CATALOGUE OF TECHNOLOGIES

IVANNIKOV INSTITUTE
FOR SYSTEM PROGRAMMING
OF THE RAS

DEDICATED
TO THE 30TH ANNIVERSARY OF ISP RAS
AND THE 300TH ANNIVERSARY OF RAS

CATALOGUE
OF TECHNOLOGIES

2024

Ivannikov Institute for System Programming
of the Russian Academy of Sciences

25, Alexander Solzhenitsyn str. 109004 Moscow, Russia
E-mail: scsec@ispras.ru

CONTENTS

SOURCE CODE ANALYSIS, VERIFICATION, TESTING

BINARY CODE ANALYSIS, FUZZING

NETWORK TRAFFIC ANALYSIS

REQUIREMENT MANAGEMENT

2024. 30 YEARS TOGETHER

ARUTYUN AVETISYAN

Academician of the RAS, ISP RAS Director

The 2024 anniversary edition presents 30 technologies divided into thematic blocks. The section “ISP RAS: Ecosystem of Innovation” provides a detailed description of the model of the institute’s operation; the following sections report on the annual results of the ISP RAS-based research centers. But first of all, on to the year’s main theme.

This year marks two important anniversaries: ISP RAS turns 30, and Russian Academy of Sciences turns 300. The institute was founded on January 25th, 1994 by Victor Ivannikov, ISP RAS academician, and being effectively an organizational structure of his scientific school. During all these years ISP RAS is evolving in three directions: we perform research, develop technologies and grow talents, uniting science, education, and innovation.

Within the last five years we have achieved important fundamental research results and used them as a basis for innovative technologies, such as a full stack of instruments and tools for secure development lifecycle (SDLC) for compiled programming languages. We have started working on trusted artificial intelligence and developed tools for creating such technologies. We have also developed a trusted version of Talisman, a platform for building informational systems, and Asperitas, a platform for performing resource-heavy calculations and storing data. We continue working on other projects such as JetOS, a real-time operating system (jointly with GosNIIAS), and the information modeling system for extra large construction sites.

Our SDLC tools are deployed in more than 200 companies. Trusted AI technologies, including countering attacks on AI models, are used in Kaspersky Lab and EC-Leasing; the trusted Talisman is utilized in MGIMO University for intellectual data analysis in foreign affairs. The Asperitas system provided a foundation for a biomedical platform.

ISP RAS is very active in education. We have started renewing courses and educational programs with the goal of developing cybersecurity as a science. Starting from 2021, we are modernizing the undergraduate program “Software Engineering" taught on the Computer Science Faculty of Higher School of Economics. In 2002, the master's program "Data analysis and foreign affair dynamics" have been founded in MGIMO. We have started collaborating with Saratov State University; students from there write theses and papers while being mentored by our employees. Our existing system programming labs in Veliky Novgorod and Yerevan have been joined by the labs in Orel and in Moscow (Plekhanov Russian University of Economics). The joined scientific group is founded in Institute for System Dynamics and Control Theory of SB RAS. Our scholarship program is founded in 2020 and currently involves students from Moscow State University, Moscow Institute for Physics and Technology, HSE, Novgorod State University, and many others. We mentor and engage in our industrial and research projects more than 50 students and 20 postgraduates. Starting from 2023, we are organizing the annual educational program "Trusted Artificial Intelligence" in Sirius University, partnered with MIPT, Nizhny Novgorod State University, Scoltech, AIRI, and Innopolis University; this year the number of program applications have tripled.

ISP RAS's harmonious development of science, education, and innovation naturally resulted in forming communities that allow significantly increasing productivity of our work. One of success stories in this area is the Technology Center for Security Analysis of System Software that we have opened with the support of FSTEC of Russia. More than 60 companies and universities are analyzing source code of the Linux kernel and other critical components to find vulnerabilities and critical errors. This activity is already resulted in more than 500 patches applied to master versions of those programs.

In 2024 ISP RAS has started to coordinate the Rudoo consortium that unites developers of open source ERP solutions. In the same year, jointly with Kurchatov Institute and Joint Institute for Nuclear Research, we have founded a consortium for IT support technologies of the megascience research facilities. We have also partnered with the company "System Solutions," and this important step resulted in creating the ACloud industrial platform for storing and processing data, with the Asperitas system as its base.

We gather communities for creating brand new and developing existing technologies, e.g. we unite companies and scientific organizations within the Research Center for Trusted Artificial Intelligence and the world-class research center "Digital Biodesign and Personalized Health Care." We continuously share our expertise, which is one of the main goals of the Academy of Sciences. Starting from 2019, ISP RAS is named Center of Excellence for Secure Development Lifecycle and Certified Software Code Analysis by the FSTEC of Russia. In 2024, we have signed an agreement with Federal Financial Monitoring Service for developing a national system for countering money laundering.

Thirty successful years resulted in ISP RAS being a distributed center of excellence for the most pressing fundamental problems of system programming and artificial intelligence. We plan to spend the next five years in constant development of tools and communities for ensuring technological independence of our country.

ISP RAS: AN INNOVATION ECOSYSTEM

ISP RAS activities are aimed at deploying fundamental research results in industry. The institute's business model consists of three closely related activities producing a synergistic effect:

- project-oriented fundamental and applied research aimed at creating new technologies (under contracts with Russian and foreign companies, the Ministry of Science and Higher Education of Russia, RAS programs, grants from Russian Science Foundation and from Advanced Research Foundation, etc.);
- deploying new technologies in partner companies and developing innovations based on industry feedback;
- educating students and postgraduates based on developed technologies (while participating in the institute's research and industrial projects).

This model of industrial research plus education is well known and applied in research laboratories of leading universities (Stanford, MIT, Berkeley, Carnegie Mellon) and industrial giants (IBM, Intel), as well as in state research centers (INRIA, Fraunhofer). When implemented effectively, the model solves the problem of the gap between science and industry, and produces highly qualified specialists in system programming.

FUNDAMENTAL RESEARCH

Fundamental research and experimental works are necessary elements of the institute's activities, allowing it to move in line with the latest trends in the IT world, as well as generate its own ideas for projects with its business partners. ISP RAS works on a large number of scientific and educational programs and cooperates with leading Russian and foreign universities and scientific centers. This allows to provide high quality research results, while ISP RAS reputation in academic and university circles makes it possible to introduce domestic technologies to international markets.

ISP RAS publishes its own journal called "Proceedings of ISP RAS," indexed in the Russian Science Citation Index (RSCI).

The institute is also responsible for publishing and editing the RAS journal called “Programming”. Both are included in the journal list of Higher Attestation Commission (the VAK).

DEPLOYMENT

ISP RAS deploys its research results in various industrial and research enterprises, which use and promote the institute’s technologies. Most of the work is carried out under contract with long-term partners, the most important of which are Kaspersky Lab, Security Code (Kod Bezopasnosti), Open Mobile Platform, SberTech, JSC NPO RusBITech, GosNIIAS, and Bazalt SPO. Currently, the institute’s technologies are used in more than 200 companies.

SCIENTIFIC COLLABORATION

Long-term cooperation with ISP RAS can be organized in a form of a joint laboratory. Having permanent funding, they allow planning flexibly available resources as well as increasing competencies in the newly emerging areas of system programming and organizing the training of young specialists with the skills needed by partners.

Since 2009, the institute has operated a joint laboratory with Samsung (aimed at program analysis, including security in the context of Android and Tizen OS, as well as research on the application of artificial intelligence and data analysis methods to software engineering tasks). In 2019, joint laboratories with Huawei were opened. Current ISP RAS research laboratories include:

- Laboratory for digital modeling of complex technical systems, which aims at developing software for nD-modeling industrial tasks;
- Linguistic platform laboratory (jointly with Institute of Linguistics of RAS and other organizations), where the work on documenting endangered languages is underway;
- Young researcher laboratory for federative learning (supported by the Ministry of Science and Higher Education of Russia).

CENTERS

The important mission of ISP RAS is creating and moderating multidisciplinary communities. Three such centers have been launched and are currently in operation:

- World-class Research Center (WCRC) “Digital biological design and personalized healthcare”;
- Research Center for Trusted Artificial Intelligence (within the federal project “Artificial Intelligence”);
- Technology Center for Security Analysis of System Software (jointly with FSTEC of Russia).

INTELLECTUAL PROPERTY

ISP RAS business model suggests that IP rights are either retained by the institute or transferred to an open source developer community under special agreements. Taking into account the specifics of this model, ISP RAS developed a unique license based on the direct financing by the customer of the research and development for the licensed technology (instead of paying royalties). The customer gets non-exclusive rights for using the technology, and the institute retains the exclusive IP rights. For some cases, decisions on managing IP rights are made individually based on long-term collaboration perspectives.

OPEN SOURCE

One of the most important components of the created ecosystem is the widely used open source software that is absolutely necessary for modern system programming. Open source is considered as:

- a tool that provides legitimate free access to all modern technologies, including ready-to-use software products and open standards;
- an ability to ensure the institute innovative research without outsourcing contracts but interacting with global market of products and services;
- a powerful educational resource, as the environment and infrastructure of international open source projects can be used to train engineers.

Scientific activity implies the result’s openness and the visibility of its author, which often contradicts IT corporate policies. For ISP RAS, the openness of research results is both motivation for work, and a tool for promoting the institute’s technologies. Open research means that each young researcher is visible in the international IT community. Their contribution and reputation are their capital, and the institute does everything to ensure that this capital grows as quickly as possible.

EDUCATION

The ISP RAS innovation ecosystem cornerstone is educational activity, which is performed in several directions:

- Cooperating with leading universities. ISP RAS manages chairs of system programming in Moscow State University, MIPT, and HSE. Starting from their first year, students attend system programming lectures and corresponding practical lessons. In the third year, students join the departments for system programming and, while continuing to attend lectures, start to work in special seminars, get acquainted with the institute’s scientific directions, participate in projects and receive a special scholarship. By the time of graduation, many students have scientific publications and become system programming experts.

ISP RAS researchers are constantly advancing cybersecurity field as a scientific direction, which results in updating education courses and bachelor programs. For example, in 2021 ISP RAS started academic advising to modernize the software engineering bachelor program at the Faculty of Computer Sciences at Higher School of Economics. Cooperation with other universities is rapidly expanding, e.g. with Bauman Moscow State Technical University, Chuvash State University, Saratov State University, Moscow Aviation Institute. Joint activities include creating and teaching educational programs and courses, mentoring theses, participating in seminars. In 2022, together with the Russian Foreign Ministry’s State University of International Relations (MGIMO), a master’s program “Data Analysis and Dynamics of International Processes" was launched; the joint laboratory of intelligent data analysis was created as well, where the ISP RAS Talisman platform is being advanced and used. In 2023, with the support of ISP RAS, a joint research group was established at the Matrosov Institute of Dynamics and Theory of Systems of the Siberian Branch of the Russian Academy of Sciences. The main activities include the development of software tools for the analysis of electronic documents and natural language processing.

- Scholarship program. In support of educational processes, ISP RAS launched a special scholarship program for students and postgraduates of MSU, MIPT, HSE, Novgorod State University, Russian-Armenian University and others.
- ISP RAS postgraduate study helps gain practical experience and learn new technologies at the same time. Postgraduates are actively involved in education: they organize seminars and practical classes for students, supervise term papers and theses. With that kind of experience, they usually become leaders of small research groups.
- System programming labs network. Currently, ISP RAS external labs are working in Yerevan, Veliky Novgorod, Orel, and Moscow (in Plekhanov Russian University of Economics). The laboratories attract successful students (including postgraduate students), and involve them in the development of promising technologies in close cooperation with industry.

CONFERENCES

ISP RAS organizes a number of annual events, including:

International ISP RAS Open Conference: https://www.isprasopen.ru/en

OS DAY, a conference on science and practice (jointly with other organizers): https://www.osday.ru/

International Ivannikov Memorial Workshop: https://www.ivannikov-ws.org/en

International Conference On Data Science In Medicine (jointly with other organizers): https://digital-med.ru/en

SYRCoSE Software Engineering Colloquium: http://syrcose.ispras.ru/

2024. WORLD-CLASS RESEARCH CENTER (WCRC) DIGITAL BIODESIGN AND PERSONALIZED HEALTHCARE

JOINTLY WITH SECHENOV UNIVERSITY, INSTITUTE OF BIOMEDICAL CHEMISTRY, YAROSLAV-THE-WISE NOVGOROD STATE UNIVERSITY, AND INSTITUTE FOR DESIGN-TECHNOLOGICAL INFORMATICS OF RAS

MOST IMPORTANT RESULTS OF 2024:

THE WCRC "DIGITAL BIODESIGN AND PERSONALIZED HEALTH CARE" CLOUD PLATFORM NOW SUPPORTS WORKING IN FEDERATIVE MODE.

The platform allows increasing computational resources by using IT infrastructure of other organizations. The resources are managed with domain-specific orchestration. Support for creating virtual resources via OASIS TOSCA descriptions has been implemented in the Michman orchestrator, which is a part of the Asperitas cloud platform. Resources can be created in Openstack and Yandex Cloud based cloud service deployments.

The federative work mode provides centralized management of distributed resources and allows scheduling tasks on computational devices based on configured rules.

THE FIRST RUSSIAN MODEL FOR 12-CHANNEL ECG CLASSIFICATION HAVE BEEN TRAINED USING A FEDERATIVE LEARNING APPROACH.

Federative learning is designed for projects with multiple participants each having own data sets. It allows for collaborative model training without exchanging data and provides new opportunities for partnering around AI-related projects.

ISP RAS, Yandex, and Sechenov University used federative learning to create a neural network that detects atrial fibrillation, one of more widespread cardiac pathologies, using ECG data. The technology shows high specificity and sensitivity.

The mode was trained using two independent ECG datasets from Sechenov University and ISP RAS. Both partners executed a few training rounds internally and then shared results within the common platform.

Yandex Cloud and ISP RAS backed up the technical side of the project. Yandex Cloud planned implementation stages, suggested the technology stack, deployed a unified training environment and calculated the needed resource amount. ISP RAS developed the model and adapted it for an open-source federative learning framework. Sechenov University evaluated the model's quality.

Now companies working with sensitive data (such as medicine, finances etc.) can implement similar projects.

THE FULL-SCALE PRE-PRODUCTION OPERATION OF THE PLATFORM HAS BEEN STARTED.

In 2024, the pre-production mode has been launched for all WCRC participants (Sechenov University, Institute of Biomedical Chemistry, Yaroslav-the-Wise Novgorod State University, and Institute for Design-Technological Informatics of RAS). Other participants have also been invited. The following problems are being tackled with the platform's tools in pre-production:

- determining risk probability for malignant tumors;
- detecting hypertensive retinopathy using digital images of fundus oculi;
- detecting lymphovascular invasion of lung cancer on histologic scan images of lung cancer;
- predicting protein expression level by transcript expression;
- detecting skin cancer and melanoma via analyzing dermatoscopic pictures;
- other biomedical tasks.

Based on the WCRC platform, the work on 12-channel ECG markup by high-skilled physicians is being further performed. This would allow creating a high-quality data set for fine-tuning the 12-channel ECG model.

PREVIOUSLY AT WCRC:

CLOUD PLATFORM WCRC "DIGITAL BIODESIGN AND PERSONALIZED HEALTH CARE" WAS CREATED AND DEPLOYED AT SECHENOV UNIVERSITY.

The WCRC platform developed by ISP RAS offers the following services:

- basic cloud services (e.g., on-demand virtual servers and blockchain appliances);
- services for collecting, storing and analyzing big medical data;
- services for medical data annotation and for applying machine learning algorithms to solve problems in the biomedical domain;
- services to support collaborative research processes.

The platform is implemented on the basis of the Asperitas cloud environment (ISP RAS). In 2023, the platform included functionality for the formation of a scientific knowledge base in medical research using technologies for the collection and analysis of large amounts of data, implemented on the basis of the Talisman information and analysis system (ISP RAS). Testing of cloud services was performed on the example of web-labs for analysis of electrocardiogram data and histological images.

The WCRC platform can be deployed on the basis of the ISP RAS cloud infrastructure or third-party cloud infrastructure for current biomedical tasks developed within the WCRC, or adapted to the tasks of other medical domains.

THE WORK ON DEVELOPMENT AND IMPLEMENTATION OF THE NEURAL NETWORK MODEL OF 12-CHANNEL ECG CLASSIFICATION IS COMPLETED.

The neural network model of 12-channel ECG classification was trained on data from different regions (Republic of Tatarstan, Moscow, Velikiy Novgorod), integrated into the "Unified Cardiologist" system, and tested on ECG data from the Republic of Tatarstan. A cooperation agreement was signed with A.S. Puchkov Emergency and Urgent Medical Aid Station. Several tens of thousands of ECGs received from the station were analyzed. The quality of prediction models is comparable to 10-second 12-channel ECGs. Test protocols have been signed. The product is currently undergoing registration as a medical device.

A MOCK-UP OF A 12-CHANNEL ECG MARKUP SYSTEM HAS BEEN DEVELOPED (HTTP://ECG1.ISPRAS.RU)

High-quality standardized markup based on a predetermined list of pathologies helps achieve a high degree of agreement between specialists. The layout of the markup system has been prepared for integration into the ISP RAS Asperitas cloud platform for transparent scaling of ECG storage and analysis capacities, but it can be also integrated into a third-party cloud ecosystem.

A NEURAL NETWORK MODEL OF ENDONET CELL NUCLEI DETECTION ON HISTOLOGICAL PREPARATIONS WAS TRAINED

The neural network was trained on the EndoNuke marked histological data set assembled jointly with the partners (PFUR, State Clinical Hospital No 31, V.I. Kulakov Institute of General Medicine, Novgorod State University, and the Research Institute of Human Morphology). The core detection model is embedded in the open software platform for bioimage analysis QuPath using the ISP RAS Fanlight technology. The modified QuPath platform and the open source CVAT image markup system are prepared for integration into the ISP RAS Asperitas cloud platform.

2024. TECHNOLOGY CENTER FOR SECURITY ANALYSIS OF SYSTEM SOFTWARE

JOINTLY WITH FSTEC OF RUSSIA AND LEADING COMPANIES
PORTAL.LINUXTESTING.RU

MOST IMPORTANT RESULTS OF 2024:

ORGANIZATIONAL ACHIEVEMENTS:

- more than 50 organizations joined the consortium for supporting the Technology Center.

Previously in 2023:

- infrastructure for examining open source system critical components have been deployed.

METHODOLOGICAL ACHIEVEMENTS:

- methodologies for examining the Linux kernel, OpenSSL, NGinx, QEMU, libvirt, podman, .NET6 Runtime, ASP .NET Core, Node.js, Python, OpenSSH have been created.

Previously in 2023:

- recommendations were prepared on how to configure the kernel to improve its security;
- recommendations were prepared for configuring trusted kernel boot.

TECHNOLOGICAL ACHIEVEMENTS:

- two Linux kernel branches are being maintained using stable versions of Linux 5.10 and 6.1;
- more than 45 thousand warnings of the Svace static analysis tool (developed by ISP RAS) were analyzed, and among them more than 17 thousands were independently cross-verified;
- more than 420 patches have been prepared and merged into the main kernel branch, and more than 120 patches are merged into the main branches of OpenSSL, QEMU, libvirt, CPython, Lua, .NET6 Runtime and others.

TECHNOLOGY CENTER PARTNERS:

- JSC Aladdin R.D.
- Aideco LLC
- Company “Aktiv”
- ANKAD LLC
- JSC ASKON
- JSC ATLAS
- Bars Group
- Basalt SPO
- Basis
- JSC Baikal Electronics
- BellSOFT LLC
- Bi.Zone
- CloudX Group
- “Confident” Ltd.
- Digital Technologies Scientific Center
- JSC NPO Echelon
- JSC Electra
- JSC Flant
- JSC FINTEKH
- Fobos-NT Center
- Garda Technologies LLC
- Inferit LLC
- JSC InfoTeX
- InfoWatch
- ITB LLC
- JSC IVK
- Kaspersky Labs
- Keysystems Group
- “Kod bezopasnosti” LLC
- JSC NTTs “Module”
- JSC MCST
- MT Integration
- NGR Softlab
- JSC NIKIRET
- JSC NPPKT
- Open Mobile Platform LLC
- Orion Soft
- PLC Technology
- Postgres Professional
- RASU JSC
- RED SOFT LTD
- Rosa
- RusBITech-Astra LLC
- R-Vision LLC
- FSUE RFYaTs-VNIIEF
- Safib LLC
- Solar Security
- JSC MVP “SVEMEL”
- TekhArgos LLC
- Usergate LLC
- VK Tech
- Yadro
- “YANDEX.CLOUD” LLC
- ZAO ZET

EDUCATION PARTNERS:

- Lomonosov Moscow State University
- Moscow Institute of Physics and Technology
- HSE University
- Vologda State University
- Moscow Power Engineering Institute
- Bauman Moscow State Technical University
- Voronezh State University
- Ilya Ulyanov Chuvashia State University
- MIREA - Russian Technological University

2024. RESEARCH CENTER FOR TRUSTED ARTIFICIAL INTELLIGENCE (RCTAI)

IN COOPERATION WITH THE MINISTRY OF ECONOMIC DEVELOPMENT OF RUSSIA,

THE ACADEMIC COMMUNITY (MIPT, SKOLTECH, MOSCOW STATE UNIVERSITY MEDICAL RESEARCH AND EDUCATION CENTER, MSU FACULTY OF MECHANICS AND MATHEMATICS, INNOPOLIS UNIVERSITY, NIZHNY NOVGOROD LOBACHEVSKY STATE UNIVERSITY, PSYCHOLOGY INSTITUTE OF THE RUSSIAN ACADEMY OF SCIENCES, JOINT SUPERCOMPUTER CENTER OF THE RUSSIAN ACADEMY OF SCIENCES),

AND THE IT INDUSTRY (KASPERSKY LAB, EC-LEASING, INTERPROCOM, TECHNOPROM).

