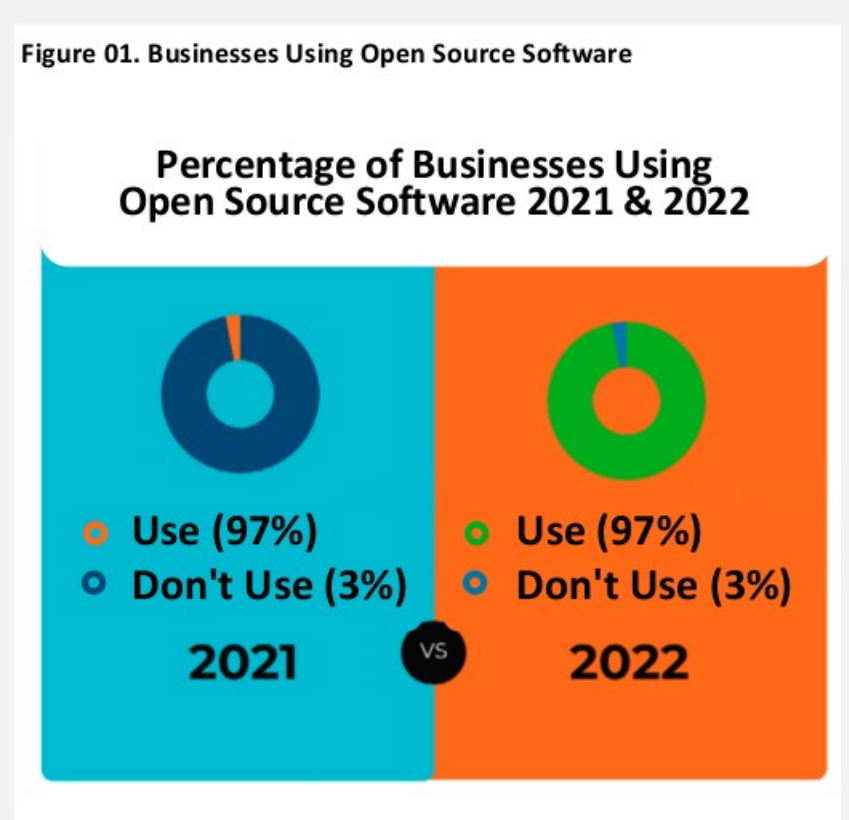
Международный ТБ ФОРУМ Технологии Безопасности

Технологический центр исследования безопасности ядра Linux и критических компонентов

Алексей Хорошилов
khoroshilov@ispras.ru

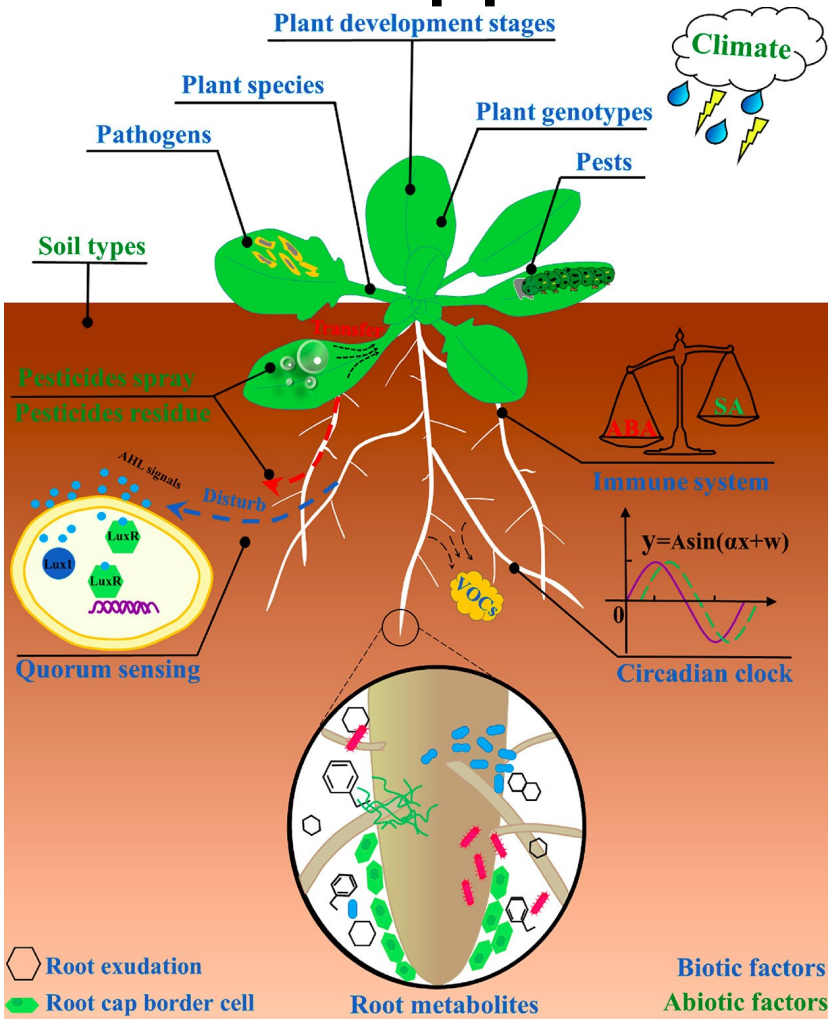
ИСП РАН
Институт системного программирования им. В.П. Иванникова
Российской академии наук

