Эссе по курсу "Защита информации", кафедра радиотехники, Московский физико-технический институт (ГУ МФТИ), http://www.re.mipt.ru/infsec

Безопасность NAS

Сидоркин О.С. 111 гр.
15/04/2005

NAS – Network Attached Storage (Система хранения присоединенная к сети) как правило подключается к локальной сети и предоставляет свое дисковое пространство. Вся его вычислительная мощность используется для задач обслуживания и хранения файлов. Основным преимуществом NAS-серверов является их независимость от операционной системы клиента. Как правило, на них применяется специализированная операционная система, либо стандартные ОС, но с ограниченной функциональностью.
Как правило, у них относительно слабые процессоры и небольшое количество памяти.

Система безопасности NAS должна решать следующие задачи:

1. Авторизация пользователя (определение и реализация прав доступа к файлам)
2. Обеспечение конфиденциальности (невозможность перехватить передаваемую информацию)
3. Обеспечение целостности данных (при хранении и передаче – невозможность изменить данные, хранящиеся на сервере)

Основными требованиями к системе безопасности следующие:

1. Малые требования к вычислительным ресурсам и памяти
2. Небольшой размер дополнительной информации, передаваемой по сети
3. Масштабируемость.

Существующие схемы

Все существующие схемы можно условно разделить на 2 группы: Производящие шифрование данных на серверах, и не производящие. Недостатки первых – малая масштабируемость, требовательность к вычислительным ресурсам. Недостатки вторых – трудность совместного использования информации.

1. NASD (Network Attached Secure Disks)
   Менеджер файлов (FM) и собственно система хранения (SS) располагаются на разных серверах. Клиент посылает запрос на FM, FM авторизует клиента, проверяет его права и выдает ему ключ. Используя этот ключ, клиент непосредственно общается с SS. При истечении срока действия ключа клиент запрашивает новый у FM.
   Конфиденциальность: авторизация в схему не входит (предлагается использовать любой из существующих методов). После авторизации и проверки прав пользователь получает сессионный ключ, который состоит из собственно ключа, сведений о правах данного клиента и MAC. Чтобы уменьшить требования к вычислительной мощности серверов применяются MAC на хэш-функциях.
   Обеспечение целостности: Каждый пакет, передаваемый между клиентом и SS, содержит MAC.
   Преимущества данной схемы – масштабируемость (легко добавляются новые SS), возможность разгрузить сеть (одно из возможных узких мест всех NAS)
   Недостатки – клиент должен доверять FM. Если FM компрометируется, все сервера перестают работать.

2. PASIS (Survivable Storage)
   Система хранения, способная функционировать при компрометации серверов. Используется пороговая схема – пока число скомпрометированных серверов не превышает определенного значения, клиент не замечает нарушения. Клиент кодирует и разделяет файл на части и распределяет фрагменты файла по нескольким серверам. Если скомпрометировано небольшое число серверов, то клиент может восстановить исходный файл.
   Конфиденциальность: клиент разделяет сообщение на n частей, из которых по любым m частям можно восстановить файл, но перехват меньше чем p частей не позволит восстановить никакую часть файла. Применение шифрования может сильно повысить устойчивость схемы.
   Обеспечение целостности: Так как файл восстанавливается по m частям, которые могут быть получены от любых из n серверов, то для нарушения целостности атакующему необходимо исказить все m частей. Если клиент не может восстановить файл по полученным частям (например, из-за искажения части из них), клиент запрашивает данные повторно.
   Недостаток – необходимость запрашивать сразу несколько серверов (схема плохо работает для маленьких файлов).

3. S4 (Self Securing Storage System)
   Основной упор сделан на обеспечение целостности данных. При каждом изменении создается новая версия файла. Все операции с файлом записываются в лог. При вторжении есть возможность «откатить» неправильные изменения. Вся работа с журналированием и созданием версий переложена на файловую систему. Авторизации не предусмотрено – предполагается стандартное решение. В случае успешной авторизации дальнейших проверок не производится. Неправомерные изменения откатываются при аудите.

4. CFS
   Cryptographic File System. Основная идея – разгрузить сервер за счет клиента. Все данные шифруются на клиенте и сохраняются сервером в зашифрованном виде. Авторизация не требуется: доступ есть у всех, у кого есть ключ к файлу. Сервер должен блокировать все попытки обращения к зашифрованному файлу напрямую в обход CFS.
   Конфиденциальность: В расшифрованном виде файл существует только у клиента. Даже при компрометации сервера злоумышленник не может получить доступ к содержимому файла.
   Существует стандартная реализация поверх NFS для UNIX систем. Недостатки: неудобство – требуется передавать ключи. Трудно отозвать права на доступ к файлу.

5. SNAD (Secure Network Attached Disks)
   Как и CFS позволяет избежать раскрытия при компрометации сервера. Расшифрованный файл существует только на машине клиента.

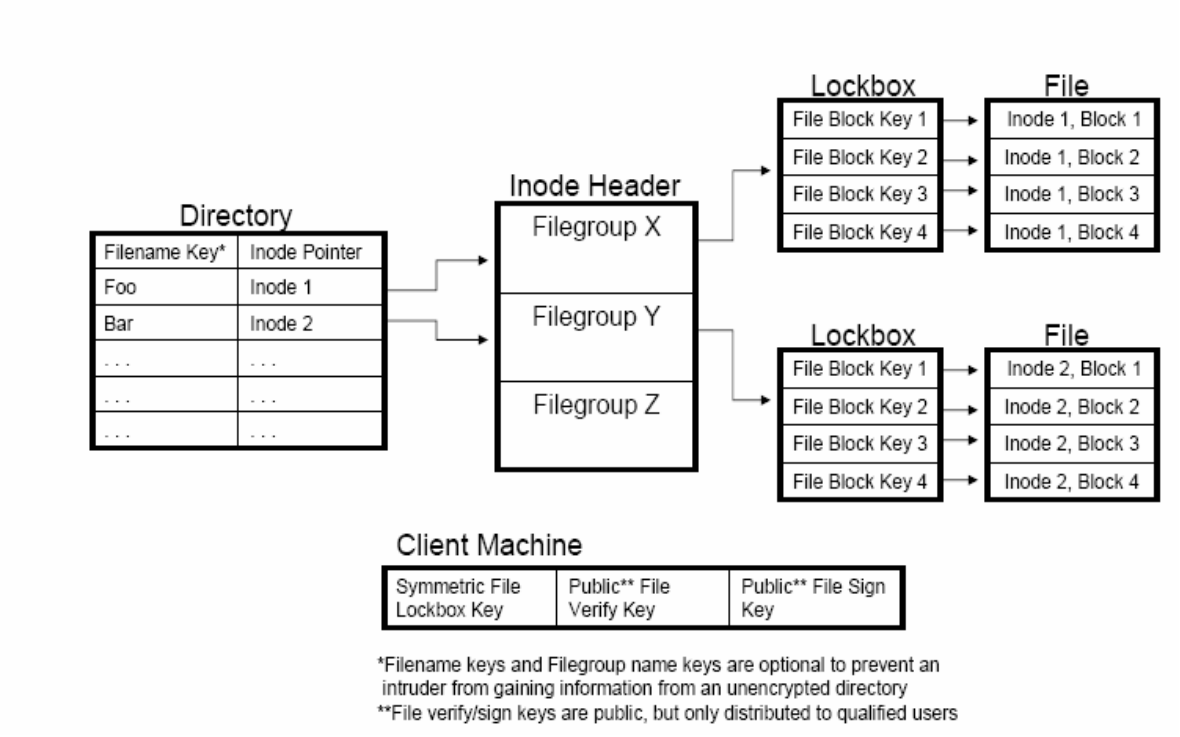
   Каждый файл разбивается на блоки, каждый из которых шифруется своим ключом по симметричной схеме. Ключевой объект содержит UserID создателя файла, ID ключа и подпись пользователя, который модифицировал файл последним. Ключевой объект состоит из записей, для каждого из пользователей, которые имеют доступ к файлу. В каждой записи содержится UserID, флаг «Право на запись» (определяет как право писать в ключевой файл, так и право на запись в

файл данных), и сам ключ, зашифрованный открытым ключом пользователя (закрытый ключ хранится на машине пользователя). Для каждого пользователя на сервере хранится запись, в которой содержится UserID, открытый ключ пользователя, HMAC для проверки целостности записи и время последней записи (для предотвращения replay-атак).

Для обеспечения целостности для каждого файла хранится нелинейная контрольная сумма расшифрованного файла.
Слабым местом схемы является хранилище ключей (доверяемый сервер) – при его отказе все хранилище перестает работать. Ещё одним недостатком является трудность отзыва прав – требуется обновлять много записей и заново шифровать весь файл.

Хотя все вычисления происходят на клиенте, они все равно достаточно сложные. Разработаны несколько упрощенных схем, позволяющие выиграть в производительности за счет снижения безопасности. Самая защищенная (подписан каждый блок, проверяется при каждой операции чтения и записи) – не приспособлена для решения большинства задач. Наименее защищенная (все подписи заменены на HMAC) – сравнима по скорости с незащищенными системами.

6. PLUTUS
   Так же как и SNAD использует хранилище ключей. Шифрование и расшифрование производится на клиентах, чтобы разгрузить сервер. Клиенты должны распространять ключи самостоятельно.
   Конфиденциальность: Так же как и в SNAD каждый файл разбивается на блоки, каждый блок шифруется уникальным ключом по симметричной схеме. Ключи от файла с данными хранится в зашифрованном виде в хранилище ключей (lockbox). Все файлы разбиты на группы, у каждой группы – свой lockbox. Для расшифровки Lockbox'а используется один ключ.



Lockbox создается владельцем группы, который раздает ключи от lockbox'а всем пользователям, которые имеют право использовать файлы данной группы. Для проверки целостности используется подпись. У пользователей, имеющих право записи есть оба ключа (открытый и закрытый), у пользователей с правами «только чтение» есть только ключ для проверки подписи. При запросе сервер выдает клиенту lockbox и запрашиваемый блок. Клиент расшифровывает lockbox, потом ключом расшифровывает блок. Недостатком схемы является трудность отзыва прав (все файлы в группе должны зашивровываться заново, при этом меняются ключи от lockbox'ов).

7. SiRiUS (Securing Remote Untrusted Storage – Обеспечение безопасности при хранении в незащищенном хранилище)
   Реализация защищенного хранения поверх незащищенной схемы. (NFS, CIFS). Обеспечивает защищенный обмен без модификации существующей серверной части.
   Конфиденциальность: Все файлы шифруются перед сохранением на незащищенном сервере. Дополнительное преимущество – разгрузка сервера (все сложные в вычислительном плане операции выполняются на клиенте), кроме того, отпадает необходимость в защищенном канале до сервера. У владельца файла есть Главный Ключ Шифрования (MEK) и Главный ключ для подписи (MSK). Для каждого файла есть файловый ключ для шифрования (FEK), который доступен всем пользователям с правом на чтение и файловый ключ для подписи (FSK), который предоставляется только тем пользователям, у которых есть право на запись. Кроме непосредственнно файлов с данными, в системе есть файлы с метаданными. Для каждого файла данных существует отдельный файл метаданных (md). В файле данных (d) содержатся зашифрованные и подписанные данные. В md содержатся данные о правах на d.
   Md состоит из нескольких записей. В каждой записи содержится UserID (в открытом виде), и FEK и (может быть) FSK, зашифрованные открытым ключом пользователя. Кроме того, в md есть имя файла (чтобы избежать подмены). Md подписывается при помощи MSK.
   Чтение: пользователь получает md, расшифровывает свою запись и получает FEK. Потом пользователь проверяет подпись и, используя FEK, расшифровывает файл.
   Запись: после изменения, пользователь подписывает зашифрованный файл используя FSK (если его нет, то он не сможет сгенерировать правильную подпись).
   Выдача прав: очевидно.
   Отзыв прав: Владелец генерирует новый FEK и FSK, зашифровывает и подписывает файл, после чего обновляет md и mdf.

   Обеспечение целостности: Часть данных хранится в открытом виде, что позволяет серверу проверить целостность данных.

   Недостатки: компрометация сервера приводит к отказу в обслуживании (но не раскрытию информации).

Перспективы.

Из-за того, что сами сервера могут являться узким местом всей системы, стараются свести количество вычислений на сервере к минимуму. Поэтому при авторизации стремятся заменить схемы, использующие криптографию с открытым ключом на схемы на хэш-функциях. Кроме того, все операции связанные с шифрованием (даже по симметричной схеме) стремятся переложить на клиента.

Ссылки

1. Paul Stanton. **Securing Data in Storage: A Review of Current Research.** Department of Computer Science, University of Illinois at Urbana-Champaign (http://www.ncassr.org/projects/storage-sec/papers/stantontechnicalreport2004.pdf)

2. Howard Gobioff, Garth Gibson, Doug Tygar. **Security for Network Attached Storage Devices.** October 23, 1997 School of Computer Science Carnegie Mellon University Pittsburgh (http://www.cs.berkeley.edu/~tygar/papers/Security_for_NASD.pdf)

3. Eu-Jin Goh,Hovav Shacham, Nagendra Modadugu, Dan Boneh. **SiRiUS: Securing Remote Untrusted Storage.** Stanford University. (http://www.isoc.org/isoc/conferences/ndss/03/proceedings/papers/9.pdf)

4. Эрнст Долгий. **Что нам стоит NAS построить?** (http://www.citforum.ru/nets/storage/nas1.shtml)