ОРГАНИЗАЦИЯ ЗАЩИТЫ МОДУЛЕЙ ИНФОРМАЦИОННЫХ СИСТЕМ АКТИВИЗИРУЕМЫХ СРЕДСТВАМИ СЕРВИС-БРАУЗЕРА В СРЕДЕ LINUX

Курако Е.А., Орлов В.Л.
Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия
kea@ipu.ru, ovl@ipu.ru

Аннотация. Проводится сравнение использования браузеров различного типа в качестве клиентов информационных систем. Рассматриваются способы защиты информационных систем, разрабатываемых как в среде Windows, так и в среде Linux и использующих сервис-браузер для активизации модулей. Определяется возможность перехода к среде Linux.

Ключевые слова: информационная система, браузер, web-браузер, сервис-браузер защита информации, Linux, модули.

Введение

Вопрос организации защиты информации при построении распределённых информационных систем (ИС) является очень важным. В частности, это обусловлено тем обстоятельством, что в качестве клиента таких систем обычно применяется браузер [1]. Обычный браузер используется для организации отображения html – страниц, которые формируются серверной частью. При этом процесс отображения может быть достаточно сложным и содержать интерпретацию программных фрагментов, написанных, как правило, на языке JavaScript, а также включать вызов сервисов. То есть основная задача сервера информационной системы состоит в подготовке страниц, обрабатываемых клиентом-браузером.

Это решение безусловно имеет преимущества, так как все клиенты таких систем фактически одинаковы, что упрощает установку и настройку системы.

Но возникает несколько вопросов, касающихся безопасности:

1. Как правило, браузерные программы являются весьма сложными и содержат много возможностей, которые не обязательно используются в информационных системах. Например,
   - обеспечивают поддержку различных расширений, которые могут включать вредоносные программы;
   - дают возможность загружать различные файлы, иногда нежелательные;
   - обеспечивают переход (часто бесконтрольный) по различным ссылкам;
   - предусматривают хранение различной информации, например, истории посещений;
   - проводят подключение рекламных блоков и, в то же время, блокировку рекламы по своим алгоритмам;
   - обеспечивают общение в социальных сетях;
   - подключают другие возможности.
2. Объем и сложность браузерной программы затрудняет ее контроль.
3. Обычно при разработке информационных систем, ориентированных на применение браузеров-клиентов, используются несколько языков. Это определяется технологией программирования, включающей направления frontend для взаимодействия с внешними пользователями и backend – для работы с серверной частью. Исторически это задавалось тем, что программная работа с клиентами обеспечивалась языком-интерпретатором JavaScript, а работа с сервером - множеством других языков. Серверная часть предусматривает успешное взаимодействие с базами данных, хранилищами разного типа и обеспечение возможных спектров вычислений. Вместе с тем, при проектировании на разных языках возникают проблемы, связанные с организацией корректной стыковки фрагментов, единым подходом к вопросам защиты информации и, наконец, квалификацией программистов.

Очевидно, решение поставленных проблем может быть обеспечено не только на основе использования универсальных широко распространенных браузеров, но и на основе создания специализированного браузера, который ориентирован на функционирование в составе информационной системы, компактен, не содержит избыточных функций, и дает возможность проводить проектирование как клиентской, так и серверной части на одном языке.

Таким специализированным браузером может быть сервис-браузер, разработанный в Институте проблем управления РАН [2-4].

1. Особенности сервис-браузера, предназначенного для использования в информационных системах

В дальнейшем обычные широко используемые браузеры будем для краткости называть web-браузерами. Если в качестве основного клиента предполагается использование не web-браузера, а сервис-браузера [3-5], то нужно отдавать себе отчет, что этот браузер должен быть компактен, не содержать фактически неиспользуемых фрагментов и в то же время включать дополнительные возможности, отсутствующие в web-браузерах, но необходимые именно для информационных систем. Важно, чтобы сервис-браузер был универсальным, то есть при установке на каждой машине-клиенте размещалось бы одно и то же базовое программное обеспечение. Впрочем, это характерно для всех браузеров.

Рассмотрим начальный этап работы web-браузера. Web-браузер обращается по IP-адресу или DNS-адресу с использованием http-запроса, обычно включающего методы POST или GET, и в ответ получает первую html-страницу, которую отображает на экране. В состав html-страницы обычно включается собственно текст HTML, таблицы стилей CSS и также при необходимости интерпретируемая программа, сформированная на языке JavaScript. Если мы работаем не просто с сайтом, а с порталом, который представляет собой фактически аналог информационной системы, то обычно на первой странице присутствует оглавление, содержащее ссылки, позволяющие переходить в различные разделы сайта (рис.1).

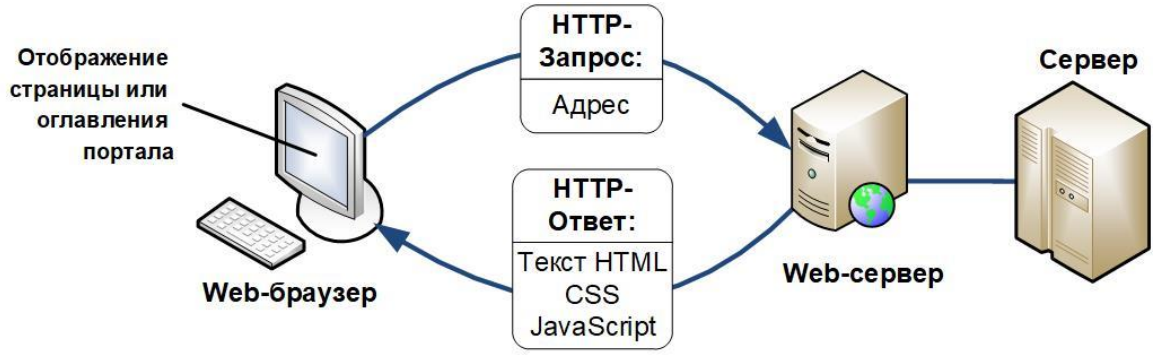

Рис. 1. Начальный этап работы web-браузера

Сервис-браузер начинает свою работу также, как и web-браузер. Он использует IP-адрес или DNS-адрес как адрес сервера. Эти адреса обычно задаются в настройках. Вместе с тем обращение клиента к серверу происходит путем вызова сервиса. Собственно, отсюда и происходит и название сервис-браузера. То есть сервис-браузер реализует метод общения с сервером путем вызовов сервисов и получения ответов на них (рис. 2). Причем ответы могут быть разнообразными, а не просто представлять собой html-страницы.

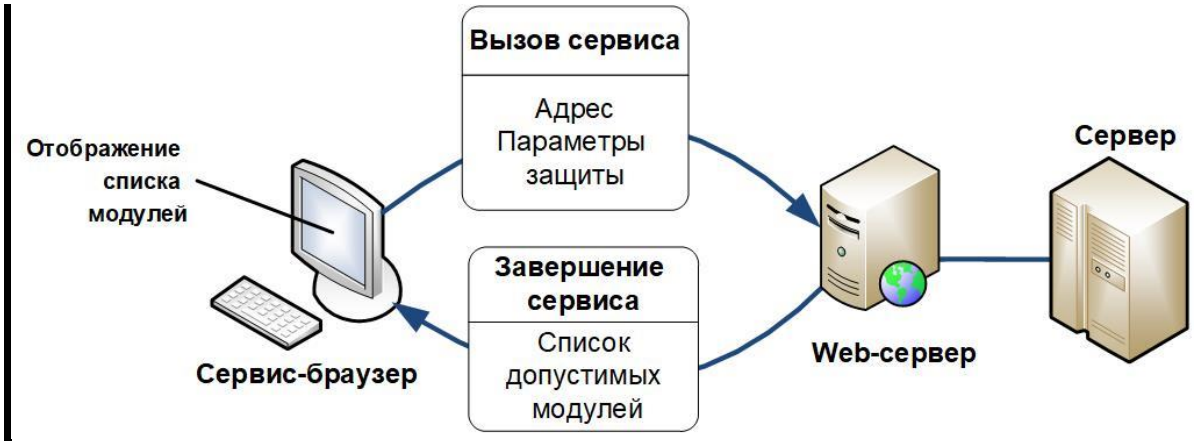

Рис. 2. Начальный этап работы сервис-браузера

Кто же обрабатывает эти разнообразные ответы на стороне клиента? С web-браузером все просто. Он получает html-страницу, и он же ее интерпретирует. Хотя нужно иметь ввиду, что интерпретация в данном случае - вопрос далеко не простой.

Сервис-браузер по определению более компактная программа. Поэтому он не включает в свой состав способы интерпретации, а считывает заранее с сервера программу, которая предназначена для работы с полученным ответом. Первично такой программой является сам сервис-браузер. Отметим, что его функции ограничены. Основное предназначение – аутентифицировать пользователя и определить, какие ресурсы с ним связаны. После определения пользователя с сервера обычно считывается перечень связанных с ним «модулей обработки», которые могут подкачиваться и обновляться с сервера. Собственно, этим и определяется универсальность сервис-браузера. Заметим, что модули обработки могут быть разными для различных информационных систем, но могут иметь в своем составе модули, которые используются в других системах. Пока для простоты рассматриваем вариант, когда сервис-браузер обслуживает одну информационную систему. Поэтому каждый пользователь имеет счетное число модулей обработки. А значит, они могут быть загружены с сервера заранее. И более того – обновлены, если произошло какое-либо изменение (рис.3).