Заимствованные компоненты с открытым исходным кодом



State of Open: The UK in 2022 “Phase One: The Open Source Journey”
7 July 2022

Заимствованные компоненты с открытым исходным кодом

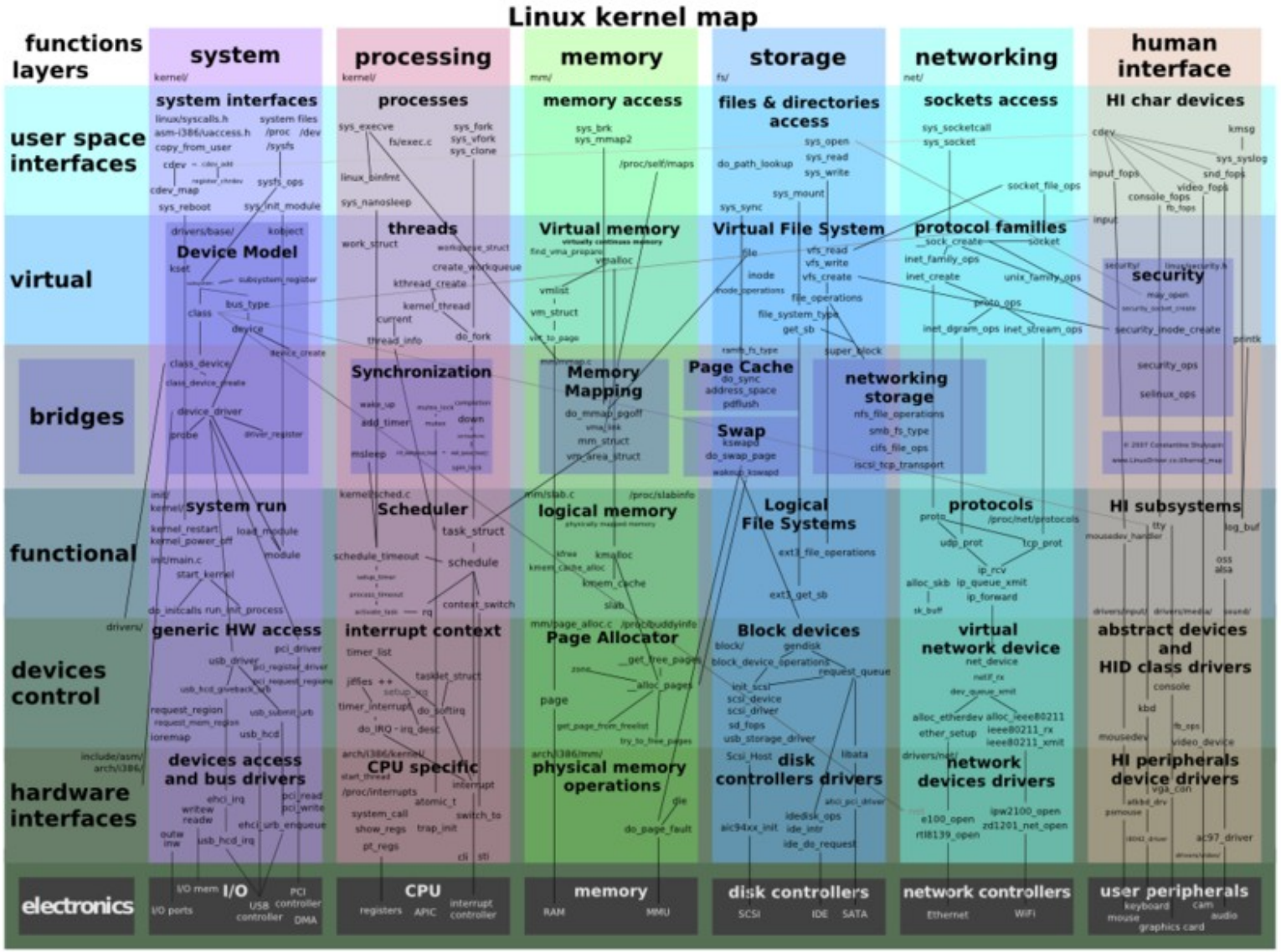


Заимствованные компоненты с открытым исходным кодом

- исследование в соответствии с Методикой ВУ и НДВ
- поддержание безопасности СЗИ в соответствии с требованиями 76-го приказа

Карта ядра Linux

- 33 млн. строк кода



https://ru.wikipedia.org/wiki/Файл:Linux_kernel_map.png

Статический анализ ядра Linux

- Задача
  - применить инструмент статического анализа
  - разметить все предупреждения высокого уровня критичности
    - подтверждённые - исправить
    - ложные и не требующие исправления – написать обоснование
  - лаборатории: провести выборочную проверку

Статический анализ ядра Linux

Предупреждения SVACE 3.3	
Критичные	**1823**
Важные	**12313**
Средние	**6205**
Низкие	**12456**
Всего	**32797**

- 14,1 тыс. высокой степени критичности
- 5-6 человеко-лет

Технологический центр исследования безопасности ядра Linux

- создан ФСТЭК России на базе ИСП РАН
- 22 партнёра:
  - 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»
  - АО «НППКТ»
  - 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»

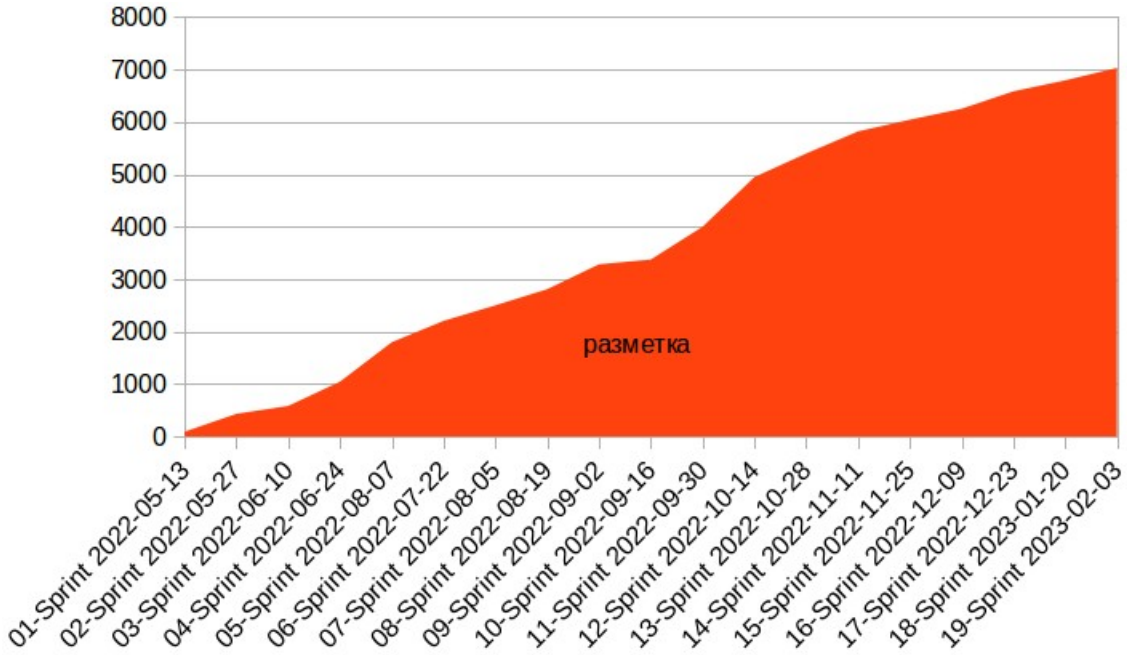
Технологический центр исследования безопасности ядра Linux

