

The Flood-Gate Principle - a Hybrid Approach to a High Security Solution*

Ernst-Georg Haffner, Thomas Engel, Christoph Meinel

Institute of Telematics, Bahnhofsstr. 30-32,
D-54292 Trier, Germany
{Haffner, Engel, Meinel}@TI.FhG.de

Abstract. The classical role of a firewall consists in protecting a computer network against attacks from the outside world, especially the Internet. Firewalls are often expensive, hard to configure and they are comprehended only by experts. Sometimes the level of security is too high to use a firewall, and information flow has not to be “online”. Here we propose to use “flood-gates” as described in the following. They provide a modern, simple and easy-to-understand method to secure a network on a very high security level. E-mails, plain files and all sorts of electronic data can be exchanged over such flood-gates without possibly compromising the “own” network by the most dangerous classes of attacks. Information passes through the flood-gates even though there is not a single moment of a physical connection between the own network and the outside world. The disadvantage of service restrictions can be overcome by a multilevel security approach. As a practical example a concrete “real-life” implementation of the *flood-gate principle* in the financial sector is described in detail in this paper.

1 Introduction

Nowadays networks are a natural part of information technologies. The Internet as “net of nets” grows rapidly and the need of information flow between the “public” Internet and “private” computer networks increases. We will call the latter the “inner network”, “internal net” or short “inside”, while we speak of the former as the “outside” or the “outside world”.

The inner network can belong to any sort of company, office, institution or university and is to be protected against attacks from the outside world, for instance the Internet or any other public accessible or private computer system.

But we will also focus on another security problem: the attack from the inside, the internal net. “Most computer crimes are in fact committed by insiders” ([13], p. 4). Secret enterprise data must not arrive at the outside, neither unintentionally nor by bad intention.

* In Proc. "International Conference on Information Security and Cryptology", 1998, Seoul, South Korea, pp. 147-160

The first thing someone has to do to protect an inner network (and certainly the information inside) is to constitute a “security policy” (a good definition can be found in [3]). The proposed *flood-gate principle* (FGP) - described in detail in the next paragraph - concerns the following questions that must be answered by any security policy:

- What kind and size of data is permitted to pass from the outside to the inner network?
- What kind and size of data must not leave the internal net?
- How can one be sure that hacker attacks do not compromise the inner network while information is transferred from outside in or vice-versa?
- How can one avoid receiving Trojan Horses, computer viruses, worms and other sorts of “beastware”¹?
- Is the applied security mechanism clear, simple and available?
- Which services are provided and what about user inconvenience?
- What about the costs?

The questions above focus on a central point in networking: the *gateway* between two networks. The classical solution is a “firewall”, even though not all of the aspects mentioned can be answered positively by this system. We will discuss soon a lot of problems of firewall architectures and compare them to the FGP.

The requirements of secure data exchange on the one side and comfortable services on the other lead to a hybrid approach or “multilevel security” [34]: the flood-gates protect only the “high secure” part of the internal net while other “untrusted” and possibly unsecured parts provide several standard protocols and services as “http”² [35], “telnet”, “ftp”³, “rlogin” (all in [14]), “smtp”⁴ or “sendmail” (both in [6]). We summarize them as *standard services*. The paragraph “4 A practical example” will deal with the security requirements of the financial sector, especially the implementation of the FGP of an important European bank. In the detailed description it will be presumable surprising how heavy the psychological acceptance weighs when security considerations are discussed.

Most of our ideas presented in this paper are not new, as explained in the paragraph “5 Related work”, but we think that the combination of the greatest thoughts together with some special additions (missing links) justifies the use of the new term “*flood-gate principle*”.

2 Flood-Gates

2.1 Suppositions

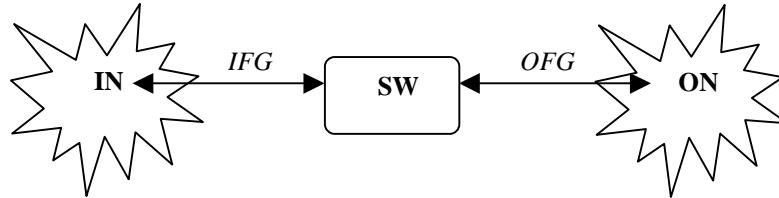
To understand the nature of electronic flood-gates let us consider two separate networks of computer systems. An “inner network” (IN) and an “outer network”

¹ This type of software will be discussed in paragraph “3.5 Beastware”.

