Interested in learning more about security?

SANS Institute InfoSec Reading Room

Remote Access IPSec VPNs: Pros and Cons of 2 Common Clients

The needs for remote access in today's enterprise networks require a cost effective method for securely connecting to company resources via the Internet. IPSec is one of the best methods of creating an encrypted, authenticated tunnel to these resources, but at the same time, the current IPSec standards do not specify a method of providing clients an internal IP configuration nor a method for authenticating more than the computer that the user is currently using for the connection. This paper discusses two client option...

GIAC Certification

GSEC Practical Assignment
Version 1.4b Option 1

Remote Access IPSec VPNs: Pros and Cons of 2 Common Clients

prepared by Jason Everard

Abstract:

The needs for remote access in today's enterprise networks require a cost-effective method for securely connecting to company resources via the Internet. IPSec is one of the best methods of creating an encrypted, authenticated tunnel to these resources, but at the same time, the current IPSec standards do not specify a method of providing clients an internal IP configuration nor a method for authenticating more than the computer that the user is currently using for the connection. This paper discusses two client options for creating this encrypted and authenticated connection, as well as options for working around the deficiencies of the current IPSec standard by combining IPSec with L2TP or by using proprietary functions to accomplish the same. Other proprietary features are discussed in this paper, such as IPSec NAT traversal, client firewall inclusion, and user authentication via the ISAKMP tunnel.

The need for remote access

In a world that is growing ever more interconnected, the ability to maintain continual access to people and data becomes the benchmark by which people measure their productivity minute by minute. Providing this access to users means, on the one hand, making it possible for people to access their data from anywhere in the world, and, on the other hand, this means ensuring that the data which is sent and received is protected from compromise and unauthorized alteration. In addition to this, the availability of this critical data should also be maintained. In other words, only authorized staff should have access to their data and no one else should be able to either view, alter, or keep people from accessing their data in a timely manner.

The best way to provide such access has taken on many forms in the past few decades, from dedicated front-end access to mainframes to the current shared Intranet LAN and emerging Remote Access IPSec VPN scenarios this paper intends to discuss.

In general, the intent in setting up remote access infrastructures is to make the user's connection as much like a dedicated, point-to-point connection as possible and to undertake precautionary measures to ensure that interception of communications does not yield any useful data for the attacker.

Providing a ubiquitous connection is one aspect of the problem, which the Internet will solve if it has not already. The other main aspects of the remote access problem are authentication and authorization of both the user and her computer as well as the VPN gateway the user is connecting to. A final aspect of the remote access problem is protecting the remote access VPN resource from viruses, trojans, worms, and flooding attacks. These aspects are detailed in the next sections.

The dangers of remote access

Remote access has always been risky. We, as security administrators, must protect the company's critical resources from external and internal threats, invariably with a mix of technologies and a large measure of trust in the employees who access this data.

The goal in securing our information systems is to make the attacker get close to us in order to actually try to gain access to our sensitive data. In other words, an attacker can't hack into a company from the internet if that company does not have an internet connection as a practical security measure, but if I glue a cell phone to the bottom of a desk in the company's boardroom, turn off the ringer, and program the phone to automatically answer after one ring, then I can listen in on the company's most intimate details with only a minor level of risk and technical ability. So we, as security administrators, try to maintain an acceptable level of risk by creating a closed user group and controlling who and what is allowed to physically access our resources so as to minimize possible damage through a breach of trust.

Remote access takes the notion of creating a closed user group and combines it with the possibility of including every possible computer user in the world in that closed user group, as long as she has the appropriate hardware to connect to the shared infrastructure at the heart of such a remote access setup. It is extremely paradoxical at best to provide a private infrastructure on a shared medium, but this is what we have to do on a daily basis as security administrators, and we need a good dose of encryption, authentication, and authorization to get it done. Also keep in mind that we need auditing mechanisms as well, so that we can be sure that the whole setup is working and that we haven't just fooled ourselves into thinking that no one has compromised the security of the remote access setup through technological, physical, or social means, because we can't image how someone could possibly do that or why.

For reasons of brevity, possible auditing processes are not discussed in this paper. Instead, we will focus on the some of the methods available for providing an encrypted and authenticated tunnel with dose of authorization, which is appropriate for the majority of our business applications.

Not all forms of remote access connections are encrypted. Those that are not, PPP connections, for example, are usually based on a circuit-switched point-to-point infrastructure, such as POTS (Plain Old Telephone Service), and employ an authentication mechanism of some sort, such as PAP, CHAP, or MSCHAP. Data encryption is not an option in PPP and the security administrator is forced to rely on strong authentication (either CHAP, MSCHAP, or some combination of PAP and one-time-password tokens) to protect the internal resources from attack by unauthorized users. The ability of an attacker to listen in on such a connection depends on the proximity and

level of access the attacker has to the infrastructure the user employs to connect to her VPN.

In general, the majority of security administrators deem such connections resistant to eavesdropping due to the relative improbability of someone, either telecom workers or others, tapping the lines at the central office or someone tapping the line close to either of the endpoints of the connection. These dialup connections are widely used today but are expensive for a company to set up and administer due to equipment and connection costs.

Another solution, which has found more and more approval in the e-world during the past few years, is the use of IPSec to create a tunnel through the Internet from the user's PC to her corporate gateway in order to provide the same or better security as a traditional point-to-point connection with fewer equipment and connection costs. This is a very promising method of remote access and is the subject of the rest of this paper.

IPSec As the partial solution

IPSec is a growing collection of standards, which are described in detail in several RFCs and drafts of RFCs published by the Internet Engineering Task Force, whose website can be found at www.ietf.org. The various security mechanisms specified in the numerous IPSec standard documents have been subjected to rigorous peer review and constitute the most secure method of providing connectivity in IP networks to date.

By using the Internet as the shared infrastructure for providing the VPN access, a user can connect to the Internet via a provider of her choice and then initiate an IPSec tunnel to her VPN gateway, typically by activating a software program to connect to the remote gateway after connecting to the regional ISP using PPP or some derivative.

The IPSec connection consists usually of at least 3 tunnels; a management tunnel, called the ISAKMP or IKE tunnel, and two IPSec tunnels – one for each direction of the data flow (one IPSec tunnel, which is initiated from the gateway to the client and one IPSec tunnel, which is initiated from the client to the gateway). The management tunnel is used to authenticate the tunnel endpoints, create dynamic key material and set up the IPSec tunnels for the actual user traffic.

The individual encryption and authentication mechanisms in IPSec and how they work are beyond the scope of this paper as there have been thousands of pages written on various options in the protocols used for IPSec tunnels, I couldn't hope to scratch the surface in this relatively small paper. For more detailed information, please refer to the appropriate RFCs located in the references section of this work.

It is enough to say that the encryption and authentication protocols, which are available to security administrators in the vast majority of IPSec

implementations, are appropriate for normal business applications and information sensitivities. IPSec is a good solution for remote access via the Internet because the protocols provide a level of security which is acceptable for most scenarios.

In addition, the costs of connecting are much lower for the user and the company due to the facts that the user can connect to a regional provider, thereby saving long-distance phone charges, and the company can use its (in most cases) existent Internet connection to terminate the users' tunnels.

Sounds like we have found a solution without any associated problems or drawbacks! Not quite.

What is lacking in IPSec Remote Access tunnels and how to get it

A typical IPSec connection scenario might look like this: The user dials up to a regional ISP and gets an IP address in the provider's address space, which enables the user to connect to her VPN gateway and authenticate the connection. The authentication can be done using a pre-shared secret, called a pre-shared key, or through the use of public/private keys in combination with digital certificates, which state the authenticity of the public keys and thereby establish their owner. The exact details of these authentication methods are also beyond the scope of this paper. For more information on ISAKMP authentication options, please refer to RFC 2409, located at http://www.ietf.org/rfc/rfc2409.txt?number=2409.

