ОРГАНИЗАЦИЯ ПРОЗРАЧНЫХ СЛОЁВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ДАННЫХ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Королёв В. В., Кобяк Е. Ф., Палатов Е. В.

Ролич О. Ч. – канд. техн. наук, доцент

В данной статье рассмотрено решение проблемы организации шифрования данных в каждом приложении в отдельности. Показан пример реализации данного подхода обеспечения безопасности данных и сделаны соответствующие выводы.

Любая деятельность на сегодняшний день так или иначе связана с процессом передачи информации, и часто ценные данные могут стать жертвой злоумышленников. Во избежание данной ситуации, разработчики добавляют в свои продукты различные способы шифрования передаваемых данных. При таком подходе данные будут в сохранности, но это требует дополнительных затрат времени со стороны разработчика. Идея AT-TLS заключается в прозрачности слоя шифрования для приложений. Разработчикам не требуется обеспечивать в создаваемых продуктах сложные алгоритмы шифровки и дешифровки, за всё это отвечает система, на которой устанавливаются продукты.

Работа AT-TLS построена на составлении правил для различных приложений. В отдельном файле находятся такие вещи как: ключи шифрования, сертификаты безопасности, правила, определяющие выполнять шифрование запроса или нет, а также возможность выставления различных уровней трейсинга, отвечающих за выводимую в лог информацию. В правилах можно указывать параметры, такие как: название приложения, тип запроса (отправка или получение), порт для прослушки и так далее. При получении или отправке запроса система пройдётся по всем правилам и если найдёт подходящее, то зашифрует данные. Если же нужное правило найдено не будет, то соединение останется незашифрованным. Схематичный пример работы AT-TLS изображен на рисунке 1.

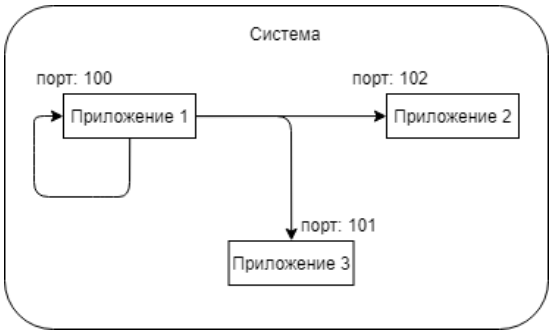

Рисунок 1 – Пример работы AT-TLS

В качестве примера представим систему с тремя приложениями. Для AT-TLS созданы правила:

- Приложение 1 отправляет зашифрованные данные для приложения 2 на 102-й порт, а также шифрует и дешифрует данные отправляемые и получаемые на свой 100-й порт;
- Приложение 2 дешифрует данные получаемые на 102-й порт;
- Приложение 3 дешифрует данные получаемые на 101-й порт.

Приложение 1 отправляет запрос себе же на 100 порт, тем самым сперва шифруя информацию, и дешифруя её при получении по 100-му порту. Также Приложение 1 отправляет запрос Приложению 2 на 102-й порт и Приложению 3 также на 102-й порт. При получении запроса на прослушиваемый порт, Приложение 2 дешифрует запрос. Приложение 3 не производит с запросом первого приложения никаких действий, так как правил для 102-го порта не имеет. При этом при отправке запроса на 101-й порт 3-го Приложения получится ошибка, так как Приложение 1 не шифрует данные отправляемые 3-му приложению, а 3-е приложение ждёт зашифрованные данные по данному порту.

Как было упомянуто ранее, AT-TLS предоставляет функцию автоматического логирования с разными уровнями трейсинга. В зависимости от настроек, в лог записываются разные сообщения, будь то только ошибки, или же информация о каждом шаге организации соединения и шифрования данных.

В ходе исследования был проведен обзор AT-TLS и представлен пример функционирования данной технологии шифрования соединений. Исходя из полученной информации, очевидно, что данный способ реализации безопасного общения приложений крайне эффективен и подойдёт для различных корпоративных систем.

Список использованных источников:

1. Application Transparent Transport Layer Security (AT-TLS) [Электронный ресурс]. – Режим доступа: https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.halx001/transtls.htm – Дата доступа: 24.03.2019.