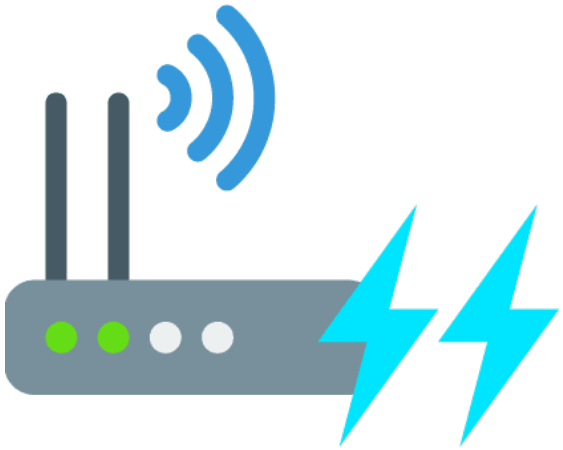
IIS EUGANEO

DEVCODE
WEB & SOFTWARE



WIFI PROBE AGGREGATOR

By Fabio Lovato

Introduzione

Siamo nel 2020, e la tecnologia è ovunque: nelle nostre case, negli uffici a lavoro, a scuola, negli ospedali, nelle piazze e in qualsiasi altro luogo. All'incirca dagli anni 2000 abbiamo subito un'enorme evoluzione digitale, in molti casi talmente veloce da non renderci conto che a volte non riusciamo a starci dietro. Alcune tecnologie nascono, vengono usate e muoiono senza che ce ne accorgiamo, nel bel mezzo di molte altre che invece hanno avuto successo.

Altre tecnologie invece sono diventate già da tempo la base su cui si fondano la rete Internet e le nostre attività quotidiane, come ad esempio il WiFi, tecnologia nata con la sua prima versione nel **1997** alla velocità di **2 Mbps**, oggi arrivata quasi a **10 Gbps** (WiFi versione 6, IEEE 802.11ax).

Dopo ben 23 anni, il WiFi continua ad essere un canale molto utilizzato, soprattutto dopo l'avvento di smartphone e tablet.

La **community hacker**, intesa come guppi di persone curiose che smontano, rimontano, modificano e sperimentano su qualsiasi cosa, è sempre sul pezzo. Molte invenzioni sono nate proprio da loro grazie alla loro sperimentazione e al rilascio di software con licenze open source. Anche molte falle di sicurezza sono state scoperte proprio da loro e spesso risolte, grazie alla loro sperimentazione continua, spesso spinta dalla passione e dalla curiosità.

Anche se una tecnologia ha molti anni alle spalle non significa che sia completamente sicura e testata, o che sia la tecnologia migliore possibile per garantire la sicurezza e privacy dei nostri dati. Nel caso del WiFi, alcuni sviluppatori curiosi (chiamiamoli così) hanno sviluppato dei piccoli software (o script) che, grazie alla **modalità promiscua della scheda di rete wireless** del proprio pc, riescono a rilevare dispositivi nelle vicinanze e tracciarli. Non stiamo parlando dei router che si possono già scansionare solamente scorrendo la lista delle WiFi disponibili delle vicinanze con uno smartphone, ma parliamo dei dispositivi client, ossia altri smartphone, tablet, notebook, pc, telecamere di sorveglianza, ecc.

Come funziona?

Il flusso di funzionamento della connessione di un dispositivo client, ad esempio uno smartphone, avviene in questo modo:

- attivo la rete WiFi
- scansiono le reti presenti
- seleziono una rete conosciuta, ad esempio “CasaWifi”, e mi collego (che sia con password o senza)
- posso utilizzare la rete WiFi “CasaWifi” configurata

Molto semplice.