It is enough for the purposes of this paper to say that pre-shared key authentication is the easiest to set up, but the most difficult to maintain if pre-shared secret ever needs be changed, (it is good security practice to change the keys at regular intervals), or if the key is ever compromised. Usually, the devices in the same user-group will have the same pre-shared secret, and thus all of them must be reconfigured with a new secret key if the key is ever discovered by an external entity, in order to maintain the integrity of the authentication mechanism.

Notice I mentioned that the same key is usually configured on all members of the same user-group. This is to improve the manageability of network devices. ISAKMP defines mutual authentication, which is a very good idea in a shared infrastructure in order to avoid simple man-in-the-middle attacks. Man-in-the-middle attacks are possible in authentication models, such as PPP, where the user is authenticated but the identity of the gateway is assumed to be consistent and unscathed by redirection, interception, or even outright replacement, thus making it possible for a hacker to fool the user into thinking that she is connecting to her gateway when, in reality, she is connecting to the attacker and is feeding him with her username and password. The attacker can then use this information to gain access to the user's restricted resource.

In an effort to disarm man-in-the-middle attacks, ISAKMP authentication must always be mutual. Due to this, we security administrators are forced to either a) configure a different pre-shared key for every client/VPN device pair, which

does not scale at all, or b) use the same pre-shared key for all clients and VPN devices, that need to talk to each other. It turns out that the latter is much better and doesn't really decrease security at all when used with other, stronger, user-based authentication methods, which are available when IPSec is used in conjunction with other tunneling protocols and proprietary extensions to the ISAKMP protocol. These authentication methods are discussed later in this paper.

The use of a PKI (Public Key Infrastructure) to issue certificates to each device in the network for ISAKMP end-point authentication is another option and is also a noble undertaking, but unfortunately not the definitive answer to our authentication woes. The idea is that a CA (Certificate Authority) certifies the public keys of every device in the network participating in the VPN and issues a certificate to this effect to the device and/or puts it in a shared directory structure, where the whole world can see it, or at least the part of the world that needs to. The IPSec end-points authenticate each other by "signing" a hash using that end-point's private key (their own private key), which only that end-point should have.

The hash, which is signed, is generated from random variables produced by the other side of the connection, so as to force the signer to perform the operation on the fly and thus eliminate any opportunity for practicing the operation or replaying a previous operation to gain access. The other side of the connection confirms the validity of the signed hash by decrypting it with the corresponding public key and comparing the resultant hash with a locally calculated hash using the same random values.

The assumption is that if the hash which resulted from decrypting the "signed" hash (actually called a "signature") with the user's public key is the same as the hash, which was locally produced using the same random values, then the user who produced the signature has the corresponding private key.

The public key, which is used to perform the decryption function, is certified by a third party (the CA) as the public encryption key assigned to a certain entity. This entity can be a person, a machine, a business, or almost anything. The certification is performed through the use of a certificate, usually an x509 v3 Certificate, which is generated by the CA and signed with the CA's private key.

The CA's public key is used to verify any certificates, which claim to come from that specific CA. (please refer to Applied Cryptography, by Bruce Schneier for more info on CA and PKI) After verifying the authentication hash by using the public key in the user's x509 v3 certificate, most assume at this point that the user who produced the hash is the same user that the private key belongs to, but this isn't necessarily the case.

The entity to which the certificate has been issued by the CA usually contains a user's name, but it is usually stored on the user's laptop used to connect to the corporate network via the VPN. So, in effect, the certificate is issued to the laptop and not necessarily the user.

ISAKMP authentication has been implemented by everyone as a means for authenticating the remote machine, not the user sitting at it. Because the pre-shared key or the public/private key pair is usually stored on the local machine (laptop, router, PDA) if the local machine just happens to be stolen or in use by an unauthorized entity, then we may never know this from the gateway side of the connection.

For both pre-shared and public/private keys, revocation of the keys must be initiated by the user whose computer was compromised, which will invariably cause a period in which the key is known by an unauthorized individual and neither the administrator nor the user have any indication of this.

There are a variety of additional security measures people can use to decrease the likelihood that the key, whether pre-shared secret or private RSA key, will be usable by an unauthorized user in the event that the device is stolen on which the key is stored. These include protecting the key with a password or passphrase, and storing the key on smart card or some other portable storage media.

These measure only increase security if the users diligently observe the acceptable usage policy for them. This means that if you protect your private RSA key with a passphrase, that this passphrase should not be your name spelled backwards or some thing else, which would be easily guessed or hacked through a dictionary attack, or that if you store your secret key on some portable smart card, don't leave the card in the laptop all the time, which is what any busy user will tend to do.

When the key is finally reported stolen, in the case of public/private keys with a certificate authority, the security administrator can include the public key's certificate in a list of revoked certificates, called a CRL (Certificate Revocation List), which basically states that the public key and its associated private key can no longer be trusted. The problem here is that most devices will wait until the former CRL has expired before requesting a new one, which increases the time that a compromised key can be effectively used to gain access to the remote network.

There has been some mention of the need to develop an open protocol for online verification of certificates by gateways and users alike, such as the Online Certificate Status Protocol specified in RFC 2560, located at http://www.ietf.org/rfc/rfc2560.txt?number=2560 , but this still does not eliminate the gap between the time when the key was stolen and the time the user notices it and reports the key as stolen.

Still more time will elapse before the overworked security administrator gets around to canceling or reconfiguring all the effected devices with a new key, in the case of pre-shared keys, or before the administrator can update all the affected devices with a new CRL, in the case of certificates.

It is clear we need something else to authenticate the user sitting at the device connecting to the VPN gateway. This issue will be discussed in the next section.

Before we discuss the solutions, I am not quite finished with the inadequacies of standard IPSec for remote access. The issues of providing the user with an IP configuration, i.e. IP address, WINS Server IP, DNS Server IPs, and user firewall protection from attacks, which come from the internet, still need to be mentioned first.

The inevitable presence of IP-based access control lists on the various gateways between the tunnel terminating gateway and the actual resource the user wants to connect to in order to exchange data means that we have to get the VPN client machine to fit through the various filter lists between the VPN gateway and the user's server.

The filters are necessary because companies need defense-in-depth, meaning that one should not assume that just because the company has a "secure" VPN setup, that everything is indeed secure. One good way to enhance security is to allow only private IP addresses to access company critical resources. This hinders anyone who does manage to find a hole in the company's perimeter defenses from directly accessing resources on the inside with the attacker's actual source IP address, thus preventing direct duplex communication. The IPSec standards, however, do not mention a standard method by which clients can get an IP configuration. There are IPSec drafts concerning this, for instance http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-04.txt, but it is only a draft standard and is not yet required in order for an implementation to be IPSec compliant.

This means that we have two options: 1) use another tunneling protocol on top of IPSec, such as L2TP, which supports IP configuration parameters, or 2) use proprietary mechanisms to get the job done directly in the ISAKMP management tunnel. I will discuss both of these options in the next sections.

We, as security administrators, configure the gateways and can therefore be relatively sure that they are secure by performing audits, both internal and external. Once they are secure, no one should meddle with their configuration without having prior approval from the appropriate authority.

By the same token, we define acceptable usage policies for users and their workstations, make sure that certain software installations or settings are made and spend a lot of time worrying about the inevitable security vulnerability or, even worse, rogue program trying collect confidential info or just simply thrash our internal network. The users should be educated on the risks associated with downloading programs from the Internet, opening mail with strange-looking attachments from anyone, etc.

We can even go one step further by requiring a client firewall on the users' PCs, so that the user is less likely to be attacked when she is surfing around

outside of the VPN, thus also reducing the likelihood that the user will bring something undesirable in with her via the VPN connection.

We can take this whole thing only so far. If the user actively decides to ignore company policy by deactivating the firewall or by even installing rogue programs for some dubious purpose or another, then we have a trust problem and no IT security countermeasure will prevent them from trying.

The idea, therefore, is not to keep authorized, evil users from doing something naughty once they're in the VPN, rather to keep the authorized, honest users from being attacked when they try to check the current status of their failing stocks when they aren't connected to the company's VPN. The options for providing this protection will be discussed later in this paper.

Client Number 1: L2TP/IPSec

L2TP stands for Layer 2 Tunneling Protocol and is specified in RFC 2661, located at http://www.ietf.org/rfc/rfc2661.txt?number=2661 . It is the successor to the proprietary PPTP from Microsoft and the proprietary L2F from Cisco Systems. The idea behind these three protocols is the same: to provide a tunneled layer-2 connection via an established layer-3 connection, such as the Internet.

The tunneled layer-2 connection is typically a PPP connection, and the layer-3 connection is IP. The layer-2 connection is tunneled over IP (usually via the Internet) and terminated at a gateway. This gateway is then able to provide the user at the other end of the tunnel an IP configuration on top of the user's current IP after authenticating her using any of the available PPP authentication mechanisms.

This all happens via the tunneled layer-2 connection and the layer-3 connection is only used as a means of transport for the layer-2 frames. These tunneled layer-2 frames also carry their own layer-3 data, which entail private addresses and sensitive company data in the clear.

Although the data is tunneled in the clear, i.e. unencrypted, we still can use L2TP's PPP authentication options (please refer to RFC 1334 for more information on PPP authentication mechanisms, located at http://www.ietf.org/rfc/rfc1334.txt?number=1334) to authenticate the user at the other end of the tunnel by requiring a username and password, which could also be combined with a one-time password system, such as a hardware token, to make it more secure than simply using static or ageing passwords. So we cut off our nose to spite our face, i.e. we tunnel sensitive data in clear text just so we can authenticate the user with a PPP connection? Not necessarily.

This is where IPSec comes in handy. Like I said, the layer-3 connection is only used for transporting the tunneled layer-2 frames across the Internet to the home-gateway. If we encapsulate this layer-3 connection in IPSec and send it over, then we have effectively encrypted and authenticated the L2TP

data and have solved the problems of authenticating the user and getting the client an internal address as well, i.e. IPSec protects the data and L2TP provides for user authentication and an internal IP configuration for the VPN connection. This is L2TP over IPSec in a nutshell and people use it because it is a standard, it has been tested, and it works.

The part about L2TP being a standard is a good thing for interoperability in heterogeneous networks. If you are employing a multi-vendor network, then interoperability between vendors is extremely important. In general, people have a variety of vendors in their networks and it is a convincing argument, that security and performance are improved by implementing multiple vendors in your network. For me, it all boils down to the fact that competition encourages the development of quality products through a diversified market.

Besides, hardly anyone has a single-vendor network, anyway. There is always some old switch, router, or PC sitting in the corner, working away as it has for six years, doing it's business faithfully while the company that produced it has been bought out, gone bankrupt, or just simply disappeared. Naturally, you'll hope that the box employs only standard protocols for the sake of forward compatibility. Although you probably won't be terminating an L2TP/IPSec connection on one of these boxes, it just serves to make the point that we all have heterogeneous networks, and interoperability is a good investment in the future of your company's network efficiency.

Standards have been tested. This means that the underlying protocols used in L2TP and IPSec have been tested and found to be relatively secure for business applications, meaning that there are no real working attacks on the protocol mechanisms employed to create and maintain the tunnels. As long as the encryption algorithm and key length specified raise the amount of effort required to break the encryption above the abilities of the attacker or beyond the commensurate value of the data protected, then the protocol should be immune to circumventing the authentication method or hijacking an authenticated connection.

There are working implementations of L2TP/IPSec which large, international companies employ to protect their remote access connections. Albeit, these "working implementations" are all Windows clients connecting to various routers, such as Cisco Systems, Nortel, and others - so much for a multi-vendor environment on the client side of the connection.

If we lived in a perfect world, everyone who uses L2TP would use it for the reasons mentioned above, there are however other reasons why people might be forced to use L2TP. Some corporations require that software must first be tested by the security department before it is employed in the company's network. This testing can be very expensive and time consuming and might lead the network security engineers to decide to use the built-in L2TP/IPSec client which comes with Windows 2000 and Windows XP clients.

Ironically, the Windows OS has been approved by the IT security department, and it will most likely have more security holes in it than any small piece of

proprietary security software ever will. But, due to costs, we are forced to lower our expectations, bite the bullet, and just use the built-in client on our bug-ridden laptops instead of taking the additional risk of using another piece of software on top of our bug-ridden OS, especially if we don't have the time or money to test the other piece of software for defects or unwanted side-effects.

Personally, I can understand this argument and I accept it as unfortunate in the light of some of the cool, proprietary features offered by some vendors but I also see the decision to use the Windows L2TP/IPSec client also as necessary in view of lacking monetary and/or personal resources to test the alternatives first.

Another reason for using the Windows L2TP/IPSec client is that the clients' internal addresses sometimes have to be registered in DNS for certain applications to work, which means that the VPN gateway must support a DHCP relay function for the clients IP configuration request via PPP NCPs tunneled in the L2TP connection. Fortunately most gateways do just that for L2TP but not necessarily for other, proprietary clients, such as when a Cisco VPN client connection is terminated at a Cisco router as the VPN device. (The Cisco router as VPN gateway only uses local address pools with the Cisco proprietary VPN Client at the time of this writing, but can indeed perform a DHCP relay function for L2TP/IPSec connections.)

Client Number 2: Cisco VPN Client

The Cisco Systems VPN Client, also called the Unified Framework Client or Unity Client, is a usable, reliable VPN client application from Cisco Systems for more than just Windows platforms. It runs on Linux, Solaris, Darwin, and Windows and offers the user a GUI as well as a command line interface to start and stop the connections, as well as for viewing and troubleshooting the configuration. The newest information on this client can be found at http://www.cisco.com/univercd/cc/td/doc/product/vpn/client/ .

It does however heavily rely on vendor specific extensions to the ISAKMP protocol to such an extent that you can only use a Cisco gateway on the company side to terminate the connection. This is not necessarily a problem for 80% of the companies out there, but many do not envision a Cisco device as a VPN gateway in their company and would much rather have some other product do this job.

There are reasons for using such a client despite the vendor specific attributes, which go against a good policy of employing only open standards in the corporate network. The reasons are mostly associated with 2 main problems that people considering the L2TP client face; 1) L2TP/IPSec does not support VPN connections from behind a gateway performing network address translation (see RFC 1631 for more details, located at http://www.ietf.org/rfc/rfc1631.txt?number=1631), and 2) the Windows L2TP/IPSec client does not provide a built-in firewall function to protect the

user from attack while she is either surfing the Internet or connected the corporate VPN.

The problem with NAT is that basically any Internet gateway performing NAT on either the source or destination addresses will disturb the creation of the IPSec tunnels when used in conjunction with L2TP. The Windows L2TP/IPSec client uses the IP address received on the global outside interface either from the PPP NCP negotiation, DHCP, or static manual configuration as the tunnel endpoint in the ISAKMP Phase II Quick Mode negotiation (see RFC 2409, located at http://www.ietf.org/rfc/rfc2409.txt?number=2409).

This address must correspond to the actual source address in the IP packets that the gateway sees in the ISAKMP negotiation. In short, a gateway between the two can hinder their connection by translating the source address of either side of the connection. Because the two addresses, the one in the ISAKMP negotiation and the one in the layer-3 IP header, don't correspond, it doesn't work and there is no workaround. Take a look at these links for some more sorry tales of trying to get L2TP/IPSec to work from behind a NAT gateway: http://www.sandelman.ottawa.on.ca/ipsec/2001/05/msg00077.html and http://www.sandelman.ottawa.on.ca/ipsec/2001/05/msg00065.html .

A proprietary client, such as the Cisco VPN Client, can create a tunnel from behind a NAT device by using one of many possible options. The reason for this is that the client first receives an internal IP address from the gateway via the ISAKMP tunnel (in a proprietary mode, called Mode Config in Cisco-speak) and uses this address as the tunnel endpoint in the ISAKMP Phase II Quick Mode negotiation. This address is internal to the connection and the NAT gateway never sees it and can thus never translate it either. The NAT gateway only sees the actual external IP the client is using and translates this address instead. This causes the VPN gateway to think that the client's external address is something else than it really is on the client machine, but this doesn't matter because the tunnel endpoint address is the internal one assigned by the VPN gateway, and this stays constant, regardless of NAT devices in between.

We have other problems with NAT gateways and IPSec, though. The problem is that the IPSec protocol ESP, which is used for transporting the users' data, is just that, an IP protocol with its own protocol number. It does not ride on top of TCP or UDP and thus it does not have any source or destination port numbers by which a NAT gateway can track the connection back to the internal host from whence it came. In other words, the ESP packets that the client sends to the gateway are NATted without trouble and then sent to the receiving gateway, where they are usually authenticated and decrypted and the internal IP packet is then sent to its final destination in the corporate network.

The return packets are the problem: the VPN gateway receives the return packet, packs it up in IPSec, and then sends it to the client via the NAT gateway. The NAT gateway then tries to figure out whose IPSec packet it

could be and it doesn't have enough information due to the lack of port numbers in ESP and the fact that the tunnel IDs for the IPSec connections were negotiated via an encrypted ISAKMP tunnel, so the NAT gateway drops the packet. There are a few ways to work around this.

The first option is to configure the NAT device to map IP protocol 50, which is Encapsulating Security Payload (see RFC 2406 for more information on ESP, located at http://www.ietf.org/rfc/rfc2406.txt?number=2406), to a specific host on the inside (the host connecting to the VPN gateway). This will give the NAT gateway the missing information it needed to forward the packet from the VPN gateway to the correct internal host. This works and is a good option for users with a home LAN behind a router performing NAT. The drawback is that only the mapped client can successfully send and receive IPSec packets via the NAT gateway and all the other clients on the inside can only send IPSec packets to their respective VPN gateways and the return packets are always sent back to the mapped client. Therefore, this solution is not ideal for corporate LANs with a gateway to the Internet, which performs NAT, because there will be a number or different IPs on that LAN who want to connect.

Another option for traversing a NAT gateway is to use a proprietary mechanism for "floating" the connection to a UDP or TCP port upon discovery of a NAT gateway between client and VPN gateway. A recent IETF draft describes this mechanism and can be found at http://www.ietf.org/internet-drafts/draft-ietf-ipsec-nat-t-ike-05.txt.

The basic idea is for the client to initiate a connection to its VPN gateway. The VPN gateway in turn can tell if the client's IP is being translated and informs the client of the presence of the NAT gateway which is between the two. The client then multiplexes the ISAKMP connection and both IPSec tunnels together via a high UDP port number (destination port number UDP 4500 for the Cisco VPN Client), after which the NAT gateway can translate the client's IP and track the UDP destination port number and the return traffic back to the correct client on the inside via the client's arbitrarily chosen source port number.

The Cisco VPN Client also comes with a built-in firewall licensed from ZoneLabs (www.zonelabs.com), which the administrator can define as required and "always on." This is a good thing for the average user who also does some Internet surfing from her laptop, because we can protect her from typical connection attempts to non-secured ports.

The Cisco VPN Client also has a proprietary method of authenticating not only the machine and its corresponding gateway, but also the user sitting at the machine. This is accomplished directly in the ISAKMP tunnel by using proprietary mechanisms and is called XAUTH in Cisco-speak.

Conclusion

The client you decide on will depend on many factors, including type of gateway, where the client is intended for use, how often the client will need to

be updated, the level of central IP address management your company requires, and the internal political and procedural requirements the company has implemented. Both of the clients presented in this paper are a reliable and profitable way to enable remote users to connect to their private networks via the Internet.

References:

Harkins, D. and Carrel, D. "The Internet Key Exchange (IKE)." The Internet Engineering Task Force. November, 1998. URL:http://www.ietf.org/rfc/rfc2409.txt?number=2409 (February, 2003)

Schneier, Bruce. Applied Cryptography. Second edition, John Wiley and Sons, 1996.

Myers, M. Ankney, R. Malpani, A Galperin, S. and Adams, C. "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP." The Internet Engineering Task Force. June, 1999. URL: http://www.ietf.org/rfc/rfc2560.txt?number=2560 (February, 2003)

Kaufman, Charlie, editor. "Internet Key Exchange (IKEv2) Protocol." The Internet Engineering Task Force. January, 2003. URL: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-04.txt (February, 2003)

Townsley, W. Valencia, A. Rubens, A. Pall, G. Zorn, G. and Palter, B. "Layer Two Tunneling Protocol "L2TP"." The Internet Engineering Task Force. August, 1999. URL: http://www.ietf.org/rfc/rfc2661.txt?number=2661 (February, 2003)

Lloyd, B. and Simpson, W. "PPP Authentication Protocols." The Internet Engineering Task Force. October, 1992. URL: http://www.ietf.org/rfc/rfc1334.txt?number=1334 (February, 2003)

Egevang, K. and Francis, P. "The IP Network Address Translator (NAT)." The Internet Engineering Task Force. May, 1994. URL: http://www.ietf.org/rfc/rfc1631.txt?number=1631 (February, 2003)

Trace, R. "RE: IPSec over L2TP Tunnels for Remote users." Sandelman Software Works. May, 2001. URL: http://www.sandelman.ottawa.on.ca/ipsec/2001/05/msg00077.html (February, 2003)

Loi, Ly. "NAT traversal clarification." Sandelman Software Works. May, 2001. URL: http://www.sandelman.ottawa.on.ca/ipsec/2001/05/msg00065.html (February, 2003)

Kent, S. and Atkinson, R. "Encapsulating Security Payload." The Internet Engineering Task Force. November, 1998. URL: http://www.ietf.org/rfc/rfc2406.txt?number=2406 (February, 2003)

Kivinen, T. Swander, B. Huttunen, A and Volpe, V. " Negotiation of NAT - Traversal in the IKE." The Internet Engineering Task Force. January, 2003. URL: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-nat-t-ike-05.txt (February, 2003)

Cisco Systems VPN Client documentation. URL:http://www.cisco.com/univercd/cc/td/doc/product/vpn/client/ (February, 2003)

ZoneLabs Homepage: URL: www.zonelabs.com (February, 2003)

Last Updated: October 21st, 2017

Upcoming SANS Training

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

SANS San Diego 2017	San Diego, CAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Seattle 2017	Seattle, WAUS	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, AE	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Milan November 2017	Milan, IT	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Amsterdam 2017	Amsterdam, NL	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Miami 2017	Miami, FLUS	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Paris November 2017	Paris, FR	Nov 13, 2017 - Nov 18, 2017	Live Event
Pen Test Hackfest Summit & Training 2017	Bethesda, MDUS	Nov 13, 2017 - Nov 20, 2017	Live Event
SANS Sydney 2017	Sydney, AU	Nov 13, 2017 - Nov 25, 2017	Live Event
GridEx IV 2017	Online,	Nov 15, 2017 - Nov 16, 2017	Live Event
SANS San Francisco Winter 2017	San Francisco, CAUS	Nov 27, 2017 - Dec 02, 2017	Live Event
SANS London November 2017	London, GB	Nov 27, 2017 - Dec 02, 2017	Live Event
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZUS	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS Khobar 2017	Khobar, SA	Dec 02, 2017 - Dec 07, 2017	Live Event
SANS Austin Winter 2017	Austin, TXUS	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Munich December 2017	Munich, DE	Dec 04, 2017 - Dec 09, 2017	Live Event
European Security Awareness Summit & Training 2017	London, GB	Dec 04, 2017 - Dec 07, 2017	Live Event
SANS Bangalore 2017	Bangalore, IN	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Frankfurt 2017	Frankfurt, DE	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DCUS	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Security East 2018	New Orleans, LAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS SEC460: Enterprise Threat Beta	San Diego, CAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, NL	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SEC599: Defeat Advanced Adversaries	San Francisco, CAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Berlin 2017	OnlineDE	Oct 23, 2017 - Oct 28, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced