to TTconsole here
#options=""
options="--keyboardlayout_it --bigfont" # --bigkeys"

# set this to 'yes' (instead of no) if you want the watchdog feeder be started
# usually you want this on Systems with Navcore 8.xxx and 9.xxx

dodog=yes

#first do some settings like the PWD, HOME and PATH

cd /mnt/sdcard

export PATH=$PATH:/mnt/sdcard/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/mnt/sdcard/lib
export HOME=/mnt/sdcard/
export TERMINFO=/mnt/sdcard/lib/terminfo
export TERMCAP=/mnt/sdcard/lib/termcap

if [ "$dodog" = "yes" ]
then
kill -STOP `pidof ttn`
dogfeed &
fi

# now start the console application
TTconsole $options

#END
```

La prima cosa che si è provata a fare è stata quella di esplorare il contenuto della root "/" del dispositivo TomTom, copiandola nella memoria interna (*/mnt/sdcard/).*

Ovviamente si è prima lanciato il comando *`ls -l -R /`* per avere una panoramica della root.

Di seguito si può vedere la porzione di codice che permette di copiare, nella memora interna, tutte le directory (in maniera ricorsiva) presenti nella root.

```
cp -R /bin /mnt/sdcard/copia
```

```
cp -R /dev /mnt/sdcard/copia
cp -R /etc /mnt/sdcard/copia
cp -R /init /mnt/sdcard/copia
cp -R /lib /mnt/sdcard/copia
cp -R /proc /mnt/sdcard/copia
cp -R /root /mnt/sdcard/copia
cp -R /sbin /mnt/sdcard/copia
cp -R /sys /mnt/sdcard/copia
cp -R /tmp /mnt/sdcard/copia
cp -R /var /mnt/sdcard/copia
```

La copia è durata alcuni minuti (circa 30MB di dati, escludendo */mnt/sdcard*). Di seguito è mostrata una visualizzazione grafica della root del dispositivo TomTom (il colore blu corrisponde ad un file, il rosso ad una directory).

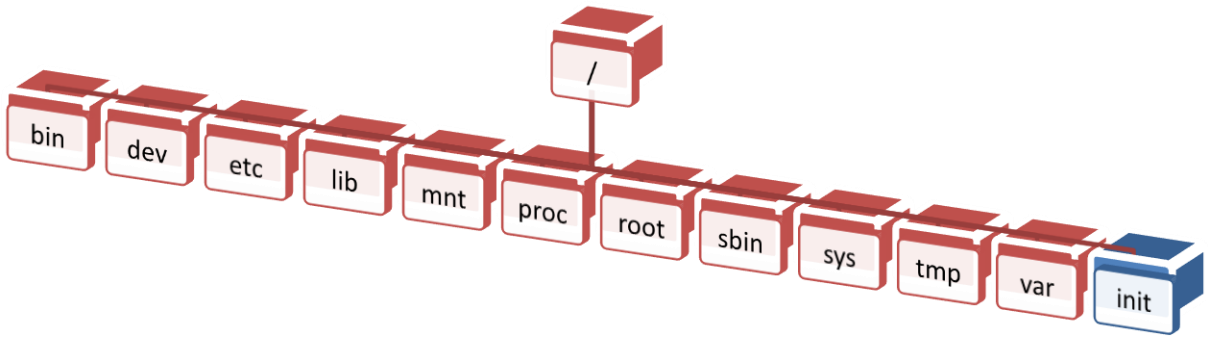


Figura 24

Si è analizzato il file *init* per comprendere cosa accade quando il dispositivo viene acceso. Questo file si compone di quattro parti:

1) Configurazione delle path, dove il dispositivo andrà a cercare le path da allocare, tra cui la sdmount, le applicazioni, i tools e i file di log della console.

