

Программное обеспечение «Servicepipe Cybert»

Описание процессов жизненного цикла программного обеспечения

Москва - 2026 г.

Оглавление

1 Общие сведения	3
1.1 Назначение документа	3
1.2 Сведения о программном обеспечении	3
1.3 Правообладатель и область применения	3
2 Процессы жизненного цикла ПО	4
2.1 Модель жизненного цикла	4
2.2 Анализ требований	4
2.3 Планирование работ и версий	5
2.4 Проектирование	5
2.5 Разработка	5
2.6 Управление зависимостями	6
2.7 Сборка и подготовка объектного кода	6
2.8 Тестирование	6
2.9 Подготовка релиза	7
2.10 Внедрение	7
3 Процессы сопровождения и обеспечения жизненного цикла	7
3.1 Поддержка версий и развитие функциональности	8
3.2 Устранение сбойных ситуаций и дефектов	8
3.3 Эксплуатационная и консультационная поддержка	8
3.4 Документирование	9
3.5 Обучение и поддержание квалификации персонала	9
3.6 Обеспечение безопасности и устойчивости	9
4 Управление конфигурацией, исходным кодом и релизными артефактами	10
4.1 Хранение исходного кода	10
4.2 Хранение объектного кода и поставочных артефактов	10
4.3 Контроль изменений и трассируемость	11
5 Персонал, обеспечивающий жизненный цикл ПО	11
6 Инфраструктура разработки, сборки, тестирования и поддержки	12
6.1 Инфраструктура хранения и разработки	12
6.2 Инфраструктура сборки и тестирования	13
6.3 Инфраструктура технической поддержки	13

1 Общие сведения

1.1 Назначение документа

Настоящий документ содержит описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения «Servicepipe Cybert» (далее - Servicepipe Cybert, Cybert, программное обеспечение, ПО). В документе рассматриваются процессы анализа требований, проектирования, разработки, тестирования, выпуска, внедрения, сопровождения, документирования, управления версиями, управления исходным и объектным кодом, а также процессы обеспечения квалификации персонала, участвующего в создании и сопровождении программного обеспечения.

Документ подготовлен для использования в составе комплекта материалов, предоставляемых при подтверждении характеристик программного обеспечения и описании организационно-технических процессов его разработки, выпуска и сопровождения.

1.2 Сведения о программном обеспечении

ПО «Servicepipe Cybert» предназначено для защиты веб-ресурсов и приложений от вредоносного автоматизированного трафика и прикладных DDoS-атак. Программное обеспечение анализирует сетевые, транспортные, TLS- и HTTP-признаки входящих запросов, поведение клиента в пределах сессии и результаты дополнительных клиентских проверок. На основании совокупности признаков определяется класс источника и выбирается итоговое действие: пропуск, блокирование, ограничение интенсивности или дополнительная проверка.

В зависимости от архитектуры проекта Cybert может использоваться по схеме «Прокси», в локальной схеме «Модуль», в гибридной схеме либо в он-прем варианте поставки. В локальных схемах программное обеспечение взаимодействует с NGINX/Angie через специализированный модуль Cybert, который передает контекст запроса в центр принятия решений Cresp и применяет полученный вердикт.

Управление пользовательскими функциями выполняется через раздел «Защита приложений» панели управления. Пользователю доступны управление защищаемыми ресурсами и их настройками, доменами и сертификатами, источниками, белыми и черными списками, геоограничениями, пользовательскими правилами, лимитами запросов, правилами сбора JS score и классификации ботов, а также статистика, аналитика, журналы запросов, обнаруженные аномалии и пользовательские события.

Серверное ядро Cybert реализовано преимущественно на языке C++ и предназначено для исполнения в GNU/Linux на серверной архитектуре x86-64. В состав программного комплекса входят собственные серверные компоненты Servicepipe, модуль интеграции с NGINX/Angie, сервисы управления конфигурацией, backend и frontend панели управления, а также сторонние инфраструктурные компоненты, необходимые для хранения данных, журналирования, мониторинга и эксплуатации.

1.3 Правообладатель и область применения

Правообладателем программного обеспечения является Общество с ограниченной ответственностью «Сервиспайп» (ООО «Сервиспайп»).

