A Protection Model for Network Communications Based on Security Levels

Jansen Carlo Sena
Paulo Lício de Geus
Computing Institute
State University of Campinas
Campinas - São Paulo - Brazil

Abstract—**IPSec was specified to provide authentication, integrity and confidentiality to IP packets. However, the large variety of cryptographic and security options available may cause that the effective cost-protection relation does not correspond to the desired requirements. In this context, this paper presents SLM (*Security Level Model*), which aims to rationalize the use of IPSec through security levels that group parameters together according to the degrees of protection they offer. High-level information, which are platform-independent and centralized on a LDAP server, are queried by a Perl-implemented component, which generates specific configurations for IPSec and IKE.**

Keywords— **IPSec, Security, System Administration**

I. Introduction

Developed to support a network structure capable of operating even in face of unexpected changes on its topology, the IP protocol did not have security as a fundamental aspect of its project.

Initially limited to the academic and military environments, the Internet used to maintain a harmonious cooperation among its users. However, its popularization made it clear that the worldwide network and its technology were extremely fragile and disarmed when faced with the vulnerabilities that had not been exploited in the past. In this context, and in an attempt to improve security to the IP protocol, the IETF (Internet Task Engineering Force) decided to specify IPSec, an IP extension set capable of avoiding attacks like address spoofing and modification and analysis of packet contents. The protection is applied based on specific rules for each type of service traffic so that pre-determined parameters are used. Cryptographic algorithms with its key lengths and lifetime, operation modes and connection direction, all must be associated with each service which one intends to protect. Besides, the security policy used must be shared by all entities that need to exchange information. These factors can increase IPSec deployment cost.

We specified SLM to reduce the complexity of the rule formulation process and to centralize the security policy in order to facilitate the use of IPSec, making it viable to diverse computing environments and also, allowing it to be applied beyond today's normal use as a tool for configuration of static environments, such as the traditional VPN.

The basic aspects of IPSec are presented in Section II. Section III describes SLM, including its components and general structure. The main features of the SLM implementation are described in Section IV. Section V and Section VI present conclusions and future work, respectively.

II. IPSec

IPSec (IP Security) [1], [2], [3] is based, fundamentally, on two protocols: AH (Authentication Header) [4], which provides authentication and integrity, and ESP (Encapsulating Security Payload) [5], which provides not only the former services, but also confidentiality. Implemented as two extra headers inserted after the IP's, AH and ESP can be used separately or together.

The main functionality of each header is related to the use of secret-key cryptographic algorithms [6], [7]. However, a basic set implementation is mandatory, despite the IPSec specification being independent of any cryptographic algorithm

Jansen Sena is sponsored by CNPq

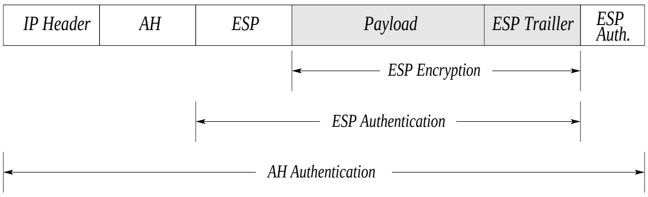

Fig. 1. Protection covering provided by both the AH and ESP protocols.

in particular. This is justified to maintain a minimum level of compatibility among various implementations, as demonstrated below:

- *HMAC-MD5-96* and *HMAC-SHA-1-96*: authentication and integrity algorithms used by AH and ESP;
- *DES-CBC*: ciphering algorithm user by ESP;
- Null authentication and ciphering algorithms used by ESP[1].

The difference between the authentication and integrity services done by AH and ESP is which packet parts are protected, as is shown in Figure 1. AH protects all packet fields, except the fields where values are altered in transit. When offered by ESP, these services cover its own header and the packet payload only, in order to protect the packet against possible attacks such as those identified in [8].

A. Security Associations

Before two hosts exchange packets safely, a set of parameters must be established, such as security protocol, operation mode, cryptographic algorithm, key value, key lifetime, key length and packet replay protection data. This set of parameters is called Security Association (SA) [1], [2], [3].

