РАЗРАБОТКА ДОВЕРЕННЫХ СИСТЕМ. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Арутюн Аветисян
академик РАН
директор ИСП РАН
arut@ispras.ru

ИСП РАН

ОСНОВА ИСП РАН – ПОСТОЯННО РАЗВИВАЮЩАЯСЯ НАУЧНАЯ ШКОЛА

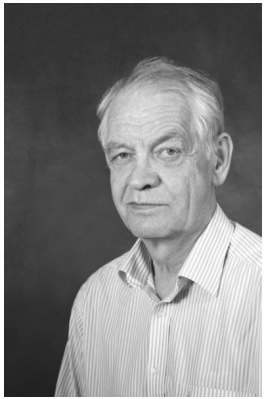
ИСП РАН

С.А. Лебедев

В.А. Мельников

БЭСМ-6 в музее науки Лондона

В.П. Иванников



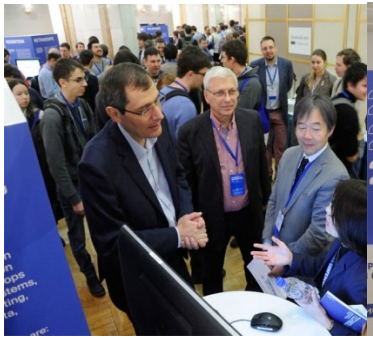
Л.Н. Королёв

«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**



2023: ИСП РАН проведёт COMputer, Software and Application Conference (COMPSAC) в Москве. Ключевая конференция IEEE CS, проводится ежегодно в течении 45 лет.

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, НДВ размыты***

Утечка информации о уязвимостях

Хакеры «для интереса»

- Взломы ради известности
- Ограничены ресурсы
- Только известные эксплойты

Хакеры «для ущерба»

- Вандализм
- Ограничены ресурсы

Преступность

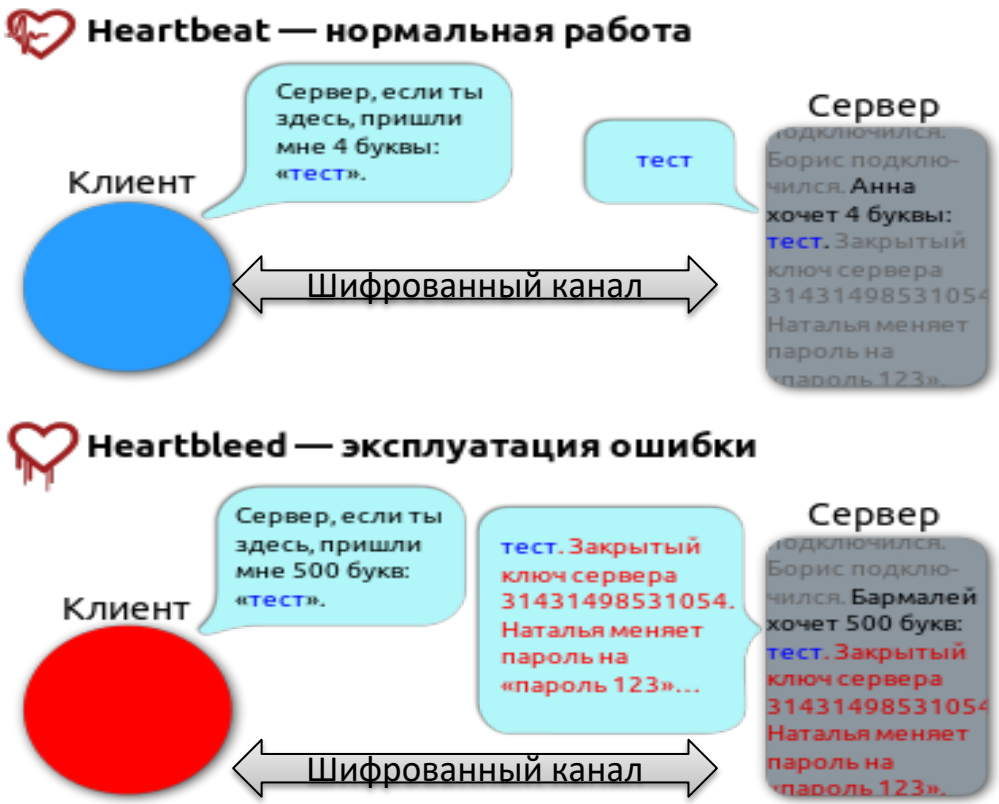
- Взлом ради прибыли
- Существенные ресурсы
- Синдикаты
- Специально разработанные программы для кражи данных

Спецслужбы

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

Рост ресурсов и сложности атак

ПРИМЕР: УЯЗВИМОСТЬ HEARTBLEED (БИБЛИОТЕКА OPENSSL)



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**

- **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
максимальное количество ошибок в исполняемом коде во время **жизненного цикла разработки безопасного ПО**

Обеспечить устойчивость
программной системы, затруднив эксплуатацию существующих ошибок или смягчив последствия их эксплуатации

ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ БЕЗОПАСНОГО 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 БЕЗОПАСНОГО ПО: НЕКОТОРЫЕ ТЕХНОЛОГИИ ИСП РАН

Статический анализ исходного кода	**Svace (ИСП РАН, Россия)** SAMSUNG kaspersky HUAWEI Postgres PROFESSIONAL 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 Postgres PROFESSIONAL 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, 2021)** 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)

ЦЕНТР КОМПЕТЕНЦИЙ ФСТЭК РОССИИ НА БАЗЕ ИСП РАН

ИСП РАН

Центр компетенций по вопросам безопасной разработки и анализа кода сертифицируемого программного обеспечения

- **Участие в разработке стандартов** (ПК 4 «Разработка безопасного ПО» в ТК 362)
- **Создание сообщества:**
  - ❑ **информирование через telegram-каналы**:

    Статика: доверенная разработка https://t.me/sdl_static

    Динамика: доверенная разработка https://t.me/sdl_dynamic (+ страница на GitHub для обмена опытом https://github.com/ispras/TrustedDynamic)

    Оргвопросы: доверенная разработка https://t.me/sdl_community)
  - ❑ **организация круглых столов** (форум «Армия», конференции)
  - ❑ **создание специализированных центров компетенций**
- **Обучение специалистов испытательных лабораторий** (курсы повышения квалификации)

НОВАЯ 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 (2021 г.)

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

*Решение поддержали: заместитель директора ФСТЭК России В.С. Лютиков, руководитель управления перспективных технологий «Лаборатории Касперского» А.П. Духвалов, заместитель гендиректора «РусБИТех» Ю.В. Соснин и др.

С РАЗВИТИЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ОБЛАСТИ КИБЕРБЕЗОПАСНОСТИ ПОЯВИЛИСЬ НОВЫЕ ВЫЗОВЫ

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ: ТОГДА И СЕЙЧАС

ИСП РАН

В 1956 появился термин «искусственный интеллект». Прошло чуть больше 40 лет и…

1997 – IBM Deep Blue выиграл в шахматы у Гарри Каспарова

2002 – первый робот-пылесос

2010 – база данных ImageNet, разметка данных обычными людьми. 14 млн изображений, 20 тысяч категорий

2011 – IBM Watson выиграл шоу Jeopardy! («Своя игра»)

2011 – персональный ассистент в смартфоне (Siri)

2016 – AlphaGO выиграла у профессионального игрока в Го

2016 – Google Translate начинает использовать нейронный машинный перевод для 8 языков

PyTorch

TensorFlow

Сейчас ИИ – это:

- **Управление финансами**
- **Цифровая медицина**
- **Беспилотные автомобили**
- **Роботы (Boston Dynamics и др.)**
- **И МНОГОЕ ДРУГОЕ**

