Current IPsec Policy Configuration Options

NAILabs

September 15, 2003

The Marriam Webster's dictionary contains the following definition for the term "Policy":

> A high-level overall plan embracing the general goals and acceptable procedures especially of a governmental body.

"Policy" is a very broad term that often has different meanings depending on the context in which it is used. A recent trend in network management has led to the attempted standardization of the term "Policy" when used in a network management context so it describes how a collection of network devices are intended to operate when properly configured. "Policy Management" has then been used to describe how a network management station may be given policy definitions to be implemented, possibly in a form easily understood by a human administrator, and the process needed to convert those high-level definitions into a machine-usable form that can be transmitted to the various network devices responsible for implementing and enforcing that given set of policies.

The two most important policy definition types being developed today describe how to define policy for devices that control Quality of Service (QoS) and for devices that control the authenticity and privacy protection levels of transmitted data via the IPsec security protocol. The definitions for Quality of Service policies are older and somewhat more mature, while the policy definitions for controlling transmission security are relatively new and have not yet been implemented by most vendors. This is not to say that the underlying functionality of devices cannot enforce common requirements of QoS and security related policy, the common requirements are simply not known. The requirements and mechanisms for both types of policy need to be more thoroughly understood and specified before commonality can be identified.

Although the term "Policy" implies that overlap occurs between the different types of policy realms, this is not the case in the specifications and early implementations currently underway. For example, the QoS and IPsec policy infrastructures overlap in that they both classify network traffic and process it in a manner specific to each type of policy. This traffic classification process could be designed in a way that could be used by both the QoS and the IPsec policy implementations. Currently, however, there is not enough consensus on the various requirements and specifications to achieve commonality in implementations. Standardization bodies, like the IETF and the DMTF, are beginning to correct this problem by developing a policy architecture framework that can provide a common basis for building multiple, more specific, policy infrastructures for various functions such as QoS and transmision security. The work produced by these bodies, however, is in very early stages of development and no implementable results have been produced yet. In addition, previously existing policy definitions, such as the older QoS definitions described above, will need to be re-designed in order to take advantage of the new shared, commonality provided by the policy framework. Reuse between the overlapping portions of the various policy realms is not possible until this common functionality has been defined in a standardized manner.

1 IPsec Policy

The Internet Security Protocol (IPsec) [1] is an Internet Engineering Task Force (IETF) [2] standardized protocol for ensuring that network traffic between hosts and/or subnet pairs is appropriately authenticated and/or protected from disclosure. It is widely deployed and most frequently used by organizations wishing to secure communications between physically separate sites that transmit potentially sensitive network traffic over an otherwise unprotected network, such as the Internet, as shown by the communicating sites A and B in Figure 1.

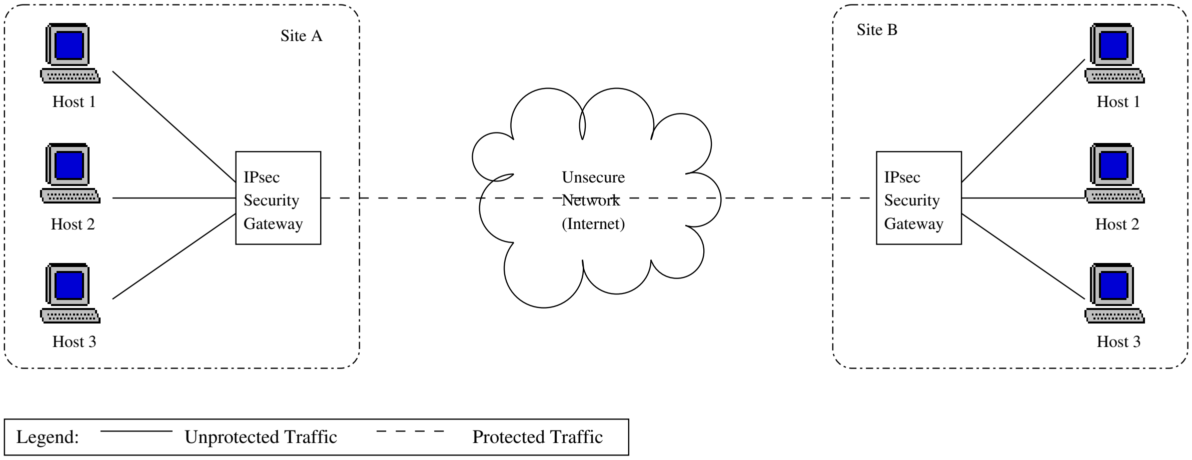

Figure 1: Typical Current Usage of IPsec.

