МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество

АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

ФАКУЛЬТЕТ «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

Допущен к защите:
зав. кафедрой «Компьютерные технологии»
д. ф.-м. н., профессор_____Куралбаев З.К.
«___»_____20__г.

Магистерская диссертация
«Построения систем защиты информации для программных пакетов, используемых в монопольном доступе»
специальность: 6М070400 – Вычислительная техника и программное обеспечение

Магистрант_____Кулумгариев М.М.
подпись (Ф.И.О.)

Руководитель диссертации_____Шайхин Б.М.
подпись (Ф.И.О.)

Алматы 2014 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 5
1. ОБЗОР И АНАЛИЗ СОВРЕМЕННЫХ АЛГОРИТМОВ ШИФРОВАНИЯ 8
1.1 Шифрование данных 8
1.2.1. Криптосистема Эль-Гамеля 11
1.2.2. Криптосистема Ревеста-Шумира-Эйдолмана 12
1.3. Адаптированный метод асимметричного шифрования 13
1.4. Преимущества применения полиморфных алгоритмов шифрования .. 15
1.5. Функциональность системы защиты 17
1.6. Описание показателей алгоритмов шифрования 17
1.7 Влияние алгоритмов шифрования на качество информационных систем 18
2 ИДЕНТИФИКАЦИЯ И АУТЕНТИФИКАЦИЯ 20
2.1 Основные понятия 20
2.2 Парольная аутентификация 22
2.3 Одноразовые пароли 22
2.4 Сервер аутентификации Kerberos 23
2.5 Идентификация и аутентификация с помощью биометрических данных 26
2.6 Аутентификация как базис безопасности 27
2.6.1 Этапы входа в систему 27
2.6.2 Способы аутентификации 28
2.7 Парольная аутентификация 28
2.7.1 Задание стойкости паролей 28
2.7.2 Двухфакторная аутентификация 30
2.7.3 Аутентификация на основе одноразовых паролей 30
2.7.4 Одноразовые пароли. Терминология. Форм-факторы 32
2.8 Oracle Siebel CRM 39
2.8.1. Аутентификация для Siebel Видеорешения 44
2.8.2. Аутентификация пользователя по обеспечению доступа к ресурсам системы 47
2.8.3.Адаптеры безопасности для LDAP и аутентификации ADSI 48
2.8.4 Обеспечение Web-сервера Siebel 50
2.8.5. Web Single Sign-On. 52
3.Исследование влияния алгоритмов шифрования на качество информационных систем 57
3.1 Криптоанализ SIEBEL 59
3.2 Описание экспериментальной модели 59
3.3 Экспериментальная часть 60
3.4 Анализ результатов эксперимента по уязвимости 61
3.5 Анализ результатов эксперимента по эффективности от криптоаналитических атак 61

3.6 Анализ результатов эксперимента по эффективности различных видов атак от типа пароля 62
3.7.Анализ результатов эксперимента эффективности по времени 63
3.8 Анализ результатов эксперимента надежности 64
Приложение А 66
ПРИЛОЖЕНИЕ Б 69
Приложение В 71
Приложение Г 72
ЗАКЛЮЧЕНИЕ 75
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 77

АҢДАТПА

Диссертация зертте- және тиімді жолдың және ақпараттың ығының әдіс-айласының ізденісіне пайдалан- монополиялық рұқсат алуда ақпараттың(СЗИ) ығының жүйелерімен бағдарламалық пакеттер үшін арнаулы. Ара сапа зертте- және ықтың әзірле- әдісінің қолданысының нысанының жүйе ша взаимоотношению клиенттермен қызмет етеді.

Жұмыста түрлі-түрлі СЗИ бағдарламалық пакет және оның қолданысының мүмкіндігі жүйе үшін қарастырылады. СЗИ ықты және оның жаса- нұсқалары сұра- ұсынылатын бұлақты жерлері талданады. Артықшылық және бар нұсқаның міндері зерттеледі және ықтың жүзеге асуының жаңа жолдары үшін СЗИ монополиялық режимде пайдалан- ұсынылады.

АННОТАЦИЯ

Диссертация посвящена исследованию и поиску эффективных путей и способов защиты информации используемой в монопольном доступе системами защиты информации (СЗИ) для программных пакетов. В качестве объекта исследования и применения разработанных методов защиты служит система по взаимоотношению с клиентами.

В работе рассматриваются разнообразные СЗИ программных пакетов и возможность их применения для системы. Анализируются ключевые места СЗИ, требующие защиты и предлагаются варианты ее осуществления. Исследуются преимущества и недостатки существующих вариантов и предлагаются новые пути реализации защиты для СЗИ используемых в монопольном режиме.

ANNOTATION

Dissertation is sanctified to research and search of effective ways and methods of defence of information by used in monopolistic access by the systems of defence of information(SDI) for programmatic packages. As an object of research and application of the worked out methods of defence the system serves on interrelation with clients.

Various SDI of programmatic packages and possibility of their application is in-process examined for the system. The key places of SDI, requiring defence and the variants of her realization are offered, are analysed. Advantages and lacks of existent variants are investigated and the new ways of realization of defence are offered for SDI used in the burst mode.

ВВЕДЕНИЕ

В настоящее время основным способом защиты информации от несанкционированного доступа (НСД) является внедрение так называемых средств А А А (Authentication, Authorization, Accounting — аутентификация, авторизация, управление правами пользователей). При использовании этой технологии пользователь получает доступ к компьютеру лишь после того, как успешно прошел процедуры идентификации и аутентификации. Стоит учесть , что на мировом рынке ИТ-услуг сегмент А А А постоянно растет. Эта тенденция подчеркивается в аналитических обзорах IDC, Gartner и других консалтинговых фирм. Такой же вывод можно сделать, внимательно просмотрев ежегодный обзор компьютерной преступности Института компьютерной безопасности США и ФБР за 2005 год.

Как видно из диаграммы, ущерб от краж и конфиденциальной информации значительно увеличился. В следствии каждая из опрошенных компаний потеряла в среднем более 350 тыс. долл. вследствие краж и конфиденциальной информации. Это исследование подтверждает тенденции, наметившиеся в последние несколько лет. Согласно отчету Института компьютерной безопасности США и ФБР за 2004 год, кража чувствительных данных уже тогда входила в число опаснейших угроз — ущерб от нее составлял около 40% от общего объема ущерба всех его угроз. При этом средний объем потерь был равен более 300 тыс. долл., а максимальный объем — 1,5 млн. долл. Исходя этого, можно сделать вывод, что кража конфиденциальной информации имеет один из наиболее высок их рейтингов среди всех ИТ-у гроз в США . Стоит отметить, что найти виновного без решения вопросов идентификации и аутентификации невозможно!

Отметим, что вопросы разграничения доступа решаются в обязательном порядке при создании любой информационной системы. В наше время, когда системы становятся все более распределенными, трудно переоценить важность корректного разграничения доступа. При этом требуется все более надежная защита систем аутентификации, как от внешних, так и от внутренних злоумышленников. Стоит понимать, что пользователи не склонны усложнять себе жизнь и стараются пользоваться как можно менее сложными паролями. А следовательно, для устранения этого в дальнейшем все чаще будут применяться программно-аппаратные средства аутентификации, которые постепенно придут на смену традиционным паролям.

Актуальность темы обусловлена необходимостью поиска новых способов шифрования и аутентификации, не требующих больших временных затрат и являющихся наиболее крипто стойкими и безопасными. Работа обусловлена тем, что с развитием компьютерных технологий в образовательном процессе появилась необходимость в создании

эффективных систем, самоконтроля, внешнего контроля и защиты информации.

Цель проекта. Исследование анализа методов защиты информации с использованием вспомогательных аппаратных средств и создание интегрируемого пакета программных модулей для защиты систем функционирующих в монопольном режиме вне доверенной вычислительной среды.

Задачи исследования. Изучить особенности различных алгоритмов;
Выделены основные ключевые объекты, подлежащие защите;
Разработаны методы защиты вне доверенной вычислительной среды;
Проведен анализ и предложены возможные способы применения разработанных методов;
На основе данных методов разработан набор программных модулей защиты, предназначенных для интегрирования в систему.

Методы исследования. Экспериментальный метод исследования работы алгоритма основан на виртуализированной модели, хранящих данные учетных записей с различными исходными параметрами.

Объект исследования. Oracle Siebel CRM - это система по управлению взаимоотношениями с клиентами, разработанная корпорацией Siebel Systems, которую приобрела корпорация Oracle в 2006 году.

Предмет исследования. Криптографические алгоритмы: Аутентификация на уровне Базы Данных, аутентификация через LDAP/ADSI, аутентификация через Web Single Sign-On, аутентификация через Security SDK.

Гипотеза. Используемые на сегодня методы шифрования данных учетных записей на серверах не обеспечивают надежность на необходимом уровне. Если использовать дополнительное шифрование конечных сумм данных, то можно добиться высокого уровня защищенности при незначительных затратах времени и мощности, используемых на вторичное шифрование данных.

Новизна или личный вклад. На данный момент исследования в сфере шифрования аутентификационных данных фактически направлены на создание новых алгоритмов или протоколов передачи информации. Фактически исследования возможности использования вторичного шифрования не ведутся. Модель, которую можно было бы использовать в исследовании криптостойкости алгоритмов шифрования, не существует. Данная работа доказывает целесообразность использования дополнительного шифрования информации. Созданная модель позволяет продолжить исследования в данной сфере.

Практическая значимость. Исследование позволяет выявить слабые стороны используемой на сегодня технологий аутентификаций, демонстрирует возможности вторичного шифрования, позволяющего значительно повысить криптографическую стойкость зашифрованной

информации, обеспечить защищенность учетных записей пользователей в различных информационных системах.

1. ОБЗОР И АНАЛИЗ СОВРЕМЕННЫХ АЛГОРИТМОВ ШИФРОВАНИЯ

1.1 Шифрование данных

Проблема защиты информации путем ее преобразования, исключающего ее прочтение посторонним лицом, волновала человеческий ум с давних времен. История криптографии - ровесница истории человеческого языка. Более того, первоначально письменность сама по себе была криптографической системой, так как в древних обществах ею владели только избранные.

Бурное развитие криптографические системы получили в годы первой и второй мировых войн. Появление вычислительных средств в послевоенные годы ускорило разработку и совершенствование криптографических методов. Вообще история криптографии крайне увлекательна, и достойна отдельного рассмотрения. В качестве хорошей книги по теме криптографии можно рекомендовать "Основы современной криптографии" Баричев С. Г. [32].

Почему проблема использования криптографических методов в информационных системах (ИС) стала в настоящий момент особо актуальна?

С одной стороны, расширилось использование компьютерных сетей, в частности, глобальной сети Интернет, по которым передаются большие объемы информации государственного, военного, коммерческого и частного характера, не допускающего возможность доступа к ней посторонних лиц.

С другой стороны, появление новых мощных компьютеров, технологий сетевых и нейронных вычислений сделало возможным дискредитацию криптографических систем, еще недавно считавшихся практически нераскрываемыми.

Все это постоянно подталкивает исследователей на создание новых криптосистем и тщательный анализ уже существующих.

Проблемой защиты информации путем ее преобразования занимается криптология. Криптология разделяется на два направления – криптографию и криптоанализ. Цели этих направлений прямо противоположны.

Криптография занимается поиском и исследованием методов преобразования информации с целью скрытия ее содержания.

Сфера интересов криптоанализа - исследование возможности расшифровывания информации без знания ключей.

Современная криптография разделяет их на четыре крупных класса.

1. Симметричные криптосистемы.
2. Криптосистемы с закрытым ключом.
3. Системы электронной цифровой подписи (ЭЦП).
4. Системы управление ключами.

Основные направления использования криптографических методов – передача конфиденциальной информации по каналам связи (например, электронная почта), установление подлинности передаваемых сообщений,

хранение информации (документов, баз данных) на носителях в зашифрованном виде.

Итак, криптография дает возможность преобразовать информацию таким образом, что ее прочтение (восстановление) возможно только при знании ключа.

Приведем определения некоторых основных терминов, используемых в криптографии.

Алфавит - конечное множество используемых для кодирования информации знаков.

Текст - упорядоченный набор из элементов алфавита.

В качестве примеров алфавитов, используемых в современных ИС можно привести следующие:

- алфавит Z_{33} – 32 буквы русского алфавита (исключая "ё") и пробел;
- алфавит Z_{256} – символы, входящие в стандартные коды ASCII и КОИ-8;
- двоичный алфавит - $Z_2 = \{0,1\}$;
- восьмеричный или шестнадцатеричный алфавит.

Шифрование – процесс преобразования исходного текста, который носит также название открытого текста, в шифрованный текст.

Расшифрование – процесс, обратный шифрованию. На основе ключа шифрованный текст преобразуется в исходный.

Криптографическая система представляет собой семейство T преобразований открытого текста. Члены этого семейства индексируются, или обозначаются символом k; параметр k обычно называется ключом. Преобразование T_k определяется соответствующим алгоритмом и значением ключа k.

Ключ – информация, необходимая для беспрепятственного шифрования и расшифрования текстов.

Пространство ключей K – это набор возможных значений ключа.

Криптосистемы подразделяются на симметричные и асимметричные (или с закрытым ключом).

В симметричных криптосистемах для шифрования, и для расшифрования используется один и тот же ключ.

В системах с закрытым ключом используются два ключа - открытый и закрытый (секретный), которые математически связаны друг с другом. Информация шифруется с помощью открытого ключа, который доступен всем желающим, а расшифровывается с помощью закрытого ключа, известного только получателю сообщения.

Термины распределение ключей и управление ключами относятся к процессам системы обработки информации, содержанием которых является выработка и распределение ключей между пользователями.

Электронной цифровой подписью называется присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и подлинность сообщения.

Криптостойкостью называется характеристика шифра, определяющая его стойкость к расшифрованию без знания ключа (т.е. криптоанализу). Имеется несколько показателей криптостойкости, среди которых:

- количество всех возможных ключей;
- среднее время, необходимое для успешной криптоаналитической атаки того или иного вида.

Эффективность шифрования с целью защиты информации зависит от сохранения тайны ключа и криптостойкости шифра.

1.2. Асимметричные криптосистемы

Теперь остановимся на асимметричные криптосистемам и кратко расскажем о них. Связано это с тем, что в дальнейшем в системе защиты будет предложен и использован механизм, построенный по принципу асимметричных криптосистем.

Асимметричные или двухключевые системы являются одним из обширным классом криптографических систем. Эти системы характеризуются тем, что для шифрования и для расшифрования используются разные ключи, связанные между собой некоторой зависимостью. При этом данная зависимость такова, что установить один ключ, зная другой, с вычислительной точки зрения очень трудно.

Один из ключей (например, ключ шифрования) может быть сделан общедоступным, и в этом случае проблема получения общего секретного ключа для связи отпадает. Если сделать общедоступным ключ расшифрования, то на базе полученной системы можно построить систему аутентификации передаваемых сообщений. Поскольку в большинстве случаев один ключ из пары делается общедоступным, такие системы получили также название криптосистем с закрытым ключом.

Криптосистема с закрытым ключом определяется тремя алгоритмами: генерации ключей, шифрования и расшифрования. Алгоритм генерации ключей открыт, всякий может подать ему на вход случайную строку r надлежащей длины и получить пару ключей (k_1, k_2). Один из ключей (например, k_1) публикуется, он называется закрытым, а второй, называемый секретным, хранится в тайне. Алгоритмы шифрования E_{k_1} и расшифрования D_{k_2} таковы, что для любого открытого текста m $D_{k_2}(E_{k_1}(m)) = m$.

Рассмотрим теперь гипотетическую атаку злоумышленника на эту систему. Противнику известен открытый ключ k_1, но неизвестен соответствующий секретный ключ k_2. Противник перехватил криптограмму d и пытается найти сообщение m, где $d = E_{k_1}(m)$. Поскольку алгоритм шифрования открыт, противник может просто последовательно перебрать все возможные сообщения длины n, вычислить для каждого такого сообщения m_i криптограмму $d_i = E_{k_1}(m_i)$ и сравнить d_i с d. То сообщение, для которого d_i = d и будет искомым закрытым текстом. Если повезет, то открытый текст будет найден достаточно быстро. В худшем же случае

перебор будет выполнен за время порядка $2^n T(n)$, где $T(n)$ – время, требуемое для шифрования сообщения длины n. Если сообщения имеют длину порядка 1000 битов, то такой перебор неосуществим на практике ни на каких самых мощных компьютерах.

Мы рассмотрели лишь один из возможных способов атаки на криптосистему и простейший алгоритм поиска открытого текста, называемый обычно алгоритмом полного перебора. Используется также и другое название: «метод грубой силы». Другой простейший алгоритм поиска открытого текста – угадывание. Этот очевидный алгоритм требует небольших вычислений, но срабатывает с пренебрежимо малой вероятностью (при больших длинах текстов). На самом деле противник может пытаться атаковать криптосистему различными способами и использовать различные, более изощренные алгоритмы поиска открытого текста.

1.2.1. Криптосистема Эль-Гамеля

Система Эль-Гамеля – это криптосистема с закрытым ключом, основанная на проблеме логарифма. Система включает алгоритм цифровой подписи.

Множество параметров системы включает простое число p и целое число g, степени которого по модулю p порождают большое число элементов Z_p. У пользователя A есть секретный ключ a и открытый ключ y, где $y = g^a \pmod p$. Предположим, что пользователь B желает послать сообщение m пользователю A. Сначала B выбирает случайное число k, меньшее p. Затем он вычисляет

$y_1 = a^k \pmod e$ и $y_2 = m \oplus (y^k \pmod e)$,

где $\oplus$ обозначает побитовое "исключающее ИЛИ". B посылает A пару (y_1, y_2).

После получения шифрованного текста пользователь A вычисляет

$m = (y_1^a \bmod e) \oplus y_2$.

Известен вариант этой схемы, когда операция $\oplus$ заменяется на умножение по модулю e. Это удобнее в том смысле, что в первом случае текст (или значение хэш-функции) необходимо разбивать на блоки той же длины, что и число $y^k \pmod p$. Во втором случае этого не требуется и можно обрабатывать блоки текста заранее заданной фиксированной длины (меньшей, чем длина числа e). Если Вы осуществляете внешнюю систему идентификации, то добавление пользователя к базе данных, ли само регистрацией или администратором, могло бы или не могло бы размножить данные о логине пользователя к внешней системе идентификации. Если верительные грамоты логина не размножаются к системе идентификации, то Вы должны создать верительные грамоты логина отдельно в системе идентификации.

Если Вы осуществляете идентификацию базы данных, то добавления пользователя к базе данных, с идентификатором пользователя и паролем, достаточно, чтобы позволить этому пользователю быть заверенным. Для

получения дополнительной информации об идентификации и распространении пользовательских данных.

1.2.2. Криптосистема Ревеста-Шумира-Эйдолмана

Система Ривеста-Шумира-Эйделмана (Rivest, Shamir, Adleman – RSA) представляет собой криптосистему, стойкости которой основана на сложности решения задачи разложения числа на простые сомножители. Кратко алгоритм можно описать следующим образом:

Пользователь А выбирает пару различных простых чисел p_A и q_A , вычисляет $n_A = p_A q_A$ и выбирает число d_A, такое что НОД(d_A, $\varphi(n_A)$) = 1, где φ (n) – функция Эйлера (количество чисел, меньших n и взаимно простых с n. Если n = pq, где p и q – простые числа, то $\varphi(n) = (p - 1)(q - 1)$). Затем он вычисляет величину e_A, такую, что $d_A \cdot e_A = 1 \pmod{\varphi(n_A)}$, и размещает в общедоступной справочной таблице пару (e_A, n_A), являющуюся закрытым ключом пользователя А.

Теперь пользователь В, желая передать сообщение пользователю А, представляет исходный текст

$x = (x_0, x_1, ..., x_{n-1})$, $x \in Z_n$, $0 \le i < n$,

по основанию n_A:

$N = c_0 + c_1 n_A + ...$

Пользователь В зашифровывает текст при передаче его пользователю А, применяя к коэффициентам c_i отображение E_{e_A, n_A} :

$$E_{e_A, n_A} : c \longrightarrow c^{e_A} \pmod{n_A},$$

получая зашифрованное сообщение N'. В силу выбора чисел d_A и e_A, отображение E_{e_A, n_A} является взаимно однозначным, и обратным к нему будет отображение

$$E_{d_A, n_A} : c \longrightarrow c^{d_A} \pmod{n_A}$$

Пользователь А производит расшифрование полученного сообщения N', применяя E_{d_A, n_A} .

Для того чтобы найти отображение E_{d_A, n_A} , обратное по отношению к E_{e_A, n_A} , требуется знание множителей $n_A = p_A q_A$. Время выполнения наилучших из известных алгоритмов разложения при $n > 10^{145}$ на сегодняшний день выходит за пределы современных технологических возможностей. Управление доступом - термин, использованный, чтобы описать набор механизмов заявления Siebel, которые управляют пользовательским доступом к прикладной функциональности и данным. Поскольку Вы работаете с этой главой, определяете, как терминология и понятия, представленные здесь, соответствуют внутренней терминологии и структуре Вашей компании. Эта глава объясняет механизмы доступа Siebel, но Вы должны решить во время перспективного проектирования, как объединить механизмы, чтобы удовлетворить Ваши потребности бизнеса и безопасности.

В условиях заявления Siebel экран представляет широкую область функциональности, такой как работа над счетами. Набор экранов, к которым у пользователя есть доступ, убежден заявлениями, что Ваша компания купила. Каждый экран представлен как счет наверху окна. В примере ниже, показан экран Accounts.

Каждый экран содержит многократные взгляды, чтобы обеспечить различные виды доступа к данным. Пользователю представление - просто Веб-страница. В пределах представления пользователь мог бы видеть списки записей данных или форм, представляя отдельные или многократные отчеты, и иногда детские отчеты. (Эти списки и формы упоминаются как апплеты в контексте конфигурации.) Каждое представление (или группировка взглядов) представлено текстом в баре связи ниже счетов экрана.

