Отчет по прохождению внешнего курса.

Раздел 2. Защита ПК/телефона.

Рогожина Надежда Александровна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Цель данного курса - узнать, как обеспечивается безопасность интернет-трафика, какие пароли нужно выбирать и как их хранить, познакомиться с методами защиты сообщений в мессенджерах (WhatsApp, Telegram), понять, как работают механизмы аунтефикации в электронных платежах, а также зачем нас иногда просят выбрать квадраты, где изображены светофоры.

2 Выполнение

2.1 Шифрование диска

1. Можно ли зашифровать загрузочный сектор диска (рис. [2.1]):

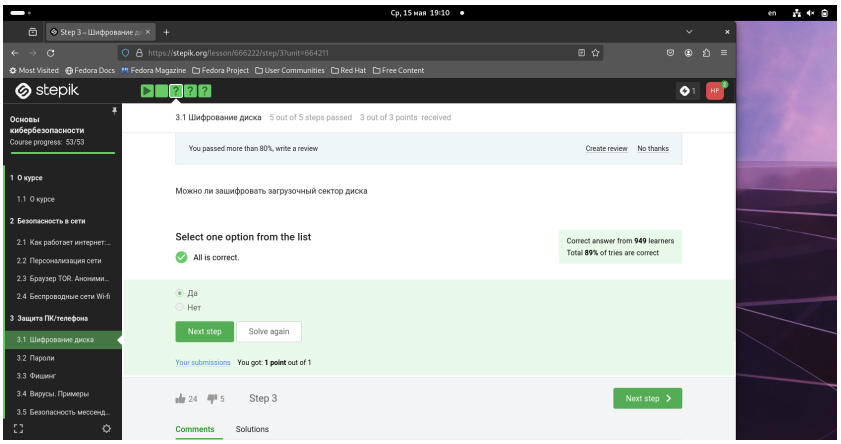

Рис. 2.1: Загрузочный сектор

Как было сказано в лекции, шифровать можно не только жесткий диск, где мы храним файлы, можно шифровать и загрузочный сектор диска. Это тот сектор, который включается сразу после того, как мы стартовали компьютер. Компьютер считывает данные в сегменте, который хранит у себя загрузочные файлы операционной системы, и мы получаем либо свой рабочий стол, либо запрос на авторизацию для доступа к нашему рабочему столу. Вот для того, чтобы зашифровать загрузочный сектор, мы тоже можем использовать методы шифрования. В этом случае вас попросят ввести ключ или скорее даже пароль к этому ключу при самом запуске компьютера.

2. Шифрование диска основано на (рис. [2.2]):

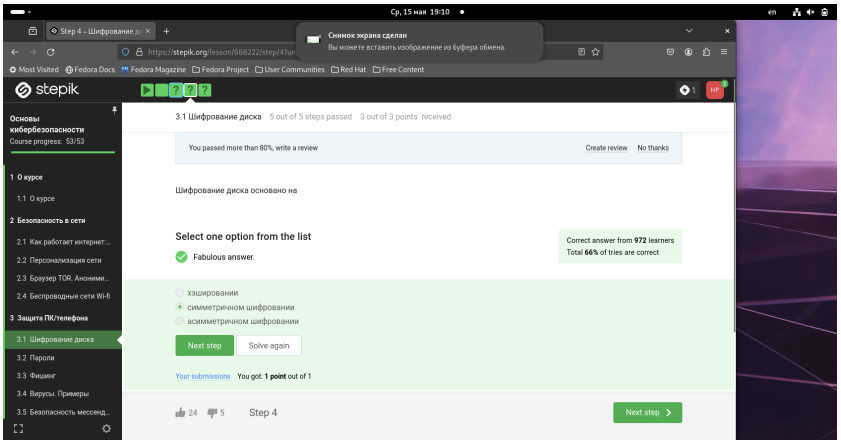

Рис. 2.2: Шифрование диска

Шифрование больших объемов данных, например, жесткого диска или сегмента жесткого диска или какой-то большой флешки, осуществляется с помощью симметричного шифрования, как правило, алгоритма AES. Это американский стандарт симметричного шифрования, он также используется для конфиденциальной передачи данных по сети. Это эффективный алгоритм, который реализован в процессорах быстро, то есть на аппаратном уровне. Благодаря тому, что это хороший алгоритм, пользователь практически не наблюдает задержек в работе, то есть данные шифруются-дешифруются быстро. Как правило, это происходит на заднем фоне, мы можем при этом работать на компьютере, будут происходить какие-то параллельные операции на шифрование и дешифрование.