THE MOST IMPORTANT ACHIEVEMENTS OF 2024:

- A unified methodology and recommendations for developing trusted systems that use AI have been created.
- A stable version of the cloud-based platform for analyzing and developing trusted systems using AI technologies has been released. The new version takes into account the feedback from beta testing done by the RCTAI industrial partners. The platform combines tools implementing MLOps best practices and novel techniques for addressing fundamentally new threats arising at all stages of the AI technologies life cycle:
- trusted machine learning frameworks and libraries;
- tools for testing machine learning models and protecting them from adversarial attacks in production;
- tools for detecting and correcting model bias;
- tools for explaining models;
- tools for detecting anomalies and data drift in data sets;
- tools for detecting and removing malicious code and backdoors from pre-trained models;
- tools for protecting trained models from unauthorized copying and data set extraction.

  All of the above tools can be deployed separately, which eases deployment into existing business processes.
- New versions of trusted machine learning frameworks have been released: TrustFlow (based on TensorFlow 2.12 and 2.16.1) and TrustTorch (based on PyTorch 1.11.0, 2.0.0, 2.3.0).
- The TrustTorch framework has passed official state testing and can be utilized in production in projects of the Ministry of Defense.
- The RCTAI employees have prepared a trained course for the cloud platform users and taught the course for students of the project session “Trusted AI” of the Sirius University.

1 PROGRAM ANALYSIS AND CYBERSECURITY

SOURCE CODE ANALYSIS, VERIFICATION, TESTING

BINARY CODE ANALYSIS, FUZZING

NETWORK TRAFFIC ANALYSIS

REQUIREMENT MANAGEMENT

ASTRAVER: A VERIFICATION TOOLSET

AstraVer Toolset is a deductive verification system for key software components. It allows developing and verifying security policy models as well as proving the correctness of software modules written in the C programming language. AstraVer is essential for ensuring the required trust levels from ADV_SPM and ADV_FSP assurance families as defined in the ISO/IEC 15408 standard.

FEATURES AND ADVANTAGES

AstraVer Toolset is a set of tools designed for industrial use. It is based on many years of scientific research and combines two verification approaches: at the model level and at the code level. AstraVer includes two separate tools for verification of C components: the stable AstraVer translator similar to Microsoft VCC and Frama-C/WP, but adapted specifically to the code of OS kernels; the advanced next-generation Isabelle/PLRDF verification tool that is based on a novel approach to interactive abstract interpretation.

AstraVer provides:

- An integrated approach to verification, starting from formalization of high-level requirements to analyzing the C source code behavior.
- Modeling functional requirements (formalizing system functional requirements, proving internal consistency and unreachability of insecure states).
- Testing whether functional requirements are satisfied in an implementation, using their formal models to check the correctness of the observable behavior thus evaluating the quality of testing and generated test cases.
- Verification of critical components written in C (requirements' formalization, correctness proof on all possible input values).
- Support for real industrial C code (GCC compiler extensions, bitwise-precise arithmetic, address arithmetic including the container_of intrinsic, function pointers, casting of integers to pointers).
- Solving most important protection profile tasks, such as
  - formal security policy modeling;
  - formal verification of a security policy model internal consistency and of unreachability of insecure states;
  - formal or a semi-formal functional specification development;
  - formal/semi-formal proof of correspondence between the security policy model and the functional specification;

  - formal/semi-formal proof of correspondence between different representations of target software, such as functional specification, design and source code.
- Ability to adjust the toolset for a specific customer to perform the C source code components verification.
- Isabelle/PLRDF, a novel interactive verification tool for C components, which:
  - is based on the Isabelle/Pure logic kernel of the Isabelle/HOL automatic proof system and on Isabelle/PIDE, a proof integrated development environment;
  - supports an expressive specification language based on the HOL logic;
  - lowers verification costs via effective combination of automatic and interactive methods for composing and transforming structured specifications;
  - implements a concept of two-step interactive abstract interpretation of contract specification along execution paths of a code block under verification. The first step is contextualizing the current specification in the new program state with fully automatic application of user-configurable sets of proven lemmas. The second step is re-abstraction, an arbitrary user modification of the resulting contextualized specification with correctness proof for the performed transformations.

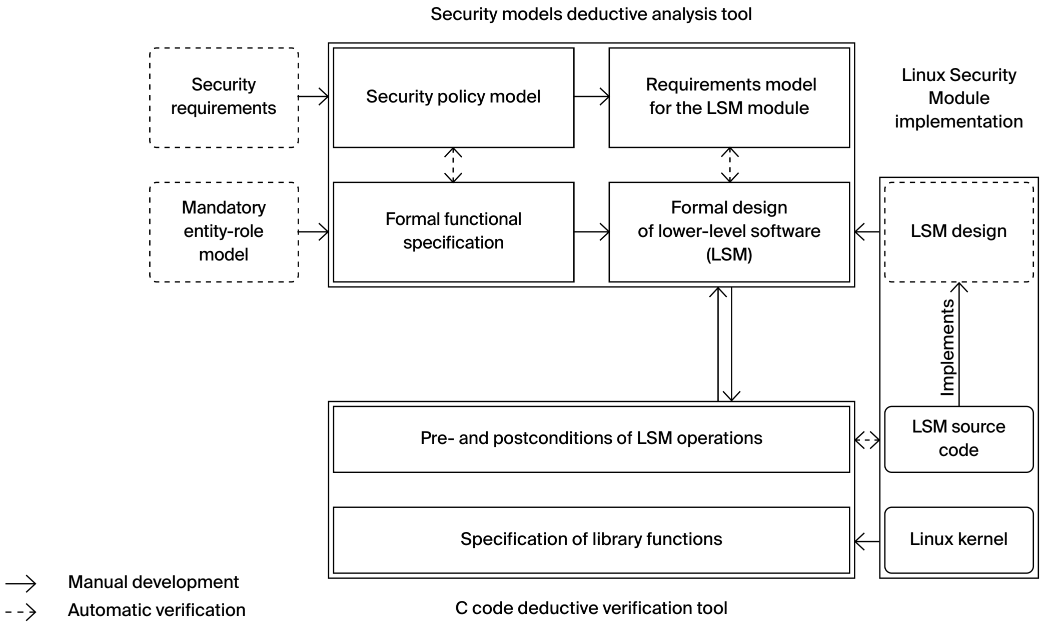
WHO IS ASTRAVER TARGET AUDIENCE?

- Companies developing critical systems, including software in aviation, railway, medical and nuclear power industries.
- Companies that need to certify their software as guided by the ISO/IEC 15408 standard.
- Certification laboratories for information protection software.

ASTRAVER DEPLOYMENT STORIES

AstraVer Toolset was used in the development of access control mechanisms for Astra Linux Special Edition (RPA RusBITech JSC). As a result, this Astra Linux edition has passed the certification for compliance with the FSTEC information security requirements, which are defined for operating systems of the 2A protection profile. The development was based on a mandatory entity-role model of access control and information flows. New model features implemented in Astra Linux Special Edition are constantly verified with AstraVer Toolset.

ASTRAVER WORKFLOW


Security models deductive analysis tool
Security requirements
Security policy model
Requirements model for the LSM module
Linux Security Module implementation
Mandatory entity-role model
Formal functional specification
Formal design of lower-level software (LSM)
LSM design
Implements
Pre- and postconditions of LSM operations
LSM source code
Specification of library functions
Linux kernel
Manual development
Automatic verification
C code deductive verification tool

KLEVER: INDUSTRIAL SOFTWARE MODELS VERIFICATION SYSTEM

Klever is a verification framework that automatically constructs models from source code written in the C programming language. Klever allows specifying various security and safety requirements and verifying them automatically with the pre-configured precision level. The framework is based on includes modular verification, environment modeling, and requirements specification methods. This allows applying formal methods to the industrial software of hundreds of thousands or millions of lines of code. Klever is an open source project: https://forge.ispras.ru/projects/klever

FEATURES AND ADVANTAGES

Klever provides:

- Scalability. Modular program verification allows applying the most rigorous program analysis methods to the large code base. The methods are model checking and symbolic execution.
- Adapting software verification framework to customer needs. Developing specifications for modeling target programs' environments and for detecting violations of program specific requirements. This specific customization is performed in addition to checking regular safe programming rules for the C language.
- Detecting non-trivial errors. Analyzing industrial source code conservatively and with high-precision allows finding all errors of given types and proving program correctness with explicit assumptions.
- Comprehensive representation of found faults. When an error is detected, the verification system provides the detailed error trace that includes concrete variable values and called functions' arguments.
- A convenient multi-user web-interface for setting and running verification and for expert analysis of verification results.

WHO IS KLEVER TARGET AUDIENCE?

- Companies developing safety-critical and security-critical software.
- Certification laboratories.

KLEVER DEPLOYMENT STORIES

The Klever verification system is used for thorough checking of various operating system kernels and drivers. In Linux kernel drivers Klever has found more than 400 errors, including:

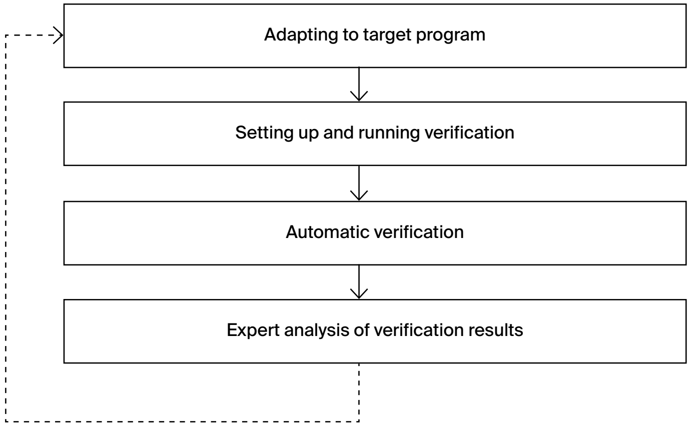
- race conditions,
- deadlocks,
- memory-related errors (buffer overruns, memory leaks etc.),
- incorrect function calls (depending on a certain context),
- incorrect initialization of Linux kernel data structures etc.

Linux kernel developers have acknowledged these errors.

SYSTEM REQUIREMENTS

Ubuntu 20.04/22.04, at least 4 x86-64 CPU cores, 16 GB of memory, 100 GB of disk space.

WORKFLOW


Adapting to target program
Setting up and running verification
Automatic verification
Expert analysis of verification results

MASIW: SUPPORT FOR DESIGNING HIGHLY RELIABLE SOFTWARE SYSTEMS

MASIW is a toolset for developing highly reliable hardware and software systems for avionics, medicine, and other safety critical areas. It is designed for engineers creating airborne hardware/software systems that are developed using the integrated modular avionics (IMA) approach. MASIW can be easily adapted for other application areas.

FEATURES AND ADVANTAGES

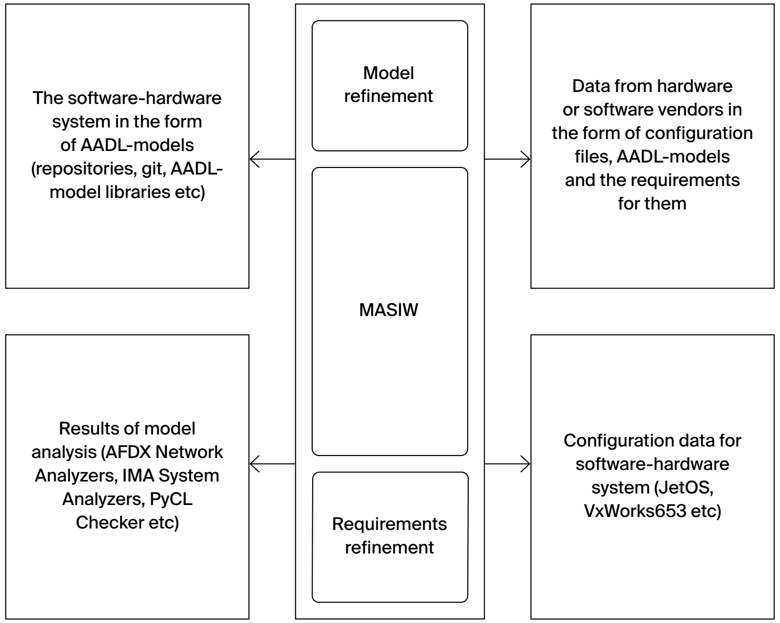
MASIW is the technology for optimizing the development and verification process of complex hardware/software systems. It allows performing a preliminary quality assessment of the product before making the first prototype, as well as performing the fault tolerance analysis. This reduces the risk of errors and defects. MASIW is being developed jointly with GosNIIAS. Despite the presence of the OSATE tool at the start of development, MASIW currently is more functional in the areas of verification, static, and dynamic analysis.

MASIW provides:

- Creation, editing and management of models based on the AADL modeling language:
  - creation and editing of models using the text and diagram editors;
  - support for team development with the ability to track and modify individual elements of a model;
  - support for the third-party AADL models reuse.
- Model analysis:
  - hardware+software system structure analysis: hardware resources sufficiency, interfaces consistency, etc.;
  - verification of the developed system for compliance with the requirements;
  - transmission characteristics analysis for the AFDX networks: message latencies, port queue depth, etc.;
  - generation and analysis of fault trees (FTA) to determine probabilities of high-level fault events;
  - architecture-model based analysis of failures and their consequences, including generation of special descriptive tables;
  - simulation of hardware+software system model with user reports generation including software-in-the-loop execution of on-board partitions with RTOS co-emulated with QEMU and with a universal AADL model simulator.

- Model synthesis:
  - distribution of software applications by computational modules taking into account hardware resource limitations and additional restrictions regarding reliability and security;
  - processor schedule generation (in particular, for ARINC-653 compatible real-time operating systems).
- Configuration data generation:
  - development of specialized configuration data tools based on the provided software interface (API);
  - configuration data generation for the VxWorks653 RTOS and for the AFDX network equipment.
- The ability to extend the toolset by creating own modules.

MASIW WORKFLOW


The software-hardware system in the form of AADL-models (repositories, git, AADL-model libraries etc)
Results of model analysis (AFDX Network Analyzers, IMA System Analyzers, PyCL Checker etc)
Model refinement
MASIW
Requirements refinement
Data from hardware or software vendors in the form of configuration files, AADL-models and the requirements for them
Configuration data for software-hardware system (JetOS, VxWorks653 etc)

MICROTESK: TEST PROGRAM GENERATOR

MicroTESK is a reconfigurable and extendable framework for generating test programs for functional verification of microprocessors and virtual machines. MicroTESK allows automatically constructing test program generators based on a formal specification of an instruction set. MicroTESK supports a wide range of architectures including RISC, CISC, stack and register-based virtual machines.

FEATURES AND ADVANTAGES

MicroTESK is a set of tools that includes a modeling framework (building a model of a system under test based on formal specifications) and a generation framework (building a test program's model based on test templates). It is distributed under the open-source Apache 2.0 license. MicroTESK is available at the ISP RAS website: https://forge.ispras.ru/projects/microtesk. The technology is also presented at http://www.microtesk.org.

MicroTESK provides:

- Using formal specification as a source of knowledge about the system under test:
  - instruction set specification in the nML language (registers, memory, addressing modes, instruction logic, text/binary instruction representation);
  - additional memory subsystem specifications in the mmuSL language (memory buffer properties (TLB, L1, and L2), address translation logic, read/write operations logic).
- Test programs generation based on object-oriented templates:
  - test templates in the Ruby language describing instruction sequences and test data, allowing reuse;
  - automatic template generation via specifications;
  - allows using different generation techniques for instruction sequences and test data (random generation, combinatorial generation, constrained-based generation, etc.).
- Auxiliary tools generation:
  - disassembler;
  - emulator;
  - test oracle.
- Wide range of supported microprocessor architectures:
  - CISC and RISC (test program generators have been developed for RISC-V, ARM, MIPS, and PowerPC architectures);
  - register and stack-based virtual machine architectures;
  - multithread/multicore variants.

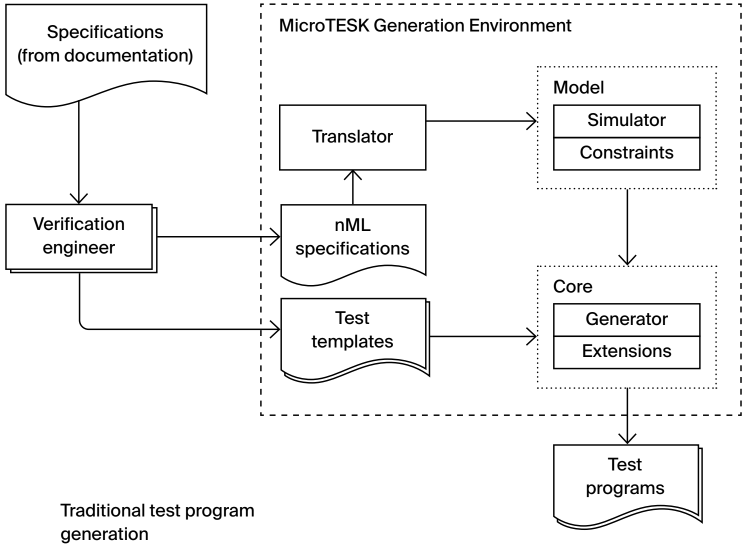
SYSTEM REQUIREMENTS

Java 11, Windows or GNU/Linux-based OS.

MICROTESK DEPLOYMENT STORIES

MicroTESK has been in development since 2007. It was used in various Russian and international projects on developing modern industrial microprocessors and virtual machines, including production projects on verifying ARMv8, MIPS64, and RISC-V microprocessors, the Ark virtual machine.

WORKFLOW



Traditional test program generation

BUILDOGRAPHY: A TOOL FOR IDENTIFYING BUILD ARTIFACTS

Buildography is a tool for identifying program’s build artifacts, namely source code files, dependencies (libraries, header files etc.), compilers and their configuration files.

FEATURES AND ADVANTAGES

Buildography is a lightweight transparent Linux process tracer that is designed for gathering data regarding a program build for further inspection.

Buildography provides:

- Gathering structured data about executed program build in the JSON format.
- Tools for analyzing gathered data. Users can create arbitrary analyzers using the stable output file format provided as a part of the tool API. The following tools are shipped within the distribution:
  - tracing origins of the files built;
  - finding binary files that have been used during build but are not created as a build result;
  - finding source files that have been used during build and are stored outside the specified folders.
- Does not depend on used compilers or build systems.
- File identification via its content hash calculated according to the GOST R 34.11-2012 standard.
- Low overhead due to asynchronous hash sum calculations in separate threads and utilizing the Seccomp-BPF Linux kernel technology to filter intercepted system calls within the kernel itself.

Buildography traces system calls within its child process tree via the ptrace() syscall and gathers the following data:

- command line arguments and working directories of created processes;
- data for identifying executed programs, i.e. executable file paths and content hashes;
- data of files that have been opened during build.

WHO IS BUILDOGRAPHY TARGET AUDIENCE?

- Developers and vendors building software from source.
- Engineers that automate secure development lifecycle (SDLC) processes (DevSecOps engineers).
- Specialists auditing SDLC processes, companies wanting to certify their SDLC processes or software.

