Подход к подготовке SSL VPN-систем к сертификации

С. О. Мамаева
Svetlana.Mamaeva@emc.com
EMC, Санкт-Петербург

SSL VPN-системы (SSL — Secure Sockets Layer, VPN — Virtual Private Network) являются широко распространённой разновидностью систем сетевого доступа удалённых пользователей к внутренним корпоративным ресурсам бизнес-компании и основаны на транспортном протоколе SSL. Этот протокол предоставляет клиентам универсальный инструмент доступа к удалённым ресурсам — Интернет-браузер, а также несложные средства настройки соединения клиента с сервером. В данной статье предлагается подход к разработке SSL VPN-системы, ориентированный на задачу успешного прохождения независимой сертификации. Сертификация является обязательной в этом сегменте рынка, ориентация на сертификацию должна пронизывать весь процесс разработки, и только в этом случае становится возможно её пройти. Предложенный в работе подход определяет жизненный цикл SSL VPN-системы: сбор и анализ требований заказчика и сертификационного центра, разработка системы, сбор данных о системе, тестирование, настройка и модификация, предварительная сертификация, сбор и отправка данных в центр сертификации. В работе предлагается методика сбора данных, которая основывается на создании различного вида матриц соответствия (Matrix of Compliance), подробно рассматриваются различные виды тестирования SSL VPN-системы. Также рассказывается о мотивации к созданию представленного подхода, приводятся данные по его апробации и эволюции.

Ключевые слова: SSL VPN-системы, тестирование, сертификация.

Введение

В современных компаниях и организациях информационные ресурсы размещаются на различных серверах и находятся, практически, в любой точке мира. Это очень удобно, так как пропадает необходимость держать большое количество данных и необходимое ПО на локальных компьютерах. Технология организации сетевого доступа к внутренним корпоративным ресурсам для удаленных пользователей носит название VPN (Virtual Private Network) [27], а системы, работающие по этой технологии, называются VPN-системами.

В настоящий момент наиболее широкое распространение получили так называемые SSL VPN-системы, которые основываются на транспортном протоколе SSL (Secure Sockets Layer). Популярность этого протокола обусловлена тем, что он обеспечивает доступ к удалённым данным с помощью Интернет-браузер, а процесс конфигурации соединения с сервером является достаточно простым. Разработка новых SSL VPN-систем является распространённым бизнесом, поскольку подобные системы просты в обслуживании, не требуют дополнительных затрат и легко внедряются. По оценке Gartner Group, в последние годы было развернуто около 3 миллионов рабочих мест с удаленным доступом по SSL VPN, а общая прибыль от них составила порядка 400 миллионов долларов [3].

Основными странами-разработчиками SSL VPN-систем являются США, Япония и страны Евросоюза. Основными компаниями-разработчиками SSL VPN-систем являются Aventail, Check Point Software Technologies, Cisco Systems, Citrix Systems, Juniper Networks, Nokia и Nortal. Существует также большое количество средних и небольших компаний, занимающихся разработкой этих систем. Такое разнообразие связано с тем, что потребности отдельных заказчиков очень индивидуальны, так как основаны на различных видах бизнеса, и для удовлетворения этих требований необходима значительная доработка стандартных систем, а иногда и разработка новых. Кроме того, западные производители SSL VPN-систем имеют многочисленные аутсорсинговые группы в разных странах, в том числе и в России.

Безопасность является одним из ключевых аспектов SSL VPN-системы. Это связано с тем, что система должна предоставлять доступ к данным на удаленных компьютерах и в то же время обеспечивать их защищенность от взломов и атак хакеров. Безопасность

гарантируется официальной сертификацией, которая осуществляется соответствующими независимыми организациями[1]. Наличие сертификата является обязательным условием передачи системы заказчику. Поэтому важной задачей SSL VPN-системы является подготовка к сертификации. Для этого требуется системный подход, так как если разработать систему без учёта требований к сертификации, то у производителей не будет возможности гарантировать нужные значения параметрам, которые влияют на безопасность. Однако в настоящий момент в литературе не представлено описание подобных подходов.

Автором статьи были проанализированы несколько основных публикаций, посвящённых SSL VPN-системам. Книга [5] содержит основные сведения о концепции создания SSL VPN-систем, а также возможные их модификации. Там поднимаются вопросы сертификации системы, однако никаких рекомендаций по сертификации системы не приведено. Одними из основных объектов сертификации системы являются алгоритмы шифрования. Тонкости применения алгоритмов шифрования обсуждаются в книгах [1,7,8]. Там же рассматриваются подходы к передаче и защите каналов передачи данных систем, а также основные достоинства и недостатки вариантов доступа к системам, но не приводятся системные подходы к сертификации SSL VPN-систем.

В данной статье предлагается подход к разработке SSL VPN-системы, ориентированный на задачу успешного прохождения независимой сертификации. Сертификация является обязательной в этом сегменте рынка, ориентация на сертификацию должна пронизывать весь процесс разработки, и только в этом случае её можно пройти успешно. Предложенный в работе подход определяет жизненный цикл SSL VPN-системы: сбор и анализ требований заказчи-

[1] Национальный институт стандартов и технологий США (National Institute of Standards and Technology, NIST) [38] — разрабатывает стандарты в области компьютерной безопасности для федеральных проектов США; Федеральная служба по техническому и экспортному контролю России (ФСТЭК) [31] осуществляет специальные и контрольные функции, а также межведомственную координацию и взаимодействие в области государственной безопасности; Международный центр сертификации и качества (International Center for Quality Certification, ICQC) [40] — группа независимых нотифицированных органов Европейского союза, имеющих право проверять и сертифицировать качество программных систем.

