УДК 004

# Практические методы защиты доступа к серверу

*Болтовский Гавриил Александрович*
*Приамурский государственный университет им. Шолом-Алейхема*
*Студент*

**Аннотация**
Целью данной статьи является обзор существующих способов защиты доступа к серверу в глобальной сети. В рамках исследования приобретён виртуальный сервер, доступ к которому обеспечивался разными способами. Результатом исследования является защищённый сервер, с подробным описанием принципов его настройки.
**Ключевые слова:** информационная безопасность, защита VPS, администрирование

# Practical methods for securing access to the server

*Boltovsky Gavriil Aleksandrovich*
*Sholom-Aleichem Priamursky State University*
*Student*

**Abstract**
The purpose of this article is to review the existing methods of protecting access to a server in a global network. As part of the research, a virtual server was purchased and access to it was provided by different methods. The result of the research will be a secure server with a detailed description of the principles of its configuration.
**Keywords:** information security, VPS protection, administration

## 1 Введение

### 1.1 Актуальность исследования

В современных условиях развития информационных технологий обеспечение безопасности серверной инфраструктуры является критически важной задачей. Статистика показывает, что количество попыток несанкционированного доступа к серверам постоянно растет, что делает вопрос защиты доступа особенно актуальным. Традиционные методы аутентификации, основанные только на паролях, уже не обеспечивают достаточный уровень защиты. В данной работе рассматриваются практические методы усиления безопасности доступа к серверу на примере реальной системы под управлением Ubuntu Server.

Актуальность данного исследования обусловлена несколькими ключевыми факторами. Во-первых, SSH-доступ является одним из основных векторов атак на серверные системы, так как этот протокол широко используется для удаленного администрирования. Во-вторых, использование

виртуальных частных серверов (VPS) становится все более распространенным, что увеличивает количество потенциальных целей для злоумышленников. В-третьих, современные методы автоматизированного сканирования портов и подбора паролей требуют внедрения дополнительных механизмов защиты помимо стандартной парольной аутентификации.

### 1.2 Обзор исследований

В исследовании И. П. Батаевой [1] приводятся общие положения информационной безопасности. Автором перечислены задачи, которые стоят перед информационной безопасностью. Указывается, какие риски несёт отсутствие защиты. Н. Ш. Козлова, В. А. Довгаль [2] рассматривают различие между понятиями «информационная безопасность» и «кибербезопасность». Информационная безопасность охватывает защиту информации во всех ее формах, включая цифровую, бумажную и устную, с акцентом на конфиденциальность, целостность и доступность. Кибербезопасность является ее частью, фокусируясь на защите цифровых данных и систем от угроз в киберпространстве, таких как хакерские атаки и вирусы. Педагогический компонент в преподавании дисциплины «Информационная безопасность» исследуется В. А. Сизовым и других [3]. Автор подчеркивает важность создания мотивирующих и значимых проблемных ситуаций, а также диалогического взаимодействия между преподавателем и студентами. На примере задачи защиты информации в государственных информационных системах с архитектурой «тонкого клиента» демонстрируется, как системный подход помогает студентам выявлять угрозы, не учтенные в нормативно-правовой базе, и разрабатывать предложения по совершенствованию механизмов информационной безопасности.

Практические методы защиты компьютерных сетей описываются Ю.И.Стародубцевым [4]. Им предложен метод снижения нагрузки на элементы сети путем динамического изменения «Черного» списка IP-адресов и корректировки правил фильтрации на взаимосвязанных межсетевых экранах, распределенных по разным участкам сети. Это позволяет уменьшить вероятность косвенного подавления сетевых элементов и повысить эффективность защиты в условиях, возросших киберугроз.

### 1.3 Цель исследования

Целью исследования является разработка и практическая реализация комплекса мер по защите удаленного доступа к серверу Ubuntu Server путем внедрения методов SSH-аутентификации по ключам и Port Knocking с последующей оценкой их эффективности.

## 2 Материалы и методы

В первую очередь требуется проанализировать текущие угрозы безопасности VPS серверов и обосновать выбор методов защиты. Далее следует осуществить развертывание тестовой среды на базе Ubuntu Server для проведения практических экспериментов. На подготовленном сервере нужно

реализовать и настроить механизм аутентификации по SSH-ключам, а также внедрить систему Port Knocking. После выполнения технической части требуется провести тестирование внедренных механизмов защиты и оценить их эффективность. Заключительной задачей является формирование практических рекомендаций по применению исследованных методов защиты для администраторов серверных систем.

**3 Результаты и обсуждения**

В качестве экспериментальной платформы был приобретен виртуальный выделенный сервер у хостинг-провайдера RuVDS [5], работающий под управлением операционной системы Ubuntu Server 22.04 LTS. Начальная конфигурация сервера предусматривает стандартный метод удаленного доступа через протокол SSH на порту 22, где аутентификация осуществляется посредством ввода логина и пароля учетной записи root. Данный метод доступа, несмотря на его широкое распространение, представляет собой потенциальную уязвимость, поскольку делает систему доступной для автоматизированных атак методом перебора паролей, а также не обеспечивает должного уровня защиты при компрометации учетных данных. Именно эта базовая конфигурация послужит отпоплравной точкой для внедрения более защищенных методов доступа к серверу.

В качестве клиентской системы для подключения к серверу используется операционная система Windows 11, где доступ осуществляется через встроенный терминал PowerShell, который представляет собой мощную платформу для автоматизации задач и управления системой. Этот инструмент поддерживает работу с протоколом SSH, что обеспечивает удобный доступ к серверу для настройки и управления. Данный выбор обусловлен тем, что PowerShell является стандартным инструментом Windows и не требует установки дополнительного программного обеспечения. Подключение к серверу выполняется с помощью команды ssh в следующем формате:

Для подключения к серверу на нестандартном порту в PowerShell с использованием команды SSH синтаксис немного дополняется. Необходимо указать порт с помощью параметра -p. Пример команды: *ssh username@server_address -p port_number*, где *username* – имя пользователя, под которым осуществляется подключение; *server_address* – IP-адрес или доменное имя сервера; *-p port_number* – параметр, указывающий порт, отличный от стандартного (22).

После выполнения команды система запрашивает пароль пользователя root, и при успешной аутентификации предоставляет доступ к серверу. Стоит отметить, что Windows 11 имеет встроенный клиент SSH, что значительно упрощает процесс подключения по сравнению с предыдущими версиями операционной системы, где требовалась установка сторонних SSH-клиентов. Синтаксис команды SSH в PowerShell практически идентичен синтаксису в Linux.

Результат работы команды и приведён на рисунке (рис. 1).

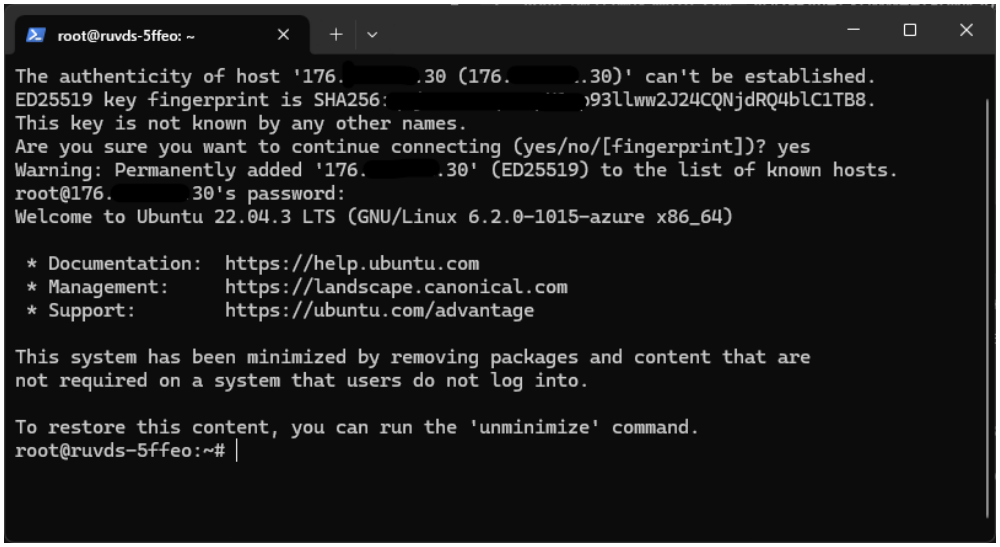


Рисунок 1 – Вход под Root по данным хостинг-провайдера

Для реализации доступа по ключам требуется сгенерировать ключевую пару. Сделать это можно в PowerShell, используя команду *ssh-keygen* (рис. 2).

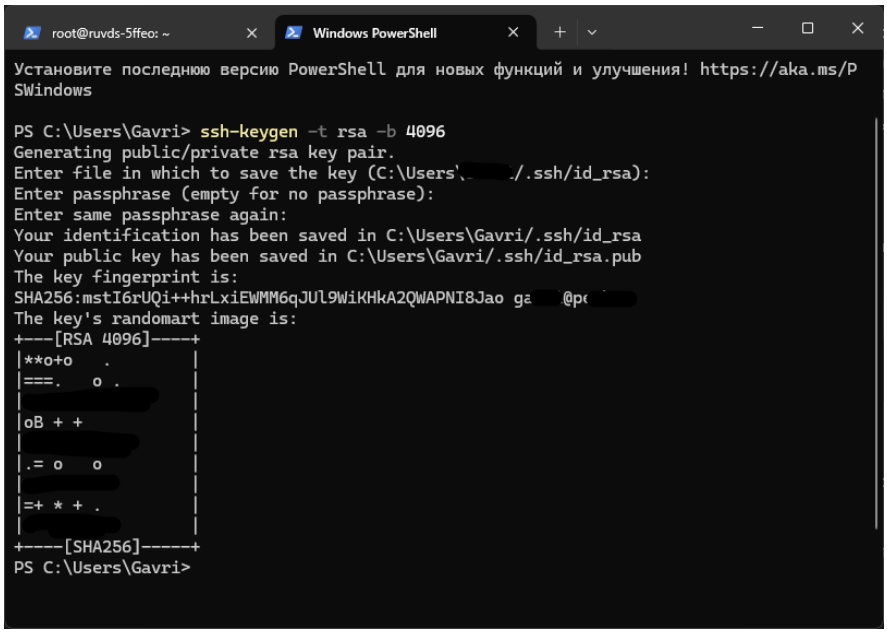


Рисунок 1 – Вход под Root по данным хостинг-провайдера

Данная команда, имеет следующий синтаксис: *-t rsa* указывает тип ключа (RSA); *-b 4096* определяет длину ключа в битах (рекомендуется использовать 4096 для повышенной безопасности). Полученная пара ключей сохраняется на компьютере, по умолчанию это *C:\Users\<Имя_пользователя>\.ssh\id_rsa*. По ходу выполнения команды можно указать пароль для приватного ключа.

Затем публичный ключ нужно сохранить на сервер. Один из способов – открыть его в текстовом редакторе, сохранить в буфер обмена, создать файл на сервере в директории и сохранить его. На стороне сервера этот ключ нужно добавить в файл *~/.ssh/authorized_keys* для пользователя, под которым вы

планируете подключаться. Права на файл *authorized_keys* на сервере должны быть ограничены, сделать это можно командой *chmod 600 ~/.ssh/authorized_keys*. *PubkeyAuthentication* в файле конфигурации SSH сервера *(/etc/ssh/sshd_config*) должен быть установлен в *yes*. После изменений необходимо перезапустить службу SSH (*sudo systemctl restart ssh*).

После успешного копирования публичного ключа сервер сможет аутентифицировать пользователя по приватному ключу (рис. 3).


Рисунок 3 – Вход под Root через ключ

Аутентификация по ключу обеспечивает высокий уровень безопасности, исключая необходимость передачи пароля через сеть. Однако этот метод не лишён определённых проблем, которые могут возникнуть как на этапе настройки, так и в процессе эксплуатации.

Изменение стандартного порта SSH (22) на нестандартный является эффективной мерой для уменьшения числа автоматизированных атак, таких как сканирование портов. Злоумышленники часто используют стандартные порты для поиска потенциальных целей. Например, смена порта на 2222 или любой другой, выбранный случайным образом, позволяет значительно снизить вероятность обнаружения сервера.

Однако изменение порта не является абсолютной защитой. Опытные злоумышленники могут обнаружить SSH-сервер путём тщательного сканирования всех открытых портов. Чтобы противодействовать таким угрозам, можно использовать более сложные методы защиты, такие как **Port Knocking.**

Port Knocking – это метод, позволяющий открыть доступ к серверу только после выполнения определённой последовательности действий на заранее заданных портах. По умолчанию все порты остаются закрытыми для внешнего мира, включая SSH. Когда клиент выполняет серию попыток подключения к портам в определённой последовательности (например, 1234 - 5678 - 9012), сервер открывает порт SSH для данного IP-адреса на ограниченное время.

Этот метод значительно усложняет задачу злоумышленнику, так как даже если он знает, что сервер использует SSH, он не сможет определить последовательность "стука", необходимую для открытия порта.

Port Knocking работает как дополнительный уровень безопасности, дополняя использование доступа по ключу.

Реализация Port Knocking на сервере под управлением Ubuntu включает несколько этапов, таких как установка необходимого ПО, настройка правил брандмауэра и тестирование. Для примера мы используем утилиту knockd, которая популярна благодаря своей простоте и гибкости.

Установка происходит через следующие команды: *sudo apt update && sudo apt upgrade -y* и *sudo apt install knockd -y*

Для реализации Port Knocking на сервере с использованием Ubuntu необходимо выполнить настройку, которая включает в себя установку необходимых инструментов, конфигурирование правил доступа к портам и проверку работы механизма. Настройка происходит редактированием соответствующего файла: *sudo nano /etc/knockd.conf*

С клиента выполнить стук можно с помощью утилиты knock. На Windows можно использовать утилиту knock.exe. Пример команды для Linux: *knock <server_ip> 1234 5678 9012*. Эта команда открывает доступ к серверу по SSH на настроенном порту. Аналогично, обратной последовательностью порт закрывается (*knock <server_ip> 9012 5678 1234*).

Эта технология добавляет дополнительный уровень защиты, делая сервер недоступным для сканирования портов до выполнения правильной последовательности действий.

Для корректной работы важно убедиться, что последовательность "стука" является уникальной и не пересекается с портами, используемыми другими приложениями. В конфигурации задаются параметры временного интервала, в течение которого сервер ожидает выполнения последовательности, а также IP-адрес клиента, для которого будет открыт порт. После внесения изменений в конфигурационный файл демон необходимо перезапустить для их применения.

Проверка работоспособности Port Knocking осуществляется с клиентской стороны. Клиент выполняет последовательность попыток подключения к указанным портам с использованием утилиты knock. Если сервер распознаёт последовательность как корректную, порт SSH временно открывается, что позволяет пользователю установить защищённое соединение.

Такой подход значительно повышает безопасность сервера, поскольку делает невозможным определение порта SSH без знания правильной последовательности.

## 4. Выводы

Таким образом, сочетание использования аутентификации по ключу, изменения порта SSH и внедрения механизма Port Knocking позволяет значительно повысить безопасность доступа к серверу, минимизируя риски несанкционированного проникновения.

## Библиографический список

1. Батаева И. П. Защита информации и информационная безопасность // НиКа. 2012. №.1 С. 143-153.
2. Козлова Н. Ш., Довгаль В. А. Кибербезопасность и информационная безопасность: сходства и отличия // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2021. №3 (286). С. 88-97.
3. Сизов В. А., Малиничев Д. М., Кучмезов Х. Х., Мочалов В. В. Применение

метода проблемного обучения в изучении дисциплины «Информационная безопасность» // Открытое образование. 2021. №3. С. 133-139.
4. Стародубцев Ю. И. и др. Способ защиты серверов услуг сети связи от компьютерных атак // Вопросы оборонной техники. Серия 16: Технические средства противодействия терроризму. 2020. №. 9-10. С. 63-67.
5. RUVDS. URL: https://www.ruvds.com/ (дата обращения: 30.12.2024).