

IT Security Evaluation of Skype in Corporate Networks

Tapio Korpela
Helsinki University of Technology
tappik@cc.hut.fi

Abstract

Whilst the Skype community continues surging, using Skype Voice-over-IP (VoIP) and instant messaging services for business purposes is also becoming more popular. Therefore, corporate IT security managers are pressed to form an opinion on utilizing Skype in their local networks and terminals. However, using Skype in corporate networks makes special demands in comparison to private use, viz. the business continuity needs to be ensured by paying special attention to security and administration of Skype client software. This paper concentrates on those issues.

KEYWORDS: Skype, Voice-over-IP (VoIP), Peer-to-Peer, security

1 Introduction

Even though Skype Technologies has not, until recently, showed significant interest to business users, according to Skype one third of their regular customers are already utilizing Skype for business purposes. Recently, Skype Technologies has been persuading the business users of especially small and medium size enterprises that Skype suits well for their communications needs. For proving it, the company has presented Skype for business: a web site and tools for aiding companies in adoption and administration of Skype.[5][8]

Despite company security policies in general, Skype client software is apparently running on a multitude of corporate devices. This paper focuses on corporate IT security perspective of Skype. The aim is to point out what are the actual concerns from the point-of-view of enterprise IT security and find some ways to cope with the risks and administration problems that Skype brings along.

Firstly, the Skype network architecture and key functionalities are described. Then the attack scenarios are represented and their impact and likelihood is estimated. Further the controlling and administration problems with Skype are investigated. On the conclusion the suitability of Skype for business purposes is considered on the grounds of the previous chapters.

2 Skype network architecture and protocols

Skype uses peer-to-peer architecture and several common algorithms for its VoIP and instant messaging service. [1] This section will briefly introduce the main network entities

and the communication functions and encryption algorithms used among the entities.

2.1 Skype network entities

This subsection describes the Skype network entities: ordinary nodes, super nodes, bootstrap super nodes and central servers.

Initially every fresh installation of the Skype client software is an ordinary node. Ordinary nodes do not participate in auxiliary activities for Skype peer-to-peer network, they merely place and receive calls and messages.

It is estimated that half of the terminals are behind a firewall or NAT.[7] Albeit useful in many ways, firewalls and NATs can set challenges when connecting terminals behind them. They can confine inbound or outbound traffic to specific ports only, or totally prevent using specific protocols. Therefore, in order for Skype caller to reach Skype callee, these ordinary nodes behind firewalls and NATs need assistance from other nodes of the Skype peer-to-peer overlay network, nodes that are reachable on the public Internet. Secondly, one strength of a peer-to-peer network lies in its capability to share functions among the nodes instead of centralizing them to one main server. Skype network introduces super nodes for those two purposes. Just like ordinary nodes, super nodes are hosts running Skype client software. However, super nodes have been tested by the Skype network to be able to provide assistance to ordinary nodes. They have enough processing power, adequate network bandwidth capacity, reachability on the public Internet, and they are frequently on-line. These characteristics furtherance an ordinary node becoming a super node.[4]

Some of the functions need to be centralized, though. When setting up a new Skype account, the counterparty has to represent the Skype Technologies to be trustworthy. Central server is such a registration and certificate authority. There are, in fact, several central servers, for scalability reasons, and on the other hand, they are specialized to do different tasks. [1]

As mentioned earlier, in order to contact another node, an ordinary node needs assistance from super nodes. To have several super nodes always available, nodes keep track of the super nodes that are on-line in a so-called host cache with maximum of 200 nodes. Host caches are updated periodically, since super nodes are not throughout attainable. For newly-installed Skype client to be able to communicate with other nodes, few super nodes are hard-coded. Baset and Chulzrinne presented these as bootstrap super nodes. By utilizing them, Skype client can obtain more super nodes for host cache.[4]

