Overview

- Web
- Consistent hashing
- Peer-to-peer
  - Motivation
  - Architectures
- TOR
- Skype
- CDN
- Video

Tor Anonymity Network

- Deployed onion routing network
  - [http://torproject.org](http://torproject.org)
  - Specifically designed for low-latency anonymous Internet communications
- Running since October 2003
  - Thousands of relay nodes, 100K-500K? of users
- Easy-to-use client proxy, integrated Web browser
  - Not like FreeNet – no data “in” TOR
  - Really an overlay – not pure peer-to-peer

Based on slides by Vitaly Shmatikov

Tor Circuit Setup (1)

- Client proxy establish a symmetric session key and circuit with relay node #1
- All data sent over the circuit is encrypted

\[ A = K(B)_k \]
Tor Circuit Setup (2)

- Client proxy extends the circuit by establishing a symmetric session key with relay node #2
  - Tunnel through relay node #1

Using a Tor Circuit

- Client applications connect and communicate over the established Tor circuit
  - Datagrams decrypted at each link
  - Also want end-to-end encryption – not done by Tor

Tor Circuit Setup (3)

- Client proxy extends the circuit by establishing a symmetric session key with relay node #3
  - Tunnel through relay nodes #1 and #2

Using Tor

- Many applications can share one circuit
  - Multiple TCP streams over one anonymous connection
- Tor router doesn't need root privileges
  - Encourages people to set up their own routers
  - More participants = better anonymity for everyone
- Directory servers
  - Maintain lists of active relay nodes, their locations, current public keys, etc.
  - Control how new nodes join the network
    - "Sybil attack": attacker creates a large number of relays
  - Directory servers’ keys ship with Tor code
Overview

- Web
- Consistent hashing
- Peer-to-peer
  - Motivation
  - Architectures
  - TOR
  - Skype
- CDN
- Video

What is Skype?

- Peer-to-peer, pc-to-pc, pc-to-phone, phone-to-pc VoIP and IM client
- Developed by people who created KaZaa
- SkypeOut (pc-to-phone)
  - SkypeOut terms of service: governed by the laws of Luxembourg
- Skypeln (phone-to-pc), voicemail
- Supported OS: Windows, Linux, MacOS, PocketPC
- A p2p illusion
  - Login server
  - Buddy-list server
  - Servers for SkypeOut and Skypeln
  - Anonymous call minutes statistic gathering

What problems does it solve?

- NAT and firewall traversal
  - Nielsen September 2005 ratings
    - 61.3% of US home internet users use broadband
      (http://www.nielsen-netratings.com/pr/pr_050928.pdf)
    - ‘Most’ users have some kind of NAT
- Phone-to-pc calling, Skypeln
- Configuration-less connectivity

The Skype Network
The Skype Network (contd…)

• Ordinary host (OH)
  • A Skype client
• Super nodes (SN)
  • A Skype client
  • Has public IP address, 'sufficient' bandwidth, CPU and memory
• Login server
  • Stores Skype id's, passwords, and buddy lists
  • Used at login for authentication
  • Version 1.4.0.84: 212.72.49.141 and 195.215.8.141

Skype Components

• Ports
  • No default listening port
  • Randomly chooses a port (P1) on installation
  • Opens TCP and UDP listener sockets at P1
  • Opens TCP listener sockets at port 80 (HTTP) and port 443 (HTTPS)

Skype Components (contd…)

• Host cache (HC)
  • IP address and port number of online Skype nodes (SNs)
  • Maximum size: 200 entries
• Liang, Kumar and Ross. Understanding KaZaA
  • 200 entries for ordinary nodes (ON)
• Login server IP address and port number
• HC Windows location
  C:\Documents and Settings\All Users\Application Data\Skype

Experimental Setup

• We have NOT reverse engineered Skype executable but it can be done (Biondi and Desclaux)
• Skype version: Linux v1.2, Windows v1.4.0.84
• Experiments performed between June-July and Nov-Dec 2005
• Tools Used
  • Ethereal (for packet capture)
  • NetPeeker (for tuning per process bandwidth)
  • NCH Tone generator (for generating tones of various frequencies)
  • API Monitor (for monitoring the system calls)
  • LD_PRELOAD: Linux shared library and system call interception
    • Skype fails to run with ltrace and strace
Skype Functions

- Startup
- Login
- User search
- Call establishment
- Media transfer
- Keep-alive
- NAT and firewall traversal
- Conferencing

What is Skype?

- Peer-to-peer, pc-to-pc, pc-to-phone, phone-to-pc VoIP and IM client
- Developed by people who created KaZaa
- SkypeOut (pc-to-phone)
  - SkypeOut terms of service: governed by the laws of Luxembourg
- SkypeIn (phone-to-pc), voicemail
- Supported OS: Windows, Linux, MacOS, PocketPC
- A p2p illusion
  - Login server
  - Buddy-list server
  - Servers for SkypeOut and SkypeIn
  - Anonymous call minutes statistic gathering

What problems does it solve?

- NAT and firewall traversal
  - Nielsen September 2005 ratings
    - 61.3% of US home internet users use broadband
    - (http://www.nielsen-netratings.com/pr/pr_050928.pdf)
  - ‘Most’ users have some kind of NAT
- Phone-to-pc calling, SkypeIn
- Configuration-less connectivity

The Skype Network

Based on slides by Baset and Schulzrinne (Infocom 06)
The Skype Network (contd…)

- Ordinary host (OH)
  - A Skype client (SC)
- Super nodes (SN)
  - A Skype client (SC)
  - Has public IP address, ‘sufficient’ bandwidth, CPU and memory
- Login server
  - Stores Skype id’s, passwords, and buddy lists
  - Used at login for authentication
  - Version 1.4.0.84: 212.72.49.141 and 195.215.8.141

Skype Components: Ports

- No default listening port
- Randomly chooses a port (P1) on installation
- Opens TCP and UDP listener sockets at P1
- Opens TCP listener sockets at port 80 (HTTP) and port 443 (HTTPS)

Skype Components: Host Cache

- IP address and port number of online Skype nodes (SNs)
- Maximum size: 200 entries
- Liang, Kumar and Ross. *Understanding KaZaA*
  - 200 entries for ordinary nodes (ON)
- Login server IP address and port number
- HC Windows location: C:\Documents and Settings\All Users\Application Data\Skype

Experimental Setup

- We have NOT reverse engineered Skype executable but it can be done (Biondi and Desclaux)
- Skype version: Linux v1.2, Windows v1.4.0.84
- Experiments performed between June-July and Nov-Dec 2005
- Tools Used
  - Ethereal (for packet capture)
  - NetPeeker (for tuning per process bandwidth)
  - NCH Tone generator (for generating tones of various frequencies)
  - APIMonitor (for monitoring the system calls)
  - LD_PRELOAD: Linux shared library and system call interception
  - Skype fails to run with ltrace and strace
Skype Functions

- Startup
- Login
- User search
- Call establishment
- Media transfer
- Keep-alive
- NAT and firewall traversal
- Conferencing

Skype Functions: LOGIN

- Public, NAT
  - Establishes a TCP connection with the SN
  - Authenticates with the login server
  - Announces arrival on the network (controlled flooding)
  - Determines NAT type
- Firewall
  - Establishes a TCP connection with the SN
  - Authenticates with the login server

Skype Functions: USER SEARCH

- From the Skype website
  - Guaranteed to find a user it exists and logged in the last 72 hours
- Search results are cached at intermediate nodes
- Unable to trace messages beyond SN
- Cannot force a node to become a SN
  - Host cache is used for connection establishment and not for SN selection
- User does not exist. How does search terminate?
  - Skype contacts login server for failed searches
- SN searches for a user behind UDP-restricted firewall
- Same wildcard (sal*) search query from two different machines initiated at the same time gives different results

Skype Functions: CALL ESTABLISHMENT

- Call signaling always carried over TCP and goes e2e
- Calls to non buddies=search+call
- Initial exchange checks for blocked users
- Public-public call
  - Caller SC establishes a TCP connection with callee SC
- Public-NAT
  - Caller SC is behind port-restricted NAT
    - Different solutions based on the nature of the NAT
  - Caller---->Skype node (SN?) ----> Callee
  - TCP connection established between caller, callee, and more than one Skype nodes
- Firewall-firewall call
  - Same as public-NAT but no in-UDP packets
Skype Functions: MEDIA TRANSFER

- No silence suppression
- Silence packets are used to
  - play background noise at the peer
  - maintain UDP NAT binding
  - avoid drop in the TCP congestion window
- Putting a call on hold
  - 1 packet/3 seconds to call-peer or Skype node
  - same reasons as above
- Codec frequency range
  - 50-8,000 Hz (total bw of 3 kilobytes/s)
- Reasonable call quality at (4-5 kilobytes/s)

Skype Functions: KEEP ALIVE

- Refresh message over TCP to SN every 120 seconds
- Refresh message size: 2 bytes

Skype Super Nodes

- Skype super node: A node with which a Skype establishes a TCP connection at login
- 8,153 successful login attempts over four days
- 35% hostnames had a .edu suffix
  - 102 universities
  - 894 unique super nodes
- Unique SN IP distribution:
  - US 83.7%, Asia 8.9%, Europe 7.1%
- Top 20 nodes received 43.8% of the total connections
- Top 100 nodes: 70.5%

Summary

- Selfish application
  - Uses best CPU and bandwidth resources
  - Evades blocking
  - Change application priority to ‘High’ after call establishment
  - No application configuration to prevent machine from becoming a super node. Possible by limiting per-process bandwidth
  - Code obfuscation, runtime decryption
  - Login server and super nodes, not strictly peer-to-peer
  - STUN and TURN equivalent functionality
  - Combination of hashing and controlled flooding
  - Multiple paths for ‘in-time’ switching in case of failures
  - Search falls back to login server