

# **Skype**

## **Security Considerations and Management**

**ipoque**

**[www.ipoque.com](http://www.ipoque.com)**

- 2002: founded by Niklas Zennström and Janus Friis
  - creators of the Kazaa P2P file sharing network
- Internet telephony (incl. video), instant messaging, file transfer
  - proprietary network protocol and implementation
  - strong encryption
- October 2005: acquisition by eBay
  - for €2.1 billion (€1.3 billion cash)
  - plus €1.2 billion performance-based options

- 276 million registered users by end of 2007
  - 129% increase in 2006
  - 61% increase in 2007
- 8-11 million users simultaneously online
- High diurnal usage variations
  - 40-50% more users during working hours
  - 25% more users on weekdays
- Longer call duration compared to PSTN
  - Ø PSTN: 3 minutes
  - Ø Skype: 13 minutes
  - most likely because Skype calls are free

- Peer-to-peer (P2P) network architecture
  - supernode architecture similar to the KaZaa FastTrack protocol
- Very easy to use
- Works in almost any network environment
- Advanced obfuscation techniques
  - both in the code and the network traffic
- Generates traffic even when idle

- Encrypted program code
  - code pieces are decrypted to memory at run time
- Code integrity checksums to prevent modifications
- Code obfuscation techniques
  - dummy conditional jumps with never-used code
  - fake error handlers
- Many anti-debugging measures
  - Skype does not run if debuggers are active (incl. the SoftICE kernel mode debugger)
  - checksums are used to detect software breakpoints

- P2P architecture
- Uses UDP and TCP, both for signaling and communication
- No fixed ports
  - a UDP port is randomly selected at installation time and used for all UDP data
  - HTTP and HTTPS ports (80 & 443) can be used
- Works behind firewalls and NAT gateways

- Penetrates most firewall systems
  - there is nothing firewalls can evaluate (such as port numbers, payload patterns)
- Works behind NAT gateways
  - uses NAT hole punching techniques similar to STUN and TURN
  - only requires a single connection to a supernode initiated by the client to be fully operational

- One peer NATed
  - the NATed peer always opens the connection, even if the unNATed peer initiates a call
- Both peers NATed
  - both peers communicate their NAT details to a supernode, which forwards them to the peer via the existing supernode connection
  - non-symmetric NAT: both simultaneously open a connection to the other peer (STUN)
  - symmetric NAT: both peers try to open a connection using a port scan (STUN+); if that fails, an external relay is used (TURN)

- Supernodes (SN)
  - implement the Global Index, the Skype user directory
  - essential for the proper operation
- Relay nodes (RN)
  - call forwarding for clients behind NAT gateways
- Differentiation between SN and RN not clear
- Every client with a public IP and sufficient resources can become a SN or RN
  - this can only be disabled for the latest Windows client by tweaking the Registry
  - easier to become a RN
- Estimate: >250,000 supernodes worldwide

- Idle traffic
  - 0-0.5 kBytes/s mainly for contact presence updates
  - $0.5 * 3600 * 24 * 30 = 1.2 \text{ GBytes/month}$  (!)
- Active call traffic
  - 3-16 kBytes/s
- Relay nodes
  - $X * 3-16 \text{ kBytes/s}$ , but how big is X?
  - Relayed file transfers capped at 1 kByte/s
- Supernodes
  - $< 5 \text{ kBytes/s}$  (?)
  - connections to many other clients
  - potential problems for routers and other network devices

- Skype is a perfect backdoor into a network
- Interferes with IDS/IPS
  - Skype looks like an attack or unknown traffic
- Known exploits

Source: <http://www.skype.com/security/bulletins.html>

Year	Vulnerabilities	Critical (injected code execution)
2004	2	1
2005	3	1
2006	2	1
2007	2	2
2008	4	4

- 2/2007: Skype gathers computer BIOS data
- Hard to enforce a security policy with Skype

- No well-known ports or server IP addresses
- Bit patterns insufficient for detection
  - nearly everything is encrypted
  - known bit patterns only cover some flows, not all
  - for instance, if Skype uses port 443, it mimics a valid HTTPS connection setup
  - patterns change from version to version
- If one flow is blocked, Skype tries something else (i.e. different port, different transport protocol)

- Requires tracking of TCP and UDP flows
- Different flow patterns for TCP and UDP
- Flow “patterns”, or signatures
  - absolute and relative packet sizes
  - flow count and arrival rate
- Different connection mechanisms require different signatures
- Regular signatures updates necessary
  - signatures change often
  - major changes in Skype 3.0
  - ⇒ Skype closely monitors and counters detection efforts

Skype v2.0.0.43	Skype v2.5.0.151	Skype v3.1.0.150
Skype v2.0.0.63	Skype v2.5.0.154	Skype v3.1.0.152
Skype v2.0.0.69	Skype v2.6.0.67	Skype v3.2.0.53
Skype v2.0.0.73	Skype v2.6.0.74	Skype v3.2.0.63
Skype v2.0.0.76	Skype v2.6.0.81	Skype v3.2.0.82
Skype v2.0.0.79	Skype v2.6.0.97	Skype v3.2.0.115
Skype v2.0.0.81	Skype v2.6.0.103	Skype v3.2.0.145
Skype v2.0.0.90	Skype v2.6.0.105	Skype v3.2.0.148
Skype v2.0.0.97	Skype v3.0.0.106	Skype v3.2.0.152
Skype v2.0.0.103	Skype v3.0.0.123	Skype v3.2.0.158
Skype v2.0.0.105	Skype v3.0.0.137	Skype v3.2.0.163
Skype v2.0.0.107	Skype v3.0.0.154	Skype v3.2.0.175
Skype v2.5.0.72	Skype v3.0.0.190	Skype v3.5.0.107
Skype v2.5.0.82	Skype v3.0.0.198	Skype v3.5.0.158
Skype v2.5.0.91	Skype v3.0.0.205	Skype v3.5.0.178
Skype v2.5.0.113	Skype v3.0.0.209	Skype v3.5.0.202
Skype v2.5.0.122	Skype v3.0.0.214	Skype v3.5.0.214
Skype v2.5.0.126	Skype v3.0.0.216	Skype v3.5.0.229
Skype v2.5.0.130	Skype v3.0.0.217	Skype v3.5.0.234
Skype v2.5.0.137	Skype v3.0.0.218	Skype v3.5.0.239
Skype v2.5.0.141	Skype v3.1.0.112	Skype v3.6.0.127
Skype v2.5.0.146	Skype v3.1.0.144	Skype v3.6.0.159

- Only the connection setup is detected
  - existing connections cannot be detected;  
a call may use an existing connection and succeed
- It takes some packets to detect a new connection
  - a call may appear to succeed, but no conversation will be possible
  - the contact list may become available periodically

- The detection engine must see all inbound and outbound packets of a Skype client
  - no asymmetric routing
  - no other device blocking Skype packets
- Unusual network conditions
  - many IPs behind a single NAT
  - unusual NAT behavior

- Logging, Statistics and Accounting
  - aggregated and for individual users
- Shaping
  - limit the bandwidth available to Skype
- Prioritization
  - bandwidth reservation for Skype to ensure QoS
- Blocking
- All can be done for users and user groups

- Bandwidth graphs
  - per last hour, day, week, month, year
- Top talkers
  - per user and protocol
- Connection logging
  - for every new connection
  - including IP, start and end time
- Statistics export
  - in CSV format via FTP
  - to a syslog server

- ipoque's Skype detection works very reliably
- But: a 100% accuracy is impossible
- Regular tests and firmware updates
- Currently version 1.4 to 3.8 have been tested

- [An Experimental Study of the Skype Peer-to-Peer VoIP System](#), Saikat Guha (Cornell University), Neil Daswani, Ravi Jain (Google), 2/2006
- [Silver Needle in the Skype](#), Philippe Biondi, Fabrice Desclaux, EADS, Black Hat Europe 2006, 3/2006
- Vanilla Skype (part [1](#)+[2](#)), Fabrice Desclaux, Kostya Kortchinsky, EADS, RECON2006, 6/2006
- [An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol](#), Salman A. Baset and Henning Schulzrinne, Columbia University, 1/2006