Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт электронного обучения
Направление 230105 «Программное обеспечение вычислительной техники и автоматизированных систем»
Кафедра Автоматики и компьютерных систем

ДИПЛОМНАЯ РАБОТА

Тема работы
Обеспечение безопасности системы управления инфраструктурой центра обработки данных ТПУ

УДК 004.056.52

Студент

Группа	ФИО	Подпись	Дата
38001	Силищев О. Ю.		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Нач. управления по информатизации ТПУ	Квасников К.Г.			

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры Менеджмента	Конотопский В.Ю.	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры экологии и БЖД	Невский Е.С.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
АиКС	Фадеев А.С.	к.т.н.		

Томск — 2016 г.

Планируемые результаты обучения

Код результатов	Результат обучения (выпускник должен быть готов)
	Профессиональные компетенции
Р1	Применять глубокие естественнонаучные и математические знания для решения научных и инженерных задач в области информатики и вычислительной техники.
Р2	Применять глубокие специальные знания в области информатики и вычислительной техники для решения междисциплинарных инженерных задач.
Р3	Ставить и решать инновационные задачи инженерного анализа, связанные с созданием аппаратных и программных средств информационных и автоматизированных систем, с использованием аналитических методов и сложных моделей.
Р4	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
Р5	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
Р6	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
	Универсальные компетенции
Р7	Использовать глубокие знания по проектному менеджменту для ведения инновационной инженерной деятельности с учетом юридических аспектов защиты интеллектуальной собственности.
Р8	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
Р9	Эффективно работать индивидуально и в качестве члена и руководителя группы, в том числе междисциплинарной и международной, при решении инновационных инженерных задач.
Р10	Демонстрировать личную ответственность и ответственность за работу возглавляемого коллектива, приверженность и готовность следовать профессиональной этике и нормам ведения инновационной инженерной деятельности. Демонстрировать глубокие знания правовых, социальных, экологических и культурных аспектов инновационной инженерной деятельности.
Р11	Демонстрировать способность к самостоятельному обучению, непрерывному самосовершенствованию в инженерной деятельности, способность к педагогической деятельности.

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт электронного обучения
Направление 230105 «Программное обеспечение вычислительной техники и автоматизированных систем»
Кафедра Автоматики и компьютерных систем

УТВЕРЖДАЮ:
Зав. кафедрой
_____ 01.10.2015 г. Фадеев А.С.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

дипломной работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	**ФИО**
38001	Силищеву Олегу Юрьевичу

Тема работы:

Обеспечение безопасности системы управления инфраструктурой центра обработки данных ТПУ	
Утверждена приказом директора (дата, номер)	от 15.04.2016, №2917/с

Срок сдачи студентом выполненной работы:	01.06.2016 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе *(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).*	технические спецификации и стандарты, структура центра обработки данных ТПУ, описание организации доступа к управляемым ресурсам, свободное программное обеспечение, методическая и техническая литература, вычислительные ресурсы центра обработки данных

Перечень подлежащих исследованию, проектированию и разработке вопросов *(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).*	Теоретические основы информационной безопасности Оценка тенденций в сфере защиты информации Информационная безопасность, шифрование и аутентификация Анализ угроз в сетях передачи данных и методология защиты информации Систематизация информации о разрабатываемой системе управления инфраструктурой Характеристика существующей системы управления Требования к разрабатываемой системе управления инфраструктурой Формирование требований к системе с учетом анализа предметной области Выбор инструментария решения задачи Проектирование системы управления инфраструктурой Реализация системы управления инфраструктурой Финансовый менеджмент, ресурсоэффективность и ресурсосбережение Социальная ответственность
Перечень графического материала *(с точным указанием обязательных чертежей)*	

Консультанты по разделам выпускной квалификационной работы

(с указанием разделов)

Раздел	**Консультант**
1, 2, 3, 4	Квасников К.Г.
5	Конотопский В.Ю.
6	Невский Е.С.
Приложение А, Б, В, Г, Д	Квасников К.Г.

Названия разделов, которые должны быть написаны на русском и иностранном языках:

На русском языке: обзор тенденций и основ информационной безопасности, Исследование системы управления ЦОД в контексте информационной безопасности, проектирование системы управления инфраструктурой, реализация системы управления инфраструктурой, финансовый менеджмент, ресурсоэффективность и ресурсосбережение, социальная ответственность

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	01.10.2015 г.

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Нач. управления по информатизации ТПУ	Квасников К.Г.			01.10.15

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
38001	Силищев О.Ю.		01.10.15

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
38001	Силищев Олег Юрьевич

Институт	Электронного обучения	Кафедра	АиКС
Уровень образования	**Специалист**	**Направление/специальность**	**230105 Программное обеспечение вычислительной техники и автоматизированных систем**

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Оценка коммерческого и инновационного потенциала НТИ	
2. Разработка устава научно-технического проекта	
3. Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок	
4. Определение ресурсной, финансовой, экономической эффективности	
1. **Перечень графического материала** *(с точным указанием обязательных чертежей):*	
1. Перечень работ и оценка трудоемкости	

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры Менеджмента	Конотопский В.Ю.	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
38001	Силищев О.Ю.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
38001	Силищев Олег Юрьевич

Институт	Электронного обучения	Кафедра	АиКС
Уровень образования	**Специалист**	**Направление/специальность**	**230105 Программное обеспечение вычислительной техники и автоматизированных систем**

1. Анализ возможных сбоев разрабатываемой системы	Проанализировать возможные сбои в разрабатываемой системе. Подготовить базу для выявления причин сбоев, доработки системы с целью исключения сбоев, либо разработки планов действий ведущих к скорейшему устранению возникших сбоев.
2. Анализ причин сбоев в работе системы	Выявления причин возможных сбоев в работе системы на основе анализа, проведенного этапом выше. Подготовка базы для устранения причин вероятных сбоев с целью повышения стабильности системы.
3. Меры по аппаратной защите системы	Разработка комплекса мер по защите системы от сбоев, носящих аппаратный характер. Исключение единых точек отказа на аппаратном уровне.
4. Организационные меры, обеспечивающие защиту системы	Формулировка мер организационного характера направленных на защиту системы от сбоев выявленных ранее.
5. Меры по программной защите системы	Разработка комплекса мер по защите системы от сбоев, носящих программный характер. Исключение вероятности заражения системы вредоносным программным обеспечением. Устранение уязвимостей программных продуктов.
6. Требования к аппаратному и программному обеспечению	Требования к аппаратному и программному обеспечению сформулированы для корректной реализации системы управления инфраструктурой, исключения вероятности возникновения сбоев в результате нехватки ресурсов, либо несовместимости программного обеспечения.
7. Влияние данной работы на существующую инфраструктуру	Сравнение исходного состояния системы с полученным в результате выполнения данной работы.

Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры экологии и БЖД	Невский Е.С.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
38001	Силищев О.Ю.		

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт электронного обучения
Направление 230105 «Программное обеспечение вычислительной техники и автоматизированных систем»
Кафедра Автоматики и компьютерных систем
Уровень образования – инженер
Период выполнения – осенний семестр 2016 учебного года

Форма представления работы:

Дипломная работа

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	01.06.2016 г.

Дата контроля	Название раздела	Максимальный балл раздела
2016 г.	Основная часть	70
2016 г.	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
2016 г.	Социальная ответственность	15

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Заведующий кафедрой АиКС	Фадеев А.С.	к.т.н.		

СОГЛАСОВАНО:

Зав. Кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
АиКС	Фадеев А.С.	к.т.н.		

РЕФЕРАТ

Выпускная квалификационная работа содержит 109 с., 2 рис., 14 табл., 57 источников, 5 приложений.

Ключевые слова: криптография, Puppet, SSH, SSL, Fail2ban, Rsyslog, Debian Linux, система управления, автоматизация.

Объектом работы является система управления большим количеством узлов, развернутых на мощностях центра обработки данных ТПУ (ЦОД).

Цель работы — модернизация существующей системы управления инфраструктурой ЦОД с учетом современных требований к безопасности и удобству.

В процессе работы проводилось исследование существующей системы управления, выяснялись характеристики управляемых узлов, выявлялись слабые места. Осуществлялся поиск и систематизация информации о современных подходах к обеспечению информационной безопасности. На основе полученных данных вырабатывались решения, призванные улучшить систему.

В результате работы была реализована система управления узлами ЦОД, отвечающая поставленным целям.

Основные конструктивные, технологические и технико-эксплуатационные характеристики: реализованная система позволяет организовать процесс управления узлами в ЦОД с учетом современных требований к обеспечению информационной.

Степень внедрения: опытная эксплуатация.

Область применения: администрирование инфраструктур с большим количеством узлов под управлением дистрибутивов операционной системы GNU/Linux.

Экономическая эффективность/значимость работы: модернизация системы без прямых денежных затрат.

Планируется распространить систему управления на максимально большое количество узлов ЦОД, использовать систему как базис для реализации новых проектов автоматизации.

Определения, обозначения, сокращения, нормативные ссылки

Обозначения и сокращения

ЦОД — центр обработки данных;

SSH (англ. Secure Shell, безопасная оболочка) — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений;

SSL (англ. Secure Sockets Layer, уровень защищённых сокетов) — криптографический протокол, назначение которого обеспечить шифрование сетевого соединения;

RAID (англ. Redundant Array of Independent Disks, избыточный массив независимых дисков) — технология объединения нескольких дисков в одно логическое устройство для повышения производительности и доступности;

RPM (RPM Package Manager) — менеджер пакетов дистрибутива Red Hat Enterprise Linux операционной системы GNU/Linux и семейства производных от него дистрибутивов, таких как CentOS Linux, Oracle Linux и многих других;

rpm-пакет — архивный контейнер содержащий подготовленное для установки программное обеспечение, инсталляционные скрипты, а так же метаданные;

YUM (Yellow dog Updater, Modified) — открытый консольный менеджер RPM-пакетов;

deb — формат файла, который содержит бинарный инсталляционный пакет программного обеспечения для дистрибутива Debian операционной системы GNU/Linux;

APT (Advanced Packaging Tool) — программа для установки, обновления и удаления deb-пакетов в операционных системах Debian;

API (англ. Application Programming Interface, интерфейс программирования приложений, интерфейс прикладного программирования) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах;

GSSAPI (GSS, GSSAPI, англ. Generic Security Services API, общий программный интерфейс сервисов безопасности) — API для доступа к сервисам безопасности.;

IETF (англ. Internet Engineering Task Force, Инженéрный совéт Интернéта) — открытое международное сообщество проектировщиков, учёных, сетевых операторов и провайдеров, созданное в 1986 году и занимающееся развитием протоколов и архитектуры Интернета.;

TCP (англ. Transmission Control Protocol, протокол управления передачей) — протокол обеспечивающий надежную передачу сквозного байтового потока по ненадежной интерсети;

S.M.A.R.T (от англ. Self-Monitoring, Analysis and Reporting Technology — технология самоконтроля, анализа и отчётности).

Оглавление

Введение

С каждым днем информация играет всё более важную роль в нашей повседневной жизни. Мы стали информационно зависимым обществом XXI века и живем в мире команд и запросов. Это означает потребность в получении необходимой информации независимо от времени и места.

Для обеспечения бесперебойной доступности информационных систем Томского Политехнического Университета (ТПУ) на базе структурного подразделения Главный информационный узел (ГИУ) создан центр обработки данных (ЦОД). ЦОД представляет собой консолидированные в рамках одной инфраструктуры ресурсы вычисления, передачи и хранения данных, обеспечивающие работу большинства информационных систем университета.

С целью обеспечить возложенную на ЦОД задачу, в аппаратную и программную части ЦОД вложены существенные средства. Закуплено дорогостоящее надежное оборудование и не менее дорогостоящие программные продукты от именитых производителей.

Однако, существенной угрозой стабильности ЦОД является ненадлежащее внимание к вопросам информационной безопасности при осуществлении процедур администрирования информационных систем ЦОД.

Целью данной работы является модернизация существующей системы управления инфраструктурой центра обработки данных ТПУ (ЦОД) с учетом современных требований к безопасности, без ухудшения удобства управления и финансовых вложений на приобретение программных или аппаратных средств. Ключевой задачей является обеспечение безопасности системы управления ЦОД.

Глава 1 Обзор тенденций и основ информационной безопасности

На первом этапе работы было проведено исследование современных тенденций в сфере защиты информации, проведено знакомство с базовыми понятиями информационной безопасности, шифрования и аутентификации.

1.1 Оценка тенденций в сфере защиты информации

В ноябре 1988 года, Роберт Моррис-младший (Robert Morris, Jr.) выпустил в Интернет своего первого “червя”. Тогда администраторы информационных систем впервые столкнулись с реальной опасностью всемирного масштаба. “Червь” Морриса привел к потере тысяч часов рабочего времени системных администраторов [1].

С тех пор и по настоящее время наблюдается рост числа инцидентов в области информационной безопасности. Международной ассоциацией, объединяющей профессионалов в области ИТ-аудита, ИТ-консалтинга, управления ИТ-рисками и информационной безопасности ISACA (Information Systems Audit and Control Association) в 2015 году было проведено исследование в области кибербезопасности. Опрос более полутора тысяч специалистов по защите информации показал:

- более 75% опрошенных отметили увеличение количества атак в 2014 году по сравнению с 2013;
- 30% опрошенных в 2014 году имели дело со взломами и кражей конфиденциальной информации.
- 82% опрошенных ожидают роста количества инцидентов в 2015 году [2].

По данным Лаборатории Касперского в 2006 году было зафиксировано 86 880 уникальных вредоносных объектов (скрипты, эксплойты, исполняемые файлы и т.д.), в 2015 году эта цифра составила 121 262 075 [3] [4].

Одновременно с количеством эволюционирует и качество угроз. Хакерская деятельность все больше приобретает черты организованной преступности [2]. Число вредоносных объектов, которые обнаруживаются в сети ежегодно, исчисляется миллиардами, их распространение ведется более чем со 100 миллионов интернет адресов [5]. Каждый год их число увеличивается на 40%. Атаки в информационном пространстве наносят ущерб, который оценивается в сотни миллиардов долларов [6]. По заявлению начальника Бюро специальных технических мероприятий МВД России Алексея Мошкова каждую секунду 12 человек на Земле становятся жертвами киберпреступников. Только в России удалось предотвратить хищение около 1 миллиарда рублей с банковских счетов граждан [7].

Эти тенденции призывают системных администраторов уделять особое внимание информационной безопасности.

1.2 Информационная безопасность

Безопасность информации — это свойство (состояние) передаваемой, накапливаемой, обрабатываемой и хранимой информации, характеризующее ее степень защищенности от дестабилизирующего воздействия внешней среды (человека и природы) и внутренних угроз, то есть ее конфиденциальность (секретность, смысловая или информационная скрытность), сигнальная скрытность (энергетическая и структурная) и целостность — устойчивость к разрушающим, имитирующим и искажающим воздействиям и помехам [8].

Базовая система безопасности строится на четырех основных принципах безопасности: учет, конфиденциальность, целостность и доступность [9].

- Учет — сохранение и обработка отчетов по всем событиям и операциям, которые произошли в инфраструктуре ЦОД.

- Конфиденциальность — обеспечение необходимой секретности информации и предоставление доступа только авторизованным пользователям.
- Целостность — неизменность информации. Целью данной службы является предотвращение неавторизованного изменения или разрушения информации.
- Доступность — означает надежный и своевременный доступ для авторизованных пользователей.

Угрозы — это потенциальные атаки, которые могут быть выполнены на инфраструктуру. Такие атаки можно классифицировать как активные или пассивные. Пассивные атаки являются попытками получить неразрешенный доступ к системе. Они ставят под угрозу конфиденциальность информации. Активные атаки означают модификацию данных, отказ в обслуживании, и атаки сокрытия авторства. Они ставят под угрозу целостность и доступность информации.

Область атаки, вектор атаки и показатель трудозатрат — это три фактора которые следует учитывать при оценке меры уязвимости окружения для угроз безопасности. Область атаки относится к различным точкам входа, которые атакующий может использовать для запуска атаки. Вектор атаки — это шаг или серия шагов, необходимых для выполнения атаки. Показатель трудозатрат означает сумму времени и усилий, необходимую для использования вектора атаки.

Идеальная безопасность недостижима, так как в модели безопасности практически любой системы есть несколько фундаментальных изъянов, которые невозможно преодолеть.

Программное обеспечение ориентировано, прежде всего, на удобство применения, что отнюдь не предполагает естественность и простоту защиты. Концепция разработки и применения современного программного обеспечения заключается в обеспечении удобного манипулирования данными в сетевой многопользовательской среде.

Программное обеспечение разрабатывается большим сообществом программистов. Все они имеют разную квалификацию, по-разному относятся к своей работе и обладают различными знаниями о строении операционной системы и ее особенностях. Поэтому даже самые современные средства зашиты, выпущенные с самыми благими намерениями, могут приводить к появлению новых уязвимостей [1].

Для создания абсолютно непробиваемой защиты придется изолировать компьютер от всех устройств доступа (и, возможно, поместить его в специальную комнату, стены которой не пропускают электромагнитное излучение). Однако, попытаться изолировать информацию от лиц, для которых она не предназначена, можно не прибегая к столь кардинальным методам. Эту задачу, с разной степенью успешности, помогает решить шифрование. Здесь нужно понимать, что шифрование — это лишь инструмент в руках человека и только лишь шифрованием данных обеспечить безопасность не получится. Как написал в соей книге Брюс Шнайер — автор нескольких бестселлеров и признанный специалист в области безопасности и защиты информации: «Безопасность – это цепь: где тонко, там и рвется» и «Безопасность – это процесс, а не продукт» [10].

Эффективность безопасности может быть измерена двумя критериями. Во-первых, стоимость применения системы должна составлять лишь небольшую долю ценности защищаемой информации. Во-вторых, потенциальный атакующий должен затратить больше времени и денег, чтобы испортить систему, в сравнении со стоимостью защищенной информации [9].

1.3 Шифрование и аутентификация

Вопросами шифрования, дешифрования и расшифровки занимается наука под названием криптология. Криптография — раздел криптологии отвечающий за изобретение шифров. Искусство взлома шифров называется

криптоанализом. Сообщения, подлежащие зашифровке, называемые открытым текстом, преобразуются с помощью функции (алгоритма), вторым входным параметром которой является ключ. В 1883 году фламандским венным криптографом Аугустом Керкгофом был высказан следующий принцип: алгоритмы шифрования общедоступны; секретны только ключи [11].

Этот принцип, хоть и далеко не сразу, но получил широкое применение в современной криптографии. В январе 1997 года Национальный институт стандартов и технологий (NIST — National Institute of Standards and Technology) — агентство Министерства торговли, занимающееся разработкой стандартов для Федерального правительства США, — даже организовал открытый конкурс на лучший алгоритм для нового стандарта. Для представления своих разработок на этот конкурс были приглашены ученые со всего мира [11]. Все представленные на конкурсе алгоритмы были общедоступны, зашифрованные этими алгоритмами экземпляры открытого текста целенаправленно подвергались криптоанализу с целью выявить слабые стороны алгоритмов.

Различают симметричные и ассиметричные алгоритмы шифрования, либо, как их еще называют, алгоритмы с симметричным и открытым ключом соответсвенно. В первом случае для шифрования и дешифрования применяется один и тот же ключ, во втором существует открытый и закрытый ключи. Открытым ключом осуществляется шифрование данных, а закрытым дешифрование. Суть ассиметричных алгоритмов шифрования состоит в том, что из открытого ключа в разумные сроки невозможно получить закрытый ключ.

Для получения открытого и закрытого ключа используются однонаправленные функции, к примеру, умножение двух больших простых чисел. Получить произведение, перемножив числа, не трудно. При этом на сегодняшний день считается, что разложить произведение на множители и получить два больших простых числа, нелегко. К несчастью для проектировщиков алгоритмов, этот процесс становится всё легче. В 1976 году Ричард Гай (Richard Guy) писал: «Я был бы немало удивлен, если бы кто-нибудь научился разлагать на множители произвольные числа порядка 10^{80} в течение

данного столетия» [12]. В 1977 году Рон Ривест (Ron Rivest) заявил, что разложение на множители 125-разрядного числа потребует 40 квадриллионов лет [13]. В 1994 году было разложено на множители 129-разрядное число [14]. Из этого можно сделать следующие выводы: во-первых, предсказывать глупо; во-вторых, длину ключа стоит выбирать исходя из вычислительной мощности современных систем и существующих математических методов.