Например, рисунок 5 показывает Представление Списка Счета, которое соответствует названию апплета Мои Счета (текущий выбор фильтра видимости). Многократные способы представления обеспечивают доступ к другим представлениям, которые фильтруют данные по-другому. В Представлении Списка Счета действующий пользователь может рассмотреть счета, принадлежавшие или назначенные на этого пользователя. Выбор Всех Счетов от фильтра видимости показывают Все Представление Списка Счета вместо этого, предположение, что у пользователя есть доступ к этому представлению.

1.3. Адаптированный метод асимметричного шифрования

Рассмотренные ранее методы построения асимметричных алгоритмов криптопреобразований хоть и интересны, но не достаточно хорошо подходят для решаемой задачи. Можно было бы взять реализацию уже готового асимметричного алгоритма, или согласно теоретическому описанию, реализовать его самостоятельно. Но, во-первых, здесь встает вопрос о лицензировании и использовании алгоритмов шифрования. Во-вторых, использование стойких крипто алгоритмов связано с правовой базой, касаться которой бы не хотелось. Сам по себе стойкий алгоритм шифрования здесь не нужен. Он просто излишен и создаст лишь дополнительное замедление работы программы при шифровании данных. Также планируется выполнять код шифрования в виртуализированной машине, из чего вытекают большие трудности реализации такой системы, если использовать сложные алгоритмы шифрования. Виртуальная машина дает ряд преимуществ, например, делает более труднодоступной возможность проведения некоторых операций. В качестве примера можно привести проверку алгоритмом допустимого срока своего использования.

Отсюда следует вывод, что создаваемый алгоритм шифрования должен быть достаточен прост. Но при этом он должен обеспечивать асимметричность и быть достаточно сложным для анализа. Поэтому исходя этих позиций берет свое начало идея создания полиморфных алгоритмов шифрования.

Основная сложность будет состоять в построении генератора, который должен выдавать на выходе два алгоритма. Один – для шифрования, другой – для расшифрования. Ключей у этих алгоритмов шифрования/расшифрования нет. Можно сказать, что они сами являются ключами, или что они содержат ключ внутри. Они должны быть устроены таким образом, чтобы производить уникальные преобразования над данными. В следствии два сгенерированных алгоритма шифрования должны производить шифрования абсолютно различными способами. И для их расшифровки возможно будет использовать только соответствующий алгоритм расшифрования, который был сгенерирован в паре с алгоритмом шифрования.

Уникальность создания таких алгоритмов должен обеспечить полиморфный генератор кода. Выполняться такие алгоритмы будут в виртуализированной машине. Анализ таких алгоритмов должен стать весьма трудным и нецелесообразным занятием.

Преобразования над данными будут достаточно тривиальны, но практически, вероятность генерации двух одинаковых алгоритмов должна стремиться к нулю. В качестве элементарных действий следует использовать такие не ресурсоемкие операции, как сложение с каким-либо числом или, например, побитовое "исключающее или" (XOR). Но повторение нескольких таких преобразований с изменяющимися аргументами операций (в зависимости от адреса шифруемой ячейки) делает шифр достаточно сложным. Генерации каждый раз новой последовательности таких преобразований с участием различных аргументов усложняет анализ алгоритма. Вы можете связать единственное положение к отдельным данным. Например, с Моей точки зрения Кавычек, сотрудник, загруженный, используя особое положение, видит только кавычки, связанные с тем положением. Другое представление, которое применяет управление доступом единственного положения, является Моими Прогнозами.

Слово Мой часто находится в названиях взглядов, применяющих управление доступом единственного положения. Однако Мой не всегда подразумевает управление доступом единственного положения. Некоторые Мои взгляды применяются личный, организация или управление доступом команды. Например, Моя точка зрения Действий применяет личное управление доступом.

Способы представления делового компонента определяют, может ли управление доступом единственного положения быть применено в представлении, которое основано на деловом компоненте. Чтобы иметь управление доступом единственного положения в наличии, у делового компонента должен быть способ представления (обычно, Торговый представитель) владельца печатают Положение с входом в колонке Области Видимости (вместо Видимости колонка MVField). Для получения информации о деловых составляющих способах представления посмотрите Способы Представления Компонента Бизнеса Просмотра. Для получения

информации об осуществлении управления доступом в представлении посмотрите Свойства Управления доступом Представления Листинга.

1.4. Преимущества применения полиморфных алгоритмов шифрования

К преимуществам применения полиморфных алгоритмов шифрования для систем, по функциональности схожим с АСДО, можно отнести следующие пункты:

- слабая очевидность принципа построения системы защиты;
- сложность создания универсальных средств для обхода системы защиты;
- легкая реализация системы асимметрического шифрования;
- возможность легкой, быстрой адаптации и усложнения такой системы;
- возможность расширения виртуализированной машины с целью сокрытия части кода.

Рассмотрим теперь каждый из этих пунктов по отдельности и обоснуем эти преимущества. Можно привести и другие удобства, связанные с использование полиморфных механизмов в алгоритмах шифрования. Но, на мой взгляд, перечисленные преимущества являются основными и заслуживающими внимания.

Слабая очевидность принципа построения системы защиты, является следствием выбора достаточно своеобразных механизмов. Во-первых, это само выполнение кода шифрования/расшифрования в виртуализированной машине. Во-вторых, наборы полиморфных алгоритмов, уникальных для каждого пакета защищаемого программного комплекса. Это должно повлечь серьезные затруднения при попытке анализа работы такой системы с целью поиска слабых мест для атаки. Если система сразу создаст видимость сложности и малой очевидности работы своих внутренних механизмов, то скорее всего это остановит человека от дальнейших исследований. Правильно построенная программа с использованием разрабатываемой системой защиты может не только оказаться сложной на вид, но и быть такой в действительности. Выбранные же методы сделают устройство данной системы нестандартным, и, можно сказать, неожиданным.

Сложность создания универсальных средств для обхода системы защиты заключается в возможности генерации уникальных пакетов защищенного ПО. Создание универсального механизма взлома средств защиты затруднено при отсутствии исходного кода. В противном случае необходим глубокий, подробный и профессиональный анализ такой системы, осложняемый тем, что каждая система использует свои алгоритмы шифрования/расшифрования. А модификация отдельного экземпляра защищенного интереса не представляет. Ведь основной упор сделан на защиту от ее массового взлома, а не на высокую надежность отдельного экземпляра пакета.

Легкая реализация системы асимметрического шифрования, хоть и является побочным эффектом, но очень полезна и важна. Она представляет

собой следствие необходимости генерировать два разных алгоритма, один для шифрования, а другой для расшифрования. На основе асимметрического шифрования можно организовать богатый набор различных механизмов в защищаемом программном комплексе. Примеры такого применения будут даны в других разделах данной работы.

Возможность легкой, быстрой адаптации и усложнения такой системы. Поскольку для разработчиков система предоставляется в исходном коде, то у него есть все возможности для его изменения. Это может быть вызвано необходимостью добавления новой функциональности. При этом для такой функциональности может быть реализована поддержка со стороны измененной виртуализированной машины. В этом случае работа новых механизмов может стать сложной для анализа со стороны. Также легко внести изменения с целью усложнения генератора полиморфного кода и увеличения блоков, из которых строятся полиморфные алгоритмы. Это, например, может быть полезно в том случае, если кем-то, не смотря на все сложности, будет создан универсальный пакет для взлома системы зашиты. Тогда совсем небольшие изменения в коде, могут свести на нет труды взломщика. Стоит отметить, что это является очень простым действием, и потенциально способствует защите, так как делает процесс создания взлома еще более нерациональным.

Поскольку программисту отдаются исходные коды система защиты, то он легко может воспользоваться существующей виртуализированной машиной и расширить ее для собственных нужд. То же самое касается и генератора полиморфных алгоритмов. Например, он может встроить в полиморфный код ряд специфической для его системы функций. Сейчас имеется возможность ограничить возможность использования алгоритмов по времени. А где-то, возможно, понадобится ограничение по количеству запусков. Можно расширить только виртуальную машину с целью выполнения в ней критических действий. Например, проверку результатов ответа. Выполнение виртуального кода намного сложнее для анализа, а, следовательно, расширяя механизм виртуализированной машины, можно добиться существенного повышения защищенности АСДО. К бизнес-процессам могут получить доступ все пользователи по умолчанию. Однако пользователь с логином администратора может ограничить доступ к указанным бизнес-процессам и может тогда связать обязанности с ограниченными бизнес-процессами или связать ограниченные бизнес-процессы с обязанностями. Это позволяет администратору ограничивать доступ к бизнес-процессам, основанным на ответственности конечного пользователя. Чтобы получить доступ к ограниченному бизнес-процессу, конечный пользователь должен быть связан с ответственностью, которая позволяет доступ к нему. Конечный пользователь, на которого возложена больше чем одна ответственность, может получить доступ к любому ограниченному бизнес-процессу, который связан с одной из его или ее обязанностей. Особая ответственность припряталась про запас, когда

пользователь загружается, кто несет ту ответственность. У пользователей есть доступ только к тем взглядам, которые были определены для применимых обязанностей в то время, когда они загрузились, даже при том, что дополнительные взгляды, возможно, были добавлены администратором с этого времени.

Если Вы добавляете, удаляете или изменяете ответственность в представлении Обязанностей (Представление Списка Обязанностей), то Вы можете очистить тайник, чтобы проинструктировать заявление Siebel прочитать обновленные ценности от базы данных. Прояснение тайника делает эти изменения доступными для пользователей, которые загружаются впоследствии или кто выходит из системы и загружается снова. Сервер Siebel не должен быть перезапущен.

1.5. Функциональность системы защиты

Ранее были рассмотрены цели, для которых разрабатывается система защиты, а также методы, с использованием которых эта система будет построена. Сформулируем функции системы защиты, которые она должна будет предоставить программисту.

1. Генератор полиморфных алгоритмов шифрование и расшифрования.
2. Виртуальная машина в которой могут исполняться полиморфные алгоритмы. Отметим также, что виртуальная машина может быть легко адаптирована, с целью выполнения программ иного назначения.
3. Асимметричная система шифрования данных.
4. Ограничение использования полиморфных алгоритмов по времени.
5. Защита исполняемых файлов от модификации.
6. Контроль за временем возможности запуска исполняемых файлов.
7. Поддержка таблиц соответствий между именами зашифрованных файлов и соответствующих им алгоритмам шифрования/расшифрования.
8. Упаковка шифруемых данных.

1.6. Описание показателей алгоритмов шифрования

Результаты проделанной работы по изучению алгоритмов-финалистов NIST сформулировал в виде отчета. Данный отчет содержит как результаты анализа алгоритмов, так и обоснование критериев, по которым выполнялась оценка. Сформулируем функции системы защиты, которые она должна будет предоставить программисту.

На основе отчета кратко сформулировал сравнительные оценки пяти алгоритмов-финалистов конкурса AES по основным критериям в виде следующей таблицы:

Таблица 1.1

№	Категория	Serpent	Twofish	MARS	RC6	Rijndael
1	Криптостойкость	+	+	+	+	+
2	Запас криптостойкости	++	++	++	+	+
3	Скорость шифрования при программной реализации	-	±	±	+	+
4	Скорость расширения ключа при программной реализации	±	-	±	±	+
5	Смарт-карты с большим объемом ресурсов	+	+	-	±	++
6	Смарт-карты с ограниченным объемом ресурсов	±	+	-	±	++
7	Аппаратная реализация (ПЛИС)	+	+	-	±	+
8	Аппаратная реализация (специализированная микросхема)	+	±	-	-	+
9	Защита от атак по времени выполнения и потребляемой мощности[3]	+	±	-	-	+
10	Защита от атак по потребляемой мощности на процедуру расширения ключа	±	±	±	±	-
11	Защита от атак по потребляемой мощности на реализации в смарт-картах	±	+	-	±	+
12	Возможность расширения ключа «на лету»	+	+	±	±	±
13	Наличие вариантов реализации (без потерь в совместимости)	+	+	±	±	+
14	Возможность параллельных вычислений	±	±	±	±	+

1.7 Влияние алгоритмов шифрования на качество информационных систем

1. Криптостойкость всех алгоритмов-финалистов оказалась достаточной — в процессе исследований не было обнаружено каких-либо реально реализуемых атак на полноценные и полнораундовые версии алгоритмов. В данном случае криптоаналитики обычно исследуют варианты алгоритмов с усеченным числом раундов, либо с некоторыми внесенными изменениями, незначительными, но ослабляющими алгоритм. Под запасом криптостойкости (security margin) эксперты NIST подразумевают соотношение полного (предусмотренного в спецификациях алгоритмов)

числа раундов и максимального из тех вариантов, против которых действуют какие-либо криптоаналитические атаки. Например, с помощью дифференциально-линейного криптоанализа вскрывается 11-раундовый Serpent , тогда как в оригинальном алгоритме выполняется 32 раунда. Эксперты NIST в отчете предупредили, что данная оценка является весьма поверхностной и не может быть значимой при выборе алгоритма-победителя конкурса, но, тем не менее, отметили, что запас криптостойкости у Rijndael и RC6 несколько ниже, чем у остальных алгоритмов-финалистов.

2. В пп. 5-8 приведена сравнительная оценка возможности и эффективности реализации алгоритмов в перечисленных устройствах.
3. В пп. 9-11 имеется в виду, насколько операции, выполняемые конкретным алгоритмом, могут быть подвержены анализу указанным методом. При этом принималось в расчет то, что операции могут быть модифицированы с целью усложнения криптоанализа за счет потери в скорости шифрования (например, проблемное в этом смысле вращение на переменное число бит может принудительно выполняться за равное число тактов — В следствиимаксимально возможное для данной операции; поэтому подобные меры противодействия атакам по времени исполнения и потребляемой мощности рекомендует их изобретатель Пол Кохер (Paul C. Kocher).
4. Из описаний алгоритмов видно, что все они поддерживают расширение ключа «на лету» (В следствииподключи могут генерироваться непосредственно в процессе шифрования — по мере необходимости), однако, только Serpent и Twofish поддерживают такую возможность без каких-либо ограничений.
5. Под наличием вариантов реализации (implementation flexibility) имеется в виду возможность различным образом реализовывать какие-либо операции алгоритма с оптимизацией под конкретные цели. Наиболее показательными в этом смысле являются упомянутые ранее варианты процедуры расширения ключа алгоритма Twofish , позволяющие оптимизировать реализацию алгоритма в зависимости, прежде всего, от частоты смены ключа.

В данном конфигурационном файле некоторые параметры не могли бы появиться по умолчанию. Другие могли бы появиться с предыдущей точкой с запятой (;), указывая, что параметр - комментарий и не интерпретируется. Точка с запятой должна быть удалена, чтобы сделать параметр активным. Изменения прикладного конфигурационного файла не активны, пока Вы не перезапускаете клиента Сервера или Siebel Siebel. Для получения дополнительной информации о работе с конфигурационными файлами, посмотрите Гида Системного администрирования Siebel. Следующие параметры расположены в [LDAPSecAdpt] или секция [ADSISecAdpt] (или эквивалентные) прикладного конфигурационного файла, согласно тому, формируете ли Вы адаптер безопасности LDAP или адаптер безопасности

ADSI. Каждый связанный с идентификацией параметр в конфигурационном файле применения интерпретируется адаптером безопасности (для LDAP или идентификации ADSI).

Некоторые параметры применяются только к внедрениям LDAP, или только к внедрениям ADSI. Некоторые параметры применяют только в Сети окружающую среду идентификации SSO. Для получения дополнительной информации см. описания для эквивалентных параметров, применимых к Веб-Клиенту Siebel и другим контекстам идентификации в Параметрах Сервера Имени Ворот Siebel.

Сформулируем основные достоинства и недостатки каждого из рассмотренных в данной статье алгоритмов-финалистов :

2 ИДЕНТИФИКАЦИЯ И АУТЕНТИФИКАЦИЯ

2.1 Основные понятия

Идентификацию и аутентификацию можно считать основой программнотехнических средств безопасности, поскольку остальные сервисы рассчитаны на обслуживание именования субъектов. Идентификация и аутентификация – это первая линия обороны, "проходная" информационного пространства организации.

Идентификация позволяет объекту (пользователю, процессу, действующему от имени определенного пользователя, или иному аппаратно-программному компоненту) назвать себя (сообщить свое имя). Посредством аутентификация вторая сторона убеждается, что объект действительно тот, за кого он себя выдает. В качестве синонима слова "аутентификация" используют словосочетание "проверка подлинности".

Происхождение русскоязычного термина "аутентификация" не совсем понятно. Английское "authentication" скорее можно прочитать как "аутентикация": трудно сказать, откуда в середине взялось еще "фи" – может, из идентификации? Тем не менее, термин устоялся, он закреплен в Руководящих документах гос. норма контроля Казахстана, использован в многочисленных публикациях, поэтому исправить его уже невозможно.

Аутентификация бывает односторонней (обычно клиент доказывает свою подлинность серверу) и двусторонней (взаимной). Пример односторонней аутентификации – процедура входа пользователя в систему.

В сетевой среде, когда стороны идентификации/аутентификации территориально разнесены, у рассматриваемого сервиса есть два основных аспекта:

– что служит аутентификатором (В следствииисполь зуется для подтверждения подлинности субъекта);

– как организован (и защищен) обмен данными идентификации/аутентификации.

Субъект может подтвердить свою подлинность, предъявив по крайней мере одну из следующих сущностей:

– нечто, что он знает (пароль, личный идентификационный номер, криптографический ключ и т.п.);

– нечто, чем он владеет (личную карточку или иное устройство аналогичного назначения);

– нечто, В следствии часть его самого (голос, отпечатки пальцев и т.п., В следствии свои биометрические характеристики).

Надежная идентификация затруднена не только из-за сетевых угроз, но и по целому ряду причин. Во-первых, почти все аутентификационные сущности можно узнать, украсть или подделать. Во-вторых, имеется противоречие между надежностью аутентификации, с одной стороны, и удобствами пользователя и системного администратора с другой. Так, из соображений безопасности необходимо с определенной частотой просить пользователя повторно вводить аутентификационную информацию (ведь на его место мог сесть другой человек), а это не только хлопотно, но и повышает вероятность того, что кто-то может подсмотреть за вводом данных. В-третьих, чем надежнее средство защиты, тем оно дороже.

Структура описывает пользовательские отчеты не сотрудника, предоставленные как данные о семени. Пароли по умолчанию не обеспечены для этих отчетов. Если Вы используете пользовательский отчет семени в качестве анонимного пользовательского отчета, то Вы должны установить параметр AnonUserName на идентификатор пользователя семени (например, GUESTCST), формируя SWSE, или устанавливать его вручную в eapps.cfg файле. Для получения информации о формировании SWSE см. Инструкцию по установке Siebel для операционной системы, которую Вы используете. Для получения информации о ручном урегулировании паролей для анонимного пользователя посмотрите Зашифрованные Пароли в eapps.cfg Файле.

Современные средства идентификации/аутентификации должны поддерживать концепцию единого входа в сеть. Единый вход в сеть – это, в первую очередь, требование удобства для пользователей. Если в корпоративной сети много информационных сервисов, допускающих независимое обращение, то многократная идентификация/аутентификация становится слишком обременительной. К сожалению, пока нельзя сказать, что единый вход в сеть стал нормой, доминирующие решения пока не сформировались.

Таким образом, необходимо искать компромисс между надежностью, доступностью по цене и удобством использования и администрирования средств идентификации и аутентификации.

Любопытно отметить, что сервис идентификации/аутентификации может стать объектом атак на доступность. Если система сконфигурирована так, что после определенного числа неудачных попыток устройство ввода идентификационной информации (такое, например, как терминал) блокируется, то злоумышленник может остановить работу легального пользователя буквально несколькими нажатиями клавиш [2].

2.2 Парольная аутентификация

Главное достоинство парольной аутентификации – простота и привычность. Пароли давно встроены в операционные системы и иные сервисы. При правильном использовании пароли могут обеспечить приемлемый для многих организаций уровень безопасности. Тем не менее, по совокупности характеристик их следует признать самым слабым средством проверки подлинности.

Чтобы пароль был запоминающимся, его зачастую делают простым (имя подруги, название спортивной команды и т.п.). Однако простой пароль нетрудно угадать, особенно если знать пристрастия данного пользователя. Известна классическая история про советского разведчика Рихарда Зорге, объект внимания которого через слово говорил "карамба"; разумеется, этим же словом открывался сверхсекретный сейф.

Иногда пароли с самого начала не хранятся в тайне, так как имеют стандартные значения, указанные в документации, и далеко не всегда после установки системы производится их смена.

Ввод пароля можно подсмотреть. Иногда для подглядывания используются даже оптические приборы.

Пароли нередко сообщают коллегам, чтобы те могли, например, подменить на некоторое время владельца пароля. Теоретически в подобных случаях более правильно задействовать средства управления доступом, но на практике так никто не поступает; а тайна, которую знают двое, это уже не тайна.

Пароль можно угадать "методом грубой силы", используя, скажем, словарь. Если файл паролей зашифрован, но доступен для чтения, его можно скачать к себе на компьютер и попытаться подобрать пароль, запрограммировав полный перебор (предполагается, что алгоритм шифрования известен).

Тем не менее, следующие меры позволяют значительно повысить надежность парольной защиты:

- наложение технических ограничений (пароль должен быть не слишком коротким, он должен содержать буквы, цифры, знаки пунктуации и т.п.);
- управление сроком действия паролей, их периодическая смена;
- ограничение доступа к файлу паролей;
- ограничение числа неудачных попыток входа в систему (это затруднит применение "метода грубой силы");
- обучение пользователей;
- использование программных генераторов паролей (такая программа, основываясь на несложных правилах, может порождать только благозвучные и, следовательно, запоминающиеся пароли).

2.3 Одноразовые пароли

Рассмотренные выше пароли можно назвать многоразовыми; их раскрытие позволяет злоумышленнику действовать от имени легального пользователя. Гораздо более сильным средством, устойчивым к пассивному прослушиванию сети, являются одноразовые пароли.

Наиболее известным программным генератором одноразовых паролей является система S/KEY компании Bellcore. Идея этой системы состоит в следующем. Пусть имеется односторонняя функция f (В следствиифункция, вычислить обратную, которой за приемлемое время не представляется возможным). Эта функция известна и пользователю, и серверу аутентификации. Пусть, далее, имеется секретный ключ K, известный только пользователю.

На этапе начального администрирования пользователя функция f применяется к ключу K n раз, после чего результат сохраняется на сервере. После этого процедура проверки подлинности пользователя выглядит следующим образом:

- сервер присылает на пользовательскую систему число (n-1);
- пользователь применяет функцию f к секретному ключу K (n-1) раз и отправляет результат по сети на сервер аутентификации;
- сервер применяет функцию f к полученному от пользователя значению и сравнивает результат с ранее сохраненной величиной. В случае совпадения подлинность пользователя считается установленной, сервер запоминает новое значение (присланное пользователем) и уменьшает на единицу счетчик (n).

На самом деле реализация устроена чуть сложнее (кроме счетчика, сервер посылает затравочное значение, используемое функцией f), но для нас сейчас это не важно. Поскольку функция f необратима, перехват пароля, равно как и получение доступа к серверу аутентификации, не позволяют узнать секретный ключ K и предсказать следующий одноразовый пароль.

Система S/KEY имеет статус Internet-стандарта (RFC 1938).

Другой подход к надежной аутентификации состоит в генерации нового пароля через небольшой промежуток времени (например, каждые 60 секунд), для чего могут использоваться программы или специальные интеллектуальные карты (с практической точки зрения такие пароли можно считать одноразовыми). Сервер аутентификации должен быть известен алгоритм генерации паролей и ассоциированные с ним параметры; кроме того, часы клиента и сервера должны быть синхронизированы [2].

2.4 Сервер аутентификации Kerberos

Kerberos – это программный продукт, разработанный в середине 1980-х годов в Массачусетском технологическом институте и претерпевший с тех пор ряд принципиальных изменений. Клиентские компоненты Kerberos присутствуют в большинстве современных операционных систем.

Kerberos предназначен для решения следующей задачи. Имеется открытая (незащищенная) сеть, в узлах которой сосредоточены субъекты – пользователи, а также клиентские и серверные программные системы.

Каждый субъект обладает секретным ключом. Чтобы субъект C мог доказать свою подлинность субъекту S (без этого S не станет обслуживать C), он должен не только назвать себя, но и продемонстрировать знание секретного ключа. C не может просто послать S свой секретный ключ, во-первых, потому, что сеть открыта (доступна для пассивного и активного прослушивания), а, во-вторых, потому, что S не знает (и не должен знать) секретный ключ C. Требуется менее прямолинейный способ демонстрации знания секретного ключа.

Система Kerberos представляет собой доверенную третью сторону (В следствиисторону, которой доверяют все), владеющую секретными ключами обслуживаемых субъектов и помогающую им в попарной проверке подлинности. Об осуществлении сети идентификация SSO Внедрение Сети идентификация SSO является тем же самым для приложений Финансовых услуг Siebel, как описано в других темах в этом гиде со следующими исключениями.

О подготовке сети SSO ответственность и новая ответственность, которые возложены на семя анонимный пользователь GUESTCST, предназначены для использования с потребительскими заявлениями Финансовых услуг Siebel. Эти обязанности отличаются от обязанностей, возложенных на GUESTCST для потребительских заявлений Siebel, которые не являются определенными для финансовых услуг, как зарегистрировано в другие разделы этого путеводителя. Если Вы развертываете или Менеджера по корпоративным мероприятиям Siebel для потребительских заявлений Финансов или Siebel, которые не являются определенными для финансовых услуг одновременно ни с какими другими потребительскими заявлениями Финансовых услуг Siebel, то Вы должны создать отдельного анонимного пользователя. Новый анонимный пользователь используется для менеджера по корпоративным мероприятиям Siebel для финансов и для потребительских заявлений Siebel, которые не являются определенными для финансовых услуг; то есть, заявления зарегистрированы в другие разделы этого путеводителя. Возложите обязанности на этого анонимного пользователя, поскольку они зарегистрированы для GUESTCST в Данных о Семени. Когда Вы добавляете TESTUSER к базе данных, определите ценности для Ответственности и Новых областей Ответственности, которые подходят для типичного зарегистрированного пользователя для применения, которое Вы настраиваете. Поскольку информация об обязанностях по семени предусмотрела определенные заявления, посмотрите Данные о Семени для Финансовых услуг Siebel и Данные о Семени. Параметры для приложений Финансовых услуг Siebel перечислены прежде всего в eapps_sia.cfg файле. eapps.cfg файл также включен, как зарегистрировано в другие разделы этого путеводителя. У eapps.cfg файла есть включать линия, которая указывает на eapps_sia.cfg файл. Ссылки всюду по этой теме к eapps.cfg файлу относятся к eapps.cfg файлу и eapps_sia.cfg файлу.

Чтобы с помощью Kerberos получить доступ к S (обычно это сервер), C (как правило – клиент) посылает Kerberos запрос, содержащий сведения о нем (клиенте) и о запрашиваемой услуге. В ответ Kerberos возвращает так называемый билет, зашифрованный секретным ключом сервера, и копию части информации из билета, зашифрованную секретным ключом клиента. Клиент должен расшифровать вторую порцию данных и переслать ее вместе с билетом серверу. Сервер, расшифровав билет, может сравнить его содержимое с дополнительной информацией, присланной клиентом. Совпадение свидетельствует о том, что клиент смог расшифровать предназначенные ему данные (ведь содержимое билета никому, кроме сервера и Kerberos, недоступно), В следствиипродемонстрировал знание секретного ключа. Значит, клиент – поэтому тот, за кого себя выдает. Подчеркнем, что секретные ключи в процессе проверки подлинности не передавались по сети (даже в зашифрованном виде) – они только использовались для шифрования. Как организован первоначальный обмен ключами между Kerberos и субъектами и как субъекты хранят свои секретные ключи – вопрос отдельный. Проиллюстрируем описанную процедуру на рисунке 2.1.

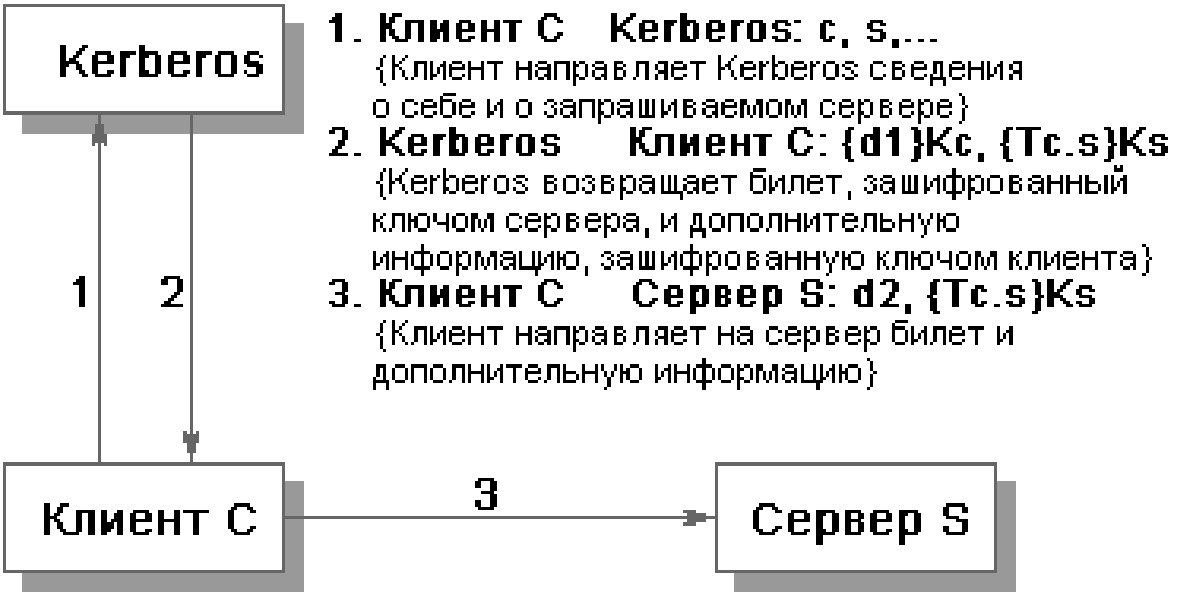


Рисунок 2.1 - Проверка сервером S подлинности клиента C.

Здесь c и s – сведения (например, имя), соответственно, о клиенте и сервере, d1 и d2 – дополнительная (по отношению к билету) информация, Tc.s – билет для клиента C на обслуживание у сервера S, Kc и Ks – секретные ключи клиента и сервера, {info}K – информация info, зашифрованная ключом K.

Приведенная схема – крайне упрощенная версия реальной процедуры проверки подлинности. Более подробное рассмотрение системы Kerberos можно найти, например, в статье В. Галатенко "Сервер аутентификации Kerberos (Jet Info, 1996, 12-13). Нам же важно отметить, что Kerberos не

только устойчив к сетевым угрозам, но и поддерживает концепцию единого входа в сеть [2].

2.5 Идентификация и аутентификация с помощью биометрических данных

Биометрия представляет собой совокупность автоматизированных методов идентификации и/или аутентификации людей на основе их физиологических и поведенческих характеристик. К числу физиологических характеристик принадлежат особенности отпечатков пальцев, сетчатки и роговицы глаз, геометрия руки и лица и т.п. К поведенческим характеристикам относятся динамика подписи (ручной), стиль работы с клавиатурой. На стыке физиологии и поведения находятся анализ особенностей голоса и распознавание речи.

Биометрией во всем мире занимаются очень давно, однако долгое время все, что было связано с ней, отличалось сложностью и дороговизной. В последнее время спрос на биометрические продукты, в первую очередь в связи с развитием электронной коммерции, постоянно и весьма интенсивно растет. Это понятно, поскольку с точки зрения пользователя гораздо удобнее предъявить себя самого, чем что-то запоминать. Спрос рождает предложение, и на рынке появились относительно недорогие аппаратно-программные продукты, ориентированные в основном на распознавание отпечатков пальцев.

В общем виде работа с биометрическими данными организована следующим образом. Сначала создается и поддерживается база данных характеристик потенциальных пользователей. Для этого биометрические характеристики пользователя снимаются, обрабатываются, и результат обработки (называемый биометрическим шаблоном) заносится в базу данных (исходные данные, такие как результат сканирования пальца или роговицы, обычно не хранятся).

В дальнейшем для идентификации (и одновременно аутентификации) пользователя процесс снятия и обработки повторяется, после чего производится поиск в базе данных шаблонов. В случае успешного поиска личность пользователя и ее подлинность считаются установленными. Для аутентификации достаточно произвести сравнение с одним биометрическим шаблоном, выбранным на основе предварительно введенных данных.

Обычно биометрию применяют вместе с другими аутентификаторами, такими, например, как интеллектуальные карты. Иногда биометрическая аутентификация является лишь первым рубежом защиты и служит для активизации интеллектуальных карт, хранящих криптографические секреты; в таком случае биометрический шаблон хранится на той же карте.

Активность в области биометрии очень велика. Организован соответствующий консорциум (см.http://www.biometrics.org/), активно ведутся работы по стандартизации разных аспектов технологии (формата обмена данными, прикладного программного интерфейса и т.п.), публикуется масса

рекламных статей, в которых биометрия преподносится как средство обеспечения сверхбезопасности, ставшее доступным широким массам.

На наш взгляд, к биометрии следует относиться весьма осторожно. Необходимо учитывать, что она подвержена тем же угрозам, что и другие методы аутентификации. Во-первых, биометрический шаблон сравнивается не с результатом первоначальной обработки характеристик пользователя, а с тем, что пришло к месту сравнения. А, как известно, за время пути... много чего может произойти. Во-вторых, биометрические методы не более надежны, чем база данных шаблонов. В-третьих, следует учитывать разницу между применением биометрии на контролируемой территории, под бдительным оком охраны, и в "полевых" условиях, когда, например к устройству сканирования роговицы могут поднести муляж и т.п. В-четвертых, биометрические данные человека меняются, так что база шаблонов нуждается в сопровождении, что создает определенные проблемы и для пользователей, и для администраторов.

Но главная опасность состоит в том, что любая "пробоина" для биометрии оказывается фатальной. Пароли, при всей их ненадежности, в крайнем случае, можно сменить. Утерянную аутентификационную карту можно аннулировать и завести новую. Палец же, глаз или голос сменить нельзя. Если биометрические данные окажутся скомпрометированы, придется, как минимум производить существенную модернизацию всей системы [2].

2.6 Аутентификация как базис безопасности

2.6.1 Этапы входа в систему

Современные веб-ресурсы содержат в себе множество функционала, который по разному доступен различным пользователям. Основной вопрос заключается в следующем: как обеспечить, чтобы система узнавала пользователей и в соответствии с результатом распознавания – предоставляла им тот или иной функционал? Безопасность процедуры входа для пользователей во многом определяет защищенность информационной системы в целом.

Процедуру входа зачастую именуют по-разному: войти, авторизоваться, аутентифицироваться, залогиниться. Обычно все понимают, о чем идет речь, но методологически термины используются не всегда корректно...

В теории информационной безопасности процесс входа в систему делится на 3 основные стадии:

1. Идентификация (Identification)
   Под идентификацией, применительно к обеспечению информационной безопасности компьютерной системы, понимают однозначное распознавание уникального имени субъекта (пользователя).
2. Аутентификация (Authentication)
   Аутентификация обеспечивает подтверждение подлинности субъекта, В

следствииподтверждение того, что предъявленное имя соответствует данному субъекту.

3. Авторизация (Authorization)
   На третьем шаге, система проводит процедуру авторизации, когда на основании результата процедуры аутентификации она дает пользователю тот или иной уровень доступа.

Этап аутентификации является самым важным в этой цепочке и основная проблема заключается в том, чтобы сделать эту процедуру максимально точной и безопасной.

Аутентификация бывает односторонней (обычно только клиент доказывает свою подлинность серверу) и двусторонней (взаимной). Пример односторонней аутентификации - процедура входа пользователя в систему, например в продукты «1С-Битрикс»

2.6.2 Способы аутентификации

Способы аутентификации пользователей в компьютерных системах делят на три группы:

- К первой группе относятся способы аутентификации, основанные на том, что пользователь знает некоторую подтверждающую его подлинность информацию (обычно это парольная аутентификация).
- Ко второй группе относятся способы аутентификации, основанные на том, что пользователь имеет некоторый материальный объект, который может подтвердить его подлинность (например, пластиковую карту с идентифицирующей пользователя информацией).
- К третьей группе относятся способы аутентификации, основанные на таких данных, которые позволяют однозначно считать, что пользователь и есть тот самый субъект, за которого себя выдает (биометрические данные, особенности клавиатурного почерка и т.п.).

2.7 Парольная аутентификация

Использование пароля в качестве аутентификационного фактора, наверное, еще очень долго будет являться наиболее распространенным способом решения задач определения подлинности. В первую очередь, в силу своей простоты и низких затрат на обеспечение всей инфраструктуры, особенно если речь идет о веб-приложениях.

Парольная защита заслуженно считается не очень надежной. В программных продуктах мы постарались максимально возможно усилить аутентификацию пользователей с использованием паролей. В следующих пунктах показано за счет чего это достигается.

2.7.1 Задание стойкости паролей

При выборе паролей пользователи системы (например, веб-сайта) должны руководствоваться двумя, по сути взаимоисключающими, правилами - пароли должны трудно подбираться и легко запоминаться (поскольку пароль

ни при каких условиях не должен нигде записываться, так как в этом случае необходимо будет дополнительно решать задачу защиты носителя пароля).

Но поскольку правила взаимоисключающие, пароли обычно простые и очень хорошо подбираются, а также записываются повсеместно без надлежащей защиты этих самых записей. Если вторую неприятность решить технически невозможно, да и организационные меры особо не помогают, то вот с обеспечением сложности пароля можно поработать.

Сложность подбора пароля определяется, в первую очередь, мощностью множества символов, используемого при выборе пароля (N), и минимально возможной длиной пароля (к). В этом случае число различных паролей может быть оценено снизу как, $C_p = N^k$. Например, если множество символов пароля образуют строчные латинские буквы, а минимальная длина пароля равна 3, то $C_p = 26^3 = 17576$ (что совсем немного для программного подбора). Если же множество символов пароля состоит из строчных и прописных латинских букв, а также из цифр и минимальная длина пароля равна 6, то $C_p = 62^6 = 56800235584$.

Сложность выбираемых пользователями КС паролей должна устанавливаться администратором при реализации установленной для данной системы политики безопасности.

Другими параметрами политики учетных записей при использовании парольной аутентификации являются:

- максимальный срок действия пароля (любой секрет не может сохраняться в тайне вечно);
- несовпадение пароля с логическим именем пользователя, под которым он зарегистрирован в системе;
- неповторяемость паролей одного пользователя.

С точки зрения теории, эти рекомендации очень полезны и обычно применяются в совокупности, но как правило на практике они не дают сколько-либо весомого усиления защиты. Так, например, требование неповторяемости паролей может быть реализовано двумя способами. Во-первых, можно установить минимальный срок действия пароля (в противном случае пользователь, вынужденный после истечения срока действия своего пароля поменять его, сможет тут же сменить пароль на старый). Во-вторых, можно вести список уже использовавшихся данным пользователем паролей (максимальная длина списка при этом может устанавливаться администратором).

К сожалению, обеспечить реальную уникальность каждого вновь выбираемого пользователем пароля с помощью приведенных выше мер практически невозможно. Пользователь может, не нарушая установленных ограничений, выбирать пароли «A1», «A2», ... где A — первый пароль пользователя, удовлетворяющий требованиям сложности. Тоже самое касается несовпадения логина с паролем. Ничто не помешает сделать логин 'dima', а пароль 'dima_1'.

2.7.2 Двухфакторная аутентификация

Двухфакторная аутентификация обеспечивает дополнительную защиту, при входе в систему запрашиваются не только имя пользователя и пароль, но и уникальные "коды подтверждения". Даже если злоумышленник узнает ваш пароль, получить доступ к аккаунту ему не удастся.

С технической точки зрения применение двухфакторной аутентификации позволит минимизировать риски, связанные с использованием долговременных паролей. Отдельным пользователям в группе доступа предоставляют доступу к данным, связывая группу доступа с каталогами или категориями данных.

Знайте о следующих поведениях пользовательского интерфейса, связанных с соединением группы доступа с каталогом или категорией:

Наследование доступа. Когда Вы связываете группу доступа с категорией, ее группы потомка также связаны с категорией. Однако это наследование осуществлено во время, которым управляют и не представлено в базе данных. Также, группы доступа потомка, связанные с категорией, не показаны в списке групп, связанных с категорией.

Каскадная кнопка. Нажатие на кнопку Cascade предоставляет данной группе доступа видимость ко всем детским категориям текущего каталога или категории. Нажатие на эту кнопку неоднократно не имеет никакого эффекта. Вы должны вручную разъединить группу с детскими категориями, чтобы отменить каскад доступа.

Частный каталог. Если Вы определяете каталог, чтобы быть частными, его категории все установлены как частные. Если Вы удаляете частную жизнь на уровне каталога, категории сохраняют частную жизнь. Вы должны тогда установить или удалить частную жизнь категории индивидуально.

2.7.3 Аутентификация на основе одноразовых паролей

Безопасность сети – важнейшая задача, стоящая перед ИТ. Решение формируется из многих составляющих, одна из них – безопасная аутентификация. OTP-технологии позволят уменьшить риски, с которыми сталкиваются компании при использовании долговременных запоминаемых паролей

Как уже неоднократно упоминалось в статьях наших авторов, вопросы безопасной аутентификации по-прежнему остаются актуальными. Очевидно, что любые попытки ограничения полномочий и предоставления доступа к ресурсам и функциям системы только тогда имеют смысл, когда мы можем быть уверены, что имеем дело с легальным пользователем. Следовательно, защита информационных систем начинается с проверки легитимности пользователя, который пытается получить доступ.

Привычный, ставший стандартным метод – использование долговременных, запоминаемых паролей. Этот способ проверки «правильности» пользователя в рамках современных ИТ-решений должен уйти в прошлое, хотя и имеет место в огромном числе компаний, в силу простоты использования и реализации.

Тем не менее использование такой аутентификации скорее должно свидетельствовать о незрелости ИТ-инфраструктуры с точки зрения безопасности или об отсутствии потребности бизнеса в построении защищенной информационной системы. Парольная аутентификация потенциально уязвима ввиду использования социотехники, внедрения клавиатурных шпионов, возможности перехвата и т.п. Я специально не касаюсь методов защиты, предлагаемых в рамках операционных систем, поскольку, во-первых, они не обеспечат решение вышеупомянутых проблем, во-вторых, эта тема подробно рассматривалась в статье «Active Directory Domain Services. Двухфакторная аутентификация. Теоретические основы»

Итак, эффективным методом, с технической точки зрения, будет применение двухфакторной аутентификации, которая в свою очередь позволит минимизировать риски, связанные с использованием долговременных паролей.

Аутентификация – процедура проверки подлинности, позволяющая достоверно убедиться в том, что, предъявивший свой идентификатор на самом деле является поэтому тем, за кого он себя выдает. Для этого он должен подтвердить факт обладания информацией, которая может быть доступна только ему одному (пароль, ключ и т.п.).

OTP – One Time Password, одноразовый пароль.

Фактор аутентификации — определенный вид информации, предоставляемый субъектом системе, например, пароль или отпечаток пальца.