```
ttntoolfound=0
fstype="vfat"
syncmntopt="rw,sync,exec,noatime,nodev,nosuid"
asyncmntopt="rw,exec,noatime,nodev,nosuid"
devbase="/dev"
sdmnt="/mnt/sdcard"
mvmnt="/mnt/movinand"
carlinkapp="carlinkd"
carlinkdir="/bin"
```

```
ttnapp="ttn"
ttntooldir="ttntools"
ttntool="ttntool.sh"
ttndir="/bin"
ttpnpdapp="ttpnpd"
ttpnpddir="/bin"
gprsengineapp="gprscomd"
gprsenginedir="/bin"
gl_baudrate="115200"
flash_gl=/mnt/flash/extra_settings
nogpsreset=/mnt/flash/nogpsreset.dat
coredumpsenable="coredumpsenable.dat"
coredir="coredir"
corepattern="%e.%t"
coresize=16000
console_logfile="cmdlog.txt"
redirect_fileflag="usecmdlog.dat"

syspath="/sys/block"
syshda0="hda"
sysmmc0="mmcblk0"
sysmmc1="mmcblk1"
countlist="0 1 2 3 4 5 6 7 8 9"
```

2) La dichiarazione delle funzioni, dove sono implementate le funzioni che verranno invocate durante l'utilizzo del dispositivo.

Le funzioni più rilevanti sono:

mount_sys()	Monta il sistema
mount_flash()	Monta la memoria flash
check_gps_receiver()	Verifica il tipo di ricevitore GPS
start_gps()	Avvia il firmware del GPS
start_ttn()	Avvia i servizi principali del dispositivo
start_shell()	Avvia la shell
start_navigator()	Invoca tutte le funzioni per la navigazione

3) Fase di start-up

La sezione che include le operazioni eseguite quando il dispositivo viene avviato: viene mostrata la versione, si monta il file system, si inizializzano i file di log e si verifica il firmware del GPS.

4) Main loop

Ciclo principale che monta la memoria interna (sdcard), carica i tools e verifica se si è nell'opzione di navigazione (avviando start_navigator). Al termine blocca il dispositivo e fa l'unmount della memoria interna (funzione unmount_storage).

Dopo aver analizzato il file di inizializzazione utilizzato dal dispositivo, si è cercato il device del ricevitore GPS, tramite il comando *mount*. Grazie alla comunità OpenTom si è appreso che i dati ricevuti dal device passavano per la pipe */var/run/gpsfeed*.
Le pipe sono uno degli strumenti di comunicazione tra processi offerti dal sistema operativo. Nei sistemi operativi Unix e Unix-like una pipe è accessibile tramite una coppia di descrittori di file, uno per scrivere dati e l'altro per leggerli. Un processo crea una pipe tramite la chiamata di sistema pipe(), che restituisce i due descrittori di file. Il flusso di dati scritto da un processo (scrittore) sul descrittore di file o HANDLE aperto in scrittura viene poi letto (nello stesso ordine) dall'altro processo (lettore) tramite il descrittore di file o HANDLE aperto in lettura. I dati generati dallo scrittore e non ancora letti sono memorizzati dal sistema operativo in un buffer di dimensioni predeterminate (tipicamente pochi KB), dal quale sono automaticamente rimossi dopo essere stati letti.
Il buffer viene usato come meccanismo di sincronizzazione tra i due processi: quando si riempie, il processo scrittore viene sospeso nell'operazione di scrittura fino a quando il lettore non ha prelevato una parte dei dati; quando il buffer si svuota il lettore viene sospeso nell'operazione di lettura fino a quando lo scrittore non ha inviato nuovi dati.

La pipe */var/run/gpsfeed* viene generata a runtime e aggiornata circa ogni secondo, quindi per visualizzare i dati dei satelliti GPS ricevuti dal dispositivo è stata fatta una redirezione dell'output della pipe *gpsfeed* all'interno di un file (denominato *log_gpsfeed*) creato nella memoria interna, tramite il seguente comando:

cat /var/run/gpsfeed >> /mnt/sdcard/nmea/log_gpsfeed.txt

Aprendo il file *log_gpsfeed.txt*, si possono visualizzare le seguenti informazioni (frammento del file):

