Original Article
ISSN (Online): 3048-894X

Journal of Digital Security and Forensics
January-June 2025 2(1), 58–66

AUTOMATED DATA POPULATION FOR IOS DEVICES WITH AUTOPODMOBILE

Dirk Pawlaszczyk [1], Philipp Engler [1], Ronny Bodach [1], Michel Margaux [2], Ralf Zimmermann [2]

[1] Faculty of Computer Sciences, Hochschule Mittweida, University of Applied Sciences, Germany
[2] Central Office for Information Technology in the Security Sector (ZITIS), Germany

Received 15 March 2025
Accepted 21 April 2025
Published 23 May 2025

Corresponding Author

Dirk Pawlaszczyk, pawlaszc@hs-mittweida.de

DOI
10.29121/DigiSecForensics.v2.i1.2025.46

Funding: This research was financially supported by the European Anti-Cybercrime Technology Development Association - EACTDA (https://www.eactda.eu/).

ABSTRACT

Training investigators with realistic datasets is crucial for mobile forensics. However, until now, most training data has been generated manually, with very few automated solutions available. Particularly for iOS devices, there is currently no automated method for transferring data to phones. This article addresses this issue by introducing an approach that enables macros to be executed on the target device using simple onboard tools. Additionally, it presents the latest version of AutoPodMobile, a tool designed for automatic data population and injection. The paper discusses the results of a proof-of-concept implementation and concludes with an analysis of the findings, along with options for future enhancements.

Keywords: Dataset Creation, Data Population, IOS, Mobile Forensics, AutoPodMobile

1. INTRODUCTION

Mobile forensics is the discipline focused on recovering digital evidence from mobile devices under forensically sound conditions using accepted methods Hummert and Pawlaszczyk (2022). Forensic experts secure and analyse data using specialized forensic toolsets. The mobile forensics field is changing rapidly, more so than many other industries Pawlaszczyk (2022). The emergence of new cell phone models and a continuous stream of new applications each year exemplify this change. Due to the constantly changing conditions, the investigators must be

regularly trained and educated with the newest tools and technologies. Therefore, we need datasets as close to reality as possible. Unfortunately, to train police officers and forensic experts with realistic case data is often impossible. Even though incriminating data from real criminal cases are the most accurate source, these data are not suitable for training. Legal and privacy issues usually do not allow the use of data from real criminal cases. For this reason, manually generated datasets containing artificially generated data are generally used Grajeda et al. (2017), Gonçalves et al. (2022). But, the manual creation of these datasets is a highly complex and time-consuming task.

AutoPodMobile (APM) is a framework designed to facilitate the (semi-)automatic population of mobile phones with data Michel et al. (2022). This toolset can simulate the behaviour of cell phone users, generating data that mimics the quality of real-world datasets. The significant advantage is that you don't need to leave your home to create these datasets; all that is required is to connect the phones to a computer. Historically, APM has only supported data population for Android devices. Transferring data to iOS-based devices, such as iPhones and iPads, has been challenging due to various security restrictions. In this contribution, we introduce a novel approach for generating datasets and automatically transferring them to iOS devices. This functionality has been implemented in the AutoPodMobile toolset as a proof of concept, making APM the first tool of its kind to actively support iOS devices.

Before we delve into the specifics of the data transfer methodology, let us first take a brief look at the current state of research in this area.

2. RELATED WORK

In this section, we review related work on generating forensic datasets for mobile phones. The term "dataset" refers to a collection of digital objects that might be found on a mobile device. These datasets can be utilized for training, testing, or benchmarking various tools. In the field of digital forensics, complete hard disk images are often employed for these purposes. A reliable forensic copy is created from the phones, which was prepared with forensic data. Additionally, full file system extraction is a commonly used technique when a physical backup cannot be obtained due to encryption.

In the recent years, government bodies like *CTFF* Horsman (2019) and the *CFReDS project* NIST (2025) addressed the problem of generating datasets for mobile devices. They describe guidelines that provide specific recommendations and guidance on creating a forensic image for training purposes or tool testing.

