A Pattern Language for Firewalls

Eduardo B. Fernandez, Maria M. Larrondo-Petrie, Naeem Seliya, Nelly Delessy, and Angela Herzberg

Dept. of Computer Science and Eng.
Florida Atlantic University
Boca Raton, FL 33431
{ed, maria, nseliya, ndelessy, azepeda @cse.fau.edu}

Introduction

Computer information security is a growing need for organizations and individuals; therefore, computer systems must be protected against attacks [Fer01b]. In the case of computers connected in a local network, attacks may come from external networks or from other local sub-networks. A common solution to protect local networks is to incorporate a firewall to serve as a network gateway [Zwi00].

Firewalls have been shown to be very effective in providing security by basically creating a choke point of entry (and exit) into a local network [Bar99]. A firewall therefore restricts unauthorized users from access to the local network and local networks from accessing external sites that are considered untrustworthy. A firewall can be used as a mechanism to enforce security policies and decisions, and also allows a limited exposure of the protected network to outsiders. Simply stated, a firewall allows access to approved traffic and denies access to traffic identified as unauthenticated or unauthorized.

We are developing a pattern language to describe firewall functions. Figure 1 shows the patterns in our pattern language and their relationships and dependencies. The Address Filter Firewall defines a basic filtering function based on network addresses. A Proxy-based firewall is used at the application layer to control access to applications. Both the basic and the proxy firewalls can be complemented with stateful filtering. A content-based firewall considers filtering based on document content. In this paper we only present in detail the Address Filter and Proxy Firewall patterns.

Basically, a firewall is a unit or group of units that enforces an access control policy among networks. Progress in firewall-based network protection and security policy enforcement has mainly focused on building components that are suited to specific networks, operating systems, and computers [Eps99, Hen01]. However, the basic underlying architectures of the various system-specific firewalls are very similar.