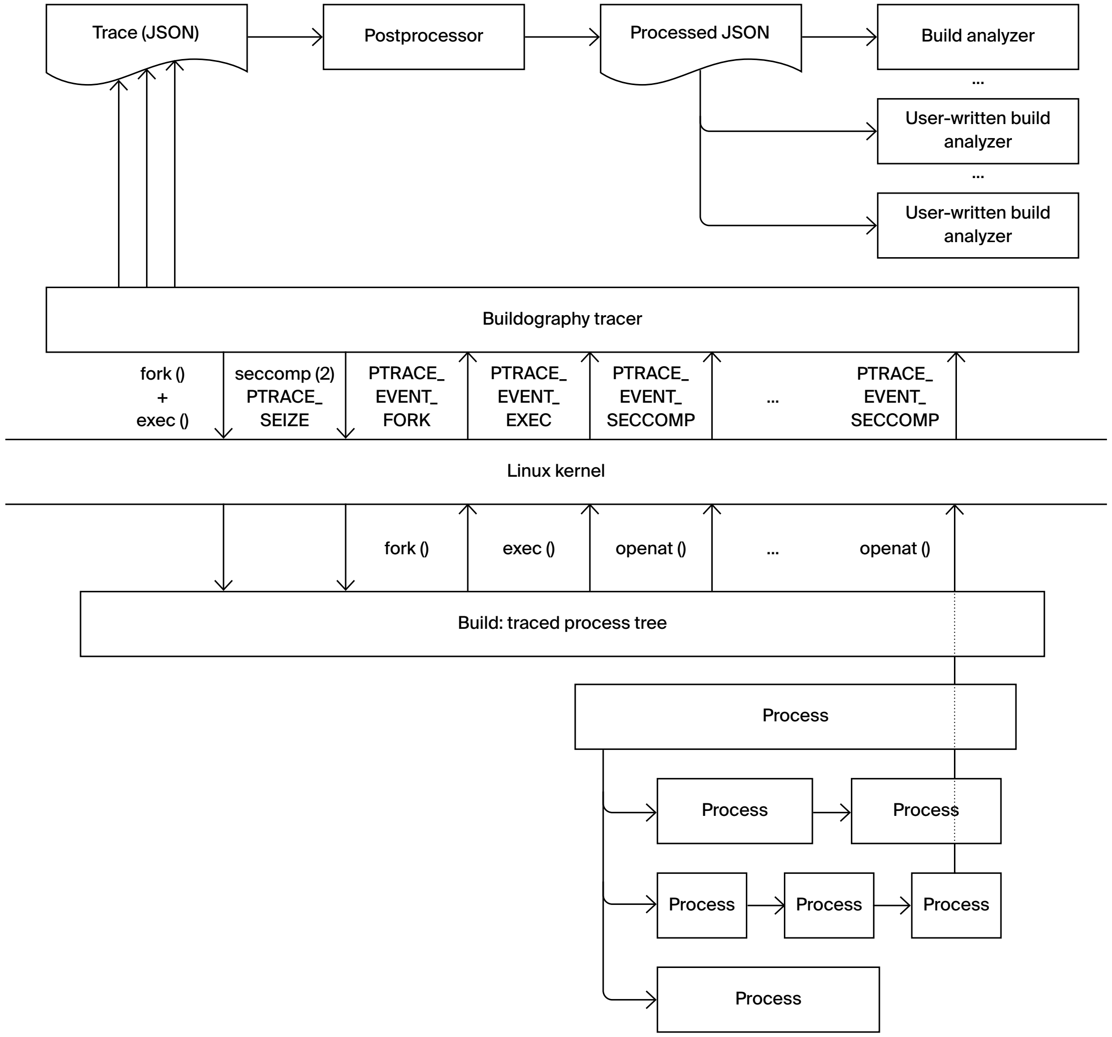
BUILDOGRAPHY DEPLOYMENT STORIES

Buildography is being beta tested within a number of Russian system and cloud software vendors.

SYSTEM REQUIREMENTS

- Linux-based OS with the Linux kernel 4.8 or higher.
- Processor architecture: Intel x86_64, ARM64.

WORKFLOW



SAFEC: SAFE COMPILER

The safe compiler avoids introducing new vulnerabilities in program's binary code when aggressively optimizing (e.g., when making use of source code constructs exhibiting undefined behavior). The compiler tries to avoid excessively restricting optimizations, which allows to avoid the significant performance drop compared to all optimizations being switched off.

FEATURES AND ADVANTAGES

We developed two implementations of the safe compiler that can be used as drop-in replacements for GCC and Clang compilers:

- SAFEC (GCC-based);
- Safelang (Clang-based).

Both compilers are fully compliant with GOST R 71206-2024 “Safe C/C++ compiler. General requirements”.

SAFEC and Safelang provide:

- Refined compiler optimizations for conservative treatment of source code places with undefined behavior, so that for these places program semantics gets defined safely and naturally.
- Forced initialization of uninitialized automatic variables.
- Issuing warnings when detecting constructions with undefined behavior.
- Adding dynamic checks for certain constructs to prevent exhibiting undefined behavior during program execution.
- Diversifying code generation during either compilation or program execution.
- No need to modify either source code or build system configuration, which makes using the compiler as simple as possible.
- Three different safety levels that provide trade-offs between generated code safety and performance. The lowest level is the third; the highest level is the first.

THE SAFE COMPILER PERFORMS THE FOLLOWING ACTIONS

On the third level:

- Avoiding integer overflow, accessing objects via pointers of incompatible types, dereferencing null pointers, using compiler built-ins instead of standard library implementations for input/output functions and for functions working with memory. Avoiding overwrites of automatic variables when calling longjmp function family (Safelang only).
- Detecting division by zero, incorrect bitwise shifts, accesses beyond stack frames, array loads/stores outside of the memory allocated for the array. Detecting automatic variables that are stored in registers when calling longjmp function family (SAFEC only).

On the second level:

- Analyzing arguments of bitwise shifts, redundant memory operations, data alignment when working with vector instructions, address arithmetic when optimizing memory accesses and changing their order.
- Initializing all automatic variables (with zero) that are not initialized explicitly by the user.
- Treating certain compiler warnings as errors and stopping compilation when they are issued.

On the first level:

- Machine code diversification, a unique memory layout for function code either statically during compilation or when performing dynamic linking. The diversification tool was previously a part of the ISP Obfuscator system, and now is integrated in safe compilers.
- Adding machine code that aborts the program when detecting undefined behavior during program execution (sanitization) in the following situations:

1 Integer and floating point operations:
   - loading a non-boolean value in a boolean variable;
   - floating point conversion that results in either integer or floating point overflow;
   - performing a bitwise shift with a negative shift value or with a shift value that is equal or greater than the shifted type width;
   - signed integer operation with the result that is non-representable in the output type;
   - integer division or module with the divisor equal to zero.

2 Pointer and array operations:
   - loads/stores via incorrectly aligned or null pointer;
   - array loads/stores using the address outside of the memory allocated for the array;
   - passing null pointer as a function parameter marked with the nonnull attribute;
   - address arithmetic resulting in integer overflow;
   - returning null value out of function that is marked with the returns_nonnull attribute;
   - allocating an automatic VLA array with incorrect size (zero or negative).

3 Function operations:
   - a function pointer call via a pointer whose declared type does not match the function declaration;
   - returning from a non-void function without actually executing the return statement;
   - calling a compiler built-in with incorrect arguments;
   - reaching a program point during program execution that is marked in the source code as unreachable.

MACHINE CODE DIVERSIFICATION (THE FIRST SAFETY LEVEL)

Diversification is a set of technologies to prevent mass exploitation of vulnerabilities resulting from bugs or backdoors. If an attacker was able to attack one of the devices with the same software installed, the others will remain protected thanks to the changes made to the code.

The diversification tool provides:

- Fine-tuning the balance between the degree of obfuscation and the level of performance (when applied to protect against reverse engineering). Minimum 1.2x slower performance, maximum 8x slower performance.
- Using the original control flow integrity method (CFI), which successfully resists most code reuse attacks (ROP, JOP, ret-to-plt, etc.). Implementation of the CFI method based on the GCC compiler resulted in average slowdown on the SPEC CPU2006 test suite of about 2%, which is noticeably lower than that of traditional methods.
- Conflict-free compatibility with other software protection tools (including the ASLR system mechanism).

- Two diversification approaches, static and dynamic diversification.
  - Dynamic code diversification guarantees the same code on all devices (for example, due to mandatory certification) and allows moving up to 98% of the code with a small increase in its size and about 1.5% performance degradation. Its advantages include:
  - shuffling with function granularity (as opposed to ASLR and Pagerando technologies, which only move large blocks of code);
  - shuffling functions in the whole system, except for the kernel, and avoiding conflict with anti-viruses (which is an advantage over the similar technology Selfrando developed for the Tor Browser);
- Static code diversification produces a unique executable file on each compilation based on the specified key. The advantages of this method include:
  - no increase in binary code size (especially important for Internet of Things);
  - performance degradation is close to zero;
  - due to working inside the compiler rather than ex post facto in the linker, an extended set of diversifying transformations can be applied and tuned with more flexibility.

WHO IS THE SAFEC TARGET AUDIENCE?

- Operating system developers.
- Companies developing high-level safe and secure software.

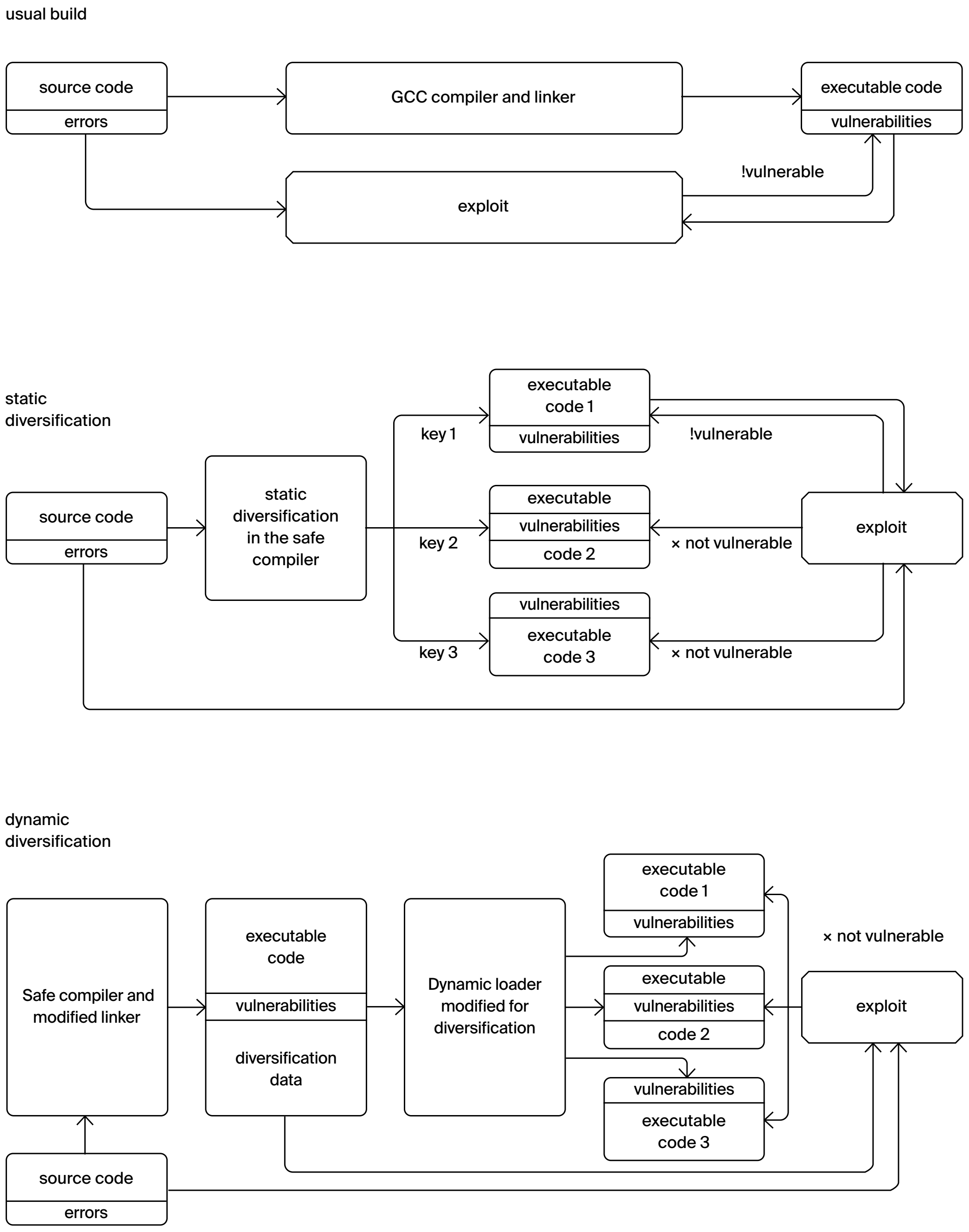
SAFEC DEPLOYMENT STORIES

The safe compiler is deployed in a number of Russian companies and government institutions. The diversification tool is deployed in OS “Zircon,” which is used by Ministry of Foreign Affairs and the Border Guard Service of the Federal Security Service of Russia.

SUPPORTED PLATFORMS

- Operating systems: Linux-based OS, Windows (MinGW, SAFEC only).
- Hardware architectures: x86 (32/64), ARMv7, ARM64, RISC-V 64.

DIVERSIFICATION TOOL WORKFLOW



SVACE: STATIC ANALYZER

Svace is an essential tool of the secure software development life cycle, the main static analyzer that is used in Samsung Corp. It detects more than 70 critical error types. Svace supports C, C++, C#, Java, Kotlin, Go, Python, Scala, and Visual Basic.NET; JavaScript is supported in the lightweight analysis. Svace is included in the Unified Register of Russian Programs (No.4047). It is distributed with the Svacer web interface (Svace History Server).

FEATURES AND ADVANTAGES

Svace is an innovative technology based on years of research that constantly evolves for customer's needs. It combines the key qualities of foreign competitors (Synopsis Coverity Static Analysis, Perforce Klocwork Static Code Analysis, Fortify Static Code Analyzer) with the unique open industrial compilers usage to provide the maximal support level for new programming language standards.

Svace provides:

- High-quality deep analysis:
  - accurate representation of the source code (due to integration with any build system);
  - symbolic execution: full path coverage taking into account connections between functions when searching for complex defects;
  - calling context sensitivity within interprocedural analysis, data flow analysis, tainted data analysis, call statistics analysis;
- Scalability and high speed:
  - parallel analysis using all available processor cores;
  - ability to analyze software with the code size of tens of millions of lines (analysis of the Tizen 7 mobile OS having 57 million lines of code takes 7-8 hours using the main Svace engine and 9–10 hours using all engines);
  - supporting incremental system analysis in addition to the full analysis mode (performs a quick re-analysis of recently changed source files); caching intermediate analysis results (allows to avoid reanalyzing program parts that did not change).
- Automatic warning assessment (true/false probability) using machine learning techniques that are based on source code structure and past review data (for C/C++, C#, and Java).
- Flexible analysis customization:
  - configuring existing detectors as well as writing individual ones available exclusively to a customer;
  - configuring sources and sinks of tainted data for precise search of critical errors important in the customer's environment;
  - an API for developing user plugins acting as detectors.

- Accelerated adaptation to new environments and tools (adding new compilers within 1-2 weeks, in complex cases up to 2 months).
- Full compatibility with GOST R 71207-2024 "Software static analysis. General requirements" as well as with regulatory documents and requirements (FSTEC of Russia).
- Can be used for adhering to the GOST R 56939-2024 requirements and to the requirements of the newest versions of the FSTEC regulation document mandating software vulnerability detection process (when certifying software within Russia).

Svacer provides:

- Review and reports:
  - Extensive features for comparing and reviewing analysis results, code and warning trace navigation, user customized filters and their configuration via an API;
  - Generating reports in PDF, CSV, JSON formats;
  - Annotating results with user files and attributes;
  - Reviewing multiple branches of the same project;
  - Supporting intellectual transfer of review data between projects, project versions and branches, which can be configured via user templates;
  - Reviewing directly in source code via custom format comments (editing mode), supporting history of modifications for the comments;
  - Flexible user interface with tab support.
- Sharing, collaboration and management:
  - Rich role model allowing flexible data access rights for users and organizations;
  - Dashboards with review statistics and user activity;
  - Project groups and group operations support when working with users, projects, review data etc.;
  - LDAP support for user authentication;
  - Mail notifications, event subscription via gRPC API;
  - API support for accessing any data;
  - Importing and exporting data: reviews, source code, comments, detector configuration (including custom user detectors).
- Fulltext search for warnings, detectors, comments, and snapshots (including user-defined attributes) with filtering by projects, branches, and snapshots.
- Web-based Eclipse/Theia IDE for simpler work with code.
- CI/CD integration:
  - Visual Studio Code integration, standard CI/CD process integration via command-line interface.
  - Support for working in containers.
- SARIF format support allowing to import results of other static analyzers, exporting results, reviews, comments, and source code.

WHAT IS SVACE TARGET AUDIENCE?

Companies focused on development of highly reliable and secure software.
- Companies that need to certify the developed software.
- Certification laboratories.

SVACE DEPLOYMENT STORIES

Svace is the main static analyzer used in Samsung Corp. since 2015. It is used to check the company’s own software based on Android OS as well as the Tizen OS source code. Tizen is used in smartphones, infotainment systems and Samsung home appliances. Since 2017, Svace checks all changes submitted for review and inclusion in the Tizen OS. Since 2020, Svace has been also used by Huawei.

Within Russia, Svace is deployed in more than 200 companies and certification labs, including RusBITech, Kaspersky Lab, Postgres Professional, Security Code, Swemel, and others.

SUPPORTED WARNING TYPES

Svace finds more than 1000 warning types including more than 70 critical errors (buffer overflows, incorrect and null pointers, uninitialized data, tainted data usage, division by zero etc.), many coding defects and style violations. The complete error list and other documentation can be obtained at https://svace.pages.ispras.ru/svace-website/docs/latest/user-guide.html.

SUPPORTED PLATFORMS AND ARCHITECTURES

- Host platforms for the Svace analyzer: Linux/x64 (version 3.10 and later, glibc version 2.17 and later), Linux/ARM 64 (Ubuntu 18.04), Windows starting with 7 SP1 with update KB2533623) and WSL (versions 1 and 2); macOS on x86-64 (starting from 10.10; C# and Visual Basic.NET are not supported); x86 architecture for build capture.
- Target architectures of the analyzed code: for C/C++ that is Intel x86/x86-64, ARM/ARM64, MIPS/MIPS64, Power PC/Power PC 64, RISC-V 32/64, SPARC/SPARC64, Hexagon (code generation via Clang); Elbrus, AEON, TriCore, HIDSP, OpenRISC (code generation via one of the previous architectures); for Go, Windows/Linux-based Intel x86-64; for C#, Java, Kotlin, Python, Scala, Visual Basic.NET host platforms are supported.
- Platforms and architectures for Svacer: x86-64; OS Linux (version 3.10 and later, glibc version 2.17 and later); OS Windows (starting with Windows 10) and WSL (versions 1 and 2); macOS on x86-64 (starting from 10.12 Sierra).

SUPPORTED PLATFORMS AND ARCHITECTURES

- Host platforms for the Svace analyzer: Linux/x64 (version 3.10 and later, glibc version 2.17 and later), Linux/ARM 64 (Ubuntu 18.04), Windows starting with 7 SP1 with update KB2533623) and WSL (versions 1 and 2); macOS on x86-64 (starting from 10.10; C# and Visual Basic.NET are not supported); x86 architecture for build capture.

- Target architectures of the analyzed code: for C/C++ that is Intel x86/x86-64, ARM/ARM64, MIPS/MIPS64, Power PC/Power PC 64, RISC-V 32/64, SPARC/SPARC64, Hexagon (code generation via Clang); Elbrus, AEON, TriCore, HIDSP, OpenRISC (code generation via one of the previous architectures); for Go, Windows/Linux-based Intel x86-64; for C#, Java, Kotlin, Python, Scala, Visual Basic.NET host platforms are supported.
- Platforms and architectures for Svacer: x86-64; OS Linux (version 3.10 and later, glibc version 2.17 and later); OS Windows (starting with Windows 10) and WSL (versions 1 and 2); macOS on x86-64 (starting from 10.12 Sierra).

SUPPORTED COMPILERS

- For C/C++ (up to C++20): GCC (GNU Compiler Collection), Clang (LLVM compiler), Microsoft Visual C++ Compiler, RealView/ARM Compilation Tools (ARMCC), Intel C++ Compiler, Elbrus C/C++ Compiler, Wind River Diab Compiler, Keil CA51 Compiler Kit, NEC/Renesas CA850, CC78K0(R) C Compilers, C/C++ Compiler for the Renesas M16C Series and R8C Family, Panasonic MN10300 Series C Compiler, C compiler for Toshiba TLCS-870 and T900 Family, Samsung CalmSHINE16 Compilation Tools, Texas Instruments TMS320C6* Optimizing Compiler, Digital Mars C and C++ Compiler, Green Hills compiler for ARM, TASKING C compiler for TriCore, CEVA Toolbox for CEVA DSP cores, IAR C/C++ Compiler for ARM / Renesas RL78 MCU, CodeWarrior Development Studio for StarCore DSPs, Open Watcom C/ C++ compiler, Freescale CodeWarrior, Cadence Tensilica Xtensa C/C++ Compiler.
- For C# (up to C#12): Roslyn, Mono.
- For Java (up to Java 21): OpenJDK Javac Compiler, Eclipse Java compiler.
- For Kotlin: Kotlin 2.0.
- For Go: Go 1.23.
- For Python: Python 3.12.
- For JavaScript: JavaScript ES 6.

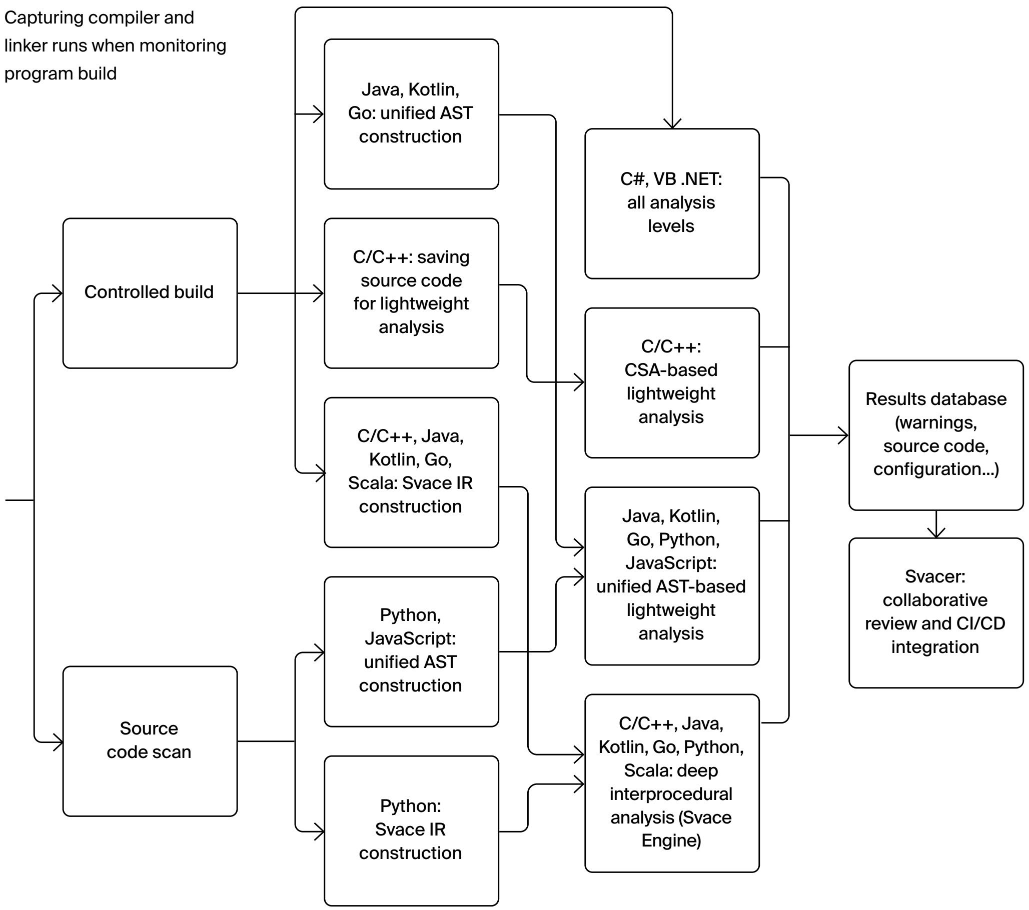
SVACE ARCHITECTURE

1. Build

Capturing compiler and linker runs when monitoring program build

2. Analysis

3. Review


Java, Kotlin, Go: unified AST construction
C/C++: saving source code for lightweight analysis
C/C++, Java, Kotlin, Go, Scala: Svace IR construction
Python, JavaScript: unified AST construction
Python: Svace IR construction
Controlled build
Source code scan
C#, VB .NET: all analysis levels
C/C++: CSA-based lightweight analysis
Java, Kotlin, Go, Python, JavaScript: unified AST-based lightweight analysis
C/C++, Java, Kotlin, Go, Python, Scala: deep interprocedural analysis (Svace Engine)
Results database (warnings, source code, configuration...)
Svacer: collaborative review and CI/CD integration

Scanning directories with source code when analyzing interpreted languages

Building intermediate representations for lightweight analysis (unified AST) and deep analysis (Svace IR)

- lightweight analysis of abstract syntax trees;
- interprocedural analysis (context- and path-sensitive based on symbolic execution);
- tainted data analysis (users can specify sources and sinks for tainted data, which includes tainted function arguments and structure fields).

- syntax highlighting and code navigation;
- warning review (marking true/false positives);
- comparing analysis runs and automatically hiding warnings previously reviewed as false ones);
- group operations for users, projects, and review data;
- API support for data access;
- importing/exporting data.

TESTOS: A SOFTWARE TESTING ENVIRONMENT

TestOS is an environment for unit testing of software on target hardware. It allows to debug software for critical applications on AArch64, ARM, PowerPC, MIPS, RISC-V, and x86 architectures to perform certification and other activities.

FEATURES AND ADVANTAGES

TestOS makes it possible to replace such critical systems verification tools as LDRA, since it is a more flexible tool with active support for domestic products.

Using TestOS ensures running tests on target hardware and generating reports with the trace for each test, with information about the composition and passing status of the test plan and with the coverage of the tested system code both for one test, and for the whole test plan. A convenient development environment for implementing module tests for C functions (for the C18 standard with GNU and Clang extensions) is provided, supporting test scenarios creation and generating stubs and wrappers. Reports are generated in HTML and TXT formats. Debugging code on the target computer is available both with or without JTAG.

TestOS with plugins enabled supports the following:

- collecting function, operator, and branch coverage using GCOV and LLVM Coverage;
- collecting coverage by MC/DC using COVERest;
- performing static analysis with:
  - Clang Tidy;
  - Clang Static Analyzer;
  - Svace.
- Dynamic code instrumentation with LLVM sanitizers:
  - AddressSanitizer (detecting memory handling errors);
  - MemorySanitizer (detecting errors of accessing uninitialized memory);
  - UndefinedBehaviorSanitizer (detecting arithmetic, floating-point, and other undefined behavior errors).

SYSTEM REQUIREMENTS

GNU/Linux distribution on x86_64 architecture (such as Ubuntu 24.04), and Apple macOS 12 or newer as the target machine.

Target machine with at least 2MB RAM on architectures:

- AArch64 (Cortex-A53, Cortex-A55);
- ARM (Cortex-A7, Cortex-A9, Cortex-M4), including i.MX6 or STM32F429 processors;

- PowerPC (e500mc, e500v2, 476FP), including the p1010 or p3041 processors;
- MIPS (MIPS Release 1, MIPS Release 2 / MIPS32, COMDIV), including the 1892VM15AF processor;
- RISC-V (RV32 IMA);
- x86 (Intel Prescott and newer).

The environment is adapted to the customer's equipment if necessary.

TESTOS DEPLOYMENT STORIES

TestOS has been in development since 2019. It is successfully applied for modular software testing for the aerospace industry.

QEMU-BASED SOFTWARE ANALYSIS PLATFORM

ISP RAS Foundation Platform for creating program analysis systems is built on top of open source QEMU emulator. This framework is essential for organizing cross platform development. It supports reverse debugging and introspection features, as well as full system emulation mode for debugging low-level software.

FEATURES AND ADVANTAGES

QEMU supports emulation of more than 10 instruction set architectures (i386 and x86-64, ARM and Thumb, MIPS, PowerPC, etc.). It implements guest debugging via GDB Remote Serial Protocol and is compatible with IDA Pro, GDB, and various IDEs. QEMU supports full system emulation mode that allows debugging low-level software such as a bootloader and an OS kernel. The QEMU source code is regularly checked by static code analysis tools, including Coverity and Svace. Thus performing malware analysis with QEMU is more secure. QEMU with reverse debugging and introspection support is available on the ISPRAS GitHub page: https://github.com/ispras/swat. The developed QEMU automatization tools are available at https://github.com/ispras/qdt, https://github.com/ispras/i3s.

ISP RAS QEMU Foundation Platform provides:

- A record and replay mechanism for a virtual machine:
  - The same VM execution is replayed every time, deterministically. All external events are recorded and replayed by the emulator. It makes finding bugs in multi-threaded applications (race conditions, deadlocks) easier;
  - GDB-compatible reverse debugging is implemented based on the record and replay mechanism. The debugging is performed by restoring previous VM snapshots and searching for the previous breakpoint stop or the previous instruction;
  - The minimum required information is recorded. This allows recording longer for debugging rarely occurring errors;
  - Low performance overhead caused by recording. This enables analysis of software that requires interacting with an uncontrolled external environment in real time.
- VM introspection solution (getting high-level information regarding guest OS work) without any guest OS kernel modifications or installing monitors:
  - Getting the list of executed system calls, accesses to named functions in shared libraries, the list of running processes, the list of open files and loaded modules;

  - Supports all Linux-based virtual machine images as well as embedded software images for various devices;
  - WinDbg server support in QEMU that allows showing guest software information in terms of Windows kernel abstractions. There is no need to enable the OS debugging mode in the guest OS.
- Speeding up QEMU development:
  - Faster development of dynamic analysis tools that can analyze binary code for specific hardware;
  - Automated support for new processor architectures using a machine instruction decoder generator and a C-like language for describing machine instructions semantics;
  - An automatic tool for preliminary virtual machine testing. The tool only requires GNU Binutils and a C compiler;
  - A tool for automating QEMU virtual devices development;
  - VM generation tool in the form of QEMU module source code. The tool can create VMs from both existing devices and new devices out of Python description. The tool provides GUI for sketching the virtual machine;
  - A Python API for an automated debugging via GDB Remote Serial Protocol. It is used to debug QEMU, the guest OS, or both at the same time.
- Convenience and user experience:
  - Easy QEMU extension due to open source code and own ISPRAS toolkit for speeding up development;
  - Binary code analysis without any guest OS modifications;
  - VM introspection mechanism that can be extended using plugins;
  - A convenient API for developing own introspection plugins;
  - Can be easily adapted for specific use cases;
  - Support for latest QEMU versions that have support for newest peripherals and CPUs.

WHO IS ISP RAS FOUNDATION PLATFORM TARGET AUDIENCE?

- Bootloader, driver, OS and other system software developers.
- DevOps teams for software bugs reproduction, cross-platform development, and scalable cloud testing.
- Programmers analyzing potential malware.
- Software certification engineers.

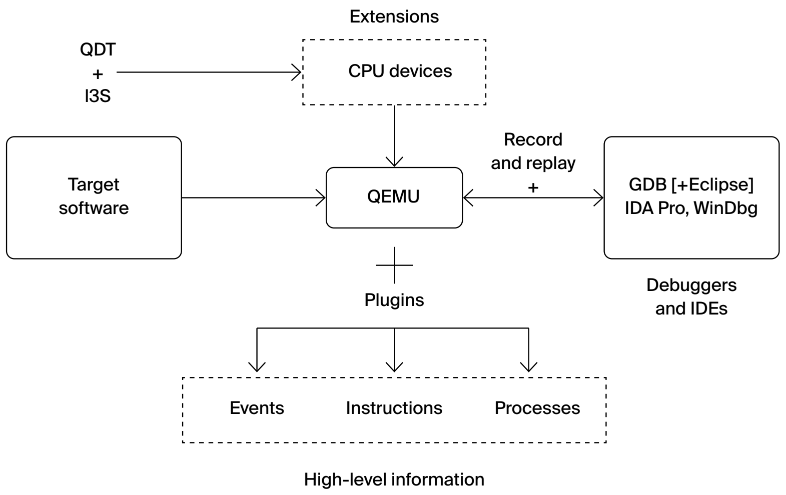
SUPPORTED GUEST PLATFORMS

Emulation of the following ISAs: i386, x86-64, ARM, MIPS, PowerPC, and others.
Guest systems supported by the introspection mechanism: Windows XP (x86), Windows 10 (x86-64), Linux 2.x-5.x (x86, x86-64, ARM, AArch64).

ISP RAS QEMU DEPLOYMENT STORIES

The QEMU community has accepted ISP RAS patches for the record and replay mechanism and added them to the open source QEMU version 3.1.

WORKFLOW



ISP CRUSHER: BINARY CODE DYNAMIC AND STATIC ANALYSIS TOOLSET

ISP Crusher is a toolset that combines various dynamic and static analysis approaches, including fuzzing (using ISP Fuzzer, a fuzzing tool), and symbolic execution (among others, the Sydr tool can act as a symbolic engine). In the near future Crusher will also include the BinSide analyzer, another ISP RAS technology. Crusher allows organizing a development process that is fully compliant with GOST R 56939-2024 and “Methodology for identifying vulnerabilities and undeclared features in software” of FSTEC of Russia. ISP Crusher is included in the Unified Register of Russian Programs (№10468).

FEATURES AND ADVANTAGES

The ISP Crusher core is ISP Fuzzer, a fuzzing tool essential for any fuzzing tests on every stage of software development phase, be it coding, testing, or deployment. The fuzzer finds program errors either with or without source code. It solves the same problems as its global competitors (Synopsys Codenomicon, beSTORM, Peach Fuzzer), but it is more convenient for Russian companies in the import phase-out context.

ISP Fuzzer provides:

- Fuzzing a wide class of software:
  - custom applications, kernel and libraries;
  - embedded software (controllers, IoT devices), COM objects and Windows services;
  - Windows and Linux operating systems;
  - applications in various programming languages: C/C++, Java, Python, C#;
  - x86/x86_64, ARM, MIPS, RISC-V architectures;
  - fuzzing of neural networks. The software reveals cases of erroneous neural network predictions when correctly classified input data is distorted; the detection is done via analyzing neuron activation map when the neural network is working. This improves quality and safety of AI systems, including but not limited to:
    - finding errors in the networks for situations that have not been originally included in the training dataset;
    - finding possible backdoors and malware.
  - fuzz-testing through different input data sources: file, command-line arguments, standard input stream, environment variable arguments, network, direct writing to memory;
  - ability to analyze server and client software running on stateful and stateless protocols;
  - fuzzing protocols by modifying the client: this allows to avoid writing a fuzz client or its specification from scratch when fuzzing the server; a mirror scheme with modifying a server to fuzz the client is supported as well;
  - extensive possibilities for fuzzing software of embedded devices through partial emulation and symbolic execution;
  - browser fuzzing: browser control via Selenium, coverage feedback via Frida;
  - fuzzing applications that require isolation in the docker mode, when each fuzzer instance works in a separate docker container;
  - fuzzing applications in rootfs via the chroot mode.
- Large capacity fuzzing:
  - Support for multi-threaded analysis on both a single machine and distributed ones;
  - ability to distribute input data corpus between fuzzer processes to increase efficiency of their work;
  - support for differential fuzzing;
  - support for collaborative fuzzing with automatic resource load balancing between fuzzers.
- Support for a large set of tool types:
  - static (mostly for C/C++) with GCC/LLVM;
  - static instrumentation of Python bytecode;
  - dynamic (mostly for ELF, PE): DynamoRIO, QEMU (user-mode), TinyInst;
  - based on partial emulation;
  - using Nyx snapshots and snapshot-API;
  - Java applications;
  - C# applications;
  - remote instrumentation (which makes it possible to perform fuzzing of an application running on a remote device).
- Integration with a number of necessary tools of secure software development lifecycle tools developed at ISP RAS:
  - the use of dynamic symbolic execution tool Sydr to improve the efficiency of fuzz-testing;
  - receiving input data to check errors marked by the BinSide static analysis tool in automated mode;
  - receiving data about unstable code blocks and boundary blocks from the BinSide tool;
  - using the data generator that is based on ANTLR grammars to generate the input data corpus.
- Integration with other dynamic analysis tools:
  - with third-party fuzzers, allowing to run a set of different synchronized fuzzers within one fuzzing session, which increases the efficiency of testing;
  - with SymCC and Angr dynamic symbolic execution tools, which makes it possible to get new input data to increase the code coverage of target software;
  - working together with the IDA PRO disassembler (saving the coverage for the Lighthouse plugin, which displays the covered basic blocks in the software, as well as displaying the percentage of covered basic blocks);
  - using the Radamsa fuzzer to generate new data.

- Additional analysis of the received input data:
  - Evaluation of the criticality of found abnormal terminations;
  - ability to launch dynamic analysis systems using new input data: Valgrind, DrMemory, QASan;
  - creation of the coverage profile by source or binary code.
- Extensive options for integrating custom extensions:
  - option to add user-side handlers that will automatically run on new input data;
  - option to add custom mutation transformations (to generate new input data and increase testing efficiency);
  - availability of input data pre-processing and post-processing modules to perform constant transformations of data before sending it to the software to be analyzed;
  - support of custom plugins for sending data over network (plugins allow interacting with client or server software and sending mutated data);
  - support of custom Python scripts to modify options (avoids conflicts when multiple fuzzing processes are running simultaneously);
  - support for custom Python plugins to control the environment for launching the target software (which makes it possible to keep an identical environment at each start-up);
  - support for custom instrumentation plugins (which makes it possible to define arbitrary classification rules for input data based on the target software behavior: definition of normal and crash termination, freezing);
  - ability to describe scenarios for fuzzing software with the user interface.
- Easy extensibility and easy adding new methods within the framework of the existing infrastructure; fast adapting to new tasks.

WHO IS ISP CRUSHER TARGET AUDIENCE?

- Companies developing highly reliable and secure software.
- Companies auditing or certifying software.

SYSTEM REQUIREMENTS

Crusher host system requirements:
- Ubuntu 18.04+, CentOS 6.9+, Astra Linux 1.5+
- Linux kernel 2.6.32+;
- glibc 2.12+.

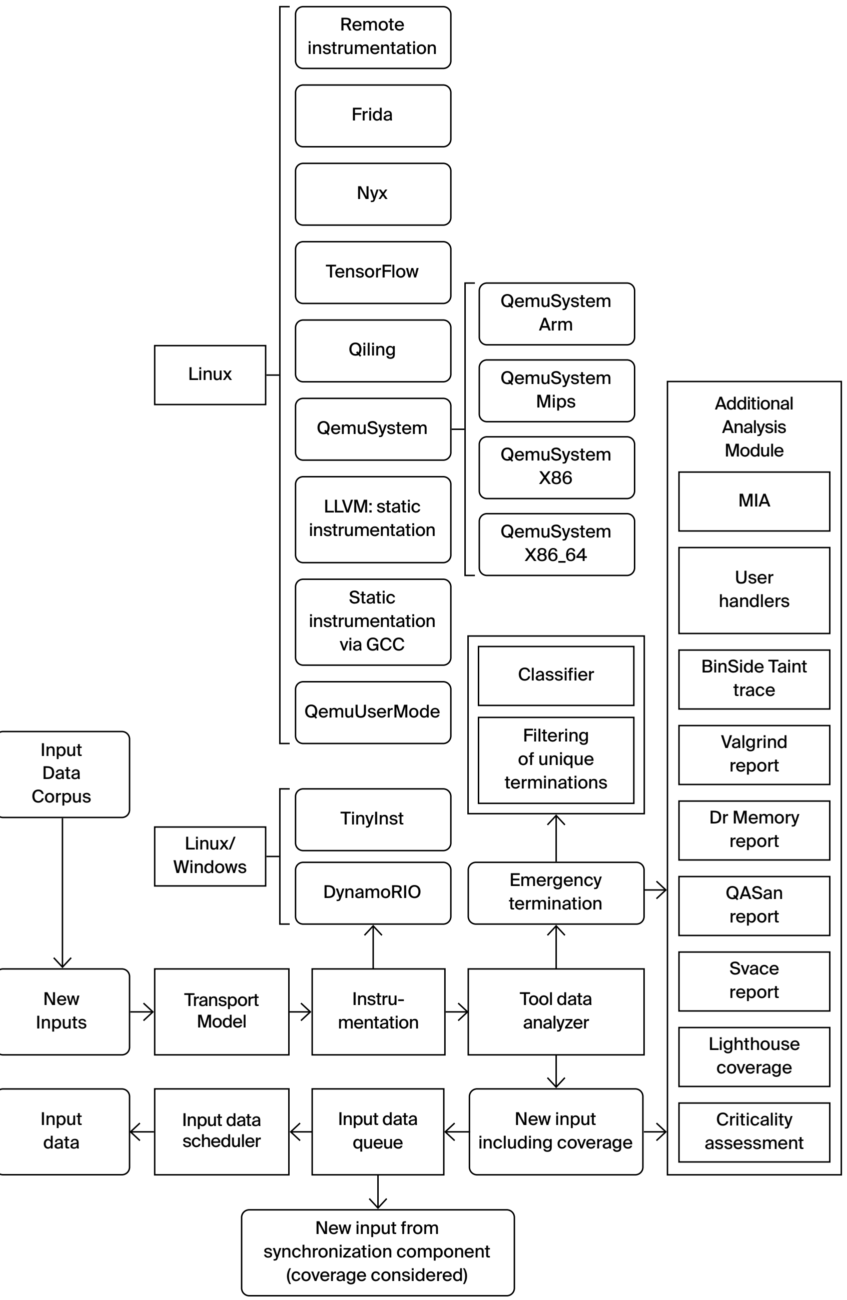
Crusher hardware requirements:
- Intel Core i7 or similar AMD processor;
- 8Gb+ RAM.

Recommended: 32-core+ Intel/AMD CPU (depending on the number of fuzzing threads); 32+ Gb RAM (depending on the fuzzed program requirements).

ISP CRUSHER DEPLOYMENT STORIES

ISP Crusher is used in more than 70 companies and certification labs, including RusBITech, Postgres Professional, Security Code, Swemel, and others.

ISP CRUSHER WORKFLOW



CATALOGUE OF TECHNOLOGIES

BINSIDE: A BINARY CODE STATIC ANALYSIS TOOL

BinSide is a static program analysis platform for finding defects in binary code. It is useful when checking programs without source code, such as closed source third-party libraries.

FEATURES AND ADVANTAGES

BinSide is a binary code analysis platform based on the BinNavi framework. An executable file is analyzed in IDA PRO or Ghidra representation. BinSide provides various analysis types such as defect detection, code clone detection, dynamic analysis optimization, analysis automation, dynamic testing optimization.

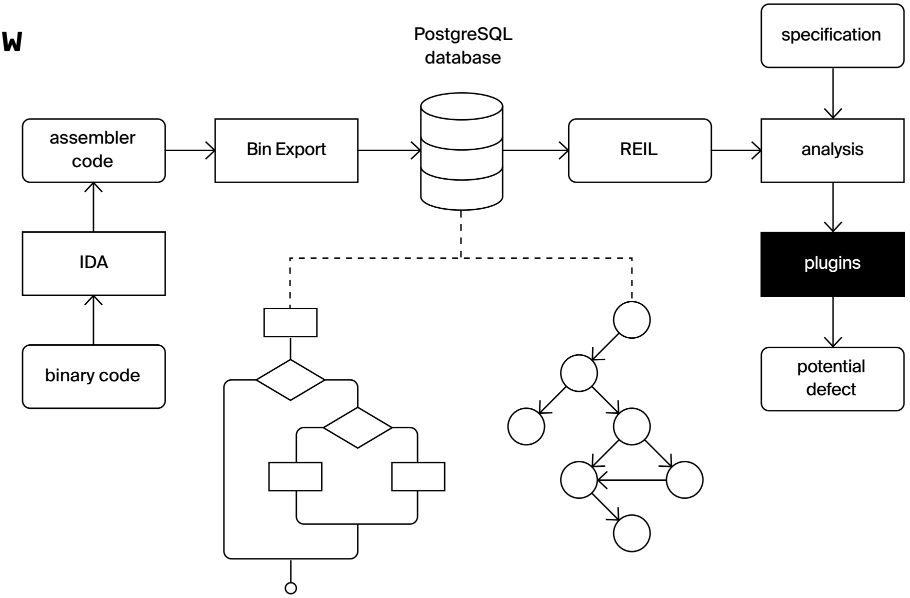
BinSide core provides:

- Easy extension:
  - individual error detectors are written as plugins;
  - the REIL representation of 17 instructions without side effects is used (each assembly instruction is translated into a set of REIL instructions);
  - it is possible to specify the functions' semantics to improve analysis quality.
- Supports analyzing executables and libraries for x86-64, ARM, and MIPS architectures, including drivers.
- Detecting the following CWE types:
  - CWE-121 (Stack-based Buffer Overflow);
  - CWE-122 (Heap-based Buffer Overflow);
  - CWE-134 (Format String Vulnerability);
  - CWE-415 (Double Free);
  - CWE-416 (Use-After-Free);
  - CWE-77 (Command Injection) ;
  - CWE-190 (Integer Overflow).
- Executing the following tasks:
  - data flow and control flow analysis: retrieval of values and pointers, labeled data propagation, determining possible heap states, determining computable edges of the control flow graph;
  - intraprocedural search for defects: search for defects is performed on the basis of the results of intraprocedural analysis of data and control flow, the results of dynamic analysis and manual code markup by the analyst. This is especially useful when analyzing complex software and embedded systems;
  - analysis of all paths regardless of code coverage.
- Integration with ISP RAS technologies:
  - Svacer (if the source code is available);
  - LibraryIdentifier (to search for code clones, e.g. to identify libraries whose code has been used for the executable);
  - the Crusher fuzzing platform.
- Operating system analysis:
  - Determining code plagiarism from an open-source OS;
  - Determining dependencies between OS components and within components;
  - Static analysis of the OS source and binary code;
  - Determining the protection of the executable code in the OS components;
  - Determining the coverage of the code in OS components by unit-tests.

WHO IS BINSIDE TARGET AUDIENCE?

- Companies that need to check thoroughly the used third-party software, including situations when there is no access to its source code.
- Developers who need to increase dynamic analysis quality with the data collected by static analysis.
- Reverse engineering experts.
- Companies performing software audition or certification.

BINSIDE WORKFLOW



CASR: CRASH ANALYSIS AND SEVERITY REPORTING TOOL

GitHub → https://github.com/ispras/casr

Casr creates automatic reports for crashes happened during program testing or deployment on Linux. The resulting reports contain the crash's severity and additional data that is helpful for pinpointing the error cause. Casr is open source.

FEATURES AND ADVANTAGES

Casr could collect crash reports using several approaches (coredump, GDB, Asan, Ubsan) and process exceptions thrown in various languages (Rust, Go, Java, Python, JavaScript, C#). Casr can be used to automate analysis of fuzzing results and to submit them to vulnerability management systems.

Casr provides:

- Detecting critical program faults that can lead to hijacking control flow.
- Classifying crashes based on a program state at a crash time (function return address corruption, null pointer dereference etc.). Fatal errors are further grouped based on severity, such as exploitable, potentially exploitable, or denial of service errors.
- An extended crash report containing the fatal error's severity and other data (OS and package versions, executed command line, call stack, open files and network connections, register state etc.).
- Deduplicating and clustering crashes based on their call stack. The detected clusters would likely contain similar reports that describe the same bug. Adding newly found crashes to existing clusters is also supported.
- Integration with modern fuzzers such as Sydr, AFL++, LibFuzzer (including go-fuzz, Atheris, Jazzer, Jazzer.js, C#).
- The libcasr library for developing custom analyzers.
- Submitting results to DefectDojo, a vulnerability management system, which allows convenient integration of fuzzing results review into CI/CD.

WHO IS CASR TARGET AUDIENCE?

- Companies that need to receive the data regarding user-deployed programs' crashes to develop high reliability and security software.
- Companies that need to certify the developed software.
- Certification laboratories.

CASR DEPLOYMENT STORIES

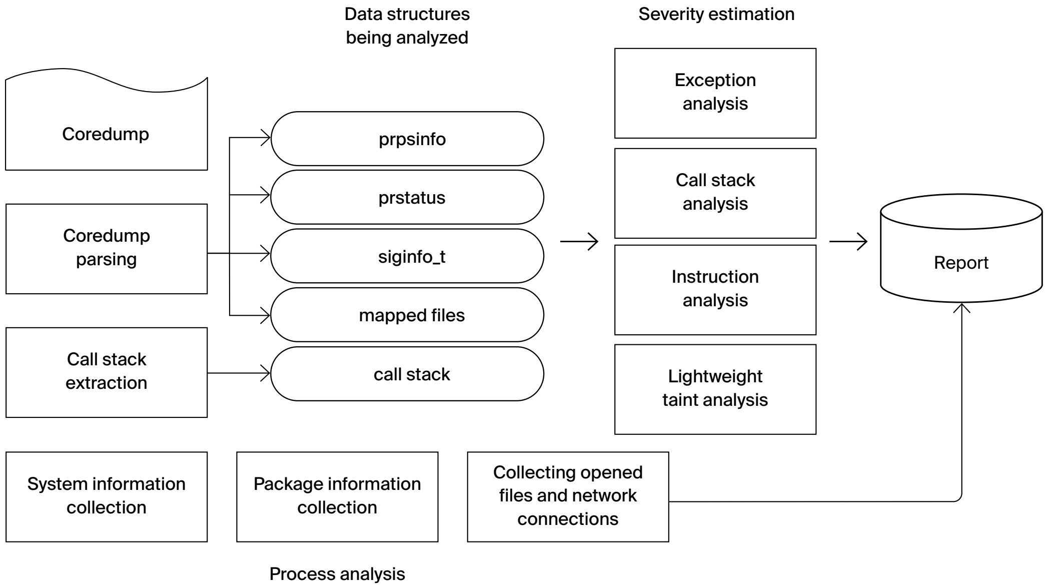
Casr is used for crush analysis in the Sydr tool, also by ISP RAS.

The libcasr library is now integrated with libAFL.

SYSTEM REQUIREMENTS

Linux-based OS for x86 (32/64), aarch64, RISC-V 64.

WORKFLOW



CATALOGUE OF TECHNOLOGIES

NATCH: A TOOL FOR DETERMINING ATTACK SURFACE

Natch is a tool for determining attack surface based on QEMU full-system emulation. Natch utilizes dynamic taint analysis, virtual machine introspection and deterministic replay. Natch is included in the Unified Register of Russian Programs (№13673).

FEATURES AND ADVANTAGES

Natch is aimed for attack surface detection, which is determining executables, dynamic libraries and functions that participate in processing input data (files, network packets) when performing a task. Gathered data is visualized in the SNatch graphical interface that is included in the distribution.

Natch is based on QEMU, a full-system emulator, which allows analyzing all software comprising a system, including the OS kernel and drivers. The important advantage of Natch is unifying key features of competitors in a single tool.

The attack surface detection task can be integrated into CI/CD for organizing system and integration testing, which makes applying fuzzing and functional testing within security development lifecycle more effective.

Natch provides:

- determining attack surface, i.e. processes, functions, and modules that were engaged in processing tainted data when executing a test scenario;
- detecting open files, sockets, and ports, as well as data flows coming through them;
- analyzing programs written on C/C++, Go, Python, and Java;
- automatic downloading of debug information for the kernel and system modules;
- extracting debug information from DWARF data;
- building a graph that shows tainted data flow through all the system crossing process and module boundaries;
- collecting network packet log in the PCAP format;
- calculating source and binary code coverage;
- collecting data corpus for further fuzzing of selected functions (for simple type arguments).

NATCH WORKFLOW:

- Natch starts with recording a scenario of analyst working session in a virtual machine;
- The analyst marks as interesting certain network traffic or file accesses;
- Natch replays the recorded work scenario and tracks tainted data flows, collects logs with tainted data operations and system events;
- The analyst loads the created log into the SNatch interface for further analysis.

FEATURES OF THE SNATCH GRAPHICAL INTERFACE:

- Graph of processes that were working with tainted data. It allows time tracking tainted data and convenient ordering for scheme elements.
- Time diagram for OS processes.
- Call stacks for tainted functions grouped by process.
- Call stacks for script functions in case they are present in the analyzed program.
- Process flame diagram with color coding for tainted and untainted functions.
- Examining process tree for processes that were executing with filtering just tainted processes if needed.
- Examining resources used by processes.
- Examining read/write accesses for files and sockets.
- Highlighting privileged processes (e.g. running from root) that were working with tainted data.
- Generating function annotations for the Futag tool.
- Forwarding filtered network traffic to Wireshark.
- Convenient search in graphs and keeping view history.
- Generating reports with important data in the PDF format.
- API for automated attack surface testing.

WHO IS NATCH TARGET AUDIENCE?

- Russian companies developing secure software.
- Certification labs and regulation authorities.

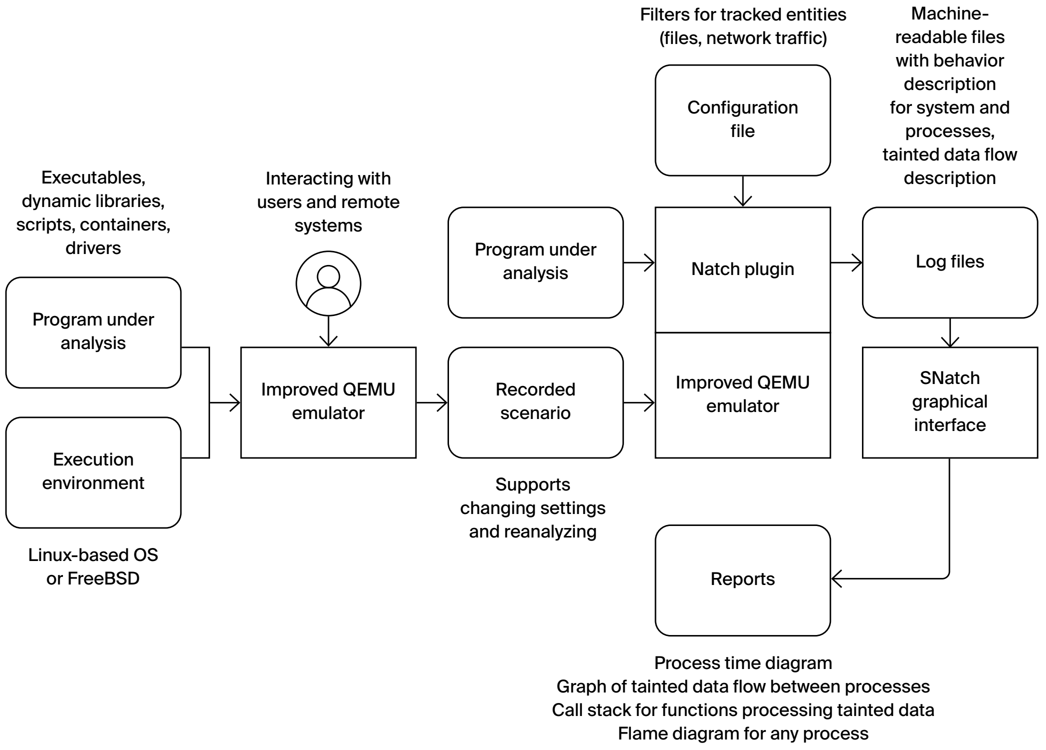
SUPPORTED PLATFORMS AND ARCHITECTURES

- Natch system requirements: OC Linux x86-64, 16+GB memory, 200+GB disk space.
- Target architectures: x86-64.
- Target OS: Linux (all versions), Windows 7-11, FreeBSD (latest versions).

NATCH DEPLOYMENT STORIES

Natch is deployed in the Aquarius IT company and a number of certification laboratories.

NATCH WORKFLOW


Filters for tracked entities (files, network traffic)
Machine-readable files with behavior description for system and processes, tainted data flow description
Configuration file
Executables, dynamic libraries, scripts, containers, drivers
Interacting with users and remote systems
Program under analysis
Natch plugin
Log files
Program under analysis
Improved QEMU emulator
Recorded scenario
Improved QEMU emulator
SNatch graphical interface
Execution environment
Supports changing settings and reanalyzing
Linux-based OS or FreeBSD
Reports
Process time diagram
Graph of tainted data flow between processes
Call stack for functions processing tainted data
Flame diagram for any process

SYDR-FUZZ: HYBRID FUZZING AND DYNAMIC ANALYSIS

GitHub →
https://github.com/ispras/oss-sydr-fuzz

Sydr is a dynamic symbolic execution tool for binary programs, which is used for automatic test generation tool to find errors and increase code coverage during testing. Sydr constructs the program's mathematical model that allows a fuzzer to explore new execution paths that are hard to discover via classic mutation approaches. Sydr improves dynamic symbolic execution methods proposed in earlier Avalanche and Anxiety analyzers developed in ISP RAS.

Sydr-fuzz is a dynamic analysis tool for security development lifecycle that implements hybrid fuzzing with Sydr and modern fuzzers (libFuzzer and AFL++) and automates code coverage, corpus minimization and crash analysis phases. The project site OSS-Sydr-Fuzz (https://github.com/ispras/oss-sydr-fuzz).

FEATURES AND ADVANTAGES

In contrast with similar open source tools, Sydr ensures the correctness of generated input data by checking whether it actually inverts the target branch. Sydr-fuzz provides a convenient way for automating dynamic analysis pipeline:

– Fuzzing and hybrid fuzzing: sydr-fuzz run;
– Corpus minimization: sydr-fuzz cmin;
– Error detection (out of bounds, integer overflow, division by zero, etc.) via security predicates: sydr-fuzz security;
– Collecting coverage: sydr-fuzz cov-report;
– Report generation, crash deduplication and clustering: sydr-fuzz casr.

Sydr-fuzz provides:

– Hybrid fuzzing with Sydr and libFuzzer/AFL++. Hybrid fuzzing supports programs written on C/C++, Rust, and Go.
– Automating fuzzing and complete dynamic analysis pipeline for Python/Java/JavaScript/C# using Atheris/Jazzer/Jazzer.jz/SharpFuzz.
– World-level efficiency: continuous benchmarking (https://sydr-fuzz.github.io/fuzzbench).
– Repository with ready to fuzz projects: 84 projects (400+ unique fuzz targets) in OSS-Sydr-Fuzz (https://github.com/ispras/oss-sydr-fuzz).
– Trophies: Sydr-fuzz found 172 new bugs in 31 open source projects (https://github.com/ispras/oss-sydr-fuzz/blob/master/TROPHIES.md); 30 errors are found via safety predicates.
– Effective concrete/symbolic execution (concolic execution): parallel inverting of conditional branches with caching support, path predicate slicing, optimistic solution heuristic, high-level standard functions modeling, and using Bitwuzla, a high-performance

SMT solver, significantly speed up analysis. Analysis of indirect dependencies is implemented, including indirect branches, jump tables, and computed gotos / symbolic pointers.

- Easy-to-use dynamic analysis automatization: unifying launch and results of different fuzzers, easy fuzzing campaign setup via a config file, parallel fuzzing support, gathering statistics, supporting parallel launch of Sydr, libFuzzer, and AFL++.
- Safety predicates. Finding errors via symbolic execution, i.e. generating input data to reproduce them. Division by zero, null pointer dereference, buffer overflow, integer overflow and underflow, format string, command injection are supported.

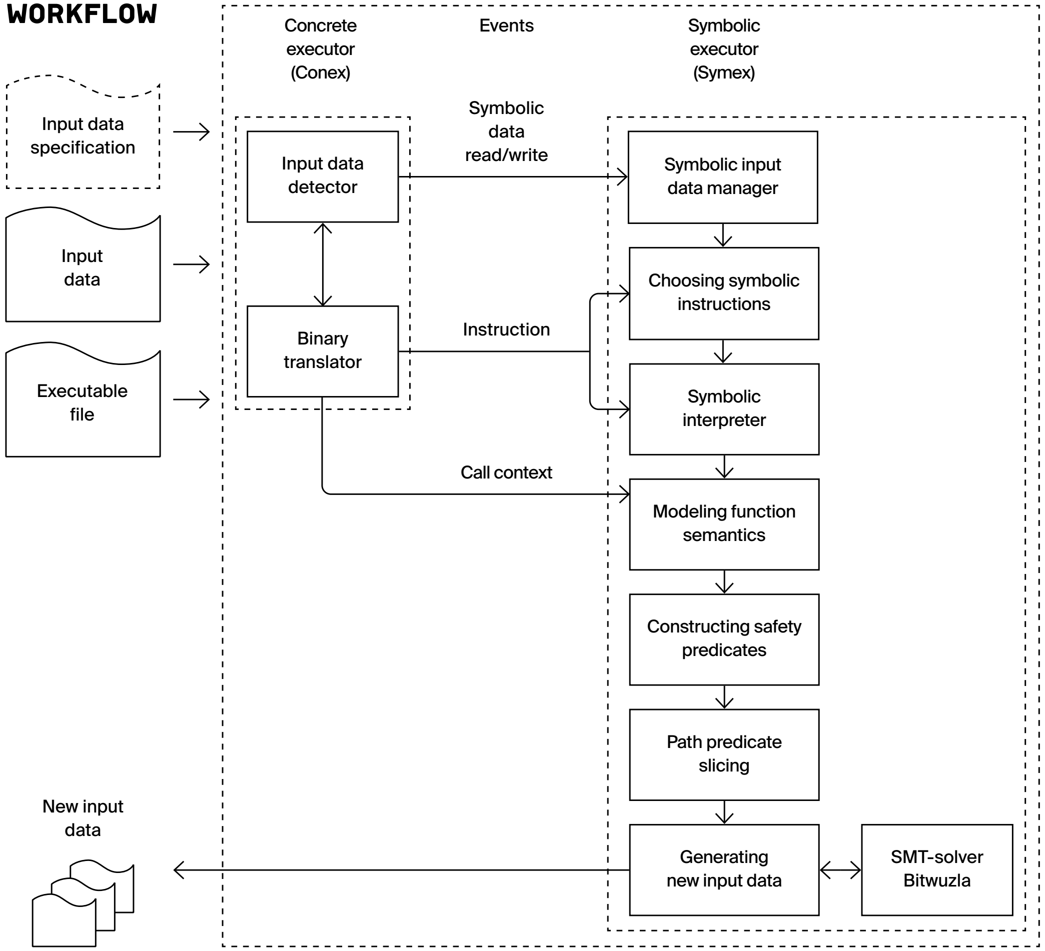
SYSTEM REQUIREMENTS

Sydr runs on x86-64, aarch64, and RISC-V 64 platforms. Sydr supports 64-bit Linux, including Ubuntu 18.04/20.04/22.04, Astra Linux 1.7, ALT Workstation 10 and similar.

SYDR DEPLOYMENT STORIES

Sydr and Sydr-fuzz are the parts of ISP Crusher system that is used in more than 70 companies and certification labs, including RusBITech, Postgres Professional, Security Code, Swemel, and others. Sydr + Sydr-fuzz is the main dynamic analysis tool in Research Center for Trusted Artificial Intelligence at ISP RAS.

WORKFLOW



PROTOSPHERE: NETWORK TRAFFIC ANALYZER

Protosphere is a system of deep packet inspection (DPI). It can serve as a part of intrusion and information leak protection systems. Protosphere detects inconsistencies between a protocol specification and the actual traffic. It allows to add support quickly for new protocols (either open or closed) due to the flexibility of its internal representation.

FEATURES AND ADVANTAGES

Protosphere is an innovative system based on the innovative research in the area of network traffic analysis. It combines the key features of similar tools (e.g. Wireshark, Microsoft Message Analyzer, nDPI) with a universal data representation model that enables rapid expansion of analysis capabilities.

Protosphere provides:

- Advanced system core:
  - universal data representation model used when parsing network traffic;
  - processing of corrupted, reordered or duplicated packets, as well as handling of packet loss and processing of asymmetric traffic;
  - compressed/encrypted data analysis;
  - support for tunnels of arbitrary configuration;
  - support for network flows causality.
- Support for all stages of network trace analysis (each stage has a visualization component that are synchronized between stages):
  - network connections localization in the network interaction graph and the network flow tree;
  - detailed view of the selected connections in the timeline diagram;
  - interactive visualization of the parsed network packets in the stream tree;
  - detection of discrepancies between a protocol implementation and the actual traffic in the diagnostic log.
- Extensive list of supported protocols:
  - DNS, RHCP, RIP;
  - TLS, Microsoft RPC, PostfreSQL;
  - FTP, HTTP, IMAP, SMTP, POP3, BitTorrent;
  - Kerberos, NTLM;
  - GRE, IpSec, PPP, OpenVPN, Wireguard.
- Easy support for new protocols:
  - access to parsing results via API;
  - localize parsing errors;
  - declarative description of network protocols.

- Support for both online and offline analysis modes.
- Support for working in DPI as a Service mode.
- Advanced GUI provides choice of the most convenient way to present the analysis results.
- Numerous options for extending supported features:
  - supporting new programming interfaces;
  - developing various mechanisms for processing results of parsing;
  - adding features to the parsing kernel.
- Adjustment to network bandwidth and available computational resources: flexible configuration allows finding a balance between analysis accuracy and consumed resources.

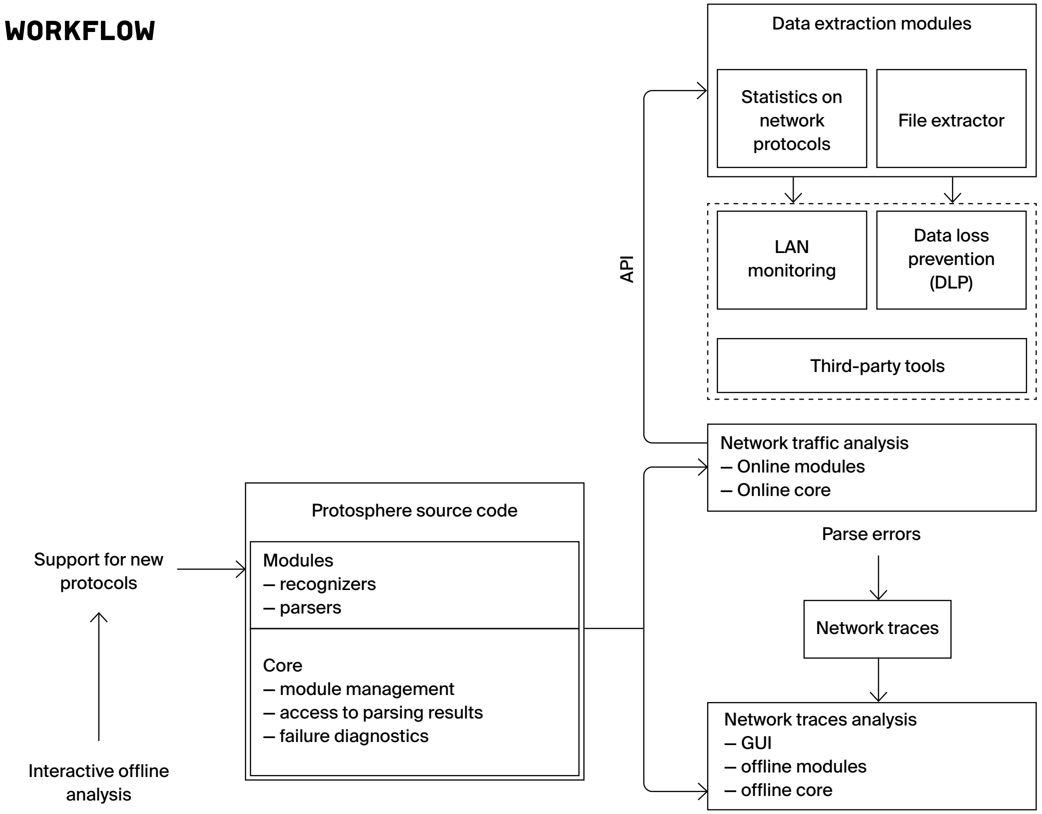
WHO IS PROTOSPHERE TARGET AUDIENCE?

- Companies that are testing network protocol implementations including those in embedded OS and network hardware.
- Developers of network security tools, such as firewalls and IDS/IPS including zero day attack protection.
- Manufacturers of network hardware that must be certified.
- Companies requiring real-time control and monitoring of network channels.

SUPPORTED PLATFORMS AND ARCHITECTURES

Architecture: Intel x86-64, ARM64.
Platforms: Windows OS, Linux-based OSes, Apple macOS.

WORKFLOW



ML IDS: A MACHINE LEARNING BASED NETWORK INTRUSION DETECTION SYSTEM

ML IDS is a network intrusion detection system (IDS) bases on state-of-the-art research for applying machine learning for attack detection. The ML models used are capable of generalization, which allows detecting previously unknown attacks (so-called zero-day attacks).

FEATURES AND ADVANTAGES

ML IDS detects a large number of web attacks on a network level via analyzing session features. The technology allows using either open (HTTP) or encrypted (HTTPS) transmission protocol. Inspecting HTTP/2 and HTTP/3 protocols is supported via a reverse proxy server within the protected network (e.g. the nghttpx HTTP router).

Testing and comparing ML IDS against signature-based open source intrusion detection and prevention systems showed that ML IDS:

- is superior to Suricata, a signature-based network IDS (NIDS), with respect to detection quality when working with the encrypted HTTPS traffic;
- is similar to ModSecurity, a signature-based web application firewall (WAF), with respect to quality in modelled environment.

ML IDS allows using signature-based information protection systems as additional sources for training data to increase the model generalization and attack detection quality.

ML IDS provides:

- Detecting attacks in almost real time.
- Detecting previously unknown attacks.
- Detecting attacks without packet contents inspection, which provides the ability to support encrypted traffic.
- Adapting to changing environment (allows fine-tuning and selecting an optimal model).
- Modular architecture that allows increasing performance.
- Automatic deployment of the infrastructure for generating and saving a training dataset.
- A convenient web interface for working with the system's output that can be easily configured for displaying necessary data.

CATALOGUE OF TECHNOLOGIES

Currently ML IDS is trained for detecting XSS, SQL Injection, Command Injection, Brute Force, Web Shell, and DoS web attacks. The developed ML model is ranked second in the “Network Intrusion Detection on CICIDS2017” independent benchmark on paperswithcode.com (as of October 24th, 2024).

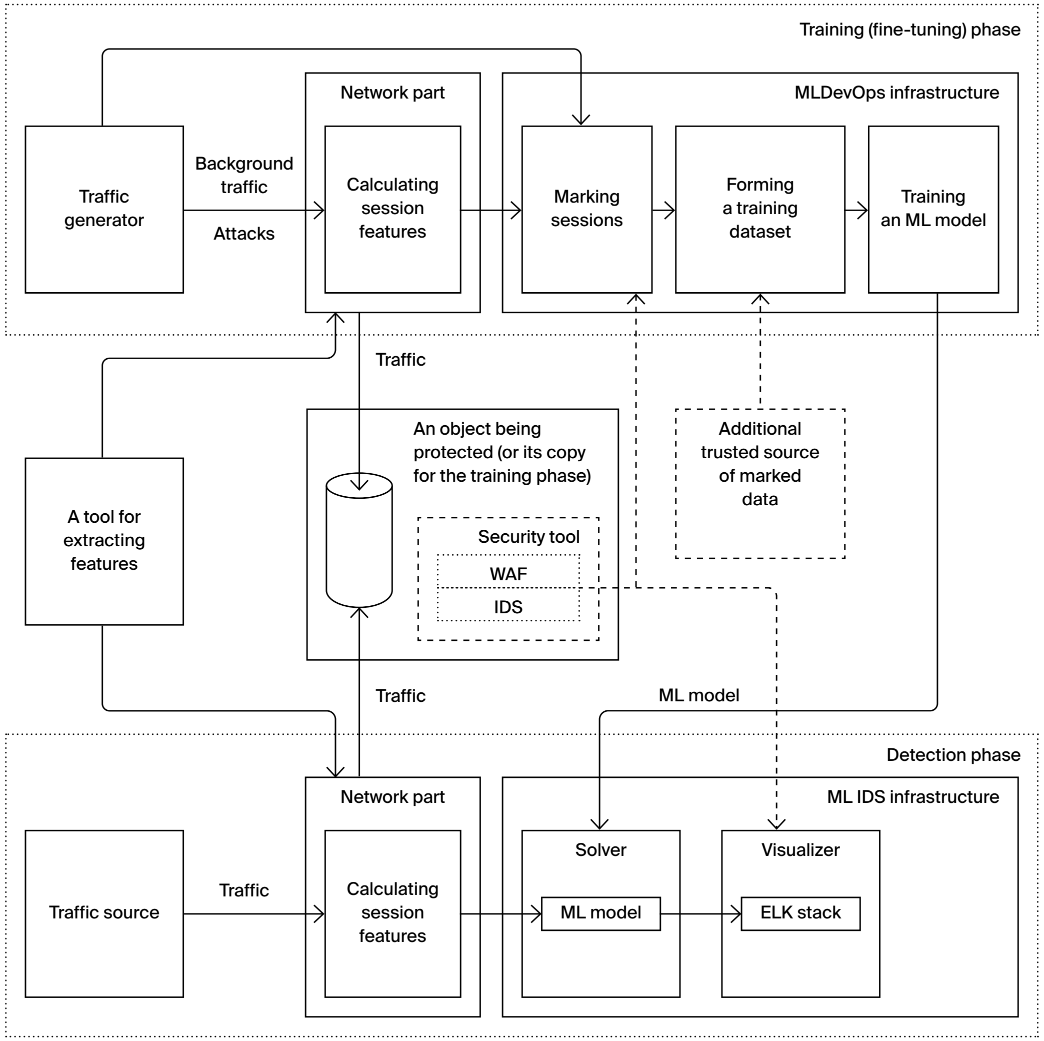
SUPPORTED PLATFORMS AND ARCHITECTURES

- Platforms: Linux-based operating systems.
- Architectures: Intel x86-64.

ML IDS TARGET AUDIENCE

- Companies that are in need of protecting their network resources from attacks.
- Companies developing network security solutions (firewalls, intrusion detection and prevention systems).

ML IDS WORKFLOW



REQUALITY: REQUIREMENT MANAGEMENT TOOL

Requality is an extensible tool for requirements management (mainly in software system development). It allows to develop software requirements from scratch as well as to create requirement catalogues by marking up existing documents and preserving the links between the requirements and the document fragments. It supports hierarchical structure of requirements, provides traceability between requirements of different levels and possibility of collaborative work on requirements using the GIT version control system.

FEATURES AND ADVANTAGES

Requality stands out by featuring the possibility of requirement catalogue creation from the markup of existing documents. Each of the created requirements preserves the link to one or several source document fragments.

The other functionality is close to existing commercial counterparts (IBM DOORS, Jama, Polarion) and surpasses some of the existing open-source products (aNimble, ProR, RM-TOO). The tool and user manual are available at the project website: https://requality.ru.

Requality provides:

- Structuring and storing a requirements catalog:
  - A requirements catalog is a structured set of linked requirements and other elements stored within a single workspace. The top level elements are projects in which individual sets of requirements are stored. This capability is used, among other things, to separate upper level requirements from the lower level requirements developed on their basis.
  - The catalogue elements include the requirements themselves as well as various other nodes. The tool supports a basic set of elements, including:
    - requirements containing descriptions of features and limitations of the object being developed;
    - text nodes that are not requirements themselves, but provide context for dealing with requirements (e.g. term definitions or notes);
    - documentary representations of the requirements on the basis of which the catalogue was developed;
    - report settings and the results of their generation;
    - comments.

The set of catalogue elements can be increased by developing extensions.
  - Node identification is supported in several ways, including the use of a unique numeric identifier within the project and a composite human readable hierarchical path;
  - Node properties include both those provided by default tools (node description, short string identifier, and others) and user-defined parameters used to indicate element characteristics;
  - using HTML markup in the text of requirements and in other features makes it possible to use different ways of formatting text and to provide supporting resources, such as images and tables.
- Link management, traceability and coverage analysis:
  - Creating and naming links between catalogue elements. Link names allow defining references of different types;
  - Building links automatically from terminology via enumerating terms used in a requirement and enabling the corresponding attribute for the node defining the term;
  - Making a link between a text fragment and a requirement allows, on the one hand, to determine the origin of an individual requirement, and on the other hand, it makes it possible to automatically transfer such links to new versions of documents;
  - Comprehensive link traceability is the ability to trace both the original requirements and the requirements developed on their basis for an individual requirement, as well as to examine the context of the catalog element within which it is to be considered;
  - Coverage is a collection of data showing degree of implementation or testing completeness for a catalogue of requirements. Coverage is estimated by links between catalogue items and external elements or between internal catalogue items. The tool supports the use of external coverage information in the form of a specific file format, and provides an extensible set of coverage data sources.
- Change management and collaboration support:
  - GIT is supported as the primary system for collaborative work on the requirements catalogue. A simplified set of commands for submitting changes and updating the local version of the project is available in the interface of the tool;
  - The interface of the tool makes it possible to view the versions of a single node as well as those of the requirements catalogue as a whole; it is also possible to compare individual versions;
  - Comparing different project versions and switching to previous versions is also supported.
- Report generation, in particular:
  - Creating various formats of the requirements catalogue, including those that provide using it for offline work (outside the scope of the tool), as well as exchanging catalogue data with other tools or solving non-standard tasks within the development process;
  - Providing traceability data to view information on the relationships between catalog elements;
  - Comparing catalogue versions to manage work progress on requirements via studying the differences in the structure and properties of the project requirements for the selected versions of the catalogue;
  - Coverage analysis to examine the status of individual catalog elements in terms of coverage information obtained from a selected source;
  - Support for user-defined templates using available information on the catalog, its versions, and coverage information.
- A programming interface (API) with the ability to modify stored data and create new projects is supported. It can also be used to exchange data with third-party tools.
- It is possible to develop extensions to define new elements, sources of information on coverage, or to get new functionality.

SYSTEM REQUIREMENTS

Windows OS or GNU/Linux based OS, Java Runtime.

REQUALITY DEPLOYMENT STORIES

Requality has been in development since 2011. It has been used to develop and manage requirements' changes in a project to develop a real-time operating system in compliance with KT-178C processes, as well as to catalog requirements from various standards (including TTCN and POSIX) in order to perform subsequent conformance testing of compatible products.

2 DATA ANALYSIS

INFRASTRUCTURE PROJECTS AND AI DEVELOPMENT TOOLS

NATURAL LANGUAGE PROCESSING

DOCUMENT PROCESSING

APPLICATIONS

ASPERITAS AND CLOUD SOLUTIONS FAMILY

Asperitas is a platform for data storage and performing complex compute-intensive tasks in scientific, educational, and commercial projects. It includes a cloud environment also called Asperitas, as well as Michman, a PaaS orchestrator, and Clouni, a multi-cloud IaaS orchestrator based on TOSCA standard. Fanlight, a web laboratories platform, and Cotea, a system tool intended for programmatic control of Ansible scripts execution, are also a part of ISP RAS cloud solutions family.

ASPERITAS CLOUD ENVIRONMENT

Asperitas cloud environment is based on Openstack and Ceph, which are the modern standard of large private cloud systems. The distribution delivery is provided as a ready-made solution with everything necessary for deployment, including a TUI installer.

Other advantages of Asperitas:

- An onsite installation option (the provided infrastructure can be installed and fully controlled in an isolated environment due to the usage of open standards and software as well as ISP RAS research).
- High security: the environment is built on top of a smaller code base and uses its own know-how solutions that increase security.
- Standard interfaces of virtual and computational clusters management using Keystone, Neutron and Nova systems.
- Block storage and scalable object storage is based on the Ceph distributed file system.
- Adaptation to specific problem classes (e.g. continuum mechanics, big data analysis, program analysis for defect detection etc.).

Asperitas cloud environment is included in the Unified Register of Russian software (No. 5921).

CLOUNI, A MULTI-CLOUD ORCHESTRATOR

GitHub → https://github.com/ispras/clouni

To enhance the capabilities of infrastructure resource management, ISP RAS is developing the Clouni tool, which allows deploying clusters of virtual infrastructure according to TOSCA Simple Profile normative templates using the Ansible configuration management tool.

Main characteristics of Clouni orchestrator include:

- Own approach for translating TOSCA declarative templates to Ansible scripts, which allows users to avoid the need of describing how infrastructure should be deployed;
- No dependency on the cloud platform being used. Currently, Clouni supports Openstack, Amazon AWS, and partially Kubernetes.
- Fine-tuning of virtual machines, security groups, ports and networks.

The TOMMANO framework, a tool for managing network services in arbitrary clouds, is being developed in a close co-operation with Clouni. The main characteristics of TOMMANO are as follows:

- Automatic deployment of virtualized network functions based on their TOSCA declarative descriptions according to the ETSI MANO standard;
- A number of network function templates is already provided for Firewall, NAT, DPI, DNS, DHCP, and traffic analyzers.
- Service function chaining support based on software defined network managed by the OpenDayLight controller. This allows managing complex network services that have different traffic types processed via different network functions.
- Support for network function deployment: in standalone mode with routing configuration from TOSCA template parameters, in service chains with automatic routing between nodes.

MICHMAN, A UNIVERSAL ORCHESTRATOR

GitHub →
https://github.com/ispras/michman

Michman is a PaaS services orchestration tool for a cloud environment performing big data analysis, machine learning, load management tools, and other tasks. It supports automatic cluster deployment in cloud environment, taking into account user requirements and parameters. It also provides an interface for deploying sets of services from predefined service templates and managing their lifecycle, including:

- A big data analysis cluster with arbitrary number of nodes having Apache Spark and Apache Hadoop fully set up for cooperative work;
- A database of various types, from classic relational to distributed analytical DBs;
- File storage and exchange systems, e.g. MiniO, NextCloud, NFS, GlusterFS;
- Slurm, a cluster management and job scheduling system with the option of GPU usage;
- Kubernetes, a flexible container orchestration system, and tools running on top of it;
- Tools for developing machine learning models, including Jupyter, MLflow, and Ray.

The key advantage of Michman is its flexibility and easy extension of supported services due to using TOSCA language and supporting the following mechanisms:

- Substitution Mapping, which allows describing resources of the same type uniformly. For example, one can describe how the deployment on various resources should be performed (in public or private clouds, on dedicated servers or within containers). Also one can describe the way of integrating an application with various DBs, connecting various file systems, etc.
- Select, which allows reusing resources created previously or utilizing external resources (like external repository or common network file system).

Also Michman allows saving the state of all components of a cloud application consistently, scaling nodes, managing cloud application components separately, updating running services.

FANLIGHT

Fanlight is a platform for providing virtual desktops (DaaS, Desktop as a Service). It allows deploying SaaS infrastructure for computing web-laboratories. It was created as a result of ISP RAS participation in the University Cluster program and in the international Open Cirrus project (founded by HP, Intel, and Yahoo). Fanlight is based on container technologies, unlike most solutions of this class that are based on virtual machines. Initially, the platform had been based on the Docker Compose technology. Later on, a Kubernetes-based implementation appeared. It only supports applications developed for a Linux kernel-based OS. Fanlight is included in the Unified Registry of Russian Software (No. 6066).

Other advantages of Fanlight:

- High efficiency of work with cloud calculations due to the use of containers:
  - comfortable work with heavy engineering CAD-CAE applications requiring 3D graphics hardware acceleration support for complex visualization;
  - support for running MPI, OpenMP, CUDA applications through access to HPC clusters, multicore processors, and NVIDIA graphics accelerators.
- Extended computing capabilities at the PaaS level through connecting hardware resources (HPC/BigData clusters, storage systems, graphic accelerator servers).
- Possibility of customization for a given application area due to integration of specialized calculation application packages and the easy way to add them. In particular, the following have been implemented:
  - in the field of MSS: OpenFOAM, SALOME, Paraview, etc;
  - in the field of Gas&Oil: tNavigator, Eclipse, Roxar, Tempest, etc.
- Operation via any thin client (including mobile devices) without any auxiliary software.
- Deployment on a server, computing farm, cloud (from the IaaS layer), in a Kubernetes cluster, or in dedicated cloud data center. The Kubernetes-based version also provides the opportunity to use different CRI container execution engines.

COTEA

GitHub →
https://github.com/ispras/cotea

Cotea is a tool that makes it possible to run Ansible programmatically and control its execution (Ansible is one of the most popular software deployment systems). Cotea allows to:

- use software control of running Ansible by iterating over the component parts of the Ansible script;
- embed Ansible into other systems;
- debug Ansible runs, including interactive mode; switching to interactive mode occurs in case of a task (part of the Ansible script) execution failure. Examples of functions provided in interactive mode:
  - restart a task that resulted in an error;
  - continue executing the Ansible script without the failed task;
  - add a new Ansible variable during runtime;
  - add a new Ansible task during runtime.

Interactive mode makes it possible to refrain from executing a script all over again in case of errors, which is especially important when working with large scripts.

Cotea is currently used in the deployment of the Asperitas platform. A part of Corea included in Michman and Clouni is called grpc-corea. Grpc-cotea enables these orchestrators to control the deployment process of cloud applications.

CLOUD SOLUTIONS DEPLOYMENT STORIES

The computing cluster based on Asperitas supports a number of ISP RAS technologies (e.g. analyzing Android OS using Svace). The following projects were also implemented: a joint project with Huawei (large graphs analysis using big data processing), and the Tizen OS lifecycle support infrastructure that allows organizing joint development of OS components and automating regular build and testing of OS images. In addition, a number of projects are performed jointly with the Ministry of Education and Science of Russian Federation. Asperitas serves as a foundation for the cloud platform of the National Center for Medical Research "Digital Biodesign and Personalized Healthcare."

In November 2023, ISP RAS and System Solutions JSC signed an agreement, which initiated development of ACloud, an Asperitas-based proprietary cloud solution designed for business, science, and education. The goal is to create the most secure and safe solution together with a full-blown infrastructure for support, integration, and product training; this allows to break dependence on foreign solutions used within critical infrastructure. The ACloud platform was released in November 2024.

The Fanlight platform was used in a number of joint projects for web laboratory deployment, including Russian Federal Nuclear Center of the All-Russian Scientific Research Institute of Experimental Physics, OOO RRS-Baltika, Keldysh Institute of Applied Mathematics (developing a technology for increasing and using efficiently the hydrocarbon raw materials resource potential of the Union State of Russia and Belarus), ISP RAS Laboratory of Continuum Mechanics (https://unicfd.ru).

TALISMAN: A PLATFORM FOR CONSTRUCTING INTELLECTUAL ANALYTICAL SYSTEMS

Talisman is a unified set of tools that automate typical data processing tasks, such as data retrieval, integration, analysis, storage and visualization. It ensures the fast development of specialized multi-user intellectual analytical systems that merge and work uniformly with the data from private databases and Internet sources (including social networks).

FEATURES AND ADVANTAGES

Talisman unifies the tools necessary for big data and cutting-edge AI tools, using them to extract information from random sources. It makes it possible to quickly create intelligent analytical systems using low-code and no-code approaches. It is constantly learning from the results of analyst work without the need for additional labor.

Talisman provides:

- A rich set of reusable components that have APIs for easy management and integration:
  - Data retrieval components. They include a framework for Internet data collection, namely, from social media (Facebook, VKontakte, Twitter, Instagram, Odnoklassniki, YouTube, LinkedIn etc.), blogs, news, MediaWiki sites, developer portals etc. There is also a system for importing data from file storages and databases.
  - Automatic data analysis components. A set of tools allowing to transform input data of any format into a unified universal representation (in particular, Dedoc, developed by ISP RAS, is used). The documents in this representation are subjected to analysis with the help of machine learning methods. It is possible to add your own handlers as containers with REST API. The processing sequence is managed by the Talisman.Stream system (No. 6045 in the Unified Registry of Russian Software).
  - Storage and indexing components. These include a number of databases and information search engines that store source data, automatic analysis results, and results of manual user work.

- An easy-to-use web interface that unifies all components requiring user interaction.
- A flexible modular architecture that allows adding new features to the interesting components without changing others.
- A scalable architecture that allows processing and storing more data just by adding more hardware without any software change.
- Specialized components that monitor system status, manage event log, perform deployment, authentication and authorization, access control, and unidirectional data transfer.
- Tools and methods for training machine learning models as well as for transferring existing algorithms to other knowledge domains.
- A configurable knowledge domain scheme that can be changed by a user when the system is in operation.
- Complete alienability of the systems under development. Each system can be deployed at the customer's site, either on existing hardware or as part of a hardware-software system.
- Integration with private customer systems via provided APIs managing all components.
- License purity. Talisman is based on open source and know-how ISP RAS tools.

TALISMAN APPLICATION AREAS

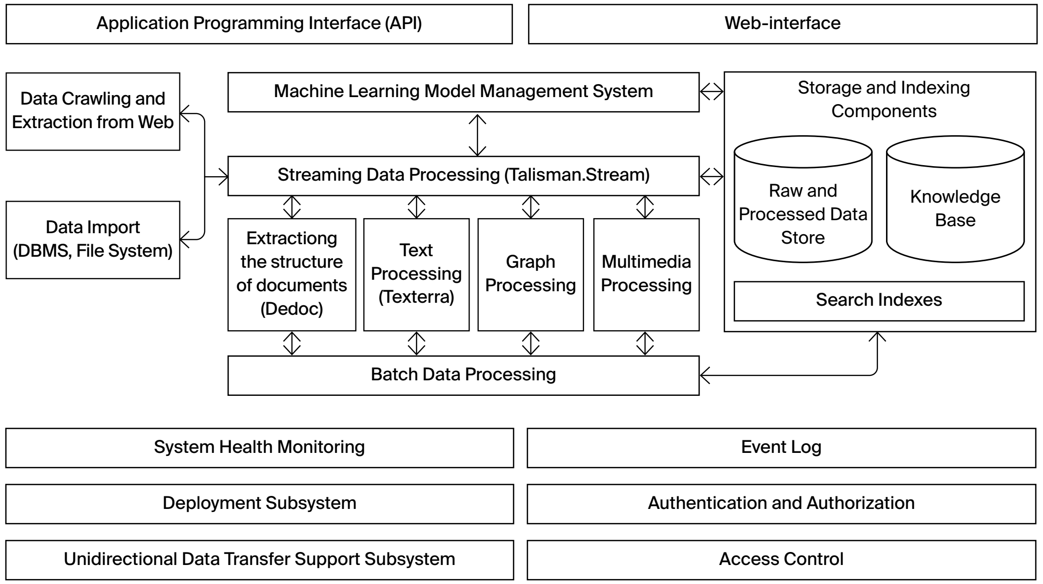
Talisman makes it possible to create analytical systems for a wide range of applications. Application examples:

- Automated knowledge base construction for a given knowledge domain and non-stop monitoring for new information regarding objects of interest (analogous to Palantir Gotham).
- Competitor intelligence based on open sources (OSINT), analogous to Maltego.
- Monitoring media for performing analytics tasks (analogous to LexisNexis).
- Optimization of personnel management: effective selection of employees, verification of questionnaire data, detection of incorrect behavior in the open information space (the Talisman. Biography system, No. 5547 in the Unified Register of Russian Software).
- Identification of information campaigns that manipulate the opinion of the target audience, as well as determining the target audience the campaign is aimed at.
- Identifying and analyzing the specifics of information distribution infrastructure (resources, users, bots), as well as analyzing the typical roles of community members in communication (source, opinion leader, distributor, moderator, bot, commentator).
- Managing business reputation of people and organizations: monitoring relevant messages, identifying problems that cause dissatisfaction, monitoring leaks and internal information disclosure.
- Objective evaluation of performance and testing strategies on target audiences for feedback.
- Management of social tension points; detection and timely prevention of conflict escalation.

SUPPORTED LANGUAGES

Talisman employs advanced neural networks to analyze data. The tools used allow to extract information from more than 100 natural languages.

TALISMAN WORKFLOW


Application Programming Interface (API)
Web-interface
Data Crawling and Extraction from Web
Data Import (DBMS, File System)
Machine Learning Model Management System
Streaming Data Processing (Talisman.Stream)
Extractiong the structure of documents (Dedoc)
Text Processing (Texterra)
Graph Processing
Multimedia Processing
Batch Data Processing
Storage and Indexing Components
Raw and Processed Data Store
Knowledge Base
Search Indexes
System Health Monitoring
Event Log
Deployment Subsystem
Authentication and Authorization
Unidirectional Data Transfer Support Subsystem
Access Control

TRUSTED ARTIFICIAL INTELLIGENCE PLATFORM

The TAI Platform is a cloud platform for developing trusted AI systems. The platform automatizes detecting and preventing threats arising on all phases of the AI models development lifecycle.

FEATURES AND ADVANTAGES

The TAI platform infrastructure is:

- a MLOps platform kernel;
- tools for testing models against attacks, protection and interpretation tools;
- a trusted frameworks repository;
- a development environment for analyzing framework security;
- benchmarks for evaluating models.

The TAI platform provides:

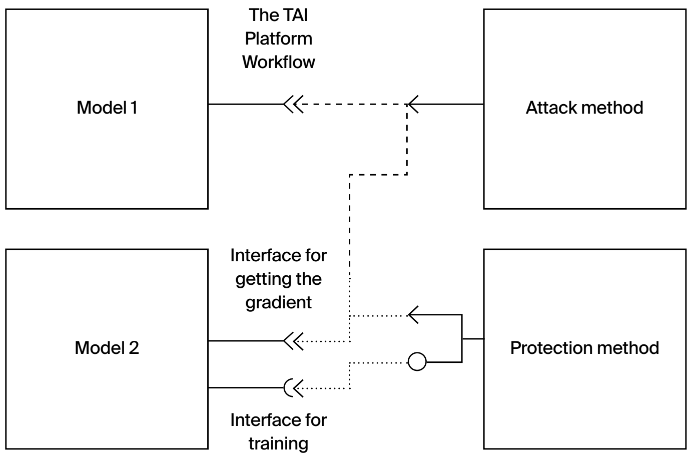
- A set of tools for ensuring trust that is being continuously expanded with published state-of-the-art techniques. The toolset supports:
  - checking models for being susceptible to popular attack types;
  - training models that are robust and interpretable a priori;
  - improving model robustness to popular attack types;
  - interpreting models.
- An API and SDK for embedding trust ensuring processes within existing development tools;
- Alienability. The platform can be deployed on premise either on existing hardware or together with provided hardware as a unified solution;
- A flexible modular architecture that allows to add tools specific to customer's domain;
- Up-to-date trusted versions of popular machine learning frameworks and libraries (PyTorch, TensorFlow etc.);
- An infrastructure backing up a methodology for creating trusted versions of necessary libraries and frameworks;
- Transparent solutions for common problems and challenges arising out of ML model development, including model and data versioning, isolating and organizing experiments, ensuring reproducibility;
- Tools for solving model decay and fighting data drift in real AI-based systems.

WHO IS THE TAI PLATFORM'S TARGET AUDIENCE?

The Platform's tools are useful for:

- ML engineers solving AI tasks in critical application domains;
- Trusted AI ML researchers for performing experiments and quickly deploying state of the art technologies into industrial processes;
- MLOps specialists, for building AI infrastructure and automating development processes of AI solutions;
- Cybersecurity engineers aiming at the highest level of trust, for formalizing processes of ensuring trust in ML-based applications.

THE TAI PLATFORM WORKFLOW



CATALOGUE OF TECHNOLOGIES

COLBA: A SYSTEM FOR DATASET CREATION

Colba is a system for creating annotated datasets for supervised machine learning. It allows flexible and iterative task configuration for annotation experts as well as optimally distributing tasks and track the quality of resulting annotations.

FEATURES AND ADVANTAGES

Colba is an innovation system designed for optimizing the process of creating annotated datasets for complex domain areas, which are problematic to handle via crowdsourcing. Colba totally covers data preparation process, from choosing a variant of task configuration to mass annotation.

Colba provides:

- Automatically distributing annotation tasks to experts:
  - uniformly;
  - taking consent into account;
  - with detecting scientific schools.
- Rich set of methods for calculating agreement between experts:
  - Cohen Kappa and Fleiss' kappa;
  - Krippendorff's alpha;
  - Precision, Recall, F1;
  - IoU.
- Iterative process of task configuration refinement.
- Support for many types of task configuration and input data types:
  - binary classification;
  - multiclass classification;
  - multiclass multilabel classification;
  - NERC;
  - image segmentation.
- Support for focused data selection for high priority annotation.
- Integration with popular annotation systems that experts know well.

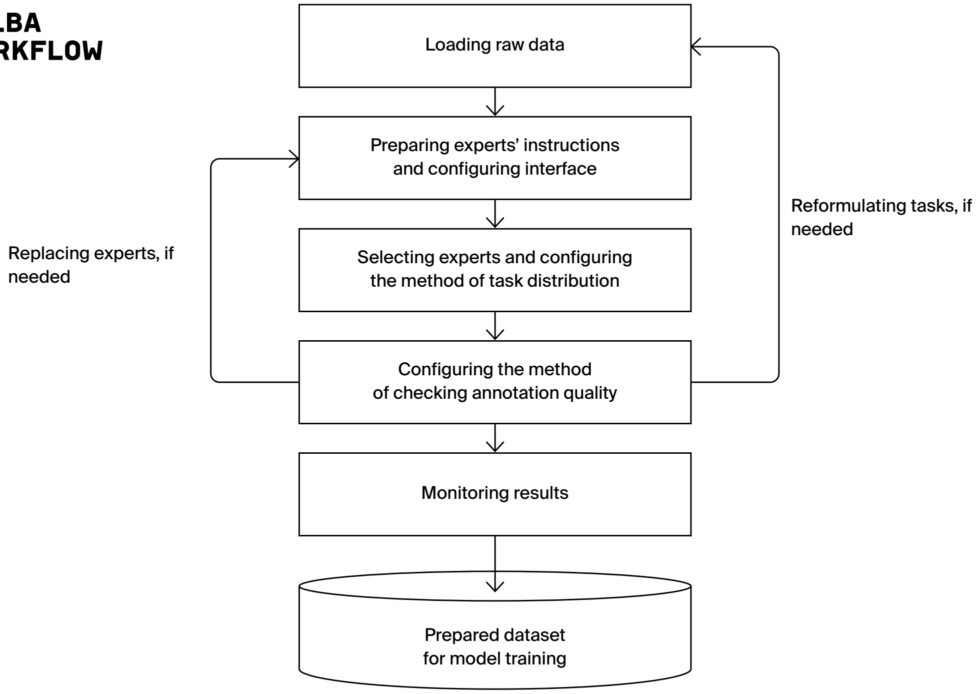
WHO IS COLBA TARGET AUDIENCE?

- Machine learning experts that need data for solving an application domain problem.

COLBA DEPLOYMENT STORIES

Colba is included in the WCRC "Digital Biodesign and personalized health care" cloud platform that is being developed at ISP RAS.

COLBA WORKFLOW


Loading raw data
Preparing experts' instructions and configuring interface
Selecting experts and configuring the method of task distribution
Configuring the method of checking annotation quality
Monitoring results
Prepared dataset for model training
Replacing experts, if needed
Reformulating tasks, if needed

LINGVODOC: VIRTUAL LAB FOR DOCUMENTING ENDANGERED LANGUAGES

GitHub → https://github.com/ispras/lingvodoc

Lingvodoc is a system intended for collaborative multi-user documentation of endangered languages, for creating multi-layered dictionaries and performing scientific work with the received sound and text data. This is a joint project with the Institute of Linguistics of the Russian Academy of Sciences and Tomsk State University. Lingvodoc is under active development since 2012 and can be found on lingvodoc.ispras.ru.

FEATURES AND ADVANTAGES

Lingvodoc is an open source cross-platform system based on an innovative research (https://github.com/ispras/lingvodoc, https://github.com/ispras/lingvodoc-react).

Lingvodoc provides:

- Collaborative work on adding new information to dictionaries (as opposed to the similar Starling project that does not support this feature).
- Saving full history of user actions.
- Working with audio-textual corpuses and dictionaries simultaneously based on the integration with the ELAN system developed by Max Planck Institute of Psycholinguistics (Netherlands).
- Creating and editing unidirectional and bidirectional connections between lexical entries within dictionaries as well as external connections between dictionaries.
- Recording, playing and storing marked-up sounds (in WAV, MP3 and FLAC formats), as well as constructing vowel formants followed by data visualization.
- Advanced search supporting multiple parameters (as opposed to the similar TypeCraft project).
- Ability to search data on a map with automatic demarcation of isoglosses.
- Conflict-free bilateral delayed synchronization.
- High automation level (compared to the similar Kielipankki project): ability to carry out automatic etymological and phonetic analysis.
- Creating dictionaries of any structure, such as typical two-layer dictionaries with lexical entry layer and paradigms layer or multi-layer dictionaries. Importing existing dictionary structures is also supported.
- Algorithms mimicking the scholars' work on phonetic and etymological analysis.
- Support for refining language and dialect classification and building 2D and 3D diagrams via glottochronology, morphology, etymologic and phonetic features.
- Support for storing text corpora in Word format, and dictionaries in the Excel format.
- An interface for automatical processing of parallel corpora.
- Built-in morphological analysis for the languages of the ethnicities of Russia in the Aperitum format.
- A convenient interface for disambiguating homonyms after completing morphological analysis.
- Either using the ISP RAS cloud infrastructure resources or locally deployed resources with data isolation.
- Desktop and web-based versions.
- Open registration (confirmation required).
- Fast development for extending the system features as well as easy adapting to other fields of knowledge.

A prototype version of a learning platform hosted at edu.ispras.ru is created using Lingvodoc glossed corpora and analysis programs. The platform contains 10,000 exercises on 9 languages and allows:

- creating exercises for languages used in Russia for any age or language proficiency level (a teacher can create own exercises and avoid checking them by himself);
- doing exercises in either own or foreign languages, and the platform will tailor the exercises based on the proficiency level of a pupil.

Artificial technologies are being introduced to Lingvodoc. Training a neural network for detecting cognates within languages of Russian ethnicities is near completion. Testing the network shows better results than those obtained with statistical programs.

WHO IS LINGVODOC TARGET AUDIENCE?

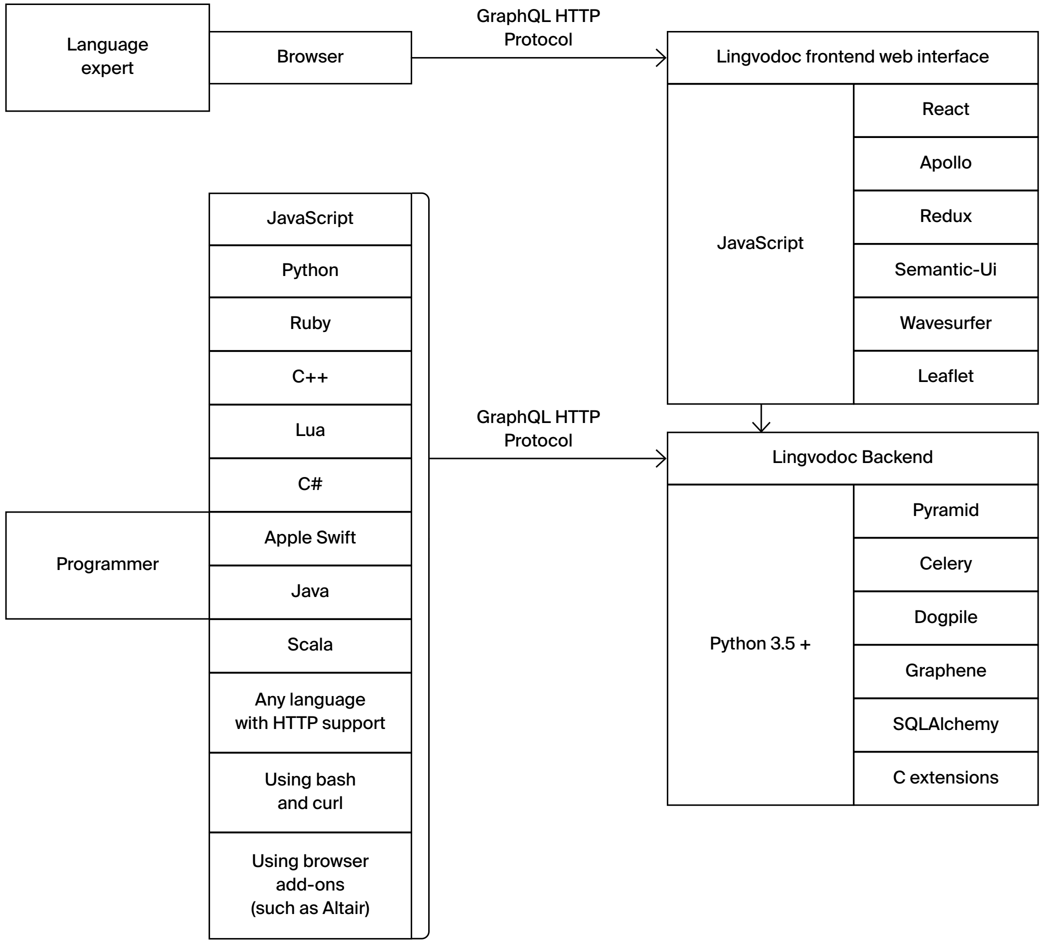
Lingvodoc is designed primarily for linguists performing a research in the area of documenting the endangered languages of Russian ethnicities. However, it is possible to adapt the technology for other purposes.

LINGVODOC DEPLOYMENT STORIES

Lingvodoc is currently used by philologists in 29 universities and scientific centers of 16 cities, including Tomsk State University, Institute of Philology (Siberian Branch of RAS). Institute of history, language and literature (Ufa scientific center of RAS), Udmurt Federal Research Center UB RAS, North-Eastern Federal University, Ugra State University, Institute of Linguistics, Literature and History (Karelian Research Centre of RAS), Murmansk Arctic State University. Specialists using the platform are ready to teach master classes for their colleagues.

In 2023, four groups of researchers from several Russian cities took part in additional training courses on “Using the features of the Lingvodoc platform in the work of linguists” (including at the Bashkir State University and the People’s Friendship University of Russia).

LINGVODOC WORKFLOW



DEDOC: DOCUMENT STRUCTURE RETRIEVAL SYSTEM

Dedoc is an open universal library for converting documents to a unified output format. It extracts a document’s hierarchical structure and content, its tables, text formatting and metadata. The document’s contents are represented as a tree storing headings and lists of any level. Dedoc can be integrated in a document contents and structure analysis system as a separate module.

FEATURES AND ADVANTAGES

Dedoc is implemented in Python and works with semi-structured data formats (DOC/DOCX, ODT, XLS/XLSX, CSV, TXT, JSON) and unstructured data formats like images (PNG, JPG etc.), archives (ZIP, RAR etc.), PDF and HTML formats. Document structure extraction is fully automatic regardless of input data type. Metadata and text formatting is also extracted automatically.

Dedoc provides:

- An open source Python library (https://github.com/ispras/dedoc).
- Extensibility due to a flexible addition of new document formats and to an easy change of an output data format.
- Support for extracting document structure out of nested documents having different formats.
- Extracting various text formatting features (indentation, font type, size, style etc.).
- Working with documents of various origin (statements of work, legal documents, technical reports, scientific papers) allowing flexible tuning for new domains.
- Working with PDF documents containing a text layer:
  - Support to automatically determine the correctness of the text layer in PDF documents;
  - Extract content and formatting from PDF-documents with a text layer using the developed interpreter of the virtual stack machine for printing graphics according to the format specification.
- Extracting table data from DOC/DOCX, PDF, HTML, CSV and image formats:
  - Recognizing a physical structure and a cell text for complex multipage tables having explicit borders with the help of contour analysis.

- Working with scanned black-and-white documents (image formats and PDF without text layer):
  - Using Tesseract, an actively developed OCR engine from Google, together with image preprocessing methods.
  - Utilizing modern machine learning approaches for detecting a document orientation, detecting single/multicolumn document page, detecting bold text and extracting hierarchical structure based on the classification of features extracted from document images.
  - Processing PDF-documents that have incorrect text layer;
  - Processing documents with a background watermark.

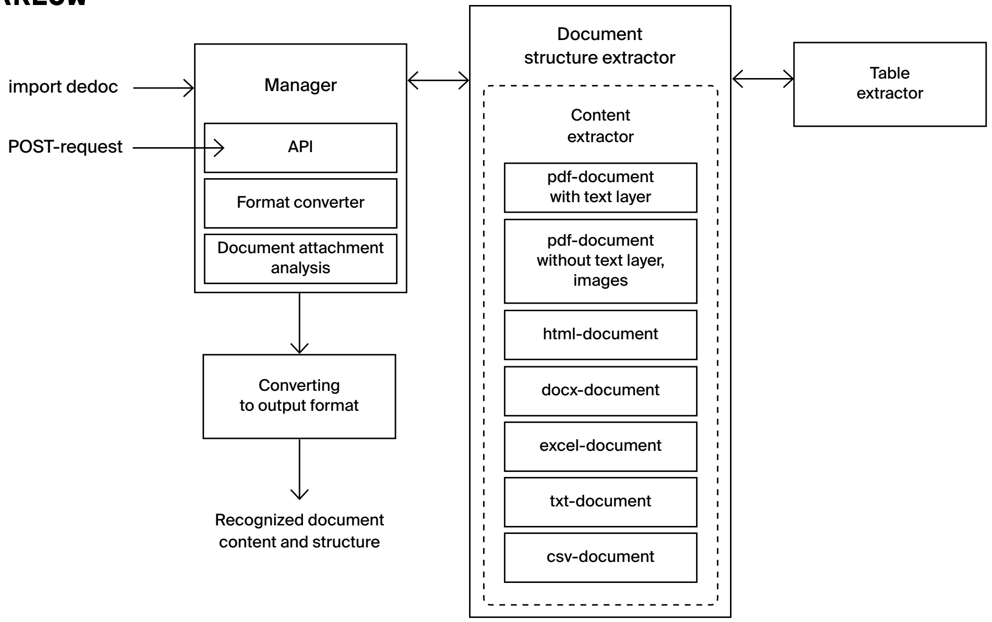
WHO IS DEDOC TARGET AUDIENCE?

- Developers of document content analysis and management systems.
- Developers of intellectual text analysis algorithms.
- Developers of automatic document processing systems.

SUPPORTED LANGUAGES

Russian and English.

WORKLOW


import dedoc
POST-request
Manager
API
Format converter
Document attachment analysis
Converting to output format
Recognized document content and structure
Document structure extractor
Content extractor
pdf-document with text layer
pdf-document without text layer, images
html-document
docx-document
excel-document
txt-document
csv-document
Table extractor

DOCMARKING: PREVENTING ANONYMITY IN DOCUMENT LEAKAGE

DocMarking is a unique system for embedding digital watermarks into text documents. It allows creating a digital or physical document copy that is almost indistinguishable from the original yet exactly identifies the user or the device that was the intended recipient.

FEATURES AND ADVANTAGES

DocMarking is based on research results in the areas of steganography, digital image processing, and machine learning. The marking system builds on the methods for text detection and classification in images and uses statistical features of document images.

DocMarking has a number of advantages compared to competing technologies. Watermark extraction does not require the original document. The system supports repeated embeddings of a watermark in the same scanned document, and the previous watermark is erased when the new one is being embedded.

DocMarking provides:

- Marking algorithms based on machine learning.
- Support for documents of all formats.
- Working with any application.
- Protecting documents either when a document is displayed on a screen or printed.
- Watermark extraction without access to original unmarked documents.
- Standalone setup and work on the client side.

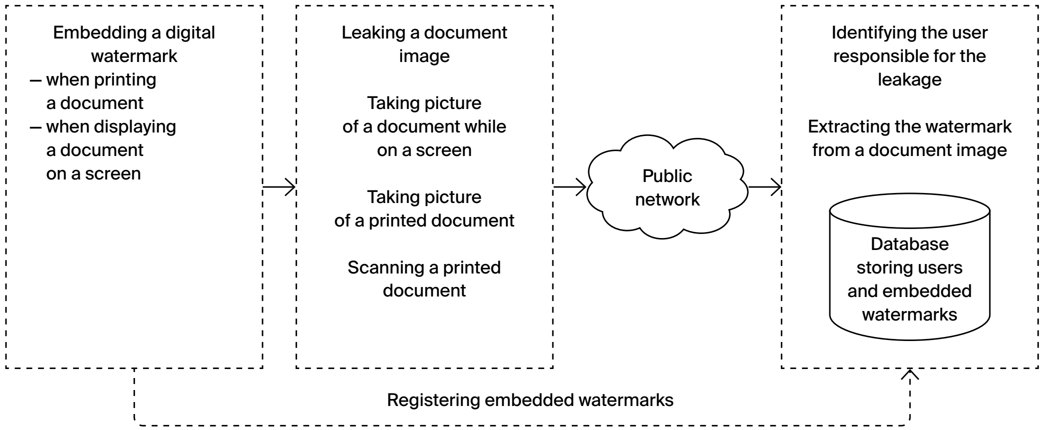
WHO IS DOCMARKING TARGET AUDIENCE?

- Government entities and public offices.
- Companies that would like to enforce their guides for handling classified documents.

SUPPORTED OPERATING SYSTEMS

Windows (32-bit, 64-bit), Linux (64-bit), including Astra Linux SE 1.6/1.7.

DOCMARKING WORKFLOW


Embedding a digital watermark
— when printing a document
— when displaying a document on a screen
Leaking a document image
Taking picture of a document while on a screen
Taking picture of a printed document
Scanning a printed document
Public network
Identifying the user responsible for the leakage
Extracting the watermark from a document image
Database storing users and embedded watermarks
Registering embedded watermarks

ECGHUB: IN-DEPTH ANALYSIS OF DIGITAL ECG

EcgHub is a 12-lead ECG labeling system and neural network models' collection for pathology prediction. The system allows to predict the presence or absence of several pathologies, as well as to perform and review the syndromic ECG markup based on the verified questionnaire, thus providing a dataset for further development of neural network models.

FEATURES AND ADVANTAGES

EcgHub is based on research results in the areas of digital signal processing and machine learning algorithms. The pathology classification system is based on deep neural networks. The expert-verified approach provides consistent ECG labeling for training and further development of predictive models for screening and diagnosis of cardiovascular diseases.

EcgHub provides:

- Trained neural networks for pathology prediction of digital ECG;
- Continuous development and refinement of neural network models, including fine-tuning for relatively small medical datasets;
- Adaptation of trained neural network models for pathology prediction of single-lead ECGs (cardiac chair, smart watches) as well as 24-hour ECGs (Holter monitoring);
- A consistent syndromic markup system to provide qualitative data for training predictive models;
- Integration of neural network models into the customer's digital circuit or remote access to the service at ISP RAS;
- Applying the markup system in the education of modern functional diagnostics professionals;
- Development of an automated population screening system.

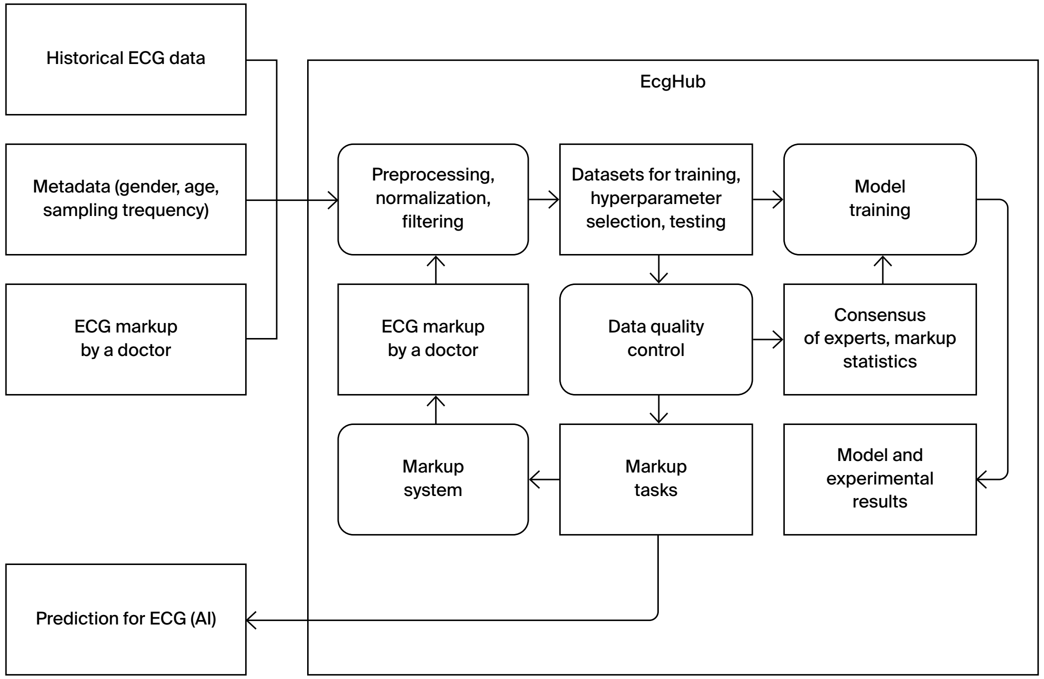
WHO IS ECGHUB TARGET AUDIENCE?

- Medical institutions: the prediction of neural network models can be used as a second opinion;
- Educational institutions: verified datasets allow evaluating the knowledge of students or novice doctors of relevant specialties;
- Developers of devices and applications that perform ECG diagnostics autonomously.

ECGHUB DEPLOYMENT STORIES

The neural network model of 12-channel ECG classification was trained on data from the Republic of Tatarstan, integrated as a proof of concept into the “Unified Cardiologist” system, and tested on ECG data from different regions (Republic of Tatarstan, Moscow, Velikiy Novgorod).

WORKFLOW


Historical ECG data
Metadata (gender, age, sampling trequency)
ECG markup by a doctor
Prediction for ECG (AI)
EcgHub
Preprocessing, normalization, filtering
Datasets for training, hyperparameter selection, testing
Model training
ECG markup by a doctor
Data quality control
Consensus of experts, markup statistics
Markup system
Markup tasks
Model and experimental results

3 OTHER TECHNOLOGIES

CONSTRUCTIVITY 4D: INDEXING, SEARCHING, AND ANALYZING OF LARGE-SCALE SPATIAL/TEMPORAL DATA

Constructivity 4D is a technology for creating innovative software services that are capable of processing highly dynamic scenes and vast arrays of spatial and temporal data. It performs visual analysis of millions of objects with individual geometry and dynamic behavior. Constructivity is deployed within the Synchro system (Bentley Systems) that is used for 4D modeling of extremely large construction sites.

FEATURES AND ADVANTAGES

Constructivity 4D is a production level technology that puts together original methods of spatio-temporal indexing, search and qualitative and quantitative data analysis. Developed methods account for the specifics of objects' geometric representation, complex organization and the apriori known nature of their dynamic changes.

Constructivity 4D provides:

Support for a well-developed set of operations:

- Temporal operations implement classical interval algebra introduced by Allen with respect to time stamps of discrete events and their intervals.
    - Metric operations allow determining the individual properties of geometric objects and the characteristics of their mutual arrangement. Diameter, area, volume, center of mass, planar projections, and distances between objects can be calculated for solid geometric objects.
    - Topological operations are intended to classify the relative location of objects and to establish the facts of their coincidence, intersection, coverage, touch, overlap or collision. In contrast with known topological models such as DE-9IM, RCC-8, RCC-3D, these operations allow constructive implementation and are applicable for the analysis of complex objects.

  - Orientational operations generalize known Frank’s and Freksa’s relative orientation calculi, cardinal direction calculi (CDC), oriented point relation algebra (OPRA) and are applicable for the analysis of objects with extended boundaries.
- Efficient query execution and typical problems solving, in particular, queries for reconstructing a scene at a given point in time, retrieving objects in a given spatial region, finding nearest neighbors, determining static and dynamic collisions, and conflict-free routing in a global dynamic environment are effectively resolved.
- A spatial-temporal indexing system including binary event trees, spatial decomposition trees, bounding volume trees, object cluster trees, space occupation trees.
- A hybrid computational strategy for determining collisions in scenes that combines methods for precise collision determination, collision localization methods using spatial decomposition, methods of hierarchies of bounding volumes, temporal coherence methods.
- An object-oriented library implemented in C++ that includes extensible set of classes, interfaces and related methods for specifying spatial-temporal data and executing typical queries.
- An original method for navigation in global dynamic environment that is based on extracting spatial, metric and topological information from geometric representation of 3D scenes and its concerted usage on path planning.
- Various options for extending the library so that it can be used both in new software applications development and in legacy applications.

WHO IS CONSTRUCTIVITY 4D TARGET AUDIENCE?

The technology is used for creating application systems in vastly different fields, including but not limited to: computer graphics and animation, geoinformatics, scientific visualization, design and manufacturing automation, robotics, logistics, project management and scheduling.

CONSTRUCTIVITY 4D DEPLOYMENT STORIES

The technology has been successfully deployed within the Synchro software system (https://www.bentley.com/software/synchro/) that is designed for visual 4D-modeling, planning and management of large-scale industrial projects in the construction and infrastructure areas, as well as others. Synchro is used in more than 300 companies in 36 countries.

VALIDBIM: A SERVICE FOR INFORMATION MODEL VERIFICATION IN ARCHITECTURE AND CONSTRUCTION

VALIDBIM is a service for verifying information models that are used in construction and architecture works. The models should be written in the IFC SPF format and support functional compatibility for applications on the Building Information Modeling (BIM) Level 3. This level of technological maturity in the Bew & Richards model assumes BIM application interoperability and integration into advanced multidisciplinary software systems that are used for various design activities in architecture, engineering, construction and buildings and structures management. Certificate of software registration No. 2023667675.

FEATURES AND ADVANTAGES

VALIDBIM is a service that is capable of bringing the advanced software, which satisfies requirements of technical, syntax and semantic interoperability and BIM maturity, to the new level.

VALIDBIM provides:

- Verifying whether construction and architecture information models comply with international and national Industry Foundation Classes (IFC) standards (ISO 16739; GOST R 10.0.02: 2019) and STEP Physical File (SPF) standards (ISO 10303-21; GOST R ISO 10303-21: 2002, 2022).
- Checking syntax and link consistency of models’ file data.
- Complete and mathematically rigorous checking model data semantics based on a formal scheme set up using the EXPRESS language of object-oriented modeling.

The following correctness checks are performed:

- object types (ENTITY),
- number and types of object attributes, including enumeration attributes (SELECT),
- mandatory and optional attributes (OPTIONAL),
- length limits for symbol and binary strings (STRING, BINARY),
- collection sizes (BAG, SET, LIST, ARRAY),

  - mandatory and optional collection attributes (OPTIONAL ARRAY),
  - uniqueness, when collection elements represent sets (SET, LIST OF UNIQUE),
  - collection size and contents, when collections are inverse attributes (INVERSE).

  The following satisfiability checks are performed:
  - rules for ranges of simple types (TYPE WHERE),
  - rules for coherency of object attributes (ENTITY WHERE),
  - rules for uniqueness of object attributes (ENTITY UNIQUE),
  - global rules for coherency of object collections (RULE).

- Verifying software for which technical, syntax and semantic interoperability of BIM Level3 is claimed.
- Supporting latest IFC standard versions including IFC 2x3, IFC 4, and IFC 4.3.
- Recording detected errors and mailing to registered users.
- Quick processing user verification jobs.

WHO ARE VALIDBIM TARGET AUDIENCE?

- BIM software developers that want to create advanced interoperable programs and are in need of reliable BIM verification tools.
- BIM users wanting to ascertain the quality and completes of their information models and to ensure that the models can be processed by tools coming from different vendors.

VALIDBIM DEPLOYMENT STORIES

VALIDBIM is developed and deployed as a part of BIM National Platform on bim.ispras.ru. The service is actively used by Russian developers and users of BIM software.

SMARTIDS: A WEB EDITOR FOR MACHINE-READABLE REQUIREMENTS IN CONSTRUCTION

SmartIDS is designed for preparing specifications for digital information models (DIM) in construction based on open IFC (Industry Foundation Classes, ISO 16739) and IDS standards (Information Delivery Specification, buildingSMART). It allows expressing DIM requirements as IDS specifications for further verification. SmartIDS transforms regulatory documents and technical regulations of construction industry, which are written in natural language, to the machine-readable IDS format. Certificate of software registration No. 2024669861.

FEATURES AND ADVANTAGES

SmartIDS is a web application that allows viewing, editing and documenting formalized requirements that are presented in the format of the IDS open standard, current version 1.0. Machine-readable DIM requirements are described in terms of the foundational IFC scheme that serves as a single conceptual and formalized data description of a construction project and fully complies to established international and national standards. An important feature of SmartIDS is advanced context help and reference for IFC schemes being applied.

SmartIDS provides:

- Loading an IDS file and checking whether it conforms to the standard.
- Viewing, editing, adding, removing requirement specifications in a multi-window web interface (editing elements for specifying conditions on object categories are provided. The categories include relations, attributes, classifiers, features, and materials supported by the latest version of the IDS standard).
- Advanced context help with filling correctly requirement specification forms (it is possible to select data types names defined in the IFC scheme and predefined types of elements, materials, and attributes from the displayed list or hierarchy).
- Validation (correctness checking) and exporting requirement specifications as an IDS file.

UNIQUE FEATURES

- IDS+IFC standards working together ensure the correct functioning of formalized rules.
- Editing features with context help ensure quick input of valid data and guarantee the requirement specification correctness and its conformity to the IFC scheme.
- Requirement coordination with the national construction information classification.
- Risks commonly found in closed implementations are lowered due to employing the IDS and IFC open standards.
- Extending the IDS scheme is supported so that more checking with wider scope is possible.

WHO IS SMARTIDS TARGET AUDIENCE?

- Government services: creating a single registry of digital requirements and versioning those.
- Customers: creating own digital requirements and reusing them.
- Contractors: performing internal audit and configuring CAD/CAE systems for digital requirements.
- Certification authorities: a single logic of requirements for all customers and absence of software lockdown.
- BIM software developers: following common open standards and supporting DIM automatic verification.

SMARTIDS DEPLOYMENT STORIES

SmartIDS is implemented and deployed within the BIM national platform at bim.ispras.ru. It can also be found at the project web site, ids.ispras.ru. Currently the editor is used successfully within interested companies and organizations for creating own machine-readable DIM requirements.

DIGITEF: COMPUTER MODELING PLATFORM

DigiTEF is a software platform for developing digital modeling applications, performing computer modeling and engineer analysis for industrial technical and scientific tasks. DigiTEF allows solving various application problems of gas dynamics, aerodynamics, hydrodynamics, and acoustics as well as performing coupled calculations.

DigiTEF is included in the Unified Register of Russian Programs (No. 5377).

FEATURES AND ADVANTAGES

DigiTEF is being developed based on open source software and ISP RAS know-how libraries and modules. Using own research results allows, for certain problem types, to calculate solutions that are more precise and correct compared to the state of the art counterparts. DigiTEF core performance and accuracy evaluations compared with ANSYS Fluent and Star CCM+ showed similar (and in some cases lower) computational costs with the same accuracy.

DigiTEF provides:

- open source code (allows improving data safety as well as controlling and adapting implemented algorithms for specific problems);
- intuitive graphical user interface that can be adapted for a specific enterprise and problems being solved;
- no limitations on user number, computational cores, cell number in meshes, which allows to decrease costs for computations and usage;
- modern algorithms on models via synchronizing technical level with international community;
- automation tools for computation and model integration that allow integrated research of technical objects;
- developing additional components according to custom requirements;
- using high-performance distributed systems (like clusters and supercomputers) for speeding up computations.

WHO ARE DIGITEF TARGET AUDIENCE?

DigiTEF is designed for use in companies and enterprises of resource-intensive industries. Using DigiTEF allows increasing engineering efficiency and profit margins as well as reducing costs and complexity when implementing industrial projects.

DEPLOYMENT STORIES

DigiTEF is used in several projects in the fields of wind energy, aerospace, aviation, shipbuilding, metallurgy, as well as in the oil and gas industry. DigiTEF open source modules are successfully used in Institut Pprime (France), Korea Atomic Energy Research Institute (Korea), Universität der Bundeswehr München (Germany), Northwestern Polytechnical University (China), Ocean University of China, Embry-Riddle University (USA), California Institute of Technology (USA), etc.

SYSTEM REQUIREMENTS

DigiTEF supports Linux OS, including Astra Linux, and Microsoft Windows 10. DigiTEF requires at least 4-core x86-64 processor, 16 GB RAM, and 100 GB disk space.

DigiTEF supports parallel computing. Using up to 1536 computational cores was tested.

Ivannikov Institute for System Programming
of the Russian Academy of Sciences

25, Alexander Solzhenitsyn str. 109004 Moscow, Russia
E-mail: scsec@ispras.ru

Ivannikov Institute for System Programming of the RAS 2024