Закупочные сервисы iSource.
Руководство администратора.

iSource/Marketpace/NVI

Версия 0.23. Декабрь 2022

is000-AGD_OPE.1

Оглавление

Термины, сокращения и обозначения

Термин/Сокращение	Определение
ACL	Access Control List, список (списки) контроля доступа
AMD	Advanced Micro Devices, коммерческая компания, производитель микроэлектроники (США)
AMQP	Advanced Message Queuing Protocol, открытый протокол прикладного уровня для передачи сообщений между компонентами системы
APAPI	Automatic Private IP Addressing, протокол автоматического выбора адреса
API	Application Programming Interface, программный интерфейс программы или приложения
ASLR	Address Space Layout Randomization, механизм ядра ОС Linux/UNIX для предоставления случайной адресации при выделении страниц виртуальной памяти
BIOS	Basic Input-Output System. Основная (базовая) система ввода-вывода, устаревший интерфейс конфигурации оборудования для компьютера
BPF	Berkeley Packet Filters. Фильтры пакетов Беркли, подсистема ядра, изначально разработанная для FreeBSD и портированная из неё в Linux, предназначенная для фильтрации пакетов в ядре
BSD	Berkeley Software Distribution, операционная система университета Беркли. В настоящее время – семейство распространенных операционных систем, прослеживаемых к оригинальной ОС BSD, объединенных общим архитектурным дизайном и генеалогией. Наиболее известные операционные системы семейства – FreeBSD, OpenBSD и NetBSD, хотя есть и некоторые другие
CIS	Center for Internet Security, Центр безопасности Интернет. Некоммерческая международная организация, разрабатывающая стандарты и инструменты в области информационной безопасности
CUPS	Common UNIX Print System. Основная современная служба печати для UNIX/Linux

Термин/Сокращение	Определение
DAC	Discretional Access Control. Дискреционное разграничение доступа. Обычно в Linux под этим понимаются стандартные для UNIX биты разрешений, атрибуты файлов, расширения POSIX ACL и Capabilities
DNS	Domain Name System. Система (служба) доменных имен
EVM	Extended Verification Module. Функция ядра Linux, которая принимает решение о запрете или разрешении запуска программы (библиотеки, драйвера), находящегося на контроле IMA (см. IMA)
FIPS	Federal Information Processing Standard – американский (и международный) стандарт в области защиты информации и криптографии
FQDN	Fully Qualified Domain Name – полностью пределенное доменное имя
GCC	GNU C Compiler – наиболее популярный компилятор языка Си с открытым исходным кодом, широко используемый в интересах разработки Linux и не только
GID	Group Identifier. Идентификатор группы в операционных системах типа Linux или UNIX
GRUB	Grand Unified Bootloader. Современный загрузчик для операционных систем типа UNIX/Linux
IANA	Internet Assigned Numbers Authority. Администрация адресного пространства Интернет. Международная организационная структура, управляющая адресным пространством сети Internet, штаб-квартира которой находится в США
IDM, IdM	Identity Management. Система управления идентификаторами. Предназначена для централизованного управления учетными записями и правами пользователей всех информационных систем компании, работая на стыке управления IT-ресурсами, обеспечения информационной безопасности и эффективности бизнеса
IMA	Integrity Measurement Architecture. Архитектура проверки целостности. Функция ядра операционной системы Linux, которая обеспечивает подписывание и проверку заданных файлов для контроля их целостности
IP	Internet Protocol – основной протокол передачи данных в сетях Internet

Термин/Сокращение	Определение
IPA	Identification Policy and Audit. Идентификация, политика и аудит, наиболее распространенная в ОС Linux доменная служба каталогов на базе LDAP и аутентификации Kerberos со стандартизированной схемой i389-directory
ISO	International Standard Organisation – Международная организация по стандартизации
Linux	Акроним от Linux Is Not UniX, Linux это не UNIX. Семейство современных операционных систем, объединенных общими архитектурными подходами и принципом открытости в разработке
LLMNR	Link-local Multicast Name Resolution – протокол разрешения имен, предложенный компанией Microsoft
LSM	Linux Security Module – модуль безопасности Linux; технология ядра и набор интерфейсов ядра ОС Linux, которые позволяют поддерживать разнообразные расширения, связанные с безопасностью
MAC (в ИБ)	Mandatory Access Control – механизм полномочного разграничения доступа
MAC (в сетях)	Media Access Control – уникальный идентификатор, или также «физический адрес». Адрес, присваиваемый каждой единице активного оборудования или некоторым их интерфейсам в компьютерных сетях Ethernet
MAC (в шифровании)	Message Authentication Code – специальный код аутентификации сообщения, или «имитовставка» – определенный набор символов, который добавляется к сообщению и предназначен для обеспечения его целостности и аутентификации источника данных
mDNS	Multicast-DNS – протокол разрешения имен, предложенный компанией Apple
NFS	Network File System – протокол сетевого обмена, стандартный для Linux/UNIX
NIST	National Institute of Standard and Technology – Национальный институт стандартов и технологии. Регулирующая организация, США
NTP	Network Time Protocol – протокол службы единого времени
NX	No execute bit - бит отключения выполнения кода процессоров AMD x86

Термин/Сокращение	Определение
PCI-DSS	Payment Card Industry Data Security Standart - современный международный стандарт безопасности для информационных систем финансового сектора
PID	Process Identificator – идентификатор процесса
RBAC	Role-Based Access Control – механизм ролевого разграничения доступа
RPM	RedHat Package Manager – менеджер пакетов в Linux для управления программным обеспечением и формат этих пакетов, разработанный компанией RedHat (США)
SGID	Superuser Group Identificator – бит смены идентификатора группы администратора (root)
SMT	Superuser Group Identificator – бит смены идентификатора группы администратора (root)
SUID	Superuser Identificator – бит смены идентификатора администратора (root)
TCP	Transmission Control Protocol – протокол передачи данных с контролем передачи пакетов
TOTP	Time-based One Time Password – одноразовый пароль на основе времени. Популярная технология усиления аутентификации, когда пользователь дополнительно к основному паролю вводит одноразовый пароль, действующий некоторое время
UDP	User Datagramm Protocol – протокол передачи данных без контроля передачи пакетов
UEFI	Unified Extensible Firmware Interface – единый расширяемый интерфейс базовой системы ввода-вывода
UID (EUID)	User Identificator – идентификатор пользователя (в т.ч. эффективный)

Термин/Сокращение	Определение
UNIX	Бывш. UNICS – Uniplexed Information and Computing System – аббревиатура, означающая в переводе: «упрощенная информационная и вычислительная система». Производная реализация от ОС MULTICS, первоначально созданная К. Томпсоном, Д. Ричи и Д. Макилроем из Bell Labs в начале семидесятых годов XX-го века. А также в настоящее время – семейство операционных систем, объединенных общими архитектурными подходами, иногда общей генеалогией, имеющих действующие сертификаты соответствия спецификации UNIX 03 компании OpenGroup (США) и приобретшие у нее же право на использование торговой марки UNIX™. В аббревиатуре UNICS буквы «CS» для краткости были заменены на «X»
USA	United States of America, см. «США»
VDSO	Virtual ELF Dynamic Shared Object – виртуальный динамически разделяемый объект, который размещается только в адресном пространстве отдельной программы
XD	Execute disable bit – бит отключения выполнения кода процессоров Intel x86
АВЗ	Антивирусная защита
АПМДЗ	Аппаратно-программный модуль доверенной загрузки
АРМ	Автоматизированное рабочее место
АС	Автоматизированная система
Авторизация	Процедура или функция, завершающая процесс идентификации и аутентификации, определяющая права доступа субъекта к ресурсам (объектам) и управления этим доступом на основе заданных атрибутов
Администратор	Сущность или роль, которая имеет некоторый уровень доверия в отношении всех политик, реализуемых функциями безопасности
Аккредитация	Процедура официального подтверждения соответствия объекта установленным критериям и показателям (стандарту)
Активы	Сущности, предположительно представляющие ценность для владельца информационной системы

Термин/Сокращение	Определение
Аукцион	Публичная продажа товаров, имущества предприятий, произведений искусства, и других объектов, которая производится по заранее установленным правилам аукциона. Общим для всех аукционов принципом является принцип состязательности между потенциальными покупателями. В процессе состязания между покупателями за право приобрести товар выявляется победитель аукциона
Аутентификационные данные	Информация, используемая для проверки предъявленного идентификатора пользователя
БД	База данных
БДУ	База данных уязвимостей
Бэк-офис	Система управления Площадкой, обеспечивающая возможность загрузить Номенклатуру, создать аукцион и обработать заказ
ВНИИФТРИ	Всероссийский научно-исследовательский институт физико-технических и радиотехнических измерений
Голландский аукцион	Аукцион, в ходе которого вначале объявляется самая высокая цена на продаваемый товар, а затем ставки снижаются до той, на которую согласится первый покупатель, которому и продается товар
Доверие	Основание для уверенности в том, что операционная система или информационная система отвечает некоторому конкретному набору функциональных требований безопасности или критерию
Договор	Договор купли-продажи - вид сделок, по которому одна сторона, осуществляющий деятельность по продаже товара, обязуется передать товар в собственность другой стороне (покупателю) для использования, а покупатель обязуется принять этот товар и уплатить за него определённую денежную сумму (цену). Обязательным для договора является указание наименования и количества товара. В противном случае он признается незаключенным
ЖЦ	Жизненный цикл
ЗБ	Задание по безопасности
Защищаемая информация	Совокупность активов, а также функции безопасности (как программные сущности) и интерфейсы к ним

Термин/Сокращение	Определение
Злоумышленник	Иначе «нарушитель», «потенциальный нарушитель»: это лицо, намеренно пытающееся воздействовать на активы и на основные свойства защищаемой информации – целостность, доступность или конфиденциальность, с целью получения выгоды или без таковой
ИБ	Информационная безопасность
ИС	Информационная система
ИТ	Информационные технологии
Идентификатор	Представление, однозначно идентифицирующее сущность (например, пользователя, процесс или диск) в контексте конкретного объекта операционной системы
Информационная безопасность	Свойство информации сохранять конфиденциальность, целостность и доступность
КЦ	Контроль целостности
Контрагент	Лицо или учреждение, берущее на себя известные обязательства по договору
Лот	Единица купли-продажи во время торгов на аукционах. Размеру лота соответствует определённый заранее объём товара в натуральном выражении. Стандартный размер сделки, контракта, совершаемых во время торгов, устанавливается правилами аукционной и биржевой торговли. Каждому аукционному лоту присваивается порядковый номер и устанавливается своя аукционная цена в ходе торга
МТР	Справочная позиция, описание предмета в рамках характеристик по стандарту (обычно ГОСТ)
Негативные действия	Действия, выполняемые источником угрозы (злоумышленником, нарушителем, и т.п.) по отношению к активам
Номенклатура	Конкретный экземпляр МТР, содержит в себе информацию о Поставщике, Складке, Количестве и т.д.
ОЗУ	Оперативное запоминающее устройство – микросхемы оперативной памяти средства вычислительной техники
ОО	Объект оценки. Оцениваемая (тестируемая, проверяемая) сущность (информационная система), установленная в заданной среде, заданным способом и в заданной конфигурации
ОС	Операционная система

Термин/Сокращение	Определение
Объект доступа	Единица информационного ресурса (файл, каталог, том, устройство, страница памяти, и т.п.), доступ к которой регламентируется правилами разграничения доступа и по отношению к которой субъекты доступа выполняют операции
Победитель аукциона	Победителем аукциона признаётся лицо, выигравшее аукцион в соответствии с его правилами. В этом случае объект приобретается победителем аукциона
Покупатель	Физическое или юридическое лицо, осуществляющее оплату деньгами и являющееся приобретателем товара
Политика	Совокупность правил, описывающих конкретный режим безопасности, реализуемый функциями безопасности, и выраженных в виде множества некоторых функциональных требований безопасности
Пользователь	Пользователь ОС, не имеющий административных полномочий
Регистрация	Процедура, завершающая процесс идентификации и аутентификации. То же, что и «авторизация»
Регистрация (для аудита)	Факт фиксации какого-либо события в журнале
Роль	Набор доступов для выполнения определённого круга задач в системе
СЗИ	Средство (средства) защиты информации
США	Соединенные Штаты Америки – самое крупное государство на континенте Северная Америка, где-то между Мексикой и Канадой
Субъект	Активная сущность (пользователь, администратор, процесс, порождаемый пользователем и/или функционирующий от его имени и т.п.), выполняющая операции над объектами
Счет на оплату	Документ, содержащий платежные реквизиты получателя (продавца), по которым плательщик (покупатель) осуществляет перевод денежных средств за перечисленные в счете товары, работы или услуги
ТК	Технический комитет
ТОИС	Техническое описание информационной системы (продукта)
Угроза	Совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации

Термин/Сокращение	Определение
Участник торгов	Физическое или юридическое лицо, участвующее в торгах и вносящее ценовые предложения
ФС	Файловая система
Физическое лицо	Субъект гражданского права. Как и любой другой субъект права, физическое лицо имеет права и обязанности
ЦП	Центральный процессор
Шаблон документа	Документ в котором уже есть все элементы, являющиеся общими для всех документов определенного типа. Заполняется данными ....
Электронная торговая площадка	Программно-аппаратный комплекс организационных, информационных и технических решений, обеспечивающих взаимодействие продавца и покупателя через электронные каналы связи
Юридическое лицо	Зарегистрированная в установленном законом порядке организация, фирма, компания, которая имеет в собственности, хозяйственном ведении или оперативном управлении обособленное имущество и отвечает по своим обязательствам этим имуществом, может от своего имени приобретать и осуществлять имущественные и личные неимущественные права, нести обязанности, быть истцом и ответчиком в суде

1 Идентификация документа

Наименование документа:
«Закупочные сервисы iSource. Руководство администратора.»

Идентификатор документа:
is000-AGD_OPE.1

Версия:
0.23

2 Соглашения, назначение и область применения

Настоящий документ «Закупочные сервисы iSource. Руководство администратора» is000-AGD_-OPE.1, содержит следующие материалы (которые также могут рассматриваться как свидетельства или части свидетельств семейств, согласно ГОСТ Р ИСО/МЭК 15408-3:2013 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности»:

- описание подготовительных процедур (AGD_PRE.1) для установки программных средств, входящих в состав комплекса «Закупочные сервисы iSource»;
- описание процедур для настройки и использования (AGD_OPE.1) программных средств, входящих в состав комплекса «Закупочные сервисы iSource», а также рекомендации по использованию и настройкам средств безопасности;
- описание процедур поставки (ALC_DEL.1) программных средств, входящих в состав комплекса «Закупочные сервисы iSource».

Также к настоящему документу относятся два приложения в составе:

- Приложение А. Руководство по настройке средств безопасности в Ubuntu 20.04 LTS. is000-AGD_PRE.1;
- Приложение Б. Определение жизненного цикла. is000-ALC_LCD.1.

При подготовке настоящего документа документации разработчик учитывает требования следующих стандартов:

- ГОСТ Р ИСО/МЭК 12207-99 «Информационная технология. Процессы жизненного цикла программных средств»;
- ГОСТ Р ИСО 9001-96 «Системы качества. Модель обеспечения качества при проектировании, разработке, производстве, монтаже и обслуживании»;

- ГОСТ Р ИСО/МЭК 9126-93 «Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению»;
- ГОСТ Р 56939-2016 «Защита информации. Разработка безопасного программного обеспечения. Общие требования»;
- ГОСТ Р 58412-2019 «Защита информации. Разработка безопасного программного обеспечения. Угрозы безопасности информации при разработке программного обеспечения»;
- ГОСТ Р ИСО/МЭК 15408-3:2013 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности».

2.1 Соглашения

Команды оболочки (выполняемые ИФБО) и параметры конфигурации выделены отдельным блоком листинга с отличающемся фоном для большего удобства:

```
# example shell command  <пример команды оболочки>
variable in configuration file "пример переменной или конфигурационного файла"
```

Листинг 1: Общий пример листинга

Конфигурационные файлы, интерфейсы, команды, директории, переменные и другие параметры, на которые следует обращать внимание, выделены:

`моноширинным шрифтом (monospace font example)`

Аннотации и ссылки выделены:

цветным курсивным шрифтом (colored italic font example)

2.2 Назначение

Документ предназначен для административного и инженерно-технического персонала, осуществляющего установку, настройку и функционирование (эксплуатацию) программного комплекса «Закупочные сервисы iSource», идентифицирует взаимозависимости программных средств, определяет порядок развертывания, описывает рекомендуемые настройки средств безопасности комплекса и среды его выполнения, а также позволяет проверить работоспособность комплекса после развертывания.

Документ не предназначен для пользователей комплекса, осуществляющих повседневную работу.

Настоящий документ отвечает некоторым элементам содержания свидетельств семейств: AGD_OPE.1 и AGD_PRE.1, согласно требований ГОСТ Р ИСО/МЭК 15408-3:2013 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности», и может быть использован при их подготовке.

2.3 Область применения

Основной областью применения настоящего документа является использование данных, изложенных в нем при развертывании и повседневной эксплуатации комплекса администратором (администратором безопасности) при осуществлении следующих процедур жизненного цикла:

- при осуществлении корректирующих действий по результатам устранения недостатков (уязвимостей);
- при осуществлении процедур поставки (обновлении);
- при осуществлении развертывания и настройки;
- при осуществлении сопровождения;
- при выводе из эксплуатации (и(или) утилизации).

3 Подготовительные процедуры

3.1 Требования к квалификации персонала

К административному и инженерно-техническому персоналу, устанавливающему и эксплуатирующему комплекс предъявляются следующие требования:

- глубокое знание, понимание и практический опыт в установке, настройке и администрировании операционных систем Linux (предпочтительный диалект – Debian, Ubuntu, Astra Linux);
- знание, понимание и практический опыт в установке, настройке и администрировании средств контейнеризации (k8s, docker, CRI-O);
- знание, понимание и практический опыт в установке, настройке и администрировании средств автоматизированного развертывания (ansible, puppet);
- знание, понимание и практический опыт в установке, настройке и администрировании сетевых и общесистемных служб в ОС Linux (NFS, DNS, NTP, т.п.);
- знание, понимание и практический опыт в установке, настройке и администрировании средств безопасности в ОС Linux – аутентификации, средств АВЗ, аудита, разграничения доступа, изоляции процессов (модули PAM, U2F, FreeIPA, KeyCloak, PAX, iptables, sshd, namspaces, capabilities, POSIX 1e extensions ACL, AppArmor/SELinux, auditd, rsyslog, journald, Kaspersky Antivirus, Доктор ВЕБ, СКЗИ Криптопро и т.п.);
- знание, понимание и практический опыт при чтении и анализе действующей документации по информационной безопасности, включая нормативные документы отечественных регуляторов (ФСТЭК РФ, ФСБ РФ, Роскомнадзор и др.), а также действующих локальных нормативных документов, регламентирующих деятельность в области информационной безопасности в организации. Рекомендуется также наличие опыта при чтении и анализе зарубежных стандартов в области ИБ.

3.2 Системные требования для компонента «Процессор»

Минимальные системные требования для развертывания и эксплуатации компонента «Процессор» представлены в таблице Таблица 1:

Модуль	Ядра ЦП	ОЗУ	Место на диске	Производительность сети	Примечание
Закупочный модуль	4 CPU (vCPU)	8 Гбайт	100 Гбайт (включая нужды ОС)	10 Гбит/сек.	–
Планирование	–	–	–	–	Кластер k8s
Camuda	–	–	–	–	Кластер k8s
Документооборот	4 CPU (vCPU)	8 Гбайт	100 Гбайт (включая нужды ОС)	10 Гбит/сек.	–
Сервис генерации документов	2 CPU (vCPU)	4 Гбайт	50 Гбайт (включая нужды ОС)	10 Гбит/сек.	–

Модуль	Ядра ЦП	ОЗУ	Место на диске	Производительность сети	Примечание
fileservice	4 CPU (vCPU)	6 Гбайт	200 Гбайт (включая нужды ОС)	10 Гбит/сек.	–
cryptoservice	4 CPU (vCPU)	8 Гбайт	50 Гбайт (включая нужды ОС)	10 Гбит/сек.	–
k8s Master	2 CPU (vCPU)	4 Гбайт	100 Гбайт	10 Гбит/сек.	–
k8s Workers	8 CPU (vCPU)	24 Гбайт	100 Гбайт	10 Гбит/сек.	–
k8s Ingress	2 CPU (vCPU)	4 Гбайт	100 Гбайт	10 Гбит/сек.	–
Итого:	**30 CPU (vCPU)**	**66 Гбайт**	**800 Гбайт (включая нужды ОС)**	**10 Гбит/сек.**	–

Таблица 1: Минимальные системные требования.

Рекомендуемые системные требования в два раза превышают минимальные.

Кроме того, для повышения качества работы программных модулей и сервисов, а также для повышения отказоустойчивости и обеспечения резервирования и защиты от сбоев, рекомендуется применять:

- аггрегацию сетевых интерфейсов (IEEE.802.3 LACP);
- использовать дисковые массивы с дисками горячей замены (hot spare);
- использовать модули ОЗУ с возможностью горячей замены;
- использовать аппаратные средства с двумя блоками питания;
- использовать специализированные аппаратные средства SAN;
- использовать в интересах резервного копирования ленточные накопители (библиотеки) стандарта не ниже LTO-6 с отдельными модулями смены кассет.

Перед установкой и настройкой комплекса необходимо, чтобы в сетевой среде, в которой осуществляется его развертывание и функционирование было обеспечено наличие не менее чем двух серверов имен (основного и резервного), а также не менее двух серверов, обеспечивающих предоставление единых меток времени по протоколу NTP (основного и резервного).

Без наличия указанных выше инфраструктурных служб (службы времени и службы имен) – штатное функционирование комплекса не гарантируется, поскольку данные службы влияют на непротиворечивую идентификацию сетевых узлов, операционных систем, обеспечивающих среду функционирования программного комплекса и получение надежных меток времени, обеспечивающих непротиворечивые данные аудита и процедуры идентификации и аутентификации пользователей и служб.

Исполняемые файлы программного комплекса предназначены для выполнения на аппаратной архитектуре x86-64 (также AMD64/Intel64/EM64T).

3.3 Предварительная подготовка

Выполнить подготовку к запуску плейбуков `ansible` на АРМ администратора. При этом считается, что на АРМ администратора установлена ОС Ubuntu 20.04 LTS (версия для десктопа), и с этого АРМ обеспечен доступ по протоколу `ssh` на серверы приложений закупочного модуля для пользователей (в примере имя пользователя `koa-support`), входящих в группу `sudo`.

От имени пользователя (для которого настроен обмен ключами `ssh` с серверами приложений модуля закупок) создать файл `requirements.txt`, описывающий установку зависимостей.

Содержимое файла `requirements.txt`:

```
ansible==5.10
ansible-lint==6.7.0
docker==4.4.4

ansible-core==2.12.5
cryptography==3.4.8
jinja2==2.11.3
netaddr==0.7.19
pbr==5.4.4
jmespath==0.9.5
ruamel.yaml==0.16.10
ruamel.yaml.clib==0.2.6
MarkupSafe==1.1.1

cachetools==5.2.0
certifi==2022.9.24
charset-normalizer==2.1.1
google-auth==2.13.0
idna==3.4
kubernetes==25.3.0
oauthlib==3.2.2
pyasn1==0.4.8
pyasn1-modules==0.2.8
python-dateutil==2.8.2
requests==2.28.1
requests-oauthlib==1.3.1
rsa==4.9
six==1.16.0
urllib3==1.26.12
websocket-client==1.4.1
```

Листинг 2: Пример содержимого файла `requirements.txt`

На управляющем узле (откуда осуществляется установка приложения) должен буть установлен `docker` и `python3.8-venv`. Для установки этих компонентов, выполнить:

```
# apt install python3.8-venv
```

Листинг 3: Пример установки зависимостей

Установка docker осуществляется из пакетов, доступных по адресу:
https://download.docker.com/linux/ubuntu/dists/focal/.

Либо, используя официальный репозиторий docker, процесс подготовки репозитория описан на странице:

https://docs.docker.com/engine/install/ubuntu/.

А в файле `/etc/docker/daemon.json` необходимо указать сетевые настройки для контейнера, в соответствии с имеющейся топологией, например:

```
{
"fixed-cidr": "172.31.0.0/16",“
"default-address-pools":
[
  {"base": "172.31.0.0/16", "size":24}
],
"registry-mirrors": ["http://spb99tp8394-04:5001"],“
insecure-registries" : ["spb99tp8394-04:5001"]
}
```

Листинг 4: Пример заполнения файла `/etc/docker/daemon.json`

Развертывание кластера kubernetes осуществляется обычным способом, описанным в руководстве по установке kubernetes.

Состав контейнеров для k8s указан в таблице Таблица 24.

На АРМ администратора скопировать каталог, содержащий эталонную копию ПО. От имени пользователя (в примере `koa-support`[1]) из каталога, содержащего эталонную копию ПО, выполнить команды:

```
$ python3 -m venv .venv
>>> source .venv/bin/activate
>>> pip install -r requirements.txt -i http://URL_внутреннего_репозитория_pypi/simple
```

Листинг 5: Пример установки окружения для автоматического развертывания ПО. Вариант 1.

В том случае, если репозиторий еще не создан (отсутствует), тогда можно выполнить альтернативный вариант развертывания:

```
$ python3 -m venv .venv
>>> source .venv/bin/activate
>>> make pip
```

Листинг 6: Пример установки окружения для автоматического развертывания ПО. Вариант 2.

3.4 Состав общесистемного и прикладного ПО

В приведенном составе общесистемного ПО не указаны средства информационной безопасности, прошедшие процедуру независимой оценки в виде сертификационных испытаний.

Состав ПО, формирующего среду выполнения программного комплекса, приведен в таблице Таблица 2:

[1] К этому времени на серверах приложения закупочного модуля уже должен быть создан пользователь `koa-support` и для него должен быть настроен доступ по `ssh` ключам.

№ п/п	Программное средство	Версия	Назначение	Сетевой порт и протокол взаимодействия
1	Ubuntu Linux	20.04.5 LTS Серверная версия	Операционная система общего назначения. Среда выполнения для средств контейнеризации.	22/TCP (SSH)
2	Alpine Linux	3.16.2 Версия Standart	Операционная система, формирующая среду выполнения внутри контейнера.	
3	Docker.io	20.10.12-0ubuntu2 20.04.1	Средство контейнеризации	2375/TCP 2376/TCP 80/TCP 8080/TCP 443/TCP
4	Docker Compose	1.25.0	Средство управления контейнерами (оркестратор)	
5	k8s	1.22.8	Средство управления контейнерами (оркестратор)	6443/TCP 2379-2380/TCP 10250/TCP 10259/TCP 10257/TCP
6	CRI-O		Средство управления контейнерами (оркестратор)	10010/TCP
7	ansible	5.10	Средство автоматизации развертывания	
8	ansible-lint	6.7.0	Средство автоматизации развертывания	
9	NFS	v.3 (v.4)	Сетевая файловая система	111/TCP 111/UDP 2049/TCP 2049/UDP 1110/TCP 1110/UDP 4045/TCP 4045/UDP
10	PostgreSQL	12.12-1.pgdg20.04+1	Реляционная СУБД	5432/TCP
11	RabbitMQ	3.10.8	Реализация протокола AMPQ	5672/TCP
12	Redis	7.0.5	Резидентная СУБД	6379/TCP
13	KeyCloak	19.0.3	Провайдер идентификации и аутентификации	80/TCP 8080/TCP 9990/TCP

Таблица 2: Состав ПО среды выполнения.

В некоторых определенных случаях при развертывании и эксплуатации программному комплексу может потребоваться доступ к программным сервисам HUB, которые могут требовать подключений к сетям общего пользования. В приведенной в настоящем руководстве информации учитывается, что сервис HUB не входит в комплект поставки.

Также, в некоторых случаях эксплуатации может потребоваться доступ к сервисам DaData, взаимодействие с которыми осуществляется по протоколу TCP/IP (прикладные протоколы HTTP/HTTPS, порт 80/443 соответственно, для обращений к API DaData), и находящимися по адресу:

https://suggestions.dadata.ru/suggestions/api/4_1/rs

Информация об обращениях к API DaData приведена в разделе 5.1.9.

Состав прикладного ПО, реализующего основные функции по назначению перечислен в таблице Таблица 3.

В некоторых случаях эксплуатации, программные компоненты комплекса требует наличия обращений к службам электронной почты (серверу электронной почты). Требуется учитывать, что рекомендованный способ взаимодействия с сервисами электронной почты – это применение собственно-

го сервера электронной почты организации (находящемся в контуре безопасности, либо иного доверенного сервера), с выделением отдельного пользователя для такого взаимодействия. Информация, о технологических параметрах для взаимодействия с сервером электронной почты приведена в разделе 5.1.9.

Сведения по идентификации наименования версий прикладного ПО приведены в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение Б. Определение жизненного цикла* is000-ALC_LCD.1 в разделе «*Порядок идентификации версий ПО*».

№ п/п	КС CRC32	Имя пакета и версия	Назначение
1	bfe85aa7	SPO_Processor_back_07_12_2022.zip	Закупочный модуль, back-end
2	eb4f214a	SPO_Processor_front_31_10_2022.zip	Закупочный модуль, front-end

Таблица 3: Состав прикладного ПО.

3.5 Состав заимствованного и привлекаемого ПО

Проверить (иначе – скопировать или создать), что в репозитории nexus содержатся зависимости, перечисленные[2] в таблицах Таблица 19, Таблица 20 и Таблица 21.

Состав контейнеров для k8s указан в таблице Таблица 24.

3.6 Меры безопасности при установке, настройке и эксплуатации

При установке, настройке и эксплуатации рекомендуется реализовывать следующие меры безопасности:

- на технических средствах (серверах), на которых выполняется программный комплекс, использовать средства доверенной загрузки, либо применять механизм загрузки UEFI SecureBoot;
- на технических средствах (серверах), на которых выполняется программный комплекс, отключать поддержку SMT, используя функции BIOS/UEFI, обеспечивая противостояние атакам типа Meltdown/Spectre;
- на технических средствах (серверах), на которых выполняется программный комплекс, включать для процессоров Intel бит XD (защиты от переполнения буфера), используя функции BIOS/UEFI, обеспечивая противодействие атакам, связанным с переполнением буфера;
- на технических средствах (серверах), на которых выполняется программный комплекс включать для процессоров AMD бит NX (защиты от переполнения буфера), используя функции BIOS/UEFI, обеспечивая противодействие атакам, связанным с переполнением буфера;

[2] В приведенном составе заимствованного и привлекаемого ПО не указаны средства информационной безопасности, прошедшие процедуру независимой оценки в виде сертификационных испытаний.

- активизировать механизмы защиты операционной системы[3], реализующие:
  - ⋆ защиту памяти;
  - ⋆ контроль целостности;
  - ⋆ идентификацию и аутентификацию;
  - ⋆ разграничение доступа;
  - ⋆ регистрацию событий безопасности (аудит);
  - ⋆ ограничение программной среды;
  - ⋆ фильтрацию сетевого потока.
- использовать межсетевые экраны, а также системы обнаружения вторжений (как минимум, на периметре инфраструктуры);
- использовать средства антивирусной защиты в операционных системах, установленных на физическом оборудовании или виртуальных машинах, в средствах виртуализации и контейнеризации и т.п.;
- использовать средства мониторинга безопасности и контроля защищенности;
- использовать в среде выполнения (в инфраструктуре) при необходимости средства, реализующие доверенный канал (VPN, TLS, IP-Sec и т.п.);
- использовать средства резервного копирования и повышения надежности и отказоустойчивости;
- применять при штатной эксплуатации комплекса пароли, с метрикой качества, определяемой требованиями, действующими в организации или в информационной системе, но, в общем случае, применять пароли длиной не менее 10 символов, с использованием алфавита не менее 70 знаков. А также не использовать в паролях словарные слова, личные данные субъектов, идентификаторы субъектов или их части, и т.п.;
- применять для подтверждения подлинности пользователей дополнительные средства аутентификации (например, использовать для администраторов ОС двухфакторную аутентификацию[4], а для пользователей – одноразовые пароли на основе времени[5]);
- руководствоваться эксплуатационной документацией;
- взаимодействовать с разработчиком при осуществлении мероприятий по отслеживанию уязвимостей и выявлении недостатков.

[3] Для ОС Ubuntu 20.04 описание данных рекомендаций по настройке её механизмов защиты, приведено в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение А. Безопасность в ОС Ubuntu.*» *is000-AGD_PRE.1*, являющееся приложением к настоящему документу.

[4] Применение описано в разделе 8.1.3 настоящего документа.

[5] Применение описано в разделе 8.1.4 настоящего документа.

4 Процедуры поставки

В настоящем разделе регламентируются процедуры поставки, которые необходимо выполнять в общем случае на этапе жизненного цикла «Поставка». Подробные сведения о жизненном цикле программного комплекса изложены в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение Б. Определение жизненного цикла*» *is000-ALC_LCD.1*.

Основанием для поставки программного комплекса является наличие действующих гражданско-правовых отношений, между поставщиком и приемщиком (договор, контракт, и т.п.).

Поставщиком является разработчик программного комплекса.

Допускается передача прав на поставку иному юридическому лицу, в том случае, если разработчик уполномочил соответствующее лицо в соответствии с требованиями действующего законодательства в области гражданско-правовых отношений.

Приемщиком может быть организация заказчика, потребителя, эксплуатанта, либо иное юридическое лицо, уполномоченное приемщиком в соответствии с требованиями действующего законодательства в области гражданско-правовых отношений.

При осуществлении процедур поставки поставщик и приемщик обязаны точно выполнять требования, изложенные в контракте (договоре), в случае, если в нем регламентируются процедуры поставки.

В ином случае, процедуры поставки необходимо выполнять руководствуясь требованиями, изложенными в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение Б. Определение жизненного цикла*» *is000-ALC_LCD.1* в разделе «*Процедуры поставки*».

5 Установка и настройка

В настоящем разделе описана установка программного комплекса, его настройка, процедуры первичного запуска и проверок, которые необходимо предпринять после установки, настройки и первичного запуска, с целью получить уверенность в том, что комплекс установился в штатном режиме и готов к обслуживанию.

В разделе приведена информация как для установки самого программного комплекса, так и для среды его выполнения (системных служб, от которых зависит функционирование комплекса).

Требования к квалификации персонала, который будет осуществлять установку, настройку и проверку функционирования программного комплекса приведены в разделе 3.1.

Подразумевается, что в среда (сеть, инфраструктура), в которую будет осуществляться установка программного комплекса, уже имеет минимум два сервера имен (DNS), которые корректно функционируют как основной (master) и резервный (slave) серверы. А также они обладают возможностью (авторитетны) выполнять корректное прямое и обратное преобразование имен для локальной зоны (локальных зон). Специальных сведений о том, как настраивать службу имен настоящий документ не содержит и не предусматривает, ввиду того, что служба имен является типовым сервисом, а инженерно-технический (или административный) персонал обязан обладать необходимыми навыками по настройке указанной службы.

Подразумевается, что в среде (сети, инфраструктуре), в которой будет осуществлена установка программного комплекса, уже имеется минимум два сервера времени (NTP), которые корректно функционируют как основной и резервный серверы службы времени, возможно с разными значениями переменной `stratum`. Требований к функционированию комплекса в режиме реального времени не предусмотрено, но, тем не менее, требуется, чтобы разница во времени между эталонными значениями времени на серверах времени и на узлах кластера, где будет разворачиваться программный комплекс была бы как можно меньше (не более 30-ти секунд). Специальных сведений о том, как настраивать службу времени настоящий документ не содержит и не предусматривает, ввиду того, что служба времени также является типовым сервисом, а инженерно-технический (или административный) персонал обязан обладать необходимыми навыками по её настройке.

Также в настоящем документе отсутствуют сведения по установке операционной системы Ubuntu Linux 20.04 LTS, так как инженерно-технический (или административный) персонал обязан обладать необходимыми навыками по её установке и настройке. Сведения, касающиеся её установки доступны по ссылкам:

https://ubuntu.com/tutorials/install-ubuntu-server#1-overview;

https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview.

Порядок настройки операционной системы и рекомендации по безопасности, изложены в Приложении А к настоящему документу: «*Закупочные сервисы iSource. Руководство администратора. Приложение А. Безопасность в ОС Ubuntu*» *is000-AGD_PRE.1*.

Перед тем, как производить установку и настройку настоятельно рекомендуется изучить материал, изложенный в указанном Приложении А к настоящему документу.

Рекомендуется создать и настроить локальную (изолированную) копию репозитория ОС Ubuntu 20.04 LTS в сети предприятия, эксплуатирующего программный комплекс. Процесс настройки такового описан по ссылкам ниже:

https://askubuntu.com/questions/170348/how-to-create-a-local-apt-repository.

https://linuxconfig.org/how-to-create-a-ubuntu-repository-server.

Используя конфигурационный файл `/etc/apt/sources.list` в ОС Ubuntu 20.04, настроить получение пакетов ПО из собственного источника. Процесс добавления (изменения) источников ПО описан по ссылке:

https://help.ubuntu.com/stable/ubuntu-help/addremove-sources.html.en

Перед каждой установкой любого пакета рекомендуется производить опрос и установку обновлений, так как показано на примере (работа с обновлениями и источниками установки требует нахождения в контексте полномочий `root`):

```
# apt-get update
# apt-get upgrade
```

Листинг 7: Пример команд проверки и установки обновлений в ОС Ubuntu 20.04 LTS

Установка выполняется последовательно. Для установки необходимо обладать полномочиями администратора `root`.

- необходимо подготовить набор `Inventory` файлов для соответствующей версии `Kubespray`. Шаблон инвентаря необходимо использовать аналогичный тому, что приведен по ссылке из `Kubespray`:

  https://github.com/kubernetes-sigs/kubespray/tree/master/inventory/sample.

- затем необходимо разместить подготовленный набор в локальном хранилище наборов (репозитории), чтобы обеспечить доступность и хранение наборов `Inventory` файлов для всей инфраструктуры централизованно (в одном месте).

  Необходимая документация по параметрам инвентаря приведена по ссылке ниже:

  https://github.com/kubernetes-sigs/kubespray/tree/master/docs

- основные параметры переменных и настройки, которые необходимо учесть:

  - ⋆ версия Kubernetes - `kube_version`;
  - ⋆ имя кластера - `cluster_name`;
  - ⋆ режим `kube proxy` - `kube_proxy_mode`: `ipvs`;

⋆ `Container runtime - container_manager: cri-o`. Настройки осуществляются согласно документации, приведенной по ссылке:

https://github.com/kubernetes-sigs/kubespray/blob/master/docs/cri-o.md;

⋆ значение переменной: `ingress_nginx_enabled: true` с учетом нужных параметров для развертывания `Nginx insgress controller` в новый (устанавливаемый) кластер.

- загрузить необходимые для `Kubernetes` версии основных образов контейнеров из основного хранилища (репозитория) в локальное хранилище (репозиторий) `Docker registry (Harbor)`, для изолированной установки.
- убедиться в том, что все узлы кластера имеют FQDN имена, а распознавание имен обеспечивается при запросах как из прямой, так и из обратной зоны при выполнении DNS-запросов;
- убедиться в том, что на узлах, обслуживающих кластер k8s сконфигурирован NTP клиент[6] *is000-AGD_-PRE.1* на выполнение запросов к двум (основной и резервный) серверам эталона времени, и выставлен действительный часовой пояс;
- убедиться в том, что учетная запись под которой будут выполняться автоматизированные сценарии развертывания (плейбуки, playbooks) имеет полномочия для повышения привилегий с помощью `sudo`[7] на всех узлах кластера;
- убедиться в том, что для нужд кластера (под директорию `CRI-O`) выделен отдельный диск или том, и что указанный отдельный диск (том) монтируется в директорию `/var/lib/containers`. Минимальный объем диска (тома) составляет 50 Гбайт, без учета накладных расходов ОС;
- убедиться в том, что на всех узлах k8s Workers установлены пакеты и запущены необходимые службы для работы с сетевыми хранилищами, поддерживающие протоколы NFS и iSCSI[8].

5.1 Автоматизированное развертывание с помощью `Ansible`

5.1.1 Общие сведения об автоматизации развертывания

Основным инструментом, обеспечивающим автоматизированное развертывание программного комплекса является система управления конфигурацией с открытым исходным кодом - `Ansible`, реализованная на языке `Python`, разработанная и впервые предложенная компанией Red Hat.

[6] Пример конфигурации клиента NTP для ОС Ubuntu 20.04 LTS приведен в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение А. Безопасность в ОС Ubuntu*»

[7] Для этого пользователь должен быть добавлен в группы `sudo` и `wheel`, либо быть непосредственно определен в файле конфигурации `/etc/sudoers`.

[8] В том случае, если планируется для нужд кластера выделяются не физические, а виртуализированные ресурсы

Документация на Ansible приведена по ссылке:

https://docs.ansible.com/.

Система, в своей работе использует шаги (jobs), декларативно описанные на языке разметки YAML. Шаги объединяются в структуры, именуемые плейбуками (playbooks).

```
---
- hosts: kube_control_plane
  become: yes
  tasks:
    - name: store kubeconfig
      ansible.builtin.fetch:
        src: /etc/kubernetes/admin.conf
        dest: "{{ playbook_dir }}/../../data/special/"
        flat: yes
      run_once: true

    - name: Replace api server url
      ansible.builtin.lineinfile:
        path: "{{ playbook_dir }}/../../data/special/admin.conf"
        regexp: 'https://127.0.0.1:{{ loadbalancer_apiserver_port }}'
        line: '    server: https://test-k8s-master-1:{{ loadbalancer_apiserver_port }}'
      delegate_to: controller
      run_once: true
      become: no
```

Листинг 8: Пример плейбука kubespray.yaml

Целевые узлы, на которых будет производиться работа Ansible, описываются в т.н. объекте Inventory (инвентарь, учетная единица).

Пример Inventory используемого Ansible приведен на листинге ниже:

```
---

all:
  hosts:
    controller:
      ansible_host: localhost
      ansible_connection: local

    test-k8s-master-1:
      ansible_host: xxx.250.99.184
    test-k8s-master-2:
      ansible_host: xxx.250.16.166
    test-k8s-master-3:
      ansible_host: xxx.201.154.186
    test-k8s-worker-1:
      ansible_host: xxx.250.111.244
    test-k8s-worker-2:
      ansible_host: xxx.84.121.161
    test-k8s-worker-3:
      ansible_host: xxx.84.120.181

    test-nexus:
```

```
      ansible_host: xxx.250.22.93
  children:
    kubernetes_apps:
      hosts:
        controller:

    nexus:
      hosts:
        depositionlabs-nexus:
```

Листинг 9: Пример описания объекта Inventory `inventory.yml`

Наборы задач `ansible`, которые выполняют настройку определенного компонента инфраструктуры логически объединяются в объект `Role` (роль).

Данные роли можно многократно использовать повторно, задавая разные входные значения переменных, которые помогают сохранить читаемость инфраструктурного кода. По этому каждый логический компонент в программном комплексе устанавливается с использованием соответствующей роли.

Хорошим примером проекта, использующего вышеописанные структуры `Ansible` является рекомендуемое решение для установки `Kubernetes` - `Kuberspray` (так же используемое в рамках развертывания программного комплекса).

Пример запуска `Kubespray` из плейбуков программного комплекса приведен на листинге ниже:

```
---

- name: Starting kubespray
  become: yes
  ansible.builtin.import_playbook: "{{ playbook_dir }}/../../additional_roles/kubespray/cluster.yml"
```

Листинг 10: Пример запуска `Kyberspray` из `kubespray.yml`

5.1.2 Роли в `Ansible`

5.1.2.1 Роль `helm-packages-installer`

Роль, специально подготовленная (реализованная) для развертывания программного комплекса:

`helm-packages-installer` – роль для установки `kubernetes-like` приложений (как системных, например `kube-prometheus-stack` или `ingress-nginx`), так и бизнес приложений.

Данная роль использует `helm` и коллекцию `Ansible kubernetes.core`.

Роль `helm-packages-installer` имеет следующий набор управляемых значений:

```
---

# helm repositories for update
helm_pi_repositories: ""
  # - name: repository-name
  #   repo_url: https://repos.loc/repo
```

```
# helm releases for upgrade
helm_pi_releases: ""
  # - name: repository-name
  #   chart_ref: https://repos.loc/repo
  #   release_namespace: test # omit
  #   create_namespace: false # omit
  #   version: # omit
  #   value: "" # omit
```

Листинг 11: Роль `helm-packages-installer` в файле `./default/main.yml`. Пример описания.

5.1.2.2 Роль nexus-docker-installer

Роль `nexus-docker-installer` предназначена для развертывания хранилища артефактов `Sonatype Nexus`, и используется при развертывании программного комплекса как точка хранения необходимых артефактов (пакетный, `docker`, `helm`, `pipl` репозиторий), а также для предварительной настройки и инициализации хранилища.

Роль `nexus-docker-installer` имеет следующий набор управляемых значений:

```
---
nexus_di_package_requirements:
  - "docker.io"

nexus_di_docker_image: "sonatype/nexus3:3.40.0"
nexus_di_service_name: nexus

nexus_di_env_variables:
  - var: INSTALL4J_ADD_VM_PARAMS
    val: '-Xms2703m -Xmx2703m -XX:MaxDirectMemorySize=2703m -Djava.util.prefs.userRoot=/nexus-data/
    javaprefs'

nexus_di_data_path: /data/nexus-data
nexus_di_user: nexus
nexus_di_group: nexus
nexus_di_uid: 200
nexus_di_gid: 200

nexus_di_expose_web_port: 8081
```

Листинг 12: Роль `nexus-docker-installer` в файле `./default/main.yml`. Пример описания.

5.1.2.3 Роль Kubespray (внешняя)

`Kubespray` - роль, реализованная для работы с `Ansible`, которая позволяет произвести полный цикл установки, настройки и обеспечения работы `Kubernetes` кластера. В случае использования в интересах программного комплекса, `Kubespray` настраивается через переменные окружения и `Inventory`, и запускается отдельно в рамках его работы.

Пример `Inventory` для использования под ролью `Kubespray`:

```
kubespray:
  children:
    k8s_cluster:
    etcd:

calico_rr:

kube_control_plane:
  hosts:
    test-k8s-master-1:
    test-k8s-master-2:
    test-k8s-master-3:

kube_node:
  hosts:
    test-k8s-worker-1:
    test-k8s-worker-2:
    test-k8s-worker-3:

k8s_cluster:
  children:
    kube_control_plane:
    kube_node:
    calico_rr:

etcd:
  children:
    kube_control_plane:
```

Листинг 13: Пример `Inventory` для использования под ролью `Kubespray` в файле `inventory.yml`

5.1.3 Структуры данных для автоматизированного развертывания

Структура данных для автоматизированного развертывания продемонстрирована на листинге ниже:

```
inventories/koa
├── group_vars
│   ├── all
│   │   ├── artifacts
│   │   │   ├── nexus.yml
│   │   │   ├── preprovision.yml
│   │   │   └── transport.yml
│   │   ├── connection.yml
│   │   └── targets.yml
│   ├── app_inspector.yml
│   ├── app_processor_documents.yml
│   ├── app_processor_market.yml
│   ├── app_processor_planning.yml
│   ├── k8s_cluster
│   │   ├── addons.yml
│   │   ├── k8s-cluster.yml
│   │   ├── k8s-net-calico.yml
│   │   └── preprovision.yml
│   │
```

```
│   ├── kafka.yml
│   ├── keycloak.yml
│   ├── kube_control_plane.yml
│   ├── kubernetes_apps
│   │   └── repos.yaml
│   ├── kubespray
│   │   ├── all.yml
│   │   ├── cri-o.yml
│   │   ├── etcd.yml
│   │   ├── oci.yml
│   │   └── offline.yml
│   ├── mysql.yml
│   ├── nexus
│   │   └── nexus.yml
│   ├── nfs.yml
│   ├── postgres.yml
│   ├── rabbitmq.yml
│   └── redis.yml
├── host_vars
│   ├── controller.yml
│   └── processor.yml
└── inventory.yml
```

Листинг 14: Структура данных для автоматизированного развертывания с помощью `ansible`

Примеры содержимого для заполнения каждого файла в структуре данных указаны ниже по тексту.

5.1.4 Общие переменные

В настоящем разделе содержится описание переменных и структур данных, которые необходимы в интересах всего кластера. Эти переменные и структуры данных влияют на формирование программного окружения и среду функционирования.

5.1.4.1 Данные структуры `inventory.yml`

Пример содержимого для структуры данных `inventories/koa/inventory.yml` приведен на листинге ниже. Необходимо заполнить инвентарь в соответствии с ролями и адресацией узлов, например:

```
---

all:
  hosts:
    controller:
      ansible_host: localhost
      ansible_connection: local
    nexus:
      ansible_host: spb99tp8394-04
    processor:
      ansible_host: spb99tp8394-01
    redis:
```

```
      ansible_host: spb99tp8394-02
    rabbitmq:
      ansible_host: spb99tp8394-05
    keycloak:
      ansible_host: spb99tp8394-06
    postgres:
      ansible_host: spb99tp8394-03

    kub-1:
      ansible_host: spb99tp8394-07
    kub-2:
      ansible_host: spb99tp8394-08
    kub-3:
      ansible_host: spb99tp8394-11
    kub-4:
      ansible_host: spb99tp8394-10
    kub-5:
      ansible_host: spb99tp8394-20
    kub-6:
      ansible_host: spb99tp8394-21

    nfs:
      ansible_host: spb99tp8394-12

    kafka:
      ansible_host: spb99tp8394-05

    mysql:
      ansible_host: spb99tp8394-05

  children:
    keycloak:
      hosts:
        keycloak:

    mysql:
      hosts:
        mysql:

    kafka:
      hosts:
        kafka:

    redis:
      hosts:
        rabbitmq:

    postgres:
      hosts:
        postgres:

    rabbitmq:
      hosts:
        rabbitmq:

    app_processor_market:
      hosts:
        processor:
```

```
nexus:
  hosts:
    nexus:

kubespray:
  children:
    k8s_cluster:
    etcd:

calico_rr:

kube_control_plane:
  hosts:
    kub-1:
    kub-2:
    kub-3:
kube_node:
  hosts:
    kub-4:
    kub-5:
    kub-6:

k8s_cluster:
  children:
    kube_control_plane:
    kube_node:
    calico_rr:

etcd:
  children:
    kube_control_plane:

kubernetes_apps:
  hosts:
    kub-1:

app_processor_planning:
  hosts:
    kub-1:

app_processor_documents:
  hosts:
    kub-1:

app_inspector:
  hosts:
    kub-1:
```

Листинг 15: Пример заполения структуры данных `inventory.yml`

5.1.4.2 Данные структуры `targets.yml`

Пример содержимого для структуры данных `inventories/koa/group_vars/all/targets.yml` приведен на листинге ниже. Необходимо проверить групповые переменные, и, при необходимости,

заполнить их, в соответствии с информацией, указанной в инвентаре. Обычно редактирование этих групповых переменных не требуется, например:

```
---

all_host_nexus_repository: "{{ hostvars['nexus']['ansible_host'] }}"
all_host_redis: "{{ hostvars['redis']['ansible_host'] }}"
all_host_rabbitmq: "{{ hostvars['rabbitmq']['ansible_host'] }}"
all_host_postgres: "{{ hostvars['postgres']['ansible_host'] }}"
all_host_keycloak: "{{ hostvars['keycloak']['ansible_host'] }}"
all_host_nfs: "{{ hostvars['nfs']['ansible_host'] }}"

all_nfs_server_path: "/srv/nfs"

all_target_planning_host: planning.isource.ru
all_target_docs_host: cmcenter.isource.ru

all_target_inspector_cabinet: cabinet.inspector
all_target_inspector_api: api.inspector
all_target_inspector_client: client.inspector
```

Листинг 16: Пример заполения структуры данных `targets.yml`

5.1.4.3 Данные структуры `connection.yml`

Структура данных `inventories/koa/group_vars/all/connection.yml` используется для автоматизации подключения к узлам, на которых осуществляется развертывание. Пример содержимого для структуры данных `inventories/koa/group_vars/all/connection.yml` приведен на листинге ниже. Необходимо проверить групповые переменные, и, при необходимости, заполнить их, в соответствии с информацией, указанной в инвентаре. Обычно редактирование этих групповых переменных не требуется, например:

```
---

ansible_user: "koa-support"
become_user: "root"
ansible_connection: "ssh"
host_key_checking: false
ansible_ssh_private_key_file: "/home/koa-support/.ssh/id_ed25519"
```

Листинг 17: Пример заполения общей структуры данных `connection.yml`

5.1.4.4 Данные структуры `preprovision.yml`

Структура `inventories/koa/group_vars/all/artifacts/preprovision.yml` также используется для автоматизации подключения к узлам, на которых осуществляется развертывание. Указанная структура данных содержит параметры настройки общее для всех узлов. При

необходимости нужно проверить и изменить параметры сетевые параметры в секции `hosts_p_docker_daemon_json`. Пример содержимого для структуры данных `inventories/koa/group_vars/all/artifacts/preprovision.yml` приведен на листинге ниже:

```
---
hosts_p_apt_repositories:
  name: "{{ all_apt_repository_apt_repo.name }}"
  key: "{{ all_apt_repository_apt_repo.keypair }}"
  uri: "{{ all_apt_repository_apt_repo.uri }}"
  id: "{{ all_apt_repository_apt_repo.id }}"

hosts_p_docker_daemon_json: |
  {
    "fixed-cidr": "172.31.0.0/16",
    "default-address-pools":
    [
      {"base":"172.31.0.0/16","size":24}
    ],
    "registry-mirrors": ["http://{{ hosts_p_docker_registry }}"],
    "insecure-registries" : ["{{ hosts_p_docker_registry }}"]
  }

hosts_p_pip_registry: "http://{{ all_host_nexus_repository }}:8081/repository/pypi-onpremise/simple"
hosts_p_pip_registry_short: "{{ all_host_nexus_repository_files }}"
```

Листинг 18: Пример заполения общей структуры данных `preprovision.yml`

5.1.4.5 Данные структуры `nexus.yml`

Структура данных `inventories/koa/group_vars/all/artifacts/nexus.yml` используется в интересах общего репозитория, содержащего все т.н. «артефакты», то есть все необходимые бинарные модули, библиотеки и компоненты. При необходимости нужно проверить и изменить параметры имени и пароля. Пример содержимого для структуры данных `inventories/koa/group_vars/all/artifacts/nexus.yml` приведен на листинге ниже:

```
---

# Nexus apt repositories variables

nexus_di_admin_login: admin
nexus_di_admin_password: password

# далее отредактировать не обязательно
hosts_p_docker_registry: "{{ artifacts_tl_docker_registry }}"
all_host_nexus_repository_files: "{{ all_host_nexus_repository }}:8081"

all_apt_repository_apt_repo:
    name: apt-repo
    passphrase: "password"
    id: "12A89BA4B9E964B6151DF300EA35025986CBE4F9"
```

```
    uri: "http://{{ all_host_nexus_repository }}:8081/repository/apt-repo"
    keypair: |
      -----BEGIN PGP PRIVATE KEY BLOCK-----

содержимое сертификата gpg 1

содержимое сертификата gpg 2

      -----END PGP PRIVATE KEY BLOCK-----

nexus_di_hosted_npm_repo:
    name: npm-onpremise
    uri: "http://{{ all_host_nexus_repository }}:8081/service/rest/v1/components?repository=npm-
    onpremise"
    writePolicy: allow_once

nexus_di_hosted_pypi_repo:
  uri: "http://{{ all_host_nexus_repository }}:8081/service/rest/v1/components?repository=pypi-
    onpremise"
  name: pypi-onpremise
  writePolicy: ALLOW

nexus_di_hosted_raw_repo:
  uri: "http://{{ all_host_nexus_repository }}:8081/service/rest/v1/components?repository=raw-
    onpremise"
  name: raw-onpremise

nexus_di_hosted_helm_repo:
  uri: "http://{{ all_host_nexus_repository }}:8081/service/rest/v1/components?repository=helm-
    onpremise"
  name: helm-onpremise
```

Листинг 19: Пример заполения структуры данных для `nexus.yml`

5.1.5 Структуры данных, для приложений (переменные, специфичные для конкретного ПО)

В настоящем разделе содержится описание переменных и структур данных, которые необходимы в интересах функциональных приложений. Эти переменные и структуры данных влияют на формирование параметров функциональных программных модулей.

5.1.5.1 Данные структуры `redis.yml`

Структура данных `inventories/koa/group_vars/redis.yml` используется в интересах REDIS (резидентной системы управления базами данных класса NoSQL). При необходимости, можно изменить образ контейнера `docker` или путь до каталога с приложением. Пример содержимого для структуры данных
`inventories/koa/group_vars/redis.yml`
приведен на листинге ниже:

```
---
hosts_p_docker_provision: true
```

```
redis_d_compose_path: /var/redis-docker
redis_d_image: "{{ hosts_p_docker_registry }}/redis:7.0.5"
redis_d_requirements:
  - docker.io
  - docker-compose

redis_d_expose_port: 6379
```

Листинг 20: Пример заполения структуры данных для `redis.yml`

5.1.5.2 Данные структуры `rabbitmq.yml`

Структура данных `inventories/koa/group_vars/rabbitmq.yml` используется в интересах RabbitMQ (программного брокера сообщений на основе стандарта AMQP). При необходимости, можно изменить пару логин:пароль и параметр `vhost` по которым к RabbitMQ будут обращаться приложения. Пример содержимого для структуры данных `inventories/koa/group_vars/rabbitmq.yml` приведен на листинге ниже:

```
---
hosts_p_docker_provision: true
hosts_p_apt_packages:
  - docker.io
  - docker-compose

rabbitmq_d_compose_path: /var/rabbitmq-docker
rabbitmq_d_image: "{{ hosts_p_docker_registry }}/rabbitmq:3.10-management"

#rabbitmq_d_requirements:
#  - docker.io
#  - docker-compose

rabbitmq_d_expose_port: 5672

rabbitmq_d_auth:
  user: rabbitmq
  pass: rabbitmq

rabbitmq_d_vhosts:
  - name: processor
    user: processor
    configure: .*
    write: .*
    read:  .*
  - name: sed
    user: sed
    configure: .*
    write: .*
    read:  .*
  - name: "%2f"
    user: sed
    configure: .*
    write: .*
```

```
    read:  .*
  - name: "%2f"
    user: processor
    configure: .*
    write: .*
    read:  .*

rabbitmq_d_users:
  - name: processor
    password: processor
    tag: processor
  - name: sed
    password: sed
    tag: sed
```

Листинг 21: Пример заполения структуры данных для `rabbitmq.yml`

5.1.5.3 Данные структуры `postgres.yml`

Структура данных `inventories/koa/group_vars/postgres.yml` используется в интересах СУБД PostgreSQL, в том случае, если она развертывается в контейнере. Развертывать СУБД PostreSQL в контейнере – это рекомендуемый способ развертывания, используемый по умолчанию. В противном случае (если развертывать СУБД в контейнере не целесообразно, или не требуется), см. информацию, приведенную в разделе 5.2.3.1. При необходимости, можно изменить пару логин:пароль, по которым к СУБД будут обращаться приложения. Пример содержимого для структуры данных
`inventories/koa/group_vars/postgres.yml`
приведен на листинге ниже:

```
---
hosts_p_docker_provision: true

postgres_d_compose_path: /var/postgresql-docker
postgres_d_data_path: /var/postgresql-docker/postgres
postgres_d_image: "{{ hosts_p_docker_registry }}/postgres:10"
postgres_d_requirements:
  - docker.io
  - docker-compose

postgres_d_expose_port: 5432

postgres_d_environments:
  - name: POSTGRES_USER
    value: processor
  - name: POSTGRES_PASSWORD
    value: processor
  - name: POSTGRES_DB
    value: processor
```

Листинг 22: Пример заполения структуры данных для `postgres.yml`

5.1.5.4 Данные структуры `keycloak.yml`

Структура данных `inventories/koa/group_vars/keycloak.yml` используется в интересах единого провайдера, обеспечивающего аутентификацию и идентификацию, в том случае, если `keycloak` развертывается в контейнере. Развертывать KeyCloak в контейнере – это рекомендуемый способ развертывания, используемый по умолчанию. В противном случае (если развертывать `keycloak` в контейнере не целесообразно, или не требуется), см. информацию, приведенную в разделе 5.2.3. При необходимости, можно изменить параметры СУБД и KeyCloak для их совместной работы после установки. Пример содержимого для структуры данных `inventories/koa/group_vars/keycloak.yml` приведен на листинге ниже:

```
---
hosts_p_docker_provision: true
hosts_p_apt_packages:
  - docker.io

keycloak_d_compose_path: /var/keycloak-docker
keycloak_d_image: "{{ hosts_p_docker_registry }}/bitnami/keycloak:19.0.3"
keycloak_d_requirements:
  - docker.io
  - docker-compose
  - nginx=1.18*

keycloak_d_expose_port: 8080

keycloak_d_admin: admin
keycloak_d_pass: Password5

keycloak_d_environments:
  - name: KC_HTTP_ENABLED
    value: "true"
  - name: KEYCLOAK_DATABASE_HOST
    value: postgres-keycloak
  - name: KEYCLOAK_DATABASE_NAME
    value: "keycloak"
  - name: KEYCLOAK_DATABASE_USER
    value: "keycloak"
  - name: KEYCLOAK_DATABASE_PASSWORD
    value: "keycloak230jf8rejerf"
  - name: KEYCKOAK_HTTP_PORT
    value: "{{ keycloak_d_expose_port }}"
  # - name: PROXY_ADDRESS_FORWARDING
  #   value: "true"

keycloak_d_network: default

postgres_d_k_data_path: "{{ keycloak_d_compose_path }}/postgres"
postgres_d_k_image: "{{ hosts_p_docker_registry }}/postgres:10"

postgres_d_k_expose_port: 5433

postgres_d_k_environments:
```

```
  - name: POSTGRES_USER
    value: keycloak
  - name: POSTGRES_PASSWORD
    value: keycloak230jf8rejerf
  - name: POSTGRES_DB
    value: keycloak
```

Листинг 23: Пример заполения структуры данных для `keycloak.yml`

5.1.5.5 Данные структуры `main.yml`

Структура данных `main.yml` обеспечивает единый шаблон развертывания. Пример содержимого для структуры данных `main.yml` приведен на листинге ниже:

```
---

- import_playbook: playbooks/common/nexus.yml
  tags:
    - ignite
    - nexus

- import_playbook: playbooks/package/artifacts.yml
  tags:
    - ignite
    - artifacts

- import_playbook: playbooks/package/artifacts-kubespray.yml
  tags:
    - ignite
    - artifacts-kubernetes

- import_playbook: playbooks/common/preprovision.yml
  tags:
    - ignite
    - preprovision

- import_playbook: playbooks/common/redis.yml
  tags:
    - ignite
    - redis

- import_playbook: playbooks/common/rabbitmq.yml
  tags:
    - ignite
    - rabbitmq

- import_playbook: playbooks/common/keycloak.yml
  tags:
    - ignite
    - keycloak

- import_playbook: playbooks/common/postgres.yml
  tags:
    - ignite
    - postgres

- import_playbook: playbooks/common/mysql.yml
```

```
  tags:
    - ignite
    - mysql

- import_playbook: playbooks/common/kafka.yml
  tags:
    - ignite
    - kafka

- import_playbook: playbooks/kubernetes/nfs.yml
  tags:
    - ignite
    - nfs

- import_playbook: playbooks/kubernetes/helm-applications.yml
  tags:
    - ignite
    - infra-kubernetes

- import_playbook: playbooks/apps/processor-planning.yml
  tags:
    - ignite
    - app-processor-planning

- import_playbook: playbooks/apps/processor-market.yml
  tags:
    - ignite
    - app-processor-market

- import_playbook: playbooks/apps/applications_healthchecks.yml
  tags:
    - ignite
    - check
```

Листинг 24: Пример заполения структуры данных для `main.yml`

5.1.5.6 Запуск окружения

Затем необходимо инициализировать поэтапный запуск окружения. Запуск производится следующей командой:

```
ansible-playbook main.yml -i inventories/dev/inventory.yml -t аргумент
```

Листинг 25: Пример команды для запуска окружения

Где «`аргумент`» это обозначенное значение в поле тега (`tags:`), приведенное в примере структуры данных `main.yml` в параграфе 5.1.5.5.

Например, для запуска установки и настройки хранилища артефактов `nexus` выполнить:

```
ansible-playbook main.yml -i inventories/dev/inventory.yml -t nexus
```

Листинг 26: Пример команды для запуска `nexus`

Для инициализации загрузки артефактов в хранилище артефактов `nexus` выполнить:

```
ansible-playbook main.yml -i inventories/dev/inventory.yml -t artifacts
```

Листинг 27: Пример команды для инициализации загрузки артефактов в хранилище

Для инициализации предварительной настройки узлов (формирования среды выполнения приложений):

```
ansible-playbook main.yml -i inventories/dev/inventory.yml -t preprovision
```

Листинг 28: Пример команды для предварительной настройки узлов

Для установки и настройки СУБД Redis:

```
ansible-playbook main.yml -i inventories/dev/inventory.yml -t redis
```

Листинг 29: Пример команды для установки и настройки СУБД Redis

Для установки и настройки ПО RabbitMQ:

```
ansible-playbook main.yml -i inventories/dev/inventory.yml -t rabbitmq
```

Листинг 30: Пример команды для установки и настройки ПО RabbitMQ

Для установки и настройки СУБД PostgreSQL:

```
ansible-playbook main.yml -i inventories/dev/inventory.yml -t postgres
```

Листинг 31: Пример команды для установки и настройки СУБД PostgreSQL

Для установки и настройки единого провайдера идентификации и аутентифкации KeyCloak выполнить:

```
ansible-playbook main.yml -i inventories/dev/inventory.yml -t keycloak
```

Листинг 32: Пример команды для установки и настройки KeyCloak

5.1.6 Интеграция KeyCloak и приложения (закупочного модуля)

5.1.6.1 Создание технологического клиента, пользователей и ассоциация их с ролями

После запуска окружения потребуется обеспечить связность между приложением и провайдером аутентификации. Для этого для приложения в провайдере KeyCloak подготавливается соответствующее описание, создаются клиент, роли, пользователи, производится их взаимная ассоциация (назначение), а также инициализируется набор ключей, и устанавливаются атрибуты. Ключ приложения содержится по адресу: *http://<адрес (URL) KeyCloak>/auth/admin/master/console/#/master/realm-settings/keys*.

Интеграция KeyCloak в интересах приложения (приложений) выполняется следующим образом (потребуется доступ администратора KeyCloak):

- выполнить вход в KeyCloak от имени администратора и выбрать в меню «`Clients`» – «`Create client`». Пример операции изображен на рисунке Рисунок 1:

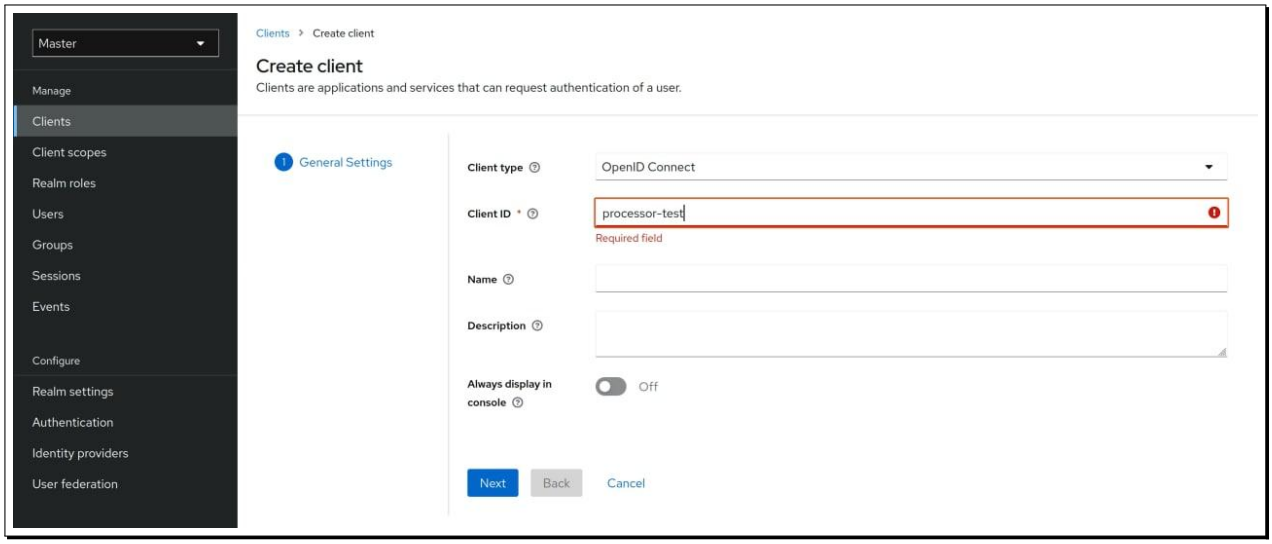


Рисунок 1: Пример создания технологического клиента для служб приложения

- нажав «Next» переключится на следующий экран. Изменять значения по умолчанию не требуется, на следующем экране нажать «Save». Пример операции изображен на рисунке Рисунок 2:

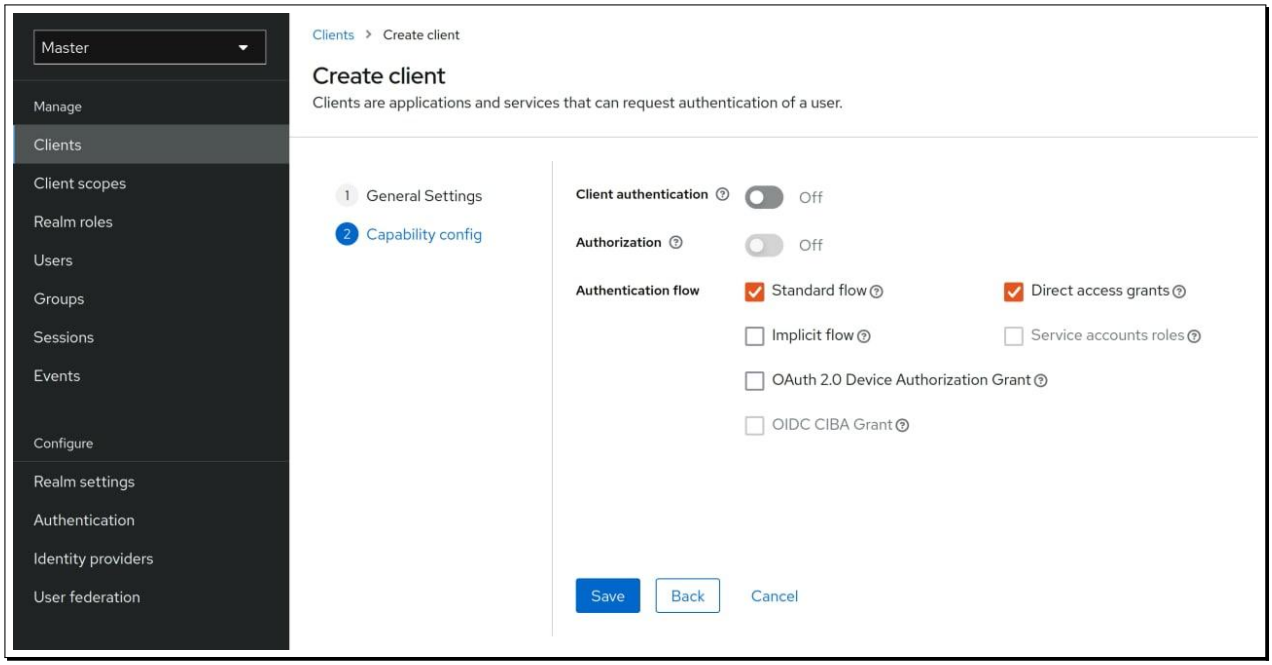


Рисунок 2: Пример создания технологического клиента для служб приложения. Продолжение.

- затем осуществить выбор созданного на предыдущем этапе клиента (для этого нажать на его имя) и заполнить данные для клиента. Образец совершения операции приведен на рисунках Рисунок 3 и Рисунок 4:

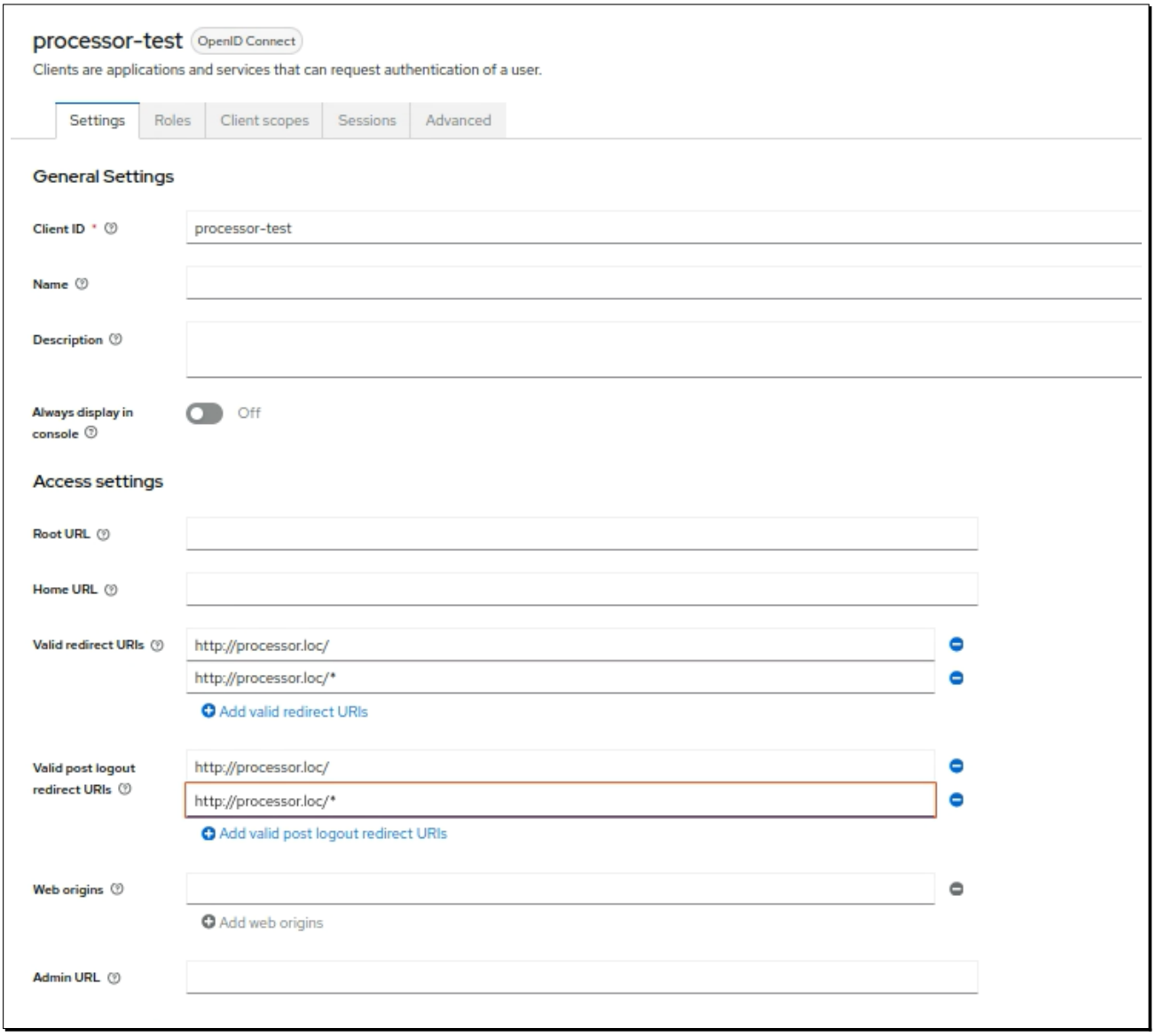


Рисунок 3: Пример заполнения данных клиента для служб приложения

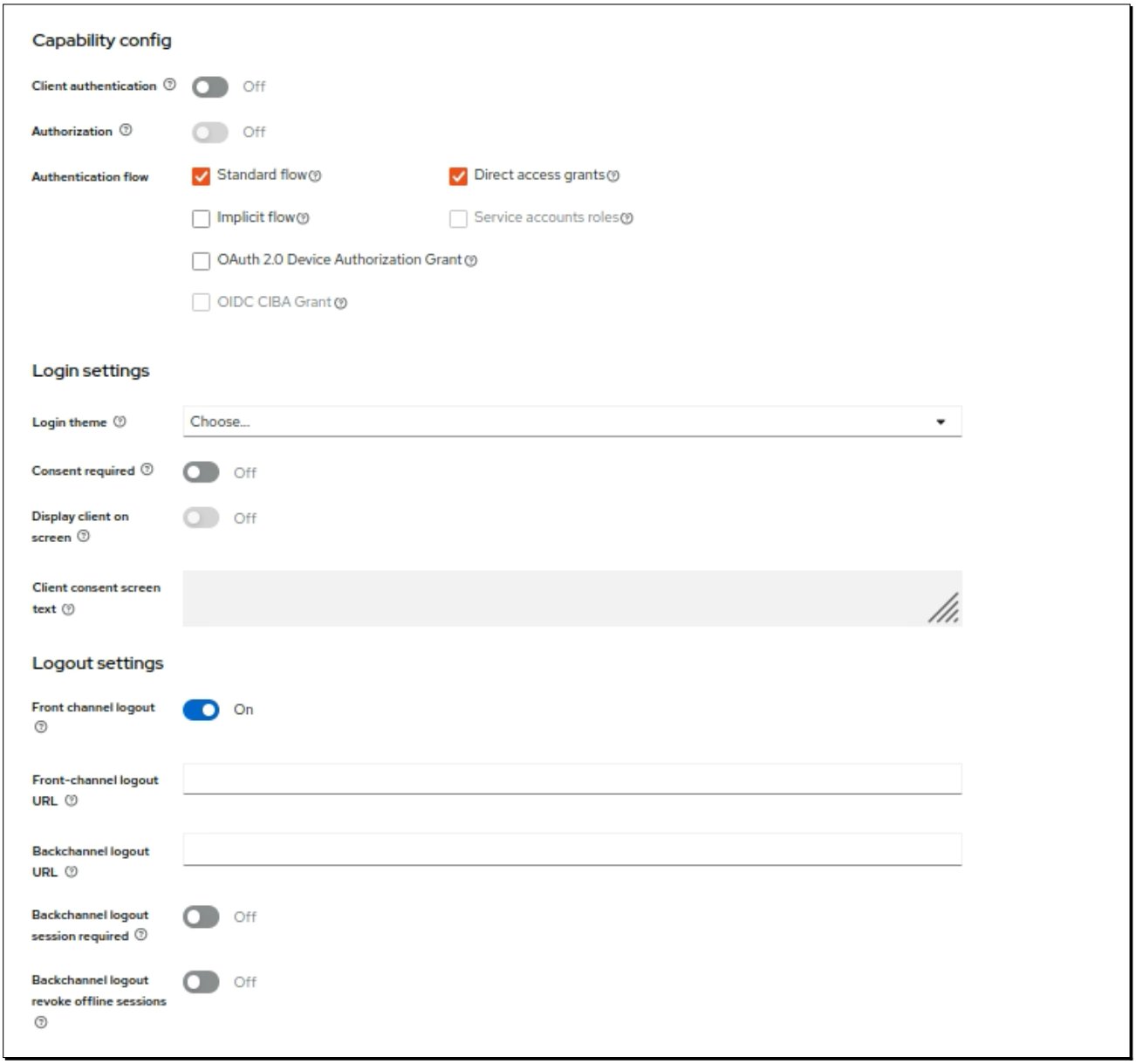


Рисунок 4: Пример заполнения данных клиента для служб приложения. Продолжение.

Где:

- ⋆ параметр «`Valid redirect URIs`» соответствует адресу (URL) приложения с маской (`*`), см. рисунок Рисунок 3;
- ⋆ параметр «`Valid post logout redirect URIs`» соответствует адресу (URL) приложения с маской (`*`), см. рисунок Рисунок 3;
- ⋆ параметр (ползунок) «`Backchannel logout session`» установить в положение «`Off`», см. рисунок Рисунок 4.

