Colibri chat v1.2

© Podobashev Dmitry / BEOWOLF, 2010

Что такое Colibri chat.

Colibri chat – это интернет-пейджер для мгновенного обмена текстовыми сообщениями.

Возможности программы.

- Общение на каналах с несколькими пользователями одновременно (конференции).
- Приватное общение двух пользователей.
- Обмен сообщениями во всплывающих окнах, которые могут отображаться поверх всех остальных окон Windows.
- Посылка надписей поверх экрана десктопа.
- Текст обмена сообщениями может быть с полноценным форматированием, в формате RTF. Таким образом, можно например, скопировать текст из документа MS Word, и передать с исходным форматированием на канал, приватным сообщением, или отобразить на экране десктопа.
- Передача всего содержимого буфера обмена Windows от одного пользователя другому.
- Конфигурирование каналов. Возможность анонимного общения на каналах.
- Широкие возможности модерирования: шесть уровней разделения прав доступа на каналах, а также возможность назначения суперпользователей с сервера.

Принцип работы.

Colibri chat работает по принципу клиент-серверного взаимодействия, клиентская часть программы взаимодействует с серверной через протокол TCP/IP. Одна или несколько копий клиента программы соединяются с сервером и обмениваются транзакциями.

Особенности программы.

Ключевой концепцией проекта было создать клиент-серверный чат с максимально простой функциональностью. Эта идея была успешно реализована, из всех настроек необходимо изначально определить только один адрес сервера, и при необходимости ник, нажать ввод – это всё что требуется сделать для начала работы. Остальные параметры работы восстанавливаются из предыдущей сессии работы программы, либо определяются автоматически со стороны клиента или сервера. Программа не задаёт никаких вопросов.

Сетевая безопасность.

Сетевой протокол реализует высокую степень конфиденциальности переписки. Это достигается за счёт того, что, во-первых, применяется собственный программный бинарный протокол передачи данных, во-вторых, используется сжатие данных, в-третьих, сжатые данные - шифруются, для аутентификации переданной информация используется CRC-код.
Рассмотрим механизм сетевой безопасности подробнее. При соединении клиента и сервера обе стороны получают «закрытый» ключ шифрования по алгоритму Диффи-Хеллмана, формируемый с использованием Skein-хеша от текстового пароля. Все транзакции - сжимаются, после сжатия всё содержимое - шифруются, включая CRC-код и заголовок транзакции. Таким образом, весь трафик представляет собой непрерывный поток зашифрованных данных. Заголовки транзакций не имеют каких-либо констант, и их размер варьируется от содержащихся в них битовых полей. Для шифрования по умолчанию применяется поточный шифр HC-256, также доступны другие поточные шифры проекта eSTREAM: Rabbit. При шифровании каждой транзакции используется уникальный «открытый» ключ шифрования, который не передаётся по сети. «Открытые» ключи создаются генератором псевдослучайных чисел, с помощью хеш-функции Skein, причём для каждого соединения инициируется последовательность на основе «закрытого» ключа. Размер «закрытого» и «открытого» зависит от алгоритма шифрования, и указывается при его выборе. Например, при использовании шифра HC-256 все используемые ключи - 256-битные.
Эти меры позволяют гарантированно выполнить свою задачу - обеспечить конфиденциальность и безопасность передачи данных по сети.

