Android:
Bypassing HTTPS Security

Seminar Project

21 January 2020

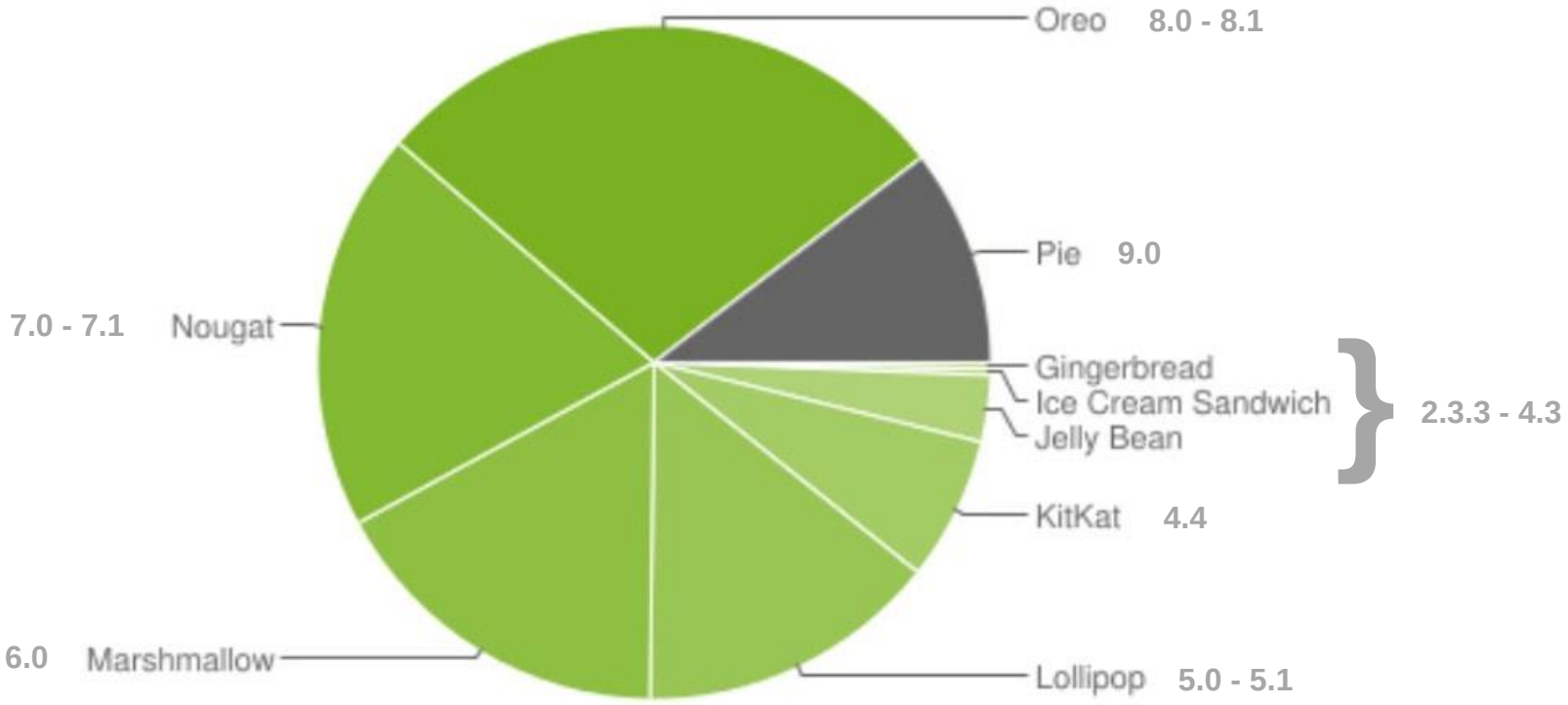
Christian Zürcher

Do you know...

*...**what data** your favourite apps **transmit**?*

The Problem

App inspections have become a non-trivial task:
Since Android 9 apps are defaulted to encrypted transport channels (HTTPS).



Data collected by Google in May 7, 2019

The Tool

FRIDA Modes

Injected

Embedded

Preloaded

```
Java.perform(function () {
    try {
        var okhttpClient = Java.use("okhttp3.OkHttpClient");
        okhttpClient.newCall.implementation = function (request) {
            var result = this.newCall(request);
            // do something with result
            return result;
        };
    }
    catch (error) {
        console.error(error);
    }
});
```

How to Get the Secured Data?

Step 01: Override certificate verification

removing any custom certificate validation code

Step 02: Inject certificate to certificate manager

inject our own certificate

... but the FRIDA API is unreliable for dynamic SSL code injection ☹

Another Problem

Multiple frameworks:

OkHttp
Retrofit
Volley
Picasso

... each with numerous versions using different methods and classes

Going Back to the Roots: HTTP(S) Traffic Interception

Why try to intercept when we can peek into the data source?

Same problems. ☹ ☹

Going Back to the Roots: HTTP(S) Traffic Interception

Why not inject into the native java code that is used everywhere?

Injected code is not executed. ☹ ☹ ☹

… But Some Things Do Work!

DEMO!

Conclusion

FRIDA is still in a rather early stage

Real world software environments are very complex

Future Work:
Create an automated way to retrieve HTTP(s) messages with the most used frameworks / versions

Retrieve the bodies of the messages