by

Arun Shyam (MS CS)

Ankush Gulati (MS CS)

Ravneet Singh Dhaliwal (MS CS)

Stony Brook University

Abstract

DirectConnect is a peer-to-peer files haring protocol. Each DC user can set up his or her own hub, and specify permissions for who can access that hub. Typically, individuals grant access to their friends so that they can share files with only people they trust.

Facebook is a social network service and website. Users may create a personal profile, add other users as friends and exchange messages. At present Facebook is the most used social networking site with more than 500 million active users.

In this project we aim to integrate this capability of file sharing with FaceBook so that the users can use Direct Control Protocol to share files among their friends on FaceBook. The task is to maintain the access controls during file sharing so that no one outside the user's friend list can access his/her files. This is in lines with the privacy settings feature of FaceBook.

Background

1).Direct Connect

1.1) Overview

Direct connect is a peer-to-peer file sharing protocol. Direct connect clients connect to a central hub and can download files directly from one another. Hubs feature a list of clients or users connected to them. Users can search for files and download them from other clients, as well as chat with other users.

1.2) Protocol[3]

The Direct connect protocol is a text-based computer protocol, in which commands and their information are sent in clear text, without encryption. As clients connect to a central source of distribution (the hub) of information, the hub is required to have a substantial amount of upload bandwidth available. The client-server as well in client-client, where one acts as server aspect of the protocol stipulates that the server speak first when a connection has been made. There is no global identification scheme; users are identified with their nickname on a hub-to-hub basis.

Hubs may send out user commands to clients. These commands are only raw protocol commands, and are used mostly for making a particular task simpler. For example, the hub cannot send a user command that will trigger the default browser to visit a website. Downloads are transported using TCP. The connection to the hub is with TCP. Protocol delimiters are '$', '|' and ' ' space. There is no escape sequence, so these characters can't be sent without possibly compromising the integrity of the message. The TCP data is a series of commands, each in the form of <command>. The | indicates the end of a command and is not followed by a newline.

2) FaceBook[6]

2.1) Overview

Facebook is a social network service and website. Users may create a personal profile, add other users as friends and exchange messages. At present Facebook is the most used social networking site with more than 500 million active users. Additionally, users may join common interest user groups, organized by workplace, school, or college, or other characteristics.

2.2) Facebook API

The Facebook API provides the users with the functionality to login to their accounts after entering their credentials. A customized application can be developed in which we can use this FaceBook API. We can then use this application to extract relevant information about the client interested in file sharing.

Introduction

DirectConnect is a peer-to-peer files haring protocol with many open-source implementations. Each DC user can set up his/her own hub, and specify permissions for who can access that hub. Typically, individuals grant access to their friends so that they can share files with only people they trust. Direct Connect is basically a text-based computer protocol, meaning that information and data is sent as a plain text rather than any encryption.

The aim is to integrate this capability of file sharing with FaceBook, which at present is the largest social networking service in the world. Users can then use Direct Connect protocol to share files among their friends on FaceBook. The main task is to maintain the access controls during file sharing so that no one outside the user's friend list can access these files, i.e. we want to maintain the privacy settings specified by the user on FaceBook.

Direct Connect being a text based protocol, it's very important to introduce some security features so that data security and integrity is maintained during transfer of files from one user to the other. For this purpose a lot of code modification needs to be done on the DC client and hub. A PHP server is developed and hosted on the same machine on which the DC hub is running. This web server acts as a third party authenticator of the clients wishing to join the hub. The concept of Magic Numbers and FaceBook ID is introduced to remove identity theft attacks and making data transfer secure.

Components

1) Direct Connect Hub[1]

Direct Connect hubs are central servers to which clients connect, thus the networks are not as de-centralized as true peer-to-peer networks. Hubs provide information about the clients, as well as file-searching and chatting capabilities. File transfers are done directly between clients, in true peer-to-peer fashion.

We are using opendchub-0.8.2 as the DC hub software to which all the clients (Facebook users) connect for the purpose of file sharing. Open DC hub is a Unix/Linux version of the hub software for the Direct Connect network. The hub is run as a daemon, i.e. it runs in the background. It's administrated through a TCP connection, for example with telnet, which makes it possible to administer the hub remotely.

2) Direct Connect Client[2]

Direct connect clients are peer-to-peer file-sharing applications that can be used to connect to the Direct Connect network. While not mandated by the protocol, most clients send a "tag". This is part of the client's description and display information ranging from client name and version to number of total available slots to the client.

In this project we are using linuxdcpp open source Direct Connect client. Linuxdcpp is a Linux port of DC++. Clients connect to a central hub where they can view a list of clients or users connected to them. A user can search for files and download them from other users. Chatting capability is also available.

3) FaceBook Application[6]

We need a FaceBook application in order for the user to login into FaceBook. For the same purpose we developed an application named FaceDirect which does the FaceBook authentication of the user for us. Our PHP server directs the user to the login page for FaceDirect from where user credentials are verified and extracted.

4) PHP Server [5]

We developed a PHP server which runs on the same machine as the DC hub. This server directs the client interested in joining the hub to the FaceDirect login page from where it extracts useful information about the client. It then stores this information in a file from which the DC hub does the verification of the client.

Design Architecture

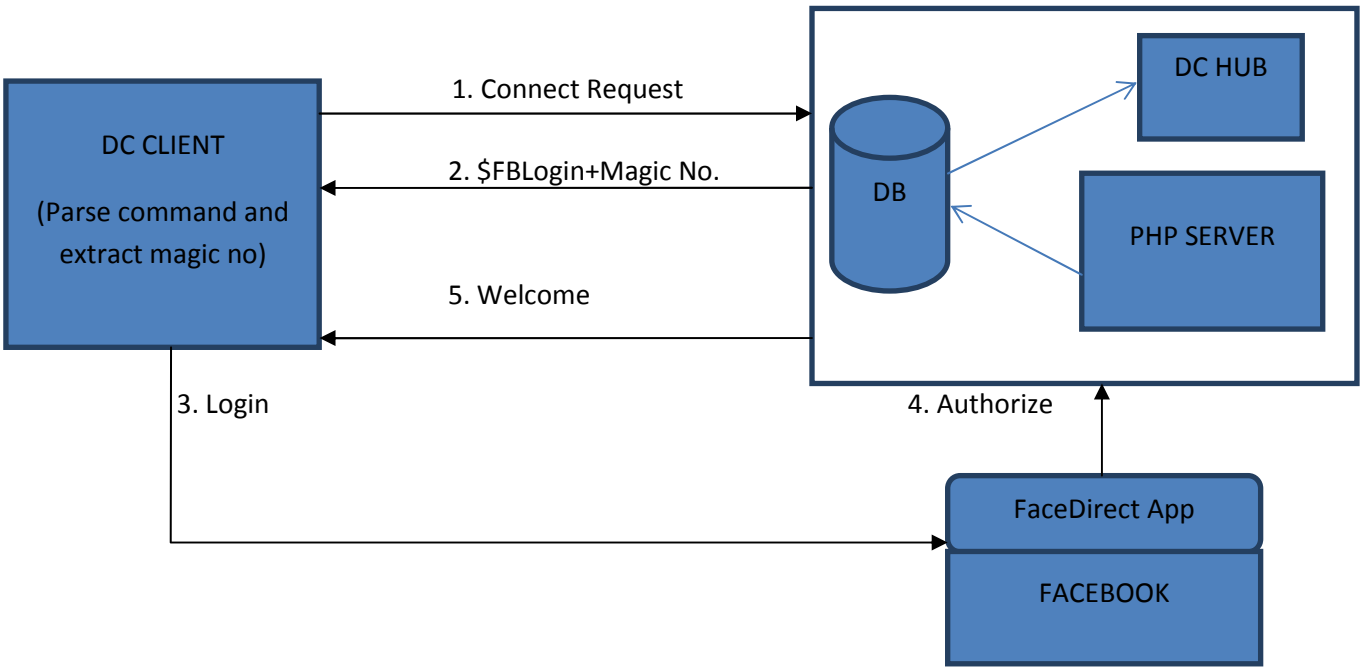

Fig 1.1

The machine on which the client application runs (client M/C) sends a connect request to the machine on which DC hub is running. The server (server M/C) replies with a FaceBook login command ($FBLogin). The server also sends a random number (Magic Number) to the client and stores one copy with itself. The client parses this command, appends the Magic Number to the login URL and proceeds. The FaceDirect application page opens and asks the user to login with his credentials. If the login succeeds, the PHP server writes the user FaceBook ID, Magic Number, friend list and other important information into a file. The DC hub then reads the Magic Number for that user off the file and compares it with the stored copy of Magic Number to verify if the same client was authorized for whom this number was generated. If the extracted Magic Number matches with the one stored by the hub, then the client gets authenticated and connection is established between the client machine and the server. Normal file sharing between the user and his friends can now take place.

Approach

The Implementation approach is as follows-:

1) Whenever a user tries to connect to the DC hub, the hub before allowing him to join sends our FaceBook login command along with a random number which we call the Magic Number. The hub also keeps the copy of the sent Magic Number with itself for future verification.

2) The DC client then parses the command received which tells it to login to FaceBook. The client code then appends this Magic Number to the URL of the PHP server (hardcoded in the DC client code) from where it will be directed to the login page.

3) The PHP server on receiving the login request redirects the user to the FaceBook application (FaceDirect) that we have developed. The Facebook application acts as an authenticator of the FaceBook user. The API display asks for the user permission to access his personal information. If the user allows the application to run, the normal FaceBook login page appears where the user enters his login name and password.

4) After the user has been authenticated, the PHP server extracts all the necessary information of that user like magic number, name, friend list, Facebook ID and stores it in file. For every authenticated user we maintain a separate file. For every authenticated user, we maintain a separate file, so in this case we achieve concurrency; that is multiple users can login simultaneously.

5) The DC hub then reads the magic number of the authenticated user from the file and compares it with the one it had stored earlier. If both the numbers match then the hub verifies that the FaceBook authenticated client is the same as the user who had earlier generated the request for joining the hub.

6) The interested user then gets connected to the DC hub and hence can participate in file sharing with other users in the hub in a normal fashion.

Implementation

Flow of DC Client and Hub code[4][1][2]:

1). Client sends a connect request to hub.

2). Client also sends validate nick list to hub.

3). Server replies back by either accepting or denying the client request.

4). Client calls the Get_nick_list API, corresponding to which the server calls the send_nick_list () function that sends back to the client the list of all connected users.

5). Client then calls the Myinfo API, corresponding to which the server calls the send_to_human () function that distributes the information of the client to all connected users.

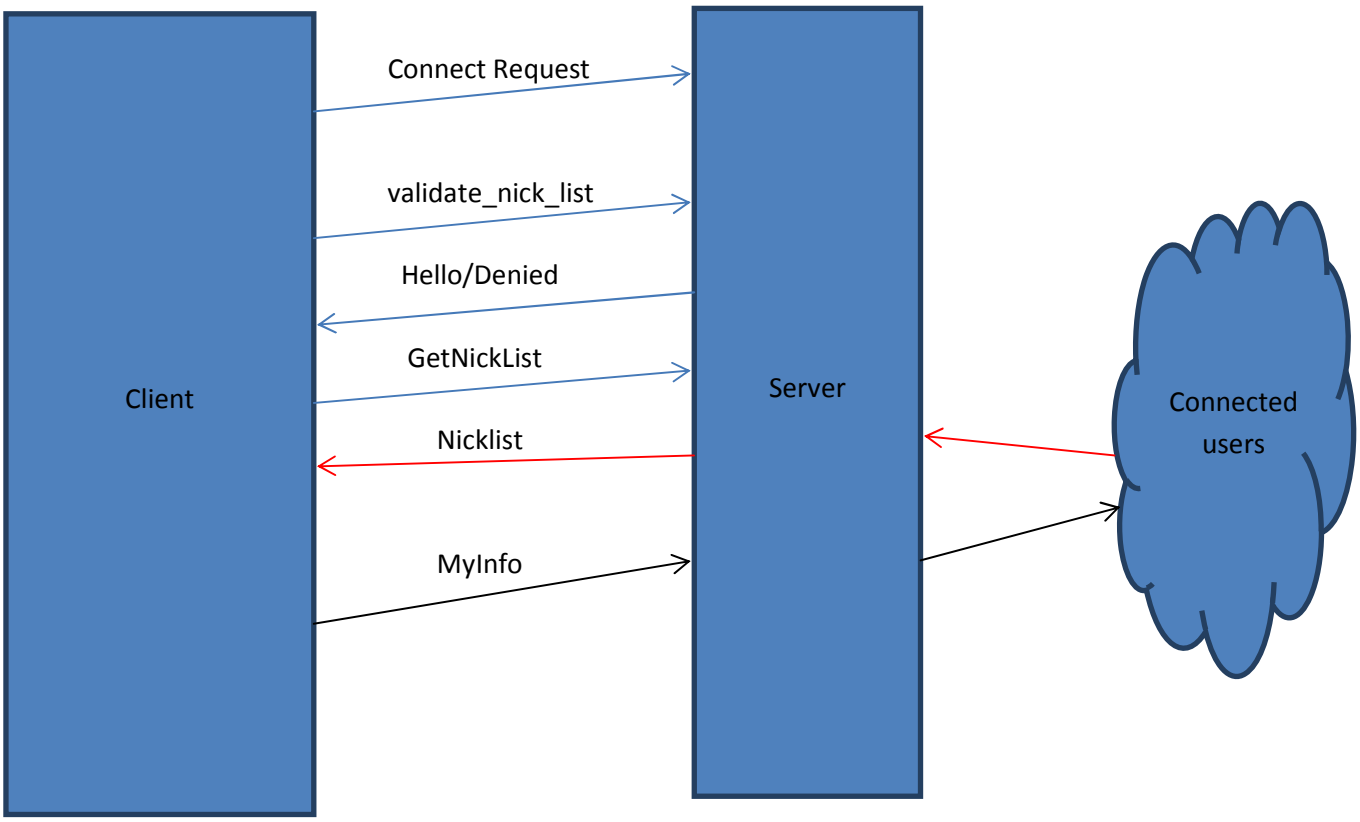

Fig 1.2

Modifications:

Hub and Client code:

1). We have maintained a 'large' file containing the information of all the users who are presently connected to the hub along with their information like FaceBook ID, Name, friend list. We also maintain separate files for each new user who connects to the server.

2). We have added an additional field (Facebook ID) to the user structure on the hub side. Now every time the connected user asks for the list of connected users, only those users are reported back to him that are in his friend list. This is achieved by comparing the FaceBook ID's of his friends with the ID's in the user list.

3). When the client calls the Myinfo API, the server calls the associated send_to_human () function. The change we have introduced in this function is that, now, instead of client information being distributed to all the connected users, is only distributed to client's friends who are at present logged in.

4). When the client calls the Get_nick_list API, the server calls the associated send_nick_list () function. We have introduced changes in this particular function. Now, the send_nick_list () instead of returning the information of all the users who are logged in, sends only the information about the users who in the client's friend list.

5). When have changed the message send function that the server calls in response to the client connect request. With this message we send our own custom command $FB_LOGIN along with our Magic Number. The $FB_LOGIN command tells the client to login to FaceBook and verify his credentials, while Magic Number is the random number generated that is used to authenticate the user on the hub end.

6). The client side on receiving the $FB_LOGIN command parses it, extracts the Magic Number and appends it to the URL of our PHP server which redirects the client to the FaceBook login page and the subsequent steps follow as discussed above in approach.

PHP Code[5]:

We use FaceBook authentication API to get the users log in. We have hosted the FaceDirect webpage on the PHP server running on our machine (the same machine on which DC Hub runs).

1). FaceDirect webpage has a login button login button takes user to FaceBook login page.

2). FaceBook validates user and the session gets validated.

3). `$cookie = get_facebook_cookie (FACEBOOK_APP_ID, FACEBOOK_SECRET);`
After login generates cookie for future sessions.

4). Now we extract the information of the user's friends by extracting the access tokens as

```
$friends = json_decode (file_get_contents (
        'Https: //graph.facebook.com/me/friends? access_token='
```

5).Similarly we extract user information i.e. his FaceBook ID as

```
$id = json_decode (file_get_contents (
            'Https: //graph.facebook.com/me? access_token='
```

6).The information thus obtained about the user and his friends is written into a file by PHP server which can be read by the DC hub for user verification thereafter.

Challenges

We faced several challenges during the implementation of this project-:

Firstly there are many versions of Direct Connect client and hub on the internet and their implementation varies version to version. It is hard to find a generic DC client whose implementation is in lines with the original DC client. Moreover, the Direct Connect protocol itself has no official documentation or specification which made understand DC's working a tedious process for us. A lot of time was used up in understanding the protocol which could have been utilized in implementation.

Being a text based protocol; it made Direct Connect a very insecure mechanism for file transfer across FaceBook users, where privacy is one of the key issues. Two new design components had to be added to transfer files in a more secure fashion. This increased the trusted computing base (TCB) of our project.

The communication between the PHP server and DC hub was a crucial issue. It was resolved by making the PHP server to write the information of the authenticated FaceBook user on a file which would then be read by the hub to verify the identity of the client requesting to join the hub.

Conclusion

We have successfully integrated one of the most commonly used peer-to-peer file sharing application- Direct Connect with the largest used social networking site- Facebook. We have integrated the file sharing capability of Direct Connect with FaceBook API. Now FaceBook users can share their files with their friends in a secure fashion maintaining their access control permissions.

We have also achieved complete mediation of code. Client cannot take any other path to enter the hub before going through the security mechanisms implemented as us. The TCB of our system is also small as it is dependent on the code modification made on the DC hub and the PHP server.

Future Work

At present we have a DC client running on the FaceBook user machine through which file sharing takes place. We would like to implement the DC client API into a FaceBook application so that the user need not install and learn the functionality of new software like DC. All file sharing can then take place through FaceBook itself to which the users are pretty familiar. It also serves as a means to reduce the TCB of this software system.

References:

1). http://opendchub.sourceforge.net/

2). http://packages.debian.org/unstable/net/linuxdcpp

3). http://www.teamfair.info/DC-Protocol.htm

4). http://en.wikipedia.org/wiki/DC%2B%2B

5). http://www.php.net/

6). http://en.wikipedia.org/wiki/Facebook