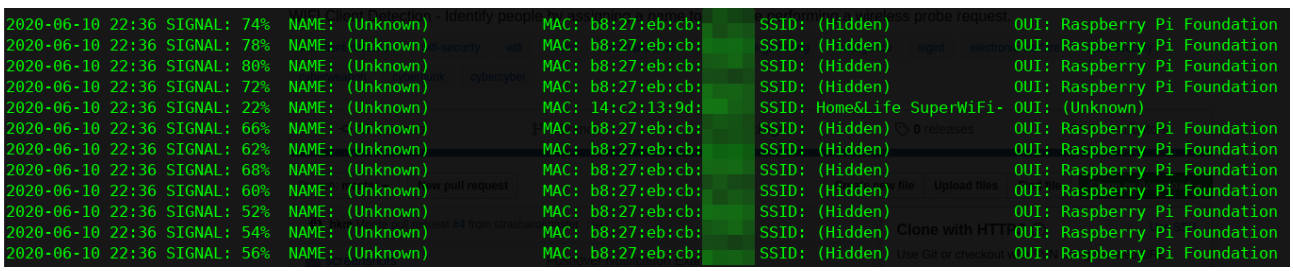
Cosa succede quando scollego lo smartphone dalla rete WiFi ed esco dal raggio di azione del WiFi? Lo smartphone, se ha la rete WiFi ancora attiva, effettua un PROBE della rete inviando un pacchetto nell’etere chiedendo: “Ehi CasaWifi, rispondimi che voglio collegarmi!”. Se il router WiFi non è nelle vicinanze, lo smartphone continuerà dopo X secondi (X dipende dal produttore del chipset) a chiedere “Ehi CasaWifi, rispondimi che voglio collegarmi!”. Quando si entrerà nel raggio della rete CasaWifi il router risponderà “Eccomi, ora puoi connetterti!”.

Ciò che interessa a noi è la fase in cui il dispositivo invia quella richiesta di connessione ogni X secondi, perché quella richiesta si può leggere da qualsiasi dispositivo WiFi che sia in **modalità promiscua**, ossia una modalità in cui riceve tutti i pacchetti nelle vicinanze, anche quelli non destinati a sé.

Il pacchetto che il client invia contiene alcune informazioni interessanti:

- il proprio MAC address, indispensabile per la connessione WiFi
- l’ESSID o il MAC address del router WiFi a cui tenta di collegarsi, indispensabile per fare in modo che il router WiFi con quell’ESSID possa sapere che qualcuno sta chiamando esattamente lui e possa quindi rispondere
- la potenza del segnale fra il client e il dispositivo che riceve il pacchetto

Questo è l’output di uno script che raccoglie queste informazioni:



Queste 3 informazioni sembrano poco o nulla, però se combinate possono farmi ottenere queste ulteriori informazioni:

- un **identificativo univoco** di un dispositivo nei paraggi (il MAC address)
- il **nome delle reti WiFi** a cui si è collegato, quindi i probabili luoghi dove è stato (per ogni rete WiFi configurata, il client chiede continuamente di collegarsi a ognuna di queste, svelando i nomi delle reti conosciute)
- la **data e ora** in cui uno specifico dispositivo è nelle vicinanze: se qualcuno ci sta pedinando o fosse nei pressi della nostra casa potremmo scoprirlo proprio così, ovviamente se ha appresso uno smartphone!
- la **marca** del client: le prime 3 cifre del MAC address corrispondono ad uno specifico produttore, per cui se in una stanza ci sono 2 persone con smartphone, posso sapere quale delle 2 ha un telefono Android e quale un Apple
- la **distanza indicativa** del dispositivo: se in una stanza ci sono 4-5 persone, in base alla potenza del segnale riesco a identificare qual è il MAC address di ciascuna in base alla loro distanza
- **DOVE** sono queste persone, o dove sono state: se si costruisce una rete di dispositivi in cui ciascuno registra i MAC address nelle vicinanze più la propria geolocalizzazione, più questa rete è numerosa e più riesco a tracciare le posizioni dei dispositivi univoci. Se poi riesco ad associare a un MAC address il nome di una persona riconoscendola visivamente, le informazioni raccolte diventano un po' più delicate
- se queste persone sono state nello **stesso luogo**: raccogliendo in un database il MAC address di ogni client e le reti a cui tenta di collegarsi, posso estrarre una lista di client che tentano di collegarsi alla stessa rete
- **livello PRO** (non trattato nel progetto): esistono dei metodi per rispondere a tutte le richieste di collegamento inviate dai client. Quando il client invia nell'etere "Ehi CasaWifi, rispondimi che voglio collegarmi!", un software creato ad-hoc potrebbe rispondere "Ehi sono io, collegati!". A quel punto il client si connette automaticamente poiché la rete è conosciuta e si può analizzarne il traffico in modalità passiva, come i siti web visitati, username e password che transitano nella rete, ecc.

Gli obiettivi del progetto possono aumentare, ma per ora ci fermeremo a raccogliere questi dati tramite un pc con scheda wireless, inviarli ad un gestore di dati (database centrale) e visualizzarli su un software web based in HTML.