Параметр	Значение
Полное наименование	Общество с ограниченной ответственностью «Сервиспайп»
Сокращенное наименование	ООО «Сервиспайп»
ИНН / КПП / ОГРН	7708257951 / 772201001 / 1157746483784
Юридический адрес	Россия, 111024, г. Москва, ул. Кабельная 2-я, д. 2, стр. 3, эт. 2, помещение XVIII, ком. 8
Телефон	+7 (495) 374-62-65
Электронная почта	welcome@servicepipe.ru
Веб-сайт	servicepipe.ru

Процессы жизненного цикла, описанные в настоящем документе, применяются к собственным компонентам Servicepipe Cybert, включая серверное ядро Cresp, Cresp-BH, модуль Cybert для NGINX/Angie, Cresp API, сервисы формирования и доставки конфигурации, backend и frontend панели управления, API-адаптеры, средства установки и обновления, конфигурационные шаблоны, тесты и эксплуатационную документацию. Сторонние open source и инфраструктурные компоненты рассматриваются как зависимости проекта и учитываются в рамках процессов управления зависимостями, сборки, тестирования и поставки.

2 Процессы жизненного цикла ПО

2.1 Модель жизненного цикла

Разработка и сопровождение ПО «Servicepipe Cybert» осуществляются в рамках итерационно-релизной модели жизненного цикла с обязательным прохождением контрольных стадий для каждой версии программного обеспечения. Такая модель позволяет планомерно развивать механизмы защиты, выпускать корректирующие обновления и сохранять управляемость изменений, влияющих на классификацию трафика, принятие решений, доставку конфигурации, наблюдаемость и пользовательский интерфейс.

Для каждой версии программного обеспечения выполняются следующие основные стадии: анализ требований, планирование работ, проектирование, разработка, управление зависимостями, сборка, тестирование, подготовка релиза, внедрение и сопровождение. Переход к последующей стадии выполняется после фиксации результата предыдущей стадии и устранения критичных несоответствий, препятствующих выпуску версии.

Жизненный цикл охватывает как функциональные изменения механизмов AntiBot/L7 DDoS, профилей и пользовательских правил, так и изменения интеграции с NGINX/Angie, Cresp API, контура формирования и доставки runtime-конфигурации, панели управления, аналитики, журналирования, инфраструктурных компонентов и эксплуатационной документации.

2.2 Анализ требований

На стадии анализа выполняется выявление, уточнение и формализация требований к программному обеспечению. Источниками требований являются продуктовая стратегия, обращения и потребности заказчиков, результаты эксплуатации текущих версий, данные о характере вредоносного автоматизированного трафика, требования к безопасности и отказоустойчивости, требования к совместимости с инфраструктурой заказчиков, а также внутренние инициативы по развитию алгоритмов защиты и повышению наблюдаемости.

В рамках анализа определяются функциональные требования к классификации запросов, профилям защиты, пользовательским правилам, rate limiting, DDoS-контексту, JavaScript-, Cookie- и CAPTCHA-проверкам, белым и черным спискам, геоограничениям, журналам запросов, статистике, аналитике и пользовательским событиям. Для панели управления анализируются сценарии создания и настройки ресурсов, формы, валидация, локализация и права доступа.

Отдельно анализируются нефункциональные требования: производительность и задержка принятия решения, горизонтальное масштабирование filter nodes, отказоустойчивость management-контура, объемы хранения журналов и временных рядов, устойчивость доставки конфигурации, возможность on-prem и offline-развертывания, безопасность межкомпонентного взаимодействия и совместимость с NGINX/Angie и инфраструктурой заказчика.

Результатом стадии анализа является согласованный перечень требований к версии программного обеспечения. Требования могут оформляться в виде технического задания, продуктового описания, задач в системе управления работами, спецификаций API, пользовательских сценариев, прототипов интерфейса, описаний алгоритмов или иных формализованных материалов, достаточных для последующего проектирования и реализации.

2.3 Планирование работ и версий

На стадии планирования определяется состав работ, входящих в очередную версию или корректирующее обновление. Для каждой работы устанавливаются ожидаемый результат, ответственные исполнители, зависимости от смежных компонентов, требования к тестированию и критерии готовности.

Планирование выполняется с учетом приоритетов развития продукта, влияния изменений на защищаемый трафик, необходимости устранения дефектов, сроков поставки, совместимости с существующими профилями и конфигурациями и готовности документации. Изменения, затрагивающие алгоритмы принятия решений Cresp, формат runtime-конфигурации, Cresp API, модуль NGINX/Angie или схемы хранения данных, рассматриваются как изменения повышенной значимости и требуют дополнительной архитектурной и регрессионной проверки.

Результатом стадии планирования является утвержденный набор задач версии, предварительный перечень релизных артефактов, план тестирования и перечень документации, которую необходимо создать или актуализировать.

2.4 Проектирование

На стадии проектирования разрабатываются архитектурные и технические решения, обеспечивающие реализацию согласованных требований. Проектирование охватывает серверное ядро Cresp, механизмы профилей и правил, поведенческий анализ, счетчики и лимиты, взаимодействие с модулем NGINX/Angie, Cresp-BH, Cresp API, backend и frontend панели управления, схемы данных, форматы конфигурации, процессы доставки конфигурации, клиентские проверки и контур наблюдаемости.