A host can establish many security associations with other hosts. This level of granularity must be defined by the system administrator: a generic policy, host-oriented, could protect all traffic between two hosts using only one security association. In counterpart, a more specific policy could require one security association for each open session between a local host and any other host. Finally, an intermediate policy would require one security association for each type of traffic (FTP, DNS etc). Other important factor to consider is that security associations have a single direction and can only use one security protocol (AH or ESP). Thus, it is possible that traffic between two hosts may have different protection levels for each direction.

The security association establishment can be static or dynamic. In the former case, all parameters must be manually supplied by the system administrator. This method not only limits the IPSec protection but also can result in communication impairing errors due to high human intervention. In the latter case, the security association is established through the IKE protocol (Internet Key Exchange) [9], [10], without any intervention by the system administrator, which represents an appropriate solution to make IPSec operate automatically. In this process, the initiator in charge of starting the security association establishment process sends a predefined parameter proposal, including a list of cryptographic algorithms, in order of preference, to the other end of the communication, defined as the responder.

When the responder receives the initiator's proposal, the responder must select the desired parameters according to its policy and send its choices back to the initiator. If the parameters received by the responder do not match its security policy (e.g. none of the proposed algorithms is acceptable to protect this type of traffic), the security association establishment can be rejected. This makes it effectively impossible for the two hosts to exchange data related to specific service that the security association intended to protect. The proposals made by the initiator and the values accepted by the responder are part of the IKE configuration process.

B. SPD and SAD

In order to submit any packet to IPSec protection, it is necessary to check it against rules contained in a repository called SPD (Security Policy Database) [1]. These rules consist of selectors that identify IP addresses, source and destination ports, transport protocol (TCP, UDP etc), and other parameters. SPD rule configuration is the responsibility of the system administrator.

Once a packet is matched by one of the SPD rules, it must be protected according to the

[1] These algorithms are necessary because authentication/integrity and confidentiality are optional services in the ESP header. However, these two null algorithms cannot be used together in the same packet [5], [2].

parameters found in SAD (Security Association Database) [1], a structure that stores all the active security associations of a host. If the security association method establishment is static, the SAD entries must be supplied by the system administrator. Otherwise, the IKE protocol will manage the SAD entries.

III. Security Level Model

Despite IPSec's ability to solve many of the problems detected in IP along the years, its high complexity and flexibility are, no doubt, limiting factors regarding its use in general environments [11]. Using IPSec without adequate knowledge of its numerous options can pose a risk to the system security [2].

In the current status of the IPSec specification, this protocol represents a feasible and low cost solution to create static and pre-defined environments, such as the traditional VPN (Virtual Private Networks) [12]. However, despite supporting such use, its utilization in scenarios where the traffic between any two entities must be adequately protected is not so simple, when considering large networks. Besides, it is necessary to have well-defined security policies and maintain the least compatibility between them in order to provide such characteristics. All these requirements might turn into a very complex task to the system administrator.

In the attempt to ease IPSec's utilization and, consequently, the creation of these scenarios, SLM (Security Level Model) is therefore proposed. It is based on high-level specifications and generates policies and parameters for use by SPD and IKE.

A. Security Levels

Fundamental to SLM, security levels are sets of parameters necessary to establish security associations [13], [14], including: security protocols, cryptographic algorithms, key lengths and security association lifetime. The specific parameters of each security level will define the effective protection provided by each level.

Four levels, organized in ascending order according to their protection degrees, were defined for SLM. These were defined based on the authentication, integrity and confidentiality services provided through different cryptographic algorithms and other IPSec parameters. The security level specifications are stored in a database called SLD (Security Level Definition) and the description of each level is presented below:

- *Unclassified*: only guarantees authentication and integrity for short periods of time. From a security point of view, information gathered from packets under Unclassified protection must not present any risk. The ciphering service should not be used inadvertently in order to avoid a decrease in performance, given the protection offered by this level. It is the only one to exclusively use AH services;
- *Confidential*: provides authentication, integrity and confidentiality to packet contents for a short period of time, guaranteeing its delivery and secrecy until its information is processed by the corresponding application. However, it is essential that the packet content sensitivity ceases after its use;
- *Secret*: protect packets through authentication, integrity and confidentiality services for a reasonable period of time. The information sensitivity in this level goes beyond its transmission and processing time. Therefore, its security must be protected for a time greater than that guaranteed by the previous level;
- *Top Secret*: provides authentication, integrity and confidentiality for an undetermined period of time. The information protected by the Top Secret level presents potential risks if maliciously handled at any time during and after its transmission.

After getting familiar with the levels and its descriptions, it is necessary to associate each service that is to be protected to a security level compatible with its security requirements[2]. These associations are stored in a database defined as Protected Services. For example, Table I shows a possible service classification. It is important to note that the main objective of the security levels is to abstract degree of protection from specific parameters and to create security policies based on security level specifications only.

Figure 2 depicts the flow of data in an entity using SLM. The services associated to the model have their packets protected automatically accord-

[2] In this context, security requirements represent the need for authentication, integrity and/or confidentiality and in which degree of protection [13], [14].

Security Level	Services
Top Secret	Telnet, SSH, FTP, POP3, HTTPS, SMTP, SNMP
Secret	DNS(*zone transfer*), NNTP, Syslog, LDAP
Confidential	FTP-DATA, HTTP, BOOTP, TFTP
Unclassified	DNS(*query*), Time, Daytime, Echo, Finger

TABLE I

Example of common services distribution in the SLM security levels.

ing to the parameters of their corresponding security level. It is important to observe that the processing of packets unrelated to the model remains the same: without IPSec protection or protected through specific policies, external to the model, inserted by the system administrator.

A.1 Cryptographic algorithms

The main security level components are the cryptographic algorithms and their respective key lengths. Despite both SLM and IPSec not being restricted to a fixed and limited cryptographic algorithm set, an initial attribution is necessary for SLM to work.

Table II presents a proposal using well-known, extensively analyzed algorithms in descending order of preference. The discovery of vulnerabilities in an algorithm can result in its relocation to a lower security level or even its exclusion from the model, depending on the gravity of the problem.

In this proposal, there are ciphering algorithms with the same key size distributed in the Secret and Confidential security levels. Notice that a 128-bit key size provides protection guarantee higher than that of the Confidential description. The grouping of algorithms with such parameter in the same level of DES, which has only 56-bit key size and is certainly more vulnerable to brute-force attacks, is apparently inconsistent. However, the computational cost to cipher messages with DES, in most implementations, is higher than that with other, larger key size algorithms. On the other hand, the use of DES in the Confidential level is justified by the fact that its protection is enough and in accordance with the security requirements of this level. Furthermore, implementations not having any other preferential algorithm can use DES to protect services classified in this security level.

B. Basic working

Having defined the security levels and finished the services association, the model then requires which service and which interaction mode (client, server or both) are to be supported by each host. For example, a common entity could use HTTP, FTP, SMTP and POP3 as a client and SSH as a client and server. Such information is stored in a third database defined as Host Services, where there must be entries corresponding to every host that will have its traffic protected by the model. This database is the main source of interaction of the system with the system administrator, who must keep it updated according to the applications running in the network hosts. Such task can be done through the SLI (Security Level Interface), another SLM component, which is a tool for querying and maintenance of the Host Services information.

At the moment the SLM protected system is booted, a component called SLC (Security Level Converter) searches for the correspondent host entry in the *Host Services* database and fetches the host services and their respective interaction modes. Next, SLC queries the Protected Services database to obtain the security level of every service supported for this host. Finally, SLC goes to the Security Levels Definition database and fetches the specific security parameters to be applied for each defined level. Based on this information, SLC generates the SPD security policies and the IKE protection suites, which are finally loaded into the system.

