



Playing with your code: a new approach to avoid potential hackers from doing exactly this!

Hugues Thiebeauld

*Android Security Symposium
Vienna – 8th of March*



Mobile application goes to a hacker
How do I know if this will be tough for him



Mobile application tools



*focus on finding vulnerabilities,
virus, malware*

*we want to know if the
armour is strong*



Mobile application tools



*focus on finding vulnerabilities,
virus, malware*

*we want to know if the
armour is strong*



Solving a critical problem



Security protections are per nature “transparent”

How confident the protections have been effectively enabled?

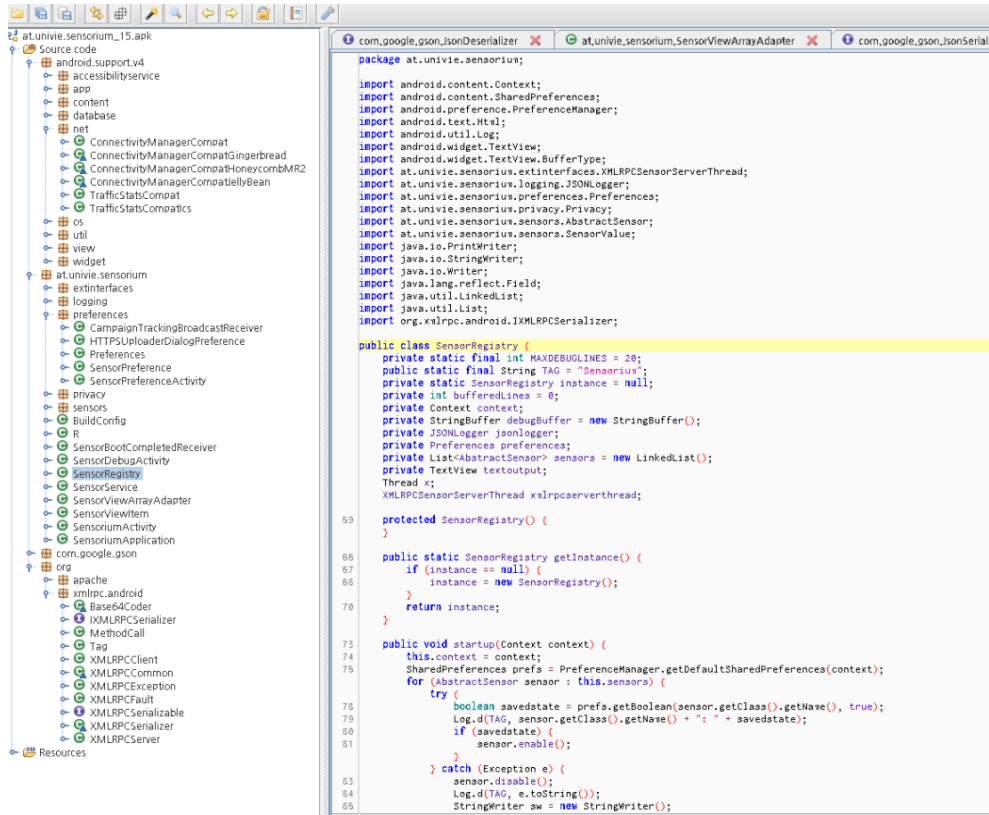


Protected mobile app

No other choice than hiring a security analyst to manually inspect

Are protections enabled at the right level?

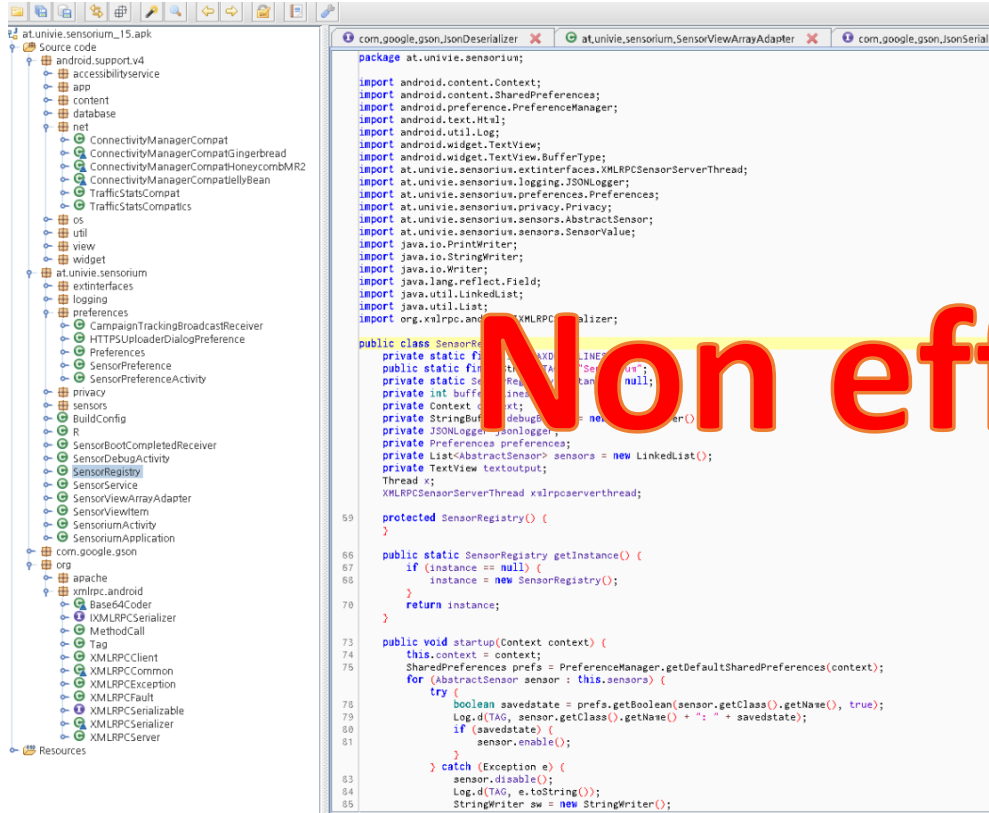
Manual inspection



How to know what to look for first?

*{Ctrl+F, Strings + Grep, Androguard, Smalisca, etc.} =
TEXTUAL, VERBOSE, SLOW,
TEDIOUS, BORING*

Manual inspection



How to know what to look for first?

Non effective !!

*{Curl + Fstrings + Grep,
Androguard, Smalisca, etc.} =
TEXTUAL, VERBOSE, SLOW,
TEDIOUS, BORING*

How many of you would like a tool...



Comprehensive

Visual

Customisable



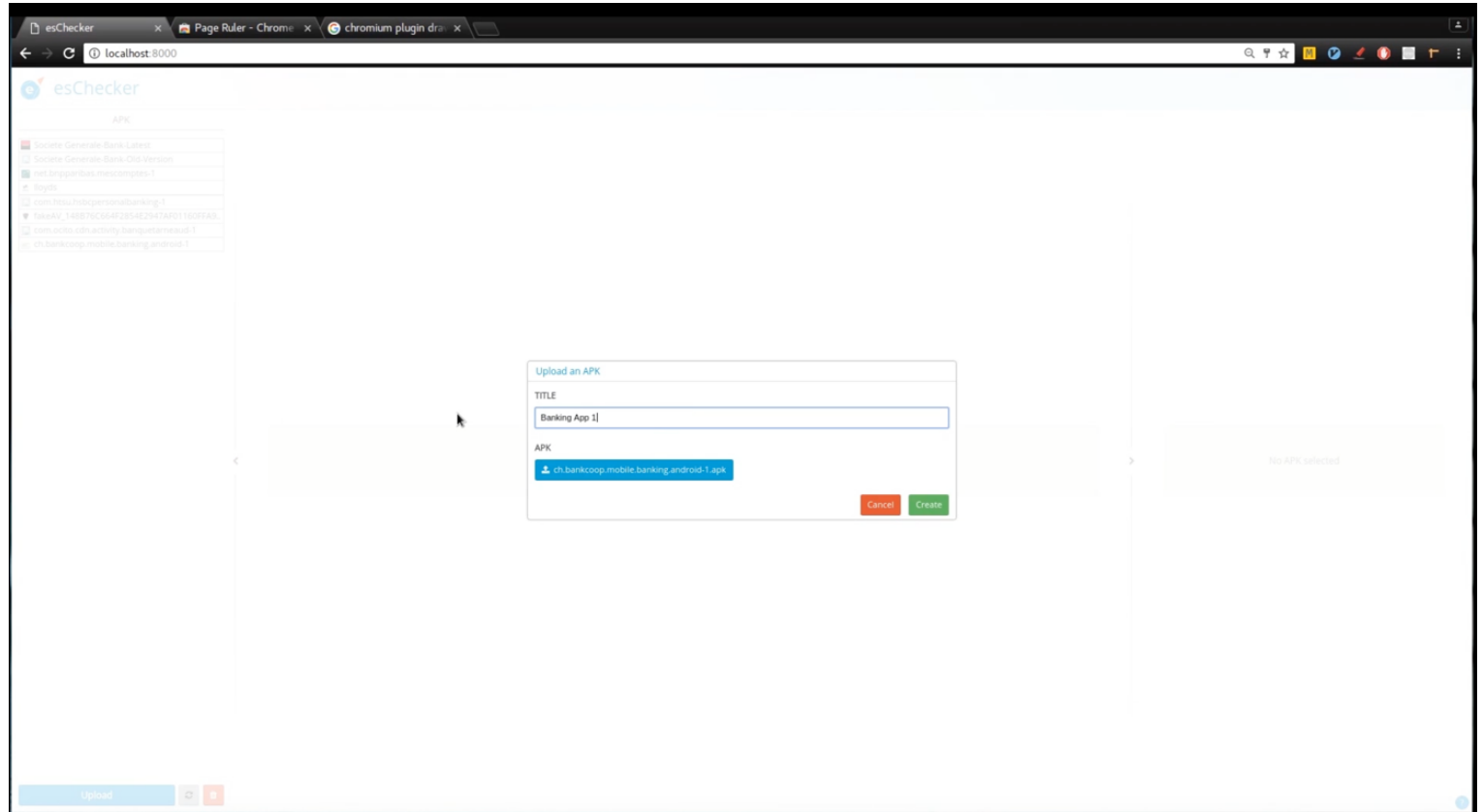
Easy to use

No black & white

Demo esChecker



- *Download your APK files*
- *The binary is disassembled and decompiled. Each piece of code is analysed against a set of heuristics*



Demo esChecker



- *As outcome of the analysis, a tile view is given.*
- *User has the opportunity to choose the APK*
- *User can browse into the file mapping view*
- *User can browse within the file tree*

The screenshot displays the esChecker web application interface. The browser address bar shows 'localhost:8000'. The application has a top navigation bar with 'Strings tab' and 'Code review tab'. Below this, there are tabs for 'Summary', 'Strings', and 'Editor'. The main content area is titled 'Banking App 1' and shows a 'Content map' with a 'Filter by type: any small dex elf resource other'. The central part of the interface is a large 'File mapping' view, which is a heatmap of red and orange squares. To the left of the main view is an 'Applications list' showing a list of APKs, with 'Banking App 1' selected. To the right is a 'File tree' showing a directory structure including 'original', 'unknown', 'lib', 'assets', 'small', 'res', 'apktool.yml', and 'AndroidManifest.xml'. At the bottom left, there is an 'Upload' button. At the bottom right, there is a table for 'Obfuscation summary' and a section for 'Compilers'.

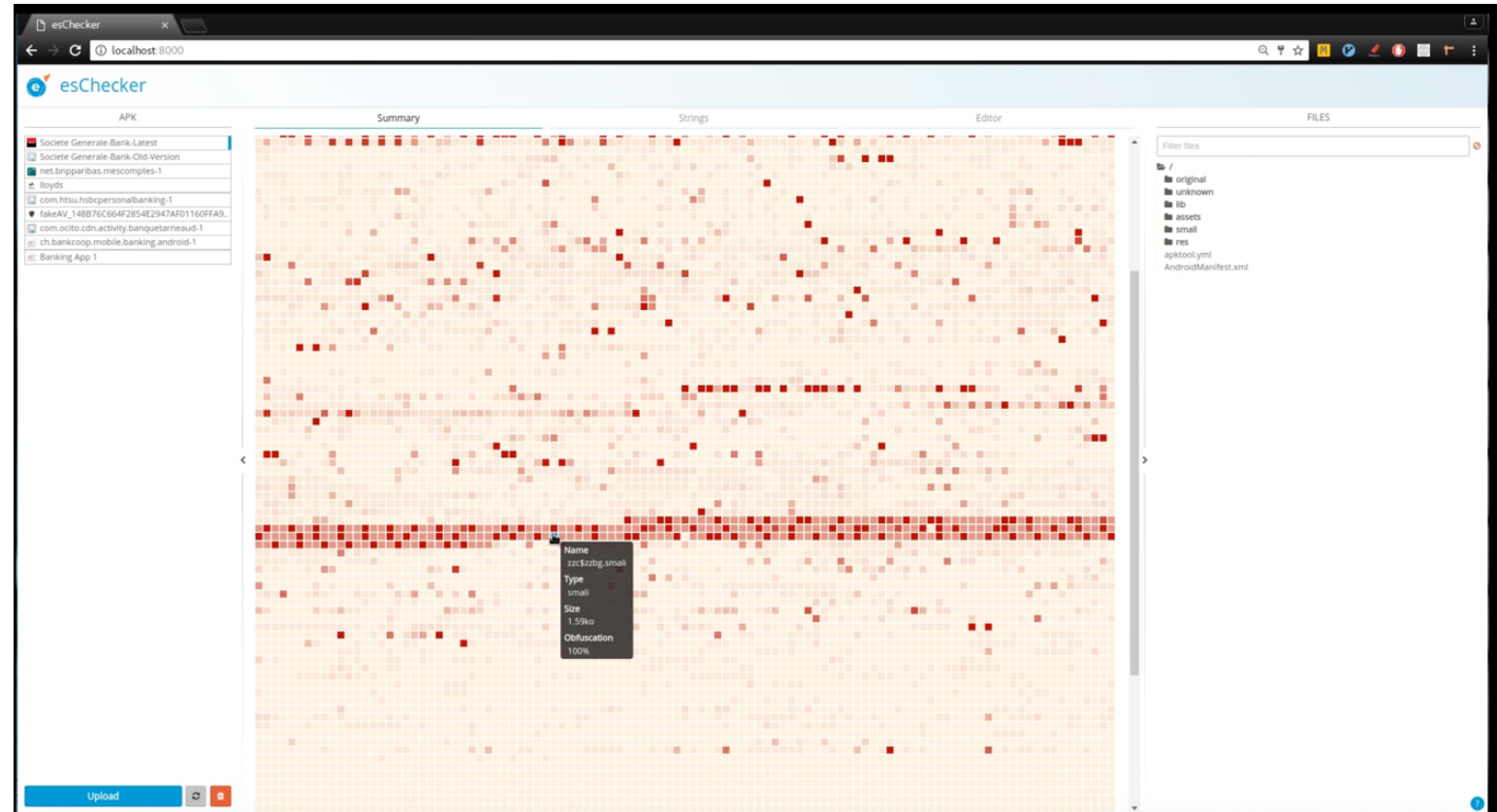
Protection	Status
Class obfuscation	✓
Code obfuscation	✓
Symbol obfuscation	✓
Crypto API	✓

Compilers
dexlib 1.x

Demo esChecker



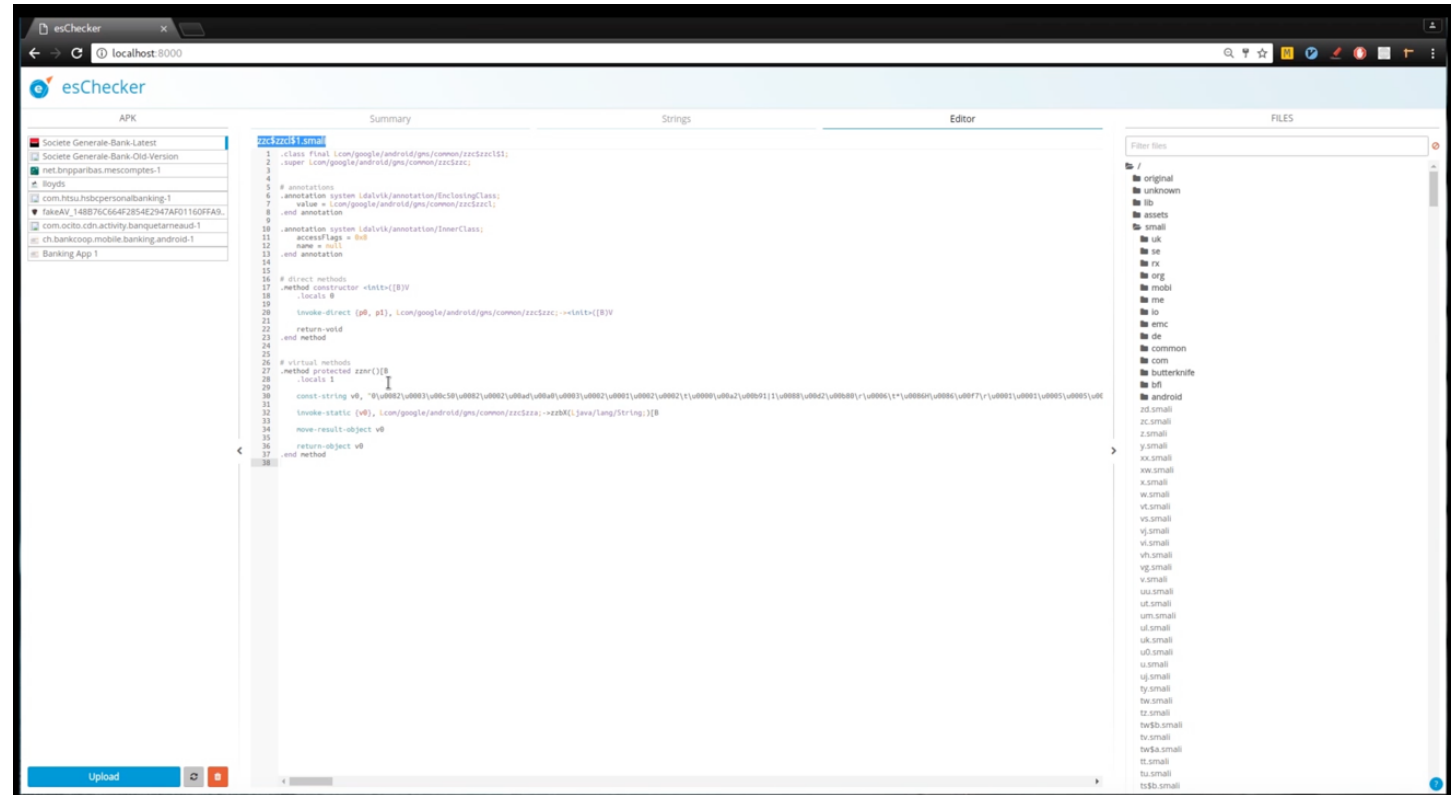
- *Each tile represents a piece of code*
- *An infobox displays information related to the file and shows how much obfuscation was found into it*
- *Different levels of colour helps to quickly see how much obfuscation was found*



Demo esChecker



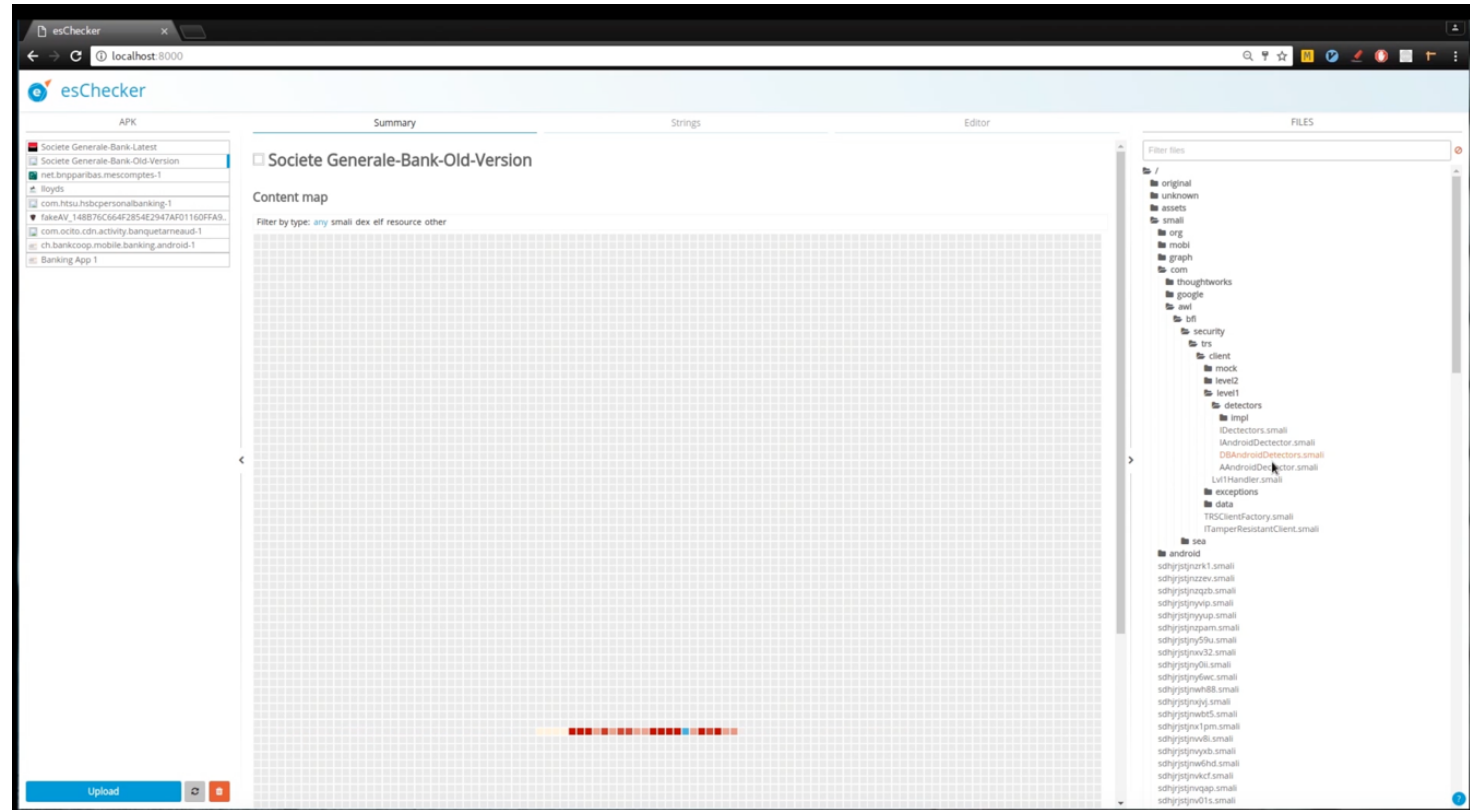
- User can jump from the graphical view to the editor view or the other way around
- Corresponding obfuscation can be spotted... here, non ASCII characters and renaming



Demo esChecker



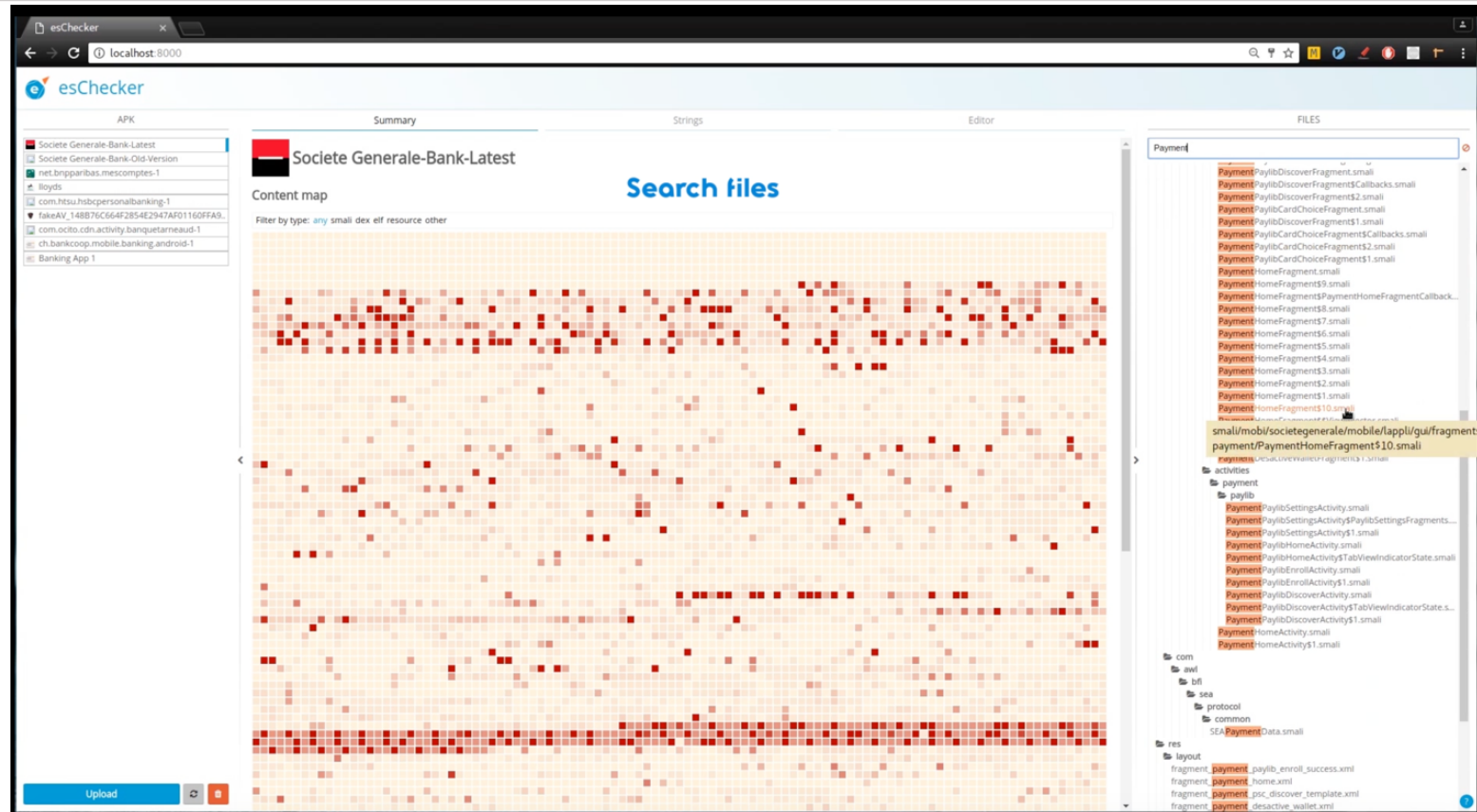
- *The view can be restricted to a given package by simply selecting the folder on the right.*
- *User can just move up and down within the folder structures.*



Demo esChecker



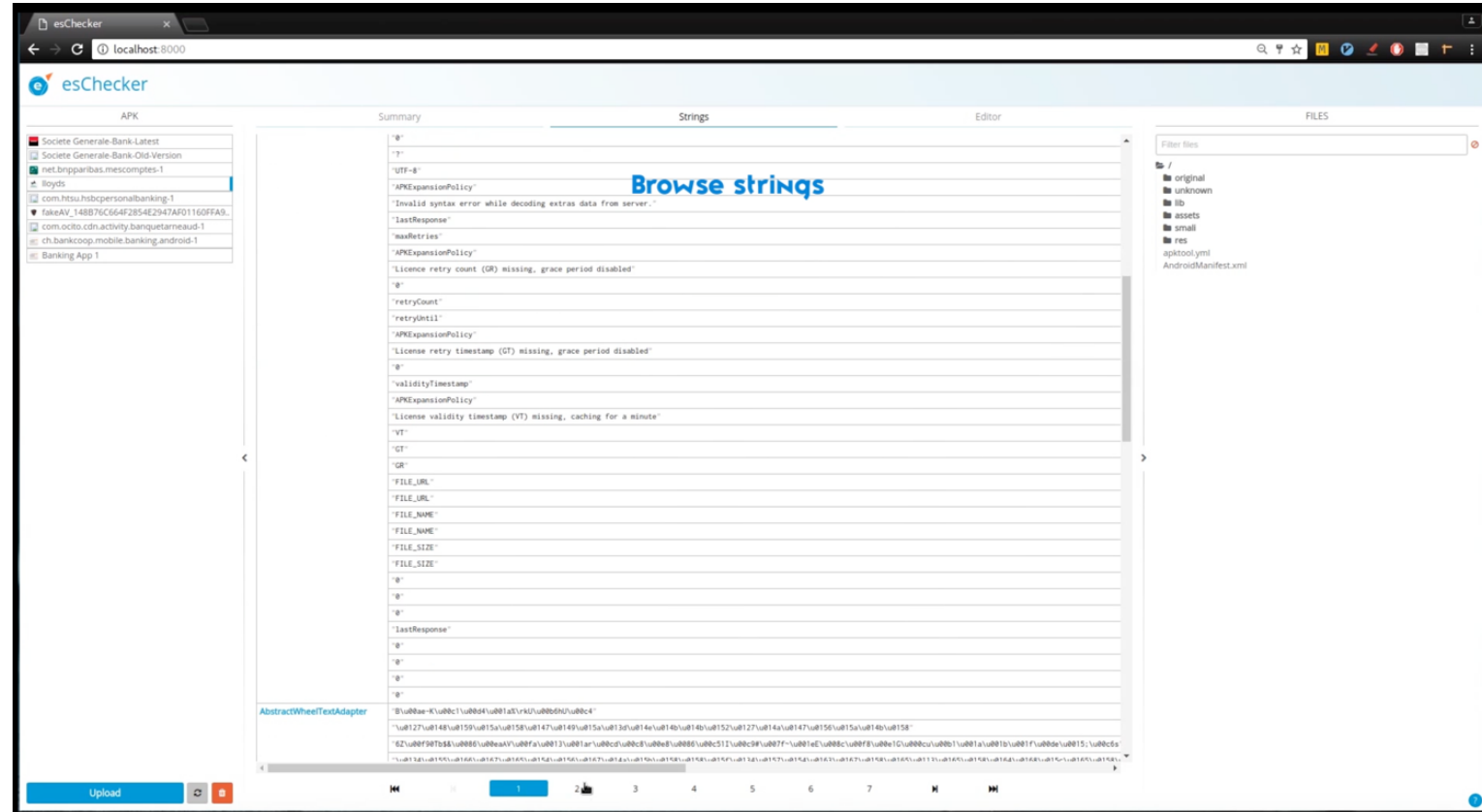
- *A search engine is available to collect all files with key words*
- *Helps to spot easily specific functions and locate them into the tile map.*



Demo esChecker



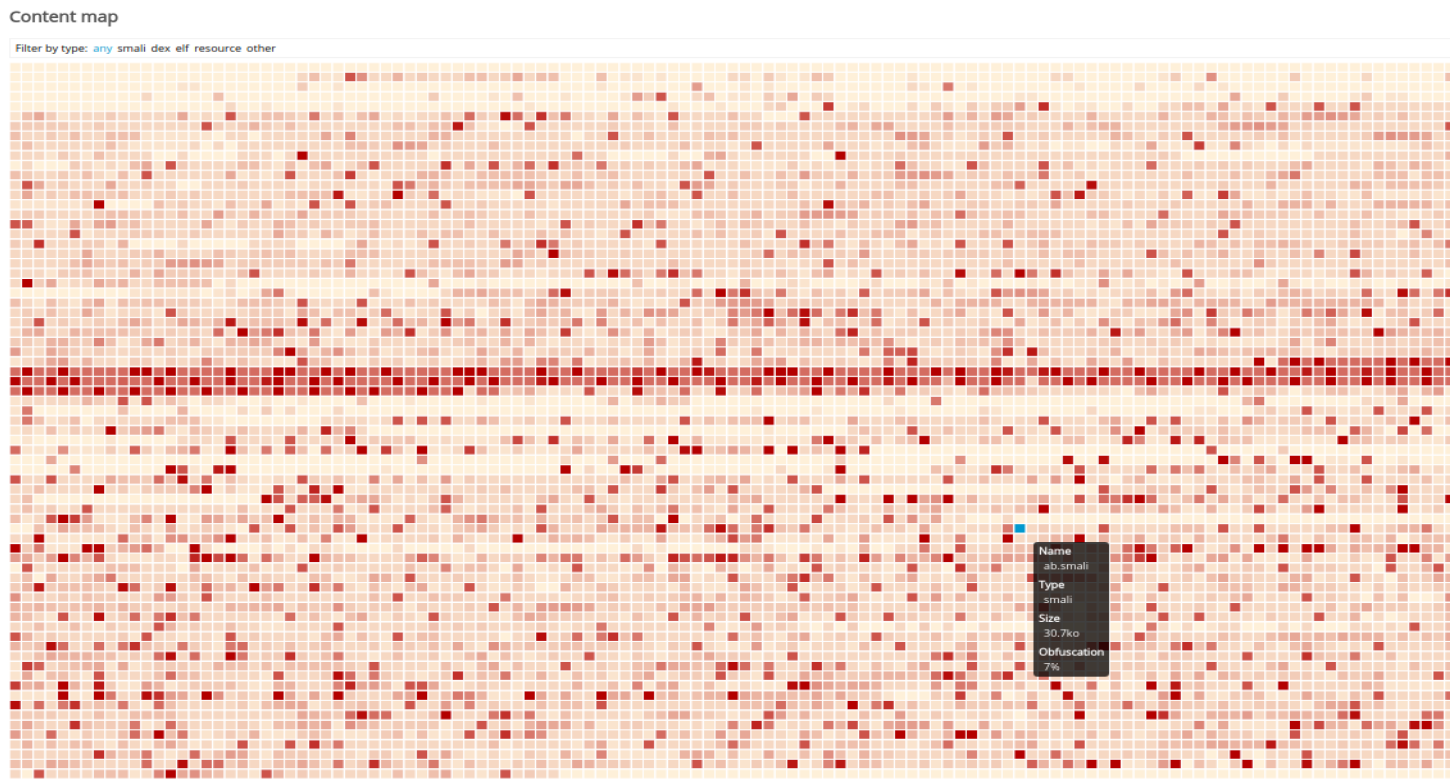
- *Strings are very useful to find his way into a binary*
- *A specific section of the tool is dedicated to strings*



A first example



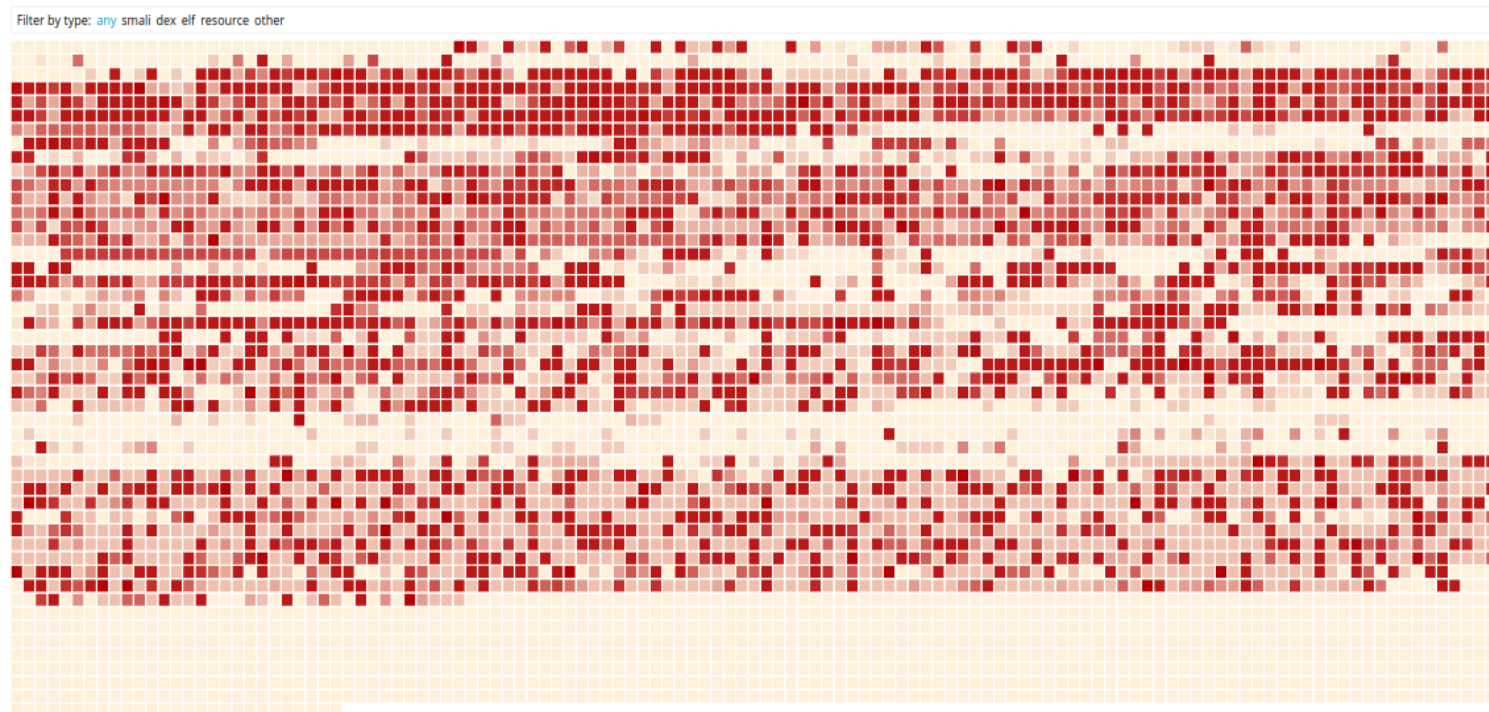
- *Weak obfuscation*
- *Strings are not obfuscated and remain way too much informative*
- *Control flow remains very easy to understand despite classes/methods/fields renaming*
- *GMS (in middle) looks more obfuscated than the rest of the app*