В рамках проектирования определяются состав и границы компонентов, модели данных, контракты GraphQL/REST и внутренних RPC-интерфейсов, структура форм пользовательского интерфейса, порядок взаимодействия компонентов, требования к хранению и преобразованию конфигурации, форматы журналов и метрик, а также сценарии безопасного внедрения и отката изменений.

Отдельно проектируется конфигурационный контур Cybert: изменения профильных настроек поступают через Cresp API, необходимые данные сохраняются в MariaDB, на их основании формируется итоговая runtime-конфигурация, после чего Cresp config deliver доставляет ее на filter nodes. На узлах фильтрации конфигурация загружается компонентом Cresp, взаимодействующим с NGINX/Angie.

Результатом стадии проектирования является комплект проектных решений, содержащий достаточные сведения для разработки: описание архитектуры, схемы взаимодействия, спецификации API, макеты интерфейса, описание изменений конфигурации и данных, требования к миграциям и критерии проверки.

2.5 Разработка

На стадии разработки осуществляется программная реализация функциональности в соответствии с архитектурными и техническими решениями. Разработка ведется с использованием системы контроля версий Git, размещенной на собственной инфраструктуре ООО «Сервиспайп». Все изменения исходного кода фиксируются в репозиториях и сопровождаются историей изменений.

Разработка охватывает следующие группы компонентов: серверное ядро Cybert на C++, Cresp и Cresp-BH, модуль интеграции с NGINX/Angie, Cresp API и сервисы конфигурационного контура, backend панели управления и API, frontend панели управления, средства установки и обновления, Ansible playbook-и и bootstrap-скрипты, конфигурационные шаблоны, автоматизированные тесты и документацию.

В ходе разработки выполняются реализация новых механизмов защиты и аналитики, доработка пользовательских правил и классификации, оптимизация производительности, исправление выявленных дефектов, актуализация тестов, подготовка миграций данных и изменение поставочных материалов. Изменения проходят внутренний контроль: проверку соответствия требованиям, анализ влияния на смежные компоненты, код-ревью и подготовку к тестированию.

Результатом стадии разработки является набор изменений, интегрированный в соответствующие ветки репозитория и подготовленный к сборке тестовой или релизной версии программного обеспечения.

2.6 Управление зависимостями

ПО «Servicepipe Cybert» использует собственные компоненты ООО «Сервиспайп» и набор сторонних компонентов под свободными или коммерчески допустимыми лицензиями. Сторонние компоненты рассматриваются как зависимости проекта; исключительные права на них как на самостоятельные проекты не заявляются.

В рамках управления зависимостями выполняются фиксация используемых версий, учет лицензий, хранение сведений о составе поставки, проверка совместимости, оценка влияния обновлений и обеспечение воспроизводимости сборки. Для JavaScript/TypeScript-зависимостей состав фиксируется lockfile; для C/C++ и инфраструктурных компонентов фиксируются версии, используемые в сборочном окружении и конкретном release manifest.

К ключевым сторонним зависимостям и инфраструктурным компонентам Cybert относятся NGINX/Angie, GNU/Linux, Boost, gRPC, Apache Thrift, PostgreSQL, MariaDB, ClickHouse, VictoriaMetrics, Kafka, Vector, Docker/OCI runtime, keepalived и, в конфигурациях, где это требуется сетевой архитектурой, BIRD и связанные механизмы маршрутизации. Для frontend и backend панели используются Node.js, TypeScript/Javascript, React, GraphQL и сопутствующие библиотеки.

Результатом процесса управления зависимостями является контролируемый перечень собственных и сторонних компонентов, позволяющий воспроизвести сборку, проверить лицензионный состав и обеспечить сопровождаемость поставки.

2.7 Сборка и подготовка объектного кода

На стадии сборки выполняется преобразование исходного текста программного обеспечения в объектный код и поставочные артефакты. Сборка выполняется на контролируемой инфраструктуре ООО «Сервиспайп» с использованием зафиксированных версий исходного кода, зависимостей, конфигураций сборки и manifest-а версии.

В результате сборки формируются исполняемые файлы и библиотеки C++-компонентов, модуль Cybert для NGINX/Angie, пакеты серверных сервисов, backend и frontend панели управления, Docker/OCI-образы для применяемых компонентов, архивы, конфигурационные шаблоны, installer, Ansible playbook-и, bootstrap-скрипты, материалы offline-поставки и сопроводительные файлы.