IPsec is also used by some organizations to secure communications internally over their local networks as well, providing complete end-to-end security. It is additionally in use by independent, but collaborating, organizations to secure communications between themselves. Another common usage of IPsec is to provide remote access to an organization's internal network through the use of IPsec connections between a remote system and the internal network, as shown in Figure 2.

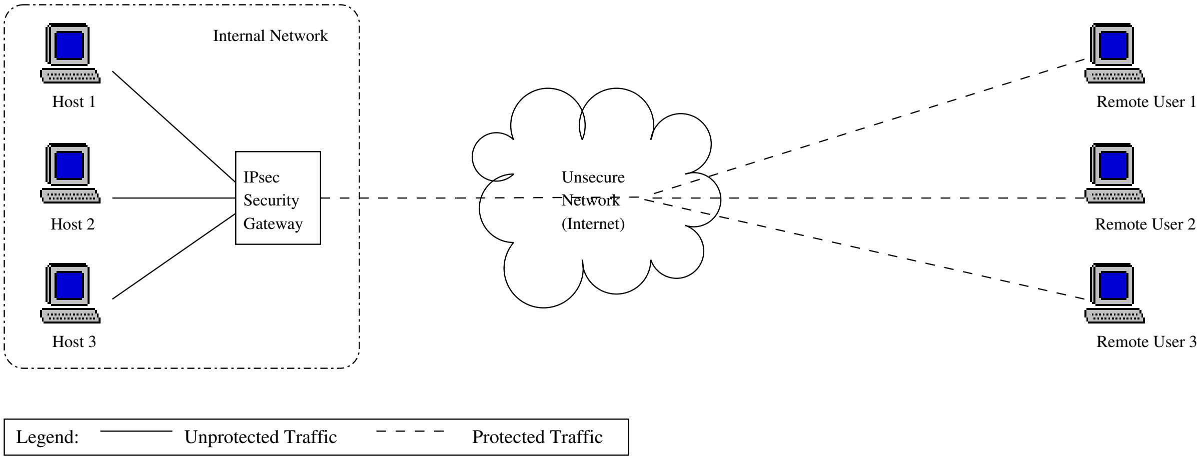

Figure 2: Remote User Access Using IPsec.

Recent trends have begun to further expand the scope of IPsec usage by attempting to establish IPsec negotiations between any remote network nodes willing to communicate via IPsec, regardless of whether they have a prearranged agreement to do so. This is done in an attempt to secure as much communication over the internet as possible.

The use of IPsec on a device requires configuration of policy based information so that the device meets and properly enforces the requirements of the policies imposed by the external administrative body. For example, the administrator of a device must determine which traffic must be protected by which type and level of service and the time periods that the various policies will be enforced based on the needs imposed by the organization's administration. Ideally, the policies enforced by IPsec capable devices should exactly implement higher level security policies established by that organization. In situations where organizations do not have administratively established security policies, the IPsec policy related information may be the catalyst for the organization to establish needed security policies. Although organizational security policy will likely increase in importance as more organizations establish IPsec relationships, this task is focused on IPsec policy related information rather than organizational security policy, thus, the remainder of this report will focus on configuring and managing IPsec policy related information.

Configuring an IPsec device to appropriately protect traffic according to the requirements of these management policies can be a complex and time-consuming task. This will become increasingly difficult as IPsec negotiations are pushed closer to the end user, thus increasing the number of IPsec devices that must be properly configured. This document describes existing technologies which can configure large scale deployments of IPsec-enabled devices, and whether these technologies support the configuration of IPsec-enabled devices produced by multiple vendors.

2 Current Standards Based Solutions

Currently, no existing standards have been defined that enable IPsec devices to be configured in a vendor-independent manner. Some initial work toward such a standard has begun in the IETF but the work is far from producing final specifications. The Internet Security Policy Working Group (IPSP) [5] has begun work on specifying mechanisms needed for IPsec devices to negotiate IPsec policy related options. The Policy Framework Working Group [6] has been tasked with specifying a broad policy framework that can be used throughout the IETF with the first usage example being an update to the quality of service provisioning models. Much of the work from the Policy Framework WG is derived from and related to work being done in the Distributed Management Task Force (DMTF) [3]. The DMTF is defining a Common Information Model (CIM) [4] that is intended to be used to model a wide range of capabilities.

Since the Policy Framework WG products are to be used throughout the IETF, the IPSP WG is now tasked with developing an information model [7] that is in line with the framework specifications developed by the Policy Framework WG [8]. Additionally, the Configuration Management with SNMP WG (SNMPConf) [9] is developing specifications intended for policy management of systems utilizing SNMP. Although the endpoint of IETF working groups is often difficult to predict much in advance of a WG reaching that endpoint, both the IPSP WG and the SNMPConf WG have indicated a strong interest in including support for policy management of IPsec devices.

2.1 Current Policy Provisioning Protocols

Policies that are to be applied to a set of network devices must be transmitted over a standardized protocol to enable multi-vendor environments to inter-operate appropriately. There are at least three protocols being defined by various IETF working groups that accommodate the need for standard protocols: LDAP, SNMP, and COPS. LDAP is often used as a policy storage database that SNMP and COPS managers query to determine which policies need to be applied via SNMP or COPS.

2.1.1 LDAP

The Lightweight Directory Access Protocol (LDAP) [11] is a simplistic network-searchable database in which data may be kept in a record-oriented fashion. A definition has been written for mapping the high-level policy schema produced by the IETF Policy Framework Working Group into an LDAP schema [12]. This work, however, has only been started recently and has not been implemented in any client to date and commercial implementations that make use of LDAP do not use a standardized LDAP schema. The LDAP schema defined by the IETF is currently very high-level and needs to be extended to provide additional mappings for each policy type (QoS, IPsec, etc.) before it can be used in a functional manner. No work has been done to begin these needed definition extensions.

Some important security considerations must be dealt with when considering the use of LDAP as described by the Policy Framework working group: no security mechanisms have been defined to insure data source authenticity and integrity. Only the issue of securely transmitting the data has been defined using LDAP and the authenticity of the data itself, once received, can not be verified utilizing the current standards work.

2.1.2 SNMP

The Simple Network Management Protocol (SNMP) is an IETF protocol designed to manage and configure remote network devices. The data transmitted over the SNMP protocol is defined in documents external to the protocol definition itself and many such definitions have been defined enabling the SNMP protocol to configure and manage everything from the fundamentals of how low level network devices inter-operate to how applications running atop end-systems operate.

The earliest versions of the SNMP protocol contained no mentionable security and was used primarily only for data collection. The third version of the protocol, SNMP version 3 [13], was designed to be secure, modular and to support fine-grained user-based access control over a device's configuration parameters. As of this writing it is in the last stages of becoming declared a "full standard" by the IETF.

2.1.3 COPS

The Common Open Policy Service (COPS) [14] [15] is another IETF protocol developed in the last few years in an effort to solve some of the perceived problems with the SNMP protocol. It was designed with the intent to significantly simplify the implementations in the distributed clients.

COPS is a light weight connection oriented protocol designed to transmit collections of related data as a group. It was designed as a solution for distribution of policy configurations, but is a generic protocol that could be used for more general system management. The definitions for the data to be transmitted via the COPS protocol are defined in documents external to the protocol definition itself. The most common use for the COPS protocol to date has been to implement quality of service based policy configuration mechanisms.

2.1.4 Comparing COPS and SNMP

When deploying a policy management infrastructure, one must consider which protocol one intends to use when managing a network. The choice is not an easy one to make and there are advantages and disadvantages to both the COPS and SNMP protocols. Many of the IETF working groups are defining models that can use both COPS and SNMP.

