Ernst-Georg Haffner, Uwe Roth, Andreas Heuer, Thomas Engel, Christoph Meinel:
Managing Distributed Personal Firewalls with Smart Data Servers
in Proceedings of World Conference on the WWW and Internet (WebNet 2001), AACE Press, Orlando, Florida, USA,
pp. 466-471, 10, 2001. ISBN: 1-880094-46-0.

# Managing Distributed Personal Firewalls with Smart Data Servers

Ernst-Georg Haffner, Uwe Roth, Andreas Heuer, Thomas Engel, Christoph Meinel
Institute of Telematics
Trier
Germany
{haffner, roth, heuer, engel, meinel}@ti.fhg.de

**Abstract**: Modern security architectures tend to become more and more complex. Not only the chances to improve Web applications using several data channels and diverse (TCP-)ports are very promising, but also the risks for criminal attacks and an intrusion into the corporate network are increasing.
The classical solution to protect networks against criminal attacks with firewalls is problematic, though. On the one hand, attacks from the inside are hardly prevented by firewalls, on the other, mobile computing poses additional security risks to the corporate networks. Personal firewalls solve some of those problems, but their central administration is very difficult. In this paper, we will discuss a possible strategy to manage distributed personal firewalls with a central tool, the Smart Data Server.

## Introduction

Today's security architectures tend to become more and more complex. There are two main reasons for this. One the one hand, modern programs and applications require an increasing amount of data exchange between networks and the information flow often goes via Internet. On the other hand, potential attackers use sophisticated tools and possess extended knowledge to compromise computer systems via online connections.

The manifold possibilities to fight against these attacks, mostly with one or more levels of firewall complexes, result in very complicated security architectures at least for greater companies. In general, firewalls operate as packet filters or application level gateways. They analyze the network traffic and allow or disallow the transfer of data on base of certain rules.

Unfortunately, most attacks against an internal network do not come from the outside, but from the inside. For this kind of attacks, firewalls are mostly useless. Additionally, mobile computing, where personal laptops are plugged into both, the corporate network and (at home) into the network of a public Internet Service Provider (ISP), poses a serious risk to the integrity of the former one: viruses, worms and all other kinds of beastware may arrive at the inner network of a company (Cohen, 1984) and (Karger, 1987).

One possible answer to the question of how to secure a corporate network in such a situation is the personal firewall. If the protection of a computer does not depend on other machines, servers or programs within the network, the mobility of the system represents no security problem any longer. Likewise, internal network attacks are secured at the last possible station: at the workstation of the employees.

Unhappily, securing a network only by personal firewalls is hardly manageable. Ensuring that all personal firewalls of a company are configured according to the security policy is very difficult. In this contribution, we will focus on solving the problem of managing what we call "Distributed Personal Firewalls" (DPF) on base of a central managing tool: the Smart Data Server (SDS).

## Personal Firewalls

Before we discuss special aspects of personal firewall implementations, we will have a closer look at classical firewalls first.

Cheswick and Bellovin define the expression "firewall" in their famous book "Firewalls and Internet Security" (Cheswick & Bellovin, 1995) 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."

Many of the existing firewalls are mainly packet filters. They analyze the source and target IP-address of each data packet, check its TCP port number and reject the packet if one of them is not allowed to pass. 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 works like a proxy server. Details can be found in (Cheswick & Bellovin, 1995). The latter describes a scheme for "special-purpose" gateways that allows only some applications to pass through.

Personal firewalls are programs that run on front end and client machines rather than on corporate network servers. A general description of personal firewalls and a discussion of some classical examples can be found in (Zych, 2000). In principle, personal firewalls have to manage the same security tasks as classical firewalls. Nevertheless, there are some very important advantages:

- Personal firewalls work even though the communication might be encrypted due to their location on one end of the information flow.
- They are mobile inasmuch as the machine they are running on (for instance a laptop) is mobile, too. Mobile computing requires special personal solutions due to foreign ISP's and several other non-controllable factors.
- They can take care more precisely and specifically of individual security requirements.
- The security level can be easily adjusted (mostly by the user himself/herself).
- Personal firewalls can help to solve the problem of several entry points into networks whereas centralized firewall systems can hardly manage more than very few entry points.
- The penetration of one personal firewall does not directly imply possible damage for the complete network (due to several other personal firewalls on the other workstations).
- Higher degrees of bandwidth, increasing line speeds and more complex protocols make it very difficult to secure a corporate network by a centralized firewall system.

The main disadvantages of personal firewall solutions, especially for enterprise use, are the following ones:

- In general, the user is responsible to maintain the personal firewall (even though system administrators may help in some cases and with the initial setup).
- Personal firewalls pose security risks for the corporate network due to possible false-configuration (e.g. lack of knowledge or intentional act) and a central maintenance of these systems is very complicated.
- Usually, there is no 7/24 support and supervision of personal firewalls.
- Extensive intrusion detection is very complicated because there is no central administration point.
- Administration rights of personal firewalls sometimes belong to the user of the workstation and not the company's system administrator (especially for older Windows based systems).

Nevertheless, personal firewalls or at least decentralized systems are the upcoming standard to secure complex corporate networks.

In the following, we present a possibility to overcome some of the disadvantages of personal firewalls by proposing a distributed personal firewall architecture that could be managed from a central point by the middleware of the Smart Data Server (SDS).

**The Smart Data Server**

The Smart Data Server (SDS) as a general framework for distributed functionality is a promising platform to provide also advanced security needs (Roth et. al., 1999a). The SDS is able to connect different data sources and to improve information exchange. The system serves as middle tier in a three-tier architecture (Roth et. al. 1999b). It can work together with several C/S-components and -structures and is a pure Java implementation. The overall idea is to put intelligence on the server side to increase the efficiency of the communication with the clients. The information channels are scalable towards the security requirements.

Especially, strongly encrypted communication between server and client is possible.

Similar approaches can be found in Satoshi Hirona's HORB, an object-oriented request-broker (HORB). The Swift Company works on a product using RMI of Java (Swift). The idea of the "Common Request Broker Architecture" (CORBA) is also related to the SDS in our context (Object Management Group). In this approach, several resources can be distributed over the network. DCOM (Brown, 1997) and Java's RMI (Sun Microsystems, 1998) are also similar to the basic ideas of the SDS.

As already stated, the SDS Java server provides several information channels with scalable security levels. The internal structure of the SDS is with three layers very strict. The layers are illustrated by figure 1.

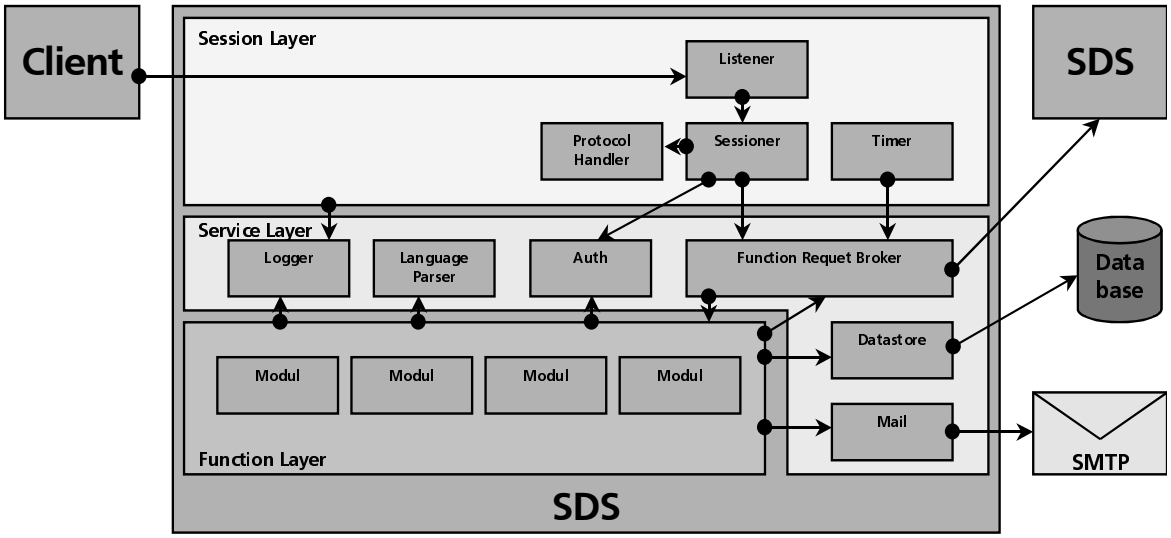


**Figure 1**: The internal layer structure of the SDS

The session layer is responsible for handling client requests, checking authorization or creating time-based functions by itself. It contains the basic functionality for network-connections, session handling, protocol analyzing and other request-related functionality.

The service layer consists of a set of general usable services that is engaged by both, the function layer and the session layer modules, e.g. a data store module and the central function request broker.

Function layer modules realize the application functionality. In the context of this contribution, the security policy has to be expressed as function layer module so that the rules for the personal firewalls can be distributed to the workstations of each single user.

One special feature of the SDS can help to manage the difficulties arising from distributed security: the SDS can simply be cloned and several instances of the server communicate pretty well with each other. Additionally, functions can be distributed between those clones while they still work on the same databases. Therefore, a clone of the SDS can be placed within the demilitarized zone (DMZ) while others are placed within the inner network to arrange security and trust management tasks. Certainly, it is necessary to design the security architecture for those requirements very carefully. We will have a closer look at this point in the subsequent sections.

**Managing Distributed Personal Firewalls**

As we have seen already, a firewall nowadays becomes a possible congestion point due to complex protocols and increasing network throughput (Ioannidis et. al., 2000). Distributed personal firewalls that are managed from a central server can help to map corporate security policies to the configuration of workstation firewall systems.

In the following, we will provide a possible methodology to overcome the problem of distributing security policy information to decentralized distributed personal firewalls by the use of the Smart Data Server architecture. Especially, firewall rule sets have to be transferred via secured and thus encrypted channels to the client machines. Therefore, every client machine must serve as a client for the SDS communication.

Figure 2 and figure 3 demonstrate the structural and logical task of the managed personal firewalls. While the former one shows a typical firewall architecture (FW) to protect a complex network against attacks from the outside with routers (R), virus scanners (VS), and mail analyzing tools (MA), the latter one illustrates the use of an SDS as central management server to configure distributed personal firewalls.

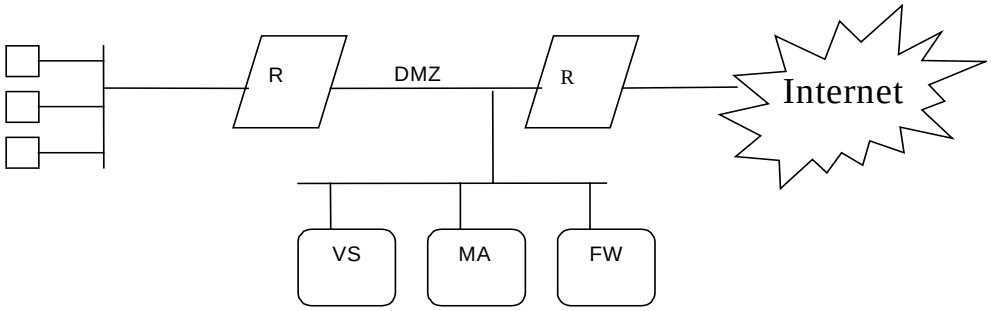


**Figure 2**: The classical Firewall architecture without use of distributed personal firewalls

The management target is to administer not only personal firewalls on every relevant workstation but also the main (classical) firewalls that run on hosts and analyze the main traffic stream from inside out and vice versa.

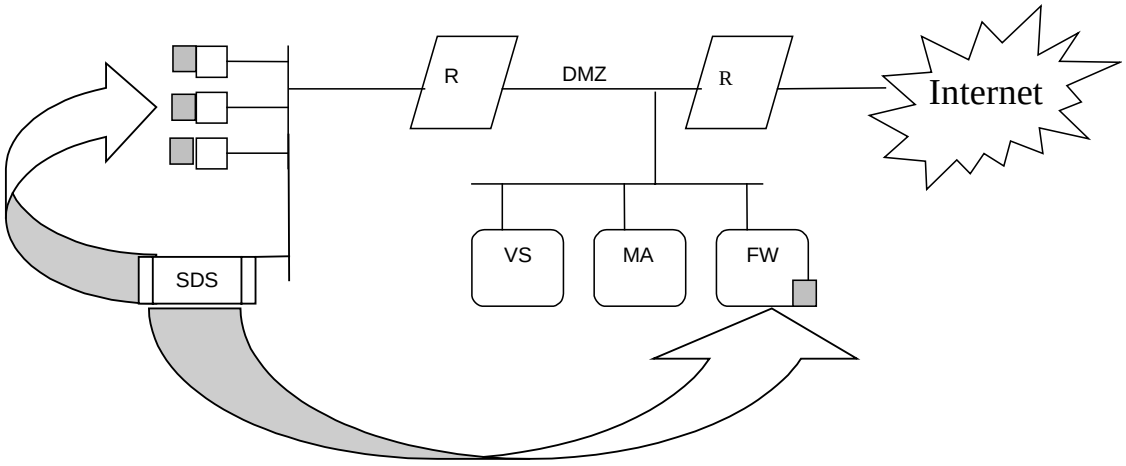


**Figure 3**: Distributed personal firewall architecture managed by an SDS

As we have seen before, a critical point in the described methodology is the vulnerability of the SDS itself. A worst-case scenario would be an attacker (from the inside) compromising the central managing server itself and thus possibly jeopardize the integrity of the whole corporate network.

Therefore, it is advisable to – at least – "harden" the operating system platform the server is running on. Several companies offer such trusted operating systems on base of Windows-NT or UNIX systems (e.g. Argus Systems Group, Computer Associates International, Hewlett-Packard Company).

The process of distributing the security policy can then be based on concepts like the *PolicyMaker* as a general approach for trust management. It provides a formal model of trust management and a framework for the development of decentralized security features. A corporate security policy can thus be distributed from the central firewall complex to the single end points of personal firewalls. Further details of the PolicyMaker approach can be found in (Blaze et. al. 1999).

## Summary and Outlook

Modern security architectures mostly operate with a central firewall solution. All traffic from the internal corporate network to the Internet and vice versa has to pass through the firewalls. Not only an increasing bandwidth cause problems to that solution, but also encrypted data transfer protocols and manifold security holes arising from mobile computing pushes other methodologies. Additionally, central security tools may cause bottlenecks and potential vulnerabilities for the whole network communication. Even though personal firewalls solve some of these difficulties, they lead to a set of additional problems, mainly administrative ones.

In this contribution, we wanted to point out the idea of distributed personal firewalls, where a central management is able to control the diverse firewalls on the end-user client machines. One possibility would be the Smart Data Server as administrating middleware that has been briefly presented. With scalable secure information channels it can serve as a central management tool for corporate and end-user personal firewalls. Then, a system like the PolicyMaker can help to administer the security tasks for the distributed firewall systems. It is important to see that the central management of distributed personal firewalls also offers new working surfaces for attackers and therefore has to be designed very carefully.

The next steps to go are the verification of the theoretical construction in practice and the evaluation of the security aspects. After this, also questions of performance and the overall efficiency of the approach have to be examined.

## References


Blaze, M. & Feigenbaum, J. & Ioannidis, J. & Keromytis, A. (1999). *The Role of Trust Management in Distributed Systems Security*. Secure Internet Programming: Issues in Distributed and Mobile Object Systems, 1603.

Blaze, M. & Feigenbaum, J. & Lacy, J. (1996). *Decentralized Trust Management*. Proceedings IEEE Conference on Security and Privacy, Oakland, CA.

Bossert, G. et al. (1997). *Considerations for Web Transaction Security*, Request for Comments: 2084, January 1997

Brown, N. (1997). *Distributed Component Object Model Protocol -- DCOM/1.0*, Microsoft Corporation http://msdn.microsoft.com/library/specs/

Cheswick, W. & Bellovin, S. (1995). *Firewalls and Internet Security*, Addison-Wesley, 5th printing April, 1995

Cohen, F. (1984). *Computer Viruses: Theory and Experiments"*, proceedings of the 7th National Computer Security Conference, Gaithersburg, 240-263

Comer, D. (1991). *Internetworking with TCP/IP: Principles, Protocols and Architecture*, Vol. 1, Prentice-Hall, second edition

Curry, D. (1992). *UNIX System Security: A Guide for Users and System Administrators*, Addison-Wesley

Denning, D. (1984). *Cryptographic Checksums for Multilevel Database Security*, Proceedings of the 1984 Symposium on Security and Privacy, Silver Spring 1984, 52-61

Eastlake, D. (1994). *Request for Comments: 1455, Physical Link Security Type of Service*, May 1994

Edwards, M. (1997). *Security gets easier, cheaper*. Communication News, November 1997, 82-83

HORB Open source code project. http://www.horb.org

Inc.: Argus Systems Group. http://www.argussystems.com

Inc.: Computer Associates International.
http://www.cai.com/solutions/enterprise/etrust/access_control/index.htm

Inc.: Hewlett-Packard Company. http://www.hp.com/security/products/virtualvault

Inc.: Sun Microsystems (1998). *Java Remote Method Invocation Specification*, Revision 1.50, JDK 1.2, Sun Microsystems ftp://ftp.javasoft.com/docs/jdk1.2/rmi-spec-JDK1.2.pdf

Inc.: Swift Company, Belgium. http://www.swift.com

Ioannidis, S. & Keromytis, A. & Bellovin, S. & Smith, J. (2000). *Implementing a distributed firewall*. Proceedings of the 7th ACM conference on Computer and communications security. 190–199

Karger, P. (1987). *Limiting the Potential Damage of Discretionary Trojan Horses*, Proceedings of the 1987 Symposium on Security and Privacy, IEEE Computer Society, 32-37

Lennox, G. (1993). *Computer Security and Industrial Cryptography*, *State of the Art and Evolution*, Lecture Notes in Computer Science 741, Springer-Verlag, 235-243

Object Management Group, (2001). *CORBA.* http://www.omg.org / http://www.corba.org

Roth, U. & Haffner, E.-G. & Engel, T. & Meinel, Ch. (1999a). *An Approach to Distributed Functionality: The Smart Data Server (SDS)*, Proceedings of the WebNet International Conference 1999

Roth, U. & Haffner, E.-G. & Engel, T. & Meinel, Ch. (1999b). *The Smart Data Server - A New Kind of Middle Tier*, Proceedings of the IASTED International Conference Internet and Multimedia Systems and Applications (IMSA '99), 1999

Zych, T. (2000). *Personal Firewalls: What are they, how do they work?* SANS Institute. 2000.
http://www.sans.org/infosecFAQ/homeoffice/personal_fw.htm