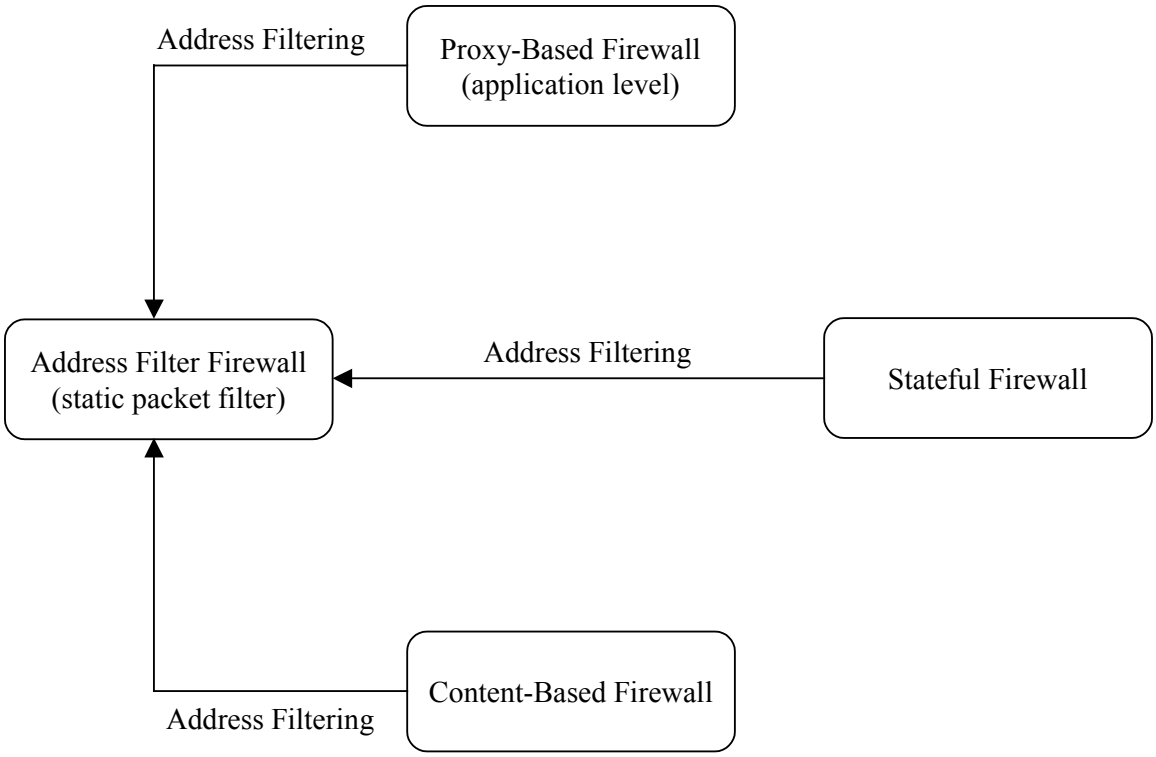

Figure 1: Firewall Pattern Language

Address Filter Firewall

Intent
To filter incoming and outgoing network traffic in a computer system based on network addresses.

Context
Computer systems on a local network connected to the Internet and to external networks.

Problem
A local network is usually attacked from the outside (external network). The local network may be partitioned and attacks may also come from other local networks.

Forces
- We need to filter input and output traffic from a computer system in a user-transparent form.
- Network administrators deploy and configure a variety of firewalls; hence it is important to have a clear model of what is being filtered and how it is filtered.
- The configuration of the firewalls must reflect the institution's security policies; otherwise, it would be difficult to decide on what to filter.
- What is being filtered is constantly changing; hence it should be easy to make changes to the configuration of the firewall.
- Rules specify what types of traffic are to be allowed, blocked, or discarded. Otherwise, it would be hard to realize specific policies.
- It may be necessary to log client requests for auditing and defense purposes.

Solution
The Client can only access the Local Network if a rule exists authorizing traffic from its address. Therefore, each association link between the Client and Local Network is controlled by a Rule. The Firewall consists of a set of access rules defined for the institution (of local network) according to its policies. A Local Network can have one or more Firewalls. If a particular request is not satisfied by any of the Explicit Rules, then the Default Rule is applied.

Dynamics
We describe the dynamic aspects of the Basic Firewall Pattern using sequence diagrams that correspond to its two basic use cases.

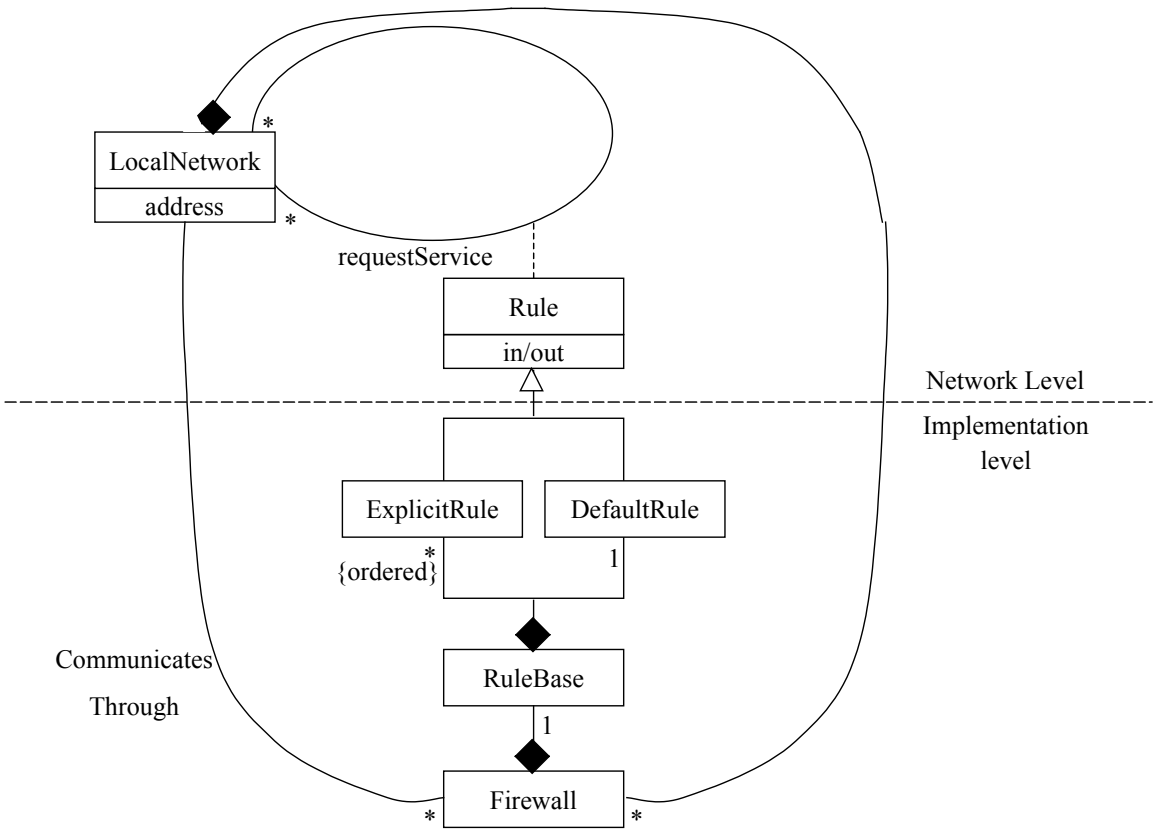

Figure 2: Class Diagram for Basic Firewall Pattern

Filtering a Client's Request

- Summary: A remote network wants access to the local network to either transfer or retrieve information. The access request is made through the firewall, which according to its set of rules determines whether to accept or deny the request, i.e., it filters the access request. Figure 3 illustrates this use case.
- Actors: External client.
- Precondition: An existing set of rules to filter the request must be in place in the firewall.
- Description:
    a. An external network requests access to the local network.
    b. A firewall filters the request according to a set of rules. If none of the rules in the rule set are satisfied then a default rule is used to filter the request.
    c. If the request is accepted, the firewall allows access to the local network.

- Alternate Flow: If the request is denied, the firewall rejects the access request by the external network to the local network.
- Postcondition: The firewall has accepted the access of a trustworthy client to the local network.

Defining a new rule

- Summary: The administrator of the firewall adds a new rule to the set of rules. The firewall checks whether the new rule to be added does not already exist in the rule set. Figure 4 illustrates this use case.
- Actors: Administrator .
- Precondition: The administrator must have authorization to add rules.
- Description:
    a. The administrator initiates the adding of a new rule.

    b. If the rule does not already exist in the rule set then it is added.
    c. The firewall acknowledges the addition of the new rule.
- Alternate Flow: The rule is not added because it already exists in the rule set.
- Postcondition: A new rule is added to the rule set of the firewall.

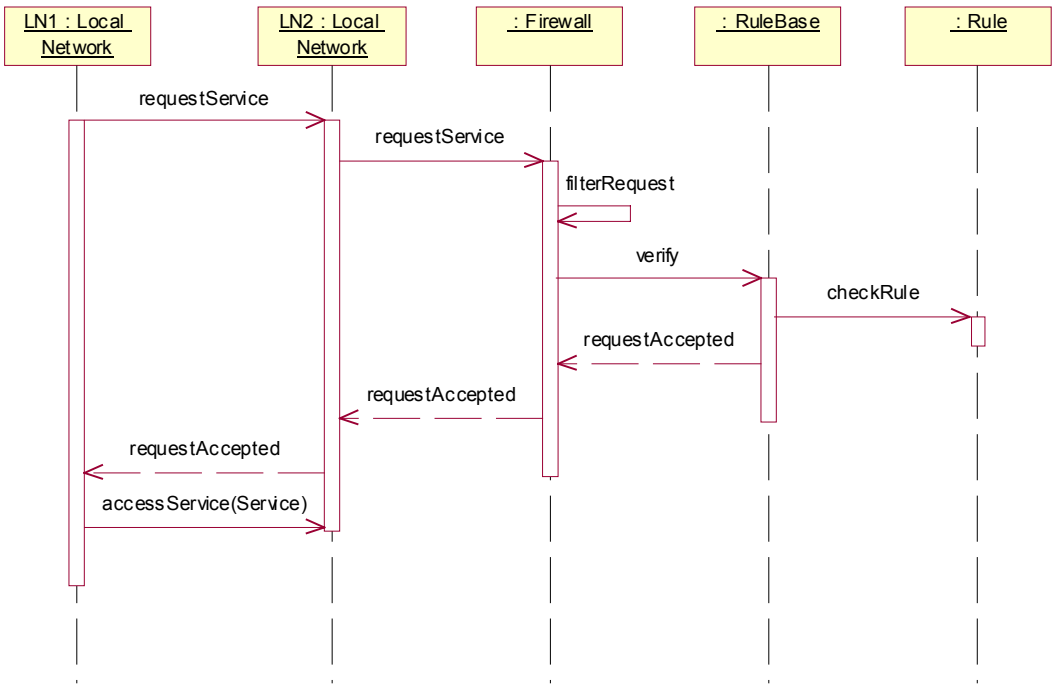

Figure 3: Sequence Diagram for Filtering a Client's Request

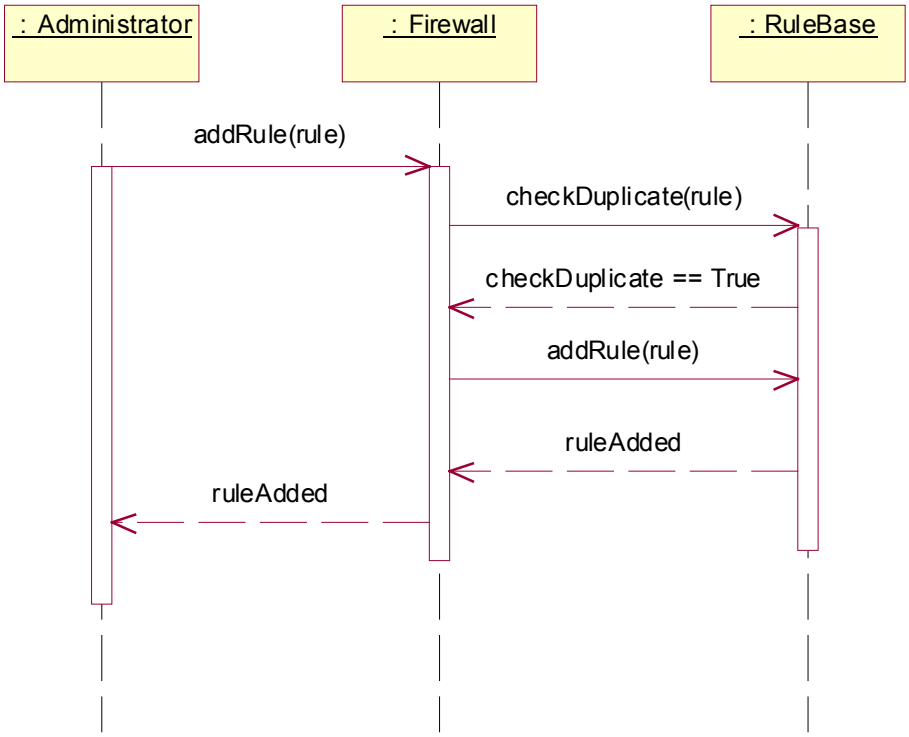

Figure 4: Sequence diagram for defining a new rule

Consequences

The Basic Firewall Pattern has the following advantages:

- A firewall filters all the traffic that passes through it, based on network addresses and transparently to applications.
- It is possible to express the institution filtering policies through its firewall rules.
- A firewall facilitates the detection of possible attacks and to hold regular users responsible of their actions [Sch03].
- A firewall lends to a systematic logging of incoming and outgoing messages.
- Low cost, it is included as part of many operating systems.
- Good performance. It only needs to look at packet headers.

The Basic Firewall Pattern has the following (possible) liabilities:

- A firewall's effectiveness may be limited due to its rule set (order of precedence).

- A firewall's effectiveness is limited to the point of entry into the local network, and once a potential attacker has passed through the firewall the security of the system may be breached.
- A firewall can only enforce security policies on traffic that goes through the firewall.
- A (basic) firewall cannot stop higher level attacks.
- A firewall generally tends to adversely affect the usability, performance, and cost of the protected system [Sch03].
- The security policies that a firewall enforces are different for different sites, networks, and systems. Addition of new rules may interfere with existing rules in the rule set; hence, a careful approach should be taken in adding and updating access rules.
- Not state aware.
- A packet filter cannot recognize forged addresses because it only examines the header of the IP packet.
- A hacker could put malicious commands or data in headers not used for routing and in the payload.

Known Uses

This model corresponds to a basic (packet filtering) firewall architecture that is seen in commercial firewall products, such as: ARGuE (Advanced Research Guard for Experimentation), which is based on Network Associates' Gauntlet Firewall [Eps99]; OpenBSD Packet Filtering Firewall [Rus02], which models the basic firewall architecture for the Berkeley Software Distribution system; and, Linux Firewalls [Zie02], which models the basic firewall architecture with the Linux operating system. The basic firewall model is used as an underlying architecture for other types of firewalls that include more advanced features, for example: CyberGuard [Hen01], which primarily uses a stateful inspection firewall architecture.

Related Patterns:

The Authorization pattern [Fer01a] defines the security model for the Basic Firewall Pattern. The Role-Based Access Control pattern, a specialization of the authorization pattern, is applicable if the networks and their access rules are respectively defined in terms of roles and rights [Fer01a]. The Firewall pattern is also a special case of the Single-Point-of-Access [Yod97]. Another approach to firewall patterns is presented in [Sch03].

Application Proxy Firewall

Intent
Inspect (and filter) incoming and outgoing network traffic based on the type of application they are accessing.

Context
Computer systems on a local network connected the Internet and to external networks where a higher level of security than the one provided by packet filters is needed.

Problem
The Address Filtering Firewall only inspects the network addresses for deciding access for a message. However, potential attacks may be embedded within the data segment of the packets whose network addresses are granted access to the local network [Sch03]. In addition, an Address Filtering Firewall does not provide security against IP spoofing. A virtual separation of the local network from the external client networks is needed to allow a complete inspection of the network traffic.

Forces
- The forces of the Address Filter are still valid.
- Network administrators deploy and configure a variety of firewalls; hence it is important to have a clear model of what is being inspected and filtered, and how the application proxies are implemented.
- The configuration of the firewalls must reflect the institution's security policies; otherwise, it would be difficult to decide what to inspect and modify in the application data.
- What is being inspected and filtered is constantly changing; hence it should be easy to make changes to the configuration of the firewalls. If a request is made for a proxy service that is not supported by the firewall, that request should be blocked.
- It may be necessary to log client requests for auditing and defense purposes.

Solution
The client only interacts with a proxy of the service requested, which in turn communicates with the protected service. The protected service only communicates with the Application Proxy Firewall. The client can only receive service from the server if an application proxy exists for the requested service. Each application proxy has its own pre-defined (stated by the administrator) access and modification rules that are used to inspect, change, and filter the incoming (or outgoing) messages.

Figure 5 shows the class diagram for this pattern. This is an extension of Figure 2, including now separate services in each local network and application proxies to filter requests for services.

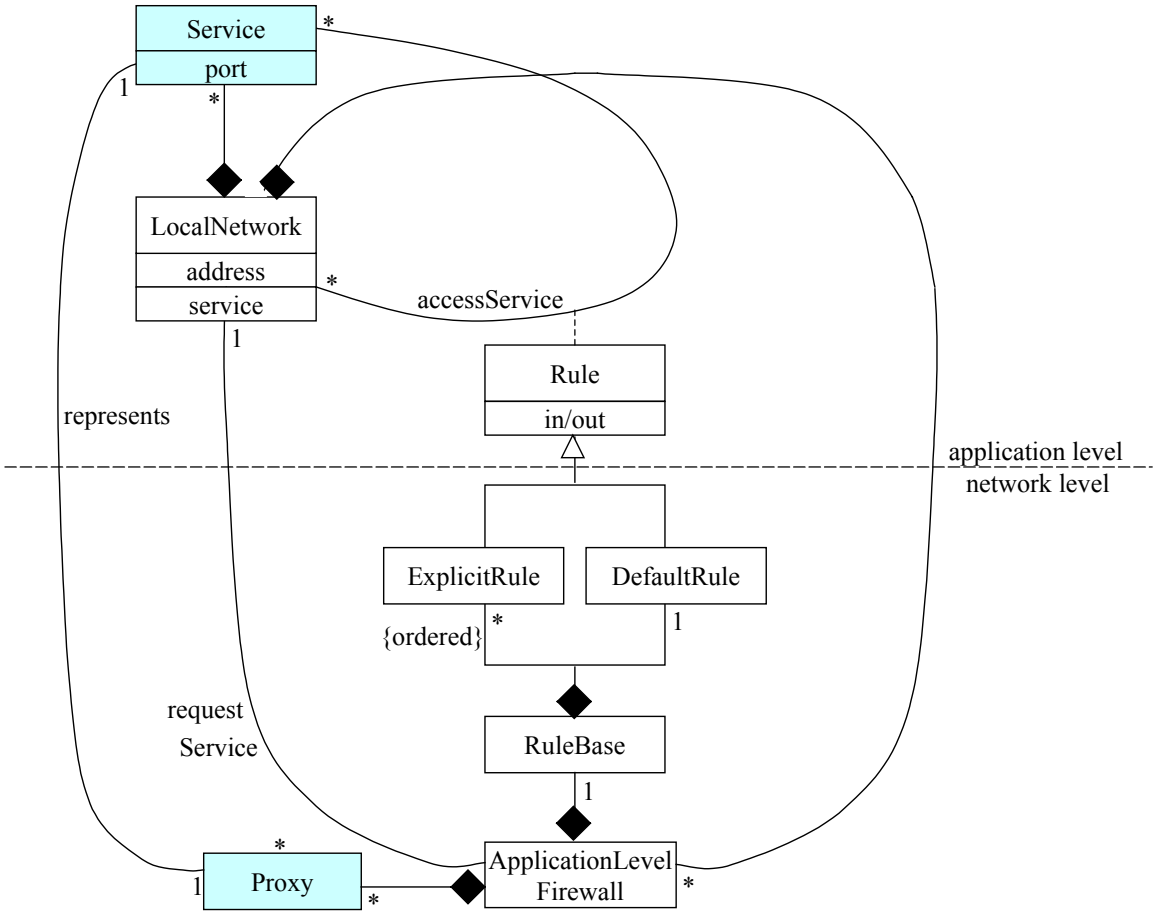

Figure 5: Class Diagram for Application Proxy Firewall Pattern

Dynamics
We show a use case for filtering requests for services.

Use Case for Providing Service to a Client

- Summary: An external client wants access to a service from the local server. The access request is made through the firewall, which according to its application proxies and their set of rules determines whether to deny or accept (with or without modification of the data content) the request. Figure 6 illustrates this use case.
- Actors: External client.
- Precondition: None
- Description:

  An external network requests service access to the Application Proxy Firewall.

The firewall filters the request according to its application proxies and their access/modification rules. If none of the rules in the rule set are satisfied then a default rule is used to filter the request.
If the request is accepted with or without modification, the client is allowed to access the service through the application proxy.

- Alternate Flow: If the service request is not supported by the Application Proxy Firewall, or the firewall considers the client untrustworthy then the firewall will not grant the service request.
- Postcondition: The firewall has accepted the service request from a trustworthy client to the local network.

Consequences
The Application Proxy Firewall Pattern has the following advantages:

- The firewall inspects, modifies (if needed), and filters all access requests based on predefined application proxies that are transparent to the client
- It is possible to express the institution's filtering policies through its application proxies and their rules.
- It is possible to modify certain portions of the information in cases where suspicious commands are included in/or the data segment of packets.
- A firewall facilitates the detection of possible attacks and helps hold regular users responsible of their actions [Sch03].
- It protects against possible implementation faults in the protocol stacks of the internal systems [Sch03]. The IP (Internet protocol) address of the internal network is always hidden to the external networks.
- A firewall lends to a systematic logging and tracking of all service requests going through it.
- Provides a high level of security because it inspects the complete packet including the headers and data segments.

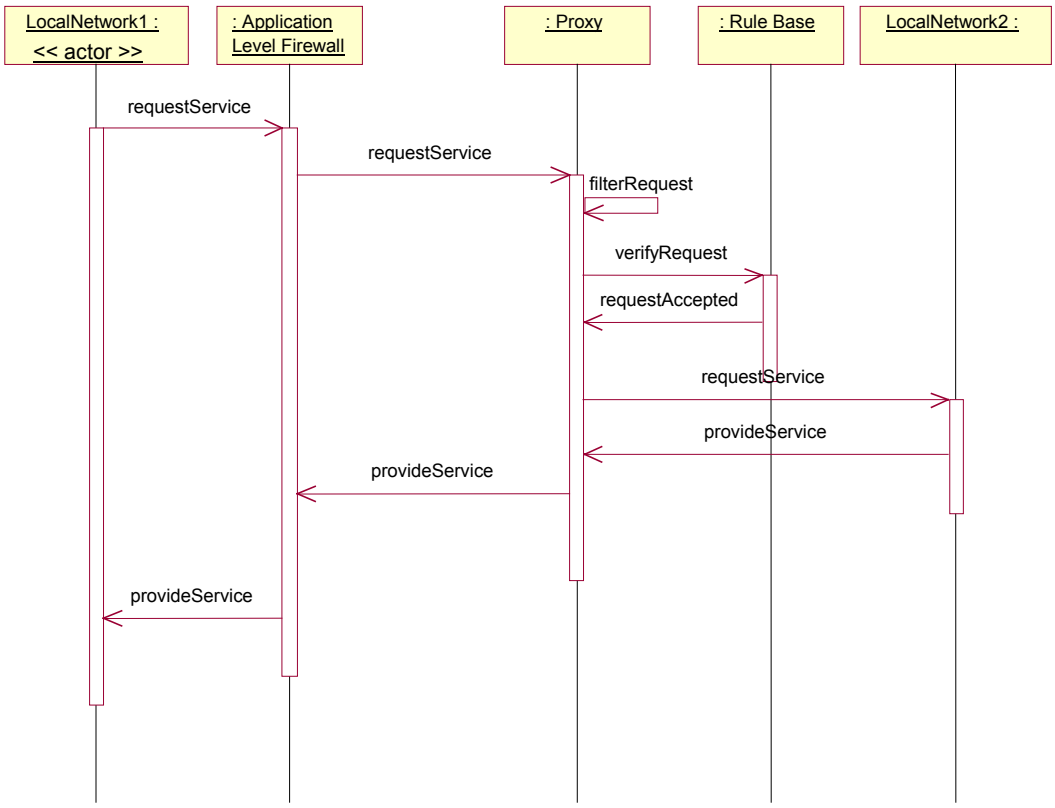

Figure 6. Sequence diagram for filtering service requests.

The Application Proxy Firewall Pattern has the following liabilities:

- Possible implementation costs due to the need for specialized proxies. On the other hand, proxies already exist for common services.
- Low speed due to the application proxy overhead and the inspection of the data segment of packets.
- Increased complexity of the firewall. Application Proxy Firewalls may require change in applications and/or the user's interaction with the system.
- The security policies that a firewall enforces are different for different sites, networks, and systems. Addition of new rules for a given application proxy may interfere with existing rules in the rule set; hence, a careful approach should be taken in adding and updating access rules.
- Not state aware.

Known Uses

An application proxy firewall uses the basic address filtering firewall model that is used in almost all firewall products, such as ARGuE Guard [Eps99]. Some specific firewall

products that use application proxies are Pipex Security Firewalls [Pip03] and InterGate Firewall.

Related Patterns:
The Address Filtering Firewall is the basis for the Application Proxy Firewall model.

References

[Bar99] Y. Bartal, A. Mayer, K. Nissim, and A. Wool. FIRMATO: A Novel Firewall Management Toolkit. In *Proceedings of the 20th IEEE Symposium on Security and Privacy*, pp 17-31, Oakland, California, April 1999.

[Che03] W. Cheswick, S.M.Bellovin, A.D.Rubin, *Firewalls and Internet security,* 2nd Ed., Addison-Wesley, 2003.

[Eps99] J. Epstein. Architecture and Concepts of the ARGuE Guard. In *Proceedings of the 15th Annual Computer Security Applications Conference*, pages 45-54, Phoenix, Arizona, December 1999.

[Fer01a] E. B. Fernandez and R. Pan. A Pattern Language for Security Models. In *Proceedings of the PLoP Conference*, 2001. http://jerry.cs.uiuc.edu/~plop/plop2001/accepted_submissions/accepted-papers.html.

[Fer01b] E. B. Fernandez. An Overview of Internet Security. In *Proceedings of the World's Internet and Electronic Cities Conference (WIECC 2001)*, Kish Island, Iran, May 2001.

[Gat00] B. Gatliff. Web by Proxy. *Embedded Systems Programming Magazine*. May 2000. http://www.embedded.com/internet/0005/0005ia1.htm

[Hay00] V. Hays, M. Loutrel, and E.B.Fernandez, "The Object Filter and Access Control framework", *Procs. Pattern Languages of Programs (PLoP2000) Conference*, http://jerry.cs.uiuc.edu/~plop/plop2k

[Hen01] P. Henry. An Examination of Firewall Architectures. CyberGuard Corporation White Paper, April 2001. http://www.cyberguard.com.

[Nou00] N. A. Noureldien and I. M. Osman. A Stateful Inspection Module Architecture. In *Proceedings of TENCON 2000*, vol. 2, pp 259-265, Kuala Lumpur, Malaysia, September 2000.

[Pip03] Pipex Firewall Solutions. http://www.security.pipex.net/about.shtml.

[Rus02] R. E. Rustad, Jr. Guide to OpenBSD Packet Filtering Firewalls. Kuro5hin Article, Nov. 2002, http://www.kuro5hin.org/story/2002/11/23/14927/477.

[Sch03] M. Schumacher. Firewall Patterns. In *Proceedings of the Euro PLoP Conference*, 2003.

[Vddd] InterGate Firewalls from Vicomsoft. http://www.vicomsoft.com/

[Wal01] B. Walder. Firewalls. *Computer Weekly Online*, 2001. http://www.nss.co.uk/Articles/firewalls01.htm

[Yod97] J. Yoder and J. Barcalow, "Architectural patterns for enabling application security". *Procs. PLOP'97*, http://jerry.cs.uiuc.edu/~plop/plop97 Also Chapter 15 in *Pattern Languages of Program Design*, vol. 4 (N. Harrison, B. Foote, and H. Rohnert, Eds.), Addison-Wesley, 2000.

[Zie02] R. Ziegler and C. Constantine. Linux Firewalls: Packet Filtering. *News Riders*, March 2002. http://informit.com.

[Zwi00] E. D. Zwicky, S. Cooper, and B. Chapman. Building Internet Firewalls.O'Reilly and Associates, 2nd edition, 2000.