Вычислительная мощь обычно измеряется в mips-годах: годовая работа компьютера, выполняющего миллион операций в секунду. Разложение на множители ключа длиной 2048 бит при помощи алгоритма решета специального поля чисел занимает $4*10^{14}$ mips-лет. Эта длина ключа рекомендована для использования в корпоративных сетях на 2015 год [15].

Помимо шифрования данных криптография нашла применение в вопросах идентификации собеседников в сети — аутентификация. Для задач аутентификации широкое применение нашли ассиметричные алгоритмы шифрования [8]. Для целей аутентификации используется та же ключевая пара, что и для шифрования: открытый ключ, закрытый ключ. Однако, применяются они с точностью до наоборот. Шифрование производят закрытым ключом, а дешифрование открытым. Суть алгоритма аутентификации по ассиметричным ключам можно выразить следующим образом:

1. Клиент инициализирует подключение к серверу;
2. Сервер генерирует произвольную строку и отправляет её клиенту;
3. Клиент шифрует полученную от сервера строку своим закрытым ключом и отправляет получившееся сообщение серверу;
4. Сервер дешифрует сообщение открытым ключом клиента и сравнивает с отправленным ранее сообщением. Если сообщения равны, то клиент считается аутентифицированным.

Таким образом осуществляется подтверждение собеседника в сети. Идея заключается в том, что зашифровать сообщение таким образом, чтобы оно дешифровалось открытым ключом клиента, может только сам клиент, обладающий закрытым ключом. Очевидно, что прежде чем использовать

алгоритм аутентификации по ассиметричным ключам необходимо каким-либо образом передать на сервер открытый ключ клиента.

Глава 2 Исследование системы управления ЦОД в контексте информационной безопасности

ЦОД ТПУ представляет собой инфраструктуру виртуализации, которая построена с целью консолидировать вычислительные ресурсы и ресурсы хранения данных. Благодаря консолидации снижаются накладные расходы на обеспечение бесперебойного электропитания и теплоотведения, появляется возможность использовать оборудование более высокого класса надежности и функциональности. Вопросами администрирования ЦОД и большинства размещенных в нем виртуальных машин занимается структурное подразделение Главный информационный узел ТПУ (ГИУ). Помимо сотрудников ГИУ административный доступ к ряду виртуальных машин должны иметь сотрудники других структурных подразделений ТПУ, чьи информационные системы размещены на мощностях ЦОД.

В начале января 2016 года на мощностях ЦОД было размещено 575 виртуальных машин. Из них 260 машин под управлением различных версий Microsoft Windows и 315 под управлением различных дистрибутивов GNU/Linux. Подробная статистика по операционным системам виртуальных машин ЦОД приведена в таблице 1.

Таблица 1 — Виртуальные машины ЦОД в разрезе гостевых операционных систем

Гостевая операционная система	Количество, шт.
Microsoft Windows	260
CentOS Linux	210
Ubuntu Linux	52
Gentoo Linux	23
Debian Linux	23
Oracle Linux	6
SUSE Linux	1
Итого	575

Виртуальные машины под управлением Microsoft Windows по большей

части представляют собой клиентские машины, серверы домена, серверы корпоративной электронной почты, терминальные серверы с различным набором прикладного программного обеспечения. Все они включены в домен при помощи LDAP-совместимой службы каталогов Active Directory. Рассмотрение вопросов обеспечения информационной безопасности операционных систем семейства Microsoft Windows выходит за рамки данной работы.

Виртуальные машины ЦОД под управлением дистрибутивов GNU/Linux составляют основу, на которой размещены ключевые информационные системы ТПУ. В число таких ресурсов входят: службы динамической настройки узлов, службы доменных имен в зоне tpu.ru, основная база данных университета, сайты домена tpu.ru, корпоративный портал, системы электронного документооборота, облачное файловое хранилище, системы резервного копирования и многое другое. Виртуальные машины размещены в публичных и приватных сетях, в зависимости от назначения. Система управления этим сегментом ЦОД имела хаотичный самоорганизующийся характер.

Учетные данные администратора атомарной виртуальной машины определялись исполнителем в момент её создания. Зачастую всё сводилось к установке пароля администратора, который знали все сотрудники отдела. Не существовало механизма оперативной смены учетных данных с правами администратора на случай их компрометации.

Настройки удаленного доступа также определялись администратором в момент создания виртуальной машины и зачастую оставлялись «по умолчанию». При этом настройки «по умолчанию» имеют ряд проблем безопасности, к примеру, разрешен удаленный вход администратора под обезличенной учетной записью.

Обновления безопасности программного обеспечения устанавливались вручную после того как административному персоналу становилось известно о наличии уязвимостей в том или ином программном пакете. Установка обновлений безопасности на большое количество обслуживаемых узлов в ручном режиме требовала больших трудозатрат и неизбежно приводила к

ошибкам обусловленным человеческим фактором. Администратор или группа администраторов могла легко пропустить какой-либо узел. Тем самым оставив его уязвимым для атак злоумышленников.

Отсутствовал централизованный сбор журналируемой информации, т.е. журналы хранились только локально на самих виртуальных машинах. Таким образом потенциальный взломщик имел возможность скрыть последствия своего проникновения отредактировав соответствующие журналы.

На рисунке 1 изображен пример поведения администраторов и злоумышленника в условиях системы управления до работ по модернизации. ПК1 и ПК2 — это рабочие станции администраторов, которые имеют санкционированный доступ к управляемым узлам. Управляемые узлы на рисунке изображены элементами У1, У2 и У3. ПК3 — это рабочее место злоумышленника, атакующего управляемые узлы. Стрелками показаны соединения к управляемым узлам с примером передаваемых данных. Администраторы, работающие за ПК1 и ПК2, аутентифицируются на управляемых узлах используя учетную запись «root» и пароль «x». Злоумышленник эксплуатирует уязвимость программного обеспечения на У1, что приводит к выводу узла из строя, затем подбирает пароль к У2 и справившись удаляет следы из журнала, на У3 заходит уже зная учетные данные.

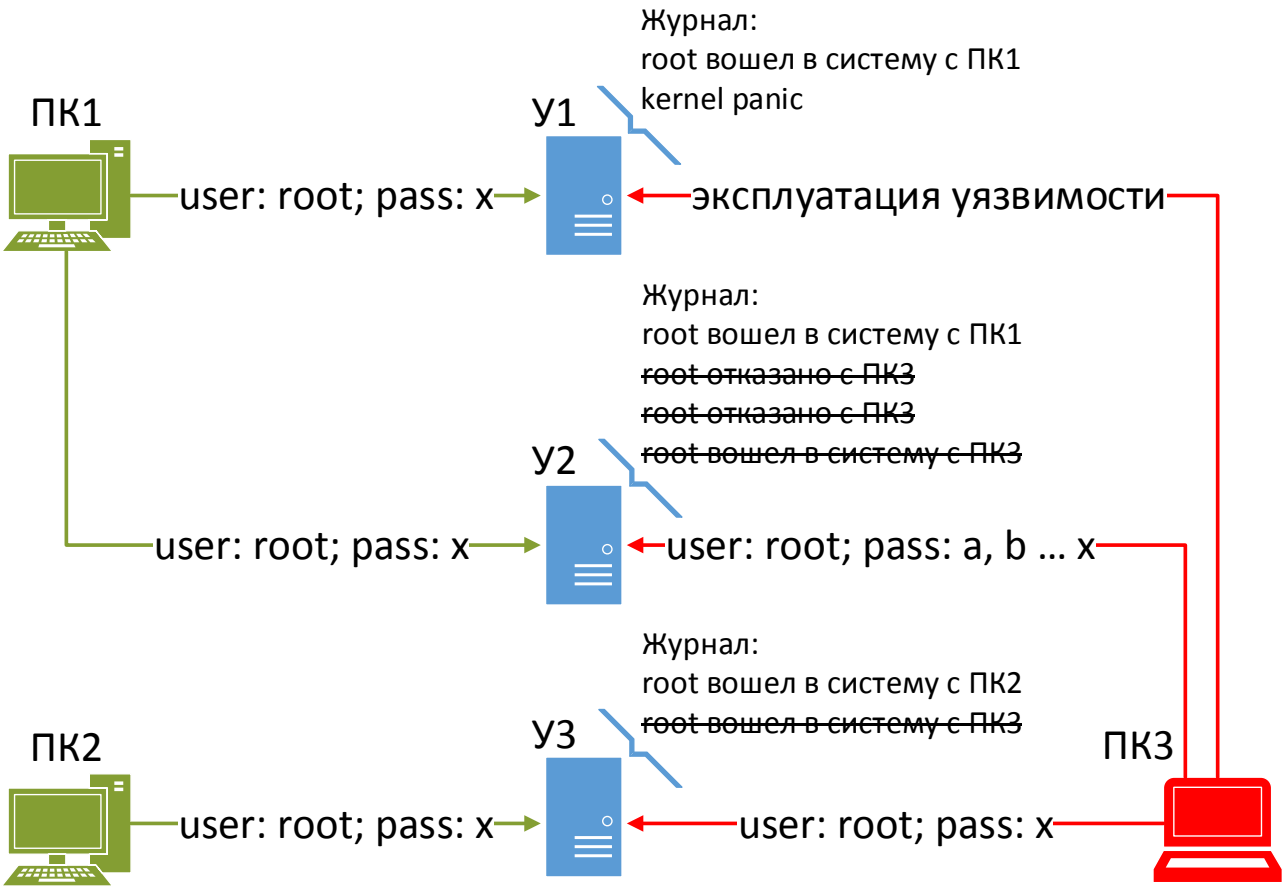

Рисунок 1 — Система управления до модернизации

Для формирования требований к системе управления инфраструктурой с учетом выявленных проблем результаты исследования были помещены в таблицу 2.

Таблица 2 — Проблемы выявленные в системе управления ЦОД

Проблема	Угроза
Аутентификация по паролю	Потенциальная возможность перехвата пароля — угроза всем четырем основным принципам безопасности
Использование обезличенной учетной записи	Невозможность отследить авторство внесенных изменений — угроза принципу учета

Проблема	Угроза
Отсутствие механизма оперативной смены учетных данных	Увеличивает время реакции при компрометации учетных данных
Отсутствие контроля за конфигурацией удаленного доступа	Могут быть включены «опасные» настройки, к примеру, уязвимые алгоритмы установления защищенного соединения, либо возможность входа под обезличенной учетной записью
Отсутствие защиты от атак подбора учетных данных	Вероятная компрометация учетных данных
Установка обновлений безопасности в ручном режиме	Увеличивает вероятность эксплуатации уязвимости программного обеспечения — угроза принципу доступности
Отсутствие централизованного сбора журналов	Невозможно анализировать ситуацию, успешная атака может остаться незамеченной — нарушение принципа учета
Использование сторонник источников обновлений	Вероятность подмены инсталляционного пакета программного обеспечения — нарушение принципа целостности

2.1 Область атаки атомарной виртуальной машины ЦОД

На основе информации собранной в таблице 2 была определена область атаки атомарной виртуальной машины под управлением операционной системы GNU/Linux.

- Эксплуатации уязвимостей программного обеспечения.

- Подмена инсталляционного пакета.
- Утечки учетных данных администратора.
- Подбор учетных данных администратора.

Эксплуатация уязвимостей программного обеспечения несет большую угрозу информационным системам. Этот тип атаки хорошо поддается автоматизации. После того как уязвимость становится известна, в сети Интернет появляются инструменты для её эксплуатации, в следствии чего резко снижается показатель трудозатрат на осуществление успешного взлома. Взломщику не приходится даже вникать в суть уязвимости, вектор атаки сводится к одному шагу — нацелить инструмент на систему, а всё остальное сделает за него программа. По данным CVE Community, сообщества стремящегося вести централизованный учет обнаруженных в программном обеспечении уязвимостей, в 2015 году было зарегистрировано более 5000 уязвимостей программного обеспечения [16]. Значительно снизить риск можно оперативным обновлением программного обеспечения. Идея заключается в том, что эксплуатировать неизвестные уязвимости значительно сложнее, чем известные.

Атака на информационную систему может быть осуществлена через подмену инсталляционного пакета с новым или обновляемым программным обеспечением. Злоумышленник может создать управляемую удаленно закладку, которая позволит ему проникнуть в систему. Для защиты от атак данного характера инфраструктура распространения программного обеспечения в популярных дистрибутивах GNU/Linux использует контрольные суммы и электронные подписи [17] [18].

Утечка учетных данных администратора может произойти в результате перехвата, в момент передачи по сети, методом подбора, либо подменой сервера аутентификации. Во всех трех случаях снизить вероятность утечки учетных данных позволяет криптография.

Для того чтобы защитить учетные данные от перехвата можно просто не передавать их по сети. В программном обеспечении OpenSSH, реализации протокола Secure Shell (SSH), эта техника используется в методе аутентификации

по открытым ключам, когда по сети передается только подпись, полученная шифрованием сообщения от сервера закрытым ключом администратора. Закрытый ключ администратора рекомендуется защитить парольной фразой, при этом ни парольная фраза, ни сам закрытый ключ в процессе аутентификации не покидают пределов рабочей станции администратора [19].

В случае успешной компрометации одной системы, к примеру, через уязвимость программного обеспечения, злоумышленник может перехватить пароль администратора, тем самым скомпрометированными окажутся все системы, на которые этот администратор имеет доступ. В случае использования обезличенной учетной записи, присутствующей на всех системах, скомпрометированными оказываются они все. Аутентификация по открытым ключам исключает компрометацию закрытого ключа даже в случае компрометации сервера аутентификации [19].

Защитить информационную систему от подбора учетных данных можно двумя способами: увеличив длину пароля, либо установив лимит на количество неудачных попыток аутентификации. При использовании аутентификации по ключам задача подбора сводится к получению открытого ключа и вычислению по нему закрытого. Здесь безопасность определяется длиной ключа и стойкостью алгоритма. Опираясь на выводы Брюса Шнайера (Bruce Schneier) примем, что оптимальная длина открытого ключа на сегодняшний день равна 2048 бит [15].

Точно так же как администратор аутентифицируется на сервере, сервер должен аутентифицироваться на рабочей станции администратора, предоставляя свой публичный ключ и сообщение зашифрованное закрытым ключом. Так как подразумевается, что закрытый ключ сервера неизвестен атакующему, то он не в состоянии предоставить администратору сообщение, дешифруемое публичным ключом оригинального сервера.

Глава 3 Проектирование системы управления инфраструктурой

Опираясь на выводы, сделанные выше были сформированы требования к системе управления. Эти требования относительно выявленных проблем собраны в таблицу 3.

Таблица 3 — Требования к системе управления инфраструктурой

Проблема	Решение
Аутентификация по паролю	Рассмотреть альтернативные методы аутентификации
Использование обезличенной учетной записи	Завести каждому администратору личную учетную запись
Отсутствие механизма оперативной смены учетных данных	Реализовать механизм оперативной смены учетных данных
Отсутствие контроля за конфигурацией удаленного доступа	Реализовать механизм контроля за конфигурацией удаленного доступа и возможность оперативного её изменения
Отсутствие защиты от атак подбора учетных данных	Реализовать механизм защиты сервисов аутентификации от атак подбора учетных данных
Установка обновлений безопасности в ручном режиме	Реализовать механизм позволяющий включать автоматическое обновление по требованию
Отсутствие централизованного сбора журналов	Реализовать централизованный сбор журналируемой информации

В дополнение к требованиям из таблицы 3, анализируя устройство существующей инфраструктуры, можно предъявить следующие требования к системе управления:

- система должна управляться централизованно;
- система должна поддерживать профили узлов и профили администраторов;
- система должна иметь возможность разделять пользователей на группы доступа;
- система должна иметь возможность управлять узлами, находящимися в приватных сетях;
- в случае отказа системы управления, управляемые узлы должны продолжать функционировать в штатном режиме;
- система должна поддерживать дистрибутивы построенные на базе операционной системы GNU/Linux из таблицы Таблица 1, т.е. использующие пакетную базу rpm, deb и исходные тексты;
- система должна быть бесплатной и основанной на программном обеспечении с открытым исходным кодом.

В результате в распоряжении структурного подразделения ГИУ должна быть система, отвечающая всем основным задачам, сформулированным в этом разделе.

3.1 Выбор типа аутентификации

Как уже было сказано выше, аутентификация по паролю имеет ряд недостатков, отрицательно влияющих на уровень безопасности. Ниже приведены альтернативные методы аутентификации доступные в свободной реализации протокола SSH программном пакете OpenSSH.

3.1.1 Аутентификация по протоколу GSSAPI

Аутентификация с использованием общего программного интерфейса

сервисов безопасности (GSSAPI). GSSAPI стандартизирован IETF и представляет собой набор стандартных интерфейсов при помощи которых обеспечивается совместимость между различными сервисами безопасности [20]. К примеру, используя GSSAPI возможно аутентифицироваться на сервере OpenSSH используя билет Kerberos от сервера Active Directory в домене Microsoft Windows. Безопасность данного метода аутентификации зависит от безопасности программного продукта OpenSSH, реализации GSSAPI, нижележащего метода обеспечивающего аутентификацию, к примеру Kerberos и всех протоколов, которые используют эти протоколы [21].

3.1.2 Аутентификация по принципу доверенных узлов

Если узел, с которой пользователь инициализирует подключение, указан в файле /etc/hosts.equiv или /etc/shosts.equiv на удаленном узле и имена пользователей совпадают на обоих узлах, либо если файлы ~/.rhosts или ~/.shosts присутствуют в домашней директории пользователя на удаленном узле и содержат имя узла, с которого пользователь инициирует подключение, и имя пользователя этой машины, то пользователь считается аутентифицированным. Дополнительно удаленный узел должен проверять ключ узла клиента в файле /etc/ssh/ssh_known_hosts или ~/.ssh/known_hosts для разрешения входа. Это дополнение закрывает угрозу атаки подменой IP, подменой DNS, либо подменой маршрута [22].

3.1.3 Аутентификация по открытым ключам

Схема базируется на криптографии открытых ключей или, как её еще называют ассиметричной криптографии, когда шифрование и дешифрование осуществляется разными ключами. Идея таких ключей заключается в том, что должно быть невозможно в короткие сроки получить ключ дешифрования

(закрытый) из ключа шифрования (открытый). Сервер должен знать открытый ключ и только аутентифицирующийся пользователь должен знать закрытый ключ. OpenSSH поддерживает следующие алгоритмы ассиметричного шифрования: DSA, ECDSA, Ed25519 и RSA [22]. Однако, начиная с версии OpenSSH 7.0 алгоритм DSA отключен в ввиду его недостаточно стойкости [23].

Подтверждения уязвимости алгоритма DSA найти не удалось. В более ранних реализациях OpenSSH, к примеру OpenSSH-6.9p1, в реализации утилиты ssh-keygen, которая отвечает за генерацию ключей шифрования, можно найти следующий участок кода:

```
1.    maxbits = (type == KEY_DSA) ?
2.        OPENSSL_DSA_MAX_MODULUS_BITS : OPENSSL_RSA_MAX_MODULUS_BITS;
3.    if (*bitsp > maxbits)
4.            fatal("key bits exceeds maximum %d", maxbits);
5.    if (type == KEY_DSA && *bitsp != 1024)
6.            fatal("DSA keys must be 1024 bits");
```

OPENSSL_DSA_MAX_MODULUS_BITS это константа из заголовочного файла библиотеки OpenSSL, которая равна 10000. Таким образом, первые четыре строки проверяют не превышает ли указанная в процессе длина ключа значения в 10000 бит. При этом следующие две строки строго ограничивают длину ключа в 1024 бит. Такой противоречивый код может служить результатом того, что он был написан в разное время, а возможно и разными людьми. Ограничение длины DSA ключа в 1024 бит было установлено стандартом FIPS 186-2, который был опубликован Национальным институтом стандартов и технологий США 27 января 2000 года [24]. Однако, с того времени было выпущено еще две редакции данного стандарта. В настоящий момент, на май 2016 года последняя версия стандарта FIPS 186-4 регламентирует использовать длину DSA ключа в 1024, 2048 и 3072 бит [25]. Можно предположить, что уязвимой является реализация DSA именно в программном продукте OpenSSH.

