A Scalable IPsec Policy Configuration System

NAILabs

September 15, 2003

1 Current Practices

IPsec-enabled devices are currently configured using one of two configuration methodologies. The first method is by-hand configuration using a command line interface either directly on the device's console or over a remote login session (E.G., telnet, remote-login or secure-shell). The second method of configuration is using an IPsec management console that is proprietary in nature and is only capable of configuring devices manufactured by a single vendor.

The problem with the command line configuration model is that it is not a scalable architecture which can be used to manage a large, complex network of IPsec devices. Simply put, having to administratively configure over 1000 devices (or even 100 devices) every time a new security policy is added or removed would be a daunting task and is subject to human error. Automated scripting of these command line tasks makes the configuration of medium sized networks seem like a plausible goal. However, it is still subject to other problems such as command line tool differences between vendor distributions, device versioning differences and detection of automated policy conflicts. It is clear to any network administrator that command line configuration is only a stop-gap measure until a better solution makes itself available.

Fortunately, many vendors selling IPsec-enabled devices also sell an IPsec management console, usually sold as a separate product. These management consoles provide the ability to easily manage a network of the particular vendor's IPsec enabled devices. However, this solution only works if all the devices in a network are bought from the same manufacturer. In cases where this is not acceptable, for example in merging or collaborating organizations and in budget constrained environments, the only alternative is to run multiple management servers and copy configuration information from one to the other by hand. This process, of course, is also subject to human error.

This problem is further documented in [1].

1.1 IPsec Security Policy Architecture

Policies within the IPsec context allow network administrators to control how both incoming and outgoing traffic is processed as it passes by the IPsec protection services. When incoming packets are delivered to the IPsec processing engine from the network or from the device's OS itself, they are run through a process to determine what action should be taken on the packets based on the installed system policies. A decision tree depicting a simplified version of this processing is shown in Figure 1.

IPsec policies apply to both incoming and outgoing data packets traveling to or through a given network device. Functionally, the policy databases are designed to answer the following basic questions:

- With whom may I perform transactions?
- What parameters must be enforced upon these transactions?
- What should be done with packets not meeting these requirements?

The IPsec Security Association Database (SAD) contains information relating to current IPsec

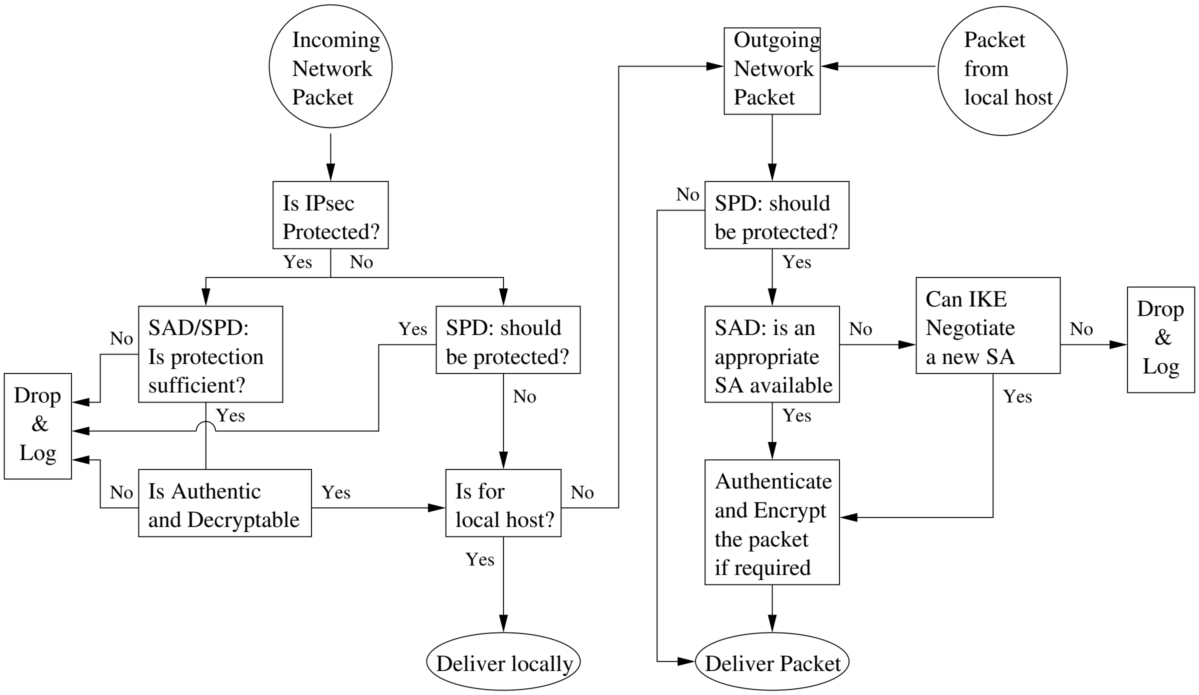
Incoming Network Packet
Is IPsec Protected?
Yes
No
SAD/SPD: Is protection sufficient?
No
Yes
SPD: should be protected?
Yes
No
Drop & Log
Is Authentic and Decryptable
No
Yes
Is for local host?
No
Yes
Deliver locally
Outgoing Network Packet
Packet from local host
SPD: should be protected?
No
Yes
SAD: is an appropriate SA available
No
Yes
Can IKE Negotiate a new SA
No
Yes
Drop & Log
Authenticate and Encrypt the packet if required
Deliver Packet