This section discusses some of the differences between COPS and SNMP but does not attempt to draw conclusions as to which is a better protocol for policy provisioning. It should be noted, however, that most devices that implement COPS will also implement SNMP as well, since COPS was designed solely for policy distribution and more generic network configuration management was not a design goal of COPS.

2.1.4.1 Management Style

Under the COPS design methodology, a network device can only be controlled by one COPS management station at any given time. This design decision was made to help simplify the implementation within the deployed network devices. However, this makes it impossible to use in situations where inter or sub-organizational administration is needed. The complex architecture described in Section 4 and depicted in Figure 3, for example, could not be implemented using COPS if organization number 1 was expected to communicate policy information with organization number 2 or if a third parent organization was responsible for ensuring a global policy was being properly implemented within both sub-organizations. On the other hand, SNMP provides fine-grained access control mechanisms that can handle defining realms of administrative responsibility. The advantage COPS gains in it's management methodology, however, is that the implementation in the deployed network nodes is fundamentally simpler.

2.1.4.2 Connection Oriented vs Connection Less Communication

COPS was designed in such a way that it is the deployed network nodes' responsibility to initiate a conversation with its manager making use of the connection oriented TCP internet protocol. A connection oriented approach was chosen so that larger size packets could more easily be transmitted. This design has the drawback of not working well in poorly operating networks where packet loss is more likely to occur. COPS, by design, is expected to be used on top of an already functional network and takes advantage of this methodology to help it gain some improvements over SNMP in both speed and simplicity. However, the COPS design does not appear to support distribution of a policy which was expected to fix existing problems in a poorly operating network. Since COPS was defined, a new document defining SNMP usage over the TCP protocol has been written that specifies how SNMP can be used over TCP when the network was performing well and still be able to fall back to the more reliable connectionless UDP protocol when the network was operating in a lossy state.

2.1.4.3 Packet Size

COPS packets are designed to provide a reduction in the average packet size compared to the average packet size of an SNMP transmission when performing large scale policy updates. In general in these cases, transmitted COPS packet sizes will be less than those of the identically performed operation using SNMP. The Evolution of SNMP IETF Working Group [16] is considering solutions to this problem within the SNMP realm, but are only just beginning to study the problem. It is possible that there are situations where SNMP would be more efficient, such as performing minor updates to a previously deployed policy set. To date, no study has been done that analyzes an existing operational network in order to prove that one methodology should be preferred over the other.

2.1.4.4 Authentication, Privacy and Access Control

Both COPS and SNMP implement authentication within their basic protocol definitions. SNMP version 3's authentication mechanism is implemented using a message authentication hash utilizing a shared secret key. SNMP's authentication scheme is user-based, with each user of the protocol having their own shared secret. Additionally, each secret is "localized" to a given SNMP agent such that if one agent is compromised, the keys stored there can not be used to gain access to other near by agents. SNMP also provides mechanisms for confidentiality and privacy as well as providing fine-grained access control methods.

COPS authenticates its transmissions using a message authentication hash utilizing a shared secret between the management station and the remote COPS-enabled devices. COPS has no access control methods and gives the COPS management station full access to all of its data upon successful verification of a message's authentication. COPS relies on the administrator to properly protect the transmissions from disclosure using a TCP protection suite such as one of the SSL, TLS [17], or IPsec protocols.

Both COPS and SNMP's authentication techniques are relatively equivalent with respect to cryptographic strength, assuming the default authentication mechanisms are used. Only SNMP, however, implements any form of authorization control beyond the initial authentication. Both COPS and SNMP require implementations to support a MD5 base authentication mechanism, but the SNMP protocol additionally defines and encourages the implementation of a SHA-based authentication mechanism as well.

3 Current Policy-Based Software Solutions

Because no standardized mechanism exists for configuring IPsec-enabled devices, all existing software solutions are only capable of controlling and configuring a single vendor's IPsec-enabled devices.

3.1 Freely Available Solutions

Only a few IPsec implementations exist for the widely used and freely available Linux and FreeBSD / OpenBSD / NetBSD operating systems. No scalable management solutions exist for these architectures.

3.1.1 Freely Available IPsec Solutions

There are three IPsec software solutions being actively developed and maintained for use in freely available operating systems such as linux, and Free-BSD / OpenBSD / Net-BSD. These solutions are shown in Table 1.

