SANS Institute
InfoSec Reading Room

Issues When Using IPsec Over Geosynchronous Satellite Links

Satellite based broadband data networks provide the means to convey large volumes of TCP traffic to individuals and organizations over an enormous geographic area. Satellite based networks can also convey data for countless types of applications. However they are vulnerable to eavesdropping like any other wireless network and may be just one of many networks that user data traverses, thus employing IPsec would appear to be a logical end-to-end security solution. However when IPsec is used, TCP headers may be encrypted....

Issues When Using IPsec Over Geosynchronous Satellite Links

Abstract

Satellite based broadband data networks provide the means to convey large volumes of TCP traffic to individuals and organizations over an enormous geographic area. Satellite based networks can also convey data for countless types of applications. However they are vulnerable to eavesdropping like any other wireless network and may be just one of many networks that user data traverses, thus employing IPsec would appear to be a logical end-to-end security solution. However when IPsec is used, TCP headers may be encrypted. TCP can suffer from poor performance over networks with high latency, as is the case for geosynchronous satellite links. Performance enhancing proxies serve to optimize protocol performance over satellite links by examining transport layer (TCP) headers. Since IPsec obscures the TCP headers which proxies rely upon, the two technologies seem incompatible. This paper describes the salient points of TCP over satellite links, performance enhancing proxies, IPsec, and the issues with the combined use of these technologies. A tradeoff solution and its security implications are then presented. More research in the area of IPsec used in conjunction with performance enhancing proxies is needed to meet security and performance needs of satellite network users.

1 Introduction

The Internet has become a media of information exchange that most people (particularly in developed countries) take for granted. People search for information, trade emails, and conduct business transactions. Users are for the most part oblivious to the types of physical networks their data travels over and how underlying protocols manage the flow of that data. Wire tapping (man in the middle), HTTP session hijacking, and spoofing are just a few forms of attack used by cyber criminals to gain access to data traversing the Internet. It's tempting for an individual to assume that since data is going over wires and there are presumably so many wires and so many other users exchanging information that the chance of someone eavesdropping on their electronic conversations is small (i.e. security through obscurity).

Consider however, the case where indeed a user's data does begin its journey over wire lines, for example a user is dialed in to their ISP, and in the course of traversing various networks to reach its ultimate destination, one of the physical networks happens to be a geosynchronous satellite. Now even the minimal physical security afforded by wire line networks is gone. In fact many of the security issues which wireless networks must face (eavesdropping is difficult to detect, physical access to network not required) become amplified when a satellite network is considered. Geosynchronous satellites have enormous coverage areas, employing global beams capable of broadcasting to the continental United States as well as spot beam technology which provides "cell" coverage areas much like terrestrial cellular phone network, but each cell is much larger and can provide broadband data capability. An example of an existing system is DirecWay™.[1] A system such as this requires a user / enterprise to install a Very Small Aperture Terminal (VSAT), "an earthbound station

[1] DirecWay.com - http://www.direcway.com/

used in satellite communications of data, connected to and end user's network or PC."[2] In simplest terms a VSAT is a small satellite dish (0.6 – 3m diameter) which connects to a user's network. User data is sent/received via satellite and exchanged with a ground station called a hub. The hub represents the center point of a star network topology* with hundreds or thousands of VSAT terminals geographically distributed from the hub. The hub acts as a gateway between the VSAT terminals and the terrestrial networks comprising the Internet.

The DirecWay™ system "utilizes the Digital Encryption Standard (DES) with 56-bit key length as the bulk encryption algorithm over DVB (Digital Video Broadcast) [3]. 56 bit DES is no longer considered secure since it has a small key space of 2^{56} keys, which can be broken by brute force in relatively short time.

In absence of strong encryption, an attacker can purchase a VSAT terminal and with a basic knowledge of the data link layer protocol, hack the terminal to listen in on data intended for others. In order to listen in, the hacker would need to reverse engineer the VSAT's embedded code, and command the terminal to tune to different frequencies and timeslots (for TDMA based systems) in order to receive transmissions from the satellite intended for other terminals. This is no script kiddy exploit, but once compromised the VSAT can act as a powerful packet sniffer, eavesdropping on potentially sensitive or valuable data. A simple brute force attack of the weakly encrypted data will yield its contents.

