An Internet Gatekeeper

Hervé Schauer, Christophe Wolfhugel
Herve.Schauer@hsc-sec.fr, **Christophe.Wolfhugel@hsc-sec.fr**

Hervé Schauer Consultants

Abstract

As needs for both connectivity and security increase, it becomes necessary for organizations to build and manage secure Internet gateways.

IP is the internetworking protocol of today. Its use continues to grow. IP is the best-known protocol, it offers the user the best combination of services, and it is the protocol choosen by the main telecommunications carriers for their new services. IP is **the** essential protocol at this time, and thus we are concentrating on IP and ignoring other protocols. Effective security recommend the use of a single common routing protocol.

As the Internet becomes more open, the number of possible kinds of risks increases, both because input from the outside world becomes easier and because the possibilities for output increase. We will try to list the potential risks which must be protected against.

The goal is to obtain a reasonably open IP network with reasonnable security, i.e. to reach a good compromise between convenience and security.

To attain this goal, we define the standard security needs of an organization, and translate these needs into security requirements, cookbooks for verification, and technical solutions.

This paper will show a technical solution for the gatekeeper, but of course this is only a small part of the work. An important effort has to be made in order to train the staff properly in the new architecture and in its requirements. Several other documents, generally specific to each organization, describe all the prerequisites and daily tasks that have to be done in order to ensure a proper and safe network service.

Introduction

Setting up a reasonnably secure IP connectivity will require the completion of several tasks. A complete and proper architecture will require the following elements, generally both site and organization specifics:

— A *gatekeeper*, being the technical element to protect the network from the external world. At least one router, the *Gatekeeper Router* and one Unix machine, the *Gatekeeper Server* are required.

— The systems on the network to be protected are classified into two categories: *trusted machines* in which the *gatekeeper* will give a certain amount of trust, and *non-trusted machines* in which it won't give any trust.

— Specific software to run on the *Gatekeeper Server*, being for our needs specific telnet and ftp daemons which will be used for identification and authentication when a service between non-trusted machines and the external world is requested.

— A set of 5 documents:

- a description of the architecture, with its justifications,
- a manual of security requirements for a *trusted machine* status,
- a cookbook to be used to check the adherence to the requirements,

- a manual of security requirements for the Gatekeeper,
- a cookbook to be used to check the adherence to the Gatekeeper's requirements.

The TCP/IP protocol suite and IP connectivity both bring new risks which might compromise an information system. The risks have to be identified and classified in order to define solutions which will be used as a protection against these risks. With TCP/IP interconnection, risks can appear in both directions: external users may easily enter the private network and the new accessible external world opens many new perspectives to internal users.

The analysis of these risks will lead to the definition of the requirements to ensure the proper actions in order to reduce those risks. It is preferable to keep a global view of the problem, even if it is sometimes necessary to dig into details of technical solutions. It is also important to include in the procedure all elements which are relevant to the security, and not spend too many ressources on just a detail, however fascinating it can be. This work has of course to be done in conjunction with the target organization in order to get usable results.

The security needs are then translated into technical solutions which will be used by the appropriate people (computer security division, systems and network administrators and of course the end users). This will introduce to the global IP network security scheme and solutions, trying to answer at best to the needs: a gatekeeper, the requirements and cookbooks and of course the adequate identification and authentication software (and eventually hardware).

There are only two possibilities for a user to cross the Gatekeeper (to go out or come in):

1. If the origin or destination is a trusted machine, the Gatekeeper Router will let the adequate IP datagrams in/out.
2. In all other cases, the user will have to identify and authentify himself on the Gatekeeper Server who will act as a pass-thru telnet/ftp server.

In order to be trusted, an internal system will have to conform to the defined administration and usage rules defined in the requirements for trusted machines. Another service (such as computer center or computer security division) will be in charge of applying the cookbooks methods and deciding when a system may become trusted or when it will loose this privilege.

The Gatekeeper

The definition of the security architecture answering the users' needs will introduce the Gatekeeper. The Gatekeeper allows one to connect IP networks between an organization (the **internal networks**) and the Internet or any other network (the **external networks**). It allows one to do this interconnection with respect to the users' needs and to the security policy which has been defined.

Using such a gatekeeper answers most classical security needs an organization might have with IP networks interconnection. The connection generally goes to the Internet, but of course it can be any kind of network such as with another site or another company, or the gatekeeper can just be used to protect networks with different levels of confidentiality such as a civil network and one dedicated to military projects.

The proposed architecture deals with situations where some systems, the trusted machines, will have a free (or nearly free) access to the outside world and where all others, thanks to the gatekeeper's services, will still be able to use the external network from whatever machine they're on, such as a single PC (which can generally not be a trusted machine), or a machine without any real administration, in order to perform elementary functions such as telnet or ftp with minimal constraints.

The Gatekeeper is formed by an IP router with elaborated packet control facilities and a Unix machine, preferably running a Berkeley Unix, both connected by an Ethernet local to the Gatekeeper. The router is called the Gatekeeper Router and the Unix machine the Gatekeeper Server. Both hardware can be doubled in order to ensure some redundancy and thus better service, if the needs are there of course.

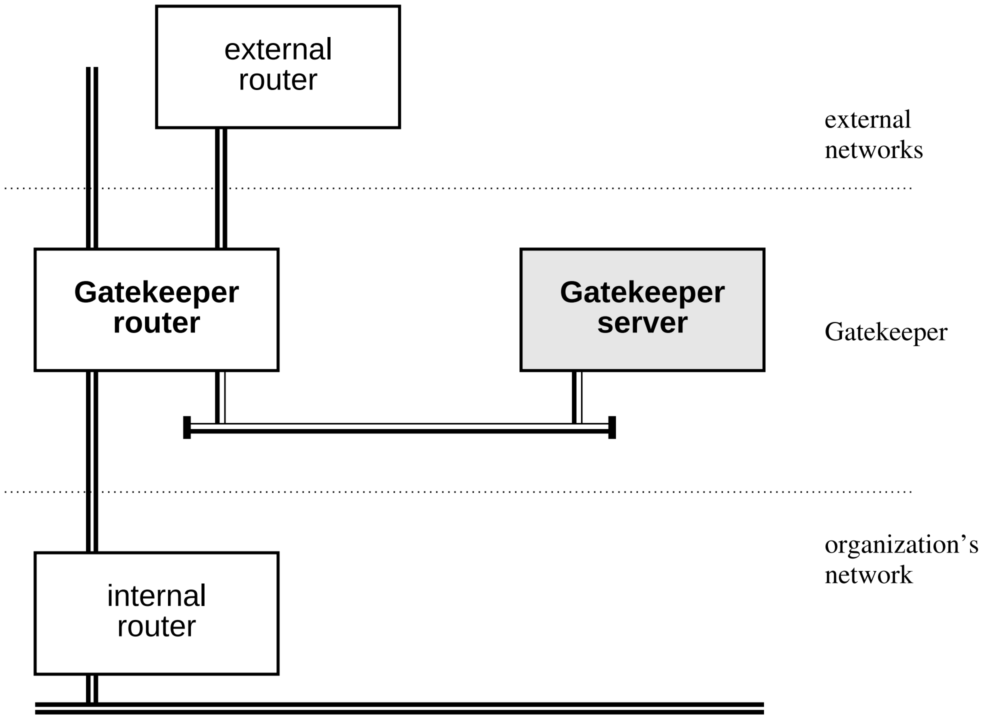

Figure 1. General framework of the Gatekeeper

Note that the *external router* and *internal router* are not part of the Gatekeeper. They have been drawn above as an example of what can be connected on each side of the Gatekeeper. Of course networks might be directly hooked to the Gatekeeper Router without the need of an intermediate router.

