УДК 004.056 doi: 10.15827/0236-235X.142.432-441 2023. Т. 36. № 3. С. 432–441

Контроль подключений USB-устройств в локальной вычислительной сети компьютеров под управлением Astra Linux SE

А.В. Баранов
П.М. Корепанов
И.А. Лепешев

Ссылка для цитирования
Баранов А.В., Корепанов П.М., Лепешев И.А. Контроль подключений USB-устройств в локальной вычислительной сети компьютеров под управлением Astra Linux SE // Программные продукты и системы. 2023. Т. 36. № 3. С. 432–441. doi: 10.15827/0236-235X.142.432-441
Информация о статье
Поступила в редакцию: 19.06.2023 После доработки: 28.06.2023 Принята к публикации: 03.07.2023

Аннотация. Рост требований к информационной безопасности, а также тенденция к импортозамещению в области системного ПО обусловили широкое распространение инфраструктурных решений, построенных на базе отечественной операционной системы Astra Linux Special Edition (Astra Linux SE). Применение Astra Linux SE позволяет строить защищенные программно-аппаратные системы для обработки информации ограниченного доступа, в том числе в научных суперкомпьютерных центрах. При этом одним из важнейших аспектов обеспечения информационной безопасности является контроль подключения USB-устройств к компьютерам в локальной вычислительной сети. Анализ доступных современных источников показывает, что готовых комплексных решений, работающих в среде Astra Linux SE, в настоящее время не существует. В статье рассмотрен возможный технологический стек подобного решения, включающий, помимо Astra Linux SE, систему организации очередей сообщений RabbitMQ, микрофреймфорк для разработки web-приложений Flask, СУБД PostgreSQL, а также средство выявления подключений USB-устройств USBRip. Рассмотрена предложенная на базе технологического стека модульная структура программной системы аудита подключений USB-устройств, включающая модули сбора информации о USB-подключениях на контролируемых компьютерах, модуль агрегации собранной информации на серверной стороне и модуль проверки легитимности выявленных подключений USB-устройств к контролируемым компьютерам под управлением Astra Linux SE. Предложенные структура и технологический стек реализованы в виде макета программной системы, получившей название ALUMNUS. Макет был развернут и прошел опытную эксплуатацию в защищенном сегменте суперкомпьютера МВС-10П ОП, установленном в Межведомственном суперкомпьютерном центре РАН.

Ключевые слова: суперкомпьютерный центр, информационная безопасность, Astra Linux SE, контроль USB-подключений
Благодарности. Работа выполнена в МСЦ РАН в рамках государственного задания по теме FNEF-2022-0016

Информационная безопасность научных суперкомпьютерных центров долгое время во многом обеспечивалась за счет механизмов защиты, встроенных в *операционные системы* (ОС) вычислительных узлов и сетевые устройства суперкомпьютеров, а также организационно-технических мер, базирующихся на этих механизмах. В связи с постоянным ростом компьютерных угроз и повышением требований к информационной безопасности вычислительно-информационная инфраструктура все чаще строится на базе системного ПО, сертифицированного по требованиям безопасности информации. Широкое распространение в последние годы получили инфраструктурные решения на базе отечественной ОС Astra Linux [1], которая имеет множество модификаций, в том числе *Astra Linux Special Edition* (Astra Linux SE).

Astra Linux SE представляет собой ОС для обработки информации ограниченного доступа, в том числе содержащей сведения, составляющие государственную тайну. Система обладает широкими возможностями [2, 3] по защите информации, позволяющими создавать на своей основе автоматизированные и информационные системы в защищенном исполнении. Важную роль Astra Linux играет в процессе импортозамещения системного ПО. Как отмечается в работе [4], важнейшими отличительными особенностями разработки и обеспечения доверия к ОС Astra Linux SE являются реализация в ней собственной подсистемы безопасности PARSEC и отказ от применения аналогичных заимствованных иностранных механизмов, в том числе SELinux [5] и AppArmor [6]. Подчеркивается, что Astra Linux SE – единственное отечественное решение в классе ОС общего назначения, получившее сертификаты соответствия по самым высоким классам защиты в системах сертификации ФСТЭК, ФСБ и Минобороны России [7]. При этом Astra Linux SE – это единая платформа для всех типов устройств [1] на базе процессоров x86-64, ARM, «Эльбрус», «Байкал», «Комдив», совместимая с большин-

ством аппаратного оборудования, в том числе отечественного производства. ОС широко применяется для обеспечения безопасности информации, а также для защиты объектов критической информационной инфраструктуры.

Важными аспектами обеспечения информационной безопасности являются контроль и учет подключаемых внешних USB-устройств в защищенном сегменте *локальной вычислительной сети* (ЛВС). Базовые средства Astra Linux SE не позволяют вести централизованный учет подключаемых USB-устройств к компьютерам в ЛВС. Впервые эта проблема и методы ее решения были рассмотрены в работе [8], в которой справедливо отмечается, что угроза использования неучтенных USB-устройств может привести к снижению уровня информационной безопасности и образованию канала утечки информации в защищенном сегменте ЛВС. В работах [8, 9] рассматриваются следующие методы учета подключений USB-устройств в среде Astra Linux:

– ручная настройка учета подключений USB-устройств, а именно генерация определенных правил для каждого устройства, согласно которым ОС разрешает или не разрешает использование того или иного устройства;

– использование встроенной в Astra Linux SE графической утилиты fly-admin-smc;

– использование службы Astra Linux Directory (ALD) [10].

Первые два метода требуют ручной работы администратора защищенного сегмента, что при наличии большого числа контролируемых компьютеров обусловливает значительную трудоемкость и высокую вероятность ошибок [8]. В основу домена ALD [3] положен принцип объединения логически связанных сетевых ресурсов и учетных записей пользователей в единую систему авторизации с централизованным управлением и мандатным разграничением доступа пользователей к информации. Использование службы ALD требует включения в его домен всех компьютеров, на которых предполагается вести аудит подключения USB-устройств, что далеко не всегда можно осуществить на практике.

В публикации [11] в качестве базового метода обнаружения подключения устройств USB предлагается метод анализа системных журналов контролируемого компьютера. Он рассмотрен в самых общих чертах, применим для контроля отдельного компьютера или автоматизированного рабочего места. Способы и средства применения метода анализа системных журналов на множестве контролируемых компьютеров в ЛВС не рассматриваются.

Среди существующих средств учета подключений USB-устройств в среде Astra Linux следует отметить программы для ЭВМ [12, 13]. Программа, рассмотренная в [12], является результатом исследований [8] и предназначена для автоматизации процесса учета и контроля использования машинных носителей информации на средствах вычислительной техники. В ее состав входят подсистемы учета и контроля использования машинных носителей информации на средствах вычислительной техники в организации. Подсистема контроля функционально разделяется на агента, непрерывно функционирующего на контролируемых компьютерах, и менеджера, принимающего данные от агентов и сигнализирующего администратору об использовании неучтенных носителей. В программе, предложенной в [13], реализован аудит подключения USB-устройств к отдельному персональному компьютеру под управлением Astra Linux, не включенному в состав ЛВС. Обе программы [12, 13] отсутствуют в свободном доступе, информация о них, полученная из рефератов свидетельств о государственной регистрации программ для ЭВМ, не содержит сведений о способах и методиках построения, структуре и стеке технологических решений.

Задача контроля подключений устройств USB к компьютерам в ЛВС может быть решена за счет применения систем управления событиями информационной безопасности (систем SIEM – Security Information and Event Management [14]). Подобные инструменты собирают информацию из системных журналов контролируемых компьютеров и анализируют ее на предмет выявления событий безопасности, связанных с реализацией тех или иных угроз. Настроив соответствующим образом SIEM-систему, можно организовать в том числе контроль USB-подключений. Как и в случае с ALD, установка, применение и, что немаловажно, сопровождение SIEM-системы требуют существенных материальных и трудовых затрат, которые та или иная организация может понести далеко не всегда.

Таким образом, анализ публикаций в открытом доступе показывает, что существующие решения задачи контроля подключения USB-устройств к компьютерам под управлением Astra Linux SE либо носят характер общих рекомендаций, либо не предназначены для приме-

нения в ЛВС, либо требуют обязательного включения контролируемых компьютеров в домен ALD, либо связаны с установкой и настройкой SIEM-системы. Настоящая работа направлена на преодоление указанных недостатков, ее целями являются исследование методов, способов и средств построения программной системы аудита подключений USB-устройств к компьютерам под управлением Astra Linux SE в ЛВС, а также разработка подобной системы.

Методы, способы и средства аудита действий пользователя компьютера под управлением ОС Astra Linux SE

Под аудитом действий пользователя понимают анализ информации о связанных с этими действиями событиях, которые происходят или происходили в ОС. Наиболее распространенным методом аудита действий пользователя в ОС Astra Linux SE является журналирование, предполагающее сбор информации о происходящих системных событиях и сохранение этой информации в специальных структурах данных, как правило, в файлах. Собранная информация анализируется, в том числе средствами SIEM, на предмет выявления событий безопасности, связанных с реализацией тех или иных угроз. Рассмотрим способы журналирования, применяемые в ЛВС контролируемых компьютеров.

Централизованное журналирование – это сбор информации о критичных с точки зрения безопасности системных событиях на одном из компьютеров (серверов) сети. Такой способ позволяет одновременно контролировать большое количество компьютеров и оперативно реагировать на события, представляющие угрозу информационной безопасности. Недостатком централизованного подхода являются накладные расходы на выделенный сервер и организацию служебного сетевого трафика для сбора информации. При децентрализованном журналировании сбор информации о событиях безопасности осуществляется на каждом из узлов ЛВС по отдельности. В этом случае накладные расходы на организацию работы центрального узла отсутствуют, но возникает необходимость анализа журналов событий отдельно на каждом компьютере. На практике это ограничивает число контролируемых машин и увеличивает время реакции системы на события безопасности. При смешанном способе журналирования контролируемые компьютеры подразделяются на подключенные к ЛВС и автономные. Внутри ЛВС организуется централизованное журналирование, автономные компьютеры накапливают информацию о событиях безопасности отдельно и независимо друг от друга.

В ОС семейства Linux, базирующихся на дистрибутиве Debian, существует стандартная система аудита, реализуемая службой rsyslog. Служба позволяет как вести журналирование на локальной машине, так и отправлять журналы событий на специальный сервер [15]. После обновления x.7 служба rsyslog не соответствует поддерживаемым сценариям эксплуатации Astra Linux SE (см. справочный центр Astra Linux, https://wiki.astralinux.ru/pages/viewpage.action?pageId=9011231), и разработчики Astra Linux рекомендуют использовать в качестве средства журналирования службу syslog-ng [16]. Эта служба предоставляет достаточно широкий функционал по сравнению со стандартными средствами журналирования Linux, однако для решения задачи выявления подключений USB-устройств требует дополнительных инструментов.

Как уже отмечалось, в Astra Linux SE существует собственная подсистема аудита, реализуемая подсистемой безопасности PARSEC [2]. Информация о системных событиях записывается в файлы kernel, mlog и user.mlog, по умолчанию размещаемые в каталоге /var/log/parsec. Каждая запись файла соответствует одному зарегистрированному событию. Подсистема PARSEC позволяет более эффективно управлять регистрацией событий, непосредственно связанных с безопасностью ОС. Однако из-за сравнительно малой распространенности Astra Linux SE в мире выбор готовых инструментов для анализа собранной подсистемой PARSEC информации весьма скуден.

Среди известных средств журналирования следует также отметить службу auditd [17], предназначенную для мониторинга событий ОС Linux и их фиксации в соответствующих журналах. Этот инструмент тесно взаимодействует с ядром ОС и, наблюдая за системными вызовами, может отслеживать практически любые события, происходящие в ОС, например, связанные с чтением, записью, выполнением, изменением прав доступа для файлов. Среди недостатков auditd выделяют то, что большинство событий безопасности на уровне системных вызовов трудно отличить от нормальной работы приложения. Кроме этого, auditd может существенно замедлять работу ОС.

Современные средства аудита сохраняют информацию о подключениях USB-устройств

в системных журналах, таких как /var/log/syslog и /var/log/messages. Чтобы узнать, было ли USB-устройство подключено к компьютеру, достаточно проверить системные журналы на наличие соответствующей записи. Ручная проверка администратором содержимого журналов практически нереализуема при большом числе контролируемых компьютеров и частом подключении USB-устройств. Как уже отмечалось, доступ к разработанным специально для Astra Linux SE программам [12, 13], автоматизирующим сбор информации о подключаемых устройствах, отсутствует, поэтому авторы рассматривают аналогичные программные средства, разработанные для других версий Linux. Таким программным средством является Usbrip (https://github.com/snovvcrash/usbrip).

Средство Usbrip анализирует данные системных журналов, выявляет события подключения USB-устройств и направляет собранную информацию в стандартный поток вывода или в файл. В удобном для анализа виде Usbrip предоставляет исчерпывающую информацию о USB-устройствах, подключенных к контролируемому компьютеру. Собранная информация позволяет понять, какого типа устройство было подключено: носитель информации, периферийное устройство ввода-вывода или беспроводное устройство. Однако Usbrip не может быть непосредственно внедрено в среду Astra Linux SE, поскольку она не позволяет устанавливать некоторые необходимые для работы Usbrip компоненты. Для возможности работы в среде Astra Linux средство Usbrip было модифицировано путем исключения необязательных компонентов.

Программная система аудита подключений USB-устройств ALUMNUS

Для достижения поставленных целей была разработана программная система аудита подключений USB-устройств, получившая название ALUMNUS (Astra Linux USB Monitoring Network Unified System). Архитектура ALUMNUS определяется следующими требованиями. Система должна обеспечивать централизованный сбор информации в ЛВС о подключениях USB-устройств к контролируемым компьютерам под управлением ОС Astra Linux SE, а также длительное хранение собранных данных и доступ к ним пользователей системы ALUMNUS. Программная система должна работать с большим количеством контролируемых компьютеров и, соответственно, организовывать очередь запросов от них к серверу. Пользователями программной системы ALUMNUS являются администраторы вычислительно-информационной инфраструктуры, в которую включена ЛВС компьютеров под управлением Astra Linux SE. Факты подключений USB-устройств к контролируемым компьютерам должны обнаруживаться ALUMNUS. Системный администратор как пользователь ALUMNUS должен иметь возможность просматривать и анализировать события, связанные с обнаруженными подключениями.

Структура системы ALUMNUS представлена на рисунке 1. На каждом контролируемом компьютере под управлением Astra Linux SE функционирует модуль сбора данных о подключениях USB-устройств, который направляет информацию об обнаруженных подключениях на специально выделенный сервер для последующей обработки. Для обеспечения отказоустойчивости системы необходимо, чтобы запросы, отправляемые с контролируемых компьютеров, гарантированно обрабатывались сервером. Для этого организована очередь сообщений-запросов, в которую осуществляют запись модули сбора информации. Запросы из очереди в порядке поступления обрабатываются функционирующим на сервере модулем проверки легитимности подключений USB-устройств. Этот модуль записывает информацию о событии в общее хранилище данных и отправляет запрос модулю обработки запросов к БД легитимных USB-устройств на наличие в этой базе информации о подключенном устройстве. Последний делает запрос к БД легитимных устройств и, используя очередь запросов, отправляет ответ серверу программной системы ALUMNUS. После того как сервер получил ответ о наличии или отсутствии информации о USB-устройстве в БД, информация о событии дополняется информацией о легитимности подключения USB-устройства к контролируемому компьютеру. Администратор вычислительно-информационной инфраструктуры в качестве пользователя системы ALUMNUS может посмотреть и проанализировать информацию о USB-подключениях через соответствующий интерфейс.

Рассмотрим предложенный технологический стек программной системы ALUMNUS, представленный на рисунке 2.

Для удобства разобьем структуру программной системы ALUMNUS на уровни (рис. 3) (синим обозначено взаимодействие с вычислительно-информационной инфраструктурой).

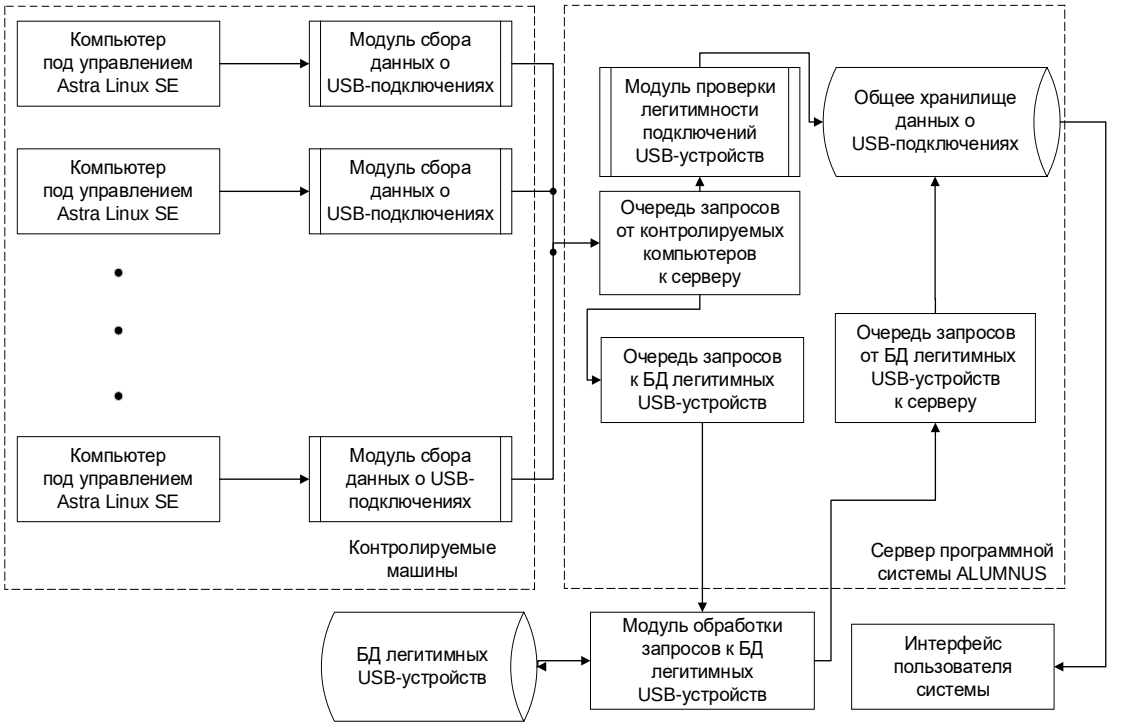

Рис. 1. Структура программной системы ALUMNUS

Fig. 1. The structure of the ALUMNUS software system

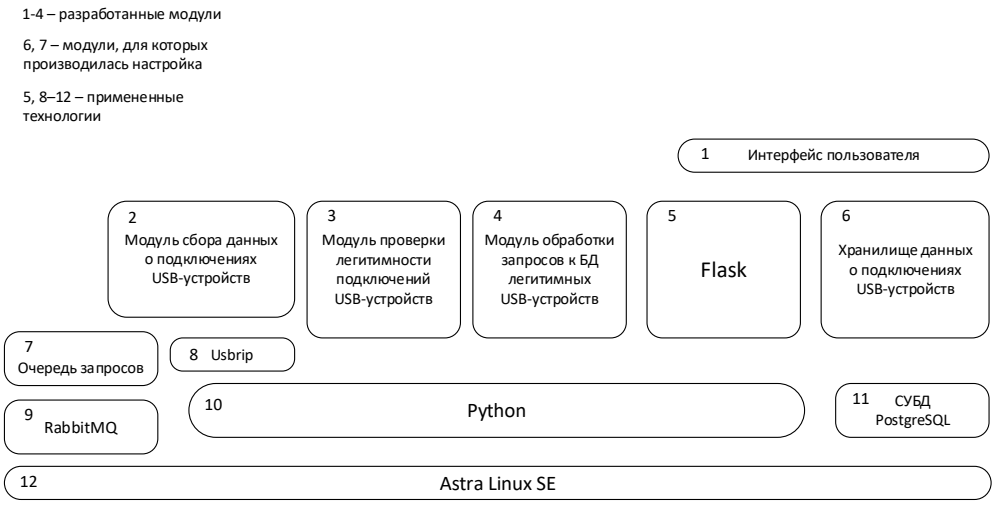

Рис. 2. Технологический стек программной системы ALUMNUS

Fig. 2. The technological stack of the ALUMNUS software system

На уровнях 1 и 2 функционирует модуль сбора данных о USB-подключениях, основанный на модифицированном для работы в Astra Linux SE программном средстве Usbrip.

Уровень 3 реализован с помощью брокера сообщений RabbitMQ [18]. Он позволяет организовать очереди запросов для обеспечения надежного взаимодействия компонентов системы (в частности, между контролируемыми компьютерами и сервером, между сервером Astra Linux и сервером БД легитимных USB-устройств). RabbitMQ совместим с Astra Linux SE и позволяет в достаточной мере масштабировать процесс приема и обработки запросов. В случае увеличения количества контролируемых компьютеров и недостаточности мощностей сервера для своевременной обработки запросов от контролируемых машин RabbitMQ

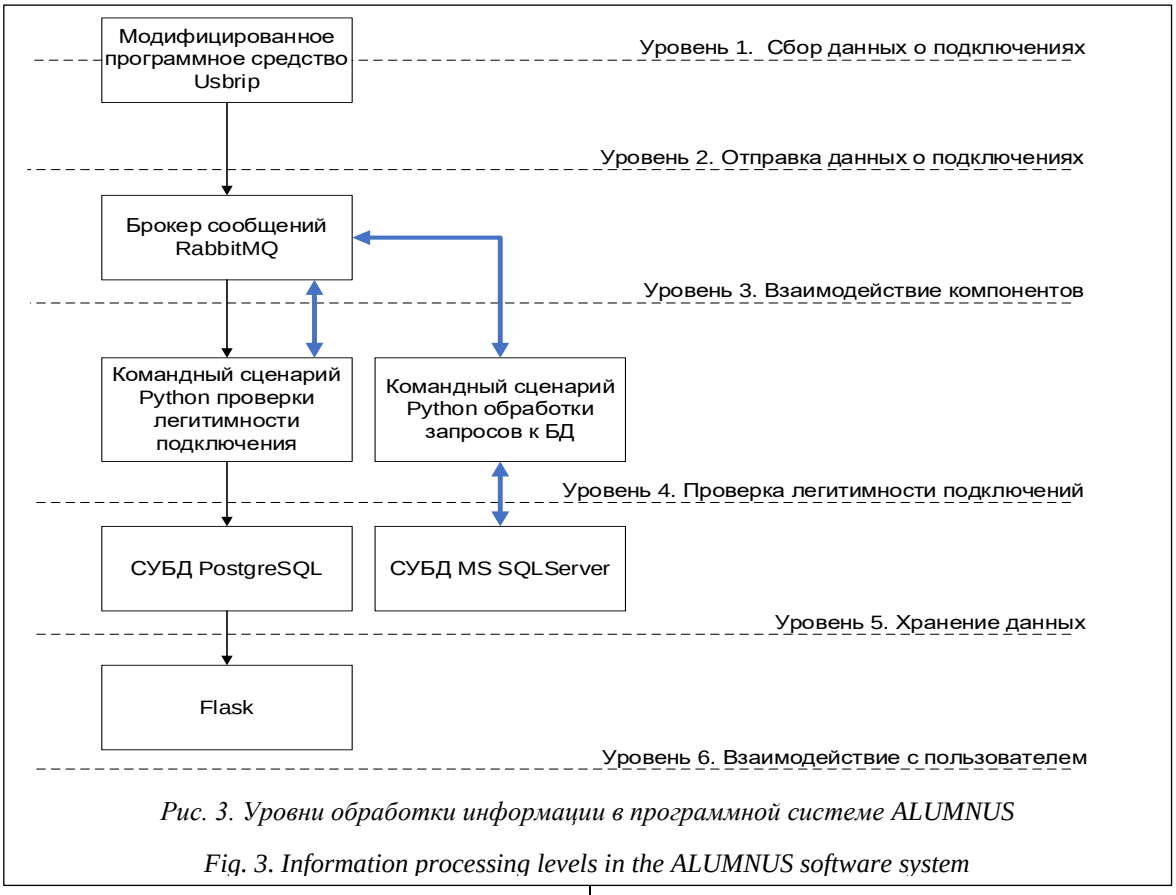

Рис. 3. Уровни обработки информации в программной системе ALUMNUS

Fig. 3. Information processing levels in the ALUMNUS software system

позволит настроить несколько обработчиков на одну очередь запросов.

На уровне 4 функционируют модуль проверки легитимности и модуль обработки запросов к БД легитимных USB-устройств. Эти модули реализованы на языке программирования Python в виде командных сценариев (скриптов).

Уровень хранения данных о подключениях USB-устройств к компьютерам под управлением ОС Astra Linux SE реализован на основе СУБД PostgreSQL [19]. PostgreSQL официально поддерживается разработчиками Astra Linux SE и обеспечивает необходимый уровень как производительности, так и безопасности. PostgreSQL поддерживает широкий спектр языков программирования и платформ, что позволяет достаточно просто связать СУБД и уровень, обеспечивающий взаимодействие компонентов системы ALUMNUS.

Данные из хранилища доступны пользователю через интерфейс, реализованный с помощью фреймворка Flask [20]. Этот фреймворк представляет собой написанный на языке Python микрофреймворк для разработки веб-приложений. Flask, не требуя для своей работы специальных инструментов или библиотек, существенно упрощает разработку приложений или расширений (плагинов). Активное сообщество разработчиков и пользователей поддерживает Flask, что позволяет минимизировать трудозатраты на разработку.

С точки зрения выявления подключений USB-устройств на контролируемых компьютерах центральным модулем системы ALUMNUS является модуль сбора данных, который, как уже упоминалось, представляет собой модифицированное программное средство Usbrip. Его модификация отображена на рисунке 4.

Из программного средства Usbrip были исключены модули, функциональность которых оказалась невостребованной в рамках системы ALUMNUS. Модуль отбора событий о подключениях USB-устройств был доработан функцией отправки событий на сервер ALUMNUS. Кроме того, добавлена возможность работы с временной меткой, необходимой для корректного определения даты и времени события. В итоге модуль сбора данных, используя модифицированный модуль Usbrip отбора событий о подключениях USB-устройств, производит на контролируемом компьютере поиск системных журналов и отбор из них информации о подключениях USB-устройств. После этого временная метка отобранного события сравнивается с

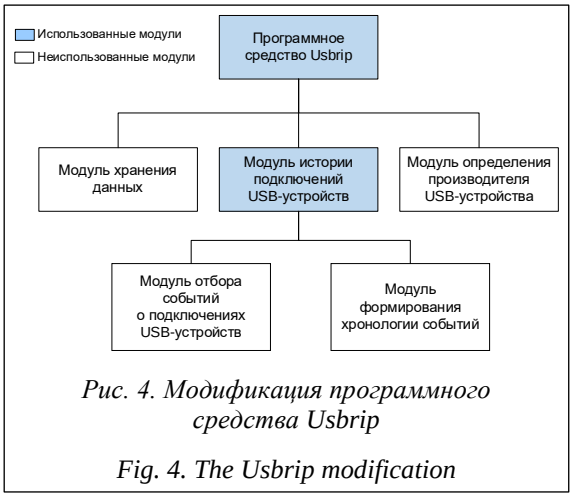

Рис. 4. Модификация программного средства Usbrip

Fig. 4. The Usbrip modification

временной меткой последнего события, информация о котором была отправлена в очередь запросов на сервере. Если выявленное подключение является новым, соответствующее сообщение отправляется в очередь запросов.

Модуль проверки легитимности подключений USB-устройств получает информацию о событии (подключение USB-устройства) из очереди запросов. Далее происходят запись информации о событии в общее хранилище данных о подключениях USB-устройств и отправка через очередь сообщений запроса модулю обработки запросов к БД легитимных USB-устройств. После получения ответа событие в общем хранилище данных помечается как представляющее угрозу информационной безопасности (если данных о USB-устройстве нет в БД) или не представляющее угрозу (если данные о USB-устройстве есть в БД).

Опытная эксплуатация программной системы ALUMNUS

Программная система ALUMNUS была реализована в виде действующего макета, развернутого в защищенном сегменте установленного в Межведомственном суперкомпьютерном центре (МСЦ) РАН суперкомпьютера МВС-10П ОП. Схема сегмента в составе ЛВС центра представлена на рисунке 5. Для обработки информации в защищенном режиме в составе суперкомпьютера часть вычислительных узлов выделена в отдельную подсеть. Изоляция и защита закрытой подсети обеспечиваются за счет применения межсетевых экранов как на сетевом маршрутизаторе, так и внутри самой подсети.

На каждом из вычислительных узлов установлена ОС Astra Linux SE в версии «Смоленск». Дополнительную защиту узлов составляют программно-аппаратный комплекс доверенной загрузки ОС «Соболь» [21] и средство антивирусной защиты Kaspersky Endpoint Security. В защищаемой подсети выделены АРМ

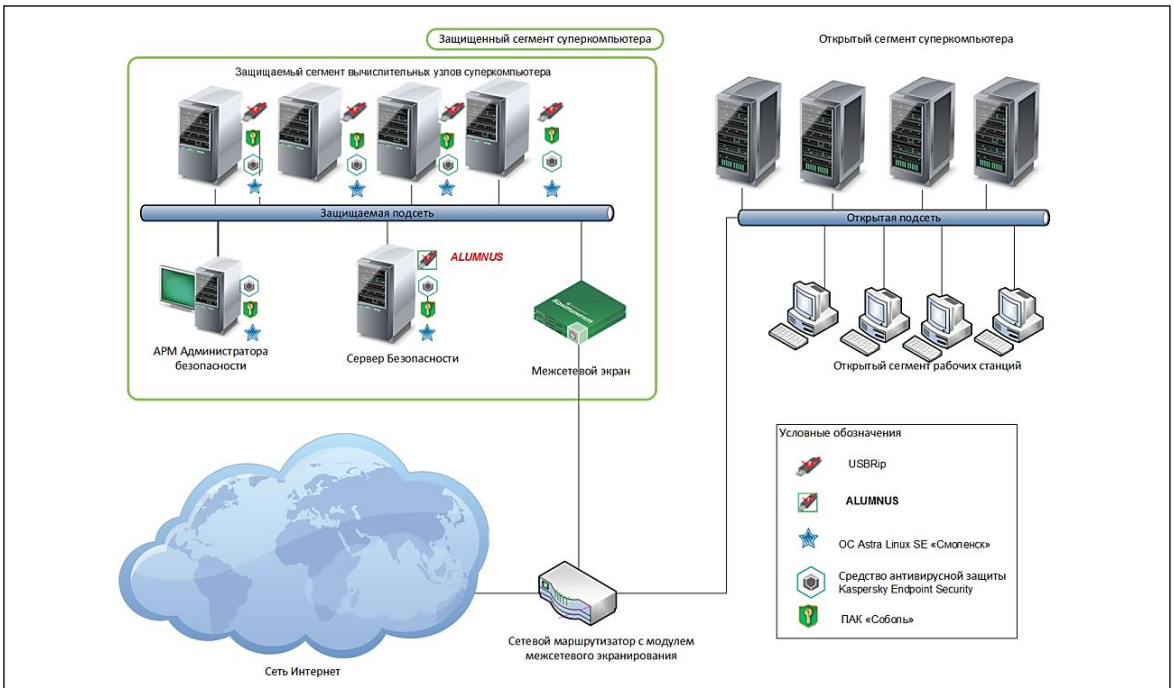

Рис. 5. Схема защищенного сегмента суперкомпьютера МСЦ РАН

Fig. 5. The secure segment of the JSCC RAS supercomputer

администратора безопасности и сервер безопасности, на котором функционирует и БД легитимных подключаемых USB-устройств.

Модули системы ALUMNUS сбора данных о подключениях USB-устройств были развернуты на вычислительных узлах. Сервер программной системы ALUMNUS функционирует на сервере безопасности, а интерфейс пользователя доступен через АРМ администратора безопасности. Таким образом, структура системы ALUMNUS органично легла на существующую структуру защищенного сегмента суперкомпьютера.

В ходе опытной эксплуатации модуль обработки запросов к БД легитимных устройств связывался с двумя разными СУБД – PostgreSQL и Microsoft SQL Server, продемонстрировав совместимость с обеими СУБД. Опытная эксплуатация макета программной системы ALUMNUS продемонстрировала применимость разработанного технологического стека для построения подобного рода программных решений. ALUMNUS позволяет в оперативном режиме автоматически выявлять подключения USB-устройств к контролируемым компьютерам в защищаемой подсети, выявлять факты нелегитимных подключений и извещать о таких фактах администратора безопасности. За время опытной эксплуатации произведено несколько десятков подключений USB-устройств, часть из которых были нелегитимными. Программная система ALUMNUS выявила все произведенные подключения и зафиксировала факты нелегитимных подключений в своей БД.

Перспективными направлениями развития ALUMNUS видятся интеграция ее модулей в функционирующую на сервере безопасности SIEM-систему и расширение спектра контролируемых действий пользователя, фиксируемых модулем сбора информации.

Заключение

Анализ доступных источников показал, что в настоящее время отсутствует комплексное программное решение, позволяющее проводить контроль подключений USB-устройств к компьютерам под управлением ОС Astra Linux SE в ЛВС. Для построения такого решения авторами предложен технологический стек компонентов, включающий, помимо ОС Astra Linux SE, систему очередей сообщений RabbitMQ, СУБД PostgreSQL, фреймворк Flask и модифицированное средство USBRip обнаружения подключений USB-устройств. На базе предложенного технологического стека разработана структура программной системы контроля USB-подключений, получившей название ALUMNUS. Структура включает модули сбора информации об USB-подключениях, функционирующие на контролируемых компьютерах, и серверную часть, агрегирующую собранную информацию. Реализованный макет программной системы ALUMNUS был развернут в защищенном сегменте суперкомпьютера МВС-10П ОП, установленного в МСЦ РАН. Опытная эксплуатация макета показала применимость предложенных технологического стека и структуры для построения программных систем оперативного контроля подключений USB-устройств к компьютерам под управлением ОС Astra Linux SE.

Список литературы

1. Буранова М.М., Вахрушева Е.А. Операционная система Astra Linux // Информационные технологии в науке, промышленности и образовании: сб. тр. науч.-технич. конф. 2021. С. 216–222.

2. Девянин П.Н., Тележников В.Ю., Третьяков С.В. Основы безопасности операционной системы Astra Linux Special Edition. Управление доступом. М.: Горячая линия–Телеком, 2022. 148 с.

3. Кочетова И.В. Возможности использования операционной системы особого назначения «Astra Linux Special Edition 1.5» // Информационные технологии XXI века: сб. науч. тр. 2020. С. 386–390.

4. Девянин П.Н., Хорошилов А.В., Тележников В.Ю. Формирование методологии разработки безопасного системного программного обеспечения на примере операционных систем // Тр. ИСП РАН. 2021. Т. 33. № 5. С. 25–40. doi: 10.15514/ISPRAS-2021-33(5)-2.

5. Negus C. Enhancing Linux security with SELinux. In: Linux Bible, 2020, pp. 669–697. doi: 10.1002/9781119209539.ch24.

6. Ecarot T., Dussault S., Souid A., Lavoie L., Ethier J.-F. AppArmor for health data access control: Assessing risks and benefits. Proc. 7th IOTSMS, 2020, pp. 1–7. doi: 10.1109/IOTSMS52051.2020.9340206.

7. Мылицын Р.Н., Девянин П.Н. Практика построения информационных систем в защищенном исполнении на базе операционной системы Astra Linux Special Edition // Состояние и перспективы развития современной науки по направлению «Информационная безопасность»: сб. статей. 2020. С. 448–453.

8. Невров А.А., Попов Г.А. Подходы к контролю использования съемных машинных носителей информации на средствах вычислительной техники под управлением ОС Astra Linux SE // Южно-Сибирский науч. вестн. 2017. № 1. С. 45–48.

9. Nevrov A.A., Andreev I.L. Automation control of use attached media devices on computers with Astra Linux SE operation system. Proc. XXIII Int. Open Sci. Conf. MIP, 2018, pp. 245–249.

10. Лобач А.О. Интеграция СЗИ со службой каталогов Astra Linux Directory: проблемы и подходы // Вопросы защиты информации. 2021. № 3. С. 3–7. doi: 10.52190/2073-2600_2021_3_3.

11. Шарафутдинова Л.В., Щерба М.В. Обнаружение несанкционированных действий в операционной системе Astra Linux // НИР-22: матер. науч.-практич. конф. 2022. С. 106–112.

12. Невров А.А., Андреев И.Л., Машошин В.Ю., Бердникова М.Р. Учет и контроль USB флеш-накопителей: Свид. о регистр. ПрЭВМ № 2018611763. Рос. Федерация, 2018.

13. Толстых А.А., Поликарпов Е.С., Лунев Ю.С., Цимбал В.Н. Программа для аудита подключений USB устройств к персональному компьютеру под управлением ОС Astra Linux: Свид. о регистр. ПрЭВМ № 2021615138. Рос. Федерация, 2021.

14. Vielberth M. Security information and event management (SIEM). In: Encyclopedia of Cryptography, Security and Privacy, 2021, pp. 1–3. doi: 10.1007/978-3-642-27739-9_1681-1.

15. Gerhards R. Rsyslog: Going up from 40K messages per second to 250K. Proc. Linux Kongress, 2010. URL: https://www.researchgate.net/profile/Rainer-Gerhards-2/publication/228694459_Rsyslog_going_up_from_40K_messages_per_second_to_250K/links/5830958108ae004f74c0f24f/Rsyslog-going-up-from-40K-messages-per-second-to-250K.pdf (дата обращения: 04.05.2023).

16. Chuvakin A., Schmidt K., Phillips C. Syslog-ng case study. In: Logging and Log Management, 2013, pp. 93–101. doi: 10.1016/B978-1-59-749635-3.00005-1.

17. Zam Zam M. Auditd: Rule writing for better threat detection on *nix devices. BS Comput. Sci., 2021, pp. 1–21.

18. Christudas B. Install, configure, and run RabbitMQ cluster. In: Practical Microservices Architectural Patterns, 2019, pp. 827–847. doi: 10.1007/978-1-4842-4501-9_21.

19. Shaik B., Vallarapu A. PostgreSQL architecture. In: Beginning PostgreSQL on the Cloud, 2018, pp. 33–61. doi: 10.1007/978-1-4842-3447-1_2.

20. Relan K. Beginning with Flask. In: Building REST APIs with Flask, 2019, pp. 1–26. doi: 10.1007/978-1-4842-5022-8_1.

21. Велюллаев Э.У., Гончаренко Ю.Ю., Девицына С.Н. Аппаратная защита автоматизированных рабочих мест // РТ-2022: матер. Междунар. науч.-технич. конф. 2022. № 5. С. 208.

Software & Systems doi: 10.15827/0236-235X.142.432-441 2023, vol. 36, no. 3, pp. 432–441

USB connections control in the local network of computers running under Astra Linux SE

Anton V. Baranov
Pavel M. Korepanov
Ignat A. Lepeshev

For citation
Baranov, A.V., Korepanov, P.M., Lepeshev, I.A. (2023) 'USB connections control in the local network of computers running under Astra Linux SE', *Software & Systems*, 36(3), pp. 432–441 (in Russ.). doi: 10.15827/0236-235X.142.432-441
Article info
Received: 19.06.2023 After revision: 28.06.2023 Accepted: 03.07.2023

Abstract. The constant growth of information security requirements, as well as a major trend towards import replacement in the system software, have led to the widespread use of infrastructure solutions based on the domestic Astra Linux Special Edition (SE) operating system. Astra Linux makes it possible to build secure computing systems including supercomputers for processing confidentional information. At the same time, the audit of connecting USB devices to computers is one of the most important problems of ensuring information security. An analysis of the existing open access works shows the lack of turnkey solutions working in the Astra Linux SE environment. The article discusses a possible technological stack of such solution. The technological stack includes besides Astra Linux the RabbitMQ message-broker sofware, the Flask micro web framework, the PostgreSQL database, and the USBRip forensics tool for keeping track of USB event history on Linux machines. The proposed modular structure of the software USB connections audit system is considered. It includes the modules for collecting USB connections artifacts on controlled computers, the collected information aggregation module, and the module for checking the USB device connection permissions. The proposed structure and technological stack were implemented as a prototype of the software system called ALUMNUS. The prototype was deployed and tested in the secure segment of the MVS-10P OP supercomputer installed at the Joint Supercomputer Center of the Russian Academy of Sciences.

Keywords: supercomputer center, information security, Astra Linux SE, USB connection control
Acknowledgements. The work has been carried out at the JSCC RAS within the framework of the state assignment FNEF-2022-0016

Reference List

1. Buranova, M.M., Vakhrusheva, E.A. (2021) 'Astra Linux Operating System', *Proc. Conf. Inform. Tech. in Sci., Industry and Education*, pp. 216–222 (in Russ.).
2. Devyanin, P.N., Telezhnikov, V.I., Tretyakov, S.V. (2022) *Fundamentals of the Operating System Astra Linux Special Edition Security. Access Control*. Moscow, 148 p. (in Russ.).
3. Kochetova, I.V. (2020) 'Possibilities of using the special-purpose operating system Astra Linux Special Edition 1.5', *Proc. Inform. Tech. of the XXI Century*, pp. 386–390 (in Russ.).
4. Devyanin, P.N., Telezhnikov, V.I., Khoroshilov, A.V. (2021) 'Building a methodology for secure system software development on the example of operating systems', *Proc. of the ISP RAS*, 33(5), pp. 25–40 (in Russ.). doi: 10.15514/ISPRAS-2021-33(5)-2.
5. Negus, C. (2020) 'Enhancing Linux security with SELinux', in *Linux Bible*, pp. 669–697. doi: 10.1002/9781119209539.ch24.
6. Ecarot, T., Dussault, S., Souid, A., Lavoie, L., Ethier, J.-F. (2020) 'AppArmor for health data access control: Assessing risks and benefits', *Proc. 7th IOTSMS*, pp. 1–7. doi: 10.1109/IOTSMS52051.2020.9340206.
7. Mylitsyn, R.N., Devyanin, P.N. (2020) 'The practice of building secure information systems based on the Astra Linux special edition operating system', *Proc. Conf. State and Prospects for the Modern Sci. Development in the Inform. Security*, pp. 448–453 (in Russ.).
8. Nevrov, A.A., Popov, G.A. (2017) 'Approaches of controlling of the use of removable computer media of computer equipment under the control of Astra Linux SE operating system', *South-Siberian Sci. Bull.*, (1), pp. 45–48 (in Russ.).
9. Nevrov, A.A., Andreev, I.L. (2018) 'Automation control of use attached media devices on computers with Astra Linux SE operation system', *Proc. XXIII Int. Open Sci. Conf. MIP*, pp. 245–249.
10. Lobach, A.O. (2021) 'Integration of information security system with the Astra Linux directory service: Problems and approaches', *Inform. Security Iss.*, (3), pp. 3–7. doi: 10.52190/2073-2600_2021_3_3 (in Russ.).
11. Sharafutdinova, L.V., Scherba, M.V. (2022) 'Unauthorized actions detection in the Astra Linux operating system', *Proc. Sci-Tech. Conf. NIR-22*, pp. 106–112 (in Russ.).
12. Nevrov, A.A., Andreev, I.L., Mashoshin, V.Yu., Berdnikova, M.R. (2018) *Accounting and Control of USB Flash Drives*, Pat. RF, № 2018611763.
13. Tolstykh, A.A., Polikarpov, E.S., Lunev, Yu.S., Cimbal, V.N. (2021) *A Program for Auditing USB Device Connections to a Personal Computer Running under Astra Linux OS*, Pat. RF, № 2021615138.
14. Vielberth, M. (2021) 'Security information and event management (SIEM)', in *Encyclopedia of Cryptography, Security and Privacy*, pp. 1–3. doi: 10.1007/978-3-642-27739-9_1681-1.
15. Gerhards, R. (2010) 'Rsyslog: Going up from 40K messages per second to 250K', *Proc. Linux Kongress*, available at: https://www.researchgate.net/profile/Rainer-Gerhards-2/publication/228694459_Rsyslog_going_up_from_40K_messages_per_second_to_250K/links/5830958108ae004f74c0f24f/Rsyslog-going-up-from-40K-messages-per-second-to-250K.pdf (accessed May 04, 2023).
16. Chuvakin, A., Schmidt, K., Phillips, C. (2013) 'Syslog-ng case study', in *Logging and Log Management*, pp. 93–101. doi: 10.1016/B978-1-59-749635-3.00005-1.
17. Zam Zam, M. (2021) 'Auditd: Rule writing for better threat detection on *nix devices', *BS Comput. Sci.*, pp. 1–21.
18. Christudas, B. (2019) 'Install, configure, and run RabbitMQ cluster', in *Practical Microservices Architectural Patterns*, pp. 827–847. doi: 10.1007/978-1-4842-4501-9_21.
19. Shaik, B., Vallarapu, A. (2018) 'PostgreSQL architecture', in *Beginning PostgreSQL on the Cloud*, pp. 33–61. doi: 10.1007/978-1-4842-3447-1_2.
20. Relan, K. (2019) 'Beginning with Flask', in *Building REST APIs with Flask*, pp. 1–26. doi: 10.1007/978-1-4842-5022-8_1.
21. Velullaev, E.U., Goncharenko, J.J., Devitsyna, S.N. (2022) 'Hardware protection of automated jobs', *Proc. Int. Sci.-Tech. Conf. RT-2022*, (5), pp. 208 (in Russ.).

Авторы

Баранов Антон Викторович [1], к.т.н., доцент, ведущий научный сотрудник , antbar@mail.ru, abaranov@jscc.ru
Корепанов Павел Михайлович [1], начальник сектора информационной безопасности , kpm@jscc.ru
Лепешев Игнат Анатольевич [1], стажер-исследователь, rin.l@yandex.ru

[1] Межведомственный суперкомпьютерный центр РАН, г. Москва, 119991, Россия

Authors

Anton V. Baranov [1], Ph.D. (Engineering), Associate Professor, Leading Researcher, antbar@mail.ru, abaranov@jscc.ru
Pavel M. Korepanov [1], Head of information security Sector, kpm@jscc.ru
Ignat A. Lepeshev [1], intern Researcher, rin.l@yandex.ru

[1] Joint Supercomputer Center of RAS, Moscow, 119991, Russian Federation