Eavesdropping however, is not the only issue. A satellite link may be just one of several networks that a user's data travels over. A uniform, end-to-end security solution where communicating parties are not required to assume anything regarding the security of the networks between them (except perhaps that they are insecure!) would provide an ideal solution.

2 The Case for Use of IPsec

The threat of eavesdropping in any network can be countered with the use of strong encryption. In absence of strong encryption provided by the network, higher layer protection such as SSL, PGP, or SSH could be employed. The problem is that each of these mechanisms is specific to applications that use them. Ideally, the network itself should offer end-to-end security mechanisms, thus providing support for any application and ensuring compatibility across all end points.

Support for IPsec in IPV4 networks is widespread: most computer and network gear vendors offer integrated IPsec protocol stacks, often bundled with the product offering. In addition to confidentiality, IPsec offers other security services such as authentication, secure key management, message integrity, and prevention of replay attacks. Perhaps the most compelling case for the use of IPsec however is its end-to-end architecture. Since IPsec operates at the network layer these services are independent of the applications which require these services. Additionally, the end-to-end nature of IPsec means trust need only be established at the end points of a connection, which is exactly what's needed when sensitive or valuable data traverses a public network vulnerable to attack.

[2] Webopedia.com, VSAT
[3] Skycasters.com, white paper
* Future broadband satellite systems will enable mesh connectivity, eliminating the need for a hub

There are many facets to IPsec (each its own research topic), but this paper covers the specific issues of satellite network performance degradation when IPsec is used and the security issues raised when the end-to-end security provided by IPsec is not used. In order to understand why IPsec results in performance problems when satellite links are used in a network, one must understand:

- the issues of running the transmission control protocol (TCP) over networks with large propagation delay and
- the techniques used to mitigate those issues – performance enhancing proxies
- how IPsec works to provide confidentiality

The following sections describe these points and in doing so illustrate the problems in supporting IPsec over satellite links.

3 The Performance Problems of TCP over Satellite

TCP operates at the transport layer of the OSI protocol stack. It provides services such as flow control and reliable delivery of data between sender and receiver. These services are provided by TCP through the use of protocol acknowledgements, timers, and state machines running on each end of a TCP connection.

Fundamental to TCP is the concept of a segment or unit of transfer. A segment is the payload that is used to convey data from higher layers of the protocol stack as well as TCP header information. The maximum size of a segment (MSS) is negotiated between the sender and receiver of a TCP connection. The MSS is the smallest maximum transmission unit (MTU) size when data traverses different physical networks. The MTU is simply the largest data frame that an underlying physical network can carry, for example an IEEE 802.3 Ethernet frame is 1492 bytes. The MSS can also be further constrained by the sending / receiving application's ability to buffer data (e.g. both sender and receiver may be on an Ethernet LAN, but the receiver may be only capable of buffering 100 bytes, thus the MSS will be less than the MTU).

TCP sends sequences of bytes in segments to a receiver in a "sliding window" fashion. This simply means that more than one segment of data can be sent before an acknowledgement for octets in those segments is received. This enables more efficient use of available network bandwidth by allowing a sender to continue sending data without needing to stop and wait for an acknowledgement for each segment sent. In an ideal setting the window size would be set such that the sender *never* has to stop sending; the receiver sends acknowledgements back to the sender such that the window never fills, thus allowing continuous transmission.

Recall however, that one of the services provided by TCP is flow control. Flow control allows a receiver to adjust the window size by telling the sender how much data the receiver can currently handle thus the window size in TCP is variable. In addition to flow control, TCP must deal with network congestion, which is a "condition when intermediate machines [between endpoints] become overloaded." [4] In order to deal

[4] Comer, Page 220

with congestion TCP uses a "slow start" and subsequent "congestion avoidance" [5] algorithm. The slow start algorithm as the name implies slowly ramps up the window size (i.e. number of segments in the window) to avoid swamping a network during connection startup or causing rapid oscillations between congestion and idle states in the network. Once a calculated window size threshold is reached, TCP enters a congestion avoidance mode where the rate of window growth is slowed until it finally reaches the receivers advertised window size.

TCP's three way handshake for startup, congestion and flow control mechanisms, and sliding window rely on acknowledgements to be sent by the receiving end station. When sender and receiver communicate via geosynchronous satellite, a long feedback loop is introduced due to the fact that it takes a signal approximately 250ms for each hop (sender, to satellite, to receiver); thus a feedback loop of at least 500ms exists. This results in poor utilization of available bandwidth and poor performance during startup and congestion recovery. The situation is even further exacerbated when transmission errors occur, (for example interference caused by rain). Since the sending TCP end point cannot distinguish transmission errors from congestion, it must assume congestion exists and begin its slow start and congestion avoidance algorithms, again resulting in poor performance. The next section discusses how these performance issues are mitigated through the use of TCP proxies.

4 Performance Enhancing Proxies (PEP)

In order for satellite networks such as VSAT systems to deal with the performance issues discussed in the previous section, TCP PEPs are placed on each end of a satellite link. Note that this usually does not correspond to the end points of the TCP connection, but may be a network in the middle, or the first or the last hop of a connection. "A PEP is used to improve the performance of the Internet protocols on network paths where native performance suffers due to characteristics of a link on the path" [6]. A PEP not only serves to enhance performance, it can also act as a lower layer protocol adapter converting for example a terrestrial wire line data link layer protocol to an air interface data link layer protocol.

VSAT networks typically use a split connection PEP configuration as illustrated in Figure 4-1.

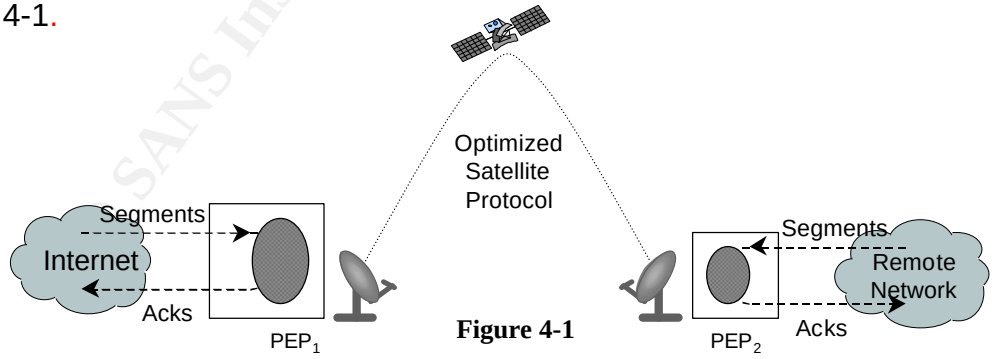

Figure 4-1

[5] Allman, Section 4.1.1. Note that these are not the only mechanisms, but the ones most relevant to the topic of this paper.
[6] Border, Section 1

This approach addresses the issues of TCP slow start and congestion avoidance by increasing the window size more rapidly than normally done by the protocol. This is accomplished through 'spoofing'. Each PEP in Figure 4-1 actually acts as a TCP endpoint (i.e. PEP_1 terminates a connection from a host on the Internet, PEP_2 terminates a connection from a host on a remote network). In doing so the PEP is able to send acknowledgements back for the originating host's transmitted segments before the destination host has even received them. In essence, the PEP is making a promise to send the originating host's data; this is what enables a PEP to rapidly open the originator's window size. The PEPs themselves are connected via satellite link, and often use a proprietary protocol optimized to the characteristics of the link. In the event that an unrecoverable error occurs, PEP relies upon application layer end to end error checking and recovery since an "application can never fully rely on TCP … to provide reliable end to end delivery."[7]

Additionally, (Border, Section 5.1.1) states that given "the relatively long round trip time [of a satellite network], TCP needs to keep a large number of packets in flight to fully utilize the satellite link." So the PEP is able to provide a larger window than may be normally used by TCP, this also serves to enhance performance.