```
[......]
$GPRMC,152826.00,A,4046.471896,N,01447.361011,E,000.0,000.0,080711,,,E
*5E
$GPGGA,152827.00,4046.479684,N,01447.359100,E,1,06,3.0,290.4,M,42.0,M,
,*65
$PGLOR,STA,152827.00,0.443,0.000,-359,1,30,1,PWR,D*26
$PGLOR,SAT,25,047,1F,29,037,17,12,044,1F,39,034,3,31,042,1F,09,022,17,
27,027,17,02,046,13*64
$PGLOR,SIO,TxERR,1,RxERR,0,TxCNT,244,RxCNT,1772,DTMS,926,DTIN,2,DTOUT,
166*76
```

```
$GPGSV,3,1,11,25,71,348,47,29,64,241,37,12,52,075,44,39,41,164,34*71
$GPGSV,3,2,11,31,27,315,42,09,19,154,22,27,09,152,27,02,01,000,46*71
$GPGSV,3,3,11,33,33,221,,40,21,119,,04,05,034,*40
$GPGSA,A,3,09,12,25,27,29,31,,,,,,,2.3,1.6,1.7*33
$PGLOR,FIX,1.0*3E
$GPRMC,152827.00,A,4046.479684,N,01447.359100,E,000.0,000.0,080711,,,A
*54
 [......]
$GPGSV,3,1,11,25,71,350,47,29,65,242,35,12,52,076,44,39,41,164,32*7D
$GPGSV,3,2,11,02,36,065,41,31,28,314,36,09,19,154,38,27,08,152,22*73
$GPGSV,3,3,11,33,33,221,,40,21,119,,04,04,034,*41
$GPGSA,A,3,02,09,12,25,27,29,31,,,,,,1.6,1.2,1.2*36
$PGLOR,FIX,1.0*3E
[......]
```

Il contenuto di questo file è espresso nel formato NMEA 0183. L'interpretazione del contenuto è la seguente: il formato NMEA è costituito da un delimitatore di partenza, seguito da una sequenza di campi separati da virgole, terminanti con il carattere '\ *', seguita da un checksum CRC32 espresso con due caratteri esadecimali (tale codice è ottenuto facendo lo XOR dei campi che compongono una riga esclusi i caratteri $ e *).

Mostriamo un esempio di calcolo del codice CRC32 nel file *gpsfeed*:

`$PGLOR,FIX,1.0 * 3E <--CRC32` (ultima riga del file sopraelencato)
Convertendo la riga in esadecimale otteniamo *24 50 47 4C 4F 52 2C 46 49 58 2C 31 2E 30 2A 33 45* con le ultime 3 byte (`2A 33 45`) che corrispondono in ASCII a `* 3E`).

Escludendo i caratteri $ (24) e * (2A) e mettendo in XOR tutti i simboli rimanenti, otteniamo: $50 \oplus 47 \oplus 4C \oplus 4F \oplus 52 \oplus 2C \oplus 46 \oplus 49 \oplus 58 \oplus 2C \oplus 31 \oplus 2E \oplus 30 = 3E$, che è proprio il checksum salvato all'interno del file.

Adesso analizziamo il campo ritenuto più importante: GPRMC.

Esso è un componente essenziale per l'informazione GPS PVT (*position, velocity and time*). E' chiamato RMC (*the Recommended Minimum specific GPS Transit Data)* che va letto come di seguito:

RMC	Recommended Minimum specific GPS Transit Data
152826.00	Orario 15:28:26 UTC
A	Stato A=active or V=Void.
4046.471896,N	Latitudine 40 gradi, 46 primi, 28 secondi, N
01447.361011,E	Longitudine 14 gradi, 47 primi, 21 secondi, E

010.3	Velocità espressa in nodi
084.4	Angolo espresso in gradi
080711	data - 8 Luglio 2011
“ ”, E	Variazione magnetica
*5E	Checksum CRC32

Tabella 3

Non conoscendo il processo che genera tale pipe *gpsfeed* presente in */var/run*, si è cercato di sostituire il contenuto. Per prima cosa è stata cancellata, se ne è creata una ad-hoc, rispettando il formato NMEA analizzato, gli sono stati assegnati i permessi e si è cercata di inserire con il comando *cp*, per spostarla in */var/run*. Tale procedimento non ha sortito esito positivo. Si è analizzata anche la pipe */var/run/gpspipe*. Questa pipe però si è dimostrata più lenta e ha prodotto l'output in ritardo rispetto alla pipe *gpsfeed* che ha un tasso di aggiornamento inferiore al secondo ed è presente soltanto dopo l'avvio di *navcore*. Anche per questa pipe si è cercato di sostituire il contenuto ma non si è ottenuto riscontro positivo.

Dopo aver interpretato il flusso degli eventi si è cercato il riferimento alla parola triplog, per vedere il file che realmente generava triplog, o un qualsiasi legame.

Questo è stato possibile grazie al comando `grep -lir "triplog" /*` esguito nella root del dispositivo, e si è ottenuto riscontro nel file *ttn*, presente nella directory */bin*. Questo però è un file eseguibile.

Si è cercato di disassemblare il file, con l'utilizzo del Software IDA PRO, ma questa operazione non ha avuto l'esito sperato.

Si è aperto allora questo file eseguibile con un editor esadecimale. All'interno di questo file si sono trovati dei riferimenti al file triplog:

```
triplog-%04d-%02d-%02d.dat
```

Oltre a questo, nel file compaiono altri riferimenti in chiaro:

```
26CTripEncryptionBlowFishRSA....23CTripEncryptionBlowF
ish...23CTripEncryptionStrategy
```

Si pensa che questi siano gli schemi crittografici utilizzati per cifrare il file *triplog*.

Altra voce interessante trovata sempre in questo file è:

```
13CTripRecorder.%02d:%02d:%02d..(?).%s,%s,%s,%s,%s,%s,
%s,%s,%s,%s,%s,%s,%s,%s,%s
```

Si è modificato il contenuto di queste voci per studiare l'effetto sul dispositivo.
La stringa "*triplog-%04d-%02d-%02d.dat*" è stata sostituita con la stringa "*xxxxlog-%04d-%02d-%02d.dat*". Grazie allo script *TTconsole*, opportunamente modificato, il file è stato copiato nella relativa path (*/bin*).

Dopo aver compiuto un viaggio è emerso che effettivamente tali modifiche venivano apportate ed il file *xxxxlog-2011-07-13* veniva salvato.

Si è tentato di cambiare anche le altre informazioni presenti in chiaro nel file, ma senza ottenere risultati.

4.6 File system proc & minor e major

Per compiere un'azione di anti-forensic volta a inserire dati GPS fittizi all'interno del file triplog, sono stati fatti vari esperimenti.

Su suggerimento del dott. Castiglione, è stato analizzato il file system proc del dispositivo per controllare l'accesso alle periferiche hardware presenti sul sistema e stabilire quando e come i vari processi interagiscono con esse.

Nei sistemi operativi Unix-like, procfs (process file system) è uno pseudo-file system usato per accedere alle informazioni, relative ai processi, fornite dal kernel. Il file system è montato nella directory */proc*; poiché è un file system virtuale, esso necessita di poco spazio sul disco rigido ed una limitata quantità di memoria.

La directory */proc* contiene una gerarchia di file speciali che rappresentano lo stato corrente del kernel consentendo alle applicazioni e agli utenti di esplorare il sistema attraverso il punto di vista del kernel.

All'interno della directory */proc* si trovano numerose informazioni sull'hardware e su qualsiasi processo attualmente in esecuzione. Inoltre, alcuni file all'interno dell'albero della directory */proc* possono essere manipolati dalle applicazioni per comunicare al kernel eventuali modifiche di configurazione.

Nei sistemi operativi Unix e Unix-like un dispositivo a blocchi (block device) è un tipo speciale di file, chiamato anche file di dispositivo, o device file, che rappresenta una periferica (come /dev/sda) o un dispositivo virtuale su cui è possibile effettuare operazioni di input/output per blocchi di byte di dimensione predeterminata. I dispositivi a blocchi sono caratterizzati da due numeri, detti major number e minor number, che li identificano internamente al kernel, e che sono specifici per la particolare implementazione. Il major number identifica il device driver mentre il minor number è utilizzato dal driver stesso per distinguere i diversi tipi di dispositivo che il driver è in grado di gestire.