Figure 1: IPsec Policy Decision Tree.

connections in use within a given device and is consulted for both incoming and outgoing packets when authentication and encryption/decryption services are to be applied. The IPsec policies that affect this database dictate what happens when packets don't appropriately match the requirements within the SAD.

The Security Policy Database (SPD) controls the IPsec processing requirements of a device as a whole. It is this database which stipulates whether traffic between two entities either must be protected by IPsec or is allowed to pass through the box unprotected. Functionally this database operates as a configuration mechanism that sits above the SAD database. The SPD configures the SAD to implement the policies defined by the security administrator.

If an incoming network packet is not protected by IPsec, then the security policy database is consulted to decide what to do with the packet based on where it came from and where it was going to. The action taken could include: discarding it; starting a new IPsec enabled connection; using an existing IPsec connection for transmission elsewhere; or logging an error or warning.

The aspects of IPsec security policy are more completely described in the IPsec specification documents [2, 3, 4].

The Internet Key Exchange (IKE) protocol document [5] describes additional policies that should be imposed upon key negotiations needed when setting up IPsec security associations. Policies that may be applied to the IKE protocol include how often keys must be updated, when perfect forward secrecy is required for key exchange, etc.

2 A Standards Based Approach

The best solution which can help reduce the difficulties associated with managing a large number of vendor-independent devices affected by security policies is to develop a standards-body based management architecture. Doing so will enable management servers to be developed independently of the devices to be managed. This type of solution worked well in the general network management area with the advent of the SNMP protocol. Today, almost all network infrastructure devices are instrumented with SNMP agents. SNMP management servers are developed independently and are written to conform to SNMP specifications such that they are capable of configuring or monitoring all of the devices within a network, regardless of the manufacturing vendor. This is of critical importance when a new device is deployed that was not previously in existence when the management server in use was designed.

2.1 Policy Related IETF Working Groups

Multiple IETF [6] working groups already exist that are attempting to produce standardized methods of IPsec specific and other types of policy based management. Unfortunately, however, the majority of these working groups are currently concentrating on conceptual modules and have yet to produce implementable results. Until the start of this project, very little work had been done in standards bodies working groups with respect to implementable results within the IPsec realm. With the start of this project, the working groups received a kick-start from of the work put forward by members of this project team and the efforts have been well received by the IPSP [7] IETF working group.

2.1.1 Policy Framework Working Group

The Policy Framework Working Group [8] 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) [9]. The DMTF is defining a Common Information Model (CIM) [10] which classifies a wide range of system capabilities including those needed for quality of provisioning and for IPsec policy management.

2.1.2 IP Security Policy Working Group

The IP Security Policy Working Group (IPSP) [7] is devoted to producing a IPsec security policy architecture to be used in managing IPsec networks. Unfortunately, almost all of their work so far has been concentrating on conceptual data models [11] and no implementable results had been produced by the working group until the work done for this project was submitted to them. The work submitted from this project had the additional benefit of bringing new needs to light and the data model being worked on by the IPSP working group was modified accordingly.

