Соответствие Deckhouse рекомендациям PCI Security Standards Council

FLANT
Deckhouse
Kubernetes Platform

	Угроза	Лучшая практика	Подход, применяемый в Deckhouse
1	Аутентификация		
1.1	Инструмент оркестрации контейнеров предоставляет неавторизованный доступ к API, делая возможным несанкционированное вмешательство в рабочие нагрузки.	**a.** Доступ пользователей или других сервисов к компонентам инструментария для оркестрации и вспомогательным сервисам — например, мониторингу — должен требовать аутентификации и индивидуальной отчетности.	В соответствии с рекомендуемой лучшей практикой.
1.2	Для управления инструментами оркестрации контейнеров используются стандартные учетные записи администраторов, что затрудняет отслеживание действий конкретных лиц, имеющих доступ к учетной записи администратора.	**a.** Все пользовательские учетные записи, используемые для аутентификации в инструментах оркестрации, должны быть привязаны к конкретным лицам. Не следует использовать стандартные учетные записи. Если учетная запись по умолчанию присутствует и не может быть удалена, измените ее пароль на надежный и уникальный с последующим отключением этой учетной записи. Злоумышленник не сможет включить ее снова или воспользоваться паролем, установленным по умолчанию.	Стандартные учетные записи не используются. Аутентификация раннеров осуществляется с помощью учетных записей сервисов, пользователей — с помощью механизма OICD.
1.3	Некоторые учетные данные, например, клиентские сертификаты, не предусматривают возможность отзыва. Их утеря влечет риск несанкционированного доступа к API-интерфейсам кластера.	**a.** Все учетные данные, используемые системой оркестрации, должны предусматривать возможность отзыва.	Сертификаты используются только для аутентификации компонентов управляющего слоя в кластере и никогда не покидают пределы соответствующих узлов.
1.4	Учетные данные, используемые для доступа к административным учетным записям для контейнеров или инструментов оркестрации контейнеров, хранятся небезопасно, что может привести к несанкционированному доступу к контейнерам или конфиденциальным данным.	**a.** Механизмы аутентификации, используемые системой оркестрации, должны хранить учетные данные в хранилище с соответствующей защитой.	В Deckhouse учетные данные для аутентификации хранятся в секретах или custom resource'ax Kubernetes. По умолчанию эти ресурсы не зашифрованы, однако пользователь может сделать это самостоятельно — все необходимые инструменты встроены в платформу.

			В настоящее время ведется работа над специализированным модулем для хранения секретов. https://github.com/deckhouse/deckhouse/issues/3053
1.5	Автоматический доступ рабочих нагрузок в кластере к учетным данным. Подобными данными легко злоупотреблять, особенно если соответствующие им роли наделены избыточными правами.	**a.** Учетные данные для доступа к системе оркестрации должны предоставляться работающим в кластере сервисам только в тех случаях, когда это явно необходимо.	По умолчанию предоставляются нулевые права. Каждое приложение или сервис получает минимальный набор прав в соответствии с RBAC-ролями, определенными в кластере. Все предоставленные права в кластере хранятся в репозитории Deckhouse, что исключает саму возможность инъекции прав. В кластере имеется 6 предустановленных ролей с четкими правилами, которые рекомендуется использовать.
		b. У учетных записей сервисов должны быть минимальные возможные права. Конкретный уровень прав в каждом случае определяется RBAC-настройками кластера.	
1.6	Статические учетные данные, т.е. пароли, используемые администраторами или учетными записями сервисов, подвержены атакам типа credential stuffing, phishing, keystroke logging, local discovery, extortion, password spray и brute force.	**a.** Интерактивные пользователи, обращающиеся к API оркестратора контейнеров, должны использовать многофакторную аутентификацию (MFA).	Многофакторная аутентификация может быть подключена к протоколу OIDC кластера, однако для этого необходимо установить соответствующие инструменты. Статические пароли не используются.
2	**Авторизация**		
2.1	Избыточные права доступа к API оркестратора контейнеров позволяют пользователям несанкционированно изменять рабочие нагрузки.	**a.** Доступ к системам оркестрации для пользователей или сервисов должен предоставляться по принципу наименьших возможных привилегий. Не следует использовать общий административный доступ.	В соответствии с рекомендуемой лучшей практикой.
2.2	Избыточные права доступа к инструментам оркестрации контейнеров могут быть получены с помощью жестко прописанных (hardcoded) групп доступа.	**a.** Весь доступ, предоставленный к инструменту оркестрации, должен допускать модификацию.	В соответствии с рекомендуемой лучшей практикой.
		b. Группы доступа не должны быть жестко прописаны (hardcoded).	В соответствии с рекомендуемой лучшей практикой.
2.3	Группы доступа не должны быть жестко прописаны (hardcoded).	**a.** Используйте ручные и автоматизированные средства для регулярного аудита установленных прав.	В платформе реализован цикл сверки. Все созданные объекты в кластере регулярно проверяются

