Код безопасности

Программно-аппаратный комплекс
квалифицированной электронной подписи

"Jinn-Server" версии 1.0

Руководство администратора

RU.88338853.501430.009 01 91

© Компания "Код Безопасности", 2017. Все права защищены.

Все авторские права на эксплуатационную документацию защищены.

Этот документ входит в комплект поставки изделия. На него распространяются все условия лицензионного соглашения. Без специального письменного разрешения компании "Код Безопасности" этот документ или его часть в печатном или электронном виде не могут быть подвергнуты копированию и передаче третьим лицам с коммерческой целью.

Информация, содержащаяся в этом документе, может быть изменена разработчиком без специального уведомления, что не является нарушением обязательств по отношению к пользователю со стороны компании "Код Безопасности".

Почтовый адрес:	**115127, Россия, Москва, а/я 66** **ООО "Код Безопасности"**
Телефон:	**8 495 982-30-20**
E-mail:	**info@securitycode.ru**
Web:	**http://www.securitycode.ru**

Оглавление

Введение **5**

Основные понятия, термины и определения **6**

Глава 1. Общие сведения о ПАК "Jinn-Server" **7**

Сервис проверки ЭП (SVS) 9
Сервис формирования ЭП (SS) 10
Сервис архивирования CRL (CAS) 10
Подсистема администрирования 11

Глава 2. Требования к техническим и программным средствам **13**

Требования к техническим (аппаратным) средствам 13
Состав программных средств 13
Требования и условия организационного характера 13
Требования и условия технического характера 14
Синхронизация времени аппаратных платформ ПАК 14
Фискальная подсистема 14
Средства криптографической защиты информации 16
Требования и условия технологического характера 16
Увеличение производительности и резервирование 16
Сервисы и порты 16
Язык описания веб-сервисов и доступа к ним 17

Глава 3. Установка программного обеспечения ПАК "Jinn-Server" **18**

Начальная установка 19
Переход на новую версию (обновление) 19
Удаление 20

Глава 4. Настройка программного обеспечения ПАК "Jinn-Server" **21**

Конфигурирование сервисов ПАК "Jinn-Server" 21
Сервис проверки ЭП (SVS) 21
Сервис формирования ЭП (SS) 25
Сервис архивирования CRL (CAS-1) 26
Сервис архивирования CRL (CAS-2) 29
Оптимизация производительности 31
Рекомендации по обеспечению отказоустойчивости ОС 32

Глава 5. Рекомендации по анализу статистики работы веб-сервера **34**

Глава 6. Общие указания по порядку технического обслуживания **36**

Глава 7. Подготовка и проверка вспомогательных данных для работы ПАК "Jinn-Server" **37**

Создание криптоконтейнеров для ПАК "Jinn-Server" 37
Создание криптоконтейнеров и запроса на сертификат 37
Добавление выданного сертификата в контейнер 37
Копирование готовых контейнеров на ПАК "Jinn-Server" 37
Настройка прав доступа к криптоконтейнерам и лицензиям "КриптоПро CSP" 37
Добавление считывателя HDIMAGE 38
Настройка прав доступа к контейнерам 38
Проверка прав доступа к контейнерам 38
Редактирование конфигурационных файлов (csm.conf, tccs.ss, tccs.svs) 39
Настройка прав доступа к конфигурационным файлам (csm.conf, tccs.ss, tccs.svs) 40
Редактирование файлов веб-сервисов 40
Запуск "КриптоПро CSP" 41
Добавление сертификата УЦ и CRL через веб-интерфейс 41
Работа с веб-интерфейсом 41
Проверка сертификата с помощью веб-интерфейса и утилиты dumpasn1 44
Проверка прав доступа к файлам pem 46

Добавление скрипта для отключения свопа 47
Редактирование файла hosts 47
Логирование 47

Приложение 49
База данных сервиса CAS 49
Структура таблиц 49
Создание таблиц 53

Документация 58

Введение

Данное руководство предназначено для администраторов изделия "Программно-аппаратный комплекс квалифицированной электронной подписи "Jinn-Server" версии 1.0" (далее – ПАК "Jinn-Server", комплекс, ПАК). В нем содержатся сведения, необходимые для установки, настройки и администрирования программного обеспечения ПАК "Jinn-Server" на компьютерах, функционирующих под управлением следующих дистрибутивов Linux:

Табл. 1. Дистрибутивы Linux, поддерживаемые ПАК "Jinn-Server"

Название дистрибутива	Версия ядра
Альт Линукс СПТ 6.0.0 x86/x64	2.6.32-el-smp-alt10.0.M55C.1
CentOS 7.0 x64	1406

Условные обозначения

В руководстве для выделения некоторых элементов текста (примечаний и ссылок) используется ряд условных обозначений.

Внутренние ссылки обычно содержат указание на номер страницы с нужными сведениями. Ссылки на другие документы или источники информации размещаются в тексте примечаний или на полях.

Важная и дополнительная информация оформлена в виде примечаний. Степень важности содержащихся в них сведений отображают пиктограммы на полях.

- Так обозначается дополнительная информация, которая может содержать примеры, ссылки на другие документы или другие части этого руководства.
- Такой пиктограммой выделяется важная информация, которую необходимо принять во внимание.
- Эта пиктограмма сопровождает информацию предостерегающего характера.

Исключения. Некоторые примечания могут не сопровождаться пиктограммами. А на полях, помимо пиктограмм примечаний, могут быть приведены другие графические элементы, например, изображения кнопок, действия с которыми упомянуты в тексте расположенного рядом абзаца.

Другие источники информации

Сайт в Интернете. Если у вас есть доступ в Интернет, вы можете посетить сайт компании "Код Безопасности" (http://www.securitycode.ru/) или связаться с представителями компании по электронной почте (support@securitycode.ru).

Учебные курсы. Освоить аппаратные и программные продукты компании "Код Безопасности" можно в авторизованных учебных центрах. Перечень курсов и условия обучения представлены на сайте компании http://www.securitycode.ru/company/education/training-courses/. Связаться с представителем компании по вопросам организации обучения можно по электронной почте (education@securitycode.ru).

Основные понятия, термины и определения

Термин	Определение
ИОК/PKI	Инфраструктура открытых ключей – комплекс программных и/или программно-аппаратных средств и организационно-технических мероприятий по обеспечению использования криптографии с открытым ключом, управления этими ключами и сертификатами, в частности, для решения задач защищенного электронного документооборота
CAS	CRL Archiving Service – сервис, предназначенный для хранения и автоматического обновления списков отозванных сертификатов и обновлений к ним с целью последующего использования хранимых CRL другими компонентами ПАК
CFV	Cetificate Format Validation – составная часть сервиса SVS, предназначенная для проверки сертификатов открытых ключей на квалифицированность
CRL/COC	Список отозванных сертификатов (основной, регулярный)
deltaCRL	Обновление к COC, выпускаемое УЦ в интервале между выпусками COC
DMZ	Демилитаризованная зона, ДМЗ – технология обеспечения защиты информационного периметра, при которой серверы, отвечающие на запросы из внешней сети, находятся в особом сегменте сети и ограничены в доступе к основным сегментам сети с помощью межсетевого экрана (файрвола) с целью минимизировать ущерб при взломе одного из общедоступных сервисов, находящихся в ДМЗ
SS	SigningService – сервис, предназначенный для формирования ЭП под некоторыми данными
SVS	SignatureValidationService – сервис, предназначенный для проверки данных, подписанных ЭП
ГУЦ	Головной удостоверяющий центр
ОС	Операционная система
ПАК	Программно-аппаратный комплекс
ПО	Программное обеспечение
СКЗИ	Средство криптографической защиты информации
УЦ	Удостоверяющий центр
ЭД	Электронный документ
ЭП	Электронная подпись

Глава 1
Общие сведения о ПАК "Jinn-Server"

Настоящее руководство предназначено для использования администратором ПАК "Jinn-Server" на этапах установки и технического обслуживания ПАК.

При установке и техническом обслуживании необходимо использовать следующие эксплуатационные документы:

Эксплуатационная документация[1]
Программно-аппаратный комплекс квалифицированной электронной подписи "Jinn-Server" версии 1.0. Руководство администратора
Программно-аппаратный комплекс квалифицированной электронной подписи "Jinn-Server" версии 1.0. Руководство программиста
Программно-аппаратный комплекс квалифицированной электронной подписи "Jinn-Server" версии 1.0. Руководство пользователя

ПАК "Jinn-Server", а также набор дополнительных программных средств предназначены для выполнения функции автоматической проверки и формирования электронной подписи с сетевой выгрузкой результата во внешний сервис.

Общая схема развертывания компонентов ПАК "Jinn-Server" и их взаимодействие с внешними информационными системами представлены на рисунке 1.

Рис. 1. Общая схема развертывания компонентов ПАК "Jinn-Server"

При использовании кластеризации число необходимых внешних сборщиков CRL (CAS-2), размещаемого в DMZ, будет диктоваться в первую очередь соображениями обеспечения надежности ПАК и никак не связано с числом узлов ПАК, расположенных в защищенном сегменте сети.

Масштабирование и отказоустойчивость ПАК реализуется с соблюдением следующих условий:

- один сборщик CRL (CAS-2) может взаимодействовать только с одним сервисом хранения CRL (CAS-1), т.е. для обеспечения отказоустойчивости необходимо иметь как минимум две пары серверов CAS-2–CAS-1;
- для обеспечения отказоустойчивости СУБД доступна настройка репликации данных между двумя и более экземплярами штатными средствами администрирования СУБД;
- несколько сервисов SS и SVS в процессе работы могут быть настроены на обращение к одному или разным серверам CAS-1 – в зависимости от того, сколько серверов (сервисов) CAS-1 имеется в наличии (развернуто), т.е. с одним CAS-1 могут взаимодействовать два и более сервиса SS и SVS.

Описание приводимых аббревиатур см. на стр. 8 настоящего руководства.

[1] Эксплуатационная документация поставляется в электронном виде в формате PDF (Adobe Acrobat Reader) и/или ODF, если не указано иного.

При использовании кластеризации возможно вынесение сборщика CRL внутри ПАК "Jinn-Server" (CAS-1) на отдельный физический сервер. При этом доступ к функциям сервиса будет производиться по протоколу TCP/IP. При этом доступ к функциям CAS-1 может осуществляться от нескольких экземпляров (до 10 штук) сервисов SS и SVS. Серверы, на которых развернуты сервисы SS и SVS, должны находиться в одной сети с сервисом CAS-1.

Архив собранных CRL может передаваться между CAS-1 и CAS-2 с помощью любого отчуждаемого носителя, а также с помощью онлайн-TCP-транспорта между компонентами.

ПАК "Jinn-Server" построен по модульному принципу и содержит в своем составе следующие компоненты (подсистемы):

- Сервис проверки ЭП (SVS – SignatureValidationService) – предназначен для проверки данных, подписанных ЭП, и проверки сертификатов ЭП на квалифицированность и действительность.
- Сервис формирования ЭП (SS – SigningService) – предназначен для формирования ЭП под некоторыми данными.
- Сервис архивирования CRL (CAS – CRLArchivingService) – предназначен для хранения и автоматического обновления списков отозванных сертификатов и обновлений к ним с целью последующего использования хранимых CRL другими компонентами ПАК. Этот сервис разделен на два модуля – внутренний сборщик CRL (CAS-1) и внешний (CAS-2). Внешний сборщик, ориентируясь на пути к CRL, публикуемые аккредитованными УЦ, выполняет загрузку указанных CRL по настраиваемому расписанию, а также передачу загруженных CRL внутреннему сборщику, к которому в свою очередь обращаются сервисы SS и SVS.
- Подсистема администрирования – предназначена для мониторинга и управления компонентами ПАК.

Доступ к криптографическим функциям производится с использованием MicrosoftCryptoAPI для СКЗИ "КриптоПро CSP" для платформы Linux.

При формировании списка доверенных издателей предусмотрена возможность загрузки TSL-списка, формат которого должен соответствовать описанию, доступному по ссылке (http://e-trust.gosuslugi.ru/docs/TSLExt.1.0.xsd), которое основано на ETSI TS 102 231 V3.1.2 (2009-12) Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information. При этом обеспечивается:

- автоматическое получение самого TSL-файла с сайта его официальной публикации;
- проверка авторства и целостности полученного TSL-файла (сертификат автора подписи должен доставляться доверенным образом и передаваться процедуре проверки TSL-файла через специальный интерфейс);
- формирование на основе TSL-файла списка доверенных УЦ в хранилище ПАК "Jinn-Server".

В административном интерфейсе ПАК "Jinn-Server" предусмотрены инструменты, позволяющие:

- управлять адресом публикации TSL и режимом его скачивания;
- осуществлять управляемую пообъектно загрузку данных из TSL.

При этом возможности администратора при обработке издателей, представленных в TLS, обеспечиваются:

- в процессе загрузки данных из TSL – возможностью отказаться от регистрации нового издателя (например, в случае ошибок в формате представления сертификата издателя);
- в процессе управления зарегистрированными издателями – редактированием списка точек публикации COC и возможностью последующего импорта COC.

В рамках обеспечения поддержки TSL в ПАК "Jinn-Server" предусмотрена возможность установки и (в случае необходимости) замены самоподписанного сертификата головного УЦ (ГУЦ), а также сертификатов промежуточных издателей, которые должны использоваться в процессе проверки загружаемых TSL. Механизм управления указанными сертификатами отделен и независим от управления списком издателей сертификатов конечных пользователей.

Для реализации проверки цепочки сертификатов длиной более 1 должен использоваться атрибут authorityCertSerialNumber.

Кроме того, алгоритм проверки предусматривает выборочную обработку(проверку) расширений, перечисленных в стандарте RFC5280 и влияющих на результат проверки цепочки сертификатов, например, расширений CertificatePolicies, NameConstraints, PolicyConstraints, PolicyMappings, inhibitAnyPolicy и других.

Выбор режима проверки цепочки сертификатов должен указываться в конфигурационном файле компонента SVS ПАК "Jinn-Server".

Сервис проверки ЭП (SVS)

Внутренняя структура и взаимодействие сервиса SVS с другими компонентами ПАК и внешними системами представлены на рисунке 2.

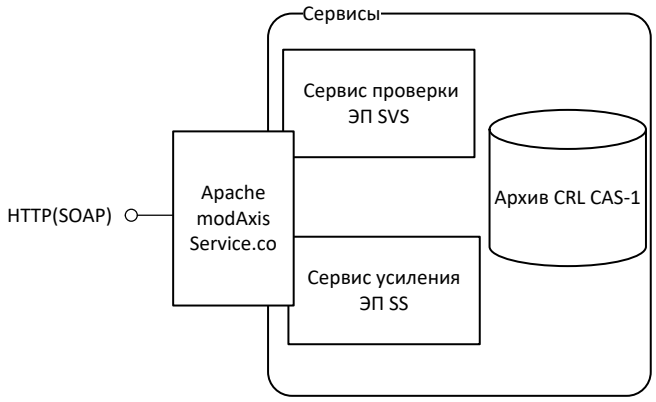


Рис. 2. Структура сервиса SVS

Сервис SVS оформлен как веб-сервис для доступа к нему по протоколу HTTP, он поддерживает:

- форматы подписи – CMS SignedData, XMLdsig (enveloped-подпись), WS-Security (detached-подпись) XAdES-C, CAdES-C, XAdES-T, CAdES-T, XAdES-A, CAdES-A;
- криптографические алгоритмы ГОСТ Р 34.10–2001, ГОСТ Р 34.11–94, ГОСТ Р 34.10–2012, ГОСТ Р 34.11–2012.

В зависимости от параметров, содержащихся в запросе, проверка подписи может осуществляться по двум сценариям:

- только проверка – в этом случае проверяются все подписи под документом и итоговый результат содержит информацию по каждой из них;
- проверка и усиление подписи – проверяется и усиливается только одна "внешняя" по отношению к остальным подпись, если документ или его части подписаны несколькими подписями. В запросе должна присутствовать идентифицирующая усиливаемую подпись информация.

