# Patterns for cloud firewalls

Eduardo B. Fernandez[1], Nobukazu Yoshioka[2], and Hironori Washizaki[3]

[1]Dept. of Computer Science and Engineering, Florida Atlantic University, USA, ed@cse.fau.edu
[2]GRACE Center, National Institute of Informatics, Tokyo, Japan, nobukazu@nii.ac.jp
[3]Waseda University, Tokyo, Japan, washizaki@waseda.jp

## Abstract

Firewalls are boundary protection systems that filter incoming and outgoing traffic to/from an Internet node. We present here two firewall patterns used in cloud systems. The Security Group Firewall divides the firewall in customer groups that have similar filtering requirements. The Cloud Web Application Firewall (CWAF) is a special case of a more general Web Application Firewall (WAF), used to protect web applications.

**Keywords:** Firewalls, Security patterns, Network security, cloud computing

## Introduction

Firewalls are boundary protection systems, they filter the input and output of a computational node. We present here two patterns about firewalls that are used in cloud systems. The Cloud Web Application Firewall (CWAF) is a special case of a more general Web Application Firewall (WAF), used to protect web applications. The pattern objectives are:

**Security Group Firewall.** A Security Group Firewall divides the firewall in customer groups that have similar filtering requirements.

**Cloud-based Web Application Firewall (CWAF). C**ontrols access to web applications communicating through HTTP according to authorization rules with the objective of stopping XSS, SQL injection, and similar attacks.

These join other types of firewalls we presented in the past:

**Packet Filter Firewall [**Sch06]. Filters incoming and outgoing network traffic based on packet inspection at the IP level.

**Proxy-based (Network Application) Firewall** [Sch06]. Inspects and filters incoming and outgoing network traffic based on the type of application service requested, which is represented by a proxy.

**Stateful Firewall** [Sch06]. Filters incoming and outgoing network traffic based on state information derived from the previous traffic to avoid checking all the packets in a connection.

**Whitelisting Firewall** [Bon13]. We want to prevent the client to get access to an external site that is considered untrustworthy, or to stop traffic from an untrusted site. Define a list of sites with which we want to communicate (whitelist).

**Application Firewall (Content Firewall**) [Fer13a]. The application firewall filters calls and responses to/from enterprise applications, based on an institution access control policies.
**XML Firewall** [Fer13a]. Filter XML messages to/from enterprise applications, based on business access control policies and the content of the message.

Figure 1 shows the relationships between these firewalls. The Packet Filter Firewall has a fundamental role and is used in conjunction with all the other firewalls,

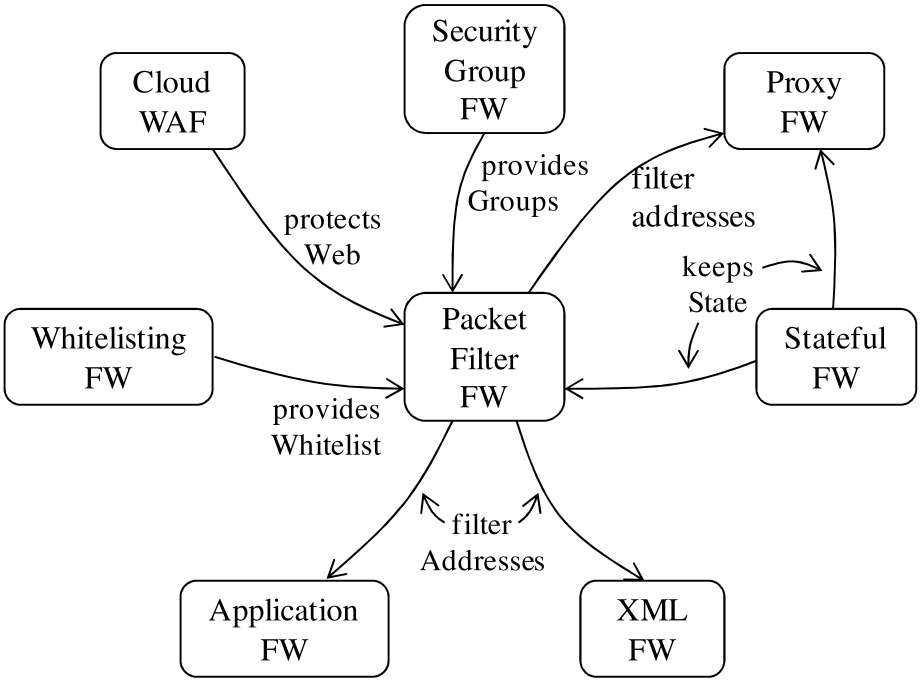

Figure 1. Relationships between firewalls

We describe the new patterns using a modified POSA template [Bus96] and our intended audience are cloud architects, cloud system designers, and cloud application developers. These patterns are part of a catalog of cloud security patterns that complement a Cloud Reference Architecture (RA) [Has13], in order to build a Security Reference Architecture (SRA) [Fer13b] They have, of course, value on their own.

## Security Group Firewall

### Intent

A Security Group Firewall divides the firewall in customer or device groups that have similar filtering requirements. Groups can share rules.

### Example.

A company sells books, records, and DVDs in its bookstore, from their web site, and through mobile salespeople. These groups have different ways of selling but we only have a company firewall to be used by all of them. To make things worse, the firewall also needs to be shared by the system administrators. The different needs make the firewall rules confusing and hard to manage that may result in security vulnerabilities. The company management has become worried about their information being attacked by hackers who could take advantage of this confusion.

**Context**

Cloud computing systems and other distributed systems using virtualization.

**Problem**

Many customers have virtual environments created from the use of a common real server. Having a real firewall shared by all of them would not allow the customers to define filtering rules appropriate for their needs. It would be very complicated for the security administrator to handle all these rules. How can we make this situation more convenient for customers and administrators?

**Forces**

***Manageability—***A large number and variety of rules will confuse the administrators and will result in errors and exposures.

***Rule maintenance***—Changes in customer access or in adapting to new threats are complex because it is not clear which rules may be affected.

***Tailoring—***Being obliged to use the same filtering rules as all the other customers will be annoying to the customers that have to accept a "one size fits all" policy.

***Fineness of filtering—***To decrease the number of rules administrators will tend to use coarser rules, which may result in exposures because the need-to-know policy may be harder to apply.

***Scalability—***If the number of customers increases we will have a consequent increase in the number of rules which will make the situation even more confusing and may lead to errors of the administrators..

**Solution**

Use the "divide and conquer" principle and separate the filtering rules into groups based on roles or another criteria. Rather than defining access rules by networks or static IPs, access is governed by user or device group memberships. By using these member roles, the addition of a new user or server requires no change to the existing rules - the administrator just needs to add the user or server to the group.

***Structure***

Figure 2 shows the class diagram of this pattern. The **Group Firewall** serves a set of users and is connected between an **External Host** and a **Local Host**. The Group Firewall contains a **Set of Filtering Rules,** which is a directory of ordered **Rules.** The Local Host is associated with a **Virtual Environment** which is hosted by a **Server.** The Server includes a set of **Firewall Groups.**

Figure 3 shows the Use Case "Request a Service".

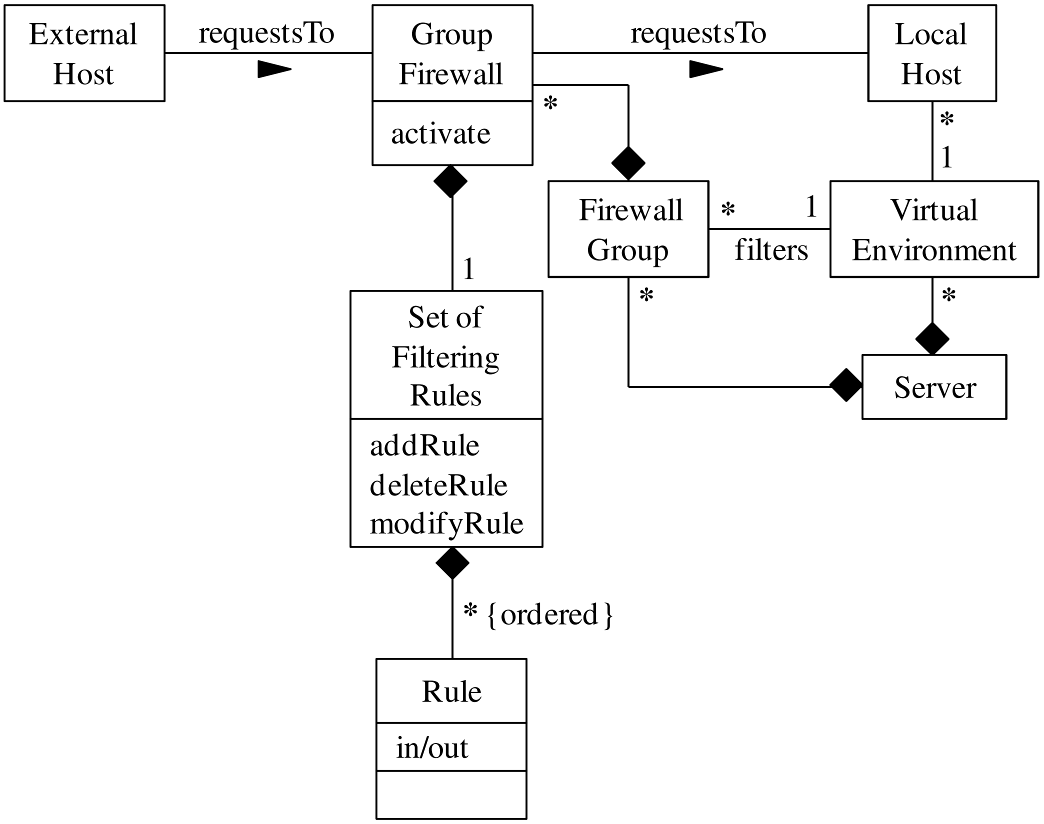

Figure 2. Class diagram of Group Firewall pattern

***Dynamics***

Figure 3 shows the use case "Request a service". An external host requests a service to the Group firewall which routes it to the FirewallGroup, which filters it and sends it to the Virtual Environment.

**Implementation**

In Amazon cloud systems, groups are dedicated for example, to web servers, database servers, and application servers. These groups could have dedicated ports for specific uses and may share some of these ports. The firewall is configured in a default deny-all mode and users would open the ports they need. Traffic may be controlled by protocol or by source address. The firewall is controlled by users' certificates and keys to authorize changes. Groups can be considered roles and be given rights according to their functions. A specific server could then have multiple roles. The rules of each security group are administered separately. Instances (VMs?) can have their own group as shown in Figure 4. The firewall resides in the hypervisor layer.

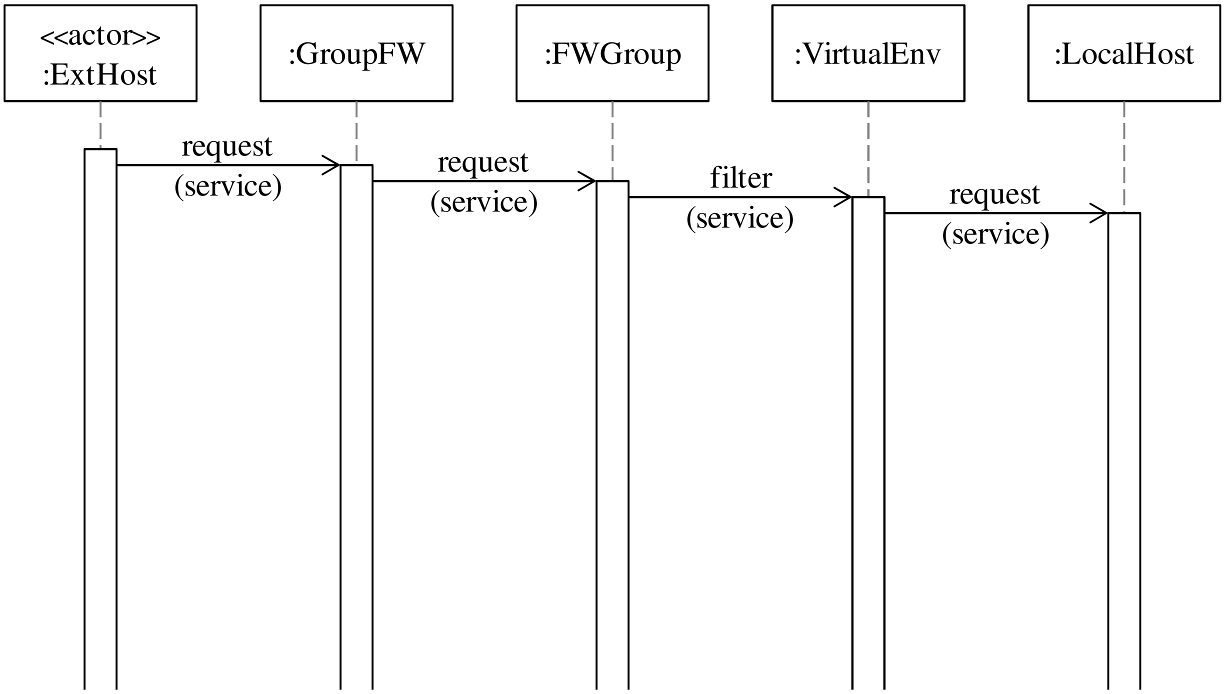

Figure 3. Sequence diagram for the use case “Request Service”

Figure 5 summarizes the use of this firewall. Additional firewalls can be added in each group. Activate and terminte instances, change firewall rules, and other administrative functions must be signed using Amazon’s Secret Access Key. API calls can be encrypted with SSL. AWS IAM (Identity and Authorization Management) is used to authorize access to APIs.

.

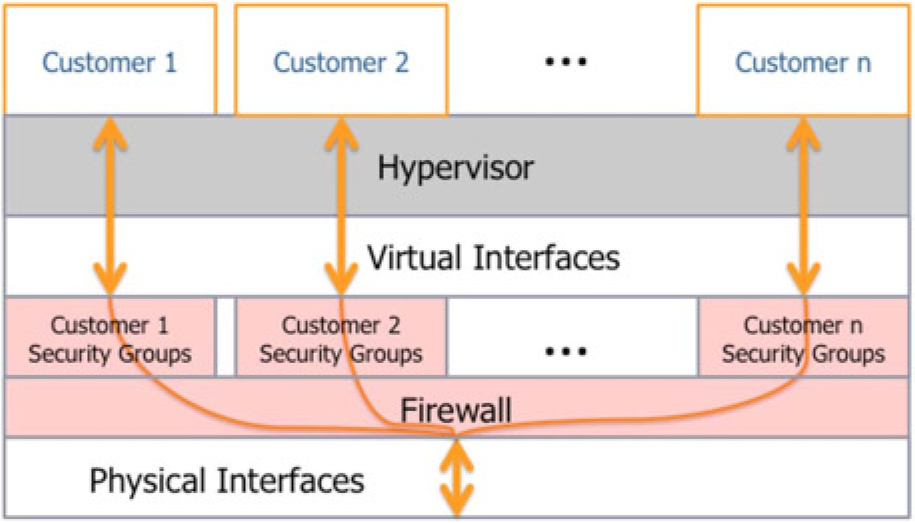

Figure 4 Customer security groups

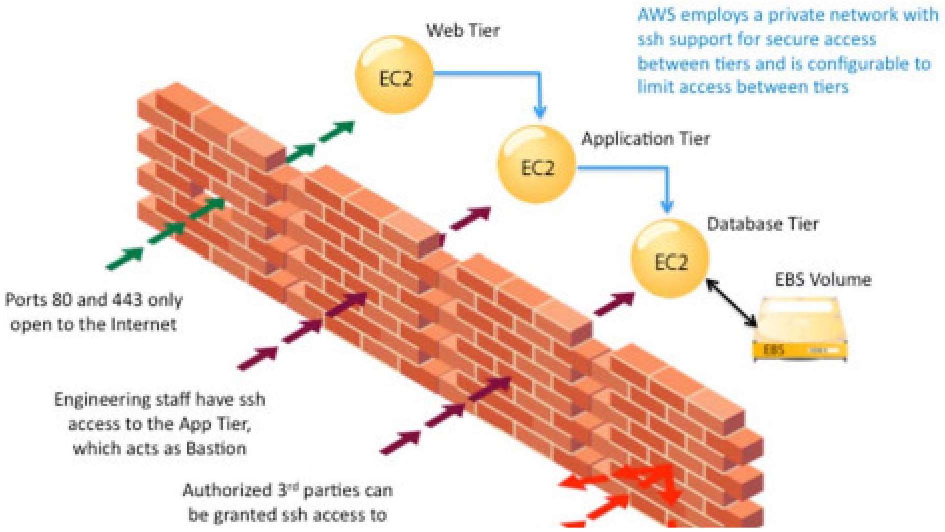
AWS employs a private network with ssh support for secure access between tiers and is configurable to limit access between tiers
Web Tier
EC2
Application Tier
EC2
Database Tier
EC2
EBS Volume
EBS
Ports 80 and 443 only open to the Internet
Engineering staff have ssh access to the App Tier, which acts as Bastion
Authorized 3rd parties can be granted ssh access to

Figure 5. Amazon EC2 Security Group Firewall

**Variants**

Amazon has two types of Group firewalls: Functional [AWS1], and Operational [AWS2].

**Known uses**

- AWS has a Security group Firewall.[AWS1] (See Implementation and Variants)
- Cisco Security Group Firewall (SGFW) allows access rules and security policies to be based on security groupings built from a combination of attributes, such as role, device type, location, time of day, and/or posture [Cis13]. An administrator can allow/deny traffic between groups. It is not clear if this firewall can be used in clouds.
- Fujitsu talks of a “logical separation of computing environments”, which appears to use group firewalls although they don’t say it explicitly [Oku10].
- Dome9 uses security groups to configure and manage the use of servers, web services, and applications in an AWS architecture [Dom].

**Example resolved**

The company decided to use a cloud system to handle its business. By using a group firewall we can assign roles to groups, a security group named “Mobile Sales” could represent all salespersons that are using a mobile device to access network resources. Another group could be for the bookstore, another for the web site, and another for the administrators. Management of the filtering rules is now much simpler and scalable.

**Consequences**

***Manageability*—**Each group may have a small and manageable number of rules.

***Rule maintenance***—A change in the functions of a customer just requires to assign her to another group.

***Tailoring*—**We can create as many groups as we want to fit precisely the needs of each customer.

***Fineness of filtering*—**More specific groups let administrators apply need-to-know policies by defining precise rules.

***Scalability*—**Adding rules in a group has a much smaller impact that adding them to a large group.

**See also**

Role-Based Access Control (RBAC) [Fer13a].--Describe how to assign rights based on the functions or tasks of people in an environment in which control of access to computing resources is required and where there is a large number of users, information types, or a large variety of resources.

See Figure 1 for its relationships to other types of firewalls.

## Cloud-based Web Application Firewall (CWAF)

**Intent**

A CWAF controls access to web applications communicating through HTTP according to authorization rules with the objective of controlling XSS, SQL injection, and similar attacks.

**Example**

A travel agency running as SaaS in a cloud has a web page where it announces tours and where it allows user input to register preferences and traveling tips. A hacker inserts in the user area some code that subsequently steals the user credentials of the visitors to the web site. This is a type of Cross-site Scripting attack (XSS) [Sha12]. After several incidents like this, clients will be afraid to visit this web site and the travel agency will incur in heavy losses.

**Context**

Cloud computing systems and other distributed systems using virtualization and whose main interfaces are through the Internet.

**Problem**

Problems such as XSS, SQL injection [Fer12], and others are common. How can we protect our web interfaces?

**Forces**

***Security—***The web site should be protected of typical attacks

***On-demand services***—users should ble to select different degrees of security and pay according to their selection.

***Transparency—***Users should not be aware of the firewall.

***Scalability***—The number of users accessing the web site should not be limited.

***Overhead***—The defenses should not introduce a significant overhead

**Solution**

Examine GET and POST requests to the web site, filter suspicious traffic and add authorization rules in different sections of the web site.

***Structure***

Figure 4 shows a class diagram of a CWAF. The **ExternalHost** requests services to the **CWAF.** This has a set of **Filtering Rules** that control access to **Resources** in the **Local DNS** of the cloud.

***Dynamics***

Figure5 shows the use case Request a resource. The External host requests a resource through the CWAF which filters improper requests by checking that the address of the external host is not in a blacklist or is in a whitelist.

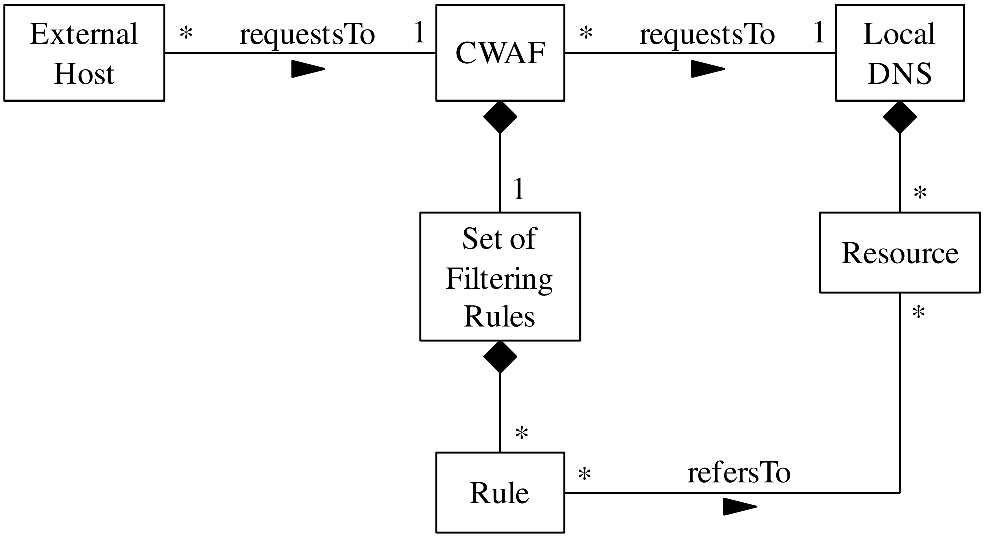

Figure 4. Class diagram of the CWAF pattern

**Implementation**

Cloud-based WAFs are typically centrally orchestrated, which means that threat detection information is shared among all the tenants of the service. This collaboration results in improved detection rates and lower false positives.

The Cloud-based Web Application Firewall is a special case of a WAF. The CWAF  is platform agnostic and does not require any hardware or software changes on the host. Almost all providers require a DNS change, wherein all web traffic is routed through the WAF where it is inspected.

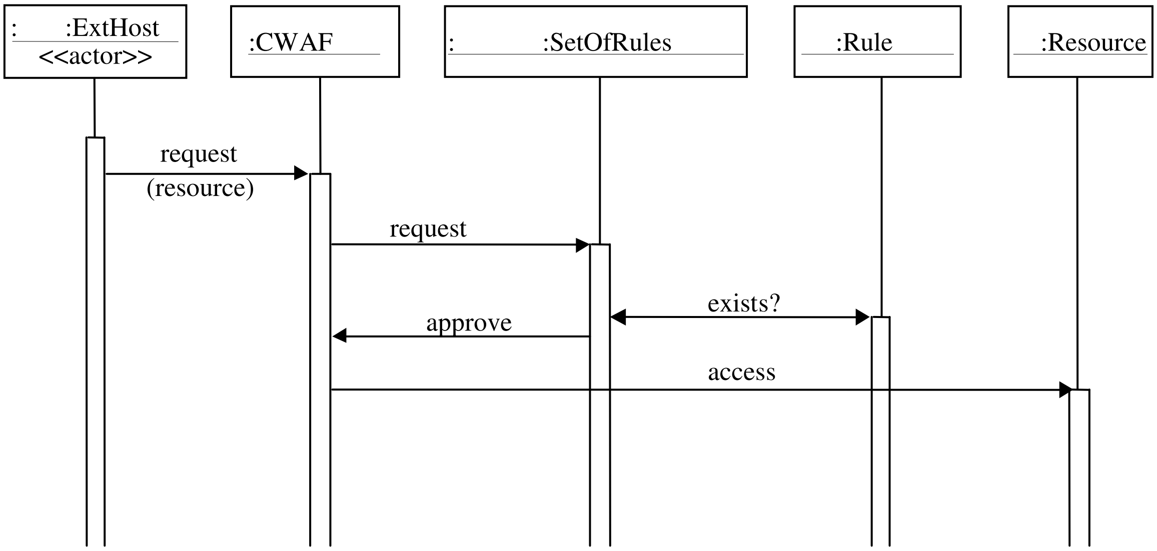

Figure 5. Use case request a resource

**Known uses**

- XyberShield is the only cloud-based WAF which does not require a DNS change, instead relying upon a local 4k script and constant communication to its global service platform of 55 points of presence.
- In 2010, Imperva spun out Incapsula to provide a cloud-based WAF to small to medium sized businesses.
- United Security Providers provides the Secure Entry Server as an Amazon EC2 Cloud-based Web Application Firewall
- Akamai Technologies offers a cloud-based WAF that incorporates advanced features such as rate control and custom rules enabling it to address both layer 7 and DDoS attacks.
- Shaka Technologies provides the Ishlangu Load Balancer ADC as a Cloud-based Web Application Firewall.
- ClearWeb proposed by Nexusguard.
- The AWS Cloud environment lets customers to install WAFs in their own proxy servers [AWS2].

A few more are listed in [APF]

**Example resolved**
Now most of those attacks can be controlled by preventing accesses to some data in the browser.

**Consequences**
***Security—***The web site can be appropriately protected against typical attacks
***On-demand services***—users can select security services that provide different degrees of security and can pay according to their selection.
***Transparency—***Users are not be aware of the firewall.
***Scalability***—The number of users accessing the web site can grow as necessary
***Overhead***—Typical defenses do not introduce a significant overhead

**See also**

- Authorizer [Fer13]: Describe who is authorized to access specific resources in a system, in an environment in which we have resources whose access needs to be controlled. It indicates for each active entity, which resources it can access, and what it can do with them.

- Controlled Virtual Address Space [Fer13]. This is a type of sandbox that defines the resources that a process can access during execution.

See Figure 1 for other types of firewalls.

## Conclusions

These two patterns describe recent types of firewalls that are used to protect cloud systems. Like all patterns, their validation will happen when designers use them in their applications. With these patterns we increase our catalog of security patterns beyond those in [Fer13], they will appear in a new version of that book.

## References

[Ama] Amazon EC2 security groups
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html

[APF] Wikipedia, “Application firewall”, http://en.wikipedia.org/wiki/Application_firewall

[AWS1] Amazon, Cloud Design Patterns, CDP: Functional Firewall Pattern
http://en.clouddesignpattern.org/index.php/CDP:Functional_Firewall_Pattern

[AWS2] Amazon, Cloud Design Patterns, CDP: Operational Firewall Pattern
http://en.clouddesignpattern.org/index.php/CDP:Operational_Firewall_Pattern

[AWS3] Amazon, Cloud Design Patterns, CDP: WAF Proxy patterns:
http://en.clouddesignpattern.org/index.php/CDP:WAF_Proxy_Pattern

[Bon13] Isaura N. Bonilla, Eduardo B. Fernandez, Maria M. Larrondo-Petrie, and Keiko Hashizume," A pattern for Whitelisting Firewalls"*20th Conf. on Pattern Languages of Programs (PLoP 2013)*

[Bus96] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerland, and M.Stal, *Pattern- oriented software architecture*, Wiley 1996.

[Cis13] Cisco, "Access control using security group firewall" http://www.cisco.com/en/US/solutions/collateral/ns340/ns414/ns742/ns744/docs/access_control_using_security.pdf

[Dom] Dome9, http:// http://www.dome9.com/

[Fer13a] E. B. Fernandez, *Security patterns in practice - Designing secure architectures using software patterns*. Wiley Series on Software Design Patterns, Wiley 2013.

[Fer13b] E. B. Fernandez, R. Monge and K. Hashizume, "A Security Reference Architecture", submitted for publication.

[Has13] K. Hashizume, E. B. Fernandez, and M. M. Larrondo-Petrie, "A Reference Architecture for Cloud Computing," submitted for publication.

[Oku10] M. Okuhara, T. Shiozaki, and T. Suzuki, "Security architectures for cloud computing", *Fujitsu Sci. Tech. J., vol. 46, No 4, October 2010, 397-402.*

[Sch06] M. Schumacher, E. B.Fernandez, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns: Integrating security and systems engineering",* Wiley Series on Software Design Patterns. Wiley 2006.

[Sha12] L.K. Shar and H.B.K. Tan, "Defending against cross-site scripting attacks", *Computer*, March 2012, 55-62, IEEE 2012.