			на соответствие желаемому состоянию.
3	**Безопасность рабочих нагрузок**		
3.1	Доступ к общим ресурсам на базовом хосте допускает выход за пределы контейнера, что ставит под угрозу безопасность совместно используемых ресурсов.	**a.** Рабочие нагрузки под управлением оркестратора по умолчанию должны быть настроены так, чтобы доступ к узлам кластера был невозможен. В случаях, когда доступ к ресурсам на узлах разрешен, привилегии должны быть настроены по принципу необходимого минимума. Использования привилегированных контейнеров следует избегать.	В Deckhouse в рамках стандартов безопасности Pod'ов (Pod Security Standards) определены три политики, покрывающие весь спектр вопросов безопасности. Эти политики носят кумулятивный характер и варьируются от крайне разрешительных до крайне ограничительных.
3.2	Использование неспецифических версий образов контейнеров может способствовать атакам типа supply chain, когда вредоносная версия образа загружается в реестр злоумышленником.	**a.** В определениях/манифестах рабочих нагрузок должны быть прописаны конкретные и хорошо изученные версии образов контейнеров. Контроль должен осуществляться с помощью надежного механизма, проверяющего криптографические подписи образов. Если подписи недоступны, следует использовать дайджесты сообщений.	Развертывание образов с использованием дайджестов SHA в качестве идентификаторов планируется в ближайшем будущем. https://github.com/deckhouse/deckhouse/issues/1825
3.3	Контейнеры, полученные из ненадежных источников, могут содержать вредоносное ПО или уязвимости.	**a.** Образы всех контейнеров, работающих в кластере, должны поступать из надежных источников.	Все образы контейнеров хранятся в приватном репозитории. Все компоненты с открытым исходным кодом собираются специально для платформы.
4	**Сетевая безопасность**		
4.1	Контейнерные технологии с контейнерными сетями, не поддерживающие сегментацию или ограничение сети, допускают несанкционированную сетевую коммуникацию между контейнерами.	**a.** Сети, имеющие отношения к инструментам для оркестрации контейнеров, должны быть настроены по принципу запрета по умолчанию; доступ должен предоставляться только в тех случаях, когда это необходимо для работы приложений.	Рекомендуемый подход реализуется на основе сетевых политик cilium. В ближайшем будущем в платформе будет доступна мультитенантность "из коробки". https://github.com/deckhouse/deckhouse/issues/1419
4.2	Доступ из контейнера или других сетей к компоненту оркестрации и административным API-интерфейсам позволяет проводить атаки с повышением привилегий.	**a.** Доступ к компонентам системы оркестрации и административным API должен быть ограничен путем формирования списка разрешенных IP-адресов.	Рекомендуемый подход реализуется на основе сетевых политик cilium. В ближайшем будущем в платформе будет доступна мультитенантность "из коробки". https://github.com/deckhouse/deckhouse/issues/1419
4.3	По умолчанию трафик, поступающий на управляющие интерфейсы API, не шифруется, что позволяет проводить атаки с подменой пакетов или сниффингом.	**a.** Весь трафик, поступающий на API компонентов системы оркестрации, должен проходить по зашифрованным соединениям, при этом ротация ключей шифрования	Все компоненты платформы взаимодействуют по протоколу TLS и авторизуются с помощью сертификатов.

