DOI: 10.15514/ISPRAS-2022-34(1)-3

Security threat level estimation for untrusted software based on TrustZone technology

D.O. Markin, ORCID: 0000-0001-5823-0632 <mdo@academ.msk.rsnet.ru>
S.M. Makeev, ORCID: 0000-0002-7451-8115 <maksm57@yandex.ru>
T.T. Ho, ORCID: 0000-0003-1149-8791 <mdo@academ.msk.rsnet.ru>
Russian Federation Security Guard Service Federal Academy,
35, Priborostroitel'naya st., Oryol, 302015, Russia

Abstract. The paper proposes a model for assessing the security of information processed by untrusted software from the components of the TrustZone technology. The results of vulnerability analysis of TrustZone technology implementations are presented. The structure of the trustlets security analysis tool has been developed. The paper deals with the problem of assessing the credibility of foreign-made software and hardware based on processors with the ARM architecture. The main results of the work are the classification of trustlets using their threat level assessment and the model of security threat level estimation of information processed by trustlets. Trustlets are software that operates in a trusted execution environment based on TrustZone technology in computers with ARM processors. An assessment of the security of information processed by trustlets for some implementations of trusted execution environments was carried out. The structural scheme of the analysis tool that allows identifying potentially dangerous code constructs in binary files of trustlets is presented. Also analysis tool's algorithm performing syntactic analysis of trustlet data is described. The calculation of the security assessment is carried out on the basis of a set of features proposed by authors. Calculated security assessment levels can be used to classify trustlets that are part of «trusted» operating systems based on TrustZone technology. The levels of potential threat to the security of the information they process are used to differ trustlets during certification tests and vulnerability search. It is advisable to use the results of the work in the interests of conducting certification tests of computer software based on processors with ARM architecture.

Keywords: ARM; TrustZone; trustlet; security estimation; software vulnerabilities

For citation: Markin D.O., Makeev S.M., Ho T.T. Security threat level estimation for untrusted software based on TrustZone technology. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 1, 2022, pp. 35-48. DOI: 10.15514/ISPRAS–2022–34(1)–3

Acknowledgements. The article is based on the materials of the report at the Seventh International Conference «Actual Problems of System and Software Engineering» (APSSE 2021).

Оценка уровня защищенности недоверенного программного обеспечения на основе технологии TrustZone

Д.О. Маркин, ORCID: 0000-0001-5823-0632 <mdo@academ.msk.rsnet.ru>
С.М. Макеев, ORCID: 0000-0002-7451-8115 <maksm57@yandex.ru>
Ч. Т. Хо, ORCID: 0000-0003-1149-8791 <mdo@academ.msk.rsnet.ru>
Академия Федеральной службы охраны Российской Федерации,
302015, Россия, г. Орёл, ул. Приборостроительная, д. 35

Аннотация. В работе предлагается модель оценки защищенности информации, обрабатываемой недоверенным программным обеспечением, состоящим из компонентов технологии TrustZone. Представлены результаты анализа уязвимостей реализаций технологии TrustZone. Разработана структура инструмента анализа защищенности трастлетов. В статье рассматривается проблема оценки надежности программно-аппаратных средств иностранного производства на базе процессоров с архитектурой ARM. Основными результатами работы являются классификация трастлетов с использованием оценки уровня их угроз и модель оценки уровня угроз безопасности информации, обрабатываемой трастлетами. Трастлеты – это программное обеспечение, работающее в доверенной среде выполнения на основе технологии TrustZone на компьютерах с процессорами ARM. Проведена оценка защищенности информации, обрабатываемой трастлетами, для некоторых реализаций доверенных сред исполнения. Представлена структурная схема инструмента анализа, позволяющего выявлять потенциально опасные конструкции кода в бинарных файлах трастлетов. Также описан алгоритм инструмента анализа, выполняющего синтаксический анализ данных трастлета. Расчет оценки безопасности осуществляется на основе комплекса признаков, предложенных автором. Вычисленные уровни оценки безопасности можно использовать для классификации трастлетов, которые являются частью «доверенных» операционных систем на основе технологии TrustZone. Уровни потенциальной угрозы безопасности обрабатываемой ими информации используются для разграничения трастлетов при сертификационных испытаниях и поиске уязвимостей. Результаты работы целесообразно использовать в интересах проведения сертификационных испытаний программного обеспечения для ЭВМ на базе процессоров с архитектурой ARM.

Ключевые слова: ARM; TrustZone; трастлет; оценка безопасности; уязвимости программного обеспечения

Для цитирования: Маркин Д.О., Макеев С.М., Хо Т.Т. Оценка уровня угроз безопасности для ненадежного ПО на основе технологии TrustZone. Труды ИСП РАН, том 34, вып. 1, 2022 г., стр. 35-48. DOI: 10.15514/ISPRAS–2022–34(1)–3

Благодарности. Статья подготовлена по материалам доклада на Седьмой Международной конференции «Актуальные проблемы системной и программной инженерии» (АПСПИ 2021).