ка и сертификационного центра, разработка системы, сбор данных о системе, тестирование, настройка и модификация, предварительная сертификация, сбор и отправка данных в центр сертификации. В работе предлагается методика сбора данных, которая основывается на создании различного вида матриц соответствия (Matrix of Compliance), подробно рассматриваются разные виды тестирования SSL VPN-системы. Также рассказывается о мотивации к созданию представленного подхода и приводятся данные по его апробации и эволюции.

1. Определение SSL VPN-системы

SSL VPN-система состоит из клиентской части, которая настраивается на компьютере удалённого пользователя, и серверной, которая реализует централизованный доступ к данным. Типовая архитектура SSL VPN-системы представлена на рис. 1.

Клиентская часть включает в себя, как правило, Интернет-браузер или независимого клиента (компонента «Независимый клиент»), в который встроены дополнительные сервисы для авторизации и пр. (компонента «Клиентские компоненты»). Независимый клиент необходим в ситуации, когда доступ к корпоративным ресурсам не может осуществляется с помощью Интернет-браузера (например, последние запрещены в компании). Компонента «Сетевые драйверы» отвечает за дополнительные возможности, используемые сервисами сетевого тоннеля и помощника транспортного уровня — поддержку виртуального сетевого адаптера, переадресацию пакетов данных и др. Компонента «Клиентская часть SSL-протокола» отвечает за реализацию SSL-протокола на стороне клиента. Наконец, «ОС клиента» — это операционная система, установленная на клиентском ПО. Это могут быть различные варианты Windows (для рабочих станций, для ноутбуков и т. д.), мобильные ОС (Windows Mobile, iOS, Android, Blackberry OS, Symbian OS) и т. д. Важно, что все возможные ОС клиентов тщательно фиксируются в рамках разработки SSL VPN-системы, так как для них должны быть проведены соответствующие проверки на безопасность и надёжность. Список поддерживаемых клиентских операционных систем обычно указывается в пользовательском документе Release Notes (сопроводительный документ выпускаемой системы).

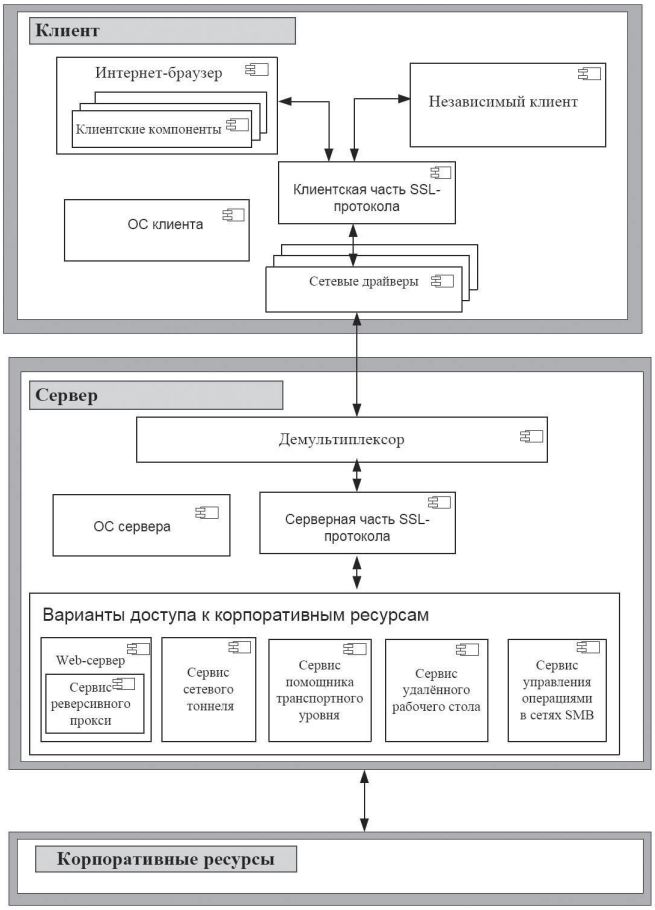

Рис. 1. Типовая архитектура SSL VPN-системы

Клиентское ПО SSL VPN-системы, установленное на разных удалённых компьютерах, создаёт многочисленные сетевые соединения к серверной части SSL VPN-системы и передаёт уникальный идентификатор клиента сетевому демону[2].

[2]Сетевой демон (network daemon) — обобщённое название для SSL VPN-клиентов к серверной части.

Серверная часть SSL VPN-системы является, как правило, отдельным специализированным компьютером. Этот компьютер, в числе прочего, имеет защиту от вскрытия — его ПО при несанкционированном вскрытии самоуничтожается. На сервер поступают данные от всех клиентов. Их обработкой занимается «Демультиплексор», который анализирует сетевые пакеты, приходящие на один сетевой адрес и порт, а также, используя идентификатор клиента, распознаёт, от какого сетевого демона пришёл запрос. Компоненты «Клиентская часть SSL-протокола» и «Серверная часть SSL-протокола» взаимодействуют друг с другом, выполняя процедуру аутентификации (распознавания) и установку соединения. После завершения процесса аунтентификации компоненты «Клиентская часть SSL-протокола» и «Серверная часть SSL-протокола» «договариваются» об используемых криптографических алгоритмах и формируют общие ключи для установки защищенного соединения. В качестве операционных систем для серверов SSL VPN-систем (компонента «ОС сервера») используют обычно Linux или Solaris.

В рамках одной системы может быть реализовано несколько разных вариантов доступа к корпоративным ресурсам (группа серверных компонент «Варианты доступа к корпоративным ресурсам»): реверсивный прокси, сервис сетевого тоннеля, сервис помощника транспортного уровня, сервер управления операциями в сетях SMB. Это связано с тем, что одна SSL VPN-система должна обслуживать несколько офисов компании, которые имеют различные потребности в корпоративных ресурсах. Далее подробно рассмотрены варианты доступа к корпоративным ресурсам, реализуемые в SSL VPN-системах.

1.1. Сервис реверсивного прокси (Reverse Proxy Service)

Этот сервис предназначен для внутрикорпоративных Web HTTP/HTTPS ресурсов. С помощью этого сервиса, пользователь открывает браузер, проходит авторизацию и по щелчку манипулятора мыши на иконке или на названии корпоративного Web-портала появляются необходимые данные, т. е. у пользователя возникает впечатление, что он находится внутри локальной сети компании. Этот способ очень прост в использовании —

удалённому пользователю не нужно устанавливать никаких дополнительных приложений, обладать правами администратора на клиентском компьютере или устанавливать виртуальные сетевые адаптеры. Необходим только Интернет-браузер с поддержкой SSL-протокола[3]. Типичным примером такого подхода является удалённый доступ к Microsoft SharePoint. Однако этот способ доступа является самым тяжёлым и трудоёмким в реализации, а также оказывается самый потенциально опасным. Для работы сервера реверсивного прокси необходим Интернет-браузер, который выдаёт пользователю гипертекстовую информацию (т. е. данные в HTML-формате) и поддерживает для удалённых ресурсов процесс преобразования HTML-страниц в структуру, которая может быть корректно распознана и обработана клиентской частью сервиса (так называемое *реверсивное проксирование страниц*).

1.2. Сервис сетевого тоннеля (Tunnel Service)

Этот сервис предназначается для доступа к сетевым ресурсам на основе протоколов туннелирования.

1.3. Сервис помощника транспортного уровня (Port Forwarder Service)

Этот сервис предназначен для предоставления удалённого доступа к приложениям, предоставляющим для соединения порты, не являющиеся http-портами (т. е. эти приложения не являются Web-приложениями). Примерами таких приложений являются FTP и Telnet. Сервис помощника транспортного уровня требует предоставления больших прав пользователю для доступа к ресурсам и позволяет упаковывать данные TCP/UDP сессий в рамках протокола SSL и использует для этого либо простой слушающий сетевой сокет, либо встраивает в сетевой SSL-стек драйвер, позволяющий прозрачно перенаправлять пользовательское соединение на тот же самый сокет. Преимущество данного сервиса в том, что данные не подвергаются никакой дополнительной обработке, т. е. SSL-

[3]На данный момент все современные браузеры (Internet Explorer, начиная с версии 6.0, Mozilla Firefox начиная с версии 2.0, все версии Google Chrome и т. д.) поддерживают SSL-протокол.

соединение устанавливается между клиентом и сервером, и сервер, в свою очередь, устанавливает его между собой и корпоративными ресурсами.

1.4. Сервис удалённого рабочего стола (Remote Desktop Service)

Этот сервис предназначен для удалённого доступа к терминальным серверам (Terminal Services Access Service) по протоколу Microsoft RDP или Citrix ICA[4]. Данный сервис использует готовые компоненты Microsoft/Citrix путём перенаправления их соединений в SSL VPN-канал. Сервис рассчитан на использование ресурсов высокопроизводительного сервера большим количеством менее производительных клиентских машин.

1.5. Сервис управления операциями в сетях SMB (Operation Management Service)

Некоторые производители SSL VPN-систем встраивают в свои системы специальный сервис, похожий на Windows-проводник (Windows Explorer) для просмотра удалённых SMB-сетей[5]. Это позволяет пользователю получать доступ к удалённым корпоративным ресурсам без установки дополнительных приложений и избежать сложного конфигурирования. Этот сервис очень часто базируется на решении Samba[6] и реализуется как Java Applet.

2. История разработки подхода

Предлагаемый в статье подход берет начало с момента появления

[4]Microsoft RDP (Remote Desktop) и Citrix ICA (Independent Computing Architecture) — протоколы для отображения и управления удалённым рабочим столом пользователя. Citrix ICA является более дорогим решением, чем Microsoft RDP. В свою очередь, ICA обладает более высокой скоростью, поддерживает UNIX-, DOS- и Web-клиентов.

[5]SMB (Service Message Blocks) — протокол доступа к ресурсам в Windows. Когда мы, например, вводим на своём компьютере путь /10.0.0.1/MyShare/MyFolder, то для доступа к запрошенным ресурсам используется протокол SMB. Обязательным условием работы SMB-протокола в сети является наличие на каждом компьютере мини-сервера, который называется SMB Network Resource.

[6]Samba — SMB-сервер для Unix-систем с открытым исходным кодом.

первых SSL VPN-систем в начале 90-х годов прошлого века. Первые такие системы появились в США и конкурировали с огромным количеством аналогичных по функциональности систем, которые уже существовали на рынке. Автор данной статьи участвовала в разработке и сертификации двух американских SSL VPN-систем, которые появились на рынке одними из первых. Одна из этих систем — AEP Netilla [34] — до сих пор продолжает развиваться.

Первыми заказчиками SSL VPN-систем были государственные службы США, в которых существует большое количество стандартов по безопасности (FIPS 140-1 [17], FIPS 140-2 [15] и др.), поэтому одной из важных задач стала проверка соответствия SSL VPN-систем этим стандартам. Именно тогда начали составляться первые матрицы соответствия, где указывались все параметры используемого оборудования. В дальнейшем, после распространения SSL VPN-систем за пределы государственных служб США, процедура сертификации таких систем оформилась как необходимый атрибут этого рынка[7].