Алгоритм Ed25519 является довольно новым для проекта OpenSSH. Этот алгоритм был добавлен в версии OpenSSH 6.5, которая вышла 29 января 2014 года [26]. Этот факт не позволяет нам использовать данный алгоритм, так как в

инфраструктуре ЦОД присутствуют дистрибутивы операционной системы GNU/Linux которые получают только обновления безопасности и в которых версия OpenSSH никогда не будет обновлена до 6.5.

Алгоритмы ECDSA был добавлен в версии 5.7, которая вышла 23 января 2011 года [27]. Однако и этой версии нет, к примеру, в CentOS 5, виртуальные машины на основе которой еще используются в инфраструктуре ЦОД.

Наиболее широко распространенным алгоритмом является RSA. На май 2016 года он всё еще считается стойким при использовании длины ключа в 4096 бит [15]. RSA имеет проблемы быстродействия из-за довольно большой длины ключа, по сравнениями с алгоритмами основанными на эллиптических кривых ECDSA и Ed25519. Однако этот недостаток не является существенным, так как ассиметричные алгоритмы используются только на этапе установления защищенного соединения и в процессе аутентификации. Всё остальное время сессия шифруется симметричным алгоритмом, к примеру, таким как AES [19].

Учитывая всё вышесказанное, для использования в системе управления, был выбран метод аутентификации по открытым ключам (Public key authentication) и алгоритма ассиметричного шифрования RSA с длиной ключа в 4096 бит.

3.2 Выбор способа распространения учетных записей

Проблемой под номером 2 требований к реализуемой системе управления из таблицы 3 являлось использование администраторами обезличенной учетной записи на всех управляемых узлах. Было решено завести каждому администратору личную учетную запись. Данное решение потребовало выработать механизм распространения учетных записей между управляемыми узлами. Были рассмотрены службы каталогов узлов NIS, LDAP и службы управления конфигурациями узлов Puppet, Chef, Ansible и Salt.

3.2.1 Network Information Service

Network Information Service (NIS) представляет собой клиент-серверный каталог для распространения между узлами такой информации как учетные данные пользователей и имена узлов. NIS был разработан компанией Sun Microsystems и изначально назывался Yellow Pages по аналогии с бумажным справочником, в котором перечисляются телефонные номера, но, из-за судебных преследований владельцами торговой марки, был переименован в NIS [28].

NIS является довольно старой разработкой, его улучшенная версия NIS+ была выпущена всё той же Sun Microsystems в 1992 году как часть операционной системы Solaris 2. В 2012 году компания Oracle, ранее купившая Sun Microsystems, объявила о прекращении поддержки NIS+. Начиная с версии Solaris 11 данный программный продукт исключен из операционной системы [29].

NIS можно сконфигурировать в качестве сервера аутентификации в сети. Однако, функционала данной службы недостаточно для распространения публичных ключей администраторов. Следовательно, необходимо разрабатывать дополнительный механизм для распространения публичных ключей администраторов.

NIS не поддерживает защиту данных на уровне сети. Вся передаваемая между клиентом и сервером информация проходит по сети в открытом виде.

Учитывая всё вышеперечисленное, был сделан вывод о том, что данная система не подходит в качестве механизма распространения учетных записей.

3.2.2 Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) предоставляет доступ к службе каталогов X.500. X.500 набор стандартов разработанных международным

консультационным комитетом по телефонии и телеграфии (МККТ, ITU-T) для упорядочивания и обеспечения взаимной совместимости каталогов описывающих объекты вычислительных сетей, такие как пользователи, компьютеры, почтовые адреса, сервисы, группы и многое другое [30]. LDAP разработан и стандартизирован IETF, как облегченный вариант разработанного ITU-IT протокола DAP. LDAP — позволяет производить операции аутентификации (bind), поиска (search) и сравнения (compare), а также операции добавления, изменения или удаления записей [31].

LDAP является широко используемым стандартом доступа к службам каталогов. Из свободно распространяемых открытых реализаций наиболее известен сервер OpenLDAP, из проприетарных — поддержка протокола имеется в Active Directory — службе каталогов от компании Microsoft, предназначенной для централизации управления сетями Windows.

LDAP поддерживает доступ к данным каталога по защищенному каналу связи. Узлы под управлением дистрибутивов GNU/Linux поддерживают аутентификацию с использованием сервера LDAP. OpenSSH можно настроить на получение публичных ключей с сервера LDAP.

Несмотря на всё вышеперечисленное данный механизм был отвергнут в силу того, что использование сервера или кластера LDAP совместимых серверов привносят потенциальную угрозу доступности в систему управления узлами. Очевидно, что при проблемах коммуникации управляемого узла и сервера LDAP, администраторы будут испытывать проблемы аутентификации.

3.2.3 Puppet

Puppet — это система автоматизации распространения конфигурации между узлами под управлением операционными системами GNU/Linux, Unix и Windows. Puppet позволяет выполнять административные задачи, такие как добавление пользователя, установка пакетов, изменение конфигурационных

файлов на основе централизованного описания.

Конфигурация узлов пишется на сервере (puppet master) на специальном декларативном предметно-ориентированном языке. По своей структуре язык похож на Ruby. На узлах устанавливается агент (puppet agent), который периодически опрашивает сервер для получения обновлений конфигурации.

Поддерживаются два механизма, обеспечивающих защиту системы на уровне сети передачи данных.

- Первый механизм, используемый по умолчанию, только подписывает передаваемые данные, но не шифрует их. Подпись производится SSL сертификатом, который генерируется для каждого узла в процессе подключения его к системе. Таким образом обеспечивается аутентификация всех передаваемых между сервером и агентом сообщений и их учет;
- Второй механизм, предполагает шифрования всех передаваемых данных симметричным алгоритмом шифрования AES, а процедуру аутентификации обеспечивает ассиметричная ключевая пара RSA [32].

3.2.4 Chef

Chef — это, как и Puppet, система автоматизации распространения конфигурации между узлами. в нем также имеется сервер и агенты, установленные на управляемых узлах. В дополнение к серверу, установка Chef также требует рабочей станции, для управления им. Агенты могут быть установлены с рабочей станции с помощью утилиты knife, которая использует протокол SSH для подключения к узлам. После этого, управляемые узлы аутентифицируются на сервере при помощи SSL сертификатов [33].

Конфигурация Chef тесно связана с системой управления версиями Git, поэтому знание того, как работает Git необходимо для работы. Так же, как и

Puppet, Chef основан на ruby, поэтому потребуется и знание этого языка. Как и в случае с Puppet. Модули могут быть загружены или написаны «с нуля», после чего установлены на управляемые узлы, в соответствии с требуемыми настройками [34].

3.2.5 Ansible

Ansible имеет отличную от Puppet и Chef архитектуру. Ansible не требует установки агентов на управляемые узлы. Взаимодействие сервера и управляемых узлов осуществляетося с использованием протокола SSH. В Puppet и Chef агенты, установленные на узлах, инициализируют подключение к серверу и получают обновления конфигурации. В Ansible подключение инициализирует сервер, применяя описанную в сценарии конфигурацию к набору узлов. Ansible написан на Python, в отличие от Puppet и Chef, основанных на Ruby. Это позволяет управлять конфигурацией узлов, где присутствует Python, без установки какого-либо дополнительного программного обеспечения [35].

Установка Ansible может быть выполнена путем клонирования Git-репозитория на машину с которой будет осуществляется управление, либо из репозитория дистрибутива. Ansible использует протокол SSH для взаимодействия с узлами. Следовательно, все вопросы безопасности на уровне сетевого взаимодействия решает SSH. Ansible, вместо стандартного OpenSSH, может использовать Paramiko — реализацию протокола SSH на языке Python.

Ansible может быть запущен из командной строки без использования конфигурационных файлов для простых задач, таких как проверка, что некий сервис запущен, или для обновления триггеров и перезагрузки. Для более комплексных задач, конфигурационные файлы создаются с помощью YAML и называются «сценарии» (playbook). В них могут быть использованы шаблоны для расширения функциональности [36].

3.2.6 Salt

Salt схож с Ansible по своей архитектуре. Он использует метод push для связи с клиентами. Он может быть установлен через Git или через систему управления пакетами на сервере и клиентах. Клиент делает запрос к головному серверу, и, если тот дает разрешение, позволяет управлять данным узлом с помощью агента (в терминах Salt — minion).

Salt может связываться с клиентами по протоколу SSH, но масштабируемость значительно расширяется за счет клиентских агентов. Также, Salt включает асинхронный файловый сервер для ускорения обслуживания агентов, позволяя создавать хорошо масштабируемые системы.

Как и в случае Ansible, есть возможность отдавать команды, такие как как запуск сервисов или установка пакетов агентам напрямую из командной строки, или использовать конфигурационные файлы в формате YAML (state), для обработки комплексных задач. Также есть централизовано размещенные наборы данных (pillar) к которым имеют доступ конфигурационные файлы во время работы.

Есть возможность получать информацию о конфигурации — такую как версия ядра или детальную информацию о сетевом интерфейсе — напрямую от агентов через командную строку. Агенты могут также задаваться через использование элементов инвентаря, называемых «зернами» (grain), позволяющими легко передавать команды на определенные сервера, безотносительно к настроенным группам. Например, одной командой можно отправить запрос к агентам, расположенным на серверах с определенной версией ядра [37].

Большим достоинством систем управления конфигурациями то, что в них описывается желаемое состояние узла, а не вектор достижения этого состояния, как при написании скриптов автоматизации на bash, perl или python. То есть описывая конфигурацию при помощи, к примеру, puppet мы задаем желаемое

состояние узла и нам не важно какие шаги для этого необходимо сделать системе. При написании своих скриптов автоматизации администратор наоборот пишет ряд шагов, которые, по его мнению, должны привести систему в желаемое состояние. Данный подход, во-первых, требует хорошей проверки как самих шагов, так и их взаимного расположения, во-вторых, может быть в корне ошибочным, если администратор недостаточно хорошо представляет начальное состояние узла.

Для более наглядного обзора механизмов распространения учетных записей, плюсы и минусы каждой рассмотренной системы были сведены в таблицу 4.

Таблица 4 — Выбор способа распространения учетных записей

Критерий\Механизм	NIS	LDAP	Puppet	Chef	Ansible	Salt
Безопасность	0	5	5	5	5	5
Децентрализация	0	0	4	4	4	4
Взаимодействие с узлами	3	3	3	3	0	0
Документация	1	2	2	1	2	1
Итого	4	10	14	13	11	10

Для построения таблицы *Таблица 4* были определены критерии по которым определялось соответствие механизма поставленным задачам. Критерии были расположены в порядке уменьшения их значимости для данной работы.

Наиболее значимым критерием является критерий безопасности. Механизмы, отвечающие современным требованиям к безопасности, получали 5 баллов, не отвечающие 0 баллов.

Под децентрализацией имеется ввиду единой точки отказа, которая могла бы привести к ситуации, когда администратор не имеет возможность аутентифицироваться установленным способом. Балл за выполнение этого требования равен 4 баллам.

Взаимодействие с узлами, т.е. как управляемые узлы подключаются к серверу управления. Если подключение инициирует сервер — 0 баллов, если управляемый узел — 3 балла.

Доступность и качество написания документации. Максимальный балл 2.

Из таблицы Таблица *4* видно, что для использования в данной работе в лучше мере подходит проект Puppet. Стоит отметить, что программный продукт Puppet, помимо распространения учетных записей между узлами, позволяет реализовать и ряд других требований, сформулированных в начале этой главы. Такие как механизм оперативной смены учетных данных, механизм контроля за конфигурацией удаленного доступа и возможность оперативного её изменения, механизм, позволяющий включать автоматическое обновление по требованию.

3.3 Выбор способа защиты от атак подбора учетных данных

Выбранная в пункте 3.1 модель аутентификации по открытым ключам исключает возможность осуществления атаки подбором учетных данных. Однако, на этапе проектирования было предположено, что ряд управляемых узлов не будут исключать возможность аутентификации по паролю, к примеру, когда кому-либо потребуется доступ к узлу под учетной записью не обладающей административными полномочиями.

3.3.1 Netfilter

Netfilter является стандартным в среде операционных систем GNU/Linux межсетевым экраном. Он включен в ядро Linux начиная с версии 2.4 и постоянно эволюционирует. Другое более распространенное название Netfilter — iptables. Iptables это утилита при помощи которой администратор может манипулировать правилами Netfilter.

OpenSSH обеспечивает доступ к узлу посредством сервиса sshd, который

в конфигурации по умолчанию прослушивает порт 22 по протоколу TCP. Каждое соединение по протоколу TCP начинается с пакета, в котором установлен флаг — SYN, а флаг ACK опущен [11]. Такой пакет однозначно обозначает начало TCP соединения. Таким образом, ограничивая число подобных пакетов в определенный промежуток времени (средствами Netfilter) и ограничивая максимальное количество попыток аутентификации за одно соединение (средствами OpenSSH), мы можем существенно усложнить либо свести на нет попытки подбора учетных данных.

Однако, у этого способа есть существенный недостаток. Это угроза доступности. Потенциальный злоумышленник может осуществить атаку на систему, подменив адрес отправителя в TCP пакете, к примеру, на адрес машины администратора. Тем самым спровоцировав разрыв соединения администратора и последующую неспособность подключиться.

3.3.2 Fail2ban

Fail2ban является приложением, которое написано на языке Python. Основная идея заключается в том, что специальный процесс регулярно перечитывает журналы и сверяет каждую новую строку журнала с регулярным выражением. Регулярное выражение описывает неудачную попытку аутентификации. Если количество неудачных попыток аутентификации от определенного источника за определенное время превышает максимально допустимое значение, то совершается действие исключающее повторные попытки аутентификации этим источником. К примеру, в Netfilter добавляется правило запрещающее всякое взаимодействие с указанным IP-адресом.

Модель работы Fail2ban исключает угрозу доступности описанную в пункте 3.3.1. Для осуществления полноценной попытки аутентификации необходимо, чтобы TCP соединение было полностью установленным (ESTABLISHED). Установка соединения в протоколе TCP предполагает

прохождения тройного рукопожатия. Инициирующая сторона отправляет TCP пакет с установленным флагом SYN, опущенным флагом ACK и значением x флага SEQ. Узел, к которому осуществляется подключение, отвечает на данный пакет пакетом с установленным флагом SYN, ACK равным x + 1 и SEQ равным y. Инициирующая сторона должна ответить пакетом с опущенным флагом SYN, SEQ равным x + 1 и ACK равным y + 1. Соединение считается установленным только когда выполнены все три вышеописанных шага [11]. Это означает, что в случае подмены адреса отправителя, соединение установлено не будет и активность потенциального злоумышленника не отразится в журнале sshd, а значит не приведет ни к каким действиям на узле.

Принимая во внимание особенности работы Netfilter и Fail2ban, описанные в пунктах 3.3.1 и 3.3.2 предпочтение было отдано проекту Fail2ban, для целей реализации механизма защиты от атак подбора учетных записей.

3.4 Выбор системы сбора журналируемой информации

Стандарты syslog берут своё начало в проекте Sendmail, а развитие получили как часть проекта Калифорнийского университета в Беркли — Berkeley Software Distribution (BSD) при создании TCP/IP [38] [39].

Первая попытка стандартизировать протокол была предпринята Инженерным советом Интернета (Internet Engineering Task Force, IETF) в августе 2001 года. Тогда был выпущен информационный бюллетень RFC3164: The BSD syslog Protocol. Публикация имела информационный характер и давала ряд рекомендаций при имплементации решений для журналирования информации. Однако, не накладывала на разработчиков четких ограничений. Затем, в ноябре 2001 года, IETF выпустил стандарт RFC3195: Reliable Delivery for syslog, в котором имелось несколько ссылок на информационный бюллетень RFC3164: The BSD syslog Protocol. Эти ссылки позволили считать стандарт RFC3195

расширением концепта из RFC3164. В конечно итоге это породило много проектов, заявляющих о соответствии с RFC3195, но при этом совершенно не совместимых между собой. Стандартизация провалилась [40].

Вторая попытка стандартизировать протокол Syslog была предпринята IETF в марте 2009. Тогда был опубликован стандарт RFC5424: The Syslog Protocol и три его расширения: защита передаваемых данных при помощи TLS (RFC5425: Transport Layer Security (TLS) Transport Mapping for Syslog), передача журналируемых сообщений поверх UDP (RFC5426: Transmission of Syslog Messages over UDP) и соглашения для представления текстовых полей Facility и Severity кодами в протоколе SNMP (RFC5428: Textual Conventions for Syslog Management) [41] [42] [43] [44]. Эти публикации возымели успех. Позже было опубликовано еще несколько стандартов расширяющих протокол дополнительными функциями, такими как: подпись журналируемых сообщений (RFC5848: Signed Syslog Messages), защита журналируемой информации передаваемой поверх UDP протоколом DTLS (RFC6012: Datagram Transport Layer Security (DTLS) Transport Mapping for Syslog), передача журналируемой информации поверх TCP (RFC6587: Transmission of Syslog Messages over TCP) [45] [46] [47].

В процессе выбора системы сбора журналируемой информации были рассмотрены две системы, наиболее широко распространенные в семействе дистрибутивов операционной системы GNU/Linux: Rsyslog и Syslog-ng. Все они основаны на протоколе syslog, что должно обеспечить совместимость на уровне представления журналируемой информации.

3.4.1 Rsyslog

Rsyslog — это программный продукт с открытым исходным кодом предназначение которого заключается в сборе, преобразовании и передаче журналируемой информации представленной в соответствии с протоколом

rsyslog. Rsyslog эволюционировал из проекта sysklogd — оригинального проекта системы журналирования разрабатываемой в рамках операционной системы BSD. Ключевые возможности Rsyslog:

- модульная архитектура;
- мультипоточность;
- передача сообщений через UNIX-socket, UDP и TCP;
- защита соединения при помощи SSL и TLS;
- перенаправление сообщений в системы управления базами данных MySQL, PostgreSQL и Oracle и многие другие;
- фильтрация по любой части журналируемого сообщения;
- полностью настраиваемый выходной формат
- поддерживается режим совместимости с RFC3195 и RFC5424.

В настоящий момент Rsyslog является системой журналирования используемой по умолчанию в таких дистрибутивах операционной системы GNU/Linux как:

- Debian Linux 6 “Squeeze”, 7 “Wheezy”, 8 “Jessie”;
- CentOS Linux 5, 6, 7;
- Red Hat Enterprise Linux 5, 6, 7;
- Ubuntu 12.04 LTS “Precise Pangolin”, 14.04 LTS “Trusty Tahr”, 15.04 “Vivid Vervet”;
- Fedora 22, 21.

Создатель проекта Rsyslog Rainer Gerhards участвовал в работе над актуальной в данный момент серией стандартов IETF посвященных протоколу syslog [40].

3.4.2 Syslog-ng

Syslog-ng (Syslog next generation) имеет такое же назначение и схожий функционал, что и Rsyslog. Syslog-ng разрабатывается компанией BalaBit IT

Security. Выпускается две модификации Syslog-ng: свободная Open Source Edition и коммерческая Premium Edition. Основной отличительной особенностью проекта Syslog-ng является синтаксис написания конфигурационных файлов, а именно правил получения, фильтрации и передачи журналируемых сообщений. Язык написания конфигурационных файлов имеет объектно-ориентированную структуру. Объектами являются источники журналируемых сообщений, сами сообщения, фильтры и назначения [48].

