System Security (201700086)
Covert Channels #1 - Create a covert channel for an Android app

Æde Symen Hoekstra, Luigi Coniglio (Group 24)

May 2018

1 Introduction

We first started thinking about possible covert channels, preferably some kind of timestamp since these are easily to manipulate and hard to detect. We looked at the wireshark capture and the only timestamp we found in the TCP and HTTP layer was the `If-Modified-Since` header field in the HTTP layer. However, this header field did not seem present in the requests of the app. We then came up with idea of encoding bits in the TCP source port.

2 The Covert Channel

The covert channel we build is hidden in the TCP source port. When creating a socket, you can specify a local port to bind to. This can be used to create a covert channel, the way we used this, is that we bind to a portal that has a specific offset from the start of the safe TCP port range. For example if we want to transmit the pin 1234, we bind to the port 49152 + 1234 = 50386. From then on we bind to the first next on available port as a decoy, since this mimics the default behaviour.

3 Limitations

The major limitation of the proposed approach is that it mostly only works on local networks since NAT messes with the TCP source ports in many cases. This limitation cannot be easily overcome for the NAT cases, if you need to make a covert channel that is not affected by NAT, you should look for a different

method. This is of course not a problem on home networks or for example the Eduroam network.

The other major limitation is that we only can receive one pin number because we only know that the pin number is in the first request the server receives.