Протокол SSL v3.0 Краткое описание.

Сравнение с предыдущими версиями и протоколом PCT

Доступ к Internet расширил возможности проникновения посторонних в хранилища важной для компании информации. Однако при наличии упреждающей политики защиты сохранить ресурсы в безопасности существенно легче.

В настоящее время широко используется целый ряд технических решений, предназначенных для предотвращения конкретной опасности извне. Возможные последствия нарушения безопасности функционирования систем можно разделить на четыре группы - взлом, несанкционированный доступ, перехват и подстановка пакетов, блокировка.

Перечень наиболее распространенных решений приведен в Таблице 1:

Решение	Взлом	Несанкционированный доступ	Перехват и подстановка пакетов	Блокировка
Повышение уровня безопасности узла		X		
Firewalls	X	X		X
"Жесткая аутентификация"(Strong authentication device)		X		
Внутреннее разделение сети		X		X
Encryption	X			
Цифровые подписи			X	

Secure Sockets Layer

Протокол SSL обеспечивает установление канала шифрованной связи между Web-клиентами и серверами. В последних версиях SSL, помимо этого, обеспечивается идентификация клиента: сервер знает, какие клиенты с ним работают, и наоборот. SSL также обеспечивает независимость от алгоритма шифрования, так что отпадает необходимость пользоваться одним и тем же алгоритмом RSA, основанным на применении открытых (public) ключей.

Netscape в настоящее время работает над созданием Web-браузера и сервера, в которых вместо RSA используется шифровальная карточка Fortezza. Предыдущие версии SSL, слишком сильно опирались на RSA.

Протокол предоставляет такие возможности :

- server authentification
- data encription
- message integrity

Он расположен под протоколами уровня приложений (такими как HTTP,NNTP, Telnet, FTP и т.п.) и над протоколом TCP/IP.Данная стратегия позволяет SSL работать независимо .С применением SSL, одновременно на сервере и клиенте,ваши сообщения передаются в зашифрованной форме, гарантирующей приватность.

SSL использует технологии аутентификации и шифрования ,разработанные в RSA Data Security Inc. К примеру, Netscape Navigator экспортирует версию SSL, использующую 40-битный ключ для алгоритма RC4.

Структура протокола

SSL предсталяет собой многоуровневый протокол, и на каждом уровне message может включать следующие поля:

- lenght field
- description field
- content field

Можно условно выделить несколько модулей, где каждый служит своей цели:

1) record – принимает данные в блоках произвольного размера , и выполняет следующие операции:

a) fragmentation – конвертирует данные в записи размером 16К(или менее);

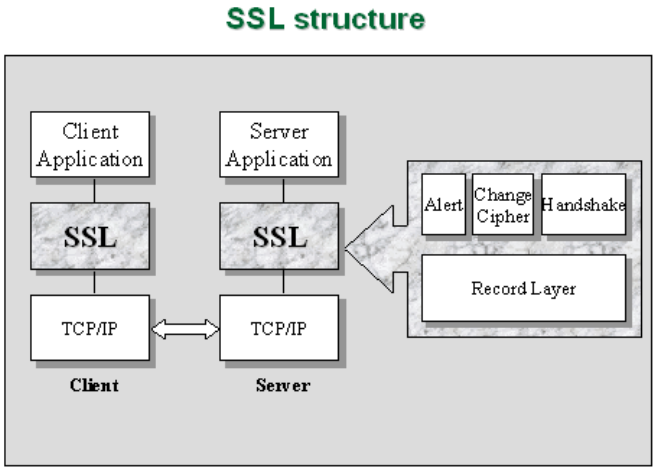
b) compression – при необходимости, выполняет компрессию записей;

c) encryption – защищает данные в записи посредством шифрования и MAC;

2) alert – оповещение об ошибках;

3) change cipher – оповещение об изменении стратегии шифрования;

a) handshake – используется при открытии или возобновлении сессии.



SSL принимает сообщение(message) для передачи, фрагментирует данные (в manageable bloks) и, при необходимости, выполняет компрессию данных,применяет MAC, шифрование, а затем передает результат транспортному уровню. На приемном узле данные дешифруется, проверяются, декомпрессируются, собираются и затем доставляются клиентскому уровню.

Состояния сессии и соединения

SSL – протокол, ориентированный на соединение.В данном контексте под состоянием (state) принимается информация, соотносящаяся со статусом и целью пакета

Сессия SSL является серией состояний. За координацию состояний клиента и сервера отвечает **SSL HandShake Protocol**, тем самым позволяет механизмам состояния протоколов каждого оперировать согласованно, не взирая на то, что это состояние не является в точности параллельным.

Логически, состояние представляется дважды:

- рабочее состояние(***operating***)
- ожидающее состояние(***pending***) - в течении handshake

Кроме того, существуют дополнительные раздельные состояния чтение и запись.Когда сервер(или клиент) получает ответное сообщение ***change cipher spec***, он переводит состояние ***pending read*** в ***current read***. И наоборот, когда абонент посылает сообщение ***change cipher spec***, то происходит смена состояний ***pending write*** - ***current write***. Когда завершается ***handshake***, клиент и сервер обмениваются сообщениями ***change cipher spec***, а затем переходят на заново согласованный ***cipher spec*** для последующей работы.

Сессия может поддерживать несколько безопасных соединений, при этом могут сосуществовать несколько одновременных сессий. Сессия SSL может включать следующие элементы:

- ***session identifier*** (идентификатор сессии) - произвольная последовательность байтов, выбранная сервером для определения состояния активной или возобновленной сессии
- ***peer sertificate***(сертификат тождественности) - X.509.v3 сертификат (данный эдемент может иметь параметр null)

- ***compression method*** (метод сжатия) - алгоритм, используемый для сжатия данных до шифрования
- ***chipher spec*** (метод шифрования) - определяет основной алгоритм шифрования данных (например null, DES) и алгоритм для MAC(MD5 или SHA). Он также определяет атрибуты шифрования:

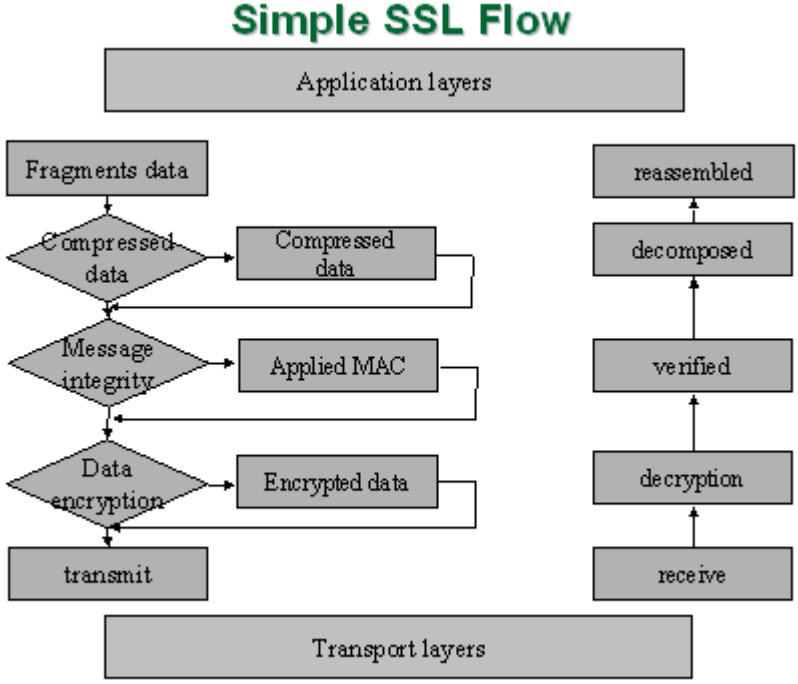
```
 enum { stream, biok } CipherType;
enum { true, false } IsExportable;
enum { null, rc4, rc2, des, 3des, des40, fortezza } BulkCipherALgorit

struct {
BulKCipherALgorithm bulk_cipher_algorithm;
MACAlgorithm mac_algorithm;
CipherType cipher_type;
IsExportable is_exportable;
uint8 hash_size;
uint8 key_materials;
uint8 IV_size;
} CipherSpec;
```

- ***master secret*** - 48-byte secret, разделяемый между сервером и клиентом
- ***is resumable*** - показывает - может ли сессия использоваться для инициации нового соединения;

Record layer

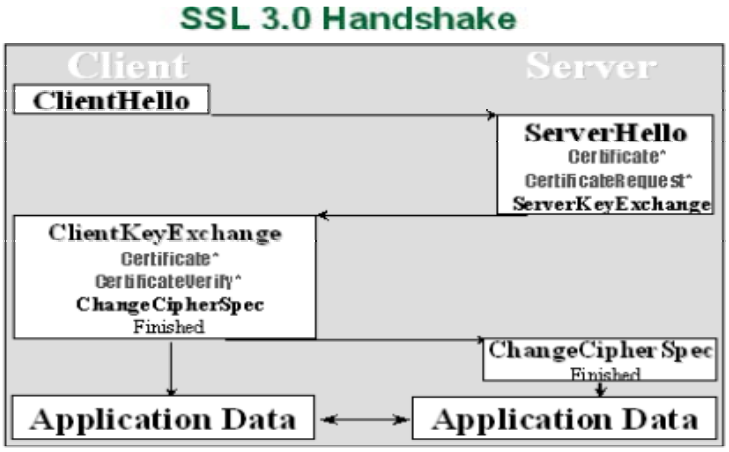
SSL Record Layer получает неинтерпритированные данные с вышестоящих уровней в виде непустых блоков произвольного размера.



Handshake protocol

Данный протокол устанавливает параметры криптографические параметры состояния сессии и работает в верхней части SSL Record Layer. Когда клиент и сервер первый раз начинают общение, они договариваются о версии протокола,

выбирают алгоритм шифрования, при необходимости удостоверяют друг друга, а также используют метод общих ключей для создания разделаемых секретов, выполняя все это в данном протоколе.



Handshake protocol flow

Клиент отправляет сообщение ***client hello***, на которое сервер должен ответить сообщением ***server hello***. В ином случае возникает фатальная ошибка и соединение прерывается. Сообщение ***client hello*** и ***sever hello*** используются для возможности усиления безопасности соединения между клиентом и сервером и включает следующие атрибуты:

- protocol version
- session ID
- cipher suite
- compression method

Дополнительно, генрируются два случайных значения: ***ClientHello.random*** и ***ServerHello.random***.

Вслед за обменом приветствиями, сервер высылает свой сертификат (если он подлежит авторизации). Дополнительно может быть затребовано сообщение ***server key exchange***, если сервер не имеет сертификата или сертификат служит только для цифровой подписи. Если сервер авторизован, он может затребовать сертификат клиента, при соответствии избранному ***cipher suite***.

Далее сервер посылает ***server hello done*** как завершение фазы обмена приветствиями и ждет ответа от клиента. Если сервер сделал запрос ***certificate request***, то клиент должен отправить сертификать или ***alert*** о его отсетствии. Затем посылается сообщение об обмене ключами ***client key exchange*** и содержание данного сообщения зависит от алгоритма общих ключей, избранного между ***client hello*** и ***sever hello***. Если клиент отправил сертификат с возможностью подписи, тогда сообщение посылается сообщение ***certificate verify***, заверенное цифровой подписью для точной верификации сертификата. В этот момент клиент посылает сообщение ***change cipher spec*** и переводит состояние ***Cipher Spec*** из ***pending*** в ***current*** . Затем немедленно отправляется завершающее сообщение(finished) под новые алгоритмы и ключи. В ответ сервер

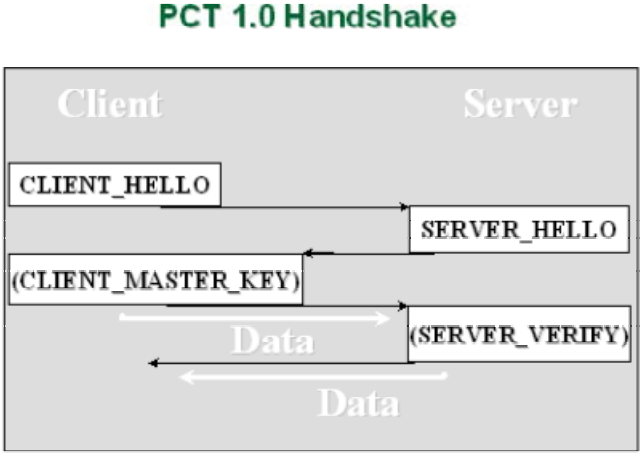
отправлает собственное сообщение ***change cipher spec*** , изменяет состояние и отправляет сообщение об окончании переговоров под новым ***Cipher Spec***. На данном моменте **handshake** завершается , и можно начинать обмен данными уровня приложений.

Private Communication Technology

Private Communication Technology представляет собой протокол безопасности для Internet. Как и SSL, протокол предназначен для предотвращения подслушивания клиент-серверных сообщений при всегда удостоверенных серверах, а клиентах - по запросам с сервера. Данный протокол исправляет и улучшает некоторые слабости SSL. Он разработан для приватности между двумя приложениями(клиентом и сервером) и так же , как и SSL, не зависит от них. В PCT для гарантированной передачи данных используется все тот же TCP/IP.Протокол начинается с договора сторон об алгоритме шифрования и симметричном ключе сессии, а также идентифицирует сервер клиенту(при необходимости и наоборот), используя при этом ассиметричный общий ключ. При передачи данных начинает использоваться уже симметричный ключ сессии.

Следует отметить, что в протоколе не предусмотренно верификация сертификатов. Наоборот, принимается, что протокол может управлять достоверностью сертификатов. Дополнительно к шифрованию и аутентификации, протокол удостоверяет целостность сообщений, используя смешанный, базирующийся на функциях, message authetification code.

Формат протокола PCT совместим с SSL. Сервер, использующий оба протокола, способен распознавать клиентов PCT и SSL, по полю version, расположенном одинаково в первом сообщении при процедуре handshake (в PCT наиболее значимый бит поля version установлен в 1)



Протокол PCT имеет следующие отличия от SSL в процедуре handshake :

- структура окружения и сообщения короче и проще, восстановленная сессия без аутентификации не требует дополнительных сообщений, кроме 1 послания в каждом направлении;
- переговоры об алгоритме и форматах для сессии предназначены, чтобы охватить большее количество характеристик, дополнительно к цифровому и

сертификационному типу добавляется функциональный и тип обменых ключей;
- при необходимоти сертификации договариваются о видах сертификатов и подписей;
- аутентификация сообщения исправлена так, что использует ключи, отличные от ключей шифрования, что позволяет значительно увеличить длину ключей, повышая надежность;
- были исправлены ошибки в стуктуре безопасности авторизации пользователей в SSL, и определение достоверности пользователя PCT методом вопрос-ответ сейчас зависит от типа определенного для конкретной сессии шифра. Определение пользователя в SSL не зависит от силы используемого в сессии шифра и от того, была ли аутентификация проведена для восстановления старой сессии или для создания новой.Т.е. используя метод взлома 'men-on-the-middle' для получения ключа сессии со слабым шифрованием, можно получить аутентификацию для сесии с сильным шифрованием. Если, к примеру, сервер обычно ограничивает ряд функций для высокосекретных сессий, то эта слабое место позволяет обмануть ограничения.
- Было добавлено поле предварительной верификации к фазе handshake, которое позволяет проверить тип шифра - не было ли вмешательства в этап переговоров, выполненных явно (в версии SSLv3.0 используется тот же механизм, но совместимость с SSLv2.0 разрушает эту возможность, так как позволяет хакеру просто сменить номер версии и шифр).

Заключение

Существует ряд общедоступных публикаций средств шифрования, но в ряде стран запрещается импорт, а в других - экспорт, подобных средств. Это приводит к определенным сложностям для межнациональных компаний (при определении каким законам следовать), тем более, что не было ясного прецендента по данногму вопросу в юрисдикции, в связи с Internet. Когда рассматривается применение вышеописанных средств, следует помнить о законах в отношении ввоза и вывоза криптографических ключей и алгоритмов. А эти законы существенно меняются от страны к стране.

В настоящий момент существуют следующие приложения шифрования для Internet:

1. для шифрования электронной почты(Pretty Good Privacy);
2. для защищенных WWW-транзакций(Secure Sockets Layer, Secure HyperText Transfer Protocol);
3. большое количество шифрованных реализаций для FTP и Telnet;

Однако, одно только шифрование не является достаточным средством в агрессивной среде, подобной Internet. В целом, проблема - в области управления ключами, доверии надежности програмного обеспечения и утраты данных. Многие производители неохотно включают криптографию из-за экспорт-контроля, налагаемого на их продукты в этом случае.

Гиберт Дмитрий. 911 гр.

При написании эссе использовалась следующая литература:

1) The SSL Protocol Version 3.0

   Alan O. Freier, Philip Karlton, Paul C. Kocher.

   November 18, 1996

2) Безопасность в компьютерных сетях. Microsoft Internet Security Framework: протоколы SSL и PCT

   Моисеев Дмитрий