2.2 Selecting a Protocol

It is likely that as the internet and it's usage of the IPsec protocol grows, IPsec-enabled network topologies are likely to increase in complexity to the point that they will become unmanageable using current IPsec management methods. This is described in greater detail in Section 1 and in [1]. It is also expected that the need for cross and inter-organizational collaboration will produce a need for policy administration to be divided among groups of administrators who each need various levels of access to policy configuration elements. SNMPv3 [12] is one such management protocol that is capable of handling these types of complex management topologies and appears to be a good starting point in solving the IPsec management problem.

2.2.1 SNMPv3

The selection of a protocol that meets the criteria defined above leads to only one currently existing and accepted protocol. The SNMPv3 protocol contains most of the security mechanisms needed to make management of these types of complex networks possible, such as multiple administrative users making use of different authentication and encryption algorithms, a built-in light-weight key management system and a fine grain level of access control. Furthermore, because SNMP instrumentation is present on most network devices, the additional overhead imposed by a policy management extension to the SNMP Management Information Base (MIB) will introduce a minimal overhead on existing devices's internal architecture and resources. A new protocol designed solely for security policy management would introduce a larger overhead with little additional gain.

An additional attractive feature of the SNMPv3 protocol is use of device "localized" keys. An authentication or encryption key which can be used to access and configure one device is not usable to access or configure another device. Thus, an attacker breaking into one box who steals SNMPv3 keys will not be able to attack other devices using the stolen keys.

The usage of SNMP as a IPsec configuration protocol should be well received by the vendor community. Many existing IPsec management server vendors make use of SNMP to control their own devices using their own proprietary MIBs. Some of these vendors have given us positive feedback about the work undertaken by this project to put a IPsec configuration SNMP MIB through the standardization process.

2.2.1.1 Needed Additions to the SNMPv3 Protocol

The SNMPv3 protocol is the protocol that best suits our needs, but a few additions might needed. Fortunately, the SNMPv3 protocol is extensible and adding these new needed pieces is fairly straight forward. If the needs are found to be true and the solutions must be implement implemented, these additions will be sent through the IETF standardization process to ensure other vendors will also be capable of implementing the defined new features.

2.2.1.1.1 New Encryption Algorithms

Currently, the SNMPv3 protocol provides support for only the DES encryption algorithm. The DES algorithm is inadequate for distributing keys to an IPsec device for use by other IPsec encryption algorithms which themselves make use of a longer key length. Performing brute force attacks against the DES encrypted configuration protocol would be easier than performing brute force attacks against the longer length encryption protocols.

Fortunately, the privacy definitions in the SNMPv3 protocol allow for future encryption algorithms to be easily added to the existing protocol without breaking previous compatibility. Work has been recently submitted to the IETF to define Rijndael/AES encryption support for SNMPv3 [13]. Unfortunately, this work is currently stalled and may need to be helped forward through the standardization process if it fails to advance on it's own.

2.2.1.1.2 New Authentication Algorithms

The SNMPv3 protocol currently supports both MD5 and SHA-1 based HMAC authentication algorithms, which provides sufficient authentication mechanisms for ensuring the authenticity of configuration traffic. However, these same hashing algorithms are also used to derive keys for any encryption algorithms used by the protocol. The key lengths produced by the MD5 and SHA-1 hashing algorithms are 128 and 160 bits respectively. This is a sufficiently long key length for most encryption protocols, but longer key lengths may be needed for generating keys suitable for use with a 256 bit Rijndael/AES algorithm. If a need for a longer length encryption algorithm is found, it is likely that a new authentication mechanism set will be needed and SHA-256 and possibly SHA-512 definitions will need to be written for use within SNMPv3. An alternative possibility is that a new document could be written that defines how longer key lengths within SNMPv3 can be derived using a shorter length hashing algorithm.

2.2.1.1.3 New Access Control Mechanisms

Currently the SNMPv3 security mechanisms implement access control by carefully limiting what actions a user may perform based on what objects the user wants to perform the actions upon. What is not possible, under the current access control mechanisms, is to restrict access based on the values associated with those objects. This new access control feature may prove to be needed for cases where local or peer administrators should be allowed to modify policies in specific fashions as long as they meet the guidelines imposed by their parent organization. Under the current access control mechanisms provided by the SNMPv3 protocol, imposing these types of value restrictions would be impossible. For example, it would not be possible to allow a local administrator to change an IPsec policy such that AES encryption was used in preference to DES encryption, but to disallow that same administrator to make use of the NULL encryption algorithm or to delete the encryption policy altogether. Fortunately, it should be fairly easy to extend the SNMPv3 access control mechanisms to incorporate this new feature.

2.2.1.1.4 Initial Key Derivation for SNMPv3 Engines

Any security protocol implementing authentication or encryption must be initialized with cryptographic keys prior to using the protocol. The SNMPv3 protocol provides no mechanisms for initial key derivation beyond those provided by the manufacturer of the device. Work has been submitted to the IETF for providing some easier-to-use mechanisms for initial key derivation [14], but the work is currently classified as experimental by the IETF and this project may provide the motivation for propelling it forward into the normal standards track of the IETF.

2.2.2 Other Candidate Protocols

Other protocols were considered for use by this project, but were not able to meet the requirements discussed above.

2.2.2.1 COPS

COPS [15, 16] does not provide a method for multi-management servers and local administrators to simultaneously (but harmoniously) modify or install policies on a given device. Because of this, it is an unacceptable solution for complex network topologies with overlapping administrative realms.

It should be noted that some companies do believe that COPS is an appropriate choice for IPsec policy provisioning. Parallel work is going forward in the IPSP working group to define a COPS PIB that will be very similar to the MIB developed by this project. The authors of the MIB produced under this project and some of the IPsec vendors do not believe, however, that COPS is wise choice for IPsec policy management in the complex network topologies discussed above.

2.2.2.2 LDAP

LDAP [17] is a excellent directory services protocol, but is not a suitable protocol for distributing policy to low-level network nodes routers and IPsec security gateways. It is, however, a storage mechanism candidate for management servers to make use of when storing shared policy related data at the management level.

3 Proposed Architecture

This project has multiple components that must be completed to meet the goal of being able to easily build, configure and deploy a vendor-independent IPsec instrumented network. All aspects of security policy management must be implemented and tested. The basic architecture of a policy management system and the IPsec-enabled devices on a network that it is responsible for is shown in Figure 2. In this figure, we can see that a policy management system, described in Section 3.3, is responsible for configuring multiple IPsec devices out on a network. To accomplish this, the management system must communicate with each device's internal SNMP engine which must in turn speak to the internal IPsec and IKE implementations within the device.

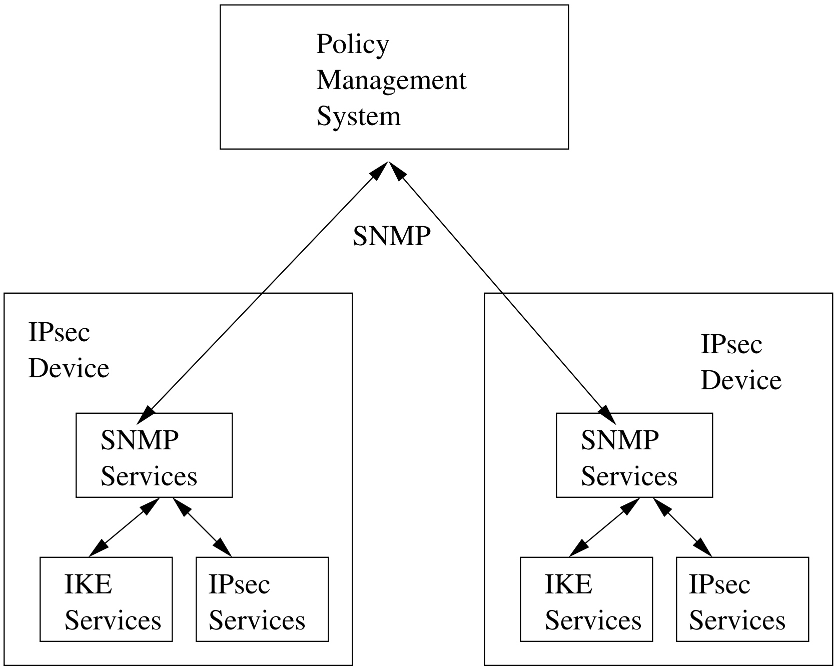

Figure 2: Managed Network Architecture.

3.1 Standards Body Work

The first task to be completed by the project is to define an architecture for the IPsec device instrumentation which meets the required criteria imposed by collaborating IPsec device vendors within the IETF. The SNMP MIB which implements this architecture must provide remotely accessible mechanisms by which a policy management system can create, modify, and delete IPsec-related policy definitions for any device under its control. Specifically, the MIB must be able to control all aspects of IPsec device configuration, including, but not limited to, the already existing Security Association Database and the Security Policy Databases described in Section 1.1. Work on the IPSEC-POLICY-MIB has already been begun under this project and the initial work has been published as an internet-draft [18] by project team members. This work has been well received by the members of the IETF IPSP working group.

3.2 IPsec Device MIB Instrumentation

Based on knowledge gained while evaluating freely-available releases of IPsec implementations [1], it was decided that the Cerberus IPsec [19] and PlutoPlus IKE [20] implementation would be the software package set this project will add instrumentation to in order to support the IPSEC-POLICY-MIB. The developers of these packages are very interested in these modifications and are looking forward to seeing the project results.

The Cerberus IPsec software resides within the Linux kernel and can not be easily instrumented with SNMP directly. However, the PlutoPlus application is a demon running in the user process space with an already existing communication and configuration mechanism set up between it and the Cerberus code within the kernel. This makes PlutoPlus an ideal location to add support for the IPSEC-POLICY-MIB. It is expected that most of the work involved with implementing this MIB will, therefore, take place within the PlutoPlus application.

3.3 Management System

The IPsec policy management system implemented in this project will consist of a few parts, as depicted in Figure 3.

The management server, seen in the middle of Figure 3, will be responsible for configuring multiple IPsec-enabled network devices within its realm of responsibility. These devices may be devices directly under its entire control, or devices for which it is allowed to configure only a subset of the policy system such as in hierarchical and peer based management scenarios. The management server itself is made up of two parts: a SQL database to house the policy definitions, and a management engine which is responsible for actually performing the necessary configuration of the IPsec-enabled devices.

A policy management console is also needed to appropriately load the SQL databases with the policy definitions to be applied to the managed IPsec devices. This console will map the data housed in the database structure into a more intuitive form displayed through it's user interface.

Although the initial target of this architecture is IPsec policy management, the architecture is flexible enough to be used for any type of policy or configuration management system. This allows the system to be easily extended for other uses, such as managing systems implementing quality of service policy agreements.

The management console user interface to be developed will allow administrators to define and assign policies to devices within the controlled network and will use a database to store it's notion of a network's policy set. The user interface interacting with the policy database will be implemented as a HTML based web console for maximum portability. It should, however, be possible to easily add additional management consoles in the future which can operate concurrently with the web based console. Multiple consoles, regardless of type, should be capable of configuring the same database housed within the management server.

The database of policy definitions will then be monitored by policy management engine(s) which are responsible for translating the policies defined in the database into SNMP requests to be sent to the IPsec devices. Because the management console will be making changes that may need to be applied immediately, optional SNMP TRAP or INFORM packets may be sent by the management console to the management engine(s) to ensure they begin updating the remote devices immediately.

Not only may SNMP SET requests need to be sent to a device to configure it, but SNMP data retrieval operations may be needed as well to ensure that policies have been properly updated and have not been removed or modified.

Finally, IPsec enabled devices should be capable of logging errors and anomalies by forwarding notifications to the policy management engine for display and archiving. To do this, the IPSEC-POLICY-MIB will need to contain SNMP notification definitions for appropriate IPsec policy related event types needed by administrators to monitor their network for policy violations. These new policy specific event types should supplement the audit events described by the IPsec protocol definition documents. The management server and management console will both need to have a section of its operations devoted to alarms sent by way of these notifications. The management console would then display these notification events in an easily understandable form. Using this piece of the management console interface, a network administrator should be able to determine which points on the network were failing to establish needed connections and which points of the network were receiving improperly protected data packets, which could give indications of a network attack taking place.

It is also a goal of this project to make this architecture as scalable as possible. The management

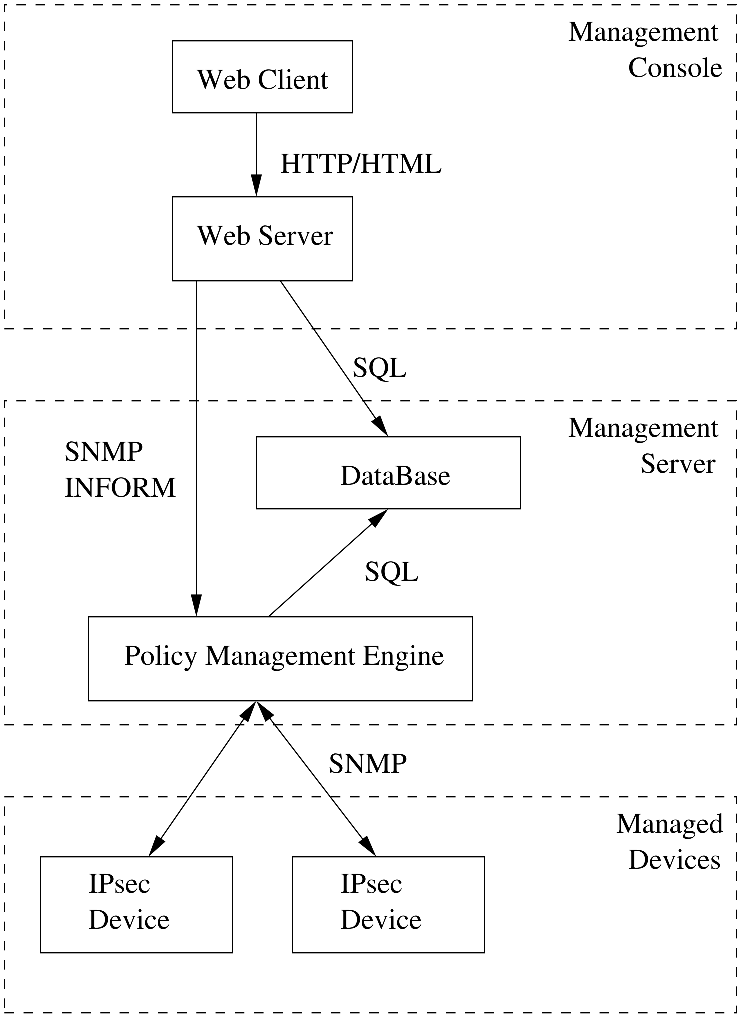

Figure 3: Policy Management Implementation Overview.

engines are carefully designed to allow multiple engines to be run simultaneously. When multiple policy management engines are operating in parallel, they can easily distribute configuration tasks among them. Similarly, multiple policy monitoring engines may also be needed and will be constructed to run in parallel as well. Extending the management and monitoring system to scale to a larger managed network should be as easy as adding a new server name to a list.

4 Future Work

Numerous future enhancements could be added to the architecture described above. Some of these may be implemented within the project itself if time and resources are available before the conclusion of the project and after the completion of the tasks described above.

4.1 IPsec and IKE Monitoring MIBS

Other SNMP MIBs [21, 22, 23] have been previously defined that provide monitoring objects for IPsec services within a device. These MIBs report not just how a device was configured but how IPsec was actually being used at any given instance in time. The configuration management system could make use of this information to provide a richer view of a network's IPsec usage.

4.2 Instrumentation of Other IPsec platforms

Though we can be confident that other commercial vendors will design and deploy devices implementing the MIB produced by this project, it can only be hoped that the MIB produced by this project will be implemented by the other freely available IPsec implementations as well.

4.3 Other Graphical User Interfaces

The management architecture described above is extensible such that other graphical front ends could be easily designed and made to cooperate with the management database.

4.3.1 Configuration Wizards

Proper setup and configuration of IPsec-enabled networks is a complex task and one subject to human-error. The policy management system designed and implemented by this project should be a major step forward in making configuration a simpler task. An obvious and highly useful extension to the User Interfaces implemented in the project would be a "Configuration Wizard" that could walk an administrator through the steps necessary to set up a new host or network by asking appropriately constructed questions.

4.3.2 IPsec Graphical Usage Overview

Given the architecture defined above (and the extra functionality described in Section 4.1), it should be fairly easy to use the data made available through this new configuration interface to accurately develop a complete usage map of IPsec within a given network. Furthermore, the map could be easily characterized by the types of IPsec connections and the parameters that make up the connections (for example, the DES encrypted links could be shown in one color while AES encrypted links could be shown using another color).

