SIP and VoIP

Skype
an example VoIP client

SIP / VoIP: what are these?

- Voice over IP (VoIP)
- Session Initiation Protocol (SIP)
  - Control channel
    - Known in telephone world as signaling channel
    - Does call setup:
      - locates other end point
      - Determines if it's available
      - Asks endpoint to alert called party
      - Passes status back to caller, ...
    - Needed even in pure IP world, e.g., to interfaces with PSTN (Public Switched Telephone Network)
  - Other control channels exist: e.g., H.323 and Skype
History of signaling channels

- Telephone signaling in the past: „in-band“ pulses or tones were sent over the same circuit as used for carrying the voice traffic for call
  - „Blue boxes“ – telephone fraud devices – worked by simulating some of the control tones used to setup free calls

- Solution: „out-of-band“ signaling
  - Separate data network, known today as CCIS (Common Channel Interoffice Signaling)
  - Advantages
    - More efficient
    - Allows creation of fancier services

VoIP challenges

- What address to use? DNS name, IP address?
  - Many endpoints do not have stable, easily-memorized domain names
  - IP addresses change frequently, e.g., dialup, hotspot users
  - NAT: many endpoints have only a few IP addresses
  - What about unreachable hosts?

- Firewalls?

- PSTN interconnection?
  - Who pays?
  - Mapping between „phone number“ and IP address?
  - Business arrangements between companies
  - What about fancy phone features?
Basic SIP architecture

- SIP endpoints speak IP
- Ideally: end-to-end conversations (SIP-to-SIP)
- Yet, each node can use a SIP proxy for call setup

Example: simple SIP call
Example: simple SIP call (2.)

- Alice uses VoIP provider 1 (VP1) as proxy
- Bob uses VoIP provider 2 (VP2) as proxy
- Alice sends SIP URI to VP1 via TCP
- VP1 determines that URI points to VP2
  - Relays call setup request to VP2 via TCP
- VP2 tells Bob about call via TCP
  - If Bob wants to he can accept it
- Notification is send back to Alice via VP1
- Alice establishes UDP data connection to Bob for voice call

SIP details

- SIP URIs
  - URIs are converted by means of DNS magic (NAPTR records) to an IP address
    - (Not important how, just that it is)
  - tel: URIs are used for ordinary phones
- Firewalls and NATs
  - If Alice and/or Bob are behind firewalls or NATs direct end-to-end data connections may not be possible
  - Data traffic can be relayed through the proxy for one or both parties
- Multiple proxies
  - Sometimes necessary
  - How to ensure trust?
Attacking SIP

Information at risk

- Voice content itself
  - Main concern: confidentiality
    - VoIP easier to wiretap than traditional phone service...
    - Only endpoints should see that info; use encryption for proxies
    - Relatively hard to spoof VoIP in real-time ⇒ authenticity not that much of a concern

- Caller and called party
  - Of great interest (see HP case)
  - Useful even after the call
  - Must be kept confidential – but proxies need it to route call
  - Must be authentic, or call can be misrouted maliciously

Billing information

- Derived in part from caller / called party information
- May use other information from call routing process
- Must be confidential – but there is no need for other parties to see any of it
- Integrity failures can lead to billing errors, in either direction
- (Can be a major privacy concern after the fact, e.g. HP)
Attacking via eavesdropping

- On link
  - e.g., listening at WiFi hotspot
  - ...

- On call
  - Simplest approach:
    - Listen on some link, e.g., their access link
  - What for mobile ones? Harder – they could be from anywhere
  - At proxy? What about encryption?
  - At provider? What if VoIP provider is in unfriendly country?

Attacking: other

- Registration hijacking: diverting calls
  - Attacker can try to register with VP2 as Bob
  - If attacker succeeds, all calls destined for Bob are routed to the attacker
  - Man in the middle attack...

- Registration hijacking: tearing down sessions
  - Violates availability

- Abusing DNS
  - Call routing is partially controlled by DNS
  - Corrupt DNS answers?
    - Create fake DNS entries and reroute call via interception station
Caller/Called party information

- Easier: proxies do not move 😊 via link eavesdropping and DNS attacks
- VoIP providers are high-value targets
  - Hack the proxy
  - Conventional phone switches have been hacked
  - SIP switch speaks a much more complex protocol than PSTN switch
- IP address
  - Hard to hide
  - Legitimate recipient sees sender address, leaks location data
  - Rerouting via proxy can thus be a privacy feature

Billing system

- Similar in nature to old-style ones
- SIP billing systems are more likely to be Internet connected
  - Need strong defenses and firewalls
  - ...
Protecting SIP

- Use crypto to guard against eavesdropping

Alice to VP1
- Alice has trust relationship with her proxy
- Authentication is relatively easy, e.g., use TLS to protect TCP session from Alice to proxy
- Alice must verify VP1's certificate
- Alice can use passwords or client-side certificates to authenticate herself
- Why not IPsec?
  - Tough to protect a specific service
  - But good for organizational SIP gateway

Protecting SIP (2.)

- Proxy to proxy traffic
  - VP1 may not have a trust relationship with VP2
  - Use PKI or Web of trust
  - Use appropriate security protocol, e.g., TLS

- Proxy to Bob
  - See Alice to proxy

- End-to-end signaling traffic
  - Some information must be secure end-to-end, e.g., Bob needs to know, authoritatively, that it is Alice who has called him
  - Digitally sign the data (e.g., S/MIME) but no encryption (Intermediate nodes may need to see this!)
Key management for VoIP

- How to establish a shared key for voice traffic encryption?
  - Alice uses S/MIME to send Bob an encrypted traffic key
  - But – how does Alice get Bob’s certificate?
    - No general PKI for SIP users
    - True end-to-end confidentiality can only happen by prearrangement...

Complex scenarios / features

- Complexity causes problems
  - In this case: complex trust patterns!
- Scenario A:
  - Alice tries to call Carol – reaches Bob, Carol’s secretary
  - Bob decides the call is worthy of Carol’s attention – wants to transfer the call to Carol
  - Bob’s phone sends Alice’s phone a message saying „Call Carol, you are authorized“
  - Carol’s phone has to verify that Bob authorized it
Complex scenarios (2.)

- Scenario A: solution 1
  - Bob uses authenticated identity body (AIB) with his name and the time
  - He sends that to Alice along with Carol's SIP URI
  - Alice presents the AIB to Carol
  - ?

- Scenario A: problem?
  - Nothing linking the AIB to referral
  - Alice can give the AIB to someone else
  - Good: timestamp defends against replay

Complex scenarios (3.)

- Scenario A: solution
  - AIB sent by Bob needs to include Alice's identity
  - Carol's phone needs to check the certificate used in Alice's call setup message, to verify that it is really from Alice
  - Alice's identity in AIB must match identity in certificate
Complex scenarios (4.)

- Scenario B:
  - Suppose SIP call is relayed to the PSTN
  - Where does the CallerID information came from?
  - Can it be spoofed?
- Phone network design
  - Based on trust – only „real“ telephone companies had phone switches
  - No authentication was done on information from other switches, including CallerID
  - Today: anyone can run a phone switch....

CallerID and VoIP

- Run Asterisk, an open source PBX program, on some machine
- Get a leased line to a VoIP-to-PSTN gateway company
- Configure Asterisk to send whatever information you want
- This is happening now, e.g.,
State of practice

- Most vendors do not implement fancy crypto
- VoIP is thus not as secure as it could be
  (But note Skype does do a lot of crypto)
- Beyond that, SIP phones tend to boot themselves over the network – is that connection secure?

Skype a P2P VoIP application
P2P: what is it?

- 1999 Napster 1. widely used P2P application

Definition of Peer-to-peer (or P2P)

- Network that relies primarily on computing power and bandwidth of participants rather than on a small number of servers
- No notion of clients or servers (client-server model), only equal peer nodes (these function simultaneously as "clients" and "servers" to other nodes)
Lots of applications

- P2P-File download
  - Napster, Gnutella, KaZaa, eDonkey,…
- P2P-Communication
  - VoIP, Skype, Messaging, …
- P2P-Video-on-Demand
- P2P-Computation
  - seti@home
- P2P-Streaming
  - PPLive, ESM, …
- P2P-Gaming
- …

Why is P2P so successful?

- Scalable – it is all about sharing resources
  - No need to provision servers or bandwidth
  - Each user brings its own resource
  - E.g., resistant to flash crowds
    (a large number of users all arriving at the same time)

Resources could be:
- Files to share;
- Upload bandwidth;
- Disk storage;…
Why is P2P so successful? (2.)

- Cheap - No infrastructure needed
- Everybody can bring its own content (at no cost)
  - Homemade content
  - Ethnic content
  - Illegal content
  - But also legal content
  - ...
- High availability – Content accessible most of the time

P2P-Overlay

- Build network at application layer
- Forward packet at the application layer
- Network is *virtual*
  - Underlying physical graph is transparent to the user
  - Edges are TCP connection or an entry of a neighboring node’s IP address
- Network has to be continuously maintained (e.g., check if nodes are still alive)
The P2P enabling technologies

- Unstructured p2p-overlays
  - Generally random overlay
  - Used for content download, telephony, streaming
- Structured p2p-overlays
  - Distributed Hash Tables (DHTs)
  - Used for node localization, content download, streaming
P2P techniques

- Unstructured p2p-overlays
  - Generally random overlay
  - Used for content download, telephony, streaming
- Structured p2p-overlays
  - Distributed Hash Tables (DHTs)
  - Used for node localization, content download, streaming

Skype overlay

- Protocol not fully understood
  - Proprietary protocol
  - Content and control messages are encrypted
- Protocol reuses concepts of the FastTrack overlay used by KaZaA
- Builds upon an unstructured overlay
  - Combines
    - Distributed index servers
    - A flat unstructured network between index servers
  - Two tier hierarchy
Skype overlay (cont’d)

- Super nodes (SN)
  - Connect to each other
  - Flat unstructured overlay (similar to Gnutella)

- Ordinary nodes (ON)
  - Connect to super nodes that act as a directory server
    (similar to index server in Napster, Gnutella clients)

- Skype login server
  - Central component
  - Stores and verifies usernames and passwords
  - Stores the buddy list

Skype Overlay (cont’d)

[Diagram showing the Skype overlay network with nodes labeled SN and ON, connected by lines indicating neighbor relationships, and a Skype login server linked by dashed lines for message exchange during login for authentication.]
How is overlay constructed?

- How to connect? == Find Super node
  - Use Super Node list implemented as host cache
  - Needs at least one valid entry!
  - Up to 200 entries
  - Some Super Nodes IP-addresses are hard-coded
    - Super Nodes provided by Skype

- Login:
  - Contact login server and authenticate
  - Advertise your presence to other peers: contact
    - Super Node
    - Your buddies (through Super Node), and notify presence

Super Nodes – Index servers

- Index servers
  - I.e. index of locally connected Skype users (and their IP addresses)
  - If buddy is not found in local index of Super Node
    - Spread search to neighboring Super Nodes
    - Not clear how this is implemented (flood the request similar to Gnutella?)

- Relay nodes
  - Enables NAT traversals
  - Avoid congested or faulty paths
Super Nodes – Relay nodes

- Alice would like to call Bob (or inversely)

Contact Relay Node

Skype relay node

Call
Super Node election

- When does an ordinary node become super node?
  - High bandwidth, public IP address, details unclear
  - Highly dynamic
    - Super Node Churn, Short Super Node session time

![Graphs showing Churn and Session time](image)

Super Node election

- A world map of Skype Super Nodes
Skype's use of ports

- One TCP and one UDP listening port
  - As configured in connection dialog box
  - Or randomly chooses one upon installation
  - Default 80 (HTTP), 443 (HTTPS)

---

Skype features

- Encryption
  - 1536 to 2048 bit RSA
    - User public key is certified by login server during login
  - AES (Rijndel) to protect sensitive information
    - 256-bit encryption: $1.1 \times 10^{77}$ possible keys
  - RSA to negotiate symmetric AES keys

- NAT and firewall
  - Conjecture use of STUN (Simple Traversal of UDP through NATs) and TURN (Traversal Using Relay NAT) to determine the type of NAT and firewall
  - Information is stored in the Windows registry
  - Use TCP to bypass UDP-restricted NAT/firewall
Skype – Functional summary

- VoIP has other requirements than file download
  - Delay
  - Jitter
- Skype network seems to handle these well in spite of
  - High node churn
- Protocol not fully understood

Skype analysis
„Silver Needle in the Skype“

Philippe Biondi and Fabrice Desclaux
BlackHat Europe, March 2006
Course overview

- Introduction
  - Attacks and threats, cryptography overview
  - Authentication (Kerberos, SSL)

- Applications
  - Web, email, ssh

- Lower layer network security
  - IPsec, firewalls, wireless

- Monitoring / information gathering
  - Intrusion detection, network scans

- Availability
  - Worms, denial of service, network infrastructure