# An analysis of Skype VoIP application for use in a corporate environment.

Version 1.3
October 2004

Dennis Bergström, CISSP
(dennis.bergstrom(at)gmail.com)

## Executive Summary

This report is an analysis of the VoIP application ”Skype” [version 1.0.0.29] available from [ www.skype.com ].

One of the scopes in this report was to investigate whether the Skype application is secure enough to deploy in a corporate environment.

The conclusion must be that the Skype application is currently *not* suitable or secure enough to be deployed in a corporate environment.

The reasons for this conclusion are:

1. **It seems to be hard to confine the Skype application to a corporate network**.

   At some point the Skype application clients *inside* the corporation must connect to *external* entities, both the centrally managed Skype back-end servers, but also to a arbitrary so called *supernode* which in most cases are some other Skype application end-user.

2. **It seems equally hard to block the Skype application in a corporate network-environment**.

   The Skype application will successfully work with a TCP-connection to port 80/tcp only and would probably traverse a corporate firewall with ease. Blocking outgoing traffic to port 80/tcp is usually not an option for a company and so the Skype application can work, albeit with reduced functionality.

3. **A corporation would probably have little or no influence of the traffic pattern of the Skype application clients**.

   Most of the *logic* behind the Skype network is handled either by Skype managed back-end servers *or* external *supernodes* and cannot probably be influenced by a corporation.

4. **All file transfers between users in the Skype application are encrypted.**

   Although the receiving Skype application users must *accept* a file before the transfer starts, this could be a possible path for virus or worms into a corporate network without ever being checked by a corporate anti-virus solution.

5. **The Skype application End User License Agreement (EULA) has very peculiar demands and vague wording.**

6. **There is no documentation of how the link-encryption and key-exchange is done**.

   As there are no documentation, it is unknown how serious the threats of intercepted conversations or *traffic analysis* are.

   There seems to be functionality in each and every Skype application client that transmits *call-statistics* to the centrally managed Skype server. This could probably have some impact on the privacy of the end-user *if* this mechanism also can be utilized for transferring the session-keys for a particular session, which would aid in the interception of the speech, instant messaging or file transfers. This alone will of course not cause any interception, but will probably aid an attacker – that have the ability to sniff the traffic between two Skype end-users – in decrypting the session-traffic.

## Contents

# 1. Introduction

This document is a work in progress. If you have any comments or additions feel free to mail the author at [*dennis.bergstrom(at)gmail.com*]

## 1.1 Revision History

- *Version 1.3 (October 2004)*
  Added Executive Summary.

- *Version 1.2* (October 2004)
  Minor changes in wordings. Added 1.4 Scope

- *Version 1.1* (October 2004):
  More detailed information about SkypeOut, Skype file transfers and Supernodes added.

- *Version 1.0* (September 2004):
  First draft.

## 1.2 Credits

The author wants to thank Mr. Aart J. Jochem, CISSP [*aart.jochem(at)capgemini.com*] for valuable input concerning file transfers in Skype.

## 1.3 Outline and Definitions

This report is a analysis of the VoIP application ”Skype” [version 1.0.0.29] available from [ www.skype.com ]. The report is divided in several parts, where the first part states information found on the Skype website, in the discussion-forums on the Skype website or on the Internet in general, the second part discusses some theoretical ways of attacking the Skype application end-user and the last part is a more technical discussion about certain aspects of the Skype application, both concerning security but also issues with deploying Skype in a corporate environment.

The word “Skype” is used in this report as an identification for the company *behind* the VoIP Skype *application*. When the VoIP Skype application is mentioned the word “application” always comes after “Skype”, so the reader can differentiate between the company and the application despite their identical names.

PSTN – which is discussed in association with SkypeOut – is an acronym for “**P**ublic **S**witched **T**elephone **N**etwork”. Telephone service carried by the PSTN is often called POTS or “**P**lain **O**ld **T**elephone **S**ervice”.

## 1.4 Scope

The scope of this document is twofold:

1. To give a brief background on the Skype application features and functionality and how the security-features of the Skype application could theoretically be circumvented.

2. To investigate whether the Skype application is suitable and secure enough to deploy in a corporate environment.

# 2. Background

## 2.1 What is Skype?

According to the Skype FAQ [ http://www.skype.com/help/faq/index.html ] the Skype application is:

*"… a free program that uses the latest P2P…technology to bring affordable and high-quality voice communications to people all over the world…"*

Users of the software – which is available for the Windows 2000, XP, Pocket PC, Mac OS X and Linux platforms – can connect to other users of Skype application to talk, send text messages or files. For an additional cost, users of the Skype application may also place calls to ordinary phones using the "SkypeOut" feature.