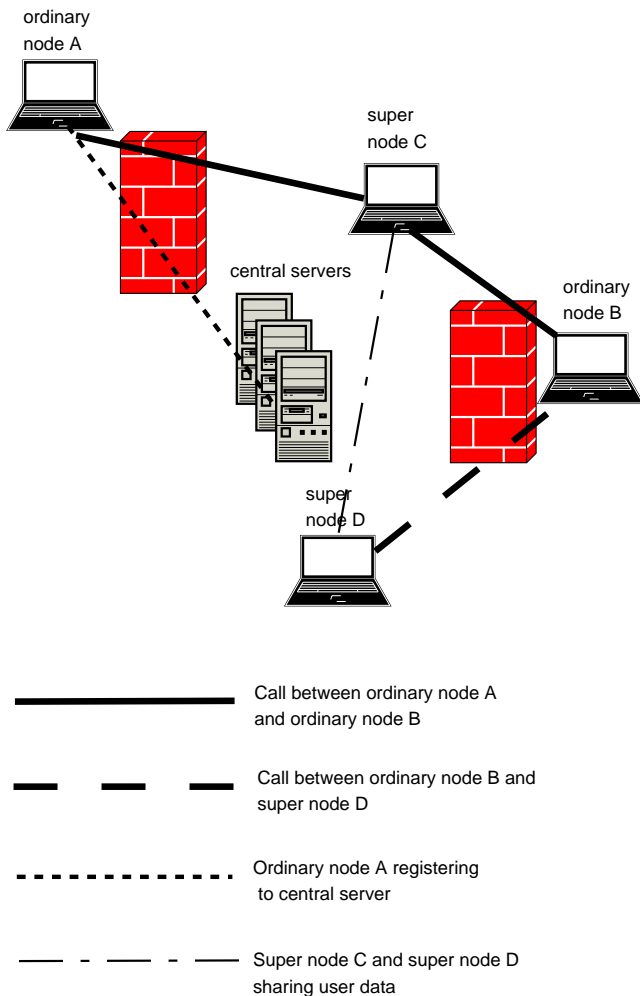


Figure 1: Skype network entities

Figure 1 depicts how nodes establish connections either directly or by using super nodes. Ordinary node A and ordinary node B are behind restricting firewalls. Therefore, they can not reach each other directly, but since they both have initiated connection to the super node C, it can mediate traffic between them. Even though the ordinary node B is behind a firewall, it can communicate directly with super node D, since outbound traffic is passed through. If super node D wishes to reach ordinary node B, assuming that there is no established connection between them, it can use other super node, say supernode C, to deliver a request for ordinary node B to contact super node D directly.

2.2 Registering to Skype network

Before a Skype client is ready to be used for placing calls and instant messages, Skype user registration must be done. For the registration process to start Skype user must firstly decide her username and password for the Skype service. Then Skype client software generates a one-way hash value from the password and an RSA key pair. Both the RSA private key and the password hash value are stored on the client computer. The client computer connects to one of the Skype Central Servers to request acceptance for the selected username

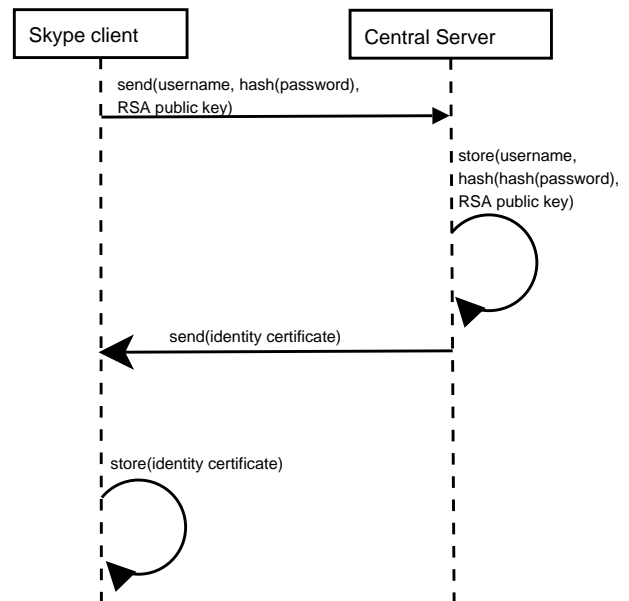


Figure 2: Sequence diagram of registration to Skype network

and password. The destination addresses are hard-coded to the Skype client software. If the chosen username is unique and in due form, the username, RSA public key and hash value of the password hash, are stored to the server. Central Server also generates and signs an identity certificate for the username.[1]

Note that according to Berson’s source code evaluation, Central Server does not store a user password, merely a hash of a hash of it, which is sufficient for verifying the correctness of a password.

Figure 2 represents the communication flow during the registration process.

2.3 Login to Skype network

According to experiments of Baset and Schulzrinne with Skype version 0.97.0.6, a node has to connect to both central server and super node in order to login to the Skype network.[4] Apparently the situation has changed since, because the tests performed by Samuel Korpi with Skype version 2.0.0.81 demonstrate that after first-time login central server is not needed for logging in to the network.[12]

2.4 User search

This subsection describes how users are found from the Skype Peer-to-Peer network.

