# Internet Information Shop

Jos Visser
Open Solution Providers

13 juli 2004

Formatted (and presumably printed) on 13 juli 2004.

# Inhoudsopgave

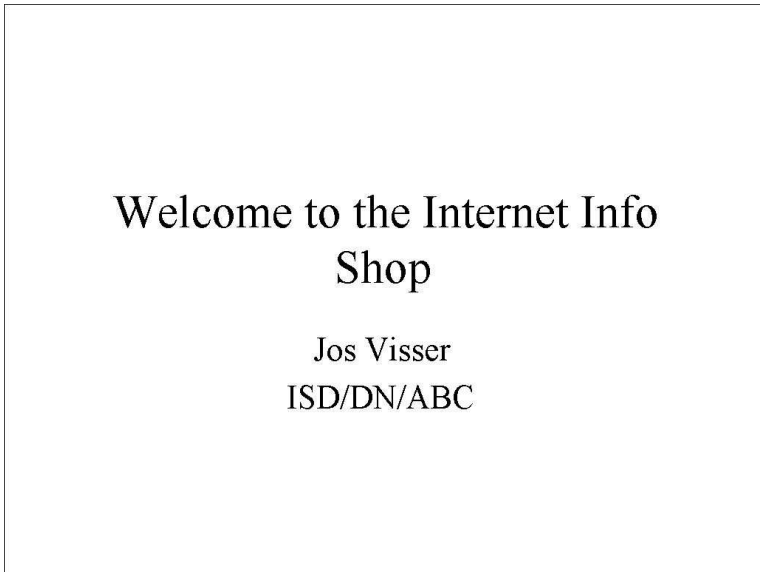
# Deel I

# Inleiding Internet

# Hoofdstuk 1

# Introduction

## 1.1 Welcome

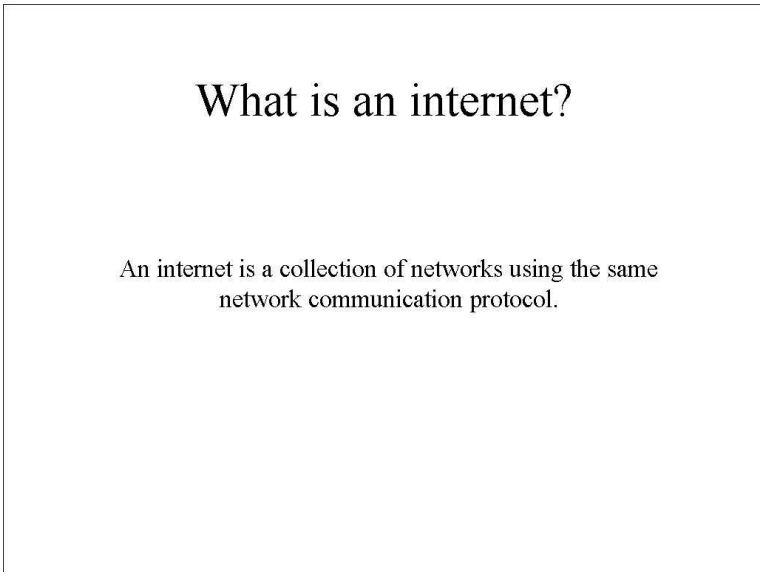



**Notities**

This shop is designed to present an introduction to Internet. It is intended for those who want to know what Internet is and how to use it.

This workbook contains all sheets presented during the shop, but can also be used as a reference after you complete the shop.

## 1.2 What is an internet

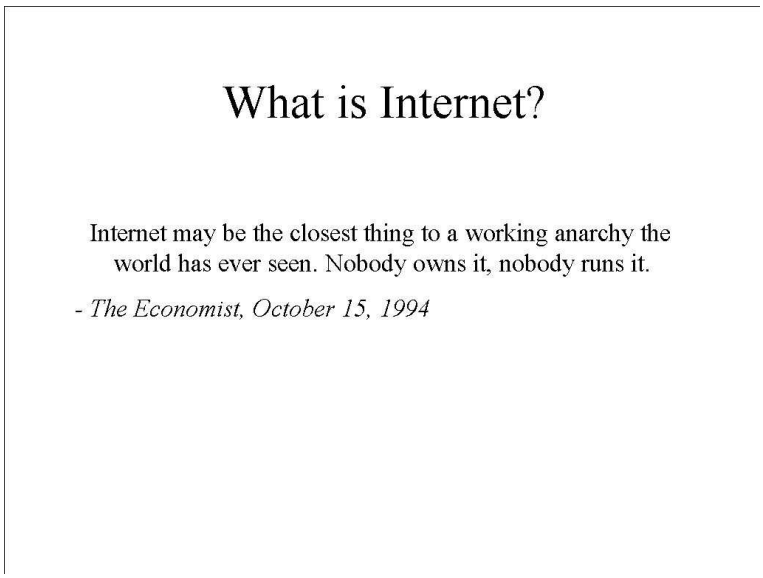



**Notities**

The word internet has more than one meaning. For those using Internet, there is just one Internet, but in fact an internet is just a collection of networks, using the same network communication protocol, linked together.

The individual networks might even not be running the same protocol, as long as the protocol used for communications between the individual networks, the internet protocol, is the same. By definition, any number of networks linked to one another would be an internet. On an internet, nodes in each of the inter-linked networks can communicate with nodes in each other network. Usually, security measures will be taken to avoid this, of course. The inter network connectivity is set up using gateways between the local network, and the other networks. Each local network is connected to the Internet by means of such a gateway.

## 1.3 What is Internet



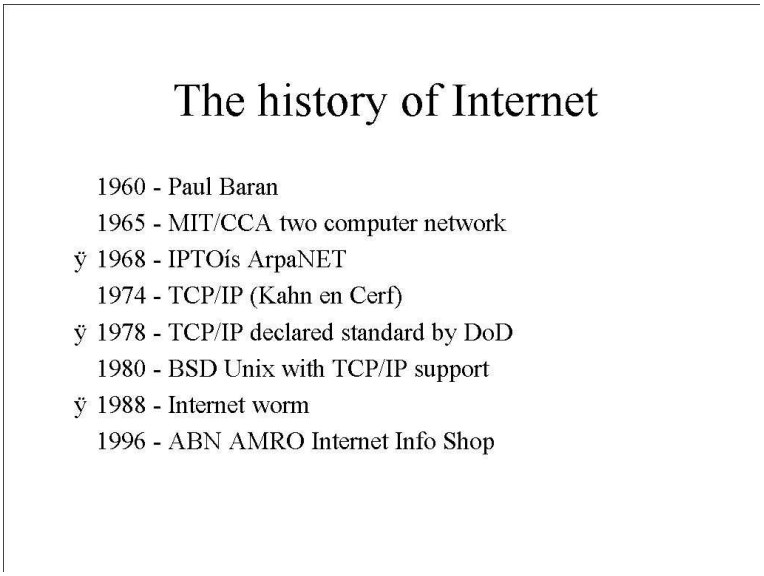

**Notities**

The Internet is not just any internet. Note that Internet is spelled with a capital 'I', whereas internet is not. Internet is an internet, existing of many computers and users, just like any other internet. The common protocol used in Internet is TCP/IP. This shop is not about TCP/IP however!

One could see Internet as a huge computer with a very high potential of computing power (imagine the power of 20.000.000 processors) not managed by anyone (imagine your own site not administered). As a matter of fact, there have been documented cases of Internet users combining their computing power to solve numerically difficult problems (like cracking encryption keys).

Despite the fact that some think of Internet as complete anarchy, in fact it is the largest working computer network in the world. This anarchy is also the success of Internet. Most people using Internet also contribute in some way (so called prosumers). Most of the software used on Internet is public domain and can be obtained from many Internet sites all over the world. Using Internet is still relatively cheap, but don't forget that commercialisation of Internet is on it's way.

## 1.4 The history of Internet




**Notities**

Some of the important mile stones in the history of Internet:

Back in the 60's, the first experiments with connecting computers were conducted. Until then, computers had been standalone; no data could be shared on-line. As computing power was expensive, it had to be used as economically as possible.

In those days, there were not many standards in the computing environment. Every machine had it's own - probably locally adapted - operating system, so it was a tough job to connect systems of an entirely different architecture to one another.

In those early 60's Paul Baran at RAND presented a series of reports on the topic of computer networks. Around 1965 there was a project at Computer Corporation of America, founded by DARPA, linking two computers to a 'network', a third computer was added later.

Another important experiment was carried out in 1968, the first experiment for ARPANET was set up. The original idea was for ARPANET to carry command and control data in case of a nuclear war, but that was forgotten soon enough.

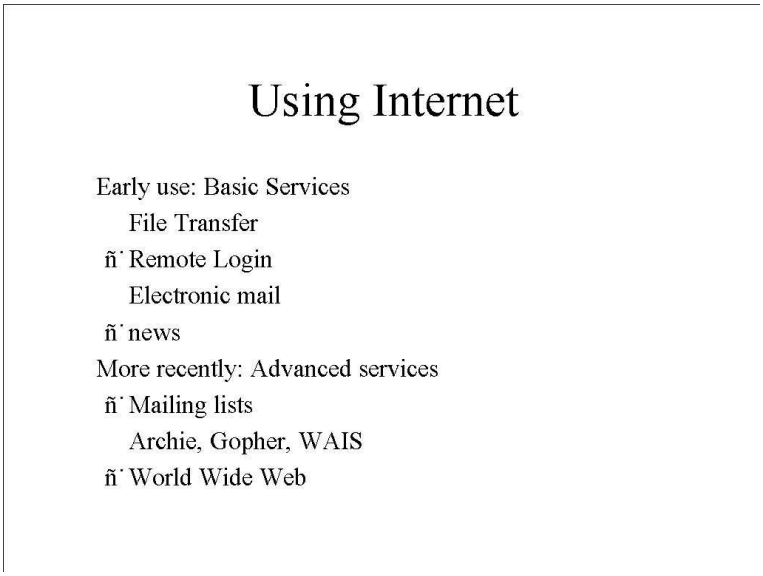
In 1974, TCP/IP was defined as a standard to overcome the many manufacturers networking protocols. It was meant as a general protocol, to be implemented by several manufacturers. In 1978 it became the preferred way to send data from one computer to another within the USA government.

In the early 80's BSD UNIX came with built-in TCP/IP support and set a new standard in UNIX communications.

There was a lot of panic on Internet at November 1st 1988, when the Internet worm

was born (and died). That day, a self- reproducing program was sent to Internet. Based on a set of well- known bugs, Robert T. Morris had written a program that spread via Internet hosts to each Internet host known locally. Unfortunately, each network on Internet has at least one node knowing at least one node not in the local network, so it spread rapidly. Needless to say that these bugs are solved by now, but as of that day, security became a real issue on the net.

## 1.5 Using Internet




**Notities**

At first, Internet was used to transmit files from one computer to another and to login on remote systems. Later electronic mail (or email) was added, thereby giving users the ability to send messages to other users of the network. The chapter on basic Internet application contains a more detailed introduction to these applications.

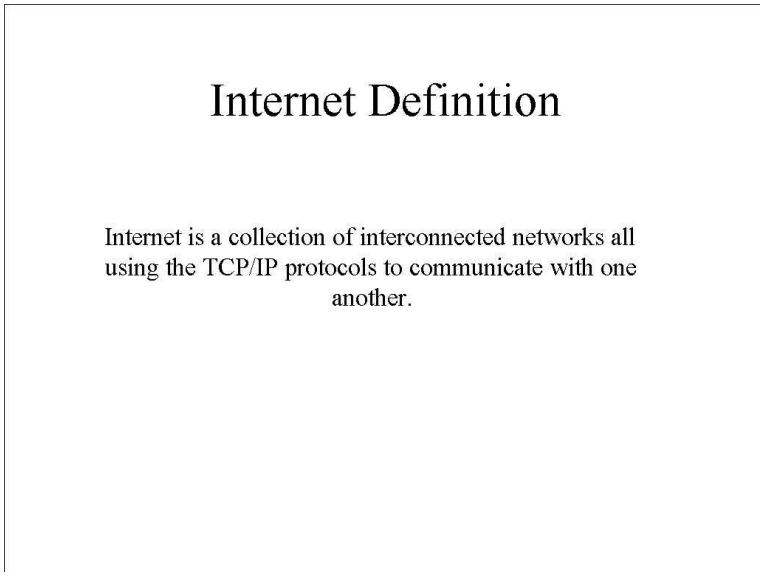
As more systems connected to the network and the amount of information available on the systems grew, new tools had to be developed to easily access the enormous amount of information, and to easily locate items of interest. As cheaper and better hardware became available the user interfaces of the various Internet applications improved and new easy-to-use applications were developed.

Examples of these newer services are WAIS, Gopher and the World Wide Web. These applications will be covered in a separate info shop.

# Hoofdstuk 2

# Internet form and structure

## 2.1 Internet Definition

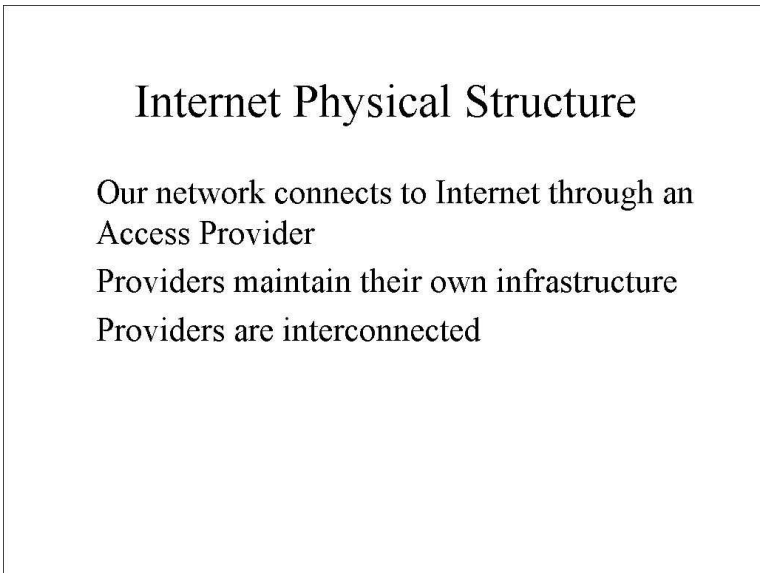



**Notities**

As has been mentioned in the Introduction module of this shop, Internet is a collection of interconnected networks. These networks use the TCP/IP protocols to communicate with one another. Strictly spoken it is not necessary for these networks to use TCP/IP for their intra network communications, but they usually do.

Most commercial online information services (like CompuServe and America Online (AOL)) offer connectivity to Internet in one form or another. Special gateways convert between the propriatry protocols of those providers and the Internet protocols. Other gateways exist to convert from X.400 to Internet and vice versa.

## 2.2 Internet physical structure



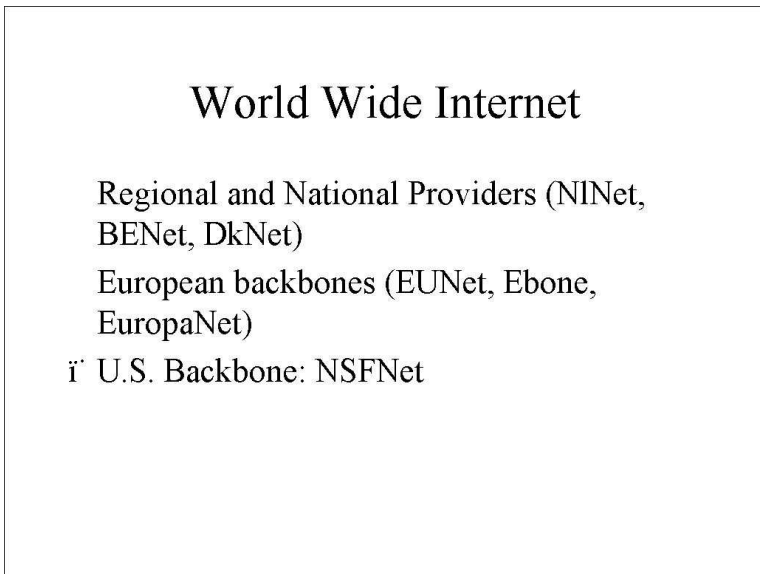

**Notities**

Knowing that Internet consists of multiple interconnected networks, the question “How are these networks connected?” invariably pops up.

The answer is that special organisations exist (providers) that offer Internet connectivity to it’s members and/or customers. These providers maintain their own high-speed backbone infrastructure. If we want to connect to Internet we have to use the services of a provider. Most providers have several Points of Presence (POP) where we can connect to. Usually we connect to the nearest POP of our provider of choice.

Internet providers maintain links between each other and all Internet providers have agreed upon relaying each others traffic. Therefore, if we connect to one provider we can communicate with the rest of Internet through the combined infrastructures of the world’s various Internet providers.

## 2.3 World Wide Internet



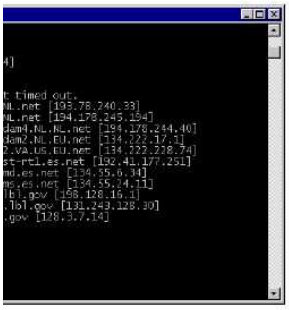

**Notities**

In Europe, each country has one or more national providers. For instance in the Netherlands the two largest providers are NLnet (a spin off from the Dutch Unix User Group - NLUUG) and SURFnet (a government sponsored organisation to connect Dutch universities). National providers usually maintain one or more high-speed national backbones.

The European national backbones are connected to each other using European backbones. Currently EUnet, Ebone and EuropaNet are the best known european network connectors.

This European backbone structure has various connections with the other parts (continents) of Internet. The worlds largest Internet backbone is the National Science Foundation network (NSFnet) in the United States of America. NSFnet currently has over fifteen major nodes connected with 45Mbps links.

## 2.4 Internet structure example



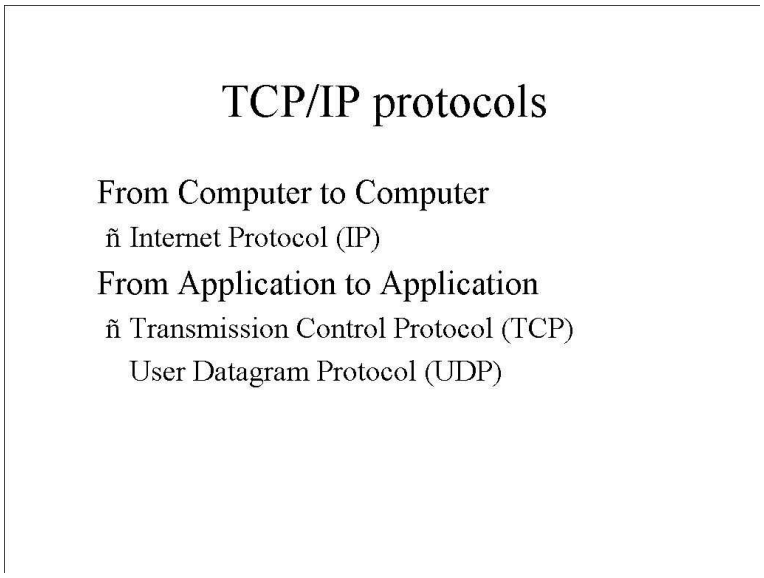

**Notities**

The screen dump shown in the slide provides an example of how Internet is structured. It shows the output of the ‘traceroute’ command. This command traces the flow of packets through Internet as packets make their way to their destination.

The originating system in this example is a PC connected to NLNet’s point-of-presence in Amsterdam. The traceroute command shows how packets find their way to www.lbl.gov, a U.S. government laboratory in California. Each printed line is a router connecting one or more networks.

As we can deduce from the output of traceroute, NLNet’s connects to EUNet’s network in Amsterdam (EUNet is located in the *same building* as NLNet!). EUNet has it’s own links to the United States (from Amsterdam to Vienna). From there on, the traffic is relayed through various other networks before it reaches the lbl network. Even within the lbl network, the packets have to take several hops before the destination www.lbl.gov is reached.

## 2.5 TCP/IP protocols




**Notities**

For one application in Internet to talk to another it is necessary for them to agree on the form and content of their communication. These agreements are commonly called protocols. A large variety of network communication protocols exist: SNA, DECNET, NS and TCP/IP. Most protocols are vendor specific, they were developed by large hard- and software vendors as a means of constructing single-vendor networks.

TCP/IP is different in that it has not been designed and implemented by one particular computer vendor. Instead, TCP/IP has been invented and built by a research group in such a way that it could be implemented on a wide range of computers and/or operating systems. TCP/IP is open.

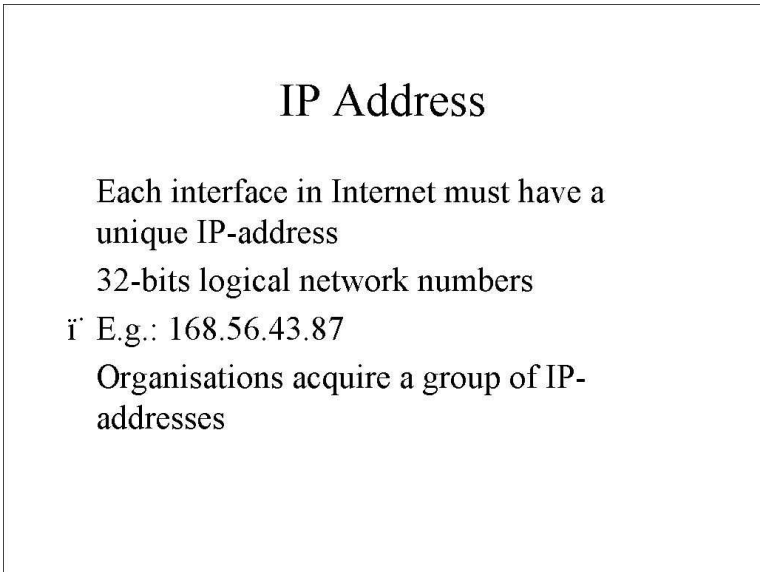
The core protocols of TCP/IP are TCP, UDP and IP. TCP and UDP take care of application to application communications (OSI model layer 4). IP is responsible for routing a packet of information through the network (OSI model layer 3). TCP/IP is not dependent on one data link layer or another.

IP operates with ethernet, token ring, arcnet, serial lines, ISDN, X25 and almost any other means of physically transporting data from one node to an adjacent node.

Other protocols that are often used by TCP/IP hosts are ARP (Address Resolution Protocol), SNMP (Simple Network Management Protocol) and ICMP (Internet Control and Monitoring Protocol).

For more information on TCP/IP and related protocols consult one of the many fine books on this subject.

## 2.6 IP address




**Notities**

In order for IP to be able to route packets through an internet, each IP-interface (a network interface that can be used by IP) has to have a unique IP-address. Such an IP-address is a 32-bit logical network number that uniquely defines this interface in the network. For convenience we usually write an IP-address as a 4- number string: 192.56.72.94

This 32-bit addressing scheme gives us a maximum of 4 billion network interfaces in Internet. Organisations acquire a group of IP-addresses that they can hand out within their own network. For instance General Electric has registered all IP addresses that start with 3. This means that all hardware that is part of the GE corporate network gets an IP- address that starts with 3, e.g. 3.162.13.9. Nobody connected to Internet is allowed to use IP-adresses that start with 3 without GE's approval!

GE can define up to 16 million interfaces in it's corporate network. Other large IP-address owners are Hewlett-Packard, IBM en Digital Equipment Corporation.

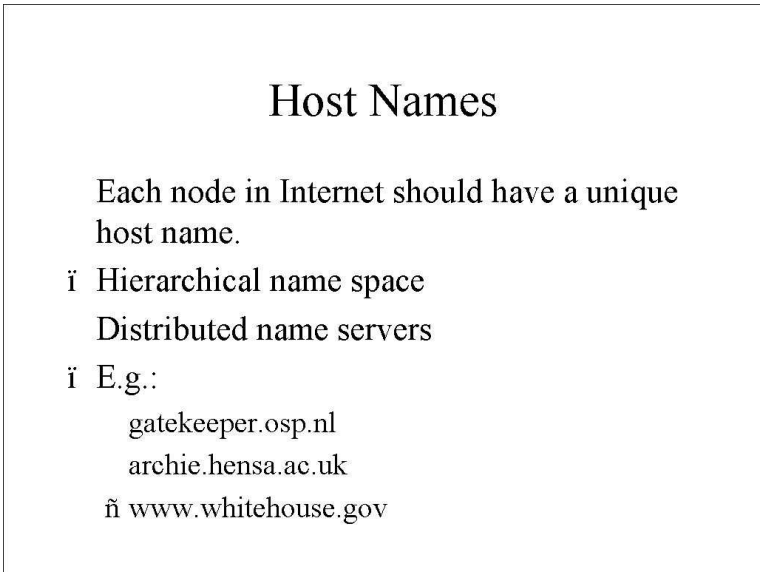
Smaller organisations get a group of IP-address with 2 bytes out of four prescribed. For instance the Rijksuniversiteit Groningen has obtained all IP-addresses that start with 129.125. They can hand out approximately 65.000 IP-addresses within their own network.

Even smaller organisations receive a 3-byte network number. For instance the well known Dutch publisher Thieme has received the IP-address range 194.178.20.*.

It is the network administrator's responsibility to refrain from handing out duplicate or unauthorised IP-addresses. A failure to do so would probably lead to the network being disconnected from Internet by a joint action of some or all other Internet users (mainly

the provider).

## 2.7 Host names




**Notities**

IP-addresses are a great concept for the IP-protocol! Humans tend to find it difficult to remember these cryptic 32-bit numbers. Furthermore, IP- addresses are often related to the TCP/IP technical structure of a network or subnet. To overcome this, each host in an internet should have a fully qualified host name.

These host names should be unique across the entire internet as well. To prevent clashes with other hosts on such obvious names as gw1, server and donald, an internet is structured in hierarchical domains. Each node therefore has a fully qualified domain name (FQDN) that consists of the specific host name and the names of the domains this host is a part of.

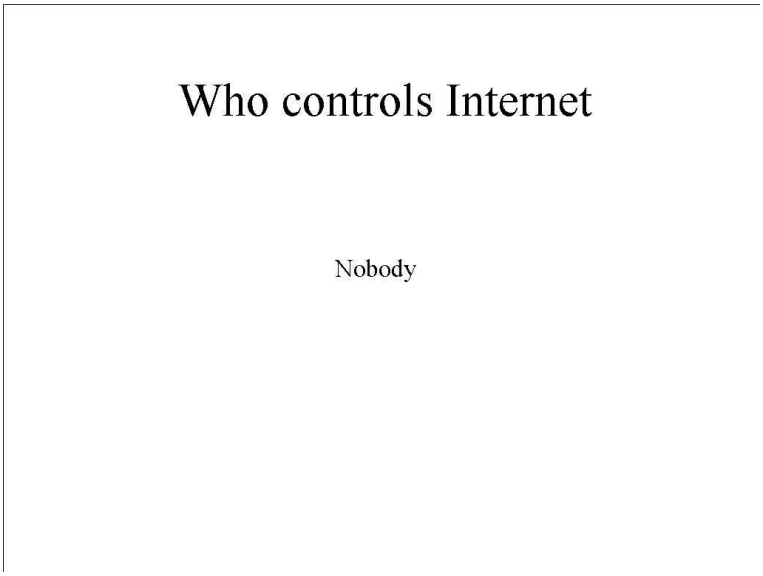
In Internet all nodes are part of the root domain. This root domain is divided in a number of subdomains for commercial companies (.com), US government institutions (.gov), educational organisations (.edu) and domains for all non-US countries (.au, .nl, .dk, .uk). Each subdomain can be subdivided in even more domains and so forth. Registered owners of a domain may subdivide their domains as they please. Examples of domains are:

- hp.com
- neth.hp.com
- macdonalds.com
- osp.nl
- gouda.osp.nl

- ddn.mil
- mtv.com
- ncsa.uiuc.edu

When we instruct an application to contact a certain computer in Internet, we usually do so by specifying the destinations host name. The TCP/IP software needs our targets IP-address in order to route the information through the network. So called name servers resolve our hostname to IP-address and vice-versa queries. All name servers in Internet cooperate with one another in order to determine a nodes IP-address or host name.

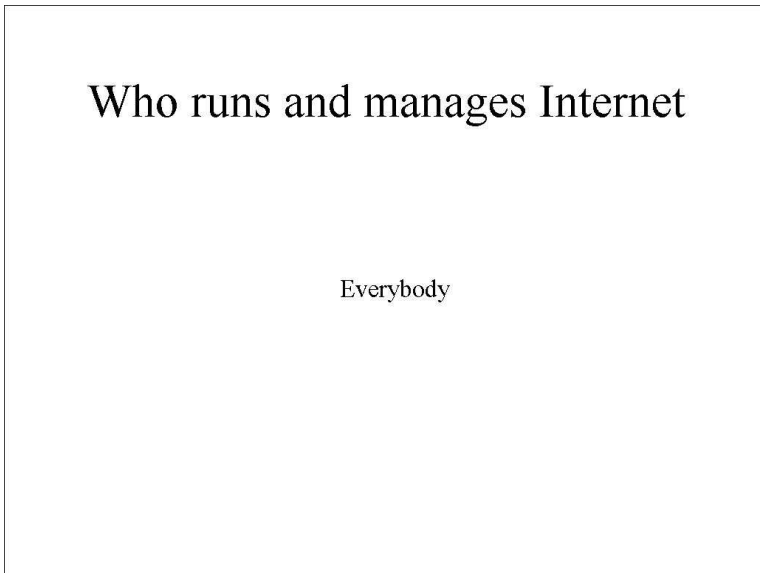
## 2.8 Who controls Internet




**Notities**

Well seriously, not any one organisation, government or institution. Internet is a collection of co-operating networks. No one has complete control over Internet. Compare this with commercial network service providers like AOL and CompuServe!

## 2.9 Who runs and manages Internet



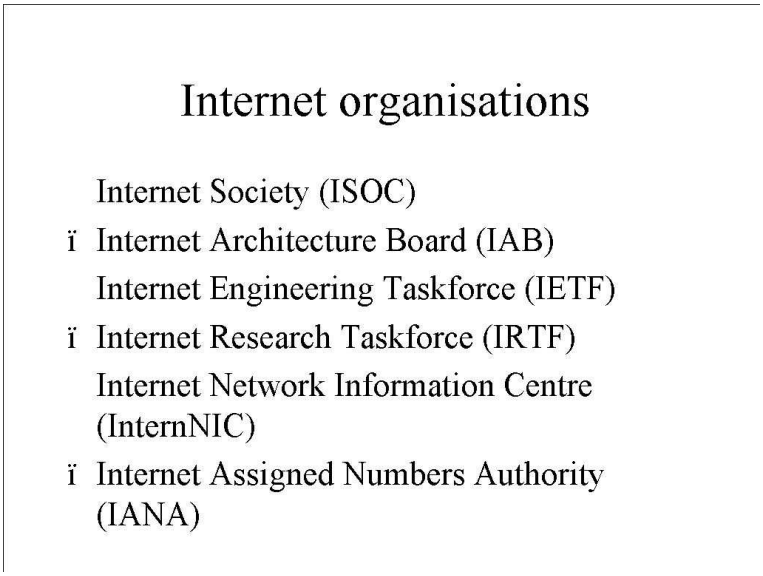

**Notities**

That is, everybody runs his/her own patch and sticks to agreements made with other Internet users. This collective effort keeps Internet running and ensures a lasting commitment from Internet's users to it's general well being.

Of course some Internet users share a greater burden of responsibility than others. Especially the providers and the operators of the large national and regional backbones play a key role in keeping Internet running. Most large providers operate Network Operations Centres (NOC's) from which they continuously monitor their part of Internet.

Apart from these day-to-day activities a lot of administrative chores have to be dealt with to keep Internet organised: IP- address ranges have to be handed out and administered, applications for domain names must be checked and registered and protocol extensions and/or new protocols must be standardised and maintained. For these and other activities a number of organisations have been set up.

## 2.10 Internet Organisations




**Notities**

An organism like Internet, that functions mainly because all parts of it stick to agreements they once made with other parts, needs some central guidance if it is to adapt itself to a changing environment. Within Internet a number of committee's have been set up to address the various organisational and research aspects of keeping a global network the size of Internet alive and well.

### 2.10.1 Internet Society (ISOC)

The Internet Society is the central body of Internet. ISOC promotes Internet and sponsors research into and standardisation of Internet technology and protocols. The members of the society and it's working committees are volunteers from the business, government and educational communities all over the world.

### 2.10.2 Internet Architecture Board (IAB)

The Internet Architecture Board operates under the responsibility of ISOC. The IAB is responsible for the standardisation of Internet technology.

### 2.10.3 Internet Engineering Taskforce (IETF)

The Internet Engineering Taskforce develops the specifications that become Internet Standards. It also provides a forum for ongoing discussion about Internet technology

and operations. The IETF also operates under the hood of ISOC. The engineering task force focuses on semi-immediate solutions for the Internet problems of today. As such it does not engage itself with long term or speculative research.

### 2.10.4 Internet Research Taskforce (IRTF)

The Internet Research Taskforce pursues long-term research projects. As such the IRTF does occupy itself with speculative research and long-term solutions for tomorrows problems.

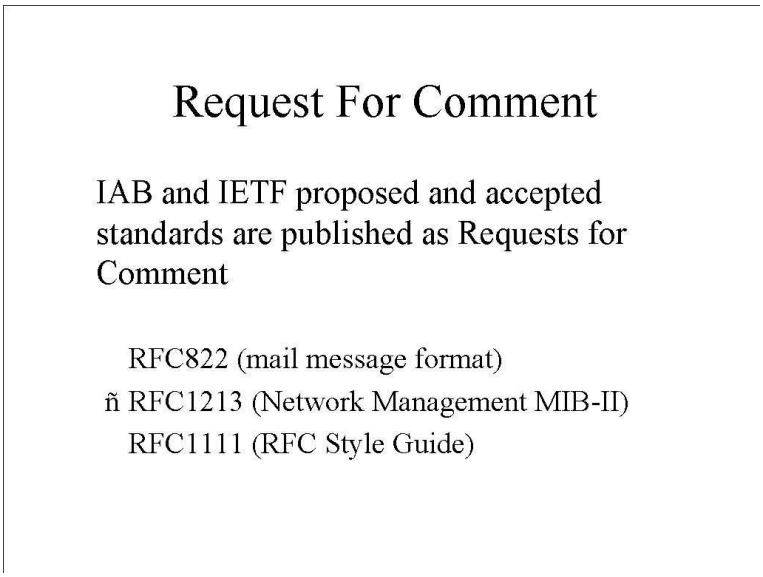
### 2.10.5 Internet Network Information Center (InterNIC)

The Internet Network Information Center is a central organisation that hands out and registers domain names and IP-address ranges. The InterNIC plays a key role in connecting new networks to Internet. Unfortunately, due to the enormous amount of domain name and IP-address applications currently swamping InterNIC a number of errors have been made, such as handing out macdonalds.com to Wired magazine and mtv.com to Adam Curry (ex-MTV veejay, ex-Veronica deejay). To regulate the amount of work InterNIC has to do it delegates parts of it's work to regional NICs (such as RIPE, the Reseaux IP Europeens). These regional NICs can further decentralise their operation to other NICs.

### 2.10.6 Internet Assigned Numbers Authority

Regularly IETF-proposed and adopted standards contain variable port numbers (TCP and UDP), enterprise id's (SNMP) or other identifications that should be kept the same throughout Internet. The Internet Assigned Numbers Authority (IANA) standardises these elements within the net. IANA does not devise new standards, but merely regulates the implementation of some aspects of existing standards.

## 2.11 Request for Comment




**Notities**

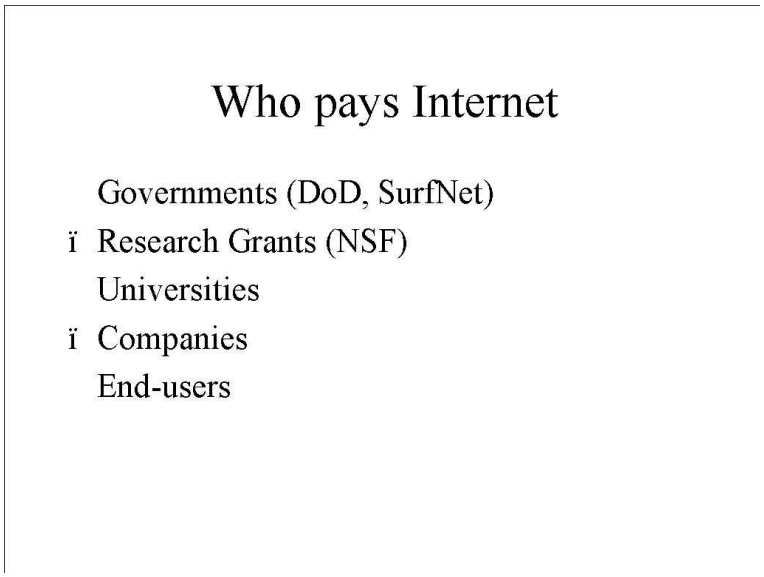
As Internet is based almost entirely on agreements about the various protocols, it is very important that there are unambiguous documents describing those agreements. As Internet and it's protocols are not designed and implemented by one single entity the standardisation process is very democratic.

IAB, IETF and IRTF form working groups to invent and/or propose new standards or extend existing standards. Participants in those working groups are volunteers from the government, education and business communities. The proposals of those working groups are written down as 'Request For Comments' (RFC). After approval by the all relevant Internet bodies an RFC gains the status of Official Internet Standard.

RFC's are a welcome relieve for those of us that have read standards defining documents from other organisations. RFC's are well written, easy to read, concise and generally very helpful. Reading RFC's is certainly not restricted to network software implementors or network consultants. In a lot of cases network and system administrators can benefit from RFC's as well.

RFC's can be downloaded from several so called Internet Repositories in Internet. The file rfc/rfc-index.txt contains an overview of all RFC's. Other indexes point to RFC's by title and RFC's by author. Apart from RFC's the Internet Repositories also store FYI-documents (For Your Information) and other interesting material.

## 2.12 Who pays Internet




**Notities**

The distributed and anarchistic nature of Internet where organisations offer free services and everybody uses everyone else's infrastructure leads to strange and complex financial questions.

Large organisations who use Internet to support their core activities (mainly companies and universities) finance their own infrastructure and usually install and maintain their own links to their closest and most important Internet neighbours.

Smaller companies and institutions link to Internet using regional or national providers. Those providers usually charge a fee for their services and those fees finance (part of) the provider-to- provider connections.

In the United States the large national backbone (NSFNet) is paid for directly by the U.S. Government through the National Science Foundation and the Department of Defence.

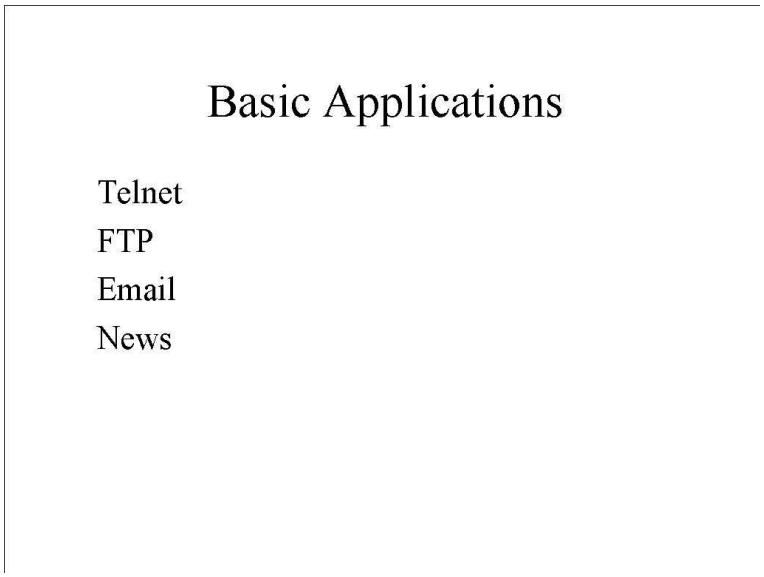
Apart from the infrastructure the various services offered on Internet cost money as well. Usually organisations offer those services for free, based on the principle of reciprocability. Examples of those free services are the file transfer sites, the archie databases, ftpmail and email store/forward.

The growing commercial use of Internet puts pressure on this unique way of financing infrastructure and free services. In the next few years we must devise a new way to pay for infrastructure and services.

# Hoofdstuk 3

# Basic Applications

## 3.1 Basic Applications




**Notities**

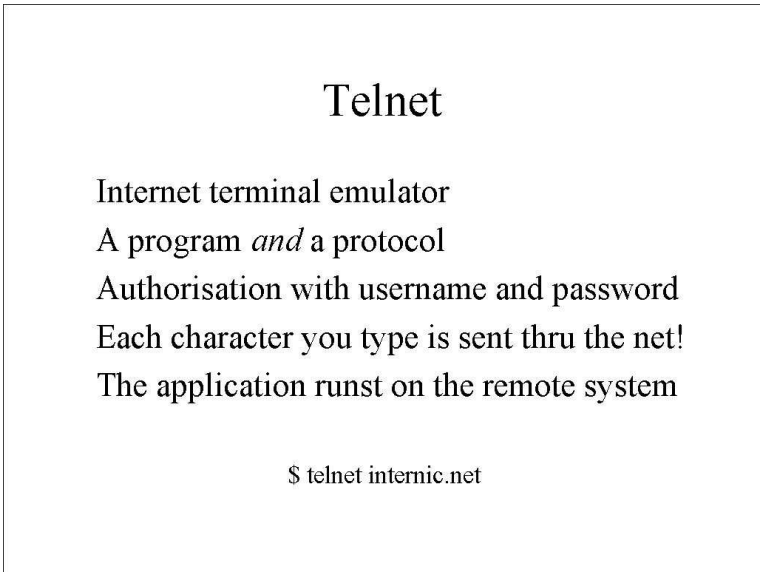
This chapter introduces the classic Internet applications, like telnet, ftp and mail. The most popular application, the World Wide Web, has a separate info shop devoted to it.

Nowadays, being spoiled with all kinds of advanced networking tools and applications, one might wonder why to get excited about the early Internet applications. Old as they might seem to you, they are still used frequently to date. Telnet and ftp are the standard tools to login on remote systems and transfer files all over the world. If it hadn't been for these old-fashioned applications, Internet wouldn't be what it is today.

Also, a lot of the fancy applications you might be using today are using the ftp or telnet protocol.

Although e-mail in it's basic form is a classic application, many changes to the original protocol have been proposed by means of RFC's.

## 3.2 Telnet




**Notities**

Using telnet, one can establish a terminal connection to a remote system, by either specifying the hostname or IP address of the system you want to connect to.

The telnet protocol, used by the telnet program, is very widespread. In fact, the terminal emulator program you probably use to connect your PC to your UNIX host, is very likely to use the telnet protocol, as are many terminal servers.

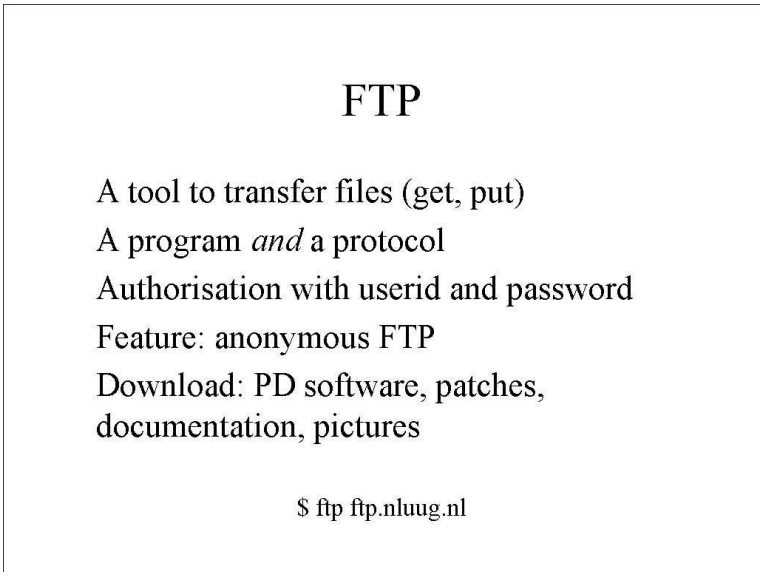
Telnet has become the protocol for remote login in the UNIX community, but can also be used to connect to non-UNIX hosts. Telnet, although not the most efficient protocol, surely is very important in the Internet, and can even be used to connect to other than login-services, like the sendmail daemon or a simple World Wide Web browser.

## 3.3 Telnet example


**Notities**

On the slide you can see a screen dump of a telnet session. This telnet client (Windows/NT 4.0) has been started with 'telnet internic.net'. InterNIC runs a service through which you can obtain information on registered domain names and IP-address ranges. When telnetting to this site you do not need to login using a username and password!

## 3.4 FTP




**Notities**

FTP (file transfer program) is an application used to transfer files from a remote system to your local system and vice verse.

In order to get a file from a machine, you need to know it's name and the name of the machine to get it from.

To load a file from a local machine called 'merlin', you could type:

```
$ ftp merlin
```

merlin will respond prompting for a username/password combination.

When we successfully logged in, we can either 'get' or 'put' a file.The commands used are 'get' and 'put' and can be typed at the ftp prompt:

```
ftp> get froboz.doc
```

There are many other commands available at the ftp prompt, like 'binary' to force binary transfer mode, necessary to transfer non- ASCII files.

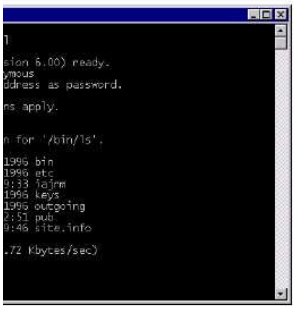
Basically FTP is a secure means of copying files from one system to another. The FTP server authenticates a request using the user name and password specified when the FTP session started. You can not retrieve or send files that you would not have been able to manipulate had you logged on using telnet!

To create a general file server, system administrators can enable anonymous ftp. With anonymous ftp everybody can connect to the FTP server using the userid anonymous or

ftp. You are usually expected to enter your name or email address as the 'anonymous' password, although this is usually not verified. Using anonymous ftp you are restricted to specific parts of the FTP server's file systems. Only those files explicitly set up for anonymous retrieval can be downloaded.

Likewise, you can only upload files to certain directories.

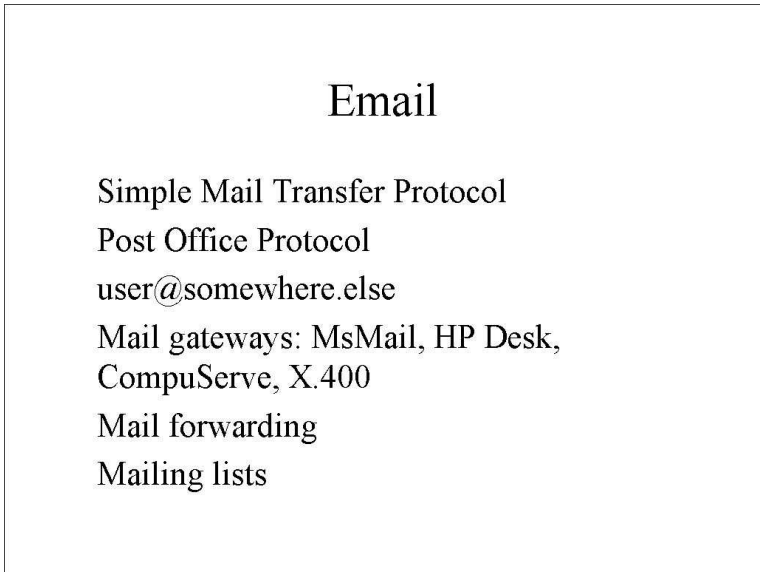
## 3.5 FTP example




**Notities**

The slide shows an example of the well known ‘line mode’ FTP client. A lot of people have allergic reactions to FTP because of this somewhat archaic interface. Rest assured, a lot of GUI implementations of FTP float around the Internet as well. Your WWW-browser can act as an anonymous FTP client as well (using a URL like: ftp://ftp.nluug.nl).

## 3.6 Email




**Notities**

Electronic mail (or e-mail for short) is used to transfer mail messages by electronic means (i.e. Internet). It can either be used to transfer files within your own organisation, or send messages all over the world. System administrators can use mail to tell their users to cleanup their files. The backup procedure that runs nightly, can mail the system administrator the results of the daily backup.

When mail is to be sent to someone on the local system, the (UNIX) mail command can be used like:

```
mail bill < message
```

message is a file containing the message to send. User bill will be notified when the mail has arrived, and could for instance reply on it.

As long as mail stays within one machine, addresses are easy to understand, as they match the user's name, and are therefore easy to remember. When mail has to be sent to other systems, things become a bit more complicated.

In the old days, one had to know the systems in between your own and the recipient of your message, as in:

```
mail mcvax!uunet!whitehouse!president < message
```

meaning that the message had to travel from your machine to mcvax, from mcvax to uunet, from uunet to the whitehouse, and there to the mailbox of the user president. It is like telling the postman to take the second street on his left and turn right at the traffic lights.

These kind of addresses are called UUCP addresses, as they were used with the UUCP command. UUCP is short for UNIX to UNIX copy and was used to transfer files from one UNIX site to the other, usually using modems to connect systems over larger distances. Email, in the old days, was usually sent by uucp.

One of the many disadvantages of UUCP was that it took long for a message to get somewhere (many connections were dial-up connections that were only established once a day) and there was a fair chance of making a typo somewhere in the, sometimes extremely long, addresses, resulting in a return of (a part of) the message. Also, in case one host went down, the mail couldn't be re-routed dynamically.

Nowadays, however, we don't have to specify for a message how to travel, just were to travel to. Routing is done by complex systems, rather than mankind, resulting in email addresses like:

```
mail president@whitehouse.gov < message
```

The @-sign is pronounced as 'at'.

How the message needs to be sent is of no concern to your average user. The network administrator's job is to make sure you don't even have to think about it, but s/he does.

The single most important tool that enables us to mail to addresses like above it is sendmail. To configure sendmail (as opposed to merely changing the domain name) is a difficult task, but can be administered by the administrator, rather than each individual user.

Fortunately, there are gateways so that we can now for instance send email to an Internet address from a CompuServe account, all without telling the postman how to travel.

Electronic mail in Internet is based on Simple Mail Transfer Protocol (SMTP for short). Non-SMTP mailers need SMTP gateways to connect to Internet. Besides the mailer used in the example, the UNIX command mail, many other mailers are available, like Microsoft mail that comes with most versions of MS Windows. Gateways using SMTP are available, so one could connect say a Windows for Workgroups network to 'the real world'.

When we look at the header of a message, we can see how the message traveled. We won't discuss the individual lines of the header, but as the example shows, a message sometimes has a long way to go.

Now, of course you only have to know the address of the person you want to send mail to.

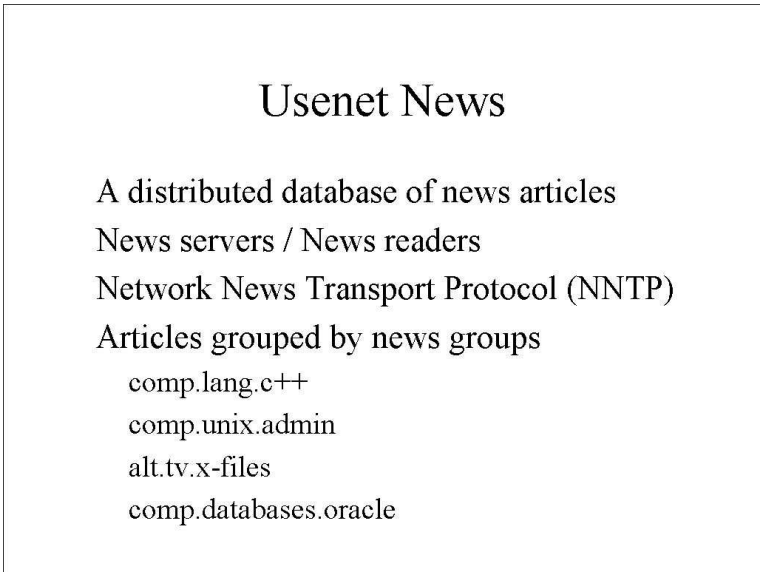
## 3.7 Email example


**Notities**

The slide shows an example of the UNIX full screen program ‘elm’. Although not the most user friendly email program in the world, elm is more or less usable and has the added advantage that it is available on almost every UNIX system in the world.

More modern email interfaces are Eudora, Microsoft Internet Mail and Netscape Mail.

## 3.8 Usenet News




**Notities**

The News system is probably the single most used traditional Internet service of all times. Through News you can ask questions, pose opinions, help fellow net surfers and discuss your favourite cooking style.

News consists of a distributed database of news articles. This entire database is distributed across all the news servers in Internet. Using the net and the network news transport protocol, news servers regularly update each other (sending new news).

Using news readers users connect to a news server. These news readers allow people to browse, search, answer and post news articles. The amount of news transported through the net still rises each year and is currently said to amount to 35MB/day (imagine: 35MB of mostly ASCII text articles).

News articles can be questions about using a certain program or operating system, opinions on some topic close-at-heart, announcements of new releases of PD-software, reports of bugs and security holes or just plain smalltalk. Users can attach an answer to an article and that answer will be sent to all other news servers as if it were an article itself (which in fact is true).

To create some order in the tens of thousands of new articles posted each day, all articles are grouped together in news groups. These news groups have descriptive names like ”comp.lang.c”, ”comp.security.unixänd ält.startrek.creative”. New newsgroups spring into existence almost every day. Currently there are over 7000 different newsgroups.

Some newsgroups are moderated. This means that someone (a volunteer) guards the quality of the articles in the newsgroup. Usually new articles may not be posted directly in such newsgroups but should be mailed to the moderator. To clean a newsgroup from

questions that are asked regularly, special FAQ (frequently asked questions) articles are prepared and posted. A user is invited to check the FAQ first before posting a simple question.

The quality of articles (and newsgroups) ranges from extremely high to extremely low. Internet is a non-censored information highway! Next to very professional newsgroups we have newsgroups on almost every hobby and interest imaginable (and unimaginable).

System administrators setting up a news server for their own organisation can determine which news groups are available to the users of their server. Furthermore, special newsgroups can be made that never leave the organisations network.

## 3.9 News example 1


**Notities**

The slide shows a screen dump from the ‘tin’ newsreader. Like elm, tin is an ASCII-only, full screen program whose main attraction lies in the fact that it is available on almost every UNIX system. Various companies have developed and sell GUI-based news readers. Several public domain graphical news readers are available as well.

The screen shows a subset of the news groups that are available on the news server. By subscripting to news groups the list is truncated to contain only the news groups you are interested in. By selecting a news group we get a list of the articles that the news group contains.

## 3.10 News example 2

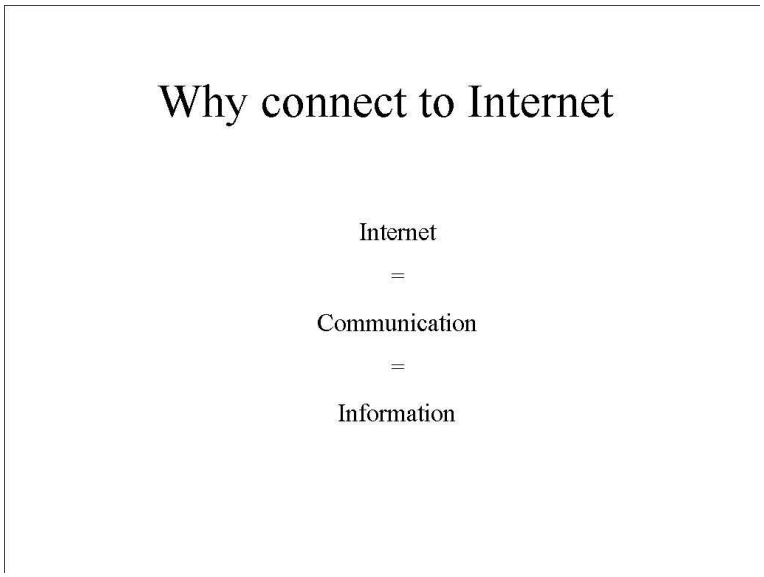

**Notities**

A news group contains articles that have been posted by other Internet users. The news reader program allows us to read articles, post responses, post new articles or reply to articles by email

# Hoofdstuk 4

# Connecting to Internet

## 4.1 Why connect to Internet




**Notities**

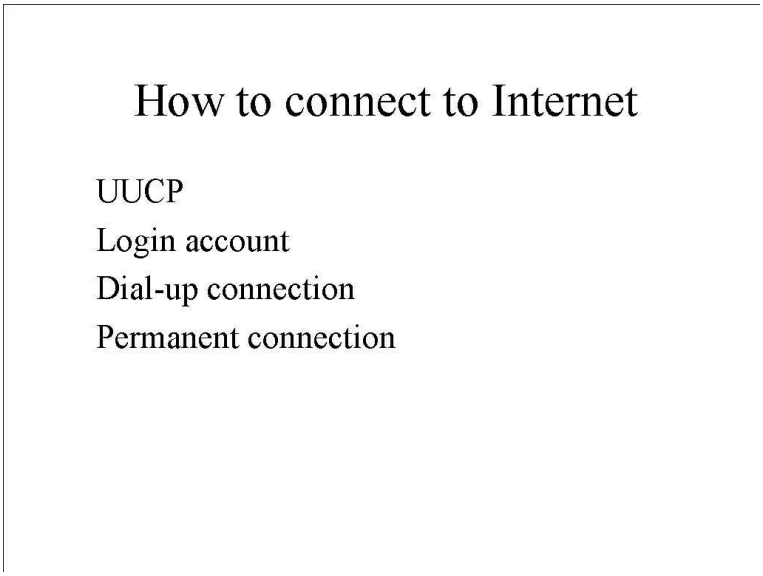
It is very clear that throughout history humanity has increasingly used better, faster and more advanced ways of communicating with one another. As our means of communication improved, so did our span of control. The availibility of information is central to making decisions, solving problems and almost every other modern business process.

It is very clear that in the not so very far future a global computer network will link almost every home and office in the western world, much as the telephone system does so today. Internet is currently the only network that can pretend to be able to grow into this Information Super Highway.

If you want to be able to communicate directly with your clients and collegues, for whatever reason, then connecting to Internet is almost a nescessity. As more companies and governments connect, the need for other organisations to connect as well will increase.

Current Internet applications and services are but a tiny example of the things that are possible in a world wide network. These services will steadily improve and companies will create and setup new services to which you will want to subscribe. Your organisation currently uses a wealth of traditional information services: the telephone system (voice and fax), the postal services, newspapers, magazines, libraries etc. These will undoubtedly be supplemented or replaced by net oriented services.

## 4.2 How to connect to Internet

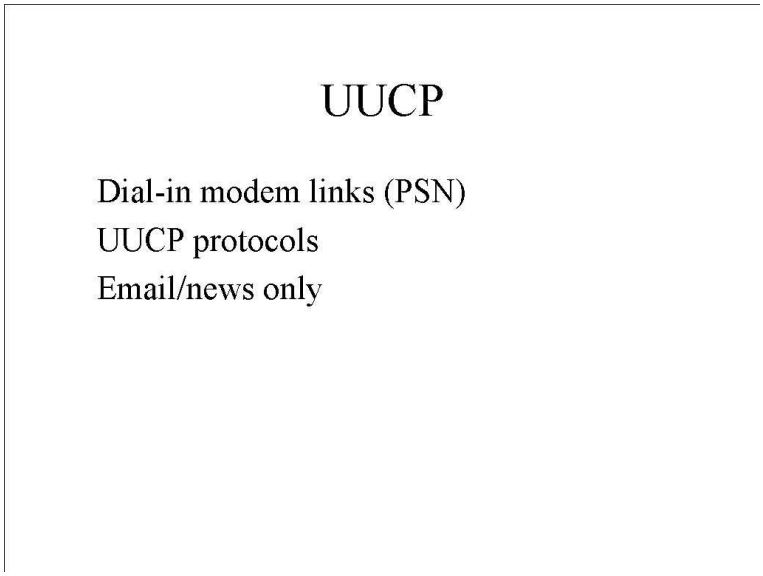



**Notities**

Now that you have seen some of the fascinating aspects of Internet, you probably wonder how to join the club.

In this chapter we will list the more popular ways to get connected. To connect to Internet you need an Internet Provider. Several providers exist today and new ones will startup over the coming years. Which provider you want to connect to depends on the kind of sevice you request and the providers nearest Point-of- Present (PoP)

## 4.3 UUCP




**Notities**

If you only want to read and send mail and news, UUCP might be the thing you're looking for. Instead of being online to your provider when reading mail and news or posting messages, UUCP sends and receives messages from and to your own computer. This means you can write mail messages and news- articles while you are offline. Also, you can have a domain name for your organisation, so that your e-mail address (assuming you live in the Netherlands) would be something like

```
you@your-organisation.nl
```

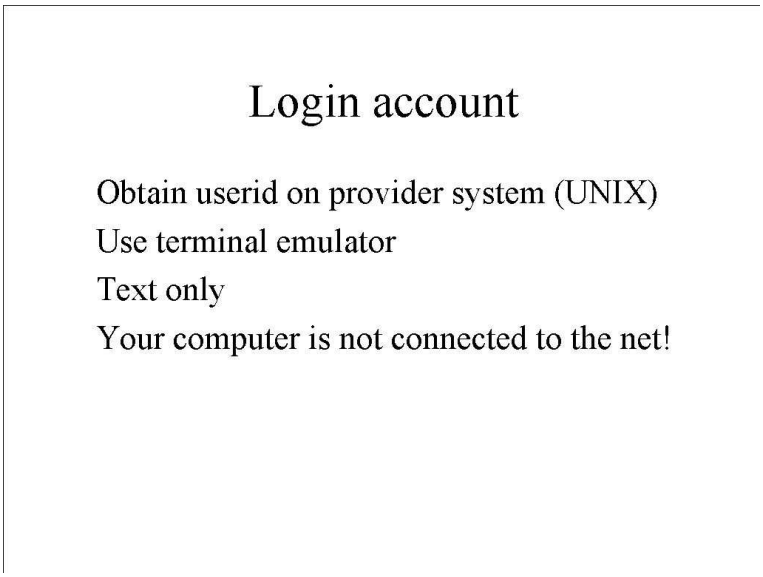
In order to be able to get this up and running, you locally need to run UUCP software, which comes standard with most UNIX systems, and a UUCP account with your provider.

Instead of you logging in, UUCP establishes a connection and transfers files, containing mail and news, to and from your system.

In the early days of electronic mail, this was the general protocol used. UUCP is a store-and-foreward system, storing files for your 'node' at your provider's, until you collect them, using a program called uucico. Note that your site always initialises the connection, reducing the risk of dial-in attacks to your network.