В состав поставки включаются собственные компоненты Servicepipe Cybert и необходимые зависимости, прошедшие проверку и размещенные в контролируемых хранилищах артефактов. Результатом стадии является сформированный комплект объектного кода и релизных материалов, пригодный для установки на тестовом стенде, приемочном стенде или в инфраструктуре заказчика.

2.8 Тестирование

На стадии тестирования проводится проверка реализованной версии программного обеспечения на соответствие функциональным и нефункциональным требованиям. Тестирование выполняется на тестовых стендах, воспроизводящих типовые схемы интеграции Cybert и сценарии прохождения HTTP(S)-трафика.

В рамках тестирования выполняются функциональные, интеграционные, регрессионные, API-, UI-, инсталляционные и, при необходимости, нагрузочные проверки. Проверяются классификация запросов, применение профилей и пользовательских правил, лимиты запросов, белые и черные списки, геоограничения, режимы DDoS, действия pass/block, JavaScript-, Cookie- и CAPTCHA-проверки, работа JS score, сбор статистики, журналирование, аналитика и пользовательские события.

Отдельно проверяется межкомпонентное взаимодействие: передача контекста от модуля Cybert для NGINX/Angie в Cresp, получение и применение вердикта, взаимодействие с Cresp-БН, работа Cresp API, сохранение конфигурационных данных в MariaDB, формирование итоговой runtime-конфигурации и ее доставка через Cresp config deliver на filter nodes. Проверяется, что Cresp на узлах фильтрации получает актуальную конфигурацию и применяет ее к новым запросам.

Для frontend и backend-компонентов проверяются формы создания и изменения ресурсов, доменов, источников, списков и пользовательских правил, валидация пользовательских данных, корректность GraphQL/REST-взаимодействия, права доступа, локализация, обработка ошибок, журналов, экспорта и аналитических выборок. Для on-prem поставки дополнительно проверяются установка, обновление, резервирование management-узлов и работоспособность инфраструктурных компонентов.

При выявлении несоответствий версия возвращается на доработку. Исправления повторно проходят проверки, связанные с измененным функционалом, а также необходимые регрессионные проверки. Результатом стадии тестирования является версия программного обеспечения, соответствующая установленным требованиям и готовая к подготовке релиза.

2.9 Подготовка релиза

На стадии подготовки релиза выполняется финальная фиксация состава версии, маркировка исходного кода, формирование release manifest-a, подготовка пакетов и образов, размещение артефактов в поставочных репозиториях и актуализация документации.

В состав релизных материалов включаются объектный код серверных компонентов, модуль Cybert для NGINX/Angie, backend и frontend панели управления, средства установки и обновления, конфигурационные шаблоны, сведения о зависимостях, инструкции по установке и эксплуатации, пользовательская документация, описание функциональных характеристик, архитектуры и процессов жизненного цикла, а также иные материалы, необходимые для эксплуатации конкретной поставки.

Перед выпуском проверяется согласованность версий Cresp, модуля NGINX/Angie, Cresp API, формата runtime-конфигурации и панели управления. Результатом стадии является релизный комплект Servicerpipeline Cybert, содержащий материалы, необходимые для установки, обновления и сопровождения версии.

2.10 Внедрение

На стадии внедрения выполняется установка или обновление программного обеспечения в инфраструктуре заказчика, на приемочном стенде, в инфраструктуре Servicerpipeline для прокси-схемы либо на ином согласованном контуре. Внедрение может выполняться силами заказчика по эксплуатационной документации либо при участии специалистов ООО «Сервиспайп».

Для он-прем варианта подготавливаются filter nodes и management nodes, устанавливаются NGINX/Angie с модулем Cybert, Cresp, Cresp-BH, Cresp API, сервисы доставки конфигурации, базы и компоненты наблюдаемости. Для схемы «Модуль» компонент Cybert интегрируется с фронтенд-сервером заказчика, а для прокси- и гибридной схем выполняются согласованные сетевые и сервисные настройки.

После установки выполняются первичная настройка защищаемых ресурсов, проверка получения конфигурации, контроль взаимодействия Cresp с NGINX/Angie, тестовые запросы с ожидаемыми действиями защиты, проверка записи журналов в ClickHouse и временных рядов в VictoriaMetrics, а также проверка доступа к панели управления.

Результатом стадии внедрения является установленная и настроенная версия программного обеспечения, готовая к эксплуатации. После внедрения сведения о версии, особенностях установки и выявленных замечаниях учитываются в процессах сопровождения и дальнейшего развития продукта.