При наличии в CMS SignedData содержимого типа SignedData (вложенная подпись) производится проверка и этих данных.

Заверяющие подписи (counterSignature) не поддерживаются.

Для приведения xml к каноническому виду используется метод xml-exc-c14n.

Проверка сертификатов авторов ЭП производится с учетом CRL, загружаемых сервисом архивирования CRL. К обработке принимаются только CRL и обновления к ним, непосредственно подписанные ключом того же доверенного издателя, что и проверяемый сертификат, либо, в случае indirectCRL, ключом специально выделенного для выпуска CRL сертификата, также непосредственно подписанным соответствующим издателем. В случае использования indirectCRL сертификат, выделенный соответствующим издателем для выпуска CRL, должен устанавливаться в ПАК административными средствами, аналогично сертификату издателя.

Для проверки блоков XML-данных используются следующие типы проверок:

- проверка непосредственно переданного XML, если входными аргументами не указано иное;
- проверка отдельного элемента, входящего XML, определяемого по переданному в аргументах ID.

Результаты проверки содержат в виде XML всю информацию, находящуюся в информационной части сертификата автора проверяемой подписи.

В SVS присутствует поддержка проверки отсоединенных подписей, в этом случае исходные данные должны передаваться дополнительным параметром в запросе к сервису.

В зависимости от характера запроса SVS либо просто проверяет подпись под полученным документом, либо, в случае успешной проверки подписи и сертификата автора, проверяемые данные дополняются необходимыми атрибутами, с тем чтобы привести обрабатываемые данные, в зависимости от исходного формата, в соответствие спецификациям XAdES-C, CAdES-C, XAdES-T, CAdES-T, XAdES-A, CAdES-A, для формата WS-Security усиление выполняется аналогично XAdES-C.

В сервис SVS встроен сервис CFV, который можно вызвать для проверки сертификатов ЭП на соответствие x.509 v 3.

Сервис формирования ЭП (SS)

Структурно сервис SS представлен на рисунке 3.

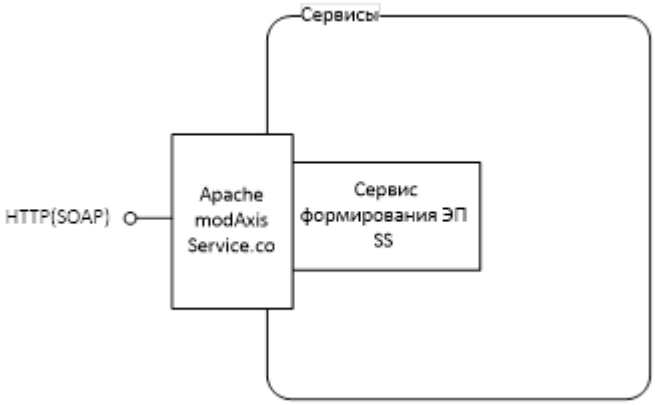


Рис. 3. Структурная схема сервиса SS

Сервис SS оформлен как веб-сервис для доступа по HTTP, он поддерживает:

- форматы подписи – CMS SignedData, XMLdsig (enveloped-подпись), XAdES-T, WS-Security (detached-подпись), XAdES-A;
- криптографические алгоритмы ГОСТ Р 34.10–2001, ГОСТ Р 34.11–94, ГОСТ Р 34.10–2012, ГОСТ Р 34.11–2012.

Также этот веб-сервис содержит функционал, предоставляющий пользователям возможность выработки хэш-функции от произвольных бинарных данных.

Сервис архивирования CRL (CAS)

CRL (CAS) предоставляет пользователям (компонентам ПАК) комплект из регулярного CRL и последнего обновления к нему, если технические средства конкретного УЦ поддерживают публикацию deltaCRL от заданного издателя на заданную дату. Если на заданную дату в хранилище сервиса будет обнаружено несколько актуальных регулярных CRL, пользователю возвращается самый свежий из них (имеющий самую позднюю дату выпуска).

Сервис (CAS-1) спроектирован с учетом необходимости размещения ПАК в защищенном сегменте сети, не имеющем доступа к сети Интернет.

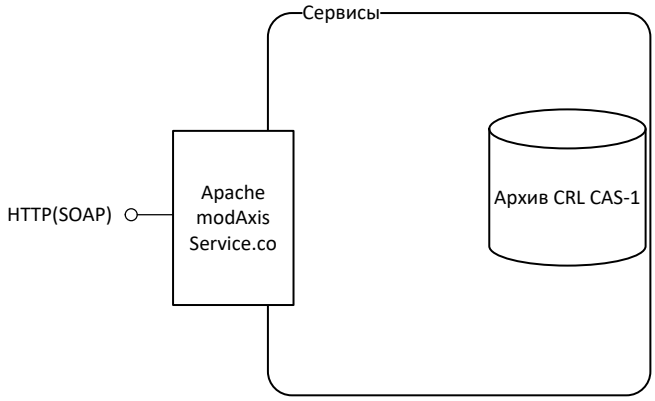


Рис. 4. Сервис внутренний сборщик CRL (CAS-1)

Компонент сервиса CAS, предназначенный для загрузки CRL и обновлений к ним из точек публикации (CAS-2), размещается в отдельном сегменте сети, откуда доступ в Интернет возможен, а коммуникация между CAS-2 и остальными компонентами ПАК осуществляется в офлайн-режиме посредством передачи файлов на отчуждаемых носителях.

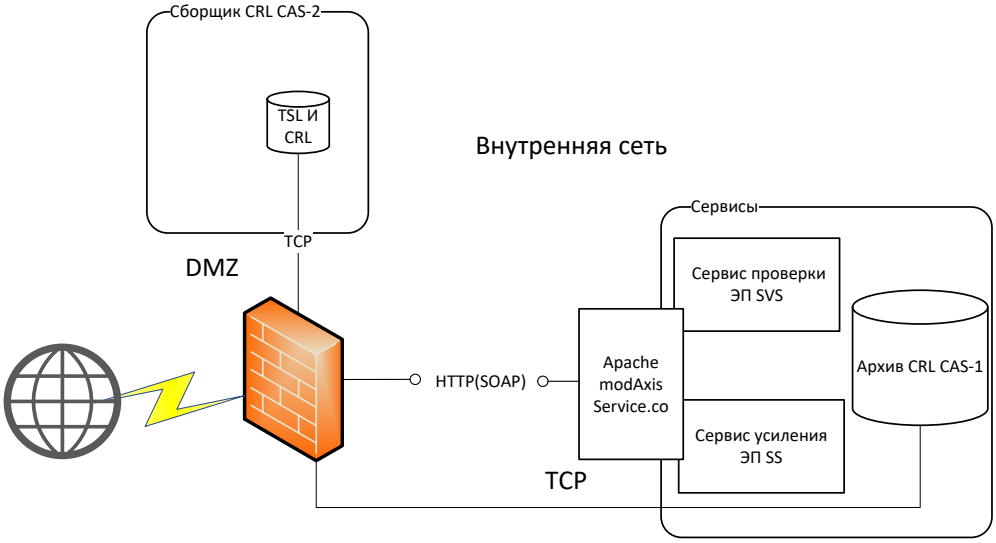


Рис. 5. Сервис внешний сборщик CRL (CAS-2)

Поддерживаемый протокол доступа к точкам публикации CRL – http.

Точка публикации CRL должна определяться на основании соответствующего (стандартного, CDP или FreshestCRL соответственно для регулярных CRL и их обновлений) расширения проверяемого сертификата. В случае отсутствия такого расширения должна использоваться точка публикации CRL по умолчанию.

Загрузка регулярных CRL должна начинаться:

- для отсутствующего CRL – немедленно;
- для очередного регулярного CRL – за некоторое фиксированное время до наступления даты, указанной в поле nextUpdate имеющегося CRL.

Подсистема администрирования

Подсистема предназначена для мониторинга и управления компонентами ПАК, в том числе она:

- Обеспечивает мониторинг компонентов системы с автоматическим восстановлением работоспособности ПАК в случае программных сбоев и уведомлением администратора в случае необходимости.
- Обеспечивает настройку ПАК для использования эталонного источника времени по протоколу NTP.

- Предоставляет средства управления списком доверенных издателей и параметрами сервиса архивирования CRL, в том числе:
  - средства импорта/экспорта сертификатов доверенных издателей и соответствующих им CRL. Средства экспорта обеспечивают выгрузку данных как для одного, так и для нескольких или всех установленных издателей;
  - возможность отметить определенного издателя как неактивного (в этом случае загрузка/обновление соответствующих CRL производиться не будет) либо полностью удалить издателя и все соответствующие ему CRL;
  - возможность задания/изменения точки публикации CRL по умолчанию для определенного издателя;
  - возможность задания упреждающего периода для загрузки очередного регулярного CRL;
  - возможность принудительной загрузки/обновления CRL для одного, нескольких или всех активных издателей.
- Предоставляет средства управления сертификатами и ключами, используемыми для выработки штампов времени и формирования ЭП в SS. В настоящее время установленные в системе контейнеры CSP защищены фиксированным предопределенным паролем.

Глава 2
Требования к техническим и программным средствам

Требования к техническим (аппаратным) средствам

Аппаратная платформа опытного образца ПАК "Jinn-Server" реализована в виде двух промышленных системных блоков:

- Блок, содержащий программный модуль внешний сборщик CRL (CAS-2), размещаемый в DMZ и имеющий выход в публичные телекоммуникационные сети. Специальных требований к аппаратной платформе не выдвигается. Для передачи данных внутреннему сборщику CRL (CAS-1) требуется USB-порт для отчуждаемого носителя или канал связи TCP/IP.
- Блок, содержащий программные модули SS, SVS, CAS-1, подсистему администрирования. Аппаратная платформа для обеспечения требований по производительности должна быть построена на серверных чипсетах Intel и укомплектована серверным процессором семейства Intel Xeon 5000 или более современным, с количеством вычислительных ядер не менее 6 и тактовой частотой не менее 2,4 ГГц, а также оперативной памятью не менее 64 ГБ. Наличие разъема PCI-Express для обеспечения возможности установки электронного замка "Соболь". 2 USB-порта.

Аппаратные платформы должны иметь возможность подключения клавиатуры, монитора и компьютерной мыши непосредственно или через KVM (keyboard, video and mouse).

Электропитание должно осуществляться через устройство бесперебойного питания. Резервное и архивное копирование информационных активов ПАК осуществляется на внешние USB-флеш-накопители общесистемными средствами.

Состав программных средств

Для развертывания ПАК используется дистрибутив Cent OS 7.0 для платформы x86-64.

В частности, на основании п. 2.2 лицензии используются компиляторы, загрузчики, препроцессоры, библиотеки, пакеты, сборки и т.п., входящие в состав данного дистрибутива.

Требования и условия организационного характера

Для обеспечения работоспособности ПАК "Jinn-Server" оперативный персонал службы, ответственной за эксплуатацию данного программного изделия, подразделяется на три ролевые группы сотрудников[2], которые в своей работе должны придерживаться требований соответствующей эксплуатационной документации на ПАК, а также сопроводительной документации на составные аппаратные и/или программные модули стороннего производства.

На системного программиста (администратора) возлагаются:

- задача поддержания работоспособности технических средств;
- задача установки (инсталляции) и поддержания работоспособности системных программных средств – операционной системы;
- задача установки (инсталляции) и поддержания работоспособности обеспечивающего ПО, СПО и т.п.;
- задача установки (инсталляции) и поддержания работоспособности программы ПАК "Jinn-Server";
- ведение резервных копий общесистемного и прикладного ПО;

[2] Ограничений на возможность совмещения обязанностей не накладывается.

- отдельной группой задач стоят задачи обеспечения информационной безопасности, работы с ключевым материалом и обеспечения штатного функционирования СКЗИ и т.д.

Обязанности программиста и пользователя см. в [2, 3].

Требования и условия технического характера

Синхронизация времени аппаратных платформ ПАК

Для синхронизации событий и журналов различных компонентов ПАК на каждой отдельной аппаратной платформе комплекса размещен специальный сервис в режиме подстройки локальных аппаратных часов по внешнему эталонному источнику времени. Реализация данного сервиса основана на использовании сетевого протокола синхронизации времени Network Time Protocol (NTP).

Фискальная подсистема

Фискальная подсистема ПАК образована совокупностью рабочих журналов компонентов системы с механизмами их формирования и включает в себя:

- Журнал функционального гипертекстового сервера.
- Журнал приложений.
- Системный журнал.

Настройки, касающиеся журналов ПАК, вносятся в конфигурационный файл системного компонента syslog (за исключением журналов функционального гипертекстового сервера).

Администратор системы должен периодически проводить анализ и создавать архивные копии рабочих журналов на отчуждаемых носителях, а также контролировать наличие свободного дискового пространства.

Автоматический контроль функционирования гипертекстовых серверов и сервисных демонов

Компоненты ПАК контролируются специальной программой (**tccs_watchdog**), режим запуска которой определяется в конфигурационных файлах системного сервиса с**rond**. Использование **tccs_watchdog** обеспечивает проверку работоспособности компонентов ПАК и их повторный запуск в случае принудительного или аварийного завершения.

Уведомления о нештатных событиях в системе

В случае недоступности какого-либо из контролируемых сервисов **tccs_watchdog** осуществляет попытку его перезапуска, а также формирует и отправляет электронные письма – уведомления о нештатной ситуации с помощью внешнего сценария /opt/tccs/bin/notifier.sh, реализующего правила и технологию доставки электронных писем администраторам (адресатам). Сценарий содержит вызов программы, которая непосредственно осуществляет доставку электронных писем, почтовые адреса, на которые высылаются письма, почтовый сервер, почтовый адрес отправителя и т.п. Также программа **tccs_watchdog** может быть запущена с параметром **notifier** (/opt/tccs/bin/tccs_watchdog **notifier**). В этом случае осуществляется отправка уведомлений не только о некорректной работе сервисов, но и об успешном их выполнении. Адрес получателя, а также включение/выключение почтовых уведомлений определяется с помощью параметров конфигурационного файла /opt/tccs/etc/csm.conf, указанным ниже параметрам должны быть присвоены приводимые на рисунке значения (email должен указываться актуальный, email на рисунке приведен для примера):

```
"notification_enable": "yes",
"notification_email": "admin@domain.ru"
```

Данные, необходимые для контроля работы процессов, расположены в конфигурационном разделе /opt/tccs/etc/csm.conf.

Пример:

```
"watchedservice_tccs.admin_enable": [
{ "hostname": "srv-jinn-cas-1" },
{ "port": 80 },
{ "socket": "" },
{ "action": "/opt/tccs/etc/init.d/tccs.admin" },
{ "description": "Графический интерфейс подсистемы
администрирования" }
],
"watchedservice_cas1d_enable": [
{ "hostname": "srv-jinn-cas-1" },
{ "port": 11112 },
{ "socket": "" },
{ "action": "/opt/tccs/etc/init.d/cas1d" },
{ "description": "Сервис сбора CRL от аккредитованных
УЦ" }
],
"watchedservice_psqld_enable": [
{ "hostname": "srv-jinn-cas-1" },
{ "port": 0 },
{ "socket": "/tmp/.s.PGSQL.5432" },
{ "action": "/opt/tccs/etc/init.d/psqld" },
{ "description": "База данных" }
],
"watchedservice_tccs.svs_enable": [
{ "hostname": "srv-jinn-cas-1" },
{ "port": 0 },
{ "socket": "/var/opt/tccs/run/tccs.svs" },
{ "action": "/opt/tccs/etc/init.d/tccs.svs" },
{ "description": "Сервис проверки и усиления ЭП" }],
```

Строка описания сервиса, работу которого нужно контролировать, должна заканчиваться словом (директивой) **enable**; для того чтобы снять сервис с контроля, нужно использовать слово (директиву) **disable**, как показано ниже:

"watchedservice_psqld_enable"

"watchedservice_psqld_disable"

Периодичность запуска **tccs_watchdog** определяется настройками crond и описана в конфигурационном файле /jail/XXX/etc/crontab (см. ниже).

```
*/1   *  *  *  *  root
/usr/local/libexec/process_analysis >/dev/null
1     3  *  *  *  root
/usr/local/libexec/process_analysis notifier >/dev/null
```

Демоном crond осуществляется запуск tccs_watchdog в двух режимах – каждую минуту и 1 раз в сутки, с указанием параметра notifier.

Первый режим используется для ежеминутного отслеживания работоспособности сервисов. Если какой-либо сервис оказывается недоступным – программа tccs_watchdog отправляет почтовое уведомление администратору системы (администраторам системы по списку) и автоматически пытается перезапустить сервис. После попытки перезапуска сервиса администратору(ам) отправляется

повторное почтовое уведомление, информирующее его либо об успешном запуске сервиса, либо о том, что сервис запустить не удалось и требуется непосредственное участие администратора. Последующие попытки запуска сервиса при отрицательном результате уже не будут сопровождаться почтовыми уведомлениями. В случае если сервис был перезапущен автоматически или в ручном режиме, т.е. изменился статус работоспособности сервиса с "остановлен" на "запущен", автоматически будет создано почтовое уведомление с указанием наименования сервиса и его текущего статуса.

Второй режим используется для ежедневного итогового отчета о работоспособности всех отслеживаемых сервисов с указанием, какие из присутствующих сервисов на момент составления отчета находятся в работоспособном состоянии.

Средства криптографической защиты информации

Для формирования и проверки электронной подписи в ПАК используется сертифицированное ФСБ России СКЗИ (вплоть до класса КС2 включительно, в части защиты информации, не содержащей сведений, составляющих государственную тайну) "КриптоПро CSP" 4.0 для платформы Linux (ia32, x64) производства компании ООО "КРИПТО-ПРО". Более полная информация приведена в сопроводительной документации на изделие, входящей в комплект поставки СКЗИ.

Для формирования итеративного хэша используется встраиваемая библиотека SCCrypt.

Требования и условия технологического характера

Для функционирования ПАК в рабочем режиме не требуется обеспечение каких-либо особых требований и условий технологического характера.

Увеличение производительности и резервирование

Для достижения требуемых показателей производительности модули ПАК "Jinn-Server" могут быть сгруппированы в кластер, где балансировщиком нагрузки может выступать любой внешний прокси-сервис, входящий в конфигурацию программно-аппаратных комплексов прикладного уровня.

Архитектурная возможность работы ПАК в составе кластера обеспечивает его масштабируемость и отказоустойчивость.

Сервисы и порты

ПАК "Jinn-Server" предоставляет/использует следующие сервисы и порты, представленные в таблице 2. Данная информация может быть использована для конфигурации внешних средств защиты, таких как пакетные фильтры, межсетевые экраны и т.п.

Табл. 2. Представленные сервисы (порты) ПАК

Компонент службы	Порт	Характеристика сервиса
CAS-1	TCP_80	Гипертекстовый интерфейс управления и администрирования CAS-1
	TCP_8080	Транспортный протокол HTTP, обслуживающий запросы к сервисам формирования и проверки ЭП
	TCP_11112	CRL Archiving Daemon
	UDP_161, 162	Службы SNMP
	TCP_22	SSH для удаленного доступа
	UDP_53 исходящее соединение	Доступ к DNS-серверу(ам)
CAS-2	TCP_11113	Сборщик COC
	UDP_161, 162	Службы SNMP
	TCP_22	SSH для удаленного доступа
	UDP_53 исходящее соединение	Доступ к DNS серверу(ам)

Компонент службы	Порт	Характеристика сервиса
АРМ РКС	TCP_8083	Гипертекстовый интерфейс управления и администрирования АРМ РКС, доступ к интерфейсу проверки ЭЦП документов и протоколам проверки
	UDP_161, 162	Службы SNMP
	TCP_22	SSH для удаленного доступа
	UDP_53 исходящее соединение	Доступ к DNS-серверу(ам)

На каждой аппаратной платформе ПАК для синхронизации времени от внешнего источника используется штатная утилита ОС "клиент NTP", запуск которой возможен в ручном режиме или по расписанию (протокол UDP, порт 123).

Язык описания веб-сервисов и доступа к ним

Язык описания веб-сервисов и доступа к ним реализован на WSDL (Web Services Description Language). В документе WSDL определяется веб-сервис с помощью следующих основных элементов:

Элемент	Определяет
<portType>	Методы, предоставляемые веб-сервисом
<message>	Сообщения, используемые веб-сервисом
<types>	Типы данных, используемые веб-сервисом
<binding>	Протоколы связи, используемые веб-сервисом

Основная структура документа WSDL выглядит следующим образом:

```
<definitions>
<types>
   определение типов........
</types>

<message>
   определение сообщения....
</message>

<portType>
   определение порта........
</portType>

<binding>
   определение связей.......
</binding>

</definitions>
```

Документ WSDL может также содержать другие элементы, например, элементы расширения и элемент service, который позволяет объединить вместе в одном отдельном документе WSDL определения нескольких веб-сервисов.

Описание сервисов на языке WSDL представлено в [2].

Глава 3
Установка программного обеспечения ПАК "Jinn-Server"

В состав программного обеспечения ПАК "Jinn-Server" входит набор бинарных RPM-пакетов, поставляемых на установочном диске. Установка и настройка ПАК производится путем запуска специальных скриптов, позволяющих автоматически выполнить установку ПО. ПО ПАК "Jinn-Server" может поставляться в виде дистрибутива, записанного на CD, для установки на аппаратные платформы заказчика или в виде аппаратной платформы АПКШ "Континент" с предустановленным ПАК "Jinn-Server".

На аппаратные платформы – компоненты ПАК – устанавливается ОС, заявленная в требованиях к ПО (CentOS 7.0).

Специальных требований к составу системных сервисов не предъявляется, Установленные серверы должны иметь правильно разрешаемые DNS-имена, сервер с компонентом CAS-2 должен иметь возможность выхода в Интернет и разрешать DNS-имена ресурсов в сети Интернет.

Запуск установочных или конфигурационных скриптов выполняется от имени суперпользователя root. Состав скриптов для варианта поставки вместе с аппаратной платформой (Setting*.sh – для настройки, передается с АПКШ "Континент") приведен в таблице ниже:

Имя скрипта и путь	Описание
/home/admin/conf/Setting_cas1.sh	Позволяет произвести финальную настройку конфигурации модуля CAS-1, развернутого на АПКШ "Континент"
/home/admin/conf/Setting_cas2.sh	Позволяет произвести финальную настройку конфигурации модуля CAS-2, развернутого на АПКШ "Континент"
/home/admin/conf/Setting_csa.sh	Позволяет произвести финальную настройку конфигурации модуля АРМ РКС, развернутого на АПКШ "Континент"

Для запуска настроечного скрипта необходимо выполнить команды:

```
cd /home/admin/conf/
./Setting_*.sh  – вместо * указать установленный модуль.
```

Состав скриптов для варианта поставки – только ПО (jinn*.sh – для установки/обновления/удаления, передается с диском) приведен в таблице ниже:

Имя скрипта и путь	Описание
/home/user/rpmdist*/ jinn_install.sh	Позволяет произвести установку и настройку: • Сборщика CAS-1 и прикладных сервисов. • Только прикладных сервисов (в случае использования нескольких аппаратных платформ для масштабирования). • Сборщика CAS-2. • АРМ РКС. • Прав и разрешений для контейнеров
/home/user/rpmdist*/ jinn_update.sh	Позволяет произвести обновление до новой версии: • Сборщика CAS-1 и прикладных сервисов. • Только прикладных сервисов (в случае использования нескольких аппаратных платформ). • Сборщика CAS-2. • АРМ РКС
/home/user/rpmdist*/ jinn_uninstall.sh	Позволяет произвести удаление: • Сборщика CAS-1 и прикладных сервисов. • Только прикладных сервисов (в случае использования нескольких аппаратных платформ). • Сборщика CAS-2. • АРМ РКС

Начальная установка

Для установки ПО:

1. Чтобы убедиться в отсутствии уже установленных компонентов ПАК, перед установкой ПО выполните команду:

```
rpm -q -a | grep tccs
```

   Если в системе не найдено установленных компонентов ПАК, перейдите к п. **2**.

2. Вставьте в CD-ROM диск с дистрибутивом ОС Linux.

3. Выполните следующие команды:

```
mkdir /mnt/iso/
mount /dev/sr0 /mnt/iso
```

4. Выполните команды для запуска установочного скрипта дистрибутива (путь к дистрибутиву ПО ПАК может отличаться от приведенного в примере):

```
 cd /home/USER/rpmdist*
./jinn_install.sh
```

   После запуска будут выведены варианты установки:

```
[root@cpro-4 rpmdist-tccs-2.0.0-1-centos-7_1_1503-x86_64]# ./jinn_install.sh
Диск успешно смонтирован
Выберите вариант установки:
    1 - Установить сервис архивирования CAS1;
    2 - Установить сервисы подписи и проверки и усиления подписи;
    3 - Установить сервис архивирования CAS2;
    4 - Установить АРМ РКС;
    5 - Настройка прав и разрешений для контейнеров;
    6 - Установка клиентской части NTP-сервера;
    0 - Выход.
```

5. Введите цифру, соответствующую нужному варианту установки, и нажмите клавишу <Enter>.

   В процессе выполнения установки производится проверка наличия прикладных данных, сохраненных в системе в результате функционирования предыдущих версий ПО. Пользователю предоставляется возможность сохранения или удаления обнаруженных данных, если данные найдены:

```
В системе найдены прикладные данные (от предыдущей
версии rpm пакета). Удалить данные?
```

   В ходе установки запрашиваются параметры, необходимые для взаимодействия сервисов (компонентов) ПАК. Изменения в конфигурационные файлы могут быть позже внесены вручную.

6. После успешного окончания установки перезагрузите компьютер для вступления в силу внесенных изменений.

Переход на новую версию (обновление)

Для перехода на новую версию продукта:

1. Убедитесь, что на сервере установлены компоненты ПО ПАК, выполнив команду:

```
rpm -q -a | grep tccs
```

2. Если ПО в системе установлено, выполните команду для запуска скрипта, обновляющего дистрибутив:

```
cd /home/USER/rpmdist*
./jinn_update.sh
```

После запуска будут выведены варианты обновления:

```
[root@cpro-4 rpmdist-tccs-2.0.0-1-centos-7_1_1503-x86_64]# ./jinn_update.sh
Выберите вариант обновления:
        1 - Обновить сервис архивирования CAS1;
        2 - Обновить сервисы подписи и проверки и усиления подписи;
        3 - Обновить сервис архивирования CAS2;
        4 - Обновить АРМ РКС;
        0 - Выход
```

3. Введите цифру, соответствующую нужному варианту обновления, и нажмите клавишу <Enter>.

   Все прикладные данные, используемые в системе предыдущей версией дистрибутива, сохраняются в системе для последующего использования их в новой версии.

   Если необходимо осуществить переход на новую версию без сохранения прикладных данных, выполните удаление текущей версии дистрибутива без сохранения данных и установите новую версию.

Удаление

Для удаления ПО необходимо:

1. Перед удалением убедитесь, что на сервере установлены компоненты ПО ПАК, выполнив команду:

   ```
   rpm -q -a | grep tccs
   ```

2. Если ПО в системе установлено, выполните команды для запуска скрипта, удаляющего дистрибутив:

   ```
   cd /home/USER/rpmdist*
   ./jinn_uninstall.sh
   ```

   После запуска будут выведены варианты обновления:

   ```
   [root@cpro-4 rpmdist-tccs-2.0.0-1-centos-7_1_1503-x86_64]# ./jinn_uninstall.sh
   Выберите вариант удаления:
           1 - Удалить сервис архивирования CAS1;
           2 - Удалить сервисы подписи и проверки и усиления подписи;
           3 - Удалить сервис архивирования CAS2;
           4 - Удалить АРМ РКС;
           0 - Выход
   ```

3. Введите цифру, соответствующую нужному варианту удаления, и нажмите клавишу <Enter>.

   В процессе удаления будет предложено сохранить все прикладные данные в системе, и на консоли появится следующее сообщение:

   ```
   Вы хотите удалить все прикладные данные?
   ```

   В случае сохранения прикладных данных на консоль будет выведено сообщение о месте расположения архивов с данными на файловой системе.

Глава 4
Настройка программного обеспечения ПАК "Jinn-Server"

Начальная настройка программного обеспечения, входящего в ПАК "Jinn-Server", выполняется автоматически в процессе его установки. Установка программного обеспечения описана в предыдущей главе настоящего документа.

В данной главе приведены действия администратора ПАК "Jinn-Server", которые должны быть выполнены вручную после установки для обеспечения корректного взаимодействия компонентов ПАК, его отказоустойчивой работы, в том числе действия по настройке компонентов ПАК "Jinn-Server", установленного на аппаратную платформу АПКШ "Континент".

Конфигурирование сервисов ПАК "Jinn-Server"

Сервис проверки ЭП (SVS)

Настройка сервера Apache

Конфигурационный файл сервера (/opt/tccs/web-services/conf/httpd.conf) уже содержит требуемые настройки. Управление сервером осуществляется командой:

```
/opt/tccs/etc/init.d/tccs.ws start|stop|restart|status
```

Настройка веб-сервисов

Для настройки веб-сервисов используются следующие конфигурационные файлы:

- /opt/tccs/web-services/services/SignatureValidationService/services.xml — настройки axis2/c веб-сервисов;
- /opt/tccs/web-services/services/SigningService/services.xml — настройки axis2/c веб-сервисов;
- /opt/tccs/oss/axis.xml — общие настройки окружения axis2/c.

Элементы управления

Управление прикладным сервисом осуществляется с помощью следующего сценария: /opt/tccs/etc/init.d/tccs.ws start|stop|restart|status.

Конфигурационный файл сервиса SVS – /opt/tccs/etc/tccs.svs.

Пример содержимого файла /opt/tccs/etc/tccs.svs:

```
{
"cspProvName":"Crypto-Pro GOST R 34.10-2001 KC2 CSP",
"cspProvType":75,
"svcUserName":"tccs",
"casAddress":"tcp://tccs1.domain.ru:11112",
"signingFQCNList":"*",
"tsaFQCNList":"*",
"svcAddress":"file:///var/opt/tccs/run/tccs.svs",
"pfdStartWorkers":10,
"pfdReadyWorkers":10,
"pfdMaxWorkers":100,
"pfdWorkerRequests":200,
"pfdWorkerMaxTTL":600,
```

```
"pfdBusyMillisec":200,
"pfdBusyToDead":2000,
"xsltCacheTTLSeconds":20,
"xsltCacheUpdateTimeout":2,
"x509ForceAKIMatch":1
}
```

Полный список параметров, указание которых возможно в данном конфигурационном файле, описан ниже. Перечисленные параметры также могут быть использованы в конфигурационных файлах других компонентов. Параметры, не имеющие смысла для конкретного компонента, игнорируются. Рекомендации по выбору значений для некоторых параметров, а также рекомендации по настройке ОС для использования с ПАК "Jinn-Server" приведены в главе 7.

cspProvName – имя криптопровайдера в терминах Microsoft CryptoAPI, который должен использоваться сервисом. Значение по умолчанию: "Crypto-Pro GOST R 34.10-2001 KC2 CSP".

cspProvType – тип криптопровайдера в терминах Microsoft CryptoAPI, который должен использоваться сервисом. Значение по умолчанию: 75.

casAddress – адрес внутреннего компонента сборщика CRL (CAS-1). Значение по умолчанию не задано.

caListRefreshSeconds – продолжительность жизни отдельной записи в кеше сертификатов издателей в секундах. Значение по умолчанию: 600.

noCRLTimeout – периодичность (в секундах) обращения сервиса в CAS-1 с запросом на получение CRL, в случае если CRL для заданного издателя отсутствует в кеше сервиса. Значение по умолчанию: 10.

crlRefreshSeconds – периодичность (в секундах) обращения сервиса в CAS-1 с запросом на получение CRL, в случае если CRL для данного издателя уже присутствует в кеше сервиса. Значение по умолчанию: 20.

signingFQCN – FQCN (Fully Qualified Container Name) для идентификации контейнера с ключом и сертификатом, который должен использоваться сервисом для подписания данных, либо маска по умолчанию -"*", либо конкретная маска. При указании маски для подписания данных будет использоваться первый попавшийся контейнер, содержащий ключ и сертификат, подходящий для данной задачи (определяется на основании статуса и расширений сертификата).

tsaFQCN – аналогично signingFQCN, определяет контейнер для подписания TSP-квитанций. Значение по умолчанию: "*".

svcAddress – полный путь к сокету на файловой системе, который будет использоваться сервисом.

svcTimeoutSec – тайм-аут в секундах, определяющий максимально допустимое время на получение WS-frontend'ом ответа от прикладного сервиса. Значение по умолчанию: 5.

svcUserName – имя системного пользователя, с полномочиями которого должен быть запущен сервис. Должно соответствовать имени владельца файлов, составляющих программный комплекс. Значение задается автоматически в процессе установки: "tccs".

pfdStartWorkers – начальное число рабочих процессов, которое будет создано сразу после старта сервиса.

pfdReadyWorkers – число рабочих процессов, находящихся в состоянии готовности к обработке запросов. В случае если часть рабочих процессов занята либо завершилась, для обеспечения оперативной обработки вновь поступающих запросов процесс-супервизор будет запускать новые рабочие процессы до достижения заданного значения.

pfdMaxWorkers – максимальное число рабочих процессов, которые могут быть запущены одновременно. Реализация прикладных сервисов допускает фактическое превышение указанного значения в два раза.

pfdWorkerRequests – максимальное число обрабатываемых запросов на один рабочий процесс.

pfdWorkerMaxTTL – максимальная продолжительность жизни рабочего процесса в секундах.

pfdBusyMillisec – период (в миллисекундах), в течение которого рабочий процесс должен регулярно уведомлять процесс-супервизор о своей готовности к обработке очередного запроса. В отсутствие уведомления в заданный период рабочий процесс отмечается как занятый. Значение по умолчанию: 200.

pfdBusyToDead – период (в миллисекундах), по истечении которого занятый рабочий процесс обязан завершить обработку запроса и уведомить процесс-супервизор о своей готовности к обработке очередного запроса. В отсутствие уведомления в заданный период рабочий процесс отмечается как сбойный и подлежит принудительному завершению. Значение по умолчанию: 2000.

pfdAllowSocketReuse – определяет возможность обработки нескольких запросов в рамках одного соединения между веб-сервером и прикладным сервисом. Значение по умолчанию: 0. В случае если указано не нулевое значение, прикладной сервис, обработав запрос, оставляет соединение открытым и может обрабатывать приходящие через него запросы до наступления одного из следующих событий:

- соединение закрыто клиентом;
- успешно установлено новое соединение.

asn1AllowAnyBroken – определяет допустимость наличия некоторых ошибок в кодировании обрабатываемых данных. Значения по умолчанию: "0" – ошибки не допускаются; "1" – ошибки допускаются. Допустимыми считаются ошибки в любых элементах данных, за исключением перечисленных в описании параметра.

asn1MustBeValid – необязательный параметр, допускается многократное включение в конфигурационный файл. Значение параметра содержит описание элементов ASN.1 кодированных данных, в которых запрещены ошибки или отклонения от стандартов. Параметр обеспечивает возможность расширения зафиксированного в исходном коде КС списка "обязательно корректных" элементов. Тип значения данного параметра – строка следующего формата:

```
NameChain ::= Name [ '::' NameChain ]
Name ::= <asn1TypeName> | <asn1MemberName> | 'ANY'
```

В случае если последним элементом в значении параметра является имя определенного ASN.1-типа либо имя элемента одного из составных ASN.1-типов, такое описание означает запрет на ошибки в кодировании непосредственно указанного элемента. Если указанный элемент имеет составной тип, то ошибки в тегах его дочерних элементов также не допускаются. Однако в зависимости от других правил ошибки в кодировании дочерних элементов более глубокого уровня вложенности могут оказаться допустимыми.

В случае если последним элементом в значении параметра является 'ANY', ошибки не допускаются для всех дочерних элементов, вне зависимости от глубины вложенности.

Элемент ASN.1-кодированных данных, соответствующий первому элементу значения параметра, не обязан быть самым внешним при декодировании, т.е. правило, начинающееся с элемента 'Certificate', одинаково запрещает ошибки как при разборе отдельных сертификатов, так и при их разборе в составе подписанных данных.

В случае нахождения такого рода ошибки в кодировании данных тип ошибочного элемента заменяется на ANY-BROKEN и может встретиться в отдельных элементах ответа сервиса.

По умолчанию ошибки запрещены в соответствии с перечисленными ниже правилами. Указанные правила не могут быть изменены, так как зафиксированы в исходном коде и обеспечивают корректность данных, критичных для работоспособности КС:

```
AuthorityKeyIdentifier::OCTET_STRING
AuthorityKeyIdentifier
CRLReason
CertificateIssuer
```

```
CRLNumber
KeyUsage
ExtKeyUsage::ANY
BasicConstraints::ANY
CertificatePolicies::ANY
IssuerSignTool::ANY
SubjectSignTool::ANY
CRLDistributionPoints::ANY
AlgorithmIdentifier::ANY
SigningCertificateV2::ANY
Name
Certificate::TBSCertificate::Validity::ANY
Certificate::TBSCertificate::Extensions
Certificate::TBSCertificate::SubjectPublicKeyInfo::ANY
Certificate::TBSCertificate
Certificate
CertificateList::TBSCertList::Time
CertificateList::TBSCertList::CRLEntries::CRLEntry::Extensions
CertificateList::TBSCertList::CRLEntries::CRLEntry
CertificateList::TBSCertList::CRLEntries
CertificateList::TBSCertList::Extensions
CertificateList::TBSCertList
CertificateList
TSTInfo
ContentInfo::SignedData::SignerInfos::SignerInfo
ContentInfo::SignedData::SignerInfos
ContentInfo::SignedData
ContentInfo
ContentType
SigningTime::ANY
SubjectKeyIdentifier
messageDigest
DeltaCRLIndicator
FreshestCRL::ANY
IssuingDistributionPoint
```

X509ForceAKIMatch – определяет необходимость проверки расширения AuthorityKeyIdentifier при проверке цепочки сертификатов. Допустимые значения: 0 – не проверять, 1 – обязательно проверять. Значение по умолчанию: 0.

xsltCacheTTLSeconds – период в секундах, в течение которого записи в XSLT-кеше считаются актуальными (по истечении периода данные будут обновлены по соответствующей ссылке). Значение по умолчанию: 600. Значение 0 полностью отключает кеширование xslt.

xsltCacheUpdateTimeout – период времени в секундах, в течение которого рабочие процессы будут ожидать обновления записи в XSLT-кеше. Такое ожидание может состояться только при первом обращении по конкретной ссылке. В противном случае рабочие процессы не будут ожидать окончания обновления

заблокированной записи, а вместо этого используют предыдущую версию записи. Значение по умолчанию: 5.

tsl_issuer_noverify – управляет процедурой проверки сертификата автора TSL. Значение по умолчанию: "1" (либо "yes"). Данное значение подразумевает безусловное доверие сертификату автора подписи под TSL. В случае указания значения "0" (либо "no") процедура загрузки TSL в CAS-1 будет сопровождаться полноценной проверкой подписи под TSL, включающей построение и проверку цепочки сертификатов с контролем статуса по CRL для сертификата автора подписи.

trust_mode – устанавливает один из трех режимов доверия к издателям:

- 0 – доверенными считаются все зарегистрированные издатели (значение по умолчанию);
- 1 – доверенными считаются издатели, специально отмеченные в CAS-1;
- 2 – доверенными считаются все самоподписанные сертификаты издателей.

При выборе режима доверия необходимо учитывать особенности PKI-окружения, в котором эксплуатируется ПК.

Сервис формирования ЭП (SS)

Настройка сервера Apache

Конфигурационный файл сервера (/opt/tccs/web-services/conf/httpd.conf) уже содержит требуемые настройки. Управление сервером осуществляется так:

```
/opt/tccs/etc/init.d/tccs.ws start|stop|restart|status
```

Настройка веб-сервисов

Для настройки веб-сервисов используются следующие конфигурационные файлы:

```
/opt/tccs/web-services/services/SignatureValidationService/services.xml - настройки axis2/c веб-сервисов
/opt/tccs/web-services/services/SigningService/services.xml – настройки axis2/c веб-сервисов
/opt/tccs/oss/axis.xml – общие настройки окружения axis2/c
```

Элементы управления

Управление прикладным сервисом осуществляется с помощью следующего сценария: /opt/tccs/etc/init.d/tccs.ss start|stop|restart|status.

Конфигурационный файл сервиса SS – /opt/tccs/etc/tccs.ss.

Содержимое файла /opt/tccs/etc/tccs.ss:

```
{
"cspProvName":"Crypto-Pro GOST R 34.10-2001 KC2 CSP",
"cspProvType":75,
"svcUserName":"tccs",
"casAddress":"tcp://tccs1.domain.ru:11112",
"signingFQCNList":"*",
"tsaFQCNList":"*",
"svcAddress":"file:///var/opt/tccs/run/tccs.ss",
"pfdStartWorkers":10,
"pfdReadyWorkers":10,
"pfdMaxWorkers":80,
```

```
"pfdWorkerRequests":200,
"pfdWorkerMaxTTL":600,
"pfdBusyMillisec":200,
"pfdBusyToDead":2000
}
```

Рекомендации по выбору значений для некоторых параметров, а также рекомендации по настройке ОС для использования с ПАК "Jinn-Server" приведены в главе 7.

Сервис архивирования CRL (CAS-1)

Настройка ODBC

Конфигурационный файл (/etc/odbc.ini) определяет драйвер в соответствии с используемой СУБД (уточняется на этапе согласования поставки), режимы работы с СУБД и место расположения сервера СУБД. Примерный вид конфигурационного файла приведен ниже:

```
[ODBC Data Sources]
csm = Postgres
[csm]
Driver=/opt/tccs/oss/lib/psqlodbcw.so
Description=csm Postgres DB
Servername=localhost
UserName=postgres
Database=csm
ReadOnly=no
TRANSLATIONDLL=
```

Настройка сервера Apache

Конфигурационный файл сервера (/opt/tccs/web-admin/conf/httpd.conf) уже содержит требуемые настройки. Управление сервером осуществляется так:

```
/opt/tccs/etc/init.d/tccs.admin
start|stop|restart|status
```

Элементы управления

Управление прикладными сервисами, входящими в состав CAS-1, осуществляется с помощью следующих сценариев:

```
/opt/tccs/etc/init.d/psql1d start|stop|restart|status
/opt/tccs/etc/init.d/cas1d start|stop|restart|status
```

Запуск СУБД Postgres должен осуществляться до запуска cas1d.

Основным элементом управления сервиса CAS-1 является конфигурационный файл /opt/tccs/etc/csm.conf, в котором указаны параметры для работы процессов сервиса.

Конфигурационный файл CAS-1 содержит:

- "crl_daemon_network" – обязательный блок параметров, определяющих адрес, по которому внешним потребителям будет доступен сервис компонента CAS-1. Параметры, входящие в данный блок, имеют названия, в точности отражающие их назначение: hostname и port.
- "crl_daemon_network_2" – обязательный блок параметров, определяющих адрес, по которому внешним потребителям будет доступен сервис компонента CAS-2. Параметры, входящие в данный блок, имеют названия, в точности отражающие их назначение: hostname и port.

- Перечисление (набор блоков) "watchedservice" – по одному блоку для каждого из сервисов, работоспособность которых должна контролироваться/поддерживаться. Каждый из таких блоков содержит по пять элементов: помимо очевидных hostname, port и description, есть также socket – для сервисов, обращение к которым производится через UNIX-socket, и action – команда, которая будет запущена с аргументов "start", в случае если описанный в данном блоке сервис не принимает соединение.
- Набор параметров, регулирующих загрузку СОС (CRL): "crl_get_time", "crl_get_period", "crl_allow_period", "delta_get_period". Назначение отдельных полей в этом наборе – см. Приложение.
- "notification_enable", "notification_email": параметры, управляющие уведомлениями для администратора – см. раздел главы 2 "Уведомления о нештатных событиях в системе".
- Необязательный параметр, определяющий уровень соответствия стандартам, которого будет придерживаться данный компонент в процессе разбора входных данных: "asn1AllowAnyBroken".
- crl_get_period_force – периодичность в секундах, с которой принудительно производятся попытки получить актуальный CRL, вне зависимости от значений параметров CRL_GET_TIME и CRL_GET_PERIOD, заданных в профиле издателя. Значение по умолчанию: 0, сохраняет штатный режим получения актуальных CRL.
- caTrustedVerify – определяет порядок формирования ответов на запросы пользователей на получение списка доверенных либо недоверенных издателей. При указании значения 1 к доверенным будут отнесены только специально отмеченные администратором издатели. При указании значения 0 (значение по умолчанию) к доверенным будут отнесены все зарегистрированные издатели.
- caRegManualDeny – управляет доступностью ручной регистрации издателей. Значение 1 допускает только регистрацию издателей, перечисленных в TSL. Значение 0 (значение по умолчанию) в дополнение к регистрации издателей из TSL позволяет администратору зарегистрировать произвольного издателя.

Пример файла /opt/tccs/etc/csm.conf:

```
{
"crl_daemon_network": [
   { "hostname": "tccs1.domain.ru" },
   { "port": 11112 }
],
"crl_daemon_network_2": [
   { "hostname": "tccs2.domain.ru" },
   { "port": 11113 }
],
"watchedservice": [
   { "hostname": "tccs1.domain.ru" },
   { "port": 80 },
   { "socket": "" },
   { "action": "/opt/tccs/etc/init.d/tccs.admin" },
   { "description": "CSM WEB ADMIN" }
],
"watchedservice": [
   { "hostname": "tccs1.domain.ru" },
   { "port": 11112 },
   { "socket": "" },
```

```
   { "action": "/opt/tccs/etc/init.d/cas1d" },
   { "description": "CRL Archive Daemon" }
],
"watchedservice": [
   { "hostname": "tccs1.domain.ru" },
   { "port": 0 },
   { "socket": "/tmp/.s.PGSQL.5432" },
   { "action": "/opt/tccs/etc/init.d/psq1d" },
   { "description": "PostgreSQL" }
],
"crl_get_time": 21600,
"crl_get_period": 900,
"crl_allow_period": 0,
"delta_get_period": 600,

"asn1AllowAnyBroken": 0,

"notification_enable": "yes",
"notification_email": "admin@domain.ru"
}
```

Исходная информация для проверки целостности и авторства доверенного списка издателей (TSL)

Для проверки загружаемых TSL предусмотрена возможность установки и замены самоподписанного сертификата ГУЦ, сертификатов промежуточных издателей, а также актуальных списков отозванных сертификатов и сертификата автора подписи под TLS.

Данная возможность реализуется посредством двух механизмов:

1. Веб-интерфейс, аналогичный интерфейсу по управлению издателями пользовательских сертификатов. Для загрузки и доставки в CAS-1 актуальных COC также будет использоваться существующий механизм – отчуждаемый USB-флеш-накопитель, перемещаемый между CAS-1 и CAS-2. Никакой дополнительной подготовки носителя не требуется.
2. Ручная установка необходимого набора файлов (сертификатов и COC) в CAS-1. Соответствующие сертификаты и CRL необходимо привести к формату PEM и разместить в одном или нескольких файлах в папке /var/opt/tccs/tsl/, убедившись, что все созданные таким образом файлы доступны на запись только администратору, а на чтение доступны всем. Данный механизм доставки сертификатов и CRL для проверки TSL является резервным. Установленные таким образом сертификаты и CRL игнорируются при наличии сертификатов, зарегистрированных через механизм №1. Автоматическая загрузка актуальных CRL для установленных таким образом издателей невозможна.

Примечание. В случае если ГУЦ либо промежуточные издатели по тем или иным причинам не в состоянии своевременно выпустить корректный COC и НЕ признают такую ситуацию аварийной, для сохранения работоспособности системы допускается временное отключение проверки сертификата автора TSL. Управление проверкой сертификата автора TSL осуществляется с помощью boolean параметра tsl_issuer_noverify в конфигурационном файле csm.conf.

- "tsl_issuer_noverify": "yes" – данное значение отключает проверку сертификата автора TSL;
- "tsl_issuer_noverify": "no" – значение по умолчанию – при загрузке TSL будут выполняться построение и проверка всех сертификатов из цепочки сертификатов автора TSL.

Настройка регулярно запускаемых задач

В ПК используется штатный механизм управления регулярным (по расписанию) запуском программ – /etc/crontable. Содержание указанного файла в отношении функциональности компонента приведено ниже:

```
*/1 * * * * root /opt/tccs/bin/cas1_crl_update
>/dev/null
*/1 * * * * root /opt/tccs/bin/cas1_watchdog >/dev/null
#* */1 * * * root /opt/tccs/bin/cas1_cas2_client
>/dev/null 2>/dev/null
```

Если удалить символ комментария "#", то будет возможно использовать сетевой TCP онлайн-обмен между CAS-1–CAS-2.

Используемые сценарии и процедуры:

- /opt/tccs/bin/cas1_watchdog – процедура автоматического контроля активности процессов компонента CAS-1.
- /opt/tccs/bin/cas1_crl_update – процедура загрузок CRL/deltaCRL в хранилище базы данных CAS-1, находящихся в директории /var/opt/tccs/update/.
- /opt/tccs/bin/cas1_cas2_client – процедура обращения к демону CAS-2 при онлайн-режиме транспортировки.

Сервис архивирования CRL (CAS-2)

Настройка ODBC

Конфигурационный файл (/etc/odbc.ini) определяет драйвер в соответствии с используемой СУБД (уточняется на этапе согласования поставки), режимы работы с СУБД и место расположения сервера СУБД. Примерный вид конфигурационного файла приведен ниже:

```
[ODBC Data Sources]
csm = Postgres
[csm]
Driver=/opt/tccs/oss/lib/psqlodbcw.so
Description=csm Postgres DB
Servername=localhost
UserName=postgres
Database=csm
ReadOnly=no
TRANSLATIONDLL=
```

Элементы управления

Основным элементом управления сервисом CAS-1 является конфигурационный файл /opt/tccs/etc/csm.conf, в котором указаны параметры (аналогичные описания из CAS-1) для работы процессов сервиса.

Содержимое файла /opt/tccs/etc/csm.conf:

```
{
"crl_daemon_network": [
     { "hostname": "tccs1.domain.ru" },
     { "port": 11112 }
],
"crl_daemon_network_2": [
     { "hostname": "tccs2.domain.ru" },
     { "port": 11113 }
```

```
],
"watchedservice": [
    { "hostname": "tccs2.domain.ru" },
    { "port": 0 },
    { "socket": "/tmp/.s.PGSQL.5432" },
    { "action": "/opt/tccs/etc/init.d/psqld" },
    { "description": "PostgreSQL" }
],
"crl_get_time": 21600,
"crl_get_period": 900,
"crl_allow_period": 0,
"delta_get_period": 600,

"asn1AllowAnyBroken": 0,

"notification_enable": "yes",
"notification_email": "admin@domain.ru"
}
```

Настройка cron-таблицы

Механизм управления квантованным по времени запуском программ определяется в описании специальной таблицы – crontable, расположенной в директории /etc. Содержание указанного файла в отношении функциональности компонента приведено ниже:

```
*/1 * * * * root /opt/tccs/bin/cas2_crl_get
>/var/opt/tccs/log/crl_ca_get.tmp
2>/var/opt/tccs/log/crl_ca_get.tmp && cat
/var/opt/tccs/log/crl_ca_get.tmp
>>/var/opt/tccs/log/crl_ca_get

*/1 * * * * root /opt/tccs/bin/cas2_crl_tsl_get
>/var/opt/tccs/log/crl_tsl_get.tmp
2>/var/opt/tccs/log/crl_tsl_get.tmp && cat
/var/opt/tccs/log/crl_tsl_get.tmp
>>/var/opt/tccs/log/crl_tsl_get

1 3 * * * root /opt/tccs/bin/cas2_tsl_get
>/var/opt/tccs/log/tsl_get.tmp
2>/var/opt/tccs/log/tsl_get.tmp && cat
/var/opt/tccs/log/tsl_get.tmp
>>/var/opt/tccs/log/tsl_get

*/1 * * * * root /opt/tccs/bin/cas2_watchdog >/dev/null
#* */1 * * * root /opt/tccs/bin/cas2_cas1_client
>/dev/null 2>/dev/null

Если удалить символ комментария «#», то будет возможно
использовать сетевой TCP on-line обмен между cas2—cas1.
```

Используемые сценарии и процедуры:

- /opt/tccs/bin/cas2_watchdog – процедура автоматического контроля активности процессов компонента CAS-2.
- /opt/tccs/bin/cas2_crl_get – процедура получения CRL/deltaCRL из источников CRLDistributionPoint / FreshestCRL и размещение их в директории /var/opt/tccs/cache/.
- /opt/tccs/bin/cas2_tsl_get – процедура получения списка TSL.
- /opt/tccs/bin/cas2_crl_tsl_get – процедура получения CRL для всех сертификатов из цепочки сертификатов автора списка TSL.
- /opt/tccs/bin/cas2_cas1_client – процедура обращения к демону CAS-1 при онлайн-режиме транспортировки.

Оптимизация производительности

Реализация сервисов проверки и формирования подписи в ПАК "Jinn-Server" содержит два слоя:

- транспортный слой, обеспечивающий доступ к функциональности через механизм веб-сервисов;
- логический (бизнес) слой, непосредственно осуществляющий обработку данных.

Бизнес-слой реализован в виде прикладных сервисов (демонов) **tccs.ss** и **tccs.svs**. Данные сервисы работают по комбинированной схеме: родительский процесс (супервизор) отвечает за инициализацию и корректный останов сервиса, а также обеспечивает наличие необходимого количества рабочих процессов, каждый из которых до своего завершения может обработать заданное количество запросов.

Количество рабочих процессов и другие характеристики прикладного сервиса определяются следующими параметрами конфигурационных файлов прикладных сервисов (/opt/tccs/etc/tccs.ss и /opt/tccs/etc/tccs.svs):

- **"pfdStartWorkers":6** – начальное число рабочих процессов, которое будет создано сразу после старта сервиса. Данный параметр не должен превышать значение параметра pfdMaxWorkers. Рекомендуемое значение (0.3 * pfdMaxWorkers).
- **"pfdReadyWorkers":6** – число рабочих процессов, находящихся в состоянии готовности к обработке запросов. В случае если часть рабочих процессов занята либо завершилась, для обеспечения оперативной обработки вновь поступающих запросов процесс-супервизор будет запускать новые рабочие процессы до достижения заданного значения. При этом общее число запущенных рабочих процессов не будет превышать значения параметра pfdMaxWorkers. Рекомендуемый диапазон значений:

```
min: tStart * maxRequestsPerSecond / pfdWorerRequests
max: 0.3 * pfdMaxWorkers
```

  где:

- **tStart** – время, необходимое для старта и инициализации рабочего процесса, определяется экспериментально для каждого конкретного экземпляра ПАК на основании диагностического вывода (syslog).
- **maxRequestsPerSecond** – максимальная расчетная нагрузка на сервис (запросов в секунду).
- **"pfdMaxWorkers":22** – максимальное число рабочих процессов, которые могут быть запущены одновременно. Реализация прикладных сервисов допускает фактическое превышение указанного значения в два раза. Рекомендуемое значение – минимальное из двух:
  - количество независимых потоков исполнения, которое обеспечивает аппаратная платформа (число физических процессоров * число физических ядер в процессоре);
  - (0.3 * объем физически доступной оперативной памяти)/(20 * средний размер запроса).

- **"pfdWorkerRequests":1000** – максимальное число запросов, которое может обработать рабочий процесс до завершения. Если процесс-супервизор не успевает запускать новые рабочие процессы взамен выбывающих (текущее число рабочих процессов, готовых к обработке запросов, меньше чем pfdReadyWorkers), реализация прикладных сервисов допускает превышение данного параметра в три раза. Рекомендуемый диапазон значений: 500–5000. Оптимальное значение можно определить только на основании стресс-тестирования конкретного экземпляра ПАК.
- **"pfdWorkerMaxTTL":600** – максимальная "продолжительность жизни" рабочего процесса в секундах. По истечении указанного времени соответствующий рабочий процесс будет заменен. Параметр может быть превышен аналогично параметру pfdWorkerRequests.
- **"pfdBusyToDead":2000** – максимально допустимая продолжительность обработки запроса рабочим процессом в миллисекундах. Если по истечении заданного времени рабочий процесс не сообщит о своей готовности к обработке следующего запроса, данный рабочий процесс будет принудительно завершен процессом-супервизором. Рекомендуемое значение – удвоенное время обработки запроса максимально допустимого размера (определяется экспериментально для каждого конкретного экземпляра ПАК).
- **"svcTimeoutSec": 5** – тайм-аут в секундах, определяющий максимально допустимое время на получение WS-frontend'ом ответа от прикладного сервиса. Рекомендуемый диапазон значений – не менее pfdBusyToDead/1000.

В случае если реальная производительность оборудования не соответствует ожидаемой, рассчитанной по методике сайзинга, следует обратить внимание на то, в каком режиме работает процессор. Некоторые платформы могут вводить процессор в режим энергосбережения и тем самым снижать рабочую частоту процессора. Для того чтобы этого не происходило, следует отключить данный режим в настройках BIOS, в параметрах ядра (/boog/grub2/grub.conf, а точнее соответствующем исходнике в /etc) следует добавить processor.max_cstate=1 idle=poll, в tccs.ss & tccs.svs значение pfdReadyWorkers увеличить так, чтобы оно несколько превышало число потоков исполнения.

Транспортный слой реализован с помощью связки из веб-сервера Apache httpd, модуля mod_axis и разделяемых библиотек, содержащих интерфейсы веб-сервисов, описанные в файлах SigningService.wsdl и SignatureValidationService.wsdl.

Файл настроек веб-сервера Apache httpd, используемого для транспорта, – /opt/tccs/web-services/conf/httpd.conf. По умолчанию используется многопоточный вариант запуска веб-сервера.

Ключевыми параметрами, влияющими на производительность и надежность системы, являются:

- **StartServers 16** – количество запускаемых рабочих процессов. Значение вычисляется на основании других параметров: MaxClients / ThreadsPerChild.
- **MaxClients 256** – максимальное количество одновременно обслуживаемых клиентов. Определяется по результатам стресс-тестирования.

Рекомендации по обеспечению отказоустойчивости ОС

Ключевыми ресурсами, которые могут оказаться "узким местом" в процессе эксплуатации ПАК "Jinn-Server", являются:

- CPU.
- Оперативная память.
- Производительность и надежность дисковой подсистемы.
- Пропускная способность и надежность сетевой подсистемы.

Компоненты ПАК "Jinn-Server" сами по себе не предъявляют существенных требований к пропускной способности дисковой подсистемы. Однако диагностический вывод (syslog), как правило, очень требователен к данному параметру. Для исключения влияния диагностического вывода на общую производительность системы рекомендуется вместо штатного syslog использовать syslog-ng, минимизировать объем локально хранимого диагностического вывода, а основной его объем направлять по сети на специально выделенную для этой цели машину.

Пропускная способность и надежность сетевой подсистемы может быть обеспечена дублированием сетевых интерфейсов (interface bonding). Если аппаратная платформа допускает такую возможность, то для описанного выше сетевого перенаправления диагностического вывода желательно использовать выделенный интерфейс.

Эффективному распределению ресурсов CPU способствует использование балансировки прерываний. Автоматическая настройка балансировки реализуется сервисом irqbalance daemon. В большинстве случаев автоматической настройки достаточно для стабильной и эффективной работы ОС. Текущая статистика по прерываниям содержится в выводе cat /proc/interrupts.

Ручная настройка балансировки прерываний (распределение конкретного прерывания одному или нескольким ядрам) может потребоваться в случае, если автоматическая балансировка не дает равномерной загрузки имеющихся ядер.

Для ручной балансировки прерываний запустите команду:

```
echo N > /proc/irq/X/smp_affinity
```

где N – маска процессора (определяет, какие ядра могут обрабатывать прерывание), а X – номер прерывания, виден в первом столбце вывода cat /proc/interrupts. Чтобы определить маску процессора, нужно возвести 2 в степень cpu_N (номер процессора) и перевести в шестнадцатеричную систему.

Ограничения в настройках прикладных сервисов и Apache httpd, связанные с оперативной памятью, описаны выше. Для того чтобы исключить нестабильное поведение системы в случае превышения расчетной нагрузки, рекомендуется отключить в ОС возможность выделения приложениям большего объема памяти, чем реально доступно (с учетом swap).

Для этого необходимо:

- временное отключение:

  ```
  echo 2 > /proc/sys/vm/overcommit_memory
  ```

- постоянная настройка:

  ```
  echo "vm.overcommit_memory = 2" >> /etc/sysctl.conf
  ```

- перезапуск ОС для вступления в силу внесенных изменений.

Глава 5
Рекомендации по анализу статистики работы веб-сервера

Задача по обработке и анализу статистики работы веб-сервера может потребовать существенных ресурсов. Для ее выполнения необходимо следовать приведенным выше рекомендациям по перенаправлению диагностического вывода от компонентов ПАК на специально выделенную для этих целей физическую машину.

Для минимизации объема статистики с одновременным увеличением ее информативности с точки зрения контроля загрузки и производительности ПАК формат записей для файла access_log был изменен (по отношению к формату по умолчанию) следующим образом:

- удалена информация о клиентском приложении (браузере);
- удалена информация об источнике запроса (referer);
- добавлена информация о времени обработки запроса;
- добавлена дата в формате unix-time.

Для обработки собираемых данных рекомендуется использовать свободно распространяемые анализаторы, имеющие возможность настройки формата входных данных. К таким анализаторам относятся в том числе:

```
Analog – http://www.analog.cx/
GoAccess – http://goaccess.prosoftcorp.com/
```

Указанные анализаторы, как и большинство других, принимают на вход целиком файл с данными о работе веб-сервера и формируют отчет в виде html. Поддержка отчетов в других форматах в том или ином виде присутствует в большинстве анализаторов.

Размер файлов с данными о работе веб-сервера будет расти довольно быстро (600 Кб/с при нагрузке 5000 запросов в секунду). Ориентировочная производительность анализаторов – менее 150000 записей в секунду.

Ограничить объем входных данных для анализа можно одним из следующих подходов (либо их одновременным использованием):

- Ограничить максимальный объем входных данных анализатора настройками ротации файлов с данными о работе веб-сервера.
- Минимизировать объем входных данных для анализатора за счет их предварительной фильтрации.

Средства для управления ротацией файлов с данными о работе веб-сервера и других компонентов ПАК встроены в ОС Linux (logrotate).

Для обеспечения возможности предварительной фильтрации данных в состав ПАК включен сценарий /opt/tccs/bin/backlog.sh. Данный сценарий предназначен для извлечения истории работы веб-сервера на заданную пользователем в секундах глубину от текущей даты. Ориентировочная производительность данного фильтра >700000 записей в секунду.

Кроме того, в состав ПАК включен демонстрационный сценарий /opt/tccs/bin/urlstat.sh – пример простого подсчета статистики обращений к заданному URL. Результатом работы сценария является текстовый вывод, содержащий следующие данные:

- общее число запросов;
- число запросов с результатом обработки "200";
- средний размер ответа в байтах;
- среднее время обработки в микросекундах.

Пример вывода данного сценария:

```
total: 4718595
status 200: 4718595
response bytes avg.: 402
bytes per second avg.: 945600
processing microseconds avg.: 545
```

Ориентировочная производительность данного сценария >200000 записей в секунду.

Глава 6
Общие указания по порядку технического обслуживания

ПАК "Jinn-Server" специальных требований и порядка технического обслуживания не предполагает, кроме процедур и мероприятий, отраженных в сопроводительной документации на используемые аппаратные платформы и программные компоненты, входящие в ПАК.

Глава 7
Подготовка и проверка вспомогательных данных для работы ПАК "Jinn-Server"

В данной главе описано, как подготовить необходимые ресурсы или проверить правильность настроек для работы ПАК "Jinn-Server" на компьютере с "Alt Linux SPT 6.0" или "CentOS-7.0-1406".

Создание криптоконтейнеров для ПАК "Jinn-Server"

Создание криптоконтейнеров и запроса на сертификат

Создавать криптоконтейнеры и запрос на сертификат нужно на отдельном, стороннем компьютере, на котором установлена утилита **cryptcp** из состава "КриптоПро CSP". Подробнее об этих действиях см. в п. 2.7 документа https://www.cryptopro.ru/sites/default/files/products/cryptcp/3-40/cryptcp.pdf.

Следует создать 2 контейнера со следующими Extended Key Usage (EKU):

- сертификат ключа подписи для сервиса SS – не содержащий EKU "1.3.6.1.5.5.7.3.8";
- сертификат ключа подписи для сервиса SVS – EKU "1.3.6.1.5.5.7.3.8" /* Сертификат X 509 может использоваться для включения значения хэш-функции при создании штампа времени на документы в Службе штампов времени */.

Для запуска сервиса формирования подписи (SS) необходимы оба сертификата.

Для запуска сервиса проверки и усиления достаточно сертификата с EKU "1.3.6.1.5.5.7.3.8".

После создания запроса на сертификат требуется его выпустить. Для этого откройте в браузере ссылку http://www.cryptopro.ru/certsrv/certrqxt.asp на тестовый удостоверяющий центр "КриптоПро CSP" (или организуйте собственный УЦ).

В появившейся форме вставьте в поле "Base-64-шифрованный запрос сертификата" содержимое, предварительно скопированное из созданного файла запроса *.csr, и нажмите кнопку "Выдать". Затем сохраните файл по ссылке "Загрузить сертификат" (имя по умолчанию certnew.cer). Предлагается сохранение в der и base64 кодировке.

Добавление выданного сертификата в контейнер

Выданные сертификаты добавляются в контейнер с помощью "КриптоПро CSP" (см. http://www.cryptopro.ru/sites/default/files/docs/csp36r3/instruction_csp_r3.pdf, рис. 53).

Копирование готовых контейнеров на ПАК "Jinn-Server"

Созданные контейнеры нужно скопировать в каталог /var/opt/cprocsp/keys/tccs, размещающийся на ПАК "Jinn-Server".

Настройка прав доступа к криптоконтейнерам и лицензиям "КриптоПро CSP"

Добавление лицензии "КриптоПро CSP":

1. Проверьте срок действия лицензии, выполнив команду:

```
cpconfig -license –view
```

2. Установите лицензию с помощью команды:

```
cpconfig -license -set <ваша_лицензия>
```

Возможно, что пакет **cpconfig** не обнаруживается.

3. Для его обнаружения модифицируйте переменную PATH, выполнив команду:

```
export PATH="$PATH:/opt/cprocsp/sbin/amd64/"
```

4. Повторно выполните команду для просмотра лицензии.

   Если команда просмотра выводит сообщение, содержащее **error**, то это означает, что лицензия не установлена либо истек срок ее действия. Чтобы ввести новую лицензию, нужно выполнить команду установки лицензии. Затем выполнить команду просмотра срока действия. Если будет выведено сообщение:

```
[root@srv-kb-crypto ~]# cpconfig -license -view
The type of the license: DRV
license  - permanent
```

   значит – лицензия установлена правильно.

Добавление считывателя HDIMAGE

Для добавления считывателя используется команда:

```
cpconfig -hardware reader -add HDIMAGE store
```

Для просмотра списка настроенных считывателей используется команда:

```
./cpconfig -hardware reader -view
```

Настройка прав доступа к контейнерам

После того как контейнеры скопированы на компьютер с ПАК "Jinn-Server", нужно выполнить настройку прав доступа к контейнерам. Для этого выполните следующие команды, строго соблюдая указанную последовательность:

```
# cd /var/opt/cprocsp/keys/tccs/
# chown -R tccs:tccs .
# chmod 700 *
# chmod 600 `find ./ -type f`
```

Либо выполните скрипт jinn_install.sh с параметром "5".

Проверка прав доступа к контейнерам

Для проверки прав доступа нужно выполнить две команды.

Первая команда:

```
# ls -l *
```

На экране должна появиться информация, подобная следующей:

```
Security.000:
итого 24
-rw------- 1 tccs tccs 78 Сен 29 18:38 header.key
-rw------- 1 tccs tccs 56 Сен 29 18:38 masks2.key
-rw------- 1 tccs tccs 56 Сен 29 18:41 masks.key
-rw------- 1 tccs tccs 16 Сен 29 18:38 name.key
-rw------- 1 tccs tccs 36 Сен 29 18:38 primary2.key
-rw------- 1 tccs tccs 36 Сен 29 18:41 primary.key
Security.001:
итого 24
-rw------- 1 tccs tccs 78 Сен 29 18:47 header.key
```

```
-rw------- 1 tccs tccs 56 Сен 29 18:47 masks2.key
-rw------- 1 tccs tccs 56 Сен 29 18:49 masks.key
-rw------- 1 tccs tccs 21 Сен 29 18:47 name.key
-rw------- 1 tccs tccs 36 Сен 29 18:47 primary2.key
-rw------- 1 tccs tccs 36 Сен 29 18:49 primary.key
```

Вторая команда:

```
# ls -l
```

На экране должна появиться информация, подобная следующей:

```
итого 8
drwx------ 2 tccs tccs 4096 Фев 18 17:09 Security.000
drwx------ 2 tccs tccs 4096 Фев 18 17:09 Security.001
```

Редактирование конфигурационных файлов (csm.conf, tccs.ss, tccs.svs)

После установки выполните первоначальную настройку ПАК "Jinn-Server".

Для первоначальной настройки ПАК:

1. Отредактируйте конфигурационные файлы, хранящиеся в каталоге /opt/tccs/etc/.
2. Замените значение "@HOST@" реальным именем компьютера с установленным ПАК "Jinn-Server".

   После внесения изменений они должны принять вид, показанный на следующих рисунках.

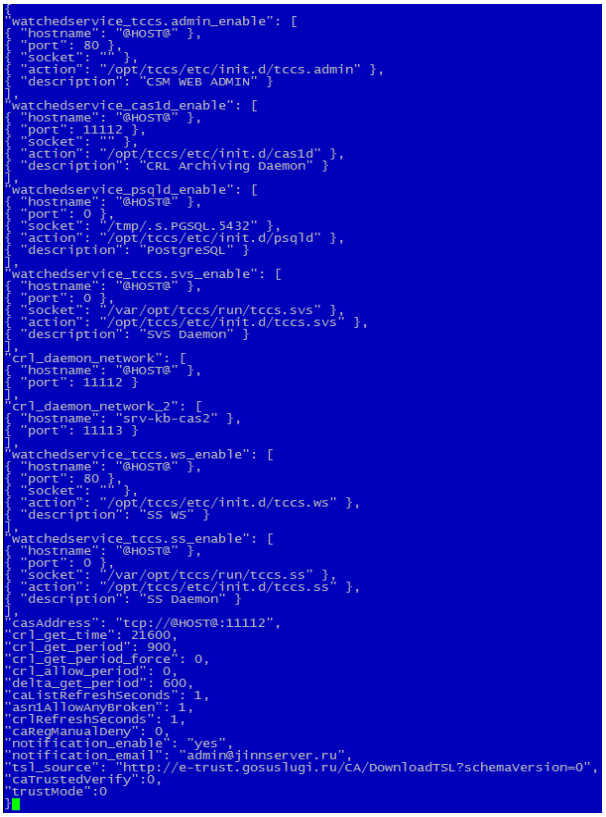


Рис. 6. Содержание файла csm.conf

Рис. 7. Содержание файла tccs.ss

Рис. 8. Содержание файла tccs.svs

Настройка прав доступа к конфигурационным файлам (csm.conf, tccs.ss, tccs.svs)

Для проверки прав доступа к конфигурационным файлам выполните команду:

```
# ls –l
```

На экране должна появиться информация, подобная следующей:

```
итого 60
-rw-rw-rw- 1 root root 2179 Дек  2 09:58 csm.conf
-rwxr-xr-x 1 root root  464 Ноя 27 17:40 tccs.ss
-rwxr-xr-x 1 root root  509 Ноя 27 17:40 tccs.svs
```

Если права доступа отличаются от показанных выше, выполните следующие команды:

```
# chown -R root:root csm.conf
# chmod 666 csm.conf
# chown -R root:root tccs.ss
# chown -R root:root tccs.svs
# chmod +x tccs.ss
# chmod +x tccs.svs
```

Редактирование файлов веб-сервисов

Отредактируйте конфигурационные файлы веб-сервисов, указав правильный IP-адрес компьютера, на котором работают сервисы.

Для редактирования файлов веб-сервисов:

1. Откройте для редактирования конфигурационный файл сервиса подписи:

```
# vim /opt/tccs/web-
services/services/SigningService/SigningService.wsdl
```

2. В открывшемся документе выполните команду:

```
:%s/62.181.53.2:18080/xxx.xxx.xxx.xxx:8080/g
```

   где вместо xxx.xxx.xxx.xxx нужно указать реальный IP-адрес.

3. Сохраните изменения командой ":wq".
4. Откройте для редактирования конфигурационный файл сервиса проверки и усиления подписи:

```
# vim /opt/tccs/web-
services/services/SignatureValidationService/Signature
ValidationService.wsdl
```

5. В открывшемся документе выполните команду:

```
:%s/62.181.53.2:18080/xxx.xxx.xxx.xxx:8080/g
```

   где вместо xxx.xxx.xxx.xxx нужно указать реальный IP-адрес.

6. Сохраните изменения командой ":wq".

Запуск "КриптоПро CSP"

Необходимо выполнить запуск "КриптоПро CSP" командой:

```
# /etc/init.d/cproGF start
```

Добавление сертификата УЦ и CRL через веб-интерфейс

Запуск служб:

1. Чтобы запустить службы веб-администрирования, выполните следующие команды:

```
# cd /opt/tccs/etc/init.d/
# ./tccs.admin status
```

2. Если на экране появится сообщение "Отказано в соединении", выполните следующие команды:

```
# ./tccs.admin stop
# ./tccs.admin start
# ./tccs.admin status
```

3. Чтобы запустить веб-сервер, нужно выполните следующую команду:

```
# ./tccs.ws status
```

4. Если на экране появится сообщение "Отказано в соединении", выполните следующие команды:

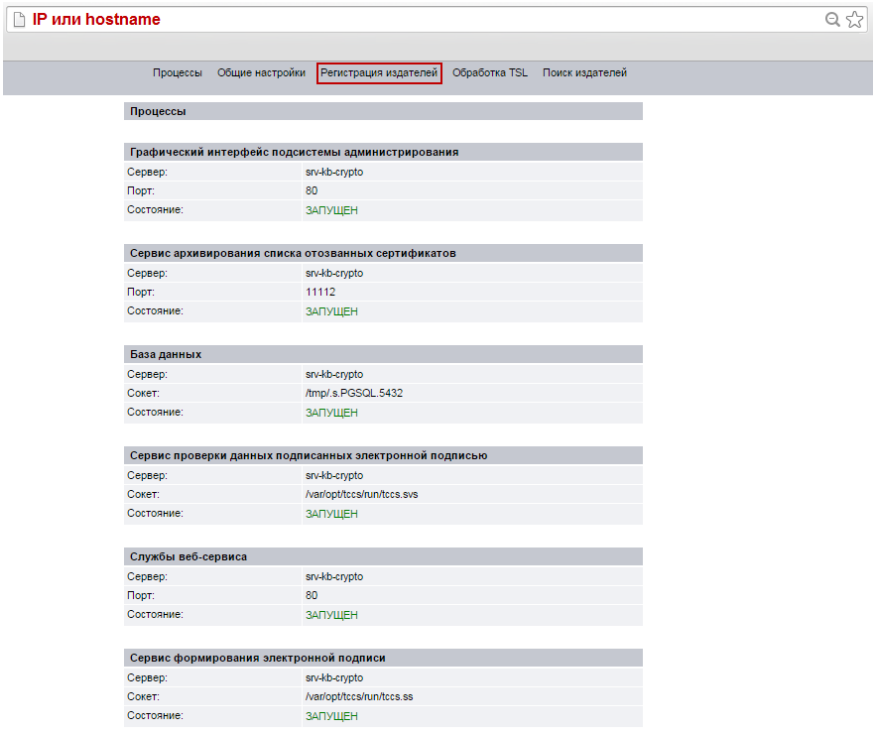
```
# ./tccs.ws stop
# ./tccs.ws start
# ./tccs.ws status
```

Работа с веб-интерфейсом

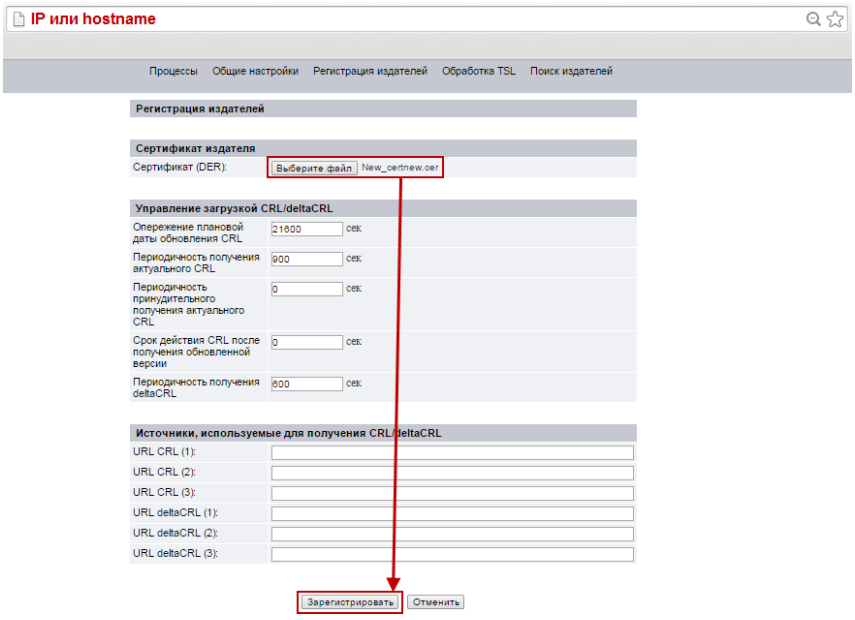
Для открытия веб-интерфейса:

1. В адресной строке браузера введите IP-адрес или hostname компьютера, на котором установлен ПАК "Jinn-Server". Откроется страница "Процессы". Если все ранее описанные настройки были выполнены правильно, то у пере-

численных на странице служб будет состояние "ЗАПУЩЕН", за исключением сервисов подписи и усиления подписи. Они будут остановлены, так как еще не установлен сертификат УЦ, выдавшего сертификаты и CRL. При отсутствии актуальных CRL для сертификата данного УЦ сервисы подписи и усиления подписи будут останавливаться.

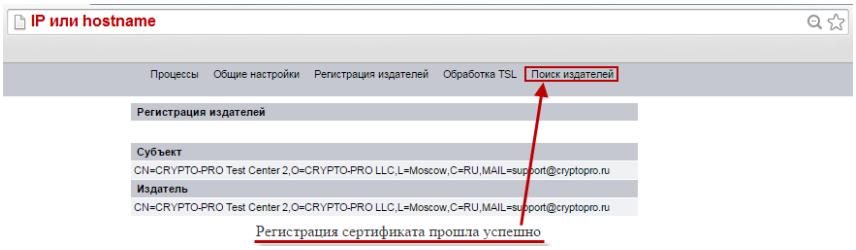


2. Перейдите в раздел "Регистрация издателей".

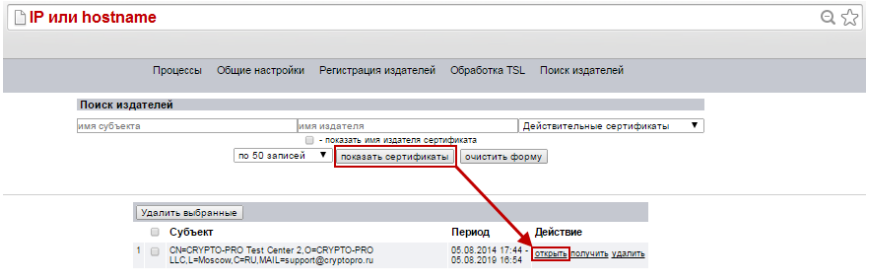


3. На странице "Регистрация издателей" в блоке "Сертификат издателя" нажмите кнопку "Выбрать" и стандартными средствами выбора файла выберите сертификат нужного УЦ. Если известны точки распространения CRL, их можно указать в блоке "Источники, используемые для получения CRL/deltaCRL", в полях "URL CRL" или "URL deltaCRL" в зависимости от типа выпускаемого CRL. После чего нажмите кнопку "Зарегистрировать".

В случае успешной регистрации будет выведено сообщение "Регистрация сертификата прошла успешно".

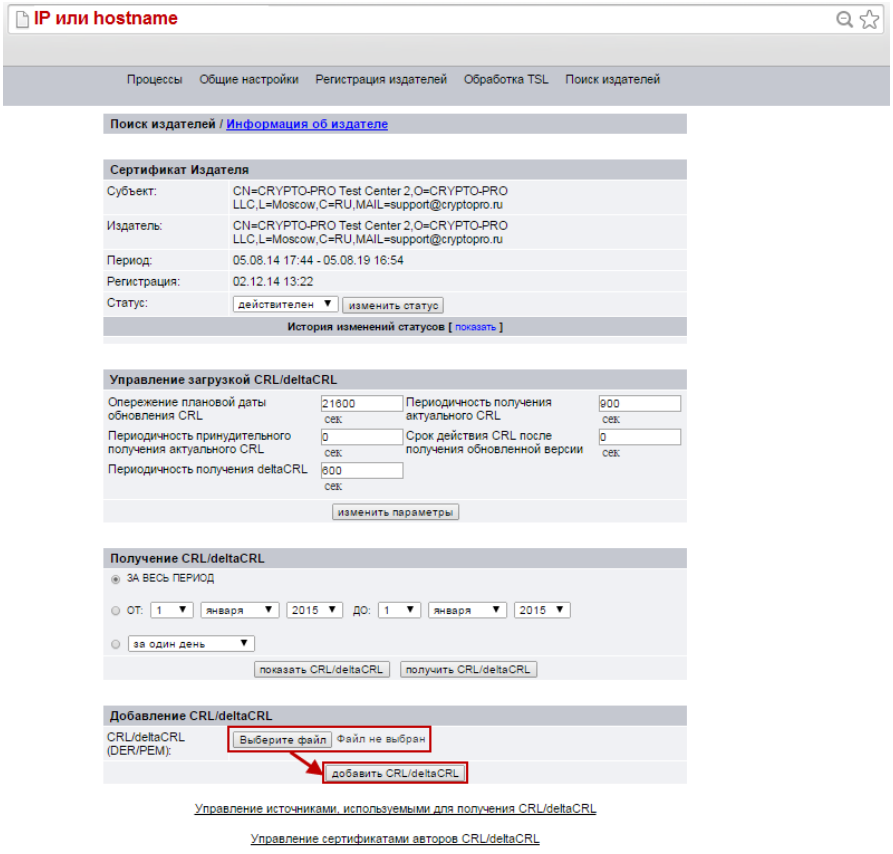


4. После этого перейдите в раздел "Поиск издателей".



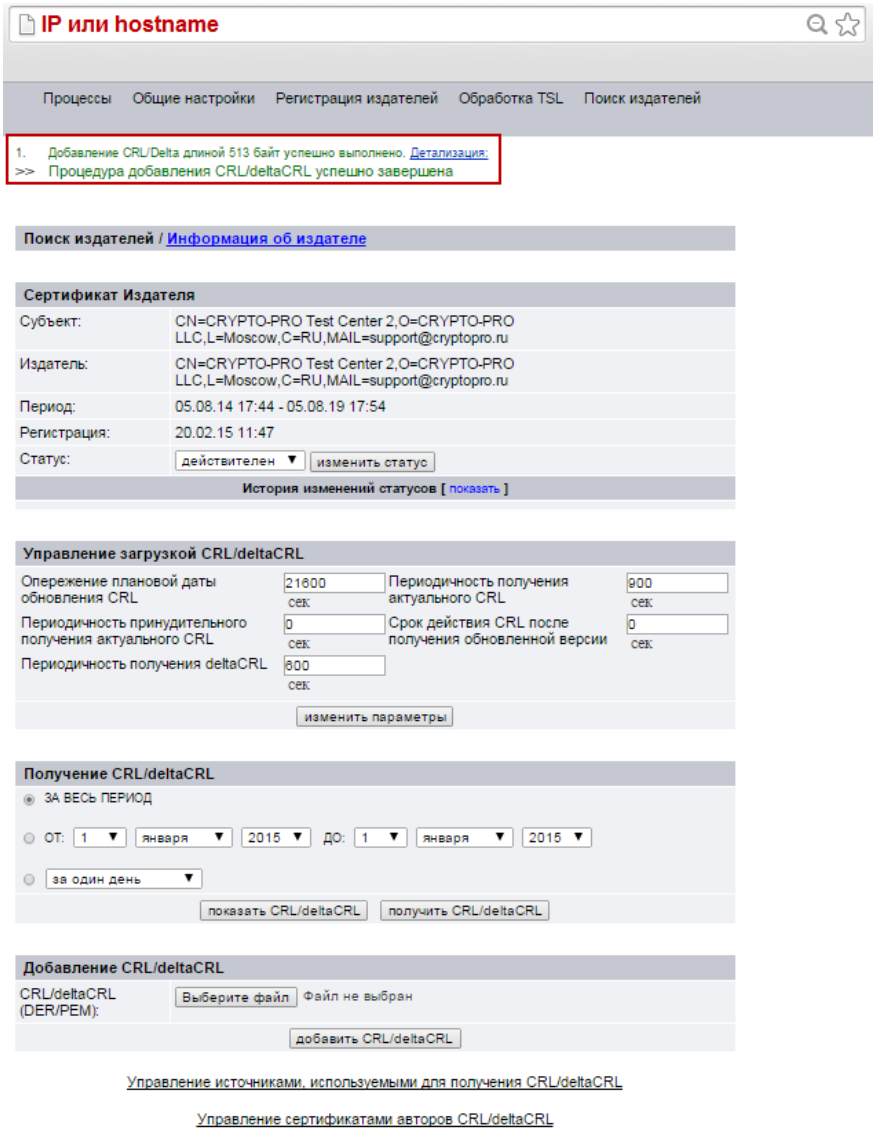
5. На странице "Поиск издателей" нажмите кнопку "Показать сертификаты". Первоначально добавляется один сертификат, и на экран будет выведен только он (в дальнейшем можно пользоваться фильтрами поиска).
6. В строке с найденным сертификатом нажмите на ссылку "Открыть".

   Будет выполнен переход на форму просмотра информации о сертификате.



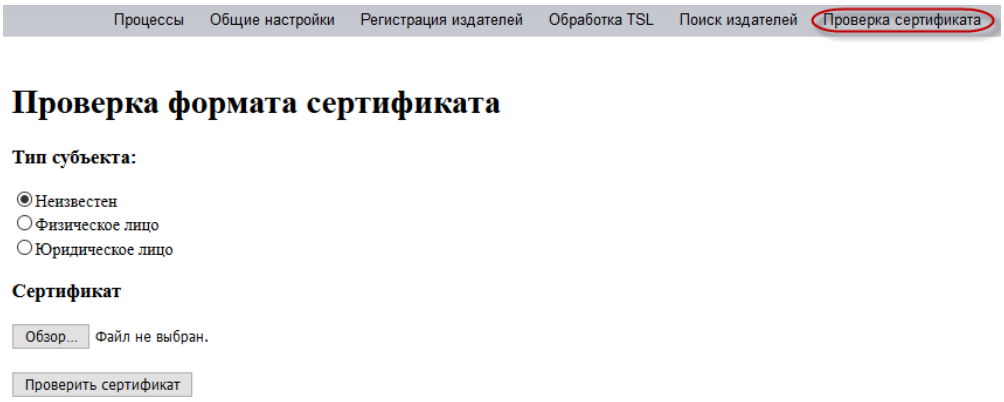
7. В блоке "Добавление CRL/deltaCRL" нажмите кнопку "Выберите файл" и стандартными средствами выбора выберите файл с CRL для данного издателя. Затем нажмите кнопку "Добавить CRL/detlaCRL".

   Если выбранный CRL корректен, на экране появится следующее сообщение.



Проверка сертификата с помощью веб-интерфейса и утилиты dumpasn1

Проверить сертификаты на соответствие требованиям к формату сертификата можно, перейдя на вкладку "Проверка сертификата":



Определяя факт наличия ошибок, этот сервис не всегда может точно указать, в чем именно состоит ошибка.

Например, в отношении переданного сертификата он дает не совсем адекватную диагностику.

До тех пор пока он не будет усовершенствован, необходимо дополнительно проверять "подозрительные" сертификаты, например, утилитой dumpasn1. (Свободно распространяемая утилита для платформ Windows и Linux.)

Вот фрагмент ее вывода от "проблемного" сертификата (ошибочное выделено; красное – ошибка вдвойне, так как и тип перепутан – должно быть BMPString, и значение пустое):

```
1395  137:         SEQUENCE {
1398    3:           OBJECT IDENTIFIER subjectAltName
(2 5 29 17)
1403  129:           OCTET STRING, encapsulates {
1406  127:             SEQUENCE {
1408   18:               [0] {
1410    3:                 OBJECT IDENTIFIER title (2 5
4 12)
1415   11:                 [0] {
1417    9:                   PrintableString
'625161011'
         :                 }
         :               }
1428    1:               [6] 30
1431   16:               [0] {
1433   10:                 OBJECT IDENTIFIER '1 2 643 3
61 502710 1 7'
1445    2:                 [0] {
1447    0:                   PrintableString
         :                     Error: Object has zero
length.
         :                 }
         :               }
1449   16:               [0] {
1451   10:                 OBJECT IDENTIFIER '1 2 643 3
61 502710 1 5'
1463    2:                 [0] {
1465    0:                   PrintableString
         :                     Error: Object has zero
length.
         :                 }
         :               }
1467   13:               [0] {
1469    7:                 OBJECT IDENTIFIER '1 2 643 1
12345 1'
1478    2:                 [0] {
1480    0:                   PrintableString
         :                     Error: Object has zero
length.
         :                 }
         :               }
1482   15:               [0] {
```

```
1484    9:                  OBJECT IDENTIFIER
         :                    friendlyName (for PKCS
#12) (1 2 840 113549 1 9 20)
1495    2:                  [0] {
1497    0:                    PrintableString
         :                      Error: Object has zero
length.
         :                    }
         :                  }
1499   16:                [0] {
1501   10:                  OBJECT IDENTIFIER '1 2 643 3
61 502710 1 8'
1513    2:                  [0] {
1515    0:                    PrintableString
         :                      Error: Object has zero
length.
         :                    }
         :                  }
1517   16:                [0] {
1519   10:                  OBJECT IDENTIFIER '1 2 643 3
61 502710 1 9'
1531    2:                  [0] {
1533    0:                    PrintableString
```

Для данной проверки необходимо выполнить команду, выводящую результат в файл:

```
# dumpasn1 test.der > test.asn
```

Затем открыть полученный файл для просмотра:

```
# vi test.asn
```

Проверка прав доступа к файлам pem

После добавления сертификатов нужно проверить наличие прав доступа к служебным файлам с информацией о добавленных сертификатах.

Для проверки наличия прав доступа:

1. Выполните следующие команды:

   ```
   # cd /var/opt/tccs/certs
   # ls -l
   ```

   На экране должна появиться информация, подобная следующей:

   ```
   итого 4
   -rw-r--r-- 1 tccs root     860 Фев 18 19:34 main.pem
   -rw-rw-rw- 1 apache apache   0 Фев 18 19:37
   other.pem.tmp
   ```

2. Если права доступа отличаются от показанных выше, выполните следующие команды:

   ```
   # chmod 666 other.pem.tmp
   # chown -R apache:apache other.pem.tmp
   ```

Добавление скрипта для отключения свопа

Чтобы избежать избыточных затрат оперативной памяти при обработке большого количества запросов, нужно добавить скрипт, который при запуске компьютера будет устанавливать блокировку на использование свопа.

Для установки блокировки:

1. Выполните команду создания файла:

```
# vim /etc/rc.d/rc.local
```

2. В открывшемся файле добавьте строку "sysctl -w vm.overcommit_memory=2".
3. Сохраните изменения командой ":wq".
4. Добавьте права доступа для созданного файла следующими командами:

```
# chmod +x rc.local
# chown -R root:root rc.local
```

5. Проверьте права доступа следующей командой:

```
# ls -l | grep rc.local
```

На экране должна появиться информация, подобная следующей:

```
-rwxr-xr-x 1 root root    33 Окт 14 11:43 rc.local
```

6. Проверьте включение блокировки свопа следующей командой:

```
# sysctl -A | grep commit
```

На экране появится информация, обязательно содержащая строку:

```
vm.overcommit_memory = 2
```

Редактирование файла hosts

Описанные ниже действия выполняются и на компьютере с ПАК "Jinn-Server", и на компьютере со сборщиком CRL, чтобы имена этих компьютеров правильно сопоставлялись их IP-адресам.

Для редактирования файла:

1. Выполните команду:

```
# cd /etc/
```

2. Откройте для редактирования файл hosts и добавьте недостающую строку, чтобы файл содержал обе указанные строки:

```
xxx.xxx.xxx.xxx          hostname_cas1
xxx.xxx.xxx.xxx          hostname_cas2
```

где: xxx.xxx.xxx.xxx – IP-адреса, hostname_cas1 – имя компьютера с ПАК "Jinn-Server", hostname_cas2 – имя компьютера со сборщиком CRL.

Логирование

Система сохраняет информацию о произошедших событиях, проведенных проверках, статусе ПАК "Jinn-Server" в логах с соответствующими именами.

Формат имен файлов с фрагментами:

```
<путь к файлу>-<компонент>.<событие>
```

- var.opt.tccs.log.access_log-ss – диагностика от веб-сервиса – обращение к сервису формирования подписи;
- var.opt.tccs.log.access_log-svs – диагностика от веб-сервиса – обращение к сервису проверки подписи;

- var.opt.tccs.log.error_log-svs.fault – диагностика от веб-сервиса – обращение к сервису проверки подписи с фатальной ошибкой в процессе обработки запроса;
- var.opt.tccs.log.error_log-svs.stopped – диагностика от веб-сервиса – обращение к сервису проверки подписи, в ситуации когда сервис не запущен;
- var.opt.tccs.log.error_log-ws.start – запуск веб-сервиса;
- var.opt.tccs.log.error_log-ws.stop – останов веб-сервиса;
- var.opt.tccs.log.misc-watchdog.ok – самоконтроль – проверка работоспособности компонентов – все компоненты в рабочем состоянии;
- var.opt.tccs.log.misc-watchdog.ss.start – самоконтроль – проверка работоспособности компонентов – обнаружено отсутствие запущенного прикладного сервиса формирования подписи, сервис запущен;
- var.opt.tccs.log.misc-watchdog.svs.start – самоконтроль – проверка работоспособности компонентов – обнаружено отсутствие запущенного прикладного сервиса проверки подписи, сервис запущен;
- var.opt.tccs.log.online-ss.sign – обращение к прикладному сервису формирования подписи;
- var.opt.tccs.log.online-ss.start – старт прикладного сервиса формирования подписи;
- var.opt.tccs.log.online-ss.stop – останов сервиса формирования подписи;
- var.opt.tccs.log.online-ss.ttl – смена поколений рабочих процессов в сервисе формирования подписи;
- var.opt.tccs.log.online-svs.start – запуск прикладного сервиса проверки подписи;
- var.opt.tccs.log.online-svs.stop – останов сервиса проверки;
- var.opt.tccs.log.online-svs.ttl – смена поколений рабочих процессов сервиса проверки;
- var.opt.tccs.log.online-svs.verify – обращение к сервису проверки с запросом.

Приложение

База данных сервиса CAS

Структура таблиц

База данных CAS включает в себя хранилища CAS-1 и CAS-2, управление которыми осуществляется системой управления базами данных (СУБД). Хранилища реализованы как наборы таблиц с взаимосвязанной или дополняющей информацией. Взаимодействие с СУБД из уровня приложений осуществляется по стандартному интерфейсу ODBC.

Для хранилища CAS-1 это:

- таблица CERT_CA – содержит информацию о сертификатах издателей;
- таблица CERT_STATUS – содержит всю историю изменений статуса для сертификатов издателей;
- таблица CERT_TRUSTED – содержит информацию, определяющую, являются ли сертификаты издателей доверенными или нет;
- таблица CERT_CA_INFO – содержит дополнительную информацию о сертификатах издателей, в частности – указываются реестровый номер и название УЦ из TSL-списка;
- таблица CERT_TSL – содержит информацию о сертификатах, участвующих в проверке ЭЦП/ЭП TSL-списков;
- таблица CERT_OTHER – содержит сертификаты авторов CRL/deltaCRL;
- таблица CERT_CONFIG – содержит конфигурационные параметры, используемые для загрузок CRL/deltaCRL;
- таблица CRL_SOURCE – содержит информацию об источниках получения CRL/deltaCRL;
- таблица CRL_MIRROR – содержит HTTP-ссылки для получения CRL/deltaCRL;
- таблица CRL – содержит регулярные CRL;
- таблица DELTA – содержит deltaCRL;
- таблица TSL – содержит TSL-списки.

CERT_CA:

Поле	Описание
cert_id	Идентификатор сертификата издателя
subject	Имя субъекта
issuer	Имя издателя
certificate	Сертификат в кодировке BASE64
not_before	Время начала действия сертификата
not_after	Время окончания действия сертификата
status	Статус сертификата
create_date	Время создания записи в таблице

CERT_STATUS:

Поле	Описание
cert_id	Идентификатор сертификата издателя
status	Статус
started	Время указания статуса
create_date	Время создания записи в таблице

CERT_TRUSTED:

Поле	Описание
cert_id	Идентификатор сертификата издателя
trusted	Флаг (доверенный/недоверенный)
create_date	Время создания записи в таблице

CERT_CA_INFO:

Поле	Описание
cert_id	Идентификатор сертификата издателя
tsl_id	Реестровый номер УЦ
tsl_name	Название УЦ
create_date	Время создания записи в таблице

CERT_TSL:

Поле	Описание
cert_id	Идентификатор сертификата, участвующего в проверке ЭЦП/ЭП TSL-списка
subject	Имя субъекта
issuer	Имя издателя
certificate	Сертификат в кодировке BASE64
not_before	Время начала действия сертификата
not_after	Время окончания действия сертификата
status	Статус сертификата
create_date	Время создания записи в таблице

CERT_OTHER:

Поле	Описание
cert_id	Идентификатор сертификата издателя
subject	Имя субъекта
issuer	Имя издателя
certificate	Сертификат в кодировке BASE64
not_before	Время начала действия сертификата
not_after	Время окончания действия сертификата
create_date	Время создания записи в таблице

CERT_CONFIG:

Поле	Описание
cert_id	Идентификатор сертификата издателя
crl_get_time	Период времени до наступления даты из атрибута next_update, с которого необходимо начинать процедуры обновления CRL
crl_get_period	Периодичность, с которой производятся попытки получить актуальный CRL
crl_allow_period	Период времени, в течение которого CRL считается действительным даже после наступления даты из атрибута next_update

Поле	Описание
delta_get_period	Периодичность, с которой производятся попытки получить deltaCRL
create_date	Время создания записи в таблице

CRL_SOURCE:

Поле	Описание
cert_id	Идентификатор сертификата издателя
source_id	Идентификатор источника
type	Тип источника (CRLDistributionPoint/FreshestCRL)
download_time	Время получения CRL/deltaCRL
check_time	Время следующей попытки получения CRL/deltaCRL
create_date	Время создания записи в таблице

CRL_MIRROR:

Поле	Описание
cert_id	Идентификатор сертификата издателя
source_id	Идентификатор источника
url_id	Идентификатор HTTP-ссылки получения CRL/deltaCRL
url	HTTP-ссылка для получения CRL/deltaCRL
priority	Приоритет, по которому выбирается первоочередная ссылка для получения CRL/deltaCRL
create_date	Время создания записи в таблице

CRL:

Поле	Описание
cert_id	Идентификатор сертификата издателя
crl_number_full	Номер CRL (текстовое представление)
crl_number	Номер CRL (численное представление)
crl	CRL в кодировке BASE64
this_update	Время начала действия CRL
next_update	Время окончания действия CRL
reasons	Причины отзыва сертификатов
create_date	Время создания записи в таблице

DELTA:

Поле	Описание
cert_id	Идентификатор сертификата издателя
crl_number_full	Номер регулярного CRL (текстовое представление)
crl_number	Номер регулярного CRL (численное представление)
delta	deltaCRL в кодировке BASE64
this_update	Время начала действия CRL
reasons	Причины отзыва сертификатов
create_date	Время создания записи в таблице

TSL:

Поле	Описание
tsl_id	Идентификатор TSL-списка
version	Номер версии TSL-списка
started	Дата создания TSL-списка
ca_count	Количество сертификатов издателей, указанных в TSL-списке
tsl	TSL-список
create_date	Время создания записи в таблице

Для хранилища CAS-2 это:

- таблица CERT_CA – содержит информацию о сертификатах издателей;
- таблица CERT_CONFIG – содержит конфигурационные параметры, используемые для загрузок CRL/deltaCRL;
- таблица CRL_SOURCE – содержит информацию об источниках получения CRL/deltaCRL;
- таблица CRL_MIRROR – содержит HTTP-ссылки для получения CRL/deltaCRL;
- таблица CERT_TSL – содержит информацию о сертификатах, участвующих в проверке ЭЦП/ЭП TSL списков.

CERT_CA:

Поле	Описание
cert_id	Идентификатор сертификата издателя
subject	Имя субъекта
issuer	Имя издателя
certificate	Сертификат в кодировке BASE64
not_before	Время начала действия сертификата
not_after	Время окончания действия сертификата
status	Статус сертификата
create_date	Время создания записи в таблице

CERT_CONFIG:

Поле	Описание
cert_id	Идентификатор сертификата издателя
crl_get_time	Период времени до наступления даты из атрибута next_update, с которого необходимо начинать процедуры обновления CRL
crl_get_period	Периодичность, с которой производятся попытки получить актуальный CRL
crl_allow_period	Период времени, в течение которого CRL считается действительным даже после наступления даты из атрибута next_update
delta_get_period	Периодичность, с которой производятся попытки получить deltaCRL
create_date	Время создания записи в таблице

CRL_SOURCE:

Поле	Описание
cert_id	Идентификатор сертификата издателя

Поле	Описание
source_id	Идентификатор источника
type	Тип источника (CRLDistributionPoint/FreshestCRL)
download_time	Время получения CRL/deltaCRL
check_time	Время следующей попытки получения CRL/deltaCRL
create_date	Время создания записи в таблице

CRL_MIRROR:

Поле	Описание
cert_id	Идентификатор сертификата издателя
source_id	Идентификатор источника
url_id	Идентификатор HTTP-ссылки получения CRL/deltaCRL
url	HTTP-ссылка для получения CRL/deltaCRL
priority	Приоритет, по которому выбирается первоочередная ссылка для получения CRL/deltaCRL
create_date	Время создания записи в таблице

CERT_TSL:

Поле	Описание
cert_id	Идентификатор сертификата, участвующего в проверке ЭЦП/ЭП TSL-списка
subject	Имя субъекта
issuer	Имя издателя
certificate	Сертификат в кодировке BASE64
not_before	Время начала действия сертификата
not_after	Время окончания действия сертификата
status	Статус сертификата
create_date	Время создания записи в таблице

Создание таблиц

Создание таблиц осуществляется в базе данных с помощью инструкций в соответствии с используемой СУБД.

Инструкции для хранилища CAS-1:

```
CREATE SEQUENCE cert_id_seq;
CREATE TABLE cert_ca (
        cert_id                 integer NOT NULL
PRIMARY KEY DEFAULT NEXTVAL('"cert_id_seq"'::TEXT),
        subject                 text,
        issuer                  text,
        certificate             text,
        not_before              bigint,
        not_after               bigint,
        status                  integer,
        create_date             bigint,
```

```
        UNIQUE (subject, issuer));
CREATE INDEX cert_ca_index1 ON cert_ca (cert_id);
CREATE INDEX cert_ca_index2 ON cert_ca (subject);
CREATE INDEX cert_ca_index3 ON cert_ca (issuer);
CREATE INDEX cert_ca_index4 ON cert_ca (not_before);
CREATE INDEX cert_ca_index5 ON cert_ca (not_after);
CREATE INDEX cert_ca_index6 ON cert_ca (status);

CREATE TABLE cert_other (
        cert_id                 integer,
        subject                 text,
        issuer                  text,
        certificate             text,
        not_before              bigint,
        not_after               bigint,
        create_date             bigint,
        UNIQUE (subject, issuer));
CREATE INDEX cert_other_index1 ON cert_other (cert_id);
CREATE INDEX cert_other_index2 ON cert_other (subject);
CREATE INDEX cert_other_index3 ON cert_other (issuer);
CREATE INDEX cert_other_index4 ON cert_other
(not_before);
CREATE INDEX cert_other_index5 ON cert_other
(not_after);

CREATE TABLE cert_config (
        cert_id                         integer NOT
NULL,
        crl_get_time            integer,
        crl_get_period          integer,
        crl_allow_period        integer,
        delta_get_period        integer,
        create_date                     bigint,
        UNIQUE (cert_id));
CREATE INDEX cert_cfg_index1 ON cert_config (cert_id);
CREATE INDEX cert_cfg_index2 ON cert_config
(crl_get_time);
CREATE INDEX cert_cfg_index3 ON cert_config
(crl_get_period);
CREATE INDEX cert_cfg_index4 ON cert_config
(crl_allow_period);
CREATE INDEX cert_cfg_index5 ON cert_config
(delta_get_period);

CREATE TABLE crl_source (
        cert_id                 integer NOT NULL,
```

```
        source_id               integer NOT NULL,
        type                    integer,
        download_time   bigint,
        check_time              bigint,
        create_date             bigint,
        UNIQUE (cert_id, source_id));
CREATE INDEX crl_source_index1 ON crl_source (cert_id);
CREATE INDEX crl_source_index2 ON crl_source
(source_id);
CREATE INDEX crl_source_index3 ON crl_source (type);
CREATE INDEX crl_source_index4 ON crl_source
(download_time);
CREATE INDEX crl_source_index5 ON crl_source
(check_time);

CREATE TABLE crl_mirror (
        cert_id                 integer NOT NULL,
        source_id               integer NOT NULL,
        url_id                  integer NOT NULL,
        url                             text,
        priority                integer,
        create_date             bigint,
        UNIQUE (cert_id, source_id, url_id));
CREATE INDEX crl_mirror_index1 ON crl_mirror (cert_id);
CREATE INDEX crl_mirror_index2 ON crl_mirror
(source_id);
CREATE INDEX crl_mirror_index3 ON crl_mirror (url_id);
CREATE INDEX crl_mirror_index4 ON crl_mirror
(priority);

CREATE TABLE crl (
        cert_id                 integer NOT NULL,
        crl_number_full text,
        crl_number              integer,
        crl                     text,
        this_update     bigint,
        next_update             bigint,
        reasons                 integer,
        create_date             bigint,
        UNIQUE (cert_id, crl));
CREATE INDEX crl_index1 ON crl (cert_id);
CREATE INDEX crl_index2 ON crl (crl_number_full);
CREATE INDEX crl_index3 ON crl (crl_number);
CREATE INDEX crl_index4 ON crl (this_update);
CREATE INDEX crl_index5 ON crl (next_update);
```

```
CREATE TABLE delta (
        cert_id                 integer NOT NULL,
        crl_number_full text,
        crl_number              integer,
        delta                   text,
        this_update     integer,
        reasons                 integer,
        create_date             bigint,
        UNIQUE (cert_id, delta));
CREATE INDEX delta_index1 ON delta (cert_id);
CREATE INDEX delta_index2 ON delta (crl_number_full);
CREATE INDEX delta_index3 ON delta (crl_number);
CREATE INDEX delta_index4 ON delta (this_update);
```

Инструкции для хранилища CAS-2:

```
CREATE TABLE cert_ca (
        cert_id                 integer NOT NULL,
        subject                 text,
        issuer                  text,
        certificate             text,
        not_before              bigint,
        not_after               bigint,
        status                  integer,
        create_date             bigint,
        UNIQUE (subject, issuer));
CREATE INDEX cert_ca_index1 ON cert_ca (cert_id);
CREATE INDEX cert_ca_index2 ON cert_ca (subject);
CREATE INDEX cert_ca_index3 ON cert_ca (issuer);
CREATE INDEX cert_ca_index4 ON cert_ca (not_before);
CREATE INDEX cert_ca_index5 ON cert_ca (not_after);
CREATE INDEX cert_ca_index6 ON cert_ca (status);

CREATE TABLE cert_config (
        cert_id                         integer NOT
NULL,
        crl_get_time            integer,
        crl_get_period          integer,
        crl_allow_period        integer,
        delta_get_period        integer,
        create_date                     bigint,
        UNIQUE (cert_id));
CREATE INDEX cert_cfg_index1 ON cert_config (cert_id);
CREATE INDEX cert_cfg_index2 ON cert_config
(crl_get_time);
```

```
CREATE INDEX cert_cfg_index3 ON cert_config
(crl_get_period);
CREATE INDEX cert_cfg_index4 ON cert_config
(crl_allow_period);
CREATE INDEX cert_cfg_index5 ON cert_config
(delta_get_period);

CREATE TABLE crl_source (
        cert_id                 integer NOT NULL,
        source_id               integer NOT NULL,
        type                    integer,
        download_time   bigint,
        check_time              bigint,
        create_date             bigint,
        UNIQUE (cert_id, source_id));
CREATE INDEX crl_source_index1 ON crl_source (cert_id);
CREATE INDEX crl_source_index2 ON crl_source
(source_id);
CREATE INDEX crl_source_index3 ON crl_source (type);
CREATE INDEX crl_source_index4 ON crl_source
(download_time);
CREATE INDEX crl_source_index5 ON crl_source
(check_time);

CREATE TABLE crl_mirror (
        cert_id                 integer NOT NULL,
        source_id               integer NOT NULL,
        url_id                  integer NOT NULL,
        url                             text,
        priority                integer,
        create_date             bigint,
        UNIQUE (cert_id, source_id, url_id));
CREATE INDEX crl_mirror_index1 ON crl_mirror (cert_id);
CREATE INDEX crl_mirror_index2 ON crl_mirror
(source_id);
CREATE INDEX crl_mirror_index3 ON crl_mirror (url_id);
CREATE INDEX crl_mirror_index4 ON crl_mirror
(priority);
```

Документация

1	Программно-аппаратный комплекс квалифицированной электронной подписи "Jinn-Server" версии 1.0. Руководство администратора
2	Программно-аппаратный комплекс квалифицированной электронной подписи "Jinn-Server" версии 1.0. Руководство программиста
3	Программно-аппаратный комплекс квалифицированной электронной подписи "Jinn-Server" версии 1.0. Руководство пользователя