1. Introduction

In the modern market of microprocessor technology, an impressive share is occupied by processors with the ARM architecture, which are a licensed development of the British company of the same name. The vast majority of mobile devices, numerous sensors, sensors, "Internet of Things" devices, on-board vehicle systems and even multiprocessor high-performance data processing systems are built on the basis of processor data. The developer of the circuitry of these devices has laid down the functionality of the processor in two modes: the so called "Secure World" – "trusted" mode, and "Normal World" – untrusted.

At the same time, in order for the software to work in the "trusted" mode, its developer must use an electronic signature, which can be obtained only if the corresponding license of the copyright holder is available. The described technology is called TrustZone, which determines the features of the functioning of computer hardware based on ARM processors, as well as ways to implement system and application software, their interaction when working in different operating modes of the ARM processor, the composition of technical and software tools that protect system components from unauthorized access, modification or blocking.

It is important to note that when using the vast majority of devices based on ARM processors, the user does not have the ability to manage the so-called "trusted" software loaded by the device provider into permanent memory. The described "trusted" software in terms of the developer the ARM company is called trustlet (TA – Trusted Application). These factors and numerous studies suggest that the use of such devices for processing protected information is strictly prohibited in a number of structures due to the lack of trust in both the technical means of the devices and its software. In this regard, there is an objective need to use a set of measures to assess the level of security of information processed by these devices and their software (trustlets and other software components), including within the framework of certification test procedures (certificate of state registration of a computer program No. 2021610311 Russian Federation).

In the absence of documentation for trustlets and system software, it is necessary to use methods for obtaining source code from trustlets, as well as syntactic analysis of binary data of trustlets for the presence of functional or information software objects that pose a threat, potential or immediate, to the protected information being processed. These circumstances determine the relevance of research devoted to improving the methods and tools of software research for identifying vulnerabilities and undeclared opportunities.

This paper describes a model for assessing the security of information processed by trustlets, which takes into account some binary code constructs that can be identified as potentially dangerous functional and information objects and thereby assess the threat level of information processed by trustlets.

Numerous works of both domestic and foreign studies are devoted to the problem of software security analysis. The most famous of them are Avetisyan A. I., Belivantsev A. A., Kurmangaleev Sh. F., Padaryan V. A., Gamayunov D. Yu. [1], Skovorody A. A. [1] and Gaivoronskaya S. A. [2], Zegzda P. D., Boyko V. P., Zaborovsky V. S., Podlovchenko R. I., Ivannikov V. P., Bagaev A. N. [3], Markov A. S. [4], Zakalkin P. V., Matskevich A. G. and Goryunov M. N. [5] etc. Among foreign researchers, the most famous works are Ruan X., Costan V. [6], Pinto S. [7], Cerdeira D. etc.

Attacks on Trustzone components based on device lacking memory protection are described here [8]. Example of Breaking TrustZone memory isolation and secure boot through malicious hardware on a modern FPGA-SoC described in work [9]. Samsung’s TrustZone Keymaster Design is described in the article [10]. Developing memory-safe ARM TrustZone applications is described here [11]. A Secure Cache for Arm TrustZone describes in the article [12].

In these works, various aspects of the problems of software research were touched upon, but not enough attention was paid to the issues of obtaining numerical estimates of the degree of security of information processed by trustlets in ARM systems.

2. Task description

2.1 The object of the study

The object of this work is a software solution for a trusted execution environment (TEE) – trusted applications created on the basis of TrustZone technology in computer systems based on processors with ARM architecture.

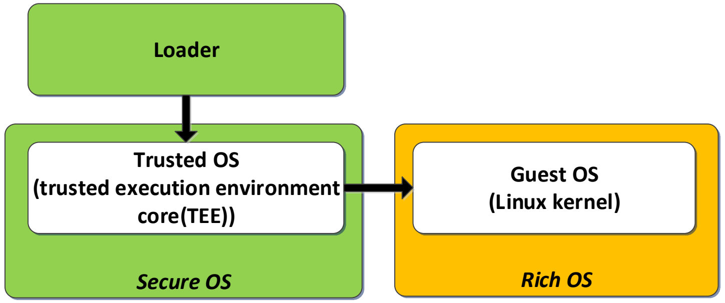

Fig. 1. Loading order in platforms based on ARM processors

TrustZone technology is a hardware-based secure boot environment that allows you to create a TEE. TEE based on TrustZone is called in the terminology of ARM-Secure World or Secure OS, "untrusted" - Rich Execution Environment or Rich OS (iOS, Android, Sailfish, Tizen, Linux, Windows, etc.). The order of loading a computer with an ARM processor is shown in fig. 1, and the interaction of the TEE (Secure OS) and guest operating system (OS) (Rich OS) modes is shown in fig. 2.