3 Процессы сопровождения и обеспечения жизненного цикла

Процессы сопровождения направлены на обеспечение стабильного функционирования ПО «Servicerpipeline Cybert» в процессе эксплуатации, поддержание актуальности программного обеспечения, устранение выявленных дефектов, развитие функциональности и обеспечение пользователей необходимой документацией и консультационной поддержкой.

Сопровождение осуществляется организацией-разработчиком на протяжении всего срока использования программного обеспечения в рамках установленных договорных и внутренних регламентов.

3.1 Поддержка версий и развитие функциональности

Управление версиями Servicepipe Cybert осуществляется на основе принятой системы нумерации релизов и release manifest-ов. В зависимости от характера изменений выпускаются корректирующие обновления, минорные версии с расширением функциональности и мажорные версии, содержащие существенные архитектурные изменения.

В рамках поддержки версий выполняются:

- выпуск корректирующих обновлений для устранения дефектов и эксплуатационных проблем;
- выпуск минорных версий с расширением механизмов защиты, аналитики и возможностей панели управления;
- выпуск мажорных версий при значимых изменениях архитектуры, алгоритмов обработки или модели поставки;
- ведение журнала изменений и фиксация состава релиза;
- поддержание совместимости обновлений с существующими профилями и конфигурациями, если это предусмотрено характером изменений;
- актуализация пользовательской, эксплуатационной и технической документации.

Решения о выпуске новых версий принимаются с учетом результатов эксплуатации текущих версий, обращений пользователей, характера обнаруживаемых автоматизированных атак, необходимости устранения выявленных недостатков, развития функций раздела «Защита приложений», требований к безопасности и инфраструктурной совместимости.

3.2 Устранение сбойных ситуаций и дефектов

ООО «Сервиспайп» обеспечивает регистрацию и обработку обращений, связанных с некорректной работой программного обеспечения, ошибками установки, отклонениями в работе Cresp, модуля Cybert для NGINX/Angie, Cresp API, контура доставки конфигурации, панели управления, журналирования, статистики, аналитики или инфраструктурных компонентов поставки.

Процесс устранения сбойной ситуации включает:

- регистрацию обращения или внутреннего инцидента;
- первичную классификацию по влиянию на доступность защищаемого ресурса, корректность фильтрации, безопасность и пользовательскую функциональность;
- сбор диагностических данных, журналов, метрик, конфигурации и описания сценария;
- анализ и, при необходимости, воспроизведение ситуации на тестовом стенде;
- определение причины сбоя или дефекта;
- подготовку корректирующего решения, настройки обходного решения или рекомендаций;
- тестирование исправления и регрессионную проверку затронутых функций;
- выпуск корректирующего обновления либо включение исправления в плановый релиз;
- информирование заявителя о результате обработки обращения.

Если исправление требует изменения исходного кода, соответствующие изменения проходят стандартные стадии жизненного цикла: разработку, сборку, тестирование и подготовку релиза. Если причина связана с конфигурацией, сетевой интеграцией или эксплуатационной средой, пользователю предоставляются рекомендации по настройке или восстановлению работоспособности.

3.3 Эксплуатационная и консультационная поддержка

Эксплуатационная поддержка включает консультации по установке, обновлению, интеграции и использованию Servicepipe Cybert. Пользователям предоставляются рекомендации по подключению защищаемых ресурсов, настройке доменов и сертификатов, источников, списков доступа, геоограничений, пользовательских правил, rate limiting, классификации ботов и клиентских проверок.

Специалисты сопровождения помогают анализировать типовые проблемы эксплуатации: отсутствие или изменение трафика, неожиданное действие защиты, некорректное срабатывание пользовательского правила, проблемы с JavaScript/Cookie/CAPTCHA-проверками, отсутствие статистики или журналов, ошибки доставки runtime-конфигурации, недоступность Cresp или Cresp API, а также проблемы интеграции модуля с NGINX/Angie.

Для диагностики используются журналы в ClickHouse, метрики в VictoriaMetrics, сведения панели управления, сервисные журналы компонентов и сетевые средства диагностики. Обращения пользователей принимаются по установленным каналам связи службы технической поддержки ООО «Сервиспайп», включая договорные каналы взаимодействия и электронную почту организации.

3.4 Документирование

Документация на программное обеспечение формируется и поддерживается в актуальном состоянии на протяжении всего жизненного цикла. Документация отражает назначение продукта, функциональные характеристики, архитектуру, порядок установки, эксплуатации, настройки, диагностики и сопровождения программного обеспечения.

Процесс документирования включает:

- разработку пользовательской и эксплуатационной документации;
- подготовку и актуализацию описания функциональных характеристик и архитектуры;
- описание технических средств хранения исходного и объектного кода;
- описание процессов жизненного цикла программного обеспечения;
- актуализацию документации при выпуске новых версий;
- контроль соответствия документации реализованной функциональности и фактической архитектуре;
- хранение актуальных и архивных версий документации.

Изменения алгоритмов защиты, пользовательского интерфейса, API, схем интеграции, состава поставки, конфигурационного контура и эксплуатационных процедур сопровождаются внесением соответствующих изменений в документацию.

3.5 Обучение и поддержание квалификации персонала

Персонал, участвующий в разработке и сопровождении Servicepipe Cyber, обладает профильной квалификацией в области разработки программного обеспечения, высоконагруженных сетевых систем, эксплуатации Linux-инфраструктуры, информационной безопасности, тестирования, технической поддержки, анализа трафика и документирования.

ООО «Сервиспайп» обеспечивает поддержание квалификации персонала за счет внутреннего обучения, обмена опытом, участия специалистов в проектировании новых функций, анализа инцидентов и результатов эксплуатации, изучения новых технологий и обновления внутренних инструкций при внедрении новых архитектурных решений или инструментов.

Поддержание квалификации направлено на обеспечение надежности, сопровождаемости, безопасности и устойчивого развития программного обеспечения, а также на сохранение экспертизы по алгоритмам обнаружения автоматизированной активности и эксплуатации систем веб-защиты.

3.6 Обеспечение безопасности и устойчивости

В рамках обеспечения жизненного цикла выполняются мероприятия, направленные на снижение рисков, связанных с изменением алгоритмов фильтрации, обработкой конфигурации, поставкой релизных артефактов и эксплуатацией Cyber в инфраструктуре Servicepipe или заказчика.

К таким мероприятиям относятся ограничение доступа к репозиториям и хранилищам артефактов, контроль изменений исходного кода, фиксация состава версий и зависимостей, проверка корректности конфигураций, тестирование отказоустойчивости, контроль совместимости Cgroup и модуля NGINX/Angie, учет требований к TLS, ролям и административному доступу, а также разделение runtime-, management- и аналитического контуров.

Для критичных изменений предусматривается дополнительная проверка влияния на легитимный трафик, вероятность ложных блокировок, производительность и задержку обработки, совместимость с существующими профилями и возможность безопасного обновления или отката.

4 Управление конфигурацией, исходным кодом и релизными артефактами

4.1 Хранение исходного кода

Исходный код проекта Servicepipe Cybert хранится в группе локальных Git-репозиториях на собственной инфраструктуре ООО «Сервиспайп». Технические средства хранения исходного кода расположены на собственных серверах, размещенных в датацентре в г. Москве. Внешние SaaS-репозитории не используются как основное место хранения исходного текста заявляемого программного обеспечения.

Основные группы репозиториях исходного кода и материалов проекта:

Репозиторий / компонент	Назначение
Серверное ядро Servicepipe Cybert	Исходный код C++-компонентов Cresp, Cresp-ВН, механизмов профилей, правил, счетчиков, поведенческого анализа, клиентских проверок, RPC-интерфейсов, тестов и связанных библиотек.
Модуль Cybert для NGINX/Angie	Исходный код модуля интеграции с фронтенд-сервером, формирование контекста запроса, взаимодействие с Cresp и применение вердикта.
Cresp API и конфигурационный контур	API управления профильной конфигурацией, работа с MariaDB, формирование runtime-конфигурации и сервисы ее доставки на filter nodes.
Backend панели управления Servicepipe	Backend-сервисы и GraphQL/REST API, прикладная логика управления учетными записями, ресурсами, настройками, журналами, аналитикой и интеграциями Cybert.
Frontend панели управления Servicepipe	Frontend панели управления: формы настройки, пользовательские правила, статистика, аналитика и журналы.
Deployment / installer Servicepipe Cybert	Installer, bootstrap-скрипты, Ansible playbook-и, inventory-шаблоны, manifest версий, скрипты упаковки и конфигурационные шаблоны.
Пользовательская и эксплуатационная документация	Исходные материалы руководств, инструкций, описаний архитектуры, функциональных характеристик, жизненного цикла и иных документов сопровождения.

В Git-репозиториях хранятся исходный текст, конфигурации сборки, тесты, документация, сведения о зависимостях и release manifest. Изменения фиксируются в истории репозитория и позволяют восстановить состав исходного кода конкретной версии.

4.2 Хранение объектного кода и поставочных артефактов

Объектный код и релизные артефакты отделены от исходных Git-репозиториях и хранятся во внутренних поставочных хранилищах артефактов. Такой подход обеспечивает разделение процесса разработки исходного текста и процесса распространения готовых пакетов, образов и материалов поставки.

Хранилище	Тип артефактов	Назначение
Внутренний пакетный репозиторий	RPM/пакеты и бинарные пакеты	Хранение пакетов Cresp, Cresp-ВН, модуля Cybert для NGINX/Angie, Cresp API, сервисов управления и иных компонентов, используемых installer-ом.
Внутреннее файловое и контейнерное хранилище	Архивы, Docker/OCI-образы, binary payload, статические frontend-артефакты	Хранение файлов поставки, образов, архивов, собранного frontend и иных материалов, используемых при развертывании.

Релизные артефакты формируются из контролируемого состояния исходного кода и зависимостей. Для каждого релиза фиксируются версия, состав компонентов, используемые зависимости, совместимые форматы конфигурации и материалы развертывания.

4.3 Контроль изменений и трассируемость

Контроль изменений обеспечивается за счет ведения задач разработки, фиксации изменений в Git-репозиториях, код-ревью, подготовки release manifest-a и ведения сведений о составе релиза. Каждое изменение связывается с соответствующим требованием, дефектом, задачей или внутренней инициативой.

Для изменений, затрагивающих алгоритмы классификации, пользовательские правила, формат runtime-конфигурации, взаимодействие Cresp и NGINX/Angie, Cresp API, схему данных MariaDB, интерфейсы панели управления или механизмы безопасности, проводится дополнительный анализ влияния. По результатам анализа определяются необходимые тесты, миграции, изменения документации и порядок внедрения.

Трассируемость позволяет определить, какие изменения исходного кода и конфигурации вошли в конкретную версию, какие артефакты были собраны, какой набор зависимостей использован и какие проверки были выполнены перед выпуском.

5 Персонал, обеспечивающий жизненный цикл ПО

Работы по разработке, сопровождению и развитию ПО «Servicepipe Cybert» выполняются специалистами ООО «Сервиспайп», обладающими соответствующей квалификацией и опытом. В жизненном цикле участвуют продуктовые, инженерные, тестовые, эксплуатационные и технические роли.

Роль	Участие в жизненном цикле	Квалификация
Руководитель продукта	Формирует требования к программному обеспечению, определяет направления развития механизмов веб-защиты, согласует функциональный состав версий, участвует в приемке результатов и подготовке релизов.	Высшее образование; опыт управления продуктом, анализа требований и взаимодействия с пользователями программного обеспечения.
Руководитель разработки / технический лидер	Организует разработку, участвует в архитектурных решениях, распределяет задачи, контролирует качество реализации и согласованность изменений между серверным ядром, API и панелью управления.	Высшее профильное образование; опыт разработки программного обеспечения и проектирования серверных или высоконагруженных систем.
Архитектор / ведущий разработчик серверного ядра	Проектирует и реализует Cresp, Cresp-BH, профили, правила, поведенческий анализ, механизмы клиентских проверок и высокопроизводительную обработку запросов.	Высшее профильное образование; опыт C/C++, сетевого, серверного или высоконагруженного программного обеспечения.
Разработчик модуля NGINX/Angie	Разрабатывает интеграцию фронтенд-сервера с Cresp, передачу параметров запроса, RPC-взаимодействие и применение решений защиты.	Высшее профильное образование; опыт C/C++, NGINX/Angie, HTTP/TLS и сетевого программирования.
Backend-разработчик	Разрабатывает Cresp API, backend панели управления, GraphQL/REST API, бизнес-логику, интеграции, работу с PostgreSQL/MariaDB и сервисные задачи.	Высшее профильное образование; опыт разработки backend-сервисов, API и работы с базами данных.
Frontend-разработчик	Разрабатывает раздел «Защита приложений», формы управления ресурсами и правилами, таблицы, валидацию, локализацию, интеграцию с API и отображение статистики/аналитики.	Высшее профильное образование; опыт разработки web-интерфейсов на React/TypeScript или аналогичных технологиях.
Инженер DevOps / инженер сборки и поставки	Поддерживает сборочную инфраструктуру, installer, bootstrap-скрипты, Ansible playbook-и, manifest-ы версий, сборку пакетов и размещение артефактов.	Высшее профильное образование; опыт Linux-администрирования, автоматизации сборки, контейнеризации и сопровождения инфраструктуры.
Инженер по тестированию	Проводит функциональное, интеграционное, регрессионное, API-, UI-, инсталляционное и нагрузочное тестирование, проверяет исправления и готовность версии к выпуску.	Высшее профильное образование; опыт тестирования серверного, web-или инфраструктурного программного обеспечения.

Роль	Участие в жизненном цикле	Квалификация
Инженер внедрения / эксплуатации	Участвует в установке, обновлении, настройке, интеграции с NGINX/Angie, диагностике и проверке работоспособности Cybert в стендовой или клиентской инфраструктуре.	Высшее профильное образование; опыт Linux, HTTP/TLS, сетевых сервисов, систем мониторинга и эксплуатации веб-инфраструктуры.
Инженер технической поддержки	Принимает и классифицирует обращения пользователей, собирает диагностические данные, взаимодействует с разработчиками при устранении дефектов и консультирует пользователей.	Высшее профильное образование; опыт технической поддержки серверного, сетевого или защитного программного обеспечения.
Технический писатель / системный аналитик	Готовит и актуализирует пользовательскую, эксплуатационную и техническую документацию, описания функций, архитектуры, инструкций и процессов.	Высшее образование; опыт подготовки технической документации и анализа программных продуктов.

Фактический количественный состав специалистов определяется штатным расписанием, планом работ и составом версии. При необходимости к работам привлекаются специалисты смежных направлений: информационной безопасности, сетевой инфраструктуры, мониторинга, сопровождения баз данных и администрирования Linux-систем.

6 Инфраструктура разработки, сборки, тестирования и поддержки

6.1 Инфраструктура хранения и разработки

Инфраструктура разработки программного обеспечения Servicepipe Cybert включает внутренние Git-репозитории, серверы разработки, средства управления задачами, хранилища документации, средства сборки, тестовые стенды и поставочные репозитории артефактов.

Технические средства хранения исходного кода расположены на собственных серверах ООО «Сервиспайп», размещенных в датацентре в г. Москве по адресу: ул. Бутлерова, д. 7, ЦОД Ростелеком. Доступ к инфраструктуре предоставляется уполномоченным сотрудникам в соответствии с их ролями и необходимостью участия в работах по жизненному циклу.

Разработчики программного обеспечения осуществляют деятельность в рамках организационного контура ООО «Сервиспайп». Юридический адрес организации: Россия, 111024, г. Москва, ул. Кабельная 2-я, д. 2, стр. 3, эт. 2, помещение XVIII, ком. 8.

6.2 Инфраструктура сборки и тестирования

Инфраструктура сборки обеспечивает формирование исполняемых файлов и библиотек серверного ядра, модуля Cybert для NGINX/Angie, пакетов Cresp API и сервисов управления, backend и frontend артефактов, Docker/OCI-образов, архивов, installer-a и сопроводительных материалов поставки. Сборка выполняется из контролируемого состояния исходного кода и фиксированного набора зависимостей.

Тестовые стенды используются для проверки Cresp и Cresp-BH, модуля NGINX/Angie, Cresp API, контура MariaDB -> формирование runtime-конфигурации -> Cresp config deliver -> filter nodes, backend и frontend панели управления, а также интеграции с PostgreSQL, ClickHouse, VictoriaMetrics, Kafka и Vector.

Для проверки используются генераторы HTTP(S)-нагрузки, тестовые backend-приложения, наборы сценариев легитимного и автоматизированного трафика, журналы запросов, метрики, средства мониторинга и инструменты анализа сетевого/TLS-трафика. Нагрузочные испытания позволяют оценивать задержку принятия решения, пропускную способность и поведение при масштабировании filter nodes.

Для on-prem поставки дополнительно проверяются установка и обновление на целевой GNU/Linux-среде, взаимодействие filter и management nodes, доставка конфигурации, резервирование management-контура, восстановление после отказов и работоспособность в изолированном или ограниченном по внешнему доступу контуре.

6.3 Инфраструктура технической поддержки

Техническая поддержка программного обеспечения осуществляется службой технической поддержки ООО «Сервиспайп». Служба поддержки обеспечивает прием, регистрацию, классификацию и обработку обращений пользователей, связанных с установкой, настройкой, эксплуатацией, интеграцией, диагностикой и устранением сбойных ситуаций.

Размещение службы технической поддержки осуществляется в организационном контуре ООО «Сервиспайп». Обращения пользователей принимаются по установленным договором каналам связи, а также через публичные контактные каналы организации, включая электронную почту support@servispipe.ru.

Инфраструктура поддержки включает средства регистрации обращений, внутренней коммуникации, хранения эксплуатационных материалов, доступа к документации, сбора диагностической информации и передачи задач в разработку при необходимости выпуска исправления. Для анализа инцидентов могут использоваться журналы ClickHouse, временные ряды VictoriaMetrics, сервисные логи и диагностические данные конкретной инсталляции.

Настоящее описание процессов жизненного цикла применяется к текущей и последующим версиям программного обеспечения Servispipe Cyber и подлежит актуализации при существенном изменении архитектуры продукта, состава компонентов, модели поставки, инфраструктуры разработки или процессов сопровождения.