**Слайд 1**

Всем привет. Тема моей дипломной работы - «Разработка защищенного видео чата на основе стандарта WebRTC»

**Слайд 2**

Целью данной работы является создание видео чата на основе webRTC стандарта для установления соединения и обмена информацией между конечными пользователями.

Цель данной работы вытекает из того факта, что на сегодняшний день большинство приложений, выполняющих функцию видеочата подвержены угрозам со стороны злоумышленника, имеют закрытый исходный код, либо имеют ограничение в плане кроссплатформенности

Для реализации этих целей были поставлены следующие задачи

* Проанализировать разновидности технологий при создании видео чата.
* Проанализировать теоретические сведения по стандарту webRTC;
* Разработать программный продукт - видео чат - для организации передачи потоковых данных между браузерами на основе стандарта webRTC;
* Сравнить разработанный продукт с другими платформами вещания в реальном времени.

**Слайд 3**

Для того, чтобы понять область исследования следует разобраться с тем, что из себя представляет определение видеочат.

Видео чат это онлайн-сервис, позволяющий пользователям Интернета общаться, в дополнение к текстовой переписке, в аудио и видео (посредством веб-камеры) режиме.

На данный момент одним из популярных решений создания видео чатов является adobe flash player

Основным недостатком flash-приложений является чрезмерная нагрузка на центральный процессор, Это связано с неэффективностью виртуальной машины Flash Player.

Так же к недостаткам можно причислить

* Задержка трафика из-за использования протокола RTMP;
* Обязательно наличие промежуточного сервера;
* Закрытые средства разработки и проигрывания.
* Раскрытие персональных данных
* Получение контроля над управлением веб-камерой и микрофоном.
* Внедрение рекламы в операционную систему пользователя

181 уязвимость Flash, из которых 152 оценены как критичные.

А теперь немного подробнее рассмотрим и остановимся на webrtc стандарте для передачи потоковых данных

WebRTC - стандарт, позволяющий построить соединение в режиме узел-узел, захватывать и вещать аудио и видео медиа-потоки, а также передавать между браузерами произвольные данные, без обязательного использования посредников.

Webrtc говорит о необходимости использования SRTP (Secure Real-time Transport Protocol) протокол передачи данных, который предназначен для шифрования, установления подлинности сообщения, целостности,

Этот стандарт был выбран по нескольким причинам:

* Не требуется установка ПО
* Высокое качество связи
* Высокий уровень безопасности - все соединения защищены и зашифрованы на протокольном уровне с помощью SSL/TLS;
* Возможность реализации любого интерфейса управления
* кросс-платформенность

Но, у webRTC есть так же ряд недостатков

Для организации групповых аудио и видеоконференций требуется сервер ВКС, который бы микшировал видео и звук от участников, т.к. браузер не умеет синхронизировать несколько входящих потоков между собой.

Все WebRTC решения несовместимы между собой, т.к. стандарт описывает лишь способы передачи видео и звука, оставляя реализацию способов адресации абонентов

**Слайд 4**

На данном слайде изображен схематический процесс установления р2р соединения с помощью WebRTC .

Первый клиент желает совершить звонок второму клиенту. WebRTC дает всю необходимую информацию, чтобы себя обозначить. Но остается открытым вопрос, как одному браузеру найти другой, как эту метаинформацию переслать, как проинициализировать вызов.

Первый клиент формирует метаинформацию, и с помощью веб-сокетов или HTTP пересылает ее на сигнальный сервер.

Второй клиент берет ее, использует, устанавливает себе, формирует ответ, и с помощью сигнального механизма пересылает ее на сигнальный сервер, тот в свою очередь ретранслирует ее первому клиенту. Таким образом оба клиента в данный момент обладают всей необходимой датой и метаинформацией.

STUN — это клиент-серверный протокол. Клиент отправляет запрос серверу STUN. Затем сервер STUN отправляет клиенту обратно информацию о том, каков внешний адрес маршрутизатора NAT, и какой порт открыт на NAT для приема входящих запросов обратно во внутреннюю сеть.

Таким образом, в процессе установки P2P соединения, каждый из клиентов должен сделать по запросу к этому STUN-серверу, чтобы узнать свой внешний IP-адрес

SDP Session Description Protocol - метаданные с характеристиками

STUN (Session Traversal Utilities for NAT, Утилиты прохождения сессий для NAT) - это сетевой протокол, который позволяет клиенту, находящемуся за сервером трансляции адресов (или за несколькими такими серверами), определить свой внешний IP-адрес

**Слайд 5**

Познакомившись с теорией и разобравшись в проблемах уже существующих видео чатов, можно переходить к практической реализации видео чата

1. На данном слайде представлен интерфейс приложения, который состоит из поля для ввода уникального идентификатора комнаты и двух кнопок для того, чтобы присоединиться, либо покинуть комнату
2. Для открытия соединения необходимо лишь ввести номер комнаты и нажать кнопку «присоединиться» После того, как первый участник присоединился к комнате - система находится в состоянии ожидания.
3. Со второго устройства так же требуется перейти по ссылки и указать прежний идентификационный номер комнаты и присоединиться к комнате.

**Слайд 6**

На следующем слайде я продемонстрирую работу видео чата, соединив между собой два устройства. Для чистоты эксперимента, устройства были подключены к интернет-соединению от разных провайдеров

**Слайд 7**

Центральным объектом в API служит RTCPeerConnection. В качестве конфигурации указывается адрес STUN сервера, в том числе и мета информация

Клиент делает запрос к STUN-серверу, узнает свой внешний IP-адрес, и с помощью onicecandidate узнает неохзоимую информацию.

Когда устанавливается соединение и требуется проинициализировать вызов, то используется метод createOffer(), чтобы сформировать начальную SDP, offer SDP, ту самую мета-информацию, которую нужно переслать другому участнику системы.

SDP Session Description Protocol - метаданные с характеристиками

STUN (Session Traversal Utilities for NAT, Утилиты прохождения сессий для NAT) - это сетевой протокол, который позволяет клиенту, находящемуся за сервером трансляции адресов (или за несколькими такими серверами), определить свой внешний IP-адрес

**Слайд 8**

Когда клиент получил доступ к медиапотоку, он может его передать в уже имеющееся P2P-соединение с помощью метода addStream, а другой клиент узнает об этом, у него стриггерится ивент onaddstream. Он получит наш поток и сможет его отобразить.

Так же существует метод getUserMedia, который принимает на вход набор констрейнтов. Это специальный объект, где клиент указывает, к каким именно устройствам он хочет получить доступ, к какой именно камере, к какому микрофону.

**Слайд 9**

Рассмотрим преимущества разработанного видеочата, написанного в рамках данной дипломной работы.

Так же данный видео чат является кроссплатформенным и поддерживается во всех популярных браузерах

Одним из значительных плюсов является то, что для пользования продуктом необходимо лишь установить браузер и иметь сетевое соединение.

**Слайд 10**

На сегодняшний день одним из самых популярных программных продуктов, основа которых написана в соответствии со стандартом webRTC являются медиа сервис «Skype» и «Google Hangouts».

Можно сделать выводы, что «Skype» и «Google Hangouts» делают упор на качество, в то время как задержка достигает больших значений.

Еще одним основным преимуществом перед той группой лиц, которые по каким-либо причинам не хотят идентифицировать свою личность, является то, что участникам не обязательно регистрироваться

**Слайд 11**

Так же при разработке данного программного продукта было отмечено, что с увеличением количества участников растет потребление системных ресурсов. Для теста в каждой из программ были созданы конференции размером от 2 до 5 участников.

Можно сделать выводы, что написанное в рамках дипломного проекта приложение превосходит аналоги по показателю потребления оперативной памяти системы, что играет немаловажную роль при пользовании продуктом.

**Слайд 12**

В данной работе решены прежде поставленные задачи:

Разработанный продукт может применяться для проведения защищенного видео чата. Для организации трансляции требуется только создать комнату и поделиться ссылкой с участниками.

Нет необходимости устанавливать дополнительного ПО, программный продукт является кроссплатформенным, данные передаются с помощью протоколов беопасности (DTLS и SRTP), поддерживается большинством браузеров.