IPsec Implementation	OS Type
KAME	BSD
FreeS/WAN	Linux
Cerberus	Linux

Table 1: Freely Available IPsec Solutions

3.1.1.1 KAME

KAME [18] is a freely available IPsec and IPv6 implementation for the freely available BSD operating systems: FreeBSD, OpenBSD, and Net-BSD. It began as a joint project between 7 different companies based in Japan [19] to introduce both IPsec and IPv6 to the BSD related operating systems. The project is still on going and the current project timeline indicates completion in March 2002, although the IPsec part of the project is considered to be mostly complete today.

To date, there is no known interface for managing KAME nodes using a simplified policy-based interface. All configuration in KAME must currently be done through the use of command line tools.

3.1.1.2 FreeS/WAN

FreeS/WAN [22] is a freely available IPsec implementation for the Linux operating systems. It is the most popular of the two IPsec implementations available for Linux and a large number of open source developers are contributing to the project and offer an excellent support base to its user population. FreeS/WAN supports a couple of interesting features as well:

First, their documentation extensively describes how to configure FreeS/WAN to speak with remote roaming users, or "Road Warriors" as the documentation refers to them, whose IP addresses may by dynamically assigned or variable, thus making authenticity and policy control based on IP addresses impossible. The configuration of this feature must be done by hand and supporting even a small number (hundreds) of “Road Warriors” will be difficult for a network administrator making use of FreeS/WAN.

Second, FreeS/WAN has preliminary support for what they refer to as “Opportunistic Encryption”, which instructs the FreeS/WAN implementation to attempt IPsec negotiations with anyone it can retrieve IPsec keys via secure DNS. Support for this is underway and is functional, but not usable by end-users yet. This powerful feature will need to be carefully integrated with the security policies of organizations that mandate a minimum security service level when communicating with certain sites. Additionally, this could trigger IPsec connections to other organizations which were not expecting or prepared for IPsec communication with unknown entities outside their local configuration setup.

There are a couple of downsides to the FreeS/WAN IPsec/IKE implementation. First, it is not a fully compliant implementation as it does not support DES encryption nor does it support IKE notifications, among other things. These features are not missing because no one has had time to write the code, but rather because the lead developer does not want them to be implemented within the code base.

The lead developer is a U.S. citizen who has out-sourced all of the actual software development to citizens of other countries in an effort to make a political statement about the current U.S. exportation laws related to cryptographic software. He also refuses to accept patches from U.S. citizens for the same reason, and does not participate in the development of the code himself, acting in more of an administrative role. For this reason, the distribution site is in Canada and most development takes place there as well.

3.1.1.3 Cerberus and PlutoPlus

Cerberus [23] is a freely available IPsec reference implementation for Linux distributed by NIST [24] . Cerberus is no longer being developed, as the implementation is considered complete by the original author. Unfortunately, this also means that the code has not been ported to the most recent versions of the Linux operating system kernel.

PlutoPlus is the IKE implementation which interacts with Cerberus, and is still under development with a projected completion date of September, 2001. It is currently functional and supports many of the IKE features that FreeS/WAN lacks. However, it does not implement some of FreeS/WAN’s other useful features such as public key retrieval using secure DNS. The currently available stable release is significantly behind the development code. For example, the stable release does not support public key certificates. The next release containing this functionality is supposedly forthcoming.

Currently both Cerberus and PlutoPlus are only available by request, as the maintainers of the Cerberus web site have not yet made the software available under the new U.S. exportation license. Work has supposedly begun to make this package more easily obtainable over the web but has yet to be reflected on the main Cerberus web site.

Past development of Cerberus was done by primarily one person at NIST and the current development of PlutoPlus is currently being done by a different developer at NIST. The distribution of both Cerberus and PlutoPlus is handled by NIST personnel.

3.1.2 IPsec Configuration Solutions for Freely Available Operating Systems

Currently, no freely available solutions exist that can manage a large number of IPsec clients. Only one freely available graphical configuration interface exists and is listed in Table 2.

IPsec Implementation	Configuration Management Package	Nodes Supported	Ref
FreeS/WAN	webmin	1	[22] [10]

Table 2: Freely Available IPsec Configuration Management Packages

3.1.2.1 webmin

The webmin [10] IPsec configuration plugin [20] can be used to configure the FreeS/WAN [22] IPsec implementation, but is limited in scope and can only be used to configure a single device at a time. It is simply a web based GUI interface capable of viewing and editing the FreeS/WAN configuration files on a host. Because it is designed to manipulate a FreeS/WAN configuration file, and it's user interface is tailored to this configuration structure, it is therefore not a multi-vendor solution. It does not provide any ability to handle configuration of multiple devices nor does it make it possible to easily manage complex sets of policy filters. These burdens are left to the administrator. Additionally, it requires that a web server be running on every managed device which would not be viable for many IPsec devices. Unfortunately, these limitations prevent webmin from being usable as a scalable IPsec policy management solution.

3.2 Commercially Available Solutions

Commercial support for IPsec is widely available. Many companies produce both IPsec gateways, which can be used to protect traffic leaving any given organization or a segment of that organization, and IPsec end-host software, which can be used to protect traffic to its final destination. However, the current recommended practice is to use a single vendor solution [27] [28] when designing and deploying an IPsec infrastructure to ensure that manageability of complex IPsec policy definitions is possible. End-host support for IPsec communication is also widely available in modern operating systems, such as Windows2000 and Solaris.

Security policy configuration management of all of these devices is currently done in an implementation-dependent manner. Many vendors provide configuration management suites capable of managing a large network of IPsec-enabled devices as a separate add-on product. All of these configuration management packages, however, are vendor specific in nature and cannot be used as a generic vendor-independent IPsec policy configuration management tool. A partial list of IPsec configuration management software provided by leading IPsec vendors is shown in Table 3. A complete list of products can be found at the Virtual Private Network Consortium's web site [26] and at standard web search engine directories like Yahoo [25].

Company Name	Configuration Management Package	IPsec Implementations Manageable With It	Nodes Supported	Ref
RedCreek	Redcreek e-Director	All Ravlin Products	unlimited	[29]
Network Associates	GEMS	Gauntlet firewall	500	[30]
Assured Digital	ADI Management Server	ADI VPN Devices	N/A	[31]
Alcatel	Secure VPN Management Suite	Alcatel systems	10 million	[32]
VPNet	VPNmanager Series	VPNware systems	unlimited	[33]
Lucent	Lucent Security Management	Lucent Secure VPN	1000 subnets 20000 nodes	[34]

Table 3: Commercially Available Configuration Management Packages

4 Summary

IPsec usage is expanding in numerous ways, all of which will make administration of IPsec-enabled devices more complex.

First, remote roaming users (possibly running different IPsec implementations) are attaching to current IPsec infrastructures. Each of these remote users and the IPsec infrastructures to which they are attached must be managed to ensure required policies are being adequately enforced on both ends of the connection. Currently no software or methodologies exist to manage this complex task in a multi-vendor environment.

Second, as organizations begin to inter-communicate using IPsec they will each have policy sets which need to be configured and may overlap with parent organization's policy sets as well. This IPsec peering will greatly increase the complexity of current configuration management needs. The software which exists today is only capable of managing devices of like types and is insufficient when managing multiple organizational structures with independent networking architectures. This is further complicated by new features of some implementations that attempt to negotiate IPsec communications whenever keys are available.

Finally, local administrators of a network segment are frequently granted control over certain aspects of the local network's configuration, as long as they adhere to the parent organization or administrator's policies. A fine level of access control must be carefully constructed to ensure that the local administrators are given the exact amount of control alloted to them. No current policy-based standard or software solution exists that makes this easily possible.

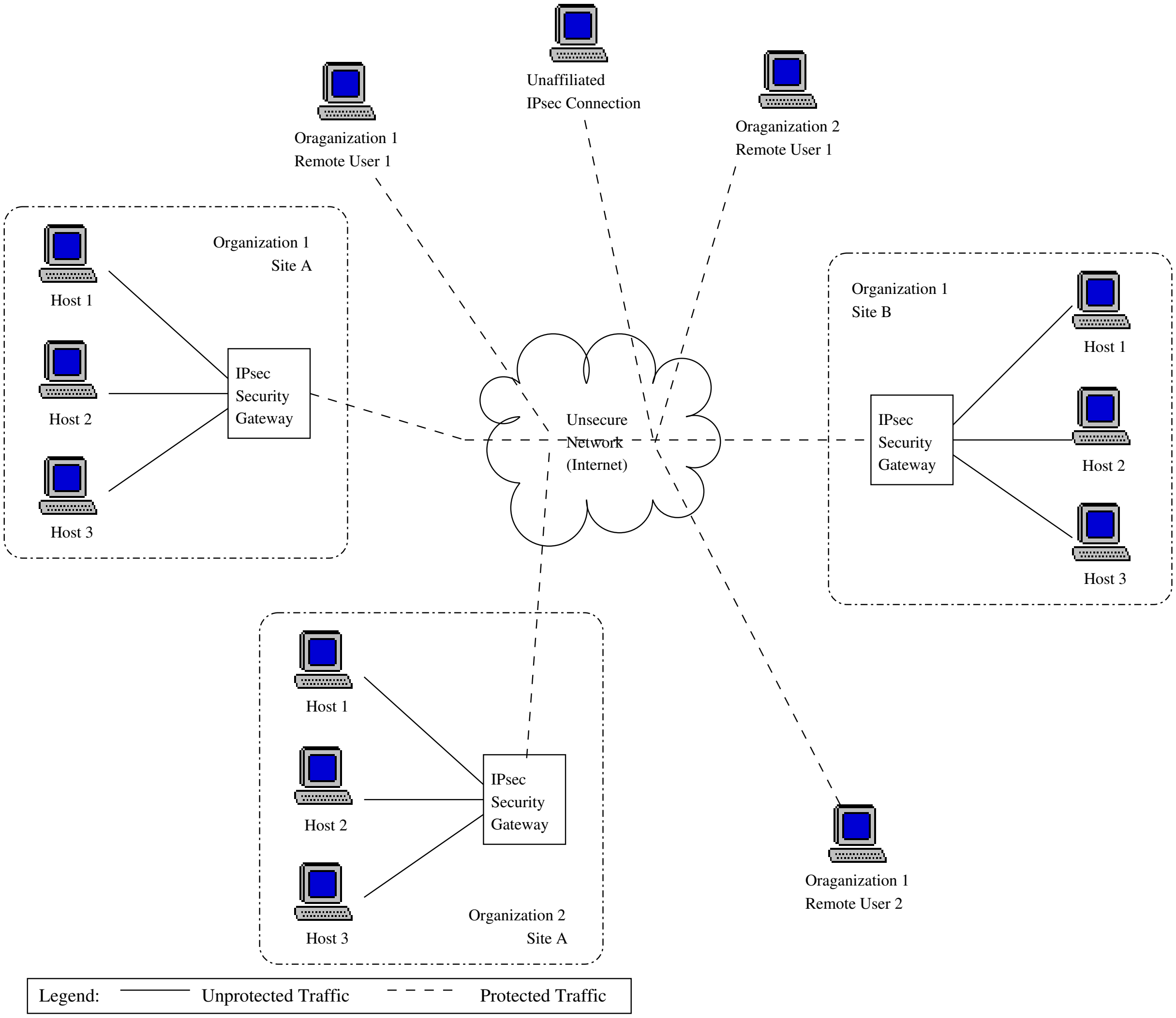

Figure 3: Predicted Future Usage of IPsec.

Figure 3 depicts what is expected to become a common usage of IPsec where, for example, two organizations, 1 and 2, with possibly multiple sites and multiple remote users, are intercommunicating. Their respective policies would likely dictate that all communication between all members of each organization be protected by a minimum level of service. This becomes increasingly difficult to manage if both organizations are frequently adding or removing remote users. Collaborating organizations in situations like this must be able to communicate vendor-independent configuration settings in an automated way and to be assured that their administrative policies (which may be different from one organization to another) are being adequately enforced. Figure 3 diagram depicts only a few example hosts in order to simplify the drawing and it is should be noted that a real deployed situation like this would probably expand to to tens of thousands of IPsec devices and possibly hundreds of organizations. As the number of nodes increase, it will become virtually impossible to manage a network like this using the existing tool sets.

At the time of this writing, a number of software solutions exist to deal with IPsec configuration management but they are implemented with proprietary mechanisms and each one deals only with a single vendor's products. As a result, there are no solutions available to manage complex, multi-vendor infrastructures currently being designed. A standards based, multi-vendor solution for IPsec configuration management must be developed or the deployment and use of IPsec will continue to be impeded by proprietary, non-interoperable management systems.

References

[1] The Internet Security Protocol. IETF RFCs 2401-2411.

[2] The Internet Engineering Task Force. http://www.ietf.org

[3] The Distributed Management Task Force. http://www.dmtf.org

[4] The Common Information Model. http://www.dmtf.org/spec/cim_schema_v25.html

[5] The IP Security Policy Working Group. http://www.ietf.org/html.charters/ipsp-charter.html

[6] The Policy Framework Working Group. http://www.ietf.org/html.charters/policy-charter.html

[7] IPsec Configuration Policy Model
http://www.ietf.org/internet-drafts/draft-ietf-ipsec-config-policy-model-02.txt

[8] Policy Core Information Model. IETF RFC 3060.

[9] The Configuration Management with SNMP Working Group.
http://www.ietf.org/html.charters/snmpconf-charter.html

[10] Webmin: an extensible Web based linux management interface.
http://www.webmin.com/webmin/

[11] Lightweight Directory Access Protocol (v3) IETF RFCs 2252

[12] Policy Core LDAP Schema IETF Internet-draft: draft-ietf-policy-core-schema-09.txt

[13] Simple Network Management Protocol Version 3. IETF RFCs 2570-2580.

[14] The COPS (Common Open Policy Service) Protocol IETF RFC 2748.

[15] COPS Usage for Policy Provisioning IETF RFC 3084.

[16] The Evolution of SNMP Working Group. http://www.ietf.org/html.charters/eos-charter.html

[17] The Transport Layer Security Working Group http://www.ietf.org/html.charters/tls-charter.html

[18] KAME: A IPsec implementation for open source BSD operating systems. http://www.kame.net

[19] Overview of the KAME IPsec/IPv6 stack, including project contributors.
http://www.kame.net/project-overview.html

[20] A FreeS/WAN IPsec configuration module for webmin.
http://www.niemueller.de/webmin/modules/freeswan/

[21] The Internet Key Exchange protocol, a sub-component of IPsec. IETF RFC 2409.

[22] Linux FreeS/WAN: An IPsec implementation for linux. http://www.freeswan.org/

[23] Cerberus: An NIST sponsored IPsec reference implementation for linux from NIST.
http://www.antd.nist.gov/itg/cerberus/

[24] National Institute of Standards and Technology http://www.nist.gov/

[25] A listing of businesses providing VPN implementations.
http://dir.yahoo.com/Business_and_Economy/Business_to_Business/
Computers/Communications_and_Networking/Virtual_Private_Networks__VPNs_/

[26] Virtual Private Network Consortium members supporting IPsec technologies.
http://www.vpnc.org/features-chart.html

[27] IPSec VPNs: Take Us To the Pilot, Mike Fratto.
http://www.networkcomputing.com/1019/1019f1.html

[28] Virtual Private Networks [an overview], Frank J Derfler, Jr.
http://www.zdnet.com/pcmag/stories/reviews/0,6755,2404675,00.html

[29] RedCreek e-Director. http://www.redcree.kcom/products/ravline-dir.html

[30] Global Enterprise Management System. http://www.pgp.com/products/gauntlet/default.asp

[31] ADI Management System. http://www.assured-digital.com/products/prodvpn/ams.htm

[32] Alcatel 5630 Secure VPN Management Suite.
http://www.cid.alcatel.com/products/droplets/summary.jhtml?productNumber=a5630

[33] VPNmanager Series. http://www.vpnet.com/Products/vpnmanager.shtml

[34] Lucent Security Management http://www.lucent.com/ins/products/lsms.html