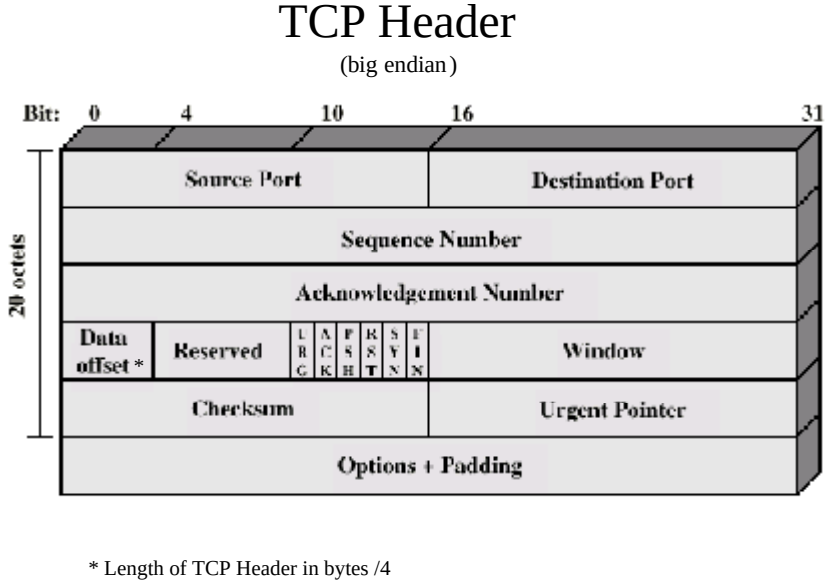

Figure 4-2 TCP Header[8]

In order to accomplish techniques such as spoofing and enhanced window sizes, PEP needs to examine fields within the TCP header. As shown in Figure 4-2 TCP Header, the header includes items such as source and destination ports, and sending and received sequence numbers. PEP uses this information to intercept and terminate a TCP connection, and spoof acknowledgements, yielding better network utilization and

[7] Border, Section 4.1.3
[8] Copeland, Page 5

end to end performance. For example a PEP on the sender's side would assume the identify of the real destination by performing a TCP three way handshake with the originating host, spoofing the destination port number and supplying acknowledgement numbers, thus appearing to be the real destination host. Note that the PEP on the receiver's side performs a similar function, thus the PEPs are transparent to the end hosts as far as transport layer is concerned.

It should also be noted that PEPs often perform data compression. In order to be effective, compression should be applied before encryption is performed. This is due to the fact that encryption by its nature randomizes the bits in a segment, thus precluding any opportunity to remove 'redundant' data. Since TCP PEP resides at the transport layer, higher layer encryption such as SSL, TLS, and SSH for example prevent effective compression. IPsec would *seem* to be a viable alternative: since it operates at the network layer compression could first be applied by the PEP before IPsec is applied, thus solving the problem. The need for PEP to examine the TCP header information is however, the crux of the problem when used in conjunction with IPsec. IPsec can scramble TCP header information, so by the time the encrypted segment is received by the local PEP it has already been rendered useless for spoofing and compression. The details of this problem are covered in the next section.

5 IPsec Data Confidentiality and PEPs

Before details of the issues regarding the use of IPsec in conjunction with PEPs is covered, it's necessary to review how IPsec provides data confidentiality. As previously noted confidentiality is not the only service offered by IPsec, but is the one most relevant to issues with PEPs, thus only confidentiality is covered.

IPsec provides confidentiality through the use of an encapsulating security payload, or ESP. As the name implies, plain text IP datagrams are encapsulated and encrypted before being forwarded.

