Создание доверенных систем. Модели долгосрочного развития

Арутюн Аветисян
директор ИСП РАН
академик РАН
arut@ispras.ru
15 ноября 2022 г.

ИСП РАН

Основа ИСП РАН – постоянно развивающаяся научная школа

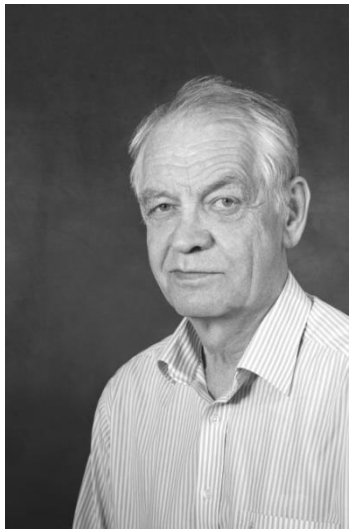
ИСП РАН

С.А. Лебедев

В.А. Мельников

БЭСМ-6 в музее науки Лондона

В.П. Иванников



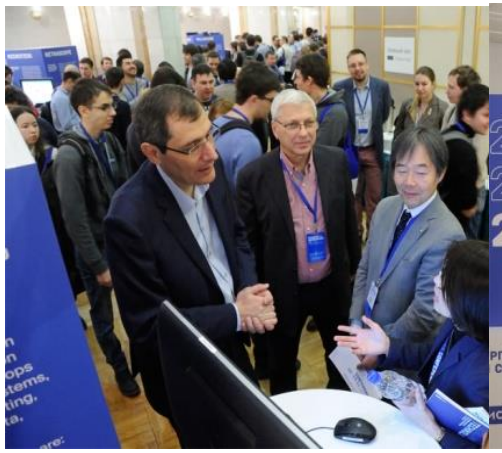
Л.Н. Королёв

«Если мы глубже разберёмся в этом эпохальном советском суперкомпьютере, это позволит пересмотреть заявления времён холодной войны об отставании русской технологии, а также подтвердить или развеять мифы о технологическом совершенстве наших союзников».

Doron Swade, senior curator of computing and information technology

2018: 70 лет IT*
2019: 25 лет ИСП РАН
2023: 75 лет IT*

***в России и странах постсоветского пространства**



- ✓ **Индустриальные исследования и внедрение научных разработок (анализ данных, анализ программ)**
- ✓ **Кафедры системного программирования на факультетах ВМК МГУ, ФУПМ МФТИ, ФКН ВШЭ**
- ✓ **Три лаборатории за пределами ИСП РАН**

ИСП РАН

Кибербезопасность: вызовы и решения

Ошибки в ПО – основная причина уязвимости систем

ИСП РАН

Принципиальное наличие уязвимостей* в ПО и аппаратуре:

функциональные, архитектурные, программного кода/микрокода.

Границы между ошибками программиста, закладками и НДВ размыты

Утечка информации о уязвимостях

Хакеры «для интереса»

- Взломы ради известности
- Ограничены ресурсы
- Только известные эксплойты

Хакеры «для ущерба»

- Вандализм
- Ограничены ресурсы

Преступность

- Взлом ради прибыли
- Существенные ресурсы
- Синдикаты
- Специально разработанные программы для кражи данных

Спецслужбы

- Атаки на критические информационные инфраструктуры, промышленный шпионаж
- Практически неограниченные ресурсы
- Сложное ВПО и программы для взлома
- Постоянные угрозы

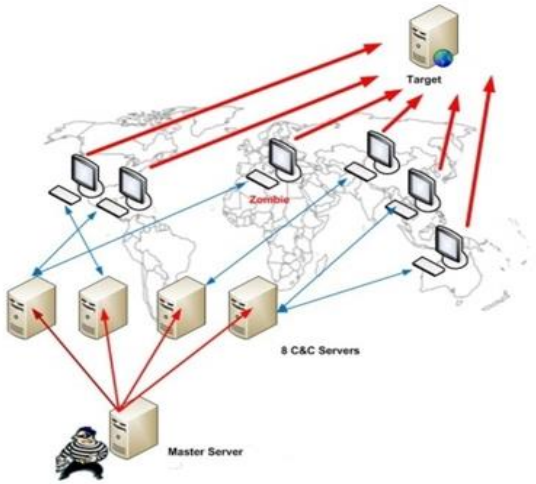
Рост ресурсов и сложности атак

Эскалация размеров и сложности (миллиарды строк кода на GitHub),
сложность среды разработки и сборки, облачные платформы и интернет вещей

Уязвимости: причины и следствия

Аварии на критических обьектах

Перехват управления



Бот-неты

Уязвимости

Кража паролей

Кража информации о кредитных картах

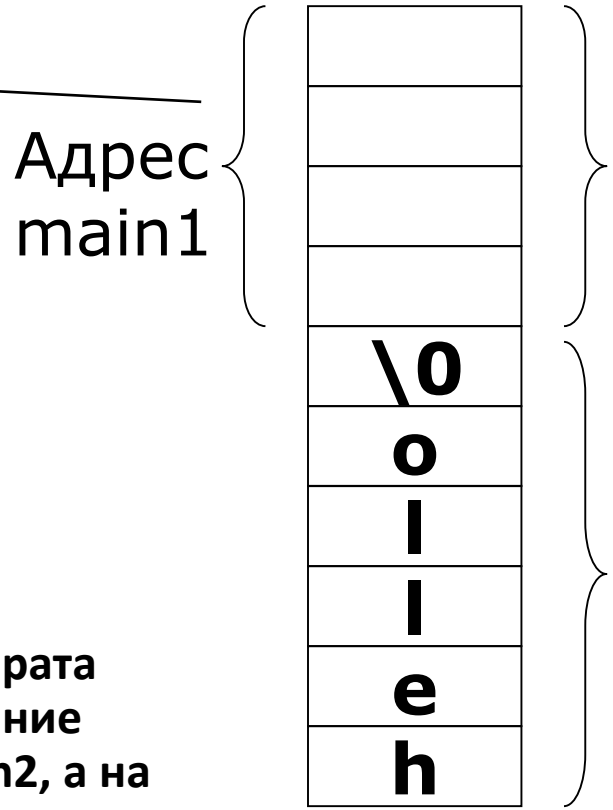
Компьютерные атаки осуществляются путём эксплуатации дефектов в ПО и аппаратуре

Пример: переполнение буфера (уровень исходного кода)

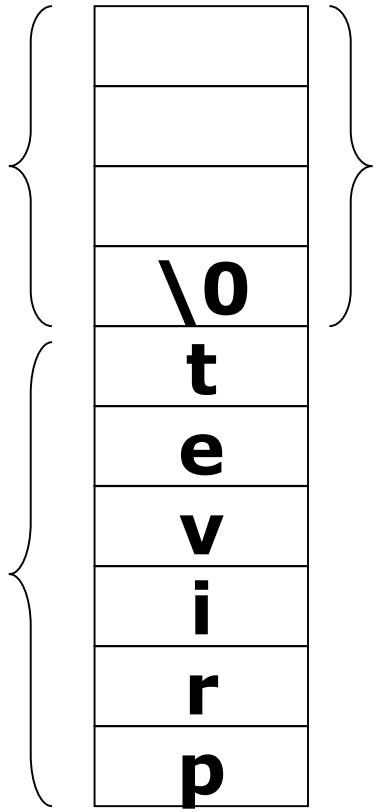
ИСП РАН

```
void f(char * p)
{    char s[6];
     strcpy(s,p);
}
void main1 ()
{    f("hello");
}
void main2 ()
{    f("privet");
}
```

