Whitelisting Firewall Pattern (WLF)

ISAURA N BONILLA VILLARREAL, Florida Atlantic University
EDUARDO B. FERNANDEZ, Florida Atlantic University
MARIA M. LARRONDO-PETRIE, Florida Atlantic University
KEIKO HASHIZUME, Florida Atlantic University

Firewalls are a useful defense for web sites and they are used in all kinds of systems, from laptops to large multiprocessors. Firewalls filter traffic to/from other web sites to prevent communication with potentially harmful locations. Firewalls typically use an open policy, where all sites are considered acceptable unless we explicitly blacklist them. To improve security we can communicate only with trusted sites using a whitelist (list of trusted sites). We present here a pattern for whitelisting firewalls that complements existing patterns for blacklisting firewalls.

Keywords: Firewall, Whitelisting, pattern, security patterns, threat prevention.

General Terms: Security Patterns

Additional Key Words and Phrases: Firewall, Whitelisting, pattern, security patterns, threat prevention

1. INTRODUCTION

The use of firewalls as a tool to protect web nodes from malicious web sites is a valuable defense to control potentially harmful requests in all kinds of systems, from laptops to large multiprocessors. Firewalls filter incoming and outcoming traffic with the objective of preventing communication with potentially harmful locations. The policy commonly used for them is an open policy, where all sites are considered acceptable unless we blacklist them. This is a useful policy for sites which, because of their purpose, need to reach the widest possible audience. However many sites only need to deal with business partners, remote employees, providers, or similar, all of which are known in advance. In that case we can improve security by communicating only with them; this is the idea of the whitelisting firewall. We describe it as a security pattern, which is a solution to a recurrent security problem (threat) in computer systems, and is described through a template that includes several sections [Fer13]. The sections include the threat being controlled, how the threat is handled using a solution described by a UML diagram, the consequences of using the pattern, the actual systems where the pattern has been used, and related patterns.

The WLF Pattern (See Figure 1) adds to our catalog of security patterns [Fer13], and complements the patterns for blacklisting firewalls [Sch06], which include the Packet Filter Firewall, the Proxy Firewall, and the Stateful Firewall. They complement these patterns because the user can choose which pattern follow up depending on their necessities. Basically, they complement to each other because they form a common solution to the protection of local networks which is to incorporate a firewall to filter unwanted traffic. Figure 1 shows how these patterns relate to each other. According to [Fer13] the Packet Filter Firewall intent is to filter incoming and outgoing network traffic in a computer system based on packet inspection at the IP level. The Proxy Firewall intent is to inspect and filters incoming and outgoing network traffic based on the type of application service to be accessed, or performing the access. This pattern interposes a proxy between the request and the access, and applies controls through this proxy. This is usually done in addition to the normal filtering based on addresses. Finally the Stateful Firewall intent is to filter incoming and outgoing network traffic in a computer system based on state information derived from past communications. State information generally describes whether the incoming packet is part of a new connection, or a continuing communication whose connection was approved previously. In other words, states describe a context for each packet.

Author's address: Isaura N Bonilla Villarreal; 777 Glades Road, Boca Raton, FL 33431; email: ibonill1@fau.edu; Eduardo B. Fernandez; 777 Glades Road, Boca Raton, FL 33431email: fernande@fau.edu; Maria M. Larrondo-Petrie; 777 Glades Road, Boca Raton, FL 33431; email: petrie@fau.edu; Keiko Hashizume; 777 Glades Road, Boca Raton, FL 33431; email: ahashizu@fau.edu

Our intended audience are software designers who need to secure their systems but the patterns could also be of interest to firewall designers.

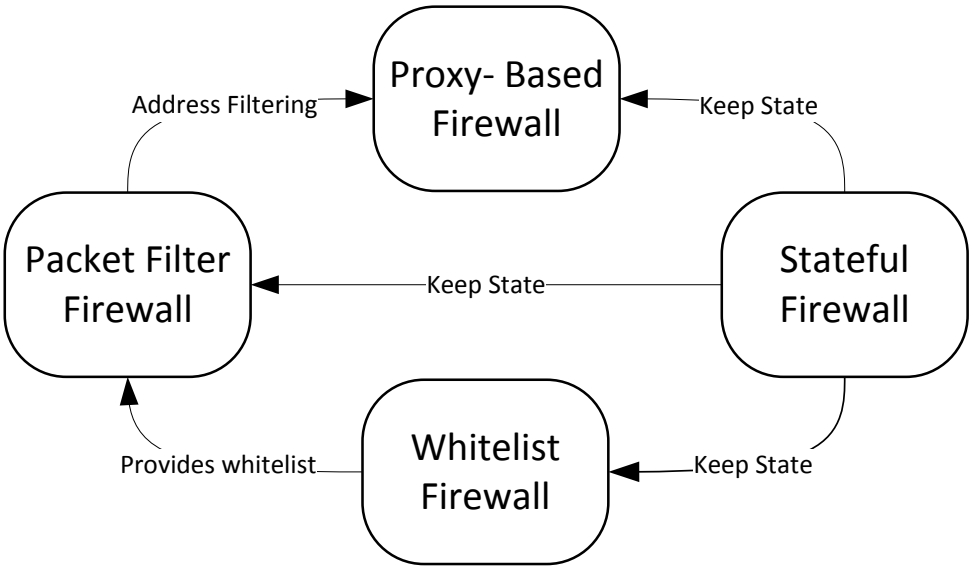

Fig. 1. Pattern Diagram for Firewalls

2. WHITELISTING FIREWALL.

2.1 INTENT

We want to prevent the client to get access to an external site or service (any kind of IP address or port) that is considered untrustworthy, or to stop traffic from an untrusted site. WLF defines a list of sites with which we want to communicate.

2.2 CONTEXT

Nodes connected to the Internet and to other networks who need to enforce their policies to all of the sites as one layer in a defense in depth strategy in their site..

2.3 PROBLEM

Some sites are insecure and may contain malware, which could attack our host or download malware if we visit them. How to prevent or stop the traffic of a system so we do not get access to external sites that are considered untrustworthy?

. The solution will be affected by the following forces:

- *Security*: We handle sensitive assets. We only want to communicate with sites that are known by the user and are trusted. This will increase security.
- *Transparency*: The users of the system should not need to perform special actions or they may not use the filtering.
- *Overhead*: Filtering should not significantly reduce performance.
- *Usability:* It should be easy to apply and manage filtering policies.
- *Number of interlocutors.* The number of sites with which we want to communicate is reasonable small.

2.4 SOLUTION

Use a Whitelisting Firewall which is a filtering mechanism that can enforce communication with only approved sites by keeping a list of acceptable interlocutors(hosts or sites).

STRUCTURE

The class diagram for the Whitelisting Firewall is shown in Figure 2. The **Whitelisting (WL) Firewall** intercepts the traffic between a **LocalHost** and a set of **ExternalHosts**. The WLFirewall includes a **Whitelist**, which is composed of a set of ordered rules. The ordering of rules accelerates checking by avoiding checking of sites included in site groups. Some of the rules may be **ExplicitRules** or may be **ImplicitRules** (default) rules, which apply to all sites.

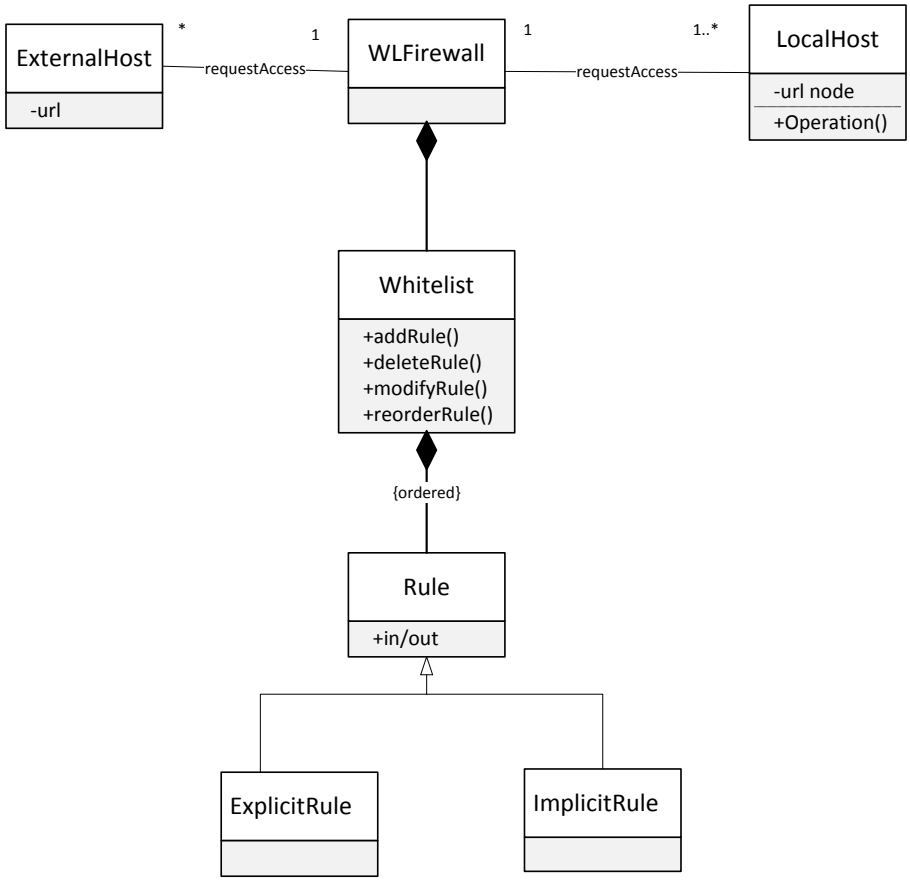

Fig. 2. Class diagram for the Whitelisting Firewall pattern.

DYNAMICS

We describe the dynamic aspects of the Whitelisting Firewall using a sequence diagram for one of its basic use cases. Basically, the Whitelisting Firewall applies the policy of a closed world, where every address is denied access unless explicitly permitted [EMA12, Fer13]. Figure 3 shows a sequence diagram for the Use Case "Filter an incoming request". A **node** requests to run some **service** or **application**. The **Whitelisting Firewall** receives this request and using a list of known and trusted sites checks if the request comes from or goes to a trusted site and allows the connection or denies the request.

UC: Filtering an Incoming Request

Summary: A host in a remote network wants access to a site 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.

Actors: A host in an external network trying to access a site (local host).

Precondition: A set of rules to filter the request exist in the whitelist of the firewall.

Description:

An external host requests access to the local host.

A whitelisting firewall filters the request according to its rules to accept or deny the access.

If the request is accepted, the firewall allows access to the site.

Alternate Flow: If no rule allows the access, the request is denied.

Post condition: The firewall has accepted the access of a trustworthy site to the Localhost.

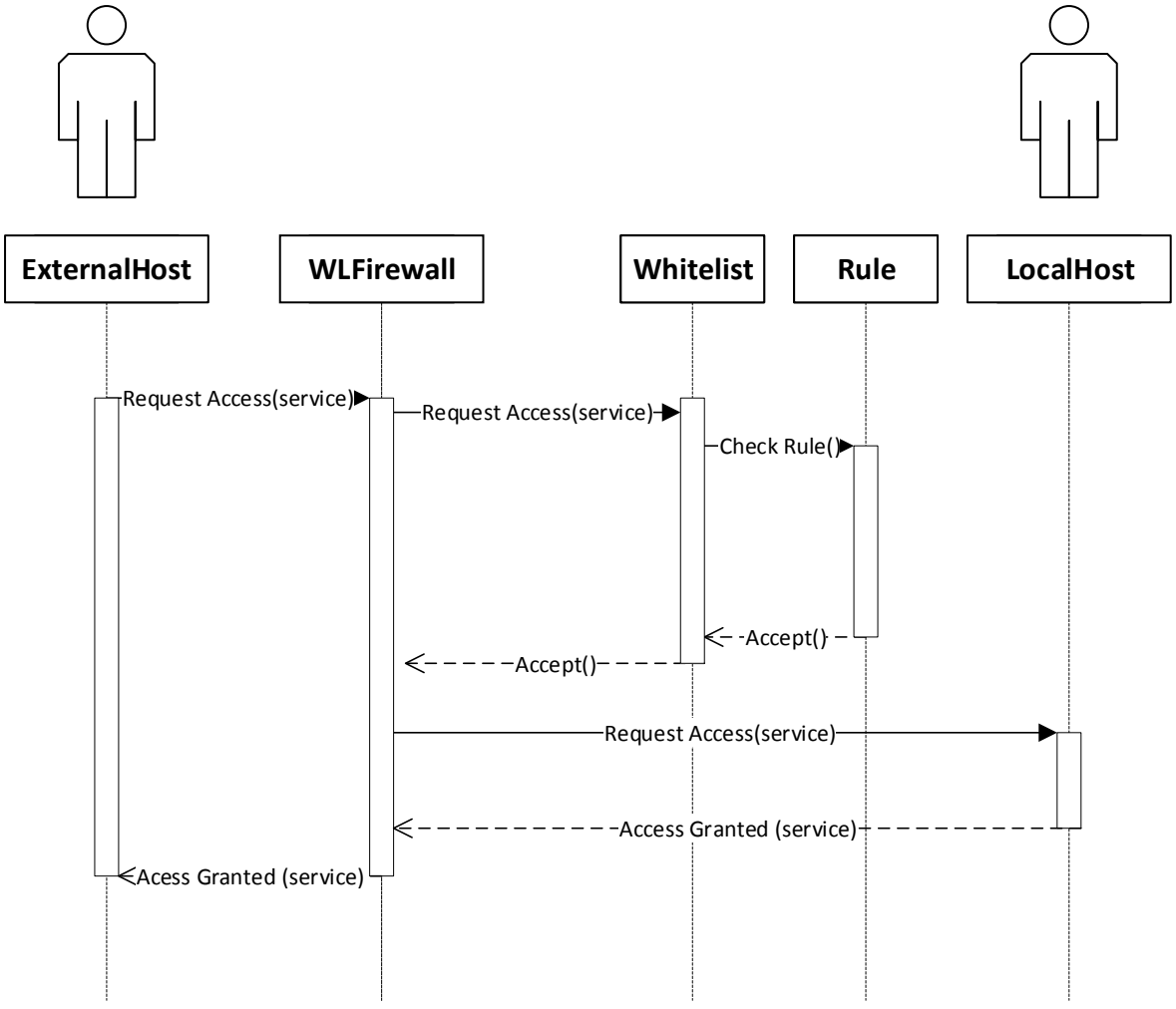

Fig.3. Sequence diagram for the UC: Filter an incoming request.

2.5 IMPLEMENTATION

The Whitelisting Firewall should reside on a host computer and maintain a local set of rules, which are maintained by a security administrator. The institution must define these rules according to their policies. The whitelist is usually rather short and there is no need for hardware assistance to search it and the filtering should happen at the IP layer. It is possible to filter also at the application layer and stateful firewalls could accelerate filtering.

There are specific approaches to use with the Whitelisting Firewall such as Gold Image and Digital Certificates. With the Gold Image for static systems, the list is created by first hashing a standard workstation image. After using an image to build the first whitelist, keeping it up to date will be the biggest challenge facing most groups.

On the other hand, digital certificates are one of the most effective techniques to trust certain publishers of software. Based on their signature the certificates are automatically in the whitelist and are considered trusted [SAN12].

2.6 VARIANTS

We can combine Whitelisting and Blacklisting by prioritizing them. For some addresses, we can apply Blacklisting, then Whitelisting for the remaining addresses. Another possibility is to apply first whitelisting and then blacklisting. Also, we can use of Gold image as was stated previously.

2.7 KNOWN USES

- Bit9 Parity: It uses a software registry, a locally installed management server and a client to enforce software policies throughout the enterprise. Bit9 developed an adaptive whitelist strategy. The proprietary Global Software Registry is an online index. This list contains unique applications and the registry acts as a reference library for IT administrators building their whitelists [Bit12]. Bit9 has also built a Parity Suite based on the qualities of transparency, flexibility and scalability [Ema12].
- Coretrace Bouncer: It contains a standard file-based whitelisting protection mechanisms [Cor12].
- McAfee: This centrally-managed whitelisting solution uses a dynamic trust model and security features that try to control advanced persistent threats [McA13].

2.8 CONSEQUENCES

This pattern has the following advantages:

- *Security*. Whitelisting is more secure than blacklisting, because we only deal with trusted sites.
- *Transparency*: Filtering is transparent to the users.
- *Overhead:* Checking is faster because whitelists are shorter than blacklists.
- *Usability*. The list is short and only requires updates when a new site is added.

This pattern has the following disadvantages:

- *Annoyance*. It may cause some users to be annoyed because they cannot download applications from anywhere or visit sites not in the list.
- *Effect of address changes*. If a trusted site in the whitelist changes its address we need to update the list.
- *Wolf in sheep clothes*. A trusted site may still be harmful.
- *Breadth.* It is inappropriate if we need to offer products or services and we want to reach a wide audience.
- *Number of interlocutors.* If the number of sites with which we want to communicate is large, this may not be a convenient approach.

2.9 RELATED PATTERNS

- Blacklisting Firewalls: Filter traffic from untrusted sites based on blacklists [Sch06].
- Credential: Provide secure means of recording authentication and authorization information for use in distributed systems [Fer13].

3. CONCLUSIONS

The whitelisting firewall pattern complements the three blacklisting firewall patterns of [Sch06]. The IDS and VPN patterns that handle some aspects of security in networks are also useful for network security [Fer13]. Finally, other patterns useful for network security are the cryptographic patterns that are described in [Fer13].

Acknowledgments: We thank our shepherd Veli-Pekka.Eloranta for his valuable suggestions that improved this paper.

REFERENCES

[Bit12] Bit9 products. Bit9 Security Platform.Retrieved from: http://www.bit9.com, Last accessed: November 17, 2012.

[Cor12] Coretrace Corp. Coretrace products. Retrieved from: http://www.coretrace.com/products-2/bouncer-overview#application. Last accessed: November 17, 2012.

[EMA12] EMA Corporation. Realistic Security, Realistically Deployed: Today's Application Control and Whitelisting. October 2012. Retrieved from: http://www.bit9.com. Last accessed: February 17, 2012.

[Fer13] E.B.Fernandez, Security patterns in practice: Building secure architectures using software patterns, Wiley Series on Software Design Patterns, June 2013..

[McA13] McAfee Products. McAfee Application Control. Retrieved from: http://www.mcafee.com/us/products/application-control.aspx. Last accessed: March 21, 2012.

[SAN12] SANS Institute. Application Whitelisting: Panacea or Propaganda? Web document. Retrieved from: http://www.sans.org/reading_room/whitepapers/application/application-whitelisting-panacea-propaganda_33599. Last accessed: November 17, 2012.

[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, 2006.