Figure 5-1 Tunnel Mode ESP and Figure 5-2 Transport Mode ESP show where encryption is applied to the original data. Transport mode ESP is intended for use between two hosts with no intervening security gateway. A security gateway is an intermediate host or router which performs the ESP and other IPsec services on behalf of one or more internal hosts that wish to send data over a network to another host(s) which may also be relying on a security gateway at its end. Tunnel mode does not require the use of security gateways, however if a security gateway is used tunnel mode is mandatory. As the figures show, transport mode ESP protects only the upper layer data, leaving the original IP header in the clear while tunnel mode ESP results in the entire original IP datagram being encrypted (i.e. encryption is applied to an IP tunnel). The 'Endpoint IP Header' in Figure 5-1 Tunnel Mode ESP, specifies the termination point of IPsec, which may be a security gateway or the target host itself depending on how the network is configured. It illustrates how encrypting the tunneled IP packet helps hide the true destination, which may be a host on a private network.

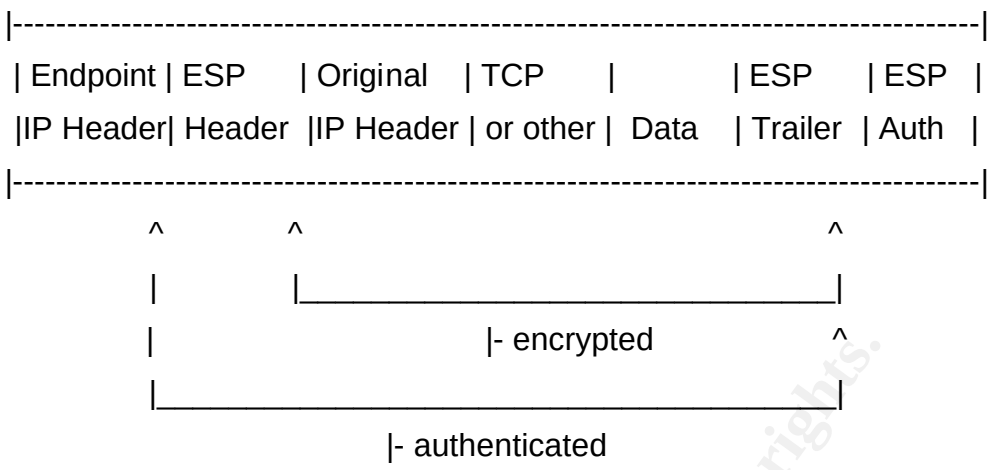

Figure 5-1 Tunnel Mode ESP[9]

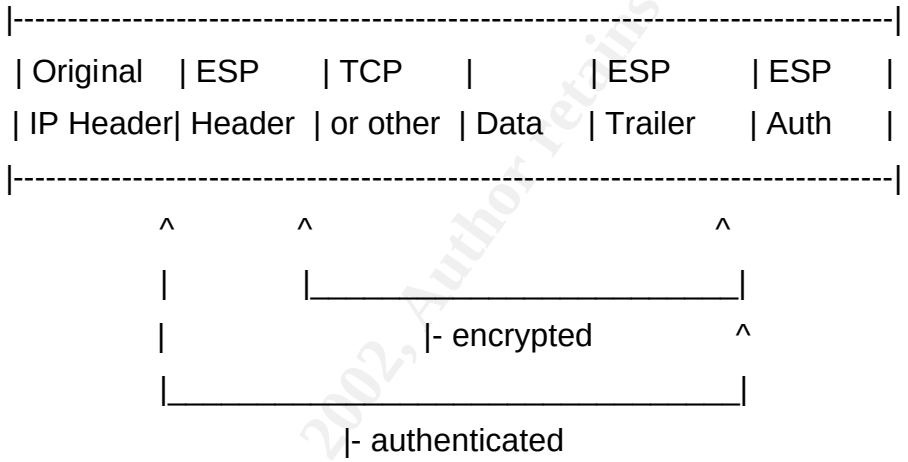

Figure 5-2 Transport Mode ESP[10]

In a nutshell the figures illustrate the problem: when IPsec ESP is used (regardless of the mode), the TCP header and all other upper layer information is encrypted. Thus, PEPs are not able to perform spoofing of packets as they cannot 'see' TCP source and destination ports, or sequence numbers.

