электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.023,62,63

Технология одновременной безопасной обработки данных разных уровней секретности

Малыгин А.О., *аспирант*
Россия, 198504, г. Санкт-Петербург,
Санкт-Петербургский Государственный Университет,
математико-механический факультет, кафедра «Системное программирование»

Одеров Р.С., *студент*
Россия, 198504, г. Санкт-Петербург,
Санкт-Петербургский Государственный Университет,
математико-механический факультет, кафедра «Системное программирование»

Серко С.А., *студент*
Россия, 198504, г. Санкт-Петербург,
Санкт-Петербургский Государственный Университет,
математико-механический факультет, кафедра «Системное программирование»

Научные руководители:
Терехов А.Н. д.ф-м.н., профессор
Россия, 198504, г. Санкт-Петербург,
Санкт-Петербургский Государственный Университет

Баклановский М.В., старший преподаватель
Россия, 198504, г. Санкт-Петербург,
Санкт-Петербургский Государственный Университет
carbon2320@gmail.com

Введение

При работе в среде с высокими требованиями к безопасности необходимо свести к минимуму возможность внедрения вредоносного ПО и, соответственно, вероятность кражи конфиденциальных данных. Если пользователь хранит или обрабатывает секретную информацию локально, то любая брешь в защите может привести к ее потере. Основными источниками угроз являются запускаемые на локальной машине приложения и сеть Интернет.

Представим, что пользователь на своей машине работает с конфиденциальными данными. Если во время нахождения этих данных на локальном компьютере он использует интернет-браузер, то не исключены ситуации внедрения вредоносного кода в систему или в ПО, а также эксплуатация уязвимостей и, как результат, хищение важной информации.

Сейчас описанная проблема решается с использованием разных компьютеров: один из которых, для взаимодействия непосредственно с важными данными, а другой, – для запуска необходимых второстепенных приложений и выхода в Интернет. Такой способ очень не рационален, поэтому мы предлагаем работать со всеми приложениями и данными в рамках одной машины, причем делать это безопасно! Для этого необходимо изолировать компьютер пользователя от доступа к сети Интернет, переместить важные данные в отдельное хранилище с повышенной степенью защиты, отказаться от исполнения приложений локально и рассредоточить их по серверам в соответствии с требуемым уровнем безопасности. Клиенту, в данном случае, будут передаваться только состояния приложений.

Идея реализации

Клиент, используя свой локальный компьютер, работает с приложениями. Он видит привычные окна запущенных программ, взаимодействует с ними через клавиатуру и мышь. Однако ключевое отличие от работы с обычной системой заключается в том, что пользователю предоставляется лишь иллюзия локальной работы. А именно, пользователь видит лишь «мгновенные снимки» (кадры, фреймы) приложения, которое на самом деле функционирует в удаленном облаке, состоящем из специальных серверов.

На локальном компьютере с высокой степенью защиты (как программной, так и аппаратной) функционирует только тонкий клиент, реализующий протокол общения с серверами приложений. Машина изолирована от сети Интернет и не имеет сторонних программ. По требованию пользователя приложения запускаются в облаках, которые могут располагаться внутри организации и тоже не иметь выхода в Интернет. Клиент получает видео-поток, отображающий состояние запущенной программы, и демонстрирует его пользователю. Пользователь, в свою очередь, видит приложение так, будто оно запущенно локально. В качестве элементов взаимодействия с приложением на соответствующий сервер отправляются сигналы нажатий клавиш клавиатуры и мыши, а также относительные координаты курсора. Обрабатываются, и обновленное состояние приложения передается пользователю.

Каждое приложение в облаке запускается в отдельной изолированной среде, которая создается с использованием технологии виртуализации. Такой подход совершенствует уровень безопасности и позволяет построить отказоустойчивую систему. Важной особенностью является возможность одновременной работы с приложениями для различных операционных систем (Linux, Windows, Mac OS) на одном компьютере.