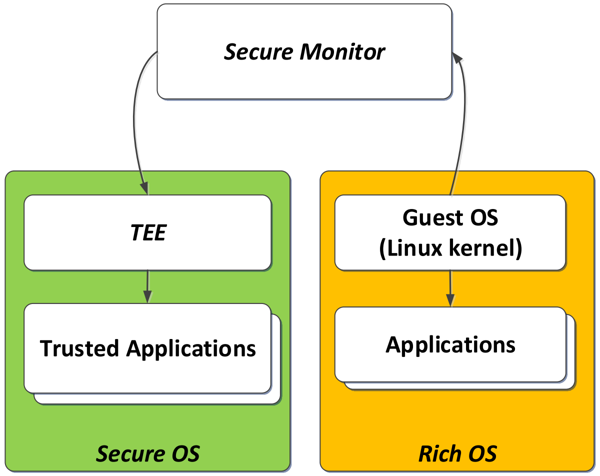

Fig. 2. Interaction of trusted and guest operating systems in platforms based on ARM processors

The essence of the TrustZone technology is to manipulate the operating modes of the processor using a signal set by the NS (Non-Secure) bit. If NS=1, then the processor is in Non-Secure mode (guest OS, or Rich OS, or Normal World OS), if NS=0, then in Trusted, that is, Secure mode (Trusted OS or Trusted OS, or Trusted Execution Environment (TEE)).

The NS bit does not just tell the processor core in which mode it should work. It is also an external signal connected from the processor to almost all the peripherals. In general, the peripheral is connected to the CPU by address, data and control buses. NS is part of the control signals for those processors where TrustZone is implemented. Thus, not just Read, Write commands go from the CPU to the device, but Secure Read, NonSecure Read, Secure Write, NonSecure Write.

ARM TrustZone, built into mobile devices, has been available to users for many years. Over the past few years, the attention to TrustZone has increased significantly. Projects were developed to apply it in various fields of activity: in mobile, industrial, automotive and aerospace. Attention to the technology is supported by the publications of large manufacturers of technical parts and recommendations for development.

The open nature of the TrustZone technology makes it possible to conduct research in the field of improving the security of data processing, in contrast to the proprietary solutions discussed above. Availability of the certificate of the "Aladdin TSM" module [13,14], based on the use of TrustZone technology, according to the requirements of the FSTEC "Requirements for trusted download tools" and the protection profile "IT.SD3.UB2.PZ", confirms this.

2.2 Comparative analysis of known implementations of "trusted" OS

Currently, there are about ten [15] so-called "trusted" operating systems that use the TrustZone functionality. A number of implementations of the TrustZone technology are closed from researchers (Trustonic TCP, Qualcomm SEE). A comparative analysis of the known implementations of "trusted" OS [18] is shown in Table 1.

Table 1. Comparative analysis of "trusted" operating systems

Name	Developer	Features
GlobalPlatform	Non-profit association of companies G&D Mobile Security,	Industrial standard TEE

	ARM, FIME, Trustonic, Gemalto, Oracle	
General Dynamics OKL4	Open Kernel Labs	Based on the L4 (L3) OS, a microkernel for i386 systems. Developed for iOS devices, since 2012 the source texts are closed
Google Trusty TEE	Google (only fo Android)	It is compatible with ARM and Intel processors. It consists of three main elements: 1) the OS kernel (based on Little Kernel); 2) the kernel driver for managing the interaction of TEE (Trusty) and REE (Android); 3) the user space library for managing the interaction of REE (Android) and TEE (Trusty)
Linaro OP-TEE	Research group Linaro, STMicroelectronics	Based on GlobalPlatform 1.1. An open source project. It consists of three main elements: 1) the OS kernel (memory management modules, interrupts, etc.); 2) the client of the untrusted user space-the monitor-intermediary between the user and kernel spaces, the libraries of the GlobalPlatform TEE Client API implementation; 3) the kernel driver for performing transactions between the trusted and untrusted OS
Jailhouse	Siemens	In a general sense, it is not an OS, but a monitor of resource accesses. It can be run as part of the OS FreeRTOS, Erika3, Linux, Zephyr. Supports processors with the following architectures: ARMv8, ARMv7, x86_64. It requires 2 processors and 50 MB of RAM
QSEE	Qualcomm Secure Execution Environment	Based on General Dynamics OKL4. Closed source text
seL4	Open Kernels labs	It is based on the L4 OS. It has passed a formal verification of correctness by determining the functionality specification and proving its correctness by means of strict logical inference. Open source text. Real-time OS for the firmware of Qualcomm wireless modem processors. The code volume is about 9600 lines. Interrupts during code execution are disabled. Supports ARM processors up to ARMv8
TrustTonic Kinibi	TrustTonic	Closed source text. For the Android OS. For Samsung devices (smartphones and tablet computers). Provides data encryption and device authentication. The trustletshave access to the network. There is a developer kit (SDK) that is compatible with the GlobalPlatform API standards
Xen	University of Cambridge	The microkernel hypervisor. Supports ARM and Intel processors
Xvisor	Developer Community Xvisor	Type 1 hypervisor. Open source text. It has a memory management module, a scheduler, a load balancer and a thread balancer
Aladdin TSM [13, 14]	Aladdin R. D.	Supports i.MX6 processors. It is certified according to the requirements of the FSTEC for the means of trusted loading of the level of the basic I / O system of the second class of protection "IT. SDZ.UB2. PZ" (certificate No. 4155)

The analysis of the features of the TEE functioning in modern ARM processors allows us to conclude that the adaptation of TrustZone technology to the needs of the domestic economy through the use of approaches to increase trust in it, including certification and development of domestic software based on TrustZone, is an important direction for improving domestic information security technologies in the absence of production of a sufficient number of elements. Especially considering the number of modern devices based on processors with ARM architecture.