ПРОГНОЗ:
объём глобального рынка ИИ в 2021 составит $327 млрд

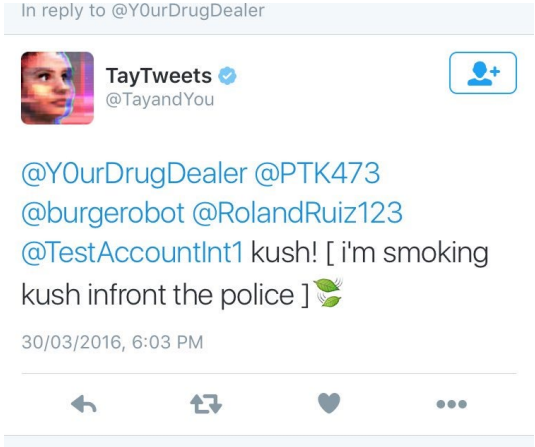
В 2024 превысит $500 млрд

РИСКИ ИСПОЛЬЗОВАНИЯ ИИ В ПРОМЫШЛЕННЫХ СИСТЕМАХ

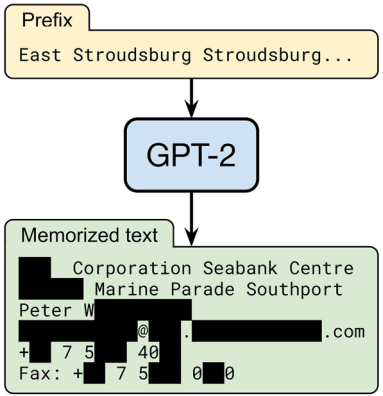
ИСП РАН

Атаки уклонения на системы детекции объектов

ДТП с участием беспилотных автомобилей из-за ошибок ИИ

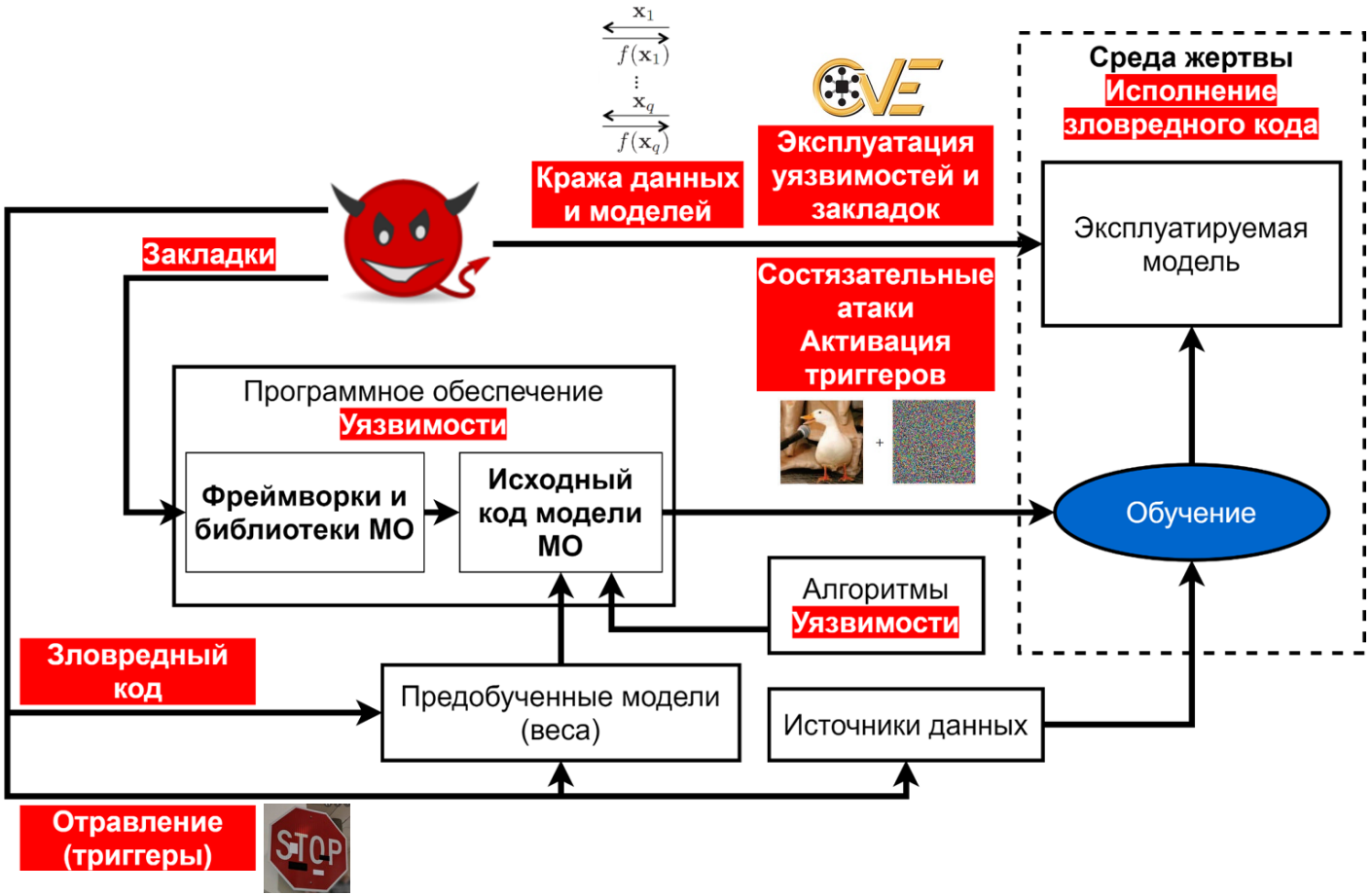


Неконтролируемое поведение дообучаемых чат-ботов



Атаки с извлечением конфиденциальных данных из обученных моделей

АТАКИ НА МОДЕЛИ МАШИННОГО ОБУЧЕНИЯ

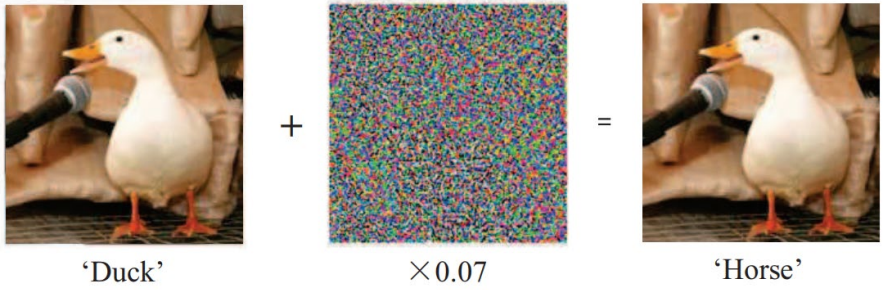


ОТРАВЛЕНИЕ ДАННЫХ И МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

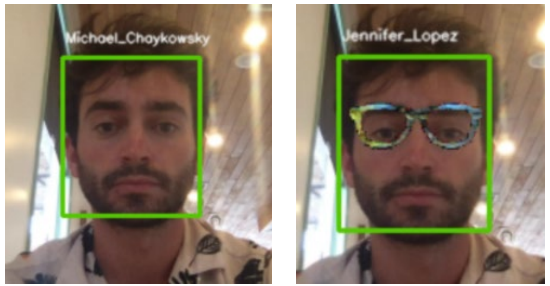
ИСП РАН

- В небольшое количество обучающих примеров добавляется **триггер** – специально подготовленный фрагмент изображения
- В результате обучения на отравленном наборе данных получается **отравленная модель**
- Триггер приводит к **заведомо ошибочному** предсказанию модели на этапе эксплуатации (в том числе к предсказанию **заведомо известного** нарушителю результата)
- Предобученные отравленные модели могут распространяться через Интернет и представлять угрозу при **переносе знаний** (transfer learning)

