



V-IPSecure

for Linux

High-Velocity IP Layer Security

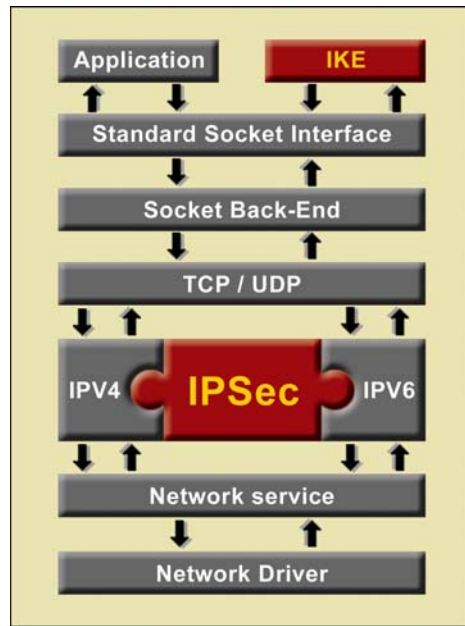
V-IPSecure is a high-performance, lean and flexible implementation of the IPsec protocol suite which provides IP extensions needed for security services at the network layer (Layer 3). Unlike other protocols that secure individual network applications, IPsec protocols secure the connection at the network layer, thus automatically and transparently securing all network applications that use it. **V-IPSecure's** implementation of these protocols provides a high-quality cryptography-based communication channel on embedded systems running *Linux®*. Its end-to-end securing of IP datagrams prevents access or modification of any information from above the IP layer, when passing through intermediate nodes in a public network. This enables secure virtual private networks (VPN) to be carved out of a public and/or insecure network. Designed exclusively for embedded use, **V-IPSecure's** robust and configurable implementation makes it an ideal fit for embedded devices such as Internet appliances, VPNs, gateways, secure terminals, and routers.

Secure Network Layer

While the TCP/IP suite of protocols has become very popular among embedded systems with the proliferation of connected embedded devices, security is not part of the original IP design. Hence any embedded application with security requirements needs to implement security at the application, transport, network, or link layer. Placing

security at the network layer has several advantages when security requirements affect all data going through the stack. Network layer security is transparent to the applications which use the network stack. Further, the security architecture is independent of the network type or topology to which the embedded device is connected and encrypted packets can be routed and switched on any network that supports IP traffic.

V-IPSecure implements a secure network layer (IPsec) that provides data integrity, origin authentication, data confidentiality, access control, partial sequence integrity, and traffic flow confidentiality services for communications between any two



Features

- ❖ AH, ESP (with authentication option)
- ❖ IPComp payload compression
- ❖ Tunnel and Transport Mode
- ❖ Support for Manual Key Exchange
- ❖ IKE with pre-shared keys, RSA digital signatures, X.509 certificates
- ❖ IKE phase 1 Main & Aggressive Mode, Phase 2 Quick Mode
- ❖ Diffie-Hellman groups: 1, 2, 5
- ❖ Support for IPV6
- ❖ IKE Hooks for Kerberos Authentication
- ❖ Configuration via commands or configuration file
- ❖ PF_KEY v2 key management API
- ❖ Support for IKE INITIAL CONTACT
- ❖ Support complex bundle option & sysctl based management
- ❖ Support for CPU types of either endian-ness including PowerPC, MIPS, X86, ARM/XScale
- ❖ Royalty-free full source distribution

networks or hosts. Replay-detection as defined by the IPsec standard is also performed by using sequence numbers combined with authentication.

Support for Standards

V-IPSecure includes a complete set of standards-based protocols for IPsec-enabling a standard TCP/IP (IPv4 or IPv6) network stack.

Authentication Header (AH) Protocol: attaches a strong crypto-checksum to packets for a guarantee of authenticity, and ties data in each packet to a verifiable signature. This allows communicating parties to verify that data was not modified in transit (connectionless integrity) and that it genuinely came from its apparent source. Optionally, it can contain protections against replay attacks.

Linux Edition Features

- ❖ Designed Exclusively for embedded Linux use
- ❖ Native support for many variants of Linux including Monta Vista Linux
- ❖ Easy integration with BSPs
- ❖ Enhanced memory management support
- ❖ Integrated with Linux build and install methodology
- ❖ Requires no special network stack, and uses only generic Linux services
- ❖ Works with existing IPv4 and IPv6 stacks

Encapsulating Security Payload (ESP) Protocol:

encrypts data using symmetric keys, to secure it against eavesdropping during transit. It provides a guarantee of confidentiality and optionally provides for integrity and message authentication as well.

IP payload compression (IPcomp): provides a way to compress a packet before encryption which increases the effective throughput, and more importantly, improves the encryption characteristics of the plain-text payload.

Internet Key Exchange (IKE): is a powerful and flexible negotiation protocol that allows communicating parties to negotiate the methods and parameters of the secure communication channel, such as the sharing of secret keys between peers.

V-IPSec seamlessly integrates with any IPv4 or IPv6 based TCP/IP stack, leveraging features such as PMTU support if the native stack provides it. Secure Channel Framework

The flexibility and power of **V-IPSec** is enhanced by a highly configurable framework for policy and secure channel management. It allows for a flexible set of rules to decide when to apply the security policies and when to skip them, and provides different levels of security

setup. For example, a secure communication channel to one network node may consist of a simple authentication scheme for traffic in both directions, while a highly secure authentication and encryption scheme may be setup for other hosts or entire networks. The management control for such flexibility is provided through a user friendly "setkey" interface to the Security Policy Databases. "setkey" is an interpreter for commands related IPsec configuration, which may be parsed from the command-line or may be provided in a file. This interface may also be used to pre-share secret keys for encryption between network nodes.

Automatic Key Negotiation

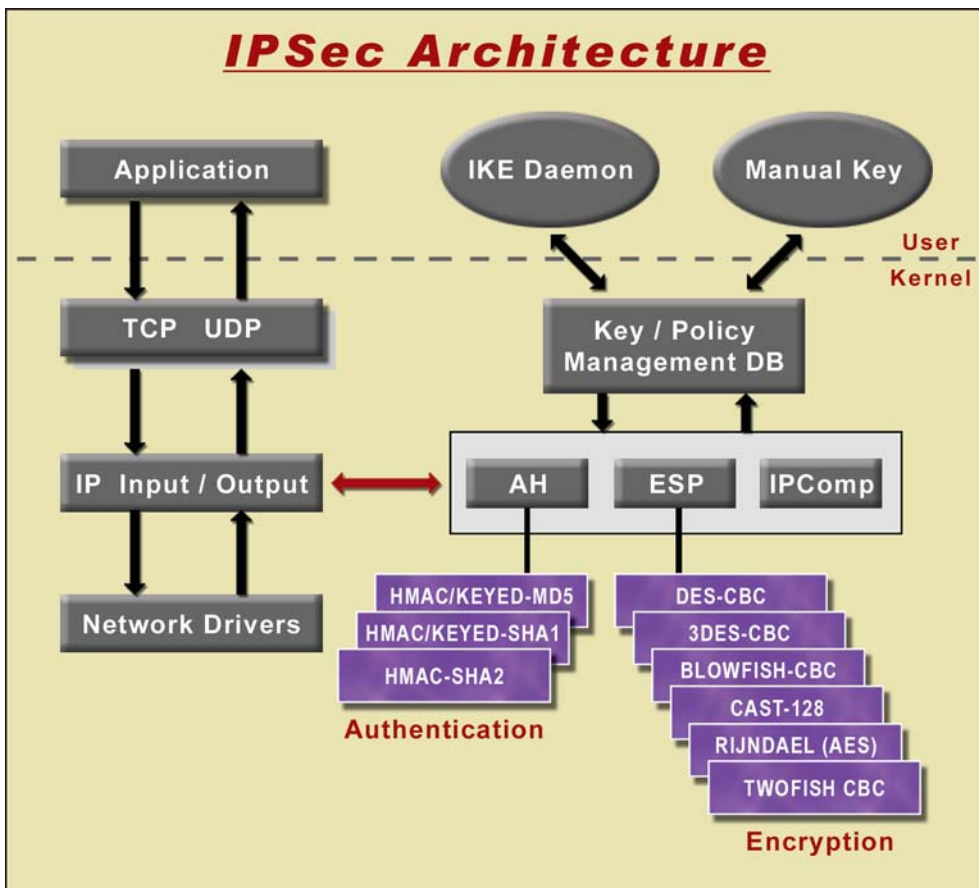
To use any encryption in a network environment, communicating peers must first exchange keys. While manually sharing keys is a possibility and is fully supported in **V-IPSec**, it can become intractable as the number of IPsec hosts increases. For this reason, **V-IPSec** includes an implementation of a mechanism for automatic key negotiation called Internet Key Exchange (IKE). IKE is based on the Diffie-Hellman key exchange and provides mechanisms for automatic

- ❖ IPsec configuration via sysctl facility
- ❖ Configurable anti-replay window dimension
- ❖ Support for explicit specification of IKE exchange
- ❖ Configurable handling of packets not compliant with an security policy
- ❖ Key/policy management via PF_KEY API
- ❖ Designed for hardware acceleration
- ❖ Available in full-source format
- ❖ Customization hooks and callouts
- ❖ Unwanted components can be scaled out

generation and frequent renewal of the crypto keys for the high security without increasing key-lengths which may slow down the encryption process. IKE integrates the Internet Security Association and Key Management Protocol (ISAKMP) with the Oakley key exchange scheme. ISAKMP defines a standardized framework to support negotiation of security associations (SA), initial generation of all cryptographic keys, and subsequent refresh of these keys. Oakley is the key management protocol that is used within the ISAKMP framework. IKE supports automated negotiation of security associations, and automated generation and refresh of cryptographic keys. Authentication in IKE can be achieved using a variety of mechanisms such as pre-shared keys, X.509 certificates, Kerberos etc. before key exchange begins. When using Kerberos for authentication, **V-IPSec**'s IKE is completely interoperable with Microsoft Windows® implementations of IPsec and IKE using Active Directory as the authentication database.

Security Associations

A Security Association (SA) is a one-way association between a sender and a receiver of security services. Each SA represents one direction of traffic. The security association separates the key management and the security mechanisms from each other. Each **V-IPSec** SA defines a set of parameters including the sequence number and window dimension for anti-replay service, the protocol mode, the lifetime of the SA, the path MTU, and other implementation details. For



authentication services in AH or ESP, and for encryption services in ESP, each SA also defines parameters such as the choice of cryptographic algorithm, keys in use, key lifetimes, initial values, etc. In this way, **V-IPSecure** makes it possible to bundle SAs to achieve the desired level of security in a fine-grained manner.

Security Policies

Security Policies are a flexible set of rules specified by network administrators to decide whether to secure an IP packet, bypass security for it, or discard it, affording different levels of security setup. For example, an SA to one network node may consist of a simple authentication scheme, while a highly secure authentication and encryption scheme may be setup for other hosts.

SA and SP Database APIs

The management control for such flexibility is provided through a set of user friendly APIs to access and modify the SP Database (SPD) and APIs for configuration commands. APIs are also included for manually setting up

Security Associations. These APIs may be called programmatically, or manually from a target-resident shell during development. This interface may also be used to pre-share secret keys for encryption between network nodes.

Tunnel and Transport Modes

Depending on the mechanism of secure IP packet transmission, **V-IPSecure** supports two types of SAs, which define the IPsec protocol mode in use:

Transport mode SA: A security association between two hosts used to secure the traffic of higher layer protocols.

Tunnel mode SA: A security association modeled similar to an IP-in-IP tunnel by encapsulating IP packets into new packets suitable for secure connections between security gateways.

Based on application requirements, **V-IPSecure** may be configured in either mode or a mix of the two modes for different connections. ESP in transport mode allows for lower processing overhead but provides neither authentication nor encryption for the IP header, making it vulnerable to spoofing. In ESP in tunnel mode, the original

Also Available

- ❖ **NetF1**
High Octane TCP/IP including IPV6
- ❖ **S Secure Family**
Security protocols (SSL, SSH, IPsec/IKE)
- ❖ **AuthAgents**
Authentication agents: Kerberos, RADIUS
- ❖ **Switchcraft Suite**
802.n Switching components

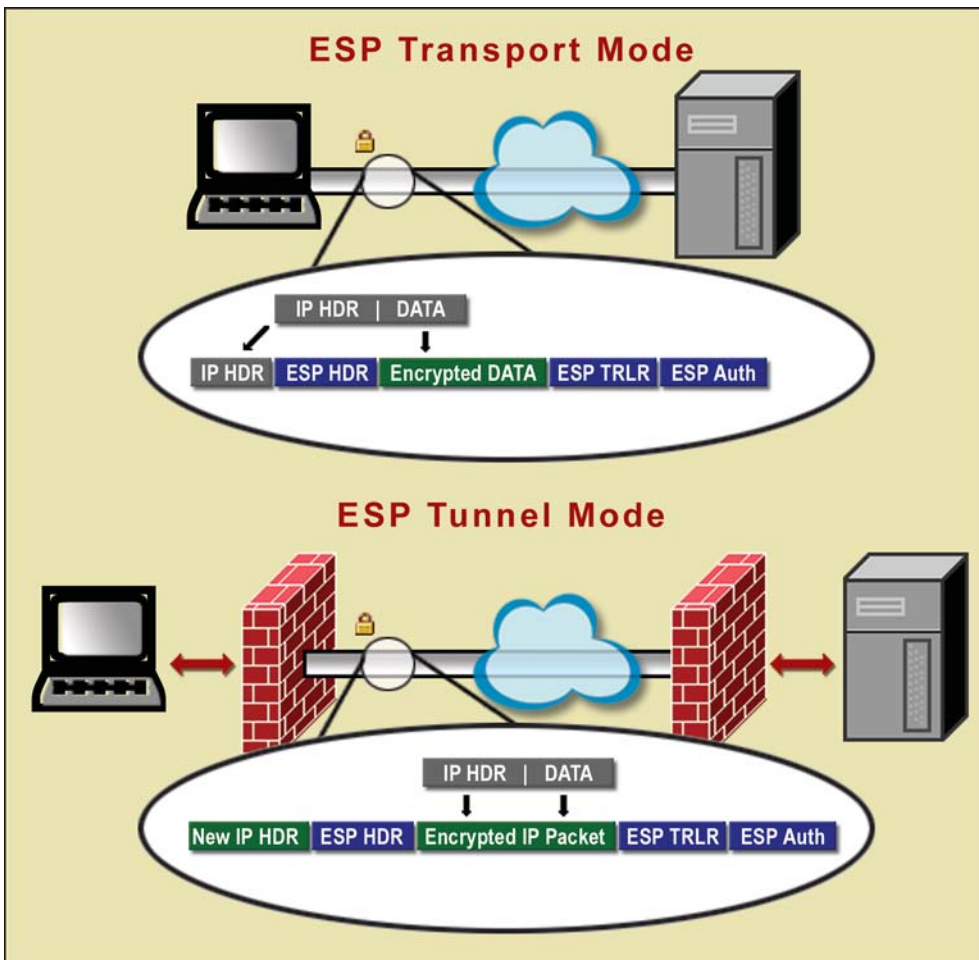
Custom Solutions

TeamF1's professional services can provide the resources and expertise to build customized implementations of **V-IPSecure** including support for other authentication protocols, support for hardware accelerators, network processors off-load, and to provide scaled-down application-specific implementations.

datagram becomes the payload data for the new ESP packet, hence protection is total if both encryption and authentication are selected, but has a higher overhead. Further, tunneling allows for the passing of illegal IP addresses through a public network, which may be required in certain applications. Tunneling with the ESP also has the advantage of hiding the original source and destination addresses from users on the public network — defeating or at least reducing the power of traffic analysis attacks.

Built for Linux

V-IPSecure is a plugin-on-the-run component optimized for *Linux 2.4.x* systems with support for multi-threading and lean, yet fast abstractions. It has been extensively validated on a variety of CPU architectures, thus minimizing development and integration efforts. It makes use of the native network stack or can be configured to run in high-performance mode with **TeamF1's NetF1** TCP/IP stack. It attaches itself to the kernel hooks provided by the standard Linux TCP/IP stack at the LocalIn (incoming), LocalOut (outgoing), and Forward (forwarding) paths on the packet flow. It can be built into an easily loadable module for *Linux* based systems and can be plugged into the kernel statically, or on the fly without any need for re-linking or re-booting.



Supported AH Algorithms

Algorithm	Key Length
hmac-md5	128
hmac-sha1	160
keyed-md5	128
keyed-sha1	160
null	0 to 2048
hmac-sha2-256	256
hmac-sha2-384	384
hmac-sha2-512	512

Supported ESP Algorithms

Algorithm	Key Length
des-cbc	64
3des-cbc	192
simple	0 to 2048
blowfish-cbc	40 to 448
cast128-cbc	40 to 128
des-deriv	64
3des-deriv	192
Rijndael/aes-cbc	128/192/256
twofish-cbc	128/192/256
null	0 to 2048

Supported RFCs

RFC	Description
3173	IP Payload Compression Protocol (IPComp)
2401	Security Architecture for the Internet Protocol
2402	IP Authentication Header
2403	Use of HMAC-MD5-96 within ESP and AH
2404	Use of HMAC-SHA-1-96 within ESP and AH
2405	The ESP DES-CBC Cipher Algorithm With Explicit IV
2406	IP Encapsulating Security Payload (ESP)
2407	The Internet IP Security Domain of Interpretation for ISAKMP
2408	Internet Security Association and Key Management Protocol (ISAKMP)
2409	The Internet Key Exchange (IKE)
2410	The NULL Encryption Algorithm and Its Use With IPsec
2451	The ESP CBC-Mode Cipher Algorithms: TripleDES, Blowfish, RC5, CAST128
2085	HMAC-MD5 IP Authentication with Replay Prevention
1852	IP Authentication using Keyed SHA (HMAC-SHA-1).
1828	IP Authentication using Keyed MD5 (ah-old)
1829	1829 (esp-old)
2367	PF_KEY Key Management API
2459	X.509 PKI Certificate and CRL Profile (RSA Only)
2535	DNS Security Extensions (Supported, but not included)
2538	Storing Certificates in the DNS (Supported, but not included)
2104	HMAC: Keyed-Hashing for Message Authentication

This product includes software from the BSD Kame project and cryptographic software from the OpenSSL cryptographic library written by Eric Young (eay@cryptsoft.com) and Tim J. Hudson (tjh@cryptsoft.com)

Email: sales@TeamF1.com
Web: www.TeamF1.com
Ph: 510-505-9931 ext. 5
Fax: (510) 505-9941



© 2002-2004 TeamF1, Inc.