Лабораторная работа №2. Протоколы удаленной аутентификации

1. Понятие аутентификации

Аутентифика́ция – процесс проверки подлинности идентификатора, предъявляемого пользователем.

Учитывая степень доверия и политику безопасности систем, проводимая проверка подлинности может быть односторонней или взаимной. Обычно она проводится с помощью криптографических способов.

- Односторонняя – серверу или веб-приложению достаточно убедиться в подлинности клиента, который инициирует соединение;
- Двусторонняя (взаимная) – для такого сценария каждый из участников должен убедиться в подлинности своего собеседника;

Любая система аутентификации представляет собой совокупность элементов, выполняющих ту или иную роль в реализуемом ей сценарии. К таким элементам относятся:

- **Субъект аутентификации** – лицо, проходящее процедуру аутентификации;
- **Характеристика субъекта (фактор)** – отличительная черта, характеризующая субъект;
- **Владелец системы аутентификации** – лицо, несущее ответственность и контролирующее работу системы;
- **Механизм аутентификации** – принцип, по которому осуществляется проверка подлинности предоставленного субъектом фактора;
- **Механизм предоставления прав** – механизм, обеспечивающий авторизацию, то есть предоставление тех или иных прав, приписанных данному субъекту, прошедшему проверку подлинности;

В общем случае процедуру аутентификации можно представить следующим образом:

1. Субъект инициирует процедуру аутентификации;
2. Субъект предъявляет один или несколько факторов аутентификации;
3. На основании механизма аутентификации принимается решение о подлинности субъекта;
4. В случае положительного решения, субъект наделяется правами доступа, присвоенными для него хозяином системы;

В качестве фактора аутентификации выступает то или иное свойство, являющееся отличительным для данного субъекта. К факторам аутентификации относят:

- **Знание** («пользователь знает») – тайные сведения, которыми обладает субъект аутентификации;
- **Обладание** («пользователь имеет») – устройство аутентификации (смарт-карта, eToken);
- **Существование** («пользователь существует») – биометрические данные;

2. Удаленная аутентификация

Под удаленной аутентификацией понимается осуществление процедуры аутентификации с использованием каналов связи. Основными проблемами данного процесса являются:

1. Обеспечение подлинности канала связи;
2. Защита механизма аутентификации пользователя от атак методом повтором (получение злоумышленником информации, передаваемой в процессе аутентификации подлинного клиента не должно позволить злоумышленнику пройти последующие процедуры аутентификации);

Основными методами обеспечения подлинности канала являются метод **запрос-ответ** и механизм **меток времени**

Метод "запрос-ответ" основан на использование некоторой случайной информации (запрос, Challenge), передаваемой пользователю B от пользователя A в случае, если пользователь A хочет проверить подлинность пользователя B. Схему этого метода можно представить в следующем виде:

1. Алиса генерирует случайное число, называемое запросом(Challenge);
2. Боб, получив запрос, прикрепляет к нему свои аутентификационные данные и подвергает его криптографическому преобразованию, а затем отправляет ответ (Response) Алисе;
3. Алиса, получив ответ, проверяет его, повторяя описанные выше операции на своей стороне, и на основании данной проверки принимает решение о подлинности сеанса связи;

Стойкость данной системы основана на том, что злоумышленник не знает секретной информации, принадлежащей Бобу - следовательно он не сможет подделать корректный ответ на полученный запрос.

Метод "меток времени" заключается в том, что в каждое пересылаемое сообщение добавляется специальная информация, называемая меткой времени (Time Stamp), которая описывает точное время отправки данного сообщения. Это позволяет каждому субъекту определить, насколько старо пришедшее сообщение и отбросить его в случае, если появится сомнение в его подлинности.

3. Протоколы, используемые при удаленной аутентификации

3.1. Password Access Protocol (PAP)

Самым простым и эффективным протоколом удаленной аутентификации является протокол доступа по паролю (Password Access Protocol, PAP)

Суть протокола заключается в аутентификации пользователя на сервере путем передачи последнему пары "логин-пароль", представляющей из себя идентификатор и информацию, известную лишь подлинному пользователю.

Основной проблемой данного протокола является то, что информация передается в открытом виде, а значит данный протокол неустойчив к атакам типа Sniffing.

Злоумышленник, обладающий доступом к открытому каналу связи и средствами перехвата пакетов может с легкостью получить пароль, что позволит ему пройти аутентификацию от лица другого пользователя.

Для повышения безопасности пароли могут передаваться не в открытом виде, а в виде хэшей, однако данная модификация не способна повысить стойкость к атакам типа Sniffing, так как злоумышленник может перехватить и хэш пароля.

3.2. Протокол CHAP

Протокол CHAP (Challenge-Handshake Authentication Protocol) - протокол удаленной аутентификации, основанный на методе "запрос-ответ"

Протокол нашел применение в технологиях RADIUS (Remote Authentication Dial In User Service) и EAP (Extensible Authentication Protocol). В самом простом случае (односторонняя аутентификация) протокол в точности повторяет схему метода "запрос-ответ":

1. Алиса генерирует случайное число N и отправляет его Бобу.
2. Боб, получив из запроса N, добавляет к нему свой пароль P и осуществляет вычисление дайджеста H1 = Hash(N, P). Полученный результат отправляется Алисе.
3. Алиса повторяет процедуры, выполненную Бобом на прошлом шаге и вычисляет значение дайджеста H2 = Hash(N, P1) от P1, которое хранится у Алисы в качестве пароля Боба. Если H1 совпадает с H2, пользователь считается аутентифицированным.

Существует модификация протокола, позволяющая проводить взаимную аутентификацию сторон. При этом производится следующая последовательность действий:

1. Алиса генерирует случайное число N1 и отправляет его Бобу вместе с запросом на аутентификацию (A);
2. Боб, получив запрос, генерирует собственное случайное число N2, которое вместе с Hash(N1, P_B) отправляется Алисе;

3. Алиса проверяет подлинность сообщения, содержащего ее зашифрованное случайное число и пароль Боба, а затем генерирует дайджест Hash(N2, P_A) и отправляет его Бобу;

4. Боб проверяет подлинность сообщения, полученного от Алисы и содержащего его зашифрованное случайное число.

3.3. Протокол использования одноразовых ключей S/KEY

Протокол S/KEY основан на независимом формировании клиентом и сервером последовательности одноразовых паролей, построенной на общем секрете K. В основе протокола лежит Метод Лампорта (Lamport's Hash Chain Method)

Пусть K - секретный пароль, известный как серверу, так и подлинному клиенту. Клиент вычисляет последовательность одноразовых ключей Y следующим образом:

$$P_1 = H(K),$$
$$P_2 = H(P_1) = H\big(H(K)\big),$$
$$\dots$$
$$P_N = H(P_{N-1}) = H(H\left(\dots H\big(H(P_1)\big)\right))$$

Сервер, независимо от пользователя может сгенерировать точно такую же последовательность, что позволяет использовать ее для проверки одноразовых паролей. После генерации паролей изначальный секрет K отбрасывается, сервер устанавливает P_N в качестве первоначального пароля пользователя и 1 в качестве текущего номера транзакции.

Процесс аутентификации выглядит следующим образом:

1. Пользователь запрашивает аутентификацию у сервера;
2. Сервер сообщает пользователю текущий номер транзакции I;
3. Пользователь передает серверу пароль с индексом N-I;
4. Сервер применяет хеш-функцию H к полученному паролю от пользователя и сверяет его со значением пароля, хранящимся на сервере.
5. В случае, если пароли совпадают, пользователь считается аутентифицированным, сервер увеличивает текущий номер транзакции на единицу и перезаписывает хранимый пароль пользователя паролем, полученным при аутентификации.

К недостаткам этого протокола можно отнести тот факт, что после исчерпания конечного множества одноразовых паролей мы не должны использовать его повторно, так как злоумышленник мог перехватить всю последовательность целиком. Это значит, что необходим механизм модификации исходных данных для процесса генерации последовательностей.

Чаще всего используют подход, основанный на передаче перед формированием последовательности одноразовых ключей случайного числа R от сервера к клиенту. Это случайное число, наряду с секретным паролем K ложится в основу пары K||R, которая используется в качестве базы для генерации последовательности. После исчерпания

одноразовых паролей для числа R, сервер передает клиенту новое случайное число и процесс повторяется.

4. Задания

4.1. Реализация протоколов PAP/CHAP

Целью данного задания является реализация базовых протоколов аутентификации PAP/CHAP в виде приложения. В интерфейсе приложения должны быть наглядно представлены:

- Исходные данные протокола (модули, ключи, секретные данные и т.п.);
- Данные, передаваемые по сети каждой из сторон;
- Проверки, выполняемые каждым из участников.

Процесс взаимодействия между сторонами протокола может быть реализован как с применением сетевых технологий, так и при помощи буферных переменных. Также необходимо выделить каждый из этапов протоколов для того, чтобы его можно было отделить от остальных.

Задание состоит из двух этапов. На первом этапе Вы осуществляете реализацию протокола PAP и одностороннего протокола CHAP. После того, как Вы получите работающее приложение, Вам необходимо расширить его функциональность и обеспечить поддержку двухстороннего протокола CHAP. Для генерации секретных параметров рекомендуется использовать криптографически стойкие генераторы случайных чисел, а в качестве хеш-функции использовать алгоритм SHA1.

4.2. Реализация протокола S/KEY

Целью данного задания является реализация протокола аутентификации S/KEY в виде приложения. В интерфейсе приложения должны быть наглядно представлены:

- Исходные данные протокола (модули, ключи, секретные данные и т.п.);
- Данные, передаваемые по сети каждой из сторон;
- Проверки, выполняемые каждым из участников.

Процесс взаимодействия между сторонами протокола может быть реализован как с применением сетевых технологий, так и при помощи буферных переменных. Также необходимо выделить каждый из этапов протоколов для того, чтобы его можно было отделить от остальных.

Необходимо обеспечить доступ ко всем паролям, сгенерированным в процессе инициализации протокола (например, вынести в отдельное окно). Для генерации секретных параметров рекомендуется использовать криптографически стойкие генераторы случайных чисел, а в качестве хеш-функции использовать алгоритм SHA1.