The Gatekeeper will act in both directions: it will protect the internal network against the outside world but also protect the outside world from unwanted action whose source could be inside the organization, after all, security is protecting yourself from the outside, it is also ensuring that external organizations can't hassle you for causing them trouble.

Using a gatekeeper is fundamental. Experience has shown many organizations being connected on the Internet on their behalf: one department needs an Internet connection, requests and gets it. But people might very well forget that they are also connected to the rest of the company's network. As a result, this isolated action has a bomb effect: the entire organization gets connected to the Internet. The goal of a gatekeeper is to give the users a good IP connectivity so that they won't have the need to get one themselves but would prefer using the Gatekeeper's services.

A security policy will anyway require the IP connectivity to be handled by some sort of central service, with the adequate highly competent staff, whose functions will be to serve all users among the organization in the domain of IP networks and security. This will guarantee a high quality service for all users. Such a connectivity should not be handled by a particular division or laboratory whose main function might even not be computers nor networking.

Also a centralized connectivity allows one to follow a stricter security policy than a non centralized one, and a better control on what's going in and what's going out, by using adequate filtering, an only authorizing communications between systems and users clearly known.

Experience also shows that the local networks can only be considered those on the same geographic site. If there are several sites using the same gatekeeper, all sites, except the one where the gatekeeper is, have to be considered as external networks in the same manner as the Internet is. Not doing so would require having a full control over the networks on the other sites. Even in the same organization it is nearly impossible to control its own site, so with remote ones.

Connections such as dialup-IP for use by employees also have to be considered as external networks, just because the authority can't control what's going on at the employee's machine.

Only local area networks are to be considered as the internal network.

A consistent and efficient policy needs to be defined, allowing the *authority* to do all necessary controls on the internal network. Another important point is to make this policy public so that users and administrators know it. People need not search for an IP connectivity, they need to remember that there already is one, that it's a providing a good service, and that its just waiting to serve some more users. People need also to be informed that if they are connected to the corporation network, they are not allowed to get their own connection with the external world.

As already explained previously, a user may cross the Gatekeeper in only two ways.

In the first situation, a communication is requested to or from a trusted system. The Gatekeeper Router lets the IP datagrams circulate as requested, as in essence the identification and authentication on a trusted machine is considered as acceptable. Telnet and ftp services are generally open to all trusted machines, but some services may be shutdown for the entire site, such as RPCs, whether the source/destination machine is trusted or not. Administrators of trusted machines can ask for some particular service. If it's not against the security rules then the appropriate configuration can be validated on the Gatekeeper Router. A machine will become trusted only after it has passed with success the adequate tests described in the cookbooks (described in following sections), guaranteeing the application of the requirements for a trusted system.

In the second case, when the system on the internal network is not a trusted machine (source/destination), identification and authentication will have to be done on the Gatekeeper server.

Of course the Gatekeeper will follow the defined requirements and will be regularly validated against the validation cookbooks. The validation is supposed to be done by some independant service, ie not the one managing the Gatekeeper. This can be the security departement or an external company.

In a few words, the Gatekeeper's features are:

— limiting the IP interconnectivity by filtering unwanted services,

— black listing undesirable sites,

— handling the list of trusted machines and of services they're authorized to use, on the other side blocking all traffic to/from non-trusted systems,

— authenticate on a (necessarily trusted) machine, aka the Gatekeeper Server, the users willing to use service to/from a non-trusted system,

— filtering incoming/outgoing connections with access control lists defined on the Gatekeeper Server, lists based on (service, site, user),

— controlling the usage of routing protocols,

— acting as the SMTP gateway for the organization,

— eventually being the organization's main News server,

— logging all important events (connections, usage statistics, error reports, ...),

— analysing and generating adequate reports from all logs coming from the router, the server and its software, as well as those from trusted machines,

— accounting of services usage,

— setting up internal/external DNS, in order to give only a limited view of the network to the external world (trusted machines only),

— handling a database with IP/Ethernet addresses equivalences,

— in fact, controlling in one central point the entire IP connectivity.

It is important to note that all the proposed security is based on the security of the architecture's main elements: the Gatekeeper Router and Server, as well as the trusted machines. Thats what gives the entire system its strength, and in fact also its weakness. The Gatekeeper protects internal networks and systems even if they are poorly administrated or do not follow the security guidelines. If one of those elements gets compromised, then the entire network security is compromised. As the Gatekeeper protects the networks and systems, it is important to have Gatekeeper and trusted machines to be correctly administrated. One open breach and a hacker can attack whatever system he wants, particularly non-trusted systems on which finding security holes might be much easier.

The security is itself based on the proper functionning of the Gatekeeper. A special attention will have to be devoted to its elements and to the trusted machines in order to ensure that all security requirements that have been defined for them are respected.

Identification and authentication

When a communication channel needs to be established to or from a non trusted machine, the user is required to use the services of the Gatekeeper Server. Two fundamental services are provided: FTP and TELNET. The identification and authentication is done by the replacement of the `ftpd` and `telnetd` daemons by our own. There are no user accounts on the Gatekeeper Server.

The new replacement daemons, called `in.gk-telnetd` and `in.gk-ftpd` are handled by `inetd` as a replacement for the old ones (`in.telnetd` and `in.ftpd`). If one does not trust `inetd`, it is possible to hack the code in order to have the daemons handle their respective incoming channels.

Other services might integrate this identification and authentification scheme in the future.

Both the implemented (telnet and ftp) servers ensure proper identification and authentification of the caller. Service will of course be denied in the case of improper identification (wrong user name) or authentification (password error). Source address and destination (requested) address are then checked against the authorization tables. Both source and destination must be authorized for the user in order to let the pass-thru telnet or ftp service to be launched. Of course the router will ensure that, except on trusted machines, no user can cross the Gatekeeper Router. The user shall instead connect to the Gatekeeper Server.

All events handled by `in.gk-telnetd` and `in.gk-ftpd` are logged thru the *syslog* service.

Sample of connections and syslog dumps:

```
<hsc.schauer: 63> telnet gk.hsc-sec.fr
Trying...
Connected to gk.hsc-sec.fr.
Escape character is '^]'.

gk.hsc-sec.fr

login: schauer
Password: password_on_the_gatekeeper_server
Host: itesec.hsc-sec.fr

Access authorized

UNIX(r) System V Release 4.0 (itesec)

login: schauer
Password: password_on_the_final_station
UNIX System V Release 4.0 AT&T NEWS3400
itesec
Copyright (c) 1984, 1986, 1987, 1988 AT&T
All Rights Reserved
Last login: Mon Jul 13 11:56:28 from spock.hsc-sec.fr
<itesec.schauer: 346>
```

Following events have been sent to the syslog. Note the signification of the abbreviations:

sa source address

u identified user (or _UNKNOWN if not in the database)

pt sequence number of password tries, starts at one for each new connection.

pc flag indicating whether the password has been changed (1) or not (0). Available with the telnet service only).

da destination address, if already known, otherwise _NOHOST.

The last part of the line contains service messages indicating the state of the connection. Following entries have been cut into multiple printed lines for convenience.

```
Jul 21 14:27:45 gk unix: Jul 21 14:27:45 gk-telnetd[10716]:
sa=192.70.106.33 u=schauer pt=1 pc=0 da=itesec.hsc-sec.fr start of session
[...]
Jul 21 14:27:45 gk unix: Jul 21 14:37:45 gk-telnetd[10716]:
sa=192.70.106.33 u=schauer pt=1 pc=0 da=itesec.hsc-sec.fr end of session
```

