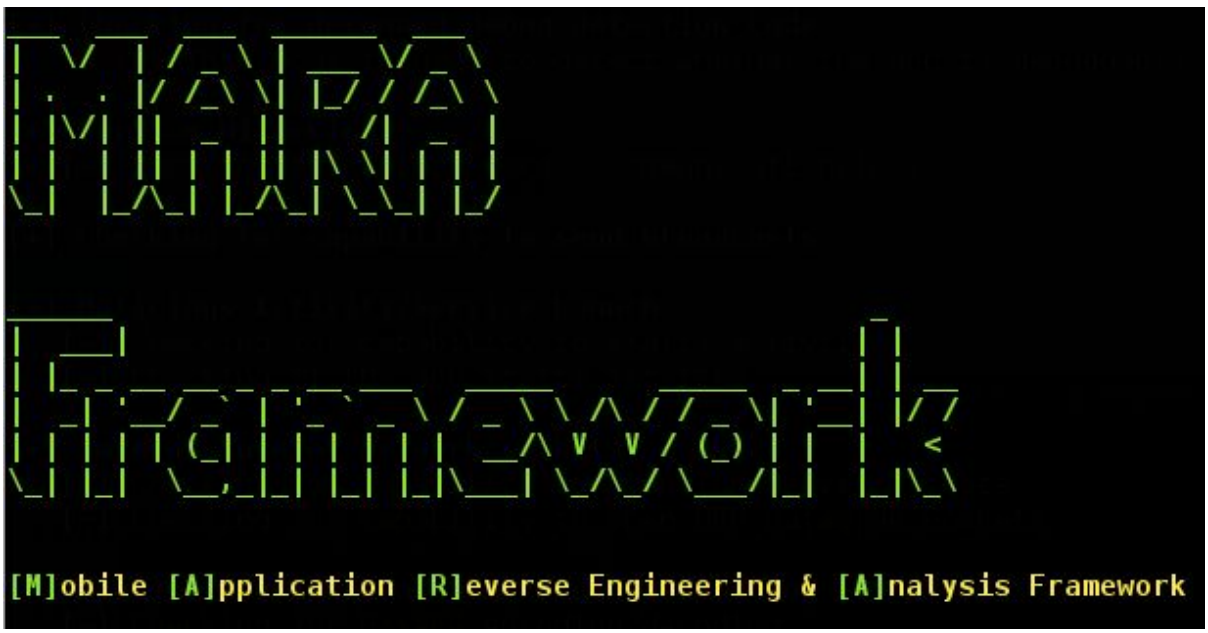
MARA Framework

Introduction

MARA is a **M**obile **A**pplication **R**everse engineering and **A**nalysis Framework. It is a tool that puts together commonly used mobile application reverse engineering and analysis tools, to assist in testing mobile applications against the OWASP mobile security threats. Its objective is to make this task easier and friendlier to mobile application developers and security professionals.

MARA is developed and maintained by @xtian_kisutsa and @iamckn. It is in its very early stages of development and there is a lot more to come, in line with our roadmap. Any contributions and suggestions to the tool will be highly appreciated.



Features supported

APK Reverse Engineering

- Disassembling Dalvik bytecode to smali bytecode via baksmali and apktool
- Disassembling Dalvik bytecode to java bytecode via enjarify
- Decompiling APK to Java source code via jadx

APK Deobfuscation

- APK deobfuscation via apk-deguard.com

APK Analysis

- Parsing smali files for analysis via smalisca
- Dump apk assets,libraries and resources
- Extracting certificate data via openssl
- Extract strings and app permissions via aapt
- Identify methods and classes via ClassyShark
- Scan for apk vulnerabilities via androbugs
- Analyze apk for potential malicious behaviour via androwarn
- Identify compilers, packers and obfuscators via APKiD
- Extract execution paths, IP addresses, URL, URI, emails via regex

APK Manifest Analysis

- Extract Intents
- Extract exported activities
- Extract receivers
- Extract exported receivers
- Extract Services
- Extract exported services
- Check if apk is debuggable
- Check if apk allows backups

- Check if apk allows sending of secret codes
- Check if apk can receive binary SMS

Domain Analysis

- Domain SSL scan via pyssltest and testssl
- Website fingerprinting via whatweb

Security Analysis

- Source code static analysis based on OWASP Top Mobile Top 10 and the OWASP Mobile Apps Checklist

Installing MARA on Linux/Nethunter

Downloading MARA

- git clone --recursive https://github.com/xtiankisutsa/MARA_Framework

Installing dependencies

MARA ships with a script that assists in downloading and installing the dependencies for each of the tools and components it ships with. Simply run the setup.sh script with sudo privileges and it will install them. To watch the MARA install guide video, please click on this link

Updating MARA

In order to make updating MARA easier, it now ships with an update script that once executed, will pull the most recent version from github and replace the files the ones stored locally. The script will not interfere with the data folder where the analysis files reside. Simply execute **./update.sh** and you are good to go. The update script will also run the new setup file that's been downloaded to ensure that the dependencies for the new tools are met.

After meeting all the requirements. If you run **./mara.sh --help** you should see the MARA help menu as shown below.

All the analysis data and file conversions are stored in the data folder i.e. **/MARA_Framework/data/file_name**. All the tools included in the Framework can be used standalone, they are all available in the tools folder i.e. **/MARA_Framework/tools**.

APK Deobfuscation

MARA facilitates the deobfuscation of APK files via apk-deguard.com. It's only files that are less than **16MB** that can be deobfucated. This is due to the restrictions by the mentioned site.

MARA ships with a stand alone deobfuscation script that could come in handy for analyzing individual APK files. Simply run **./deobfusctor.sh** and point it the APK you would like to deobfuscate. This feature requires an active internet connection.

SSL Scanner

MARA ships with a SSL scanner script that makes use of pyssltest and testssl. The domain SSL scanning component requires an active internet connection. The standalone SSL scanner can be run using the command **./ssl_scanner.sh** and follow the instructions displayed.

The findings from the scan are dumped in the domain scans folder i.e./MARA_Framework/data/domain_scans/. Please note that pyssltest scanner is intended to be used for scanning domains with SSL enabled. Do not scan IP addresses.

While analyzing APK files, MARA provides the option of scanning domains found in the apk using the above mentioned tools. This scan runs in the background and can be skipped. In the event the scan is performed, the user is required to tail the two log files i.e **pyssltest.log** and **testssl.log** in **/MARA_Framework/data/apk_name/analysis/static/ssl_scan/log/**

Smali control flow graphs

MARA is capable of generating control flow graphs from smali code. This is achieved by utilizing Smali-CFGs. The graph generation is optional and can be time consuming depending on the size of the android app being analyzed. The graphs are stored in two folders i.e. apktool_cfgand baksmali_cfg respectively in the location **/MARA_Framework/data/apk_name/smali/**

The graph generation runs in the background and you can check its completion by tailing the log files **apktool_cfg.log** and **baksmali_cfg.log** in the location mentioned above.

Progress monitoring

- The analysis data dumped by MARA will be located at **data/app_name** folder.
- You can monitor the APK deobfuscation process by tailing **data/app_name/source/deobfuscated/deobf.log**
- You can monitor the smali CFG generation by tailing these two files i.e.**data/app_name/smali/apktool_cfg.log** and **data/app_name/smali/baksmali_cfg.log**
- You can monitor the domain ssl scan by tailing these two log **filesdata/app_name/analysis/dynamic/ssl_scan/logs/pyssltest.log** and **data/app_name/analysis/dynamic/ssl_scan/logs/testssl.log**

To do list

MARA is still in the very early stages of development. We intend to work on the following features:

- Integrate dynamic mobile application analysis
- Rewrite the MARA Framework in python
- Integrate iOS, Blackberry and Windows phone application analysis
- Develop web panel to display data
- Include additional disassembly and analysis tools

Credits

These are the people who have assisted in ensuring the success of this tool's capabilities.

- Chrispus - @iamckn - https://www.ckn.io (co-developer)
- Ajin - @ajinabraham - Mobile Security Framework - MobSF
- Munir - @muntopia - http://munir.skilledsoft.com/
- Gabby - @_V1VI - https://thevivi.net
- AfricaHackOn Team - @AfricaHackon - http://africahackon.com

A lot of the tools integrated into MARA belong to their respective authors. We would like to thank each and every one of them for their amazing work. Developing MARA wouldn't be possible without your awesome tools :)

Contributors

- Charles - @icrackthecode - [https://github.com/icrackthecode]
- Ruby - @doobie

Disclaimer

MARA Framework is intended to be used for ethical hacking and educational purposes. Ensure consent is provided to allow reverse engineering of the various mobile applications as well as the scanning and interaction with the identified domains/IP addresses.

Licensing