La figura seguente illustra il riconoscimento e i riferimenti di un dispositivo da parte del kernel Linux:

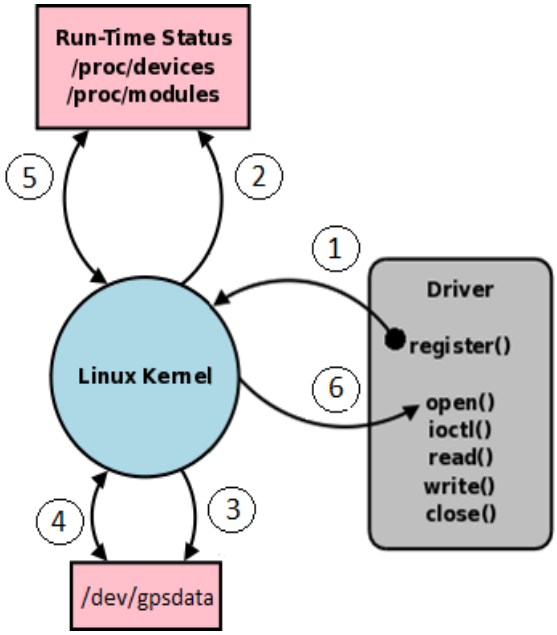


Figura 25

Nelle prime 3 fasi viene mappato il dispositivo:

1. Viene caricato il modulo del kernel.

2. Vengono aggiornate le variabili di runtime (/proc/devices, /proc/modules).

3. Viene creato il device file (/dev/gpsdata) con il tipo appropriato e gli vengono assegnati i riferimenti major e minor.

Nelle altre 3 fasi avviene l'interazione con il dispositivo:

4. Viene identificato il tipo di dispositivo e i riferimenti minor e major (comando `ls -l /dev`).

5. Vengono identificati i moduli associati al device file.

6. Il controllo viene trasferito alla funzione scelta nel driver del dispositivo.

L'idea è quella di sostituire il block device della periferica GPS (presente in */dev/gpsdata*) con uno fittizio, utilizzando major e minor ad-hoc. E' stato rimosso il block device originale tramite il comando *rm /dev/gpsdata* ed è stato creato un nuovo block device GPS, tramite il comando *mknod /dev/gpsdata b <major> <minor>.*
Non avendo trovato i riferimenti major e minor in */dev/* tale lavoro è risultato inefficace.

L'ultimo esperimento tentato è stato quello di modificare i riferimenti al block device della periferica GPS all'interno del file */etc/rc.glgps* che si occupa del controllo del GPS e che sembra generi la pipe */dev/gpspipe*. Questo file è stato copiato nella memoria interna, poi è stato sostituito il percorso */dev/gpsdata* con il percorso di un file contenente dati NMEA fittizi.

Purtroppo nessuno degli esperimenti condotti ha permesso di sostituire i dati GPS ricevuti dal dispositivo con dei dati fittizi.

5 Conclusioni e sviluppi futuri

Da questo studio è emerso che i dispositivi GPS possono rivelarsi una fonte inesauribile di informazioni di carattere forense. Nel caso specifico del TomTom la letteratura disponibile è però carente. Le fonti presenti spesso non sono il frutto di studi accademici, bensì di sperimentazioni compiute da possessori di questi dispositivi.

I risultati di maggior rilievo ottenuti sono stati i seguenti:

- Aver reso la copia forense della memoria del dispositivo un'operazione ripetibile nel tempo, perché viene preservata l'integrità dei dati;

- Aver individuato una tecnica per risalire all'ultima posizione GPS rilevata dal dispositivo.

Pur conducendo vari esperimenti sui file di log dei viaggi (*triplog*), che risultano essere i file di maggior interesse da un punto di vista forense, non si è riuscito né a comprendere la struttura o le informazioni in essi contenute, né si è riuscito a compiere un'azione di anti-forensics volta alla compromissione dei dati presenti in questi file.