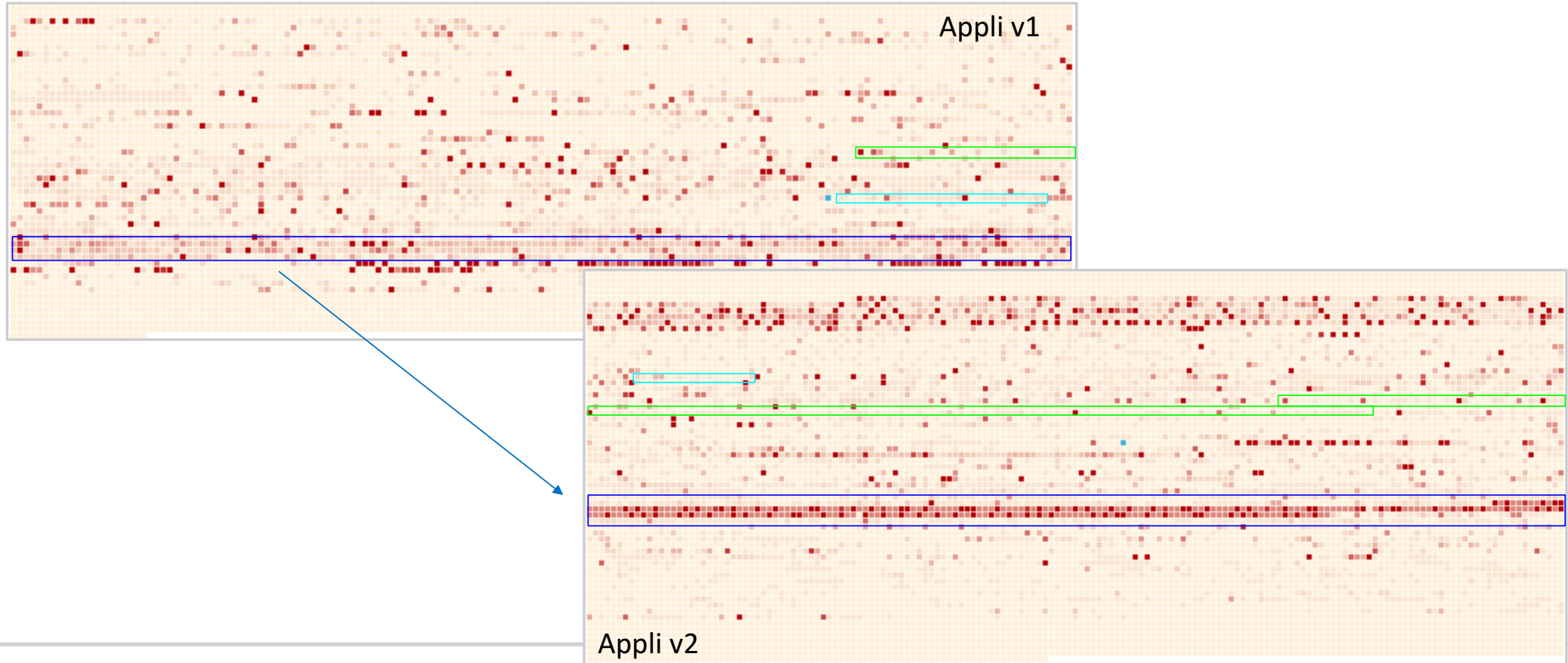
A second one...

- *Medium obfuscation*
- *Classes/Methods/Fields are renamed*
- *Usage of non-ascii characters for the renaming*
- *Some strings are obfuscated/encoded*
- *Control flows have been slightly obfuscated*

Content map



Delta version





**provides an easy
access to mobile app
protections**

Security Analysts

Save your time
Go straight to the point
Reach comprehensiveness

Everyone in the area

No need to be an expert to get a first insight
Manage the versioning and avoid regression
Use it as a benchmark
Reproducible process



Want to know more?

www.eshard.com



@eshardNews



eshard