Il limite dell'utilizzo di un pc o di un portatile è che si deve avere appresso un computer che è un po' ingombrante. Una interessante evoluzione può essere lo sviluppo di un'app, così da sfruttarne la comodità e la geolocalizzazione, oppure un dispositivo con Arduino o Raspberry a batterie, sempre per la portabilità.

Struttura del progetto

1) Raccoglitore dati

Esistono dei Poc (Proof of Concept) già realizzati per raccogliere questi frame WiFi inviati dai client, che in questo momento utilizzeremo per non svilupparli completamente da zero. L'idea è di modificarli per poterli adattare e inviare i dati al gestore.

Il software identificato e testato si chiama **WhoIsHere**, è scritto in Python 2, e sarà da modificare per inviare i dati a un database oltre che visualizzarli a video:

https://github.com/hkm/whoishere.py

Nelle ultime versioni è stato integrato un sistema per inviare in formato JSON la lista dei dispositivi rilevati per ricevere una notifica di presenza su un'app da loro consigliata, ma nel nostro caso non serve. Potremo invece inviare l'oggetto JSON al nostro gestore, opportunamente predisposto, per poterli salvare poi su database, e metterli a disposizione della GUI per la consultazione.

Tempo stimato: 5gg

2) Gestore dati (Web Service)

I dati possono essere raccolti localmente da ogni pc, ma creando invece un gestore di dati centralizzato è possibile far confluire in un unico archivio tutti i dati raccolti e visualizzare molti più dati aggregati.

Si può usare come base dati un database MySQL con il minimo di dati indispensabili, comunicando direttamente con il database o, meglio, creando un web service, ossia un'interfaccia di comunicazione in PHP che riceve dal nostro pc i MAC address dei dispositivi attorno, in formato JSON.

Un secondo web service che mette a disposizione i dati raccolti leggendoli da database e fornendoli sempre in formato JSON, opportunamente rielaborati, per la GUI.

Tempo stimato: 5gg

3) GUI

La visualizzazione dei dati è l'ultimo step. Permette di rendere intuitiva la consultazione dei dati raccolti.

I dati da visualizzare potrebbero essere molti, ma spesso le GUI di successo hanno pochi dati, ben ordinati e intuitivi, così che anche chi non conosce lo scopo del software a prima vista riesce a capirne l'utilità.
Potrebbero essere sviluppate queste schermate:

Tempo stimato: 5gg

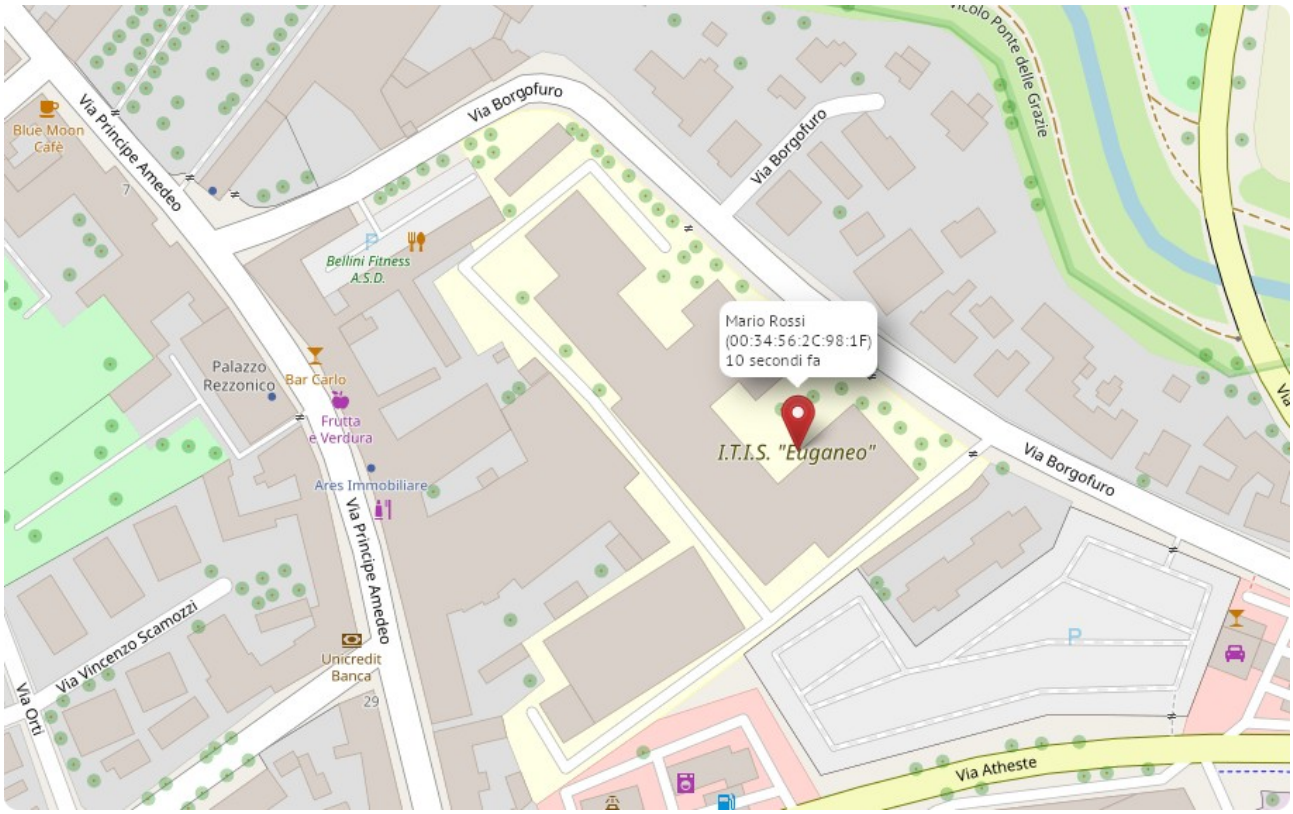
Dashboard

Schermata in forma tabellare che visualizza i dispositivi nelle vicinanze. Le colonne visualizzabili potrebbero essere:

- logo del produttore (Huawei, Apple, Samsung, ecc), in base alle prime cifre del MAC Address
- MAC address del client rilevato (successivamente si potrà creare una funzione per associare un nome ad un MAC address se abbiamo identificato la persona)
- data e ora del rilevamento
- potenza del segnale

Mappa

Schermata che visualizza su mappa i dispositivi rilevati negli ultimi X minuti. La geolocalizzazione per il momento sarà statica, nel senso che inviando i dati tramite un pc conviene inizialmente inviare staticamente la propria latitudine e longitudine al web service. I dispositivi WiFi rilevati avranno come posizione la nostra stessa, poiché sono nelle vicinanze e non si possono tracciare con più precisione.



Ovviamente, se la raccolta dati è fatta dallo stesso pc, la geolocalizzazione sarà sempre statica, mentre se è fatta da più persone, è possibile vedere più punti sulla mappa.

Luoghi

Schermata che visualizza in forma tabellare una lista di reti a cui i client wireless hanno tentato di connettersi. Per ciascuna rete saranno visualizzati la lista dei dispositivi wireless che hanno tentato di connettersi, così da eseguire delle statistiche su quanti e quali dispositivi wireless si sono connessi alla stessa rete e identificare dispositivi (o meglio, persone) che si conoscono.

Strumenti

- **Python:** è il linguaggio usato per il primo step. Whoishere.py è uno script già realizzato in Python al quale servono alcune modifiche per inviare i dati al nostro web service in formato JSON.
  https://stackoverflow.com/questions/9733638/post-json-using-python-requests

- **PHP e JSON:** con PHP è possibile gestire dati in formato JSON. Qui alcuni esempi di utilizzo e di come funziona il formato JSON.
  https://www.html.it/articoli/introduzione-a-json/
  https://www.html.it/pag/64673/funzioni-php-per-json-2/

- **Bootstrap:** è un framework CSS per sviluppare GUI in HTML. Con l'utilizzo di classi CSS è possibile creare interfacce grafiche molto accattivanti e soprattutto responsive, ossia adattabili automaticamente alla risoluzione dei dispositivi, come smartphone o tablet.
  https://getbootstrap.com/

- **SB Admin 2:** è un template per Bootstrap con degli stili aggiuntivi per creare GUI adatte per un software web-based. Per l'uso non ci sono particolari guide, dal link qui sotto è necessario solamente scaricare i CSS e JS a disposizione, e scrivere i tag HTML con le classi che sono state usate per gli esempi:
  https://startbootstrap.com/themes/sb-admin-2/

- **LeafletJS:** è una libreria javascript che semplifica l'inclusione di mappe con effetti e funzionalità molto interessanti e predisposte in modo semplice da usare, anche utilizzando diversi sorgenti dati (Google Maps, OpenStreetMap, ecc). Nel nostro caso useremo OpenStreetMap:
  https://leafletjs.com/