

Регистрация присутствия и биометрических данных пользователя по протоколу bluetooth в GNU/Linux

A. Dubitsky, D. Kostiuk, D. Kulbeda, A. Markina,

Brest, Belarus*

A review of personal wireless biometric devices GNU/Linux compatibility is presented. Classification of the devices and data access patterns are reviewed, including general and specialized APIs, and dedicated tools. Using such devices as alternatives in the field of hardware identification and authentication is also considered.

Введение

Биометрические данные можно разделить на статические и динамические. Статические практически не изменяются (отпечатки пальцев, распознавание лица, ДНК, сетчатка глаза), а динамические значимы именно в разрезе времени (частота сердечных сокращений, артериальное давление, электроэнцефалография). Статическая биометрия (пример – сканер отпечатков пальцев) характерна для встроенных устройств и используется в задачах, где биометрия – не цель, а средство (для целей аутентификации многие устройства сохраняют даже не измеренные данные, а их сигнатуры). Для динамической биометрии характерно автономное исполнение и нацеленность непосредственно на получение данных: фитнес-трекеры и умные часы используют Bluetooth и получают данные о пульсе пользователя и его кинематической активности ради самих данных.

Поэтому при необходимости идентификации пользователя с помощью Bluetooth-устройств потребительского сегмента оказываются доступны именно гаджеты динамической биометрии, и задача решается путём идентификации самого пользовательского устройства.

*. <https://lves.org/ru/abstracts/321>

Динамическая биометрия

Категория динамически-изменяющихся биометрических данных, которые измеряют Bluetooth-устройства потребительского сегмента, в настоящий момент практически полностью сводится к частоте сердечных сокращений и электрической активности головного мозга (к последней категории относятся портативные энцефалографы, которые предназначены для сферы развлечений, регистрируют концентрацию внимания по уровню альфа- и бета-волн, а также, в ряде случаев, эмоциональные реакции).

Устройства данной категории применялись нами для биометрического мониторинга при оценки эффективности графических интерфейсов в проекте UXDump [1]. В частности, список устройств измерения пульса, с которыми удалось работать в качестве провайдеров данных в рамках данного проекта, включают фитнес-трекеры Xiaomi (Mi Band 1S, 2 и 3), Amazfit Bip, а также Fitbit Charge HR, а список портативных энцефалографов – устройства фирмы NeuroSky (MindWave и MindSet) и фирмы Emotiv (Epos и Insight) [2].

Способы извлечения биометрических данных из устройства

Чем большей автономностью обладает устройство, больше его разработчики склонны использовать отложенную передачу данных, что снижает энергоемкость и требования к онлайн-подключению, но взамен ограничивает либо исключает снятие результатов измерений в реальном времени. Так, отложенная передача данных практически всегда реализована в фитнес-трекерах, в то время как энцефалографы и другие менее портативные устройства динамической биометрии (например, проводные айтрекеры) передают поток данных непрерывно.

При необходимости получения данных из потребительских Bluetooth-устройств, перед разработчиком стоит выбор из трех возможных вариантов:

- Использование какого-либо универсального API для работы с Bluetooth;
- Использование API, предоставляемых вендором устройства – как правило, через специальный веб-сервис;

- Применение специализированных утилит, разработанных для получения данных от конкретного устройства.

Универсальные API

Для работы с Bluetooth (либо его вариант Bluetooth Low Energy) в GNU/Linux в качестве нижнего уровня абстракции может использоваться стек BlueZ, который поддерживает все основные протоколы и уровни Bluetooth [3]. BlueZ был первоначально разработан Qualcomm, доступен для ядра Linux версии 2.4.6 и выше, и используется как в традиционных компьютерах, так и в портативных устройствах с Linux: например, в прошивках Android-смартфонов. Различные языки программирования представляют свои реализации API доступа к возможностям BlueZ. Данные API функционируют как оболочка для BlueZ, и позволяет напрямую манипулировать устройствами Bluetooth.

Пример одной из таких реализаций – модуль bluepy в Python, который обеспечивает связь с Bluetooth-устройствами с низким энергопотреблением [4].

Еще одним характерным примером реализации Bluetooth API и профиля Bluetooth на основе стека BlueZ является Bluetooth-подсистема библиотеки Qt — QtBluetooth [5].

API от вендора устройства

Заметим, что результаты биометрии относятся к категории личных данных, а потому потребительские биометрические устройства часто ограничивают к ним доступ. Характерный паттерн защиты данных в этом случае – их частичное или полное шифрование при передаче из фитнес-трекера, а также программный доступ к ним через специализированный веб-сервис с двухфакторной аутентификацией.

В качестве примера такого подхода приведём программный доступ к данным трекеров Fitbit. Используется веб-сервис и REST API, требующий зарегистрированной учетной записи разработчика. Пример запроса к сервису Fitbit для получения частоты сердечного ритма в виде временного ряда выглядит так [6]:

```
GET https://api.fitbit.com/1/user/-/activities/heart/date/  
[date]/[end-date]/[detail-level]/time/[start-time]/  
[end-time].json
```

В результате данного запроса, в случае успешной авторизации, возвращается json-объект с временным рядом.

Специализированные утилиты

Поскольку рассматриваемые устройства относятся к потребительскому сегменту и производятся крупными партиями, для многих из них существуют свободные программные проекты, обеспечивающие доступ и получение данных. Некоторые из таких программных средств имеют графический интерфейс пользователя, а некоторые представляют собой библиотеку либо утилиту командной строки, которую можно задействовать в собственном проекте.

Примером такого проекта является Galileo – утилита командной строки, написанная на Python, позволяющая извлекать данные из фитнес-трекеров Fitbit (без их расшифровки) и синхронизировать их с веб-сервисом Fitbit. Список поддерживаемых ею устройств на основе стека Bluetooth включает Fitbit One, Fitbit Zip, Fitbit Flex, Fitbit Force, Fitbit Charge и Fitbit Charge HR [7].

В случае использования электроэнцефалографов можно воспользоваться одной из нескольких библиотек (например, python-mindwave для одноименных энцефалографов) либо более универсальным проектом Puzzlebox Synapse исследовательской компании Puzzlebox. Puzzlebox Synapse реализует графический интерфейс и в частности выполняет как отображение графиков необработанных сигналов и их частотных составляющих в режиме реального времени, а также вычисляемых энцефалографом интегральных параметров, и позволяет экспортировать данные в формате csv или txt для последующей внешней обработки [8].

Авторизация и аутентификация с помощью Bluetooth-гаджетов

Как уже упоминалось, биометрическая аутентификация остаётся прерогативой встроенных и проводных устройств. Однако используемая в GNU/Linux для авторизации и аутентификации система PAM по определению является модульной, и стандартная схема предполагает использование libnss для получения информации о пользователях и группах, а затем последовательное применение модулей для попытки аутентификации. Наиболее известны биометрические PAM-модули, не имеющие отношения к bluetooth (напри-

мер, проекты **fprint**, **PAM Fingerprint**, **pam-bioapi** для аутентификации с помощью датчика отпечатков пальцев, классический и, пожалуй, наиболее известный модуль **PAM face-recognition**, пытающийся распознать лицо пользователя с помощью веб-камеры и библиотеки **OpenCV** [9]). Но существуют также немейнстримные PAM-модули, использующие персональные гаджеты для идентификации (и, в случае не критичных к безопасности задач, авторизации). Список PAM-модулей, поддерживающих аппаратную аутентификацию, включает **pam-blue** — модуль для аутентификации с помощью гаджетов по протоколу **bluetooth** [10] (используется MAC-адрес Bluetooth-устройства), а также **pambl** — сходный модуль для **Bluetooth Low Energy** [11].

Литература

- [1] UXDump project. <https://bitbucket.org/AsyaAliset/uxdump>
- [2] EEG Headset Comparison Chart. <https://www.emotiv.com/comparison/>
- [3] Bluetooth — ArchWiki. <https://wiki.archlinux.org/index.php/bluetooth>
- [4] bluepy — a Bluetooth LE interface for Python. <https://ianharvey.github.io/bluepy-doc/>
- [5] Qt Bluetooth Module. <https://doc.qt.io/archives/qttxtended4.4/qtbluetoothmodule.html>
- [6] Web API Reference. <https://dev.fitbit.com/build/reference/web-api/>
- [7] galileo. <https://pypi.org/project/galileo/>
- [8] synapse-python. <https://github.com/PuzzleboxIO/synapse-python>
- [9] Wills N. A look at PAM face-recognition authentication <https://lwn.net/Articles/523199>
- [10] Аутентификация при помощи Bluetooth телефона или USB Flash в Debian/Ubuntu Linux https://www.opennet.ru/tips/1973_pam_auth_usb_bluetooth.shtml
- [11] Pluggable authentication module for low level authentication in unix systems by bluetooth low energy device. <https://github.com/jaffetv/pambl>