3. С помощью каких программ можно зашифровать жесткий диск (рис. [2.3]):

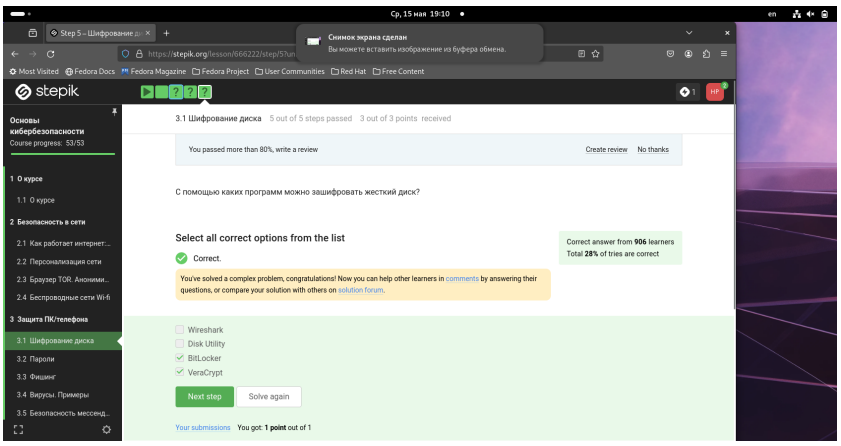

Рис. 2.3: Пример программ

Во всех популярных операционных системах есть встроенные утилиты, которые позволяют шифровать жесткий диск: для Windows это Bitlocker, в Linux – LUKS, в MacOS – это FileVault. Кроме того, есть и сторонние опенсорсные (open source) программы, то есть бесплатные: это Veracrypt, PGPDisk, которые вы можете установить себе и использовать их для шифрования ваших жестких дисков, загрузочных секторов или флешек.

2.2 Пароли

1. Какие пароли можно отнести с стойкими (рис. [2.4]):

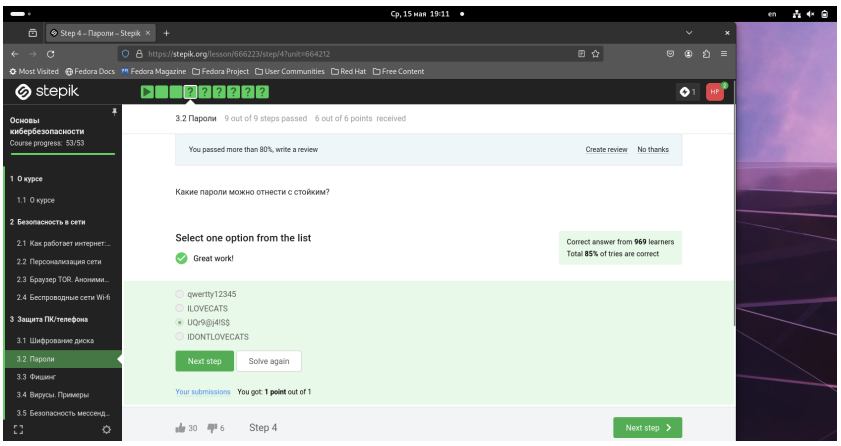

Рис. 2.4: Примеры надежных паролей

Основной критерий стойкости пароля - это сложность его перебора. Самая основная атака на наши пароли - это банальный перебор всевозможных паролей. Например, если мы возьмем пароли длины 6, которые состоят только из цифр от 0 до 9, то множество всех таких паролей будет содержать в себе 106, то есть один миллион элементов, и это множество перебирается на самом простом, недорогом ноутбуке за секунду в сегодняшних возможностях. Важно то, что, если мы будем увеличивать длину пароля, допустим, возьмем не 6, а 10, и опять рассмотрим только цифры, мы получим с вами 1010 всевозможных паролей, это 10 миллиардов паролей. Это уже больше, для этого нам потребуются более серьёзные мощности, нежели домашний лаптоп, однако это все равно вполне себе перебираемое множество, на которое мы просто потратим не секунду, а, например, часы или дни, в зависимости от оборудования. Важно то, что, если мы с вами перейдем от цифр к буквам, например, возьмём латинский алфавит, даже только нижний регистр, маленькие буквы, то мы значительно увеличим мощность всевозможных паролей длины 8. Если у нас есть 10 цифр и 26 букв алфавита, то мы получим 368 всевозможных паролей для перебора, это уже почти 3 триллиона паролей, и перебор вот такого множества уже становится практически не реализуем, для этого нужно действительно очень много машин соединить, очень большие мощности, и вот такой пароль считается уже относительно без-

опасным. Естественно, если мы с вами добавим в эти 8 символов еще и 8 слотов для символов и ещё спецсимволы, то мы уже увеличим с вами область перебора до числа, перебор которого на сегодняшний день невыполним. Таким образом, поэтому, когда мы создаем пароль, он всегда автоматически проверяется на стойкость. Проверяется его длина и из какого алфавита он состоит: из цифр, букв нижнего регистра, верхнего регистра и спецсимволов. Как правило, вас просят добавить как минимум один спецсимвол, как минимум одну цифру, как минимум одну букву верхнего регистра, нижнего регистра. Это все сделано для того, чтобы сложность перебора вашего пароля была большой.

2. Где безопасно хранить пароли (рис. [2.5]):

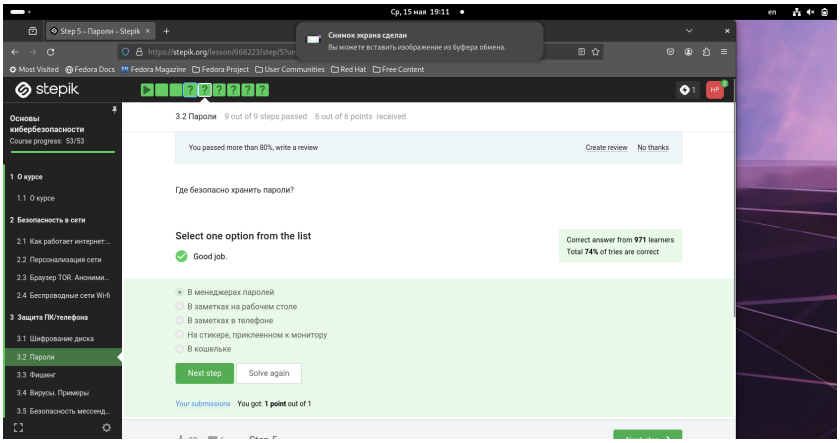

Рис. 2.5: Пароли

Перейдем к тому, как хранить пароли на компьютере. Мы уже договорились, что пароли должны быть длинные, длинные пароли сложно запомнить. Поэтому корректно хранить свои ключи в хранилище паролей. Хранилища у нас есть на разных операционных системах. Я привела в пример KeepassXC, по-моему, он кроссплатформенный, он есть и на Linux, на Windows; на Mac есть Keychain Access, это утилита, которая хранит пароли. И если у вас есть такое хранилище паролей, достаточно запомнить мастер-ключ, мастер-пароль от этого самого хранилища, и когда вы получите доступ к своему хранилищу, вы можете посмотреть все

свои длинные безопасные пароли к разным сервисам, запоминая только один хороший пароль.

3. Зачем нужна капча (рис. [2.6]):