At this point it would appear an unsolvable problem exists. In fact (Border, Section 4.1.1) states that "In general, a user or network administrator must choose between using PEPs and using IPsec." While the two technologies are incompatible, they are not necessarily mutually exclusive. As is often the case, security must be traded off against other factors, in this instance performance. The next section describes an architectural compromise and points out the various security and performance trades that result.

[9] Weber, Page 3
[10] Weber, Page 4

6 PEP and IPSec – An Architectural Compromise

Fundamental to IPsec is its end to end architecture. The philosophy is that an end point (i.e. a host or security gateway) must be able to establish a trust relationship with another end point and they do not depend on intermediate nodes or the networks that connect them to exchange information securely. (Kent, Section 4.1) describes such end to end relationships as "security associations". Security associations (SA) can exist using various combinations of hosts and security gateways, in fact nested security associations can also be configured as shown in Figure 6-1 Security Association Example. In this example a remote host (H1) would like to connect securely to host (H2) within a private network. H2 sits behind a security gateway (SG2) which performs the functions of a firewall in addition to an IPsec tunnel end point. SG2 is configured to in turn forward tunneled traffic to H2 where it is then decrypted.

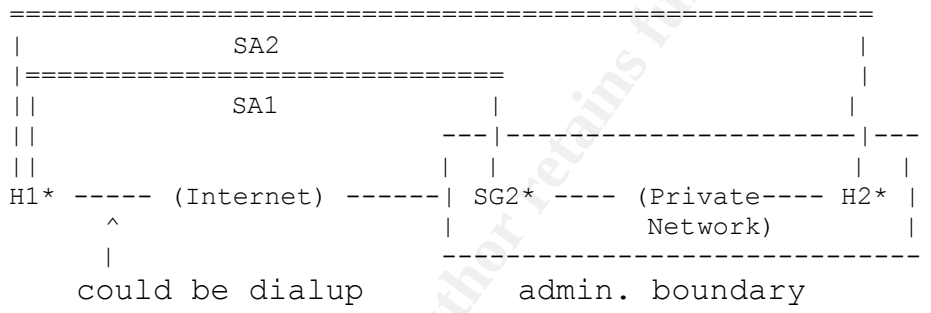

Figure 6-1 Security Association Example[11]

Setting aside the issues with using PEP for the moment, the example shown above could represent a remote host (H1) that accesses a VSAT hub protected by a firewall (SG2). H2 is connected to a VSAT that provides a private network connection via satellite with the hub. Through the hub and via satellite H1 and H2 are able to communicate. Such a configuration is certainly feasible: IPsec *can* be used over satellite links, the performance however will greatly suffer without the use of PEPs.

As an alternative, consider the configuration shown in Figure 6-2 Security Association Terminating at PEP, adapted from the previous figure.

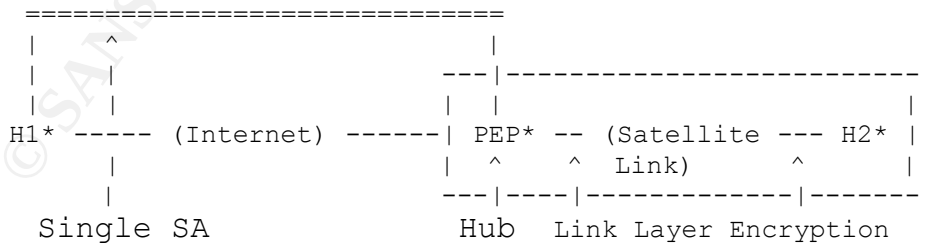

Figure 6-2 Security Association Terminating at PEP

In the configuration shown above, a single IPsec tunnel terminates at the PEP which is playing the role of SG2. Since the IPsec tunnel is terminated at the PEP, the TCP datagram including its header is in the clear, thus allowing the PEP to carry out its

[11] Kent, Section 4.5

performance enhancing functions. The PEP is now able to spoof packets and pretend to be H2, allowing the TCP window size to increase rapidly as discussed in section 4, Performance Enhancing Proxies (PEP). The PEP forwards packets to H2 via the satellite link between itself and H2. Note that even though the IPsec tunnel from H1 terminates at the PEP, the packets sent by the PEP over the satellite link to H2 are **not** sent in the clear. This is due to the fact that satellite link layer encryption can be used after the PEP between the hub and the VSAT at H2 to provide privacy and guard against eavesdropping.

