LinuxDay

26 Ottobre 2019

Catturare il Traffico delle App Android Senza Root

Emanuele Faranda

black.silver@hotmail.it

Contribute!

https://github.com/emanuele-f/RemoteCapture

Perchè analizzare il Traffico delle App

- **Con chi parla l’app?**
- **L’app usa canali sicuri per lo scambio dei dati?**
- **Quali richieste DNS/HTTP fa?**
- **Catturare un PCAP del traffico**

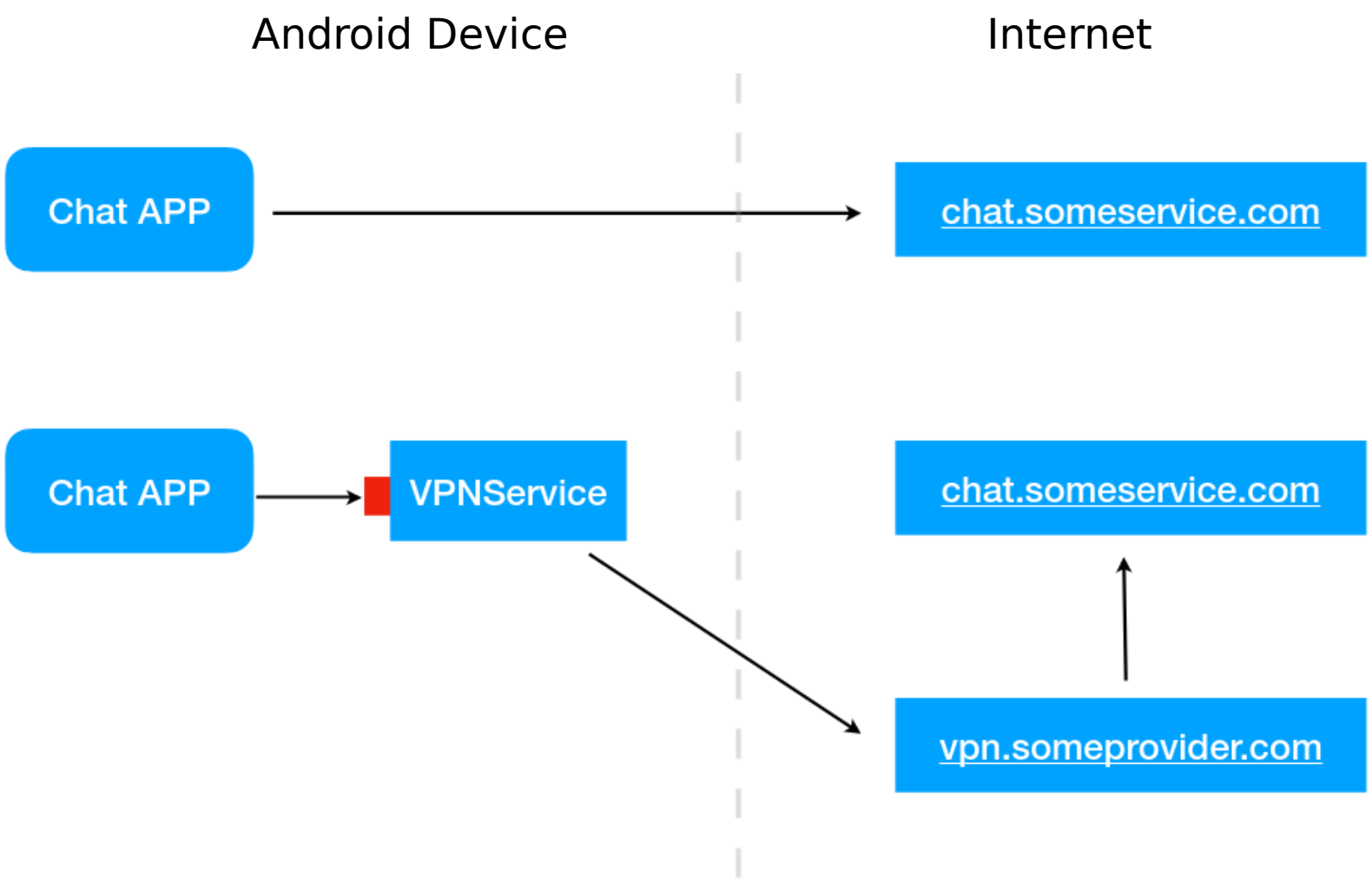
VPNService

- **Servizio Android ideato per implementare app di VPN**
- **Permette di deviare i pacchetti inviati dalle app e gestirli con una logica propria**
- **Indicato da una chiave nella barra di stato**