2.3 Comparative analysis of file formats for trusted application execution environments based on TrustZone technology

The analysis of publications on the TEE research based on the TrustZone technology allowed us to identify the peculiarities of binary files (trustlets) of software executed in the TEE data. According to the list of well-known "trusted" OS for computer systems based on ARM TrustZone, the following binary file formats are used:

for TrustTonic Kinibi OS, the MobiCore Load Format (MCLF) is used, for trustlet files-the extension *.tlbin*;

for Linaro OP-TEE and Aladdin TSM OS-HSTO form-mate, trastlet files have an extension *.ta*.

Trustlets are located in the normal file system of the device and are files containing executable code. They have a similar format to ELF (Executable Linux Format), but with a smaller amount of header and additional structures.

Unlike. ta trustlets,. tlbin does not have the usual ELF file header. To determine the address of the beginning of the executable code, it is necessary to add its length (0x80 + TSHL) to the offset of the text segment header. The formats of the HSTO and MCLF trustlets are shown in fig. 3, 4.

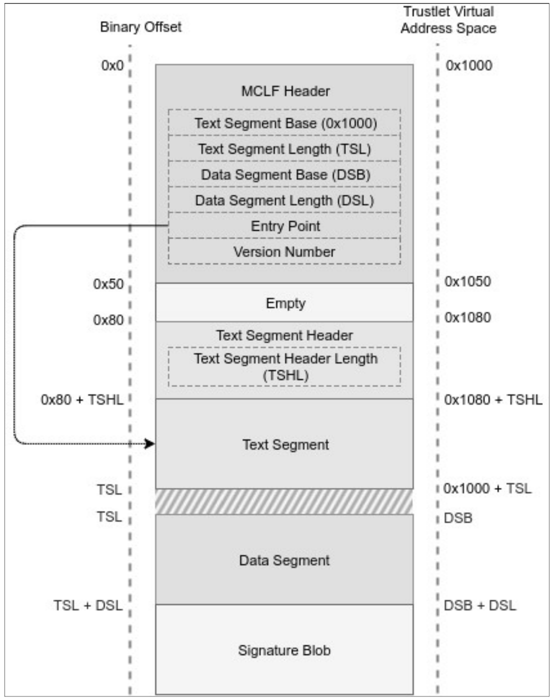

Fig. 3. File structure .tlbin

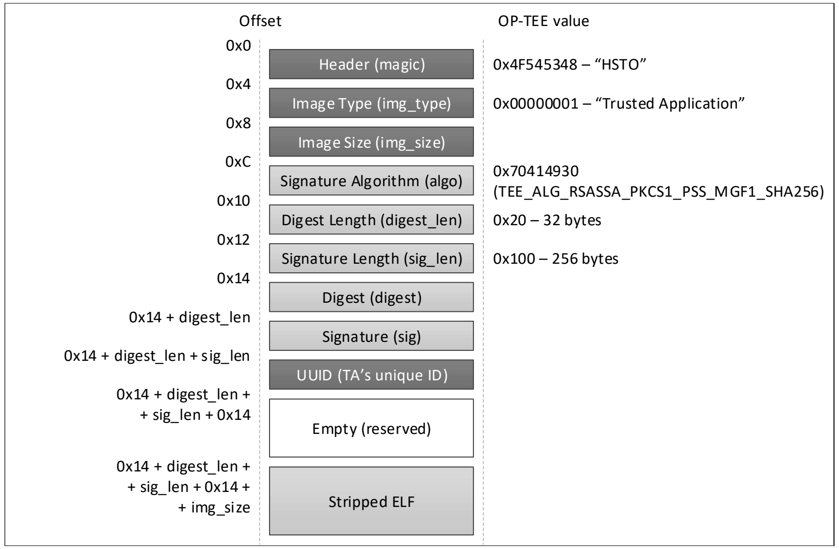

Fig. 4. File structure .ta

The digital signature is located after the "Data Segment" field, the address of its beginning can be determined by adding the lengths of "Text Segment"and" Data Segment". The SHA-256 algorithm is used as a digital signature. The first 4 bytes determine the length of the module, followed by the module itself. After the module, 4 bytes are allocated to describe the length of the public key (0x1), followed by the public key itself, the final 256 bytes are occupied by the signature.

The main differences in the file structure .ta and .tlbin:

- format .ta allows you to specify which crypto algorithm will be used to sign the trustlet,
- .tlbin always uses the SHA-256 algorithm as a signature .ta uses an ELF file without service information (stripped elf) as an executable code,
- .tlbin does not use the ELF file format.

At the same time, when extracting stripped elf from .ta its use is impossible without modification.

From the analysis of the formats, it can be seen that they differ from the ELF format, so it is impossible to run the trustlet in the Unix system emulator without additional preparation of the trustlet or the environment [16]. The task of preparing a trustlet to run in an emulator (for example, QEMU, Virtual Box, etc.) can be solved using static instrumentation methods [3]. To do this, you need to convert the trustlet file to the ELF format.