Процесс сертификации первой SSL VPN-системы, в разработке которой принимала участие автор статьи, закончился провалом. Центр сертификации предъявил требования и тесты, которым система не могла соответствовать в силу аппаратных и программных ограничений — эти требования не были приняты во внимание при разработке системы. Вторая версия системы разрабатывалась уже с учётом этих требований, но к моменту её выхода сертификационный центр ужесточил свои требования, и второй этап сертификации также закончился неудачей. С другой стороны, у автора и её коллег появилось и стало крепнуть убеждение, что подготовка к сертификации требует системного подхода. Появился первый вариант подхода к подготовке SSL VPN-системы к сертификации, включающий в себя сбор и анализ части требований, влияющих на безопасность корпоративных данных. Не были учтены особен-

[7]Эта история похожа на появление стандарта в области процессов разработки ПО под названием CMMI. Сначала этот стандарт появился в виде требований, которые предъявляли военные и федеральные структуры США к своим подрядчикам по разработке ПО, впоследствии эти требования вместе со стандартом и процедурой сертификации широко распространились за пределами исходной области — как для коммерческих разработок в США, так и в других странах, включая Россию.

ности использования экспортируемых шифров, требования к хранению приватных ключей системы[8], а также не поддерживались европейские стандарты. Однако третья версия продукта получила сертификат, а в компании под руководством автора была создана тестовая лаборатория.

Дальнейшее совершенствование подхода происходило в контексте расширения рынка продаж системы — изначально она продавалась только в США, но было решено распространять её также и в Европе. В силу отличий американских и европейских стандартов технологию подготовки к сертификации SSL VPN-систем доработали и сделали более универсальной. После этого с помощью предложенного подхода было подготовлено к сертификации ещё шесть систем (включающие как версии исходной, так и новые), и все они успешно прошли сертификацию.

Всего под руководством автора за 7 лет была проведена сертификация восьми SSL VPN-систем, которые были проданы более чем тысячи заказчикам. Это было достигнуто благодаря представленному в данной статье подходу.

Описание подхода

Предлагаемый подход является моделью жизненного цикла разработки SSL VPN-системы и включает в себя следующие фазы (рис. 2):

- сбор и анализ требований;
- разработка системы;
- сбор и анализ данных о системе;
- тестирование;
- настройка и модификация;
- сбор и отправка данных в центр сертификации.

Сбор и анализ требований необходимы для того, чтобы все требования (как заказчика, так и сертификационного центра) были идентифицированы и описаны до начала разработки, а также не конкурировали друг с другом. Разработка системы, как правило,

[8] В этой версии не был продуман механизм самоуничтожения приватных ключей в случае несанкционированного проникновения внешних устройств.

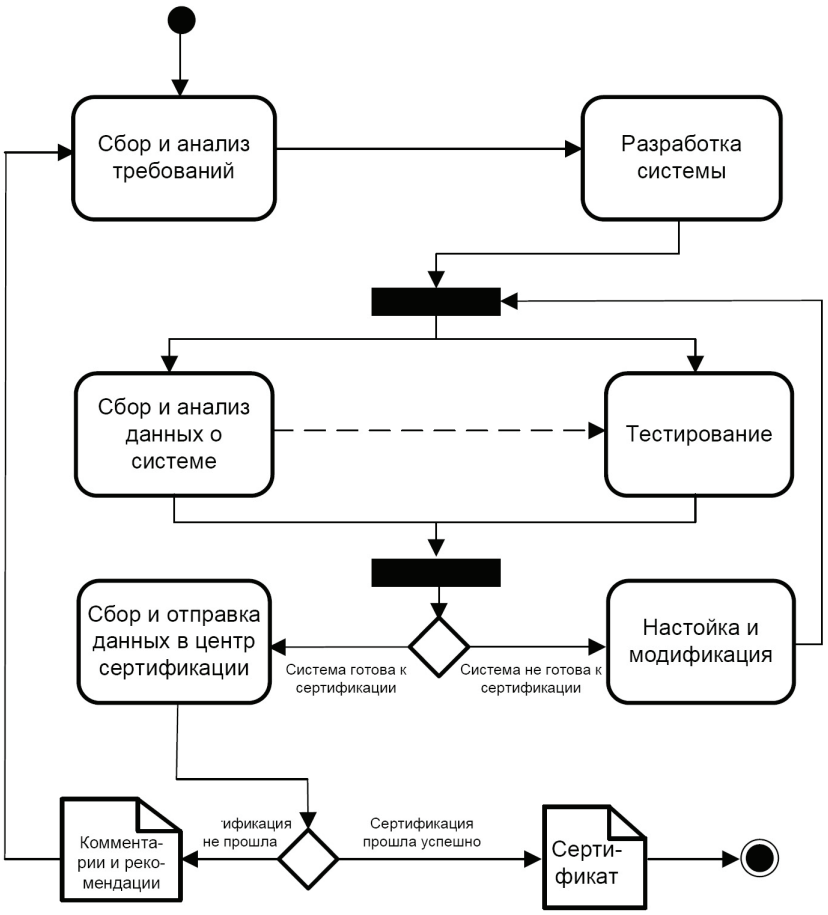

Рис. 2. Жизненный цикл SSL VPN-системы

заключается в сборке аппаратной платформы из готовых компонент и адаптации существующего ПО для этой платформы. После окончания разработки проводится *сбор данных о системе*. Это нужно для того, чтобы понять, насколько разработанная система соответствует требованиям сертификационного центра, а также для последующего *тестирования*. По результатам сбора данных

и тестирования принимается решение о необходимости *настройки и модификации системы* или о её готовности для отправки в сертификационный центр. Настройка и модификация могут проводиться несколько раз до тех пор, пока система, с точки зрения тестировщиков компании, не удовлетворит всем требованиям сертификационного центра. После этого собранные данные оформляются в соответствии с требованиями сертификационного центра (у каждого из сертификационных центров есть свои предустановленные стандарты) и вместе с исполняемыми модулями самой системы отсылаются для проверки. Сертификационный центр проводит свои собственные испытания системы (как правило, такие центры имеют очень мощные тестовые лаборатории) и в случае успешных испытаний выдаёт сертификат, в противном случае предоставляет комментарии и рекомендации. В последнем случае, как правило, оказывается, что необходимо вернуться в начало разработки, ведет к дополнительным затратам денежных и трудовых ресурсов. Чтобы минимизировать время и затраты на разработку системы, необходима специальная технология.