When ordinary nodes login to Skype network, they send information about their presence to several super nodes, whereas the super nodes share the obtained information with other super nodes. Each super node has contact information of practically every other super node[11]. With this Global Index technique Skype Technologies claims that any node logged in within the last 72 hours can be found.[7]

User search starts by Skype client requesting from one of the super nodes in its host cache to either point out the lo-

cation of the searched user or deliver a list of super nodes that would potentially locate the object of the search. Skype client goes through the given list and sends the same kind of request to each of the listed super nodes. This procedure continues until the searched user is found or Skype client determines that the search was unsuccessful. Searches are made preferably by using UDP, but in case of restricting firewall or NAT, TCP can be used. Found user data can be stored to the so-called Buddy list in order to avoid going through the same search procedure next time.[4]

2.5 Authentication and session establishment between peers

This subsection describes the methods used to verify the identity of a counterparty and how a session is established between parties.

When establishing a connection between peers, direct TCP connection is made between them if both peers are freely reachable on the public Internet. If either one or both of the communicating peers are behind firewall or NAT restricting the reachability, a super node is used for connection establishment.[4] Since every on-line node has initiated connection to at least one super node, request for session start-up can be mediated via that route. That is, if a caller can not reach callee directly, it can ask the callee to try initiating direct connection to the caller. In case the caller does not receive direct messages, super node is used for transmitting the messages in both directions.[6]

Before a session between the communicating parties can be established, the peers need to check the identity of each other and agree on a session key. Identity is checked by exchanging the Identity Certificates and by using proprietary challenge-response protocol. Random number challenges of 64-bits are sent to each other, handled as agreed by the parties, signed by using RSA private keys and then the responses are sent back. For creating the 256-bit AES session key, both of the parties participate by generating a 128-bit pseudo-random number. After the session key has been established, all the data sent between the peers within the session is encrypted by using it.[1]

When transferring Skype session data, via a super node or not, UDP is favoured due to its more lightweight structure in comparison to TCP. Moreover, when sending voice or live video, acknowledging every packet is unnecessary since small packet loss is indistinguishable and retransmitted packets are likely to arrive too late anyway. However, Skype clients can use TCP for transmitting session data in case it is necessary due to restricting firewalls or NATs.[4]

3 Security issues with Skype

This section is divided to two subsections: attack scenarios and problems in administrating and controlling Skype in corporate networks and terminals.

3.1 Attack scenarios against Skype

3.1.1 Man-in-the-middle attack

In man-in-the-middle attack scenario a malicious attacker finds a way to get in the middle of conversation and pretending to both of the parties of being the other party. That way attacker can listen and pass through all the Skype traffic of the communicating parties.

Pretending to be someone else requires faking the identity, which in Skype network means both counterfeit or stolen identity certificate and private key. And in man-in-the-middle scenario, identity of both parties needs to be faked.

Besides, traffic needs to be directed through the attacker. Super node is a transit place due to its special role, but if the objective of an attacker is to follow a conversation of distinct two peers, the predictability of those peers utilizing attacker super node is challenging.

Man-in-the-middle attack could be possible by compromising the central server and its private signing key. It could also be achieved by defeating one of the hosts and having successful intervention, or by defeating both of the hosts.[1] With these terms man-in-the-middle attack seems intractable and fairly implausible to succeed.

3.1.2 Replay attack

In replay attack scenario an attacker studies the communication flow and tries to form a session by copying recently used communication data.

In Skype network one replay attack scenario could be learning challenges and responses. Attacker could replay previously used challenge. The other peer would send its own challenge back. If the attacker has learned the correct response for this, challenge-response would be defeated. Learning responses is laborious, though, since 64-bit challenges are used. Even then the attacker still needs to compromise AES key somehow, and replay attack against AES is even more arduous. Some other means of compromising AES is needed.[1] Considering the combinations, this type of attack scenario is hardly feasible.

3.1.3 Attacks against the password

As mentioned earlier, central servers do not store passwords, but hashes of hashes of them. Despite that, one scenario is flooding the central server with dictionary attack until match is found. However, central servers disconnect the login attempt after a few tries.[1]

3.1.4 Side channel attacks

Information about keys can sometimes be obtained by observing memory and CPU usage within a host. By exploiting such side channel leaks, a local malware program could compromise private key. However, Berson points out that local malware program could play a havoc a lot easier.[1]

3.1.5 Attack against the SkypeIn and SkypeOut

SkypeIn provides a phone number to Skype client. Callers from PSTN can place calls to Skype users that have bought a

SkypeIn number. SkypeOut instead provides a way to place calls to PSTN from Skype network. Both of these extra services are chargeable.

Since with SkypeIn and SkypeOut money is involved, more attention has been paid to security in using these. With SkypeIn and SkypeOut a 2048-bit Central Server key pair is used instead of a normally used 1536-bit modulus[1]. Although session is encrypted from the Skype client to the PSTN border, the voice call has to be decrypted for PSTN. Since that point is a transit place of a multitude of calls, it is also ideal place for eavesdropping, either for private or official quarter.[2]

3.2 Lack of control over Skype

The focus in this subsection is on controllability and administration problems with Skype.