Failed connections:

```
$ telnet gk.hsc-sec.fr
Connected to gk.hsc-sec.fr.
Escape character is '^]'.

gk.hsc-sec.fr

login: wolf
Password: a_bad_password
Login incorrect
login: schauer
Password: another_bad_one
Login incorrect
login: notwolf
Password: last_but_not_least
Login incorrect
Connection closed by foreign host.
```

After three mistakes, the session is closed. Of course the *syslog* has reported the hacker's try, note *pt* being incremented up to three:

```
May  6 14:31:42 gk unix: May  6 14:31:42 gk-telnetd[10778]:
sa=192.70.106.33 u=wolf pt=1 pc=0 da=_NOHOST bas password
May  6 14:31:42 gk unix: May  6 14:31:47 gk-telnetd[10778]:
sa=192.70.106.33 u=schauer pt=2 pc=0 da=_NOHOST bad password
May  6 14:31:42 gk unix: May  6 14:31:59 gk-telnetd[10778]:
sa=192.70.106.33 u=_UNKNOWN pt=3 pc=0 da=_NOHOST bad password
```

A user may also pass identification/authentification and request a host he's not authorized to join:

```
$ telnet gk.hsc-sec.fr
Connected to gk.hsc-sec.fr.
Escape character is '^]'.

gk.hsc-sec.fr

login: wolf
Password: wolf's_password
Host: 134.135.136.137
Unauthorized destination.
[...]
```

```
May  6 14:35:57 gk unix: May  6 14:35:57 gk-telnetd[10790]: sa=192.70.106.33
 u=wolf pt=1 pc=0 da=134.135.136.137 destination address rejected
[...]
```

The behavior of the ftp server is very similar, but of course the interface is totally different:

```
<hsc.schauer: 63> ftp gk.hsc-sec.fr
Connected to gk.hsc-sec.fr.
220- gk.hsc-sec.fr FTP server / HSC ready.
     After logging in, use 'site machine' to connect
     to the desired machine.
220  Time is 1992/07/24 16:48:21 GMT
Name (gk:schauer): schauer
331 Password required for schauer.
Password: password_on_the_gatekeeper_server
230 User schauer logged in. Please select your host.
Remote system type is UNIX.
ftp> site kirk.hsc-sec.fr
220 kirk FTP server (NCC-1701) ready.
ftp> user schauer
331 Password required for schauer.
Password: password_on_kirk
230 Welcome on board Captain schauer.
ftp>
```

The syslog will report following lines. Note that the *pc* field has been kept in the line but it has no signification as password change has not been implented in the ftp server.

```
Jul 21 14:27:45 gk unix: Jul 21 14:27:45 gk-telnetd[10716]:
sa=192.70.106.33 u=schauer pt=1 pc=0 da=kirk.hsc-sec.fr start of session
[...]
Jul 21 14:27:45 gk unix: Jul 21 14:37:45 gk-telnetd[10716]:
sa=192.70.106.33 u=schauer pt=1 pc=0 da=kirk.hsc-sec.fr end of session
```

With unauthorized destinations:

```
<hsc.schauer: 63> ftp gk.hsc-sec.fr
Connected to gk.hsc-sec.fr.
220- gk.hsc-sec.fr FTP server / HSC ready.
     After logging in, use 'site machine' to connect
     to the desired machine.
220  Time is 1992/07/24 16:48:21 GMT
Name (gk:schauer): schauer
331 Password required for schauer.
Password: password_on_the_gatekeeper_server
230 User schauer logged in. Please select your host.
Remote system type is UNIX.
ftp> site 134.135.136.137
550 You are not authorized to call 134.135.136.137.
ftp> site 134.135.136.138
550 You are not authorized to call 134.135.136.138.
ftp> site 134.135.136.139
221-You are not authorized to call 134.135.136.139.
221 cul8r.
ftp>
```

The *syslog* will indicate:

```
May  6 14:35:57 gk unix: May  6 14:35:57 gk-ftpd[10790]: sa=192.70.106.33
u=schauer pt=1 pc=0 da=134.135.136.137 destination address rejected
May  6 14:35:57 gk unix: May  6 14:36:27 gk-ftpd[10790]: sa=192.70.106.33
u=schauer pt=1 pc=0 da=134.135.136.138 destination address rejected
May  6 14:35:57 gk unix: May  6 14:36:57 gk-ftpd[10790]: sa=192.70.106.33
u=schauer pt=1 pc=0 da=134.135.136.1398destination address rejected
```

Following figure shows how a ftp connection thru the Gatekeeper Server is handled:

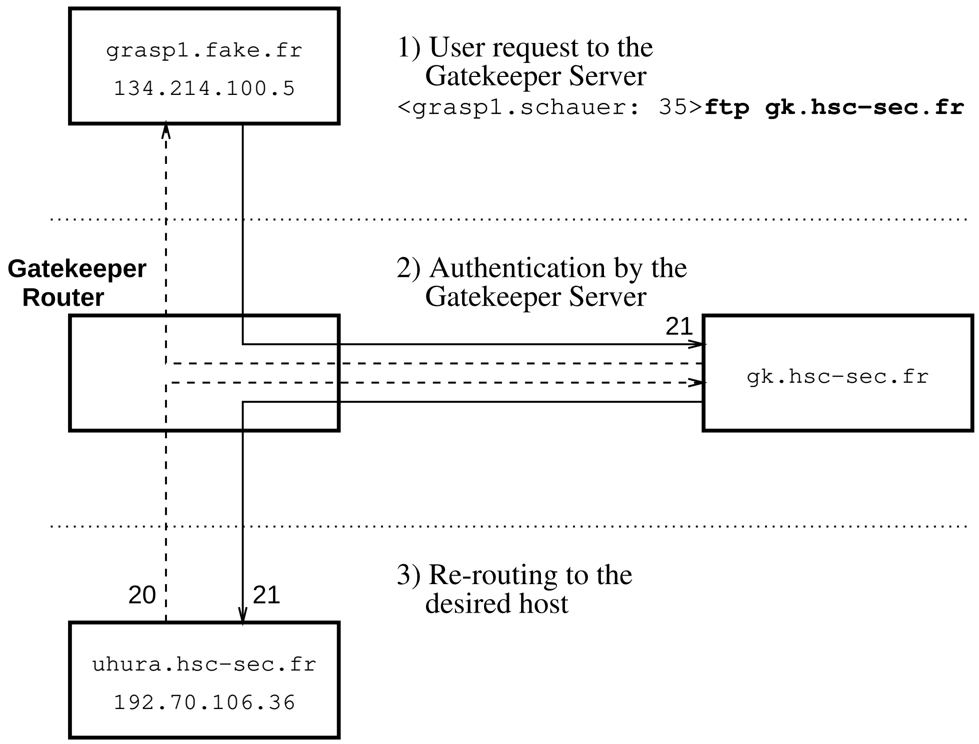

Figure 2. Sample FTP connection thru the Gatekeeper Server

The arrows are indicating the direction of establishement of the session, not the direction of the data transfer.

Configuring the authorization tables for both daemon is an easy task. Three system tables have been introduced:

— a password file,

— a group-like file for source address filtering,

— a group-like file for destination address filtering.

All of these files have to be handled with the same care as the real `/etc/passwd` and `/etc/group` as they contain vital information.

The password file is a lightweight *Unix password file*, ie it has the same number of fields, just that some of them are unused. We currently only use the *user* and *password* fields. All other may or may not contain information. Identification and authentication for the services is based on the contents of this file.

```
wolf:1234567890123:::Christophe Wolfhugel::
schauer:2345678901234:::Herve Schauer::
```

The group files also have the same syntax than the unix `/etc/group` file, except that most fields have beem changed to fit our needs:

```
134.214.0.0:255.255.0.0::wolf
192.70.106.0:255.255.255.0::schauer,wolf
192.70.107.1::schauer
```

The first field corresponds to the network address which is authorized and the second is the netmask to apply. A netmask of 0.0.0.0 authorizes anyone, whereas 255.255.255.255 only gives access to the machine indicated in the first field. The last field is the list of authorized users. To summarize: the IP address (source/destination) is logically ANDed with the mask. If the result is the indicated network, the source/destination is validated, otherwise it is rejected.

