Modern spyware and the problems of "Discord newspeak"

The history of modern instant messaging

Remote messaging has taken many forms during the known history of humanity. Before electricity was invented, messages were transmitted using smoke signals and carrier pigeons. With electricity the telegraph was invented, and human voice was turned into an electric signal and transmitted through long wires - the latter is still in use today and is known as *"a telephone"*. The significance of a telegraph in communication has practically been lost after digital computers that route messages automatically became common.

First digital messaging applications used very simple protocols and often no other client program than just a telnet client was needed. The SMTP protocol, which is still actively used in transferring e-mails between servers, was developed in 1982. The IRC protocol, which was developed in Finland by Jarkko Oikarinen in 1988, became the first broadly used instant messaging protocol.

Definitions of technical terms used in this writing

Client: A program that the user installs to their computer to use the service.

Server: A program that is installed to the server computer, which relays the messages between different users.

Protocol: A set of rules that the client and server programs use to exchange information between each other.

Payload: The "useful" data that is transferred from a client to an another client via the protocol - in the case of communicator applications this is usually a message from a user to an another user.

End-to-end encryption: The payload is encrypted by the client that sends it, and it is decrypted by the client that receives it, so that the server that relays the message cannot see the decrypted contents of the payload.

Instant messenger: A synonym to the word "chat".

Bridge bot: A bot that relays messages between two or more conversation channels. Different conversation channels can use different protocols - for example, the bot can relay messages between an IRC channel and a Matrix channel.

Free: Free as in freedom. Has nothing to do with the price. May or may not be gratis.

Differences between closed and free instant messengers

Probably the most important difference between closed and free instant messengers is how their name is used. The name of a closed instant messenger program is usually associated to their client program, but when we are speaking about a free instant messenger, we are speaking about the protocol and not any particular computer program. In fact a free instant messenger IS the protocol - there may or may not exist a client program of the same name, and "THE client program" certainly doesn't exist. Good examples of this are two well known instant messengers IRC and Mumble: Both of them have many existing client programs and the first of them doesn't even have a client program that would be named after the protocol. A free instant messenger is a protocol that is implemented in the context of the server or client program.

The protocol of a closed instant messenger is usually completely undocumented, and because of that creating an alternative client program is very hard and laborious work. Using an alternative client program may also be completely forbidden in their terms of service, which is the case with Discord. A closed and undocumented protocol creates a situation where the instant messenger in question works only on those devices that the official client program has been made for. In contrast to this free instant messengers have an open and documented protocol, which makes it easy to write a client program. The most commonly used free instant messengers have client programs for practically every type of computing device.

An undocumented protocol also makes it difficult to create bridge bots, and often the terms of service completely forbids using bridge bots. Usually the main business idea of a closed instant messenger is to keep its users trapped inside its walled garden. Apart from some exceptions closed instant messengers don't usually have a publicly available server program and thus creating an own server is impossible, which makes them more vulnerable to denial-of-service attacks and censorship by totalitarian governments.

How marketing changes the meaning of words

Most proprietary closed-source computer programs are produced by profit-seeking companies. Often their marketing is not exactly honest - especially when the workings of the product in question is not completely understood by its average user. The same phenomena can also be seen with computer software, and one of the most used means of untruthful marketing is to change the meaning of words to better match the agenda of the company. The worst case scenario is that the new way of using the words becames established as the new normal, which is at an especially high risk of happening within the contexts of technical literature, where the target group of the marketing often initially learns about the new word from the purposely untruthful marketing material. The establishment of the newspeak definitions of the words into the normal usage of language makes technical things even harder to understand.

Often it seems that the misleading use of words has caused almost irrepairable damage, and this can even be seen happening to university students of information technology, the very people who should learn these things properly or else in the near future no-one is able maintain the digital systems that our society increasingly relies on. After one has initially learned a misleading

definition for a technical term, it becomes very hard to adjust the inner paradigms to understand the real technically correct meaning of the word.

Usually the purpose of using words misleadingly is to "flatten" the meanings of words that are considered positive things. For example, the word "secure" may be used for an instant messenger program that does not even have end-to-end encryption and all messages are saved to the server in plaintext form - in that case what they actually mean is that only the connection between the client and the server is encrypted. Discord calls the groups inside their service "servers" to create a misconception that everyone can create their own servers for Discord. Of course, in reality, it has nothing to do with actual servers - the word "server" means and has always meant, in the hardware level the computer that runs the server software, and in software level a program that listens to connection requests from clients, and neither of those can be created via clicking some links in some Electron app.

Conceptual problems when speaking about information security

During the recent years the security of closed source computer programs has often been in the headlines around the world. Spying features have been found, among the others, from the operating systems of Microsoft and Apple. The fact that Facebook makes money by selling information about its users has raised conserns. Many countries have been boycotting Huawei when building 5G networks because of the possibility of China using their network devices for spying purposes.

Often the discussion about security gets derailed or becomes completely impossible, because the definitions of words are unclear. Already long before the era of the modern "Discord newspeak" security companies have had the habit of marketing their products like security was something that could be bought from a store, which has already made it difficult to educate people about the subject.

Understanding the security of instant messengers is not possible, if the concept of a server is unclear. Essential things are the *encryption* and the question of whether the *server* is trusted or not, and if not, does it see the messaging between the *clients*. Because of the "Discord newspeak" a typical conversation about security goes like this:

1: To be actually secure, the messenger program has to encrypt the messages between the users.

2: Discord uses an encrypted connection. Therefore Discord is secure.

1: Discord is not secure, because its encryption is not done between the users. Its encryption only exists between the client and the server, and the messages are saved to the server in a plain-text format. They probably also sell all your messages to advertisers.

2: I created my own Discord *server* that I trust, because it is my own server and I can always trust my own server! In addition to that, creating a *server* to Discord is very easy, because it only requires clicking couple of links from the client program! Only with outdated legacy messenger apps you need to install some server program and leave the computer powered on 24/7 just to have a server!

What went wrong? The conversationalist no. 2 knows that encryption between the client and the server is sufficient for security, if the server is trusted. However, they thinks that the Discord group they created is a "server" and therefore concludes that the messages cannot end up in the hands of any untrusted parties. Discord erroneously calls the social media groups within their service "servers".

Other examples of "Discord newspeak" and untruthful marketing

- Discord states on its website that it uses only the WebRTC and SSL protocols and aims to create a misconception that an actual protocol as a set of rules between the client and server programs is *old-fashioned.* It means that Discord does not officially have any protocol between the client and the server. That makes it unclear how the communication between the client and server programs actually happens within Discord, but it is clear that Discord does it in a *modern* way.
- Discord has officially stated to its beta testers that the client program crashes because of denial-of-service attacks against Discord "servers" which aren’t actually servers and thus cannot be attacked like that. Naturally that also shouldn’t crash the *client*.
- Discord's web site has a oneliner "Discord <3 open source", which has created a common misconception that Discord is an open source program. In reality Discord has never released their source codes and neither uses any free or open source license.
- Most communication programs market themselves as "multi-platform" and even claim to "work on all devices", when in reality their client program is only available for certain versions of Android, iOS and Windows operating systems. They are actively hostile towards technological diversity.
- Discord is marketing themselves as "replacement for IRC" and rides with the positive things that are associated to IRC, even though Discord itself does not have those properties. On their website Discord has also presented a claim that the server program of a competing product Teamspeak costs money, which is not true.