АТАКИ УКЛОНЕНИЯ (СОСТЯЗАТЕЛЬНЫЕ ПРИМЕРЫ)



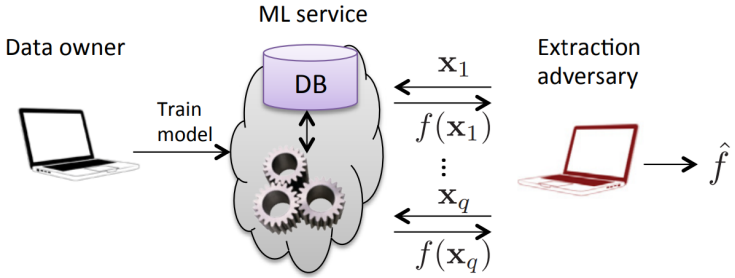
- **Неразличимые** (imperceptible) атаки: l_p-норма отклонения от исходного изображения
  - $p=\infty$ – разрешается изменять каждый пиксель не более чем на ϵ
  - $p=0$ – разрешается изменять произвольным образом не более определенного количества пикселей (известны атаки изменением **лишь одного** пикселя)



- Атаки **белого** ящика – нарушителю доступна полная информация о модели машинного обучения
  - пример – блокировщики рекламы
- Атаки **черного** ящика – нарушителю доступны только предсказания модели на произвольных входных данных (метки либо вероятности классов)
- Атаки часто **универсальны** (могут переноситься на другие модели)
- Методы **защиты** в сценариях белого ящика обходятся **адаптивными** атаками

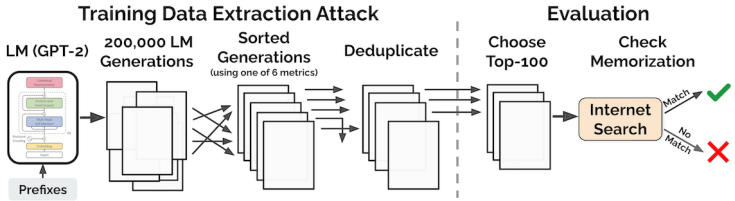
Кража моделей (model extraction)

- Формирование цепочек запросов к модели (сценарий “черного” ящика)
- Использование ответов модели как обучающих данных для собственной модели-заместителя (surrogate model)
- Возможно применение **активного обучения**
  - запросы наиболее “сложных” примеров (расположенных близко к границе классов согласно распределению данных)
- Развиваются методы **защиты** от кражи моделей
  - обнаружение подозрительных цепочек запросов, в том числе активного обучения

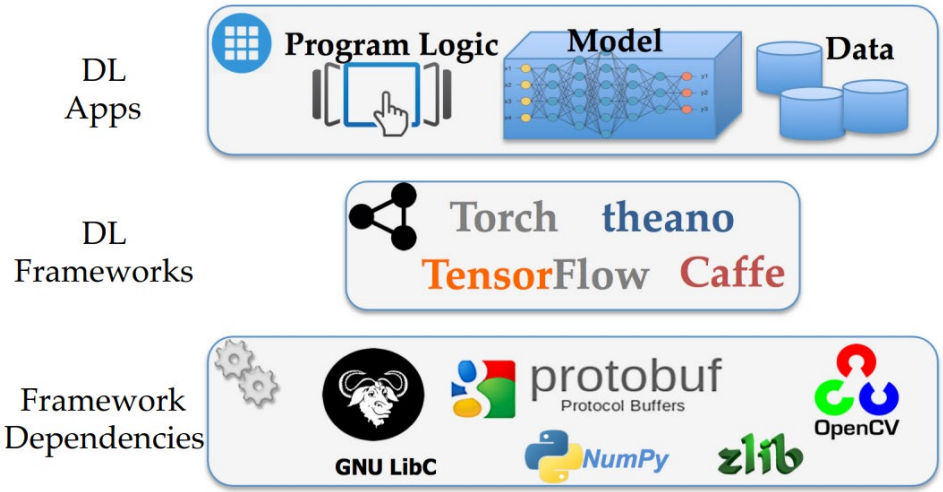


Кража данных

- Конфиденциальные данные как результат генерации ответа моделью
- Атака определения принадлежности обучающей выборке (membership inference)



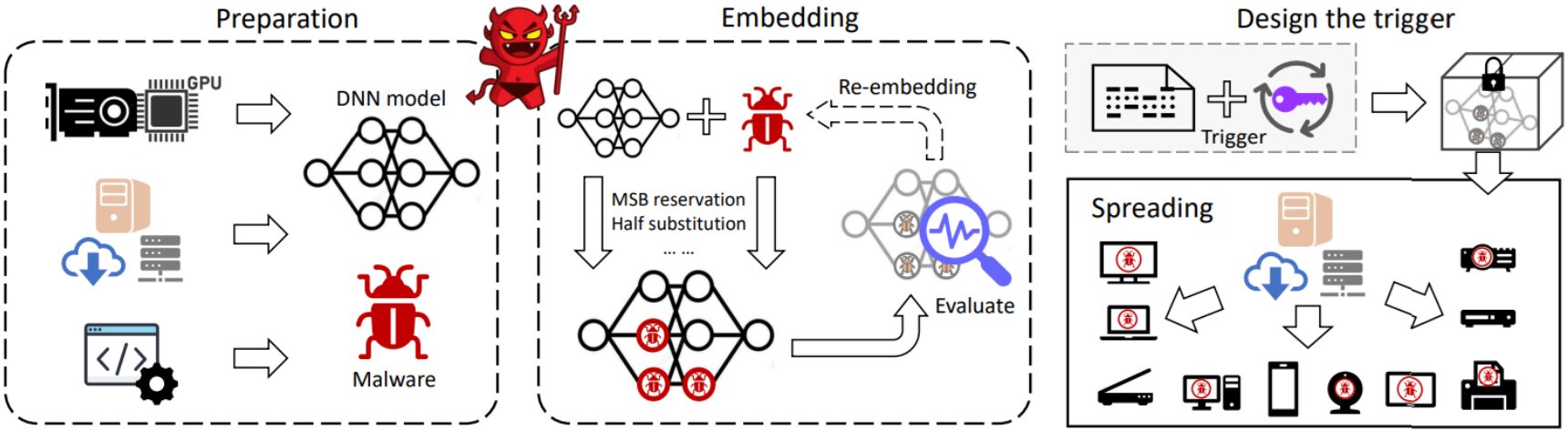
УЯЗВИМОСТИ В ИСХОДНОМ КОДЕ ФРЕЙМВОРКОВ МО



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

- Фреймворк машинного обучения TensorFlow содержит около **3 млн строк** кода и несколько десятков библиотек-зависимостей (NumPy и др.)
- 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**

ВСТРАИВАНИЕ ЗЛОВРЕДНОГО КОДА В МОДЕЛИ МО



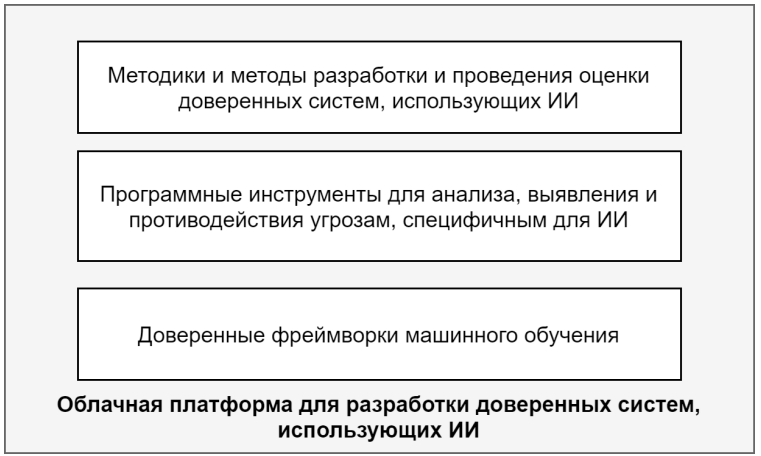
Wang et al. (2021) EvilModel 2.0: Hiding Malware Inside of Neural Network Models

- Встраивание зловредного кода размером **до нескольких мегабайт** в вещественные параметры (веса) нейросетевых моделей **без существенной потери их точности**
- **Не обнаруживается** антивирусным программным обеспечением
- **Компрометация** устройства жертвы при использовании предобученных моделей с зловредным кодом, распространяемых через Интернет (GitHub и другие ресурсы)

Многоуровневость и многообразие угроз, специфичных для ИИ

- Угрозы возникают на **разных уровнях**:
  - уязвимости и закладки в **фреймворках** машинного обучения
  - **данные** (отравление, конфиденциальность)
  - **алгоритмы** (неустойчивость нейросетевых моделей)
- С некоторыми угрозами (отравление) требуется бороться **сразу на нескольких этапах** жизненного цикла систем с ИИ
- Угрозы **отличаются** в разных прикладных системах с ИИ, количество которых **возрастает**

- **Требуется комплексное решение** – Платформа, включающая в себя **методологию** создания доверенных систем с ИИ:
  - модели угроз (нарушителя)
  - критерии и методики оценки доверия (бенчмарки)



