Научная статья
УДК 004.451

ПОСТРОЕНИЕ ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА НА БАЗЕ СВОБОДНО РАСПРОСТРАНЯЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

✉Буйневич Михаил Викторович.
Санкт-Петербургский университет ГПС МЧС России, Санкт-Петербург, Россия.
Молчанов Дмитрий Александрович.
ООО «АИС» (Автоматические Интеллектуальные Системы), Санкт-Петербург, Россия
✉bmv1958@yandex.ru

Аннотация. Операционная система Windows является самой популярной программной платформой для построения информационных систем различного масштаба: от «настольных» до корпоративных. Этот продукт от Microsoft имеет привычный и интуитивно понятный пользовательский интерфейс, а также работает на широком спектре аппаратного обеспечения, что позволяет пользователям выбирать из различных брендов и конфигураций компьютеров. Многие пользователи и технические специалисты давно знакомы с Windows и способны освоить новые версии.

Однако в свете новых реалий использование операционной системы Windows в качестве платформы информационных систем российских организаций или предприятий не просто считается нерациональным, а является и вовсе технически невозможным. Необходим оперативный поиск альтернативного решения и переход (миграция) информационных систем на другую платформу. Очевидной альтернативой для перехода с операционной системы Windows является Linux, как единственная, действительно конкурентоспособная операционная система, удовлетворяющая всем установленным критериям и имеющая в своём множестве свободно распространяемых дистрибутивов.

В работе приведён аналитический обзор наиболее популярных дистрибутивов Linux, пригодных для построения отказоустойчивого кластера; уставлены его свойства, необходимые для дальнейшего безопасного, независимого и бесперебойного функционирования.

Выбран один из «полярных» способов миграции – разработка авторского средства автоматизации решения ее подзадач. Итогом работы является Bash-скрипт, который компании могут использовать для выбора и обоснования дистрибутива Linux и дальнейшего оперативного развёртывания (автоматизированной настройки центрального сервера и рабочих станций) на нем своей инфраструктуры. Показана его достаточно высокая эффективность через оценку временных затрат на миграцию.

Ключевые слова: миграция, свободно распространяемое программное обеспечение, дистрибутив, Bash-скрипт, отказоустойчивый кластер

Для цитирования: Буйневич М.В., Молчанов Д.А. Построение отказоустойчивого кластера на базе свободно распространяемого программного обеспечения // Науч.-аналит. журн. «Вестник С.-Петерб. ун-та ГПС МЧС России». 2023. № 2. С. 147–160.

Scientific article

BUILDING A FAILOVER CLUSTER BASED ON FREE SOFTWARE

✉**Buinevich Mikhail V.**
Saint-Petersburg university of State fire service of EMERCOM of Russia, Saint-Petersburg, Russia.
Molchanov Dmitry A.
AIS LLC, (Automatic Intelligent Systems), Saint-Petersburg, Russia
✉***bmv1958@yandex.ru***

Abstract. The Windows operating system is the most popular software platform for building information systems of various sizes: from «desktop» to corporate. This Microsoft product has a familiar and intuitive user interface and runs on a wide range of hardware, allowing users to choose from different brands and configurations of computers. Many users and technicians have long been familiar with Windows and are able to master new versions.

However, in light of the new realities, the use of the Windows operating system as a platform for the information systems of Russian organizations or enterprises is not only considered irrational, but is even technically impossible. A prompt search for an alternative solution and the transition (migration) of information systems to another platform is necessary. An obvious alternative for the transition from the Windows operating system is Linux, as the only truly competitive operating system that meets all established criteria and has in its multitude of freely distributed distributions.

This paper gives an analytical overview of the most popular Linux distributions suitable for building a fault-tolerant cluster; its properties necessary for further secure, independent and uninterrupted operation are set out.

One of the «polar» ways of migration was chosen – development of the author's tool to automate the solution of its subtasks. The result of the work is aash-script, which companies can use to select and justify the Linux distribution and further operational deployment (automated configuration of the central server and workstations) of their infrastructure on it. It is shown to be quite efficient through estimation of migration time costs.

Keywords: migration, free software, distribution, Bash-script, fault-tolerant cluster

For citation: Buinevich M.V., Molchanov D.A. Building a failover cluster based on free software // Scientific and analytical journal «Vestnik Saint-Petersburg university of State fire service of EMERCOM of Russia». 2023. № 2. P. 147–160.

Введение

Подавляющему большинству организаций и предприятий требуется реализация серверной части их информационных систем в виде кластерного исполнения. Так, например, в компании ООО «АИС» (*аббр.:* Автоматические Интеллектуальные Системы) техническим руководством было принято решение использовать именно такое ее исполнение. В пользу кластеризации, как правило, выступает повышенная (к предельным значениям) отказоустойчивость, практически неограниченная масштабируемость и такой немаловажный аргумент, как регулируемое и балансируемое распределение нагрузки.

«Старт» выполнение этой миссии был в 2018 г., и тогда было принято решение использовать в качестве основной платформы операционную систему Windows Server 2016[1]. Построенное на ней серверное «ядро» инфраструктуры выполняет роли доменного

[1] По статистике за 2020 г. более 80 % российских компаний использовали в качестве операционной системы продукт компании Microsoft, а именно OS Windows разных поколений и интерпретаций // Доля рынка операционных систем в Российской Федерации | Statcounter Global Stats. URL: https://gs.statcounter.com/os-market-share/all/russian-federation/#monthly-202001-202012 (дата обращения: 13.03.2023)

контроллера, DNS-, DHCP-, VPN- и файл-сервера. Системотехнически решение реализовано на двух физических серверах, установленных в разных зданиях и даже районах города, что «работает» на результирующую отказоустойчивость ввиду различных поставщиков электропитания и интернет-провайдеров.