Для повышения степени защиты можно использовать сертифицированные методы шифрования информации, передаваемой между облаком и локальной машиной.

Серверы приложений в облаках могут иметь различные уровни безопасности. Наиболее защищенным должен быть сервер для запуска приложений, обрабатывающих секретные данные: желательно запретить ему доступ в Интернет и ограничить к нему физический доступ сотрудников организации.

Метод взаимодействия приложения с клиентом.

В процессе взаимодействия с пользователем, становится необходимым запускать приложения на удаленной машине. Для этого, воспользуемся, хорошо известным, протоколом X11. При помощи X-клиента, мы сможем передавать окно приложения на локальную машину, а используя X-сервер, сможем рисовать это окно и передавать происходящие события мыши и клавиатуры обратно на X-клиент. Таким образом, мы получаем, некое подобие безопасной среды для взаимодействия с пользователем, которая обеспечивает целостность наших данных.

На данный момент, по протоколу X11, возможно обеспечить передачу данных в следующих направлениях среди, ниже приведенных, ОС:

1. С Linux на Linux, т.к. протокол X11 лежит в основе системы;
2. С Linux на Windows OS, т.к. существуют решения для Windows OS (MobiXTerm);

Но, необходимо учесть, что не были реализованы следующие варианты, поскольку в основе Windows OS лежит протокол отличный от X11:

1. С Windows на Linux;
2. С Windows OS на Windows OS;

Хочется отметить, что в данном случае рассматриваются отдельные окна одного приложения, а не весь рабочий стол в целом со всеми имеющимися приложения. С точки зрения приложения, это делает клиент тонким.

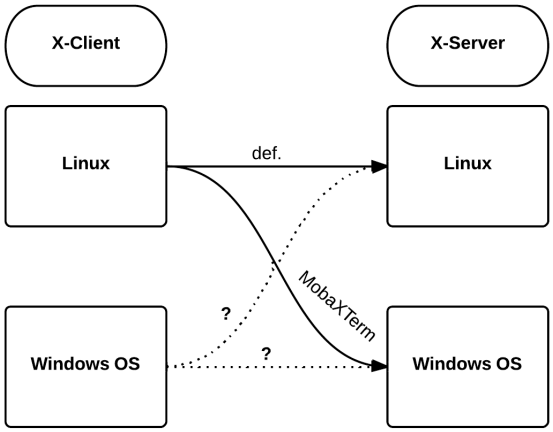

Рис. 1. Реализованные решения передачи окна по протоколу X11

Реализация такого X-клиента для Windows OS, позволит работать с разными типами приложений, запущенных на разных операционных системах, например, таких как Windows 3.1, Windows XP, Windows 7, Windows 8. (рис.2).

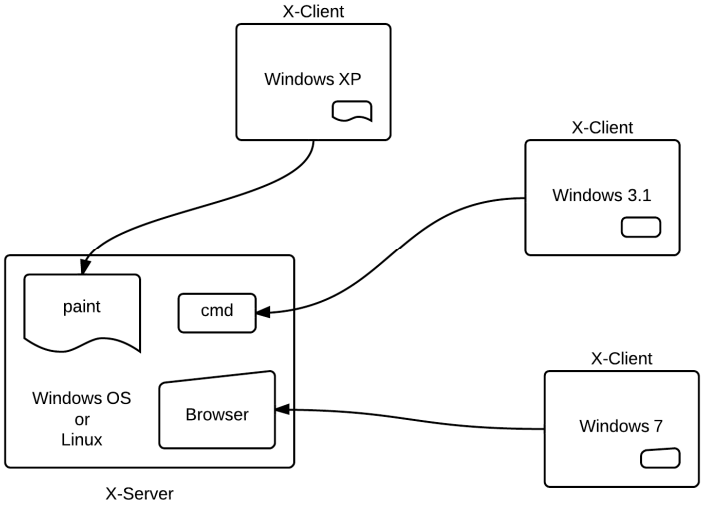

Рис. 2. Демонстрация тонкого клиента и отображения приложений с разных ОС

При оценки сложности реализации клиента для Windows OS, было выяснено, что переписывание всех существующих приложений Windows OS с учетом протокола X11 не представляется возможным, точно также, как и изменения GDI на X11, без исходного кода, является трудоемкой задачей. Поэтому было разработано решение, которое позволяет передавать фреймовый (кадровый) буфер приложения на X-сервер в потоке данных.

В ходе реализации X-клиента для Windows OS предстояло решить следующие задачи:

1. Захват изображения приложения;
    a. Получение/изменение окна приложения;
    b. Захват многооконного приложения;
2. Передача фрейма по протоколу X11;
    a. Сжатие изображения;
    b. Создание окна из изображения с учетом пересчета координат;
3. Получение и обработка событий X-сервера;
    a. Получение кликов мыши/символов клавиатуры;
    b. Подбрасывание полученных событий в буфер приложения;

Стоит заметить, что многие из задач трудно разрешимы, как для разных версий Windows OS, так и для одной. Многие из методов являются эвристическими. В некоторых методах, можно воспользоваться Windows API, например, для решения таких задач, когда разные приложения Windows OS используют разные стили окна, скажем, установлена тройная рамка или одинарная, а возможно и пользовательская надстройка. В результате был реализован демо-клиент, охватывающий основные подзадачи в рамках дипломной работы.

Управление политиками контроля доступа на основе ролевой модели.

Существует множество организаций, сотрудники которых в рамках своих должностных обязанностей работают с документами различных уровней секретности. Например, оборонные предприятия, служба безопасности, разведывательное управление, серьезные административные органы и т.п. Документы, с которыми работают сотрудники этих организаций, обычно делятся на следующие категории:

- документы, отнесенные к Государственной Тайне;
- конфиденциальные/Для служебного пользования;
- несекретные;

В свою очередь документы, отнесенные к Гос.Тайне, снабжаются грифами секретности - реквизитами, свидетельствующими о степени секретности сведений, содержащихся в этих документах. В Российской Федерации согласно постановлению №870 от 4 сентября 1995 года “Об утверждении правил отнесения сведений, составляющих государственную тайну, к различным степеням секретности”, определяются следующие грифы секретности (в порядке уменьшения степени секретности):

- особой важности,
- совершенно секретно,

- секретно.

Чем выше степень секретности обрабатываемой информации, тем больше потенциальный ущерб, который будет нанесен организации/министерству/стране в случае непредвиденного рассекречивания этих сведений. Поэтому при работе с документами различных уровней конфиденциальности необходимо соблюдать ряд требований, связанных с обеспечением должного уровня безопасности системы, обрабатывающей эти данные. Такие требования подробно описаны в законах, государственных стандартах, руководящих документа

Неотъемлемой частью этих “спецификаций” является описание *политик контроля доступа*, которые контролируют действия всех процессов в рассматриваемой системе. Политики разграничения доступа позволяют идентифицировать пользователей и их процессы и запрещать или разрешать им определенный набор действий в системе в зависимости от присвоенного уровня доступа к данным. Таким образом, осуществляется контроль над системой, способствующий предотвращению некорректных/запрещенных действий, направленных на нарушение целостности/доступности/конфиденциальности информации.

Политики контроля доступа основаны на *моделях контроля доступа* - математически формализованных системах, включающих в себя (в самом общем случае) субъекты (пользователи/процессы), объекты (данные/документы/файлы) и правила доступа, по которым вычисляется возможность выполнения субъектом набора операций над объектом. Среди моделей контроля доступа можно выделить две наиболее распространенных: MAC (Mandatory Access Control / Мандатный контроль доступа) и DAC (Discretionary access control / Дискреционный контроль доступа).

Эти модели входят в список требований (ГОСТы, ISO/Common Criteria) по обеспечению безопасности систем, работающих с секретными данными. В зависимости от уровня секретности данных в системе, к ней применяются *разные* модели.

Ключевой особенностью системы MCD[3] является возможность *одновременной* работы с документами *разных* уровней секретности. Таким образом, следуя стандартам, необходимо сочетать в рамках одной системы несколько концептуально разных моделей контроля доступа.

Описание моделей

DAC

В рамках классической дискреционной модели контроль доступа осуществляется на основе списков доступа (ACL, access control list) или матриц контроля доступа (ACM, access control matrix). Каждая пара субъект-объект имеет явный перечень разрешенных операций. Т.о. при попытке субъекта “S” выполнить действие “A” применительно к объекту “O” система контроля доступа просканирует данные в ячейке матрицы ACM (или обратится к списку доступа ACL, связанному с “O”) и попытается найти “A” в наборе разрешенных операций. В случае успешного поиска субъекту “S” будет дано право на выполнение “A”. Иначе - действие запрещается.

DAC - довольно уязвимая модель, т.к. владелец объекта вправе произвольным образом раздавать другим пользователям права на выполнение различных операций с этим объектом. Это может привести к серьезным последствиям в случае целенаправленной или случайной передаче злоумышленнику критически важны прав на секретный объект.

MAC

Мандатные модели делятся на два типа: одни направлены на поддержание конфиденциальности, а другие - на поддержание целостности данных. В зависимости от выбранной модели, система безопасности будет варьировать правила, по которым определяется возможность доступа субъектов к объектам.

Мандатная модель основана на присвоении объектам и субъектам специальных *меток безопасности* (для субъектов их также называют *уровнями доступа*). С формальной точки зрения все метки образуют решетку с частичным отношением порядка (доминирования). В классических MAC моделях обычно определяются две операции: чтение и запись. При попытке субъекта S выполнить чтение/запись в отношении объекта O, система сравнивает их метки.

Если модель направлена на сохранение конфиденциальности данных, то:

- Чтение разрешено только в том случае, когда метка объекта *не больше* метки субъекта
- Запись разрешена только в том случае, когда метка объекта *не меньше* метки субъекта

Если модель направлена на поддержание целостности данных, то:

- Чтение разрешено только в том случае, когда метка объекта *не меньше* метки субъекта
- Запись разрешена только в том случае, когда метка объекта *не больше* метки субъекта

Эти правила являются вполне естественными и отражают суть конфиденциальности (запрет на запись информации в менее доверенную среду и на чтение информации более секретного уровня) и целостности (запрет на запись информации в “более целостную” среду и на чтение информации из “менее целостного” источника).

RBAC

RBAC-модель является, по мнению автора, наиболее естественным способом построения политик контроля доступа, т.к. она основана на интуитивно понятном процессе присвоения пользователям конкретных “должностей”, в рамках которых им разрешается выполнять определенные действия.

Основными элементами ролевой модели являются Пользователи (Users), Роли (Roles), Разрешения (Permissions), Сессии (Sessions). Разрешение - множество пар (Объект, Операция), определяющих, какие действия можно совершать относительно объекта. Роль - “должность” с соответствующим набором разрешений. Можно определить специальное отношение PA (от англ. “permission assignment”), которое сопоставляет каждой роли подмножество множества Разрешений, и таким образом определяет, какие действия будут разрешены пользователю в пределах этой роли. Администратор системы заранее определяет, какие роли будут доступны конкретным пользователям. При входе в систему пользователь инициирует новую сессию, выбирает из доступных ему ролей те, которые хочет активировать для работы, после чего он может выполнять только операции, предусмотренные *разрешениями* выбранных (активных) ролей.

Существуют дополнения к базовой модели RBAC. Одна из них вводит *ролевые иерархии* (РИ), которые являются естественным отражением должностных структур практически любой организации, и поэтому легко вписываются в их политики безопасности. Цель РИ - возможность наследования привилегий/разрешений.

Также существует расширение описанной модели, внедряющее так называемые *ограничения*. Ограничения - некоторые правила, касающиеся любых компонент модели. Классическое ограничение, известное как Статическое Разделение Обязанностей (SSoD, Static Separation of Duties), может запрещать некоторым ролям одновременно быть в списке доступных для одного пользователя. Например, в вымышленной организации пользователь не может обладать и ролью “Директор”, и ролью “Бухгалтер”.

Решение задачи управления различными политиками доступа в рамках системы MCD

Важным теоретическим результатом, касающимся моделей контроля доступа, является тот факт, что в терминах RBAC можно описать как DAC, так и MAC модели. Это доказывает состоятельность предположения о возможности построения системы MCD. Существуют четко определенные последовательности действий, необходимых для перевода объектов одной модели в объекты другой модели “без потерь” (т.е. с сохранением ограничений, правил и требований безопасности).

Однако стоит отметить, что сложность и размер RBAC, в которой моделируются упомянутые выше MAC и DAC, существенно возрастает, что может повлечь за собой проблемы поддержки и управления. Поэтому необходимо продумать архитектуру и инфраструктуру хранения и работы модели RBAC, а также реализовать удобные инструменты управления, в которых предусмотрены интерфейсы работы как с RBAC, так и с моделируемыми в ней MAC, DAC. Что касается последнего, здесь особенно к месту была бы набирающая популярность концепция визуального программирования, в рамках которой пользователи (в том числе далеко не профессиональные программисты) могли бы в доступной и понятной манере конфигурировать политики доступа путем создания визуальных диаграмм/схем, а затем загружать их в систему управления доступом.

Были поставлены следующие задачи, необходимые для реализации *Proof of Concept* системы управления политиками контроля доступа для MCD:

- Проектирование архитектуры системы управления политиками контроля доступа
- Реализация структуры хранения модели RBAC и алгоритмов работы с ней
- Возможность пакетного внесения данных в модель (например, на основе XML-файлов)
- Возможность удобного/понятного отображения политик доступа и содержимого моделей (визуализация в виде схем, диаграмм и т.п.)
- Возможность визуального программирования политик контроля доступа

На текущий момент уже реализовано, а также находятся в процессе реализации:

- Структура хранения базовой модели RBAC
- Возможность пакетного внесения данных на основе файлов XML
- Управляющая функциональность базовой модели RBAC (без иерархий и ограничений)
- Визуализация содержимого модели в виде схем/диаграмм

Организация эффективного хранения образов виртуальных машин (ВМ) за счет дедупликации

Безопасность рабочего пространства пользователя в системе достигается путем запуска каждого приложения в отдельной виртуальной машине на сервере и удаленной работой с ним посредством протокола X11. Потенциально большие количества одновременно работающих пользователей и доступных им приложений ставят вопрос об организации эффективного хранения порождаемых ими образов виртуальных машин.

Для эффективного хранения предполагается использовать сочетание методов архивации и дедупликации. Под эффективностью хранения мы понимаем оптимальное сочетание времени запуска виртуальной машины. С использованием технологии дедупликации, обнаруживаются и исключаются избыточные данные в дисковом хранилище. Это позволяет сократить объёмы физических носителей для хранения тех же объёмов данных.

Тема является актуальной в области систем хранения данных на протяжении нескольких лет. Отметим, что MCD придется хранить большой объем данных, где неизбежно будут встречаться дубликаты, а также идентичные, по многим критериям, данные. В следствии устранения которых, можно значительно сократить объемы хранения, что позволит обеспечить эффективность работы.