Выбор пароля. По статистике большинство пользователей используют легко запоминающиеся пароли, составленные из своего дня рождения, номера телефона, или записанный транслитом свой ник / имя / фамилию, либо совсем простые пароли наподобие 111111, 123456, password1, qwerty, и т. п. Все подобные пароли можно легко угадать, и при атаке шифра по словарю они перебираются первыми. Также не нужно доверять генераторам паролей на основе некриптостойких генераторов псевдослучайных чисел. Если, например, известно, что пароль выбран с использованием генератора на основе текущего времени, то зная дату регистрации, можно повторить последовательность паролей и «взломать» шифр. При выборе пароля лучше всего использовать длинную случайную последовательность символов, которая может содержать буквы различных алфавитов, цифры и другие символы. В данной программе для паролей допускаются любые символы Unicode. Например, если вы в качестве пароля используете последовательность из 6 заглавных латинских букв, то количество возможных комбинаций будет 26^6, если при этом используете также цифры и строчные латинские буквы, то количество возможных комбинаций возрастёт до 62^6. Шифр, защищённый паролем не более 6 буквенно-цифровых символов, что соответствует 36-битному ключу, может быть взломан за несколько минут на одном персональном компьютере путём полного перебора всех комбинаций. На сегодняшний день считается надёжным шифрование, защищенное, по крайней мере, 80-битным ключом, что соответствует текстовому паролю из 14 буквенно-цифровых символов.

Сравнительные характеристики.

Существует выбор между различными программными решениями, служащими для общения по локальной сети и через интернет. Эти решения предоставляют выбор между альтернативными возможностями, которые удовлетворяют разным вкусам и соответствуют различным потребностям. Colibri chat - обладает своими особенностями и уникальными техническими характеристиками по сравнению с аналогами. Сравним их на примере двух известных заменителей Colibri chat:

	Colibri chat	Vypress Chat	mIRC (RFC 1459)
Дата разработки сетевого протокола	2010 г. (версия 1.2)	1997 г. (версия 1.9 - 2003 г.)	1989 г. (последний IETF draft - 2000 г.)
Сервер платный?	Нет	Нет (сервер не нужен)	Для Windows платформы платный, для бесплатных версий UNIX – бесплатный
Клиент платный?	Нет	Да - от $12 и ниже (клоны для бесплатных UNIX - бесплатные)	Да - от $30 долларов
Максимальное количество поддерживаемых клиентов	Оптимально - не более 1000 на 1 сервер	Оптимально - не более 1000	Теоретически не ограничено, на практике зависит от реализации сервера
Поддержка Unicode	Есть	Есть	Нет
Поддержка RTF	Есть	Нет	Нет
Максимальная длина псевдонима пользователя	Регулируется на сервере (20 Unicode символов по умолчанию)	35 Unicode символов	9 ASCII символов
Максимальная длина сообщения в чате	Регулируется на сервере (80 килобайт по умолчанию)	960 Unicode символов	512 ASCII символов
Возможности модерирования	Широкие	Нет	Средние
Возможность анонимного общения	Есть	Нет	Нет
Сложность настройки для конечного пользователя	Минимальна, работает сразу после установки	Минимальна, работает сразу после установки	Максимальна (попробуйте!)
Usability (удобство пользовательского интерфейса)	Очень удобно	Очень удобно	Программа для настоящих профессионалов борьбы с компьютером
Возможность скриптования функциональности	Есть, используется известный документированный язык Lua	Нет	Есть, используется собственный скриптовый язык
Шифрация трафика	Есть	Есть	Нет
Сжатие трафика	Есть, компрессия через zlib	Нет	Нет
Избыточность протокола	Минимальна	Средняя	Очень велика
Сложность перехвата сообщений	Высокая - применяется собственный бинарный протокол + сжатие + шифрование	Средняя - сообщения передаются напрямую, по нестандартному протоколу	Низкая - протокол полностью описан, существуют готовые средства
Поддержка IPv6	Нет	Есть	Нет
Централизованная настройка клиентов	Есть	Есть	Нет
Ориентация на ис-	Есть	Нет	Есть

	Colibri chat	Vypress Chat	mIRC (RFC 1459)
пользование в интернете			
Presence control (контроль реальной активности пользователя)	Есть	Есть	Нет
Flood control (контроль вредительской активности пользователя)	Нет	Есть	Есть
Поддержка макросов	Нет	Есть	Есть
Работа в многосегментных сетях	Есть (по TCP)	Есть (по IP Multicast)	Есть (по TCP)

С чего начать.