МЕТОДОЛОГИЯ: ТРЕБОВАНИЯ К ЖИЗНЕННОМУ ЦИКЛУ СИСТЕМ С ИИ

ИСП РАН

Анализ и проектирование	Разработка	Тестирование	Эксплуатация
Согласование модели нарушителя с заказчиком. Доступ к среде функционирования, моделям и наборам данных **Формирование требований к устойчивости к атакам** уклонения, отравления и извлечения информации	**Методы противодействия атакам на модели машинного обучения.** Методы робастного обучения моделей. Противодействие атакам с помощью «объяснения» результатов моделей	**Технологическая тестовая база оценки безопасности СИИ** (аналог инженерно-криптографического анализа системы). Измерение устойчивости разработанной системы с ИИ к атакам на модели машинного обучения, входящие в систему	**Экспертиза обучающих выборок и моделей**. Обнаружение атак и аномалий в данных, мониторинг дообучаемых моделей

ЦЕНТР ДОВЕРЕННОГО ИИ ИСП РАН

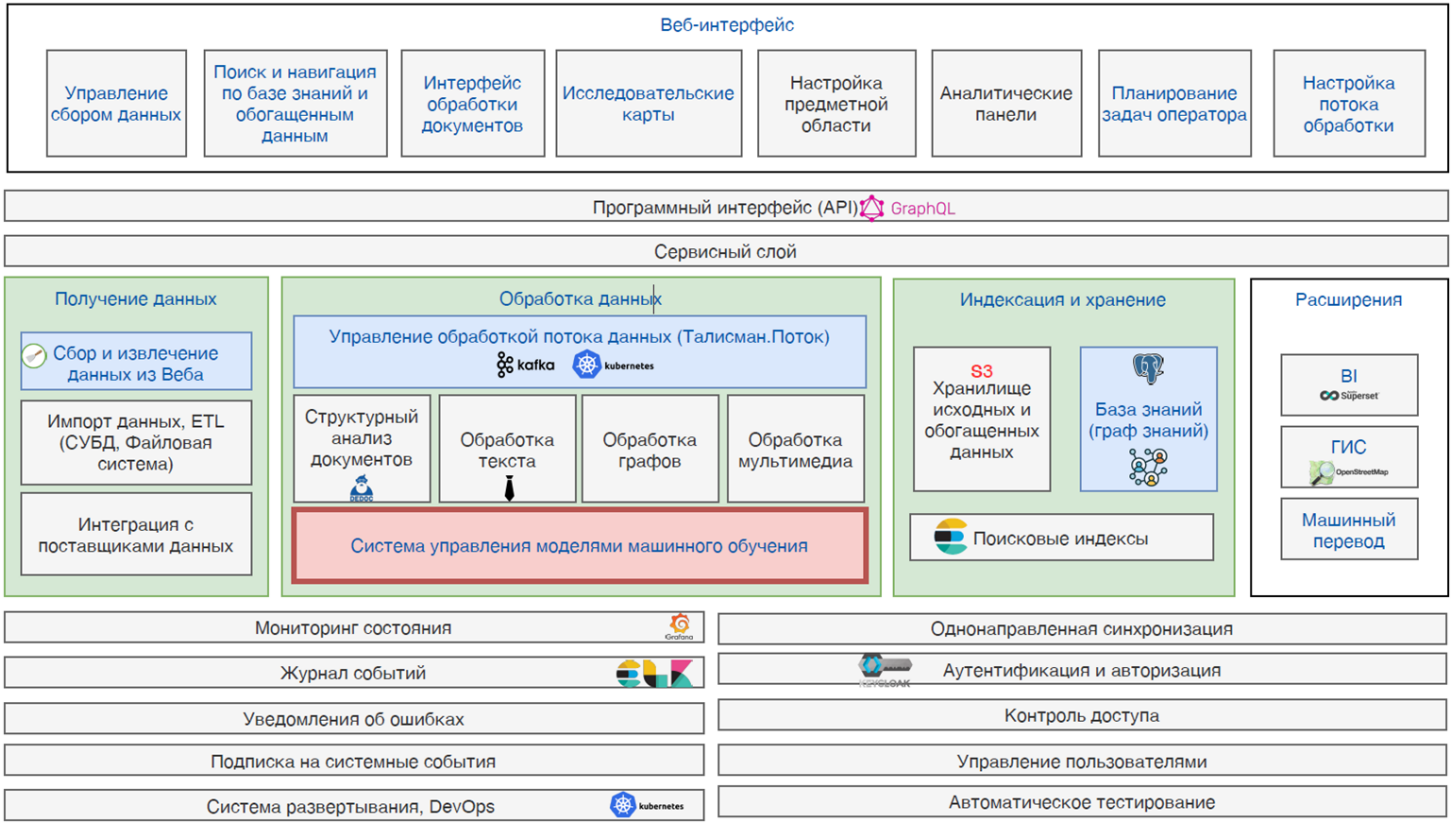
Программа Центра — создание методик и соответствующих программных и аппаратно-программных платформ для разработки и верификации технологий ИИ с требуемым уровнем **доверия**

Классификация угроз и **разработка программных инструментов** для анализа, выявления и противодействия угрозам, специфичным для ТИИ: - состязательные атаки на модели - атаки с внедрением закладок и зловредного кода - кража моделей и данных **Повышение интерпретируемости** моделей	Создание **методик и бенчмарков** на основе **реальных** приложений: - Медицина - Социология - Информационная безопасность	Создание **доверенных сред разработки** моделей машинного обучения	Создание **отчуждаемой облачной платформы** для разработки доверенных систем, использующих ТИИ

Ключевые направления Программы

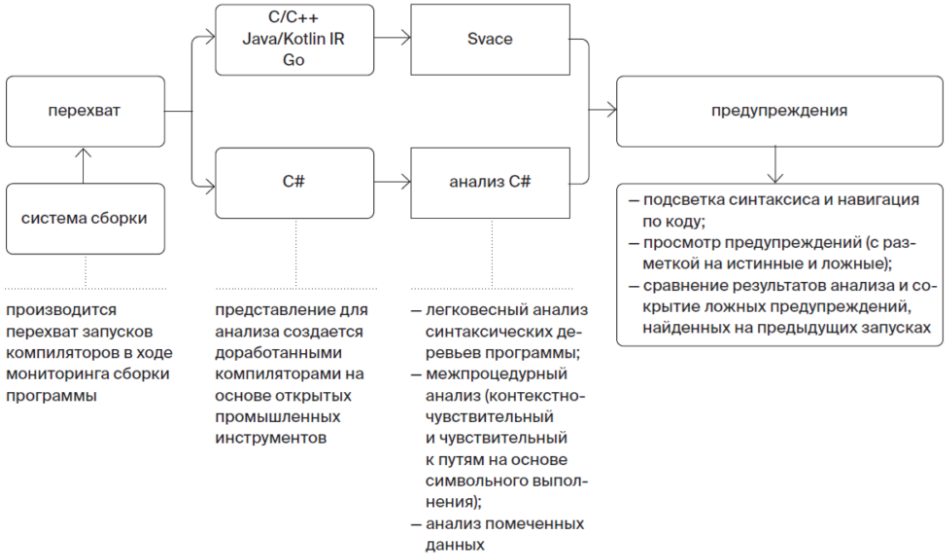
- Реализуемость Программы основывается на промышленных технологиях ИСП РАН (Talisman, Асперитас и др.), долгосрочных партнерских отношениях с индустрией и академическим сообществом
- Выполнен ряд НИР (в том числе с Академией криптографии РФ), долгосрочный контракт с Samsung Electronics по тематике доверенного ИИ (интерпретируемость моделей МО)

Платформа Talisman



Облачная среда Asperitas

Статический анализатор Svace



ДОВЕРЕННЫЙ ИИ В МИРОВОМ НАУЧНОМ СООБЩЕСТВЕ

ИСП РАН

Проблемы создания доверенных систем, использующих ИИ, уже активно рассматриваются в мировом научном сообществе

- 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

источник: Nicholas Carlini (Google Brain)

Количество публикаций, посвященных атакам на системы с ИИ, превысило 3000

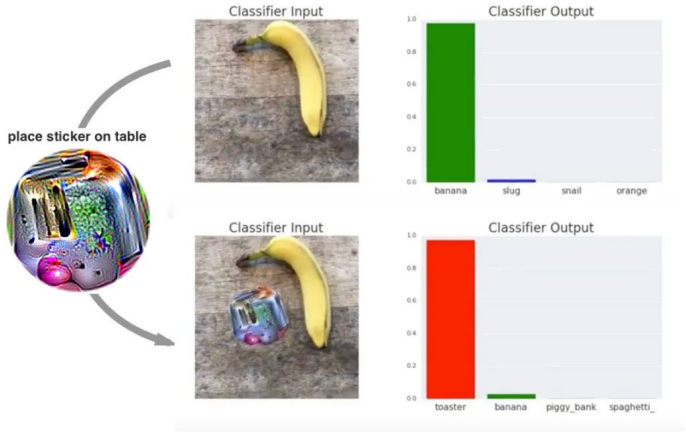
- Более того, остается **открытым и требующим исследований** вопрос применимости методов защиты от угроз ИИ к **реальным системам***
- Без этих исследований **невозможно** говорить о долгосрочном развитии технологий ИИ в целом и их массовом внедрении в индустрию

* Florian Tramèr “Does Adversarial Machine Learning Research Matter?” KDD 2021 Workshop on Adversarial Machine Learning

ОТКРЫТЫЕ НАПРАВЛЕНИЯ ИССЛЕДОВАНИЙ В ДОВЕРЕННОМ ИИ

ИСП РАН

Новые модели нарушителя



Brown, T.B., Mané, D., Roy, A., Abadi, M., & Gilmer, J. (2017). Adversarial Patch. ArXiv, abs/1712.09665.