Ad oggi l'unica via percorribile per ottenere i file triplog in chiaro è quella di richiedere la decifratura all'azienda TomTom. Questa via è tortuosa, in quanto, anche richieste pervenute da parte di agenzie di law enforcement non hanno ottenuto alcuna risposta.

Una possibile sperimentazione per cercare di decifrare questi file potrebbe essere l'analisi della memoria RAM del dispositivo (tramite l'utilizzo di JTAG [8]), in cui è probabile che siano presenti le chiavi degli algoritmi di cifratura oppure i dati in chiaro destinati al file *triplog*, così da poter effettuare un attacco di tipo known-plain-text.

Un'ulteriore spunto potrebbe essere quello di andare ad individuare i riferimenti major e minor utilizzati dal kernel per interfacciarsi con il device */dev/gpsdata* e modificare il flusso degli eventi andando a collocarsi tra kernel e device cosicché i dati presi in input dal kernel siano dati fittizi, per far cifrare file a piacimento così da condurre un attacco di tipo known-palin-text o quantomeno andare a falsificare i viaggi effettuati inducendo l'azienda TomTom, o un analista, a credere che tali viaggi siano stati compiuti, creando un alibi.

L'intera analisi potrebbe essere condotta su un dispositivo TomTom con funzionalità Bluetooth, poiché è possibile accedervi tramite un indirizzo di rete, direttamente durante la modalità utente.

6 Riferimenti

[1] GPS, Wikipedia (http://en.wikipedia.org/wiki/Global_Positioning_System).

[2] TomTom, Wikipedia (http://en.wikipedia.org/wiki/TomTom).

[3] Colombini Maria Clara - Experimental Testing of a Forensic Analysis Method on the TomTom GPS Navigation Device, April 2009 (http://www.iisfa.net/index.php?option=com_docman&task=doc_download&gid=19&Itemid=40).

[4] Mattia Epifani - Digital Evidence: dall'hard disk ai social network, May 2011. (www.associazionearchimede.it/Unisa/phocadownload/ssi2011.pdf).

[5] OpenTom (http://www.opentom.org/Main_Page).

[6] Beverley Nutter - Pinpointing TomTom location records: A forensic analysis, Digital Investigation, Volume 5, Issues 1–2, September 2008, Pages 10-18, ISSN 1742-2876, 10.1016/j.diin.2008.06.003. (http://www.sciencedirect.com/science/article/pii/S1742287608000479).

[7] AD - TomTom tipt politie over verkeersmisbruik, April 2011. (http://www.ad.nl/ad/nl/5597/Economie/article/detail/2426526/2011/04/27/TomTom-tipt-politie-over-verkeersmisbruik.dhtml).

[8] Van Eijk Onno & Roeloffs Mark - Forensic acquisition and analysis of the random access memory of tomtom GPS navigation systems, Digital Investigation 2010, Volume 6 (3-4), 179-188. (http:/www.sciencedirect.com/science/article/pii/S1742287610000137).