SATBAYEV UNIVERSITY

Computing & Engineering

Volume 3 (2025), Issue 1, 29-33

https://doi.org/10.51301/ce.2025.i1.05

Research and analysis of embedded software for undeclared capabilities

A.B. Batyrgaliev[1], O.A. Lizunov[1,2*]

[1]*Satbayev University, Almaty, Kazakhstan*

[2]*The Institute of Information and Computational Technologies SC MSHE RK, Almaty, Kazakhstan*

**Corresponding author: o.lizunov@bk.ru*

Abstract. This paper presents the process of extracting, studying, and analyzing embedded software to confirm the presence of an undeclared capabilities. For this purpose, an educational scenario was modeled in which a software backdoor is inserted into an encryption algorithm implemented using Arduino IDE 2.3.2. The backdoor enables covert transmission of a secret key from the electronic device to an attacker over an Ethernet network in response to a predefined code phrase received via the same network interface. The electronic device used is the open-source Arduino UNO platform. To activate an undeclared feature by sending a passphrase to an electronic device, the attacker is expected to use a software backdoor implemented in the Python programming language in the PyCharm 2021.2.2 IDE.

Keywords: *undeclared capabilities, Arduino UNO, software backdoor, XOR encryption, embedded software*.

1. Введение

Современные электронные устройства, как правило, содержат встроенное программное обеспечение (ВПО), которое выполняет важные функции управления аппаратной частью и взаимодействия с другими компонентами системы. К таким устройствам относятся смартфоны и планшеты, персональные компьютеры и ноутбуки, бытовая техника, автомобили, IoT-устройства, медицинское оборудование, вооружение и военная техника и др. ВПО обладает более высокими привилегиями исполнения в сравнении с прикладным программным обеспечением и может содержать недекларированные (недокументированные) возможности как в изначальной версии или загружаться при обновлении.

Определение терминов «недекларированные возможности» и «программная закладка» раскрываются в национальном стандарте РК СТ РК 3515-2019 «Информационные технологии. Защита от несанкционированного доступа к информации. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей».

Так, в указанном стандарте под термином недекларированные возможности понимаются функциональные возможности программного обеспечения (ПО), не описанные или не соответствующие описанным в документации, при использовании которых возможно нарушение конфиденциальности, доступности или целостности обрабатываемой информации. Реализацией недекларированных возможностей, в частности, являются программные закладки.

При этом, программные закладки – это преднамеренно внесенные в ПО функциональные объекты, которые при определенных условиях (входных данных) инициируют выполнение не описанных в документации функций ПО, приводящих к нарушению конфиденциальности, доступности или целостности обрабатываемой информации [1].

Таким образом, оба эти понятия описывают специально внесенные в ПО вредоносные функции.

Следует отметить, что Агентство национальной безопасности США проводило разведывательные операции с использованием большого арсенала программных и аппаратных закладок, которые внедрялись в программное обеспечение и аппаратную часть устройств от известных производителей: Samsung, Western Digital, Seagate, Maxtor, Huawei, Cisco, Juniper, Dell, HP, и т.д. [2].

Другим примером использования программных закладок является проведенная операция национальной разведывательной службы Израиля «Моссад» против военизированной ливанской шиитской организации и политической партии «Хезболла». Цель проведенной операции заключалась в нанесении урона членам «Хезболла» в виде серии подрывов пейджеров AR924 тайваньской марки Gold Appollo и портативных любительских радиостанций ICOM. В результате получили ранения и погибли определенное количество человек [3].

Также, в результате использования программной закладки при проведении операции «Буря в пустыне» в Персидском заливе, система противовоздушной обороны (ПВО) Республики Ирак оказалась заблокированной. Предположительно компьютеры, входящие в состав комплекса технических средств системы ПВО, закупленные Ираком у Франции, блокировали работу вычислительной системы [4].

В этой связи, проведение мероприятий по исследованию электронных устройств на наличие недекларирован-

ных возможностей, является важным элементом обеспечения информационной и физической безопасности.

2. Методы и материалы

С целью исследования возможностей ВПО выбрана открытая платформа Arduino UNO. В среде разработки Arduino IDE реализован алгоритм XOR-шифрования (исключающее или). В алгоритм внедрена программная закладка, которая активируется в момент получения платой Arduino UNO по сети Ethernet кодовой фразы «***send_me_key***» и передаче на компьютер злоумышленника секретного ключа «mysecretkeys», хранящегося в энергонезависимой памяти EEPROM. Компьютер злоумышленника находится в одной локальной сети с платой Arduino UNO. Для того чтобы Arduino UNO могла передавать и принимать данные по сети Ethernet, к ней подключена плата расширения Arduino Ethernet Shield W5100 (рисунок 1).

Рисунок 1. Arduino UNO и Arduino Ethernet Shield W5100

Фрагмент исходного кода программы в Arduino IDE и результат ее работы представлен на рисунке 2.

Рисунок 2. Фрагмент исходного кода программы в Arduino IDE

Логика работы программы заключается в шифровании сообщения и непрерывном ожидании поступления кодовой фразы на порт 41234. В момент поступления кодовой фразы, происходит ее проверка и в случае положительного результата, срабатывает код полезной нагрузки по отправке секретного ключа из памяти EEPROM на компьютер злоумышленника (рисунок 3).

```
116     if (incomingData == "***send_me_key***") {
117       Serial.print("Отправка ключа ... ");
118       client.println(keyEEPROM);  // Отправляем секретный ключ
119     }
```

Рисунок 3. Триггер (кодовая фраза) и полезная нагрузка (отправка секретного ключа)

На компьютере злоумышленника реализована программа на языке программирования Python в IDE PyCharm. Суть работы программы заключается в отправке по сети кодовой фразы на плату Arduino UNO и получении от нее секретного ключа (рисунки 4-5).

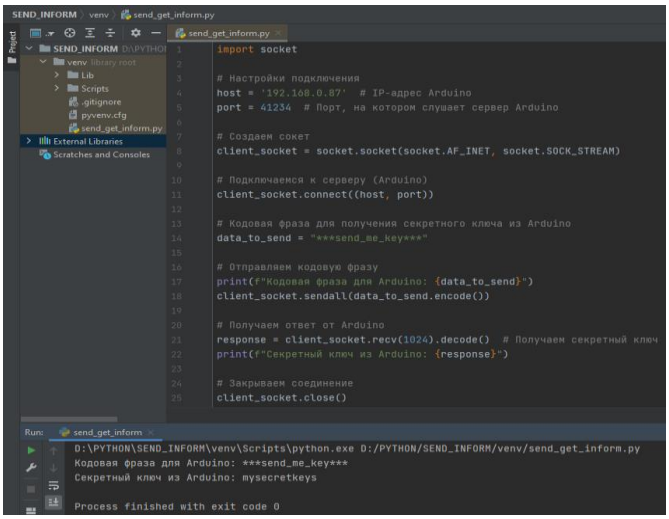

Рисунок 4. Отправка кодовой фразы и получение секретного ключа

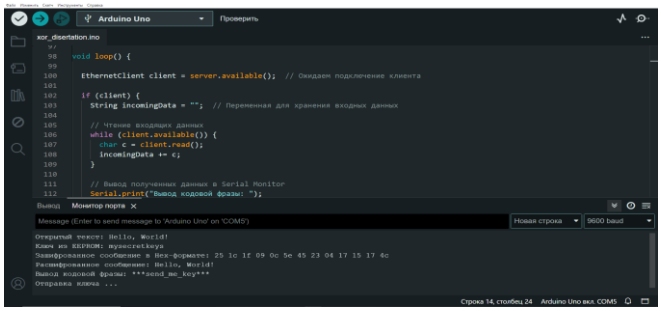

Рисунок 5. Отправка секретного ключа злоумышленнику

Далее, ВПО необходимо извлечь из памяти электронного устройства. Для этого использована программа Avrdudess 2.7, являющаяся графической надстройкой консольной программы Avrdude (AVR Downloader-Uploader – консольная программа, предназначенная для считывания, изменения и записи ВПО микроконтроллеров Atmel серии AVR) [5].

Перед считыванием ВПО из памяти Arduino UNO, необходимо определить к какому COM-порту она подключена (рисунок 6) и какая скорость передачи данных (для Arduino обычно составляет 115 200 бит/с).

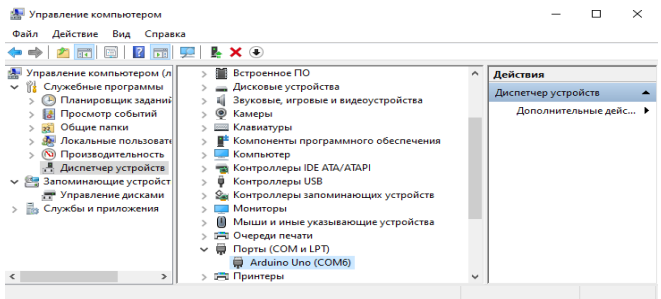

Рисунок 6. COM-порт Arduino UNO

Извлечение ВПО из flash-памяти и eeprom-памяти в программе Avrdudess 2.7 осуществляется путем установления параметров, представленных на рисунке 7.

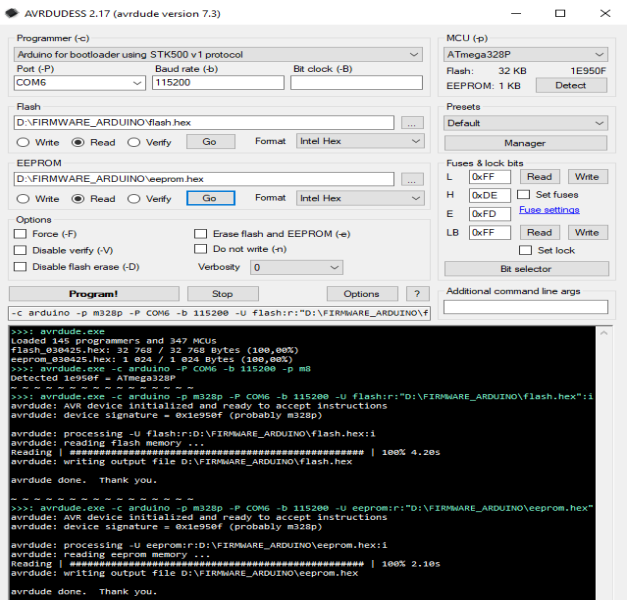

Рисунок 7. Считывание ВПО из flash-памяти и eeprom-памяти

Результат считанных данных приведен на рисунке 8.

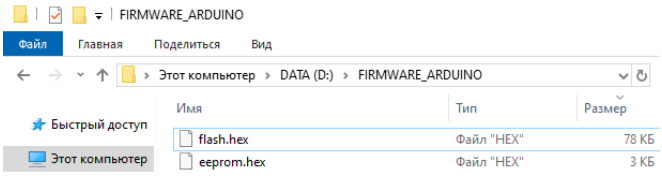

Рисунок 8. Считанные файлы flash-памяти и eeprom-памяти

Для просмотра содержимого файла flash.hex использован текстовый редактор Notepad++. Содержимое файла представлено в формате Intel-hex (рисунок 9).

Intel-hex является текстовым файлом ASCII, где каждая строка представляет собой одну запись в шестнадцатеричной системе. Формат Intel-hex используется для хранения данных, которые могут быть записаны в память устройства или ВПО [6].

Формат Intel-hex хорошо документирован и по нему можно проводить анализ содержимого файла. Однако, в данном примере исследуется файл в бинарном формате. Для этого с помощью программы Avrdudess 2.7 считывается содержимое flash-памяти в формате bin (рисунок 10).

Для анализа содержимого считанного бинарного файла flash_bin_row.bin используется утилита Strings (консольная утилита, позволяющая анализировать бинарные файлы, а также файлы данных. По умолчанию она ищет строки в формате ASCII и Unicode, возвращая последовательности длинной от трех и более символов) [7].

Необходимо отметить, что если ВПО зашифровано или упаковано, то перед поиском строк с помощью утилиты Strings, необходимо определить алгоритм шифрования и подобрать ключ для расшифрования, либо определить упаковщик и распаковать ВПО.

В результате работы программы найдены строки, представленные на рисунках 11 и 12.

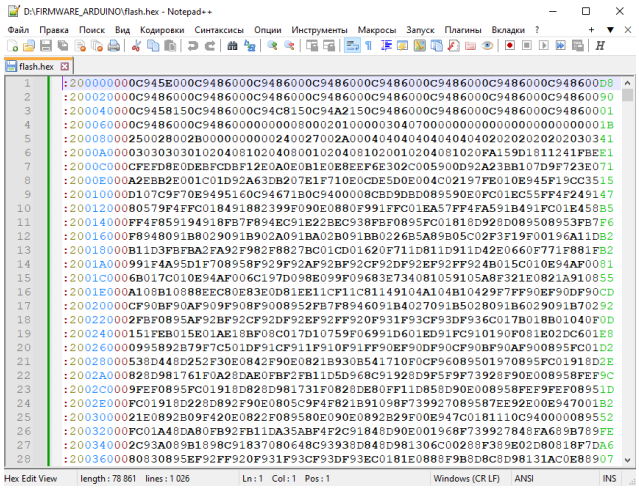

Рисунок 9. Содержимое файла flash.hex в Notepad++

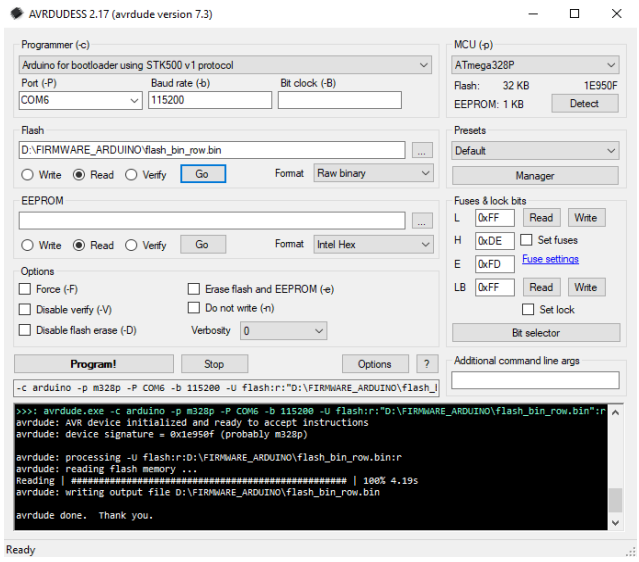

Рисунок 10. Считывание ВПО из flash-памяти в бинарном формате

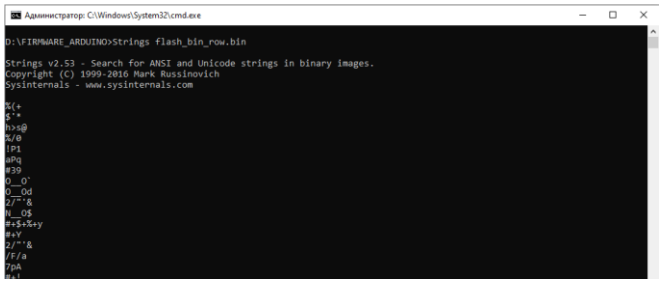

Рисунок 11. Найденные строковые значения в файле flash_bin_row.bin

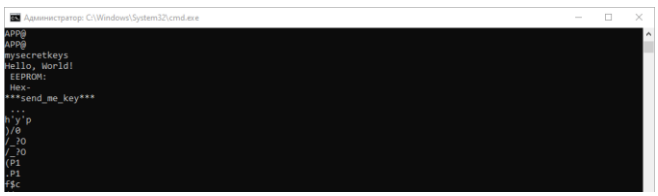

Рисунок 12. Найденные строковые значения в файле flash_bin_row.bin

После чего, необходимо провести поиск представляющих интерес строк в файле с целью проведения дальнейшего исследования и анализа. При просмотре содержимого файла flash_bin_row.bin, проведен поиск «mysecretkeys». При визуальном просмотре файла рядом со словом «mysecretkeys» обнаружены ранее найденные строки, где после словосочетания «Hello, world!» обнаружен текст, представленный предположительно в кодировке Windows-1251 (CP-1251) (рисунок 13).

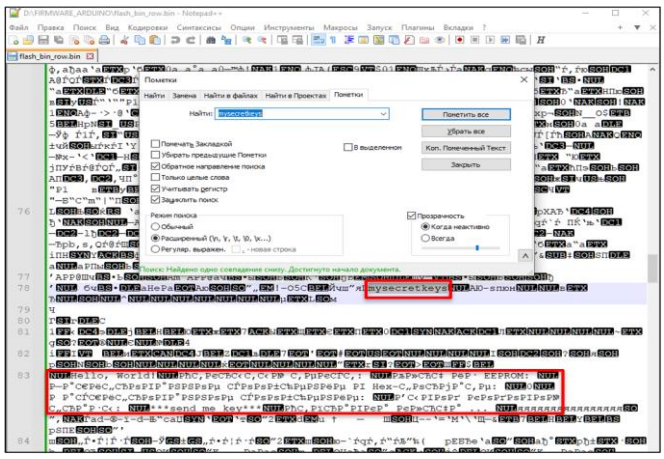

Рисунок 13. Найденные слова и словочетания в файле flash_bin_row.bin

Для декодирования найденного текста использован онлайн декодер [8]. Результат декодирования представлен на рисунке 14.

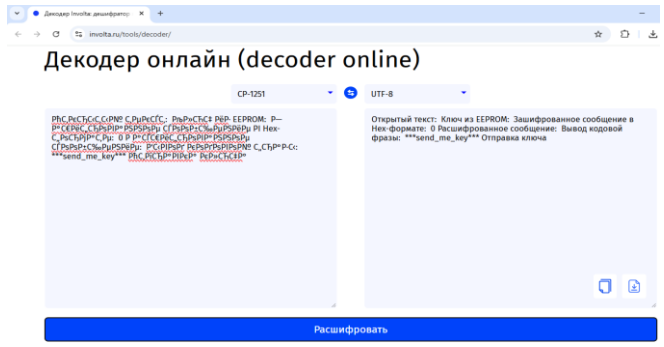

Рисунок 14. Результат декодирования

3. Результаты и обсуждение

В результате декодирования получен текст, в котором говорится о кодовой фразе «***send_me_key***» и отправке ключа. Исходя из этого, можно сделать вывод, о том, что в ВПО имеется недекларированная возможность, триггером для активации которой служит кодовая фраза – «***send_me_key***», а полезной нагрузкой является секретный ключ – «mysecretkeys».

4. Выводы

На основании учебного примера продемонстрирована работа по считыванию, исследованию и анализу ВПО платы Arduino Uno. Проведенный анализ показал возможность получения данных, на основании которых можно сделать вывод о существовании недекларированной возможности в ВПО.

References / Литература

[1] ST RK 3515-2019. (2019). Informacionnye tehnologii. Zashhita ot nesankcionirovannogo dostupa k informacii. Programmnoe obespechenie sredstv zashhity informacii. Klassifikacija po urovnju kontrolja otsutstvija nedeklari-rovannyh vozmozhnostej. *Retrieved from:* https://online.zakon.kz/Document/?doc_id=33933878

[2] Der Spiegel. (2013). Catalog Advertises NSA Toolbox. *Retrieved from:* https://www.spiegel.de/international/world/catalog-reveals-nsa-has-back-doors-for-numerous-devices-a-940994.html

[3] Kedrov, I. (2024). Pejdzhery ot «Mossad». Oborona. *Retrieved from:* https://oborona.ru/product/kedrov-ilya/pejdzhery-ot-mossad-46449.shtml

[4] Belous, A.I., Soloduha, V.A. & Shvedov, S.V. (2019). Programmnye i apparatnye trojany – sposoby vnedrenija i me-tody protivodejstvija. *Moskva: Tehnosfera*

[5] ZakKemble. (n.d.). Avrdude Gui (Avrdudess). GitHub. *Retrieved from:* https://github.com/ZakKemble/AVRDUDESS/releases/

[6] ARM. (n.d.). Intel-hex format description. *Retrieved from:* https://developer.arm.com/documentation/ka003292/latest/

[7] SecurityLab. (n.d.). Strings – programma dlja poiska strok v ispolnjaemyh faj-lah. *Retrieved from:* https://www.securitylab.ru/software/443924.php

[8] Involta. (n.d.). Dekoder Involta: deshifrator kodirovok teksta online. UTF-8, CP-1251, Base64 decode and encode online. *Retrieved from:* https://involta.ru/tools/decoder/

Жарияланбаған мүмкіндіктердің болуына ендірілген бағдарламалық қамтамасыз етуді зерттеу және талдау

А.Б. Батыргалиев[1], О.А. Лизунов[1,2*]

[1]*Satbayev University, Алматы, Қазақстан*
[2]*ҚР ГЖБМ ҒК Ақпараттық және есептеу технологиялары институты, Алматы, Қазақстан*

**Корреспонденция үшін автор: o.lizunov@bk.ru*

Аңдатпа. Осы жұмыста кіріктірілген бағдарламалық қамтамасыз етуді шығару, зерттеу және талдау процесі жарияланбаған мүмкіндіктердің бар екенін растау мақсатында қарастырылған. Оқу мақсатында Arduino IDE 2.3.2 көмегімен шифрлау алгоритміне бағдарламалық «құпия есік» енгізу жағдайы үлгіленген. Бағдарламалық «құпия есік» электрондық құрылғыдан желі арқылы құпия кілтті шабуылдаушыға жасырын түрде жіберуді жүзеге асырады, бұл әрекет алдын ала анықталған кодтық сөз тіркесін алған кезде іске қосылады. Электрондық құрылғы ретінде ашық

платформа - Arduino UNO қолданылады. Жасырын мүмкіндікті іске қосу үшін шабуылдаушы кодтық сөзді электрондық құрылғыға жіберу арқылы әрекет етеді, бұл әрекет Python бағдарламалау тілінде және PyCharm 2021.2.2 ортада жүзеге асырылған бағдарламалық «құпия есік» арқылы орындалады.

Негізгі сөздер: *жарияланбаған мүмкіндіктер, Arduino UNO, бағдарламалық жасақтама бетбелгісі, XOR-шифрлау, ендірілген бағдарламалық жасақтама*.

Исследование и анализ встроенного программного обеспечения на наличие недекларированных возможностей

А.Б. Батыргалиев[1], О.А. Лизунов[1,2*]

[1]*Satbayev University, Алматы, Казахстан*

[2]*Институт информационных и вычислительных технологий КН МНВО РК, Алматы, Казахстан*

**Автор для корреспонденции: o.lizunov@bk.ru*

Аннотация. В данной работе представлен процесс извлечения, исследования и анализа встроенного программного обеспечения с целью подтверждения наличия недекларированных возможностей. Для этого, в учебных целях, смоделирована ситуация внедрения программной закладки в алгоритм шифрования, реализованный с использованием Arduino IDE 2.3.2. Программная закладка реализует скрытую передачу секретного ключа из электронного устройства злоумышленнику по сети Ethernet в ответ на получение заранее определенной кодовой фразы, поступающей через тот же сетевой интерфейс. В качестве электронного устройства выступает открытая платформа Arduino UNO. Для активации недекларированной возможности путем осуществления отправки кодовой фразы на электронное устройство, предполагается использование злоумышленником программной закладки, реализованной на языке программирования Python в IDE PyCharm 2021.2.2.

Ключевые слова: *недекларированные возможности, Arduino UNO, программная закладка, XOR-шифрование, встроенное программное обеспечение.*

Received: 12 October 2024
Accepted: 16 March 2025
Available online: 31 March 2025