Figure 3 presents the general functioning model of the SLM. There is can be seen that the end result of using the SLM model is that an abstraction layer is inserted above the specific parameters of IPSec components. Starting with high-level, platform-independent definitions, the platform-dependent IPSec parameters are created without any intervention of the system administrator.

Before transmitting packets corresponding to a service protected by a security level, the IKE protocol will start the security association establishment process. It proposes to the other end of the communication a set of cryptographic algorithms

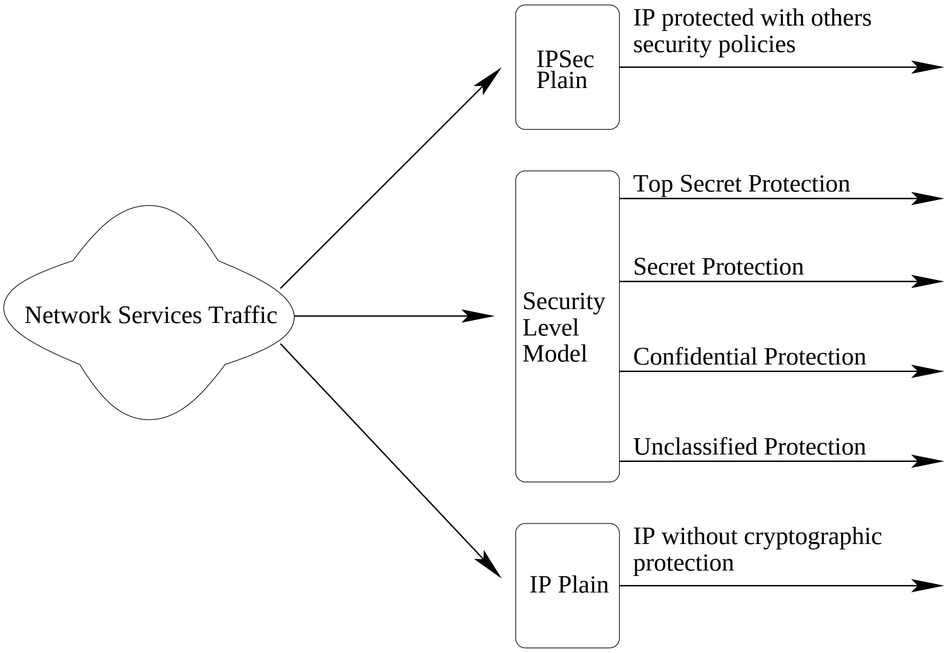

Fig. 2. Entity traffic protection using SLM.

Security Level	Authentication		Ciphering	
	Algorithm	Key size	Algorithm	Key size
Top Secret	HMAC-SHA2-512-96	512	Rijndael Twofish Serpent Cast Blowfish	256 256 256 256 448
Secret	HMAC-SHA2-384-96 HMAC-SHA2-256-96	384 256	Rijndael Twofish Serpent Idea Blowfish Cast 3DES	128/192 128/192 128/192 128 192 128 168
Confidential	HMAC-SHA1-96 HMAC-RIPEMD160-96	160 160	Rijndael Twofish Serpent Idea Blowfish Cast DES	128 128 128 128 128 128 56
Unclassified	HMAC-MD5-96	128	Not applicable	

TABLE II
Cryptographic algorithm distribution proposal into security levels.

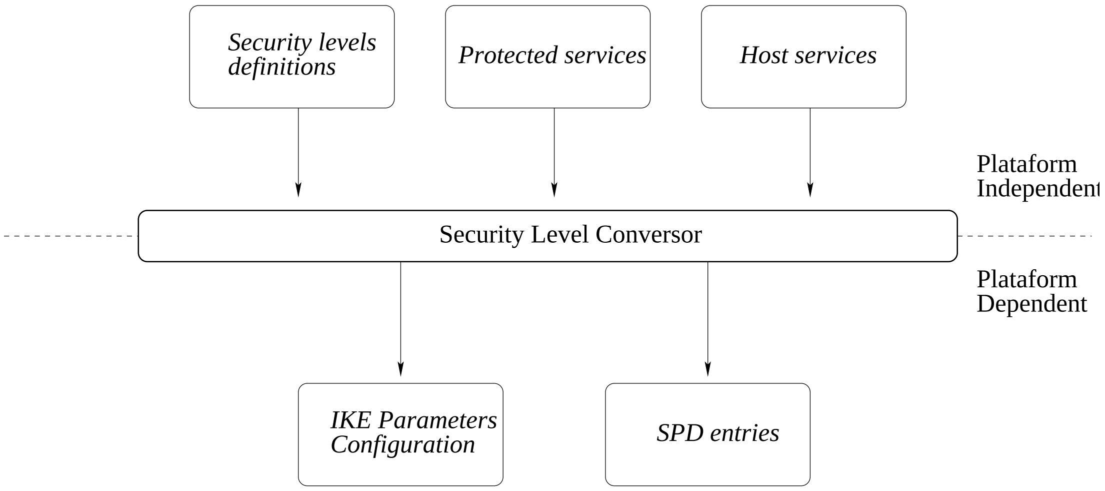

Fig. 3. Abstraction scheme provided by SLM.

arranged according to preference settings in its security level. Notice that the model ensures that the protection applied to any traffic is the same in both directions, not allowing one direction to be more vulnerable than the other. This feature is guaranteed by the way the policies are generated from the SLC.

IV. Implementation

The SLM runs under the IPSec implementation maintained by the KAME project, using FreeBSD.

The three databases of SLM are implemented through class objects defined for the model in an LDAP (Lightweight Directory Access Protocol) server [15]. LDAP is a protocol capable of centralizing information and distributing them to a set of entities according to its parameters, which were previously defined. The arrangement of these objects are based on a hierarchical tree whose structure is shown on Figure 4. On the left portion are grouped class objects from the Security Level Definition and Protected Services databases, which host general protection parameters that should serve as a reference to all hosts protected by SLM. On the right portion are grouped class objects from the Host Services database, containing the specific parameters for each host, that is, its services and interaction modes.

Data stored in LDAP for the SLM model is handled by SLC and SLI, both written in Perl. SLC queries the LDAP server and fetches the information necessary to create the SPD policies, loaded onto the system through the *setkey* utility, and the IKE protection suites, used by the *racoon* daemon, which is the KAME project implementation for IKE. The system administrator uses SLI for maintenance of the Host Services class objects.

By default, the security policies created are based on IPSec's transport mode. However, a few parameter changes are needed to create policies in tunnel mode, allowing for the packet exchange protected by SLM between the two IPSec gateways.

V. Conclusions

The complexity and the various possible uses of IPSec can make its use unattractive for the protection the common services on a network.

On one hand, general protection policies can either put a system security at risk or insert unnecessary extra processing to packets, since all traffic is protected according to the same security policy. On the other hand, many specific policies require high maintenance cost. It can also generate parameter consistency problems, which can result in communication failures.

In this context, SLM was developed in an attempt to facilitate the IPSec utilization by hiding the specific protocol parameters through the use of abstracted security levels, which provide different protection degrees, and through the service association to these levels. By being based on high-level, implementation-independent structures, SLM enables much better and efficient use of IPSec, summarizing the process to the configuration of the protections required for the intended data traffic

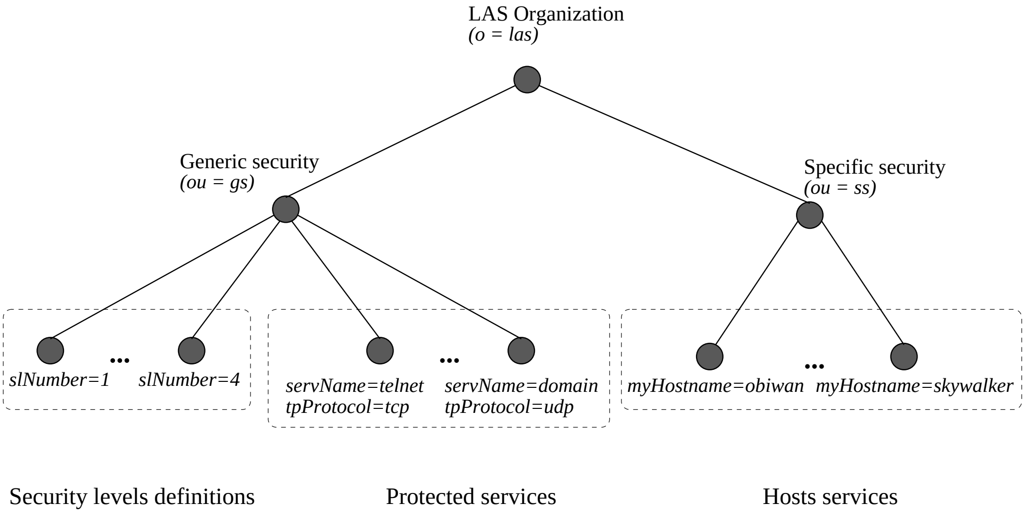

Fig. 4. Hierarchical tree and classes defined in LDAP to represent the SLM databases.

categories for each host on the network.

VI. Future Work

A desirable extension to this work is to make SLM multi-platform compliant, enabling it to generate security parameters and protection suites for other IPSec implementations, such as FreeS/WAN for Linux.

References

[1] S. Kent and R. Atkinson, *Security Architecture for the Internet Protocol*, Internet Engineering Task Force, RFC 2401, 1998.

[2] Sheila Frankel, *Demystifying the IPsec Puzzle*, Artech House, Norwood, Massachusetts, 2001.

[3] Uyless Black, *Internet Security Protocols*, Prentice Hall, Upper Saddle River, New Jersey, 2 edition, 2000.

[4] S. Kent and R. Atkinson, *IP Authentication Header (AH)*, Internet Engineering Task Force, RFC 2402, 1998.

[5] S. Kent and R. Atkinson, *IP Encapsulating Security Payload (ESP)*, Internet Engineering Task Force, RFC 2406, 1998.

[6] Bruce Schneier, *Applied Cryptography*, John Wiley Sons, New York, 2 edition, 1996.

[7] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Confirmar, 2 edition, 1996.

[8] Steven M. Bellovin, “Problem Areas for the IP Security Protocols,” in *Proceedings of the Sixth Usenix UNIX Security Symposium*, San Jose, California, 1996.

[9] D. Harkins and D. Carrel, *The Internet Key Exchange (IKE)*, Internet Engineering Task Force, RFC 2409, 1998.

[10] D. Maughan and et al. M. Schertler, *Internet Security and Key Management Protocol (ISAKMP)*, Internet Engineering Task Force, RFC 2408, 1998.

[11] Niels Ferguson and Bruce Schneier, “A Cryptographic Evaluation of IPsec,” Tech. Rep., Counterpane Internet Security, Inc., San Jose, CA, USA, 2000.

[12] Elizabeth D. Zwicky, Simon Cooper, and D. Brent Chapman, *Building Internet Firewalls*, O’Reilly Associates, Sebastopol, California, 2 edition, 2000.

[13] Jansen Carlo Sena and Paulo Lício de Geus, “Um Mecanismo para Estabelecimento de Associações de Segurança Baseado em Categorias de Serviço,” in *III Simpósio Segurança em Informática*, São José dos Campos, São Paulo, Brazil, 2001, pp. 193–202.

[14] Jansen Carlo Sena and Paulo Lício de Geus, “Uma Ferramenta para Proteção do Tráfego de Serviços Utilizando o IPSec,” in 20^{o} *Simpósio Brasileiro de Redes de Computadores, WSeg2002 (Workshop em Segurança de Sistemas Computacionais)*, Búzios, Rio de Janeiro, Brazil, 2002.

[15] Tim Howes, Mark Smith, and Gordon Good, *Understanding and Deploying LDAP Directory Services*, NewRiders Publishing, 1998.