4.4 Other Policy Management

The policy management system described above in Section 3.3 is designed to be fairly generic, enabling it to be used in types of policy management other than IPsec specific policies. For instance, it could be easily and quickly extended to additionally manage policies affecting quality of service provisioning.

A Prioritized Goals

The following goals are expected to be completed within the prototype release:

- A basic web-based graphical frontend to the policy database
- A prototype web based policy management engine capable of configuring remote IPsec policy devices.
- A Preliminary IPSEC-POLICY-MIB definition.
- Proof of concept instrumentation of the Cerberus IPsec software allowing it to be configured in at least minimal fashion.

The following goals are expected to be completed by the end of the project:

- A revised IPSEC-POLICY-MIB document published through the IETF IPSP working group.
- A full featured management console including improved scalability utilizing the multi-threaded OpenSNMP code base.
- A monitoring agent capabile of detecting improperly configured IPsec devices.
- Notification monitoring services providing immediate policy update requests and event logging.
- Policy collision and conflict detection.
- Mobile device support.

The following items will be completed only if time and resources are available or they may be items for possible follow on work:

- Implementation of the already existing monitoring MIBs.
- Instrumentation of other IPsec implementations (E.G.: FreeS/WAN, Kame).
- Other non-web based graphical frontends.
- Configuration “wizards” allowing easy policy configuration for a given network.

References

[1] Network Associates Laboratories. Current IPsec Policy Configuration Options. Febuary 2001.

[2] S. Kent and R. Atkinson. IETF RFC 2401: Security architecture for the Internet Protocol, November 1998. Obsoletes RFC1825 [24]. Status: PROPOSED STANDARD.

[3] D. Piper. IETF RFC 2407: The Internet IP security domain of interpretation for ISAKMP, November 1998. Status: PROPOSED STANDARD.

[4] D. Maughan, M. Schertler, M. Schneider, and J. Turner. IETF RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP), November 1998. Status: PROPOSED STANDARD.

[5] D. Harkins and D. Carrel. IETF RFC 2409: The Internet Key Exchange (IKE), November 1998. Status: PROPOSED STANDARD.

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

[7] The ip security policy working group. http://www.ietf.org/html.charters/ipsp-charter.html.

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

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

[10] The common information model. http://www.dmtf.org/spec/cim_schema_v25.html.

[11] Ipsec configuration policy model.
http://www.ietf.org/internet-drafts/draft-ietf-ipsec-config-policy-model-02.txt.

[12] J. Case, R. Mundy, David Partain, and Bob Stewart. IETF RFC 2570: Introduction to version 3 of the internet-standard network management framework, April 1999. Status: INFORMATIONAL.

[13] U. Blumenthal. Rijndael encryption protocol with snmpv3 usm, December 2000.

[14] Michael C. StJohns. Diffie-helman usm key management information base and textual convention, March 2000.

[15] D. Durham, J. Boyle, R. Cohen, S. Herzog, R.Rajan, and A. Sastry. The cops (common open policy service) protocol, January 2000.

[16] G. Bossert, S. Cooper, and W. Drummond. IETF RFC 2084: Considerations for Web transaction security, January 1997. Status: INFORMATIONAL.

[17] M. Wahl, A. Coulbeck, T. Howes, and S. Kille. IETF RFC 2252: Lightweight Directory Access Protocol (v3): Attribute syntax definitions, December 1997. Status: PROPOSED STANDARD.

[18] W. Hardaker, M. Baer, R. Charlet, D. Partain, and J. Saperia. Ipsec policy configuration mib, February 2001.

[19] NIST Cerberus: An IPsec reference implementation for Linux.
http://www.antd.nist.gov/itg/cerberus/.

[20] NIST PlutoPlus: An IKE Reference Implementation for Linux.
http://www.antd.nist.gov/itg/plutoplus/.

[21] J. Shiver T. Jenkins. Ipsec monitoring mib, July 2000.

[22] J. Shiver T. Jenkins. Isakmp doi-independent monitoring mib, July 2000.

[23] J. Shiver T. Jenkins. Ike monitoring mib, July 2000.

[24] R. Atkinson. IETF RFC 1825: Security architecture for the Internet Protocol, August 1995. Obsoleted by RFC2401 [2]. Status: PROPOSED STANDARD.