An example of the names of traslites in the Kinibi Trustonic OS is presented in Table 2. Examples of the names of trustletsin the OS based on Aladdin TSM and OP-TEE are presented in Table 2.

Table 2. Examples of MCLF format TAs` names in Kinibi TrustZone

vendor.app.mcRegistry
07010000000000000000000000000000.tlbin
07060000000000000000000000000000.tlbin
ffffffff00000000000000000000000f.tlbin
system.app.mcRegistry
00060308060501020000000000000000.tlbin
ffffffff00000000000000000000000c.tlbin
ffffffff00000000000000000000000d.tlbin
fffffffd000000000000000000000004.tlbin

Table 3. Examples of HSTO format TAs` names in Alladdin TSM TrustZone (JaCarta Box)

Aladdin TSM (JacartaBox)
5b9e0e40-2636-11e1-ad9e0002a5d5c51b.ta
5ce0c432-0ab0-40e5-a056782ca0e6aba2.ta
48002366-708f-4b23-ab0921ce46766a11.ta
e13010e0-2ae1-11e5-896a0002a5d5c51b.ta
OP-TEE
5b9e0e40-2636-11e1-ad9e-0002a5d5c51b.ta
5ce0c432-0ab0-40e5-a056-782ca0e6aba2.ta
5dbac793-f574-4871-8ad3-04331ec17f24.ta
ffd2bded-ab7d-4988-95ee-e4962fff7154.ta

In the file systems of computer systems based on ARM processors, the trustlets files are usually available to unprivileged users.

2.4 Task description

The task description is as follows.

Source data:

1) the set of trusted applications (truslets) extracted from the TEEs based on the TrustZone technology:"Aladdin TSM", "OP-TEE", "TrustTonic Kinibi";

Necessary:

1) to develop a model for assessing the level of security of information processed by the TEE software based on TrustZone technology, which differs from the known ones by taking into account the logical features of information input-output flows and API calls.

Limitations and assumptions:

1) the source code of the trustlets is not available;
2) the studied trustletshave one of the following formats: "HSTO" (Linaro OPTI OS) and "MCLF" (Trustonic Kinibi OS);
3) the format of trustlets (ELF-like) does not allow them to be executed in a traditional environment (untrusted UNIX-like operating system – Android, Debian, Ubuntu, etc.).

3. Investigation of vulnerabilities of the trusted application execution environment based on TrustZone technology

Trusted Execution Environments is one of the modern security mechanisms for protecting the integrity and confidentiality of applications. This technology is de facto a hardware technology for implementing TEE in mobile environments, as well as industrial control systems, servers, and budget household devices. The most common implementations of TEE are the developments of Qualcomm, Trustonic, Huawei, Nvidia and Linaro [17].

The features of vulnerabilities detected in the TrustZone-based TEE [18,19] are usually associated with:

- classic input data validation errors, such as "buffer overflow";
- numerous architectural shortcomings of TEE systems, such as the lack of ASLR technology and other system protection tools for applications;
- the lack of consideration of hardware properties at the architectural and microarchitectural level when implementing system TEE, for example, associated with the appearance of side channels

for transmitting information in cache memory or interaction with the memory of reconfigurable equipment capable of accessing confidential data.

The E1 exploit exploits an error in the LCB kernel and is able to execute arbitrary code with the EL1 privilege level, and, as a result, is able to develop the attack further, up to extracting secret keys and decrypting the disk and unlocking the device bootloader [20].

The E6 exploit allows an attacker to take control of the Linux kernel by sending a special set of data from a user-level application to trusted Widevine applications [21].

The TEE system requires drivers in the software to access protected software, technical components, I/O devices that process critical information. The complexity of implementing drivers that are traditional sources of errors and their functioning with an extended privilege level lead to the appearance of critical vulnerabilities.

Interfaces between the TEE components allow you to exchange sufficiently large amounts of data with privileged access rights to a sufficiently large number of trusted applications – trustlets. For example, in some implementations of the "trusted" OS Trustonic TEE, their number reaches 32, and in Widevine-70.

Another architectural problem is the excessively large amounts of data that are transferred between trusted and untrusted environments. For example, in Qualcomm's TEE, the buffer size for such data reaches 1.6 MB, and this volume can indirectly increase.

In some implementations of TEE, trustlets can be displayed in the physical memory of the unprotected mode. This applies primarily to Qualcomm's TEE. After scanning the physical address space of the Linux kernel and fixing it, it is possible to introduce a backdoor into the system (exploit E6 [21]). Unlike Qualcomm's TEE, in the implementation of Trustonic TEE, the display of the trustlet in the physical memory of the unprotected mode is excluded.

Some security problems are associated with the appearance of side channels through application debugging tools. Such channels are found in Huawei's TEE implementations [20]. In particular, using the TEE system call, the trustlet (trusted application) uploads its own stack trace to public memory, which allows an attacker to study the address space of the trustlet and use it to create an exploit in the future. The same vulnerability was found in the Trustonic TEE.