² Hypertext Transfer Protocol

³ File Transfer Protocol

⁴ Simple Mail Transfer Protocol



(ON). Between these we construct a rather small and simple network that is called sluice-way (SW). Figure 1 sketches the topology of these networks.

Fig. 1. Topology of the flood-gate networks

The connection between IN and SW is called “inner flood-gate” (IFG) and that between SW and ON is called “outer flood-gate” (OFG). These are already the abstract requirements of the FGP. Concrete spoken, we can think of IN as a companies internal net (LAN⁵ or WAN⁶), and the Internet plays the role of the ON; the SW is our special flood-gate net which consists - in the simplest case – of only one computer, a so-called flood-gate communication server (FG-CS). The connections between these networks are at most ISDN⁷-wires. The flood-gates IFG and OFG are called “open” if the corresponding connection is established while the gates are “closed” if the respective connection is not active at the moment.⁸

The essence of the hole construction can be expressed with the following rule:

Rule 1:

It must be physically impossible that both IFG and OFG are open at the same time.

At least one communication server is part of every network to provide the flood-gate data exchange: an inner net communication server (IN-CS), an outer net communication server (ON-CS) and FG-CS as described above.

2.2 Information flow

At first we will glance on the flow of information from the inner network to the outside world.

Both IFG and OFG are closed. Now IFG opens and data is transferred from IN to SW. Then IFG is closed and data may be analyzed within SW.⁹ When there is for instance a mail with a big attachment or something else that is not allowed to be sent

⁵ Local Area Network

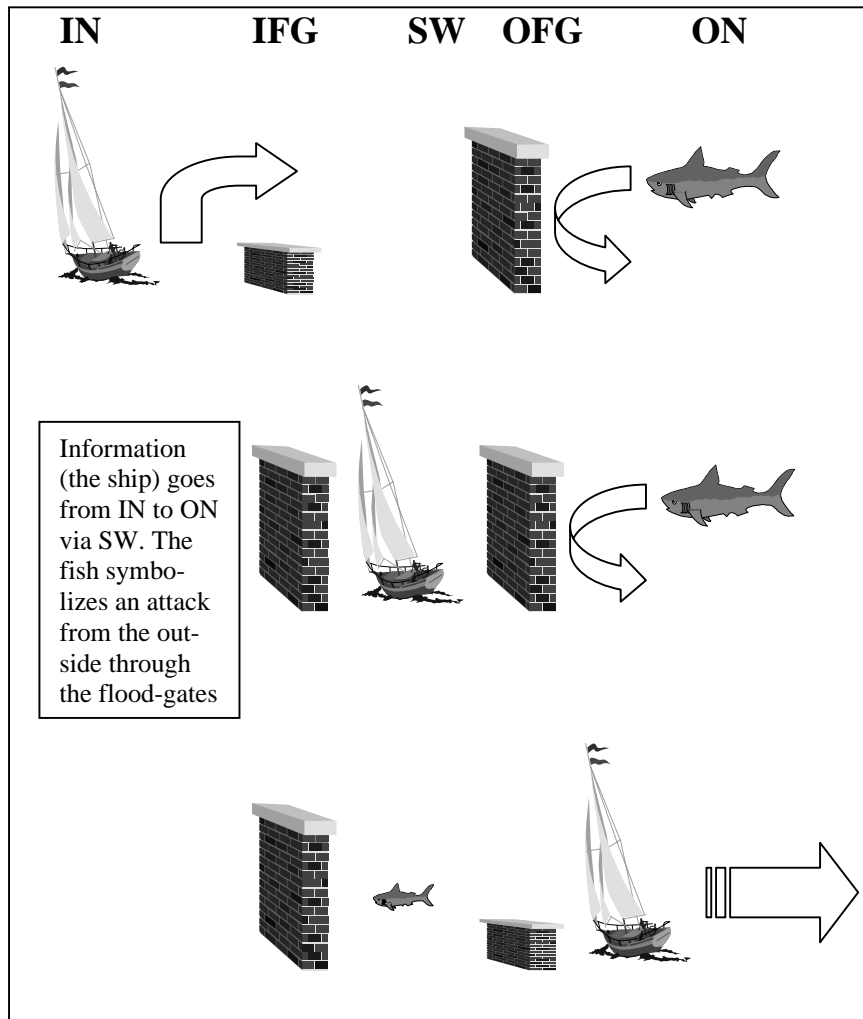
⁶ Wide Area Network

⁷ Integrated Services Digital Network

⁸ This setting should not confuse the reader, but belongs to the metaphor of flood-gates: if and only if the gate is open, the ship can pass from one side to the other; this means that the connection between the two corresponding networks is established (“closed”).

⁹ Certainly some if not all analyzing processes can be done on the IN-side!

by the security policy, the document is rejected by the SW and sent back to the IN. The FG-CS as heart of the SW is a fully functioning computer system where all kinds



of analyzes can be made. The OFG opens after IFG has been closed and data can be transferred from SW to ON, for instance to the Internet.

Fig. 2. Information flow through the flood-gates

Now let's have a look on the other side: information must be transported from ON to IN. This time OFG opens first and lets the data "come in". Then the gate is closed and the connection between SW and ON is no longer established. Within SW the incoming data must be analyzed and checked for viruses and Trojan Horses (see the

next paragraph for details). Finally IFG opens and the data is transferred to the inner network. Table 1 resumes the possible states of both gates IFG and OFG.

No.	Description of the processes	IFG	OFG	State
0	Initial state; analyzing of outgoing data on IN side possible	C	C	α
1	Data is transferred from IN to SW	O	C	β
2	Data is stored in the SW, possibly analyzed here again	C	C	α
3	Data is transferred from SW to ON	C	O	γ
4	Data is delivered to the ON; other data from ON (i.e. e-mails from the Internet) is collected here	C	C	α
5	Data is transferred from ON to SW	C	O	γ
6	Incoming data is checked for viruses, Trojan Horses etc.	C	C	α
7	Data is transferred from IN to SW	O	C	β

Table 1. Different states of the flood-gates; C=closed, O=open

One can easily determine that all processes with the same state (α , β or γ) of the two flood-gates can be treated at the same time. An open flood-gate provides the means to exchange data bidirectionally¹⁰. To explain the metaphor: while data passes through the flood-gate from the inside to the outside world, its security level is lowered whereas it is raised while passing from the outside to the inside network. Again we should recapitulate that *it must be physically granted* that in no case both flood-gates are open at the same time.

2.3 Physical realization

The FGP does not fix exactly what kind of hardware must be used to realize the mechanism. We thought of TCP/IP¹¹ [5] based systems as communication servers in the confounded networks IN, ON and SW, because this is the standard protocol of the Internet and base for the higher level protocols and programs stated earlier as “standard services”.

As mentioned above the flood-gates themselves are normally ISDN-connections (2 B-channels with channel bundling for one gate or one B-channel per gate respectively). It is possible to realize the IFG as part of an enterprises telephone installation. Usually the OFG is a long distance wire and belongs to a telephone company. If the data volume on the flood-gates exceeds the wires’ capacity, more ISDN devices can be connected to the communication servers or additional flood-gates can be added to the SW (for problems with ISDN connections see [12] and [3]).

Our proposal as operating system for the communication servers is UNIX [14], even though it is not considered to be very secure [7]. The connections between the

¹⁰ Its the same as in real-life: ships can pass the stages of a flood-gate in both directions, depending on where they want to go.

