International Journal of Multidisciplinary and Current Research

Research Article

ISSN: 2321-3124
Available at: http://ijmcr.com

Policy Management for IPsec Protocol

Sameer

Shah Satnam Ji P.G Boys ' College, Sirsa, India

Accepted 18 Oct 2015, Available online 26 Oct 2015, **Vol.3** (**Sept/Oct 2015 issue)**

Abstract

IPsec, the standard suite of protocols to provide security in IP networks, and IKE, the commonly used key management protocol for IPsec, do not address the more general problem of how security policies should be distributed to IPSec nodes. Recent IETF work in the area of network security provides a definition of the basic requirements of an IP Security Policy System (IPSP) and a proposal of a Security Policy Protocol (SPP) to exchange security policies. IPSP recommends that traditional mechanisms for distributing network management information (SNMP, COPS) should also be taken into consideration. The first objective of this paper is to evaluate the suitability of existing network management mechanisms to achieve the goals of IPSP. Subsequently, the paper describes and discusses an approach followed in the implementation of an IPSP system, with emphasis on the implementation of SPP.

Keywords: *IPsec, IETF etc.*

Introduction

With the advent of IPsec [I], security at the network layer is becoming increasingly popular. IPsec is used to create a wide range of protection schemes, like secure virtual private networks, secure end-to-end communications, secure remote access, and so on. However, IPsec scenarios have a major drawback they require prior mutual agreement on acceptable cryptographic parameters and security services between all IPsec parties that participate in the communication. This results in complex management tasks that become especially difficult as networks scale up. Consider the general case of two hosts that need to protect their communication across the Internet with IPsec. The related traffic may travel across heterogeneous networks, and may be subjected to different security policies, controlled by different administrative entities, in other words, along the communication path between the two hosts there may be other IPsec devices (security gateways), enforcing different policies on the traffic they are forwarding. The end nodes need a way to discover the presence of such gateways, together with the policies that they enforce. While a mechanism for negotiating IPsec Security Associations (SAS) between hosts that share common security policy already exists (Internet Key Exchange, IKE [2]), neither the IPsec protocols nor IKE offer a way to provide the security policies under which the SAS operate. IPSP [3] was created to offer a comprehensive solution to these problems. The goal of IPSP is to provide a scalable, decentralized framework for managing, discovering, and resolving IPsec policies. After a short introduction to the rationales of the approached area, the paper continues, in Section 2, with the presentation of the state of the art in the field of network-level security policy distribution, together with concise analysis of each mechanism brought into light Section 3 addresses important IPSP practical issues of IPSP implementation. Finally, the conclusions and future work are presented in Section 4.

A Rationales and Goals of IPSP Systems

Policy systems compliant to the lETF's IPSP standards to provide a common policy model that defines the semantics of IPsec policy, a reliable mechanism for gateway and policy discovery, a comprehensive and rich enough language to express policies, a mean to delegate responsibility for policy configuration/distribution, and mechanisms for policy resolution and compliance checking. The administration of policy systems is inherently heterogeneous and decentralized, so IPSP must offer a viable model of trust and authentication for policy exchanges. These issues are addressed by SPP [4], which requires message authentication based on digital signatures, and proposes a trust model for guaranteeing that the policies transported by its messages are provided by authoritative entities. Specific examples of SPP operation and a more detailed evaluation of the overall IPSP and SPP architectures are discussed in [5].

SPP provides mechanisms for gateway discovery and for policy discovery, exchange, and resolution. SPP can be

used inside single security domains, nested domains and across domains that are under different administration. From an operational point of view, SPP is very similar to DNS, clients ask servers for a specific piece of information and servers cooperate in resolving the query. For increasing performance, each server also maintains a cache with the most recent queries resolved. Unlike DNS domains, the SPP domains (i.e., the security domains) do not respect a hierarchical structure, except for the particular case of nested domains. The security models proposed by the two systems are also different: in secure DNS, each Resource Record in a message is signed by the originating server, where as in SPP the whole message is digitally signed. SPP messages are verified and re-signed by each server that participates in an SPP exchange. Section 3.4 examines the authentication mechanism in more depth.

Existing policy distribution mechanism

Since network level security policy can be seen as a part of the more general network management framework, since automated mechanisms for network management already exist, it is natural to adopt these mechanisms for security policy distribution. Some of the protocols that can be taken into consideration are SNMP, COPS, and LDAP. In the following sections, we examine different aspects related to the use of such protocols to achieve the goals of an I PSP system.

A SNMP: SNMP [6] is traditionally used for network level management. The I PSP WG has recently proposed a configuration SNMP MIB for IPsecZIKE policy, which allows network administrators to configure IPsec devices [7].

The SNMP's scope in a heterogeneous network environment is controlled by the same administrative authority. Therefore, from the perspective of IPSP requirements, SNMP answers the need for policy distribution inside single security domains, by using the SNMP-GONF [8] configuration model, via SNMP the policy server configures and is directly controlled by policy clients with the appropriate IPSec policies, by setting the values of the appropriate MIB policy objects. A client can initiate/accept an IPsec communication if configuration for that specific communication type has already been distributed to it. SNMP provides authentication/confidentiality services based on symmetric cryptography. As a consequence, it is appropriate for securely distributing configuration information inside single administrative domains and across domains that have prior agreements. SNMP can neither support secure communications between two unknown entities, nor dynamic policy discovery mechanisms as implemented by SPP. Inside single administrative domains, though SNMP offers a viable solution for policy configuration, it is regarded as low cost one due to the fact that at some sites SNMP-based management schemes are already used for common network management tasks.

B COPS: COPS (Common Open Policy Service) [10] with the extensions for provisioning [11] is another protocol that can be used to distribute policy information to IPsec devices. COPS defines two basic roles: the policy server, or policy decision point (PDP), and policy clients, also called policy enforcement points (PEPs). The IPSP working group has proposed a Policy Information Base (PIB) to be used for IPSec/IKE policy provisioning [12], The IPsec PIB downloaded to IPsec devices enables them to construct a Security Policy Database (SPD). The policy information transferred with COPS is always the local policy, except for static IPsec scenarios, in which policy configuration on PDP's has agreed earlier. In general, there cannot exist COPS exchanges between different administrative domains because PDP-to-POP communications are not supported by COPS, while PEP's from remote domains could hardly communicate with PDP's in domains that are not trusted, according to the COPS architecture and to its symmetric cryptography-based authentication model, the PEP's would have to maintain persistent TCP connections and share it secretly with the remote PDP's. These strong limitations make COPS unsuitable for policy discovery across different domains. Furthermore, COPS may prove not to be appropriate for domains that have provision of IPsec policy to numerous end-nodes, a trade-off has to be reached between performance and the number of PDP's inside the domain. In our opinion, COPS is a good solution for policy provisioning in the domains that enforces IPsec policy.

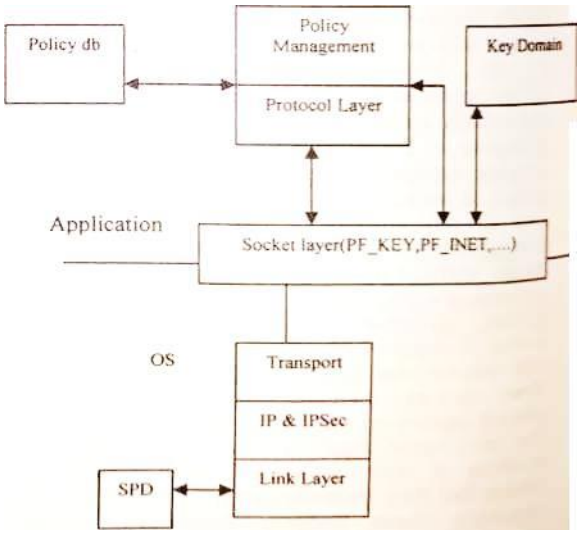

Fig 1 Two layer policy module architecture

Recent IETF work [14] specifies the mapping of the generic policy information model classes defined in [13] to a directory that uses LDAPv3 [15] as its access protocol. Since the policy LDAP schema potentially exists, let us examine some of the policy distribution scenarios that

could make use of it. Inside a security domain, the nodes that need IPsec policies can be given direct access to an LDAP policy repository, or the policy server can rely upon an LDAP server for providing policy information to policy client's queries (the queries may be encapsulated in other protocols, tike SPP). Both these architectures seem to follow a centralized approach. However, this may not be always true, with the intrinsic distributed nature of LDAP, behind the LDAP server known by a single client, a whole distributed mesh of LDAP servers may be hidden. The servers manage different policy repositories and cooperate in providing policy information to requestors (for example, by using the LDAPv3 standard referral mechanism). This model makes us to consider LDAP appropriate for policy distribution even across security domains under different administration. The more appropriate security model is the one that is based on public-key infrastructures (PKI), the system can work if servers rely on mutually trusted third parties (CA, Certification Authority) for authentication, the availability and performance issues of the LDAP model are to be evaluated, IPsecZIKE processing already introduces a considerable overhead to inputZoutput traffic processing, and therefore, an efficient policy distribution system would be preferred.

Experimental implementation of IPSP

The conclusion of analysis of ISPS [5] is, fit is stated that in- spite of its deliberate flexibility), ISPS can offer a viable uniform solution to the problem of security policy distribution in TCP/IP networks. Hence our implementation could now help us in better evaluating pros and cons of the system.

In implementation [16], IPSP functionalities are provided by an application-level policy module, a daemon present on any IPSP system. The module provides three different behaviours, depending upon the local system instantiation: policy server, policy client or security gateway, the latter being a special case of the policy client instantiation.

A Policy Module Architecture

The policy module is divided into two layers. The upper layer handles the communication with the local IPsec kernel and manages the policy databases (the cache database clients, the cache and the domain database on servers) while the lower layer handles the SPP protocol exchanges (it implements the security policy communication protocol). The two-layer architecture (Figure 1) permits higher parallelism of policy module operations, therefore a performance enhancement is obtained. Separation of tasks also makes the system easier to manage and control. The protocol layer is practically identical for all module types, with two exceptions: common policy clients do not manage SPP KEEPALIVE messages (they are only exchanged between servers and security gateways), while servers also implement authorization control. Indeed, the FSM's that model the protocol layer behaviour have the same number of states on all IPSP systems (five), but server FSM's have slightly more transitions (21, while client FSM's have 19). It is the policy layer that discriminates between the various IPSP module types. On clients, the policy layer provides basically two functions, communication with the local IPsec kernel via PF-KEY and management of die policy cache database. On servers, die policy layer provides policy resolution to hosts inside the local security domain, which responds to queries that come from outside local domain, then forwards queries to other domains, manages the domain's policy repository, and provides a policy compliance checking mechanism applied to all die received policies. The policy processing on an end-node of IPSP client is triggered by the receipt of a local request for policy resolution. Whenever the IPSec kernel cannot find an SPD entry pointing to an active SA for an outgoing IP packet, it asks the policy module to acquire one. The policy may be found in the local cache from a previous SPP exchange, or a policy resolution process is started (an SPP query message is sent to die policy server authoritative for the domain). On policy servers the IPSP processing is basically triggered by die receipt of a policy query, while on security gateway clients there are two other triggering events: the receipt of an incoming policy configuration SPP message from the authoritative server or the receipt of an SPP message from a remote domain. A fundamental task of security gateways is to forward all SPP messages that comes from outside the domain's boundaries to the local policy server.

B Extended PF-KEY API

PF-KEY is a standard interface defined for communication between trusted applications and an operating system's IPsec kernel. SA information can be inserted into and retrieved from the kernel's SA databases using a defined set of messages. The PF-KEY interface is traditionally used the IPsec key management layer. By extending the basic PF-KEY functionality, we have the policy module interacting with the IPsec kernel in a similar way the key management applications do as in Figure 1. Our extended PF-KEY specification defines four message types together with the related message.

The policy validation mechanism is the validation of the chain-of-trust. By means of the SPP policy server record, a server claims its authorization over a node. The presence of such a record is not enough for proving the server's authoritativeness. For this purpose, a signed proof has to be provided (e.g., inside the server's certificate). This proof can be provided by either an ad-hoc X.509v3 extension, put into the server certificate, or by using an additional attribute certificate, whose format is currently being defined in X.509v4. Both options have

relative merits that should be investigated once the attribute certificate issues have been clarified by the IETF-PKIX working group.

References

[1]. S. Kent, R. Atkinson, Security Architecture for the Internet protocol: RFC-2401, November 1998.

[2]. D. Harkins, D. Carrel, The Internet Key Exchange, RFC 2409, November 1998

[3]. M. Blaze, A. Keromytis, M. Richardson , L. A. Sanchez, Policy Architecture, Internet Draft, July 2000.

[4]. LA. Sanchez, M. N. Condell, Security Policy Protocol, Internet Draft, July 2000.

[5]. M. Baltatu, A. Lioy, D. Mazzocchi , Security Policy System: Status and Perspective, Proceedings of the IEEE International Conference on Networks 2000 (ICON -2000), Pages 278-284, September 2000.

[6]. J. Case, R. Mundy, D. Partain, B. Stewart, An introduction to Version 3 of the Internet-standard Network Management Framework, RFC 2570, April 1999.

[7]. M. Baer, R. Charlet, W. Hardaker, D. Partain, J. Sapeira, C. Wang, IPsec Policy Conjguration MIB, Internet Draft, February 2001.

[8]. M. MacFaden, J. Sapeira, W. Tackabury , Configuring Networks and Devices with SNMP, Internet Draft, November 2000

[9]. U. Blumenthal, B. Wijnen, User-based Security Model for version of the SNMP, RFC 2574, April 1999.

[10]. D.Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan , A. Sastry, The COPS (Common Open Policy Service) Protocol, RFC 2748, January 2000.

[11]. K.H. Chan , D . Durham , S. Gai S.H erzog, F.Reichmeyer J. eligson , A. Smith, R. Yavatkar , COPS Usage for Provisioning (COPS-PR), Internet Draft, September 2000.

[12]. M. Li, D. Ameson , A. Doria, J . Jackson , C WangJPsec Policy Information Base, Internet Draft, Nov 2000.

[13]. B. Moore, E. Ellesson, J. Strassner, A. Westerinen, Policy Core Information Model Version 1 Specifcation .RFC, 3060, February 2001.

[14]. J. Strassner , E . Ellesson , B Moore, R. Moats, Policy Core LDAP Schema, Internet Draft, March 2001.

[15]. M.Wahl , T. Howes , S. Kille, Lightweight Directory Access Protocol (v3), RFC 2251, December 1997.

[16]. M. Baltatu, TCP/IP Networks Security and Policy Based Ph.D. thesis , Politecnico di Torino , Computer Engineering Department, October 2000.

[17]. D. McDonald, C . Metz, B. Phm, PF-KEY Management API Version 2, RFC 2367, July 1998.

[18]. A . Keromytis , J. Ioannidis, J. Smith, Implementing IPsec, Proceedings of Global Internet (GlobeCom) 97