UUCP software is also available for non UNIX systems, like VMS, DOS and MAC. When opting for a UUCP connection to your provider you can not provide Internet services!

## 4.4 Login account




**Notities**

When opting for a login account you obtain a user account on a provider system (usually a UNIX system). Using a terminal emulator with dial-out support, you connect to the system of your provider. Your personal mailbox is on your provider's system and so are the files you ftp. You don't have a domain or IP address of your own, you are a member of the provider's domain.

In the Netherlands Simplex is one of the many providers of this kind of service. Simplex has a registred domain in the 'nl' domain, called Simplex. Thus, your e-mail address would be

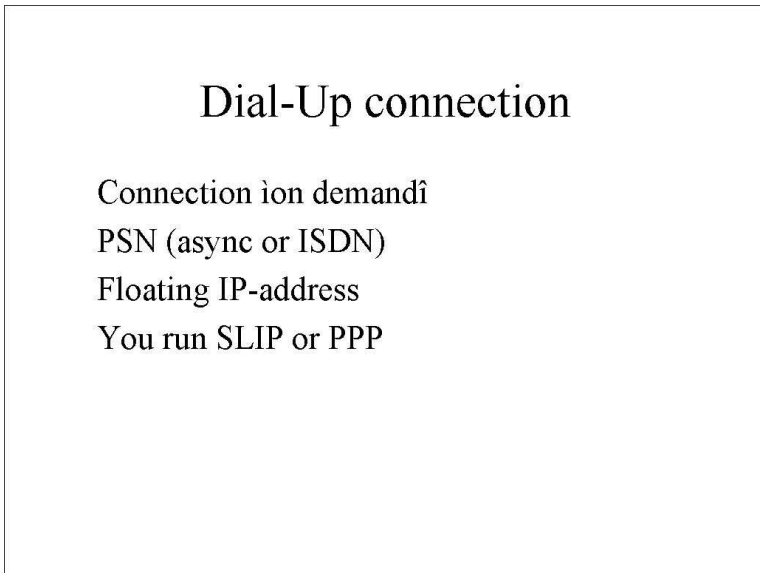
```
you@simplex.nl
```

Also, you can post messages to the news, ftp files to your directory at simplex's hosts and much more. Data you receive, either by e-mail or news, remain on Simplex hosts, unless of course you download it from there using kermit, Zmodem or any other supported protocol.

This kind of service is usually used by individuals, especially because companies probably don't want to be addressed as someone in the Simplex domain. Your organisation probably wants a domain of it's own.

You are usually not capable of providing services using a login account. Some providers are willing to run a Web or Gopher server for you (extra charge of course).

## 4.5 Dial-up connection




**Notities**

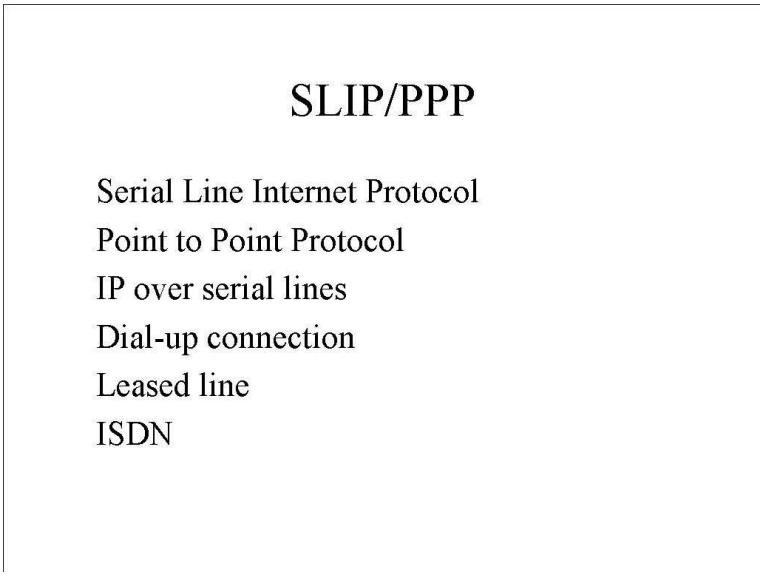
The most advanced ‘simple’ way of connecting to Internet is by using a TCP/IP dial-up connection. Most private citizens connect to Internet this way. When using a dial-up connection you subscribe with an Internet access provider providing these kind of services.

The basic idea is that your computer runs special software (dial up stack) that can run TCP/IP over serial lines (SLIP or PPP). This software dials the PoP of your provider (using a modem and a fairly standard telepone line) and after authenticating yourself with a userid and password establishes a TCP/IP connection. Your computer is part of the Internet as long as the connection is open.

This means that you can run TCP/IP applications (like Netscape) directly on your own computer. These applications can setup TCP/IP connections directly from your computer to the server. In theory you could start providing services as well, however, your computer is not registered with a name server and it’s IP-address usually changes from one dial-up session to another.

The software needed to setup serial TCP/IP connections is bundled with all major modern operating systems: Windows 95 (Dial-up networking), Windows/NT (Remote Access Services) and UNIX (ppl, pppd).

## 4.6 SLIP/PPP




**Notities**

As the quality of telephone lines increased, and dial-time reduced (no operators to connect to a, possibly wrong, number nor waiting for the dial-wheel to return to it's original position) it became possible to use the IP protocol over phone lines.

Two protocols are very much used and widely available in this sense; SLIP (Serial Line IP) and PPP (Point to Point Protocol).

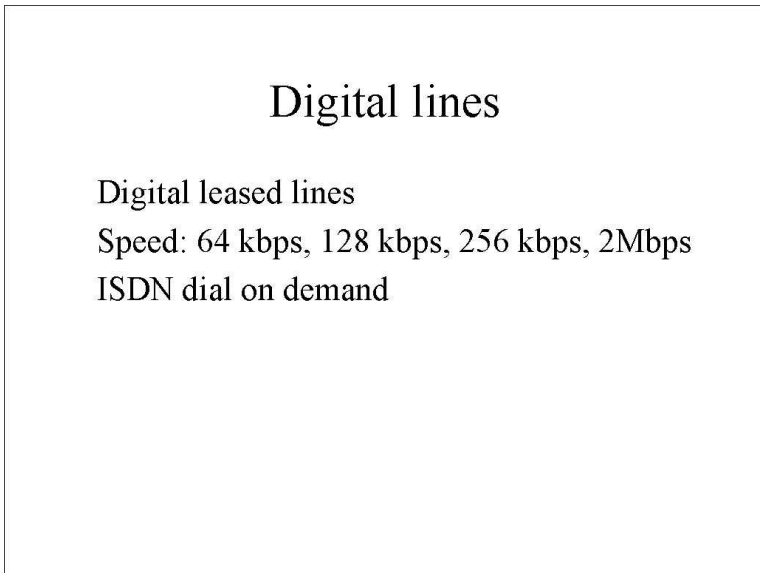
As with UUCP, SLIP and PPP are in no way limited to UNIX, but as opposed to UUCP, you can run all Internet client applications at your own site, instead of only mail and news.

New versions of PC operating systems like OS/2 come with built in SLIP support, so you can run your favourite Internet applications direct on the PC at your desktop. All you need is a SLIP provider.

SLIP provides a means to talk IP over a serial line (generally using modems). Your system becomes a node in the Internet. Depending on the configuration, it might be that your IP address changes every time the SLIP connection is established, but that is not necessarily the case.

Using SLIP, you could even set up a gateway from your local network to Internet, enabling other systems in your network to connect to Internet. As with a local network connection, several IP-connections can be established over one single SLIP line, by multiplexing the connections. This, however, requires a high- speed connection.

## 4.7 High Speed Digital Lines




**Notities**

In order to become 'a permanent member of the Internet society', and provide services rather than merely consuming them, that is to become a prosumer, real IP connectivity is required. This is achieved by placing a router at your site, that takes care of routing IP packets from your network to the outside world (i.e. Internet) and also handles IP packets coming from the net to your site. That implies that not only you can connect to other systems, using ftp and telnet, but they can also connect to your systems.

The router at your site is linked to a similar router of your internet provider. In order to set this up, your site needs to have a registered IP network address. Also, remember that every IP address on Internet need to be unique in order to be able to address every single node in the network.

Routers often offer filters so that only those packets you allow can travel through the router into your network. One could for instance allow ftp to one's domain, but deny telnet packets.

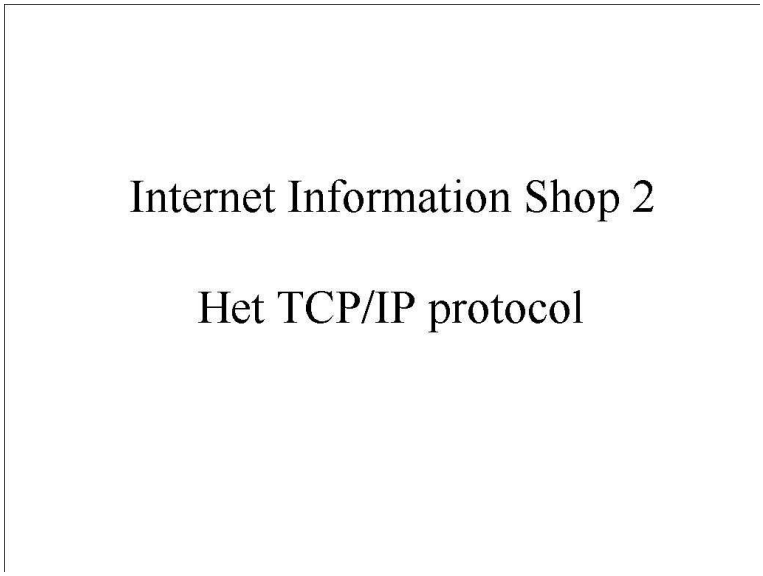
# Deel II

# De kern van Internet: het TCP/IP protocol

# Hoofdstuk 5

# Introduktie

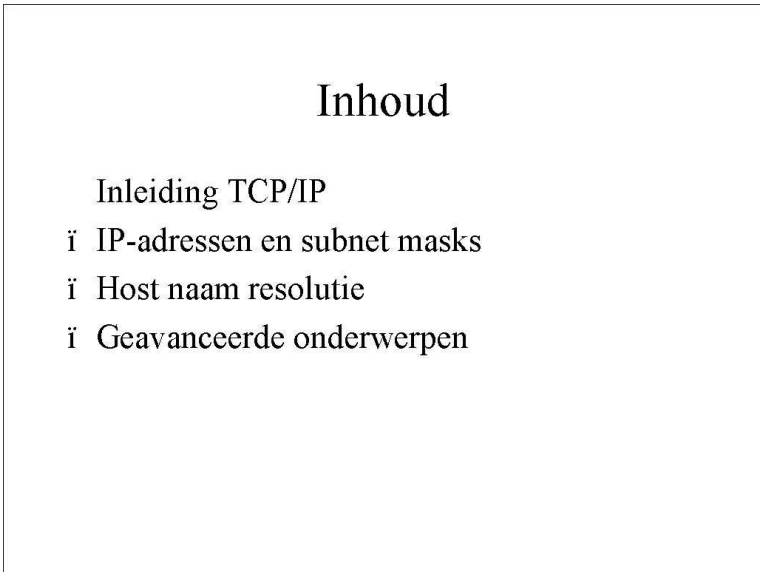
## 5.1 Welkom




**Notities**

Welkom bij deze tweede Internet Info Shop. In deze info shop gaan we dieper in op het TCP/IP protocol. Deze shop is niet bedoeld voor systeembeheerders die uitgebreid willen weten hoe TCP/IP op hun specifieke computer dient te worden geconfigureerd. De nadruk ligt op het beschrijven van de concepten en implementatie van TCP/IP. Enige technische diepgang wordt daarbij niet geschuwd, maar staat niet centraal.

## 5.2 Inhoud




**Notities**

In de shop gaan we in op:

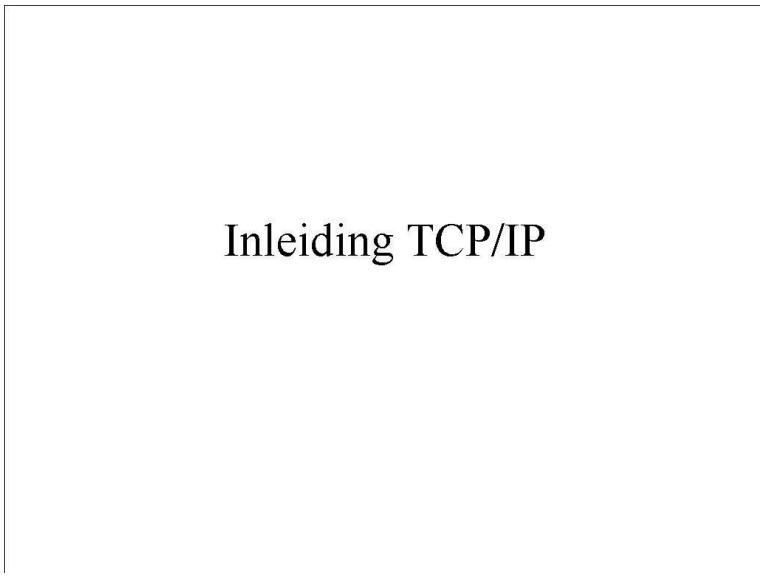
- De structuur van TCP/IP
- De adressering en naamgeving van nodes in TCP/IP netwerken
- Hoe hostnamen worden gekoppeld aan IP-adressen en omgekeerd
- TCP en UDP poortnummers
- Routering in TCP/IP netwerken
- Sockets
- IP next generation

De laatste vier onderwerpen hebben het label “geavanceerd” gekregen en zullen alleen worden behandeld indien het verloop van de shop daar gelegenheid toe laat.

# Hoofdstuk 6

# Inleiding TCP/IP

## 6.1 Inleiding




**Notities**

In deze module wordt ingegaan op de noodzaak en structuur van netwerkprotocollen in het algemeen, en TCP/IP in het bijzonder.

## 6.2 Het probleem

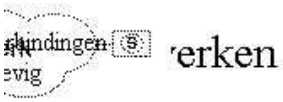

**Notities**

De kern van de gehele netwerkproblematiek kan worden omschreven als de wens van twee applicaties die niet in dezelfde computer draaien om informatie uit te wisselen. Denk hierbij aan:

- Het raadplegen van een remote database
- Het versturen van een electronisch postbericht
- Het versturen van een document naar een remote printer
- Het kopiëren van bestanden van een computer naar een andere

Daar de applicaties niet in dezelfde computer draaien zijn de traditionele wijzen van gegevensuitwisseling tussen applicaties (via diskbestanden of het interne geheugen) niet van toepassing. Indien beide computers fysiek met elkaar zijn verbonden zal de uitwisseling plaats moeten vinden over het netwerk. Via speciale hardware kunnen bytes worden omgezet in electrische of optiesche signalen en vice versa. Dit biedt in principe de mogelijkheid om twee applicaties gegevens uit te laten wisselen.

## 6.3 Eigenschappen van netwerken




**Notities**

Netwerken hebben echter een aantal zeer vervelende eigenschappen:

1. Complexe structuur
   In een groot aantal gevallen zullen de computers waarin de applicaties draaien niet rechtstreeks met elkaar verbonden zijn. Netwerken hebben meestal een complexe structuur en om succesvol te kunnen communiceren dient de informatie dus door het netwerk te worden geloodst. Het zoeken van de optimale route door het netwerk is niet triviaal.

2. Storingsgevoelig
   Netwerken bestaan uit allerlei soorten verbindingen op electrische, electromagnetische en optische basis. Deze verbindingen zijn gevoelig voor magnetische en fysieke verstoringen. De data die dan toevallig over de lijn reist kan dan geheel of gedeeltelijk worden verminkt. Ook het tijdelijk of permanent uitvallen van de verbinding behoort tot de mogelijkheid.

3. Verschillende soorten verbindingen
   Netwerken zijn vaak opgebouwd uit zeer verschillende soorten verbindingen. Denk hierbij an telefoonlijnen, ISDN, ethernet segmenten, token ringen, microgolf radio verbindingen en glasvezel kabels. Al die verschillende “kabels” moeten op hun eigen specifieke wijze worden aangestuurd en kennen hun eigen mogelijkheden, snelheden en beperkingen.

4. Aan verandering onderhevig
   De structuur van een groot netwerk verandert continue. Er komen verbindingen bij, de verkeersdrukte varieert, verbindingen vallen weg of krijgen juist een grotere capaciteit.

De wijze waarop informatie door het netwerk moet worden gestuurd moet dus met al deze factoren rekening houden. Vanzelfsprekend wenst een applicatieprogrammeur of gebruiker niet met deze achterliggende technische zaken te worden vermoeid!

## 6.4 De oplossing: netwerkprotocollen

De oplossing: netwerkprotocollen

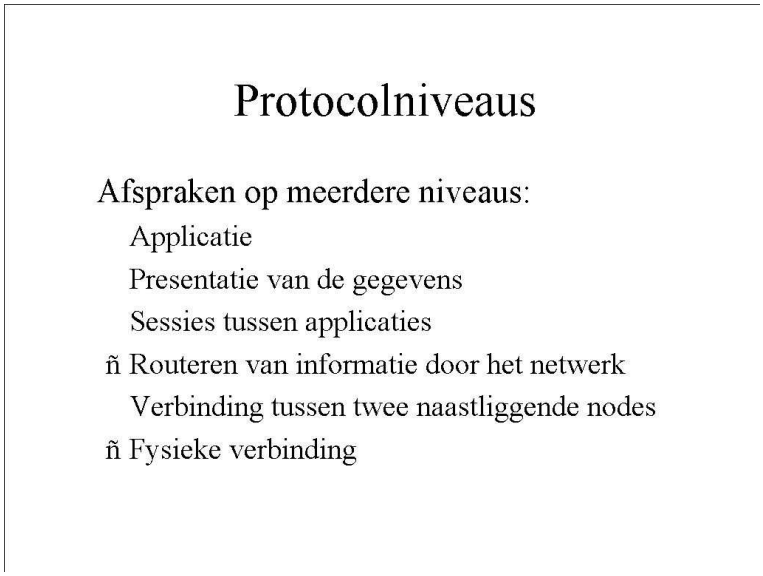
- Afspraken over:
  - Vorm en inhoud berichtenuitwisseling
  - Identificaties nodes en applicaties (netwerkadressering)
  - Verzending van berichten door het netwerk (routering, prioriteit)
  - Ontvangstbevestiging en herverzending
  - Fysieke toegang tot het netwerk
  - Maximale grootte van de pakketten

**Notities**

De oplossing voor al deze problemen wordt gevormd door het ontwikkelen van specifieke stukken programmatuur die de uitwisseling van gegevens via het netwerk voor hun rekening nemen. Deze netwerkprogrammatuur dient alle met de verzending samenhangende problemen te ondervangen. Deze programmatuur moet natuurlijk onderling goed samenwerken. Het is van groot belang dat de netwerkprogrammatuur van de communucatiepartners hetzelfde idee hebben over hoe de communicatie is gestructureerd.

Deze afspraken noemen we *netwerkprotocollen*. In de netwerkprotcollen wordt uitgebreid vastgelegd hoe de verzending van informatie door het netwerk plaatsvindt.

## 6.5 Protocolniveaus



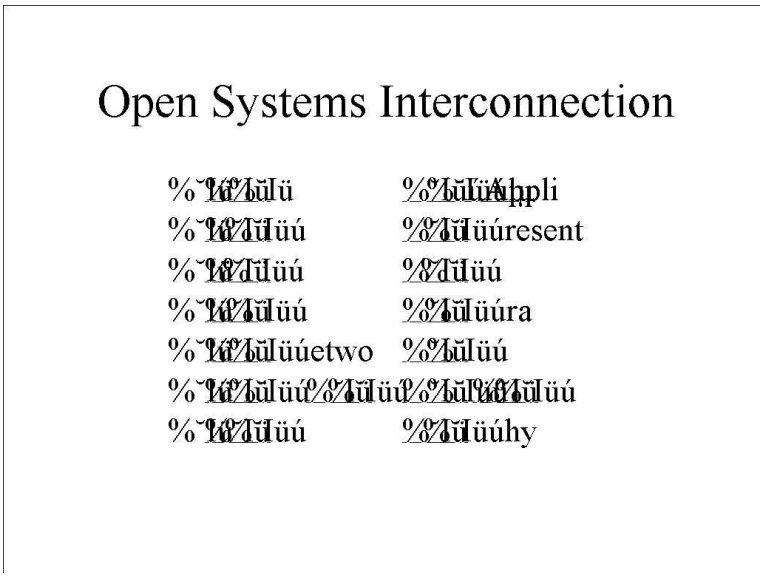

**Notities**

In het verleden is er wel netwerkprogrammatuur ontwikkeld volgens het “kluitmodel”: alle functies van de communicatie geïntegreerd in één groot programma. Al snel kwam men erachter dat dit niet erg handig is (bijvoorbeeld voor onderhoud en interoperabiliteit). De problematiek van het verzenden van gegevens over netwerken kan op verschillende abastractieniveau’s worden bekeken en opgelost:

1. Het via een kabel verzenden van bits
   Afspraken over el(ectrische verschijnselen, timings, weerstand.
2. Het verzenden van een pakketje informatie naar een naastliggende node (één hop verwijderd))
3. Het doorverzenden van pakketjes die niet voor deze node zijn bestemd
   (Routeren van pakketjes)
4. Het verzenden van een informatiestroom tussen twee applicaties
   (O.a. ervoor zorgen dat de pakketjes in de juiste volgorde aankomen)
5. Het maken van afspraken over de presentatie van de gegevens
   (Karaktercodering (ASCII, EBCDIC), encryptie, compressie).

Mits goed opgezet ontstaat op deze wijze een bouwwerk van oplossingen die gebaseerd zijn op de onderliggende verdiepingen. We noemen een dergelijk bouwwerk ook wel een *protocolstapel*.

## 6.6 Open System Interconnection



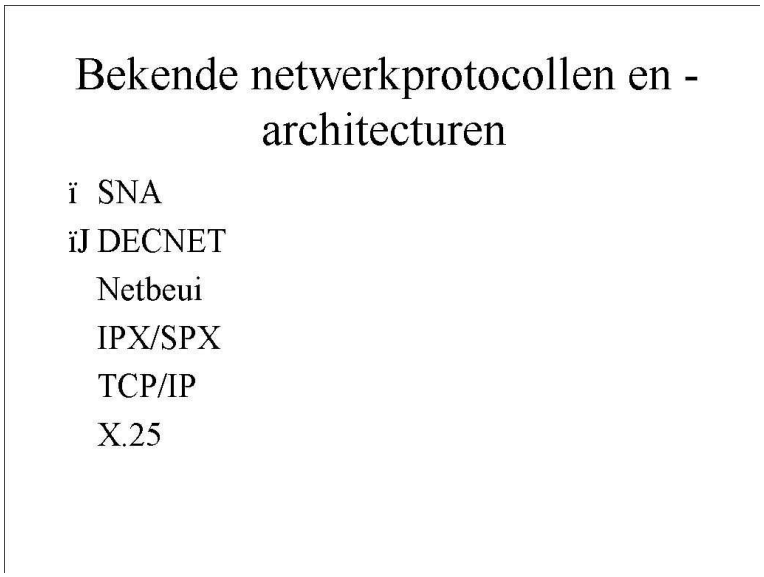

**Notities**

Gebaseerd op de ideeën van protocolstapels is er door de internationale standaardiseringsorganisatie ISO een raamwerk afgesproken voor het beoordelen van netwerkprotocollen. Dit raamwerk staat alom bekend als het OSI (Open Systems Interconnection) model. Dit OSI model is een formele weerslag van de niveaus waarop netwerkcommunicatie tussen twee applicaties betrekking heeft.

Voor het onthouden van de niveaus kan gebruik worden gemaakt van het ezelsbruggetje op de slide[1].

[1] Met dank aan Mike van Laar

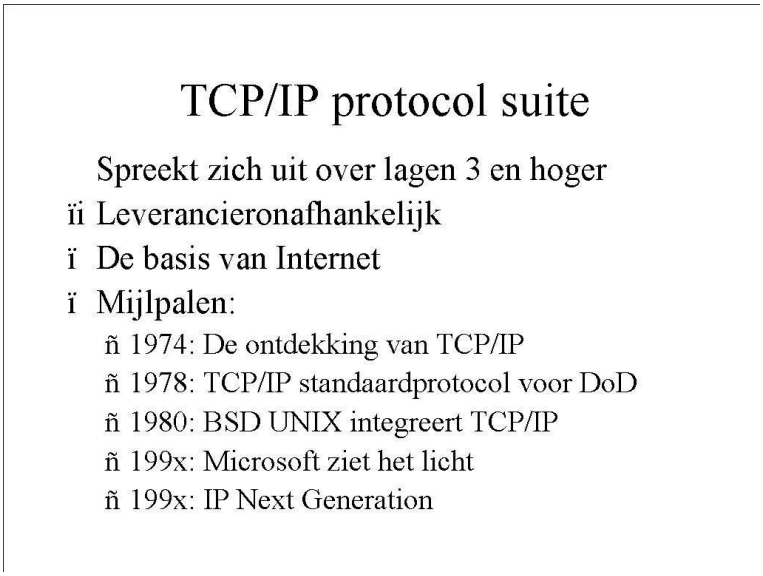
## 6.7 Bekende netwerkprotocollen en -architecturen




**Notities**

Door de jaren heen hebben een groot aantal leveranciers gewerkt aan het bieden van mogelijkheden voor het communiceren over netwerken. Het resultaat daarvan is een groot aantal onderling niet uitwisselbare netwerkprotocollen. Op het fysieke niveau is de standaardisatie een stuk beter gelukt. Transmissiemethoden als RS232, ethernet en token ring zijn wereldwijd gestandaardiseerd.

## 6.8 De TCP/IP protocol suite




**Notities**

Één van de meest populaire protocolstapels aller tijden wordt gevormd door de TCP/IP protocolsuite. Dit is een onderling samenhangende verzameling protocollen voor communicatie tussen applicaties over grote netwerken (Wide Are Networks). TCP/IP spreekt zich alleen uit over de lagen 3 en hoger van het OSI-model. Voor de data link en fysieke lagen wordt gebruik gemaakt van andere standaarden zoals RS232, ethernet, token ring en PPP. Om deze reden kan TCP/IP worden gedraaid over vrijwel elk denkbaar transmissiemedium.

Een erg in het oog springend feit is dat TCP/IP niet is ontwikkeld door een leverancier, maar door een research projectgroep. Dientengevolge is TCP/IP public domain en kan het door iedereen worden geımplementeerd zonder auteursrechtelijke of octrooi consequenties. Alle zichzelf respecterende leveranciers hebben TCP/IP daarom geımplementeerd en het is onder andere deze leverancieronafhankelijkheid die ervoor heeft gezorgd dat TCP/IP de basis vormt van het wereldwijde Internet.

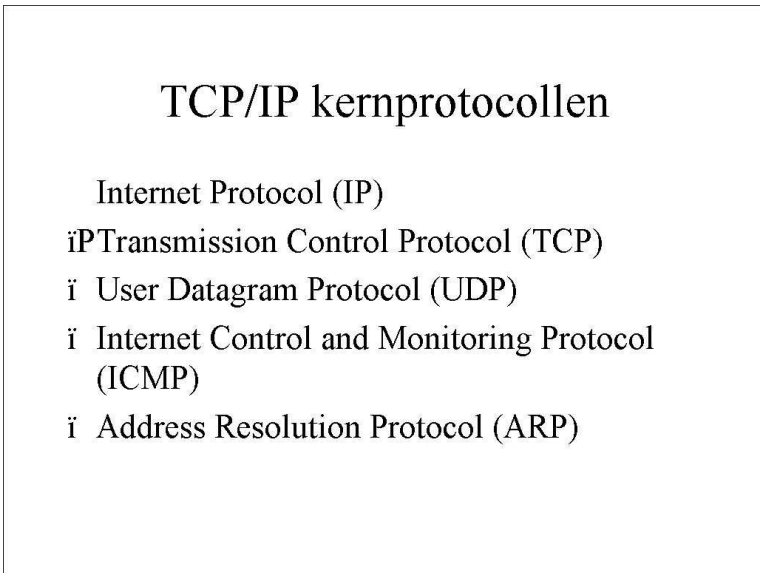
TCP/IP is omstreeks 1974 “ontdekt” door de Amerikanen Kahn en Cerf. Deze waren bezig met de ontwikkeling van een WAN-protocol wat gebruikt kon worden in het ARPANET onderzoeksnetwerk. De tot dan toe gebruikte ARPANET protocollen voldeden namelijk niet meer. TCP/IP werd een aantal malen geïmplementeerd en bleek al snel een goed werkend protocol te zijn. Het feit dat TCP/IP niet gebonden was aan één leverancier bracht het Amerikaanse ministerie van defensie (DoD) ertoe om in 1978 TCP/IP uit te roepen tot de datacommunicatiestandaard van de Amerikaanse overheid.

In een revolutionaire beweging besloot de universiteit van Berkeley om hun gratis versie van UNIX (BSD UNIX) standaard te voorzien van TCP/IP. Tot dan toe was het namelijk tamelijk ongebruikelijk om netwerksoftware te bundelen in een besturings-

systeem (is sindsdien erg gebruikelijk). Naast de kale TCP/IP communicatieprogrammatuur ontwikkelde de professoren, docenten en studenten van de Berkeley universiteit ook allerlei programmatuur die van TCP/IP gebruik maakte. Vrijwel al die programmatuur wordt in het hedendaagse Internet nog gebruikt!

BSD UNIX kon tegen een schappelijk tarief worden gekocht (source) en vond op die wijze zijn ingang op veel hogescholen en universiteiten wereldwijd. Ook het bedrijfsleven ging (onder aanvoering van de ex-studenten, nu manager) over op UNIX en TCP/IP. Anti-TCP/IP bolwerken als IBM (SNA) en DEC (DECNET) hebben het een tijd volgehouden maar moesten na verloop van tijd ook overstag. Zelfs Microsoft ontdekte de kracht, eenvoud en schoonheid van TCP/IP en implementeerde dit protocol in zijn besturingssystemen.

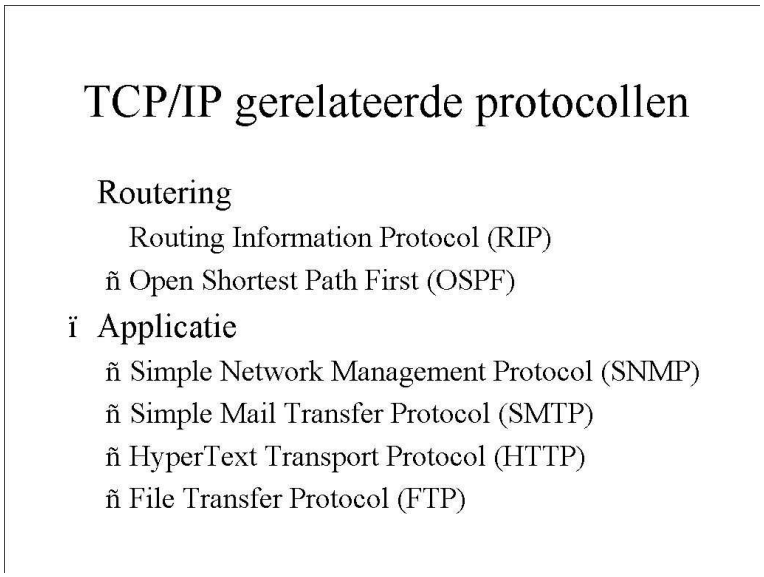
## 6.9 TCP/IP kernprotocollen




**Notities**

TCP/IP bestaat uit een groot aantal onderling samenhangende protocollen die ieder een gedeelte van de gehele netwerkproblematiek adresseren. De meest belangrijke protocollen die op iedere TCP/IP node zijn geımplementeerd zijn IP, TCP, UDP, ICMP en ARP.

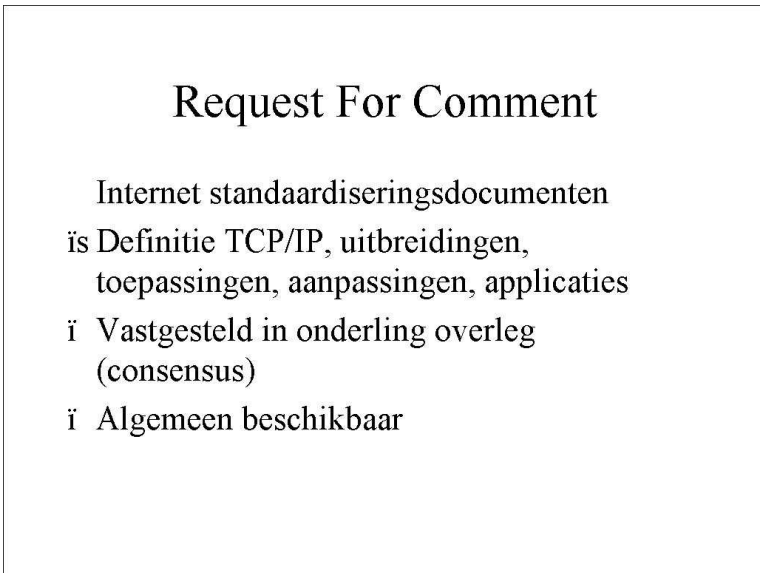
## 6.10 TCP/IP gerelateerde protocollen




**Notities**

Aan TCP/IP gerelateerde protocollen zijn applicatie of ondersteunende protocollen die extra functies bieden bij het opzetten of beheren van een TCP/IP netwerk. Op de slide staan een aantal van de meest bekende protocollen op dit gebied.

## 6.11 Request for Comment

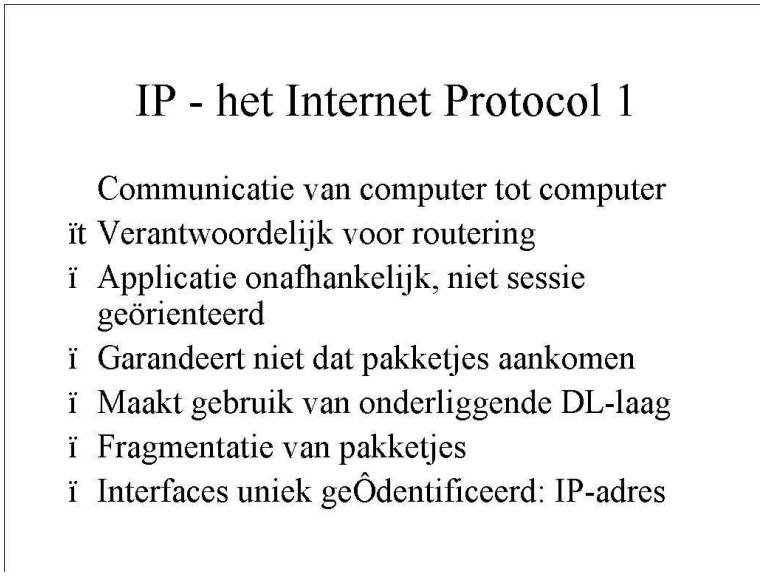



**Notities**

Omdat TCP/IP niet in handen is van een leverancier worden standaarden op het gebied van TCP/IP (en Internet) gezet door commissies van “vrijwilligers” onder aansturing van de Internet Architecture Board (IAB). Werkgroepen van specialisten stellen nieuwe standaarden op, verwerken commentaar en passen bestaande standaarden aan. Een nieuwe standaard wordt opgesteld als *Request for Comment*. Zolang de standaard nog de status *”Proposed standard”* heeft kan er door iedereen commentaar worden geleverd welke vervolgens in de RFC wordt verwerkt. Zodra de standaard door de IAB is geaccepteerd wordt een ieder geacht rekening te houden met de in de RFC vastgelegde wijsheden.

RFC’s zijn algemeen beschikbaar op informatie CD’s en op Internet FTP servers.

## 6.12 IP - Het Internet Protocol 1



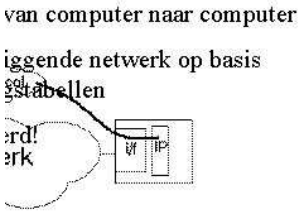

**Notities**

Het hart van TCP/IP wordt gevormd door het *Internet Protocol*. IP is verantwoordelijke voor de verzending van pakketjes door het netwerk. IP ontvangt deze pakketjes van de hoger gelegen lagen (4 en hoger) en stuurt ze via een bepaald netwerkinterface door naar de volgende node in de reis. Voor dit laatste maakt IP gebruik van netwerkdrivers die voldoen aan OSI laag 2 (data link)

Als IP een pakketje onvangt dan wordt gecontroleerd of dat pakketje bedoeld is voor deze node. Als dat zo is dan wordt dit pakketje doorgegeven aan de hoger gelegen netwerkprogrammatuur. Is het pakketje niet bestemd voor de huidige node dan bekijkt IP de eindbestemming en hij bedenkt waar hij dit pakketje vervolgens eens naar toe zal sturen (volgende hop). Één van IP's belangrijkste taken is dus de routering van pakketjes door het netwerk. Een pakketje hopt van node tot node door het netwerk totdat het op zijn eindbestemming is. Iedere TCP/IP node doet aan deze routeringstaak mee! De in grotere netwerken gebruikelijke routers zijn speciale computers waarin in principe alleen maar IP-software draait.

IP weet niet waarom de pakketjes worden vestuurd. Het protocol kent geen sessies of applicaties. Ieder pakketje staat op zich en wordt onafhankelijk van alle andere pakketjes verstuurd. Als een IP-pakketje zoekraakt (omdat bijvoorbeeld een router uitvalt terwijl die het pakketje aan het verwerken is) dan is dat pech. De hoger gelegen lagen moeten dit oplossen.

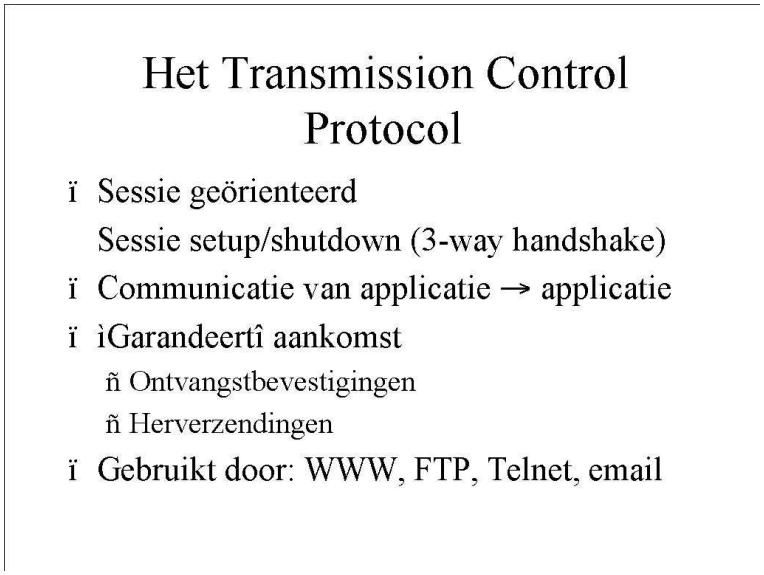
## 6.13 IP - Het Internet Protocol 2




**Notities**

Op deze slide staat met een plaatje aangegeven wat de taak van het Internet Protocol is. Zodra een pakketje op de gewenste bestemming is aangekomen wordt het daar verder verwerkt door de hoger gelegen netwerkprogrammatuur.

## 6.14 Het Transmission Control Protocol




**Notities**

IP lost een aantal belangrijke problemen in het verzenden van informatie voor ons op, maar laat evengoed een aantal andere taken liggen. Zo kunnen IP pakketjes zoekraken, in de verkeerde volgorde binnenkomen en weet IP niet voor welke applicatie in de doelcomputer een pakketje bestemd is. Het Transmission Control Protocol (TCP) lost deze problemen wel op.

TCP ondersteunt *sessies* tussen twee applicaties. Applicaties openen een TCP verbinding en gebruiken TCP-functies om informatie te verzenden en te ontvangen. De TCP laag is er vervolgens verantwoordelijk voor dat de informatie in de juiste volgorde en met de juiste inhoud aankomt. Vrijwel alle sessie geörienteerde TCP/IP applicaties (zoals email, World Wide Web, FTP) gebruiken TCP. De programmeur kan erop vertrouwen dat als hij een pakketje data aan TCP heeft aangeboden dat TCP er zorg voor draagt dat die informatie ter bestemde plekke wordt afgeleverd.

Het is dus TCP zijn taak om een betrouwbare sessieverbinding op te zetten over een onbetrouwbaar medium (IP). Om dit voor elkaar te boksen nummert TCP alle uitgaande berichten. De ontvangende TCP stuurt bevestingen van alle ontvangen pakketten. Stel nu dat TCP de pakketjes 1, 2 en 3 verstuurt. Na verloop van tijd ontvangt hij van zijn partner aan de andere kant van het netwerk de bevestiging dat de pakketjes 1 en 2 zijn ontvangen. TCP gaat er dan vanuit dan pakketje 3 zoek is geraakt en verzend dat nog een keer. Stel nu dat de verzending van pakketje 3 nu wel goed gaat. De ontvangende TCP zendt een bevestiging voor pakketje 3 naar de zender. Ook zo'n bevestiging kan echter zoekraken. In dat geval gaat de zender (time out) ervan uit dat pakketje 3 nog steeds niet is aangekomen, en zendt het dus nog een keer. De ontvanger ziet nu voor de tweede keer pakketje 3 binnenkomen, gooit het pakketje weg en stuurt wederom een bevestiging. Op deze wijze kan TCP een betrouwbaar pad tussen twee applicaties

opzetten.

## 6.15 Het User Datagram Protocol

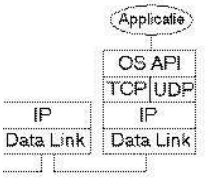

**Notities**

Een bekend nadeel van TCP is dat het protocol relatief veel overhead met zich meebrengt als de hoeveelheid te verzenden data gering is. Bij het opzetten en sluiten van een sessie wisselen de TCP-partners wat informatie met elkaar uit over volgnummers en het aantal pakketjes wat mag worden verzonden zonder eerst op een bevestiging te wachten (de zogenaamde *window size*.

Als we een applicatie hebben die slechts één bericht wenst te verzenden zonder antwoord van de ontvanger dan is de relatieve overhead van het opzetten en sluiten van de sessie zeer groot. Een goed voorbeeld hiervan is het HTTP protocol (World Wide Web) wat TCP verbindingen gebruikt voor het ophalen van HTML pagina's.

Speciaal voor dit soort applicaties is er het User Datagram Protocol. Dit protocol maakt telegramachtige gegevensuitwisselingen tussen applicaties mogelijk. Met UDP functies kunnen applicaties een datagram sturen naar een andere applicatie in het netwerk. UDP verpakt het datagram in één of meer IP-pakketjes en stuurt deze (via IP) naar de doelcomputer. Aangezien UDP gebruik maakt van IP zonder extra maatregelen te treffen kunnen UDP datagrammen zoekraken. Een applicatie is in dat geval genoodzaakt om zelf zorg te dragen voor herverzendingen en aanverwante verschijnselen. Het grote voordeel van UDP is de lage overhead en efficiënt gebruik van de beschikbare bandbreedte.

## 6.16 De informatiestroom




Xb∞u∞vùbaùâav∞∞∞

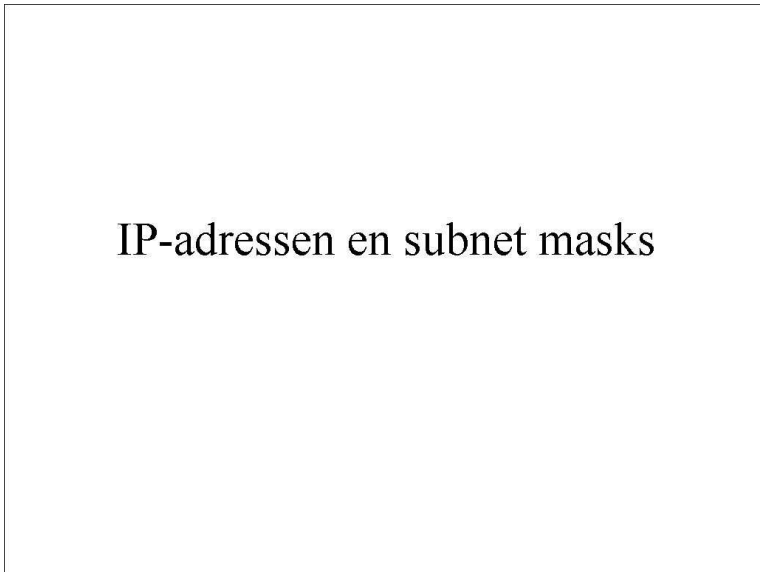
**Notities**

Op de slide is weergegeven hoe informatie van applicatie tot applicatie stroomt via de TCP/UDP/IP protocollen. Bedenk dat iedere computer in een TCP/IP netwerk in principe ook kan functioneren als router. Met name computers met meer dan één interface lenen zich goed voor deze functie. TCP/IP routers kunnen dus ook “vertalen” van het ene data link medium naar het andere. Veel routers hebben meerdere verschillende soorten interfaces. TCP/IP merkt hier verder niets van. Ieder data link medium is voor IP gelijk.

# Hoofdstuk 7

# IP-adressen en subnet masks

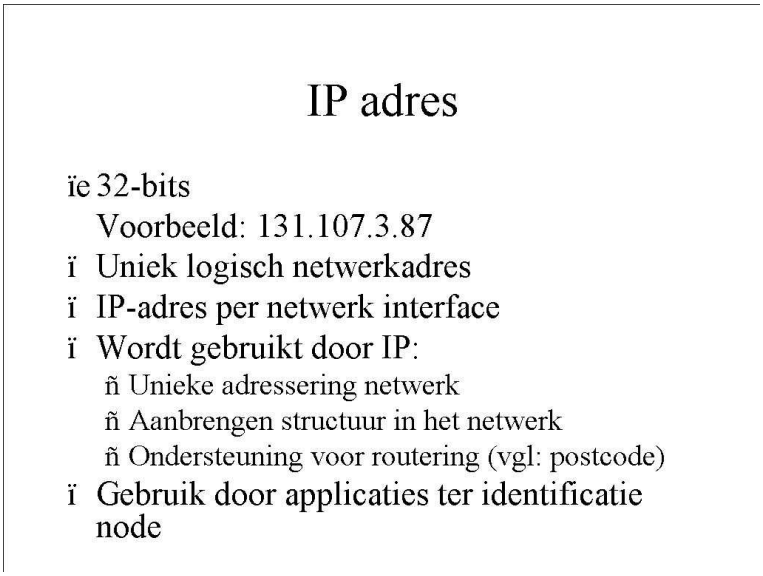
## 7.1 IP-adressen en subnet masks




**Notities**

In deze module gaan we dieper in op de adressering en naamgeving van nodes in een TCP/IP netwerk.

## 7.2 IP adres




**Notities**

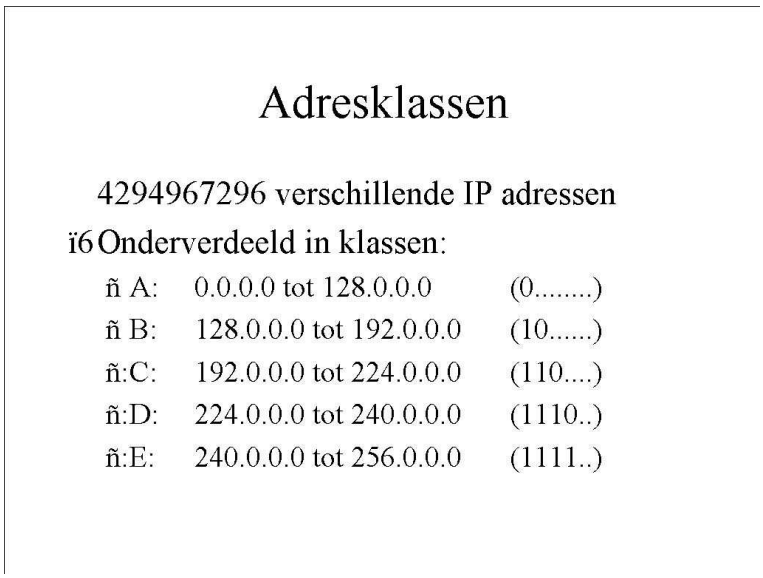
Ieder interface in een TCP/IP netwerk dient te zijn voorzien van een uniek IP-adres. Zo'n adres is een 32-bits logisch netwerknummer wat wordt toegekend door de netwerkbeheerder. Het "logische" aspect aan het IP-adres is dat het toegekende nummer niet is verbonden met het soort of type netwerkinterface aan wie het is toegekend. Een IP-adres dient in het gehele TCP/IP netwerk uniek te zijn. In Internet omgevingen houdt dit in dat toegekende IP-adressen wereldwijd uniek moeten zijn!

Uit notatieredenen schrijven we een IP-adres meestal als een combinatie van vier getallen van nul tot en met tweehonderdvijfenvijftig. Voor de wiskundigen onder ons: $256^4 = (2^8)^4 = 2^{(8*4)} = 2^{32}$.

Theoretisch kunnen er in een TCP/IP netwerk dus iets meer dan vier miljard verschillende interfaces worden opgenomen. Strikt gesproken (pyramide van Maslov) zou dit meer dan genoeg moeten zijn voor de gehele wereld. Het aantal IP-adressen begint echter al behoorlijk op te raken. De volgende generatie van IP, IP next generation, lost dit op door langere IP adressen. Om het gebruik van IP-adressen wereldwijd te reguleren is er een instantie die IP-adressen centraal registreert en uitgeeft: het InterNIC. Netwerken die zijn aangesloten op Internet mogen alleen geregistreerde IP-adressen toekennen.

IP gebruikt IP-adressen voor de interne adressering in het netwerk. Op basis van het IP-adres beslist IP of er gerouteerd moet worden (zie ook subnetting). Als er per ongeluk dubbele IP-adressen worden uitgedeeld ontstaan er netwerkproblemen in de communicatie van, naar en tussen de nodes met het conflicterende adres.

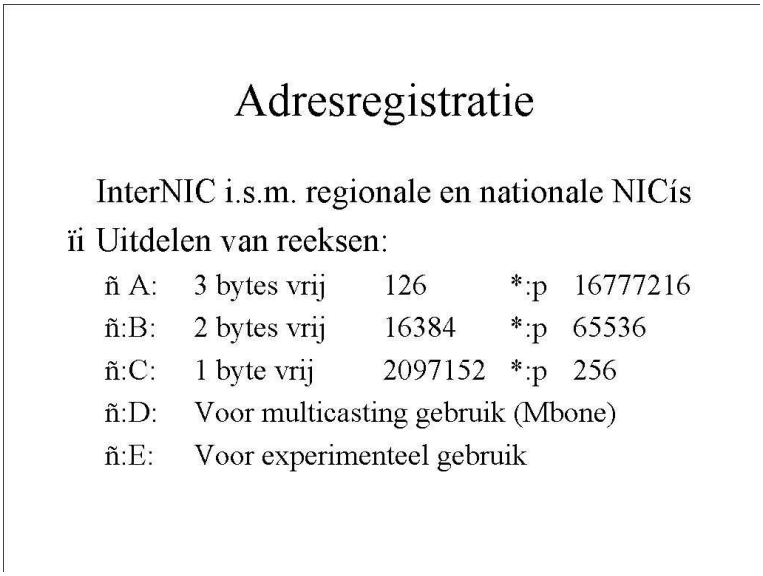
## 7.3 Adresklassen




**Notities**

Ten behoeve van het uitdelen van IP-adressen hebben de TCP/IP wijsgeren van het eerste uur besloten de gehele reeks van IP-adressen te splitsen in vijf deelreeksen (klassen).

## 7.4 Adresregistratie




**Notities**

Zoals reeds eerder gesteld verzorgt het InterNIC de registratie van IP-adressen. Organisaties die op Internet willen aansluiten dienen bij het InterNIC (of één van haar regionale zusters) een reeks van IP-adressen aan te vragen. Hoe groot de toegewezen reeks is hangt af van de grootte van het aan te sluiten netwerk en de verwachte groei over de komende jaren.

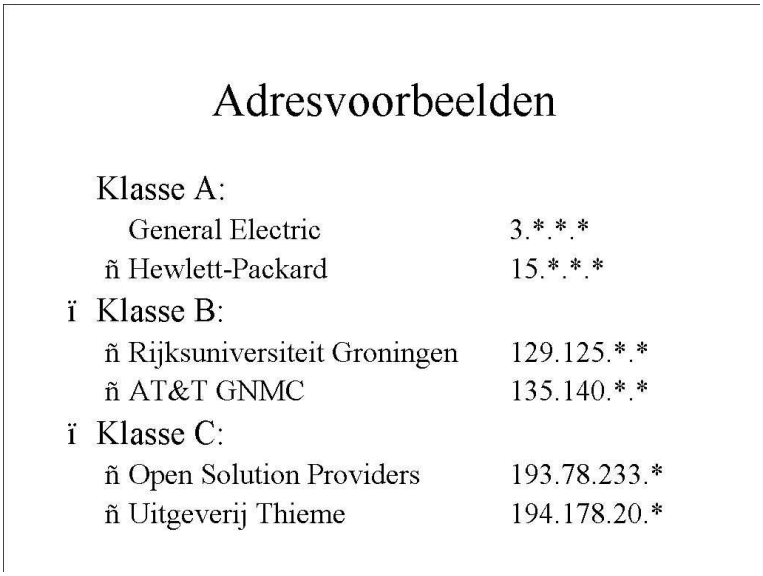
De allergrootste organisaties hebben een klasse A reeks toegewezen gekregen. Zo'n A-adresreeks heeft één byte voorgeschreven door het InterNIC en drie bytes die vrij mogen worden ingevuld. Hierdoor kunnen in zo'n netwerk zestien miljoen verschillende IP-adressen worden uitgedeeld. Wat kleinere organisaties krijgen een klasse B adresreeks toegewezen. Deze krijgen dan twee bytes voorgeschreven en mogen twee bytes zelf invullen. In deze netwerken zijn theoretisch 65536 verschillende IP-adressen te verzinnen. De kleinste netwerkjes krijgen klasse C adressen toegekend. Één byte mag worden gekozen en er worden er drie voorgeschreven. In een klasse C netwerk kunnen dus maximaal 256 verschillende IP-adressen worden bedacht. Organisaties die te groot zijn voor een klasse C reeks, maar te klein voor een klasse B, krijgen een aantal aaneengesloten klasse C adresreeksen toebedeeld.

Alle klasse A adressen zijn al vergeven. Van klasse B zijn er nog wel wat over, maar je moet van goede huize komen om er nog eentje toegewezen te krijgen. Klasse C reeksen zijn nog volop voorhande, maar het einde is niet meer onzichtbaar!

Alhoewel vier miljard+ IP-adressen op het eerste gezicht ruimschoots afdoende lijkt gaan er nogal wat IP-adressen verloren door technische oorzaken. Zo worden de klassen D en E voor andere doeleinden gebruikt. Verder mogen IP-adressen met allemaal '0'en of '1'en in het subnet of host gedeelte niet worden gebruikt. Verder moeten voor

seriële segmenten met maar twee nodes (aan de uiteinden) vaak toch een IP-adres worden toegekend. Adresschattingen van Christian Huitema doen vermoeden dat er tussen de $3 * 10^4$ en $2 * 10^8$ uitdeelbare adressen in de IP-adresreeks zitten. zijn gebleven.

## 7.5 Adresvoorbeelden




**Notities**

De uitgedeelde adressen worden bijgehouden door InterNIC en haar regionale zusters in publiek toegankelijke databases[1]. Op de slide staan voorbeelden van toegekende adresreeksen.

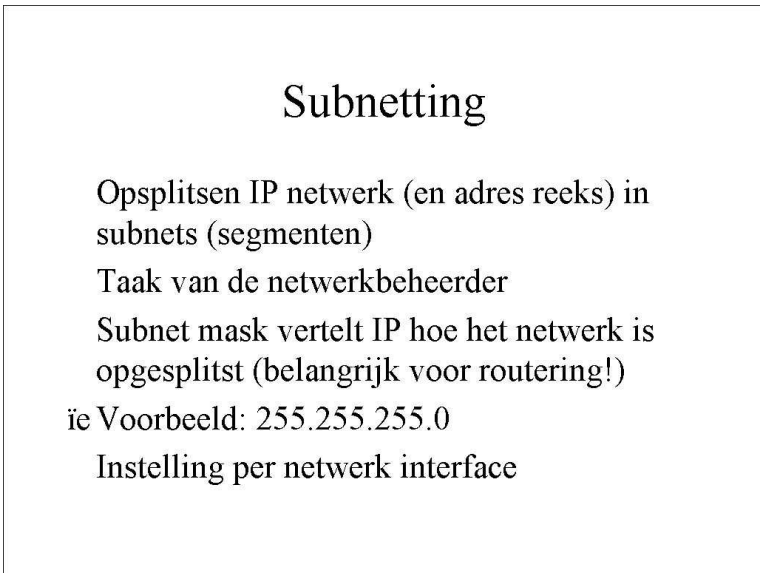
[1] telnet internic.net en telnet info.ripe.net

## 7.6 Configuratie IP-adres


**Notities**

Één van de beheertechnisch belangrijke aspecten van TCP/IP is dat iedere node in het net moet zijn geconfigureerd met een uniek, voor dat subnet van toepassing zijnd, IP-adres. Iedere PC, printer en X-terminal moet op die wijze van een IP-adres zijn voorzien. In de hedendaagse dynamische PC-netwerken levert dit aanzienlijke bezwaren op. Het beheer van uitstaande IP-adressen levert menige systeembeheerder aanzienlijke migraine aanvallen op. Microsoft heeft hierop geantwoord met DHCP, een protocol voor het automatisch toekennen en vrijgeven van IP-adressen.

## 7.7 Subnetting



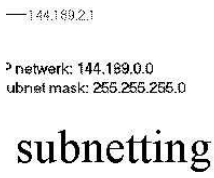

**Notities**

TCP/IP deelnetwerken moeten vaak ook weer worden opgesplitst in meerdere subnets. De toegewezen adresreeks moet dan weer verder worden opgesplitst om toekenning van subnet gebonden IP-adressen mogelijk te maken. De TCP/IP programmatuur in iedere node moet weten hoe het netwerk is opgesplitst in subnets. Deze kennis is van belang om te kunnen vaststellen of een andere node in hetzelfde segment (subnet) zit of dat er via TCP/IP gateways moet worden gerouteerd.

We maken de opsplitsing van het netwerk in subnets bekend door de specificatie van een *subnet mask*. Dit is een bitpatroon wat vertelt welke bits van het IP-adres behoren bij het net+subnet (de 1 bits in het masker), en welke bits horen bij het hostnummer binnen het subnet (de 0 bits). Een subnet mask van 255.255.255.0 vertelt TCP/IP dat de eerste drie bytes van het IP-adres hetzelfde zijn voor alle nodes in het lokale segment (subnet). Zodra TCP/IP zich voor de taak gesteld ziet om een pakketje te sturen naar een node met een afwijkend net+subnet id (eerste drie bytes in dit geval) dan moet het pakketje via een router reizen.

Net als het IP-adres dient het subnet mask in iedere node in een TCP/IP netwerk te worden ingesteld. Vergissingen zijn hier niet van de lucht en ook het configureren van subnet masks levert netwerkbeheerders bij tijd en wijle stevige hoofdpijnen op.

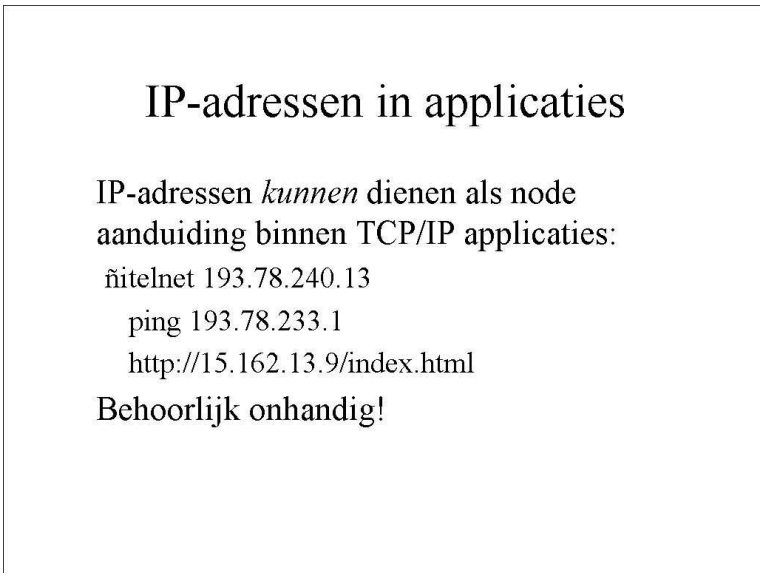
## 7.8 Voorbeeld subnetting




**Notities**

Het hele idee van IP-adressering en subnetting is wat beter te behappen met een ter zake doende voorbeeld. Zie hiervoor het netwerkdiagram op de slide.

## 7.9 IP-adressen in applicaties

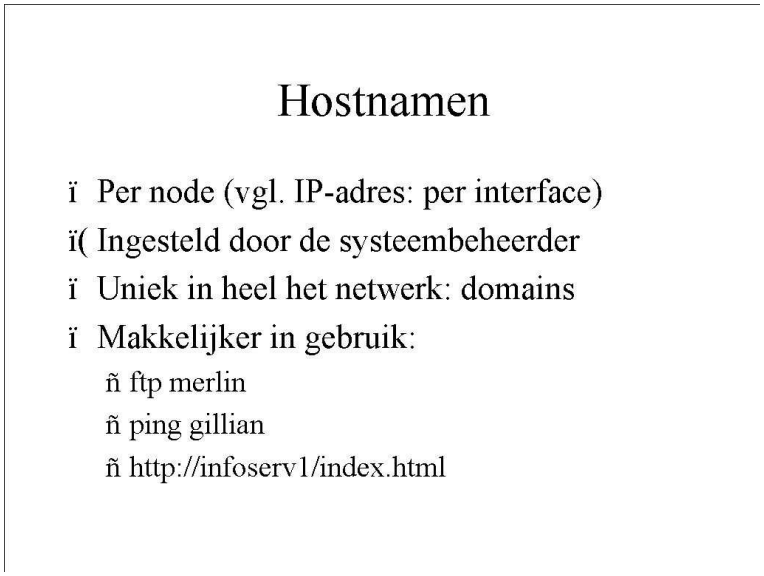



**Notities**

Omdat iedere node in een TCP/IP netwerk één of meer netwerkinterfaces heeft en aan een netwerkinterface één of meer unieke IP-adressen zijn gekoppeld kan een IP-adres worden gebruikt als een identificatie van een node in een TCP/IP netwerk. Om een aantal redenen verdient dit echter niet de voorkeur:

- IP-adressen zijn moeilijk te onthouden, het betreft tamelijk cryptische getallen die geen relatie hebben tot de functie van een node.
- IP-adressen kunnen wijzigen indien het deelnetwerk waarin nodes staan worden omgenummerd of opnieuw ingedeeld.