		должна соответствовать требованиям PCI к ключам и секретам.	
5	**Инфраструктура открытых ключей (PKI)**		
5.1	Неспособность некоторых инструментов оркестрации контейнеров осуществлять отзыв сертификатов может привести к злоупотреблению украденным или потерянным сертификатом.	**a.** Аутентификация на основе сертификатов не должна использоваться в случаях, когда отзыв сертификатов не поддерживается. **b.** Ротация сертификатов должна осуществляться в соответствии с требованиями PCI (Payment Card Industry Data Security Standard) или политикой клиента, а также проводиться в случае компрометации контейнеров.	Сертификаты используются только для аутентификации компонентов управляющего слоя в кластере и никогда не покидают пределы узлов управляющего слоя.
5.2	PKI и сервисы центров сертификации, интегрированные в инструменты оркестрации контейнеров, могут не обеспечивать достаточную безопасность за пределами окружения инструмента оркестрации контейнеров, что может привести к взлому других сервисов, использующих данную цепочку доверия.	**a.** Сертификаты, выданные инструментами оркестрации, не должны приниматься за пределами окружения оркестратора, поскольку его приватный ключ, используемый для взаимодействия с центром сертификации, может быть защищен слабее, чем другие цепочки доверия корпоративной PKI.	В соответствии с рекомендуемой лучшей практикой.
6	**Управление секретами**		
6.1	Секреты (в том числе учетные данные), хранящиеся ненадлежащим образом и предоставляемые через инструмент оркестрации контейнеров, могут попасть к неавторизованным пользователям или злоумышленникам с некоторым доступом к окружению.	**a.** Все секреты, необходимые для приложений, работающих под управлением платформы оркестрации, должны храниться в зашифрованном виде в специализированных системах управления секретами.	В соответствии с рекомендуемой лучшей практикой.
6.2	Хранение секретов вне системы контроля версий может привести к простою, если активные секреты окажутся скомпрометированы и возникнет необходимость в их быстрой ротации.	**a.** Используйте систему контроля версия для управления секретами. Это позволит быстро их обновить или отозвать в случае компрометации.	В соответствии с рекомендуемой лучшей практикой.
7	**Аудит инструментов для оркестрации контейнеров**		
7.1	Существующих решений по управлению ресурсами и логированию может оказаться недостаточно из-за эфемерной природы контейнеров и интеграции	**a.** Доступ к API-интерфейсу(-ам) системы оркестрации должен подвергаться аудиту и отслеживаться на предмет признаков несанкционированного доступа. Логи аудита	Платформа собирает логи аудита и позволяет хранить их во внешнем хранилище с помощью модуля log-shipper.

	инструментов оркестрации контейнеров.	должны надежно храниться в централизованной системе.	
8	**Мониторинг контейнеров**		
8.1	Локальные решения для сбора логов затрудняют обработку событий, связанных с безопасностью, и поиск соответствующих корреляций, поскольку контейнеры регулярно уничтожаются.	**a.** Должен быть реализован централизованный сбор информации об активности контейнеров. Это позволит коррелировать события (выявлять закономерности) в различных экземплярах одного и того же контейнера.	Модуль log-shipper обеспечивает централизованный сбор логов контейнеров. Хранилище логов Loki станет частью платформы в ближайшем будущем. https://github.com/deckhouse/deckhouse/issues/2532
8.2	Без соответствующих средств обнаружения эфемерная природа контейнеров позволяет злоумышленникам проводить атаки, оставаясь незамеченными.	**a.** Необходимо внедрить средства контроля для обнаружения попыток добавить исполняемые файлы в работающие контейнеры и запустить их, а также неавторизованного изменения файлов контейнеров.	Информация обо всех действиях в кластере сохраняется в логи модулем log-shipper. Falco будет внедрен в качестве инструмента анализа в ближайшем будущем. https://github.com/deckhouse/deckhouse/issues/2423
9	**Безопасность среды исполнения для контейнеров**		
9.1	Стандартная политика безопасности Linux для контейнеров, реализуемых в виде процессов, потенциально обеспечивает широкое пространство для атаки, поскольку все процессы используют одно общее ядро. Без дополнительных мероприятий по повышению безопасности система может быть восприимчива к эксплойтам, позволяющим злоумышленнику выходить за пределы контейнера.	**a.** Для рабочих нагрузок с высоким уровнем риска следует либо предусмотреть возможность использования сред исполнения для контейнеров с изоляцией на уровне гипервизора, либо специальных "песочниц".	В соответствии с рекомендуемой лучшей практикой. Интегрированный в платформу Open Policy Agent проверяет Pod'ы на наличие избыточных прав и привилегированного статуса.
9.2	Механизм реализации контейнеров в виде процессов в ОС Windows не обеспечивает необходимой безопасности (см. рекомендации Microsoft), допуская выход за пределы контейнера.	**a.** При использовании Windows-контейнеров для запуска приложений необходимо развертывать изоляцию типа Hyper-V в соответствии с рекомендациями Microsoft по безопасности.	Windows-контейнеры не поддерживаются.
10	**Патчи**		
10.1	Устаревшие компоненты средств оркестрации контейнеров могут быть уязвимы для эксплойтов, способных скомпрометировать установленный кластер	**a.** Все инструменты для оркестрации контейнеров должны поддерживаться и регулярно получать исправления для защиты от угроз в области безопасности.	В Deckhouse регулярно проверяются списки CVE на наличие новых уязвимостей, своевременно пересобираются базовые образы для всех