Простейшей из реализации этой технологии является дедупликации на «файловом» уровне. Такой способ достаточно известен в *UNIX* системах, посредством механизма линкования. Физическая цепочка блоков может адресоваться из разных мест файловой системы. Например, при использовании стандартных библиотек различными программами, вместо того, чтобы копировать один и тот же файл в десятки мест на диске, можно хранить одну копию, а остальные заменять на соответствующий линк. Когда операционная система или приложение обращается к файловой системе за этим файлом, то происходит "прозрачное" перенаправление по этому линку к экземпляру файла.Для данных которые не являются "файловыми" на помощь приходит механизм под названием "субфайловая" или, так называемая, блочная дедупликация. Таким образом, можно получить один из шагов к эффективному распределению и хранению образов ВМ.

Заключение

В результате проделанной работы, были сделаны важные шаги для реализации безопасной системы с одновременной обработкой конфиденциальных данных. В которой

используется новый метод взаимодействия пользователя и приложения по известному протоколу X11. Также рассмотрены способы управления политиками контроля доступа, как среди приложений, так и среди пользователей. Выделены отдельные аспекты реализации таких политик, а также было рассказано возможности композиции известных решений. Было показано, каким образом, можно эффективно реализовать хранилище виртуальных машин, что является, одной из потенциально важных задач.

На данный момент, уже были сделаны тестовые сборки, для дальнейших исследовании описанной системы. Область исследований является обширной, а задача остается актуальной на сегодняшний день. По данной теме были защищены курсовые и дипломные работы, написаны статьи и опубликованы многие работы. Для получения сведений о функционировании и работоспособности, в скором времени, планируется реализовать и протестировать рабочий прототип MCD.

Список литературы

1. Малыгин А.О. Разработка и реализация X-Client для ОС Windows // 4-я Всероссийская Научная Конференция по проблемам информатики “СПИСОК 2013”: труды. СПб: СПбГУ, Мат-Мех, 2013 г. С. 25-38.
2. Одеров Р.С., Серко С.А., Отказоустойчивая виртуализация на основе гипервизора Xen, // Научно-практическая конференция "Актуальные проблемы организации и технологии защиты информации": труды. СПб: СПбНИУ ИТМО, 2012 г. С. 112-122.
3. Баклановский М.В., Малыгин А.О., Одеров Р.С., Серко С.А. Технология одновременной безопасной обработки данных разных уровней секретности (MCD) // XXXVIII Академические чтения по космонавтике, секция 22: труды. СПб: СПбГУ, 2014 г. С. 651-652.
4. Evaluator Group, Inc. Data Deduplication, why when where and how? // Fellows, Russ, Evaluator Group, Inc.
5. Integrity considerations for secure computer systems, Prepared for DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS, ELECTRONIC SYSTEMS DIVISION, AIR FORCE SYSTEMS COMMAND. Hanscom Air Force Base, Bedford, Massachusetts, April 1977.
6. Mutual exclusion of roles as a means of implementing Sepration of Duty in RBAC systems, D.Richard Kuhn, 1997.

7. Matunda Nyanchama, Sylvia Osborn, Access rights administration in Role-Based Security Systems,. The dep. of CS, The University of Western Ontario, 1994
8. Sadekhi Cubaleska, Access Control Fundamentials, 2009. Режим доступа: http://www.trust.rub.de/media/ei/lehrmaterialien/232/OSS_chap3.pdf (дата обращения: 21.02.2014).
9. Sylvia Osborn ,Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies, The University of Western Ontario; RAVI SANDHU and QAMAR MUNAWER, George Mason University. 2000.
10. Secure Computer Systems: Mathematical Foundations by D. Elliott Bell and Leonard J. LaPadula, 1 March 1973 // MITRE Technical Report 2547, Vol. I.
11. Windows application programming interfaces. Режим доступа: http://msdn.microsoft.com/en-us/library/windows/desktop/ ff818516(v=vs.85).aspx (дата обращения: 21.02.2014).
12. X.Org Foundation (or X.Org for short), Режим доступа: http://www.x.org (дата обращения: 21.02.2014).