- используя вкладку «`Roles`», создать следующие роли:
  - ⋆ «`gpnmarket_backoffice_buyer`»;
  - ⋆ «`gpnmarket_backoffice_finance_approver`»;

- «gpnmarket_backoffice_observer»;
- «gpnmarket_customer_approver»;
- «gpnmarket_customer_buyer»;
- «gpnmarket_senior_backoffice»;
- «gpnmarket_system_admin».

Пример совершения операции по созданию ролей приведен на рисунке Рисунок 5:

Рисунок 5: Пример совершения операции по созданию ролей

- добавить пользователя для закупочного модуля. Для этого перейти во вкладку «Users» и с помощью кнопки «Add users» добавить пользователя и назначить ему пароль. Пример совершения операций приведен на рисунках Рисунок 6 и Рисунок 7:

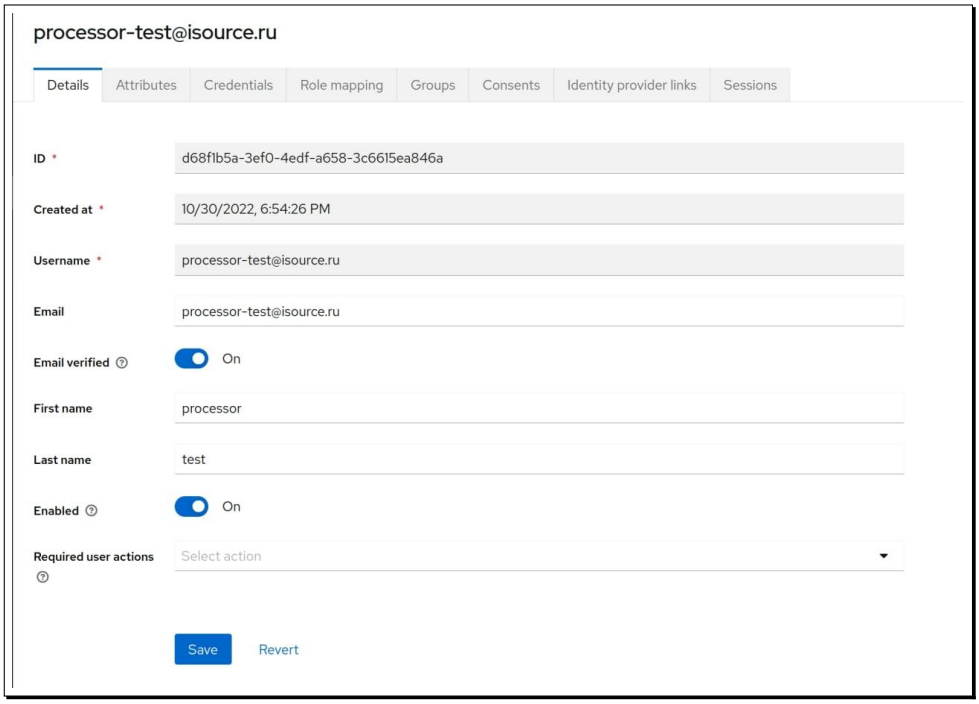


Рисунок 6: Пример добавления пользователя `processor-test@isource.ru`

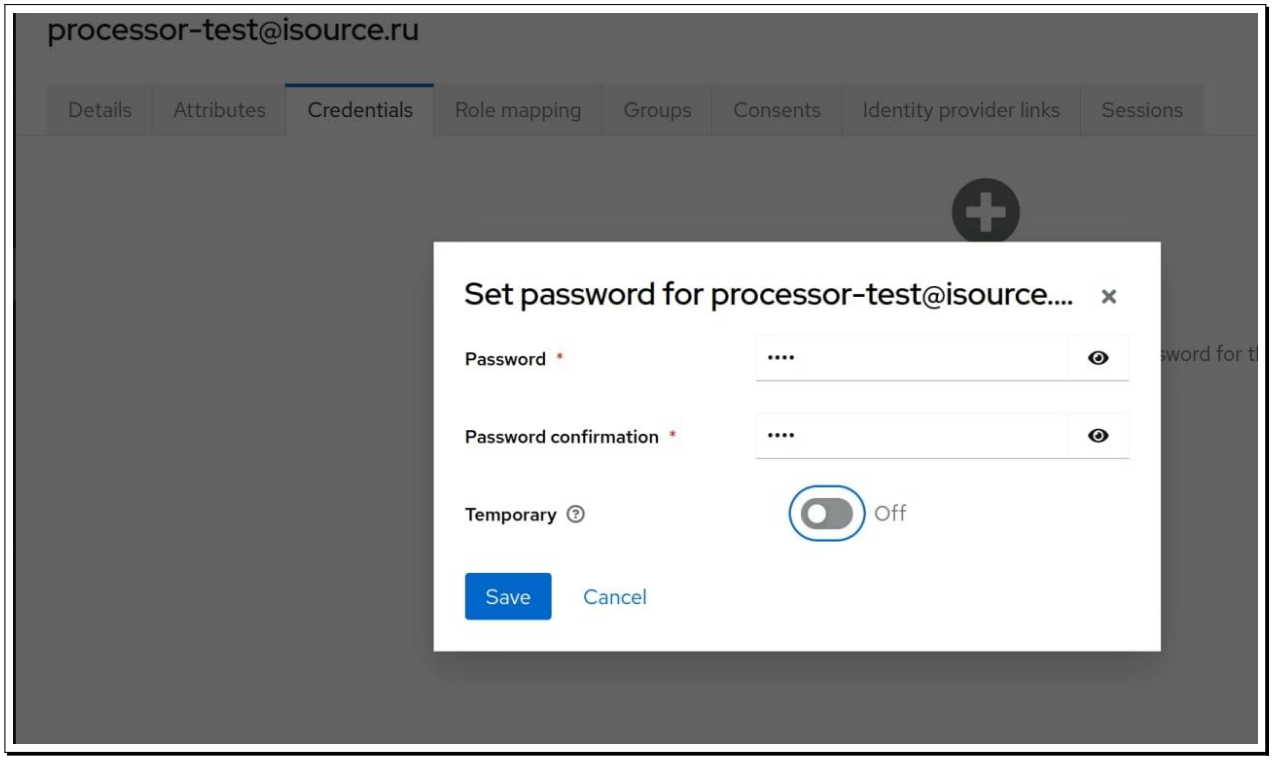


Рисунок 7: Пример назначения пароля пользователю `processor-test@isource.ru`

- после указания всех данных пользователя, необходимо нажать кнопку «`Create`», для его создания. Пароль пользователю назначается во вкладке «`Credentials`».

Необходимые для заполнения поля отмечены знаком (`*`). Пароль должен быть постоянным, то есть переключатель в поле «`Temporary`» должен быть в положении «`Off`», см. рисунок Рисунок 7, а

переключатель проверки адреса «Email verified» должен быть в положении «On», см. рисунок Рисунок 7.

- затем аналогичным образом требуется создать пользователя user-backoffice@main.ruz. Для этого необходимо создать его тем же самым способом, что и предыдущего пользователя. Отличия состоят в том, что используя вкладку «Attributes» ему необходимо указать мобильный телефон (значение (число номера телефона) может быть произвольным, а код страны должен начинаться с «+7»). Значение ключевого атрибута (Key) – «mobile». Кроме того, ключевое значение «Key» «generalAgreement» должно быть установлено в единицу. Пример совершения указанных операций изображен на рисунках Рисунок 8 и Рисунок 9:

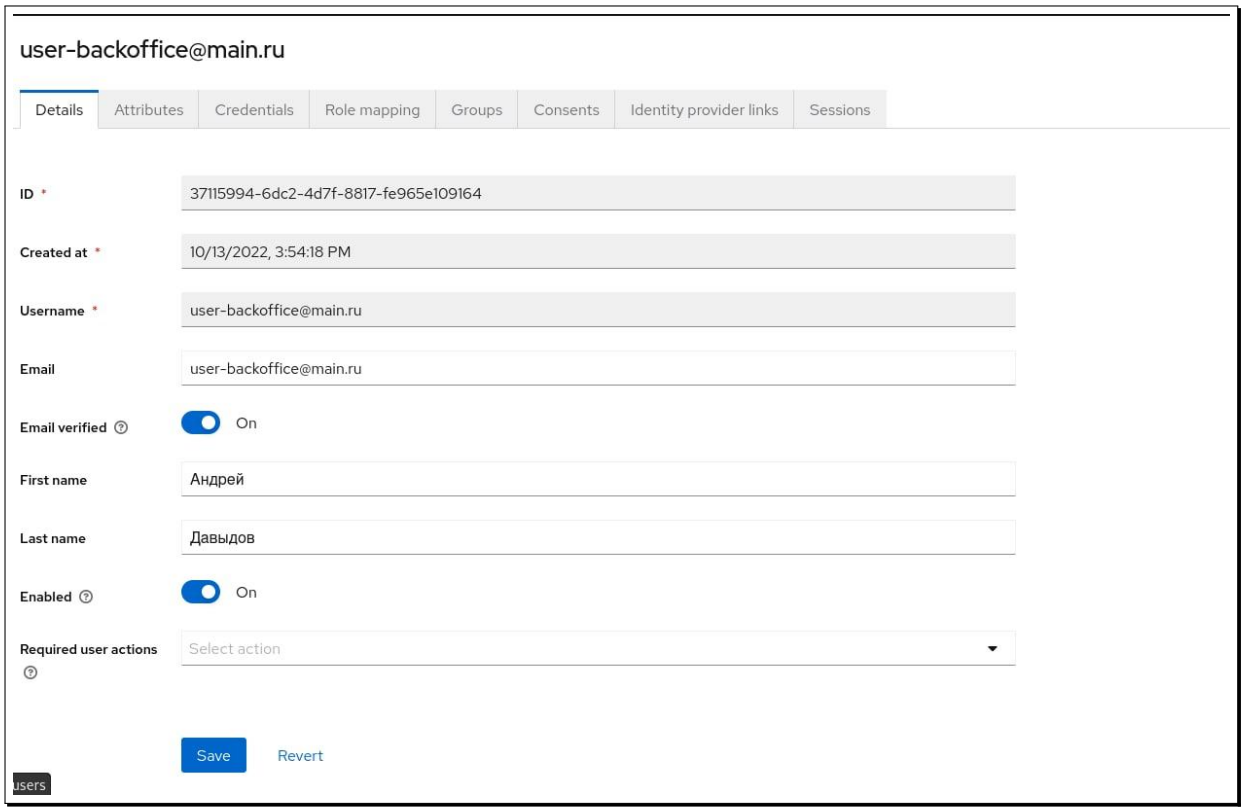


Рисунок 8: Пример добавления пользователя user-backoffice@main.ruz

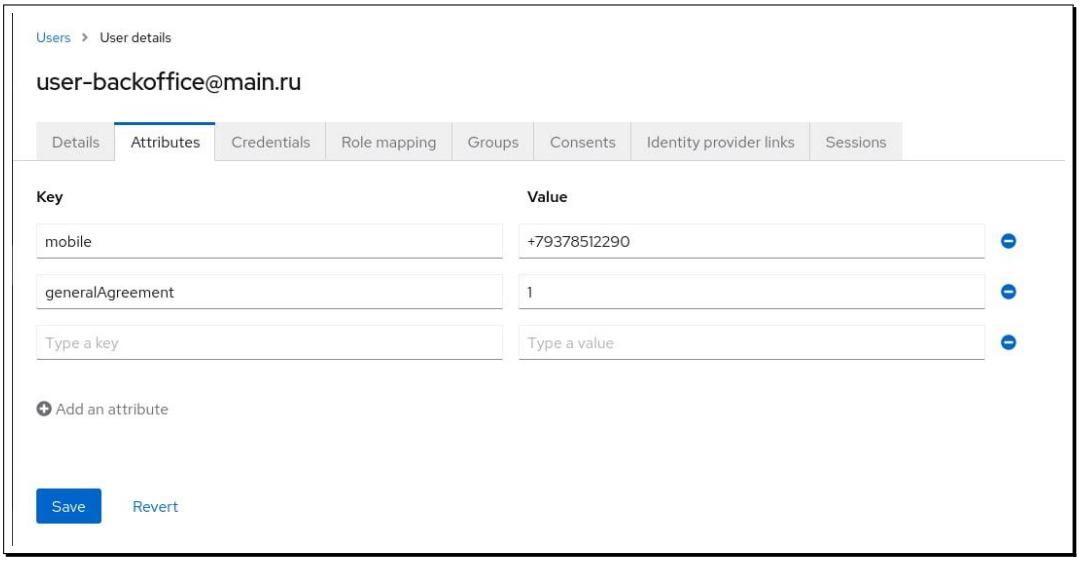


Рисунок 9: Атрибуты пользователя `user-backoffice@main.ruz`

- после этого пользователя `user-backoffice@main.ruz` необходимо ассоциировать с ролью «`gpnmarket_senior_backoffice`». Для этого используется вкладка «`Role mapping`». Пример совершения операции изображен на рисунке Рисунок 10:

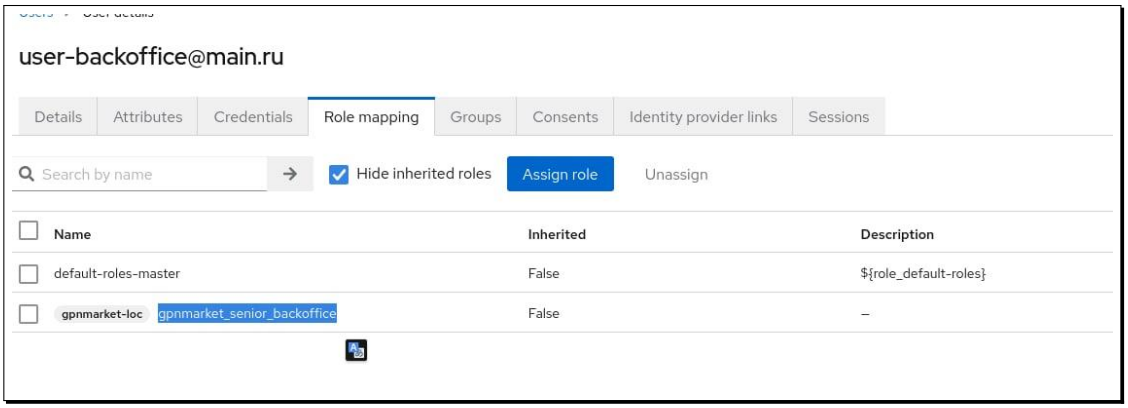


Рисунок 10: Ассоциация с ролью пользователя `user-backoffice@main.ruz`

- далее, используя вкладку «`Groups`» создать требуемую группу, например «`Группа БО "Меркурий"`», и после ее создания, переключившись в настройках группы во вкладку «`Attributes`» задать следующие ключевые значения:

  ⋆ «`Key`»: «`organizer_inn`»;

  ⋆ «`Value`»: «`6460053186`».

Пример совершения операции приведен на рисунке Рисунок 11:

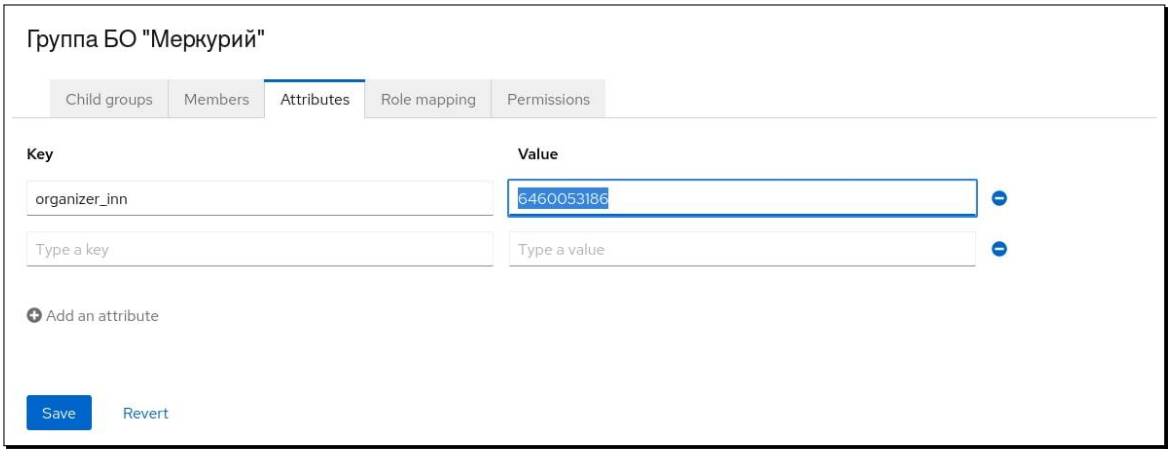


Рисунок 11: Значения атрибутов для группы «Группа БО "Меркурий"»

- затем необходимо добавить пользователя user-backoffice@main.ruz в созданную на предыдущем этапе группу «Группа БО "Меркурий"». Для этого используя вкладку «Groups» произвести назначение. Пример совершения данной операции изображен на рисунке Рисунок 12:

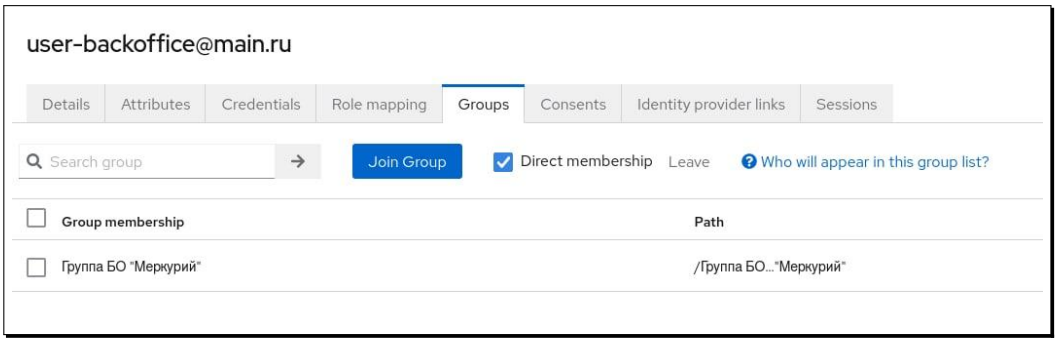


Рисунок 12: Пример добавления пользователя user-backoffice@main.ruz в группу

- на данном этапе необходимо ассоциировать для пользователя processor-test@isource.ru перечисленные ниже роли:

  - «manage-clients»;
  - «view-users»;
  - «query-groups»;
  - «query-clients»;
  - «view-events»;
  - «view-clients»;
  - «manage-users»;
  - «query-users»;
  - «view-realm».

Для этого следует в контексте пользователя `processor-test@isource.ru` и используя вкладку «`Role mapping`» назначить роли с помощью кнопки «`Assign role`». Пример совершения операции изображен на рисунке Рисунок 13:

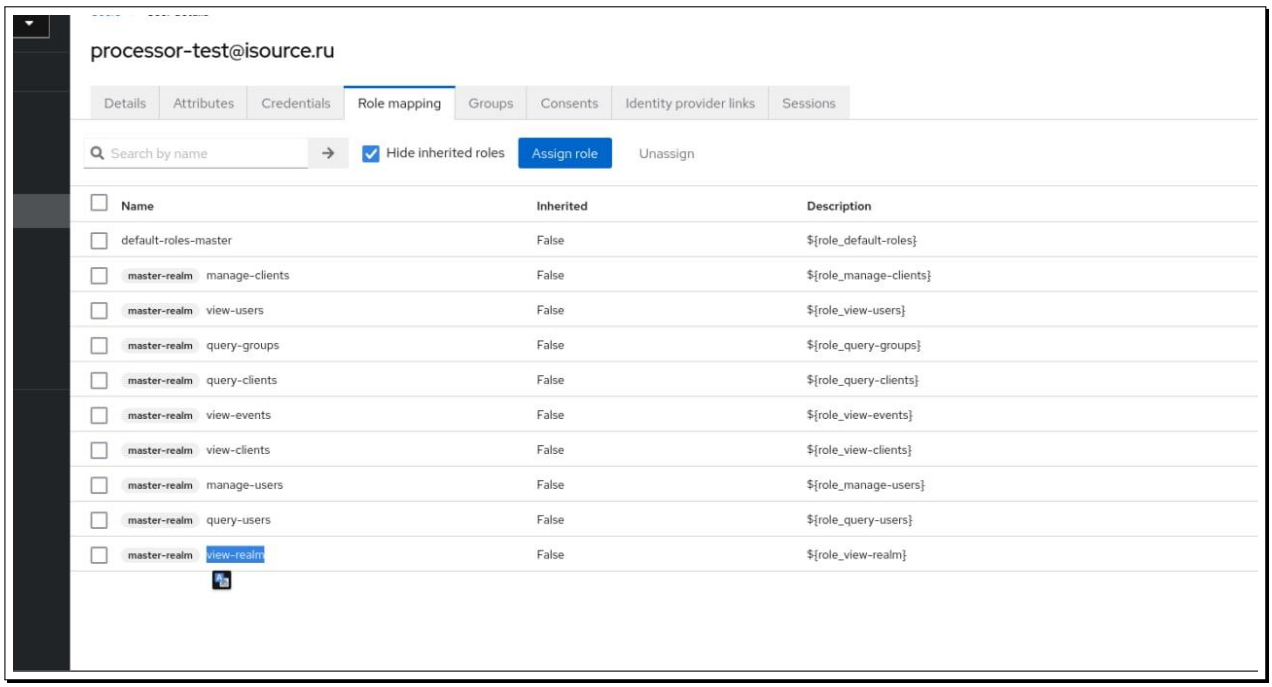


Рисунок 13: Назначение ролей для пользователя `processor-test@isource.ru`

5.1.6.2 Автоматизация импорта пользователей из файла `.json`

Для автоматизации создания пользователей, в составе дистрибутива по пути

`/data/artifacts/keycloak/*.json`

находятся файлы, с описанием пользовательских атрибутов. Например, для закупочного моделя, имя файла будет `processor_realm.json`, для модуля инспектора – `inspector_realm.json` и т.п.

Пример импорта изображен на рисунках Рисунок 14 – Рисунок 16.

Осуществить вход в интерфейс управления KeyCloak от имени администратора. Затем выбрать меню «`Administration Console`». Пример операции изображен на рисунке Рисунок 14:

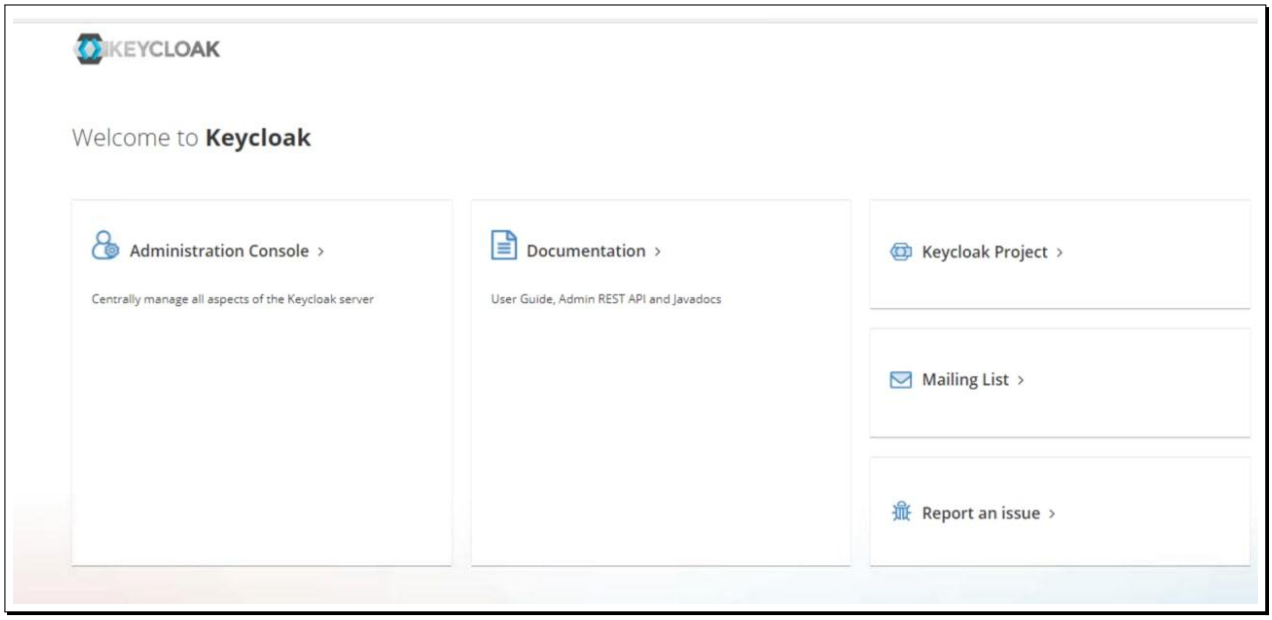


Рисунок 14: Образец входа в панель управления KeyCloak и начало операции импорта

Затем в отобразившейся странице осуществить выбор реалма (слева) и далее нажать «`Create Realm`». Пример операции изображен на рисунке Рисунок 15:

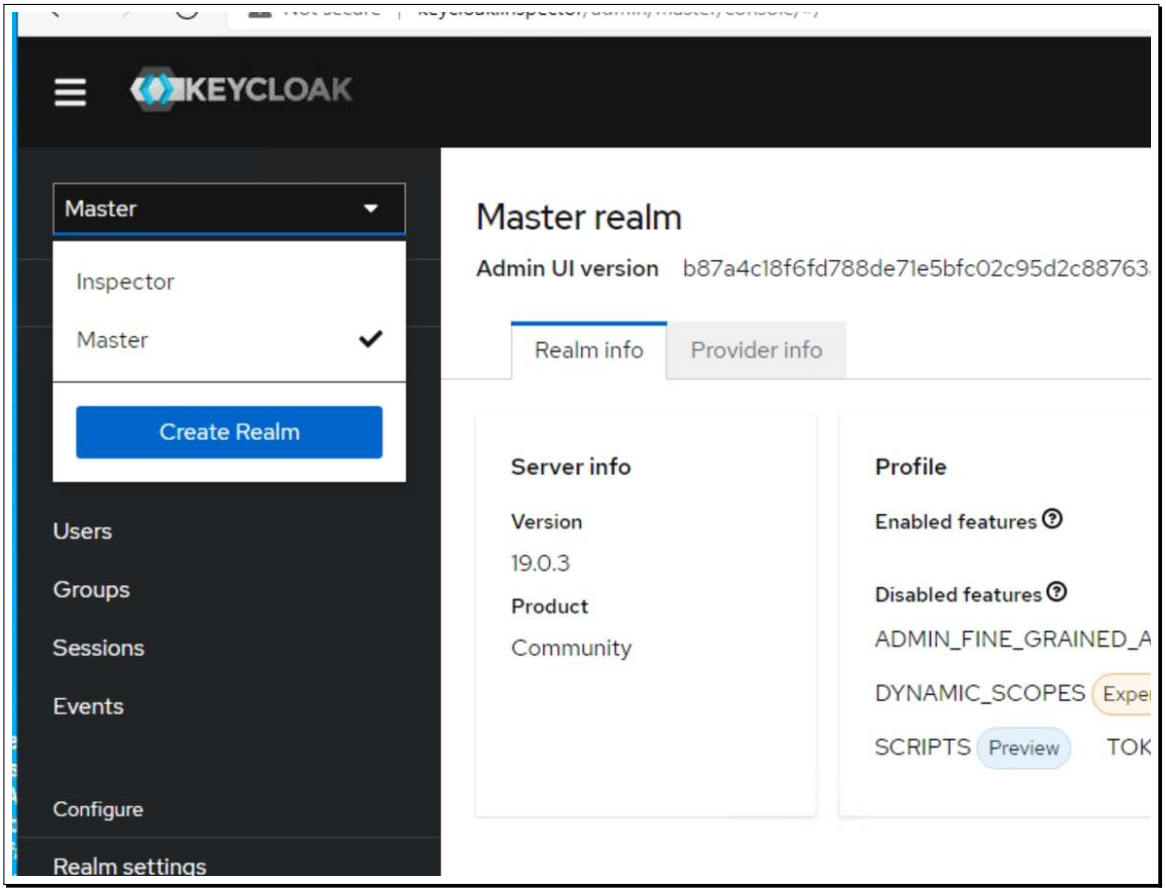


Рисунок 15: Пример выбора реалма для импорта пользователей

В появившемся окне, нажать «`Browse`» и выбрать для загрузки `.json` файл с реалмом программного компонента и нажать «`Create`». Пример операции изображен на рисунке Рисунок 16:

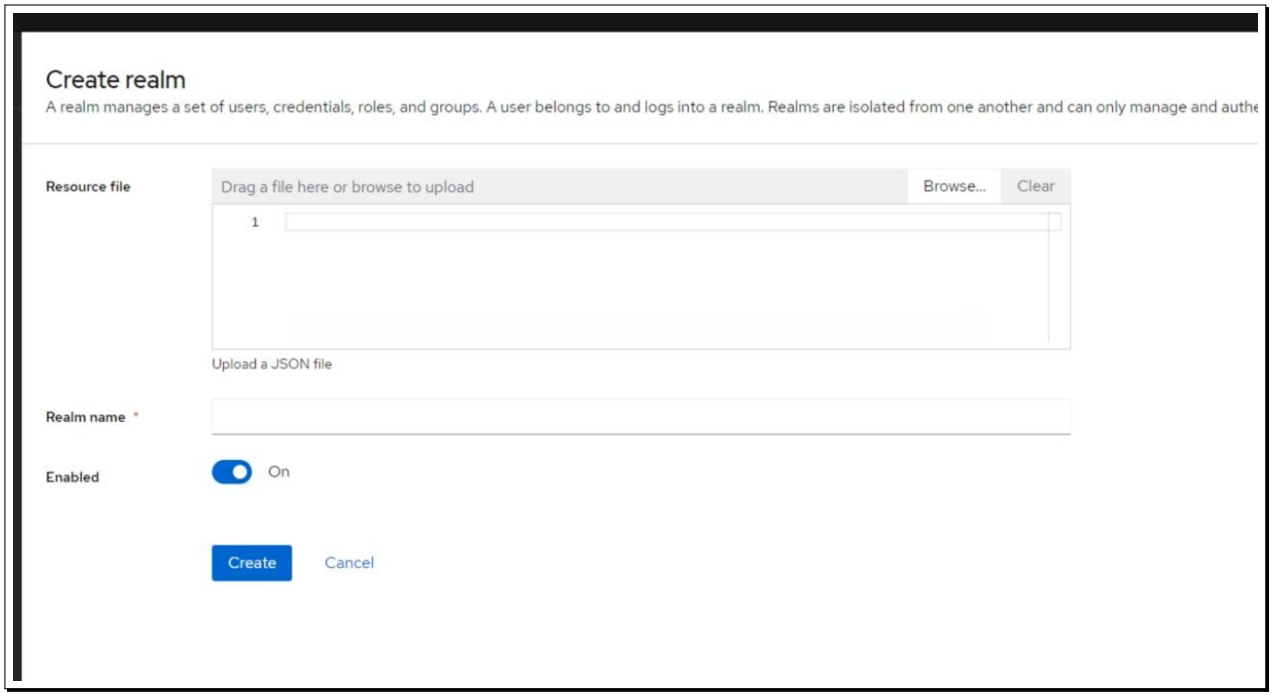


Рисунок 16: Пример выбора файла `.json` для импорта пользователей

5.1.6.3 Необходимые значения атрибутов для дальнейшего развертывания закупочного модуля

Для формирования сценария автоматизированного развертывания (плейбука) и дальнейшей установки закупочного модуля необходимы атрибуты, которые могут быть получены из провайдера KeyCloak.

Для получения токена (ключа) в формате RS256, который обеспечивает интеграцию запросов между прилодением и провайдером идентификации и аутентификации KeyCloak,требуется обратиться по адресу:

http://key.loc/auth/admin/master/console/#/master/realm-settings/keys

Где `key.loc` – пример адреса (URL) установленного провайдера KeyCloak, но данный адрес приведен только для примера, так как фактически для этого необходимо использовать FQDN.

Для выбора ключа необходимо переключиться на вкладку «`Keys`», затем выбрать ключ формата RS256 и нажать кнопку «`Public key`», после чего скопировать ключ и внести его в поле данных `app_processor_integrations_keycloak_key` структуры `app_processor_market.yml`. Данная структура данных описывается в параграфе 5.1.7.2.

Пример операции изображен на рисунках Рисунок 17, Рисунок 18 и Рисунок 19:

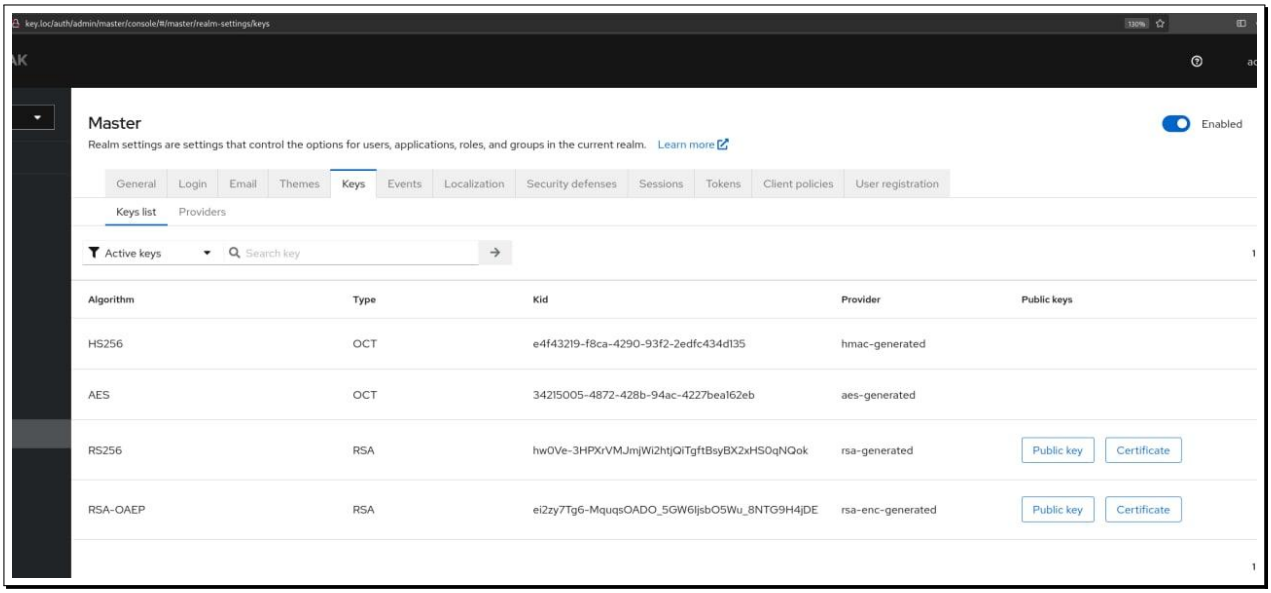


Рисунок 17: Пример содержимого вкладки Keys

Рисунок 18: Кнопка выбора публичного ключа

Рисунок 19: Пример значения публичного ключа RS256

Остальные значения переменных, необходимые для развертывания закупочного модуля, также выбираются из данных KeyCloak. Для этого необходимо выполнить запрос по адресу: http://key.loc/auth/admin/master/console/#/master/clients/f7a41e65-b219-45de-bc0a-14d67cbe18d9/settings

Где:

- `key.loc` – пример адреса (URL) установленного провайдера KeyCloak
- `f7a41e65-b219-45de-bc0a-14d67cbe18d9` – идентификатор (UUID) клиента (с именем в примере `processor-test`).

Для выполнения запроса нужно в провайдере KeyCloak выбрать «`Clients`», а затем «`Client details`». Пример окна с запросом изображен на рисунке Рисунок 20

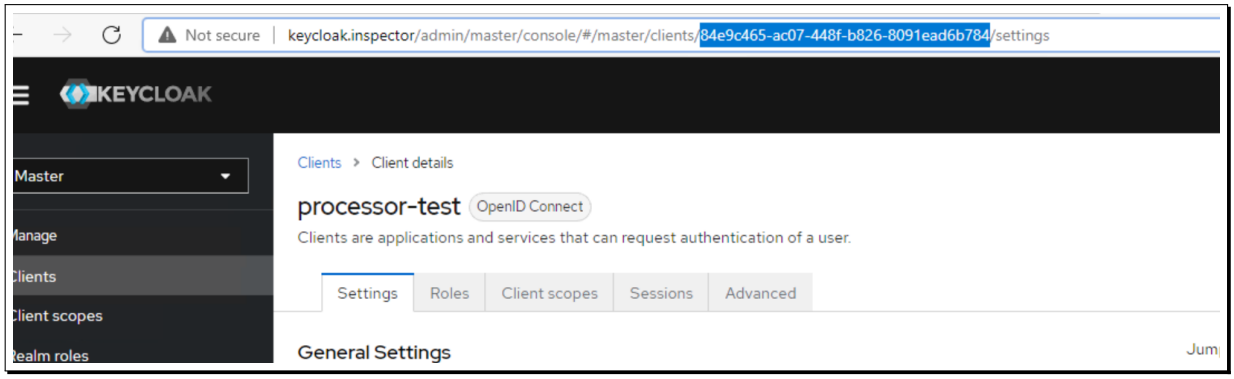


Рисунок 20: Пример окна со свойствами клиента

5.1.6.4 Необходимые значения параметров в структуру данных `app_processor_market.yml`

Значения в структуру данных `app_processor_market.yml` приведены в листинге ниже:

```
---
hosts_p_apt_packages:
  - unzip

npm_install: true
app_processor_market_directory: /data/gpnmarket_backend/releases/deploy
app_processor_market_shared_directory: /data/gpnmarket_backend/shared

app_processor_src: "{{ inventory_dir }}/../../data/artifacts/app_sources/
    SPO_Processor_back_07_12_2022.zip"
```

Листинг 33: Пример значений для структуры данных для `app_processor_market.yml`

5.1.7 Установка и настройка закупочного модуля

Программный компонент (модуль) «Закупочный модуль» предназначен для сопровождения процессов корпоративных закупок, организации и проведении процедур коммерческих закупок полного цикла на условиях инсорсинга и аутсорсинга.

5.1.7.1 Необходимые значения конфигурационных файлов СУБД PostgreSQL для закупочного модуля

В общем случае выполнение данного пункта необязательно, поскольку создание и настройка схемы PostgreSQL производится с помощью Ansible. Однако,

```
listen_addresses = '*'
```

Листинг 34: Пример настроек в `/etc/postgresql/12/main/postgresql.conf`

```
host    all             all             0.0.0.0/0               md5
```

Листинг 35: Пример настроек в `/etc/postgresql/12/main/pg_hba.conf`

Для создания необходимых пользователей в СУБД и структуры для хранения данных использовались следующие команды:

```
# sudo -u postgres psql
postgres=# create role gpnmarket;
postgres=# ALTER ROLE gpnmarket WITH PASSWORD 'gpnmarket';
postgres=# create database gpnmarket;
postgres=# grant all privileges on database gpnmarket to gpnmarket;
postgres=# alter role gpnmarket  superuser;
postgres=# \q
# systemctl restart postgresql-12.service
```

Листинг 36: Пример настройки СУБД `postgresql` для закупочного модуля

5.1.7.2 Данные структуры `app_processor_market.yml`

Структура данных `inventories/koa/group_vars/app_processor_market.yml` используется в интересах приложения (закупочного модуля). Пример содержимого для структуры данных
`inventories/koa/group_vars/app_processor_market.yml`
приведен на листинге ниже:

```
---
npm_install: true
app_processor_market_directory: /data/gpnmarket_backend/releases/deploy
app_processor_market_shared_directory: /data/gpnmarket_backend/shared

app_processor_src: "{{ inventory_dir }}/../../data/artifacts/app_sources/gpnmarket.zip"

app_processor_integrations_database:
    user: gpnmarket
    password: gpnmarket
    port: "5432"
    host: "{{ all_host_postgres }}"
    dbname: gpnmarket

app_processor_integrations_hub:
    baseurl: "https://hub-test.etpgpb.ru/api/v2"

app_processor_integrations_keycloak:
    baseurl: "http://{{ all_host_keycloak }}/"
```

```
    realm: master
    clientid: gpnmarket-loc-service
    clientsecret: d65f7284-db4c-4d77-8f8f-e1e830baac24

app_processor_integrations_keycloak_integration:
    clientid: gpnmarket-loc
    clientuuid: f7a41e65-b219-45de-bc0a-14d67cbe18d9
    auth:
      user: processor-test@isource.ru
      pass: Processor-test2
    service_account:
      clientid: gpnmarket-test-service
      clientUuid: ce5598e3-10ce-41d0-972b-09cc42445595
      clientSecret: b749af0e-1dc0-41ef-97ee-4a7507cfdf94

app_processor_integrations_keycloak_key: <значение ключа RS256>
app_processor_integrations_rabbitmq:
    user: rabbitmq
    pass: rabbitmq
    host: "{{ all_host_rabbitmq }}"
    port: "5672"
    url: "processor/messages"
app_processor_integrations_rabbitmq_connection_string: "amqp://{{ app_processor_integrations_rabbitmq
    .user }}:{{ app_processor_integrations_rabbitmq.pass }}@{{ app_processor_integrations_rabbitmq.
    host }}:{{ app_processor_integrations_rabbitmq.port }}/{{ app_processor_integrations_rabbitmq.url
     }}"

app_processor_integrations_radar:
    url: "https://radar-api-test.isource.ru"
    login: testUserAdmin
    pass: testUserAdmin2

app_processor_integrations_redis:
    host: "{{ all_host_redis }}"
    port: "6379"

app_processor_front_src: "{{ inventory_dir }}/../../data/artifacts/app_sources/gpnmarket-front.zip"

app_processor_user: worker
app_processor_group: worker
app_processor_front_project_directory: "/data/processor-front"

app_processor_front_npm_registry: "repos.techpark.local/repository/npm-proxy"

app_processor_front_integrations:
  keycloak:
    url: http://{{ all_host_keycloak }}/auth
    realm: master
    clientid: gpnmarket-loc
  chattest: https://chat-test.isource.ru/
  wsChat: wss://notifications-test.isource.ru/ws
  wsnotifi: wss://notifications-test.isource.ru/ws
  notification: notifications-test.isource.ru
```

Листинг 37: Пример заполения структуры данных для `app_processor_market.yml`

5.1.7.3 Запуск закупочного модуля

После подготовки и запуска всего окружения (описано в разделах 5.1.3, 5.1.4, 5.1.5 и 5.1.6), можно инициализировать запуск закупочного модуля (приложения). Запуск осуществляется командой:

```
ansible-playbook main.yml -i inventories/koa/inventory.yml -t app-processor-market
```

Листинг 38: Пример запуска закупочного модуля

После запуска рекомендуется произвести проверку корректности развертывания программного комплекса. Для этого нужно выполнить действия, описанные в разделе 6.

На этом установка закупочного модуля завершена.

Процедуры, необходимые для установки модулей «Модуль планирования» и «Договорной модуль», приведены в разделах 5.1.11 и 5.1.10 соответственно, и на данном этапе не осуществляются.

5.1.8 Установка модуля цифрового инспектора

Программный компонент (модуль) «Цифровой инспектор» предназначен для сопровождения процессов управления качеством МТР, организации и проведения Технического аудита, Инспекционного контроля на всех этапах производства сложного технологического оборудования и Входного контроля поставленных МТР.

В данном разделе приведено описание процедур установки модуля цифрового инспектора. Данное описание приведено с учетом того, что закупочный модуль уже установлен и в разделе описаны процедуры установки, которые отличаются (или выполняются дополнительно) от тех, которые приведены в разделе 5.1.7, описывающем установку и настройку закупочного модуля.

5.1.8.1 Отличия в структурах данных для модуля цифрового инспектора

Для модуля цифрового инспектора отличия от модуля закупок в структурах данных приведены ниже:

```
inventories/koa/group_vars
├── all
├── k8s_cluster
│   ├── addons.yml
│   ├── k8s-cluster.yml
│   ├── k8s-net-calico.yml
│   └── preprovision.yml
├── kube_control_plane.yml
├── kubernetes_apps
│   └── repos.yaml
├── kubespray
│   ├── all.yml
│   ├── cri-o.yml
│   ├── etcd.yml
│   ├
```

```
| ├── oci.yml
| └── offline.yml
```

Листинг 39: Структура данных для автоматизированного развертывания модуля цифрового инспектора с помощью `ansible`

Указание версии k8s осуществляется в файле

`inventories/koa/group_vars/k8s_cluster/k8s-cluster.yml`

Списки зависимостей и формирование путей для них при оффлайн установке кластера k8s описываются в файле:

`inventories/koa/group_vars/kubespray/offline.yml`

Состав контейнеров (с указанием версий) для кластера k8s указан в таблице Таблица 24.

5.1.8.2 Развертывание модуля цифрового инспектора

5.1.8.2.1 Общие переменные и структуры данных

В файле `inventories/koa/inventory.yml` заполнить данные для кластера k8s и сервера NFS, согласно приведенному примеру:

```
# 20 строчка@@
    kub-1:
      ansible_host: spb99tp8394-07
    kub-2:
      ansible_host: spb99tp8394-08
    kub-3:
      ansible_host: spb99tp8394-11
    kub-4:
      ansible_host: spb99tp8394-10
    kub-5:
      ansible_host: spb99tp8394-20
    kub-6:
      ansible_host: spb99tp8394-21
    nfs:
      ansible_host: spb99tp8394-12

#75 строчка
    kubespray:
      children:
        k8s_cluster:
        etcd:

    calico_rr:

    kube_control_plane:
      hosts:
```

```
        kub-1:
        kub-2:
        kub-3:
    kube_node:
      hosts:
        kub-4:
        kub-5:
        kub-6:

    k8s_cluster:
      children:
        kube_control_plane:
        kube_node:
        calico_rr:

    etcd:
      children:
        kube_control_plane:

    kubernetes_apps:
      hosts:
        kub-1:
```

Листинг 40: Пример заполения структуры данных для модуля цифрового инспектора в `inventory.yml`

Затем необходимо с АРМ, откуда осуществляется автоматизированное развертывание, выполнить предварительную настройку целевых ВМ:

```
ansible-playbook main.yml -i inventories/koa/inventory.yml  -t preprovision
```

Листинг 41: Пример запуска целевой настройки для узлов, обслуживающих модуль цифрового инспектора

Для хранилища данных NFS используется файл:

`inventories/koa/group_vars/nfs.yml`

В этом файле указать желаемые настройки NFS, например:

```
---
nfs_s_root_paths:
  - name: "{{ all_nfs_server_path }}"
    owner: nobody
    group: nogroup
    inputs: '*'
    options:
      - rw
      - sync
      - no_subtree_check
```

Листинг 42: Пример заполения структуры данных для модуля цифрового инспектора в `nfs.yml`

Затем осуществить непосредственное развертывание[9] кластера k8s средствами kubespray:

[9] В зависимости от аппаратных характеристик кластера, выделенного под обслуживание k8s, выполнение команды может занимать длительное время (30 минут, и более).

```
ansible-playbook playbooks/kubernetes/kubespray.yml -i inventories/koa/inventory.yml --become
```

Листинг 43: Пример инициализации развертывания кластера k8s

Затем развернуть хранилище данных NFS:

```
ansible-playbook main.yml -i inventories/koa/inventory.yml -t nfs
```

Листинг 44: Пример инициализации развертывания NFS

Общие переменные в структуре данных для infra-kubernetes в файле

`inventories/koa/group_vars/kubernetes_apps/repos.yml`:

```
---

helm_pi_kubeconfig_path: "/etc/kubernetes/admin.conf"
helm_pi_repositories:
  - name: helm-onpremise
    repo_url: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise

# helm releases for upgrade
helm_pi_releases:
  - name: csi-driver-nfs
    chart_name: csi-driver-nfs
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: kube-system # omit
    create_namespace: false # omit
    version: "v4.1.0"
    value:
      image:
        nfs:
          repository: "{{ artifacts_tl_docker_registry }}/sig-storage/nfsplugin"
          tag: v4.1.0
          pullPolicy: IfNotPresent
        csiProvisioner:
          repository: "{{ artifacts_tl_docker_registry }}/sig-storage/csi-provisioner"
          tag: v3.2.0
          pullPolicy: IfNotPresent
        livenessProbe:
          repository: "{{ artifacts_tl_docker_registry }}/sig-storage/livenessprobe"
          tag: v2.7.0
          pullPolicy: IfNotPresent
        nodeDriverRegistrar:
          repository: "{{ artifacts_tl_docker_registry }}/sig-storage/csi-node-driver-registrar"
          tag: v2.5.1
          pullPolicy: IfNotPresent

  - name: ingress-nginx
    chart_name: ingress-nginx
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: ingress-nginx # omit
    create_namespace: true # omit
    version: "4.2.5"
    value:
      controller:
        name: controller
        image:
```

```
          chroot: false
          registry: "{{ artifacts_tl_docker_registry }}"
          image: ingress-nginx/controller
          tag: "v1.3.1"
          digest: ""
          digestChroot: ""
          pullPolicy: IfNotPresent
          runAsUser: 101
          allowPrivilegeEscalation: true
        containerPort:
          http: 80
          https: 443
        service:
          type: ClusterIP
        addHeaders: {}
        dnsConfig: {}
        dnsPolicy: ClusterFirst
        hostNetwork: true
        ingressClassResource:
          name: nginx
          enabled: true
          default: false
          controllerValue: "k8s.io/ingress-nginx"
        ingressClass: nginx
        kind: DaemonSet
        admissionWebhooks:
          enabled: true
          patch:
            enabled: true
            image:
              registry: "{{ artifacts_tl_docker_registry }}"
              image: ingress-nginx/kube-webhook-certgen
              tag: v1.3.0
              digest: ""
              pullPolicy: IfNotPresent
      imagePullSecrets: []

helm_pi_manifests:
  - namespace: kube-system
    definition: |
      apiVersion: storage.k8s.io/v1
      kind: StorageClass
      metadata:
        name: nfs-csi
      provisioner: nfs.csi.k8s.io
      parameters:
        server: "{{ all_host_nfs }}"
        share: "{{ all_nfs_server_path }}"
      reclaimPolicy: Delete
      volumeBindingMode: Immediate
      mountOptions:
        - nfsvers=4.1
```

Листинг 45: Пример содержимого структуры данных в файле `repos.yml`

Далее выполнить развертывание системных компонентов k8s (таких как ingress-nginx - вывод трафика и единой точки подключения к кластеру и nfs csi provisioner - для автоматической подготовки томов данных приложений, их монтирования и поддержки):

```
ansible-playbook main.yml -i inventories/koa/inventory.yml -t infra-kubernetes
```

Листинг 46: Пример инициализации развертывания системных компонентов кластера k8s

Пример содержимого структуры данных в файле `k8s-cluster.yml`:

```
---
helm_enabled: true
# Kubernetes configuration dirs and system namespace.
# Those are where all the additional config stuff goes
# the kubernetes normally puts in /srv/kubernetes.
# This puts them in a sane location and namespace.
# Editing those values will almost surely break something.
kube_config_dir: /etc/kubernetes
kube_script_dir: "{{ bin_dir }}/kubernetes-scripts"
kube_manifest_dir: "{{ kube_config_dir }}/manifests"

# This is where all the cert scripts and certs will be located
kube_cert_dir: "{{ kube_config_dir }}/ssl"

# This is where all of the bearer tokens will be stored
kube_token_dir: "{{ kube_config_dir }}/tokens"

kube_api_anonymous_auth: true

## Change this to use another Kubernetes version, e.g. a current beta release
kube_version: v1.23.7
```

Листинг 47: Пример содержимого структуры данных в файле `k8s-cluster.yml`

```
---

hosts_p_docker_provision: true
hosts_p_apt_packages:
  - docker.io

kafka_zd_compose_path: /var/kafka-zookeper-docker

kafka_zd_data_paths:
  - "{{ kafka_zd_compose_path }}/zoo/data"
  - "{{ kafka_zd_compose_path }}/zoo/datalog"
  - "{{ kafka_zd_compose_path }}/kafka/data"
  - "{{ kafka_zd_compose_path }}/kowl"

kafka_zd_requirements:
  - docker.io
  - docker-compose

kafka_zd_kafka_image: "{{ hosts_p_docker_registry }}/confluentinc/cp-kafka:6.2.0"
kafka_zd_zookeeper_image: "{{ hosts_p_docker_registry }}/zookeeper:3.4.14"
kafka_zd_kowl_image: "{{ hosts_p_docker_registry }}/quay.io/cloudhut/kowl:v1.2.1"

kafka_zd_kafka_topics:
  - inspector

kafka_za_kowl: true
kafka_zd_kowl_port: 8080
```

Листинг 48: Пример содержимого структуры данных в файле `kafka.yml`

Где `kowl` - адрес веб-интерфейса для подключениям (обращениям) к kafka.

Выполнить установку и запуск kafka:

```
ansible-playbook main.yml -i inventories/koa/inventory.yml -t kafka
```

Листинг 49: Пример инициализации развертывания kafka

Переменные в структуре данных `inventories/koa/group_vars/mysql.yml`:

```
---
hosts_p_docker_provision: true
hosts_p_apt_packages:
  - docker.io
  - python3-pip

mysql_d_compose_path: /var/mysql-docker
mysql_d_data_path:
  - "{{ mysql_d_compose_path }}"

mysql_d_image: "{{ hosts_p_docker_registry }}/mysql:8.0.31"
mysql_d_requirements:
  - docker.io
  - docker-compose
  - mysql-client-8.0
  - python3-pymysql

mysql_d_services:
  - name: mysql_inspector
    port: 3306
    environments:
    - name: MYSQL_ROOT_PASSWORD
      value: "inspector-local-db-pass"
    - name: MYSQL_DATABASE
      value: "inspector"
    - name: MYSQL_TCP_PORT
      value: 3306
    dump: "{{ inventory_dir }}/../../data/artifacts/dumps/inspector.sql"
```

Листинг 50: Пример содержимого структуры данных в файле `mysql.yml`

Выполнить установку и запуск СУБД mysql в кластере k8s:

```
  ansible-playbook main.yml -i inventories/koa/inventory.yml -t mysql
```

Листинг 51: Пример инициализации развертывания kafka

5.1.8.2.2 Переменные и структуры данных для модуля цифрового инспектора

Переменные в структуре данных содержатся в файле

`inventories/koa/group_vars/app_inspector.yml`

```
---

helm_pi_kubeconfig_path: "/etc/kubernetes/admin.conf"
```

```
helm_pi_repositories:
  - name: helm-onpremise
    repo_url: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise

helm_pi_manifests:
  - definition: |
      apiVersion: v1
      kind: Namespace
      metadata:
        name: inspector
  - namespace: inspector
    definition: |
      apiVersion: v1
      kind: PersistentVolumeClaim
      metadata:
        name: inspector-upload-providers
      spec:
        storageClassName: nfs-csi
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 4Gi

  - namespace: inspector
    definition: |
      apiVersion: v1
      kind: PersistentVolumeClaim
      metadata:
        name: inspector-import-files
      spec:
        storageClassName: nfs-csi
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 4Gi
  - namespace: inspector
    definition: |
      apiVersion: v1
      kind: PersistentVolumeClaim
      metadata:
        name: inspector-public
      spec:
        storageClassName: nfs-csi
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 4Gi

# helm releases for upgrade
helm_pi_releases:
  - name: inspector-redis
    chart_name: planning
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: inspector # omit
    create_namespace: false # omit
    version: "0.0.3"
```

```
    value:
      namespace:
        name: inspector
      statefulsets:
        redis:
          replicas: 1
          volumes:
          - name: data
            emptyDir:
              medium: ""
          containers:
            redis:
              image: "{{ hosts_p_docker_registry }}/redis"
              imageTag: '7.0.5'
              command:
                - "/bin/sh"
              args:
                - "-c"
                - "redis-server --requirepass RediS123pass --appendonly yes"
              containerPorts:
              - 6379
              memory: "256Mi"
              securityContext:
                readOnlyRootFilesystem: true
              volumeMounts:
                - name: data
                  mountPath: /data
          servicePorts:
          - name: redis
            port: 6379
            targetPort: 6379
- name: inspector-backend
    chart_name: planning
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: inspector # omit
    create_namespace: false # omit
    version: "0.0.3"
    value:
      namespace:
        name: inspector
      deployments:
        back:
          replicas: 1
          volumes:
          - name: upload-providers-xlsx
            persistentVolumeClaim:
              claimName: "inspector-upload-providers"
          - name: import-files
            persistentVolumeClaim:
              claimName: "inspector-import-files"
          - name: public
            persistentVolumeClaim:
              claimName: "inspector-public"
          containers:
            inspector-backend:
              image: "{{ hosts_p_docker_registry }}/koa/inspector/inspector-backend"
              imageTag: '131122'
              containerPorts:
              - 9100
```

```
            memory: "4Gi"
            cpu: "4"
            securityContext:
              readOnlyRootFilesystem: false
            volumeMounts:
            - name: upload-providers-xlsx
              mountPath: /inspector-backend/upload-providers-xlsx
            - name: import-files
              mountPath: /inspector-backend/import-files
            - name: public
              mountPath: /inspector-backend/public
            env:
              CFG_LOGGER_ENABLED: 1
              CFG_API_PORT: 9100
              CFG_UPLOAD_PROVIDERS_XLSX: upload-providers-xlsx
              CFG_DB_TYPE: mysql
              CFG_DB_HOST: "{{ all_host_mysql }}"
              CFG_DB_USERNAME: root
              CFG_DB_USERPASS: "inspector-local-db-pass"
              CFG_DB_DATABASE: inspector
              CFG_DB_PORT: 3306
              CFG_DB_LOGGER: 0
              CFG_SSO_REALM: inspector
              CFG_SSO_BEARER_ONLY: 1
              CFG_SSO_AUTH_SERVER: "http://keycloak.inspector"
              CFT_SSO_AUTH_DOCKER_PORT_OVERRIDE: ""
              CFG_SSO_CLIENT_ID: ci-test
              CFG_SOURCE_IMPORT_FETCH_INTERVAL_MIN: 180
              CFG_EMAIL_SENDER_HOST: smtp.yandex.ru
              CFG_EMAIL_SENDER_PORT: 465
              CFG_EMAIL_SENDER_PASSWORD: ""
              CFG_EMAIL_DEFAULT_SENDER: ""
              CFG_EMAIL_DEFAULT_FROM: ""
              CFG_EMAIL_FEEDBACK_RECEIVERS: ""
              CFG_EMAIL_MAX_ATTACHMENTS_SIZE_MB: 20
              CFG_TELERECEPTION_TOKEN: 5
  ec91f5f36f9549de28ef1756ecd53a0cd976a023ecce5a6fdd9c8c0a2117487
              CFG_REDIS_HOST: redis
              CFG_REDIS_PORT: 6379
              CFG_REDIS_PASS: RediS123pass
              CFG_REDIS_DB_ID: 0
              CFG_BROKER_SERVER: "{{ all_host_kafka }}:9092"
              CFG_BROKER_CLIENT_ID: backend
              CFG_BROKER_TOPIC_PREFIX: inspector
              CFG_DOCS_CONVERTER_HOST: http://libreoffice
              CFG_DOCS_CONVERTER_PORT: 6000
              CFG_FRONT_ADMIN: http://admin.inspector
              CFG_FRONT_CLIENT: http://client.inspector
              CFG_FRONT_INSPECTOR: http://cabinet.inspector
          servicePorts:
          - name: inspector-backend
            port: 9100
            targetPort: 9100
      hostAliases:
      - ip: "{{ all_host_keycloak }}"
        hostnames:
        - keycloak.inspector

  - name: inspector-frontend
```

```
    chart_name: planning
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: inspector # omit
    create_namespace: false # omit
    version: "0.0.3"
    value:
      namespace:
        name: inspector
      hostAliases:
      - ip: 127.0.0.1
        hostnames:
        - cabinet.inspector
        - api.inspector
        - admin.inspector
        - client.inspector
        - keycloak.inspector
      configmaps:
        nginx-config:
          data:
            nginx.conf: |
              server {
                listen 80;
                server_name api.inspector;
keepalive_timeout 70;
                add_header Strict-Transport-Security max-age=15768000;
                add_header X-XSS-Protection "1; mode=block";
                add_header X-Content-Type-Options nosniff;
                add_header X-Frame-Options DENY;

                location / {
                  proxy_read_timeout 120s;
                  proxy_pass http://inspector-backend:9100;
                  proxy_set_header Host $host;
                  proxy_http_version 1.1;
                  proxy_set_header Upgrade $http_upgrade;
                  proxy_set_header Connection "upgrade";
                }

                location ^~/public/ {
                  expires 10d;
                  add_header Cache-Control "Public";
                  root /sites/backend/radar-back/test/;
                }
              }

              # кабинет инспектора
              server {
                listen 80;
                server_name cabinet.inspector;

                keepalive_timeout 70;
                add_header Strict-Transport-Security max-age=15768000;
                add_header X-XSS-Protection "1; mode=block";
                add_header X-Content-Type-Options nosniff;
                add_header X-Frame-Options DENY;

                root /usr/share/nginx/html/cabinet-inspector;

                location / {
```

```
                try_files $uri $uri/ /index.html;
            }
        }

        # кабинет клиента
        server {
            listen 80;
            server_name client.inspector;

            keepalive_timeout 70;
            add_header Strict-Transport-Security max-age=15768000;
            add_header X-XSS-Protection "1; mode=block";
            add_header X-Content-Type-Options nosniff;
            add_header X-Frame-Options DENY;

            root /usr/share/nginx/html/cabinet-client;

            location / {
                try_files $uri $uri/ /index.html;
            }
        }

        # кабинет админа
        server {
            listen 80;
            server_name admin.inspector;

            keepalive_timeout 70;
            add_header Strict-Transport-Security max-age=15768000;
            add_header X-XSS-Protection "1; mode=block";
            add_header X-Content-Type-Options nosniff;
            add_header X-Frame-Options DENY;

            root /usr/share/nginx/html/cabinet-administrator;

            location / {
                try_files $uri $uri/ /index.html;
            }
        }
        server {
            listen 80;
            server_name keycloak.inspector;

            keepalive_timeout 70;

            location / {
                proxy_read_timeout 120s;
                proxy_pass http://{{ all_host_keycloak }}:80;
                proxy_set_header Upgrade           $http_upgrade;
                proxy_set_header Connection        "upgrade";
                proxy_set_header Host              $host;
                proxy_set_header X-Real-IP         $remote_addr;
                proxy_set_header X-Forwarded-For   $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header X-Forwarded-Host  $host;
                proxy_set_header X-Forwarded-Port  $server_port;
            }
        }
deployments:
```

```
    front:
      replicas: 1
      volumes:
      - name: nginx-config
        configMap:
          name: nginx-config
          items:
          - key: "nginx.conf"
            path: "nginx.conf"
      containers:
        nginx:
          image: "{{ hosts_p_docker_registry }}/koa/inspector/inspector-frontend"
          imageTag: '131122'
          containerPorts:
          - 80
          memory: "128Mi"
          securityContext:
            readOnlyRootFilesystem: false
          volumeMounts:
          - name: nginx-config
            mountPath: /etc/nginx/conf.d/
            readOnly: true
      servicePorts:
      - name: front
        port: 80
        targetPort: 80
  ingress:
    planning:
      rules:
        - host: "{{ all_target_inspector_cabinet }}"
          http:
            paths:
              - path: /
                pathType: Prefix
                backend:
                  service:
                    name: front
                    port:
                      number: 80
        - host: "{{ all_target_inspector_api }}"
          http:
            paths:
              - path: /
                pathType: Prefix
                backend:
                  service:
                    name: front
                    port:
                      number: 80
        - host: "{{ all_target_inspector_client }}"
          http:
            paths:
              - path: /
                pathType: Prefix
                backend:
                  service:
                    name: front
                    port:
                      number: 80
```

```
            - host: "admin.inspector"
              http:
                paths:
                  - path: /
                    pathType: Prefix
                    backend:
                      service:
                        name: front
                        port:
                          number: 80
```

Листинг 52: Пример содержимого структуры данных в файле `app_inspector.yml`

Запуск развертывания модуля цифрового инспектора осуществляется командой:

```
ansible-playbook main.yml -i inventories/koa/inventory.yml -t app-inspector
```

Листинг 53: Пример инициализации развертывания kafka

Добавление данных о пользователях инспектора в провайдер аутентификации KeyCloak осуществляется аналогичным способом, описанном в разделе 5.1.6.2. За тем исключением, что файл `.json`, содержащий мета-данные пользователей модуля цифрового инспектора находится по пути:

`./data/artifacts/dumpls/keycloak/inspector-realm.json`

Для примера[10] ниже приводится часть содержимого файла `inspector-realm.json`:

```
{
  "id" : "6a7c341a-9ab7-4a02-9f2d-8052699ad96a",
  "realm" : "inspector",
  "notBefore" : 0,
  "defaultSignatureAlgorithm" : "RS256",
  "revokeRefreshToken" : false,
  "refreshTokenMaxReuse" : 0,
  "accessTokenLifespan" : 300,
  "accessTokenLifespanForImplicitFlow" : 900,
  "ssoSessionIdleTimeout" : 1800,
  "ssoSessionMaxLifespan" : 36000,
  "ssoSessionIdleTimeoutRememberMe" : 0,
  "ssoSessionMaxLifespanRememberMe" : 0,
  "offlineSessionIdleTimeout" : 2592000,
  "offlineSessionMaxLifespanEnabled" : false,
  "offlineSessionMaxLifespan" : 5184000,
  "clientSessionIdleTimeout" : 0,
  "clientSessionMaxLifespan" : 0,
  "clientOfflineSessionIdleTimeout" : 0,
  "clientOfflineSessionMaxLifespan" : 0,
  "accessCodeLifespan" : 60,
  "accessCodeLifespanUserAction" : 300,
  "accessCodeLifespanLogin" : 1800,
  "actionTokenGeneratedByAdminLifespan" : 43200,
  "actionTokenGeneratedByUserLifespan" : 300,
  "oauth2DeviceCodeLifespan" : 600,
  "oauth2DevicePollingInterval" : 5,
```

[10] Содержимое файла приводится справочно, и не целиком (первые 50 строк). Цель примера – отразить концепцию применения методов добавления пользователей.

```
  "enabled" : true,
  "sslRequired" : "external",
  "registrationAllowed" : false,
  "registrationEmailAsUsername" : false,
  "rememberMe" : false,
  "verifyEmail" : false,
  "loginWithEmailAllowed" : true,
  "duplicateEmailsAllowed" : false,
  "resetPasswordAllowed" : false,
  "editUsernameAllowed" : false,
  "bruteForceProtected" : false,
  "permanentLockout" : false,
  "maxFailureWaitSeconds" : 900,
  "minimumQuickLoginWaitSeconds" : 60,
  "waitIncrementSeconds" : 60,
  "quickLoginCheckMilliSeconds" : 1000,
  "maxDeltaTimeSeconds" : 43200,
  "failureFactor" : 30,
  "roles" : {
    "realm" : [ {
      "id" : "6754d73b-b414-4253-a02a-0576cc0f42f7",
      "name" : "default-roles-inspector",
      "description" : "${role_default-roles}"
```

Листинг 54: Пример содержимого структуры данных в файле `inspector-realm.json`

На этом установка и конфигурирование модуля цифрового инспектора завершено.

5.1.9 Установка модуля монитора поставки

Программный компонент (модуль) «Монитор поставки» предназначен для сопровождения процесса поставок МТР, отслеживания движения транспорта и грузов заказчика, управления графиком поставок товаров, и обеспечивает мониторинг и информирование о нарушении сроков поставок.

В данном разделе приведено описание процедур установки модуля «Монитор поставки». Данное описание приведено с учетом того, что закупочный модуль и модуль цифрового инспектора уже установлены. В разделе описаны процедуры установки, которые отличаются (или выполняются дополнительно) от тех, которые приведены в разделах 5.1.7 и 5.1.8.

Для установки модуля монитора поставки необходимо создать структуру данных `inventories/koa/group_vars/app_radar.yml`:

```
---

helm_pi_kubeconfig_path: "/etc/kubernetes/admin.conf"
helm_pi_repositories:
  - name: helm-onpremise
    repo_url: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise

helm_pi_manifests:
  - definition: |
      apiVersion: v1
      kind: Namespace
      metadata:
        name: radar
```

```
# helm releases for upgrade
helm_pi_releases:
  - name: radar-redis
    chart_name: planning
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: radar # omit
    create_namespace: false # omit
    version: "0.0.3"
    value:
      namespace:
        name: radar
      statefulsets:
        redis:
          replicas: 1
          volumes:
          - name: data
            emptyDir:
              medium: ""
          containers:
            redis:
              image: "{{ hosts_p_docker_registry }}/redis"
              imageTag: '7.0.5'
              command:
                - "/bin/sh"
              args:
                - "-c"
                - "redis-server --requirepass RediS123pass --appendonly yes"
              containerPorts:
              - 6379
              memory: "256Mi"
              securityContext:
                readOnlyRootFilesystem: true
              volumeMounts:
                - name: data
                  mountPath: /data
          servicePorts:
          - name: redis
            port: 6379
            targetPort: 6379

  - name: radar-backend
    chart_name: planning
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: radar # omit
    create_namespace: false # omit
    version: "0.0.3"
    value:
      namespace:
        name: radar
      deployments:
        back:
          replicas: 1
          # volumes:
          # - name: upload-providers-xlsx
          #   persistentVolumeClaim:
          #     claimName: "inspector-upload-providers"
          containers:
            radar-backend:
```

```
            image: "{{ hosts_p_docker_registry }}/koa/radar/radar-backend"
            imageTag: 'latest'
            containerPorts:
            - 9201
            memory: "4Gi"
            cpu: "4"
            securityContext:
              readOnlyRootFilesystem: false
            # volumeMounts:
            # - name: upload-providers-xlsx
            #   mountPath: /inspector-backend/upload-providers-xlsx
            env:
              # Настройки бека
              CFG_LOGGER_ENABLED: "1"
              CFG_API_PORT: "9201"
              CFG_UPLOAD_PROVIDERS_XLSX: "upload-providers-xlsx"
              Подключение# к БД локально
              CFG_DB_TYPE: "mysql"
              CFG_DB_HOST: "{{ all_host_mysql }}"
              CFG_DB_DBNAME: "radar"
              CFG_DB_USERNAME: "root"
              CFG_DB_USERPASS: "radar-local-db-pass"
              CFG_DB_DATABASE: "radar"
              CFG_DB_PORT: 3307
              #Keycloak
              # CFG_SSO_REALM: "master"
              # CFG_SSO_BEARER_ONLY: "1"
              # CFG_SSO_AUTH_SERVER: "https://passport-preprod.isource.ru/auth/"
              # CFG_SSO_CLIENT_ID: "radar"
              # CFT_SSO_AUTH_DOCKER_PORT_OVERRIDE: "8080"
              # DaData
              CFG_DADATA_BASE_URL: "https://dadata.ru/api"
              CFG_DADATA_SUGGESTION_BASE_URL: "https://suggestions.dadata.ru/suggestions/api/4_1/rs
"
              CFG_DADATA_API_TOKEN: "6326edda5e3b1926bca85abd3869a9ee440ff0e2"
              CFG_DADATA_SECRET_TOKEN: "3d160ec6b8d62ea6df0553d27d064daf800463df"
              CFG_DADATA_REQUEST_LIMIT: "10000"
              CFG_DADATA_REQUEST_THRESHOLD: "500"
              #OSM
              CFG_OSM_BASE_URL: "https://route-radar.isource.ru"
              CFG_NOMINATIM_BASE_URL: "https://geo-radar.isource.ru"
              CFG_RAILWAY_OSM_BASE_URL: "https://route-railway-radar.isource.ru"
              #SOURCE_IMPORT ДВ(, 773)
              CFG_SOURCE_IMPORT_FETCH_INTERVAL_MIN: "180"
  # отправка email
              CFG_EMAIL_SENDER_HOST: "smtp.yandex.ru"
              CFG_EMAIL_SENDER_PORT: "465"
              CFG_EMAIL_SENDER_PASSWORD: "LCGB+)Qv,6f!@ZW"
              CFG_EMAIL_DEFAULT_SENDER: "testusersystem@yandex.ru"
              CFG_EMAIL_DEFAULT_FROM: "testusersystem@yandex.ru"
              CFG_EMAIL_FEEDBACK_RECEIVERS: "testusersystem@yandex.ru"
              CFG_EMAIL_MAX_ATTACHMENTS_SIZE_MB: "20"
              # Redis
              CFG_REDIS_HOST: "redis"
              CFG_REDIS_PORT: "6379"
              CFG_REDIS_PASS: "RediS123pass"
              CFG_REDIS_DB_ID: "0"
              # Broker (Kafka)
              CFG_BROKER_SERVER: "{{ all_host_kafka }}:9092"
```

```
              CFG_BROKER_CLIENT_ID: "backend"
              CFG_BROKER_TOPIC_PREFIX: "radar"
              # Sync Kafka
              CFG_SYNC_BROKER_SERVER: "127.0.0.1:9492"
              CFG_SYNC_BROKER_CLIENT_ID: "backend"
              CFG_SYNC_BROKER_TOPIC_PREFIX: "sync-develop"
              CFG_ETRAN_EMAILS: ""
              CFG_AUTH_TOKEN_GATE: ""
              # #GATE
              CFG_GATE_API_URL: "http://gate.radar"

        servicePorts:
        - name: radar-backend
          port: 9201
          targetPort: 9201
    hostAliases:
    - ip: 10.50.62.18
      hostnames:
      - keycloak.radar

- name: radar-frontend
  chart_name: planning
  chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
  release_namespace: radar # omit
  create_namespace: false # omit
  version: "0.0.3"
  value:
    namespace:
      name: radar
    hostAliases:
    - ip: 127.0.0.1
      hostnames:
      - cabinet.radar
      - api.radar
      - admin.radar
    - ip: 10.50.62.18
      hostnames:
      - keycloak.radar

    configmaps:
      nginx-config:
        data:
          radar.conf: |
            server {
              listen 80;
              server_name {{ all_target_radar_api }};

              keepalive_timeout 70;
              add_header Strict-Transport-Security max-age=15768000;
              add_header X-XSS-Protection "1; mode=block";
              add_header X-Content-Type-Options nosniff;
              add_header X-Frame-Options DENY;

              location / {
                proxy_read_timeout 120s;
                proxy_pass http://radar-backend:9201;
                proxy_set_header Host $host;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
```

```
            proxy_set_header Connection "upgrade";
          }

          location ^~/public/ {
            expires 10d;
            add_header Cache-Control "Public";
            root /sites/backend/radar-back/test/;
          }
        }

        # кабинет клиента
        server {
          listen 80;
          server_name {{ all_target_radar_cabinet }};

          keepalive_timeout 70;
          add_header Strict-Transport-Security max-age=15768000;
          add_header X-XSS-Protection "1; mode=block";
          add_header X-Content-Type-Options nosniff;
          add_header X-Frame-Options DENY;

          root /usr/share/nginx/html/radar;

          location / {
              try_files $uri $uri/ /index.html;
          }
        }

        # кабинет админа
        server {
          listen 80;
          server_name {{ all_target_radar_admin }};

          keepalive_timeout 70;
          add_header Strict-Transport-Security max-age=15768000;
          add_header X-XSS-Protection "1; mode=block";
          add_header X-Content-Type-Options nosniff;
          add_header X-Frame-Options DENY;

          root /usr/share/nginx/html/radar-admin;

          location / {
              try_files $uri $uri/ /index.html;
          }
        }

    # сайт администратора
        # server {
        #   listen 80;
        #   server_name keycloak.radar;

        #   keepalive_timeout 70;

        #   location / {
        #       proxy_read_timeout 120s;
        #       proxy_pass http://{{ all_host_keycloak }}:80;
        #       proxy_set_header Upgrade           $http_upgrade;
        #       proxy_set_header Connection        "upgrade";
        #       proxy_set_header Host              $host;
```

```
            #       proxy_set_header X-Real-IP         $remote_addr;
            #       proxy_set_header X-Forwarded-For   $proxy_add_x_forwarded_for;
            #       proxy_set_header X-Forwarded-Proto $scheme;
            #       proxy_set_header X-Forwarded-Host  $host;
            #       proxy_set_header X-Forwarded-Port  $server_port;
            #   }
            # }

    deployments:
      front:
        replicas: 1
        volumes:
        - name: nginx-config
          configMap:
            name: nginx-config
            items:
            - key: "radar.conf"
              path: "radar.conf"
        containers:
          nginx:
            image: "{{ hosts_p_docker_registry }}/koa/radar/radar-frontend"
            imageTag: 'latest'
            containerPorts:
            - 80
            memory: "128Mi"
            securityContext:
              readOnlyRootFilesystem: false
            volumeMounts:
            - name: nginx-config
              mountPath: /etc/nginx/conf.d/
              readOnly: true
        servicePorts:
        - name: front
          port: 80
          targetPort: 80
    ingress:
      radar:
        rules:
          - host: "{{ all_target_radar_cabinet }}"
            http:
              paths:
                - path: /
                  pathType: Prefix
                  backend:
                    service:
                      name: front
                      port:
                        number: 80
          - host: "{{ all_target_radar_api }}"
            http:
              paths:
                - path: /
                  pathType: Prefix
                  backend:
                    service:
                      name: front
                      port:
                        number: 80
          - host: "{{ all_target_radar_admin }}"
```

```
            http:
              paths:
                - path: /
                  pathType: Prefix
                  backend:
                    service:
                      name: front
                      port:
                        number: 80
```

Листинг 55: Пример значений структуры данных для `app_radar.yml`

Запуск развертывания[11] осуществляется с целевого АРМ администратора командой (при необходимости, с указанием пароля пользователя `root`):

```
ansible-playbook main.yaml -i inventories/koa/inventory.yml -t app-radar (--ask-become-pass)
```

Листинг 56: Пример инициализации развертывания модуля монитора поставки

Проверка корректности, доступности и готовности к работе программного модуля после установки осуществляется командой:

```
ansible-playbook main.yaml -i inventories/koa/inventory.yml -t check
```

Листинг 57: Пример проверки корректности установки модуля монитора поставки

В указанной структуре данных `app_radar.yml` присутствуют переменные для взаимодействия с почтовым сервисом, а именно:

`CFG_EMAIL_SENDER_HOST`: – определяет адрес (имя) почтового сервера;

`CFG_EMAIL_SENDER_PORT`: – определяет порт взаимодействия почтового сервера;

`CFG_EMAIL_SENDER_PASSWORD`: – определяет пароль пользователя почтового сервера;

`CFG_EMAIL_DEFAULT_SENDER`: – определяет пользователя почтового сервера, для осуществления взаимодействий;

`CFG_EMAIL_DEFAULT_FROM`: – определяет имя пользователя почтового сервера, от имени которого будут осуществляться отправления;

`CFG_EMAIL_FEEDBACK_RECEIVERS`: – определяет имя пользователя почтового сервера, которому будут осуществляться отправления;

[11] Необходимо учитывать, что в текущей структуре данных `app_radar.yml`, применяемой по умолчанию, используются следующие переменные и их значения:
– CFG_DADATA_BASE_URL: "https://dadata.ru/api";
– CFG_DADATA_SUGGESTION_BASE_URL: "https://suggestions.dadata.ru/suggestions/api/4_1/rs";
– CFG_DADATA_API_TOKEN: "6326edda5e3b1926bca85abd3869a9ee440ff0e2";
– CFG_DADATA_SECRET_TOKEN: "3d160ec6b8d62ea6df0553d27d064daf800463df";
– CFG_DADATA_REQUEST_LIMIT: "10000"
Которые определяют атрибуты для подключения к сервису внешнего взаимодействия компании DaData (https://dadata.ru/).
Не исключена ситуация, что потребуется увеличить количество запросов в сутки к сервисам DaData в некоторых вариантах эксплуатации программного модуля «Монитор поставки». Следовательно, возможно переопределение этих параметров. В этом случае потребуется расширить (или оформить специализированные условия) возможности подписки на сервисы DaData, согласно условиям подключения к сервису. Количество бесплатных запросов в сутки составляет не более десяти тысяч. Дополнительная информация по тарифам на количество запросов приведена на странице:
https://dadata.ru/pricing/.

CFG_EMAIL_MAX_ATTACHMENTS_SIZE_MB: – определяет макисмально разрешенный размер вложения.

В варианте установки, описанной в настоящем документе, с целью проверки работоспособности, указанные переменные сконфигурированы для взаимодействия с почтовой службой Yandex.

Успешный результат запроса, демонстрирующий правильную установку, настройку и функционирование модуля приведен на рисунке Рисунок 21

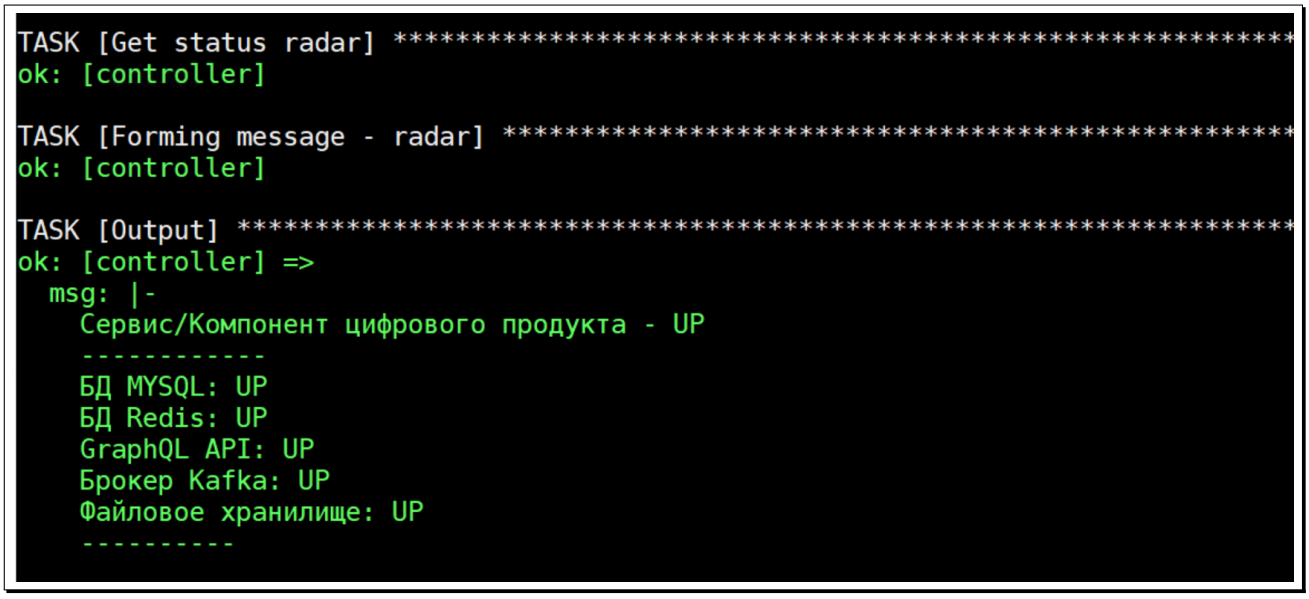


Рисунок 21: Результат проверки корректности установки модуля монитора поставки

На этом установка и настройка модуля монитора поставки завершена.

5.1.10 Установка договорного модуля

Программный компонент (модуль) «Договорной модуль» предназначен сопровождения процесса заключения договора с победителем закупки товаров/работ/услуг. Модуль обеспечивает формирование, согласование и подписание электронных договорных документов.

В данном разделе приведено описание процедур установки модуля «Договорной модуль». Данное описание приведено с учетом того, что остальные модули уже установлены. В разделе описаны процедуры установки, которые отличаются (или выполняются дополнительно) от тех, которые приведены в разделах 5.1.7, 5.1.8 и 5.1.9.

Для установки договорного модуля необходимо создать структуру данных inventories/koa/group_vars/app_processor_documents.yml:

Значения в структуру данных app_processor_documents.yml приведены в листинге ниже:

```
---

helm_pi_kubeconfig_path: "/etc/kubernetes/admin.conf"
helm_pi_repositories:
  - name: helm-onpremise
    repo_url: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
```

```
helm_pi_manifests:
  - definition: |
      apiVersion: v1
      kind: Namespace
      metadata:
        name: documents
  - namespace: documents
    definition: |
      apiVersion: v1
      kind: PersistentVolumeClaim
      metadata:
        name: isource-docgen-claim
      spec:
        storageClassName: nfs-csi
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 4Gi

# helm releases for upgrade
helm_pi_releases:
  - name: docgen
    chart_name: planning
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: documents # omit
    create_namespace: false # omit
    version: "0.0.3"
    value:
      namespace:
        imagePullSecrets: ""
      nodeselector: ""
      tolerations: ""
      deployments:
        docgen-api:
          replicas: 2
          volumes:
            - name: run-nginx
              emptyDir:
                medium: ""
            - name: cache-nginx
              emptyDir:
                medium: ""
            - name: cache-fpm
              emptyDir:
                medium: ""
            - name: templates
              persistentVolumeClaim:
                claimName: "isource-docgen-claim"
          containers:
            nginx:
              image: "{{ hosts_p_docker_registry }}/koa/isource-cmcenter/docgen/nginx"
              imageTag: production-203175-eb6bc0a3
              containerPorts:
                - 80
              memory: "64Mi"
              securityContext:
```

```
              readOnlyRootFilesystem: true
            volumeMounts:
              - name: run-nginx
                mountPath: /var/run/
              - name: cache-nginx
                mountPath: /var/cache/nginx/
          fpm:
            image: "{{ hosts_p_docker_registry }}/koa/isource-cmcenter/docgen/fpm"
            imageTag: production-203175-eb6bc0a3
            containerPorts:
              - 9000
            memory: "256Mi"
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 82
              runAsNonRoot: true
            volumeMounts:
              - name: cache-fpm
                mountPath: /var/www/html/var/cache/
              - name: templates
                mountPath: /var/www/html/var/files/templates
            env:
              APP_DEBUG: 0
              DATABASE_MIDATABASE_MIGRATIONS_URL: "postgresql://docgen:docgen@{{ all_host_postgres
  }}:5432/docgen?serverVersion=12&charset=utf8"
              API_HOST_NAME: https://docgen-api
              DB_HOST: $DB_HOST
            envsecret:
              DATABASE_URL: "postgresql://docgen:docgen@{{ all_host_postgres }}:5432/docgen
  ?serverVersion=12&charset=utf8"
        servicePorts:
          - name: docgen-api
            port: 80
            targetPort: 80
    envForAll:
      APP_LOG_LEVEL: info
      DATABASE_MIGRATIONS_URL: "postgresql://docgen:docgen@{{ all_host_postgres }}:5432/docgen
  ?serverVersion=12&charset=utf8"
- name: sed-front
  chart_name: planning
  chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
  release_namespace: documents # omit
  create_namespace: false # omit
  version: "0.0.3"
  value:
    namespace:
      imagePullSecrets: ""
    nodeselector: ""
    tolerations: ""
    deployments:
      sed-front:
        replicas: 2
        volumes:
        - name: run-nginx
          emptyDir:
            medium: ""
        - name: cache-nginx
          emptyDir:
            medium: ""
```

```
        containers:
          nginx:
            image: "{{ hosts_p_docker_registry }}/koa/isource-cmcenter/sed-front/nginx"
            imageTag: production-213982-b778dd54
            containerPorts:
            - 80
            memory: "128Mi"
            securityContext:
              readOnlyRootFilesystem: true
            volumeMounts:
            - name: run-nginx
              mountPath: /var/run/
            - name: cache-nginx
              mountPath: /var/cache/nginx/
        servicePorts:
        - name: sed-front
          port: 80
          targetPort: 80
    envForAll:
      APP_LOG_LEVEL: info

- name: sed
  chart_name: planning
  chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
  release_namespace: documents # omit
  create_namespace: false # omit
  version: "0.0.3"
  value:
    namespace:
      imagePullSecrets: ""
    nodeselector: ""
    tolerations: ""
    deployments:
      sed-api:
        replicas: 2
        volumes:
        - name: run-nginx
          emptyDir:
            medium: ""
        - name: cache-nginx
          emptyDir:
            medium: ""
        - name: cache-fpm
          emptyDir:
            medium: ""
        containers:
          nginx:
            image: "{{ hosts_p_docker_registry }}/koa/isource-cmcenter/sed/nginx"
            imageTag: production-213981-7721f5bd
            containerPorts:
            - 80
            memory: "64Mi"
            securityContext:
              readOnlyRootFilesystem: true
            volumeMounts:
            - name: run-nginx
              mountPath: /var/run/
            - name: cache-nginx
              mountPath: /var/cache/nginx/
```

```
        fpm:
          image: "{{ hosts_p_docker_registry }}/koa/isource-cmcenter/sed/fpm"
          imageTag: production-213981-7721f5bd
          containerPorts:
          - 9000
          memory: "256Mi"
          securityContext:
            readOnlyRootFilesystem: false
            runAsUser: 82
            runAsNonRoot: true
          volumeMounts:
          - name: cache-fpm
            mountPath: /var/www/html/var/cache/
          envsecret:
            DATABASE_URL: "postgresql://sed:sed@{{ all_host_postgres }}:5432/sed?serverVersion
=12&charset=utf8"
            KEYCLOAK_CLIENT_SECRET: password
            MESSENGER_TRANSPORT_DSN: "amqp://sed:sed@{{ all_host_rabbitmq }}:5672/%2f"
            PROCESSOR_TRANSPORT_DSN: "amqp://processor:processor@{{ all_host_rabbitmq }}:5672/"
            CLICKHOUSE_PASSWORD: password
            DATABASE_MIGRATIONS_URL: "postgresql://sed:sed@{{ all_host_postgres }}:5432/docgen
?serverVersion=12&charset=utf8"
      servicePorts:
      - name: sed-api
        port: 80
        targetPort: 80
    sed-workers:
      replicas: 2
      containers:
        async:
          image: "{{ hosts_p_docker_registry }}/koa/isource-cmcenter/sed/fpm"
          imageTag: production-213981-7721f5bd
          memory: "128Mi"
          cpu: 0.2
          command: ["/var/www/html/scripts/wrapper_script_async.sh"]
          envsecret:
            DATABASE_URL: "postgresql://sed:sed@{{ all_host_postgres }}:5432/sed?serverVersion
=12&charset=utf8"
            KEYCLOAK_CLIENT_SECRET: password
MESSENGER_TRANSPORT_DSN: "amqp://sed:sed@{{ all_host_rabbitmq }}:5672/%2f"
            PROCESSOR_TRANSPORT_DSN: "amqp://processor:processor@{{ all_host_rabbitmq }}:5672/"
            CLICKHOUSE_PASSWORD: password
        create-contracts:
          image: "{{ hosts_p_docker_registry }}/koa/isource-cmcenter/sed/fpm"
          imageTag: production-213981-7721f5bd
          memory: "128Mi"
          cpu: 0.2
          command: ["/var/www/html/scripts/wrapper_script_create_contracts.sh"]
          envsecret:
            DATABASE_URL: "postgresql://sed:sed@{{ all_host_postgres }}:5432/sed?serverVersion
=12&charset=utf8"
            KEYCLOAK_CLIENT_SECRET: password
            MESSENGER_TRANSPORT_DSN: "amqp://sed:sed@{{ all_host_rabbitmq }}:5672/%2f"
            PROCESSOR_TRANSPORT_DSN: "amqp://processor:processor@{{ all_host_rabbitmq }}:5672/"
            CLICKHOUSE_PASSWORD: password
  cronjobs:
    approval-check-timeout:
      image: "{{ hosts_p_docker_registry }}/koa/isource-cmcenter/sed/fpm"
      imageTag: production-213981-7721f5bd
```

```
        schedule: '1 0 * * *'
        completions: '1'
        parallelism: '1'
        activeDeadlineSeconds: '60'
        backoffLimit: '100'
        command: ["php", "/var/www/html/bin/console", "approval:check-timeout"]
        envsecret:
          DATABASE_URL: "postgresql://sed:sed@{{ all_host_postgres }}:5432/sed?serverVersion=12&
  charset=utf8"
          KEYCLOAK_CLIENT_SECRET: password
          MESSENGER_TRANSPORT_DSN: "amqp://sed:sed@{{ all_host_rabbitmq }}:5672/%2f"
          PROCESSOR_TRANSPORT_DSN: "amqp://processor:processor@{{ all_host_rabbitmq }}:5672/"
      logs-clean:
        image: "{{ hosts_p_docker_registry }}/koa/isource-cmcenter/sed/fpm"
        imageTag: production-213981-7721f5bd
        schedule: '0 0 * * *'
        completions: '1'
        parallelism: '1'
        activeDeadlineSeconds: '60'
        backoffLimit: '100'
        command: ["php", "/var/www/html/bin/console", "logs:clean"]
        envsecret:
          DATABASE_URL: "postgresql://sed:sed@{{ all_host_postgres }}:5432/sed?serverVersion=12&
  charset=utf8"
          KEYCLOAK_CLIENT_SECRET: password
          MESSENGER_TRANSPORT_DSN: "amqp://sed:sed@{{ all_host_rabbitmq }}:5672/%2f"
          PROCESSOR_TRANSPORT_DSN: "amqp://processor:processor@{{ all_host_rabbitmq }}:5672/"
    ingress:
      sed:
        rules:
          - host: cmcenter.isource.ru
            http:
              paths:
                - path: /
                  pathType: Prefix
                  backend:
                    service:
                      name: sed-front
                      port:
                        number: 80
                - path: /api
                  pathType: Prefix
                  backend:
                    service:
                      name: sed-api
                      port:
                        number: 80
                - path: /bundles
                  pathType: Prefix
                  backend:
                    service:
                      name: sed-api
                      port:
                        number: 80
    envForAll:
      APP_ENV: prod
      APP_LOG_LEVEL: info
      DATABASE_MIGRATIONS_URL: "postgresql://sed:sed@{{ all_host_postgres }}:5432/sed?serverVersion
  =12&charset=utf8"
```

```
      API_HOST_NAME: "http://{{ all_target_docs_host }}"
      MESSENGER_VHOST: /
      KEYCLOAK_PUBLIC_KEY: "-----BEGIN PUBLIC KEY-----\
  nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0q4nlFEOCJsfAt600G3obfH8WS+
  bAkdgx5Kxdc65iEKvrAHJ8Nl4f3SYYDIjm8z1d0CZpV16HSdFSqJ6j8r32B1/
  VofnJUL1vzNqA0PTloZPJyb8iRVCr1ZhsFxO5hS+dwsX4w1fZ+mUkF8mTkXEohF6tGRUC24ydGgtrMXMh/
  Z3z0lkaHRMrXIhCxX3737izvlr6VFLlEdm1oKugaq+PmfUhD7lY4X+Ishf0+T+VAwth/2Qyiuxv68/
  zd7myftccOkx9CH94caPPZr8qo9a0HSLYycDmKpMpsu7PiE8bzMyMesLBg0SCT/Cnz3zJVIL8tOy7Oru+WnyO/
  QpAHVYaQIDAQAB\n-----END PUBLIC KEY-----"
KEYCLOAK_URL: "http://{{ all_target_docs_host }}"
      KEYCLOAK_REALM: master
      KEYCLOAK_CLIENT_ID: CMCenter-service
      HUB_URL: https://hub-api.etpgpb.ru
      FILESERVICE_IP: $FILESERVICE_IP
      FILESERVICE_URL: http://$FILESERVICE_IP
      FILESERVICE_SOURCE: CMCenter
      FILES_AUTOFILL_DIR: data/files/autoFill
      FILES_LOCAL_TEMP_DIR: var/files/temp
      FILES_ALLOWED_MIME: '["application/msword","application/vnd.openxmlformats-officedocument.
  wordprocessingml.document","application/vnd.ms-excel","application/vnd.openxmlformats-
  officedocument.spreadsheetml.sheet","text/plain","application/rtf","application/zip","application
  /x-zip-compressed","application/x-compressed-zip","application/x-rar-compressed","application/
  octet-stream","application/x-7z-compressed","image/jpeg","image/pjpeg","image/gif","image/png","
  application/pdf"]'
      FILES_ALLOWED_EXTENSION: '["jpeg", "jpg", "png", "pdf", "rtf", "doc", "docx", "xls", "xlsx",
  "zip", "rar"]'
      FILES_ALLOWED_SIZE_IN_BYTES: 20971520
      DOC_GENERATOR_URL: http://docgen-api
      CRYPTOSERVICE_URL: http://api.cryptoservice
      CRYPTOSERVICE_NAMESPACE: cryptoservice
      CRYPTO_IGNORE_CHECK_USER_CERTIFICATE: false
      PROCESSOR_VHOST: processor
      NOTIFICATION_SERVICE_URL: https://notifications.isource.ru
      NOTIFICATION_SERVICE_SOURCE: cmcenter-prod
      NOTIFICATION_SERVICE_DEBUG_TO: null
      DB_HOST: "{{ all_host_postgres }}"
      MESSENGER_TRANSPORT_DSN_SERVER: "{{ all_host_rabbitmq }}"
      PROCESSOR_TRANSPORT_DSN_SERVER: "{{ all_host_rabbitmq }}"
      CORS_ALLOW_ORIGIN: /^https:\/\/cmcenter\.isource\.ru$/gm
      FRONTEND_BASE_URL: "http://{{ all_target_docs_host }}"
      ANTIVIRUS_URL: http://10.29.1.101:8085
      CLICKHOUSE_HOST: test
      CLICKHOUSE_DATABASE: test
      CLICKHOUSE_PORT: 80
      CLICKHOUSE_USERNAME: test
```

Листинг 58: Пример значений для структуры данных для `app_processor_documents.yml`

Запуск развертывания осуществляется с целевого АРМ администратора командой (при необходимости, с указанием пароля пользователя `root`):

```
ansible-playbook main.yaml -i inventories/koa/inventory.yml -t app-processor-docs (--ask-become-pass)
```

Листинг 59: Пример инициализации развертывания модуля монитора поставки

Проверка корректности, доступности и готовности к работе программного модуля после установки осуществляется командой:

```
ansible-playbook main.yaml -i inventories/koa/inventory.yml -t check
```

Листинг 60: Пример проверки корректности установки договорного модуля

Результат запроса, демонстрирующий текущую конфигурацию установки, настройки и функционирования модуля приведен на рисунке Рисунок 22

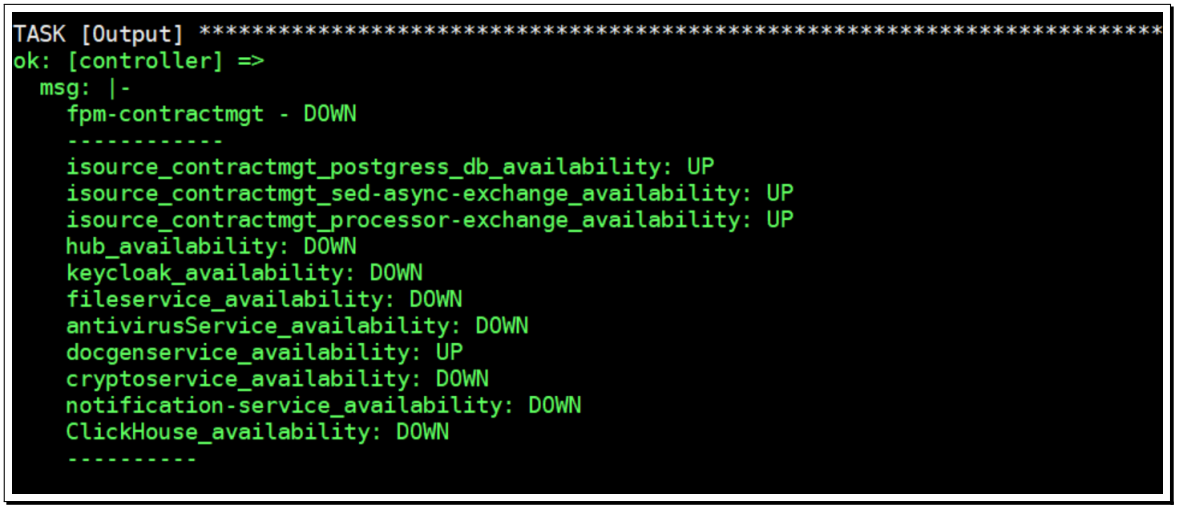


Рисунок 22: Результат проверки корректности установки договорного модуля

На этом установка и настройка договорного модуля завершена.

5.1.11 Установка модуля планирования

Программный компонент (модуль) «Модуль Планирования» предназначен для сопровождения процесса планирования процедур закупок, и обеспечивает формирование и согласование перечня планируемых к закупке товаров/работ/услуг (ТРУ) с выполнением контроля бюджетных лимитов.

В данном разделе приведено описание процедур установки модуля планирования. Данное описание приведено с учетом того, что остальные модули уже установлены. В разделе описаны процедуры установки, которые отличаются (или выполняются дополнительно) от тех, которые приведены в разделах 5.1.7, 5.1.8, 5.1.9 и 5.1.10.

Для установки договорного модуля необходимо создать структуру данных `inventories/koa/group_vars/app_processor_planning.yml`:

Значения в структуру данных `app_processor_planning.yml` приведены в листинге ниже:

```
---

helm_pi_kubeconfig_path: "/etc/kubernetes/admin.conf"
helm_pi_repositories:
  - name: helm-onpremise
    repo_url: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise


helm_pi_manifests:
  - definition: |
      apiVersion: v1
```

```
      kind: Namespace
      metadata:
        name: planning
  - namespace: planning
    definition: |
      apiVersion: v1
      kind: PersistentVolumeClaim
      metadata:
        name: isource-planning-uploads-claim
      spec:
        storageClassName: nfs-csi
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 4Gi

  - namespace: planning
    definition: |
      apiVersion: v1
      kind: PersistentVolumeClaim
      metadata:
        name: isource-planning-reports-claim
      spec:
        storageClassName: nfs-csi
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 4Gi

# helm releases for upgrade
helm_pi_releases:
  - name: planning
    chart_name: planning
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: planning # omit
    create_namespace: false # omit
    version: "0.0.3"
    value:
      namespace:
        name: planning

      deployments:
        front:
          replicas: 2
          volumes:
          - name: run-nginx
            emptyDir:
              medium: ""
          - name: cache-nginx
            emptyDir:
              medium: ""
          containers:
            nginx:
              image: "{{ hosts_p_docker_registry }}/koa/isource-planning/back/nginx"
              imageTag: 'dev-233057-ff1faaa5'
              containerPorts:
              - 80
```

```
            memory: "64Mi"
            securityContext:
              readOnlyRootFilesystem: true
            volumeMounts:
            - name: run-nginx
              mountPath: /var/run/
            - name: cache-nginx
              mountPath: /var/cache/nginx/
        servicePorts:
        - name: front
          port: 80
          targetPort: 80

      back:
        replicas: 2
        volumes:
        - name: run-nginx
          emptyDir:
            medium: ""
        - name: cache-nginx
          emptyDir:
            medium: ""
        - name: cache-fpm
          emptyDir:
            medium: ""
        - name: uploads
          persistentVolumeClaim:
            claimName: "isource-planning-uploads-claim"
        - name: reports
          persistentVolumeClaim:
            claimName: "isource-planning-reports-claim"
        containers:
          nginx:
            image: "{{ hosts_p_docker_registry }}/koa/isource-planning/back/nginx"
            imageTag: 'dev-233057-ff1faaa5'
            containerPorts:
            - 80
            memory: "64Mi"
            securityContext:
              readOnlyRootFilesystem: true
            volumeMounts:
            - name: run-nginx
              mountPath: /var/run/
            - name: cache-nginx
              mountPath: /var/cache/nginx/
    fpm:
            image: "{{ hosts_p_docker_registry }}/koa/isource-planning/back/fpm"
            imageTag: 'dev-233057-ff1faaa5'
            containerPorts:
            - 9000
            memory: "256Mi"
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 82
              runAsNonRoot: true
            volumeMounts:
            - name: cache-fpm
              mountPath: /var/www/html/var/cache/
            - name: uploads
```

```
            mountPath: /var/www/html/public/uploads/
          - name: reports
            mountPath: /var/www/html/var/reports/
          env:
            APP_ENV: prod
            APP_SECRET: d6e737a8bd191079490670aaf3af3acb
            KESL_HOST: http://10.29.1.102:8085 #http://10.29.1.101:8085
            SPRING_PROFILES_ACTIVE: dev
            CAMUNDA_HOST: http://camunda
            REDIS_HOST: "redis"
            REDIS_PORT: 6379
            DATABASE_NAME: planning-prod-test
            TOKEN_REQUEST_HEADER: Authorization
            # KEYCLOAK
            KEY_CLOAK_HOST: https://passport.isource.ru
            KEYCLOAK_GRANT_TYPE: client_credentials
            KEYCLOAK_CLIENT_ID: plan-service
            KEYCLOAK_REALM: master
            # hub
            HUB_HOST: https://hub-api.etpgpb.ru
            ELEMENT_HOST: http://element.isource.ru
            PRODUCTION_NEED_ERROR_REPORTS_DIR: var/reports/
            APIDOC_TITLE: "Planning API"
            APIDOC_DESCRIPTION: "Planning API Documentation"
            APIDOC_VERSION: 1.0.0
            APIDOC_SERVER_URL: "{{ all_target_planning_host }}"
            CORS_ALLOW_ORIGIN: "^http://{{ all_target_planning_host }}$"
            TRUSTED_PROXIES: "10.29.1.128/25"
          envsecret:
            KEYCLOAK_CLIENT_SECRET: $KEYCLOAK_CLIENT_SECRET
            EXTERNAL_TOKEN: ?84_]^D?Rs4sW!:,f6FU*X=GQBs)j0A^
            DATABASE_URL: "postgresql://planning:planning@{{ all_host_postgres }}:5432/planning
?serverVersion=12&charset=utf8"
            MESSENGER_TRANSPORT_DSN: "amqp://processor:processor@{{ all_host_rabbitmq }}:5672/
processor/messages"
      servicePorts:
      - name: api
        port: 80
        targetPort: 80

    consumer:
      replicas: 1
      volumes:
      - name: cache-fpm
        emptyDir:
          medium: ""
      - name: uploads
        persistentVolumeClaim:
          claimName: "isource-planning-uploads-claim"
      - name: reports
        persistentVolumeClaim:
          claimName: "isource-planning-reports-claim"
      containers:
        fpm:
          image: "{{ hosts_p_docker_registry }}/koa/isource-planning/back/fpm"
          imageTag: 'dev-233057-ff1faaa5'
          memory: "256Mi"
          command: ["php", "bin/console", "messenger:consume", "processor.create_need_result_v1.
planning", "processor.state_changed_v1.planning"]
```

```
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 82
              runAsNonRoot: true
            volumeMounts:
            - name: cache-fpm
              mountPath: /var/www/html/var/cache/
            - name: uploads
              mountPath: /var/www/html/public/uploads/
            - name: reports
              mountPath: /var/www/html/var/reports/
            env:
              APP_ENV: dev
              APP_SECRET: d6e737a8bd191079490670aaf3af3acb
              KESL_HOST: http://10.29.1.101:8085
              SPRING_PROFILES_ACTIVE: dev
              CAMUNDA_HOST: http://camunda

              REDIS_HOST: "redis"
              REDIS_PORT: 6379
              DATABASE_NAME: planning-prod

              TOKEN_REQUEST_HEADER: Authorization
  # KEYCLOAK
              KEY_CLOAK_HOST: https://passport.isource.ru
              KEYCLOAK_GRANT_TYPE: client_credentials
              KEYCLOAK_CLIENT_ID: plan
              KEYCLOAK_REALM: master
              # hub
              HUB_HOST: https://hub-api.etpgpb.ru
              ELEMENT_HOST: https://element.isource.ru
              PRODUCTION_NEED_ERROR_REPORTS_DIR: var/reports/
              APIDOC_TITLE: "Planning API"
              APIDOC_DESCRIPTION: "Planning API Documentation"
              APIDOC_VERSION: 1.0.0
              APIDOC_SERVER_URL: "{{ all_target_planning_host }}"

              CORS_ALLOW_ORIGIN: "^http://{{ all_target_planning_host }}$"
              TRUSTED_PROXIES: "10.29.1.128/25"
            envsecret:
              KEYCLOAK_CLIENT_SECRET: $KEYCLOAK_CLIENT_SECRET
              EXTERNAL_TOKEN: ?84_]^D?Rs4sW!:,f6FU*X=GQBs)j0A^
              DATABASE_URL: "postgresql://planning:planning@{{ all_host_postgres }}:5432/planning
  ?serverVersion=12&charset=utf8"
              MESSENGER_TRANSPORT_DSN: "amqp://processor:processor@{{ all_host_rabbitmq }}:5672/
  processor/messages"

      camunda:
        replicas: 1
        volumes:
        - name: camunda-tmp
          emptyDir:
            medium: ""
        - name: application
          configMap:
            name: camunda-application
            items:
            - key: "application.yaml"
              path: "application.yaml"
```

```
        containers:
          camunda:
            image: "{{ hosts_p_docker_registry }}/koa/isource-planning/camunda"
            imageTag: 'latest'
            containerPorts:
            - 8080
            memory: "2Gi"
            securityContext:
              readOnlyRootFilesystem: true
            volumeMounts:
            - name: camunda-tmp
              mountPath: /tmp
            - name: application
              mountPath: app/resources/config/
              readOnly: true
        servicePorts:
        - name: camunda
          port: 80
          targetPort: 8080

    statefulsets:
      redis:
        replicas: 1
        volumes:
        - name: data
          emptyDir:
            medium: ""
        containers:
          redis:
            image: "{{ hosts_p_docker_registry }}/redis"
            imageTag: '7.0.5'
            containerPorts:
            - 6379
            memory: "128Mi"
            securityContext:
              readOnlyRootFilesystem: true
            volumeMounts:
              - name: data
                mountPath: /data
        servicePorts:
        - name: redis
          port: 6379
          targetPort: 6379
    cronjobs:
      inspect-files:
        schedule: "*/2 * * * *"
        completions: 1
        parallelism: 1
        activeDeadlineSeconds: 60
        backoffLimit: 100
        image: "{{ hosts_p_docker_registry }}/koa/isource-planning/back/fpm"
        imageTag: 'dev-233057-ff1faaa5'
        command: ["php", "bin/console", "planning:file:inspect"]
        volumes:
        - name: uploads
          persistentVolumeClaim:
            claimName: "isource-planning-uploads-claim"
        volumeMounts:
        - name: uploads
```

```
          mountPath: /var/www/html/public/uploads/
        env:
          APP_ENV: dev
          APP_SECRET: d6e737a8bd191079490670aaf3af3acb
          KESL_HOST: http://10.29.1.101:8085
          TRUSTED_PROXIES: "10.29.1.128/25"
        envsecret:
          DATABASE_URL: "postgresql://planning:planning@{{ all_host_postgres }}:5432/planning
?serverVersion=12&charset=utf8"

    configmaps:
      camunda-application:
        data:
          application.yaml: |
            camunda:
              bpm:
                admin-user:
                  id: admin
                  password: xO4hvKWMn21Eug5W
                job-execution:
                  enabled: true
                default-serialization-format: application/json
            spring:
              application:
                name: etp-gpb
              datasource:
                type: com.zaxxer.hikari.HikariDataSource
                url: jdbc:postgresql://{{ all_host_postgres }}:5432/camunda
                username: camunda
                password: camunda
                hikari:
                  poolName: Hikari
                  auto-commit: false
              h2:
                console:
                  enabled: false
              jpa:
                database-platform: io.github.jhipster.domain.util.FixedPostgreSQL10Dialect
                show-sql: false
            server:
              port: 8080
            application:
              purchasing-department-url: http://api:80
              purchasing-department-token: ?84_]^D?Rs4sW!:,f6FU*X=GQBs)j0A^

    ingress:
      planning:
        rules:
          - host: "{{ all_target_planning_host }}"
            http:
              paths:
                - path: /
                  pathType: Prefix
                  backend:
                    service:
                      name: front
                      port:
                        number: 80
                - path: /api
```

```
                    pathType: Prefix
                    backend:
                      service:
                        name: api
                        port:
                          number: 80
```

Листинг 61: Пример значений для структуры данных для `app_processor_planning.yml`

Запуск развертывания осуществляется с целевого АРМ администратора командой (при необходимости, с указанием пароля пользователя `root`):

```
ansible-playbook main.yaml -i inventories/koa/inventory.yml -t app-processor-docs (--ask-become-
  pass)
```

Листинг 62: Пример инициализации развертывания модуля монитора поставки

На этом установка и настройка модуля планирования поставки завершена.

5.1.12 Установка модуля «Портал Партнер»

Программный компонент (модуль) «Портал Партнер» предназначен для организации и проведения процедур купли-продажи материально-технических ресурсов между участниками закрытого контура и на открытом рынке.

В данном разделе приведено описание процедур установки модуля «Портал Партнер». Данное описание приведено с учетом того, что остальные модули уже установлены. В разделе описаны процедуры установки, которые отличаются (или выполняются дополнительно) от тех, которые приведены в разделах 5.1.7, 5.1.8, 5.1.9, 5.1.10 и 5.1.11.

Для установки договорного модуля необходимо создать структуру данных `inventories/koa/group_vars/app_portal_partner.yml`:

Значения в структуру данных `app_portal_partner.yml` приведены в листинге ниже:

```
---

helm_pi_kubeconfig_path: "/etc/kubernetes/admin.conf"
helm_pi_repositories:
  - name: helm-onpremise
    repo_url: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise


helm_pi_manifests:
  - definition: |
      apiVersion: v1
      kind: Namespace
      metadata:
        name: portal-partner
  - definition: |
      apiVersion: v1
      kind: Namespace
      metadata:
        name: vsklad
  - namespace: portal-partner
```

```
  definition: |
    apiVersion: v1
    kind: PersistentVolumeClaim
    metadata:
      name: portal-partner-backend-www
    spec:
      storageClassName: nfs-csi
      accessModes:
        - ReadWriteMany
      resources:
        requests:
          storage: 4Gi
- namespace: portal-partner
  definition: |
    apiVersion: v1
    kind: PersistentVolumeClaim
    metadata:
      name: portal-partner-backend-storage
    spec:
      storageClassName: nfs-csi
      accessModes:
        - ReadWriteMany
      resources:
        requests:
          storage: 4Gi
- namespace: portal-partner
  definition: |
    apiVersion: v1
    kind: PersistentVolumeClaim
    metadata:
      name: portal-partner-front-shared
    spec:
      storageClassName: nfs-csi
      accessModes:
        - ReadWriteMany
      resources:
        requests:
          storage: 4Gi

- namespace: vsklad
  definition: |
    apiVersion: v1
    kind: PersistentVolumeClaim
    metadata:
      name: portal-partner-vsklad-www
    spec:
      storageClassName: nfs-csi
      accessModes:
        - ReadWriteMany
      resources:
        requests:
          storage: 4Gi
- namespace: vsklad
  definition: |
    apiVersion: v1
    kind: PersistentVolumeClaim
    metadata:
      name: portal-partner-vsklad-storage
    spec:
```

```
        storageClassName: nfs-csi
        accessModes:
          - ReadWriteMany
        resources:
          requests:
            storage: 4Gi
# helm releases for upgrade
helm_pi_releases:
  - name: partner
    chart_name: planning
    chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
    release_namespace: portal-partner # omit
    create_namespace: false # omit
    version: "0.0.3"
    value:
      namespace:
        name: portal-partner
      statefulsets:
        redis:
          replicas: 1
          volumes:
          - name: data
            emptyDir:
              medium: ""
          containers:
            redis:
              image: "{{ hosts_p_docker_registry }}/redis"
              imageTag: '7.0.5'
              containerPorts:
              - 6379
              memory: "128Mi"
              securityContext:
                readOnlyRootFilesystem: true
              volumeMounts:
                - name: data
                  mountPath: /data
          servicePorts:
          - name: redis
            port: 6379
            targetPort: 6379

      deployments:
        fpm:
          replicas: 1
          volumes:
          - name: var-log
            emptyDir: {}
          - name: www-storage
            emptyDir: {}
          - name: public-shared
            emptyDir: {}
          - name: www-certs
            emptyDir: {}
          containers:
            fpm:
              image: "{{ hosts_p_docker_registry }}/koa/isource-partner/fpm"
              imageTag: 'latest'
              containerPorts:
              - 9000
```

```
            memory: "512Mi"
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 0
              runAsNonRoot: false
            volumeMounts:
            - name: var-log
              mountPath: /var/www/html/var/log
            - name: www-storage
              mountPath: /var/www/storage
            - name: public-shared
              mountPath: /var/www/html/public/shared
            - name: www-certs
              mountPath: /var/www/certs
            env:
              APP_CONTAINER_NAME: fpm
  cron:
            image: "{{ hosts_p_docker_registry }}/koa/isource-partner/fpm"
            imageTag: 'latest'
            memory: "512Mi"
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 0
              runAsNonRoot: false
            command:
              - "/bin/sh"
            args:
              - "-c"
              - "/var/www/cron/start-cron-alpine.sh"
              - "cron.crontab"
              - "www-data"
            volumeMounts:
            - name: var-log
              mountPath: /var/www/html/var/log
            - name: www-storage
              mountPath: /var/www/storage
            - name: public-shared
              mountPath: /var/www/html/public/shared
            - name: www-certs
              mountPath: /var/www/certs
            env:
              APP_CONTAINER_NAME: cron

          subscriber:
            image: "{{ hosts_p_docker_registry }}/koa/isource-partner/fpm"
            imageTag: 'latest'
            memory: "512Mi"
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 0
              runAsNonRoot: false
            command:
              - "/bin/sh"
            args:
              - "-c"
              - >-
                chmod +x /var/www/html/start-rabbit-readers.sh &&
                /var/www/html/start-rabbit-readers.sh
            volumeMounts:
```

```
            - name: var-log
              mountPath: /var/www/html/var/log
            - name: www-storage
              mountPath: /var/www/storage
            - name: www-certs
              mountPath: /var/www/certs
            env:
              APP_CONTAINER_NAME: subscriber
              POOL_SIZE: 10
              WORKERS_COUNT: 20
        servicePorts:
        - name: fpm
          port: 9000
          targetPort: 9000

      nginx:
        replicas: 1
        containers:
          nginx:
            image: "{{ hosts_p_docker_registry }}/koa/isource-partner/nginx"
            imageTag: 'latest'
            containerPorts:
            - 80
            memory: "512Mi"
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 0
              runAsNonRoot: false
        servicePorts:
        - name: nginx
          port: 80
          targetPort: 80
          envForAll:
      ## Postgres
      DB_HOST: "{{ all_host_postgres }}"
      DB_PORT: 5432
      DB_NAME: partner
      # Кредлы для связки приложение <-> БД:
      DB_USER: partner
      DB_PASS: partner
      # Кредлы для миграций:
      DB_OWNER: partner
      DB_OWNERPASS: partner
      # Кредлы для симфони
      DATABASE_URL: postgresql://partner:partner@{{ all_host_postgres }}:5432/partner?serverVersion
=10.12&charset=utf8
      DATABASE_MIGRATIONS_URL: postgresql://partner:partner@{{ all_host_postgres }}:5432/partner?
serverVersion=10.12&charset=utf8
      ## RabbitMQ
      RABBITMQ_HOST: "{{ all_host_rabbitmq }}"
      RABBITMQ_PORT: 5672
      # Настройки для подключения приложения к кролю. Можно создать
      RABBITMQ_USER: partner
      RABBITMQ_PASSWORD: partner
      RABBITMQ_VHOST: "partner"
      RABBITMQ_HUB_VHOST: "hub"
      RABBITMQ_RESERVE_VHOST: "reserve"
      RABBITMQ_VSKLAD_VHOST: "vsklad"
      ## Main env
```

```
      CONTAINER_STORAGE_PATH: /var/www/storage
      SYMFONY_PHPUNIT_DIR: /var/www/html/vendor/phpunit
      # Токен гитхаба:
      CI_GITHUB_OAUTH_TOKEN: ""
      # Где лежит дамп:
      POSTGRES_DUMP_PATH: /home/sm/rep/etp/partner/data/partner-test.sql.gz
      XDEBUG_TRIGGER: StartXdebug
      KEYCLOAK_BASE_URI: 'https://id-preprod.etpgpb.ru/'
      KEYCLOAK_REALM: 'master'
      KEYCLOAK_CLIENT_ID: 'NVI-preprod'
      KEYCLOAK_CLIENT_SECRET: $KEYCLOAK_CLIENT_SECRET
      KEYCLOAK_TOKEN_LIFETIME_SECONDS: '60'
      KEYCLOAK_ADMIN_USERNAME: $KEYCLOAK_ADMIN_USERNAME
      KEYCLOAK_ADMIN_PASSWORD: $KEYCLOAK_ADMIN_PASSWORD
      KEYCLOAK_URL_PART: '/consents'

      NOTIFICATION_BASE_URI: http://not
      NOTIFICATION_ADMIN_USERNAME: NOTIFICATION_ADMIN_USERNAME
      NOTIFICATION_ADMIN_PASSWORD: NOTIFICATION_ADMIN_PASSWORD

      CHAT_BASE_URI: http://char
      CHAT_ADMIN_USERNAME: CHAT_ADMIN_USERNAME
      CHAT_ADMIN_PASSWORD: CHAT_ADMIN_PASSWORD

      IMG_PROXY_CDN_POSTFIX: "ru-cdn"
      IMG_PROXY_KEY: test
      IMG_PROXY_SALT: test

      PRIMA_INFORM_USER: test
      PRIMA_INFORM_PASS: test

      DADATA_API_KEY: test
      DADATA_SECRET_KEY: test

      SBERBANK_PAYMENTS_LOGIN: test
      SBERBANK_PAYMENTS_PASSWORD: test
      SBER_BUSINESS_CLIENT_SECRET: test
      SBER_BUSINESS_PAYEE_ACCOUNT: test
      SBER_BUSINESS_SSL_KEY_PASSWORD: test

      APP_ENV: preprod
      MAILER_DSN: test
      FRONTEND_HOST: front
- name: partner-front
  chart_name: planning
  chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
  release_namespace: portal-partner # omit
  create_namespace: false # omit
  version: "0.0.3"
  value:
    namespace:
      name: portal-partner
    deployments:
      front:
        replicas: 1
        volumes:
        - name: front-shared
          persistentVolumeClaim:
            claimName: "portal-partner-front-shared"
```

```
        containers:
          front:
            image: "{{ hosts_p_docker_registry }}/partner-front/front"
            imageTag: '0.1'
            containerPorts:
            - 80
            memory: "512Mi"
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 0
              runAsNonRoot: false
            volumeMounts:
            - name: front-shared
              mountPath: /var/www/html/public/shared
        servicePorts:
        - name: front
          port: 80
          targetPort: 80
      front-ssr:
        replicas: 1
        containers:
          front-ssr:
            image: "{{ hosts_p_docker_registry }}/koa/partner-front/ssr"
            imageTag: '0.1'
            containerPorts:
            - 4000
            memory: "128Mi"
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 0
              runAsNonRoot: false
        servicePorts:
        - name: front-ssr
          port: 4000
          targetPort: 4000
    ingress:
      sed:
        rules:
          - host: reserve.isource.ru
            http:
              paths:
                - path: /
                  pathType: Prefix
                  backend:
                    service:
                      name: front
                      port:
                        number: 80
                - path: /market/api
                  pathType: Prefix
                  backend:
                    service:
                      name: nginx
                      port:
                        number: 80
                - path: /api
                  pathType: Prefix
                  backend:
                    service:
```

```
                      name: front
                      port:
                        number: 80
                - path: /health
                  pathType: Prefix
                  backend:
                    service:
                      name: nginx
                      port:
                        number: 80
- name: vsklad
  chart_name: planning
  chart_ref: http://{{ all_host_nexus_repository }}:8081/repository/helm-onpremise
  release_namespace: vsklad # omit
  create_namespace: false # omit
  version: "0.0.3"
  value:
    namespace:
      name: vsklad
    deployments:
      vsklad-fpm:
        replicas: 1
        volumes:
        # - name: vsklad-www
        #   persistentVolumeClaim:
        #     claimName: "portal-partner-vsklad-www"
        - name: vsklad-storage
          persistentVolumeClaim:
            claimName: "portal-partner-vsklad-storage"
        containers:
          vsklad-fpm:
            image: "{{ hosts_p_docker_registry }}/koa/newetp/vsklad/fpm"
            imageTag: 'f124ebbb'
            # command:
            #   - "/usr/local/bin/init"
            # args:
            #   - "fpm"
            containerPorts:
            - 9000
            memory: "512Mi"
            securityContext:
              readOnlyRootFilesystem: false
              runAsUser: 0
              runAsNonRoot: false
            volumeMounts:
            # - name: vsklad-www
            #   mountPath: /var/www/html
            - name: vsklad-storage
              mountPath: /var/www/html/storage
        servicePorts:
        - name: fpm
          port: 9000
          targetPort: 9000
        vsklad-nginx:
        replicas: 1
        volumes:
        - name: vsklad-storage
          persistentVolumeClaim:
            claimName: "portal-partner-vsklad-storage"
```

```
      containers:
        vsklad-nginx:
          image: "{{ hosts_p_docker_registry }}/koa/newetp/vsklad/nginx-back"
          imageTag: 'f124ebbb'
          containerPorts:
          - 80
          memory: "512Mi"
          securityContext:
            readOnlyRootFilesystem: false
            runAsUser: 0
            runAsNonRoot: false
          volumeMounts:
          - name: vsklad-storage
            mountPath: /var/www/html/storage
      servicePorts:
      - name: nginx
        port: 80
        targetPort: 80

    vsklad-cron:
      replicas: 1
      # volumes:
      # - name: vsklad-www
      #   persistentVolumeClaim:
      #     claimName: "portal-partner-vsklad-www"
      containers:
        vsklad-cron:
          image: "{{ hosts_p_docker_registry }}/koa/newetp/vsklad/cron"
          imageTag: 'f124ebbb'
          # command:
          #   - "/bin/sh"
          # args:
          #   - "-c"
          #   - "/var/www/cron/start-cron-alpine.sh"
          #   - "cron.crontab"
          #   - "www-data"
          memory: "512Mi"
          securityContext:
            readOnlyRootFilesystem: false
            runAsUser: 0
            runAsNonRoot: false
          # volumeMounts:
          # - name: vsklad-www
          #   mountPath: /var/www/html
  ingress:
    sed:
      rules:
        - host: vsklad.isource.ru
          http:
            paths:
              - path: /
                pathType: Prefix
                backend:
                  service:
                    name: nginx
                    port:
                      number: 80

  envForAll:
```

```
        #XDEBUG_TRIGGER=StartXdebug
        # TODO Если хотим тут что-то поменять локально. Создаем копию этог офайла с именем .env
        ## Править: TODO: Заменить путь на свой и указать токен
        # Токен гитхаба:
        CI_GITHUB_OAUTH_TOKEN: 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
        # Где лежит дамп:
        POSTGRES_DUMP_PATH: ./data/partner-test.sql.gz
        # Путь для локального файлового хранилища
        CONTAINER_STORAGE_PATH: /var/www/storage
        # Путь до phpunit
        SYMFONY_PHPUNIT_DIR: /var/www/html/vendor/phpunit
        # Имя проекта
        PROJECT_NAME: vsklad
        DB_NAME: vsklad
        DB_PORT: 5432
        DB_HOST: "{{ all_host_postgres }}"
        DB_GROUP_RO: vsklad
        DB_GROUP_RW: vsklad
        DB_USER: vsklad
        DB_PASS: vsklad
        DB_USER_OWNER: vsklad
        DB_USER_OWNER_PASS: vsklad
        RABBITMQ_EXCHANGE_WAREHOUSE: vsklad.warehouse.fanout
        RABBITMQ_EXCHANGE_PROVIDER: vsklad.provider.fanout
        RABBITMQ_EXCHANGE_DICTIONARY: vsklad.dictionary.fanout
        RABBITMQ_EXCHANGE_ATTACHMENT: vsklad.attachment.topic
        RABBITMQ_EXCHANGE_ATTRIBUTE: vsklad.attribute.topic
        RABBITMQ_EXCHANGE_RESERVE: vsklad.reserve.topic
        RABBITMQ_EXCHANGE_NOMENCLATURE: vsklad.nomenclature.topic
        KEYCLOAK_BASE_URI: 'http://{{ all_host_keycloak }}/'
        KEYCLOAK_REALM: 'master'
        KEYCLOAK_CLIENT_ID: 'vsklad'
        KEYCLOAK_CLIENT_SECRET: 'vsklad'
        KEYCLOAK_ADMIN_USERNAME: 'vskald-admin'
        KEYCLOAK_ADMIN_PASSWORD: "vskald-admin"
        KEYCLOAK_TOKEN_LIFETIME_SECONDS: '60'
        RABBITMQ_HOST: "{{ all_host_rabbitmq }}"
        RABBITMQ_PORT: 5672
        RABBITMQ_USER: 'partner'
        RABBITMQ_PASSWORD: 'partner'
        RABBITMQ_VHOST: vsklad
        RABBITMQ_QUEUE_RESERVE_CREATE: reserve_create_q
        RABBITMQ_QUEUE_RESERVE_RELEASE: reserve_release_q
        RABBITMQ_EXCHANGE: notification_e
        DATABASE_URL: "pgsql://vsklad:vsklad@{{ all_host_postgres }}:5432/vsklad?serverVersion=10.12&
    charset=utf8"
        DATABASE_MIGRATIONS_URL: "pgsql://vsklad:vsklad@{{ all_host_postgres }}:5432/vsklad?
    serverVersion=10.12&charset=utf8"
```

Листинг 63: Пример значений для структуры данных для `app_portal_partner.yml`

Запуск развертывания осуществляется с целевого АРМ администратора командой (при необходимости, с указанием пароля пользователя `root`):

```
ansible-playbook main.yaml -i inventories/koa/inventory.yml -t portal-partner (--ask-become-pass)
```

Листинг 64: Пример инициализации развертывания модуля «Портал Партнер»

На этом установка и настройка модуля «Портал Партнер» поставки завершена.

5.2 Установка провайдера идентификации и аутентификации KeyCloak без применения контейнеров

5.2.1 Общие сведения о KeyCloak

KeyCloak — это программное средство для централизованного управления идентификацией, аутентификацией и разграничением доступа с открытым исходным кодом, предназначенное для использования в ИС где могут использоваться элементы микросервисной архитектуры.

KeyCloak предлагает такие функции, как единый вход (SSO), брокерская идентификация и социальный вход в систему, федерация пользователей, клиентские адаптеры, консоль администратора и консоль управления учетными записями.

Базовый функционал, поддерживаемый в KeyCloak:

- Single-Sign On and Single-Sign Out для браузерных приложений;
- поддержка протоколов аутентификации OpenID/OAuth 2.0/SAML;
- Identity Brokering – аутентификация с помощью внешних OpenID Connect или SAML идентификационных провайдеров;
- Social Login – поддержка Google, GitHub, Facebook, Twitter для идентификации пользователей;
- User Federation – синхронизация пользователей из LDAP и Active Directory серверов и других идентификационных провайдеров;
- Kerberos bridge – использование Kerberos сервера для автоматической аутентификации пользователей;
- Admin Console — консоль администратора для единого управления настройками и параметрами решения через Web.
- Account Management Console – консоль пользователя, для самостоятельного управления профилем пользователей;
- 2FA Authentication – поддержка TOTP/HOTP с помощью Google Authenticator или FreeOTP;
- Login Flows – поддержка самостоятельной регистрация пользователей, восстановление и сброс пароля (при необходимости);
- Session Management – централизованное управление из единой точки сессиями пользователей для администраторов;
- Token Mappers – привязка атрибутов пользователей, ролей и иных требуемых атрибутов в токены;
- гибкое управление политиками через realm, application и пользователей; CORS Support – клиентские адаптеры имеют встроенную поддержку CORS (поддержка взаимного обмена ресурсами);

- Service Provider Interfaces (SPI) – большое количество SPI, позволяющих настраивать различные аспекты работы сервера: потоки аутентификации, идентификационных провайдеров, сопоставление протоколов и многое другое;
- интерфейсы поддержки для широкого спектра языков;
- поддержка работы с различными приложениями, поддерживающими протоколы OpenID Connect Relying Party library или SAML 2.0 Service Provider Library;
- поддержка расширений.

Для процессов CI/CD, а так же автоматизации процессов управления, в KeyCloak, поддерживается REST API/ JAVA API.

Документация на программные интерфейсы KeyCloak доступна в электронном виде:

https://www.keycloak.org/docs/18.0/api_documentation/.

Документация на KeyCloak доступна в электронном виде:

https://www.keycloak.org/documentation.

5.2.2 Вариант развертывания KeyCloak и СУБД PostgreSQL без применения контейнеров

В настоящем разделе описаны дополнительные варианты развертывания единого провайдера аутентификации KeyCloak и СУБД PostgreSQL в его интересах, без применения контейнерной виртуализации. Этот способ развертывания считается дополнительным и не рекомендуется к применению по умолчанию. В этом случае, необходимо будет соответствующим образом изменить переменные, описанные в разделе 5.1.3.

5.2.3 Установка и настройка KeyCloak без применения контейнерной виртуализации

Установка и настройка KeyCloak производится в заранее развернутую ОС Ubuntu 20.04.5 LTS.

Используемая версия KeyCloak – 19.0.3.

Для хранения данных KeyCloak интегрируется с СУБД PostgreSQL. Используемая версия PostgreSQL – 12.214.

5.2.3.1 Установка и настройка СУБД PostgreSQL для работы с KeyCloak

Действия выполняются в контексте учетной записи суперпользователя `root`.

Установить и проверить запуск СУБД PostgreSQL:

```
# apt -y install postgresql
# systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor pr>
Active: active (exited) since Thu 2022-10-13 20:10:46 UTC; 5min ago
Main PID: 3303 (code=exited, status=0/SUCCESS)
Tasks: 0 (limit: 2274)
Memory: 0B
CGroup: /system.slice/postgresql.service
```

Листинг 65: Пример установки СУБД PostgreSQL

Выполнить настройку СУБД:

```
# sudo -u postgres psql
postgres=# create user keycloak with password 'mysuperpasswd';
postgres=# create database keycloak owner keycloak;
postgres=# grant all privileges on database keycloak to keycloak;
postgres=# \q
```

Листинг 66: Пример настройки СУБД PostrgeSQL для KeyCloak

На этом установка СУБД PostgreSQL завершена[12].

5.2.3.2 Установка и настройка KeyCloak

Добавить пользователя и группу, от имени которых будет работать KeyCloak:

```
# groupadd -r keycloak
# useradd -m -d /var/lib/keycloak -s /sbin/nologin -r -g keycloak keycloak
```

Листинг 67: Пример создания пользователя и группы для KeyCloak

Создать каталог, распаковать и установить в него дистрибутив KeyCloak:

```
# mkdir -p /opt/keycloak
# apt -y install unzip
# cd /opt/
# unzip /opt/keycloak/keycloak-19.0.1.zip -d /opt/keycloak
# chown -R keycloak. keycloak
# chmod o+x /opt/keycloak/keycloak-19.0.1/bin/
```

Листинг 68: Пример установки KeyCloak

Установить и проверить версию JAVA:

```
# apt -y install openjdk-11-jdk
# java -version
openjdk version "11.0.16" 2022-07-19
OpenJDK Runtime Environment (build 11.0.16+8-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.16+8-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)
```

Листинг 69: Пример установки JAVA для KeyCloak

Создать сертификат для `keycloak` (или импортировать имеющийся):

[12] Описываемый вариант установку СУБД учитывает только создание схемы данных в интересах KeyCloak

```
# openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout /opt/keycloak/keycloak-
  19.0.1/conf/server.key.pem -out /opt/keycloak/keycloak-19.0.1/conf/server.crt.pem
```

Листинг 70: Пример создания сертификата для KeyCloak

Назначить права доступа на конфигурационный файл:

```
# The database vendor.
db=postgres

# The username of the database user.
db-username=keycloak

# The password of the database user.
db-password=mysuperpasswd

# The full database JDBC URL. If not provided, a default URL is set based on the selected database
  vendor.
db-url=jdbc:postgresql://localhost/keycloak

# Observability

# If the server should expose metrics and healthcheck endpoints.
metrics-enabled=true

# HTTP

# The file path to a server certificate or certificate chain in PEM format.
https-certificate-file=/opt/keycloak/keycloak-19.0.1/conf/server.crt.pem

# The file path to a private key in PEM format.
https-certificate-key-file=/opt/keycloak/keycloak-19.0.1/conf/server.key.pem


# Hostname for the KeyCloak server.
hostname=keycloak.mydomain.com:8443

#http-enabled=true

# Enable autit
log-console-output=default
log=console,file
log-file=/tmp/keycloak.log
```

Листинг 71: Файл `/opt/keycloak/keycloak-19.0.1/conf/keycloak.conf`. Пример конфигурации.

5.2.3.3 Запуск KeyCloak

Выполнить первичный запуск KeyCloak в отладочном режиме:

```
# cd /opt/keycloak/keycloak-19.0.1
# bin/kc.sh start-dev
# export KEYCLOAK_ADMIN=admin
# export KEYCLOAK_ADMIN_PASSWORD=passwd
# bin/kc.sh build
```

```
# bin/kc.sh start
```

Листинг 72: Пример первичного запуска KeyCloak в режиме отладки

Затем прервать процесс выполнения `keycloak` (`Ctrl+C`) и подготовить сценарий автоматического запуска `keycloak` для службы инициализации `systemd`:

```
[Unit]
Description=KeyCloak
After=network.target

[Service]
Type=idle
User=keycloak
Group=keycloak
SuccessExitStatus=0 143
ExecStart=!/opt/keycloak/keycloak-19.0.1/bin/kc.sh start --hostname=keycloak.mydomain.com
TimeoutStartSec=600
TimeoutStopSec=600

[Install]
WantedBy=multi-user.target
```

Листинг 73: Пример содержимого сценария `/etc/systemd/system/keycloak.service`

Инициализировать службу KeyCloak для автоматического запуска:

```
# systemctl daemon-reload
# systemctl start keycloak
# systemctl enable keycloak
```

Листинг 74: Пример инициализации службы `keycloak`

5.2.3.4 Настройка федерации провайдера KeyCloak и FreeIPA

При необходимости, можно настроить федерализацию аутентификационных отношений между провайдером аутентификации на базе Kerberos/LDAP (таким как FreeIPA или AD) и KeyCloak. Для этого необходимо уже иметь настроенный домен. Указанный ниже пример конфигурации приведен исходя из предположений, что используется федерация с FreeIPA.

Установить FreeIPA (https://www.freeipa.org/page/Documentation) можно, руководствуясь следующей документацией:

- для ОС Ubuntu 20.04 – https://www.howtoforge.com/how-to-add-ubuntu-system-to-freeipa-server/;
- для ОС Alt Linux – https://www.altlinux.org/FreeIPA;
- для ОС Astra Linux – https://wiki.astralinux.ru/display/doc/FreeIPA+Astra+Linux;
- для ОС CentOS – https://www.dmosk.ru/miniinstruktions.php?mini=freeipa-centos
- для ОС RedHat – https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/linux_domain_identity_authentication_and_policy_guide/installing-ipa

- для ОС RedOS – https://redos.red-soft.ru/base/server-configuring/installation-ipa/install-ipa/
- для ОС ROSA Linux – wiki.rosalab.ru

Сервер FreeIPA является универсальным сервером домена, обладающем поддержкой аутентификации с помощью Kerberos и схемой каталогов LDAP стандарта 389 Directory Server, являющейся реализацией стандарта RFC1777.

Для федерации необходимо в KeyCloak сконфигурировать новую область (Realm), отличную от Master. Для этого, нужно нажать кнопку `Add Realm` и задать параметры, например:

Тип записи	Значение
`Name`	kubernetes
`Display Name`	Kubernetes
`HTML Display Name`	<img src="https://kubernetes.io/images/nav_logo.svg" width="400" >

Таблица 4: Таблица данных Realm для федерации FreeIPA и KeyCloak.

Kubernetes по умолчанию проверяет подтвержден у пользователя адрес e-mail или нет. Так как используется собственный LDAP-сервер, то тут эта проверка всегда будет возвращать `false`. Нужно отключить представление этого параметра в Kubernetes, для этого выбрать:

`Client scopes` -> `Email` -> `Mappers` -> `Email verified (Delete)`.

Теперь можно настроить федерацию, для этого перейти в:

`User federation` -> `Add provider...` -> `ldap`

И заполнить данные, например:

Тип записи	Значение
`Console Display Name`	freeipa.example.org
`Vendor`	Red Hat Directory Server
`UUID LDAP attribute`	ipauniqueid
`Connection URL`	ldaps://freeipa.example.org
`Users DN`	cn=users,cn=accounts,dc=example,dc=org
`Bind DN`	uid=keycloak-svc,cn=users,cn=accounts,dc=example,dc=org
`Bind Credential`	<password>
`Allow Kerberos authentication:`	on
`Kerberos Realm:`	EXAMPLE.ORG
`Server Principal:`	HTTP/freeipa.example.org@EXAMPLE.ORG
`KeyTab:`	/etc/krb5.keytab

Таблица 5: Таблица данных для федерации FreeIPA и KeyCloak.

Пользователя `keycloak-svc` нужно создать заранее на LDAP-сервере.

В случае с Active Directory, достаточно просто выбрать `Vendor`: `Active Directory` и необходимые настройки подставятся в форму автоматически.

Нажать `Save`.

Для ассоциации пользовательских имен выполнить переход:

`User federation` -> `freeipa.example.org` -> `Mappers` -> `First Name`

и настроить маппинг пользователей указав значение:

Тип записи	Значение
Ldap attribute	givenName

Таблица 6: Таблица данных для федерации пользователей.

Аналогично, для групп:

User federation -> freeipa.example.org -> Mappers -> Create

Тип записи	Значение
Name	groups
Mapper type	group-ldap-mapper
LDAP Groups DN	cn=groups,cn=accounts,dc=example,dc=org
User Groups Retrieve Strategy	GET_GROUPS_FROM_USER_MEMBEROF_ATTRIBUTE

Таблица 7: Таблица данных для федерации групп.

На этом настройка федерации закончена, затем нужно настроить клиента. Клиент – это приложение, которое будет получать пользователей из KeyCloak. Для его настройки нужно перейти:

Clients -> Create

И указать значения:

Тип записи	Значение
Client ID	kubernetes
Access Type	confidential
Root URL	http://kubernetes.example.org/
Valid Redirect URIs	http://kubernetes.example.org/*
Admin URL	http://kubernetes.example.org/

Таблица 8: Таблица данных для клиентских запросов.

Для пространства групп перейти к:

Client Scopes -> Create и заполнить данные:

Тип записи	Значение
Template	No template
Name	groups
Full group path	false

Таблица 9: Таблица данных для пространства имен групп.

Client Scopes -> groups -> Mappers -> Create :

Тип записи	Значение
Name	groups
Mapper Type	Group membership
Token Claim Name	groups

Таблица 10: Таблица данных для соответствия групп.

Затем включить отображение групп между доменами:

Clients -> kubernetes -> Client Scopes -> Default Client Scopes, выбрать groups в Available Client Scopes и нажать Add selected.

Для настройки аутентификации клиента перейти в Clients -> kubernetes и выбрать Authorization Enabled – ON.

Затем нажать Save, на этом настройка клиентского приложения завершена.

Получить Secret для дальнейшей настройки можно на странице:

Clients -> kubernetes -> Credentials.

Для настройки k8s нужно скопировать CA-сертификат вашего OIDC-сервера в /etc/kubernetes/pki/oidc-ca.pem и добавить необходимые опции для kube-apiserver. Для этого нужно обновить /etc/kubernetes/manifests/kube-apiserver.yaml на всех мастер-серверах:

```
  ...
  spec:
  containers:
  - command:
  - kube-apiserver
  ...
  - --oidc-ca-file=/etc/kubernetes/pki/oidc-ca.pem
  - --oidc-client-id=kubernetes
  - --oidc-groups-claim=groups
  - --oidc-issuer-url=https://keycloak.example.org/auth/realms/kubernetes
  - --oidc-username-claim=email
  ...
```

Листинг 75: Пример файла конфигурации /etc/kubernetes/manifests/kube-apiserver.yaml

Затем обновить конфигурационный файл kubeadm-config в кластере, что бы не потерять эти настройки при обновлении:

```
kubectl edit -n kube-system configmaps kubeadm-config
```

Листинг 76: Пример обновления kubeadm-config

И внести туда следующие данные:

```
...
data:
  ClusterConfiguration: |
    apiServer:
      extraArgs:
        oidc-ca-file: /etc/kubernetes/pki/oidc-ca.pem
        oidc-client-id: kubernetes
        oidc-groups-claim: groups
        oidc-issuer-url: https://keycloak.example.org/auth/realms/kubernetes
        oidc-username-claim: email
```

На этом настройка Kubernetes завершена. Вы можете повторить данные действия во всех ваших Kubernetes-кластерах.

При настройке RBAC можно ссылаться как на имя пользователя (поле name в jwt-токене), так и на группу пользователей (поле groups в jwt-токене). Ниже приведен пример настройки прав для группы kubernetes-default-namespace-admins:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
```

```
metadata:
  name: default-admins
  namespace: default
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: kubernetes-default-namespace-admins
  namespace: default
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: default-admins
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: Group
  name: kubernetes-default-namespace-admins
```

Листинг 77: Пример содержимого файла `kubernetes-default-namespace-admins`

6 Проверка работоспособности после установки

Проверить работоспособность программного интерфейса приложения (API) можно, обратившись к нему по адресу URL. Для этого необходимо выполнить простой запрос[13] от имени любого пользователя, формата:

```
$ curl https://<адрес приложения>/api/health | grep -o '"state":"UP","status":"pass"'
$ curl https://<адрес приложения>/api/health | grep -o 'fail'
$ curl https://<адрес приложения>/api/health | grep -o 'down'
```

Листинг 78: Пример запроса проверки API на корректность с помощью `curl`

При выполнении таких запросов API выведет информацию о состоянии и статусе программного интерфейса, которое будет свидетельствовать о его готовности к обработке запросов. При этом, состояние (`state`) должно быть отмечено как UP (работает), а статус проверки (`status`) должен свидетельствовать о прохождении проверки (`pass`).

Также в выводе не должно быть строк со статусом `fail` и(или) состоянием `down`. Образец корректного вывода приведен на листинге ниже[14]:

```
$ curl https://processor-preprod.isource.ru/api/health | grep -o '"state":"UP","status":"pass"'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total   Spent    Left  Speed
```

[13] Требуется наличие программы `curl` и сетевого доступа к приложению.
[14] В образце проверка произведена по адресу размещения приложения https://processor-preprod.isource.ru/.

```
100  8461    0  8461    0     0  15496      0 --:--:-- --:--:-- --:--:-- 15468
"state":"UP","status":"pass"
"state":"UP","status":"pass"
"state":"UP","status":"pass"
"state":"UP","status":"pass"
"state":"UP","status":"pass"
"state":"UP","status":"pass"
"state":"UP","status":"pass"
"state":"UP","status":"pass"
"state":"UP","status":"pass"
"state":"UP","status":"pass"
$ curl https://processor-preprod.isource.ru/api/health | grep -o 'fail'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total   Spent    Left  Speed
100  8461    0  8461    0     0  91967      0 --:--:-- --:--:-- --:--:-- 91967
$
$ curl https://processor-preprod.isource.ru/api/health | grep -o 'down'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total   Spent    Left  Speed
100  8462    0  8462    0     0  82960      0 --:--:-- --:--:-- --:--:-- 82960
$
```

Листинг 79: Образец корректного вывода проверки API

Для осуществления еще одной проверки программного интерфейса – отредактировать (при необходимости – создать) файл `playbooks/apps/applications_healthchecks.yml`, приведя его к следующему виду:

```
- hosts: controller
  tasks:

    - name: Get status market
      uri:
        url: "http://{{ hostvars['processor']['ansible_host'] }}/api/health"
        body_format: json
        headers:
          Host: "{{ all_target_market_api }}"
        return_content: true
        timeout: 3000
      delay: 5
      register: market_health
      ignore_errors: true
      no_log: true

    - name: Forming message - market
      ignore_errors: true
      set_fact:
        market_out: |
          {{ market_health.json.type }} - {{ market_health.json.state }}
          ------------
          {% for i in market_health.json.checks  %}
          {{ i.type }}: {{ i.state }}
          {% endfor %}
          ----------

    - name: Output
      ignore_errors: true
      debug:
```

```
      msg: |
        {{ market_out }}

  - name: Get status planning
    ignore_errors: true
    uri:
      url: "http://{{ hostvars['kub-5']['ansible_host'] }}/api/health"
      body_format: json
      headers:
        Host: "{{ all_target_planning_host }}"
      return_content: true
      timeout: 3000
    delay: 5
    register: planning_health
    ignore_errors: true
    no_log: true

  - name: Forming message - planning
    ignore_errors: true
    set_fact:
      planning_out: |
        {{ planning_health.json.type }} - {{ planning_health.json.state }}
        ------------
        {% for i in planning_health.json.checks  %}
        {{ i.type }}: {{ i.state }}
        {% endfor %}
        ----------

  - name: Output
    ignore_errors: true
    debug:
      msg: |
        {{ planning_out }}

  - name: Get status docs
    uri:
      url: "http://{{ hostvars['kub-5']['ansible_host'] }}/api/health"
      body_format: json
      headers:
        Host: "{{ all_target_docs_host }}"
      return_content: true
      timeout: 3000
    delay: 5
    register: docs_health
    ignore_errors: true
    no_log: true

  - name: Forming message - docs
    ignore_errors: true
    set_fact:
      docs_out: |
        {{ docs_health.json.type }} - {{ docs_health.json.state }}
        ------------
        {% for i in docs_health.json.checks  %}
        {{ i.type }}: {{ i.state }}
        {% endfor %}
        ----------

  - name: Output
```

```
    ignore_errors: true
    debug:
      msg: |
        {{ docs_out }}

  - name: Get status inspector
    uri:
      url: "http://{{ hostvars['kub-5']['ansible_host'] }}/api/health/details?key=healthkey"
      body_format: json
      headers:
        Host: "{{ all_target_inspector_api }}"
      return_content: true
      timeout: 3000
    delay: 5
    register: inspector_health
    ignore_errors: true
    no_log: true

  - name: Forming message - inspector
    ignore_errors: true
    set_fact:
      inspector_out: |
        {{ docs_health.json.type }} - {{ inspector_health.json.state }}
        ------------
        {% for i in inspector_health.json.checks  %}
        {{ i.type }}: {{ i.state }}
        {% endfor %}
        ----------

  - name: Output
    ignore_errors: true
    debug:
      msg: |
        {{ inspector_out }}
```

Листинг 80: Пример содержимого файла `applications_healthchecks.yml`

При осуществлении проверки происходит выполнение следующих операций:

- выполнение запроса к системе (с помощью вызова функции `Get status`);
- получение ответа (с помощью вызова функции `Output`);
- формализованное форматирование вывода (с помощью вызова функции `Forming message`)

Запуск теста производится командой:

```
ansible-playbook main.yml -i inventories/koa/inventory.yml -t check
```

Листинг 81: Пример запуска теста для проверки программного интерфейса

Пример корректного возврата значений при проверке программного интерфейса приведен на рисунке Рисунок 23:

```
TASK [Output] ******************************************************************
ok: [controller] =>
  msg: |-
    Сервис цифрового продукта - UP
    ------------
    БД Postgres: UP
    Кэш сервиса цифрового продукта 'Процессор' в Redis: UP
    Очередь брокера сообщений RabbitMQ: UP
    Очередь брокера сообщений RabbitMQ: UP
    Очередь брокера сообщений RabbitMQ: UP
    Очередь брокера сообщений RabbitMQ: UP
    Очередь брокера сообщений RabbitMQ: UP
    Очередь брокера сообщений RabbitMQ: UP
    Очередь брокера сообщений RabbitMQ: UP
    Очередь брокера сообщений RabbitMQ: UP
    ----------
```

Рисунок 23: Образец корректного результата проверки программного интерфейса

Для полной проверки работоспособности требуется выполнить вход в интерфейс управления KeyCloak, и убедиться в том, что вход возможен, и пользователи созданы:

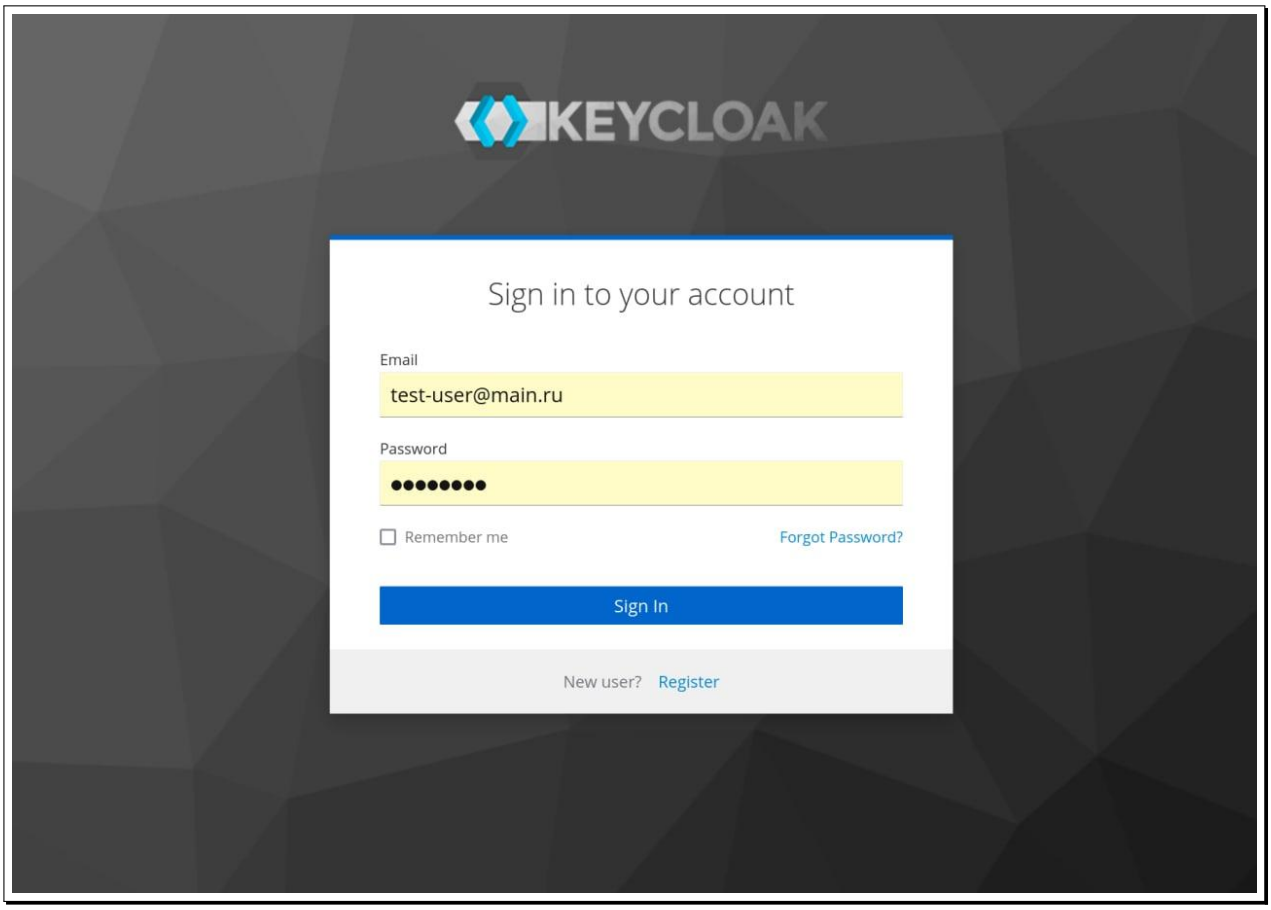


Рисунок 24: Пример входа в KeyCloak

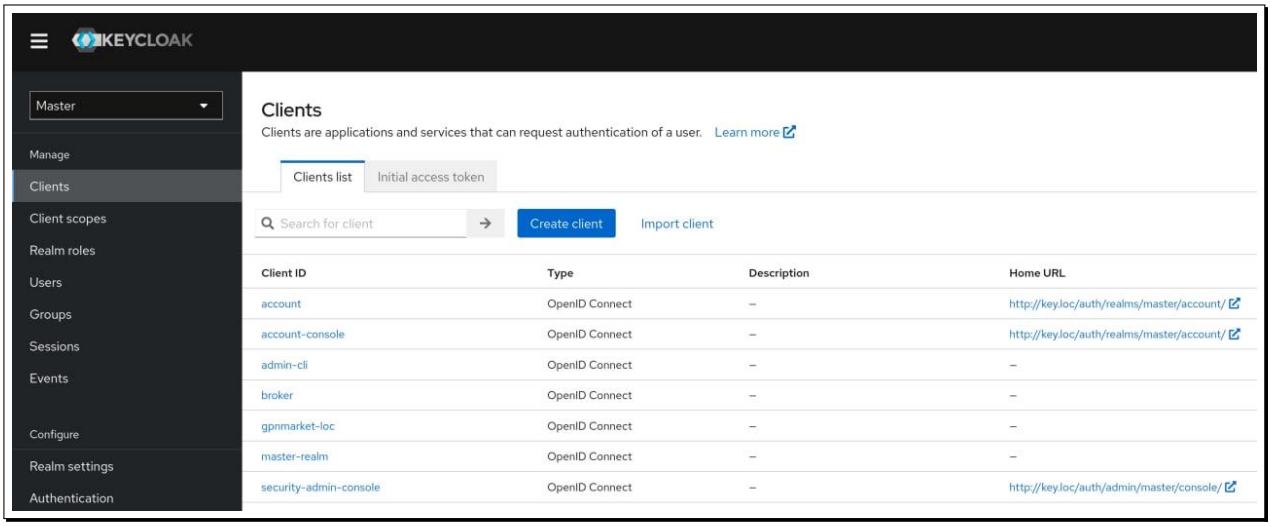


Рисунок 25: Пример основного окна администратораkeycloak

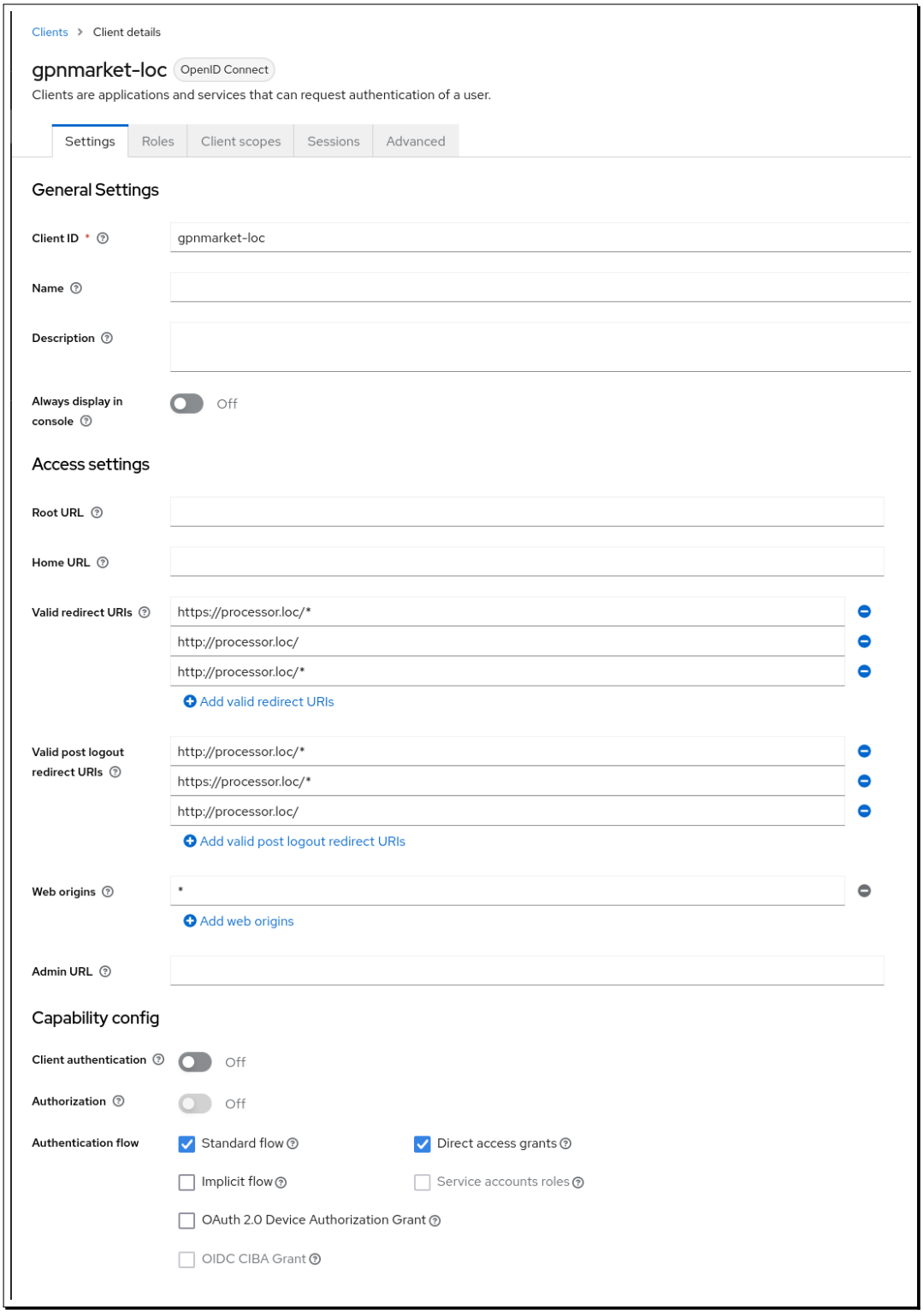


Рисунок 26: Пример метаданных клиента

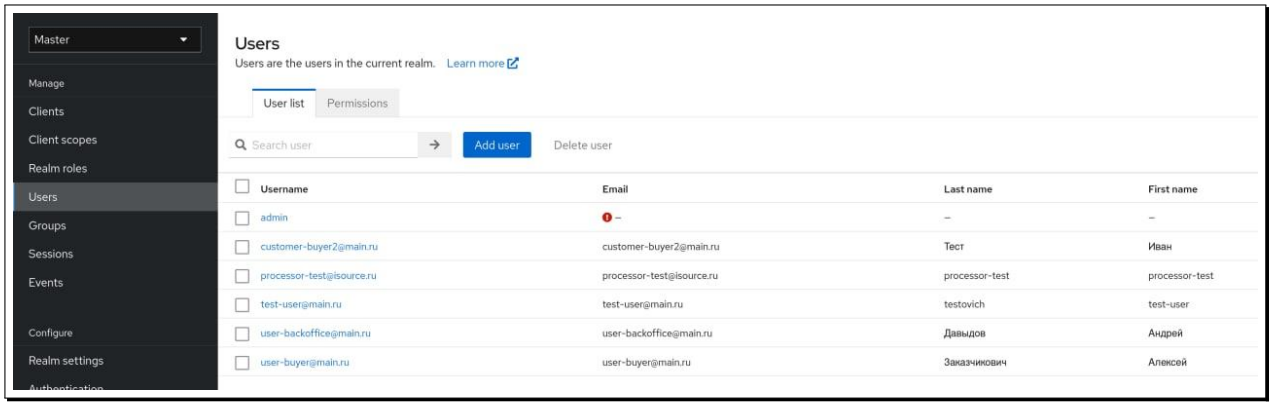


Рисунок 27: Пример созданных пользователей в KeyCloak

Затем для проверки выполнить вход в интерфейс модуля закупок, и убедится в том, что перенаправление аутентификации работает. Для этого нажать на кнопку «Контрагенты» и убедиться в том, что окно аналогично тому, которое изображено на рисунке Рисунок 24:

Рисунок 28: Пример входа в модуль закупок

Затем убедиться в том, что метаданные субъекта можно изменять:

Рисунок 29: Пример списка контрагентов

Рисунок 30: Пример редактирования данных контрагента

7 Известные ошибки установки и порядок их устранения

В случае выявления тех или иных недостатков, а также при необходимости взаимодействия с разработчиком (поставщиком) для сообщений об уязвимостях, неточностях, подаче рекламаций и т.п. – требуется руководствоваться регламентом, изложенным в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение Б. Определение жизненного цикла*» *is000-ALC_LCD.1* в разделе «*Процедуры обратной связи при выявлении недостатков (уязвимостей)*».

7.1 Ошибка установки пакетов `grub-efi-amd64-signed` и `shim-signed`

В некоторых случаях возникает ошибка, изображенная на рисунке Рисунок 31:

```
root@spb99tp8394-15:~# apt install htop
Reading package lists... Done
Building dependency tree
Reading state information... Done
htop is already the newest version (2.2.0-2build1).
htop set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 122 not upgraded.
1 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Setting up grub-efi-amd64-signed (1.167.2+2.04-1ubuntu44.2) ...
mount: /var/lib/grub/esp: special device /dev/disk/by-id/scsi-SVMware_Virtual_disk_6000c296fb5b941aaeb01ff46526a20c-part1 does not exist.
dpkg: error processing package grub-efi-amd64-signed (--configure):
 installed grub-efi-amd64-signed package post-installation script subprocess returned error exit status 32
dpkg: dependency problems prevent processing triggers for shim-signed:
 shim-signed depends on grub-efi-amd64-signed | grub-efi-arm64-signed; however:
  Package grub-efi-amd64-signed is not configured yet.
  Package grub-efi-arm64-signed is not installed.

dpkg: error processing package shim-signed (--configure):
 dependency problems - leaving triggers unprocessed
Errors were encountered while processing:
 grub-efi-amd64-signed
 shim-signed
E: Sub-process /usr/bin/dpkg returned an error code (1)
root@spb99tp8394-15:~#
```

Рисунок 31: Пример ошибки при установке пакетов `grub-efi-amd64-signed` и `shim-signed`

7.1.1 Причина появления ошибки

Причина появления ошибки заключается в том, что с высокой вероятностью всего была осуществлена неверная разметка дискового пространства, состоящая в том, что раздел для установки загрузчика и его подписи не был снабжен нужным флагом ESP (EFI System Partition).

7.1.2 Устранение ошибки

7.1.2.1 Вариант №1. Устранение ошибки

При осуществлении разметки дисков, или после разметки и установки ОС, но до установки программного комплекса, необходимо установить метку ESP на раздел, отведенный под размещение загрузчика (обычно для загрузчика используется раздел `/boot/efi`, но необязательно).

Для этого, используя содержащийся в составе ОС Ubuntu 20.04 менеджер дисков – программу `Gparted`, необходимо установить соответствующий флаг загрузки.

В том случае, если программа `Gparted` не установлена, то её можно установить с помощью команды (потребуются полномочия `root`):

```
# apt -y install gparted
```

Листинг 82: Пример установки программы `Gparted`

Затем необходимо запустить программу `Gparted`, осуществить выбор нужного раздела, и через выбор в верхнем меню «`Раздел`», затем «`Управление флагами`», и выполнить операцию назначения флага ESP (потребуются полномочия `root`).

Пример совершения операции по установке флага ESP изображен на рисунке Рисунок 32:

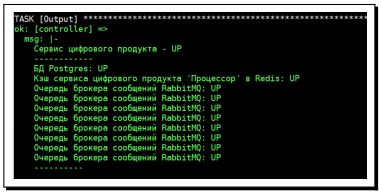


Рисунок 32: Пример установки флага ESP на раздел `/boot/efi`

В том случае, если графический интерфейс для запуска программы `Gparted` использовать не представляется возможным, то можно выполнить аналогичную операцию, используя интерфейс командной строки, и вызвать эту программу с соответствующими аргументами используя терминал с интерфейсом оболочки (потребуются полномочия `root`):

```
# parted set <раздел> flag esp
```

Листинг 83: Пример установки флага ESP используя интерфейс командной строки

Сведения о функциональных возможностях программы `Gparted` и использовании её интерфейса приведены на страницах электронной справки `parted(8)` и `gparted(8)`.

Руководство пользователя программы `Gparted` (на русском языке) приведено на странице: https://gparted.org/display-doc.php?name=help-manual&lang=ru.

7.1.2.2 Вариант №2. Нейтрализация ошибки

Для нейтрализации ошибки можно подготовить специализированный сценарий (playbook) и поместить его в файл `playbooks/other/efi-fix.yml`:

```
---
- hosts: all:!controller
  become: true
  tasks:
    - name: Test apt update
      ansible.builtin.apt:
        name: jq
        update_cache: true
      register: test_apt_update
      ignore_errors: true

    - name: dist upgrade
      command: apt dist-upgrade -y
      when: test_apt_update is failed
      ignore_errors: true

    - name: Ensure problem efi file extist
      ansible.builtin.stat:
        path: /var/lib/dpkg/info/grub-efi-amd64-signed.postinst
      register: stat_result_efi

    - name: fix efi
      block:
        - name: Ensure problem efi mark file extist
          ansible.builtin.stat:
            path: /var/lib/dpkg/info/grub-efi-amd64-signed.postinst.complete
          register: stat_result_efi_mark

        - name: Replace files efi
          ansible.builtin.lineinfile:
            path: "/var/lib/dpkg/info/grub-efi-amd64-signed.postinst"
            regexp: '^#! /bin/sh'
            line: "#! /bin/sh\nexit 0"
          when: not stat_result_efi_mark.stat.exists

        - name: Create complete mark efi
          ansible.builtin.file:
            path: "/var/lib/dpkg/info/grub-efi-amd64-signed.postinst.complete"
            state: touch
          when: not stat_result_efi_mark.stat.exists
      when: stat_result_efi.stat.exists and test_apt_update is failed



    - name: Ensure problem shim file extist
      ansible.builtin.stat:
        path: /var/lib/dpkg/info/shim-signed.postinst
      register: stat_result_shim

    - name: shim
      block:
        - name: Ensure problem mark file extist
          ansible.builtin.stat:
            path: /var/lib/dpkg/info/shim-signed.postinst.complete
          register: stat_result_shim_mark
```

```
      - name: Replace files shim
        ansible.builtin.lineinfile:
          path: "/var/lib/dpkg/info/shim-signed.postinst"
          regexp: '^#! /bin/sh'
          line: "#! /bin/sh\nexit 0"
        when: not stat_result_shim_mark.stat.exists

      - name: Create complete mark shim
        ansible.builtin.file:
          path: "/var/lib/dpkg/info/shim-signed.postinst.complete"
          state: touch
        when: not stat_result_shim_mark.stat.exists
    when: stat_result_shim.stat.exists and test_apt_update is failed

  - name: Test apt update
    shell:
      cmd: "apt install -f"

  - name: Reboot a slow machine
    ansible.builtin.reboot:
      reboot_timeout: 3600
```

Листинг 84: Пример сценария `playbooks/other/efi-fix.yml`

А затем активизировать сценарий командой:

```
ansible-playbook playbooks/other/efi-fix.yml -i inventories/koa/inventory.yml
```

Листинг 85: Пример активизации сценария `playbooks/other/efi-fix.yml`

8 Настройки, связанные с безопасностью

Для реализации мер защиты, которые связаны со средой выполнения (настройки безопасности операционной системы Ubuntu 20.04 LTS), необходимо следовать указаниям, приведенным в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение А. Безопасность в ОС Ubuntu.*» *is000-AGD_PRE.1*.

8.1 Настройки идентификации и аутентификации

В настоящем разделе приведены сведения, которые описывают возможные конфигурации, используемые для усиления политик аутентификации пользователей и администраторов программного комплекса. Дополнительно рекомендуется использовать сведения, приведенные в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение А. Безопасность в ОС Ubuntu.*» *is000-AGD_PRE.1*.

8.1.1 Настройки идентификации и аутентификации программного комплекса

Для установления (изменения) политики паролей с учетом

- возможности изменения длины пароля (в символах);
- возможности изменения алфавита пароля со значением не менее, чем 70 знаков, а именно наличие в пароле: цифр, букв латинского алфавита (в верхнем и нижнем регистрах), спецсимволов (за исключением UTF 128+);
- возможности хранить заданную историю пароля (не менее, чем в 1 предыдущий пароль);
- возможности установления срока действия пароля.

требуется произвести вход в IDM KeyCloak от имени администратора и используя в меню «Аутентификация» вкладку «Политики пароля» указать требуемые значения политики, после чего нажать кнопку «Сохранить». Пример совершения операции приведен на рисунке Рисунок 33:

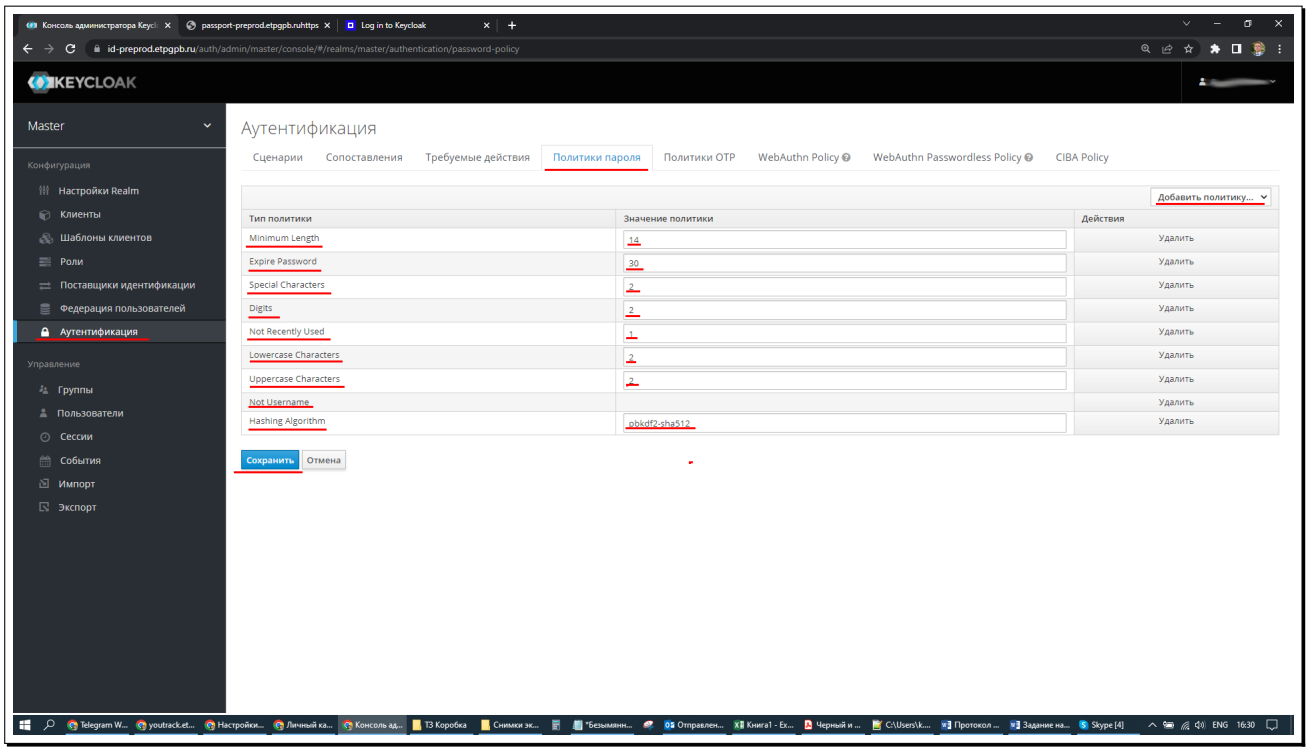


Рисунок 33: Пример изменения парольной политики

Где:

- поле типа политики «Minimum Lenght» – задает значение политики длины пароля;
- поле типа политики «Expire Password» – задает значение политики максимального срока действия пароля;
- поле типа политики «Special Characters» – задает значение политики обязательного наличия в пароле указанного минимального (в примере – 2 символа) количества специальных символов;
- поле типа политики «Digits» – задает значение политики обязательного наличия в пароле указанного минимального (в примере – 2 символа) количества цифр;

- поле типа политики «Not Recently Used» – задает значение (число) политики истории хранения предыдущих паролей (в примере задается обязательство хранить один предыдущий пароль);
- поле типа политики «Lowercase Characters» – задает значение политики обязательного наличия в пароле указанного минимального (в примере – 2 символа) количества строчных символов;
- поле типа политики «Uppercase Characters» – задает значение политики обязательного наличия в пароле указанного минимального (в примере – 2 символа) количества прописных символов;
- поле типа политики «Not Username» – задает запрет создания пароля, совпадающего с именем пользователя[15];
- поле типа политики «Hashing Algorithm» – задает значение алгоритма одно-обратимой хэш-функции, которая будет использована при хранении аутентификационной информации (изменять данную политику со значения по умолчанию не требуется и не рекомендуется).

Для удаления или добавления политик можно применять кнопки «Удалить» и «Добавить политику» соответственно.

После задания новой политики, необходимо нажать кнопку «Сохранить».

8.1.2 Настройка ограничительных политик

IDM KeyCloak поддерживает широкий набор политик по реализации ограничений сеанса, попыток входа, и т.п.

Для настройки значений, ограничивающих количество попыток ввода пароля, а также действий, которые будут осуществляться в ответ на превышение ограничений, необходимо использовать возможности, предоставляемые в меню «Realm Settings» – вкладка «Security Defence» – меню «Brute Force Detection», определяя разную политику для каждого обслуживаемого реалма. Пример совершения операций по ограничению политики количества попыток ввода пароля и варианты ответных действий провайдера IDM, представлены на рисунках Рисунок 34– Рисунок35:

[15]Политика не носит обязательного характера, исходя из требований ИБ.

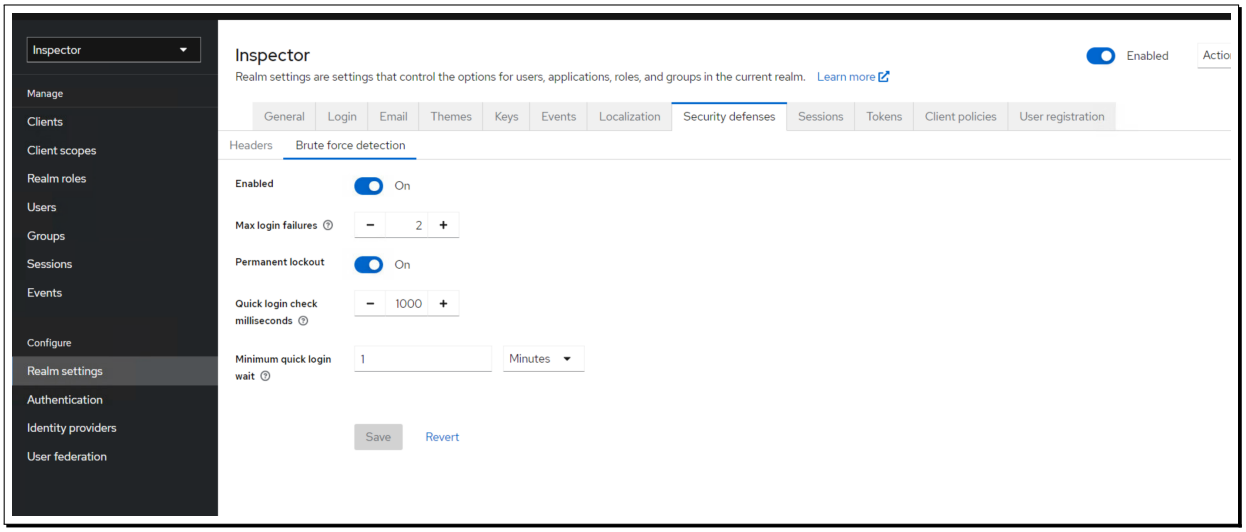


Рисунок 34: Пример настройки политики постоянной блокировки пользователя в ответ на превышение количества попыток ввода пароля

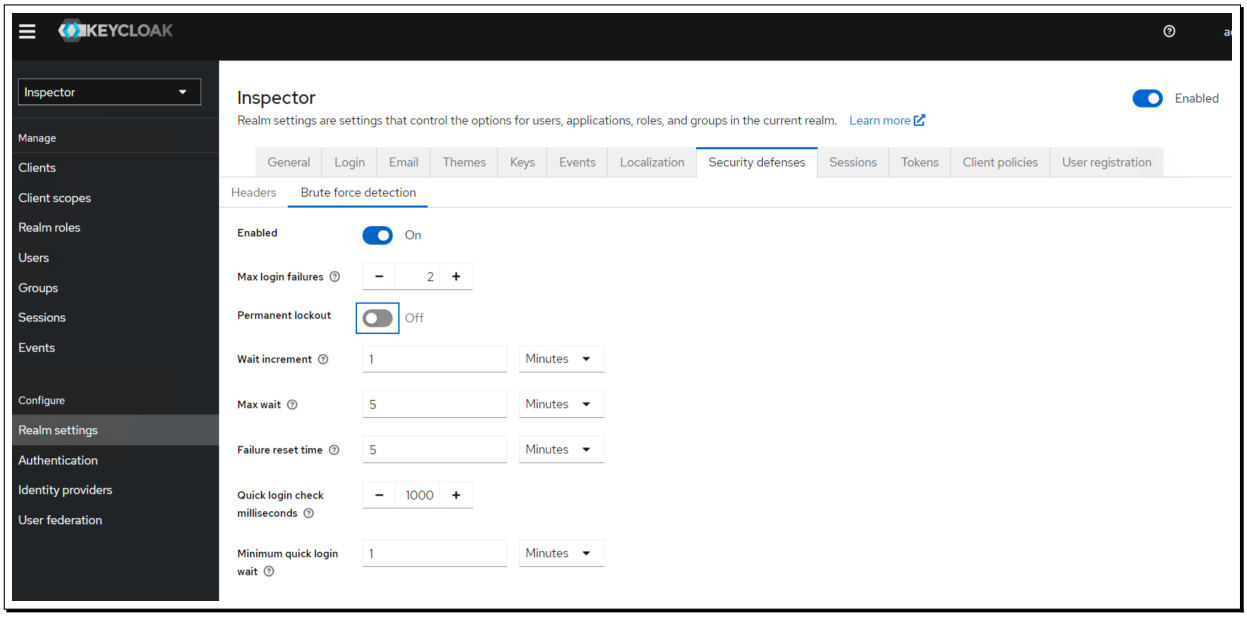


Рисунок 35: Пример настройки политики временной блокировки пользователя в ответ на превышение количества попыток ввода пароля

Для примера, если установить политику постоянной блокировки так, как указано в примере на рисунке Рисунок 34, пользователь, превысивший ограничение, будет заблокирован до тех пор, пока администратор не примет решение о его разблокировании. Пример окна в административном интерфейсе IDM, отображающего блокировку пользователя, приведен на рисунке Рисунок 36:

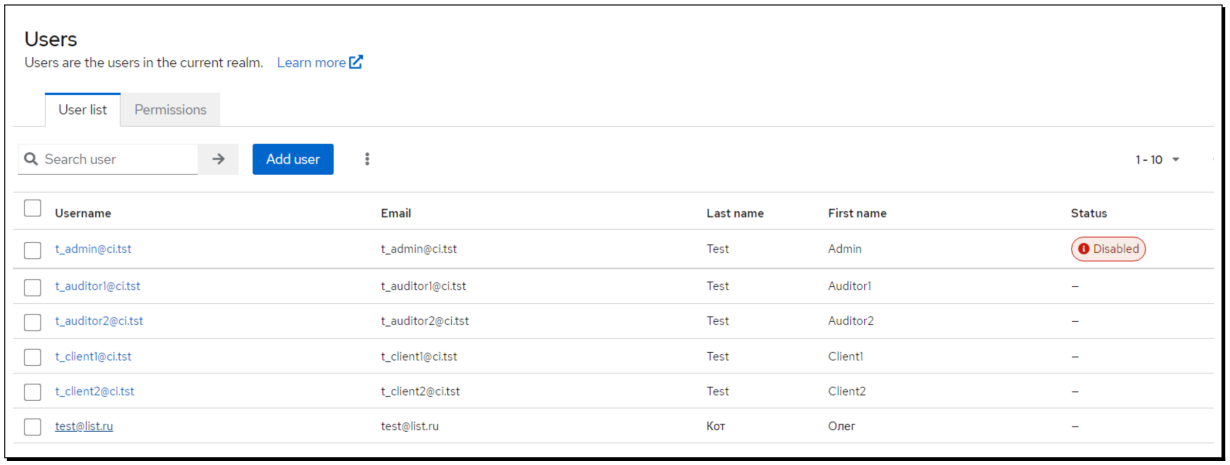


Рисунок 36: Пример окна, отражающего постоянную блокировку пользователя

В том случае, если политика определяет временную блокировку, пользователь будет лишен возможности совершать вход в систему в течение заданного времени.

Для настройки политики, определяющей тайм-аут истечения сессии пользователя и действия, которые необходимо предпринять по истечению тайм-аута, необходимо использовать меню «`Sessions`», также задавая желаемые политики отдельно для каждого реалма. Пример окна с параметрами приведен на рисунке 37:

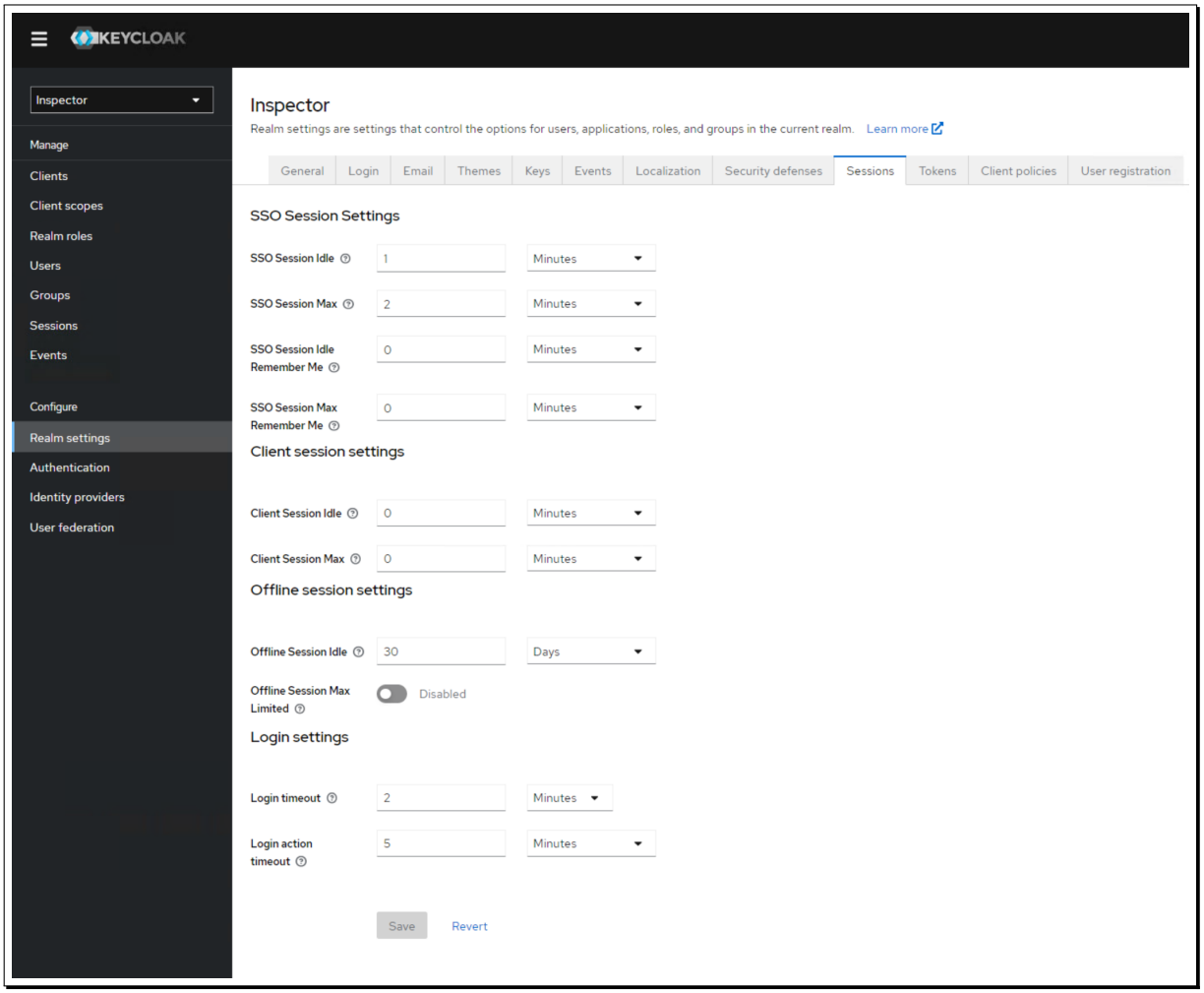


Рисунок 37: Пример окна политик, ограничивающих пользовательский сеанс

Описание наиболее важных политик приведено в таблице Таблица 11:

№ п/п	Значение по умолчанию	Параметр политики	Описание параметра политики
1	30 попыток	`Max Login Failures`	Максимальное количество неудачных попыток входа.
2	1000	`Quick Login Check Milliseconds`	Минимальное время между попытками входа (в миллисекундах)
3	1 минута	`Minimum Quick Login Wait`	Минимальное время периода ожидания аутентификации, в случае, если попытки входа в систему выполняются быстрее, чем определено параметром `Quick Login Check Milliseconds`
4	1 минута	`Wait Increment`	Время, добавляемое к времени, в случае если пользователь временно отключен, когда количество попыток входа пользователя превышает значение, определенное параметром `Max Login Failures`
5	15 минут	`Max Wait`	Время, на которое пользователю будет заблокирована возможность совершать вход в систему
6	12 часов	`Failure Reset Time`	Время сброса счетчика неудачных попыток входа. Таймер сброса счетчика запускается с момента последнего неудачного входа в систему
7	Нет	`SSO Session Idle`	Этот параметр предназначен только для клиентов OIDC. Если пользователь неактивен дольше этого тайм-аута, сеанс пользователя становится недействительным. Это значение времени ожидания сбрасывается, когда клиенты запрашивают повторную проверку подлинности или отправляют запрос маркера обновления

№ п/п	Значение по умолчанию	Параметр политики	Описание параметра политики
8	Нет	SSO Session Max	Максимальное время до истечения сеанса пользователя
9	Нет	Client Session Idle	Если пользователь неактивен дольше этого тайм-аута, запросы маркера обновления увеличивают тайм-аут простоя. Этот параметр задает более короткий тайм-аут простоя маркеров обновления, чем тайм-аут простоя (SSO Session Idle) сеанса, но пользователи могут переопределить его для отдельных клиентов. Этот параметр является дополнительной конфигурацией и, если он равен нулю, использует тот же тайм-аут простоя, что и в конфигурации бездействия сеанса единого входа
10	Нет	Client Session Max	Максимальное время до истечения срока действия токена обновления и его аннулирования. Этот параметр указывает более короткое время ожидания маркеров обновления, чем время ожидания сеанса, но пользователи могут переопределить его для отдельных клиентов. Этот параметр является дополнительной конфигурацией и, если он равен нулю, использует тот же тайм-аут простоя, заданный параметром SSO Session Idle
11	Нет	Login timeout	Общее время, необходимое для входа в систему. Если аутентификация занимает больше времени, чем это время, пользователь должен снова запустить процесс аутентификации.
12	Нет	Login action timeout	Максимальное время, которое пользователи могут проводить на любой странице в процессе аутентификации

Таблица 11: Описание наиболее важных ограничительных политик.

Дополнительные сведения и описание всех возможных ограничений, приведены на страницах справки:

https://www.keycloak.org/docs/latest/server_admin/#managing-user-sessions

и

https://www.keycloak.org/docs/latest/server_admin/#mitigating-security-threats

8.1.3 Настройка строгой двухфакторной аутентификации для учетных записей ОС

Для любой (в т.ч. технологической административной) учетной записи пользователя в операционной системе доступна возможность использования строгой двухфакторной аутентификации с применением токена Yubikey[16] и протокола U2F (https://www.yubico.com/).

Либо, вместо аппаратного токена Yubikey, может быть применен любой токен, с реализацией протокола аутентификации U2F. К приобретению доступен, например, токен Rutoken U2F, производимый отечественной компанией «Актив» (https://www.rutoken.ru/products/all/rutoken-u2f/).

На сервере должны быть установлены следующие пакеты:

```
 #id
uid=0(root) gid=0(root) группы=0(root)
 #dpkg -s libpam-u2f | grep installed
Status: install ok installed
 #dpkg -s libu2f-udev | grep installed
Status: install ok installed
 #dpkg -s pamu2fcfg | grep installed
```

[16] Понадобится физический доступ к серверу.

```
Status: install ok installed
 #dpkg -s yubikey-personalization-gui | grep installed
Status: install ok installed
 #dpkg -s yubikey-manager | grep installed
Status: install ok installed
```

Листинг 86: Пример проверки наличия в системе пакетов поддержки протокола U2F и токена Ybikey.

Иначе, установить указанные пакеты:

```
 #id
uid=0(root) gid=0(root) группы=0(root)
# apt install yubikey-manager yubikey-personalization-gui pamu2fcfg libpam-u2f libu2f-udev
```

Листинг 87: Пример установки в систему пакетов поддержки протокола U2F и токена Ybikey.

Выполнить настройку[17] PAM U2F (токен Yubikey должен быть очищен от предыдущей конфигурации, если таковая была):

```
$ id
uid=1000(user) gid=1000(user) группы=1000(user),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(
  lpadmin),132(lxd),133(sambashare)
$ pamu2fcfg > ~/.config/Yubico/u2f_keys
```

Листинг 88: Пример настройки протокола U2F и токена Ybikey для пользователя с административными полномочиями.

Затем скопировать ключи в каталог `/etc/Yubico/u2f_keys`:

```
$ id
uid=1000(user) gid=1000(user) группы=1000(user),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(
  lpadmin),132(lxd),133(sambashare)
$ sudo mv  ~/.config/Yubico/u2f_keys /etc/Yubico/u2f_keys
```

Листинг 89: Пример настройки протокола U2F и токена Ybikey для пользователя с административными полномочиями. Продолжение.

Для настройки U2F при использовании `sudo` выполнить конфигурацию (от имени `root`) файла `/etc/pam.d/sudo`, и внести туда после строки `@include common-auth` следующее содержимое:

```
@include common-auth
auth       required   pam_u2f.so authfile=/etc/Yubico/u2f_keys cue [cue_prompt=Требуется
  подтвердить присутствие ... ]
```

Листинг 90: Пример настройки двухфакторной аутенификации при использовании `sudo`. Пример содержимого файла `/etc/pam.d/sudo`

Выполнить аналогичную настройку для остальных точек входа. Для стандартного входа в терминал с применением двухфакторной аутентификации – отредактировать файл `/etc/pam.d/login`:

```
@include common-auth
```

[17] Настройка выполняется при установке (пробросе) в USB разъем сервера токена Yubikey.

```
auth       required   pam_u2f.so authfile=/etc/Yubico/u2f_keys cue [cue_prompt=Требуется
  подтвердить присутствие ... ]
```

Листинг 91: Пример настройки двухфакторной аутенификации при использовании входа в терминал TTY. Пример содержимого файла `/etc/pam.d/login`

Для входа в графическую сессию (GDM) с применением двухфакторной аутентификации – отредактировать файл `/etc/pam.d/gdm-password`:

```
@include common-auth
auth       required   pam_u2f.so authfile=/etc/Yubico/u2f_keys cue [cue_prompt=Требуется
  подтвердить присутствие ... ]
```

Листинг 92: Пример настройки двухфакторной аутенификации при использовании GDM. Пример содержимого файла `/etc/pam.d/gdm-password`

Для входа в сессию `ssh` с применением двухфакторной аутентификации – отредактировать файл `/etc/pam.d/sshd`:

```
@include common-auth
auth       required   pam_u2f.so authfile=/etc/Yubico/u2f_keys cue [cue_prompt=Требуется
  подтвердить присутствие ... ]
```

Листинг 93: Пример настройки двухфакторной аутенификации при использовании `ssh`. Пример содержимого файла `/etc/pam.d/sshd`

8.1.4 Настройка двухшаговой проверки с использованием протокола TOTP

Для пользователей программного комплекса также поддерживается возможность настройки аутентификации с помощью одноразового пароля, вводимого дополнительно после основного пароля. Его использование поддерживается конфигурацией централизованного провайдера идентификации и аутентификации KeyCloak и базируется на применении протокола аутентификации TOTP.

Для настройки KeyCloak и активизации TOTP, необходимо от имени администратора KeyCloak выполнить следующие операции, пример которых приведен на рисунках Рисунок 38 – Рисунок 42:

В разделе основного меню «`Authentication`», используя вкладку «`Required Actions`» отметить чек-бокс «`Configure OTP`». Пример операции приведен на рисунке Рисунок 38:

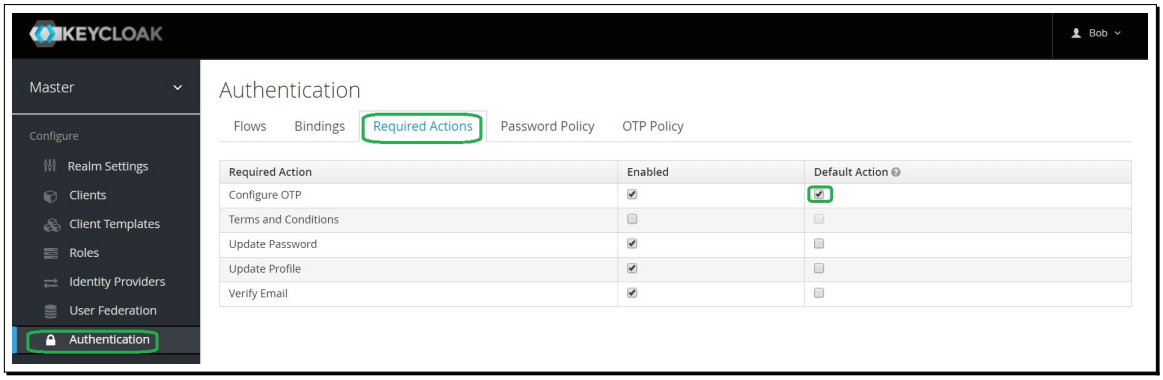


Рисунок 38: Пример настройки TOTP

Затем во вкладке «OTP Policy» задать необходимые политики одноразового пароля. Например, алгоритм одно-обратимой хеш-функции, используемой для исчисления аутентификационной информации, период времени жизни пароля, количество знаков, и окно сдвига применения пароля. Пример совершения операции приведен на рисунке Рисунок 39:

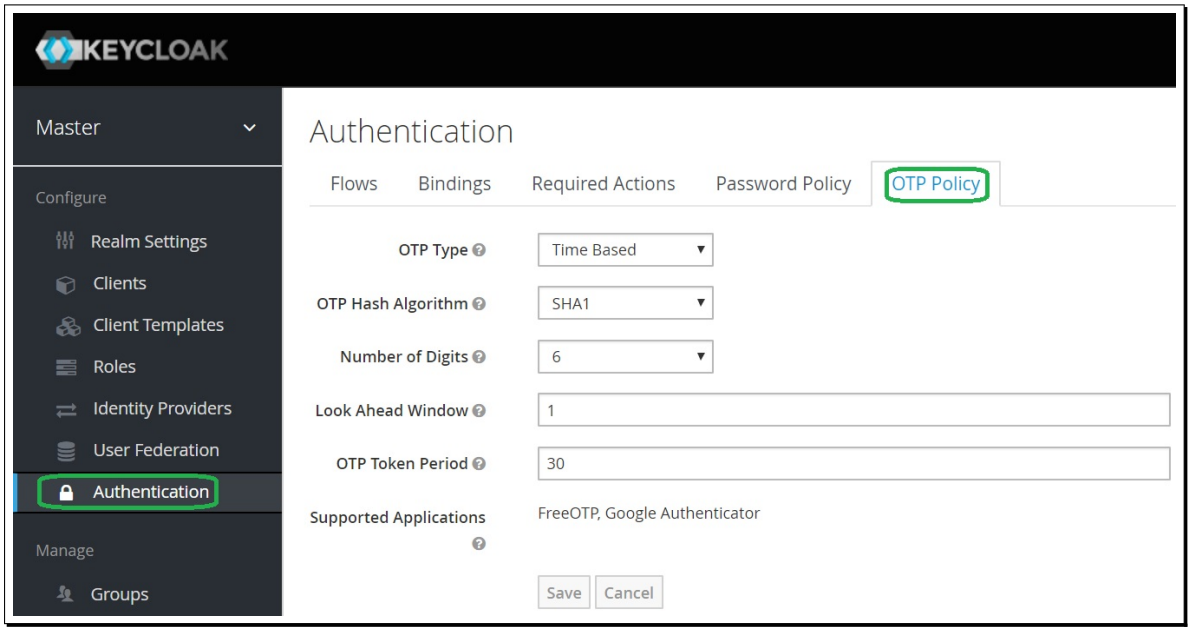


Рисунок 39: Пример настройки TOTP: (алгоритм SHA1, срок жизни одноразового пароля – 30 секунд, длина 6 символов, сдвиг – 1 пароль

Для ассоциации TOTP с учетной записью конкретного пользователя, в меню «Users» можно произвести настройки для тех пользователей, которым будет обязательно использовать двухшаговую проверку. Во вкладке «Details» в разделе «Required User Actions» у конкретного пользователя из выпадающего меню можно выбрать «Configure OTP». Пример совершения операции приведен на рисунке Рисунок 40:

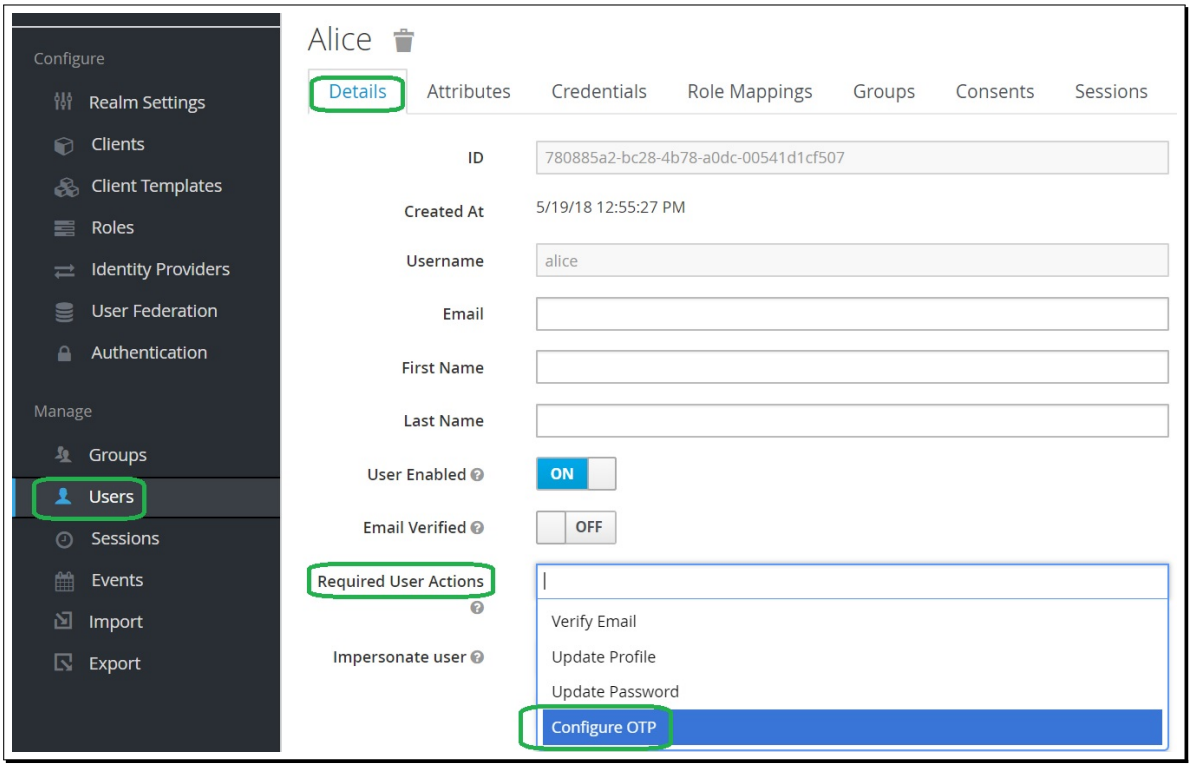


Рисунок 40: Пример настройки TOTP для заданного пользователя

Для ассоциации TOTP с программным компонентом комплекса (например, с модулем закупок), там же можно установить необходимость использования TOTP для всех пользователей службы или компонента. Для использования ключа, на экран будет выведен QR-код, содержащий необходимую первичную аутентификационную информацию. Далее код нужно будет отсканировать с помощью установленного на мобильный телефон (смартфон) с ОС Android™или Apple iOS™приложения FreeOTP, Google Authenticator или аналогичного (поддерживающего протокол TOTP).

Пример совершения операций приведен на рисунке Рисунок 41:

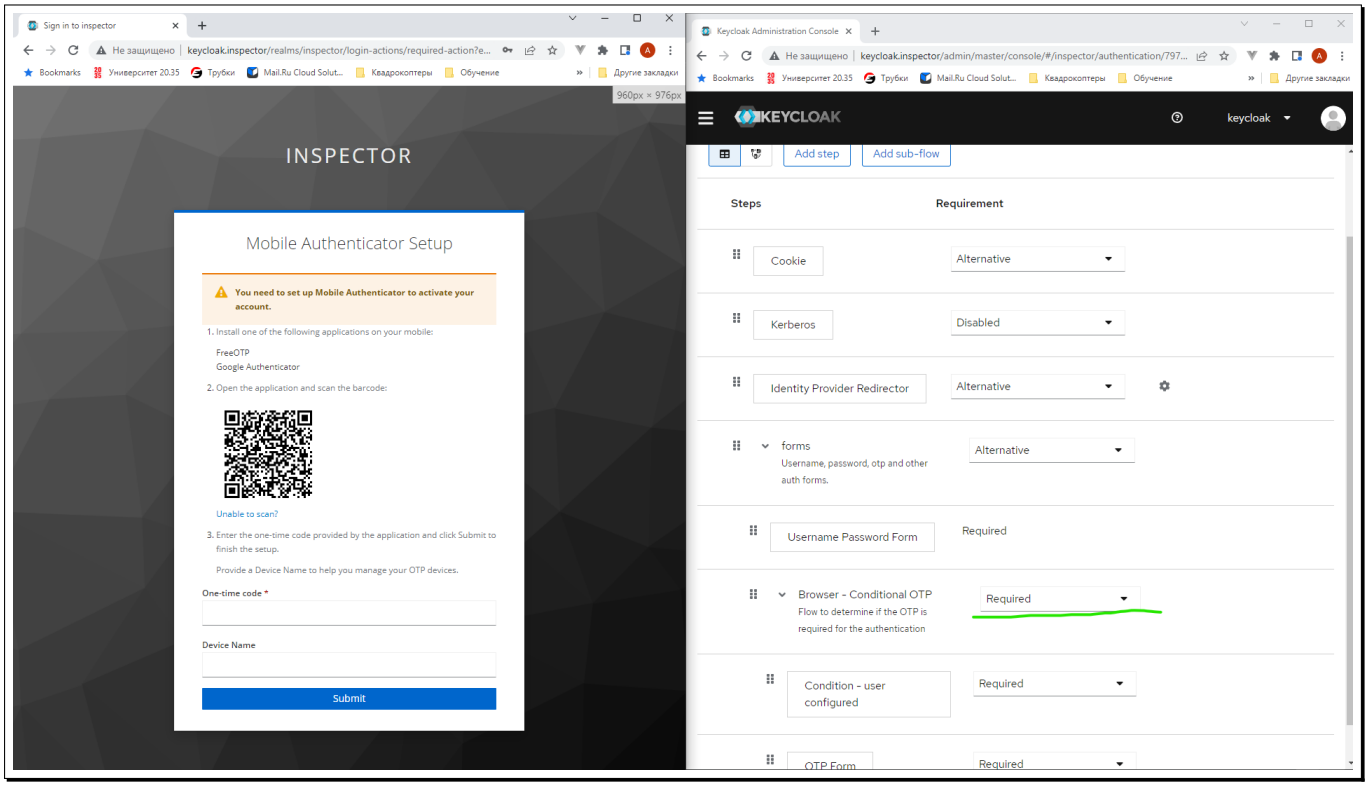


Рисунок 41: Пример настройки TOTP для пользователей модуля инспектора

После этого, от пользователя будет требоваться ввод временного одноразового пароля. Пример запроса пароля TOTP приведен на рисунке Рисунок 42:

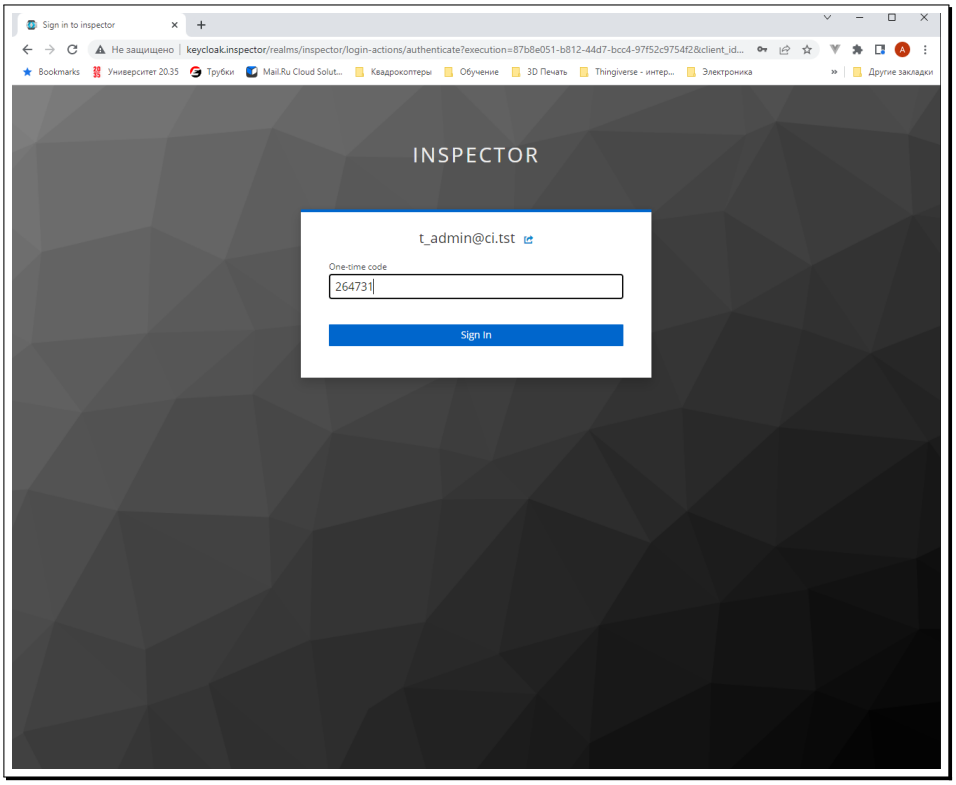


Рисунок 42: Пример входа пользователя в модуль инспектора с применением TOTP

8.1.5 Настройка межсервисного взаимодействия

Если сервис требует межсервисного взаимодействия (`grant_type=client_credentials`), например во всех асинхронных операциях (cron, mq) пользователь отсутствует, то необходимо использовать дополнительного клиента.

Пример создания такого клиента приведен ниже. Операции по созданию нового клиента, производятся аналогично тому, как приведено в примерах на рисунках Рисунок 43– Рисунок 47:

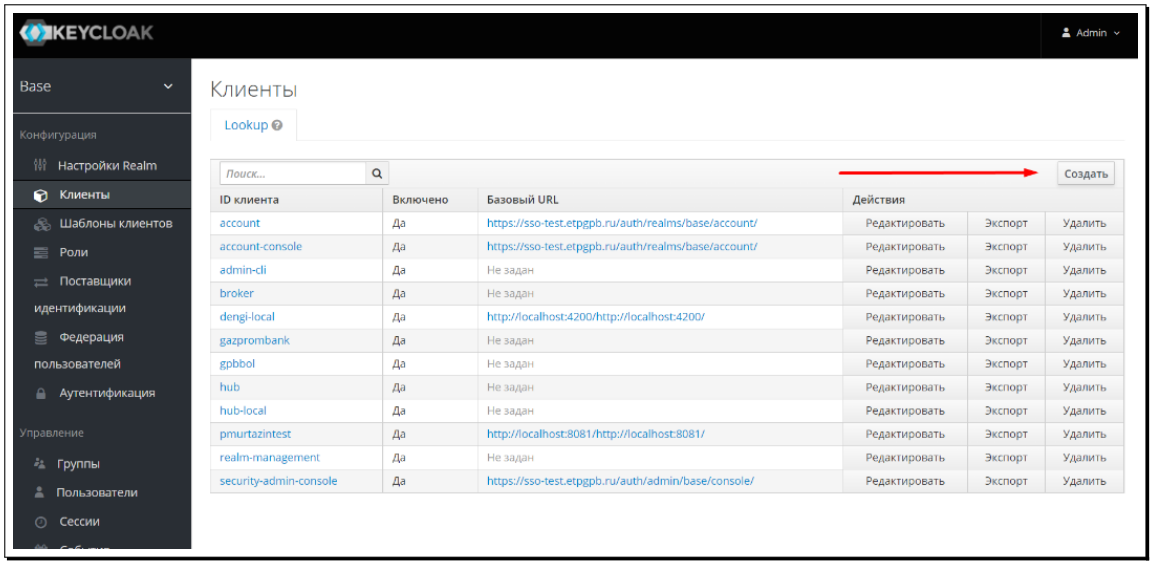


Рисунок 43: Пример начала операции по созданию клиента для межсервисного взаимодействия

Задать имя, аналогичное имени публичного клиента, но с использованием суффикса `-service` в поле `client_id`:

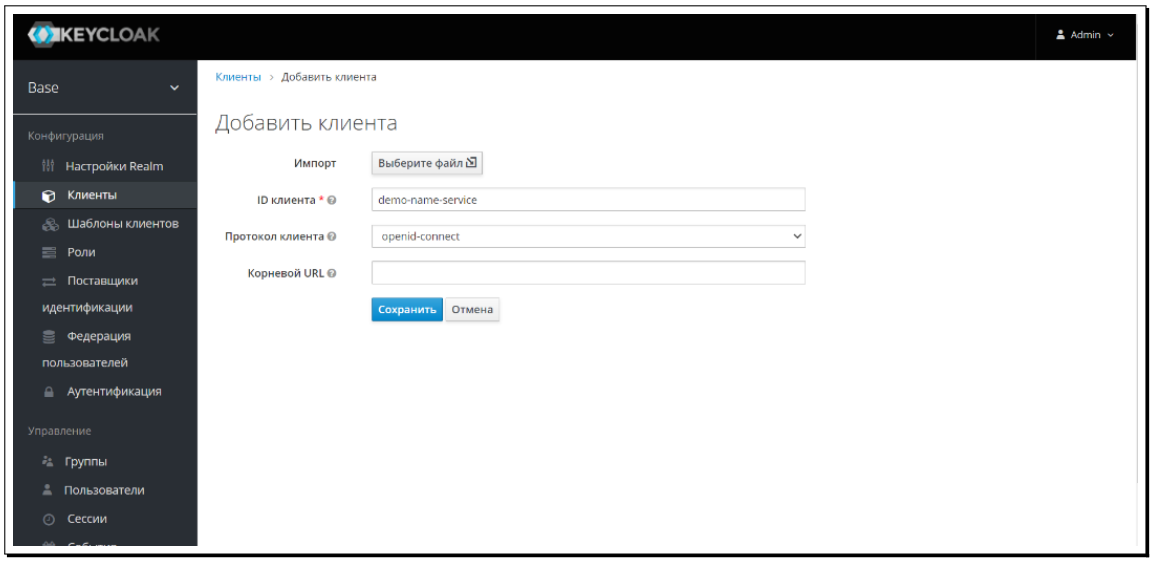


Рисунок 44: Пример задания имени клиента для межсервисного взаимодействия

Указать тип и включить поддержку `client_credentials`:

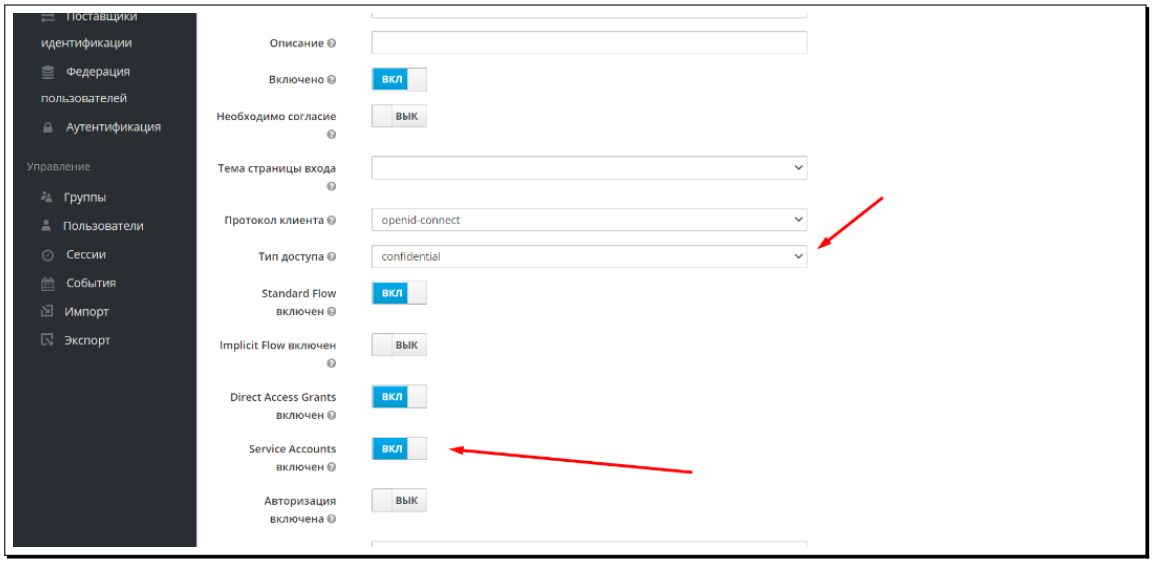


Рисунок 45: Пример задания типа и активизации поддержки `client_credentials`

Пример использования значений `client_secret` и `client_id` в переменных окружения приложения:

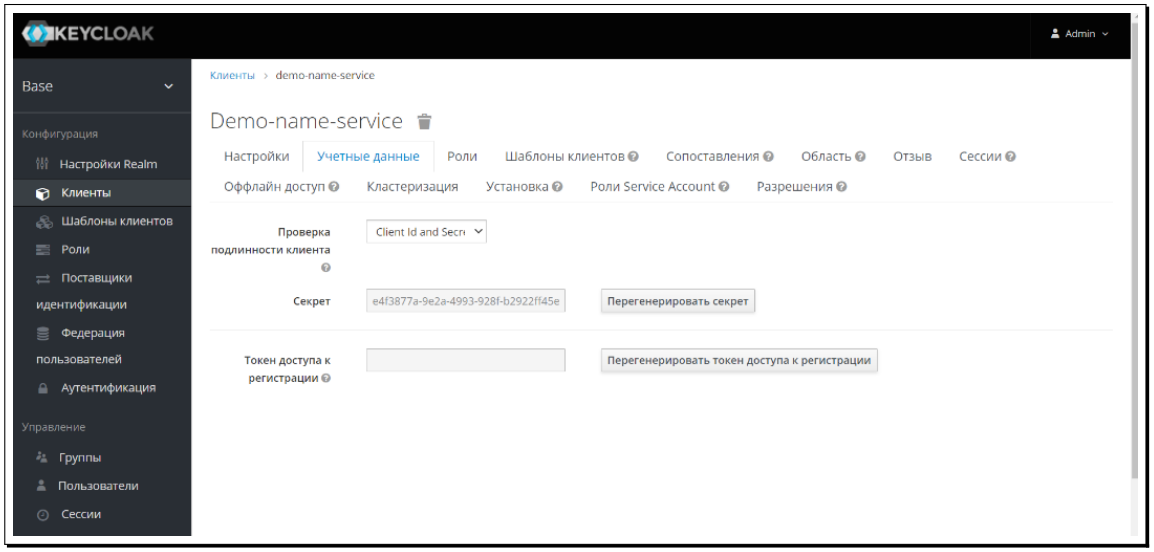


Рисунок 46: Пример использования значений `client_secret` и `client_id` в переменных окружения приложения

Затем задать авторизацию:

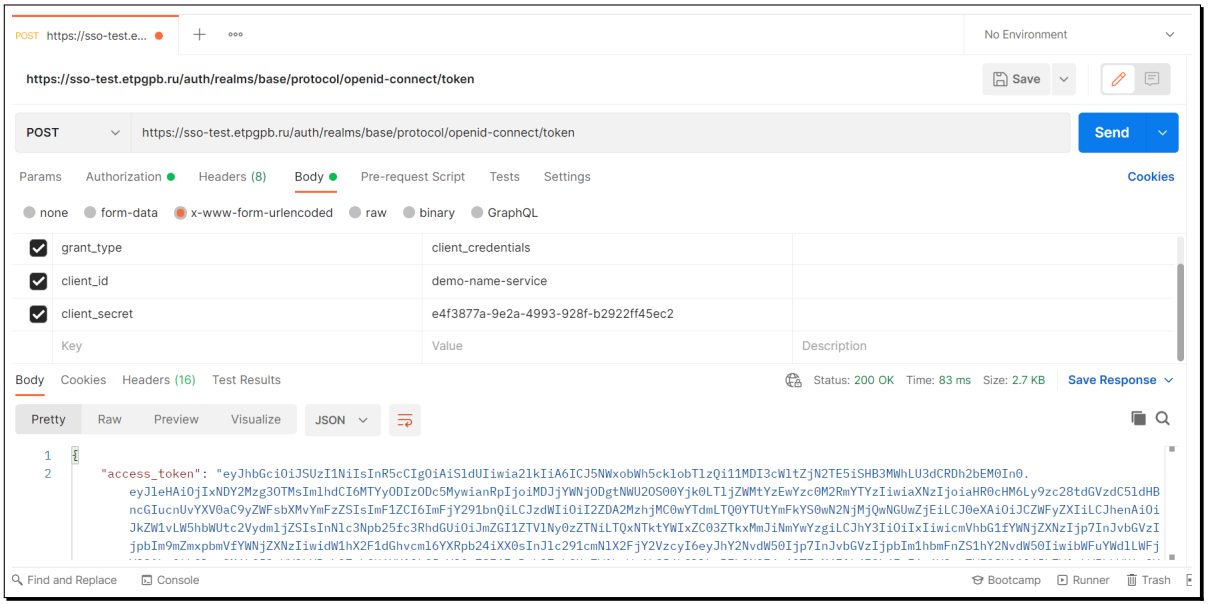


Рисунок 47: Пример авторизации сервиса

8.2 Общесистемные настройки аудита

В настоящем разделе описаны настройки и варианты получения событий (включая события безопасности), с использованием применяемых в составе комплекса средств аудита (включая средства аудита операционной системы как среды выполнения), и сведения по централизованной конфигурации аудита. Дополнительно рекомендуется использовать сведения, приведенные в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение А. Безопасность в ОС Ubuntu.*» *is000-AGD_PRE.1*.

8.2.1 Аудит событий программного комплекса

Для получения информации аудита по событиям, связанным с пользователями прикладных служб системы, требуется выполнить вход от имени администратора IDM KeyCloak и переключиться на вкладку «События», где можно осуществлять поиск, фильтрацию и получение информации о конкретных событиях в системе.

Примеры совершения операций, направленных на получение информации аудита приведена на рисунках Рисунок 48 – Рисунок 62.

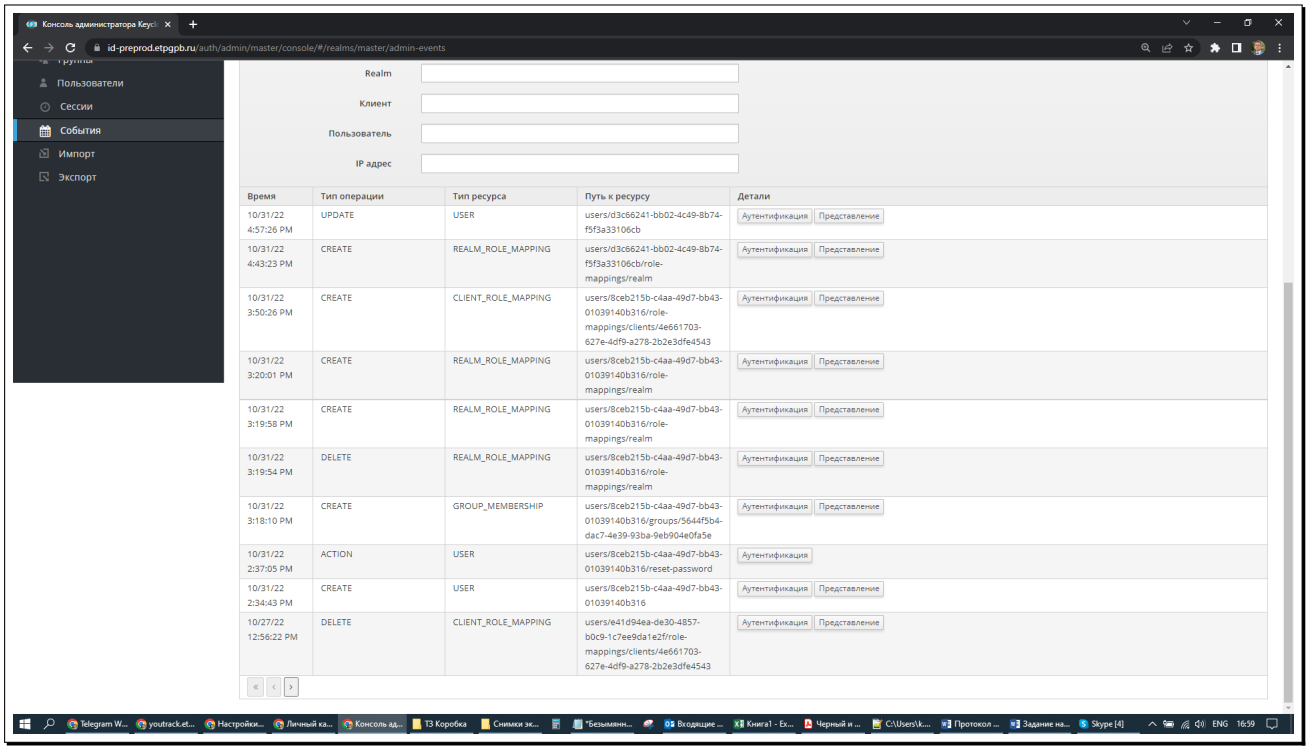


Рисунок 48: Пример меню «События». Общий вид

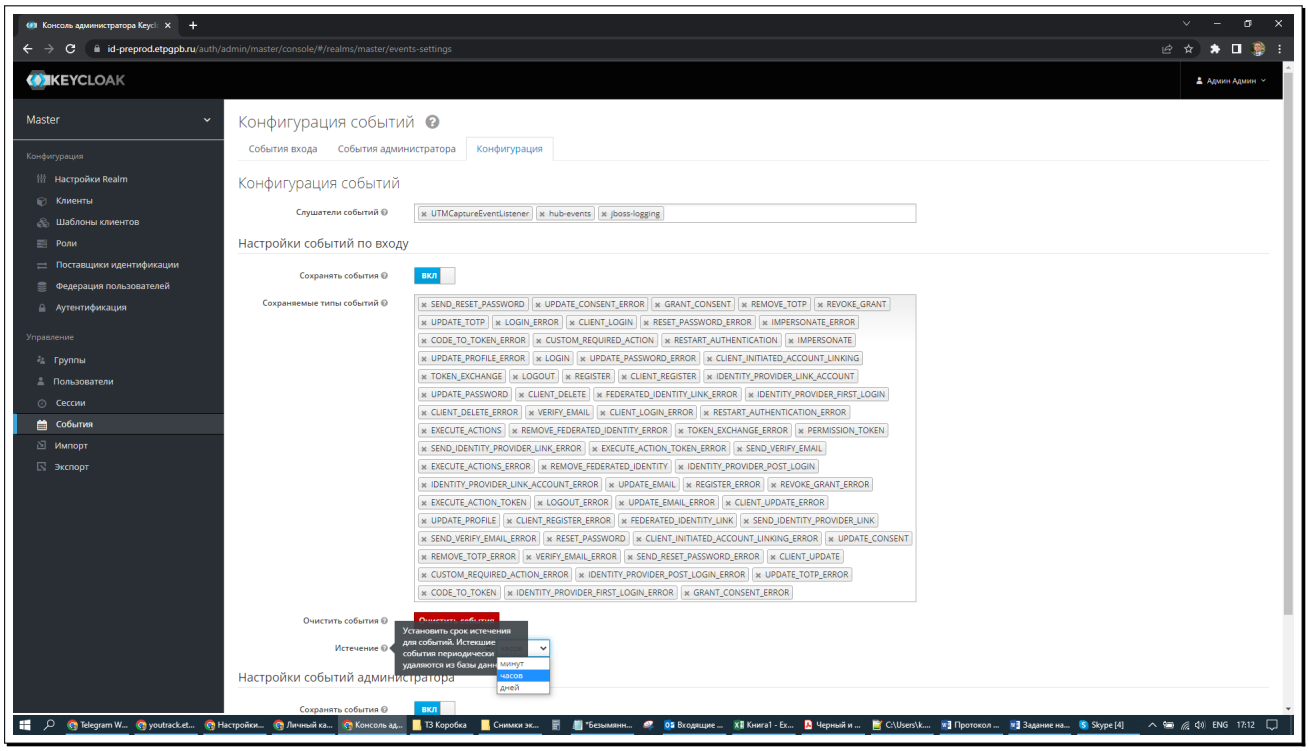


Рисунок 49: Пример конфигурации фильтрации информации аудита и задание срока хранения информации аудита

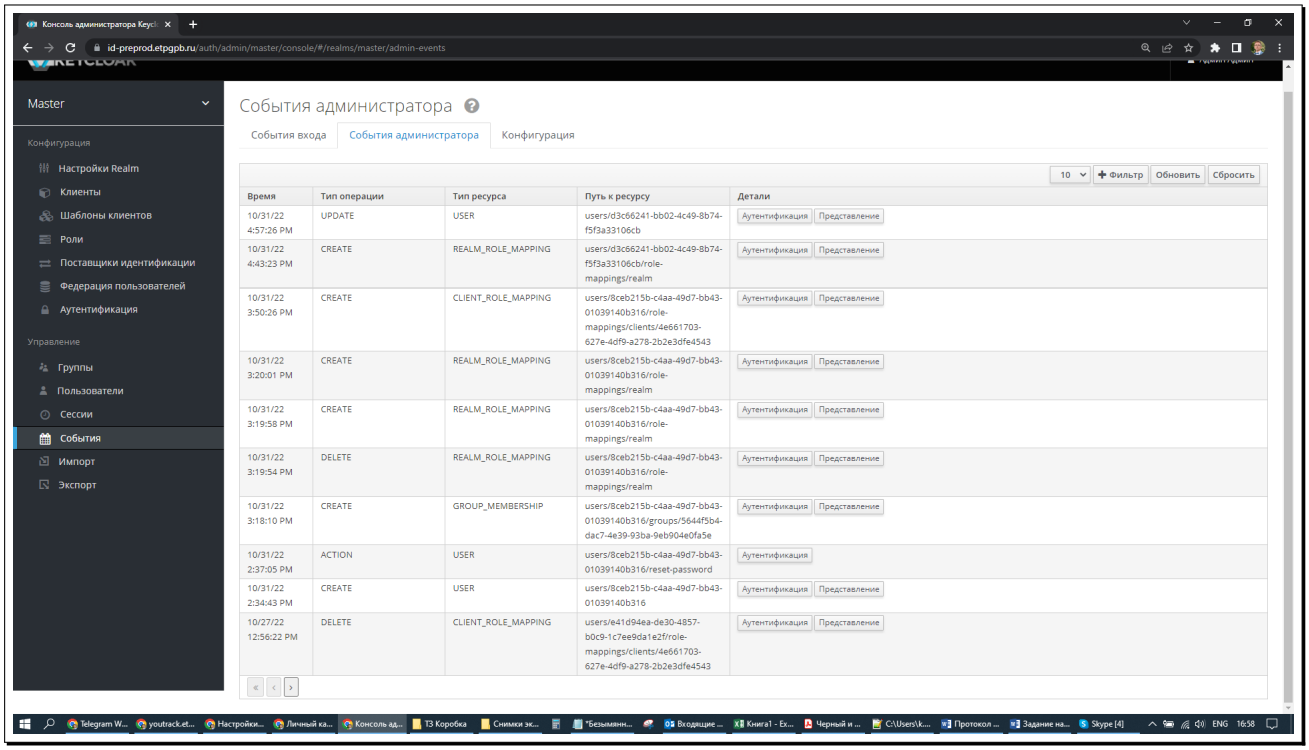


Рисунок 50: Пример информации аудита об операциях над пользователями, группами и ролями

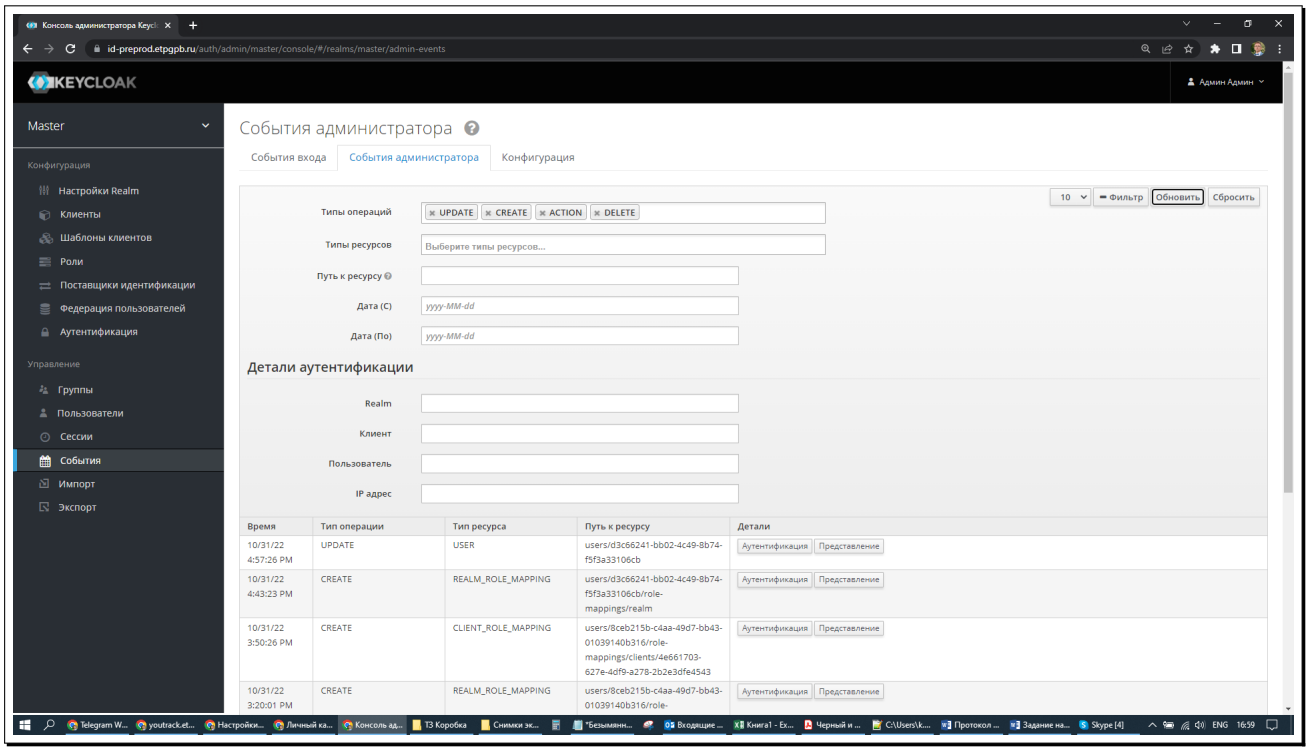


Рисунок 51: Пример задания фильтрации информации аудита по типам событий

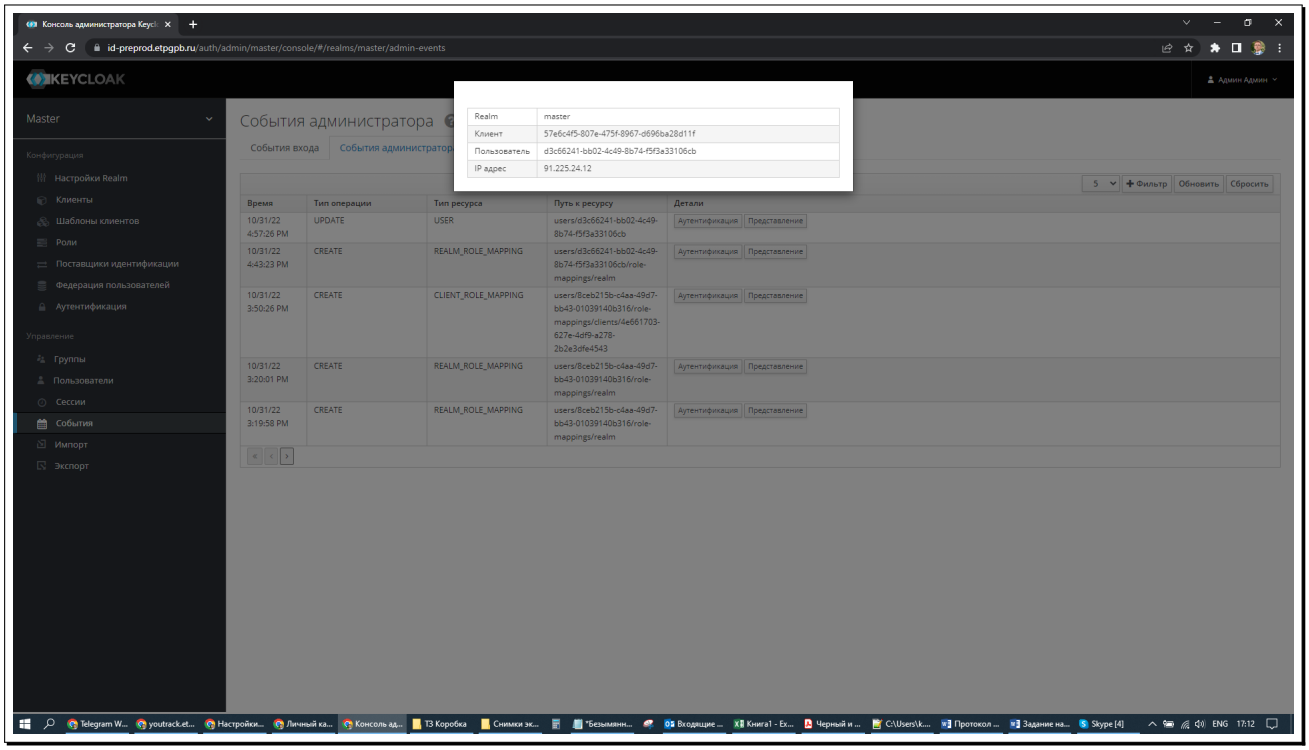


Рисунок 52: Пример детализированного отчета о выбранном событии

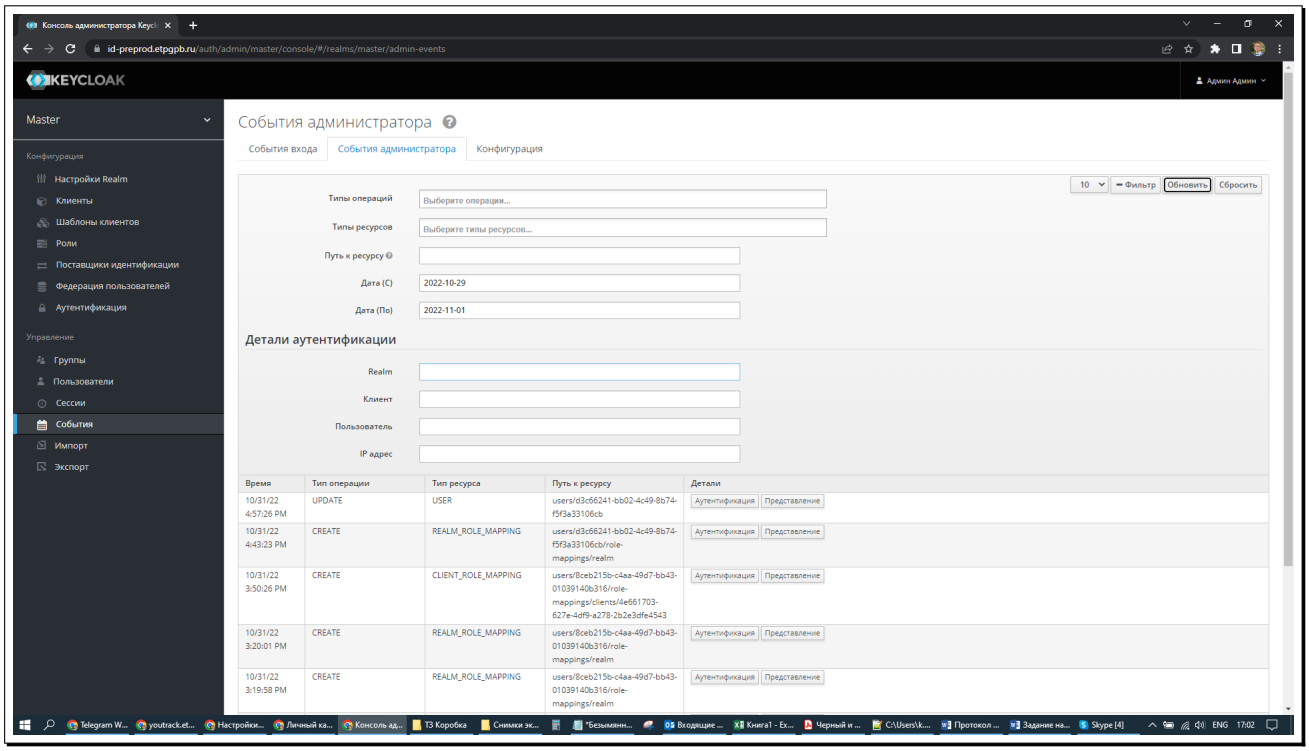


Рисунок 53: Пример задания фильтрации информации аудита по дате событий

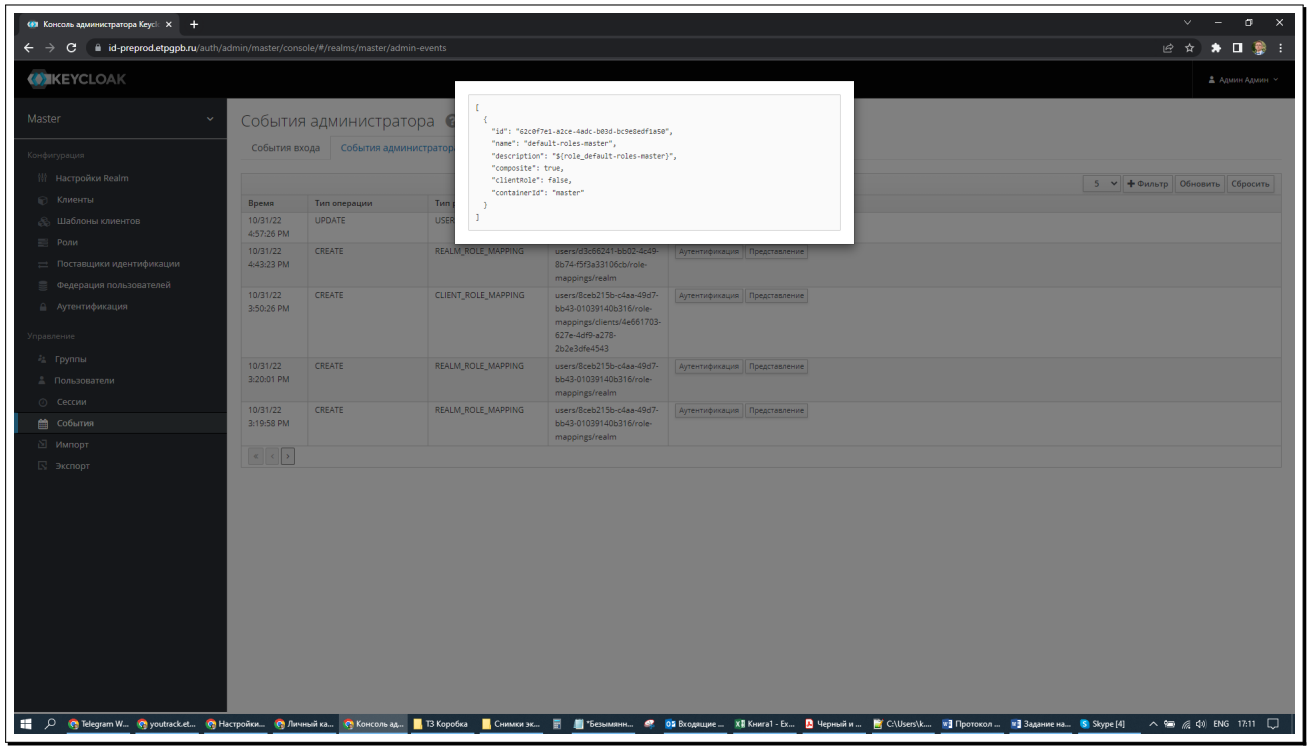


Рисунок 54: Пример детализированного отчета о событии изменения роли

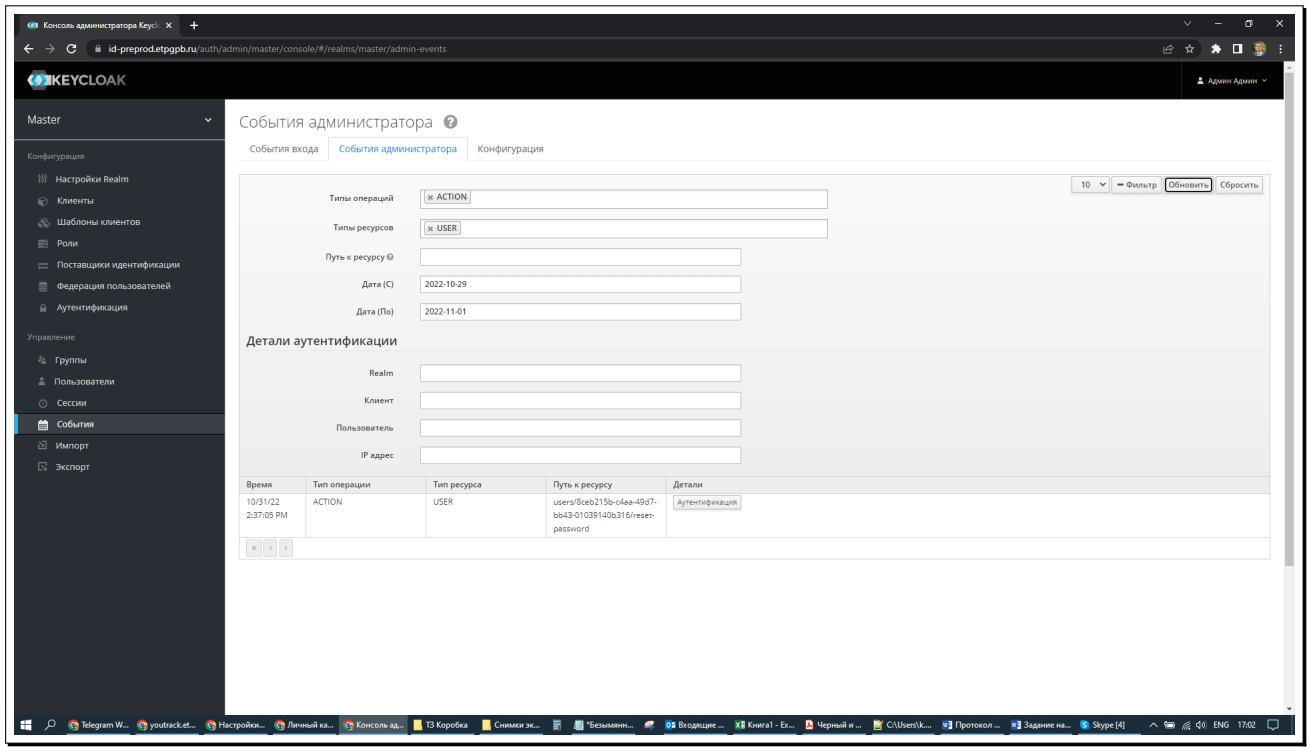


Рисунок 55: Пример задания фильтрации информации аудита по типу, субъекту и дате события

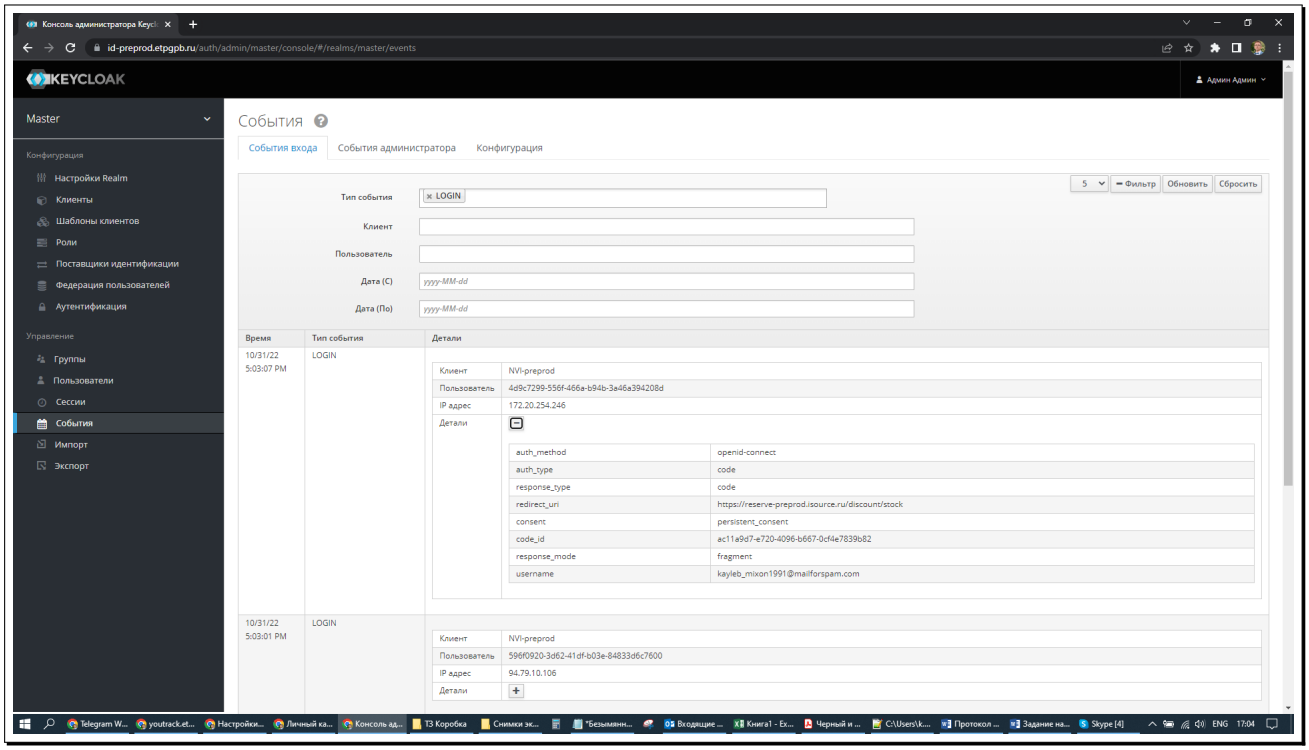


Рисунок 56: Пример получения сведений о событиях входа пользователей

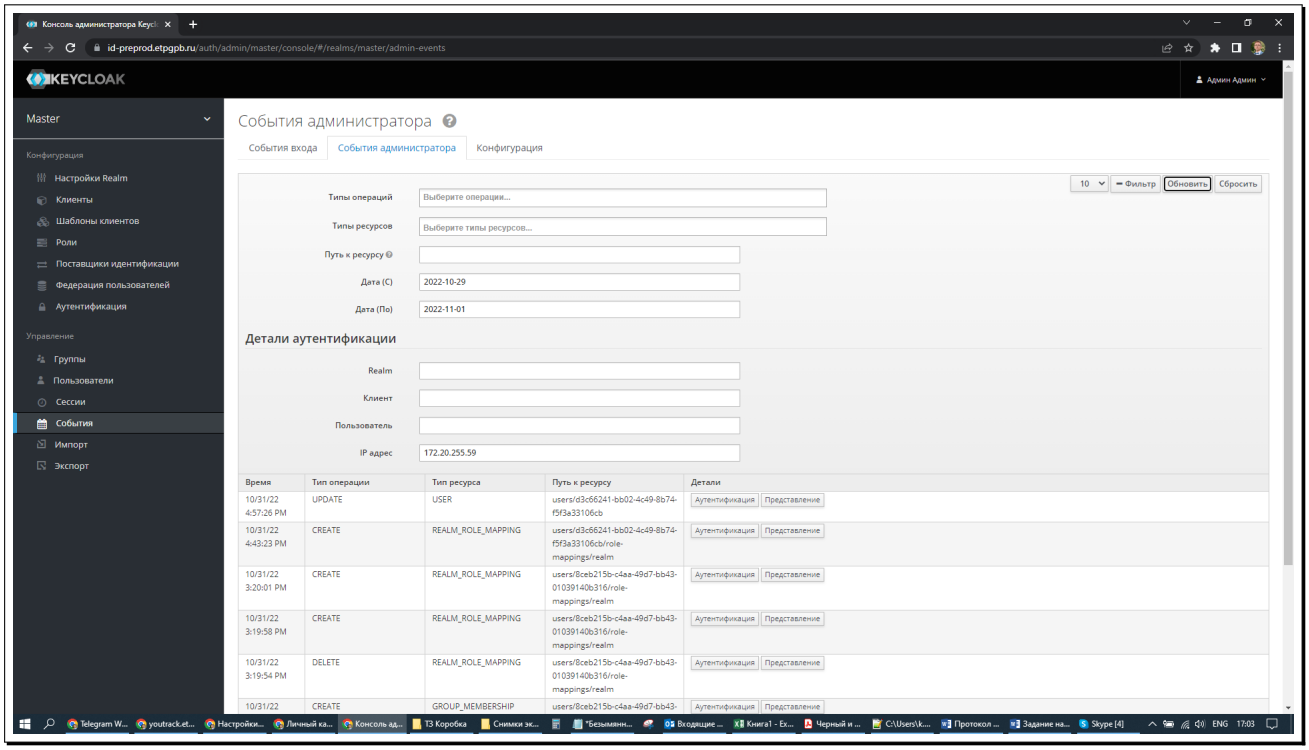


Рисунок 57: Пример получения сведений о событиях создания (изменения) пользователей и ролей

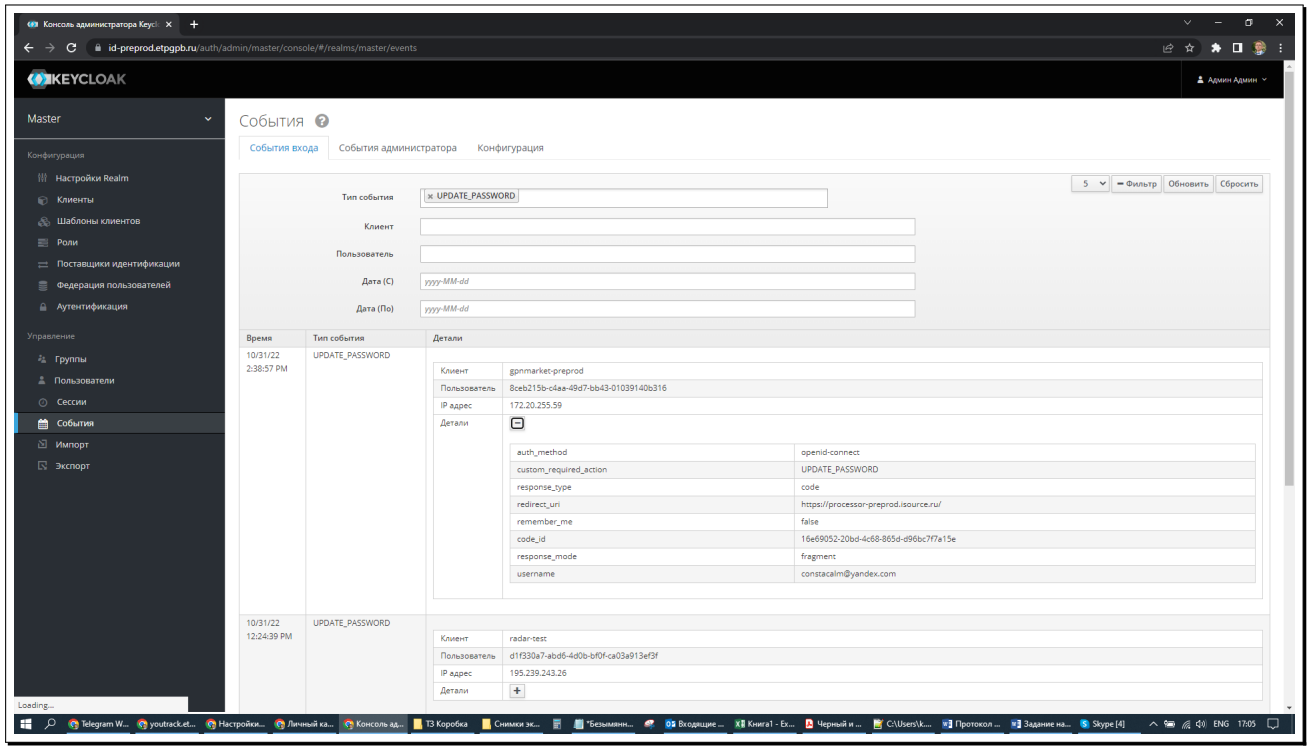


Рисунок 58: Пример получения сведений о событиях связанных с изменением (назначением) пароля

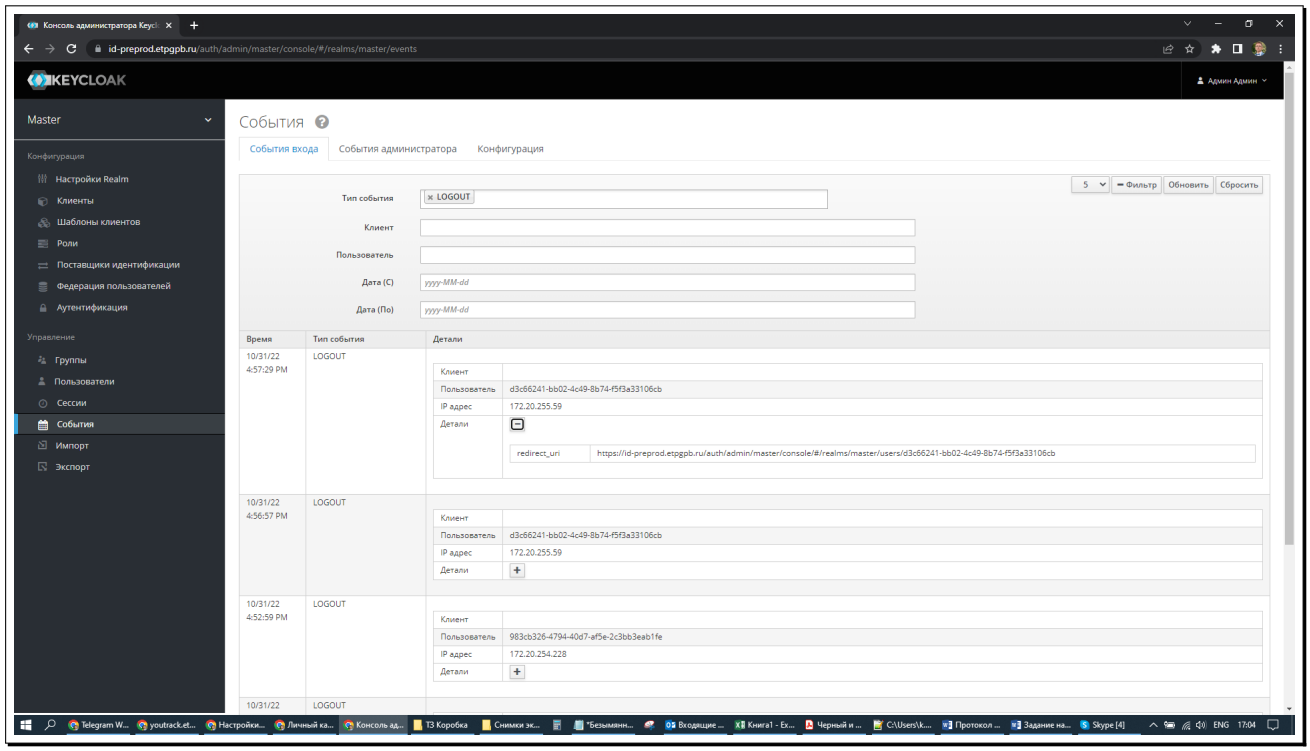


Рисунок 59: Пример получения сведений о событиях выхода пользователей

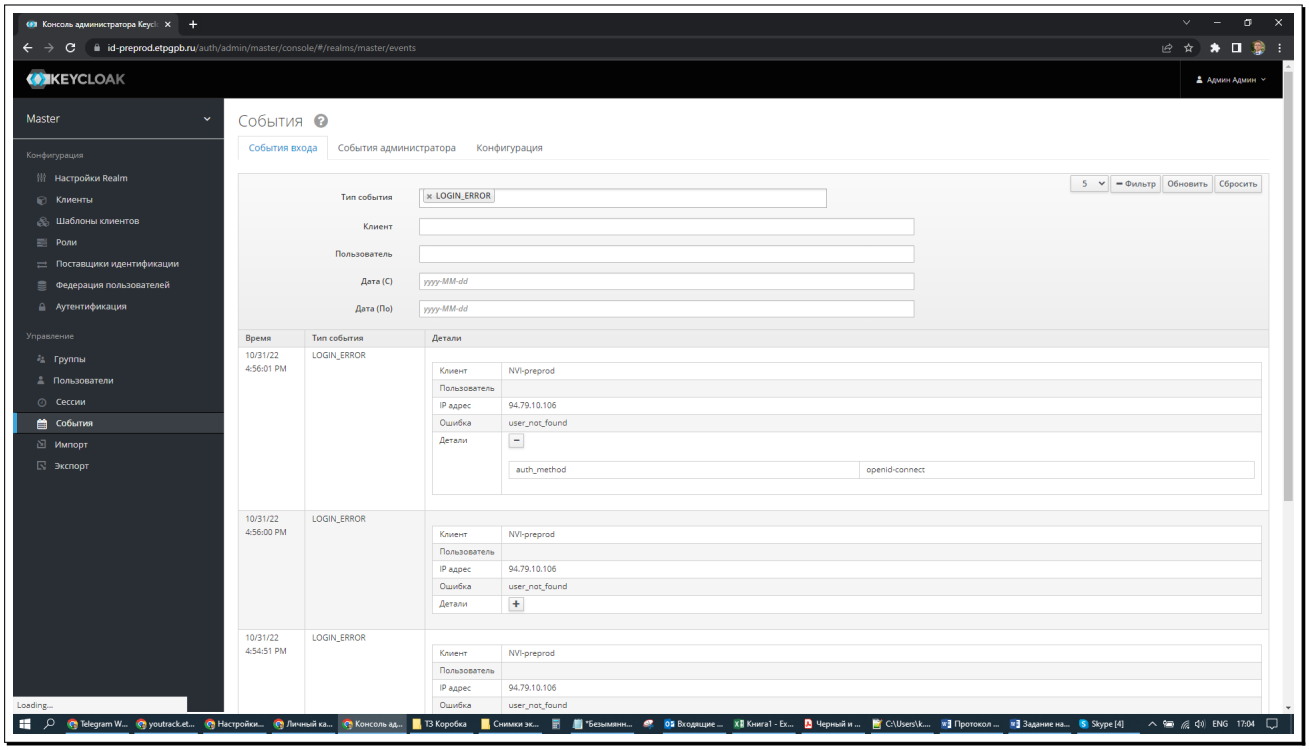


Рисунок 60: Пример получения сведений о событиях, связанных с ошибками входа

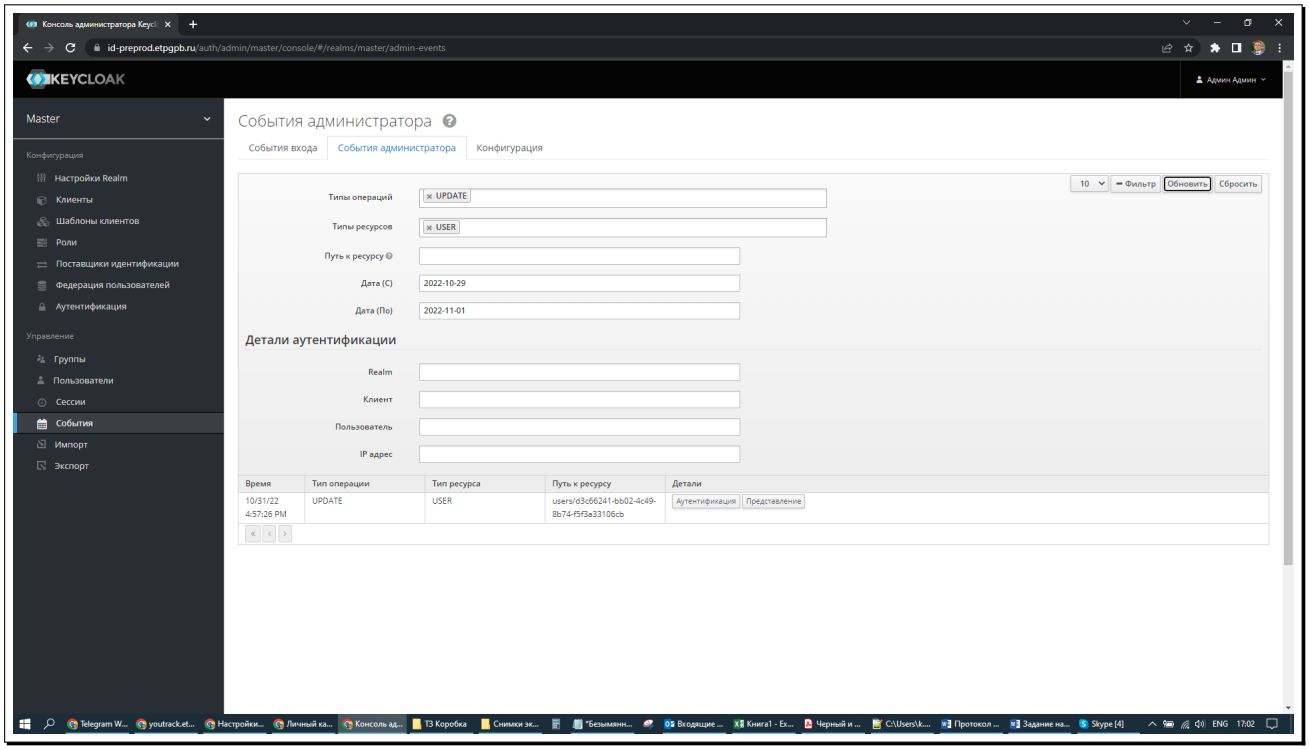


Рисунок 61: Пример задания фильтрации информации аудита по типу, субъекту и дате события, связанных с изменением атрибутов пользователя

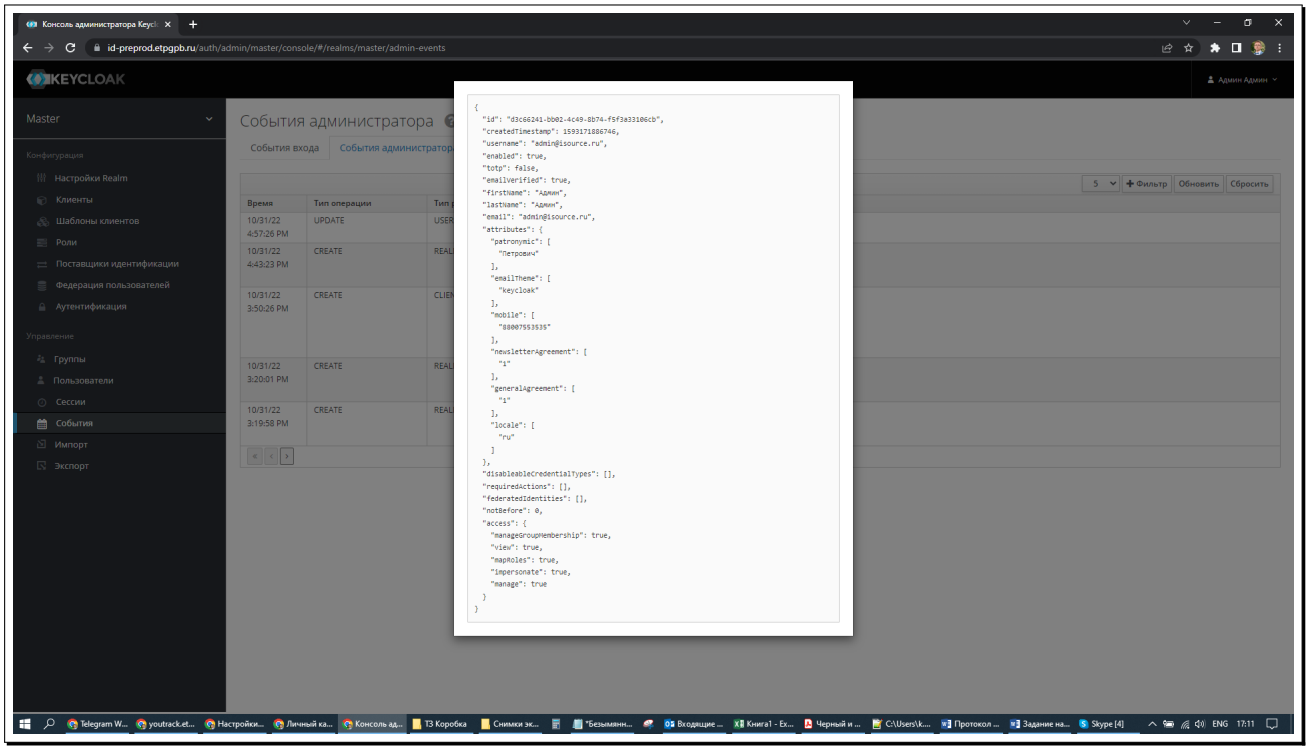


Рисунок 62: Пример детализированного отчета информации аудита по типу, субъекту и дате события, связанных с изменением атрибутов пользователя

8.2.2 Централизованный аудит с помощью `rsyslog`

В системе обеспечивается централизованный сбор, обработка и хранение сообщений аудита. Для этих целей используется служба аудита `rsyslog`. Служба позволяет осуществлять передачу сообщений аудита с помощью протокола TCP, сохранять сообщения с использованием СУБД, а также скрывать содержимое сообщений аудита по пути доставки сообщения от конкретного узла на центральный сервер.

Принципиальная схема централизации сбора данных аудита в программном комплексе приведена на рисунке Рисунок 63.

8.2.2.1 Установка и настройка `rsyslog` для автоматического запуска

Для проверки того, что служба `rsyslog` установлена, требуется выполнить:

```
# dpkg -s rsyslog
Package: rsyslog
Status: install ok installed
Priority: important
...
```

Листинг 94: Проверка наличия в системе службы `rsyslog`

Если вывод свидетельствует о том, что `rsyslog` отсутствует в системе, то требуется установить `rsyslog`:

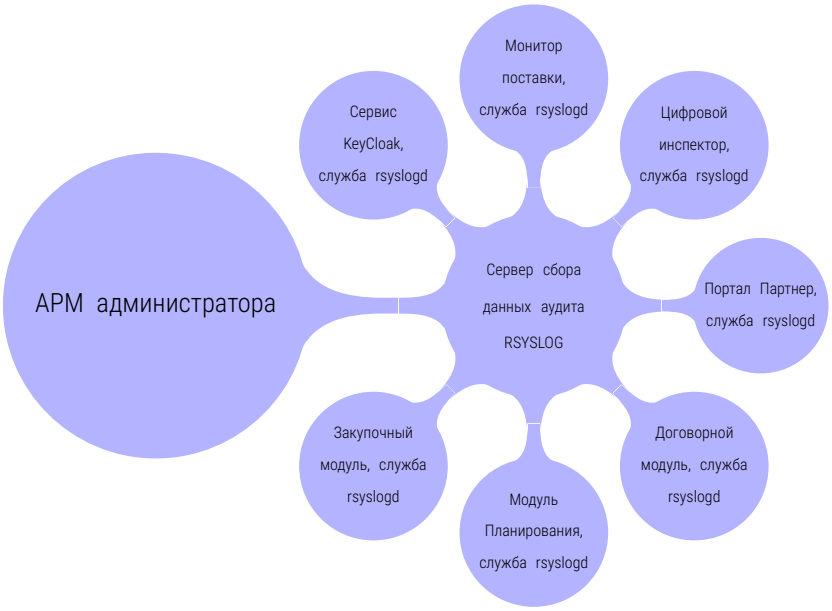


Рисунок 63: Принципиальная схема централизации сбора данных аудита в программном комплексе

```
# apt install rsyslog
```

Листинг 95: Установка в систему службы `rsyslog`

Для проверки того, что служба `rsyslog` сконфигурирована для автоматического запуска, выполнить:

```
# systemctl is-enabled rsyslog
enabled
```

Листинг 96: Проверка включения службы `rsyslog` для автоматического запуска

Иначе, требуется выполнить настройку `rsyslog` для автоматического запуска:

```
# systemctl --now enable rsyslog
```

Листинг 97: Включение службы `rsyslog` для автоматического запуска

Для проверки того, установлены ли в конфигурационном файле параметры (адрес или имя) удаленного централизованного сервера сбора информации аудита, выполнить:

```
# grep -E "^[^#]\s*\S+\.\*\s+"
```

Листинг 98: Проверка настроек службы `rsyslog`

Вывод должен содержать имя и/или адрес централизованного сервера `rsyslog`.

8.2.2.2 Аудит событий в `rsyslog`

Конфигурационный файл rsyslog, а именно /etc/rsyslog.conf[18] определяет политики (правила) обработки аудита. Предполагается, что как минимум, на удаленный централизованый сервер аудита будут отправлены данные о следующих событиях[19]:

– успешные и неуспешные операции переключения контекста с помощью su;
– неуспешные попытки входа пользователей;
– любые попытки совершить вход в контексте полномочий суперпользователя (root);
– информация аудита от почтовой службы (при наличии);
– информация аудита всех системных служб;
– любая информация аудита с уровнем критичности не менее warning;
– информация аудита о загрузке.

Для конфигурации rsyslog на передачу указанной выше информации аудита, внести следующее в /etc/rsyslog.conf:

```
*.emerg                 :omusrmsg:*
auth,authpriv.*         /var/log/auth.log
mail.*                  -/var/log/mail
mail.info               -/var/log/mail.info
mail.warning            /var/log/mail.warn
mail.err                /var/log/mail.err
news.crit               -/var/log/news/news.crit
news.err                -/var/log/news/news.err
news.notice             -/var/log/news/news.notice
*.=warning;*.=err       -/var/log/warn
*.crit                  /var/log/warn
*.*;mail.none;news.none -/var/log/messages
local0,local1.*         -/var/log/localmessages
local2,local3.*         -/var/log/localmessages
local4,local5.*         -/var/log/localmessages
local6,local7.*         -/var/log/localmessages
user.* -                /var/log/user.log
kern.* -                /var/log/kern.log
daemon.* -              /var/log/daemon.log
```

Листинг 99: Рекомендуемые параметры настройки службы rsyslog

Для применения настройки выполнить перезапуск rsyslog:

```
# systemctl reload rsyslog
```

Листинг 100: Перезапуск службы rsyslog

8.2.2.3 Установка прав на файлы аудита rsyslog

Необходимо, чтобы файлы аудита rsyslog не были доступны для чтения обычным пользователям ОС. Это позволит избежать возможного раскрытия информации о системе. Рекомендуемая маска прав доступа – 0640.

[18] Могут использоваться файлы конфигурации в /etc/rsyslog.d/conf/.
[19] Для указания дополнительных данных рекомендуется свериться с содержимым каталога /var/log и принять решение о том, что еще можно отправлять на удаленный централизованный сервер аудита.

Для проверки прав доступа выполнить:

```
# grep ^\s*\$FileCreateMode /etc/rsyslog.conf /etc/rsyslog.d/*.conf
/etc/rsyslog.conf:$FileCreateMode 0640
```

Листинг 101: Проверка настроек прав доступа журналов аудита службы `rsyslog`

Если значение маски отличается, требуется установить корректное значение. Для этого выполнить редактирование файла `/etc/rsyslogd.conf` и установить значение маски прав доступа в директиве `$FileCreateMode` в значение `0640`, или более строгое:

```
# echo "$FileCreateMode 0640" >> /etc/rsyslogd.conf
```

Листинг 102: Настройка прав доступа к журналам аудита службы `rsyslog`

Для применения настройки выполнить перезапуск `rsyslog`:

8.2.2.4 Аудит `systemd-journald` совместно с `rsyslog`

Служба инициализации `systemd` имеет собственную служба аудита – `systemd-journald`. Эта служба аудита всегда выполняется, пока выполняется основная служба инициализации `systemd`.

Данные аудита, предоставляемые `systemd-journald` также должны быть переданы в службу `rsyslog` для централизованного сбора, обработки и хранения. Для проверки того, что служба аудита `systemd-journald` настроена на отправку событий в `rsyslog`, выполнить следующее:

```
# grep -e ForwardToSyslog /etc/systemd/journald.conf
ForwardToSyslog=yes
```

Листинг 103: Проверка настроек службы аудита `systemd-journald`

Иначе, требуется произвести соответствующую настройку. Для этого нужно отредактировать файл `/etc/systemd/journald.conf` и указать директивы, которые обязуют `systemd-journald` всегда отправлять сообщения аудита в `rsyslog`, сжимая их при отправке (для экономии трафика) и сохраняя при этом локальную копию сообщений:

```
ForwardToSyslog=yes
Compress=yes
Storage=persistent
```

Листинг 104: Настройка службы аудита `systemd-journald`

После этого перезапустить службу `systemd-journald`:

```
# systemctl restart systemd-journald
```

Листинг 105: Перезупуск службы аудита `systemd-journald`

8.2.2.5 Журналы `rsyslog` их права доступа и ротация

Требуется убедится, что данные аудита не доступны обычным пользователям ОС для чтения. Для проверки выполнить:

```
# find /var/log -type f -ls
```

Листинг 106: Проверка прав доступа к журналам аудита ОС

Пример вывода:

```
...
844 -rw-r-----   1 syslog   adm        861296 Apr 25 00:00 /var/log/syslog.7.gz
1408 -rw-r-----   1 syslog   adm       1434306 May  5 00:00 /var/log/syslog.1
0 -rw-r-----   1 root     utmp           0 May  3 00:02 /var/log/btmp
20 -rw-------   1 root     root        18970 Apr 22 10:19 /var/log/boot.log.7
4 -rw-r-----   1 root     root         2563 Mar  2 11:43 /var/log/dpkg.log.4.gz
60 -rw-r-----   1 root     root        54522 May  5 11:46 /var/log/dpkg.log
4 -rw-r-----   1 syslog   adm          2942 Apr 20 15:00 /var/log/mail.log.3.gz
0 -rw-r-----   1 root     utmp           0 Apr  2 00:00 /var/log/btmp.1
136 -rw-------   1 root     root       135926 Apr 25 00:00 /var/log/boot.log.6
328 -rw-r-----   1 syslog   adm        333807 Apr 21 10:22 /var/log/kern.log.3.gz
4 -rw-r-----   1 root     root         1164 Mar  8 22:26 /var/log/dpkg.log.3.gz
204 -rw-r-----   1 syslog   adm        208424 Apr 28 10:18 /var/log/syslog.4.gz
0 -rw-r-----   1 root     adm            0 Mar  2 14:47 /var/log/rkhunter.log
```

Листинг 107: Пример вывода корректных прав доступа к журналам аудита ОС

В случае, если вывод свидетельствует о назначении публичных прав на чтение, выполнить переназначение прав на журналы аудита:

```
# find /var/log -type f -exec chmod g-wx,o-rwx "{}" + -o -type d -exec chmod g-w,o-rwx "{}" +
```

Листинг 108: Назначение корректных прав доступа к журналам аудита ОС

Внести в файл `/etc/logrotate.conf` директиву (если отсутствует), не позволяющую обычным пользователям просматривать журналы аудита:

```
# echo "create 0640 root utmp" >> /etc/logrotate.conf
```

Листинг 109: Настройка ротации журналов аудита и прав доступа к ротированным журналам

Для применения настройки выполнить перезапуск `rsyslog`:

8.2.3 Использование анализатора аудита `logwatch`

Чтобы не запутаться в большом количестве сообщений аудита рекомендуется применять программы, которые умеют собирать и анализировать сообщения аудита. Затем такие программы выдают отчеты в удобном виде. Например, для этих целей можно использовать программу `logwatch`.

Для её установки нужно выполнить команду:

```
# apt install logwatch
```

Листинг 110: Установка `logwatch`

После этого можно сразу получать отчеты:

```
# logwatch --detail High --range Yesterday --logdir /var/cache/logwatch

 ################### Logwatch 7.5.2 (07/22/19) ####################
Processing Initiated: Mon Aug  8 19:02:23 2022
Date Range Processed: yesterday
( 2022-Aug-07 )
Period is day.
Detail Level of Output: 10
Type of Output/Format: stdout / text
Logfiles for Host: rosafresh12.tiger.kingdom
 ##################################################################

 --------------------- Disk Space Begin ------------------------

 Filesystem      Size  Used Avail Use% Mounted on
 /dev/sda2        12G  7.4G  3.5G  68% /
 /dev/sda7       1.9G  383M  1.4G  22% /var
 /dev/sda9       966M  179M  722M  20% /var/tmp
 /dev/sda5       2.9G  318M  2.4G  12% /var/log
 /dev/sda3       4.8G   16M  4.5G   1% /var/log/audit
 /dev/sda1       488M  189M  263M  42% /boot
 /dev/sda10      7.4G   91M  6.9G   2% /home
 /dev/sda8       966M  120K  900M   1% /tmp


 ---------------------- Disk Space End -------------------------
 --------------------- Connections (secure-log) Begin ------------------------


 **Unmatched Entries**
 PackageKit: uid 1000 is trying to obtain org.freedesktop.packagekit.system-sources-refresh auth (
   only_trusted:0): 1 Time(s)
 PackageKit: uid 1000 obtained auth for org.freedesktop.packagekit.system-sources-refresh: 1 Time(s)
 systemd-logind: Lid closed.: 1 Time(s)
 systemd-logind: Lid opened.: 1 Time(s)
 systemd-logind: Operation 'sleep' finished.: 2 Time(s)
 systemd-logind: Power key pressed.: 1 Time(s)
 systemd-logind: Suspending ... : 1 Time(s)

 ---------------------- Connections (secure-log) End -------------------------

 --------------------- lm_sensors output Begin ------------------------

 ucsi_source_psy_USBC000:001-isa-0000
 Adapter: ISA adapter
 in0:           5.00 V  (min =  +5.00 V, max =  +5.00 V)
 curr1:         0.00 A  (max =  +0.00 A)

 iwlwifi_1-virtual-0
 Adapter: Virtual device
 temp1:        +37.0 C

 pch_cannonlake-virtual-0
```

```
Adapter: Virtual device
temp1:        +45.0 C

nvme-pci-0200
Adapter: PCI adapter
Composite:    +33.9 C  (low  = -273.1 C, high = +82.8 C)
(crit = +86.8 C)
Sensor 1:     +33.9 C  (low  = -273.1 C, high = +65261.8 C)

acpitz-acpi-0
Adapter: ACPI interface
temp1:        +25.0 C  (crit = +107.0 C)

coretemp-isa-0000
Adapter: ISA adapter
Package id 0:  +48.0 C  (high = +100.0 C, crit = +100.0 C)
Core 0:        +48.0 C  (high = +100.0 C, crit = +100.0 C)
Core 1:        +48.0 C  (high = +100.0 C, crit = +100.0 C)
Core 2:        +48.0 C  (high = +100.0 C, crit = +100.0 C)
Core 3:        +46.0 C  (high = +100.0 C, crit = +100.0 C)

dell_smm-virtual-0
Adapter: Virtual device
fan1:           0 RPM

BAT0-acpi-0
Adapter: ACPI interface
in0:           7.40 V
curr1:         1.51 A


---------------------- lm_sensors output End -------------------------
```

Листинг 111: Пример использования `logwatch`

Эти отчеты можно получать регулярно с помощью `cron` или `systemd`, и направлять по почте нужному пользователю.

8.2.4 Аудит с помощью `auditd`

8.2.4.1 Проверка наличия в системе службы аудита `auditd`

Для проверки того, установлена ли в системе служба аудита `auditd` выполнить:

```
# dpkg -s auditd audispd-plugins
Package: auditd
Status: install ok installed
....
Package: audispd-plugins
Status: install ok installed
....
```

Листинг 112: Проверка наличия в системе службы аудита `auditd`

Для проверки того, запускается ли при старте системы служба `auditd` выполнить:

```
# systemctl is-enabled auditd
enabled
```

Листинг 113: Проверка того, запущена ли в системе служба аудита `auditd`

Если вывод отличается от приведенного выше, требуется установить службу аудита и настроить ее на автоматический запуск при старте ОС.

```
# apt install auditd audispd-plugins
```

Листинг 114: Установка в систему службы аудита `auditd`

Для запуска службы аудита `auditd` (в том числе при старте операционной системы) выполнить:

```
# systemctl --now enable auditd
```

Листинг 115: Включение службы аудита `auditd` для автоматического запуска при старте ОС

8.2.4.2 Настройка сбора информации о событиях до старта `auditd`

Поскольку запуск службы аудита производится системной службой инициализации `systemd`, то следовательно, сначала будет загружено ядро ОС вместе со всеми модулями и системным окружением (`initrd`). Затем будет запущена сама служба инициализации `systemd`, потом еще ряд системных служб, и только после всего этого будет запущена служба аудита `auditd`. Для того, чтобы было возможно отслеживать потенциальную подозрительную активность даже при описанном выше поведении системы, требуется сообщить ОС, чтобы события регистрировались бы ядром. При старте службы аудита `auditd` события должны быть переданы из ядра в службу `auditd` для обработки и хранения.

Для проверки того, настроена ли фиксация для событий, предшествующих запуску службы `auditd` требуется выполнить:

```
# grep "^\s*linux" /etc/default/grub | grep -v "audit=1"
```

Листинг 116: Проверка наличия функции регистрации событий до запуска службы аудита

Вывода быть не должно.

Иначе требуется настроить ОС на фиксацию событий, предшествующих запуску службы аудита `auditd`.

Для настройки системы на фиксацию событий, предшествующих запуску службы аудита `auditd`, выполнить следующее. Отредактировать параметры загрузчика `/etc/default/grub` и добавить «`audit=1`» к параметрам загрузки и затем обновить конфигурацию загрузчика:

```
GRUB_CMDLINE_LINUX="audit=1"
```

Листинг 117: Настройка регистрации событий до запуска службы аудита

При указании опции «`audit=1`», как указано выше, кольцевой буфер аудита ядра способен хранить только 64 записи. В том случае, если при загрузке ОС будет создано более 64-х записей, какие-то

сообщения могут потеряться. Это приведет к невозможности отслеживать потенциальную подозрительную активность. Чтобы этого избежать, требуется установить размер кольцевого буфера аудита ядра для хранения как минимум 8192 байт сообщений, или более.

Для проверки того, установлена ли конфигурация кольцевого буфера аудита ядра требуется выполнить:

```
# grep "^\s*linux" /proc/cmdline | grep -v "audit_backlog_limit="
```

Листинг 118: Проверка наличия кольцевого буфера аудита ядра. Вариант 1

Или:

```
# grep "^\s*linux" /boot/grub/grub.cfg | grep -v "audit_backlog_limit="
```

Листинг 119: Проверка наличия кольцевого буфера аудита ядра. Вариант 2

Вывода быть не должно. Иначе требуется задать конфигурацию кольцевого буфера аудита ядра.

Для проверки же текущего значения (в байтах) размера кольцевого буфера аудита ядра, в том случае, если буфер сконфигурирован, требуется выполнить:

```
# grep "audit_backlog_limit=" /proc/cmdline
... audit_backlog_limit=8192 ...
```

Листинг 120: Проверка текущего значения кольцевого буфера аудита ядра. Вариант 1

Или:

```
# grep "audit_backlog_limit=" /boot/grub/grub.cfg
... audit_backlog_limit=8192 ...
```

Листинг 121: Проверка текущего значения кольцевого буфера аудита ядра. Вариант 2

И убедиться при этом, что значение буфера составляет не менее 8192 байт. В том случае, если текущее значение буфера меньше, нужно настроить значение в 8192 байт или более.

Для настройки использования кольцевого буфера аудита ядра и задания ему значения 8192 байт требуется внести изменения в файл конфигурации загрузчика GRUB2 `/etc/defaults/grub`:

```
GRUB_CMDLINE_LINUX="audit_backlog_limit=8192"
```

Листинг 122: Настройка кольцевого буфера аудита ядра и его параметров

Затем обновить конфигурацию загрузчика:

```
# update-grub
```

Листинг 123: Обновление конфигурации загрузчика при модификации параметров буфера аудита ядра

8.2.4.3 Настройка размера журнала аудита

Важно знать, что по-умолчанию служба `auditd` сохраняет сообщения аудита в файл, размером не более 5 Мбайт, и сохраняет только 4 копии файла. Предыдущие версии удаляются и заменяются

более новыми. Требуется скоординировать конфигурацию хранения и ротации журналов в соответствии с политикой, принятой в конкретной информационной системе. Эта политика может зависеть не только от объема записей. Также может быть использована политика, ориентированная на время хранения, или на количество записей. Поэтому в данном случае сложно выдать какие-то универсальные рекомендации. Однако, обычно в качестве критерия максимального значения журнала используется размер файла. В данном примере подразумевается, что используется настройка, ориентированная на размер файла, но не на время или количество записей.

Для проверки текущего значения размера файла аудита нужно выполнить проверку значения параметра `max_log_file` в файле `/etc/audit/auditd.conf`, указав нужное значение в мегабайтах:

```
# grep max_log_file /etc/audit/auditd.conf
max_log_file = <MB>
```

Листинг 124: Проверка текущей политики, задающей размер файла аудита `auditd`

Для настройки нужного значения размера файла аудита, требуется установить нужное значение параметра `max_log_file` в файле `/etc/audit/auditd.conf`:

```
max_log_file = <MB>
```

Листинг 125: Настройка параметра, определящего размер файла аудита `auditd`

Затем перезапустить службу `auditd`.

```
# systemctl restart auditd
```

Листинг 126: Пример перезапуска службы `auditd` при изменении конфигурации

8.2.4.4 Настройка хранения журналов аудита

Можно настроить ОС так, чтобы все без исключения журналы аудита постоянно хранились, и не перезаписывались[20].

Если переменная `max_log_file_action` файла `/etc/audit/auditd.conf` установлена в значение `keep_logs`, то это означает, что служба аудита будет хранить все сообщения, и никогда не будет перезаписывать журналы. Нужно учитывать, что рано или поздно это приведет к переполнению файловой системы `[/var/log/audit]`.

Для проверки текущего значения этой переменной, выполнить:

```
# grep "max_log_file_action" /etc/audit/auditd.conf
max_log_file_action = keep_logs
```

Листинг 127: Проверка наличия политики, определяющей поведение аудита при достижении лимита

[20] Хранение журналов аудита без их ротации (перезаписи) применяется реже, однако это зависит от политики аудита, принятой в информационной системе.

Если вывод отличается, то для того, чтобы auditd сохранял все сообщения, нужно установить переменную max_log_file_action в значение keep_logs в файле /etc/audit/auditd.conf и перезапустить службу auditd.

8.2.4.5 Настройка системы при достижении лимитов аудита

Служба аудита auditd может быть настроена на автоматический останов операционной системы, при переполнении соответствующей ФС (обычно /var/log/audit) журналами аудита. В этом случае указанному администратору (в примере – суперпользователю root) будет отправлено почтовое сообщение. В качестве почтового адреса можно указать адрес администратора аудита в конкретной информационной системе. Для проверки того, настроена ли операционная система на автоматический останов нужно выполнить:

```
# grep "space_left_action" /etc/audit/auditd.conf
space_left_action = email
# grep "action_mail_acct" /etc/audit/auditd.conf
action_mail_acct = root
# grep "admin_space_left_action" /etc/audit/auditd.conf
admin_space_left_action = halt
```

Листинг 128: Проверка значений текущей политики при достижении лимита auditd

Если вывод отличается от приведенного выше, и необходимо настроить останов ОС при переполнении журналов а такде предусмотреть уведомление администратора, то нужно выполнить соответствующую настройку в файле /etc/audit/auditd.conf:

```
space_left_action = email
action_mail_acct = root
admin_space_left_action = halt
```

Листинг 129: Настройка текущей политики при достижении лимита для auditd

Затем перезапустить службу auditd.

8.2.4.6 Аудит изменений времени

Неожиданные изменения значений даты и времени могут быть признаком подозрительной активности. Требуется отслеживать все возможные способы изменения времени. Для этого необходимо записывать сообщение аудита каждый раз при выполнении ядром ОС следующих системных вызовов:

– adjtimex() – осуществляет тонкую подстройку внутреннего счетчика времени ядра ОС Linux;

– settimeofday() – устанавливает внутреннее представление времени в ядре ОС Linux (в том числе собственно значение времени и временной зоны);

– stime() – устанавливает значение времени в семантике UNIX Epoch (с 01 января 1970 года);

– clock_settime( ) – осуществляет различные преобразования значений времени в ядре ОС Linux.

Для проверки имеющейся политики в отношении фиксации изменений времени в файлах /etc/audit.rules.d/*.rules, выполнить:

```
# grep "time-change" /etc/audit/rules.d/*.rules
-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time-change
-a always,exit -F arch=b32 -S adjtimex -S settimeofday -S stime -k time-change
-a always,exit -F arch=b64 -S clock_settime -k time-change
-a always,exit -F arch=b32 -S clock_settime -k time-change
-w /etc/localtime -p wa -k time-change
```

Листинг 130: Проверка наличия политики фиксации событий изменения времени

Для проверки того, активизирована ли данная политика в настоящее время, выполнить:

```
# auditctl -l | grep "time-change"
-a always,exit F arch=b64 -S adjtimex,settimeofday -F key=time-change
-a always,exit F arch=b32 -S stime,settimeofday,adjtimex -F key=time-change
-a always,exit F arch=b64 -S clock_settime -F key=time-change
-a always,exit F arch=b32 -S clock_settime -F key=time-change
-w /etc/localtime -p wa -k time-change
```

Листинг 131: Проверка активизации политики фиксации событий изменения времени

Сверить вывод, и если он отличается, то выполнить настройку. Для этого отредактировать (при отсутствии – создать) файл /etc/audit/rules.d/time-change.rules и внести в него соответствующую политику аудита:

```
-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time-change
-a always,exit -F arch=b32 -S adjtimex -S settimeofday -S stime -k time-change
-a always,exit -F arch=b64 -S clock_settime -k time-change
-a always,exit -F arch=b32 -S clock_settime -k time-change
-w /etc/localtime -p wa -k time-change
```

Листинг 132: Настройка политики фиксации событий изменения времени

Затем перезапустить службу auditd.

8.2.4.7 Аудит изменений пользователей, паролей и групп

Непредвиденные изменения в системных файлах, содержащих информацию о пользователях, группах и файлах паролей, могут быть признаком того, что система скомпрометирована. Или признаком того, что нарушитель пытался скрыть свою активность, повлиять на бюджеты пользователей, групп и/или хранимую аутентификационную информацию. Критически важно отслеживать все подобные изменения. Для этого служба аудита auditd[21] должна быть, настроена на аудит любых изменений в перечисленных ниже файлах:

– /etc/group – общий файл хранения информации о группах;

[21] Аудит изменений должен осуществляться с помощью auditd независимо от использования инструментов контроля целостности AIDE. Также см. раздел 2.6 «Контроль целостности с помощью AIDE».

– /etc/passwd – общий файл хранения информации о пользовательских бюджетах;
– /etc/gshadow – теневой файл хранения информации о группах;
– /etc/shadow – файл хранения аутентификационной информации;
– /etc/security/opasswd – файл хранения предыдущей аутентификационной информации.

Для проверки того, настроена ли служба аудита auditd на отслеживание изменений в указанных файлах, выполнить:

```
# grep "identity" /etc/audit/rules.d/*.rules
-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity
```

Листинг 133: Проверка наличия политики фиксации событий изменения данных субъектов

Для проверки того, активизирована ли данная политика в настоящее время, выполнить:

```
# auditctl -l | grep "identity"
-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity
```

Листинг 134: Проверка задействования политики фиксации событий изменения данных субъектов

Сверить вывод, и если он отличается, то выполнить настройку. Для этого отредактировать (при отсутствии – создать) файл /etc/audit/rules.d/identity.rules и внести в него соответствующую политику аудита:

```
-w /etc/group -p wa -k identity
-w /etc/passwd -p wa -k identity
-w /etc/gshadow -p wa -k identity
-w /etc/shadow -p wa -k identity
-w /etc/security/opasswd -p wa -k identity
```

Листинг 135: Настройка политики фиксации событий изменения данных субъектов

Затем перезапустить службу auditd.

8.2.4.8 Аудит изменений сетевой конфигурации

Требуется отслеживать любые изменения в сетевой конфигурации, а именно:
– системный вызов sethostname( ) – служит для установления имени узла сети;
– системный вызов setdomainname( ) – служит для изменения имени домена;
– изменения в файлах /etc/hosts и /etc/network;
При этом записи аудита тегируются с помощью тега «system-locale».

Для проверки того, настроена ли служба аудита `auditd` на отслеживание изменений в сетевой конфигурации, выполнить:

```
# grep system-locale /etc/audit/rules.d/*.rules
-a always,exit -F arch=b64 -S sethostname -S setdomainname -k system-locale
-a always,exit -F arch=b32 -S sethostname -S setdomainname -k system-locale
-w /etc/hosts -p wa -k system-locale
-w /etc/network -p wa -k system-locale
```

Листинг 136: Проверка наличия политики фиксации изменений данных идентификации узла

Для проверки текущей загруженной конфигурации `auditd` на аудит сетевых изменений, выполнить:

```
# auditctl -l | grep system-locale
-a always,exit -F arch=b64 -S sethostname -S setdomainname -k system-locale
-a always,exit -F arch=b32 -S sethostname -S setdomainname -k system-locale
-w /etc/hosts -p wa -k system-locale
-w /etc/network -p wa -k system-locale
```

Листинг 137: Проверка задействования политики фиксации изменений данных идентификации узла

Сверить вывод, и если он отличается, то выполнить необходимую настройку.

Для этого отредактировать (или создать) файл `/etc/audit/rules.d/system-locale.rules` и внести в него соответствующую политику аудита:

```
-a always,exit -F arch=b64 -S sethostname -S setdomainname -k system-locale
-a always,exit -F arch=b32 -S sethostname -S setdomainname -k system-locale
-w /etc/hosts -p wa -k system-locale
-w /etc/network -p wa -k system-locale
```

Листинг 138: Настройка политики фиксации изменений данных идентификации узла

Затем перезапустить службу `auditd`.

8.2.4.9 Аудит событий входа и выхода

Отслеживание событий входа пользователей в операционную систему и и выхода из нее критически важно для безопасности.

Кроме того, такая настройка помогает отслеживать атаки нарушителя, использующего техники подбора пароля.

В интересах аудита осуществляется отслеживание следующих файлов и событий:

– `/var/log/faillog` – отслеживаются события неудачного входа;

– `/var/log/lastlog` – отслеживаются события удачного входа;

– `/var/log/tallylog` – отслеживаются события отказа во входе (по времени и другим ограничениям) с помощью модуля PAM (`pam_tally2`).

Для проверки наличия текущих правил политики аудита, отслеживающих события входа и выхода, выполнить поиск правил в каталоге `/etc/audit/rules.d/*.rules`:

```
# grep logins /etc/audit/rules.d/*.rules
-w /var/log/faillog -p wa -k logins
-w /var/log/lastlog -p wa -k logins
-w /var/log/tallylog -p wa -k logins
```

Листинг 139: Проверка наличия политики фиксации событий входа и выхода пользователей

Для проверки текущей конфигурации правил службы аудита `auditd`, выполнить:

```
# auditctl -l | grep logins
-w /var/log/faillog -p wa -k logins
-w /var/log/lastlog -p wa -k logins
-w /var/log/tallylog -p wa -k logins
```

Листинг 140: Проверка задействования политики фиксации событий входа и выхода пользователей

Сверить вывод, и если он отличается, то выполнить настройку. Для этого отредактировать (при отсутствии – создать) файл `/etc/audit/rules.d/logins.rules` и внести в него соответствующую политику аудита:

```
-w /var/log/faillog -p wa -k logins
-w /var/log/lastlog -p wa -k logins
-w /var/log/tallylog -p wa -k logins
```

Листинг 141: Настройка политики фиксации событий входа и выхода пользователей

Затем перезапустить службу `auditd`.

```
# systemctl restart auditd
```

8.2.4.10 Аудит получения сессии

В защищенной системе требуется отслеживать события получения пользовательской сессии. Изменения в указанных ниже файлах могут свидетельствовать о подозрительной активности, такой как попытки получения сессии в необычные часы, и т.п. Указанные ниже параметры настройки отслеживают изменения в следующих файлах:

– `/var/run/utmp` – содержит информацию о пользователях, сеанс которых выполняется в данный момент;

– `/var/log/wtmp` – содержит информацию о событиях входа/выхода пользователей, а также о событиях останова и перезагрузки операционной системы;

– `/var/log/btmp` – содержит информацию о неудачных попытках входа.

Этот файл может быть просмотрен с помощью команды `/usr/bin/last -f /var/log/btmp`. Для просмотра `/var/log/wtmp` команда `last` используется без параметров. Для просмотра `/var/log/utmp` используется `last -f /var/log/utmp`.

Для удобства поиска событий в этом разделе применена конфигурация, которая помечает события получения/отказа сессии меткой (тегом) «`logins`».

Для проверки наличия текущих правил политики аудита, отслеживающих события получения сессии, выполнить поиск правил в каталоге `/etc/audit/rules.d/*.rules`:

```
# grep -E "(session|logins)" /etc/audit/rules.d/*.rules
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k logins
-w /var/log/btmp -p wa -k logins
```

Листинг 142: Проверка наличия политики фиксации событий получения сессии

Для проверки применимости правил службой аудита `auditd`, выполнить:

```
# auditctl -l | grep -E "(session|logins)"
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k logins
-w /var/log/btmp -p wa -k logins
```

Листинг 143: Проверка задействования политики фиксации событий получения сессии

Сверить вывод, и если он отличается, то выполнить настройку. Для этого отредактировать (при отсутствии – создать) файл `/etc/audit/rules.d/session.rules` и внести в него соответствующую политику аудита:

```
-w /var/run/utmp -p wa -k session
-w /var/log/wtmp -p wa -k logins
-w /var/log/btmp -p wa -k logins
```

Листинг 144: Настройка политики фиксации событий получения сессии

Затем перезапустить службу `auditd`.

8.2.4.11 Аудит изменений файловых атрибутов

В составе ядра ОС Linux содержится механизм дискреционного разграничения доступа (DAC, Discretional Access Control). Системные вызовы ядра ОС, отвечающие за работу с функциями DAC в ядре способны с помощью перехватчика функций LSM (Linux Security Module) отслеживать выполнение функций т.н. «монитора обращений» DAC при совершении операций с объектами доступа (файлами и папками), такими как чтение файла (`r` – поток данных от объекта к субъекту), запись файла (`w` – поток данных от субъекта к объекту), выполнение (`x`) файла, изменение атрибутов объекта и т.п.

В применяемой ниже политике предусматривается отслеживание выполнения следующих системных вызовов:

– `chmod()`, `fchmod()` и `fchmodat()` – системные вызовы, выполняющие взаимодействия с функциями ядра ОС, отвечающими за изменение дискреционных прав объектов доступа (битов чтения, записи и выполнения);

– `chown()`, `fchown()`, `fchownat()` и `lchown()` – системные вызовы, выполняющие взаимодействия с функциями ядра ОС, отвечающими за изменение владельца и группы объектов доступа (`owner`, `group`, `others`);

– setxattr( ), lsetxattr( ), fsetxattr( ) – системные вызовы, выполняющие взаимодействия с функциями ядра ОС, отвечающими за установку атрибутов объекта (стандарт POSIX.1e: не изменяемый, архивный, только для чтения и т.п.);

– removexattr( ), lremovexattr( ), fremovexattr( ) – системные вызовы, выполняющие взаимодействия с функциями ядра ОС, отвечающими за удаление атрибутов объекта (стандарт POSIX.1e: не изменяемый, архивный, только для чтения и т.п.). Требуется учитывать, что в описанной ниже настройке отслеживаются обращения только для субъектов с идентификатором (UID) больше или равно 1000. Все записи аудита, описанные в этом пункте помечаются меткой (тегом) «perm_mod» для удобства поиска.

Для проверки наличия текущих правил политики аудита, отслеживающих события изменения атрибутов, выполнить поиск правил в каталоге /etc/audit/rules.d/*.rules:

```
# grep perm_mod /etc/audit/rules.d/*.rules
-a always,exit -F arch=b64 -S chmod -S fchmod -S fchmodat -F auid⩾1000 -F auid≠4294967295 -k
  perm_mod
-a always,exit -F arch=b32 -S chmod -S fchmod -S fchmodat -F auid⩾1000 -F auid≠4294967295 -k
  perm_mod
-a always,exit -F arch=b64 -S chown -S fchown -S fchownat -S lchown -F auid⩾1000 -F auid
  ≠4294967295 -k perm_mod
-a always,exit -F arch=b32 -S chown -S fchown -S fchownat -S lchown -F auid⩾1000 -F auid
  ≠4294967295 -k perm_mod -k perm_mod
-a always,exit -F arch=b64 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr -S
  fremovexattr -F auid⩾1000 -F auid≠4294967295
-a always,exit -F arch=b32 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr -S
  fremovexattr -F auid⩾1000 -F auid≠4294967295 -k perm_mod
```

Листинг 145: Проверка наличия политики фиксации изменений атрибутов файлов

Проверить текущее состояние загруженных правил политики можно так:

```
#auditctl -l | grep auditctl -l | grep perm_mod
-a always,exit -F arch=b64 -S chmod,fchmod,fchmodat -F auid⩾1000 -F auid≠-1 -F key=perm_mod
-a always,exit -F arch=b32 -S chmod,fchmod,fchmodat -F auid⩾1000 -F auid≠-1 -F key=perm_mod
-a always,exit -F arch=b64 -S chown,fchown,lchown,fchownat -F auid⩾1000 -F auid≠-1 -F key=
  perm_mod
-a always,exit -F arch=b32 -S lchown,fchown,chown,fchownat -F auid⩾1000 -F auid≠-1 -F key=
  perm_mod
-a always,exit -F arch=b64 -S setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F
   auid⩾1000 -F auid≠-1 -F key=perm_mod
-a always,exit -F arch=b32 -S setxattr,lsetxattr,fsetxattr,removexattr,lremovexattr,fremovexattr -F
   auid⩾1000 -F auid≠-1 -F key=perm_mod
```

Листинг 146: Проверка задействования политики фиксации изменений атрибутов файлов

Сверить вывод, и если он отличается, то выполнить настройку.

Для этого отредактировать (при отсутствии – создать) файл /etc/audit/rules.d/perm_mod.rules и внести в него соответствующую политику аудита:

```
-a always,exit -F arch=b64 -S chmod -S fchmod -S fchmodat -F auid⩾1000 -F auid≠4294967295 -k
  perm_mod
-a always,exit -F arch=b32 -S chmod -S fchmod -S fchmodat -F auid⩾1000 -F auid≠4294967295 -k
  perm_mod
```

```
-a always,exit -F arch=b64 -S chown -S fchown -S fchownat -S lchown -F auid⩾1000 -F auid
  ≠4294967295 -k perm_mod
-a always,exit -F arch=b32 -S chown -S fchown -S fchownat -S lchown -F auid⩾1000 -F auid
  ≠4294967295 -k perm_mod
-a always,exit -F arch=b64 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr -S
  fremovexattr -F auid⩾1000 -F auid≠4294967295 -k perm_mod
-a always,exit -F arch=b32 -S setxattr -S lsetxattr -S fsetxattr -S removexattr -S lremovexattr -S
  fremovexattr -F auid⩾1000 -F auid≠4294967295 -k perm_mod
```

Листинг 147: Настройка политики фиксации изменений атрибутов файлов

Затем перезапустить службу `auditd`.

8.2.4.12 Аудит отказа при обращении к файлу (папке)

В защищенной системе требуется, как минимум, отслеживать отказы при попытках обращений к защищаемой информации.

Например, можно фиксировать попытки создания файла или папки, попытки обращения к ним, либо попытки добавить (изменить) данные в файле или папке.

Для этого предусматривается создание правил, ориентированных на отслеживание следующих системных вызовов ядра ОС Linux:

– `creat()` – системный вызов, осуществляющий взаимодействие с функциями ядра ОС Linux, использующимися при создании объекта доступа;

– `open()`, `openat()` – системные вызовы, осуществляющие взаимодействие с функциями ядра ОС Linux, использующимися при обращении к объекту доступа;

– `truncate()`, `ftruncate()` – системные вызовы, осуществляющие взаимодействие с функциями ядра ОС Linux, использующимися при изменении (или добавлении информации) к объекту доступа.

При этом описанная ниже политика предусматривает отслеживание только тех обращений DAC, по которым было принято решение об отказе.

То есть в тех случаях, когда системный вызов вернул EACCES (отказ в предоставлении доступа, ввиду явного несоответствия атрибутов субъекта и объекта) и EPERM (любые другие случаи отказа предоставления доступа к объекту).

Требуется учитывать, что в описанной ниже настройке отслеживаются обращения только для субъектов с идентификатором (UID) больше или равно `1000`. Все записи аудита, описанные в этом пункте помечаются меткой (тегом) «`access`» для удобства поиска.

Для проверки текущих правил политики аудита, отслеживающих события отказа при обращении к объектам, выполнить поиск правил в каталоге `/etc/audit/rules.d/*.rules`:

```
# grep access /etc/audit/rules.d/*.rules
-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EACCES -F
  auid⩾1000 -F auid =4294967295 -k access
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EACCES -F
  auid⩾1000 -F auid≠4294967295 -k access
```

```
-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EPERM -F
  auid⩾1000 -F auid≠4294967295 -k access
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EPERM -F
  auid⩾1000 -F auid≠4294967295 -k access
```

Листинг 148: Проверка наличия политики фиксации отказов обращений к файлу

Или:

```
# auditctl -l | grep access
-a always,exit -F arch=b64 -S open,truncate,ftruncate,creat,openat EACCES -F auid⩾1000 -F auid≠-1
   -F key=access
-a always,exit -F arch=b32 -S open,creat,truncate,ftruncate,openat EACCES -F auid⩾1000 -F auid≠-1
   -F key=access
-a always,exit -F arch=b64 -S open,truncate,ftruncate,creat,openat EPERM -F auid⩾1000 -F auid≠-1
  -F key=access
-a always,exit -F arch=b32 -S open,creat,truncate,ftruncate,openat EPERM -F auid⩾1000 -F auid≠-1
  -F key=access
```

Листинг 149: Проверка задействования политики фиксации отказов обращений к файлу

Сверить вывод, и если он отличается, то выполнить настройку. Для этого отредактировать (при отсутствии – создать) файл `/etc/audit/rules.d/access.rules` и внести в него соответствующую политику аудита:

```
-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EACCES -F
  auid⩾1000 -F auid≠4294967295 -k access
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EACCES -F
  auid⩾1000 -F auid≠4294967295 -k access
-a always,exit -F arch=b64 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EPERM -F
  auid⩾1000 -F auid≠4294967295 -k access
-a always,exit -F arch=b32 -S creat -S open -S openat -S truncate -S ftruncate -F exit=-EPERM -F
  auid⩾1000 -F auid≠4294967295 -k access
```

Листинг 150: Настройка политики фиксации отказов при обращении к файлу

Затем перезапустить службу `auditd`.

8.2.4.13 Аудит выполнения привилегированных команд

В защищенной системе критически важно отслеживать выполнение системных команд, требующих привилегированого доступа. Под такими командами здесь понимаются команды с установленным битом смены идентификатора суперпользователя или полномочной группы (биты SUID, SGID). Требуется учитывать, что в описанной ниже настройке отслеживаются обращения только для субъектов с идентификатором (UID) больше или равно 1000. Все записи аудита, описанные в этом пункте помечаются меткой (тегом) «`priveleged`» для удобства поиска. Для проверки наличия текущих правил политики аудита, отслеживающих события выполнения привилегированных команд, выполнить поиск правил в каталоге `/etc/audit/rules.d/*.rules` (где «раздел ФС» – обозначение файловой системы, в примере используется корневая ФС «/»):

```
# find Точка< монтирования> -xdev \( -perm -4000 -o -perm -2000 \) -type f | awk '{print "-a always
  ,exit -F path=" $1 " -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged" }'
-a always,exit -F path=/usr/bin/passwd -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/gpasswd -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/mlocate -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/chfn -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/umount -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/fusermount -F perm=x -F auid>=1000 -F auid!=4294967295 -k
  privileged
-a always,exit -F path=/usr/bin/pkexec -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/dotlockfile -F perm=x -F auid>=1000 -F auid!=4294967295 -k
  privileged
-a always,exit -F path=/usr/bin/sudo -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/ssh-agent -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/chage -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/expiry -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/newgrp -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/chsh -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/bsd-write -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/mount -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/crontab -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/bin/wall -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/sbin/unix_chkpwd -F perm=x -F auid>=1000 -F auid!=4294967295 -k
  privileged
-a always,exit -F path=/usr/sbin/pam_extrausers_chkpwd -F perm=x -F auid>=1000 -F auid!=4294967295
  -k privileged
-a always,exit -F path=/usr/sbin/pppd -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/sbin/postqueue -F perm=x -F auid>=1000 -F auid!=4294967295 -k
  privileged
-a always,exit -F path=/usr/sbin/postdrop -F perm=x -F auid>=1000 -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/lib/policykit-1/polkit-agent-helper-1 -F perm=x -F auid>=1000 -F auid
  !=4294967295 -k privileged
-a always,exit -F path=/usr/lib/x86_64-linux-gnu/utempter/utempter -F perm=x -F auid>=1000 -F auid
  !=4294967295 -k privileged
-a always,exit -F path=/usr/lib/cupsPPD/prlinuxcupsppd -F perm=x -F auid>=1000 -F auid!=4294967295
  -k privileged
-a always,exit -F path=/usr/lib/snapd/snap-confine -F perm=x -F auid>=1000 -F auid!=4294967295 -k
  privileged
-a always,exit -F path=/usr/lib/mc/cons.saver -F perm=x -F auid>=1000 -F auid!=4294967295 -k
  privileged
-a always,exit -F path=/usr/lib/openssh/ssh-keysign -F perm=x -F auid>=1000 -F auid!=4294967295 -k
  privileged
-a always,exit -F path=/usr/lib/spice-gtk/spice-client-glib-usb-acl-helper -F perm=x -F auid>=1000
  -F auid!=4294967295 -k privileged
-a always,exit -F path=/usr/lib/dbus-1.0/dbus-daemon-launch-helper -F perm=x -F auid>=1000 -F auid
  !=4294967295 -k privileged
-a always,exit -F path=/usr/lib/eject/dmcrypt-get-device -F perm=x -F auid>=1000 -F auid
  !=4294967295 -k privileged
-a always,exit -F path=/usr/lib/xorg/Xorg.wrap -F perm=x -F auid>=1000 -F auid!=4294967295 -k
  privileged
-a always,exit -F path=/usr/libexec/camel-lock-helper-1.2 -F perm=x -F auid>=1000 -F auid
  !=4294967295 -k privileged
```

Листинг 151: Пример поиска и формирования правил аудита при запуске файлов с битом SUID

Если вывода нет или отличается, то требуется задать соответствующее правило. Для этого отредактировать (при отсутствии – создать) файл `/etc/audit/rules.d/priveleged.rules` и внести в него соответствующую политику аудита:

```
# find / -xdev \( -perm -4000 -o -perm -2000 \) -type f | awk '{print "-a always,exit -F path=" $1
  " -F perm=x -F auid≥'"$(awk '/^\s*UID_MIN/{print $2}' /etc/login.defs)"' -F auid≠4294967295 -k
  privileged" }' » /etc/audit/rules.d/priveleged.rules
```

Листинг 152: Пример формирования политики аудита для фиксации запуска файлов с битом SUID

Затем перезапустить службу `auditd`.

8.2.4.14 Проверка аудита операций монтирования

В системе отслеживаются события подключения (монтирования) и отключения (размонтирования) файловых систем.

Для этого аудит настроен на фиксацию выполнения системных вызовов `mount()` и `umount()`, которые взаимодействуют с соответствующими функциями ядра ОС, отвечающими за подключение и отключение файловых систем.

Требуется учитывать, что в описанной ниже настройке отслеживаются обращения только для субъектов с идентификатором (UID) больше или равно `1000`.

Все записи аудита, описанные в этом пункте помечаются меткой (тегом) «`mounts`» для удобства поиска. Для проверки текущих правил политики аудита, отслеживающих события монтирования/размонтирования файловых систем, выполнить поиск правил в каталоге `/etc/audit/rules.d/*.rules`:

```
# grep mounts /etc/audit/rules.d/*.rules
-a always,exit -F arch=b64 -S mount -F auid≥1000 -F auid≠4294967295 -k mounts
-a always,exit -F arch=b32 -S mount -F auid≥1000 -F auid≠4294967295 -k mounts
```

Листинг 153: Проверка наличия политики аудита операций монтирования

Или:

```
# auditctl -l | grep mounts
-a always,exit -F arch=b64 -S mount -F auid≥1000 -F auid≠-1 -F key=mounts
-a always,exit -F arch=b32 -S mount -F auid≥1000 -F auid≠-1 -F key=mounts
```

Листинг 154: Проверка активизации политики аудита операций монтирования

Сверить вывод, и если он отличается, то выполнить настройку политики аудита. Для этого отредактировать (при отсутствии – создать) файл `/etc/audit/rules.d/mounts.rules` и внести в него соответствующую политику аудита:

```
-a always,exit -F arch=b64 -S mount -F auid≥1000 -F auid≠4294967295 -k mounts
-a always,exit -F arch=b32 -S mount -F auid≥1000 -F auid≠4294967295 -k mounts
```

Листинг 155: Настройка политики аудита операций монтирования

Затем перезапустить службу `auditd`.

8.2.4.15 Проверка аудита при переключении контекста

В системе отслеживаются операции переключения контекста, а также операции, связанные с изменением контекста пользовательских полномочий. Для этого описанная ниже политика предполагает осуществлять надзор над:

– файлом `/etc/sudoers`, сигнализируя о возможных операциях по изменению контекста ролей пользователей;

– операциями смена контекста, учитывая даже временную смену, путем применения механизма `sudo`.

Для проверки текущих правил политики аудита, отслеживающих операции по изменению контекста ролей (изменения в `/etc/sudoers`), выполнить поиск правил в каталоге `/etc/audit/rules.d/*.rules`:

```
# grep scope /etc/audit/rules.d/*.rules
-w /etc/sudoers -p wa -k scope
-w /etc/sudoers.d/ -p wa -k scope
```

Листинг 156: Проверка регистрации событий при изменении контекста пользоваталя. Вариант 1

Или:

```
# auditctl -l | grep scope
-w /etc/sudoers.d/ -p wa -k scope
-w /etc/sudoers -p wa -k scope
```

Листинг 157: Проверка регистрации событий при изменении контекста пользоваталя. Вариант 2

Для проверки текущих правил политики аудита, отслеживающих операции по изменению контекста пользователя (включая изменения с помощью `sudo`), выполнить поиск правил в каталоге `/etc/audit/rules.d/*.rules`:

```
# auditctl -l | grep actions
-a always,exit -F arch=b64 -S execve -C uid≠euid -F euid=0 -F auid⩾1000 -F auid≠-1 -F key=
  actions
-a always,exit -F arch=b32 -S execve -C uid≠euid -F euid=0 -F auid⩾1000 -F auid≠-1 -F key=
  actions
```

Листинг 158: Проверка активности политики аудита при переключении контекста

Сверить вывод, и если он отличается, то выполнить настройку. Для этого отредактировать (при отсутствии – создать) файл `/etc/audit/rules.d/actions.rules` и внести в него соответствующую политику аудита:

```
-a exit,always -F arch=b64 -C euid≠uid -F euid=0 -Fauid⩾1000 -F auid≠4294967295 -S execve -k
  actions
-a exit,always -F arch=b32 -C euid≠uid -F euid=0 -Fauid⩾1000 -F auid≠4294967295 -S execve -k
  actions
```

Листинг 159: Настройка политики аудита при переключении контекста пользователя

Сверить вывод, и если он отличается, то выполнить настройку. Для этого отредактировать (при отсутствии – создать) файл `/etc/audit/rules.d/scope.rules` и внести в него соответствующую политику аудита:

```
-w /etc/sudoers -p wa -k scope
-w /etc/sudoers.d/ -p wa -k scope
```

Листинг 160: Настройка регистрации событий при изменениях в файлах конфигурации контекста

Затем перезапустить службу `auditd`.

8.2.4.16 Проверка аудита операций с модулями ядра

В системе отслеживаются все операции с модулями ядра ОС. Для этого отслеживается использование следующих системных вызовов:

– `init_module( )` – связывает (добавляет) модуль с ядром;

– `delete_module( )` – удаляет модуль из ядра;

Также отслеживается выполнение соответствующих ИФБО (`insmod`, `rmmod`, `modprobe`).

Для проверки текущих правил политики аудита, отслеживающих операции по взаимодействию с модулями ядра, выполнить поиск правил в каталоге `/etc/audit/rules.d/*.rules`:

```
# grep modules /etc/audit/rules.d/*.rules
-w /sbin/insmod -p x -k modules
-w /sbin/rmmod -p x -k modules
-w /sbin/modprobe -p x -k modules
-a always,exit -F arch=b64 -S init_module -S delete_module -k modules
```

Листинг 161: Проверка регистрации событий при загрузке(выгрузке) модуля ядра

Или:

```
# auditctl -l | grep modules
-w /sbin/insmod -p x -k modules
-w /sbin/rmmod -p x -k modules
-w /sbin/modprobe -p x -k modules
-a always,exit -F arch=b64 -S init_module,delete_module -F key=modules
```

Листинг 162: Проверка активизации аудита при загрузке(выгрузке) модуля ядра

Сверить вывод, и если он отличается, то выполнить настройку. Для этого отредактировать (при отсутствии – создать) файл `/etc/audit/rules.d/modules.rules` и внести в него соответствующую политику аудита:

```
-w /sbin/insmod -p x -k modules
-w /sbin/rmmod -p x -k modules
-w /sbin/modprobe -p x -k modules
-a always,exit -F arch=b64 -S init_module -S delete_module -k modules
```

Листинг 163: Настройка регистрации событий при загрузке(выгрузке) модуля ядра

Затем перезапустить службу `auditd`.

8.2.4.17 Проверка неизменности конфигурации аудита

Чтобы нарушитель не смог оказать воздействие на текущие выполняемые правила аудита, например, с помощью `auditctl`, в системе реализована конфигурация, при которой все описанные выше политики аудита переведены в состояние неизменности, (соответствующие файлы политик регистрации событий снабжены атрибутом `immutable`).

Для проверки текущей конфигурации неизменности правил аудита выполнить:

```
# grep "^\s*[^#]" /etc/audit/rules.d/99-finalize.rules | tail -1
-e 2
```

Листинг 164: Проверка неизменности конфигурации аудита

Если вывод отличается от приведенного выше, то требуется выполнить команду:

```
# echo "-e 2" > /etc/audit/rules.d/99-finalize.rules
```

Листинг 165: Настройка неизменности конфигурации аудита

Затем перезапустить службу `auditd`.

8.2.4.18 Интерпретация сообщений аудита `auditd`

Для примера выполнить запрос файла паролей от имени пользователя и посмотреть получившееся сообщение аудита, и получить вывод аналогичный приведенному:

```
$ id
uid=500(freshuser) gid=500(freshuser) группы=500(freshuser),10(wheel),100(users),430(sambashare)
$ cat /etc/shadow
cat: /etc/shadow: Отказано в доступе
```

Листинг 166: Пример запроса файла паролей от имени пользователя

Затем от имени `root` можно поискать информацию об этом обращении и посмотреть конкретное сообщение аудита:

```
# ausearch -l | grep /etc/shadow
type=EXECVE msg=audit(1659951987.026:8962): argc=2 a0="cat" a1="/etc/shadow"
```

Листинг 167: Пример сообщения в журнале аудита

Как видно из примера, есть сообщение с номером (после двоеточия) 8962. Можно посмотреть еще подробнее:

```
# ausearch -a 8962
----
time->Mon Aug  8 12:46:27 2022
type=PROCTITLE msg=audit(1659951987.026:8962): proctitle=636174002F6574632F736861646F77
type=PATH msg=audit(1659951987.026:8962): item=1 name="/lib64/ld-linux-x86-64.so.2" inode=81 dev
  =08:02 mode=0100755 ouid=0 ogid=0 rdev=00:00 obj=unlabeled nametype=NORMAL cap_fp=0 cap_fi=0
  cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(1659951987.026:8962): item=0 name="/bin/cat" inode=523327 dev=08:02 mode
  =0100755 ouid=0 ogid=0 rdev=00:00 obj=unlabeled nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0
  cap_fver=0 cap_frootid=0
type=CWD msg=audit(1659951987.026:8962): cwd="/home/freshuser"
```

```
type=EXECVE msg=audit(1659951987.026:8962): argc=2 a0="cat" a1="/etc/shadow"
type=SYSCALL msg=audit(1659951987.026:8962): arch=c000003e syscall=59 success=yes exit=0 a0=24f43b0
   a1=2506830 a2=235b460 a3=7fcfd91328e0 items=2 ppid=3485 pid=5415 auid=4294967295 uid=500 gid=500
   euid=500 suid=500 fsuid=500 egid=500 sgid=500 fsgid=500 tty=pts0 ses=4294967295 comm="cat" exe
  ="/bin/coreutils" subj=kernel key="FRU_PRS"
```

Листинг 168: Пример подробного вывода сообщения (здесь – № 8962) в журнале аудита

Как видно из примера, данное обращение вызвало несколько записей аудита. Можно рассмотреть их подробнее и попробовать расшифровать.

Поле тип (`type`) - тип события. В данном примере тип первого события указан как SYSCALL, т.е. тип события - системный вызов ядра операционной системы.

Поле CWD - это тип события, созданный по информации, полученной из текущего системного окружения процесса, и указывает на текущий рабочий каталог, откуда была вызвана программа или процесс вызвавший системный вызов, указанный в событии типа SYSCALL.

Поле - PATH - связано с файлом, по отношению к которому было создано событие аудита.

Особый тип поля - PROCTITLE. Назначение этого поля в том, чтобы фиксировать выполнявшуюся команду. В примере сообщения аудита `proctitle=636174002F6574632F736861646F77` зафиксировано шестнадцатеричное (HEX) значение. Которое может быть преобразовано в текстовое значение. Для этого можно использовать как онлайн-переводчик https://onlinehextools.com/convert-hex-to-text, так и вывод команды `ausearch` с аргументом `-i`, т.е. `ausearch -i`:

```
# ausearch -a 8962 -i
----
type=PROCTITLE msg=audit(08.08.2022 12:46:27.026:8962) : proctitle=cat /etc/shadow
type=PATH msg=audit(08.08.2022 12:46:27.026:8962) : item=1 name=/lib64/ld-linux-x86-64.so.2 inode
  =81 dev=08:02 mode=file,755 ouid=root ogid=root rdev=00:00 obj=unlabeled nametype=NORMAL cap_fp=
  none cap_fi=none cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(08.08.2022 12:46:27.026:8962) : item=0 name=/bin/cat inode=523327 dev=08:02
  mode=file,755 ouid=root ogid=root rdev=00:00 obj=unlabeled nametype=NORMAL cap_fp=none cap_fi=
  none cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(08.08.2022 12:46:27.026:8962) : cwd=/home/freshuser
type=EXECVE msg=audit(08.08.2022 12:46:27.026:8962) : argc=2 a0=cat a1=/etc/shadow
type=SYSCALL msg=audit(08.08.2022 12:46:27.026:8962) : arch=x86_64 syscall=execve success=yes exit
  =0 a0=0x24f43b0 a1=0x2506830 a2=0x235b460 a3=0x7fcfd91328e0 items=2 ppid=3485 pid=5415 auid=unset
   uid=freshuser gid=freshuser euid=freshuser suid=freshuser fsuid=freshuser egid=freshuser sgid=
  freshuser fsgid=freshuser tty=pts0 ses=unset comm=cat exe=/bin/coreutils subj=kernel key=FRU_PRS
```

Листинг 169: Пример подробного вывода сообщения (здесь – № 8962) в журнале аудита с расшифровкой

Поле счетчика события (`msg`). Данное поле заключается в скобки, и является уникальным идентификатором события. В примере это (1659951987.026:8962). Каждый идентификатор состоит из двух полей, разделенных двоеточием. Где слева от двоеточия располагается штамп времени, в формате UNIX Epoch Time (от 01.01.1970 года), в секундах. Данное поле позволяет интерпретировать время в любой удобный формат счисления времени. Число после двоеточия - представляет собой просто уникальный произвольный возрастающий идентификатор, позволяющий как отличать события друг от друга, так и находить записи аудита, ассоциированные с одним и тем же событием.

Уже упоминавшаяся команда `ausearch -i` позволяет интерпретировать числовое значение в читаемый формат с датой, временем и номером (идентификатором) события.

Поле архитектуры (arch) - позволяет идентифицировать аппаратную архитектуру (архитектуру процессора). Команда ausearch -i позволяет интерпретировать архитектуру процессора в более удобный вид, например архитектура arch=c000003e будет представлена как arch=x86_64.

Поле системного вызова (syscall) - позволяет отследить определенный системный вызов, с которым связано событие.

В примере - syscall=59. При этом номер конкретного системного вызова будет извлечен из системного файла /usr/include/asm/unistd.h (либо, в зависимости от того, 32-х битный вызов или 64-х битный из файлов /usr/include/asm/unistd_32.h и /usr/include/asm/unistd_64.h соответственно). Номер системного вызова 59 соответствует системному вызову execve( ) в указанном примере.

Поле результата операции (success) - позволяет идентифицировать результат операции, и установить успехом или неудачей закончился системный вызов, что в общем случае позволяет отследить результат.

Поле выходных данных системного вызова (exit) - позволяет установить выходное значение данных для системного вызова. В данном случае (3) - это возвращение значения блокировки файла.

Поля входных данных системного вызова в числовом формате (от a0 до a3, в примере aa0=24f43b0 a1=2506830 a2=235b460 a3=7fcfd91328e0) - носят служебный характер и при необходимости позволяют отслеживать начальный адрес имени пути (a0), a1 - значение в шестнадцатеричном формате (HEX), которое при преобразовании в десятичный формат позволяет идентифицировать формат входных данных для заданного системного вызова, поле a2 - режим работы системного вызова, как и поле a3. Эти поля могут быть специфичны, в зависимости от конкретного системного вызова.

Поле строк (items) - позволяет отследить количество строк, переданных как аргумент для команды.

Поле идентификатора родительского процесса (ppid).

Поле с номером анализируемого процесса (pid).

Поле первичного идентификатора субъекта доступа (auid), то есть - пользователя, от имени которого был вызван интерактивный сеанс, приведший к возникновению события. Это поле может отличаться от непосредственного инициатора события, в зависимости от того, от чьего имени выполнялся первичный вход в сеанс.

Поле непосредственного идентификатора субъекта доступа (uid), то есть - пользователя, от чьего имени непосредственно было выполнено действие.

Поле непосредственного идентификатора группы субъекта доступа (gid), то есть - группы пользователя, от чьего имени непосредственно было выполнено действие.

Поля непосредственного эффективного идентификатора субъекта доступа (euid, suid, fsuid), а именно - пользователя, от чьего эффективного имени непосредственно было выполнено действие.

Поля непосредственного эффективного идентификатора группы субъекта доступа (egid, sgid, fsgid), а именно - эффективной группы пользователя, от чьего имени непосредственно было выполнено действие.

Поле с номером устройства терминала (`tty`), на котором произведено событие (операция, вызов).

Поле идентификатора сессии (`ses`). Данный атрибут процесса устанавливается каждый раз при входе любого пользователя в интерактивный сеанс и позволяет производить ассоциацию (прослеживание) любого процесса, вызванного пользователем к его интерактивному сеансу.

Поле команды (`comm`) - позволяет отследить имя команды (приложения) под которым команда (приложение) будет отображена в списке (перечне) процессов (`ps`, `top` и т.п.)

Поле вызова (`exe`) - позволяет отследить полный путь вызова команды (сценария, приложения и т.п.).

Поле ключа аудита (`key`) - позволяет ассоциировать событие с конкретным правилом аудита (при задании ключа) в файле конфигурации политики аудита.

Поле PATH содержит текущий рабочий каталог процесса, по отношению к которому создано сообщение аудита.

Поле имени объекта (`name`): определяет идентификатор объекта доступа (файла), по отношению к которому выполнен системный вызов.

Поле метаданных файла (`inode`) - позволяет проследить имя файла к номеру иноды.

Поле режима битовой маски (`mode`) - позволяет определить текущую битовую маску прав доступа файла.

Поля владельца и группы файла (`ouid` и `ogid`) - позволяют установить владельца и группу файла.

Поле файла устройства (`rdev`) позволяет ассоциировать объект доступа с файлом устройства, независимо, с символьным или блочным. В том случае, если объект доступа является обычным файлом - данное поле не заполняется.

Подробная информация, облегчающая интерпретацию сообщений аудита, а также их детализация по типам приведена ниже, в Таблице 12.

Тип записи	Расшифровка
ADD_GROUP	Операция создания группы
ADD_USER	Операция создания пользователя
ANOM_LOGIN_FAILURES	Превышено количество попыток входа пользователя
ANOM_LOGIN_SESSIONS	Превышено количество одновременных сеансов
ANOM_LOGIN_TIME	Попытка входа осуществлялась в запрещенное время
ANOM_PROMISCUOUS	Изменение режима прослушивания сетевого устройства
AVC	Обращение к функций SELinux MAC
CONFIG_CHANGE	Модификация конфигурации правил аудита
CRED_ACQ	Запрос на получение пользовательских реквизитов входа
CRED_DISP	Получение пользовательских реквизитов входа
CRED_REFR	Актуализация пользовательских реквизитов входа
CWD	Текущего каталог, откуда совершен вызов программы
DAEMON_CONFIG	Изменение конфигурации службы аудита
DAEMON_END	Успешный останов службы аудита
DAEMON_ROTATE	Ротация журнала `auditd`
DAEMON_START	Успешный запуск службы аудита
DEL_GROUP	Удаление группы
DEL_USER	Удаление пользователя
EXECVE	Выполнение системного вызова execve()
INTEGRITY_DATA	Событие контроля целостности в ФС
INTEGRITY_HASH	Обработка контрольной суммы объекта
INTEGRITY_METADATA	Обработка контрольной суммы атрибутов объекта

Тип записи	Расшифровка
INTEGRITY_RULE	Изменение политики КЦ
INTEGRITY_STATUS	Проверка статуса КЦ
LOGIN	Событие входа пользователя
MAC_CIPSOV4_ADD	Создано правило фильтрации с учетом метки безопасности SELinux
MAC_CIPSOV4_DEL	Удалено правило фильтрации с учетом метки безопасности SELinux
MAC_CONFIG_CHANGE	Изменение политики SELinux MAC
MAC_POLICY_LOAD	Загрузка политики SELinux MAC
NETFILTER_CFG	Изменение политики фильтра пакетов
PATH	Путь к имени файла
PROCTITLE	Полнотекстовая запись команды
ROLE_ASSIGN	Назначение роли
ROLE_REMOVE	Удаление роли
SERVICE_START	Запуск службы
SERVICE_STOP	Останов службы
SOCKADDR	Запись для адреса сокета
SOFTWARE_UPDATE	Изменение состава ПО
SYSCALL	Выполнение системного вызова
SYSTEM_BOOT	Загрузка ОС
SYSTEM_RUNLEVEL	Изменение уровня выполнения
SYSTEM_SHUTDOWN	Останов ОС
USER_ACCT	Изменение данных пользователя
USER_AUTH	Попытка аутентификации пользователя
USER_AVC	Обращение к функции SELinux MAC (польз.)
USER_CHAUTHTOK	Модификация бюджета пользователя
USER_CMD	Вызов оболочки (польз.)
USER_END	Завершение сеанса пользователя
USER_LOGIN	Вход пользователя
USER_MAC_POLICY_LOAD	Загрузка политики SELinux (польз.)
USER_ROLE_CHANGE	Изменение роли пользователя

Таблица 12: Наиболее распространенные типы в сообщениях аудита .

Таким образом, если нужно найти подробную информацию о добавлении пользователя, то можно просто сразу вызвать поиск сообщений по типу ADD_USER:

```
# ausearch -m ADD_USER
----
type=ADD_USER msg=audit(30.07.2022 18:07:23.050:1845) : pid=7166 uid=root auid=freshuser ses=3 subj
  =kernel msg='op=adding user id=unknown(501) exe=/usr/sbin/useradd hostname=rosafresh12.tiger.
  kingdom addr=? terminal=tty2 res=success'
----
type=ADD_USER msg=audit(30.07.2022 18:07:23.154:1850) : pid=7166 uid=root auid=freshuser ses=3 subj
  =kernel msg='op=adding home directory id=newuser exe=/usr/sbin/useradd hostname=rosafresh12.tiger
  .kingdom addr=? terminal=tty2 res=success'
```

Листинг 170: Пример поиска сообщений аудита по типу ADD_USER

В принципе, уже сразу видно, кто и когда добавил пользователя `freshuser`, что ему также был создан домашний каталог, и что все эти операции закончились успешно.

Можно, кроме того, использовать ассоциированный с правилом аудита нужный ключ, который был присвоен при создании правила:

```
# ausearch -k identity -i | grep adduser
type=PROCTITLE msg=audit(30.07.2022 18:07:23.084:1847) : proctitle=adduser newuser
```

```
type=SYSCALL msg=audit(30.07.2022 18:07:23.084:1847) : arch=x86_64 syscall=rename success=yes exit
  =0 a0=0x7ffe83697c80 a1=0x4277a0 a2=0x7ffe83697bf0 a3=0x100 items=5 ppid=5923 pid=7166 auid=
  freshuser uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=
  tty2 ses=3 comm=adduser exe=/usr/sbin/useradd subj=kernel key=identity
type=PROCTITLE msg=audit(30.07.2022 18:07:23.097:1848) : proctitle=adduser newuser
type=SYSCALL msg=audit(30.07.2022 18:07:23.097:1848) : arch=x86_64 syscall=rename success=yes exit
  =0 a0=0x7ffe83697c50 a1=0x426040 a2=0x7ffe83697bc0 a3=0x100 items=5 ppid=5923 pid=7166 auid=
  freshuser uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=
  tty2 ses=3 comm=adduser exe=/usr/sbin/useradd subj=kernel key=identity
type=PROCTITLE msg=audit(30.07.2022 18:07:23.110:1849) : proctitle=adduser newuser
type=SYSCALL msg=audit(30.07.2022 18:07:23.110:1849) : arch=x86_64 syscall=rename success=yes exit
  =0 a0=0x7ffe83697c50 a1=0x4272e0 a2=0x7ffe83697bc0 a3=0x100 items=5 ppid=5923 pid=7166 auid=
  freshuser uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=
  tty2 ses=3 comm=adduser exe=/usr/sbin/useradd subj=kernel key=identity

# ausearch -i -a 1849
----
type=PROCTITLE msg=audit(30.07.2022 18:07:23.110:1849) : proctitle=adduser newuser
type=PATH msg=audit(30.07.2022 18:07:23.110:1849) : item=4 name=/etc/gshadow inode=131503 dev=08:02
   mode=file,440 ouid=root ogid=shadow rdev=00:00 obj=unlabeled nametype=CREATE cap_fp=none cap_fi=
  none cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(30.07.2022 18:07:23.110:1849) : item=3 name=/etc/gshadow inode=131504 dev=08:02
   mode=file,440 ouid=root ogid=shadow rdev=00:00 obj=unlabeled nametype=DELETE cap_fp=none cap_fi=
  none cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(30.07.2022 18:07:23.110:1849) : item=2 name=/etc/gshadow+ inode=131503 dev
  =08:02 mode=file,440 ouid=root ogid=shadow rdev=00:00 obj=unlabeled nametype=DELETE cap_fp=none
  cap_fi=none cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(30.07.2022 18:07:23.110:1849) : item=1 name=/etc/ inode=130819 dev=08:02 mode=
  dir,755 ouid=root ogid=root rdev=00:00 obj=unlabeled nametype=PARENT cap_fp=none cap_fi=none
  cap_fe=0 cap_fver=0 cap_frootid=0
type=PATH msg=audit(30.07.2022 18:07:23.110:1849) : item=0 name=/etc/ inode=130819 dev=08:02 mode=
  dir,755 ouid=root ogid=root rdev=00:00 obj=unlabeled nametype=PARENT cap_fp=none cap_fi=none
  cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(30.07.2022 18:07:23.110:1849) : cwd=/root
type=SYSCALL msg=audit(30.07.2022 18:07:23.110:1849) : arch=x86_64 syscall=rename success=yes exit
  =0 a0=0x7ffe83697c50 a1=0x4272e0 a2=0x7ffe83697bc0 a3=0x100 items=5 ppid=5923 pid=7166 auid=
  freshuser uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=
  tty2 ses=3 comm=adduser exe=/usr/sbin/useradd subj=kernel key=identity
```

Листинг 171: Пример поиска сообщений аудита связанных с добавлением пользователя по ключу поиска

А если нужно посмотреть, кто когда и с каким результатом входил в систему, опять-таки можно использовать нужный тип события:

```
# ausearch -m LOGIN -i
----
type=LOGIN msg=audit(06.08.2022 22:19:54.820:197) : pid=934 uid=root subj=kernel old-auid=unset
  auid=gdm tty=(none) old-ses=4294967295 ses=1 res=yes
----
type=LOGIN msg=audit(06.08.2022 22:29:14.434:256) : pid=1384 uid=root subj=kernel old-auid=unset
  auid=freshuser tty=(none) old-ses=4294967295 ses=2 res=yes

# ausearch -m USER_LOGIN -i
----
type=USER_LOGIN msg=audit(08.08.2022 10:48:16.818:6952) : pid=3774 uid=root auid=unset ses=unset
  subj=kernel msg='op=login id=freshuser exe=/bin/login hostname=rosafresh12.tiger.kingdom addr=?
  terminal=tty2 res=failed'
----
```

```
type=USER_LOGIN msg=audit(08.08.2022 10:48:34.281:6967) : pid=3775 uid=root auid=unset ses=unset
  subj=kernel msg='op=login id=freshuser exe=/bin/login hostname=rosafresh12.tiger.kingdom addr=?
  terminal=tty2 res=failed'
```

Листинг 172: Пример поиска сообщений по типам `LOGIN` и `USER_LOGIN`

Можно даже сразу узнать, не менялись ли правила пакетного фильтра:

```
# ausearch -m NETFILTER_CFG -i
<no matches>
```

Листинг 173: Пример поиска сообщений аудита по типу `NETFILTER_CFG`

Более того, служба `auditd` обладает возможностями сразу получать отчеты. По такой краткой сводке можно сразу многое понять. Например, видно, что в систему за отчетный период кто-то 13 раз безуспешно пытался войти, а бюджеты пользователей менялись 8 раз:

```
# aureport

Summary Report
======================
Range of time in logs: 30.07.2022 18:04:40.511 - 08.08.2022 18:44:20.230
Selected time for report: 30.07.2022 18:04:40 - 08.08.2022 18:44:20.230
Number of changes in configuration: 374
Number of changes to accounts, groups, or roles: 8
Number of logins: 0
Number of failed logins: 2
Number of authentications: 30
Number of failed authentications: 13
Number of users: 4
Number of terminals: 11
Number of host names: 3
Number of executables: 115
Number of commands: 171
Number of files: 1679
Number of AVC's: 0
Number of MAC events: 0
Number of failed syscalls: 610
Number of anomaly events: 32
Number of responses to anomaly events: 0
Number of crypto events: 0
Number of integrity events: 0
Number of virt events: 0
Number of keys: 33
Number of process IDs: 2442
Number of events: 16921
```

Листинг 174: Пример отчета аудита

Можно смотреть только кто и когда входил (`aureport --auth`) или только выполненные команды (`aureport --comm`). Ключи `aureport` можно узнать в интерактивной справке `aureport(8)`.

Подробные сведения об интерпретации полей событий аудита представлены в таблицах настоящего раздела (Таблице 12 и Таблице 13).

Поле в данных аудита	Расшифровка
a0, a1, a2, a3	Записи первых четырех аргументов системного вызова, представленные в шестнадцатеричной системе счисления (HEX)
acct	Учетная запись, от имени которой выполнялся процесс
action	Совершенное действие механизма КЦ[22]
appraise_type	Произведенная оценка механизмом КЦ[22]
addr	Адрес узла абонента
arch	Архитектура ЦП, представленная в шестнадцатеричной системе счисления
auid	Идентификатор пользователя (при наличии), от имени которого совершен первичный вход в систему (с последующим возможным переключением контекста)
capability	Число (в битах), обозначающее номер возможности из перечня (Capability)
cap_fe	Параметр эффективного бита возможностей для объекта ФС
cap_fi	Параметр унаследованной возможности для объекта ФС
cap_fp	Параметр разрешенной возможности для объекта ФС
cap_fver	Информация о версии для перечня возможностей (capabilities)
cap_pe	Параметр эффективного бита возможностей для субъекта
cap_pi	Параметр унаследованной возможности для субъекта
cap_pp	Параметр разрешенной возможности для субъекта
cause	Запись о причине в правилах политики КЦ[22]
cgroup	Путь к контрольной группе, во время генерации сообщения аудита, к которой относился процесс (при наличии)
comm	Запись исполняемой команды или запускаемой программы
cwd	Текущий путь в файловой системе, откуда осуществлялся вызов операции
dev	Старший и младший номер для файла устройства, на котором расположен файл
egid	Эффективный идентификатор группы для субъекта
euid	Эффективный идентификатор субъекта
exe	Полный реальный путь(идентификатор) к файлу при выполнении операции
exit	Код возврата для системного вызова. Зависит от системного вызова.
filetype	Тип файла
fsgid	Идентификатор группы пользователя, инициировавшего процесс
fsuid	Идентификатор пользователя, инициировавшего процесс
func	Выполняемая функция процесса КЦ[22]
gid	Идентификатор группы
hash	Хэш-сумма объекта при проверке КЦ [22]
hostname	Идентификатор узла, на котором произошло действие, попавшее в текущую запись аудита
icmptype	Тип пакета ICMP, которое может фиксироваться фильтром пакетов
id	Идентификатор учетной записи пользователя, чьи учетные данные фиксируются при входе, выходе, изменении и т.п.
inode	Номер ячейки с мета-данными (иноды), описывающими объект и содержащие его атрибуты
inode_gid	Идентификатор группы владельца иноды
inode_uid	Идентификатор владельца иноды
ip	Запись для адреса сокета
items	Количество записей о путях, связанных с данной записью аудита
key	Связанное с записью аудита значение ключевого фильтра в правиле аудита, заданное администратором для удобства (при наличии)
list	Идентификатор списка правил аудита (при наличии). Может принимать значения: 0 — user, 1 — task, 4 — exit, 5 — exclude
mode	Битовая маска прав доступа DAC
msg	Значение уникального идентификатора записи аудита
msgtype	Значение возвращаемого типа отказа монитора обращений MAC в некоторых случаях
name	Полный путь к файлу, переданный отслеживаемому системному вызову в качестве аргумента
new_gid	Значение нового идентификатора группы для пользователя
obj	Значение контекста MAC для объекта доступа
ogid	Группа владельца объекта
ouid	Идентификатор (реальный) пользователя, инициировавшего процесс
path	Путь к объекту, при фиксировании событий MAC
perm	Тип права DAC (запись, чтение, выполнение или смена атрибутов, при необходимости)

Поле в данных аудита	Расшифровка
pid	Идентификатор процесса
ppid	Идентификатор родительского процесса
proctitle	Полнотекстовая запись выполняемой команды
prom	Флаг для сетевого устройства, изменившего режим прослушивания
proto	Тип протокола при прохождении пакета через сетевой фильтр
result, res	Запись результата операции
saddr	Адрес сетевого сокета
sauid	Идентификатор учетной записи, от имени которой событие попало в аудит, в случае его передачи через системную шину сообщений
sec	Значение, в секундах, на которое изменилось системное время
ses	Номер (число) сессии пользователя
sgid	Значение установленной группы
sig	Значение переданного сигнала программе при сбое
subj	Значение контекста субъекта доступа SELinux MAC
success	Возвращенное системным вызовом значение успеха или неудачи
suid	Идентификатор пользователя, от имени которого запущен процесс
syscall	Идентификатор системного вызова
terminal	Имя устройства первичного терминала, связанного с процессом (при наличии)
tty	Имя устройства терминала, непосредственно связанного с процессом (при наличии)
uid	Реальный идентификатор пользователя, от имени которого запущен процесс
xattr	Значения расширенных атрибутов DAC[22]

Таблица 13: Расшифровка наиболее важных полей в записях аудита.

8.3 Ограничение использования устройств USB

Для контроля устройств USB в ОС Ubuntu Linux можно рекомендовать использование программного средства `usbguard`. Данное программное средство взаимодействует с менеджером устройств ОС (device manager) через пространство служебной ФС ядра `devfs` и является, по сути, высокоуровневым логическим интерфейсом к функциям менеджера устройств ядра в части обмена данными с устройствами USB.

Программное средство `usbguard` позволяет назначать политики доступа ко всему спектру устройств USB, определяя политики по умолчанию для доверенных устройств. В ОС Ubuntu Linux удобно то, что при установке `usbguard` все устройства USB, определенные программой в момент её установки, автоматически добавляются в политику.

Далее, после установки и автоматической настройки программа `usbguard` функционирует в качестве службы. Основной конфигурационный файл службы – это `/etc/usbguard/usbguard-daemon.conf`. Основной файл, содержащий перечень устройств и политику – это /etc/usbguard/rules.conf. Файл аудита программы – это `/var/log/usbguard/usbguard-audit.log`.

Для установки программы контроля USB устройств и её автоматической настройки в ОС Ubuntu Linux выполнить (от имени суперпользователя `root`):

```
# apt-get install usbguard
```

Листинг 175: Пример установки `usbguard`

[22] Заполняется в случае активизации механизмов контроля целостности IMA/EVM

Служба запустится автоматически, самостоятельно сформировав нужные конфигурационные файлы `/etc/usbguard/usbguard-daemon.conf` и `/etc/usbguard/rules.conf`.

При этом в конфигурационный файл политики автоматически будут занесены все обнаруженные на момент установки USB устройства и они в дальнейшем будут считаться доверенными. Для того, чтобы переделать политику целиком в случае необходимости, требуется выполнить (от имени суперпользователя `root`):

```
# usbguard generate-policy > /etc/usbguard/rules.conf
```

Листинг 176: Пример настройки политики `usbguard`

В том случае, если необходимо разрешить подключение какого-то конкретного устройства USB, например, накопителя данных, требуется (от имени суперпользователя `root`) выполнить поиск устройства:

```
# usbguard list-devices
...
33: block id 01e9:63f0 serial XXX"12345678" name "Portable SSD T5" hash
"kKaIs6W/ZI3OnNWaCBHgmXh1234567/lfyBVVfy494YQ=" parent-hash
"x9ceLltloDm7ceQyad6543210YVSX1Twdj/bnTelsH2c=" via-port "4-1.4" with-interface
{ 08:07:05 08:07:05 } with-connect-type "unknown"
```

Листинг 177: Пример просмотра подключенных устройств USB

В ответ система сообщит пронумерованные текущие идентификаторы устройств. Пример выше демонстрирует подключение SSD диска Samsung T5 на 512 Гбайт, к тому же содержащий зашифрованный раздел с ФС для резервного копирования. Как видно из примера, система присвоила этому диску номер 33. Для разового разрешения подключения данного устройства требуется выполнить (от имени суперпользователя root) следующую команду:

```
# usbguard allow-device 33
```

Листинг 178: Пример разового разрешения для подключения устройства USB

То же, но с постоянным разрешением:

```
# usbguard allow-device 33 -p
```

Листинг 179: Пример постоянного разрешения для подключения устройства USB

После чего система предложит смонтировать (подключить) диск. В том случае, если данные на нем зашифрованы с помощью LUKS, то ввести пароль расшифровки. Смонтированный в системе диск далее отображается штатным образом:

```
/dev/mapper/luks-d0c22f on /media/xxx/LUKSDEVICE type ext4
(rw,nosuid,nodev,relatime,stripe=8191,uhelper=udisks2)
```

Листинг 180: Пример отображения смонтированного устройства USB

8.4 Защита ядра и ограничение отладки (профилирования)

Операционная система Linux весьма хорошо подходит для нужд разработки. И Ubuntu Linux не исключение. Однако, разработка подразумевает получение отладочной информации, доступ к отладочным интерфейсам ядра, получение снимков памяти, получение трассировки, чтобы использовать эту информацию, в интересах профилирования и т.п.

Все это негативно влияет на безопасность. В общем случае, осуществлять разработку там, где ОС будет иметь безопасную настройку – не рекомендуется.

8.4.1 Лимиты при создании отладочных файлов

В файле `/etc/security/limits.conf` нужно проверить следующие значения, иначе установить их:

```
*   hard core 0
root  hard core 0
```

Листинг 181: Содержимое файла `/etc/security/limits.conf` для запрета на сброс дампов памяти

Если в этом файле не указаны приведенные значения, то нужно добавить приведенные выше строки, используя любой удобный текстовый редактор, и сохранить файл `/etc/security/limits.conf`.

8.4.2 Переменная ядра, воспрещающая файлы отладки

В файле `/etc/sysctl.conf` проверить, что установлена следующая переменная ядра, воспрещающая создание дампов памяти от имени процессов с установленным битом смены идентификатора пользователя (SUID bit):

```
# cat /etc/sysctl.conf | grep fs.suid
fs.suid_dumpable = 0
```

Листинг 182: Проверка текущей переменной ядра при обработки дампов

Иначе, установить переменную:

```
# echo "fs.suid_dumpable = 0" >> /etc/sysctl.conf
```

Листинг 183: Установка переменной ядра для запрета сброса дампа памяти

Требуется учитывать, что устанавливаемые переменные ядра по умолчанию применяются только после перезагрузки ОС. Если необходимо применять новые значения переменных ядра немедленно, следует использовать команду:

```
# sysctl -p
```

Листинг 184: Пример перечитывания конфигурации переменных ядра

8.4.3 Ограничения для пользователей при крахе приложений

Установить ограничение (лимит) в файле `/etc/profile` для всех интерактивных пользователей системы:

```
# echo "ulimit -S -c 0 > /dev/null 2>&1" >> /etc/profile
```

Листинг 185: Ограничение переменной среды при обработки краха

Это значение принудительно ограничивает системное окружение пользователя, но не применяется к его текущему сеансу. Чтобы оно применялось, требуется получить новый сеанс (то есть выйти из системы и повторно зайти в неё).

Для ограничений `systemd`, при том, что компонент `systemd-coredump` установлен по умолчанию, тоже нужно выполнить в файле `/etc/systemd/coredump.conf` нужные изменения:

```
Storage=none
ProcessSizeMax=0
```

Листинг 186: Ограничения службы `systemd` при обработке краха

Перезапустить службу `systemd-coredump`:

```
# systemctl daemon-reload
# systemctl restart systemd-coredump.socket
```

Листинг 187: Пример перечитывания конфигурации `systemd` и перезапуск `systemd-coredump`

8.4.4 Отключение сброса страниц памяти с помощью SysRq

Для отладки ОС Ubuntu Linux поддерживает обработку т.н. «магических клавиш» `SysRq` — это сокращение от System Request (системный запрос).

Ниже приведено описание наиболее важных сочетаний клавиш `SysRq`:

Alt+SysRq+B — Немедленно перезагрузить ОС (без синхронизации и размонтирования ФС);

Alt+SysRq+C — Выполнить принудительный крах ОС со сбросом на диск состояния памяти;

Alt+SysRq+E — Послать сигнал `SIGTERM` всем процессам кроме `Init` (`systemd`);

Alt+SysRq+I — Послать сигнал `SIGKILL` всем процессам кроме `Init` (`systemd`);

Alt+SysRq+O — Выключить компьютер;

Alt+SysRq+R — Вернуть управление клавиатурой в случае сбоя X-сервера;

Alt+SysRq+U — Перемонтировать все файловые системы в режиме «только для чтения»;

Alt+SysRq+S — Записать весь имеющийся кеш из оперативной памяти на диск.

Alt+SysRq+K — Уничтожить все процессы в текущем терминале;

Alt+SysRq+N — Сбросить приоритет всех высоко приоритетных процессов;

Alt+SysRq+F — Запустить механизм `oom_kill`, который уничтожит процесс занимающий очень много памяти;

Alt+SysRq+T — Вывести всю информацию о запущенных процессах на текущую консоль;

Alt+SysRq+L — Послать сигнал `SIGKILL` всем процессам включая `Init` (`systemd`);

Alt+SysRq+P — Выдать сброс текущего состояния регистров процессора в текущий терминал.

Для проверки текущей конфигурации `SysRq` выполнить:

```
# cat /proc/sys/kernel/sysrq
0
```

Листинг 188: Проверка текущих параметров `SysRq`

Иначе, если значение вывода отлично от нуля, выполнить:

```
# echo "0" > /proc/sys/kernel/sysrq
# echo "kernel.sysrq = 0" >> /etc/sysctl.conf
# sysctl -p
```

Листинг 189: Настройка запрета использования `SysRq`

8.4.5 Отключение трассировки процессов

На первом этапе нужно проверить, есть ли в выполняющемся ядре LSM модуль YAMA:

```
# /boot/config-5.10.0-19-amd64 | grep YAMA
CONFIG_SECURITY_YAMA=y
```

Если есть, то в ОС может использоваться настройка, позволяющая производить ограничения на трассировку процессов. Если нет (выдано значение `CONFIG_SECURITY_YAMA is not set`), то можно пропустить эту настройку.

Для проверки текущих значений трассировки выполнить (от имени администратора `root`):

```
# sysctl -a | grep ptrace
kernel.yama.ptrace_scope = 2
```

Листинг 190: Проверка текущей политики трассировки процессов

В случае, если вывод отличается от двойки, то нужно установить запрет трассировки, используя значение «2» для переменной `kernel.yama.ptrace_scope` (или более строгое). Значение «2» определяет, что трассировку процессов может осуществлять только /glsадминистратор `root`. Значение «3» полностью отключает трассировку. В описанной ниже конфигурации трассировка разрешается только пользователю `root`:

```
# sysctl -w kernel.yama.ptrace_scope=2
# echo "kernel.yama.ptrace_scope = 2" >> /etc/sysctl.conf
```

Листинг 191: Запрет возможности трассировки процессов для обычных пользователей

8.4.6 Ограничения на просмотр сообщений ядра

Популярной в Linux командой является команда `dmesg`, которая выводит на экран сообщения ядра ОС. По умолчанию эту команду может использовать любой пользователь в системе, следовательно и злоумышленник.

`kernel.dmesg_restrict` – переменная ядра ОС, отвечающая за доступ к интерфейсу кольцевого буфера аудита ядра (файлу `/dev/kmsg`) для обычных пользователей (кроме `root`). В стандартной системе Linux доступ пользователей к этому интерфейсу как правило не запрещен. Соответственно, любой пользователь ОС сможет или напрямую обратиться к этому файлу (`cat /dev/kmsg`), или использовать программы чтения кольцевого буфера аудита ядра, такие как `dmesg` или `syslog`. Рекомендуется ограничивать пользователей в возможности получать сообщения кольцевого буфера аудита ядра. Рекомендуемое значение переменной «`1`», если установлено значение «`0`», то доступ пользователей к буферу аудита ядра не ограничивается.

Проверить текущее значение политики доступа к интерфейсу кольцевого буфера аудита ядра можно так:

```
# sysctl -a | grep dmesg
kernel.dmesg_restrict = 1
```

Листинг 192: Проверка текущей политики ограничений для `dmesg` и `/dev/kmsg`

Если при проверке значение отличается от единицы, то нужно выполнить настройку запрета чтения из этого интерфейса всем, кроме `root`:

```
# sysctl -w kernel.dmesg_restrict=1
# echo 'kernel.dmesg_restrict = 1' >> /etc/sysctl.conf
```

Листинг 193: Настройка политики ограничений для `dmesg` и `/dev/kmsg`

8.4.7 Технология защиты ядра Lockdown

В ядре Linux начиная с версии 5.4 появилась поддержка специальной технологии защиты ядра под названием Lockdown.

Эта технология нужна для ограничения воздействия на выполняющееся ядро даже со стороны `root`. Логика здесь в том, что если злоумышленник все же добился прав `root`, то нужно препятствовать его попыткам загрузить другое ядро или прочесть важные данные из памяти ядра (например, ключи шифрования и т.п.).

Однако стоить помнить, что в таком случае[23] невозможен переход в сон (режим гибернации). А также ограничивается доступ для `root` к довольно большому количеству интерфейсов ядра: `/dev/mem` (`/dev/kmem`), `/dev/port`, `/proc/kcore`, `debugfs`, отладочному режиму `kprobes`, `mmiotrace`, `tracefs`, BPF, некоторым интерфейсам ACPI и MSR-регистрам процессора, блокируется использование системных вызовов `kexec_file()` и `kexec_load()`, не допускаются манипуляции с портами ввода/вывода, в том числе изменение номера прерывания и порта ввода/вывода для последовательного порта, а также блокируются некоторые другие интерфейсы ядра, используемые реже. Одним словом, для разработки такой режим не годится.

Поскольку в ОС Ubuntu Linux 20.04.5 используется ядро версии не ниже 5.4, то защиту ядра Lockdown можно активизировать. По умолчанию она выключена[24]. У технологии Lockdown два возможных режима работы. Менее строгий, и более строгий.

Первый режим называется `integrity`, и препятствует воздействию на работающее ядро как со стороны пользовательского пространства, так и со стороны `root`. Проще говоря, нельзя будет выполнить загрузку другого ядра с помощью `kexec_load()` или `kexec_file_load()`[25], или сбросить дамп ядра с помощью `kdump`. Интерфейсы отладки и возможности со стороны администратора `root` получать данные из ядра в этом режиме сохраняются.

Второй вариант, более строгий, называется `confidentiality`. Помимо того, что нельзя манипулировать с работающим ядром, все отладочные и некоторые другие интерфейсы ядра блокируются даже для администратора `root`, как было описано выше.

Проверить, возможно ли включение технологии Lockdown на текущем ядре, можно просмотрев конфигурационный файл загруженного ядра, например:

```
cat /boot/config-5.10.0-18-amd64 | grep LOCKDOWN
CONFIG_SECURITY_LOCKDOWN_LSM=y
CONFIG_SECURITY_LOCKDOWN_LSM_EARLY=y
```

Листинг 194: Пример возможности включения Lockdown в текущем ядре

Если ядро поддерживает включение режима Lockdown, то нужно загрузить ядро с соответствующими опциями. Для этого в файле `/etc/default/grub` в строке параметров загрузки ядра нужно задать:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash lockdown=confidentiality"
```

Листинг 195: Включение Lockdown в режим `confidentiality`

После чего обновить конфигурацию загрузчика и перезагрузить ОС с режимом Lockdown:

```
# update-grub2
# reboot
```

Листинг 196: Обновление параметров загрузчика

[23] Имеется ввиду режим максимальной защиты `confidentiality`.
[24] Если установить систему с поддержкой UEFI Secure Boot, то Lockdown может включится в режиме `integrity`.
[25] Интерфейсы использования этих системных вызовов входят в состав пакета `kexec-tools`.

Режим работы Lockdown указывается в квадратных скобках:

```
# cat /sys/kernel/security/lockdown
none integrity [confidentiality]
```

Листинг 197: Проверка режима фукционирования Lockdown

При включенном Lockdown в режиме конфиденциальности можно убедиться в том, что даже `root` лишился возможности запрашивать данные из интерфейсов ядра, обычно доступных по умолчанию:

```
# cat /dev/mem
cat: /dev/mem: Операция не позволена
# cat /dev/port
cat: /dev/port: Операция не позволена
# cat /proc/kcore
cat: /proc/kcore: Операция не позволена
```

Листинг 198: Попытка запроса даных из интерфейсов ядра при включенном Lockdown

8.5 Отключение поддержки протокола IPv6

Рекомендуется выключить поддержку протокола IPv6, если необходимость в его использовании отсутствует. Это снижает избыточность, и общую площадь возможной атаки на систему.

Для проверки того, поддерживает ли система работу по протоколу IPv6, выполнить:

```
# grep "^\s*linux" /boot/grub/grub.cfg | grep -v "ipv6.disable=1"
```

Листинг 199: Проверка поддержки IPv6

Вывода быть не должно.

Если вывод есть, то можно отключить поддержку протокола IPv6. Для этого рекомендуется отредактировать файл `/etc/default/grub` и переустановить загрузчик GRUB2:

```
GRUB_CMDLINE_LINUX="ipv6.disable=1"
```

Листинг 200: Отключение IPv6 в файле `/etc/default/grub`

8.6 Настройка фильтра пакетов

Если не использовать правильно настроенный фильтр пакетов, то защищаемая информация может незаметно покинуть пределы системы. Кроме того, злоумышленник сможет незаметно и безнаказанно воздействовать на защищаемую систему с помощью сетевых средств анализа и нападения.

Фильтр пакетов встроен в ядро ОС Linux (в подсистеме ядра `NetFilter`), где и происходит обработка пакетов. Такой подход обеспечивает его высокую надежность, безопасность, производительность и гарантирует сложность обхода функций управления информационными потоками.

Для пользователя существует несколько программ управления фильтром пакетов. Например, в ОС есть несколько возможных интерфейсов к фильтру пакетов ядра, но из соображений универсальности

в данном разделе описывается только конфигурация с помощью `iptables`. Остальные интерфейсы в разделе не рассматриваются.

Фильтр пакетов представляет собой набор правил. Когда сетевой пакет (с данными или без них) проходит фильтр, то содержимое такого сетевого пакета (например, адреса источника и получателя, сетевой порт, протокол соединения, и т.п.) должно быть исследовано правилами фильтра пакетов, чтобы сделать заключение (принять решение) о правомерности движения сетевого пакета (или информационного потока).

Программа `IPTables` – это приложение пользовательского уровня (user space application), позволяющее настраивать таблицы, цепочки и правила пакетного фильтра ядра ОС Linux. При этом `IPTables` содержит несколько программных модулей:

- программа `/usr/sbin/iptables` – управляет правилами пакетного фильтра, предназначенного для протоколов семейства IPv4;
- программа `/usr/sbin/ip6tables` – управляет правилами пакетного фильтра, предназначенного для протоколов семейства IPv6;
- программа `/usr/sbin/arptables` – управляет правилами пакетного фильтра, предназначенного для протоколов семейства ARP;
- программа `/usr/sbin/ebtables` – управляет правилами пакетного фильтра, предназначенного для протоколов семейства Ethernet.

Рекомендуется использовать только один пакетный фильтр во избежание путаницы. Данный раздел описывает настройку пакетного фильтра `IPTables`, при этом предполагается, что использование других интерфейсов или пакетных фильтров не предусмотрено. Также предполагается, что работа с соединениями протокола IPv6 не предусмотрена, и данный функционал отключен на уровне ядра операционной системы (как отключить IPv6 написано в разделе 8.5).

8.6.1 Установка фильтра пакетов `IPTables`

Проверить, что фильтр пакетов `nftables` недоступен:

```
# dpkg -s nftables
dpkg-query: пакет «»nftables не установлен, информация о нём недоступна
Use dpkg --info (= dpkg-deb --info) to examine archive files.
```

Листинг 201: Проверка установки `nftables`

Вывода быть не должно. Иначе нужно удалить пакетный фильтр `nftables`:

```
# apt purge nftables
```

Листинг 202: Удаление `nftables`

Для проверки того, установлен ли фильтр пакетов `IPTables`, выполнить:

```
# dpkg -s iptables
Package: iptables
Status: install ok installed
```

Листинг 203: Проверка установки `iptables`

Если пакетный фильтр `iptables` не установлен, то установить его можно так:

```
# apt install iptables iptables-persistent
```

Листинг 204: Установка `iptables`

8.6.2 Пример настройки пакетного фильтра `IPTables`:

В общем случае описываемая настройка предполагает установку следующей политики:

- конфигурацию для соединений интерфейса «обратной петли» (loopback);
- конфигурацию соединений с учетом состояний;
- разрешение на прием соединений для `ssh` откуда угодно;
- разрешение на создание соединений от абонента куда угодно;
- политику отказа во всех остальных случаях.

Для реализации перечисленных выше политик предлагается создать следующий сценарий (например, `iptables.sh`) пакетного фильтра и затем его выполнить:

```
#!/bin/bash
# Сброс всех правил пакетного фильтра
iptables -F

# Установка политики отказа в установлении соединения по-умолчанию
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Настройка фильтра для интерфейса «обратной петли»
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables  -A INPUT -s 127.0.0.0/8 -j DROP

# Настройка исходящих и установленных соединений с учетом состояния
iptables -A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p udp -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -p udp -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -p icmp -m state --state ESTABLISHED -j ACCEPT

# Разрешение входящих соединий для службы ssh
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
```

Листинг 205: Пример сценария политики `iptables`

```
# chmod u+x ./iptables.sh
# ./iptables.sh
# iptables -L
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
DROP       all  --  localhost/8          anywhere
ACCEPT     tcp  --  anywhere             anywhere            state ESTABLISHED
ACCEPT     udp  --  anywhere             anywhere            state ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere            state ESTABLISHED
ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ssh state NEW

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere            state NEW,ESTABLISHED
ACCEPT     udp  --  anywhere             anywhere            state NEW,ESTABLISHED
ACCEPT     icmp --  anywhere             anywhere            state NEW,ESTABLISHED
```

Затем можно добавить этот набор правил в автозагрузку при старте системы:

```
# iptables-save > /etc/sysconfig/iptables
# systemctl enable iptables
# systemctl start iptables
```

Желательно постоянно отслеживать состояния сетевых соединений для выявления подозрительной активности. Это поможет предотвратить появление скрытых каналов передачи информации,усилит защиту от сетевых атак.

Для этого можно проверять состояние доступных к соединению сокетов и портов, а затем сопоставлять правила и выявленные сокеты следующим образом:

```
# ss -4tuln
Netid  State   Recv-Q  Send-Q   Local Address:Port    Peer Address:Port Process
udp    UNCONN  0       0        127.0.0.53%lo:53           0.0.0.0:*
udp    UNCONN  0       0              0.0.0.0:44194        0.0.0.0:*
udp    UNCONN  0       0              0.0.0.0:5353         0.0.0.0:*
tcp    LISTEN  0       4096     127.0.0.53%lo:53           0.0.0.0:*
tcp    LISTEN  0       128            0.0.0.0:22           0.0.0.0:*

# iptables -L INPUT -v -n
```

Листинг 206: Пример отслеживания политик `iptables`

8.7 Защита памяти

8.7.1 Аппаратная защита от переполнения буфера

Для проверки того, задействованы ли в BIOS или UEFI функции аппаратной защиты от переполнения буфера, требуется от имени любого пользователя (пользователя) выполнить:

```
$ journalctl | grep "protection: active"
окт 01 12:19:46 dl5310 kernel: NX (Execute Disable) protection: active
```

Листинг 207: Проверка аппаратной защиты от переполнения буфера.

В том случае, если вывод отличается от приведенного выше, то требуется включить соответствующие опции в BIOS/UEFI и перепроверить.

8.7.2 Программная защита от переполнения буфера

Разработчики ОС Ubuntu Linux предпринимают усилия, направленные на защиту от переполнения буфера. Но эта защита никак не конфигурируется в операционной системе. Такая защита осуществляется на этапе сборки, прямо в сборочной среде с помощью специальных опций компилятора GCC типа `fstack-protection`.

Хотя эту защиту нельзя сконфигурировать, можно просмотреть каким образом собран тот или иной пакет. Информация о используемых опциях сборки, влияющих на безопасность, для всех пакетов приведена на официальной странице Ubuntu:

https://wiki.ubuntu.com/ToolChain/CompilerFlags

Проверить это самостоятельно можно, выполнив команду:

```
$ dpkg-buildflags
CFLAGS=-g -O2 -ffile-prefix-map=. -fstack-protector-strong -Wformat -Werror=format-security
CPPFLAGS=-Wdate-time -D_FORTIFY_SOURCE=2
CXXFLAGS=-g -O2 -ffile-prefix-map=. -fstack-protector-strong -Wformat -Werror=format-security
DFLAGS=-frelease
FCFLAGS=-g -O2 -ffile-prefix-map=. -fstack-protector-strong
FFLAGS=-g -O2 -ffile-prefix-map=. -fstack-protector-strong
GCJFLAGS=-g -O2 -ffile-prefix-map=. -fstack-protector-strong
LDFLAGS=-Wl,-z,relro
OBJCFLAGS=-g -O2 -ffile-prefix-map=. -fstack-protector-strong -Wformat -Werror=format-security
OBJCXXFLAGS=-g -O2 -ffile-prefix-map=. -fstack-protector-strong -Wformat -Werror=format-security
```

Листинг 208: Проверка программной защиты от переполнения буфера.

Вывод свидетельствует о том, что для любой сборки по умолчанию применяется опция защиты от переполнения буфера, т.н. «канарейка»: `-fstack-protector-strong`.

8.7.3 Защита от атак типа Meltdown и Spectre

Семейство атак типа Meltdown и Spectre, вызываемые ошибками контроля доступа к памяти при спекулятивном выполнении инструкций процессора и особенностями функционирования модуля прогнозирования ветвлений процессоров – могут привести к тому, что злоумышленник сможет получить доступ к защищенной памяти из программы, не обладающей соответствующими привилегиями (путём анализа данных, записываемых в кэш процессора).

Эти уязвимости были обнаружены более пяти лет назад, но все еще продолжают появляться различные их варианты. Одним из наиболее эффективных способов противостояния им по-прежнему

(помимо обновлений инструкций самих процессоров и разнообразных патчей к ядру Linux и компиляторам) является аппаратное отключение SMT, особенно там, где это не нужно.

Рекомендуется проанализировать, нужно ли использовать технологию SMT (которая, безусловно, полезна для производительности и разделения ресурсов). Но если без неё все же можно обойтись – то лучше отключить SMT.

Для проверки того, используется ли технология SMT, требуется от имени любого пользователя выполнить следующую команду:

```
$ cat /sys/devices/system/cpu/smt/active
0
```

Листинг 209: Проверка поддержки технологии SMT

Где «0» свидетельствует об отсутствии поддержки.

Иначе, если вывод «1», то требуется отключить поддержку SMT, сначала в BIOS/UEFI. А затем выключить на уровне операционной системы.

Для отключения поддержки SMT в ОС, требуется от имени root выполнить изменение строки «GRUB_CMDLINE_LINUX» конфигурационного файла /etc/default/grub, дописав в ее конец следующие директивы:

```
mitigations=auto,nosmt
```

Листинг 210: Настройка политик загрузчика и ядра, воспрещающей использование SMT

После этого установить новую конфигурацию загрузчика (от имени администратора root):

```
# update-grub2
```

Листинг 211: Обновление конфигурации загрузчика GRUB2 при настройке SMT

Далее выполнить перезагрузку и перепроверить.

8.7.4 Защита адресного пространства

kernel.kptr_restrict – переменная ядра ОС, отвечающая за доступ к интерфейсу ядра /proc/kallsyms и просмотру значений адресов в памяти для некоторых функций ядра. Переменная может принимать значения «0», «1» и «2».

Если определено значение «0», то просматривать значения адресов в памяти может любой пользователь ОС. Если задано значение «1», то просматривать адресацию функций может только root. Если значение равно «2», то никто не получит информацию об адресации. Рекомендуемое значение – два. При значении единица – отображение адресов памяти заменяется на нули для всех пользователей, кроме root. При значении два – отображение адресов памяти заменяется на нули для всех пользователей, включая root.

Для проверки текущего значения этой переменной выполнить:

```
# sysctl -a | grep kptr
```

```
kernel.kptr_restrict = 2
```

Листинг 212: Проверка политики ограничений для `/proc/kallsysms`

Если значение отличается, то выполнить установку этой переменной:

```
# sysctl -w kernel.kptr_restrict=2
# echo 'kernel.kptr_restrict = 2' >> /etc/sysctl.conf
```

Листинг 213: Настройка политики ограничений для `/proc/kallsysms`

8.8 Настройка изоляции процессов

В страницах памяти виртуального адресного пространства, выделяемых процессам, содержится информация, необходимая им для выполнения и обработки. Поэтому в страницах памяти могут храниться ключи шифрования, защищаемые данные, хеши паролей пользователей, идентификаторы пользователей и файлов, содержимое файлов и т.п.

Злоумышленник может получить доступ к данным, хранящимся в страницах памяти, если механизма изоляции процессов настроен неправильно. Поэтому требуется обеспечить невозможность или существенно затруднить злоумышленнику доступ к чужому или предыдущему содержанию страниц памяти. Для этого в составе ядра содержится функция поддержки случайного выделения страниц памяти. Правильная настройка ASLR обеспечивает изоляцию памяти для процессов.

Применение ASLR существенным образом затрудняет для злоумышленника возможность эксплуатации уязвимостей, связанных с повторным получением доступа к страницам памяти. Для настройки ASLR используется переменная ядра `kernel.randomize_va_space`. Эта переменная ядра может принимать разные значения:

- Значение «`0`», определяет, что случайного выделения адресного пространства не происходит, и распределение страниц памяти происходит статично.
- Значение «`1`» определяет консервативную рандомизацию. Однако, данные об общих библиотеках, стеке, `mmap( )` VDSO и куча рандомизированы.
- Значение «`2`» определяет полную рандомизацию. В дополнение к элементам, перечисленным ранее, управляемая память `brk( )`также рандомизирована.

Рекомендуется использовать полную рандомизацию адресного пространства, следовательно, значение переменной `kernel.randomize_va_space` должно быть установлено в «`2`».

Для проверки текущего значения переменной ядра для функции изоляции процессов необходимо выполнить следующую команду:

```
# sysctl -a | grep kernel.randomize_va_space
kernel.randomize_va_space = 2
```

Листинг 214: Проверка текущей политики ядра в отношении изоляции процессов

В ответ система должна сообщить текущее значение параметра ядра в отношении изоляции процессов. В том случае, если выведенное на экран значение изоляции отлично от «2», требуется произвести настройку ASLR. Для активизации поддержки функции изоляции процессов ASLR и ее настройки на максимальную рандомизацию, нужно выполнить:

```
# echo "kernel.randomize_va_space = 2" >> /etc/sysctl.conf
# sysctl -w kernel.randomize_va_space=2
```

Листинг 215: Настройка политики ядра для рандомизации выделения виртуальной памяти процессу

8.9 Рекомендуемые безопасные значения переменных ядра ОС

Ядро ОС Linux может принимать значительное количество переменных (опций) для безопасной настройки. Все они описаны в данном разделе и сведены в таблицу 14 с рекомендуемыми значениями и описанием.

№ п/п	Переменная	Значение	Описание
1	`dev.tty.ldisc_autoload`	0	Переменная ядра ОС, отвечающая за автоматическое определение и назначение параметров терминала. Значение «0» запрещает автоматическое назначение параметров дисциплины линии (line discipline, ldisk).
2	`fs.protected_fifos`	2	Переменная ядра ОС, отвечающая за создание специальных файлов-сокетов FIFO в общедоступных каталогах. Потенциальный нарушитель может попытаться несанкционированно создать сокет в общедоступном каталоге, например, с целью создания попытки организации связи с оборудованием или воздействия на механизмы обработки взаимодействия между процессами в системе. Рекомендуемое значение этой переменной – не менее «1», которое сигнализирует ядру о том, что ядру нельзя выполнять системный вызов `open()` или `creat()` с флагом `O_CREAT` (и, следовательно, нельзя создать файл) в общедоступном каталоге любому пользователю, за исключением владельца каталога. Значение «2» учитывает также группу. Значение «`0`» разрешает создавать сокеты без ограничений.
3	`fs.protected_hardlinks`	2	Задает политику ужесточения при разграничении доступа к ссылкам, где значение «`0`» разрешает создание и переход по ссылкам без ограничений, значение «1» определяет, что переход по символическим ссылкам разрешен только в том случае, если они находятся за пределами каталога с установленным битом Sticky, или когда UID владельца символической ссылки и подписчика совпадают, или когда владелец каталога совпадает с владельцем символической ссылки. Значение «2», то же, что и «1», но с учетом принадлежности к группе.
4	`fs.protected_symlinks`	2	То же, что и `fs.protected_hardlinks`
5	`fs.protected_regular`	2	То же, что и `fs.protected_fifos`
6	`fs.suid_dumpable`	0	Переменная ядра, воспрещающая создание дампов памяти от имени процессов с установленным биты смены идентификатора суперпользователя (SUID bit)

№ п/п	Переменная	Значение	Описание
7	kernel.ctrl-alt-del	0	Переменная отвечает за порядок обработки аппаратного прерывания, создающегося в результате нажатия кнопок «Alt+Crtl+Del». Может принимать значение «0» и «1». Если установлена в «0», то ядро производит перехват события, но не производит обработку прерывания. Вместо этого, параметр «0» сигнализирует ядру, чтобы обработку события нажатия «Alt+Ctrl+Del» произвела служба инициализации (systemd, init, upstart и т.п.) в соответствии с её настройками. Если задано значение «1», то ядро будет осуществлять немедленную и безусловную перезагрузку. Даже без синхронизации кеша буферов. В защищенной системе рекомендуется использовать значение «0» для этой переменной.
8	kernel.dmesg_restrict	1	Переменная ядра ОС, отвечающая за разграничение доступа к интерфейсу кольцевого буфера аудита ядра (файлу /dev/kmsg) для обычных пользователей (кроме root). В стандартной системе Linux доступ пользователей к этому интерфейсу не запрещен. Соответственно, любой пользователь ОС сможет или напрямую обратиться к этому файлу (cat /dev/kmsg), или использовать программы чтения аудита кольцевого буфера ядра, такие как dmesg или syslog. В защищенной системе требуется ограничивать пользователей в возможности получать сообщения аудита ядра. Рекомендуемое значение переменной «1», если установлено значение «0», то доступ пользователей к буферу аудита ядра не ограничивается.
9	kernel.kptr_restrict	2	Переменная ядра ОС, отвечающая за разграничение доступа к интерфейсу ядра /proc/kallsyms и просмотру значений адресов в памяти для некоторых функций ядра. Может принимать значения «0», «1» и «2». Если определено значение «0», то просматривать значения адресов в памяти может любой пользователь ОС. Если задано значение «1», то просматривать адресацию функций может только root. Если значение «2», то никто, кроме ядра, не получит информацию об адресации функций. В защищенной системе рекомендуемое значение «2». При значении «1» адреса памяти заменяются на нули для всех пользователей, кроме root. При значении «2» адреса памяти заменяются на нули для всех пользователей, включая root.
10	kernel.modules_disabled	Зависит от оборудования	Переменная ядра ОС, отвечающая за загрузку/выгрузку модулей (драйверов) ядра. Значение «0» означает, что модуль может быть загружен или выгружен. Значение «1» означает, что загрузка и выгрузка модулей невозможна. В защищенной системе должен быть четко определен состав технических и программных средств. Следовательно, модули ядра, их количество и параметры загрузки также должны быть четко определены. Однако требуется учитывать, что иногда загрузка модулей должна быть разрешена (например, при использовании ноутбуков или специфической периферии). В общем случае рекомендуемое значение этой переменной «1».

№ п/п	Переменная	Значение	Описание
11	kernel.perf_event_paranoid	2	Переменная ядра ОС, которая отвечает за возможности профилирования ядра и получения информации о производительности. Эта переменная может принимать значения «-1», «0», «1», «2» и более. При этом «-1» означает, что пользователь имеет возможность получать прямые необработанные данные трассировки ядра ОС. Значение «0» означает, что пользователь может получать трассировку о событиях процессора. Значение «1» означает, что пользователь может получать данные профилирования ядра (получать данные порядка обработки команд). Значение «2» препятствует получению данных профилировки ядра. В защищенной системе рекомендуется использовать значение «2» или выше для этой переменной.
12	kernel.randomize_va_space	2	Используется для настройки ASLR. Значение «0», определяет, что случайного выделения адресного пространства не происходит, и распределение страниц памяти происходит статично. Значение «1» определяет консервативную рандомизацию. Однако, данные об общих библиотеках, стеке, mmap( ) VDSO и куча рандомизированы. Значение «2» определяет полную рандомизацию. В дополнение к элементам, перечисленным ранее, управляемая память brk( )также рандомизирована. Рекомендуется использовать полную рандомизацию адресного пространства, следовательно, значение переменной kernel.randomize_va_space должно быть установлено в «2».
13	kernel.sysrq	0	Описание приведено в 8.4.4
14	kernel.unprivileged_bpf_disabled	1	Переменная ядра ОС, которая разрешает непривилегированный доступ к подсистеме ядра BPF (Berkley Packet Filter). Может принимать значение «0» и «1». В значении «0» доступ предоставляется любому пользователю. В значении «1» доступ предоставляется только пользователю root. В защищенной системе рекомендуется устанавливать в значение «1».
15	net.core.bpf_jit_harden	1	Переменная ядра ОС, регламентирующая доступ к JIT компилятору фильтра BPF. Значение «0» не препятствует доступу, значение «1» предоставляет доступ только пользователю root, значение «2» никому не предоставлять доступ. В защищенной системе рекомендуется использовать значение «1» или более строгое.
16	net.ipv4.conf.all.accept_redirects net.ipv4.conf.all.send_redirects net.ipv4.conf.all.accept_source_route net.ipv4.conf.default.accept_redirects net.ipv4.conf.default.accept_source_route	0	Отключают возможности перенаправления трафика (forwarding), что противодействует атакам навязывания маршрутов.
17	net.ipv4.conf.all.log_martians net.ipv4.conf.default.log_martians	1	Обеспечивает противодействие атакам типа «IP-Spoofing»
18	net.ipv4.conf.all.rp_filter net.ipv4.icmp_ignore_bogus_error_responses	1	Обеспечивает противодействие фиктивным ICMP пакетам
19	net.ipv4.icmp_echo_ignore_broadcasts	1	Обеспечивает противодействие атакам типа «Smurf»
20	net.ipv4.tcp_syncookies	1	Обеспечивает противодействие атакам типа «SYN-Flood»

Таблица 14: Рекомендуемые значения переменных ядра ОС

8.10 Использование fail2ban

Для ограничения попыток аутентификации произвольных приложений (в том числе, для любых веб-приложений), рекомендуется использовать программу fail2ban.

Программа fail2ban служит для централизованного управления доступом к различным сервисам (в том числе – к произвольным), отслеживая соединения и управляя правилами сетевого фильтра для блокировки траффика. Fail2ban — это инструмент, который помогает защитить сервисы от атак с помощью перебора паролей (в том числе от автоматизированных атак), отслеживая журналы служб и состояние соединений на предмет вредоносной активности. Программа использует регулярные выражения для сканирования файлов журнала. Подсчитываются все записи, соответствующие шаблонам, и когда их количество достигает определенного предопределенного порога, fail2ban блокирует нарушающий IP-адрес с помощью системного брандмауэра на определенный период времени. По истечении срока запрета IP-адрес удаляется из запретного списка.

Пакет fail2ban включен в репозитории Ubuntu 20.04 по умолчанию. Чтобы установить его, введите следующую команду от имени пользователя root или пользователя с привилегиями sudo:

```
# apt-get install fail2ban
```

Листинг 216: Пример установки fail2ban

После завершения установки служба fail2ban запустится автоматически. В этом можно убедиться, проверив статус услуги:

```
# systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
Active: active (running) since Wed 2020-08-19 06:16:29 UTC; 27s ago
Docs: man:fail2ban(1)
Main PID: 1251 (f2b/server)
Tasks: 5 (limit: 1079)
Memory: 13.8M
CGroup: /system.slice/fail2ban.service
└─1251 /usr/bin/python3 /usr/bin/fail2ban-server -xf start
```

Листинг 217: Пример проверки выполения fail2ban

Стандартная установка fail2ban включает два файла конфигурации:

/etc/fail2ban/jail.conf

и

/etc/fail2ban/jail.d/defaults-debian.conf.

Не рекомендуется изменять эти файлы, так как они могут быть перезаписаны при обновлении пакета.

Программа fail2ban читает файлы конфигурации в следующем порядке. Каждый файл .local отменяет настройки из файла .conf:

/etc/fail2ban/jail.conf

/etc/fail2ban/jail.d/*.conf

/etc/fail2ban/jail.local
/etc/fail2ban/jail.d/*.local

Для большинства пользователей, самый простой способ настроить fail2ban это скопировать jail.conf в jail.local и изменить файл .local.

Более опытные пользователи могут создать .local файл конфигурации с нуля. Файл .local не должен включать в себя все параметры из соответствующего файла .conf, только те, которые требуется заменить.

Создайте файл конфигурации .local из файла по умолчанию jail.conf:

```
# cp -v /etc/fail2ban/jail.{conf,local}
```

Листинг 218: Пример создания файла конфигурации fail2ban

В директиву ignoreip можно добавить IP-адреса, диапазоны IP-адресов или хосты, которые требуется исключить из запрета. Здесь можно добавить IP-адреса локальной сети, и т.п., которые необходимо добавить в белый список.

Раскомментируйте строку, начинающуюся с ignoreip и добавьте свои IP-адреса через пробел:

```
ignoreip = 127.0.0.1/8 ::1 123.123.123.123 192.168.1.0/24
```

Листинг 219: Пример создания белого списка адресов fail2ban

Варианты значения параметров bantime, findtime и maxretry – это определения времени запрета и условия запрета.

bantime – это срок, на который IP заблокирован. Если суффикс не указан, по умолчанию используются секунды. По умолчанию установлено bantime значение 10 минут. Чтобы установить время блокировки в 1 день, можно использовать директиву:

```
bantime  = 1d
```

Листинг 220: Пример изменения параметра bantime

Чтобы навсегда заблокировать нежелательный адрес IP, используйте отрицательное число.

findtime- это промежуток времени между количеством сбоев до установки запрета. Например, если fail2ban настроен на запрет IP-адреса после пяти сбоев (см. ниже maxretry), эти сбои должны произойти в течение указанного периода findtime.

Например:

```
findtime  = 10m
```

Листинг 221: Пример изменения параметра findtime

maxretry- количество отказов до блокировки IP-адреса. По умолчанию установлено значение пять, что должно подойти большинству пользователей. Для указания количества попыток совершения соединения в 10, укажите соответствующее значение параметра.

```
maxretry = 10
```

Листинг 222: Пример изменения параметра `maxretry`

Программа `fail2ban` может отправлять уведомления по электронной почте, когда IP-адрес заблокирован. Чтобы получать электронные письма, вам необходимо установить SMTP на вашем сервере и изменить действие по умолчанию, которое запрещает только IP `%(action_mw)s`, как показано ниже:

```
action = %(action_mw)s
```

Листинг 223: Пример изменения параметра `maxretry`

Действие `%(action_mw)s` заблокирует подозрительный IP-адрес и отправит электронное письмо с отчетом Whois. Если вы хотите включить соответствующие журналы в электронное письмо, установите для действия значение `%(action_mwl)s`.

Вы также можете настроить адреса электронной почты для отправки и получения:

```
destemail = adminfail2ban@domain.com
sender = fail2ban@domain.com
```

Листинг 224: Пример указания адресов для получения отчетов

Fail2ban использует концепцию т.н. «тюрем». «Тюрьма» – это логическое представление сетевой изоляции. Там описывается служба и включаются фильтры и действия. Записи журнала, соответствующие шаблону поиска, подсчитываются, и при выполнении заранее определенного условия выполняются соответствующие действия.

Такие изолированные сущности описываются в файле конфигурации `/etc/fail2ban/jail.local`

Fail2ban поставляется с несколькими тюрьмами для различных сервисов. Вы также можете создавать свои собственные конфигурации тюрьмы.

По умолчанию включен только `ssh jail`. Для включения `jail` вам нужно добавить `enabled = true` после заголовка `jail`. В следующем примере показано, как включить `jail` для службы `proftpd`:

```
[proftpd]

port     = ftp,ftp-data,ftps,ftps-data
logpath  = %(proftpd_log)s
backend  = %(proftpd_backend)s
```

Листинг 225: Пример активизации jail для службы `proftpd` в `/etc/fail2ban/jail.local`

Параметры ограничений также могут быть установлены для каждой «тюрьмы», например:

```
[sshd]
enabled   = true
maxretry  = 3
findtime  = 1d
bantime   = 4w
```

```
ignoreip  = 127.0.0.1/8 23.34.45.56
```

Листинг 226: Пример определения ограничений для службы `sshd` в `/etc/fail2ban/jail.local`

Фильтры расположены в каталоге `/etc/fail2ban/filter.d`, хранящемся в файле с тем же именем, что и «тюрьма». Если есть требования к индивидуальным настройкам ограничений служб и опыт работы с регулярными выражениями, то можно точно настроить фильтры.

Каждый раз после изменения файла конфигурации, необходимо перезапускать сервис `fail2ban`, чтобы изменения вступили в силу:

```
# systemctl restart fail2ban
```

Листинг 227: Пример перезапуска `fail2ban`

Программа fail2ban поставляется с инструментом командной строки с именем `fail2ban-client`, который вы можете использовать для взаимодействия со службой `fail2ban`.

Для просмотра всех доступных опций вызовите команду с опцией `-h`.

Примеры использования команды управления `fail2ban`:

```
  # fail2ban-client status sshd
```

Листинг 228: Пример проверки статуса изоляции для `sshd`

```
  # fail2ban-client set sshd unbanip 12.34.56.78
```

Листинг 229: Пример исключения адреса из списка заблокированных для заданной jail

```
  # fail2ban-client set sshd banip 12.34.56.78
```

Листинг 230: Пример включения адреса в список заблокированных для заданной jail

8.11 Рекомендации по проведению анализа защищенности

В настоящем разделе приведены рекомендации по контролю состояния системы, ее пользователей, потреблению ресурсов и проведению анализа защищенности, которые рекомендуется регулярно производить с целью выявления нетипичного поведения системы, которое может свидетельствовать о признаках подозрительной активности. А это, в свою очередь, может быть признаком того, что система скомпрометирована. Для достижения приемлемого критерия безопасности рекомендуется использовать сведения, приведенные в документе «*Закупочные сервисы iSource. Руководство администратора. Приложение А. Безопасность в ОС Ubuntu.*» *is000-AGD_PRE.1*.

8.11.1 Контроль ресурсов системы

В защищенной системе крайне важно отслеживать потребление ресурсов. Любое бесконтрольное потребление ресурсов может, во-первых, негативно влиять на систему с точки зрения выполнения

функций по назначению, во-вторых, с точки зрения ИБ может приводить к реализации атак типа «отказ в обслуживании» и, в третьих, может свидетельствовать о подозрительной активности.

В составе ОС Ubuntu Linux можно использовать несколько популярных инструментов для отслеживания потребления ресурсов. В данном разделе не освещаются средства контроля функционирования, например, такие как `Zabbix` (хотя его использование представляется чрезвычайно разумным) или его аналоги. Также не освещаются интерфейсы, такие как `/proc/meminfo`, `/proc/cpuinfo`, `free`, `top` или `ps`, ввиду того, что знание об их использовании предполагается обязательным для персонала, обслуживающего систему. В настоящем разделе описывается использование ПО `sysstat` и некоторых других утилит контроля.

8.11.1.1 Использование инструментов `sysstat`

В состав утилит мониторинга sysstat входят следующие программные средства:

- `sar` – сбор информации об активности в системе;
- `iostat` – отчеты об использовании ЦП и статистика операций ввода- вывода;
- `mpstat` – глобальная статистика и отчеты по каждому процессу;
- `sadf` – отображение информации от `sar` в различных форматах;
- `nfsiostat` – статистика операций ввода-вывода для NFS;
- `cifsiostat` – статистика операций ввода-вывода для SMB/CIFS.

Для установки инструментов мониторинга `sysstat` выполнить (в контексте полномочий суперпользователя `root`):

```
# apt-get install sysstat
```

Листинг 231: Пример установки утилит `sysstat`

После установки набора утилит мониторинга `sysstat`, входящая в её состав утилита `sar` обеспечивает:

- предоставление статистики использования ЦП;
- предоставление статистики использования ЦП по отдельным процессам или пользователям;
- предоставление статистики использования памяти (ОЗУ);
- предоставление статистики использования swap (подкачки);
- предоставление статистики использования операций ввода-вывода;
- предоставление статистики использования переключения контекстов;

- предоставление статистики сетевой активности;
- предоставление сводных данных за указанный период времени.

Для запуска сбора статистики состояния системы требуется настроить конфигурационный файл `/etc/default/sysstat` согласно директиве ниже и запустить службу `sysstat`:

```
ENABLED="true"
```

Листинг 232: Пример активизации сбора статистики в `/etc/default/sysstat`

```
# systemctl enable sysstat
# systemctl start sysstat
```

Листинг 233: Пример запуска `sysstat`

Для ежечасного сбора статистики выполнить конфигурацию планировщика ОС:

```
# ln -s /usr/lib/sysstat/debian-sa1 /etc/cron.hourly/sysstat_hourly
# systemctl restart cron
```

Листинг 234: Пример конфигурирования планировщика на ежечасный сбор статитстики

Для просмотра[26] статистики используется `sar` с аргументами (по всем процессорам, 10 проходов с интервалом в 5 секунд):

```
# sudo sar -u ALL 5 10
Linux 5.10.0-19-amd64 (dl5310)  22.11.2022  _x86_64_  (4 CPU)

CPU      %usr     %nice      %sys   %iowait    %steal      %irq     %soft    %guest    %gnice     %idle
all      0,85      0,00      0,95      0,50      0,00      0,00      0,00      0,00      0,00     97,69
all      4,98      0,00      2,82      0,25      0,00      0,00      0,10      0,00      0,00     91,85
all      1,46      0,00      1,81      0,00      0,00      0,00      0,10      0,00      0,00     96,63
```

Листинг 235: Пример просмотра статистики утилизации по всем процессорам

Для просмотра данных утилизации по конкретному процессору требуется использовать ключ «-P», при этом процессоры нумеруются с нуля (первый процессор, три прохода каждые две секунды):

```
# sudo sar -P 0 2 3
Linux 5.10.0-19-amd64 (dl5310)  22.11.2022  _x86_64_  (4 CPU)

12:12:05        CPU     %user     %nice   %system   %iowait    %steal     %idle
12:12:07          0      3,50      0,00      1,00      0,00      0,00     95,50
12:12:09          0      5,58      0,00      3,05      0,00      0,00     91,37
12:12:11          0      2,02      0,00      4,04      0,00      0,00     93,94
Среднее:      0      3,70      0,00      2,69      0,00      0,00     93,61
```

Листинг 236: Пример просмотра статистики утилизации по заданному процессору

8.11.1.2 Контроль активности пользователей

Для контроля активности пользователей и процессов рекомендуется применять набор утилит `acct`. В составе этого набора следующие средства:

[26] Для просмотра статистики потребуются полномочия учетной записи `root`.

- `ac` – выводит статистику о времени подключения пользователей, на основании входов в систему и выходов из нее, беря информацию из файла `/var/log/wtmp`. А также может подводить итоговое время по дням (опция `-d`) и по пользователям (опция `-p`);
- `accton` – применяется для включения и выключения учета процессов;
- `last` – обрабатывает файл `/var/log/wtmp` и выводит статистику о времени вхождения пользователей в систему;
- `sa` – подводит итоги исполнения команд, работы с портами ввода-вывода, загрузки процессора в соответствии с информацией в файле учета процессов `/var/account/pacct`;
- `lastcomm` выводит информацию об исполненных командах в соответствии с файлом `/var/account/pacct`.

Для установки набора утилит acct необходимо выполнить (находясь в контексте учетной записи `root`):

```
# apt-get install acct
```

Листинг 237: Пример установки утилит `acct`

Для запуска сбора статистики состояния системы требуется настроить конфигурационный файл `/etc/default/acct` согласно директивам ниже:

```
ACCT_ENABLE="1"
ACCT_LOGGING="30"
```

Листинг 238: Пример активизации сбора статистики в `/etc/default/acct`

Где первая переменная активизирует сбор статистики по процессам и пользователям, а вторая переменная определяет время (в сутках) для хранения информации до ее перезаписи.

Команда `ac` без указания какого-либо аргумента будет отображать общую статистику времени нахождения в системе в часах для текущего пользователя, на основе данных из `/var/log/wtmp`:

```
$ ac
total      955.21
```

Листинг 239: Пример использования `ac` для текущего пользователя

Для заданного пользователя можно указать в аргументе его имя:

```
# ac consta
total      955.24
# ac root
total        0.00
```

Листинг 240: Пример использования `ac` для заданного пользователя

Для получения сводки выполнения команд и процессов, можно выполнить:

```
# sa
53823    9432.83re      60.89cp         0avio      6069k
63       32.52re       21.60cp          0avio    105118k   xelatex
3        28.75re       14.21cp         0avio      9758k   aide
63       25.40re       13.94cp          0avio     11387k   xdvipdfmx
25     1496.04re        2.92cp          0avio    614016k   IPC I/O Child
...
<листинг опущен>
```

Листинг 241: Пример использования `sa`

Где:

- `32.52re` – «реальное время», в минутах;
- `21.60cp` – сумма использованного времени процессора, в минутах;
- `105118k` – усредненное значение тактов процессорного ядра (ядер), занятых на выполнение задачи, в тысячах тактов;
- `xelatex` – выполняемая команда или процесс.

```
# sa --print-users
root       0.00 cpu      613k mem      0 io hostname
root       0.00 cpu     1358k mem      0 io date
root       0.00 cpu     1604k mem      0 io grep
root       0.00 cpu     1468k mem      0 io head
root       0.00 cpu     1343k mem      0 io cut
root       0.00 cpu     1705k mem      0 io aide
...
consta     0.00 cpu      564k mem      0 io false
consta     0.00 cpu     2352k mem      0 io dbus-daemon    *
```

Листинг 242: Пример использования `sa` с сортировкой по ресурсам и пользователям

Для просмотра команд и процессов пользователя, выполненных на конкретном терминале, можно выполнить:

```
# lastcomm --strict-match --user consta --tty pts/1
powerline              consta   pts/1      0.00 secs Wed Nov 23 17:33
powerline              consta   pts/1      0.00 secs Wed Nov 23 17:33
powerline              consta   pts/1      0.00 secs Wed Nov 23 17:33
bash              F    consta   pts/1      0.00 secs Wed Nov 23 17:33
wc                     consta   pts/1      0.00 secs Wed Nov 23 17:33
bash              F    consta   pts/1      0.00 secs Wed Nov 23 17:33
man                    consta   pts/1      0.03 secs Wed Nov 23 17:33
pager                  consta   pts/1      0.00 secs Wed Nov 23 17:33
man               F    consta   pts/1      0.00 secs Wed Nov 23 17:33
nroff                  consta   pts/1      0.00 secs Wed Nov 23 17:33
```

Листинг 243: Пример использования `lastcomm` с сортировкой по терминалу и пользователю

8.11.2 Использование сканера аудита безопасности Lynis

Сканер безопасности Lynis https://cisofy.com/lynis/ предназначен для проверки системы по требованиям безопасности международного стандарта PCI-DSS. Данный стандарт широко признан в международном сообществе и соответствие его требованиям необходимо при использовании в финансовой и банковской сфере.

Необходимо отметить, что сам стандарт PCI-DSS прежде всего ориентирован на информационную систему в целом. Однако для операционной системы он предполагает свыше двухсот контролей различных параметров, большинство из которых описано в настоящем документе.

Сканер безопасности Lynis имеет две редакции, общедоступную бесплатную версию, ориентированную только на аудит, и коммерческую платную, с большим количеством подключаемых расширений, ориентированную не только на аудит, а еще и на проведение настроек, направленных на достижение критерия безопасности. В целом, сканер Lynis направлен на современные UNIX (Solaris, Mac OS X и др.) и Linux (Ubuntu, Red Hat, SUSE др.) системы, а также совместимые с ними (различные системы диалекта BSD). К плюсам сканера можно отнести его очень высокую универсальность и простоту. По сути он представляет собой сценарий оболочки и не требует установки каких- то серьезных зависимостей. Он может быть выполнен на значительном количестве архитектур и систем. Существует и переносимая версия, не требующая компиляции.

Для установки актуальной версии сканера безопасности Lynis (бесплатная версия с открытым исходным кодом) требуется выполнить подключение актуального репозитория разработчика сканера. Для этого от имени пользователя `root` выполнить:

```
# wget -O - https://packages.cisofy.com/keys/cisofy-software-public.key | apt-key add -
# echo "deb https://packages.cisofy.com/community/lynis/deb/ stable main" | tee
/etc/apt/sources.list.d/cisofy-lynis.list
# apt install lynis
```

Листинг 244: Пример установки сканера `lynis`

Для запуска сканера безопасности Lynis и получения отчета выполнить:

```
# lynis audit system
```

Листинг 245: Пример запуска сканера `lynis`

Сканер в отчете отображает примерный процент соответствия стандарту PCI-DSS, выводит предупреждения и рекомендации. Хорошим результатом будет достижения соответствия свыше 80 баллов и отсутствие предупреждений.

Образец вывода приведен ниже:

```
... <листинг опущен> ...

-[ Lynis 3.0.8 Results ]-

Great, no warnings

Suggestions (14):
```

```
----------------------------

... <листинг опущен> ...

================================================================================

Lynis security scan details:

Hardening index : 89 [#################   ]
Tests performed : 269
Plugins enabled : 0

Components:
- Firewall               [V]
- Malware scanner        [V]

Scan mode:
Normal [V]  Forensics [ ]  Integration [ ]  Pentest [ ]

Lynis modules:
- Compliance status      [?]
- Security audit         [V]
- Vulnerability scan     [V]

Files:
- Test and debug information      : /var/log/lynis.log
- Report data                     : /var/log/lynis-report.dat

================================================================================
```

Листинг 246: Пример отчета сканера `lynis`

8.11.3 Анализ уязвимостей в среде выполнения

Для анализа уязвимостей в среде выполнения (в ОС) рекомендуется использовать информацию, предоставляемую разработчиком ОС Ubuntu 20.04 и сканер `openscap`, которые используют методологию и язык OVAL. Информация о наличии уязвимостей ежедневно открыто предоставляется в виде файла на языке OVAL самим разработчиком. А сканер входит в состав штатных пакетов и может быть установлен из репозиториев.

Для установки сканера `openscap` выполнить:

```
# apt install libopenscap8
```

Листинг 247: Пример установки `openscap`

Для получения информации разработчика ОС Ubuntu 20.04 (Canonical), содержащую сведения об уязвимостях, необходимо получить файл с их описанием. Для этого выполнить[27]:

```
# wget https://security-metadata.canonical.com/oval/com.ubuntu.focal.usn.oval.xml.bz2
--2022-11-05 23:37:36--  https://security-metadata.canonical.com/oval/com.ubuntu.focal.usn.oval.xml.
    bz2
```

[27] Потребуется `wget` или `curl`.

```
Распознаётся security-metadata.canonical.com (security-metadata.canonical.com)… 185.125.190.29,
    185.125.190.21, 185.125.190.20, ...
Подключение к security-metadata.canonical.com (security-metadata.canonical.com)
    |185.125.190.29|:443 ... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа… 200 OK
Длина: 306980 (300K) [application/x-bzip2]
Сохранение в: «com.ubuntu.focal.usn.oval.xml.»bz2

com.ubuntu.focal.usn.oval.xml.bz2
    100%[=============================================================================>] 299,79K  1,37MB
    /s    за 0,2s

2022-11-05 23:37:37 (1,37 MB/s) - «com.ubuntu.focal.usn.oval.xml.»bz2 сохранён [306980/306980]
```

Листинг 248: Пример получения информации об уязвимостях вендора ОС Ubuntu

Распаковать файл с описаниями:

```
# bunzip2 com.ubuntu.focal.usn.oval.xml.bz2
```

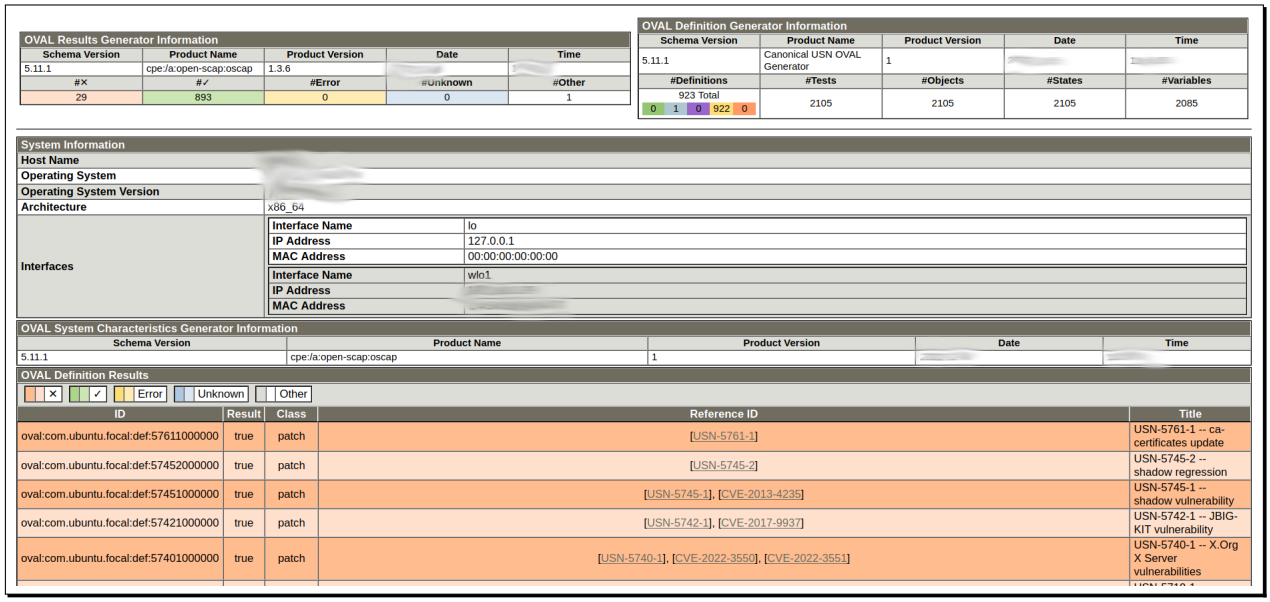
Листинг 249: Распаковка файла с описаниями OVAL

Затем запустить сканер, передав ему аргументом файл с описаниями:

```
# oscap oval eval --report report.html ./com.ubuntu.focal.usn.oval.xml
...
Definition oval:com.ubuntu.focal:def:43332000000: false
Definition oval:com.ubuntu.focal:def:43322000000: false
Definition oval:com.ubuntu.focal:def:43302000000: false
Definition oval:com.ubuntu.focal:def:41716000000: false
Definition oval:com.ubuntu.focal:def:100: false
Evaluation done.
```

Листинг 250: Пример запуска и получения отчета сканера `openscap`

Затем можно открыть получившийся файл отчета (в формате html) и просмотреть текущее состояние и статус пакетов с обновлениями и установленными пакетами. Образцы отчетов для системы, нуждающейся в обновлении, и для системы, программное обеспечение которой находится в актуальном состоянии, приведены на рисунках Рисунок 64 – Рисунок 65 соответственно.



OVAL Results Generator Information

Schema Version	Product Name	Product Version	Date	Time
5.11.1	cpe:/a:open-scap:oscap	1.3.6	[redacted]	[redacted]

#×	#✓	#Error	#Unknown	#Other
29	893	0	0	1

OVAL Definition Generator Information

Schema Version	Product Name	Product Version	Date	Time
5.11.1	Canonical USN OVAL Generator	1	[redacted]	[redacted]

#Definitions	#Tests	#Objects	#States	#Variables
923 Total: 0 1 0 922 0	2105	2105	2105	2085

System Information

Host Name	[redacted]
Operating System	[redacted]
Operating System Version	[redacted]
Architecture	x86_64
Interfaces	Interface Name: lo; IP Address: 127.0.0.1; MAC Address: 00:00:00:00:00:00
	Interface Name: wlo1; IP Address: [redacted]; MAC Address: [redacted]

OVAL System Characteristics Generator Information

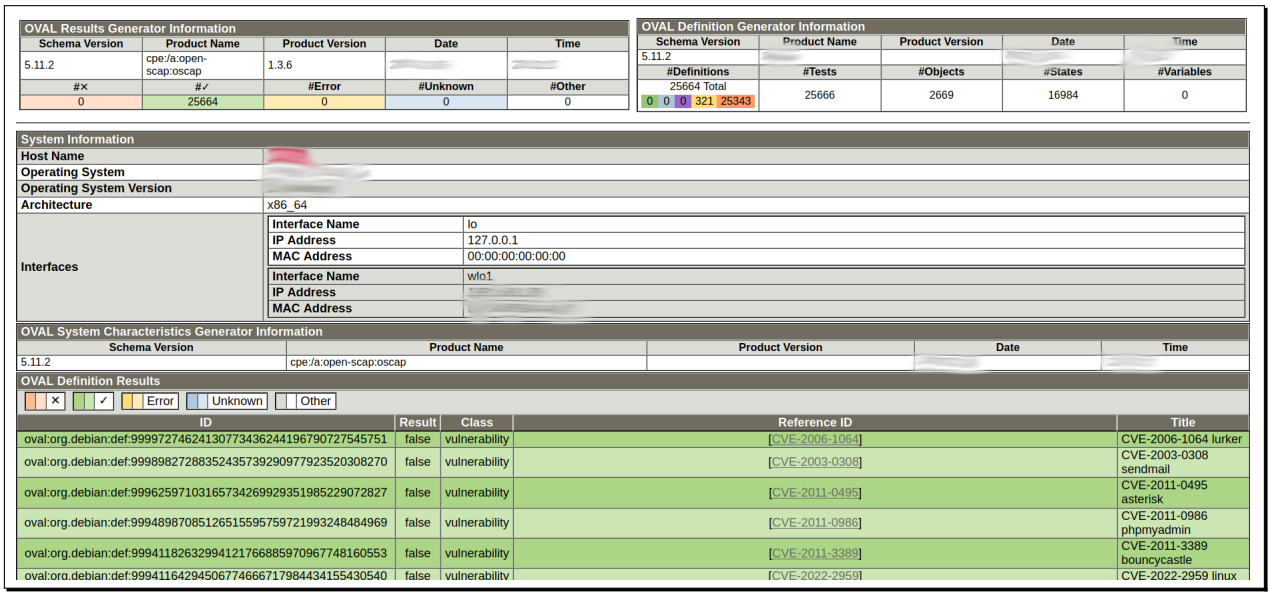
Schema Version	Product Name	Product Version	Date	Time
5.11.1	cpe:/a:open-scap:oscap	1	[redacted]	[redacted]

OVAL Definition Results

✕ ✓ Error Unknown Other

ID	Result	Class	Reference ID	Title
oval:com.ubuntu.focal:def:57611000000	true	patch	[USN-5761-1]	USN-5761-1 -- ca-certificates update
oval:com.ubuntu.focal:def:57452000000	true	patch	[USN-5745-2]	USN-5745-2 -- shadow regression
oval:com.ubuntu.focal:def:57451000000	true	patch	[USN-5745-1], [CVE-2013-4235]	USN-5745-1 -- shadow vulnerability
oval:com.ubuntu.focal:def:57421000000	true	patch	[USN-5742-1], [CVE-2017-9937]	USN-5742-1 -- JBIG-KIT vulnerability
oval:com.ubuntu.focal:def:57401000000	true	patch	[USN-5740-1], [CVE-2022-3550], [CVE-2022-3551]	USN-5740-1 -- X.Org X Server vulnerabilities

Рисунок 64: Пример отчета `openscap` для системы, нуждающейся в обновлении



OVAL Results Generator Information

Schema Version	Product Name	Product Version	Date	Time
5.11.2	cpe:/a:open-scap:oscap	1.3.6	[redacted]	[redacted]

#×	#✓	#Error	#Unknown	#Other
0	25664	0	0	0

OVAL Definition Generator Information

Schema Version	Product Name	Product Version	Date	Time
5.11.2	[redacted]		[redacted]	[redacted]

#Definitions	#Tests	#Objects	#States	#Variables
25664 Total: 0 0 0 321 25343	25666	2669	16984	0

System Information

Host Name	[redacted]
Operating System	[redacted]
Operating System Version	[redacted]
Architecture	x86_64
Interfaces	Interface Name: lo; IP Address: 127.0.0.1; MAC Address: 00:00:00:00:00:00
	Interface Name: wlo1; IP Address: [redacted]; MAC Address: [redacted]

OVAL System Characteristics Generator Information

Schema Version	Product Name	Product Version	Date	Time
5.11.2	cpe:/a:open-scap:oscap		[redacted]	[redacted]

OVAL Definition Results

✕ ✓ Error Unknown Other

ID	Result	Class	Reference ID	Title
oval:org.debian:def:99997274624130773436244196790727545751	false	vulnerability	[CVE-2006-1064]	CVE-2006-1064 lurker
oval:org.debian:def:99989827288352435739290977923520308270	false	vulnerability	[CVE-2003-0308]	CVE-2003-0308 sendmail
oval:org.debian:def:99962597103165734269929351985229072827	false	vulnerability	[CVE-2011-0495]	CVE-2011-0495 asterisk
oval:org.debian:def:99948987085126515595759721993248484969	false	vulnerability	[CVE-2011-0986]	CVE-2011-0986 phpmyadmin
oval:org.debian:def:99941182632994121766885970967748160553	false	vulnerability	[CVE-2011-3389]	CVE-2011-3389 bouncycastle
oval:org.debian:def:99941164294506774666717984434155430540	false	vulnerability	[CVE-2022-2959]	CVE-2022-2959 linux

Рисунок 65: Пример отчета `openscap` для системы в актуальном состоянии

8.12 Очистка данных и затруднение их восстановления

В составе ОС Linux может использоваться несколько программных средств для очистки остаточной информации. Часть из них работает в интерфейсе командной строки, некоторые имеют графический интерфейс. В данном разделе будут приведены описания наиболее популярных из них.

Для безопасного стирания данных разумно использовать специализированный набор утилит из пакета `secure-delete`. Он обеспечивает работу с интерфейсом оболочки в составе следующих программ:

- `srm` – утилита удаления файлов, используется для стирания содержимого файлов;

- `sfill` – утилита для удаления содержимого инодов (мета-данных) и свободного места;
- `sswap` – утилита для удаления содержимого раздела подкачки;
- `sdmem` – утилита для удаления содержимого ОЗУ.

Все указанные выше утилиты имеют общий синтаксис аргументов, где `f` - быстрый (fast) режим с наименьшей безопасностью, `z` - режим последнего прохода с использованием специальной битовой последовательностью (стирание нолями). В других случаях в качестве источника энтропии используется псевдоустройство `/dev/urandom`. Для установки утилит выполнить:

```
# apt-get install secure-delete
```

Листинг 251: Пример установки утилит `secure-delete`

Помимо этого, возможно использовать специальное расширение к проводнику Gnome – пакет `nautilus-wipe`. Такое решение не предоставляет возможностей стирать содержимое ОЗУ и раздела подкачки, однако предлагает стереть незанятое место или отдельный файл, причем с выбором опций по желанию оператора, который может как выбирать количество проходов, так и комбинировать специальные и случайные битовые последовательности. Для установки `nautilus-wipe` нужно выполнить:

```
# apt-get install nautilus-wipe
```

Листинг 252: Пример установки утилиты `nautilus-wipe`

Затем требуется перезапустить графическую сессию Gnome.

8.13 Рекомендации по защите `systemd`

Взломав или скомпрометировав какую-нибудь службу, работающую под управлением `systemd` злоумышленник может получить полномочия взломанной службы в операционной системе. Поскольку некоторые службы работают от имени `root`, то злоумышленник может получить любые полномочия, вплоть до `root`.

Настоятельно рекомендуется использовать инструменты изоляции, которые предоставляет служба `systemd` *для усиления безопасности операционной системы и работающих в ней служб и сервисов.*

8.13.1 Общие сведения о механизмах безопасности, предоставляемых службой `systemd`

Служба `systemd` является первичной службой инициализации операционной системы и прародителем всех остальных пользовательских процессов в операционной системе. Поэтому она и её безопасность представляются критически важными. Особенно с учетом того, что служба `systemd` работает в пространстве пользователя и представляет пользователям широкий набор интерфейсов для конфигурации, создания, обслуживания и поддержки собственных сервисов.

Поддержка в безопасном состоянии служб `systemd` – это постоянная и кропотливая работа для администратора и разработчиков системы. Работа по постоянному совершенствованию безопасности служб `systemd` должна проводится ими непрерывно, активно и в любых условиях.

К счастью, служба `systemd` имеет широчайшие возможности по использованию механизмов безопасности ядра операционной системы. К таким механизмам можно отнести:

- изоляцию на уровне файловой системы, т.н. «песочница» (Filesystem namespaces);
- изоляцию на уровне пользователя ОС (User namespaces);
- разграничение доступа с помощью «возможностей» (Capabilities);
- фильтр системных вызовов с помощью `seccomp` и BPF.

В Таблице 15 перечислены наиболее важные опции[28], которые можно использовать при конфигурации служб, управляемых с помощью `systemd`, их рекомендуемые[29] значения и описание:

Опция и рекомендуемое значение	**Описание**
`RestrictNamespaces=true`	Запрет на создание собственных изолированных пространств
`LockPersonality=true`	Блокировка системного вызова `personality()`
`NoNewPrivileges=true`	Служба не получит новых привилегий, кроме заданных
`ProtectKernelModules=true`	Службе запрещена загрузка модулей ядра
`SystemCallArchitectures=native`	Службе разрешено выполнять системные вызовы только той архитектуры, что и ядро ОС
`ProtectHostname=true`	Запрет для службы на изменение имени узла
`RestrictAddressFamilies=AF_INET`	Разрешено использовать только адреса протокола IPv4
`RestrictAddressFamilies=AF_INET6`	Разрешено использовать только адреса протокола IPv6
`RestrictAddressFamilies=AF_UNIX`	Разрешено использовать только сокеты UNIX
`RestrictRealtime=true`	Службе запрещено использовать возможности ядра, связанные с работой в реальном времени (низкими задержками)
`ProtectControlGroups=true`	Запрет на работу с контрольными группами (`cgroups`)
`ProtectKernelTunables=true`	Службе запрещено изменять параметры ядра ОС (`sysctl`)
`RestrictSUIDSGID=true`	Службе запрещено создавать файлы с битами SUID/SGID
`ProtectClock=true`	Службе запрещено изменять системное время
`ProtectSystem=strict`	Службе запрещена запись в любое место файловой иерархии
`PrivateDevices=true`	Служба может использовать только псевдо-устройства, такие как `/dev/zero`, `/dev/random`, `/dev/null` и т.п. И каждое такое устройство будет создано для неё в собственном изолированном пространстве.
`PrivateTmp=true`	Службе будет создан изолированный каталог `/tmp`
`ProtectKernelLogs=true`	Служба не получит доступа к файлам аудита ядра ОС
`ProtectProc=invisible`	Служба не получит информации о чужих процессах
`PrivateUsers=true`	Служба не получит доступа к данным других пользователей

[28] Более подробно это описано в разделе 8.13.3.

[29] Всегда правильные рекомендации дать в общем случае для данных опций крайне сложно. Ввиду того, что оценивать нужно не только сам настраиваемый сервис, а ещё и его потомков.

Опция и рекомендуемое значение	Описание
`ProtectHome=true`	Служба не получит доступа к домашним каталогам
`UMask=0077`	Значение UMASK для файлов, создаваемых службой

Таблица 15: Полезные опции усиления безопасности для служб, управляемых с помощью `systemd`.

Для того, чтобы проверить безопасность служб `systemd`, можно выполнить команду:

```
# systemd-analyze security
```

Листинг 253: Команда для ценки состояния безопасности служб `systemd`

После этого можно понять, какие службы и сервисы в системе и насколько безопасны. Результат может не обрадовать, см. Рисунок 66:

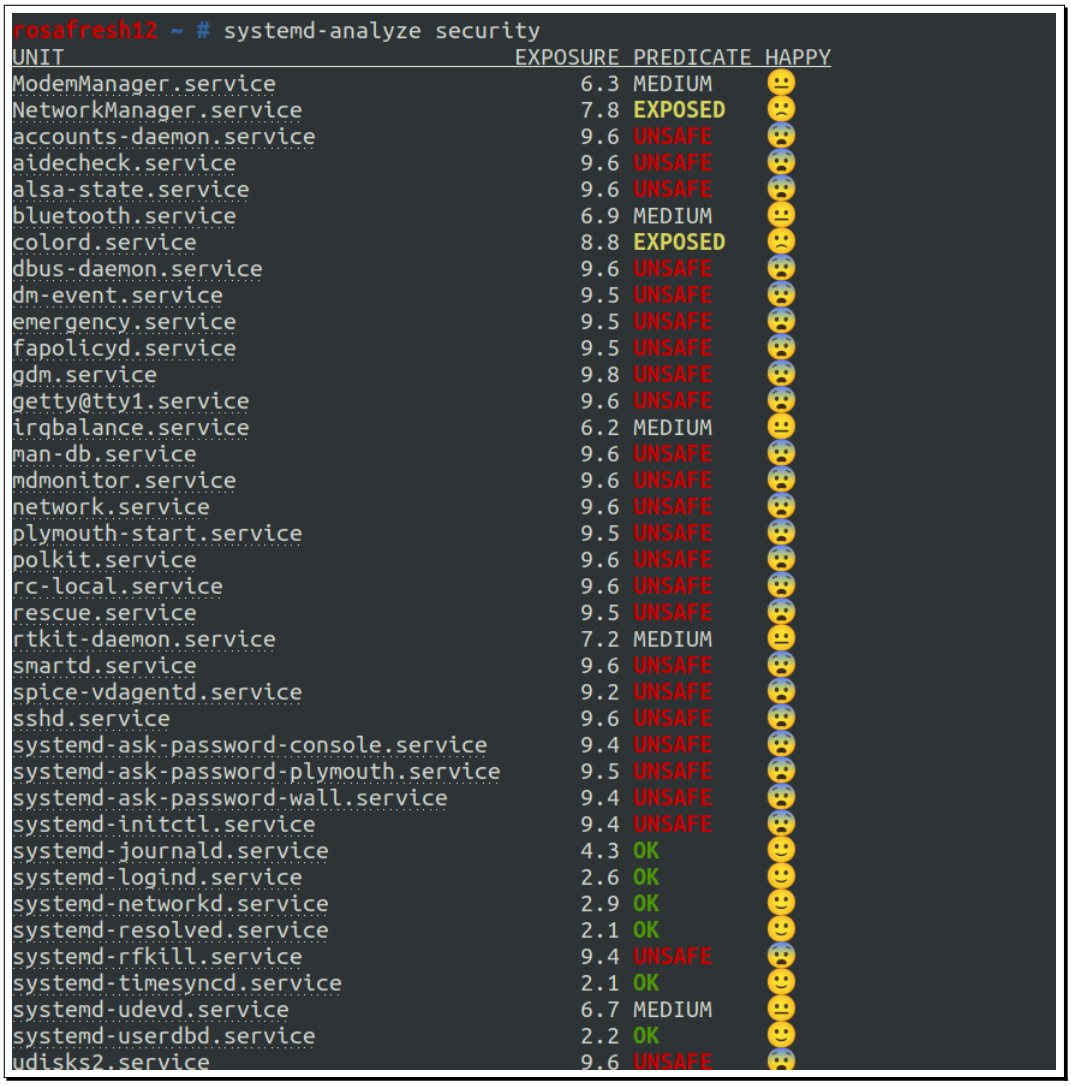


Рисунок 66: Пример вывода `systemd-analyze security`

8.13.2 Как с помощью `systemd` повысить безопасность `sshd`

Для примера можно попробовать усилить защиту какого-нибудь важного и нужного сервиса, например `sshd`. Тем более следует помнить о том, что `sshd` запускает дочерние процессы, если вход

пользователя успешен, формирует его окружение. Кроме того, `sshd` имеет доступ к файлам ключей, должен иметь доступ к службе аудита, чтобы можно было регистрировать разные события связанные со входом, выходом, попытками аутентификации, операциями обмена ключами и т.п. Опять-таки, `sshd` может взаимодействовать с модулями PAM, должен иметь доступ к сетевым устройствам для организации сетевых подключений и сокетов. Таким образом, с одной стороны служба `sshd` должна иметь довольно широкий доступ к системе, на которой она запущена. С другой стороны, имея такой доступ, в случае её компрометации, злоумышленник тоже может получить массу прав.

Кое-что по ограничению доступа службы `sshd` к данным всей системы и увеличению её безопасности сделать все же можно. Можно попытаться улучшить безопасность `sshd` хотя бы до среднего уровня.

Для начала нужно детально определить текущее состояние службы `sshd`. Это можно сделать с помощью команды:

```
# systemd-analyze security sshd
```

Листинг 254: Команда оценки состояния безопасности службы `sshd`

В ответ `systemd` выдает детали. Как видно, защита для `sshd` практически не активизирована, см. Рисунок 67:

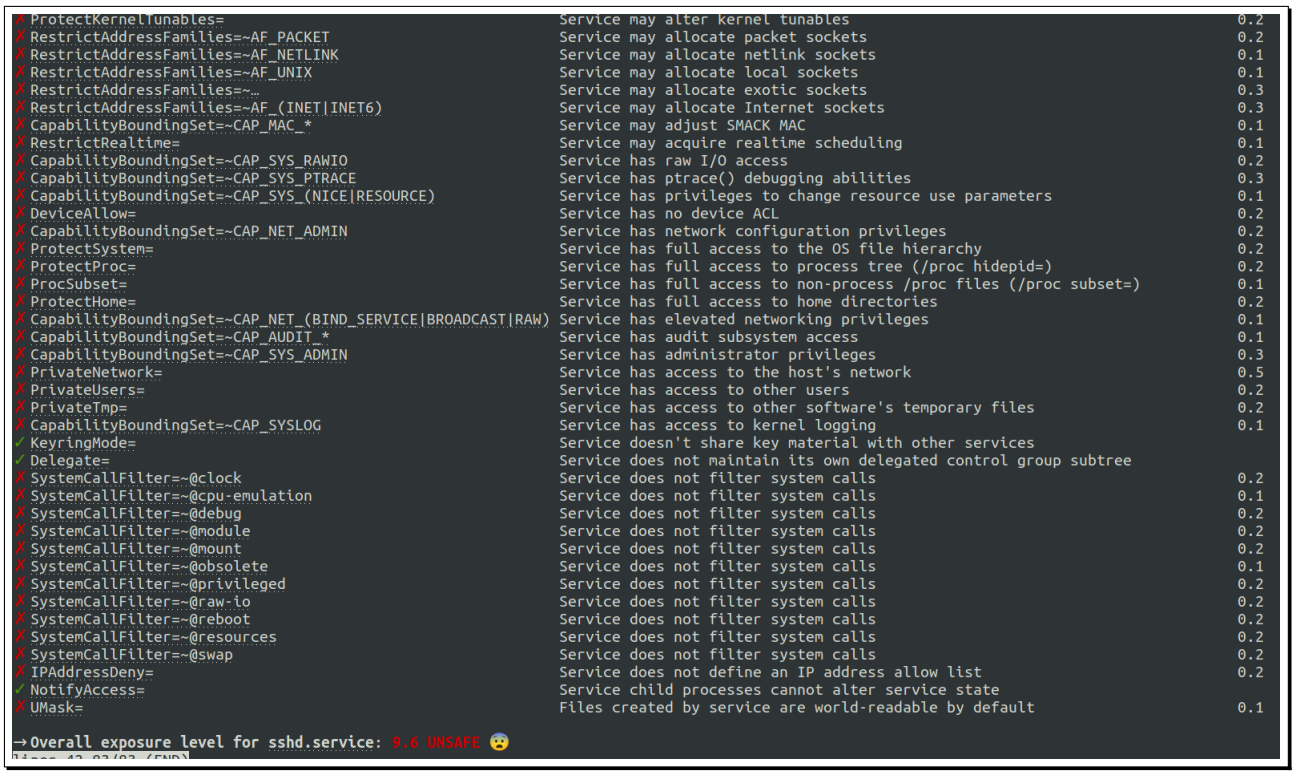


Рисунок 67: Пример вывода `systemd-analyze security sshd`

Затем можно внести изменения в его конфигурацию с помощью такой команды:

```
# systemctl edit sshd
```

Листинг 255: Команда редактирования параметров службы `sshd`

В окне редактирования есть подсказка, указывающая в какое именно место файла нужно вносить изменения.

Сперва можно ограничить для `sshd` самые очевидные вещи – попробуем надежнее изолировать её (надо помнить, что это же будет касаться и потомков, порожденных службой `sshd`):

- Службе `sshd` запрещается использовать системный вызов `personality()`;
- Службе `sshd` запрещается загружать модули ядра;
- Службе `sshd` запрещается менять `hostname`;
- Службе `sshd` запрещается работа в режиме реального времени;
- Службе `sshd` запрещается воздействовать на переменные ядра ОС;
- Службе `sshd` запрещается изменять системное время;
- Службе `sshd` будут созданы изолированные устройства `/dev/zero`, `/dev/random`, `/dev/null` и т.п.;
- Службе `sshd` будет создан изолированный каталог `/tmp`;
- Службе `sshd` будет установлено новое значение UMASK, равное `0077`;

При редактировании конфигурационных параметров службы `sshd` это будет выглядеть так:

```
### Editing /etc/systemd/system/sshd.service.d/override.conf
### Anything between here and the comment below will become the new contents of the file

[Service]
LockPersonality=true
ProtectKernelModules=true
ProtectHostname=true
RestrictRealtime=true
ProtectKernelTunables=true
ProtectClock=true
PrivateDevices=true
PrivateTmp=true
UMask=0077
### Lines below this comment will be discarded
....
Дальнейшее содержимое опущено
```

Листинг 256: Пример редактирования параметров службы `sshd`

Затем можно вновь проверить состояние безопасности и перезагрузить `sshd`:

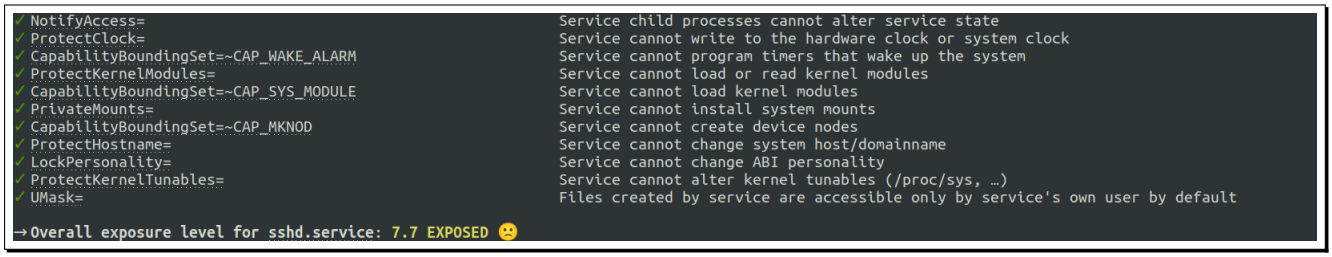


Рисунок 68: Пример вывода `systemd-analyze security sshd` после изоляции

Как видно на Рисунке 68 – прогресс налицо. Теперь можно подумать о том, что еще добавить, так, чтобы с одной стороны увеличить безопасность, а с другой – не сильно повлиять на работу пользователей, подключающихся по `ssh`. Дальнейшие ограничения могут выглядеть так:

- Службе `sshd` разрешено взаимодействовать только с ФС типа `ext4` и `tmpfs`;
- Службе `sshd` запрещено изменять значение приоритета;
- Службе `sshd` установлено нулевое (среднее) значение приоритета;
- Службе `sshd` разрешается выполнять системный вызовы только той архитектуры, что и ядро;
- Службе `sshd` запрещено создавать страницы памяти, которые одновременно будут доступны на запись и выполнение;
- Службе `sshd` запрещается изменять системное время;
- Запрещается трассировка `sshd`;
- Для службы `sshd` в файловой системе `/proc` делаются невидимыми все файлы и каталоги, не связанные напрямую с управлением процессами;
- Службе `sshd` устанавливается запрет на применение устаревших и не используемых системных вызовов;
- Службе `sshd` устанавливается доступ только на чтение в служебную ФС `/sys/fs/cgroup/`, это отключит для нее возможности взаимодействия с механизмом контрольных групп (Linux Control Groups (`cgroups(7)`).

В конфигурацию добавляются опции, а общее состояние конфигурации будет теперь выглядеть так:

```
### Editing /etc/systemd/system/sshd.service.d/override.conf
### Anything between here and the comment below will become the new contents of the file

[Service]
LockPersonality=true
ProtectKernelModules=true
ProtectHostname=true
```

```
RestrictRealtime=true
ProtectKernelTunables=true
ProtectClock=true
PrivateDevices=true
PrivateTmp=true
UMask=0077
RestrictFileSystems=tmpfs ext4
LimitNICE=0
Nice=0
SystemCallArchitectures=native
MemoryDenyWriteExecute=true
ProtectProc=ptraceable
ProcSubset=pid
SystemCallFilter=!@obsolete
ProtectControlGroups=true

### Lines below this comment will be discarded
.....
```

Листинг 257: Пример конфигурации параметров безопасности службы `sshd` с помощью `systemd`

Еще раз проверить состояние после добавления дополнительных параметров безопасности и перезапустить службу. Состояние безопасности выведено на приемлемый (для такой большой и важной службы, конечно, как `sshd`) уровень.

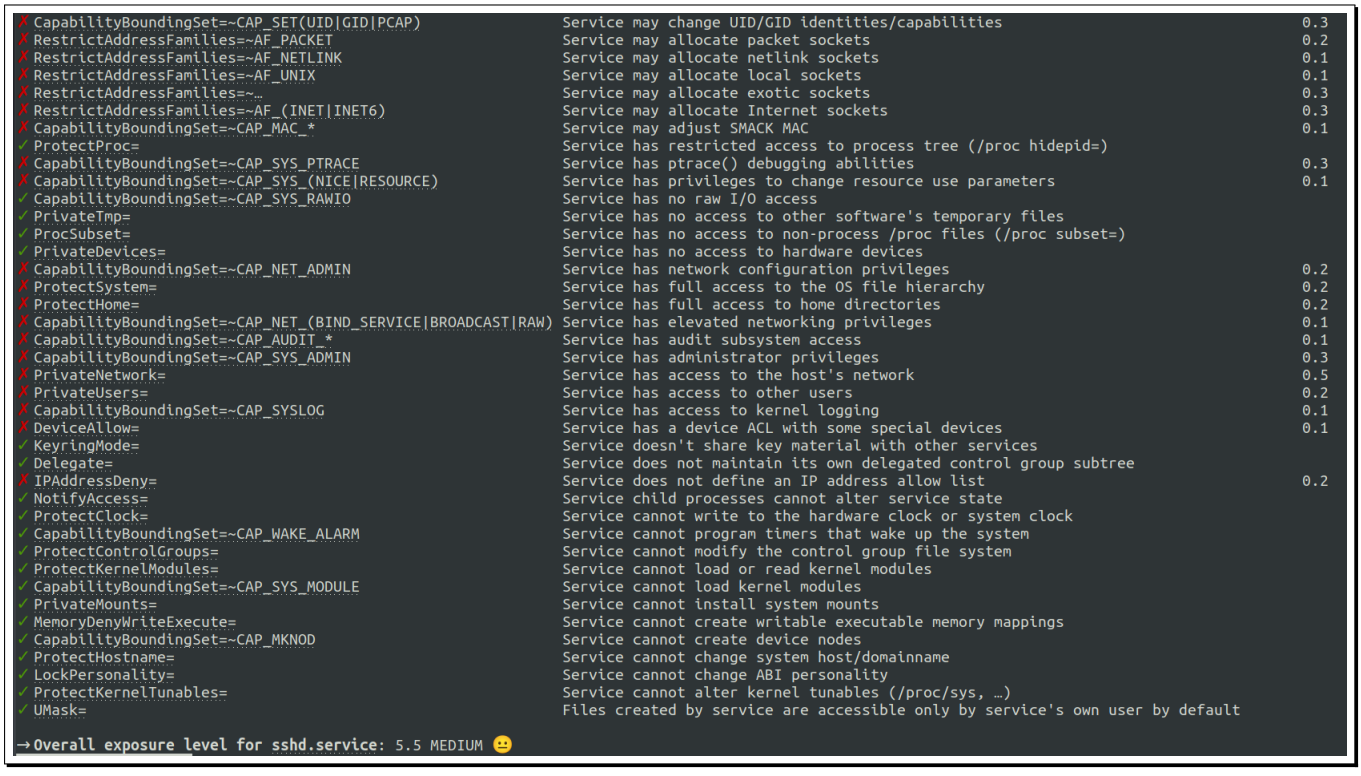


Рисунок 69: Пример вывода `systemd-analyze security sshd` после применения всех заданных настроек

8.13.3 Краткий справочник опций безопасности, предоставляемых службой `systemd`

Как видно, служба `systemd` обладает широчайшими возможностями ограничивать те процессы, службы и сервисы, которыми она управляет. Ниже приведена краткая справка по некоторым наибо-

лее важным опциям настройки параметров systemd для служб, в добавление к той информации, что приведена в Таблице 15. Более подробно данная информация приведена в разделе интерактивной справки systemd.exec(5).

8.13.3.1 Изоляция процесса в systemd на уровне пространства ФС и доступа к данным

Первая группа опций посвящена изоляции управляемого процесса от остальной системы или от данных выполняющихся соседних процессов.

`ExecSearchPath=`
Изоляция пути запуска.

Параметр предназначен для явного указания пути, по которому будет осуществлен запуск. При этом, этот параметр отменит пути поиска запуска, заданные в переменной окружения $PATH, так как приоритет его выше. Пути можно перечислять через двоеточие.

`WorkingDirectory=`
Изоляция пути выполнения.

Путь корневого рабочего каталога для службы, то есть устанавливает для выполняемого процесса каталог, где он будет выполнятся. Если путь не указан или «-», то будет использоваться домашний каталог того пользователя, от чьего имени служба выполняется (см. User=).

`RootDirectory=`
Изоляция пути корневого каталога.

Определяет путь корня файловой системы для службы (выполняемого процесса) с помощью системного вызова chroot(). Нужно обращать внимание, что если этот параметр задан, то все данные, и всё, с чем служба (процесс) взаимодействует, должны быть а пределах указанного пути.

`ProtectProc=`
Изоляция при обмене данными.

Параметр может принимать 4 аргумента: noaccess, invisible, ptraceable и default. Последний применяется по умолчанию, если не задан иной аргумент. Фактически, параметр контролирует состояние флага монтирования hidepid= файловой системы /proc. Если применен аргумент noaccess, то для управляемого процессов сервиса изымается возможность доступа к метаданным большинства пользовательских процессов. Если применен аргумент invisible,

то процессы других пользователей скрываются от управляемого процесса или сервиса. При применении аргумента `ptraceable` запрещена трассировка процессов-абонентов. Если применять аргумент `default` – никаких ограничений не устанавливается. В общем случае рекомендуется применять аргумент `invisible`, но это может быть возможно не для любого сервиса.

`ProcSubset=`
Изоляция при обмене данными.

Параметр может принимать 2 аргумента: `all` (применяется по умолчанию) и `pid`. Фактически, параметр контролирует состояние флага монтирования `subset=` файловой системы `/proc`. Если применяется аргумент `pid` (рекомендуемый способ), то в файловой системе `/proc` для управляемого процесса делаются невидимыми все файлы и каталоги, не связанные напрямую с управлением процессами.

`ProtectSystem=`
Изоляция процесса от основной системы.

Параметр принимает аргументом или логическое значение `true/false`(используется по умолчанию), или значения `full` и `strict`. Если аргумент равен `true`, то для процесса каталоги `/usr`, `/boot` и `/efi` монтируются в режиме «только для чтения». Если значение аргумента установлено в `full`, то дополнительно в режиме «только для чтения» монтируется и каталог `/etc`. Если значение аргумента установлено в `strict`, тогда вся файловая система (за исключением каталогов `/dev`, `/proc` и `/sys`) будет смонтирована в режиме «только для чтения». Параметр `ReadWritePaths=` может использоваться для исключения того, чтобы определенные каталоги были доступны только для чтения.

`ProtectHome=`
Изоляция процесса от домашних каталогов пользователей.

Принимает логический аргумент `true/false`(используется по умолчанию), или специальные значения аргумента «`read-only`» или «`tmpfs`». При значении `true` каталоги `/home`, `/root` и `/run/user` становятся недоступными и пустыми для службы и её процессов. Если установлено значение «`read-only`» те же три каталога становятся доступными только для чтения. Если установлено значение «`tmpfs`», временные файловые системы монтируются в указанных трех каталогах в режиме только для чтения. Значение «`tmpfs`» полезно, чтобы скрыть домашние каталоги, не относящиеся к процессам.

`RuntimeDirectory=`
`StateDirectory=`

`CacheDirectory=`
`LogsDirectory=`
`ConfigurationDirectory=`
Изоляция данных службы (процесса).

Эти параметры принимают аргументами список имен каталогов (пути), и отделяются пробелами, если путей несколько. Пути имен каталогов должны быть относительными и не должны включать « `..` ». Если параметр и аргумент установлен, то при запуске службы будет создан один или несколько каталогов с указанными именами. Кроме того, будет определена соответствующая переменная среды с полными путями к каталогам. Если задано несколько каталогов, то в переменной среды пути объединяются двоеточием, см. Таблицу 16:

Параметр	Каталог ФС по умолчанию	Переменная XDG	Переменная среды
`RuntimeDirectory=`	`/run/`	`$XDG_RUNTIME_DIR`	`$RUNTIME_DIRECTORY`
`StateDirectory=`	`/var/lib/`	`$XDG_CONFIG_HOME`	`$STATE_DIRECTORY`
`CacheDirectory=`	`/var/cache/`	`$XDG_CACHE_HOME`	`$CACHE_DIRECTORY`
`LogsDirectory=`	`/var/log/`	`$XDG_CONFIG_HOME/log/`	`$LOGS_DIRECTORY`
`ConfigurationDirectory=`	`/etc/`	`$XDG_CONFIG_HOME`	`$CONFIGURATION_DIRECTORY`

Таблица 16: Переменные служебных каталогов для службы и её процессов

`RuntimeDirectoryMode=`
`StateDirectoryMode=`
`CacheDirectoryMode=`
`LogsDirectoryMode=`
`ConfigurationDirectoryMode=`
Права доступа при изоляции данных службы (процесса).
Переменные, которые устанавливают права доступа в десятичной нотации для файлов, которые будут созданы в каталогах, определенных в Таблице 16. Значение по умолчанию – `0755`.

`PrivateTmp=`
Изоляция временных файлов.

Параметр принимает логический аргумент `true` или `false`(используется по умолчанию). Для службы и её процессов будут созданы собственные изолированные каталоги `/tmp` и `/var/tmp`. Содержимое их будет очищаться при каждом старте службы.

`PrivateDevices=`

Изоляция устройств.

В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается изолированное окружение в каталоге `/dev`, в котором создаются только псевдо-устройства, такие как `/dev/null`, `/dev/zero`, `/dev/[u]random` и т.п., но не создается никаких устройств, гарантирующих доступ к реальной периферии (запрещается создание устройств типа `/dev/sda`, `/dev/mem`, портов ввода-вывода `/dev/ports` и т.п.).

`PrivateNetwork=`
Изоляция сетевых устройств.

В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается изолированное сетевое окружение, и создается только один изолированный интерфейс `lo` (обратной петли, loopback).

8.13.3.2 Изоляция процесса в `systemd` на уровне пользователя или группы

Следующая группа опций относится к изоляции на уровне пользователей и групп.
Эти параметры доступны только для системных служб и не поддерживаются для служб, работающих в пользовательских окружениях `systemd`.

`User=` `Group=`
Изоляция на уровне пользователя или группы.

Устанавливает для службы имя пользователя или группы соответственно, от имени которых выполняются её процессы. Принимает в качестве аргумента одно имя пользователя/группы (или числовой идентификатор). Для системных служб и для служб администратора `root` (службы, управляемые экземпляром `systemd --user`) по умолчанию используется аргумент «`root`». Иначе, аргументу `User=` можно присваивать значение, указывающее на любого другого пользователя.

`DynamicUser=`
Изоляция на уровне пользователя или группы.

Принимает логический аргумент `true` или `false`. Если установлено `true`, пара для значений пользователя и группы выделяется динамически при запуске службы и освобождается, как только служба останавливается. Пользователь и группа не будут добавлены в `/etc/passwd`

или /etc/group, но временно управляются во время выполнения. Модуль nss-systemd(8) glibc NSS обеспечивает интеграцию этих динамических пользователей/групп в базы данных пользователей и групп системы (например при использовании доменов Microsoft™ Active Directory или IPA).

PAMName=
Изоляция на уровне модуля PAM.

Аргументом к этому параметру можно указать имя модуля PAM для настройки сеанса. Если значение установлено, то будет зарегистрирован сеанс PAM от имени указанной службы. Это полезно только в сочетании с настройкой User=, в противном случае игнорируется. Если значение аргумента не установлено, то сеанс PAM не будет открываться для выполняемых процессов службы.

PrivateUsers=
Изоляция на уровне пользователя или группы.

В качестве аргумента параметр принимает логическое значение true или false (принимается по умолчанию). Если значение аргумента true, то для службы и её процессов устанавливается изолированное окружение, в котором существуют только пользователи и группа root и пользователь nobody с такой же группой.

8.13.3.3 Изоляция процесса в systemd на уровне перечня возможностей (Capabilities)

Параметры ниже применимы только для системных служб systemd или для служб, работающих в пользовательских окружениях systemd, если не отключено использование пространства имен.

CapabilityBoundingSet=
Назначение определенных возможностей (Capabilities).

Определяет, какие возможности (Capabilities) включать в ограничивающий набор для выполняемого процесса. Подробнее см. в интерактивной справке capabilities(7). Принимает перечень возможностей, разделенных пробелами, например: CAP_SYS_ADMIN, CAP_DAC_OVERRIDE, CAP_SYS_PTRACE и т.п. Указанные в перечне возможности будут включены в ограничивающий набор, работа всех остальных будет заблокирована. Синтаксис определения аргументов допускает применение + и − для добавления или исключения элементов перечня. Например, если агруметом указать −CAP_SYS_PTRACE, то будут разрешены все Capabilities, кроме указанной. Также указывать параметр CapabilityBoundingSet= можно несколько раз. Тогда чтение этого параметра будет выполнятся последовательно, в заданном порядке. По умолчанию никакие возможности не блокируются.

8.13.3.4 Общие параметры безопасности процесса в systemd

`NoNewPrivileges=`
Общая безопасность.

Принимает логический аргумент `true` или `false` (последний аргумент принимается по умолчанию). Если аргумент `true`, то служба, её процессы и все дочерние процессы никогда не смогут получить новые привилегии при выполнении системного вызова `execve()` (например, через биты `setuid` или `setgid`, даже если они явно будут установлены). Это самый простой и эффективный способ гарантировать, что процесс и его потомки никогда больше не смогут повысить свои привилегии. Следует помнить, что ряд параметров перезаписывает параметр `NoNewPrivileges=`. Список параметров при которых игнорируется описываемый:

`DynamicUser=`
`LockPersonality=`
`MemoryDenyWriteExecute=`
`PrivateDevices=`
`ProtectClock=`
`ProtectHostname=`
`ProtectKernelLogs=`
`ProtectKernelModules=`
`ProtectKernelTunables=`
`RestrictAddressFamilies=`
`RestrictNamespaces=`
`RestrictRealtime=`
`RestrictSUIDSGID=`
`SystemCallFilter=`
`SystemCallLog=`
`SystemCallArchitectures=`

`ProtectHostname=`
Защита имени узла.
В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается запрет на изменение имени (включая полное доменное имя) узла.

`ProtectClock=`
Защита системного времени.
В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается запрет

на изменение системного времени.

`ProtectKernelTunables=`
Защита параметров ядра ОС.

В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается запрет на изменение параметров ядра ОС с помощью `sysctl`, а доступ к содержимому каталогов и файлов в `/proc/sys`, `/sys`, `/proc/sysrq-trigger`, `/proc/latency_stats`, `/proc/acpi`, `/proc/timer_stats`, `/proc/fs` и `/proc/irq` устанавливается в режиме «только для чтения».

`ProtectKernelModules=`
Защита параметров ядра ОС.

В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается запрет на загрузку и выгрузку модулей ядра операционной системы.

`ProtectKernelLogs=`
Защита кольцевого буфера аудита ядра ОС.

В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается запрет на чтение и запись устройств `/dev/kmsg` и `/proc/kmsg`. Таким образом, служба и ее процессы могут получать и записывать информацию аудита только от программ, работающих в пользовательском пространстве.

`ProtectControlGroups=`
Защита доступа к контрольным группам (Control Groups).

В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается запрет на запись в каталог `/sys/fs/cgroup`. Подробнее о контрольных группах Linux можно прочесть в страницах интерактивной справки `cgroups(7)`.

`RestrictFileSystems=`
Защита на уровне файловой системы.

Параметр принимает в качестве аргумента разделенный пробелами список файловых систем, на которых можно совершать операции с файлами. Может быть несколько вхождений, каждое из которых обрабатывается последовательно. А если перед файловой системой указан знак –, то доступ инвертируется. Например, если указано:

```
RestrictFileSystems=ext4 tmpfs
RestrictFileSystems= ext4
```

то будет предоставлен доступ только к файловой системе `tmpfs`.

`RestrictAddressFamilies=`

Защита сети.

Параметр принимает четыре аргумента, указывающие семейство адресов, доступ к которым может быть предоставлен. Семейства перечисляются последовательно, и отделяются пробелом. Если указан аргумент `none`, то доступ к любому семейству адресов не предоставляется (фактически этим вводится запрет на сетевое взаимодействие). Если указан аргумент `AF_UNIX`, то службе или её процессу предоставляется доступ только к сокетам UNIX. Если указан аргумент `AF_INET`, то службе или её процессу предоставляется доступ только к адресам семейства IPv4. Если указан аргумент `AF_INET6`, то службе или её процессу предоставляется доступ только к адресам семейства IPv6.

`RestrictNamespaces=`

Изоляция на уровне пространства имен.

Параметр принимает в качестве аргумента разделяемый пробелами список пространств имен, или логический аргумент. Список состоит из следующих значений: `cgroup`, `ipc`, `net`, `mnt`, `pid`, `user`, `time` и `uts`. Если указано значение аргумента `true`, то доступ к любому пространству имен блокируется. Если `false` (значение по умолчанию) – наоборот, доступ разрешается без ограничений. Подробнее о пространстве имен (namespaces), см. интерактивную справку `namespaces(7)`.

`MemoryDenyWriteExecute`

Защита памяти.

В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается запрет на создание страниц памяти, которые доступны одновременно на запись (изменение) и выполнение. Также запрещено выделение разделяемых областей (сегментов) памяти процесса как исполняемых.

`RestrictRealtime=`

Общая защита.

В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается запрет на выполнение в реальном времени (с низкими задержками).

`RestrictSUIDSGID=`
Общая защита.

В качестве аргумента параметр принимает логическое значение `true` или `false` (принимается по умолчанию). Если значение аргумента `true`, то для службы и её процессов устанавливается запрет на создание файлов и папок, снабженных битами SUID и SGID.

8.13.3.5 Изоляция процесса в `systemd` с использованием полномочного доступа MAC

Указанные параметры применимы только для системных процессов и только в том случае, если используется та или иная реализация полномочной (мандатной) модели разграничения прав доступа. Поскольку описание полномочного разграничения доступа не является материалом данного руководства, параметры приводятся справочно и не все.

`SELinuxContext=`
Разграничение доступа на уровне контекста SELinux.

Определяет контекст безопасности SELinux для исполняемого процесса. Если контекст устанавливатся с помощью `systemd`, то это переопределит автоматический переход домена. Однако, общесистемная политика должна разрешать переход. Этот параметр игнорируется, если SELinux отключен.

`AppArmorProfile=`
Разграничение доступа на уровне профиля AppArmor.

Принимает имя профиля AppArmor в качестве аргумента. Служба и её процессы при запуске переключатся на заданный профиль. Профили должны быть уже загружены в ядро. Если задан префикс «-», все ошибки будут проигнорированы. Этот параметр не действует, если разграничение доступа с помощью AppArmor не активизировано.

8.13.3.6 Ограничения для процесса в `systemd` на уровне доступа к ресурсам

С помощью `systemd` можно установить мягкие (soft) и строгие (hard) лимиты на доступ к разнообразным ресурсам системы для службы и её процессов. Лимиты устанавливаются аргументами

к параметру, и разделяются двоеточиями. Если указывается один аргумент, то `systemd` будет считать, что это мягкое ограничение. Иначе, нужно поставить двоеточие, а потом задать ограничение. К аргументу можно приписывать суффиксы байт: K, M, G, T и P (по модулю 1024).

Основные опции ограничения ресурсов перечислены в Таблице 17:

Опция	Эквивалент команды `ulimit`	Единица расчета
`LimitCPU=`	`ulimit -t`	В секундах
`LimitFSIZE=`	`ulimit -f`	В байтах
`LimitDATA=`	`ulimit -d`	В байтах
`LimitSTACK=`	`ulimit -s`	В байтах
`LimitCORE=`	`ulimit -c`	В байтах
`LimitRSS=`	`ulimit -m`	В байтах
`LimitNOFILE=`	`ulimit -n`	Число файловых дескрипторов
`LimitAS=`	`ulimit -v`	В байтах
`LimitNPROC=`	`ulimit -u`	Число разрешенных процессов
`LimitMEMLOCK=`	`ulimit -l`	В байтах
`LimitLOCKS=`	`ulimit -x`	Число блокировок
`LimitSIGPENDING=`	`ulimit -i`	Число сигналов в очереди
`LimitMSGQUEUE=`	`ulimit -q`	В байтах
`LimitNICE=`	`ulimit -e`	Приоритет
`LimitRTPRIO=`	`ulimit -r`	Приоритет реального времени

Таблица 17: Опции ограничения ресурсов для служб, управляемых с помощью `systemd`.

`UMask=`

Маска прав доступа при создании файлов.

Параметр `UMask` задает (в десятичной нотации) маску прав доступа на создаваемые файлы. Рекомендуемое значение – 0027 или более строгое.

`CoredumpFilter=`

Ограничение на данные, которые могут попасть в снимок памяти при сбросе снимка на диск.

Параметр контролирует набор данных, который может попасть на диск в снимок памяти при крахе программы или при явном сбросе снимка (дампа) пользователем или при получении необходимого сигнала. Аргументы могут принимать следующие значения:

`all` – любой тип (то есть никакие данные на попадут в снимок), рекомендуемое значение;

`default` – значение по умолчанию (используется текущее значение для процесса, указанное в файле `/proc/PID/coredump_filter`);

Остальные значения можно узнать из страницы интерактивной справки `core(5)`.

`OOMScoreAdjust=` `OOMPolicy=`

Политика реагирования на потребляемую процессом память.

Параметр `OOMScoreAdjust=` задает значение регулировки OOM (Out-Of-Memory) killer. OOM-killer – механизм яра Linux, назначение которого в том, чтобы прерывать процессы, которые потребляют слишком много оперативной памяти (обычно это происходит ввиду ошибок утечки памяти, или повышенной нагрузки на обработку данных процессом). Значение может задаваться в пределах от `-1000` до `1000`. Где `-1000` – это не отслеживать процесс и его потребление, а `1000` – делает приоритетное прерывание процесса ядром ОС наиболее вероятным.

Параметр `OOMPolicy=` предназначен для `systemd`, и задает поведение `systemd` в отношении процессов, которые были прерваны ядром с помощью OOM-killer. Аргументы значения параметра могут быть получены из страницы интерактивной справки `systemd.service(5)`.

`Personality=`

Ограничение архитектуры.

Параметр принимает аргументом значение архитектуры процессора. Обычно используется для того, чтобы выполнять процессы (службы) 32-х битной архитектуры в 64-х битной системе. Возможные значения – `x86`, `x86-64`, `ppc`, `ppc-le`, `ppc64`, `ppc64-le`, `s390` или `s390x`. По умолчанию значение этого параметра совпадает со значением архитектуры ядра.

`LockPersonality=`

Ограничение архитектуры.

Параметр принимает аргументом логическое `true` или `false` (используется по умолчанию). Если значение аргумента `true`, то блокируется изменение параметра `Personality` (см. выше).

`Nice=`

Приоритет.

Параметр принимает аргументом значение приоритета в диапазоне от `-20` (самый высокий) до `19` (самый низкий).

8.13.3.7 Фильтрация системных вызовов для процесса в `systemd`

Служба `systemd` предоставляет возможность использовать фильтр системных вызовов для дочерних служб и процессов. По умолчанию, доступ к системным вызовам ничем не ограничивается.

`SystemCallArchitectures=`

Ограничение архитектуры.

Параметр в качестве аргумента принимает разделенные пробелом значения аппаратных архитектур, системные вызовы которых можно выполнять. Возможные значения – `x86`, `x86-64`, `ppc`, `ppc-le`, `ppc64`, `ppc64-le`, `s390`, `s390x` или `native`. Если использовать аргумент `native`, то используется та архитектура, под которую собран сам менеджер `systemd`.

`SystemCallFilter=`
Ограничение архитектуры.

Параметр в качестве аргумента принимает разделенные пробелом названия системных вызовов, запуск которых разрешен. Попытки запуска иных системных вызовов будут блокированы. Так как поддерживаемых системных вызовов в современных ядрах Linux много (почти четыреста для каждой из самых популярных архитектур процессора), то их можно группировать. Такие группы указываются с помощью знака «@».

Для получения подробных сведений о наборах и перечне вызовов в каждом наборе можно выполнить:

```
# systemd-analyze syscall-filter
# systemd-analyze syscall-filter <имя набора>
```

Листинг 258: Пример запроса информации по фильтрам системных вызовов `systemd`

Уже предусмотренные наборы системных вызовов, собранных в группы, указаны в Таблице 18:

Группа	Описание
`@aio`	Системные вызовы для асинхронного ввода-вывода, такие как `io_setup(2)`, `io_submit(2)` и т.п.
`@basic-io`	Системные вызовы для основных операций (чтение, запись, поиск, копирование метаданных, закрытие) ввода-вывода (`read(2)`, `write(2)` и т.п.).
`@chown`	Системные вызовы `chown(2)`, `fchownat(2)` и аналогичные, предназначенные для смены владельца.
`@clock`	истемные вызовы `adjtimex(2)`, `settimeofday(2)` и аналогичные, предназначенные для смены времени.
`@cpu-emulation`	Системные вызовы `vm86(2)` и аналогичные, предназначенные для эмуляции процессоров.
`@debug`	истемные вызовы `ptrace(2)`, `perf_event_open(2)` и аналогичные, предназначенные для отладки, трассировки и профилирования.
`@file-system`	Системные вызовы для файловых операций (создание, открытие файлов и директорий, запись в них, чтение (включая атрибуты), удаление (переименование), создание и удаление ссылок).
`@io-event`	Системные вызовы обработчиков событий типа `poll(2)`, `select(2)`, `epoll(2)`, `eventfd(2)`.
`@ipc`	Системные вызовы для обработки очередей сообщений (SYSV, POSIX, и др.) типа `mq_overview(7)`, `svipc(7)`.
`@keyring`	Системные вызовы доступа к хранилищу ключей ядра типа `keyctl(2)` и др.
`@memlock`	Системные вызовы блокировки страниц памяти `mlock(2)`, `mlockall(2)` и др.

Группа	Описание
@module	Системные вызовы загрузки/выгрузки модулей ядра init_module(2), delete_module(2), и др.
@mount	Системные вызовы подключения/отключения ФС mount(2), chroot(2), и др.
@network-io	Системные вызовы для работы с сокетами socket(7), unix(7), и др.
@obsolete	Устаревшие и пока не реализованные системные вызовы.
@privileged	Системные вызовы, которые требуют специальные привилегии (Capabilities) root.
@process	Системные вызовы, отвечающие за выполнение процессов clone(2), kill(2), namespaces(7) и др.
@raw-io	Системные вызовы прямого доступа к портам ввода-вывода ioperm(2), iopl(2) и др.
@reboot	Системные вызовы перезагрузки и подготовки к ней (reboot(2), kexec() и др.
@resources	Системные вызовы изменения приоритетов ресурсов и планирования setrlimit(2), setpriority(2) и др.
@setuid	Системные вызовы изменения пользовательского контекста setuid(2), setgid(2), setresuid(2) и др.
@signal	Системные вызовы обработки и назначения сигналов signal(2), sigprocmask(2) и др.
@swap	Системные вызовы включения/отключения подкачки swapon(2), swapoff(2).
@system-service	Разумный набор системных вызовов, используемых системными службами, за исключением специальных. Это рекомендуемый минимум списка разрешенных вызовов для системных служб, поскольку он содержит то, что обычно требуется системным службам, но исключает специальные наборы. Например, исключаются следующие наборы: @clock, @mount, @swap, @reboot и некоторые другие.
@timer	Системные вызовы планировщика alarm(2), timer_create(2) и др.
@known	Все системные вызовы, определенные в выполняющемся ядре.

Таблица 18: Таблица группировки системных вызовов для параметра SystemCallFilter

8.14 Блокировка сессии терминала по тайм-ауту

Для блокировки используется мультиплексор терминала tmux с вызовом программы блокировки vlock.

Для установки мультиплексора терминала tmux и программы блокировки vlock выполнить (от имени root):

```
# apt-get install vlock tmux
```

Листинг 259: Пример установки утилит tmux и vlock

Для конфигурации пользовательской сессии с учетом принудительного запуска tmux внести следующие директивы в файл /etc/bash.bashrc:

```
if
command -v tmux &> /dev/null && [ -n "$PS1" ] && [[ ! "$TERM" =~ screen ]] && [[ !
"$TERM" =~ tmux ]] && [ -z "$TMUX" ]; then
exec tmux
```

```
fi
```

Листинг 260: Пример конфигурации `/etc/bash.bashrc` для запуска мультиплексора терминала `tmux`

Убрать (или закомментировать) переменную среды TMOUT из файла `/etc/profile`, если она была задана.

Установить параметры `tmux` на отсчет времени истечения неактивности сессии (рекомендуемое значение = 900 секунд, 15 минут) и принудительный запуск блокировки с помощью `vlock` в файле `/etc/tmux.conf`:

```
set -g lock-command vlock
set -g lock-after-time 900
bind L lock-session
set -g mouse on
set-option -g history-limit 30000
```

Листинг 261: Пример конфигурации `/etc/tmux.conf`

Об успешном запуске мультиплексора терминала будет свидетельствовать зеленая полоса внизу экрана. Помимо тайм-аута, можно ограничить количество строк для прокрутки в терминале (в примере – 30000). Для прокрутки настройка предусматривает использование мышки либо Alt+b+[, после чего можно использовать стрелки вниз и вверх на клавиатуре, либо PgUp/PgDn.

9 Справочные таблицы

Приведенные в настоящем разделе справочные таблицы содержат сведения о составе заимствованного и привлекаемого ПО, а также со сведения по технологическим пользователям и пользователям, созданным предварительно в целях проведения испытаний.

Состав пакетов DEB:

№ п/п	КС CRC32	Имя пакета и версия	Назначение
1	e4898df8	apache2_2.4.41-4ubuntu3.12_amd64.deb	Веб-сервер, метапакет
2	833340f1	apache2-bin_2.4.41-4ubuntu3.12_amd64.deb	Веб-сервер, исполняемые файлы
3	597b4d50	apache2-data_2.4.41-4ubuntu3.12_all.deb	Веб-сервер, данные
4	815a3d5e	apache2-utils_2.4.41-4ubuntu3.12_amd64.deb	Веб-сервер, утилиты
5	159ce94a	containerd_1.5.9-0ubuntu1 20.04.4_amd64.deb	Исполняемая среда и служба для запуска контейнеров
6	46eb752c	cri-o_1.23.0 0_amd64.deb	Исполняемая среда для систем оркестрации контейнеров
7	a18b4eb4	libapache2-mod-php7.4_7.4.3-4ubuntu2.13_amd64.deb	Модуль поддержки PHP для веб-сервера
8	39f02139	libapr1_1.6.5-1ubuntu1_amd64.deb	Переносимая библиотека поддержки веб-сервера
9	a90840fa	libaprutil1_1.6.1-4ubuntu2_amd64.deb	Интерфейсы для переносимой библиотеки поддержки веб-сервера
10	583049d1	libaprutil1-dbd-sqlite3_1.6.1-4ubuntu2_amd64.deb	Интерфейсы для переносимой библиотеки поддержки веб-сервера
11	5a02155e	libaprutil1-ldap_1.6.1-4ubuntu2_amd64.deb	Интерфейсы для переносимой библиотеки поддержки веб-сервера
12	c118521f	liblua5.2-0_5.2.4-1.1build3_amd64.deb	Язык программирования LUA
13	af7a69aa	libnginx-mod-http-image-filter_1.18.0-0ubuntu1.3_-amd64.deb	Модуль преобразования графических форматов дя веб-сервера
14	a1a2346b	libnginx-mod-http-xslt-filter_1.18.0-0ubuntu1.3_amd64.deb	Модуль фильтра преобразования XML для веб-сервера

№ п/п	КС CRC32	Имя пакета и версия	Назначение
15	82975abd	libnginx-mod-mail_1.18.0-0ubuntu1.3_amd64.deb	Модуль поддержки протоколов электронной почты для веб-сервера
16	2d3d2de2	libnginx-mod-stream_1.18.0-0ubuntu1.3_amd64.deb	Модуль поддержки потокового прокси для веб-сервера
17	bcd71c9e	libonig5_6.9.4-1_amd64.deb	Библиотека регулярных выражений
18	b09ab51a	librabbitmq4_0.10.0-1_amd64.deb	Библиотека поддержки протокола AMQP
19	031193e9	libxmlrpc-epi0_0.54.2-1.2_amd64.deb	Библиотека поддержки запросов XML-RPC
20	9eef2ecd	libzip5_1.5.1-0ubuntu1_amd64.deb	Библиотека поддержки сжатия данных
21	1067d7f6	nginx_1.18.0-0ubuntu1.3_all.deb	Веб- и прокси сервер
22	5ccb1ed7	nginx-common_1.18.0-0ubuntu1.3_all.deb	Веб- и прокси сервер, данные
23	ebc83759	nginx-core_1.18.0-0ubuntu1.3_amd64.deb	Веб- и прокси сервер, основные файлы
24	871b2911	nodejs_12.22.12-deb-1nodesource1_amd64.deb	Среда выполнения для JavaScript
25	12ca4197	openssh-client_1 3a8.2p1-4ubuntu0.5_amd64.deb	Клиент для протокола SSH
26	fdbf0eaf	openssh-server_1 3a8.2p1-4ubuntu0.5_amd64.deb	Служба SSH
27	f563f3a7	openssh-sftp-server_1 3a8.2p1-4ubuntu0.5_amd64.deb	Поддержка передачи файлов для службы SSH
28	910c7d33	php7.4_7.4.3-4ubuntu2.13_all.deb	Язык программирования PHP
29	943f8ebc	php7.4-bcmath_7.4.3-4ubuntu2.13_amd64.deb	Библиотека поддержки вычислений с произвольной точностью для PHP
30	725b69ec	php7.4-fpm_7.4.3-4ubuntu2.13_amd64.deb	Сервер приложений для PHP
31	bb9c0dfd	php7.4-gd_7.4.3-4ubuntu2.13_amd64.deb	Библиотека поддержки графики для PHP
32	5aa4fe0c	php7.4-intl_7.4.3-4ubuntu2.13_amd64.deb	Библиотека поддержки локализаций для PHP
33	857860d3	php7.4-mbstring_7.4.3-4ubuntu2.13_amd64.deb	Библиотека поддержки многобайтовых строк для PHP
34	88ef4e21	php7.4-pgsql_7.4.3-4ubuntu2.13_amd64.deb	Библиотека поддержки СУБД PostreSQL для PHP
35	30cb1479	php7.4-soap_7.4.3-4ubuntu2.13_amd64.deb	Библиотека поддержки протокола SOAP для PHP
36	470635ad	php7.4-xmlrpc_7.4.3-4ubuntu2.13_amd64.deb	Библиотека поддержки XML-RPC для PHP
37	32aabfb9	php7.4-zip_7.4.3-4ubuntu2.13_amd64.deb	Библиотека поддержки сжатия данных для PHP
38	866017ae	php-amqp_1.9.4-3ubuntu1_amd64.deb	Библиотека поддержки протокола AMQP для PHP
39	03214392	php-igbinary_3.1.2+2.0.8-1build1_amd64.deb	Библиотека, альтернативная встроенным функциям сериализации для PHP
40	0dddfea3	php-mongodb_1.6.1-4build1_amd64.deb	Библиотека поддержки СУБД MongoDB для PHP
41	2f29d421	php-redis_5.1.1+4.3.0-1_amd64.deb	Расширение, поддерживающее интерфейсы к Redis для PHP
42	b306eb65	php-xdebug_2.9.2+2.8.1+2.5.5-1build1_amd64.deb	Расширение, поддерживающее интерфейсы отладки для PHP
43	04f999fd	runc_1.1.0-0ubuntu1 20.04.1_amd64.deb	Среда, поддерживающая выполнение контейнеров
44	38d8c609	ssh_1 3a8.2p1-4ubuntu0.5_all.deb	Служба SSH
45	7d5406eb	wget_1.20.3-1ubuntu2_amd64.deb	Клиент для протокола HTTP

Таблица 19: Состав заимствованного и привлекаемого ПО. DEB пакеты.

Состав пакетов PIP:

№ п/п	КС CRC32	Имя пакета и версия	Назначение
1	683022ae	ansible-5.10.0.tar.gz	Среда автоматизации развертывания
2	6db246f7	ansible_compat-2.2.1-py3-none-any.whl	Среда автоматизации развертывания, библиотеки совместимости
3	7b20bee4	ansible-core-2.12.5.tar.gz	Среда автоматизации развертывания, основные компоненты
4	69e1d1f0	ansible-core-2.12.7.tar.gz	Среда автоматизации развертывания, основные компоненты
5	6b25d4cd	ansible_lint-6.7.0-py3-none-any.whl	Среда автоматизации развертывания, утилиты
6	cced1556	attrs-22.1.0-py2.py3-none-any.whl	Среда автоматизации развертывания, библиотека поддержки классов
7	181eaacf	black-22.10.0-cp38-cp38-manylinux_2_17_x86_-64.manylinux2014_x86_64.whl	Библиотека поддержки форматирования для Python
8	6fc83162	bracex-2.3.post1-py3-none-any.whl	Библиотека обработки скобок для Python
9	cde1d93d	calico-3.22.3-v3.22.3.tar.gz	Контроллер сетевых политик для k8s
10	89720fcf	cffi-1.15.1-cp38-cp38-manylinux_2_17_x86_-64.manylinux2014_x86_64.whl	Интерфейсы Python для C

№ п/п	КС CRC32	Имя пакета и версия	Назначение
11	fe22ee16	click-8.1.3-py3-none-any.whl	Библиотека поддержки командной строки
12	771d05e9	cni-plugins-linux-amd64-v1.1.1.tgz	Набор данных (шаблонов и плагинов) для конфигурации сети
13	9e4ea5ef	commonmark-0.9.1-py2.py3-none-any.whl	Парсер и библиотеки поддержки языка разметки Markdown
14	4a8b06fa	cryptography-3.4.8-cp36-abi3-manylinux_2_17_x86_-64.manylinux2014_x86_64.whl	Библиотека поддержки криптографических примитивов
15	75b7164f	cryptography-38.0.1-cp36-abi3-manylinux_2_17_x86_-64.manylinux2014_x86_64.whl	Библиотека поддержки криптографических примитивов
16	f53136e4	enrich-1.2.7-py3-none-any.whl	Библиотека поддержки программного интерфейса RichAPI, предназначенная для обработки вывода в строку терминала
17	82fdc1fc	etcd-v3.5.3-linux-amd64.tar.gz	Реализация распределенного хранилища ключей для общей конфигурации, обнаружения служб и координации планировщика распределенных систем или кластеров машин
18	2390dc88	filelock-3.8.0-py3-none-any.whl	Модуль языка Python, реализующий верхнеуровневую обработку файловых блокировок в пространстве пользователя
19	13d6aae9	importlib_resources-5.10.0-py3-none-any.whl	Модуль, обеспечивающий обратную совместимость функций импорта данных с устаревшими версиями Python
20	a5b995a7	Jinja2-2.11.3-py2.py3-none-any.whl	Библиотека шаблонов
21	64fafe7d	Jinja2-3.1.2-py3-none-any.whl	Библиотека шаблонов
22	75e21d0f	jmespath-0.9.5-py2.py3-none-any.whl	Библиотека обработки структур данных JSON
23	bf063f88	jsonschema-4.16.0-py3-none-any.whl	Реализация обработчика данных схемы JSON
24	aa2cfe67	helm-v3.8.2-linux-amd64.tar.gz	Набор инструментов для определения, установки и обновления приложений, работающих в Kubernetes
23	67850007	MarkupSafe-1.1.1-cp38-cp38-manylinux2010_x86_64.whl	Библиотека обработчика данных ESCAPE-последовательностей
25	2df5c5cc	MarkupSafe-2.1.1-cp38-cp38-manylinux_2_17_x86_-64.manylinux2014_x86_64.whl	Библиотека обработчика данных ESCAPE-последовательностей
26	aa107136	mypy_extensions-0.4.3-py2.py3-none-any.whl	Библиотека расширений, реализующая функции дополнительных проверок при вводе данных
27	58527057	netaddr-0.7.19-py2.py3-none-any.whl	Библиотека обработки сетевых адресов
28	19920e95	packaging-21.3-py3-none-any.whl	Набор утилит, обеспечивающий соответствие спецификациям PEP 440 и PEP 425
29	450177a5	pathspec-0.10.1-py3-none-any.whl	Библиотека утилит, обеспечивающая обработку путей по заданному шаблону
30	315ef7ef	pbr-5.4.4-py2.py3-none-any.whl	Библиотека обработки данных Python отслеживающая корректность при сборке (комплексировании)
31	a0f47be9	pkgutil_resolve_name-1.3.10-py3-none-any.whl	Модуль, обеспечивающий обратную совместимость функций разрешения имен с устаревшими версиями Python
32	1f86e8cc	platformdirs-2.5.2-py3-none-any.whl	Библиотека обработчика директорий, специфичных для заданной платформы
33	322d3168	pycparser-2.21-py2.py3-none-any.whl	Реализация парсера, преобразующего код на языке Си в абстрактное синтаксическое дерево
34	6e140f6c	Pygments-2.13.0-py3-none-any.whl	Модуль, обеспечивающий подсветку синтаксиса
35	c7cd0b16	pyparsing-3.0.9-py3-none-any.whl	Универсальный модуль обработки и проверки синтаксиса
36	2c224bb9	pyrsistent-0.18.1-cp38-cp38-manylinux_2_17_x86_-64.manylinux2014_x86_64.whl	Набор постоянных синтаксических структур
37	f34fc3ed	PyYAML-6.0-cp38-cp38-manylinux_2_5_x86_-64.manylinux1_x86_64.manylinux_2_12_x86_-64.manylinux2010_x86_64.whl	Библиотека поддержки парсера YAML для Python
38	ff6c0958	resolvelib-0.5.4-py2.py3-none-any.whl	Библиотека, обеспечивающая обработку функций разрешения имен
39	d60b5536	rich-12.6.0-py3-none-any.whl	Библиотека поддержки программного интерфейса RichAPI, предназначенная для обработки вывода в строку терминала
40	213a3032	ruamel.yaml-0.16.10-py2.py3-none-any.whl	Модуль, обеспечивающий загрузку и выгрузку данных YAML
41	cb571859	ruamel.yaml.clib-0.2.6-cp38-cp38-manylinux1_x86_64.whl	Модуль, обеспечивающий загрузку и выгрузку данных YAML
42	249b4dfe	setuptools-65.5.0-py3-none-any.whl	Набор утилит пакетного менеджера для Python
43	f667def0	subprocess_tee-0.3.5-py3-none-any.whl	Библиотека оперативного вывода и вывода данных
44	9f6e2cf1	tomli-2.0.1-py3-none-any.whl	Парсер для языка (формата) TOML

№ п/п	КС CRC32	Имя пакета и версия	Назначение
45	2c11bbde	typing_extensions-4.4.0-py3-none-any.whl	Библиотека расширений при вводе данных
46	414f0c35	wcmatch-8.4.1-py3-none-any.whl	Библиотека фильтров к интерпретатору Bash
47	419368cd	yamllint-1.28.0-py2.py3-none-any.whl	Библиотека проверки синтаксиса YAML
48	1fbf1e8b	zipp-3.10.0-py3-none-any.whl	Обработчик метаданных для файлов (архивов) в формате zip

Таблица 20: Состав заимствованного и привлекаемого ПО. PIP пакеты.

Состав пакетов NPM:

№ п/п	КС CRC32	Имя пакета и версия	Назначение
1	c3fc331e	abab-2.0.5.tgz	Утилита для кодирования/декодирования BASE64
2	070ca0ec	abbrev-1.1.1.tgz	Парсер для аббревиатур
3	799caadb	accepts-1.3.7.tgz	Модуль согласования содержимого
4	6b192672	acorn-8.5.0.tgz	Парсер сценариев ECMA
5	7bbfdd99	acorn-import-assertions-1.8.0.tgz	Парсер, обеспечивающий импорт утверждений
6	6e181b4c	adjust-sourcemap-loader-4.0.0.tgz	Загрузчик, настраивающий исходные карты
7	7b59bab1	adm-zip-0.4.14.tgz	Реализация библиотеки сжатия данных
8	0f353864	after-0.8.2.tgz	Библиотека контроля потоков
9	a9b6efc3	agent-base-4.3.0.tgz	Библиотека для переопределения функции в экземпляр http.Agent
10	2b362b69	agentkeepalive-3.5.2.tgz	Модуль расширений для поддержки соединений HTTP/HTTPS
11	ec80c9bf	agentkeepalive-4.1.4.tgz	Модуль расширений для поддержки соединений HTTP/HTTPS
12	b0c1dbd2	aggregate-error-3.1.0.tgz	Библиотека обработчика ошибок
13	daff799b	ajv-6.12.6.tgz	Библиотека проверки корректности схемы JSON
14	3d34e505	ajv-errors-1.0.1.tgz	Обработчик ошибок для библиотеки проверки корректности схемы JSON
15	f0a91234	ajv-formats-2.1.0.tgz	Библиотека проверки корректности формата данных AJV, вер.7
16	210a80ce	ajv-keywords-3.5.2.tgz	Шаблоны ключевых слов для библиотеки проверки корректности схемы JSON
17	fe516593	allure-js-commons-1.3.2.tgz	Набор разделяемых библиотек
18	7388eb04	alphanum-sort-1.0.2.tgz	Реализация алгоритма сортировки данных
19	005930ee	angular2-text-mask-9.0.0.tgz	Директивы Angular для скрытия текстового ввода
20	76decfdd	ansi-colors-4.1.1.tgz	Поддержка вывода для текста с назначением цвета
21	87ff0551	ansi-escapes-4.3.2.tgz	Обработчик ESCAPE-последовательностей
22	7e1918dc	ansi-html-0.0.7.tgz	Реализация преобразования текста
23	b5151c35	ansi-regex-2.1.1.tgz	Библиотека регулярных выражений
24	40493c0f	ansi-styles-3.2.1.tgz	Библиотека стилей
25	5b1f66ed	anymatch-3.1.2.tgz	Библиотека для сравнения входных данных
26	1092bf27	app-root-path-3.0.0.tgz	Парсер для определения путей
27	b471f42d	aproba-1.2.0.tgz	Модуль проверки аргументов ввода
28	37fc3413	are-we-there-yet-1.1.7.tgz	Модуль для отслеживания завершения процессов
29	e885db21	argparse-1.0.10.tgz	Парсер аргументов командного интерпретатора
30	f54b9caa	aria-query-3.0.0.tgz	Библиотека поддержки запросов БД ARIA
31	ef252156	arraybuffer.slice-0.0.7.tgz	Библиотека экспорта функций для работы с массивами
32	4e615b31	array-flatten-2.1.2.tgz	Библиотека для сведения вложенных массивов
33	277ceeca	array.prototype.reduce-1.0.4.tgz	Реализация работы с массивами
34	692cfc73	array-union-1.0.2.tgz	Реализация функций создания массивов
35	f977ab47	array-uniq-1.0.3.tgz	Реализация функций создания массивов без повторений
36	bd1c9486	array-unique-0.3.2.tgz	Реализация функций создания массивов с удалением повторений
37	605cb5d7	arr-diff-4.0.0.tgz	Функция, возвращающая массив с уникальными значениями
38	8b345366	arr-flatten-1.1.0.tgz	Реализация функций создания массивов с рекурсией
39	1db7d0a6	arrify-1.0.1.tgz	Функция преобразования значений в массив
40	549cf2a7	arr-union-3.1.0.tgz	Реализация функций сравнения массивов

№ п/п	КС CRC32	Имя пакета и версия	Назначение
41	12e7181e	asap-2.0.6.tgz	Реализация очередей задач с высоким приоритетом
42	aff97d4b	asn1-0.2.4.tgz	Парсер для ASN.1
43	b2b6c89f	assert-plus-1.0.0.tgz	Модуль контроля входных данных
44	9da7f4ef	assign-symbols-1.0.0.tgz	Модуль назначения исчисляемых данных (метаданных)
45	914dcaa5	ast-types-flow-0.0.7.tgz	Библиотека типизации потоков данных
46	831d680c	async-2.6.3.tgz	Функции высшего порядка и общие шаблоны для асинхронного кода
47	a4fd041d	async-each-1.0.3.tgz	Функции асинхронных параллельных запросов
48	fafec1e6	asynckit-0.4.0.tgz	Служебная библиотека асинхронных заданий с поддержкой потоков
49	505ad974	async-limiter-1.0.1.tgz	Асинхронная очередь функций с настройкой параллельного выполнения
50	d158a524	atob-2.1.2.tgz	Библиотека для эмуляции некоторых функций браузера
51	b63adebd	autoprefixer-9.8.8.tgz	Анализатор стилей CSS
52	ae87734a	aws4-1.10.0.tgz	Библиотека подготовки и подписи запросов
53	bf189f6f	aws-sign2-0.7.0.tgz	Библиотека подписи запросов
54	fe29b705	axobject-query-2.0.2.tgz	Библиотека с реализацией программного интерфейса к информации о модели AXObject
55	c40082e2	babel-loader-8.2.2.tgz	Модуль проверки орфографии по словарю
56	23849c0d	babel-plugin-dynamic-import-node-2.3.3.tgz	Модуль поддержки проверки орфографии по словарю
57	129d29ed	babel-plugin-polyfill-corejs2-0.2.2.tgz	Модуль поддержки проверки орфографии по словарю
58	ee28dbd9	babel-plugin-polyfill-corejs3-0.2.5.tgz	Модуль поддержки проверки орфографии по словарю
59	69aa4038	babel-plugin-polyfill-regenerator-0.2.2.tgz	Модуль поддержки проверки орфографии по словарю
60	f6c08a4e	backo2-1.0.2.tgz	Модуль экспоненциального отката без сборщика мусора
61	737b75b9	balanced-match-1.0.0.tgz	Модуль проверки соответствия парам символов, напр. "{" и "}"
62	2b343337	base-0.11.2.tgz	Фреймворк для быстрого создания высококачественных серверных приложений node.js
63	2e0c1def	base64-arraybuffer-0.1.4.tgz	Библиотека для преобразования данных из кодировки BASE64 в массивы
64	0b847006	base64id-2.0.0.tgz	Библиотека поддержки создания идентификаторов в кодировке BASE64
65	bffe6716	base64-js-1.3.1.tgz	Реализация кодирования/декодирования BASE64
66	8d5f2810	batch-0.6.1.tgz	Модуль асинхронных запросов с контролем параллельного выполнения и отчетами о ходе выполнения
67	2bb1e510	bcrypt-pbkdf-1.0.2.tgz	Библиотека с реализацией одно-обратимой хеш-функции
68	629844b0	big.js-5.2.2.tgz	Библиотека десятичной арифметики произвольной точности
69	12911fed	binary-extensions-2.0.0.tgz	Модуль списка расширений бинарных файлов
70	41149d18	bl-4.1.0.tgz	Библиотека для сбора данных буферов
71	f921fd90	blob-0.0.5.tgz	Реализация библиотеки абстракций к объектам
72	54e50dbc	blocking-proxy-1.0.1.tgz	Проверочный интерфейс к прокси-серверу
73	62386632	bluebird-3.7.2.tgz	Полнофункциональная реализация доступа к шаблонам ECMA
74	18799840	body-parser-1.19.0.tgz	Интерфейс парсера для кода Node.JS
75	55f3c704	bonjour-3.5.0.tgz	Библиотека поддержки протоколов Bonjour/Zeroconf
76	cf049bc4	boolbase-1.0.0.tgz	Реализация набора базовый логических функций
77	6ffab8bb	brace-expansion-1.1.11.tgz	Парсер для командной оболочки
78	d8fa664c	braces-3.0.2.tgz	Расширение для работы со скобками
79	be745102	browserslist-4.17.3.tgz	Библиотека совместного использования запросов браузера между различными интерфейсами front-end
80	af658477	browserstack-1.6.0.tgz	Реализация клиента для работы с API BrowserStack
81	00ec312e	buffer-5.7.1.tgz	Реализация Node.js Buffer API для браузера
82	0cbe4ae1	buffer-from-1.1.1.tgz	Программный интерфейс к созданию массивов буферов
83	bf260b66	buffer-indexof-1.1.1.tgz	Библиотека поиска индекса буфера
84	97fd57a4	builtin-modules-3.2.0.tgz	Библиотека базовых модулей Node.Js
85	eba49cd4	builtins-1.0.3.tgz	Список базовых модулей Node.Js
86	0c371cd0	bytes-3.0.0.tgz	Утилита для прямого и обратного преобразования строк в байты

№ п/п	КС CRC32	Имя пакета и версия	Назначение
87	77513053	cacache-15.2.0.tgz	Реализация быстрого, отказоустойчивого дискового и независимого от данных кэша (с адресацией по содержимому)
88	2ae9fee6	cache-base-1.0.1.tgz	Базовый кеш объектов с методами get, set, del и has
89	91cb00c6	call-bind-1.0.2.tgz	Реализация надежной функции .call.bind()
90	693706e8	callsites-3.1.0.tgz	Библиотека для получения данных вызовов из API стека V8
91	6be91c51	camelcase-5.3.1.tgz	Библиотека для преобразования строк с разделителями
92	d88f9464	caniuse-api-3.0.0.tgz	Библиотека запросов для проверки совместимости браузеров
93	aa480cac	caniuse-lite-1.0.30001265.tgz	Библиотека запросов для проверки совместимости браузеров
94	89e50b71	canonical-path-1.0.0.tgz	Библиотека обработки путей
95	38ddaa03	caseless-0.12.0.tgz	Библиотека объектов set/get/has, используемая при работе с заголовками HTTP
96	32abc3de	chalk-2.4.2.tgz	Библиотека обработки стиля терминальной строки
97	8729038d	chardet-0.7.0.tgz	Детектор кодировки символов
98	68a7c1be	chokidar-3.5.2.tgz	Реализация библиотеки для просмотра файлов
99	76ccd7bd	chownr-2.0.0.tgz	Реализация утилиты chown с рекурсией по умолчанию
100	fc9fe55c	chrome-trace-event-1.0.2.tgz	Библиотека для создания трассировки приложения в соответствии с форматом событий трассировки Google
101	6a16de52	circular-dependency-plugin-5.2.2.tgz	Библиотека для обнаружения модулей с циклическими зависимостями
102	a69facbb	class-transformer-0.4.0.tgz	Библиотека преобразования классов
103	1d643c44	class-utils-0.3.6.tgz	Утилиты для работы с классами JavaScript и методами-прототипами
104	c6c10fc0	clean-stack-2.2.0.tgz	Библиотека для удаления отладочной информации
105	86305283	cli-10.2.4.tgz	Инструментарий для быстрого создания приложений командной строки
106	bbd8dcbd	cli-12.2.18.tgz	Инструментарий для быстрого создания приложений командной строки
107	ab0582f9	cli-cursor-3.1.0.tgz	Библиотека для переключения курсора
108	68a9392a	clipboard-2.0.6.tgz	Библиотека для поддержки буфера обмена
109	b4c91f44	cli-spinners-2.6.1.tgz	Библиотека, обеспечивающая быстрый способ выбора одного значения из набора
110	29b5f0b2	cliui-7.0.4.tgz	Библиотека для создания сложных интерфейсов с поддержкой нескольких колонок
111	212af5b5	cli-width-3.0.0.tgz	Функция, получающая ширину окна стандартного вывода с двумя запасными вариантами
112	16b2e9af	clone-1.0.4.tgz	Функция, обеспечивающая клонирование объектов, типов, переменных и массивов
113	085ec7d4	clone-deep-4.0.1.tgz	Функция, обеспечивающая рекурсивное клонирование объектов, типов, переменных и массивов
114	b0e27369	codelyzer-6.0.2.tgz	Утилита проверки синтаксиса на соответствие спецификациям Angular
115	aa7de823	code-point-at-1.1.0.tgz	Утилита проверки синтаксиса строк
116	828a79ac	collection-visit-1.0.0.tgz	Функция метода взаимодействия между элементами в объекте или карте объектов в массиве
117	82e3f65d	color-convert-1.9.3.tgz	Функция преобразования цветов
118	01233155	colord-2.8.0.tgz	Утилита функции преобразования цветов
119	46abe84d	colorette-1.4.0.tgz	Утилиты преобразования цветов
120	42243ab7	color-name-1.1.3.tgz	Библиотека цветов в численном представлении
121	21bfa5d3	colors-1.4.0.tgz	Утилиты преобразования цветов
122	48694ec9	combined-stream-1.0.8.tgz	Реализация потока данных, который может порождать несколько других потоков последовательно
123	4707761b	commander-2.20.3.tgz	Библиотека поддержки для программ командной строки node.js
124	826d5086	commondir-1.0.1.tgz	Функция вычисления ближайшего общего родительского пути к файлам
125	0bcb4b5d	component-bind-1.0.0.tgz	Утилита привязки функций

№ п/п	КС CRC32	Имя пакета и версия	Назначение
126	8d796d82	component-emitter-1.3.0.tgz	Функция генератора событий
127	0eedf255	component-inherit-0.0.3.tgz	Утилита наследования прототипов
128	de496463	compressible-2.0.18.tgz	Утилита проверки типов в сжатых данных
129	9c4f4de0	compression-1.7.4.tgz	Реализация библиотеки сжатия данных
130	de680beb	concat-map-0.0.1.tgz	Утилита отображения данных при их слиянии
131	ef86ed5b	concat-stream-1.6.2.tgz	Реализация доступного для записи потока, который объединяет строки или двоичные данные и возвращает обратный вызов с результатом
132	625700b9	connect-3.7.0.tgz	Библиотека, обеспечивающая программное взаимодействие между компонентами
133	3dcf5b43	connect-history-api-fallback-1.6.0.tgz	Обработчик ошибочных объектов
134	c86d873c	console-control-strings-1.1.0.tgz	Библиотека позиционирования для текста, цвета и курсора
135	824425f5	content-disposition-0.5.3.tgz	Библиотека для создания и фильтрации заголовков
136	ce73d6d7	content-type-1.0.4.tgz	Обработчик заголовков HTTP в части типов данных
137	bb86953d	convert-source-map-1.7.0.tgz	Библиотека, преобразующая исходную карту данных потока из/в разные форматы и позволяющая добавлять или изменять свойства данных
138	ffd64b66	cookie-0.4.1.tgz	Библиотека для разбора и учета файла cookie (идентификатора сессии) HTTP-сервера
139	cac17059	cookie-signature-1.0.6.tgz	Библиотека для подписи (обеспечение контроля целостности) файла cookie (идентификатора сессии)
140	b8c94681	copy-anything-2.0.3.tgz	Библиотека поддержки копирования данных
141	dd452b3d	copy-concurrently-1.0.5.tgz	Библиотека поддержки копирования файлов, каталогов и символических ссылок с возможностью параллельного вызова и и поддержкой программного интерфейса win32
142	86db564b	copy-descriptor-0.1.1.tgz	Библиотека поддержки копирования метаданных
143	bdb3d56f	copy-webpack-plugin-9.0.1.tgz	Библиотека поддержки копирования данных
144	8e681a70	core-js-3.16.0.tgz	Основная стандартизированная библиотека, реализующая функции программного интерфейса
145	08b9a9f1	core-js-compat-3.18.2.tgz	Основная стандартизированная библиотека, реализующая функции программного интерфейса. Переносимая реализация
146	54be31fb	core-util-is-1.0.2.tgz	Набор функций программного интерфейса
147	eb57b467	cosmiconfig-7.0.1.tgz	Реализация поиска и загрузки конфигурации, согласно свойствам package.json, (файла rc или модуля CommonJS)
148	a963b85c	critters-0.0.10.tgz	Библиотека загрузки основных стилей
149	8d785ee3	cross-env-6.0.3.tgz	Библиотека поддержки сценариев и переменных среды
150	ecf6ff3e	cross-spawn-6.0.5.tgz	Независимая от платформы реализация библиотеки вызова и отслеживания дочерних процессов
151	1590f0ed	crypto-pro-2.1.0.tgz	Реализация программного интерфейса для взаимодействия с СКЗИ КриптоПРО
152	a95ac4ad	css-3.0.0.tgz	Парсер стилей
153	f42f30d6	cssauron-1.4.0.tgz	Библиотека, обеспечивающая выбор стиля и реализацию иерархии вложенных объектов на основе этого выбора
154	49811e0d	css-blank-pseudo-0.1.4.tgz	Библиотека элементов формы стилей
155	bdf022fc	css-color-names-1.0.1.tgz	Библиотека объектов JSON, содержащая имена цветов в стилях css, сопоставленные с их шестнадцатеричным значением
156	0eb17439	cssdb-4.4.0.tgz	Библиотека функций CSS
157	9c6bd537	css-declaration-sorter-6.1.3.tgz	Библиотека для автоматической сортировки объявлений CSS в заданном порядке
158	73ad8b77	cssesc-3.0.0.tgz	Библиотека JavaScript для экранирования строк и идентификаторов CSS при создании максимально короткого вывода только в формате ASCII
159	817f705e	css-has-pseudo-0.10.0.tgz	Библиотека относительных элементов стилей
160	bf6a4bae	css-loader-6.2.0.tgz	Модуль загрузки стилей
161	1c14983f	css-minimizer-webpack-plugin-3.0.2.tgz	Модуль загрузки стилей, библиотека поддержки

№ п/п	КС CRC32	Имя пакета и версия	Назначение
162	b3a2cf85	cssnano-5.0.8.tgz	Модуль для поддержки минимизации стилей
163	c5bcd531	cssnano-preset-default-5.1.4.tgz	Модуль для поддержки минимизации стилей, стандартная библиотека данных
164	69845dcd	cssnano-utils-2.0.1.tgz	Модуль для поддержки минимизации стилей, набор утилит
165	d0ed5053	csso-4.2.0.tgz	Модуль для поддержки минимизации стилей, с поддержкой ограничения структуры
166	19ce94ce	css-parse-2.0.0.tgz	Парсер стилей
167	f53126e1	css-prefers-color-scheme-3.1.1.tgz	Библиотека цветовой схемы стиля
168	3d21ac1c	css-select-4.1.3.tgz	Библиотека выбора стиля
169	1a26d7f2	css-selector-tokenizer-0.7.2.tgz	Библиотека выбора стиля, строковый обработчик
170	661f25a3	css-tree-1.1.3.tgz	Набор утилит для работы со стилями
171	1f233d5b	css-what-5.0.1.tgz	Парсер выбора стиля
172	5964c1b2	cuint-0.2.2.tgz	Библиотека поддержки беззнаковых целых чисел
173	2030bc5a	custom-event-1.0.1.tgz	Библиотека для создания произвольных событий
174	92c42905	cyclist-1.0.1.tgz	Реализация циклического списка
175	f4362478	damerau-levenshtein-1.0.6.tgz	Библиотека поддержки относительных расстояний
176	9e282355	dashdash-1.14.1.tgz	Парсер опций командного интерфейса
177	faa3e49e	date-format-3.0.0.tgz	Библиотека представления данных для даты и времени
178	2fa190f3	debug-4.3.1.tgz	Библиотека поддержки отладки
179	8fbe51b3	debuglog-1.0.1.tgz	Библиотека поддержки сообщений отладки
180	854a03aa	decamelize-1.2.0.tgz	Библиотека обработки строк с разделителями
181	2dbdd6d0	decode-uri-component-0.2.0.tgz	Библиотека, обеспечивающая представление данных URI
182	6a82966f	deep-equal-1.1.1.tgz	Библиотека, реализующая алгоритм сравнения неравенств
183	af8da785	deepmerge-4.2.2.tgz	Библиотека для глубокого (рекурсивного) слияния объектов Javascript
184	f2b6ee5f	default-gateway-4.2.0.tgz	Библиотека, обеспечивающая получение адреса шлюза (маршрута) по умолчанию
185	bbeff5b6	defaults-1.0.3.tgz	Библиотека, объединяющая значения по умолчанию со значениями объекта конфигурации
186	0358d4fa	define-lazy-prop-2.0.0.tgz	Библиотека определения свойств объекта
187	7c3fc565	define-properties-1.1.3.tgz	Библиотека определения нескольких неисчислимых свойств объекта за одну операцию
188	eacdfbd2	define-property-2.0.2.tgz	Библиотека определения нескольких неисчислимых свойств объекта
189	dfbf4786	del-4.1.1.tgz	Реализация интерфейса удаления файлов
190	b581439b	delayed-stream-1.0.0.tgz	Функция буферизации события(ий) из потока, обеспечивающая сохранение данных вплоть до события готовности их последующей обработки
191	cb91d38e	delegate-3.2.0.tgz	Библиотека обработки событий
192	d69b0a0c	delegates-1.0.0.tgz	Функция, делегирующая методы доступа к другому свойству объекта
193	0d529632	depd-1.1.2.tgz	Набор функций для операций отмены
194	4a944f40	dependency-graph-0.11.0.tgz	Библиотека для построения простых графов
195	9ea434bd	destroy-1.0.4.tgz	Утилита для уничтожения потока данных
196	0efaa893	detect-node-2.1.0.tgz	Функция надежного обнаружения
197	0a195028	dezalgo-1.0.4.tgz	Реализация асинхронного программного интерфейса
198	8bb9d8b1	di-0.0.1.tgz	Библиотека внедрения зависимостей для Node.js
199	d3957be1	diff-3.5.0.tgz	Утилита для сравнения файлов или данных
200	82c0d31b	dir-glob-3.0.1.tgz	Библиотека для преобразований имен каталогов в строки
201	5dd1d90f	dns-equal-1.0.0.tgz	Библиотека для сравнения записей службы имен
202	5eb57288	dns-packet-1.3.4.tgz	Модуль для кодирования/декодирования пакетов DNS
203	5428d5cf	dns-txt-2.0.2.tgz	Модуль для кодирования/декодирования записей DNS типа TXT
204	11ae3494	domelementtype-2.2.0.tgz	Модуль типизации узлов DOM
205	c7cb7520	domhandler-4.2.2.tgz	Обработчик значений DOM
206	3ab5d9ff	dom-serialize-2.2.1.tgz	Функция указателя значения узла DOM в строку

№ п/п	КС CRC32	Имя пакета и версия	Назначение
207	ecf4fe86	dom-serializer-1.3.2.tgz	Функция указателя значения узла DOM в строку
208	37d65676	domutils-2.8.0.tgz	Набор утилит для типов данных DOM
209	32d65cd0	duplexify-3.7.1.tgz	Реализация преобразования доступного для записи и чтения потока в дуплексный поток с поддержкой асинхронной инициализации ввода
210	79e7b109	ecc-jsbn-0.1.2.tgz	Функции контроля целостности данных потока
211	e8a5c7f2	ee-first-1.1.1.tgz	Функция, возвращающая первое событие из набора событий
212	ac3fa3ad	electron-to-chromium-1.3.860.tgz	Функция, предоставляющая список сопоставлений версий
213	9b271fd0	emoji-regex-8.0.0.tgz	Библиотека регулярных выражений
214	955bbf1f	emojis-list-3.0.0.tgz	Список регулярных выражений
215	d671024a	encodeurl-1.0.2.tgz	Интерпретатор адресов
216	d2ab6b70	encoding-0.1.13.tgz	Библиотека работы с кодировками
217	c50b8ed8	end-of-stream-1.4.4.tgz	Функция, определяющая конец потока данных
218	3df28220	engine.io-3.5.0.tgz	Библиотека, обеспечивающая двунаправленные соединения между клиентом и сервером
219	7a6031ba	engine.io-client-3.5.2.tgz	Библиотека, обеспечивающая двунаправленные соединения от клиента к серверу
220	067a612e	engine.io-parser-2.2.1.tgz	Парсер библиотеки двунаправленных соединений
221	2b2b01a2	enhanced-resolve-5.8.3.tgz	Расширения к библиотеке распознавания имен
222	239052c0	ent-2.2.0.tgz	Кодировщик объектов HTML
223	c91f92cc	entities-2.2.0.tgz	Кодировщик объектов HTML/XML
224	f61622e6	env-paths-2.2.1.tgz	Функция, возвращающая значение пути к объектам
225	2aba3555	err-code-1.1.2.tgz	Набор функций для обработки ошибок
226	f62825c6	err-code-2.0.3.tgz	Набор функций для обработки ошибок
227	0c9afe4e	errno-0.1.8.tgz	Набор функций для обработки ошибок
228	4aec63b8	error-ex-1.3.2.tgz	Набор функций для обработки ошибок
229	e4dc2b49	es6-promise-4.2.8.tgz	
230	d6b10a03	es6-promisify-5.0.0.tgz	
231	b1635053	es-abstract-1.20.4.tgz	
232	e7fdc8ee	es-array-method-boxes-properly-1.0.0.tgz	
233	ab89c082	esbuild-0.12.29.tgz	
234	5a878e64	esbuild-wasm-0.12.29.tgz	
235	32b85789	escalade-3.1.1.tgz	
236	3c9f8888	escape-html-1.0.3.tgz	
237	56406411	escape-string-regexp-1.0.5.tgz	
238	b7ca93d4	eslint-scope-5.1.1.tgz	
239	3d224bd7	es-module-lexer-0.7.1.tgz	
240	d7bbbbdb	esprima-4.0.1.tgz	
241	7c9cf28e	esrecurse-4.3.0.tgz	
242	43f25daa	es-to-primitive-1.2.1.tgz	
243	9b448625	estraverse-4.3.0.tgz	
244	d9eb45b4	estree-walker-2.0.2.tgz	
245	c4475052	esutils-2.0.3.tgz	
246	065f4a29	etag-1.8.1.tgz	
247	957f32c2	eventemitter3-4.0.4.tgz	
248	45ce7ea2	eventemitter-asyncresource-1.0.0.tgz	
249	1570a986	events-3.3.0.tgz	
250	68a6cd12	eventsource-1.1.0.tgz	
251	d9efc0f3	execa-1.0.0.tgz	
252	33650c63	exit-0.1.2.tgz	
253	b6b87229	expand-brackets-2.1.4.tgz	
254	417082cd	express-4.17.1.tgz	
255	5e4ea11e	extend-3.0.2.tgz	
256	67ac2b0d	extend-shallow-3.0.2.tgz	
257	0ce2955d	external-editor-3.1.0.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
258	62f45a98	extglob-2.0.4.tgz	
259	cd4e9033	extsprintf-1.3.0.tgz	
260	d2e75e10	fast-deep-equal-3.1.1.tgz	
261	e1084fd2	fast-glob-3.2.7.tgz	
262	624be616	fast-json-stable-stringify-2.1.0.tgz	
263	cc42f174	fastparse-1.1.2.tgz	
264	0e6a3ed2	fastq-1.13.0.tgz	
265	9fce858f	faye-websocket-0.11.4.tgz	
266	f8b613a6	figgy-pudding-3.5.2.tgz	
267	e7ed1c99	figures-3.2.0.tgz	
268	3449fe15	file-saver-2.0.2.tgz	
269	979cc10e	file-type-7.7.1.tgz	
270	7fead95f	fill-range-7.0.1.tgz	
271	c6bbb314	finalhandler-1.1.2.tgz	
272	5b9ad405	find-cache-dir-3.3.1.tgz	
273	20f80a28	find-parent-dir-0.3.1.tgz	
274	75826c56	find-up-4.1.0.tgz	
275	602779f8	flatpickr-4.6.3.tgz	
276	9e907d2b	flatted-2.0.2.tgz	
277	9a8bad2c	flatten-1.0.3.tgz	
278	03ff4e3a	flush-write-stream-1.1.1.tgz	
279	ceaee735	follow-redirects-1.11.0.tgz	
280	4bbf9ce9	forever-agent-0.6.1.tgz	
281	9aa2b6ea	for-in-1.0.2.tgz	
282	a9193609	form-data-2.3.3.tgz	
283	66c8560d	forwarded-0.2.0.tgz	
284	82c2dda7	fragment-cache-0.2.1.tgz	
285	fb74cc6f	fresh-0.5.2.tgz	
286	85c014c5	from2-2.3.0.tgz	
287	327d8763	fs-extra-6.0.1.tgz	
288	b74b4ae1	fs-minipass-2.1.0.tgz	
289	4ff03754	fs-monkey-1.0.3.tgz	
290	197d966e	fs.realpath-1.0.0.tgz	
291	d8fce70b	fs-write-stream-atomic-1.0.10.tgz	
292	38ddf70a	function-bind-1.1.1.tgz	
293	d3cf7f59	function.prototype.name-1.1.5.tgz	
294	027b4ef1	functions-have-names-1.2.3.tgz	
295	9c842003	gauge-2.7.4.tgz	
296	e67dc8ba	genfun-5.0.0.tgz	
297	78db2ff2	gensync-1.0.0-beta.2.tgz	
298	b5baf0b6	get-caller-file-2.0.5.tgz	
299	ecdf0019	get-intrinsic-1.1.1.tgz	
300	87471149	getpass-0.1.7.tgz	
301	6af6d437	get-stream-4.1.0.tgz	
302	3d45814d	get-symbol-description-1.0.0.tgz	
303	c438379c	get-value-2.0.6.tgz	
304	24e0c164	glob-7.1.7.tgz	
305	30e67357	globals-11.12.0.tgz	
306	3e5c2210	globby-11.0.4.tgz	
307	7b3ce6c8	glob-parent-5.1.2.tgz	
308	e19f9856	glob-to-regexp-0.4.1.tgz	
309	fb5287f7	good-listener-1.2.2.tgz	
310	b6f72fe9	graceful-fs-4.2.8.tgz	
311	cb1edd92	guid-typescript-1.0.9.tgz	
312	e5f59a2e	handle-thing-2.0.1.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
313	bde4be87	har-schema-2.0.0.tgz	
314	3113167d	har-validator-5.1.3.tgz	
315	4d0b33e8	has-1.0.3.tgz	
316	20850bc9	has-ansi-2.0.0.tgz	
317	5e8c3989	has-bigints-1.0.2.tgz	
318	76e7a852	has-binary2-1.0.3.tgz	
319	42c06780	has-cors-1.1.0.tgz	
320	6c7ea131	has-flag-3.0.0.tgz	
321	118f67ef	has-property-descriptors-1.0.0.tgz	
322	a8ece3dd	has-symbols-1.0.2.tgz	
323	57b46e97	has-tostringtag-1.0.0.tgz	
324	9f0deaa5	has-unicode-2.0.1.tgz	
325	9435806c	has-value-1.0.0.tgz	
326	73fde7f1	has-values-1.0.0.tgz	
327	fef621c7	hdr-histogram-js-2.0.1.tgz	
328	c869a775	hdr-histogram-percentiles-obj-3.0.0.tgz	
329	fe73f040	hosted-git-info-3.0.8.tgz	
330	2eab6afb	hosted-git-info-4.0.2.tgz	
331	c647bcbb	hpack.js-2.1.6.tgz	
332	6e47749b	html-entities-1.4.0.tgz	
333	7cc1a264	html-escaper-2.0.2.tgz	
334	a26b3527	http-cache-semantics-3.8.1.tgz	
335	d1f1f16a	http-cache-semantics-4.1.0.tgz	
336	8d972fb6	http-deceiver-1.2.7.tgz	
337	afd9a5b0	http-errors-1.7.2.tgz	
338	e93d052d	http-parser-js-0.5.3.tgz	
339	7eb5b4ea	http-proxy-1.18.1.tgz	
340	4502624d	http-proxy-agent-2.1.0.tgz	
341	99cb3770	http-proxy-agent-4.0.1.tgz	
342	5e78a2ca	http-proxy-middleware-0.19.1.tgz	
343	04df7653	http-signature-1.2.0.tgz	
344	dddae7e1	https-proxy-agent-2.2.4.tgz	
345	4830640d	humanize-ms-1.2.1.tgz	
346	3b852a95	husky-6.0.0.tgz	
347	ae60721a	iconv-lite-0.4.24.tgz	
348	565ec375	icss-utils-5.1.0.tgz	
349	4233e6f4	ieee754-1.2.1.tgz	
350	e769e295	iferr-0.1.5.tgz	
351	b450573d	ignore-5.1.8.tgz	
352	5f276659	ignore-walk-3.0.4.tgz	
353	d68e2eef	image-size-0.5.5.tgz	
354	ab4c135e	immediate-3.0.6.tgz	
355	b967f448	import-fresh-3.3.0.tgz	
356	dc8b788f	import-local-2.0.0.tgz	
357	f4683a87	imurmurhash-0.1.4.tgz	
358	470aa26a	indent-string-4.0.0.tgz	
359	cc87ccd6	indexes-of-1.0.1.tgz	
360	66fd969b	indexof-0.0.1.tgz	
361	95f8af7f	infer-owner-1.0.4.tgz	
362	8a398cf2	inflight-1.0.6.tgz	
363	c44a20c8	inherits-2.0.4.tgz	
364	6738394b	ini-1.3.6.tgz	
365	185bd6b6	injection-js-2.4.0.tgz	
366	33c025fd	inquirer-8.1.2.tgz	
367	51dbb012	internal-ip-4.3.0.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
368	4e06fd40	internal-slot-1.0.3.tgz	
369	5619f1a1	ip-1.1.5.tgz	
370	c130ba33	ipaddr.js-1.9.1.tgz	
371	cec1027c	ip-regex-2.1.0.tgz	
372	14ed058f	is-absolute-url-3.0.3.tgz	
373	53f0dd0f	is-accessor-descriptor-1.0.0.tgz	
374	72f38451	is-arguments-1.1.1.tgz	
375	01320fa4	isarray-1.0.0.tgz	
376	bba07a00	is-arrayish-0.2.1.tgz	
377	f1e1e606	is-bigint-1.0.4.tgz	
378	b4b59407	isbinaryfile-4.0.8.tgz	
379	5b0bf960	is-binary-path-2.1.0.tgz	
380	0c773236	is-boolean-object-1.1.2.tgz	
381	8120f89f	is-buffer-1.1.6.tgz	
382	086ab5e3	is-callable-1.2.7.tgz	
383	9c90f065	is-core-module-2.7.0.tgz	
384	db7b28f7	is-data-descriptor-1.0.0.tgz	
385	24354c87	is-date-object-1.0.5.tgz	
386	b3790af5	is-descriptor-1.0.2.tgz	
387	0b219f42	is-docker-2.2.1.tgz	
388	8421f358	isexe-2.0.0.tgz	
389	ac937f6f	is-extendable-1.0.1.tgz	
390	48d4b0d8	is-extglob-2.1.1.tgz	
391	25613163	is-fullwidth-code-point-1.0.0.tgz	
392	610d6de9	is-glob-4.0.3.tgz	
393	87225322	is-interactive-1.0.0.tgz	
394	1eab7ef0	is-lambda-1.0.1.tgz	
395	cd9d49c3	is-module-1.0.0.tgz	
396	92aca981	is-negative-zero-2.0.2.tgz	
397	190bee8e	is-number-7.0.0.tgz	
398	7e71ca19	is-number-object-1.0.7.tgz	
399	b6b3e3aa	isobject-3.0.1.tgz	
400	e8d5febd	isource-element-1.0.18.tgz	
401	8e26c513	is-path-cwd-2.2.0.tgz	
402	ec2ccf2e	is-path-in-cwd-2.1.0.tgz	
403	f389e11a	is-path-inside-2.1.0.tgz	
404	badb45a3	is-plain-object-2.0.4.tgz	
405	b74a6530	is-reference-1.2.1.tgz	
406	094581f1	is-regex-1.1.4.tgz	
407	fa57a798	is-resolvable-1.1.0.tgz	
408	07b1f02c	is-shared-array-buffer-1.0.2.tgz	
409	bf615090	isstream-0.1.2.tgz	
410	b1f033a9	is-stream-1.1.0.tgz	
411	69762cbb	is-string-1.0.7.tgz	
412	e0089871	is-symbol-1.0.4.tgz	
413	15146c5f	istanbul-lib-coverage-3.0.1.tgz	
414	12c70fd3	istanbul-lib-instrument-4.0.3.tgz	
415	b240f4cd	istanbul-lib-report-3.0.0.tgz	
416	1a24fe5e	istanbul-lib-source-maps-3.0.6.tgz	
417	9f37c640	istanbul-reports-3.0.2.tgz	
418	7511351c	is-typedarray-1.0.0.tgz	
419	a685c132	is-unicode-supported-0.1.0.tgz	
420	bf790109	is-weakref-1.0.2.tgz	
421	473b05c0	is-what-3.14.1.tgz	
422	7cf95d2c	is-windows-1.0.2.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
423	21a4988d	is-wsl-2.2.0.tgz	
424	0f2f3c46	jasmine-2.8.0.tgz	
425	4bdc973a	jasmine-allure-reporter-1.0.2.tgz	
426	d960f899	jasmine-core-3.6.0.tgz	
427	19c781d3	jasmine-spec-reporter-5.0.2.tgz	
428	e164e4dc	jasminewd2-2.2.0.tgz	
429	cdf6a1b9	jest-worker-27.2.4.tgz	
430	5ed51185	js2xmlparser-3.0.0.tgz	
431	304a5b77	jsbn-0.1.1.tgz	
432	e7fe3158	jsesc-2.5.2.tgz	
433	a7dd1bb1	json3-3.3.3.tgz	
434	5bfdf508	json5-2.1.3.tgz	
435	a36d36b1	jsonc-parser-3.0.0.tgz	
436	98db9cef	jsonfile-4.0.0.tgz	
437	541d9757	jsonparse-1.3.1.tgz	
438	4e45c0c7	json-parse-better-errors-1.0.2.tgz	
439	dbd132b5	json-parse-even-better-errors-2.3.1.tgz	
440	45dc9bb3	json-schema-0.2.3.tgz	
441	13fe6266	json-schema-traverse-0.4.1.tgz	
442	86b5326e	JSONStream-1.3.5.tgz	
443	9943f3c1	json-stringify-safe-5.0.1.tgz	
444	bd7dce60	jsprim-1.4.1.tgz	
445	e601486a	js-sha256-0.9.0.tgz	
446	89475376	js-tokens-4.0.0.tgz	
447	815155c1	js-yaml-3.14.0.tgz	
448	ed555d3f	jszip-3.4.0.tgz	
449	0d2e2d0b	karma-5.0.9.tgz	
450	f33e1b2e	karma-chrome-launcher-3.1.0.tgz	
451	4423474b	karma-coverage-istanbul-reporter-3.0.3.tgz	
452	8c31168a	karma-jasmine-4.0.1.tgz	
453	644c3512	karma-jasmine-html-reporter-1.5.4.tgz	
454	7efb3987	karma-source-map-support-1.4.0.tgz	
455	1fada49d	keycloak-angular-8.4.0.tgz	
456	1ac00693	keycloak-js-10.0.2.tgz	
457	d8000a7c	killable-1.0.1.tgz	
458	2792d053	kind-of-6.0.3.tgz	
459	8cb3e320	klona-2.0.4.tgz	
460	41bf6cda	less-4.1.1.tgz	
461	64a67b5f	less-loader-10.0.1.tgz	
462	1ec0827f	license-webpack-plugin-2.3.20.tgz	
463	7930ed98	lie-3.3.0.tgz	
464	9973ffd9	lilconfig-2.0.3.tgz	
465	57683b2b	lines-and-columns-1.1.6.tgz	
466	35aedcb8	loader-runner-4.2.0.tgz	
467	f59f0707	loader-utils-2.0.0.tgz	
468	9dc9891f	locate-path-5.0.0.tgz	
469	980690ad	lodash-4.17.21.tgz	
470	c65c9010	lodash.debounce-4.0.8.tgz	
471	622fbc1a	lodash-es-4.17.15.tgz	
472	3ec18e61	lodash.memoize-4.1.2.tgz	
473	82233601	lodash.uniq-4.5.0.tgz	
474	f14f944e	log4js-6.3.0.tgz	
475	e588f7b7	loglevel-1.7.1.tgz	
476	04e9ac34	log-symbols-4.1.0.tgz	
477	865c8bbd	lru-cache-6.0.0.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
478	2816a83f	magic-string-0.25.7.tgz	
479	d482eee6	make-dir-2.1.0.tgz	
480	6226a9d8	make-error-1.3.6.tgz	
481	696db835	make-fetch-happen-5.0.2.tgz	
482	cd9fd04a	make-fetch-happen-9.1.0.tgz	
483	9399cafb	map-age-cleaner-0.1.3.tgz	
484	7b42a69d	map-cache-0.2.2.tgz	
485	203a6268	map-visit-1.0.0.tgz	
486	267698dc	mdn-data-2.0.14.tgz	
487	4aa75f82	media-typer-0.3.0.tgz	
488	d985aeff	mem-8.1.1.tgz	
489	47263cc7	memfs-3.3.0.tgz	
490	ecba95cd	memory-fs-0.4.1.tgz	
491	82e4d00a	merge2-1.4.1.tgz	
492	c5615f21	merge-descriptors-1.0.1.tgz	
493	789243d5	merge-source-map-1.1.0.tgz	
494	cac53895	merge-stream-2.0.0.tgz	
495	20a819e2	methods-1.1.2.tgz	
496	0f9a9cab	micromatch-4.0.4.tgz	
497	7145ead6	mime-2.5.2.tgz	
498	eac9b78d	mime-db-1.50.0.tgz	
499	58574a5d	mime-types-2.1.33.tgz	
500	4fe41098	mimic-fn-2.1.0.tgz	
501	c3807f2b	mini-css-extract-plugin-2.2.1.tgz	
502	063c9399	minimalistic-assert-1.0.1.tgz	
503	d9d195a3	minimatch-3.0.4.tgz	
504	b257dae4	minimist-1.2.5.tgz	
505	88850753	minipass-3.1.5.tgz	
506	6eecf523	minipass-collect-1.0.2.tgz	
507	f626ac0d	minipass-fetch-1.4.1.tgz	
508	8668a568	minipass-flush-1.0.5.tgz	
509	f4590464	minipass-json-stream-1.0.1.tgz	
510	0ed03eb0	minipass-pipeline-1.2.4.tgz	
511	527e3f9e	minipass-sized-1.0.3.tgz	
512	e0bada46	minizlib-2.1.2.tgz	
513	88270c3a	mississippi-3.0.0.tgz	
514	482a5d96	mixin-deep-1.3.2.tgz	
515	a24cb25c	mkdirp-0.5.5.tgz	
516	ee4a636d	moment-2.26.0.tgz	
517	4db69aa1	move-concurrently-1.0.1.tgz	
518	2e7e06d1	ms-2.1.2.tgz	
519	87bc6acf	multicast-dns-6.2.3.tgz	
520	72315957	multicast-dns-service-types-1.1.0.tgz	
521	8e4fc6df	mute-stream-0.0.8.tgz	
522	69eb0384	nanocolors-0.1.12.tgz	
523	fab4309f	nanoid-3.1.29.tgz	
524	646f1727	nanomatch-1.2.13.tgz	
525	40c7c191	needle-2.9.1.tgz	
526	eb80705e	negotiator-0.6.2.tgz	
527	6ec4e6c1	neo-async-2.6.2.tgz	
528	946ca05e	ng2-file-upload-1.4.0.tgz	
529	4e0ed2b3	ng2-flatpickr-9.0.0.tgz	
530	5b386cd0	ng2-nouislider-1.8.2.tgz	
531	d08b7631	ng-packagr-12.2.2.tgz	
532	6aeafbf4	ngx-infinite-scroll-10.0.1.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
533	2c66a9ef	ngx-mask-12.0.0.tgz	
534	00d0701d	ngxs-reset-plugin-1.4.1.tgz	
535	1dc34783	ngx-virtual-scroller-4.0.3.tgz	
536	8e2731aa	nice-napi-1.0.2.tgz	
537	ad80b8f4	nice-try-1.0.5.tgz	
538	f914b589	node-addon-api-3.2.1.tgz	
539	c27b3590	node-fetch-npm-2.0.4.tgz	
540	c088d8a7	node-forge-0.10.0.tgz	
541	9dfd9c49	node-gyp-7.1.2.tgz	
542	5663ab34	node-gyp-build-4.3.0.tgz	
543	a862d5ce	node-releases-1.1.77.tgz	
544	04834204	node-sass-tilde-importer-1.0.2.tgz	
545	e218dcf2	nopt-5.0.0.tgz	
546	f52b623f	normalize-package-data-2.5.0.tgz	
547	8a304507	normalize-path-3.0.0.tgz	
548	efe069a9	normalize-range-0.1.2.tgz	
549	775a39fe	normalize-url-6.1.0.tgz	
550	9d6ca728	nouislider-14.5.0.tgz	
551	dfb3da06	npm-bundled-1.1.2.tgz	
552	00b2ab89	npm-install-checks-4.0.0.tgz	
553	16c750f3	npmlog-4.1.2.tgz	
554	fb5214ad	npm-normalize-package-bin-1.0.1.tgz	
555	249bda1f	npm-package-arg-8.0.1.tgz	
556	4984ed25	npm-package-arg-8.1.5.tgz	
557	7eee795b	npm-packlist-1.4.8.tgz	
558	b99434e2	npm-packlist-2.2.2.tgz	
559	071221c5	npm-pick-manifest-6.1.0.tgz	
560	ea48d769	npm-pick-manifest-6.1.1.tgz	
561	75a633f1	npm-registry-fetch-11.0.0.tgz	
562	d7c5a494	npm-registry-fetch-4.0.7.tgz	
563	925092c2	npm-run-path-2.0.2.tgz	
564	c028e8ad	nth-check-2.0.1.tgz	
565	51966335	num2fraction-1.2.2.tgz	
566	3822a82b	number-is-nan-1.0.1.tgz	
567	582eb407	oauth-sign-0.9.0.tgz	
568	3266b630	object-assign-4.1.1.tgz	
569	7fc9b2d0	object.assign-4.1.2.tgz	
570	b920c148	object-copy-0.1.0.tgz	
571	ab33ff22	object.getownpropertydescriptors-2.1.4.tgz	
572	a7ee67e4	object-inspect-1.12.2.tgz	
573	283f2528	object-is-1.1.5.tgz	
574	31a88169	object-keys-1.1.1.tgz	
575	26812dc0	object.pick-1.3.0.tgz	
576	cbec4cf7	object-visit-1.0.1.tgz	
577	313b3b38	obuf-1.1.2.tgz	
578	596da130	once-1.4.0.tgz	
579	a55396ca	onetime-5.1.2.tgz	
580	177e7ac5	on-finished-2.3.0.tgz	
581	9c3f4cac	on-headers-1.0.2.tgz	
582	d5c2f663	open-8.2.1.tgz	
583	7b5c8390	opencollective-postinstall-2.0.3.tgz	
584	a36a7c11	opn-5.5.0.tgz	
585	7b1edca8	ora-5.4.1.tgz	
586	f38d2afa	original-1.0.2.tgz	
587	fee71103	osenv-0.1.5.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
588	edd5dd11	os-homedir-1.0.2.tgz	
589	cd35c922	os-tmpdir-1.0.2.tgz	
590	4a91d841	pacote-11.3.5.tgz	
591	7bdfaa23	pacote-9.5.12.tgz	
592	fe37e0cd	pako-1.0.11.tgz	
593	0bf01963	parallel-transform-1.2.0.tgz	
594	33365e04	parent-module-1.0.1.tgz	
595	ac3e472d	parse5-6.0.1.tgz	
596	687331e0	parse5-htmlparser2-tree-adapter-6.0.1.tgz	
597	bd79d13e	parse5-html-rewriting-stream-6.0.1.tgz	
598	0edc8128	parse5-sax-parser-6.0.1.tgz	
599	74eaf5d5	parse-json-5.2.0.tgz	
600	e5477c7f	parse-node-version-1.0.1.tgz	
601	ec3e4212	parseqs-0.0.6.tgz	
602	e70ce530	parseuri-0.0.6.tgz	
603	fa3c7bef	parseurl-1.3.3.tgz	
604	a2e7e003	pascalcase-0.1.1.tgz	
605	4b17a0de	path-dirname-1.0.2.tgz	
606	c773940f	path-exists-3.0.0.tgz	
607	af66187f	path-is-absolute-1.0.1.tgz	
608	fcf75133	path-is-inside-1.0.2.tgz	
609	e2fff0be	path-key-2.0.1.tgz	
610	4d5ab27b	path-parse-1.0.6.tgz	
611	cc2555bc	path-to-regexp-0.1.7.tgz	
612	73e08728	path-type-4.0.0.tgz	
613	0e13bb0f	p-defer-1.0.0.tgz	
614	c07e322d	performance-now-2.1.0.tgz	
615	abe86b17	p-finally-1.0.0.tgz	
616	82ff6ba9	picocolors-0.2.1.tgz	
617	9b3601ef	picomatch-2.3.0.tgz	
618	b08112af	pify-4.0.1.tgz	
619	975ffc70	pinkie-2.0.4.tgz	
620	21f984be	pinkie-promise-2.0.1.tgz	
621	e9fae476	piscina-3.1.0.tgz	
622	932202cd	pkg-dir-3.0.0.tgz	
623	6099bef7	p-limit-3.1.0.tgz	
624	d469e569	p-locate-4.1.0.tgz	
625	0b7802e2	p-map-4.0.0.tgz	
626	f72c4220	popper.js-1.16.1.tgz	
627	67882466	portfinder-1.0.28.tgz	
628	2c7fd5dd	posix-character-classes-0.1.1.tgz	
629	ba61ca17	postcss-8.3.6.tgz	
630	8e03e260	postcss-attribute-case-insensitive-4.0.2.tgz	
631	8a81f960	postcss-calc-8.0.0.tgz	
632	e9b1f17f	postcss-color-functional-notation-2.0.1.tgz	
633	7d25bc4e	postcss-color-gray-5.0.0.tgz	
634	6135720f	postcss-color-hex-alpha-5.0.3.tgz	
635	5cf6da79	postcss-colormin-5.2.0.tgz	
636	f4d7fd22	postcss-color-mod-function-3.0.3.tgz	
637	678d4eb9	postcss-color-rebeccapurple-4.0.1.tgz	
638	ac78665e	postcss-convert-values-5.0.1.tgz	
639	fea4f5dc	postcss-custom-media-7.0.8.tgz	
640	2e4f6e09	postcss-custom-properties-8.0.11.tgz	
641	f3934f41	postcss-custom-selectors-5.1.2.tgz	
642	e8aef5ec	postcss-dir-pseudo-class-5.0.0.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
643	5170b199	postcss-discard-comments-5.0.1.tgz	
644	c92f4b52	postcss-discard-duplicates-5.0.1.tgz	
645	3770f795	postcss-discard-empty-5.0.1.tgz	
646	ee3df552	postcss-discard-overridden-5.0.1.tgz	
647	7d5f16b4	postcss-double-position-gradients-1.0.0.tgz	
648	c51186fe	postcss-env-function-2.0.2.tgz	
649	7187b8ad	postcss-focus-visible-4.0.0.tgz	
650	a55b68c4	postcss-focus-within-3.0.0.tgz	
651	7aa7449a	postcss-font-variant-4.0.1.tgz	
652	7ffb0157	postcss-gap-properties-2.0.0.tgz	
653	c8edeb8a	postcss-image-set-function-3.0.1.tgz	
654	ab1c9723	postcss-import-14.0.2.tgz	
655	0f66bde6	postcss-initial-3.0.4.tgz	
656	844f1a56	postcss-lab-function-2.0.1.tgz	
657	b98b6d3c	postcss-loader-6.1.1.tgz	
658	3e70ea90	postcss-logical-3.0.0.tgz	
659	d4754460	postcss-media-minmax-4.0.0.tgz	
660	120ecdaa	postcss-merge-longhand-5.0.2.tgz	
661	a5751d8d	postcss-merge-rules-5.0.2.tgz	
662	05cd0965	postcss-minify-font-values-5.0.1.tgz	
663	8bde5256	postcss-minify-gradients-5.0.2.tgz	
664	64ba8063	postcss-minify-params-5.0.1.tgz	
665	456c9993	postcss-minify-selectors-5.1.0.tgz	
666	a96f0d45	postcss-modules-extract-imports-3.0.0.tgz	
667	6efd8d5f	postcss-modules-local-by-default-4.0.0.tgz	
668	6bf5ec46	postcss-modules-scope-3.0.0.tgz	
669	493bb9cb	postcss-modules-values-4.0.0.tgz	
670	f2644940	postcss-nesting-7.0.1.tgz	
671	fa659bc5	postcss-normalize-charset-5.0.1.tgz	
672	58e0ad05	postcss-normalize-display-values-5.0.1.tgz	
673	c218aad4	postcss-normalize-positions-5.0.1.tgz	
674	1b1fae0e	postcss-normalize-repeat-style-5.0.1.tgz	
675	d08ca7be	postcss-normalize-string-5.0.1.tgz	
676	faf00f94	postcss-normalize-timing-functions-5.0.1.tgz	
677	89a823b9	postcss-normalize-unicode-5.0.1.tgz	
678	e5507301	postcss-normalize-url-5.0.2.tgz	
679	44d4f349	postcss-normalize-whitespace-5.0.1.tgz	
680	e83145ac	postcss-ordered-values-5.0.2.tgz	
681	86d0aeea	postcss-overflow-shorthand-2.0.0.tgz	
682	ab289122	postcss-page-break-2.0.0.tgz	
683	bfe1f50a	postcss-place-4.0.1.tgz	
684	1eb4c895	postcss-preset-env-6.7.0.tgz	
685	3e4dba76	postcss-pseudo-class-any-link-6.0.0.tgz	
686	d14ab105	postcss-reduce-initial-5.0.1.tgz	
687	4837b12d	postcss-reduce-transforms-5.0.1.tgz	
688	d68371e7	postcss-replace-overflow-wrap-3.0.0.tgz	
689	76b1d81a	postcss-selector-matches-4.0.0.tgz	
690	fa6c8a2d	postcss-selector-not-4.0.1.tgz	
691	299c4b97	postcss-selector-parser-6.0.6.tgz	
692	71afc08f	postcss-svgo-5.0.2.tgz	
693	5223757b	postcss-unique-selectors-5.0.1.tgz	
694	24c2ccfa	postcss-url-10.1.3.tgz	
695	24a654d2	postcss-value-parser-4.1.0.tgz	
696	e196011b	postcss-values-parser-2.0.1.tgz	
697	f02c76ae	p-retry-3.0.1.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
698	6f727c95	pretty-bytes-5.6.0.tgz	
699	34084614	prismjs-1.20.0.tgz	
700	903c7161	prism-themes-1.4.0.tgz	
701	f7a1fc55	process-nextick-args-2.0.1.tgz	
702	ec295173	promise-inflight-1.0.1.tgz	
703	3c48cc65	promise-retry-1.1.1.tgz	
704	0d47f775	promise-retry-2.0.1.tgz	
705	2c8a45e5	proper-lockfile-4.1.2.tgz	
706	e20077d2	protoduck-5.0.1.tgz	
707	e6d521f7	protractor-7.0.0.tgz	
708	4853197a	proxy-addr-2.0.7.tgz	
709	08b75d31	prr-1.0.1.tgz	
710	373539d9	psl-1.8.0.tgz	
711	cfa02d6e	p-try-2.2.0.tgz	
712	60f4f9b6	pump-3.0.0.tgz	
713	ea03ab46	pumpify-1.5.1.tgz	
714	84772292	punycode-2.1.1.tgz	
715	e490c77c	qjobs-1.2.0.tgz	
716	2f6c965c	qs-6.5.2.tgz	
717	4ea39999	querystring-0.2.0.tgz	
718	843501d3	querystringify-2.2.0.tgz	
719	bf56eb0e	queue-microtask-1.2.3.tgz	
720	f3f0a140	randombytes-2.1.0.tgz	
721	20d0840d	range-parser-1.2.1.tgz	
722	48b5d8e0	raw-body-2.4.0.tgz	
723	d960653c	readable-stream-2.3.7.tgz	
724	3133f694	read-cache-1.0.0.tgz	
725	04758b1a	readdirp-3.6.0.tgz	
726	3ea54d73	readdir-scoped-modules-1.1.0.tgz	
727	800ae635	read-package-json-2.1.2.tgz	
728	71039f03	read-package-json-fast-2.0.3.tgz	
729	2efeba7b	read-package-tree-5.3.1.tgz	
730	80af320f	reflect-metadata-0.1.13.tgz	
731	6f71a8ee	regenerate-1.4.2.tgz	
732	178e3c88	regenerate-unicode-properties-9.0.0.tgz	
733	9c3ac74e	regenerator-runtime-0.13.9.tgz	
734	9e2d4ef7	regenerator-transform-0.14.5.tgz	
735	74aa3ffc	regex-not-1.0.2.tgz	
736	0047c74e	regex-parser-2.2.11.tgz	
737	d8dfb2c1	regexp.prototype.flags-1.3.1.tgz	
738	abe8cd3d	regexpu-core-4.8.0.tgz	
739	c9e2308f	regjsgen-0.5.2.tgz	
740	143d8dec	regjsparser-0.7.0.tgz	
741	79b91b2d	remove-trailing-separator-1.1.0.tgz	
742	0f2b876d	repeat-element-1.1.4.tgz	
743	6e5c041c	repeat-string-1.6.1.tgz	
744	75c338ef	request-2.88.2.tgz	
745	e9f4478c	require-directory-2.1.1.tgz	
746	fe32a18a	require-from-string-2.0.2.tgz	
747	6a6bacff	require-main-filename-2.0.0.tgz	
748	633ee110	requires-port-1.0.0.tgz	
749	86204cbd	resolve-1.20.0.tgz	
750	60504a45	resolve-cwd-2.0.0.tgz	
751	5cb6e9c4	resolve-from-4.0.0.tgz	
752	b7142068	resolve-url-0.2.1.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
753	8dddba09	resolve-url-loader-4.0.0.tgz	
754	fad0eb32	restore-cursor-3.1.0.tgz	
755	60e768dd	ret-0.1.15.tgz	
756	ffdf37d4	retry-0.12.0.tgz	
757	51664565	reusify-1.0.4.tgz	
758	8c49ce7b	rfdc-1.3.0.tgz	
759	725c4622	rimraf-3.0.2.tgz	
760	c7f51032	rollup-2.58.0.tgz	
761	77c4d801	run-async-2.4.1.tgz	
762	c886bf20	run-parallel-1.2.0.tgz	
763	06e839de	run-queue-1.0.3.tgz	
764	b62aaccf	rxjs-6.6.7.tgz	
765	419a4877	safe-buffer-5.1.2.tgz	
766	71c68b6f	safer-buffer-2.1.2.tgz	
767	59dde1bb	safe-regex-1.1.0.tgz	
768	36e4b3e1	safe-regex-test-1.0.0.tgz	
769	55fd2549	sass-1.36.0.tgz	
770	a922ba62	sass-loader-12.1.0.tgz	
771	2d868c85	saucelabs-1.5.0.tgz	
или 772	e327a66e	sax-1.2.4.tgz	
773	902c6026	schema-utils-2.7.1.tgz	
774	62c83013	select-1.1.2.tgz	
775	f8a255c2	select-hose-2.0.0.tgz	
776	2052d721	selenium-webdriver-3.6.0.tgz	
777	076f0c33	selfsigned-1.10.11.tgz	
778	1b5ba59f	semver-7.3.2.tgz	
779	1afc7cd8	semver-dsl-1.0.1.tgz	
780	87124fa5	semver-intersect-1.4.0.tgz	
781	58959346	send-0.17.1.tgz	
782	60aedb70	serialize-javascript-6.0.0.tgz	
783	72d0d2a6	serve-index-1.9.1.tgz	
784	3719363c	serve-static-1.14.1.tgz	
785	2be2c996	set-blocking-2.0.0.tgz	
786	8c076b78	set-immediate-shim-1.0.1.tgz	
787	12f87de4	setprototypeof-1.1.1.tgz	
788	133b398d	set-value-2.0.1.tgz	
789	53413f4e	shallow-clone-3.0.1.tgz	
790	bda90dcd	shebang-command-1.2.0.tgz	
791	07b0a445	shebang-regex-1.0.0.tgz	
792	ebb78529	side-channel-1.0.4.tgz	
793	7a3fc7ff	signal-exit-3.0.3.tgz	
794	6a7d6c02	slash-3.0.0.tgz	
795	3a60f222	smart-buffer-4.2.0.tgz	
796	9c873c88	snapdragon-0.8.2.tgz	
797	c6fd41b9	snapdragon-node-2.1.1.tgz	
798	0ba8a62c	snapdragon-util-3.0.1.tgz	
или 799	ee43a60f	socket.io-2.4.1.tgz	
800	b300b0bd	socket.io-adapter-1.1.2.tgz	
801	edb8813b	socket.io-client-2.4.0.tgz	
802	7270a4cf	socket.io-parser-3.4.1.tgz	
803	9812d8fb	sockjs-0.3.21.tgz	
804	3b09faaa	sockjs-client-1.5.2.tgz	
805	3802da4b	socks-2.3.3.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
806	8b56d753	socks-2.6.1.tgz	
807	edb71bb8	socks-proxy-agent-4.0.2.tgz	
808	59b25bfb	socks-proxy-agent-6.1.0.tgz	
809	16744b8f	source-list-map-2.0.1.tgz	
810	d0564da7	source-map-0.7.3.tgz	
811	4d6fcd70	sourcemap-codec-1.4.8.tgz	
812	fedf71cf	source-map-js-0.6.2.tgz	
813	ddbfb259	source-map-loader-3.0.0.tgz	
814	44f99084	source-map-resolve-0.6.0.tgz	
815	304f58ce	source-map-support-0.5.19.tgz	
816	241f6e55	source-map-url-0.4.1.tgz	
817	7c71e56c	spdx-correct-3.1.1.tgz	
818	c58c9c0d	spdx-exceptions-2.3.0.tgz	
819	9b0bdc0e	spdx-expression-parse-3.0.1.tgz	
820	e52657a9	spdx-license-ids-3.0.12.tgz	
821	26b9e34a	spdy-4.0.2.tgz	
822	01f79ac8	spdy-transport-3.0.0.tgz	
823	fcf0bfe3	split-string-3.1.0.tgz	
824	c2bb810d	sprintf-js-1.0.3.tgz	
825	17243487	sshpk-1.16.1.tgz	
826	9d67fe08	ssri-8.0.1.tgz	
827	faa48a10	stable-0.1.8.tgz	
828	ee170a7a	static-extend-0.1.2.tgz	
829	453f2a75	statuses-1.5.0.tgz	
830	77653ed4	stream-each-1.2.3.tgz	
831	7e60d993	streamroller-2.2.4.tgz	
832	7d62bb7f	stream-shift-1.0.1.tgz	
833	13ddc01e	string_decoder-1.1.1.tgz	
834	782c070e	string.prototype.trimend-1.0.5.tgz	
835	295495a5	string.prototype.trimstart-1.0.5.tgz	
836	915fe907	string-width-4.2.3.tgz	
837	f1307a15	strip-ansi-3.0.1.tgz	
838	5ca8cfa8	strip-eof-1.0.0.tgz	
839	5bfce55a	stylehacks-5.0.1.tgz	
840	eef57123	style-loader-3.2.1.tgz	
841	e7ce12a4	stylus-0.54.8.tgz	
842	26c209f5	stylus-loader-6.1.0.tgz	
843	ab5f40d1	supports-color-5.5.0.tgz	
844	15204257	svgo-2.7.0.tgz	
845	fa8ea453	symbol-observable-1.2.0.tgz	
846	07453a73	symbol-observable-4.0.0.tgz	
847	122ab742	tapable-2.2.1.tgz	
848	c4dc0fc4	tar-6.1.11.tgz	
849	dc61825c	terser-5.7.1.tgz	
850	1a5b2ad1	terser-webpack-plugin-5.1.4.tgz	
851	4d5d99c4	text-mask-addons-3.8.0.tgz	
852	6ab7550d	text-mask-core-5.1.2.tgz	
853	00cc0b8e	text-table-0.2.0.tgz	
854	2fab6d13	through2-2.0.5.tgz	
855	f7ee24ef	through-2.3.8.tgz	
856	2c79426f	thunky-1.1.0.tgz	
857	5bc83735	timsort-0.3.0.tgz	
858	f75402ac	tiny-emitter-2.1.0.tgz	
859	b944ba92	tmp-0.0.33.tgz	
860	76fb5ca1	to-array-0.1.4.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
861	063627c4	to-fast-properties-2.0.0.tgz	
862	0eb0c6d2	toidentifier-1.0.0.tgz	
863	0225446c	to-object-path-0.3.0.tgz	
864	338e7b1c	to-regex-3.0.2.tgz	
865	9020be21	to-regex-range-5.0.1.tgz	
866	e1d98a7e	tough-cookie-2.5.0.tgz	
867	19451cce	tree-kill-1.2.2.tgz	
868	38f53225	tslib-2.3.1.tgz	
869	c08d9a5d	tslint-6.1.3.tgz	
870	464a38d6	ts-node-7.0.1.tgz	
871	be05b44d	tsutils-2.29.0.tgz	
872	94efb39b	tunnel-agent-0.6.0.tgz	
873	15c7cc6a	tweetnacl-0.14.5.tgz	
874	54fca6f3	typedarray-0.0.6.tgz	
875	10fb6339	type-fest-0.21.3.tgz	
876	b28a04e5	type-is-1.6.18.tgz	
877	d87a5a1e	typescript-4.3.5.tgz	
878	58a586a4	ua-parser-js-0.7.21.tgz	
879	abe91007	unbox-primitive-1.0.2.tgz	
880	397e99f7	unicode-canonical-property-names-ecmascript-2.0.0.tgz	
881	6476f898	unicode-match-property-ecmascript-2.0.0.tgz	
882	4c0aa539	unicode-match-property-value-ecmascript-2.0.0.tgz	
883	d313e844	unicode-property-aliases-ecmascript-2.0.0.tgz	
884	26a799bf	union-value-1.0.1.tgz	
885	de4c5e66	uniq-1.0.1.tgz	
886	9179deba	uniqs-2.0.0.tgz	
887	1de80a14	unique-filename-1.1.1.tgz	
888	808fba21	unique-slug-2.0.2.tgz	
889	0c3b6297	universal-analytics-0.4.23.tgz	
890	bc7bba8a	universalify-0.1.2.tgz	
891	1eb2d8f0	unpipe-1.0.0.tgz	
892	d197a45c	unset-value-1.0.0.tgz	
893	a585edae	upath-1.2.0.tgz	
894	5c371c50	uri-js-4.2.2.tgz	
895	f877a0e1	urix-0.1.0.tgz	
896	10703347	url-0.11.0.tgz	
897	3d6cb49b	url-parse-1.5.3.tgz	
898	63057a3e	use-3.1.1.tgz	
899	3e9ed024	util-deprecate-1.0.2.tgz	
900	548b50da	util-promisify-2.1.0.tgz	
901	0b9e3759	utils-merge-1.0.1.tgz	
902	06a4f474	uuid-3.4.0.tgz	
903	75a75a71	uxg-1.0.18.tgz	
904	3b09d517	validate-npm-package-license-3.0.4.tgz	
905	8c2b4f5b	validate-npm-package-name-3.0.0.tgz	
906	4796964e	vary-1.1.2.tgz	
907	b5862276	vendors-1.0.4.tgz	
908	c0073600	verror-1.10.0.tgz	
909	a213f40e	void-elements-2.0.1.tgz	
910	4b08e7f5	watchpack-2.2.0.tgz	
911	5d9c895b	wbuf-1.7.3.tgz	
912	71e83f10	wcwidth-1.0.1.tgz	
913	2cf464e1	webdriver-js-extender-2.1.0.tgz	
914	d750db24	webpack-5.50.0.tgz	
915	be65b5c4	webpack-dev-middleware-5.0.0.tgz	

№ п/п	КС CRC32	Имя пакета и версия	Назначение
916	55a72d25	webpack-dev-server-3.11.2.tgz	
917	d3e83de6	webpack-log-2.0.0.tgz	
918	db69aa93	webpack-merge-5.8.0.tgz	
919	166b54f2	webpack-sources-1.4.3.tgz	
920	ba9f2587	webpack-subresource-integrity-1.5.2.tgz	
921	6a083447	websocket-driver-0.7.4.tgz	
922	5660cd0f	websocket-extensions-0.1.4.tgz	
923	8f52c51a	which-1.3.1.tgz	
924	b086588e	which-boxed-primitive-1.0.2.tgz	
925	44c24d8c	which-module-2.0.0.tgz	
926	7cf0d7f2	wide-align-1.1.3.tgz	
927	1ef6def2	wildcard-2.0.0.tgz	
928	6c2d781c	wrap-ansi-7.0.0.tgz	
929	2173f9b5	wrappy-1.0.2.tgz	
930	95744004	ws-7.4.6.tgz	
931	b3c2eb62	xml2js-0.4.23.tgz	
932	cb52de0f	xmlbuilder-11.0.1.tgz	
933	d26dce6a	xmlcreate-1.0.2.tgz	
934	8bcd083f	xmlhttprequest-ssl-1.6.3.tgz	
935	5f8e5643	xtend-4.0.2.tgz	
936	4544bbdd	xxhashjs-0.2.2.tgz	
937	81ccedc7	y18n-4.0.0.tgz	
938	6a9d2a7e	yallist-4.0.0.tgz	
939	9638ea06	yaml-1.10.2.tgz	
940	e93cf667	yargs-15.4.1.tgz	
941	e2bcb5b8	yargs-parser-20.2.9.tgz	
942	17264886	yeast-0.1.2.tgz	
943	bd97756b	yn-2.0.0.tgz	
944	85e134d4	yocto-queue-0.1.0.tgz	
945	2970804a	zone.js-0.11.4.tgz	

Таблица 21: Состав заимствованного и привлекаемого ПО. NPM пакеты.

Пользователи (в т.ч. технологические), а также их пароли, применяемые по умолчанию, указаны в таблице Таблица 22:

№ п/п	Имя пользователя	Пароль по умолчанию	Назначение пользователя, узел (адрес) обращения
1	admin	mirrors_in_SKY20w20	Администратор провайдера идентификации и аутентификации KeyCloak (sbp99tp8394-06.techpark.local 10.52.62.18)
2	processor-test@isource.ru	processor-test	Администратор модуля закупок (sbp99tp8394-01.techpark.local 10.50.62.42)
3	processor	processor	Технологический пользователь СУБД PostgreSQL, использующийся для взаимодействия с модулем закупок
4	keycloak	keycloak230jf8rejerf	Технологический пользователь СУБД PostgreSQL, использующийся для взаимодействия с KeyCloak
5	root	inspector-local-db-pass	Технологический пользователь СУБД MySQL, использующийся для взаимодействия с модулем Цифрового инспектора
6	processor	processor	Технологический пользователь СУБД PostgreSQL, использующийся для взаимодействия с модулем сервиса очередей сообщений RabbitMQ (sbp99tp8394-05.techpark.local 10.52.62.23)
7	planning	planning	Технологический пользователь СУБД PostgreSQL, использующийся для взаимодействия с модулем планирования ()
8	camunda	camunda	Технологический пользователь СУБД PostgreSQL, использующийся для взаимодействия с модулем планирования ()

№ п/п	Имя пользователя	Пароль по умолчанию	Назначение пользователя, узел (адрес) обращения
9	sed	sed	Технологический пользователь СУБД PostgreSQL, использующийся для взаимодействия с договорным модулем ()
10	admin	password	Технологический пользователь (администратор) файлового сервиса (хранилища) nexus (sbp99tp8394-02.techpark.local 10.50.62.13)
11	docgen	docgen	Технологический пользователь СУБД PostgreSQL, использующийся для взаимодействия с сервисом генерации документов
12	t_admin@ci.tst	CI-admin-!	Администратор модуля цифрового инспектора (spb99tp8394-20.techpark.local)
13	t_client1@ci.tst	CI-client1-!	Пользователь модуля цифрового инспектора (spb99tp8394-20.techpark.local)
14	t_logistician@ci.tst	CI-logistician-!	Пользователь модуля цифрового инспектора (spb99tp8394-20.techpark.local)
15	t_client2@ci.tst	CI-client2-!	Пользователь модуля цифрового инспектора (spb99tp8394-20.techpark.local)
16	t_inspector@ci.tst	CI-inspector-!	Пользователь модуля цифрового инспектора (spb99tp8394-20.techpark.local)
17	t_auditor1@ci.tst	CI-auditor1-!	Пользователь модуля цифрового инспектора (spb99tp8394-20.techpark.local)
18	t_auditor2@ci.tst	CI-auditor2-!	Пользователь модуля цифрового инспектора (spb99tp8394-20.techpark.local)
19	t_provider@ci.tst	CI-provider-!	Пользователь модуля цифрового инспектора (spb99tp8394-20.techpark.local)
20	t_rest_integration@ci.tst	CI-rest_integration-!	Технологический пользователь модуля цифрового инспектора (spb99tp8394-20.techpark.local)
21	t_rest_processor@ci.tst	CI-rest_processor-!	Технологический пользователь модуля цифрового инспектора для взаимодействия с модулем закупок (spb99tp8394-20.techpark.local)
22	t_inspectorsupervisor@ci.tst	CI-inspectorsupervisor-!	Пользователь модуля цифрового инспектора (spb99tp8394-20.techpark.local)
23	user-backoffice@main.ruz	mirrors_in_SKY50t50	Пользователь закупочного модуля, используемый в целях проведения испытаний.
24	*-service	–	Пользователи с указанным суффиксом используются в интересах межсервисного взаимодействия.

Таблица 22: Перечень пользователей (в т.ч. технологических) и паролей, применяемых по умолчанию.

Привилегия	Гость	Заказчик	Заказчик согласующий	Согласующий финблока	Бэк-офис старший менеджер	Бэк-офис младший менеджер	Бэк-офис наблюдатель	Администратор
Получение доступа	да	да	да	да	да	да	да	да
Смена пароля	да	да	да	да	да	да	да	да
Регистрация (создание) контрагента	нет	нет	нет	нет	да	да	нет	да
Редактирование контрагента	нет	нет	нет	нет	да	да	нет	да
Просмотр списка контрагентов	нет	да для поставщиков	нет	да для поставщиков и заказчиков	да для поставщиков и заказчиков	да для поставщиков и заказчиков	нет	да
Привязка ответственного к контрагенту	нет	нет	нет	нет	да	да	нет	да

Привилегия	Гость	Заказчик	Заказчик согласующий	Согласующий финблока	Бэк-офис старший менеджер	Бэк-офис младший менеджер	Бэк-офис наблюдатель	Администратор
Просмотр списка пользователей бэк-офиса	нет	нет	нет	нет	да	нет	нет	нет
Просмотр карточки контрагента	нет	да для поставщиков	нет	да	да для поставщиков и заказчиков	да для поставщиков и заказчиков	нет	да

Таблица 23: Роли в модуле закупок.

Перечень образов для контейнеров:

№ п/п	КС CRC32	Имя образа и версия	Назначение
1		kube-proxy v1.23.7	Сетевой прокси для k8s
2		kube-apiserver v1.23.7	Реализация программного интерфейса для k8s
3		kube-controller-manager v1.23.7	Служба, управляющая ЖЦ контейнеров k8s
4		kube-scheduler v1.23.7	Служба планировщика k8s
5		calico/node v3.22.3	Узел управления сетевыми настройками для k8s
6		calico/cni v3.22.3	Наборы шаблонов и плагинов сетевых настроек k8s
7		calico/kube-controllers v3.22.3	Модуль мониторинга и управления программного интерфейса k8s
8		calico/pod2daemon-flexvol v3.22.3	Модуль, обеспечивающий безопасные сетевые соединения в кластере k8s
9		kubernetesui/metrics-scraper v1.0.7	Модуль, обеспечивающий сбор, очистку и хранение метрик для k8s
10		metrics-server/metrics-server v1.0.7	Модуль источника метрик ресурсов контейнера для встроенных конвейеров автоматического масштабирования k8s
11		pause v3.3	Образ эталонного контейнера, который загружает данные для создания всех контрольных групп, системного окружения и пространств имен до того, как будут созданы отдельные экземпляры целевых контейнеров
12		coreos/etcd v3.5.3	Надежное хранилище ключевых значений для наиболее важных данных распределенной системы
13		coredns/coredns v1.8.6	Реализация сервера имен на языке Go, обеспечивающая обработку и перенаправление запросов DNS в интересах кластера k8s
14		dns/k8s-dns-node-cache v1.21.1	Модуль, обеспечивающий кеширование запросов DNS в интересах оптимизации производительности кластера k8s
15		cpa/cluster-proportional-autoscaler-amd64 v1.8.5	Модуль, отслеживающий количество планируемых узлов и ядер кластера k8s, изменяющий размер реплик для требуемого ресурса
16		sig-storage/nfsplugin v4.1.0	Динамический поставщик томов, обеспечивающий взаимодействие k8s и сервера NFS
17		sig-storage/csi-provisioner v3.2.0	Модуль, отслеживающий API-сервер Kubernetes на наличие объектов PersistentVolumeClaim
18		sig-storage/livenessprobe v2.7.0	Модуль, обеспечивающий единую точку подключения для веб-запросов к кластеру k8s
19		sig-storage/csi-node-driver-registrar v2.5.1	Модуль, обеспечивающий взаимодействие между подключениями и точками хранения данных
20		ingress-nginx/controller v1.3.1	Модуль k8s, использующий веб-сервер nginx в качестве обратного прокси и балансировщика нагрузки
21		ingress-nginx/kube-webhook-certgen v1.3.0	Модуль k8s, обеспечивающий поддержку и обработку запросов https

Таблица 24: Перечень контейнерных образов.

10 Список листингов, иллюстраций и таблиц

Список листингов

Список иллюстраций

Список таблиц