Unblocking IPv6 Applications:
Safely Connecting Through Host and Edge Firewalls with IPsec

By William Dixon
President, V6 Security

Executive Summary

Host firewalls have become required to defend against constant attacks from untrusted systems on the Internet and on internal networks. But they threaten the end-to-end benefits IPv6 provides to applications. To enable inbound connections, firewalls currently open holes for an application, which also opens the application and the host to untrusted attack. This paper explains how the IETF design for IP Security (IPsec) policy and Internet Key Exchange (IKEv1 and IKEv2) moderate inbound network access to the host. Thus they enable the host firewall to open holes which can be accessed only by trusted and authorized peers. IPsec-aware firewalls can provide tightly controlled access based on source identity and specific upper-level protocol connection details passed during the IKE negotiation.

Using IPsec no longer requires a ubiquitous public key infrastructure. IKEv2 provides flexible identification and authentication methods, including email addresses, passwords, tokens, non-infrastructure public keys, and Kerberos credentials. Therefore, by combining host IPsec policy with firewall access policy, IKEv2 can be used to negotiate IPsec secure connections for temporary, adhoc application groups, as well as for long-lived communities of trusted hosts. The firewalled hosts in these groups are resilient to untrusted network attacks while providing authorized, secure connectivity for IPv6 applications end-to-end through their host firewalls. A scenario using secure host-to-host file sharing is examined, indicating the points of integration necessary for a seamless user experience. Results of testing this model are presented using Windows XP SP2, along with references to more detailed testing guides and opportunities.

Since many business and home networks are connected to the Internet through edge firewalls, there needs to be an IPv6 solution for edge firewall traversal. This paper reviews mechanisms for traversing the gateway contained in the recently updated IETF IPsec Architecture (RFC4301) and IKEv2 protocols. However, IPv6 hosts are not currently required to implement all of the features necessary for using IKEv1 or IKEv2 and IPsec to traverse the gateway and host firewall. A consensus within the IPv6 community is needed in order to solidify the details for achieving these scenarios and thus update the standardized requirements for IPv6 hosts. If the IPv6 community does not provide a consensus solution to host firewall traversal, then the IPv6 end-to-end benefits for Internet applications may be lost. Similarly, interoperability for a given scenario (such as file sharing) will be difficult to achieve among IPv6 devices, appliances and hosts when deployed within internal networks.

Table of Contents

Caveats and Terminology

This paper is intended for a wide audience. The first half is oriented to those involved primarily at the business level. The second half, starting with the IPv6 scenario topics, assumes knowledge of IPv6 technical design and operation, though not necessarily the details of IKE and IPsec. For brevity, discussion of other firewall traversal techniques, such as SOCKS, Realm-Specific IP and other work could not be covered. The scenarios review what is possible using IKE and IPsec features as defined in the RFCs. It should be understood that current IPv6 IPsec product capabilities may not support certain features or behaviors that would be needed to realize the scenario.

The views expressed here are those of V6 Security only, and are not intended to express the views or product plans of any other company, organization or individual. While V6 Security has thoroughly researched the issues and design(s) presented here, the design(s) may not be a fail-safe solution for all situations.

The terms "secure" and "safe" as used in this paper are intended to mean the design(s) provide reasonable protection from untrusted attackers by requiring source authentication and authorization, and provide the minimal but sufficient application connectivity this operates within limits of acceptable risk. We realize that acceptable risk differs for many circumstances and for many organizations. Also, the untrusted attacker may still find vulnerabilities to exploit in the IPsec and IKE design, implementation, configuration, in the authentication system, or in the credential distribution methods. A defense-in-depth strategy is necessary to sufficiently mitigate risks against both untrusted and trusted attacks.

The terms "network connection" and "network connectivity" can be used to describe having a physical or wireless link connection to a network, such as being "on" the local switched Ethernet network and having achieved network connectivity through a dialup or wireless connection. These terms can also refer to the ability of a host to connect to another host using their respective IP addresses, such as a network connection using a TCP port. The latter usage is intended throughout this paper, except where the connection process with involving the link is specifically mentioned.

The term "end-to-end" as used in this paper is not the same as "peer-to-peer.” End-to-end refers to the IP network path from the originating source IP address to the destination IP address. "Peer-to-peer" often means the application layer path of messages between two clients, which may involve many server hosts and proxied connections. For simplicity in this paper, "peer-to-peer" refers to the scenario of two clients each having a routeable IP address and reachable on the network to each other. Thus "peer-to-peer" file sharing would be accomplished using a TCP connection between the two client hosts.

IP Security (IPsec) is used to refer the capabilities of IPsec policy and IPsec encapsulation protocols in transport mode or tunnel mode as described by The Security Architecture for the Internet Protocol (RFC2401 and the newly revised version RFC4301).

Internet Key Exchange (IKE) is the negotiation, authentication, and key management protocol for IPsec. The term IKE is used to refer to either IKEv1 (RFC2409) or IKEv2 (RFC4306) except where the version number is specifically used.

Legal Notes

The information contained in this document represents the current view of V6 Security on the issues discussed as of the date of publication. The views expressed do not reflect or represent the views of any third parties, including but not limited to any former employers, current business partners or clients of V6 Security. This white paper is provided for informational purposes only, "as is" without any warranties, express or implied. Microsoft is a registered trademark of Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries. Mac OS is a trademark of Apple Computer, Inc., registered in the U.S. and other countries. The names of other companies and products mentioned herein may be the trademarks of their respective owners.

The Business Risk of Connectivity

For the IPv6 transition, organizations must focus even more on the security of network hosts, leveraging all IPv6 mechanisms. A constant concern of security experts and CIOs is the 0-day attack — that automated virus/worm infections which takes down a significant percentage of the machines in their network causing business process interruptions or disabling the business altogether. Gartner's June 2005 survey of 133 large businesses reported that viruses and worms are the #1 perceived security threat, followed by outside hacking or cracking.[1] The 0-day attack is of course possible, takes many forms, and is occurring in specific areas, such as the Trojan infections targeting UK banks and financial businesses reported by the June 15, 2005 briefing from the UK National Security Coordination Centre.[2]

Host network security is also a significant part of the solution to unsolicited commercial (and noncommercial) email (UCE), also known as SPAM. Infected home PCs with Internet connectivity are responsible for approximately 51% of all UCE.[3] While annoying, UCE is also quite dangerous to an individual, to businesses and government organizations. F-Secure reports that phishing attacks through UCE were responsible for $70M USD in losses to German banks; were responsible for one of the largest Internet banks Nordea shutting down; denying online service to some 4 million customers in 8 countries; and were responsible for innumerable attempts to trick people into visiting websites that infect with malware and conduct both individual and organizational identity theft.[4] 2005 estimates from MXLogic and F-Secure are that 68-85% of the mail traffic in the Internet is UCE, and only 4% of UCE is conforming to anti-spam laws (CAN-SPAM). These home PCs could have been infected in many ways. If the PC didn't have an effective firewall, then a direct Internet connection would expose it to frequent attacks.

The SANS Internet Storm Center measures the average time before an infection attempt is received by a computer using an average IP address directly "on" the IPv4 Internet, called the "survival time." For December 2005, the survival time was 14-74 minutes for all types of Windows® PCs, depending on the type of connectivity. This reflects the time from the moment of having an IPv4 address to the time the system received an attack from somewhere on the Internet. It doesn't mean that the attack was successful to cause an infection. Systems which were already patched, which were not listening directly "on" the Internet or which were firewalled against the attack would not be infected. Broadband networks are specifically targeted by attackers because the hosts are capable of higher volume of UCE or denial of service (DOS) attacks when controlled as a zombie member of a bot network. It is clear then that hosts connected directly to the Internet are constantly under attack. A recent AOL/National Cyber Security Alliance survey found that 81% of home PCs didn't have one of three basic security mechanisms (antivirus, antispyware, and firewall) — 44% did not have a properly configured firewall.[5] Certainly IPv6 hosts must be secure enough to connect and remain connected directly to the IPv6 Internet.

[1] "Gartner Survey Ranks Viruses and Worms as Top IT Security Threats ," Gartner Group, June 15, 2005, http://www.gartner.com/press_releases/asset_129199_11.html. Study covered 133 North American organizations having global operations and revenues exceeding $750M.

[2] "Targeted Trojan Email Attacks," UK National Security Coordination Centre, June 15, 2005, http://www.niscc.gov.uk/niscc/docs/ttea.pdf

[3] "MXLogic Email Threat Wrapup" MXLogic press release, http://www.mxlogic.com/news_events/press_releases/12_13_05_CAN_SPAM.html.

[4] "Data Security Summary July to December 2005" whitepaper, F-Secure, December 2005, http://www.f-secure.com/2005/2

[5] AOL/NCSA Online Safety Study, AOL and National Cyber Security Alliance, http://www.staysafeonline.info/pdf/safety_study_2005.pdf , Dec 2005

Host network security is not just a concern for being directly on the Internet. The top network security threats on a large internal network are not that much different from those on the Internet. In fact, internal hosts may be more exposed to network infection by worms because:

- Internal hosts often do not use a host firewall
- There are delays in the application of client and server patches
- Peer-to-peer applications are used, such as file sharing
- Inbound connections are required by centralized IT management
- The business depends upon older systems that everyone must have access to

Enterprise IT spending funds multiple layers of host defense mechanisms: antivirus, antispam, antiadware, antispyware, secure configuration, patch management, and data backup and recovery. Unfortunately, these approaches are not sufficient to mitigate the 0-day worm attack in a typical large internal network because hosts have open TCP/IP access to each other. One host infection would spread rapidly to all vulnerable hosts on the internal network, as well as spread this infection to network-connected business partners.

Both the internal home and business network see a large mix of devices with varying security capabilities and vulnerabilities. To save cost and leverage the efficiencies of fast connectivity, most business functions and administrative activity have been consolidated on the same internal IP network.[6] Network convergence has succeeded in cost-effective voice, video and data sharing the same IP network. Increasingly intelligent phones, PDAs and device appliances all use the same network. Mobile PCs and devices are increasingly bringing network based infections into the internal network. In 2005 F-Secure counted over 150 infections for mobile devices (not to include laptops), including malicious code for some versions of Symbian phone/PDA operating systems that can cross-infect to Windows® platforms.[7] The perimeter of an internal network is increasingly difficult to define and to "secure." Organizations managing 200,000+ hosts on an internal network are starting to realize that the network perimeter really begins at the host network interface. Solutions such as wireless and wired 802.1X are useful in controlling access to a managed network. But IPv6 hosts must be able to securely communicate using any type of network connection.

Due to the UCE phishing problems, it is interesting to note that the security of an organization's internal identity and data depends in part on the security of all hosts on the network. Thus it is important for the IPv6 community to reach consensus on an IPv6 connectivity model that strongly protects any type of IPv6 host from network attack, while still enabling safe, end-to-end application connectivity. Many of the current IPv6 networks have been deployed with all hosts being openly accessible to each other. However, the first IPv6-specific worm was detected in 2002.[8] There are now several studies which demonstrate that IPv6 addressing by itself is not a sufficient defense against worm propagation.[9,10,11]

[6] The term "network connectivity" can be used both to describe having a layer 2 link connection to a network, such as being "on" the local switched Ethernet network, and also to mean the ability of a host to connect to another host using their respective IP addresses. For most of this paper, the latter definition is assumed.

[7] "Data Security Summary July to December 2005" whitepaper, F-Secure, December 2005, http://www.f-secure.com/2005/2

[8] "Security in IPv6 Networks," Michael Warfield, Internet Security Systems, 2003. http://documents.iss.net/whitepapers/IPv6.pdf

[9] "The Effect of DNS Delays in Worm Propagation in the IPv6 Internet," H. Feng, A. Kamra, V. Misra, and A. D. Keromytis, in Proc. of INFOCOM 2005, March 2005. Also at: http://www1.cs.columbia.edu/~angelos/Papers/2005/dns-worm.pdf

[10] "Fast Worm Propagation in IPv6 Networks," Jing Yang, University of Virginia, 2005, http://www.cs.virginia.edu/~evans/malware/yang.ppt

Consequently, hosts on current IPv6 networks will be increasingly required to protect themselves with host firewalls.

Limitations of Current Solutions

There are many network-based solutions for controlling host-to-host connectivity on an IPv4 network, including:

- VLAN address assignment
- Multiple interface cards with static IP, DNS addresses and host routes
- Client VPN tunnels (which optionally force all traffic to be sent and received inside the tunnel)
- Network firewalls

These solutions work with varying degrees of success for specific scenarios mainly by blocking or limiting network access to the host. The availability of these infrastructure solutions vary depending on the topology, management expertise, organizational priorities and budget of each network. From an architectural perspective, the security of the host and the application must be ensured regardless of which network infrastructure is being used. From a practical perspective, hosts and applications can not be sure that adequate defense against network attack will be provided by the network itself. Therefore they must provide their own defenses against network attack and provide their own solutions for safe connectivity. The requirements for connectivity are defined by the user, applications and services on each host.

Host-based solutions for secure connectivity include the host firewall and higher code quality of both host operating system and application software.[12] The host firewall is recommended by many ISPs, businesses, governments and industry organizations. Microsoft® enabled the Windows® Firewall to be "on" by default for both IPv4 and IPv6 in Windows XP® SP2. They also provided IT administrators with Active Directory® policy for managed firewall behavior when hosts are connected to the internal network. While a host firewall has been effective in defending against attacks, it has greatly complicated end-to-end connectivity.

As might be expected, host firewalls are gradually losing their effectiveness. Host firewalls are forced to permit exceptions to their blocking behavior, called "holes," for inbound connections to a listening application port. The hole can be created manually by the user or, in some cases, programmatically by the application itself. While users are usually given a choice about whether to open a programmatic hole, there are several reasons why it is difficult to make a decision about the real risk inherent with this action. A hole is usually open to receive a connection from any Internet source address, which means attacks are allowed from someone in the neighborhood as well as from the other side of the world. Some firewalls provide the name of the requesting program, protocol and port number to help the user make a decision about opening a port to receive connections. But often such detailed information is difficult for

[11] "Importance-Scanning Worm Using Vulnerable-Host Distribution," Z. Chen and C. Ji, Georgia Institute of Technology, to appear in IEEE GLOBECOM 2005. Also at: http://www1.cs.columbia.edu/~angelos/worm05/worm05-chen.pdf

[12] The discussion here does not address the many options available for application layer communications security because they are largely independent of the network and firewalls and because existing applications are not expected to change their security model when becoming IPv6 compatible. Applications are still expected to secure their own communications, and are required to for end-to-end security, such as S/MIME for email and an SSL connection passing through a TCP proxy. However, new IPsec socket APIs are becoming available. Traffic protected between hosts by an IPsec security association has many strong security properties which may not be provided by the application itself. See draft-ietf-ipsec-properties-02.txt for details.

a user to understand. The user often does not know what the current risks are for this particular port in their current network location. Applications written to use other services and libraries for communications are difficult to associate with the original program. Also, a "trusted" application may become authorized to open any number of ports (and thus holes), and to open these holes at all times in the future. The threat environment can rapidly change, such as when a new worm attacks, or when a new infected or malicious host connects to the local network.

Limiting inbound connections to the "private" local network (e.g. local subnet like 192.168.*.*) might be effective for some IPv4 networks, such as a private home network. But it also does not provide protection for the host that is connected to another network using this same address range, such as a hotel or conference local network. Clearly two neighbors could accidentally infect each other without ever intending to connect their computers to each other. Likewise, limiting the scope of inbound connections to IPv6 link local addresses or specific subnets will not be sufficient. An attack could certainly happen within the 14-minute average "survival time" that a client chat program opened an inbound hole to talk to a friend, or during the one hour required to attend an online business meeting using a multimedia conferencing application. Server application ports like file sharing are certain to be attacked. Instead of authorizing holes that can be used for untrusted attacks, we need to be able to authorize inbound connections from specific source identities for specific reasons (applications) that are manageable by the user, no matter what network connection or address that source is using at the time. Part of this can be achieved by better user interface design for access policy in host firewalls. But, IPv6 is critical to provide the ability to authenticate the source using the Internet Key Exchange (IKE) negotiation protocol and an IPsec protocol.

An IPv6 Model for Safer Host Connectivity

A challenge for IPv6 transition planning is to ensure there is no additional risk to hosts and applications (thus to business processes) when transitioning to IPv6. Network attack threats posed by malicious systems on the IPv6 network are similar to those using an IPv4 network. The IPv6 host address generation scheme is important to help limit the exposure to scanning attacks from attackers not on the local link. Attackers who are on the local link are likely to be able to use neighbor discovery (ND) to target all other local link hosts. Yet it doesn't seem practical to disable it because ND and peer name resolution are necessary for peer-to-peer applications to make connections, as well as for client-server connections in small or unmanaged networks. Internal firewalls help isolate enclaves or other groups of hosts based on network topology from attacks external to that group. Taking this concept several steps further, a firewall might logically isolate one application on one host. The applications must then have a mechanism for safely connecting out of their own host firewall and through the firewall of other hosts.

What will make IPv6 adoption beneficial for applications and yet safe from untrusted attacks is ensuring that hosts have the capability of being selectively accessible by trusted peers. IPv6 IPsec has a critical role in providing this by using IKE to authenticate the source of an inbound connection request and pass details about the specific connection being requested. The IKEv1 negotiation protocol has the capability to authenticate not just with digital certificates, but also with an email name and shared passphrase (preshared authenticate key) or a Kerberos ticket. IKEv2 provides even more flexible methods, including user passwords, tokens and biometric credentials through the integration of the extensible authentication protocol (EAP, RFC3748). The key benefit of using IPv6 IPsec-aware firewall then is that the host is able to both make connections to and receive connections from trusted systems, while being protected against all untrusted systems, and against constant inbound attacks from those untrusted systems. The host can securely communicate even when connected to insecure networks. And it should be less complex to manage than what we have in IPv4 IPsec because every IPv6 host should have this IKE/IPsec and firewall capability. This combination also allows the host firewall to inspect traffic after it has been safely received into the host by IPsec (after IPsec inbound processing). Trust but verify. Inspection of inbound network traffic on the host is still recommended for defense-in-depth against application layer attacks from trusted peers using authorized connections.

When the use of IPsec and firewalls is assumed, IPv6 transition is not just the ability of hosts to use an IPv6 address. IPv6 transition becomes a transition to a new environment of safe and secure connectivity — of hosts into trusted groups, virtual application "networks,” secured business units, etc. Hosts within a company are given network access to each other based on business needs, not based on the fact they all have an internal network IP address. The groups communicate effectively among themselves and share the same address space, layer 2 links and routed IP reachability. Worm infections in one group should not necessarily impact vulnerable hosts in another group. End users should be able to create adhoc, temporary and longer lived peer groups and communities that can exist and operate independent of each other within the same IPv6 link local address space (e.g. collaborating groups on a conference wireless network).

One client could have access to several groups simultaneously, but use a different credential and application when communicating with members of each group. Organization IT professionals should be able to centrally manage network access policies for their managed groups, independent of the adhoc collaborative groups managed by end users. They could completely isolate one group of hosts from all the others during an attack. Alternately, the tighter restriction of communication to within the group could be automated based on intrusion detection alerts. The possibilities are open for many scenarios once the firewall is integrated with IPsec. In fact, IPsec with IPv4 can be used to test out most grouping concepts for IT managed systems. IPv6 is needed for enabling end-to-end IPsec connectivity outside the managed environment.

Step-By-Step Requirements for Connectivity

The user's experience of getting connected to the network using IPv6 should be just as easy and secure (if not more so) as the experience using IPv4. After widely adopting unauthenticated DHCP for dynamic address management, the industry is now moving toward authenticating layer 2 access with 802.1X on wired networks as well as wireless networks. So we have a common architectural process for gaining access to the IP network:

1. Physically connect
2. Authenticate & validate & authorize through layer 2 (PPP, PPPoE, 802.1X, etc)
3. Receive/generate layer 3 IP address and outbound name resolution services configuration
4. Enable discoverability for inbound IP-based connections through a name service (optional)

If enabled, IPv6 neighbor discovery allows address resolution for link local connections. Step 4 is noted as optional because it would be needed for remote (off-link) peers to resolve the host name to current IP address. An IPv6 host may choose to disable the ability to receive inbound connections. Nevertheless, the application on that host must be concerned with several additional steps to traverse the firewalls in the outbound IPv6 connection path:

5. Traverse local host firewall outbound
6. Traverse local network edge firewall outbound (and perhaps other filtering devices)
7. Traverse remote edge firewall inbound (and perhaps other filtering devices)
8. Traverse remote host firewall inbound

The outbound host firewall traversal decision would be managed by the host firewall as it does today. The outbound edge firewall traversal would be possible using a routable IPv6 address. The security policy enforced at this point would have to permit IKE and IPsec traffic at least to enable the end-to-end scenario. Organizations with high security requirements may decide to break end-to-end IP connectivity to more tightly control the flow of information outside of their managed environment. For a home network or small business however, the default policy is likely to permit outbound traffic without restriction. Although there could be any number of firewalls in the path across managed networks, one would hope these would not be boundaries which require authentication. Thus they are condensed into steps 6 and 7. We can for the moment assume there is no network address translation (NAT) in the IPv6 path.

Step 7 is the first serious barrier for access to many types of home and business networks. Solutions today typically use remote access VPN tunnels using IPsec tunnel mode, L2TP/IPsec tunnels, PPTP tunnels and in some cases SSL tunnels. In addition to authentication, the tunnel establishment provides internal address assignment for the client to be able to use an address routable only on the private internal network. When the internal network has globally coordinated and routable IPv6 addresses, is the same type of VPN tunnel still needed? Standard IPv6 IPsec features solve the scenario for non-mobile hosts. For mobile hosts, either mobile IKE (IETF work in progress) or a full mobile IPv6 implementation is needed.

Since host firewalls are likely to be always-on, step 8 is the mandatory barrier to solve first for making a connection to an IPv6 host. If the traversal problem is left up to applications, we can expect they will give up on peer-to-peer connectivity and remain burdened with client-server architectures.

The remainder of this paper discusses how an IKE/IPsec-based solution for firewall traversal already has the necessary pieces defined in the IETF standard Security Architecture for the Internet Protocol, RFC2401, and how it is improved by the recent update, RFC4301. Today, IPv6 IPsec implementations are not expected to have all of the features and integration necessary for the solution. Work is needed in the IPv6 community to reach consensus that IKE and IPsec should be used as a solution for both edge and host firewall traversal. Without consensus, an IPv6 host implementation may lack features needed for one or both scenarios.

The next section addresses the features of the RFCs which a computer can use to authorize inbound file sharing connections across the Internet through its host firewall (Step 8). Then features of the RFCs are used to solve edge firewall traversal in Step 7, while still enabling traversal of the internal host firewall.

IPv6 Scenario: Peer-To-Peer File Sharing Through Host Firewalls

There are a number of security bulletins citing attacks through file sharing networks and cautions about using file sharing in peer-to-peer networks.[13] Yet peer-to-peer file sharing remains an important capability as people need to access their own data on remote machines (such as music files), as the size of files exceeds the practical ability to attach them in email to a distribution list, and as privacy concerns restrict the use of third party servers. Third party server services often force the user into service agreements which do not guarantee the privacy or ownership of the user's content after it is uploaded, or as it passes through their servers.[14]

Host firewalls protect against file sharing attacks because they block the inbound connections to poorly secured file shares. A user should be able to safely authorize access to files by requiring source authentication for the inbound connection to the file sharing application, having the firewall and antivirus inspect, and having the file sharing system control file access based on the source identity and the user's intent. Beyond file sharing, the same source authentication approach can enable other applications to make inbound connections, for example to receive a video chats connection from a neighbor. Using IPv6 IPsec would not defend against an attack from an infected file or application that is received through the secure network connection. Likewise IPv6 IPsec would not address risks of infection from making outbound connections to browse websites or to download email. The same mechanisms for application security used today (code quality, antivirus and traffic inspection) are still needed on the host to defend against attacks within application content.

Consider how Alice could enable the scenario of sharing her Christmas photos she stored on her PC. She could send an email to family and friends containing a URL for the photos, perhaps file://alicepc.alicehomenetworkdnsname.name/christmas05 with a password unique to the group of email recipients. This technique does not use the current peer-to-peer file sharing networks. It uses the host file sharing capability provided by the Common Internet File System (CIFS) which uses the Server Message Block (SMB) protocol.[15,16] This protocol is supported in most Windows® platforms and in non-Windows platforms which provide a CIFS/SMB implementation called Samba. The intention is for a recipient to simply click on the link, get prompted for a userid and password and then get a folder listing of her photos from the Christmas05 directory on Alice's PC. This example trusts relatively insecure email to distribute a password used to access these files on Alice's PC. Alice might decide email was sufficiently secure, just as she might email out conference call codes or other access tokens for online meetings. Alternately, she could verbally communicate the password over the phone. There are also other more secure forms of credentials Alice might use to recognize her friends and family.

With sufficient integration and credential flexibility, this scenario can allow Alice the end-to-end benefits of her new IPv6 service. While applications might integrate with IPsec using newly available IPsec socket options, inbound firewall traversal can be enabled without changing the application.

[13] "Risks of File-sharing Technology," Cyber Security Tip ST05-007, US-CERT, 2005. http://www.us-cert.gov/cas/tips/ST05-007.html

[14] For example, the Electronic Freedom Foundation provides this review about End User License Agreements, http://www.eff.org/wp/eula.php

[15] "CIFS:A Common Internet File System," Paul Leach and Dan Perry, Microsoft Interactive Developer, Nov 1996, http://www.microsoft.com/mind/1196/cifs.asp

[16] "What is SMB," Richard Sharpe, Oct 2002, http://www.samba.org/cifs/docs/what-is-smb.html

Likewise any URL supported by an application on her PC might be able to be used. The point is that her computer can recognize trusted friends and family through the process of authenticating the source of the network connection, and provide carefully authorized access for only a particular protocol and only for these particular files. File sharing application changes may be needed to more easily share using explicit credentials or indicate that IPsec must be used to protect the connection. Implementations of Samba have the option of offering user names in the URL as described by the current IETF draft-crhertel-smb-url-09.txt.

Using IPsec to Authorize Specific Inbound Connections

As mentioned earlier, the IPsec-related RFC documents already contain the pieces needed for this scenario. In the process of creating the URL to share her Christmas05 files, Alice's computer configures itself to receive the future inbound connections with these steps:

1. Create the explicit user credential or leverage other credentials already used for each recipient
2. Create the authorization for this credential to access the proper files
3. Create the IPsec policy configuration for this identity and credential to be recognized by IKE authentication as a responder
4. Create the firewall access policy to accept successful IKE authentication using this identity
5. Create the IPsec policy authorizing only SMB protocol connections using this credential
6. Create the firewall access policy to accept only IPsec protected traffic related to the IKE identity to be permitted inbound to the file sharing open port (a hole for trusted inbound connections)
7. Generate the URL for inbound access that Alice can send out. For an IPsec integrated application, there should be an indication that IPsec is required for the inbound connection (and therefore an IKE negotiation must be initiated).

Alice's friend Bob receives her email. From the "file" part of the URL, the file sharing application on Bob's PC creates an outbound SMB connection to Alice's PC. Bob's host firewall may need to be configured with IPsec policy to negotiate IPsec for this traffic, or the URL may indicate it to the IPsec-integrated application.

IKE and IPsec are chosen as the mechanisms to enable this scenario for many reasons. IPsec and thus a form of IKE is likely to be available because every IPv6 stack requires IPsec support. So no additional protocols and software should be needed for firewall traversal. This is particularly important for small host devices which must minimize the amount of code they are required to support. When implemented and configured properly IKE and IPsec provide strong security for network traffic. IPsec is the same security technology used by corporations and governments to provide authorized access to their internal networks with Virtual Private Network (VPN) tunneling. The peer-to-peer scenario is not a VPN by the technical definition of that term because it does not establish an IP tunnel. Host-to-host IPsec would use IPsec transport mode to establish security associations between the real IPv6 source and destination address of each host.

IKE and IPsec are implemented as a core Internet security protocols. This means they are, by design and by careful implementation, intended to be resistant and hardened against nearly all types of network attacks. The IPsec implementation also is designed to be controlled by administrative policy and be transparent to upper layer protocols and applications. Most types of TCP/IP traffic (including some types of multicast) can be secured by it and therefore safely accepted by the firewall. The security of the inbound connection does not have to be fully dependent on the application to provide adequate security. The management model suggested by this paper involves the user or host administrator defining policy for authorizing inbound connections through the host firewall. Since this scenario requires access over the Internet, the default policy might be to require IPsec ESP encryption in addition to authentication to ensure confidentiality. IPsec also provides the ability to tunnel when necessary to traverse the home or business security gateway for access to the end host. IPsec can provide authentication-only transport mode or tunnel mode security associations. Authentication-only tunnels should a better performance choice for access through a gateway when end-to-end encryption is used.

A version of IKE is suggested for firewall traversal because it is integrated with IPsec and better protects the responding host. IKE requires source authentication first, then destination authentication, and typically performs mutual authentication. SSL/TLS requires the responder to authenticate first. Thus Alice's PC would be open to reconnaissance and attacks which force her computer to authenticate to attackers. IKEv2 in particular provides many authentication methods which can be used by users as well as hosts or the IKE process itself. For defense-in-depth or ease of deployment, it may be important for IKE to use a different credential for negotiating access through the firewall than the application layer uses for authorizing access to data. For example, the host security policy may allow any member of a trusted Kerberos realm have access through the firewall to use a shared printer. Access to the file sharing application might require authentication using a temporary userid and password created for specific users during the file sharing process.

By requiring IPsec and firewall filtering, Alice's PC is protected against network attacks from any untrusted device. Even for authorized connections, the host firewall provides a defense against application layer attacks in the content and prevents access to other ports that Bob's PC might attempt inadvertently or maliciously. Like any well-designed security system, this is a layered, defense-in-depth approach that is manageable and audited.

Table 1-Defense-in-depth layers for access to Alice's PC file data

7th defense	Filesystem/Data Object Permissions based on peer identity
6th defense	Application Authentication and Permissions
5th defense	Host Firewall Packet Inspection – intrusion/attack detection from trusted peer
4th defense	Host Firewall Filters — Access Control to open port (Permit, Block)
3rd defense	Network Layer — IKE/IPsec mutual authentication and optional IPsec encryption
2nd defense	Link Layer — 802.1x or other low level access/admission control to network
	Network Card
1st defense	Wire/wireless signal

Hosts generally have each of these capabilities, though they may not be well integrated. The user interface responsible for sharing files could potentially integrate all of these tasks. The IPv6 Node Requirements draft makes clear the mandate for IPv6 hosts to have IPsec according to the Security Architecture for the Internet Protocol specification, RFC2401 and presumably the updated version of the architecture defined by RFC4301. What this means is that two essential components should be available to secure the file sharing application connections:

- A configurable IPsec policy that specifies actions of discard, bypass or secure for a particular packet or connection

- A security association & key management protocol for negotiable source and destination authentication, and specifying IPsec security for a specific connection

Thus if the file sharing application had the opportunity (see IPsec socket APIs) and chose to integrate with IPsec and the host firewall, it could create a very easy to use scenario for the user.

IPv6 Scenario: Traversing The Edge Firewall

Consider the scenario of connecting to hosts inside Alice's internal business network from anywhere on the IPv6 Internet using her laptop. For IPv4 remote access, VPN products are typically required. However, this scenario is supported natively using IPv6 hosts that support the latest revision to the IPsec architecture, RFC4301 using IKEv2. IPv6 hosts which support the older RFC2401 and IKEv1 may also support the scenario. For simplicity, the internal host and external host are not considered to be mobile.

Using the Peer Authorization Database

The revised IPsec architecture RFC4301 added the feature of a Peer Authorization Database (PAD) that can be used by both initiators and responders to control IKEv2 negotiation of identity. IPsec systems built using the original architecture RFC2401 and IKEv1 may support something similar, since the PAD identity types are the same as for the Security Policy Database (SPD) in RFC2401. The PAD is expected to contain a list of the peers with whom either an initiator or responder will negotiate. In the case of Alice's laptop client, a PAD entry defines computers with DNS names ending in **`corp.v6security.com`** as behind the gateway.v6security.com.

```
Alice Laptop PAD Entry -------------------------
Peer ID:           Name=DNS:*.corp.v6security.com
Auth Proto:        IKEv1, IKEv2
Peer Auth:         certificate=DC=corp, DC=v6security, DC=com; CA root = root DN
My Auth:           certificate=SubjectAltName=
                   RFC822Name=Alice_email@v6security.com,
                   smartcard, chains to CA root = rootDN
Gateway:           DNS:gateway.v6security.com
Child SA Constraint:    use IKE ID as symbolic name for SPD
```

The configuration of the PAD entry tells Alice's computer which IPSec security gateway to contact for access to hosts with the DNS name extension, `corp.v6security.com`. Either IKEv1 or IKEv2 can be used to negotiate IPsec for `corp.v6security.com` computers. The authentication in this case is based on certificates which have been enrolled under a common trusted root CA, identified by an X.509 Distinguished Name (DN). The certificate expected from the peer computers using `corp.v6security.com` DNS names must be a certificate from any issuer which contains a subtree of the DN for Alice's company, also which chains to the particular root CA name.

While not mentioned in RFC4301 explicitly, the My Auth attribute above assists in local credential selection when Alice's laptop authenticates to internal hosts. As an IKE initiator, she may not receive a hint from the IKE responder about which credential is most appropriate to use. For example, internal hosts may authorize different access if she authenticates as a user vs. authenticating as her laptop or other identity. So in this example, IKE will use the certificate specifying her email user name. This might also avoid the requirement of having to deploy a machine certificates in addition to a user certificate. The My Auth attribute also specifies that the certificate must be a smartcard certificate. With a smartcard credential, she can be less worried about a stolen laptop having access through her security gateway and to her internal systems. An IKEv2 initiator can optionally send an ID payload containing the identity for a responder which it seeks to contact, IDr. This helps determine which of potentially many local identities should be used to authenticate the responder. In the absence of this field, or with IKEv1, it is helpful to know a specific credential to select as both an initiator and responder. So this scenario works better with IKEv2 functionality, but could be supported with IKEv1.

The Child SA constraint setting indicates that IKEv2 should use the identity provided in the IKE ID payload field to find the corresponding entry in the Security Policy Database (SPD).

Using the Security Policy Database (SPD)

The SPD entry controls what types of IPsec security associations are allowed to be negotiated by IKE Phase 2 negotiation. In this case, her laptop IKE will use `Alice_email@v6security.com` as the initiator IKE IDi, and expect back an FQDN of an internal computer, IDr being something like `Alicepc.corp.v6security.com`. To find SPD policy for this IDr, it will need to match the following SPD Name entry:

```
Alice laptop SPD entry for internal machines ------------------
Name:                   DNS=*.corp.v6security.com
Selector Set:
 Local Address Range:   ANY
 Remote Address Range:  <ISP-Global-IPv6-prefix /48>::0 to
                        <ISP-Global-IPv6-prefix /48>:FFFF:FFFF:FFFF:FFFF:FFFF
 Next Layer Protocol:   ANY
                        IPsec SA PopulateFromPacket for all fields
Action:                 Protect=ESP transport mode, 3DES/SHA1
```

In this configuration, the /48 IPv6 network prefix is configured to be sure hosts with this name use addresses within the specified range. It also shows that internal hosts are using globally routed IPv6 Internet addresses, even though they might not be directly accessible from outside the security gateway, or resolvable via Internet DNS AAAA records. The only way to access these hosts would be through the `gateway.v6secuirty.com` device indicated by the PAD entry of the previous section.

How this SPD entry is managed requires some attention. The ISP-Global-IPv6-prefix may be changed by the ISP while Alice is remote. If the ISP did change the prefix, this configuration would still enable IKE and IPsec to recognize the proper gateway for `alicepc.corp.v6security.com` based on the wildcard used in the Name field of the PAD and SPD entries. Thus the local and remote address range might not be a statically configured value in the laptop SPD entry. The risk of not having this range specified is that it would not be available to check whether an attacker spoofed the DNS name resolution to return his own address for AlicePC. A safeguard against Alice successfully connecting to the attacker is that her PAD and SPD entries require IPsec to all host names ending with `corp.v6security.com`. Alice's laptop would attempt to negotiate IKE to the attacker address, fail to authenticate because mutual authentication is required in IKE. Thus she would fail to communicate to his IP address with IPsec. She may have been tricked into revealing her source identity in IKE to the attacker through this active attack. But she is protected from establishing an application connection to the attacker.

In this particular case, the laptop policy configuration for IPsec does not limit which protocols and ports Alice can use when connecting to her internal computers. However, the entry requires that IKE propose a specific IPsec SA selector based on the protocols and port information in the outbound packet. This makes the IPsec SA connection-specific, and also allows the greatest chance of success with a more narrow SPD inbound traffic selector policy on the responder (see selector discussion under IPsec Interoperability). The IPsec SPD traffic selectors and/or host firewalls on internal machines will have policies that restrict inbound access to specific ports. Lastly, the SPD action for this address range specifies that IPsec must encrypt with 3DES to any address in that range using ESP transport mode with SHA1 for authentication, regardless of what source address is used.

Tunneling Through The Edge Firewall

The configuration above showed the essentials for how IPsec policy controls IKEv1 or IKEv2 authentication, as well as how it uses IPsec protocols to protect traffic host-to-host. However, in order to reach the internal machines from the Internet, Alice's laptop must first traverse the edge firewall.

If this were a home gateway, Microsoft® recommends certain IPv6 home gateway functionality, which includes stateful filtering firewall functionality to protect legacy and simple hosts. Stateful filtering also blocks inbound connections. For an IPv6 unmanaged gateway, the gateway could allow only IKE and IPsec for inbound connections by default. Or the gateway might require the laptop to authenticate with a unique preshared key and use a tunnel to gain access through the gateway. In either case, likely the laptop's PAD entry would be the main control for whether a tunnel to the gateway was required to access the internal hosts. IKEv2 selector negotiation is flexible enough for the gateway to notify the laptop whether IPsec transport mode was allowed through the tunnel for accessing internal hosts. While it is feasible within the RFC IPsec design, the support of IPsec transport mode through an IPsec tunnel is not required. Thus consensus on this model would help ensure host IPv6 products where capable.

For managed security boundaries like corporate firewalls, by RFC2401 and RFC4301, Alice should be able to use IKEv1 or IKEv2 to negotiate an IPv6 IPsec tunnel to the firewall to access the systems behind it. This is very similar to the remote access VPN behavior used for IPv4, only much simpler because IPv6 supports access natively. The organization would not need to purchase and manage VPN clients for every remote access client. They would still need to manage identity and a minimal IPsec PAD and SPD policy configuration for these clients to know when and how to use IKE to negotiate with the gateway. Since the PAD already identified the gateway for Alice, she might need only one additional SPD entry to describe how to secure the IPsec tunnel to the managed firewall, like this:

```
Alice laptop SPD entry for V6security gateway ------------------
Name:             DNS=gateway.v6security.com
Selector Set:
 Local Address Range:   ANY
 Remote Address Range:  <ISP-Global-IPv6-prefix /48>::0 to
                        <ISP-Global-IPv6-prefix /48>:FFFF:FFFF:FFFF:FFFF:FFFF
 Next Layer Protocol:   ANY
                        IPsec security association=PopulateFromPacket
Action:                 Protect=ESP tunnel mode, null-encryption/SHA1
Local Tunnel Address:   <Alice's laptop current globally routable IPv6 address>
Remote Tunnel Address:  <ISP-Global-IPv6-prefix /48>::<gateway 64bit host ID> or
                        Gateway.v6security.com
```

The gateway's global address does resolve in Internet DNS, and is used an IKE ID with this FQDN for proper selection of this SPD entry using the symbolic name. An IPsec ESP null-encryption, authentication-only tunnel SA pair is one possible option for a light-weight tunnel with the gateway.

It is possible that the internal network uses only link local addressing. In order to resolve names on the internal remote network behind the firewall, Alice's laptop IPv6 IPsec implementation will need to support sending link-local multicast traffic through the IPsec tunnel. To do this, the virtual tunnel interface on her laptop requires an internal link local source address. In some cases, larger organizations may have more than one subnet behind the gateway and be using IPv6 unique local addresses (ULA, RFC4193). In either case, if IKEv1 is used, then the local SPD configuration needs to indicate what types of addresses are used for internal addresses by the laptop. IKEv2 has the capability of requesting and receiving an internal IP address configuration through a CP(CFG_REQUEST) and CP(CFG_REPLY) payload, which can contain internal DNS server IP address as well.

However, normal IPv6 address assignment mechanisms could be used once the tunnel interface is established. IPv6 router solicitation/advertisement followed by DHCPv6 through the tunnel could configure the internal address (similar to RFC 2436, DHCPv4 Configuration of IPsec Tunnel Mode). This has the advantage of not complicating IKEv2 with configuration option support, and not changing IKEv2 with new configuration options. Once an appropriate internal address configuration is obtained for the virtual tunnel interface, then name resolution can proceed in the same manner that the IPv6 node is normally capable of resolving names to addresses. Once the internal names are resolved to internal addresses, an SPD entry controls negotiation with internal host addresses in the /48 prefix range.

Note that IKE and IPsec authentication & encryption end-to-end is still necessary, even though there could be an IPsec authenticated and encrypted tunnel to the gateway. The main reason is because the internal hosts are assumed to be protected by their host firewalls. The secondary reason may be because the local network does not provide security for traffic to and from the internal hosts, such as an open wireless network. Lastly, the negotiation with the gateway must be decided prior to the negotiation of security properties for IPsec transport mode between hosts.

Scenario Requirements for IPv6 Transition

Today the IPv6 IPsec and firewall implementations are capable of enabling the file sharing and gateway scenarios today with much manual configuration. With a relatively small amount of work on the file sharing application, the user experience can be almost seamless. The key issue is ensuring that the host IPv6 IPsec implementation supports firewall traversal. There are two significant efforts underway to improve the IPsec capabilities of IPv6 hosts, interoperability testing and the provision of APIs which applications can use to secure their connections. Neither are intended to provide IPsec-based firewall traversal at this point. However, they are necessary steps for making IPsec more easy to use and thus an available, viable approach for firewall traversal for IPv6.

IPsec Interoperability

While there are IPsec interoperability testing efforts by both ICSA Labs and the Virtual Private Network Consortium (VPNC), these are not focused on host-to-host scenarios for IPv6. Most interoperability tests focus on various forms of IPsec tunnel mode for IPv4 VPNs from host to gateway, and from gateway-to-gateway.

The current University of New Hampshire and IPv6Ready Logo definitions for IPv6 interoperability do not specify requirements for IPsec. However, the specification for how IPv6 IPsec interoperates is being prepared for Phase 2 of the IPv6Ready logo program in collaboration with the University of NH IPv6 testing program, see http://www.ipv6ready.org/pdf/IPsec_1_7_5.pdf

It is important that customers review the test scenarios and provide feedback to the program about whether they are sufficient to cover the customer's real world IPv6 host security requirements. For example, a host could provide two application services which require IPsec for inbound traffic. The high-level security policy might be expressed as:

- Allow file sharing (using SMB TCP port 445), for specific users only, authenticate by email address and password
- Allow printer sharing (using Internet Printing Protocol TCP port 631) for any computer or user in a trusted Kerberos realm

This high level policy might be interpreted into the following IPsec security policy database (SPD) selectors on the responding host:

Table 2 — Responding host IPsec SPD Selectors for two IPsec-secured TCP applications

Source Address	Destination Address	Protocol	Source Port	Destination Port
Any	Any	TCP	Any	445
Any	Any	TCP	Any	631

This type of high-level firewall access policy can be problematic for an IPsec implementation to support in an interoperable way. By IETF design, neither the IKEv1 responder nor the IKEv2 responder knows which port the initiator wants to secure with IPsec in phase 1 of the IKE negotiation. IKE Phase 1 is where the authentication method is chosen. The initiator could propose the ability to authenticate with both an email address/password and with Kerberos. If the responder chose Kerberos, then access to the file share would be prevented. If the responder chose email address/password as the authentication method, then access to the shared printer would be prevented. The typical solution is to have the IPsec policy SPD configuration of the initiator exactly match the responder policy. But forcing a client on the network to have the exact policy configuration of all potential servers is impractical. So a consensus set of guidelines for negotiating different host-to-host scenarios needs to be defined to achieve better interoperability.

IPsec Application Programming Interfaces (APIs)

There are three types of APIs which may be supported for use with IPsec on a host. The first type of API is that used by key management protocols to control IPsec. RFC2367 PF_KEY Key Management API, Version 2 defines a raw socket management interface that can used by IKE and other keying modules to control IPsec SPD and SADBs in kernel. This API can also be used to manually specify the parameters of an IPsec security association and SPD entry using a command line utility. Thus the PF_KEY API is not suitable for a file sharing application to define IPsec policy to secure traffic, unless the file sharing application created the key material and managed the IPsec SAs directly, or also implemented an automated key management protocol.

The second type of API is a management API that allows an IPsec policy service to interpret higher level administrator-defined policy into the detailed configuration for IKE negotiation and the corresponding SPD entries for IPsec processing of IP packets. This type of API is vendor defined for each product and not standardized by the IETF. However, the IETF IPsec Policy Working Group completed RFC3585, the IPsec Configuration Policy Information Model. Since the model represents the full configuration of all RFC-defined IPsec features, only parts of it may be supported by host implementations corresponding to the IPsec features supported by that host. Host implementations typically use command line or graphic configuration tools to generate IPsec policies in a variety of formats, such as an LDAP schema, XML, or structured ASCII text files. Thus the policy creation can usually be programmed using a scripting method. Most hosts assume the scripted policy to be applied by an administrator to cover all traffic, not necessarily by and for just one application's traffic.

The third type of API is a method of programming IKE and IPsec policy by an application for it's own traffic, typically through socket options. The RFC3542 Advanced Socket API for IPv6 specifically omitted options for controlling IPsec protection of a socket's traffic. The basic IPsec socket options were described for BSD Unix sockets in a 1997 draft by Dan McDonald that has long since expired. However, the WIDE/KAME project had been continuing this work and produced stable IPsec socket options for IPv6 IPsec using IKEv1 for BSD Unix variants (FreeBSD, NetBSD, OpenBSD and BSDi). The KAME work on IPsec, including a stable IKEv2 for IPv6 will be continued by a separate WIDE IPsec working group. These links provide details on the IPsec socket API support in popular operating systems:

- **IPsec socket API expired draft — draft-mcdonald-simple-ipsec-api-01.txt**
- **FreeBSD IPsec Socket Options**
- **Max OS® X IPsec Socket Options**
- **Solaris™ 9 IPsec Socket Options**
- **Windows® IPsec Socket Options** — Windows 2000, Windows XP and Windows Server 2003 support policy-based IPsec configurations for IPv4, but only manual IPsec SAs for IPv6 testing. No socket options are supported for IPv4 or IPv6. Windows Vista™ should have IPv6 IPsec socket options documented in the current December 2005 Customer Technical Preview (CTP) Platform SDK, available for those participating in the Windows Vista™ CTP program.

Programmatic use of IPsec to secure connections will enable the host firewall to better identify and authorize inbound connections. But the IPsec socket API specification needs to be standardized, along with interoperability tests for scenarios such as the example above.

The IETF has been using IPsec to secure new protocols and new versions of older protocols. RFC3723 Securing Block Storage Protocols over IP requires IPsec integration with iSCSI. Work is also in progress for IPsec integration with NFSv4 in draft-ietf-nfsv4-channel-bindings-03.txt.[17] IPsec is also being incorporated as part of the Generic Security System API channel bindings in the IETF Kitten working group document draft-ietf-kitten-gssapi-channel-bindings-01.txt.

IPv6 Transition Plan For Host Security

With a mean time to attack of just 14 minutes today, our devices and hosts have to be protected before they connect to the network. We can't expect to get our systems connected and then add security later. Hosts will need to make secure connections using whatever type of IPv6 network connection is available.

It is sometimes thought that IPsec for IPv6 requires a global public key infrastructure (PKI). However, the IPsec key management protocols IKEv1, IKEv2 and KINK all have non-PKI authentication methods which are deployable, including IKEv2's support of the Extensible Authentication Protocol (EAP) that enables use of passwords and tokens like RSA's SecureID™. PKI-based authentication does not necessarily mean that the infrastructure is public. IKEv1 and IKEv2 for example can use digital certificates issued under a root that an organization manages itself, and can use self-signed certificates if there is a policy which allows these to be trusted.

The specific scenarios for IPsec connectivity should be identified as part of the IPv6 transition plan. The IPv6 transition plan should identify the strategy for IPv6 hosts handling Steps 2-8 above. To be more specific, the plan should:

1. Identify the connectivity environments, connection methods and application communication requirements for IPv6 applications and hosts
2. Identify the potential security risks of untrusted network attacks on trusted IPv6 applications and hosts in each of these connectivity environments
3. Identify solution alternatives for mitigating the impact of untrusted and trusted inbound network attacks on IPv6 applications and hosts.
    a. Consider which security boundaries (Steps 1-8) may be present in each connectivity environment
    b. Consider how each security boundary can be safely managed and traversed.
        i. Consider how adhoc, temporary and long lived communication groups can be formed between applications, users and/or hosts using trust.
        ii. Consider how logical group boundaries can by designed to safely allow business as usual inside of the group, keeping its members safely isolated from infections in other groups.
        iii. Consider how network access policies for the host can keep it isolated; keep it safe from infections propagating from other trusted members of a group.

[17] See also this precursor presentation by Nicolas Williams during the Security Area Advisory Group meeting in 2003.

c. Consider and document your specific IPv6 host security requirements, to include:

   i. Requirements and objectives or policies for each connectivity environment. How hosts will be secure prior to connection, during the connection process, and during operation.

   ii. How the source of connections can be authenticated (what identities are used, which protocols)

   iii. How inbound connections are authorized with that authenticated identity

   iv. What traffic protection is required or is specifically disallowed

   v. How IPv6 host software currently can or can not fulfill these requirements.

4. If IPv6 IPsec capabilities are needed, identify the type of high-level host access policies needed for each host and gateway and how those policies need to be managed.

   a. Is there an IPsec policy defined per application, per IP address, per prefix range, per subnet, per network, etc? Is the IPsec policy different from the firewall policy?

   b. Is there an IPsec policy distributed to the host when the host connects to the network? Or is the host given a policy that it uses all the time, no matter which network it is on?

   c. What other vendor products would need to be able to use the same or a compatible IPsec policy? How should interoperability be defined?

With these requirements and capabilities documented, current IPsec implementations can be evaluated and any missing features identified to the IPv6 implementer.

Testing The Concepts Today

The scenario where a managed host provides inbound access only to trusted peers over IPsec is fully deployable today using IPsec for IPv4 in a solution Microsoft® calls "Server and Domain Isolation." IPsec policy acts as the firewall policy that requires inbound authentication. IPsec policy is easily deployed to all computers which are members of an Active Directory domain. The IPsec policy requires the use of IKE and IPsec when machines are communicate using IPv4 addresses and subnets defined in the policy. Trust based on IKE authentication can use a preshared secret, Kerberos or digital certificates. Authorization for inbound connections is provided either implicitly through the ability to authenticate, or explicitly using Group Policy security options, or by configuration of the Windows Firewall for authenticated bypass using IPsec. Full details are available at http://www.microsoft.com/ipsec.

The scenario where a client connects to a file share over IPsec specifically through two host firewalls was successfully tested using two Windows XP SP2 computers using IPsec for IPv4. Windows XP does not support full IKE and IPsec policy capabilities for IPv6. The scenario works for IPv4 IPsec, not only using the Windows Firewall, but also with third party host firewall products. However, the difficult manual configuration limits adoption to scenarios that can be professionally managed.

Updated information and full details to test these scenarios are available at http://www.v6security.com.

Summary

The extreme risks present on both the Internet and on internal networks have resulted in the always-on host firewall. IPsec and IKE when integrated with host firewalls provide a layered, defense-in-depth strategy for securely enabling inbound connectivity to applications. The IPv6 IPsec IETF standardized capabilities, with either RFC2401/IKEv1 or RFC4301/IKEv2, provide sufficient functionality to negotiate host firewall access policies. The same IPv6 IPsec mechanisms provide reasonably safe traversal of edge-based firewalls, such as home gateways and corporate firewalls. While these capabilities are possible by IETF design, some work is required both on the customer side and on the vendor side to use these mechanisms. Customers and transition planners need to identify requirements and scenarios as part of an IPv6 transition plan. IPv6 vendors need to support IKE and IPsec features and integration that make end-to-end connectivity safe, yet easy to use and manage.

The question for the IPv6 community is whether it can reach consensus on solutions that can be widely adopted by devices and hosts alike to solve these communication barriers. If IPv6 does not provide a consensus solution to these problems, then a number of perhaps competing and non-interoperable solutions will be attempted that will greatly complicate if not break the end-to-end benefit of IPv6.

Acknowledgements

The author is grateful to the reviewers of this paper who provided valuable input and comments: Merike Kaeo of Doubleshot Security and several others who wish to remain unnamed. Thanks also to Michael Ambrose and the USIPv6 staff for their help and motivation to get the first version of this paper written for the 6sense IPv6 newsletter.

About V6 Security

V6 Security was established in 2004 by former Microsoft program manager for Windows network security and IPsec development, William Dixon. V6 Security provides experienced enterprise IT security analysis, security risk assessment, planning for secure IPv6 transition, and state-of-the-art IPsec architecture, training, and deployment assistance. More information is available at http://www.v6security.com.