Potenzialità offerte del VPNService

- **Modificare il traffico delle app**
- **Forzare l’uso di un DNS server specifico**
- **Collegare una sottorete remota (es. aziendale)**
- **Bloccare selettivamente il traffico di un’app**
- **Intercettare il traffico cifrato delle app (prima di Android 7)**

VPNService Funzionamento



Setup VPNService

1. intent = VpnService.prepare(MainActivity.this)
2. startActivityForResult(intent)
3. onActivityResult -> startService(intent)

4. Configurare opzioni tramite Builder VPN
5. Builder.establish() -> file descriptor

6. Gestire i pacchetti tramite il file descriptor

7. OnRevoke -> termina

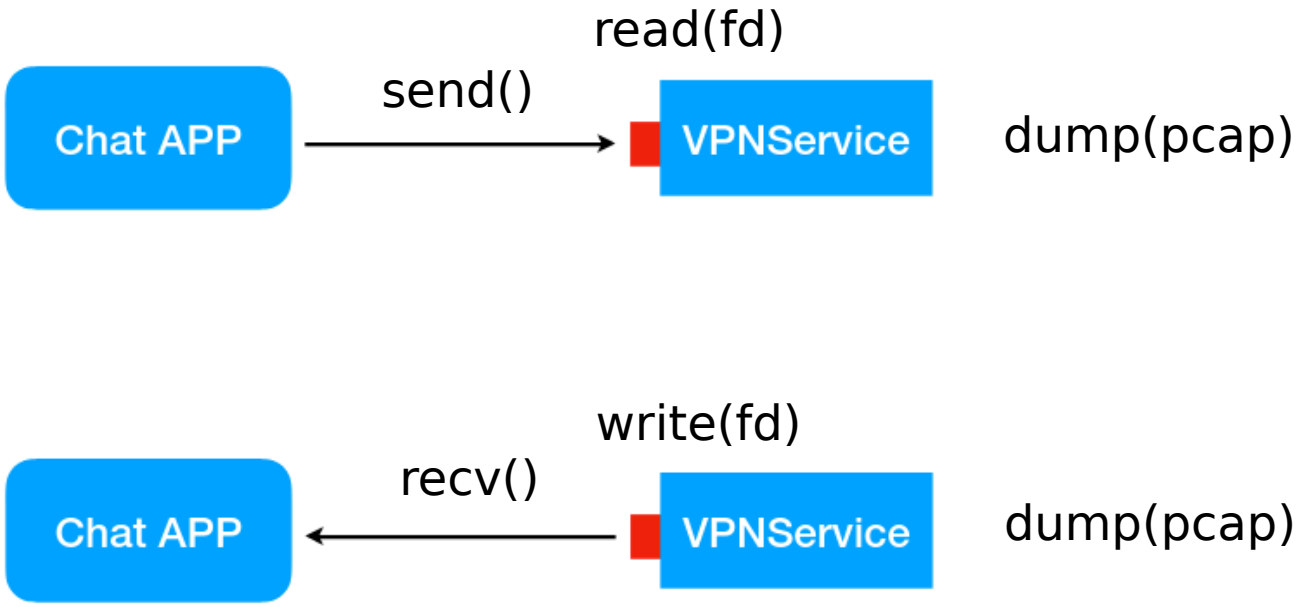
VPNService Builder

```
int onStartCommand(Intent intent, int flags, int startId) {
  Builder builder = new Builder()
    .addAddress("10.215.173.1", 30)        // VPN interface IP and netmask
    .addRoute("0.0.0.0", 1)                // Route all the traffic to the VPN
    .addDnsServer("10.215.173.2");         // The (fake) DNS server to use

  fd =  builder.setSession(“My VPN”).establish();
  runPacketLoop(fd.detachFd(), this);
}


public static native void runPacketLoop(int fd, CaptureService vpn);
```

Catturare i Pacchetti



Problema: come inoltro il pacchetto ricevuto al suo destinatario originale?

Strategia

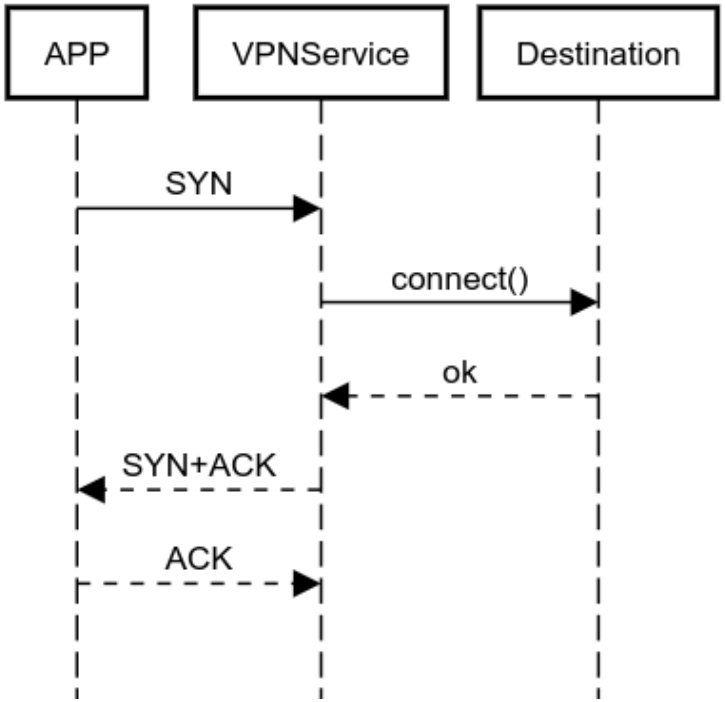
- **Android non fornisce un API per reiniettare il pacchetto nel suo percorso originale**
- **Sarebbe semplice forwardare i pacchetti tramite socket RAW ma sono richiesti i privilegi di root**

Soluzione:

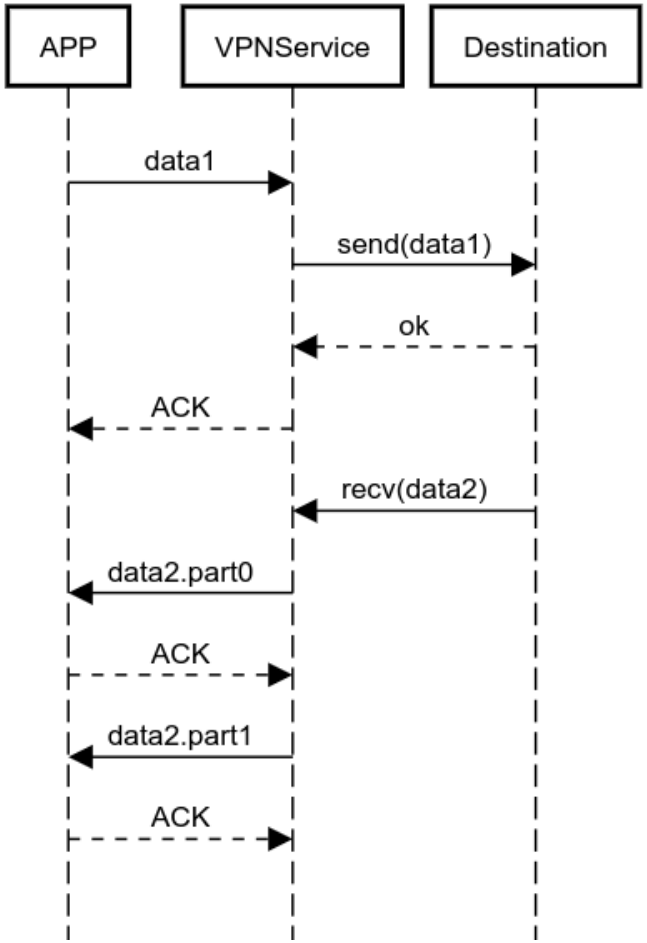
- **Usare i comuni socket IP in versione UDP/TCP/ICMP**
- **Aprire un socket nuovo per ogni nuova comunicazione di un app verso internet**
- **Tener traccia dello stato del socket**

Esempio

Handshake



Invio e Ricezione dati



Complicazioni

- **Gestire i numeri di sequenza e ack TCP**
- **Gestire il three-way-handshake TCP con l’app**
- **Bufferizzare i dati ricevuti da internet per rispettare la window size TCP dell’app**
- **Bisogna gestire in maniera asincrona le richieste per evitare di rallentare tutto**

- **Tuttavia, non è necessario gestire ritrasmissioni, poiché Android ci assicura che le write sul fd arrivino all’app**

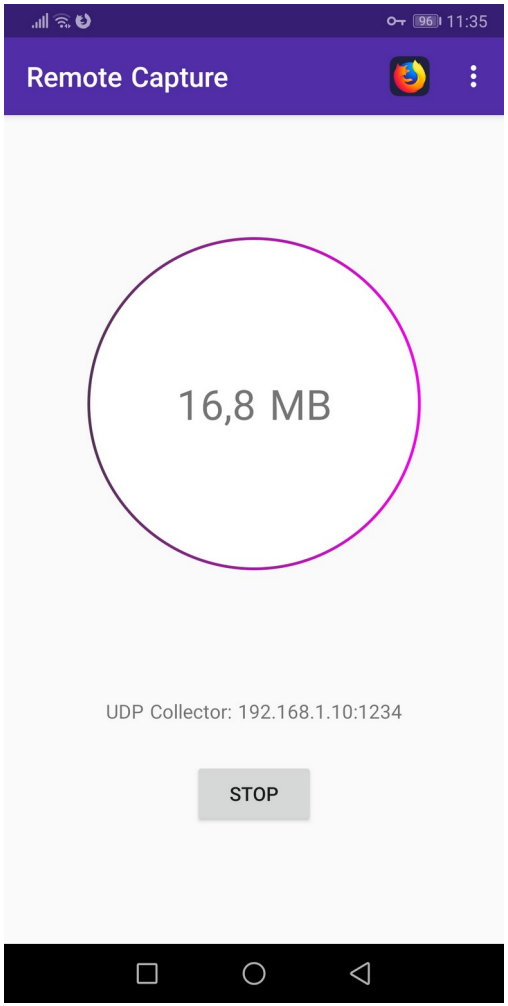
Cattura DNS

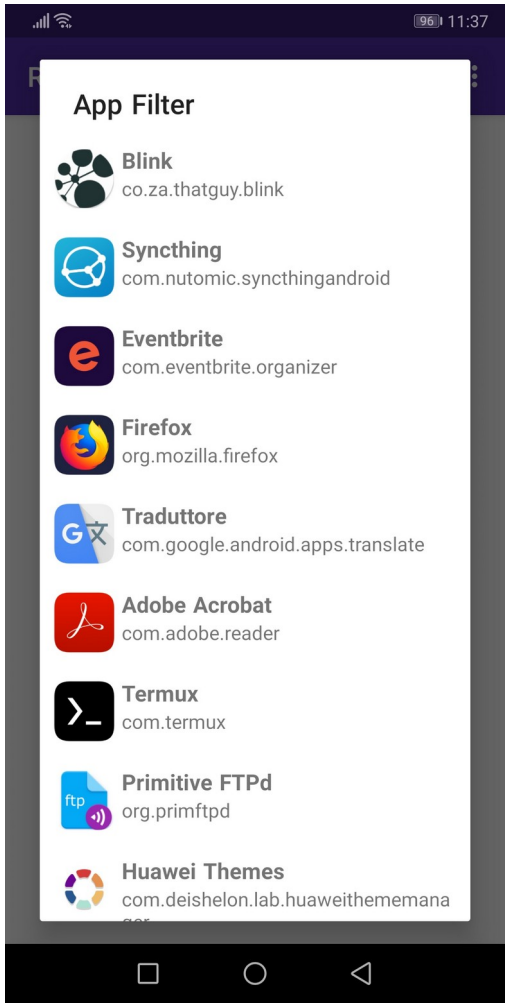
- **Il traffico DNS segue un percorso alternativo dentro al VPNService**
- **Per catturare questo traffico è necessario impostare tramite Builder.addDnsServer un server DNS sulla stessa sottorete dell'interfaccia VPN**
- **Es. l'interfaccia VPN ha IP 10.215.173.1/30 e il DNS server è 10.215.173.2**
- **Per non rompere le comunicazioni DNS, anche qui bisogna fare da proxy UDP con un DNS server**

Filtrare App

- **Data una connessione di rete, determinare lo “UID” dell’app corrispondente**
- **Da android Q in poi si può usare ConnectivityManager.getConnectionOwnerUid**
- **In versioni precedenti, è necessario parsare /proc/net/tcp e /proc/net/udp e cercare la connessione**
- **Problema DNS: molte app risolvono i nomi tramite android, risultano netd (UID 1051)**

RemoteCapture [1/2]





RemoteCapture [2/2]

- **Cattura il traffico delle app e lo invia su un socket UDP in formato PCAP**
- **Utilizza la libreria zdtun per la gestione dei socket**
- **Si può impostare un filtro per catturare solo il traffico di determinate app**
- **E’ possibile analizzare in tempo reale il traffico con tool quali wireshark e ntopng**
- **Supporta UDP e TCP, non ancora ICMP**

Collezionamento

wireshark:

- **socat -b 65535 - udp4-listen:1234 | wireshark -k -i -**

ntopng:

- **socat -b 65535 - udp4-listen:1234 | ntopng -m “10.215.173.0/24” -i -**

tcpdump:

- **socat -b 65535 - udp4-listen:1234 | tcpdump -r -**

Link Utili

https://github.com/emanuele-f/RemoteCapture

https://github.com/emanuele-f/zdtun

https://developer.android.com/reference/android/net/VpnService

https://github.com/WireGuard/WireGuard

Grazie per l’attenzione