Лабораторная работа №16

Базовая защита от атак типа «brute force»

Газизянов Владислав Альбертович

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Получение практических навыков работы с программным средством **Fail2ban** для обеспечения базовой защиты сервера от атак типа «brute force».

2 Задание

1. Установить и настроить **Fail2ban** для отслеживания работы сетевых служб сервера.
2. Проверить работу **Fail2ban** через попытки несанкционированного доступа по протоколу SSH.
3. Создать скрипт для автоматической установки и настройки **Fail2ban** в инфраструктуре **Vagrant**.

3 Выполнение лабораторной работы

3.1 16.4.1. Установка и настройка Fail2ban

3.1.1 Установка и запуск Fail2ban

На сервере выполнена установка пакета **Fail2ban** с помощью менеджера пакетов **dnf**. После установки служба была запущена и добавлена в автозагрузку для обеспечения постоянной защиты.

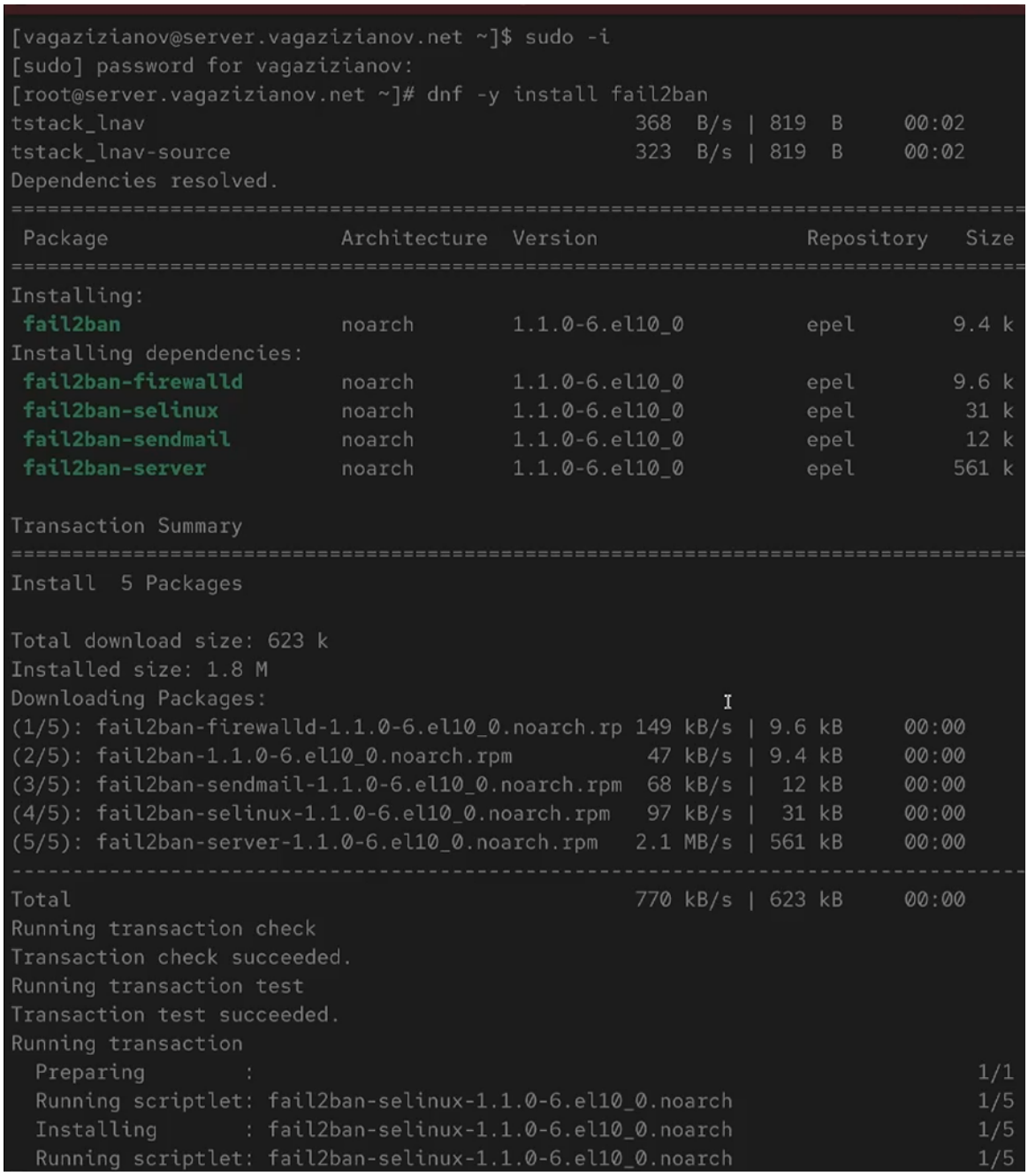

Рисунок 3.1: Установка и запуск службы Fail2ban

3.1.2 Мониторинг журнала событий

В отдельном терминале запущен мониторинг журнала событий **Fail2ban** для наблюдения за работой системы в реальном времени. Это позволяет отслеживать

все действия по блокировке и разблокировке IP-адресов.

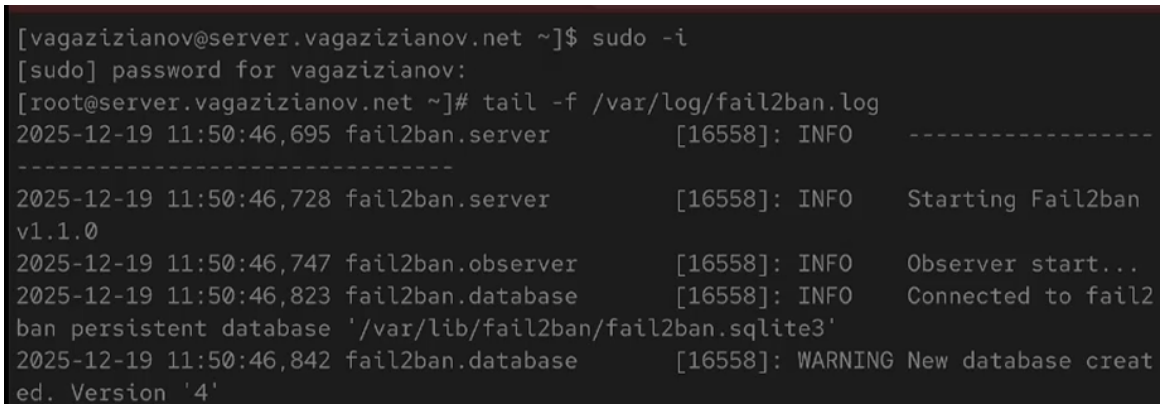

Рисунок 3.2: Запуск мониторинга журнала Fail2ban

3.1.3 Создание локальной конфигурации

Создан файл локальной конфигурации `customisation.local` в каталоге `/etc/fail2ban/jail.d/`. В файле заданы глобальные настройки, включая время блокировки злоумышленников на 1 час (3600 секунд).

Рисунок 3.3: Создание файла локальной конфигурации Fail2ban

3.1.4 Настройка защиты SSH

В конфигурационный файл добавлены настройки для защиты службы **SSH**. Настроена защита для стандартного порта 22 и альтернативного порта 2022, активированы различные фильтры для обнаружения атак.

Рисунок 3.4: Настройка защиты SSH в конфигурационном файле

3.1.5 Настройка защиты HTTP-служб

Добавлены настройки для защиты веб-сервера **Apache**. Активированы различные фильтры для защиты от атак на аутентификацию, вредоносных ботов, переполнений и других типов угроз.

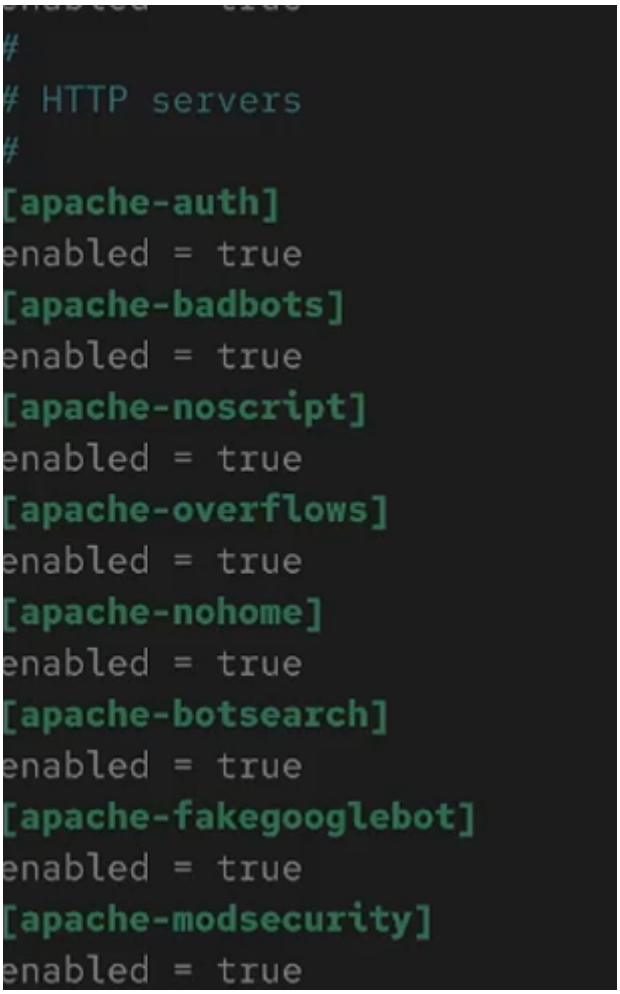

Рисунок 3.5: Настройка защиты HTTP-служб Apache

3.1.6 Настройка защиты почтовых служб

Включена защита для почтовых служб **Postfix** и **Dovecot**. Настроены фильтры для защиты от спама, атак на аутентификацию SASL и других угроз, связанных с почтовой системой.

Рисунок 3.6: Настройка защиты почтовых служб Postfix и Dovecot

3.1.7 Перезапуск и проверка службы

После каждой настройки выполнены перезапуск службы **Fail2ban** и проверка журнала событий для подтверждения корректного применения изменений.

Рисунок 3.7: Перезапуск службы Fail2ban и проверка журнала

3.2 16.4.2. Проверка работы Fail2ban

3.2.1 Проверка статуса службы

Проверен общий статус **Fail2ban** и статус защиты конкретной службы **SSH**. Убедились, что служба работает корректно и защита SSH активна.

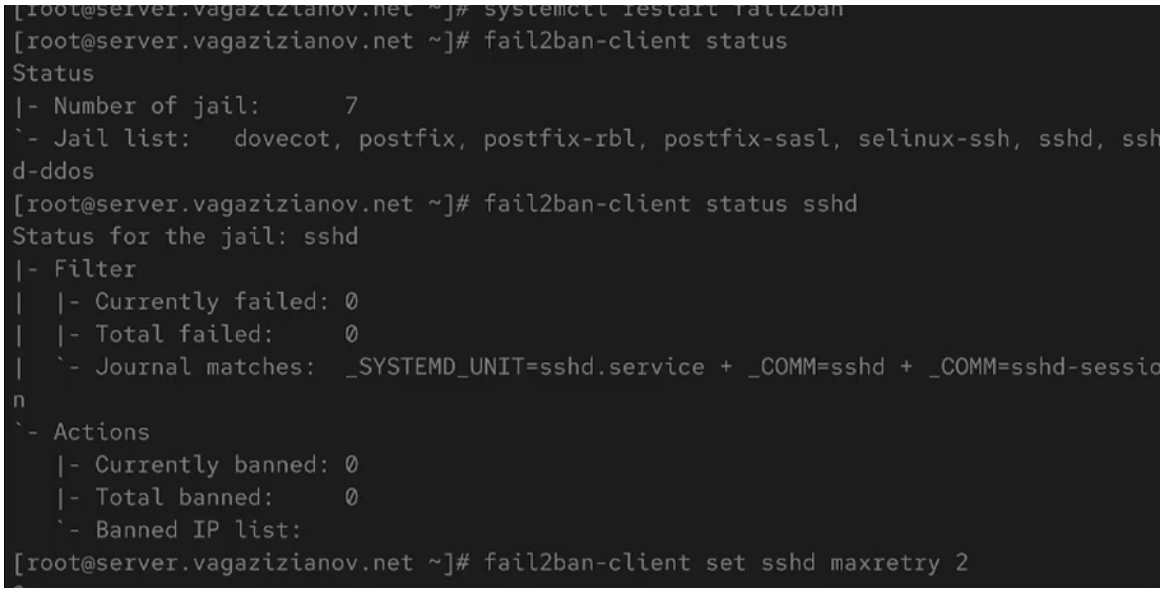

Рисунок 3.8: Проверка статуса Fail2ban и защиты SSH

3.2.2 Настройка параметров защиты SSH

Установлено максимальное количество неудачных попыток входа для SSH равное 2. Это позволяет быстрее блокировать злоумышленников при попытках подбора пароля.

Рисунок 3.9: Настройка параметра maxretry для SSH

3.2.3 Тестирование блокировки

С клиентской машины выполнены попытки входа по SSH с неверным паролем. После превышения лимита попыток IP-адрес клиента был автоматически заблокирован системой **Fail2ban**.

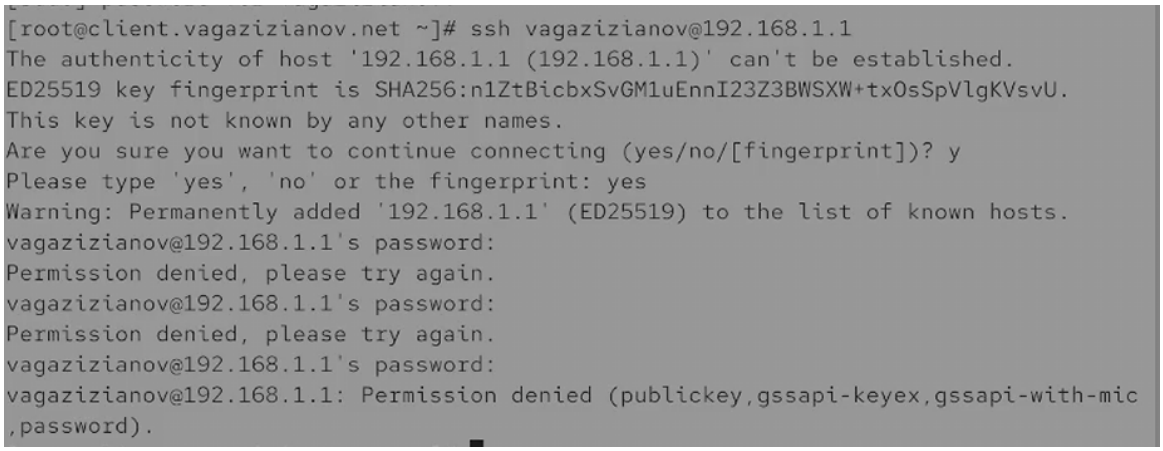

Рисунок 3.10: Тестирование блокировки IP-адреса при неудачных попытках SSH

3.2.4 Разблокировка IP-адреса

Выполнена ручная разблокировка IP-адреса клиента через клиент **Fail2ban**. Проверен статус защиты SSH для подтверждения снятия блокировки.

Рисунок 3.11: Ручная разблокировка IP-адреса клиента

3.2.5 Настройка исключений

В конфигурационный файл добавлены IP-адреса для игнорирования (белый список). После перезапуска службы и повторных попыток входа с неверным паролем подтверждено, что адрес из белого списка не блокируется.

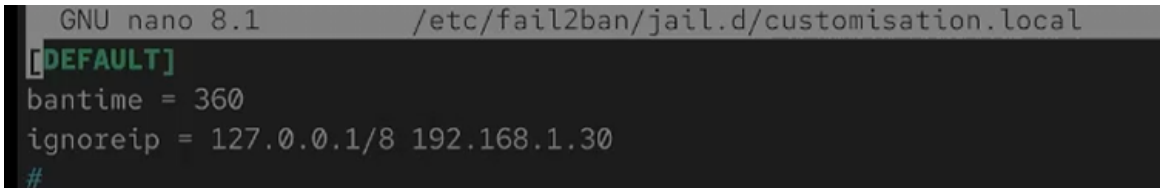

Рисунок 3.12: Добавление IP-адреса в белый список и проверка

3.3 16.4.3. Автоматизация настройки

3.3.1 Подготовка конфигурационных файлов

Создана структура каталогов в проекте **Vagrant** и скопирован конфигурационный файл `customisation.local` для последующего использования в скрипте автоматической настройки.

```
[root@server.vagazizianov.net ~]# cd /vagrant/provision/server
[root@server.vagazizianov.net server]# mkdir -p /vagrant/provision/server/protect/e
tc/fail2ban/jail.d
[root@server.vagazizianov.net server]# cp -R /etc/fail2ban/jail.d/customisation.loc
al /vagrant/provision/server/protect/etc/fail2ban/jail.d/
[root@server.vagazizianov.net server]# cd /vagrant/provision/server
[root@server.vagazizianov.net server]# touch protect.sh
[root@server.vagazizianov.net server]# chmod +x protect.sh
[root@server.vagazizianov.net server]# nano protect.sh
```

Рисунок 3.13: Подготовка конфигурационных файлов для проекта Vagrant

3.3.2 Создание provisioning-скрипта

Создан исполняемый bash-скрипт `protect.sh` для автоматической установки и настройки **Fail2ban**. Скрипт включает установку пакета, копирование конфигураций, настройку прав доступа и запуск службы.

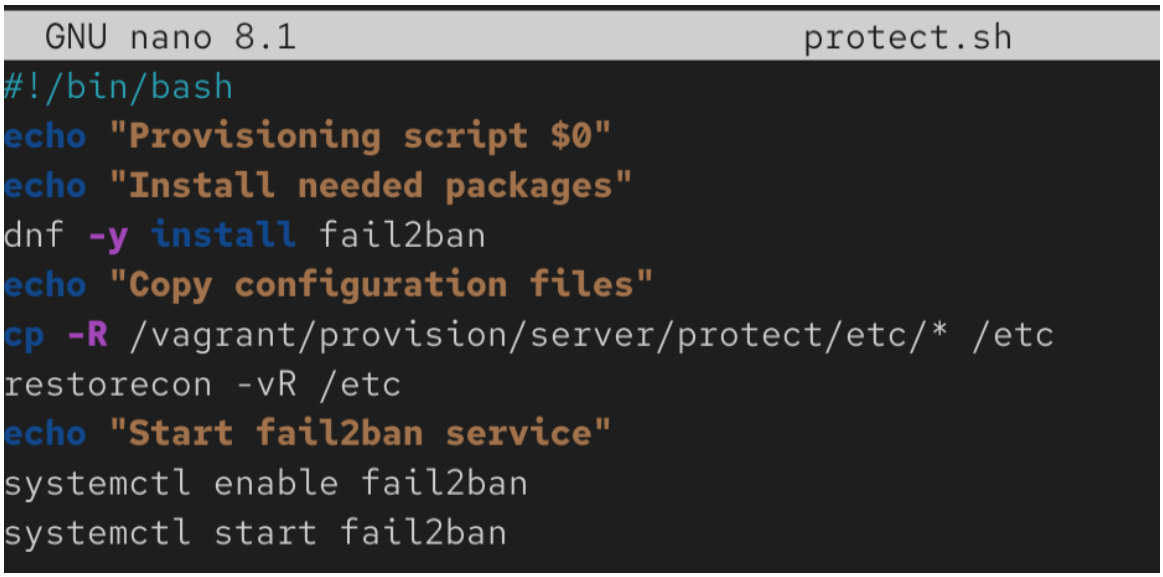

Рисунок 3.14: Создание скрипта автоматической настройки protect.sh

3.3.3 Интеграция с Vagrant

В конфигурационный файл **Vagrantfile** добавлен вызов созданного скрипта в качестве provisioning-шага для виртуальной машины сервера. Это обеспечивает автоматическую настройку защиты при развертывании среды.

```
server.vm.provision "server netlog",
                  type: "shell",
                  preserve_order: true,
                  path: "provision/server/netlog.sh"

server.vm.provision "server protect",
                  type: "shell",
                  preserve_order: true,
                  path: "provision/server/protect.sh"

server.vm.provision "server firewall",
                  type: "shell",
                  preserve_order: true,
                  path: "provision/server/firewall.sh"
```

Рисунок 3.15: Интеграция скрипта в конфигурацию Vagrantfile

4 Контрольные вопросы

1. **Поясните принцип работы Fail2ban.**

   Fail2ban отслеживает логи сетевых служб, анализирует их на наличие подозрительной активности (например, множественные неудачные попытки входа) и автоматически блокирует IP-адреса злоумышленников через добавление правил в межсетевой экран.

2. **Настройки какого файла более приоритетны: jail.conf или jail.local?**

   Настройки из файла `jail.local` имеют более высокий приоритет и переопределяют соответствующие параметры из `jail.conf`. Рекомендуется вносить изменения именно в `jail.local` или файлы в каталоге `jail.d/`.

3. **Как настроить оповещение администратора при срабатывании Fail2ban?**

   Оповещение настраивается через параметры `action` в конфигурации тюрьмы. Можно использовать встроенные действия типа `action_mw` (отправка письма с whois-информацией) или `action_mwl` (письмо с логами), предварительно настроив почтовую систему.

4. **Поясните построчно настройки по умолчанию в конфигурационном файле /etc/fail2ban/jail.conf, относящиеся к веб-службе.**

   В секции `[apache-auth]` задаются параметры для защиты аутентификации Apache: `port = http,https`, `filter = apache-auth`, `logpath` указывает путь к логам ошибок Apache, `maxretry` определяет количество

допустимых попыток.

5. **Поясните построчно настройки по умолчанию в конфигурационном файле /etc/fail2ban/jail.conf, относящиеся к почтовой службе.**
   В секции `[postfix]` настроена защита SMTP-сервера: `port = smtp,ssmtp`, `filter = postfix`, `logpath` указывает на логи Postfix, параметры `maxretry` и `findtime` определяют условия блокировки.

6. **Какие действия может выполнять Fail2ban при обнаружении атакующего IP-адреса? Где можно посмотреть описание действий для последующего использования в настройках Fail2ban?**
   Fail2ban может блокировать IP через iptables, firewalld, отправлять уведомления, выполнять произвольные команды. Описание действий находится в файлах в каталоге `/etc/fail2ban/action.d/`.

7. **Как получить список действующих правил Fail2ban?**
   Команда `fail2ban-client status` показывает список активных «тюрем» (jails). Для конкретной службы: `fail2ban-client status <XXX_XXXXXX>`.

8. **Как получить статистику заблокированных Fail2ban адресов?**
   Статистика отображается в журнале `/var/log/fail2ban.log`. Также можно использовать `fail2ban-client status <XXX_XXXXXX>` для просмотра заблокированных IP в конкретной «тюрьме».

9. **Как разблокировать IP-адрес?**
   Команда `fail2ban-client set <XXX_XXXXXX> unbanip <IP-XXXXX>` разблокирует указанный адрес. Для разблокировки всех адресов в «тюрьме» можно использовать `fail2ban-client set <XXX_XXXXXX> unban --all`.

5 Выводы

- Освоены практические навыки установки и настройки системы защиты **Fail2ban**.
- Настроена комплексная защита сетевых служб (SSH, HTTP, почтовых) от атак типа «brute force».
- Проверена работоспособность системы через тестирование блокировки и разблокировки IP-адресов.
- Реализована автоматизация процесса развертывания защиты через provisioning-скрипт в инфраструктуре **Vagrant**.
- Приобретены знания по мониторингу и управлению системой защиты от несанкционированного доступа.