## 7.10 Hostnamen




**Notities**

De oplossing voor het op de vorige slide gepresenteerde probleem wordt geboden door het feit dat iedere node in een TCP/IP netwerk een eigen, symbolische, hostnaam heeft. De hostnaam van een systeem wordt geconfigureerd door de systeembeheerder op een voor het systeem toepasselijk wijze. Voor een UNIX systeem houdt dit in dat één of ander configuratiebestand moet worden gewijzigd, terwijl op een Windows 95 machine dit via het control panel moet worden gewijzigd.

Alle TCP/IP applicaties kunnen ook worden geparametriseerd met een hostnaam. Dus in plaats van:

```
telnet 193.78.240.13
```

kunnen we ook zeggen:

```
telnet dbserver
```

Dit laatste is natuurlijk veel eenvoudiger te onthouden. Ondanks dat we een commando parametriseren met een hostnaam werkt de TCP/IP software intern met IP-adressen. Dit betekent dat de applicatie de hostnaam moet vertalen naar een IP-adres alvorens een verbinding kan worden opgezet. Zie voor meer informatie hierover het hoofdstuk “Hostnaam resolutie” op pagina 8.

Hostnamen dienen net als IP-adressen in het hele netwerk uniek te zijn. Binnen een deelnet is dat meestal geen probleem; de systeem- en netwerkbeheerders van de organisatie hebben de volledige controle over de naamgeving van de hosts en kunnen deze dus uniek houden. In een wereldwijd gedistribueerd netwerk als Internet is dit

natuurlijk een groter probleem. voor de hand liggende namen als ‘server’, ‘gateway’ en ‘lp’ komen natuurlijk zonder problemen in meerdere netwerken voor. De oplossing voor dit probleem wordt gevormd door hostnamen te voorzien van extra componenten waardoor ze uniek worden gemaakt. We noemen deze extra componenten de *domain names*.

## 7.11 Domain Names

### Domain names

Hierarchische Internet naamgevings standaard

- Iedere node heeft een Fully Qualified Domain Name (FQDN)
- FQDN bestaat uit drie of meer componenten, gescheiden door ë.í
- FQDN eindigt op toegekende domain naam
- Domain registratie door InterNIC en nationale domain registries

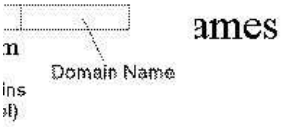
**Notities**

Om de uniciteit van hostnamen in het Internet te bevorderen[2] is afgesproken dat de volledige naam (de Fully Qualified Domain Name) van een node eindigt op een door de organisatie geregistreerde unieke domain naam. Doordat die domain namen wereldwijd worden uitgegeven en geregistreerd kan daardoor nooit een dubbele hostnaam voorkomen (tenzij de systeembeheerders hun werk niet goed doen).

Voor bijvoorbeeld de ANWB houdt dit in dat alle hostnamen in hun netwerken in principe eindigen op '.anwb.nl'. Aangezien geen enkele andere aangesloten organisatie die domain naam mag gebruiken is de hostnaam 'gateway.anwb.nl' wereldwijd uniek.

[2] Understatement is a form of irony in which you say *less* then you mean

## 7.12 Fully Qualified Domain Names




**Notities**

De volledig gekwalificeerde naam van een node bestaat uit *n* componenten, onderling gescheiden door een '.'. Een dergelijke naam bestaat minimaal uit drie componenten:

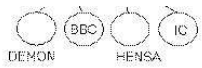
1. Simpele hostnaam (b.v. gateway)
2. Organisatie identificatie (b.v. anwb)
3. Top level domain (b.v. nl)

Organisaties mogen in principe tussen de simpele hostnaam en het verplichte gedeelte nog één of meer subdomain identificaties plaatsen. Deze dragen bij aan de FQDN. Bijvoorbeeld: 'gateway.zeeland.kantoren.anwb.nl'. Sommige landen hebben een verplicht gedeelte van drie niveaus. Bijvoorbeeld in het Verenigd Koninkrijk krijgen bedrijven een naam toegewezen die eindigt op '.co.uk'. Hierdoor is aan de FQDN te zien met wat voor soort organisatie we te maken hebben.

Domain namen worden wereldwijd gecoördineerd met het InterNIC. Per land is er vervolgens een instituut die de uitgifte voor dat land regelt. In Nederland kennen we de Stichting Domein Registratie (zie http://www.nl.net) die de uitgifte heeft uitbesteed aan KEMA.

Een spannend verschijnsel bij de uitgifte van domain namen is dat een gegeven naam maar één keer kan worden uitgegeven. In het handelsrecht van Nederland is het zo dat een bepaalde naam best twee keer mag worden gebruikt, mits de bedrijven in geheel verschillende branches opereren. Bij domain naamgeving geldt een dergelijke mogelijkheid niet. Meestal geldt hiervoor de regel: 'Wie het eerst komt, die het eerst maalt'. Verder hebben er met name in de Verenigde Staten nog wel zaken gespeeld als het registreren van merk- en handelsnamen en begrippen (www.toothpaste.com), het registreren van de handelsnaam van concurrenten en het registreren van namen van bedrijven die zo dom waren geweest zich nog niet te registreren, om vervolgens de naam door te kunnen verkopen.

## 7.13 Domain name hierarchie




ıe hierarchie

**Notities**

In de domain naamgeving die op Internet wordt gehanteerd is een strikte hierarchie aanwezig. De top level domains (laatste component in de FQDN) zijn eenduidig bepaald. Onder de top level domains hangen de organisaties die door de landelijke domain registries worden uitgegeven, weer daarondern hangen de subdomains en nodes die door de netwerkbeheerders worden bepaald.

Binnen de Verenigde Staten kunnen domains worden geregistreerd die als laatste component en aanduiding hebben van het soort organisatie:

- COM, bedrijven
- EDU, scholen en universiteiten
- ORG, stichtingen en verenigingen
- GOV, overheid
- MIL, leger
- NET, Internet gerelateerde bedrijven en instellingen

Aangesloten organisaties in andere landen krijgen een domain naam die eindigt op de ISO landcode:

- NL, Nederland
- ES, Spanje
- AU, Australië
- CH, Zwitserland
- AT, Oostenrijk
- ...

In tegenstelling tot wat veel mensen denken is er ook een .US domein. Dit wordt gebruikt door hele kleine organisaties en particulieren die een domein willen registreren. Het .US domein is geografisch ingedeeld; bijvoorbeeld '.SF.CA.US' (San Francisco, Californië).

Recentelijk zijn er enkele bewegingen op gang gekomen om het aantal toplevel domains uit te breiden met qualifiers als ".sex", ".recën ".humor". Dit vereist echter wereldwijde coördinatie omdat alle root domain name servers de nieuwe domains moeten erkennen.

## 7.14 Aliassen

Aliassen

Mogelijkheid om meerdere hostnamen aan een IP-adres te koppelen.

gatekeeper.osp.nl
www.osp.nl
ns.osp.nl
...

Vereenvoudigt later splitsen van servers

**Notities**

De diverse methoden voor hostnaam resolutie (zie volgende hoofdstuk) ondersteunen allemaal de mogelijkheid van het opnemen van één of meerdere aliassen voor een host. Zo is de machine ‘gatekeeper.osp.nl’ ook te benaderen als:

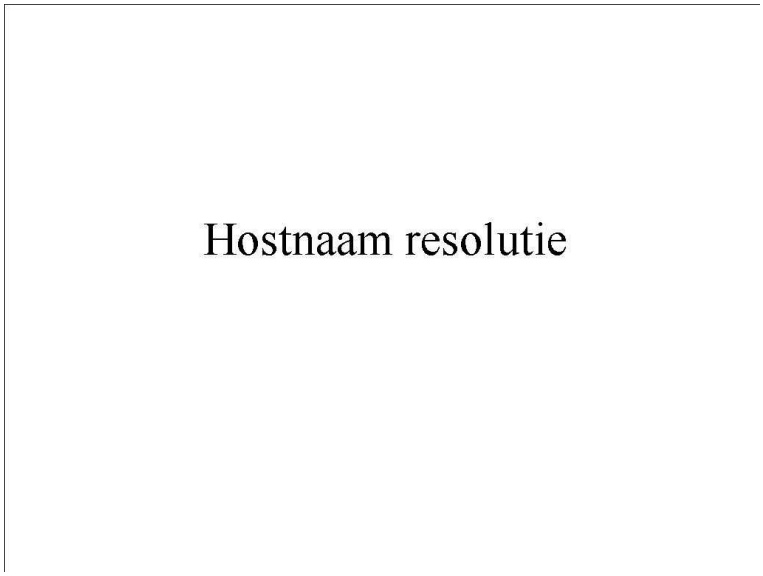
- www.osp.nl
- ftp.osp.nl
- mail.osp.nl
- ns.osp.nl
- news.osp.nl
- osp.nl

In dit geval wordt simpelweg hetzelfde IP-adres aan meerdere hostnamen gekoppeld. Hiervoor wordt vaak gekozen omdat het in Internet gebruikelijk is geworden om de hostnaam van een machine gedeeltelijk af te laten hangen van de service die de machine verleent. Web servers zijn bijvoorbeeld vrijwel altijd bekend als ‘www.nogwat’, terwijl FTP servers vaak ‘ftp.nogwat’ heten. Als één machine meerdere functies verricht moet hij conform deze “standaard” ook onder meerdere namen bereikbaar zijn.

# Hoofdstuk 8

# Hostnaam resolutie

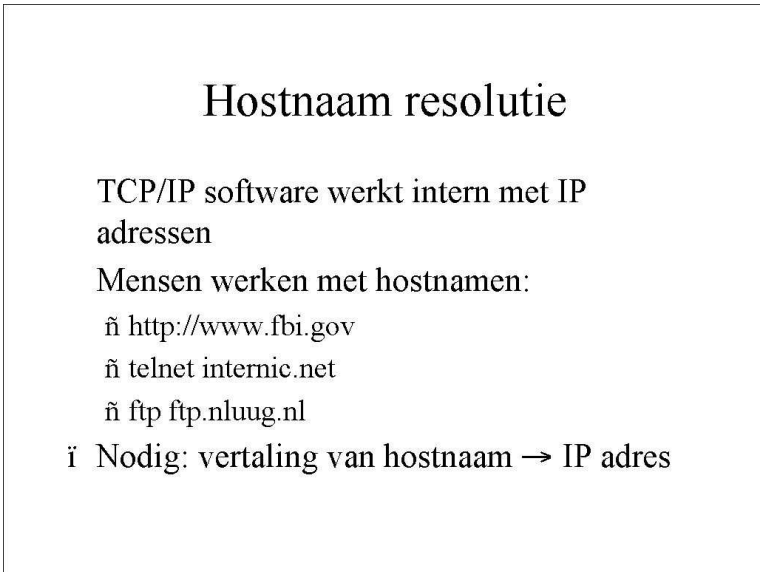
## 8.1 Hostnaam resolutie




**Notities**

Welkom bij het hoofdstuk over hostnaam resolutie. In dit hoofdstuk behandelen we de meest voor de hand liggende manieren waarop hostnamen kunnen worden vertaald naar IP-adressen en omgekeerd.

## 8.2 Hostnaam resolutie

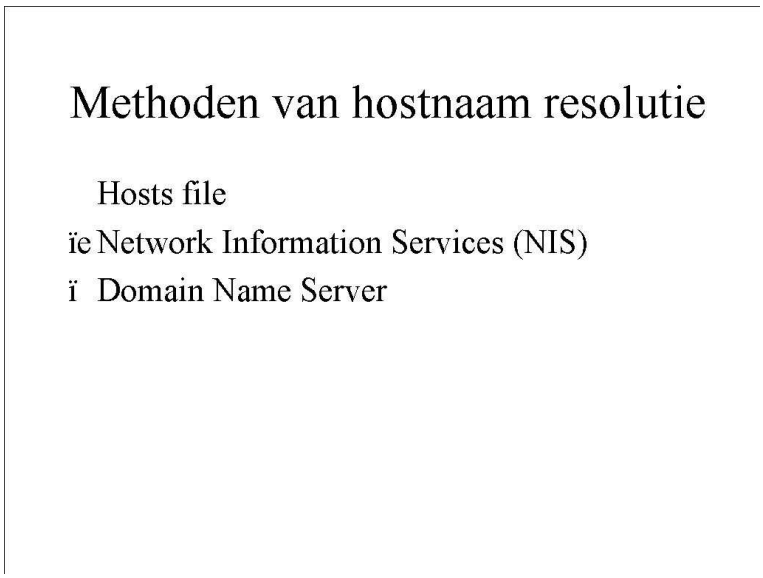



**Notities**

Zoals reeds eerder in de shop aan de orde is gekomen werkt de TCP/IP programmatuur intern met IP-adressen, terwijl applicaties en mensen liever werken met hostnamen. Bij het opzetten van een verbinding moet een applicatie echter een IP-adres specificeren en als een applicatie informatie opvraagt over een verbinding dan krijgt deze van de TCP/IP laag weer IP-adressen terug. Er is dus behoefte aan een vertaalmechanisme van hostnamen naar IP-adressen en vice versa.

Technisch gesproken wordt deze vertaling uitgevoerd door de ‘gethostbyname()’ en ‘gethostbyaddr()’ calls. Deze UNIX en Windows systeemaanroepen voeren de gewenste vertalingen uit.

## 8.3 Methoden van hostnaam resolutie




**Notities**

In principe is iedere leverancier vrij om een mechanisme te verzinnen waarmee de vertaling kan worden uitgevoerd. Historisch en praktisch gezien worden er echter meestal één of meer van de volgende drie methoden toegepast:

1. De hosts file
2. NIS
3. De Domain Name Server

Een aantal leveranciers (zoals Microsoft) hebben hiernaast nog hun eigen methoden. Zo kan een Windows machine ook nog gebruik maken van Netbios Name Resolution mechanisme zoals local broadcast, de LMHOSTS file en de NetBios Name Server (WINS).

## 8.4 Hosts file


**Notities**

De hosts file is een tekst bestand met daarin regels met op iedere regel (afgezien van lege regels en commentaar) een IP-adres en één of meer hostnamen. Het bestand heeft zijn historische oorsprong in de file 'HOSTS.TXT'. Deze file bevatte in de vroege dagen van Internet een opsomming van alle nodes in het net met hun netwerkadressen. Vanzelfsprekend is een dergelijke aanpak vandaag aan de dag niet langer haalbaar.

De naam en plaats van de hosts file wordt door de leverancier bepaald. In UNIX systemen staat de file altijd in de /etc directory. Onderhoud op de hosts file geschiedt met een gewone ASCII editor[1]. Het grote voordeel van de hosts file is de ongekende simpelheid. Om de hosts file te gebruiken hoeft niets te worden geconfigureerd. Het onderhoud van de file is op zich redelijk simpel. De hosts file kent echter ook belangrijke nadelen:

- De file moet op iedere node in het netwerk aanwezig zijn. Dit kan leiden tot grote problemen in het distribueren van de correcte file. Met name in omgevingen waar regelmatig wijzigingen in de hosts file nodig blijk zwerven al gauw verschillende versies van de hosts file door het netwerk.
- De koppeling tussen IP-adressen en hostnamen is 1:n, terwijl de eigenlijke koppeling n:1 of n:m is. In het geval een node meerdere interfaces, en dus meerdere IP-adressen, heeft kan toch slechts één IP-adres aan een hostnaam worden gekoppeld (zie voorbeeld volgende slide).
- De hosts file kent geen faciliteiten voor het opslaan van andere informatie zoals email gateways.

[1] vi rules!

## 8.5 Voorbeeld hosts file

### Voorbeeld hosts file

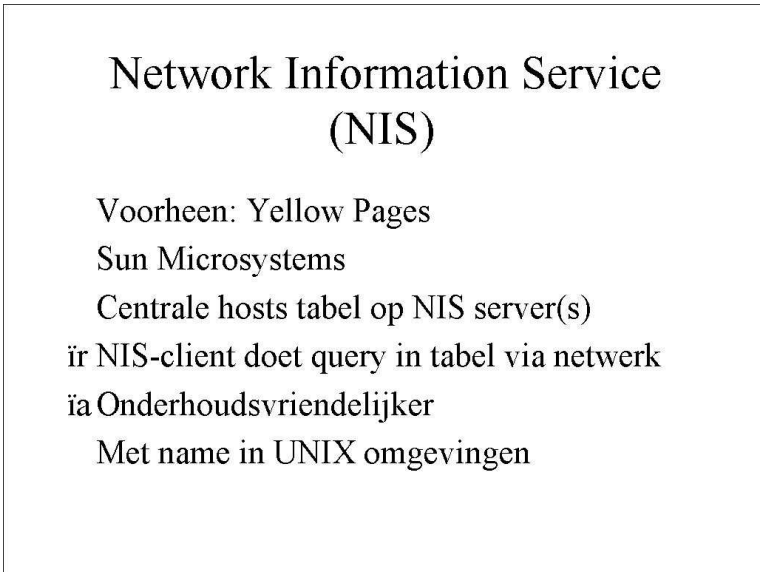
```
# Demo hosts file
# Laatste wijziging: 4 december 1996 door Jos Visser

127.0.0.1        localhost loopback
144.189.1.2      server1 server1.myorg.nl
144.189.1.2      router1 router1_eth
144.189.4.1      router1_tr
144.189.3.2      router1_ser
193.78.240.13    solair1.inter.nl.net
```

**Notities**

Op de slide staat een voorbeeld van een hosts file. Zoals te zien is betreft het een vrij recht-toe-recht-aan tekstbestand waarin de koppelingen tussen hostnamen en IP-adressen wordt gelegd.

Nadere analyse van de hosts file leidt ons tot de gedachte dat er een router is met de naam ‘router1’ met drie verschillende netwerk interfaces. Deze interfaces hebben de IP-adressen 144.189.1.2, 144.189.4.1 en 144.189.3.2. Het hosts file mechanisme staat ons niet toe om aan één hostnaam meerdere IP-adressen te hangen. We zijn dus genoodzaakt om aliassen op te nemen voor de verschillende netwerk interfaces: ‘router1_eth’, ‘router1_tr’ en ‘router1_ser’. Een probleem daarbij is dat applicaties die met een bepaalde hostnaam werken (bijvoorbeeld ‘router1’) altijd met hetzelfde IP-adres (netwerkinterface) praten. Afhankelijk van hoe intelligent de routering is opgezet vormt dit geen, een klein of een groot probleem.

## 8.6 Network Information Service



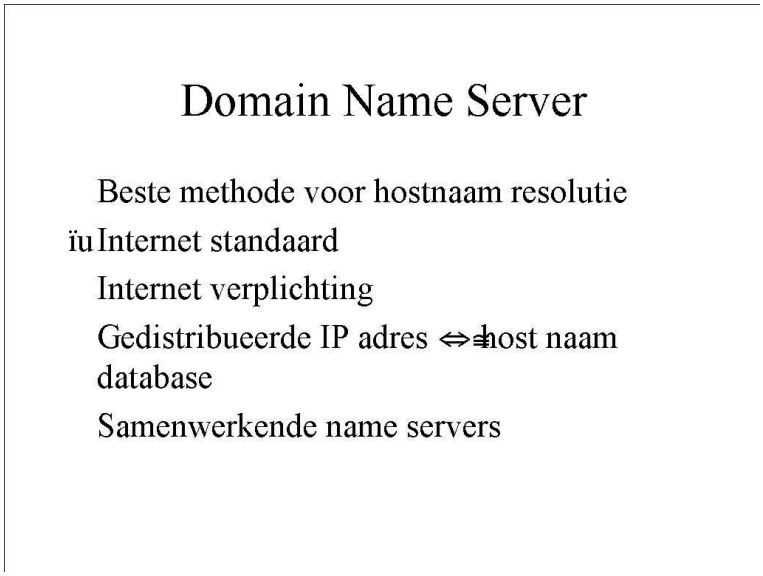

**Notities**

De tweede mogelijkheid voor het vertalen van hostnamen naar IP-adressen wordt gevormd door de Network Information Service (NIS). Dit is een door Sun Microsystems bedacht subsysteem wat in staat is om systeemconfiguratietabellen centraal op een netwerkserver te beleggen.

Werken met NIS is conceptueel gelijk aan het werken met een hosts file. Alleen ligt de hosts file nu niet op het lokale systeem maar op de NIS server. Vrijwel alle nadelen van de hosts file gaan ook op voor hostnaam resolutie via NIS. Het enige voordeel is dat het onderhoud van de tabel maar op één plek hoeft te geschieden, namelijk op de NIS server. Daar staat dan weer tegenover dat er een extra schakel is die kapot kan gaan, namelijk de netwerkverbinding tussen de NIS client en de NIS server.

Een operating systeem of TCP/IP protocolstapel moet expliciet ondersteuning bieden voor NIS om hiervan gebruik te kunnen maken. De in de PC-wereld alomtegenwoordige Microsoft TCP/IP software biedt die ondersteuning niet! NIS heeft dan ook vrijwel alleen opgang gemaakt in de UNIX wereld. Daarnaast is standaard-NIS ook een beveiligingsrisico van de eerste orde.

## 8.7 Domain Name Server

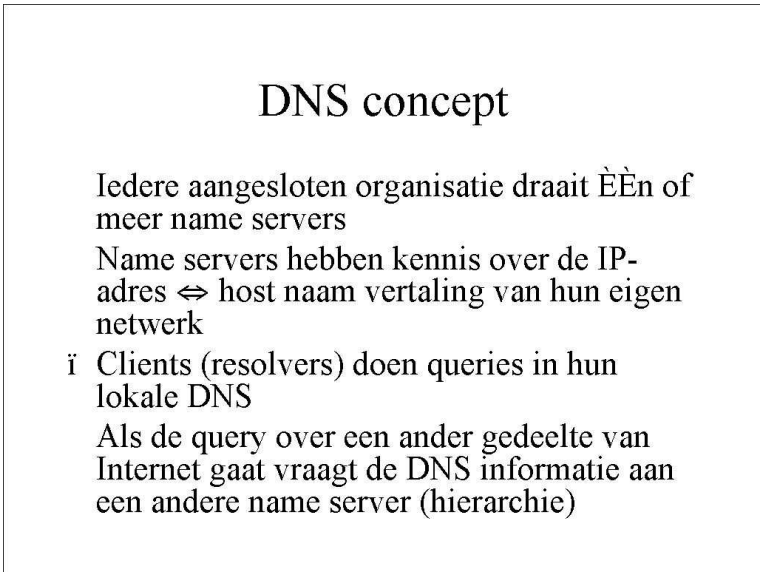



**Notities**

De derde methode van hostnaam resolutie is werken met een Domain Name Server. Een Domain Name Server is een serverapplicatie met kennis over de relatie tussen de hostnamen en IP-adressen van een gedeelte van het netwerk. De name server accepteert queries van klanten (de zogenaamde resolvers) en voert die queries vervolgens uit. Een resolver die wil weten welke IP-adressen er bij een hostnaam horen, of welke hostnaam er bij een IP-adres hoort, kan dit dus aan zijn name server vragen.

Tot op dit punt lijkt de werking van de DNS vrijwel gelijk aan NIS: een klant heeft een vraag over hostnaam¡-¿IP-adres vertaling en stuurt die door naar zijn lokale server. Een belangrijk verschil tussen de DNS en NIS is echter dat de DNS een query over een gedeelte van het netwerk waar hij geen verstand van heeft doorstuurt naar een andere (hogere) name server. Één name server hoeft dus niet alle hostnaam¡-¿IP-adres relaties in het gehele netwerk te weten. Alle DNS'sen samen vormen een soort gedistribueerde database waarmee de hostnaam naar IP-adres vertalingen (en vice versa) van het gehele netwerk kunnen worden opgelost. Om deze reden is de DNS het ideale mechanisme voor gebruik in Internet. Het zou immers onmogelijk zijn om een vertaaltabel (à la NIS of de hosts file) te maken voor het gehele Internet. Het gebruik van een name server is dan ook verplicht voor de aan Internet aangelosten netwerken.

## 8.8 DNS concept




**Notities**

De werking van de Domain Name Server is sterk gebaseerd op de domain naamgeving die in Internet wordt gehanteerd. Voor ieder (sub)domain geldt dat er een name server moet zijn die *authoritative* moet zijn voor dat (sub)domain. Een authoritative name server heeft de ultieme kennis op het gebied van hostnamen en IP-adressen voor dat gedeelte van het netwerk. Stel dat het bekende bedrijf Muppet Laboratories zich aansluit op het Internet dan wordt van ze verwacht dat ze een name server inrichten dir authoritative is voor het domain '.muplab.com'.

Nu kunnen de netwerkbeheerders van Muppet Laboratories ervoor gekozen hebben om het domain verder onder te verdelen in subdomains, bijvoorbeeld '.sales.muplab.com', '.research.muplab.com' en '.swamp.muplab.com'. Ook voor deze subdomains dient er een authoritative name server te zijn. Dit mag dezelfde name server zijn als de authoritative name server voor '.muplab.com'. Een name server mag namelijk autoritative zijn voor meerdere (sub)domains. Het is echter ook mogelijk om aparte authoritative name servers in te richten voor de subdomains. In dat geval zeggen we dat de autoriteit voor de subdomains is *gedelegeerd*.

Een client computer in het Muppet Laboratories netwerk die een vraag heeft over hostnamen en/of IP-adressen richt die vraag aan zijn lokale name server. Deze name server kijkt of de vraag over een (sub)domain gaat waarover hij de autoriteit heeft. Als dat zo is beantwoord hij de vraag uit zijn eigen tabellen. Als de vraag over een gedeelte van het netwerk gaat waarover deze name server geen kennis heeft stuurt hij de vraag op naar een andere name server. In de configuratie van de name server kan worden aangegeven welke name servers verantwoordelijk zijn voor welk gedeelte van het netwerk. Indien voor het gevraagde (sub)domain niet expliciet is aangegeven welke name

server ervoor verantwoordelijk is wordt de vraag doorgestuurd naar een zogenaamde *root name server*. Deze root name server kennen de name servers voor de Internet top level domains (.nl, .com, .org, ...).

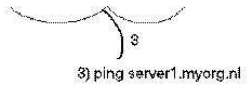
Indien een query is doorgegeven aan een andere name server kunnen er twee soorten antwoorden komen:

1. Het antwoord
   In dit geval zijn we blij. De name server aan wie we de query hebben doorgegeven wist klaarblijkelijk het juiste antwoord. Onze name server geeft nu dit antwoord terug aan de client (resolver).

2. Het IP-adres van een andere name server
   De name server wist het antwoord op de vraag niet, maar hij kent wel een name server waarvan hij vermoedt dat die het antwoord wel weet. In dat geval zal onze name server de query nog eens herhalen, maar nu naar de ‘nieuwe’ name server.

Let op dat een name server nooit een query forward over een (sub)domain waarvoor hij zelf authoritative is.

## 8.9 DNS voorbeeld




orbeeld

**Notities**

In dit voorbeeld is grafisch weergegeven hoe een DNS query in zijn werk gaat.

## 8.10 Resolver configuratie voorbeeld


**Notities**

In een netwerk waarin gebruik wordt gemaakt van een Domain Name Server dienen een aantal zaken te zijn geregeld. Allereerst dienen er natuurlijk één of meer systemen als DNS te zijn ingericht. Deze server systemen moeten 7x24 uur bereikbaar zijn voor het beantwoorden van name queries. Dit laatste geldt met name indien het netwerk is aangesloten op het Internet. Domain Name Servers worden daarom meestal geımplementeerd op de UNIX of Windows/NT systemen die ook andere Internet server taken verrichten (zoals email server en World Wide Web server).

De configuratie van een DNS is tamelijk complex. Naast de wat meer voor de hand liggende vertalingen van hostnaam naar IP-adres moet een DNS ook antwoorden kunnen geven op omgekeerde queries (IP-adres naar hostnaam), of queries voor email servers (de zogenaamde MX records).

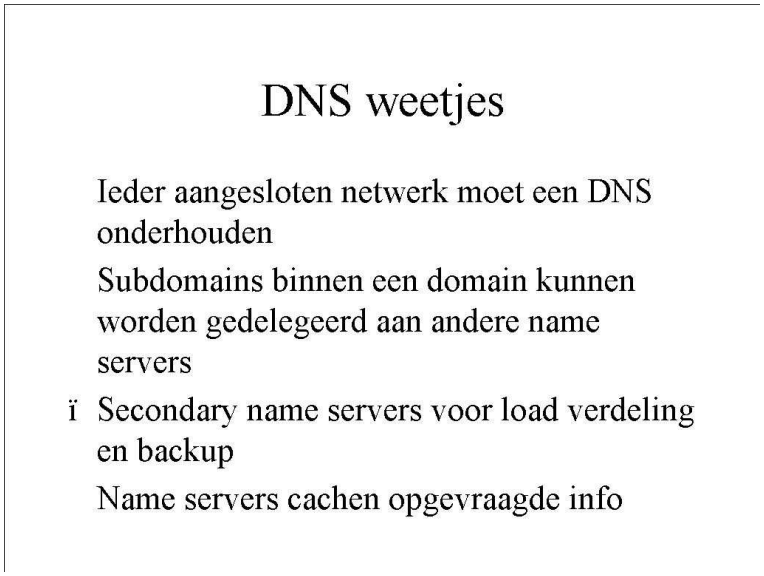
De DNS clients (de zogenaamde resolvers) zijn wat eenvoudiger te configureren. Op

de slide ziet u een voorbeeld van de configuratie van een Windows NT machine die moet worden geconfigureerd als DNS client. In principe zijn minimaal drie instellingen nodig om dit succesvol aan de praat te krijgen:

1. De (simpele) hostnaam van het eigen systeem
2. Het (sub)domain waarin deze node ‘zit’
3. Een lijst van name servers waarvan deze node gebruik kan maken

Het tweede item wordt door de resolver gebruikt om een simpele hostnaam die door de gebruiker wordt opgegeven uit te breiden met een (sub)domain gedeelte.

## 8.11 DNS weetjes




**Notities**

In principe moet dus ieder netwerk wat aan Internet is aangesloten een Domain Name Server in stand houden. Door de onderlinge samenwerking tussen de name servers kan de gehele wereld weten wat de relatie tussen IP-adressen en hostnamen is in dat gedeelte van het Internet. Het is vaak raadzaam om hele grote netwerken verder op te splitsen in subdomains. In dat geval kunnen er voor die subdomains aparte name servers worden opgetrokken. Dit verdient met name om beheertechnische redenen de voorkeur. De ‘hoofd name server’ van het netwerk delegeert dan de autoriteit over dat gedeelte van het domain naar de name server van het subdomain.

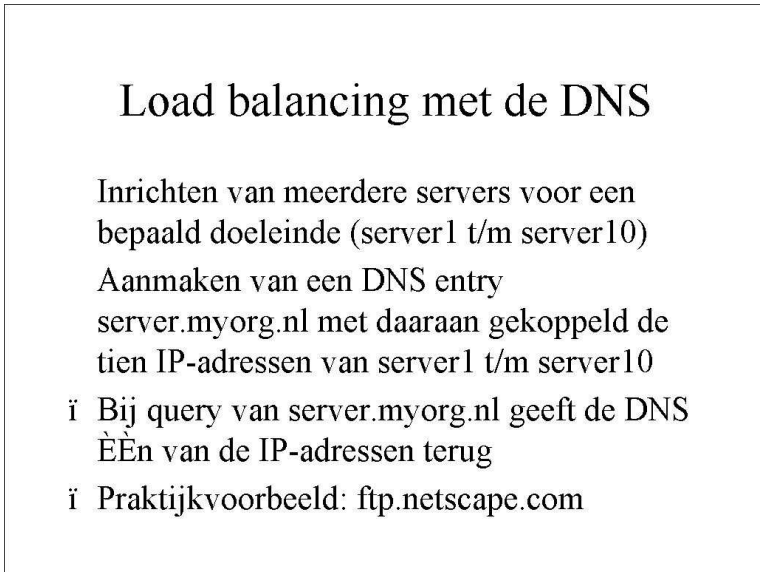
In netwerken die aan Internet zijn aangesloten heerst vaak terechte angst dat via de Domain Name Server teveel informatie ‘naar buiten’ komt over hoe het interne netwerk is opgedeeld. Beheerders hebben namelijk de mogelijkheid om via zogenaamde HINFO records extra informatie over de nodes in het netwerk op te nemen in de name server database. Deze HINFO informatie kan inbrekers op het spoor brengen van interessante systemen in het interne netwerk. Om die reden draaien veel organisaties twee primaire name servers voor het domain. Één voor intern gebruik, waarin alle informatie is opgenomen, en één voor extern gebruik, met daarin slechts een aantal (publiek toegankelijke) systemen en met beperkte informatie.

Het DNS concept kent naast primaire name servers die de ultieme kennis hebben over een (gedeelte van een) domain ook secondary name servers. Deze secondary name servers fungeren als backup name server in het geval de primaire server uitvalt en voor het verdelen van de aanvragen over de verschillende servers. Een secondary name server heeft in principe dezelfde kennis over een domain als de primaire name server. Een secondary name server hoeft echter niet te worden voorzien van deze informatie. Hij

downloadt deze automatisch van de primaire name server. Indien de tabellen van een primaire name server worden aangepast zullen de bij die primaire server behorende secondary servers automatisch (na enige tijd) een download doen van de nieuwe informatie. Door op strategische punten in het netwerk secondaire name servers te plaatsen kan de netwerkbeheerder de kwetsbaarheid en performance van het netwerk ten aanzien van name queries gunstig beınvloeden.

In een DNS omgeving is het dus in principe zo dat de resolvers (de clients) alle queries naar hun lokale name server sturen en dat die het op zich neemt om met de andere name servers op de aardbol te gaan praten over het oplossen van de query. Als de lokale name server dan uiteindelijk het antwoord op een query boven water heeft getoverd zou het natuurlijk zonde zijn indien hij dat dan onmiddelijk weer zou vergeten. Het zou namelijk best eens kunnen gebeuren dat deze of een andere client over niet al te lange tijd deze hostnaam weer eens gebruikt. Om redenen van efficiëntie *cachen* name servers daarom de antwoorden van de uitgevoerde queries. Indien een name server een query ontvangt waarvoor hij het antwoord in zijn cache heeft kan het antwoord onmiddelijk worden teruggegeven zonder dat daar verdere raadpleging van andere name servers voor hoeven te worden gedaan. Dit verhoogt de response van de name server aanzienlijk. Indien nu echter de netwerkbeheerder van de remote node besluit om de hostnaam ¡-¿ IP-adres koppelingen te wijzigen dan zijn er wellicht nog een aanzienlijk aantal name servers op de planeet die nog de oude informatie in hun caches hebben staan. Om hier enige controle over te kunnen uitoefenen kan de beheerder van een DNS voor “zijn” entries een zogenaamde *Time To Live* (TTL) specificeren. Dit is een parameter die bepaalt hoe lang de informatie in de caches van de remote name servers mag blijven staan. Name servers gooien entries uit hun cache weg waarvan de TTL is verstreken en doen indien nodig een nieuwe query bij hun collega name servers.

## 8.12 Load balancing met de DNS




**Notities**

De recente populariteit van Internet hebben de beheerders van zeer populaire sites ertoe gebracht om de verzoeken voor bestanden of HTML pagina's van die sites met behulp van de DNS op zeer creatieve wijze te verdelen over meerdere servers. Een voorbeeld van een dergelijke populaire site wordt gevormd door de FTP servers van Netscape. Het aantal verzoeken dat in piekuren aan Netscape wordt gericht is dermate groot dat het vrijwel onmmogelijk is om dit door slechts één server te laten afhandelen.

Om die reden gaan dit soort sites ertoe over om niet één maar bijvoorbeeld tien servers in te richten met precies dezelfde inhoud. Toch willen zij slechts één hostnaam aan de rest van Internet bekend maken. Via creatief gebruik van het DNS mechanisme kan ervoor worden gezorgd dat de queries voor die ene hostnaam over meerdere fysieke servers worden verspreid.

Hoe wordt dit opgelost?

Stel we richten tien servers in, met de IP-adressen 145.87.77.1 tot en met 145.87.77.10. Naar de buitenwereld willen we echter maar één hostnaam bekend maken voor gebruik door deze servers: "www.muplab.com". Wat we nu doen is dat we in de DNS van Muppet Laboratories de naam "www.muplab.com" opnemen met daaraan gekoppeld de tien IP-adressen 145.87.77.1 t/m 10. De DNS denkt nu dat www.muplab.com een machine is met tien netwerkinterface kaarten. Dat deze IP-adressen in werkelijkheid aan tien verschillende machines toebehoren weet de DNS niet.

Als een klant nu contact maakt met "www.muplab.com" wordt de name query uiteindelijk ontvangen door de DNS van Muppet Laboratories. Deze sorteert de lijst van IP-adressen in (semu-)willekeurige volgorde en beantwoordt de query. De resolver zal

normaal gesproken het eerste IP-adres in de lijst pakken en daarmee contact zoeken. Door de semi-random volgorde van de IP-adressen in het antwoord is dus nooit vooraf te zeggen met welke fysieke server er contact wordt gemaakt. Indien een groot aantal clients met www.muplab.com wil praten worden de verzoeken dus verdeeld over de tien verschillende servers van het bedrijf.

Om dit mechanisme goed te laten werken moet de DNS beheerder van Muppet Laboratories de TTL waarde van de www.muplan.com entry de waarde 0 hebben[2].

[2]The proof of this is left to the reader as an exercise

# Hoofdstuk 9

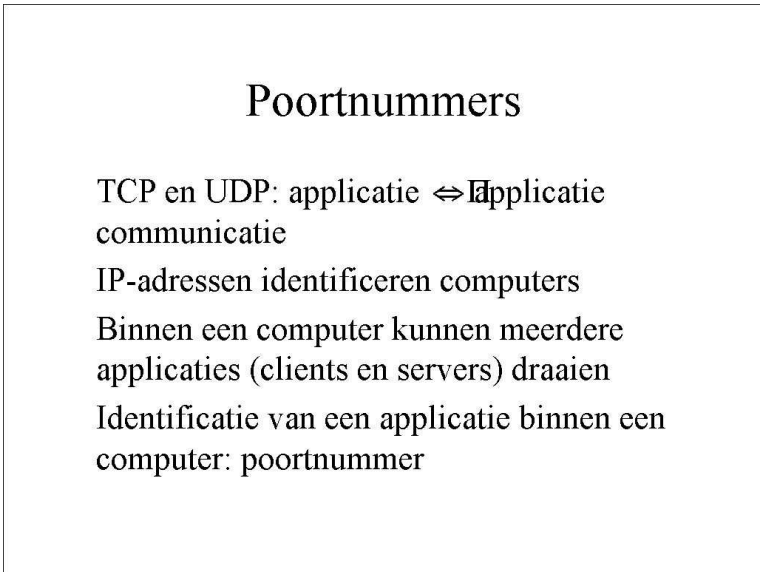
# Geavanceerde onderwerpen

## 9.1 Gevanceerde onderwerpen


**Notities**

In dit hoofdstuk worden enkele geavanceerde onderwerpen op het gebied van TCP/IP behandeld

## 9.2 Poortnummers



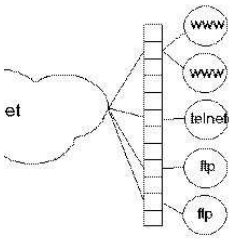

**Notities**

Één van de vragen die we nog niet hebben beantwoord is hoe TCP/IP bij het sturen van een pakketje naar een applicatie weet voor welke van de applicaties die er in een node draaien het pakketje bedoeld is. In principe kunnen er namelijk in een node makkelijk meerdere applicaties tegelijk draaien, en meestal zelfs meerdere *instances* van dezelfde applicatie.

De oplossing hiervoor wordt geboden door iedere TCP/IP verbinding zowel aan de client als aan de server zijde te voorzien van een uniek nummer: een zogenaamd poortnummer. Een server applicatie zal normaal gesproken bij het opstarten een TCP/IP verbinding openzetten en dan in de zogenaamde *accept* mode gaan. Dit betekent dat de applicatie stil staat totdat er door een client een poging wordt gedaan om aan die service aan te koppelen. Bij het maken van die verbinding dient de server applicatie het poortnummer wat hij wenst te gebruiken te specificeren.

Een client moet bij het opzetten van een verbinding in principe twee poortnummers specificeren: het poortnummer wat hij zelf wil gebruiken om de verbinding aan de client zijde te identificeren, en het poortnummer van de applicatie in de remote node.

## 9.3 Poortnummer schematisch voorbeeld




**Notities**

Dit concept kan worden gevisualiseerd door iedere computer te zien als een hotel met verschrikkelijk veel kamers. Op iedere kamerdeur zit een nummer geplakt, en in die kamer wonen één of meer applicaties (van hetzelfde type). Een server applicatie betrekt een kamer en hangt een bordje met “onbezet” aan de deur. Vervolgens wacht deze totdat er iemand langs komt.

Een client applicatie zoekt een vrije kamer in zijn eigen hotel en stuurt een boodschapper op pad. Deze boodschapper krijgt de opdracht om een bericht af te leveren in een bepaald hotel (IP-adres) en in een bepaalde kamer (poortnummer). De boodschapper (TCP/IP) kan aan de afzender en geadresseerde precies zien van wie het bericht afkomt en naar wie het toe moet.

## 9.4 Poortnummers toelichting


**Notities**

Met het commando “netstat -an” (UNIX en Windows/NT) kunnen we een overzicht maken van welke verbindingen er open zijn en welke servers er in de “accept” mode staan.

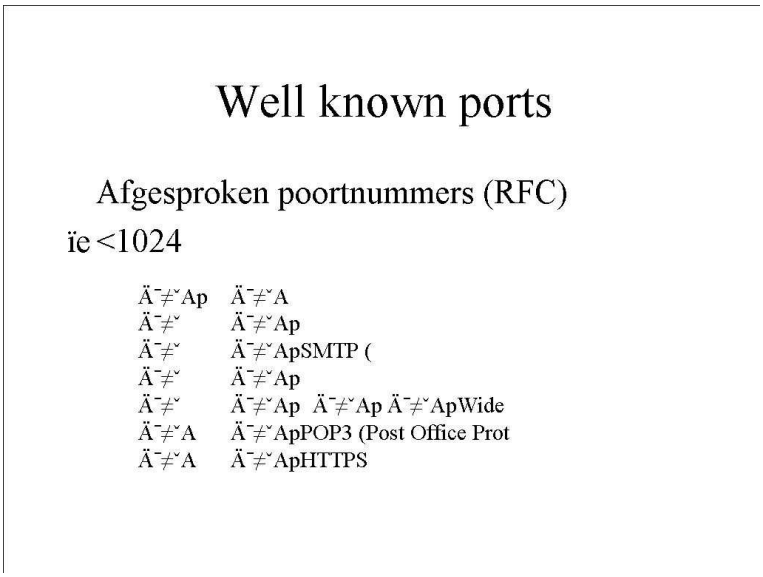
```
$ netstat -an
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:1025         127.0.0.1:1026         ESTABLISHED
  TCP    127.0.0.1:1026         127.0.0.1:1025         ESTABLISHED
  TCP    193.78.240.176:1043    193.78.233.1:23        ESTABLISHED
  UDP    0.0.0.0:135            *:*
  UDP    10.49.0.87:137         *:*
  UDP    10.49.0.87:138         *:*
  UDP    193.78.240.176:137     *:*
  UDP    193.78.240.176:138     *:*
```

Een verbinding kan uniek worden geıdentificeerd aan de combinatie van Source IP-adres+Source poortnummer en Destination IP-adres+Destination poortnummer. Aan bovenstaande output kan ik zien dat er een open verbinding is tussen een applicatie op het lokale systeem poort 1043 en een applicatie op systeem 193.78.233.1 poort 23.

Voor client systemen maakt het normaal gesproken niet uit op welke poort ze huizen. In principe kan elke poort worden gebruikt voor een uitgaande verbinding. Voor server

applicaties daarentegen is het vrijwel verplicht dat we vantevoren weten op welke poort ze zitten. Het zou anders voor een client applicatie zo goed als onmogelijk zijn om een verbinding met de server op te bouwen. Meestal is er om die reden een afspraak tussen de client en de server over welke poort er voor de communicatie wordt gebruikt. Bij een aantal TCP/IP applicaties kan de systeembeheerder bij de installatie een vrije poort selecteren en aan de applicatie toekennen.

## 9.5 Well known ports




**Notities**

Om te voorkomen dat veelgebruikte applicaties ruzie krijgen over welke poortnummers ze gebruiken is er in Internet een afspraak gemaakt over welke populaire applicaties welke poortnummers gebruiken. We noemen deze de *Well Known Ports*. Zo is bijvoorbeeld afgesproken dat telnet altijd poort 23 gebruikt, het World Wide Web poort 80 en email poort 25.

## 9.6 Routering

### Routering

- Iedere node is medeverantwoordelijk voor routering
- Een IP-pakketje hopt van node tot node
- Iedere node neemt een beslissing over wat de volgende node in het pad wordt
- Geen ìcentraal overzichtî over de te volgen route
- *Time To Live* bewaakt vicieuze cirkels

**Notities**

Een aspect waarin TCP/IP wel tamelijk uniek is, is routering. Routering is het proces volgens welke pakketje door het complexe netwerk reizen. TCP/IP is een zogenaamd *packet switched network*. Dit betekent dat per pakketje beslissingen worden genomen omtrent hoe pakketjes door het netwerk reizen. In een TCP/IP netwerk is er geen centrale instantie die beslist hoe de pakketjes door het netwerk moeten reizen. In principe is iedere node medeverantwoordelijk voor de routering in het netwerk.

Op het moment dat een node een pakketje moet (door)zenden naar een andere node gaat hij bekijken of die andere node met één hop kan worden bereikt. Dit doet zich voor als de nodes onderdeel zijn van hetzelfde local area network of als er een rechtstreekse point-to-point verbinding tussen de nodes is. Als dat het geval is dan wordt het pakketje naar die node gezonden.

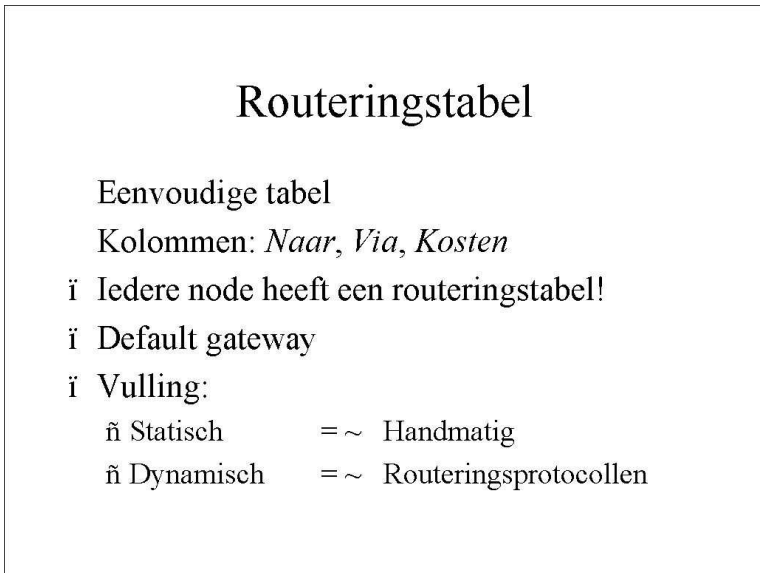
Indien de doelnode niet met één hop kan worden bereikt gaat de node zijn routeringstabellen raadplegen om te bekijken naar welke volgende node het pakketje moet worden gestuurd. Als dat kan worden vastgesteld dan wordt het pakketje bij die node afgeleverd. Vervolgens gaat die weer eenzelfde procedure in om te bekijken of het pakketje met één hop kan worden afgeleverd en zo voorts...

In principe zou zich een situatie voor kunnen doen waarbij pakketjes oneindig door het netwerk reizen zonder ooit op de plaats van bestemming aan te komen. Met name indien de routeringstabellen verkeerd zijn gevuld (bijvoorbeeld omdat ze naar elkaar wijzen) kan een dergelijke cirkel ontstaan. Om dit tegen te gaan zit er in een IP-pakketje een *Time To Live* veldje[1]. Bij iedere hop wordt dit veldje met één verlaagd. Zodra een

[1] niet te verwarren met het Time To Live veld in de Domain Name Server tabellen

node merkt dat het veldje de waarde nul krijgt, gooit deze het pakketje weg.

## 9.7 Routeringstabel

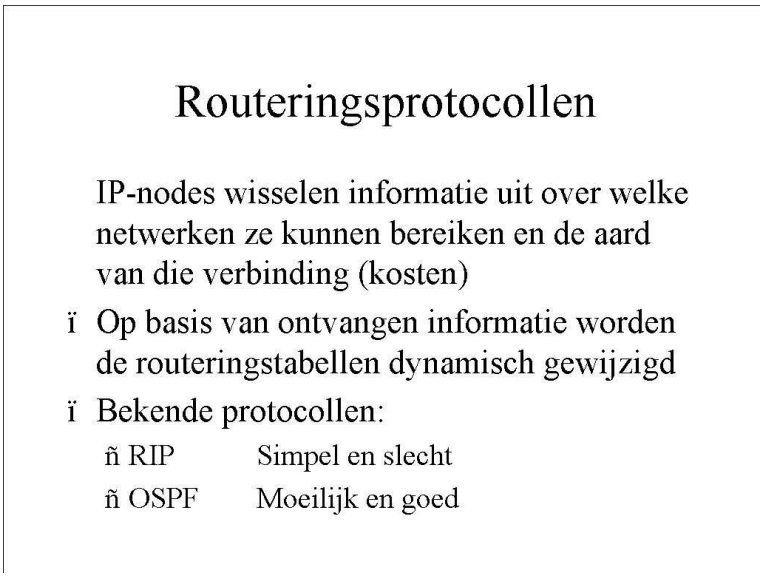



**Notities**

Om de routering in een TCP/IP netwerk rond te krijgen dient iedere node dus te zijn geconfigureerd met een routeringstabel. Met name de centrale nodes in het netwerk (de routers) hebben grote routeringstabellen. In principe is een IP-routeringstabel een eenvoudig ding. Het bestaat uit entries die beschrijven “Om bij *x* te komen kun je het beste het pakketje doorsturen naar *y*”. Hierbij staat *x* voor een systeem of netwerk en *y* voor het IP-adres van de node waarheen het pakketje moet worden verzonden voor de volgende hop. Aan een entry kan optioneel nog een gewicht worden gehangen. IP routeert een pakketje via de node waaraan het laagste gewicht hangt.

In veel gevallen heeft zo’n routeringstabel ook nog een *default* entry. Dit is een node waarnaartoe alle pakketje moeten worden gestuurd waarvoor geen specifiekere entry in de routeringstabel is opgenomen. We noemen dit ook wel de *default gateway*.

## 9.8 Routeringsprotocollen

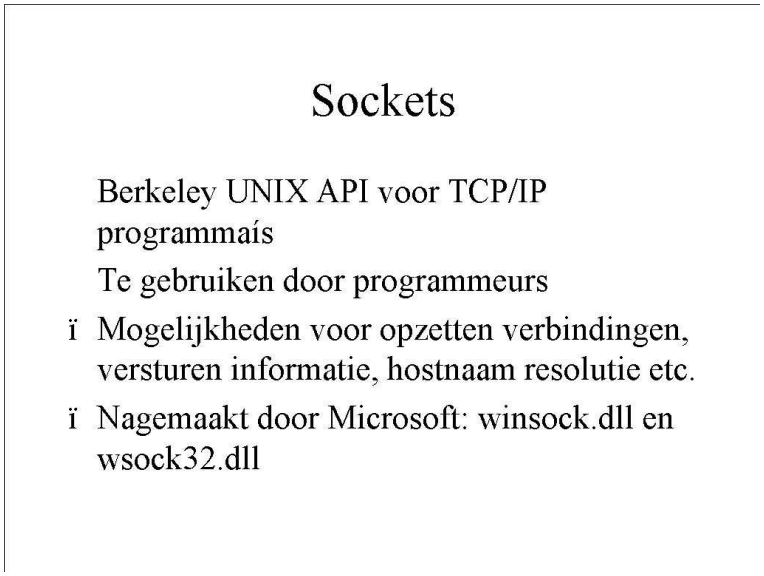



**Notities**

In complexe, dynamische, netwerken is het natuurlijk onbegonnen werk om de routeringstabellen van de verschillende nodes met de hand te configureren. Als verbindingen wegvallen, overbelast raken of weer *online* komen dienen de verschillende routeringstabellen te worden aangepast om deze feiten te reflecteren. Om deze dynamiek ten gunste aan te wenden zijn er door diverse bedrijven en universiteiten routeringsprotocollen bedacht. Via routeringsprotocollen houden nodes in het TCP/IP netwerk elkaar continu op de hoogte van de stand van zaken in het netwerk. Routers sturen elkaar gegevens toe over welke interfaces *up* en *down* zijn, wat de belasting van die interfaces is en welke subnetwerken via die interfaces kunnen worden bereikt. Op basis van die informatie kan een ieder zijn eigen routeringstabellen aanpassen.

Routeringsprotocollen kunnen worden beoordeeld op hoe snel en nauwkeurig ze informatie over het netwerk door het netwerk verspreiden. Twee van de meest gebruikte routeringsprotocollen van dit moment zijn RIP (Routing Information Protocol) en OSPF (Open Shortest Path First). Met name OSPF maakt de laatste tijd opgang.

## 9.9 Sockets




**Notities**

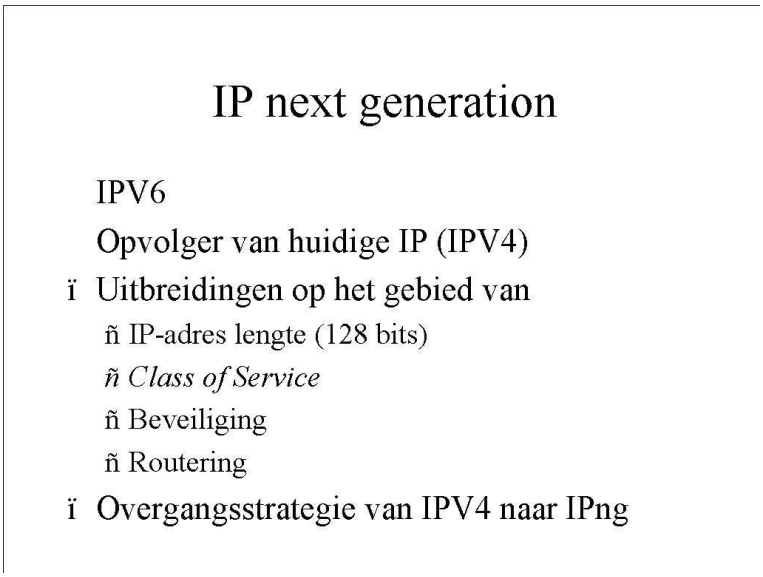
Een veelgehoorde kreet in TCP/IP land is het begrip *socket*. Wat zijn deze sockets nu eigenlijk?

Een "socket" is een programmatisch concept waarmee een programmeur een netwerkverbinding kan abstraheren. Populair gezegd vormen sockets voor de TCP/IP wereld wat *file descriptors* voor de disk wereld zijn: identificaties van open verbindingen of bestanden. In weze vormen sockets een *Application Programmer Interface* voor het openen, lezen, schrijven en sluiten van netwerkverbindingen.

De socket libraries zijn ontwikkeld door de universiteit van Berkeley als middel om vanuit een C programma TCP/IP netwerkverbindingen te manipuleren. Met de calls in de socket library kan een programmeur netwerkverbindingen opzetten, sluiten en gebruiken. Sinds de invoering zijn de Berkeley sockets de enige API voor het gebruik van TCP/IP geweest onder UNIX en VMS.

Sinds ook Microsoft het TCP/IP licht heeft zien branden en TCP/IP protocolstapels ter beschikking zijn gekomen onder DOS en Windows zijn ook de Berkeley sockets ter beschikking gekomen onder deze Microsoft besturingssystemen. De befaamde "winsock.dll" is in weze niets anders dan de van Berkeley sockets gekopieerde Microsoft standaard voor het gebruik van TCP/IP onder DOS/Windows.

## 9.10 IP next generation



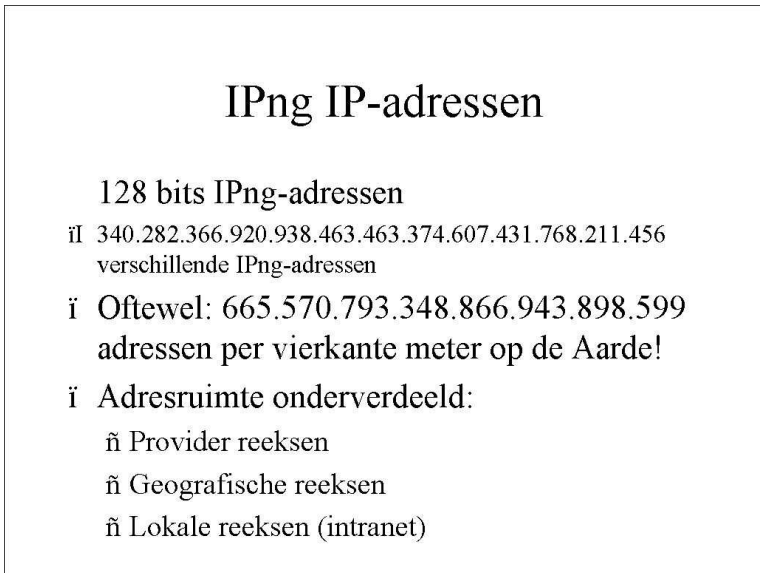

**Notities**

Bij tijd en wijle blijtk dat de huidige versie van IP (IP versie 4, IPV4) al weer een aantal jaartes oud is. Alhoewel IPV4 al zijn tanden nog heeft en nog leest zonder bril, blijkt toch regelmatig dat dit protocol (in computertermen) stokoud is. Om deze reden hebben de TCP/IP goeroes van het IAB de koppen bij elkaar gestoken en is er nagedacht over de volgende versie van IP.

Deze niewe versie gaat door het leven als IP next generation (IPV6, ook wel IPng). Alhoewel de wensenlijst voor uitbreiding van IPV4 tamelijk gigantisch was zijn de IAB-architecten erin geslaagd het protocol eenvoudig en simpel te houden. De meest opvallende uitbreidingen in IPng zijn de uitgebreidere IP-adressen (128 bits) en de introduktie van een *Class of Service* flow label waarmee aan een IP-pakketje een indicatie kan worden gehangen van de gewenste netwerk service (interactief verkeer, online video, file transfer, bulk verkeer en dergelijke).

Toen het ARPANET indertijd overging van de ARPANET protocollen naar TCP/IP is dit gedaan volgens het *Big Bang* model. Op uur U gingen alls computers uit en herstartten ze met de juiste netwerksoftware. Een dergelijk aanpak voor de overgang naar IPV6 is natuurlijk vandaag aan de dag uit den boze. IPV4 en IPV6 zullen geruime tijd naast elkaar moeten bestaan en er zijn dan ook allerlei voorzieningen getroffen die een geleidelijke conversie mogelijk maken.

## 9.11 IPng IP-adressen




**Notities**

Een vrij opvallende en broodnodige uitbreiding in IPng is de introduktie van langere IP-adressen: 128 bits in plaats van de 32 bits in IPV4. Dit geeft ruimte voor een werkelijk verbluffend aantal IP-adressen: namelijk $2^{128}$. Volgens moderne schattingen zitten er minder dan $2^{128}$ atomen in heel het universum, dus het zit er wel in dat we met de 128 bits IPng adressen een tijdje vooruit kunnen.

Net als in IPV4 is in IPV6 de gehele adresreeks onderverdeeld in verschillende reeksen. Nieuw is echter dat daarin reeksen zijn onderkend voor lokaal gebruik (in een Intranet), reeksen per Internet provider en reeksen voor diverse geografische gebieden op de wereld. Deze voorzieningen bleken nodig om ook in een IPng netwerk met eindige routeringstabellen te kunnen werken.

# Deel III

# De wondere wereld van Internet email

# Hoofdstuk 10

# Introductie

## 10.1 Welkom

Internet Info Shop 3

De wondere wereld van Internet
email

**Notities**

Welkom bij de derde Internet Info Shop: de wondere wereld van Internet email. In deze shop komen de basisbeginselen van het gebruik van email in Internet aan de orde. De kennis die in de eerste twee shops is aangedragen wordt hier bekend verondersteld.

## 10.2 Inhoud

Inhoud

Inleiding
De basis van Internet email
Email communicatieprotocollen
Randverschijnselen
Email beveiliging

**Notities**

In deze shop gaan we in op de volgende onderwerpen:

1. Inleiding
   Een algemene inleiding in Internet email. Wat zijn de belangrijkste eigenschappen.
2. De basis van Internet email
   De grondslagen van Internet email. Hoe ziet een bericht eruit? Hoe ziet een Internet email adres eruit? Welke componenten en programma's zijn benodigd om succesvol Internet email te gebruiken?
3. Email communicatieprotocollen
   Hoe communiceren de Internet email componenten met elkaar? Welke protocollen worden hiervoor gebruikt? Hoe werken die protocollen?
4. Randverschijnselen
   Overige onderwerpen, waaronder: aliassen, forwarding, adres herschrijving en MIME.
5. Email beveiliging
   Beveiliging van email berichten: encryptie, authenticatie en integriteit.

De nadruk ligt met name op de werking van Internet email zoals dit door de bank genomen is geımplementeerd. In theorie zijn er namelijk een groot aantal mogelijkheden die praktisch gesproken niet of nauwelijks voorkomen. Deze worden eventueel aangestipt, maar zeker niet diepgaand behandeld.

# Hoofdstuk 11

# Inleiding

## 11.1 Inleiding

Inleiding

**Notities**

In deze inleidende module maken we kennis met Internet email. Kennis van andere email systemen (bijvoorbeeld: MEMO, Lotus Notes of X.400 wordt bekend verondersteld.