The call-setup and routing is all handled behind the scene by Peer-to-Peer technology acquired from JoltID [ http://www.joltid.com ] – founded by Mr. Niklas Zennström who was one of the original founders of Kazaa and also is a founder of Skype.

## 2.2 Behind the scene Peer-to-Peer

Resembling of Kazaa, the Skype application relies on *supernodes* – which in theory could be any private Skype application user with a decent bandwidth and CPU – in order to facilitate communication between users.

All communications – speech as well as text messages – are supposedly secured by AES 256-bit key encryption. Skype seems to be very secretive about how exactly the key-exchange mechanism work: there are hints that RSA is used, but there are no documentation to be found of the actual security of the link-encryption or how the client keys are protected.

The *supernodes* are actually not the whole truth about the Skype network, as there are other servers involved as well – all probably owned or operated by Skype – for instance servers that stores usernames, passwords and SkypeOut credits in a central database, servers that check user credentials against this database and so on.

The Skype application seems to be constantly evolving and there have been talk in the forums about migrating to a more harmonized Peer-to-Peer environment and move away from the centralized servers. If this is true remains to be seen. For the time being there are central servers for the management of users, SkypeOut credits, credentials etc. and *supernodes* for the call-setup and handling and occasionally *relayed* file-transmissions.

## 2.3 Some concerns with the Skype application

When reading through the Skype discussion-forums, several security concerns are constantly mentioned that have not – at the time of writing– been satisfactory resolved. Among these concerns are the following:

### 2.3.1 Encryption and Sessionkey exchange

According to the Skype site, the Skype application creates a new AES sessionkey for each transmission-session, be it speech, textmessages or file-transfer. It is unclear whether the session-key is *renegotiated* during a session or not. It is hinted that the session-keys are exchanged between clients with public-key encryption, but there is no public documentation available.

### 2.3.2 The End User License Agreement (EULA )

Although the Skype application is free to use, there are some issues with the End User License Agreement (EULA). As far as the author is aware the EULA is only showed when installing a new Skype application client and cannot be found on the Skype website. Despite this the EULA states that

*"…You acknowledge and agree that by clicking on the button labeled "SUBMIT", "DOWNLOAD", "I ACCEPT" or such similar links as may be designated by Skype to download the Skype Software to accept the terms and conditions of this Agreement, you are entering into a legally binding contract. You acknowledge that your electronic submissions constitute your agreement and intent to be bound by this Agreement…*"

Of course you cannot read the End User License Agreement *before* you have downloaded and installed the software, but nevertheless according to the extract above a user is bound to the agreement *before* s/he even have had a chance to review it.

Some other examples of concerns are for instance the following extract from the *License Restrictions* section of the EULA:

*"…Notwithstanding anything to the contrary, you may not: …(vii) collect any information or communication about the Network or users of the Skype Software or Services by monitoring, interdicting or intercepting any process of the Skype Software or the Network; or (viii) use any type of bot, spider virus, clock, timer, counter, worm, software lock, drop dead device*, packet-sniffer*, Trojan-horse routing, trap door, time bomb or any other codes or instructions that are designed to be used to provide a means of surreptitious or unauthorized access or that are designed to distort, delete, damage or disassemble the Skype Software, the Services or the Ne*twork…"

where the user is prohibited to use a packet-sniffer to capture traffic from the locally installed Skype application client. Several of the *other* prohibited types of "attacks" are quite strange: what is for instance a spider virus? (this should probably read spider, virus) ; why can't the user use a timer or a clock? etc. etc.

One could speculate that the real issue that Skype wants to protect with the EULA is the SkypeOut fee-based feature – where a Skype application user can dial users in the PSTN-network – since the EULA mentions "*…codes or instructions that are designed to be used to provide a means of surreptitious or unauthorized access…"* .

This could also possibly address a situation where a users impersonates another valid user.

One of the things with an End User License Agreement with very vague wordings as the examples show, is that the EULA can be applied to just about everything concerning the software.

The *License Restrictions* also mentions the following:

*"…The Skype Software and Services may be incorporated into, and may incorporate, technology, software and services owned and controlled by third parties. Skype emphasizes that it will only incorporate such third party software for the purpose of (i) adding new or additional functionality or (ii) improving the technical performance of the Skype Software and Services. Any other third party software which could be distributed together with Skype will be subject to you explicitly accepting a license agreement with this third party. Use of such third party software or services is subject to the terms and conditions of the applicable third party license agreements, and you agree to look solely to the applicable third party and not to Skype to enforce any of your right*s…"

which could open up for future inclusions of dubious "*third party software*", which may – or may not! – benefit the end-user.

### 2.3.3 Proactive countermeasures

Pay attention to the line "*…, interdicting or intercepting any process of the Skype Software…*" above:

The Skype application actually have a check so it will refuse to run if the debugger "*SoftICE*"
[ http://www.compuware.com/products/driverstudio/softice.htm ]
is present on the computer. SoftICE doesn't need to execute in system memory, just be locally installed on the harddisk to trigger the error-message:
" *…Skype is not compatible with system debuggers like SoftICE…*"

This is actually not true, since there are people that successfully circumvented this countermeasure. Please refer to
[ http://forum.skype.com/bb/viewtopic.php?t=5982&sid=6a3b255db066c835c0553bf32592c215 ] for a discussion of this concern.

SoftICE could of course possibly be used to reverse-engineer the application or the home-brewed protocols, but it is interesting to consider what *exactly* Skype would loose if knowledge of the internal workings of the application came to public knowledge?

In the discussion thread above there are speculations that Skype is actively hiding something in the code that may be used in the future, perhaps some sort of trojan or spyware. The infamous inclusion of several rather nasty spywares in Kazaa comes to mind and some of the people originally behind Kazaa are behind Skype.

### 2.3.4 SkypeOut

#### 2.3.4.1 Gateways

SkypeOut is a fee-based feature of the Skype application and network where a Skype application user can dial users in the PSTN-network. The translation between VoIP and the PSTN-network is done at *gateways*, which probably not is owned – or operated? – by Skype. Up to the gateway the calls made by SkypeOut is encrypted, but it is in the clear when travelling onto the PSTN-network.

The country locations of these gateways are not officially documented, but a search in the Skype forums [ http://forum.skype.com ] seems to indicate that *gateways at least* exists in the UK, USA, Sweden, Germany and China (!). It is unclear whether there are gateways in Africa, Australia or rest of Asia (not counting China).

According to a Skype employee on the forum:
"*…[Skype] have multiple gateways on several continents and the list is ever-increasing - it obviously makes sense to have the shortest possible call paths to everywhere around the world…*"

Nevertheless there are numerous complaints about bad sound-quality and call-setup problems to certain countries, for instance India.

#### 2.3.4.2 SkypeOut rates

The SkypeOut rates are not calculated from where you call, but *to which destination you call*. A user calling from China to the UK and a user calling from Sweden to the UK would pay the same rate.

SkypeOut features two different kind of rates:

- SkypeOut Global Rates (€ 0.017 / minute) to the following destinations:

  Argentina (Buenos Aires), Australia, Austria, Belgium, Canada, Canada (mobiles), Chile, Denmark, France, Germany, Ireland, Italy, Mexico (Mexico City, Monterrey), Netherlands, New Zealand, Norway, Portugal, Russia (Moscow, St. Petersburg), Spain, Sweden, United Kingdom, United States (except Alaska and Hawaii), United States (mobiles) and the Vatican.

- Individual Rates, ranging from € 0.026 / minute (Argentina except Buenos Aires) to € 1.080 / minute (Cook Island) and everything in-between.

#### 2.3.4.3 Credits and Credit cards

To be able to place calls with SkypeOut credits have to be purchased through the ”Skype Store” [ http://www.skype.com/store/buy/skypeout.html ].

A user can only purchase credits for €10 or €25 and only VISA and Dinners credit cards are accepted. There seems to be some sort of check whether the IP-address used when purchasing is located in the same country as the card-issuer.

There have been quite a few users on the Skype forum that complains about the Credit Card broker – “Bibit” [ http://www.bibit.com ] – not accepting certain VISA or Dinners cards. During an on-line discussion with the Skype support staff, the issue about declined credit cards came up and according to Skype the problem could be with the issuing bank as well, not only the broker.

One could speculate whether the declined credit cards were valid and legally acquired in the first place, but a question like this is probably out of scope for this document.

Some users apparently also have had problems with the Skype Live Help feature, but the author received help within a couple of minutes so this issue may have been resolved.

### 2.3.5 Supernodes

As mentioned elsewhere in this report, the Skype application heavily relies on Peer-to-Peer technology – although with some exceptions – and the path of the transmission is probably quite hard to predict.

Not only is the Skype infrastructure dependant on *supernodes* on the Internet that the end user cannot influence in any way, but it is impossible – at least in time of this writing – to locally deploy a Skype application on a corporate LAN/WAN without using public *supernodes* and the infrastructure and central servers of Skype.

Since the *supernodes* in reality is ordinary users that happen to have decent bandwidth and CPU, there is absolutely no guarantee that a specific *supernode* will exist for a prolonged period of time.

Perhaps one could say that this is a strength of Peer-to-Peer networking: that the network itself is marginally affected – if all – by a suddenly vanishing *supernode*.

The concept with *supernodes* that serve a specific number of end-users and where each *supernode* knows about a limited number of other *supernodes* probably presents a small risk of temporarily loosing connection to one or more Skype applications end-users if the local Skype application client looses its connection to a *supernode*. There are probably a reconnect feature of the Skype application client that finds another *supernode*, but it is unclear whether this *supernode* is aware of the location of the – now – temporarily lost end-users. This is probably why Skype uses a hybrid of centrally managed authentication servers and *supernodes*, in order to mitigate this threat. This is turn introduces a *single point of failure* since a breakdown of the central servers sooner or later must affect the *supernodes* and the end users.

As mentioned elsewhere in this report Skype seems to be very secretive about how things exactly work, so the comments above are obviously only speculations as there are no documentation about the exact interactions between *supernodes*, the centrally managed Skype servers and the end users.

It seems though that the *supernodes* are involved not only in call-setup when one or more parties are hidden behind a NAT-device, but also in so called *relayed* file transfers when a direct connection between two parties cannot be made:

*"…A relayed transfer means that you are unable to make a direct connection to the other party because of your firewall or NAT (Network Address Translation / router) configuration or that of the remote party. In this case, the file transfer is relayed though other peers on the network. When a transfer is relayed, Skype will limit the file transfer speed to 0.5 kB/second*. **Only peers with plenty of available bandwidth are used for relay purposes**…" [The authors emphasis]
[ http://www.skype.net/help/faq/filetransfer.html ]

The "peers" discussed above are probably *supernodes*, that – as mentioned elsewhere – *are* normal "peers" that happen to have decent bandwidth and CPU. The *relayed* file transfers – as the "normal" direct connection transfers – are supposedly encrypted

Interesting is also what is briefly mentioned in the Skype FAQ
[ http://www.skype.com/help/faq/index.html ]:

Skype is apparently gathering statistics from every call made by every Skype application client:

*"...What is the "minutes served" counter on skype.com front page?*

*The counter indicates that in its first year of operation, Skype has served more than 1 billion minutes of free Skype-to-Skype calls to its users. The counter is frequently updated based on the actual current number of minutes.*

*How do you know how many minutes Skype users have called to each other if all calls are encrypted?*

*Skype has built-in facilities to automatically gather anonymous usage statistics from its network and users, including the number of minutes spent on calls. We cannot track those minutes back to individual users and calls - your Skype calls are and continue to be secure*…"

This could probably have some impact on the privacy of the end-user *if* this mechanism also can be utilized for interception of the speech, instant messaging or file transfers by transferring the session-keys for that particular session. (This alone will of course not cause any interception, but will probably aid an attacker – that have the ability to sniff the traffic between two Skype end-users – in decrypting the session-traffic) It could also be an issue if *traffic analysis* can be done, i.e. revealing who talked to who, the time spent in conversation etc.

As before, this is only speculation, but it is probably not impossible to think that this *could* be done. Remember that Skype created both the *closed-source* Skype application as well as the protocols used between end-users, the backend servers and the *supernodes* and could possibly implement any number of features on any of those.

### 2.3.6 File transfers

File transfers between Skype application end-users are straightforward:

*"…There are several ways to start a file transfer. You can drag a file directly onto a Contact that is online. You can also right-click an online Contact and select "Send File". Lastly, you can select a Contact and press the "Send File" toolbar button (blue document icon). Skype then lets you send a file to send. The recipient will have to accept the transfer in order for it to begin. You can only send files to contacts who have authorized you…*" [ http://www.skype.com/help/faq/filetransfer.html ]

All file transfers are encrypted the same way as the Instant Messaging and speech, which raises some concerns where a Skype application user resides within a corporate LAN/WAN and receives one or more – encrypted – files that cannot be scanned by corporate anti-virus or screened by a firewall. Although the receiving users must *accept* a file before the transfer starts, this could be a possible path for virus or worms into a corporate network without ever being checked by a corporate anti-virus solution.

A mitigating factor could possible be a Deep Packet Inspection device that only allows certain well-known and well-formed protocols through the corporate perimeter.

As mentioned elsewhere in this report the supernodes can also be involved in a relayed file transfer where one or both of the parties *"…are unable to make a direct connection to the other… because of…[a] firewall or NAT (Network Address Translation / router) configuration…*"

### 2.3.7 Instant Messaging

The Skype application Instant Message feature allows a user to send instant messages to other end-users of the Skype application. The IM conversations held between Skype application users are protected by link-encryption, the same way as with speech and file transfers.

By accessing the Skype application *call-list* – which shows every call made from the end-user, it is possible to view the Instant Message History, which will be shown in a web browser.

The Instant Message History can also be retrieved by going to the directory

C:\Documents and Settings\<**logged in user in O/S**>\Application Data\Skype\<**Skype application Username**>\IMHistory

and viewing the files in this directory. It seems that the IM history is saved in .html-files named after the other party's Skype application username, i.e. a conversation with the Skype Echo Test Service – *echo123* – resulted in a file called *echo123.html*.

This could possible have some privacy issues where a complete log of every Instant Message sent to other users are available and saved on disk, unless manually cleared.

# 3. Attacking the end-user

As with every solution that uses link-encryption the best approach of attack is to attack one or both of the end points, which in this case are the Skype application end-users.

Actually, this have little to do with Skype and the Skype application *per se*, but the scenarios outlined below could very well jeopardize the security of the Skype application for the affected end-user.

## 3.1 File transfer intercept

To intercept file transfers between users, an attacker probably needs to have local disk access to one or both of the Skype application end-users or some other means of accessing the local disks. A local share, a Remote Access Trojan or a running Remote Desktop service would probably all be sufficient. In reality the file transfer *itself* could possibly not be intercepted – since the link-encryption would probably defeat this – but it is most probably sufficient to steal the files when they are eventually stored on the local harddrive.

A mitigating factor would probably be a good personal firewall or other protecting device, updated anti-virus / anti-trojan software and an updated operating system.

## 3.2 Instant Messaging intercept

To intercept Instant Messaging between users an attacker probably needs to have local disk access to one or both of the Skype application end-users *or* some other means of accessing the Skype application end-users *desktops*, since viewing or controlling the desktop will also reveal any Instant Messages. As with the File transfers above, the Instant Messages could probably not be intercepted *in transit* – since the link-encryption would defeat this – but it is sufficient to grab the Instant messages when they *appear on the desktop* or are saved in the Instant Message History directory

C:\Documents and Settings\<**logged in user in O/S**>\Application Data\Skype\<**Skype application Username**>\IMHistory

A local share, a Remote Access Trojan, a VNC-installation or running Remote Desktop service would probably all be sufficient for this purpose. Locally installed screen-grabbing or keylogging software could probably also be utilized in defeating the security of the Skype applications Instant messaging feature. As above, a mitigating factor would probably be a good personal firewall or other protecting device, updated anti-virus / anti-trojan software and an updated operating system.

## 3.3 Speech intercept

Intercepting speech is probably somewhat more complex than just grabbing files or viewing Instant Messages.

There are some trojans – for instance the infamous Sub7 (with the plug-in “Micro Recorder) or the grand old Netbus – that have capabilities to access the microphone of the victim computer. It is unclear whether these kinds of older trojans in reality could intercept speech in the Skype application, but sooner or later – with the rapid spread of VoIP applications – there *will* emerge more advanced trojans targeted at VoIP end-users.

As before, a mitigating factor would probably be a good personal firewall or other protecting device, updated anti-virus / anti-trojan software and an updated operating system.

# 4. Technical Aspects

## 4.1 Registry Entries

When the Skype application is first installed, the *SkypeSetup.exe* executable make seven very rapid changes to the Registry-entry *HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed* which could indicate a key generation phase.

It actually seems that the installation-file *SkypeSetup.exe* is an installation executable created with the freeware software "Inno Setup" [ http://www.jrsoftware.org/isinfo.php ] which – according to the Inno Setup website – is:
*"…a free installer for Windows programs. First introduced in 1997, Inno Setup today rivals and even surpasses many commercial installers in feature set and stability…*"

Among the features of Inno Setup are:

- Support for all 32-bit Windows versions in use today -- Windows 95, 98, 2000, 2003, XP, Me, NT 4.0. (No service packs are required.)
- Supports creation of a single EXE to install your program for easy online distribution. Disk spanning is also supported.
- Installation of files:
  Includes integrated support for "deflate", bzip2, and 7-Zip LZMA file compression. The installer has the ability to compare file version info, replace in-use files, use shared file counting, register DLL/OCX's and type libraries, and install fonts.
- Creation of registry and .INI entries.
- Integrated Pascal scripting engine.
- Full source code is available (Borland Delphi 2.0-5.0).

For some older versions of Inno Setup there have been a utility that could *extract* executable- and scriptfiles from the installation EXE-file, but this utility have been discontinued and the URL where the utility – *Inno Extractor* – once was located [ http://www.wintax.nl/ix ] does no longer work.

Considering that the full source code of Inno Setup is available and given enough time it should be possible to recreate a similar utility as above that can extract executable- and scriptfiles from the installation EXE-file of the *newest* version of Inno Setup but this is currently beyond the scope of this paper.

## 4.2 Installed files

When the Skype application is first installed the following files are saved on the disk by the installation program:

**Skype.lnk**
C:\DOCUMENTS AND SETTINGS\ALL USERS\START MENU\PROGRAMS\SKYPE\
**Skype.lnk**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\DESKTOP\
**SkypeVersionChecker.dll**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\LOCAL SETTINGS\TEMP\IS-99KQD.TMP\
**business-skype.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**hula-skype.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**make-skype-not-war.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-aid.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-cola.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-cool-shades.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-extreme.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-goaaaaal.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-headset.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-in-one.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-jah.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-me-sweetheart.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE

PICTURES\
**skype-safety.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-san.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-smiley.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype-up.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skype.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skypeaholic.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skypeness.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**skypers-of-the-caribbean.jpg**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\MY DOCUMENTS\MY SKYPE PICTURES\
**Skype.exe**
C:\PROGRAM FILES\SKYPE\PHONE\
**unins000.dat**
C:\PROGRAM FILES\SKYPE\PHONE\
**unins000.exe**
C:\PROGRAM FILES\SKYPE\PHONE\
**SKYPESETUP.EXE-015A2EF0.pf**
C:\WINDOWS\PREFETCH\
**SKYPESETUP.EXE-07F7392A.pf**
C:\WINDOWS\PREFETCH\

Up till now no users have been defined, only the supporting executables and pictures have been installed. When a user – *real_wiseman* – is created from the Skype application GUI the following files are added:

**shared.lck**
C:\DOCUMENTS AND SETTINGS\ALL USERS\APPLICATION DATA\SKYPE\
**shared.xml**
C:\DOCUMENTS AND SETTINGS\ALL USERS\APPLICATION DATA\SKYPE\
**config.lck**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\APPLICATION DATA\SKYPE\REAL_WISEMAN\
**config.xml**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\APPLICATION DATA\SKYPE\REAL_WISEMAN\
**index.dat**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\APPLICATION DATA\SKYPE\REAL_WISEMAN\
**profile256.dbb**
C:\DOCUMENTS AND SETTINGS\DBERGSTR\APPLICATION DATA\SKYPE\REAL_WISEMAN\
**SKYPE.EXE-30AE1A60.pf**
C:\WINDOWS\PREFETCH\

If the locally stored user-files above for some reason are deleted they are recreated at the next login, confirming that there are indeed centrally located and Skype managed authentication back-end servers.

## 4.3 Supernodes

Please note the file "*shared.xml*", which holds a list of *supernodes* and probably also other Skype applications peers:

```
"<HostCache>

<_1>140.115.111.219:25465</_1>
<_10>202.199.162.66:24983</_10>
<_100>83.89.66.162:20714</_100>
<_101>129.123.212.37:22135</_101>
<_102>68.58.65.165:53510</_102>
<_103>24.167.51.203:1345</_103>
<_104>130.238.140.170:23005</_104>
<_105>193.226.227.142:63106</_105>
<_106>81.108.194.153:24469</_106>
<_107>24.161.189.79:11086</_107>
<_108>64.246.49.61:52528</_108>
<_109>62.194.90.226:51121</_109>
<_11>68.10.200.24:63291</_11>
<_110>24.250.145.217:17672</_110>
<_111>140.115.51.161:62784</_111>
<_112>128.195.10.230:59886</_112>
<_113>69.137.137.95:8581</_113>
<_114>219.233.154.138:22509</_114>
<_115>24.210.94.156:24633</_115>
<_116>213.118.111.203:6319</_116>
<_117>24.90.210.119:15205</_117>
<_118>61.126.140.106:15502</_118>
<_119>140.127.192.67:7676</_119>
<_12>140.117.241.165:14163</_12>
<_120>24.13.20.127:47102</_120>
<_121>130.236.233.126:62693</_121>
<_122>155.69.21.134:14057</_122>
<_123>24.14.13.217:25099</_123>
<_124>203.68.230.216:41361</_124>
<_125>203.32.82.120:23806</_125>
<_126>24.111.12.75:8941</_126>
<_127>140.114.207.223:35623</_127>
<_128>68.39.53.181:48800</_128>
<_129>220.105.139.12:20485</_129>
<_133>150.146.26.86:33053</_133>
(Entries omitted for readability)
</HostCache>"
```

Quite interesting is that the Skype application only makes TCP-connections with a few of the hosts above, the rest are contacted with an UDP-connections.
This could mean that only *some* – those contacted with TCP – are *supernodes* and the rest possibly are only *peers* connected to the same supernodes as the authors Skype application client. If this really is the case is unclear.

It is also unclear how the list above is compiled:

An uninstall of Skype leaves the file
*C:\Documents And Settings\All Users\Application Data\Skype\ shared.xml*
seemingly untouched. When manually deleted it is reconstructed when a new copy of the Skype application is installed.

One could speculate whether this list is dynamically downloaded from the Skype-servers or not. When sniffing the conversation from the Skype application with *Ethereal* there are only two brief initial connections to the Skype backend servers:

One seems to be a version-checking connection to 80.160.91.13:

*GET /ui/0/1.0.0.29/en/getlatestversion HTTP/1.1*
*Content-Type: text/html*
*Host: ui.skype.com*
*Accept: text/html, */**
*User-Agent: SkypeSetup 1.0.0.29*

*HTTP/1.1 200 OK*
*Date: Thu, 30 Sep 2004 07:33:32 GMT*
*Server: Apache*
*Cache-control: no-cache, must revalidate*
*Pragma: no-cache*
*Expires: 0*
*Connection: close*
*Transfer-Encoding: chunked*
*Content-Type: text/html; charset=utf-8*
*Content-Language: en*

*8*
*0.98.0.1*
*0*

The other connection seems to be some sort of “phone-home” connection to the same server as above:

*GET /ui/0/1.0.0.29/en/installed HTTP/1.1*
*User-Agent: Skype. 1.0*
*Host: ui.skype.com*
*Cache-Control: no-cache*

*HTTP/1.1 200 OK*
*Date: Thu, 30 Sep 2004 07:35:54 GMT*
*Server: Apache*
*Cache-control: no-cache, must revalidate*
*Pragma: no-cache*
*Expires: 0*
*Content-Length: 0*
*Connection: close*
*Content-Type: text/html; charset=utf-8*
*Content-Language: en*

Nowhere in the trace could the data in the *HostCache* section of the *shared.xml* file be seen, but immediately after the connection above was finished, the locally installed Skype Application client started to send UDP-packets to hosts in that very list.

This seems to indicate that some potions of the list is hardcoded into the application itself, but this – as almost all things Skype! – is unclear as there are no officially technical documentation that covers the internal workings of Skype.

The importance of the *supernodes* seems quite clear though, since all call-handling is seemingly done by *supernodes*. Since the end-user have no way of influencing which *supernode* the Skype application connects to or even if the end-users computer *itself* is promoted to a *supernode* there are no easy steps – at least in the time of writing – for running the Skype application confined to a corporate network. At some point the Skype application clients *inside* the corporation must connect to external entities, both the centrally managed Skype back-end servers, but also to a arbitrary *supernode* which in most cases is some other Skype application end-user.

## 4.4 Ports used

According to [ http://www.skype.com/help/faq/technical.html ]:

*"…The minimum requirement is that Skype needs unrestricted outgoing TCP access to all destination ports above 1024 or to port 80 (the former is better, however). If you don't allow either of those, Skype will not work reliably at all. Voice quality and some other aspects of Skype functionality will be greatly improved if you also open up outgoing UDP traffic to all ports above 1024, and allow UDP replies to come back in.*

*In the quest for even better voice quality, it is also advisable to open up incoming TCP and/or UDP to the specific port you see in Skype Options. This port is chosen randomly when you install Skype. In the case of firewalls, this should be easy to arrange. In some routers, however, you cannot configure incoming UDP at all (but you still can configure incoming TCP port forwarding, which you could/should do).*

*The randomness in port selection is to improve NAT traversal for cases where several users are behind the same NAT; if they all used same ports, many NATs would behave in a way that would reduce Skype voice quality…*"

Skype is quite skilled at traversing NAT-devices. To do this each Skype application client probably uses the *supernodes* as a relay and utilize something called "*UDP Hole Punching*"

[ http://mirrors.isc.org/pub/www.watersprings.org/pub/id/draft-ford-natp2p-00.txt ]

Blocking the Skype application in a corporate network-environment could prove to be quite hard. Looking at the extract above the Skype application can successfully work with a TCP-connection to port 80/tcp only and would probably traverse the corporate firewall with ease. Blocking outgoing traffic to port 80/tcp is usually not an option for a company and so the Skype application can work, albeit with reduced functionality.

By using a Deep Packet Inspection device that only allows certain well-known, well-formed protocols through the corporate perimeter this situation could possibly be mitigated., but whether it would be 100% successful or not is anybody's guess.

# 5. Conclusions

The Skype VoIP application is interesting, since it uses Peer-to-Peer technology to let end-users place calls more or less free of charge and for private use Skype and the Skype application is probably a quite decent choice.

For the corporate world however, Skype and the Skype application are probably unsuitable for several reasons:

1. As call-handling is seemingly done by *supernodes* it is hard – at least in the time of writing – to confine the Skype application to a corporate network. At some point the Skype application clients *inside* the corporation must connect to external entities, both the centrally managed Skype back-end servers, but also to a arbitrary *supernode* which is in most cases are some other Skype application end-user.
   Blocking the Skype application in a corporate network-environment could also prove to be quite hard. The Skype application will successfully work with a TCP-connection to port 80/tcp only and would probably traverse a corporate firewall with ease. Blocking outgoing traffic to port 80/tcp is usually not an option for a company and so the Skype application can work, albeit with reduced functionality.

2. A corporation would probably have little of no influence of the traffic pattern of the Skype application clients, nor any chance of decent error-handling, since most of the *logic* behind the Skype network is handled either by Skype managed back-end servers *or* external *supernodes.*

3. All file transfers in the Skype application are encrypted the same way as the Instant Messaging and Speech, which raises some concerns where a Skype application user resides within a corporate LAN/WAN and receives one or more – encrypted – files that cannot be scanned by corporate anti-virus or screened by a firewall. Although the receiving users must *accept* a file before the transfer starts, this could be a possible path for virus or worms into a corporate network without ever being checked by a corporate anti-virus solution.

4. The Skype application in itself raises some concerns: not only the quite peculiar *End User License Agreement* (EULA) mentioned elsewhere in this document but also the pro-active measures by the application to withstand reverse-engineering by the debugger SoftICE. One of the speculations is that Skype is actively hiding something in the code that may be used in the future, possibly some sort of trojan or spyware. The infamous inclusion of several rather nasty spywares in Kazaa comes to mind and some of the people originally behind Kazaa are behind Skype.

5. The absence of decent documentation of how the link-encryption and key-exchange is done is quite worrying. Private conversations by end-users are probably often not confidential, but corporate use of the Skype application would probably lead to confidential things being discussed, which may – or may not! – be properly protected.

   As mentioned elsewhere in this report every single Skype application client forwards calls *statistics* to the centrally managed Skype server, which is evident in the banner "*Minutes Served*" at [ http://www.skype.com ]. Skype claims that usage statistics are completely anonymized. This could probably have some impact on the privacy of the end-user *if* this mechanism can beutilized for transferring the session-keys for a particular session, which would aid in the interception of the speech, instant messaging or file transfers. (This alone will of course not cause any interception, but will probably aid an attacker – that have the ability to sniff the traffic between two Skype end-users – in decrypting the session-traffic.) It could also be an issue if *traffic analysis* can be done, i.e. revealing who talked to who, the time spent in conversation etc.

   As before, this is only speculation, but it is probably not impossible to think that this *could* be done. Remember that Skype created both the *closed-source* Skype application as well as the protocols used between end-users, the backend servers and the *supernodes* and could possibly implement any number of features on any of those.

# 6. Appendix A – References

Schneier, Bruce
*Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd Ed.
ISBN:0-471-11709-9
[ http://www.amazon.com/exec/obidos/tg/detail/-/0471117099/qid=1096538271/sr=1-1/ref=sr_1_1/104-0104982-3071926?v=glance&s=books ]

Bibit Global Payment Services
[ http://www.bibit.com ]

Ethereal – Network Protocol Analyzer
[ http://www.ethereal.com/ ]

Filemon
[ http://www.sysinternals.com/ntw2k/source/filemon.shtml ]

Inno Setup
[ http://www.jrsoftware.org/isinfo.php ]

Packetyzer - Packet Analyzer for Windows
[ http://www.networkchemistry.com/products/packetyzer ]

Regmon
[ http://www.sysinternals.com/ntw2k/source/regmon.shtml ]

Skype
[ http://www.skype.com ]

SoftICE
[ http://www.compuware.com/products/driverstudio/softice.htm ]

“UDP Hole Punching”
[ http://mirrors.isc.org/pub/www.watersprings.org/pub/id/draft-ford-natp2p-00.txt ]