Сегодня во времена большого количества санкций и ограничений, наложенных на наше государство, перед многими организациями и предприятиями остро стоит вопрос миграции своих инфраструктур на другие операционные системы, ведь такие компании, как Microsoft и Apple, прекратили распространение, лицензирование и поддержку своих продуктов на территории Российской Федерации. Более того, в стране введены ряд ограничений на закупки программного обеспечения (ПО) и коммерческое сотрудничество с иностранными поставщиками [1]. В этих условиях бизнес и государство вынуждены искать выход и подбирать решение на базе свободного распространения ПО, которое не будет зависеть от политического и санкционного давления, не прекратит работать и поддерживаться по одностороннему решению вендоров. При этом необходимо, чтобы решение имело надёжную защиту, было понятно в своём устройстве, освоении и использовании как специалистам информационной безопасности и системным администраторам, так и рядовым пользователям. Нужно, чтобы оно имело большую аудиторию сообщества, ведь это способствует упрощению решения возникающих проблем, а также стимулирует разработчиков различного ПО выпускать и поддерживать свои продукты на выбранной операционной системе. С учетом вышеизложенного, решаемая в статье задача созвучна теме и формулируется как построение отказоустойчивого кластера на базе свободно распространяемого ПО.

W2L

Такой открытой операционной системой является Linux, и это означает, что ее исходный код доступен для всех, и пользователи имеют право изменять, распространять и улучшать систему по своему усмотрению, что предоставляет большую свободу выбора и контроля над ней. Большинство ее дистрибутивов распространяются бесплатно и не прекратят осуществлять поддержку из-за политических взглядов правообладателей.

Наиболее популярными бесплатными дистрибутивами Linux, которые имеют серверные решения, являются:

1. Ubuntu Server – один из самых популярных дистрибутивов. Основан на Debian и предоставляет широкий выбор инструментов и функциональности для управления серверами, включая поддержку контейнеров, виртуализацию, облачные вычисления и многое другое. Обладает простым в использовании интерфейсом и активным сообществом поддержки.

2. CentOS (*аббр. от англ.* Community Enterprise Operating System) – свободная версия Red Hat Enterprise Linux (RHEL). Предлагает стабильную и надежную платформу для серверных решений. Обеспечивает долгосрочную поддержку и безопасность, что делает его популярным выбором для серверных окружений.

3. Debian – один из наиболее «старых» и стабильных дистрибутивов. Предоставляет широкий выбор ПО и инструментов для создания и управления серверами. Обладает активным сообществом разработчиков и предлагает надежную и безопасную операционную систему для серверных задач.

4. Fedora Server – образец поддерживаемый сообществом дистрибутивом, специально разработанным для использования в серверных окружениях. Предлагает последние версии ПО и технологий, таких как контейнеры, виртуализация и облачные вычисления. Также является лабораторией для новых функций и инноваций, что делает его привлекательным для разработчиков и технических энтузиастов.

5. openSUSE Leap является стабильным дистрибутивом, основанным на SUSE Linux Enterprise. Предоставляет полноценное решение для серверных задач с поддержкой

широкого спектра ПО. Обладает простым в использовании инсталлятором и широкими возможностями настройки, что делает его гибким выбором для серверных сред.

Глобального отличия для поставленных задач в теоретическом сравнении не обнаружено, поэтому выбрать «лидера», основываясь на умозаключениях, не представляется возможным. В конечном итоге выбор дистрибутива зависит от индивидуальных потребностей и предпочтений, например, Ubuntu предлагает простоту использования и обширную поддержку, openSUSE обладает гибкостью и инновациями, а CentOS ориентирован на стабильность и серверные среды. Поэтому итоговый выбор дистрибутива будет сделан ниже, исходя из результатов автоматизированного тестирования безопасности серверных версий.

Linux славится своей безопасностью из-за кардинально иного подхода к архитектуре и благодаря всё той же модели разработки с открытым исходным кодом; ошибки и уязвимости быстро обнаруживаются и исправляются сообществом разработчиков. Linux также предлагает различные дополнительные инструменты и возможности для обеспечения безопасности системы.

После нахождения оптимального решения, выбранный дистрибутив будет необходимо подготовить к работе: сделать первоначальную настройку, подготовить ядро доменной сети в виде отказоустойчивого кластера из двух серверов с функциями доменного контроллера, DNS-сервера, DHCP-сервера, VPN-сервера, файлового сервера, реализовать резервное копирование информации [2].

Но главный вопрос состоит в том, как можно максимально оптимизировать, сократить простой, удешевить и ускорить миграцию (под миграцией понимается перевод программной компоненты инфраструктуры предприятия или организации с Windows на Linux (*по англ.* Windows To Linux, или символично – W2L) имеющейся инфраструктуры организации или предприятия? Решением поставленной задачи являются два «полярных способа». Первый – «ручное» воспроизведение всех тестов безопасности и дальнейшая настройка каждого хоста инфраструктуры, что является весьма затратным с точки зрения времени либо вынуждает привлекать больше дорогостоящих квалицированных специалистов для ускорения – и в этом случае присутствует «человеческий фактор»: ведь может быть допущена ошибка (от незнания) либо упущение (от невнимательности, ввиду ментальной перегрузки либо монотонности процедуры).

Второй способ заключается в разработке либо покупке средства автоматизации необходимых процессов, что позволяет не только ускорить достижение имеющихся целей, но даёт возможность сэкономить на человеческом ресурсе и компетенциях привлекаемых специалистов – ведь применение автоматизации зачастую является крайне простой и посильной даже стажёрам процедурой. Присутствуют и иные, а точнее промежуточные гибридные вариации между указанными «полярными» способами. Научный и практический интерес представляет именно второй способ, средству реализации которого и посвящено основное содержание статьи.

Потребности vs возможности

Сформулируем основные потребности задачи построения отказоустойчивого кластера, которые затем могут быть отзеркалированы в возможности свободно распространяемого ПО по ее решению.

Центральный сервер

Центральные сервер в стандартном исполнении берёт на себя следующие роли:

1. Выполняет роль доменного контроллера, отвечающего за аутентификацию и авторизацию пользователей в сети организации или предприятия, обеспечивая централизованное управление учетными записями.

2. Служит файловым сервером, предоставляющим сетевое хранилище для общего доступа к электронным документам организации или предприятия.

3. Выполняет роль backup-сервера, отвечая за регулярное создание резервных копий данных и их восстановление в случае сбоя или потери.

4. Является центральной точкой мониторинга и управления для всех хостов и сетевых устройств. Собирает информацию о состоянии систем, оповещает администраторов о проблемах, помогает управлять и настраивать хосты и сеть.

5. Берёт на себя функцию VPN-сервера, обеспечивая безопасное удаленное подключение к информационной системе организации или предприятию через зашифрованный канал, позволяя сотрудникам работать удаленно и обеспечивать защищенное подключение к внутренней сети.

6. Функционирует в качестве DNS-сервера. Он хранит и обновляет информацию о доменных именах и соответствующих им IP-адресах, обеспечивает распределение запросов, разрешение имен, кеширование данных и может быть настроен как авторитетный сервер для определенных зон.

7. Выступает в качестве DHCP-сервера, автоматически настраивая сетевые параметры для клиентских устройств в сети. Он предоставляет IP-адреса, маски подсети, адреса шлюзов, DNS-серверы и другие сетевые настройки клиентам. Центральный DHCP-сервер управляет пулом доступных IP-адресов, выдает их клиентам по запросу и поддерживает динамическое обновление настроек сети.

Этих ролей центрального сервера достаточно для полноценного функционирования информационной системы большинства стандартных организаций или предприятий без специфических и узконаправленных задач. Перед началом использования сервера Linux и наделением его описанных выше ролей необходимо выполнить несколько первоначальных настроек:

1. Произвести обновление системы до последних доступных пакетов, что поможет устранить возможные уязвимости, получить последние исправления и новые (или обновленные) функции.

2. Настроить сетевое подключение сервера. Это включает присвоение статического IP-адреса, настройку DNS-серверов, настройку шлюза по умолчанию и другие сетевые параметры.

3. Принять меры по обеспечению безопасности сервера. Меры могут включать в себя настройку межсетевого экрана, установку и настройку системы обнаружения вторжений (Intrusion Detection System), настройку доступа по SSH, создание пользователей с соответствующими правами доступа и паролями и т.д.

4. Настроить логирование сервера, чтобы отслеживать его работу, обнаруживать проблемы и анализировать производительность. Для этого потребуется установить инструменты мониторинга.

В завершение рекомендуется провести тестирование сервера, чтобы убедиться, что все функции работают должным образом. Тестирование может включать проверку сетевого подключения, доступности сервисов и приложений, а также проверку безопасности.

Кластеризация

Определённо центральный и тем более единственный сервер организации или предприятия необходимо выполнять в виде кластера (как указано во введении) как минимум из двух максимально независимых друг от друга хостов. Помимо высокой отказоустойчивости (за счет резервирования) и оптимального использования ресурсов (за счет балансировки нагрузки) такое решение обеспечивает горизонтальную масштабируемость, позволяя добавлять новые узлы в кластер для увеличения общей вычислительной мощности с ростом требований и нагрузки.

При реализации кластера серверов рекомендуется учесть следующие рекомендации:

– размещение серверов кластера в разных физических местах или дата-центрах. Это поможет избежать единой точки отказа и снизит риск влияния одного сбоя на все серверы;

– размещение дублируемых серверов на разных узлах кластера. Это означает, что каждый узел имеет свою независимую копию ресурсов, таких как процессоры, память, хранилище данных и т.д. В случае отказа одного узла, другой может продолжать работу без прерывания;

– использование механизмов балансировки нагрузки для равномерного распределения запросов и задач между узлами кластера. Это поможет избежать перегрузки одного узла и обеспечит оптимальное использование ресурсов;

– регулярное резервное копирование данных, хранящихся на серверах кластера. Это позволит восстановить данные, в случае их потери или повреждения. Резервные копии должны храниться на отдельных и надежных устройствах или удаленных местах;

– реализация системы мониторинга, которая будет отслеживать состояние серверов кластера и оповещать о любых проблемах или сбоях;

– регулярное тестирование кластера серверов для проверки его работоспособности и отказоустойчивости, а также автоматическое восстановление. Также важно обновлять ПО и патчи системы, чтобы исправить уязвимости и проблемы безопасности.

Настройка рабочих станций

Первоначальная настройка рабочих станций Linux включает ряд шагов для подготовки рабочего места сотрудника к использованию.

Шаг 1. После установки операционной системы рекомендуется выполнить обновление системы до последних доступных пакетов. Это поможет обновить устаревшие программы, получить исправления безопасности и новые функции.

Шаг 2. В зависимости от потребностей сотрудника установить необходимое ПО на рабочую станцию. Это может быть офисный пакет, веб-браузер, мультимедийные приложения, инструменты разработчика и др.

Шаг 3. Установить сетевые параметры для рабочей станции, ввести ее в домен. Так как используется DHCP, необходимо убедиться, что соединение сети настроено для автоматического получения параметров.

Шаг 4. Настроить межсетевой экран, чтобы ограничить доступ к системе, проконтролировать обновления безопасности, проверить права доступа к файлам и папкам.

Шаг 5. Создать необходимые пользовательские учетные записи и сконфигурировать их права доступа. Убедиться, что каждый пользователь имеет свой уникальный аккаунт и пароль для обеспечения безопасности системы.

Выбор и обоснование средства автоматизации

Для автоматизации решения задачи **миграции W2L** – сравнительного тестирования безопасности и последующей настройки дистрибутивов Linux, установки необходимых ролей центрального сервера и пользовательских рабочих станций – авторами спроектирован и разработан универсальный Bash-скрипт. Bash (*аббр. от англ.:* Bourne Again Shell) – это командный интерпретатор командной строки (shell), один из наиболее распространенных и широко используемых в Linux и других Unix-подобных операционных системах; является продолжением и расширением оригинального интерпретатора командной строки Bourne shell (sh) [3]. Он обеспечивает широкий набор функций, таких как выполнение команд, управление файлами и директориями, перенаправление ввода/вывода, обработка переменных окружения, создание скриптов и многое другое [4].

Разработанный Bash-скрипт условно разделён на четыре части, предназначенные для разных подзадач:

Подзадача 1. Проверка безопасности различных дистрибутивов для их сравнительного анализа и дальнейшего выбора наиболее подходящего.

Подзадача 2. Исправление и улучшение безопасности выбранного дистрибутива согласно рекомендациям по безопасности операционных систем.

Подзадача 3. Настройка и наделение базового кластера всеми необходимыми ролями центрального сервера организации (предприятия).

Подзадача 4. Подготовка и введение в эксплуатацию рабочих станций сотрудников.

Далее, покажем решение каждой подзадачи, условно пронумеровав используемую итерацию Bash-скрипта как «скрипт_номер подзадачи».

Проектирование и экспериментальная проверка работоспособности Bash-скрипта

Сравнительный анализ дистрибутивов Linux

Исключительно для примера сузим пространство дистрибутивов сравнительного анализа, и для тестирования выберем ***Ubuntu Server*** (свободно распространяемое ПО, используемое авторами при построении отказоустойчивого кластера и решения задач миграции W2L, выделено шрифтом), ***openSUSE Leap*** и ***CentOS***. Автоматизированное тестирование разработанным Bash-скриптом первой итерации (скрипт_1) будет производиться на каждом из указанных дистрибутивов, установленном на виртуальную машину ***HyperV***. Для обеспечения объективности результатов тестирования уточним их системные требования, а затем выделим одинаковые мощности каждому, отталкиваясь от самого требовательного варианта.

В табл. 1 приведены минимальные системные требования, которые должны быть соблюдены для комфортной работы тестируемых дистрибутивов.

Таблица 1

Системные требования дистрибутивов

Дистрибутив	Процессор	Оперативная память (ОЗУ)	Свободное пространство на диске
openSUSE	Pentium 4 2.4 ГГц или мощнее	2 ГБ	5 ГБ
Ubuntu	Двухъядерный процессор 2 ГГц	4 ГБ	25 ГБ
CentOS	На каждый ГБ ОЗУ иметь 1 ядро процессора	1 ГБ	10 ГБ

В итоге, учитывая самый «требовательный» дистрибутив и принимая во внимание современные реалии и запросы ПО, а также сохраняя разумный запас, было принято решения использовать следующие конфигурации виртуальных машин: 30 ГБ свободного пространства на диске, 4 ГБ оперативной памяти и 4-х ядерный центральный процессор.

Скрипт_1, ориентированный на обнаружение наличия уязвимостей установленных дистрибутивов без каких-либо их первоначальных настроек, выполняет базовые проверки [5], а именно:

1) наличия вредоносных программ с помощью ***chkrootkit***;
2) системы на уязвимости с помощью ***nmap***;
3) открытых сетевых соединений (здесь и до конца списка – с помощью ***lynis***);

4) наличия незащищенных файловых разрешений;
5) активных процессов;
6) наличия обновлений системы;
7) конфигурации SSH;
8) наличия необычных аккаунтов (аккаунты с UID менее 1000);
9) наличия незащищенных сетевых сервисов;
10) системных разрешений;
11) безопасности паролей;
12) системных служб;
13) целостности системных файлов, а также осуществляет анализ системных журналов на наличие ошибок и проблем.

Изучив отчёты выполнения скрипта_1, были обнаружены отличия в ряде тестов. Ниже приведены результаты проверок № 2, 3, 9 соответственно.

Согласно отчёту ***nmap***, представленному на рис. 1, явным аутсайдером является дистрибутив ***CentOS 7*** – у него четыре открытых порта против двух у каждого из конкурентов.

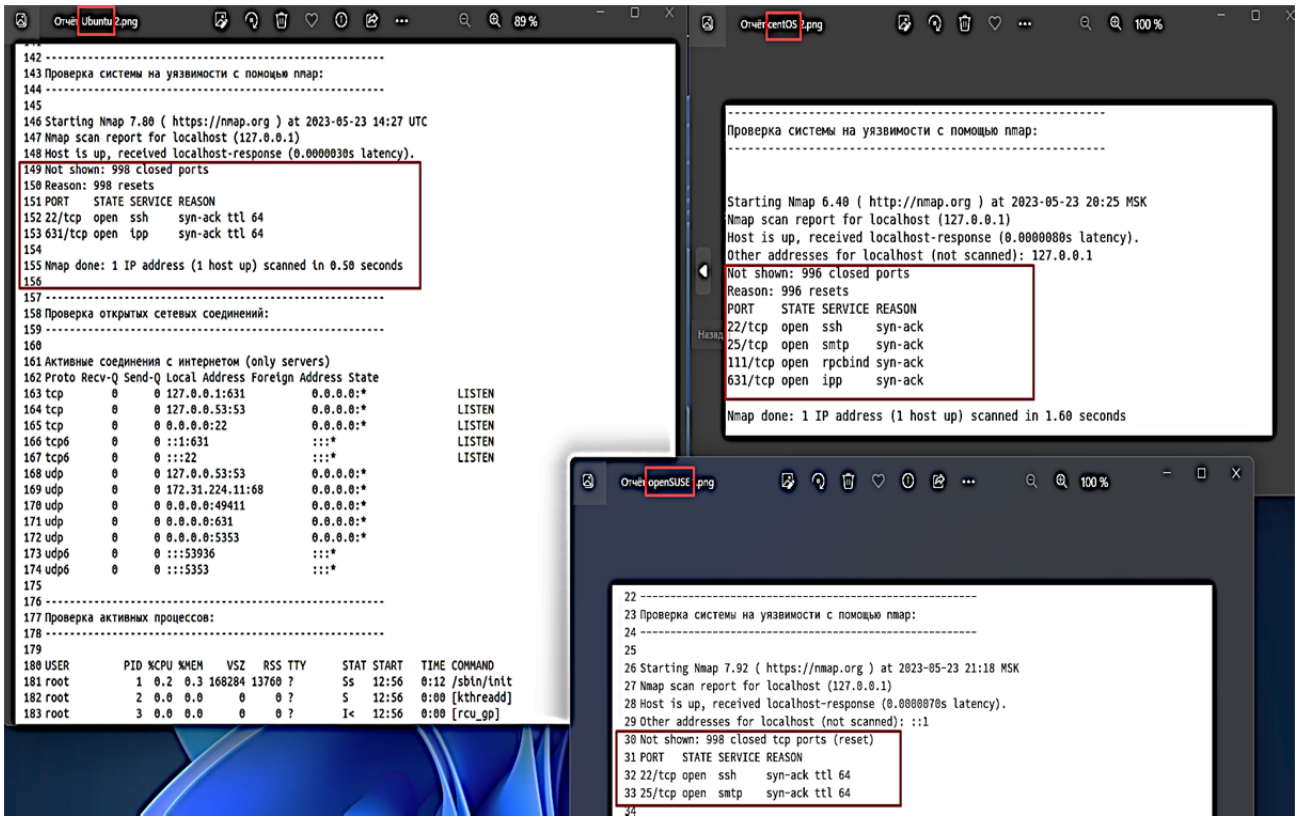

Рис. 1. **Отчёт NMAP**

Рис. 2 демонстрирует результаты теста на наличие незащищённых сетевых сервисов – худшие показатели у ***CentOS 7***, у ***Ubunru*** и ***openSUSE*** – сопоставимые.

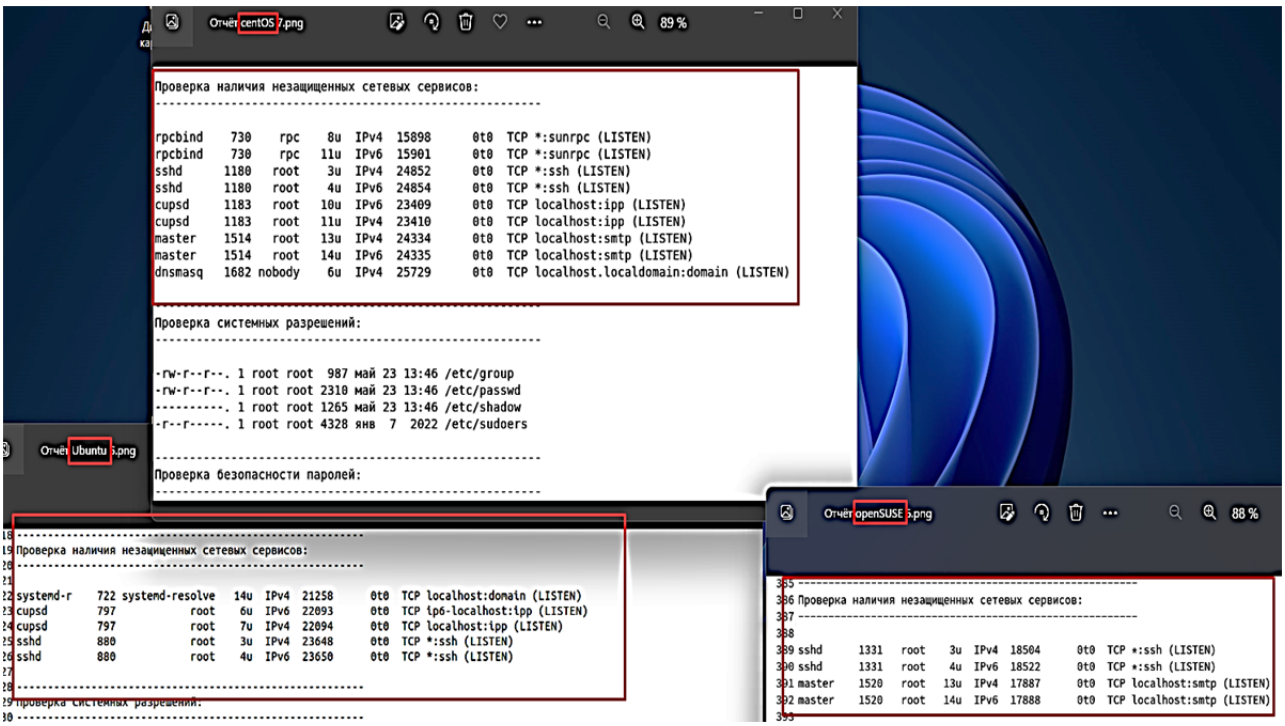

Рис. 2. **Проверка незащищённых сетевых сервисов**

На рис. 3 представлены результаты проверки открытых сетевых соединений – больше всего таких соединений у ***CentOS 7***.

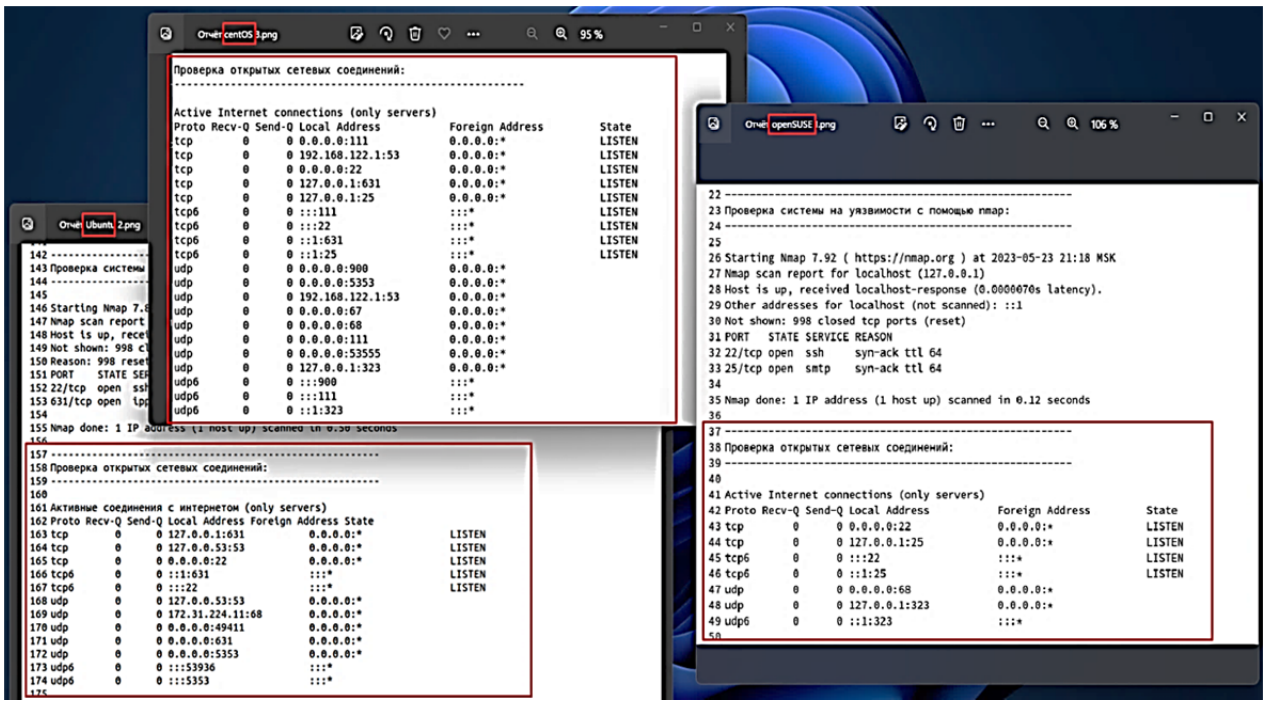

Рис. 3. **Тестирование открытых сетевых соединений**

По результатам сравнительного тестирования наихудшие показатели у ***CentOS 7***, поэтому его использование в наших задачах однозначно исключаем; ***Ubuntu*** и ***openSUSE*** показали схожие результаты. Выбор дистрибутива для данной разработки делаем в пользу ***Ubuntu*** за счёт более обширной базы пользователей и разработчиков. Также реальным его преимуществом является наиболее интуитивный интерфейс и элементы управления,

настройки: ***Ubuntu*** действительно проще в освоении и использовании для рядовых пользователей и системных администраторов.

Исправление уязвимостей и улучшение защиты

Дальнейшим шагом было проектирование Bash-скрипта (скрипт_2), который закроет обнаруженные предыдущим скриптом уязвимости и сделает эталонную настройку безопасности [6, 7].

Разработанный авторами продукт в автоматическом режиме выполняет следующие рутинные операции:

– обновляет систему и устанавливает все доступные обновления пакетов;
– устанавливает и настраивает межсетевой экран для ограничения входящего трафика;
– устанавливает инструменты безопасности для сканирования уязвимостей;
– сканирует систему на наличие уязвимостей;
– удаляет ненужные пакеты и очищает кэш;
– включает автоматическое обновление безопасности;
– устанавливает инструменты для мониторинга целостности файлов;
– устанавливает систему обнаружения вторжений;
– шифрует диск;
– устанавливает и настраивает систему защиты от DDoS-атак;
– настраивает системный журнал;
– устанавливает и настраивает систему защиты от вредоносного ПО.

Рис. 4 иллюстрирует часть процесса по устранению уязвимостей.

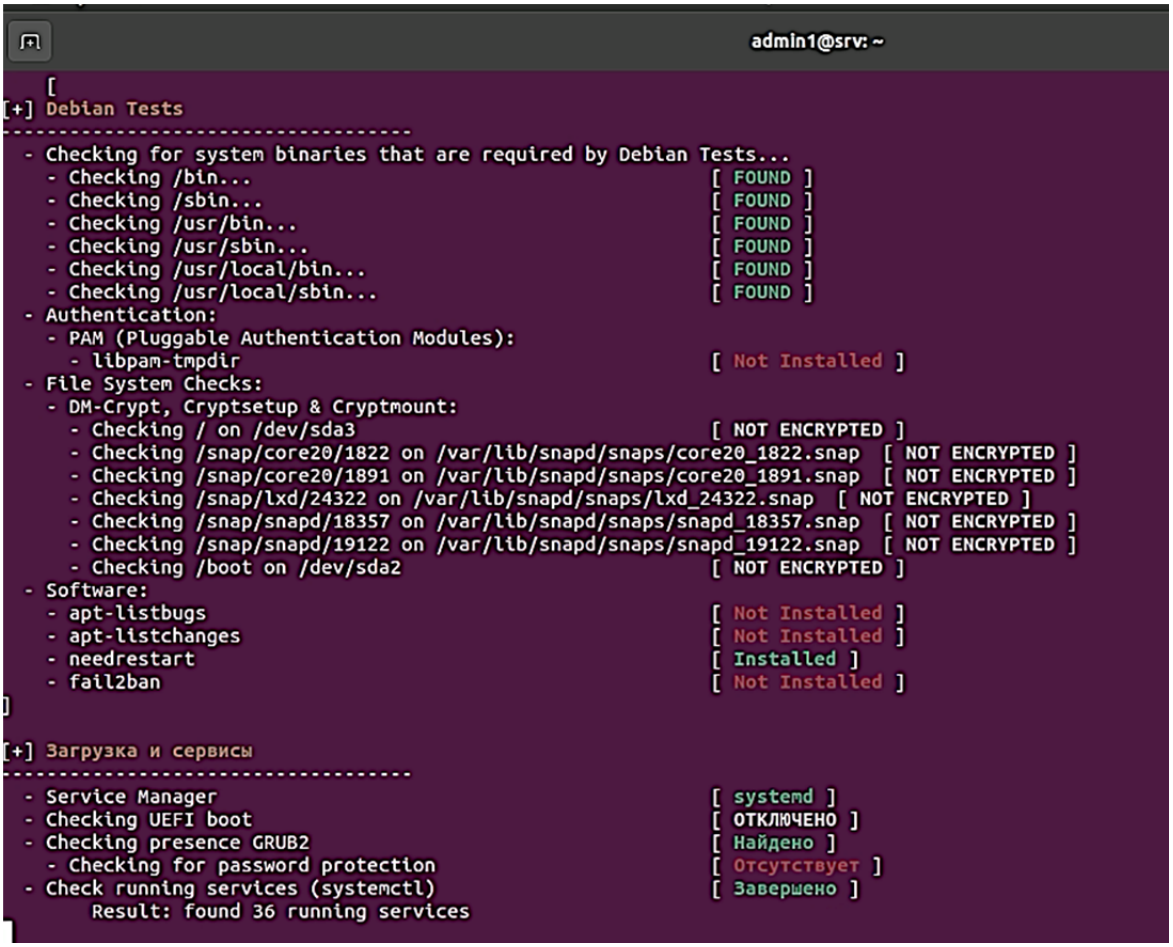

Рис. 4. **Устранение уязвимостей**

Разработка базовых ролей и настроек серверной части.

Автоматическая настройка рабочих станций

На третьем этапе решения задачи Bash-скрипт автоматизировал настройку кластеризации (с помощью ***pacemaker corosync***) серверной части главного хоста и наделил его необходимыми ролями и функциями [8].

С помощью скрипт_3, помимо резервного копирования (***rsync***), автоматизированы следующие конкретные операции установки той или иной роли, а именно: доменного контроллера (***kerberos***); мониторинга и управления всех хостов (***monitoring-tool***); VPN- (***openvpn***), DNS- (***bind9***), DHCP- (***isc-dhcp-server***) и файлового (***samba***) серверов.

На финальном (четвертом) этапе спроектирован и написан Bash-скрипт для клиентской части информационной системы [9].

Данный скрипт (скрипт_4) автоматизировал следующие процессы: настройка сети; введение хоста в домен; добавление систем мониторинга уязвимостей и целостности файлов, обновления безопасности, централизованного управления учетными записями, защиты от DDoS-атак, централизованного журналирования и обнаружения вторжений.

Оценка эффективности

Оценка эффективности использования разработанного Bash-скрипта для автоматизации миграции W2L проводилась опытным путём. Были протестированы и сопоставлены три варианта решения задачи миграции – сравнения дистрибутивов (подзадача 1), настройки серверной (подзадача 3) и клиентской частей (подзадача 4).

Вариант 1. Вручную с поиском команд для выполнения поставленных подзадач.

Вариант 2. Вручную с готовым списком команд для выполнения поставленных подзадач.

Вариант 3. Использование разработанного Bash-скрипта.

В табл. 2 представлено наглядное сравнение затраченного времени на вариативное решение подзадач миграции W2L.

Таблица 2

Затраченное время на решение подзадач миграции

Вариант	Сравнение дистрибутивов	Настройка кластера серверов	Настройка рабочих стаций
1	27 мин	3 ч	55 мин
2	15 мин	1,5 ч	25 мин
3	7 мин	40 мин	10 мин

Как видно из содержимого первого столбца таблицы, использование разработанного Bash-скрипта вдвое сокращает время решения подзадачи сравнения дистрибутивов относительно ручного тестирования с заготовленным списком команд и втрое, если искать каждую команду в момент решения.

Примерно такой же выглядит разница во времени в пользу автоматизированной настройки двух хостов кластера главного сервера со всеми минимально необходимыми ролями (второй столбец). Неопытному администратору, который будет искать каждую команду в интернете, разработанный Bash-скрипт поможет ускорить решение подзадачи, примерно в четыре с половиной раза. А опытному, с заготовленными или заученными командами – более чем в два раза.

Наиболее разительные отличия во времени и преимущества выполненного Bash-скрипта демонстрирует решение подзадачи настройки клиентского компьютера (третий столбец). В данном тестировании преимущество автоматизации почти в шесть раз относительно ручной настройки без подготовки, а ручной настройки с предварительной подготовкой – в два с половиной раза.

В табл. 2 приведено сравнение для «единичных» экземпляров. Но каковы будут отличия, если оценить эффективность автоматизированной настройки инфраструктуры предприятия (организации), состоящей из многосерверного кластера и десятков пользовательских компьютеров? Ответ содержится в табл. 3.

Таблица 3

Время настройки инфраструктуры небольшого/большого предприятия

Подзадача	Вариант 1	Вариант 2	Вариант 3
3	3 ч / 9 ч	1 ч 30 мин / 4 ч 30 мин	40 мин / 2 ч
4	9 ч 55 мин / 91 ч 40 мин	4 ч 10 мин / 41 ч 40 мин	1 ч 40 мин / 16 ч 40 мин
Всего	12 ч 55 мин / 100 ч 40 мин	5 ч 40 мин / 46 ч 10 мин	2 ч 20 мин / 18 ч 40 мин

Примечание: небольшое предприятие = кластер из 2 серверов + 10 рабочих станций; большое предприятие = кластер из 6 серверов + 100 рабочих станций

Изучив табл. 3, можно увидеть, что в реальных практических задачах значимость использования (а значит, и предшествующей ему разработки) Bash-скрипта действительно внушительна: с его помощью экономия времени на инфраструктурную миграцию W2L достигает порядка. И это при том, что отлаженный Bash-скрипт выполняет все действия безошибочно (что нельзя сказать о многочасовой монотонной, но ментально напряженной «ручной» работе системного администратора). Еще более наглядно эту разницу можно прочувствовать с помощью гистограммы на рис. 5, где по оси абсцисс отложены варианты (1–3), а по оси ординат – время (в часах) миграции (позадачно и суммарно).

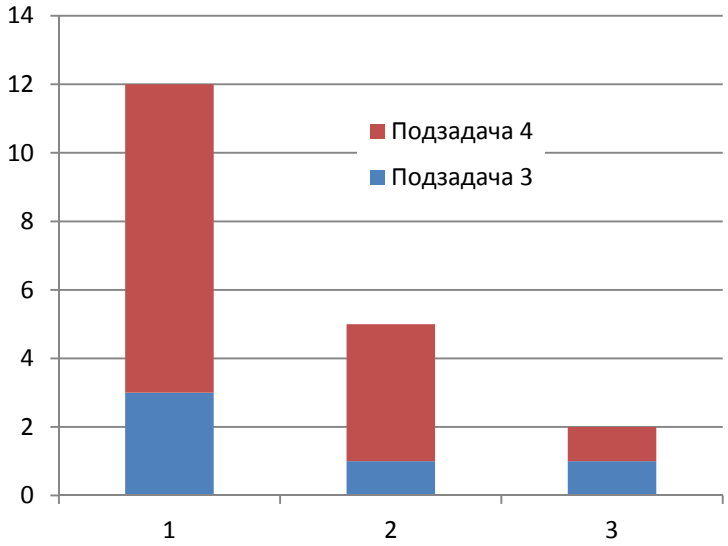

Рис. 5. **График затрат времени на миграцию инфраструктуры большого предприятия по трем вариантам**

Заключение

Авторами разработан инструмент, который помогает не только решить задачу выбора конкретного дистрибутива Linux – практически единственного возможного выхода из сложившейся «санкционной» ситуации, – но и значительно ускорить последующее развёртывание информационной системы на платформе свободно распространяемого ПО. Представленная разработка способствует максимальной автоматизации настройки серверной и клиентской части информационной системы организации или предприятия, что является наиболее ресурсоёмким этапом развёртывания. Согласно проведённой экспериментальной проверке такая автоматизация (для предприятий разного масштаба) позволяет экономить временной ресурс от 2,5 до 6 раз. Этот инструмент – Bash-скрипт – кроме экономии внушительного количества человеческого жизненного ресурса, также акцентирован на вопросах безопасности.

В ходе тестирования были выявлены некоторые ошибки и упущения, которые удалось успешно исправить с помощью штатных настроек. После успешного исправления данная разработка сработала на «отлично», в автоматическом режиме произвела проверку уязвимостей в выбранных дистрибутивах и сформировала отчёт с результатами для дальнейшего изучения. Следует сделать важное уточнение: данное решение разработано для дистрибутива Ubuntu, так как, проведя аналитику, было определено, что он подходит лучше всего. Опираясь на произведённое аналитическое сравнение, авторы рекомендуют строить информационную сеть также на Ubuntu, если нет серьёзных причин использовать другой дистрибутив. При использовании иного дистрибутива могут потребоваться правки и доработки данного скрипта под используемую систему.

Разработка уже является коммерчески успешной и востребованной: бизнес всегда готов платить за хороший инструмент, который ускоряет выполнение актуальных задач без потери качества.

Список источников

1. О мерах по обеспечению технологической независимости и безопасности критической информационной инфраструктуры Российской Федерации: Указ Президента Рос. Федерации от 30 марта 2022 г. № 166. Доступ из справ.-правового портала «Гарант».

2. Wale Soyinka. Linux Administration:A Beginner’s Guide // Fifth Edition. McGraw-Hill Education. 2008. URL: https://translated.turbopages.org/proxy_u/en-ru.ru.b67791fd-648c2235-0f8c2f16-74722d776562/https/www.overdrive.com/media/199510/linux-administration/ (дата обращения: 13.03.2023).

3. Справочное руководство по Bash. URL: https://www.gnu.org/software/bash/manual/bash.html (дата обращения: 27.03.2023).

4. Искусство программирования на языке сценариев командной оболочки. URL: https://www.opennet.ru/docs/RUS/bash_scripting_guide/ (дата обращения: 13.03.2023).

5. Фленов М.Е. Linux глазами хакера: учеб. пособие. 5-е изд., испр. и доп. СПб.: БХВ-Петербург, 2019. 418 с.

6. Пол Тронкон, Карл Олбинг. Bash и кибербезопасность: атака, защита и анализ из командной строки Linux. СПб.: Питер, 2020. 288 с.

7. Richard Blum, ChristineBresnahan. Linux® Command Line and Shell Scripting BIBLE. 4th Edition. Wiley. Indianapolis, Indiana, 2021. 982 p. URL: https://yandex.ru/search/?text=Ричард+Блум+Linux+Command+Line+and+Shell+Scripting+Bible&lr=2 (дата обращения: 01.05.2023).

8. Учебные пособия по Ubuntu Server. URL: https://ubuntu.com/server/docs/tutorials (дата обращения: 01.05.2023).

9. Руководство по написанию скриптов в Linux Bash. URL: https://selectel.ru/blog/tutorials/linux-bash-scripting-guide/?ysclid=li2by5sktw73474995 (дата обращения: 13.04.2023).

References

1. O merah po obespecheniyu tekhnologicheskoj nezavisimosti i bezopasnosti kriticheskoj informacionnoj infrastruktury Rossijskoj Federacii: Ukaz Prezidenta Ros. Federacii ot 30 marta 2022 g. № 166. Dostup iz sprav.-pravovogo portala «Garant».

2. Wale Soyinka. Linux Administration:A Beginner's Guide // Fifth Edition. McGraw-Hill Education. 2008. URL: https://translated.turbopages.org/proxy_u/en-ru.ru.b67791fd-648c2235-0f8c2f16-74722d776562/https/www.overdrive.com/media/199510/linux-administration/ (data obrashcheniya: 13.03.2023).

3. Spravochnoe rukovodstvo po Bash. URL: https://www.gnu.org/software/bash/manual/bash.html (data obrashcheniya: 27.03.2023).

4. Iskusstvo programmirovaniya na yazyke scenariev komandnoj obolochki. URL: https://www.opennet.ru/docs/RUS/bash_scripting_guide/ (data obrashcheniya: 13.03.2023).

5. Flenov M.E. Linux glazami hakera: ucheb. posobie. 5-e izd., ispr. i dop. SPb.: BHV-Peterburg, 2019. 418 s.

6. Pol Tronkon, Karl Olbing. Bash i kiberbezopasnost': ataka, zashchita i analiz iz komandnoj stroki Linux. SPb.: Piter, 2020. 288 s.

7. Richard Blum, ChristineBresnahan. Linux® Command Line and Shell Scripting BIBLE. 4th Edition. Wiley. Indianapolis, Indiana, 2021. 982 p. URL: https://yandex.ru/search/?text=Richard+Blum+Linux+Command+Line+and+Shell+Scripting+Bible&lr=2 (data obrashcheniya: 01.05.2023).

8. Uchebnye posobiya po Ubuntu Server. URL: https://ubuntu.com/server/docs/tutorials (data obrashcheniya: 01.05.2023).

9. Rukovodstvo po napisaniyu skriptov v Linux Bash. URL: https://selectel.ru/blog/tutorials/linux-bash-scripting-guide/?ysclid=li2by5sktw73474995 (data obrashcheniya: 13.04.2023).

Информация о статье:
Статья поступила в редакцию: 06.06.2023; одобрена после рецензирования: 16.06.2023; принята к публикации: 20.06.2023

Information about the article:
The article was submitted to the editorial office: 06.06.2023; approved after review: 16.06.2023; accepted for publication: 20.06.2023

Сведения об авторах:

Буйневич Михаил Викторович, профессор кафедры прикладной математики и информационных технологий Санкт-Петербургского университета ГПС МЧС России (196105, Санкт-Петербург, Московский пр., д. 149), доктор технических наук, профессор, e-mail: bmv1958@yandex.ru, https://orcid.org/0000-0001-8146-0022

Молчанов Дмитрий Александрович, специалист технической поддержки и системный администратор ООО «АИС» (Автоматические Интеллектуальные Системы) (192019, Санкт-Петербург, пр. Обуховской обороны, д. 45 лит О), e-mail: dmolchanov@ai-sys.ru

Information about authors:

Buinevich Mikhail V., professor of the department of applied mathematics and information technologies, Saint-Petersburg university of State fire service of EMERCOM of Russia (196105, Saint-Petersburg, Moskovsky ave., 149), doctor of technical sciences, professor, e-mail: bmv1958@yandex.ru, https://orcid.org/0000-0001-8146-0022

Molchanov Dmitry A., technical support specialist and system administrator AIS LLC (Automatic Intelligent Systems) (192019, Saint-Petersburg, Obukhovskaya Oborona pr., 45 lit O), e-mail: dmolchanov@ai-sys.ru