Теперь рассмотрим описанные выше шаги в деталях.

3.1. Сбор и анализ требований

Требования к системе собираются как от заказчика, так и от сертификационного центра. Типичными видами требований к SSL VPN-системам являются требования к уязвимости, отказоустойчивости, быстродействию и пропускной способности системы[9], а также защищённости сетевых каналов и данных, передаваемых по ним, и т. д.

После окончания сбора требований формируется так называемый *документ дизайна системы*, который описывает архитектуру системы, компоненты, из которых она будет состоять, правила соединения компонент, а также как будут взаимодействовать клиенты с сервером (через браузер, через специальные программы и т. д.). Далее создаются дизайн-документы по каждой компоненте системы. Каждый такой документ содержит информацию о предполагаемых решениях, описывает, на каких языках программирования

[9]Пропускная способность — количество одновременно обрабатываемых соединений SSL VPN-системы.

он будет реализован и как будет происходить его взаимодействие с другими компонентами системы. После этого каждой группе разработчиков выделяется своя компонента.

Во время создания дизайн-спецификаций происходит анализ требований, выяснение того, на сколько они реализуемы и не конкурируют ли они — последнее особенно касается требований заказчика и сертификационного центра. Если на последующих этапах выяснится, что требования противоречивы, то это может оказаться проблемой разработчиков — как правило, заказчик очень неохотно меняет требования во время разработки.

3.2. Разработка системы

После того, как были собраны и проанализированы требования, а также создана дизайн-спецификация системы, происходит разработка самой системы.

Сначала разрабатывается каждая компонента системы отдельно на основе дизайн-документов для компонент. Среда разработки может быть произвольной. Обычно, если компонента относится к клиентской части, она разрабатывается под Windows, если к серверной — под UNIX.

Когда все компоненты системы готовы, они собираются в виде исполняемых файлов и проводится настройка взаимодействия системы целиком. Взаимодействие настраивается на основе дизайн-документа системы. В процессе настройки выявляются и устраняются ошибки, допущенные при разработке компонент.

3.3. Сбор и анализ данных о системе

После окончания разработки системы начинается этап сбора данных, ориентированный на поиск потенциальных уязвимостей программной и аппаратной частей системы. Сложность этого этапа заключается в большом количестве данных, которые необходимо собрать и проанализировать.

С точки зрения сбора данных о потенциальной уязвимости, удобно разбить ПО SSL VPN-системы на следующие группы:

- *основное ПО* — программная реализация SSL-протокола;
- *вспомогательные ПО* — сервисы, Интернет-браузеры и т. д.;

- *дополнительные ПО* — операционные системы и др., которые не влияют непосредственно на функционал системы, но могут повлечь её выход из строя.

Сбор данных состоит из следующих этапов.

1. Сбор информации о базовом уровне аппаратной платформы, на которой установлена SSL VPN-система. На этом этапе проверяются единичный отказ отдельных компонент и влияние отказа на работоспособность системы в целом.
2. Сбор информации об аппаратных компонентах системы.
3. Сбор информации об основном ПО системы.
4. Сбор информации о вспомогательном ПО системы.
5. Сбор информации о дополнительном ПО.

Опишем каждый из этих этапов более подробно.

3.3.1. Сбор информации о базовом уровне аппаратной платформы

На этом этапе инженер по тестированию собирает параметры аппаратной части (фактически, это параметры производителей аппаратуры) — электромеханические, тепловые и прочие эксплуатационные характеристики. Далее эти параметры сравниваются с эталонными параметрами аппаратной части SSL VPN-систем (информация от сертификационного центра). Все собранные данные помещаются в матрицу соответствия, которая состоит из двух столбцов «параметры производителя — допустимые параметры».

3.3.2. Сбор информации об аппаратных компонентах системы

На этом этапе инженер по тестированию составляет список всех параметров аппаратных компонент системы и сравнивает их с эталонными, помещает результаты в матрицу соответствия. Далее представлен список собираемых и анализируемых на этом этапе параметров.

1. Параметры BIOS — версия и производитель материнской платы, список выявленных ошибок у данного BIOS, возможность отключения перепрошивки BIOS, параметры безопасности.

2. Параметры сетевых адаптеров SSL VPN-системы — версия и производитель, параметры отказоустойчивости, полученные от производителя, параметры средств аппаратной акселерации[10] сетевых соединений, наличие сертификата на данный функционал.
3. Параметры чипов и платы аппаратной шифрации данных — версия и производитель, параметры отказоустойчивости, полученные от производителя, средства аппаратной акселерации сетевых соединений, наличие сертификата на данный функционал.
4. Параметры устройств хранения данных — наличие RAID-подсистемы[11], тип используемого RAID-тома, версия и производитель прошивки, список выявленных ошибок прошивки данной модели устройства, возможность аппаратной шифрации данных на уровне устройства, возможность самостоятельного принятия устройством решения на полное уничтожение информации при обнаружении им попытки несанкционированного доступа.

3.3.3. Сбор информации об основном ПО системы

На этом этапе производится проверка программной реализации SSL-стека системы. Инженер по тестированию составляет список всех параметров SSL-стека и его возможностей и сравнивает с эталонными, а результаты помещает в матрицу соответствия. Собираются и анализируются следующие параметры.

