Безопасность Java

Презентация по курсовой работе

Безбородый И.Е. гр.3881

Платформа Java

Платформа Java была разработана с упором на безопасность. По своей сути сам язык Java является строго типизированным и предоставляет автоматическую сборку мусора и повышенную надежность кода приложения. Механизм загрузки и проверки безопасности класса гарантирует выполнение только надежного Java-кода.

Инструменты

Сегодня архитектура включает в себя набор API, инструменты и реализации протоколов, механизмов и алгоритмы безопасности наиболее часто используемые в приложениях. Это обеспечивает разработчику средства создания безопасных приложений, а также предоставляет пользователю или администратору набор инструментов для безопасного управления приложениями.

Простота

Язык Java строго типизирован и прост в использовании. Он обеспечивает автоматическое управление памятью, сбор мусора и проверки диапазона на массивах. Это уменьшает общие нагрузки на разработчиков, позволяет совершать меньше тонких ошибок программирования и писать более безопасный и надежный код.

Уровни доступа

Язык определяет четыре различных уровня доступа: частный, защищенный, публичный и пакетный. Наиболее открытый спецификатор доступа — публичный. Закрытый – частный. Защищенный модификатор позволяет получить доступ к любому подклассу, или другому классу в пределах того же пакета. Доступ на уровне пакета позволяет получить доступ к классам в рамках только того же пакета.

Байт-код

Компилятор преобразует Java программы в представление машинно-независимый байт-код. Верификатор байт-кода гарантирует, что только доверенный байт-код выполняются в среде выполнения Java. Он проверяет соответствие байт-кода спецификации языка Java и не нарушение ими правил языка Java или ограничений имен.

API безопасности

Построены на основе следующих принципов:

- Независимость
- Взаимодействие
- Расширяемость

API криптографии

- Алгоритмы дайджест сообщений
- Алгоритмы цифровой подписи
- Симметричное шифрование
- Потоковое симметричное шифрование
- Асимметричное шифрование
- Шифрование на основе паролей (ПБО)
- Криптография эллиптических кривых (ECC)
- Алгоритмы согласования ключей
- Генераторы ключей
- Коды проверки подлинности сообщений (Mac)
- (Псевдо-) генераторы случайных чисел

PKI

Инфраструктура открытых ключей (PKI) позволяет безопасно обмениваться информацией на основе криптографии открытых ключей. Включает в себя ключи, сертификаты, шифрование с открытым ключом и доверенные центры сертификации (ЦС), создает и подписывает сертификаты.

Проверка подлинности

Проверка подлинности — это процесс установления личности пользователя. В контексте среды выполнения Java это процесс идентификации пользователя исполнителем Java-программы.

Платформа Java предоставляет интерфейсы API, позволяющие приложению выполнять проверку подлинности пользователя через подключаемые логин-модули.

Логин-модули

Java-платформа обеспечивает следующие встроенные LoginModules:

- *Krb5LoginModule* для проверки подлинности с использованием протокола Kerberos
- *JndiLoginModule* для проверки подлинности имени пользователя и пароля с использованием баз данных LDAP или NIS
- *KeyStoreLoginModule* для входа в хранилище ключей любой тип, в том числе в хранилище ключей маркера PKCS # 11

Безопасная связь

Прошедшие через сеть данные могут быть получены кем-то, кто не должен был их получать. Необходимо принимать меры, чтобы сделать данные непонятными для сторон, получивших их несанкционированно. Важно также гарантировать, что данные не были изменены умышленно или неумышленно во время перевозки.

Протоколы

- SSL/TLS
- SASL
- GSS-API и Kerberos

Управление доступом

Архитектура управления доступом в платформе Java защищает доступ к конфиденциальным ресурсам или коду чувствительных приложений (например, к методам в классе). Контроль доступа происходит при посредстве менеджера безопасности, представленного классом *java.lang.SecurityManager*

Разрешения

Когда код загружается в загрузчик классов среды выполнения, загрузчик класса автоматически связывает с этим кодом следующую информацию:

- Откуда загружен код
- Кем подписан
- Разрешения по умолчанию

Политика

Платформа Java инкапсулирует понятие политики безопасности в классе *java.security.Policy*. Существует только один объект политики, установленный в среде выполнения Java в любой момент времени. Основная ответственность объекта политики - определить разрешен ли доступ к защищенному ресурсу. Как именно объект политики делает это, зависит от реализации.

Контроль доступа выполнения

Среда выполнения Java отслеживает количество последовательных вызовов, которые производятся в процессе выполнения. Когда запрашивается доступ к защищенному ресурсу, стек вызовов оценивается на разрешение запрашиваемого доступа.

Список литературы

- http://docs.oracle.com/javase/6/docs/technotes/guides/security/overview/jsoverview.html
- http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136007.html