Introduction

Privilege Escalation Vulnerability (CVE-2017-0490) is a Google Android vulnerability in Wi-Fi, which gives unauthorized applications permission to delete user data. It is considered as a privilege escalation vulnerability, because it gives elevated access to resources that are normally protected from user or an application. An unknown component of Wi-Fi is affected by the exploitation in which currently unknown input triggers the vulnerability. Since it requires local bypass to activate, it’s considered as a moderate vulnerability.

The vulnerability in question was discovered on the 7th of March 2017 and it was found that it affects the following Android versions: 7.1.1, 6.0.1, 7.0, and 6.0.

The technical details and the exploitation are unfortunately publicly unavailable, so that’s why this essay is mostly theoretical.
Exploitation

The vulnerability was discovered to be connected with the following commitment of Wifiinstaller.java:

```java
import android.content.Context;
import android.content.Intent;
import android.content.res.Resources;
import android.net.Uri;
import android.net.wifi.WifiConfiguration;
import android.net.wifi.WifiEnterpriseConfig;
import android.net.wifi.WifiManager;
import android.widget.TextView;
import android.widget.Toast;
import android.os.AsyncTask;
import android.content.SharedPreferences;

import java.security.PrivateKey;
import java.security.interfaces.RSAPrivateKey;

WifiManager wifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
WifiConfiguration wifiConfiguration = wifiManager.getConfiguredNetworks().get(0);
if (wifiConfiguration != null) {
    WifiEnterpriseConfig enterpriseConfig = wifiConfiguration.enterpriseConfig;
    dropFile(Uri.parse(enterpriseConfig.enterpriseConfig), getApplicationContext());
}
```

Although it is found that the vulnerability needs local interaction to work, it is still considered unpredictable for oblivious Google Android users. For example, if the attacker somehow convinces the user to use his (or her) malware, he (or she) instantly has access to modify the user’s files. It was discovered that the method `dropFile()` was the most dangerous as it lets the attacker delete files from the Android device – files that could be really sensitive to the owner of the device.
Solution

There is no information on possible countermeasures, but it is suggested that the affected object should be replaced with an alternative product instead of letting the file get deleted by the app, which I think is the proper solution to address the issue.

Theoretically, you can get rid of the issue by deleting the `dropFile()` method from the `WifiInstaller.java` class, which created the issue in the first place, but that’s just wiping the car crash under the carpet if you don’t come up with an alternative solution.

At the moment it is recommended to upgrade to the latest version of Android, which was released on March 5th, 2017.
References

2. https://vuldb.com/?id.97675
   a9b49c9185dad11b28b02e72124d95%5E%21/#F0