A framework for enforcing user-based authorization policies on packet filter firewalls

André Zúquete[1], Pedro Correia[2], Miguel Rocha[2]

[1] Dep. of Electronics, Telecommunications and Informatics/IEETA, Univ. of Aveiro
[2] IEETA, Univ. of Aveiro

Abstract. Packet filter firewalls are fundamental elements to prevent unauthorized traffic to reach protected networks or hosts. However, they have to take decisions about packets based on their contents, and currently packets do not contain any information about the entity responsible for its generation. In this paper we propose a framework that tackle this problem. The framework adds extra information to packets, which enables a firewall to authenticate its origin and to get an identity attribute for discriminating the entity responsible for the packet, upon which an access control policy can be implemented. This framework uses trusted third party services for authenticating people and providing related identity attributes for firewalls. For a proof of concept we implemented a prototype in Linux machines using `iptables` and personal identity smartcards.

1 Introduction

Packet filter firewalls use fundamentally the information present in a packet to take three fundamental decisions: accept, discard, or reject it with an error. However, currently IP packets do not carry any information regarding the identity of the entity responsible for their generation. They carry hosts' information (IP address), which may, in some cases, be loosely linked to an identity, but this information is not trustworthy, as there is no cryptographic guaranties of the identity of the source (IP addresses can be spoofed). Therefore, packet filters cannot be used for trustworthy, identity-based filtering.

In this paper we describe a framework for enhancing existing packet filtering strategies in order to enable them to explore filtering rules based on the identity of the user responsible for the traffic under scrutiny. With this enhancement, we can use a packet filtering system to enforce user-based access control, on a per-packet basis, to a particular host or network. The proposed framework borrows its key concepts from single-sign on (SSO) proposals, such as Kerberos [1] and Web browser SSO authentication using SAML [2], but it is the first one that we know of that works at IP level.

2 Framework Architecture

Our identity-related packet filtering strategy is based on the following credentials used by a source host: an Access Identity Pseudonym (AIP), a Pseudonym Access Key (PAK), an Access Identity Token (AIT), and an Access Authenticator (AA).

1	C → S	: packet
2	FW→ C	: no source UId, S, IdP
3	C →IdP	: user authentication
4	IdP→ C	: AIP, PAK, AIT
5	C →FW	: UId registration, AIT, AA
6	FW→ C	: UId registered, AA
7	C → S	: packet, AA
	FW	: AA's AIP → PAK, UId
	FW	: Validate AA with PAK
	FW	: check UId access
8	FW→ S	: packet

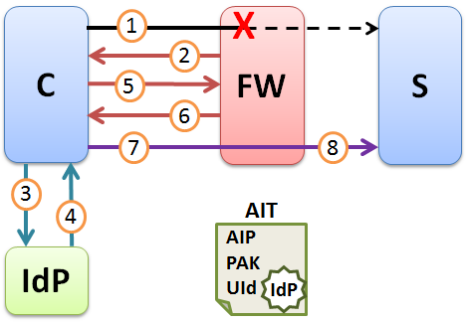

Fig. 1. High-level protocol for adding source-identity marks to packets from a client C to a server S when the latter is protected by a firewall FW enforcing identity-based access control using the UId attribute. This attribute is extracted from an AIT issued and sealed by an IdP.

These credentials will be used to produce and verify extra identity marks on IP packets, providing identification and authentication of the source entity (user).

The usage of these credentials can be summarized as follows (see Fig. 1). When a user initiates an interaction from client C to a server S protected by a packet filter firewall (FW), the latter will look for an identity mark in ingress traffic. If not present, FW sends back an error, signaling the absence of that mark and providing instructions about how to proceed, namely a reference to an Identity Provider (IdP) with which FW has some trust relationship. Then the user enters a dialog with the IdP to prove his/her identity and to get a temporary AIP, PAK and AIT trio. The AIT is similar to a Kerberos ticket and includes a User Identifier (UId) attribute associated with a person, and provided by the IdP, upon which the FW should enforce access control decisions.

The operating system of C caches those credentials for as long as the identity of the user is the same, discarding them as soon as the user logs out of C. While cached, they will be used to prove the identity of the user responsible for the traffic sent from C to S. First, C sends to S (and, transparently, to FW), its AIT, together with an AA (which includes the AIP), to prove the correctness of the AIT. The FW caches the AIT, which will be used thereafter to identify the user responsible for all the traffic including the its AIP. Afterwards, each packet from C to S will carry an AA, which enables the FW to check if the packet really comes its AIP and to get the user's UId attribute from the cached AIT, upon which an identity-based, per-packet access control policy can be enforced. At the end, this whole process boils down to provide a trustworthy binding of a UId to each packet on a firewall.

3 Prototype Implementation

We have implemented a prototype of our packet authentication framework in Linux systems, using the `iptables` packet facilities, namely the possibility to send packet to user-land processes with `nfqueues`. For user authentication we used personal identification smartcards, namely the ones deployed in Portugal.

The packet marking process with an AA uses a new, 24-byte IP header option, which doesn't need to be removed by a FW before forwarding the packet to the server. On the other hand, the size available for all IP options is shortly limited to a maximum of 40 bytes, therefore some coexistence problems with other options may occur. The structure of this option is similar to the IPSec's AH [3].

The client application is a process that intercepts all outbound packets and decides if they are to be marked or not, depending on their destination. This application also tackles the remote interaction with firewalls and IdPs and the local interaction with the user and with his/her identification smartcards.

The firewall access control is performed by an application and by `iptables` rules using packet marks. The application intercepts all inbound packets and checks if they are marked or not. If a packet is not marked, the application drops it and sends an ICMP error message (with a new type). Otherwise, it verifies the AA using cached AIT information. If correctly marked, the UId associated to the cached AIT is used to mark the packet internally to the `iptables`.

The UId-based access control part of the firewall was implemented with `iptables` rules using internal packet marks. This very simple method for enforcing UId-based access control works as long as users' identifiers are numbers; for richer identifiers, such as names, it doesn't work. Because of this restriction, we used civil identification numbers for UId (extracted from the subject's SERIALNUMBER of users' public key certificates).

4 Conclusion

We presented a framework for enabling a packet filter firewall to enforce trustworthy identity-based authorization policies. The framework uses one or more central IdP services, which can use different authentication paradigms; we used only personal identification and authentication based on a national identity smartcard, namely the Portuguese one. The framework enables users to be authenticated only when necessary, not in advance. A fully operational proof-of-concept prototype was implemented.

Acknowledgements

This work was partially funded by FEDER through the Operational Program Competitiveness Factors - COMPETE and by National Funds through FCT - Foundation for Science and Technology in the context of projects FCOMP-01-0124-FEDER-022682 and BoDes (FCT references PEst-C/EEI/UI0127/2011 and PTDC/EEA-TEL/101880/2008).

References

1. Neuman, C., Yu, T., Hartman, S., Raeburn, K.: The Kerberos Network Authentication Service (V5). RFC 4120 (July 2005)
2. Hughes, J., Cantor, S., Hodges, J., Hirsch, F., Mishra, P., Philpott, R., Maler, E.: Profiles for the OASIS Security Assertion Markup Language (SAML) 2.0. OASIS Standard (March 2005)
3. Jones, P.: Registration of the text/red MIME Sub-Type. RFC 4102 (June 2005)