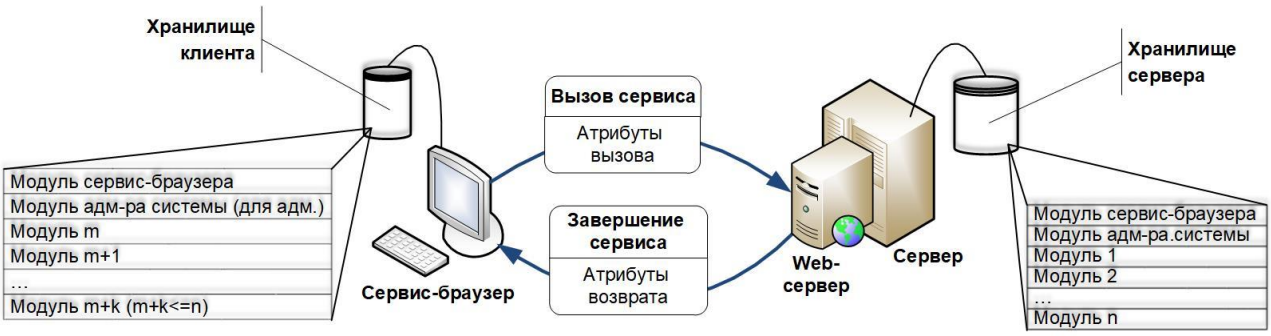

Рис. 3. Миграция модулей из хранилища сервера в хранилище клиента

Это похоже на динамическое управление страницами в web-браузере. Но вместе с тем организуется принципиально просто. Достаточно привязать процедуру обновления к какому-либо событию, например, к перезапуску клиента. Тогда при первичном запуске, когда на конкретном клиенте кроме сервис-браузера еще ничего нет, считываются с сервера все необходимые модули. Считывание происходит по списку, полученному ранее. При перезапуске – проходит проверка и коррекция на клиенте тех модулей, которые изменились на сервере. То есть web-браузер на клиенте ведет архив кэшированных html-страниц, а сервис-браузер – организует, пополняет и модифицирует коллекцию модулей. Процедуры переноса html-страниц и модулей похожи, но модули пишутся обычно (хоть и не обязательно) на том же языке, что и программы сервера. Это дает возможность часто одному и тому же программисту вести как клиентскую, так и серверную часть проекта, что, как показывает практика, дает существенные преимущества при организации работы и минимизации возможных ошибок. Кроме того, резко сокращаются затраты на согласование, так как клиент и сервер используют один и тот же язык.

В общий список модулей (рис.3) на сервере входят два обязательных модуля:

- Модуль сервис-браузера;
- Модуль администратора системы.

Модуль сервис-браузера обычно развертывается в процессе инсталляции сервис-браузера на любом клиенте. Каждый клиент, кроме того, имеет свой список модулей, являющийся подмножеством списка на сервере. Модули, указанные в списке для конкретного типа клиента, копируются в процессе его развертывания, если они не были уже считаны ранее, или были обновлены, те есть содержат на сервере более свежую версию. Нужно учитывать, что типовой модуль администратора системы необходим лишь определенным клиентам. В список модулей клиента может включаться также произвольное количество различных модулей, отражающих особенности конкретной информационной системы.

Таким образом, при использовании сервис-браузера в процессе проектировании информационных систем основным элементом вместо html-страниц представляется модуль, загружаемый в среду клиента и исполняющийся на клиенте.

Основным методом общения с сервером является вызов сервисов из модулей и обработка ответов. Причем в качестве сервисов могут использоваться как web-сервисы, отвечающие требованиям технологии Microsoft (SOAP-сервисы), так и сервисы, использующие архитектуру REST.

Нужно отметить, что обычный web-браузер не имеет встроенных средств защиты информации. При этом предполагается, что средства обеспечения безопасности проектируются отдельно и встраиваются в систему. Сервис-браузер предназначен для проектирования информационных систем и имеет

встроенный комплект средств защиты, что дает возможность разработчикам сосредоточиться на создании приложений, используя существующие методы и средства, обеспечивающие безопасность.

Основные различия web-браузера и сервис-браузера представлены на рис. 4.