Пример для задачи классификации изображений:

- наиболее исследованная модель нарушителя в атаках **состязательными примерами** – l_p-норма разности между изображениями – **редко полезна на практике**
- требуются исследование и разработка **новых, реалистичных** моделей нарушителя, а также методов противодействия, в том числе основанных на **интерпретируемости** моделей МО
- на практике задача классификации изображений служит **лишь вспомогательной;** методы защиты от угроз в реальных задачах (детекция объектов и др.) **отличаются и исследованы в меньшей степени**

Вывод: методы и технологии создания систем доверенного ИИ не могут разрабатываться:

- в отрыве от **индустриальных партнеров** и их прикладных задач
- без создания **сообщества** ученых по этой тематике

СООБЩЕСТВО УЧЕНЫХ И ЭКОСИСТЕМА ПОТРЕБИТЕЛЕЙ

Экосистема доверенного ИИ в модели треугольника «исследования и разработка – образование – инновации»

Индустриальные партнеры проводят:

- подготовку и передачу **наборов данных**
- **тестирование** доверенных фреймворков МО
- **опытную эксплуатацию**

[1] ИСП РАН – Центр компетенций по вопросам безопасной разработки и анализа кода сертифицируемого ПО (ФСТЭК России) и Технологический центр исследований безопасности ядра Linux

[2] АО «ЦНИИмаш» заинтересовано в разработке прикладных систем, использующих технологии ИИ, для корпорации Роскосмос

ПАРТНЕРЫ ЦЕНТРА

ИСП РАН

АО «Лаборатория Касперского»

- Жизненный цикл разработки ПО с заданным уровнем доверия (Svace*, Crusher* и др.) **без учета ИИ**
- Инфраструктура разработки доверенного ПО, **использующего ИИ**

ЗАО «ЕС-Лизинг»

- Интеллектуальный анализ данных в медицине, социальных медиа, банковском секторе (Talisman*)
- Доверенные прикладные системы, **использующие ИИ**, для банковского сектора

«ТЕХНОПРОМ»

- Внедрение в государственных организациях ПАК (Асперитас*, Fanlight* и др.)
- Разработка прикладных систем с заданным уровнем доверия, **использующих ИИ**

ООО «Интерпроком»

- Совместная* реализация ведомственной программы цифровой трансформации МИД России
- Аналитические системы поддержки принятия решений **с использованием ИИ** (в частности ПО Talisman*)

АО «ЦНИИмаш» доверенные прикладные системы, **использующие ИИ**, для корпорации Роскосмос

* Продукты разработаны ИСП РАН и внесены в единый реестр российских программ для электронных вычислительных машин и баз данных

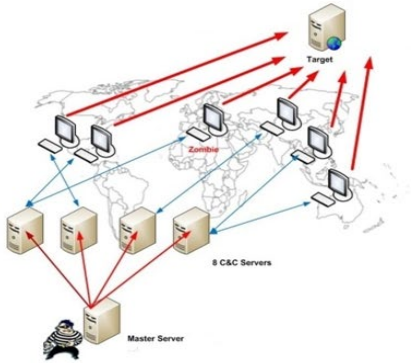
Спасибо!
Вопросы?

ИСП РАН

УЯЗВИМОСТИ: ПРИЧИНЫ И СЛЕДСТВИЯ

Аварии на критических обьектах

Перехват управления



Бот-неты

Уязвимости

Кража паролей

Кража информации о кредитных картах

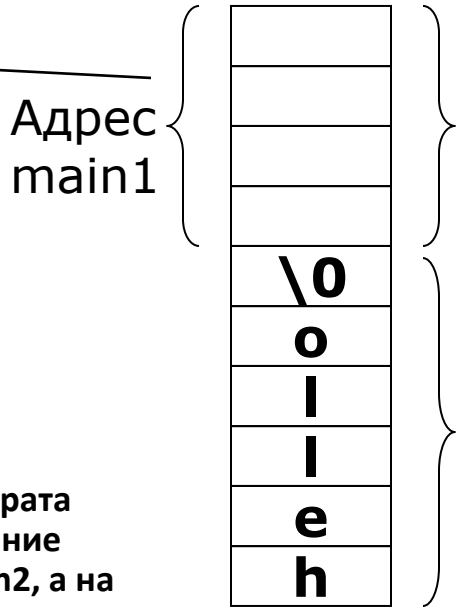
Компьютерные атаки осуществляются путём эксплуатации дефектов в ПО и аппаратуре.

ПРИМЕР: ПЕРЕПОЛНЕНИЕ БУФЕРА (УРОВЕНЬ ИСХОДНОГО КОДА)

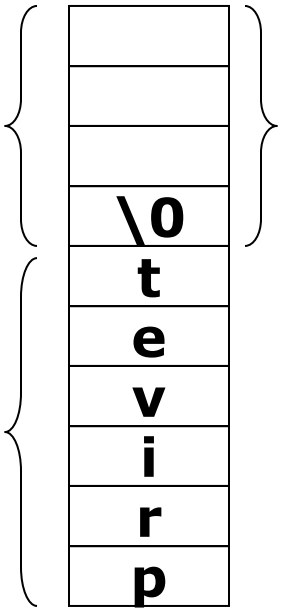
ИСП РАН

```
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: 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                                  EAT, buffer, 3 + payload + paddi
                Go to declaration
                Go to definition
                Show usage
```

De facto стандарт безопасной разработки

Определены сквозные действия по обеспечению безопасности программы на каждом этапе разработки

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: 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

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

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