It should be noted that in all known implementations of TrustZone there is no ASLR technology (with the exception of its limited implementation in Qualcomm TEE), i.e. all trusted applications are always loaded at a fixed address of the virtual address space (0x1000). In addition, a shared library (mcLib) is provided, which is also downloaded at a permanent address for each trustlet (0x7D01000). Thus, any vulnerability found in the trustlet can be used without much effort when determining the address of loading the trustlet into memory. In addition, the shared library used by trustlets contains a significant amount of code that serves as a source of gadgets (data) for identifying system calls and calls to trusted applications.

Another system vulnerability is the lack of a separate stack for cookie data, as well as the lack of protective (barrier) pages between the data of protected processes, which leads to the appearance of a technical possibility for implementing a "buffer overflow"attack.

The TEE has mechanisms for protecting against code execution in memory areas by means of the WXN bit in the SCTLR register, as well as the XN memory page attribute. However, well-known TEE implement these protection mechanisms partially. For example, there is no stack for cookie data in the Trustonic TEE. Cookie data is stored as global variables from the data segment of the trustlet without buffer protection pages. In addition, the layout of data in memory is implemented in such a way that the stack is located at the end of the data segment, and global variables are in front of it, which creates ideal conditions for exploiting a buffer overflow vulnerability.

Unlike Trustonic, Qualcomm's TEE creates a stack with a random location in the address space, but there is also no mechanism for monitoring the integrity of protective buffer pages.

Huawei's TEE lacks both data execution protection and protective buffer pages.

A common disadvantage of TEE is also the lack of hardware control of the integrity of the TEE code, which weakens the overall level of trust in "trusted" software.

In addition to the considered architectural vulnerabilities, there are numerous typical software errors in the TEE implementations associated with:

- incorrect processing of input or output data;
- name verification errors;
- errors that lead to the possibility of exploiting a vulnerability such as "buffer overflow";
- incorrect processing of parameters.

Such errors can be used as starting points for extending privileges and are found in almost all existing components of the TEE of various manufacturers.

Thus, the analysis of a number of detected vulnerabilities in the implementation of TEE based on ARM TrustZone shows that even the implementation of the technology that increases the overall level of security of the system has numerous vulnerabilities and errors that can be used by attackers for unauthorized access and performing illegitimate operations with data.

4. Security threat level estimation for untrusted software based on TrustZone technology

The analysis of open sources of information [2,3,5] allowed us to identify signs that the presence of which in the software implementation may indicate the potential danger of this software from the point of view of the possible implementation of threats to the security of the processed information.

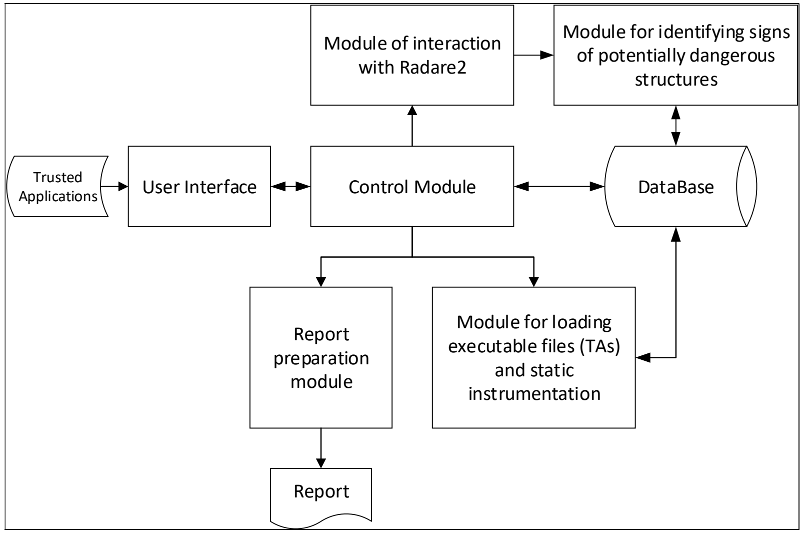
Module of interaction with Radare2
Module for identifying signs of potentially dangerous structures
Trusted Applications
User Interface
Control Module
DataBase
Report preparation module
Module for loading executable files (TAs) and static instrumentation
Report

Fig. 5. Block diagram of the program for identifying signs of potentially dangerous structures

These include [18]:

- the presence and number of data entry and exit points (N_{in} and N_{out}, respectively), the amount of input/output data (V_{in} and V_{out}, respectively);
- the variety of APIs $F^{unsafe}_{API\_TEE}$, including specific API functions of TEE that handle critical data and information (account and authentication information, key information for the cryptographic subsystem that identifies(address) of the network, and information about the technical characteristics of the computer, etc.);

- the presence of calls to function objects (functions) of the standard system libraries F_{system}^{unsafe}, including the processing critical information and data;
- the presence of specified sequences of calls to functional objects, including calls to certain API functions and calls to functional objects of standard system libraries IF^{unsafe};
- the volume V_{WSM} and frequency of processing data N_{WSM} coming from the "untrusted" OS via the WSM (World Shared Memory) buffer $\langle V_{in}^{WSM}, N_{in}^{WSM} \rangle$ and transmitted from the TEE to the "untrusted" OS – $\langle V_{out}^{WSM}, N_{out}^{WSM} \rangle$.