Тенденции консьюмеризации, по сути своей уже ставшие в последнее время реальным положением вещей, приводят к тому, что сотрудникам требуется использовать разные типы устройств для доступа к ресурсам корпоративной сети. При работе в офисе используется стационарный или мобильный компьютер, в поездках проще обойтись планшетом или смартфоном. Разумеется, не всегда использование смарт-карт или USB-ключей возможно, у нас просто может не быть USB-разъема для подключения, тем не менее потребность в строгой аутентификации только возрастает с увеличением разновидностей применяемых устройств, которые далеко не всегда управляются и контролируются ИТ-службами компании.

Консьюмеризация – тенденция широкого внедрения пользовательских устройств (смартфонов, планшетов) в корпоративную IT-систему.

Здесь существенную помощь окажет технология одноразовых паролей (OTP), которая, с одной стороны, может помочь реализовать строгую двухфакторную аутентификацию, а с другой, не потребует существенных затрат на внедрение и поддержку. Соединение Access Group с категорией.

Связывая группу доступа с категорией основных данных, Вы предоставляете доступ к данным в категории отдельным пользователям в группе доступа.

Для категории и всех ее подкатегорий, чтобы быть видимым только группам доступа, связанным с ним, Частный флаг категории должен быть

установлен или Частный флаг каталога или категории, с которой спускается категория, должен быть установлен.

2.7.4 Одноразовые пароли. Терминология. Форм-факторы

Одноразовые пароли (OTP, One-Time Passwords) – динамическая аутентификационная информация, генерируемая различными способами для однократного использования. Применение одноразового пароля возможно лишь один раз либо в некоторых реализациях в течение незначительного промежутка времени.

OTP-токен – мобильное персональное устройство, принадлежащее определенному пользователю, генерирующее одноразовые пароли, используемые для аутентификации данного пользователя.

Одноразовый пароль (OTP) практически неуязвим для атаки сетевого анализа пакетов, что является существенным преимуществом перед обычными долговременными паролями. Даже если одноразовый пароль будет перехвачен, вероятность того, что им смогут воспользоваться, весьма сомнительна, чтобы ее рассматривать всерьез.

Пользователь с логином администратора может управлять доступом к задачам, связывая задачи с обязанностями по пользователю. Чтобы получить доступ к задаче, на пользователя должна быть возложена ответственность, которая позволяет доступ к задаче. Пользователь, на которого возложена больше чем одна ответственность, может получить доступ к любой задаче, которая связана с одной из его или ее обязанностей.

Администратор может также определить гиперссылки к задачам, связанным с ответственностью; эти связи задачи тогда появляются на домашней странице пользователей, на которых возложена ответственность.

Для пользователя, чтобы получить доступ к задаче, по крайней мере одни из обязанностей пользователя должны быть явно возложены на задачу.

Еще одним важным преимуществом использования OTP-токенов является то, что они дополнительно требуют от пользователя ввода PIN-кода, например, при активации, генерации OTP, для предъявления аутентифицирующему серверу совместно с OTP. Следовательно, у нас возникает еще один фактор аутентификации, и мы говорим о двухфакторной аутентификации пользователя в системе на основе обладания чем-либо (Authentication by Ownership) и на основе знания чего-либо (Authentication by Knowledge).

Существуют различные варианты подобных устройств. Это может быть карманный OTP-калькулятор, брелок, смарт-карты, устройство, обеспечивающее двойную функциональность (USB-ключ + OTP), например, SafeNet eToken NG-OTP. Кроме того, может использоваться программное обеспечение, скажем, выполняемое на смартфоне пользователя. Существует целый ряд компаний, выпускающих подобные средства, как программные, так и аппаратные, например, SafeNet, RSA, Gemalto.

Принципы работы

Для генерации одноразовых паролей OTP-токены используют криптографические алгоритмы:

- симметричная криптография – в этом случае пользователь и сервер аутентификации используют один и тот же секретный ключ;
- асимметричная криптография – в этом случае в устройстве хранится закрытый ключ, а сервер аутентификации использует соответствующий открытый ключ.

Как правило, в таких устройствах применяется поэтому симметричная криптография. Устройство содержит уникальный секретный ключ, который будет использован для шифрования некоторых данных, используемых для генерации пароля. Тот же самый ключ содержится и на сервере аутентификации. Сервер шифрует их же и сравнивает результат с тем, что получает от клиента. При совпадении значений считается, что процесс аутентификации прошел успешно. Поэтому этот вариант мы и рассмотрим в статье.

После создания ответственности Вы можете создать связи с задачами, обычно выполняемыми сотрудниками, которые несут ту ответственность. Эти связи тогда показаны в списке задачи на домашней странице для этих сотрудников.

Для каждой связи задачи Вы входите в заголовок, файл изображения и описание. Кроме того, определите представление, где задача выполнена. Когда пользователь нажимает на гиперссылку для этой задачи на домашней странице, это представление появляется. Персонализация этого типа уже определена для различных обязанностей по семени.

Следующая процедура описывает, как создать связи задачи для ответственности.

Токены, использующие симметричную криптографию, могут работать в асинхронном и синхронном режимах. Асинхронный режим – технология «запрос-ответ», синхронный режим – «только ответ» (в этом варианте синхронизация может быть по времени или по событию).

«Запрос-Ответ» (Challenge-Response)

Рассмотрим принципы работы метода «запрос-ответ»

1. Пользователь вводит свое имя на рабочей станции. Имя пользователя передается серверу по сети в открытом виде.
2. Сервер генерирует случайный запрос.
3. Этот запрос передается по сети в открытом виде.
4. Пользователь вводит полученные данные в свой OTP-токен.
5. OTP-токен шифрует запрос с помощью секретного ключа пользователя, и формируется ответное значение, которое отображается на экране токена.
6. Пользователь вводит это значение на своей рабочей станции.
7. Затем ответ передается по сети в открытом виде.

8. Сервер осуществляет поиск записи пользователя в аутентификационной БД, используя секретный ключ пользователя, шифрует тот же запрос.

9. Далее сравниваются два значения: полученное от пользователя и вычисленное сервером. При их совпадении аутентификация считается успешной.

Нетрудно заметить, что такой метод предполагает большое количество шагов и несколько большую вовлеченность пользователя в процесс аутентификации, добавляется необходимость еще одного дополнительного ввода данных, что создает неудобства.

Каждому пользователю можно было назначить многократные обязанности, чтобы обеспечить доступ ко всем необходимым взглядам. Одна ответственность определена как основная ответственность. Пользователь видит расположение счета, связанное с его или ее основной ответственностью. Карта сайта предоставляет этому пользователю доступ к супернабору экранов и рассматривает определенный в обязанностях, с которыми связан пользователь.

Чтобы возложить основную ответственность на пользователя, выполните следующую процедуру.

Альтернативным вариантов является метод «только ответ».

«Только ответ» (Response only)

В методе «только ответ» OTP-устройство и сервер аутентификации генерируют скрытый запрос, используя значение предыдущего запроса. На первоначальной стадии используется уникальное случайное начальное значение, формируемое при инициализации токена.

Рассмотрим принципы работы.

1. Пользователь активирует токен, который отображает ответ на скрытый запрос.
2. Пользователь вводит свое имя и этот ответ на рабочей станции.
3. Имя пользователя и ответ передаются по сети в открытом виде.
4. Сервер находит запись пользователя и генерирует такой же скрытый запрос, шифруя его с помощью секретного ключа пользователя.
5. Сервер сравнивает ответ, полученный от пользователя, с вычисленным.
6. При совпадении результатов аутентификация считается успешной.

Определяет имя пользователя отчета в справочнике с достаточных разрешений прочитать информацию любого пользователя и сделать любую необходимую администрацию.

Этот пользователь обеспечивает начальное закрепление каталога LDAP или Активного Справочника с Прикладным диспетчером объектов, когда пользователь просит страницу логина, или иначе анонимный просмотр справочника требуется.

Вы входите в этот параметр как в полное выдающееся имя (DN), например "uid=APPUSER, ou=people, o=example.com" (включая кавычки) для LDAP. Адаптер безопасности использует это имя, чтобы связать.

При таком способе сервер аутентификации и OTP-устройство генерируют пароль, базируясь на показании внутренних часов. При этом возможность применения данного пароля ограничена определенным интервалом времени. Одним из примеров такого аутентификационного устройства является RSA SecurID. Посмотрим, как работает этот механизм .

1. Пользователь активизирует свой токен, который генерирует OTP, зашифровывая показания часов с помощью своего секретного ключа.
2. Пользователь вводит свое имя и этот OTP на своей рабочей станции. Имя пользователя и OTP передаются по сети в открытом виде.
3. Сервер находит запись пользователя и шифрует показания своих часов с помощью хранимого им секретного ключа, получая в результате OTP.
4. Далее сравниваются вычисленный и полученные OTP. При совпадении значений аутентификация считается успешной.

Необходимо учитывать, что такие системы чувствительны к синхронизации времени.

Синхронизация по событию

В режиме «синхронизация по событию» (см. рис. 4) токен и сервер ведут количественный отсчет прохождения аутентификаций пользователем и на основе этого числа генерируют OTP.

1. Пользователь активирует токен, который генерирует OTP, зашифровывая количество прохождений аутентификаций данного пользователя с помощью своего секретного ключа.
2. Пользователь вводит свое имя и OTP на рабочей станции.
3. Эти данные передаются по сети в открытом виде.
4. Аутентификационный сервер находит запись пользователя и шифрует значения количества прохождений аутентификаций данного пользователя с помощью хранимого секретного ключа, получая в результате OTP.
5. Далее сравниваются полученные значения, и при совпадении аутентификация считается успешной.

В этом варианте проблему представляет ситуация, когда устройство аутентификации «обгонит» сервер, например, при многократном повторном нажатии кнопки генерации OTP на устройстве.

Характерными представителями таких устройств являются eToken PASS и eToken NG-OTP компании SafeNet.

Типичные атаки и методы борьбы с ними

Для того чтобы понимать, как противостоять типовым уязвимостям системы при использовании одноразовых паролей, посмотрим существующие варианты атак, которые может осуществить «взломщик» системы.

Определяет тип признака, под которым пароль логина пользователя сохранен в справочнике.

Вход LDAP должен быть userPassword. Однако, если Вы используете адаптер безопасности LDAP, чтобы подтвердить подлинность против Microsoft Active Directory, затем установите ценность этого параметра к unicodePWD.

Активный Справочник не хранит пароль в признаке, таким образом, этот параметр не используется адаптером безопасности ADSI. Вы должны, однако, определить стоимость для параметра Типа Признака Пароля, даже если Вы используете адаптер ADSI. Определите ценность unicodePWD.

Этот вариант атаки подразумевает, что злоумышленник перехватывает одноразовый пароль, посланный законным пользователем, блокирует пользователя, использует перехваченный пароль для собственной аутентификации в системе. Решением данной проблемы будет использование технологии «запрос-ответ» и отказ от использования одноразовых паролей, имеющих легитимность в течение периода времени, либо использование синхронизации по событию. В следствиипроводить аутентификацию потребуется каждый раз при новом соединении.

Определяет, используется ли Secure Sockets Layer (SSL) для связи между адаптером безопасности LDAP и справочником.

Если этот параметр пуст, то SSL не используется. Чтобы использовать SSL, ценность этого параметра должна быть абсолютным путем бумажника, произведенного менеджером Oracle Wallet, который содержит свидетельство для центра сертификации, который используется сервером LDAP. Это типичная ситуация, и решением в этом случае будет использование не только одноразовых паролей для аутентификации в системе, но и обязательное применение PIN-кода, В следствиипереход к двухфакторной аутентификации.

Определяет абсолютный путь (не относительно BaseDN) объекта в справочнике, у которого есть общая база данных, составляют применение. Если это пусто, то счет базы данных ищется в DN пользователя, как обычно. Если это не пусто, то счет базы данных на всех пользователей ищется в общих верительных грамотах DN вместо этого. Тип признака все еще определен ценностью CredentialsAttributeType.

Например, если SharedCredentialsDN установлен в:

"uid=HKIM, ou=people, o=example.com"

когда пользователь заверен, адаптер безопасности восстанавливает счет базы данных от соответствующего признака в отчете HKIM. Значение по умолчанию этого параметра - пустая последовательность. Такой вид атаки трудно воспринимать всерьез, поскольку решение проблемы лежит на поверхности – блокировка токена при многократном неправильном вводе PIN-кода.

Извлечение значения секретного ключа из программного аутентификационного токена

Наряду с аппаратными реализациями OTP-токенов, существуют также и программные варианты. В этом случае возможен вариант атаки, когда злоумышленник копирует программное обеспечение и пытается отыскать хранящийся в нем секретный ключ, чтобы в дальнейшем использовать для аутентификации под видом легального пользователя. Решение указанной проблемы достигается тем, что PIN-код является частью ключа, без знания которого не удастся генерировать OTP, даже зная часть секретного ключа, хранимого в самом программном токене.

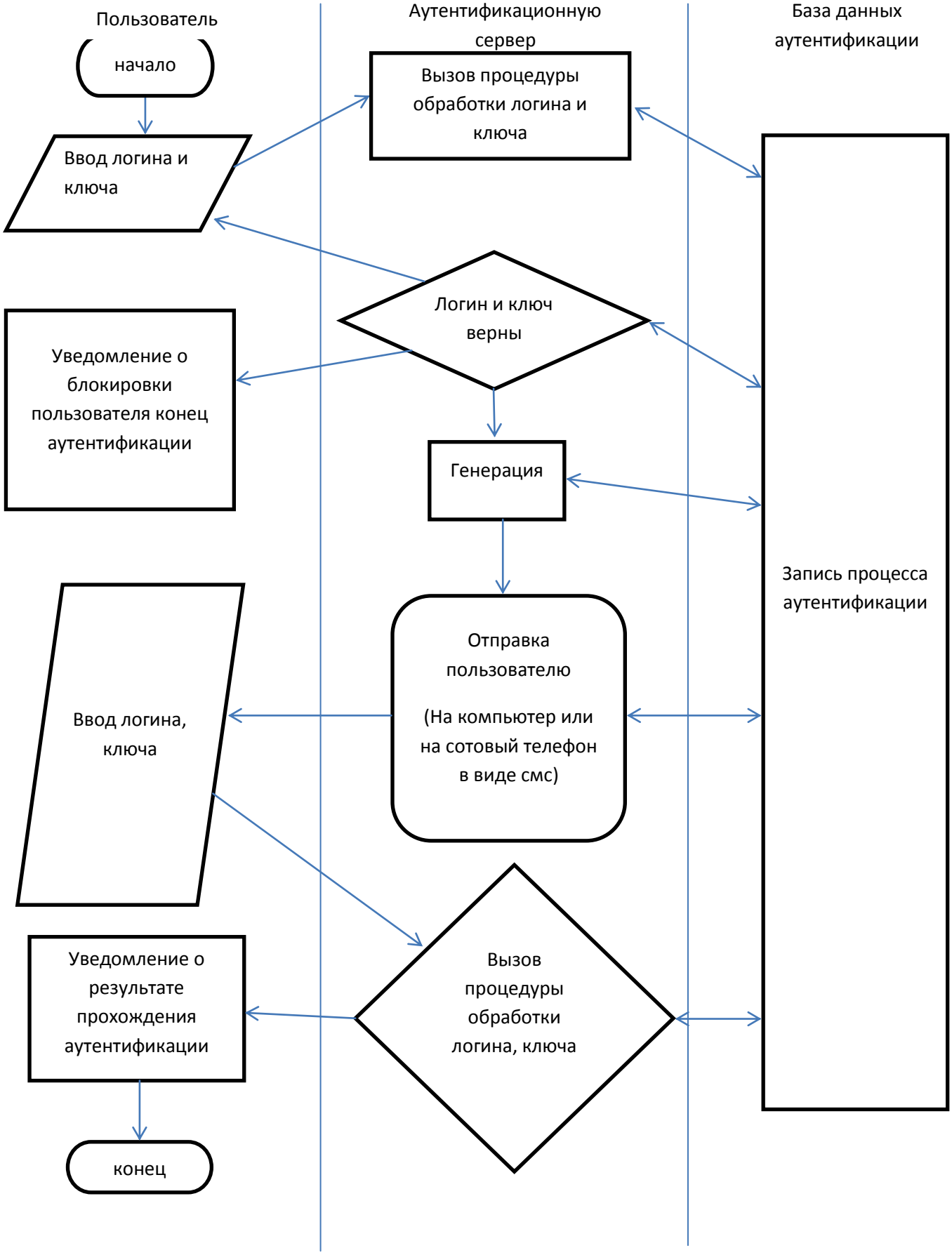
Нечестный администратор безопасности

Встречается ситуация, когда злоумышленником является сам администратор, отвечающий за выдачу средств аутентификации. Следовательно, на первый взгляд он может обладать возможностью создания дубликатов аутентификационных средств, посредством которых легко получить доступ к системе от имени другого пользователя. Для решения вышеназванной проблемы следует обеспечить разделение функций, В следствиив процессе активирования токена должно участвовать не менее двух человек, каждый из которых обязан выполнять строго определенный набор задач.

Итак, мы видим, что внедрение одноразовых паролей позволит снизить риски, возникающие при обычной парольной аутентификации. Появляется возможность решения проблемы мобильных пользователей, для которых применение внешних подключаемых устройств (смарт-карт, USB-ключей) нередко представляет сложности, при этом мы можем говорить о двух факторах, поскольку OTP и PIN используются совместно в процессе аутентификации. При внедрении решений в корпоративной среде имеет смысл рассмотреть комбинированные устройства.

Определяет DLL, который осуществляет API адаптера безопасности, требуемый для интеграции с Бизнес-приложениями Siebel. Расширение файла не должно быть явно определено. Например, sscfsadb.dll осуществляет адаптер безопасности базы данных Siebel во внедрении Windows, и libsscfsadb.so делает так во внедрении UNIX. Если название DLL адаптера используется во внедрении UNIX, то это преобразовано внутренне в фактическое имя файла DLL.

Блок схема двухфакторной аутентификации



2.8 Oracle Siebel CRM

Oracle Siebel CRM — это система по управлению взаимоотношениями с клиентами, разработанная корпорацией Siebel Systems, которую приобрела корпорация Oracle в 2006 году.

В состав Oracle Siebel CRM входят следующие решения:

- Бизнес-аналитика;
- Управление продажами;
- Управление маркетингом;
- Контакт-центры и центры телефонного обслуживания;
- Управление обработкой заказов;
- Управление отношениями с партнерами;
- Управление отношениями с сотрудниками;

Преимущества

- Широкие функциональные возможности;
- Гибкость и расширяемость - архитектура и средства настройки Siebel позволяют конфигурировать продукт в соответствии с требованиями бизнеса;
- Модульная структура - позволяет компаниям выбирать и использовать только необходимые модули. Это дает возможность внедрять систему поэтапно, начиная с базовых модулей, постепенно наращивая возможности;
- Быстрое внедрение - достигается за счет готовой конфигурации и большого количества типовых объектов;
- Наличие более 20-ти полнофункциональных отраслевых решений - отраслевые CRM решения, адаптированные под особенности конкретных отраслей, снижают стоимость доли услуг в CRM-проекте (а также время на внедрение системы).

Идентификация - процесс подтверждения личности пользователя. Бизнес-приложения Siebel поддерживают многократные подходы для подтверждения пользователей. Вы выбираете или идентификацию адаптера безопасности или Сеть идентификация SSO для Ваших пользователей заявления Siebel:

- Идентификация адаптера безопасности. Бизнес-приложения Siebel служат основой адаптера безопасности, чтобы поддержать несколько различных пользовательских сценариев идентификации:
- Идентификация базы данных. Бизнес-приложения Siebel поддерживают идентификацию против основной базы данных. В этой архитектуре адаптер безопасности подтверждает подлинность пользователей против базы данных Siebel. Бизнес-приложения Siebel обеспечивают адаптер безопасности базы данных (он формируется как адаптер безопасности по умолчанию).
- Lightweight Directory Access Protocol (LDAP) или идентификация Active Directory Service Interfaces (ADSI). Бизнес-приложения Siebel поддерживают идентификацию против LDAP-послушных

справочников или Microsoft Active Directories. В этой архитектуре адаптер безопасности подтверждает подлинность пользователей против справочника. Бизнес-приложения Siebel обеспечивают следующие два адаптера безопасности, чтобы подтвердить подлинность против директивных серверов:
  - Адаптер безопасности ADSI
  - Адаптер безопасности LDAP
- Для получения дополнительной информации займитесь LDAP или Идентификацией Адаптера безопасности ADSI.
- Обычай. Вы можете использовать таможенный адаптер, который Вы обеспечиваете и формируете Бизнес-приложения Siebel, чтобы использовать этот адаптер. Для получения дополнительной информации посмотрите Адаптер безопасности SDK.
- Сеть Единственный Знак - На (Сеть SSO). Этот подход использует внешнее обслуживание идентификации подтвердить подлинность пользователей, прежде чем они получат доступ к заявлению Siebel. В этой архитектуре адаптер безопасности не подтверждает подлинность пользователя. Адаптер безопасности просто ищет и восстанавливает идентификатор пользователя Siebel пользователя и счет базы данных от справочника, основанного на ключе идентичности, который принят от внешнего обслуживания идентификации.

Вы можете выбрать подход для пользовательской идентификации индивидуально для каждого применения в Вашей среде, основанной на определенных основных эксплуатационных характеристиках. Однако есть административные преимущества для использования последовательного подхода через все Ваши Бизнес-приложения Siebel, потому что последовательный подход понижает полную сложность развертывания.

Ценности параметра конфигурации определяют, как взаимодействуют Ваши компоненты архитектуры идентификации. Для получения информации о цели параметров конфигурации посмотрите Параметры Конфигурации, Связанные с Идентификацией..

Аутентификация это процесс проверки личности пользователя. Siebel Business Applications поддерживать несколько подходов для аутентификации пользователей . Вы выбираете либо безопасности аутентификации адаптера или аутентификации Web SSO для ваших бизнес-приложений Siebel пользователей. Аутентификации адаптер безопасности . Siebel Business Applications обеспечивают основу адаптера безопасности для поддержки несколько различных сценариев аутентификации пользователей. Проверка подлинности базы данных . Siebel Business Applications поддерживает аутентификацию при основной базе данных. В этой архитектуре , адаптер безопасности проверку подлинности пользователей к базе данных Siebel . Siebel Business Applications предоставления адаптера безопасности базы данных.

Облегченный протокол доступа к каталогам (LDAP) или услуги Active Directory Интерфейсы (ADSI) аутентификации . Аутентификации поддержка Siebel Business Applications против LDAP -совместимых или Microsoft Active Directory (AD) каталоги . В этой архитектуре , адаптер безопасности проверяет подлинность пользователей от каталога . Siebel Business Applications предоставить следующие два адаптера безопасности для проверки подлинности серверов каталогов. Для получения дополнительной информации см. в разделе О LDAP или ADSI аутентификации адаптер Безопасности. Вы можете использовать собственный адаптер , которую Вы предоставили , и настроить Siebel бизнес-приложений , чтобы использовать этот адаптер . Для получения дополнительной информации см. в разделе Безопасность адаптер SDK. Веб Single Sign-On (Web SSO) . Этот подход использует внешний сервис аутентификации для аутентификации пользователей , прежде чем они доступ Siebel бизнес-приложений . В этой архитектуре , адаптер безопасности не аутентификации пользователя. Адаптер безопасности просто смотрит и получает Siebel идентификатор пользователя и базы данных учетной записи пользователя из каталога на основе ключа , удостоверяющего личность , которая принимается от внешней службы аутентификации. Для получения дополнительной информации см. один веб- регистрации аутентификации . Вы можете выбрать подход к аутентификации пользователя индивидуально для каждого приложения в вашей среде , на основе конкретных требований применения . Однако, есть административные преимущества использования единого подхода во всех ваших Siebel бизнес-приложений , потому что последовательный подход снижает общую сложность развертывания . Обратите внимание, чтовеб-клиент Siebel Mobile может использовать только проверку подлинности базы данных против локальной базе данных на мобильном клиенте. Для получения дополнительной информации о проверке подлинности для веб-клиента Siebel Mobile см. Siebel Remote и Replication Manager Администрация Guide.Referential и процедурная информация в следующих разделах относится ко всем основным стратегий аутентификации. Большая часть определенной информации в этих тем относится к более чем одной стратегии аутентификации. Часть информации относится как к аутентификации и управления пользователями. Параметры конфигурации, связанные с аутентификацией. Значения параметров конфигурации определить, как взаимодействуют ваши аутентификации компонентов архитектуры. Для получения информации о цели конфигурационных параметров, см. описание параметров, связанных с аутентификацией. Данные семена. При установке Siebel бизнес-приложений, вы получаете данные семя, какая есть, связанные с аутентификацией, регистрации пользователей и доступа пользователей к Siebel бизнес-приложений. Для получения более подробной информации о данных семян, которая предоставляется и процедуры для просмотра и редактирования семян данных. Предоставление возможности элементов управления ActiveX

Бизнес-приложения Siebel в способе высокой интерактивности используют технологию ActiveX, чтобы поставить несколько особенностей, например, почтовой интеграции клиента. Браузеру, запускающему приложение высокой интерактивности, нужно позволить получить доступ и использовать Элементы управления ActiveX. Вы можете сделать одно из следующего:

- Позвольте пользователям загружать Элементы управления ActiveX по требованию с Web-сервера.
- Этот выбор не предпочтен, потому что он требует, чтобы пользователи были назначенными разрешениями, связанными с продвинутыми пользователями.
- Разверните необходимые Элементы управления ActiveX на компьютерах пользователей (рекомендуемый выбор).
- Если Вы развертываете Элементы управления ActiveX на компьютерах пользователей, то Вы можете формировать параметры настройки браузера клиента, чтобы препятствовать тому, чтобы были загружены дополнительные Элементы управления ActiveX.

Личная безопасность устройства клиента обработана за пределами Бизнес-приложений Siebel. Вы можете использовать утилиты, которые обеспечивают безопасность компьютерного уровня, проводя в жизнь компьютерные пароли или шифруя компьютерный жесткий диск. Самые ведущие переносные устройства позволили пользователями пароли.

Рекомендуется, чтобы Вы использовали подход идентификации с двумя факторами (например, RSA Безопасный ID) для сетевых компонентов; это - процесс безопасности, который подтверждает пользовательские личности, используя что-то, что пользователи имеют и что-то, что они знают. Требование двух различных форм электронной идентификации снижает риск мошенничества и защищает от нападений пароля. Пользователи не должны оставлять автоматизированные рабочие места без присмотра, в то время как они загружены к Бизнес-приложениям Siebel; выполнение так делает их компьютер потенциально доступным для неавторизованных пользователей. Определите корпоративную политику для обработки оставленных без присмотра сессий PC. Oracle рекомендует использовать запертые паролем функции скринсейвера на всех PC. Программное обеспечение браузера обновления, когда новые версии выпущены; новые выпуски часто включают дополнительные механизмы безопасности. Если Вы используете Internet Explorer, то проверьте веб-сайт Microsoft на последние участки безопасности браузера.

Определенные особенности и функции в Бизнес-приложениях Siebel работают вместе с безопасностью или другими параметрами настройки на Web-браузере. Некоторые механизмы безопасности, обеспеченные поддержанными браузерами и операционными системами, не поддержаны, когда используется с Бизнес-приложениями Siebel.

Подробная информация о параметрах настройки браузера, используемых в развертывании клиентов Siebel, предоставлена в Гиде Системного

администрирования Siebel. Для получения дополнительной информации о параметрах настройки в Вашем Web-браузере, см. документацию, которая шла с Вашим браузером, и Системными требованиями Siebel и Поддержанными Платформами на Oracle Technology Network. Чтобы защитить от злонамеренного программного обеспечения (вредоносное программное обеспечение), примените участки безопасности, обеспеченные настольным поставщиком операционной системы на регулярной основе. То же самое верно для участков, выпущенных поставщиками антивирусного программного обеспечения, и компаниями, которые обеспечивают другие сторонние программные продукты, поддержанные Бизнес-приложениями Siebel. Радио Siebel обеспечивает беспроводной доступ в реальном времени к Бизнес-приложениям Siebel через позволенные браузером мобильные устройства. Взгляды Радио Siebel, предоставленные в XML или HTML, посылают через Web-сервер в беспроводную сеть и в конечном счете в позволенное браузером беспроводное устройство просителя. Бизнес-приложения Siebel поддерживают Web-сервер.

В окружающей среде, используя Радио Siebel, Web-сервер и Сервер Siebel проживают в пределах брандмауэра, таким образом защищая защиту информации. Стандартные протоколы используются, чтобы обеспечить основанные на браузере передачи данных через беспроводную сеть.

Многократные методы обеспечения данных доступны, включая Беспроводной транспортный Слой безопасности, эквивалент Secure Sockets Layer (SSL) для беспроводных устройств и сторонних продуктов. Для получения дополнительной информации о Радио Siebel посмотрите Гида администрации Радио Siebel. Корреспонденция Siebel, Представления Siebel и Предложения Siebel все использование Сервер Документа Siebel, чтобы произвести документы Microsoft Word и Microsoft PowerPoint через Сеть. Все шаблоны документа входят через Сервер Siebel. Также, Сервер Siebel управляет безопасностью и представляет единственного клиента, который взаимодействует непосредственно с Сервером Документа Siebel. Для получения дополнительной информации о Сервере Документа Siebel, см. Корреспонденцию Siebel, Предложения и Путеводитель Представлений.

Выполните шаги в следующей процедуре, чтобы обеспечить Сервер Документа Siebel. Эта тема предоставляет информацию об обеспечении почтового сервера и почтовых коммуникаций в окружающей среде Siebel.

Ответ на электронную почту Siebel позволяет организациям управлять и отвечать на большой объем входящей электронной почты. Ответ на электронную почту Siebel работает вместе с Коммуникационным Сервером Siebel и Вашим сторонним почтовым сервером, чтобы обработать электронную почту. И Коммуникационный Сервер Ответа и Siebel электронной почты Siebel установлен с Сервером Siebel.

Коммуникационный Сервер Siebel использует коммуникационные файлы водителя, чтобы общаться с почтовой системой и поддержать прибывающую и почтовую обработку за границу. Oracle поддерживает Интернет Сервер

SMTP/POP3 и Интернет Сервер SMTP/IMAP для использования с почтовыми серверами, которые поддерживают протокол SMTP для электронных писем за границу, или POP3 или протокол IMAP для прибывающей электронной почты.
Бизнес-приложения Siebel используют Oracle Business Intelligence Publisher (Oracle BI Publisher), чтобы произвести отчеты Siebel. В разъединенной окружающей среде Отчетов Siebel не требуются пользовательские механизмы идентификации.
В Сибеле Отчеты соединили окружающую среду, Oracle BI Publisher установлена отдельно из Бизнес-приложений Siebel, и доступ к Серверу Издателя ВИСМУТА заверен. Чтобы подтвердить подлинность пользовательского доступа к Серверу Издателя ВИСМУТА в Сибеле, Отчеты соединили окружающую среду, Вы можете осуществить одно из следующего:

- Модель безопасности Siebel. Эта модель обеспечивает идентификацию, используя Прикладной диспетчер объектов EAI.
- Модель безопасности LDAP. Эта модель обеспечивает идентификацию против справочника.
- Для получения информации об этих вариантах идентификации посмотрите Гида Отчетов Siebel.
- Окружающая среда Отчетов Siebel включает связи между двумя отдельными системами: Бизнес-приложения Siebel и Oracle BI Publisher. Чтобы предотвратить перехват данных, обеспечьте связи между Бизнес-приложениями Siebel и Oracle BI Publisher, используя SSL. Для получения информации о формировании SSL см. Путеводитель безопасности Siebel и документацию Oracle BI Publisher относительно Oracle Technology Network.

2.8.1. Аутентификация для Siebel Видеорешения

Руководство пользователя и аутентификации обрабатывает контроля доступа обеими анонимными пользователями и зарегистрированных пользователей. Скрытых пользователей . Незарегистрированные посетители вашего сайта Siebel Видеорешения может просмотреть только определенные его части , например, базы знаний общественности или самодиагностики инструментов. Незарегистрированные пользователи могут зарегистрироваться , создать свои персональные профили , и получить доступ к дополнительным возможностям Siebel Видеорешения . Незарегистрированные пользователи также могут получить логины и пароли от представителей обслуживания клиентов . Зарегистрированные пользователи . Польователи , которые зарегистрированы с Вашего веб-сайта Siebel Видеорешения будут иметь свои сохраненные логины и пароли , прошедшие проверку подлинности на последующих посещениях , и иметь доступ к видом , соответствующих возложенных администратором Siebel , как только они вошли в свой сайт. Например, они могли бы быть в состоянии создать запросы на обслуживание , полную

регистрацию продукта и статус вид заказа. Siebel администратор или обслуживания клиентов представитель можете:

- Добавить или создания новых записей пользователей и профили
- Добавить или изменить логины и пароли
- Редактировать существующую информацию о пользователях, например, счетов

Ответственность и Новая Ответственность, которые возложены на пользователя семени GUESTCST, предназначены для использования с потребительскими заявлениями Финансовых услуг Siebel. Эти обязанности отличаются от обязанностей, возложенных на GUESTCST для потребительских заявлений Siebel, которые не являются определенными для финансовых услуг, как зарегистрировано в другие разделы этого путеводителя.

Если Вы развертываете или Менеджера по корпоративным мероприятиям Siebel для потребительских заявлений Финансов или Siebel, которые не являются определенными для финансовых услуг одновременно ни с какими другими потребительскими заявлениями Финансовых услуг Siebel, то Вы должны создать отдельного анонимного пользователя. Новый анонимный пользователь используется для Менеджера по корпоративным мероприятиям Siebel для Финансов и для потребительских заявлений Siebel, которые не являются определенными для финансовых услуг; то есть, заявления зарегистрированы в другие разделы этого путеводителя. Возложите обязанности на этого анонимного пользователя, поскольку они зарегистрированы для GUESTCST в Данных о Семени. Для получения информации о данных о семени, определенных для приложений Финансовых услуг Siebel, посмотрите Данные о Семени для Финансовых услуг Siebel.

Веб делегированный администратор клиентов является назначенным администратором в компании клиента в сценарии бизнес-для - бизнеса. Этот администратор был присвоен ответственность Siebel , которая делает возможным добавить или изменить пользователей для учетной записи. Эта ответственность Siebel дает им доступ к представлению администрации Пользователь Siebel . ID пользователей и пароли обычно хранятся во внешней системе аутентификации, такие как Netscape Directory Server. Интерфейсный адаптер безопасности Siebel и соответствующие настройки параметров [LDAP] в файле eservice.cfg поддерживать интеграцию между Siebel Видеорешения и внешней системы аутентификации. Благодаря этой поддержке , зарегистрированные пользователи будут проходить проверку подлинности вашей системой аутентификации , а затем разрешение на доступ к приложению Siebel Видеорешения. Настройте ограничения доступа на исполняемые файлы, файлы с данными, Веб-страницы, справочники и административные инструменты следующим образом:

- На каждом сервере, который является частью развертывания Siebel, ограничьте местный пользовательский доступ к справочникам Siebel администраторам Siebel только. Это ограничение предотвращает

посвященные лица с доступом к компьютеру, но без привилегий администратора Siebel, от доступа к чувствительной информации, которая может использоваться, чтобы извлечь пользу, или поднять привилегии Siebel, таким образом позволяя более значительным нарушениям безопасности произойти.

- Для развертывания Siebel, которое хранит очень уязвимые данные или у которого есть другие требования высокой степени безопасности, рекомендуется, чтобы Вы зашифровали Файловую систему Siebel и все диски сервера, содержащие данные о Бизнес-приложениях Siebel, или использующие сторонние продукты или особенности шифрования, обеспеченные Вашей операционной системой.
- Если Вы формируете Siebel-определенные переменные окружения, которые включают уязвимые данные на компьютере, принимающем модуль в развертывании Siebel, например, если Вы осуществили Прикладной диспетчер объектов Конфигурации продукта Siebel на выделенном Сервере Siebel, то шифровка дисков сервера также рекомендуется.

Для получения информации о развертывании Конфигуратора Siebel посмотрите, что Развертывание Siebel Планирует Путеводитель. Для получения информации об урегулировании Siebel-определенных переменных окружения посмотрите Гида Системного администрирования Siebel. Контрольные разрешения файла, собственность файла и доступ к файлу. Ограничьте доступ к счетам и услугам. Управление доступом является важным элементом в поддержании безопасности. Самая безопасная окружающая среда следует за принципом наименьшего-количества-привилегии, который предоставляет пользователям наименьшее количество суммы доступа, который все еще позволяет им закончить свою необходимую работу. Настройте хозяев, чтобы разрешить только те услуги (порты), которые необходимы и управляются только с наименьшим количеством возможных услуг. Устраните услуги с известными слабыми местами. Управляйте полезностью контрольной суммы на системных файлах, когда установлено и проверяйте на троянское вредоносное программное обеспечение часто. (Троянским является программное обеспечение, которое кажется законным, но которое содержит вредоносный код, который используется, чтобы нанести ущерб Вашему компьютеру.) Проверяют пользовательские файловые системы на слабые места и неподходящие средства управления доступом. Проверьте аккаунты операционной системы и удостоверьтесь, что у них есть пароли, которые трудно предположить. Автоматически отключите счета после нескольких неудавшихся попыток логина. Компоненты архитектуры безопасности Siebel включают:

- Пользовательская идентификация для безопасного системного доступа
- Непрерывное шифрование для конфиденциальности данных
- Разрешение для соответствующей видимости данных
- Контрольный журнал для непрерывности данных

- Обеспечьте физическое развертывание, чтобы предотвратить вторжение
- Безопасность для мобильных устройств
- Параметры настройки безопасности web-браузера

Функциональность, чтобы вознаградить участников Лояльности Siebel за общественные действия и приобрести участников лояльности через социальные сети обеспечена методами следующих Веб-сервисов Лояльности Siebel:

- Регистрация контактов как участники лояльности. Используя Siebel CRM Социальная функциональность Интеграции СМИ, контакты Siebel могут быть созданы для пользователей на социальных платформах СМИ, используя социальную информацию о профиле только человека. Эти контакты Siebel могут тогда быть зарегистрированы на программе лояльности, используя функциональность, обеспеченную методом EnrolContactAsMember Пакетного обслуживания Регистрации LOY. Метод MemberEnrollment не может использоваться при этих обстоятельствах, потому что он требует обязательных входов, таких как идентификатор Программы и ID продукта, который не мог бы быть доступен для контактов, созданных, используя социальные данные о профиле.
- Завоевание данных об общественных действиях. Чтобы захватить действия участников лояльности на сайтах социальной сети так, чтобы участникам можно было назначить вознаграждения за те действия, метод ProcessSocialActionTxn Обслуживания Наращивания LOY используется.
- Метод ProcessSocialActionTxn захватил детали общественных действий участников лояльности и создает сделки Лояльности Siebel, которые могут использоваться, чтобы вознаградить признаки обновления или участники.
- Полезные участники с премией направления, когда их друзья регистрируются в продвижениях. Чтобы зарегистрировать участников в продвижение и вознаградить участника, который отослал их, метод PromotionEnrolmentForSMS членского Обслуживания LOY осуществлен.
- Метод PromotionEnrolmentForSMS используется, чтобы зарегистрировать участника на продвижение, наградить участника содействующей премией регистрации и назначить премию направления на ссылающийся домен.

2.8.2. Аутентификация пользователя по обеспечению доступа к ресурсам системы

Siebel Business Applications обеспечивает открытую архитектуру аутентификации , который интегрируется с выбранной инфраструктуры аутентификации клиента. Для получения дополнительной информации см.

подлинности адаптер безопасности и односпальная Веб Sign-On Аутентификация . Siebel Business Applications поддерживает эти типы аутентификации пользователей:

- Проверка подлинности базы данных
- Адаптер безопасности баз данных предоставляется, поддерживает этот тип аутентификации пользователя.
- Протокол доступа к LDAP (Lightweight Directory) и Active Directory Services Interface (ADSI) аутентификации
- LDAP и безопасности ADSI адаптеры предоставляются для поддержки этих типов аутентификации пользователя.
- Веб Single Sign-On (Web SSO)

Siebel Business Applications обеспечивает механизм адаптера безопасности баз данных для учетных сбора и проверки. Форма умолчанию войти собирает Siebel имя пользователя и пароль учетных данных. Адаптер безопасности работает с базовых систем безопасности в базе данных для проверки учетных данных пользователей. При аутентификации базы данных, каждый пользователь должен иметь действительный учетную запись базы данных для доступа к Siebel бизнес-приложений. Администратор базы данных (DBA) должен добавить все счета базы данных пользователя. Развертывание аутентификации базы данных поддерживает хэширования паролей для защиты от хакерских атак. Все Siebel Бизнес Приложения могут использовать проверку подлинности базы данных, которое задается по умолчанию. Тем не менее, некоторые функциональные возможности, предоставляемые Siebel бизнес-приложений, таких как рабочих процессов для поддержки пользователей самостоятельную регистрацию или забытые сценарии Пароль (возможности , обычно используемые в клиентских приложений) , требует аутентификации с помощью LDAP или ADSI адаптеры безопасности. По этой причине аутентификации базы данных редко используется с клиентских приложений.

2.8.3.Адаптеры безопасности для LDAP и аутентификации ADSI

Для сотрудников или клиентов приложений , Siebel Business Applications включает предварительно настроенный интерфейс адаптера безопасности , чтобы позволить организациям воплощать учетных проверку в LDAP или каталога ADSI . Интерфейс подключается к адаптеру безопасности , который содержит логику для проверки учетных данных для конкретной услуги аутентификации. Поэтому Siebel Бизнес Применения клиенты могут проверить учетные данные пользователя с стандартами безопасности, таких как LDAP или ADSI . Siebel Business Applications разработали адаптеры безопасности для ведущих сервисы аутентификации. Интеграция LDAP безопасности адаптер в настоящее время сертифицированы и поддерживаются для Oracle Internet Directory , IBM Directory Server , Novell NDS Edirectory , Sun Java System Directory Server и Microsoft Active Directory . ADSI безопасности интеграция адаптер сертифицирован и поддерживается

Microsoft Active Directory . Для получения информации о сторонних производителей , поддерживаемых или утвержденных для использования с Siebel бизнес-приложений см. в разделе Системные Siebel Требования и поддерживаемые платформы на Oracle Technology Network . Вы также можете создать адаптеры безопасности для поддержки различных технологий аутентификации . Для получения информации о пользовательских адаптеров безопасности см. в разделе Безопасность адаптер SDK.

Рекомендуется, чтобы вы защитили чувствительные данные приложения в базе данных Siebel, шифруя данные. Вы можете зашифровать следующее:

- Определенные области базы данных
- Определенные таблицы базы данных
- Вся база данных

Бизнес-приложения Siebel поддерживают шифрование полевого уровня чувствительной информации, хранившей в базе данных Siebel, например, номерах кредитной карточки или числах национального самосознания. Вы можете формировать Бизнес-приложения Siebel, чтобы зашифровать полевые данные, прежде чем они будут написаны базе данных Siebel, и расшифруйте те же самые данные, когда они восстановлены. Эта конфигурация предотвращает попытки рассмотреть уязвимые данные непосредственно от базы данных Siebel.

Бизнес-приложения Siebel поддерживают использование шифрования данных Advanced Encryption Standard (AES) и алгоритмы RC2. По умолчанию шифрование данных не формируется. Рекомендуется, чтобы Вы установили шифрование данных для деловых составляющих областей, используя Инструменты Siebel. Для получения информации о шифровке данных посмотрите Гида безопасности Siebel.

Когда шифрование полевого уровня осуществлено, данные не расшифрованы, пока это не показано пользователем, у которого есть необходимые привилегии рассмотреть данные. Данные остаются зашифрованными, даже когда они загружены в память, которая увеличивает защиту информации. Однако использование шифрования полевого уровня затрагивает работу.

Как альтернатива шифрованию полевого уровня, Вы можете обеспечить уязвимые данные, используя продукты, такие как следующее:

- Прозрачное Шифрование Данных. Если Вы используете базу данных Microsoft или Oracle с Бизнес-приложениями Siebel, то Вы можете использовать Прозрачную функцию Шифрования Данных, чтобы зашифровать данные в базе данных Siebel. Базы данных Oracle поддерживают использование Прозрачного Шифрования Данных, чтобы зашифровать данные на уровне табличного пространства и колонке. Базы данных Microsoft поддерживают использование Прозрачного Шифрования Данных, чтобы зашифровать данные на уровне базы данных и клетке.

- Прозрачное Шифрование Данных шифрует данные, когда это написано базе данных и расшифровывает его, когда к этому получают доступ Бизнес-приложения Siebel. Прикладные страницы расшифрованы, поскольку они прочитаны и сохранены в памяти в открытом тексте. Поскольку данные не зашифрованы, когда их посылают в Бизнес-приложения Siebel, Вы должны также позволить TLS или SSL защитить связи между сервером и клиентами. Исполнительное воздействие осуществления Прозрачного Шифрования Данных минимально.
- Если Вы позволяете Прозрачное Шифрование Данных, то все резервные копии файла базы данных также зашифрованы. Для получения информации о поддержке Oracle Прозрачного Шифрования Данных пойдите в веб-сайт Oracle Technology Network
- Для получения информации о поддержке Microsoft Прозрачного Шифрования Данных пойдите в веб-сайт Microsoft MSDN
- Oracle Database Vault. Если Вы используете базу данных Oracle с Бизнес-приложениями Siebel, то Вы можете использовать Oracle Database Vault, чтобы ограничить доступ ко всем схемам и объектам в Вашей прикладной базе данных, или к отдельным объектам и схемам пользователей, включая пользователей с административным доступом к базе данных.
- Oracle Database Vault позволяет Вам определять Сферу, границу защиты, вокруг всех или некоторых объектов в Вашей базе данных. Администратор базы данных может работать со всеми объектами в пределах Сферы, но не может получить доступ к данным приложения, которые они содержат. Это ограничение защищает Ваши данные от угроз посвященного лица от пользователей с обширными привилегиями базы данных.
- Вы можете объединить Oracle Database Vault с Прозрачным Шифрованием Данных без потребности в дополнительной конфигурации. Для получения дополнительной информации о Oracle Database Vault пойдите в веб-сайт Oracle Technology Network

2.8.4 Обеспечение Web-сервера Siebel

Поскольку Web-сервер - один из наиболее выставленных и предназначенных злоумышленниками элементов в сети, обеспечивание Web-сервера является приоритетом. Перед использованием Вашего Web-сервера в развертывании Бизнес-приложений Siebel обеспечьте свой Web-сервер, обращаясь рекомендуемый продавцами меры безопасности и методы, как описано в Вашей документации Web-сервера. Тогда считайте осуществление рекомендаций обрисованным в общих чертах в этой теме. Разверните сервер обратный по доверенности в демилитаризированной зоне, чтобы защитить Web-сервер от нападений, касающихся директивное пересечение и отказ в обслуживании. Для получения дополнительной информации посмотрите

Серверы По доверенности. Контролируйте дисковое пространство, доступное на Вашем Web-сервере Siebel. Если Web-серверу позволяют достигнуть предела дискового пространства, то события отказа в обслуживании могут иметь место, когда Веб-клиенты Сервера или Siebel Siebel соединяются с Web-сервером Siebel. Для получения информации об инструментах, которые доступны, чтобы контролировать дисковое использование для Вашего Web-сервера, см. свою документацию продавца Web-сервера. Для продавца документацию безопасности для получения информации об удалении ненужных подкаталогов в окружающей среде Windows. Файл Web-сервера набора и директивные разрешения соответственно, чтобы удостовериться только зарегистрированные пользователи могут получить доступ и изменить определяемые файлы. Справочники Siebel Web Server Extension (SWSE) содержат исполняемые файлы, которые должны быть защищены, например, Явские подлинники (.js файлы), льющиеся каскадом таблицы стилей (.css файлы), и Элементы управления ActiveX (.cab файлы). Используйте следующую процедуру, чтобы назначить разрешения файла Web-сервера в окружающей среде Windows.

Назначать разрешения файла Web-сервера в окружающей среде Windows

Проведите, чтобы Начаться, Программы, Административные Инструменты, и затем выбрать менеджера Internet Information Services (IIS).

Сообщение Siebel телерадиовещательная функциональность позволяет администраторам Siebel показывать важную информацию непосредственно в баре сообщения экранов пользователей. Текст радиопередачи сообщений может быть до 2,000 знаков в длине и может содержать HTML-тэги, которые рассматривают как HTML код на баре сообщения.

Телерадиовещание сообщения доступно для заявлений сотрудника, но не для заявлений клиента или партнера. По умолчанию телерадиовещание сообщения позволено, хотя администратор может позволить или отключить его. В окружающей среде с требованиями очень высокой степени безопасности рекомендуется, чтобы телерадиовещание сообщения было отключено. Для получения информации о выведении из строя телерадиовещания сообщения посмотрите Гида администрации Заявлений Siebel. Внешние деловые компоненты привыкли к данным о доступе, которые проживают в столе нон-Siebel или представлении, используя компонент бизнеса Siebel. Формируя внешние деловые компоненты, Вы должны определить источник данных для внешнего стола, который содержит данные, к которым Вы хотите получить доступ. Чтобы предотвратить пользователей, имеющих необходимость загружаться, получая доступ к внешнему источнику данных, для каждого источника данных, к которому получает доступ внешний деловой компонент, определяют имя пользователя источника данных и детали пароля, используя DSUsername и ценности DSPassword, формируя источник данных, названный подсистемой. DSUsername и параметры DSPassword активированы только, используя адаптер безопасности базы данных. Для получения информации о

формировании внешних деловых компонентов посмотрите Integration Platform Technologies: Интеграция прикладных систем предприятия Siebel.

2.8.5. Web Single Sign-On.

Siebel Business Applications предлагают клиентам возможность позволяет одной логин несколькими веб-приложениями, также известные как веб-Single Sign-On (SSO). Siebel Business Applications обеспечивает настраиваемый механизм для взаимодействия с Web SSO инфраструктур, идентификации пользователей, и при входе пользователей в Siebel бизнес-приложений. С веб-служба SSO , пользователи проходят проверку подлинности независимо от Siebel бизнес-приложений, таких как через службы аутентификации сторонних или через веб-сервер . Oracle имеет альянсы с ведущими поставщиков систем защиты интеграции единого входа. Провайдеры перечислены в категории SSO раствора. Для получения информации о продуктах Oracle Identity Management, которые сертифицированы для использования с Siebel см. в разделе безопасности Поддерживаемые продукты по Siebel. Организации, которые полагаются на предприятии приложения, такие как Siebel часто нуждаются поддерживать несколько удостоверений пользователей и методы аутентификации по целому ряду критических ресурсов. Пользователи обычно есть учетная запись Active Directory и входа в систему Windows, для доступа к Приложения для Windows - основанных необходимые изо дня в день работать . Однако, когда критическое применение необходимо запустить на сервере , не Windows, и он изначально не оборудован для аутентификации по стандартам Windows, пользователи требуют дополнительных логинов. Следовательно, ИТ должны создавать, поддерживать и аудит совершенно отдельный набор идентичностей , которые приводит к неэффективности , вопросы безопасности , и озабоченности по поводу соблюдения . Службы аутентификации обеспечивает решение этих проблем, кроме своих основных возможностей интеграции Unix, Linux и Mac OS X системы с Microsoft Active Справочник для централизованной аутентификации , службы проверки подлинности также обеспечивает Интеграция для ключевых приложений. Через встроенный адаптер для системы безопасности Siebel , Siebel установок, работающих на Unix и Linux могут проходить проверку подлинности с то же самое Каталог войти активность с помощью те же правила и стандарты безопасности что уже в силе в течение Окна войдите в систему. Службы аутентификации является первым и Единственное решение, специально предназначенные для использовать родовое Siebel Security Интерфейсный адаптер 3.0 . API для интеграции Службы проверки подлинности для Siebel Следующее поколение технологии Active Directory моста для Siebel .

Преимущества:

• подлинность приложений Siebel работает на Unix Каталог в простой, эффективной и безопасным способом

• Достижение оптимальной интеграции через Siebel безопасности Active Directory Адаптер на Unix и Apache модуль для единого входа
• Обеспечить соблюдение пароль Active Directory и контроля доступа политика в Приложения Siebel , работающие на Unix или Linux
• Используйте группу Active Directory членство для ролей Siebel
О защите данных от инъекции HTML

Эта тема описывает меры, которые Вы можете принять, чтобы защитить данные приложения Siebel от нападений инъекции HTML.
Показ содержания HTML

Бизнес-приложения Siebel позволяют Вам показывать содержание HTML в областях в пользовательском интерфейсе. Используя объекты Контроля, которые являются полевыми данными, Вы можете установить ценность собственности Режима работы монитора HTML управлять, как значение поля показано в пользовательском интерфейсе. Вы можете определить следующие ценности для собственности Режима работы монитора HTML:

- EncodeData. Если значение поля содержит зарезервированные характеры HTML, то они закодированы, прежде чем они будут показаны так, чтобы HTML показал как текст в пользовательском интерфейсе и не был выполнен как команда HTML. Рекомендуется, чтобы Вы установили собственность Режима работы монитора HTML в EncodeData для каждого объекта Контроля гарантировать, что выполнимые заявления не включены в записи данных Siebel.
- DontEncodeData. Используйте эту стоимость только, когда ценность области - текст HTML, и Вы хотите, чтобы HTML был выполнен. Отбор этой стоимости не рекомендуется, потому что текст HTML может быть объектом злонамеренного вмешательства.
- FormatData. Эта стоимость используется, когда описание или области комментария находятся в расположении только для чтения. Урегулирование FormatData к ИСТИННЫМ данным о причинах, которые будут отформатированы в HTML. Для получения дополнительной информации посмотрите Ссылку Типов Объекта Siebel.

Oracle рекомендует, чтобы Вы рассмотрели все объекты Контроля, собственность Режима работы монитора HTML которых установлена или в DontEncodeData или в FormatData, и рассмотрите изменение ценности собственности к EncodeData. Следующие команды SQL могут использоваться, чтобы возвратить список объектов Контроля, у которых есть имущественный набор Режима работы монитора HTML к ценности или FormatData или DontEncodeData:

```
SELECT
HTML_DISPLAY_MODE
FROM
```

SIEBEL.S_CONTROL
WHERE
HTML_DISPLAY_MODE = 'FormatData' OR
HTML_DISPLAY_MODE = 'DontEncodeData'

Рассмотрите список объектов Контроля, возвращенных в вопросе. Вы не можете изменить ценность собственности Режима работы монитора HTML к EncodeData для всех объектов Контроля в одной операции из заявления Siebel. Собственность должна быть установлена для каждого контроля индивидуально.

Если Вы выбираете, другой метод изменения собственности Режима работы монитора HTML к EncodeData для всех объектов Контроля возвратился в вопросе, то рассмотрите последствия тщательно перед переходом. Рекомендуется, чтобы Вы связались со своим торговым представителем Oracle для Oracle Advanced Customer Services, чтобы просить помощь с этой задачей от Application Expert Services Oracle.

Чтобы усилить Ваше заявление Siebel и данные против нападений, Вы можете определить название каждого из серверов хозяина, которые разрешены для использования с заявлением Siebel. Следующая процедура описывает, как определить названия этих серверов, которым доверяют:

- Начните инструменты Siebel.
- В Исследователе Объекта выберите Прикладной тип объекта.
- Прикладной список появляется.
- Вопрос для названия Вашего заявления Siebel в Редакторе Списка Объекта.
- Например, для применения Call-центра Siebel, подвергните сомнению для Siebel Вещество Universal.
- Захватите прикладной объект.
- В Исследователе Объекта расширьте Прикладной тип объекта, затем выберите Прикладной Пользовательский тип объекта Опоры.
- Прикладной Пользовательский список Опор появляется.
- В Редакторе Списка Объекта добавьте прикладную пользовательскую собственность для каждого сервера, используемого заявлением Siebel. Например:
  Имя: AllowedServerNamesUrl0 value:server_name1
  Имя: AllowedServerNamesUrl1 value:server_name2
- Соберите проект, связанный с применением в файл SRF.

Протокол HTTP поддерживает много методов, которые используются, чтобы определить операцию, которая будет выполнена на ресурсе в Сети. Бизнес-приложения Siebel поддерживают HTTP, получают и отправляют методы только. Все другие методы HTTP заблокированы, чтобы максимизировать безопасность Вашего заявления Siebel. Для получения информации об использовании HTTP получите и отправьте методы с Бизнес-приложениями Siebel, посмотрите транспортные средства и Интерфейсы: Интеграция прикладных систем предприятия Siebel. Продвижения, которые

вознаграждают участников лояльности за их действия в социальных сетях, таких как рейтинг или рассмотрение продукта или продвижения или Симпатии продукта или продвижения. Участники могут также быть вознаграждены за вход в их детали местоположения, или за находящиеся на местоположении действия. Знание местоположения клиента позволяет участникам быть подаренными продвижения, которые являются определенными для того местоположения.

Лояльность Siebel обеспечивает признаки Типа Общественных действий и Кодовый признак Местоположения, чтобы поддержать эту функциональность. Чтобы видеть пример создания продвижения премиальных направлений для общественных действий, посмотрите Гида администрации Лояльности Siebel.

Продвижения, которые вознаграждают участников премия направления, когда их друзья регистрируются в продвижениях в социальных сетях в результате их рекомендаций. Содействующее действие, Премиальный Ссылающийся домен, обеспечено, который поддерживает эту функциональность.

Премиальное действие Ссылающегося домена может также использоваться, чтобы назначить повторяющиеся вознаграждения участнику, который получил начальную премию направления для продолжающихся сделок наращивания отнесенного участника. Для примера создания продвижения премиальных направлений, чтобы вознаградить участников, посмотрите Гида администрации Лояльности Siebel.

Вы можете использовать существующий класс ряда Лояльности Siebel и структуры типа пункта, чтобы создать ряды и типы пункта, которые признают социальное влияние участников и другие действия по социальным сетям. Например, Вы можете определить тип пункта, такой как Социальное Вознаграждение, каким участникам можно поручить вознаградить их за действия по социальным сетям, и Вы можете создать продвижения ряда, например, Фирменного Посла, Фирменного Защитника, и так далее чтобы признать участников, основанных на их социальных признаках профиля. Участникам можно назначить статус ряда согласно числу Социальных накопленных Призовых баллов. Для получения информации о создании рядов и типов пункта, посмотрите Гида администрации Лояльности Siebel.

Поддержка Бизнес-приложений Siebel XML позволяет Вам общаться с любым заявлением Siebel или внешним заявлением, которое может прочитать и написать XML (или произвольный XML или Сибел XML, также известный как формат сообщения Siebel).

Документам XML поставляют непосредственно и из Бизнес-приложений Siebel, или через промежуточное программное обеспечение, используя любые из поддержанных транспортных средств: HTTP, IBM WebSphere MQ, Файл, и так далее. XML, сообщенный таким образом, может подвергнуть сомнению Базу данных Siebel, upsert (обновление или вставка) данные,

синхронизировать эти две системы, удалить данные или выполнить процесс технологического процесса.
Объекты от различных систем, таких как объекты бизнеса Siebel и данные приложения Oracle, могут быть представлены, поскольку интеграция Siebel возражает.
Сибел CRM может также сообщить двунаправлено с использованием Веб-сервисов Simple Object Access Protocol (SOAP) и Представительную государственную Передачу (ОТДЫХ) через Siebel Application Integration (SAI) для Oracle Fusion Middleware. Для получения дополнительной информации посмотрите Integration Platform Technologies: Интеграция прикладных систем предприятия Siebel и Интеграция приложений Siebel для Oracle Fusion Middleware Guide.
Если Вы делаете минимальную установку клиента, удостоверьтесь, что Вы выбираете выбор анализатора XML; иначе, Вы столкнетесь со следующей ошибкой, пытаясь управлять любым процессом клиента, который использует анализатор XML: Неспособный создать Деловую услугу 'EAI XML Конвертер'. Анализатор XML включен по умолчанию в полной установке. Имущественный Тип набора (который наносит на карту к названию элемента XML) и названия отдельных свойств (которые наносят на карту к названиям атрибута XML) не обязательно следует за XML называющие правила. Например, имя может включать знаки, такие как пространство, цитата, двоеточие, левая круглая скобка или правильная круглая скобка, которые не позволены в элементе XML или названиях атрибута XML. В результате Вы должны выполнить некоторое преобразование, чтобы произвести действительный документ XML.
Когда создание документа XML от собственности установило иерархию, Конвертер XML удостоверится, что произведены юридические имена XML. Есть два разных подхода, обеспеченные, чтобы обращаться с переводом имени. Подход определен пользовательской собственностью EscapeNames на обслуживании Конвертера XML. Эта пользовательская собственность может быть или Верной или Ложной. Элемент области интеграции включает ценность указанной области. Это должно появиться в случае его родительского типа объекта интеграции. Если у полевого элемента нет содержания (показанный признаком начала, немедленно сопровождаемым конечным тэгом), он интерпретируется, чтобы означать, что стоимость области должна собираться опустеть. То же самое верно, когда стоимость области пуста; у полевого элемента немедленно будет признак начала сопровождаемым конечным тэгом.
Заказ, в котором области XML появляются в пределах своего родительского составляющего элемента, определен областью Последовательности в определении Инструментов области.
Все области дополнительные. Если полевой элемент не присутствует в составляющем элементе, область не создана в случае объекта интеграции.

3.Исследование влияния алгоритмов шифрования на качество информационных систем

На сегодняшний день существует огромное множество различных алгоритмов и средств защиты информации. Одним из таких способ является криптография с использованием алгоритма SIEBEL.

SIEBEL один из серии алгоритмов по построению дайджеста сообщения, разработанный профессором Рональдом Л. Ривестом из Массачусетского технологического института. Разработан в 1991 году, как более надёжный вариант предыдущего алгоритма. Используется для проверки подлинности опубликованных сообщений путём сравнения дайджеста сообщения с опубликованным.

Процесс шифрования

На вход алгоритма поступает входной поток данных, хеш которого необходимо найти. Длина сообщения может быть любой (в том числе нулевой). Запишем длину сообщения в L. Это число целое и неотрицательное. Кратность каким-либо числам необязательна. После поступления данных идёт процесс подготовки потока к вычислениям.

Шаг 1. Выравнивание потока.

Сначала дописывают единичный бит в конец потока (байт 0x80), затем необходимое число нулевых бит. Входные данные выравниваются так, чтобы их новый размер L' был сравним с 448 по модулю 512 ($L' = 512 \times N + 448$). Выравнивание происходит, даже если длина уже сравнима с 448.

Шаг 2. Добавление длины сообщения.

В оставшиеся 64 бита дописывают 64-битное представление длины данных (количество бит в сообщении) до выравнивания. Сначала записывают младшие 4 байта. Если длина превосходит $2^{64} - 1$, то дописывают только младшие биты. После этого длина потока станет кратной 512. Вычисления будут основываться на представлении этого потока данных в виде массива слов по 512 бит.

Шаг 3. Инициализация буфера.

Для вычислений инициализируются 4 переменных размером по 32 бита и задаются начальные значения шестнадцатеричными числами (шестнадцатеричное представление, сначала младший байт):

A = 01 23 45 67;
B = 89 AB CD EF;
C = FE DC BA 98;
D = 76 54 32 10.

В этих переменных будут храниться результаты промежуточных вычислений. Начальное состояние **ABCD** называется инициализирующим вектором.

Определим ещё функции и константы, которые нам понадобятся для вычислений.

Потребуются 4 функции (1)-(4) для четырёх раундов. Введём функции от трёх параметров - слов, результатом также будет слово.

1 раунд $FunF(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$ (1)

2 раунд $FunG(X,Y,Z) = (X \wedge Z) \vee (\neg Z \wedge Y)$ (2)

3 раунд $FunH(X,Y,Z) = X \oplus Y \oplus Z$ (3)

4 раунд $FunI(X,Y,Z) = Y \oplus (\neg Z \vee X)$ (4)

Определим таблицу констант $T[1..64]$ - 64-элементная таблица данных, построенная следующим образом:

$$T[i] = int(4294967296 \cdot |\sin(i)|), \text{ где } 4294967296 = 2^{32} \quad (5)$$

Выровненные данные разбиваются на блоки (слова) по 32 бита, и каждый блок проходит 4 раунда из 16 операторов. Все операторы однотипны и имеют вид $[abcd\ k\ s\ i]$, определяемый по формуле (6):

$$a = b + ((a + Fun(b,c,d) + X[k] + T[i]) <<< s), \quad (6)$$

где X – блок данных;

$X[k] = M[n \cdot 16 + k]$, где k – номер 32-битного слова из n-го 512 битного блока сообщения;

s – циклический сдвиг влево на s бит полученного 32-битного аргумента.

Шаг 4. Вычисление в цикле.

Заносим в блок данных элемент n из массива. Сохраняются значения A, B, C и D, оставшиеся после операций над предыдущими блоками (или их начальные значения, если блок первый).

AA = A
BB = B
CC = C
DD = D

Суммируем с результатом предыдущего цикла:

A = AA + A
B = BB + B
C = CC + C
D = DD + D

Подробно четыре раунда преобразований показаны на рисунке 1 Приложения А.

После окончания цикла необходимо проверить, есть ли ещё блоки для вычислений. Если да, то изменяем номер элемента массива (n++) и переходим в начало цикла /2/.

3.1 Криптоанализ SIEBEL

На данный момент существуют несколько видов «взлома» хешей SIEBEL – подбора сообщения с заданным хешем.

Перебор, или атака по словарю – метод преодоления криптографической защиты путём перебора большого числа вариантов, однако, проверяются не все возможные варианты, а лишь уже отобранные до этого и загружаемые из списка слов, или словаря. Этот метод оказывается достаточно эффективным потому, что многие люди выбирают в качестве пароля одиночные слова или их простые вариации (например, добавляют к ним одну цифру).

Метод решения задачи путем перебора всех возможных вариантов. Сложность полного перебора зависит от количества всех возможных решений задачи. Если пространство решений очень велико, то полный перебор может не дать результатов в течение нескольких лет или даже столетий.

В криптографии на вычислительной сложности полного перебора основывается оценка криптостойкости шифров. В частности, шифр считается криптостойким, если не существует метода «взлома» существенно более быстрого, чем полный перебор всех ключей. Криптографические атаки, основанные на методе полного перебора, являются самыми универсальными, но и самыми долгими.

Сспециальный вариант таблиц поиска, использующий механизм разумного компромисса между временем поиска по таблице и занимаемой памяти. Радужные таблицы используются для вскрытия паролей, преобразованных при помощи необратимой хеш-функции, а также для атак на симметричные и асимметричные шифры на основе известного открытого текста.

3.2 Описание экспериментальной модели

При создании модели процесса криптоанализа, был использован язык программирования высокого уровня Visual Basic.

Данная программа позволяет произвести анализ криптостойкости различных криптографических алгоритмов, применяемых в современных информационных системах, на серверах, при хранении конфиденциальной информации (пароли, шифры и т. д.).

Интерфейс программы представляет собой окно, включающее рабочую область, панель инструментов и командную строку (рисунок 1 Приложения Б).

Процесс создания модели атаки происходит в следующем порядке:

1) Организуется модель компьютерной сети, состоящей из рабочих серверов, хранящих информацию и подвергающихся атакам (из панели инструментом путем перетаскивания в рабочую область иконок сервера и моделируемых атак).

2) В открывшихся окнах выбираются параметры каждой атаки и каждого сервера по отдельности (рисунки 1- 3 Приложения Г, Д, Е).

3) Выбираются сервера и атаки нажатием на соответствующие иконки.

4) После нажатия кнопки «Start» происходит процесс моделирования атаки на серверы, и выводятся результаты исследования для каждого сервера (рисунок 2 Приложения В).

Изменяя параметры шифрования на сервере и параметры криптоаналитических атак на серверы, можно достичь различных объемов потерянной информации.

Допустимые параметры серверов:

– тип данных: SIEBEL;

– тип шифрования: SIEBEL; SIEBEL + UNIX; SIEBEL + Base 64; SIEBEL + HMAC; SIEBEL + AES;

– количество символов в пароле: вводится целое число (которое ранжируется по категориям: от 0 до 3; от 4 до 8; более 8);

– типы символов в пароле: только цифры; только буквы; буквы и цифры; буквы, цифры и специальные символы.

Допустимые параметры атак:

– тип атаки: Brute; Dictionary Attack; Rainbow Attack;

– длительность атаки: вводится целое число, которое определяет длительность атаки в секундах.

На рисунке 2 Приложения Б показана модель процесса атаки, включающая 5 серверов и 5 атак. Линиями показаны направления атак на сервера.

Результаты моделирования программа выводит в виде таблицы непосредственно на рабочую область (рисунок 2 Приложения В) и дополнительно выводится в текстовой форме в файл Statistics.txt, хранящийся в корневой папке ПО.

Дополнительно программа предусматривает вызов Кодировщика SIEBEL – модуля, позволяющего находить SIEBEL любой введенной текстовой информации (рисунок 3 Приложения В).

3.3 Экспериментальная часть

Для выявления наиболее криптостойкого алгоритма была создана модель на разработанном ПО, состоящая из 5 серверов и 5 криптоаналитических атак (рисунок 1 Приложения В).

Параметры использованных серверов и атак показаны в Приложениях Г, Д, Е.

Таким образом, для каждого сервера обеспечивалось три типа атаки длительностью в 600 секунд.

Таким образом, в эксперименте рассматриваются все пять типов шифрования (SIEBEL, SIEBEL Base64, SIEBEL Unix, SIEBEL HMAC, SIEBEL AES), работающих с паролями различных длин с использованием букв, цифр и специальных символов. При этом каждый сервер (тип защиты) подвергается всем трем типам криптографического анализа (Brute-Force, Dictionary attack, Rainbow attack) одной длительности (10 минут).

3.4 Анализ результатов эксперимента по уязвимости

Наиболее уязвимым явилось хеширование без вторичного шифрования (75% потерянной информации). SIEBEL Base64, SIEBEL HMAC SIEBEL Unix потеряли 35,4%, 17,5% и 16,6% информационных битов соответственно. На рисунках 3.1, 3.2 показаны соотношения потерянных битов для вышеперечисленных алгоритмов. Результат доказывает гипотезу о недостаточной криптостойкости применяемого на сегодня алгоритма хеширования SIEBEL.

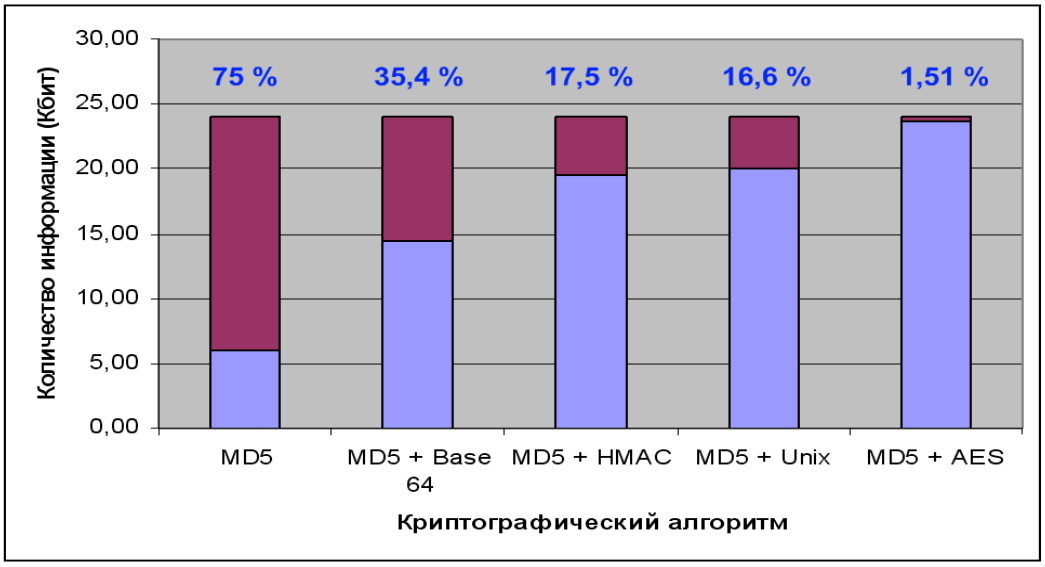


Рисунок 3.2 – Доля потерянной (красный маркер) в общем объеме переданной (синий маркер) информации

3.5 Анализ результатов эксперимента по эффективности от криптоаналитических атак

Вторым результатом стало выявление характеристик эффективности рассмотренных способов криптографического анализа. Были построены графики эффективности* криптоаналитических атак в зависимости от способа вторичного шифрования (рисунок 3.3), длины пароля и

* За главный показатель эффективности метода бралась вероятность расшифровки бита информации P(x)

используемых в нем символов (рисунок 3.4). При этом из рисунка 3.3 можно говорить о линейном спаде эффективности все трех способ криптоанализа по мере движения от SIEBEL к SIEBEL AES. График подтверждает гипотезу о возможности увеличения криптостойкости путем вторичного шифрования хеш-сумм.

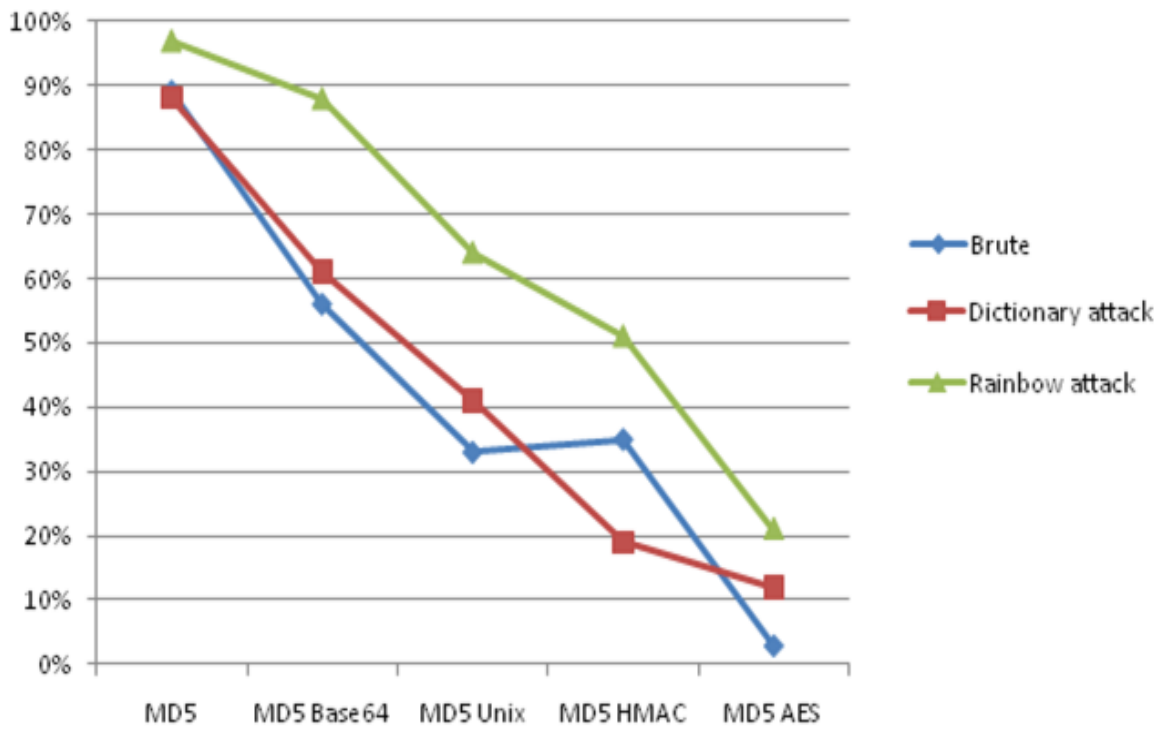


Рисунок 3.3 – Эффективность различных видов атак от способа шифрования

3.6 Анализ результатов эксперимента по эффективности различных видов атак от типа пароля

Из рисунка 3.4 можно заключить, что Brute attack наиболее эффективен при криптоанализе простых паролей (малой длины и без использования специальных символов). Dictionary attack следует использовать при расшифровке с ограничением по времени, когда в пароле присутствуют специальные символы. Наибольшей эффективностью обладает Rainbow attack. Данный метод было бы разумно применять при расшифровке сложных паролей с высокой криптостойкостью.

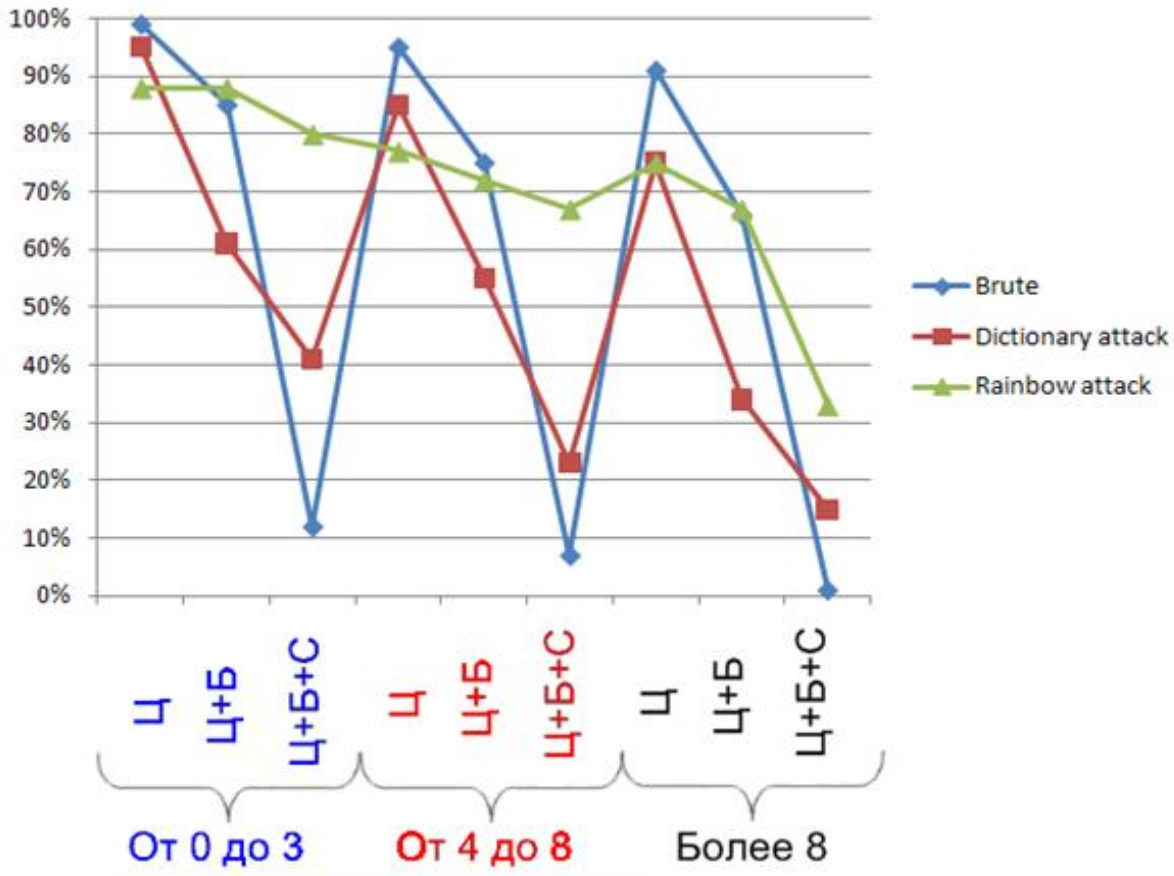


Рисунок 3.4 – Эффективность различных видов атак от типа пароля

3.7. Анализ результатов эксперимента эффективности по времени

Следующим результатом эксперимента стало вычисление необходимы временных затрат, связанных с дополнительным шифрованием конечных сумм данных. На рисунке 3.5 показана гистограмма, иллюстрирующая временные задержки[**].

Из рисунка видно, что по мере увеличения криптостойкости алгоритмов, растет и время обработки информации. При этом алгоритм SIEBEL Base64 требует в 1,5 раза больше времени на обработку данных, алгоритм SIEBEL Unix – 2,25, SIEBEL HMAC – в 2,3 и SIEBEL AES – в 2,5 раза.

В таблице 3.1 приведены результаты проведенного эксперимента.

[**] На графике время показано в процентном соотношении, поскольку количественные показатели зависят от производительности конечного оборудования. При этом 100% времени бралось за временную задержку при шифровании SIEBEL без применения дополнительных алгоритмов.

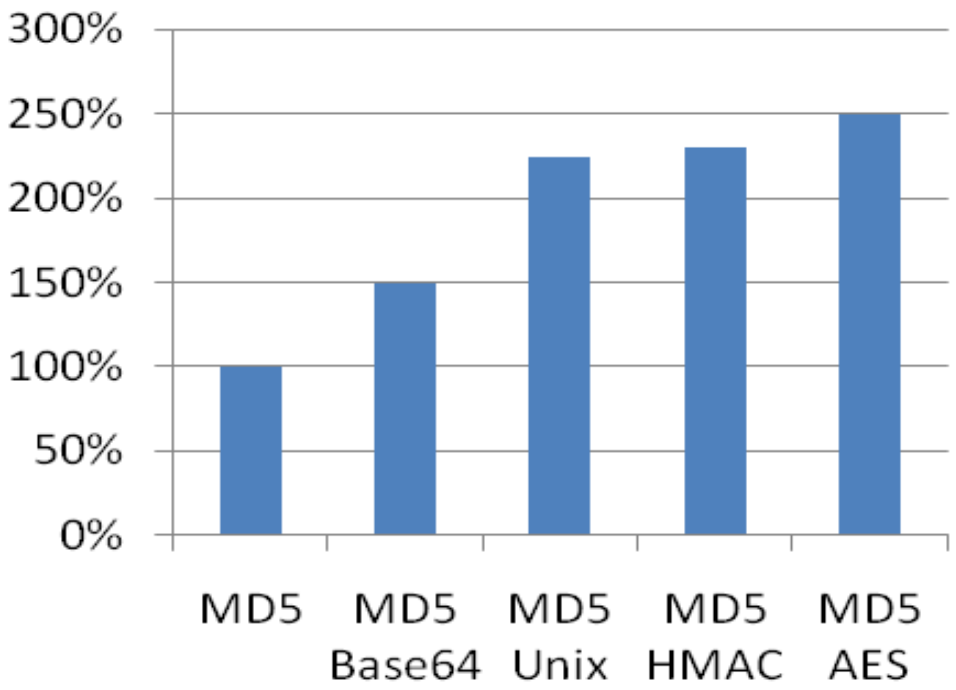


Рисунок 3.5 – Гистограмма дополнительных затрат времени на вторичное шифрование

3.8 Анализ результатов эксперимента надежности

По результатам эксперимента наиболее надежным алгоритмом шифрования стал SIEBEL AES (1,51% потерянных данных) рисунок 3.1.

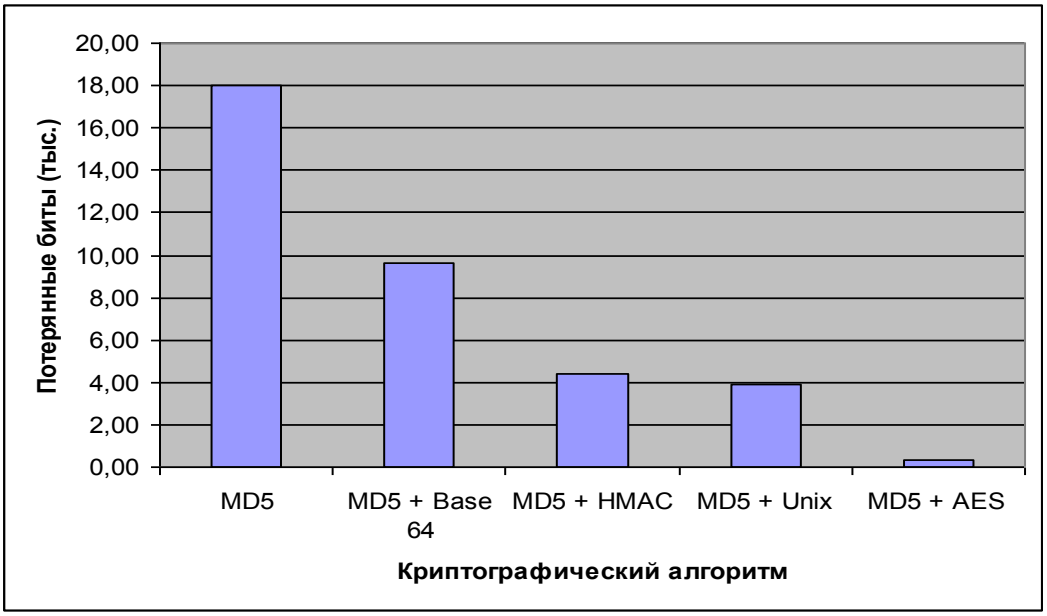


Рисунок 3.1 – Количество потерянной информации

Таблица 3.1 – Результаты эксперимента

№ сервера	Тип применяемого шифрования	Тип пароля	Объем потерянной информации, Кбит	Объем сохраненной информации, Кбит	Эффективность Brute-Force атаки, %	Эффективность Dictionary атаки, %	Эффективность Rainbow атаки, %	Временная задержка, %
1	SIEBEL	9 символов, только буквы	18,0	6,0	89	88	97	100
2	SIEBEL Base64	8 символов, цифры и буквы	8,496	15,504	56	61	88	15-
3	SIEBEL Unix	11 символов, только цифры	3,984	20,016	33	41	64	225
4	SIEBEL HMAC	4 символа, цифры, буквы и спец. символы	4,2	19,8	35	19	51	230
5	SIEBEL AES	12 символов, цифры, буквы и спец. символы	0,362	23,638	3	12	21	250

Приложение А

```
function WebApplet_PreInvokeMethod (MethodName)
{
	if(MethodName == "ShowPopup")
	{
		var ChannelNum = this.BusComp().GetFieldValue("CSA Channel
Number");
		if (ChannelNum == "")
		TheApplication().RaiseErrorText("Необходимо указать номер
канала");
		return(ContinueOperation);
	}
	if ((MethodName == "ModifyProdSvc") || (MethodName ==
"DisconnectProdSvc"))
	{
		var ChannelNum = this.BusComp().GetFieldValue("CSA Channel
Number");
		if (ChannelNum == "")
		TheApplication().RaiseErrorText("Необходимо указать номер
канала");
		var ChannelId = this.BusComp().GetFieldValue("Channel Id");
		this.BusComp().ActivateField("Integration Id");	var IntegrationId =
this.BusComp().GetFieldValue("Integration Id");

		if((ToNumber(ChannelId.length) > 0) && (ChannelId != "No Match
Row Id")) ///у устройства есть связка с каналом
		{
			var strStatusClosed =
TheApplication().InvokeMethod("LookupValue", "FS_ORDER_STATUS",
"Closed"); //Завершено
			var strStatusAbandoned =
TheApplication().InvokeMethod("LookupValue", "FS_ORDER_STATUS",
"Abandoned"); //Отменен
			var strStatusRejected =
TheApplication().InvokeMethod("LookupValue", "FS_ORDER_STATUS",
"Rejected"); //Отказано
			var strStatusBilling_Accepted =
TheApplication().InvokeMethod("LookupValue", "FS_ORDER_STATUS",
"Billing Accepted"); //Принят в биллинге
			var strStatusTP_Rejected =
TheApplication().InvokeMethod("LookupValue", "FS_ORDER_STATUS", "TP
Rejected"); //Нет ТВ
```

```
var strStatusTP_Approved =
TheApplication().InvokeMethod("LookupValue", "FS_ORDER_STATUS", "TP
Approved"); //Есть ТВ
var strStatusTP_Check =
TheApplication().InvokeMethod("LookupValue", "FS_ORDER_STATUS",
"Event Check"); //Проверка ТВ
var strStatusOpen =
TheApplication().InvokeMethod("LookupValue", "FS_ORDER_STATUS",
"Open"); //Исполнение

//DIVLEV\\\11_07_12\\\CSA-589\\\Проверка открытых
заявок при создании заказа
var bcWorkOrder = TheApplication().GetBusObject("Com
Work Order").GetBusComp("CSA Com Work Order - Orders Light");
bcWorkOrder.ActivateField("Status");
bcWorkOrder.ActivateField("Order Number");
bcWorkOrder.ClearToQuery();
bcWorkOrder.SetSearchExpr("[CSA Channel Id] = '"+
ChannelId +"' AND [Status] = '"+ strStatusOpen +"'");
bcWorkOrder.ExecuteQuery(ForwardOnly);
if (bcWorkOrder.FirstRecord())
{
    var strOrderNumber =
bcWorkOrder.GetFieldValue("Order Number");
    bcWorkOrder = null;
    TheApplication().RaiseErrorText("На данном канале
существует открытая рабочая заявка №"+ strOrderNumber);
}
else bcWorkOrder = null;
//

var bcLineItems = TheApplication().GetBusObject("CSA Order
Line Item").GetBusComp("CSA Order Entry - Line Items Light");
bcLineItems.ActivateField("Status");
bcLineItems.ActivateField("Order Number");
bcLineItems.ActivateField("Asset Integration Id");
var strItemType = "";
var OrderNumber  = "";
bcLineItems.ActivateField("Action Code");
bcLineItems.ClearToQuery();
//bcLineItems.SetSearchSpec("CSA Channel Id", ChannelId);
bcLineItems.SetSearchSpec("Asset Integration Id",
IntegrationId);
bcLineItems.ExecuteQuery(ForwardOnly);
```

```
                if (bcLineItems.FirstRecord())
                {
                        do
                        {
                                var Status = bcLineItems.GetFieldValue("Status");
                                if(Status != strStatusClosed && Status !=
strStatusAbandoned && Status != strStatusRejected && Status !=
strStatusBilling_Accepted)
                                {
                                        strItemType =
bcLineItems.GetFieldValue("Action Code");
                                        OrderNumber =
bcLineItems.GetFieldValue("Order Number");
                                        bcLineItems = null;
                                        TheApplication().RaiseErrorText("Для
данного устройства есть открытый заказ №"+ OrderNumber +" типа '"+
strItemType +"'!");
                                }
                        }while (bcLineItems.NextRecord())
                }
                bcLineItems = null;
                return (ContinueOperation);
        }
    }
    else
    if(MethodName == "DisconnectAsset")
    {
        var ChannelNum = this.BusComp().GetFieldValue("CSA Channel
Number");
        if (ChannelNum == "")
        TheApplication().RaiseErrorText("Необходимо указать номер
канала");
    //  if(TheApplication().GetProfileAttr("Login Name") != "VKIM" &&
TheApplication().GetProfileAttr("Login Name") != "AIKA" &&
TheApplication().GetProfileAttr("Login Name") != "FROGOZIN")
        //  TheApplication().RaiseErrorText("Метод недоступен");

        DisconnectAsset();
        return (CancelOperation);
    }
```

ПРИЛОЖЕНИЕ Б

```
function BusComp_PreSetFieldValue (FieldName, FieldValue)
{
	/*	if(//FieldName == "CSA Port Number" || FieldName == "CSA
Port Number_2"||
		FieldName == "CSA Connection Point")
	{
		var sFieldName = '';
		switch(FieldName)
	{
		//case "CSA Port Number": sFieldName = "'№ Порта_1'";
break;
			//case "CSA Port Number_2": sFieldName = "'№
Порта_2'"; break;
			case "CSA Connection Point": sFieldName = "'Тел.
точки подкл.'"; break;
	}
	var sText = FieldValue;
	var sChar, i;
	for (i=0; i < sText.length; i++)
	{
	sChar = sText.charAt(i);
	  if(isNaN(parseInt(sChar)))
		TheApplication().RaiseErrorText("Поле "+ sFieldName +"
может содержать только числовые значения.");
	}
   }*/



	if(FieldName == "CSA Login")
	{
		var bcChannel = TheApplication().GetBusObject("CSA
Channel").GetBusComp("CSA Channel");

		with(bcChannel)
		{
			ClearToQuery();
			ActivateField("CSA Login");
			SetSearchExpr("[CSA Login] = '"+FieldValue+"'");
			ExecuteQuery();

			if(FirstRecord())
```

```
                        {
                                TheApplication().RaiseErrorText("Данный Логин
уже используется");
                        }
                }
                bcChannel = null;
        }
        else if(FieldName == "CSA Town CD" && this.GetFieldValue("CSA
Town CD")!="")
        {
                var ChannelId =
TheApplication().ActiveBusObject().GetBusComp("CSA
Channel").GetFieldValue("Channel Orig Id");
                var bcChannel = TheApplication().GetBusObject("CSA
Channel").GetBusComp("CSA Channel");

                bcChannel.ClearToQuery();
                bcChannel.SetViewMode(AllView);
                bcChannel.ActivateField("CSA Town CD");
                bcChannel.SetSearchExpr("[Id] = '"+ChannelId+"'");
                bcChannel.ExecuteQuery(ForwardOnly);
                var Channel = bcChannel.GetSearchExpr();

                if(bcChannel.FirstRecord())
                {
                        if(bcChannel.GetFieldValue("CSA Town CD") !=
FieldValue)
                                TheApplication().RaiseErrorText("Выбранный
адрес не соответствует городу "+bcChannel.GetFieldValue("CSA Town
CD"));
                }
                bcChannel = null;
        }
        return (ContinueOperation);
}
```

Приложение В

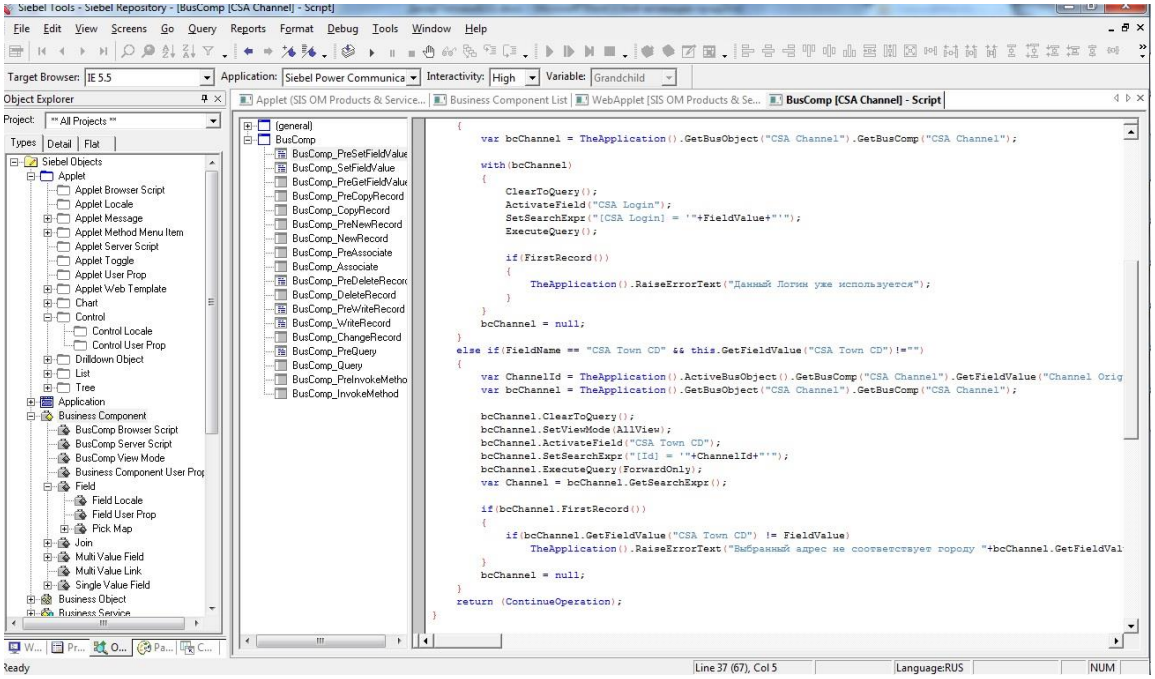


Рисунок 1 – Экспериментальная модель

Сервер #4

Шифр - MD5 + HMAC

Пароль - Цифры, буквы и спец.символы

Тип атаки	Время атаки	Потерянная информация
Rainbow	600	240000
Dictonary	600	150000
Brute	600	30000
Total	1800	420000

Рисунок 2 – Пример вывода результатов моделирования атаки на сервер

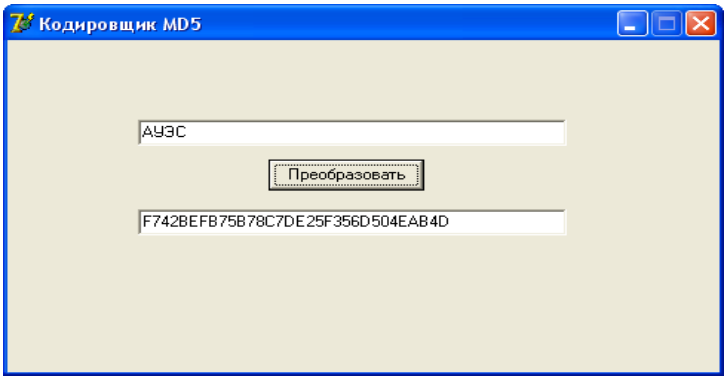


Рисунок 3 – Диалоговое окно разработанного Кодировщика SIEBEL

Приложение Г

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{

	if(MethodName == "CreateRequest")
	{
		var bcChannel = TheApplication().GetBusObject("CSA Channel
BO").GetBusComp("CSA Channel For Distribution");
		var Query = Inputs.GetProperty("Query");
		var LoginId = Inputs.GetProperty("LoginId");
		var LoginName = Inputs.GetProperty("LoginName");
		var Count = 0;
		var psAccId = TheApplication().NewPropertySet();

		bcChannel.ClearToQuery();
		bcChannel.SetViewMode(AllView);
		bcChannel.ActivateField("CSA Account Id");
		bcChannel.ActivateField("CSA Account Email Flg");
		bcChannel.SetSearchExpr(Query);
		bcChannel.ExecuteQuery(ForwardOnly);

			if(bcChannel.FirstRecord())
			{
				do
					{
						var AccountId =
bcChannel.GetFieldValue("CSA Account Id");
						var EmlFlg =
bcChannel.GetFieldValue("CSA Account Email Flg");
						if(AccountId != "" && EmlFlg == "Email
Info")
						{
							psAccId.SetProperty(AccountId,
AccountId);
						}

					}while(bcChannel.NextRecord());
				}

					var bcCommRequest =
TheApplication().GetBusObject("Comm Request").GetBusComp("Comm
Request");
```

```
var bcCommRequestSrc =
TheApplication().GetBusObject("Comm Request").GetBusComp("Comm Request
Source");
bcCommRequest.NewRecord(0);

var ComReqId =
bcCommRequest.GetFieldValue("Id");
Outputs.SetProperty("CommReqId", ComReqId);

bcCommRequest.SetFieldValue("Name","Default Name");
bcCommRequest.SetFieldValue("Recipient
Group","CSA Account");
bcCommRequest.SetFieldValue("Login Id",
LoginId);

bcCommRequest.SetFieldValue("LoginName", LoginName);
bcCommRequest.SetFieldValue("CSA
Query Text", Query);
bcCommRequest.WriteRecord();


var strPrName;
var ContId = "1-UVX2N";
strPrName = psAccId.GetFirstProperty();

while ((strPrName != "") && (strPrName !=
null))
{
bcCommRequestSrc.NewRecord(0);

bcCommRequestSrc.SetFieldValue("Comm Request Id", ComReqId);

bcCommRequestSrc.SetFieldValue("Source Row Id",
psAccId.GetProperty(strPrName));
bcCommRequestSrc.WriteRecord();
strPrName = psAccId.GetNextProperty();
}
bcCommRequestSrc.NewRecord(0);

bcCommRequestSrc.SetFieldValue("Comm Request Id", ComReqId);

bcCommRequestSrc.SetFieldValue("Source Row Id", ContId);
bcCommRequestSrc.WriteRecord();
```

```
                bcChannel = null;
                psAccId = null;
                bcCommRequest = null;
                bcCommRequestSrc = null;
                strPrName = null;

                return (CancelOperation);
            }
    return (ContinueOperation);
}
```

ЗАКЛЮЧЕНИЕ

В данной работе были изучены характеристики алгоритмов LDAP/ADSI, Web Single Sign-On, Security SDK..
Алгоритмы LDAP/ADSI, Web Single Sign-On практически равнозначны по совокупности характеристик, за исключением алгоритма Security SDK, имеющего существенно больше недостатков, в том числе, алгоритм практически нереализуем в условиях ограниченных ресурсов.

Была изучена и доработана парольная система авторизации с применением технологии шифрования методом “нелинейной динамики” captcha. Так же алгоритм был модифицирован двухфакторной аутентификации.

В процессе исследования были изучены 5 основных видов криптографических алгоритмов, применяемых при хешировании SIEBEL в современных информационных сетях, оперирующих конфиденциальной информацией: SIEBEL, SIEBEL Base 64, SIEBEL Unix, SIEBEL HMAC, SIEBEL AES.

Было разработано специальное программное обеспечение, которое позволяет создавать модели атак на серверы различными способами анализировать криптостойкость алгоритмов, используемых в современных информационных сетях, имеющих дело с данными аутентификации (пароль, имя учетной записи).

В эксперименте была рассмотрена модель атаки на серверы различными возможными способами. На каждом сервере были установлены уникальные параметры шифрования информации. Каждый сервер был подвергнут трем видам атак в течение 600 секунд. По итогам эксперимента самым эффективным оказался SIEBEL с шифрованием по алгоритму AES. Самый эффективный способ атаки – Rainbow Attack. Наименее криптостойкая система защиты – SIEBEL без дополнительного шифрования. При этом различные способы криптоанализа имеют разную эффективность в зависимости от способа защиты информации. Так, например, наиболее эффективным при шифровании SIEBEL является Brute, Dictionary attack имеет хорошую эффективность по взлому паролей с применением специальных символов, а Rainbow attack лучше применять при расшифровке сообщений со сложным шифрованием.

Однако, несмотря на все плюсы сложных криптографических алгоритмов (в частности SIEBEL Unix и SIEBEL HMAC), потери времени, которые возникают при их использовании, могут создать коллапс в информационных системах. Это, пожалуй, является основным препятствием в использовании вторичного шифрования. Ведь такие крупные системы, как ВКонтакте или Одноклассники не могут себе позволить увеличения времени обработки данных в 2 или 2,5 раза. Поэтому столь сложные алгоритмы было бы разумнее использовать в системах, требующих большого уровня защищенности конфиденциальных данных (например, в системах электронной оплаты). А для сетей с большим количеством пользователей

(социальные сети, почтовые сервисы) приемлемо было бы использование алгоритма SIEBEL Base64, обеспечивающего повышение криптографической стойкости в 2 раза по сравнению с хешированием SIEBEL без дополнительного шифрования (35,4% по сравнению с 75% потерянной информации). Время обработки информации при этом возрастает в 1,5 раза – приемлемые потери при двойном повышении защищенности данных.

Модель, разработанная в процессе исследования, полностью работоспособная и может использоваться для проведения дальнейших исследований. В будущем планируется подключение модулей взаимодействия с другим ПО, расширение возможности атак и защиты серверов, расширение функциональности (добавление возможности моделирования частных ЛВС, корпоративных сетей и т. д.), увеличение стабильности работы программы.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. AES Round 1 Information. // http://csrc.nist.gov — January 26, 2001.
2. Anderson R., Biham E. Two Practical and Provably Secure Block Ciphers: BEAR and LION. //http://citeseer.ist.psu.edu — 1995.
3. Anderson R., Biham E., Knudsen L. Serpent: A Proposal for the Advanced Encryption Standard. // http://csrc.nist.gov.
4. Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard (AES). // http://csrc.nist.gov — Department of Commerce — National Institute of Standards and Technology — Federal Register: September 12, 1997.
5. Biham E., Dunkelman O., Keller N. Differential-Linear Cryptanalysis of Serpent. //http://citeseer.ist.psu.edu — Technion, Haifa, Israel.
6. Burwick C., Coppersmith D., D'Avignon E., Gennaro R., Halevi S., Jutla C., Matyas S.M.Jr., O'Connor L., Peyravian M., Safford D., Zunic N. MARS — a candidate cipher for AES. //http://www.ibm.com — IBM Corporation — Revised, September, 22 1999.
7. Courtois N., Castagnos G., Goubin L. What do DES S-boxes Say to Each Other? //http://eprint.iacr.org — Axalto Cryptographic Research & Advanced Security, Cedex, France.
8. Daemen J., Knudsen L., Rijmen V. The Block Cipher Square. //http://www.esat.kuleuven.ac.be.
9. FIPS 46-3. Data Encryption Standard (DES). // http://csrc.nist.gov — Reaffirmed 1999 October 25.
10. FIPS 81. DES Modes of Operation. // http://csrc.nist.gov — 1980 December 2.
11. Kocher P.C. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. // http://citeseer.ist.psu.edu — Cryptography Research, Inc., San Francisco, USA.
12. Nechvatal J., Barker E., Bassham L., Burr W., Dworkin M., Foti J., Roback E. Report on the Development of the Advanced Encryption Standard (AES). // http://csrc.nist.gov — National Institute of Standards and Technology.
13. Nechvatal J., Barker E., Dodson D., Dworkin M., Foti J., Roback E. Status report on the first round of the development of the advanced encryption standard. // http://csrc.nist.gov — National Institute of Standards and Technology.
14. Rivest R.L., Robshaw M.J.B., Sidney R., Lin Y.L. The RC6 Block Cipher. //http://www.rsasecurity.com — Version 1.1 — August 20, 1998.
15. Schneier B. Description of a new variable-length key 64-bit block cipher (blowfish). //http://www.schneier.com.
16. Schneier B., Kelsey J., Whiting D., Wagner D., Hall C., Ferguson N. Twofish: A 128-bit Block Cipher. // http://www.schneier.com — 15 June 1998.
17. What are RC5 and RC6? // http://www.rsasecurity.com.
18. Панасенко С. Алгоритм шифрования DES и его варианты. // Connect! Мир связи. — 2006 — №№ 3-6.
19. Панасенко С. Интересные алгоритмы шифрования, часть 2. // BYTE/Россия. — 2006 — № 5 — с. 74-79.
20. Панасенко С.П., Батура В.П. Основы криптографии для экономистов: учебное пособие. Под ред. Л.Г. Гагариной. — М.: Финансы и статистика, 2005 — 176 с.
21. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. — М.: ДМК Пресс, 2002 — 656 с.
22. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. — Пер. с англ.: М.: Издательство ТРИУМФ, 2002 — 816 с.
23. Галатенко В.А., Основы информационной безопасности. – М.: ИНТУИТ.РУ «Интернет-Университет Информационных Технологий», 2003. – 280 стр.

24. Лапонина. О.Р. Криптографические основы безопасности. – М.: ИНТУИТ.РУ «Интернет- Университет Информационных Технологий», 2003. – 250 стр.
25. Сайт http://www.habrahabr.ru/hub/infosecurity/2346325.php.
26. Сайт http://www.securitylab.ru/news/405505.php.
27. Фролов А., Фролов Г. Практика применения Perl, PHP, Apache и MySQL для активных Web-сайтов. – М.: Русская Редакция, 2002. – 300 стр.
28. Базылов К.Б., Алибаева С.А., Бабич А.А. Методические указания по выполнению экономического раздела выпускной работы бакалавра для студентов всех форм обучения специальности 050719 – Радиотехника электроника и телекоммуникации – Алматы: АИЭС, - 2008. – 19 стр.
29. Дюсебаев М.К., Бегимбетова А.С. Методические указания к выпускной работе (для студентов всех форм обучения специальностей 050719 – Радиотехника электроника и телекоммуникации, 050704 – Вычислительная техника и программное обеспечение) Алматы: АИЭС, 2008. – 10 стр.
30. Кнорринг Г.Н. Справочная книга для проектирования электрического освещения. – Л.: Энергия, 1986. – 150 стр.
31. А.А. Байзаков, А.С. Бегимбетова, М.К. Дюсебаев, Т.С. Санатова. Охрана труда. Методические указания к выполнению лабораторных работ (для студентов всех специальностей очно-заочной формы обучения). – Алматы: АИЭС, 2004 – 44 стр.
32. Баклашов Н.И., Н.Ж. Китаева, Б.Д. Терехов. Охрана труда на предприятиях связи и охрана окружающей среды: Учебник для вузов – М.: Радио и связь, 2000г. – 288 стр.
33. Дунаев В.В. Сценарии для Web-сайта: PHP и JavaScript. 2-е изд. – СПб. BHV-Петербург, 2008. – 280 стр.
34. Дронов В.А. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов – СПб. BHV-Петербург, 2007. – 360 стр.
35. ГОСТ 30494-96 Здания жилые и общественные. Параметры микроклимата в помещениях. – 15 стр.
36. СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Общие требования. -М.: Сройиздат, 2002. – 100 стр.