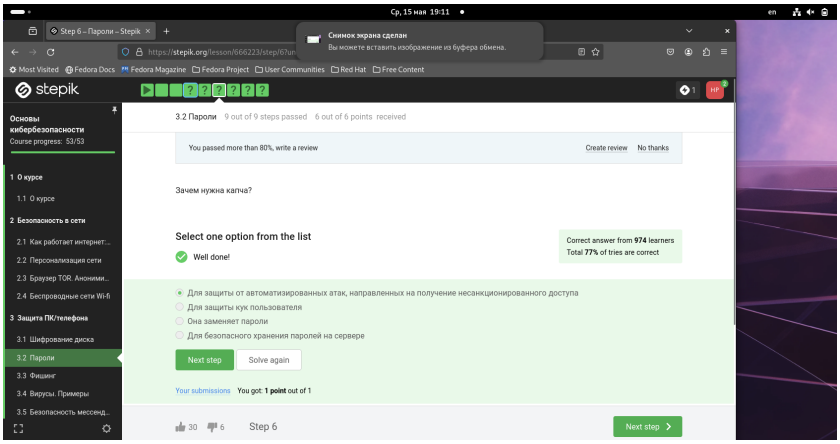

Рис. 2.6: Капча

Капча - это аббревиатура с английского; это тест для определения, является ли пользователь, который общается с веб-сервисом, человеком или компьютером, ботом, которой пытается просто-напросто перебрать все пароли. После того, как мы ввели имя пользователя и пароль, часто помимо этого нас еще какой-то веб-сайт спрашивает какой-то тест, в котором мы должны там увидеть какие-то плохо написанные буквы или символы, и цель этого - отличить нас от компьютера, который пытается автоматически перебрать пароли конкретного пользователя или даже в сумме пользователя и пароля, просто пусть какой-то доступ к ресурсу.

4. Для чего применяется хэширование паролей (рис. [2.7]):

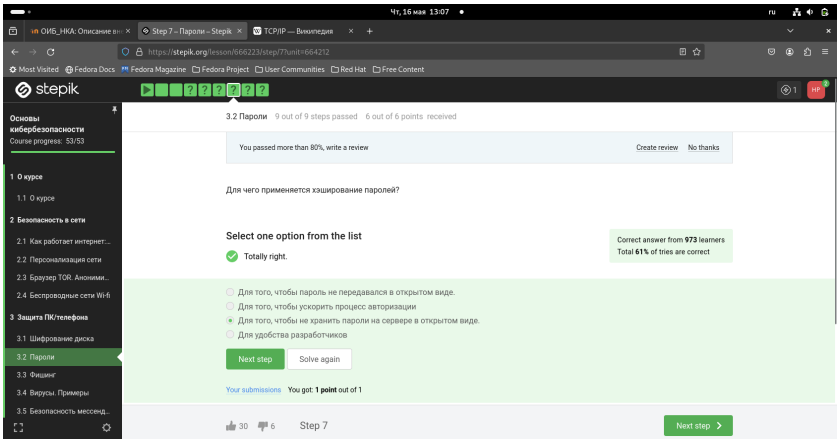

Рис. 2.7: Хэширование

Что такое хэш? Хэш произошло от английского to hash – перемалывать, работать мясорубкой, то есть это преобразование какого-то числа информации в какую-то другую информацию. В частности, хэш-функция - эта функция, которая берет на вход любую строку битовую и преобразует ее в строку фиксированной длины, то есть она сжимает её. Есть разные типы хэш-функций, важное свойство конкретно криптографический хэш-функции заключается в том, что, во-первых, она эффективна, то есть она считается буквально за миллисекунды; во-вторых, получив на вход произвольные данные, будь то pdf-файл, будь то какой-то сектор жесткого диска, эта функция выдаст нам независимо от входа какое-то фиксированное число бит. И важное свойство этого фиксированного числа бит - выходные биты не должны никаким образом видимо зависеть от входного числа бит. Основная идея хэш-функция состоит в том, что, имея выход хэш-функции, сложно найти входные данные. Если говорить более строгим математическим языком, если мы имеем образ хэш-функции (то есть выход хэш-функции), сложно найти прообраз хэш-функции, то есть вход, который привёл к этому выходу.

5. Поможет ли соль для улучшения стойкости паролей к атаке перебором, если злоумышленник получил доступ к серверу (рис. [2.8]):

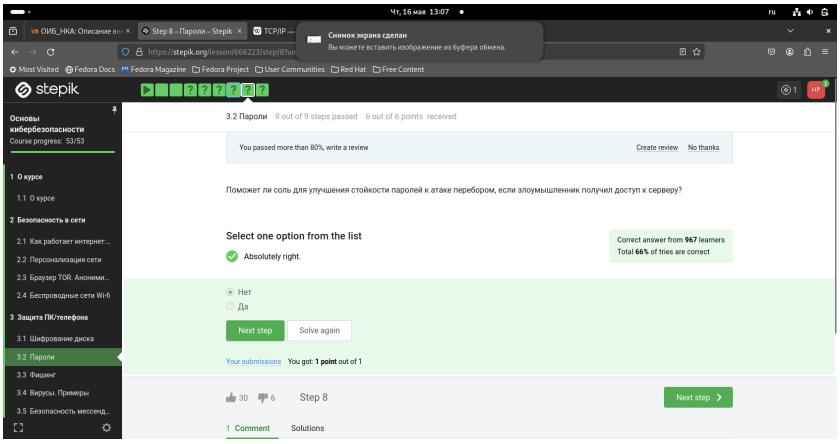

Рис. 2.8: Соль

Соль используется для того, чтобы увеличить стойкость пароля для пользователей, которые сами не догадались о стойкости своих паролей. Например, если у нас alice логинится с паролем 12345, что делает при этом сервер? Сервер не хочет хранить хэш от пароля 12345, потому что, скорее всего, злоумышленник имеет таблицы с самыми популярными паролями и их хэшами. Какая функция хэш используется в конкретном сервере, все знают. В основном, много серверов используют в своих таблицах SHA2 или SHA3, и, естественно, злоумышленник может для самых частых паролей посчитать эти значения, поэтому хранить хэш от 12345 абсолютно бессмысленно, и так понятно, что это соответствует паролю 12345. Поэтому умный сервер делает следующее: он сам генерирует какой-то действительно хороший пароль или кусочек какого-то пароля, не обязательно, чтобы он был длинный, добавляет эти случайные биты к паролю и хэширует результат. Под словом добавляет я имею в виду, что он приписывает кусочек после самого пароля alice, еще это называется конкатенация строк. Соль - это и есть тот кусочек случайного бита, который генерирует сервер в своей таблице. Следовательно, при доступе злоумышленника к серверу, у него будет доступ и к соли, а следовательно и к паролю, поэтому, соль не поможет для улучшения стойкости паролей к атаке перебором.

6. Какие меры защищают от утечек данных атакой перебором (рис. [2.9]):

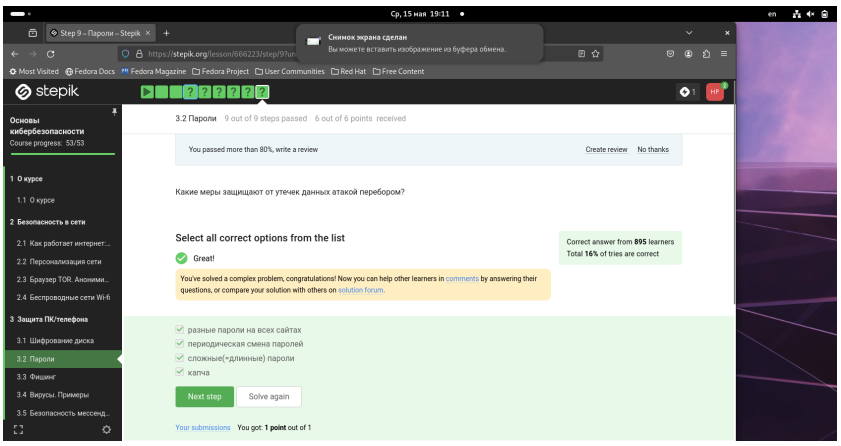

Рис. 2.9: Меры защиты

Все вышеперечисленные пункты смогут защитить аккаунт от утечки данных атакой перебором.

2.3 Фишинг

1. Какие из следующих ссылок являются фишинговыми (рис. [2.10]):

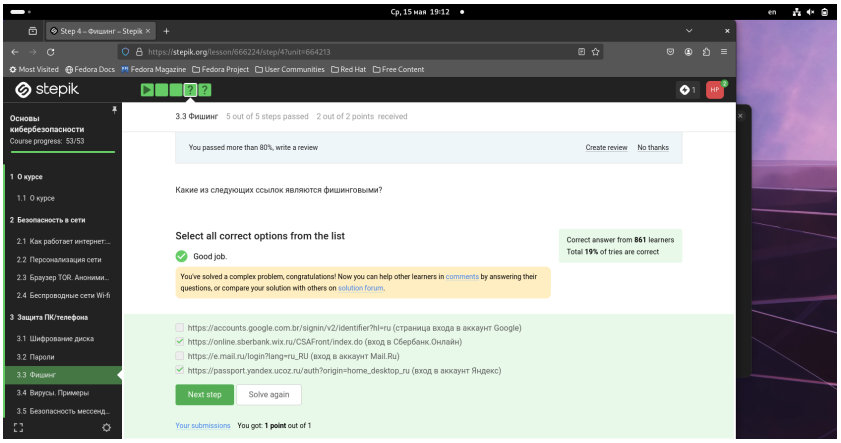

Рис. 2.10: Ссылки

Для того, чтобы адресный фишинг сработал, чтобы вас поймали на удочку, вам необходимо всего лишь некорректно с точки зрения безопасности отреагировать

на email, который содержит ссылку на какую-то фальшивую страницу. Допустим, вам пришло какое-то письмо с интересной ссылкой vk.club5.ru. Если вы зайдёте по этой ссылке, вы наверняка встретите интернет-страницу, идентичную странице известной соцсети ВКонтакте. Если вы не будете внимательными, захотите внести свои данные аутентификации в это окно и нажмёте «войти», то все ваши данные, которые вы вбили - юзернейм и пароль – естественно, при нажатии на клавишу «войти» не аутентифицируют вас в эту соцсеть, а перешлют эти данные хакеру. При этом сработает какой-то код, который сгенерирует сообщение, отправит его благополучно хакеру, вы при этом ничего не заметите. Другой пример фишинга - эта маскировка под известные веб-сайты только с другим доменным именем, начало может быть одинаковое или середина. В примере с vk.com понятно, на какой сервис это ссылается, ну а потом идет какая-то белиберда, которая не имеет ничего общего с реальной ссылкой. Естественно, никакие данные вводить сюда нельзя, но и более того, заметьте, что соединение вот с этим сайтом произошло не по HTTPS протоколу, а по небезопасному HTTP протоколу, это тоже звоночек, потому что с этого сайта лучше побыстрее уйти.

2. Может ли фишинговый имейл прийти от знакомого адреса (рис. [2.11]):

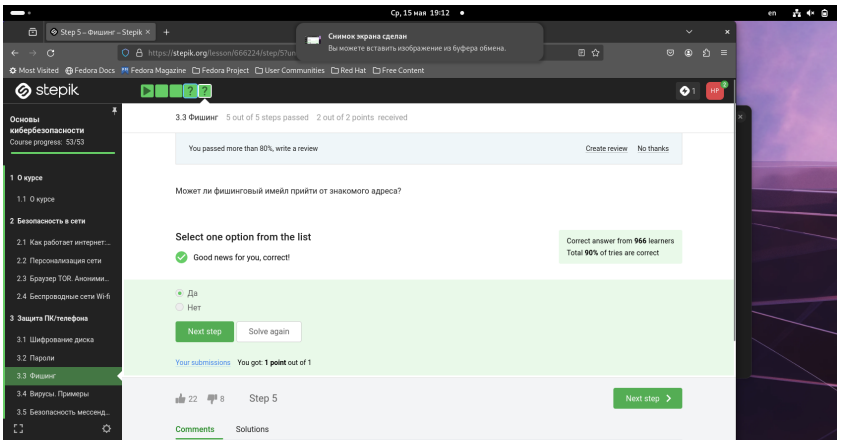

Рис. 2.11: имейл

Если вернемся к адресному фишингу имейлов, возникает вопрос, можем ли

мы получить вот такое фишинговое письмо от отправителя, которого мы знаем. Наверняка кто-либо из вас уже получал такой спам, и адрес отправителя оказывался адресом человек, которого вы как бы знаете, с которым общаетесь иногда. Это называется email spoofing/спуфинг от английского spoof – подменить. И спуфинг – это глобальный термин атак, есть IP spoofing - это подмена IP-адреса, есть email spoofing - подмена адреса отправителя. Суть состоит в том, что мы получаем фишинговое письмо от якобы знакомого нам человека, а на самом деле отправлено оно было не им. Это работает потому, что классический протокол отправки писем (SMTP) не включает проверку адреса отправителя, и поэтому у нас существуют вот такие надстройки на классический протокол SMTP, которые называются, например SPF (Sender Policy Framework). Это расширение протокола SMTP, которое позволяет получать письма только от некоторого списка авторизованных IP-адресов. Этим занимается, как правило, наш почтовый сервер, но и большинство почтовых серверов, которые позволяют нам регистрировать свои почтовые ящики (Google, Yandex, Mail.ru и другие крупные почтовые сервисы), конечно, позволяют это расширение подключать. Как правило, вместе с этим расширением вдогонку идет другое расширение, оно называется DMARC. Это протокол аутентификации email, он содержит правила, согласно которым сервер либо отправляет, либо блокирует письмо, и этим также заведует наш почтовый сервер.

2.4 Вирусы. Примеры

1. Email Спуфинг - это (рис. [2.12]):

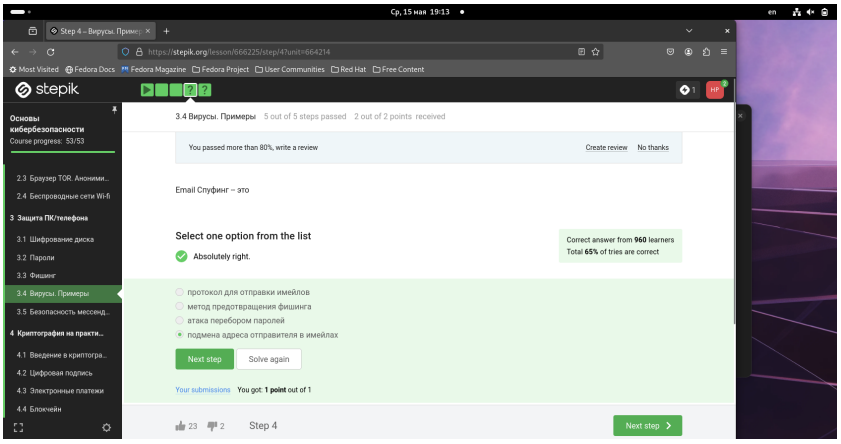

Рис. 2.12: Спуфинг

Ввиду комментария к предыдущему пункту, email спуфинг это подмена адреса отправителя.

2. Вирус-троян (рис. [2.13]):

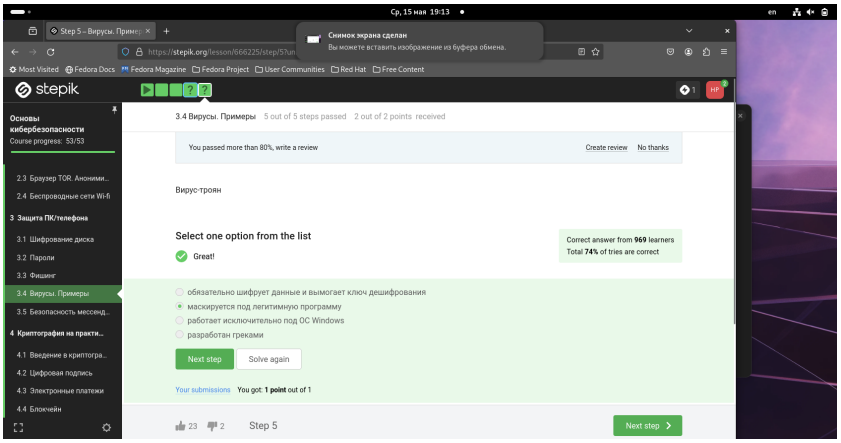

Рис. 2.13: Троян

Троян - это вирус, который проникает в систему под видом какого-то легитимного программного обеспечения, это аллюзия к троянскому коню.

2.5 Безопасность мессендеров

1. На каком этапе формируется ключ шифрования в протоколе мессенджеров Signal (рис. [2.14]):

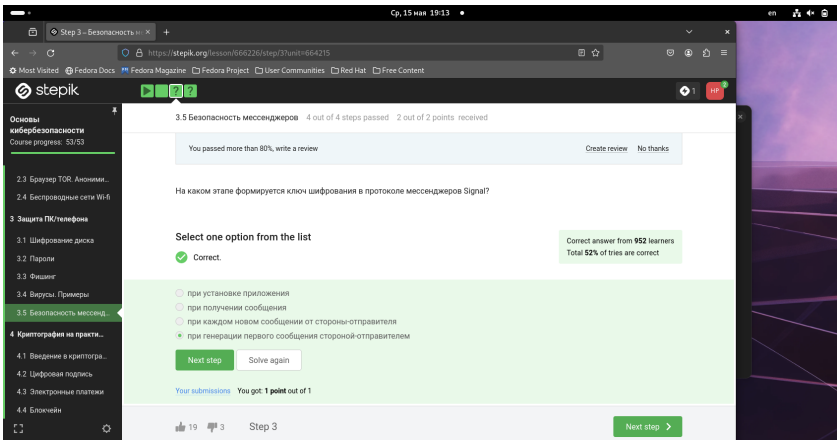

Рис. 2.14: Signal

Сквозное шифрование в протоколах Signal реализовано в двух больших шагах: это те же самые шаги, которые мы с вами уже встречали в протоколе TLS, когда говорили о безопасной коммуникации с веб-сервером. Вначале мы генерируем общий ключ, и уже после с помощью этого общего ключа отправляет сообщение уже зашифрованное под этим общим ключом Бобу. Кроме этого зашифрованного сообщения, она отправляет Бобу свой кусочек открытого ключа, при этом Боб, получив этот кусочек открытого ключа, имея какую-то свою секретную информацию, формирует тот же самый общий ключ, с помощью которого Алиса шифровала сообщение. Получив зашифрованное сообщение и вычислив общий ключ, Боб может уже дешифровать корректно это сообщение.

2. Суть сквозного шифрования состоит в том, что рис. [2.15]):

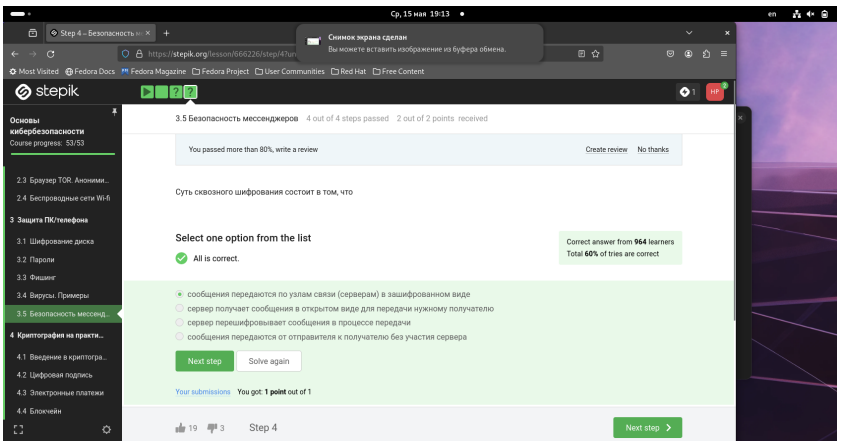

Рис. 2.15: Суть сквозного шифрования

Суть довольно простая: у нас есть два участника - Алиса и Боб, A и B, и сквозное шифрование заключается в том, что сервер, который передает сообщение, который направляет сообщение от Алисы к Бобу или от Бобу к Алисе, знает только то, куда эти сообщения должны быть направлены, но сообщения он передает в зашифрованном виде, то есть он как бы работает маршрутизатором сообщений, не зная о том, что он передает. Что происходит, если мы хотим отправить сообщение от Алисы к Бобу? Алиса шифрует свои данные, кладет на сервере шифр-текст с пометкой, что этот шифр-текст предназначен для Боба. Когда Боб заходит в сеть, сервер видит: «Ага, Боб зашел в сеть, надо обновить его сообщение», и отправляет шифр-текст от Алисы. Боб получает этот шифр-текст, дешифрует его, получает сообщение в открытом виде. При этом сервер не знает ни ключ, с помощью которого Алиса шифровала, ни тем более сообщение в открытом виде.

3 Выводы

В ходе работы мы изучили, что такое шифрование диска и как оно происходит, какой пароль лучше поставить и как обезопасить себя от утечки данных атакой перебором, что такое фишинг и спуфинг, а также изучили безопасность в мессенджерах.

Список литературы