¹¹ Transport Control Protocol/Internet Protocol

servers are almost always PPP¹²-connections [14]. Very critical applications as “sendmail” – from the standpoint of security [3] – can be used to exchange the data. We recall here that it must be physically impossible for every program, a self-written tool or a standard application, to open both IFG and OFG at the same time. There are several possibilities to fulfill this condition. It can be achieved by using special ISDN end devices (non standard NTBA’s¹³) that allow only one connection at a time. Some telephone installations can be configured to solve this problem. Other solutions depend on the concrete devices that should be used. One must not rely upon software implementations because they may be wrong or penetrated! To verify that there are no viruses or other kind of penetration of the FG-CS, the IN side must check the state of the FGN at the beginning of every connection!¹⁴

In the paragraph “4 A practical example” we will present a way to grant this aim without changing any existing hardware environment.

3 Comparison between flood-gates and firewalls

3.1 Definitions

Before we discuss a concrete implementation of the FGP we have to compare such a method to classical firewall gateways.

Cheswick and Bellovin define the expression “firewall” in their famous book “Firewalls and Internet Security” [3] as “a collection of components placed between two networks that collectively have the following properties:

- All traffic from inside to outside, and vice-versa, must pass through the firewall.
- Only authorized traffic, as defined by the local security policy, will be allowed to pass.
- The firewall itself is immune to penetration.” ([3], p.9)

On the one side we can state that flood-gates implements a very special kind of firewall between ON and IN. On the other side firewalls can surely be a part of a flood-gate architecture: between ON and SW and between SW and IN is a perfect place to insert a (classical) firewall.

Most of the existing firewalls are packet filters. They analyze the source and target IP-address of each packet, check its TCP port number and reject the packet if one of them is not allowed to pass. There are many possibilities to deceive this system and to pretend being someone else. We will have a closer look at this aspect soon (“3.2 Security considerations”). Other kinds of firewalls are the “Circuit-Level” and the “Application-Level” gateway. The former is a more elaborated and complex type of packet analyzer that functions like a proxy server. Details can be found in [3]. The latter describes a scheme for “special-purpose” gateways that allows only some

¹² Point to Point Protocol

¹³ Network Terminal Basis Adapter

¹⁴ At most it is enough to check the „fingerprints“ of the concerned programs and shell-scripts

applications to pass through. This resembles our flood-gates and is discussed in the paragraph “5 *Related work*”.

3.2 Security considerations

What we want to show next is that the providing of standard services - as described above – forces security lacks. There are many known classes of attacks against internal nets. We compare the possible penetrations of a networks secured by a classical, IP-filtering firewall and secured by a flood-gate. Table 2 shows the results:

Class of attack	Description of the class of attack, examples of attacks	Firewall penetrat.	Flood-gate penetration
Stealing passwords	Passwords are stored in plain files; or can be logged at the IP-layer; dictionary attacks	Possible ¹⁵	Impossible, even though a password is known there are no standard services to use them!
Social engineering	Passwords are guessed or the users tell someone their passwords (via mail-spoofing, phoning etc.)	Possible	Impossible, same as above
Bugs and backdoors	Errors in program code; or viruses and worms create new “entrances”	Possible	Impossible, network is secured physically!
Authenticat-ion failures	Programs show login masks and send the passwords to someone outside	Possible	Senseless, network is secured physically!
Protocol failures	TCP sequence number attack; tunneling; message encapsulating; tiny fragment attack; overlapping fragment attack [36]	Possible	Impossible, protocol is not provided
Information leakage	Some protocols present information (e.g. finger), that can be used for others types of attacks (“social engineering” etc)	Possible ¹⁶	Senseless!
Denial-of-service	Worms, Trojan Horses, and viruses can crash a system ¹⁷	Possible	Possible, but the FGP provides a good platform to defend such kind of attacks

Table 2. Classes of Attacks [3]

Even Cheswick and Bellovin state that “... firewalls provide almost no protection against problems with higher level protocols, except by peeking” ([3], p. 82).

There are some tools to find out if an *attack has been committed*. For instance “tcpdump” [14] is quite useful but as with analyzing of log files, one can state: often, this is too late!

¹⁵ Certainly only an insider is able to steal a password without passing any firewall. Later he can use it from the outside to penetrate the inner network!

¹⁶ Even though a firewall may not be responsible for that attack, it can be penetrated by its results!

¹⁷ This „beastware“ can – of course – penetrate a network by other classes of attacks too!

3.3 Psychological factors

There is a remarkable and important aspect in security policies: the psychological factor. The feeling “to be secure” is a most distinct aspect while discussing with the confounded persons. This means that the way of making the internal net secure must be understood to a very great extent. Firewalls trouble because of their complexity, the need to be attended by experts and regular new reports of penetration (for instance lately ATM¹⁸-problems [26], [18]). “Furthermore, even if we had implemented defenses against the known flaws, we would still be vulnerable to next years.” ([3], p. 83). What about flood-gates? They are easy to use, simple to understand and everyone can trust such a rather harmless architecture. “Keep it simple, if it is complex, it’s probably wrong” ([3], p. 253). In a complex program with more than one million lines of source code one can never be sure to have a bug-free tool. Sometimes even the vendors of software or hardware must not be trusted [2].

But what about the service restrictions? Does the “normal” user accept that there are only few services to get data from the inside of the security perimeter [13]? Perhaps the question is to be asked as follows: “Do I accept some inconvenience in the treatment if I accept the hole way to secure my network?” We saw in our practical work that the acceptance is much greater when the reasons are clearer and easier to understand.

3.4 Costs

There are not only hardware and software costs to consider, but also regular administration work is to be paid. How easy is it to adapt some router configurations? How long is the time delay? What about new ways to penetrate a network? All these questions result in a remarkable sum if one decides to protect the internal net through firewalls [3].

We believe that flood-gates are much cheaper. There is no need of special hardware and even critical software can be used. The program “sendmail” is a good example. From the standpoint of classical security considerations it can not be used ([3], p. 82), but as part of the FGP its use is acceptable, because most of the penetration attacks are not possible¹⁹. The administration of the hole flood-gate construction is easier, because the network is secured physically, not by software configuration. Furthermore flood-gates can be applied from the very beginning without having fixed all details needed to configure a firewall.

3.5 Beastware

As we saw in Table 2, there are even some possibilities to penetrate a network that is secured by the FGP. Trojan Horses, worms and viruses, also called “beastware” [2] are still a remarkable problem. They can arrive at a network “offline” (for instance via

¹⁸ Asynchronous Transfer Mode

¹⁹ Indeed, we will show in “4 A practical example” the secure use of this very large program

e-mail or as part of other files [13]) and can put damage to the system without any active doing of the attacker who sent it (“denial-of-service attack”). The only position to solve this problem is the analyzing stage of the SW. Here every known tool to check the files can be placed and called while the data waits for opening the IFG. These can be virus killers, macro virus detectors and other procedures to detect viruses ([2], [4]). Even though this grants no absolute security, it is the best way to protect the inner network. Also (classical) firewalls can do no more.

3.6 Legal considerations

Is it legal to verify (outgoing and incoming) mail? It would not be very helpful to find an attacker by monitoring and analyzing files if – at the end – the creators of the security policy were arrested. This is a large and very complex field and as far as the Internet is concerned, a case for national and international laws. Some arguments make us feel that it is legal to analyze all sorts of data if everybody of the confounded knows about it ([3], [33]).

For a closer look to legal aspects we refer to [15], [32], [28], [12] and [13].

4 A practical example

4.1 Security considerations of the financial sector

The *flood-gate principle* (FGP) has been applied by the *Institute of Telematics* at first for the mail exchange mechanism of an important European bank. The aim of the project had been the secure exchange of data (in form of e-mails) between the bank and an Internet provider. It had to be granted that information flow had to happen on the highest security restrictions. The inner net must not be penetrated in any way and only very little information may leave the bank. But browsing in the Internet should nevertheless be granted.

So we did two things: we considered the actual network of the bank as the security perimeter which must be secured by flood-gates.

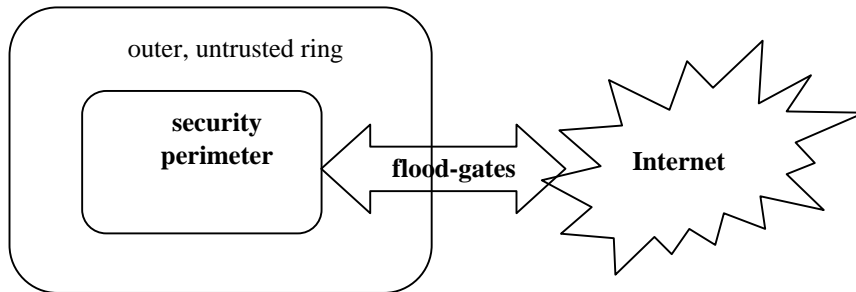


Fig. 3. Use of flood-gates in a bank environment

And another “outer” ring should provide Internet surfing and was considered as untrusted (Figure 3). All e-mails²⁰ and their attachments should be analyzed for “beastware” (incoming) and “critical content” (outgoing).

The bank uses a token-ring MicrosoftTM-NT network with NT 4.0 workstations as front-end computers. All systems - except some of the system administration - are installed without any flexible device (no floppy disk, no tapes). So there is no possibility for an employee to put files into “his” workstation or to receive some without asking the system administration.

Some PC’s of the “outer ring” are connected to the internet via ISDN-wires, but there is no physical connection between the security perimeter and this computers.

We chose “sendmail” as the program to transfer the mail files from one server to the other. It is a rather giant tool and can be configured to queue or forward electronic mails at any time [6].

The way to transfer data from the Internet (or the outer ring) into the security perimeter is rather simple: the file may just be sent as e-mail (with own receiver address) and so it passes through the flood-gates.

4.2 Physical realization of the flood-gates

We installed a Linux (Kernel 2.0.33) [20] Compaq-PC on the side of the security perimeter as IN-CS (communication server of the inner network). The FG-CS has also been a PC of the same type. At first the SW consisted of only the FG-CS. But in the meantime the SW has been growing. On the provider side (far away), where all e-mails from the Internet are received, the SMTP mail-server is used also as communication server of the ON side (ON-CS). It is possible to connect to the ON-CS via ISDN wires.

The local environment showed us a very simple way to obey “rule 1”. A standard ISDN-NTBA provides two B-channels and a D-channel, that means, at most two

²⁰ Certainly from the very beginning all users were informed not to send “private” e-mails to friends from the inside of the security perimeter ([33], [21]).

active connections with 64K baud bandwidth are possible. The trick is to connect *both* FG-CS and IN-CS to the *same NTBA!*

When IFG is open, both B-channels are used, one to call out and one to receive the call (*with the same NTBA!*). So it is physically impossible for the FG-CS to establish a connection with ON-CS (there are no free B-channels!), no matter what the software inside wants to do.

On the contrary as soon as OFG is open and so one of the two B-channels is used between FG-CS and ON-CS, it is physically impossible for the IN-CS to connect to FG-CS. Only one B-channel is free to call out, but FG-CS has no means to receive the call, because the second B-channel is in use. In this case the IN-CS would receive an “occupied” signal and again no kind of software bug can change this behavior.

In the SW we used “procmal”, “formail” (both in [6]) and some other tools to analyze the volume and content of the incoming and the outgoing mails. “Critical” outgoing mails are answered automatically to the sender, whereas “critical” incoming mail is analyzed by the (human!) mailmaster.

The classification “critical” in both cases has been set by the management of the bank together with us and the system administration. Mails are restricted by size, type of attachment (no programs!) and some other, finer rules.

4.3 Résumé

After a critical examination, the bank administrators called our security concept “charming”. It was accepted even by the users because of its simple and clear architecture. The inconvenience not to be able to surf in the Internet from the “own” workstation was allowed to be included in a bargain.

Some time ago we considered the flood-gates to be a “first approach” to security of a company, as long as other architectures are not installed or accepted. But now we see many interesting aspects and reasonable arguments so use them as part of the hole all-time security policy. We think that this project was a success especially by reason of the psychological advantages. Perhaps we can help to change the saying from “Build it first, secure it later” to “Secure it first, build it later”.

5 Related work

The idea to introduce different levels of security rings (“multilevel security”) is not new. Parker [27] describes them already in 1981, but there they were applied to the kernel of a single computer system. Gasser mentions the “security perimeter” in 1988 [13] as boundary of the systems “trusted” inner ring.

The *flood-gate principle* extends this thoughts to secure hole computer networks. Certainly we need different levels of security to provide the standard services, which are not believed to be very secure.

Also the approach to separate physically network components has a long history. Woodward [34] and Denning [5] use the term “Security Guard” (SG) as a (hardware) link between a “low” (untrusted) and a “high” (trusted) computer. The SG grants a “one way traffic” between two systems and a kind of “Human Review” is a central

part of this structure. The most important difference between SG and the FGP is the need of a special hardware on the SG side, whereas the FGP – as we saw in “4 A practical example” – does not need any unusual chip to provide a very clear hardware security.

Further on the psychological advantages of the FGP can not be reached by any “new kind of software or hardware” - which must be trusted first!

The main difference to a pure hardware solution as proposed in [25] lies in the flexibility of the flood-gates. Whereas a “black box”, which separates two networks from each other, is actually able to defend successfully attacks from the outside via standard services, a lot of other penetration possibilities (“beastware”) are not part of the hole concept. There is no configurable “demilitarized zone” (DMZ) between the two networks which are to be separated. The “black box” could be throughout thought of as a part of a flood-gate mechanism!

The term “Application-Level gateway” as mentioned in “3.1 Definitions” describes a class of firewalls, which are not all-purpose gateways for a network, but provide entrance to only some defined programs. From that standpoint the flood-gates are Application-Level gateways. But the term does not describe how to realize such a system. Moreover though it is clear what types of services are provided by the FGP, there are many possibilities - secure and insecure ones – to select the right applications to build a good Application-Level gateway.

The fact remains that the FGP provides the means to integrate perfectly a high security (and low cost) mechanism in an enterprises security policy, that could be used from the scratch.

6 Summary and view

We presented in this paper the *flood-gate principle* as a high secure, easy to understand and use, simple and modern way to protect networks of computer systems by low budget implementation possibilities.

In detail this mechanism has been compared to classical firewall solutions. The price to be paid for high security protection is the impossibility to provide “interactive” standard services. This leads us to a hybrid approach: we divide internal networks in two parts. The trusted part within the security perimeter is to be secured by flood-gates. The untrusted part may be protected by classical firewall methods depending on what services should be provided here. All data exchanging between the two parts must pass the flood-gates. A concrete implementation with some flood-gate realization tricks had been presented in this article.

The future will show if the FGP will spread all over enterprise security solutions or if it will stay a niche for very special requirements and starting policy concepts. Perhaps the use of cryptography can help to improve the FGP or at least cryptographic solutions has to be integrated into the hole concept of flood-gates.

We do not subscribe to the idea to use any kind of non-standard software or hardware components to realize the FGP. The psychological disadvantages can not be counterbalanced by efficiency considerations. But we will work at the famous rule “Security should not effect users who obey the rules”.

