СОЗДАНИЕ БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА БАЗЕ ПЛАТФОРМЫ .NET

В.В. Михаленок

Одним из главных компонентов современных Web-ориентированных платформ являются механизмы обеспечения безопасности. Под защищенным программным обеспечением мы, вслед за М. Ховардом, будем понимать программное обеспечение, которое обеспечивает конфиденциальность, целостность и доступность информации клиента программы, а также целостность и доступность вычислительных ресурсов, управляемых владельцем системы. Следует заметить, что вопросам безопасности при обучении студентов уделялось небольшое внимание. В частности, В. Липаев отмечает, что опыт создания сложных комплексов программ высокого качества, который наблюдался в 60-80-е гг. на предприятиях оборонной промышленности, утрачен и что «современные молодые специалисты должны вновь, почти «с нуля», создавать и осваивать проектирование крупномасштабных программных средств с требуемым качеством». Он считает, что почти все вузы не готовят специалистов по комплексированию и обеспечению качества крупных проектов программных систем. В результате многие планы на создание сложных программных средств подготавливаются и оцениваются небрежно. Значительные ошибки в определении требуемых показателей качества, в оценке трудоемкости, стоимости и длительности создания программных средств являются достаточно массовыми и типичными. Стандарты ИСО серии 9000 определяют обеспечение качества как «совокупность планируемых и систематически проводимых мероприятий, необходимых для уверенности в том, что продукция или процессы удовлетворяют определенным требованиям качеству». Системой обеспечения качества будем считать совокупность методов и средств организации управляющих и исполнительных подразделений предприятия, участвующих в проектировании,

разработке и сопровождении комплексов программ с целью придания им свойств, обеспечивающих удовлетворение определенных потребностей заказчиков и потребителей при минимальном или допустимом расходовании ресурсов. В. Липаев считает, что «требуемое качество при разработке проектов программных средств можно обеспечить двумя методами:

- путем использования только заключительного контроля и испытаний готовых объектов и исключения из поставки или направлением на доработку продуктов, не соответствующих требуемому качеству;
- посредством применения регламентированных технологий и систем обеспечения качества проектирования и разработки, предотвращающих дефекты и гарантирующих высокое качество продукции во время ее создания и модификации».

Именно внедрение технологий и систем обеспечения качества является по мнению российских [1, 2] и зарубежных [3, 4] исследователей перспективным способом разработки защищенного программного обеспечения. Важно отметить, что обучение по обеспечению качества разрабатываемого программного обеспечения проводилось при его тестировании. Высшие школы выработали различные и многоуровневые системы тестов, что может и должно использоваться при создании курсов по разработке программного обеспечения. Дисциплина «методы и средства защиты компьютерной информации», которая преподается для специалистов в области информатики и вычислительной техники включает в себя следующие разделы:

- определение основных понятий безопасности систем;
- создание политики безопасности;
- описание и использование криптографических моделей и алгоритмов безопасности;
- использование моделей безопасности основных операционных систем, а также основы администрирования сетей и алгоритмы аутентификации пользователей;

• многоуровневая защита корпоративных сетей и защита информации в сетях.

Безусловно, навыки и умения, получаемые студентами после успешного изучения данной дисциплины, являются крайне необходимыми в настоящее время. Вместе с тем, необходимо отметить, что в данной дисциплине уделено недостаточно внимания созданию безопасного кода. Стоит подчеркнуть, что безопасное обеспечение является подмножеством качественного программного обеспечения. Создание качественного программного обеспечения включает в себя комплекс методов и стандартов, которые непосредственно обеспечивают эффективный жизненный цикл сложных высококачественных программных средств и баз данных. По мнению В. Липаева, для обеспечения высокого качества, надежности функционирования и безопасности применения сложных комплексов программ необходимо отдельно выделять специалистов, ответственных за соблюдение технологии создания и развития программ, за обеспечение и контроль качества. Следовательно, по нашему мнению, необходимы специальные дисциплины для подготовки таких специалистов. Целью же курса «создание и использование программных средств на базе современных Web-ориентированных платформ» является подготовка специалистов, создающих безопасный код.

При проектировании безопасного программного обеспечения, использующего возможности Web-ориентированных платформ (это означает, что разрабатываемое программное обеспечение будет работать в небезопасной среде Интернет), студент должен знать методики классификации опасностей, грозящих разрабатываемой системе, методики оценки риска, а также основные методы защиты.

Одной из популярных методик классификации опасностей, которая применяется при разработке программного обеспечения во многих компаниях-производителях программных средств, является методика STRIDE [3] (в соответствии с первыми буквами английских названий категорий). Она состоит из следующих классификаций основных опасностей:

• Подмена сетевых адресов (Spoofing identity). Атаки подобного типа позволяют взломщику выдавать себя за другого пользователя или подменять настоящий сервер подложным.

• Модификация данных (Tampering with data). Атаки этого типа предусматривают злонамеренную порчу данных. Несанкционированное изменение постоянных данных или информации, пересылаемой через открытую сеть являются примерами данной категории.

• Отказ от авторства (Repudiation). Данные уязвимости возникают в системах, в которых не ведется аудит выполнения важных операций. В таких ситуациях неавторизованный пользователь может удалять важные данные, а администраторы систем не смогут идентифицировать злоумышленника.

• Разглашение информации (Information disclosure). Подразумевается раскрытие информации лицам, доступ к которой им запрещен.

• Отказ в обслуживании (Denial of service). В атаках этого типа взломщик пытается лишить доступа к сервису правомочных пользователей, сделав программное средство временно недоступным к использованию. В настоящее время является самым распространенным методом.

• Повышение привилегий (Elevation of privilege). Эта классификация относится к случаям, когда непривилегированный пользователь получает привилегированный доступ и становится частью защищенной системы.

Методика DREAD [2] (по первым буквам английских названий категорий) позволяет оценивать и распределять опасности по мере убывания их серьезностей. Она состоит из следующих категорий:

• Потенциальный ущерб (Damage potential) – мера реального ущерба от успешной атаки. Наивысшая степень (10) опасности означает практически беспрепятственный взлом средств защиты и выполнение практически любых операций.

• Воспроизводимость (Reproducibility) – мера возможности реализации опасности. Бреши, доступные постоянно получают наивысшую оценку.

• Подверженность взлому (Exploitability) – мера усилий и квалификации, необходимых для атаки. Если успешная атака на программную систему возможна неопытным программистом, то данные бреши получают наивысшую оценку (10).

• Круг пользователей, попадающих под удар (Affected users) – доля пользователей, работа которых нарушается из-за успешной атаки.

• Вероятность обнаружения (Discoverability) – зависит от опыта и квалификации тестировщиков разрабатываемой системы. Данная оценка является самой субъективной.

Используя описанные выше методики, можно следить за качеством безопасности создаваемых объектов или компонентов разрабатываемого программного обеспечения.

По заверениям разработчиков Web-ориентированной платформы .Net, она была спроектирована и реализована с учетом требований безопасности. «Безопасность – важнейший компонент создаваемых программ на платформе .Net» [4]. Опишем основные механизмы обеспечения безопасности, доступные разработчикам и пользователям платформы .Net.

Аутентификация. М. Ховард определяет процесс аутентификации как «процесс, в котором один объект, или участник безопасности (principal), проверяет подлинность другого объекта, т.е. устанавливает, действительно ли он тот, за кого себя выдает. Участники безопасности – это пользователи, исполняемый код или компьютер. Аутентификация требует доказательств в виде реквизитов, которые могут принимать различные формы». В настоящее время разработчикам и пользователям программных средств на платформе .Net, работающим под управлением операционных систем Windows 2000 (и далее) доступны следующие методы аутентификации:

• базовая аутентификация;

• стандартная аутентификация Windows;

• аутентификация по протоколу NTLM (NT LAN Manager);

• аутентификация на основе хэша.

Отдельно стоит упомянуть недавно появившееся методы аутентификации:

• Аутентификация на основе форм. Данным методом аутентификации могут пользоваться разработчики приложений ASP .NET. Причем у разных разработчиков реализация может быть разной. Реквизиты пользователей могут хранится в базах данных, XML файлах.