The few publicly available datasets are mostly older and do not cover the current state of the art Grajeda et al. (2017). Beyond this, available forensic corpora generally offer only a small number of traces and are therefore not very realistic Gonçalves et al. (2022). The typical white noise is also absent in artificially generated data sets. In practice, only a very small portion of the data collected from a telephone is relevant to the case at hand. Unfortunately, most artificially generated data models, especially those involving manual input, do not take this fact into account.

What is missing is the availability of suitable tools for generating realistic datasets. There are barely a handful of contributions in this field. Examples of this include the *Forensic Test Image Generator* Visti et al. (2015), Du et al. (2021), *EviPlant Framework in* Scanlon et al. (2017) and the *For* Göbel et al. (2022). Unfortunately, none of these dataset synthesis frameworks allow the creation of datasets for mobile devices. At the time of writing, there are just two

projects, despite the AutoPodMobile toolset, that are dedicated to data generation for mobile phone forensics:

- The first example of a data injection tool is FADE Delgado et al. (2021). This proof-of-concept framework enables the injection of static traces into an Android-based emulator. The tool utilizes the Android Debug Bridge (ADB) for data transfer. The developers of FADE modify files and database entries within an Android virtual machine to mimic user-generated content as closely as possible. It is unclear whether the tool also supports the creation of physical devices. The dataset results were tested exclusively with the Autopsy forensics toolkit.
- In the study by Demmel et al. (2024), a prototype implementation based on the standard Android emulation environment is introduced, utilizing tools from the user interface testing community. The authors employ gesture control to mimic realistic user behaviour. Even our APM framework makes use of UI-Automator functionalities in some instances to simulate user keystrokes. However, this approach does have some limitations. If the app vendor changes the names or arrangement of its user interface elements, the method will fail and require adjustments. AutoPodMobile relies heavily on APIs provided by app vendors, which are generally stable and not frequently altered. This stability is crucial, as vendors need to maintain backward compatibility with their APIs.

All frameworks discussed in the study are designed to support Android devices. Currently, there is no solution available for iOS devices.

3. METHODOLOGY

We now want to describe a unique approach to enable access to an iOS-device. The operating system vendor (Apple) only offers limited external access to its devices due to security restrictions. For the development of apps with XCode/Swift**,** the standard development IDE for IOS, the developer can communicate with the device. As Apple has recognized the need to automate certain activities for app testing, XCode 15 provides a reworked and improved access interface called *devicectl*. APM makes use of this interface. The device controller component makes it possible to access a connected iPhone or iPad and control it remotely using special commands. Another significant innovation from Apple was the introduction of the *Shortcuts* app for iOS. The *Shortcuts* app from Apple is a powerful automation tool that lets you create custom macros for tasks you do regularly on your iOS, iPadOS, or macOS devices. Key features of the Shortcuts app include:

- Creating automated workflows that combine multiple actions across apps
- Running shortcuts via Siri, the widget on your home screen, or automatically based on triggers
- Sharing shortcuts with others and downloading pre-made shortcuts from the Shortcuts Gallery
- Integration with hundreds of apps and system functions

The app comes pre-installed on modern Apple devices and has replaced the previous Workflow app that Apple acquired. Shortcuts has been available on IOS-based devices via a corresponding app since 2019. Using this interface, we are able to simulate different user activities on the cell phone. This includes sending messages, creating contacts and events and even making phone calls. A

corresponding macro (called a *shortcut*) is provided for each of these individual activities.

In order to successfully access an iPhone or iPad, a number of technical requirements must be met. To run a shortcut remotely from a Mac, you need to have Xcode version 15.3 or higher installed, along with the command line tools. To simplify access to shortcuts, we have developed a Python class called *IOSCore*. This class serves as a wrapper, providing several methods that facilitate access to an iOS device running version 17.0 or higher. We use this class to trigger specific shortcuts on the device. The command line tool `devicectl` is utilized to connect to the iPhone via IOSCore. `devicectl` is a device controller that enables remote access to mobile devices.

At this point it is important to understand that calling or executing a macro via devicectl is not an officially supported function. However, we have found a way to do this indirectly. One of the standard functions offered by devicectl is to open a website with the built-in Safari browser. All you have to do is enter a valid URL and the corresponding website will be displayed. But you can also start a shortcut via the browser as an alternative to classic websites. You can launch the app to a particular shortcut in your collection. Open a URL with the following structure: *shortcuts://open-shortcut?name=[name]*, and provide the name of the shortcut in the name parameter. We use this URL schemes to trigger a shortcut. A URL scheme is the component of a link that determines the type of application with which a URL is opened. Many standards iOS apps support such URL schemes. A typical call would look like this, for example:

```
$> xcrun devicectl device process launch
    --device F60AXX3-DDSFDF-3300
    --payload-url
shortcuts://run-shortcut?name=APod_XXX&input=
{"param1":"value1","param2":"value2"}
    com.apple.mobilesafari
```

Table 1

Table 1 Elements of a Shortcut URL

shortcuts://	call a shortcut from shortcuts app
run-shortcut?name=APod_XXX	search for a shortcut with the name
&input={"param1":"value1","param2":"value2"}	handover all input parameters for this shortcut

Figure 1

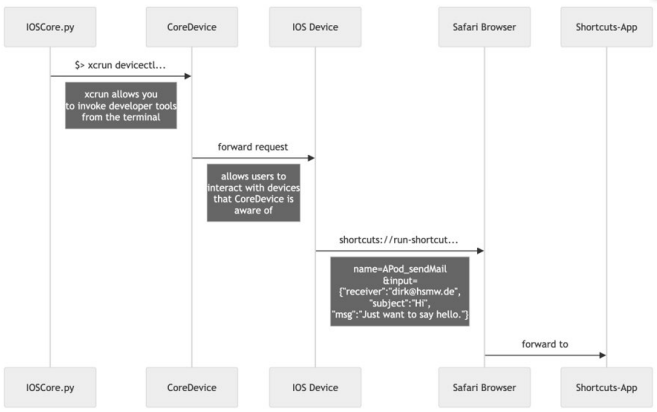

Figure 1 The Command Chain for Remote Execution of a Shortcut (Example)

The command *xcrun* is executed from the command line interface (cli). Xcrun is a tool that's part of the developer tools for macOS. It can query information about SDKs and their included tools. It's also used to run binaries, like in this case. The binary *devicectl* is called. The command with the *'process* ***launch'*** **option** enabled starts a process on the remote device. In our example the process to start is the Safari browser (*com.apple.mobilesafari*). The URL to be opened in Safari is specified via the ***--payload-url*** parameter. In our case, this is a special URL (see Table 1 Elements of a shortcut URL). In terms of functionality, it is reminiscent of a remote procedure call.

The parameters for calling the shortcut are transferred as a JSON list consisting of one or more key-value pairs. If the corresponding shortcut is installed on the device, it will be executed automatically. The conceptual structure and the interaction of the individual elements in this process can be seen in the following sequence diagram Figure 1.

Figure 2

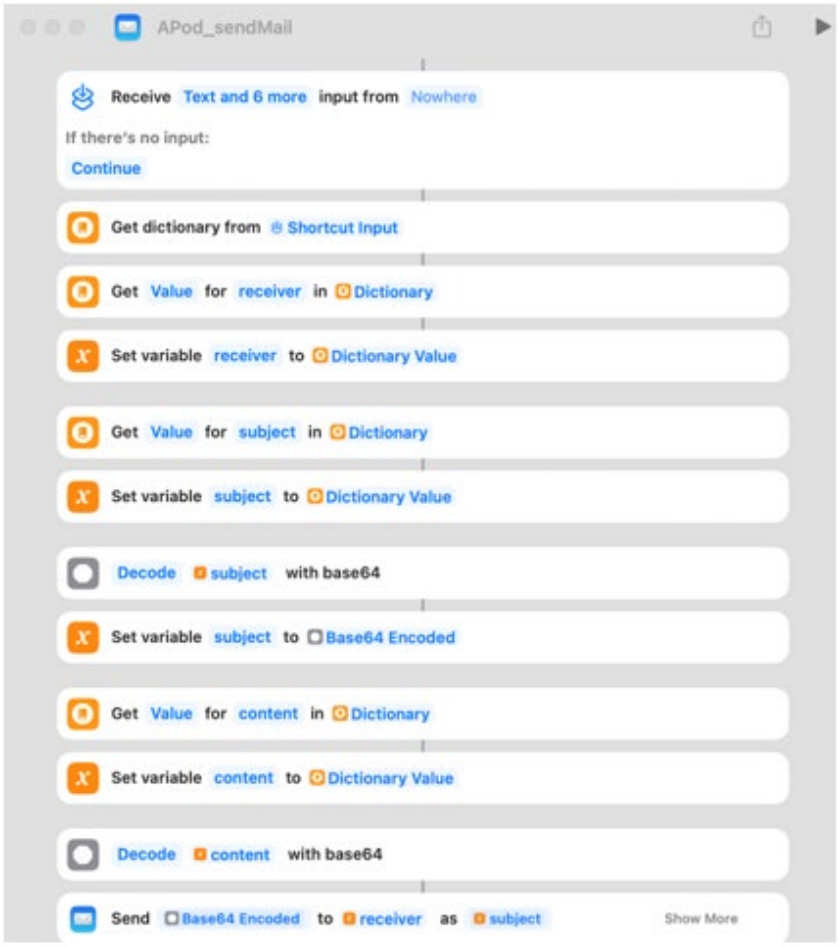

Figure 2 A shortcut Macro for Sending Mails (Example)

Using this method, you can execute any macro on your connected iOS device. All you need to do is define the shortcut you want to execute, and you're all set. The parameters that are transferred can be read one by one from the input. For example, Figure 2 illustrates what a macro for sending emails would look like.

4. IMPLEMENTATION – PROOF OF CONCEPT

The concept described in the last section was successfully implemented in APM-toolset. This makes APM the world's only tool for data population of iOS devices that currently exists. The version of the program originally presented in Michel et al. (2022)was controlled via the command line only. In the latest version a graphical user interface has been added see Figure 3 improve usability of the application.

Figure 3

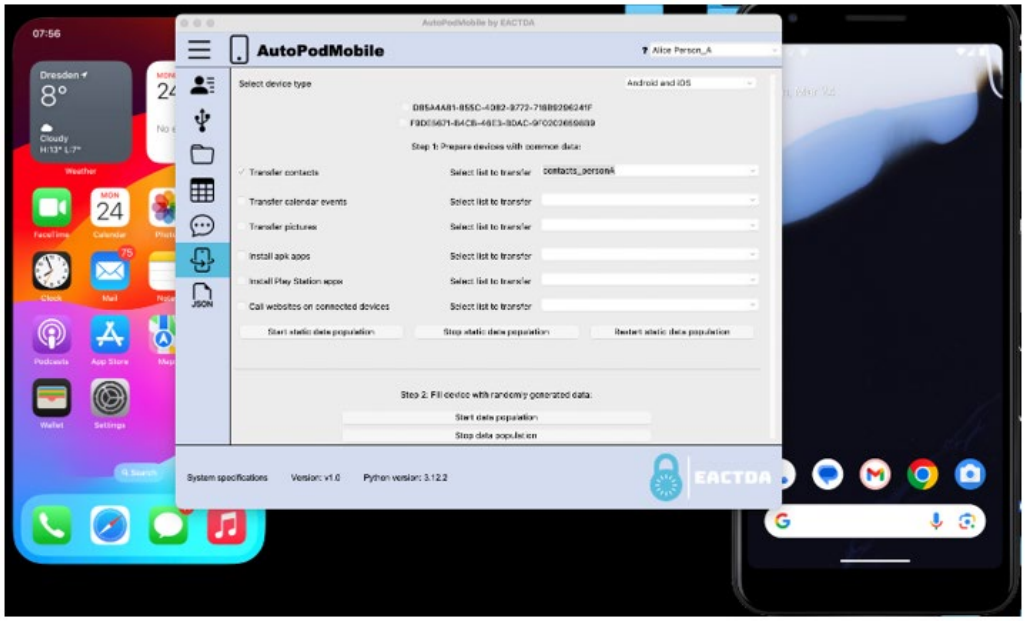

Figure 3 The AutoPodMobile User Interface with Two Mobile Phones in Background

APM is a dataset creation tool with automatic support for data injection on the attached devices. Before we populate a device with data, we need to model the data to transfer. With APM the user always creates a model from the perspective of a criminal case. When modelling a case, the actors and their mobile devices are involved and the data is assigned to them. The creation and transfer of a new dataset is divided into 6 simple steps see Figure 4.

Figure 4

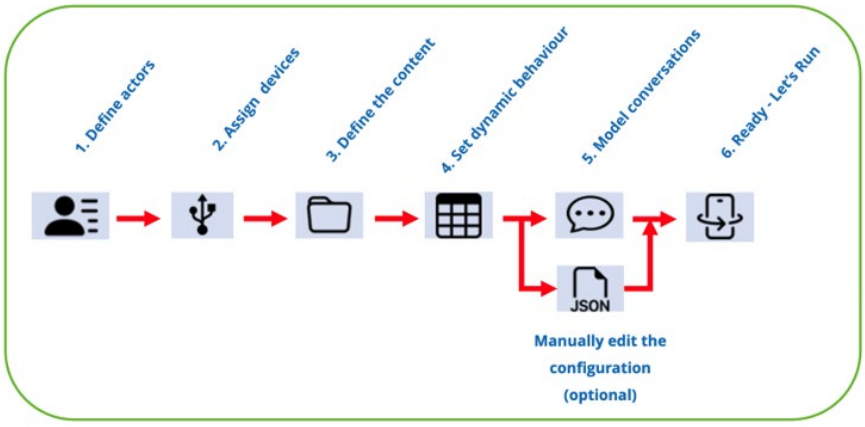

Figure 4 The APM Dataset Creation Process

Normally, the actors involved are defined first. Specific smart phones are then assigned to them in the second step. Then the actual data modelling is done in the 3rd step. The phones can be loaded with static data such as contacts, calendar events or photos Figure 5. Mails and other messages can also be defined. Dynamic user behaviour is defined using a series of finite-state machines Michel et al. (2022). The messages are not simply written to a database. Instead, the communication behaviour of real telephone users is emulated in real-time. The state machines manage the transitions between different user activities on the phone.

Figure 5

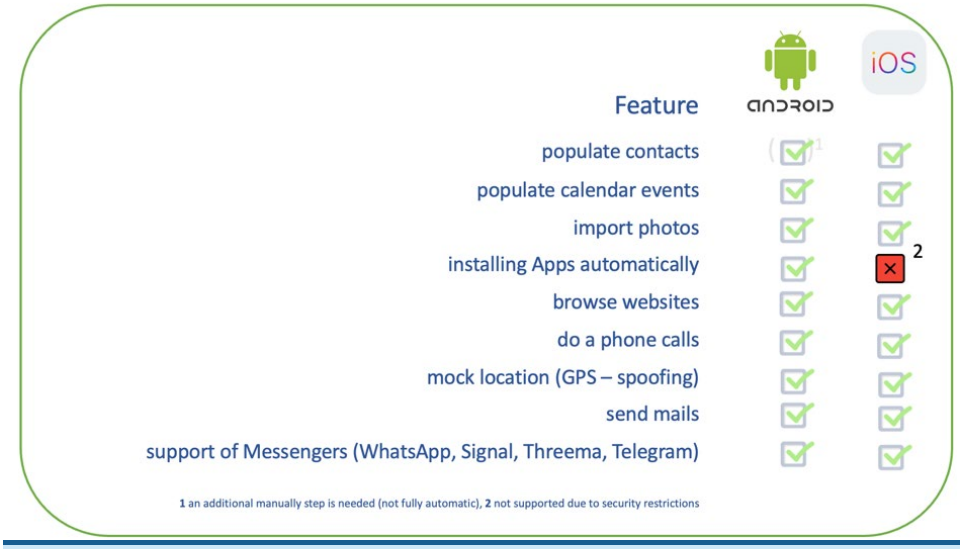

Figure 5 Feature List of Autopodmobile for Data Injection Operations

With the latest version of APM support for iPhones was integrated directly into the interface. iOS devices can now be populated with data in a similar way to Android devices. From a modelling perspective, it doesn't matter whether I want to provide data for an Android phone or an iPhone. The data acquisition and description of the data to be generated is completely independent of the device to be populated. A list of all user data currently supported by APM can be viewed in Figure 5. Apart from the automatic installation of apps, all functions that are offered for Android are also supported for iOS.

5. CONCLUSIONS AND FUTURE WORK

The automatic generation of datasets for training purposes is a time-consuming and highly challenging task. Currently, this process is primarily performed manually. There are only a few toolsets available for automatic dataset creation, and they primarily focus on data population for Android devices. To access these devices, these solutions utilize the Android Debugging Bridge (adb). For a long time, it seemed unlikely that similar functionality could be offered for iOS devices due to security restrictions. In this article, we present an approach that enables automatic data input for iOS devices. Our contribution provides a detailed explanation of how this can be accomplished. The solution revolves around the combination of the devicectl interface and the Shortcuts app -both of which are standard tools on MacOS and iOS. By integrating these two tools, we enable the remote execution of macros on iOS. We successfully implemented a proof-of-concept solution within the

AutoPodMobile toolset. To the best of our knowledge, it is currently the only program that offers such an option for iOS devices.

The introduction of further shortcuts to support a wide range of functions is planned for the future version of the program. AI-supported generation of data sets is also planned.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Delgado, A. A. C., Glisson, W. B., Grispos, G., & Choo, K. R. (2021). FADE: A Forensic Image Generator for Android Device Education. WIREs Forensic Science, 4(2), e1432. https://doi.org/10.1002/wfs2.1432

Demmel, M., Göbel, T., Gonçalves, P., & Baier, H. (2024). Data Synthesis is Going Mobile—On Community-Driven Dataset Generation for Android Devices. Digital Threats, 5(3), Article 30. https://doi.org/10.1145/3688807

Du, X., Hargreaves, C., Sheppard, J., & Scanlon, M. (2021). TraceGen: User Activity Emulation for Digital Forensic Test Image Generation. Proceedings of the 1st Annual DFRWS APAC Conference. https://doi.org/10.1016/j.fsidi.2021.301133

Gonçalves, P., Dološ, K., Stebner, M., Attenberger, A., & Baier, H. (2022). Revisiting the Dataset Gap Problem–on Availability, Assessment, and Perspective of Mobile Forensic Corpora. Forensic Science International: Digital Investigation, 43, 301439. https://doi.org/10.1016/j.fsidi.2022.301439

Grajeda, C., Breitinger, F., & Baggili, I. (2017). Availability of Datasets for Digital Forensics – and What is Missing. Digital Investigation, 22(Supplement), S94-S105. https://doi.org/10.1016/j.diin.2017.06.004

Göbel, T., Maltan, S., Türr, J., Baier, H., & Mann, F. (2022). ForTrace–A Holistic Forensic Data Set Synthesis Framework. Forensic Science International: Digital Investigation, 40, 301344. https://doi.org/10.1016/j.fsidi.2022.301344

Horsman, G. (2019). Tool Testing and Reliability Issues in the Field of Digital Forensics. Digital Investigation, 28, 163–175. https://doi.org/10.1016/j.diin.2019.01.009

Hummert, C., & Pawlaszczyk, D. (2022). Mobile Forensics – The File Format Handbook. Springer. https://doi.org/10.1007/978-3-030-98467-0_5

Michel, M., Pawlaszczyk, D., & Zimmermann, R. (2022). AutoPoD-Mobile—Semi-Automated Data Population Using Case-Like Scenarios for Training and Validation in Mobile Forensics. Forensic Sciences, 2(2), 302-320. https://doi.org/10.3390/forensicsci2020023

National Institute for Standards and Technologies (NIST). (2025). The CFReDS Project. 2025, April, 20.

Pawlaszczyk, D. (2022). Mobile Forensics – The End of a Golden Age? Journal of Forensic Science & Criminal Investigations, 15(4), 555917. https://doi.org/10.19080/JFSCI.2022.15.555917

Scanlon, M., Du, X., & Lillis, D. (2017). EviPlant. Digital Investigation, 20(S), S29–S36. https://doi.org/10.1016/j.diin.2017.01.010

Visti, H., Tohill, S., & Douglas, P. (2015). Automatic Creation of Computer Forensic Test Images. In Computational Forensics (pp. 163–175). Springer. https://doi.org/10.1007/978-3-319-20125-2_14