Certificate pinning bypass.

Certificate pinning is a defence where an app only trusts a specific server certificate (or its public key), instead of trusting any certificate signed by a recognised authority. It is meant to stop an attacker from intercepting the app's encrypted traffic with a fraudulent certificate. A penetration tester needs to see that traffic to test the backend API, so bypassing pinning on a device they control is a routine part of a mobile engagement. The bypass does not mean pinning is worthless: it means pinning protects ordinary users, not an attacker who fully controls the device.

HOW IT WORKS

01 Why does a tester need to bypass it?

A mobile pentest has to test the backend API. To do that, the tester intercepts the app's traffic, reads the requests and responses, and probes the endpoints. The standard way to intercept HTTPS traffic in testing is to route it through an interception proxy with the tester's own certificate installed on the device.

Certificate pinning blocks exactly this: the app refuses the tester's certificate, just as it would refuse an attacker's. So before the tester can examine the API, they have to bypass pinning on the test device.

This is expected and routine. Bypassing pinning on a device the tester controls is not a finding against the app; it is a prerequisite for testing the part that matters most, the backend.

02 How is pinning bypassed?

Pinning is a check the app performs on the device, and any on-device check can be defeated when you control the device. The common approaches:

- Runtime hooking. Using an instrumentation toolkit like **Frida**, the tester hooks the function that performs the pinning check and forces it to accept the tester's certificate. This is the most common method.
- Repackaging. The tester modifies the app to remove or weaken the pinning logic, then repackages and reinstalls it.

SOURCES

[1] OWASP MASTG Network Communication Testing

[2] OWASP Certificate and Public Key Pinning

[3] OWASP MASVS Network requirements

- Network-layer tricks. Depending on how pinning is implemented, certain configurations can be worked around at the network or framework level.

The right method depends on how the app implements pinning. A well-implemented pin in native code is harder to bypass than one configured at the framework level, but a tester with a controlled device gets past both.

03 If it can be bypassed, is pinning worth implementing?

Yes, for the right apps, because the bypass only applies to an attacker who fully controls the device.

Think about who pinning actually protects against:

- An attacker on the same network as a victim (a malicious hotspot, a compromised router) trying to intercept a normal user's traffic. Pinning defeats this. The attacker does not control the victim's device, so they cannot hook the pinning check.
- An attacker who has the victim's unlocked device or who controls their own device for analysis. Pinning does not stop this, because they can bypass the check.

So pinning is a real defence against network-level interception of ordinary users, which is a genuine threat. It is not a defence against a determined attacker analysing the app on their own device, and it was never meant to be. For apps handling sensitive data (payments, health, finance), pinning is worth implementing as one layer.

04 What should developers take from this?

Two things.

First, implement pinning for sensitive apps, because it genuinely protects your real users from network-level interception. Use the platform's recommended configuration so it is correct and maintainable.

Second, do not treat pinning as a security boundary for your data. The backend must still authenticate and authorise every request

properly, because a determined attacker will see and replay your API traffic regardless of pinning. Pinning protects the channel for ordinary users; it does not protect the API from someone who has bypassed it. The API's own access controls are what protect the data.

Test your transport security and the API behind it.

securelayer7.net/learn/mobile-security/certificate-pinning-bypass

Open online