Стек после выполнения функции f, вызванной из main1



Стек после выполнения функции f, вызванной из main2



На место адреса main2 записали лишний байт

В случае main2 адрес возврата перезапишется, и управление будет передано не на main2, а на другой участок кода

Пример: небезопасная оптимизация (среда компиляции и сборки)

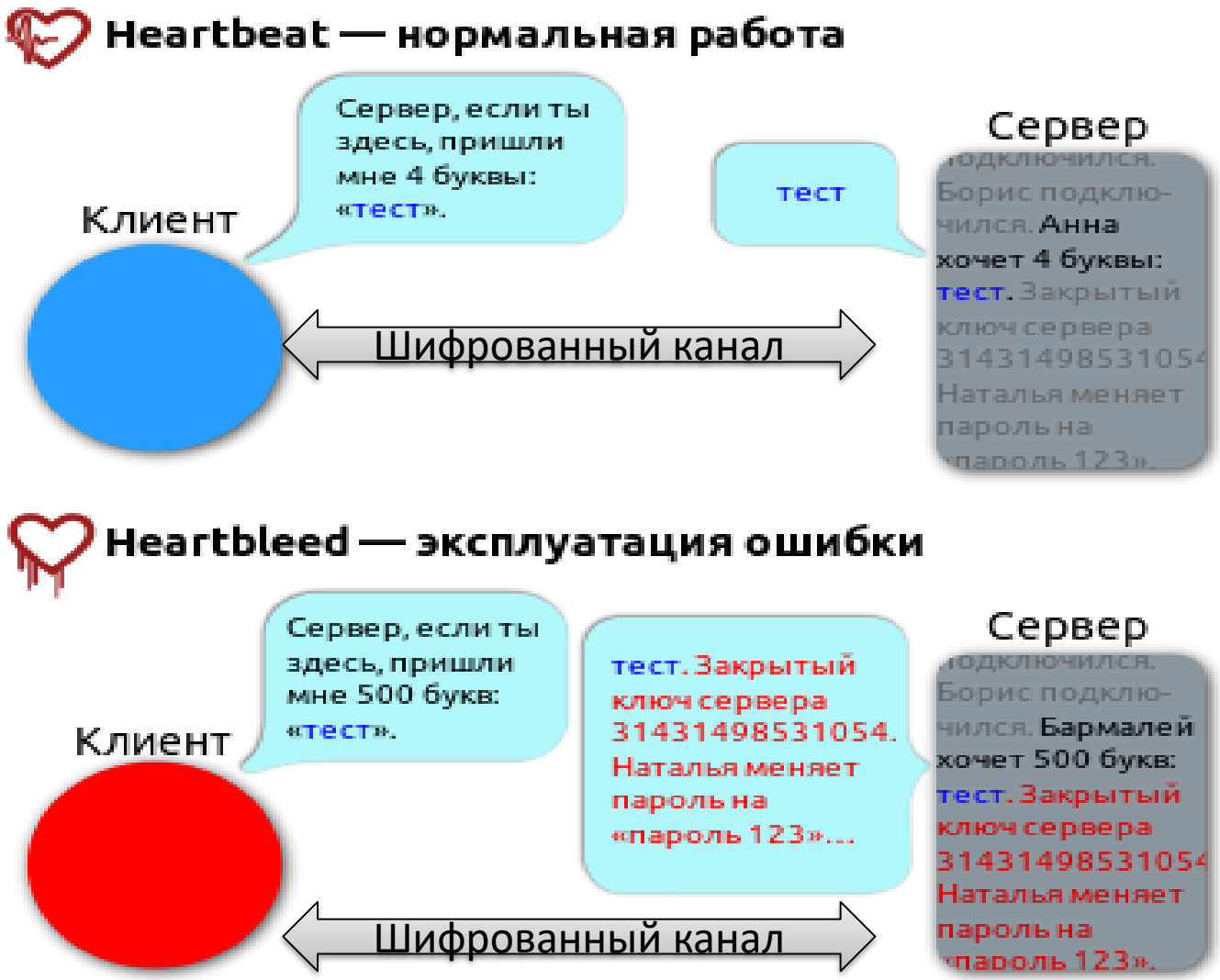
ИСП РАН

```
bool auth() {
  char buf[N];
  bool res;

  read_password(buf, N);
  res = check_password(buf);

  memset(buf, 0, N);
  return res;
}
```

Компилятор удаляет обнуление буфера с паролем, т.к. с его точки зрения после обнуления буфер не используется.
При этом пароль останется на стеке.

Пример: уязвимость Heartbleed (библиотека OpenSSL)

ИСП РАН



500000 сайтов заражено
$500 млн потерь

- **Ошибка чтения данных за границей буфера: злоумышленник контролирует длину посланного текста**
- **Происходит утечка пользовательских данных**
- **Весь обмен данными строго следует зашифрованному протоколу**

Пример: слабость кодирования обработки входных данных

ИСП РАН

Типы ошибки: слабость кодирования обработки входных данных, переполнение буфера

Прежде чем достичь места реализации ошибки, введённые извне данные «проходят» по многим функциям разных модулей

Модуль с функцией считывания файла-архива

```
[tainted] Call of read
95              if(read(fd, file_hdr, SIZEOF_NEWLHD) != SIZEOF_NEWLHD) {
96                  free(file_hdr);
97                  return NULL;
98              }
99              file_hdr->flags = unrar_endian_convert_16(file_hdr->flags);
100             file_hdr->head_size = unrar_endian_convert_16(file_hdr->head_size);
101             file_hdr->pack_size = unrar_endian_convert_32(file_hdr->pack_size);
102             file_hdr->unpack_size = unrar_endian_convert_32(file_hdr->unpack_size);
103             file_hdr->file_crc = unrar_endian_convert_32(file_hdr->file_crc);
Composite 'file_hdr' taints element 'file_hdr->name_size'
104             file_hdr->name_size = unrar_endian_convert_16(file_hdr->name_size);
116             return file_hdr;
```

Функция в другом модуле. Ранее считанные извне данные определяют размер копируемой памяти

```
1484            /* Enter response type, length and copy payload */
1485            *bp++ = TLS1_HB_RESPONSE;
1486            s2n(payload, bp);
Tainted data from /home/shimnik/openssl/ssl/s3_pkt.c+239 reached a sink.
8. [SINK] *(s->s3->rrec.data + @ ) reaches the sink
1487            memcpy(bp, pl, payload);
1488            bp += payload;
1489            /* Random paddi
1490            RAND_pseudo_byt
1491
1492            r = dtls1_write                    3EAT, buffer, 3 + payload + paddi
                Go to declaration
                Go to definition
                Show usage
```

От обучения до реагирования

ИСП РАН

- **Недостаточно использовать классические методы защиты (защита по периметру, проверка доступа, антивирусы и др.)**
- **Необходима разработка новых моделей, методов и технологий в области анализа и трансформации программ**

Найти и устранить	**Обеспечить устойчивость**
максимальное количество уязвимостей в исполняемом коде во время **жизненного цикла разработки доверенного ПО**	программной системы, затруднив эксплуатацию существующих уязвимостей или смягчив последствия их эксплуатации

Обучение	Требования	Проектирование	Реализация	Проверка	Выпуск	Реагирование
Обучение основам безопасности	Задание требований безопасности Создание контрольных условий качества и панелей ошибок Оценка рисков безопасности и конфиденциальности	Задание требований проектирования Анализ возможных направлений для злоумышленных атак Моделирование рисков	Применение утвержденных инструментов Отказ от небезопасных функций Статический анализ	Динамический анализ Нечеткое тестирование Проверка возможных направлений для злоумышленных атак	Планирование реагирования на инциденты Окончательная проверка безопасности Архивация выпуска	Выполнение плана реагирования на инциденты

Жизненный цикл разработки доверенного ПО: стандарты

ИСП РАН

США (NIST и другие)

NIST: Национальный институт стандартов и технологий

С 1999: развитие госстандартов Common Criteria

С 2004: появление Microsoft Security Development Lifecycle

Инструменты анализа выбираются сертификационными лабораториями

Разработчики ПО обязаны пользоваться инструментами требуемого уровня. Заложено непрерывное обновление стандартов по мере развития технологий

Принятие нормативных документов привело к взрывному росту рынка технологий анализа

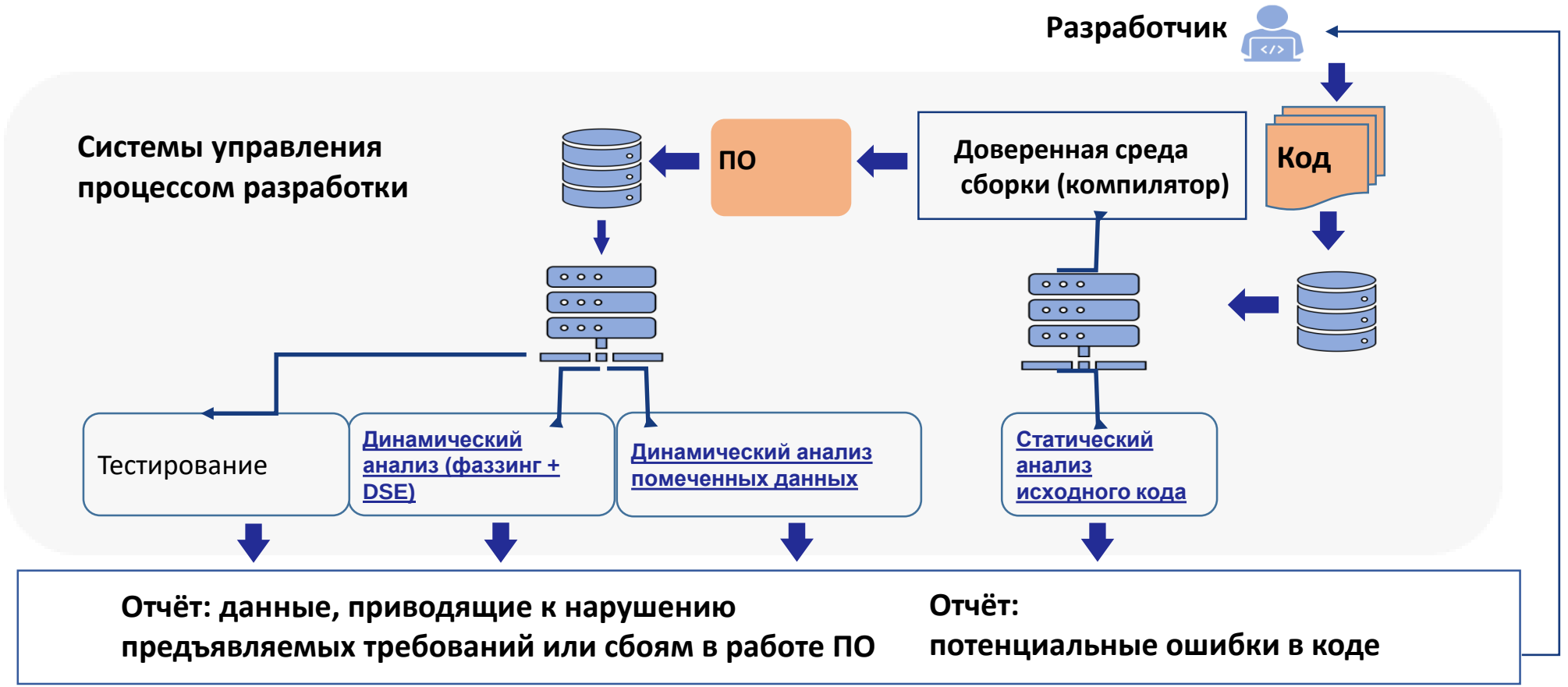
Россия (ФСТЭК и другие)

ГОСТ Р 56939-2016 «Защита информации. Разработка безопасного программного обеспечения. Общие требования»

Методика выявления уязвимостей и недекларированных возможностей в программном обеспечении

В разработке:
ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Руководство по проведению статического анализа. Общие требования**»
ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Руководство по проведению динамического анализа. Общие требования**»
ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. **Безопасный компилятор языков Си/Си++. Общие требования**»
…

Жизненный цикл разработки доверенного ПО: схема



!! Проблемы: сложные инструменты, траты ресурсов на анализ миллионов строк кода

Жизненный цикл разработки доверенного ПО: технологии I

ИСП РАН

Статический анализ исходного кода	**Svace (ИСП РАН, Россия)** SAMSUNG kaspersky HUAWEI PROFESSIONAL Postgres КОД безопасности …	**Аналоги:** Klocwork (Perforce, США), Coverity (Synopsys, США), Fortify (MicroFocus (ранее HP), США, Великобритания). **Открытые**: Clang Static Analyzer, SpotBugs
Фаззинг + DSE (динамический анализ)	**ИСП Crusher (ИСП РАН, Россия)** КОД безопасности kaspersky PROFESSIONAL Postgres ASTRA LINUX® …	**Аналоги, недоступные в РФ:** Peach Fuzzer (США, Peach Tech), Synopsys Defensics (Synopsys, США), MAYHEM (ForAllSecure, США). **Открытые**: angr (США), American Fuzzy Lop (США), Driller (США)
Динамический анализ помеченных данных	**Блесна (ИСП РАН, Россия)** Предназначена для испытательных лабораторий, органов по сертификации и др.	**Аналоги, недоступные в РФ**: MAYHEM (ForAllSecure, США), TETRANE's Timeless Analysis (Tetrane, США), APAC (2013-2015), VET (2013), CGC (2016), CHESS (2019) (проекты DARPA)

Жизненный цикл разработки доверенного ПО: технологии II

ИСП РАН

Безопасная компиляция (не добавляющая ошибки в бинарный код)	**SafeC (ИСП РАН, Россия)** kaspersky Postgres PROFESSIONAL ASTRA LINUX®	**Аналоги:** отдельные части в GCC и Clang (флаги, санитайзеры), исследовательские или ограниченно применимые компиляторы и диверсификаторы (CompCert C, kcc, Selfrando, Oxymoron, Shuffler и др.)
Определение поверхности атаки	**Natch (ИСП РАН, Россия)** Предназначен для разработчиков безопасного ПО и для испытательных лабораторий Распространяется в сообществе	**Аналоги**: фреймворки Panda, Decaf (нет средств интроспекции), инструмент Panorama (исследовательский проект для Windows)

Технологии ИСП РАН используют более 100 компаний в России и за рубежом:
Samsung, Huawei, «Лаборатория Касперского», «РусБИТех», Postgres Professional и др.

Особенности применения SDLC: операционные системы

ИСП РАН

Статический анализ: аннотации служебных функций ОС (выделение и освобождение ресурсов, ввод-вывод и пр.),
анализ вызовов по указателю, восстановление графа вызовов

Динамический анализ: использование инструментов, предназначенных для ядра ОС (фаззер syzkaller)

- Специальная сборка ядра с поддержкой инструментации и санитайзеров
- Подготовка списка системных вызовов, для фаззинга которых будет генерироваться тестовая программа

Возможно использование гипервизора (создание снапшота – снимка состояния ОС – и запуска ОС на новых данных через гипервизор, начиная с этого снапшота)

Особенности применения SDLC: анализ для встраиваемых архитектур

ИСП РАН

Статический анализ: использование контролируемой сборки через кросс-компиляторы на серверной хост-системе, возможность совместного анализа кода всей системы вместе

Динамический анализ: использование подходов частичной эмуляции (выполнение тестируемой программы через QEMU) или удаленной инструментации (фаззинг на хост-системе, легковесная инструментация на целевой встраиваемой системе)

- Требуется снятие дампа памяти тестируемой программы в эмуляторе
- Требуется эмуляция системных вызовов и периферии

Особенности применения SDLC: БД и многокомпонентные программы

ИСП РАН

Требуется анализ зависимостей для поиска уязвимых компонент в многокомпонентной программе

Динамический анализ: фаззинг входных интерфейсов (базы данных: JSON-формат, выполнение и обработка SQL-запросов)

- Нашли зависание БД (20+ минут) на определенном запросе
- Нашли переполнение буфера при фаззинге XML-парсера
- Нашли целочисленное переполнение в библиотеке FreeImage, приводящее к отказу в обслуживании

Эффективный фаззинг внутренних интерфейсов требует написания функций-оберток, которые непосредственно передают данные от фаззера внутрь программы

- Такой обертке нужно передавать сложный контекст выполнения, который нужно инициализировать
- Для написания обертки нужен квалифицированный разработчик

Особенности применения SDLC: виртуальные машины и JIT-компиляторы

ИСП РАН

Статический анализ: написание аннотаций библиотек, настройка критичности предупреждения (например, для языка Java некоторые предупреждения имеют меньшую критичность)

Динамический анализ: инструментация интерпретируемого или компилируемого байткода и перехват исключений интерпретатора

- Фаззинг программ, выполняемых JIT-компилятором или виртуальной машиной, может потребовать их модификации

Научная специальность ВАК 1.2.4 «Кибербезопасность»

ИСП РАН

2018: принято решение Президиума РАН о новом научном направлении «Анализ, трансформация программ и кибербезопасность»*

2021: новая специальность ВАК «Кибербезопасность» утверждена Приказом Минобрнауки №118

Направления исследований:

- Анализ и систематизация уязвимостей
- Моделирование политик информационной безопасности, угроз и атак
- Методы, алгоритмы и средства пост-релизного глубокого анализа защищенности ПО
- Методы интеграции средств защиты на уровне аппаратуры и на уровне ПО
- Интеллектуальный масштабируемый мониторинг инцидентов безопасности в распределенных программно-аппаратных системах
- Масштабируемые средства интеллектуального анализа данных и процессов в распределенных системах

И другие

*По предложению ИСП РАН; поддержано ФСТЭК России, АО «Лаборатория Касперского», ОАО «РусБИТех».

ИСП РАН

Искусственный интеллект: новые вызовы в кибербезопасности

Искусственный интеллект: история

ИСП РАН

Искусственный интеллект

Компьютер выполняет «интеллектуальные» действия, которые до этого были доступны только человеку:

- Играет в шахматы
- Ведет простой диалог

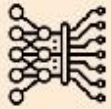
Решения основанные на правилах

Машинное обучение

Решения основанные **на данных:**

- **Этап обучения**: алгоритм получает примеры вопросов и правильных ответов
- **Этап применения**: алгоритм получает вопрос и выдает ответ

Глубокое обучение



Иерархия алгоритмов машинного обучения: ответы одних попадают в вход другим

Модель нейронной сети

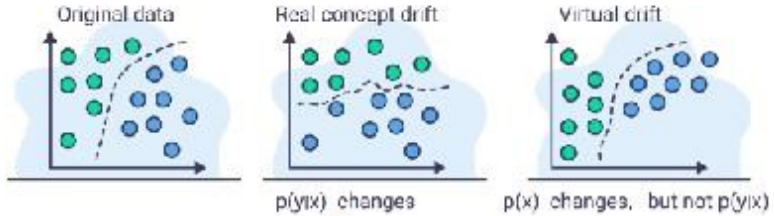
1950 — 1980 — 2010

Переход от построения **модели** объекта автоматизации к решению проблемы **по аналогии**

Проблемы машинного обучения

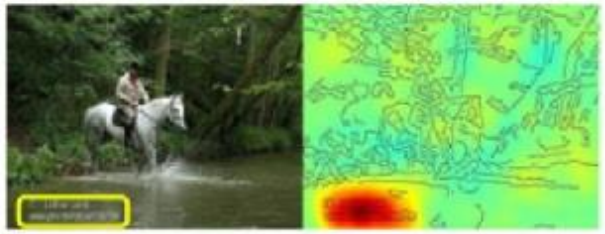
ИСП РАН

Проблемы разработки и эксплуатации (переобучение, сдвиги в данных в течение эксплуатации)

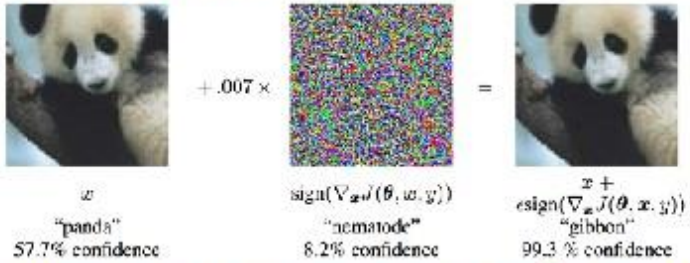


Предвзятость моделей

Интерпретация результатов

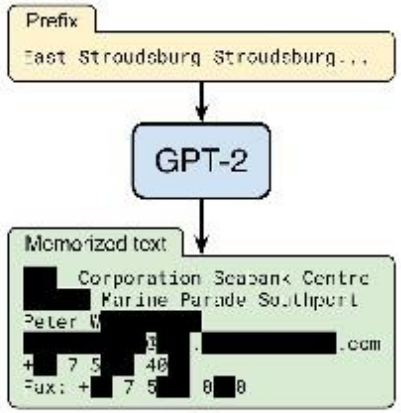


Состязательные атаки для манипуляции поведением системы



Встраивание закладок на этапе обучения

Извлечение конфиденциальных данных из обученных



«Кража» самих моделей

Положение дел в мире

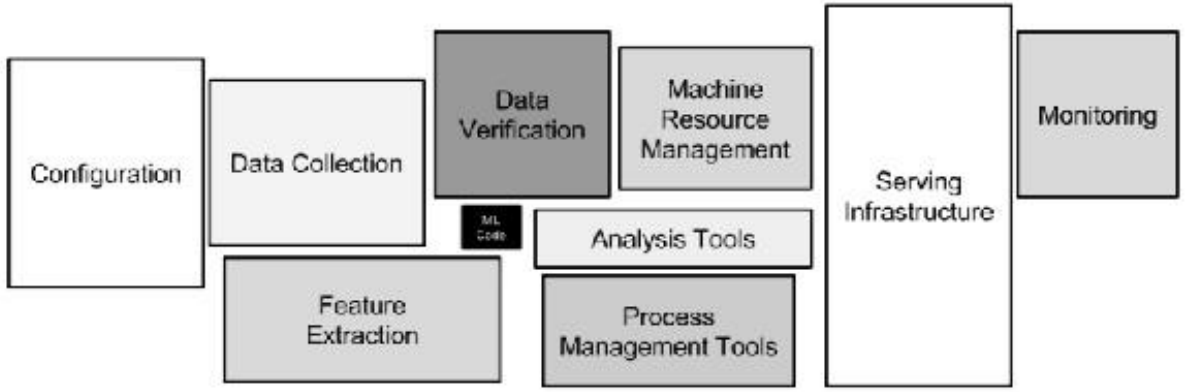
ИСП РАН

Проблемы создания доверенных систем, использующих ТИИ, уже активно рассматриваются:

- NIST AI Risk Management Framework (США)
- DIN DKE German Standardization Map on Artificial Intelligence (Германия)
- MITRE ATLAS, Adversarial Threat Landscape for Artificial-Intelligence Systems
- Google Responsible AI practices

Историческая аналогия

- 1999: развитие госстандартов Common Criteria
- 2004: появление Microsoft Security Development Lifecycle
- Продукты не соответствующие требованиям не допускаются к внедрению в госсекторе США и в крупных международных компаниях

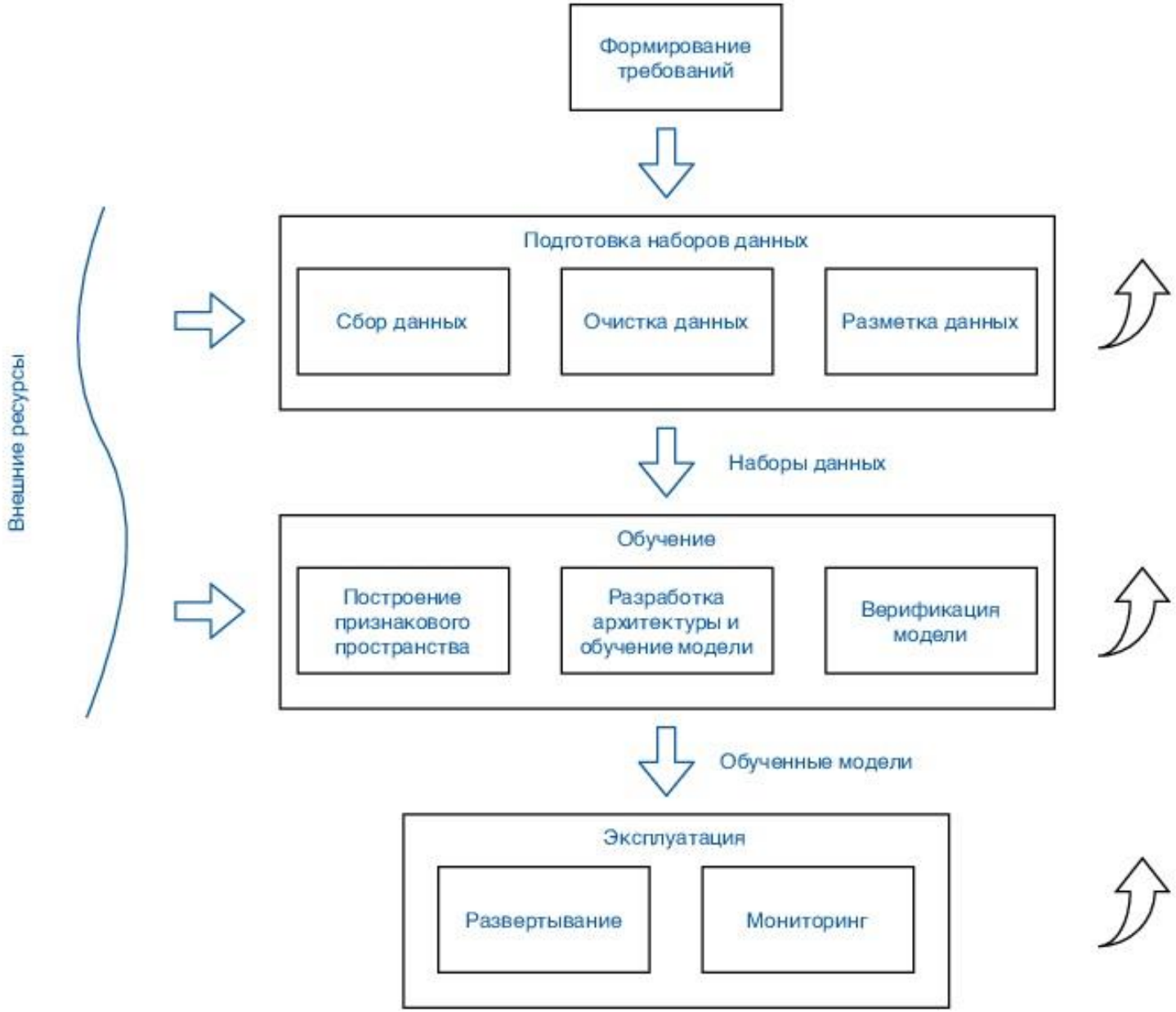


- Модели машинного обучения —- центральная, но не самая большая часть интеллектуальных систем
- Понятие «доверие» к программным системам определяется национальными стандартами (ГОСТ Р 56939-2016, приказ ФСТЭК №76 от 2 июня 2020 г.)
- Принципиальное отличие «интеллектуальных систем» - информация содержится не в программном коде, а в данных
- Для обеспечения доверия к интеллектуальным системам **отсутствует научно-технологическая база**

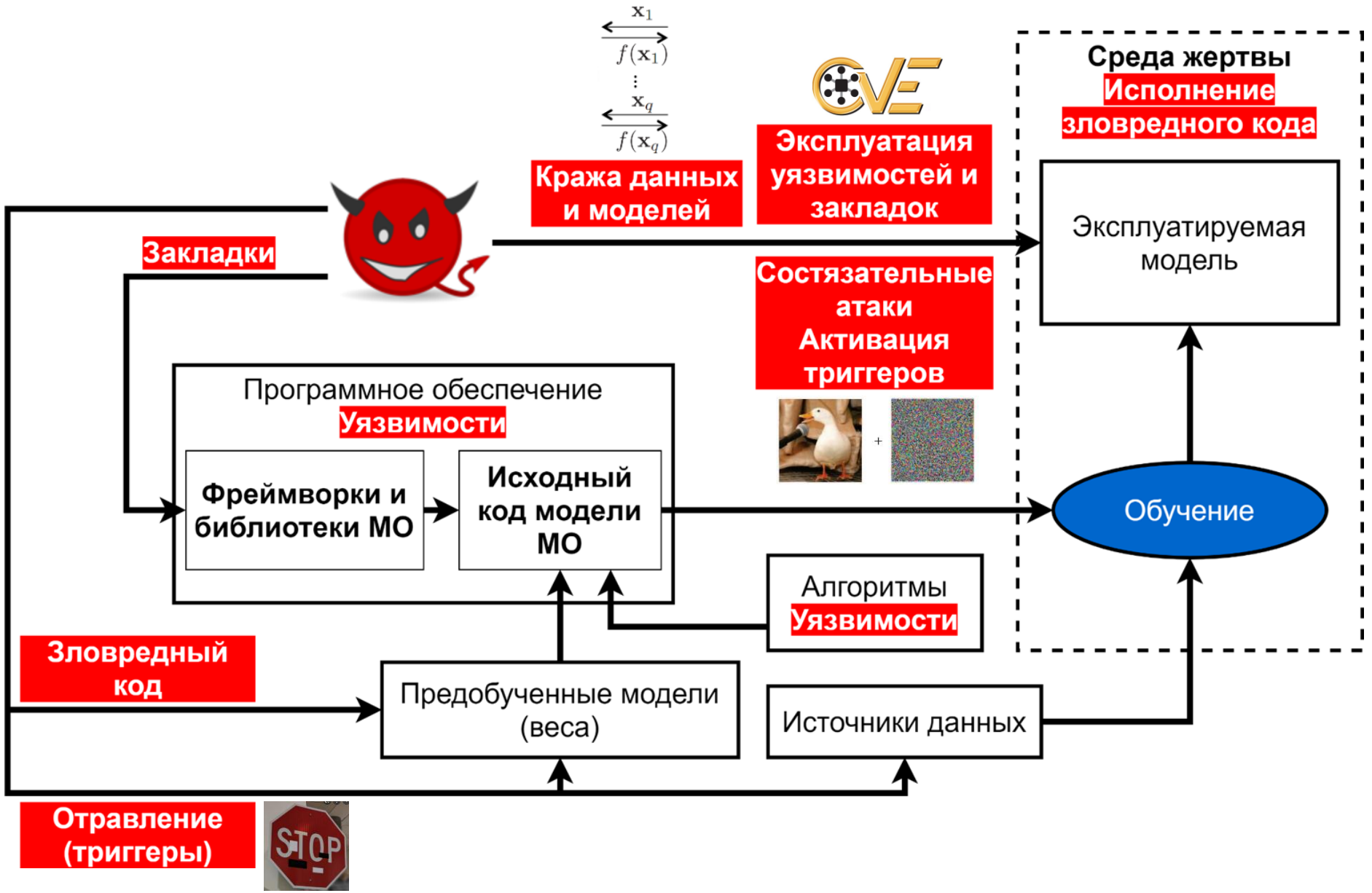
Интеллектуальные системы: технологический стек

ИСП РАН

Жизненный цикл машинного обучения



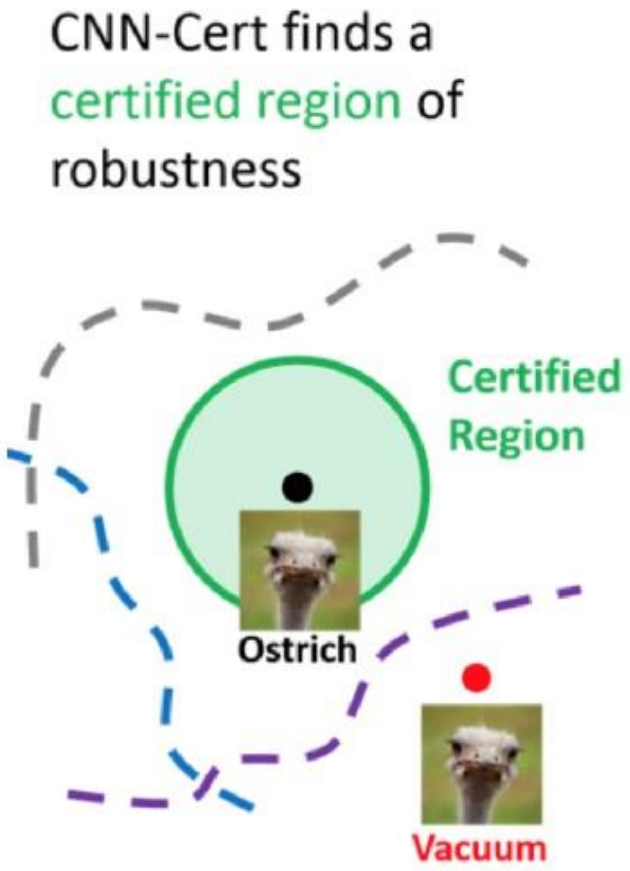
Атаки на модели машинного обучения



Противодействие атакам на машинное обучение

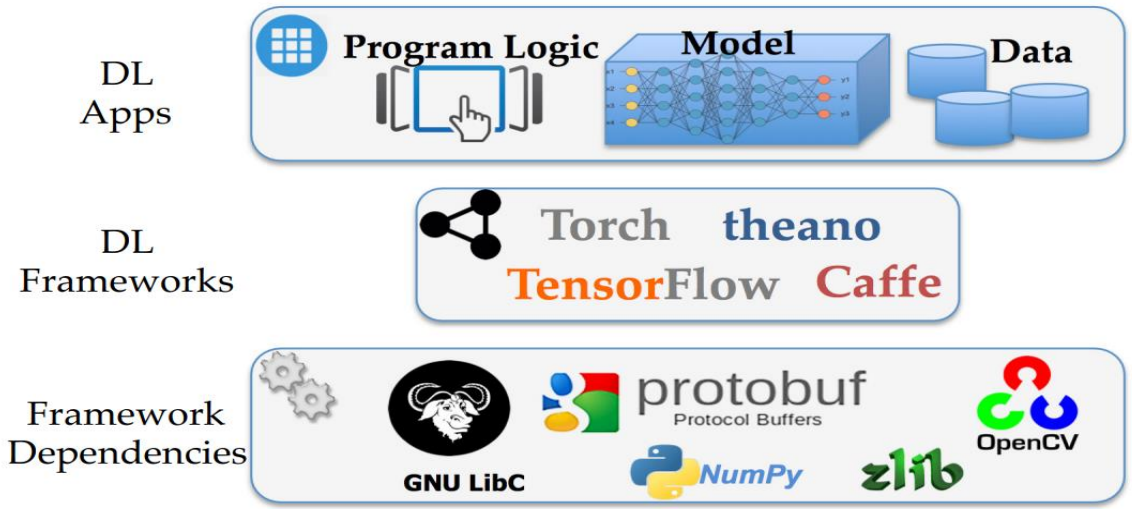
ИСП РАН

- Модели, устойчивые к заданному типу атак
  - Сетифицированная робастность
  - Состязательное обучение
- Выбор алгоритма обучения, для поиска устойчивых решений



Уязвимости в исходном коде фреймворков машинного обучения

ИСП РАН



Xiao, Q., Li, K., Zhang, D., & Xu, W. (2018). Security Risks in Deep Learning Implementations. 2018 IEEE Security and Privacy Workshops (SPW), 123-128.

- Классические уязвимости (CVE) в исходном коде фреймворков и библиотек расширяют поверхность атаки на эксплуатируемые модели
- Пример: переполнение буфера в библиотеке OpenCV, атака с помощью специально подготовленного BMP-изображения
- Кроме того, в исходном коде возможны **закладки**
- Необходимы **статический анализ** исходного кода фреймворков и создание их **доверенных версий**

7 миллионов строк кода PyTorch

TensorFlow **10 миллионов строк кода**

! ЭТО ЛИШЬ ЧАСТЬ СТЕКА ПО АНАЛИЗУ ДАННЫХ, НА КОТОРОМ ОСНОВАНЫ ТЕХНОЛОГИИ ИИ !

Фреймворки: пример уязвимости

ИСП РАН

> • WARNING
>
> `torch.load()` uses `pickle` module implicitly, which is known to be insecure. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling. Never load data that could have come from an untrusted source, or that could have been tampered with. **Only load data you trust**.

- Загрузка моделей из недоверенных источников (Hugging Face)
- PyTorch использует Pickle
- Pickle позволяет сохранять и загружать **любые** объекты Python
- То есть возможно дописать к сохраненной модели произвольную функцию, и вызвать ее через eval при загрузке

https://youtu.be/2ethDz9KnLk

✓ **Внедрение технологий ИИ без технологий обеспечения доверия невозможно**

✓ **В одиночку обеспечить доверие нельзя**

ИСП РАН

Модели развития доверенного системного ПО: что есть и что нужно?

Глобальная модель разработки на основе СПО

ИСП РАН

Проблемы

Неэффективность: многократное дублирование работ, затраты ресурсов
Технологические риски: недостаточно высокое качество исследования безопасности
Политические риски: ограничение доступа к технологиям

Модель не обеспечивает необходимый уровень доверия и долгосрочное устойчивое развитие

Модель, реализуемая ИСП РАН с партнёрами в РФ

ИСП РАН

Open Source:
сообщество +
инструменты

>73 млн чел. >4 млн чел.

GitHub SOURCEFORGE

GitLab Bitbucket

>30 млн чел. >10 млн чел.

Код

Патчи

ИСП РАН

Доверенная экосистема (+ репозиторий доверенного ПО)

- ✓ Технологический центр исследования безопасности ядра Linux (2021)
- ✓ Исследовательский центр доверенного ИИ (2021)
- ✓ Инфраструктура для систематического исследования безопасности критичных компонентов (2022)

Исследования

Кадры

Академическое сообщество

Компании*

Продукты
+ доп. возможности

Продукты
+ доп. возможности

Продукты
+ доп. возможности

Эффективность +
Продуктивность +
Trust +

ПЛЮСЫ:

- ✓ Единый язык науки, бизнеса и образования
- ✓ Генерация кадров и технологий
- ✓ Репозиторий доверенных технологий

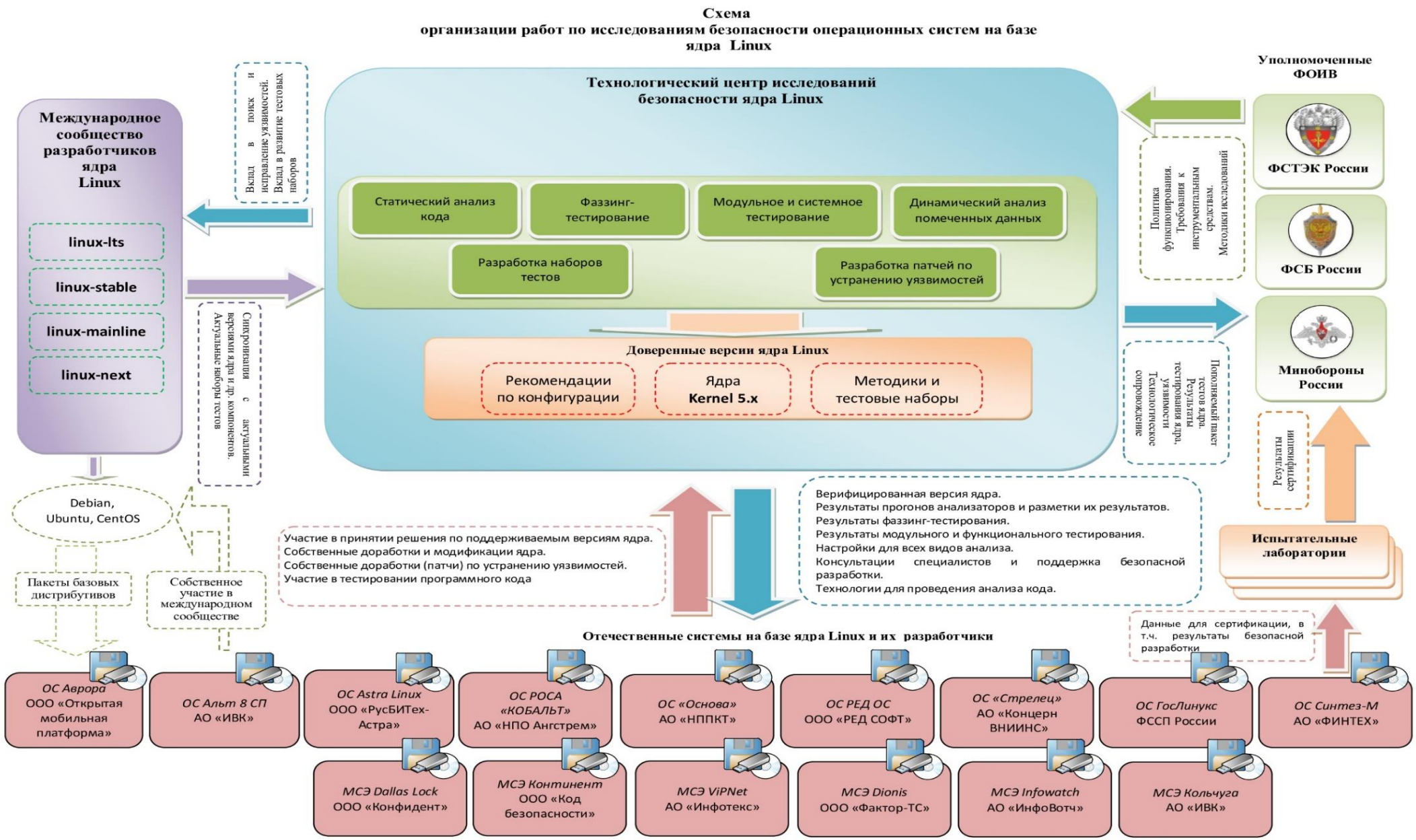
МИНУСЫ

~~Технологические риски~~
~~Политические риски~~

Ограниченное число проектов (человеческие ресурсы)

*АО «Аладдин Р.Д.», ООО «Айдеко», ООО «Базальт СПО», АО «Байкал электроникс», ООО «БЕЛЛСОФТ», АО «ИВК», АО «ИнфоТеКС», ООО «ИТБ», ООО «Код Безопасности», ООО «Конфидент», АО НТЦ «Модуль», АО «МЦСТ», ООО «Открытая мобильная платформа», АО «РАСУ», ООО «РЕД СОФТ», ООО «РусБИТех-Астра», АО МВП «Свемел», ООО «НТЦ ИТ РОСА», ООО «Фактор-ТС», АО «ФИНТЕХ», ООО «ЯНДЕКС.ОБЛАКО»

Доверенная экосистема ИСП РАН: Центр Linux



Центр Linux: результаты 2021-2022

ИСП РАН

- **Началось сопровождение ветки ядра Linux, основанной на стабильной версии 5.10.**
- **Подготовлены методики проведения исследования ядра, включая:**
  - ✓ статический анализ при помощи инструмента Svace;
  - ✓ системное и модульное тестирование (наполнение тестовыми наборами продолжается);
  - ✓ фаззинг-тестирование при помощи инструмента syzkaller;
  - ✓ проведение архитектурного анализа с целью определения поверхности атаки;
  - ✓ проведение анализа помеченных (чувствительных) данных.
- **Создана экспертная группа из представителей 21 компании, в рамках которой:**
  - ✓ формируются принципы функционирования Технологического центра;
  - ✓ проведена разметка более 6000 предупреждений инструмента Svace;
  - ✓ подготовлено ~100 исправлений ошибок в ядре; 53 патча уже приняты в основную ветку ядра.
- **Ведётся подготовка рекомендаций по конфигурированию ядра с целью повышения его безопасности.**
- **Ведётся разработка улучшений в ядре, нацеленных на повышение безопасности его работы на стадии развёртывания и инициализации.**

ПАРТНЕРЫ:

АО «Аладдин Р.Д.»
ООО «Айдеко»
ООО «Базальт СПО»
АО «Байкал электроникс»
ООО «БЕЛЛСОФТ»
АО «ИВК»
АО «ИнфоТеКС»
ООО «ИТБ»
ООО «Код Безопасности»
ООО «Конфидент»
АО НТЦ «Модуль»
АО «МЦСТ»
ООО «ОМП»
АО «РАСУ»
ООО «РЕД СОФТ»
ООО «РусБИТех-Астра»
АО МВП «Свемел»
ООО «НТЦ ИТ РОСА»
ООО «Фактор-ТС»
АО «ФИНТЕХ»
ООО «ЯНДЕКС.ОБЛАКО»

Центр ИИ: результаты анализа фреймворков

ИСП РАН

- **Создана аппаратно-программная инфраструктура для обеспечения доверия к базовым фреймворкам машинного обучения (TensorFlow, PyTorch)**
  - ✓ Первичный анализ исходного кода
  - ✓ Постоянная проверка новых изменений кода
  - ✓ Синхронизация с оригинальными открытыми версиями
- **Разработаны образцы доверенных фреймворков машинного обучения. Тестирование осуществляется индустриальными партнерами центра**

kaspersky ЕС-лизинг ТЕХНОПРОМ IPC InterProCom

ПОДРОБНОСТИ

ЗАДАЧА

Создать доверенные версии фреймворков Tensorflow и PyTorch

РЕШЕНИЕ

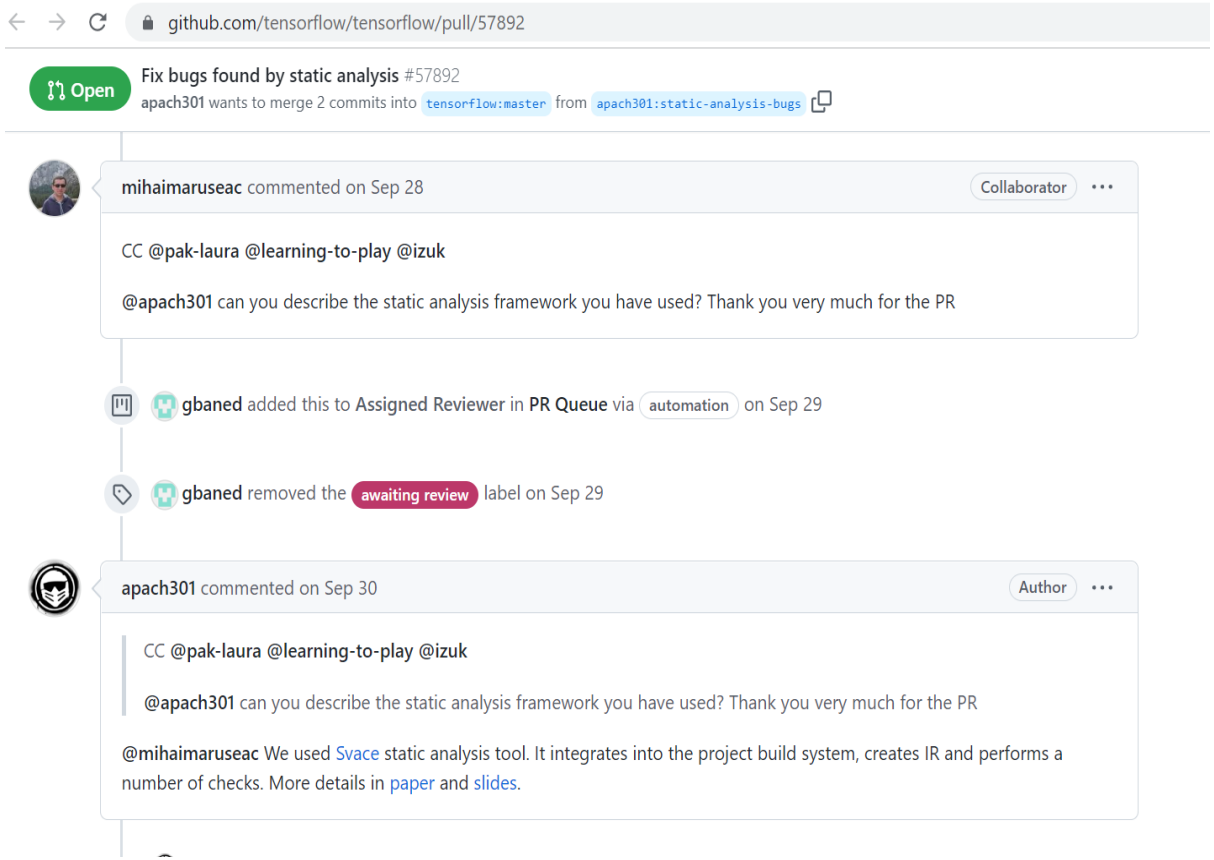
Проведён анализ поверхности атаки. Перспективным вектором атаки был выбран компонент загрузки моделей. Кроме того, для фреймворка TensorFlow были восстановлены фаззинг-цели в проекте OSS-Fuzz. Фаззинг фреймворков производился с помощью инструмента **Sydr** (ИСП РАН). Исходный код был проанализирован инструментом **Svace** (ИСП РАН).

РЕЗУЛЬТАТ

В результате фаззинга обнаружено 7 ошибок для PyTorch и одна ошибка для TensorFlow.

В результате анализа исходного кода обнаружено 13 ошибок в PyTorch и 8 ошибок в TensorFlow.

Принятые сообществом исправления



Ошибки, найденные Svace

PyTorch:

5 поданы и приняты

6 поданы и рассматриваются

TensorFlow:

6 поданы и рассматриваются

Ошибки, найденные фаззингом:

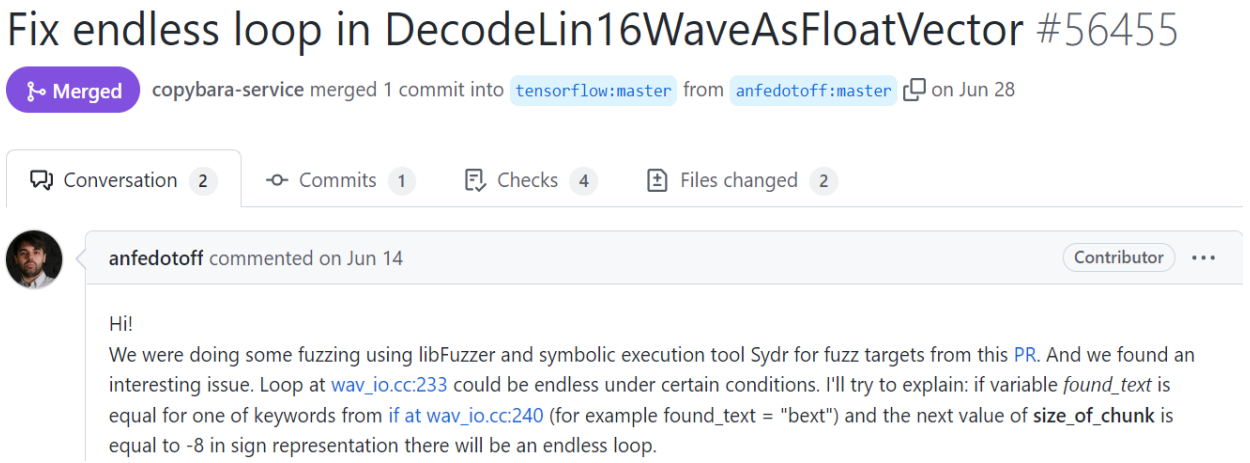
PyTorch:

5 поданы и приняты

3 поданы и рассматриваются

TensorFlow:

1 подана и принята



Глобальная модель долгосрочного устойчивого развития

ИСП РАН

Глобальный вызов:
Долгосрочное устойчивое развитие доверенного системного ПО

Глобальная цель:
Технологическая независимость для всех

ПЛЮСЫ:

✓**Необходимый уровень доверия** без потери конкурентоспособности (эффективности и продуктивности ПО)

✓**Открытое академическое сообщество** квалифицированных экспертов

✓**Полный контроль** над кодовой базой без каких-либо ограничений

МИНУСЫ

~~Технологические риски~~
~~Политические риски~~
~~Нехватка ресурсов~~

Спасибо за внимание!

Арутюн Аветисян

arut@ispras.ru

ИСП РАН