For the example, consider the previous sample file as a source address authorization table. The first line will authorize the user *wolf* to call from any machine in the 134.214 network. The second line authorizes both *schauer* and *wolf* to connect to the 192.70.106, and finally, only *schauer* may call from the 192.70.107.1 machine.

The destination group file is used for the same checking against the requested machine once the user has been properly identified and its source machine is authorized.

We use the standard Unix format just because our code implementation uses the Unix system calls to access them!

Security requirements for trusted machines

The defined security architecture does also specify too fundamental books defining the set of requirements in order to get to the wanted security level. The first book describes the requirements for the security of trusted machines, the second describes the requirements for the security of the Gatekeeper itself. Each of this book is completed by a cookbook containing adequate test-cases which will be used for validating the effectiveness of the requirements.

The requirements for a trusted machine allow to determine whether or not a machine (and its users), or a set of machines on a network, will be allowed to use external services, or receive external information, without using the identification and authentication of the Gatekeeper Server. Once a system gets the agreement, it becomes a trusted machine. That means that the Gatekeeper will trust it, and in fact trust the identification and authentication done on the target system. This also means that its administration is considered and being correct (according to the defined requirements).

The manual of security requirements for trusted machines contains a little bit more than 150 requirements, classified in several chapters. There are two set of requirements:

- requirements,
- guidelines.

A requirement must be followed, whereas a guideline should be followed, it is highly recommended but not mandatory.

Each requirement will belong to a given class. Two classes are currently already used:

- generic requirements,
- Unix specific requirements.

Generic requirements are supposed to be applicable to any kind of operating system, whether it is Unix or not. When there is a direct application, its Unix counterpart is indicated. As an example, methods for choosing a good password is a generic requirement. Unix specific requirements are depending on the Unix system, as an example the `/etc/exports` and `/etc/dfs/dfstab` files for the NFS system. All of the given requirements are applicable to Unix, and about half of them are specific to this system.

Requirements and guidelines are about:

- administrators, and as example: their training level,
- users, and as example: their responsibilities when using a trusted machine,
- physical access and control to the trusted machine, as example: controlling the usage of the PROM boot password,
- identification and authentication of users, part of this would be all problems related to the choice of a bad password,
- managing user's sessions, for example how their `PATH` is set, or the access rights on `/dev/kbd`,
- accounts management with a special interest to *root* accounts (UID=0), example: definition of a policy concerning the use of shared passwords on administrative accounts,
- checking against bad permissions on the files, directories, as example: how are temporary directories protected,
- management of users by the administrators, for example what action should be taken when a user definitely does not respect the requirements about the choice of his password,
- accounting, logging of events, definition of the handling procedures of all those important files.
- general organization, as example: which modifications will have to be indicated to the departement delivering the agreement to a system,
- access control and machine identification, example: use of fully qualified domain names,
- network services, each service has to be detailed: TELNET, FTP, SENDMAIL, NFS, NIS, x11, DNS, etc. This part is most important one.

With this requirements book goes a cookbook to be used in order to check the good application of the requirements. The cookbook will give methods and test-cases for verifying each of the defined requirements.

For each test-case, it is indicated who should use it, when it should be used and with what frequency. Is also indicated if a test-case should be run in exploitation or test configuration. The previsible result is indicated for each test-case as well as a non exhaustive list of actions to perform if the result is not the one expected. The goal is to facilitate the work of the administrators and of the auditing staff who will deliver (or take them back) the agreements and ensure that they are valid in the time. In some cases, the solution is somewhat easy (such as using COPS) whereas in others users not behaving as expected must be red-handed by the auditing service (such as verifying if users really use `xlock` when they leave their screen).