Рис. 4. Основные различия web-браузера и сервис-браузера

Основным достоинством сервис-браузера является компактность и возможность использования одного языка как при проектировании серверных, так и клиентских конструкций.

2. Организация защиты при использовании сервис-браузера

При использовании сервис-браузера используется принцип единой идентификации для различных информационных систем, объединенных в одну группу. То есть процессы аутентификации и выполнения разделены. Пользователь получает уникальный идентификатор в процессе представления и может использовать его при входе в любую информационную систему данной группы. Естественно, что это создает ряд удобств в процессе использования.

Но здесь очень важно также одно обстоятельство. Обычно при подключении к системе используются два понятия:

- организация;
- пользователь.

За организацией закреплены одна или несколько информационных систем. Пользователь в общем случае должен иметь имеет доступ к одному или нескольким модулям той или иной системы, которая связана с организацией. Но на начальном этапе пользователи не подключены к системам. И только в процессе настройки должно осуществляться их присоединение к выбранным ИС. Это обеспечивает универсальность использования системы, то есть возможность подключения к ней новых структур, но только с разрешения администратора.

Таким образом, уже в процессе настройки мы сталкиваемся с необходимостью проведения процессов аутентификации и авторизации. При аутентификации определяются идентификаторы системы, организации, пользователя. Например, для пользователя формируется пара «логин-пароль», добавляется «соль», производится хеширование и после передачи и получения результата на сервере система аутентифицирует пользователя [6]. Возможно также использование сертификата, хранящегося на клиенте, и аутентификация с использованием цифровой подписи.

После завершения аутентификации сервер направляет ответ аутентифицированному клиенту, включающий обычно закрытый перечень модулей, которые данный клиент может активизировать. То есть получение списка модулей для конкретного пользователя определяет его полномочия и таким образом формирует параметры авторизации. Здесь прослеживается аналог передачи оглавления портала для обычного браузера. Вместе с тем есть принципиальная разница, которая заключается в том, модули представляют собой не html-страницы, а программы, которые могут быть запущены из сервис-браузера таким же образом, каким вызываются элементы библиотеки.

Нужно отметить, что определением списка модулей процедура авторизации не завершается. Обычно проводится также дополнительная настройка, определяющая, что каждый модуль выступал в определенной роли для каждого пользователя. Например, роль может задавать просмотр данных, вводить разрешения по корректировке данных, определять дополнительные возможности. Настройка ролей производится с использованием языка описания ролей, что позволяет расширять спектр применения тех или иных модулей.

Каждый модуль, копируемый с сервера в хранилище клиента, обычно подписывается цифровой подписью, которая проверяется при запуске. Это определяет достоверность модуля и отсутствие в нем сторонних изменений.

До последнего времени сервис-браузеры в основном работали в среде Windows [3]. Сейчас появились новые возможности, позволяющие переносить как собственно браузеры этого типа, так и информационные системы, построенные на их основе, в среду Linux [5]. Вместе с тем такой перенос, как показала практика, не является простым делом. Это обусловлено тем, что в процессе миграции требуется проводить изменения в программах, причем весьма значительные. С одной стороны, здесь играет определенную роль разность в форматах операционных систем. С другой, в настоящее время графические библиотеки, являющиеся основными, для разных операционных системах не совпадают, хотя и имеют общие черты.

В этом ключе очень важно перенос средств криптозащиты проводить таким образом, чтобы изменения были минимальными. Здесь важно, что наиболее распространенные средства, в частности, КриптоПро имеют варианты, обеспечивающие работу в средах Windows и Linux. Кроме того, существенно, что библиотеки, которые предоставляются в этом случае для работы, имеют близкий формат.

3. Организация защиты обеспечении работы модулей

Основным программным компонентом, отвечающим за решение прикладных задач и интерактивное взаимодействие с пользователем, является модуль. С точки зрения языка программирования это отдельная библиотека, содержащая набор готовых методов, классов и дополнительных ресурсов, например, изображений.

Организацию защиты модуля можно условно разбить на:

- скачивание модуля с сервера и контроль целостности и подлинности модуля перед запуском;
- запуск модуля по стандартному протоколу и передача необходимых данных в модуль;
- работа со средствами криптозащиты сервис-браузера при передаче информации;
- корректное завершение работы модуля.

После создания или коррекции готовый модуль размещают на сервере, с которым взаимодействует сервис-браузер. При размещении администратор системы формирует электронную подпись этого модуля с помощью закрытых ключей текущего сервера.

На клиентском месте после авторизации пользователя, сервис-браузер скачивает файл этого модуля, если он используется данным клиентом. Если файл большого размера, система загрузки разбивает его на части. Сервис-браузер склеивает эти части в один файл, если было разбиение, а затем проверяет контрольную сумму и электронную подпись, полученную с сервера отдельным запросом. Данный механизм направлен на защиту от угрозы Man-In-Middle [7].