References

1. Greg Bossert et al.: Request for Comments: 2084, Considerations for Web Transaction Security, January 1997
2. Klaus Brunnstein: Beastware (Viren, Wuermer, trojanische Pferde) Paradigmen Systemischer Unsicherheit, Sichere Daten, sichere Kommunikation, Springer-Verlag, 1994, 44-60
3. William R. Cheswick, Steven M. Bellovin: Firewalls and Internet Security, Addison-Wesley, 5th printing April, 1995
4. F. Cohen: Computer Viruses: Theory and Experiments”, proceedings of the 7th National Computer Security Conference, Gaithersburg 1984, 240-263
5. Douglas E. Comer: Internetworking with TCP/IP: Principles, Protocols and Architecture, Vol. 1, Prentice-Hall, second edition, 1991
6. B. Costales, E. Allmann: sendmail, O'Reilley and Associates, 2nd edition, 1997
7. David A. Curry: UNIX System Security: A Guide for Users and System Administrators, Addison-Wesley, 1992
8. D. E. Denning: Cryptographic Checksums for Multilevel Database Security, Proceedings of the 1984 Symposium on Security and Privacy, Silver Spring 1984, 52-61
9. Department of Defense: DoD Trusted Computer System Evaluation Criteria, DOD Washington D.C, 1985 (the “orange book”)
10. Donald E. Eastlake: Request for Comments: 1455, Physical Link Security Type of Service, May 1994
11. M. Edwards: Security gets easier, cheaper, Communication News, November 1997, 82-83
12. Detlef Garbe: Datenschutz im ISDN, Sichere Daten, sichere Kommunikation, Springer-Verlag, 1994, 167-208
13. Morrie Gasser: Building a secure Computer System, Van Nostrand Reinhold, 1988
14. P. Gulbins, UNIX Version 7, bis System V.3, Springer-Verlag, 1988
15. Joachim Jacob: Nationale und internationale Datenschutzgesetzgebung, Sichere Daten, sichere Kommunikation, Springer-Verlag, 1994, 61-70
16. P. A. Karger: Limiting the Potential Damage of Discretionary Trojan Horses, Proceedings of the 1987 Symposium on Security and Privacy, IEEE Computer Society, 1987, 32-37
17. C. E. Landwehr: The Best Available Technologies for Computer Security, Advances in Computer Security, vol. 2, Artech House, 1984, 108-122
18. M. Laubach: Request for Comments: 1577, Classical IP and ARP over ATM, 1994
19. G. Lennox: Computer Security and Industrial Cryptography, State of the Art and Evolution, Lecture Notes in Computer Science 741, Springer-Verlag, 1993, 235-243
20. Linux, Kernel 2.0.33, for instance S.u.S.E. Linux 5.2, March 1998
21. N. Luckhardt: Kryptokampagne, Heisse Verlag, c't 6/98, 1998, 32-33
22. Ch. Meinel: Wie funktioniert das Internet?, ITWM-Preprint 97-01, 1997
23. J. Mogul: Simple and Flexible Datagram Access Controls for Unix-based Gateways, Digital Equipment Corporation, March 1981
24. S. Mueller, T. Engel, Ch. Meinel: Das Internet – Neues Medium für kommerzielle Aktivitaeten, ITWM-Trier, 1997
25. Newspaper article from the “Luxembourger Wort”: Sicherheit auf dem Web “Made in Luxembourg”, 31st January 1997
26. I. Pakhomenko, E. Pless: Sicherheitsprobleme in IP-über-ATM-Netzen, iX 3/98, 1998, 118-121
27. T. A. Parker: A Secure European System for Applications in a Multivendor Environment (The SESAME Project), Proceedings of the 14th American National Security Conference 1991
28. Heribert Peuckert: Datenschutz und Datensicherheit aus technischer Sicht, Sichere Daten, sichere Kommunikation, Springer-Verlag, 1994, 13-26

- 29.J. Postel: Internet Protocol – DARPA Internet Program Protocol Specification, DARPA, September 1981
- 30.Kare Presttun: Security in Distributed Data Systems, Secure Information, Secure Communication, Springer-Verlag, 1994, 251-259
- 31.M. J. Ranum: Network security: safety is next, Data Communication, 21.10.97, S. 128-132
- 32.Tinnefeld, Ehmann: Einfuehrung in das Datenschutzrecht, Oldenbourg, 1992, 253-257
- 33.T. E. Weber: Should only the paranoid get E-mail protection?, Wall Street Journal, September 1997
- 34.J. P. L. Woodward: Applications for Multilevel Secure Operating Systems, proceedings of the NCC 48, 1979, 319-328
- 35.World Wide Web Consortium, Hypertext Transfer Protocol, <http://www.w3c.org/pub/WWW/Protocols>
- 36.G. Paul Ziemba et al.: Request for Comments: 1858, Security Considerations – IP Fragment Filtering, October 1996