It would thus seem that a solution exists which provides security as well as the requisite performance. Careful examination reveals this is not the case. Recall IPsec's case for end to end security and the issues become apparent. In this configuration, end to end security is broken since the IPsec tunnel terminates at the PEP. This leaves the user's potentially sensitive or valuable data exposed from the point where the tunnel terminates to the point where satellite link layer encryption is performed. Analysis of vulnerabilities shows that attacks can take many forms. Dishonest employees with access to the PEP host(s) or the hub LAN could gather data from each customer that uses the hub. If the hub has poor perimeter defenses for example a poorly configured firewall or has modem connections that essentially bypass the firewall; outside attackers can gain entry to the hub network and hosts. If the PEP host(s) is poorly configured (i.e. unnecessary services running, required patches not installed, poor password policies, and/or poor configuration management) an attacker's job is made easier. If the operational staff of the hub doesn't have proper security training, social engineering attacks could be used to gain information to assist in illegal access to the system. Even the physical security of the site and availability of the hub itself could be an issue – is the site vulnerable to flood, fire, or loss of power?

The hub could be a high value site; for example it may be used by a credit card validation center that performs authorization checks with numerous remote sites connected with VSATs. Thus the threats are real and the resulting risk could be high depending on the number of vulnerabilities that actually exist in a given hub.

The risks of data theft / loss can certainly be mitigated by applying effective counter measures at the hub. (E.g. hardening servers, installing intrusion detection on the network and hosts, regular audits of firewall / system logs, removal of modems inside the firewall, use of a diverse site for disaster recovery, and proper training.) The cost and complexity of implementing such countermeasures must then be traded against the potential loss in revenue should an attack occur.

Assuming that the various hub vulnerabilities are addressed, the challenge still remains to establish credibility with potential customers that their data is indeed secure while in transit. The hub is not the only point of concern; issues about the security of the link layer must be addressed. A secure key exchange protocol is required as well as strong encryption and authentication between the hub and VSATs. It's clear that breaking the end to end security of IPsec does introduce cost, complexity, and risk, but in return the customer is able to benefit from the higher throughput made possible by PEP processing. Properly identifying the system's vulnerabilities and threats should minimize risk to the system and user data.

7 Conclusions

VSAT systems are obviously not a dominant form of data network. However they fill an important niche for the enterprise or home user that requires broadband capability, but has no other alternatives such as cable or DSL. It's evident that the developers of the IPsec protocol were interested in providing a network layer standard which would enable secure VPN solutions to be developed, and the standard has been widely adopted.

This paper has walked through the various issues regarding the use of IPsec when TCP PEPs are in the communication path, culminating in a compromise solution based upon the existing IPsec and TCP standards. Perhaps the most revealing point is that any system under consideration must be carefully analyzed for vulnerabilities even after security mechanisms (IPsec, satellite link layer encryption in this case) have been applied. Without careful examination of the PEPs and hub environment, one might have reached the conclusion that equivalent end to end security has indeed been achieved in the compromise architecture presented.

The next generation of VSAT systems (e.g. systems such as Spaceway[12]) will take advantage of highly advanced satellite payloads that will eliminate the need for a hub, allowing direct user-to-user communication via satellite. The need for security as ubiquitous as IPsec will no doubt increase as more users run more applications over such networks. However the laws of physics and the nature of TCP will continue to dictate the need for PEPs. Users of such systems will demand both security and performance.

Further research in the area of IPsec used in conjunction with PEPs is needed. Some research does exist; one approach would enable PEPs to decrypt just the TCP headers, leaving the rest of the packet protected[13]. Another approach introduces the concept of a transport layer friendly ESP, or alternatively a 'disclosure header'. This would result in certain portions of the TCP header to be exposed so TCP PEPs or other transport layer mechanisms could examine them[14]. Either approach would require a change in the IPsec standard itself and are not complete end to end solutions since some header information is being exposed, but they are a step towards an ultimate solution that would sacrifice neither performance nor security.

