Global Information Assurance Certification Paper

NAT TRAVERSAL: PEACE AGREEMENT BETWEEN NAT AND IPSEC

August 12, 2001
Haluk AYDIN

Introduction

The first time I have dealt with a case where I had to pass IPSec traffic through a NAT device -a firewall-, and learned the difficulties, I thought that this is such a weird setup that I would never met another one. I was wrong. After meeting two more customers, all with reasonable reasons for such a setup, I decided this topic deserves more attention.

Network Address Translation (NAT) proposed a nice solution to the tight IPv4 address space by allowing use of unregistered IP addresses within the organization. Since it hides the local IP addresses from the outer world, only a small number of registered IP addresses would be enough. This means economy. Besides, NAT adds another security level by preventing access to the local hosts unless you explicitly allow it. Using NAT, when you change your ISP, you never have to renumber your hosts. NAT may also be a good method to translate between IPv4 and IPv6 addresses in the future.

IPSec, on the other hand, is the most widely used VPN standard and being part of IPv6, it will be so in the future. Everyday, more and more companies deploy IPSec to transfer their private information in a secure manner through Internet.

It's a pity but these two very important tools are hard to use together; because of their nature… Now a new approach called NAT Traversal offers a nice solution to the problem.

IPSEC

The de-facto Internet standard Internet Protocol (IP) is not secure. The reason is simple: in the early times of the Internet, the need for security was much less. All types of business conducted through Internet, raised the requirement for a secure protocol. IP Security Working Group of IETF has been working on IPSec (IP Security) protocols to protect the IP traffic on the packet level. The development for IPSec was part of the development efforts for IPv6, but since transition to IPv6 was slow and the need for secure IP communications were increasing, IPSec is modified to support IPv4.

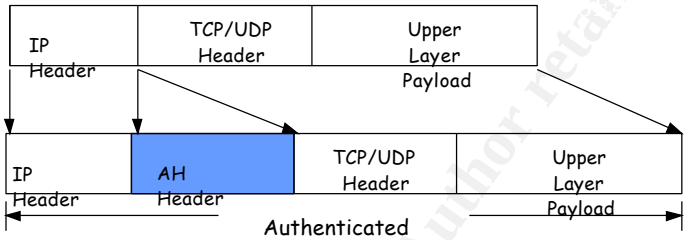
IPSec provides 4 mandatory features for a secure data communication: confidentiality, integrity, authentication and non-repudiation. This is acquired by applying cryptographic algorithms on the IP packet. It employs well-known algorithms like Diffie-Hellman, DES, 3DES, MD5, SHA1, to perform required transformations on the packet. The flexible design of the IPSec architecture allows use of newly developed algorithms and standards without making changes in the protocol.

There are two main transformations that is applied to an IP packet: AH (Authentication Header) and ESP (Encapsulating Security Payload). AH inserts a header into the IP packet including a

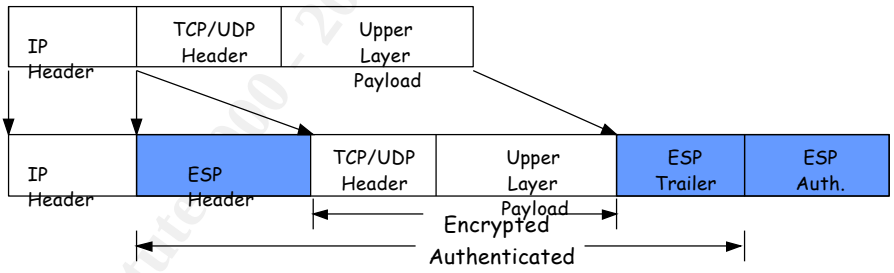
cryptographic checksum of the packet's contents. AH is used only to authenticate the IP packet, and no change is applied to the payload. ESP is used for encrypting the packet payload as well as authenticating it. ESP also inserts a header into the IP packet. AH and ESP could be used together or separately, but using both of them simultaneously increases the overhead without providing a significant gain. But when confidentiality is not a concern, then AH could be used alone.

IPSec can process an IP packet in two different ways: Transport and Tunnel *Modes*. Transport Mode is used for direct Host-to-Host VPN's, where Tunnel mode is used for Host-to-Gateway or Gateway-to-Gateway VPN's. Transport and Tunnel Mode differ in the way they encapsulate the IP packet.

AH in Transport Mode

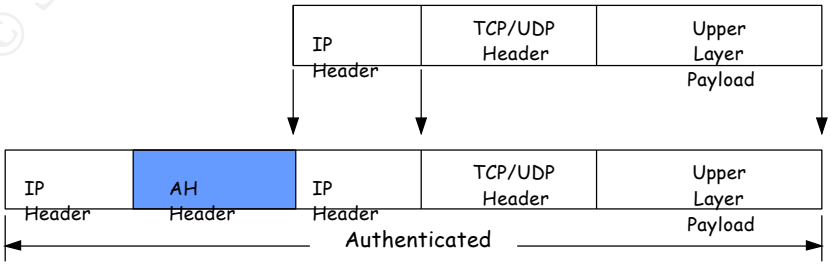


ESP in Transport Mode

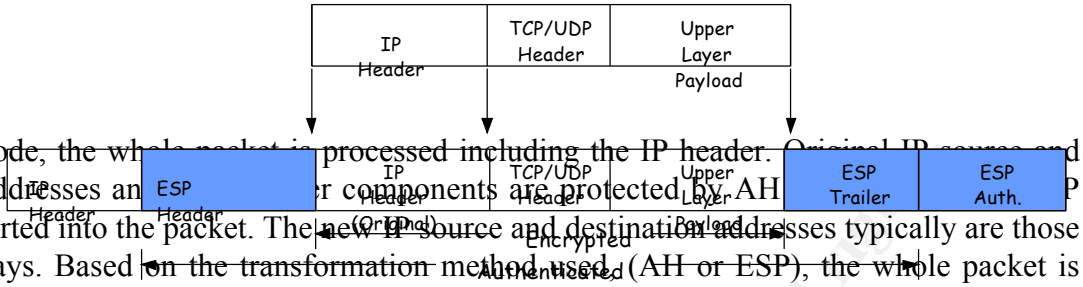


In Transport mode, only the Transport layer of the IP packet is transformed. This transformation means authentication or encryption, or both. When AH is used in Transport mode the whole packet is authenticated but nothing is done to provide confidentiality.

AH in Tunnel Mode



ESP in Tunnel Mode



In Tunnel Mode, the whole packet is processed including the IP header. Original IP source and destination addresses and other components are protected by AH or ESP. A new IP header is inserted into the packet. The new IP source and destination addresses typically are those of the gateways. Based on the transformation method used (AH or ESP), the whole packet is either authenticated, encrypted or both.

One important part of IPSec is key management process. In order to establish a secure communication channel between IPSec peers, keys should be exchanged in a secure way. There are two main ways to accomplish this: manual key exchange and IKE (Internet Key Exchange). Manual key exchange involves manually typed parameters, which makes life easier for a few peers, but management gets harder as the number of IPSec points increased. IKE is a widely used standard for scalable VPN architectures. IKE provides an automated way to negotiate and establish SA's (Security Associations), which contain parameters like encryption and hash algorithms, keys, and encapsulation methods. IKE also enables communicating parties to verify the identities of each other; and to exchange keys. IKE communicates over UDP port 500.

IKE negotiation takes place in two phases: Phase 1 and Phase 2. Phase 1 is the initial negotiation and an SA is established for IKE, so that the peers agree on a common ground for securely exchanging IKE data. In phase 2, another SA is established for IPSec communication. Phase 1 can employ one of the two modes: Main mode or Aggressive mode. In main mode besides all of the other parameter exchange, the identities of the peers are verified. In aggressive mode the identities are not verified, which makes it a faster method. Phase 2 is accomplished using Quick Mode.

NAT

Network Address Translation is basically a conversion applied to the source and destination IP addresses of an IP packet. Generally registered -or routable- IP addresses are replaced by unregistered IP addresses and vice versa. The reason is obvious: the scarce IPv4 space. NAT is generally implemented on gateway devices like routers, broadband routers, firewalls and the like. Thus a whole private network with unregistered IP addresses can access to Internet since the source IP addresses of the outbound packets are replaced by the NAT device with a registered address, such that the modified packets can be routed properly. Similarly, the destination IP address of inbound packets is replaced by the address of the ultimate destination.

There are 2 main types of NAT: Static NAT, and Port Address Translation (PAT). The latter is sometimes referred as Network Address Port Translation (NAPT). Static NAT is simple one-to-one conversion, where each unregistered IP address is mapped to a registered IP address. NAPT provides a mapping from multiple IP addresses to a single registered IP address. The source port of the outgoing packet is mapped dynamically to an available value. Thus a whole network is hidden behind a single IP address. Some vendors also employ a third method called Pooled NAT.

In this method the internal IP addresses are mapped to an allocated pool IP addresses. NAT device keeps a translation table where it records the source IP address and the source port of an outbound packet. Then it replaces these with the new ones, and these values are also recorded. The return packet is compared to this table to find the matching entry, and the destination IP address and port number is replaced accordingly.

The Problem

Remember that one of the main objectives of IPSec is to protect an IP packet's INTEGRITY. This means that IPSec tries to prevent any modifications to the packet. Remember how NAT works: It modifies the IP header! When do you face such a setup? Many times: When you are a traveling employee, and you want to reach your Enterprise network through VPN, with your notebook attached to one of your customer's network. Or when you are a telecommuter trying to reach your corporate servers from behind an ISDN router with NAT enabled.

Either in Transport or in Tunnel mode, AH authenticates the whole IP datagram. Contrary to ESP authentication, AH also authenticates the IP header that precedes itself. When NAT modifies the IP header, IPSec evaluates this as a violation of integrity and discards the packet. Thus AH and IPSec can never work together. Then we are left with two possibilities: ESP in Transport mode or ESP in Tunnel mode.

ESP in Transport mode protects the TCP/UDP header, but does not care about the source and destination IP addresses. Thus, modification of the IP address does not violate the integrity check. But if the packet is a TCP or UDP packet, NAT should modify the checksum that is protected by ESP, which in turn causes the integrity check to fail.

This leaves us with only one working solution: ESP in Tunnel mode. But there may be other problems regarding IKE. If you are using IKE in Main Mode, and if IKE authentication employs pre-shared keys, then you are in trouble. Since Main Mode requires peer authentication, and pre-shared key IKE authentication includes the host IP addresses, a NAT device in the middle would cause IKE authentication to fail.

Resolution

There may be some workarounds and interim solutions for the problem, but we will discuss the most promising solution: NAT Traversal.

NAT Traversal is a draft proposed to IETF by two different groups. One is prepared by SSH Communications, and the other is a result of a joint workshop of vendors like F-Secure, Microsoft, Cisco and Nortel. Two proposals are merged to form a single standard after a meeting in March 2001, since they had many common points. Although this is a new concept, major VPN vendors started using NAT Traversal in their products.

NAT Traversal is designed as a simple solution, which does not require any changes in the middle devices and existing protocols. Only requirement is the support for NAT Traversal at the edge

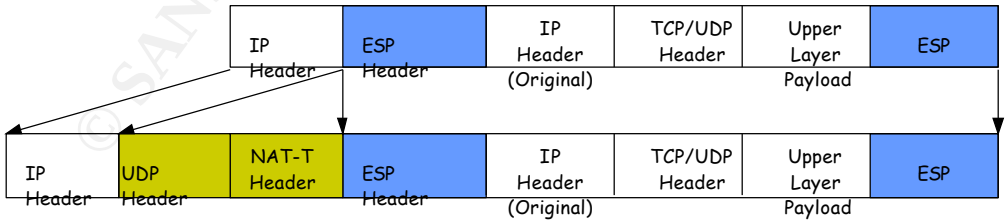
devices. It also provides an automated way to accomplish NAT Traversal procedures, which minimizes the user intervention. The first step of NAT Traversal operation is determining whether the communicating parties support NAT Traversal or not. This is done in the first phase of IKE negotiation, by sending a special vendor ID string (MD5 hash of "draft-ietf-ipsec-nat-t-ike-00" - ["4485152d 18b6bbcd 0be8a846 9579ddcc"]) to the other side. Successful exchange of vendor strings indicates that both sides support NAT Traversal.

Upon verifying both parties support NAT Traversal, the second step is to discover if there is a NAT device in between. NAT discovery is also used to discover which peer is behind NAT, so that only that side sends the keep-alive messages. NAT discovery is a procedure to determine if the IP addresses or the ports change along the path. To accomplish this both sides calculates and sends the hashes of IP addresses and ports of both the source and destination to each other. Peers compare these values and judges that there is a NAT device on the path if they cannot find a match. These hashes are sent in NAT Discovery (NAT-D) payloads. NAT-D payloads are inserted in the third and forth messages in main mode, and second and third messages in aggressive mode.

When a NAT device is discovered on the path, NAT-T negotiation and decision to use NAT-T takes place in quick mode. NAT-T negotiation decides what encapsulation mode is going to be used for NAT-T. These are UDP-Encapsulated-Tunnel and UDP-Encapsulated-Transport to replace normal tunnel and transport modes. Peers also send their original IP addresses if needed. In transport mode, peers should send their original IP addresses and ports whereas in tunnel mode peers should not, since they are included in the ESP payload. AH is supported in the draft because of the requirements of the standards but it is not expected to be implemented.

If there is a NAT device on the path, something must be done to pass IPSec traffic through this device. This is the heart of the solution: the IPSec traffic between the hosts is encapsulated in UDP using the IKE port. Thus the encapsulated packets follow the same route as IKE packets. This avoids further modifications in the firewalls and assures that the NAT device modifies the NAT-T (NAT Traversal) packets, the same way it modifies the IKE packets.

NAT-T Encapsulation



Encapsulating is done by inserting a UDP header right after the “minimal” IPv4 header, which contains the source and the perceived destination IP addresses. UDP source port is 500 (IKE port). This UDP header overwrites the 8 bites of IKE-initiator-cookie-field by all zeros, which is not allowed normally. This provides the ability to differentiate normal IKE traffic and NAT-T traffic. Original IP header length and protocol type is stored in NAT-T header. Finally the IP Header length is set to 20 bytes and the protocol is set to UDP.

Decapsulation is done the reverse way, and it takes place before further processing. The incoming packet is checked to see whether it is a NAT-T packet or not. This is done by looking for the following conditions: protocol type is UDP, destination port is IKE (500) and IKE-initiator-cookie field is all zeros. The IP ports of the incoming packet is compared to the IKE/IPSec data to find a match, in order to confirm that this packet is part of a previously negotiated SA. The IP addresses are replaced by that of the IPSec peers, other IP header information is extracted from the NAT-T headers, and the UDP and NAT-T information is removed from the packet.

Since NAT devices can clear port assignments after a period of inactivity, a still open NAT-T session may be broken. The reason is obvious. When a new packet arrives after a certain period of inactivity, NAT device may assign a new dynamic source port for the packet, which cause the check in decapsulation phase fail. To avoid such a problem, keep alive packets should be sent from the peer, which is behind the NAT device. In SSH Communication's proposal the period for keep alive packets is 9 seconds. If no keep alive packet is received from the corresponding peer for a certain period of time, the SA is killed prematurely. The keep alive packet's payload is of no importance.

Of course all of these does have an overhead. This is about 200 bytes for the Phase 1 IKE negotiation and 20 bytes for each packet. The processing time can also be an issue. In addition to these, there is the overhead of the timeout packets, which is sent every 9 seconds. But as compared to the simplicity and functionality NAT-T provides, this overhead may be regarded as negligible, or at least as tolerable unless the network is too slow.

Another issue needs resolution: Since the unregistered IP addresses are used in the local networks, there is a possibility that a server has two clients with the same IP addresses, from two distant networks. This may cause a problem since the clients are mapped based on IP address, port and protocol. In the Internet Draft, Built-in-NAT is offered as a solution to the problem. This technique relies on another NAT conversion, which is applied to the incoming packet after IPSec process. The configuration may need careful network redesign though.

There may be some security issues with the solution. NAT-T negotiation does not authenticate hosts and is open to everybody. It may also leave the IPSec server prone to DoS attacks since one host can start negotiation from 65535 ports, which may easily overflow the server. Finally an attacker can achieve internal IP address information since the hashes of the IP addresses are negotiated and since it is not a significant job to scan the whole address range.

NAT-T cannot cope with protocols like FTP, and LDAP. These and some other protocols include IP addresses of the peers embedded in the application level of the packet. Normally this part of the payload is encrypted and there is no way that NAT-T modifies the contents. Precautions should be taken to avoid these type of problems.

Summary

NAT-T is an Internet Draft proposed to IETF to overcome the problems faced when an IPSec

traffic is intended to pass through a NAT device. NAT and IPSec is very hard to use together without a standard way which would provide simple management and interoperability between vendors. After merging two different works from different vendors, NAT-T is the most promising solution for the near future so that some vendors started implementing it in their VPN products.

References

Kivinen, et. al. Negotiation of NAT-Traversal in the IKE, Internet Draft, 2001
http://search.ietf.org/internet-drafts/draft-ietf-ipsec-nat-t-ike-00.txt

Dixon, et. al. IPSec over NAT Justification for UDP Encapsulation, Internet Draft, 2001
http://search.ietf.org/internet-drafts/draft-ietf-ipsec-udp-encaps-justification-00.txt

Huttunen, et. al. IPsec ESP Encapsulation in UDP for NAT Traversal. Internet Draft, 2001
http://search.ietf.org/internet-drafts/draft-huttunen-ipsec-esp-in-udp-01.txt

Stenberg, et.al. IPsec NAT-Traversal. Internet Draft, 2001
http://search.ietf.org/internet-drafts/draft-stenberg-ipsec-nat-traversal-02.txt

SSH Communications, NAT Traversal Toolkit ,White Paper, 2001
http://www.ssh.com/tech/whitepapers/SSH_NAT_Traversal_Toolkit.pdf

Fratto, Mike. Why Can't IPsec and NAT Just Get Along, Network Computing, 2000
http://www.networkcomputing.com/1123/1123ws2.html

Phifer, Lisa. IP Security and NAT: Oil and Water?, ISP Planet, 2000
http://www.isp-planet.com/technology/nat_ipsec.html

Phifer, Lisa. Slipping IPsec Past NAT, ISP Planet, 2001
http://www.isp-planet.com/technology/2001/ipsec_nat.html

Fratto, Mike. Network Address Translation: Hiding in Plain Sight. Network Computing
http://www.networkcomputing.com/917/917ws2.html

Garcia, Rafael. Network Address Translation – A real solution? SANS Reading Room, 2001
http://www.sans.org/infosecFAQ/encryption/NAT.htm

Kosiur,David. Building and Managing Virtual Private Networks, John Wiley & Sons, 1998