До появления проекта Rsyslog в 2004 году, Syslog-ng использовали многие дистрибутивы GNU/Linux, такие как SLES, Gentoo Linux, Arch Linux. Однако, отказались от него в пользу Rsyslog. В 2008 году в рассылке debian-devel@lists.debian.org был поднят вопрос о смене системы журналирования по умолчанию с sysklogd на Syslog-ng или Rsyslog. В пользу второго высказалось сразу несколько разработчиков. Основными доводами в пользу Rsyslog были: поддержка конфигурационных файлов формата sysklogd без дополнительных модификаций, разработка в соответствии с принципами свободного программного обеспечения. Главным недостатком Syslog-ng был тот факт, что многие функции доступны только в коммерческой версии, а доработки и исправления, предлагаемые сообществом, принимались только после подписания разработчиком соглашения о передаче всех прав на код и зачастую включались в первую очередь в коммерческую версию [49].

В силу популярности проекта Rsyslog среди основных дистрибутивов операционной системы GNU/Linux и большей открытости, для реализации системы централизованно сбора журналируемой информации был выбран проект Rsyslog.

3.5 Выбор дистрибутива операционной системы

В 1987 году Эндрю Таненбаум написал первую версию операционной

системы MINIX для иллюстрации к своей книге «Операционные системы: Разработка и реализация». MINIX в то время была выпущена под лицензией ограничивающей её использование только для образовательных целей. Линус Торвальдс будучи студентом Хельсинского университета основываясь на ядре MINIX разработал ядро Linux, первая версия которого вышла 5 октября 1991 года. К тому моменту проект Ричарда Столлмана GNU уже содержал солидный набор приложений, составляющих пользовательское окружение операционной системы. Единственно чего не хватало GNU — это ядра. Таким образом появилась операционная система GNU/Linux.

По данным сайта distrowatch.org, на 30 мая 2016 года насчитывалось 279 активно развивающихся дистрибутивов операционной системы GNU/Linux, а всего их насчитывается более 800 [50].

Большая часть дистрибутивов GNU/Linux берет свое начало от:

- Debian Linux;
- SLS, в последствии Slakware Linux
- Red Hat Linux, позднее переименованный в Red Hat Enterprise Linux (RHEL).

Популярным проектом на основе Debian Linux является Ubuntu, а на базе Red Hat Linux основаны дистрибутивы Fedora и CentOS Linux.

При этом есть и не малое число самостоятельных проектов, наиболее известные из них:

- Enoch, в последствии Gentoo Linux
- Arch Linux [51].

Выбор стоял между Gentoo Linux, Debian Linux и CentOS Linux, так как у автора данной работы имелось больше опыта работы именно с этими тремя дистрибутивами.

3.5.1 Gentoo Linux

Gentoo Linux развился из проекта Enoch Linux Дэниэля Робинса (Daniel Robbins). Официальный поддерживаемый список, на май 2016 года, насчитывает 18,999 пакетов программного обеспечения [52], помимо официального существует большое количество пользовательских, которые может создать любой желающий. Пользовательские списки называются оверлеями. Оверлеи насчитывают 46398 пакетов программного обеспечения [53];

В Gentoo Linux отсутствуют регулярные выпуски. Дистрибутив обновляется постоянно. С одной стороны, это избавляет от необходимости раз в 10 лет переносить решение на новую версию дистрибутива, как в случае с CentOS. С другой стороны, администраторы Gentoo Linux постоянно живут в процессе обновления. Если сравнивать с Debian, то обратно несовместимые обновления могут поступить только при смене стабильных выпусков, что случается примерно раз в три года.

Разработчиками проекта Gentoo создан мощнейший инструмент управления пакетами программного обеспечения — Portage. Его концептуальная идея очень похожа на порты FreeBSD. Все пакеты устанавливаясь проходят процедуру компиляция из исходных текстов, посредством системы USE-флагов и опций компиляции каждый программный продукт гибко настраивается под конкретную задачу, система слотов позволяет одновременно устанавливать несколько версий одного и того же программного обеспечения.

Система отслеживания уязвимостей Gentoo Linux Security Advisories (GLSA) — позволяет получать отчеты о наличии в системе уязвимых пакетов и исправлять проблемы безопасности в автоматическом режиме.

3.5.2 Debian Linux

Один из старейших и популярных дистрибутивов GNU/Linux [54]. Проект Debian имеет собственную инфраструктуру распространения пакетов программного обеспечения в формате .deb. На май 2016 стабильный релиз Debian

8 “Jessie” насчитывает более 43000 пакета программного обеспечения [55].

Самым главным достоинством Debian является его репутация в мире свободного программного обеспечения. Большое сообщество, отлаженная за многолетнюю историю и четко регламентированная система выпуска и поддержки дистрибутива располагают к тому чтобы использовать Debian в работе.

В среднем срок поддержки каждой версии дистрибутива составляет пять лет. С момента выпуска стабильного релиза и до релиза следующего исправления ошибок и обновления безопасности предоставляют разработчики, ответственные за пакеты и команда безопасности Debian. После того как выпускается новый стабильный релиз, еще в течение двух лет поддержку обеспечивает команда добровольцев, именуемая LTS [56].

Команда разработчиков Debian обеспечивает поддержку обновления между стабильными выпусками дистрибутива. Подобное обновление достаточно хорошо тестируется и документируется на этапе подготовки нового стабильного выпуска.

3.5.3 CentOS Linux

CentOS Linux основан на базе коммерческого RHEL. RHEL распространяется только между подписчиками, оплатившими поддержку дистрибутива, но компания Red Hat обязана публиковать исходные тексты тех программных продуктов, которые разрабатываются под лицензией GPL. Таким образом, сообщество CentOS имеет возможность компилировать приложения на основе исходных текстов RHEL. Компиляция пакетов выполняется в максимальном соответствии с RHEL, используются те же версии компиляторов и библиотек. Всё это позволяет достичь полной бинарной совместимости этих дистрибутивов, что, в свою очередь, позволяет заявлять о поддержке дистрибутивом CentOS Linux всех программных продуктов совместимых с

RHEL.

CentOS не обеспечивает поддержку обновлений между мажорными выпусками дистрибутива. При этом срок поддержки выпусков CentOS Linux составляет 10 лет. Подобный срок является очень хорошим показателем в индустрии.

На май 2016 выпуск CentOS Linux 7 содержит немногим более 9000 пакетов. Это существенно меньше чем в Debian и Gentoo. Однако, благодаря бинарной совместимости с RHEL, можно использовать пакеты из многих других RHEL-совместимых дистрибутивов — это сам RHEL, Fedora Linux, Scientific Linux, Oracle Linux и многие другие. Нужно отметить, что использование пакетов из других дистрибутивов может привести к взаимным конфликтам зависимостей программного обеспечения.

В качестве дистрибутива операционной системы GNU/Linux для реализации проекта был выбран дистрибутив Debian 8 “Jessie” архитектуры x86_64. Решающим качеством дистрибутива Debian перед Gentoo являются: простота поддержки за счет системы регулярных стабильных выпусков. Решающим качеством дистрибутива Debian перед CentOS является широкий выбор программного обеспечения и возможность обновления между стабильными выпусками.

3.6 Концептуальная модель разрабатываемой системы управления

Было принято решение построить систему управления инфраструктурой ЦОД ТПУ из двух узлов.

- Узел управления для осуществления административных операций, таких как: добавление пользователей на управляемые узлы, распространение публичных ключей, изменение конфигурации и обновление управляемых узлов.
- Узел журналирования для сбора журналируемой информации с узла управления и управляемых узлов.

Было решено не управлять узлом журналирования при помощи узла управления, разместить узел журналирования в приватной сети ЦОД, доступ извне приватной строго ограничить на уровне межсетевого экрана шлюза сети.

3.6.1 Узел управления

Виртуальная машина, на которую были возложены функции автоматизации администрирования системы управления инфраструктурой ЦОД или узел управления. На этом узле было запланировано разместить сервер Puppet и разработанные к нему модули. Модули были разделены по основному назначению:

- модуль настройки политик удаленного доступа;
- модуль пользователей и групп;
- модуль настройки политик обновления;
- модуль настройки журналирования.

Модуль настройки политик удаленного доступа определяет настройки OpenSSH с возможностью выбрать уровень безопасности для каждого управляемого узла. Под уровнем безопасности понимается строгость политик удаленного доступа, т.е. какие методы аутентификации будут доступны пользователям. Первый уровень безопасности предполагает аутентификацию только по публичным ключам. Второй уровень безопасности позволяет аутентифицироваться по паролю. Оба уровня запрещают аутентификацию под обезличенной учетной записью администратора. Дополнительно этот модуль

определяет конфигурацию Fail2ban для защиты от атак подбора учетных данных.

Модуль пользователей и групп содержит перечисление всех пользователей системы управления и их основные сведения, такие как: имя, учетная запись в домене ТПУ, публичный ключ, возможность повышать привилегия до административных. Пользователи соотнесены с группой, к которой они принадлежат. Пользователи включаются в ту или иную группу в соответствии с выполняемыми обязанностями. Модуль обеспечивает распространение учетных записей и публичных ключей по управляемым узлам.

Модуль настройки политик обновления задает источники обновлений программного обеспечения, обеспечивает возможность запускать обновление по требованию, а также отключать автоматическое обновление.

Модуль настройки журналирования активирует функцию отправления журналируемой информации на узел журналирования.

3.6.2 Узел журналирования

Виртуальная машина, на которую были возложены функции сбора журналируемой информации с управляемых узлов инфраструктуры ЦОД или узел журналирования. Было запланировано разместить на этом узле специальным образом сконфигурированный сервер Rsyslog. Назначение сервера Rsyslog — принимать журналируемую информацию от управляемых узлов по протоколу UDP, фильтровать и перенаправлять в базу данных СУБД MySQL. Было решено предусмотреть возможность доступа к сохраненной в базе данных журналируемой информации через web-интерфейс обеспечивающий быстрый поиск по базе данных.

Глава 4 Реализация системы управления инфраструктурой

Система управления инфраструктурой ЦОД была реализована в соответствии с требованиями из Глава 2 и концептуальной моделью, сформулированной в пункте 3.6. На рисунке 2 изображен пример поведения администраторов и злоумышленника в условиях системы управления после модернизации.

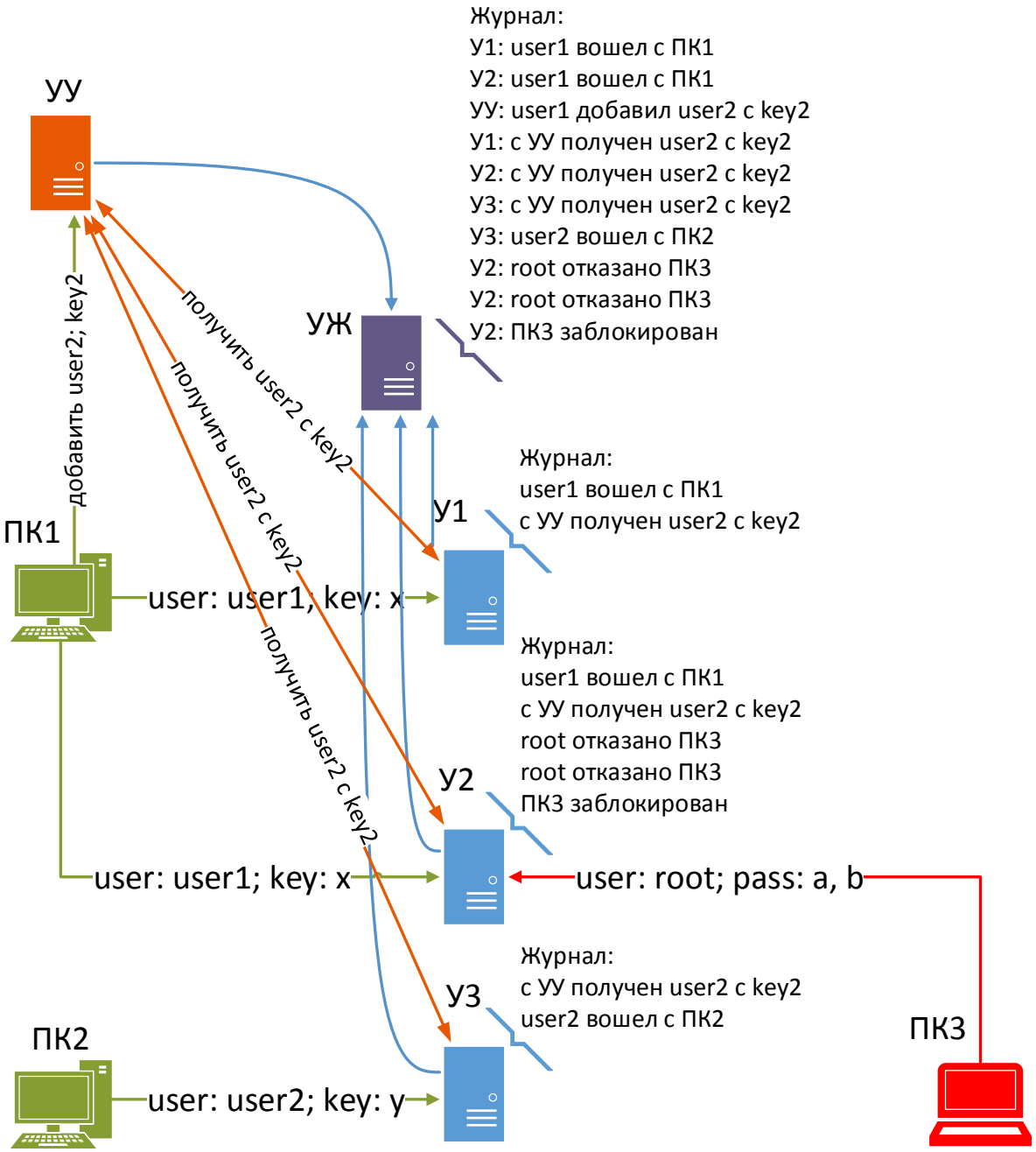

Рисунок 2 — Система управления после модернизации

Как видно из рисунка 2, администраторы аутентифицируются с ПК1 и ПК2 используя четко идентифицирующие их личные учетные записи и личные ключи. Администратор с именем пользователя user1 использует УУ для добавления учетной записи user2 на У1, У2 и У3. Для этого он изменяет конфигурацию этих узлов на УУ, а узлы получают обновление конфигурации при следующем обращении к УУ. Злоумышленник с ПК3 пытается осуществить атаку подбора учетных записей на У2. После нескольких неудачных попыток аутентификации он оказывается заблокированным. Все журналируемые события сохраняются в локальных журналах узлов и дублируются на УЖ.

4.1 Узел управления

Для размещения УУ была создана виртуальная машина суммарная информация о которой отображена на рисунке 3.

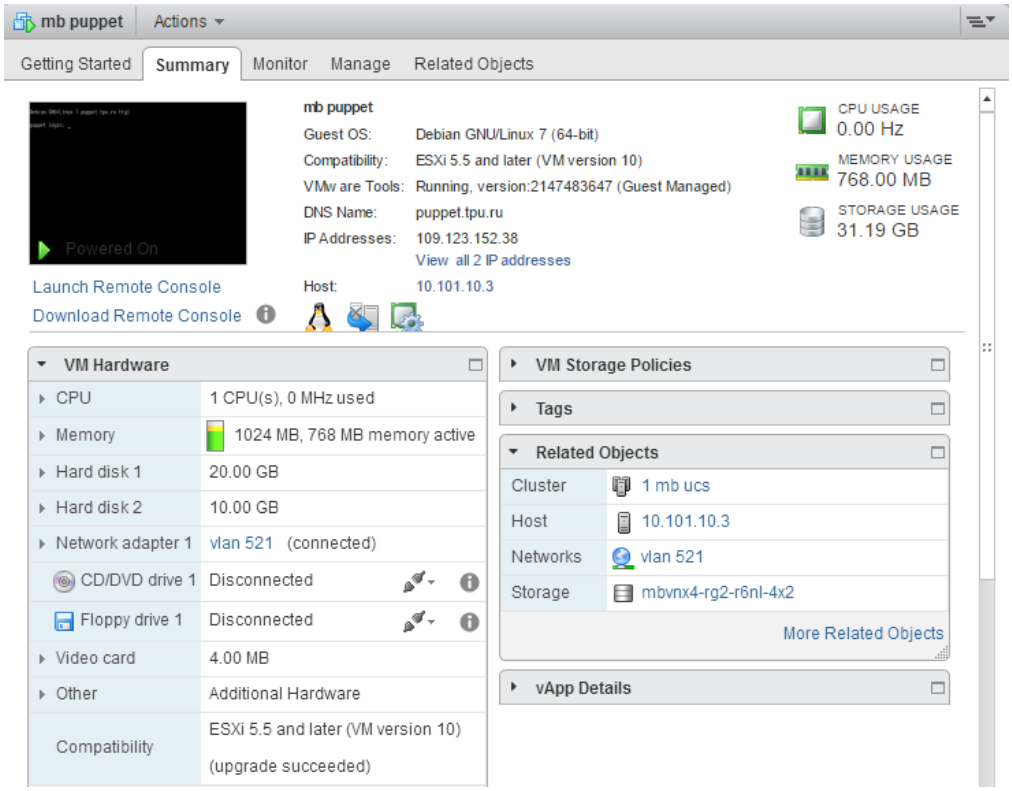

Рисунок 3 — Суммарная информация о виртуальной машине УУ

Следует отметить, что на рисунке 3 значение параметра «Guest OS»

неверно отображает версию гостевой операционной системы. Данный параметр выставляется вручную из списка поддерживаемых операционных систем конкретной версией гипервизора VMware ESXi. Debian GNU/Linux 7 является максимальной версией Debian, которую поддерживает гипервизор VMware ESXi 5.5. Несмотря на то, что гипервизор 10.101.10.3, на котором обеспечивается работа данной виртуальной машины имеет версию VMware ESXi 6.0U2, которая поддерживает Debian GNU/Linux 8, версия виртуальной машины оставлена совместимой с VMware ESXi 5.5, так как инфраструктура ЦОД еще содержит гипервизоры данной версии. Сделано это с целью обеспечения возможности отката версий гипервизоров в случае серьезных проблем в работе VMware ESXi 6.0 и выше.

После создания виртуальной машины туда была установлена операционная система Debian GNU/Linux 8 «Jessie». Были произведены базовые настройки сети в конфигурационных файлах /etc/network/interfaces и /etc/resolv.conf. Первый содержит такие параметры IPv4, как IP-адрес, маска подсети и шлюз по умолчанию. Во втором файле перечисляются IP-адреса серверов доменных имен и параметры поиска имени в доменах. Несмотря на то что в подсети, где был расположен УУ, имеется DHCP-сервер, производящий автоматическое конфигурирование сетевых параметров узлов, было принято решение установить статические параметры IPv4 на УУ. Для того чтобы работоспособность УУ не зависела от работоспособности сервера DHCP.

Была настроена почтовая служба Exim для пересылки локальной почты через почтовый релей ТПУ — relay.tpu.ru. В конфигурационном файле /etc/aliases учетной записи администратора root был присвоен псевдоним в виде адреса электронной почты группы администраторов GNU/Linux структурного подразделения ГИУ. Это позволило группе администраторов получать электронную почту о важных событиях в операционной системе, так как обновление программного обеспечения или реакция программного продукта Fail2ban на атаку подбора учетных записей.

Был установлен пакет программного обеспечения open-vm-tools. Этот

программный содержит открытую версию программного продукта VMware Tools. Назначение данного программного продукта в обеспечении лучшей совместимости гостевой операционной системы с гипервизором VMware ESXi. Достигается это благодаря интерфейсу между гостевой операционной системой и гипервизором, который предоставляет служба Open VMware Tools. К примеру, это позволяет отдавать виртуальным машинам больше оперативной памяти, чем есть у гипервизора. Open VMware Tools увеличивают стандартное значение таймаута блочных устройств с 30 до 180 секунд для обеспечения корректной обработки аварийной ситуации в дисковой подсистеме гипервизора, когда активный путь к системе хранения данных выходит из строя и гипервизор осуществляет переключение на резервный.

Был настроен межсетевой экран Netfilter. Для этого был создан файл /etc/iptables.rules, в котором были описаны правила экранирования в формате пригодном для загрузки утилитой командной строки iptables-restore. Правила Netfilter были настроены в стиле «запрещено всё что не разрешено». Правила, на которые стоит обратить внимание приведены ниже:

```
1. -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
2. -A INPUT -s 10.11.12.13/32 -p tcp -m conntrack --ctstate NEW \
-m tcp --dport 22 -j ACCEPT
3. -A INPUT -s 109.123.155.0/24 -p tcp -m conntrack --ctstate NEW \
-m tcp --dport 8140 -j ACCEPT
```

Правило под номером один разрешает прохождение всех пакетов по входящим соединениям, находящимся в состояниях установлено (ESTABLISHED) и порождено (RELATED). Соединение считается установленным, если оно прошло все три стадии тройного рукопожатия протокола TCP, подробнее в пункте 3.3.1 настоящей работы. Соединение считается порожденным, если оно связано с другим соединением, имеющим статус установлено.

Правило под номером два разрешает прохождение входящего TCP пакета с адреса 10.11.12.13/32 на порт 22 для соединений в состоянии новое (NEW). Состояние новое означает, что для данного правила подходят только пакеты с

поднятым флагом SYN и опущенным флагом ACK, т.е. пакеты инициирующие тройное рукопожатие протокола TCP.

Правило под номером три работает точно так же, как и предыдущее, но только для TCP порта 8140 и отправителей из диапазона адресов 109.123.155.0/24 — в котором расположены управляемые узлы. Именно на этот порт обращаются агенты puppet расположенные на них.

Для того чтобы правила межсетевого экранирования сохранялись после перезагрузки операционной системы был создан исполняемый файл /etc/network/if-pre-up.d/iptables, следующего содержания:

```
#!/bin/bash
/sbin/iptables-restore < /etc/iptables.rules
```

Первая строка файла /etc/network/if-pre-up.d/iptables говорит о том, что этот файл должен быть интерпретирован интерпретатором /bin/bash. Вторая строка содержит путь до утилиты iptables-restore, которой на вход передается содержимое файла /etc/iptables.rules. Исполняемые файлы, содержащиеся в каталоге /etc/network/if-pre-up.d/, исполняются каждый раз, до того как какой-либо сетевой интерфейс операционной системы переходит из выключенного состояния во включенное. Утилита iptables-restore считывает содержимое файла /etc/iptables.rules и загружает находящиеся в нем правила в ядро операционной системы.

Были установлены пакеты puppet, puppetmaster, puppetdb, puppetdb-terminus. Первый отвечает за установку агента puppet при помощи которого осуществляется настройка локального узла. Второй пакет обозначает серверную часть puppet — именно к ней будут подключаться агенты с удаленных узлов. Третий содержит схему базы данных, в которую будет складироваться вся необходимая серверу информация. Четвертый пакет отвечает за подключение сервера к базе данных, с этой целью были отредактированы конфигурационные файлы /etc/puppet/puppetdb.conf и /etc/puppet/puppet.conf. В первом были установлены параметры подключения к базе данных, а во втором указано серверу использовать базу данных для сохранения информации о управляемых узлах и

тип базы данных.

В файле /etc/puppet/manifests/site.pp был определен один управляемый узел — это непосредственно сам сервер управления. Его определение выглядит следующим образом:

```
node 'puppet.tpu.ru' {
    service { puppet: ensure => running, enable => true, }
    service { puppetdb: ensure => running, enable => true, }
    service { puppetmaster: ensure => running, enable => true, }
}
```

Здесь мы указали puppet агенту, что сервисы puppet, puppetdb и puppetmaster должны быть запущены и включены в автозагрузку. Если настоящее состояние сервисов отлично от указанного, то агент должен выполнить все необходимые действия, для приведения сервисов к требуемому состоянию. Примеры определений управляемых узлов в файле /etc/puppet/manifests/site.pp приведены в приложении А.

4.1.1 Модуль tpu_sshd

Для управления настройками удаленного доступа был разработан модуль к puppet — tpu_sshd, в котором описан класс tpu_sshd и два его наследника tpu_sshd::sec_lvl1 и tpu_sshd::sec_lvl2. Исходный код модуля tpu_sshd приведен в приложении Б. Класс tpu_sshd описывает требуемое состояние службы OpenSSH на управляемых узлах:

- `package { 'openssh-server': ensure => latest }` — пакет с OpenSSH должен быть установлен и иметь последнюю доступную версию;
- `service { 'ssh': ensure => 'running' }` — сервис должен быть запущен;
- `service { 'ssh': enable => 'true' }` — сервис должен быть включен в автозагрузку;

- `service { 'ssh': subscribe => File[sshdconfig] }` — сервис подписан на файл sshdconfig, если в дальнейшей конфигурации будет определен файл sshdconfig и его конфигурация будет отличаться от текущей, то сервис ssh, будет считаться измененным и агент перезапустит его.

В классе tpu_sshd, помимо состояния OpenSSH, описано требуемое состояние программного пакета Fail2ban. Требуемое состояние Fail2ban почти во всем повторяет состояние OpenSSH. Разница заключается в том, что для Fail2ban прописаны конфигурационные файлы /etc/fail2ban/jail.local и /etc/fail2ban/jail.conf.

```
file { '/etc/fail2ban/jail.local' :
    ensure => present,
    owner => root,
    group => root,
    mode => '0600',
    source =>
"puppet:///modules/tpu_sshd/jail.local.${::osfamily}.${::os_maj_version}"
,
    require => Package['fail2ban'],
    notify => Service['fail2ban'],
}
```

Получая такую конфигурацию агент puppet должен убедиться в наличии конфигурационного файла /etc/fail2ban/jail.local и привести его к виду доступному по адресу puppet:///modules/tpu_sshd/jail.local.${::osfamily}.${::os_maj_version}, где .${::osfamily} и ${::os_maj_version} раскрываются в имя дистрибутива и номер версии дистрибутива соответсвенно. К примеру, для Debian 7 полный путь к файлу получится puppet:///modules/tpu_sshd/jail.local.Debian.7. Адрес этого файла на узле управления /etc/puppet/modules/tpu_sshd/files/jail.local.Debian.7. Параметры owner, group и mode определяют хозяина, группу и режим доступа к конфигурационному файлу соответственно. Конфигурационный файл требует описания пакета fail2ban и оповещает сервис fail2ban в случае изменений конфигурации.

К сожалению разные дистрибутивы GNU/Linux используют разные версии Fail2ban с сильно разнящейся конфигурацией. По этой причине для каждого дистрибутива и каждой версии дистрибутива были написаны свои конфигурационные файлы.

Два класса tpu_sshd::sec_lvl1 и tpu_sshd::sec_lvl2 являются наследниками класса tpu_sshd и расширяют его функционал в части описания конфигурационного файла sshdconfig.

```
class tpu_sshd::sec_lvl1 inherits tpu_sshd {
    case $::osfamily {
        'Gentoo': {
            file { 'sshdconfig':
                name => '/etc/ssh/sshd_config',
                source => 'puppet:///modules/tpu_sshd/sshd_config_lvl1',
                require => Package['ssh'],
            }
        }
        'Debian', 'RedHat': {
            file { 'sshdconfig':
                name => '/etc/ssh/sshd_config',
                source => 'puppet:///modules/tpu_sshd/sshd_config_lvl1',
                require => Package['openssh-server'],
            }
        }
    }
}
```

В классах tpu_sshd::sec_lvl1 и tpu_sshd::sec_lvl2 был использован оператор case при помощи которого осуществляется персонализация конфигурации для разных дистрибутивов.

На сервере puppet было определено два конфигурационных файла OpenSSH "/etc/puppet/modules/tpu_sshd/files/sshd_config_lvl1" и "/etc/puppet/modules/tpu_sshd/files/sshd_config_lvl2". Оба файла запрещают аутентификацию под учетной записью root — опция PermitRootLogin в значении no. Отличительной особенностью первого является запрет аутентификации по паролю — опция PasswordAuthentication в значении no и опция KbdInteractiveAuthentication в значении no.

4.1.2 Модуль tpu_ssh_acc

Для распространения учетных записей администраторов между управляемыми узлами был написан модуль tpu_ssh_acc. Экземпляры данного модуля подробно приведены в приложении В. В модуле определен родительский класс tpu_ssh_acc, который накладывает требование require ::tpu_sshd — это означает, что этот класс и все дочерние классы будут применены только при условии, что к узлу применен класс tpu_sshd.

В модуле определены дочерние к tpu_ssh_acc классы: mind, ora, net, ns, backup. Эти классы содержат администраторов соответствующих направлений в структуре организации управления по информатизации ТПУ.

- mind (Main Informational Department) — ГИУ;
- ora (Oracle) — группа администраторов продуктов Oracle;
- net (Network) — группа администраторов сети;
- ns (Name Server) — группа администраторов серверов доменных имен;
- backup — группа администраторов резервного копирования.

Администраторы определяются конструкцией

```
user { 'user1':
    ensure     => present,
    forcelocal => true,
    managehome => true
}
```

Которая определяет имя администратора, в данном случае user1, состояние — он должен существовать на управляемом узле. Независимо от того, какие механизмы аутентификации настроены — администратор должен присутствовать на узле локально. Для администратора должен быть создан домашний каталог.

Помимо базовых параметров, администратору может быть добавлен открытый ключ, для аутентификации на узлах и параметры на каких условиях он

может повышать свои привилегия до уровня root.

Открытый ключ описывается в блоке ssh_authorized_key. Связь блока описывающего администратора и блока с открытым ключом осуществляется через строку require => User[' user1'].

4.1.3 Модуль tpu_autoupdate

Для управление процедурой автоматического обновления управляемых узлов был написан модуль tpu_autoupdate. Исходный текст данного модуля приведен в приложении Г. Процедура автоматического обновления для дистрибутивов операционной системы GNU/Linux зависит от пакетного менеджера этого дистрибутива.

Узлы под управлением дистрибутива Debian используют в качестве пакетного менеджера программный продукт APT. Автоматической обновление дистрибутива Debian было реализовано при помощи пакета unattended-upgrades. Было определено два конфигурационных файла /etc/apt/apt.conf.d/50unattended-upgrades и /etc/apt/apt.conf.d/20auto-upgrades.

В файле /etc/apt/apt.conf.d/50unattended-upgrades определили, что следует устанавливать только обновления безопасности, отчеты о изменениях направлять локальному пользователю root, не допускать автоматической перезагрузки узла.

В файле /etc/apt/apt.conf.d/20auto-upgrades производится включение или отключение автоматического обновления изменением значения параметра APT::Periodic::Unattended-Upgrade. Значение 1 — автоматическое обновление включено, 0 — выключено.

Узлы под управлением дистрибутивов, основанных на пакетной базе Red Hat Enterprise Linux, используют в качестве пакетного менеджера программный продукт YUM. Автоматическое обновление подобных дистрибутивов было реализовано при помощи программного пакета yum-cron. Синтаксис и расположение конфигурационного файла программного пакета yum-cron

отличается в разных версиях дистрибутивов. Для решения это проблемы наряду с макросом ::osfamily был задействован макрос ::os_maj_version, чтобы персонифицировать конфигурацию к конкретной версии конкретного дистрибутива.

4.1.4 Модуль tpu_mirror_repository

В процессе опытной эксплуатации модуля tpu_autoupdate было принято решение разработать модуль tpu_mirror_repository. Назначение этого модуля определять на управляемых узлах в качестве источников пакетов для пакетных менеджеров APT и YUM сервер, расположенный в пределах адресного пространства ЦОД. Во-первых, этот шаг уменьшил нагрузку на дорогостоящие внешние каналы передачи данных. Во-вторых, упростил настройку узлов, которым запрещен доступ во внешнюю сеть. Исходный текст разработанного модуля приведен в приложении Д.

4.1.5 Описание конфигурации атомарного управляемого узла

После того как были разработаны вышеперечисленные модули была проведена из опытная эксплуатация на ряде узлов ЦОД. Для этого в файле /etc/puppet/manifests/site.pp были созданы записи этих узлов следующего вида

```
node 'filecloud.tpu.ru' {
    include tpu_sshd::sec_lvl1
    include tpu_ssh_acc::mind
    include tpu_mirror_repository
    include tpu_autoupdate
}
```

Вышеприведенная конструкция описывает конфигурацию узла filecloud.tpu.ru. Предписывает использовать профиль удаленного доступа первого уровня, т.е. разрешена аутентификация только по открытым ключам.

Предоставляет доступ администраторам из группы mind, т.е. сотрудникам ГИУ. Настраивает в качестве источников пакетов программного обеспечения ресурс в пределах адресного пространства ЦОД mirror.tpu.ru. Определяет параметры автоматического обновления.

4.2 Узел журналирования

Для размещения УЖ была создана виртуальная машина суммарная информация о которой размещена на рисунке 4.

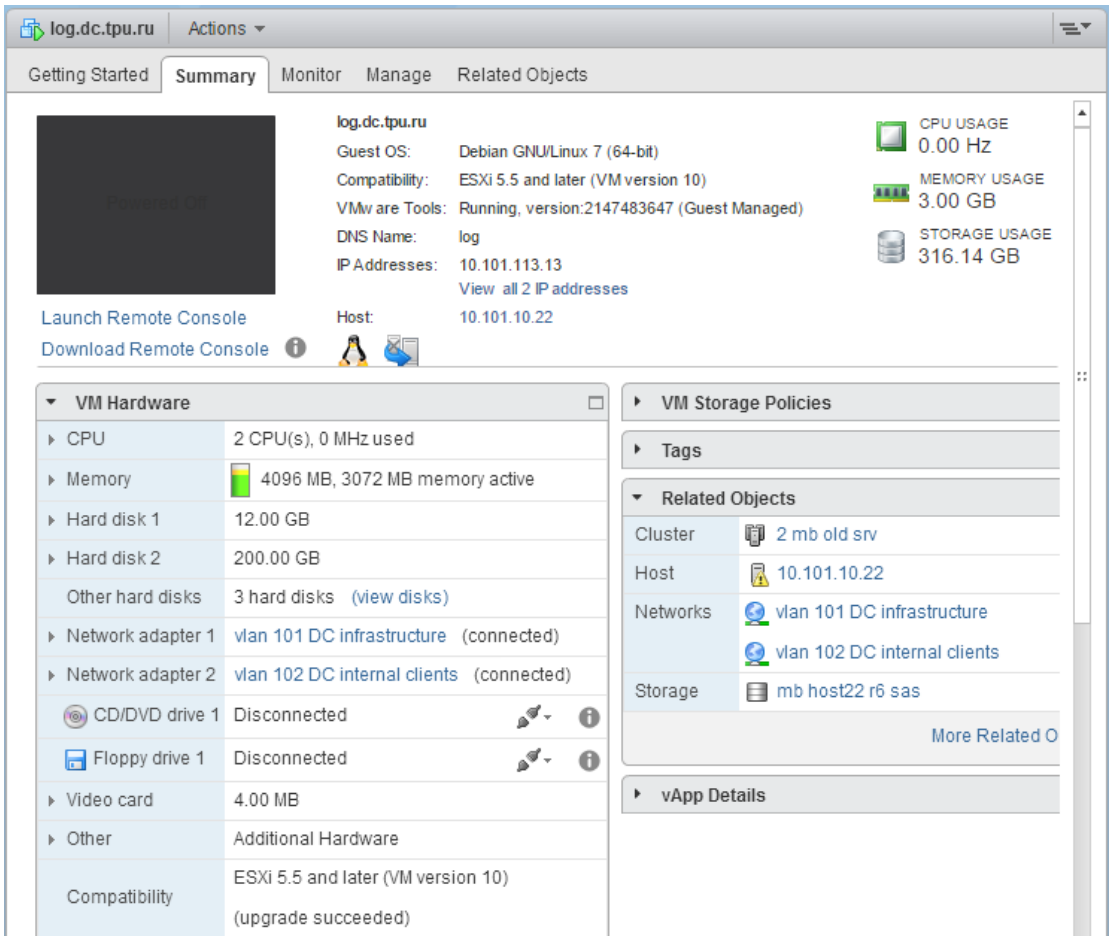

Рисунок 4 — Суммарная информация о виртуальной машине УЖ

Были произведены базовые настройки операционной системы. Правила межсетевого экрана Netfilter для данного узла отличаются от УУ в части перечисления разрешенных входящих портов. В данном случае для входящих

подключений был разрешен порт 514 по протоколам UDP и TCP. Порт 514 прослушивается службой rsyslogd, которая отвечает за прием, фильтрацию и передачу журналируемых сообщений на хранение.

Параметры Rsyslog были определены в конфигурационном файле /etc/rsyslog.d/00-remote.conf. Для того чтобы Rsyslog прослушивал порт 514 по протоколам UDP и TCP были задействованы модули imudp и imtcp.

В качестве хранилища журналируемой информации была использована СУБД MySQL. Средствами пакетного менеджера APT к Rsyslog был установлен модуль rsyslog-mysql, который произвел инициализацию схемы базы данных Syslog.

В конфигурационном файле /etc/rsyslog.d/00-remote.conf был определен фильтр, передающий все сообщения, источником которых не является сам узел журналирования, в базу данных Syslog. Следом за фильтром идет конструкция “& ~”, которая говорит службе Rsyslog прекратить обработку сообщения, если они были обработаны предыдущим правилом.

```
1. if $fromhost-ip != '127.0.0.1' then :ommysql:localhost,Syslog,rsyslog,pass
2. & ~
```

Для оперативного доступа к журналируемой информации и возможности производить поиск и фильтрацию п ней было принято решение установить на УЖ программный продукт LogAnalayzer 3.6.6, который на май 2016 всё еще остается последним стабильны выпуском [57]. Дистрибутив данного программного продукта был получен с официального сайта проекта и развернут в каталог /var/www/ узла журналирования, а в /etc/apache2/sites-available/ добавлен конфигурационный файл с описанием параметров доступа к данному веб-интерфейсу. Параметры подключения веб-интерфейса к базе данных Syslog сохранены в файле /var/www/loganalyzer/config.php.

Для предотвращения чрезмерного увеличения объема базы данных Syslog было принято решение регулярно архивировать часть информации и удалять из базы данных строки, переданные в архив. С этой целью был написан небольшой

сценарий.

```
1. /usr/bin/mysqldump Syslog | gzip -c > /root/backup/ \
rsyslog-`/bin/date +%F`.sql.gz 2>>/var/log/rsync-backup.log
2. cd /var/www/loganalyzer/cron
3. /usr/bin/php ./maintenance.php cleandata 1 olderthan 1296000 \
&>>/var/log/rsync-backup.log
```

В первой строке используется утилита mysqldump, которая преобразует содержимое базы данных Syslog в текстовый описанный на языке SQL. Результат работы утилиты mysqldump передается на вход утилите gzip, которая сжимает получаемую информацию без потерь алгоритмом LZ77. Выход утилиты gzip перенаправляется в файл вида /root/backup/rsyslog-2016-05-15.sql.gz. После завершения архивирования базы данных системном интерпретатору php передается для исполнения файл maintenance.php из поставки программного продукта LogAnalayzer с параметрами указывающими удалить строки с датой регистрации старше 1296000 секунд, что равняется 15 дням.

В системный планировщик узла журналирования была добавлена строка запуска сценария архивирования и удаления строк таблицы Syslog.

```
1 0 1,15 * * root /usr/local/bin/rsyslog_backup.sh
```

В соответствии с этой настройкой каждую первую секунду нулевого часа каждого 1 и 15 числа каждого месяца от пользователя root будет запущен сценарий, который расположен по адресу /usr/local/bin/rsyslog_backup.sh.

Глава 5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

В данной работе была поставлена цель создать систему управления инфраструктурой центра обработки данных при помощи существующих свободных программных решений.

Система должна была интегрироваться в уже существующую инфраструктуру без покупки дополнительных программных или аппаратных средств. Следовательно, необходимо было провести исследование существующей инфраструктуры, ознакомиться с перечнем доступных свободных решений, выбрать наиболее подходящие, реализовать систему на основе выбранного решения и внедрить его для последующей эксплуатации.

5.1 Организация и планирование работ

Для реализации целей ВКР необходимо было выполнить задачи, показанные в таблице 5.

При организации процесса реализации данного проекта была приблизительно распланирована занятость каждого из его участников.

Далее приведен перечень работ и продолжительность их выполнения каждым из участников (научным руководителем и исполнителем).

Таблица 5 — Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей разработки и предложение возможных способов их решения	НР, И	НР — 50% И — 50%

Этапы работы	Исполнители	Загрузка исполнителей
Исследование существующей инфраструктуры	НР, И	НР — 10% И — 90%
Поиск подходящих решений	НР, И	НР — 10% И — 90%
Разработка системы на основе выбранного решения	И	И — 100%
Оформление пояснительной записки	И	И — 100%
Подведение итогов	НР, И	НР — 50% И — 50%

5.2 Продолжительность этапов работ

В нашем случае оценить продолжительность этапов работа можно при помощи опытно-статистического метода экспертным способом. Ожидаемое значение продолжительности работ $t_{ож}$ находится по формуле

$$t_{ож} = \frac{3*t_{min}+2*t_{max}}{5}, \quad (1)$$

где t_{min} — минимальная продолжительность работы, дн.;

t_{max} — максимальная продолжительность работы, дн.

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести ее в календарные дни. Расчет продолжительности выполнения каждого этапа в рабочих днях ($Т_{РД}$) ведется по формуле:

$$Т_{РД} = \frac{t_{ож}}{К_{ВН}} * К_{Д}, \quad (2)$$

где $К_{вн}$ — коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, в частности, возможно $К_{вн} = 1$;

$К_Д$ — коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ. Примем $К_Д = 1,1$.

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$Т_{КД} = Т_{РД} * Т_К, \quad (3)$$

где $Т_{КД}$ — продолжительность выполнения этапа в календарных днях;

$Т_К$ — коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях. При пятидневной рабочей неделе $Т_К = 1,4$. Результаты расчетов приведены в таблице Таблица *6*.

Таблица 6 — Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.- дн.			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
1	2	3	4	5	6	7	8	9
Постановка целей разработки и предложение возможных способов их решения	НР, И	3	5	3,8	2,1	2,1	2,9	2,9
Исследование существующей инфраструктуры	НР, И	5	10	7	0,8	6,9	1,1	9,7
Поиск подходящих решений	НР, И	10	20	14	1,5	13,9	2,2	19,4
Разработка системы на основе выбранного решения	И	20	30	24	0	26,4	0	37
Оформление пояснительной записки	И	20	30	24	0	26,4	0	37
Подведение итогов	НР, И	1	5	2,6	1,4	1,4	2	2
Итого:		59	100	75,4	5,8	77,1	8,2	108

Для наглядного отображения трудозатрат участниками проекта составим был составлен линейный график работ, приведенный в таблице 7. В линейном графике закрашенной линией отмечены трудозатраты исполнителя, а прозрачной трудозатраты научного руководителя. В столбце 1 указаны номера по порядку соответствующих этапов трудовой деятельности из таблицы 6, а в столбцах 2 и 3 числовые показатели трудозатрат научным руководителем и исполнителем соответственно.

Таблица 7 — Линейный график работ

Этап	НР	И	Февраль			Март			Апрель			Май	
			10	20	30	40	50	60	70	80	90	100	110
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	2,9	2,9											
2	1,1	9,7											
3	2,2	19,4											
4	0,0	37,0											
5	0,0	37,0											
6	2,0	2,0											

5.3 Расчет накопления готовности проекта

Цель данного пункта — оценка текущих состояний (результатов) работы над проектом. Величина накопления готовности работы показывает, на сколько процентов по окончании текущего (i-го) этапа выполнен общий объем работ по проекту в целом.

Введем обозначения:

$ТР_{общ.}$ — общая трудоемкость проекта;

$ТР_i$ ($ТР_k$) − трудоемкость i-го (k-го) этапа проекта, $i = \overline{1,I}$;

$ТР_{iН}$ − накопленная трудоемкость i-го этапа проекта по его завершении;

$ТР_{ij}$ ($ТР_{kj}$) − трудоемкость работ, выполняемых j-м участником на i-м этапе, здесь $j = \overline{1,m}$ − индекс исполнителя, в нашем примере m = 2.

Степень готовности определяется формулой

$$СГ_i = \frac{ТР_i^Н}{ТР_{общ.}} = \frac{\sum_{k=1}^{i} ТР_k}{ТР_{общ.}} = \frac{\sum_{k=1}^{i}\sum_{j=1}^{m} ТР_{km}}{\sum_{k=1}^{I}\sum_{j=1}^{m} ТР_{km}}. \quad (4)$$

Результаты расчетов приведены в таблице 8.

Таблица 8 — Нарастание технической готовности работы

Этап	$ТР_i$, %	$СГ_i$, %
Постановка целей разработки и предложение возможных способов их решения	5,0	5,0
Исследование существующей инфраструктуры	9,3	14,3
Поиск подходящих решений	18,6	32,9
Разработка системы на основе выбранного решения	31,8	64,7
Оформление пояснительной записки	31,8	96,6
Подведение итогов	3,4	100,0

5.4 Расчет сметы затрат на выполнение проекта

В состав затрат на создание проекта включается величина всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной работы. Расчет сметной стоимости ее выполнения производится по следующим статьям затрат:

- материалы и покупные изделия;
- заработная плата;

- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- прочие (накладные расходы) расходы.

5.4.1 Расчет затрат на материалы

К данной статье расходов относится стоимость материалов, покупных изделий и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ над объектом проектирования. Затраты на материалы и покупные изделия сведены в таблицу 9.

Таблица 9 — Затраты на материалы и покупные изделия

Наименование материалов и покупных изделий	Единица измерения	Количество	Цена за ед., руб.	Стоимость, руб.
Бумага для печати	уп.	2	200	400
Ручка шариковая	шт.	2	12	24
Файл-вкладыш	шт.	50	2	100
Скоросшиватель	шт.	3	11	33
Картридж для принтера	шт.	1	2800	2800
Транспортно-заготовительные расходы				335,7
Итого				3692,7

5.4.2 Расчет заработной платы

Данная статья расходов включает заработную плату научного руководителя и исполнитель проекта, а также премии, входящие в фонд заработной платы. Затраты на заработную плату сведены в таблицу 10. В расчете учитывается, что в 2016 году, при пятидневной рабочей неделе, насчитывается 247 рабочих дней, а, следовательно, в месяце в среднем 20,58 рабочих дня.

Таблица 10 — Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднеднев-ная ставка, руб./раб.день	Затраты времени, раб.дни	Коэффициент	Фонд з/платы, руб.
НР	20 776,45	1 009,55	5,80	1,62	9 485,69
И	7 483,58	363,63	77,10	1,62	45 418,57
Итого:					54 904,26

5.4.3 Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН), включающий в себя отчисления в пенсионный фонд, на социальное и медицинское страхование, составляют 30 % от полной заработной платы по проекту, т.е. $С_{соц} = С_{зп}*0,3$. Итак, в нашем случае $С_{соц} = 54\ 904,26 * 0,3 = 16\ 471,28$ руб.

5.4.4 Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, потраченную в ходе выполнения проекта на работу используемого оборудования, рассчитываемые по формуле:

$$С_{эл.об} = Р_{об} * t_{об} * Ц_э, \quad (5)$$

где $Р_{об}$ — мощность, потребляемая оборудованием, кВт;

$Ц_э$ — тариф 1 кВт*ч;

$t_{об}$ — время работы оборудования, час.

Для ТПУ $Ц_Э$ = 5,257 руб./квт·час (с НДС).

Время работы оборудования вычисляется на основе итоговых данных таблицы Таблица *6* для исполнителя ($Т_{РД}$) из расчета, что продолжительность рабочего дня равна 8 часов. Коэффициент $К_t$ в нашем случае равен 0,3.

$$t_{об} = T_{РД} * 24 * К_t \quad (6)$$

Мощность, потребляемая оборудованием, определяется по формуле:

$$Р_{ОБ} = Р_{ном} * К_С, \quad (7)$$

где $Р_{ном}$ — номинальная мощность оборудования, кВт;

$К_С$ — коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности $К_С = 1$.

Таблица 11 — Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{ОБ}$, час	Потребляемая мощность $Р_{ОБ}$, кВт	Затраты $Э_{ОБ}$, руб.
Персональный компьютер	555,192	0,6	1 751,19
Лазерный принтер	5	0,5	13,14
Итого:			1 764,33

5.4.5 Расчет амортизационных расходов

В статье «Амортизационные отчисления» рассчитывается амортизация используемого оборудования за время выполнения проекта. Результаты расчетов приведены в таблице 12.

Используется формула

$$С_{АМ} = \frac{Н_А * Ц_{ОБ} * t_{рф} * n}{F_Д}, \quad (8)$$

где $Н_А$ — годовая норма амортизации единицы оборудования, в нашем случае берется из диапазона 2 — 3 года на основании постановления правительства РФ «О классификации основных средств, включенных в амортизационные группы», код ОКОФ 14 2894000. Примем $Н_А$=1/2,5=0,4;

$Ц_{ОБ}$ — балансовая стоимость единицы оборудования с учетом ТЗР.

$F_Д$ — действительный годовой фонд времени работы соответствующего

оборудования. В нашем случае, при пятидневной рабочей неделе, в 2016 году насчитывается 247 рабочих дня, а значит $F_Д$=247*8=1976 часов.

$t_{рф}$ — фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта;

n — число задействованных однотипных единиц оборудования.

Таблица 12 — Затраты на амортизационные отчисления

Наименование оборудования	Стоимость, руб	$Н_А$	Время работы оборудования $t_{ОБ}$, час	$F_Д$, час	Стоимость аморт. отч., руб
Персональный компьютер	50000	0,4	555,192	1976	5 619,35
Лазерный принтер	20000	0,4	5	1976	20,24
Итого:					5 639,60

5.4.6 Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на выполнение проекта, которые не учтены в предыдущих статьях, их размер принят равными 10% от суммы всех предыдущих расходов, т.е.

$$С_{проч.} = (С_{мат} + С_{зп} + С_{соц} + С_{эл} + С_{ам}) * 0,1 =$$
$$= (3692,7 + 54904,26 + 16471,28 + 1764,33 + 5639,6) * 0,1 =$$
$$= 8247,22 \text{ руб.}$$

5.4.7 Расчет общей себестоимости работы

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость работы «Обеспечение безопасности системы управления инфраструктурой центра обработки данных ТПУ». Результаты расчета приведены в таблице 13.

Таблица 13 — Смета затрат на данную работу

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$С_{мат}$	3 692,70
Основная заработная плата	$С_{зп}$	54 904,26
Отчисления в социальные фонды	$С_{соц}$	16 471,28
Расходы на электроэнергию	$С_{эл.}$	1 764,33
Амортизационные отчисления	$С_{ам}$	5 639,60
Прочие расходы	$С_{проч}$	8 247,22
Итого:		90 719,38

Таким образом, затраты на разработку составили С = 90 719,38 руб.

5.4.8 Расчет прибыли

Размер прибыли примем равным 20% от себестоимости проекта, таким образом она составит 18 143,88 руб.

5.4.9 Расчет НДС

НДС составляет 18% от суммы затрат на разработку и прибыли. В нашем случае это (90 719,38 + 18 143,88) * 0,18 = 19 595,39 руб.

5.4.10 Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС, в нашем случае

$$Ц_{НИР} = 90\,719{,}38 + 18\,143{,}88 + 19\,595{,}39 = 128\,458{,}64 \text{ руб.}$$

5.5 Оценка экономической эффективности проекта

Основными факторами ожидаемого экономического эффекта данного проекта являются: снижение рисков и повышение эффективности при администрировании существующей инфраструктуры организации. Количественная оценка эффективности в рамках данной работы невозможна, в виду методологической сложности и очень высокой стоимости подобного исследования. Причины обуславливающие данные эффекты изложены ниже.

Снижение рисков в контексте информационной безопасности трудно выразить количественно. Нам заранее неизвестно какого рода атаки будут выполнены на инфраструктуру и какие действия предпримет потенциальный взломщик в случае её успеха. Можно лишь предположить, что произойдет в худшем случае развития событий.

Во-первых, в случае кражи базы данных организации будет нарушена конфиденциальность персональных данных хранящихся в ней. Требование обеспечивать конфиденциальность данных накладывает федеральный закон Российской Федерации №152-ФЗ от 27.07.2006 «О персональных данных». Лица, виновные в нарушении требований настоящего Федерального закона, несут предусмотренную законодательством Российской Федерации ответственность. Которая может носить гражданский, уголовный, административный или дисциплинарный характер. Так статья 19.5 Кодекса об административных правонарушениях (КоАП) РФ «Невыполнение в срок законного предписания (постановления, представления, решения) органа (должностного лица), осуществляющего государственный надзор (контроль)» предусматривает наложение штрафа в размере до 500 000 руб. и дисквалификацию должностного лица сроком до 3 лет. Статья 137 Уголовного кодекса (УК) РФ «Нарушение неприкосновенности частной жизни» предусматривает наложение штрафа в размере до 300 000 руб., исправительные работы сроком до 240 часов, арест

сроком до 6 месяцев. Пожалуй, самый непредсказуемый и потенциально наиболее серьезный ущерб организации может нанести ответственность по статье 24 152-ФЗ от 27.07.2006 «О персональных данных» и статье 237 Трудового кодекса РФ «Моральный вред, причинённый работнику неправомерными действиями или бездействием работодателя», которые дают физическим лицам право на возмещение морального и имущественного вреда, а также понесённых убытков. Размер возмещения будет определяться судом, и заранее он ничем не ограничен.

Во-вторых, нарушение работоспособности ключевых информационных систем организации может привести к невозможности осуществлять свою деятельность, что ставит под угрозу само её существование

В-третьих, даже в случае успешного и своевременного устранения последствий взлома информационных ресурсов, организация неизбежно теряет свою репутацию, что оказывает прямой отрицательный эффект на её конкурентоспособность, а, следовательно, и эффективность.

Результаты деятельности не планируется превращать в коммерческий продукт и использовать за пределами организации.

При этом, стек используемых технологий, и методика решения поставленной задачи могут использоваться другими организациями в качестве положительного опыта. Учитывая тот факт, что реализация подобной системы не требует покупки дополнительных аппаратных или программных средств, а влечет только трудозатраты по внедрению, можно сделать вывод о благотворном влиянии применения результатов данной работы.

5.6 Оценка научно-технического уровня НИР

Научно-технический уровень характеризует влияние проекта на уровень и динамику обеспечения научно-технического прогресса в данной области. Для оценки научной ценности, технической значимости и эффективности, планируемых и выполняемых НИР, используется метод балльных оценок.

Балльная оценка заключается в том, что каждому фактору по принятой шкале присваивается определенное количество баллов — таблица *Таблица 14*. Обобщенную оценку проводят по сумме баллов по всем показателям. На ее основе делается вывод о целесообразности НИР.

Сущность метода заключается в том, что на основе оценок признаков работы определяется интегральный показатель (индекс) ее научно-технического уровня по формуле:

$$I_{\text{НТУ}} = \sum_{i=1}^{3} R_i * n_i, \qquad (9)$$

где $I_{НТУ}$ — интегральный индекс научно-технического уровня;

R_i — весовой коэффициент i-го признака научно-технического эффекта;

n_i — количественная оценка i-го признака научно-технического эффекта, в баллах.

Таблица 14 — Оценка научно-технического уровня НИР

Значимость	Фактор НТУ	Уровень фактора	Выбранный балл	Обоснование выбранного балла
0.4	Уровень новизны	Относительно новая	4	Используются уже известные сведения для увеличения эффективности
0.1	Теоретический уровень	Разработка способа	6	Система разработана и реализована
0.5	Возможность реализации	В течение первых лет	10	Для применения на практике необходимо меньше 1 года

Отсюда интегральный показатель научно-технического уровня для нашего проекта составляет $I_{нту} = 0,4*4 + 0,1*6 + 0,5*10 = 1,6 + 0,6 + 5 = 7,2$.

Интегральный показатель равен 7.2. Таким образом, данный проект имеет средний уровень научно-технического эффекта.

Глава 6 Социальная ответственность

В данном разделе выпускной квалификационной работы проведен анализ возможных сбоев разрабатываемой системы и последствия, возникающие в результате этих сбоев, а также выявлены причины возможных сбоев. На основании полученного анализа разработаны меры по предотвращению аварийных ситуаций и сбоев как для системы в целом, так и для сервера управления инфраструктурой, в частности.

6.1 Анализ возможных сбоев разрабатываемой системы

Сбои в работе системы управления инфраструктурой могут привести к уничтожению информационных систем, управлением которых занимается данная система, либо к невозможности выполнения задач, возложенных на данную систему.

В первом случае работа организации будет частично нарушена на время восстановления из резервных копий уничтоженных информационных систем.

Во втором случае административный персонал будет испытывать неудобства ввиду невозможности совершения поставленных задач в автоматизированном режиме.

Все сбои, происходящие в ходе работы системы, можно разделить на несколько категорий:

- аппаратные сбои;
- сбои, возникающие в результате неправильной эксплуатации системы;
- программные сбои.

Система управления имеет клиент-серверную архитектуру, где сервером является сервер управления инфраструктурой, а клиентами узлы, подвергающиеся управлению. Общение сервера с узлами осуществляется

посредством сетевого соединения.

Аппаратные сбои возможны в случае отказа компонент электронно-вычислительного комплекса, обеспечивающего работу операционной системы сервера управления инфраструктурой, либо в случае отказа компонент, обеспечивающих его сетевое взаимодействие с узлами.

В обоих случаях административный персонал будет испытывать неудобства ввиду невозможности совершения поставленных задач в автоматизированном режиме. При этом управляемые узлы продолжат функционировать в штатном режиме. Единственный эффект, который будет наблюдаться на узлах — отсутствие изменений состояния конфигурации.

Сбои, возникающие в результате неправильной эксплуатации системы, могут привести к наиболее серьезным последствиям. Теоретически возможна ситуация, приводящая к выходу из строя всех управляемых узлов в следствии неверно заданной администратором конфигурации.

К программным сбоям можно отнести ошибки в коде используемых в данной системе программ. Результат их влияние может привести к тем же последствиям, что описаны выше.

6.2 Анализ причин сбоев в работе системы

Отказ аппаратных средств, лежащих за пределами операционной системы сервер управления, может быть вызван несовершенством технологий, ошибками, допущенными при производстве, нарушениями условий эксплуатации аппаратных средств.

Сбои, возникающие в результате неправильной эксплуатации системы, могут возникнуть в силу низкой квалификации, либо невнимательности администратора.

Программные сбои возможны в виду того что программное обеспечение разрабатывается большим сообществом программистов. Все они имеют разную

квалификацию, по-разному относятся к своей работе и обладают различными знаниями о строении операционной системы и ее особенностях. К тому же среди пользователей сети Интернет существует большое количество злоумышленников, т.е. лиц которые по различным причинам задаются целью взломать какую-либо не принадлежащую им систему.

6.3 Меры по аппаратной защите системы

В целях защиты системы управления от аппаратных сбоев принято решение разместить её внутри виртуальной машины в центре обработки данных (ЦОД) ТПУ. Применяемые в ЦОД аппаратные средства имеют высокую надежность, призванную обеспечить высокую доступность размещенных сервисов.

Хранение данных осуществляется на системах хранения данных (СХД) корпоративного класса фирмы EMC. Все компоненты СХД дублированы. Диски сконфигурированы в RAID группы уровня 6, что позволяет сохранить данные доступными при одновременном выходе из строя двух дисков в одной группе. Операционная система СХД постоянно отслеживает состояние дисков и других своих аппаратных компонент.

СХД содержит постоянное запоминающее устройство (ПЗУ) для отслеживания состояния всех данных массива. Если произошло отключение питания после того как данные были записаны, но до того, как была обновлена контрольная сумма, это будет отмечено в ПЗУ. После восстановления питания СХД допишет необходимые данные основываясь на состояние ПЗУ. В худшем случае, при потере ПЗУ, в результате сбоя, будет запущен процесс проверки целостности всех данных массива.

Как упоминалось выше, операционная система СХД в постоянном режиме отслеживает состояние своих аппаратных компонент и имеет встроенные алгоритмы предсказания сбоев. К примеру, основанная на технологии S.M.A.R.T

(от англ. self-monitoring, analysis and reporting technology — технология самоконтроля, анализа и отчётности), техника предварительного копирования данных с подозрительных жестких дисков резервный позволяет сократить время восстановления массива до нуля. Следовательно, избежать нахождение массива в уязвимом деградированном состоянии.

Доступ к СХД осуществляется посредством двух независимых сетей Fibre Channel, называемых фабриками. Две полностью независимые сети призваны исключить появления единых точек отказа между хостами с гипервизорами и системами хранения данных.

В качестве хостов с гипервизорами используются унифицированные вычислительные серверы компании Cisco. Каждый сервер имеет две независимые питающие линии и оперативную память с контролем четности.

6.4 Организационные меры, обеспечивающие защиту системы

Организационные меры по защите системы включают:

- регулярный аудит безопасности сервера системы управления;
- строгий учет лиц, имеющих доступ к системе управления;
- выявление должного уровня квалификации администратора посредством стажировки в момент приема на работу;
- обучение администратора правилам работы с системой управления;
- регулярная принудительная смена ключей аутентификации администраторов;
- отслеживание и журналирование изменений, вносимых в конфигурацию системы.

6.5 Меры по программной защите системы

Целостность данных на уровне СХД ЦОД обеспечивается следующими техниками. Диски имеют нестандартный размер сектора, равный 520 байт, что на 8 байт больше общепринятого значения в 512 байт. Дополнительные 8 байт несут в себе информацию о целостности данных в секторе (контрольную сумму), дату последней проверки фоновым процессом, дату записи и информацию для контроля четности. Специальный фоновый процесс операционной системы СХД постоянно перечитывает все диски, выявляет и исправляет найденные несоответствия между данными и контрольной суммой. Контрольная сумма используется для проверки целостности данных и при каждом обращении к ним.

На уровне передачи данных от гипервизоров до СХД целостность данных обеспечивает протокол Fibre Channel. Fibre Channel спроектирован комитетом T11 исходя из идеи сквозного контроля целостности данных. Каждый фрейм Fibre Channel имеет четырехбайтовое поле содержащее контрольную сумму заголовка и данных фрейма. Контрольная сумма располагается в конце фрейма, между данными и четырехбайтовым примитивом, завершающим фрейм. Приемник получает фрейм, бит за битом, и считает контрольную сумму. Когда получатель получает примитив конца фрейм, он сравнивает рассчитанную контрольную сумму с полученной. В случае совпадения фрейм принимается, в противном случае помечается как поврежденный и может быть отброшен.

На уровне хостов надежность атомарного сервера не имеет высокого значения ввиду того что поверх аппаратных платформ установлены гипервизоры виртуализации. Очевидно, что при выходе из строя одного гипервизора виртуальные машины запустятся на любом другом гипервизоре кластера. Гипервизоры используют технику «сердцебиения» для определения работоспособности друг друга в кластере. Одновременно на двух независимых хранилищах гипервизоры создают специальную область, в которую периодически пишут свой статус. Если гипервизор не обновляет свой статус в течение определенного времени, то его соседи по кластеру делают вывод, что он неработоспособен и включают виртуальные машины на своих мощностях. Данная техника работает через сеть хранения данных, а значит не подвержена

проблемам связности в сети Ethernet.

В качестве операционной системы сервера управления выбран дистрибутив Debian операционной системы GNU/Linux. Debian следует принципам свободного программного обеспечения, новые версии Debian не выпускаются до тех пор, пока они не будут готовы. Разработчики не связаны каким-либо графиком, они не должны торопиться, чтобы завершить всё к какому-то сроку. Сообщество Debian уделяет большое внимание качеству и безопасности программного обеспечения. Сопровождение пакетов является относительно регламентированной деятельностью, оно хорошо документировано и даже строго регулируется [21]. В результате, пакет должен соответствовать стандартам, устанавливаемым политикой Debian [22]. За каждым пакетом на добровольной основе закрепляется разработчик, который и занимается сопровождением программного продукта: подготовкой обновлений, исправлением ошибок, о которых сообщают пользователи. Существует система отслеживания ошибок и система отслеживания уязвимостей. В совокупности с тем фактом, что программное обеспечение является открытым — это предоставляет возможность отслеживать, предлагать исправления ошибок, либо проводить аудит безопасности любому человеку в мире. Инфраструктура распространения программного обеспечения в популярных дистрибутивах GNU/Linux использует контрольные суммы и электронные подписи [14] [15].

Сервер системы управления защищен от сетевых атак тремя уровнями сетевых экранов. От сетевых атак из-за пределов инфраструктуры ТПУ сервер защищает граничный сетевой экран, находящийся в ведении Управления по информатизации ТПУ. Периметр инфраструктуры центра обработки данных ТПУ контролируется дополнительным сетевым экраном, администрирование которого осуществляет Главный информационный узел ТПУ. На сервере управления установлен и настроен программный продукт Netfilter, встроенный в ядро Linux начиная с версии 2.4.

Соединения администраторов и управляемых узлов защищается при помощи современных техник инициализации шифрованного соединения с

последующим шифрованием всей передаваемой информации. Алгоритмы установления шифрованного соединения и алгоритмы шифрования соединения отвечают современным требованиям безопасности. Подробнее этот вопрос рассмотрен в разделе 2 настоящей работы.

Администраторы проинструктированы о необходимости защищать парольной фразой закрытый ключ, используемый ими для аутентификации.

6.6 Требования к аппаратному и программному обеспечению

Система управления накладывает требования к аппаратному обеспечению исходя из числа узлов, подвергающихся управлению, частоты обновления состояния конфигурации агентами, количеством ресурсов, описанных в конфигурации каждого узла. В минимальной конфигурации сервер управления требует наличия двух ядер процессора и 1ГБ оперативной памяти. Для комфортного управления узлами в количестве 1000 штук рекомендуется использовать от двух до четырех ядер процессора и не менее 4ГБ оперативной памяти.

Система может быть развернута на дистрибутивах:

- Debian Linux 6 “Squeeze”, 7 “Wheezy”, 8 “Jessie”;
- CentOS Linux 5, 6, 7;
- Red Hat Enterprise Linux 5, 6, 7;
- Ubuntu 12.04 LTS “Precise Pangolin”, 14.04 LTS “Trusty Tahr”, 15.04 “Vivid Vervet”;
- Fedora 22, 21.

Список дистрибутивов операционной системы GNU/Linux, где может быть развернута система управления не ограничивается вышеприведенным. В их число входит любой дистрибутив, пакетная база которого основана на дистрибутивах из списка выше, а также другие независимые проекты. Такие как:

- SUSE Linux;

- Gentoo Linux;
- Arch Linux.

Кроме дистрибутивов GNU/Linux систему правления инфраструктурой можно развернуть на Unix подобных операционных системах:

- Oracle Solaris;
- AIX;
- FreeBSD;
- OpenBSD;
- HP-UX.

Реализовано управление агентами на узлах под управлением операционных систем:

- Centos 5, 6, 7;
- Debian 6, 7, 8;
- Oracle Linux 6, 7;
- Gentoo Linux.

6.7 Влияние данной работы на существующую инфраструктуру

Перед началом данной работы обеспечение безопасности системы управления инфраструктурой центра обработки данных ТПУ имела хаотичный спонтанно организующийся характер.

Учетные данные администратора атомарной виртуальной машины определялись исполнителем в момент её создания. Зачастую всё сводилось к установке пароля администратора, который знали все сотрудники отдела. Не существовало механизма оперативной смены учетных данных с правами администратора на случай их компрометации.

Настройки удаленного доступа также определялись администратором в момент создания виртуальной машины и зачастую оставлялись «по умолчанию». При этом настройки «по умолчанию» имеют ряд проблем безопасности, к

примеру, разрешен удаленный вход администратора под обезличенной учетной записью.

Обновления безопасности программного обеспечения устанавливались вручную после того как административному персоналу становилось известно о наличии уязвимостей в том или ином программном пакете. Установка обновлений безопасности на большое количество обслуживаемых узлов в ручном режиме требовала больших трудозатрат и неизбежно приводила к ошибкам обусловленным человеческим фактором. Администратор или группа администраторов могла легко пропустить какой-либо узел. Тем самым оставив его уязвимым для атак злоумышленников.

Отсутствовал централизованный сбор журналируемой информации, т.е. журналы хранились только локально на самих виртуальных машинах. Таким образом потенциальный взломщик имел возможность скрыть последствия своего проникновения отредактировав соответствующие журналы.

6.7.1 Результирующее состояние системы управления инфраструктурой

На момент окончания практической реализации данной работы в распоряжении сотрудников Главного информационного узла ТПУ была система управления инфраструктурой, отвечающая всем поставленным задачам и решающая наблюдавшиеся на начальном этапе проблемы.

Учетные данные атомарной виртуальной машины определяются исходя из её конфигурации на сервере системы управления инфраструктурой. Что позволяет оперативно предоставлять права и лишать прав конкретных администраторов, либо групп администраторов.

Настройки удаленного доступа определены на сервере системы управления инфраструктурой. Это гарантирует однородность настроек. Упрощает поддержку конфигураций удаленного доступа в актуальном

состоянии, т.е. отвечающим существующим на данный момент требованиям безопасности.

Автоматическая установка последних обновлений безопасности снижает вероятность успешной атаки на узлы при помощи эксплуатации известных уязвимостей.

Выделенный сервер сбора журналируемой информации регистрирует события, происходящие на управляемых узлах. Таким образом, если потенциальный злоумышленник успешно получил контроль над тем или иным узлом и смог устранить последствия своего проникновения, то эта информация сохранится на сервере сбора журналов. Это облегчит анализ ситуации и позволит административному персоналу принять соответствующие меры.

Косвенным положительным эффектом существования централизованной системы управления инфраструктурой является автоматическое документирование состояния узлов инфраструктуры. В свою очередь отслеживание конфигурации на сервере системы управления при помощи сервиса контроля версий позволяет проследить историю изменений на узлах.

6.7.2 Угрозы, привнесенные разработанной системой управления инфраструктурой

Комплекс мер примененных к системе управления инфраструктурой центра обработки данных в полном объеме устраняет описанные выше проблемы. Однако, одновременно с этим привносит новые угрозы.

Использование техники аутентификации по ассиметричным ключам ведет к угрозе безопасности в случае компрометации ключа администратора. Для снижения вероятности компрометации ключа администратора приняты соответствующие организационные меры, а именно: рекомендует защищать закрытый ключ парольной фразой, хранить ключ на отчуждаемом от рабочей станции носителе, не оставлять носитель с закрытым ключом без личного

присмотра. Установлен период принудительной смены ключевой пары размером в два года.

Возможность централизованно манипулировать учетными данными на большом количестве узлов несет в себе угрозу безопасности, во-первых, ввиду человеческого фактора, во-вторых, ввиду того что потенциальный злоумышленник может воспользоваться данным функционалом и тем самым установить контроль над всеми управляемыми узлами. Этот фактор накладывает на администраторов повышенную ответственность к защите сервера управления. Однако, идея состоит в том, что контролировать ситуацию и поддерживать порядок на одном узле легче, чем на многих.

Автоматическая установка обновлений безопасности несет в себе угрозу стабильности систем из-за некорректно сформированных обновлений. Обновлений, несущих в себе ошибки программного кода. Обновления, предоставляемые разработчиками используемых в центре обработки данных ТПУ, дистрибутивов достаточно хорошо тестируются перед их публикацией. Однако, несмотря на это все виртуальные машины центра обработки данных подвергаются регулярному резервному копированию, что позволяет устранить даже самые драматические последствия обновлений.

Заключение

Реализованная система отвечает всем требованиям поставленной задачи. Безопасность управления серверами, находящимися в ведении ГИУ ТПУ обеспечивается за счет механизмов аутентификации, авторизации и шифрования в соответствии с современными требованиями к стойкости алгоритмов и длине ключей.

Для защиты от атак посредника используется аутентификация на всех этапах взаимодействия: системы управления и серверов, администратора и системы управления, администратора и серверов. Аутентификация осуществляется при помощи криптографических алгоритмов асимметричного шифрования. При подключении нового сервера к системе, на сервер устанавливается агент puppet. Агент, совершая первый запрос к серверу, генерирует ключевую пару: открытый ключ и закрытый ключ, затем открытый ключ передается системе в виде запроса на выпуск сертификата. Коммуникация между сервером и системой возможна только после того, как администратор системы подпишет данный запрос и выпустит сертификат.

Для защиты от атак типа подбора на все серверы устанавливается программный продукт fail2ban, который отслеживает неудавшиеся попытки авторизоваться по протоколу ssh и, в случае превышения установленных лимитов, временно блокирует атакующего. Данная стратегия исключает возможность подбора пароля в разумные промежутки времени.

Для защиты от эксплуатации уязвимостей предусмотрен механизм автоматической проверки наличия обновлений безопасности в источниках дистрибутивов операционной системы.

Система обеспечивает централизованное администрирование из одной точки, т. е. с узла управления. Система спроектирована и реализована исходя из требования к обеспечению профилирования серверов и пользователей. Поддерживаются группы доступа. За счет того, что на каждом подключенном к системе сервере установлен агент, который инициирует подключение к системе,

решается проблема управления узлами, находящимися в закрытых подсетях ЦОД.

Отказ системы управления не может привести управляемые серверы в нештатный режим, так как вся конфигурация состояния сервера хранится на каждом сервере локально, включая пользователей их публичные ключи. Единственным негативным явлением, в данном случае, будет неспособность серверами получить обновление своего состояния.

Все примененные компоненты системы являются бесплатными. Их использование допустимо в коммерческой среде. Код данных приложений открыт.

Список литературы

1. Эви Немет, Гарт Снайдер, Трент Хейн, Бэн Уэйли. Unix и Linux: руководство системного администратора. 4-е изд. Москва: ООО “И.Д. Вильямс”, 2012.
2. State of Cybersecurity: Implications for 2015 [Электронный ресурс] // Information Technology - Information Security – Information Assurance: [сайт]. URL: http://www.isaca.org/cyber/Documents/State-of-Cybersecurity_Res_Eng_0415.pdf (дата обращения: 25.12.2015).
3. Kaspersky Security Bulletin 2006 [Электронный ресурс] // Сетевая штаб-квартира экспертов «Лаборатории Касперского»: [сайт]. URL: https://securelist.ru/analysis/ksb/987/kaspersky-security-bulletin-2006-razvitie-vredonosny-h-programm/ (дата обращения: 22.05.2016).
4. Kaspersky Security Bulletin 2015 [Электронный ресурс] // Сетевая штаб-квартира экспертов «Лаборатории Касперского»: [сайт]. URL: https://securelist.ru/analysis/ksb/27519/kaspersky-security-bulletin-2015-evolyuciya-ugroz-informacionnoj-bezopasnosti-v-biznes-srede/ (дата обращения: 22.05.2016).
5. Trustwave 2015-Global-Security-Report [Электронный ресурс] // Trustwave: Smart Security On Demand: [сайт]. URL: https://www2.trustwave.com/rs/815-RFM-693/images/2015_TrustwaveGlobalSecurityReport.pdf (дата обращения: 25.12.2015).
6. 2015 Norton Cybercrime Report [Электронный ресурс] // Symantec - Global Leader In Next - Generation Cyber Security: [сайт]. URL: https://www4.symantec.com/mktginfo/whitepaper/ISTR/21347931_GA-internet-security-threat-report-volume-20-2015-appendices.pdf (дата обращения: 25.12.2015).
7. IX Евразийский форум информационной безопасности и информационного взаимодействия [Электронный ресурс] // МВД России: [сайт]. URL: http://

mvd.ru/news/item/1033853 (дата обращения: 25.12.2015).

8. Гатчин Ю. А., Сухостат В. В. Теория информационной безопасности и методология защиты информации. СПб.: СПбГУ ИТМО, 2010.
9. EMC. От хранения данных к управлению информацией. 1-е изд. СПб.: Питер, 2010.
10. Брюс Ш. Секреты и ложь. Безопасность данных в цифровом мире. СПб.: Питер, 2003.
11. Таненбаум Э. Компьютерные сети. 5-е изд. СПб.: Питер, 2012.
12. Guy R.K. How to Factor a Number // Fifth Manitoba Conference on Numeral Mathematics Congressus Numerantum. Winnipeg. 1976.
13. Gardner M. A New Kind of Cipher That Would Take Millions of Year to Break // Scientific American. August 1977. No. 237. pp. 120-124.
14. Atkins D., Graff M., Lenstra A.K., Leyland R. The Magic Words are Squeamish Ossifrage // Advances in Cryptology ASIA CRYPT'94. Proceedings. 1994. pp. 263-277.
15. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd ed. New-York: Wiley, 2015.
16. Common Vulnerabilities and Exposures [Электронный ресурс] // CVE: [сайт]. URL: https://cve.mitre.org/cve/index.html (дата обращения: 25.12.2015).
17. Securing Debian Manual [Электронный ресурс] // Debian: [сайт]. URL: https://www.debian.org/doc/manuals/securing-debian-howto/ch7.en.html#s-deb-pack-sign (дата обращения: 25.12.2015).
18. Checking a Package's Signature [Электронный ресурс] // CentOS Project: [сайт]. URL: https://www.centos.org/docs/5/html/Deployment_Guide-en-US/s1-check-rpm-sig.html (дата обращения: 25.12.2015).
19. RFC4251: The Secure Shell (SSH) Protocol Architecture [Электронный ресурс] // Internet Engineering Task Force (IETF): [сайт]. URL: https://www.ietf.org/rfc/rfc4251.txt (дата обращения: 22.05.2016).

20. RFC2743: Generic Security Service Application Program Interface Version 2, Update 1 [Электронный ресурс] // Internet Engineering Task Force (IETF): [сайт]. URL: https://tools.ietf.org/html/rfc2743 (дата обращения: 22.05.2016).
21. RFC4462: Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the Secure Shell (SSH) Protocol [Электронный ресурс] // Internet Engineering Task Force (IETF): [сайт]. URL: https://tools.ietf.org/html/rfc4462 (дата обращения: 22.05.2016).
22. SSH General Commands Manual [Электронный ресурс] // OpenBSD manual pages: [сайт]. URL: http://man.openbsd.org/ssh.1 (дата обращения: 22.05.2015).
23. OpenSSH: Legacy Options [Электронный ресурс] // OpenSSH: [сайт]. URL: http://www.openssh.com/legacy.html (дата обращения: 22.05.2016).
24. FIPS PUB 186-2 DIGITAL SIGNATURE STANDARD (DSS) // National Institute of Standards and Technology. URL: http://csrc.nist.gov/publications/fips/archive/fips186-2/fips186-2.pdf (дата обращения: 22.05.2016).
25. FIPS PUB 186-4 Digital Signature Standard (DSS) // National Institute of Standards and Technology. URL: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf (дата обращения: 22.05.2016).
26. OpenSSH 6.5 released [Электронный ресурс] // LWN.net: [сайт]. URL: https://lwn.net/Articles/583485/ (дата обращения: 22.05.2016).
27. OpenSSH 5.7 released [Электронный ресурс] // LWN.net: [сайт]. URL: https://lwn.net/Articles/424392/ (дата обращения: 22.05.2016).
28. System Administration: Yellow Pages, part 1 [Электронный ресурс] // The Public's Library and Digital Archive: [сайт]. URL: http://www.ibiblio.org/pub/Linux/docs/LDP/linuxfocus/English/July2001/article148.shtml (дата обращения: 22.05.2016).
29. End of Features (EOF) Planned for Future Releases of Oracle Solaris [Электронный ресурс] // Oracle | Integrated Cloud Applications and Platform Services: [сайт]. URL: http://www.oracle.com/technetwork/systems/end-of-

notices/eonsolaris11-392732.html (дата обращения: 22.05.2016).

30. ITU-T X.500 [Электронный ресурс] // ITU: Committed to connecting the world: [сайт]. URL: http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11732 (дата обращения: 22.06.2016).
31. RFC4511: Lightweight Directory Access Protocol (LDAP): The Protocol [Электронный ресурс] // Internet Engineering Task Force: [сайт]. URL: https://tools.ietf.org/html/rfc4511 (дата обращения: 22.05.2016).
32. Security Overview [Электронный ресурс] // Puppet Documentation: [сайт]. URL: https://docs.puppet.com/mcollective/security.html (дата обращения: 22.05.2016).
33. Security [Электронный ресурс] // Chef Docs: [сайт]. URL: https://docs.chef.io/server_security.html (дата обращения: 22.05.2016).
34. An Overview of Chef [Электронный ресурс] // Chef Docs: [сайт]. URL: https://docs.chef.io/chef_overview.html (дата обращения: 22.05.2016).
35. Installation [Электронный ресурс] // Ansible Documentation: [сайт]. URL: http://docs.ansible.com/ansible/intro_installation.html (дата обращения: 22.05.2016).
36. Introduction [Электронный ресурс] // Ansible Documentation: [сайт]. URL: http://docs.ansible.com/ansible/intro.html (дата обращения: 22.05.2016).
37. SaltStack Documentation [Электронный ресурс] // CloudOps, ITOps & DevOps: [сайт]. URL: https://docs.saltstack.com/en/latest/ (дата обращения: 22.05.2016).
38. Sendmail [Электронный ресурс] // The Architecture of Open Source Applications: [сайт]. URL: http://www.aosabook.org/en/sendmail.html (дата обращения: 22.05.2016).
39. RFC 3164: The BSD syslog Protocol [Электронный ресурс] // Internet Engineering Task Force: [сайт]. URL: https://tools.ietf.org/html/rfc3164 (дата обращения: 22.05.2016).
40. Syslog Standardization [Электронный ресурс] // rsyslog: [сайт]. URL: http://www.rsyslog.com/doc/syslog_parsing.html (дата обращения: 22.05.2016).

41. RFC5424: The Syslog Protocol [Электронный ресурс] // Internet Engineering Task Force: [сайт]. URL: https://tools.ietf.org/html/rfc5424 (дата обращения: 22.05.2016).
42. RFC5425: Transport Layer Security (TLS) Transport Mapping for Syslog [Электронный ресурс] // Internet Engineering Task Force: [сайт]. URL: https://tools.ietf.org/html/rfc5425 (дата обращения: 22.05.2016).
43. RFC5426: Transmission of Syslog Messages over UDP [Электронный ресурс] // Internet Engineering Task Force: [сайт]. URL: https://tools.ietf.org/html/rfc5426 (дата обращения: 22.05.2016).
44. RFC5427: Textual Conventions for Syslog Management [Электронный ресурс] // Internet Engineering Task Force: [сайт]. URL: https://tools.ietf.org/html/rfc5427 (дата обращения: 22.05.2016).
45. RFC5848: Signed Syslog Messages [Электронный ресурс] // Internet Engineering Task Force: [сайт]. URL: https://tools.ietf.org/html/rfc5848 (дата обращения: 22.05.2016).
46. Datagram Transport Layer Security (DTLS) Transport Mapping for Syslog [Электронный ресурс] // Internet Engineering Task Force: [сайт]. URL: https://tools.ietf.org/html/rfc6012 (дата обращения: 22.05.2016).
47. Transmission of Syslog Messages over TCP [Электронный ресурс] // Internet Engineering Task Force: [сайт]. URL: https://tools.ietf.org/html/rfc6587 (дата обращения: 22.05.2016).
48. The syslog-ng Open Source Edition 3.7 Administrator Guide [Электронный ресурс] // BalaBit IT Security: [сайт]. URL: https://www.balabit.com/sites/default/files/documents/syslog-ng-ose-latest-guides/en/syslog-ng-ose-guide-admin/html/index.html (дата обращения: 22.05.2016).
49. changing the default syslog daemon for lenny? [Электронный ресурс] // debian-devel Jan 2008 by thread: [сайт]. URL: https://lists.debian.org/debian-devel/2008/01/thrd3.html#01002 (дата обращения: 22.05.2016).

50. DistroWatch Weekly, Issue 663, 30 May 2016 [Электронный ресурс] // DistroWatch.com: Put the fun back into computing. Use Linux, BSD: [сайт]. URL: http://distrowatch.com/weekly.php?issue=20160530 (дата обращения: 22.05.2016).
51. GNU/Linux Distribution Timeline 12.10 [Электронный ресурс] // GNU/Linux Distribution Timeline: [сайт]. URL: http://futurist.se/gldt/2012/10/29/gnulinux-distribution-timeline-12-10/ (дата обращения: 22.05.2016).
52. Home [Электронный ресурс] // Gentoo Packages: [сайт]. URL: https://packages.gentoo.org/ (дата обращения: 22.05.2016).
53. Statistics [Электронный ресурс] // Gentoo Portage Overlays: [сайт]. URL: https://gpo.zugaina.org/Statistics (дата обращения: 12.05.2016).
54. DistroWatch Page Hit Ranking [Электронный ресурс] // DistroWatch.com: Put the fun back into computing. Use Linux, BSD: [сайт]. URL: http://distrowatch.com/dwres.php?resource=popularity
55. Выпущен Debian 8 "Jessie" [Электронный ресурс] // Debian -- Новости: [сайт]. URL: https://packages.debian.org/stable/allpackages?format=txt.gz (дата обращения: 22.05.2016).
56. Долгосрочная поддержка Debian [Электронный ресурс] // Debian Wiki: [сайт]. URL: https://wiki.debian.org/ru/LTS (дата обращения: 22.05.2016).
57. LogAnalyzer 3.6.6 (v3-stable) [Электронный ресурс] // Adiscon LogAnalyzer: [сайт]. URL: http://loganalyzer.adiscon.com/downloads/loganalyzer-3-6-6-v3-stable/ (дата обращения: 22.05.2016).

Приложение А

Экземпляры объявления узлов в файле manifests/site.pp

```
# Gentoo
node 'zero.tpu.ru' {
    include nagios::server
    include tpu_ssh_acc::mind
    include tpu_ssh_acc::backuppc
    include tpu_ssh_acc::net
    # vvish@tpu.ru
    sudo::conf { 'vvish': priority => 20,  content  => "vvish ALL=NOPASSWD:
ALL" }
}

# OracleLinux 6
# owner: N/A
node 'echo.tpu.ru' {
    include tpu_mirror_repository
    include tpu_sshd::sec_lvl2
    include tpu_ssh_acc::mind
    include tpu_ssh_acc::backuppc
}
# CentOS 6
# owner: N/A
node 'webserver-5.tpu.ru' {
    include centos_template
    include tpu_sshd::sec_lvl2
}

# CentOS 7
# owner: oys1@tpu.ru
node 'mirror.tpu.ru' {
    include tpu_sshd::sec_lvl1
    include tpu_ssh_acc::mind
    include tpu_mirror_repository
    include tpu_autoupdate
    include tpu_nagios
}
```

Приложение Б

Исходный текст файла modules/tpu_sshd/manifests/init.pp

```
# This module configure SSH daemon and fail2ban on ssh port.
#
# Class: tpu_sshd - do basic preparation. Install and run sshd.
#        Install, configure and run file2ban.
# Class: tpu_sshd::sec_lvl1 - configure sshd to high secure mode.
#        Enabled only publickey authentication. No permit root login.
# Class: tpu_sshd::sec_lvl2 - configure sshd to medium secure mode.
#        Enabled publickey and password authentication. No permit root
#        login.
#

class tpu_sshd {
     case $::osfamily {
          'Gentoo': {
               package {'ssh':
                    ensure => present,
               }
               service { 'sshd':
                    subscribe => File[sshdconfig],
                    require => Package['ssh'],
                    ensure => 'running',
                    enable => 'true',
               }
          }
          'Debian': {
               package { 'openssh-server':
                    ensure => latest
               }
               service { 'ssh':
                    subscribe => File[sshdconfig],
                    require => Package['openssh-server'],
                    ensure => 'running',
                    enable => "true",
               }
               package { fail2ban: ensure => latest } ->
               service { fail2ban: ensure => running, }
               file { '/etc/fail2ban/jail.local' :
                    ensure => present,
                    owner => root,
                    group => root,
                    mode => '0600',
                    source                                          =>
"puppet:///modules/tpu_sshd/jail.local.$osfamily",
                    require => Package['fail2ban'],
                    notify => Service['fail2ban'],
               }
               file { '/etc/fail2ban/jail.conf' :
                    ensure => present,
                    owner => root,
                    group => root,
                    mode => '0600',
                    source                                          =>
"puppet:///modules/tpu_sshd/jail.conf.$osfamily",
```

```
                    require => Package['fail2ban'],
                    notify => Service['fail2ban'],
               }
          }
          'RedHat', 'CentOS': {
               require ::tpu_mirror_repository
               package { 'openssh-server':
                    ensure => latest
               }
               service { 'sshd':
                    subscribe => File[sshdconfig],
                    require => Package['openssh-server'],
                    ensure => 'running',
                    enable => "true",
               }
               package { fail2ban: ensure => latest } ->
               service { fail2ban: ensure => running, }
               file { '/etc/fail2ban/jail.local' :
                    ensure => present,
                    owner => root,
                    group => root,
                    mode => '0600',
                    source                                                    =>
"puppet:///modules/tpu_sshd/jail.local.${::osfamily}.${::os_maj_version}"
,
                    require => Package['fail2ban'],
                    notify => Service['fail2ban'],
               }
               file { '/etc/fail2ban/jail.conf' :
                    ensure => present,
                    owner => root,
                    group => root,
                    mode => '0600',
                    source                                                    =>
"puppet:///modules/tpu_sshd/jail.conf.${::osfamily}.${::os_maj_version}",
                    require => Package['fail2ban'],
                    notify => Service['fail2ban'],
               }
          }
     }
}

class tpu_sshd::sec_lvl1 inherits tpu_sshd {
     case $::osfamily {
          'Gentoo': {
               file { 'sshdconfig':
                    name => '/etc/ssh/sshd_config',
                    owner => root,
                    group => root,
                    mode => '0600',
                    source                                                    =>
'puppet:///modules/tpu_sshd/sshd_config_sl1',
                    require => Package['ssh'],
               }
          }
          'Debian', 'RedHat': {
               file { 'sshdconfig':
                    name => '/etc/ssh/sshd_config',
```

```
                    owner => root,
                    group => root,
                    mode => '0600',
                    source                                          =>
'puppet:///modules/tpu_sshd/sshd_config_sl1',
                    require => Package['openssh-server'],
                }
        }
    }
}

class tpu_sshd::sec_lvl2 inherits tpu_sshd {
    case $::osfamily {
        'Gentoo': {
            file { 'sshdconfig':
                    name => '/etc/ssh/sshd_config',
                    owner => root,
                    group => root,
                    mode => '0600',
                    source                                          =>
'puppet:///modules/tpu_sshd/sshd_config_sl2',
                    require => Package['ssh'],
                }
        }
        'Debian', 'RedHat': {
            file { 'sshdconfig':
                    name => '/etc/ssh/sshd_config',
                    owner => root,
                    group => root,
                    mode => '0600',
                    source                                          =>
'puppet:///modules/tpu_sshd/sshd_config_sl2',
                    require => Package['openssh-server'],
                }
        }
    }
}
```

Приложение В

Экземпляры определения учетных записей администраторов в файле modules/tpu_ssh_acc/manifests/init.pp

```
class tpu_ssh_acc {
    require ::tpu_sshd
}
#
# Class: tpu_ssh_acc::mind - members of Main Information Department.
#        Have full access to all nodes.
#
class tpu_ssh_acc::mind inherits tpu_ssh_acc {

# user1
    user { 'user1':
        ensure     => present,
        forcelocal => true,
        managehome => true
    }
    ssh_authorized_key { 'user1@tpu.ru ssh auth key':
        name     => 'user1@tpu.ru',
        ensure   => present,
        key                                                                 =>
'AAAAB3NzaC1yc2EAAAADAQABAAABAQClDXy6QWJ9DoRoQ4NUPSfudDTEgQNjAHbrvO/LFTK/
mVrkGAKyTEwQhhdBuR1ZtBLSxjNdurnKBTHr6Z15o44Hfj0NjfXRHkjcSEa8UJVG53cY0sB9s
rEhKjL29eDkivxQg2MEWbtAVjF6GGBm1mI9KKliegiOC/M5iOn8X5T9+g/Y1rWo0J7R4oOQB6
hPYJHKBk23T/tknqyrdnyw3L//ziteznBfxrw3oZWPcUDFwuE2HaTAYcZb53x7Gse0vmDmCAe
7yAc7bNdOck+HP3/wqXgA8n/fz3XVQyMSnCGMesf3GK8auPnr4W7Se4t2VmU6lD0pmur3z7z4
Ot7I2r/p',
        type     => 'ssh-rsa',
        user     => 'user1',
        require  => User['user1'],
    }
    sudo::conf { 'user1':
        priority => 10,
        content  => "user1  ALL=NOPASSWD: ALL"
    }
}
class tpu_ssh_acc::net inherits tpu_ssh_acc {
# user7
    user { 'user7':
        ensure     => present,
        forcelocal => true,
        managehome => true
    }
    ssh_authorized_key { 'user7@tpu.ru ssh auth key':
        name     => 'user7@tpu.ru',
        ensure   => present,
        key                                                                 =>
'AAAAB3NzaC1yc2EAAAABIwAAAQEAvIcGl/dpFXKUxkWu9izVL2CIS+xvskHPKSAqxeSKwcCZ
bzSjoy7V7WZqoxXStXyuGnCsDH1ZBxF/CRV6wAKt6lxdPA1bRHNWpd0YnkCIOYn3P6jlHIpaj
t/aVhlUZn0jKLujkMjfIQs+ju29OfTfcDZY6JD6Yxy9k64XvvtNarceeJPuO6f0xLYyPDjAPo
kAI9Qpio2qWqQKoyKDKw3ML/OqWs3yz8TsFkWC0toY/CSSYXS2u0N8rLW6/XI1W4roSpb7l75
tweW04+NFkMx49Fdy0k4vIcdqh8gykrrP3AnFUnYRlbuqEGX6ITH8jXbSduHEzy86NqrKBdBe
2eMwdw==',
```

```
        type     => 'ssh-rsa',
        user     => 'user7',
        require  => User['user7'],
    }
    sudo::conf { 'user7':
        priority => 10,
        content  => "user7  ALL=NOPASSWD: ALL"
    }
}
#
# Class: tpu_ssh_acc::ns - service accounts for DNS replication
#
class tpu_ssh_acc::ns inherits tpu_ssh_acc {
# user9
    user { 'user9':
        ensure     => present,
        forcelocal => true,
        managehome => true
    }
    ssh_authorized_key { 'user9@tpu.ru ssh auth key':
        name     => 'user9@tpu.ru',
        ensure   => present,
        key                                                                  =>
'AAAAB3NzaC1yc2EAAAADAQABAAABAQCpWMysZHMH1VsvIFMOtopYfOqbNsTG5vOqPNs3nbUu
5iIOECEEyc0OHLI5eYodIYKYsoiZaRl7DuchmwtnnEMg53858dejH31woVLZQsLVkD0FU5yAf
b1rLKgxT0nfqBYZUGdiQaU3ByrfpVp1xU4WQTYk1VgILF9/mLiTX20wkvlcfHkSlXzoRmhZN7
paXmnE7MPpAPKeBN73vUdRMyQVAjdx4E8U8lGrHIAleX9QwpxhZJ1zuVl5OLW4/hnb3CDgdjo
atHmaF+KQs9uQb8EbVBYMH7hOfkvpaLBzh9YaRmLqRQlmgvnKLipJqq4BaqLa/qr/fk0oUGD1
CDkvstXL',
        options  => 'from="109.123.152.2"',
        type     => 'ssh-rsa',
        user     => 'user9',
        require  => User['user9'],
    }
    sudo::conf { 'user9':
        priority => 10,
        content  => "user9  ALL=NOPASSWD: ALL"
    }
}
```

Приложение Г

Исходный текст файла modules/tpu_autoupdate/manifests/init.pp

```
# Class: tpu_autoupdate
#
# This module enables automatic periodic package updates.
#

class tpu_autoupdate {

      require tpu_mirror_repository

      case $::osfamily {
            'Debian': {
                  package { 'unattended-upgrades': ensure => latest }
                  package { 'apt-listchanges': ensure => latest }
                  file { '/etc/apt/apt.conf.d/50unattended-upgrades' :
                        ensure => present,
                        owner => root,
                        group => root,
                        mode => '0644',
                        source   =>   "puppet:///modules/tpu_autoupdate/apt-
50unattended-upgrades.${::osfamily}.${::os_maj_version}",
                        require => Package['unattended-upgrades'],
                  }
                  file { '/etc/apt/apt.conf.d/20auto-upgrades' :
                        ensure => present,
                        owner => root,
                        group => root,
                        mode => '0644',
                        source   =>   "puppet:///modules/tpu_autoupdate/apt-
20auto-upgrades.${::osfamily}.${::os_maj_version}",
                        require => Package['unattended-upgrades'],
                  }
                  file { '/etc/apt/listchanges.conf' :
                        ensure => present,
                        owner => root,
                        group => root,
                        mode => '0644',
                        source   =>   "puppet:///modules/tpu_autoupdate/apt-
listchanges.conf.${::osfamily}.${::os_maj_version}",
                        require => Package['apt-listchanges'],
                  }
            }
            'CentOS', 'RedHat': {
                   package { 'yum-cron': ensure => latest } ->

                  case $::os_maj_version {
                        '6': {
                              file { '/etc/sysconfig/yum-cron' :
                                    ensure => present,
                                    owner => root,
                                    group => root,
                                    mode => '0644',
                                    source                                    =>
"puppet:///modules/tpu_autoupdate/yum-
```

```
cron.${::osfamily}.${::os_maj_version}",
                                require => Package['yum-cron'],
                                notify => Service['yum-cron'],
                           }
                           service {['yum-cron']: enable => true, ensure
=> running, } # uncomment whenever updates should be installed
#                          service {['yum-cron']: enable => false, ensure
=> stopped, } # otherwise disable auto-updates
                     }
                     '7': {
                           file { '/etc/yum/yum-cron.conf' :
                                ensure => present,
                                owner => root,
                                group => root,
                                mode => '0644',
                                source                                  =>
"puppet:///modules/tpu_autoupdate/yum-
cron.conf.${::osfamily}.${::os_maj_version}",
                                require => Package['yum-cron'],
                                notify => Service['yum-cron'],
                           }
                           service {['yum-cron']: enable => true, ensure
=> running, } # uncomment whenever updates should be installed
#                          service {['yum-cron']: enable => false, ensure
=> stopped, } # otherwise disable auto-updates
                     }
               }
          }
     }
}
```

Приложение Д

Исходный текст файла modules/tpu_mirror_repository/manifests/init.pp

```
# Class: tpu_mirror_repository
#
# This module adds mirror.tpu.ru repositories to apt/yum
#
class tpu_mirror_repository {

      case $::osfamily {
            'Debian': {
                  file { '/etc/apt/apt.conf.d/10tsk-cache-proxy' :
                        source                                                  =>
"puppet:///modules/tpu_mirror_repository/apt_tsk-cache-
proxy.${::operatingsystem}",
                  }
            }
            'RedHat', 'CentoOS', 'OracleLinux': {
                  file { '/etc/yum.repos.d/tsk-base.repo' :
                        source                                                  =>
"puppet:///modules/tpu_mirror_repository/yum_tsk-
base.repo.${::operatingsystem}",
                  }
                  file { '/etc/yum.repos.d/tsk-puppetlabs.repo' :
                        source                                                  =>
"puppet:///modules/tpu_mirror_repository/yum_tsk-
puppetlabs.repo.${::operatingsystem}",
                  }
                  file { '/etc/yum.repos.d/CentOS-Base.repo' :
                        source                                                  =>
"puppet:///modules/tpu_mirror_repository/CentOS-Base.repo.${::osfamily}",
                  }
            }
      }
}
```