[12] Spaceway
[13] Zhang
[14] Bellovin

List Of References

1. DirecWay™ (Hughes Network Systems Inc.). URL:http://www.direcway.com/ (June 25, 2002).

2. Webopedia.com. "VSAT", June 19, 2001, URL: http://inews.webopedia.com/TERM/V/VSAT.html (July 2, 2002).

3. Skycasters.com. "VPN Over Satellite", URL: http://www.skycasters.com/supervpnwhitepaper.htm (July 17, 2002).

4. Comer, E. Douglas. Internetworking With TCP/IP, Volume 1, Fourth Edition. Upper Saddle River: Prentice Hall, 2000. Page 220.

5. Allman, M. et al. "Enhancing TCP Over Satellite Channels Using Standard Mechanisms." RFC2488, January 1999. URL: http://www.faqs.org/rfcs/rfc2488.html, (July 19, 2002).

6. Border, J., et al. "Performance Enhancing Proxies Intended To Mitigate Link-Related Degradations." RFC3135, June 2001. URL: http://www.ietf.org/rfc/rfc3135.txt?number=3135 (June 20, 2002).

7. Copeland, J. "Ethernet/IP/TCP Packet Headers". January 8, 2001. URL: http://www.csc.gatech.edu/~copeland/4005c/slides/ (July 25, 2002)

8. Weber, Chris. "Using IPsec in Windows 2000 and XP". INFOCUS, December 5, 2001. URL: http://online.securityfocus.com/infocus/1519 (July 25, 2002)

9. Kent, S. "Security Architecture for the Internet Protocol." RFC2401, November 1998. URL: http://www.faqs.org/rfcs/rfc2401.html (May 5, 2002).

10. Hughes Network Systems, "Spaceway". URL: http://www.direcway.com/default.asp?CurrentPath=spaceway/overview.htm (August 1, 2002).

11. Zhang, Y., Singh, B. "A Multi-Layer IPsec Protocol". INTERNET DRAFT, April 2000. URL: http://www.wins.hrl.com/people/ygz/ml-ipsec/draft-zhang-ipsec-mlipsec-00.txt (August 3, 2002).

12. Bellovin, Steven M. "Transport Friendly ESP". URL: http://www.research.att.com/~smb/talks/tfesp-ndss/index.htm (August 7, 2002).

Last Updated: May 27th, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Madrid 2017	Madrid, ES	May 29, 2017 - Jun 03, 2017	Live Event
SANS Atlanta 2017	Atlanta, GAUS	May 30, 2017 - Jun 04, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CAUS	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DCUS	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TXUS	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Milan 2017	Milan, IT	Jun 12, 2017 - Jun 17, 2017	Live Event
SEC555: SIEM-Tactical Analytics	San Diego, CAUS	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NCUS	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Secure Europe 2017	Amsterdam, NL	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, COUS	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MNUS	Jun 19, 2017 - Jun 24, 2017	Live Event
DFIR Summit & Training 2017	Austin, TXUS	Jun 22, 2017 - Jun 29, 2017	Live Event
SANS Paris 2017	Paris, FR	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, AU	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MDUS	Jun 26, 2017 - Jul 01, 2017	Live Event
SEC564:Red Team Ops	San Diego, CAUS	Jun 29, 2017 - Jun 30, 2017	Live Event
SANS London July 2017	London, GB	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, JP	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, SG	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CAUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS ICS & Energy-Houston 2017	Houston, TXUS	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, DE	Jul 10, 2017 - Jul 15, 2017	Live Event
SANSFIRE 2017	Washington, DCUS	Jul 22, 2017 - Jul 29, 2017	Live Event
Security Awareness Summit & Training 2017	Nashville, TNUS	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TXUS	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, CZ	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MAUS	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Hyderabad 2017	Hyderabad, IN	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UTUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NYUS	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, ILUS	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Adelaide 2017	Adelaide, AU	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Stockholm 2017	OnlineSE	May 29, 2017 - Jun 03, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced