МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная математика»
Кафедра: 806 «Вычислительная математика и программирование»

Лабораторная работа № 4
по курсу «Криптография»

Группа: М8О-307Б-21

Студент(ка): Ф. А. Меркулов

Преподаватель: А. В. Борисов

Оценка:

Дата: 19.04.2024

Москва, 2024

ОГЛАВЛЕНИЕ

1. Тема

Аутентификация с асимметричными алгоритмами шифрования.

2. Задание

1. Выбрать не менее 2-ух web-серверов сети Интернет различной организационной и государственной принадлежности.

2. Запустить Wireshark и используя Firefox установить https соединение с выбранным сервером.

3. Провести анализ соединения.

4. Сохранить данные необходимы для последующего сравнительного анализа:

Имя сервера, его характеристики.Версия TLS.

Выбранные алгоритмы шифрования.Полученный сертификат: версия. Валидность сертификата, валидность ключа,удостоверяющий центр.

Время установки соединения (от ClientHello до Finished)

5. Если список исследуемых серверов не исчерпан выбрать другой сервер и повторить соединение.

6. Если браузер поддерживал соединение TLS 1.2 принудительно изменить параметры TLS соединения в Firefox на TLS 1.0 (в браузере перейти по адресу “about:config” и изменить раздел SSL\TLS) и провести попытки соединения с выбранными серверами).

7. Провести сравнительный анализ полученной информации.

8. В качестве отчета представить результаты сравнительного анализа, выводы в отношении безопасности и корректности настройки веб-серверов с учетом их организационной и государственной принадлежности.

3. Теория

Аутентификация – проверка id на принадлежность пользователю.

Авторизация – проверка прав пользователя.

Hypertext transfer protocol secure (HTTPS) – безопасная версия HTTP, который является основным протоколом, используемым для передачи данных между веб-браузером и веб-сайтом. HTTPS зашифрован для повышения безопасности передачи данных.

HTTPS использует протокол шифрования для шифрования сообщений. Протокол называется Transport Layer Security (TLS), хотя ранее он был известен как Secure Sockets Layer (SSL). Этот протокол обеспечивает безопасность связи с использованием асимметричной инфраструктуры открытых ключей. Этот тип системы безопасности использует два разных ключа для шифрования сообщений между двумя сторонами:

Закрытый ключ - контролируется владельцем веб-сайта и хранится в тайне. Этот ключ хранится на веб-сервере и используется для расшифровки информации, зашифрованной открытым ключом.

Открытый ключ - доступен каждому, кто хочет взаимодействовать с сервером безопасным способом. Информация, зашифрованная с помощью открытого ключа, может быть расшифрована только с помощью закрытого ключа.

TLS-рукопожатие — это процесс, который запускает сеанс связи, использующий TLS. Во время TLS-рукопожатия две взаимодействующие стороны обмениваются сообщениями, чтобы подтвердить друг друга, проверить друг друга, установить алгоритмы шифрования, которые они будут использовать, и согласовать ключи сеанса.

TLS-рукопожатие происходит всякий раз, когда пользователь переходит на веб-сайт по протоколу HTTPS, и браузер сначала начинает запрашивать сервер-источник сайта. TLS-рукопожатие также происходит всякий раз, когда любые другие коммуникации используют HTTPS, включая вызовы API и запросы DNS-over-HTTPS.

TLS-рукопожатие происходит после того, как с помощью TCP-рукопожатия было установлено TCP-соединение.

В ходе TLS-рукопожатия клиент и сервер вместе выполняют следующие действия:

- Указывают, какую версию TLS (TLS 1.0, 1.2, 1.3 и т. д.) они будут использовать
- Решают, какие наборы шифров они будут использовать
- Проверяют подлинность сервера с помощью открытого ключа сервера и цифровой подписи центра сертификации SSL.
- Генерируют сессионные ключи, чтобы использовать симметричное шифрование после завершения рукопожатия

TLS-рукопожатия – это серия датаграмм, или сообщений, которыми обмениваются клиент и сервер. TLS-рукопожатие включает в себя несколько этапов, в ходе которых клиент и сервер обмениваются информацией, необходимой для завершения рукопожатия и обеспечения возможности дальнейшего общения.

Точные шаги в рамках TLS-рукопожатия зависят от используемого алгоритма обмена ключами и наборов шифров, поддерживаемых обеими сторонами. Алгоритм обмена ключами RSA, который в настоящее время считается небезопасным, использовался в версиях TLS до версии 1.3. Это происходит примерно следующим образом:

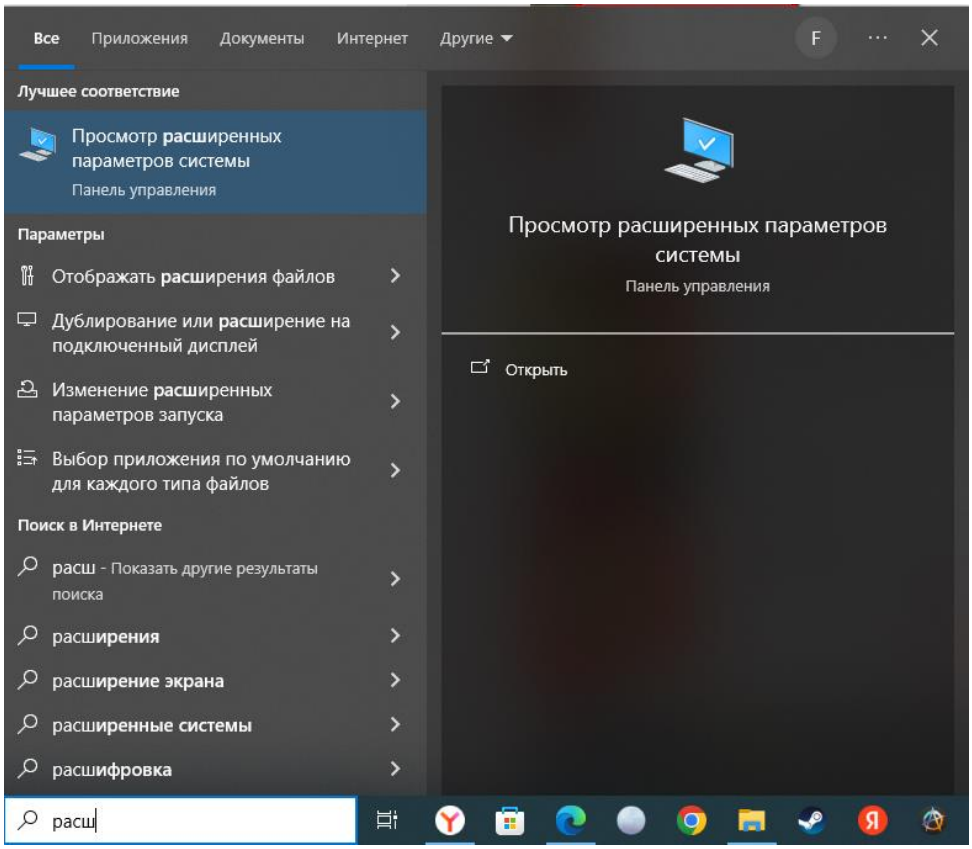
1. Client-hello: Клиент инициирует рукопожатие, посылая серверу сообщение "hello". В сообщении будет указано, какую версию TLS поддерживает клиент, поддерживаемые наборы шифров и строка случайных байтов "client random".
2. Server-hello: В ответ на сообщение "hello" клиента сервер отправляет сообщение, содержащее SSL-сертификат сервера, выбранный сервером набор шифров и "server random" – еще одну строку случайных байтов, генерируемую сервером.
3. Аутентификация: Клиент проверяет SSL-сертификат сервера в центре сертификации, который его выдал. Он подтверждает, что сервер является тем, за кого себя выдает, и что клиент взаимодействует с реальным владельцем домена.
4. Premaster secret: Клиент посылает еще одну строку случайных байтов, которая называется "premaster secret". Premaster secret шифруется открытым ключом и может быть расшифрован

сервером только с помощью закрытого ключа. (Клиент берет открытый ключ из SSL-сертификата сервера).

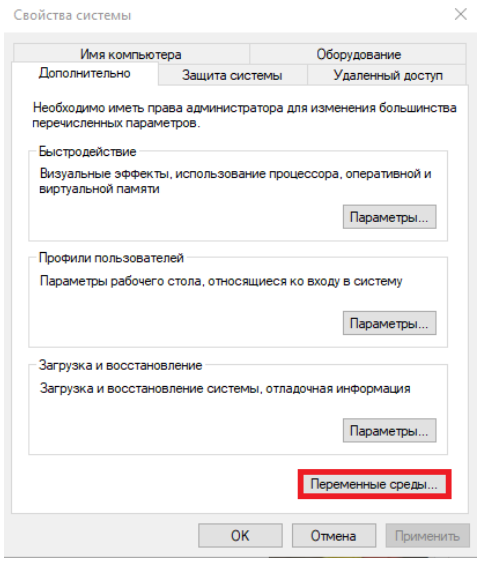
5. Используемый закрытый ключ: Сервер расшифровывает "premaster secret".
6. Создание сессионных ключей: Клиент и сервер генерируют сессионные ключи из random'a клиента, random'a сервера и premaster secret. Они должны прийти к одинаковым результатам.
7. Клиент готов: Клиент отправляет сообщение "ready", зашифрованное сессионным ключом.
8. Сервер готов: Сервер отправляет сообщение "ready", зашифрованное сессионным ключом.
9. Безопасность симметричного шифрования достигнута: Рукопожатие завершено, и связь продолжается с использованием сессионных ключей.

4. Ход лабораторной работы

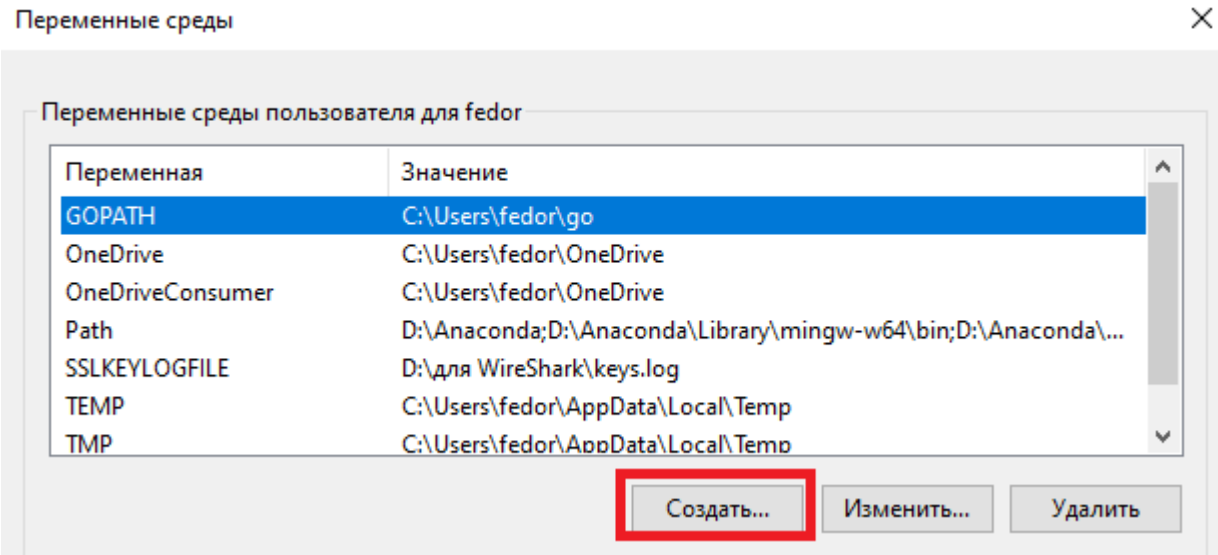
1. С помощью интернета я нашел как сделать расшифровку с помощью **pre-master secret key** сейчас распишу шаги того, что нужно сделать на ОС Windows, так как для меня это было не очевидно



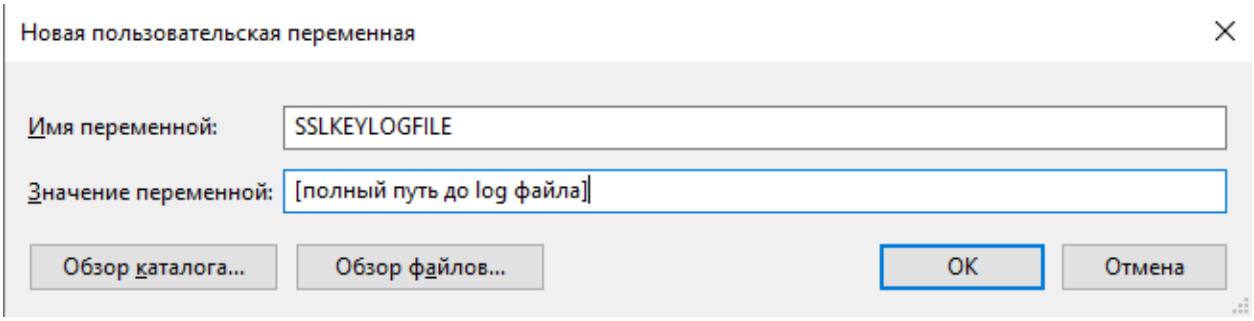
Переходим в “Просмотр расширенных параметров системы”



Нажимаем на кнопку “Переменные среды”

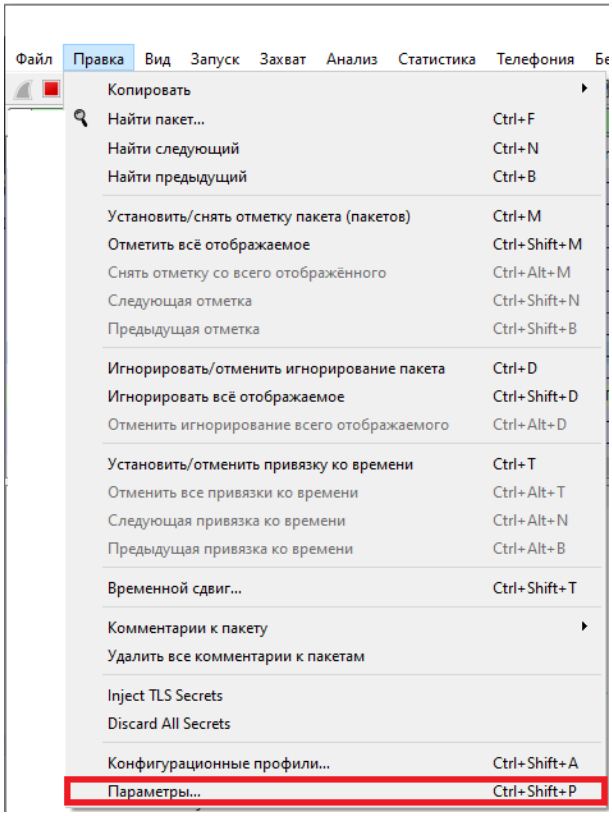


Нажимаем на кнопку “Создать”

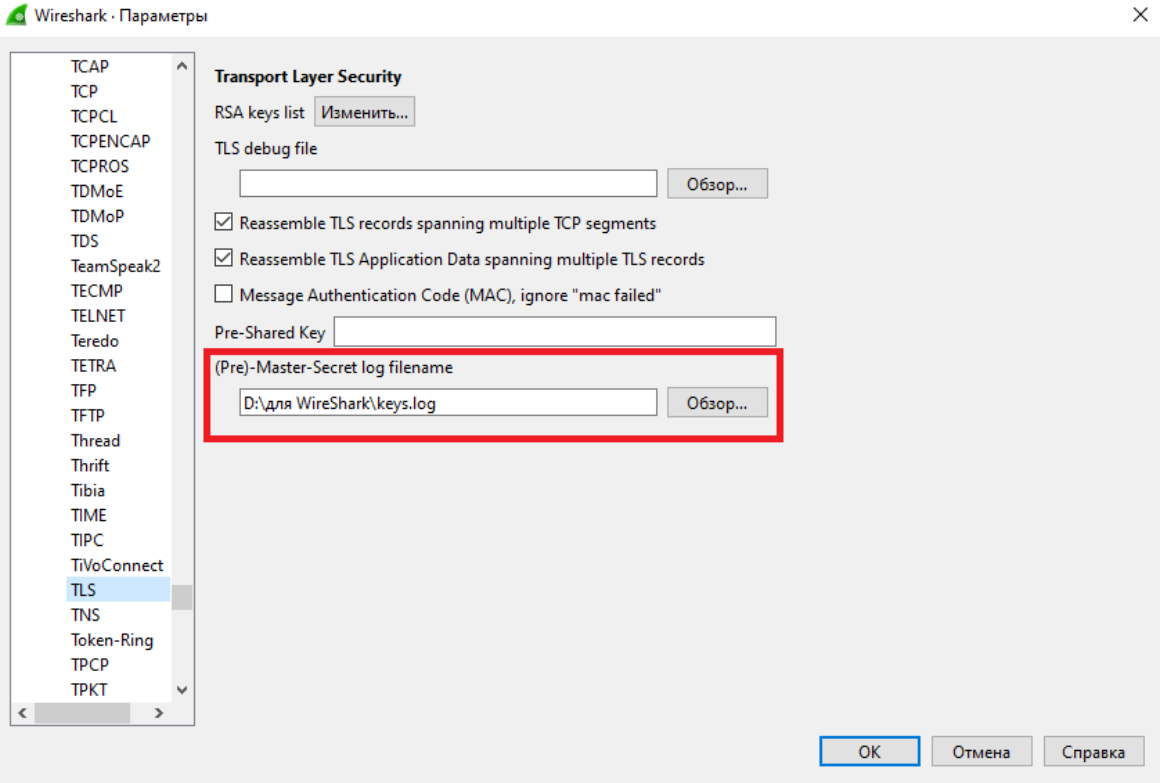


Записываем в “Имя переменной”: SSLKEYLOGFILE, а в “Значение переменной” путь до файла file.log в котором будут храниться логи

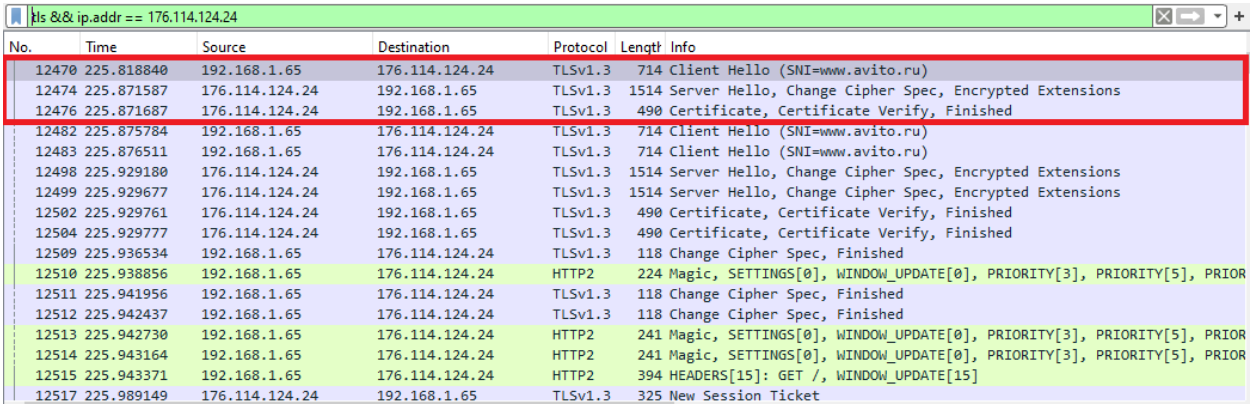
2. Укажем в Wireshark этот файл:



Далее в разделе “Protocols” находим “TLS”



3. Запускаем захват пакетов в Wireshark и ставим фильтр tls && ip.addr == 176.114.124.24. Это IP-адреса сайта https://www.avito.ru/ (который я определил с помощью команды ping www.avito.ru) он использует TLSv1.2 и его я выбрал первым:

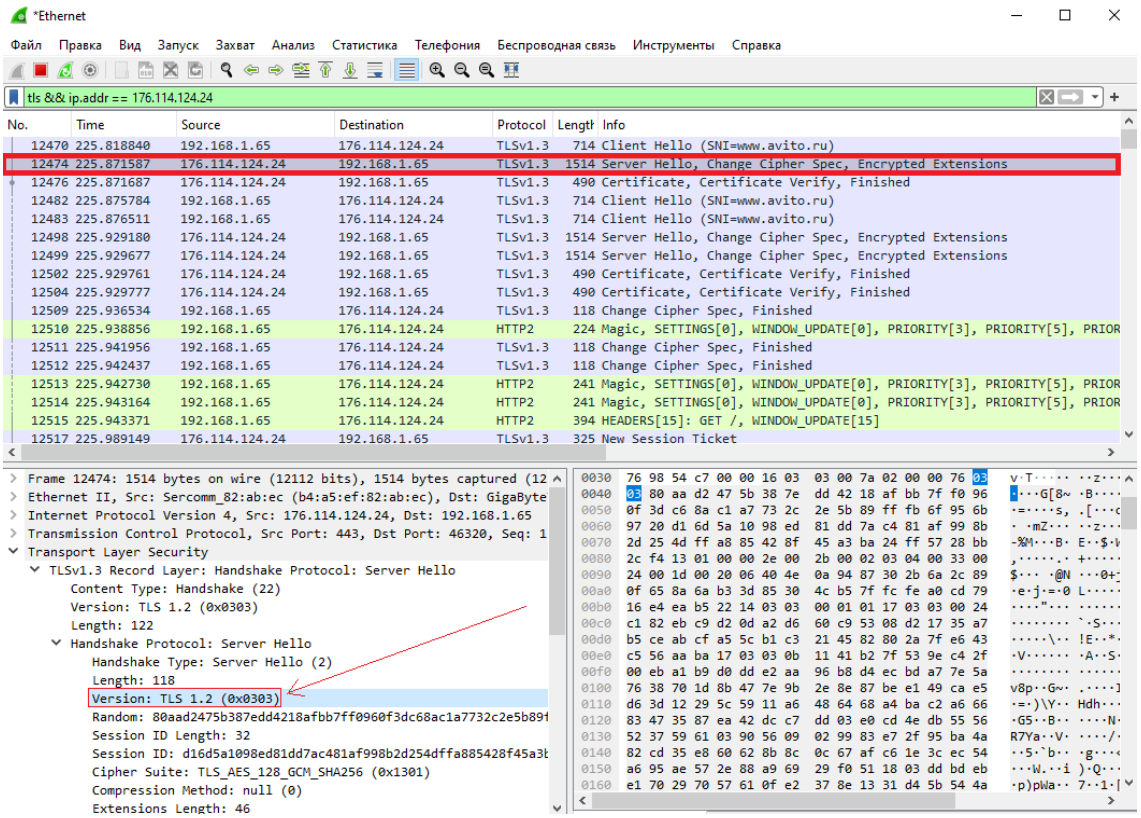


4. Анализ соединения:

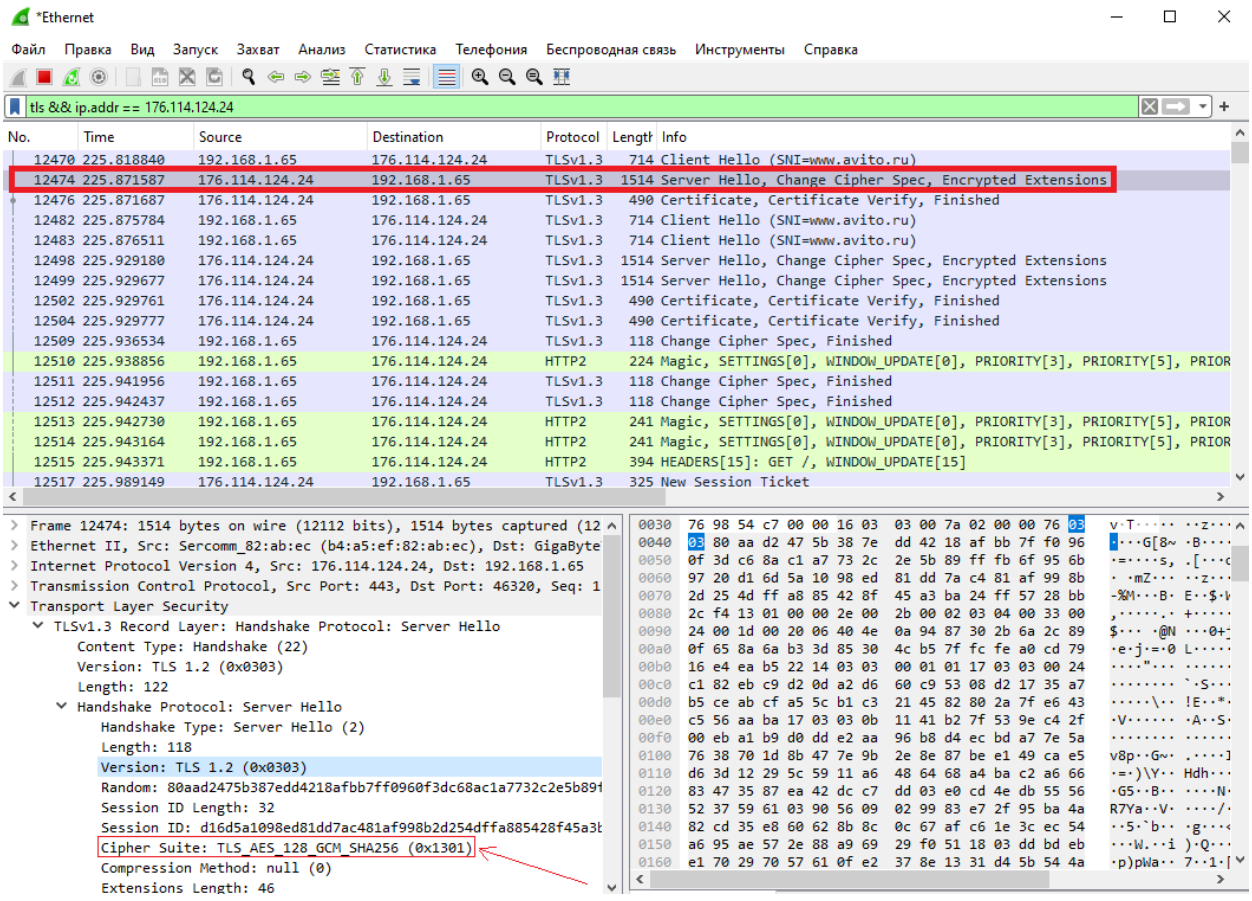
Имя сервера: “Client Hello (SNI= www.avito.ru)” (SNI - Server Name Indication), говорит о том, что – имя сервера www.avito.ru.

IP-адрес: 176.114.124.24

Версию TLS определим по Server Hello: Version: TLS 1.2 (0x0303).



Алгоритмы шифрования, выбранный сервером можно также узнать в Server Hello: Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301).



Версия сертификата: version: v3 (2).

Валидность сертификата и ключа: можно проверить даты начала и окончания действия сертификата.

validity

notBefore: utcTime (0)

utcTime: 2023-04-06 13:03:05 (UTC)

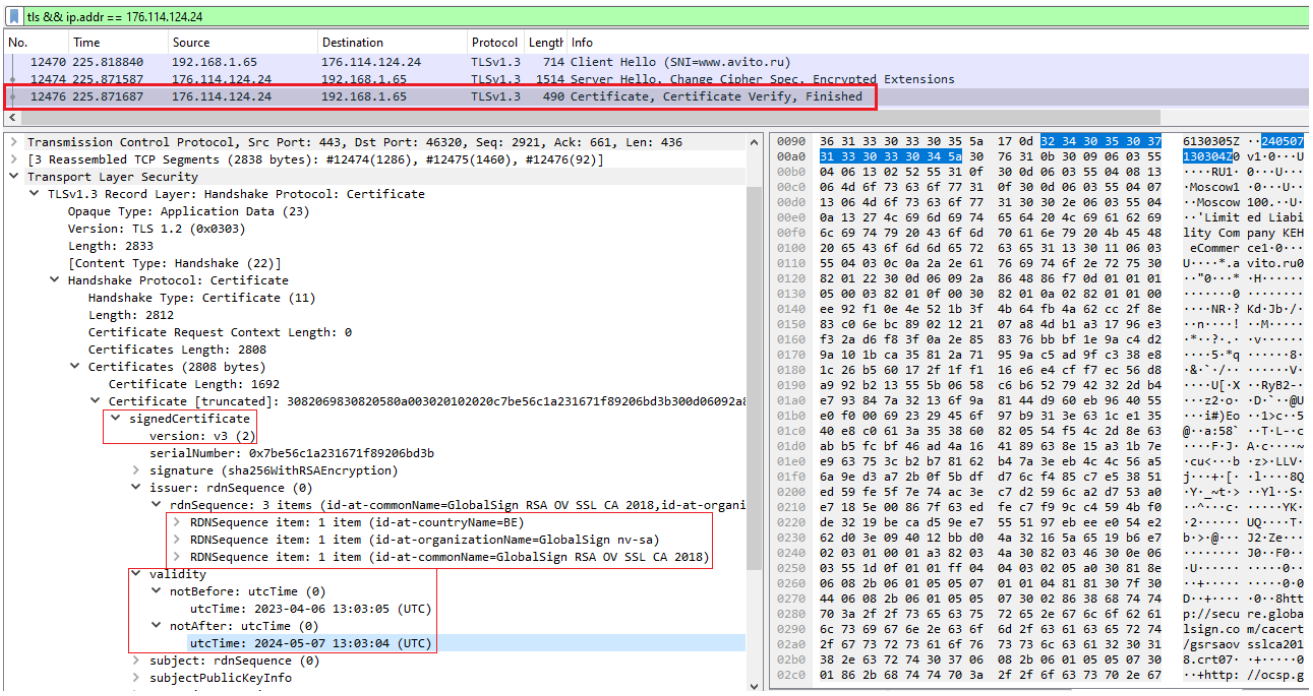
notAfter: utcTime (0)

utcTime: 2024-05-07 13:03:04 (UTC)

Сертификат валиден до 7 мая этого года.

issuer: rdnSequence (0)

rdnSequence: 3 items (id-at-commonName=GlobalSign RSA OV SSL CA 2018,id-at-organizationName=GlobalSign nv-sa,id-at-countryName=BE)



Время установки соединения легко вычислить как разницу времени между первым Client Hello и последним пакетом, содержащим сообщение Finished от сервера: 225.942437 - 225.818840 = 0.123597 секунды (~124 мс)

То же самое проделаем с сайтом **mos.ru**: tls && ip.addr == 212.11.151.57

Имя сервера: www.mos.ru

IP-адрес: 212.11.151.57

Версия TLS: Version: TLS 1.2 (0x0303)

Алгоритмы шифрования: Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)

Версия сертификата: version: v3 (2)

Валидность сертификата и ключа:

validity

notBefore: utcTime (0)

utcTime: 2023-10-13 09:57:50 (UTC)

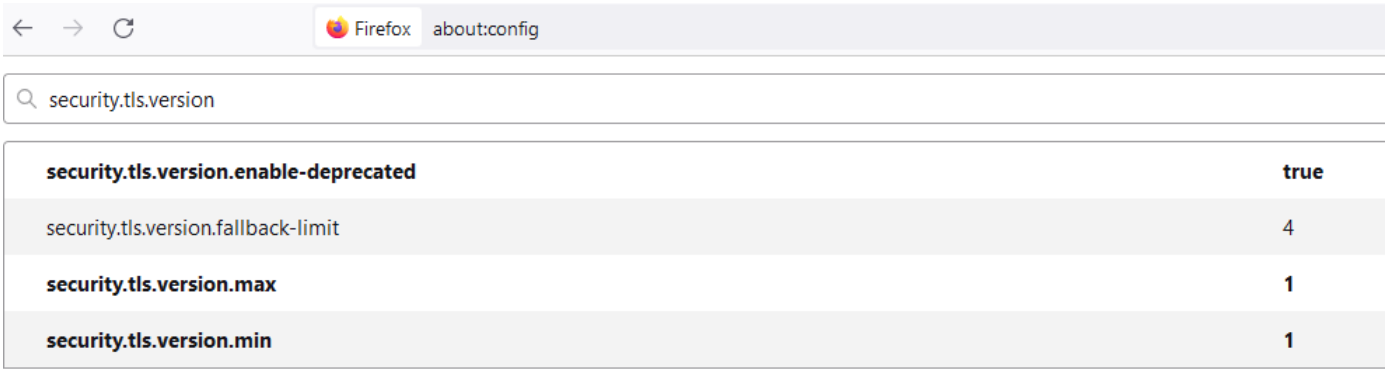
notAfter: utcTime (0)

utcTime: 2024-11-13 09:57:49 (UTC)

Удостоверяющий центр: rdnSequence: 3 items (id-at-commonName=AlphaSSL CA - SHA256 - G4,id-at-organizationName=GlobalSign nv-sa,id-at-countryName=BE)

Время установки соединения: 65 мс.

5. Теперь изменим версию TLS на TLS 1.0:



Avito:

```
66164 2740.269682   192.168.1.65      176.114.124.24    TLSv1     202 Client Hello (SNI=www.avito.ru)
66168 2740.322709   176.114.124.24    192.168.1.65      TLSv1    1514 Server Hello
66169 2740.322825   176.114.124.24    192.168.1.65      TLSv1    1514 Certificate
66170 2740.322825   176.114.124.24    192.168.1.65      TLSv1     351 Server Key Exchange, Server Hello Done
66172 2740.324570   192.168.1.65      176.114.124.24    TLSv1     155 Client Key Exchange, Change Cipher Spec, Finished
66173 2740.376294   176.114.124.24    192.168.1.65      TLSv1     336 New Session Ticket, Change Cipher Spec, Finished
```

Имя сервера: www.mos.ru

IP-адрес: 176.114.124.24

Версия TLS: Version: TLS 1.0 (0x0301)

Алгоритмы шифрования:

Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)

Версия сертификата: version: v3 (2)

Валидность сертификата и ключа:

validity

notBefore: utcTime (0)

utcTime: 2023-04-06 13:03:05 (UTC)

notAfter: utcTime (0)

utcTime: 2024-05-07 13:03:04 (UTC)

Удостоверяющий центр: rdnSequence: 3 items (id-at-commonName=GlobalSign RSA OV SSL CA 2018,id-at-organizationName=GlobalSign nv-sa,id-at-countryName=BE)

Время установки соединения: 107 мс.

mos.ru:

Имя сервера: www.mos.ru

IP-адрес: 212.11.151.57

Версия TLS: Version: TLS 1.0 (0x0301)

Алгоритмы шифрования:

Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)

Версия сертификата: version: v3 (2)

Валидность сертификата и ключа:

validity

notBefore: utcTime (0)

utcTime: 2023-10-13 09:57:50 (UTC)

notAfter: utcTime (0)

utcTime: 2024-11-13 09:57:49 (UTC)

Удостоверяющий центр: rdnSequence: 3 items (id-at-commonName=AlphaSSL CA - SHA256 - G4,id-at-organizationName=GlobalSign nv-sa,id-at-countryName=BE)

Время установки соединения: 36 мс.

Анализ

И **avito.ru**, и **mos.ru** используют TLS 1.2, что является хорошей практикой. Однако после принудительного изменения параметров TLS соединения в Firefox на TLS 1.0 и попытки соединения с выбранными

серверами они оба меня пустили, хотя соединение через TLS 1.0 может представлять собой риск безопасности.

Я считаю, что обоим серверам рекомендуется отключить поддержку TLS 1.0, чтобы исключить потенциальные уязвимости и соответствовать лучшим практикам безопасности.

Также сравнив tls-handshake для сервисов **Avito** и **mos.ru**, я выяснил, что:

- Оба сайта используют и ту же версию tls;

- Они используют различные алгоритмы шифрования

- Время установки соединения с **mos.ru** быстрее, чем с **Avito**. Это значит, что физический сервер находится ближе;

- Центр сертификации **mos.ru** и **Avito** находятся в Бельгии;

5. Выводы

В процессе выполнения данной ЛР я познакомился с принципами TLS-рукопожатий, прочитал зашифрованные пакеты с помощью Wireshark и сравнил защищенность сервисов avito.ru и mos.ru**.**

6. Список используемой литературы

1. https://www.comparitech.com/net-admin/decrypt-ssl-with-wireshark/
2. https://www.cloudflare.com/ru-ru/learning/ssl/what-happens-in-a-tls-handshake/
3. https://www.youtube.com/watch?v=efzQEAm7-Jc
4. https://www.cloudflare.com/learning/ssl/what-is-https/