Thus, the model for assessing the level of security of information processed by the TEEe based on TrustZone technology can be represented as a tuple:

$$Z = \langle N_{in}, N_{out}, V_{in}, V_{out}, F_{API\_TEE}^{unsafe}, F_{system}^{unsafe}, IF^{unsafe}, \langle V_{in}^{WSM}, N_{in}^{WSM} \rangle, \langle V_{out}^{WSM}, N_{out}^{WSM} \rangle \rangle.$$

In order to identify the listed features, a program was developed (certificates of state registration of a computer program No. 2020666135, 2021610311 Russian Federation), the block diagram of which is shown in the fig. 5.

In order to study the machine code instructions in the binary files of trustlets, a tool for analyzing binary files of trustlets was developed and implemented in Python.

Table 4 shows a list of some functions found using a Python software tool.

Table 4. Names of system functions in the ffd2bded-ab7d-4988-95ee-e4962fff7154.ta file

utee_storage_free_enum, utee_storage_reset_enum, utee_storage_start_enum, utee_storage_next_enum, utee_storage_obj_read, utee_storage_obj_write, utee_storage_obj_trunc, utee_storage_obj_seek	Working with the file system
utee_hash_final, utee_cipher_init, utee_cipher_update, utee_cipher_final, utee_cryp_obj_get_info, utee_cryp_obj_restrict_usage, utee_cryp_obj_get_attr	Cryptographic functions
utee_open_ta_session, utee_invoke_ta_command, utee_close_ta_session	Working with sessions (data exchange between the trustler and the application via the OS)

After studying the last group of system calls, we can conclude that all calls are preceded by a call to the_open_ta_session function. This function returns the session ID, which is used in subsequent calls. And after working with the system functions, the session is closed by ut ee_close_t a_session. That is, the exchange of translit information with other applications will occur between the call of ut ee_open_t a_session and ut ee_close_ta_session.

5. Conclusions

As a result, based on this information, it is possible to identify the places of system calls in the machine code and determine the type of functions for further classification according to the degree of vulnerability or the information being processed.

As part of the work, the analysis of executable file formats for TEE based on TrustZone technology – trustlets for Kinibi Trustonic OS (Samsung smartphones), OP-TEE (Linaro) and JacartaBox (Aladdin TSM) was carried out. Their distinctive features and approaches, such as static instrumentation, are identified, which allow using dynamic analysis methods in relation to trustlets in the future.

References / Список литературы

[1] Zakharkin P.V., Melnikov P.V. The system of software analysis for the absence of undeclared capabilities. Software Engineering, vol. 9, no. 2, 2018, pp. 69-75 (in Russian) / Закалкин П.В., Мельников П.В. Система анализа программного обеспечения на предмет отсутствия недекларированных возможностей. Программная инженерия, том 9, no. 2, 2018 г., стр. 69-75.

[2] Skovoroda A.A., Gamayunov D.Yu. Dynamic analysis of mobile applications. Software Engineering. 2019, No. 7-8, pp. 324-333 (in Russian) / Сковорода А.А., Гамаюнов Д.Ю. Динамический анализ мобильных приложений. Программная инженерия, том 10, no. 7-8, 2019 г., стр. 324-333.

[3] Gaivoronskaya S.A. Hybrid method for detecting shellcodes. Systems of high availability. 2012, vol. 2, No. 8, pp. 33-44 (in Russian) / Гайворонская С.А. Гибридный метод обнаружения шеллкодов. Системы высокой доступности, vol. 8, no. 2, 2012 г., pp. 33-44

[4] Begaev A.N., Kashin S.V. et al. Identification of vulnerabilities and undeclared opportunities in software. St. Petersburg, ITMO University, 2020, 38 p. (in Russian) / Бегаев А.Н., Кашин С.В. и др. Выявление уязвимостей и недекларированных возможностей в программном обеспечении. Учебно-методическое пособие. Санкт-Петербург, Университет ИТМО, 2020 г., 38 стр.

[5] Markov A. S., Cirlov V.L., Barabanov A.V. Methods for assessing the inconsistency of information security tools. Moscow, Radio and Communications, 2012, 192 p. (in Russian) / Марков А.С., Цирлов В.Л., Барабанов А.В. Методы оценки несоответствия средств защиты информации. Москва, «Радио и связь», Москва, «Радио и связь», 2012 г., 192 стр.

[6] Goryunov M.N., Eremenko V.T. et al. Recognition of functional objects of software in the absence of source texts. Information systems and technologies, no. 5, 2013, pp. 112-120 (in Russian) / Горюнов М.Н., Ерёменко В.Т. и др. Распознавание функциональных объектов программного обеспечения в условиях отсутствия исходных текстов. Информационные системы и технологии, no. 5, 2013 г., стр. 112-120.

[7] Costan V., Devadas S. Intel SGX Explained. URL: https://eprint.iacr.org/2016/086.pdf, 2016.

[8] Pinto S., Santos N. Demystifying Arm TrustZone: A Comprehensive Survey. ACM Computing Surveys, vol. 51, issue 6, 2019, article no. 130, 36 p.