• Аутентификация Microsoft Passport. Passport-аутентификация – это централизованная схема аутентификации. Ее основное преимущество заключается в том, что для входа в Passport-службу, при переходе на другой Web-сервис, использующий Passport, не нужно заводить заново реквизиты. Поддержка данной технологии в ASP .NET реализована в классе *PassportAuthenticationModule*.

• Аутентификация по протоколу Kerberos v5. В операционных системах Windows 2000 и более поздних Kerberos применяется при развертывании службы каталогов Active Directory. Одно из важнейших преимуществ Kerberos – взаимная аутентификация, т.е. возможна проверка в обоих направлениях: от клиента к серверу и наоборот. Kerberos считается более надежным и более быстрым, чем NTLM.

• Аутентификация на основе сертификатов X.509. Данная аутентификация используется в протоколах SSL/TLS. Кроме того, существуют реализации клиентских сертификатов для работы со смарт-картами.

• Использование протокола IPSec (Internet Protocol Security). От рассмотренных выше механизмов аутентификации он отличается только тем, что предусматривает только аутентификацию серверов. Кроме того, он поддерживает целостность и конфиденциальность данных.

• RADIUS. Данная служба используется для аутентификации удаленных пользователей.

Авторизация. Под авторизацией будем понимать проверку, в процессе которой выясняется круг доступных аутентифицированному участнику ресурсов и предоставляется доступ к ним. На платформе .Net доступны следующие механизмы авторизации:

• Списки управления доступом (Access Control lists, ACL). ACL – это набор записей управления доступом, каждая из которых определяет, какие действия по отношению к ресурсу доступны участнику безопасности.

• Привилегии. Привилегия – это право, предоставляемое пользователю, действующее в масштабах всей системы.

• IP-ограничения относятся к особенностям реализации Web-серверов, способных ограничивать доступ к информации на основе IP адресов клиентов.

• Серверные разрешения. Данный вид авторизации применяется во многих серверных программных продуктах, способных вести списки и разрешения для пользователей этих продуктов.

Кроме того, разработчик может применять следующие технологии защиты:

• Использование протоколов SSL/TLS при передачи данных по сети.

• Использование IPSec.

• Использование механизмов безопасности DCOM и RPC.

• Использование возможностей шифрующей файловой системы (EFS) Windows.

• Использование CryptoAPI.

• Безопасность типов платформы .Net. Данные проверки выполняются автоматически при разработке программ. При создании классических настольных или Web-приложений на языках C/C++ данные проверки не выполняются автоматически и программист должен сам заботиться о безопасных приведениях типов.

• Цифровая подпись программ. На платформе .Net доступны симметричные и асимметричные алгоритмы шифрования и возможность подписывать созданные проекты в целях предотвращения нарушения целостности данных.

• Безопасность доступа кода. Используя данную технологию, разработчик может указать привилегии, необходимые для выполнения кода, и выполняемый код не получит большие привилегии, чем ему требуется. Кроме того,

администратор системы может запретить доступ определенного кода к важным ресурсам системы и код не сможет получить к ним доступ.

• Безопасность на основе ролей пользователя. Разработчики могут указывать, каким пользователям разрешен доступ к защищаемым ресурсам, на основе групп безопасности, к которым принадлежат данные пользователи. Данные решения могут основываться на встроенные в систему политики безопасности, или на основе правил, разработанных создателем программных средств.

• Изолированное хранилище. Изолированным хранилищем называется специально отведенное место для пользователя, в котором он может безопасно хранить свои данные и выполнять код.

Очевидно, что используя возможности Web-ориентированной платформы .Net, разработчикам и пользователям доступны самые современные механизмы обеспечения безопасности разрабатываемых и используемых программных систем.

Литература

1. Липаев В.В. Обеспечение качества программных средств. М.: Синтег, 2001.

2. Орлов С.А. Технологии разработки программного обеспечения. Спб.: Питер, 2002.

3. Ховард М., Лебланк Д. Защищенный код. Пер. с англ. М.: Издательско-торговый дом «Русская Редакция», 2003.

4. Материалы конференции «Обеспечение информационной безопасности. Cisco Systems». М.: 2004.