	или рабочие нагрузки.	Поставщиком таких исправлений выступает либо основной проект, либо поставщик системы оркестрации.	компонентов, а Kubernetes обновляется до последних минорных версий.
10.2	Уязвимости, специфичные для хостов, на которых работают инструменты для оркестрации контейнеров (обычно это виртуальные машины Linux), делают возможной компрометацию как самих инструментов, так и других компонентов.	**a.** Операционная хост-система на всех узлах, входящих в кластер, управляемый инструментом для оркестрации контейнеров, должна поддерживаться в актуальном состоянии, а патчи накладываться своевременно. Динамическое перепланирование рабочих нагрузок позволяет обслуживать узлы по очереди, без необходимости выделять специальное окно.	Платформа не отвечает за обновления безопасности на уровне операционной системы.
10.3	Поскольку инструменты оркестрации контейнеров обычно запускаются в кластерах как контейнеры, любой контейнер с уязвимостями способен их скомпрометировать.	**a.** Все образы контейнеров для приложений, работающих в кластере, следует регулярно проверять на наличие уязвимостей, своевременно накладывать патчи, а исправленные образы развертывать в кластере.	Регулярная проверка уязвимостей проводится компонентом Trivy. В скором времени ее планируется автоматизировать: https://github.com/deckhouse/deckhouse/issues/2679
11	**Управление ресурсами**		
11.1	Скомпрометированный контейнер может нарушить работу приложений из-за чрезмерного потребления совместно используемых ресурсов.	**a.** У всех рабочих нагрузок, запускаемых с помощью системы оркестрации контейнеров, должны быть установлены лимиты на ресурсы. Это снизит риск "шумных соседей" и, соответственно, проблем с доступностью рабочих нагрузок в кластере.	Open Policy Agent — инструмент для принудительного применения политик — следит за запросами и лимитами на ресурсы у всех запускаемых контейнеров. В скором времени будет реализована мультитенантность, что позволит задавать ресурсы по умолчанию. https://github.com/deckhouse/deckhouse/issues/1419
12	**Сборка образов контейнеров**		
12.1	Базовые образы контейнеров, загружаемые из ненадежных источников или содержащие лишние пакеты, повышают риск атак на цепочки поставок.	**a.** Образы контейнеров должны собираться из надежных и актуальных минималистичных базовых образов.	Разработчики Deckhouse самостоятельно собирают все образы, имеющие отношение к платформе.
12.2	Базовые образы, загруженные из сторонних реестров, могут содержать вредоносные программы, бэкдоры и уязвимости.	**a.** Набор стандартных базовых образов контейнеров должен храниться в реестре, находящемся под контролем организации.	Все образы контейнеров, имеющие отношение к платформе, хранятся в приватном репозитории.
12.3	В Linux контейнеры по умолчанию запускаются от имени root-пользователя, что повышает риск выхода	**a.** Образы контейнеров должны собираться с учетом запуска от имени обычного (не-root) пользователя.	Запуск от имени обычного пользователя проверяется линтером на уровне платформы.

	за их пределы.		
13	**Реестр**		
13.1	Неавторизованная модификация образов контейнеров организации позволяет злоумышленнику внедрить вредоносное программное обеспечение в production-окружение.	**a.** Доступ к реестрам контейнеров, управляемым организацией, должен находиться под строгим контролем.	В Deckhouse доступ к реестрам образов осуществляется с помощью лицензионных токенов.
		b. Только уполномоченные лица должны быть наделены правами на изменение или замену образов.	В соответствии с рекомендуемой лучшей практикой.
13.2	Отсутствие разделения между production и non-production-реестрами контейнеров может привести к развертыванию небезопасных образов в production-окружение.	**a.** Рассмотрите возможность использования двух реестров: одного — для критически важных рабочих нагрузок и нагрузок production-уровня, другого — для разработки и тестирования. Это снизит путаницу и сделает невозможным случайное попадание необслуживаемого или уязвимого образа в production-кластер.	В Deckhouse образы под каждый тип окружения (production и development) собираются и хранятся раздельно.
13.3	Уязвимости могут присутствовать в базовых образах независимо от их источника из-за ошибок в конфигурации и других причин.	**a.** При возможности реестры должны регулярно сканировать образы и предотвращать развертывание уязвимых образов в средах исполнения для контейнеров.	Регулярная проверка образов проводится компонентом Trivy. В скором времени ее планируется автоматизировать. https://github.com/deckhouse/deckhouse/issues/2679
13.4	Образы, которые считаются надежными, могут быть злонамеренно или непреднамеренно подменены или изменены и развернуты в среде исполнения для контейнеров.	**a.** Реестры должны быть настроены на интеграцию с процессами сборки образов таким образом, чтобы только подписанные образы от авторизованных сборочных пайплайнов были доступны для развертывания в средах исполнения для контейнеров.	Список всех образов, имеющих отношение к платформе, хранится в контейнере Deckhouse, манифест которого берется из доверенного источника. В качестве тегов используются хэши, основанные на содержимом. Развертывание образов с использование дайджестов SHA в качестве идентификаторов планируется реализовать в ближайшем будущем. https://github.com/deckhouse/deckhouse/issues/1825
14	**Управление версиями**		
14.1	Без надлежащего контроля и версионирования конфигурационных файлов в системе оркестрации контейнеров злоумышленник может внести неавторизированные изменения в настройки окружения.	**a.** Для управления всеми несекретными конфигурационными файлами следует использовать систему контроля версий.	В соответствии с рекомендуемой лучшей практикой.

		b. Связанные объекты должны группироваться в один файл.	В соответствии с рекомендуемой лучшей практикой.
		c. Для семантической идентификации объектов следует использовать лейблы.	Все компоненты, развернутые в платформе, обязательно помечаются соответствующими лейблами.
15	**Управление конфигурациями**		
15.1	Инструменты для оркестрации контейнеров могут быть неправильно настроены, создавая уязвимости в области безопасности.	**a.** Перед развертыванием все конфигурации и образы контейнеров должны тестироваться в окружении, идентичном production-окружению.	В соответствии с рекомендуемой лучшей практикой.
		b. Для всех компонентов системы, включая средства оркестрации контейнеров, должны быть разработаны стандарты конфигурации, учитывающие все известные уязвимости и соответствующие принятым в отрасли стандартам и руководствам по безопасности от вендоров. **i.** Устраните все известные уязвимости. **ii.** Придерживайтесь принятых в отрасли стандартов по защите систем или соответствующих рекомендаций вендора в области безопасности. **iii.** Обновляйте ПО по мере выявления новых уязвимостей.	Регулярная проверка контейнеров и исполняемых файлов проводится с помощью Trivy. В скором времени процесс будет автоматизирован. https://github.com/deckhouse/deckhouse/issues/2679 Кроме того, код сканируется линтерами.
16	**Сегментация**		
16.1	В системах оркестрации, изначально не предназначенных для безопасного использования в multitenancy-режиме, смешение политик безопасности в рамках общего окружения позволяет злоумышленникам проникать из окружения с низкими требованиями к безопасности в окружение с высокими требованиями к безопасности.	**a.** Там, где это целесообразно, компоненты с повышенными требованиями к безопасности следует размещать в выделенных кластерах. Если это невозможно, следует позаботиться о полном разделении рабочих нагрузок с разными требованиями к безопасности.	Deckhouse позволяет вручную вынести компоненты с повышенными требованиями к безопасности за пределы кластера. В будущем эта опция будет встроена в платформу. Также будет реализована поддержка мультитенантности. https://github.com/deckhouse/deckhouse/issues/1419
16.2	Размещение критически важных систем на тех же узлах, на которых находятся обычные контейнеры приложений, позволяет подорвать безопасность кластера за счет	**a.** Критически важные системы должны работать на выделенных узлах.	В Deckhouse отдельные узлы выделяются для компонентов управляющего слоя, системных нужд (например, мониторинга), ingress-ресурсов

	использования общих ресурсов на узле.		и рабочих нагрузок.
16.3	Размещение рабочих нагрузок с различными требованиями к уровню безопасности на одних и тех же узлах кластера позволяет злоумышленникам получить неавторизованный доступ к окружениям с высокими требованиями к уровню безопасности путем взлома базового узла.	**a.** Необходимо обеспечить разделение пулов узлов кластера таким образом, чтобы пользователь, работающий с приложениями с низкими требованиями к уровню безопасности, не мог планировать рабочие нагрузки на узлы с высокими требованиями к уровню безопасности.	Рабочие нагрузки могут быть запущены только на выделенных рабочих узлах. Taint'ы и селекторы узлов позволяют объединять узлы в различные группы в соответствии с требованиями безопасности.
16.4	Модификация общих ресурсов кластера пользователями, имеющими доступ к отдельным приложениям, может привести к несанкционированному доступу к критически важным общим ресурсам.	**a.** Рабочие нагрузки и пользователи, управляющие отдельными приложениями, не должны иметь прав на изменение общих ресурсов кластера или ресурсов, используемых другими приложениями.	В платформу изначально встроены роли с необходимым разделением прав, например, 'Пользователь' или 'Администратор кластера'.