3.2.1 Skype as a black box

Skype Technologies has made an abound effort to prevent outsiders learning the code. Skype software hinders using debuggers. It tries to discover debuggers and refuses to execute if it detects that it is running with a debugger. Code is also most thoroughly obfuscated to prevent reverse engineering. All this is troubling the corporate IT security personnel in general, since they want to investigate the existence of possible backdoors and security problems, and find sensible ways to block Skype traffic from their networks in case the software is considered harmful.[11]

3.2.2 NAT and firewall traversal

Skype uses STUN or TURN like techniques in super nodes for discovering restrictions in reaching ordinary nodes and bypassing them, thus providing access to ordinary nodes behind restricting NAT or firewall.[3] Since nodes can use UDP or TCP with random port numbers, packet sizes and packet spacing varies, it is hard to Skype traffic from corporate network. [4] This can be considered as a problem from corporate security point-of-view.

3.2.3 Super node

Super nodes accept connections from ordinary nodes and other super nodes. This is done for gathering user data and spreading it over the Skype peer-to-peer network. Another task of super nodes is to assist in placing calls when NATs and firewalls are involved.[6] Since the communication between super nodes is obscured, the traffic of a super node is hard to distinguish from other traffic. [10] However, it is unlikely that company network becomes a super node, since access to it is probably somehow limited.[9]

4 Conclusion

Anything unknown can be considered as a threat. But after all, Skype Technologies is most likely just aiming to defend its business by making Skype client a black box and trying to keep it as such. That is not even so extraordinary within

software business. The fact that a super node host is not entirely under the control of a Skype user is naturally worrying, but when using Skype in business networks this is normally not an issue, since they are not likely to be promoted to super nodes because the hosts are behind a firewall or NAT. Dreaded NAT and firewall traversal capability is a lifeline for Skype, since these network elements are hindering access to approximately half of the hosts. And it is in many respects due to using super nodes and ability to pass NAT that has made Skype services easy to use, robust and popular.

Seasoned cryptography and security expert with access to Skype source code could not find focal security flaws. And although Skype uses a lot of proprietary protocols in its software, cryptographic means are well known standards, and according to Berson's evaluation, they are correctly implemented. After reading several network monitoring studies and reverse-engineering attempts the results have not been alarming. In fact, Skype has given an image of a security-wisely well implemented product.

Skype could be (and probably is) accepted for business use within technology-friendly small organizations. However, in large enterprises with masses of users the lack of administrability becomes an issue. Presumably they are also seeking for legally binding contract, that is, they are willing to pay for their services to work.

References

- [1] Tom Berson. *Skype Security Evaluation*. Anagram Laboratories. October 2005. <http://www.skype.com/security/files/2005-031%20security%20evaluation.pdf>
- [2] Simson L. Garfinkel. *VoIP and Skype Security*. January 2005. <http://www.skypetips.internetvisitation.org/files/VoIP%20and%20Skype.pdf>
- [3] Saikat Guha, Neil Daswani, Ravi Jain. *An Experimental Study of the Skype Peer-to-Peer VoIP System*. January 2006. <http://www.guha.cc/saikat/pub/iptps06-skype.pdf>
- [4] Salman A. Baset, Henning Schulzrinne. *An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol* Columbia University. September 2004 <http://www1.cs.columbia.edu/~library/TR-repository/reports/reports-2004/cucs-039-04.pdf>
- [5] Skype. <http://www.skype.com>
- [6] Skype Technologies S.A. *Skype Guide for Network Administrators*. April 2005. <http://www.skype.com/security/guide-for-network-admins.pdf>
- [7] Skype Technologies S.A. *P2P Telephony Explained*. <http://www.skype.com/products/explained.html>
- [8] Jeremy Green. *Skype launches business version*. Ovum. March 2006. <http://www2.ovum.com/secure/p,61816>
- [9] Peter Hall. *Why do enterprises love to hate Skype?* Ovum. August 2006. <http://www.ovum.com/go/content/s,66507>

-
- [10] Edwin Mier, David Mier, Anthony Mosco. *Assessing Skype's network impact*. Miercom. Appeared on Network World. December 2005. <http://www.networkworld.com/reviews/2005/121205-skype-test.html>
- [11] Philippe Biondi, Fabrice Desclaux. *Silver Needle in the Skype*. Presentation in BlackHat Europe. March 2006. <http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-biondi/bh-eu-06-biondi-up.pdf>
- [12] Samuel Korpi. *Internet Telephony - Security Issues in Skype*. Helsinki University of Technology. 2006. http://www.tml.tkk.fi/Publications/C/21/Korpi_ready.pdf