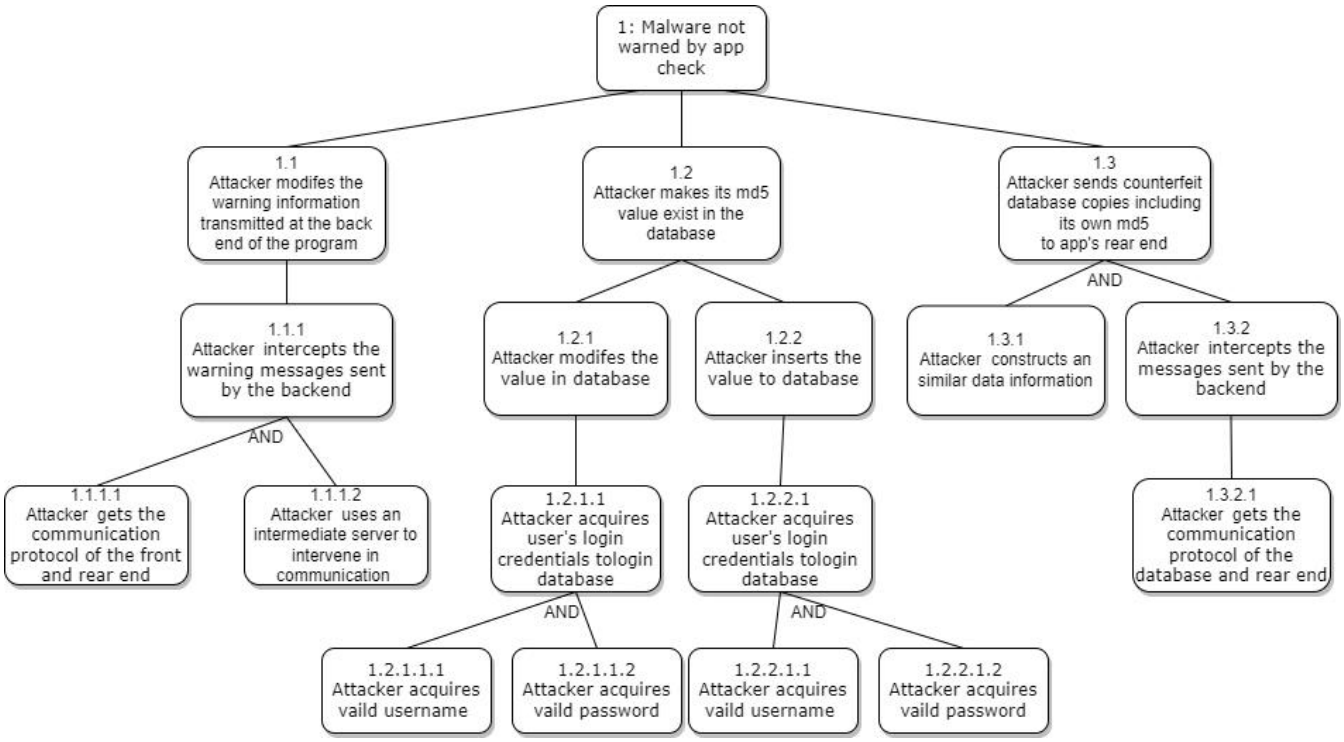

In this attack tree, you can find four ways to attack the target.

The principle extended from 1.2 is to attack the database. It shows two different ways to modify the database, but both need an effective database user information. Although there are many applications that are inadvertent to guard against when running, there are also many that do well in this area, such as preventing SQL injection or using a dedicated database server, which can be a good way to guard against database attacks.

1.3 It is used to block the link between the app program backend and the database, simulate a database, and make a false copy of the database including its own md5 value to the backend, so that it can recognize itself However, this largely depends on the way the database is linked to the backend. If a persistent link is used, it will generate an error response when an attacker intervenes. However, if the database is linked only once a day and the transcript is obtained, it can be intercepted in advance It is unclear how this app is linked to the database. It is linked once a day to obtain data; Or persistent links, fetching data once a day. Although success is

difficult, this method is more likely than 1.2.

1.1 It uses a server to simulate the interface between the front end and the back end of the app. It uses the communication protocol of the front end and the back end to build itself into an intermediate server. Any message will pass through here. This is a very mature and excellent hacking method At the same time, the communication protocol is also easy to obtain. You can obtain it by checking the message types, parameter types, ports and paths sent by the front and rear terminals respectively However, if this app attaches great importance to security, it will add symmetric or asymmetric encryption to the communication protocol, which is also a major difficulty. However, as an app, it needs to be maintained regularly. If an attacker has mastered the communication protocol of the front and rear ends before, he or she can intercept it when waiting for the app maintenance server to finish and start running again. Whether it is a security certificate or a symmetric or asymmetric encrypted public or private key, he can obtain it and use this pair of keys to communicate, making the back-end unable to detect it. This is the most likely attack route in my opinion.