## 11.2 Internet email

Internet email

Wijd verbreid
Gebaseerd op standaarden (RFCís)
KISS: *Keep It Simple Stupid!*
Geen *featurism* (in tegendeel)

**Notities**

Email is één van de oudste toepassingen van Internet. De standaarden volgens welke berichten worden geformatteerd stammen voor het grootste gedeelte uit het tijdperk dat TCP/IP nog niet bestond of nog niet algemeen was toegepast. Dientengevolge kan Internet email ook met andere communicatieinfrastructuren (zoals UUCP) worden verstuurd. Organisaties die zich aansluiten op Internet geven doorgaans email als één van de drie belangrijkste redenen voor de aansluiting.

Vrijwel alle Internet gebruikers maken gebruik van email. Internet email is daardoor de enige wereldwijd geımplementeerde email structuur. Internet email is grotendeels gebaseerd op algemeen geaccepteerde standaarden. Deze zijn, zoals in Internet gebruikelijk is, vastgelegd in RFC's. Deze standaarden zorgen ervoor dat alle email componenten met elkaar samen kunnen werken.

Een heel belangrijk aspect van Internet email is de overwegende simpelheid. De grondleggers van Internet wilden een eenvoudig te begrijpen en te implementeren email systeem. Ten gevolge van deze aanpak bevat Internet email alleen die features die strikt noodzakelijk zijn. Alhoewel diverse email systemen allerlei uitbreidingen op de standaarden implementeren wordt de basis gevormd door een beperkte verzameling mogelijkheden.

## 11.3 Wijd verbreid

Wijd verbreid

Want: simpel
Geintegreerd in UNIX
Veel public domain en shareware software
Gateways naar andere email en messaging systemen
Grootste Gemene Deler

**Notities**

De populariteit van Internet email kan uit de volgende factoren worden verklaard:

1. Simpel
   Internet email is eenvoudig te begrijpen, en eenvoudig te gebruiken. Voor Internet gebruikers is er een hele lage drempel om Internet email te gaan gebruiken. Zo is er bijvoorbeeld geen centrale instantie waar men zich dient te registreren.

2. Geıntegreerd in UNIX
   Het UNIX besturingssysteem wordt standaard uitgeleverd met de benodigde software om Internet email te bedrijven (als gebruiker en als mail hub). Organisaties die gebruik maken van UNIX servers hebben dus automatisch de mogelijkheid om Internet email te implementeren, zonder dat daarvoor extra software hoeft te worden aangeschaft. Zodra deze organisaties zich aan Internet aansluiten kan er dus vrijwel meteen van Internet email gebruik worden gemaakt.

3. Veel public domain en shareware software
   De eenvoud van de Internet email standaarden heeft ervoor gezorgd dat er in de loop van de jaren veel Internet email programmatuur is ontwikkeld. Het betreft hier zowel programmatuur voor gebruikers als voor systeembeheerders. Goede voorbeelden hiervan zijn de diverse email robots die mailing lists en dergelijke kunnen onderhouden. Voor organisaties die niet afwijzend staan tegenover het inzetten van dergelijke software staan er dus zeer veel mogelijkheden open.

De eenvoud en verspreiding van Internet email heeft er ondermeer voor gezorgd dat de leveranciers van proprietary email systemen gateways naar Internet email hebben

ontwikkeld. Hierdoor zijn interne email systemen vaak zonder al te veel problemen aan te sluiten op Internet.

## 11.4 Standaarden

Standaarden

ìDe jureî (RFC) en ìde factoî
Wel:
- Bericht formaat
- Bericht uitwisseling protocol
- Adressering

Niet (of nauwelijks):
- User interfaces
- Directory services
- Beveiliging

**Notities**

Email in een heterogeen netwerk als Internet kan natuurlijk alleen maar succesvol zijn als het goed gestandaardiseerd is. De meeste standaarden op het gebied van Internet email zijn vastgelegd in RFC's. Daarnaast zijn er nog enkele algemeen ingevoerde 'de facto' standaarden waar men zich optioneel aan kan conformeren (bijvoorbeeld PGP). In Internet email is alleen het hoogstnodige gestandaardiseerd: het formaat van de berichten, de adressering van gebruikers en de protocollen volgens welke berichten worden uitgewisseld.

Luxe verschijnselen als "Directory Services" en "Beveiliging" zijn niet of nauwelijks gestandaardiseerd.

## 11.5 KISS

KISS

Berichten formaat (gebaseerd op ASCII)
Uitwisselings protocollen (tekstueel)
Adressering
Geen complexe features

**Notities**

Internet email is erg eenvoudig van opzet. De bedenkers van de standaarden streefden naar een eenvoudig email systeem wat zou voorzien in de meest voor de hand liggende functies. Dit streven wordt tot op de dag van vandaag gehandhaafd. Recente uitbreiding aan de email standaard blijven eenvoudig van opzet. Complexe, moeilijk implementeerbare functies, worden geschuwd. Berichten, PDU's (Protocol Data Units) en adressen zijn bijvoorbeeld allemaal gebaseerd op leesbare ASCII tekst. Hierdoor kunnen deze componenten eenvoudig worden bekeken, aangemaakt en aangepast. De protocollen voor de uitwisseling van berichten kunnen bijvoorbeeld met de hand worden nagespeeld met telnet.

Aan Internet email componenten worden weinig verplichtingen gesteld. De diverse RFC's bevatten maar weinig functies die *moeten* worden geımplementeerd. Zo er al luxe features worden geboden zijn die optioneel. Een Internet email programma kan niet voorhands aannemen dat andere email software bepaalde features ondersteunt.

## 11.6 Geen featurism

Geen featurism¸úù∞Xuúúb◊

Niet:

- Moet antwoorden
- Terugnemen verzonden berichten
- Ontvangstbevestiging
- Zekerheid omtrent aankomst

Nauwelijks:

- Groepen
- Beveiliging (authenticatie, encryptie)

**Notities**

Op de side staan voorbeelden van features die Internet email standaard ontbeert. Sommige hiervan kunnen wel met behulp van opties of extra programma's worden geımplementeerd, maar ook hier geldt dat dit eigenlijk alleen kan als je zeker weet dat de tegenpartij in de email uitwisseling deze uitbreidingen ook ondersteunt.

## 11.7 Waarom dan toch Internet email

Waarom dan toch Internet email?

Het is er!
- Simpel
- Ruimschoots voorhanden (UNIX, PD, SW)

Het is makkelijk

Het is *cool*!

**Notities**

Zo op het eerste gezicht zitten er aan het gebruik van Internet email aardig wat beperkingen. Waarom zouden we dan Internet email toch gebruiken. De redenen hiervoor staan op de slide opgesomd:

1. Het is er!
   Internet email is wijd verbreid. Email gaat over communicatie met anderen. Vanuit dit oogpunt gezien kun je beter gebruik maken van een featureloos systeem wat iedereen kent dan van een (wellicht beter) *full feature* email systeem wat niemand gebruikt. Veel organisaties streven naar de *best of both worlds* en gebruiken een *full function* proprietary email systeem als intern email systeem met een gateway naar Internet email.

2. Ruimschoots voorhanden
   De benodigde software om Internet email te gaan bedrijven wordt ofwel met het operating system meegeleverd (met name UNIX) of kan tegen lage kosten worden verkregen als public domain of shareware software. De kosten zijn dus laag.

# Hoofdstuk 12

# De basis van Internet email

## 12.1 De basis van Internet email

De basis van Internet email

**Notities**

In deze module gaan we dieper in op de grondslagen van Internet email. Met name het formaat van een Internet email bericht en de structuur van een email adres komen aan de orde.

## 12.2 Email bericht

Email bericht

ASCII
Regels afgesloten met CRLF
Headers
Lege regel
Body (optioneel)

**Notities**

Het formaat van Internet email berichten is vastgelegd in RFC 822. Deze standaard is dermate aangeslagen dat andere Internet applicaties (zoals USENET news) gebruik maken van een nauw verwant berichtformaat. In een email bericht komen we de volgende informatie tegen:

1. Headers
   Deze bevatten de metainformatie van het bericht: van wie is het afkomstig, waar moet het naartoe, hoe heeft het gereisd, wat is de titel en zo voorts.
2. Body
   De inhoud van het bericht

De headers worden samengesteld, onderhouden en verwerkt door de email programmatuur; de body wordt ingevoerd en gelezen door de gebruikers. Het gehele bericht (headers en body) is een ASCII tekst, bestaande uit regels afgesloten met CRLF (Carriage Return + Line Feed). Een bericht begint altijd met de headers gevolgd door een lege regel en de body. Deze lege regel scheidt de headers van de body en is dus verplicht!

Het feit dat het gehele bericht (ook de metainformatie) uit ASCII tekst bestaat heeft tot gevolg dat een bericht kan worden aangemaakt met een gewone teksteditor.

## 12.3 Email bericht voorbeeld

Email bericht voorbeeld

```
Received: (from josv@localhost) by localhost (8.6.12/8.6.12)
      id QAA05758 for josv; Wed, 1 Jan 1997 16:13:51 +0100
Date: Wed, 1 Jan 1997 16:13:51 +0100
From: Jos Visser <josv>
Message-Id: <199701011513.QAA05758@localhost>
To: josv
Subject: Test bericht
Status: O

Dit is een testbericht

++jos
```

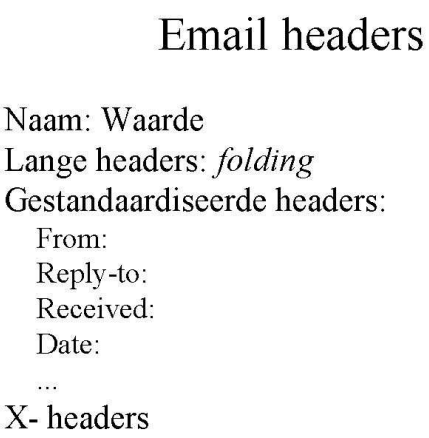
**Notities**

Op de slide staat een voorbeeld van een email bericht. Dit bericht is aangemaakt met het UNIX *mail* commando:

```
$ mail josv
Subject: Test bericht
Dit is een testbericht

++jos
.
```

De gebruiker heeft alleen de geadresseerde (josv), de titel (Subject) en het bericht ingegeven. De UNIX email programmatuur formatteert het bericht volgens de regels van RFC 822 en verstuurt het. Tegen de tijd dat het bij de geadresseerde aankomt ziet het eruit zoals op de slide is gepresenteerd. Merk op dat de ontvanger het bericht enigzin modificeert (voegt bijvoorbeeld een ‘Received’ header toe). De meeste email software laat alleen de afzender, titel en body zien. De metainformatie is echter volledig toegankelijk (met een optie van de email software) en begrijpbaar.

## 12.4 Email headers




**Notities**

De headers van een email bericht bevatten allerlei metainformatie over het bericht en de verzending ervan. Deze headers zijn, net als de rest van het bericht, gesteld in ASCII tekst. Een header bestaat uit een naam, een dubbele punt en een waarde. Omdat is vastgelegd dat headers altijd aan het begin van een bericht staan gevolgd door een lege regel (RFC 822) kan er nooit discussie zijn over of een bepaalde regel een header regel is of onderdeel van de body. Lange header regels mogen worden gesplitst over meerdere regels (*folding*). Een header regel die begint met LWSP (Linear White SPace) dient te worden opgevat als een vervolg op de voorgaande header regel.

In diverse RFC's is vastgelegd welke headers er in een bericht voor kunnen komen en wat ze betekenen. In zijn algemeenheid zal een email programma dat een bepaalde header niet begrijpt deze negeren. In de standaarden is tevens vastgelegd wat er in het 'waarde' gedeelte van een header mag staan. Sommige headers (zoals de 'Subject' header) zijn ongestructureerd en mogen zo ongeveer alles bevatten, terwijl het formaat van andere headers (bijvoorbeeld de 'Reply-To' header) aan vastgestelde regels dient te voldoen.

Het is in principe niet toegestaan om zelf nieuwe headers te verzinnen. Zelfs als een bepaalde header momenteel niet wordt gebruikt is het niet aan te raden om zonder nadere standaardisatie (door het IAB) een nieuwe header aan het bericht toe te voegen. Het is namelijk niet uit te sluiten dat de gekozen header naam in de toekomst wel zal worden gestandaardiseerd. In RFC 822 is vastgelegd dat alleen header namen die beginnen met 'X-' door gebruikers zelf mogen worden bedacht. Email software mag in principe aan deze user headers geen echte betekenis toekennen. De header 'X-Planet: Terra' mag dus zonder problemen door mijzelf aan een emailtje worden toegevoegd.

## 12.5 Email body

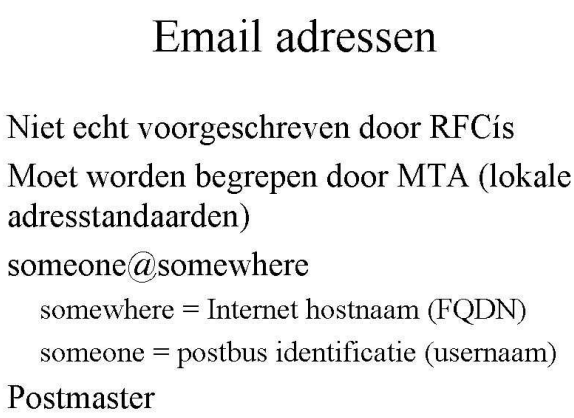
Email body

Ongestructureerde tekst
7-bit ASCII

**Notities**

Voor de body van een email bericht gelden eigenlijk nagenoeg geen regels. Alle tekst na de eerste lege regel in het bericht wordt opgevat als body. De berichtformaatstandaard spreekt zich niet echt uit over de structuur van de body, alleen dat deze bestaat uit regels van ASCII karakters afgesloten met een CRLF. De maximale lengte van de regels wordt beperkt door het transportprotocol (SMTP) en is duizend karakters.

## 12.6 Email adressen




**Notities**

Een belangrijk onderdeel van de Internet email standaarden wordt gevormd door het formaat van een Internet email adres. De RFC 822 standaard spreekt zich niet echt uit over het daadwerkelijke formaat van een email adres. Wel worden er enkele headers beschreven waarin adressen staan (zoals 'From:', 'To:' en 'Reply-To:') maar het precieze formaat waaraan de adressen dienen te voldoen wordt hier niet uitgebreid uit de doeken gedaan. Dit komt omdat email adressering niet zozeer een te maken heeft met het formaat van het bericht maar met het protocol wat de berichten verstuurt. RFC 822 compliant berichten kunnen op diverse manieren worden verstuurd (bijvoorbeeld SMTP of UUCP) en beide manieren kennen hun eigen adresseringsstructuur. In principe is een email adres alles wat door de lokale email programmatuur wordt begrepen. Hierdoor mag in principe iedere Internet site zijn eigen adresstandaarden vaststellen. Verschijnselen als geneste adressen en dergelijke worden hierdoor mogelijk gemaakt.

De adressen die doorgaans in Internet email worden gebruikt zijn die van het SMTP protocol. Deze SMTP email adressen hebben het formaat *someone@somewhere*. Hierbij is *someone* de benaming van een gebruiker of een postbus en *somewhere* een indicatie van waar die postbus uithangt, meestal een Internet node of domain naam. Aan *someone* mag geen betekenis worden toegekend. Meestal is het de gebruikersnaam van een gebruiker op een systeem, maar dit hoeft niet. De interpretatie van *someone* kan en mag alleen worden gedaan door de email software op de eindbestemming.

De email RFC's schrijven voor dat op iedere Internet email node (hub) een *someone* aanwezig moet zijn met de naam 'postmaster'. Dit is de postbus naar wie berichten/vragen kunnen worden gestuurd die te maken hebben met de email systemen. Zo stuurt de meeste email software foutberichten die gerelateerd zijn aan de verzending van email (ook) naar de 'postmaster' van het systeem in kwestie.

## 12.7 Email adres voorbeelden

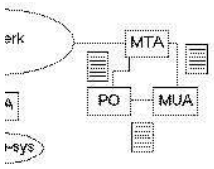
Email adres voorbeelden

josv@gatekeeper.osp.nl
jos@osp.nl
Jos.Visser@mail.osp.nl
J.C.W.Visser@localhost

**Notities**

Op de slide staan een aantal voorbeelden van geldige Internet email adressen.

## 12.8 Email en het netwerk




**Notities**

De kern van email zit hem natuurlijk in het verzenden van berichten door het netwerk. Het gehele Internet email systeem is onder te verdelen in drie componenten:

1. MUA - Mail User Agent
   Het user interface van email.
2. MTA - Mail Transport Agent
   De component die zorg draagt voor verzending van berichten door het netwerk.
3. PO - Post Office
   De component die berichten van deze node opslaat tot ze door de gebruiker worden gelezen

De gehele Internet email functionaliteit wordt geboden door het samenspel van deze drie componenten.

## 12.9 Mail User Agent

Mail User Agent

Gebruikersprogramma
Aanmaken nieuwe berichten (editor)
Verzenden berichten (naar MTA of PO)
Raadplegen PO voor ontvangen berichten
Lezen en beantwoorden ontvangen berichten
V.b.: /usr/bin/mail, Eudora, Netscape mail, Microsoft Internet mail en Exchange client

**Notities**

De MUA is het gebruikersinterface van Internet email. Via deze component maakt een gebruiker nieuwe berichten, verzendt ze en leest hij/zij ontvangen berichten. MUA's communiceren met de MTA's voor het verzenden van berichten. Hierbij gaat de MUA ervan uit dat de MTA zorg draagt voor de verdere aflevering. Zodra de MUA het bericht bij de MTA heeft gedropt wordt het als verzonden beschouwd. MUA's communiceren tevens met de PO's voor het ophalen/verwijderen van ontvangen berichten.

De MUA is volledig verantwoordelijk voor het formatteren van het bericht (RFC 822 standaard). Hiertoe bevat de MUA meestal een editor voor het ingeven van nieuwe berichten. Sommige MUA's bevatten een eigen berichten database waarin ontvangen berichten worden opgeslagen. Dit soort MUA's verwijdert meestal de ontvangen berichten meteen uit de PO.

Voorbeelden van MUA's zijn het UNIX mail programma, elm, PINE, Eudora, Netscape Mail, Microsoft Internet Mail en de MS Exchange client.

## 12.10 Mail Transport Agent

Mail Transport Agent

Ontvangen van email
- Andere MTAís in het netwerk
- MUAís

Verwerken ontvangen email
- Doorsturen naar andere MTAís (relaying, forwarding)
- Afleveren bij PO

V.b.: Sendmail, MS Exchange, Postal Union, smail

**Notities**

Internet MTA's dragen zorg voor verzending van mail door Internet. MTA's ontvangen berichten van MUA's en andere MTA's. Een MTA die een bericht ontvangt bepaalt wat de eindbestemming is van het bericht. Indien de huidige node de eindbestemming is levert de MTA het bericht af bij het lokale PO. Als de huidige node niet de eindbestemming is bepaalt de MTA de node waar het bericht naar toe moet en stuurt het bericht naar die MTA. Alhoewel de basisfuncties van MTA's tamelijk eenvoudig zijn kunnen MTA's al met al behoorlijk complexe programma's worden. Veel MTA's kunnen berichten aan verschillende typen MTA's en PO's afleveren.

Email gateways zijn in principe MTA's die de ontvangen berichten converteren naar een ander formaat en vervolgens afleveren bij het andere email systeem (PO of MTA).

Verreweg de bekendste Internet email MTA is 'sendmail'. Dit pakket wordt standaard meegeleverd met de meeste UNIX systemen. Wat sendmail zo bijzonder maakt is de combinatie van:

- ongekende kracht
- ongekende complexiteit
- ongekende security bugs

Ter illustratie: een aantal van de bekendste UNIX leveranciers geven geen support op sendmail indien de standaard meegeleverde configuratie file door de beheerder is aangepast!

Andere Internet MTA's zijn:

1. smail
   Een minder complexe vervanging van sendmail.
2. MS Exchange Server
3. Postal Union
   Gateway naar onder andere MS Mail.

## 12.11 Post Office

Post Office

Vasthouden ontvangen berichten
Mailbox structuur
Aanleveren berichten aan MUA
V.b.: /var/mail (UNIX)

**Notities**

De derde component in het Internet email gebeuren wordt gevormd door het post-kantoor. Deze component houdt berichten vast voor gebruikers van deze email node. MTA's leveren berichten aan het PO aan die deze vervolgens op een of andere manier opslaat en vervolgens op verzoek weer doorgeeft aan de MUA.

PO kunnen heel eenvoudige componenten zijn. In UNIX bestaat de PO bijvoorbeeld uit bestanden in de directory /var/mail. UNIX MUA's kunnen deze bestanden rechtstreeks raadplegen. MTA's kunnen met het programma 'mail.local' berichten aan de bestanden toevoegen. Een voorbeeld van een complexer PO is de Microsoft Exchange Server die er een geheel eigen database op nahoudt voor het opslaan van berichten.

# Hoofdstuk 13

# Email communicatie protocollen

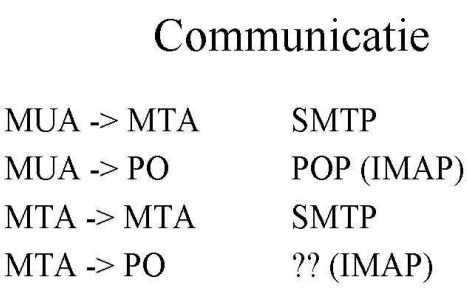
## 13.1 Email communicatie protocollen

Email communicatieprotcollen

**Notities**

In deze module wordt de wijze waarop de verschillende componenten met elkaar communiceren behandeld.

## 13.2 Communicatie




**Notities**

In een Internet email omgeving vindt de volgende communicatie plaats:

1. MUA - MTA, voor het verzenden van nieuwe berichten
2. MUA - PO, voor het ophalen van ontvangen berichten
3. MTA - MTA, voor het doorzenden van berichten die nog niet op hun eindbestemming zijn.
4. MTA - PO, voor het opslaan van berichten die op hun eindbestemming zijn aangekomen.

Om componenten van verschillend allooi met elkaar samen te laten werken zijn de SMTP, POP en IMAP protocollen gestandaardiseerd. Deze protocollen vormen een standaard manier waarop berichten tussen componenten kunnen worden uitgewisseld. Hierbij wordt SMTP gebruikt voor het verzenden van berichten, POP voor het ophalen van berichten en IMAP kan voor beide worden gebruikt. Los van deze standaard protocollen kunnen componenten nog proprietary wijzen hebben voor het verzenden of ophalen van berichten.

### 13.2.1 MUA - MTA

De communicatie tussen MUA en MTA geschiedt op het moment dat de gebruiker (via de MUA) een nieuw bericht heeft gemaakt en wil verzenden. De MUA neemt dan contact op met de MTA en levert het bericht daar aan. Daar iedere Internet mail MTA

het SMTP protocol ondersteunt is dit één van de meest voor de hand liggende mogelijkheden voor de communicatie tussen MUA en MTA. De meeste PC MUA's, zoals Netscape Mail en Eudora, gebruiken dan ook SMTP om mail bij de geconfigureerde MTA af te leveren. Sommige MUA's kunnen slechts met één soort MTA praten. In dat geval kan de MUA een MTA-specifieke methode gebruiken om het bericht bij de MTA af te leveren. Het UNIX mail commando roept bijvoorbeeld rechtstreeks het sendmail commando aan om zodoende het bericht in de mail queue van de lokale MTA te krijgen

### 13.2.2 MUA - PO

Een andere functionaliteit van de MUA is het lezen van ontvangen berichten. Deze berichten worden door de MTA's ontvangen en in de PO geplaatst. De communicatie tussen MUA en PO kan op drie manieren:

1. POP protocol
2. IMAP protocol
3. Proprietary methode

Zowel het POP als het IMAP protocol zijn gestandaardiseerde protocollen om over het netwerk te communiceren met een PO. Van deze twee protocollen is POP verreweg het populairst. IMAP, alhoewel completer en technisch superieur, is nog bijna nergens geımplementeerd. Vrijwel alle PC MUA' maken gebruik van het POP protocol om toegang te krijgen tot het PO. Ook hier geldt dat sommige MUA's slechts met één soort PO kunnen samenwerken. Het UNIX mail commando gebruikt geen POP of IMAP maar kijkt rechtstreeks in de PO bestanden (in de /var/mail directory).

### 13.2.3 MTA - MTA

Communicatie tussen Internet MTA's gebeurt volgens het SMTP protocol. Een MTA die een bericht ontvangt met SMTP controleert of hij het eindpunt is van het bericht. Zo ja dan wordt het bericht via een zogenaamde *delivery method* in het geconfigureerde PO geplaatst. Is het bericht nog niet op de eindbestemming dan wordt het in de uitgaande mail queue geplaatst. Een MTA heeft in principe niet door of hij een bericht ontvangt van een MUA of van een andere MTA.

### 13.2.4 MTA - PO

Zodra het bericht op de eindbestemming is aangekomen zal de MTA aldaar het bericht in de PO plaatsen. Alhoewel het IMAP protocol hiervoor in principe geschikt is wordt dit nog vrijwel nergens toegepast. De communicatie tussen MTA en PO is vrijwel altijd op non-standaard wijze geımplementeerd. Heel vaak wordt de MTA en PO functionaliteit door één software systeem geboden (bijvoorbeeld MS Exchange Server). De sendmail MTA daarentegen kan worden geconfigureerd met de methode volgens welke een bericht in de PO moet worden geplaatst (de zogenaamde *delivery agent* of *delivery method*.

## 13.3 SMTP

SMTP

Simple Mail Transfer Protocol
RFC 821
Diverse uitbreidingen
Transport van RFC822 compliant berichten via TCP/IP naar MTAís
Well known port: 25
Text based command-response protocol
Verzendende MTA/MUA heeft het initiatief
Simpeler wordt het niet meer!

**Notities**

Het SMTP protocol is dus *het* protocol voor het verzenden van berichten door Internet. Het protocol wordt begrepen door alle Internet MTA's en de meeste MUA's. De volledige specificatie van het protocol is neergelegd in RFC 821.

Kenmerkend voor SMTP is de overwegende simpelheid van het protocol. Het protocol is gebaseerd op ASCII commando's en antwoorden. Hierdoor is het bijvoorbeeld met een telnet programma handmatig uit te voeren. Het standaard RFC 821 protocol kan worden gebruikt om RFC 822 compliant ASCII berichten, bestaande uit regels met een maximale lengte van duizend karakters, te verzenden. Uitbreidingen aan SMTP staan onder andere het verzenden van 8 bits karakters toe. De bekende MTA welkomsboodschap "220 ESMTP spoken here" vertelt de verzendende MUA/MTA dat SMTP uitbreidingen zijn geımplementeerd.

## 13.4 SMTP sessie 1


**Notities**

Op deze en de volgende slide staan een eenvoudige SMTP sessie weergegeven. Deze sessie is met de hand uitgevoerd door een telnet naar poort 25 van de OSP mail hub (gatekeeper.osp.nl). Sommige MUA's kennen opties waarmee het verslag van de SMTP sessie kan worden ingezien.

Na het opzetten van de verbinding vertelt de mail server wie hij is. Het initiatief ligt vervolgens bij de client die zich met het 'HELO' commando moet aanmelden. De server antwoord vervolgens met een "250 ... pleased to meet you" melding. De verzending kan beginnen. De client geeft vervolgens een 'MAIL FROM' commando. Hiermee wordt gespecificeerd wie de afzender van het bericht is. De MTA reageert met een "250 ... ok" melding. Vervolgens komen één of meer 'RCPT TO' commando's waarmee de geadresseerden worden gespecificeerd.

## 13.5 SMTP sessie 2

SMTP sessie 2

data
354 Enter mail, end with "." on a line by itself
Subject: Weersomstandigheden

Wegens een depressie boven de noordelijke ijszee dit jaar geen kadootjes.
.
250 RAA23304 Message accepted for delivery
quit
221 gatekeeper.osp.nl closing connection

**Notities**

Vervolgens geeft de client het ‘DATA’ commando. De MTA antwoord nog met wat instructies, maar in principe volgt nu gewoon het bericht in RFC 822 formaat, eerst de headers, gevolgd door een lege regel en de body van het bericht. Het bericht wordt afgesloten door een regel met een enkele ‘.’. Regels in het bericht die beginnen met een ‘.’ worden voorzien van een extra ‘.’ om te voorkomen dat een dergelijke regel per ongeluk het SMTP DATA protocol beëindigt (character stuffing). Als de MTA het bericht accepteert reageert hij met een “250 .. message accepted for delivery” response. We kunnen nu vervolgen met het volgende bericht (‘MAIL FROM’ commando) of de SMTP sessie beëindigen met het ‘QUIT’ commando.

## 13.6 SMTP eigenschappen

SMTP eigenschappen

Simpel

Geen authenticatie verzendende MTA/MUA

Geen controle op verzender van het bericht

Geen encryptie

**Notities**

De in het SMTP voorbeeld behandelde SMTP commando's zijn ook zo ongeveer de enige commando's in het SMTP protocol. Het protocol voorziet dus niet in:

- Authenticatie
  De SMTP client hoeft zich niet te authenticeren. De MTA weet met welke client node hij communiceert (*getpeername()* socket functie) maar verder er is geen informatie beschikbaar over de verzendende MUA/MTA. Ook het email adres van de afzender wordt niet gecontroleerd maar *as is* doorgegeven.
- Encryptie
  Het bericht gaat clear text door het Internet en is daardoor gevoelig voor afluisteren en ongeautoriseerde wijzigingen.

Een ander kenmerkende eigenschap van SMTP is dat alleen de ontvangende MTA op de eindbestemming van een bericht de geldigheid van het email adres kan controleren. Er is immers geen centrale database met daarin de geldige postbusidentificaties (*someone*) van een systeem. Indien een fout wordt gemaakt in de naam van de geadresseerde wordt de email teruggestuurd door de voorlaatste MTA in het pad. We noemen dit verschijnsel een *mail bounce*.

## 13.7 POP

POP

Post Office Protocol
RFC 1725
Transport van berichten via TCP/IP van PO naar MUAís
Well known port: 110
Text based command-response protocol
MUA heeft het initiatief
Simpeler wordt het niet meer!

**Notities**

Het SMTP protocol is alleen geschikt voor het verzenden van email berichten. Voor het ophalen van berichten is dus een ander protocol nodig: POP, het Post Office Protocol. In één oogopslag is te zien dat het hier een broertje van het SMTP protocol betreft. Met het POP protocol kan een MUA contact maken met een PO en deze ondervragen. In tegenstelling tot SMTP moet een POP client zich autenticeren met een gebruikersnaam en wachtwoord. Verder kent POP wat meer commando's waarmee bijvoorbeeld het aantal berichten kan worden opgehaald, de grootte van de berichten en waarmee een bericht kan worden verwijderd uit de PO.

POP is gespecificeerd in RFC 1725. Het is momenteel het meest gebruikte protocol voor het ophalen van berichten uit PO's en wordt onder andere toegepast door de populaire Internet email applicaties voor PC's. POP server software is verkrijgbaar voor onder andere het UNIX besturingssysteem (public domain). Hiermee kan een UNIX PO (/var/mail) toegankelijk worden gemaakt via POP.

## 13.8 POP sessie 1

POP sessie 1

```
+OK UCB Pop server (version 1.6) at gatekeeper.osp.nl starting.
user josv
+OK Password required for josv.
pass <removed>
+OK josv has 29 message(s) (78205 octets).
list 29
+OK 29 470
retr 29
+OK 470 octets
Received: from noordpool (gd99-19.Gouda.NL.net [193.79.241.244]) by
    gatekeeper.osp.nl (8.6.11/1.2) with SMTP id RAA23304 for
    <josv@osp.nl>; Wed, 1 Jan 1997 17:07:27 +0100
```

**Notities**

Op deze en de volgende slide is te zien hoe een POP sessie verloopt. Ook deze sessie is weer handmatig uitgevoerd met het telnet commando (well known poort 110). De onderstreepte commando's zijn door mij ingetoetst, de overige tekst zijn de responses van de POP server. Het bericht wat hier met POP wordt opgehaald is hetzelfde bericht wat in het SMTP voorbeeld is verstuurd. Merk op dat de MTA in kwestie (sendmail) zelf aardig wat headers aan het bericht heeft toegevoegd.

## 13.9 POP sessie 2

POP sessie 2

Date: Wed, 1 Jan 1997 17:07:27 +0100
From: Kerstman@noordpool
Message-Id: <199701011607.RAA23304@gatekeeper.osp.nl>
Subject: Weersomstandigheden
Apparently-To: <josv@osp.nl>

Wegens een depressie boven de noordelijke ijszee dit jaar geen kadootjes.

.
quit
+OK Pop server at gatekeeper.osp.nl signing off.

**Notities**

## 13.10 POP eigenschappen

POP eigenschappen

Simpel

Authenticatie d.m.v. userid + password

Geen encryptie (password gaat cleartext over het netwerk!)

**Notities**

Net als SMTP staat POP niet bol van de luxe features. Zo worden de opgehaalde email berichten clear text over het tussenliggende netwerk verstuurd, met alle gevaren vandien. Wel moet een POP gebruiker zich met een userid en password bij de POP server authenticeren. Ook dit password gaat echter onversleuteld over het netwerk!

## 13.11 IMAP

IMAP

Internet Message Accesss Protocol
RFC 2060 (IMAP4rev1)
Veel uitgebreider:
- Mailbox beheer
- Berichten beheer
- Selectief ophalen van berichten

Weinig toegepast

**Notities**

Een wat nieuwer protocol op dit gebied is IMAP, het Internet Message Access Protocol. IMAP lijkt qua structuur veel op POP, maar is veel uitgebreider. Zo ondersteunt IMAP onder andere het op afstand aanmaken en verwijderen van postbussen in PO's. Ook kunnen berichten (zonder ze op te halen) wiorden doorzocht op sleutelwoorden. IMAP wordt voor zover ik weet nog vrijwel nergens toegepast.

# Hoofdstuk 14

# Randverschijnselen

## 14.1 Randverschijnselen

Randverschijnselen

**Notities**

In deze module worden diverse onderwerpen op het gebied van Internet email behandeld.

## 14.2 Van email adres naar mail host

Van email adres naar mail host

Internet email adres: someone@somewhere
DNS lookup van ìsomewhereî:
  Eerst: MX record
  Dan: A record
  Anders: faling
ìsomewhereî hoeft geen host te zijn!
  premier@catshuis.nl
  josv@localhost

**Notities**

Één van de vragen die we nog niet aan de orde hebben gesteld is: “hoe weet een MTA naar welke MTA een bericht moet worden gestuurd?”. Tot nu toe hebben we steeds gesteld dat een MTA een bericht krijgt aangeleverd en dat deze weet wat ermee dient te gebeuren. Internet mail routering wordt gestuurd door:

1. De configuratie van de MTA
2. Het email adres
3. Informatie die zit opgeslagen in de Domain Name Server (DNS)

Het is meestal mogelijk om MTA’s te configureren om alle ontvangen email onmiddelijk door te sturen naar een andere MTA. Dit verschijnsel doet zich meestal voor in een netwerk met meerdere machines die ieder als MTA op zouden kunnen treden (meestal bij UNIX machines). In zo’n geval willen we waarschijnlijk alle wat meer intelligente email akties bij een centrale MTA (mail hub) beleggen, en moeten alle andere MTA’s ontvangen berichten naar de centrale MTA versturen.

De MTA die zich verantwoordelijk voelt voor de verzending kijkt vervolgens indringend naar het email adres van de geadresseerde. Deze is gesteld in het formaat *someone@somewhere*. Hierbij is *somehwere* een indicatie van een Internet host of domain naam. De MTA doet vervolgens een DNS lookup voor *somewhere*. Herinner: de DNS is de Internet brede gedistribueerde database waarin hostnamen en IP-adressen aan elkaar zijn gekoppeld. DNS’sen kunnen echter ook zogenaamde MX records bevatten. Deze MX records beschrijven naar welke mail hub email voor een bepaalde node moet worden gestuurd.

De MTA doet allereerst een MX lookup voor *somewhere*. Indien deze slaagt heeft de beheerder van *somewhere* gespecificeerd naar welke nodes de email voor *somewhere* moet worde gestuurd. In dat geval zet de MTA een SMTP sessie met de MTA van de mail hub op en verzend het bericht. Indien er geen MX records voor *somewhere* te vinden zijn doet de MTA een A record (IP adres) lookup. Indien deze slaagt is het IP-adres van *somehwere* bekend en wordt het bericht door middel van SMTP verzonden naar de MTA van *somewhere* zelf. Indien geen MX of A records van *somewhere* bekend zijn faalt de verzending.

Dit mechanisme betekent dus dat email voor een bepaalde node niet per definitie op die node hoeft te worden afgeleverd. Systeembeheerders kunnen met MX records in de DNS regelen dat email op andere nodes wordt afgeleverd. Dit vergemakkelijkt de implementatie van Internet email aanzienlijk. De *somewhere* in een email adres hoeft dus ook geen bestaande node te zijn. De DNS staat toe dat er MX records bestaan voor nodes waarvoor geen A records zijn gedefinieerd. Hiermee kunnen email adressen worden gebruikt die niet zijn gekoppeld aan de daadwerkelijke fysieke machines waarop de email binnenkomt en wordt verwerkt. Wel is het altijd noodzakelijk dat *somewhere* een QDN (Qualified Domain Name) van het eigen geregistreerde Internet domain is. Het email adres “josv@osp” heeft geen Internet brede geldigheid omdat de MTA’s op de DNS ondervraging voor ‘osp’ geen succes zouden boeken.

## 14.3 DNS records

DNS records

A records bevatten het IP-adres van een node

MX records verwijzen naar mail exchangers van een node:

- Meerdere MX records mogelijk (fallback mail exchangers)
- MX record kan voorkomen zonder bijpassend A record

**Notities**

MTA's gebruiken dus de Domain Name Server om uit te vinden naar welke MTA een email bericht moet worden gestuurd. Hiervoor gebruikt de MTA MX (Mail eXchanger) records en A (Address) records. Een MTA doet eerst een MX query. De DNS staat het toe om voor een node of domain meerdere MX records te specificeren. In dat geval worden de MX records gesorteerd naar afnemende prioriteit. De MTA probeert eerste het bericht te verzenden naar de mail exchanger met de hoogste prioriteit (met het laagste getal). Mocht deze om een of andere reden het bericht niet kunnen accepteren dan probeert de MTA de volgende mail exchanger uit de lijst en zo voort. Via dit mechanisme kunnen dus voor een node of domain fall back mail exchangers worden gespecificeerd. Indien de primaire MTA uitvalt verstuurt de rest van Internet de email naar de volgende mail exchanger uit de lijst. Deze kan worden geconfigureerd om de mail vast te houden of na verloop van tijd door te sturen naar de primaire mail exchanger. Veel Internet providers draaien op verzoek een fall back mailer voor de aangesloten klanten.

## 14.4 DNS records voorbeeld


**Notities**

Op de slide staan een aantal voorbeelden uit de DNS configuratie. DNS configuratie files hebben helaas een nogal obscuur formaat maar de essentie is hoop ik duidelijk. De configuratie van MTA en DNS kan nogwel eens gevoelig zijn. Verkeerd geconfigureerde programmatuur kan *mail loops* tot gevolg hebben. Hierbij dwalen berichten oneindig lang door Internet, hoppend van MTA tot MTA zonder ooit het einddoel te bereiken. Een hieraan verwant verschijnsel is dat van de *mail storm*. Dit komt nog wel eens voor als twee MTA's zijn geconfigureerd om berichten automatisch te bevestigen en ze de ontvangst van elkaars ontvangstbevestigingen gaan bevestigen.

De '@' in bovenstaande voorbeeld staat in de plaats van de zonenaam 'osp.nl'. Alle email voor *someone*@osp.nl wordt dus afgeleverd bij de node 'gatekeeper.osp.nl'. Mocht deze tijdelijk niet beschikbaar zijn dan wordt de email afgeleverd bij 'sun.osp.nl'.

## 14.5 Aliassen


**Notities**

Een functie van veel Internet email software zijn “aliassen”. Ik doel hier op de mogelijkheid om meerdere email adressen af te beelden op één of meer postbussen. Zo wijzen de email adressen ‘jos@osp.nl’, ‘Jos.Visser@osp.nl’ en ‘J.C.W.Visser@osp.nl’ allemaal naar één en dezelfde postbus (namelijk die van ‘josv@osp.nl’). Aliassen zijn doorgaans een functie van de MTA of van de PO.

Speciale alias functies zijn die van de groepsaliassen en programma-aliassen. Bij groepsaliassen wordt een email adres afgebeeld op meerdere postbussen. Hierdoor wordt een email bericht in meerdere postbussen gestopt. Het email adres ‘mt@osp.nl’ is bijvoorbeeld afgebeeld op ‘erik@osp.nl’, ‘frans@osp.nl’ en ‘josv@osp.nl’. De MTA/PO software zorgt er voor dat het bericht meerdere keren wordt afgeleverd. Sommige MTA software is zelfs in staat om naar aanleiding van een alias conversie een bericht door te sturen naar een andere MTA.

Programma-aliassen is de mogelijkheid van MTA/PO software om een binnenkomend bericht te laten verwerken door een programma. Met name de UNIX/Internet MTA sendmail is hierin een ster. Op basis van deze faciliteit zijn een groot aantal programma’s ontwikkeld, de zogenaamde *mail robots*. Met mail robots kunnen bijvoorbeeld ‘mailing lists’ worden onderhouden. Mailing lists zijn applicaties waarbij emailtjes naar de mailing list automatisch worden doorgestuurd naar alle leden van de mailing list. De bekendste mailing list robots zijn ‘listserv’ en ‘majordomo’. Mail robots zijn extreem gevoelig voor beveiligingsproblemen.

## 14.6 Forwarding

Forwarding

Functie van MTA of PO
Bijvoorbeeld in UNIX:
$HOME/.forward file
Resent- headers

**Notities**

Een aan aliassen verwant onderwerp is ‘forwarding’. Hiermee bedoelen we het doorsturen van berichten naar één of meer ander email adressen. Doordat de meeste UNIX als kant-en-klare MTA/PO’s worden afgeleverd is iedere UNIX machine meestal een mail hub. Dit zou betekenen dat ik op iedere UNIX machine waarop ik een userid heb (vele tientallen) ik email zou kunnen ontvangen. Om te voorkomen dat ik iedere ochtend vele machines langs moet om email te lezen configureer ik iedere machine om mijn email door te sturen naar een centraal verzamelpunt. In UNIX kan dit tamelijk eenvoudig door het aanmaken van een ‘.forward’ bestand in mijn home directory. De inhoud van het ‘.forward’ bestand zijn de email adressen waarnaartoe de email moet worden doorgestuurd.

Het grote verschil tussen aliassen en forwarding is dat forwarding meestal door de individuele gebruiker kan worden geactiveerd, terwijl aliassen meestal door de mail beheerder moeten worden aangemaakt. Sommige MTA’s voorzien doorgestuurde post van aparte ‘Resent-’ headers om aan te geven dat het bericht is doorgestuurd (zie RFC 822).

## 14.7 Adres herschrijving

Adres herschrijving

Functie van MTA
De kampioen: sendmail (*by far*)
Interne adresconventies

**Notities**

Een andere feature die nog wel eens door MTA's wordt geboden is adres herschrijving. Met name de UNIX Internet mail MTA sendmail is hierin de absolute kampioen. Adres herschrijving is het concept dat de MTA de email adressen van afzender en geadresseerden wijzigt alvorens het bericht voor nadere verzending in overweging te nemen. Dit is bijvoorbeeld handig om interne adresconventies af te laten dwingen door de MTA.

Neem bijvoorbeeld een organisatie met een aantal UNIX machines. Normaal gesproken fungeert iedere UNIX machine als MTA en PO. Als ik een mailtje componeer en verstuur met een van de beschikbare MUA's op de UNIX machine 'arena' dan zal de UNIX MUA de afzender normaal gesproken noteren als 'josv@arena' of 'josv@arena.myorg.nl'. Als de ontvanger van het bericht met zijn/haar MUA een antwoord componeert dan wordt dit antwoord geadresseerd an 'josv@arena' of 'josv@arena.myorg.nl'. De problemen hiervan zijn meerledig:

1. Het bericht zal worden afgeleverd bij de machine 'arena'. Als deze UNIX machine wordt vervangen kan het antwoord dus niet worden afgeleverd. De meeste organisaties willen echter alle binnenkomende email op één centralemail server binnen laten komen. Dit kan ook met MX records worden afgedwongen, maar dat heeft tot gevolg dat de MX records ook moeten blijven bestaan nadat de machine is afgevoerd.

2. Het email adres 'josv@arena' kan niet buiten het myorg.nl domain worden geınterpreteerd.

Veel organisaties hebben als conventie dat email adressen voldoen aan de standaard 'someone@domain', dus in dit geval 'someone@myorg.nl'. Één van de manieren om

dit te realiseren is het herschrijven van de afzender adressen door de MTA. Deze dient dan de afzenders aan te passen aan de adresconventies. Hierdoor lijkt alle mail van deze organisatie afkomstig uit ‘myorg.nl’, ongeacht van welke machine ze uiteindelijk zijn verstuurd.

Een ander voorbeeld van adresherschrijving is het gebruik van geneste adressen. In een aantal gevallen willen we het wellicht toch mogelijk maken om email af te laten leveren bij een andere machine dan de centrale mail server. In dat geval kunnen we bijvoorbeeld dergelijke mail adresseren aan ‘josvbericht afgeleverd bij de MTA van myorg.nl (op basis van de MX of A records voor myorg.nl). We kunnen deze MTA configureren om dit adres te herschrijven naar ‘josv@arena’ en dan door te sturen.

## 14.8 MIME

MIME9u9◊

Multipurpose Internet Mail Extensions
RFCís 1521 en 1522
Insluiten non-ASCII data in Internet email:
- Plaatjes
- Geluid
- non-ASCII karakter sets (b.v. ISO 8859-1)

Uitbreiding RFC 822

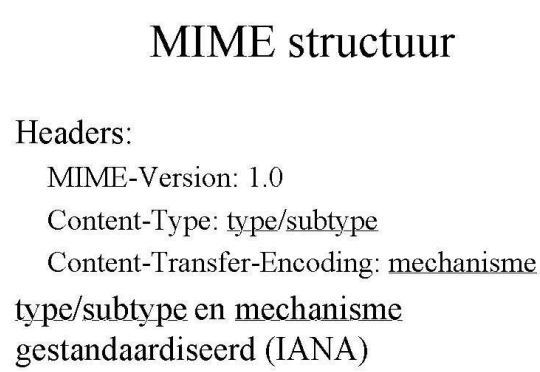
**Notities**

Een behoorlijke beperking van de traditionele RFC 822 berichtenstandaard is dat er in principe alleen maar ASCII (7 bit) teksten in de body van het bericht kunnen worden meegestuurd. Moderne hard- en software hebben het mogelijk gemaakt om zonder problemen ook andere dan tekstuele data te manipuleren (multimedia). Het is dan natuurlijk gewenst om deze niet ASCII data ook via Internet email te kunnen versturen.

Vroeger werd dit meestal gedaan met het ‘uuencode/uudecode’ protcol. ‘uuencode’ is een UNIX commando waarmee een binair bestand kan worden omgevormd tot een transporteerbaar tekstbestand. Uuencode doet dit door de binaire data om te vormen tot leesbare ASCII karakters. Hierdoor wordt het bestand aanzienlijk groter omdat er gemiddeld meer dan één ASCII karakter nodig is om een byte te coderen. De tegenhanger van uuencode is uudecode. Het uudecode commando voert de omgekeerde bewerking uit en herstelt het originele bestand. Het nadeel van uuencode/uudecode is dat het handmatige bewerkingen zijn die niet in de MUA zijn geıntegreerd.

Heden ten dage wordt voor het versturen van niet-ASCII data per Internet email meestal gebruik gemaakt van meer recente uitbreidingen aan de email standaarden. Deze uitbreidingen staan bekend onder de naam MIME: Multipurpose Internet Mail Extensions. MIME is gestandaardiseerd in RFC 1521 en 1522. Nieuwe MIME toepassingen (MIME types) worden in eigen RFC’s gestandaardiseerd. MIME maakt het mogelijk om in de body van een email bericht allerlei non-ASCII data in te sluiten. De MIME standaard is zodanig opgesteld dat MIME berichten via standaard SMTP kunnen worden verstuurd. De MIME standaard is een berichtenstandaard en vereist dus geen uitbreidingen aan het transportprotocol.

## 14.9 MIME structuur




**Notities**

MIME introduceert een aantal nieuwe RFC 822 headers. De aanwezigheid van deze headers vertelt de MUA dat er sprake is van een MIME bericht. De ‘MIME-Version’ header is hiervoor de belangrijkste aanwijzing; deze geeft aan volgens welke versie van de MIME standaard het bericht is opgemaakt. Een MIME bericht bevat ook een ‘Content-Type’ header. Deze header geeft aan uit wat voor soort data de body bestaat. Een compleet MIME type bestaat uit een type, een subtype en optioneel een aantal parameters voor het type. De MIME compatible MUA interpreteert het type en geeft de data op en geschikte wijze weer.

Aangezien MIME berichten met standaard MTA’s moeten kunnen worden verscheept moet de body op een bepaalde manier worden bewerkt voordat het aan de MTA kan worden aangeboden. De header ‘Content-Transfer-Encoding’ geeft aan met welk mechanisme de data is omgevormd tot transporteerbare tekst. De ontvangende MUA dient deze bewerking om te keren voordat de orginele (non-ASCII) data weer bruikbaar is.

MIME types en ‘transfer encodings’ zijn Internet breed gestandaard bij de Internet Assigned Numbers Authority (IANA).

## 14.10 MIME header voorbeelden

MIME header voorbeelden

Content-Type:
- text/plain; charset=charset
- audio/wav
- text/html
- image/eps
- multipart/mixed; boundary=aboundarystring

Content-Transfer-Encoding:
- base64
- 8bit

**Notities**

Op de slide staan een aantal voorbeelden van MIME types en transport coderingen. Een MIME compatible MUA interpreteert deze en kan dus de ingesloten data op een van toepassing zijnde manier aan de gebruiker tonen. Indien een bepaald MIME type niet kan worden ondersteund (bijvoorbeeld door het ontbreken van bepaalde hardware mogelijkheden zoals graphics, geluid of video) slaat de MUA de data meestal op in een disk bestand.

Een interessant MIME type is ‘multipart/mixed’. Dit type geeft aan dat er in de body van het bericht meerdere MIME gecodeerde componenten aanwezig zijn. De MUA heeft dan de verplichting om deze onderdelen uit de body te vissen en te presenteren.

Het bekendste MIME transportmechanisme is ‘base64’. Dit is een methode voor het (de)coderen van binaire data waarbij voor iedere 6 bits uit de binaire data een ASCII karakter in de reeks ‘A-Za-z0-9+-’ wordt gesubstitueerd. De base64 methode wordt ook gebruikt door andere Internet standaarden voor het coderen van binaire data.

## 14.11 MIME aandachtspunten

MIME aandachtspunten

Speciale MUA benodigd:
- Netscape mail
- PINE
- Eudora

Benodigde bandbreedte

Beveiliging
- Message/External-Body; access=ftp
- Application/subtype (krachtige interpreters)

**Notities**

Om van MIME gebruik te kunnen maken is een MIME compatible MUA nodig. De meeste PC MUA's ondersteunen MIME volledig. Ook voor UNIX en andere besturingssystemen zijn MIME compatible MUA's voorhanden. Email gateways dienen ook MIME te ondersteunen voor het heen en weer converteren van 'attachments' en ingesloten objecten.

MIME maakt het mogelijk om grote hoeveelheden data gemakkelijk in te sluiten en per email te verzenden. De hiervoor benodigde Internet bandbreedte kan aanzienlijk oplopen. Denk hierbij niet alleen aan netwerkcapaciteit maar ook aan de benodigde disk ruimte op de MTA's.

Bepaalde MIME types zijn gevoelig voor misbruik. Zo is er een MIME type 'message/external-body' wat aangeeft dat de eigenlijke body van het bericht apart door de MUA moet worden opgehaald. De parameters van dit type vertellen met welke methode de body moet worden opgehaald (bijvoorbeeld ftp). Als de MUA dit automatisch doet bestaat de kans dat een snoodaard een MIME bericht verstuurt met daarin de opdracht om automatisch een bestand op te halen en op de hard disk op te slaan. Dit is natuurlijk een prachtige manier om virussen te verspreiden. Andere MIME types worden door zeer krachtige interpreters verwerkt. Deze staan soms het opnemen van commando's en macro aanroepen in de data toe (Postscript, Word documenten).

# Hoofdstuk 15

# Email beveiliging

## 15.1 Email beveiliging

Email beveiliging

**Notities**

In deze module gaan we in op de diverse beveiligingsaspecten van Internet email.

## 15.2 Beveiligingsproblemen

Beveiligingsproblemen

Denial of service
Privacy
Integriteit
Authenticatie
Fysiek

**Notities**

Gedurende de shop is al meerdere keren aangestipt dat de Internet email standaarden niet zijn ontwikkeld voor zekere en veilige berichtenuitwisseling. Op de slide staan de verschillende categorieën beveiligingsaandachtspunten. In de komende slides worden deze stuk voor stuk behandeld.

Een speciale categorie beveiligingsproblemen wordt veroorzaakt door bugs in, en problemen met, de email software. Met name sendmail staat erom bekend dat het een goudmijn aan beveiligingsproblemen vormt. De enorme complexiteit van sendmail maakt het vrijwel onmogelijk om dit pakket goed en foutvrij te configureren. Deze problemen worden niet veroorzaakt door fouten in de standaarden maar zijn een gevolg van de implementatie ervan.

## 15.3 Denial of service

Denial of service

Mail bomb

Vullen van het PO door zenden grote berichten

Oplossing: in MTA (b.v. SSMTP)

Weinig MTAís ondersteunen dit

**Notities**

Een bekend Internet verschijnsel is de ‘mail bomb’. Een mail bomb is een bericht van enkele megabytes groot wat naar een MTA wordt gestuurd. MTA’s beperken meestal de grootte van berichten niet, met als gevolg dat de disks van de MTA en de PO al snel vollopen wanneer meerdere bommen worden verstuurd. Zodra de disks vol zitten is de MTA/PO niet meer in staat om berichten te ontvangen of te versturen. Zich misdragende Internet gebruikers worden door hun medegebruikers nogal eens met een mail bombardement ‘gestraft’. Dit overkwam bijvoorbeeld een Amerikaans advocatenkantoor wat tegen de regels in adverteerde in USENET nieuwsgroepen.

De oplossing voor dit probleem moet worden gezocht in de MTA. Slechts weinig MTA’s ondersteunen echter faciliteiten op dit gebied.

## 15.4 Privacy

Privacy

Email wordt clear text:
- verstuurd van MTA naar MTA (door SMTP)
- opgeslagen in PO
- verstuurd van PO naar MUA (POP)

Oplossing:
- encryptie (PEM of PGP)
- SSMTP

**Notities**

Een ander bekend probleem van Internet email is dat de berichten clear text worden verstuurd en opgeslagen. Dit betekent bijvoorbeeld dat de systeembeheerders van de MTA's en PO's de berichten zonder problemen kunnen inzien. Wat minder voor de hand liggend, maar nog steeds mogelijk, is dat hackers die rechtstreeks op Internet verbindingen mee kunnen kijken (via packet sniffers) de inhoud van het bericht tot zich kunnen nemen.

De oplossing hier wordt geboden door het gebruiken van encryptie. Verreweg het beste is natuurlijk als het bericht 'end-to-end' wordt versleuteld. Bekende mogelijkheden hiervoor zijn PEM en PGP (worden verderop behandeld). Een andere (mindere) optie is om de communicatie tussen MTA's te versleutelen. Dit kan bijvoorbeeld door het SMTP protocol te vervatten in een SSL verbinding (SSMTP). De berichten worden dan nog steeds clear text door de MTA's en PO's opgeslagen, maar zijn beveiligd gedurende de communicatie tussen MTA's. SSMTP wordt nog niet breed gebruikt. te versleutelen

## 15.5 Integriteit

### Integriteit

Email kan eenvoudig worden aangepast terwijl het bericht:

- In de queue van de MTA staat
- In het PO is opgeslagen

Oplossing: Message Authentication Code

- PEM/PGP

**Notities**

Aangezien berichten clear text worden verzonden en opgeslagen is het relatief eenvoudig mogelijk om de berichten onderweg te veranderen, zonder dat afzender of geadresseerde dit merken. Met name tijdens het verblijf van het bericht in de queues van de MTA of de PO kan dit eenvoudig geschieden, doorgaans met een standaard tekst editor.

De oplossing hiervoor is natuurlijk om het bericht te beschermen met een digitale handtekening. De twee meest populaire Internet email beveiligings hulpmiddelen (PEM en PGP) ondersteunen dit.

## 15.6 Authenticatie

Authenticatie

Identiteit van de afzender van een bericht is niet gegarandeerd:

- Iedereen kan met telnet het SMTP protocol naspelen.
- Veel MUAís staan toe dat de afzender wordt gespecificeerd

Oplossing: signering

- PEM/PGP

**Notities**

Vervolgens is het normaal gesproken in Internet nooit zeker wie de afzender van een bericht is. In een aantal MUA's kan de afzender bijvoorbeeld worden geconfigureerd door de gebruiker. Daarnaast is het vrijwel altijd mogelijk om met de hand het SMTP protocol uit te voeren. Ook dit probleem kan worden opgelost door het inzetten van PEM of PGP.

## 15.7 Fysiek

Fysiek

Geen ontvangstbevestiging
Vergelijk: postduif
Gevaren:
disk crash
system crash
Oplossing:
???

**Notities**

Het laatste te noemen probleem is de fysieke bescherming van de email. Internet email werkt op basis van onderling communicerende MTA's die verder niets van elkaar weten. Zodra een bericht (via SMTP) bij de volgende MTA in de route is aangekomen, vergeet de verzendende MTA dat hij het bericht heeft behandeld. Er vindt nooit meer een synchronisatie tussen de componenten (MUA, MTA, PO) plaats om te verifiëren of een bericht is aangekomen. Indien een MTA of PO 'crasht' gaat doorgaans ieder spoor van de daar opgeslagen berichten verloren. Noch de verzender, nog de ontvanger, is zich hier van bewust.

Voor dit probleem is er geen echte oplossing. De email standaarden voorzien immers niet in een mechanisme van ontvangstbevestigingen tussen afzender en geadresseerde. Sommige MUA's ondersteunen dit wel, maar dit kan ongewenste gevolgen hebben zoals een mail explosie. Bij de inrichten van een mail server dient dit fysieke aspect de nodige aandacht te krijgen. Een infrastructuur met verhoogde beschikbaarheid (fall back systemen, mirroring) is geen overbodige luxe.

## 15.8 PEM

PEM$f\Diamond 9$

Privacy Enhanced Mail
RFC 1421 t/m 1424
Ondersteunt:

| | |
|---|---|
| Encryptie | *Confidentiality* |
| Authenticatie | *Authentication* |
| Integriteit | *Message Integrity* |

Geimplementeerd in de MUA

**Notities**

Momenteel heersen in Internet twee standaarden voor het beveiligen van email: PEM en PGP. Het wonderbaarlijke is dat PEM de offiële Internet standaard is, maar dat PGP het meest wordt gebruikt.

PEM staat voor Privacy Enhanced Mail, en is dus de officiële Internet standaard voor encryptie, en signering van email berichten. De PEM standaard is vastgelegd in RFC 1421 tot en met 1424. PEM ondersteunt encryptie van berichten en het signeren van berichten om de integriteit te beschermen en de afzender te bewijzen.

Om van PEM gebruik te maken is een PEM compatible MUA benodigd. PEM is momenteel niet breed geımplementeerd. Er zijn twee referentie implementaties: TIS/PEM en RIPEM.

## 15.9 PEM eigenschappen

PEM eigenschappen

PEM compatible MUA benodigd

PEM certificaten (X.509-achtig)

RFC 934 encapsulatie:

-----BEGIN PRIVACY-ENHANCED MESSAGE -----

Ondersteunt diverse encryptie en signerings protocollen

Implementaties: TIS/PEM en RIPEM

**Notities**

PEM implementaties signeren en encrypten het bericht en vormen het vervolgens weer om tot leesbare tekst (bijvoorbeeld met base64). Dit bericht wordt dan voorzien van een RFC 934 compatible header en op de standaard wijze (via SMTP) verzonden. De ontvangende MUA herkent dat het een PEM bewerkt bericht is, interpreteert de RFC 934 headers, decodeert het bericht, controleert de handtekening en toont het bericht vervolgens aan de gebruiker. De PEM standaard ondersteunt diverse algoritmes voor encryptie en signering (waaronder DES en RSA).

Om dit soort bewerkingen mogelijk te maken is het noodzakelijk dat afzender en geadresseerde één of meer sleutels met elkaar delen. Het kan hier gaan om geheime sleutels voor symmetrische algoritmes (zoals DES) of de publieke sleutels van public key algoritmes. Één van de PEm RFC's is geheel gewijdt aan het management van de sleutels en de certificaten (X.509 compatible) waarmee die sleutels worden verspreidt.

Het belangrijkste nadeel van PEM is dat het slechts mondjesmaat is geımplementeerd. In principe ondersteunt geen van de populaire MUA's van dit moment de PEM standaard. De belangrijkste boosdoener hierbij is de Amerikaanse ITAR (International Trade in Arms Regulation) wetgeving die export van encryptieprogrammatuur expliciet verbiedt zonder vergunning.

## 15.10 PGP

PGP

Pretty Good Privacy
Phil Zimmerman (*my hero*)
Niet RFC gestandaardiseerd!
Wijd verbreid
Ondersteunt:
Encryptie, Authenticatie, Message ntegrity
Geimplementeerd als los programma

**Notities**

De andere grote speler in de email encryptie wereld is PGP. In tegenstelling tot PEM is PGP niet gestandaardiseerd met RFC's, het is een de facto standaard gebaseerd op de wereldwijde acceptatie van het PGP programma.

PGP is een algemeen encryptieprogramma wat is ontwikkeld door Phil Zimmerman. PGP maakt het mogelijk om bestanden (en dus ook email berichten) te encrypten en te signeren. PGP is ontwikkeld in de Verenigde Staten en via ftp (per ongeluk) ook in het buitenland terecht gekomen. Vanwege deze verboden export van encryptieprogrammatuur heeft Zimmerman jarenlang ruzie gehad met de Amerikaanse overheid. Pas eind 1996 heeft het Amerikaanse ministerie van justitie hem ontslagen van rechtsvervolging. Verder maakt PGP gebruik van in de U.S.A. gepatenteerde encryptietechnologie (het RSA algoritme). Vanwege deze patentinbreuk heeft hij jarenlang overhoop gelegen met RSA Data Inc., de patenthouder. Ook deze ruzie is inmiddels beslecht door een overeenkomst tussen Phil Zimmerman en RSA Data Inc.

Al deze problemen hebben PGP er niet van weerhouden de wereld snel te veroveren. De eenvoud en kracht van het programma hebben velen ertoe overgehaald het programma te gebruiken voor het beveiligen van bestanden en correspondentie. PGP is momenteel niet geımplementeerd in de populaire MUA's. Het is een los programma en het bewerken van berichten met PGP dient meestal buiten de MUA om te geschieden. Het programma is buiten de U.S.A. public domain software en beschikbaar voor alle denkbare platformen.

## 15.11 PGP eigenschappen

PGP eigenschappen

Gebruikt IDEA en RSA algoritmes
Geen certificaten: *Web Of Trust*
RFC 934 encapsulatie:
-----BEGIN PGP SIGNED MESSAGE -----
Verschillende implementaties:
Internationaal
Met/zonder RSAREF
U.S. domestic

**Notities**

PGP combineert symmetrische en ublic key encryptie met elkaar. Versleuteling vindt plaats met het IDEA algoritme. Dit is een symmetrisch encryptiealgoritme wat werkt met sleutels van 128 bits. Voor de versleuteling wordt een random 128 bits sleutel gegenereerd waarmee het IDEA algoritme wordt gevoerd. De sleutel zelf wordt met de public key van de ontvanger versleuteld (RSA) en met het bericht meegestuurd. De ontvanger ontsleuteld eerst met zijn private key de 128 bits sessiesleutel en gebruikt die vervolgens om het eigenlijke bericht weer leesbaar te maken (met IDEA). De sterkste versies van PGP ondersteunen 2048 bit public key encryptie. In combinatie met de 128 bits IDEA encryptie levert dit het beste algemeen verkrijgbare encryptieprogramma.

Door al het geharrewar rondom patenten en export beperkingen zijn er per PGP versie diverse varianten in omloop. Er is een internationale versie (geen beperkingen), een beperkte Amerikaanse public domain versie die alleen voor niet commerciële doeleinde mag worden gebruikt (maakt gebruik van de RSAREF toolkit van RSA Data Inc., maximaal 512 bits public key encryptie) en een Amerikaanse commerciële versie zonder beperkingen. Deze laatste wordt verkocht in de V.S. door ViaCrypt Inc.

Een opvallend aspect van PGP is dat het programma geen gebruik maakt van centraal uitgegeven certificaten (zoals PEM en SSL). Om de echtheid van PGP public key certificaten in te schatten kunnen gebruikers elkaars certificaten signeren. Hierdoor ontstaat een *Web of Trust* waarin gebruikers zelf aan kunnen geven in hoeverre ze andere gebruikers vertrouwen en parallel daaraan, in hoeverre ze certificaten accepteren die door die andere gebruikers zijn gesigneerd.

## 15.12 Email encryptie


**Notities**

Momenteel zijn er dus twee opties voor Internet email gegevensbeveiliging: produkten gebaseerd op PEM en PGP. Een nadeel van PEM is dat het nauwelijks geımplementeerd is. Het belangrijkste nadeel van PGP is dat het niet handig in gebruik is omdat het een extern programma betreft wat buiten de MUA om moet worden gebruikt.

# Deel IV

# De techniek achter het World Wide Web

# Hoofdstuk 16

# Introductie

## 16.1 Welkom

# Internet Info Shop 4

De techniek achter het
World Wide Web

**Notities**

Welkom bij de vierde Internet Information Shop over het World Wide Web. In deze shop gaan we dieper in op de oorsprong en werking van het World Wide Web. Van de cursisten wordt verwacht (verondersteld) dat ze gebruikerservaring hebben met het web. Op zaken als ”hoe gebruik ik het webën ”hoe volg ik een hyperlink”wordt niet nader ingegaan.

Ook is deze info shop geen cursus in het programmeren van documenten in HTML of het schrijven van CGI scripts.

## 16.2 Inhoud

Inhoud



**Notities**

De inhoud van de cursus is gericht op automatiseerders en andere technisch c.q. inhoudelijk betrokken medewerkers die willen weten hoe het World Wide Web is ingericht. We gaan in op de structuur van het web, de werking van HTML en HTTP, hoe web applicaties in elkaar zitten, hoe “levende” WWW-pagina’s kunnen worden gemaakt en de beveiligingsaspecten van het web. Tenslotte komen nog een aantal aanverwante onderwerpen aan de orde.

# Hoofdstuk 17

# Inleiding

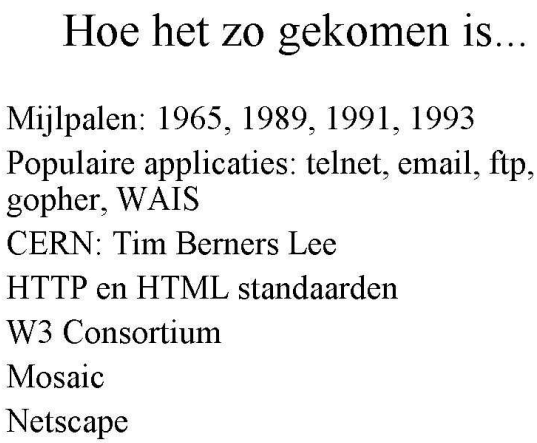
## 17.1

# Inleiding

**Notities**

Welkom bij de inleidende module van deze info shop. In deze module beschrijven we (wellicht ten overvloede) kort en bondig waar het World Wide Web vandaan komt en wat de grondslagen van het web zijn.

## 17.2 Hoe het zo gekomen is...




**Notities**

Het idee van een op hypertekst gebaseerde informatiedienst is ouder dan je wellicht zou denken. Al in 1945 werd een systeem beschreven (MEMEX) voor het opslaan en toegankelijk maken van informatie via associatieve indexen. In 1965 beschreef Ted Nelson het "Xanadu" systeem: een universum van onderling verbonden documenten voor het toegankelijk maken van informatie.

In 1989 begon de aan CERN verbonden Tim Berners Lee met de ontwikkeling van standaarden voor het electronisch publiceren van wetenschappelijke rapporten. In dergelijke rapporten zitten traditiegetrouw enorm veel verwijzingen naar andere rapporten. Het systeem van Tim moest het mogelijk maken van van het ene document naar het andere te springen zonder dat de gebruiker door had waar die documenten uithingen en hoe ze daar heetten. Deze standaarden, HTTP en HTML, werden, zoals in Internet gebruikelijk is, aan het grote publiek bekend gemaakt. De populariteit van dit "World Wide Web" nam met sprongen toe toen de programmeurs van het "National Centre for Supercomputing Application" (NCSA) een uiterst fraaie en gebruikersvriendelijke client voor het World Wide Web ontwikkelden en in het public domain stopten. Deze client (Mosaic) verwierf in "no time" een grote schare volgelingen. Het was ook eigenlijk het eerste gebruikersvriendelijke programma voor het gebruik van Internet.

Marc Andreesen, het brein achter Mosaic, richtte samen met enkele anderen het bedrijf Netscape Communication op. Netscape ontwikkelde in vervolg op Mosaic de bekende Netscape Navigator browser. De kracht en flexibiliteit van Netscape verleidde nog eens vele duizenden tot het inrichten van World Wide Web sites en het gebruik van het web. De hieruit voortvloeiende hausse op het gebied van het web heeft de populariteit van Internet tot ongekende hoogte opgestuwd. Voor een groot aantal gebruikers is het

Internet gelijk aan het World Wide Web!

## 17.3 Het World Wide Web

Het World Wide Web

ìA wide-area hypermedia
information retrieval initiativeî

**Notities**

Wat is het World Wide Web?

Deze vraag is niet eenvoudig te beantwoorden. Het hedendaagse web is veel verschillende dingen voor veel verschillende mensen. De definitie van het W3 Consortium staat op de slide.

## 17.4 Het World Wide Web (again)

Het World Wide Web

Een gedistribueerde hypertext informatieservice

Een interactieve online transactieservice

Toepassingen:

- Electronische informatiediensten
- Web applicaties, bijvoorbeeld
  - Electronische winkels
  - Raadplegen databases

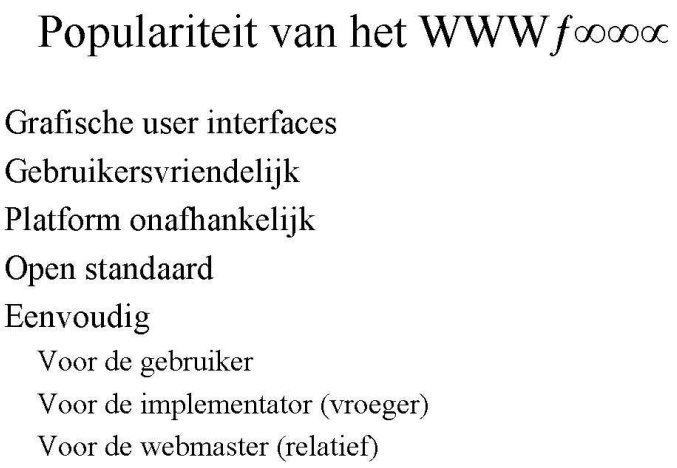
**Notities**

De twee belangrijkste functies van het web zijn:

1. Een gedistribueerde hypertext informatieservice
   De mogelijkheden van het World Wide Web zijn veelvuldig toegepast voor het inrichten van electronische informatiediensten. Voorbeelden hiervan zijn electronische catalogi, online handleidingen en electronische folderstands. Het gemak waarmee een WWW-site kan worden ingericht heeft ervoor gezorgd dat niet alleen grote professionele organisaties Web sites hebben ingericht. Kleine bedrijven, stichtingen en privé personen zijn ook grootscheeps overgegaan tot het opzetten van zogenaamde ‘home pages’. De hoeveelheid beschikbare informatie op het WWW is daardoor zeer gigantisch.

2. Een interactieve online transactieservice
   Met de standaard mogelijkheden van het World Wide Web kunnen op eenvoudige wijze simpele transactiemogelijkheden worden geïmplementeerd. (Complexe interactie is ook mogelijk, maar is minder simpel.) Op basis van deze faciliteiten zijn er al een groot aantal sites waar niet alleen informatie kan worden opgehaald, maar waarmee online interactief mee kan worden gecommuniceerd. Zo kunnen bezoekers bijvoorbeeld online een spel spelen, hun naam achterlaten (voor het per landpost opsturen van meer informatie) of een bestelling doen. Op basis van deze faciliteiten zijn ondertussen al een aantal electronische winkels gebouwd.

## 17.5 Populariteit van het WWW




**Notities**

Het World Wide Web is in zeer korte tijd ongekend populair geworden en heeft in zijn kielzog het Internet meegesleurd in de vaart der volkeren. Ondanks het feit dat het WWW slechts één van de vele mogelijke Internet applicaties is (en een zeer recent ontwikkelde) denken een groot aantal gebruikers dat Internet en het World Wide Web synoniemen van elkaar zijn.

De populariteit van het World Wide Web valt uit de volgende factoren te verklaren:

- Grafische gebruikers interface
  Het WWW is één van de weinige applicaties met een fraai grafisch gebruikersinterface. Oudere Internet applicaties waren ook al eens voorzien van een grafisch interface, maar in dit geval blijft het altijd een oude tekst geörienteerde applicatie die van een grafisch jasje is voorzien.

- Gebruikersvriendelijk
  De World Wide Web client software is zeer gebruikersvriendelijk van aard. Dit is een grote tegenstelling met de traditionele Internet client programmatuur die eigenlijk alleen door automatiseerders kan worden gebruikt. Het WWW is eigenlijk de eerste eindgebruikersgerichte applicatie in het Internet.

- Platformonafhankelijk
  Het World Wide Web is platformonafhankelijk. Het maakt absoluut niet uit met wat voor soort computer of besturingssysteem de client en de server zijn uitgerust. In het heterogene Internet waar een grote verzameling hard- en software rondtolt is dit een groot voordeel. De toegankelijkheid van het World Wide Web

is hierdoor zeer goed. Het maakt niet uit wat voor soort computer of OS je hebt, er is altijd wel een client of server software pakket te vinden.

- Open standaard
  Het World Wide Web is gebaseerd op open en goed gedefinieerde en toegankelijke standaarden. Hierdoor kan er vrij eenvoudig nieuwe client en server programmatuur worden ontwikkeld.

- Eenvoudig
  Het World Wide Web is eenvoudig. Gebruikers die al enige ervaring hebben met moderne grafische applicaties behoeven geen nadere toelichting bij het gebruik van hun WWW client. Ook het opzetten van een WWW site is tamelijk eenvoudig. De hiervoor benodigde software is goed verkrijgbaar en goedkoop (meestal public domain). Een groot aantal Internet providers biedt haar klanten de mogelijkheid om één of meer pagina's op het Web te publiceren. Hierbij heeft de provider de benodigde infrastructuur geregeld en hoeven de gebruikers alleen nog maar de pagina's te bedenken. De programmeertaal waarin deze pagina's moeten worden ontwikkeld is ook redelijk simpel.

  Het schrijven van WWW client- en server software was met name vroeger ook niet al te moeilijk. De toenemende concurrentie tussen Microsoft en Netscape heeft er echter voor gezorgd dat de moderne WWW clients ongekend krachtig zijn en vol met features zitten. Het ontwikkelen van een nieuwe client is hierdoor een niet-triviale aangelegenheid geworden.

## 17.6 Organisatie van het web

Organisatie van het web

W3 Consortium
Internet Architecture Board (RFCís)
Leveranciers:

| | |
|---|---|
| Microsoft | VBScript, ActiveX |
| Sun | Java |
| Netscape | JavaScript, SSL |
| TIS | S-HTTP |

**Notities**

De standaardisering van het web is momenteel in handen van drie groepen (instanties):

1. Het W3 consortium
   Dit is een groep instellingen, personen en bedrijven die onder aanvoering van het MIT het gebruik en de standaardisering van het World Wide Web willen bevorderen. Zij bedenken nieuwe standaarden en voorzien in referentie implementaties van die standaarden.

2. Het Internet Architecture Board
   Het IAB houdt zich bezig met de standaardisering van Internet technologie door middel van RFC's. Bepaalde onderdelen van het WWW zijn inmiddels al in een RFC vastgelegd.

3. Leveranciers
   Producenten van Internet software zijn momenteel erg actief in het verzinnen van nieuwe uitbreidingen en het implementeren van die uitbreidingen in hun produkten. Leveranciers als Sun, Netscape en Microsoft komen om de haverklap met verbeteringen die ze vervolgens ook weer ter beschikking stellen aan hun concurrenten in de hoop een Internet brede de facto standaard neer te zetten. De ontwikkelingen volgen elkaar momenteel zo snel op dat de traditionele standaardiserings trajecten (IAB/RFC) vaak niet de tijd hebben de nieuwe technieken formeel te standaardiseren.

## 17.7

HyperMedia

- Hypertext
  - Niet lineair
  - Zelfstandige teksteenheden
  - Teksteenheden onderling verbonden via *links*
- Hypermedia
  - Generalisatie van hypertext
  - Documenten met niet-tekstuele componenten
    - Plaatjes, geluid, invulvelden, video

**Notities**

De grondgedachte van het World Wide Web is gebaseerd op de begrippen *HyperMedia* en *HyperText*. Hypertext documenten bestaan uit onderling verbonden teksteenheden. Deze verbindingen (links) kunnen door gebruikers worden gebruikt om van de ene teksteenheid naar de andere te komen. Een normaal boek of artikel is *lineair*, het moet van voor naar achter worden gelezen. Hypertext is niet lineair. Iedere teksteenheid is een op zichzelf staande eenheid. Via de verbindingen kan van een teksteenheid naar een veelheid van andere teksteenheden worden gesprongen. Zo kunnen begrippen die niet geheel duidelijk zijn onmiddelijk worden verklaard door een verbinding op te nemen naar een teksteenheid die het begrip in kwestie verder uitdiept. Gebruikers lezen hypertext niet in de gebruikelijke zin des woords, ze navigeren door een universum van documenten (ook wel “docuverse” genoemd). Hierdoor kan de gebruiker de informatie tot zich nemen in de volgorde en op het detailniveau die aansluiten bij de kennis en ervaring van de gebruiker.

Het begrip “hypermedia” wordt in de literatuur gedefinieerd als een generalisatie van hypertext. In hypermedia documenten is niet alleen tekst opgenomen maar ook niet-tekstuele elementen zoals plaatjes, geluid en video. In de hedendaagse terminologie worden hypertext en hypermedia door elkaar gebruikt.

## 17.8

# Hyper

Node, document

(Hyper)link

Logische verbinding tussen twee documenten

(Hyper)web

Verzameling links tussen hypermedia document

Hyperdocument

Logische verzameling documenten en links

**Notities**

Een “docuverse” van hypermedia documenten kan door de onderlinge verbindingen worden gezien als een netwerk, of een web. De knooppunten in dit web zijn de documenten, ook wel “nodes” genoemd. De verbindingen tussen de nodes zijn de hyperlinks.

# Hoofdstuk 18

# De structuur van het web

## 18.1 De structuur van het web

De structuur van het web

**Notities**

In deze module gaan we in op de basisstructuur van het World Wide Web.

## 18.2 WWW Componenten

WWW Componenten

Client
- Browser

Server(s)
- HTTP, FTP, Telnet, Gopher, ...

Internet
- Communicatiemechanisme

**Notities**

Het World Wide Web is een client/server applicatie.

De gebruikers van het web draaien een WWW client applicatie (een zogenaamde browser). Deze browsers maken over het Internet contact met WWW server applicaties voor het ophalen van documenten en het opsturen van informatie. WWW browsers kunnen informatie ophalen van een groot aantal verschillende server applicaties. Naast de WWW-specifieke HTTP servers kunnen browsers ook informatie ophalen van FTP en Gopher servers. Hierdoor kunnen de gebruikersvriendelijke browsers ook worden gebruikt als user interface voor traditionele Internet applicaties. Browsers kunnen zelfs worden geïnstrueerd om telnet sessies met remote servers op te starten.

WWW Servers draaien server applicaties die op verzoek documenten en bestanden teruggeven aan WWW browsers. De server applicatie heeft meestal niet door of hij door een WWW client of door een andere client wordt benaderd. Het is de taak van de WWW client om het applicatieprotocol van de WWW server applicatie te praten.

## 18.3 WWW Transacties

WWW Transacties

Request/Response transacties

*Retrieval*

Browser haalt een document op bij een server

Protocollen: HTTP, FTP, Gopher

*Reply*

Browser stuurt informatie op naar een server

Invulformulieren

HTTP

**Notities**

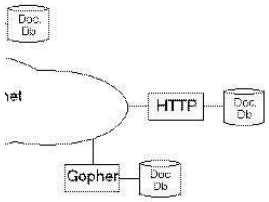
Het World Wide Web is een transactie geörienteerd medium. De client zet een verzoek uit bij een WWW server applicatie (*request*). De server beantwoord dit verzoek met het gevraagde document of met een foutmelding. Deze *response* wordt door de WWW client verwerkt. Meestal komt deze verwerking neer op het tonen van het opgehaalde document. De gebruiker kiest vervolgens één van de hyperlinks in het document, waarop de browser het volgende document ophaalt en zo voorts.

In principe kan een WWW client twee soorten transacties initiëren:

- Retrieval
  Met een “retrieval” transactie worden documenten van WWW servers opgevraagd. De client dient exact aan te geven waar het document huist, hoe het daar heet en met welk protocol het kan worden opgehaald. Het opgehaalde document wordt door de browser getoond.

- Submit
  Indien een document invulvelden bevat kan een gebruiker deze invullen en de ingevulde waarden terugsturen naar een WWW server. De WWW server verwerkt de opgestuurde waarden en produceert op basis daarvan weer een nieuw document. Met dit mechanisme kunnen onder andere electronische winkels en dergelijke in elkaar worden geknutseld. Momenteel ondersteunen alleen HTTP servers het opsturen van informatie. Gopher en FTP servers beperken zich tot het ophoesten van documenten op verzoek.

## 18.4 World Wide Web




**Notities**

Één van de kernmerkende eigenschappen van WWW is dat de gebruiker totaal het overzicht verliest van waar welke informatie vandaan komt. Dit komt omdat de verbindingen tussen hypermedia documenten vrij eenvoudig de grenzen van een server kunnen overschrijden. In het “adres” van een document (een zogenaamde URL) komt namelijk onder andere de hostnaam voor van de server waar een document thuishoort. De programmeurs van de WWW documenten kunnen bij het bouwen van de hypermedia pagina’s verbindingen opnemen naar ieder document op Internet. Dit gebeurt dan ook veelvuldig. Daarnaast zijn er gespecialiseerde “zoek-en-index” servers die er hun taak van maken om verwijzingen naar ieder WWW document op de planeet op te nemen. Hierdoor worden alle WWW servers van de wereld aan elkaar gekoppeld tot een wereldwijd web.

## 18.5 WWW Client

WWW Client

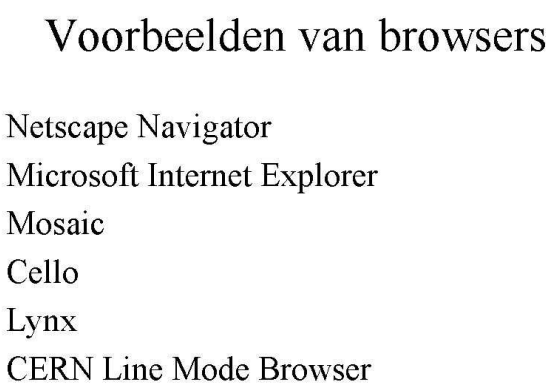
Browser
Gebruikersinterface
Haalt documenten op van servers
Geeft documenten weer
Invullen van documenten met invulvelden
Opsturen ingevulde formulieren

**Notities**

Iedere WWW eindgebruiker moet in het bezit zijn van een WWW client applicatie, de zogenaamde “browsers”. De eigenlijke taak van dergelijke browsers is eenvoudig: ze halen documenten op van WWW servers, geven deze weer en laten gebruikers met deze documenten interacteren (kiezen van hyperlinks, invullen van invoervelden, opsturen van een ingevuld formulier). De allereerste browsers waren inderdaad tamelijk eenvoudige applicaties. Uit de begintijd van Internet komt de opmerking: *“Every bozo can write a browser, a guy in our office wrote one in a week”*.

Vandaag aan de dag is dat zeker niet meer het geval. De concurrentie tussen grote software leveranciers (met name Microsoft en Netscape) heeft ervoor gezorgd dat browsers grote, complexe en featurevolle applicaties zijn geworden. De hedendaagse browser is groter dan de kernel van een operating system!

## 18.6 Voorbeelden van browsers




**Notities**

Op de slide staan voorbeelden van bekende (en minder bekende) browsers.

## 18.7 Eigenschappen van browsers


**Notities**

In de diverse standaarden is vrij goed vastgelegd waaraan browsers moeten voldoen. De implementatoren hebben echter behoorlijk wat vrijheid. Browsers verschillen met name in hun mogelijkheden voor het weergeven van niet-tekstuele elementen (plaatjes, video, geluid), ondersteuning van niet-standaard documenttypen (PostScript, Adobe Acrobat) en de wijze waarop ze zogenaamde "Active Content" ondersteunen. Het is absoluut niet voorspelbaar met wat voor soort browsers de gebruikers in Internet werken. Hierdoor moet de "webmaster" van een site goed nadenken voordat hij/zij features gebruikt die niet door alle browsers worden ondersteund. Er is niets zo vervelend als een web site die niet met de door jouw gebruikte browser kan worden bekeken. Vrij kenmerkend is dat verschillende browsers hetzelfde document op totaal verschillende wijze kunnen weergeven. Dit wordt niet veroorzaakt door verschillende interpretatie van de standaarden maar door een (toegestande) verschillende wijze van implementeren van de standaard. Veel webmasters hebben een slecht begrip over wat de standaarden op het gebied van WWW documentopmaak precies inhouden. Om mooiere documenten te maken als standaard mogelijk is worden alle registers opgetrokken en wordt er nogal eens gebruik gemaakt van *undocumented features*.

De browsers halen documenten op met een TCP verbinding (meestal door het Internet). Dit kan (met name bij grote documenten) een grote vertraging met zich meebrengen. Om deze reden leggen veel browsers een lokale cache van recent opgehaalde documenten aan. Zodra een gebruiker een document ophaalt kijkt de browser eerst in zijn lokale cache. Is het document daar nog/al aanwezig dan krijgt de gebruiker deze lokaal opgeslagen kopie te zien. Modernere browsers kunnen eerst nog een controle over Internet doen bij de WWW server om te kijken of het document intussen op de server verouderd is.

## 18.8 Documenttypen


**Notities**

De World Wide Web browsers kunnen verschillende typen documenten tonen. De wijze waarop een document wordt getoond is grotendeels afhankelijk van het documenttype. De typenaamgeving is afgeleid van de MIME standaard (RFC's 1521 en 1522). Bekende documenttypen zijn:

- text/plain
  Gewone ASCII tekst, wordt eenvoudigweg getoond.
- text/html
  Een document in de WWW documentopmaaktaal HTML. De browser interpreteert het document en toont het in opgemaakte vorm.
- image/gif, image/jpeg
  Plaatjes in een of andere codering. De browser toont het plaatje.

Andere bekende documenttypen zijn audio/wav en video/avi. Indien een browser een bepaald documenttype totaal niet begrijpt geeft hij een foutmelding op biedt hij aan het document op te slaan op de hard disk van de gebruiker.

## 18.9 Filetypes

Filetypes

Hoe weet een browser hoe een document af te beelden?

Opgehaald met FTP of Gopher:

- Filetype tabel
- Relatie filenaam -> MIME type

Opgehaald met HTTP:

- Server antwoord bevat MIME type

**Notities**

Om een document correct af te beelden moet een browser natuurlijk wel in staat zijn om te bepalen wat het documenttype van een document is. De wijze waarop hij dit doet is afhankelijk van de manier waarop het document is opgehaald:

- FTP, Gopher
  Bij bestanden die via FTP of Gopher worden opgehaald is de bestandsnaam bepalend voor het documenttype. De browser gebruikt de bestandsnaam als sleutel in een tabel van documenttypen. In deze tabel staat bijvoorbeeld vermeld dat bestanden waarvan de naam voldoet aan het patroon “*.doc” van het type application/msword zijn.
- HTTP
  Indien de documenten met het HTTP protocol worden opgehaald stuurt de HTTP server in het antwoord het MIME type mee. Onderdeel van het antwoord is de zogenaamde “Content-Type” header welke het MIME type van het meegestuurde document bevat.

## 18.10 WWW Server

WWW Server

Stuurt op verzoek een document (bestand) op.

Anonymous FTP server

Gopher server

HTTP server

**Notities**

Een WWW server heeft als belangrijkste taak het op verzoek opsturen van documenten. Op het moment dat het World Wide Web werd ontwikkeld werd er in Internet al volop gebruik gemaakt van Gopher en FTP, beiden applicaties voor het ophalen van documenten van remote servers. Het leek de bedenkers van het web dan ook een goed idee om deze server applicaties in het web te integreren. Een web browser kan dus documenten van zowel Gopher servers als anonymous FTP servers ophalen. Naast deze twee bestaande server applicaties is er ook een WWW specifiek protocol bedacht: HTTP. HTTP server applicaties zijn bij uitstek geschikt voor het aanleveren van documenten aan WWW browsers. Bepaalde WWW applicaties (zoals invoerformulieren) zijn alleen mogelijk met het HTTP protocol.

## 18.11 URL

# URL

*Uniform Resource Locator*
ìAdresî van een document in het WWW
Worden geinterpreteerd door de browser
*protocol://hostname*[*:port*]*/URI*

| | |
|---|---|
| *Protocol*: | ftp, http, https, gopher, mailto |
| *Hostname*: | Internet hostname (FQDN) |
| *Port*: | TCP poortnummer |
| *URI*: | Identificatie document binnen de server |

**Notities**

In een wereldwijd systeem als het World Wide Web moet er natuurlijk een wereldwijd unieke adressering voor documenten zijn. Via deze adressering moet ieder document op de aardbol uniek te identificeren zijn. Binnen het WWW wordt gebruik gemaakt van URL's: *Uniform Resource Locator*. Een URL is een unieke aanduiding van een document en beschrijft:

- Het protocol waarmee het document moet worden opgehaald.
- De server waarvandaan het document moet worden opgehaald.

Optioneel De TCP poort waarachter die applicatie huist.
Indien deze niet wordt meegegeven neemt de browser aan dat de server applicatie achter de voor die applicatie gebruikelijke (well known) poort hangt.

Optioneel De naam van het document binnen de server (URI) Indien deze niet wordt meegegeven neemt de server een default document (meestal 'index.html').

Daar hostnamen in het Internet uniek zijn, en URI's binnen een server uniek moeten zijn is een URL een unieke documentadressering.

## 18.12 URI

# URI

*Uniform Resource Identifier*

Formaat:

| | |
|---|---|
| *Padnaam* | (bestandsverwijzing) |
| *Padnaam#sectie* | (intra document verwijzing) |
| *Padnaam?data* | (variabelen) |

**Notities**

De benaming van een document binnen een server noemen we een *Uniform Resouce Identifier*, URI. De browser gebruikt het protocol, host en poortnummer gedeelte van een URL om te bepalen met welke server hij contact op moet nemen en hoe hij met deze server moet praten. Is de verbinding eenmaal tot stand gekomen dan gebruikt de browser de protocolspecifieke instructies om het document aangeduid door de URI op te halen. Een WWW server weet dus niet met welke URL hij is aangeduid, en waar die URL vandaan kwam.

URI's kunnen worden gezien als padnamen van bestanden op de WWW server. Het is echter niet gezegd dat de bestanden van de disk af komen (meestal wel). Het is ook goed mogelijk dat een WWW server wordt aangedreven door een database van documenten. In dat geval is de URI een andersoortige identificatie van het op te halen document. De browser kent meestal geen betekenis toe aan de URI, het is hoofdzakelijk een identificatie voor de WWW server.

Via een speciale syntax kan met een URI meteen worden doorverwezen naar een sectie in een document (URI#sectie). Binnen het document moet dan opp één of andere manier (via HTML instructies) zijn aangegeven waar de sectie begint. In dit geval wordt de URI wel (gedeeltelijk) ook door de browser geïnterpreteerd. Ook kan een URI worden gebruikt om variabelen mee te geven aan de WWW server. Via de syntax URI?var1=a&var2=b kan een lijst van variabelen worden doorgegeven. Deze mogelijkheid wordt alleen toegepast in het HTTP protocol voor het doorgeven van de ingevulde velden op een invoerformulier.

## 18.13 URL voorbeelden

URL voorbeelden

http://www.fbi.gov
gopher://gopher.osp.nl
ftp://ftp.uu.net/pub/rfc/rfc822.txt.gz
http://localhost:8080/aap
http://www.osp.nl/triviant
https://www.shop.com/order#howto
http://www.osp.nl/cgi-bin/viewrfc?n=822

**Notities**

Op de slide staan een aantal voorbeelden van bekende en minder bekende URL's.

# Hoofdstuk 19

# HTTP en HTML

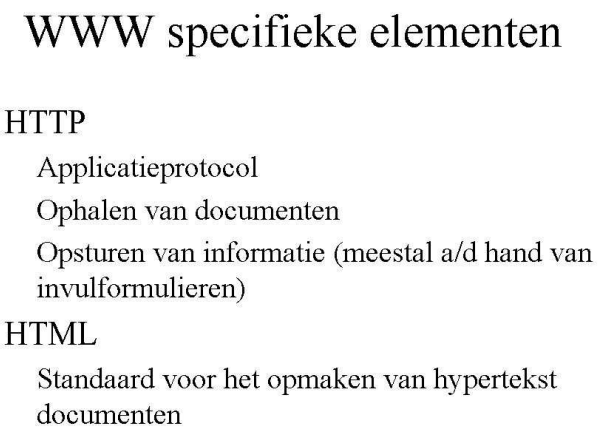
## 19.1 HTTP en HTML

HTTP en HTML¸◊◊ƒXú∞X¸

**Notities**

Welkom bij de module over HTTP en HTML. In deze module gaan we dieper in op twee bekende fenomenen in het World Wide Web: het HTTP protocol en de HTML opmaaktaal.

## 19.2 WWW specifieke elementen




**Notities**

Naast ondersteuning voor bekende applicaties en documenttypen kent het World Wide Web twee “eigen” elementen: HTTP en HTML. HTTP is een lichtgewicht toestandsloos protocol voor het ophalen van documenten en het opsturen van informatie. HTML is *de* standaard voor het opmaken van hypermedia documenten in het World Wide Web. Alle World Wide Web gebruikers en webmasters maken gebruik van deze twee elementen.

## 19.3 HTML

HTML ◊9◊

*HyperText Markup Language*

Taal voor het logisch beschrijven van een document

De HTML interpreter (browser) bepaalt hoe het document er fysiek uit komt te zien

HTML documenten zijn tekstbestanden

HTML *tags* bevatten de metainformatie

**Notities**

Één van de grondslagen van het World Wide Web is het toegankelijk maken van informatie op basis van hypermedia documenten. Deze documenten kunnen naast tekst ook andere elementen bevatten zoals plaatjes en verbindingen met andere documenten. Dit soort elementen moeten dus op één of andere manier in de documenten kunnen worden "gehangen". De "webfathers" hebben er voor gekozen om de WWW hypermedia documenten op te stellen in een ASCII gebaseerde opmaaktaal: HTML, de HyperText Markup Language. Met behulp van deze opmaaktaal wordt de inhoud van een document logisch gespecificeerd. De HTML-programmeur geeft met HTML-opdrachten (zogenaamde *tags*) in de tekst weer hoe de structuur van het document is. Niet-tekstuele elementen en metainformatie (zoals verwijzingen naar andere documenten) worden met HTML-tags gespecificeerd. HTML lijkt wat dat betreft veel op andere opmaaktalen zoals troff (UNIX), DCF (VM en MVS) en TEX.

De browser die een HTML-document moet weergeven interpreteert de tags en doet zijn best om het document af te beelden. Hierbij is het aan de browser om te bepalen hoe bepaalde elementen worden weergegeven. Verschillende browsers geven bijvoorbeeld hoofdstuktitels en verwijzingen naar andere documenten verschillend weer. Hierdoor kan hetzelfde document er in twee browsers tamelijk verschillend uitzien. Zo mag een browser ook zelf bepalen wat hij doet met elementen die niet kunnen worden afgebeeld (door hardware/software beperkingen).

## 19.4 HTML voorbeeld

HTML voorbeeld

```
<html>
<head>
<title>ABN AMRO Internet Info Shop 4</title>
</head>
<body>
<h1>Voorbeeld document</h1>
Dit is een voorbeeld document. De items tussen <> zijn de
HTML <i>tags</i>. Druk <a href=ìnextdoc.htmlî>hier</a>
voor het volgende document.
</body>
</html>
```

**Notities**

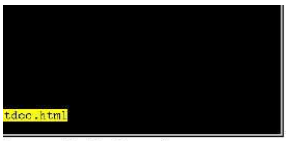
Op de slide ziet u een voorbeeld van een HTML-document. De elementen tussen de ’kleiner-dan’ en ’groter-dan’ tekens zijn de HTML-tags die vertellen hoe het document moet worden opgemaakt. Voor een volledige beschrijving van de HTML taal verwijs ik volgaarne naar één van de vele goede boeken op dit gebied.

## 19.5 HTML voorbeeld in IE


**Notities**

Op deze slide ziet u hoe het voorbeeld van de vorige slide eruit ziet indien het wordt getoond door de Microsoft Internet Explorer. Zoals u ziet interpreteert IE de HTML en geeft het document opgemaakt weer. Met een optie van de browser kunnen we meestal de originele HTML source weer boven water toveren.

## 19.6 HTML voorbeeld in lynx




beeld in lynx

**Notities**

Ziehier hetzelfde document, maar nu afgebeeld door de lynx browser (universiteit van Kansas). Lynx is een browser voor ASCII beeldschermen. Hij kan dan ook een aantal tags niet of nauwelijks weergeven. Toch doet ook lynx zijn best om het document correct af te beelden. Documenten die alleen maar grafische elementen bevatten kunnen met lynx dus niet worden bekeken!

## 19.7 HTML faciliteiten

HTML faciliteiten

Algemene opmaak

Hoofdstuk, paragraaf, lijsten, vet, *schuin*

Invoegen van plaatjes (<img> tag)

Invulvelden (tekst, knopje, *listbox*)

Gesplitste documenten (*frames*)

**Notities**

In HTML zitten allerlei faciliteiten voor het opmaken van documenten. Hieronder vallen tags voor logische opmaak (hoofdstuktitels, lijsten), fysieke opmaak (vet, schuin, gecentreerd) en invoerformulieren (knopjes, invulvelden). De meest recente uitbreidingen aan HTML maken het mogelijk om gesplitste documenten met subdocumenten, ook wel *frames* genoemd, te maken.

## 19.8 HTML aandachtspunten

HTML aandachtspunten

Daadwerkelijke presentatie afhankelijk van de browser

Verschillende HTML versies:

- HTML 1.0
- HTML +
- HTML 2.0
- HTML 3.0

Browser specifieke uitbreidingen

**Notities**

Wat erg belangrijk is bij het programmeren van HTML is dat browsers een grote vrijheid hebben bij het bepalen van hoe het document er uiteindelijk uit komt te zien. Het is dan ook aan te bevelen om een pagina met verschillende browsers zelf te bekijken alvorens hem in produktie te brengen. Verder zijn er verschillende versies van HTML geweest, en nog steeds in omloop. Dit betekent dat de allernieuwste HTML features niet door alle momenteel gebruikte browsers worden begrepen. Niet alle gebruikers upgraden continu naar de nieuwste versies van de browsers (bijvoorbeeld omdat ze nog vast zitten aan Windows 3.1).

Tot overmaat van ramp hebben implementeren sommige browsers hun eigen uitbreidingen aan HTML. Dit betekent dat die browsers bepaalde niet-standaard HTML tags begrijpen. Een document dat van die tags gebruik maakt kan niet door andere browsers worden geïnterpreteerd.

## 19.9 HTML, pro en contra

HTML, pro en contra

Pro:

Eenvoudig

Contra:

Veel verschillende versies in omloop

Zwakke *markup language*

Mengsel van logische en fysieke opmaak (b.v. de <h1> versus de <i> en <b> tags)

Geen zoek of index tags

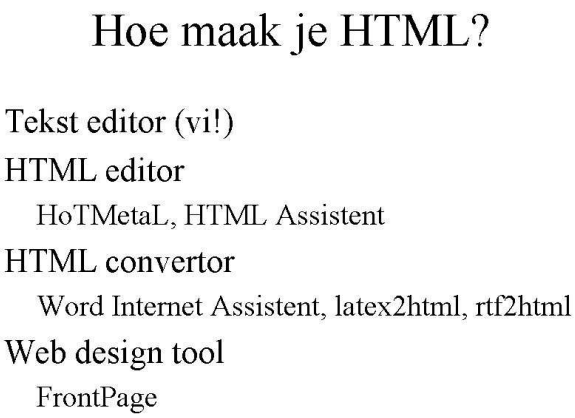
**Notities**

Het grote argument voor HTML is de ongekende simpelheid. Populair gesproken kan iedere dwaas met enige automatiseringskennis een HTML pagina in elkaar draaien. De eenvoud van HTML heeft er ook zorg voor gedragen dat er al snel na een groot aantal browsers was die HTML documenten goed konden weergeven.

Er zitten echter ook een groot aantal nadelen aan HTML: er zijn veel verschillende versies in omloop, het is een zwakke opmaaktaal met weinig mogelijkheden en er ontbreken een aantal features die het zoeken en indexeren van HTML documenten zouden vergemakkelijken. Minder belangrijk (maar daarom niet minder waar) is dat HTML geen “schone” opmaaktaal is; er zitten zowel tags in met een logische betekenis (bijvoorbeeld de hoofdstuktitel tag) als tags met instructies voor de fysieke opmaak (de *bold* tag).

Ondanks alle problemen is HTML zeer wijd verspreid. Het is een van de belangrijkste bouwstenen van het hedendaagse WWW en kan mede om die reden niet snel worden vervangen. Betere opmaaktalen zijn vaak dusdanig gecompliceerd dat ze niet snel kunnen worden getoond, of staan niet het niveau van controle toe dat webmasters over de opmaak van de pagina’s willen hebben.

## 19.10 Hoe maak je HTML?




**Notities**

Op zijn eenvoudigste manier kunnen HTML documenten worden aangemaakt met een willekeurige tekst editor. HTML documenten zijn immers ASCII tekstbestanden met daarin diverse ASCII opmaakcommando's. Een probleem met deze aanpak is dat de HTML-programmeurs de syntax van de verschillende HTML tags uit zijn/haar hoofd moet kennen. Verder bevatten de hedendaagse HTML documenten allerlei niet-tekstuele componenten (meestal plaatjes) die in geen geval met een tekst editor kunnen worden aangemaakt. Om die reden zijn er diverse hulpmiddelen ontwikkeld waarmee op een gebruikersvriendelijkere manier HTML pagina's in elkaar kunnen worden gesleuteld.

Een voorbeeld van dergelijke hulpmiddelen zijn de HTML editors. Dit zijn interactieve applicaties waarmee onder besturing van een syntax editor HTML documenten kunnen worden gemaakt. De HTML editor kent de HTML syntax en voorkomt syntaxfouten in het document. Meestal werken die HTML editors volgens de WYSIWYHYG ("what you see is what you hope you get") methode. Met enkele oogopslagen is duidelijk hoe het document ongeveer door een gemiddelde browser zal worden weergegeven.

Ook zijn er een aantal zogenaamde HTML convertors ontwikkeld. Dit zijn hulpmiddelen die de bestanden van bestaande tekstverwerkings- en andere applicaties kunnen converteren naar HTML. Hierdoor kunnen reeds gemaakte documenten naar HTML worden geconverteerd ter voorbereiding van de publicatie van die informatie op het World Wide Web. De bekendste convertors zijn de Internet Assistents van Microsoft waarmee verschillende Office bestanden in HTML kunnen worden weggeschreven.

Het meest complete hulpmiddel wordt gevormd door de "web design tools". Het betreft hier applicaties waarmee volledige netwerken van hypermedia documenten kun-

nen worden ontwikkeld. Een bekend voorbeeld hiervan is FrontPage van Microsoft (ingekocht van Vermeer Technologies). Met FrontPage kan bijvoorbeeld een heel web van documenten worden ontwikkeld. FrontPage bewaakt onder andere de consistentie in opmaak van, en verwijzingen tussen, de documenten.

## 19.11 HTTP

HTTP

*HyperText Transport Protocol*
Functies:
Ophalen documenten
Opsturen ingevulde formulieren
Text based command/response protocol
Commandoís: GET, HEAD, POST
Well known port: 80
Versies: HTTP/0.9 en HTTP/1.0
*Stateless*

**Notities**

De meeste transacties tussen WWW clients en servers worden uitgevoerd volgens het HTTP protocol. HTTP is een lichtgewicht protocol voor het ophalen van documenten en opsturen van informatie. Het is specifiek ontwikkeld voor toepassing in het World Wide Web. De client zijde van het HTTP protocol wordt ingevuld door de HTTP browsers. Op de HTTP server draait een HTTP server applicatie welke de verbindingen opneemt en afhandelt.

Het HTTP protocol is volgens de beste Internet tradities een tekst geörienteerd command/response protocol (net als SMTP en POP). Dit heeft tot gevolg dat HTTP sessies met telnet kunnen worden nagespeeld om de werking van een server te testen. De commando's van het HTTP protocol zijn GET (om een document op te halen), HEAD (om alleen de headers van document te krijgen) en POST (om informatie naar de server op te sturen). Het HTTP protocol maakt gebruik van de well known port 80. De eerste HTTP implementaties hadden nog slechts een beperkte functionaliteit. Deze implementaties staan heden ten dage bekend als HTTP 0.9. Alle hedendaagse HTTP clients en servers ondersteunen de volledige variant HTTP 1.0. Aan de standaardisering van HTTP 1.1 wordt momenteel gewerkt.

Kenmerkend voor HTTP is dat het een toestandsloos protocol is. Dit houdt in dat de HTTP server geen informatie onthoudt van verbinding tot verbinding. Iedere request wordt op zichzelf beoordeeld, zonder referentie aan eerdere requests. HTTP kent dan ook het begrip "sessie" absoluut niet. Iedere keer dat een document wordt opgehaald of informatie wordt opgestuurd bepaalt de HTTP server of die transactie is toegestaan.

HTTP vereist verder ook geen authenticatie. De browsers hoeven normaal gesproken niet aan te loggen om een document op te kunnen halen. In het HTTP protocol zijn

slechts marginale functies aanwezig voor het bepalen van de identiteit van de gebruiker.

## 19.12 HTTP server software

HTTP server software

NCSA httpd
CERN httpd
Microsoft Internet Information Server
WebSite
Netscape Server
Apache
Stronghold
OSP httpd

**Notities**

Op de slide staan voorbeelden van diverse HTTP server implementaties. Er bestaan een groot aantal public domain implementaties van HTTP servers. Hiervan zijn de NCSA, CERN en Apache servers het bekendst. Daarnaast hebben een aantal bedrijven ook commerciële HTTP servers ontwikkeld.

## 19.13 HTTP voorbeeld

HTTP voorbeeld

GET / HTTP/1.0
From: josv@osp.nl
*... overige headers*
<== Lege regel = scheider

HTTP/1.0 200 OK
Content-Type: text/html
Content-Length: 391
*... overige headers*

<html><head><h1>ABN AMRO</h1></head>
<body><h1>Welkom bij de server van de ABN AMRO Bank</h1>
*... meer html*
*</html>*

**Notities**

Zoals gezegd is HTTP een “traditioneel” text based command/response protocol. Op de slide staat een met telnet uitgevoerd voorbeeld van HTTP transactie (telnet naar poort 80 van een WWW server). De client opent de verbinding en geeft een GET commando. In dit GET commando wordt (onder andere) ook aangegeven wat de URI is van het document dat moet worden teruggegeven. Naast het GET commando stuurt de client ook nul of meer request headers mee. De server weet dat de request compleet is ontvangen zodra de eerste lege regel binnenkomt.

De server spoort vervolgens het gevraagde document op en stuurt het terug. De server response begint met een “HTTP/1.0 200 OK” regel om aan te geven dat het gevraagde document beschikbaar is. Vervolgens stuurt de server nul of meer response headers. Hierin wordt onder andere aangegeven wat het documenttype van het gevraagde document is. Na de response headers volgt een lege regel en de inhoud van het gevraagde document.

Indien een document ingesloten objecten bevat (zoals plaatjes) moet de browser meerdere HTTP GET transacties uitvoeren alvorens het gehele hypermedia document “binnen” is.

## 19.14 Waar komen documenten vandaan?

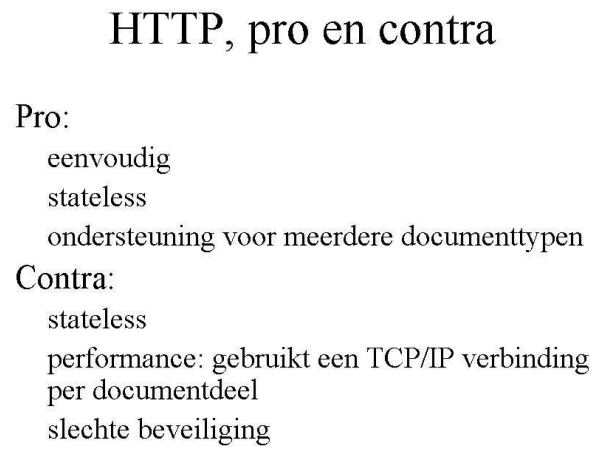
Waar komen documenten vandaan?

- Disk van de HTTP server
- Gegenereerd door HTTP server
  - Bijvoorbeeld directory index
- Gegenereerd door *server side* applicaties

**Notities**

Bij alle in gebruik zijnde HTTP servers wordt de “document store” gevormd door een gedeelte van de hard disk van de server. URI’s corresponderen in dat geval met de relatieve padnaam van een bestand in de document directory van de HTTP server. Het HTTP protocol schrijft dit echter niet voor. Er is tenminste één server waarbij de documentenverzameling wordt gevormd door een database (de HyperWave server van de universiteit van Graz). HTTP servers ondersteunen meestal ook dat URI’s verwijzen naar uitvoerbare programma’s. In het geval een browser naar een dergelijke uitvoerbare URI verwijst voert de HTTP server de URI uit en stuurt de output van dit programma door naar de browser.

## 19.15 HTTP, pro en contra




**Notities**

Er is al veel gediscussieerd over het HTTP protocol. Feit is dat het protocol eigenlijk totaal niet bedoeld was voor de manier waarop het vandaag aan de dag wordt gebruikt. Veel organisaties maken gebruik vab het feit dat er veel geode browsers in omloop zijn om HTTP en HTML in te zetten voor het bouwen van complexe client/server applicaties. In principe biedt HTTP hier geen features voor. Met extra software en een aantal trucjes kan het protocol wel worden opgelapt om wat meer functionaliteit te bieden, maar het blijft behelpen.

Één van de belangrijkste eigenschappen van HTTP is dat het een toestandsloos protocol is. De HTTP server beziet iedere request als een op zichzelf staand gebeuren en doet geen pogingen om informatie te onthouden van verbinding tot verbinding. Voor het implementeren van “eenvoudige” algemeen toegankelijke electronische informatiediensten is dit een voordeel. Voor meer complexe client/server transacties is de toestandsloosheid echter een groot nadeel. Met trucjes als *cookies* kunnen we deze toestandsloosheid opheffen. De verantwoordelijkheid voor het onthouden van die toestand ligt dan echter bij de applicatie. Moderne HTTP servers hebben ingebouwde mechanismen voor het onthouden van informatie van verbinding tot verbinding (zoals de OSP httpd).

HTTP kent ook weinig ingebouwde beveiligingsmethoden. Er is een vorm van authentication, de zogenaamde *Basic Authentication* waarbij gebruikers zich moeten melden met userid en password. Dit betekent echter dat vanaf dat moment bij iedere request het userid password onversleuteld met de request worden meegestuurd!.

Een ander nadeel van HTTP is de slechte performance voor complexe hypermedia documenten. Indien een complex document meerdere elementen (zoals plaatjes, video,

geluid) bevat wordt ieder element door middel van een aparte HTTP request opgehaald. Dit heeft als gevolg dat er per HTML element een aparte TCP verbinding moet worden opgezet.

Een gedeelte van de hier genoemde nadelen zal naar verwachting worden opgeheven in de volgende release van HTTP: HTTP/1.1.

# Hoofdstuk 20

# Web based applicaties

## 20.1 Web based applicaties

Web based applicaties

**Notities**

In de inleiding van de info shop is al gesteld dat het WWW kan worden gebruikt voor eenvoudige (en meer complexe) interactieve client/server applicaties. In deze module gaan we dieper in op de manier waarop *server side* web applicaties kunnen worden gebouwd. Hierbij is de applicatie volledig geïmplementeerd op de WWW server. Met nieuwere WWW technologie kunnen we ook een gedeelte van de applicatie op de WWW client laten draaien. Deze mogelijkheden worden in de volgende module “Active Content” uitgediept.

## 20.2 Web based applicaties

Web based applicaties

Gebruik van WWW voor transactie georienteerde applicaties.

Dynamisch genereren van documenten (HTML) door programmatuur op de HTTP server.

**Notities**

Veel organisaties hebben het web ontdekt als universeel interactief transactiemedium voor het realiseren van client/server applicaties. Hierbij speelt een grote rol dat de client en server applicaties van het web wijd verbreid zijn: gebruikers hoeven niet eerst een proprietary client applicatie te installeren, ze kunnen meestal gelijk aan de gang met hun huidige web browser.

Toepassingen als electronische winkels hebben naast een puur informatief gedeelte vaak ook een interactieve sectie waar goederen en dergelijke kunnen worden besteld. Dit vereist dat er interactie is tussen de gebruiker en de WWW server van de winkel. Ook remote database applicaties vereisen dit soort interactie. Steeds meer traditionele applicaties komen ook met een WWW interface zodat een gebruiker met een web browser de applicatie over het intranet of Internet kan gebruiken.

Web based applicaties zijn gebaseerd op twee fundamenten:

1. HTML invoerformulieren die door gebruikers kunnen worden ingevuld.
2. Programmatuur op de WWW server die de ingevulde formulieren verwerkt en op basis daarvan *dynamisch* weer nieuwe HTML pagina's (of formulieren) genereert.

## 20.3 Voorbeeld: OSP Triviant 1

**Notities**

Op de slide ziet u een voorbeeld van een web based applicatie. Het betreft hier een tamelijk eenvoudige applicatie voor het spelen van het spel triviant. De gebruiker moet allereerst aanloggen aan de triviant applicatie met behulp van een userid en password. De applicatie biedt vervolgens een aantal mogelijkheden: het beantwoorden van de vraag van de dag, het bekijken van de scorelijsten en het invoeren van een nieuwe vraag.

## 20.4 Voorbeeld: OSP Triviant 2


**Notities**

Het op de slide getoonde HTML document is dynamisch gegenereerd op basis van de aanloginformatie van de gebruiker.

## 20.5 Implementatie

Implementatie

CGI Programmatuur
NSAPI/ISAPI applicaties
Tools:
- WebObjects
- Internet Database Connector

Nieuw: Java *servlets*

**Notities**

Om web based applicatie te kunnen ontwikkelen dient er programmatuur aan de server zijde te worden geïmplementeerd. Deze programmatuur is verantwoordelijk voor het afhandelen van een HTTP request en het genereren van HTML pagina's op basis van deze request.

Op de slide staan een aantal mogelijkheden voor het implementeren van web based applicaties. De simpelste (en gestandaardiseerde) methode hiervoor is CGI-programmatuur. Deze mogelijkheid wordt door alle HTTP server applicaties ondersteund. Daarnaast kennen verschillende HTTP servers hun eigen mogelijkheden voor het implementeren van web applicaties.

## 20.6 CGI

### CGI

*Common Gateway Interface*

URL refereert een programma i.p.v. een document

HTTP server software start dit programma

Input : Allerhande informatie (variabelen)

Output : Naar de browser

CGI programmaís meestal in C, C++ of scripttalen (Perl, shell script)

**Notities**

De oudste manier om web applicaties te bouwen is via het *Common Gateway Interface*. Dit is een standaard volgens welke de HTTP server programma's opstart en de gegevens uit het originele request meegeeft. Het is de taak van het CGI programma om de request te verwerken en op basis van de gegevens uit de request (bijvoorbeeld de meegegeven waarden van de invulvelden) een nieuw document te genereren. De uitvoer van het CGI programma wordt als antwoord naar de client gestuurd. Het triviant voorbeeld van de voorvorige slide is met behulp van CGI programma's geïmplementeerd.

De werking van CGI programma's is tamelijk transparant voor de WWW client. Een CGI programma wordt gestart door middel van een URI die door de HTTP server wordt herkend als een uitvoer programma. Hoe de HTTP server dit beslist is niet universeel vastgelegd. Iedere HTTP server leverancier mag daar zijn eigen standaard voor kiezen. Meestal is gevoelsmatig aan de URL wel te zeggen of het een CGI programma betreft of een gewoon document. Veel sites maken er een gewoonte van om CGI programma's in een aparte /cgi-bin directory op de HTTP server te plaatsen.

CGI programma's kunnen in iedere gewenste programmeertaal worden ontwikkeld. Een erg populaire taal is de UNIX script taal 'perl', maar ook C, C++ en python zijn populaire talen voor het bouwen van CGI programma's. Een belangrijk aandachtspunt bij CGI programma's is beveiliging. Door middel van het CGI mechanisme zijn browsers immers in staat om programma's (eventueel met parameters) op te starten op de HTTP server. Deze programma's draaien daar met de rechten van de HTTP server applicatie. Indien er in de CGI programmatuur bugs zitten kunnen die wellicht worden uitgebeuit door hackers. Het niet meegeven van verwachte argumenten, of het meegeven van onverwachte waarden in de argumenten is een bekende aanvalsmethode.

Een moeilijk probleem bij CGI programma's is dat de HTTP server geen toestand onthoudt van verbinding tot verbinding. Dit betekent dat een applicatie die bestaat uit meerdere CGI programma's zelf een methode moet implementeren om informatie te onthouden. Ook is niet bekend of een gebruiker zijn browser heeft gestopt, dus de onthouden informatie moet ook regelmatig worden opgeschoond.

## 20.7 NSAPI / ISAPI

# NSAPI / ISAPI

*Netscape Server API*
*Internet Server API* (Microsoft)
Serverzijde applicaties in *shared libraries* of DLLís die door de HTTP server worden geladen
Onderscheppen HTTP request
Genereren HTTP respons (HTML)
Geschreven in C / C++

**Notities**

Een nadeel van CGI programmatuur is dat het losstaande programma's betreft die iedere keer door de HTTP server worden gestart zodra er een request binnenkomt. Indien deze programma's informatie moeten bijhouden doen ze dit meestal door middel van bestanden op de disk. Het is niet mogelijk dit in het geheugen te doen omdat het CGI programma iedere keer stopt zodra er een transactie is afgerond. Het veelvuldig starten en stoppen van CGI programma's heeft ook geen erg gunstig effect op de performance van het systeem.

Om deze reden hebben Netscape en Microsoft een manier bedacht om web applicaties als componenten te laten draaien in de HTTP server applicatie. De Netscape Server ondersteunt hiervoor de NSAPI; Microsoft's Internet Information Server kent de ISAPI. Beiden zijn programmeursinterfaces waarmee applicaties in C/C++ kunnen worden geschreven en *in* de server applicatie kunnen worden geladen (via zogenaamde *shared libraries* of *shared objects*. Bij het starten van de HTTP server laadt deze automatisch alle geconfigureerde applicatiemodules in het geheugen. Op het moment dat er een HTTP request binnenkomt krijgen de geladen applicaties de mogelijkheid om deze request af te handelen. Als zij dit doen zijn ze ervoor verantwoordelijk om het antwoord naar de client te genereren (HTML).

In tegenstelling tot CGI programma's zijn NS/IS API applicaties modules die in het interne geheugen van de server informatie kunnen onthouden met betrekking tot de transacties. Verder hoeven de applicaties niet iedere keer te worden geladen. Hierdoor is de performance van NS/IS API applicaties beter dan die van CGI programma's. Een nadeel van dit soort applicaties is dat ze gebonden zijn aan één HTTP server applicatie. Een ISAPI applicatie kan alleen maar draaien in de HTTP server van Microsoft. CGI is daarentegen een standaard die door alle HTTP servers wordt ondersteund.

## 20.8 Hulpmiddelen

Hulpmiddelen

NeXt WebObjects
- Ondersteuning voor serverzijde applicaties
- Toegang tot databases
- Bijhouden van *state*
- Faciliteiten voor genereren van HTML

Microsoft Internet Database Connector
- ISAPI DLL
- Genereren van HTML uit ODBC databases
- Invoer in database uit HTML formulieren

**Notities**

Ter ondersteuning van het ontwikkelen van web applicaties zijn er door diverse leveranciers hulpmiddelen op de markt gebracht ter ondersteuning van het implementeren van dergelijke applicaties. Een aantal voorbeelden hiervan zijn WebObjects van NeXt en de Merchant Server van Microsoft. Deze hulpmiddelen vergemakkelijken meestal het genereren van HTML pagina's, het bijhouden van toestandsinformatie en de toegang tot diverse databasesystemen.

Daarnaast krijgen steeds meer server applicaties componenten voor het functioneren in een WWW omgeving. Zo kunnen moderne databasesystemen bijvoorbeeld automatisch HTML genereren zodra er iets in de database wijzigt (bijvoorbeeld produktinformatie). De bekende workgroup applicatie Lotus Notes heeft zelfs een hele HTTP server component gekregen voor het toegankelijk maken van de gehele document database via het World Wide Web.

## 20.9 Java servlets

Java servlets

Mogelijkheid om Java programmatuur op de server te draaien.

HTTP server bevat Java *Virtual Machine*

Conceptueel gelijk aan NSAPI / ISAPI

**Notities**

Een nieuwe ontwikkeling op het gebied van web applicaties is de mogelijkheid om web applicatie te implementeren met Java programmatuur op de server zijde. Technisch gesproken is dit vrijwel hetzelfde als het implementeren van een WWW applicatie met een in C/C++ geschreven NS/IS API applicatie. In het geval van Java *servlets* is de applicatie geschreven in Java. De HTTP server bevat een Java *virtual machine* voor het uitvoeren van deze Java programmatuur. De in Java geschreven applicatie krijgt van de HTTP server de kans om HTTP requests te onderscheppen en te interpreteren.

## 20.10 Web applicaties, pro en contra

### Pro en contra

Pro
- Eenvoudig voor simpele toepassingen
- Weinig/geen eisen aan de client (browser)
- *Cool!*

Contra
- Complex voor moeilijkere toepassingen
- In principe stateless
- Beveiliging
- Internet is geen gegarandeerd communicatiemedium

**Notities**

In principe is het World Wide Web bedacht voor het electronisch ter beschikking stellen van informatie en documentatie. Ter ondersteuning hiervan zijn simpele transactiemogelijkheden geïmplementeerd. Veel organisaties “misbruiken” het web echter als vehikel voor complexere client/server applicaties. De in het WWW toegepaste protocollen en technieken bieden hier eigenlijk geen ondersteuning voor. Dit betekent dat de programmeur van een complexe web applicatie een behoorlijk grote toverdoos met trucjes moet hebben om de ontbrekende (maar benodigde) features zelf te programmeren. Inmiddels zijn diverse toverdozen te koop voor het implementeren van web applicaties.

Een groot voordeel van het ter beschikking stellen van een applicatie via het web (Internet of intranet) is dat de benodigde client software in ruime mate voorhanden is op alle denkbare (en niet denkbare) platforms. Hierdoor is de applicatie zonder al te veel problemen breed toegankelijk. Ook in grote organisatie met een heterogeen intern netwerk kan het veel voordelen opleveren om een applicatie via WWW technologie ter beschikking te stellen. Alhoewel nog niet te zeggen valt hoe de toekomstige ontwikkelingen op het gebied van Web-TV en Network Computers zullen verlopen is het investeren in een Web applicatie (gezien de reeds aanwezige kritieke massa) een betrekkelijk veilige aangelegenheid.

# Hoofdstuk 21

# Active Content

## 21.1 Active Content

Active Content

**Notities**

In deze module gaan we in op de mogelijkheden van het World Wide Web om de browsers een wat meer actieve rol te laten spelen bij het tonen van documenten.

## 21.2 "Plain Vanilla" WWW

*Plain Vanilla* WWWXLLM

Domme client
Vindt geen applicatieverwerking plaats
Slimme server
*In flight* genereren van documenten
HTML
GIF

**Notities**

In de originele WWW gedachte bestond het web uit relatief slimme servers die documenten konden ophoesten (of genereren) en domme clients die slechts de taak hadden de opgeleverde documenten te tonen. In die basisgedachte was het niet mogelijk voor een WWW server om een stukje programmatuur op de client uit te (laten) voeren. De oudere web browsers, zoals lynx en Mosaic, vormen een perfecte implementatie van die gedachte. Het grote voordeel hiervan is dat er aan de kant van de WWW client niet veel rekenkracht of opslagcapaciteit aanwezig hoeft te zijn.

## 21.3 Nadelen van “plain vanilla” WWW

Nadelen van *plain vanilla* WWWLL!

Geen verwerking aan de client zijde, dus:
- Controle op fouten op de server (loze transactie)
- Beperkte animatie in de documenten
- Mogelijkheden beperkt door de browser
  - Documenttypen
  - Acties (zoals chipcard betalingen)

**Notities**

Webmasters zochten echter al snel naar wegen om de statische HTML pagina’s te voorzien van bewegende elementen. Ook het toenemende gebruik van het web voor transactie georienteerde web applicaties riep om meer kracht en flexibiliteit aan de kant van de browser. Sites wilden graag de mogelijkheid om:

- Animaties uit te voeren op de client.
- Controle op invulvelden uit te voeren op de client *voordat* deze naar de server worden gezonden.
- De browser uitbreiden met mogelijkheden voor het tonen van niet standaard ondersteunde documenttypen.
- Acties (zoals betalingen en dergelijke) uit te kunnen laten voeren op de client.

.

Om dit mogelijk te maken dient de browser te worden uitgebreid met extra functionaliteit.

## 21.4 Uitbreiding van de browser

Uitbreiding van de browser

Helper applicaties
Plug-Ins
JavaScript
VB Script
Java
ActiveX

**Notities**

Om te ontkomen aan de beperkingen van de “domme’ WWW client dient deze te kunnen worden uitgebreid met extra functionaliteit. Met die extra functionaliteit kunnen statische pagina’s “tot leven worden gewekt”, en kunnen niet-standaard acties aan de client worden geïnitieerd.

Een nadeel van dergelijke uitbreidingen aan de client zijde is dat daarmee afhankelijkheden worden geïntroduceerd naar de configuratie van de client. Een web site die gebruik maakt van dergelijke uitbreidingen kan er meestal niet meer vanuit gaan dat iedere web browser de mogelijkheden heeft om de web applicatie(s) in kwestie te ‘draaien’. Meestal bieden dit soort web sites dan ook de mogelijkheid om de benodigde extra programmatuur te downloaden.

Op de slide zijn de momenteel in zwang zijnde mogelijkheden voor ‘Active Content’ opgesomd. In de nuvolgende slides worden deze één voor één besproken.

## 21.5 Helper applicaties


**Notities**

De meest eenvoudige methode om de functionaliteit van de browser uit te breiden is de ‘Helper’ applicatie. Het betreft hier een programma wat door de browser wordt gebruikt om documenttypen weer te geven die niet standaard door de browser worden begrepen. Zoals eerder in de shop is verteld kan een browser op een of andere manier bepalen wat het MIME bestandstype is van het opgehaalde document. Dit gebeurt aan de hand van de bestandsnaam van het opgehaalde document of aan de hand van de ‘Content-Type’ header in de HTTP response. Indien de browser het bestandstype ‘kent’ kan hij het zelf op een bepaalde manier tonen.

Voor documenttypen die de browser niet kent kan deze worden geconfigureerd om een helper applicatie op te starten. In de configuratie van de browser wordt dan aangegeven dat voor documenttype ‘abc/xyz’ de helper applicatie ‘zus-en-zo’ moet worden gestart. In het geval de browser een document van het gespecificeerde type ophaalt slaat hij dit document op en start hij de helper applicatie. Het is dan de verantwoordelijkheid van de helper applicatie om het document in kwestie af te beelden.

Aangezien de helper applicatie een losstaand programma is vindt er na het starten van de helper door de browser geen communicatie meer plaats tussen helper en browser. De helper heeft in principe de beschikking over de volledige mogelijkheden van de client, en kan dus netwerkverbindingen openen, met de gebruiker communiceren en bestanden manipuleren.

Helper applicaties hoeven niet specifiek voor het World Wide Web te worden ontwikkeld. Standaard applicaties kunnen ook als helper worden ingeschakeld. Zo kan de Word Viewer applicatie van Microsoft ook worden gebruikt als helper applicatie voor het weergeven van Word documenten die via het World Wide Web worden opgehaald.

Helper applicaties hoeven ook niet speciaal voor een bepaalde browser te worden ontwikkeld. De enige interactie tussen de browser en de helper is dat de browser op de voor het client-OS geldende standaardmanier wordt gestart.

## 21.6 Plug-Ins

Plug-Ins

Applicatie voor het afbeelden van documenttypen die niet door de browser worden begrepen (MIME type)
<EMBED> tag
In de browser geintegreerd (dynamisch laden)
Client OS en browser specifiek
Plug-in:
- krijgt eigen subwindow in de browser
- ontvangt de data van het (sub)document
- kan browser functies aanroepen (API)

**Notities**

Een andere mogelijkheid om de functionaliteit van een browser uit te breiden is door middel van plug-ins. Een plug-in is een soort helper applicatie die *in* de browser wordt opgestart. Als output windows krijgt een plug-in een gedeelte van het window van de browser toegewezen. Dit betekent dat voor de gebruiker de plug-in een integraal onderdeel uitmaakt van het getoonde HTML document. Plug-ins hebben echter een eigen 'leven', ze kunnen onafhankelijk van de browser allerlei aktiviteiten ontplooien. Plug-ins kunnen ook met de browser communiceren via een programmeur interface (API). In tegenstelling tot een helper applicatie is een plug-in geschreven voor een specifieke browser en client-OS.

Plug-ins worden geaktiveerd door de EMBED tag. De browser die de HTML interpreteerd maakt een subwindow voor de plug-in, laadt de plug-in, download de plug-in data en start de plug-in.

De integratie tussen een plug-in en de browser is optimaal. Dientengevolge ziet het getoonde document er zeer gelikt uit. Het niet standaard onderdeel wordt integraal in het document getoond in het door de plug-in aangestuurde subwindow. Een belangrijk nadeel van een plug-in is dat de plug-in apart op de computer van de gebruiker moet zijn geïnstalleerd. Verder zullen documenten waarin gebruik wordt gemaakt van de EMBED tag (om een plug-in te aktiveren) alleen werken gebruikers die de plug-in ook hebben geïnstalleerd. Veelal bevatten sites die gebruik maken van plug-ins ook mogelijkheden om de gebruikte plug-ins te downloaden.

## 21.7 JavaScript

JavaScript

Voorheen *LiveScript*
Ondersteund door Netscape en Microsoft IE
Script programmeertaal
Source gaat mee in het document via
<SCRIPT> tag
Script wordt door de browser uitgevoerd
Beperkte functionaliteit

**Notities**

Een andere moderne mogelijkheid van het uitvoeren van programmatuur aan de client zijde is het meesturen van die programmatuur in het HTML document. De bedenker van dit fenomeen is de Amerikaanse leverancier van WWW technologie Netscape. De uitvinding van Netscape, LiveScript (later hernoemd naar JavaScript), maakt het mogelijk om met het HTML document allerlei programmatuur (in source code) mee te sturen. De ontvangende browser interpreteert naast de HTML ook de JavaScript source en voert die uit.

JavaScript bevat allerlei mogelijkheden voor het afvangen van gebruikersinteractie (muisbeweging, tekstinvoer, drukken op knopjes) en het automatisch uitvoeren van programmatuur zodra een pagina wordt geladen. Hierdoor kan een web applicatie allerlei JavaScript code met een document meesturen om een gedeelte van de applicatieverwerking aan de client zijde uit te laten voeren. Hierdoor kunnen bijvoorbeeld controles op ingevulde velden al door de browser worden uitgevoerd alvorens het hele formulier naar de server wordt gestuurd. Verder kunnen JavaScript procedures allerlei berekeningen uitvoeren en het resultaat daarvan terugplaatsen in het document.

JavaScript biedt geen algemene functionaliteit voor het benaderen van bestanden of het netwerk. Hierdoor brengt het uitvoeren van JavaScript programmatuur door de browser geen onaanvaardbare beveiligingsrisico's met zich mee. Een belangrijk nadeel van JavaScript is natuurlijk dat WWW pagina's die JavaScript code bevatten alleen werken als de browser waarmee die pagina's worden bekeken ook JavaScript compatibel is. Non-JavaScript browsers geven vaak foutmeldingen indien ze een JavaScript script tegenkomen of laten slechts een halve pagina zien.

## 21.8 VBScript

VBScript

*Visual Basic Script*
Ondersteund door Microsoft IE
A la JavaScript

**Notities**

Microsoft kon zich natuurlijk met het aanstormende Java en JavaScript geweld niet onbetuigd laten en het heeft dan ook zijn standaard macrotaal Visual Basic geschikt gemaakt voor het World Wide Web. het resultaat hiervan is VBScript, een geïnterpreteerde programmeertaal die net als JavaScript met de HTML documenten mee kan worden gestuurd (ook via de SCRIPT tag). VBScript is qua toepassing en implementatie vrijwel gelijk aan JavaScript. VBScript wordt echter bij mijn weten zo goed als nergens toegepast.

## 21.9 Java

Java

Hype, Hyper, Java
Object georienteerde programmeertaal
Complete programmeertaal (vgl: C++)
Ontwikkeld door Sun Microsystems
Java compiler vertaalt naar architectuur neutrale bytecode (pseudo machine taal)
Breed omarmd door IT-industrie
WWW Toepassing: *servlets* en *applets*

**Notities**

De grootste Internet hype van de laatste eeuw is zonder meer Java. Deze programmeertaal van leverancier Sun heeft in 'no time' een zeer aanzienlijke schare aanzienlijke fans verkregen. Grote IT leveranciers als IBM, HP, Netscape en Microsoft kondigden vrijwel gelijk met de vrijgave van Java aan de programmeertaal in meer of mindere mate te gaan ondersteunen.

Java is een door Sun ontwikkelde object geörienteerde, volledige, platform onafhankelijke, *multi threading*, robuuste, veilige, architectuur neutrale programmeertaal. De Java vertaler zet Java source code om in een neutrale tussencode (P-code, bytecode) die door een run time interpreter (de zogenaamde Java Virtual Machine) moet worden geïnterpreteerd. Java is een moderne en goede programmeertaal waarin zonder problemen grotere en kleinere applicaties kunnen worden ontwikkeld. Java bevat ingebouwde mogelijkheden voor bestandsmanipulatie, het opbouwen van netwerkverbindingen en het bouwen van grafische gebruikers interfaces.

De grote populariteit van Java in Internet is ontstaan doordat Sun in Java een browser ontwikkelde (HotJava) die het mogelijk maakte om stukken gecompileerde Java programmatuur (Java executables) over het netwerk te laden en in de browser uit te voeren. We noemen deze gedownloade Java programmatuur ook wel 'applets'.

## 21.10 Java applets

Java applets

Soort dynamische plug-in

<APPLET> tag

- Browser download Java ë.classí file (bytecode)
- Applet krijgt eigen subwindow
- Browser voert Java executable uit

Applet kan:

- Tekenen (animatie)
- User interactie
- Netwerkverbindingen opbouwen (remote databases)

**Notities**

Een Java applet is een soort dynamische gedownloade plug-in. De HTML programmeur geeft in zijn document aan dat op een bepaalde plek in het document een Java programma moet draaien. Hij/zij geeft dit aan door het specificeren van een APPLET tag. De browser die dit document interpreteert herkent de APPLET tag, downloadt de Java executable (in de platform onafhankelijk bytecode) met het HTTP protocol en interpreteert deze bytecode. De draaiende Java applet krijgt net als een plug-in een eigen subwindow in de browser. Hierdoor voert de applet *in* de browser uit. Voor de gebruiker is de draaiende applet dus een integraal onderdeel van het document. Conceptueel kan een Java applet dus worden gezien als een kruising tussen een plug-in en een JavaScript script.

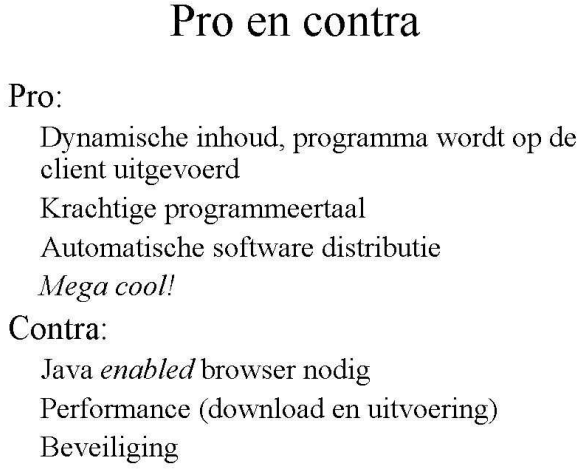
Een applet heeft de kracht van een plug-in (want geschreven in een volledige, krachtige programmeertaal) en de dynamiek van een JavaScript script (want gedownload over het netwerk). De gebruiker hoerft dus niet eerst een apart stuk programmatuur te installeren voordat een Java applet kan gaan draaien! De applet wordt ‘on demand’ over het netwerk opgehaald en uitgevoerd. De mogelijkheden hiervoor zijn natuurlijk eindeloos! Browsers kunnen dynamisch worden uitgebreid met de mogelijkheden om bepaalde typen documenten te tonen of om bepaalde acties te ondernemen (zoals het uitvoeren van een betaling).

Het over het netwerk ophalen en zonder meer uitvoeren van een programma is natuurlijk beveiligingstechnisch een hachelijke zaak. Om deze reden hebben de bedenkers van Java bewust allerlei faciliteiten niet in de taal geïmplementeerd (zoals pointers en het rechtstreeks kunnen benaderen van geheugen). De Java bytecode interpreters in de browsers laten vervolgens een groot aantal zaken niet toe. Zo kunnen Java applets

geen bestanden benaderen op de lokale disks van de gebruiker en kunnen ze alleen maar netwerkverbindingen opbouwen met de site waarvandaan ze gedownload zijn. De Java applet draait als het ware in een software gevangenis waaruit ze niet kunnen ontsnappen.

Alhoewel er intussen meerdere Java gerelateerde beveiligingsbugs boven water zijn getoverd zijn deze allemaal terug te voeren op fouten in de implementatie. In het Java beveiligingsmodel zijn nog geen structurele fouten aangetoond!

## 21.11 Java, pro en contra




**Notities**

Java wordt door velen gezien als de toekomst van Internet en intranet. De mogelijkheid om applicaties in een architectuur neutrale tussencode op te sturen naar een client om daar te worden uitgevoerd is natuurlijk ook ongekend krachtig. Het maakt niet uit wat voor soort computer de gebruiker heeft, als zijn/haar browser Java ondersteunt kan de Java applet ter plekke worden uitgevoerd. Java is intussen zelfs doorgedrongen op de server waar web applicaties kunnen worden ontwikkeld in Java. Deze Java servlets draaien natuurlijk niet in een software gevangenis en kunnen alle mogelijkheden van het systeem benaderen. Er wordt door diverse organisaties grootscheeps geïnvesteerd in Java en het is ontegenzeggelijk dat deze technologie een grootste toekomst tegemoet gaat.

In de hedendaagse omgeving brengt het gebruik van Java nogal wat nadelen met zich mee. Allereerst vereist Java natuurlijk een browser die Java bytecode begrijpt en kan uitvoeren. Op dit moment zijn er nog geen Java enabled browsers beschikbaar voor DOS en Windows 3.x. Verder is de uitvoering van Java applets bij tijd en wijle nog ontzettend traag. Dit wordt veroorzaakt door de tijd benodigd voor het downloaden van de applet en de vertraging die het interpreteren van de tussencode met zich meebrengt. Nieuwere en betere technologie (zoals snellere netwerkverbindingen en het *Just-In-Time* omzetten van de Java bytecode naar echte machinetaal) zullen deze nadelen naar verwachting snel opheffen.

Daarnaast is en blijft beveiliging een belangrijk aandachtspunt.

## 21.12 ActiveX

ActiveX

Microsoft standaard
Afgeleid van OLE
Alleen voor 32-bit Windows OSíen
  Windows 95, Windows/NT
Downloaden en uitvoeren Windows
executable als onderdeel van de browser
<OBJECT> tag

**Notities**

Microsoft's antwoord op Java is ActiveX. Net als Java betreft het hier een faciliteit voor het over het netwerk ophalen en in de browser uitvoeren van programma's. In tegenstelling tot Java betreft het echter bij ActiveX geen executables in een neutrale tussencode maar in echte machinetaal. Omdat deze executables rechtstreeks door de processor van de klant kunnen worden uitgevoerd is de performance van ActiveX objecten aanzienlijk beter dan die van Java applets.

## 21.13 ActiveX, pro en contra

### Pro en contra

Pro:

- Alles kan met een ActiveX object (draait niet in software gevangenis)
- Snel en krachtig

Contra:

- Niet open
- Alleen Windows 95 en Windows/NT
- Beveiliging (afgezien van *code signing*)
- Heel moeilijk te programmeren

**Notities**

ActiveX objecten draaien in tegenstelling tot Java applets niet in een software gevangenis en hebben dus volledige toegang tot alle faciliteiten van de client computer. Uit beveiligingsoogpunt kunnen ActiveX controls worden gesigneerd door de leverancier zodat je zeker weet van wie de module afkomstig is. Desalniettemin is de beveiliging van ActiveX controls een “hot item”.

Alhoewel Microsoft zijn uiterste best doet om ActiveX tot algehele Internet standaard te verheffen is ActiveX momenteel volledig verbonden met de 32-bits Microsoft besturingssystemen Windows 95 en Windows/NT. Hierdoor kunnen pagina’s met ingesloten ActiveX controls alleen worden bekeken op client computers met een Win32 besturingssysteem.

# Hoofdstuk 22

# Beveiliging en het WWW

## 22.1 Beveiliging en het WWW

Beveiliging en het WWWÎú∞úbł

**Notities**

In de nuvolgende module gaan we in op de beveiligingsproblematiek die in het World Wide Web heerst.

## 22.2 Beveiligingsproblemen

Beveiligingsproblemen

WWW gebaseerd op open protocollen

Doel:

wel: informatieverspreiding

niet: veilig interactief transactiemedium

Problemen:

Privacy

Integriteit

Authenticatie

Beveiligingsgaten

**Notities**

Allereerst moet natuurlijk worden opgemerkt dat het World Wide Web (net als Internet) nooit verzonnen is voor het gebruik wat er heden ten dage van wordt gemaakt. Het WWW is indertijd opgezet voor het vrijelijk verspreiden van informatie en het algemeen toegankelijk maken van documenten. Dientengevolge speelde beveiligingsaspecten geen enkele rol bij het bedenken van de standaarden die in het web worden toegepast.

## 22.3 Clear Text

*Clear text*

WWW protocollen (met name HTTP) niet versleuteld

Iedereen kan bericht lezen (met een packet sniffer)

Berichtenverkeer kan eventueel worden gewijzigd (moeilijk maar niet onmogelijk)

**Notities**

Allereerst kent het standaard HTTP protocol geen faciliteiten voor het versleutelen van de berichten. Dit houdt in dat hackers, mits toegerust met de juiste hard- en software, al het WWW verkeer zonder probleem kunnen lezen en eventueel veranderen.

## 22.4 Authenticatie

### Authenticatie

Client en server hoeven elkaar niet te authenticeren

Met wie praat ik?

Beperkt autorisatiemechanisme:

- userid/password
- *Cookies*

Marketingdoeleinden

**Notities**

Verder hoeven een WWW client en server elkaar normaal gesproken niet te authenticeren. De client opent een verbinding en de server geeft normaal gesproken gewoon antwoord. Het HTTP/1.0 protocol kent een beperkte vorm van autorisatie waarbij de server de client een userid en password vraagt. Dit userid en password wordt dan vervolgens bij *iedere* HTTP transactie meegestuurd zodat de server een wachtwoordcontrole kan uitvoeren. Userid en password worden natuurlijk niet versleuteld meegestuurd :-(.

Omdat HTTP niet echt voorziet in client authenticatie wordt dit meestal geïmplementeerd door de web applicaties (zie onder andere het triviant voorbeeld). Goede authenticatie is niet alleen gewenst uit beveiligingsoverwegingen maar ook om reden van het verkrijgen van marketinginformatie.

## 22.5 Overige beveiligingsproblemen

Overige beveiligingsproblemen

Bugs in CGI programmaís
Windows/NT: perl.exe
Java, ActiveX, plugins en helpers

**Notities**

Naast het ontbreken van structurele encryptie en authenticatie kent het web nog talloze andere beveiligingsproblemen. Meestal vloeien deze voort uit slecht geïmplementeerde web applicaties of bugs in WWW programmatuur. Legendarisch is het ‘perl.exe’ probleem. Hierbij zetten nietsvermoedende webmasters een kopie van perl.exe (de NT interpreter voor perl programma’s) in de HTTP document directories ter ondersteuning van het uitvoeren van perl CGI programma’s. Door het handmatig manipuleren van de opgevraagde URI kan een inbreker echter zonder problemen perl.exe opstarten met ieder gewenst commando. Hierdoor ligt de Windows/NT gebaseerde web server van voor tot achter open!

## 22.6 Oplossingen

Oplossingen

S-HTTP
SSL

**Notities**

De structurele oplossingen voor de WWW beveiligingsproblemen wordt geboden door het gebruiken van andere protocollen dan het standaard HTTP protocol. De twee meest geschikte kandidaten hiervoor zijn S-HTTP en SSL.

## 22.7 S-HTTP

S-HTTP

Secure HTTP
TIS (Trusted Information Systems)
Encryptie en authenticatie op HTTP berichtniveau
Niet/nauwelijks geimplementeerd

**Notities**

S-HTTP (Secure HTTP) is een uitbreiding van het HTTP protocol met mogelijkheden voor encryptie en authenticatie. Qua implementatie is S-HTTP verwant aan de Internet email standaard PEM (Privacy Enhanced Mail). Net als PEM wordt ook S-HTTP nauwelijks toegepast.

Bij S-HTTP authenticeren client en server elkaar door het uitwisselen van certificaten met publieke sleutels. Het HTTP bericht wordt vervolgens met behulp van symmetrische encryptie versleuteld. De sessiesleutel wordt onder bescherming van public key encryption meegestuurd. S-HTTP is ontwikkeld door Trusted Information Systems Inc. Voor meer informatie zie http://www.tis.com.

## 22.8 SSL

SSL

*Secure Socket Layer*
Encryptie op niveau van de verbinding
Ondersteunt:
Authenticatie,Encryptie
Compressie
Wijd geimplementeerd
Internationaal alleen zwakke encryptie
Kan ook worden gebruikt voor andere toepassingen

**Notities**

De meeste populaire WWW beveiligingstechnologie is SSL, de *Secure Socket Layer*.

SSL is gebaseerd op Berkeley Sockets (BSD sockets). Deze Berkeley sockets is een Application Programmer Interface (API) voor programmeurs die netwerkverbindingen op willen bouwen. BSD sockets ontlenen hun naam aan de universiteit in Berkeley (California) die het concept van de BSD sockets heeft ontwikkeld en geïmplementeerd. BSD sockets zijn intussen de standaard API voor het opbouwen van TCP/IP verbindingen. Dez standaard wordt bijvoorbeeld ondersteund door alle UNIX leveranciers en Microsoft (WINSOCK.DLL is gebaseerd op Berkeley sockets).

De standaard sockets bestaat uit een bibliotheek van aanroepen zoals connect(), accept(), send() en receive() waarmee programmeurs vanuit hun applicaties netwerkverbindingen kunnen opbouwen. De socketlaag van het operating systeem bouwt op basis van die aanroepen TCP/IP pakketjes en stuurt die het netwerk in. BSD sockets vormen een programmeur interface op het operating systeem en bieden in principe geen functies (zoals encryptie) die niet in TCP/IP zijn voorzien.

De door Netscape Communications Inc. voorgestelde Secure Socket Layer is een laag bovenop de standaard BSD sockets. Deze extra laag verzorgt functies zoals encryptie en authenticatie. De programmeur roept in principe geen functies meer aan in de BSD socket bibliotheken, maar maakt alleen nog maar gebruik van de SSL versies van die aanroepen: SSLaccept(), SSLconnect(), SSLsend() en SSLreceive().

Bij het opzetten van een SSL verbinding (met de functies uit de SSL libraries) wisselen de SSL-lagen in de client en de server eerst wat informatie uit omtrent elkanders identiteit, de te gebruiken compressie/encryptie en welke sleutels moeten worden gebruikt (de SSL handshake). De verbinding tussen twee SSL partners wordt door de

SSL laag gecomprimeerd en versleuteld. SSL kan gebruik maken van een groot aantal (symmetrische) encryptie algoritmes. De sessie sleutel wordt random gegenereerd (door de client) en onder public key encryption uitgewisseld. SSL ondersteunt de authenticatie van SSL- partners door het uitwisselen van getekende certificaten. SSL 2.0 ondersteunt alleen server authenticatie (met server certificaten), SSL 3.0 ondersteunt ook client authenticatie (met client certificaten).

Een SSL verbinding komt als volgt tot stand:

1. De client opent de verbinding en stuurt een *client hello*. In dit bericht staat (onder andere) welke encryptie algoritmes de client begrijpt.

2. De server accepteert de verbinding en stuurt een *server hello*. Hierin staat ondermeer welk encryptiealgoritme de server met de client wil praten.

3. Onmiddelijk hierna stuurt de server een server certificaat. Hiermee geeft de server aan wat zijn public key is.

4. De client kan nu een client certificate sturen (SSL 3.0). Deze bevat de public key van de client.

5. De client genereert een random sessie key, encrypt deze met de public key van de server (uit het server certificaat) en stuurt die op. Dit is de sleutel waarmee de sessie zal worden versleuteld (client key exchange en server key exchange berichten).

6. De client genereert vervolgens een MD5 hashcode over alle handshake meldingen, signeert die met zijn eigen private key en stuurt die naar de server. Met dit *certificate verify* bericht kan de server controleren of de client de rechtmatige bezitter is van de public key (expliciete verificatie, alleen in SSL 3.0).

De veiligheid van SSL is gebaseerd op het feit dat de client een random sessie sleutel genereert en deze public key encrypted opstuurt. De zwakke schakel hierin is dat de client de public key van de server moet kennen. Deze public key wordt in stap 3 van de SSL handshake overgezonden. De zwakke schakel hierin is dat een ‘valse’ server natuurlijk ook een vals certificaat zal opsturen. Hetzelfde probleem doet zich in SSL 3.0 voor bij client certificates.

Dit probleem wordt opgelost doordat de public keys in een certificaat zijn gesigneerd met de private key van algemeen bekende en vertrouwde Certificate Authorities (CA). Een SSL partner vertrouwt een certificaat als dit is getekend door een CA die hij reeds kent. Dit betekent dat de public key van die CA dus onomstotelijk bekend moet zijn! Dit wordt in het World Wide Web opgelost door de public keys van de bekendste CA’s met de browsers mee te installeren. De meeste WWW-browsers kennen dus de public keys van de bekendste CA’s. Op basis hiervan vertrouwen ze de door die CA’s getekende certificaten die ze aangeleverd krijgen van servers. Een nieuwe server moet dus een sleutelpaar genereren en dat door een bekende CA laten tekenen.

Als je een certificaat krijgt aangeleverd (‘server certificate’ bericht) wat door een onbekende CA is getekend dan heb je twee keuzes:

1. De server niet vertrouwen en de verbinding verbreken.

2. De server wel vertrouwen (op basis van ????? niets!)

De Netscape browsers (3.0 en hoger) geven de gebruiker de keuze om de server wel of niet te vertrouwen. De Microsoft Internet Explorer vertrouwt alleen ‘erkende’ certificaten en vertrouwt andere CA’s per definitie niet!

Netscape heeft een referentie implementatie van SSL die door derde partijen in licentie kan worden genomen. Hiermee kunnen andere bedrijven hun software SSL compatible maken. De Amerikaanse exportbeperkingen zorgen ervoor dat SSL alleen mag worden geëxporteerd met zwakke encryptie algoritmes. Naast Netscape’s implementatie is er ook een public domain implementatie van SSL. De implementatie: SSLeay is ontwikkeld door Eric Young uit Queensland, Australia. SSLeay mag vrijelijk worden gebruikt en ondersteunt ook sterke algoritmes zoals IDEA.

## 22.9 Implementatie van SSL

### Implementatie van SSL

URL: https://www.securesite.com/....

Well known port: 443

Netscape en SSLeay (public domain)

Moet worden ondersteund door client (browser) en HTTP server

Medewerking van *Certifying Authority* voor afgifte certificaten

**Notities**

Een browser die SSL ondersteunt weet dat hij contact op moet nemen via een SSL bbeschermde verbinding als hij in een URL het protocol *https* tegenkomt. Op dat moment neemt hij contact op met de in de URL genoemde server op de well known port voor HTTPs (443, of natuurlijk met een andere in de URL genoemde poort). Één van de fraaie eigenschappen van SSL is dat het een presentatie-laag bescherming is rondom de standaard protocollen. Aan het HTTP protocol hoeft dus niets te worden gewijzigd! Het belangrijkste probleem op dit moment met de algehele implementatie van SSL is dat de Amerikaanse overheid geen export van sterke encryptie toestaat.

# Hoofdstuk 23

# Overige onderwerpen

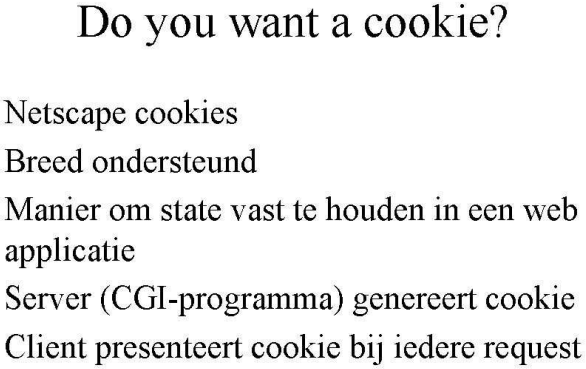
## 23.1 Overige onderwerpen

Overige onderwerpen

**Notities**

In deze module komen nog wat overige onderwerpen rondom het World Wide Web aan de orde.

## 23.2 Do you want a cookie?




**Notities**

Zoals reeds vele malen in deze shop aan de orde is gekomen is het HTTP protocol toestandsloos: de HTTP server onthoudt geen informatie van verbinding tot verbinding. Voor web applicaties is dit een groot nadeel. Netscape heeft ter verlichting van de problemen een mechanisme geïntroduceerd waarmee web applicaties in samenwerking met WWW browsers toestand kunnen onthouden van verbinding tot verbinding. Deze zogenaamde *Netscape cookies* worden inmiddels door een groot aantal web browsers ondersteund.

Applicaties die toestandsinformatie willen onthouden hebben bij volgende verbindingen van dezelfde client een identificerend element nodig om de opgeslagen (of onthouden) toestand weer te aktiveren. Het is namelijk in het web zeer wel mogelijk dat een applicatie door zeer veel gebruikers tegelijk wordt gebruikt. In dat geval is het nodig dat die applicatie van meerdere sessies tegelijk de toestand kan bewaren.

Op het moment dat de applicatie de eerste transactie van een nieuwe sessie ontvangt (meestal via het login scherm van de web applicatie) genereert de applicatie een uniek *cookie*. Zo'n cookie is een tekststring van onbepaalde lengte. Meestal bevat een cookie de datum, tijd en gebruikersnaam die met de sessie worden geassocieerd. De web applicatie geeft de cookie vervolgens terug aan de browser in de vorm van een HTTP response header "Set-Cookie". De browser zal vervolgens bij iedere request naar die server het cookie weer meesturen in een HTTP request header. De web applicatie kan het aldus ontvangen cookie gebruiken om de toestand van de sessie op te zoeken en de request te verwerken.

Cookies kennen ook een vervaldatum. Als deze is verstreken wordt de cookie niet meer meegezonden en kan (moet) de web applicatie (als hij daar zin in heeft) een nieuw coo-

kie genereren. Cookies kunnen ook worden toegepast om een gebruiker onmiddelijk weer te herkennen om bijvoorbeeld speciale aanbiedingen te doen of persoonlijke instellingen te herstellen. Cookies worden meestal door de browser opgeslagen op de disk van de gebruiker.

## 23.3 VRML

VRML ˛Î9◊

*Virtual Reality Modelling Language*

Markup language voor 3D werelden

Browser staat gebruiker toe om in 3D wereld rond te wandelen

Meestal geimplementeerd als plug-in of helper applicatie

Moeilijk te ontwikkelen

**Notities**

Een “hot item” van enige tijd geleden, maar intussen weer wat minder prominent aanwezig, is de *Virtual Reality Modelling Languae*, VRML (of, zoals collega Theo de Ridder zegt: Frummel). VRML is een opmaaktaal voor driedimensionale werelden. In een ASCII bestand (een zogenaamd WRL bestand) geven we met VRML instructies weer hoe de wereld in elkaar zit, waar bepaalde objecten zich bevinden en waar het licht vandaan komt. VRML interpreters staan toe dat de gebruiker zich met het toetsenbord of met de muis door de 3D wereld beweegt.

In het World Wide Web worden VRML werelden geïnterpreteerd door helper applicaties of plug-ins. Er zijn zelfs enkele implementaties die VRML, Java en HTML met elkaar verweven tot een driedimensionaal, interactief informatielandschap. Het belangrijkste probleem met VRML is dat het weergeven van een grote, realistische, wereld tamelijk moeilijk is voor zowel de VRML-programmeur als de client computer. VRML zou op zich ideaal zijn voor het weergeven van een electronische winkel of het opzoeken van informatie in een virtuele bibliotheek.

## 23.4 HyperWave

HyperWave

*Next Generation WWW*
Universiteit van Graz
Uitbreidingen op WWW:
- versiebeheer
- link management
- toegangscontrole
- zoekmogelijkheden

Aparte client en server nodig

**Notities**

Terwijl de meeste inwoners van de wereld nog niet toe zijn aan het gebruik van het hedendaagse World Wide Web wordt aan de universiteit van Graz al vrolijk geknutseld aan het web van de toekomst. Aldaar is onder de bezielende leiding van Hermann Mauer de opvolger van het World Wide Web ontwikkeld: HyperWave (voorheen: HyperG). HyperWave stelt zich tot doel een groot aantal van de problemen van het World Wide Web op te lossen. Zo kennen HyperWave browsers en servers faciliteiten voor versiebeheer van documenten, het beheren van hyperlinks (geen blinde links meer) en ingebouwde zoekmogelijkheden.

Om HyperWave te kunnen bedrijven is een speciale client (browser) en server benodigd. Deze server gedraagt zich ook als een normale HTTP server en kan dus ook door standaard browsers worden benaderd, maar deze kunnen natuurlijk niet de verbeterde HyperWave faciliteiten gebruiken. De universiteit van Graz heeft een HyperWave server en een aantal HyperWave clients ontwikkeld.

## 23.5 Web robots

Web robots

Programmaís die het web onderzoeken

Web slurping (htget)

Zoeken

Indexeren

**Notities**

Een ander fenomeen in het World Wide Web is het bestaan van zogenaamde *webbots*, ook wel “web robots” genoemd. Het betreft hier programma’s die contact opnemen met HTTP servers en geautomatiseerd documenten ophalen en verwerken. Dergelijke programma’s zijn bijvoorbeeld in staat om hele web sites te kopiëren, documenten te vinden die voldoen aan een bepaald sleutelwoord of om indexen op te bouwen van documenten. De bekende zoek servers van Yahoo en Lycos gebruiken onder andere web robots om hun informatie te verzamelen.

Nieuwe ontwikkelingen zijn intelligente web robots die er door hun meester op uit worden gestuurd om alle documenten te vinden die aan bepaalde voorwaarden (sleutelwoorden) voldoen. Deze programma’s draaien volcontinu geven een waarschuwing zodra ergens nieuwe documenten zijn gevonden die aan de criteria voldoen. Op die manier kunnen gebruikers bijvoorbeeld automatisch worden geattendeerd op nieuwe ontwikkelingen op een bepaald gebied.

## 23.6 Proxy servers

Proxy servers

Intermediairs in de WWW transacties
Maken lokale kopieen van documenten
Brengt het aantal Internet verbindingen terug
Kans op (ietwat) verouderde informatie

**Notities**

De praktijk wijst uit dat het vaak voorkomt dat gebruikers in een organisatie of van een provider regelmatig dezelfde documenten ophalen. De individuele browsers hebben hier natuurlijk geen weet van en halen ieder voor zich de informatie van de WWW servers op. Om dit proces ietwat te stroomlijnen is het mogelijk om één of meer *proxy servers* te installeren. Dit zijn speciale server applicaties die alle HTTP requests van een bepaald (deel)netwerk op zich afgeschoten krijgen (de browsers moeten wel expliciet worden geconfigureerd om van de proxy server gebruik te maken). HTTP proxy servers houden een cache bij van populaire documenten zodat die zonder Internet verkeer te veroorzaken door de lokale gebruikers kunnen worden geraadpleegd.

Zodra hij een request ontvangt kijkt de proxy server of hij al een lokale kopie van het gevraagde document bezit. Is dit het geval dan retourneert hij deze lokale versie. Is de lokale kopie verouderd of heeft hij geen lokale kopie dan wordt het document door de proxy server bij de juiste HTTP server opgehaald. Indien deze HTTP server caching toestaat (wordt meegegeven in de HTTP response headers) maakt de proxy server gelijk een lokale kopie van het document. Als andere gebruikers het document ophalen krijgen deze vervolgens de zojuist opgehaalde lokale kopie te zien.

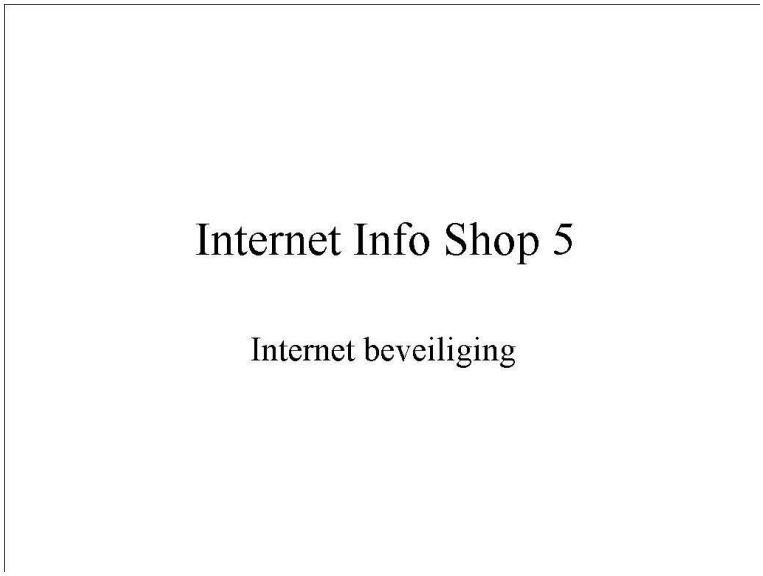
# Deel V

# Internet beveiliging

# Hoofdstuk 24

# Introductie

## 24.1 Welkom

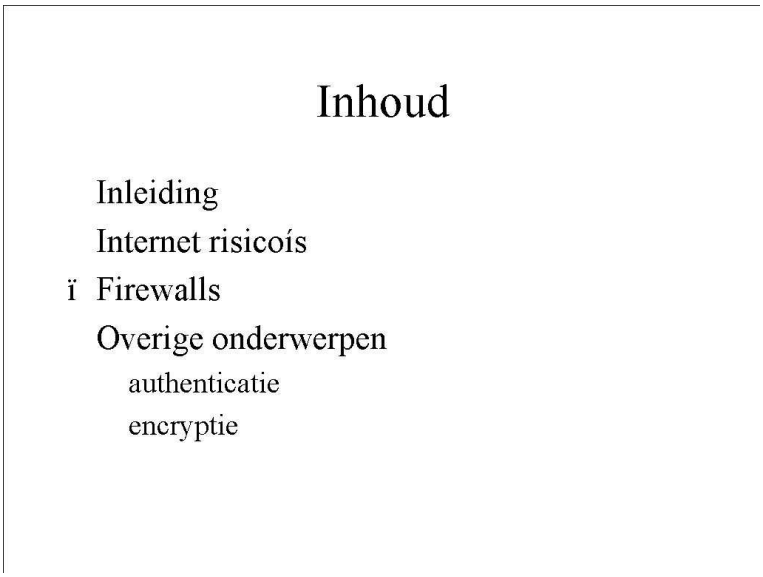



**Notities**

Welkom bij de vijfde ABN AMRO Internet Information Shop. Deze shops is geheel gewijd aan *Internet beveiliging*.

Dit document bevat bij tijd en wijle materiaal met een diepgaand technisch karakter. Dit kan helaas niet worden voorkomen omdat een groot gedeelte van de Internet beveiligingsproblematiek nu eenmaal technisch van aard is.

## 24.2 Inhoudsopgave

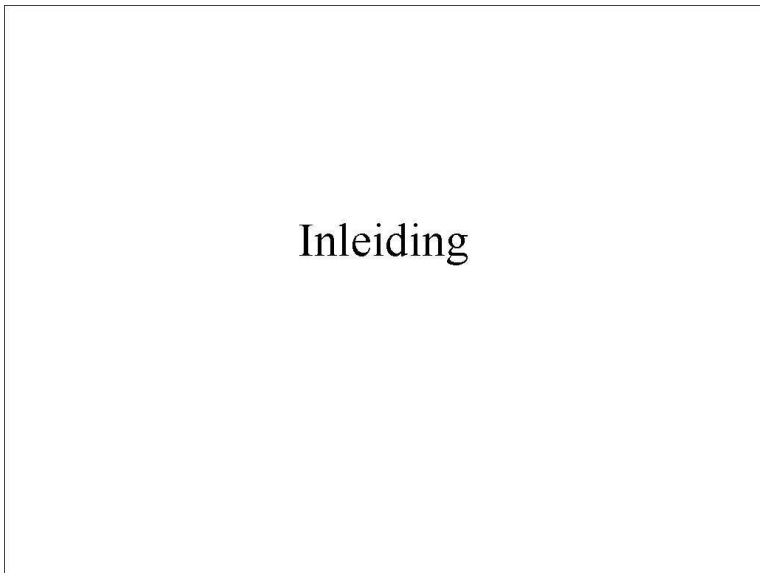



**Notities**

# Hoofdstuk 25

# Inleiding

## 25.1 Inleiding




**Notities**

Enige tijd geleden sprak ik met een collega over de beveiliging van computers en netwerken. Hij vertelde mij dat de Binnenlandse Veiligheidsdienst pas recentelijk beveiligingsprogrammatuur had genstalleerd op hun belangrijkste computers. Toen ik mijn verbazing hierover kenbaar maakte wees hij me op de fysieke beveiligingsmaatregelen die de BVD had getroffen. De computers in kwestie waren niet via netwerken verbonden. Ook waren er geen inbelvoorzieningen aanwezig. De enige aangesloten terminals waren rechtstreeks met de machines verbonden en stonden (net als de computers zelf) in een zwaar beveiligde gepantserde kluis met maar één toegangsdeur.

Deze opzet voldoet vrijwel volledig aan de waarneming van de bekende Amerikaanse beveiligingsspecialisten Grampp en Morris:

*"It is easy to run a secure computer system. You merely have to disconnect all dial-up connections and permit only direct-wired terminals, put the machine and its terminals in a shielded room, and post a guard at the door."*

De meeste computers worden echter niet op die wijze gebruikt. Lokale netwerken verbinden de computers van een organisatie met elkaar. Wide Area Networks, zoals Internet, verbinden vervolgens de netwerken (en dus de computers) van de aangesloten organisaties.

Netwerken brengen om drie redenen beveiligingsrisico's met zich mee:

1. Eenmaal aangesloten op een netwerk zijn er meer punten van waaruit een aanval op de computer kan worden uitgevoerd.

2. Het is moeilijk om berichten die vanuit het netwerk arriveren op echtheid te

controleren. Het is niet zonder meer met zekerheid te zeggen dat een bericht afkomstig is van de vermeende afzender. Vervolgens is het ook niet zeker dat een bericht dat je uitstuurt alleen door de geadresseerde wordt gelezen.

3. Via netwerken kan contact worden gemaakt met een groot aantal verschillende applicaties. Inbrekers hebben dan ook meerdere wegen langs welke ze kunnen in breken. Alle applicaties dienen apart te worden beveiligd.

Bovenstaande problemen zijn met name aanwezig op Internet. Om een aantal redenen heeft Internet een slechte naam gekregen op het gebied van beveiliging. Het is dan ook zaak dat organisaties die aan Internet zijn aangesloten de beveiliging van hun systemen en netwerken hoog in het vaandel hebben.

In deze info shop staat Internet beveiliging centraal. We gaan uitgebreid in op de organisatorische en technische aspecten van de belangrijkste Internet beveiligingsvraagstukken. Beveiligingsaspecten die te maken hebben met de fysieke integriteit van systemen komen niet aan de orde.

N.B.: We hebben ervoor gekozen computer inbrekers aan te duiden met de in de populaire pers gebruikelijke term *hacker*. Deze aanduiding is echter in wezen incorrect. *Hacker* is van oorsprong een benaming voor de uiterst creatieve programmeurs die in zeer korte tijd inventieve programma's schrijven. *Hacking* heeft dan ook niets met inbraak of aanverwante illegale aktiviteiten te maken. De afgelopen jaren is het predikaat *hacker* door de massamedia geassocieerd met computer inbrekers. Op gelijksoortige wijze wordt de kreet *hacking* nu algemeen gebruikt om het proces van inbreken in computers en netwerken aan te duiden. Er zijn diverse pogingen gedaan om de term *hacker* in zijn oorspronkelijk betekenis te herstellen (o.a. door invoering van *cracker* als aanduiding voor de inbrekende hacker). Deze pogingen hebben echter weinig succes gehad. Wij sluiten ons dan ook aan bij de algemeen gangbare terminologie.

Beveiliging is een zeer breed begrip. De definitie van beveiliging uit het woordenboek is "beschermen, beschutten". Kort gezegd komt beveiliging dus neer op het beschermen van hulpbronnen tegen ongewenste gebeurtenissen. In een continu veranderende omgeving als Internet kan een eenmaal aangebrachte beveiliging nooit een vast gegeven zijn. Beveiliging is een aktiviteit en verdient derhalve permanente aandacht.

## 25.2 Wat gaan we beveiligen


**Notities**

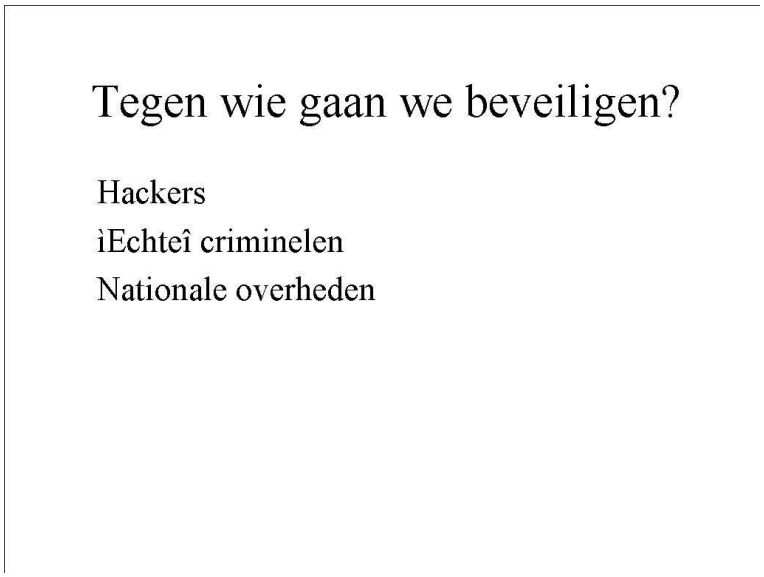
Alvorens effectieve beveiligingsmaatregelen kunnen worden genomen is het natuurlijk van groot belang eerst een overzicht te hebben van de hulpbronnen die moeten beveiligen.

- Gegevens
  In vrijwel alle gevallen komt beveiliging vandaag aan de dag neer op het beveiligen van gegevensbestanden tegen ongewenst inkijken, kopiren en wijzigen. Normaal gesproken verdienen voornamelijk de bestanden die zijn opgeslagen op de harde schijven van computers de aandacht. Een op Internet aangesloten netwerk dient dusdanig te zijn beveiligd dat het voor ongeautoriseerde buitenstaanders onmogelijk is niet vrijgegeven bestanden te lezen of aan te passen. Organisaties transporteren echter steeds vaker gegevens via elektronische weg naar medewerkers of collega's (bijvoorbeeld via elektronische post of het World Wide Web). Vanzelfsprekend dienen de gegevens ook tijdens transport (bijvoorbeeld via email) te zijn beschermd tegen ongewenste toegang.

- Computer hulpbronnen
  Een ander te beschermen goed wordt gevormd door de fysieke computer hulpbronnen van een organisatie. Via zogenaamde *denial of service* aanvallen kunnen hackers trachten de computers van een organisatie lam te leggen. Een bekend voorbeeld hiervan is het volledig opslurpen van vrije diskruimte door het sturen van mail bombs (elektronische postberichten van enkele megabytes groot). Ook het stelen van computertijd (door het ongeautoriseerd uitvoeren van programma's) valt hieronder.

- Identiteit en oorsprong
  Een minder voor de hand liggend goed wat op Internet moet worden beschermd is de identiteit van een organisatie. De open structuur van Internet maakt het mogelijk om (elektronische post-) berichten te versturen met een valse afzender. Dit betekent dat men in principe niet op de ogenschijnlijke afzender van een bericht kan vertrouwen. We moeten dus op één of andere wijze voorkomen dat derden namens ons berichten kunnen waarvan de afzender niet kan vermoeden dat er valsheid in geschrifte in het spel is.

## 25.3 Tegen wie gaan we beveiligen



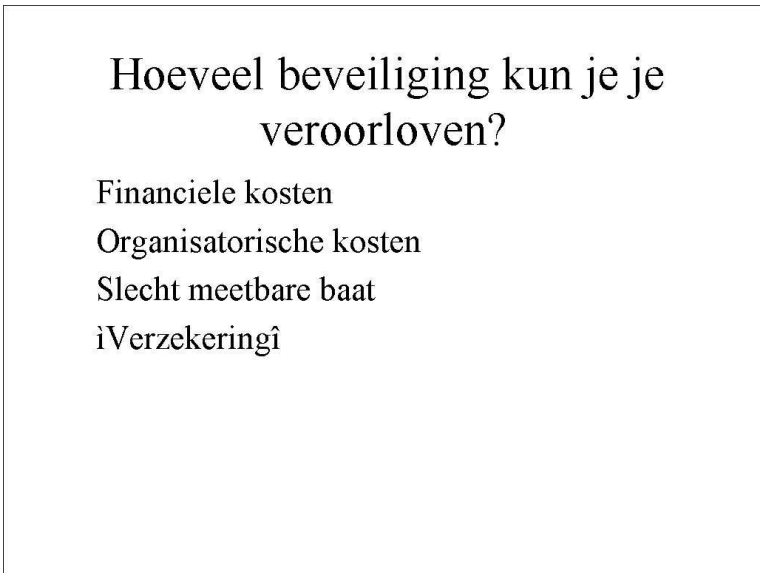

**Notities**

Onderdeel van een goed beveiligingsplan is een analyse van de mogelijke tegenstanders, hun motivatie en de tot hun beschikking staande hulpbronnen. De mate van beveiliging die een organisatie zich aan moet meten wordt natuurlijk gedeeltelijk voorgeschreven door de kracht van de potentiële aanvallers. Maatregelen die afdoende beschermen tegen een student met een PC en een modem zijn vrijwel nutteloos indien de aanvaller uit de kringen van de georganiseerde misdaad komt!

De authentieke Internet hacker is een mannelijke student die vanuit de computerzaal van een universiteit of hogeschool het wereldwijde Internet afschuimt op zoek naar servers waar kan worden ingebroken. Eenmaal binnen kijkt hij wat rond, laat een bericht achter voor de systeembeheerder en verdwijnt weer. Recente ontwikkelingen doen vermoeden dat dit een uitstervend soort is. Onderzoeken wijzen uit dat er tegenwoordig sprake is van een redelijk georganiseerd computer underground die inbreken vanuit ideologische of commerciële achtergronden. Naarmate meer bedrijven zich op Internet gaan manifesteren kan ook industriële spionage via Internet een probleem gaan worden.

Ook nationale overheden gaan een deuntje meeblazen. Een recent in de Verenigde Staten aangenomen wet (de Digital Telephony Bill) heeft onder andere tot gevolg dat Internet providers het voor de FBI mogelijk moeten maken om (indien voorzien van gerechtelijke toestemming) Internet verkeer af te tappen.

## 25.4 Hoeveel beveiliging kun je je veroorloven




**Notities**

Beveiliging kost geld. De belangrijkste kostenposten worden gevormd door de noodzakelijke hardware, software en inspanning die nodig zijn om een Internet aansluiting afdoende te beveiligen. Het vervelende van beveiligingskosten is echter dat er een slecht meetbare baat mee wordt gerealiseerd. Het is moeilijk, zo niet ondoenlijk, om de mate van beveiliging financieel inzichtelijk te maken. De gebruikelijke methoden voor het meten van de opbrengsten van investeringen gaan dus niet op voor beveiligingsinvesteringen. Een goede methode voor het beoordelen van de kosten van Internet beveiliging kan wellicht worden gedestilleerd uit de risicoanalyse en premieberekeningsmethoden die in de verzekeringswereld worden gebruikt.

Naast direkte financiële kosten brengt beveiliging ook organisatorische kosten met zich mee. Hoe beter een netwerk is beveiligd, hoe moeilijker het ook voor de geautoriseerde gebruikers is om er gebruik van te maken. Gebruiksgemak en (daardoor) produktiviteit hebben relatief zwaar te leiden onder beveiliging. Het is bekend dat zwaar beveiligde systemen dermate onwerkbaar kunnen worden dat de gebruikers trachten de beveiligingsmaatregelen te omzeilen of buiten werking te stellen. Hiermee wordt de beveiliging van binnen uit aangevreten.

Te veel beveiliging is net zo slecht als te weinig beveiliging!

## 25.5 Beveiliging is geen projekt!


**Notities**

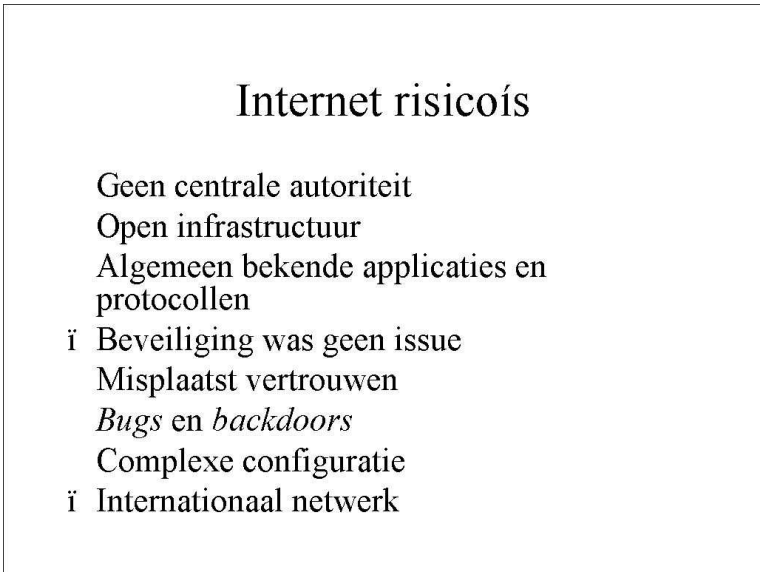
Nog veel te vaak gieten organisaties elementaire systeem- en netwerkbeveiliging in projektvorm.

*Beveiliging is echter een doorlopende aktiviteit. Beveiliging is geen projekt, het is een onderdeel van ieder ander projekt.*

Kenmerkend voor een projekt is dat er sprake is van een duidelijk omschreven doel en een eindige projektduur. Beveiliging stopt echter nooit en verdient permanente aandacht.

Het opzetten van een degelijk beveiligde omgeving kan natuurlijk wel in projektvorm. Daarna zal bij iedere wijziging in de infrastructuur (hard- en software) de impact op de beveiliging moeten worden bepaald. Het beschikbaar komen van nieuwe versies van programmatuur, nieuwe Internet verbindingen en de implementatie van een nieuwe elektronische dienst zijn allemaal voorbeelden van implementaties die significante consequenties voor de beveiliging kunnen hebben. Om die reden zou er eigenlijk in ieder automatiseringsprojekt een beveiligingshoofdstuk moeten zitten waarin de beveiligingsconsequenties worden bekeken en waar, indien nodig, aanvullende beveiligingsmaatregelen worden beschreven.

## 25.6 Internet risico's




**Notities**

Het aansluiten van een netwerk aan een publiek toegankelijk WAN (Wide Area Network) brengt altijd bepaalde beveiligingsrisico's met zich mee. Om een aantal redenen zijn de risico's van de aansluiting op Internet groter dan in het geval van een aansluiting op andere publieke netwerken zoals Datanet-1 (X25 netwerk van de PTT) en POTS (Plain Old Telephone System). Met name het open karakter van Internet brengt de nodige beveiligingsrisico's met zich mee.

- Geen centrale autoriteit
  Internet wordt gekenmerkt door een open structuur. Er is geen centrale instantie of autoriteit die de integriteit van de infrastructuur controleert en waarborgt. De communicatie op Internet komt tot stand door vrijwillige ondersteuning van algemeen aanvaarde standaarden. Het is voor kwaadwillende gebruikers relatief eenvoudig mogelijk die standaarden te omzeilen met als doel de beveiliging van een systeem of netwerk te doorbreken. Er is (nog) geen Internet politie die op eigen initiatief overtredingen van de "regels" opspoort en vervolgt. Als een inbreker een bericht verstuurt met een valse afzender is er in principe niemand die dat detecteert.

- Open infrastructuur
  De Internet infrastructuur wordt gevormd door de gezamenlijke netwerken van de providers. De meeste providers hebben her en der verspreid in het land Internet knooppunten (zogenaamde Points of Presence (PoP)). De informatiestromen van de klanten van de provider loopt over die knooppunten naar de eindbestemming (wellicht door het netwerk van andere providers). Er zijn gevallen bekend van hackers die inbraken op systemen van een provider en die daar speciale pro-

grammatuur installeerden (zogenaamde Sniffers) waarmee het netwerkverkeer wat door de netwerken van de provider loopt kan worden afgeluisterd. Omdat vrijwel alle Internet applicaties hun informatie onversleuteld doorsturen kan de inbreker op deze wijze gevoelige informatie, zoals wachtwoorden en elektronische post, ongestoord afluisteren. Een hiervoor uiterst geschikte applicatie wordt standaard meegeleverd met de System Management Server van Microsoft. Deze "Network Monitor" is uitermate krachtig en zeer eenvoudig (op afstand) te bedienen.

- Algemeen bekende applicaties en protocollen
  Op Internet wordt voor het grootste gedeelte gebruik gemaakt van algemeen bekende protocollen en applicaties. De ins en outs van de gebruikte programmatuur zijn dan ook volledig bekend. *Security by Obscurity* is dan ook geen realistische mogelijkheid in een Internet omgeving. Meestal kan de complete werking van een Internet protocol of applicatie worden gevonden in een Request For Comment (RFC) document. Dit houdt in dat eventuele beveiligingsfouten in die applicaties ook algemeen bekend zijn. Er circuleren dan ook uitgebreide lijsten met security holes van veel toegepaste operating systemen en applicaties. Op basis van een dergelijke checklist hebben de Internet deskundigen Dan Farmer en Wietse Venema het programma SATAN ontwikkeld. SATAN is in staat om volautomatisch een netwerk te onderzoeken op gaten in de beveiliging.

- Beveiliging was geen issue
  Internet is het resultaat van een researchprojekt. De nadruk bij de ontwikkeling van het netwerk en de meeste applicaties heeft gelegen bij het op eenvoudige wijze beschikbaar maken van informatie. De beveiliging van systemen en verbindingen heeft nooit een hoofdrol gespeeld bij het ontwerp van Internet en Internet applicaties. Hierdoor bevatten een groot aantal Internet protocollen features die:

  – relatief eenvoudig kunnen worden misbruikt, of
  – moeilijk zijn af te schermen voor indringers van buitenaf

  Zo bevat vrijwel geen enkele Internet protocol of applicatie ingebouwde mogelijkheden voor het versleutelen van informatie. Daar waar applicaties wel ingebouwde beveiligingsmogelijkheden hebben is de default instelling vaak zo dat alles open staat. Een voorbeeld hiervan is het Network File System (NFS) pakket van Sun. Via NFS kunnen computers elkaars schijven over een netwerk (dus ook over Internet) benaderen. Als de systeembeheerder bij het configureren van NFS niet expliciet de toegang beperkt, hebben alle systemen lees- en schrijfrechten op de geëxporteerde schijven.

- Misplaatst vertrouwen
  Een groot aantal Internet applicaties staan bepaalde akties toe op basis van het systeem waarvandaan zo'n aktie wordt uitgevoerd (of de afzender van het bericht). Er is dan een expliciete vertrouwensrelatie tussen die twee systemen. Een voorbeeld hiervan is het UNIX commando rsh (Remote Shell). Met rsh kan een UNIX commando worden uitgevoerd op een ander systeem in het netwerk, zonder dat daarvoor een wachtwoord hoeft te worden opgegeven! Het ontvangende systeem voert het opgestuurde commando zonder meer uit, als de afzender van het commando maar een bekend (en vertrouwd) systeem is. Één van de problemen hiermee is dat het in Internet mogelijk is om de afzender van een bericht te

simuleren. Hierdoor kan een inbreker doen voorkomen dat het commando vanaf een vertrouwd systeem afkomstig is. Een ander probleem is dat het vaak mogelijk is om een default instelling te configureren waarmee impliciet alle systemen worden vertrouwd! In het voorbeeld van rsh kan door het opnemen van twee plussen ("+ +") in het juiste configuratiebestand de rsh functionaliteit worden opengezet vanaf alle systemen (één van de zaken waar SATAN op controleert).

- Insekten en achterdeuren (bugs and backdoors)
  De verzameling applicaties die van Internet gebruik kan maken voor het verzenden van commando's en informatie is in de loop der jaren in elkaar geknutseld door een zeer gevarieerde groep van programmeurs, studenten, afstudeerders, onderzoekers, leveranciers en hobbyisten. In veel gevallen is het originele programma vele malen uitgebreid en/of aangepast. Hierdoor zijn in menige applicatie fouten geslopen die kunnen worden aangewend om de beveiliging van het systeem of netwerk te doorbreken. Ook was (en is) het niet ongebruikelijk dat programmeurs verborgen features (zogenaamde backdoors) inbouwen. Zo'n achterdeur heeft vaak de vorm van een geheim commando of toetscombinatie waarmee de beveiliging kan worden omzeild. Één van de eerste releases van Sendmail (de meest gebruikte applicatie voor het ontvangen en doorsturen van electronische post), bevatte bijvoorbeeld een niet gedocumenteerd DEBUG commando. Na het ingeven van het DEBUG commando (wat door iedereen kon worden gedaan) konden systeemcommando's worden gegeven die met de hoogst mogelijke systeemautoriteit werden uitgevoerd. Na het ontdekken van fouten of achterdeuren wordt natuurlijk onmiddellijk een verbeterde versie van het programma verspreid. Desalniettemin blijven de verouderde varianten van die programmatuur nog jaren in omloop.

- Veel mogelijkheden, complexe configuratie
  De researchmentaliteit die een groot gedeelte van de ontwikkeling van Internet stuurt, heeft applicaties opgeleverd die zeer dynamisch, flexibel en veelzijdig zijn. Veel van deze applicaties kennen uiterst complexe configuratiebestanden waarmee de functionaliteit van het programma op zeer gedetailleerd niveau kan worden gestuurd. Door de complexiteit van deze configuratiebestanden is de kans relatief groot dat ongemerkt gaten in de beveiliging worden geschoten. Het voorbeeld bij uitstek hiervan is het reeds genoemde Sendmail programma. De Sendmail configuratie file is dermate complex dat er in Nederland waarschijnlijk maar vijf mensen zijn die de mogelijkheden volledig begrijpen. Ter illustratie: het handboek Sendmail in a Nutshell bevat ongeveer 900 (!) pagina's. Ongeveer eens per maand verschijnen er berichten op Internet dat door een bepaalde combinatie van Sendmail parameters een gat in de beveiliging ontstaat.

- Internationaal netwerk
  Internet is een internationaal netwerk. Hierdoor is de opsporing, aanhouding en berechting van computerinbrekers een moeilijke zaak. Diverse verslagen van Internet inbraken beschrijven een moeizame internationale jacht op inbrekers. De verschillende wetgeving op het gebied van strafvordering (aanhouding, huiszoeking, afluisteren) en strafrecht (wat is strafbaar) kunnen ervoor zorgen dat een inbreker door de mazen van de wet heen kunnen glippen. Een groep Nederlandse studenten heeft een tijd lang regelmatig ingebroken op computersystemen van Amerikaanse universiteiten en AT&T. Omdat hacking toen nog niet strafbaar was in Nederland weigerde het ministerie van Justitie haar medewerking te verlenen

aan de opsporing en vervolging. Het verhaal gaat dat de guiten hun aktiviteiten pas staakten toen een FBI-agent rechtstreeks contact opnam met één van hun moeders om uit te leggen wat haar zoon allemaal uitspookte in zijn vrije tijd.

## 25.7 Beveiliging en de organisatie

Beveiliging en de organisatie

Houding
*Social engineering*
Kennis
Beveiligingsbeleid

**Notities**

Om de in de vorige slide genoemde risico's af te dekken dient een netwerk wat met Internet is verbonden structureel te worden beveiligd. De kunst van het beveiligen komt neer op het inschatten van de risico's en het treffen van afdoende maatregelen. Als vuistregel wordt nogal eens gesteld dat het doorbreken van de beveiliging de inbreker meer moet kosten dan het oplevert. In ons oogpunt moet echter bij het beveiligen van het netwerk ook rekening worden gehouden met de kosten die nodig zijn om de schade van een inbraak weer te repareren. Die schade kan heel divers van aard zijn. Denk hierbij aan:

1. Reclamecampagne om het geschonden vertrouwen te herstellen
2. Nalopen van de bestanden op onterechte wijzigingen
3. Uitzoeken wat er met gestolen informatie is gebeurd
4. Kapitaal- of vermogensschade door het vrijkomen van vertrouwelijke informatie
5. Analyseren van de inbraak en treffen van nieuwe maatregelen

### 25.7.1 Houding

Een belangrijke faktor in de beveiliging van een object is de houding van de beheerder en de gebruikers ten opzichte van beveiliging. Iedere beveiliging kan namelijk worden ongedaan worden gemaakt door onachtzaamheid of opzettelijke verstoringen door de legitieme gebruikers. Net zoals het vrij ongebruikelijk is om sleutels van je voordeur

uit te delen moeten gebruikers ook voorzichtig omgaan met wachtwoorden en andere toegangsmiddelen tot het systeem. Naast de EDP-auditor (specialist op het gebied van informatiebeveiliging) en de beheerder hebben ook de gebruikers een belangrijke taak in het veilig maken en houden van de computersystemen en netwerken. Voorlichting speelt hier natuurlijk een belangrijke rol in.

### 25.7.2 Social Engineering

Een relatief succesvolle manier om achter wachtwoorden te komen is Social Engineering. Deze tactiek speelt in op de bereidwilligheid van nietsvermoedende medewerkers. De inbreker belt op en introduceert zichzelf als een lid van de systeembeheergroep die bezig is met het oplossen van een netwerkprobleem. Hiervoor heeft hij even de medewerking nodig van het slachtoffer. Kan die even de verbinding met het systeem verbreken en precies vertellen wat hij/zij intoetst bij het weer aanloggen? Keer op keer blijkt dat mensen dan zonder problemen hun gebruikerscode en wachtwoord doorgeven. Door gebruikers hierop te wijzen kunnen dergelijke missers worden voorkomen.

### 25.7.3 Kennis

Verder is het erg belangrijk dat de systeembeheerders die verantwoordelijk zijn voor de beveiliging hun kennis op dit gebied up-to-date houden. Het is al eerder in deze info shop gesteld dat een eenmaal aangebrachte beveiliging structureel moet worden onderhouden om zijn waarde te behouden. Om de haverklap verschijnen er nieuwe applicaties of worden beveiligingsproblemen opgespoord. Er zijn diverse mailing lists en USENET nieuwsgroepen aan beveiliging gewijd. CERT (Computer Emergency Response Team) stuurt regelmatig alerts rond als een nieuw beveiligingsgat of -probleem wordt gevonden. De World Wide Web site van CERT (http://www.cert.org) bevat adviezen en tips voor het beveiligen van aan Internet aangesloten netwerken.

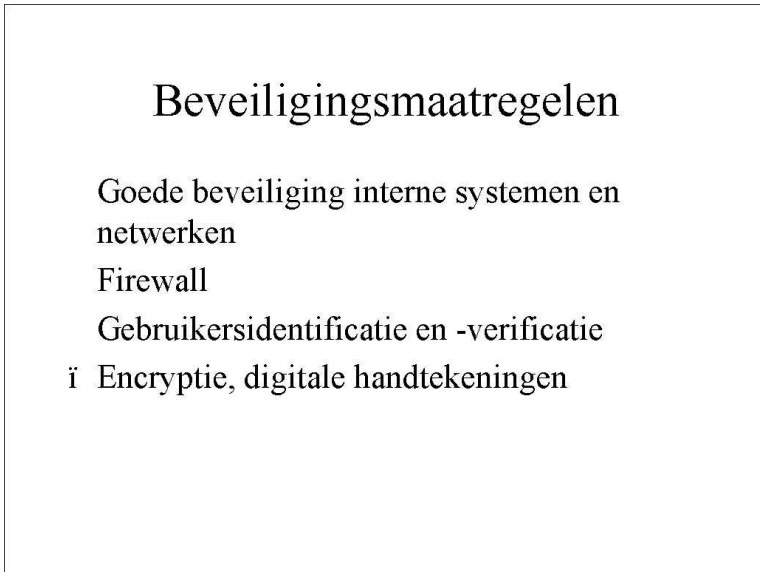
### 25.7.4 Beveiligingsbeleid

Het ijkpunt van de beveiligingsaktiviteiten van een organisatie moet worden gevormd door een expliciete Security Policy. Hierin moet het beleid ten aanzien van beveiliging uiteen worden gezet:

1. Wat mogen externe gebruikers? (van buiten naar binnen)
2. Wat mogen interne gebruikers? (van binnen naar buiten)
3. Wat is het standpunt ten aanzien van versleuteling
4. Hoe is de beveiligingsfunktie ingebed in de organisatie?
5. Welke controlemaatregelen moeten worden geïmplementeerd?
6. Hoe te reageren bij een inbraak?
7. Welke beveiligingsmaatregelen zijn wel/niet toegestaan?

Zonder een dergelijke security policy krijgt de beveiliging van een systeem of netwerk een willekeurig karakter. De Amerikaanse producent van firewalls Raptor (http://www.raptor.com) heeft een handboek uitgebracht waarin adviezen zijn opgenomen over het ontwikkelen van een security policy.

## 25.8 Beveiligingsmaatregelen




**Notities**

Er zijn gelukkig een groot aantal maatregelen te nemen waardoor een netwerk en de daarin opgestelde systemen afdoende zijn te beveiligen. Perfekte beveiliging bestaat echter niet. Het simpele feit dat systemen vanuit Internet toegankelijk zijn voor legitieme toepassingen brengt een bepaalde kwetsbaarheid met zich mee voor fouten en achterdeuren. Bij het implementeren van beveiligingsmaatregelen dienen we de balans tussen beveiliging, risico's en bruikbaarheid in het oog te houden.

In principe is het doel van een inbreker om zich via Internet toegang te verschaffen tot de computers van een organisatie. Eenmaal binnen kan de inbreker proberen een hogere systeemautoriteit te verkrijgen, bestanden te lezen of te wijzigen. Het compleet beveiligen van de toegang tot de systemen zou dus voldoende moeten zijn. Om een aantal redenen is dit helaas niet het geval. Het is met enkelvoudige maatregelen vrijwel onmogelijk de toegang tot een computersysteem in een Internet omgeving afdoende te beveiligen. Veel Internet protocollen bevatten bijvoorbeeld mogelijkheden om over het netwerk bepaalde commando's te geven, zonder dat de gebruiker die die commando's opstuurt zich hoeft te legitimeren. Verder is het onverstandig om de gehele beveiliging op één pijler te laten rusten. Zodra er in die pijler, bijvoorbeeld door vergissingen, onachtzaamheid of kwade wil, barsten ontstaan valt de gehele beveiliging in elkaar.

Een goede beveiliging bestaat uit een pakket aan maatregelen waardoor de inbreker meerdere muren dient te slechten alvorens hij binnen is. Zoals al eerder gesteld kan een eenmaal geïmplementeerde beveiliging ook niet permanent beschutting bieden tegen aanvallen van buitenaf. Naarmate het netwerk en de gebruikte applicaties veranderen dient de beveiliging mee te veranderen.

### 25.8.1 Firewall

Een zeer bekende Internet beveiligingsmaatregel is het plaatsen van een firewall tussen het eigen netwerk en Internet. Een firewall is een complex van apparatuur en programmatuur wat tot taak heeft ongeautoriseerd netwerkverkeer buiten de deur te houden. Meestal bestaat een firewall uit één of meer computers, een eigen netwerksegment en een stuk of wat routers . Voordat een firewall effektief kan worden ingericht dient precies bekend te zijn welke verbindingen zijn toegestaan en welke niet (security policy).

### 25.8.2 Gebruikersidentificatie en -verificatie

Een andere algemeen geaccepteerde maatregel is het invoeren van een identificatieverplichting voor gebruikers. Alvorens een gebruiker met een systeem of applicatie kan werken dient hij of zij zich aan het systeem bekend te maken. Om misbruik van het systeem te voorkomen moeten gebruikers na de identificatie meestal nog bewijzen dat ze zijn, wie ze zeggen dat ze zijn. Hiermee kan het systeem de identiteit van de gebruiker verifiëren. Het meest gebruikte systeem is dat van het wachtwoord. Hierbij “bewijst” de gebruiker zijn identiteit door het ingeven van een geheim woord dat (hopelijk) alleen de gebruiker en het systeem weten.

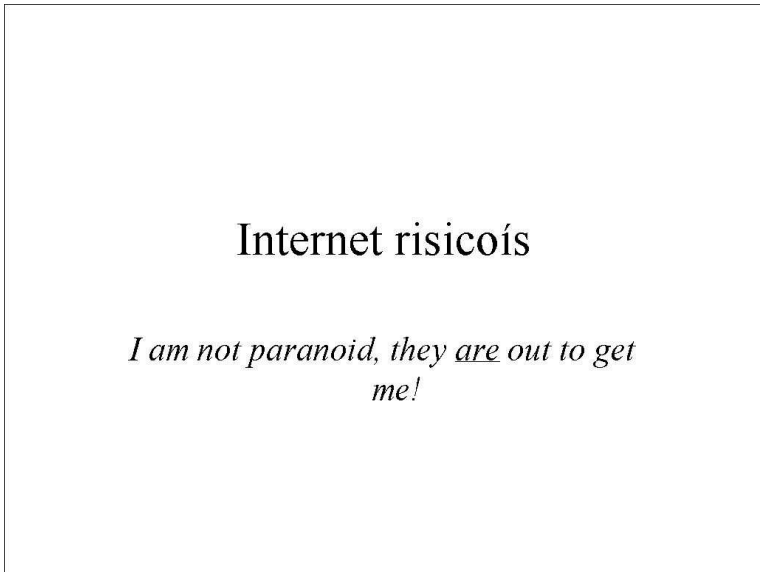
### 25.8.3 Encryptie en digitale handtekening

Een maatregel die tot op heden nog slechts (te) spaarzaam wordt toegepast is het versleutelen van gevoelige informatie. Een goed versleutelingssysteem zorgt ervoor dat alleen de geadresseerden de informatie kunnen lezen. Er zijn door de jaren heen behoorlijk goede encryptie programma’s ontwikkeld. Met de kracht van de hedendaagse computers kan een bestand of bericht zo goed worden versleuteld dat het met de snelste supercomputers niet meer leesbaar te maken is. Met aan encryptie verwante technieken kunnen berichten ook worden voorzien van een digitale handtekening. Hierdoor kan de ontvanger van een bericht er zeker van zijn dat het afkomstig is van de afzender.

# Hoofdstuk 26

# Internet risico’s

## 26.1 Internet risico's

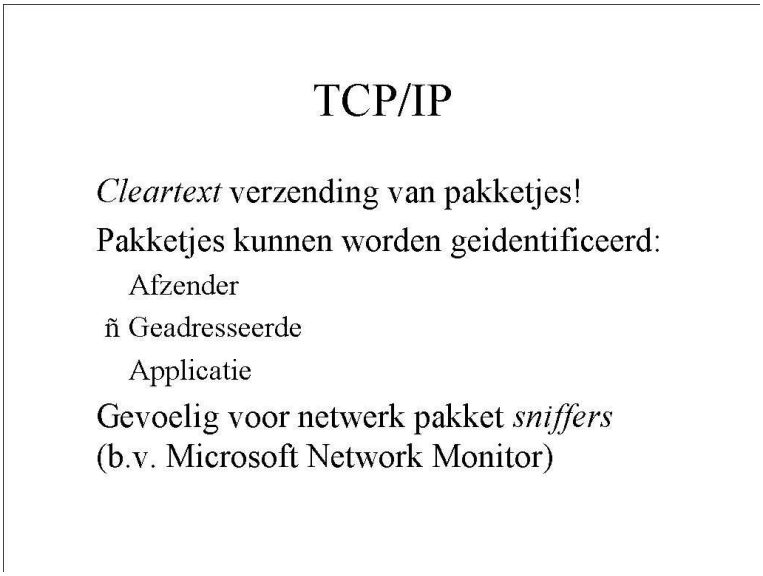



**Notities**

Een groot gedeelte van de onveiligheid van Internet wordt veroorzaakt door features in de gebruikte Internet applicaties. De meeste Internet applicaties zijn het resultaat van onderzoeksprojecten waarbij dynamiek en gemak voorop stonden. Beveiliging, zo het al een aandachtspunt was, had veelal een lage prioriteit.

In deze module worden de meest bekende risico's van de protocollen en applicaties die in Internet worden gebruikt behandeld.

## 26.2 TCP/IP




**Notities**

TCP/IP is de benaming van de verzameling datacommunicatieprotocollen die op Internet worden gebruikt. TCP/IP bestaat uit een aantal subprotocollen die ieder een bepaalde taak hebben in het versturen van informatie door het netwerk. Een diepgaande bespreking van de TCP/IP protocollen gaat buiten het bestek van dit document. Een belangrijk aantal beveiligingsproblemen op Internet ontstaat echter door de interne werking van TCP/IP. Om die reden bespreken we hier een aantal aspecten van TCP/IP. [1]

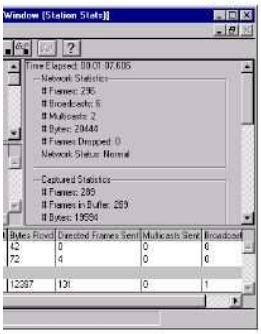
Berichten worden door TCP/IP opgedeeld in pakketjes (Engels: packets) met een maximale grootte (onder meer afhankelijk van de gebruikte netwerktechnologie). Deze pakketjes reizen onafhankelijk van elkaar door het netwerk. Zo kan het gebeuren dat één bericht (of bestand) wordt opgedeeld in meerdere pakketje die ieder via een andere route naar de eindbestemming reizen. Eenmaal daar aangekomen worden ze in de juiste volgorde gezet en aan de applicatie doorgegeven.

De TCP/IP protocollen ondersteunen geen encryptie. Ieder pakketje wat door het Internet reist bevat plaintext. Dit betekent dat als het pakket wordt afgeluisterd de volledige inhoud leesbaar is. De ruggegraat van Internet wordt gevormd door de netwerken van de access providers. De pakketjes van alle berichten die door Internet reizen komen vroeger of later door de systemen of routers van de providers heen. In principe kunnen de systeembeheerders van die providers alle Internet verkeer wat door hun netwerken reist afluisteren. Iemand die inbreekt op de systemen van de provider bezit die macht dus ook! Hoe onwaarschijnlijk dit ook lijkt, de in 1995 opgepakte hacker Kevin Mitnick had zich onder andere systeembeheerdersautoriteit verschaft op de systemen van

[1] Zie voor meer informatie over TCP/IP Internet Info Shop 2 over TCP/IP

The Well and Netcom, twee Amerikaanse providers.

## 26.3 Netwerk monitor 1

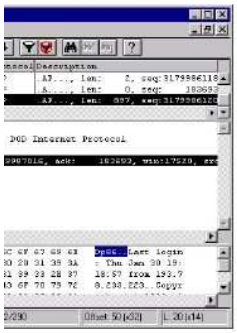



**Notities**

Programmatuur voor het afluisteren van netwerken is eenvoudig te verkrijgen. Vrijwel alle hedendaagse netwerkkaarten ondersteunen de zogenaamde *promiscuous mode*. In deze toestand haalt een netwerkkaart *alle* pakketjes van het netwerk (in tegenstelling tot de normale gang van zaken, waarbij netwerkkaarten alleen die pakketjes van het netwerk halen die expliciet voor hen zijn bedoeld).

Naast diverse public domain programma's levert Microsoft tegenwoordig met System Management Server (een Microsoft backoffice applicatie voor het beheer van PC netwerken) de *Network Monitor* mee. Met de Network Monitor kan *vrijwel iedereen* het netwerk monitoren en de onderschepte pakketjes inzichtelijk maken! Op de slide ziet u een voorbeeld van de Network Monitor.

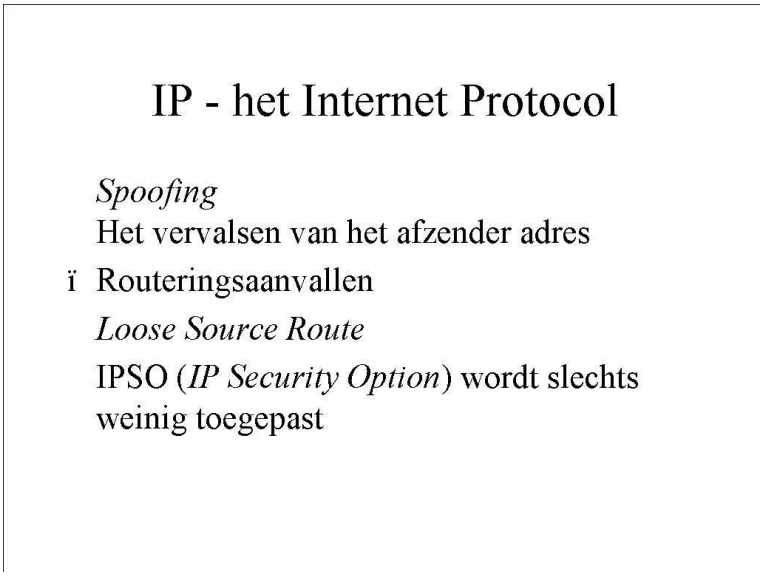
## 26.4 Netwerk monitor 2



**Notities**

Op de slide ziet u een voorbeeld van door de Network Monitor inzichtelijk gemaakt netwerk verkeer. De Network Monitor heeft vanzelfsprekend allerlei gebruikersvriendelijke features in zich voor het formatteren, filteren, doorzoeken, afdrukken en opslaan van de onderschepte netwerkpakketjes.

## 26.5 IP, het Internet Protocol




**Notities**

De kern van TCP/IP wordt gevormd door IP, het Internet Protocol. IP is verantwoordelijk voor het routeren van pakketjes door Internet en het afleveren van die pakketjes op de eindbestemming. IP heeft zelf geen idee voor welke applicaties de pakketje bedoeld zijn. Ook compenseert IP niet voor verminkingen in pakketjes, verloren gegane pakketjes of het in de verkeerde volgorde afleveren van pakketjes. Deze taken worden verricht door de hoger gelegen protocollen TCP en UDP.

IP is verzonnen met als achterliggende gedachte een zo dynamisch en flexibel netwerk mogelijk te maken. Hierdoor zitten er in IP nogal wat features die kunnen worden aangewend om beveiligingen van systemen te doorbreken. Een weinig gebruikt onderdeel van IP is de IP Security Option (IPSO). Hiermee kan worden voorkomen dat gevoelige informatie over onbeveiligde verbindingen wordt verstuurd. De grootste gebruiker van IPSO is het Amerikaanse leger. Er wordt wel onderzoek gedaan naar commerciële varianten van IPSO maar het zal nog wel even duren voordat die algemeen toepasbaar zijn. De meeste beveiligingsproblemen die in IP zitten kunnen worden opgelost met een degelijke firewall.

### 26.5.1 Spoofi ng

Om IP's taak, het doorsturen en afleveren van pakketjes, mogelijk te maken dient iedere computer op Internet uniek te zijn geïdentificeerd. Hiervoor is aan ieder netwerk interface een zogenaamd IP-adres toegekend. Zo'n IP-adres is een wereldwijd uniek getal in de reeks van 0 tot 4 miljard. Uit notatieoverwegingen worden IP- adressen geschreven als vier getallen van 0 tot en met 255, gescheiden door een "." , bijvoorbeeld

193.78.233.1. Omdat IP-adressen wereldwijd uniek moeten zijn, worden ze centraal uitgedeeld en geregistreerd. In principe kan je dus van een IP- adres vaststellen tot welke organisatie de computer aan wie het adres is uitgedeeld behoort.

IP stuurt in een pakketje zowel het IP-adres van de geadresseerde als dat van de afzender mee. Dit heeft ertoe geleid dat er een groot aantal applicaties zijn die beveiliging implementeren op basis van het IP-adres van de afzender van een bericht of commando. In zo'n applicatie dient in een configuratiebestand een lijst van IP-adressen te worden opgegeven. Als dan een verzoek, opdracht of bericht binnenkomt van een computer waarvan het IP-adres is gespecificeerd in de tabel wordt die door de applicatie geaccepteerd.

Deze beveiliging is helaas verre van waterdicht. Het is met aangepaste TCP/IP software mogelijk om IP-pakketjes te genereren met iedere gewenste afzender (IP-adres) erin. Als een inbreker weet dat een applicatie op die wijze is beveiligd, en hij weet (of gokt) een IP-adres wat in de autorisatielijst voorkomt, dan is die applicatie het haasje. De inbreker genereert dan commando's voor de applicatie en verstuurt die in IP-pakketjes met een valse afzender. De applicatie denkt dat het commando van een vertrouwd systeem afkomt en accepteert de opdracht. Let wel, de antwoorden van de applicatie komen natuurlijk nooit bij de inbreker aan. Die worden door de applicatie naar de computer gestuurd die bij het valse IP-adres hoort. Van de meeste applicaties hoef je echter niet de antwoorden te zien om een succesvolle aanval te plegen.

Het hierboven omschreven proces wordt in het Engels *spoofing* (neppen) genoemd. Het is technisch niet eenvoudig realiseerbaar, maar er zijn meerdere gevallen bekend waarbij over Internet in computers is ingebroken door toepassing van deze techniek. Met een goede firewall is spoofing voor een gedeelte uit te sluiten.

Een ander nadeel van autorisatie op basis van IP- adres is dat het adres geen indicatie geeft voor welke gebruiker er aktief is. Als meerdere gebruikers tegelijk op één computer werken krijgen alle Internet pakketjes die daarvandaan komen hetzelfde IP-adres als afzender.

### 26.5.2 Routering

Ieder systeem in een Internet omgeving heeft de TCP/IP protocollen aan boord. Iedere computer met meer dan twee netwerk interfaces is dan ook in staat om als router op te treden. Normaal gesproken worden Internet alleen maar *special purpose* routers gebruikt. Dit zijn computers die speciaal zijn ontworpen voor het routeren van pakketjes door een netwerk. De IP software die echter met PC's wordt meegeleverd kan ook worden gebruikt om pakketjes die niet voor die PC zijn bedoeld door te sturen. Dit heeft bijvoorbeeld als consequentie dat een PC met een uitbelverbinding op Internet als toegangspoort tot het interne netwerk kan worden gebruikt. Het seriële interface waaraan het modem is gekoppeld fungeert in dit geval als het tweede netwerk interface. Meestal kan deze automatische routering worden uitgezet. De standaardinstelling bij aflevering is echter vrijwel altijd om IP routing uit te voeren!

### 26.5.3 Routeringsprotocollen

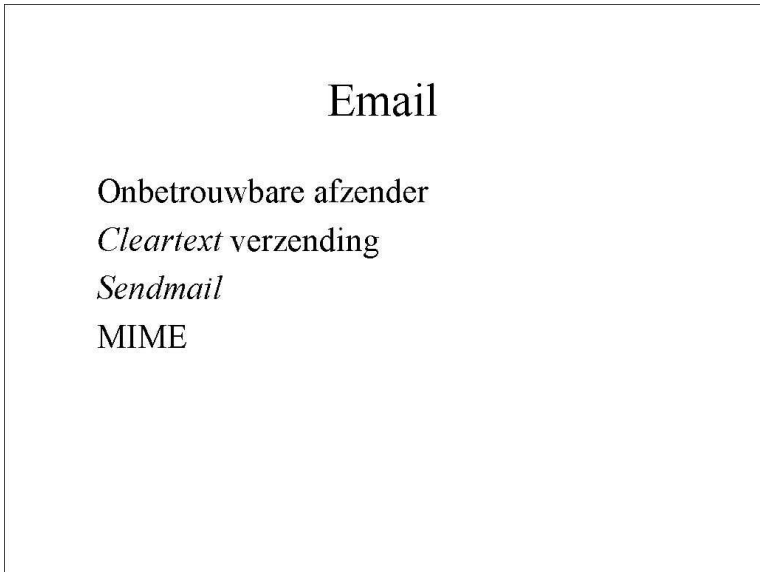
IP maakt gebruik van routeringstabellen om de beslissingen te nemen langs welke weg een pakketje door het netwerk reist. Vanwege het dynamische karakter van Internet worden die routeringstabellen meestal dynamisch bepaald met behulp van zogenaamde routeringsprotocollen. Routers die meedoen met zo'n routeringsprotocol zenden regelmatig pakketjes uit met daarin informatie over welke netwerken zij kunnen benaderen. De ontvangers van die pakketjes gebruiken die informatie om de optimale routes door Internet te bepalen. Als er een nieuw netwerk op Internet wordt aangesloten worden de routeringstabellen van de cruciale routers door dit mechanisme automatisch op de hoogte gesteld va de aanwezigheid van het nieuwe netwerk.

Een inbreker die geïnteresseerd is in het aftappen van informatie kan een poging doen de routeringstabellen van routers op afstand te wijzigen door het zenden van valse routeringsprotocolpakketjes. Hierdoor kan hij de routers laten denken dat de meest optimale route naar een bepaalde bestemming via de systemen van de inbreker loopt. Door het installeren van Sniffer programma's (zoals de Network Monitor) kunnen de pakketjes wederrechtelijk worden bekeken. In het ergste geval kan de inbreker de pakketjes zelfs wijzigen alvorens ze door te sturen! (via de uiterst handige *screening* faciliteit van een aantal UNIX implementaties)

### 26.5.4 Loose Source Route

Uit beveiligingsoogpunt worden nogal eens bepaalde gedeelten van een netwerk onbereikbaar gemaakt door het niet opnemen van routes naar dat deelnetwerk. Hierdoor kunnen inbrekers geen pakketjes naar servers in dat beveiligde netwerk sturen omdat er geen gedefinieerde route naar dat netwerk is. Helaas, er bestaat een standaardmethode om de routering in Internet te omzeilen. Dit kan door gebruik te maken van de *loose source route* optie. Deze IP-faciliteit maakt het mogelijk om in een IP-pakketje te specificeren welke route het pakketje door het netwerk moet volgen. Routers die de loose source route optie ondersteunen kijken niet meer in hun routeringstabellen maar versturen het pakketje via de route die in dat pakketje is opgenomen.

## 26.6 Email




**Notities**

Één van de belangrijkste Internet applicaties is electronische post. Op Internet wordt electronische post van systeem naar systeem verzonden met het Simple Mail Transport Protocol (SMTP). Van oorsprong was SMTP alleen in staat om 7 bits ASCII data te verzenden (alle letters en leestekens uit het Engelse alfabet). Recente uitbreidingen (waaronder ESMTP, Extended SMTP) staan ook toe om 8 bits karakters te verzenden. Hiermee kunnen ook diakrieten (bijvoorbeeld ë, ï, é) in een email bericht voorkomen.

### 26.6.1 Onbetrouwbare afzender

Het SMTP protocol voorziet niet in authenticatie van de afzender. Het is zeer eenvoudig mogelijk om een email bericht van iedere willekeurige afzender te voorzien. Via het telnet commando kunnen we rechtstreeks contact opnemen met de email server van een systeem (ligt achter poort 25). Door de juiste commando's van het SMTP protocol in te geven kunnen we zaken als afzender en aanverwante eenvoudig manipuleren. Dit betekent dat we niet kunnen vertrouwen op de afzender van een email bericht! Als we zeker willen zijn van de afzender van een bericht moeten we het voorzien van een digitale handtekening (zie ook hoofdstuk over encryptie).

In onderstaande voorbeeld wordt weergegeven hoe rechtstreeks contact is opgenomen met onze email server voor het ingeven van een bericht. De onderstreepte zinnen zijn door mij ingegeven.

```
$ telnet gatekeeper.osp.nl 25
Trying 193.78.233.1...
```

```
Connected to gatekeeper.osp.nl.
Escape character is '^]'.
220-gatekeeper.osp.nl OSP Sendmail 8.6.11/1.2 ready at Sat, 18 May 1996
12:55:44 +0200
220 ESMTP spoken here
helo noordpool
250 gatekeeper.osp.nl Hello gatekeeper.osp.nl
          [193.78.233.1], pleased to meet you
mail from: kerstman@noordpool
250 kerstman@noordpool... Sender ok
rcpt to: josv
250 josv... Recipient ok
data
354 Enter mail, end with "." on a line by itself

In verband met een depressie boven de noordelijke ijszee kan ik
vandaag helaas niet langskomen.

++De Kerstman
.
250 MAA23447 Message accepted for delivery
quit
221 gatekeeper.osp.nl closing connection
Connection closed by foreign host.
```

Het aldus verstuurde emailtje wordt door de ontvangende node in de inbox van de gebruiker josv geplaatst:

```
$ mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/josv": 1 message 1 new
>N  1 kerstman@noordpool    Sat May 18 12:56  12/457
& 1
Message 1:
From kerstman@noordpool Sat May 18 12:56:50 1996
Date: Sat, 18 May 1996 12:56:03 +0200
From: kerstman@noordpool

In verband met een depressie boven de noordelijke ijszee kan ik
vandaag helaas niet langskomen.

++De Kerstman

& q
```

### 26.6.2 Algemeen leesbaar

Internet email voorziet erin dat berichten niet rechtstreeks op de plaats van bestemming worden afgeleverd. Het is mogelijk om electronische postkantoren te bouwen

waar berichten worden opgeslagen voor verzending op een later tijdstip. Met name in complexere firewalls of bij conversie van Internet email naar een ander email systeem (zoals Microsoft Mail of Lotus ccMail) wordt er gebruik gemaakt van die electronische postkantoren. Uiteindelijk komt een mail bericht terecht in de electronische postbus van de ontvanger. Hier blijft het bericht staan totdat de gebruiker het met zijn email applicatie eruit haalt.

Vrijwel alle mail berichten worden onversleuteld verstuurd. Dit heeft tot gevolg dat de systeembeheerder van een systeem alle email die in de electronische postbussen op het systeem staat kan lezen en kan wijzigen. Op een systeem wat als electronisch postkantoor fungeert (bijvoorbeeld bij een provider) kan de systeembeheerder alle berichten die in het postkantoor aanwezig zijn lezen en wijzigen. Een inbreker die zichzelf systeembeheerdersautoriteit weet te verschaffen kan dat dus ook! Om de inhoud van een bericht volledig ontoegankelijk te maken voor systeembeheerders en inbrekers moeten we het bericht versleutelen (zie ook hoofdstuk over encryptie).

### 26.6.3 Sendmail

De meeste systemen op Internet maken gebruik van de applicatie Sendmail voor het ontvangen en versturen van email. Sendmail is door zijn ongelofelijke complexiteit een beveiligingsnachtmerrie! Er zijn veel verschillende Sendmail versies in omloop. Verder heeft Sendmail een lange geschiedenis van beveiligingsgerelateerde bugs. Sommige releases waren dermate gehandicapt dat algemeen wordt afgeraden om die releases te gebruiken! Sendmail draait veelal met een zeer hoge systeemautoriteit waardoor een inbreker die Sendmail naar zijn hand weet te zetten het systeem ook vrijwel onbeperkt kan manipuleren.

Er bestaan methoden om de meeste Sendmail beveiligingsproblemen in de tang te houden. Zo zijn er speciale front-end programma's die de SMTP berichtenstroom filteren en aan Sendmail doorgeven. Ook bestaan er simpelere en veiligere varianten van Sendmail. Gezien de enorme functionaliteit en flexibiliteit van Sendmail zullen een groot aantal organisaties toch nog van dit produkt gebruik moeten maken.

### 26.6.4 MIME

De Multipurpose Internet Mail Extensions (MIME) vormen een standaard voor het verzenden van allerlei binaire en multimediale informatie via Internet email. Met MIME is het mogelijk plaatjes, geluid, video en uitvoerbare programma's als een attachment mee te sturen met een email bericht. Een MIME-compatible mail programma herkent het type van de attachment en voert automatisch de statements uit die nodig zijn om het attachment op te slaan of weer te geven.

Het gevaar bij MIME zit hem in het automatisme waarmee het mail programma bepaalde acties uitvoert. Sommige MIME-types zorgen ervoor dat het mail programma automatisch via FTP een bestand ophaalt en op de lokale hard disk zet (met MIME berichttype message/external-body en access="anon-ftp") . Een inbreker kan op die wijze een MIME bericht aan de systeembeheerder sturen en automatisch een aangepaste versie van een configuratiebestand op laten halen. Ook kunnen via MIME Postscript bestanden worden opgehaald waarin gevaarlijke acties zitten. De MIME RFC

beschrijft deze en andere aan MIME gerelateerde beveiligingsaspecten.

## 26.7 Telnet

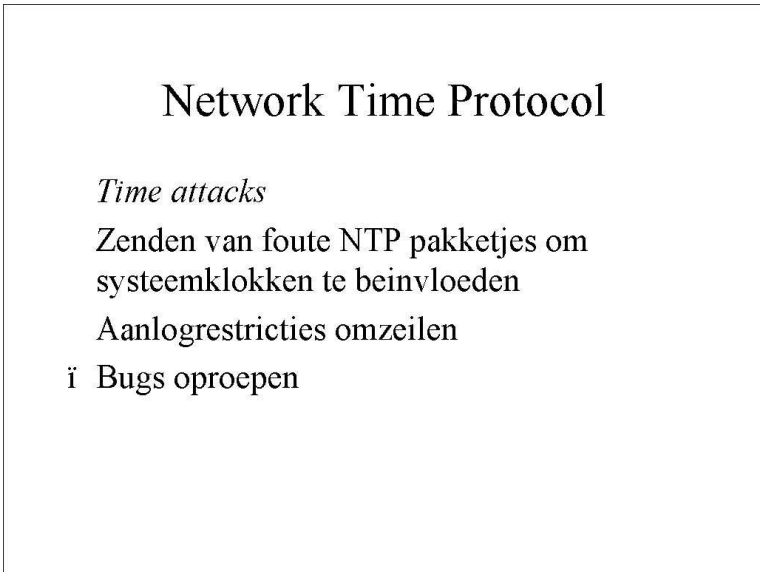

**Notities**

De telnet service staat het toe om op afstand aan te loggen aan een computer alsof je aan een lokale terminal zit. De meeste multi user systemen (UNIX, VAX/VMS, Windows/NT) ondersteunen een variant van telnet. Telnet authenticeert de login door de gebruiker zijn/haar gebruikerscode en wachtwoord te vragen. Eenmaal aangelogd fungeert telnet als een Internet terminal emulator: alle toetsaanslagen van de gebruiker worden doorgestuurd naar het systeem en de antwoorden van het systeem verschijnen op het scherm van de telnet gebruiker.

Telnet versleutelt de communicatie met het systeem niet. Een inbreker die in staat is het Internet verkeer af te luisteren ziet de gehele sessie (inclusief gebruikerscode en wachtwoord!) voorbij schieten. Dit afluisteren is niet zo exotisch als het lijkt. Met System Management Server (een Microsoft applicatie voor Windows/NT), wordt de Network Monitor meegeleverd. Dit is een interactief, gebruikersvriendelijk, programma voor het aftappen en zichtbaar maken van netwerkverkeer.

Er zijn een aantal telnet versies in omloop die de gehele sessie versleutelen zodat afluisteraars geen kans krijgen.

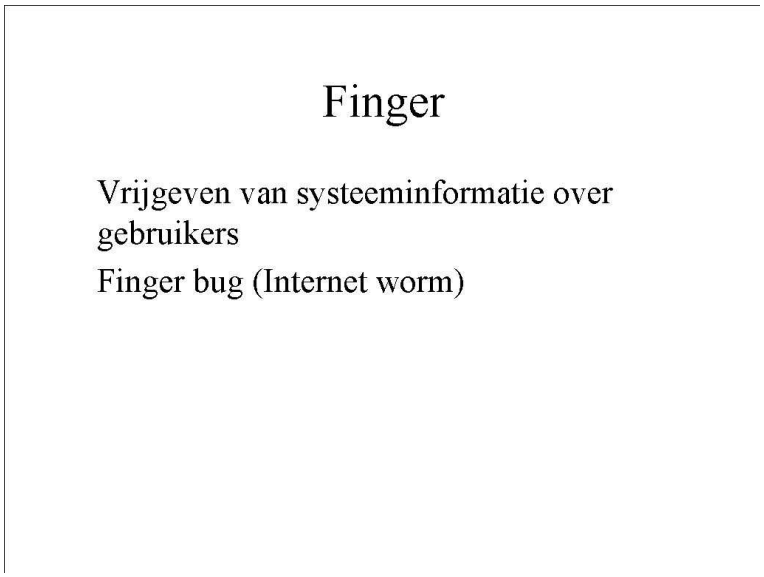
## 26.8 Network Time Protocol




**Notities**

Om de systeemklokken van systemen in een Internet omgeving gelijk te houden kan gebruik worden gemaakt van het Network Time Protocol (NTP). Met NTP synchroniseren systemen hun systeemklokken met een zogenaamde time source, een systeem wat de juiste tijd heeft. Organisaties die gebruik maken van NTP zijn gevoelig voor Time Attacks. Hierbij sturen inbrekers valse NTP pakketjes om op die manier de klok van een systeem te beïnvloeden. Hiermee kunnen beveiligingsmechanismen die zijn gebaseerd op tijd (bijvoorbeeld aanlogrestricties) worden omzeild.

## 26.9 Finger




**Notities**

Het finger commando kan worden gebruikt om op afstand gebruikersinformatie uit een systeem boven water te toveren. Finger geeft bijvoorbeeld informatie over wie er momenteel zijn aangelogd of detailinformatie over één gebruiker. Finger is op zichzelf niet echt gevaarlijk. Het is echter een uitstekend middel om informatie te krijgen over een systeem en zijn gebruikers. Hiermee geeft het systeem waardevolle informatie weg aan potentiële inbrekers die kunnen zien of gebruikers regelmatig aanloggen en wie er momenteel zijn aangelogd.

Finger wordt vaak rechtmatig gebruikt om te kijken wat de gebruikerscode van een gebruiker is zodat het email adres van die gebruiker kan worden bepaald. Om hieraan tegemoet te komen is het mogelijk een nieuwe finger service op een systeem te installeren die vertelt hoe email naar iemand kan worden gestuurd.

In onderstaande voorbeeld wordt het finger commando uitgevoerd om informatie over de gebruiker jvisser op te halen van twee verschillende Internet sites. De ene geeft de informatie gewoon vrij, de andere bevat onze eigen OSP finger service:

```
$ finger jvisser@simplex.nl
[simplex.nl]

Login: jvisser         Name: Jos Visser
Directory: /var/home/user/j/jvisser  Shell: /usr/local/bin/bash
Office: Gouda Home Phone: 01820-39088
Never logged in.
No Mail.
```

```
No Plan.
$
$ finger jvisser@osp.nl
[gatekeeper.osp.nl]

Welcome

To mail someone at the Open Solution Providers, use
Firstname.Lastname@osp.nl

eg: Jos.Visser@osp.nl

If you are not sure, please contact info@osp.nl or check our
web server at http://www.osp.nl

$
```

## 26.10 NIS, Network Information Service

NIS - Network Information Service

*Security bug* van de eerste orde!

Opvragen userid/password info via het netwerk

Bevat diverse zwakheden

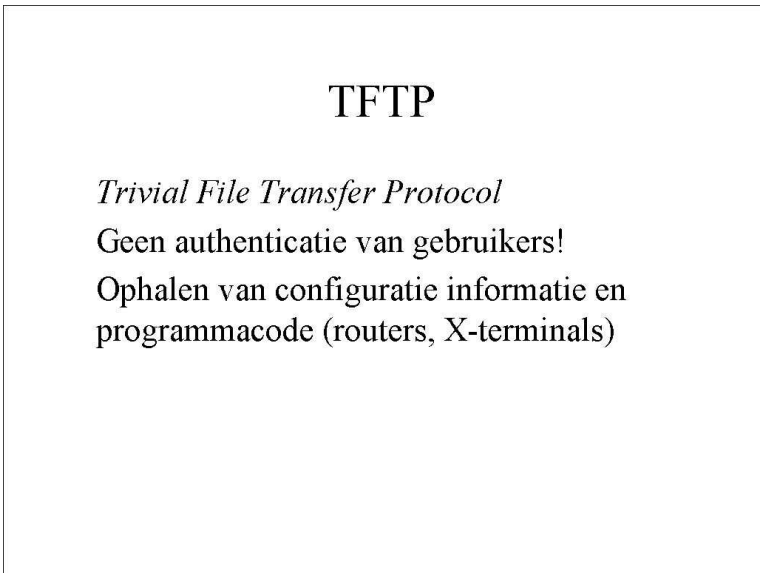
- V.b.: *ypghost*

**Notities**

De Network Information Service (NIS) is een onderdeel van het UNIX besturingssysteem waarmee belangrijke systeemconfiguratiebestanden (zoals de gebruikers- en groepeninformatie) centraal op een server in het netwerk kunnen worden opgeslagen. Het probleem hierbij is dat in principe iedereen via de juiste commando's die informatie over het netwerk kan ophalen. Hiermee krijgt een aanvaller waardevolle gegevens over gedefinieerde gebruikers, wachtwoorden en groepen. Sommige versies van NIS kunnen op afstand worden geconfigureerd om een andere NIS server te gebruiken. Hierdoor kan een inbreker een eigen NIS server opzetten en een NIS-client opdracht geven gebruiker/wachtwoord informatie van die valse server af te halen!

Een andere populaire inbraak methode is het algemeen verkrijgbare programma "ypghost". Dit programma luistert het netwerk af voor NIS requests en geeft zodra hij dit voorbij ziet komen zelf een vals antwoord.

## 26.11 TFTP, Trivial File Transport Protocol



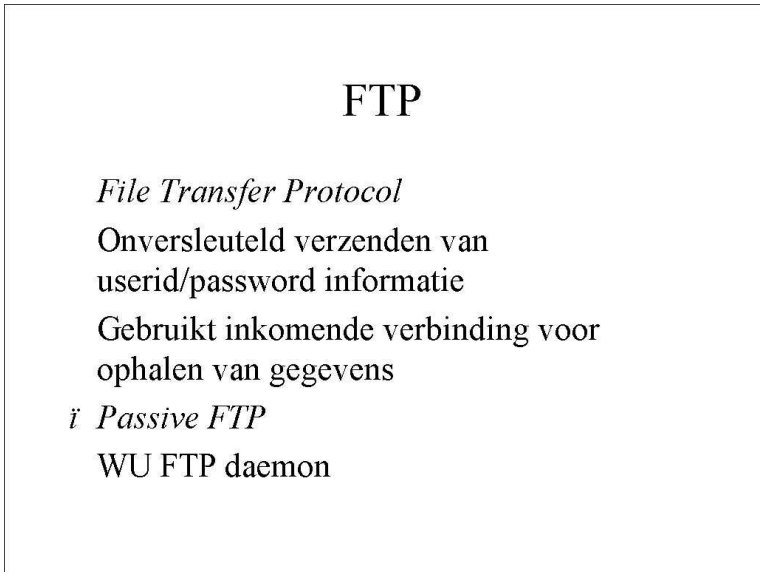

**Notities**

TFTP, het Trivial File Transport Protocol, is een simpele en onveilige variant van FTP. TFTP wordt vaak gebruikt door netwerkapparaten als netwerkprinters en netwerkterminals voor het ophalen van configuratiebestanden van centrale TFTP servers. TFTP vereist geen identificatie en verificatie van de gebruiker! In principe kan iedereen die contact op kan nemen met de server een TFTP sessie starten en files ophalen en opsturen!

```
$ tftp srv01.somewhere.com
tftp> get router01-conf
Received 3645 bytes in 1.2 seconds
tftp> quit
```

De meeste TFTP servers kunnen worden geconfigureerd om alleen het ophalen van bestanden uit bepaalde directories toe te laten. Ook dit is echter in veel gevallen al een beveiligingsrisico. De configuratiebestanden die doorgaans via TFTP worden verspreid bevatten vaak wachtwoorden en andere interessante configuratieinformatie. Via een packet filter kan toegang tot TFTP worden afgeschermd.

## 26.12 FTP, File Transfer Protocol




**Notities**

Het FTP protocol is het meest gebruikte Internet protocol voor het ophalen verzenden van bestanden. Er zijn verschrikkelijk veel verschillende FTP programma's in omloop. Ze gebruiken echter allemaal hetzelfde protocol zodat alle FTP programma's met alle FTP servers kunnen praten.

De beveiliging van FTP heeft net als bij telnet te leiden onder het feit dat gebruikerscodes, wachtwoorden en bestanden onversleuteld door Internet worden gestuurd. Inbrekers die sniffers hebben geïnstalleerd kunnen dus volop meegenieten van de informatie die langs komt zetten.

Via FTP worden veel documenten en programma's op Internet verspreid. Een inbreker die zich toegang heeft verschaft tot een populaire FTP server kan de inhoud van de bestanden die op die server staan aanpassen. Een populair programma wat op die FTP server staat zou door de inbreker wederrechtelijk kunnen worden aangepast om beveiligingsgaten in het systeem van de nietsvermoedende downloader te schieten.

Een ander probleem bij FTP is de meeste FTP- implementaties gebruik maken van een inkomende verbinding voor het ophalen van bestanden. Voor een FTP sessie worden dus twee verbindingen onderhouden: een control connection waarover de FTP commando's worden gestuurd en een data connection waarover de bestanden worden verzonden. Het probleem hier is dat de FTP server de data connection opzet! Een FTP client opent een random poortnummer (¿1024) op het client systeem en instrueert de FTP server om een verbinding te maken met de opgegeven poort. Dit heeft tot gevolg dat als we uitgaande FTP sessies toe willen staan (vanuit het interne netwerk naar Internet) we dus inkomende verbindingen moeten accepteren op random poortnummers voor de dataverbindingen!

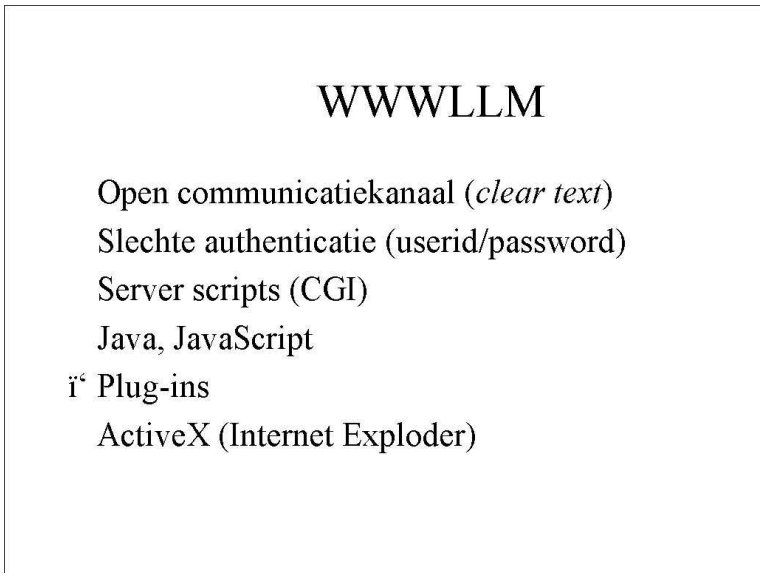
Een eventueel packet filter moet dus worden geprogrammeerd om inkomende verbindingen op poorten groter dan 1024 toe te staan. Dit kunnen immers de data verbindingen van FTP-sessies zijn! Echter, niet alle poorten boven de 1024 zijn veilig. Een groot aantal Internet services (NFS, X- Windows) gebruiken poorten boven de 1024!

Om FTP op een veilige wijze mogelijk te maken kunnen we gebruik maken van een circuit of application level gateway. Alle commerciële firewalls ondersteunen faciliteiten voor het veilig maken van FTP sessies zonder dat we inkomende verbindingen naar heel het netwerk toe hoeven te staan.

Een andere mogelijkheid is het implementeren van Passive FTP. Dit is een variant van FTP waarbij de client de data verbinding opent. Er hoeven dan dus geen inkomende verbindingen te worden geaccepteerd. Een probleem met passive FTP is echter dat niet alle FTP client en server programma's het PASV commando (wat nodig is voor passive FTP) ondersteunen.

De standaard FTP server applicatie die met de meeste besturingssystemen wordt meegeleverd ondersteunt slechts een beperkt aantal beveiligingsmogelijkheden. De public domain WU FTP Server (van de Washington University in St. Louis) bevat een ruime hoeveelheid mogelijkheden voor het inrichten van een veiligere FTP file server.

## 26.13 World Wide Web




**Notities**

Zelfs bij nieuwere Internet services zoals het World Wide Web is er maar in beperkte mate rekening gehouden met beveiligingsaspecten. Via het WWW worden documenten die op centrale servers staan opgehaald en afgebeeld. Naast documenten kan een WWW-browser ook bestanden ophalen via FTP.

### 26.13.1 Open communicatiekanaal

Informatie die via Internet wordt uitgewisseld tussen WWW-browsers en servers wordt normaal gesproken niet versleuteld. Net als bij vrijwel alle andere Internet informatiediensten heeft dit tot gevolg dat personen die in staat zijn datacommunicatieverbindingen in Internet af te luisteren de volledige inhoud aan zich voorbij zien trekken. In hele bijzondere gevallen zou een inbreker zelfs in staat kunnen zijn om de inhoud van een bericht te wijzigen terwijl het door Internet reist. Dit maakt het World Wide Web in principe ongeschikt voor het versturen van gevoelige informatie.

Recente uitbreidingen aan het World Wide Web maken het mogelijk om informatiepagina's en ingevulde formulieren te versleutelen. Met name Netscape en Microsoft ondersteunen de Secure Socket Layer (SSL). Dit is een extra laag software die de versleuteling van de verbinding voor zijn rekening neemt. [2]. Naast SSL ondersteunen sommige servers en browsers het S-HTTP (Secure HTTP) protocol. Net als SSL is ook dit een methode om de verbinding tussen WWW-servers en browsers te versleutelen.

[2] Zie voor meer informatie over SSL de documentatie van info shop 4 over het World Wide Web

### 26.13.2 Toegang

Normaal gesproken zijn WWW-servers algemeen toegankelijk. Dit betekent dat alle gebruikers alle documenten van die server af kunnen plukken. In een aantal gevallen moet het echter mogelijk zijn om een gedeelte van de WWW-server af te schermen. Alleen geselecteerde gebruikers zouden dan pagina's van die beschermde gedeelten af mogen halen. De meeste servers ondersteunen mogelijkheden om de toegang tot (gedeelten van) de pagina's af te schermen op basis van het systeem waarvandaan de opvraagaktie wordt uitgevoerd. Daar we op Internet echter de afzender van een pakketje niet mogen vertrouwen (IP-spoofing) is deze methode niet volledig betrouwbaar.

De meeste WWW-servers kunnen ook een beveiliging leggen op gebruikersniveau. Hierbij moeten gebruikers aanloggen aan de WWW-server alvorens ze document van de server kunnen ophalen. Eenmaal aangelogd kunnen de gebruikers alleen die documenten ophalen waarvoor ze expliciet zijn geautoriseerd. Het grote nadeel van deze zogenaamde *Basic Authentication* is dat de userid/passwords onversleuteld door het netwerk worden gestuurd. Voor een volledige beveiligde verbinding is dus nog steeds verstandig om bovenop gebruikersauthenticatie nog versleuteling toe te passen met SSL of S-HTTP.

Voor gevoelige informatie die alleen door de eigen werknemers mag worden geraadpleegd kan het beste een aparte WWW-server in het interne netwerk worden ingericht.

### 26.13.3 Server scripts

Één van de krachtige mogelijkheden van het World Wide Web is de mogelijkheid om de informatiepagina's dynamisch te laten genereren door zogenaamde CGI-programma's (CGI = Common Gateway Interface). Bij het ophalen van een document kan de WWW-server zien of het een hypertext document is of dat het opgevraagde document eigenlijk een programma is. Indien het laatste het geval is wordt het opgevraagde programma uitgevoerd en wordt de uitvoer van dat programma doorgegeven aan de WWW-browser. Ook na het invullen van een WWW-formulier worden de door de gebruiker ingevulde waarden doorgegeven aan een zo'n CGI-programma.

Deze faciliteit maakt het dus mogelijk dat ongeautoriseerde gebruikers programma's op de server opstarten en van parameters voorzien! Deze CGI-programma's worden doorgaans geschreven in krachtige geïnterpreteerde talen zoals Perl of de UNIX script taal. Indien deze programma's niet waterdicht in elkaar worden gestoken is het voor kwaadwillenden mogelijk om door het meegeven van foute (onverwachte) parameters allerlei neveneffecten te genereren. Aan het ontwikkelen van CGI-programma's moet dan ook veel zorg worden besteed.

### 26.13.4 Java en JavaScript

Tot op heden was het de taak van de WWW-browsers om documenten die de WWW-server opstuurde op te maken en weer te geven. Nieuwere mogelijkheden op Internet maken het echter mogelijk voor de WWW-servers om naast documenten ook programma's op te sturen en die uit te laten voeren door de WWW-browsers. Deze programma's zijn geschreven in de programmeertalen Java of JavaScript. Zodra een browser

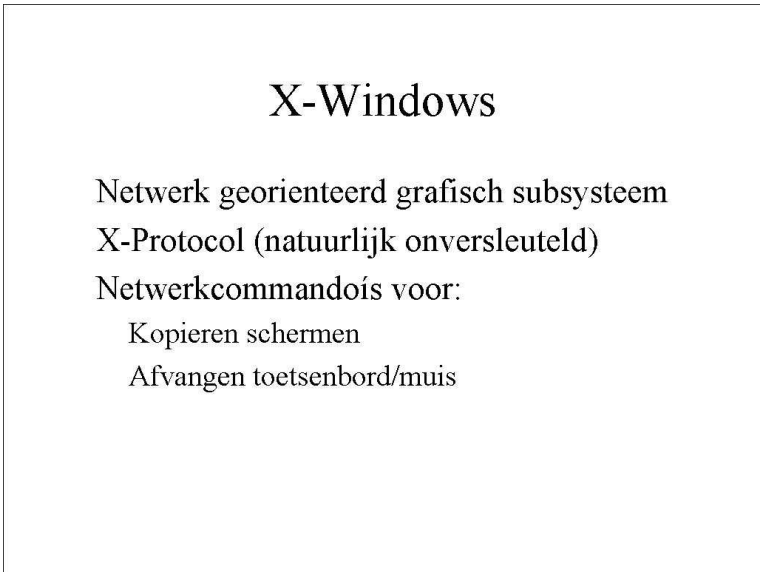
herkent dat er in een WWW-pagina een programma wordt aangeroepen haalt hij dat programma op en voert het uit.

Deze mogelijkheden geven natuurlijk ongekende nieuwe dimensies aan het begrip beveiliging! Een gebruiker die een WWW-pagina bekijkt krijgt immers volledig dynamisch een programma opgestuurd wat op zijn lokale computer wordt uitgevoerd! Een niet-latente angst voor virussen, Trojaanse paarden en andere ongewenste programma's is natuurlijk begrijpelijk. Zowel Java als JavaScript bevatten ingebouwde beveiligingsfeatures. Zoals het zich nu laat aanzien vormen deze ingebouwde beveiligingen voldoende bescherming tegen virussen en andere programmagerelateerde gevaren. Desalniettemin bevatten moderne browsers de mogelijkheid om het uitvoeren van Java programma's te verbieden.

### 26.13.5 Plug-ins

Een andere wijze om een WWW-browser uit te breiden is via een Plug-in. Een plug-in is een programma wat in de WWW-browser wordt geplugd om de functionaliteit van die browser uit te breiden met nieuwe documenttypen. Via plug-ins kan een browser worden "geleerd" hoe het allerlei niet-standaard documenten zoals spreadsheets, tekstverwerkingsdocumenten, Postscript documenten en 3D-werelden (Virtual Reality) moet weergeven. Het gebruik van deze plug-ins is niet zonder gevaar. Sommige documenttypen (Microsoft Word files, Postscript files) kunnen commandos bevatten die voor of tijdens het weergeven moeten worden uitgevoerd. Bij het weergeven van dat soort documenten door een plug-in zouden de ingebakken commando's automatisch kunnen worden uitgevoerd. De plug-ins zouden zo moeten worden ontworpen dat ze geen gevaar kunnen vormen voor de computer van de gebruiker.

## 26.14 X-Windows



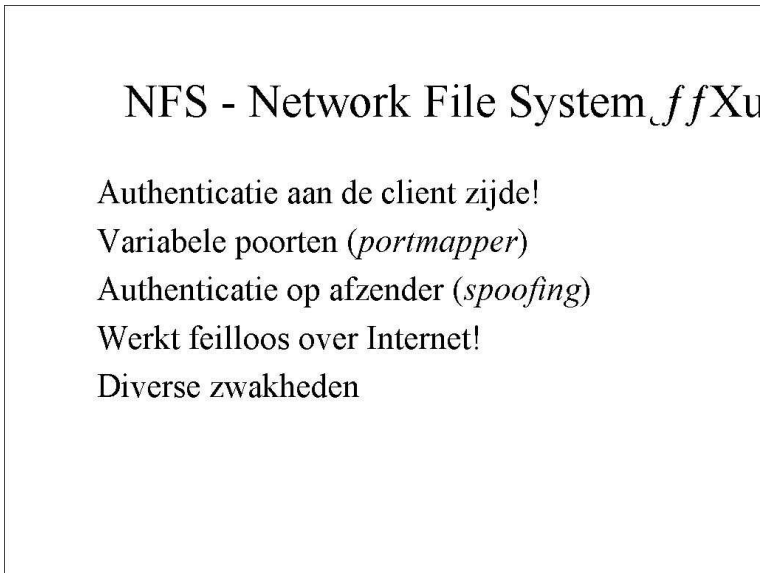

**Notities**

X-Windows is het grafische subsysteem van het UNIX besturingssysteem. Één van de kenmerken van X-Windows is dat de communicatie tussen de grafische terminal (de X-Server) en het grafische programma (de X-client) over het netwerk verloopt. Alle grafische terminals in een X-Windows omgeving zijn in principe servers die opdrachten die over het netwerk worden aangeleverd accepteren.

Een inbreker die weet dat in een organisatie X-Windows applicaties worden gebruikt kan proberen om via opdrachten uit het X-protocol de grafische terminals van de gebruikers over te nemen. De meeste X-terminals staan standaard zodanig ingesteld dat ze opdrachten van iedereen accepteren. In het X-protocol zitten een aantal zeer gevaarlijke mogelijkheden. Zo is het bijvoorbeeld mogelijk om toetsaanslagen af te vangen, toetsaanslagen te genereren en de inhoud van windows te kopiëren. Vrijwel al deze aktiviteiten kunnen worden ontplooid zonder dat de gebruiker hier iets van merkt! Verder is mogelijk om een X- Windows (grafische) login prompt te krijgen vanaf een UNIX systeem wat als X login server is geconfigureerd. Indien een inbreker de weg via telnet afgesloten vindt kan hij via deze manier vaak nog rechtstreeks aanloggen op het systeem.

Er zitten in X-Windows een aantal mogelijkheden om grafische terminals en login servers af te schermen. Daar wordt echter nog verbazingwekkend weinig gebruik van gemaakt. Het is dan ook aan te bevelen om X-Windows mogelijkheden in de firewall af te schermen met een packet filter. X-Windows verkeer verloopt meestal op poorten in de reeks 6000 tot 6100.

## 26.15 NFS, het Network File System




**Notities**

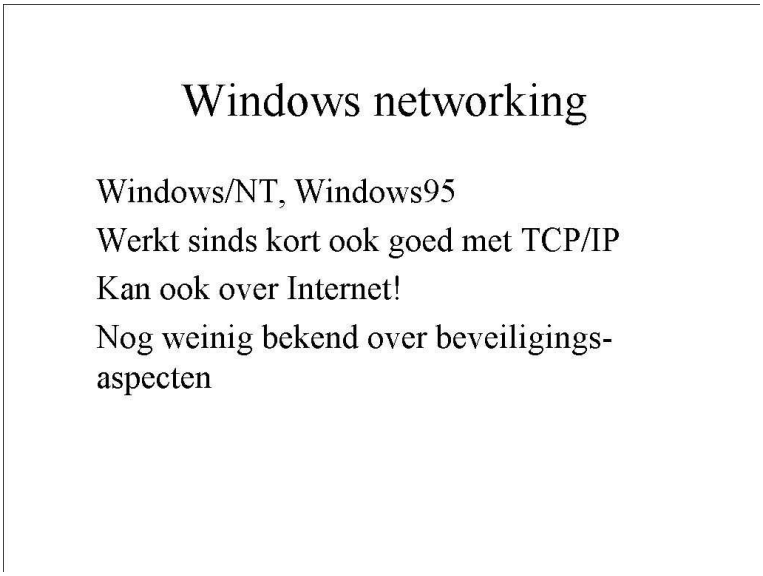
NFS is de netwerk file server applicatie van het UNIX besturingssysteem. NFS servers stellen een gedeelte van hun directory boom beschikbaar en NFS clients kunnen zich die geëxporteerde directories benaderen alsof het directories op de lokale hard disk betreft. De communicatie tussen de NFS client en de NFS server verloopt over het netwerk.

NFS gaat ervan uit dat de client en de server integraal worden beheerd. Een verzoek om een file van de server op te halen wordt door die server nauwelijks onderzocht omdat de server veronderstelt dat de gebruiker op de client computer al is geauthenticeerd. Verder is het bij NFS niet noodzakelijk dat de client computer officieel aanmeldt dat hij voornemens is gebruik te gaan maken van de server. Een NFS server zal normaal gesproken ieder verzoek (met de juiste parameters) honoreren.

Al deze eigenschappen zorgen ervoor dat NFS verkeer natuurlijk ten alle tijde door de firewall moet worden geblokkeerd. Een inbreker in Internet die weet dat een organisatie NFS gebruikt kan zich relatief eenvoudig toegang verschaffen tot de gexporteerde directories van de NFS servers van die organisatie. De Zweedse telefoonmaatschappij Telia heeft dat recentelijk aan den lijve kunnen ondervinden toen een hacker via NFS de World Wide Web home page van Telia veranderde.

De meeste NFS versies communiceren met de client op poort 2049. Verkeer op deze poort kan dus met een packet filter in de firewall worden tegengehouden. Er zijn echter ook versies die van random poortnummers gebruik maken (in combinatie met de portmapper).

## 26.16 Windows Networking




**Notities**

Onder de term Windows Networking scharen we hier alle file en print server faciliteiten van Microsoft operating systemen als Windows for Workgroups, Windows/95 en Windows/NT. Ook de UNIX pakketten LanManager/X en SAMBA die bedoeld zijn om te dienen als file en print server voor Windows PC's vallen onder deze noemer.

Recente uitbreidingen aan Windows Networking hebben het mogelijk gemaakt voor de Windows clients en servers om met elkaar te communiceren over TCP/IP. Met name met Windows/NT Server servers kunnen Wide Area Networks worden gebouwd waarin clients ook file servers buiten hun lokale netwerk kunnen benaderen. Ook Windows/95 en Windows for Workgroups machines kunnen echter als server in een WAN optreden.

Een Windows Networking client kan dus over een TCP/IP netwerk contact opnemen met een Windows Networking server om toegang te krijgen tot de gedeelde hulpbronnen van die server. Aangezien Internet een TCP/IP netwerk is kunnen die clients en servers dus ook over Internet met elkaar communiceren. Nu vereist Windows Networking dat een gebruiker zich eerst identificeert voordat hij de hulpbronnen van het systeem mag gebruiken. Windows for Workgroups ondersteunt echter geen gebruikersgeoriënteerde beveiliging van hulpbronnen. Windows/95 ondersteunt dit wel, maar alleen in combinatie met een Windows/NT server. Als een gebruiker zich niet kan identificeren kan hij/zij, afhankelijk van de instellingen, gasttoegang verkrijgen.

Een inbreker die weet dat een organisatie gebruik maakt van Windows Networking over TCP/IP kan stelselmatig proberen contact te maken met Windows/NT, Windows/95 of Windows for Workgroups servers die niet compleet zijn beveiligd. In vrijwel alle gevallen is Windows Networking over Internet natuurlijk ongewenst. In dat geval dienen de poortnummers die door Windows Networking worden gebruikt te worden geblokkeerd

(137, 138 en 139).

# Hoofdstuk 27

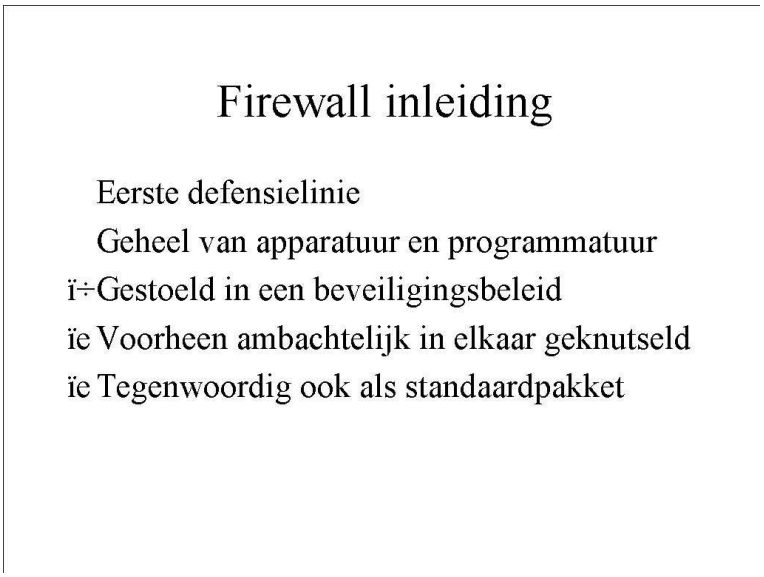
# Firewalls

## 27.1 Firewalls


**Notities**

Een firewall is een complex van apparatuur en programmatuur wat tot taak heeft ongeautoriseerd netwerkverkeer buiten de deur te houden. De traditionele plaats van een firewall is tussen Internet en het eigen netwerk. Grote organisaties hebben vaak ook interne firewalls die gedeeltes van het interne netwerk afschermen.

## 27.2 Firewall inleiding



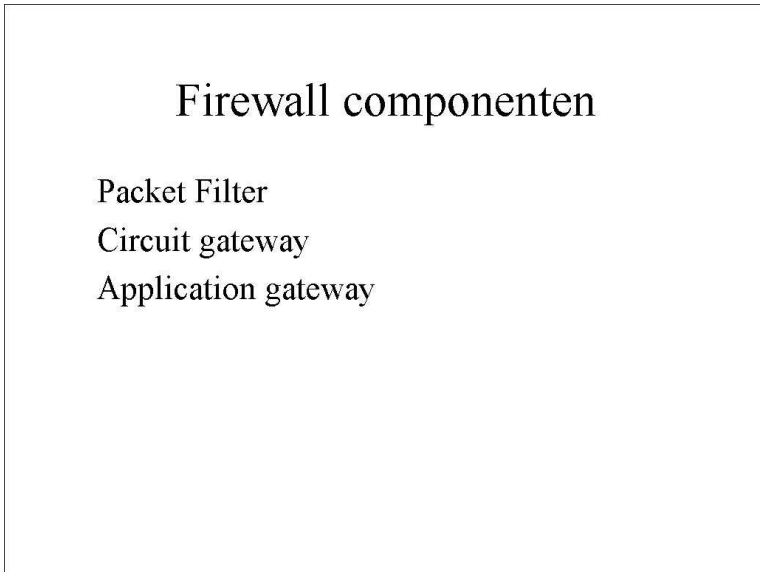

**Notities**

Een firewall fungeert als eerste linie in de beveiliging van het interne netwerk tegen aanvallen vanuit Internet. Het is echter onverstandig om alleen op de firewall te vertrouwen. De beveiliging van de interne system moet natuurlijk ook in orde zijn. Een firewall compenseert gedeeltelijk voor het gebrek aan beveiliging en beveiligingsmogelijkheden in de interne systemen en applicaties. Dientengevolge zijn firewalls vaak ook zeer specifiek ingericht voor het afschermen en beveiligen van vooraf gedefinieerde applicaties.

Het is lange tijd gebruikelijk geweest om firewalls op ambachtelijke manier in elkaar te knutselen op basis van zelf ontwikkelde en public domain programmatuur. Zo bevat de TIS firewall toolkit (ftp://ftp.tis.com/pub/firewalls/toolkit) een verscheidenheid aan programmaatjes waarmee een firewall kan worden gebouwd. Sinds enige tijd zijn er echter ook standaard firewall applicaties te koop die faciliteiten bevatten voor het afschermen van de bekendste applicaties. Er zitten natuurlijk grote voordelen aan het werken met standaardapplicaties. De configuratie van die firewall applicaties vereist echter ook veel inhoudelijke kennis. Verder zal ook zo'n standaard firewall moeten worden uitgebreid met aanvullende beveiligingsmaatregelen. Internet beveiliging blijft voorlopig het terrein van specialisten.

Een firewall dient natuurlijk zeer goed te worden beheerd. Als eerste verdedigingslinie tegen inbrekers is het natuurlijk van het grootste belang dat er in de beveiliging van de firewall geen gaten vallen. Om een indruk te krijgen van wat de vijand uitvoert dienen de logbestanden van de firewall nadrukkelijk te worden bestudeerd op mogelijke activiteiten van inbrekers.

## 27.3 Firewall componenten




**Notities**

Firewalls bestaan doorgaans uit één of meer van de volgende componenten:

1. Packet filter
2. Circuit gateway
3. Application gateway

### 27.3.1 Packet Filter

Packet filtering is een faciliteit van routers waarmee pakketjes op basis van protocol (TCP, UDP etc.), waar ze vandaan komen (IP-adres en poort) en waar ze naar toe gaan (IP-adres en poort) kunnen worden tegengehouden of doorgelaten. Hiermee kan dus een grove selectie worden gemaakt op basis van afzender of geadresseerde. Packet filters kunnen niet in de pakketjes te kijken om de inhoud ervan te beoordelen.

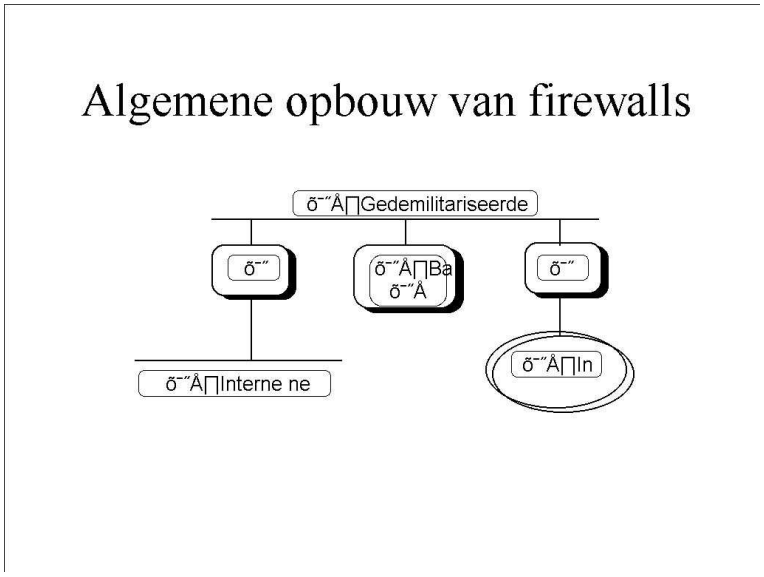
### 27.3.2 Circuit gateway

Circuit gateways zijn speciale applicaties die als doorgeefluik kunnen fungeren in verbindingen. Hierdoor hoeft er geen rechtstreekse verbinding tussen een interne en een externe applicatie te zijn. Beide partijen communiceren namelijk met de circuit gateway die de informatie transparant doorgeeft.

### 27.3.3 Application gateways

Applicatie gateways zijn programma's die net als circuit gateways bemiddelen tussen een interne en een externe applicatie. In tegenstelling tot circuit gateways hebben applicatie gateways echter verstand van de inhoud van de pakketjes. Hierdoor kunnen applicatie gateways netwerkverkeer ook inhoudelijk controleren.

## 27.4 Algemene opbouw van firewalls




**Notities**

De specifieke opbouw van een firewall verschilt van situatie tot situatie. Afhankelijk van de systemen en applicaties die moeten worden beschermd kan een firewall eenvoudiger of complexer van opbouw zijn.

In een firewall vinden we de volgende componenten:

- DMZ - Gedemilitariseerde Zone
  Voor de firewall wordt vaak een apart netwerksegment opgetrokken. Dit aparte segment wordt in de vakliteratuur aangeduid als de gedemilitariseerde zone (DMZ = DeMilitarized Zone).

- Ro - Router outside
  Deze router vormt de eerste verdediging tegen aanvallen van buitenaf. Door deze router te voorzien van een packet filter kan hier een eerste grove scheiding van het verkeer worden aangebracht. Meestal wordt deze router zodanig geconfigureerd dat aanvragen voor nieuwe verbindingen alleen naar de bastion host mogen worden gestuurd. Pakketjes van open verbindingen (dit zijn antwoorden van netwerkverbindingen die van binnen naar buiten zijn gemaakt) kunnen worden afgeleverd bij router Ri.

- Bastion Host
  De bastion host is een computer die voorzien is van de application en circuit gateway programmatuur van de firewall. Nieuwe verbindingen van buiten naar binnen worden typisch alleen toegestaan naar de bastion host. De gateway programmatuur op de bastion host kan dan beoordelen of de verbinding verder moet worden toegestaan. Door zijn positie in het netwerk is de bastion host ook de

aangewezen computer om te dienen als Web server, FTP server en email server voor de organisatie.

- Ri - Router inside
  De Ri router vormt de verbinding tussen het DMZ segment en het interne netwerk. Ook op deze router wordt doorgaans een packet filter geïmplementeerd wat alleen verkeer van de bastion host en de buitenste router Ro toestaat.

De integriteit van de componenten in de firewall dient natuurlijk zo goed mogelijk te worden gewaarborgd. Een firewall moet dan ook zeer zorgvuldig te worden ingericht en beheerd. Het is niet ongebruikelijk dat op een firewall geen beheer op afstand wordt toegestaan. Om aan te kunnen loggen op de firewall dient de beheerder dan fysiek achter de apparaten plaats te nemen!

## 27.5 Packet Filtering


**Notities**

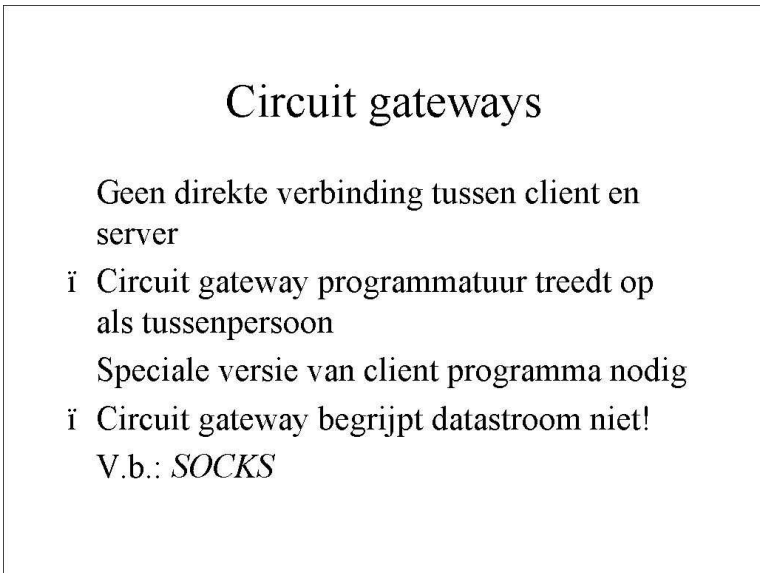
Packet filters maken onderdeel uit van vrijwel alle firewall implementaties. De mogelijkheid om pakketjes te filteren wordt met vrijwel alle routers meegeleverd. Bij packet filtering wordt aan een netwerkinterface van een router een toegangslijst (access list) gehangen. In zo'n access list worden toegangsregels beschreven waarmee een pakketje expliciet wordt doorgelaten of tegengehouden op basis van het protocol (TCP, UDP etc.), de afzender (IP-adres en poortnummer) en de geadresseerde (IP-adres en poortnummer).Het is gebruikelijk dat pakketjes die niet expliciet door de packet filter worden toegestaan worden weggegooid.

Zoals al eerder is genoemd is het op Internet mogelijk om IP-pakketjes te genereren met een valse afzender. Het doorlaten van verkeer op basis van de afzender is dus geen perfekte beveiliging. In combinatie met andere beveiligingsmaatregelen zoals wachtwoorden en application gateways is een packet filter echter zeer goed inzetbaar.

Met packet filters kan een gedeelte van de spoofing problematiek worden opgelost. We kunnen de packet filters namelijk zodanig configureren dat pakketjes die uit Internet arriveren met een afzender-adres (IP-adres) wat alleen in het interne netwerk voor mag komen door de filter wordt tegengehouden. Ook de loose source routing optie en valse routeringsprotocolpakketjes kunnen met packet filters worden onderschept.

Een packet filter vereist echter wel dat de poortnummers van de applicaties vooraf bekend zijn. Voor een groot aantal applicaties is dit zonder meer mogelijk. Er zijn echter ook applicaties waarvan het poortnummer random wordt gekozen op het moment van opstarten. De toegang tot deze applicaties kan dan ook niet met packet filters worden gereguleerd.

## 27.6 Circuit gateways



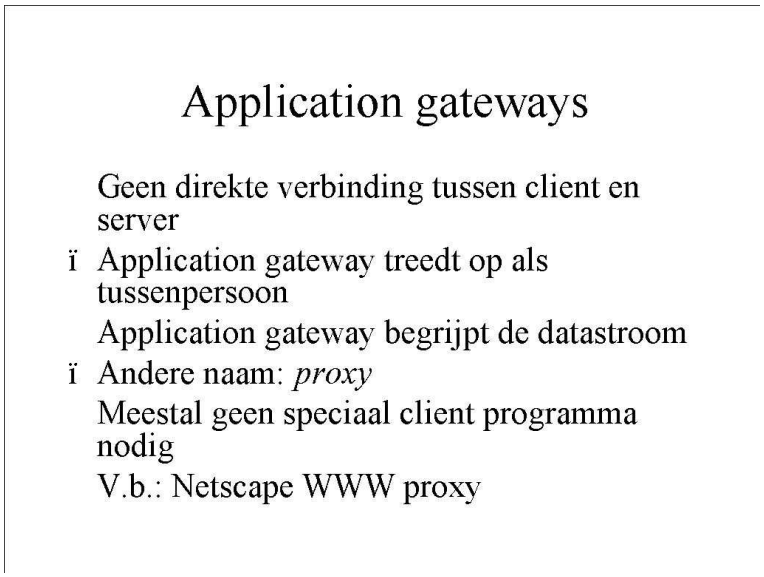

**Notities**

Circuit gateways kunnen worden gebruikt als we een applicatie in Internet gebruik willen laten maken van een server in het interne netwerk, zonder dat we een rechtstreekse verbinding van buiten naar binnen willen toestaan.

Neem als voorbeeld een gebruiker in Internet die om één of andere redenen iets af moet kunnen drukken op een netwerkprinter in het interne netwerk. Het is natuurlijk ongewenst om de toegang tot die netwerkprinter vanuit Internet open te zetten. In dat geval kunnen we in de firewall (op de bastion host) een circuit gateway installeren. De externe applicatie moet dan contact opnemen met de circuit gateway. De gateway neemt de verbinding op en creëert vervolgens een eigen verbinding met de netwerkprinter in kwestie. De externe applicatie denkt tegen de netwerkprinter aan te praten maar in werkelijkheid heeft hij contact opgenomen met de circuit gateway. De gateway geeft vervolgens de data in beide richtingen transparant door. Het beveiligingsvoordeel hier is dat er geen rechtstreekse verbinding tussen de externe en de interne applicatie nodig is. Circuit gateways hebben normaal gesproken geen verstand van de data die ze doorgeven.

Een nadeel van circuit gateways is dat de client applicatie moet worden aangepast om tegen de circuit gateway aan te praten in plaats van tegen de echte applicatie. In sommige gevallen betekent dit dat er een aparte versie van de client moet worden geïnstalleerd. De bekendste algemene circuit gateway is SOCKS.

## 27.7 Application gateways




**Notities**

Het concept van de applicatie gateway staat eigenlijk haaks op dat van de circuit gateway. In plaats van een algemene doorgeefluik faciliteit is een applicatie gateway specifiek ontwikkeld voor het afhandelen en doorgeven van pakketjes van één applicatie. Hierdoor kan een applicatie gateway in de pakketjes kijken en die beoordelen op hun inhoud. De mate van beveiliging die wordt geboden door applicatie gateways is hoog. Een probleem met deze gateways is echter dat ze specifiek moeten worden ontwikkeld. Voor de allernieuwste applicaties is dan ook vaak nog geen gateway beschikbaar.

Afhankelijk van de applicatie kan het noodzakelijk zijn dat de client applicatie specifiek is aangepast op het samenwerken met de (een) applicatie gateway. De populaire WWW-browser Netscape bezit bijvoorbeeld ingebouwde mogelijkheden voor het samenwerken met de bekendste circuit en applicatie gateways.

# Hoofdstuk 28

# Overige onderwerpen

## 28.1 Overige onderwerpen


**Notities**

In deze module worden een aantal overige onderwerpen, aan Internet beveiliging gerelateerd, behandeld.

Er wordt met name ingegaan op:

1. Authenticatie
2. Encryptie

## 28.2 Authenticatie


**Notities**

De meeste Internet applicaties kennen een mechanisme waarmee de functionaliteit van die applicaties selectief ter beschikking kan worden gesteld. De basis voor die mechanismen wordt gevormd door een verplichting voor de gebruiker om zich aan de applicatie bekend te maken. Het begrip gebruiker dient hier in brede zin te worden toegepast. Ook als twee applicaties gegevens uitwisselen kan het nodig zijn dat die applicaties zich aan elkaar bekend maken.

Inbrekers kunnen proberen deze beveiliging te omzeilen door zich valselijk voor te doen als een geautoriseerde gebruiker. Het authenticatiemechanisme dient dusdanig in elkaar te zitten dat het voor een inbreker onmogelijk is zich van een valse identiteit te voorzien.

### 28.2.1 Identificatie

De gebruiker zal zich eerst aan de applicatie bekend moeten maken. Meestal gebeurt dit met een gebruikerscode (userid). De eerste stap voor een inbreker om dit mechanisme te doorbreken is het bepalen van een geldige gebruikerscode. Om deze reden is het publiekelijk verspreiden van userids onverstandig. Manieren om een userid te achterhalen zijn:

1. raden,
2. het finger commando,
3. het SMTP vrfy commando (hiermee kunnen geldige email id's worden bepaald,

4. social engineering,
5. thrashing (het uitgrazen van oud-papier containers).

### 28.2.2 Authenticatie

Authenticatie is het proces waarmee de gebruiker zijn identiteit bewijst. Naast een gebruikerscode dient de gebruiker nog iets te hebben, iets te weten of iets te zijn. Hiermee kan die gebruiker bewijzen de rechtmatige gebruiker van de applicatie te zijn. Het doel van een inbreker is natuurlijk om het authenticatieproces beentje te lichten door een complete identiteit te stelen (gebruikerscode en bewijs) of door het gehele proces te omzeilen. Naarmate het te beschermen belang groter is zal het authenticatieproces completer en moeilijker te kraken dienen te zijn.

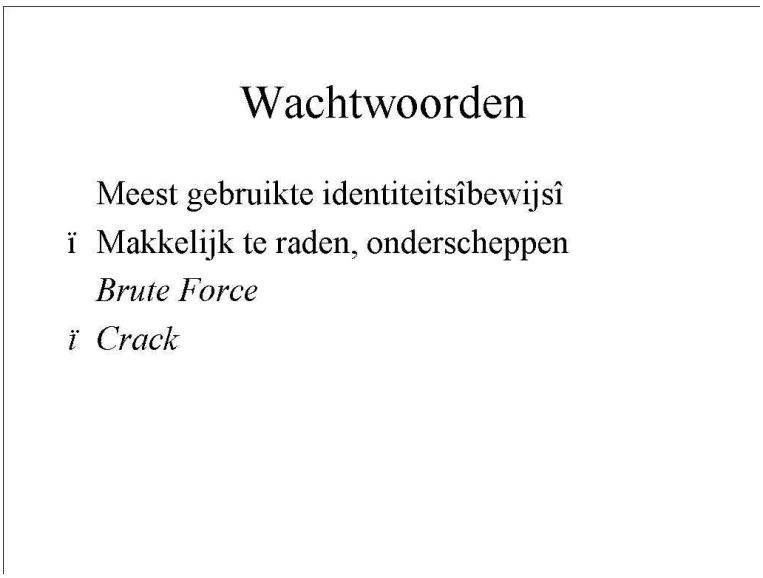
### 28.2.3 Autorisatie

Nadat de gebruiker zijn/haar identiteit heeft bewezen is het aan de applicatie om te bepalen wat die gebruiker wel en niet mag. Veel inbrekers zullen na het verkrijgen van toegang tot de applicatie proberen een hogere autorisatie te krijgen dan standaard aan het gekraakte userid is toegekend. Met name complexe en functioneel rijke besturingssystemen als UNIX en MVS bevatten nog wel eens gaten in de beveiliging waarmee dit mogelijk is.

### 28.2.4 Wederzijdse authenticatie

Het is recentelijk populair geworden om bij het aanloggen aan een server een wederzijdse authenticatie uit te voeren. Hierbij identificeert de gebruiker zich bij de applicatie en identificeert de applicatie zich bij de gebruiker. Met name bij het doorsturen van commerciële transacties waarin credit card nummers of andere gevoelige gegevens zijn opgenomen is het van belang voor de gebruiker om te weten of hij niet met een valse server communiceert.

## 28.3 Wachtwoorden



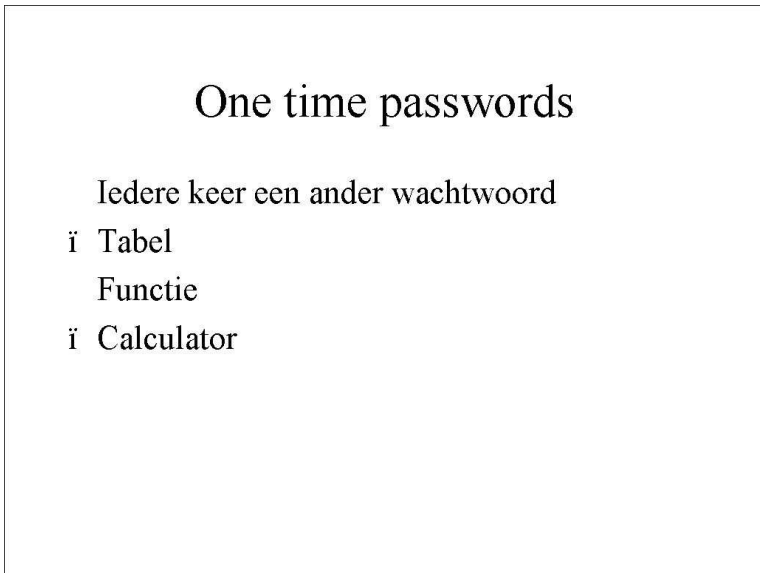

**Notities**

De meest populaire methode van authenticatie is het gebruik van wachtwoorden. Het gebruik van wachtwoorden is goedkoop, eenvoudig en gemakkelijk. Helaas is de mate van beveiliging die met wachtwoorden wordt geboden schrikbarend laag. Het is gebleken dat gebruikers:

1. Makkelijk te raden wachtwoorden kiezen.
2. Wachtwoorden doorgeven (per email!) of over de gang roepen.
3. Wachtwoorden opschrijven en aan de terminal plakken.

Het public domain programma Crack is naar verluidt in staat om gemiddeld 40wachtwoorden van de gebruikers van een systeem te raden. Hiervoor gebruikt Crack vuistregels en tabellen van veel gebruikte wachtwoorden (automerken, maanden, persoonsnamen). In een UNIX systeem is het gebruikelijk om de lijst van gebruikers en hun (versleutelde) wachtwoorden algemeen leesbaar op te slaan. Alhoewel de wachtwoorden niet terug kunnen worden gesleuteld is het met een Pentium PC en een woordenboek op disk mogelijk om een Dictionary Attack uit te voeren. Hierbij worden alle woorden uit de woordenlijst versleuteld en vergeleken met de versleutelde wachtwoorden uit de gebruikerslijst. Gebruikers die een wachtwoord hebben wat in een woordenboek voorkomt lopen hierdoor grote risicos. Ook het vervangen van o (letter o) door 0 (nul) en aanverwante voor de hand liggende mutatieregels zijn inmiddels ruimschoots bij inbrekers bekend.

## 28.4 One-time passwords

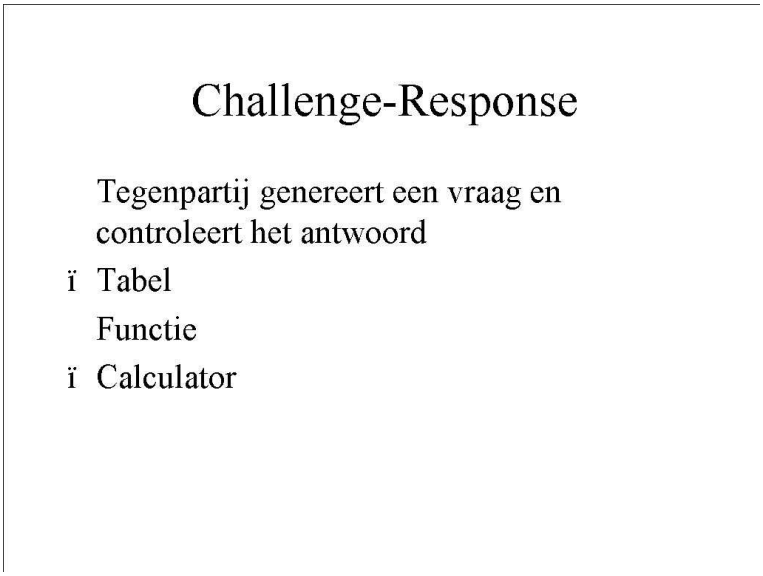



**Notities**

Een beter mechanisme is om te werken met wachtwoorden die slechts één keer kunnen worden gebruikt. Hierbij heeft het systeem een tabel of wiskundige functie die wordt gebruikt om het nu geldige wachtwoord uit te rekenen. Per keer dat de gebruiker aanlogt moet dus een ander wachtwoord worden ingegeven. Het voordeel van dit mechanisme is dat afluisteraars of meekijkers niets hebben aan het onderschepte wachtwoord. De volgende keer dat de gebruiker aanlogt moet immers een nieuw wachtwoord worden ingegeven.

Om dit voor de gebruiker toepasbaar te maken dient deze over een tabel of calculator te beschikken waarmee het wachtwoord kan worden bepaald. Hiermee is het authenticatiemechanisme dus gebaseerd op iets wat je hebt (een bezitsattribuut). Het S/KEY algoritme van Bellcore is een voorbeeld van een one-time password schema wat regelmatig wordt toegepast. Om misbruik van de lijst of calculator te voorkomen wordt er naast het one-time wachtwoord vaak ook nog een gewoon wachtwoord gebruikt.

## 28.5 Challenge-Response




**Notities**

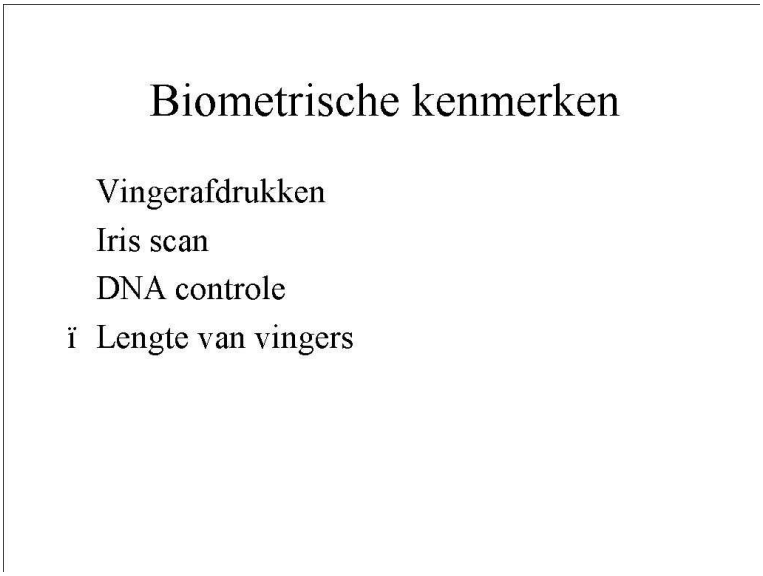
Bij een ander populair authenticatiemechanisme genereert de applicatie een vraag (challenge) waarop de gebruiker antwoord (response) moet geven. Meestal heeft de challenge de vorm van een getal wat door de gebruiker op een speciale calculator in moet worden gevoerd. Deze calculator berekent aan de hand van dat getal en een ingeprogrammeerde geheime sleutel een nieuw getal. Dit antwoord (de response) wordt vervolgens weer door de gebruiker aan de applicatie aangeboden. De applicatie kan dit antwoord controleren door de berekening na te doen met de challenge en de geheime sleutel waarvan bekend is dat die hoort bij de calculator van de gebruiker. Indien die berekening hetzelfde getal oplevert als de response van de gebruiker is vastgesteld dat de gebruiker de bezitter is van de goede calculator. Net als bij andere one-time password schema's wordt bij challenge-response meestal nog gebruik gemaakt van een ander wachtwoord om illegale toegang na diefstal van de calculator uit te sluiten. Deze combinatie van een kennis- en bezitskenmerk levert sterke authenticatie op.

Het nadeel van dit soort challenge-response oplossingen is dat de gebruiker de beschikking moet hebben over een calculator of chip kaart waarmee de berekening aan gebruikerszijde kan worden uitgevoerd. Deze apparaten zijn nog tamelijk prijzig en niet altijd eenvoudig mee te nemen.

In experimentele systemen wordt wel gebruik gemaakt van een challenge-response mechanisme waarbij geen extra hardware nodig is. De applicatie beschikt dan over een uitgebreide serie (persoonlijke) vragen waar hoogstwaarschijnlijk alleen de gebruiker het antwoord op weet. Voorbeelden van dit soort vragen zijn "Wat is je favoriete groente?" en "Hoe heette het hotel in Xalapa?". De applicatie kiest random een vraag uit de lijst, stelt die aan de gebruiker en vergelijkt het antwoord met het antwoord uit de lijst.

Gebruikers moeten in dit systeem de database met vragen regelmatig aanvullen. Het voordeel van dit schema is dat er geen extra apparaten nodig zijn om de response uit te rekenen. De veiligheid van dit systeem is echter gekoppeld aan de kwaliteit van de vragen in de database.

## 28.6 Biometrische kenmerken

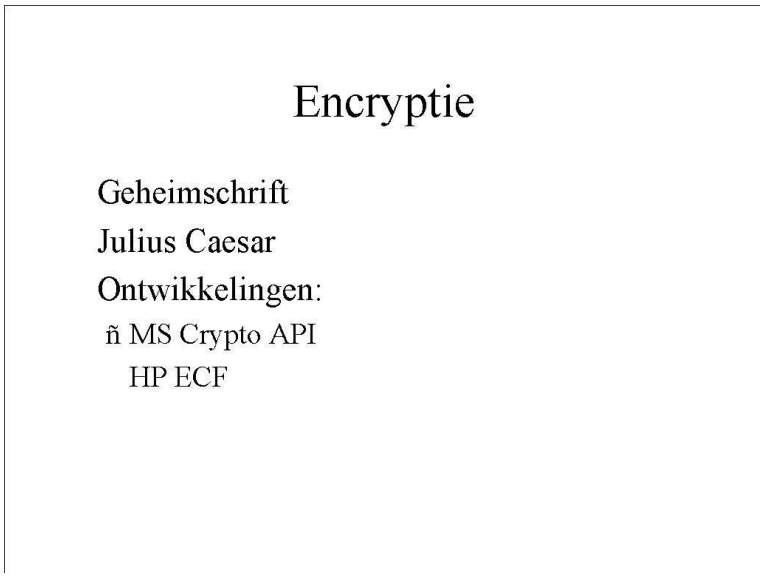



**Notities**

Er wordt al geruime tijd onderzoek gedaan naar het authenticeren van gebruikers op basis van meetbare persoonsgebonden biologische kenmerken als vingerafdrukken en stem. Tot op heden heeft dit onderzoek nog geen breed inzetbaar en veilig authenticatiemechanisme opgeleverd. In principe zou deze methode van authenticatie erg veilig zijn. Goede persoonlijke kenmerken zoals vingerafdrukken kunnen niet eenvoudig worden gestolen of nagemaakt.

Op dit moment is authenticatie door het meten vergelijken van biometrische kernmerken nog niet mogelijk.

## 28.7 Encryptie

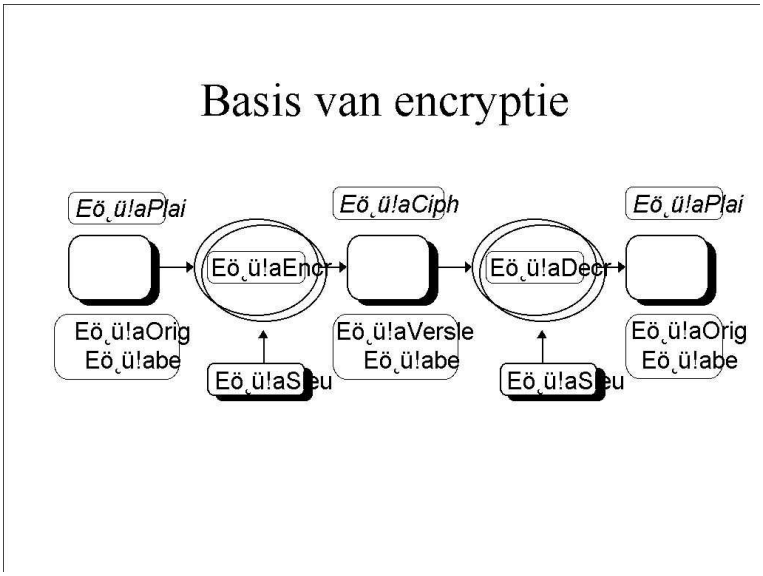



**Notities**

Om de exclusiviteit en integriteit van berichten die over Internet worden verstuurd te waarborgen kunnen we gebruik maken van encryptie (versleuteling). Steeds meer besturingssystemen en Internet applicaties ondersteunen encryptie. Rondom encryptie is een hele wetenschap, cryptographie, ontstaan die zich bezighoudt met het bedenken van nieuwe encryptiealgoritmes en het breken van bestaande algoritmes.

Het doel van encryptie is het zodanig veranderen van een bericht dat er bijzondere maatregelen nodig zijn om het weer leesbaar te maken. Idealiter hebben alleen de geadresseerden van het bericht de mogelijkheid om het bericht weer leesbaar te maken. Door een bericht te versleutelen wordt de inhoud ontoegankelijk voor afluisteraars die opzettelijk of toevallig het bericht onder ogen krijgen. Encryptie is al zo oud als de mensheid zelf. Naar verluidt paste Julius Caesar encryptie toe om de communicatie tussen het front en Rome te beveiligen. Er bestaan momenteel een groot aantal encryptiealgoritmes die met meer of minder succes kunnen worden ingezet om berichten te (de)coderen.

## 28.8 Basis van encryptie




**Notities**

In het plaatje op de slide wordt het proces van encryptie schematisch weergegeven. Het originele bericht (de plaintext) wordt door het encryptieprogramma versleuteld en omgevormd tot het versleutelde bericht (de ciphertext). Deze ciphertext kan vervolgens door Internet heen worden gestuurd naar de ontvanger. Deze voert het versleutelde bericht aan het decryptieprogramma die het originele bericht weer boven water weet te toveren. Idealiter mag de ciphertext door iedereen worden onderschept en bekeken. Alleen de legitieme ontvanger heeft de beschikking over de mogelijkheden om het bericht terug te sleutelen.

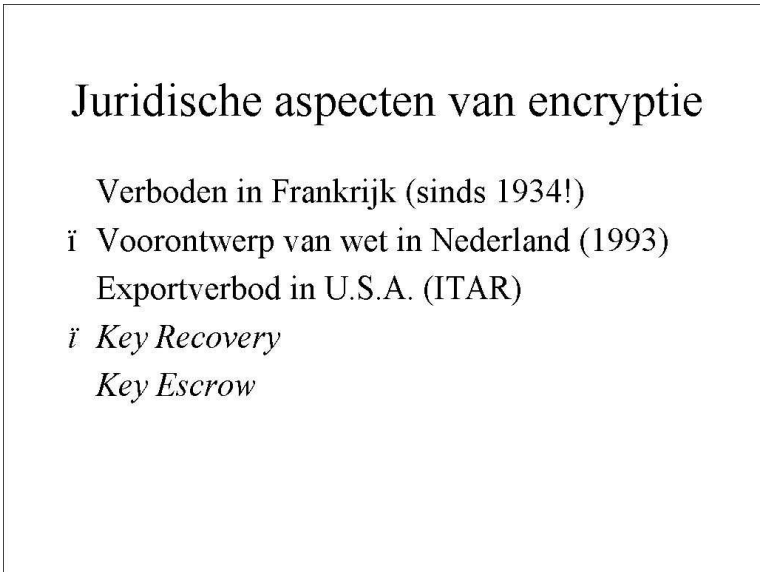
Het is natuurlijk verreweg het handigst indien de encryptie- en decryptiefaciliteit zijn geïntegreerd in de applicatie die de berichten genereert en verwerkt (bijvoorbeeld het email programma of de WWW-browser). Er zijn echter ook losse encryptieprogramma's in omloop waarmee berichten buiten de applicatie om kunnen worden ont- en versleuteld. Het bekendste (en beste) publiek verkrijgbare encryptieprogramma op dit moment is PGP (Pretty Good Privacy). Meestal wordt voor de encryptie en de decryptie hetzelfde programma gebruikt (met eventueel andere opties).

De encryptie en decryptie van berichten wordt beïnvloed door twee sleutels die samen met het bericht in het encryptieprogramma moeten worden gevoerd. Het encryptieprogramma voert een wiskundige bewerking uit op het bericht en de sleutel. Bij een goed encryptieprogramma is het gebruikte algoritme bekend. Hierdoor hangt de beveiliging van het bericht af van de kwaliteit van het algoritme en de sleutel. Van een geheim algoritme kan de kwaliteit niet worden onderzocht en een dergelijk algoritme kan dus gaten bevatten die het breken van de code eenvoudiger maken.

Het doel van de inbreker is om de versleuteling te breken. Hierbij wordt gepoogd om

de ciphertext weer leesbaar te maken zonder dat men op voorhand de beschikking heeft over de juiste sleutel. Een algoritme wat relatief eenvoudig valt te breken noemen we een zwak algoritme. Sterke algoritmes zijn daarentegen moeilijk te breken of zelfs onbreekbaar (naar de huidige stand van de techniek).

## 28.9 Juridische aspecten van encryptie




**Notities**

Alvorens dieper in te gaan op de toepassing van encryptie op Internet is het zinnig eerst enkele woorden te wijden aan de juridische aspecten van encryptie. De hedendaagse encryptiealgoritmes en computers maken het mogelijk een bericht dermate sterk te versleutelen dat het vrijwel onmogelijk is om het bericht, zonder de beschikking over de sleutel, weer leesbaar te maken.

Er zijn echter voor de overheid legitieme redenen om een versleuteld bericht weer leesbaar te willen maken. Gecodeerde telefoongesprekken tussen criminelen en versleutelde boekhoudingen van criminele aktiviteiten zijn hier voorbeelden van. Om deze redenen gaan er met de regelmaat van de klok stemmen op om encryptie aan banden te leggen. Een vorm van regulering die recentelijk onder de aandacht is gekomen is Key Escrow. Hierbij zijn partijen verplicht om de sleutels die ze gebruiken te deponeren bij een centraal depot. De overheid zou dan met toestemming van de rechter een sleutel uit het depot kunnen halen om een bericht terug te sleutelen.

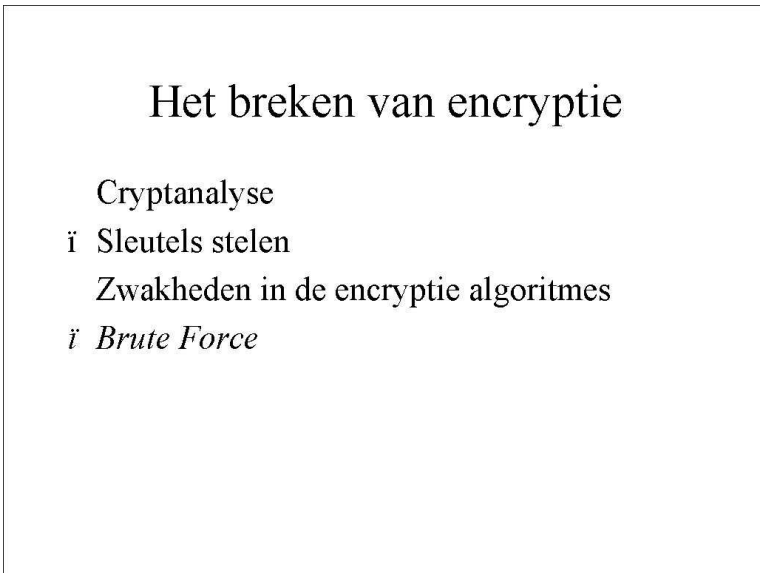
De Amerikaanse overheid heeft naar analogie hiervan een encryptie chip ontwikkeld, de Clipper chip, waarbij de sleutels die nodig zijn voor de decryptie automatisch (tijdens de fabricage van de chip) bij een centrale instantie worden belegd. De Clipper chip wordt vrijelijk (tegen lage kosten) beschikbaar gemaakt voor industriële en bedrijfstoepassingen.

Naar mijn inzicht zal deze methode onwerkbaar zijn en niet het gestelde effect bereiken. Criminelen zijn al crimineel en het is mijns inziens onzinnig om te veronderstellen dat een crimineel zijn sleutels bij het centrale depot zal beleggen. Vervolgens hoef je in Nederland niet aan je eigen veroordeling mee te werken. Het vooraf verplicht deponeren van decryptiesleutels valt wellicht onder het verplicht meewerken aan een

mogelijke veroordeling. Ook zijn er encryptiealgoritmes bekend die als ciphertext een gedicht opleveren. Van de buitenkant is dan niet te zien dat het hier een versleuteld bericht betreft!

In de Verenigde Staten valt encryptieprogrammatuur onder de wapenwet (de ITAR, in de categorie munitie). Hierdoor moeten leveranciers die applicaties willen exporteren waarin encryptiemogelijkheden zijn opgenomen een exportvergunning aanvragen. De geldende praktijk is dat alleen zwakke algoritmes toestemming krijgen voor export. Hierdoor kunnen Amerikaanse software producenten als Microsoft, Lotus en Netscape geen sterke encryptiemogelijkheden in hun software bieden. Een Franse onderzoeker toonde dit korte tijd geleden aan door binnen enkele weken een bericht wat met Netscape was versleuteld te breken. De houding van de Amerikaanse overheid in deze bevordert de beveiliging van Internet dataverkeer niet!

## 28.10 Het breken van encryptie




**Notities**

De veiligheid die door encryptie wordt geboden is natuurlijk geheel afhankelijk van de moeite die moet worden gedaan om het gekozen algoritme te breken. Bij het breken van een versleuteld bericht proberen we de inhoud van het bericht te achterhalen zonder dat we op voorhand de sleutels kennen.

### 28.10.1 Sleutels stelen

Verreweg de meest effectieve methode om een bericht te ontsleutelen is door het stelen van de sleutel die nodig is om het bericht leesbaar te maken. In veel gevallen bewaren gebruikers sleutels in bestanden op hun computer. Een inbreker die zich op andere wijze toegang heeft verschaft kan de sleutels uit het bestand kopiren zonder dat de gebruiker weet dat zijn sleutels niet langer geheim zijn. Het PGP programma versleutelt om die reden het bestand met sleutels met een extra sleutel (de pass phrase). Ook het onderscheppen van sleutels tijdens transport (bijvoorbeeld per email) behoort tot de mogelijkheden.

### 28.10.2 Zwakheden in het algoritme

Sommige algoritmes bevatten inherente zwakheden die het mogelijk maken een bericht geheel of gedeeltelijk leesbaar te maken. Een eenvoudig voorbeeld van zo'n zwakheid is een algoritme die de statistische samenhang van een bericht bewaart. Van de Nederlandse taal is per letter bekend hoe vaak die gemiddeld in een bericht voorkomt. Ook van combinaties van letters is frequentie waarmee die voorkomen uitgezocht. Oudere

encryptiealgoritmes (op het niveau: Donald Duck) bewaarden deze statische samenhang in de ciphertext. Hierdoor kon met een statistische analyse van de ciphertext de originele inhoud van het bericht worden geraden. Moderne cryptanalysis technieken zijn natuurlijk vele malen geavanceerder maar proberen ook dergelijke zwakheden in een algoritme uit te buiten.

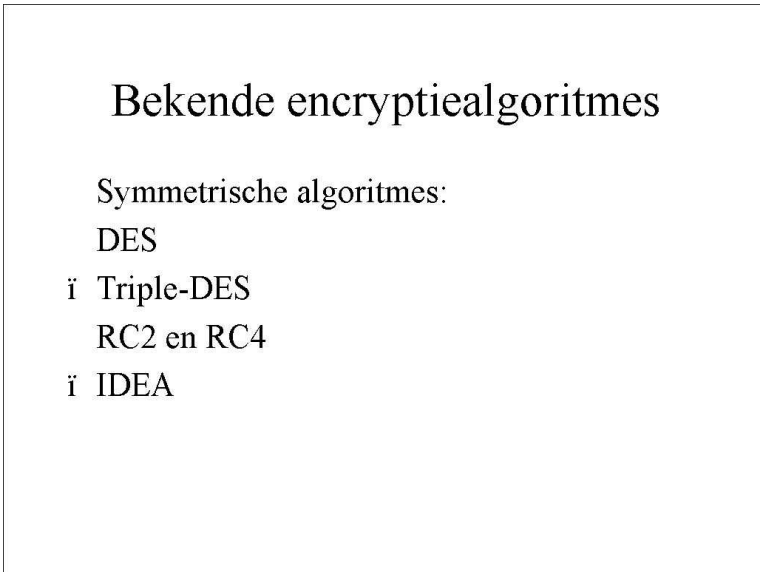
### 28.10.3 Brute force

Een zekere manier om binnen een vastgestelde tijd de originele inhoud van een versleuteld bericht te achterhalen is het domweg proberen van alle mogelijke sleutels. Een nadeel van een dergelijke brute kracht aanval is dat het afhankelijk van de lengte van de sleutel is hoe lang het duurt voordat het bericht in zijn originele staat is teruggebracht. Voor een algoritme waar nog geen zwakheden van bekend zijn is brute force echter de enige mogelijke aanvalstechniek. Door een sleutel van voldoende lengte te kiezen kunnen we de tijd die nodig is voor een dergelijke aanval tot ongekende hoogte opvoeren.

Met een sleutel van 16 karakters (128 bits) en een computer die honderd miljoen sleutels per seconde kan proberen zijn we gemiddeld 53.951.415.354.000.000.000.000 jaar bezig voor we het originele bericht hebben achterhaald. Dezelfde computer doet er echter maar 33 dagen over om alle mogelijke sleutels van 6 karakters (48 bits) te proberen!

De Amerikaanse overheid geeft naar verluidt geen exportvergunningen af voor encryptiealgoritmes die met sleutels werken die langer zijn dan 40 bits.

## 28.11 Bekende encryptiealgoritmes




**Notities**

In de loop der jaren zijn er een grote hoeveelheid encryptiealgoritmes verzonnen en weer afgezonken (in verband met zwakheden). Wij zijn voorstander van het gebruiken van een algemeen bekend encryptiealgoritme waarvan de kwaliteit vast staat en aan wetenschappelijk onderzoek onderhevig is.

De meeste encryptiealgoritmes zijn symmetric private key algoritmes. Dit betekent dat voor de encryptie en decryptie dezelfde geheime sleutel wordt gebruikt. Het grote probleem met deze klasse van algoritmes is het distribueren van de sleutels. Alvorens zo'n algoritme kan worden gebruikt dienen zender en ontvanger beiden dezelfde geheime sleutel te kennen. Deze sleutel moet dus op een veilige wijze kunnen worden uitgewisseld. Door het inzetten van een leger van koeriers die ieder met een gedeelte van de sleutels op pad gaan kan dit probleem worden opgelost.

### 28.11.1 DES

Het DES (Data Encryption Standard) algoritme vormde jarenlang de encryptiestandaard van de Amerikaanse overheid en het internationale zakenleven. Het algoritme gebruikt een sleutel van 56 bits (7 karakters) en is dientengevolge vandaag aan de dag gevoelig voor brute force aanvallen. Michael Wiener van Bell Northern Research toonde aan dat voor tien miljoen dollar een computer kan worden gebouwd die ieder bericht wat met DES is versleuteld in 21 minuten weer leesbaar kan maken.

### 28.11.2 Triple-DES

In een poging om het leven van DES nog ietwat te rekken is Triple-DES ontwikkeld. Hierbij wordt het DES algoritme drie keer uitgevoerd met twee verschillende sleutels. Het resulterende algoritme kan worden bezien als een versie van DES met een effectieve sleutel van 112 bits.

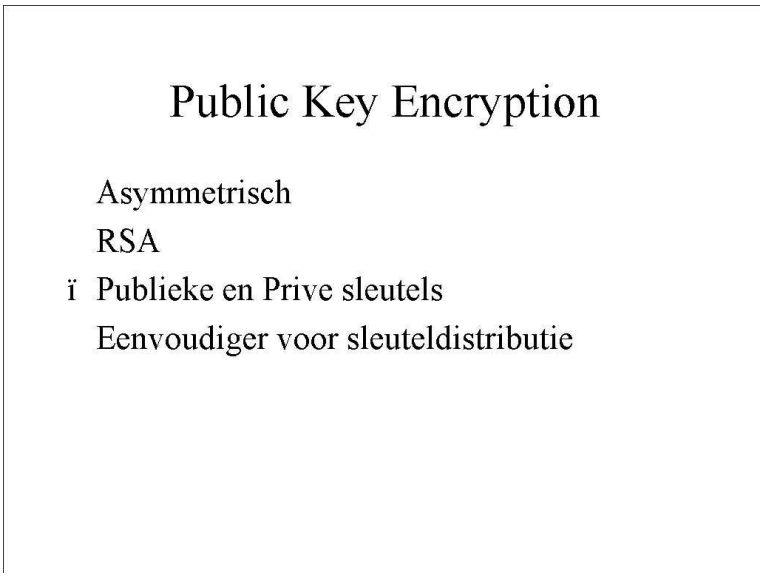
### 28.11.3 RC2 en RC4

Deze encryptie algoritmes (uitgevonden door professor Ronald Rivest) worden momenteel tamelijk algemeen toegepast in Internet. De algoritmes werken met een variabele sleutellengte van 1 tot 1024(!) bits. Indien sleutels korter dan 48 bits (6 karakters) worden gebruikt zijn de algoritmes relatief eenvoudig te breken. Doordat de algoritmes intellectueel eigendom zijn van RSA Data Security Inc. weet niemand precies hoe veilig ze zijn bij langere sleutels. Netscape gebruikt (onder andere) RC4 met een sleutellengte van 40 bits als SSL encryptieprotocol in de export versie van de Netscape Navigator.

### 28.11.4 IDEA

Een nieuwe ster aan het encryptiefirmament is het International Data Encryption Algorithm van James Massey en Xuejia Lai. In tegenstelling tot de hierboven ontwikkelde algoritmes betreft het hier een Europees (Zwitsers) produkt. Het encryptieprogramma PGP gebruikt onder andere IDEA voor het versleutelen van berichten. Voor zover nu bekend bevat IDEA geen zwakheden. Het algoritme werkt met sleutels van 128 bits (16 karakters).

## 28.12 Public Key Encryption




**Notities**

Een bijzondere vorm van encryptie wordt gevormd door de Public Key Encryption (PKE) methodes. Bij PKE is er sprake van een sleutelpaar bestaande uit twee verschillende sleutels. Een bericht wat met de ene sleutel wordt versleuteld kan alleen met de bijbehorende andere sleutel weer leesbaar worden gemaakt. Één van de twee sleutels, de zogenaamde public key, wordt publiekelijk bekend gemaakt. De andere sleutel, de private key, wordt ten strengste geheim gehouden.

Mensen die mij een bericht willen versturen versleutelen dit bericht met mijn public key. Eenmaal versleuteld kan het rustig over Internet worden verzonden want alleen ik kan het bericht weer leesbaar maken met mijn private key. Merk op dat zelfs de originele verzender een eenmaal versleuteld bericht niet meer kan terugsleutelen!

Het bekendste PKE algoritme is dat van Rivest, Shamir en Adleman: het RSA-algoritme. RSA encryptie wordt breed toegepast door onder andere Netscape, Lotus Notes en PGP. Het bedrijf RSA Data Security Inc. heeft momenteel het alleenrecht op de verkoop en toepassing van public key encryption en het RSA-algoritme in de Verenigde Staten.

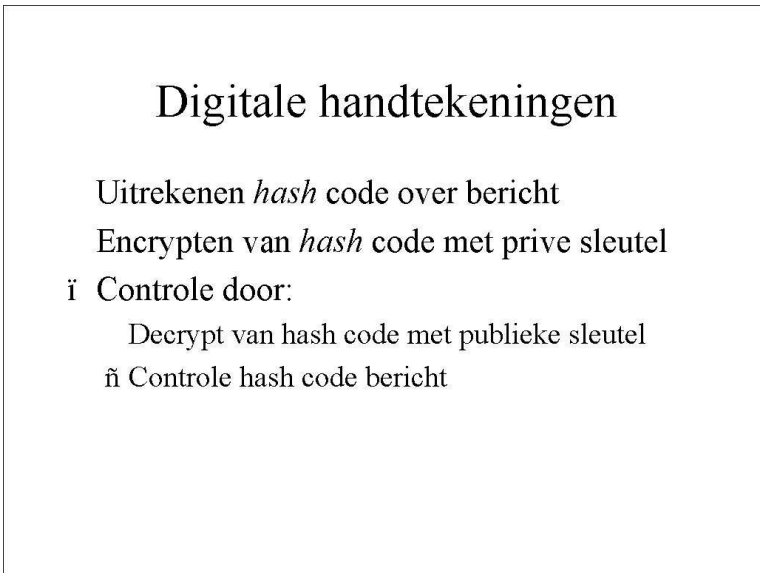
Het grote voordeel van PKE is dat er vrijwel geen sleuteldistributieproblemen zijn. De sleutel die nodig is om een bericht te versleutelen, de publieke sleutel, kan algemeen bekend worden gemaakt. Publieke sleutels kunnen vrijelijk worden gepubliceerd en gekopieerd zonder dat de inhoud van de berichten in gevaar komt.

De beveiliging van een PKE versleuteld bericht hangt volledig af van de geheimhouding van de priv sleutel. Als die eenmaal op een of andere manier bekend is geworden kan een inbreker alle berichten die voor het slachtoffer bestemd zijn weer leesbaar maken. Daar PKE sleutels tamelijk lang kunnen worden (PGP kan met sleutels van

1024 bits werken!) worden ze meestal opgeslagen in bestanden. Om te voorkomen dat die bestanden worden gestolen dienen die weer apart te worden beveiligd door ze te versleutelen met een private key algoritme als IDEA.

Een probleem wat zich met PKE kan voordoen is dat de versleutelaar van een bericht er wel zeker van moet zijn dat de gebruikte publieke sleutel daadwerkelijk bij de geadresseerde hoort. Een inbreker die het dataverkeer van een slachtoffer wil afluisteren zou kunnen proberen valse publieke sleutels van het slachtoffer in omloop te brengen. Zouden die valse sleutels worden gebruikt voor het versleutelen van berichten dan kan alleen de inbreker (die de bijbehorende privé sleutel heeft) het bericht weer leesbaar maken. Op Internet zijn er een aantal vertrouwde centrale instanties (Trusted Key Authorities) die publieke sleutels publiceren.

## 28.13 Digitale handtekeningen




**Notities**

Public key encryption kan ook worden gebruikt om de ontvanger van een bericht ervan te overtuigen dat de vermeende afzender ook de echte afzender is. Hiertoe dient de verzender van een bericht een digitale handtekening te plaatsen. Door een bericht te versleutelen met mijn privé sleutel geef ik dat bericht een unieke status. Iedereen ter wereld kan het bericht terugsleutelen met mijn publieke sleutel, maar het feit dat dit kan bewijst dat ik de afzender was. Alleen met mijn privé sleutel kunnen namelijk berichten worden versleuteld die met mijn publieke sleutel zijn terug te sleutelen!

Bij het veilig verzenden van een bericht vindt er dus dubbele encryptie plaats! Eerst signeer ik het bericht (versleuteling met mijn privé sleutel) en daarna versleutel ik het met de publieke sleutel van de geadresseerde. Digitale handtekeningen zijn vrijwel onmisbaar op Internet, waar immers vrijwel iedereen (potentieel) de afzender van een bericht kan vervalsen.

# Bijlage A

# Literatuurlijst


1. Firewalls and Internet Security, Repelling the Wily Hacker; William R. Cheswick, Steven M. Bellovin, ISBN 0-201-63357-4
2. PGP, Pretty Good Privacy; Simson Garfinkel; ISBN 1-56592-098-8
3. Practical UNIX Security; Simson Garfinkel, Gene Spafford; ISBN 0-937175-72-2
4. Managing Internet Information Services; Cricket Liu, Jerry Peek, Russ Jones, Bryan Buus, Adrian Nye; ISBN 1-56592-062-7