AGDER UNIVERSITY COLLEGE
Faculty of Engineering and Science

# SIP based IP-Telephony Network Security Analysis

Master Thesis in Information and
Communication Technology

Dag Ove Valsgaard

Kristiansand, June 2004

## Abstract

This thesis evaluates the SIP Protocol implementation used in the Voice over IP (VoIP) solution at the fibre/DSL network of Èlla Kommunikasjon AS. The evaluation focuses on security in the telephony service, and is performed from the perspective of an attacker trying to find weaknesses in the network.

For each type of attempt by the malicious attacker, we examined the security level and possible solutions to flaws in the system.

The conclusion of this analysis is that the VoIP service is exploitable, and that serious improvements are needed to achieve a satisfying level of security for the system.

# Preface

This report presents the master thesis concluding my studies to Master of Science in information and communication technology at Agder University College (AUC), Faculty of Engineering and Science in Grimstad.

The aim and object of such a thesis work, is to gain experience in independent research work, and achieve a deeper understanding of a key subject area. The total workload is about 30 credits.

The thesis has been conducted at Èlla Kommunikasjon AS, a company in the Agder Energi group. The assignment origins from the project "LOS IP-telefoni til Sørlandet" and is handled within the product development and telecom department.

I would like to thank my supervisor, engineer Tor Setane at Èlla Kommunikasjon AS, for valuable help and inspiration. I would also like to thank Frode Sorensen at Agder University College for handling the primary contact back with campus in Grimstad. Additionally I want to thank my family for patience and support during my work.

Kristiansand, 28th of May 2004.

Dag Ove Valsgaard

# Table of Contents

**ABSTRACT** ..... **2**

**PREFACE** ..... **3**

**TABLE OF CONTENTS** ..... **4**

**1 INTRODUCTION** ..... **6**

1.1 BACKGROUND THEME AND PROBLEM ..... 6
1.2 STATUS ..... 6
1.3 GOAL OF THE WORK ..... 6
1.4 OUTLINE OF THE REPORT ..... 7

**2 THEORETICAL BACKGROUND** ..... **8**

2.1 SESSION INITIATION PROTOCOL (SIP) ..... 8
*2.1.1 History* ..... *8*
*2.1.2 Premise* ..... *8*
*2.1.3 Operation Breakdown* ..... *9*
*2.1.4 Type of Operation* ..... *9*
*2.1.5 Performing Calls* ..... *10*
*2.1.6 SIP addressing:* ..... *11*
*2.1.7 Example: SIP Call Flow* ..... *11*
*2.1.8 Characteristics* ..... *13*
*2.1.9 Comparing SIP to H.323* ..... *18*
*2.1.10 Connectivity* ..... *19*
*2.1.11 Main Advantages* ..... *20*
*2.1.12 Main Drawbacks:* ..... *21*
*2.1.13 Interfacing* ..... *21*
*2.1.14 Real Time Transport Protocol* ..... *22*
*2.1.15 Quality of Service (QoS)* ..... *23*
*2.1.16 Encryption* ..... *23*
2.2 NETWORK TECHNOLOGY ..... 24
*2.2.1 OSI-model and IP packets* ..... *24*
*2.2.2 IP-addresses* ..... *27*
*2.2.3 Subnets* ..... *28*
*2.2.4 Routing, ARP and IP* ..... *30*
*2.2.5 Services in an IP-network (DHCP and DNS)* ..... *33*
*2.2.6 Virtual local area network (VLAN)* ..... *36*
*2.2.7 Point to Point Protocol over Ethernet (PPPoE)* ..... *39*
*2.2.8 Transport Layer Security (TLS)* ..... *40*
*2.2.9 Internet Protocol Security (IPSec)* ..... *42*
*2.2.10 Secure Shell (SSH)* ..... *48*
2.3 CRYPTOGRAPHY ..... 51
*2.3.1 Introduction to Cryptography* ..... *51*
*2.3.2 Ciphers and Keys* ..... *52*
*2.3.3 Message Digests* ..... *53*
*2.3.4 Certificates and Key Management* ..... *54*
*2.3.5 Summary* ..... *55*

**3 SCENARIO** ..... **56**

3.1 OVERVIEW ..... 56
*3.1.1 Network* ..... *56*
*3.1.2 IP-telephony traffic* ..... *57*
3.2 EQUIPMENT ..... 60
*3.2.1 Networking* ..... *61*
*3.2.2 IP-Telephony Servers* ..... *65*
*3.2.3 IP-Telephony Clients* ..... *68*

**4 SECURITY ANALYSIS** ..... **76**

4.1 SECURITY THREATS ..... 76
*4.1.1 Passive threats* ..... *77*
*4.1.2 Active threats* ..... *77*
*4.1.3 Security Services* ..... *78*
4.2 TRAFFIC ANALYSIS ..... 78
*4.2.1 SIP* ..... *79*
*4.2.2 TELNET* ..... *81*
*4.2.3 VLAN* ..... *82*
*4.2.4 An organized assault* ..... *82*
*4.2.5 Summary* ..... *87*
4.3 SERVER ANALYSIS ..... 87
*4.3.1 Port scanning the Server* ..... *87*
*4.3.2 Analysis of Port Scan Results* ..... *90*
*4.3.3 Organized Assaults* ..... *93*
4.4 UNAUTHORIZED LISTENING TO CALLS ..... 98
4.5 SUMMARY ..... 100

**DISCUSSION ..... 101**

UNAUTHORIZED PHONE CALLS ..... 101
DENIAL OF SERVICE ..... 103
TOTAL SERVER CONTROL ..... 105
LISTENING TO PHONE CALLS ..... 105
ADDITIONAL SECURITY MATTERS ..... 109
SECURITY OBLIGATIONS ..... 111

**CONCLUSION ..... 113**

**LIST OF FIGURES AND TABLES ..... 114**

**GLOSSARY ..... 116**

**REFERENCES ..... 117**

# 1 Introduction

## 1.1 Background theme and problem

Èlla Kommunikasjon is a company in the Agder Energi group, and is a supplier of broadband services as Internet (ISP), broadband television, server services, and radio communication as well as regular telephony services. These products are mainly delivered to a limited geographical area (Agder). The customer groups are both B2B (Business to Business) and B2C (Business to Consumer).

Since the broadband technology now has made way for better bandwidth, Èlla Kommunikasjon has seen the possibilities to expand the use of their fibre network to supply more services than what we see today. One of the main areas of interest is telephony based on the fibre and DSL network for bearing, so-called IP-telephony or VoIP. This broadband network has been put together mainly of components from the suppliers Hewlet Packard and Allied Telesyn.

In corporation between the sales and service department in Agder Energi (LOS), and their own product development department, Èlla Kommunikasjon AS has decided to offer IP-telephony as a service for the consumer market. This should be done by summer 2004. This deployment will work as a pilot and experience case for later deployment in the corporate market.

For this reason a project group has been established within Èlla Kommunikasjon. This group has the goal to make the service available to the customers of LOS. The writer of this report is a part of the group, and has been a member of Èlla Kommunikasjon staff since early 2003.

The background for this report is a wish to locate and handle security issues that can be found within a pilot system running in small scale today. This pilot is the beginning of the total VoIP solution for the consumer market, and will be the one deployed summer of 2004.

## 1.2 Status

As of today, the SIP protocol deployments are growing in extent. We can read more about IP-telephony in the newspaper than before (i.e. Telio [1]), and an increasingly number of people has an opinion of what VoIP is.

ISP's have become focused on running more services than only internet access in their networks. Triple play (Internet access, telephony and Television in a shared IP-environment) is a hot potato in the Information and Communication world this day. It can represent cost reduction by converging different service networks into one.

Customers have therefore indirectly demanded the coming of VoIP, and some ISP's and telecom companies have started deploying it.

When deploying a new service large scale, security needs to be handled correctly. There are only a few examples of how to do this, and to get an overview of which exploits and threats that exists when offering such a service; research as of today on this matters are inadequate.

## 1.3 Goal of the work

The goal of this thesis is to examine how secure the VoIP solution running with Èlla Kommunikasjon AS is today.

[1] Telio is an Norwegian company offering all broadband clients independent of supplier, free VoIP with access to the PSTN. More can be found on telio.no

To achieve this, we will focus on how a malicious attacker can take advantage of weaknesses in the system. This based on the deployment of the VoIP server SER (SIP Router Express), in the existing fibre and DSL network.

We will play the role of the attacker, and try to get access to the system, enabling us to do unauthorised phone calls, unauthorised listening to calls, make the system collapse (Denial of Service) and get total control of the entire system; play the role of an administrator.

The experience gained by analysis will suggest improvements to the system, hopefully making it almost impossible to exploit.

## 1.4 Outline of the report

This chapter serves as an introduction to the report. A brief synopsis of the remaining parts follows.

The first part gives a theoretical background for the thesis [chapter 2]. It represents the theory behind all services, protocols and applications running in the network and in the VoIP system. It gives firm knowledge to the Session Initiated Protocol (SIP), and will give the reader a good understanding of networking and cryptography.

The second part presents the security analysis of the VoIP solution [chapters 3 and 4]. First section gives an extensive overview of the network, devices and services. Then the second part of the analysis gives summaries of the attacks tried in the experiment.

Last section is divided in two [chapter 5 and 6]. First a discussion based on what are discovered in the security analysis. In this discussion we try to find how the system can be strengthened when it comes to security. This is done by discussing the results of the analysis with origin in successfully performed attacks. When all parts of the analysis are discussed, we find a chapter that handles what obligations such systems has to their users and the law of the country it is deployed in (In our case; Norway).
It finally results in a chapter of conclusion. This chapter is short, and delivers the essence of what we have found during analysis and discussion. It will give an understanding on how to secure this particular VoIP deployment, and also an indication on how to secure other similar solutions.

In addition to this, the end of the report includes a listing of figures and tables, glossary and references.

# 2 Theoretical Background

This chapter will give the reader a platform for following discussions in later chapters. It creates an extensive basis for understanding the problems related to this report. The chapter is split into different subsections, and the reader might leave sections where he has a thorough knowledge.

## 2.1 Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) is an application-layer control protocol that can establish, modify and terminate multimedia sessions or calls. These multimedia sessions include multimedia conferences, distance learning, Internet telephony and similar applications. SIP can invite both persons and "robots", such as a media storage service. SIP can invite parties to both unicast and multicast sessions; the initiator does not necessarily have to be a member of the session to which it is inviting. Media and participants can be added to an existing session. SIP can be used to initiate sessions as well as invite members to sessions that have been advertised and established by other means. Sessions can be advertised using multicast protocols such as SAP, electronic mail, news groups, web pages or directories (LDAP), among others.

### 2.1.1 History

SIP has its origins in late 1996 as a component of the "Mbone" set of utilities and protocols. The Mbone, or multicast backbone, was an experimental multicast network overlaid on top of the public Internet. It was used for distribution of multimedia content, including talks and seminars, broadcasts of space shuttle launches, and IETF meetings. One of its essential components was a mechanism for inviting users to listen in on an ongoing or future multimedia session on the Internet. Basically - a session initiation protocol.
Since its approval in early 1999 as an official standard, the Session Initiation Protocol has gained tremendous market acceptance for signalling communications services on the Internet.
Despite its historical strengths, SIP saw relatively slow progress throughout 1996 and 1997. That's about when interest in Internet telephony began to take off. People began to see SIP as a technology that would also work for VoIP, not just Mbone sessions. The result was an intensified effort towards completing the specification in late 1998, and completion by the end of the year. It received official approval as an RFC (Request for Comments, the official term for an IETF specification) in February and issuance of an RFC number, 3261 Session Initiated Protocol [1], in March.
From there, industry acceptance of SIP grew exponentially. Its scalability, extensibility, and - most important - flexibility appealed to service providers and vendors who had needs that a vertically integrated protocol, such as H.323, could not address. Among service providers MCI (particularly MCI's Henry Sinnreich, regarded as the "Pope" of SIP) led the evangelical charge. Throughout 1999 and into 2000, it saw adoption by most major vendors, and announcements of networks by service providers.

### 2.1.2 Premise

As an Mbone tool (and as a product of the IETF), SIP was designed with certain assumptions in mind. First was scalability:
Since users could reside anywhere on the Internet, the protocol needed to work wide-area from day one. Users could be invited to lots of sessions, so the protocol needed to scale in both directions. A second assumption was component reuse: Rather than inventing new protocol tools, those already developed within the IETF would be used. That included things like MIME, URLs, and SDP (already used for other protocols, such as SAP). This resulted in a protocol that integrated well with other IP applications (such as web and e-mail).
Interoperability was another key goal, although not one specific to SIP. Interoperability is at the heart of IETF's process and operation, as a forum attended by implementers and operational experts who actually build and deploy the technologies they design. To these practical-minded

standardizes, the KISS (Keep It Simple Stupid) principle was the best way to help ensure correctness and interoperability.

### 2.1.3 Operation Breakdown

- establishing connection
- adding parties
- changing session parameters
- terminating multimedia communications

- User location: determination of the end system
- User capabilities: determination of the media and parameters
- User availability: determination of the willingness for communications
- Call setup: "ringing", setting call parameters at called and calling party

As the name implies, the session initiation protocol (SIP) is about initiation of interactive communications sessions between users. SIP also handles termination and modifications of sessions as well. SIP actually doesn't define what a "session" is; this is described by content carried in SIP messages. Most of SIP is about the initiation part, since this is really the most difficult aspect. "Initiating a session" requires determining where the user to be contacted is actually residing at a particular moment. A user might have a PC at work, a PC at home, and an IP desk phone in the lab. A call for that user might need to ring all phones at once. Furthermore, the user might be mobile; one day at work, and the next day visiting a university. This dynamic location information needs to be taken into account in order to find the user.

Once the user to be called has been located, SIP can perform its second main function - delivering a description of the session that the user is being invited to. As mentioned, SIP itself does not know about the details of the session. What SIP does do is convey information about the protocol used to describe the session. SIP does this through the use of multipurpose internet mail extensions (MIME), widely used in web and e-mail services to describe content (HTML, audio, video, etc.). The most common protocol used to describe sessions is the session description protocol (SDP), described in RFC2327. SIP can also be used to negotiate a common format for describing sessions, so that other things besides SDP can be used.

Once the user has been located and the session description delivered, SIP is used to convey the response to the session initiation (accept, reject, etc.). If accepted, the session is now active. SIP can be used to modify the session as well. Doing so is easy - the originator simply re-initiates the session, sending the same message as the original, but with a new session description. For this reason, modification of sessions (which includes things like adding and removing audio streams, adding video, changing codec's, hold and mute) are easily supported with SIP, so long as the session description protocol can support them (SDP supports all of the above).
Finally, SIP can be used to terminate the session (i.e., hang up)

### 2.1.4 Type of Operation

SIP is designed as part of the overall IETF multimedia data and control architecture. This multimedia data and control architecture is currently
incorporating protocols such as

- RTP the real-time transport protocol for transporting real-time data and providing QOS feedback,
- RTSP the real-time streaming protocol for controlling delivery of streaming media,
- SAP the session announcement protocol for advertising multimedia sessions via multicast, and
- SDP the session description protocol for describing multimedia sessions.

The functionality and operation of SIP does not depend on any of these protocols!!

SIP is based on the request-response paradigm. To initiate a session, the caller (known as the User Agent Client, or UAC) sends a request (called an INVITE); addressed to the person the caller wants to talk to. In SIP, addresses are URLs. SIP defines a URL format that is very similar to the popular mailto URL. If the user's e-mail address is me@some-domain.com, their SIP URL would be sip:me@somedomain.com. This message is not sent directly to the called party, but rather to an entity known as a proxy server. The proxy server is responsible for routing and delivering messages to the called party. The called party then sends a response, accepting or rejecting the invitation, which is forwarded back through the same set of proxies, in reverse order.
A proxy can receive a single INVITE request, and send out more than one INVITE request to different addresses. This feature, aptly called "forking," allows a session initiation attempt to reach multiple locations, in the hopes of finding the desired user at one of them. A close analogy is the home phone line service, where all phones in the home ring at once.

### 2.1.5 Performing Calls

This section explains the basic protocol functionality and operation. Callers and callees are identified by SIP addresses. When making a SIP call, a caller first locates the appropriate server and then sends a SIP request. The most common SIP operation is the invitation. Instead of directly reaching the intended callee, a SIP request may be redirected or may trigger a chain of new SIP requests by proxies. Users can register their location(s) with SIP servers.

Assuming the caller (me@somedomain.com) wishes to place a call to you@someotherdomain.com. "Me" sends his SIP INVITE message to the proxy for somedomain.com (Step 1).

This proxy then forwards the request out to someotherdomain, where it reaches the someotherdomain.com server (Step 2).

This server is actually not a proxy, but a similar device called a redirect server. Instead of forwarding calls, a redirect server asks the requestor to contact the next server directly. The someotherdomain.com server looks up "you" in its database, and determines that today; "You" is on sabbatical to foo.com. It therefore sends a special response, called a redirect, to the somedomain.com proxy, instructing it to instead try you@foo.com (Step 3).

The somedomain proxy then acts on this response, which means it directly tries to contact you@foo.com. So, it sends the INVITE to the foo.com server (Step 4).

This server consults its database (Step 5),

and learns (Step 6) that "You" is actually in sales.

So, it constructs a new URL, you@sales. foo.com, and sends the INVITE to the sales.foo.com proxy (Step 7).

The proxy for the sales department then needs to forward the INVITE to the PC where "You" is currently sitting. For getting out which PC "You" is currently using, SIP defines another request, called REGISTER, which is used to inform a proxy of an address binding. In this case, when "You" turned on his SIP client on his PC, the client would register the binding sip:you@sales.engineering.com to sip:you@mypc.sales.foo.com. This would allow the proxy to know that "You" is actually at mypc, a specific host on the network. The bindings registered through SIP are periodically refreshed, so that if the PC crashes, the binding is eventually removed.
The sales.foo.com proxy consults this registration database, and forwards the INVITE to you@mypc.sales.foo.com (Step 8).

This INVITE then reaches "You" at his PC. "You" can then respond to it (thus the request-response model). SIP provides many responses, and these include acceptance, rejection,

redirection, busy, and so on. The response is forwarded back through the proxies to the original caller (Steps 9, 10, 11, 12). An acknowledgement is sent (another type of request, called ACK) in Step 13, and the session is established. Media can then flow (Step 14).

### 2.1.6 SIP addressing:

A SIP URL follows the guidelines of RFC 2396 [2] and has the syntax shown in [Table 1 SIP Addressing]. It is described using Augmented Backus-Naur Form. Note that reserved characters have to be escaped and that the "set of characters" reserved within any given URI component is defined by that component. In general, a character is reserved if the semantics of the URI changes if the character is replaced with its escaped US-ASCII encoding

SIP URLs are used within SIP messages to indicate the originator (From), current destination (Request-URI) and final recipient (To) of a SIP request, and to specify redirection addresses (Contact). A SIP URL can also be embedded in web pages or other hyperlinks to indicate that a particular user or service can be called via SIP. When used as a hyperlink, the SIP URL indicates the use of the INVITE method. The SIP URL scheme is defined to allow setting SIP request-header fields and the SIP message-body. This corresponds to the use of mailto: URLs. It makes it possible, for example, to specify the subject, urgency or media types of calls initiated through a web page or as part of an email message.
Some examples for use and default values of URL components for SIP headers:

| |
|---|
| sip:j.doe@big.com |
| sip:j.doe:secret@big.com;transport=tcp |
| sip:j.doe@big.com?subject=project |
| sip:+1-212-555-1212:1234@gateway.com;user=phone |
| sip:1212@gateway.com |
| sip:alice@10.1.2.3 |
| sip:alice@example.com |
| sip:alice%40example.com@gateway.com |
| sip:alice@registrar.com;method=REGISTER |

***Table 1 SIP Addressing***

SIP URLs are case-insensitive, so that for example the two URLs sip:j.doe@example.com and SIP:J.Doe@Example.com are equivalent. All URL parameters are included when comparing SIP URLs for equality. The Request-URI is a SIP URL or a general URI. It indicates the user or service to which this request is being addressed. Unlike the To field, the Request-URI MAY be re-written by proxies

### 2.1.7 Example: SIP Call Flow

### 2.1.7.1 Basic Call Flow

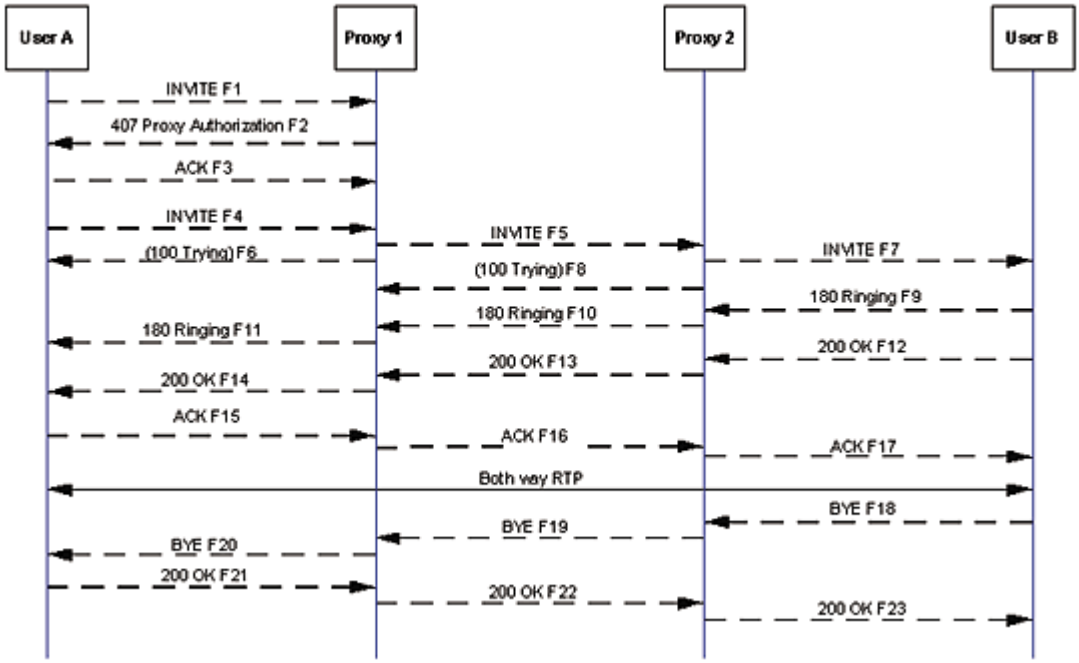


***Figure 1 SIP Basic Call Flow***

In Figure 1 SIP Basic Call Flow, Caller A completes a call to User B using two proxies: Proxy 1 and Proxy 2. The initial INVITE (F1) does not contain the Authorization credentials that Proxy 1 requires, so an Authorization response is sent containing the challenge information. A new INVITE (F4) is then sent containing the correct credentials and the call proceeds. The call terminates when User B disconnects by initiating a BYE message.

F1 INVITE A -> Proxy 1

The call begins, as always, with an INVITE message that contains information on caller and called party as well as the session description request (2nd part).

```
INVITE sip:UserB@ss1.wcom.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: BigGuy
To: LittleGuy
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Contact: BigGuy
Content-Type: application/sdp
Content-Length: 147

v=0
o=UserA 2890844526 2890844526 IN IP4 here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

F2 407 Proxy Authorization Required Proxy 1 -> User A

SIP always works in a request-response mode and in this example Proxy 1 challenges Caller A for authentication

SIP/2.0 407 Proxy Authorization Required
Via: SIP/2.0/UDP here.com:5060
From: BigGuy
To: LittleGuy
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Proxy-Authenticate: Digest realm="MCI WorldCom SIP",
domain="wcom.com", nonce="wf84f1ceczx41ae6cbe5aea9c8e88d359",
opaque="", stale="FALSE", algorithm="MD5"
Content-Length: 0

As we move further down the call flow, the actual voice call begins, using Realtime Transport Protocol (RTP) to move the voice stream.

F17 ACK Proxy 2 -> B
ACK sip: UserB@there.com SIP/2.0
Via: SIP/2.0/UDP ss2.wcom.com:5060
Via: SIP/2.0/UDP ss1.wcom.com:5060
Via: SIP/2.0/UDP here.com:5060
From: BigGuy
To: LittleGuy ;tag=314159
Call-ID: 12345601@here.com
CSeq: 1 ACK
Content-Length: 0
Calls are then terminated with a BYE request to the caller.

F18 BYE User B -> Proxy 2
BYE sip: UserA@ss2.wcom.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
Route: ,
From: LittleGuy ;tag=314159
To: BigGuy
Call-ID: 12345601@here.com
CSeq: 1 BYE
Content-Length: 0

***Table 2 SIP Basic Call Flow***

### 2.1.8 Characteristics

In this chapter you can learn more about layers, Messaging and ABNF.

#### *2.1.8.1 Layers*

SIP makes minimal assumptions about the underlying transport and network-layer protocols. The lower-layer can provide either a packet or a byte stream service, with reliable or unreliable service. In an Internet context, SIP is able to utilize both and TCP as transport protocols, among others. UDP allows the application to more carefully control the timing of messages and their retransmission, to perform parallel searches without requiring TCP connection state for each outstanding request, and to use multicast. Routers can more readily snoop SIP UDP packets. TCP allows easier passage through existing firewalls. Possibly lower layers:

UDP
TCP
ATM AAL5
IPX
frame relay
X.25

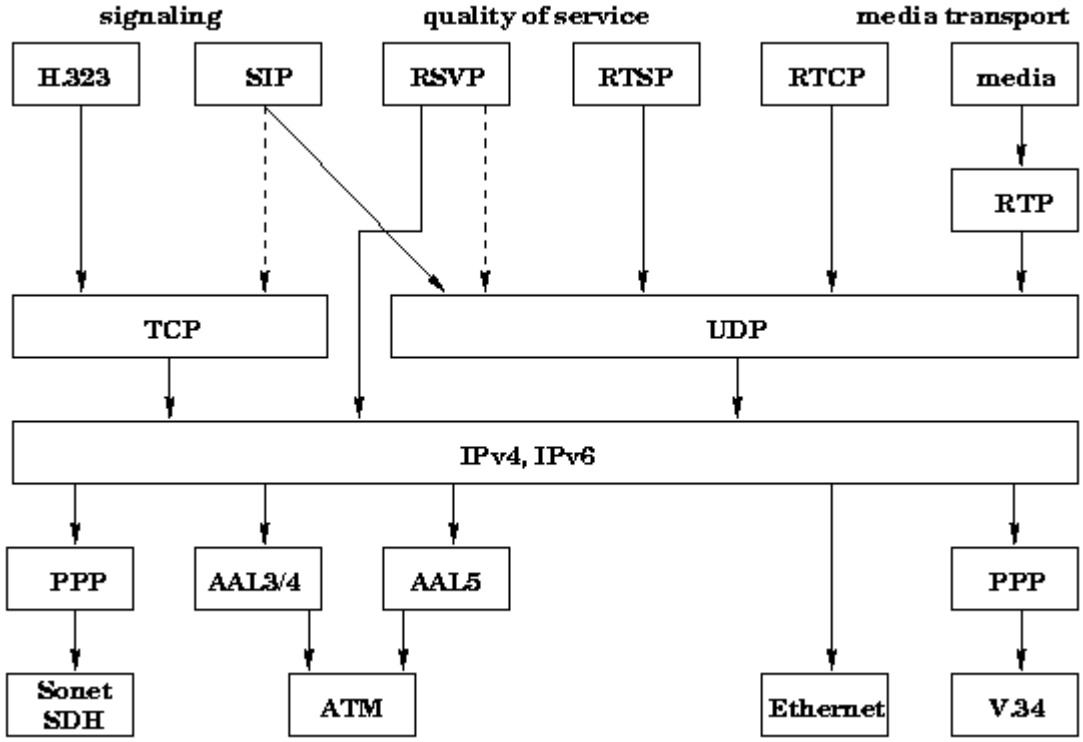


***Figure 2 SIP Layers***

### *2.1.8.2 Messaging*

SIP is patterned after HTTP in many ways. HTTP is also request-response. SIP borrows much of the syntax and semantics from HTTP. The textual message formatting, usage of headers, MIME support, and many headers are identical. An http expert looking at a SIP message would have difficulty distinguishing them.

The 1500 bytes accommodates encapsulation within the "typical" Ethernet MTU without IP fragmentation. The next lower common MTU values are 1006 bytes for SLIP and 296 for low-delay PPP (RFC 1191). Thus, another reasonable value would be a message size of 950 bytes, to accommodate packet headers within the SLIP MTU without fragmentation.

### *2.1.8.3 Text based*

If you are missing explanations on setting Bits in specific Bytes to gain a special function you can seek for a very long time! SIP is not using binary mode messaging for its work.
SIP is text-based, using ISO 10646 in UTF-8 encoding throughout. This allows easy implementation in languages such as Java, Tcl and Perl, allows easy debugging, and most importantly, makes SIP flexible and extensible. As SIP is used for initiating multimedia conferences rather than delivering media data, it is believed that the additional overhead of using a text-based protocol is not significant. Except for the above difference in character sets, much of the message syntax is and header fields are identical to HTTP/1.1, but cannot be seen as an

extension to HTTP!

### *2.1.8.4 ABNF*

A SIP message is either a request from a client to a server, or a response from a server to a client. SIP header fields are similar to HTTP header fields in both syntax and semantics. In particular, SIP header fields follow the syntax for message-header as described below.

SIP-message = Request | Response

#### 2.1.8.4.1 Request

Request = Request-Line *( general-header | request-header | entity-header ) CRLF [ message-body ]

Request-Line = Method SP Request-URI SP SIP-Version CRLF
Method = "INVITE" | "ACK" | "OPTIONS" | "BYE" | "CANCEL" | "REGISTER"

INVITE
The INVITE method indicates that the user or service is being invited to participate in a session. The message body contains a description of the session to which the callee is being invited. For two-party calls, the caller indicates the type of media it is able to receive and possibly the media it is willing to send as well as their parameters such as network destination. A success response must indicate in its message body which media the callee wishes to receive and may indicate the media the callee is going to send.

ACK
The ACK request confirms that the client has received a final response to an INVITE request. (ACK is used only with INVITE requests.) 2xx responses are acknowledged by client user agents, all other final responses by the first proxy or client user agent to receive the response. The Via is always initialized to the host that originates the ACK request, i.e., the client user agent after a 2xx response or the first proxy to receive a non-2xx final response. The ACK request is forwarded as the corresponding INVITE request, based on its Request-URI.
The ACK request MAY contain a message body with the final session description to be used by the callee. If the ACK message body is empty, the callee uses the session description in the INVITE request.

OPTIONS
The server is being queried as to its capabilities. A server that believes it can contact the user, such as a user agent where the user is logged in and has been recently active, may respond to this request with a capability set. A called user agent MAY return a status reflecting how it would have responded to an invitation, e.g., 600 (Busy). Such a server SHOULD return an Allow header field indicating the methods that it supports. Proxy and redirect servers simply forward the request without indicating their capabilities.

BYE
The user agent client uses BYE to indicate to the server that it wishes to release the call. A BYE request is forwarded like an INVITE request and may be issued by either caller or callee. A party to a call should issue a BYE request before releasing a call ("hanging up"). A party receiving a BYE request must cease transmitting media streams specifically directed at the party issuing the BYE request.

CANCEL
The CANCEL request cancels a pending request with the same Call-ID, To, From and CSeq (sequence number only) header field values, but does not affect a completed request. (A request is considered completed if the server has returned a final status response.)

A user agent client or proxy client may issue a CANCEL request at any time. A proxy, in particular, may choose to send a CANCEL to destinations that have not yet returned a final response after it has received a 2xx or 6xx response for one or more of the parallel-search requests. A proxy that receives a CANCEL request forwards the request to all destinations with pending requests.
The Call-ID, To, the numeric part of CSeq and From headers in the CANCEL request are identical to those in the original request. This allows a CANCEL request to be matched with the request it cancels. However, to allow the client to distinguish responses to the CANCEL from those to the original request, the CSeq Method component is set to CANCEL. The Via header field is initialized to the proxy issuing the CANCEL request. (Thus, responses to this CANCEL request only reach the issuing proxy.)
Once a user agent server has received a CANCEL, it must not issue a 2xx response for the cancelled original request.
REGISTER
A client uses the REGISTER method to register the address listed in the To header field with a SIP server.
A user agent MAY register with a local server on start-up by sending a REGISTER request to the well-known "all SIP servers" multicast address "sip.mcast.net" (224.0.1.75). This request SHOULD be scoped to ensure it is not forwarded beyond the boundaries of the administrative system. This MAY be done with either TTL or administrative scopes, depending on what is implemented in the network. SIP user agents MAY listen to that address and use it to become aware of the location of other local users; however, they do not respond to the request. A user agent MAY also be configured with the address of a registrar server to which it sends a REGISTER request upon start-up.
Requests are processed in the order received. Clients SHOULD avoid sending a new registration (as opposed to a retransmission) until they have received the response from the server for the previous one.
The meaning of the REGISTER request-header fields is defined as follows. We define "address-of-record" as the SIP address that the registry knows the registrant, typically of the form "user@domain" rather than "user@host". In third-party registration, the entity issuing the request is different from the entity being registered.
To: The To header field contains the address-of-record whose registration is to be created or updated.
From: The From header field contains the address-of-record of the person responsible for the registration. For first-party registration, it is identical to the To header field value.

#### 2.1.8.4.2 Response:

After receiving and interpreting a request message, the recipient responds with a SIP response message. The response message format is shown below:
Response = Status-Line *( general-header | response-header | entity-header ) CRLF [ message-body ]
Status-Line = SIP-version SP Status-Code SP Reason-Phrase CRLF
Status-Code = Informational | Success | Redirection | Client-Error | Server-Error | Global-Failure | extension-code
extension-code = 3DIGIT
Reason-Phrase = *<TEXT-UTF8, excluding CR, LF>

1xx:
Informational -- request received, continuing to process the request;
2xx:
Success -- the action was successfully received, understood, and accepted;
3xx:
Redirection -- further action needs to be taken in order to complete the request;
4xx:
Client Error -- the request contains bad syntax or cannot be fulfilled at this server;
5xx:

Server Error -- the server failed to fulfil an apparently valid request;
6xx:
Global Failure -- the request cannot be fulfilled at any server.

SIP response codes are extensible. SIP applications are not required to understand the meaning of all registered response codes, though such understanding is obviously desirable. However, applications must understand the class of any response code, as indicated by the first digit, and treat any unrecognized response as being equivalent to the x00 response code of that class, with the exception that an unrecognized response must not be cached.

#### 2.1.8.4.3 SIP-MESSAGE

Both Request and Response messages use the generic-message format of RFC 822 for transferring entities (the body of the message). Both types of messages consist of a start-line, one or more header fields (also known as "headers"), an empty line (i.e., a line with nothing preceding the carriage-return line-feed (CRLF)) indicating the end of the header fields, and an optional message-body.

| | |
|---|---|
| generic-message | = start-line *message-header CRLF [ message-body ] |
| start-line | = Request-Line \| Status-Line |
| message-header | = ( general-header \| request-header \| response-header \| entity-header ) |

***Table 3 Generic Messages***

| | |
|---|---|
| general-header | = Accept \| Accept-Encoding \| Accept-Language \| Call-ID \| Contact \| CSeq \| Date \| Encryption \| Expires \| From \| Record-Route \| Timestamp \| To \| Via |
| entity-header | = Content-Encoding \| Content-Length \| Content-Type |
| request-header | = Authorisation \| Contact \| Hide \| Max-Forwards \| Organization \| Priority \| Proxy-Authorisation \| Proxy-Require \| Route \| Require \| Response-Key \| Subject \| User-Agent |
| response-header | = Allow \| Proxy-Authenticate \| Retry-After \| Server \| Unsupported \| Warning \| WWW-Authenticate |

***Table 4 Generic Messages - Headers***

#### 2.1.8.4.4 SIP-URL

A SIP URL follows the guidelines of RFC 2396 and has the syntax shown below. It is described using Augmented Backus-Naur Form. Note that reserved characters have to be escaped and that the "set of characters reserved within any given URI component is defined by that component. In general, a character is reserved if the semantics of the URI changes if the character is replaced with its escaped US-ASCII encoding".

This part of the theory has been added cause we need the understanding of SIP URL's later on, when sniffing SIP messages.

| | |
|---|---|
| SIP-URL | = "sip:" [ userinfo "@" ] hostport url-parameters [ headers ] |
| userinfo | = user [ ":" password ] |
| user | = *( unreserved \| escaped \| "&" \| "=" \| "+" \| "$" \| "," ) |
| password | = *( unreserved \| escaped \| "&" \| "=" \| "+" \| "$" \| "," ) |
| hostport | = host [ ":" port ] |
| host | = hostname \| IPv4address |
| hostname | = *( domainlabel "." ) toplabel [ "." ] |
| domainlabel | = alphanum \| alphanum *( alphanum \| "-" ) alphanum |
| toplabel | = alpha \| alpha *( alphanum \| "-" ) alphanum |

| | |
|---|---|
| IPv4address | = 1*digit "." 1*digit "." 1*digit "." 1*digit |
| port | = *digit |
| url-parameters | = *( ";" url-parameter ) |
| url-parameter | = transport-param \| user-param \| method-param \| ttl-param \| maddr-param \| other-param |
| ttl-param | = "ttl=" ttl |
| ttl | = 1*3DIGIT ; 0 to 255 |
| transport-param | = "transport=" ( "udp" \| "tcp" ) |
| maddr-param | = "maddr=" host |
| user-param | = "user=" ( "phone" \| "ip" ) |
| method-param | = "method=" Method |
| tag-param | = "tag=" UUID |
| UUID | = 1*( hex \| "-" ) |
| other-param | = ( token \| ( token "=" ( token \| quoted-string ))) |
| headers | = "?" header *( "&" header ) |
| header | = hname "=" hvalue |
| hname | = 1*uric |
| hvalue | = *uric |
| uric | = reserved \| unreserved \| escaped |
| reserved | = ";" \| "/" \| "?" \| ":" \| "@" \| "&" \| "=" \| "+" \| "$" \| "," |
| digits | = 1*DIGIT |
| telephone-subscriber | = global-phone-number \| local-phone-number |
| global-phone-number | = "+" 1*phonedigit [isdn-subaddress] [post-dial] |
| local-phone-number | = 1*(phonedigit \| dtmf-digit \| pause-character) [isdn-subaddress] [post-dial] |
| isdn-subaddress | = ";isub=" 1*phonedigit |
| post-dial | = ";postd=" 1*(phonedigit \| dtmf-digit \| pause-character) |
| phonedigit | = DIGIT \| visual-separator |
| visual-separator | = "-" \| "." |
| pause-character | = one-second-pause \| wait-for-dial-tone |
| one-second-pause | = "p" |
| wait-for-dial-tone | = "w" |
| dtmf-digit | = "*" \| "#" \| "A" \| "B" \| "C" \| "D" |

***Table 5 SIP-URL***

### 2.1.9 Comparing SIP to H.323

There are numerous differences between SIP and H.323. The first is scope;

H.323 specifies a complete, vertically integrated system. Not much room is left for flexibility or different architectures. SIP, on the other hand, is a single component. It works with RTP, for example, but does not mandate it. SIP systems can be composed into a variety of architectures, and numerous protocols and additional systems can be plugged in at the discretion of the service provider. SIP can be considered a building block, whereas H.323 is a specific system.

| | |
|---|---|
| H.323 | ITU developed H.323.<br>Version 1 standardized in 1996.<br>Focus was multimedia communications services for LANs without QoS. H.323 |

| | |
|---|---|
| | v.1 not targeted for IP specifically, but any type of packet LAN.<br>Version 2, released in 1998.<br>Version 3 and 4 has been completed |
| SIP | IETF, origins in late 1996 as a component of the "Mbone" set of utilities and protocols.<br>Focus on distribution of multimedia content, including talks and seminars, broadcasts of space shuttle launches, and IETF meetings.<br>mechanism for inviting users to listen in on an ongoing or |
| H.323 | Complete, vertically integrated suite of protocols architecture for delivering multimedia conferencing applications.<br>Includes signaling, registration, admission control, security, interworking requirements with H.320, H.321, and other ITU conferencing systems, inter-domain data exchange, transport, and codec's.<br>Defines several entities, including terminals (end systems, like PCs), gateways, multipoint conferencing units, and something called a gatekeeper. A gatekeeper is similar to a SIP proxy, in that it plays the role of a signaling relay. |
| SIP | single component, works with e.g. RTP but does not mandate it<br>SIP systems can be composed into a variety of architectures, and numerous protocols and additional systems can be plugged in at the discretion of the service provide |
| H.323 | LAN protocol; numerous enhancements (such as FastStart) added to gain behaviour as a wide-area protocol |
| SIP | Designed as a wide-area protocol, no enhancements needed. |
| H.323 | borrows its call-signaling component from existing work done in ITU, namely the Q.931 protocol, used for user-to-network signaling in ISDN => telephony-centric flavour |
| SIP | Borrows much of its concepts from HTTP => web flavour<br>allows to integrate with web, e-mail, and other existing IP applications.<br>KISS (Keep It Simple Stupid) principle => easier to implement and interoperate |
| H.323 | extendable by add non-standard elements identified by vendor ID and version change => backward compatible, takes up more room than its predecessor |
| SIP | extended in numerous ways: including adding headers, new methods, new bodies, and parameters to existing headers |
| H.323 | H.245 contains powerful mechanisms for conference control for distributed multiparty conferences. (deny - grant speaking privileges) |
| SIP | kind of control possible within SIP-established conference, but not addressed by SIP itself, currently no standalone standard protocols that can do this |

### 2.1.10 Connectivity

SIP transparently supports name mapping and redirection services, allowing the implementation of ISDN and Intelligent Network telephony subscriber services. The phone identifier is to be used when connecting to a telephony gateway. Even without this parameter, recipients of SIP URLs MAY interpret the pre-@ part as a phone number if local restrictions on the name space for user name allow it.

### 2.1.11 Main Advantages

#### *2.1.11.1 Services*

SIP transparently supports name mapping and redirection services, allowing the implementation of ISDN and Intelligent Network telephony subscriber services. These facilities also enable personal mobility.

Internet telephony began on the premise that it was cheaper than normal phone calling. Users were willing to tolerate degraded quality or reduced function for lower cost. However, the cost differentials are rapidly disappearing. To continue to exist, Internet telephony must find another reason to be. The answer is services.
Some of the most exciting applications have already found killer status on the Internet, though not (yet) in the form of multimedia services. Now think of integrating multimedia communications, such as voice, with web, e-mail, buddy lists, instant messaging, and online games. Whole new sets of features, services, and applications become conceivable.
SIP is ideally suited here. Its use of URLs, its support for MIME and carriage of arbitrary content (SIP can carry images, MP3s, even Java applets), and its usage of e-mail routing mechanisms, means that it can integrate well with these other applications. For example, it is just as easy to redirect a user to another phone as it is to redirect a user to a web page.

#### *2.1.11.2 Scalability*

SIP uses the Internet model for scalability - fast and simple in the core, smarter with less volume in the periphery. To accomplish this, SIP defines several types of proxy servers. "Call-stateful" proxies generally live at the edge of the network. These proxies track call state, and can provide rich sets of services based on this knowledge. Closer to the core, "transaction-stateful" (also known as just "stateful") proxies track requests and responses, but have no knowledge of session or call state. Once a session invitation is accepted, the proxy forgets about it. When the session termination arrives, the proxy forwards it without needing to know about the session.
Finally, "stateless" proxies exist in the core. These proxies receive requests, like INVITE, forward them, and immediately forget. The SIP protocol provides facilities to ensure that the response can be correctly routed back to the caller. Stateless proxies are very fast, but can provide few services. Call-stateful proxies are not as fast, but they live at the periphery, where call volumes are lower.

#### *2.1.11.3 Extensibility*

History has taught Internet engineers that protocols get extended and used in ways they never intended (e-mail and web are both excellent examples of this). So, they've learned to design in support for extensibility from the outset. SIP has numerous mechanisms to support extensions. It does not require everyone to implement the extensions. Facilities are provided that allow two parties to determine the common set of capabilities, so that a session initiation can always be completed, no matter what.

#### *2.1.11.4 Flexibility*

SIP is not a complete system for Internet telephony. It does not dictate architecture, usage patterns, or deployment scenario. It does not mandate how many servers there are, how they are connected, or where they reside. This leaves operators tremendous flexibility in how the protocol is used and deployed. One way to think of it is that SIP is a LEGO block; operators can piece together a complete solution by obtaining other LEGO blocks, and putting them together in the way that they see fit.

#### *2.1.11.5 Multimedia*

Besides the traditional call-forwarding, follow-me, and do-not-disturb, SIP has the potential for enabling a whole new class of services that integrate multimedia with web, e-mail, instant messaging, and "presence" (meant here as, "are you currently online?"). The value that the Internet brings to Internet telephony is the suite of existing applications that can be merged with

voice and video communications. As an example, at the end of a call, a user can transfer the other party to a web page instead of another phone. This transfer would end the call, and cause the other party's web browser to jump to the new page. In essence, the value of VoIP and SIP comes not from integration at the network layer (i.e., run your voice services on top of your data network), but at the services layer (i.e., combine your voice services with your data services)

### 2.1.12 Main Drawbacks:

Emerging issues in the Internet could ruin the promise of SIP (as well as H.323) over the long term. The problem is the shortage of IP numbers and the growing use of network address translators (NATs). There are similar issues when running SIP and H.323 through firewalls.

NATs break many protocols that act as establishment mechanisms for other protocols, such as SIP. NATs provide a boundary between the private IP addressing of a network and the public Internet. They are most often used if an enterprise is unable to secure access to a sufficient block of IP numbers from their ISP, or if the enterprise wants the presumed luxury of being able to switch ISPs without having to renumber their network.

SIP, fundamentally, is a control channel for establishing other sessions (namely, the media sessions). These kinds of protocols (of which FTP and H.323 are other examples) cause problems for NATs, since the addresses for the established sessions are in the body of the application layer messages.

When used with SDP, SIP messages carry the IP addresses and ports to be used for the media sessions. There may be multiple media sessions within a particular SIP call. Since SDP carries IP addresses and not host names, the external caller user agent will send media to an IP address that is not globally routable. It is only a valid IP address within the private network.

A nearly identical problem exists for firewalls. When a user inside the firewall sends media to an address outside the firewall, it will be dropped by the firewall unless a rule is established to allow it to pass. Since the media is sent on dynamic ports to dynamic addresses, these rules must be dynamically installed through application-aware devices, such as proxies.

### 2.1.13 Interfacing

Developing services, of course, requires APIs. What kind of APIs are used to program services delivered by SIP? There has been significant activity in this area, resulting in numerous new interfaces, each with its own distinct set of strengths and weaknesses.

The first API that surfaced is the call processing language (CPL). CPL is not actually an API, but rather an XML-based scripting language for describing call services. It is not a complete programming language, either. It has primitives for making decisions based on call properties, such as time-of-day, caller, called party, and priority, and then taking actions, such as forwarding calls, rejecting calls, redirecting calls, and sending e-mail. CPL is engineered for end-user service creation.

A server can easily parse and validate a CPL, guarding against malicious behaviour. The running time and resource requirements of a CPL can also be computed automatically from the CPL. An interpreter for CPL is very lightweight, allowing CPL services to execute very quickly. For these reasons, it is possible for an end user to write a CPL (typically with some kind of GUI tool), upload it to the network, and have it instantly verified and instantiated in real time.

At the opposite end of the spectrum in SIP is CGI (the common gateway interface). Many web designers are familiar with HTTP CGI; it's an interface that allows people to generate dynamic web content using Perl, Tcl, or any other programming language of choice. Since HTTP and SIP are so similar, it was recognized that an almost identical interface could be used for SIP. The result is SIP CGI, which is roughly 90% equivalent to HTTP CGI. Like HTTP CGI, SIP CGI passes message parameters through environment variables to a script that runs in a separate process.

The process sends instructions back to the server through its standard output file descriptor. The benefit of SIP CGI is that it makes development of SIP services work much like the creation of dynamic web content. In fact, for SIP services that contain substantial web components, development will closely mirror web-only services. The importance of leveraging web tools for voice service creation is that a much larger class of developers becomes available.

CGI has substantially more flexibility than CPL (CGI doesn't even mandate a particular programming language), but is much more risky to execute. Furthermore, because of its usage of separate processes, SIP CGI doesn't scale as well as CPL. Somewhere in the middle are SIP Servlets. HTTP Servlets are in wide use for developing dynamic web content. Servlets are very similar to the CGI concept. However, instead of using a separate process, messages are passed to a class that runs within a JVM (Java Virtual Machine) inside of the server. As a result, Servlets are restricted to Java, but suffer less overhead than SIP CGI. Use of a JVM for executing servlets means that the Java "sandbox" concept can be applied to protect the server from the script. Like SIP CGI, SIP Servlets closely mirror the operation of HTTP Servlets; they simply enhance the interface to support the wider array of functions a proxy can execute, as compared to an HTTP origin server

### 2.1.14 Real Time Transport Protocol

IETF audio-video transport group started to develop RTP in 1993. The aim of the protocol was to provide services required by interactive multimedia conferences, such as play-out synchronization, demultiplexing, media identification and active party identification. However, not only multimedia conferencing applications can benefit from RTP, but also storage of continuous data, interactive media distribution, distributed simulation, and control applications can utilize RTP

RTP consists of a data and a control part. The latter is called RTCP. Implementation will often be integrated into application rather than being implemented as a separate protocol layer. In applications RTP is typically run on top of UDP to make use of its port numbers and checksums. The RTP framework is relatively "loose" allowing modifications and tailoring depending on application. Additionally, a complete specification for a particular application will require a payload format and profile specification. The payload format defines how a particular payload is to be carried in RTP. A payload specification defines how a set of payload type codec's are mapped into payload formats.

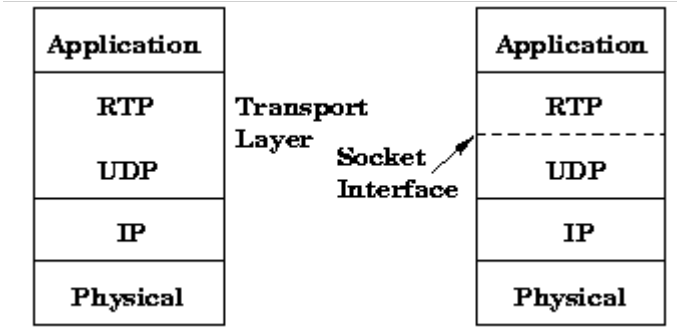


***Figure 3 Real-Time Transport Protocol (RTP)***

RTP session setup consists of defining a pair of destination transport addresses one IP address and UDP port pair, one for RTP and another for RTCP. In the case of multicast conference the IP address is a class D multicast address. In multimedia session each medium is carried in a separate RTP session with its own RTCP packets reporting only the quality of that session. Usually additional media are allocated in additional port pairs and only one multicast address is used for the conference.

RTP has important properties of a transport protocol: it runs on end systems, it provides demultiplexing. It differs from transport protocols like TCP in that it (currently) does not offer any form of reliability or a protocol-defined flow/congestion control. However, it provides the necessary hooks for adding reliability, where appropriate, and flow/congestion control; (Application-level framing), as lower layers are required to transfer data RTP is not really real time, but provides functionality suited for carrying real-time content, e.g., a timestamp and control mechanisms for synchronizing different streams with timing properties.

User Datagram Protocol RFC 768:

User Datagram Protocol is a 'Connectionless' protocol. It uses IP to send datagram's in a similar way to TCP, except that like IP, and unlike TCP, UDP does not care if the packets reach their destination. UDP is used in applications where it is not essential for 100% of the packets to arrive. This may sound strange, but often you don't need all the packets. You wouldn't use UDP to transmit a program, because if one single bit was wrong (let alone loosing a whole packet) the file would be absolutely useless. It is up to program designers to choose what method is most suitable. While TCP is safer, UDP is becoming more common. It especially favoured for 'Streaming' or Real-time applications. More recently, internet applications have used both UDP and TCP. TCP is used for the essential or Control data, while UDP is used for data for which losses are acceptable.

### 2.1.15 Quality of Service (QoS)

Perhaps the most vexing problem in voice-over-IP, in general, has been the issue of quality of service. The delay in conversations that many VoIP users encounter is caused by the jitter and latency of packet delivery within the Internet itself. It's useful to review some of the basic principles of the Internet to understand what can be done about the problem, what the IETF's response has been, and how it impacts SIP.

Currently, the Internet offers a single service, traditionally referred to as "best effort." In other words, all packets are created equal. There is no difference to the Internet whether a packet is e-mail, FTP, or the download of a web page. If the Internet gets very busy, packets get dropped or delayed.

Unfortunately, the human ear is extremely sensitive to latency in the delivery of sound. The human ear can detect delays of 200 milliseconds or greater in voice conversations.
SIP itself does not get involved in reservation of network resources or admission control. This is because SIP messages may not even run over the same networks that the voice packets traverse. The complete independence of the SIP path and the voice path enables ASPs to provide voice services without providing network connectivity. This is an extremely important advantage of the SIP architecture. Given this, SIP relies on other protocols and techniques in order to provide quality of service.

### 2.1.16 Encryption

SIP requests and responses can contain sensitive information about the communication patterns and communication content of individuals. The SIP message body MAY also contain encryption keys for the session itself. SIP supports three complementary forms of encryption to protect privacy:

End-to-end encryption of the SIP message body and certain sensitive header fields;
hop-by-hop encryption to prevent eavesdropping that tracks who is calling whom;
hop-by-hop encryption of Via fields to hide the route a request has taken.

Not all of the SIP request or response can be encrypted end-to-end because header fields such as To and Via need to be visible to proxies so that the SIP request can be routed correctly. Hop-by-

hop encryption encrypts the entire SIP request or response on the wire so that packet sniffers or other eavesdroppers cannot see who is calling whom. Hop-by-hop encryption can also encrypt requests and responses that have been end-to-end encrypted. Note that proxies can still see who is calling whom, and this information is also deducible by performing a network traffic analysis, so this provides a very limited but still worthwhile degree of protection.

SIP Via fields are used to route a response back along the path taken by the request and to prevent infinite request loops. However, the information given by them can also provide useful information to an attacker. End-to-end encryption relies on keys shared by the two user agents involved in the request. Typically, the message is sent encrypted with the public key of the recipient, so that only that recipient can read the message. All implementations should support PGP-based encryption and may implement other schemes.

A SIP request (or response) is end-to-end encrypted by splitting the message to be sent into a part to be encrypted and a short header that will remain in the clear. Some parts of the SIP message, namely the request line, the response line and certain header fields need to be read and returned by proxies and thus MUST NOT be encrypted end-to-end. Possibly sensitive information that needs to be made available as plaintext includes destination address (To) and the forwarding path (Via) of the call. The Authorization header field must remain in the clear if it contains a digital signature as the signature is generated after encryption, but MAY be encrypted if it contains "basic" or "digest" authentication. The From header field should normally remain in the clear, but MAY be encrypted if required, in which case some proxies MAY return a 401 (Unauthorized) status if they require a From field.

#### *2.1.16.1 Privacy of SIP Responses*

SIP requests can be sent securely using end-to-end encryption and authentication to a called user agent that sends an insecure response. This is allowed by the SIP security model, but is not a good idea. However, unless the correct behaviour is explicit, it would not always be possible for the called user agent to infer what a reasonable behaviour was. Thus when end-to-end encryption is used by the request originator, the encryption key to be used for the response should be specified in the request. If this were not done, it might be possible for the called user agent to incorrectly infer an appropriate key to use in the response. Thus, to prevent key-guessing becoming an acceptable strategy, we specify that a called user agent receiving a request that does not specify a key to be used for the response should send that response unencrypted.

Any SIP header fields that were encrypted in a request should also be encrypted in an encrypted response. Contact response fields MAY be encrypted if the information they contain is sensitive, or MAY be left in the clear to permit proxies more scope for localized searches.

## 2.2 Network Technology

The goal of this chapter is to give the reader a platform for understanding how the Internet works. That includes addressing, routing, subnets, TCP/IP and related services. This will give a platform for understanding how IP-Telephony traffic is handled in a TCP/IP network, and will be useful knowledge for the reader in later chapters (Pages 56 and out). As mentioned before, the theory chapters where the reader has good knowledge can be skipped. However, all theory chapters are used as a basis for understanding the analysis later in this paper.

### 2.2.1 OSI-model and IP packets

The OSI model is a model that shows how communication takes place in “layers,” where the layers have responsibility for different functions. Below is a simplified model of the OSI-model

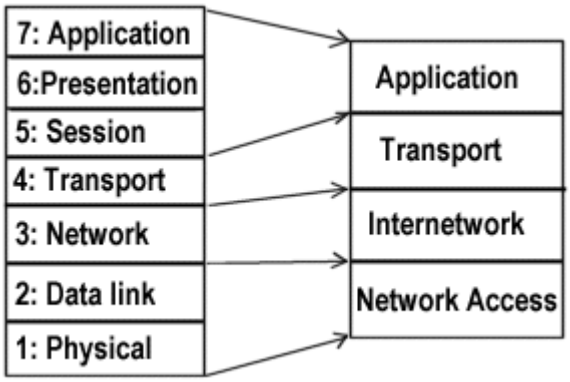


***Figure 4 Simplified OSI-model***

The left layers make the traditional OSI-model. To the right is a simplified version that is easy to use and understand. In short the communication takes place like this (simplified model):

An application sends data to the transport layer
The transport layer splits the data in suitable parts, and sends these parts to the internetwork layer.
Internetwork layer uses the IP-protocol and makes packages of the data. Then it sends these packages on to the Network Access layer.
The Network Access layer puts the IP-packages in frames, and sends them out on the network.

At the receiver, the process is reversed:

The Network Access layer reads the bit pattern that is sent on the network and collects it to a frame. Then the IP-packet within the frame is sent up to the Internetwork layer.
When received, the Internetwork layer decides what to do with the packages; if it is to be sent to another network interface, or up to the transport layer. If the packages have an Application on this computer as destination, it is sent to the transport layer.
The transport layer puts the IP-packages together in data blocks. Dependent on which protocol this layer uses, it can support resending of missing packages. When all packages in a data block is received, this block is sent to the application

In Figure 5 we will look at use of the IP-protocol in the Internetwork layer and TCP or UDP in the Transport layer.
For each layer there is added a header that is specific for the protocol and the respective layer. If we assume that TCP is used in the Transport layer, IP in the Internet layer and Ethernet as network, it will look like this:

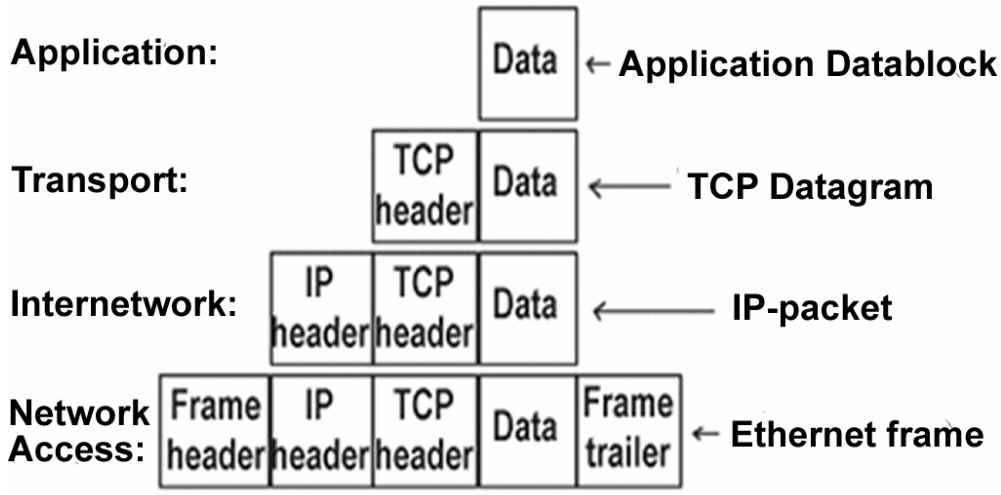


***Figure 5 IP in TCP and UDP***

More detailed it looks like the tables below, number of bytes in parenthesis and the header or trailer is in blue:
TCP-datagram, (header 20 bytes):

| Source port no (2) | Destination port no (2) |
|---|---|
| Sequence no (4) | |
| Acknowledge no (4) | |
| Flags (2) | Window size (2) |
| TCP checksum (2) | Urgent pointer (2) |
| Options | |
| Data | |

***Figure 6 TCP Datagram***

With an IP-package an IP-address for sender and receiver is added (header 20 bytes):

| Version/length/TOS (2) | Total length (2) |
|---|---|
| Identification (2) | Flags (2) |
| TTL (1) Protocol (1) | Header checksum (2) |
| Source IP-address (4) | |
| Destination IP-address (4) | |
| Options | |
| Data | |

***Figure 7 IP Package***

Some of the fields are simplified. Look at RFC 0791 for a complete description. The data field here is a TCP-datagram.
If we use the IEEE802.3 protocol, an Ethernet frame looks like this:

| destination address (6) |
|---|
| source address (6) |
| type (2) |
| Data (46-1500) |
| CRC (4) |
| |

***Table 6 Ethernet frame with IEEE802.3 Protocol***

If we use Ethernet the type is given by the value $080_{16}$ and the data part is an IP-package with variable length 46 to 1500 bytes. The trailer CRC (Cyclic Redundancy Check) is a checksum for error detection. The addresses are 6 bytes addresses that are a hardware address that identifies the network card.
Description of the capsuling of IP-packages in Ethernet frames can be found in RFC 894.

**On the Ethernet the MAC-addresses defines sender and receiver. A MAC-address is an address deployed in all units that can send information on an Ethernet. The IP-package lies as data in an Ethernet frame.**

### 2.2.2 IP-addresses

An IP-address contains of 4 bytes, and is either shown as decimals or binaries. In both cases the address is written with a "full stop" between each byte. I.e. 158.36.51.106 as decimal, or 10011110.00100100.00110011.01101010 as binary. The use of IP-addresses as binary may look somehow unpractical, but that is because you often give sub groups of bits different meaning in the address, and that is why binary often is needed. This will be discussed later.

If we don't consider some special cases, it is possible to say that all machinery connected to the global Internet has a specific IP-address. The target of giving all specific addresses is of course a clear definition of the address to the receiver. For this to happen in an effective way, we depend on a certain structure for the addresses. That is; deploying addresses in a hierarchic way, and not by random – i.e. chronologically.

The main idea for doing this is that the IP-address is split into a part that gives the network number (often called Network ID), and the rest of the address is used to identify the computer in this network with a host ID- To send IP-packages to another host is done by finding the net where the host is, and send the package there.

The first that is done is to split the entire address span into 5 different classes of net. These nets are classified as A, B, C, D and E–nets:

- A class-A net is given like this: The first byte defines network number, and the first bit in network ID is 0. The IP-addresses starts with values from 1 to 126. That is 126 possible values. The last 3 bytes is used to specify host ID and for each A-network we can have 2.777.214 host ID's.

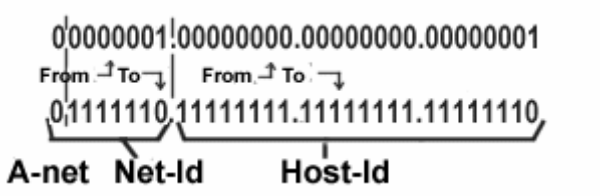


***Figure 8 Class-A net***

- A class-B net is given by the two first bit specifying the network number and the 2 first bits in first byte are 10. The IP-addresses starts with values from 128 to 191. That gives 64 possible values in first byte. For each of these values in the first byte, the second byte can have 256 different values – which give a total of 16.384 class-B nets. The last two bytes is used for host ID, and for each B-net you can have 65.534 host ID's.

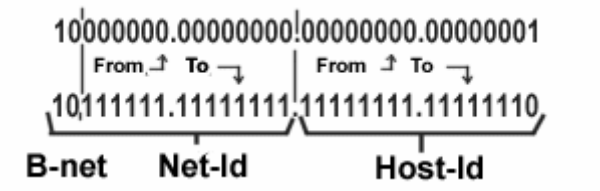


***Figure 9 Class-B net***

- For a class-C net, the first 3 bytes is used to give the network number, and the first byte starts with the bit values 110. The IP-addresses in these nets starts with 11000000 and ends with 11011111. That is 192 to 223 as decimal values. The two next bytes in network ID can operate with values from 0 to 255 and in total this gives 2.097.152 possible nets. The last byte is used for host ID and for each C-net it is possible to have 254 host ID's.

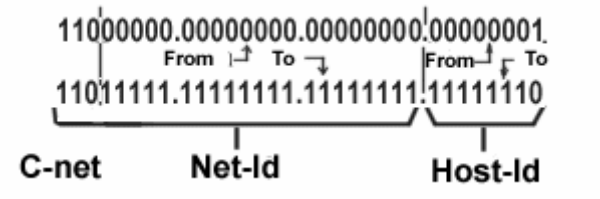


***Figure 10 Class-C net***

In all 3 net-classes the rule that host ID's cannot consist of only 0 or 1 complies. Therefore we get i.e. 254 possible host ID's on a C-net, and not 256 as you easily could think. That is; the host ID 00000000 and 11111111 can not be used. This also complies for an A and B-net. If you calculate it, there is a total of $2^{24}$-2 host ID's in an A-net, and $2^{16}$-2 in a B-net.
Another particularity can be found in A-nets, where the first allowed network ID is 00000001 and the last is 01111110. This is a result of not using the 00000000 net, and that all addresses starting with 01111111 is so-called "loop back" addresses. All packages sent to such an address is to be sent back to the application it originated from. These packages are therefore not sent out on the net at all.
The network Classes E and D is identified by first bit starting with 1110 and 1111. These nets are reserved for particular porpoises.
The splitting into net classes A, B and C is still used, but there is also another notation for specifying net classes. What separates net classes is how many bits that is used for net ID and we can se that class-A uses 8 bits, class-B 16 bits and class-C 24 bits. This is also written as /8, /16 and /24 bit nets and a specific C net can be written as i.e. 158.36.51.0/24. That means that the net uses 24 bits for network ID, and the 3 first bytes in the IP-addresses in the net is 158, 36 and 51. The last byte is used for host ID.
This opens for a use of other sizes for net ID than 8, 16 and 24 bit, and that is discussed in the next chapter.

### 2.2.3 Subnets

When a company is given a net, they get a net of either A, B or C-class. Within the given net, they are themselves responsible for giving out host ID's to the equipment. If you i.e. are given the net 158.36.51.0/24, you need to make a local network and give out IP-addresses possible to use within this net.

It is fully possible to make a single segment for all these addresses, but often you wish to organize the local net into smaller parts by geographical or organizational categories. To do this you often establish so called subnets. A subnet is no other than a segregation of a given net into smaller parts. It is also possible to split subnets into more subnets and subnet levels. If split into a lot of subnets, the subnets are often getting small (can only contain a small number of hosts). Therefore it is rarely used a lot of subnet levels.
Each subnet is most often realized by a unique network segment, even if it is possible to use several subnets on one segment.

Splitting into subnet takes place by taking a certain part of the bit group that is meant for host ID, and splitting it into a subnet ID and a new host ID. If you initially i.e. have a 8 bit host ID, it is possible to decide that 3 of this bits is to be used for subnet ID and then it is left 5 bits for the host ID in each subnet.

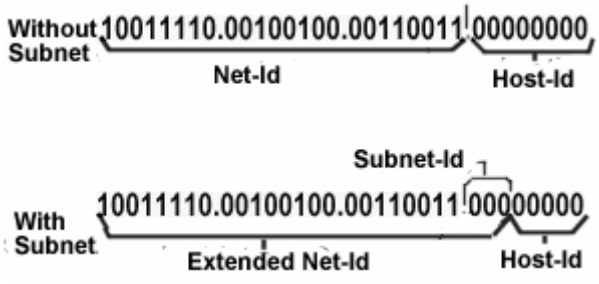


***Figure 11 Subnet and Subnet ID's***

The given network address gives the possibility to use IP-addresses on a certain interval, and this interval is split into a number of subnets. For each subnet you get access to a number of addresses. With 3 bits for subnet ID you should get 8 (=$2^3$) subnets each with 32 ($2^5$) hosts. To find the IP-addresses possible to use in each subnet, the subnet ID and host ID is put together as a bit pattern, and the decimal value is calculated. With a 3 bit subnet, the last byte in the IP-addresses is like this:

| Subnet ID | Lowest host ID | Highest host ID | Lowest IP (binary) | Highest IP (binary) | Lowest IP (decimal) | Highest IP (decimal) |
|---|---|---|---|---|---|---|
| 001 | 00001 | 11110 | 00100001 | 00111110 | 33 | 62 |
| 010 | 00001 | 11110 | 01000001 | 01011110 | 65 | 94 |
| 011 | 00001 | 11110 | 01100001 | 01111110 | 97 | 126 |
| 100 | 00001 | 11110 | 10000001 | 10011110 | 129 | 158 |
| 101 | 00001 | 11110 | 10100001 | 10111110 | 161 | 190 |
| 110 | 00001 | 11110 | 11000001 | 11011110 | 193 | 222 |

***Figure 12 Subnet and IP-addresses***

From the table it is possible to se that we get subnet with IP-address 158.36.51.33 – 158.36.51.62 for the first subnet and so on.

If you calculate, it is possible to see that legal IP-addresses are reduced from 254 without subnet, to 180 with subnet (6 subnets each with 30 addresses). This is caused by the rule of subnets and host that is lost. This "waist" of IP-addresses has resulted in a lot of equipment that doesn't follow the rules, and allows subnet ID's with only 0's and 1's – so that a full use of the address spans is achieved. Be aware that this breaks with the rules, and that it is possible to run into equipment that does not follow the rules, and therefore can make possible error situations. Host ID with only zero's is to be reserved for network address and host ID, while only 1's are reserved for broadcast.

Sometimes it is demanded to state something called a net mask. This is 4 bytes, similar to an IP-address. The net mask gives a bit pattern that shows how much of an IP-address that is extended network ID and host ID. The bit pattern shows bit value equal 1 for bits related to network ID and 0 for bits related to host ID. It is always filled up with 1 from left towards right, without any zeros in-between. I.e. The net mask for 158.36.51.0/27 equals 11111111.11111111.11111111.11100000, or decimal 255.255.255.224. Since you always fill up 1 from the left, it is only 9 values possible in a net mask. These are:

| Bit pattern | Decimal value |
|---|---|
| 00000000 | 0 |
| 10000000 | 128 |
| 11000000 | 192 |
| 11100000 | 224 |
| 11110000 | 240 |

| | |
|---|---|
| 11111000 | 248 |
| 11111100 | 252 |
| 11111110 | 254 |
| 11111111 | 255 |

***Figure 13 Net mask***

The net mask i.e. used to find if an address belongs in a certain net or not, something you need to know when IP-packages are to be routed in the net.

The RFC 950 describes how you split a net into subnets.

### 2.2.4 Routing, ARP and IP

Earlier it has been described briefly how an IP-package is made, put within a frame, sent on the locale network segment and caught by the receiver. This description was simplified and did among other things not consider that an IP-package can be sent for a receiver that is not connected to the local segment, and therefore needs to be sent on somehow.

This way to send an IP-package is called routing. To do this you need a router connected to the network. A router can also be called a gateway.

As an example in this chapter, we are going to look into the following network:

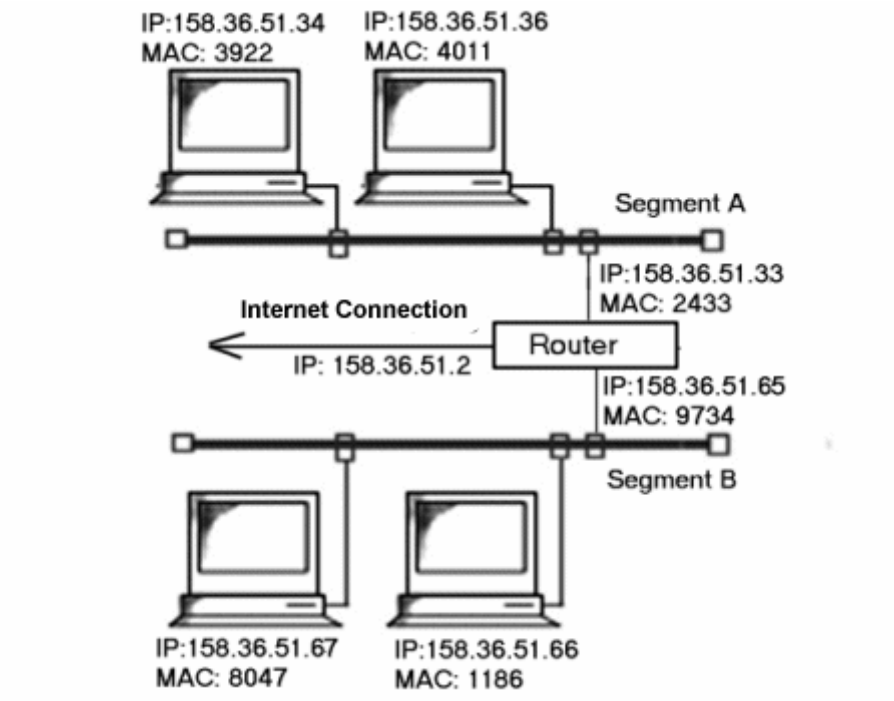


***Figure 14 Sample network for routing issues***

As we see; the network consists of 2 segments each with 2 hosts and a router. The router has 3 network connections, and therefore also 3 IP and Ethernet addresses. All addresses are given in Figure 14 Sample network for routing issues). With this configuration we will look at the following possibilities:

*1. A host in segment A sends an IP-package to a host on the same segment*
*2. A host in segment A sends an IP-package to a host on segment B*
*3. A host in segment A sends an IP-package to a host outside the segments A and B*

#### 2.2.4.1 ARP

We look at case 1 first.
Host 158.36.51.34 will send an IP-package to a host on the local segment, i.e. 158.36.51.36. The sender has the receivers IP-address, but not the Ethernet address. To make a frame and send it on the local segment you need the Ethernet address. To find this Ethernet address, the protocol ARP is used. The host that sends the IP-package first sends a request on the local segment if a host has the specific IP-address. If there is a host on the net that has, it replies by sending its Ethernet address. In our case the host 158.36.51.36 will answer by saying it has Ethernet address 4011. With this Ethernet address the sender is able to make a frame and send it on the local segment.

Routing and routing tables
Next case is 2; sending to a host on another segment.
If the receiver of an IP-package does not exist on the same network segment as the sender, we have to apply routing.

All equipment that is configured to work in an IP-network needs a routing-table. This table tells how IP-packages are to be sent, depending on if the receiver is on the local segment, another segment or if the receivers net is unknown. The table also tells how special cases as broadcast and loop back are to be handled.

The routing table can be found by using the command **route print**. The command **route** can also be used to make or change a table, but is rarely done. A routing table is usually automatically setup when we give the machine IP-number, setting net mask and so on - or it is sat up automatically when we use DHCP. The service DHCP is used for automatic configuration of hosts in an IP-network (this is looked into later).

Anyway it can be interesting to study a table, and first we look at a routing table for an ordinary host with one network interface:

| Network Address | Netmask | Gateway Address | Interface | Metric |
|---|---|---|---|---|
| 0.0.0.0 | 0.0.0.0 | 158.36.51.1 | 158.36.51.111 | 1 |
| 127.0.0.0 | 255.0.0.0 | 127.0.0.1 | 127.0.0.1 | 1 |
| 158.36.51.0 | 255.255.255.0 | 158.36.51.111 | 158.36.51.111 | 1 |
| 158.36.51.111 | 255.255.255.255 | 127.0.0.1 | 127.0.0.1 | 1 |
| 158.36.255.255 | 255.255.255.255 | 158.36.51.111 | 158.36.51.111 | 1 |
| 224.0.0.0 | 224.0.0.0 | 158.36.51.111 | 158.36.51.111 | 1 |
| 255.255.255.255 | 255.255.255.255 | 158.36.51.111 | 158.36.51.111 | 1 |

***Figure 15 Routing table for ordinary host with one network interface***

The table contains 5 rows. The first to rows are used together to decide if a given IP-address fits and that a particular line in the table is to be used. I.e. line 3; 158.36.51.0 and 255.255.255.0 means that the addresses having the 3 first bytes equal 158, 36 and 51 is to be treated from a description in this line. Net mask 255.255.255.255 means that there needs to be an exact relation between the IP-address that is handled and the address given in the row "Network Address." Net mask 0.0.0.0 means no demand to the address handled. This is some sort of "gathering" of all "other" addresses that is not handled by a particular line in the table. Packages not caught by this line are to be sent for a default gateway that is the router that sends packages further on. The router has its own routing table, and the routers further handling of the packages is determined by this table.

In our example all hosts in segment A will have a line in their own routing tables looking like this:

| | | | | |
|---|---|---|---|---|
| 0.0.0.0 | 0.0.0.0 | 158.36.51.33 | 158.36.51.XX | 1 |

***Table 7 Client routing table***

Where XX is 34 or 36. This means that all IP-packages not fitting into other lines in the routing table shall be sent from the net connection 158.36.51.XX to 158.36.51.33.
The router also has its table that among other contains the lines:

| | | | | |
|---|---|---|---|---|
| 0.0.0.0 | 0.0.0.0 | 158.36.51.2 | 158.36.51.1 | 1 |
| 158.36.51.32 | 255.255.255.224 | 158.36.51.33 | 158.36.51.33 | 1 |
| 158.36.51.64 | 255.255.255.224 | 158.36.51.65 | 158.36.51.65 | 1 |

***Table 8 Router table, simple sample***

Lets now again look at example 2, where a host from segment A will send an IP-package to a host on segment B. I.e. from 158.36.51.34 to 158.36.51.66. Host 158.36.51.34 uses ARP, and discover that 158.36.51.66 does not exist in the same segment. From the routing table host 158.36.51.34 discovers that the package should be sent to gateway 158.36.51.33. A new ARP-try is done to find the Ethernet address of this IP-address. The router answers 2433. The sender 158.36.51.34 puts the IP-package for 158.36.51.66 in a frame and sends it to 2433.

The package arrive the router, with destination address 158.36.51.66. The router checks its table, and discovers that the address fits last line above. The router now knows that the package is to be sent on the 158.36.51.65 interface. It makes an ARP-search in this segment, and host 158.36.51.66 answer with its Ethernet address 1186. A frame with the IP-package is made by the router, and sent to 1186.

Sample 3 is quite similar to 2, but the router will not find a particular line in its table, and needs to send the package on to the next router. This it specified in first line, and has IP = 158.36.51.1. The package is sent there, but the details on how this is done depend on the connection between the router and the Internet.

#### *2.2.4.2 IPv6*

What we so far have described is IP version 4. When Internet and IPv4 was designed, this was meant to be a protocol for use a long time, without any need of changes. Problem was that even the most foresighted person couldn't forsee the popularity IP would have, and what problems this would produce. Main problems with Ipv4 are that:

- Soon the available IP addresses will be none.
- The numbers of domains and subnets is so big that routing has become ineffective.

In addition there are some other minor weaknesses, which also needed to be fixed. IP version 5 was an in-between protocol, but the final version for new IP protocol became version 6. The main changes consist of:

- Its used 16 bytes addresses (4 bytes in IPv4). This gives an increase from $4{,}3*10^9$ to about $3{,}4*10^{38}$ possible addresses. A need for more addresses than this is unlikely. With this number you can give about $10^{21}$ addresses pr. $cm^2$ of the earths face. Anyway it is now possible to structure addresses so that routing becomes much easier.
- New fields in the IP-package are added for security (signing and cryptography), reservation of bandwidth and some other possibilities for multimedia support.

IPv6 is designed to support auto configuration, so that machines connecting to a net themselves discover their neighbours, register in their neighbourhood and can take use of services without any manual installation.

For now the use of IPv6 is small, but an increase is expected. Some of the challenges are to let packages from IPv4 and IPv6 use the same net in a transition face. This i.e. means that all routers and services must support both versions.

### 2.2.5 Services in an IP-network (DHCP and DNS)

Services offered within an IP-network (and other nets) take place in a client/server model. This means that services we want to use, is offered by a server – and our machine is considered as a client, served by the server.

On the server that offers services, there is a program waiting for a request from a client, and all services are configured to communicate through a "port." Such type of programs is often called daemons in "UNIX", and services in the "Microsoft world". To identify these services, that is to send an IP-package to a daemon, the servers IP-address and port number is needed.

It is worth mentioning that we are not talking about physical ports on the server, only something found in the servers platform that can be administered by software.

For a client to contact a server, we now know that it needs to know IP-address and port number. The IP-address varies, and the user needs to specify this. Port number can also vary, but some standard services are configured to use defined port numbers. A complete list of reserved port numbers can be found in RFC 1340 [3].

#### *2.2.5.1 DHCP (Dynamic Host Configuration Protocol)*

**The intention with DHCP is to simplify the installation of a host in an IP-network. With DHCP; IP-addresses, configuration of net mask and default gateway is given automatically.**

A normal procedure for setting up a host in an IP-network consists of some manual installation. Every host at least needs to have registered their own IP-address, the local segments net mask and the IP-address for default gateway. In addition to this, the strategy for giving IP-addresses often ends in wasting a lot of addresses. Often you have a lot of machines on the LAN, but there is seldom more than a small share of them always in use. In these cases a dynamic way to give out IP-addresses can be a good thing.

Advantages with use of dynamic IP-addresses therefore are:

- Simple setup of hosts on the net and simpler to make changes.
- More efficient use of the address pan, less waist of addresses.
- In addition it is possible to move a machine with dynamic configuration between segments, and the machine will install itself (it is possible that not all application will work if you move the machine, depending on configuration).

Disadvantages:

- The IP-addresses of machines are changed whenever you log of the network. Applications that register the machines address (its own or others) needs to be updated every time it's used.
- Address conflicts can be experienced. This happens if someone uses an address that is reserved for the dynamic range as a set and locked IP-address.

The protocol used in IP is called DHCP, and it is short for Dynamic Host Configuration Protocol. To use this protocol, it is demanded to have a DHCP server on the segment that is setup so that it can give out sine part of the addresses on the segment on request form hosts. Notice that this

server itself needs to be set up with a specific IP-address. Default gateway, mail-server and some other services on the segment also needs to be setup with a none changing address, so only some of the address pan can be given out dynamically.

With a DHCP-server on the segment, an installation takes place almost like this:

- A host set up to use DHCP sends a request on the local network to find if there is an available DHCP-server. The request is sent without a recipient address, giving that a DHCP server listens for such requests. The senders Ethernet address is attached.
- The server catches the request, finds a free IP-address, and makes an answer. This answer is sent to the registered Ethernet address, with information of IP-address, network mask and IP-address for the default gateway.
- The host receives the answer and is configured with the information that it has received. I.e. a routing table is made based on the given information.

DHCP is described in RFC 2131 [4].

### 2.2.5.2 DNS (Domain Name Service/Space)

**DNS is a system for hierarchic grouping of hosts connected to the Internet. This hierarchy follows its own structure that can be different from grouping of IP-addresses or a geographical grouping of hosts. For the system there is connected a service that connects a host name in DNS and the IP-address.**

**URL's and domain names are important to SIP, since SIP uses this for addressing.**

To communicate with a machine on an IP-network (an intranet or Internet), we have to know the machines IP-address. The problem is that these 4-bytes addresses can be difficult to remember and does not say anything of who owns the given address. To simplify this, there is established a system for splitting all hosts on the Internet into a hierarchy with their catalogue service. This hierarchy is called Domain Name Space, and the service is called Domain Name Service. Both called DNS.

The hierarchy starts with a set of main domains that all points to sub domains. Since Internet started in the USA, these domains starts directly on different organisation areas, while in other countries the domain name starts with a code for that particular country. The existing root domains are therefore:

| Root-domains in USA | | | | | | Country codes, from Afghanistan to Norway, to Zimbabwe | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| com | edu | mil | org | gov | net | af | al | .... | no | .... | zw |

Below those root-domains, we find a lot of sub-domains, and in many cases there are several levels of sub domains. A name on a host in DNS first is written by the host-name, sub domain(s) and in the end the root domain. I.e.:

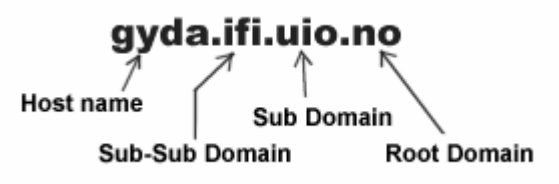


***Figure 16 DNS overview***

The catalogue service for DNS should now give me the IP-address of this host, so that I can use a service on this machine. We first assume that the sub-domain "*ifi*" itself has a DNS-server with a database of all hosts within this domain. What this database contains we will come back to later.

What happens is:

- In an application on my computer, I give a message that a package should be sent to the host with name"*gyda.ifi.uio.no*" (i.e. by starting a FTP session against this host). The problem is that my computer doesn't know the IP-address to"*gyda*."
- My application sends a query for the DNS-server that serves my machine. That is the DNS server for the zone my machine is a part of.
- This DNS-server does not know the IP-address for gyda.ifi.uio.no, so it sends the request on to the DNS-server for the root-domain for "*no*".
- The DNS-server does not know all hosts in the sub domain, but it knows where all DNS-servers to all sub domains in the root-domain "*no*" are. That is; it has the IP-addresses for all DNS servers. The request is forwarded to the DNS-server for "*uio.no*."
- Same procedure as for the root-domain "*no*": The DNS-server for uio.no does not know the host "*gyda.fif.uio.no*," but knows the DNS server for "*ifi.uio.no*". The request is forwarded to this server.
- The DNS-server for "*ifi.uio.no*" receives my request, finds the IP-address for "*gyda.ifi.uio.no*" and sends this to me, that is; the application started on my machine.

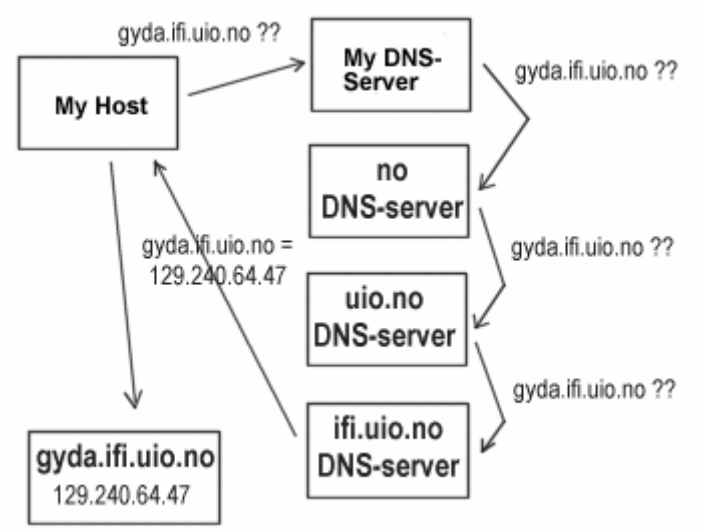


***Figure 17 DNS lookup***

After this lookup, the address and the name of "*gyda*" are stored on my local DNS server for a while, so that similar requests don't have to go all the way for similar lookups later.

In this example the sub domain had its own DNS server. It is not always adequate that all sub domains has one, and then it is needed to agree with the DNS server on level above are to serve requests on hosts in one or more sub domains as well. The domain and related sub domains that a DNS server shall cover is called a zone. In the hierarchy below the zone for ifi.uio.no is marked with grey.

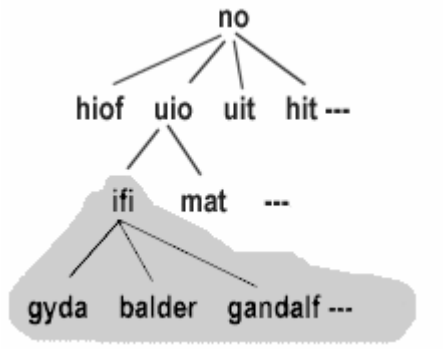


***Figure 18 DNS zones***

- The DNS-server also has another task in addition to address-lookups, and that is related to e-mail. We will not cover that here, but in short it is like this: The DNS-server are to keep track of which host within a zone that handles e-mail (mail-server) When an email is sent for a user in a domain, a DNS lookup is done to find out which host should receive the users message.

### 2.2.6 Virtual local area network (VLAN)

A virtual (or logical) LAN is a local area network with a definition that maps workstations on some other basis than geographic location (for example, by department, type of user, or primary application). The virtual LAN controller can change or add workstations and manage load balancing and bandwidth allocation more easily than with a physical picture of the LAN. Network management software keeps track of relating the virtual picture of the local area network with the actual physical picture.

There are several approaches to implementing a VLAN. One of these is described in an official standard, IEEE 802.1Q.

#### *2.2.6.1 The need for VLANs*

By the 1980's, most networks consisted of a simple, hierarchical arrangement in which multiple, shared-media networks were connected by a router [Figure 19]. With their sophisticated packet handling, routers allowed communication between networks when necessary, while effectively segmenting traffic so that large shared networks were not swamped by excessive traffic. Unfortunately, traditional routers were slow, complicated and expensive. As the need for faster networks emerged, a new solution was needed.

Switches spearheaded the next evolution of network structure. By segmenting the network and providing dedicated bandwidth where needed, they greatly increased performance, while reducing cost and complexity [Figure 20]. However, traditional switches segment only unicast, or node-to-node, traffic. Unlike routers, they do not limit broadcast traffic (packets that are addressed to all the nodes within the network) or multicast traffic (packets that are distributed to a group of nodes).

As networks have grown and traffic has increased, IT managers have been forced to segment their networks into more and more switched subnets to meet increasing performance demands. With these changes, broadcast and multicast traffic have placed a greater burden on network bandwidth. In the worst case scenario, broadcast traffic can spiral out of control, creating broadcast storms that can bring down the network.

As switched networks have become more common, routers have continued to exist within the network. But they've been forced toward the periphery, where speed is generally less critical. VLANs offer an effective solution to swamped routers and broadcast storms. By limiting the distribution of broadcast, multicast and unicast traffic, they can help free up bandwidth, reduce

the need for expensive and complicated routing between switched networks, and eliminate the danger of broadcast storms. With these advantages, VLANs revive many of the key advantages of LAN routing, but with greater flexibility, performance, simplicity and affordability.

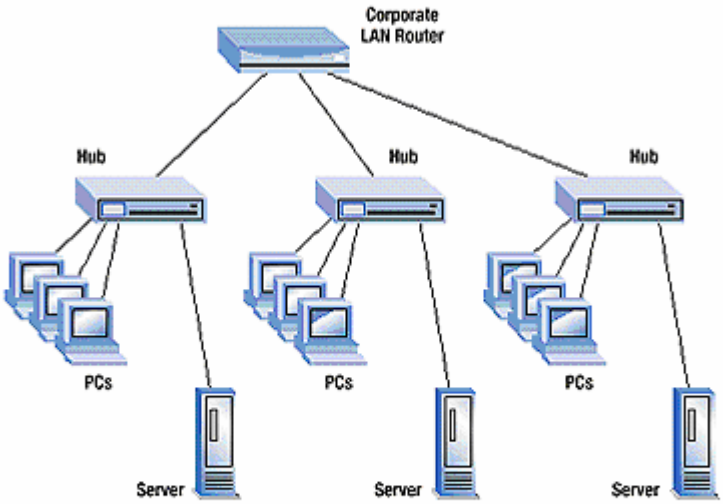


***Figure 19 Traditional fully routed network***

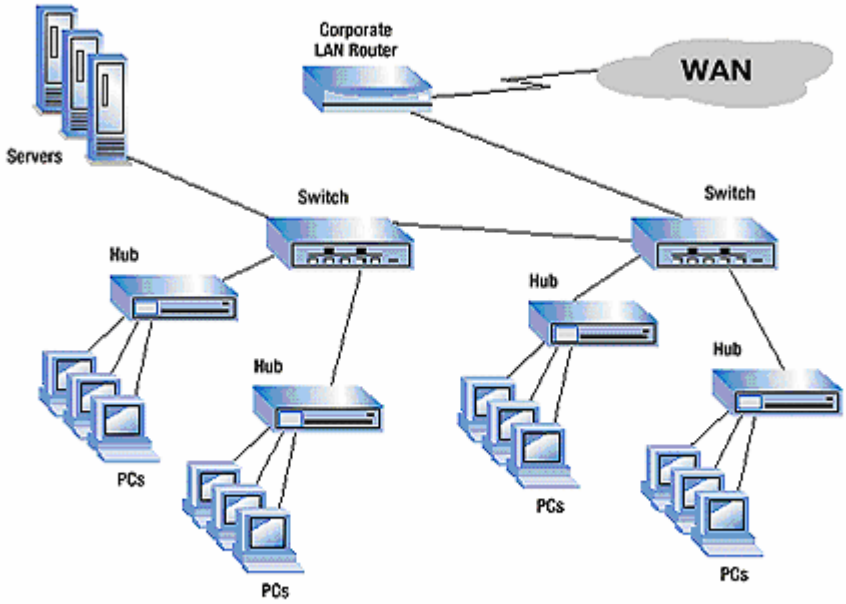


***Figure 20 Standard switched network***

### 2.2.6.2 Benefits of VLANs

**Flexible network segmentation**
Users and resources that communicate most frequently with each other can be grouped into common VLANs, regardless of physical location. Each group's traffic is largely contained within the VLAN, reducing extraneous traffic and improving the efficiency of the whole network.

**Simple management**
The addition of nodes, as well as moves and other changes can be dealt with quickly and conveniently from the management console rather than the wiring closet.

**Increased performance**
VLANs free up bandwidth by limiting node-to-node and broadcast traffic throughout the network.

**Better use of server resources**
With a VLAN-enabled adapter, a server can be a member of multiple VLANs. This reduces the need to route traffic to and from the server.

**Enhanced network security**
VLANs create virtual boundaries that can only be crossed through a router. So standard, router-based security measures can be used to restrict access to each VLAN as required.

### 2.2.6.3 VLAN Technology description

In general, there are three basic models for determining and controlling how a packet gets assigned to a VLAN.

#### 2.2.6.3.1 Port-based VLANs

In this implementation, the administrator assigns each port of a switch to a VLAN. For example, ports 1-3 might be assigned to the Sales VLAN, ports 4-6 to the Engineering VLAN and ports 7-9 to the Administrative VLAN [Figure 21]. The switch determines the VLAN membership of each packet by noting the port on which it arrives.

When a user is moved to a different port of the switch, the administrator can simply reassign the new port to the user's old VLAN. The network change is then completely transparent to the user, and the administrator saves a trip to the wiring closet. However, this method has one significant drawback. If a repeater is attached to a port on the switch, all of the users connected to that repeater must be members of the same VLAN.

#### 2.2.6.3.2 MAC address-based VLANs

The VLAN membership of a packet in this case is determined by its source or destination MAC address. Each switch maintains a table of MAC addresses and their corresponding VLAN memberships. A key advantage of this method is that the switch doesn't need to be reconfigured when a user moves to a different port.

However, assigning VLAN membership to each MAC address can be a time consuming task. Also, a single MAC address cannot easily be a member of multiple VLANs. This can be a significant limitation, making it difficult to share server resources between more than one VLAN. (Although a MAC address can theoretically be assigned to multiple VLANs, this can cause serious problems with existing bridging and routing, producing confusion in switch forwarding tables.)

#### 2.2.6.3.3 Layer 3 (or protocol)-based VLANs

With this method, the VLAN membership of a packet is based on protocols (IP, IPX, NetBIOS, etc.) and Layer 3 addresses. This is the most flexible method and provides the most logical grouping of users. An IP subnet or an IPX network, for example, can each be assigned their own VLAN. Additionally, protocol-based membership allows the administrator to assign non-routable

protocols, such as NetBIOS or DECnet, to larger VLANs than routable protocols like IPX or IP. This maximizes the efficiency gains that are possible with VLANs.

Another important distinction between VLAN implementations is the method used to indicate membership when a packet travels between switches. Two methods exist — implicit and explicit.

**Implicit —** VLAN membership is indicated by the MAC address. In this case, all switches that support a particular VLAN must share a table of member MAC addresses.

**Explicit — A** tag is added to the packet to indicate VLAN membership. Cisco ISL and the IEEE 802.1q VLAN specifications both use this method.

#### *2.2.6.4 Summary*

To summarize, when a packet enters its local switch, the determination of its VLAN membership can be port-based, MAC-based or protocol-based. When the packet travels to other switches, the determination of VLAN membership for that packet can be either implicit (using the MAC address) or explicit (using a tag that was added by the first switch). Port-based and protocol-based VLANs use explicit tagging as their preferred indication method. MAC-based VLANs are almost always implicit.
The bottom line is that the IEEE 802.1q specification is going to support port-based membership and explicit tagging, so these will be the default VLAN model in the future.

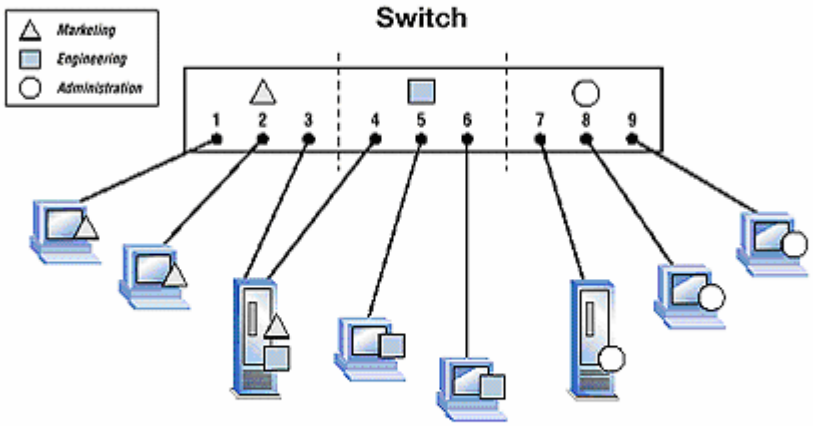


***Figure 21 Port-based VLAN***

### 2.2.7 Point to Point Protocol over Ethernet (PPPoE)

A TCP/IP network and its traffic can be compared to a network of city streets with vehicle traffic. There are many points at which a car can get on or off each street. Additional access points can be added with little disruption. But it is hard to tell how many cars are actually using each street. PPP can be compared to an overhead monorail. Travel is generally between two well-defined points; passengers can only get on and off at those points and need a ticket to board, making it relatively easy to count and monitor passengers. So PPP over Ethernet is similar to a monorail

running over the city street system. It offers speedy access between two well-defined points, and its traffic can be monitored.

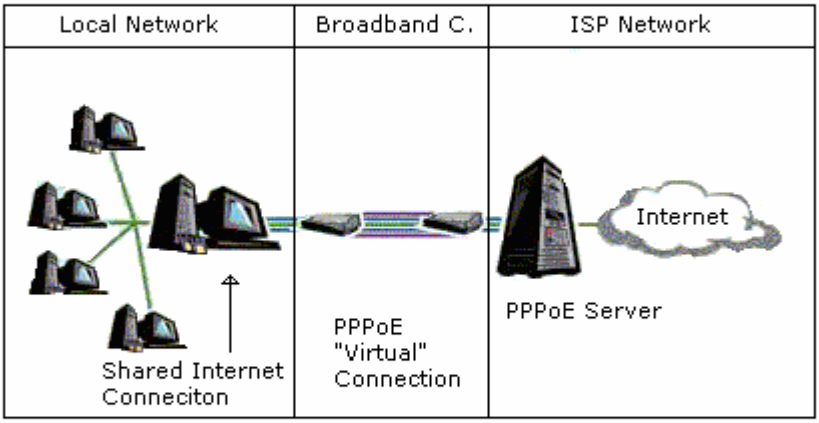


***Figure 22 PPPoE network model***

PPPoE provides the ability to connect a network of hosts over a simple bridging access device to a remote access concentrator. Each host utilizes its own PPP stack and the user is presented with a familiar interface. Access control, billing, and type of service can be done on a per-user, rather than a per-site, basis.

To provide a point-to-point connection over Ethernet, each PPP session must learn the Ethernet address of the remote peer, as well as establish a unique session identifier. PPPoE includes a discovery protocol to do this.

The PPPoE standard requires that PPPoE software place an additional header at the beginning of each TCP/IP packet. This may cause the packet to become larger than the maximum allowable size. Some software solutions handle this transparently, but some require modification of the TCP/IP settings on all of the client computers on the LAN.

PPPoE requires no more knowledge of the end-user than is required to set up standard dial-up Internet access. Additionally, PPPoE does not require any major changes in the operation model for service providers and carriers.

### 2.2.8 Transport Layer Security (TLS)

TLS is short term for Transport Layer Security. It is a protocol that provides communication security over the Internet. The purpose of TLS is to prevent eavesdropping, tampering and message forgery between client server communications. It is based on SSLv3 (Secure Socket Layer version 3), and is an IETF standard.

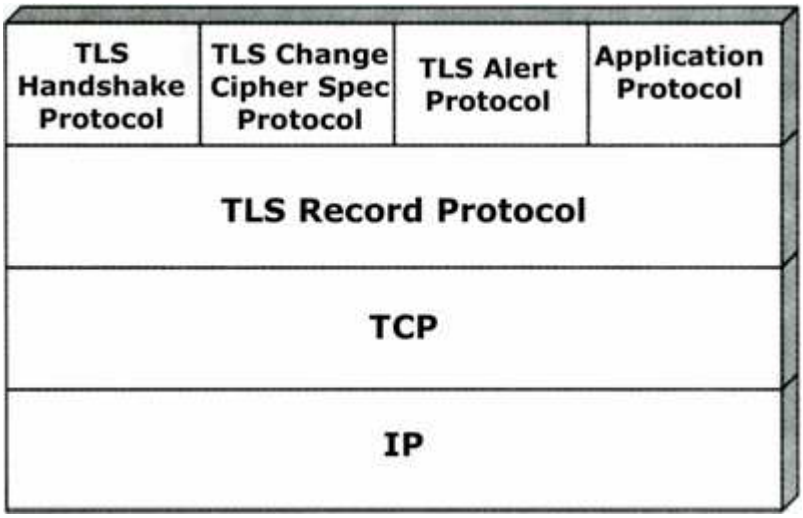


***Figure 23 Visualization of Protocol Stack***

The protocol is composed of two layers:

- The TLS Record Protocol
- The TLS Handshake Protocol, TLS Change Cipher Specification Protocol and TLS Alert Protocol

#### *2.2.8.1 TLS Record Protocol*

This protocol provides security with two basic properties:

- Connection Privacy. Symmetric cryptography is used for data encryption (e.g. DES, 3DES, RC4). Encryption can be turned off.
- The connection is securely reliable. Message transport includes a keyed cryptographic message authentication check (MAC).

#### *2.2.8.2 TLS Handshake Protocol*

This protocol allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data.

The procedure of the Handshake Protocol is given below [Figure 24 TLS Handshake Protocol Timeline.

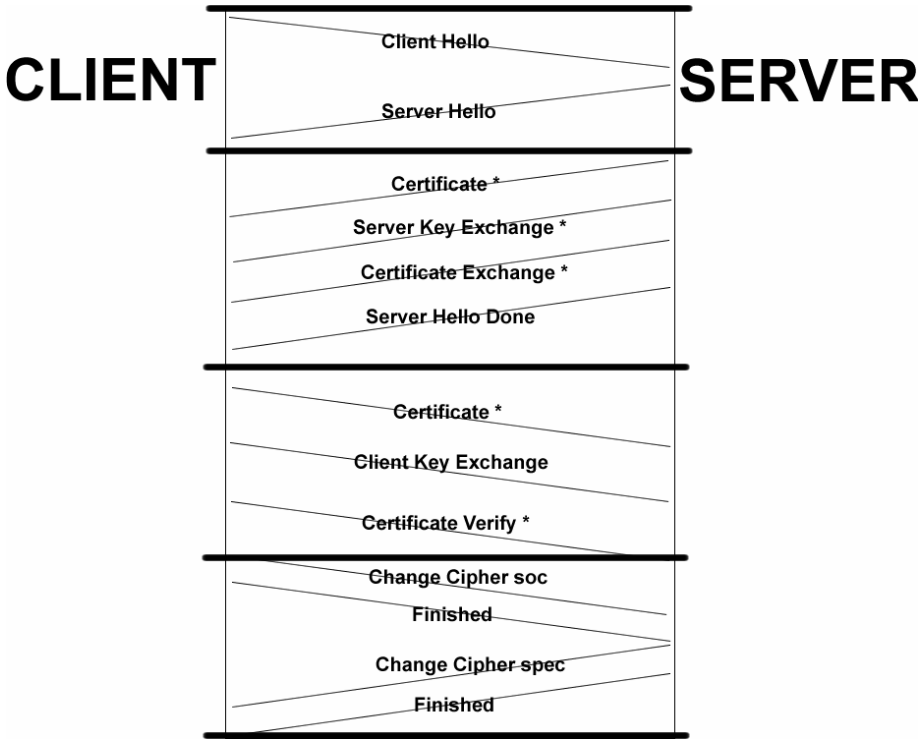


***Figure 24 TLS Handshake Protocol Timeline. Transactions marked with * are dependent on situation.***

#### *2.2.8.3 Security Level*

**Protocol stack layer**
The TLS protocol operates at the transport layer. It can provide end-to-end security for SNMP and other packets running on TCP.

**Authors Comments**
The TLS protocol was never tested since the agent entity (router) doesn't support TLS for in-band management traffic.

**Encryption algorithms supported**
TLS supports a broad range of encryption algorithms, e.g. DES, 3DES and RC4.

**Authentication method**
A TLS connection is securely reliable using a keyed cryptographic message authentication check (MAC) for authentication and data integrity.

**Key management**
The generation and distribution of session-keys is integrated into the protocol.

### 2.2.9 Internet Protocol Security (IPSec)

The internet protocol security (IPSec) takes place at the network layer. It provides security from peer-to-peer, independent of under- and overlying (e.g. TCP/UDP) layers

| RFC | Title | Date |
|---|---|---|
| 2401 | An overview of a security architecture | November 1998 |
| 2402 | Description of a packet authentication extension to IPv4 and IPv6 (AH) | November 1998 |
| 2426 | Description of a packet encryption extension to IPv4 and IPv6 (ESP) | November 1998 |
| 2408 | Specification of key management capabilities (ISAKMP) | November 1998 |

***Table 9 Key IPSec documents***

The IPSec is defined in the documents listed in Table 9. The architecture covers the general concepts, security requirements, definitions and mechanisms defining IPSec technology. The Authentication Header (AH) covers the packet format and general issues related to the use of AH for packet authentication. The Encapsulating Security Payload (ESP) covers the packet format and general issues related to the use of the ESP for packet encryption and optionally authentication. Key Management documents describe key management schemes.

**IPSec Services**
IPSec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to user for the service(s), and put in place any cryptographic keys required to provide the required the requested services.

Two protocols are used to provide security: an authentication protocol designated by the header of the protocol, Authentication Header (AH); and a combined encryption/authentication protocol designated by the format of the packet for that protocol, Encapsulation Security Payload (ESP). The services are as follows:

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed paces (a form of partial sequence integrity)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

The differences between AH and ESP in these services is that:
1. AH doesn't have confidentiality and traffic flow confidentiality
2. ESP with encryption only, doesn't have connectionless integrity and data origin authentication.
3. ESP with both encryption and the optional authentication has all six services.

**Security Associations**
Security Associations is a key concept that appears in both the authentication and confidentially to mechanisms for IP. An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it. If a peer relationship is needed, for two-way secure exchange, the two security associations are required. Security services are afforded to an SA for the use of AH or ESP, but not both.

A security association is uniquely identified by three parameters:
- Security Parameters Index (SPI): A bit string assigned to this SA and having local significance only. The SPI is carried in Ah and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
- IP Destination Address: Currently, only unicast addresses are allowed; this is the address of the destination endpoint of the SA, which may be an end user system or a network system such as a firewall or router.
- Security Protocol Identifier: This indicates whether the association is an AH or ESP security association.

Hence in any IP datagram, the security association is uniquely identified by the destination address in the header and the SPI in the enclosed extension header (AH or ESP).

In every IPSec implementation there is a security association database that defines the parameters associated with each SA. A SA is normally defined by the following requirements:

- Sequence Number Counter: a value for generating sequence number
- Sequence Counter Overflow: Flag indication overflow of the sequence number counter.
- Anti-Replay Window: used to determine whether incoming datagram's are replays.

- AH Information: Authentication algorithm, keys, key lifetimes, and related parameters being used with AH.
- ESP Information: Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP.
- Lifetime of this Security Association: A time interval after which an SA must be replaced with a new SA or terminated.
- IPSec Protocol Mode: Transport or tunnel mode, discussed in coming section, or a wildcard.
- Path MTU: Maximum Transmission Unit before datagram fragmentation, and aging variables.

**Transport and tunnel modes**
Both AH and ESP supports two modes of use: transport and tunnel mode.

Transport mode provides protection primarily for upper-layer protocols. Its protection extends to the payload of an IP datagram. It is typically used for end-to-end communication between two hosts (e.g. manager and agent)

Tunnel mode provides protection to the entire IP datagram. To achieve this, after the AH or ESP fields are added, the entire datagram plus security fields is treated as the payload of a new 'outer' IP datagram with a new outer IP header. The entire original datagram travels trough a tunnel from one point of an IP network to another. No routers along the way are able to examine the inner IP header. It is possible to set up a tunnel e.g. between two routers and let all the traffic between hosts behind these gateways be encrypted. This is a normal way to protect all traffic e.g. between remote offices in an enterprise, making a so called virtual private network (VPN) in a public network.

### *2.2.9.1 Authentication Header (AH)*

The Authentication Header (AH) provides support for data authentication, data integrity, and optional anti-replay services. AH is embedded in the data to be protected (a full IP Datagram). The authentication is based on the use of a message authentication code (MAC), as described in previous chapters. Hence two parties must share the same key and support one common hash code (MD5 or SHA-1)

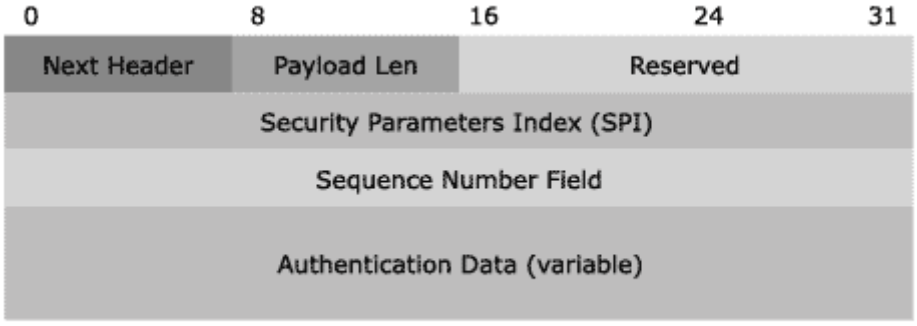


***Figure 25 IPSec Authentication Header (AH)***

The AH header [Figure 25] contains the following fields:

- *Next Header* - Identifies the IP payload by using the IP protocol ID. For example, a value of 6 represents TCP.
- *Length* - Indicates the length of the AH header.
- *Security Parameters* Index (SPI) - Used in combination with the destination address and the security protocol (AH or ESP) to identify the correct security association for the

communication. The receiver uses this value to determine with which security association the packet is identified.
*Sequence Number* - Provides anti-replay protection for the packet. The sequence number is a 32-bit, incrementally increasing number (starting from 1) that indicates the packet number sent over the security association for the communication. The sequence number cannot repeat for the life of the quick mode security association. The receiver checks this field to verify that a packet for a security association with this number has not already been received. If one has been received, the packet is rejected.

- *Authentication Data* - Contains the integrity check value (ICV), also known as the message authentication code, which is used to verify both message authentication and integrity. The receiver calculates the ICV value and checks it against this value (which is calculated by the sender) to verify integrity. The ICV is calculated over the IP header, the AH header, and the IP payload.

**Transport and tunnel mode**
An Authentication Header can be used in both transport and tunnel mode. Transport mode is normally only used for communication directly between hosts. Tunnel mode is used for other situations such as between host and subnet or subnets in between.

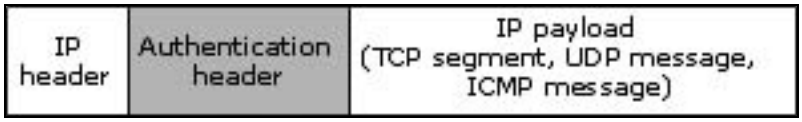


***Figure 26 AH in transport mode***

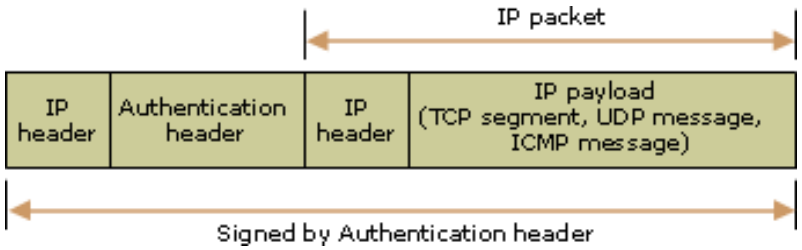


***Figure 27 AH in tunnel mode***

### 2.2.9.2 Encapsulating Security Payload (ESP).

The Encapsulating Security Payload (ESP) provides data confidentiality, data integrity, and protection services, optional data origin authentication, and anti-replay services.

The ESP header contains the following fields:

- *Security Parameters* Index (SPI) – Identifies the security association.
- *Sequence Number* – A monotonically increasing counter value; this provides the same anti-replay function as for AH.
- *Payload Data*: This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.
- *Padding*: The purpose of this field
- *Pad length*: Indicates the number of pad bytes immediately preceding this field
- *Next header*: Identifies the next header in the payload data.
- *Authentication Data*: A variable length field that contains the integrity check value (ICv) computed over the ESP packet minus the Authentication Data field.

**Transport and tunnel mode**

The Authentication Header can also be used in both transport and tunnel mode. Transport mode (Figure 28) is used for communication directly between hosts, while tunnel mode (Figure 29) is used for other situations such as between host and subnet or subnets in between.

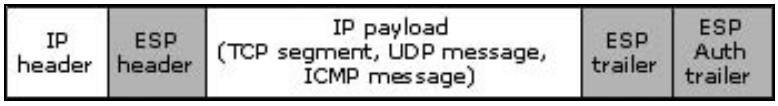


***Figure 28 ESP in transport mode***

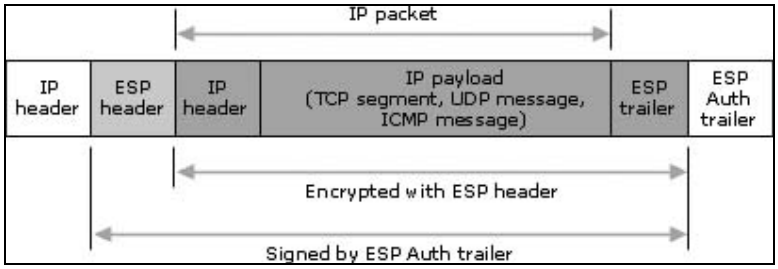


***Figure 29 ESP in tunnel mode***

### *2.2.9.3 Key Management*

The key management part of IPSec involves rules for distribution of secret keys. The IPSec architecture defines support of two types of key management:

- Manual: A system administrator configures each system manually with algorithms, own keys and with keys from other communicating systems.
- Automatically (ISAKMP/IKE): Protocols for automatic key and session management that makes it possible to configure security associations (SA) with keys on demand, and simplifies the use of keys in expanding distributed systems.

ISAJMP/IKE is the standard method for key management in IPSeec, and consists of the following elements:

- Internet Security Association and Key Management Protocol (ISAKMP): This is a protocol framework that defines formats of payload, mechanisms for implementation of protocols, for key exchange and negotiation of security associations.
- Internet Key Exchange (IKE): IKE is a protocol for key exchange that is implemented within the ISAKMP framework. IKE ensures authentication of IPSec communication, negotiation of IPSec keys, and negotiation of IPSec security associations.

**ISAKMP**

ISAKMP does not itself specify any particular algorithms for key exchange; ISAKMP rather consists of a set of message types that supports the use of algorithms for key exchange.

The specification defines five standard message exchanges that is supported:

- Base exchange: allows key exchange and authentication material to be sent together, something that minimizes the number of exchanges, at the cost of not provide identity protection.
- Identity protection exchange: Extends the base exchange to also protect the users identity.
- Just authentication exchange: used to perform mutual authentication without key exchange.
- Aggressive exchange: minimizes the number of exchanges at the cost of not supporting identity protection.

**IKE**
Internet Key Exchange (IKE) is a hybrid of the two key exchange protocols Oakley and Scheme, that defines how authentication material is created, wher Scheme in addition also defines a method for fast key update.

The IKE algorithm is characterized by five important features:
1. It uses a mechanism known as cookies to prevent clogging attacks.
2. It makes it possible for the two parties to negotiate a group; this specifies in short terms the global parameters for the Diffie-Hellman key exchange
3. It uses nonces to protect from attacks.
4. It makes exchange of values for the Diffie-Hellman keys possible.
5. It authenticates Diffie-Hellman exchanges to prevent from man in the middle attacks.

Three different authentication methods can be used with IKE:
- *Digital signatures*: Exchange is authenticated by signing a mutual gettable hash; each part encrypts the hash with theyr provate key. The Hash is generated through important parametres, such as user-id and nonces.
- *Public key encryption*: the exchange is automated by encryption parameters, such as ID's and nonces with the senders private key.
- *Symmetric Key Encryption*: A key derived from a outer mechanism can be used to authenticate the exchange with symmetric encryption of exchange parameters.

#### *2.2.9.4 Security Level*

**Protocol stack layer**
The IPSec operates at the network layer. It provides host-to-host security for SNMP and other messages.

**Encryption algorithms supported**
There is a broad range of encryption algorithms supported by the IPSec ESP method. Our test scenario however only has one of these available. The Cisco agents supports both DES (56-bit) and triple DES (192-bit) encryption, but our FreeS/WAN IPSec implementation only supports 3DES. The project group feels that the Single DES algorithm is too weak and therefore refuses to support it. I must admit that I still tried to install a side-stream implementation called SuperFreeS/WAN but had troubles installing and configuring this properly. With it I would have gotten support for many other algorithms including Single DES and the null encryption algorithm. Hopefully we will also see the Advance Encryption Standard (AES) as it is implemented into the Cisco IOS Software.

**Authentication method**
The authentication options available for both IPSec AH and ESP(optional) are the same as for i.e. SNMPv3. The HMAC-MD5 uses a 128-bit key and the HMAC-SHA1 uses a 160-bit authentication key.

**Key management**
The key management scheme for IPSec is perhaps one of its best advantages. With the automatic keying negotiation of the Oakley protocol combined with a Public-Key Infrastructure system, this provides very easily manageable keying method.

**Authors Comments**
The transport mode of IPSec operation provides manager-host to agent-host security. This is the operation mode that is most relevant for SNMP traffic, protecting the datagram's all the way.

### 2.2.10 Secure Shell (SSH)

SSH Secure Shell is a replacement for the Berkley 'r' commands, telnet ftp functionalities. Introduced in the BSD (Berkley) version of UNIX, the 'r' commands (rsh, rlogin, rcp) allow you to communicate with remote systems. Although they're not as extensive as ftp and telnet, the 'r' commands are useful tools for extending the reach of your network.

SSH functions as a type of tunnel for encoding login procedures. All connections between the local and the remote hosts are encrypted, protecting the data sent between these machines. Secure Shell provides several security improvements over the telnet, ftp and rlogin protocols. In particular, passwords are never sent over the network in a clear text format as they are when using telnet, ftp or rlogin. This encryption makes it difficult for someone to breach the confidentiality of your data and/or compromise passwords. SSH Secure Shell is based on the SSH2 protocol that is standardized by IEFT in a draft format (www.ietf.org/ids.by.wg/secsh.html).

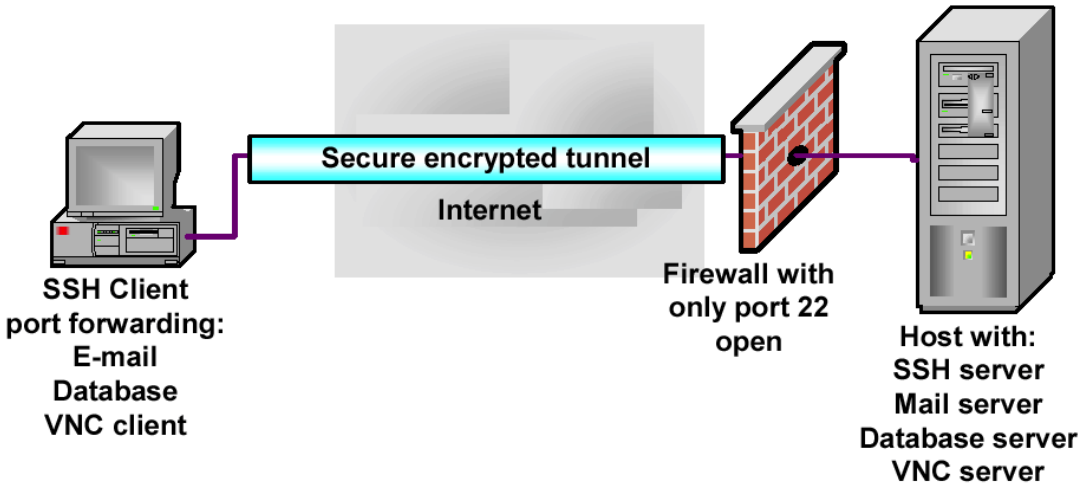


***Figure 30 SSH Secure encrypted tunnel***

Secure Shell does not close all network security holes of course, but it is one step toward a more secure network.

#### *2.2.10.1 How SSH works*

The SSH2 login procedure can be illustrated in the following client/server model.

Note: SSH1 uses server and host keys to authenticate systems where SSH 2 only uses host key. A server key (768 bit) is generated every hour by default and is not saved in a file. The server key ensures the encoded data can no longer be decoded once the server key, after an hour, has been regenerated (in the case that the private key is ever compromised).

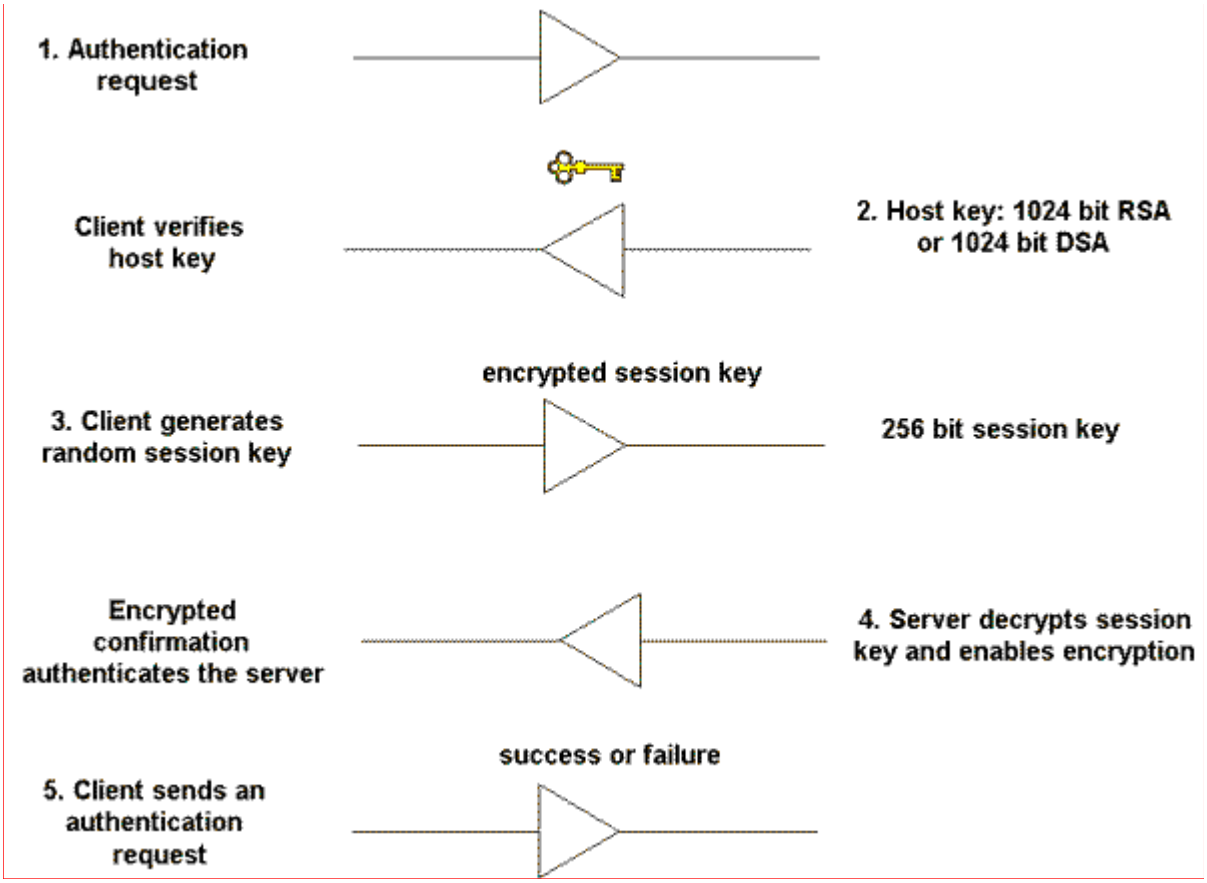


***Figure 31 SSH Key Introduction***

1. The client makes a connection to a server.
2. The server identifies itself wit its public host key. The length is 1024 bit RSA or DSA. The client looks in its local database to verify the public host key is authentic / known. An unknown key is added to the database or the session can be broken. If the client determines the host key does not belong to the server, the client is alerted (SSH generate a warning).
3. The client ten generates a random 256-bit number and chooses an encryption algorithm (e.g. 3DES). The random number is then encoded with RSA or DSA. Pure RSA / DSA authentication never trusts anything but the private key. The encoded key is sent to the server. The host key ensures the authentication of the particular server.
4. The server decodes the RSA / DSA encryption and reconstructs the session key. Furthermore, the server sends the client, via the encoded session key, a confirmation. The rest of the session is encrypted using a symmetric cipher.
5. The client then sends a username authentication request. The server replies with a success or failure.

### *2.2.10.2 SSH Features and Specifications*

**Authentication methods**

- Host-based authentication: password protection with 1024 bit RSA-key (public-key cryptographic algorithm).
- Ability to add certificate and public key authentication.
- Password authentication
- Public key algorithm support: DSA and Diffie-Hellman key exchange.

- PGP key support

**Data encryption for confidentiality and integrity**

- Encryption algorithms: DES, 3DES Blowfish, Twofish, Arcfour, CAST128-CBC, 128 bit AES or 256 bit AES.
- Hash Algorithms: MD5 and SHA1

**Additional functionality**

- File transfer. Files can be copied remotely using a utility called scp (secure copy).
- SFTP (secure file transfer protocol) was introduced with SSH2. SFTP-server is not called directly but through the SSH2 daemon – the SSH2 protocol therefore secures the connection. See www.ietf.org/ids.by.wg/secsh.html for details.
- TCP/IP port forwarding: SSH supports port forwarding over a secure tunnel. You configure your SSH client to accept connections on the local machine for certain ports. Any data that is sent to these ports is then forwarded and returned across the tunnel. On the other side of the tunnel, the SSH server passes the data back and forth to a server you wish to access. Typical implementations of port forwarding are services that have no encryption on their own built-in such as FTP, IMAP, SMTP and POP3. Note: only root can redirect privileged ports.
- X11 connections for secure X Window System sessions are a popular implementation. SSH Communications Security offers special support for this feature because it is so popular. SSH creates a fake X server (Fake Xauthority information) on the same machine that the SSH client is run. SSH then functions as a go-between between the connection and forwards it to a real X server over a secure connection.
- TCP Wrapper support. The TCP Wrappers is a public-domain tool/package you can monitor and filter incoming requests for the SYSTAT, FINGER, FTP, TELNET, RLOGIN, RSH, EXEC, TFTP, TALK, and other services. It is used to restrict inbound network access to the services defined in the /etc/inetd/inetd.conf file. TCP Wrappers is controlled by two configuration files: /etc/hosts.allow and /etc/host.deny. If they do not exist or are empty, TCP Wrappers will allow all hosts to access all services.
- Built-in SOCKS 4 and 5 support for traversing firewalls.
- Support for SSH1 fallback functionality. The SSH1 and SSH2 protocols are not compatible. SSH1 clients cannot connect to a SSH2 daemon. The SSH2 daemon can be configured to start up a SSH1 daemon for a SSH1 client. (Must be installed with compatible option). The SSH1 and SSH2 daemon can be running on two separate machines or on the same machine.
- Multiple channel support. "All terminal sessions, forwarded connections, etc, are channels. Either side may open a channel. Multiple channels are multiplexed into a single connection"
- Distributed key management. The advantage of distributed key management is that there is no hierarchy. This avoids having a single target for attacks as every machine can hold its own keys. It is also possible to configure these machines to change session keys every hour.

### *2.2.10.3 Popular SSH clients*

SSH Communications Security. URL: www.ssh.com
F-Secure. URL: www.f-secure.com
Van Dyke Technologies. URL: www.vandyke.com
Open SSH. URL: www.openssh.com

For a complete list of free and licensed clients, visit the www.freessh.org website.

#### 2.2.10.4 SSH Security issues

**General**
TCP/IP was not designed with focus on security. That is why it is important to implement security techniques such as SSH Secure Shell.

SSH2 connections are encrypted, protecting the data sent between machines. This encryption makes it difficult for someone to breach the confidentiality of your data and/or compromise your passwords. Typical attacks that SSH protects against are: Trojan horses, DNS spoofing and Man-in-the-middle attacks.

**Key Management**
SSH2 offers the following solution to prevent public key substitution:

- SSH2 automatically maintains and checks a database containing public keys of hosts
- When logging on to a host for the first time, the host's public key is stored to a file in the user's personal directory
- If a host's identification changes, SSH2 issues a warning and disables password

#### 2.2.10.5 SSH advantages and disadvantages

**SSH Advantages**

- SSH is secure – it protects against a wide variety of potential security breaches. It solves the most important security problem on the Internet: hackers stealing passwords easily
- Interoperability – there are a wide variety of SSH clients that run on Linux, Macintosh, UNIX, Windows and Open VMS that work with the SSH server. Also, a user can select from a wide variety ciphers to use for strong encryption
- Support for X-11 Forwarding – provides an encrypted X-11 display
- SSH is a flexible protocol – using the port forwarding feature, any application that has a static port assignment, such as POP, SMTP, and Oracle database connections can be encrypted. A system administrator may only choose to encrypt his telnet session and not POP
- End-to-end security – SSH encrypts the connection from the source system to its destination. This provides security on the public and internal corporate network. It can also restrict the user from accessing any other system on the internal network other then the target system.
- SSH2 will be an IETF standard and SSH1 is a de-facto standard for encrypted terminal connections and secure file transfers

**SSH Disadvantages**

- SSH cannot secure (encrypt) all applications, only if there is a known port number. Therefore applications such as NFS can not be protected using SSH port forwarding.

## 2.3 Cryptography

We will now talk about Cryptography in general, as a brief introduction to Cryptography and its main methods. The main effort will focus on how keys relate.

### 2.3.1 Introduction to Cryptography

Cryptography is a branch of mathematics that has powerful implications for data security. The basic principle of cryptography is that some math problems are computationally expensive, which is a fancy way of saying they take a long time to solve. Cryptography relies on the use of keys. A key is a number that makes it easy to solve a math problem. By keeping the key a secret, you can build a system that protects data from people who don't have the key.

Cryptography is one tool in the belt of the security architect. It's a big tool, and an effective one, but adding cryptography to a system is useless unless every other part of the system is also made more secure. If you don't lock the door to the server room, all the cryptography in the world won't help.

### 2.3.2 Ciphers and Keys

A cipher is an algorithm useful for keeping data confidential. It can translate between regular data, called plaintext, and an encrypted form of the data, called cipher text. Essentially, the cipher is an equation that takes one number (the plaintext) and makes it into another number (the cipher text).

Most ciphers use keys to encrypt and decrypt data. Keys are just numbers that are used in the cipher's equation. Different keys produce different cipher texts from the same plaintext.

Ciphers provide *confidentiality* for data because it's extremely hard for attackers to decrypt the cipher text without the right key, even if they know the algorithm.

As an example, you can use a cipher to encrypt credit card information on its way from the wireless client to the server. Even if an attacker intercepts this information, either from the air or using wired packet sniffers, decrypting it will be prohibitively difficult without the right key.

There are **two types of ciphers, symmetric and asymmetric**. A *symmetric* cipher uses a single key for both encryption and decryption. Two people with the same key on opposite sides of the Internet can use a symmetric cipher to send encrypted messages to each other. Symmetric cipher keys are sometimes called secret keys or private keys. Despite their usefulness, symmetric ciphers can be tricky because both people using the cipher must have the same key. One person can generate the key, but it must be safely transmitted to the other person.

*Asymmetric* ciphers use a key pair, two keys that are related to each other. One is a public key; the other is a private key. Data encrypted using one key can be decrypted using the other key. The public key can be freely distributed without compromising security; the private key must be kept private. Imagine how paired keys might work in practice: Someone wanting to send you a secret message can encrypt it using your public key and send the cipher text to you. Assuming you haven't let anyone steal your private key, you are the only person who can decrypt the cipher text.

Asymmetric ciphers are useful for *authentication (*means to prove identity). Anyone sending you a message encrypted with your public key is sure that only you can decrypt the message, so long as you keep your private key hidden. You are effectively authenticated to the sender. Asymmetric ciphers work the other way around, too. If you encrypt a message with your private key, anyone decrypting it with your public key is assured that you originated the message, because only you possess your private key. Here you have authenticated yourself to the recipient. If the recipient uses your public key to decipher a message from anyone lacking your private key, the result is gibberish.

The math for asymmetric ciphers is more complicated than for symmetric ciphers, so symmetric ciphers usually run faster. Encrypting large messages using an asymmetric cipher usually takes too long. A hybrid approach is sometimes useful, where two systems use an asymmetric cipher to agree on a symmetric cipher key. They then use a symmetric cipher and this session key for the remainder of the interaction.

*Common* cipher algorithms are DES, Rijndael, Blowfish, and ElGamal. Keys are specific to cipher algorithms; if you are using a Rijndael cipher, you have to have a Rijndael key. Many algorithms

can use keys of different lengths, commonly measured in bits. Longer keys are slower to use than shorter keys but the cipher text they produce is harder to break.

**Where Do Keys Come From?**

Keys can be generated from *random numbers*. The public and private keys in a key pair are mathematically related to each other but can be generated randomly. "Random" is a dubious word in this context. Computers are surprisingly bad at finding random numbers. Most use a pseudo-random number generator (PRNG), which produces a repeatable sequence of bits. Use two PRNGs, initialized identically, and you'll get exactly the same sequence of numbers. What good is a supposedly random key if an attacker can use the same PRNG to determine its value?

Another way to generate keys is to use a *key agreement protocol*. This is a clever mathematical trick two parties can use to agree on a session key. Neither of the parties needs prior knowledge of the other, and eavesdroppers who listen to the entire exchange will still be unable to determine the value of the session key.

The most common key agreement protocol is Diffie-Hellman. A key agreement protocol is used by SSL and TLS.

### 2.3.3 Message Digests

A message digest is used to create a "fingerprint" of a piece of data. It takes an arbitrarily large message or file and mashes it down into a short, "digested" version, called the message digest value. Change just one bit of the original message and the digest value will be entirely and unpredictably different. You can use message digests to assure data integrity. When you download a file from a server, you can compute its digest value and compare it to the value computed by the server. If the two are the same, you can be sure that the file has not been modified on its way to you.

Common message digest algorithms are MD5, SHA-1 and RipeMD.

**Signatures**
You use your handwritten signature to guarantee the validity of checks, contracts, and other documents. Digital signatures perform the same function, more reliably, on electronic documents. Supply a message and a private key (the signing key) to a digital signature algorithm, and out pops a number that, in essence, is an encrypted message digest value. This signature is unique to your private key and the message itself.

Suppose you sign a file, send the file and the signature to your friend. She can use your public key and the message itself to verify your signature. She uses your public key to decrypt your signature, which gives her the digest value you computed. For comparison, she then computes her own digest value for the message. If her value matches yours, she knows that she received the file exactly as you sent it. (This process is a good way to conceptualize the verification of a digital signature, but the steps may not be explicit in practice.) If an attacker intercepts the file and modifies it, the message digest values won't match. He can't simply create a new signature for the modified file because he doesn't have your private key.

Common signature algorithms are DSA and RSA.

As in the sections above (*Message Digests & Signatures*) an application could use digital signatures to authenticate users to the server. Suppose the user's private key is stored on a device and the corresponding public key is stored on a server. Knowing the public key, the server can verify the signature and trust the user's identity. Note that anyone who steals the device is also stealing a private key. A runtime password challenge would make it harder for the thief to use the application.

### 2.3.4 Certificates and Key Management

All these discussions of ciphers and signatures neatly sidestep the ugly monster in cryptography's closet: key management. How do you find someone's public key? Where do you keep your private key? Suppose someone you don't know, “Pablo”, sends you a message with a signature. You need to get his public key to verify the signature. How do you get it? How do you know you've got “Pablo's” real public key and not a fake?

A cryptographic certificate offers one solution. A certificate is a container for a public key. It's an electronic document that says something like "Violet certifies that Pablo's public key has this value". The certificate would contain information about Violet, information about “Pablo”, and the value of “Pablo's” public key. The whole thing would be signed by “Violet”. The certificate allows for extension of trust. If you know “Violet's” public key, and if you think she's reliable and a good judge of character, and if you can verify the certificate signature, then you can be pretty sure that “Pablo's” public key has the value contained in the certificate.

You've really only shifted the problem, however. Fine, “Violet's” public key verifies “Pablo's” public key, but who verifies hers, and how do you get it? Where's the end of the chain? The answer is a self-signed certificate, a certificate asserting the value of a public key, signed using the corresponding private key. This kind of certificate is called a root certificate, and companies or institutions that use them to sign other certificates are called certificate authorities (CAs). CAs generates their own root certificates and distributes them as widely as possible. The problem with self-signed certificates is that anyone can generate one, claiming to be the U.S. Post Office or the King of Norway. Trust in root certificates is based on the fact that they are widely published, making them hard to spoof. If you have a root certificate in hand, you should be able to verify its validity by comparing its signature to the signature published on the CA's web site.

Certificates can be assembled in chains, a ladder of verification starting at the bottom and ending at a CA. For example, “Pablo” could send you a message and a certificate chain consisting of the following certificates:

“Pablo's” certificate, signed by “Violet”
“Violet's” certificate, signed by “Henry”
“Henry's” certificate, signed by the “King of Norway”
Assuming you're sure you have the King of Norway's genuine root certificate, you can verify the whole chain of certificates, assuring you that “Pablo's” public key is authentic.

Key management is a matter of keeping track of your private key (or keys), and of managing certificates from other people and companies, including root certificates you can use to verify certificate chains.

The de facto standard for certificates is X.509v3. (Authentication Service set recommended by ITU)

The example mentioned could use certificates in several places. First, the server needs users to authenticate themselves, to verify they are paid subscribers entitled to request services. The client device can send a message, the user's signature of the message, and a certificate chain to the server, enabling the server to authenticate the user.

On the flip side, the client may well wish to authenticate the server to make sure it's not talking to an attacker instead. The server can cooperate by following a similar strategy, sending the client a signed message and its certificate chain. Another possibility: a client that already has the server's certificate embedded in the application can verify signed messages sent by the server immediately.

### 2.3.5 Summary

In this section, we have talked about how you can build security into the system to balance the value of its contents with the cost of breaking it. Cryptography is a powerful tool for security. It includes ciphers for encrypting and decrypting data, signatures for assuring data integrity, and certificates for authentication.

# 3 Scenario

Our test scenario will be the fibre network of Èlla Kommunikasjon AS. In this chapter you will learn how this network is configured, what services that run in this environment, both telephony related servers and equipment, and other services. The focus will be on IP-telephony related equipment. We start with the description of the service bearer (network), and afterwards we will look into how the IP-telephony is organized.

PS / Additional: Èlla Kommunikasjon AS operates with several networks. That is one for corporate clients and one for private customers plus several others. The network we will discuss is the private network, since this is the current running IP-telephony pilot network. There are normally only small changes between core configuration of these different networks, so what applies for one – normally also apply for the others.

## 3.1 Overview

### 3.1.1 Network

Èlla Kommunikasjon has a network based on layer 2 switches and fibre + TP cables for traffic. Below is a simplified model of how the network looks.

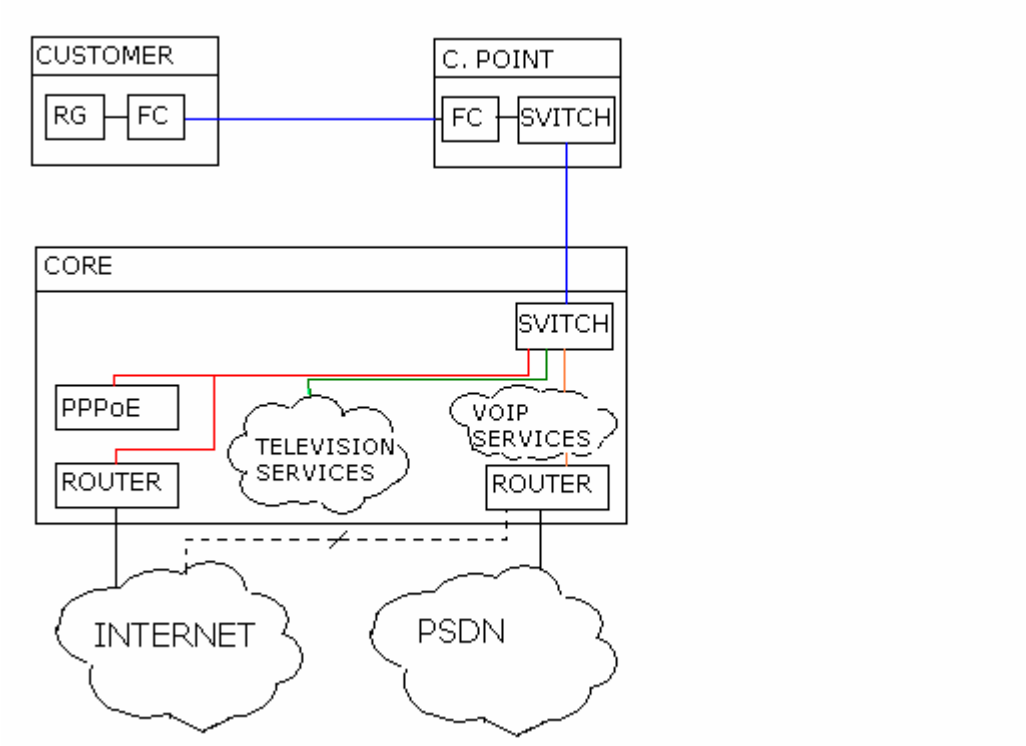


***Figure 32 Simplified Network Model - Èlla Kommunikasjon AS***

#### *3.1.1.1 Customer*

We will look at the physical network, and start at the block called "CUSTOMER." This block is geographically limited to the actual house of a customer. The configuration of a customer consists of a fibre cable into his house (blue line from C.POINT). This cable is terminated in an Allied Telesyn MC103XL fibre optical converter (FC). From this point there is a TP cable going to an Allied Telesyn RG 213 Residential VoIP Gateway (or similar model by same manufacturer). This is called a RG in the figure. This RG works as a switch, and has 5 ports + WAN port. The configuration of this switch will be discussed later.

#### 3.1.1.2 Connection Point

The next geographic point is the local connection point (C.POINT). This is typically a connection place (rack or cabins), where all customers in an area are connected. From customer side a fibre cable goes from the fibre converter (FC) to an equal converter at the local connection point. This is typically mounted in racks of 12 or more converters, all corresponding to a single household somewhere in the local area. From this converter rack each converter has its corresponding port on a HP Procurve Series switch. These switches are working on layer 2 in the OSI model. For further information generally on switches, look at the theory chapters above [2.2 and especially 2.2.6.1]. HP Procurve switches will be discussed in detail later. We now have all customers in a geographic area connected to a single switch. In the switch, we have a gigabit fibre-module that connects the connection point (C.POINT) to the core of the network by fibre cable. There is a need to mention that the model is simplified, and that there can be several levels of connection points before you hit the core. This is often hierarchic designed. I.e. if the core is the city of "Bigplace", there can be a connection point connected to the core in a county of "Bigplace" called "Medium place", and another connection point "Small place" within "Medium place" connected to "Medium place," and routed on through "Medium place" to hit the core.

#### 3.1.1.3 Core

When we hit the core, traffic is switched to its representative service. This is done by filtering traffic with VLAN technology [2.2.6]. This will also be discussed in detail later. Let us for now just assume that the core switch knows where to send the packages. Packages are filtered by category that is Multicast / Unicast television packages, Internet packages or VoIP packages. In our case VoIP = IP-telephony related packages.

Internet traffic (here HTTP, POP3 and similar) can not take place until the user has logged on the network by using his PPPoE client [2.2.7]. When logged on, the traffic will be routed to the Internet, as described in [2.2.4].

The initial Television packages follow somehow same way to operate as the Internet traffic. An IP compliant tuner logs onto an authentication unit, and has to identify itself. When this is done, a registry keeps track of which channels and services the user has access to. The package stream for Television follows a Multicast protocol. We will not describe this service further in this report.

Our main interest is of course the IP-telephony service. The simplified network model has following information related to this service within the core:
VoIP packages enter the core, get filtered in the switch, and hit the VoIP server. It is not completely correct to say that all VoIP packages hit the server. The only packages hitting the server are typical what is known as signalling in ISDN. When ever you establish a phone call, signalling is sent between your client equipment, the server and the receiver. The data (sound, images and so on) is sent directly to the client you talk to; Peer to Peer. When making a phone call, the server first tells you to identify yourself. When that is done, the request for a call is made through the server, and the receivers address is returned to you. From the model you can see that a VoIP server is connected to a router. The reason for this is that if a receiver does not exist in the VoIP server client database, the request is forwarded to the router and from there routed out on the PSTN. This makes it possible to communicate between SIP phones and analogue/ISDN equipment outside the local network. From the model it is also possible to see that the routers connection to the Internet is blocked. That has been done in Èlla Kommunikasjon for security reasons. To make this simple, you can say that all clients not existing in the server database exists in the PSTN. Why Internet connection has been shut will be discussed later.

### 3.1.2 IP-telephony traffic

Traffic of VoIP in Èlla Kommunikasjon network follows the 3 models below.

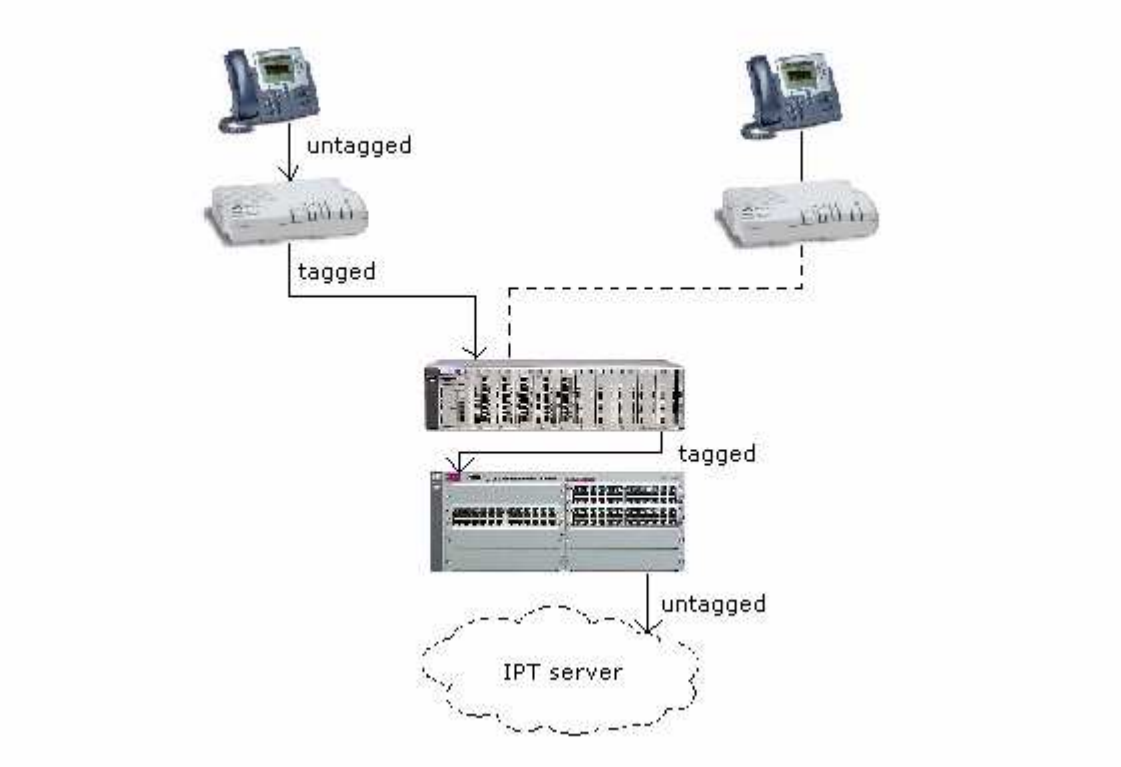


***Figure 33 VoIP traffic from Client to Server***

From client side we start with some sort of SIP device. In this sample we have used a Cisco 7960 SIP phone. The signals form the phone into the residential gateway / switch, normally an Allied Telesyn RG213 or similar model, are untagged. That is; there has not been applied VLAN tagging for data traffic between these two devices. When signals enter the RG213, a VLAN tagging is added on the specific port that the IP device is connected to. From there, traffic is tagged on WAN port, all the way to a HP Procurve 4000 or similar switch. Traffic is switched tagged until it hits the core switch, in this case a HP Procurve 5300 model. In this switch packets are untagged, and sent to the port configured to handle traffic from the particular VLAN. This port is connected to the IP-Telephony server, and there requests are handled. Below we will look at how communication the other way takes place.

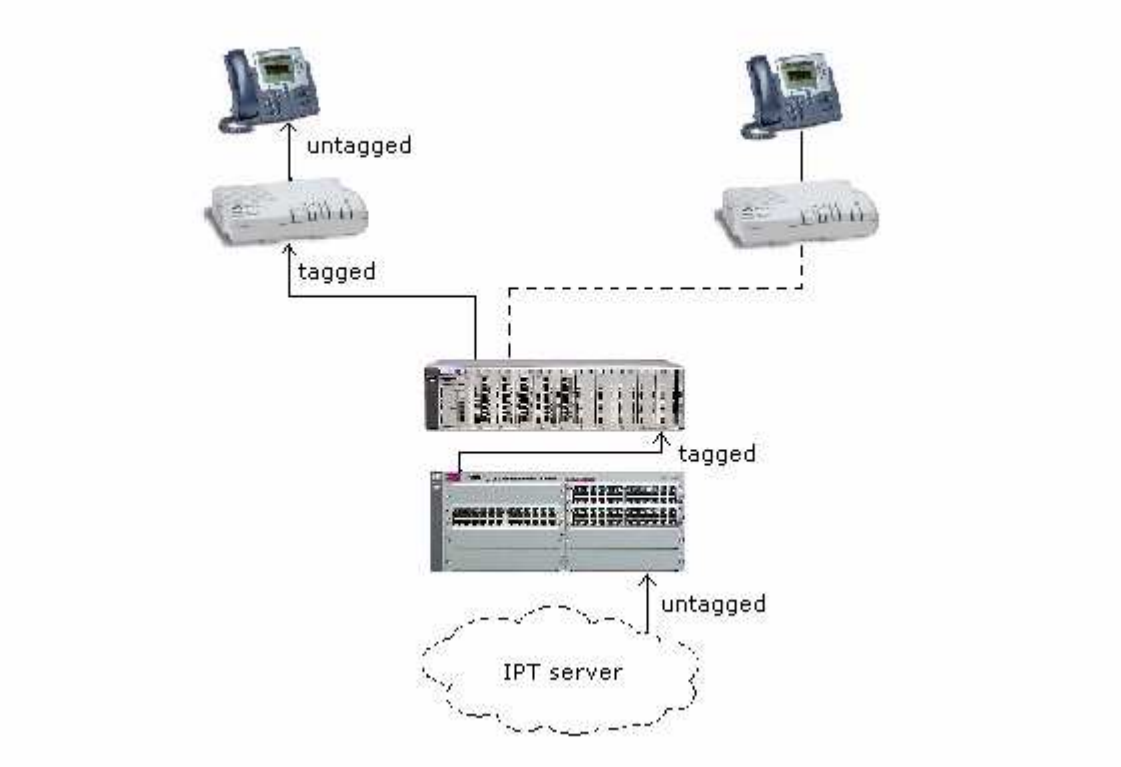


***Figure 34 VoIP traffic from Server to Client***

Communication from server to client is handled in a similar way as from client to server. The server sends data to the core switch, the switch tag's packages, sends them to the respective port, is tagged all the way until it gets to the RG213. When entering WAN port on the RG, it sends data to the defined VLAN port for VoIP traffic; where it is untagged and sent to the phone device.

Traffic between IP-telephony server and the VoIP client device is typically registration, phone initialization, establishing calls and similar traffic. Look at chapter [2.1.7] for a thorough explanation. This type of traffic will be discussed in later chapters as well.

When a phone call has been established, all data traffic takes place between the client equipment signed on that particular session. When connecting and disconnecting a call, the server is needed.

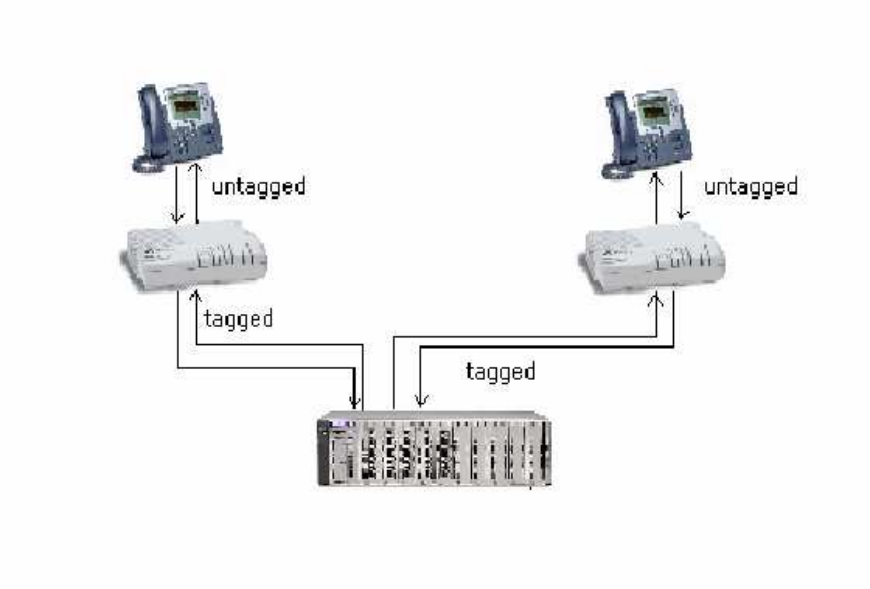


***Figure 35 Client to Client traffic - when within the same Network***

For the communication from a client within the local network to a client on the PSTN, the picture gets a little bit more complex.

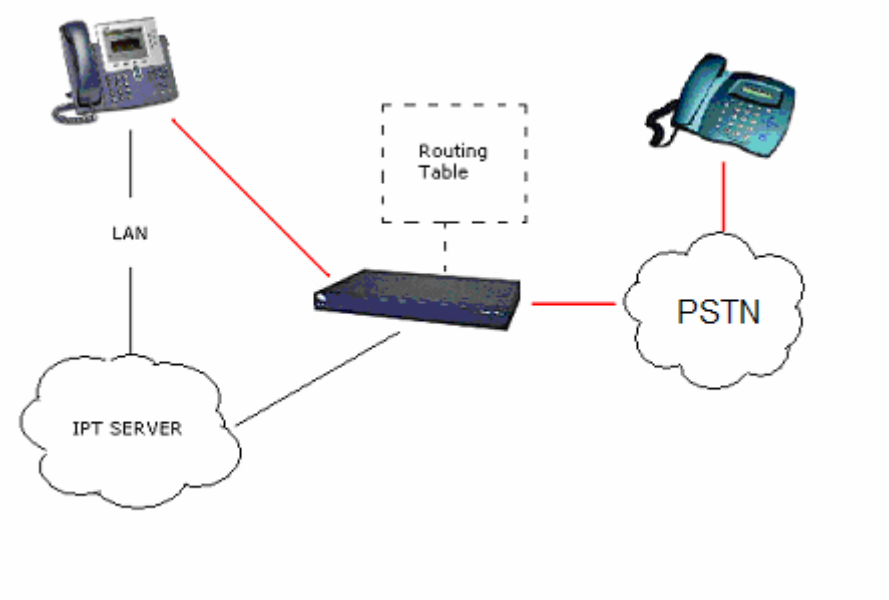


***Figure 36 VoIP traffic against PSTN***

In this figure the switching and VLAN tagging is left out, and referred to as LAN. We have two lines of different colour, one for point to point communication (marked as red), and one for signalling with the server (black). Let's start with signalling. A request for connection to a phone device on the PSTN is sent to the IP-telephony server. The server has not got this number in its internal database, and forwards the request to a Cisco 2651 series router. This router has a telephony routing table with a list of allowed phone numbers. If the number is allowed, a message for trying to connect is sent back to the server, and the caller will get the IP-address for the router and a dial tone; a direct connection to the PSTN phone is established through the router (red line). The router has an interpiter that makes it possible to talk to analogue or ISDN phones on the PSTN. If not the number has qualified, an error message is sent to the client, and the try is terminated. Signalling data is sent between the router and the server, as well as between client caller equipment, and the speech data is sent directly between the SIP phone and the PSTN phone. There is of course also the case of a router and SIP phones in the other end (through the PSTN), but the case is somehow equal.

In our case, the Cisco router works more as an proxy between the PSTN and the IP network that as a traditional router (even though telephony routing is one of its tasks).

## 3.2 Equipment

This chapter describes the equipment that is used in Èlla Kommunikasjons IP-telephony service. It will form a basis for later security analysis. The Allied Telesyn RG213 will be discussed as both a networking device as well as a SIP phone. Therefore you can find it in both the networking and the IP-telephony client chapters.

### 3.2.1 Networking

#### *3.2.1.1 Allied Telesyn Switches and Converters*

***Figure 37 Allied Telesyn RG213TX Residential VoIP Gateway***

The AT-RG213 Residential VoIP Gateway is a home-use access device which integrates the services of fast internet, digital video and telephony over Internet (VoIP). The device has three (3) LAN ports to be connected to PC's or home/office peripherals and one WAN port to connect the CPE (Customer Promise Equipment) to an ISP (Internet Service Provider) network. Through the Line port, the AT-RG213 can be linked to a standard phone/fax analogue (PSTN) line. The AT-RG213 supports a number of different VoIP protocols - these are factory build options but the unit may be firmware re-loaded to a different VoIP protocol if required.

The AT-RG213 is supplied with default settings that allow it to operate immediately as a Residential Gateway. Even if this is all you want to use the gateway for, you should still gain access to the gateway configuration, if only to change the *manager* password to prevent unauthorized access. The AT-RG213 is provided with a Command Line Interface (CLI) for configuration and management.

To use the command line interface (CLI) for configuring the AT-RG213, the first thing you need to do after physically installing the AT-RG213 is to start a terminal session to access the AT-RG213.

A user accessing the AT-RG213 from a terminal or PC connected to the side panel RS-232 terminal port, or via a Telnet connection, must enter a login name and password to gain access at the command prompt.

The AT-RG213 is controlled with commands described in reference [5].

Below is a list of protocols and standards supported by the RG213. These standards will only be listed, as we discuss them further in the security analysis.

| PROTOCOL / STANDARD | REFERENCE |
|---|---|
| ARP | RFCs 826, 925. |
| Assigned Numbers | RFC 1700 |
| DHCP | RFCs 1541, 1542 |
| H.323 | ITU H.323, ITU H.225, ITU H.245 |
| ICMP | RFCs 792, 950. |
| IEEE 802.2 | ANSI/IEEE Std 802.2-1985 |
| IEEE 802.3 | ANSI/IEEE Std 802.3-1985, 802.3a, b, c, e-1988 |
| IGMP | RFC 3228 |
| IP | RFCs 791, 821, 950, 951, 1009, 1055, 1122, 1144, 1349, 1542, 1812, 1858 |
| IP addressing | RFC 1597 |
| L2TP | RFC 2661 |
| NTP | RFCs 958, 1305, 1510 |
| RTP-RTCP | RFC 1889, ITU G.711, ITU G.723, ITU G.729 |
| SDP | RFC 2327 |

| | |
|---|---|
| SIP | RFC 3261 |
| SNMP | MIBs RFCs 1155, 1157, 1213, 1239, 1315, 1398, 1493, 1514, 1573, 2233. |
| TCP | RFC 793 |
| Telnet | RFCs 854–858, 932 1091 |
| TFTP | RFC 1350 |
| UDP | RFC 768 |
| VLAN | IEEE 802.1q |

***Table 10 Allied Telesyn RG213TX VoIP Gateway Protocol / Standards***

Allied Telesyn also has a RG600 series that is compliant with ISDN phones, as RG213 only can have analogue phones connected. For both two series there are 2 different versions; one with a TP cable compliant WAN port, and one with a Fibre WAN port (integrated fibre converter). The one used mostly in Èlla Kommunikasjon is the RG213 with TP connection WAN port.

Software in these models is quite similar, so we will therefore not separate them – but talk about them all as the residential gateway of Allied Telesyn.

For those without an integrated fibre converter, the Allied Telesyn MC103XL fibre converter is used. Both at client side and in the connection points (rack mounted).

***Figure 38 Allied Telesyn MC103X Fibre Converter***

### *3.2.1.2 HP Procurve Switches*

#### 3.2.1.2.1 HP Procurve 4000 Series

This switch operates on OSI-model layer 2. It is used for switching typically at connection points. It has following features:

| Service | Purpose |
|---|---|
| Port monitoring | allows you to monitor traffic using a switched port so you can view several ports at one time with a network analyzer |
| Web interface | allows you to configure the switch from any Web browser on the network |
| Protocol filtering | provides traffic control |
| VLAN support and tagging | supports up to 30 port-based VLANs and dynamic configuration of 802.1Q VLAN tagging providing security between workgroups |
| Class of Service (CoS) tagging | sets 802.1p priority field using policy-based management |
| IP multicast (IGMP) | prevents flooding of IP multicast traffic |
| Port security | prevents unauthorized access using MAC address lockdown |
| Switch meshing | delivers high availability with high performance |
| Hot swappable modules | allow you to add or swap modules without interrupting the network |
| Optional redundant power supply | provides uninterrupted power |
| Spanning Tree Protocol | provides redundant links while preventing network loops |
| Port trunking | for higher switch-to-switch and switch-to-server throughput and link-level redundancy |
| Automatic Broadcast Control (ABC) | minimizes IP and IPX broadcast traffic throughout the network |
| TACACS+ | eases administration of switch management security by using a password authentication server |

| | |
|---|---|
| Cisco Discovery Protocol (CDP) | enables real-time mapping of end nodes to switch ports |
| Cisco Fast EtherChannel | provides higher throughput to other devices that support FEC |

***Table 11 HP Procurve 4000 Series Features***

***Figure 39 HP Procurve 4000 Series Switches - Layer 2***

Security in the services offered by this device will be discussed in the security analysis.

#### 3.2.1.2.2 HP Procurve 5300 Series

This switch operates on OSI-model layer 2, 3 and 4. It is used for switching in central places like i.e. the network core. Features:

| Feature | Purpose |
|---|---|
| | |
| IP layer 3 routing | provides routing of IP at media speed; supports static routes, RIP, RIPv2, and OSPF |
| Router redundancy (XRRP) | allows groups of 2 routers to dynamically back each other up to create highly available routed environments |
| IP multicast routing (PIM Dense) | routes IP multicast using the PIM Dense routing protocol |
| IP multicast (data-driven IGMP) | automatically prevents flooding of IP multicast traffic |
| HP switch meshing | dynamically load-balances across multiple active redundant links to increase available aggregate bandwidth |
| 802.1s Multi-Instance Spanning Tree | high link availability in multiple VLAN environments by allowing multiple spanning trees |
| 802.1w Rapid Convergence Spanning Tree Protocol | increases network uptime through faster recovery from failed links |
| 802.3ad Link Aggregation Control Protocol (LACP) and HP trunking | support up to 36 trunks, each with up to 8 links (ports) per trunk; trunking across modules is supported |
| Cisco Fast EtherChannel (FEC) | supports Cisco's proprietary FEC trunking protocol |
| Web-based authentication | similar to 802.1X, provides a browser-based environment to authenticate clients that do not support the 802.1X supplicant |
| Access control lists (ACLs) | provide IP layer 3 filtering based on source/destination IP address/subnet and source/destination TCP/UDP port number |
| VLAN support and tagging | support complete 802.1Q (4,096 VLAN IDs) and 256 VLANs simultaneously |
| 802.1v Protocol VLANs | isolate select non-IPv4 protocols automatically into their own VLANs |
| Group VLAN Registration Protocol (GVRP) | allows automatic learning and dynamic assignment of VLANs |
| Port security | prevents unauthorized access using MAC address lockdown |
| MAC address lockout | prevents configured particular MAC addresses from connecting to the network |
| Source port filtering | allows only specified ports to communicate with each other |
| TACACS+ | eases switch management security administration by using a password authentication server |
| Secure Shell (SSHv1/SSHv2) | encrypts all transmitted data for secure CLI remote access over IP networks |
| Secure Sockets Layer (SSL) | encrypts all HTTP traffic, allowing secure access to the browser- |

| | |
|---|---|
| | based management GUI in the switch |
| Secure FTP | allows secure file transfer to/from the switch—protects against unwanted file downloads or unauthorized copying of switch configuration file |
| Secure access to manage the 5300xl series | all access methods—CLI, GUI, or MIB—are securely encrypted through SSHv2, SSL, and/or SNMPv3 |
| Static NAT | hide up to 32 nodes per switch from the rest of the network through static IP address translation |
| Layer 4 prioritization | enables prioritization based on TCP/UDP port numbers |
| Traffic prioritization (802.1p) | allows real-time traffic classification into 8 priority levels |
| Rate limiting | limit the maximum ingress traffic on a per-port basis to a configured percentage of that port's total bandwidth |
| Class of Service (CoS) | sets 802.1p priority tag based on IP address, IP Type of Service (ToS), L3 protocol, TCP/UDP port number, source port, and DiffServ |
| RMON, XRMON, sFlow, and SMON | provide advanced monitoring and reporting capabilities for statistics, history, alarms, and events |
| Friendly port names | allow assignment of descriptive names to ports |
| Find-Fix-and-Inform | finds and fixes common network problems automatically, then informs administrator |
| HP Auto-MDIX | automatically adjusts for straight-through or crossover cables on all 10/100/1000 ports |
| Hot-swappable modules | permit modules and mini-GBICs to be added or swapped without interrupting the network |
| Dual flash images | provide independent primary and secondary OS files for backup while upgrading |
| Optional redundant power supply | provides uninterrupted power |
| iSCSI support | enables the deployment of Ethernet storage area network solutions using the iSCSI standard |
| Lifetime warranty | for as long as you own the product |

***Table 12 HP Procurve 5300 Series Features***

***Figure 40 HP Procurve 5300 Series Switches - Layer 2, 3 and 4***

Security in the services offered by this device will be discussed in the security analysis

### *3.2.1.3 Cisco 2600 Series Routers*

In Èlla Kommunikasjon the router (Cisco 2651MX) works as a connection between the local network and PSTN through 100 ISDN lines. That is; the Cisco is used as a telephony gateway between the IP world and the PSTN.

It can provide LAN and WAN configurations, multiple security options and voice/data integration with the following support:

- Multiservice voice/data integration
- VPN access with Firewall and Encryption options
- Analogue dial access services

- Routing with bandwidth management
- Inter-VLAN routing
- Delivery of high-speed business class DSL access
- Integration of flexible routing and low density switching
- Integration of Content Networking
- Integration of Intrusion Detection Systems

The modular architecture of the Cisco 2600 Series allows interfaces to be upgraded to accommodate network expansion or changes in technology as new services and applications are deployed. Network modules available for the Cisco 2600 Series support a broad range of applications, including Multiservice voice/data integration, integrated switching, analogue and ISDN dial access, and serial device concentration.

*Voice/Data integration:* The Cisco 2600 Series allows network managers to provide scalable analogue and digital telephony. Using the Voice/Fax modules, the Cisco 2600 Series may be deployed in both Voice over IP (VoIP) and Voice over Frame Relay (VoFR) networks. The packet voice trunk network module supports up to 60 simultaneous voice calls as well as supporting routing and other services. For Èlla Kommunikasjon, this module has been implemented to gain access to the PSTN for SIP IP-telephony traffic.

With the integration of optional VPN Modules, Cisco IOS based Firewall and IDS, Content Engine Network Modules, or Intrusion Detection Network Modules; it offers a robust and adaptable Security Solution. VPN Modules can be used to provide up to 10 times the performance over software only encryption. Additionally, if used; the Cisco Intrusion Detection System Network Module allows decryption, tunnel termination and traffic inspection at the first point of entry into the network while freeing the router CPU from process-intensive IDS tasks.

***Figure 41 Cisco 2651XM Router***

### 3.2.2 IP-Telephony Servers

In the current IP-telephony configuration at Èlla Kommunikasjon, a SIP Express Router (VoIP server) is handling the VoIP traffic. We have also run a server called Vocal, and therefore they both will be commented, as they are all good freeware alternatives.

#### *3.2.2.1 Vocal*

The VOCAL system is a distributed network of servers that provides Voice over Internet Protocol (VoIP) telephony services. VOCAL supports devices that communicate using the Session Initiation Protocol. VOCAL also supports analogue telephones via residential gateways. It supports on-network and off-network calling. Off-network calling enables subscribers to connect to parties through either the Internet or the Public Switched Telephone Network (PSTN).

Below is given a simplified model of how the system works. There is also a table that explains different servers and elements in the system:

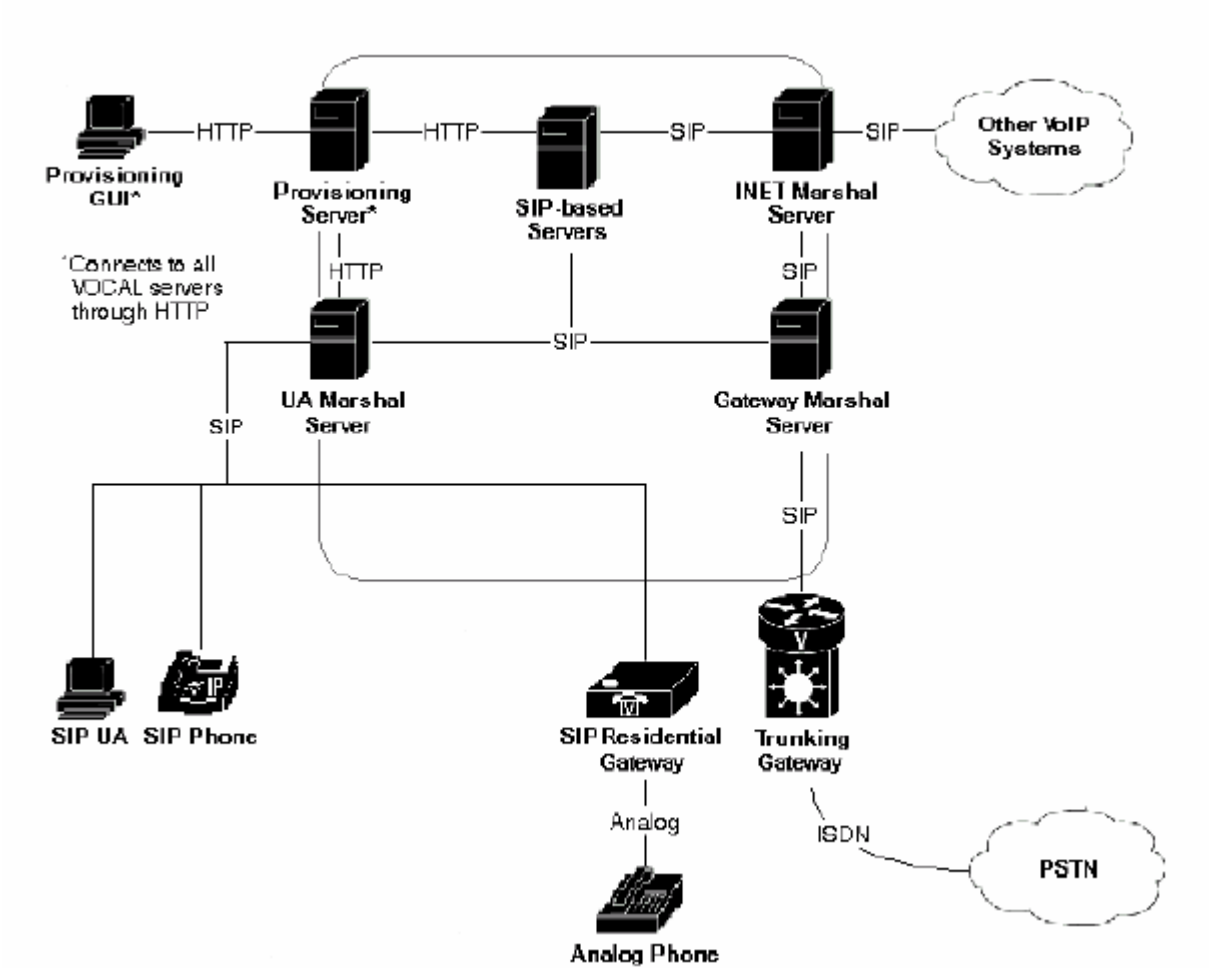


***Figure 42 Simplified View of the VOCAL System Source: Cisco Systems Inc***

| COMPONENT | DESCRIPTION |
|---|---|
| Marshal Server | The Marshal Server (MS) is an implementation of the SIP proxy server and acts as the initial point of contact for all SIP signals that enter the VOCAL system. The MS provides authentication, forwarding and billing functions. |
| Redirect Server | The Redirect Server (RS) is a combined implementation of the SIP redirect, Registration and location servers. The RS stores contact and feature data for all registered subscribers and a dialling plan to enable routing for off-network calls. |
| Call Detail Record Server | The Call Detail Record (CDR) server receives call data from the Marshal Servers and formats it into data that can be transmitted to third party billing systems for invoicing. |
| Voice Mail Server | The Voice Mail server provides unified messaging whereby voice mail messages can be distributed as .wav files attached to e-mail messages. |
| Feature Server | The Feature Servers are another implementation of the SIP proxy server. These servers are scripted in Call Processing Language (CPL) and provide basic system features such as Call Forward and Call Blocking. |
| Provisioning Server | The Provisioning Server (PS) stores data records about each system user and server module, and distributes this information throughout the system via a subscribe-notify model. The PS provides a web-enabled graphical user interface (GUI) to permit technicians and system administrators to manage the system. |
| Heartbeat Server | The Heartbeat Server monitors the flow of signals emitted by the other servers, and provides information about to the flow of heartbeats to the Simple Network |

| | |
|---|---|
| | Management Protocol (SNMP, RFC 1157) GUI. This information helps the System Administrator know if the server modules are up or down. F |

***Table 13 Vocal System Components Description***

Below is a table that compares the Vocal system to the SIP RFC2543 Definition; however it can be hard to understand if you are not familiar with Cisco using its own names for things that has already gotten standard names:

| Server Type | RFC 2543 Definition | VOCAL Functionality |
|---|---|---|
| Location Server | A Location Server can be used by a SIP redirect or proxy server to obtain information about a called party's possible location. The location server can also be an entity outside of the SIP network that uses an alternative protocol, such as Telephony Routing over IP (TRIP, RFC 3219) to communicate with the Redirect Server. | The Location server is a logical function within the VOCAL Redirect server. |
| Proxy Server | An intermediary program that acts as both a server and a client for the purpose of making requests on behalf of other clients. Unlike User Agents, Proxy Servers do not initiate new SIP requests. A Proxy Server interprets, and, if necessary, rewrites a request message before forwarding it. Requests are serviced internally or by passing them on, possibly after translation, to other servers. | The VOCAL system includes specialized SIP Proxy servers called Marshal and Feature servers. |
| Redirect Server | A redirect server is a server that accepts a SIP request, maps the address into zero or more new addresses and returns these addresses to the client. Unlike a proxy server, it does generate SIP requests on behalf of UA's and it does not accept calls. | The SIP Redirect server is a logical function within the VOCAL Redirect server. |
| Registrar | A registrar is a server that accepts REGISTER requests. A registrar is typically co-located with a proxy or redirect server and *may* offer location services. | The SIP Registrar server is a logical function within the VOCAL Redirect server. |

***Table 14 RFC 2543 Definition compared to Vocal Functionality***

The Vocal server has been used in Èlla Kommunikasjon to serve IP-telephony from June 2003 until November 2003. It was stable and ran without any downtime for that period, on an Intel Pentium 3 with Linux SuSe 7.2. It is a Freeware and OpenSource project that has now been incorporated in the Linux SuSe 9.0 distribution.

#### *3.2.2.2 SIP Express Router*

The SIP Express Router has been running stable from November 2003, on the same configuration as the Vocal server (described in previous chapter).

*SIP Express Router (SER)* is an industrial-strength VoIP server based on the session initiation Protocol (SIP RFC2543-bis). It is engineered to power IP telephony infrastructures up to large scale. The server keeps track of users, sets up VoIP sessions, relays instant messages and creates space for new plug- in applications. Its proven interoperability guarantees seamless integration with components from different vendors, eliminating the risk of a single-vendor trap. It has successfully participated in various interoperability tests in which it worked with the products of other leading SIP vendors.

*The SIP Express Router* enables a flexible plug –in model for new applications: Third parties can easily link their plug- ins with the server code and provide thereby advanced and customized services.

Its performance and robustness allows it to serve millions of users and accommodate needs of very large operators. With a dual-CPU, the *SIP Express Router* is able to power IP telephony services in an area as large as the Bay Area[2] of San Francisco during peak hours. Even on an IPAQ PDA, the server withstands 150 calls per second (CPS)! ***Source: iptel.org***

Figure 43 A PDA can process 150 calls per second with SIP Express Router

The *SIP Express Router* is very configurable**,** and allows the creation of various routing and admission policies as well as setting up new and customized services. Its configurability allows it to serve many roles: network security barrier, application server, or PSTN gateway guard for example.

The *SIP Express Router (SER)* includes support for registrar, proxy and redirect mode. Further it acts as an application server with support for CPL, instant messaging and presence (IM&P) including a 2G/SMS gateway, a call control policy language, call number translation, private dial plans and accounting, authorization and authentication (AAA) services. SER runs on Sun/Solaris, PC/Linux, IPAQ/Linux platforms and supports both IPv4 and IPv6.

The *SIP Express Router* (SER) can be easily deployed as the glue connecting SIP components together, be it soft phones, hard phones, PSTN gateways or any other SIP-compliant devices.

As for structure it operates with the same model as Vocal, according to the SIP RFC – with Proxy Servers gateways and so on. In addition to the basic server, we have added a standalone Voicemail server, giving a total solution for corporate cases.

### 3.2.3 IP-Telephony Clients

#### *3.2.3.1 Allied Telesyn AT-RG213TX - SIP module*

The basic functions of the Allied Telesyn RG213 have earlier been described in chapter [3.2.1.1]. In this chapter we will discuss SIP related functionality.

On the RG213 there are 6 different ports. 3 for LAN connection (Internet, television etc.), 1 WAN port for connection to the network, and 2 ports for telephony. This can be either for analogue or ISDN phones, dependent on RG model (213 for analogue and 600 for ISDN). Each of these 2 ports can be assigned a unique phone number. A basic configuration of phones through a Telnet session (after login) looks like this:

```
-------------------------------------------------------------------------------
// Sets SIP active
-------------------------------------------------------------------------------
sip enable

-------------------------------------------------------------------------------
```

[2] San Francisco with a population of 7 200 000

```
// Configure phone port 0
-------------------------------------------------------------------------------
c sip port=0 phno=38140202 a-usr=38140202 a-pwd=ABCD ps=ipt-server.ivisjon.no do=ipt-server.ivisjon.no ls=ipt-
server.ivisjon.no cap=all
s phone port=0 defaultcall=voip

-------------------------------------------------------------------------------
// Configure phoneport 1
-------------------------------------------------------------------------------
c sip port=1 phno=38140203 a-usr=38140203 a-pwd=ABCD ps=ipt-server.ivisjon.no do=ipt-server.ivisjon.no ls=ipt-
server.ivisjon.no cap=all
s phone port=0 defaultcall=voip

-------------------------------------------------------------------------------
// Setting prefix for use of PSTN instead of VoIP – if you have a backup connection to PSTN
-------------------------------------------------------------------------------
set phone port=0 prefix=0
set phone port=1 prefix=0

-------------------------------------------------------------------------------
// Save configuration
-------------------------------------------------------------------------------
create config=dov.cfg

-------------------------------------------------------------------------------
// Set he configuration to load at boot
-------------------------------------------------------------------------------
set config=dov.cfg

-------------------------------------------------------------------------------
// Reboot system with new configuration
-------------------------------------------------------------------------------
restart reboot
```

***Table 15 SIP config of the Allied Telesyn RG213TX***

When the system is rebooted, the new configuration is loaded, and the SIP module in the RG213 registers towards the IP-telephony server: *ipt-server.ivisjon.no.* When an analogue phone is plugged into one of the phone ports, it will be able to make phone calls. This configuration is not setup using encryption.

**Notice:** If you want to use external SIP devices(I.e. Cisco 7960 Series SIP Phones), you need to tag a LAN port on the RG213 with the VoIP VLAN – that is a limitation within the Èlla Kommunikasjon network, caused by the VLAN tagging structure.

### *3.2.3.2 Cisco IP Phone 7960 Series*

The Cisco IP Phone has been tested for deployment in the corporate market. It is normally configured on one of the LAN ports of the Allied RG, added the proper VLAN tagging. This (VLAN tagging) is also possible to do directly in the phone configuration, but cause of Èlla Kommunikasjon policy; it has been done within the Allied switch. For phone setup, it uses TFTP to download its config. This is done by making unique configuration files and storing them in the TFTP server directory. Config download authentication and filtration is done by checking for the Cisco Phone's Mac address. Typical config file looks like this:

```
# SIP Configuration Generic File

# Line 1 appearance
line1_name: 8540

# Line 1 Registration Authentication
line1_authname: "8540"

# Line 1 Registration Password
line1_password: "8540"

# Line 2 appearance
```

```
#line2_name:

# Line 2 Registration Authentication
#line2_authname: ""

# Line 2 Registration Password
#line2_password: ""


####### New Parameters added in Release 2.0 #######

# All user_parameters have been removed

# Phone Label (Text desired to be displayed in upper right corner)
phone_label: "Dag Ove Valsgaard  "            ; Has no effect on SIP messaging

# Line 1 Display Name (Display name to use for SIP messaging)
line1_displayname: "8540"

# Line 2 Display Name (Display name to use for SIP messaging)
line2_displayname: ""


####### New Parameters added in Release 3.0 ######

# Phone Prompt (The prompt that will be displayed on console and telnet)
phone_prompt:   "SIP Phone"      ; Limited to 15 characters (Default - SIP Phone)

# Phone Password (Password to be used for console or telnet login)
phone_password: "cisco" ; Limited to 31 characters (Default - cisco)

# User classifcation used when Registering [ none(default), phone, ip ]
user_info: phone
```

***Table 16 Cisco 7960 IP phone Config Specific for single phone***

I addition to this config file, a default config file is loaded. This file contains settings that are common for all client equipment, and the one above is unique for that phone. It is rather long and can be uninteresting, but I would like to point out the fact that proxy addresses, codec and ports are defined in this document, and therefore it is rather important.

```
# SIP Default Generic Configuration File

# Image Version
image_version: P0S3-06-1-00

# Proxy Server
proxy1_address: "212.4.33.26"              ; Can be dotted IP or FQDN
proxy2_address: ""                  ; Can be dotted IP or FQDN
proxy3_address: ""                  ; Can be dotted IP or FQDN
proxy4_address: ""                  ; Can be dotted IP or FQDN
proxy5_address: ""                  ; Can be dotted IP or FQDN
proxy6_address: ""                  ; Can be dotted IP or FQDN

# Proxy Server Port (default - 5060)
proxy1_port: 5060
proxy2_port: 5060
proxy3_port: 5060
proxy4_port: 5060
proxy5_port: 5060
proxy6_port: 5060

# Proxy Registration (0-disable (default), 1-enable)
proxy_register: 1

# Phone Registration Expiration [1-3932100 sec] (Default - 3600)
timer_register_expires: 3600

# Codec for media stream (g711ulaw (default), g711alaw, g729a)
preferred_codec: g711alaw

# TOS bits in media stream [0-5] (Default - 5)
tos_media: 5

# Inband DTMF Settings (0-disable, 1-enable (default))
dtmf_inband: 1

# Out of band DTMF Settings (none-disable, avt-avt enable (default), avt_always - always avt )
dtmf_outofband: avt

# DTMF dB Level Settings (1-6dB down, 2-3db down, 3-nominal (default), 4-3db up, 5-6dB up)
dtmf_db_level: 3

# SIP Timers
timer_t1: 500                       ; Default 500 msec
timer_t2: 4000                      ; Default 4 sec
sip_retx: 10                        ; Default 10
sip_invite_retx: 6                  ; Default 6
timer_invite_expires: 180 ; Default 180 sec

######## New Parameters added in Release 2.0 #######

# Dial plan template (.xml format file relative to the TFTP root directory)
dial_template: dialplan

# TFTP Phone Specific Configuration File Directory
tftp_cfg_dir: ""              ; Example:  ./sip_phone/

# Time Server (There are multiple values and configurations refer to Admin Guide for Specifics)
sntp_server: "ntp.uio.no"                   ; SNTP Server IP Address
sntp_mode: directedbroadcast       ; unicast, multicast, anycast, or directedbroadcast (default)
time_zone: CET                      ; Time Zone Phone is in
dst_offset: 1                       ; Offset from Phone's time when DST is in effect
dst_start_month: Mars               ; Month in which DST starts
dst_start_day: 30                   ; Day of month in which DST starts
dst_start_day_of_week: Sun          ; Day of week in which DST starts
dst_start_week_of_month: 4          ; Week of month in which DST starts
dst_start_time:  02          ; Time of day in which DST starts
dst_stop_month: Oct                 ; Month in which DST stops
dst_stop_day: 26            ; Day of month in which DST stops
dst_stop_day_of_week: Sunday      ; Day of week in which DST stops
dst_stop_week_of_month: 4           ; Week of month in which DST stops 8=last week of month
```

```
dst_stop_time: 3            ; Time of day in which DST stops
dst_auto_adjust: 1                  ; Enable(1-Default)/Disable(0) DST automatic adjustment
time_format_24hr: 1                 ; Enable(1 - 24Hr Default)/Disable(0 - 12Hr)

# Do Not Disturb Control (0-off, 1-on, 2-off with no user control, 3-on with no user control)
dnd_control: 0                      ; Default 0 (Do Not Disturb feature is off)

# Caller ID Blocking (0-disbaled, 1-enabled, 2-disabled no user control, 3-enabled no user control)
callerid_blocking: 0                ; Default 0 (Disable sending all calls as anonymous)

# Anonymous Call Blocking (0-disabled, 1-enabled, 2-disabled no user control, 3-enabled no user control)
anonymous_call_block: 0             ; Default 0 (Disable blocking of anonymous calls)

# DTMF AVT Payload (Dynamic payload range for AVT tones - 96-127)
dtmf_avt_payload: 101               ; Default 101

# Sync value of the phone used for remote reset
sync: 1                             ; Default 1

######## New Parameters added in Release 2.1 ########

# Backup Proxy Support
proxy_backup: ""            ; Dotted IP of Backup Proxy
proxy_backup_port: 5060             ; Backup Proxy port (default is 5060)

# Emergency Proxy Support
proxy_emergency: ""                 ; Dotted IP of Emergency Proxy
proxy_emergency_port: 5060          ; Emergency Proxy port (default is 5060)

# Configurable VAD option
enable_vad: 0                       ; VAD setting 0-disable (Default), 1-enable

######## New Parameters added in Release 2.2 #######

# NAT/Firewall Traversal
nat_enable: 0                ; 0-Disabled (default), 1-Enabled
nat_address: ""                     ; WAN IP address of NAT box (dotted IP or DNS A record only)
voip_control_port: 5060             ; UDP port used for SIP messages (default - 5060)
start_media_port: 16384  ; Start RTP range for media (default - 16384)
end_media_port: 32766    ; End RTP range for media (default - 32766)
nat_received_processing: 0          ; 0-Disabled (default), 1-Enabled

# Outbound Proxy Support
outbound_proxy: ""                  ; restricted to dotted IP or DNS A record only
outbound_proxy_port: 5060       ; default is 5060

######## New Parameter added in Release 3.0 ########

# Allow for the bridge on a 3way call to join remaining parties upon hang-up
cnf_join_enable : 1                 ; 0-Disabled, 1-Enabled (default)

######## New Parameters added in Release 3.1 ########

# Allow Transfer to be completed while target phone is still ringing
semi_attended_transfer: 1; 0-Disabled, 1-Enabled (default)

# Telnet Level (enable or disable the ability to telnet into the phone)
telnet_level: 2                     ; 0-Disabled (default), 1-Enabled, 2-Privileged

######## New Parameters added in Release 4.0 ########

# XML URLs
services_url: ""            ; URL for external Phone Services
directory_url: ""           ; URL for external Directory location
logo_url: "http://212.4.33.4/ella.bmp"        ; URL for branding logo to be used on phone display

# HTTP Proxy Support
http_proxy_addr: ""                 ; Address of HTTP Proxy server
http_proxy_port: 80                 ; Port of HTTP Proxy Server (80-default)

# Dynamic DNS/TFTP Support
```

```
dyn_dns_addr_1: ""            ; restricted to dotted IP
dyn_dns_addr_2: ""            ; restricted to dotted IP
dyn_tftp_addr: ""           ; restricted to dotted IP

# Remote Party ID
remote_party_id: 1                  ; 0-Disabled (default), 1-Enabled

######## New Parameters added in Release 4.4 #######

# Call Hold Ringback (0-off, 1-on, 2-off with no user control, 3-on with no user control)
call_hold_ringback: 0                        ; Default 0 (Call Hold Ringback feature is off)
```

***Table 17 Cisco 7960 IP Phone Config Common***

In addition to this, a dialplan is loaded. The dialplan follows XML formatting.

```
<DIALTEMPLATE>
   <TEMPLATE MATCH="99"     Timeout="0" User="Phone"/> <!-- Operator -->
   <TEMPLATE MATCH="...."    Timeout="0" User="Phone"/> <!-- Local numbers -->
   <TEMPLATE MATCH="0,........"         Timeout="0" User="Phone"/> <!-- National calls-->
   <TEMPLATE MATCH="0,00*"         Timeout="4" User="Phone"/> <!-- International calls-->
   <TEMPLATE MATCH="0,11."         Timeout="0" User="Phone"/> <!-- Service numbers -->
   <TEMPLATE MATCH="0,0[1-9]..."    Timeout="0" User="Phone"/> <!-- Service numbers -->
   <TEMPLATE MATCH="0,*"          Timeout="4" User="Phone"/> <!-- Anything else -->
   </DIALTEMPLATE>
```

***Table 18 Cisco 7960 IP Phone Config Dialplan***

To get this file, the Cisco phone uses the default TFTP server, given in the DHCP request. It loads files as dial plan and common config, and then it loads the config file specific to that phone based on its MAC address (the file has the MAC address as filename). It is now possible to make phone calls.

Users need to enter a password through the user interface on the phone screen to get the configuration. This way, it is hard for users to gain access to the configuration. Authentication for the SIP server is encrypted using Digest – so packet sniffing will not automatically give the password.

***Figure 44 Cisco IP Phone 7960 Series***

### *3.2.3.3 LeadTek Broadband Videophone*

The LeadTek broadband Videophone was used for test purposes during the Q4 2003. Èlla Kommunikasjon looked for a cost effective videoconferencing solution. After a wide product search the LeadTek videophone was selected. It is cost efficient, has a lot of possibilities – and works as a desktop model – both for voice (specifically) and video. With a total cost below 350£, it has been found useful for its purpose, and is an exciting alternative in a price range where you normally only find voice solutions.

The main drawback with the phones we tested was a lack of good software for SIP. The phone was initially made with H.323 in thought, but we came across early Beta software that we were allowed to use. This software lacked support for encryption and several media types.

Configuration of this phone can only be done manually, even though H.323 versions can handle TFTP. This is caused by the beta software – and will hopefully be improved in the near future. Passwords and network settings is set through a user interface on the phone. You need to login with a password to reach the configuration program.

Once it has been configured, it registers with the IP-telephony service as all the other devices. It offers good quality video transmission, and has a possibility of connection several video devices and switching between those. Else you can also transfer incoming signals onto a video projector, as well as external speakers and microphones.

***Figure 45 LeadTek Broadband Videophone***

### *3.2.3.4 Soft phones*

Soft phones are used as applications on i.e. Microsoft or UNIX platforms. They need a speaker, microphone and a soundcard installed. This devices are as the name implies; pure software phones. You are given a user interface, making you able to process phone calls, and use the pc as a personal telephone. This can come in handy when you travel, or if you use different workspace each day. That way you can bring your whole office in i.e. a laptop. In most soft phones you will have to configure the device manually. That is; give sip server address, phone number, password and so on in a configuration interface.

We have tested a variety of soft phones with the SIP servers at Èlla Kommunikasjon, and they all work fine (both for UNIX and Windows platforms). The main problem for these devices is that they cannot use VLAN tagging, and users can list the configuration. Therefore the phones have been put directly on the core switch to be able to process calls.

Èlla Kommunikasjon has no intention to use Soft phones for their customers now, but in the future the demand for new services probably will make its way and deployment of these devices will take place.

In this paper we will not do a thorough investigation of these devices, but it will be mentioned why it is a problem to do this according to security policy at the moment, and what can be done to offer this service.

Clients as MSN Netmeeting and Cisco Softphone are used in some networks today, and an updated list can be found at: http://www.iptel.org/info/products/sipphones.php. We have used the Estara Softphone for testing, and this phone works well with both the SIP Express Router and Vocal.


***Figure 46 Estara Softphone***

# 4 Security Analysis

In this chapter we will try to compromise the VoIP service of Èlla Kommunikasjon AS. This will be done by pretending to be a malicious attacker, wanting to gain as much control as possible over the system. First we will look at security threats, and then we will use different approaches to achieve our goal.

*At the end of this document on page 116, you can find a small dictionary with explanation of some hacker related glossaries. It can be useful if you get confused.*

## 4.1 Security Threats

The different types of computer and network security threats are best characterized with basis in the function of the computer system to be providing information. In general there is always a flow of information from a source to a destination.

- *Interception* – An unauthorized party gains access to an asset.
- *Interruption* - An asset of the system is destroyed or becomes unavailable or unusable.
- *Modification* – An unauthorized party not only gains access but also tampers with an asset.
- *Fabrication* – An unauthorized party inserts counterfeit objects into the system.

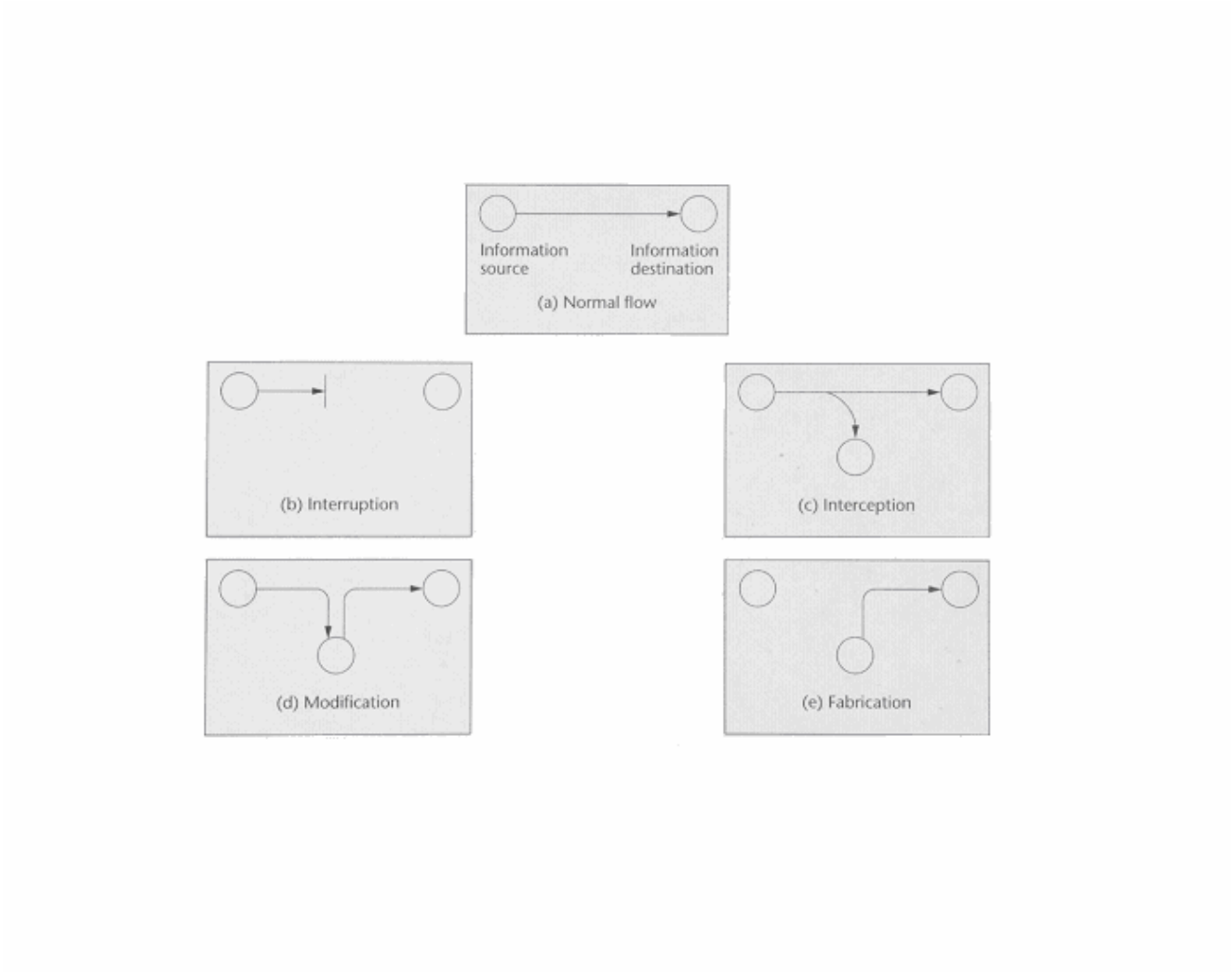


***Figure 47 Security threats. Source: ISBN 0201485346 [6]***

With regards to data-communication, we can classify these types of threats into passive and active.

### 4.1.1 Passive threats

Passive threats involve learning or making use of information from the system (interception) without affection the system resources. The next two paragraphs present two passive threats, both of the type interception.

*Release of message content (Interception)*
*Release of message content* is a self-explaining threat. An intruder is able to interpret and extract information being transmitted. The highest risk is the release of authentication information, which could be used to compromise additional system resources.

*Traffic analysis (Interception)*
Another type of interception is *traffic analysis*. An intruder, who is not able to interpret and extract the transmitted information, might still be able to derive (infer) information from the traffic characteristics. Determination of the location and identity of communicating hosts and observing frequency and length of messages being exchanged might be useful in guessing the nature of the communication that is taking place, even if it is encrypted.

### 4.1.2 Active threats

Active threats involve altering (interruption, modification or fabrication) of the system resources to affect their operation. The three next paragraphs present three active threats, one of each type.

Message stream modification (Modification)
*Message-stream modification* simply means that some portion of a legitimate message is altered or those messages are delayed, replayed, or reordered, in order to produce an unauthorized effect.

Masquerade (Fabrication)
A *masquerade* takes place when one entity pretends to be a different entity. A masquerade attack usually includes one of the other two forms of active attack. Such an attack can take place, for example, by capturing and replaying an authentication sequence.

Denial of Service (Interruption)
The *denial of service* prevents or inhibits the normal use or management of communication facilities. This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (for example, the security audit service). Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

### 4.1.3 Security Services

Security services in ISO 7498-2 are a special class of safeguard applying to a communications environment.

**Authentication**
– Ensuring that users are the entities they claim to be
Peer entity authentication
Data origin authentication

**Access control**
- Ensuring that users access only those resources and services that they are entitled to access and that qualified users are not denied access to services that they legitimately expect to receive

**Data Confidentiality**
Connection Confidentiality
Connectionless Confidentiality
Selective-Field Confidentiality
Traffic-flow Confidentiality

**Data integrity**
Connection Integrity with Recovery
Connection Integrity without Recovery
Selective-Field Connection Integrity
Connectionless Integrity
Selective-Field Connectionless Integrity

**Availability**
- Ensuring that a system is operational and functional at a given moment, usually provided through redundancy, loss of availability is often referred to as "denial-of-service"

**Non-repudiation**
- Ensuring that the originators of messages cannot deny that they in fact sent the messages. This is on short terms:

- Non-repudiation, Origin
- Non-repudiation, Destination

## 4.2 Traffic Analysis

One of the most common ways to find network data is to listen to packages on the network. This is called packet sniffing. Packet sniffing is done by using a client that listens to traffic on the network. This client then enterpits this information and gives feedback in form of useful data (not 0’s and 1’s). I.e. ARP, SIP or other message formats. Sample of a decoded SIP package below:

```
Frame 712 (578 bytes on wire, 578 bytes captured)
Ethernet II, Src: 00:04:80:1c:d2:00, Dst: 00:30:84:d0:f1:a4
802.1q Virtual LAN
Internet Protocol, Src Addr: ipt-server1.ivisjon.no (10.100.10.118), Dst Addr: 10.5.4.105 (10.5.4.105)
User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
Session Initiation Protocol
   Status line: SIP/2.0 100 trying -- your call is important to us
   Message Header
      From: <sip:38140229@ipt-server1.ivisjon.no>;tag=1052398365
        To: <sip:38608540@ipt-server1.ivisjon.no>
        Call-ID:1230223512@10.5.4.105
        CSeq:184153295 INVITE
        Via:SIP/2.0/UDP 10.5.4.105;branch=z9hG4bK15a69165
        Server: Sip Express router (0.8.12 (i386/linux))
        Content-Length: 0
        Warning: 392 10.100.10.118:5060 "Noisy feedback tells:  pid=1886 req_src_ip=10.5.4.105 req_src_por
```

***Figure 48 Decoded SIP message from Traffic Analyzer***

This message is a screenshot from an *Ethereal Network Protocol Analyzer v.10.0*. The package decoded is a SIP package. From the figure it is possible to see that the analyzer also informs us of the VLAN. From this you can see what VLAN tagging and identification is used. We will later use the information from such sessions to discuss security issues.

To achieve effective and useful results, a sniffing environment has been configured [Figure 49].

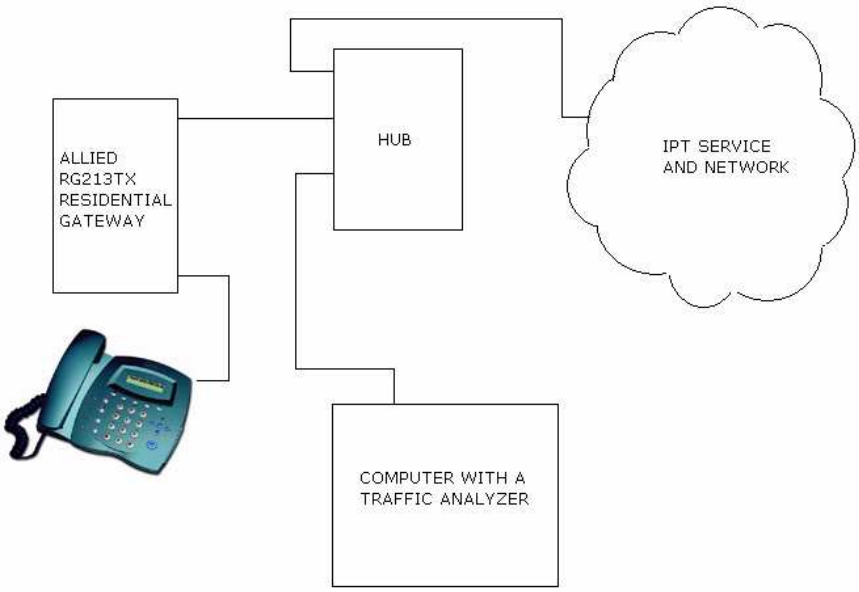


***Figure 49 Sniffing packages from the Allied Telesyn Residential Gateway***

A hub has been put between the Allied RG213 and the network. Hubs operate in such a way that we can listen to all traffic sent on the network at that specific cable. A computer with a packet sniffer listens to the traffic, and generates reports from what it "hears."

This test environment has been put up in such a way, since this also is possible for all customers – as well as it gives all information needed from the transmitted data.

In the following sub chapters focus will be on critical sessions, and displayment of data from them.

### 4.2.1 SIP

The critical areas of the SIP sessions (security focus) will typically be authentication towards server. According to the theory [2.1], the messages sent are:

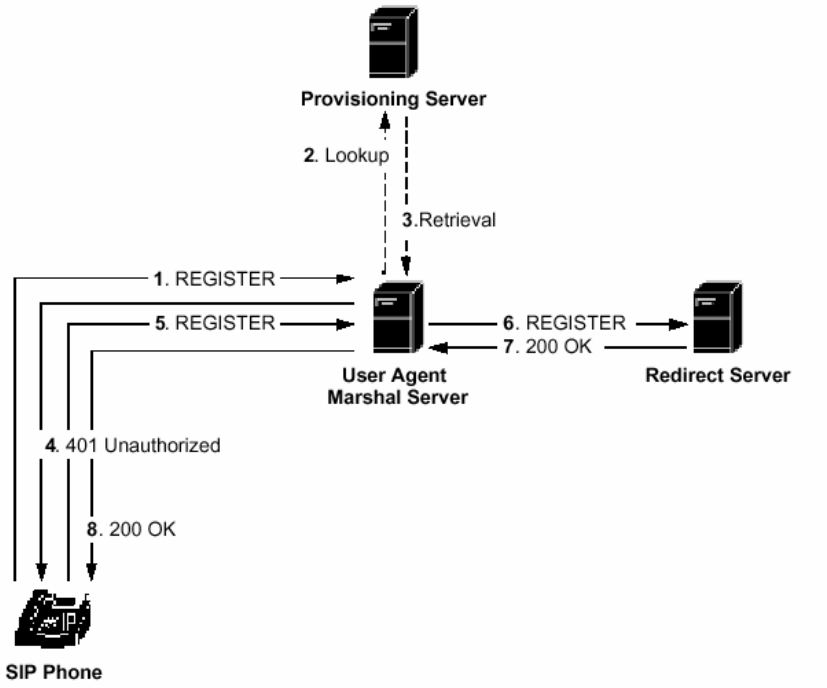


***Figure 50 SIP Registration Process Source: Cisco Systems Inc***

The most interesting message here is number 5 ”Register.” In this message, the username and password is sent. We will first look at a message encrypted with Digest [2.3.3].

```
⊞ Frame 562 (617 bytes on wire, 617 bytes captured)
⊞ Ethernet II, Src: 00:30:84:d0:f1:a4, Dst: 00:04:80:1c:d2:00
⊟ 802.1q Virtual LAN
    111. .... .... .... = Priority: 7
    ...0 .... .... .... = CFI: 0
    .... 0000 0000 0101 = ID: 5
    Type: IP (0x0800)
⊞ Internet Protocol, Src Addr: 10.5.4.105 (10.5.4.105), Dst Addr: ipt-server1.ivisjon.no (10.100.10.118)
⊞ User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
⊟ Session Initiation Protocol
  ⊞ Request line: REGISTER sip:ipt-server1.ivisjon.no SIP/2.0
  ⊟ Message Header
      User-Agent: ATI-RG213TX/6-4-0
    ⊞ From: <sip:38140229@ipt-server1.ivisjon.no>;tag=1052398365
      To: <sip:38140229@ipt-server1.ivisjon.no>
      Call-ID:1450604030@0.0.0.0
      CSeq:1901860875 REGISTER
      Contact: <sip:38140229@0.0.0.0>;expires=0
      Max-Forwards:70
      Via:SIP/2.0/UDP 10.5.4.105;branch=z9hG4bK9df067e
      Authorization: Digest username="38140229",
       realm="ipt-server1.ivisjon.no",
       nonce="4028f655c1ad36c1f5cd811e652b25254bd10c3e",
       uri="sip:ipt-server1.ivisjon.no",
       response="e7feebdfcf0924dc869fffdf135eadd0"
      Content-Length:0
```

***Figure 51 SIP Register with Digest***

As you can see, the message contains both username and password. For the password encryption has been added. That is; Username is public and Password has been encrypted.
For registration without digest encryption the register message has some differences. First of all the Authorisation has a basic http type, and the password looks like it has been encrypted (this cause we know what the password really is).

It is possible to see the username and password. Password looks kind of weird, and therefore it is easy to assume it has been encrypted. That is however not right. In the message it is possible to find that the password has been hidden with the basic method. This is a method that uses a

predefined "key" to hide data. This key is the same everywhere, and it is easy to find websites or software that can decode the message for you. I.e. you can try the tool on this URL: http://www.securitystats.com/tools/base64.php - it offers basic decoding and encoding.

If you try to decode the basic password it will be like this:

1. Insert the following information: "encrypted password".
2. Choose decode from the dropdown menu and press decode.
3. You are now displayed the original password: "plain password".

It is also possible to do this the reversed way. In our case the "encrypted password" delivers the "encrypted password."

### 4.2.2 TELNET

With the information found in the previous chapter, we focused on the SIP part of the messages. This information could give us username and password for SIP authentication in some cases (basic).

For the cases where digest encryption is used, we have no knowledge of the password, therefore it is crucial to be able logging into the user devices (primary Allied RG213TX) and try to read this information. If it is possible to login to the Allied boxes, we can also configure them as we wish. This will be covered later.

If we listen to the traffic long enough, we will find that the RG is configured and managed through the Telnet protocol. We can define a sniffers session, so that it only listens for telnet traffic. A telnet session with the RG looks like this in a sniffers:

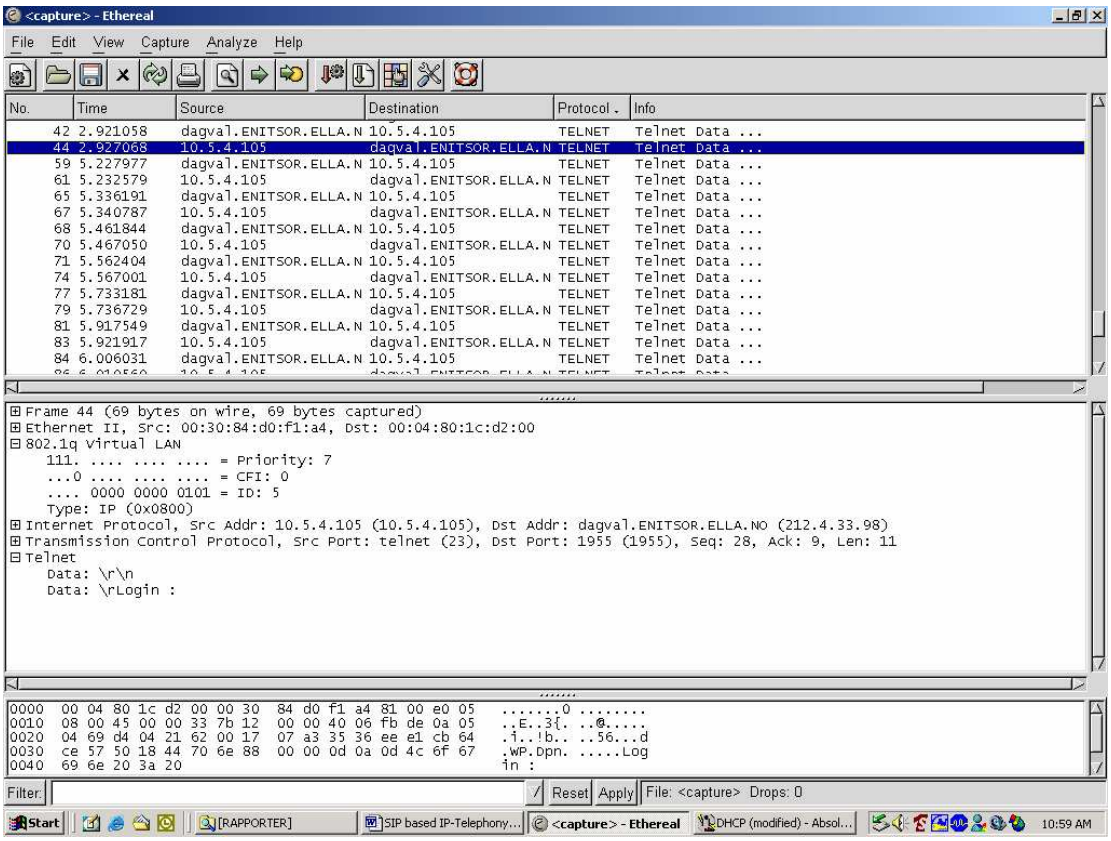


***Figure 52 Sniffing Telnet Sessions***

The package displayed shows the Login sequence. For finding the username and password, you just browse the next few messages – and you find that it is sent unencrypted. This way it is possible to read out both username and password.

If we now can add the proper VLAN tagging to our network device, it should be possible to log into the device.

#### *4.2.2.1 In case of SSH*

If telnet has been exchanged with SSH to ensure a secure login, we cannot sniff the username and password, and therefore a secure login can be made.

**Notice: SSH 2.0 servers up to version prior to 3.0.2p1 have a bug that allows users to gain root access.**

### 4.2.3 VLAN

As we have seen in previous message samples, it is possible to read out VLAN tagging directly from the messages. I.e. Figure 52 shows that Telnet sessions to the Allied RG takes place on VLAN tagged with ID=5. This way we know that if we want to connect to the RG with a telnet session, it most likely has to take place on the virtual LAN tagged with id=5.

Our next goal is therefore to apply tagging to our communication. This will be covered in the next chapter. Main idea behind this is to get access to all VLAN's, and in some networks this can enable you pretending being an administrator on i.e. a so-called management VLAN.

### 4.2.4 An organized assault

Based on what we know from the packet analysis, we now want to perform an organized assault. The goal of this assault will be to make a phone call from a Softphone located on a computer communicating between the network and the Allied RGW. To do this, we need to have a network card somehow connected to the network – and in-between we will also have to add VLAN tagging to gain access to the various VLAN's. The main picture can be explained by the figure below:

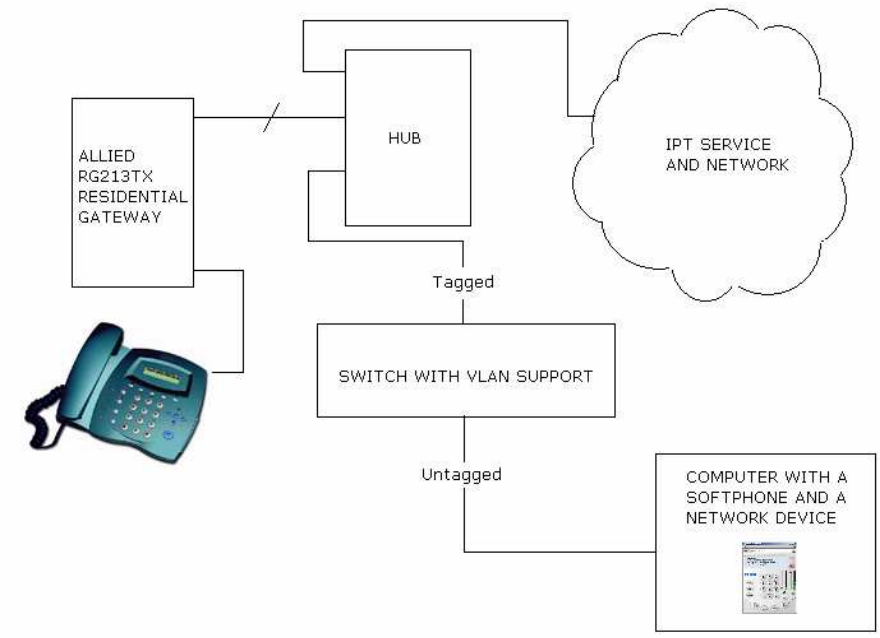


***Figure 53 Organized Assault - Main Picture***

Here is a checklist of how we want to do the assault:

1. Sniff packages, get the following information
- IP-address, subnet mask, default gateway and DNS server for the Allied RGW
- VLAN for telnet traffic and IP-telephony traffic
- Telnet username and password
2. Apply appropriate VLAN tagging to the network connection
3. Get access from the DHCP server on this VLAN. If not given IP-address, take a logical one, based on what you know from the RG IP-configuration.
4. Start a Telnet session towards the RG. Use the username and password given through packet sniffing.
5. List the phone settings (Might need to use a help function in the telnet interface, or gain access to RG documentation – for command knowledge).
6. Take the power to the RG, and now use the RG IP-configuration on your network card. This way we ensure that no one has the same IP as we have.
7. Install a Softphone on the computer, and configure it with the same configuration as the RG.
8. If possible; make a phone call.

We split this into sub chapters.

Sniffing packages, applying appropriate VLAN tagging, getting / setting an IP, telnet session with the RG and Softphone unauthorized call.

#### *4.2.4.1 Sniffing packages*

The goal of the package sniffing is to get IP-address, subnet mask, default gateway and DNS server for the Allied RG213TX. We also need the VLAN tagging of the telnet and IP-telephony VLAN, as well as Telnet session username and password.

Earlier we have in [Figure 52] found the VLAN and Telnet information. What remain are TCP/IP details. To find this, we pull the power from the RG213, start sniffing and put power on again. The RG213 will now try getting an IP-address. The reply is broadcasted, and we will find what we look for[3]. The package we sniff looks like this in the Analysis tool:

[3] Notice that DHCP broadcasts the reply; this means that all other devices in our subnet also get their IP by broadcast on the same subnet (for most switched networks). This means that we can get TCP/IP details for all devices in our subnet. It might come in handy if you i.e. want to break into your neighbors RG and steal his telephony account; your neighbor pays for the airtime.

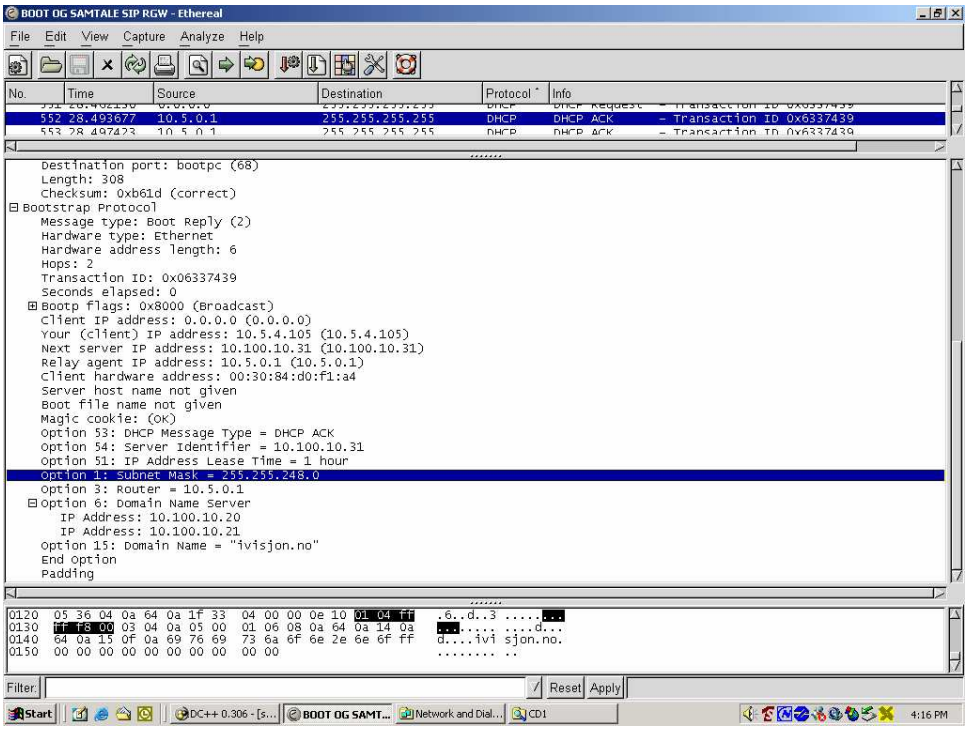


***Figure 54 Sniffing DHCP requests and Replies***

We can now see that the device uses following information:

IP-address: 10.5.4.105
Subnet mask: 255.255.248.0
Gateway: 10.5.0.1
DNS: 10.100.10.20 alternate 10.100.10.21

We already know from [Figure 52] that the VLAN tagging ID=5 and Telnet username=manager and password=rgwpass.

### *4.2.4.2 Applying appropriate VLAN tagging*

Now we need to apply the appropriate VLAN tagging to the communication between the hacker computer and the network. This is done by putting a HP Procurve 2524 switch between us and the network [Figure 53]. In the switch we untag information between the switch and the network card, as well as we tag information between the switch and the network. The configuration of the switch can be found on the figure below:

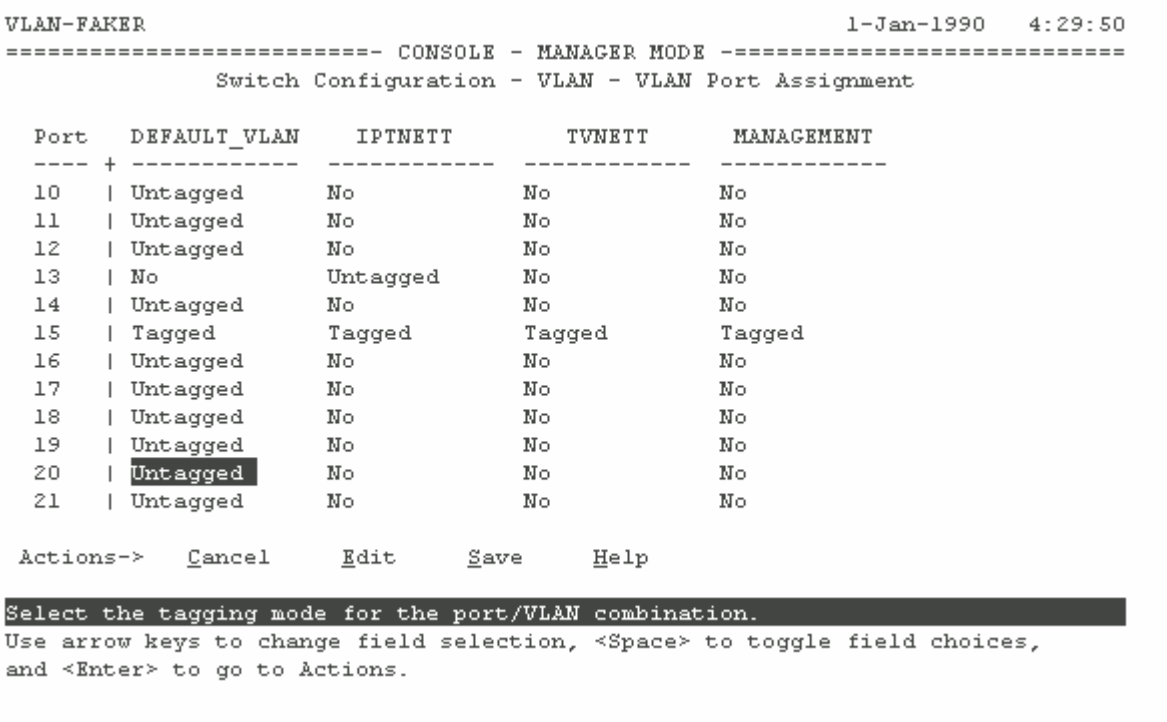


***Figure 55 VLAN Faker – configuration***

In short, this means that all traffic to port 13 will be untagged VLAN id=5. Here we use an alias called IPTNETT. Further all traffic on port 15 will be tagged with various LAN's, also the IPT VLAN.

Now we should be able to communicate on the desired VLAN. It has also been added different VLAN's on other ports (found this ones during traffic analysis);

Management = Port 3
Internet (default) = Port 1
TV = Port 5

This way we can easily switch between the VLAN's when we need to.

### *4.2.4.3 Getting / setting an IP*

Now we try getting an IP-address from the DHCP server existing on the IPT vlan. For the network card, we set TCP/IP all with automatic settings. The experience is that we are not given an IP-address. This can be caused by several things, but it is most likely that there is an access list on the DHCP server, saying that only devices with Mac address within a defined range will get an IP dynamically.[4] In our case it can be limited to the Allied Telesyn RG213TX boxes, effective within the Mac address range 00 30 84 d0 00 00 – 00 30 84 d0 FF FF + some Cisco 7960 series phones. In reality we don't need to get an IP-address from the DHCP server, since we already know all TCP/IP settings for the RG we have sniffed. Therefore we set IP address on the network device manually.

[4] A brief check with the system administrator confirms this. Of cource something an assaulting party cannot do, but anyway he doesn't have to – this was done only to confirm the thesis.

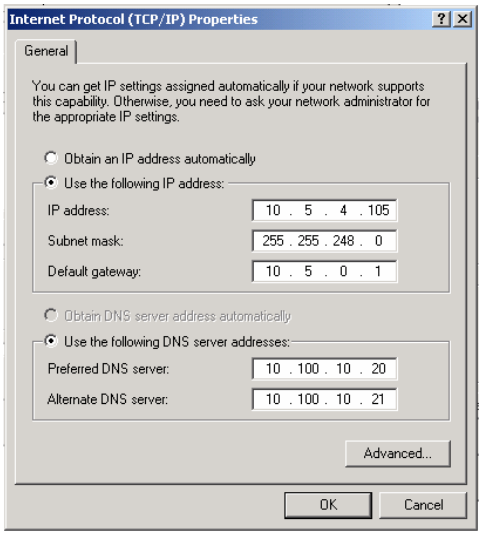


***Figure 56 Assault: Manually configured IP address***

Now we have a network device with an allowed address on the right VLAN. Next thing we need is the username, password and configuration for the SIP server.

### *4.2.4.4 Telnet session with the RG*

We know the username and password for our RG, as well as the IP-address. With a telnet session, we log into the device. There we can run a help command that gives us a list of possible commands. We use the "*show sip*" command, and following information is displayed:

```
 > show sip

SIP information
---------------------------------------------------

Port 0
   Phone Number       38140229
   Authorization      UserName: "38140229"
                      Password: "38140229"
   Domain             ipt-server1.ivisjon.no
   Location Server    ipt-server1.ivisjon.no
   Proxy Server       ipt-server1.ivisjon.no
   TOS                0
   Registered         YES
   Capability         PCMU
                      PCMA
                      G723
   RTP port           dynamic assignment
   RTCP protocol      ON
Port 1
   Port NOT available

---------------------------------------------------
```

***Figure 57 Assault: Allied Show SIP***

From the figure it is possible to read both username, password and server configuration. It is also possible to see that the password policy is very bad[5].

[5] If we assume that this policy has been used everywhere in the network, it would be horrible. In theory it would be possible to just know that a phone number belongs to a client in this network and the password would be the same. That is; you don't even need to break into a phone to get the password, you just do some qualified guessing. Subnets would then not help increasing security at all.

#### *4.2.4.5 Softphone unauthorized call.*

With the information now available, it should be possible to make unauthorized calls. We install an X-lite Softphone, and configure it with the details found in the previous chapter, and outbound / inbound phone calls can be made.


***Figure 58 X-lite Softphone***

### 4.2.5 Summary

It has now been proven that the system has weaknesses, and that unauthorized phone calls can be performed from the client side. We have seen that you can pretend to be anyone on your subnet, and perform calls – letting the attacked person pay the bill. Now we want to take the attack further; by trying to gain access to the IP-telephony server.

## 4.3 Server Analysis

By performing a server attack, we can achieve several things. The first is to mess with caller records and client lists, but we can also make the server stop. The last case can be the worst; disabling thousands of clients to perform phone calls. In cases where we can find services giving us remote access and root privileges; we can operate as administrators of the system. The first thing to do is to choose the proper VLAN tagging, and then perform a port scan.

### 4.3.1 Port scanning the Server

Port scanning is a widely used way to find server exploits. By performing a port scan, you can find which ports are open, and what services runs on the server. If you use a state-of-the-art scanning tool, it can also try different known exploits and scripts; generating a report containing this exploits and how to take advantage of or secure them.

For the scanning performed the "Retina Network Security Scanner" was used. This is a commercial tool, produced by eEye Digital Security. A screenshot from the scan can be found below.

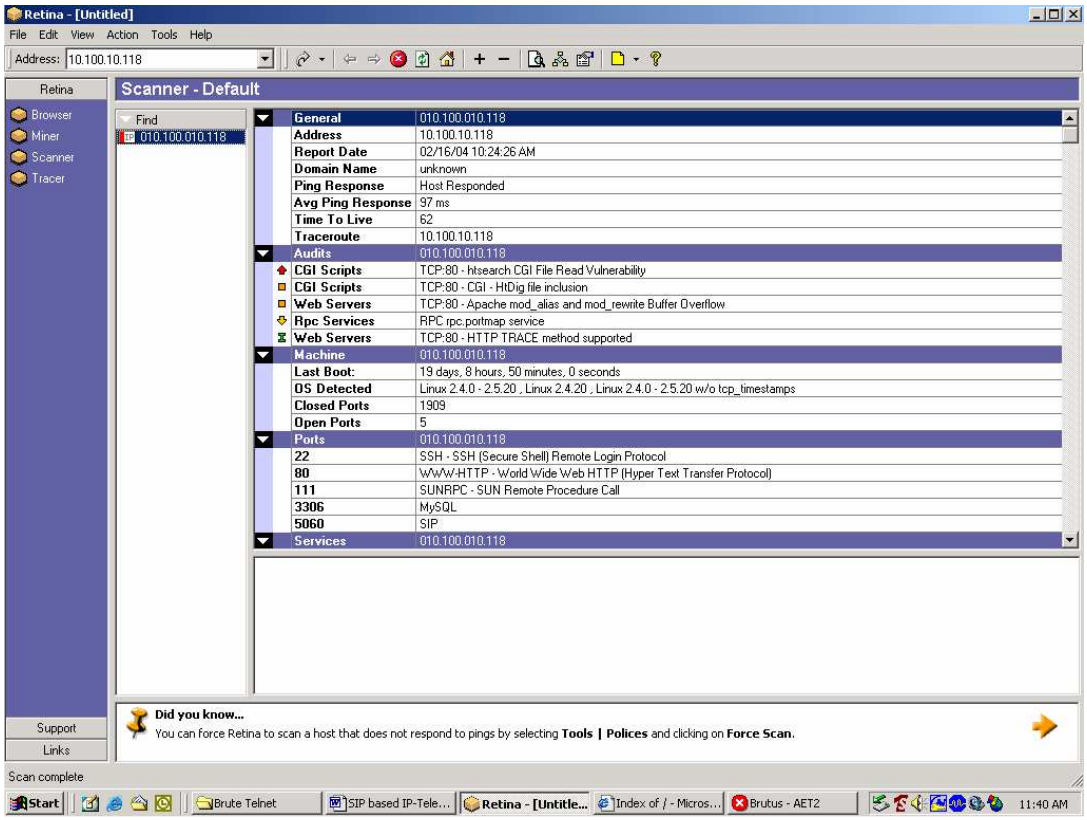


***Figure 59 Retina Network Security Scanner - IP-telephony server security scan***

Further we will list what was found during the security scan.

### *4.3.1.1 Hardware and platform*

**Last Boot**: 19 days, 8 hours, 50 minutes, 0 seconds
No More Details Available

**OS Detected**: Linux 2.4.0 - 2.5.20, Linux 2.4.20, Linux 2.4.0 - 2.5.20 w/o tcp_timestamps
No More Details Available

**Closed Ports**: 1909
No More Details Available

**Open Ports**: 5
No More Details Available

### *4.3.1.2 Ports*

**22: SSH - SSH (Secure Shell) Remote Login Protocol**
Port State: Open
Version: SSH-1.99-OpenSSH_3.7.1p2

**80: WWW-HTTP - World Wide Web HTTP (Hyper Text Transfer Protocol)**
Detected Protocol: HTTP
Port State: Open
Version: APACHE/1.3.28 (LINUX/SUSE) PHP/4.3.3

**111: SUNRPC - SUN Remote Procedure Call**
Port State: Open

**3306: MySQL**
Port State: Open

**5060: SIP**
Port State: Open

#### *4.3.1.3 Audits*

**CGI Scripts: TCP:80 - htsearch CGI File Read Vulnerability**
**Risk Level: High**
**Description:** It is possible for a remote attacker to manipulate the htsearch CGI program in order to access any file within your web server.

**CGI Scripts: TCP:80 - CGI - HtDig file inclusion**
**Risk Level: Medium**
**Description:** The HTDIG:// search engine has had a problem with allowing files to be specified for inclusion into a search query. This could allow a remote attacker the ability to view the contents of any file that the program has rights to read.

**Web Servers: TCP:80 - Apache mod_alias and mod_rewrite Buffer Overflow**
**Risk Level: Medium**
**Description:** A buffer overflow vulnerability exists within Apache 2.0.47 and 1.3.28 in mod_alias and mod_rewrite when they are configured using a regex with more than 9 captures. An attacker must create a specially crafted .htaccess file in order to successfully exploit this vulnerability.

**Rpc Services: RPC rpc.portmap service**
**Risk Level: Low**
**Description:** Retina has detected that the RPC portmapper service (rpc.portmap) is running on the scanned host. Attackers may use information provided by the portmapper to ascertain the host's operating system and identify other possibly vulnerable RPC services.

**Web Servers: TCP:80 - HTTP TRACE method supported**
**Risk Level: Information**
**Description:** Retina has discovered that the target host supports the HTTP TRACE method.

#### *4.3.1.4 Services*

**portmapper: Sun Portmapper Service**
**Port: 111**
Protocol: TCP
Protocol: UDP
Version: 2

#### 4.3.1.5 Summary

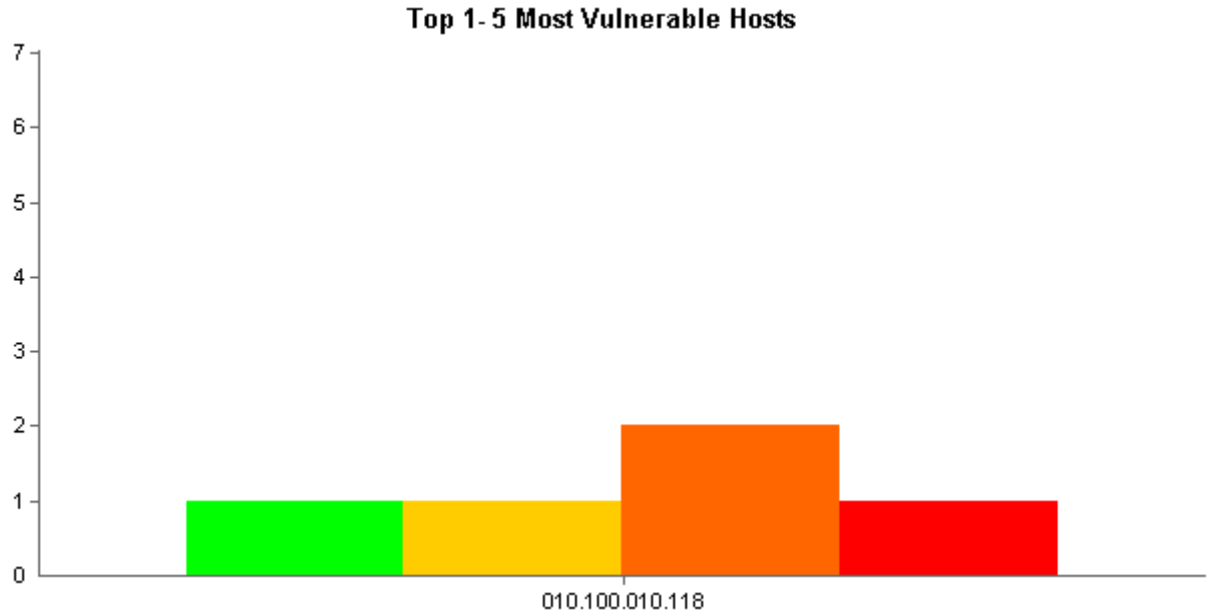


***Figure 60 Telephony Server Risk Level Graph***

Green = Information
Yellow = Low Risk Level
Orange = Medium Risk Level
Red = High Risk Level

### 4.3.2 Analysis of Port Scan Results

From the information given in the port scanning, we can conclude that the server runs a web server, a database server, SSH, SIP and SUNRPC.

Earlier we have discussed SSH [4.2.2.1] and SIP [4.2.1]. Therefore the focus now will be on the Web Server, the Database server and SUNRPC.

#### 4.3.2.1 High/Medium Risk Audits

##### 4.3.2.1.1 HTSEARCH & HTDIG

These commands are web server related. A CGI package called htdig is distributed along with the standard Apache 1.3.28 version. HtDig can run various commands; htsearch and htdig are among the command set.

The htsearch CGI runs as both the CGI and as a command-line program. The command-line program accepts the -c [filename] to read in an alternate configuration file. On the other hand, no filtering is done to stop the CGI program from taking command-line arguments, so a remote user can force the CGI to stall until it times out (resulting in a DoS) or read in a different configuration file.

For a remote exposure, a specified configuration file would need to be readable via the web server UID, e.g. via anonymous FTP with upload enabled or samba world-readable log files are the possible targets) to potentially retrieve files readable by the web server UID.

The remote CGI htsearch allows the user to supply his own configuration file using the '-c' switch, as in:

/cgi-bin/htsearch?-c/some/config/file

This file is not displayed by htsearch. However, if an attacker manages to upload a configuration file to the remote server, it may make htsearch read arbitrary files on the remote host.
An attacker may also use this flaw to exhaust the resources on the remote host by specifying /dev/zero as a configuration file.

The 'htsearch' also allows a malicious user to view any file on the target computer.

**Sample of use:**

nothing_found_file: /path/to/the/file/we/steal
http://your.host/cgi-bin/htsearch?-c/dev/zero
http://your.host/cgi-bin/htsearch?-c/path/to/my.file

#### 4.3.2.1.2 Apache mod_alias and mod_rewrite Buffer Overflow

It is reported that both mod_alias and mod_rewrite contain a buffer overflow. If the administrator has configured a regular expression with more than 9 captures, the overflow can be triggered.

The Apache HTTP server distribution includes a number of supplemental modules that provide additional functionality to the web server. Two of these modules, mod_alias and mod_rewrite provides for mapping different parts of the host file system into the document tree and for URL redirection and a rule-based rewriting engine to rewrite requested URLs on the fly based regular expressions. Several of the `mod_alias` directives can make use of regular expressions rather than simple prefix matches. A buffer overflow has been discovered in the way that `mod_alias` handles regular expressions containing more than 9 captures (stored strings matching a particular pattern). This flaw results in a remotely exploitable vulnerability on web servers that specify such a regular expression to the `mod_alias` module in their configuration files.

### *4.3.2.2 Port related weaknesses*

#### 4.3.2.2.1 Web and Database

The audits given on web and database ports have already been handled. However, we can find more weaknesses by browsing the server root from a browser as Internet Explorer or Opera. In the root directory we can read folders and from that guess services as well as browse some of these folders. Outline of what was found can be seen below.

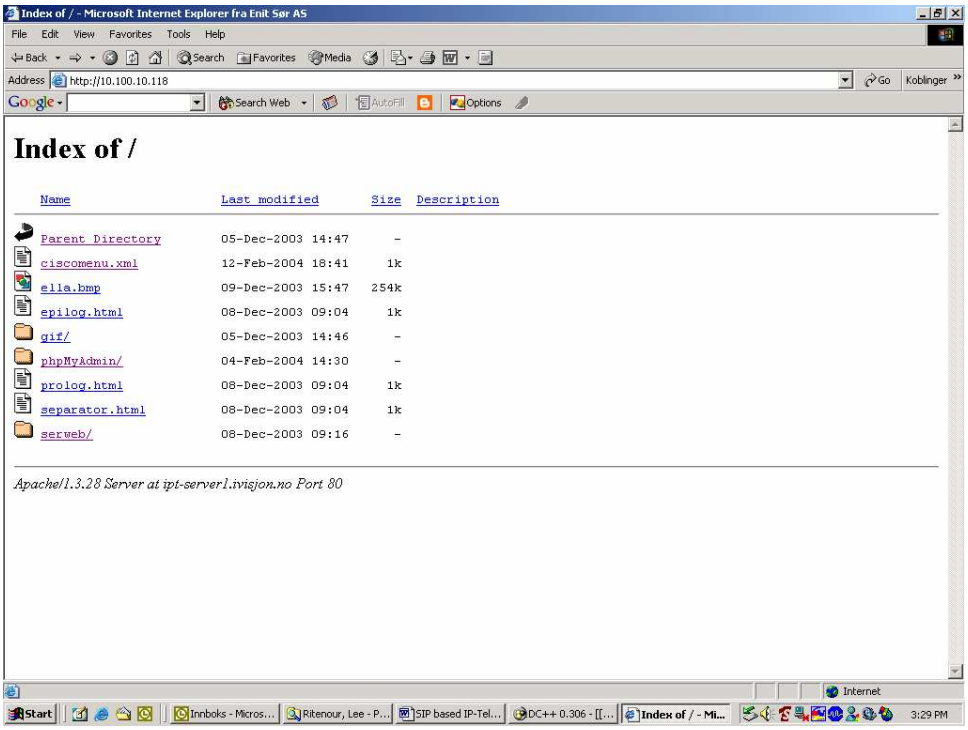


***Figure 61 IP-telephony server Web Root***

As we can see from the figure, the server runs two different tools; phpMyAdmin and Serweb. This in addition to some html/xml files and images. After an evaluation of the single files we find that these are quite harmless. As for the directories, we can find exploitable data.

This will be done in later as we perform an assault [4.3.3.1].

#### 4.3.2.2.2 SUNRPC

By using the command "*rpcinfo –p host*" in a UNIX environment, you can read which services runs on the SUNRPC. In our case the only service is Sun Portmapper Service. As mentioned in the port scan report, this can be used to list services and platform running on the server. In this particular case; trying this command, we found that no risky services ran. The server platform had already been revealed directly in the port scan, so being able to find this gained no new information.

| Service | Description |
|---|---|
| rpcbind | This service name corresponds to the portmapper itself. If you want to run *any* RPC service then you need to be running the portmapper. |
| nfs, mountd, nfs_acl | The Network Filing System. If you want to share some of your files with other systems then you must run these services. `nfs_acl` is the same as `nfs` but has support for Access Control Lists as well. |
| status, llockmgr, nlockmgr | These services are used for file locking over NFS. You need them if you are exporting your own files or importing someone else's. |
| walld | A utility for letting people send messages to every user of the system. Rarely useful and more often a pain in the arse. |
| rstatd | A utility for letting remote systems know your load average. |
| rusersd | A service for letting remote systems knows which of your users are logged on. |
| rquotad | If you export or import file systems with quotas on them then you need to run this service. Otherwise you don't. |
| bootparam | A silly RPC based replacement for `bootp`. If you are a boot server you may need this, otherwise not. |

| | |
|---|---|
| ypbind | All systems in a YP (a.k.a. NIS) domain need to be running this service. If you are not using YP then you should not be running it. |
| ypserv | All YP servers (Master and Slave servers) should be running this service. YP clients should not. |
| tooltalk | This is used for some graphical operations like drag and drop |
| cmsd | This is the CDE Calendar Manager service which is rarely used and was the source of a serious security hole. |

***Table 19 RPC Services***

All RPC services are assigned a service number at system start up. The portmapper service, or rpcbind, is run in order to convert service numbers into TCP/IP usable port numbers and as the RPC server is started up it tells the portmapper what services are being offered on what port. These ports can vary from system to system, but are usually found in the 32770-32789 port range. In essence the portmapper maps the incoming RPC to the port the service is listening on. Once a system is identified as running this service on port 111 the attacker can as mentioned query portmapper with the command:***rpcinfo –p (hostname/IP)***

For all RPC services that are running, the query will return:

- Service Name
- Service Number
- Port Number

For more information on how to take advantage of SUNRPC exploits, try:
http://www.iss.net/security_center/advice/Intrusions/2003016/default.htm

### 4.3.3 Organized Assaults

For the assaults we will take advantage of the knowledge gained during port scanning and packet sniffing. Two separate attacks will be performed; A DoS (Denial of Service) and if possible an attack to gain total control of the server.

#### *4.3.3.1 Denial of Service Attack*

By performing a denial of service attack, you normally want to achieve a stop in the service. There can be several degrees of accomplishment here. A service can be gone for as long as it takes to restart a server, or it can be gone for ours, days or months. The severity is determined by how long it takes to bring the service on-line again.

This assault will take basis in getting into the database administration, and from there delete all field entries, tables, and databases. This way all clients will be erased from the client database, as well as the server configuration. That will bring the server off-line in a serious way; and all configuration data as well as client data needs to be restored before it can come online again. The time this takes depends on backup of the database – but we will also probably loose all traffic data, and new subscriber data from the time of last database backup until the DoS attack. Therefore no records of phone calls (as well as billing) can be found; resulting in a severe economical loss. In addition you can imagine the anger of customers unavailable to perform phone calls.

When trying to enter the database administration tool[6], we are displayed an authentication dialog. The type of authentication used is a basic http method. For an advanced brute-force tool, we should have no trouble finding the username and password. Our tool is called Brutus Authentication Engine Test Release 2, and can be found at http://www.hoobie.net/brutus/

Brutus is an online or remote password cracker. More specifically it is a remote interactive authentication agent. Brutus is used to recover valid access tokens (usually a username and password) for a given target system.

[6] We are now operating on the IP-telephony VLAN.

In the target window of Brutus, we apply the path to authentication directory as well as authentication type (HTTP – Basic Auth) and method (HEAD). Then we choose how many combinations in the username and password file that is to be used. The time it takes to gain access to the username and password depends of how many characters and the range of characters that is used. Anyway we will get the right username and password in the end. Below we can see a screenshot from the Brutus session taken when the right combination has been found.

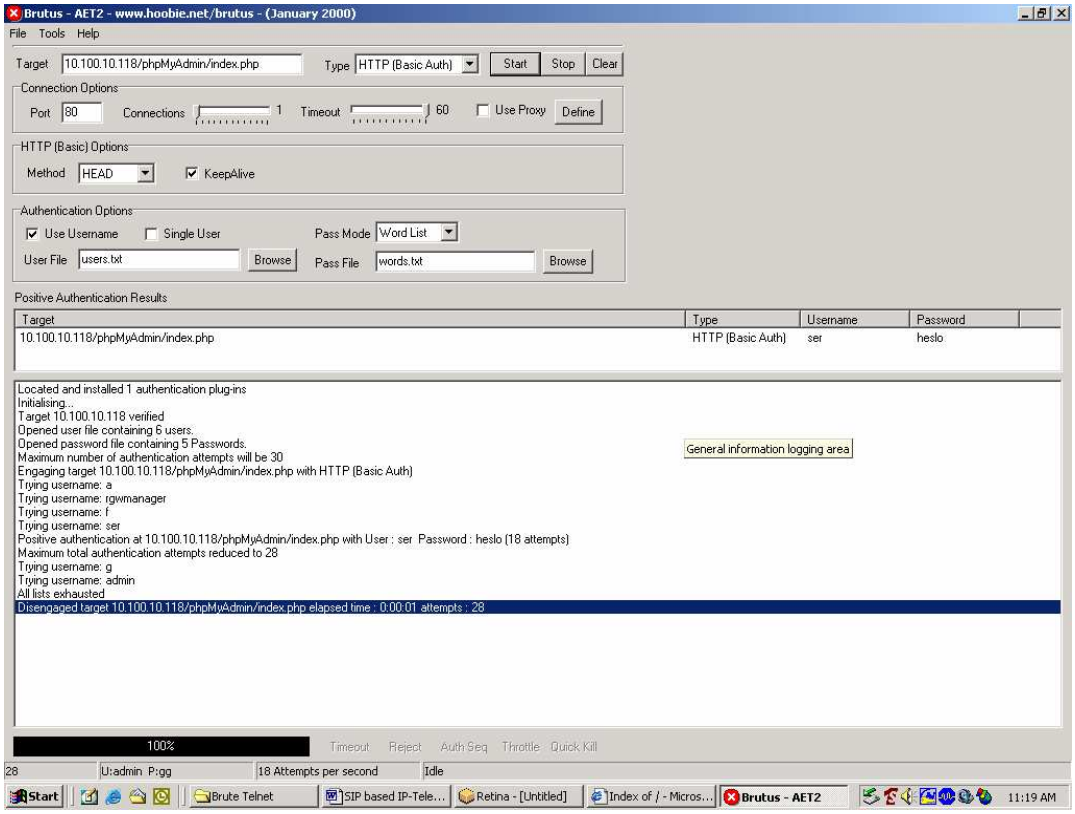


***Figure 62 Brute HTTP (basic) web login***

Now we know the username and password of the mySqlAdmin tool. The next thing we want to do is of course to login. When logged in, we can decide on which database to browse, and then also which tables and entries we want to look at. From a short run in the database administration tool, we find that all users as well as passwords and other user / server information can be listed as well as changed. When trying to add a user, we fail – cause users can only be added through a SSH session in command line interface. This failure is caused by an encrypted key field added by this command in command line interface. Therefore we now know that to gain a total control of the server we need to hack SSH. Anyway, our mission is to stop the server for as long as possible – so we use the interface to drop all tables and delete the database. Now the server is completely messed up, unsuited to understand any request.

Notice that bruting passwords can be a process that demands a lot of time. In our case, the customer is located on the same fibre network as the server, and therefore the process speeds up. Some servers can also set limitations on how many tries we can perform until our login gets rejected / banned.

Brute Force attacks are performed by either dictionary attack, or by trying combinations. For the dictionary attack, all words used in the attack are picked from a dictionary. Such dictionaries can be downloaded, and often contain a set of the most familiar usernames and passwords. If you brute by trying combinations; all combinations of letters and numbers are tried. This normally takes a very long time, but will eventually succeed.

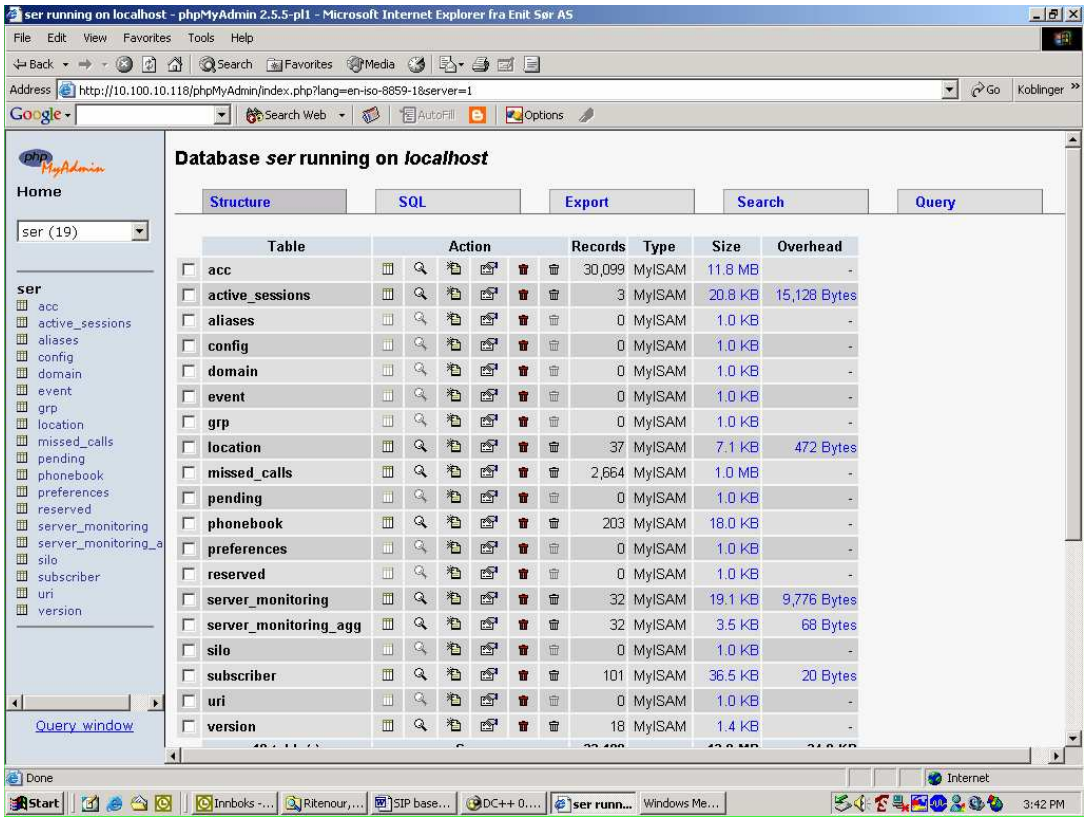


***Figure 63 sqlMyAdmin Interface - hacked***

As described earlier in this chapter, the sever ness of this attack depends on when last database backup was taken, if taken at all.

To understand which attackers that can gain on this DoS attack, two cases can be mentioned:

1. A competitor that wants to ruin our company – gaining our clients
2. A user that has done a lot of high cost phone calls, and that would like not to be billed for them.

If we try a similar attack towards the serweb directory on the server root, we can only read some caller records and perform listing of clients; but no alteration can be made.

We have now proved that the server can be made unavailable for all clients for a certain amount of time.

### *4.3.3.2 Gain Total Control of Server*

To gain total control of the telephony server, we need to use the weaknesses revealed in the port scanning. The HTTP services, we have already found that can stop the server, but else we cannot take remote control of the server by this audits. Though we can list all files below the web server root, this does not give us further information on the system.

This way we are left with the SUN RPC and the SSH as means to gain control. Remote procedure calls (RPCs) allow programs on one computer to execute procedures on a second computer by passing data and retrieving the results. RPC is therefore widely used for many distributed network services such as remote administration, NFS file sharing, and NIS. However there are numerous flaws in RPC which are being actively exploited.

If you use vulnerability scanner or the 'rpcinfo' command to determine if you are running one of the most commonly exploited RPC services:

| RPC Service | RPC Program Number |
|---|---|

| | |
|---|---|
| rpc.ttdbserverd | 100083 |
| rpc.cmsd | 100068 |
| rpc.statd | 100024 |
| rpc.mountd | 100005 |
| rpc.walld | 100008 |
| rpc.yppasswdd | 100009 |
| rpc.nisd | 100300 |
| sadmind | 100232 |
| cachefsd | 100235 |
| snmpXdmid | 100249 |

***Table 20 RPC exploit service list***

RPC services are typically exploited through buffer overflow attacks which are successful because the RPC programs do not perform sufficient error checking or input validation. Buffer overflow vulnerabilities allow an attacker to send unexpected data (often in the form of malicious code) into the program memory space. Due to poor error checking and input validation, the data overwrite key memory locations that are in line to be executed by the processor. In a successful overflow attack, this malicious code is then executed by the operating system. Since many RPC services execute with elevated privileges, successful exploitation of these vulnerabilities can provide unauthorized remote root access to the system.

If trying to exploit the SUN RPC port map service, we gain nothing but what was explained in chapter [4.3.2.2.2]. That is none of this listed RPC programs ran.

Since we gained nothing by using the RPC service, we will have to look into the SSH, and try to exploit this one. Secure shell (SSH) is a popular service for securing logins, command execution, and file transfers across a network. Most UNIX-based systems use either the open-source OpenSSH package or the commercial version from SSH Communication Security. Although SSH is vastly more secure than the telnet, ftp, and R-command programs it is intended to replace, there have been multiple flaws found in both implementations. Most are minor bugs, but a few are major security issues that should be repaired immediately. The most dangerous of these actively exploited holes allows attackers to remotely obtain root access on a vulnerable machine.

The reason for SSH having an exploit is more the case of a mismanagement of SSH, specifically misconfiguration and the failure to apply updates and patches in a timely manner.

SSH2 is actually a powerful tool that when properly configured and maintained can help those services that send material in clear text across untrusted networks like the Internet. Many of the vulnerabilities found in protocols such as POP3, FTP (replace with SSH2s SFTP), Telnet, HTTP, and the rhost based tools (rlogin, rcp, and rsh) involve eavesdropping on clear text transmissions or manipulating client server sessions. This makes encryption and authentication key management provided by SSH2 along with its ability to forward or redirect sessions, an attractive VPN type of wrapper for otherwise vulnerable traffic.

The SSH1 protocol itself has been demonstrated to be potentially vulnerable to having a session decrypted in transit given certain configurations. For this reason, administrators are encouraged to use the stronger SSH2[7] protocol whenever possible.

To find if your system is vulnerable, we use a vulnerability scanner to see whether you are running a vulnerable version, or check the software version reported by running the command 'ssh -V'.

[7] SSH1 and SSH2 are not compatible. With only a few exceptions, the version of SSH on both the client and the server must match. Some implementations allow fallback to earlier version (From SSH2 to SSH1).

The ScanSSH tool is particularly useful for remotely identifying SSH servers that are dangerously un-patched. The ScanSSH command line tool scans a list of addresses and networks for SSH protocol servers and reports their version numbers.

This tool is available at http://www.monkey.org/~provos/scanssh/. When we run this tool, we find that the IP-telephony server runs ***Version: SSH-1.99-OpenSSH_3.7.1p2***. This is the latest version of OpenSSH, and after a wide Internet search, also using contacts within System Sikkerhet[8] in Norway and FFI[9]; we find that there does not exist any known exploits with this version of SSH.

The options now are few, but a scenario could be to use the ScanSSH tool to find switches on the network using SSH or telnet. If switches ran SSH1 or telnet, it would be possible to gain remote access to it. We might find that someone had used a switch and connected to the telephony server from the switch. This way a log of last used commands could result in a SSH connection from switch to IP-telephony server, allowing us to read the username and password, or to go directly to the server. However, none of the switches can run SSH clients; so this is not an option.

We can conclude that this attack will not succeed with the knowledge we now have gained. To get access, we will somehow need to find a document describing all usernames and passwords for the network. To find such a document for a remote attacker is not very likely.

**Therefore we can conclude that with our knowledge; it is not possible to gain remote control of the IP-telephony server.**

Anyway, we would like to show that such an attempt could be successful in some cases. Therefore we will now work on an alternative scenario, where the SSH1 runs on the server we want to take control of.

#### 4.3.3.2.1 Alternate scenario

Servers running SSH1 and early versions of SSH2 with fallback to SSH1 have a number of documented exploits. The most severe is the CRC32 compensation attack detection function. If we uninstall the current OpenSSH version on the IP-telephony server, and exchange it with OpenSSH 2.2.0p1, an attack should be made possible. Many servers run this OpenSSH, and the scenario can therefore be defended.

We will not perform an attack, but point to earlier successful attacks. A link to the source for the attack can be found at: http://staff.washington.edu/dittrich/misc/ssh-analysis.txt

The SSH CRC32 Compensation Attack Detector Vulnerability was actually inserted in sshd to compensate for a deficiency in the SSH-1 protocol. The exploited code watches for an attempt to attack the deficiency.

The attack detector creates a dynamically allocated table in memory to store the connection information it uses to detach an attack. Using a crafted packet, it is possible to create a table with zero length and to then push data into the zero length table, overwriting memory including the functions return address. As soon as an intruder can change a functions return address, he can run any code and use it to open a shell running with the privilege of the sshd daemon (usually root). All this information can be found in public on the Internet.

From the source, we can read that successful attacks have been performed up to OpenSSH 3.0.2p1. This audit has been locked for later versions, so our server running 3.7.1p2 is for now secure.

[8] System Sikkerhet ASA is one of the largest Network Security companies in Norway. Their clients span from IT companies with a confidentiality demand, to Internet Banking systems and the Norwegian Army.
[9] Forsvarets Forsknings Institutt (Norwegian Defence Research Establishment)

To exploit the vulnerability, you first need to get the SSH daemon version. This is done as we earlier have described, by using the ScanSSH tool. We will then be prompted with the version running on the server we try connecting to. If the reply is OpenSSH version 3.0.2p1 or earlier, an attack will be successful; giving you root access to the server through a shell.

In the shell we can browse the files and folders, and this way find the command enabling us to add telephony users, as well as defining additional usernames and passwords for the server. This then enables the later use of a straight SSH2 client login, with root rights. It is important to create a new user, since the attack only creates a hole in the system for about 10 minutes.

Now a total control of the server has in theory been gained.

## 4.4 Unauthorized listening to calls

For SIP IP-telephony the setup of phone calls, billing and disconnecting (equals signalling in the ISDN world), takes place between clients and the servers. However, for the speech data a different way is applied. The data of a call goes from client to client, without the servers involved. The data stream uses RTP for transport. Therefore, if we want to listen in on calls, we need to listen for packages between the clients, using a traffic analyzer that can analyse RTP data.

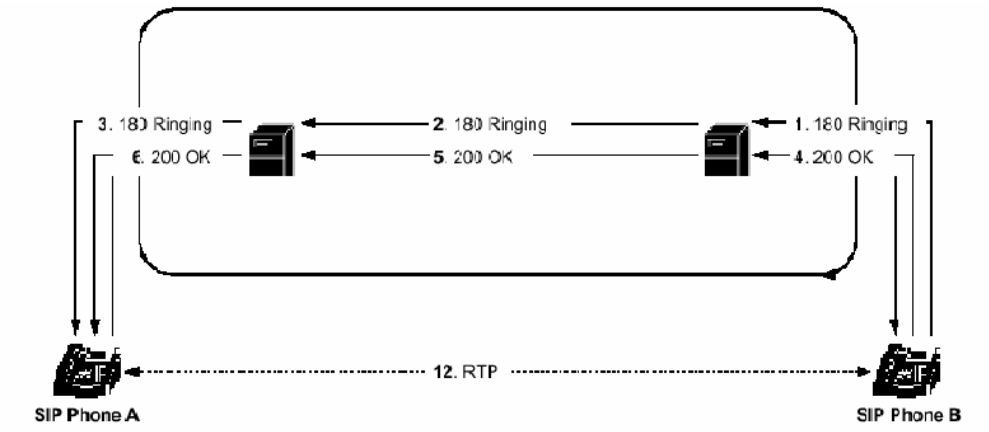


***Figure 64 Data load of a SIP conversation uses RTP. Source: Cisco Systems Inc***

We have earlier used the Ethereal Network Traffic Analyzer to sniff TCP/IP and SIP packages. Ethereal also decodes RTP data, and has a utility that translates this data into audio files (AU - format).

Therefore we start the packet listener, and make a phone call with a client that sits on the same network as the traffic analyzer. When the analyzer has run for some time (as well as having confirmed a phone call), we stop the packet sniffing.

By clicking: "*ANALYZE -> STATISTICS -> RTP STREAMS -> ANALYZE*", you will be displayed the dialog shown below.

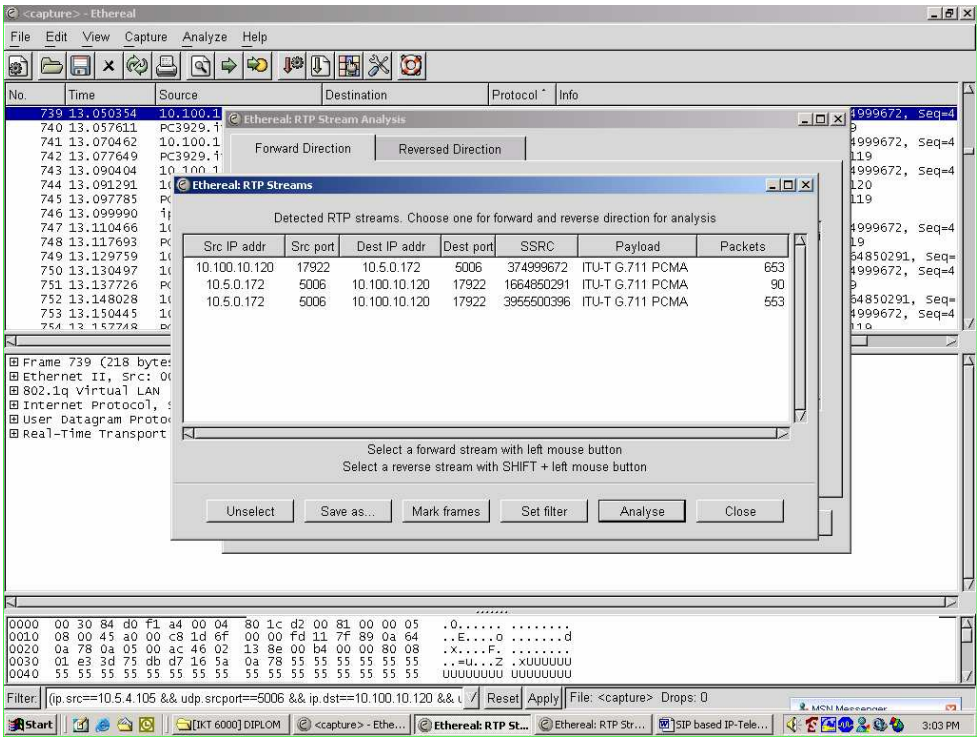


***Figure 65 Ethereal RTP analyzer function***

In the figure above it is possible to choose from the different RTP streams. If you highlight one of the streams and save that particular stream as an audio file (Figure 66), you can listen to it in an audio player i.e. windows media player or similar.

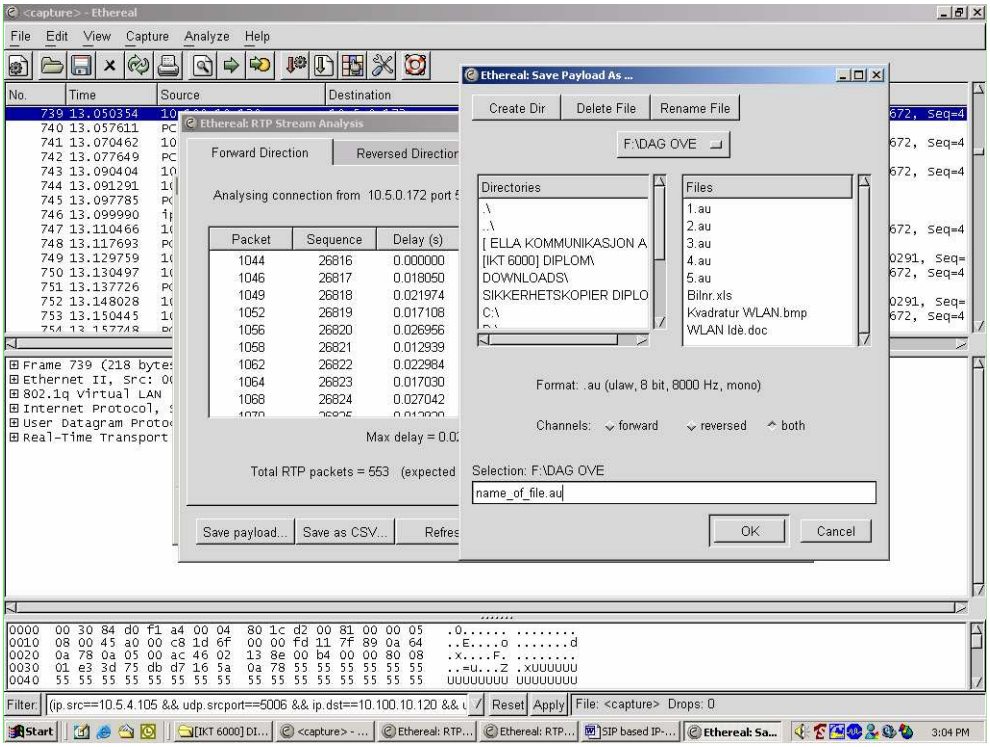


***Figure 66 Saving RTP streams as Audio files***

The RTP stream has been sent unencrypted, and therefore it is possible to easily decode. This way we have seen that it is possible to listen to calls made by others, with the limitation that they need to be in the same net as the analyzer software. Anyway, if you are located within the core

of the network that will not automatically give you access to all conversations; since the data stream travels from point to point (multipoint if a conference call). That is, if you call your neighbour (same access switch), a person listening further into the network core cannot listen to that particular call.

If you want to control more than your neighbours, you will need to access the access switches (typically HP 4000) on the management VLAN (this is known by us from earlier sniffing; username and password should not be a problem). In the switch you need to mirror the VoIP VLAN, connecting a listener physically to that switch and listen on the mirror port. However this is further complexified by a rule that says only a set of IP-addresses are allowed to connect to the switches. All this complex actions makes it unlikely for anyone to succeed.

An alternative to the Ethereal analyser is PacketScan from GL Communications Inc. This is a commercial alternative that enables real-time listening to SIP calls, but costs 4795$ and was therefore unavailable for this project.

## 4.5 Summary

Organized assaults have been performed from an attacker perspective. The assault discovered serious weaknesses with the system. These flaws resulted in mainly three issues:

1. An attacker can perform unauthorized phone calls
2. Denial of Service Attack successfully performed
3. Limited listening / recording of phone calls possible

In short this means that almost all aspects with the telephony service have been compromised. The experiment to gain total control of the service failed. But in total, that was the only strength uncovered during the tests. This strength was caused by an updated version of SSH. It could just as well have been an older version that could have been successfully attacked as described.

# Discussion

In this chapter we will discuss the security flaws, and how it is possible to improve security in our scenario. As the weaknesses were encountered during experiments from an attacker perspective, it is natural to look at improvement in order of the attacks. Therefore this chapter is divided into five different sub chapters. One for each of the attacker goals, and in the end one that covers VoIP traffic in general, and in what degree security weaknesses can be accepted in such systems.

## Unauthorized phone calls

Essential for the unauthorized phone call is the ability to sniff in on network traffic. The main reason for its success, were the network configuration; using VLAN's as a security mechanism, along with getting telnet information in plain text.

When sniffing traffic, we had to physically connect to the network. Therefore, one way of improvement can be to change interface type, making it harder to connect to the network. Since sniffing had to be done on the client side, we need to look at hardening opportunities in that end of the network. In the house, a client has an Allied Telesyn Residential gateway, connected to a fibre converter with category 5 cables. The fibre converter then connects to an access switch with a fibre interface. In my opinion, fibre is a more difficult medium to sniff, and therefore we could exchange the cat5 cables with a pure fibre solution.

Allied Telesyn has a Residential gateway with fibre interface. This means that a client will have the Residential Gateway as his final interface between the house and an access switch. Therefore, by switching to the fibre model, it will be harder for an attacker to achieve his mission.

However; if an attacker is determined to get access; no matter cost and effort, it will still not be totally safe. The reason for this is that the attacker could buy himself two fibre converters, and then change interface, put in a hub and listen in on traffic again. Se model below [Figure 67].

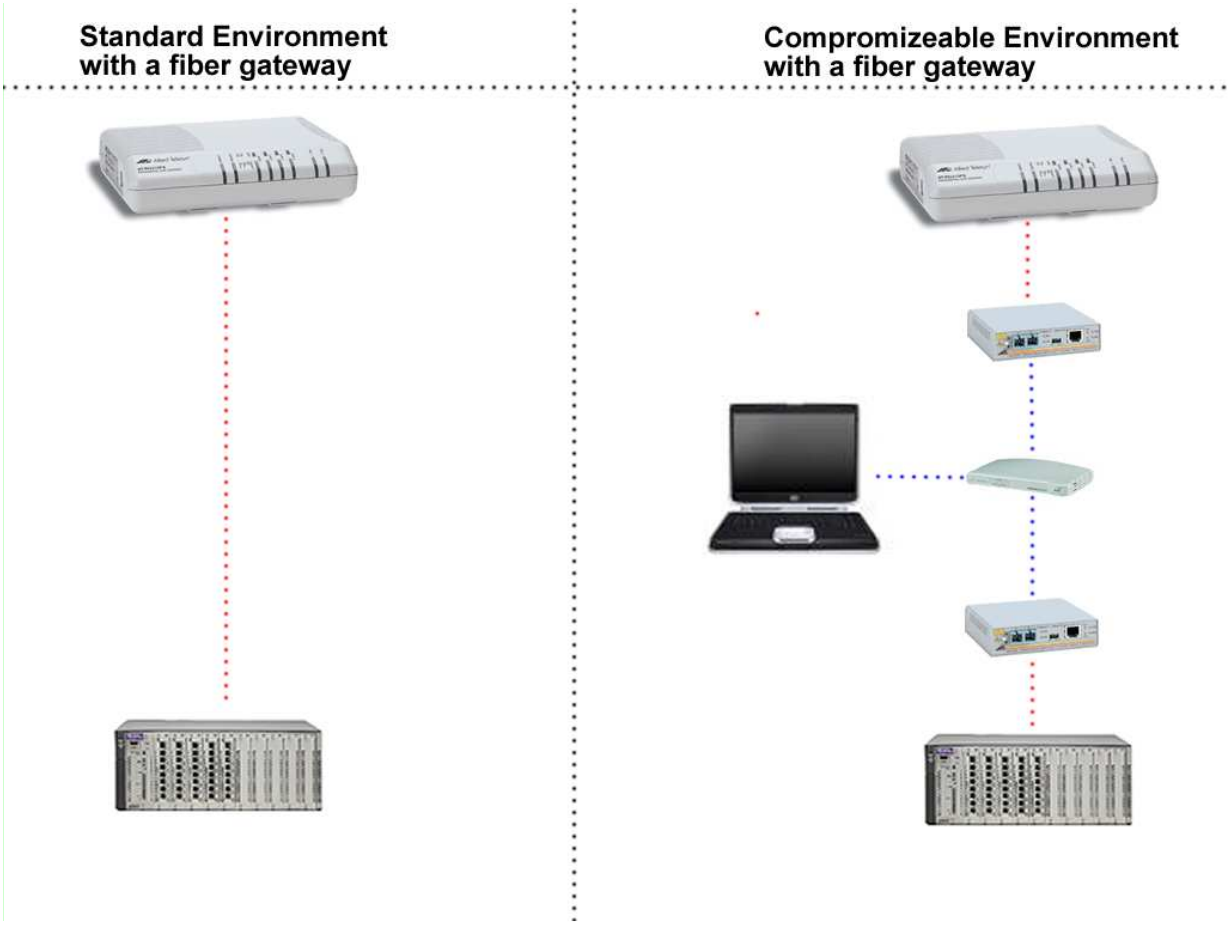


***Figure 67 How to sniff traffic on a fibre environment***

In the left part of the picture, you can see a red dotted line going from the Allied Gateway directly to the access switch (HP Procurve 4000). That particular environment has been made sniff able in the right part of the picture. It works like this: A fibre connection is terminated into a fibre converter (Top – down). This fibre converter produces a copper environment, with TP cables into a hub (blue dotted lines), with a laptop listening in on traffic. The wan port of the HUB is connected to a second fibre converter, connected to an access switch with a fibre interface. In this environment a traffic analyzer as Etheral, can sniff all data traffic.

The price of each fibre converter is quite high, so it is very likely that such an exchange would at least cut of the amateurs from trying out the network weaknesses. Anyway, this will not prevent a professional attacker from trying. Therefore, we need to go to the next level; securing the traffic on the network.

Say that an attacker has gained access to sniff the network. If the data on that particular network is sent in plain; unencrypted, it will be easy to take advantage of the knowledge within the data stream. In our case, this was the reality. We were able to get all subscriber data for our phone call, because we sniffed telnet username and password for the Residential Gateway (phone), and in its configuration we found phone username, password, proxy server and so on. Therefore, even if the phone authentication towards the proxy itself was encrypted, we found the information needed elsewhere; in the allied box.

If we had not found the authentication data by breaking into the Allied gateway, it would have been much harder for us to achieve the unauthorized phone call. The key weakness here would be that the gateway uses telnet to communicate with administrators. Telnet offers no kind of encryption, and usernames / passwords are sent plain. In our case, anyone who could listen in on the network would eventually get the username and password for the gateway.

As we experienced during the server attack, it is much harder to break an SSH connection than a telnet connection. So the proper thing to do would be to implement SSH to all gateways, and use this protocol for administrative communication instead of using telnet. By doing this, we would be secured quite well. This since our only option now would be to break password from the SIP authentication message; and this one is encrypted using digest. Some of the gateways had phone configuration saying authenticate with basic "encryption". These devices would still be possible to impersonate. If they had been upgraded using digest for authentication, we would have made it much harder for an attacker to succeed.

If we look back at the scenario, and forget the suggested improvements; more things can be said. For example a note on the password politic employed. In our telnet attack at the gateway, we could read the phone feature configuration [Figure 57].

As seen, the password used for authentication is the same as the phone number, only backwards. It is likely to believe that the same politic has been applied all through the network, leaving us with a list of all user authentication data for all customers within the whole VoIP solution. That is; if you can get hold of client list, or get knowledge of who has a phone from this company; you will automatically be able to "transfer your bills" to other subscribers.

On the phone VLAN, all devices got IP-addresses from a DHCP server. Without an IP-address within the correct range, connection to the SIP proxy could not be made. This is a good thing, especially since no devices could get an IP-address from the DHCP server unless they had a mac address within a predefined range: the range of mac addresses for Allied Telesyn Residential Gateways deployed in the network. We went around this problem by reading the DHCP message on the network, sent to the gateway from the DHCP server. These messages are broadcasted on the subnet that the specific client request is sent from. So in general; I can get hold of the IP-address for all gateways on my subnet. This way I can "get into" all gateways on my subnet with the IP information, or I can simply guess IP's. However, to subnet can also be a security mechanism. This can be explained by thinking of very small subnets, and that DHCP broadcasts will only be sent to very few clients on a small subnet. In the test case, we operated with a very large subnet, so to make a new subnet politic could improve security.

In addition to the fact that you can sniff fewer IP-addresses, it would also be possible for a network "police officer" to easily find where an attack most likely came from. To understand this, you can imagine the entire network being one subnet. If you knew an attacker's IP-address, he could have been sniffing this IP on the network, making it possible that this person could be located anywhere in the network. On the other hand, if the attacker was a member of a very small subnet, say 10 clients – it would be easier for us to guess 1 from 10 instead of 1 from thousands.

## Denial of Service

In our Denial of Service experiment, the main method for getting system knowledge and following DoS success was a port scan. This scan discovered what services ran on the server. This information was then used to exploit weaknesses in the running services.

We can look at hardening the system using several different approaches. One can be to remove or close unnecessary services from the Sip Server Express VoIP server. The other would be to limit access by applying a firewall solution.

We can look at the first approach first, even though the last one is more likely to do us good.

The services found where: mysql database server, web server, sun rpc and ssh. The attack was performed by a combination between database server and web server. Main reason for success was a database management web interface running on the apache web server, called

phpMyAdmin. This interface demanded a login, and if successful login was achieved, an attacker could flush all database tables – controlling the Sip Express Router, and stop the server completely (also removing all caller detail records).

First weakness in this chain would be the web server. If we analyze the system requirements, we find that the entire server could run without a web server; by using SSH and command line functions for user management, instead of a web interface. By disabling the web server, you would also remove the next weakness in the chain; remote database access.

However, say that the web server is a "must have," and that it therefore cannot be removed. We would have to move further into the chain to strengthen the web applications running on the web server.

First we have the client interface tool called serweb. This application is secured by a login; applying cookies as a security mechanism. I would prefer to have this application removed if possible. However, this is only an interface for reading data, and no changes can be made to data.

Second we have the phpMyAdmin web application. By getting access to this application we stopped the entire server. Therefore this should be removed. The application is intended for no other users than administrators. If needed, the same service can be found using SSH and command line functions. If it as earlier discussed is a demanded application, we need to at least strengthen the login for this application. By default, it uses basic http login, and this is a login type we have attacked earlier, using a brute force tool. As in our case, this was successfully done. In the phpMyAdmin distribution, it is possible to select type of login: http basic, cookies or sessions. If we use cookies or preferable sessions, an application would handle the login. This way the login would not be done in a basic way, where the attacker knew all parameters. Therefore it could be seen as a hardening of the system. Sessions is to be preferred because no data is stored with the client device.

If a successful login is performed, the DoS attack will succeed. This because the web application runs as localhost. In the mysql database server configuration it is set a range of IP-addresses or domains that can access the database server. In our case this is defined as localhost – a good policy for database servers. However, as mentioned the web application runs as localhost, and access is therefore granted.

If security should not be compromised, the phpMyAdmin must be removed. If not, we need to run a firewall and do some filtering from there. This naturally leads us to the next approach; firewall connected to the Sip Express Server.

With a firewall, we can disable the possibility for an attacker to port scan the server computer. In a scenario where we close all ports except the pure SIP ports, we would not disable any running services on the server; we would just make them (except SIP) unavailable for any client. You might miss the SSH port in this sort of firewall configuration, but that does not necessary need to be a problem. This caused by the fact that in most firewalls, a predefined range or list of IP-addresses allowed to connect on the SSH port can be defined. This way you will secure increasingly from leaving SSH open for everyone. If applied, you can also give access to the web server or all services for this range of IP's. In the SuSe distribution running on the Sip Express Router, at least two software firewalls are included. Either IPChains or IPTables can be used, achieving the same result.

If you want to add even more security to the system, it is possible to add an IP-list corresponding to the IP-range of the DHCP server(all residential gateways) as a filter on the firewall, and closing all ports on the Sip Express Router for all clients, except given ports for administrators, and SIP ports for the residential gateways. This way we get security even more increased. In our case the attacker faked an accepted ip-address, so in our test this would not help any further.

Before we go on to the encryption solutions available for SIP client-server communication, we need to look a little bit further into the DoS attack. As mentioned, the database tables were flushed, and therefore all caller detail records, as well as client information was erased. For the Sip Express Router, several configuration alternatives leaving this possibility out are available. Say that all caller details did not exist in the mysql database, but a third party server handled these matters. Then it would not be possible to flush tables, not existing... If you run the built in radius function in the server, billing and authentication could be handled by a third party billing solution and an AAA server for authentication. An encrypted pipe between the radius server and SER could be established, leaving SER running as an interface or router only; not handling any internal client data. In addition to the security matters, an instant billing and client administration interface can be established. As an example, you can authenticate both PPPoE internet access requests and VoIP requests with radius towards a third party billing and client system, including SER to the big picture. A lot of such business solution exist today, i.e. Mind solutions that has a product based on the configuration discussed, successfully running in a SER environment.

## Total server control

In our experiment to achieve a total server control, we failed. This due to an updated version of the SSH service. The other services running on the server, was not possible to exploit giving us a total server control. As mentioned in the previous chapter, an integrated software firewall solution as IPChains or IPTables would decrease the possibility to take control of the server.

For the matter of discussion, we can imagine a scenario where the SSH service ran on an old breakable version of OpenSSH, as mentioned in [4.3.3.2.1]. If an attacker had gotten SSH access as i.e. root, the attacker would have a total control of the server. That is if it ran the configuration running billing and authentication internally with the mysql database. It would then be possible to start/stop the server, add, remove or edit users, groups and locations. All this through the serctl command line interface controlling the Sip Express Router.

Anyway, if Radius had been used to communicate billing and authentication data with an external system, this would not be that easy. The main possibility would be to start and stop the server, but client data would be unavailable for alteration. In addition, quite an advanced piece of hacking would have to take place to get access to the radius server, if ports had been closed as they should with the billing / auth. server.

In our experiment to gain control of the server, we experienced one of the most important issues for host masters and server personnel; to keep your software updated with the last security patches or upgrades. If this had not been done in our case, the server would have been totally compromised. This is a key method to keep security on a high level. It is understandable that managers cannot look around for all upgrades all times, therefore a range of e-mail lists are available for subscription. Managers will then get notifications whenever bugs occur or security patches are released. Else, it would be possible with the SuSe 9.0 distribution running the X-windows interface, to have an automatically software or fix notification. Like the windows update for Windows 2000 Professional and XP, the latest SuSe distributions has a similar application. This will notify you whenever an upgrade is available for installation. However, this is limited to the software included in the platform, and not for the Sip Express Router distribution.

## Listening to phone calls

In chapter [4.4] we can find how listening to phone calls were successfully achieved. Main problem for an attacker is that traffic travels from client – to – client. In the network of Èlla Kommunikasjon AS, listening to any call within the client range of the SIP server is not straight forward. This can be illustrated as on the figure below.

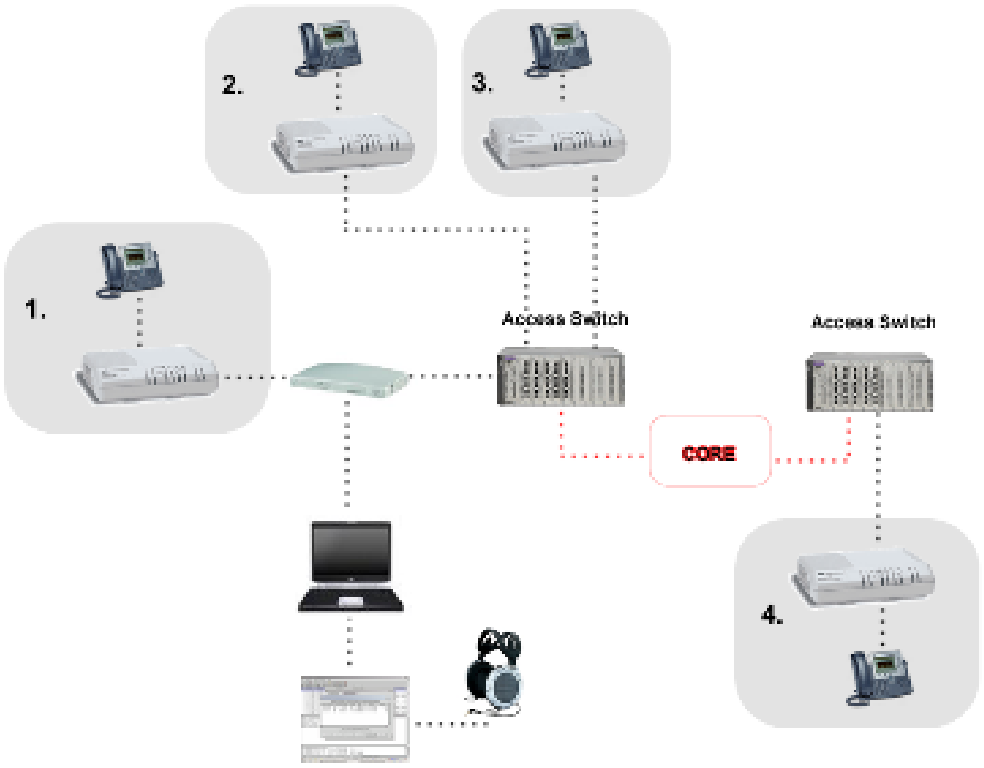


***Figure 68 Listening to Phone calls in Èlla Kommunikasjon Switched network***

In the figure above, we operate with 4 clients (phones /residential gateways). The first 3 are connected to the same access switch, and the fourth is connected to another access switch somewhere else in the network. On the first client we have a computer listening to all VoIP traffic.

The listener will only be able to hear VoIP traffic from client 1, to all other clients in the network – or all phone calls from any client to client 1. It is not possible for the listener to hear VoIP traffic between 2 and 3 or similar. This is due to the fact that the RTP streams are point-to-point, and therefore not available for sniffing for client 1.

Even if the listener was located within the core, traffic between 1, 2 and 3 is not listenable. However, if client 4 made a call or received a call from 1, 2 or 3; it would be possible. For an attacker to achieve this, it would be necessary to physically get into the core. That is the core server room or a central place of traffic. It is likely that the listener would get access to more phone calls the closer he gets to the core.

If we open the gateway to the Internet (through the Cisco 2651 Router), or redefine our network so that all RTP streams need to go through the core; the number of listenable phone calls within the core will increase. The figure below illustrates that.

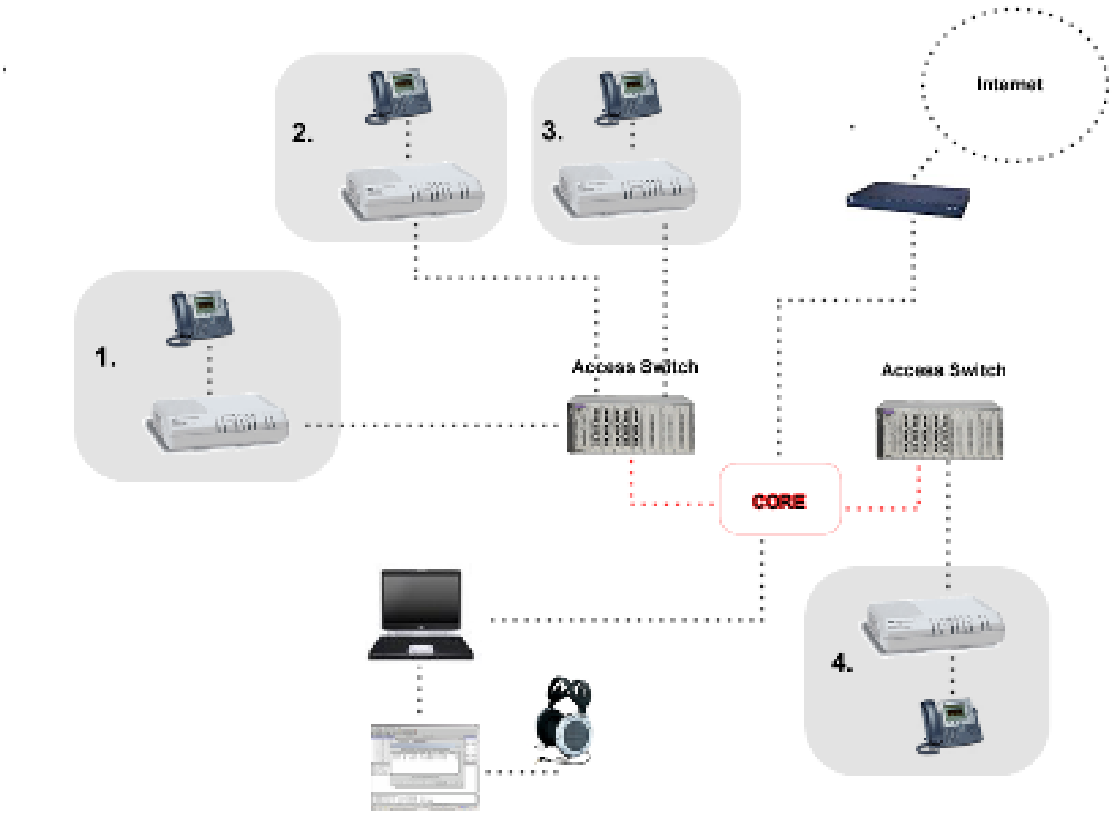


***Figure 69 Listening to Phone calls When All Traffic goes through Core***

In the figure, all RTP traffic needs to go through the core of the network (not the real case, but for sample purposes). If listening in on traffic within the core, we can now ”hear” all phone calls; both internal, and external (internet) to internal clients.

In both cases, RTP streams are unencrypted, and therefore vulnerable for sniffing. The solution to this can be to use SIP phones that support encryption. I.e. the ZIP 4x4 hard phone from Zultys.

***Figure 70 SRTP encrypted ZIP 4x4 hard phone from Zultys.***

The ZIP 4x4 phone can use Secure RTP (http://www.ietf.org/rfc/rfc3711.txt) to secure the RTP stream. It uses encryption to transport voice traffic in a secure manner. For this particular phone, users can engage the function before or during a call by pressing an encryption button on the phone.

What happens is that the speech is encrypted by 128-bit AES. As mentioned before, the encryption counts for client to client communication, and does not affect the client to server communication.

To use the encryption, you will have to use a 4x4 phone or other phone supporting the same encryption and transport methods. The figure below illustrates this.

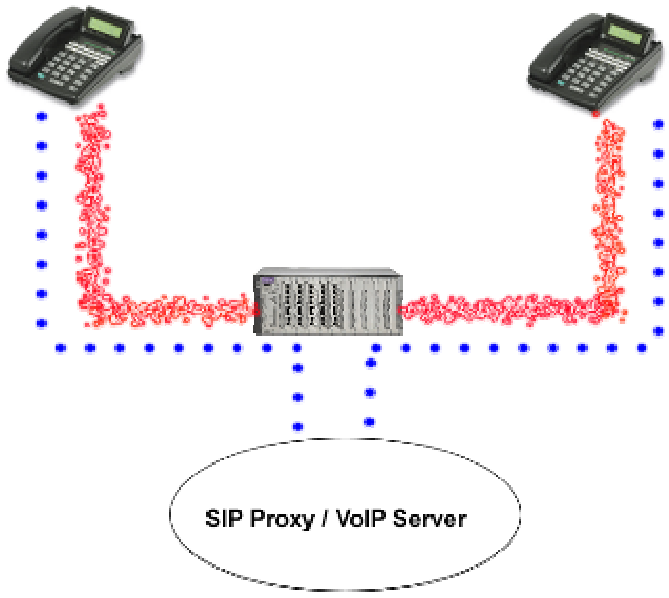


***Figure 71 End to End Client Encryption AES***

The blue dotted line illustrates signalling between i.e. Ser Express Router and the clients. This signalling is unencrypted, if we ignore the authentication sequence that is encrypted by Digest. Information sent here is like signalling for ISDN equipment (Duration of calls, getting recipient address and so on).

Else the red distorted line illustrates an end to end SRTP stream (the voice traffic between clients), encrypted by AES.

Listening to the voice traffic will now result in for an attacker non understandable data.

In our particular case, it is impossible to apply such a service, because the SIP phones in the Allied residential gateway have no support for this feature. A solution could be to disable the SIP phone in the residential gateway, tag traffic with the proper VLAN on one of the LAN ports, and connect a 4x4 or similar phone to this port. Other encryption methods as TLS or IPSec can be applied to some phones as well.

If we compare to the level of security implemented in PSDN, we will find that the communication between clients, are not encrypted. There are however some client equipment that can add this service to a phone call, just as with the SIP equipment we have discussed. Such services are mostly used by government officials on a high level (I.e. the prime minister), to ensure an acceptable level of privacy. For other citizens, these services are rarely used.

It is in the end a matter of policy, for the suppliers to decide on security level. Some would say that the same rules should apply for VoIP as for ordinary telephony. In my mind, every computer is a possible tool for analysis and exploits, therefore I would say that other policies should apply for VoIP. I.e. how many attack tools are connected to the PSDN directly at all times? Probably very few, on the other hand we have very many in the VoIP world; All computers on a network.

## Additional security matters

In this chapter, we will discuss relevant security matters not considered previously in our analysis.

**Console cable**
On the Allied Telesyn Residential Gateway [3.2.1.1], a console cable connection exists. This can be used for local configuration of the gateway, and is done by establishing a telnet session via a RS232 interface. This means that you don't necessary need to communicate on the appropriate VLAN to access the gateway configuration tool. This means that if you sniff telnet username and password, a console cable can be used to log in to the gateway, read / use the phone configuration and to get access to the appropriate VLAN for phone calls without paying for it; you can use one of the three LAN ports on the gateway, configure them with the appropriate VLAN and all types of equipment on that port will be transported on the telephony VLAN. To do this makes it possible not to use an expensive switch to get VLAN access, and achieve the same goal.

*Solution: The reason we did not do this in our experiment, was a possibility to disable this console cable connection via software or dip switch on the gateway.*

**Mac filtering on HP Procurve Switches**
If you read the product specification of the HP Procurve 4000 switch [3.2.1.2.1], used as access switches in the network, you might have discovered that this switch support mac filtering on port. This means that a list of allowed mac addresses can filter if a device should get access to the port it is connected at. If you apply this, it is obvious that security will be enhanced. Our computer doing the unauthorized softphone phone call [4.2.4], would not succeed if this had been done, because the computer would not have an registered mac address. However, the example of the console cable would succeed. Anyway, to manage such lists will demand a lot of administration, and is therefore not considered in our case.

*Solution: This will not be done, because it will demand too much administration.*

**Physical access to network equipment**
So far we have only discussed abstract security matters as protocols and data streams. However, it is just as important to have physical segregation of the network equipment, as well as power access and stability. We have not discussed this in detail, because the physical security has been taken good care of in the pilot scenario. Anyway, we feel that this needs to be mentioned, since this is an important issue. All access switches has been installed in locked concrete cabins, with power backup systems. In the core of the network, a stricter policy is applied. All core traffic runs through a central place with servers, switches etc. These rooms have been ensured with both battery and diesel power backup, as well as the entire installation is located in a faraday cage. Traffic from this point has been supported with redundancy and all fibre connections to main locations have possibilities to be routed in alternative routes if i.e. a fibre cable should be cut. How severe and massive destructions that needs to be applied to make the core functionalities unavailable, has not been tested in practice; but situation of war or major sabotage can bring down parts of the network. However, it is extremely unlikely that someone would do all that to ruin the VoIP service. To access this central core traffic rooms, you will have to use both electronic access cards and keys; and very few have got the proper security clearance to enter.

*Solution: Good security has been applied today, and to ensure future acceptable level; the access needs to be limited to a few key persons.*

**Mirror a port on access switch**
It is not possible to listen in on your neighbour's data traffic, when located at the end of the client side. However, on the HP Procurve, it is possible to mirror all traffic to a certain port, listen to this port and get knowledge of traffic close to you. This demands that you can access the switch through a SSH session, and the switch demands that you come from a certain subnet and IP-address. We tested this solution, but did not succeed.

*Solution: Security on this matter is adequate today.*

**Practical password theft**
Some attackers do not bother to even turn on their computer to sniff or get i.e. passwords from other clients on the system. They simply call the user that they would like to exploit, say "Hello, I'm from customer care and see that you have a problem with the connection. If you give me your password, I will take care of it as we speak." This is only a sample of how an attacker could get an answer from a thoughtless client.

*Solution: do not give client passwords visible to them; hardcode this information in boxes given them.*

**Intrusion Detection Systems (IDS)**
IDS systems are automated systems that analyze traffic for matches against a user defined rule set and perform several actions based upon what it sees. These rules can trigger alarms in the system (i.e. by SNMP) when it recognises something suspicious. It can also provide logs of suspicious behaviour. IDS systems exist for both the Linux and Windows platform, both as freeware and expensive software.

We have added a Linux distribution of an IDS system called Snort [7] with this report.

*Solution: set proper rules in an IDS solution to automatically discover suspicious behaviour.*

**Reasons for using VLANs**
In our case, we have looked at VLANs, and found that they don't provide much as security functions. However, we need to mention the reason for why this technology has been applied in the network. Mainly, in our case they are used for two reasons:

1. Control of traffic
2. Ensure Quality of service (QoS)

In addition to this, we can look at VLANs as a security mechanism in this particular network, since we have a situation of triple play (Internet, Multicast Television and Telephony). This because we can experience both in television and Internet traffic so called "multicast junk" being transmitted in the net. A lot of the latest viruses have a tendency to pour out this multicast traffic, jamming up the lines, or at least stressing the switches. By having a particular VLAN for telephony, we will decrease the junk traffic on our VoIP network.

**Proxy to Proxy security**
In our system we have not been talking of problems related to server to server traffic (SIP servers, or so called proxies). Say we have a larger system, and want i.e. our proxy to talk to a proxy in Australia, to give the clients cheap airtime – and not loose billing. We need to set up some form of communication between our proxies. This communication will likely have to be transmitted on the Internet, and encryption for this transmission should be applied.
In SIP and the SIP Express Router, a plug-in for TLS encryption has been added, making it possible for two proxies to talk without sending the data in public. Other solutions can be applied as well. It is important to notice this particular case, since we have not discussed it cause of the nature of our network, but it is essential to larger systems.

*Solution: In our system this has not been an issue, and we have therefore not tested such a configuration. For future expanding, this is an important place to put your attention.*

**Security in commercial systems**
During the analysis, we tried to find which level of security other providers applied to their VoIP solutions, to get a view of which level of security would be appropriate to us. One of the largest providers in Norway is a company called Telio. We run some tests against their servers, and

looked at the product they gave their clients. What we found was that the only security applied to their system was the digest hidden password in SIP register messages. No other encryption had been applied.

**Redundancy for the SIP server**
The SIP server in our scenario had been set up without any redundancy. This is a weakness, i.e. if some hardware fail and we need to change it; the whole system goes down for this maintenance period. You don't need much fantasy to understand that the core server is much to important to allow this.

*Solution: Achieve redundancy for the SIP server and the Cisco 2651xm Router to ensure traffic at all times.*

## Security obligations

In this chapter, we will discuss the demands for security in our particular case. We will talk about the level of acceptance, or which amount of security that is needed, in the picture of possible threats towards the system.

In [4.1] we discussed the security threats. Let us now look at the range of persons that in our world are likely to perform attacks to our system. A short list of some significant types can be:

- Competitor that wants to get all the clients.
- Client that uses the service a lot, and wants his bill to "disappear.
- The Intelligence or similar organization or people that want to get sensitive information by listening to phone calls.
- Persons who just got to know everything, or typical hackers
- Unsatisfied customer.
- Former employee that wants to get even cause of some disagreement.

In the situation of war, or similar we need to add sabotage, terrorist and acts of war to the list. Anyway, that is a different scenario, and can for the especially interested be found in NOU 2000 chapter 6[8], unfortunately this is written in Norwegian. In short terms it claims that VoIP among other can be considered as weaker in a situation of war than the old analogue phone system.

The next to decide is which weaknesses should be accepted, and which should not. In the last chapter, we find that the PSDN does not offer end to end encryption. Should VoIP have to support this when the old system does not?

To support us when needing to find our obligations, it is obvious to think of what the law says.

**Obligations**
If we look at the demands set by Norwegian law, we will probably find that VoIP have less obligations than other telecom solutions. This is among other due to the possibility to define a VoIP network as a private telephone network, and therefore escape some of the demands within the Norwegian law for telephony; Teleloven[9]. In a document written by the writer of this report, you can look further into this [10]. Both these documents are only available in Norwegian.

**Rights**
In 2003 the Department of Police and Justice published a document called "NOU 2003: 27 Lovtiltak mot datakriminalitet [11]" This is a report that discusses the European council convention on fighting crime bound to information and communication technology. This document has proposed new laws to protect both users and suppliers of this technology. This document handles law proposals against; illegal interception, illegal access, data interference, system interference, misuse of devices, computer related forgery, computer related fraud and more. These are also the threats against the SIP telephony service of Èlla Kommunikasjon AS. In the

future, they will at least have better adapted laws to protect against crimes that today are hard to fight.

Because the laws ("teleloven" in particular) don't demand exceptionally much from a telephony supplier; if defined as in our case, the security measurement will be more or less totally up to the supplier to decide. However, most of the threats and possible attacks against the VoIP of Èlla Kommunikasjon AS, will also lead to unsatisfied clients. Therefore its in Èlla Kommunikasjon AS own interest to secure the system in a good way.

# Conclusion

In this thesis we have evaluated the security of Èlla Kommunikasjon AS's SIP protocol based VoIP service, by simulating the role of a malicious attacker. The result of this assessment was that we found the service having several serious exploitable weaknesses.

The proposal for securing the system ends up in the following checkpoints:

- Use encrypted protocols for management of the Allied Telesyn Gateway. Switch the telnet administration with a SSH based one when available.
- Use fibre interfaces all the way to the residential gateway to avoid or decrease packet sniffing. There already exists such devices, and they are more affordable than the solution chosen today.
- Change password politics for the password in the SIP authentication message.
- Install a firewall on the SIP proxy / Server, using an IP-address filter to give administrators SSH access to the server.
- Divide the network into appropriate sized subnets.
- Close all services not needed by an administrator. That is also the web server and web applications within the public_html directory. If not possible, make sure these services are protected through a filter in the firewall.
- Use a third party billing and authentication system (by radius) to avoid DoS attacks, and unauthorized access to change billing data on the SIP server.
- For some clients, confidentiality can be important enough to demand end-to-end encryption for conversations. Establish this as a service / additional product, to meet customer demands. This can be done by using devices supporting such an encryption.
- Apply full redundancy for the SIP Server and the Cisco 2651xm router to ensure traffic at all times.

Additionally we have found that using VLAN's as a security mechanism is not a good solution. It works well for separating traffic types, but not to prevent sniffing of data packets.

The result of this thesis may help other individuals using VoIP services based on similar network topology to increase their level of security. Hopefully it may also increase the use of secure ways to deploy SIP based IP-telephony services.

In my opinion, the work found interesting results already early on in the analysis. It was especially useful to look at the security with "the eyes of an attacker", as this can be compared to being "looking with the eyes of the enemy". The result of this report have been satisfying, already making way for a new and secure deployment of the VoIP service in Èlla Kommunikasjon AS's network.

Further work that remains to be done is the complete testing of the new proposed configuration. We will first see how useful the work being done is when a full scale service has been launched.

However, we keep in mind the wisdom of Aristotle (384-322 B.C.) saying:

"It is likely that something unlikely will happen"

# List of Figures and Tables

FIGURE 1 SIP BASIC CALL FLOW ..... 12
FIGURE 2 SIP LAYERS ..... 14
FIGURE 3 REAL-TIME TRANSPORT PROTOCOL (RTP) ..... 22
FIGURE 4 SIMPLIFIED OSI-MODEL ..... 25
FIGURE 5 IP IN TCP AND UDP ..... 25
FIGURE 6 TCP DATAGRAM ..... 26
FIGURE 7 IP PACKAGE ..... 26
FIGURE 8 CLASS-A NET ..... 27
FIGURE 9 CLASS-B NET ..... 27
FIGURE 10 CLASS-C NET ..... 28
FIGURE 11 SUBNET AND SUBNET ID'S ..... 29
FIGURE 12 SUBNET AND IP-ADDRESSES ..... 29
FIGURE 13 NET MASK ..... 30
FIGURE 14 SAMPLE NETWORK FOR ROUTING ISSUES ..... 30
FIGURE 15 ROUTING TABLE FOR ORDINARY HOST WITH ONE NETWORK INTERFACE ..... 31
FIGURE 16 DNS OVERVIEW ..... 34
FIGURE 17 DNS LOOKUP ..... 35
FIGURE 18 DNS ZONES ..... 36
FIGURE 19 TRADITIONAL FULLY ROUTED NETWORK ..... 37
FIGURE 20 STANDARD SWITCHED NETWORK ..... 37
FIGURE 21 PORT-BASED VLAN ..... 39
FIGURE 22 PPPOE NETWORK MODEL ..... 40
FIGURE 23 VISUALIZATION OF PROTOCOL STACK ..... 41
FIGURE 24 TLS HANDSHAKE PROTOCOL TIMELINE. TRANSACTIONS MARKED WITH * ARE DEPENDENT ON SITUATION. ..... 42
FIGURE 25 IPSEC AUTHENTICATION HEADER (AH) ..... 44
FIGURE 26 AH IN TRANSPORT MODE ..... 45
FIGURE 27 AH IN TUNNEL MODE ..... 45
FIGURE 28 ESP IN TRANSPORT MODE ..... 46
FIGURE 29 ESP IN TUNNEL MODE ..... 46
FIGURE 30 SSH SECURE ENCRYPTED TUNNEL ..... 48
FIGURE 31 SSH KEY INTRODUCTION ..... 49
FIGURE 32 SIMPLIFIED NETWORK MODEL - ÈLLA KOMMUNIKASJON AS ..... 56
FIGURE 33 VOIP TRAFFIC FROM CLIENT TO SERVER ..... 58
FIGURE 34 VOIP TRAFFIC FROM SERVER TO CLIENT ..... 59
FIGURE 35 CLIENT TO CLIENT TRAFFIC - WHEN WITHIN THE SAME NETWORK ..... 59
FIGURE 36 VOIP TRAFFIC AGAINST PSTN ..... 60
FIGURE 37 ALLIED TELESYN RG213TX RESIDENTIAL VOIP GATEWAY ..... 61
FIGURE 38 ALLIED TELESYN MC103X FIBRE CONVERTER ..... 62
FIGURE 39 HP PROCURVE 4000 SERIES SWITCHES - LAYER 2 ..... 63
FIGURE 40 HP PROCURVE 5300 SERIES SWITCHES - LAYER 2, 3 AND 4 ..... 64
FIGURE 41 CISCO 2651XM ROUTER ..... 65
FIGURE 42 SIMPLIFIED VIEW OF THE VOCAL SYSTEM SOURCE: CISCO SYSTEMS INC ..... 66
FIGURE 43 A PDA CAN PROCESS 150 CALLS PER SECOND WITH SIP EXPRESS ROUTER ..... 68
FIGURE 44 CISCO IP PHONE 7960 SERIES ..... 73
FIGURE 45 LEADTEK BROADBAND VIDEOPHONE ..... 74
FIGURE 46 ESTARA SOFTPHONE ..... 75
FIGURE 47 SECURITY THREATS. SOURCE: ISBN 0201485346 [] ..... 76
FIGURE 48 DECODED SIP MESSAGE FROM TRAFFIC ANALYZER ..... 79
FIGURE 49 SNIFFING PACKAGES FROM THE ALLIED TELESYN RESIDENTIAL GATEWAY ..... 79
FIGURE 50 SIP REGISTRATION PROCESS SOURCE: CISCO SYSTEMS INC ..... 80
FIGURE 51 SIP REGISTER WITH DIGEST ..... 80
FIGURE 52 SNIFFING TELNET SESSIONS ..... 81
FIGURE 53 ORGANIZED ASSAULT - MAIN PICTURE ..... 82
FIGURE 54 SNIFFING DHCP REQUESTS AND REPLIES ..... 84
FIGURE 55 VLAN FAKER – CONFIGURATION ..... 85
FIGURE 56 ASSAULT: MANUALLY CONFIGURED IP ADDRESS ..... 86
FIGURE 57 ASSAULT: ALLIED SHOW SIP ..... 86

FIGURE 58 X-LITE SOFTPHONE ..... 87
FIGURE 59 RETINA NETWORK SECURITY SCANNER - IP-TELEPHONY SERVER SECURITY SCAN ..... 88
FIGURE 60 TELEPHONY SERVER RISK LEVEL GRAPH ..... 90
FIGURE 61 IP-TELEPHONY SERVER WEB ROOT ..... 92
FIGURE 62 BRUTE HTTP (BASIC) WEB LOGIN ..... 94
FIGURE 63 SQLMYADMIN INTERFACE - HACKED ..... 95
FIGURE 64 DATA LOAD OF A SIP CONVERSATION USES RTP. SOURCE: CISCO SYSTEMS INC. ..... 98
FIGURE 65 ETHEREAL RTP ANALYZER FUNCTION ..... 99
FIGURE 66 SAVING RTP STREAMS AS AUDIO FILES ..... 99
FIGURE 67 HOW TO SNIFF TRAFFIC ON A FIBRE ENVIRONMENT ..... 102
FIGURE 68 LISTENING TO PHONE CALLS IN ÈLLA KOMMUNIKASJON SWITCHED NETWORK ..... 106
FIGURE 69 LISTENING TO PHONE CALLS WHEN ALL TRAFFIC GOES THROUGH CORE ..... 107
FIGURE 70 SRTP ENCRYPTED ZIP 4X4 HARD PHONE FROM ZULTYS ..... 107
FIGURE 71 END TO END CLIENT ENCRYPTION AES ..... 108

TABLE 1 SIP ADDRESSING ..... 11
TABLE 2 SIP BASIC CALL FLOW ..... 13
TABLE 3 GENERIC MESSAGES ..... 17
TABLE 4 GENERIC MESSAGES - HEADERS ..... 17
TABLE 5 SIP-URL ..... 18
TABLE 6 ETHERNET FRAME WITH IEEE802.3 PROTOCOL ..... 26
TABLE 7 CLIENT ROUTING TABLE ..... 32
TABLE 8 ROUTER TABLE, SIMPLE SAMPLE ..... 32
TABLE 9 KEY IPSEC DOCUMENTS ..... 42
TABLE 10 ALLIED TELESYN RG213TX VOIP GATEWAY PROTOCOL / STANDARDS ..... 62
TABLE 11 HP PROCURVE 4000 SERIES FEATURES ..... 63
TABLE 12 HP PROCURVE 5300 SERIES FEATURES ..... 64
TABLE 13 VOCAL SYSTEM COMPONENTS DESCRIPTION ..... 67
TABLE 14 RFC 2543 DEFINITION COMPARED TO VOCAL FUNCTIONALITY ..... 67
TABLE 15 SIP CONFIG OF THE ALLIED TELESYN RG213TX ..... 69
TABLE 16 CISCO 7960 IP PHONE CONFIG SPECIFIC FOR SINGLE PHONE ..... 70
TABLE 17 CISCO 7960 IP PHONE CONFIG COMMON ..... 73
TABLE 18 CISCO 7960 IP PHONE CONFIG DIALPLAN ..... 73
TABLE 19 RPC SERVICES ..... 93
TABLE 20 RPC EXPLOIT SERVICE LIST ..... 96

## Glossary

**DoS Attack:** A Denial of Service (DoS) attack is a remote attack against a servers TCP/IP stack or services. DoS attacks can saturate a server's bandwidth, saturate all available connections for a particular service, or even crash a server.

**Exploit:** A script or program that takes advantage of vulnerabilities in services or programs to allow an attacker to gain unauthorized or elevated system access.

**Host:** A node on a network. Usually refers to a computer or device on a network which both initiates and accepts network connections.

**IP Address:** The 32-bit address defined by the Internet Protocol in STD 5, RFC 791. It is usually represented in dotted decimal notation. Any device connected to the Internet that used TCP/IP is assigned an IP Address. An IP Address can be likened to a home address in that no two are alike.

**Netbios:** Network Basic Input Output System. The standard interface to networks on IBM PC and compatible networks.

**Ping:** A program used to test reach ability of destination nodes by sending them an ICMP echo request and waiting for a reply.

**Port:** A port in the network sense is the pathway that a computer uses to transmit and receive data. As an example, Web Servers typically listen for requests on port 80.

**Registry:** The internal system configuration that a user can customize to alter his computing environment on the Microsoft Windows Platform. The registry is organized in a hierarchical structure of sub trees and their respective keys, sub keys, and values that apply to those keys and sub keys

**Service:** A service is a program running on a remote machine that in one way or another provides a service to users. For example, when you visit a website the remote server displays a web page via its web server service.

**Share:** A folder, set of files, or even a hard drive partition set up on a machine to allow access to other users. Shares are frequently set up with incorrect file permissions which could allow an attacker to gain access to this data.

**Sniffers:** frequently attackers will place a sniffers program on a compromised machine. The sole purpose of a sniffers is to collect data being transmitted on the network in clear-text including usernames and passwords.

**Subnet:** A portion of a network, which may be a physically independent network segment, which shares a network address with other portions of the network and is distinguished by a subnet number.

**Vulnerability:** A weakness or a flaw in a program or service that can allow an attacker to gain unauthorized or elevated system access.

# References

Below you can find references made in this paper. It contains both software, hardware and manual references; as well as RFC's and other useful documentation.

All reference types are organized into chapters. Following chapters has been used:
1. SIP related RFC's
2. Other RFC's
3. Law relations
4. Software
5. Hardware Manuals
6. Other

## SIP related RFC's


**RFC 3524 Mapping of Media Streams to Resource Reservation Flows**
*Summary:* *This document defines an extension to the Session Description Protocol (SDP) grouping framework. It allows requesting a group of media streams to be mapped into a single resource reservation flow. The SDP syntax needed is defined, as well as a new "semantics" attribute called Single Reservation Flow (SRF).*
*Author:* *The Internet Society*
*Date:* *April 2003.*
*Source:* *http://www.rfc-editor.org/rfc/rfc3524.txt*

**RFC 3515 The Session Initiation Protocol (SIP) Refer Method**
*Summary:* *Defines the REFER method. This Session Initiation Protocol (SIP) extension requests that the recipient REFER to a resource provided in the request. It provides a mechanism allowing the party sending the REFER to be notified of the outcome of the referenced request. This can be used to enable many applications, including call transfer. In addition to the REFER method, this document defines the the refer event package and the Refer-To request header.*
*Author:* *The Internet Society*
*Date:* *April 2003*
*Source:* *http://www.rfc-editor.org/rfc/rfc3515.txt*

**RFC 3487 Requirements for Resource Priority Mechanisms for the Session Initiation Protocol (SIP)**
*Summary:* *Summarizes requirements for prioritizing access to circuit-switched network, end system and proxy resources for emergency preparedness communications using the Session Initiation Protocol (SIP).*
*Author:* *The Internet Society*
*Date:* *February 2003*
*Source:* *http://www.rfc-editor.org/rfc/rfc3487.txt*

**RFC 3486 Compressing the Session Initiation Protocol (SIP)**
*Summary:* *Describes a mechanism to signal that compression is desired for one or more Session Initiation Protocol (SIP) messages. It also states when it is appropriate to send compressed SIP messages to a SIP entity.*
*Author:* *The Internet Society*
*Date:* *February 2003*

*Source:* http://www.rfc-editor.org/rfc/rfc3486.txt

**RFC 3485 The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp)**

*Summary:* *The Session Initiation Protocol (SIP) is a text-based protocol for initiating and managing communication sessions. The protocol can be compressed by using Signaling Compression (SigComp). Similarly, the Session Description Protocol (SDP) is a text-based protocol intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. This memo defines the SIP/SDP-specific static dictionary that SigComp may use in order to achieve higher higher efficiency. The dictionary is compression algorithm independent.*
*Author:* *The Internet Society*
*Date:* *February 2003*
*Source:* http://www.rfc-editor.org/rfc/rfc3485.txt

**RFC 3428 Session Initiation Protocol (SIP) Extension for Instant Messaging**

*Summary:* *Instant Messaging (IM) refers to the transfer of messages between users in near real-time. These messages are usually, but not required to be, short. IMs are often used in a conversational mode, that is, the transfer of messages back and forth is fast enough for participants to maintain an interactive conversation. This document proposes the MESSAGE method, an extension to the Session Initiation Protocol (SIP) that allows the transfer of Instant Messages. Since the MESSAGE request is an extension to SIP, it inherits all the request routing and security features of that protocol. MESSAGE requests carry the content in the form of MIME body parts. MESSAGE requests do not themselves initiate a SIP dialog; under normal usage each Instant Message stands alone, much like pager messages. MESSAGE requests may be sent in the context of a dialog initiated by some other SIP request.*
*Author:* *The Internet Society*
*Date:* *December 2002*
*Source:* http://www.rfc-editor.org/rfc/rfc3428.txt

**RFC 3420 Internet Media Type message/sipfrag**

*Summary:* *This document registers the message/sipfrag Multipurpose Internet Mail Extensions (MIME) media type. This type is similar to message/sip, but allows certain subsets of well formed Session Initiation Protocol (SIP) messages to be represented instead of requiring a complete SIP message. In addition to end-to-end security uses, message/sipfrag is used with the REFER method to convey information about the status of a referenced request.*
*Author:* *The Internet Society*
*Date:* *November 2002*
*Source:* http://www.rfc-editor.org/rfc/rfc3420.txt

**RFC 3388 Grouping of Media Lines in Session Description Protocol (SDP)**

*Summary:* *Extensions to SDP that allow grouping of media streams for lip synchronization and to represent the same content on different network addresses*
*Author:* *The Internet Society*
*Date:* *December 2002*
*Source:* http://www.rfc-editor.org/rfc/rfc3388.txt

**RFC 3361 Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers**

*Summary:* *Defines a DHCP option for locating the outbound SIP proxy server*
*Author:* *The Internet Society*
*Date:* *August 2002*

*Source:* http://www.rfc-editor.org/rfc/rfc3361.txt

**RFC 3319 Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers**
*Summary:* *Defines a DHCPv6 options for locating the outbound SIP proxy server*
*Author:* *The Internet Society*
*Date:* *July 2003*
*Source:* http://www.rfc-editor.org/rfc/rfc3319.txt

**RFC 3327 Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts**
*Summary:* *Defines the Path header field that registers a list of proxies between the UA and the registrar*
*Author:* *The Internet Society*
*Date:* *December 2002*
*Source:* http://www.rfc-editor.org/rfc/rfc3327.txt

**RFC 3326 The Reason Header Field for the Session Initiation Protocol (SIP)**
*Summary:* *For creating services, it is often useful to know why a Session Initiation Protocol (SIP) request was issued. This document defines a header field, Reason, that provides this information. The Reason header field is also intended to be used to encapsulate a final status code in a provisional response. This functionality is needed to resolve the "Heterogeneous Error Response Forking Problem", or HERFP.*
*Author:* *The Internet Society*
*Date:* *December 2002*
*Source:* http://www.rfc-editor.org/rfc/rfc3326.txt

**RFC 3325 Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks**
*Summary:* *Defines P-Asserted-Identity and P-Preferred-Identity header fields, allowing SIP proxies to add user identity information and callers to request privacy*
*Author:* *The Internet Society*
*Date:* *November 2002*
*Source:* http://www.rfc-editor.org/rfc/rfc3325.txt

**RFC 3324 Short Term Requirements for Network Asserted Identity**
*Summary:* *Defines requirements for caller identities established by network entities*
*Author:* *The Internet Society*
*Date:* *November 2002*
*Source:* http://www.rfc-editor.org/rfc/rfc3324.txt

**RFC 3323 A Privacy Mechanism for the Session Initiation Protocol (SIP)**
*Summary:* *Describes SIP caller privacy issues and defines the Privacy header field*
*Author:* *The Internet Society*
*Date:* *November 2002*
*Source:* http://www.rfc-editor.org/rfc/rfc3323.txt

**RFC 3329 Security Mechanism Agreement for the Session Initiation Protocol (SIP)**
*Summary:* *This document defines new functionality for negotiating the security mechanisms used between a Session Initiation Protocol (SIP) user agent and its next-hop SIP entity. This new functionality supplements the existing methods of choosing security mechanisms between SIP entities.*
*Author:* *The Internet Society*
*Date:* *January 2003*
*Source:* http://www.rfc-editor.org/rfc/rfc3329.txt

**RFC 3313 Private Session Initiation Protocol (SIP) Extensions for Media Authorization**
*Summary:* *Describes the need for Quality of Service (QoS) and media authorization and defines a Session Initiation Protocol (SIP) extension that can be used to integrate QoS admission control with call signaling and help guard against denial of service attacks. The use of this extension is only applicable in administrative domains, or among federations of administrative domains with previously agreed-upon policies, where both the SIP proxy authorizing the QoS, and the policy control of the underlying network providing the QoS, belong to that administrative domain or federation of domains.*
*Author:* *The Internet Society*
*Date:* *January 2003*
*Source:* *http://www.rfc-editor.org/rfc/rfc3313.txt*

**RFC 3312 Integration of Resource Management and SIP**
*Summary:* *Framework for preconditions*
*Author:* *The Internet Society*
*Date:* *October 2002*
*Source:* *http://www.rfc-editor.org/rfc/rfc3312.txt*

**RFC 3311 The Session Initiation Protocol (SIP) UPDATE Method**
*Summary:* *This specification defines the new UPDATE method for the Session Initiation Protocol (SIP). UPDATE allows a client to update parameters of a session (such as the set of media streams and their codecs) but has no impact on the state of a dialog. In that sense, it is like a re-INVITE, but unlike re-INVITE, it can be sent before the initial INVITE has been completed. This makes it very useful for updating session parameters within early dialogs.*
*Author:* *The Internet Society*
*Date:* *September 2002*
*Source:* *http://www.rfc-editor.org/rfc/rfc3311.txt*

**RFC 3261 SIP: Session Initiation Protocol**
*Summary:* *Core protocol specification; obsoletes RFC 2543*
*Author:* *The Internet Society*
*Date:* *June 2002*
*Source:* *http://www.rfc-editor.org/rfc/rfc3261.txt*

**RFC 3262 Reliability of Provisional Responses in the Session Initiation Protocol (SIP)**
*Summary:* *Making 1xx responses reliable; introduces PRACK method*
*Author:* *The Internet Society*
*Date:* *June 2002*
*Source:* *http://www.rfc-editor.org/rfc/rfc3262.txt*

**RFC 3263 Session Initiation Protocol (SIP): Locating SIP Servers**
*Summary:* *Describes DNS mechanisms (NAPTR, SRV) for locating SIP servers*
*Author:* *The Internet Society*
*Date:* *June 2002*
*Source:* *http://www.rfc-editor.org/rfc/rfc3263.txt*

**RFC 3264 An Offer/Answer Model with the Session Description Protocol (SDP)**
*Summary:* *How SDP is used within SIP to negotiate sessions*
*Author:* *The Internet Society*
*Date:* *June 2002*
*Source:* *http://www.rfc-editor.org/rfc/rfc3264.txt*

**RFC 3265 Session Initiation Protocol (SIP)-Specific Event Notification**
*Summary:* *SIP event model; defines SUBSCRIBE and NOTIFY*
*Author:* *The Internet Society*
*Date:* *June 2002*
*Source:* *http://www.rfc-editor.org/rfc/rfc3265.txt*

**RFC 3087 Control of Service Context using SIP Request-URI**
*Summary:* *Defines how the SIP URI can be used to invoke services such as voicemail*
*Author:* *The Internet Society*
*Date:* *April 2001*
*Source:* *http://www.rfc-editor.org/rfc/rfc3087.txt*

**RFC 3050 Common Gateway Interface for SIP**
*Summary:* *sip-cgi, as scripting interface*
*Author:* *The Internet Society*
*Date:* *January 2001*
*Source:* *http://www.rfc-editor.org/rfc/rfc3050.txt*

**RFC 2976 The SIP INFO Method**
*Summary:* *Defines INFO method for carrying SIP-related information*
*Author:* *The Internet Society*
*Date:* *October 2000*
*Source:* *http://www.rfc-editor.org/rfc/rfc2976.txt*

**RFC 2848 The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services**
*Summary:* *Defines how SIP events can be used to invoke PSTN services such as Internet call waiting*
*Author:* *The Internet Society*
*Date:* *June 2000*
*Source:* *http://www.rfc-editor.org/rfc/rfc2848.txt*

## Other RFC's

**RFC 2396 - Uniform Resource Identifiers (URI): Generic Syntax**
*Author:* *The Internet Society*
*Date:* *January 1999.*
*Source:* *http://rfc.x42.com/*

**RFC 1340 - Assigned Numbers**
*Author:* *The Internet Assigned Numbers Authority (IANA)*
*Date:* *July 1992.*
*Source:* *http://rfc.x42.com/*

**RFC 2131 - Dynamic Host Configuration Protocol**
*Author:* *R. Droms*
*Date:* *March 1997.*
*Source:* *http://rfc.x42.com/*

**RFC 2246 The TLS Protocol Version 1.0**
*Author:* IETF Working Group on Transport Layer Security / The Internet Society
*Date:* *January 1999.*
*Source:* *ftp://ftp.ietf.org/rfc/rfc2246.txt*

**RFC 2401 Security Architecture for the Internet Protocol (IPSec)**
*Author:* The Internet Society
*Date:* *November 1998*
*Source:* *http://www.ietf.org/rfc/rfc2401.txt*

## Law relations

**NOU 2000: 24 Et Sårbart Samfunn**
*Author:* *Department of Police and Justice*
*Date:* *July 2000.*
*Source:* *http://odin.dep.no ISBN: 82-583-0537-9*

**LOV 1995-06-23 nr 39, Lov om telekommunikasjon (teleloven)**
*Author:* *Department of Police and Justice*
*Date:* *June 1995.*
*Source:* *http://odin.dep.no*

**LOV IP-telefoni mot Privatmarkedet, Aktuell lovgivning**
*Author:* *Dag Ove Valsgaard, Èlla Kommunikasjon AS*
*Date:* *April 2003.*
*Local Source:* *CD-ROM attached to this document:*
*"cd_root:_ref\"*

**2003: 27: Lovtiltak mot datakriminalitet**
*Author:* *Department of Police and Justice*
*Date:* *November 2003.*
*Source:* *http://odin.dep.no ISBN: 82-583-0736-3*

## Software

**Allied Telesyn RG213TX Residential Gateway SIP 6.1 Software Reference**
*Author:* *Allied Telesyn*
*Date:* *February 2003*
*Source:* *http://www.alliedtelesyn.com*
*Local Source:* *CD-ROM attached to this document:*
*"cd_root:_ref\AT-RG213_SIP_6-1-0_sw_reference.pdf"*

**Vocal SIP Proxy Server v.1.5.0**
*Author:* *Cisco Systems Inc*
*Date:* *unknown*
*Source:* *http://vovida.org/*
*Local Source:* *CD-ROM attached to this document:*
*"cd_root:_sw\vocal-1.5.0.tar.gz"*

***SIP Express Router (SER)***
*Author:* *iptel.org*
*Date:* *unknown*
*Source:* *http://www.iptel.org*
*Local Source:* *CD-ROM attached to this document:*
*"cd:_sw\ser-0.8.12_src.tar.gz"*

**Etheral Network Traffic Analyzer v10.0**
*Author:* *Ethereal*

*Date:* *unknown*
*Source:* *http://www.ethereal.com*
*Local Source:* *CD-ROM attached to this document:*
*"cd_root:_sw\ethereal-setup-0.10.0.exe" AND*
*"cd_root:_sw*WinPcap_3_0.exe"

**X-Lite Softphone**
*Author:* *Xten Networks Inc*
*Date:* *unknown*
*Source:* *http://www.xten.com*
*Local Source:* *CD-ROM attached to this document:*
*"cd_root:_sw\X-Lite_Install.exe"*

**Retina Network Security Scanner v4.9.115**
*Author:* *eEye*
*Date:* *unknown*
*Source:* *http://www.eeye.com/html*
*Local Source:* *CD-ROM attached to this document:*
*"cd_root:_sw\retina.exe"*

**SuSE 9.0 Linux Platform**
*Author:* *SuSE Inc*
*Date:* *2003*
*Source:* *http://www.suse.com/us/*

**phpMyAdmin 2.5.6**
*Author:* *Open Source*
*Date:* *unknown*
*Source:* *http://www.phpmyadmin.net/home_page/*
*Local Source:* *CD-ROM attached to this document:*
*"cd_root:_sw\phpMyAdmin-2.5.6.tar.gz"*

**Snort v.2.1.3RC1-1**
*Author:* *Open Source*
*Date:* *unknown*
*Source:* *http://www.snort.net*
*Local Source:* *CD-ROM attached to this document:*
*"cd_root:_sw\ snort-mysql-2.1.3RC1-1.i386.rpm"*

## Hardware Manuals

**Allied Telesyn RG213TX Residential Gateway**
*Author:* *Allied Telesyn*
*Date:* *unknown*
*Source:* *http://alliedtelesyn.com/*

**Allied Telesyn MC103X Fibre Converter**
*Author:* *Allied Telesyn*
*Date:* *unknown*
*Source:* *http://alliedtelesyn.com/*

**HP Procurve 4000 Series Switches (Layer 2)**
*Author:* *HP Procurve Networking*

*Date:* *unknown*
*Source:* *http://www.hp.com/rnd/index.htm*

**HP Procurve 5300 Series Switches (Layer 2, 3 & 4)**
*Author:* *HP Procurve Networking*
*Date:* *unknown*
*Source:* *http://www.hp.com/rnd/index.htm*

**Cisco 2651XM Router**
*Author:* *Cisco Systems Inc*
*Date:* *unknown*
*Source:* *http://www.cisco.com/en/US/products/hw/routers/ps259/ps4834/index.html*

**Cisco IP Phone 7960 Series**
*Author:* *Cisco Systems Inc*
*Date:* *unknown*
*Source:* *http://cisco.com/en/US/products/hw/phones/ps379/index.html*

**LeadTek broadband Videophone**
*Author:* *LeadTek Research Inc*
*Date:* *unknown*
*Source:* *http://leadtek.com/videophone/bvp_8770_1.html*

**ZIP 4x4 HardPhone**
*Author:* *Zultys Technologies*
*Date:* *unknown*
*Source:* *http://www.zultys.com/ZIP4x4.htm*

## Other

***'SNMP, SNMPv2, SNMPv3, and RMON 1 and 2'* - 3rd ed**
*Author:* *W. Stallings*
*Date:* *December 1998.*
*Source:* *Addison-Wesley Pub Co ISBN: 0201485346*

## Software and Documentation