В случае поочередного доступа к клиентскому компьютеру для нескольких пользователей браузер позволяет включить режим проверки подписи запускаемых модулей. Что нивелирует угрозы подмены программного обеспечения.

Браузер уже выполнил механизм аутентификации и авторизации и передает управление модулю, но каждая программная реализация сервис-браузера имеет свой протокол запуска модулей. В тоже время можно выделить ряд данных, хранящихся в сервис-браузере, которые н получает при запуске и которые являются обязательными для работы каждого модуля. Такими данными являются:

- идентификатор сеанса;
- идентификатор пользователя;
- информация о пользователе (например, ФИО и должность);
- сертификат и закрытые ключи пользователя;
- роль пользователя;
- идентификатор и/или название организации.

Понятно, что существует вариант, когда модуль, получив идентификатор сеанса, запрашивает все данные у сервера. Такой алгоритм работы может быть реализован, когда требуется передавать большой объем данных для запуска, который значительно больше, чем имеет сервис-браузер.

Часто во время работы информационной системы для пользователя возникает необходимость подписывать документы своей электронной подписью. При этом пользователь уже запустил браузер с использованием своих закрытых ключей. Поэтому разработчик браузера может использовать упрощенный интерфейс криптофункции, который реализован в браузере и связан с данными пользователя, зашедшего в систему.

Последним пунктом защищенной работы модуля является корректное окончание работы. Не часто, но возникают ситуации, когда модуль работает с большим объёмом критичных данных. Если же прервать процесс, целостность данных может быть нарушена. Сервис-браузер поддерживает механизм опроса модулей, для обеспечения корректного завершения работы.

4. Заключение

В настоящее время в качестве клиента информационных систем все чаще применяются web-браузеры. При этом следует учитывать, что эти браузеры достаточно сложны, объемны, и как правило, при написании программных комплексов требует использования нескольких языков программирования. В связи с этим возможно в качестве клиента применять специализированный браузер, который специально разрабатывается для информационных систем. Таким браузером может быть сервис-браузер, разработанный в институте проблем управления РАН.

Особенностью использования сервис-браузера является применение вместо html-страниц программных модулей. Общение «клиент-сервер» происходит путем копирования на клиентскую часть программных модулей и вызова из этих модулей сервисов, а также обработки завершения каждого сервиса. В качестве сервисов могут использоваться как SOAP-сервисы, так и REST-сервисы. Нужно отметить, что обычный web-браузер не имеет в своем составе средств обеспечения безопасности данных. В то же время сервис-браузер по определению включает такие средства, что дает возможность при проектировании проводить разделение информационной части и части, обеспечивающей защиту.

В настоящее время появилась возможность использовать не только сервис-браузеры в среде Windows, но и сервис-браузеры, ориентированные на операционные системы типа Linux. Реальное проектирование показало, что возможен перенос информационных систем, работающих под управлением таких браузеров, в среду Linux с использованием соответствующих защитных функций.

Литература

1. *Klaus-Dieter Schewe, Bernhard Thalheim* Design and Development of Web Information Systems. – Berlin, Heidelberg: Springer, 2019. – P. 599.
2. *Kurako E.A., Orlov V.L.* Development Trends of Web Browser and Service Browser Clients in Large-Scale Systems / Proceedings of the 16th International Conference Management of Large-Scale System Development (MLSD). Moscow: IEEE, 2023. С. https://ieeexplore.ieee.org/document/10303806.
3. *Курако Е.А., Орлов В.Л.* Сервис-браузеры для информационных систем // Программная инженерия. М., 2017. Т. 8, № 9. С. 413-421.
4. *Курако Е. А., Орлов В. Л.* Способ организации взаимодействия клиента с сервером приложений с использованием сервис-браузера: Патент на изобретение № 2656735 РФ; Зарег. 06.06.2018
5. Курако Е.А., Асратян Р.Э., Орлов В.Л. Сервис-браузер в среде Linux // Программная инженерия. 2024. N5. – С. 219-228.
6. *Козлов А.Д., Орлов В.Л.* Методы и средства обеспечения информационной безопасности распределенных корпоративных систем. – М. ИПУ РАН,2018. -155с.
7. *Орлов В.Л., Курако Е.А.* Сервис-браузер и атаки типа Man in the middle // Материалы 29-й Международной конференции «Проблемы управления безопасностью сложных систем» (ПУБСС'2021, Москва). М.: ИПУ РАН, 2021. С. 265-268.