Requirements for the security of the Gatekeeper

After having defined the requirements for trusted machines it is necessary to do the same job for the Gatekeeper itself. This book will allow one to validate the administration and exploitation of the Gatekeeper against the defined security policy.

This book contains over 300 set requirements concerning both the Gatekeeper Router and Gatekeeper Server. The requirements are classified into three categories. Apart from the requirements and guidelines indicated in the previous book (requirements for the trusted machines), we added informational data. This informations being suggested procedures to follow in case of an incident (hacker, hardware failure on the Gatekeeper, etc...).

Of course the contents of the book will be highly dependant on the hardware and software platforms that will be used, as an example, each router will have its own rules for handling packet filtering.

The requirements and guidelines are about:

- physical protection of the Gatekeeper,
- the qualities of the Gatekeeper's system and network administrators.
- the administration of the Gatekeeper Server,
- the protection against access to the Gatekeeper Server from the network. Requirements are detailed for each service: DNS, equivalences, IP routing, TFTP with the Gatekeeper Router, electronic SMTP mail gateway, using X11 on the Gatekeeper's ethernet, SNMP, identification and authentication services for TELNET and FTP (with `in.gk-telnetd` and `in.gkftpd`), etc...
- administration and management of the Gatekeeper Router,
- IP filtering administration and management on the Gatekeeper Router,
- logging the Gatekeeper usage,
- handling incidents, Gatekeeper reconfiguration if necessary,
- backups and archiving.

The manual of requirements for the Gatekeeper is completed by a cookbook which will be, in parts, specific to each implentation. This cookbook will reuse the set of requirements and proposed for each of them test-cases to be used during the Gatekeeper normal operation in order to verify the application of the requirements.

For each test-case, it is indicated which part of the Gatekeeper (router, server) is concerned, in which case to use it and with what frequency. The normal-operations result is also indicated as well as actions to perform in the case of a erroneous result of the run. The goal is to allow the administrators to follow as easily as possible the requirements, and to give to the controlling team (security departement) a convenient mean of verifying the effectiveness of those requirements.

General organization

In order to have a successful use of the proposed security architecture, it is necessary to design well the general oranization which will accompany the installation of the gatekeeper. Installing `in.gk-ftpd`, simply giving the manual of security requirements to the system administrator, applying the cookbook methods in hurry is for sure not the good way of proceeding...

Security is a whole. This is why t is necessary to keep a global view on the problem and design the solutions in concordance with the already existing organization. One must think on the human ressources: who will be in charge of which task, what training will be given to these persons, how will the technical service delivering agreements be organized, what authority will be competent in case of conflicts, who will be in charge of external relations (with the NIC for requesting IP addresses, ...) are as many questions that need to be asked – and answered properly – in order to setup properly a global security policy.

A really important point is information and education of end users. They are the first ones concerned by security and its for them that all this has been setup, to allow them to do their work in good conditions. It is necessary to explain (and persuade) them that the security policy is not here to annoy them but to give them a better service. That's why all members of the organization should be be involved in the action.

Hurrying in order to get connected to the Internet as soon as possible is neither a good solution if one is not ready. Setting up a security solution requires some time. A test period, allowing to validate the Gatekeeper in production is welcome, during this period the internal network will not be connected, but rather there will be a fake internal network with fake machines. Meanwhile it is possible to start checking the machines who have requested an agreement to become a trusted machine, so that once the testing period of the Gatekeeper is finished they can be immediately connected and access the external IP networks.

Availability

The proposed security architecture is running in production in several major organizations in France for their Internet connectivity or between networks having different security labels.

Conclusion

As needs for connectivity with the outside world increase in all organizations, needs for security also do so. Those two requirements are not contradictory, and in fact a security architecture based on a gatekeeper allows to get a good compromise between convenience of use and security.