- (1) Ведётся сопровождение ветки ядра, основанной на стабильной версии 5.10
- (2) Подготовлены методики проведения исследования ядра Linux, 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
- (3) Создана экспертная группа, состоящая из представителей 22 компаний, в рамках которой
  - формируются принципы функционирования Технологического центра
  - проведена разметка более 7 тыс. предупреждений инструмента статического анализа SVACE
  - подготовлен ряд исправлений ошибок в ядре, 105 из которых уже были приняты в основную ветку ядра
- (4) Готовятся рекомендации по конфигурированию ядра с целью повышения его безопасности
- (5) Ведётся доработка исправлений, нацеленных на повышение безопасности работы ядра, на стадии его развёртывания и инициализации

Статический анализ: статистика

14 февраля 2023

	Всего	Обработано	Подтверждено	Не требует исправления	Ложные срабатывания
Критичные	**1823**	1821	294	380	1147
Важные	**12313**	4468	555	1536	2377
Средние	**6205**	175	38	68	69
Низкие	**12456**	568	116	290	162
Всего	**32797**	**7032**	**1003**	**2274**	**3755**



Статический анализ: кросс-верификация

14 февраля 2023

	Всего	Won't Fixed				False Positive			
		Без вериф.	Обсуждается	Подтверждено	Всего	Без вериф.	Обсуждается	Подтверждено	Всего
Критичные	**1823**	40	8	332	**380**	82	42	1023	**1147**
Важные	**12313**	951	16	569	**1536**	1884	14	479	**2377**
Средние	**6205**	61	-	7	**68**	68	-	1	**69**
Низкие	**12456**	288	-	2	**290**	162	-	-	**162**
Всего	**32797**	1340	24	910	**2274**	2196	56	1503	**3755**

OVERFLOW_AFTER_CHECK.EX: drivers/iio/gyro/bmg160_core.c:224

- **Review status:** Confirmed -> Won't fix

24 June 2022, 16:16 (anton.fadeev)

Переполнение возникнет, нужно менять условие на
for (i = 0; i < ARRAY_SIZE(bmg160_samp_freq_table) - 1; ++i)

31 January 2023, 17:30 (karpov)

Review: не согласен, Won'tFix
Автор явно уверен что значение filter найдется в таблице `bmg160_samp_freq_table[]`
Тогда break сохранит i с допустимым значением индекса.
Было бы полезно поисследовать - с какими значениями val вызывается bmg160_set_filter(), т.к. bmg160_samp_freq_table - статический массив со значениями filter: 32, 64, 12, 23, 47, 116, 230

Сокращать перебор массива уменьшая итерации на 1 нельзя, т.к. это лишит функцию возможности найти значение в последнем элементе.

3 February 2023, 14:47 (anton.fadeev)

Согласен. Чип инициализируется значениями {100, 32, 0x07} . Значения в bmg160_set_bw проверяются. Смутило, что в datasheet на чип возможных значений регистра 0x10(BW) 8. Но судя по коду значение 0 туда никогда не попадёт.

Статический анализ: исправления

14 февраля 2023

	Всего	Подтверждено				
		В работе	Сообщено	Исправлено	в 5.10	Всего
Критичные	**1823**	240	24	23	7	**294**
Важные	**12313**	459	28	56	12	**555**
Средние	**6205**	30	-	8	-	**38**
Низкие	**12456**	44	2	70	-	**116**
Всего	**32797**	773	54	157	19	**1003**

Подготовка исправлений

Компания	Количество принятых патчей
ООО «Базальт СПО»	**1**
ООО «БЕЛЛСОФТ»	**7**
АО «ИнфоТеКС»	**3**
ИСП РАН	**44**
ООО «Открытая мобильная платформа»	**24**
АО «РАСУ»	**3**
ООО «РЕД СОФТ»	**3**
ООО «РусБИТех-Астра»	**6**
АО МВП «Свемел»	**1**
ООО «Фактор-ТС»	**4**
ООО «ФИНТЕХ»	**4**
ООО «ЯНДЕКС.ОБЛАКО»	**5**
Всего:	**105**

Статический анализ ядра Linux

- Задача
  - применить инструмент статического анализа
  - разметить все предупреждения высокого уровня критичности
    - подтверждённые - исправить
    - ложные и не требующие исправления – написать обоснование
  - лаборатории: провести выборочную проверку

Фаззинг-тестирование ядра Linux

- Задача
  - выполнить фаззинг-тестирование ядра
    - в продуктовой конфигурации, адаптированной под фаззинг-тестирование
    - с задействованием поверхности атаки, соответствующей модели угроз продукта
  - исправить выявленные падения
    - как минимум воспроизводимые
  - добиться покрытия кода ядра, не хуже чем обеспечивают актуальные методики

Фаззинг-тестирование ядра Linux

- Технологический центр
  - подготовлены и опубликованы методики фаззинг-тестирования при помощи syzkaller
    - публично доступны на портале*
    - у 8+ партнёров развёрнуто фаззинг-тестирование на собственных ядрах
  - проведён вебинар по разбору срабатываний
    - запись доступна на портале*
    - 16 исправлений / 25 бэкпортирование
  - формируются требования к покрытию

* https://portal.linuxtesting.ru

Поддержание безопасности СЗИ

- Задача
  - отслеживать уязвимости, выявляемые в заимствованных компонентах
  - своевременно передавать пользователям и в БДУ ФСТЭК России информацию об актуальных уязвимостях и компенсирующих мерах
  - выпускать обновления, устраняющие уязвимости
  - проводить испытания обновлений

Критические компоненты

- Задачи
  - те же, но для большого числа компонентов

Addressing Cybersecurity Challenges in Open Source Software, Linux Foundation 2022

Критические компоненты

- .Net6

  Статический анализ: 869 из 18500 (750 из 5300)

  - АО «Аладдин Р.Д.»
  - ООО «Гарда Технологии»
  - ООО «НПЦ «КСБ»
  - АО «Лаборатория Касперского»
  - ООО НТЦ «Фобос НТ»
- OpenSSL

  Статический анализ: 87 из 305 (913)

  - АО «Аладдин Р.Д.»
  - АО «ИнфоТеКС»

Критические компоненты

- EDK2 (UEFI)
  - ИСП РАН
  - ООО «Новые платформы»
- Node.js
  - АО «Лаборатория Касперского»
  - ООО «Р-Вижн»
  - ООО НТЦ «Фобос НТ»

Фаззинг:
драйвера NTFS, EXT4
14 исправлений принято

Заключение

Что делать?

- Разрабатывая свои продукты
  - Определите заимствованные компоненты, находящиеся на поверхности атаки
  - Оцените испытания каких компонентов предпочтительно вести самостоятельно, а каких совместно
    - Требования к минимальному качеству испытаний формируются экспертным сообществом
  - Присоединяйтесь к сообществу!

LinuxTesting.org

Спасибо!

Алексей Хорошилов
khoroshilov@ispras.ru
https://portal.linuxtesting.ru/

ИСП РАН
Институт системного программирования им. В.П. Иванникова РАН