После инсталляции программы можно запустить клиентское приложение чата, вписать на странице сервера DNS-имя или IP-адрес сервера, ниже собственный ник, нажать ввод для соединения с сервером. Если на сервере установлен пароль на соединение, отличный от пароля по умолчанию, то перед попыткой соединения с сервером, необходимо справа на странице вписать этот пароль. Также нужно указать порт, если он отличается от порта по умолчанию.
После соединения с сервером будет отображён список доступных каналов на сервере. Вы можете заходить на каналы двойным щелчком мышью в списке, либо можете написать имя канала слева внизу списка, при этом можно зайти как на уже существующий канал, также создать, таким образом, новый. Если введённое имя – ник пользователя, то с ним будет открыт приват.
На каналах справа имеется список пользователей, общающихся на канале. Двойным щелчком в списке можно открыть приватный разговор с выбранным пользователем, в контекстном меню этого списка можно выбрать другие нужные действия, например для отсылки сообщения во всплывающем окне, передачи содержимого буфера обмена Windows, посылки звукового сигнала.
Если есть необходимые права доступа, то можно модифицировать параметры работы на канале.
При закрытии клиентской программы сохраняются все параметры её работы, таким образом, при следующем запуске программы восстанавливается предыдущее состояние работы, и вам не придётся что-либо настраивать заново, сразу после повторного запуска программы можно приступать к общению в уже настроенных условиях.

Имена.

Все имена контактов в Colibri chat – уникальны. Это значит, что ник, используемый пользователем – может быть только один на сервере, другой пользователь не может присвоить себе точно такой же ник. Кроме того, ники пользователей и имена каналов – не могут совпадать, если существует какой-либо ник, то снова задать пользователя или канал с таким же ником – невозможно.
Существует несколько предопределённых имён, которые никогда не могут быть присвоены пользователям, либо каналам:
Server – идентифицирует связь клиента с сервером, также имя первой страницы.
Channels – идентифицирует список каналов, вторая страница.

Noname – идентифицирует пользователя без ника, который не получил по запросу уникальный ник с сервера. Также пользователь может идентифицировать самого себя на сервере вместо собственного ника, эквивалентно понятию «я».
Anonymous – анонимный пользователь.
God – идентифицирует режим «бога».
Devil – идентифицирует режим «дьявола».

Права доступа на каналах.

Для решения задач модерирования общения на каналах, в Colibri chat создано 6 уровней прав доступа пользователей. Ниже описаны возможности каждого уровня в порядке возрастания, каждый последующий уровень включает все возможности предыдущих уровней:
Outsider – нет никаких возможностей.
Reader – чтение сообщений других пользователей на канале.
Writer – отправка сообщений на канал.
Member – изменение топика канала.
Moderator – удаление пользователей с канала, изменение прав доступа пользователей предыдущих уровней.
Administrator – изменение настроек канала, режимов работы канала.
Founder – изменение прав доступа любых пользователей на канале. Переименование канала.

Суперпользователи.

Colibri chat позволяет назначить два вида привилегий пользователям: god-cheat (режим «бога») и devil-cheat (режим «дьявола»). Для каждого из режимов имеется соответствующая пиктограмма, отображаемая слева в списке пользователей канала. Оба режима могут быть получены одновременно.
В режиме «бога» пользователь имеет следующие возможности:

- Может зайти на любой канал, несмотря ни на какие ограничения, то есть: может зайти на канал без приглашения, если канал только по приглашению; может зайти на канал, если его лимит исчерпан; также может зайти на канал под паролем без учёта пароля.
- Может переименовать (изменить ник) любого пользователя.
- Может переименовать любой канал.
- Может изменить права доступа любого пользователя любого канала, с любого уровня на любой другой.
- Обладает всеми правами основателя канала, даже если установлен уровень доступа только на чтение.
- Может изменить любые настройки любого канала.
- Может открывать приваты, посылать сообщения и выполнять другие действия относительно всех пользователей, независимо от их режима, даже если у пользователей установлен режим, ограничивающий эти действия.
- Пользователь в режиме «бога» всегда может видеть всех других пользователей на анонимных каналах, а также видит авторство сообщений.

В режиме «дьявола» пользователь всегда может удалить с любого канала любого другого пользователя, независимо от его прав доступа. При этом самого пользователя в режиме «дьявола» другие обычные пользователи удалить не могут.

Эти привилегии могут быть выданы пользователям с сервера, для этого нужно открыть список соединений сервера и в контекстном меню выбрать нужный пункт для выделенных пользователей.
Режим «бога» и «дьявола» также могут быть получены самими пользователями по паролям, установленным на сервере. Для получения режима «бога» необходимо на странице списка каналов написать в левом нижнем окне «**God**», правее пароль выдачи режима, по умолчанию «**godpassword**», нажать ввод. Для получения режима «дьявола» нужно написать «**Devil**» в левом нижнем окне и пароль на выдачу режима «дьявола» правее, по умолчанию «**devilpassword**». Повторное вышеописанное действие – снимет соответствующий режим. Пользователь не может выдать эти привилегии другим пользователям.
Установить или изменить пароли сервера на режим «бога» и «дьявола» можно через диалог «Passwords», который вызывается из контекстного меню в системном трее.

Скриптование.

Этот раздел предназначен для пользователей, желающих перепрограммировать какие-либо возможности функционирования чата.
Прежде чем приступить к программированию функциональности чата, рекомендую сначала ознакомиться с документацией по скриптовому языку Lua 5.1 на сайте разработчиков http://www.lua.org/, либо на русскоязычном сайте http://www.lua.ru/.

***Примечание № 1**. Ошибки в скрипте могут привести к сбоям в функционировании программы, либо к падению программы при запуске. Разработчик программы не отвечает за какие-либо ошибки программы, произошедшие из-за ошибок, внесённых в скрипты к программе другими лицами.*
***Примечание № 2**. Для редактирования скриптов Lua рекомендую использовать программу Notepad++.*

Код на все основные события клиентской части чата сведён в файле **events.client.lua**. Файл состоит из пяти секций:

1. ***Startup code and initialization***. Здесь определяются основные константы, которые будут в дальнейшем использоваться в коде скрипта. Также производится первичная инициализация.
2. ***Network events***. Дополнительные действия, которые должны осуществляться при событиях на сокетах. Все функции-отклики не модифицируют никаких входных параметров, и ничего не возвращают.
3. ***Transactions***. Функции-отклики на сетевые транзакции между клиентом и сервером, все не возвращают никаких значений.
4. ***Windows messages***. Функции-отклики на стандартные оконные сообщения, все не возвращают никаких значений.
5. ***Commands response***. Функции-отклики на команды оконной системы Windows, то есть команды, которые приходят с сообщением WM_COMMAND. Функции-отклики не возвращают никаких значений.

Хост-программа предоставляет возможность вызывать некоторые функции из скрипта. Эти функции делятся на 2 вида: статические функции, и функции-методы класса.
Статические функции в скрипте могут принадлежать разным пространствам имён, в программе их определено два: **profile** (для работы с веткой реестра

Windows, содержащей установки программы), и **str** (для обработки строк). Например, чтобы определить, разрешено ли автоматическое открытие каналов при первом после запуска клиента программы соединении с сервером, необходимо сделать следующий вызов в скрипте:

profile.getInt(RF_AUTOOPEN, "UseAutoopen", 0)

Функции-методы класса вызываются применительно к какому-либо объекту, причём первым параметром в неявном виде передаётся указатель на этот объект. В Lua 5.1 нет реализации всех возможностей ООП, функции-методы привязаны только лишь к единичному объекту через метатаблицу. В программе заскриптован один класс – **Client**, и экспортируется один объект этого класса – **chat**, который сопоставлен с объектом клиента чата. Таким образом, все вызовы функций-методов должны иметь вид:

chat:func(params)

что эквивалентно следующему вызову:

Client.func(chat, params)

Доступны для вызова из скрипта следующие функции-методы:

Имя метода	Описание
regFuncs	Регистрация статических функций в скрипте. Вызов этой функции должен быть только один раз в начале скрипта, до вызова каких-либо статических функций. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
getGlobal	Регистрирует глобальные переменные в скрипте, инициализирует значениями из хост-программы. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
PlaySound	Проигрывает медиа-файл через MCI от начала до конца, интерфейс хост-программы при этом не блокируется. Ориентируется на аудиофайлы: WAV, MP3, WMA и т. д., видео проигрывается без изображения. ***Параметры***. Принимает 1 строковый параметр, который должен содержать путь и имя файла для воспроизведения, параметр – обязательный. ***Возвращаемые значения***. Нет.
ShowTopic	Отображает текст в заголовке программы. ***Параметры***. Принимает 1 строковый параметр, который должен содержать текст для отображения в заголовке. Если строковый параметр отсутствует, то выводит в заголовок программы пустую строку. ***Возвращаемые значения***. Нет.
saveAutoopen	Сохраняет в ветке реестра Windows для автосохранения каналов, текущее состояние открытых каналов. Сохранение – безусловное. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
openAutoopen	Открывает ранее сохранённое состояние открытых каналов. Открытые прежде каналы – не закрываются. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
Log	Выводит текст с разметкой в лог на странице сервера. ***Параметры***. Принимает 1 строковый параметр, содержащий текст с разметкой. Параметр – обязательный. ***Возвращаемые значения***. Нет.
HideBaloon	Скрывает всплывающую подсказку в виде надувного пузыря. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
Connect	Осуществляет соединение с сервером. ***Параметры***. Принимает 1 булевский параметр. Если значение – **true**, то параметры соединения считываются со страницы сервера, а именно: DNS-адрес или IP сервера, номер порта, пароль на соединение с сервером, ник. Если значение – **false**, то используются прежние параметры соединения с сервером. Если булевский параметр функции – не указан, то

Имя метода	Описание
	по умолчанию принимается **false**. ***Возвращаемые значения***. Нет.
Disconnect	Разрывает соединение с сервером. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
getConnectCount	Получает номер попытки соединиться с сервером. ***Параметры***. Нет. ***Возвращаемые значения***. Возвращает 1 целочисленное значение – номер попытки соединиться с сервером.
setConnectCount	Устанавливает номер попытки соединиться с сервером. ***Параметры***. Принимает 1 целочисленный параметр, который будет идентифицировать номер попытки соединиться с сервером. Если параметр отсутствует, то по умолчанию принимается 0. ***Возвращаемые значения***. Нет.
getSocket	Получает дескриптор соединения с сервером. ***Параметры***. Нет. ***Возвращаемые значения***. Возвращает 1 целочисленное значение – SOCKET установленного соединения между клиентом и сервером. Если соединение не установлено, то возвращает 0. Таким образом, после вызова **Connect** значение будет ненулевым, после получения уведомления **EvLinkClose** значение будет нулевым.
checkConnectionButton	Изменяет состояние кнопки-индикатора соединения с сервером. ***Параметры***. Принимает 1 булевский параметр, который интерпретируется, как состояние кнопки. Если значение **true**, то кнопка – вдавлена, если **false**, то выдавлена. ***Возвращаемые значения***. Нет.
WaitConnectStart	Устанавливает длительность паузы перед очередной попыткой соединения с сервером, вызов функции необходим только при неудачной попытке соединиться с сервером. ***Параметры***. Принимает 1 целочисленный параметр, который исчисляет длительность паузы в миллисекундах. Если параметр не указан, то по умолчанию принимается 30000 миллисекунд, то есть длительность паузы будет 30 секунд. ***Возвращаемые значения***. Нет.
WaitConnectStop	Прерывает паузу перед следующей попыткой соединения с сервером, установленной через вызов WaitConnectStart. Попытки соединения при этом не осуществляется. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
MinimizeWindow	Сворачивает главное окно программы. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
MaximizeWindow	Разворачивает главное окно программы на весь экран. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
RestoreWindow	Восстанавливает исходное положение главного окна программы. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
FlashWindow	Включает, либо выключает выделяющее мерцание окна программы. ***Параметры***. Принимает 1 булевский параметр. Если значение – **true**, то мерцание включается, если **false** – то выключается. Если параметр не указан, либо не булевский, то по принимается значение **true**. ***Возвращаемые значения***. Нет.
DestroyWindow	Закрывает главное окно программы. Закрытие происходит без каких-либо условий. ***Параметры***. Нет. ***Возвращаемые значения***. Нет.
PageEnable	Активирует указанную страницу. ***Параметры***. Принимает 1 строковый параметр – имя вкладки. Например, для вкладки сервера имя будет **“Server”**, для списка каналов – **“List”**, для

Имя метода	Описание
	открытого канала - имя, отображаемое в списке каналов слева, для приватного разговора - ник пользователя. Если параметр не указан, то по умолчанию принимается страница сервера. ***Возвращаемые значения***. Нет.
PageDisable	Дезактивирует указанную страницу. ***Параметры***. Принимает 1 строковый параметр - имя вкладки. Например, для вкладки сервера имя будет **"Server"**, для списка каналов - **"List"**, для открытого канала - имя, отображаемое в списке каналов слева, для приватного разговора - ник пользователя. Если параметр не указан, то по умолчанию принимается страница сервера. ***Возвращаемые значения***. Нет.
PageAppendScript	Добавляет текст с разметкой в конец лога на указанной странице. ***Параметры***. Принимает 2 строковых параметра. Первый строковый параметр - имя вкладки. Например, для вкладки сервера имя будет **"Server"**, для списка каналов - **"List"**, для открытого канала - имя, отображаемое в списке каналов слева, для приватного разговора - ник пользователя. Если первый параметр не строковый, принимается страница сервера. Второй строковый параметр - текст с разметкой, добавляемый в конец лога. ***Возвращаемые значения***. Нет.
Say	Посылает на сервер транзакцию, содержащую сообщение на канале, либо в приватном разговоре. ***Параметры***. Принимает 2 строковых параметра. Первый строковый параметр - имя вкладки, то есть имя канала, либо ник пользователя, с которым открыт приватный разговор. Если первый параметр не строковый, принимается страница сервера. Второй строковый параметр - текст, содержащий сообщение. Текст сообщения интерпретируется всегда только как текст в ANSI-кодировке. ***Возвращаемые значения***. Нет.
Message	Посылает на сервер транзакцию, содержащую сообщение пользователю во всплывающем окне. ***Параметры***. Принимает 2 строковых параметра. Первый строковый параметр - ник пользователя. Если первый параметр не строковый, принимается страница сервера. Второй строковый параметр - текст, содержащий сообщение. Текст сообщения интерпретируется всегда только как текст в ANSI-кодировке. ***Возвращаемые значения***. Нет.
Alert	Посылает на сервер транзакцию, содержащую сообщение пользователю во всплывающем окне, которое отобразится поверх остальных окон Windows. ***Параметры***. Принимает 2 строковых параметра. Первый строковый параметр - ник пользователя. Если первый параметр не строковый, принимается страница сервера. Второй строковый параметр - текст, содержащий сообщение. Текст сообщения интерпретируется всегда только как текст в ANSI-кодировке. ***Возвращаемые значения***. Нет.
Beep	Посылает на сервер транзакцию для посылки звукового сигнала пользователю. ***Параметры***. Принимает 1 строковый параметр - ник пользователя. Если параметр не строковый, принимается страница сервера. ***Возвращаемые значения***. Нет.

Примечание № 3. *Хост-программа работает со строками в юникоде, в то время как скрипт содержит строки в мультибайтной ANSI-кодировке. Поэтому при интерпретации строковых параметров функций осуществляется перевод строк из ANSI в Unicode.*