[9] Stajnrod R., Yehuda R.B, Zaidenberg N.J. Attacking TrustZone on devices lacking memory protection. Journal of Computer Virology and Hacking Techniques, 2021, 11 p.

[10] Gross M., Jacob N. et al. Breaking TrustZone memory isolation and secure boot through malicious hardware on a modern FPGA-SoC. Journal of Cryptographic Engineering, 2021, 16 p.

[11] Shakevsky A., Ronen E. Avishai Wool Trust Dies in Darkness: Shedding Light on Samsung's TrustZone Keymaster Design. URL: https://eprint.iacr.org/2022/208.pdf, 2022.

[12] Wan S., Sun M. et al. RusTEE: developing memory-safe ARM TrustZone applications. In Proc. of the Annual Computer Security Applications Conference (ACSAC 2020), 2020, pp. 442–453.

[13] Benedito O., Delgado-Gonzalo R., Schiavoni V. KeVlar-Tz: A Secure Cache for Arm TrustZone. Lecture Notes in Computer Science, vol 12718, 2021, pp. 109-124.

[14] Trusted platform for ARM processors. URL: https://www.aladdin-rd.ru/catalog/tsm, accessed: 10.12.2021 (in Russian) / Доверенная платформа для процессоров ARM.

[15] Certificate of Conformity FSTEC of Russia No. 4155. URL: https://www.aladdin-rd.ru/upload/certified/sertifikat_fstek_4155_tsm.pdf, accessed: 10.12.2021 (in Russian) / Сертификат соответствия ФСТЭК России No 4155.

[16] Yehuda R.B., Leon R., Zaidenberg N.J. ARM Security Alternatives. In Proc. of the 18th European Conference on Cyber Warfare and Security, 2019, pp. 604–612.

[17] Markin D.O., Makeev S.M. et al. Methodology of research of system software of network equipment of the Cisco family for the presence of undeclared capabilities. Scientific Notes of the Orel State University, no. 3(88), 2020, pp. 215-221 (in Russian) / Маркин Д.О., Макеев С.М. и др. Методика исследования системного программного обеспечения сетевого оборудования семейства cisco на предмет наличия недекларируемых возможностей. Ученые записки Орловского государственного университета, no. 3(88), 2020 г., стр. 215-221.

[18] Cerdeira, D., Santos N. et al. SoK: Understanding the Prevailing Security Vulnerabilities in TrustZone-assisted TEE Systems. In Proc. of the IEEE Symposium on Security and Privacy, 2020, pp. 1416-1432.

[19] Markin D.O., Ho T.T., Meshkov N.P. Features of the search for software vulnerabilities based on TrustZone technology. Problems of information security. Computer systems, no. 4, 2020, pp. 79-87 (in Russian) / Маркин Д.О., Хо Т.Ч., Мешков Н.П. Особенности поиска уязвимостей программного обеспечения на основе технологии TrustZone. Проблемы информационной безопасности. Компьютерные системы, no. 4, 2020 г., стр. 79-87.

[20] Markin D. O., Ho T. T. Research of vulnerabilities of the trusted application execution environment based on TrustZone technology. Izvestiya Tula State University. Technical sciences, issue 9, 2020, pp. 316-328

(in Russian) / Маркин Д.О., Хо Т.Ч. Исследование уязвимостей доверенной среды исполнения приложении на основе технологии TrustZone. Известия тульского государственного университета технические науки, вып. 9, 2020 г., pp. 316-328.

[21] Extracting Qualcomm’s KeyMaster Keys – Breaking Android Full Disk Encryption (Jun 2016). URL: https://bits-please.blogspot.com/2016/06/extracting-qualcomms-keymaster-keys.html, accessed: 10.12.2021.

[22] Di Shen. Attacking your "Trusted Core". Exploiting TrustZone on Android. URL: https://www.blackhat.com/docs/us-15/materials/us-15-Shen-Attacking-Your-Trusted-Core-Exploiting-Trustzone-On-Android.pdf, accessed: 10.12.2021.

Information about authors / Информация об авторах

Dmitry Olegovich MARKIN – candidate of engineering sciences, employer of the Academy of Federal Guard Service. His research interests include: information security, machine learning methods, pattern recognition, cryptographic methods for protecting information.

Дмитрий Олегович МАРКИН – кандидат технических наук, сотрудник Академии ФСО России. В круг его научных интересов входят: информационная безопасность, методы машинного обучения, распознавание образов, криптографические методы защиты информации.

Sergey Mikhailovich MAKEEV – candidate of engineering sciences, employer of the Academy of Federal Guard Service. His research interests include: decision support systems, machine learning methods, pattern recognition, information security.

Сергей Михайлович МАКЕЕВ – кандидат технических наук, сотрудник Академии ФСО России. Его исследовательские интересы включают: системы поддержки принятия решений, методы машинного обучения, распознавание образов, информационную безопасность.

Trung Thaj HO is an employer of the Academy of Federal Guard Service. His research interests include: information security, machine learning methods, cryptographic methods for protecting information.

Чунг Тхай ХО — сотрудник Академии ФСО России. В круг его научных интересов входят: информационная безопасность, методы машинного обучения, криптографические методы защиты информации.