1. Основные параметры: производитель SSL-стека, наличие сертификатов соответствия у всего стека или его частей (у стека могут быть сертифицированы только некоторые алгоритмы шифрации или алгоритмы обмена ключами).
2. Параметры поддерживаемых шифров. В понятие шифра входят параметры алгоритмов ассиметричной и симметричной криптографии, хэш-функций, поточных/блочных алгоритмов

[10] Средства аппаратной акселерации — аппаратные средства ускорения работы в браузере.

[11] RAID-подсистема (Redundant Array of Inexpensive Disks) — отказоустойчивая система избыточного хранения данных [37].

шифрования и разрядность ключей. Фактически, конкретный шифр представляет собой комбинацию алгоритмов ассиметричной и симметричной криптографии.

3. Параметры поддерживаемых версий протокола.
4. Параметры максимального размера приватного ключа серверной части.
5. Параметры поддерживаемых сертификатов и их расширения для серверной части. Под сертификатом будем понимать совокупность информации о владельце, его публичный ключ и средства, которые позволяют проверить достоверность выдачи данного сертификата.
6. Параметры поддержки клиентских сертификатов — являются важными, так как некоторые компании авторизуют своих сотрудников через клиентские сертификаты.
7. Поддержка возможностей стека по работе с аппаратными хранилищами приватных ключей.
8. Поддержка возможностей стека запрещать использование определённых версий протокола и определённых видов шифров.
9. Параметры известных уязвимостей SSL-протокола.
10. Возможность использования внешнего генератора случайных чисел.
11. Возможности стека протокола восстановить SSL-сессии.

3.3.4. Сбор информации о вспомогательном ПО системы

На этом этапе инженер по тестированию составляет список вспомогательного ПО системы и собирает информацию о его ошибках и уязвимостях. Эта информация собирается, как правило, через сайты производителей ПО. Полученные параметры сравниваются с эталонными и заносятся в матрицу соответствия. В таблице 1 представлен фрагмент типичной матрицы соответствия.

Таблица 1. Фрагмент типичной матрицы соответствия

Компонента и её описание	Ключи компиляции	Модули	Ошибки и узявимости	Порты
Встроенный Web Server Apache 2.x.x, 64-битная архитектура	Скомпилирован со следующими ключами...	В состав входят следующие модули: ssl (версия $x.x.x$), perl (версия $x.x.x$), deflate (версия $x.x.x$)	На данный момен в данной версии официального зарегистрировано X ошибок и Y уязвимостей с кодовыми номерами ...	Используемые порты ...
Интерпретатор Perl 5.xxx, 64-битная архитектура	Скомпилирован со следующими ключами...	Установленные модули ...	На данный момент в данной версии официально зарегистрировано X ошибок и Y уязвимостей с кодовыми номерами...	
Samba 3.x, 64-битная архитектура	Скомпилирован со следующими ключами...	Установленные модули ...	На данный момен в данной версии официально зарегистрировано X ошибок и Y уязвимостей с кодовыми номерами...	Используемые порты ...

3.3.5. Сбор информации о дополнительном ПО

На этом этапе инженер по тестированию составляет весь список параметров дополнительного ПО, влияющий на уязвимость системы, сравнивает эти параметры с эталонными, помещая собранную

информацию в матрицу соответствия. Собирается и анализируется следующая информация.

1. Параметры операционной системы — тип и версия загрузчика, описание известных уязвимостей загрузчика, наличие сертификатов соответствия.
2. Параметры ядра операционной системы — версия, описание известных уязвимостей. Анализ параметров ядра является очень важным, так как оно, как правило, содержит наибольшее количество уязвимостей. Если ядро содержит собственный код, то необходимо также провести его анализ на наличие уязвимостей в организациях, в которых будет в дальнейшем использоваться SSL VPN-система.
3. Параметры файловой системы — её тип, возможности по восстановлению в случае сбоя, существующие ошибки и уязвимости и другие особенности.
4. Известные ошибки и уязвимости TCP/IP стека системы — перечисляются все выявленные на данный момент уязвимости стека и степень их опасности. Также для каждой уязвимости описываются способы её устранения (например, использование каких-либо сторонних программно-аппаратных средств).
5. Параметры библиотеки среды исполнения языка — заносятся все выявленные на данный момент уязвимости библиотеки и компонент, а также указывается степень их опасности. В случае, если по оценке экспертов некоторая уязвимость этого вида является критической, вся система может быть признана ненадёжной.

3.4. Тестирование системы

После того, как были собраны параметры системы, начинается процесс её тестирования. Для этого необходимо, чтобы у компании-разработчика была лаборатория тестирования, способная имитировать работу системы в реальных условиях. Тестирование системы проходит по двум направлениям — применение типовых и специализированных тестов.

Типовые тесты — это тесты, которые можно применить к широкому классу программ, не только к SSL VPN-системам. Типовые

тесты предоставляются сертификационными центрами, разрабатываются инженерами по тестированию на основной функционал системы, поставляются в составе используемых сред разработки, а также могут скачиваться из Интернета.

Специализированные тесты разрабатываются для конкретной SSL VPN-системы и не могут быть применены к широкому классу программ. Они, в основном, разрабатываются компанией, производящей SSL VPN-систему, а также предлагаются сертификационными центрами и доступны на их сайтах.

3.4.1. Применение типовых тестов SSL VPN-систем

Перечислим основные типовые тесты для SSL VPN-систем.

1. Тесты на максимальную нагрузку процессора, которые обеспечивают полную «нагрузку» всех его ядер в течение длительного времени и проверяют, не привело ли это к каким-либо отрицательным последствиям на клиентской и серверной сторонах.
2. Тесты на максимальную нагрузку сетевых карт, которые обеспечивают полную загрузку сетевого канала таким образом, чтобы он использовал всю свою полосу пропускания.
3. Тесты с использованием больших и сверхбольших объёмов данных (терабайты), которые обеспечивают загрузку с помощью SSL VPN-системы файлов/данных/web.
4. Тесты для проверки специфических Samba-сервера, SOCKS-сервера[12] и т. д. В рамках тестирования каждый сервис проверяется своими типовыми тестами.
5. Тесты на переполнение журнала. Многие системы ведут журнал, где хранится информация об ошибках, доступе к ресурсам и т. д. Данный вид тестов направлен на то, чтобы «замусорить» журнал большим количеством информации. Переполнение журнала не должно влиять на работоспособность системы.
6. Тесты для проверки Web-сервера — проверка на производительность и отказоустойчивость.

[12]SOCKS (SOCKet Secure) — сетевой протокол, который позволяет клиент-серверным приложениям прозрачно использовать сервисы за межсетевыми экранами (файерволами).

7. Тестирование SSL VPN-системы при повреждении её внутренней программной целостности (например, в случае сбоя Web-сервера). В этом случае имитируются сбои внутренних программ SSL VPN-системы, и оценивается общая реакция системы. Ожидается, что система после сбоя восстановит нормальное рабочее состояние.
8. Тесты компонент на XSS-уязвимости[13], которые обеспечивают искусственные XSS-атаки с целью доступа к внутренней информации системы или введения пользователя в заблуждение относительно его реального нахождения в системе.
9. Тесты для проверки компонент портала, которые обеспечивают искусственные SQL-атаки[14] с целью доступа к внутренней базе данных системы.
10. Тесты на переполнение оперативной памяти SSL VPN-системы.

3.4.2. Использование специализированных тестов SSL VPN-систем

Основная задача применения специализированных тестов — тестирование производительности системы, т. е. проверка выполнения ее специфических функций при больших нагрузках и сбоях. Нагрузочное тестирование и имитацию сбоев системы целесообразно производить по направлениям, перечисленным далее.

1. Тестирование базового уровня аппаратной платформы SSL VPN-системы — имитируем резкий перепад таких параметров, как температура, влажность, напряжения в сети и т. д., а также проверяются некорректные значения подобных параметров.

[13]XSS-уязвимости (Cross Site Scripting) — тип уязвимости интерактивных информационных Интернет-систем, когда в генерируемые сервером страницы можно внедрить клиентские скрипты. Специфика подобных атак на сервер заключается в том, что для этого обычно используется авторизованный на данном сервере клиент.

[14]SQL-атака — один из распространённых способов взлома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода.

2. Тестирование аппаратного уровня каждой из компонент, в частности, имитация максимальной загрузки входящих в систему устройств, а также сбоя отдельных компонент. С этой целью для каждой компоненты создаются специальные скрипты.
3. Проверка вспомогательного ПО системы на известные уязвимости и ошибки. Чаще всего соответствующие специализированные тесты не доступны широкой аудитории, а приобретаются у компаний, специализирующихся на информационной безопасности.
4. Тестирование основного ПО системы — SSL-стека. Тесты для SSL-стека разрабатываются компанией самостоятельно или приобретаются у сторонних фирм. Как правило, задача этих тестов — проверить известные уязвимости SSL-протокола на используемой в системе реализации.
5. Для дополнительного ПО обычно не проводится специализированного тестирования.

3.5. Настройка и модификация системы

Модификация и настройка системы необходимы, когда в процессе тестирования было выявлено, что она не соответствует требованиям клиента или сертификационного центра. Система может быть либо аппаратно модифицирована, либо программно настроена. В отношении аппаратной части у разработчиков системы часто имеется небольшой диапазон для варьирования компонент (обычно можно лишь заменить одну аппаратную компоненту на другую, аналогичную в своей семействе). Кроме того, часто аппаратная платформа является уже законченным решением и не подлежит серьезным изменениям. Поэтому разработчики стремятся настроить и модифицировать систему через изменение её программной части. Это может быть выполнено либо путём оптимизации ПО (перенастройка конфигурационных параметров различных программных компонент), либо переписыванием отдельных программных компонент или всей системы целиком.

Первый способ является более простым и его стараются применять в первую очередь. Однако настройка параметров не всегда позволяет решить проблему, например, устранить неприемлемую

для клиента уязвимость системы. Это приводит к тому, что необходимо перестраивать связи между модулями, добавлять или модифицировать отдельные модули и функции системы.

После того, как система модифицирована и настроена, для неё снова необходимо вновь собрать параметры и сравнить их с эталонными, а также выполнить тестирование.

3.6. Сбор и отправка данных в сертификационный центр

Когда все требования к системе тщательно проанализированы и оттестированы, собрана и проверена вся информация об уязвимостях, формируется пакет документов, и система вместе с документами отсылается в сертификационный центр. Требования к формированию этого пакета документов предъявляет сертификационный центр, и их также нужно соблюсти. Обычно это совокупность файлов исходного кода, документы, созданные на основе проведенных тестов, а также собранные (в виде таблиц) параметры.

Как правило, сертификационный центр может работать через Web-интерфейс. В этом случае все данные закачиваются через Интернет, а разработчики системы предоставляют доступ к SSL VPN-системе сертификационному центру, указывая адрес системы в виде ссылки.

Сертификационный центр проводит анализ присланных документов, а также сам тестирует систему и сравнивает полученные результаты с присланными. Если находятся разночтения и/или требования сертификационного центра не соблюдены, он формирует список замечаний, где указываются, какие найдены ошибки и противоречия. Причины ошибок приводятся, как правило, в общем виде, так что сертификационный центр не может использоваться для тестирования системы (кроме того, процедура сертификации дорогая). Пример описания ошибки, выданной сертификационным центром, представлен в таблице 2.

Таблица 2. Пример описания ошибки

Test	Status	Reason
Название теста	Failed	Система потеряла данные при нагрузочном тестировании.

Когда все тесты сертификационного центра прошли успешно, центр выдает сертификаты на каждую компоненту системы, а также на всю SSL VPN-систему.

Заключение

Разработка SSL VPN-систем тесно связано с их сертификацией, так как именно сертификаты системы могут гарантировать безопасность данных пользователя и устойчивость системы. Процедура сертификации является сложной и комплексной, она должна учитывать стандарты страны, в которой будет использоваться система, а также особенности задач клиентов, на которых она ориентирована. При этом надо учитывать, что любые изменения аппаратной и программной частей могут повлечь за собой новые, неизвестные ранее уязвимости системы. Поэтому сертифицируются те конфигурации системы, которые будут продаваться реальным заказчикам.

В статье представлен комплексный подход к процедуре сертификации, который может быть применён при разработке SSL VPN-систем с различной архитектурой. Подход позволяет проанализировать всю совокупность параметров, влияющих на сертификацию системы, провести необходимое для сертификации тестирование и пройти процесс сертификации за минимальное количество итераций.

Список литературы

[1] *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. Основание информатики, М.: МЦНМО. 1998. 703 с.

[2] *Иванов М. А.* Криптографические методы защиты информации в компьютерных системах и сетях, М.: КУДИЦ-ОБРАЗ. 2001. 368 с.

[3] *Кадер М.* Особенности российского рынка VPN // Средства защиты информации и бизнеса. Агентство CNews Analytics (CNA). 2006. http://www.cnews.ru/reviews/free/security2006/articles/vpnmarket

[4] *Кияев В. И.* О терминалогии и требованиях международных стандартов качества // Системное программирование. СПб: СПбГУ. 2005. Т. 1. № 0. С. 311–334.

[5] *Кияев В. И., Кищенко Д. М., Окомин И. С.* Опыт усовершенствования и стандартизации процесса по цифровым электронным станциям // Системное программирование. СПб.: СПбГУ. 2006. Т. 2. № 1. С. 219–239.

[6] *Кульгин М.* Технологии корпоративных сетей. Энциклопедия, Питер. 2000. 704 с.

[7] *Лавин Д.* VPN и IPSec на пальцах // Сетевые решения A-Z. Изд-во «Нестор». 2005. http://www.nestor.minsk.by/sr/2005/03/050315.html

[8] *Лукин В.* Первый кирпич в стене VPN. Обзор устройств VPN начального уровня // Сети и серверы. 2002. http://www.ixbt.com/comm/vpn1.shtml

[9] *Маслов Е.* Перспективы VPN SSL-систем // SSL технологии. Безопасность в сети. 2006. http://www.inssl.com/ssl-vpn-perspectives.html

[10] *Норман Р.* Выбираем протокол VPN // Microsoft Windows для профессионалов. Изд-во «Открытые системы». 2001. http://www.osp.ru/win2000/2001/07/175027/

[11] Обзор оборудования VPN // Сетевая безопасность. 2012. http://www.networkaccess.ru/articles/security/vpn_hardware

[12] *Олифер В. Г., Олифер Н. А.* Компьютерные сети. Принципы, технологии, протоколы: учебник для ВУЗов. СПб.: Питер. 2001. 672 с.

[13] *Романец Ю. В., Тимофеев П. А., Шаньгин В. Ф.* Защита информации в компьютерных системах и сетях, М: Радио и связь. 2002. 328 с.

[14] *Снайдер Д.* VPN: поделенный рынок // Коммуникации для бизнеса. Изд-во «Открытые системы». 1999. http://www.osp.ru/nets/1999/11/144377/

[15] Федеральная служба по техническому и экспортному контролю. Россия. http://www.fstec.ru

[16] *Столлингс В.* Основы защиты сетей. Приложения и стандарты, М.: Вильямс. 2002. 432 с.

[17] *Фёдоров А. Р.* Способ кодирования информации для построения программных отказоустойчивых дисковых массивов // Системное программирование. 2012. Т. 7. № 1. С.

[18] *Ященко В. В.* Введение в криптографию, СПб.: Питер. 2007. 288 с.

[19] AEP Netilla Secure Remote Access SSL VPN. http://www.dshi.com//nssolutions//netilla

[20] *Bruce S.* Applied Cryptography, ISBN 9780471128458. 1995. 784 p.

[21] *Cameron R. J.(r).* Networks Secure Access SSL VPN Configuration Guide. 2010. 656 p.

[22] *Gleeson B., Heinanen J., Lin A.* A Framework for IP Based Virtual Private Networks // RFC. The Internet Engineering Task Force.2000. http://www.ietf.org/rfc/rfc2764.txt

[23] International Center for Quality Certification. http://www.icqc.eu

[24] *Lynn G. M.* Vulnerability Assessment of Physical Protection System, London: Butterworth-Heinemann. 2005. 400 p.

[25] *Snyder J., Elliott C.* Pure Hardware VPNs Rule High-Availability Tests // Reviews 2000. Network World. http://www.networkworld.com/reviews/2000/1211rev.html

[26] Federal Information Processing Standards Publications. US. 2008. http://www.itl.nist.gov/fipspubs

[27] FIPS PUB 140-1. Security Requirements for Cryptographic modules, U.S. Department of Commerce / National Institute of Standards and Technology. http://csrc.nist.gov/publications/fips/fips1401.pdf

[28] FIPS PUB 140-2. Security Requirements for Cryptographic modules, U.S. Department of Commerce / National Institute of Standards and Technology. http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf