Dr.-Ing. Mario Heiderich, Cure53
Rudolf Reusch Str. 33
D 10367 Berlin
cure53.de · mario@cure53.de

Pentest-Report Tornaj Pentest 02.2017

Cure53, Dr.-Ing. Mario Heiderich, Dipl.-Ing. Abraham Aranguren, BSc. Christopher Kean

Index

Introduction

This report documents the findings of a penetration test against the Tornaj Android mobile application, carried out in February 2017 by Cure53. The tests, which involved three members of the Cure53 team, yielded seven security-relevant discoveries.

A total of 6.5 working days dedicated to testing was needed to cover the entire scope of the assignment. As for the technical preparations and approach, the testers were given access to a debug APK and the application's sources. When the project was first initiated, the app's localized strings were first translated by the Cure53 in order to ease the tasks of debugging and testing. The application was then rebuilt and thoroughly tested.

Given the application's purpose, the test had a strong focus on information leakage, spoofed information and UI redressing vectors. The primary goal was to identify any issues that might lead the user to exposing sensitive information and making it available to remote and local attackers. As the tests progressed, the focus expanded beyond the usual mobile security test procedures and special attention was additionally placed on the ways the app communicates with the backend, notably a Firebase instance.

Anticipating the conclusions, it can be stated that the tests managed to determine a very limited number of issues. More importantly, the findings were generally of low severity and constrained impact. The app correctly exposes a very small attack surface and overall makes a very robust impression. This applies both to the main focus of the tests, and with reference to a very close look at the communications and protocol, where no serious issues could be spotted either. The process of assessing Tornaj was accompanied by good support from the app's developer and PM team. The assignment was therefore straightforward and efficient.

This report proceeds with discussing the few and far between issues found during testing. The findings are supplemented with recommendations regarding the possible fixing strategies. It is believed that the proper repairs can be deployed very quickly since no complex alterations are viewed as necessary.

Scope

- **Tornaj Mobile Android App**
  - Debug APK was shared with Cure53
  - Sources were shared with Cure53

Identified Vulnerabilities

The following sections list both vulnerabilities and implementation issues spotted during the testing period. Note that findings are listed in a chronological order rather than by their degree of severity and impact. The aforementioned severity rank is simply given in brackets following the title heading for each vulnerability. Each vulnerability is additionally given a unique identifier (e.g. *TOR-01-001*) for the purpose of facilitating any future follow-up correspondence.

TOR-01-001 App: External Links pointing to unsafe Clear-Text HTTP *(Low)*

It was discovered that the Tornaj app provides clear-text links in some areas, for example to the following URLs located in the *Experts Activity*.

URLs:

- http://www.khanehamn.org/
- http://vakilmoshaver.com
- http://www.boujari.com/
- http://persianlawyer.org/
- http://iraniansaed.org/

- http://vakilmoshaver.com/
- http://poyar.ir/
- http://sedayevakil.com/
- http://44951295.com/
- http://www.87132.ir/
- http://7585.ir/
- http://moshaver.behzisti.ir/
- http://123.behzisti.ir/modules/showframework.aspx

This defeats all possible TLS and Pinning protections, furthermore assisting attackers with the ability to modify network communications, who can as a result change the rendered page and use it for phishing purposes. Although these links are opened in mobile browsers, a page can be trivially crafted to resemble the app. In other words, the "fake" instance would appear nearly identical as far as visual appeal was concerned.

It is recommended to ensure that all links that a user could possibly tap on from the app employ TLS in a consistent manner.

TOR-01-002 App: Lack of Certificate Pinning allows for MitM Atacks *(Low)*

It was found that the Tornaj app fails to implement Pinning to protect TLS communications. This allows attackers with the ability to forge a certificate trusted by the Android CA store (i.e. most governments, some companies) to intercept and modify communications to the apps.

Under this premise, an attacker could alter the content invoked by the *Experts Activity* and leverage it for Phishing purposes by replacing the URLs mentioned in TOR-01-001.

It is recommended to implement Pinning to protect TLS communications. This should not only cover the content of the *Experts Activity*, but ideally also include any other sensitive domains that the application interacts with. For more information and Android/iOS examples of how to implement Pinning, please see the *OWASP Pinning Cheat Sheet*[1].

TOR-01-003 App: Complete Lack of Tapjacking Mitigations *(Medium)*

It was found that the Tornaj Android application fails to implement Tapjacking protections. This means that a malicious app can render an overlay, then launch the Tornaj app in the background, and, ultimately, attempt to trick a user into tapping on certain screen areas to perform actions on their behalf. While this happens, the Tornaj application will accept the taps, even though the overlay is being rendered on top of it.

[1] https://www.owasp.org/index.php/Pinning_Cheat_Sheet

Dr.-Ing. Mario Heiderich, Cure53
Rudolf Reusch Str. 33
D 10367 Berlin
cure53.de · mario@cure53.de

In the context of the Tornaj application, a malicious app could leverage this weakness to trigger an emergency call. This could result in financial damage for the user, effectively also signaling potential damage of the application's reputation. This issue was confirmed with a custom Proof-of-Concept (PoC), which was made available to the Tornaj team for verification purposes. Furthermore, a quick demo was also recorded[2].

Implementing the *filterTouchesWhenObscured*[3][4] attribute at the Android WebView level[5] is recommended to mitigate this issue. This will ensure that all taps from potentially malicious apps rendered on top are ignored, thus eradicating this attack vector. Ideally, this should be implemented in a base view that all other views inherit. As a consequence, the potential for human error with reference to leaving certain buttons unprotected would be reduced.

TOR-01-004 App: Info Leak through lack of Screenshot Protection *(Medium)*

It was found that the Tornaj app fails to implement Screenshot Capture mitigations. This allows apps with either *root* or *screen capture* permissions to access information that should otherwise only be available to the app. As an example, the *Settings* screen alone reveals the trusted emergency contacts, emergency messages, and user-address.

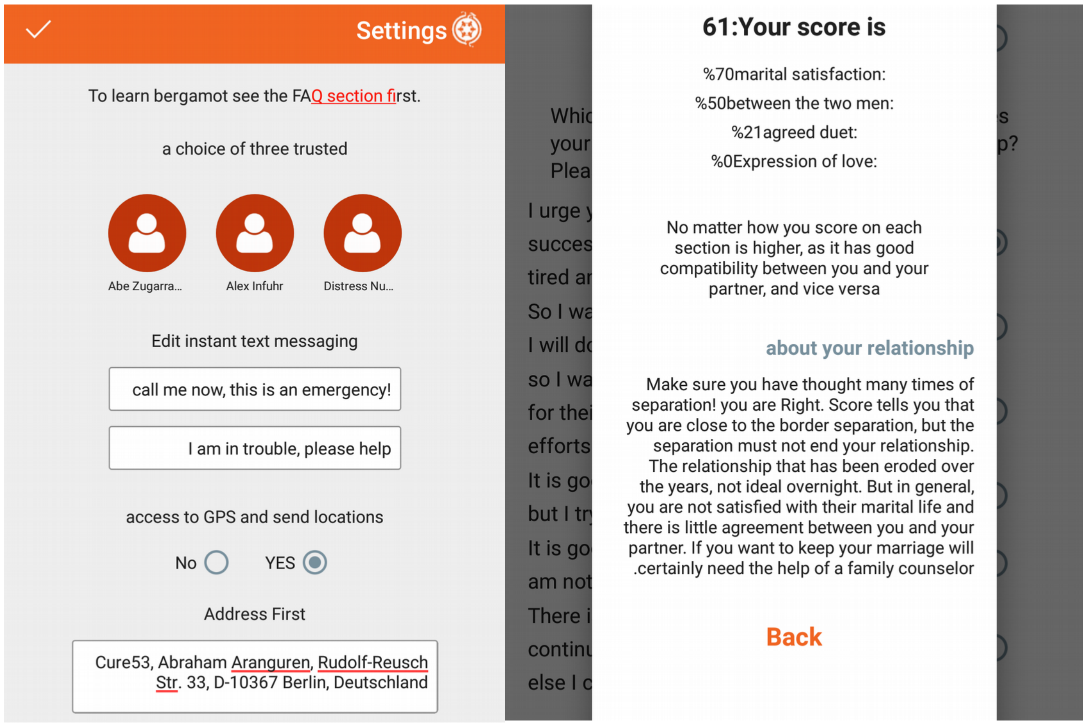
This issue can be confirmed when the following commands are run from a computer connected to either an Android emulator or a real phone, provided that the app is running.

Command:
```
adb shell screencap -p /mnt/sdcard/screenshot1.png
adb pull /mnt/sdcard/screenshot1.png
```

With the use of the above commands, the testing team managed to capture screenshots demonstrated next. This scenario illustrates that the sensitive information stored on the *Settings* or *Survey* screens can be acquired through screenshots:

[2] https://cure53.de/exchange/7859743432543/Tornaj_Tapjacking_PoC.zip
[3] http://developer.android.com/reference/an...ew.html#setFilterTouchesWhenObscured(boolean)
[4] http://developer.android.com/reference/android/vi...html#attr_android:filterTouchesWhenObscured
[5] https://cordova.apache.org/docs/en/latest/guide/platforms/android/webview.html

Dr.-Ing. Mario Heiderich, Cure53
Rudolf Reusch Str. 33
D 10367 Berlin
cure53.de · mario@cure53.de



To learn bergamot see the FAQ section first.

a choice of three trusted

Edit instant text messaging

call me now, this is an emergency!

I am in trouble, please help

access to GPS and send locations

No [✓ YES]

Address First

Cure53, Abraham Aranguren, Rudolf-Reusch Str. 33, D-10367 Berlin, Deutschland

Whic
your
Plea
I urge y
succes
tired ar
So I wa
I will do
so I wa
for thei
efforts
It is go
but I try
It is go
am not
There i
continu
else I c

61:Your score is

%70marital satisfaction:
%50between the two men:
%21agreed duet:
%0Expression of love:

No matter how you score on each section is higher, as it has good compatibility between you and your partner, and vice versa

about your relationship

Make sure you have thought many times of separation! you are Right. Score tells you that you are close to the border separation, but the separation must not end your relationship. The relationship that has been eroded over the years, not ideal overnight. But in general, you are not satisfied with their marital life and there is little agreement between you and your partner. If you want to keep your marriage will .certainly need the help of a family counselor

Fig.: Leakage of user trusted contacts, address, emergency messages and survey

It is advised to ensure that all WebViews have the Android *FLAG_SECURE* flag[6] set. This will guarantee that even the applications running with root privileges are unable to directly capture information displayed by the app on the screen.

[6] http://developer.android.com/reference/android/view/Display.html#FLAG_SECURE

Miscellaneous Issues

This section covers those noteworthy findings that did not lead to an exploit but might aid an attacker in achieving their malicious goals in the future. Most of these results are vulnerable code snippets that did not provide an easy way to be called. Conclusively, while a vulnerability is present, an exploit might not always be possible.

TOR-01-005 App: Leakage via debuggable and backup flags *(Info)*

It was found that the APK-build provided for testing has the backup and debug flags enabled. The backup flag could result in leakage in the rare situations when a user has USB debugging enabled. The debugging flag could result in information leakage via log messages and, ultimately, be conducive to an analysis by third-party apps.

Please note that this results in debugging information being logged not just by the app but also the invoked Android components. For example, the trusted contact's phone number and emergency text are leaked even though the app does not store these messages.

Logcat output:
```
02-13 12:53:24.856 D/baseband-sms(   98): newsms
02-13 12:53:24.856 D/baseband-sms(   98): sender:(N/A)
02-13 12:53:24.856 D/baseband-sms(   98): receiver:+123456789123
02-13 12:53:24.856 D/baseband-sms(   98): index:1/2
02-13 12:53:24.856 D/baseband-sms(   98): txt:'!call me now, this is an emergency'
```

This root cause for this issue can be observed in the *Android Manifest* pertaining to the app.

File:
AndroidManifest.xml

Affected Code:
```
<application android:allowBackup="true" android:debuggable="true"
android:icon="@mipmap/ic_launcher" android:label="@string/app_name"
android:name="org.netfreedompioneers.domvio.DomVio" android:supportsRtl="true"
android:theme="@style/AppTheme">
```

It is recommended to remove the *allowBackup* and *debuggable* flags from the *Android Manifest*. Ideally, the automated build process should be modified for this to happen by default. This will reduce the potential of leaving these flags "on" by mistake.

Dr.-Ing. Mario Heiderich, Cure53
Rudolf Reusch Str. 33
D 10367 Berlin
cure53.de · mario@cure53.de

TOR-01-006 App: Possible leakage via SD Card files (*Info*)

It was found that the Tornaj app saves PDF files on the SD Card. These files do not reveal personal information but a malicious attacker with physical access could extract the SD Card without having to unlock the phone. As a result, an adversary with access could figure out that a victim of domestic violence is attempting to gather information through the app. Please note that malicious apps on the phone could also collect similar data and send it to arbitrary third-party websites.

This issue can be verified when tapping on any of the "*export*" buttons on the "*Help and Legal Forms*" area:

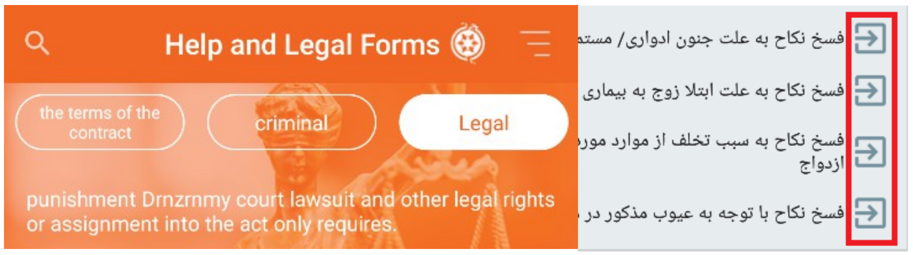

Help and Legal Forms
the terms of the contract
criminal
Legal
punishment Drnzrnmy court lawsuit and other legal rights or assignment into the act only requires.
فسخ نکاح به علت جنون ادواری/ مستد
فسخ نکاح به علت ابتلا زوج به بیماری
فسخ نکاح به سبب تخلف از موارد مورد ازدواج
فسخ نکاح با توجه به عیوب مذکور در

Fig.: Access to the PDF export functionality

This results in the Tornaj app saving known PDF files on the *Download* location of the SD Card:

File:
/storage/emulated/0/Download/07-2.pdf
/storage/emulated/0/Download/17-1.pdf
[...]

The root cause for this issue can be found on the location specified below.

File:
src/dom-vio-android-master/app/src/main/java/org/netfreedompioneers/domvio/DomVio.java

Affected Code:
```
public void openPdf(String fileName) {
    InputStream in = null;
    OutputStream out = null;
    String filePath = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS).getPath() + "/" + fileName;
    File outFile = new File(filePath);
```

```
           try {
               in = getAssets().open("pdfs/" + fileName);
               out = new FileOutputStream(outFile);
               copyFile(in, out);
[...]
           Intent intent = new Intent(Intent.ACTION_VIEW);
           intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
           intent.setDataAndType(Uri.parse("file://" + filePath),
"application/pdf");
           startActivity(intent);
       }
```

It is recommended to consider saving the files in question under less predictable filenames. Alternatively, the users should be confronted with an option to delete them. This way the user will have a chance to remove all sensitive hints from the SD Card. As it stands, the currently released data suggests to an attacker that the phone's user might be a potential domestic violence victim.

TOR-01-007 App: Firebase Service exported without permissions *(Info)*

It was found that the Tornaj app exports the *FirebaseInstanceIdService* without protecting it with appropriate permissions. This allows malicious third-party apps to invoke this service. The impact of this issue is limited since the service does not currently expose much functionality. However this could become a problem in a later release when the app makes more use of the *Firebase* features.

The problem can be observed in the *Android Manifest,* which shows that the service is exported to third-party apps and is not protected with permissions.

File:
AndroidManifest.xml

Affected Code:
```
<service android:exported="true"
android:name="com.google.firebase.iid.FirebaseInstanceIdService">
            <intent-filter android:priority="-500">
                <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
            </intent-filter>
</service>
```

Any third-party app can invoke the above service. In turn, it can be emulated with the *adb* command offered next.

Dr.-Ing. Mario Heiderich, Cure53
Rudolf Reusch Str. 33
D 10367 Berlin
cure53.de · mario@cure53.de

ADB Command:
```
adb shell am startservice -n
"org.netfreedompioneers.domvio/com.google.firebase.iid.FirebaseInstanceIdService
" -a "com.google.firebase.INSTANCE_ID_EVENT"
```

It is recommended to review the business need for maintaining this feature and, if possible, flag it for removal. Alternatively, the service could be protected with a permission and/or the exported flag could be set to false.

Conclusions

It is evident that the results of this February 2017 penetration test are very promising and positive for the Tornaj Android mobile application. Despite covering the entirety of the scope, the three testers of the Cure53 team were only able to highlight seven minor security issues.

The Tornaj app makes a good impression and keeps the attack surface small by simply focusing on the necessary features and avoiding any extra ornamentations which tend to contribute to problems if not causing them directly. Furthermore, the appropriate strategies and mechanisms are deployed to protect this already small surface, ultimately translating to very few minor recommendations.

The proposed solutions can only aid the application which presents itself as trustworthy with reference to keeping its security promises already. To strengthen and polish the application’s safety even further, it is recommended to implement additional hardening techniques mentioned in the tickets above, notably with reference to protecting users from falling victim to Tapjacking attacks and information leaks. The dominant expectation is that implementing the advised fixes signifies a rather easy task. Since there is no need for structural changes, it is assumed that the repairs can be attained quickly. The latter conclusion is reinforced by the fact that both the Development and the Project Management units did tremendously well when working on this project. In sum, the fact that nothing of high criticality or actual security relevance could be found is a clear testimony to the Tornaj Android mobile application being successful from a security standpoint.

Cure53 would like to thank Simin Kargar, Reza Ghazinouri and Arsalan Namdar for their excellent project coordination, support and assistance, both before and during this assignment. We would like to further express our gratitude to the Open Technology Fund in Washington D.C., USA, for generously funding this and other penetration test projects, as well as enabling us to publish the results.