North American IPv6 Task Force (NAv6TF) Technology Report

"IPv6 Security Technology Paper"

Version 1.0

July 22, 2006

Primary Author/Editor: Merike Kaeo

Contributing Authors: David Green,

Jim Bound, Yanick Pouffary

merike@doubleshotsecurity.com

1. SCOPE 4

2. INTRODUCTION 4

3. INFORMATION SECURITY FUNDAMENTALS 4

3.1. SECURITY PROPERTIES 5
3.2. SECURITY SERVICES 6

4. COMPARING IPV4 AND IPV6 SECURITY 8

5. (RE)INTRODUCING THE END-TO-END SECURITY MODEL 9

5.1. HYBRID END-TO-END AND NETWORK CENTRIC SECURITY 10
5.1.1. Distributed Firewalls 10
5.1.2. How IPsec Will Affect Distributed Firewall Architectures 12
5.2. EVOLVING TO CREATE A FLEXIBLE SECURITY ARCHITECTURE 14

6. FUNDAMENTALS OF IPSEC 14

6.1. IPSEC PROTOCOLS FOR AUTHENTICATION, INTEGRITY AND CONFIDENTIALITY 15
6.2. SECURITY ASSOCIATIONS AND ASSOCIATED DATABASES 17
6.3. MANAGING SECURITY ASSOCIATIONS AND CRYPTOGRAPHIC KEYS 21
6.4. API CONSIDERATIONS 21

7. ADDRESSING SECURITY CONSIDERATIONS 22

7.1. MANUALLY CONFIGURED ADDRESSES 23
7.2. STATELESS AUTOCONFIGURATION 24
7.2.1. Secure Neighbor Discovery (SeND) 27
7.2.2. Using IPsec to Secure Neighbor Discovery 28
7.3. STATEFUL AUTOCONFIGURATION AND DHCP CONSIDERATIONS 28
7.4. FURTHER DHCP CONSIDERATIONS 29
7.5. PRIVACY ADDRESSES 30
7.6. DNS CONSIDERATIONS 30

8. TRANSITION MECHANISM SECURITY CONSIDERATIONS 31

8.1. MANUALLY CONFIGURED TUNNELS 32
8.2. AUTOMATIC TUNNELS 33
8.2.1. 6to4 33
8.2.2. ISATAP 35
8.2.3. Teredo 36
8.3. TUNNEL BROKERS 37
8.4. NEW TUNNELING STANDARDS 37

9. MOBILITY SECURITY CONSIDERATIONS 37

9.1. BASIC MOBILE IPV6 OPERATIONS 38
9.2. MOBILE IPV6 SECURITY USING IPSEC 41

9.2.1. Using IPsec to Protect MN and HA Communications *41*
9.2.2. MIPv6 with IKEv2 and Revised IPsec Architecture *42*
9.2.3. MOBIKE *43*
9.2.4. Using IPsec To Protect MN and CN Communications *44*
9.3. ADDITIONAL MOBILE IPV6 SECURITY MECHANISM 44
9.3.1. Alternative Authentication Protocol *44*
9.3.2. Securing Mobile IPv6 Route Optimization *45*
9.4. MOBILE IPV6 SECURITY ARCHITECTURES 45

10. IPV6 MANAGEMENT / SECURITY AUDITING TOOLS 47

10.1. MANAGEMENT TOOLS 47
10.2. SECURITY AUDITING AND NETWORK ASSESSMENT TOOLS 47

11. IPV6 SECURITY DEPLOYMENT BEST PRACTICE GUIDELINES 48

11.1. SECURITY POLICY CONSIDERATIONS 49
11.2. END-HOST SECURITY 50
11.3. NETWORK INFRASTRUCTURE SECURITY 51
11.3.1. Infrastructure Device Security *51*
11.3.2. Routing Control Plane Security *52*
11.3.3. Firewalls / Filtering *53*
11.3.4. Logging / Auditing *55*
11.3.5. IPv6 Security Deployment Summary *55*

12. FUTURE CONSIDERATIONS 55

12.1. MODELS FOR MORE AUTOMATED END-TO-END SECURITY 55
12.2. PKI REQUIREMENT AND ANALYSIS 56

13. A BASIC FRAMEWORK FOR IPV6 SECURITY 56

14. ACKNOWLEDGEMENTS 59

15. NAV6TF DISCLAIMER 59

16. ABOUT NAV6TF 59

17. ABOUT THE AUTHOR 60

18. APPENDIX A – IPV6 CAPABLE NETWORK ASSESSMENT TOOLS 60

19. REFERENCES 77

1. Scope

This security technology paper focuses on the fundamental security deployment issues for IPv6-enabled networks. It is not meant to define a definitive security policy for any particular environment but rather it is an attempt to enumerate all of the considerations to be accounted for when creating an appropriate security policy and architecting the IPv6 network to incorporate appropriate security measures. The technical merits and tradeoffs are discussed to add a practical component based on recent deployment experiences. It is assumed that the reader is familiar with basic IPv6 operation and has a fundamental understanding of network security issues.

2. Introduction

As IPv6 networks migrate from lab environments into dependable production systems, we are presented with both the challenge of adapting our Information Assurance (IA) architecture to a new protocol and the opportunity to leverage new features to enhance network security. Native IPv6 networks will coexist with environments where IPv6 capabilities are introduced into production networks with existing IPv4-based infrastructures. While security of our current production networks must be evolved for IPv6, there are features in IPv6 and new trends in networking that should lead us to changing security paradigms. End-to-end security between hosts has had limited practicality in IPv4-based networks but is a key feature of IPv6. A return to the end-to-end network model should be architected into any dual stacked transition architecture with careful consideration for not compromising IPv4 security.

The controversy of whether host based security is better than network based security should be resolved with the understanding that a layered security approach is necessary. A combination of application, host and network-based security is required to securely conduct business on the network of networks which make up the Internet.

This white paper will enumerate the security advantages which are relevant in today's IPv6 networks and will detail the deployment considerations to effectively design and architect secure IPv6 networks.

3. Information Security Fundamentals

What does it mean to provide a secure network? Invariably, the goal is to protect electronic communication from malicious individuals and applications who are determined to spoof, corrupt, alter or destroy the data or render critical services unavailable. Protection is required by every device that is participating in networked communication and all information that is either stored on a device or is in transit between communicating devices or is processed by the devices.

Protecting the critical devices that make up these network infrastructures and the business processes which are dependent on the network is a key concern for everyone. Too many people today are

agonizingly familiar with the increasing threats of email spam, phishing scams, worms, viruses and numerous DoS attacks which impact business services and communication needs. Computer network attacks no longer target simply a single machine or even a single network. Today's attack trends are increasingly more automated and sophisticated and can result in large distributed denial-of-service attacks that broadly affect key components of information networks. Even unsuspecting users can cause a risk if unbeknownst to them their infected system begins to spread a worm or virus throughout the corporate network. Alternatively, device mis-configuration or a down-rev with respect to operating system patch levels can also create a new vulnerability that opens the network to external attack. A secure network architecture incorporates mitigation techniques which decreases the risk of both deliberate attacks or unintentional events.

3.1. *Security Properties*

It is critical to today's business needs that all networked devices and be accessible at all times in a reliable and secure environment. The mechanisms to provide the security regulation can take many forms, but essentially all forms pertain to the preservation of confidentiality, integrity, accountability and availability.

- ***Confidentiality*** is the property by which access to information is restricted to those who are privileged to see it. Examples of violations of confidentiality include bypassing access control rules or having the capability to read unauthorized information while it is in transit from sender to the recipient.
- ***Integrity*** can pertain to the data as well as the communicating parties. Data integrity is having trust that the information has not been altered during its transit from source to destination. Host/user integrity is having trust that the sender and / or recipient of the information is who it is supposed to be. Data integrity can be compromised when information has been corrupted, willfully or accidentally, before it is read by its intended recipient. Host/user integrity is compromised when an imposter "spoofs" a sender's identity and supplies incorrect information to a recipient.
- ***Accountability*** is synonymous with non-repudiation. Non-repudiation refers to the property that you cannot deny having done something.
- ***Availability*** is the property that the information or resources are accessible when required within a reasonable period of time.

At the most fundamental level, these are the security properties that must be considered and incorporated into a sound security policy. What information is confidential? Does it need to be kept confidential while that information is accessed via the network? Does it need to be kept confidential while it is stored in a database or file? How is integrity of the data preserved? Only a comprehensive corporate risk assessment will provide the answers required to determine the protection that is warranted to any specific environment. For readers looking to supplement their existing network security policies, one of the best resources for examples and templates can be found at the following url: http://www.sans.org/resources/policies/.

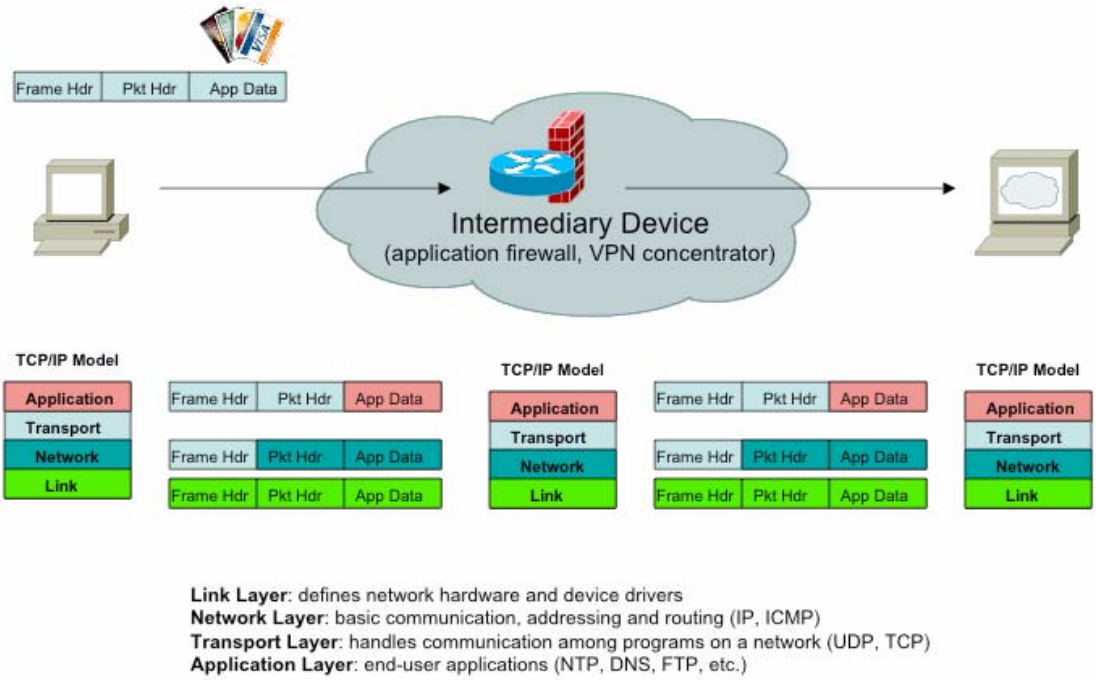
3.2. *Security Services*

How to implement the security properties as defined by a given security policy is a different problem. Usually, there exist a variety of mechanism which need to be considered. The following services are primarily used to implement the properties of confidentiality, integrity, accountability and availability:

- ***Authentication*** is the process of verifying the claimed identity of a device, user and/or application trying to access the resources.
- ***Authorization*** is the rights and permission granted to a user or application that enables them the access to network or computing resources.
- ***Access control*** is the means by which an authorized user has access to resources.
- ***Encryption*** is the mechanism by which information is kept confidential from unauthorized users.
- ***Auditing*** is the process that keeps track of what an authorized or unauthorized user or application is doing.

What makes the problem complex is that these services can be applied at varying levels of the TCP/IP model. Take for example the problem of wanting to provide confidentiality by encrypting a web-based financial transaction as illustrated in figure 1.

Figure 1: TCP/IP Layered Security Example



The encryption can be performed at either the application layer, the network layer or the link layer. Note that encryption can also be performed at the transport layer although for visual simplicity, this case was not shown in the figure. The trade-off as you go up the TCP/IP-layer stack is that you perform the security service, in this case encryption, at a greater granularity for the specific data that requires the specific service. Additionally, the security services can be provided on the end-hosts that are participating in the communication or by intermediary network devices. An effective security architecture will ensure that the security services are applied in an efficient manner to avoid duplication of effort and unnecessary processing cycles.

Security services will always be required at varying layers of the TCP/IP stack due to varying policies and the need to integrate easy deployment with the appropriate granularity to offer the required security protection. When specifically dealing with the network layer, all of the security service considerations required to protect networked communication is independent of whether IPv4 or IPv6 is used for the networking layer transport.

4. Comparing IPv4 and IPv6 Security

Although any security architecture requires a layered approach, let's look at how security concerns compare and contrast in IPv4 and IPv6 environments. As pointed out in the previous section, the fundamental security properties and security services used to protect the network infrastructures and the information traversing these networks are the same in both IPv4 and IPv6 environments.

A comparison of IPv4 and IPv6 threat analysis by Darrin Miller and Sean Convery shows the similarities of potential threats and mitigation techniques in both types of networks.[1] The paper recommends that secure IPv6 deployments should be ensured from the start and not be provided as an add-on as was done with IPv4 deployments.

It is important to recognize security enhancements that have been incorporated into the IPv6 base protocol specification (rfc2460) and the added advantage of re-introducing an end-to-end security model without some of the legacy constraints which exist in today's IPv4 networks.

The designers of the IPv6 protocol took into consideration the known security vulnerabilities affecting IPv4 networks at that time and architected a solution which would mitigate many of the risks of those known vulnerabilities. This included issues of broadcast storms, fragmentation attacks and security services such as device authentication, data integrity and confidentiality.

IPv4 networks are susceptible to varying types of fragmentations attacks. The IPv6 standard provides better fragmentation attack mitigation because it requires that::

- Fragmentation is prohibited by intermediary devices – this has a subtle advantage when it is definitively known between some communicating peers that no fragmented traffic will be used.
- Overlapping fragments are not allowed – this is implied by specifying that only the source can actually create fragmented traffic.
- Devices are required to drop reassembled packets that are less than the 1280 byte minimum MTU

Broadcast amplification was another concern in IPv4 networks. The IPv6 specification removes the concept of dedicated broadcast from the protocol and specifies specific language in RFC2463 to mitigate these types of attacks by specifying the following:

> " ICPMv6 messages should not be generated as a response to a packet with an IPv6 multicast destination address, a link-layer multicast address or a link-layer broadcast address"

The IPv6 standard also mandates that all IPv6 capable devices support IPsec for providing authentication, integrity and confidentiality services at the network layer. Whereas the IPv4

[1] S.Convery, D. Miller IPv6 and IPv4 Threat Comparison and Best Practice Evaluation http://www.cisco.com/secxurity_services/ciag/documents/v6-v4-threats.pdf

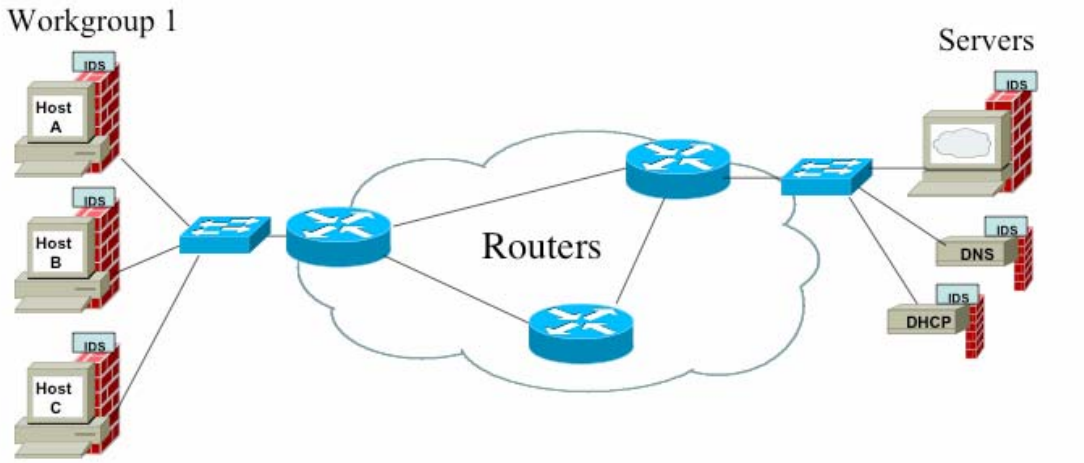
protocol had to retrofit IPsec headers into the original IPv4 frame, IPv6 has the capability to support IPsec within the defined packet structure using extension headers.

As will be pointed out in subsequent sections of this paper, if IPv6 deployments follow the same architectures of IPv4 today, the security models will be much the same with only minor advantages. However, IPv6 security architectures should look to take advantage of the end-to-end security model and make appropriate policy decision modifications where appropriate.

5. (Re)Introducing The End-to-End Security Model

IPv6 network architectures can easily adapt to an end-to-end security model where the end hosts have the responsibility of providing the security services necessary to protect any data traffic between them. This results in greater flexibility for creating policy-based trust domains that are based on varying parameters including node address and application, as shown in figure 2. Each device or end-host can be a member of multiple trust domains, each subject to varying security policies.

Figure 2. End-To-End Security



- Secured communication between Host A and Host B and Application X
- Secured communication between all hosts in Workgroup 1
- Secured communication between routers
- Secured communication between servers
- Secured communication between DNS server & all other non-server devices
- Secured communication between DHCP server & all other non-server devices

When any pair of end devices want to communicate securely, the devices can initiate an authenticated and confidential exchange. Note that these end devices can be end-hosts, servers or routers since the end points in an end-to-end model define the device that is either initiating or receiving the data. Most workstation or server based security implementations augment or enhance local security measures to enforce data integrity, prevent exploitation of the system, and ensure system availability. These hosts can protect themselves from unwanted traffic by providing access control (i.e. firewall) protection on the hosts such that any traffic gets inspected after it gets decrypted and before being forwarded to any upper layer processing. Auditing functions at each host log any potentially malicious activity and provide the means to audit any malicious behavior.

5.1. *Hybrid End-to-End and Network Centric Security*

An end-to-end security model does not mean that there will not be any security services within the network infrastructure. On the contrary, security services should be deployed in both areas to increase the defense in depth. There exist a number of hybrid scenarios which combine end-to-end and network centric security architectures when deploying IPv6. For many transition networks these hybrid solutions can provide a gradual move to native IPv6 networks while still maintaining a secure network which mitigates most of the known vulnerabilities. The tradeoff is often a decision based on performance versus management.

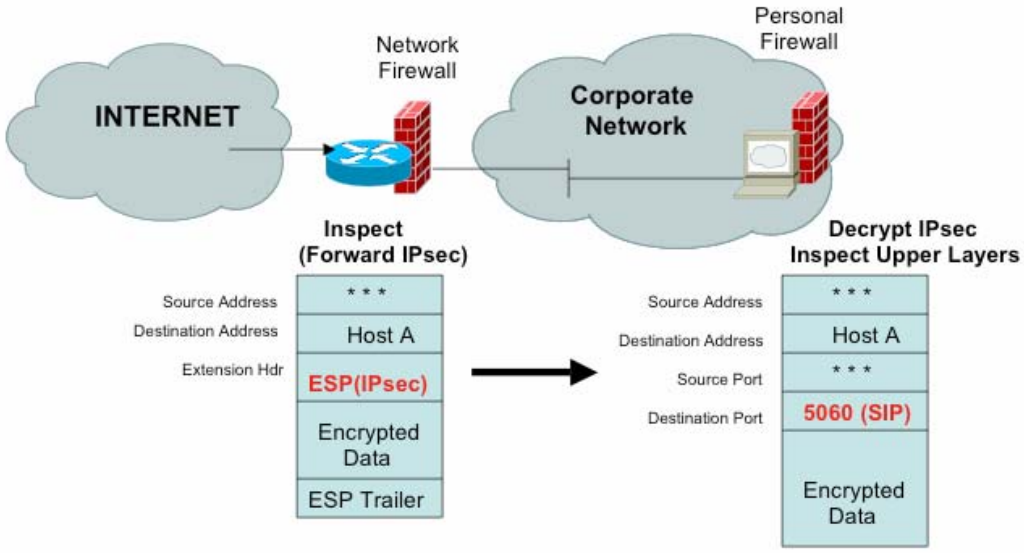
5.1.1. Distributed Firewalls

The most common hybrid security model will incorporate the concept of distributed firewalls[2] . The distributed firewall model consists of managed host-based firewalls in addition to the conventional perimeter firewall model. The addition of managed host-based firewall security adds “defense in depth” to an enterprise’s security architecture and reduces reliance on a single "chokepoint" perimeter security network design. Current firewall systems typically perform all security screening through a common checkpoint. The performance of a single checkpoint approach is increasingly degraded as broadband traffic increases over time, new network protocols are added, and as end-to-end networking and encrypted tunneling become more common. With most netcentric enterprises investing in enhanced IT performance, a network-based firewall model is a definite drawback.

In future security architectures, more coordination will be established between network and host-based firewalls as illustrated in figure 3.

[2] Ref:S. Ioannidis, A. Keromytis, S. Bellovin, and J. Smith. Implementing a Distributed Firewall. In Proceedings of Computer and Communications Security (CCS) 2000, November 2000

Figure 3. Distributed Firewalls



Router packet filters and stand-alone network firewalls will perform a first line screening to ensure that the packet is valid, arrives from a valid source host address and can be sent on to the destination host. At the destination, the host firewall will need to perform a more detailed packet inspection, usually incorporating some intelligent IPsec-aware function, especially if communications to the host are using encryption that prevents detailed screening at perimeter firewalls. In this case, the end-host would first decrypt the incoming packet, perform an inspection on the upper layer protocols, and if successful, send the packet on to the application process. Upon finding a security violation in the packet, a host firewall should reject the packet and report the violation to its security management system.

A distributed firewall can be used to augment a perimeter firewall or reduce the reliance on the perimeter firewall. Host-based firewalls may also be integrated into a single managed system with one or more perimeter firewalls to form a “hybrid distributed firewall” system for a managed defense in depth. A dual perimeter-firewall/distributed firewall system or a hybrid system augments the quality of perimeter defense as the internal firewalls bolster the enterprises ability to distribute, monitor, enforce IA policy and defeat attacks.

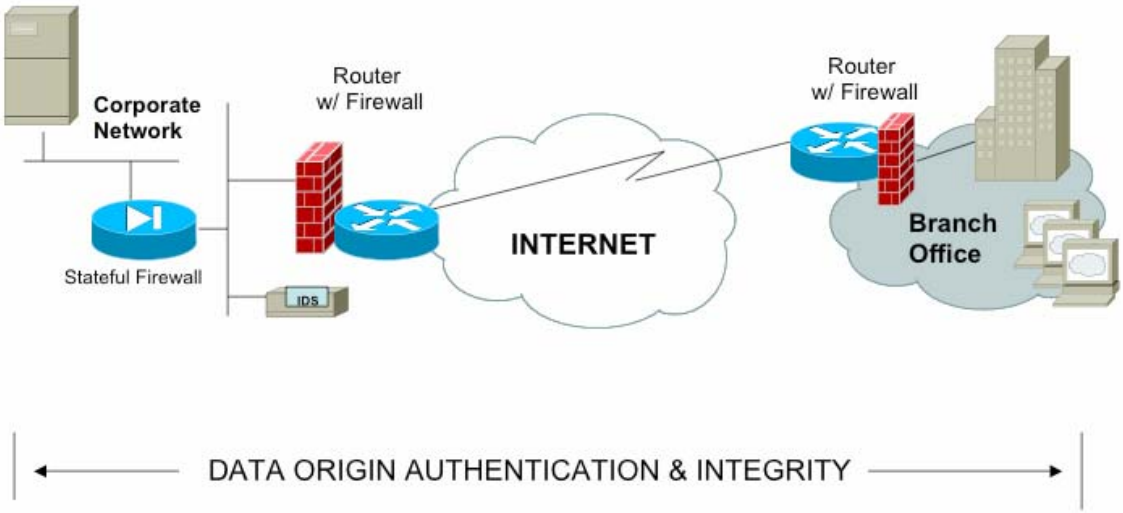
5.1.2. How IPsec Will Affect Distributed Firewall Architectures

IPsec, described in detail in the next section, is often misunderstood to be synonymous with encryption. On the contrary, IPsec does not always require that encryption be implemented or deployed to provide security services. IPsec can be used to provide the following security services:

- Data origin authentication and data integrity
- Data origin authentication, data integrity AND data confidentiality

Some security policies mandate that traffic has to be visible for signature based intrusion detection system observation or deep firewall inspection. Or sometimes the policy simply dictates that traffic has to be observable for some other reason. In those cases, end-to-end IPsec security will only provide authentication and integrity services as shown in figure 4.

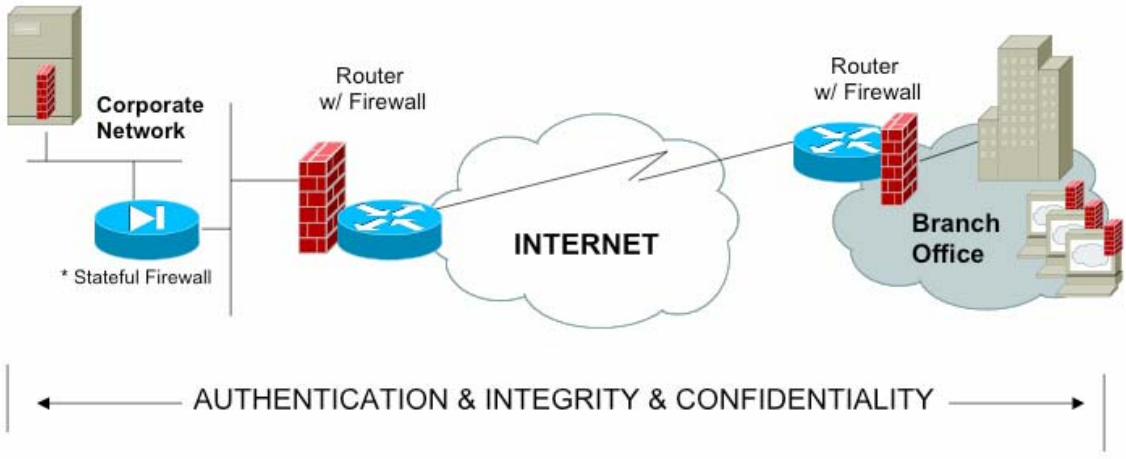
Figure 4. End-to-End IPsec (Authentication & Integrity)



This scenario would ensure IPsec authentication and integrity protection for every data packet from the originating host to the data recipient while still keeping intact the policies which require deep packet inspection and traffic auditing via network IDS systems. If confidentiality is required but cannot impact current IDS and/or firewall filtering policies, then intermediary devices can add IPsec confidentiality protection and encrypt the traffic at allowable intermediary points.

In architectures where there is no encryption policy constraint or the policy is modified to incorporate a true end-to-end security model with confidentiality services allowed between communicating end-hosts, scenarios such as shown in figure 5 can be deployed.

Figure 5. End-To-End IPsec
(Authentication, Integrity & Confidentiality)



* The stateful firewall may still perform deep packet inspection of traffic that may not be subject to confidentiality services

Note that this scenario may still employ some network level packet inspection although it may be limited to simply IP address checking. The deployment of intelligent host-based firewall devices could be used to perform deep packet inspection at the host rather than using a network-based stateful firewall. Or, the two can be used in parallel with the deep packet inspection being performed for any traffic that does not require confidentiality services end-to-end.

A major consideration for future security policies is where to enforce confidentiality. It has often been the case that corporations do not allow for encrypted traffic across specific infrastructures due

to regulatory requirements that must have the capability to have access to the data at any time. However, if that requirement were met in some other manner, such as requiring a corporate-wide key escrow system, then perhaps the current policy of having date traverse the network unencrypted can be modified. These policy decisions will be dictated by whether it is easier for an environment to enforce more granular security at the host versus network infrastructure level. It is the flexibility of having that choice that creates the greatest advantage for future IPv6 networks. The end-to-end model can allow for more intelligent applications to take advantage of the flexible host-based security controls.

5.2. *Evolving To Create A Flexible Security Architecture*

As we get closer to more effectively utilizing an end-to-end security model, we will rely more heavily on distributed security with the communicating hosts providing the policy enforcement for their own communication. This has the advantage of creating specific policies for securing communications based on currently running applications rather than having a central enforcement point try and provide a single group-based policy. With distributed security it is possible to create more dynamic security policies which can vary over time based on changing trust relationships.

Distributed security endpoints consisting of host-resident firewalls, intrusion detection, security patching, and security status monitoring can be accomplished by kernel-mode processes within an operating system, These host-based security checkpoints would be managed by a central system used to distribute and monitor security policies and updates. A managed distributed host-based firewall system utilizing end-to-end IPsec can implement separate multi-level security policies with fine granularity. Using this end-to-end model it is possible to divide users and servers into various trust groups and interest communities to implement separate security rules. Applications and services that are used exclusively in one community may be blocked in other communities, This simplifies the screening rules (and exceptions) at a perimeter firewall and may prevent a breach in one network area from spilling into other network segments. If and when a breach occurs, containment of that breach is more easily managed. An additional benefit is that an, incorrectly implemented security policy in one area (or at the perimeter) does not necessarily compromise the entire system.

6. Fundamentals of IPsec

IPv6 relies heavily on the IPsec standard(s) for security. IPsec is a framework of open standards that provides data confidentiality, data integrity, and data origin authenticity between participating peers. IPsec provides these security services at the IP (i.e. network) layer. IPsec uses the Internet Key Exchange (IKE) protocol to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used. IPsec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. The original IPsec protocol suite of standards is documented in RFC's [RFC2401] through [RFC2412] and [RFC2451]. As of December 2005,

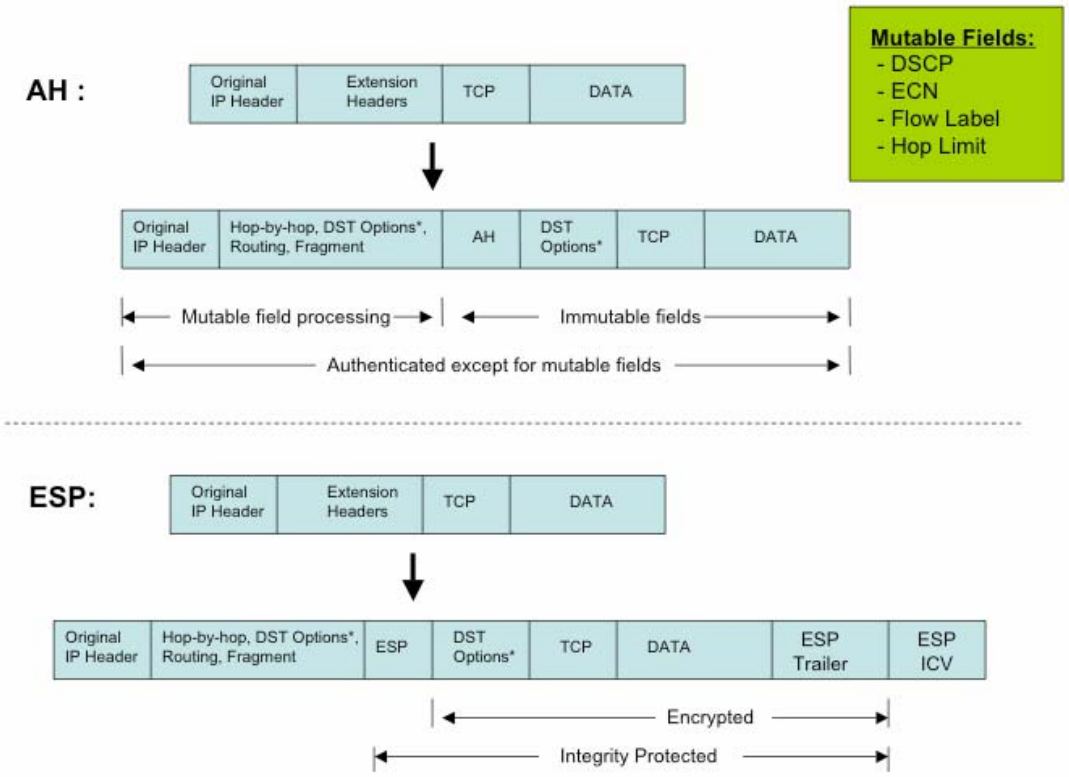
there exist updated versions of the IPsec standard which provide clarifications and enhanced functionality to that of the original specifications.

6.1. *IPsec Protocols For Authentication, Integrity and Confidentiality*

The Authentication Header (AH) protocol, originally defined in RFC2402 and updated in RFC4302, provides data authentication and optional anti-replay services. The Encapsulating Security Payload (ESP) protocol, originally defined in RFC2406 and updated in RFC4303, provides confidentiality, data origin authentication, connectionless integrity, an anti-replay service and traffic flow confidentiality.

The AH and ESP protocols each support two modes of operation: transport mode and tunnel mode. In transport mode, two hosts provide protection primarily for upper-layer protocols. The cryptographic endpoints (where the encryption and decryption take place) are the source and destination of the data packet. In IPv4, a transport mode security protocol header appears immediately after the IP header and before any higher-layer protocols (such as TCP or UDP). In IPv6, the transport mode security protocol header appears after the base IP header and selected extension headers. It may appear before or after destination options but must appear before next layer protocols (e.g., TCP, UDP, SCTP). Figure 6 illustrates the IPsec AH and ESP protection services in transport mode.

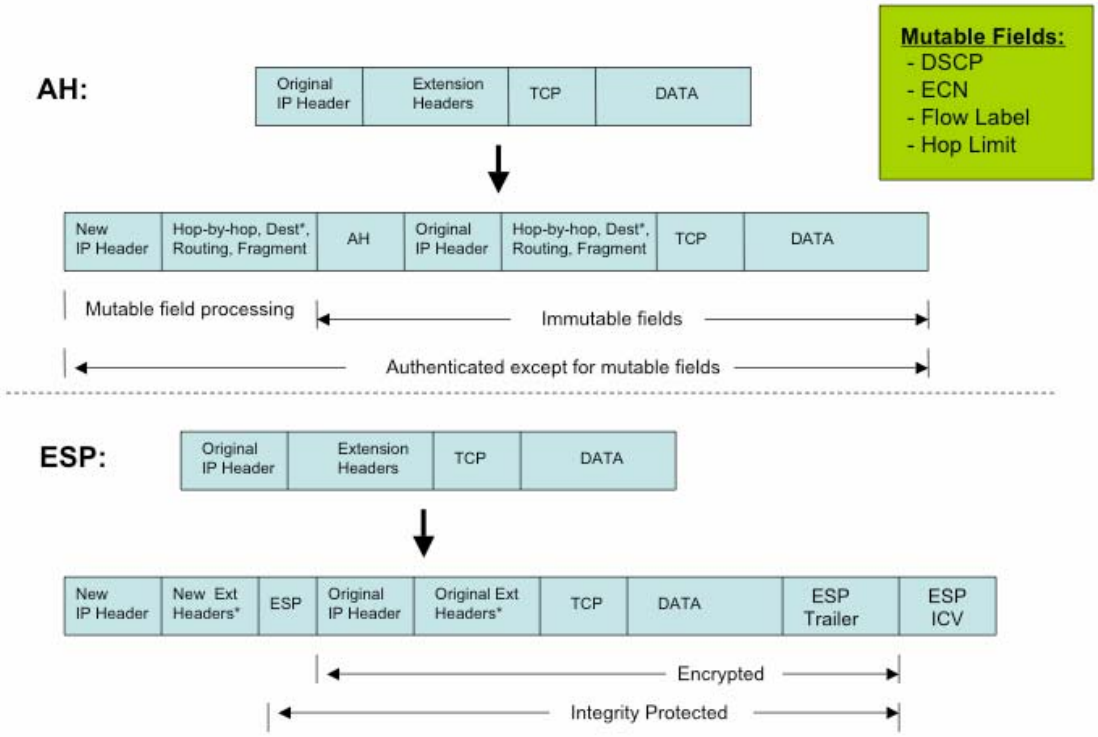
Figure 6. IPv6 IPsec AH/ESP in Transport Mode



In the case of AH in transport mode, security services are provided to selected portions of the IP header preceding the AH header, selected portions of extension headers, and selected options (contained in the IPv4 header, IPv6 Hop-by-Hop extension header, or IPv6 Destination extension headers). Any fields in these headers/ extension headers which are modified in transit are set to 0 before applying the authentication algorithm. If a field is mutable, but its value at the receiving IPsec peer is predictable, then that value is inserted into the field before applying the cryptographic algorithm. In the case of ESP in transport mode, security services are provide only for the higher-layer protocols, not for the IP header or any extension headers preceding the ESP header.

Both the AH and ESP protocols can be used in tunnel mode for data packet endpoints as well as by intermediate security gateways. As shown in figure 7, in tunnel mode, there is an "outer" IP header that specifies the IPsec processing destination, plus an "inner" IP header that specifies the ultimate destination for the packet. The source address in the outer IP header is the initiating cryptographic endpoint; the source address in the inner header is the true source address of the packet. The security protocol header appears after the outer IP header and before the inner IP header.

Figure 7. IPv6 IPsec AH/ESP in Tunnel Mode



If AH is employed in tunnel mode, portions of the new outer IP header are given protection (those same fields as for transport mode, described earlier in this section), as well as all of the tunneled IP packet (that is, all of the inner IP header is protected as are the higher-layer protocols). If ESP is employed, the protection is afforded only to the tunneled packet, not to the new outer IP header.

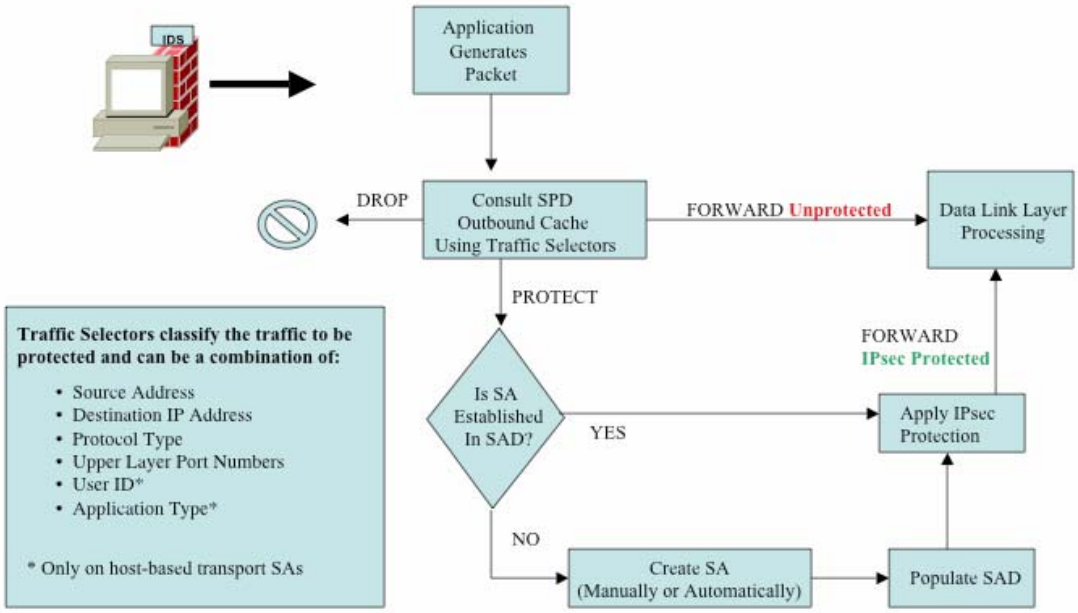
6.2. *Security Associations and Associated Databases*

The concept of a Security Association (SA) is fundamental to IPsec. A SA is a relationship between two or more entities that describes how the entities will use security services to communicate. The SA includes: the participating nodes and the upper layer protocols that require protection, the protocol used to provide security services (AH and/or ESP), the mode that the AH/ESP protocol is using (transport or tunnel mode) and the associated cryptographic algorithms and keys. An SA is unidirectional which means that for bi-directional communication, there are usually 2 SA's that need to be created at a given end-point. One for incoming traffic and the other for outgoing traffic.

An SA Database (SAD) is used to store and maintain all the SAs. A security policy database (SPD) specifies the policies that define which inbound and/or outbound packets require IPsec protection.

Figure 8 illustrates the actions required for outbound IPsec processing for a device.

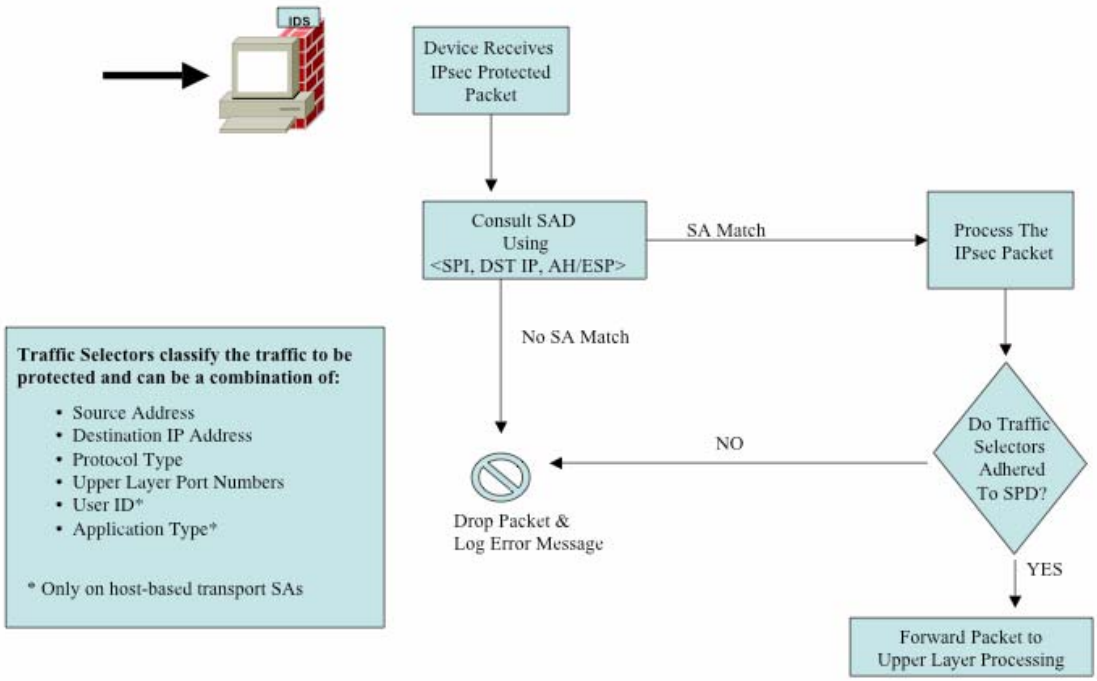
Figure 8. Outbound IPsec Processing



When an application generates a packet, the security policy database is consulted for any outbound entries matching the appropriate traffic selectors. If the packet requires IPsec protection, it looks up the SA in the SAD and applies the specified security protocol (AH/ESP) with its associated cryptographic algorithm and keys, inserting the SPI from the SA into the IPsec header.

Figure 9 illustrates the actions required for inbound IPsec processing.

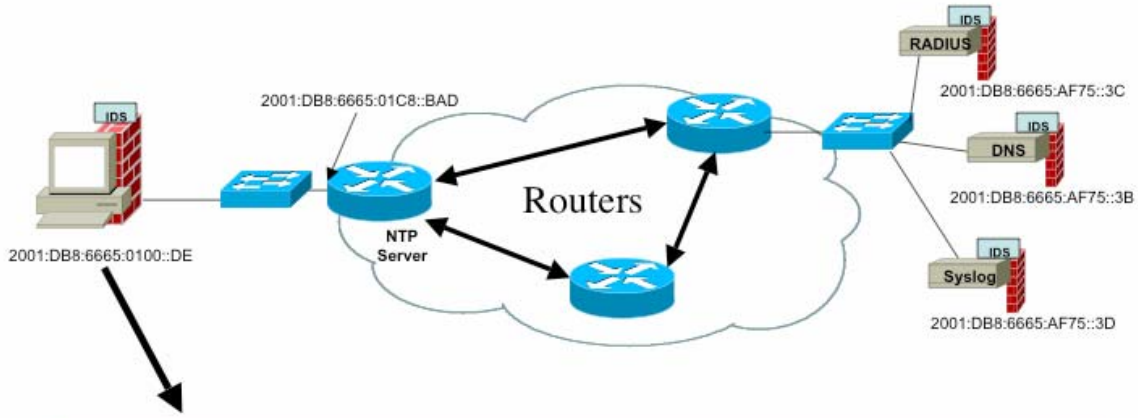
Figure 9. Inbound IPsec Processing



When the IPsec peer receives an IPsec protected packet, it looks up the SA in its database by destination address, security protocol, and SPI and then processes the packet as defined by the SA. If it cannot find the SA, it drops the packet and logs an error. Note that each entry in the SAD must indicate whether the SA lookup makes use of the destination address alone or a combination of the destination and source IP addresses, in addition to the SPI and security protocol. Upon successful IPsec processing, the SPD is consulted to ensure that the packet matches the correct IPsec selector parameters before it is forwarded up the stack to be processed by the upper layers.

SPD entries must be allowed to be explicitly ordered to enable a user or administrator to specify an access control policy in such a manner that traffic processing is reproducible and predictable, similar to the filtering rules that are common in packet-filtering firewalls. An example is shown in Figure 10 where a server requires varied protected communication.

Figure 10. SPD Ordering Requirement



Security Policy Database

From	To	Protocol	Dst Port	Policy
2001:DB8:6665:0100::DE	2001:DB8:6665:01C8::BAD	TCP	23 (Telnet)	ESP: SHA1, AES-256
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::3C	UDP	1812/1813 (RADIUS)	ESP: SHA1, AES-256
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::3B	TCP / UDP	53 (DNS)	ESP: SHA1, AES-128
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::3D	UDP	514 (Syslog)	ESP: SHA1, 3DES
2001:DB8:6665:0100::DE	2001:DB8:6665:AF75::/48	TCP / UDP	ANY	ESP: SHA1

The security policy dictates that all NTP, DNS, Syslog and AAA communications to the appropriate servers be protected by using ESP with encryption whereas all other traffic between those devices can use ESP with null encryption. [ESP with null encryption is the term used to specify that IPsec is being used for data origin authentication and integrity but not confidentiality.] If there wasn't the capability to define an explicit ordering, some implementations could erroneously cause traffic to not be encrypted by always matching on the more general rule which may have been to protect all traffic from the server to all others using ESP w/ null encryption.

The updated IPsec Architecture standard specifies that in host systems, applications may be allowed to create SPD entries. This enhancement can provide for more efficient application security developments, where applications can be programmed to automatically interact with a host's IPsec stack to create appropriate security controls per individual application.

6.3. *Managing Security Associations and Cryptographic Keys*

The updated IPsec Architecture standard mandates the use of both manual and automated SA and cryptographic key management. Manually creating SAs and cryptographic keys is useful in test scenarios and very small scale networks, but in operational deployments an automated key management mechanism is needed to make widespread IPsec usage practical. The Internet Key Exchange (IKE) protocol automates authentication of the IPsec peers, negotiates IPsec security associations, and establishes IPsec keys. IKE [RFC2409] has been updated to a new version, IKEv2 [RFC4306], which is intended to replace the original protocol (now referred to as IKEv1). It is important to note that IKEv2 is not backwards compatible with IKEv1 and appropriate considerations need to be accounted for in early deployments that already utilize IKEv1 and wish to migrate to IKEv2. An interim solution may be to include both IKEv1 and IKEv2 in systems until IKEv2 capable operating systems are deployed throughout an enterprise.

6.4. API Considerations

The IPsec APIs are still evolving. An early attempt to standardize a new socket protocol family (PF-KEY)[3] was defined in 1998. It was meant to be used by trusted privileged key management applications to communicate with an operating system's key management internals (the Security Association Database (SADB)) . An added attempt to standardize a basic IPsec socket option in the late 1990's was never finalized.[4]

The Advanced Socket API for IPv6 (rfc3542) specifically omitted options for controlling IPsec. However a joint project by WIDE and KAME (***www.kame.net***) had continued the initial IPsec API work and produced stable IPv6 IPsec and IKEv1 implementations for BSD Unix variants (FreeBSD, NetBSD, OpenBSD and BSDi). It includes the PF_KEY socket API which conforms to RFC2367 is used to access the IPsec key mangement engine. In addition, the API was extended to include control of the IPsec policy engine, i.e. the SPD and SAD.

A more recent API from NPI, the IPsec Service API (SAPI)[5] provides a generic interface for configuring and managing the IPsec rule databases (the Security Policy Database (SPD) and the Security Association Database (SAD)). These databases contain various attributes allowing a given IPsec implementation in the forwarding plane to determine how to handle ingress and egress IP data packets. Within the IPsec realm, the SPD defines what to do in handling a given IP packet, whereas the SAD defines how to do this. This IPsec SAPI allows a client application to receive event notifications indicating state changes, alerts and other information data.

[3] Daniel L. McDonald, Craig Metz, and Bao G. Phan, PF_KEY Key Management API, Version 2, RFC, 2367

[4] D. L. McDonald, A Simple IP Security API Extension to BSD Sockets, draft-mcdonald-simple-ipsec-api-03.txt

[5] http://www.npforum.org/techinfo/IA_Overview.shtml#IP_Security_IPSec_Service_API

It is expected that the efforts to create APIs will be consolidated and that one interoperable standard will emerge. With the recent updates to the Security Architecture for the Internet Protocol (RFC 4301) and Internet Key Exchange version 2 (IKEv2 ,RFC 4306) we believe the protocol is well enough defined to create a single interoperable standard. This would greatly simplify the work required by application developers and would facilitate migrating existing IPv4 applications to use IPv6 in conjunction with secured services utilizing IPsec. Note also that parts of the socket API is changing and certain address agnostic calls have been introduced to take into consideration the trend of utilizing names which are becoming more important than actual addresses.

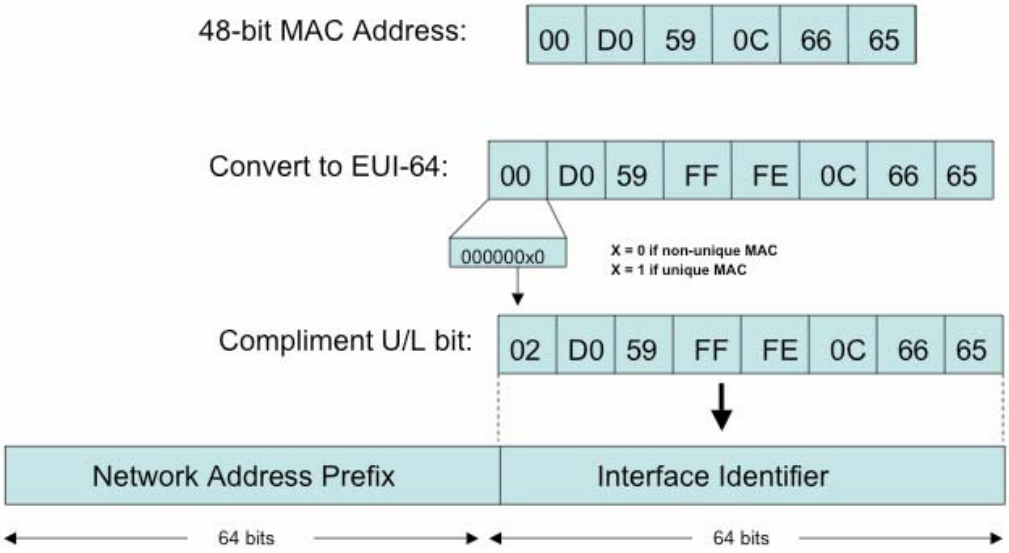
7. Addressing Security Considerations

One of the advantages of IPv6 is the large address space where it is expected that any conceivable networked device, be it in the commercial, consumer or government space, will have a unique address. However, since any IPv6 device can, and will, have multiple addresses associated with it, addressing architectures play an important role in how security policies can be constructed and enforced. A single device may eventually end up with multiple IPv6 addresses, such as a server with multiple applications if it is desired that each application have associated it's own /64. Addressing policies need to keep in mind filtering rule applicability such that appropriate filters can be applied at both the network and host levels.

Global IPv6 address assignment policies are still subject to some modifications but at the time of this writing it is a general policy for Regional Internet registries (i.e. RIPE, APNIC, ARIN, LACNIC and AFNIC) to allocate a /32 to qualified service providers who in turn would follow RFC3177 which provides recommendations on policies for assigning IPv6 address blocks to end sites. In particular, it recommends the assignment of a /48 in the general case, a /64 when it is known that one and only one subnet is needed and a /128 when it is absolutely known that one and only one device is connecting.. It is then up to the end-site to define their own address allocation policy, just like in IPv4, although some of the policy considerations may be different from that of the IPv4 policy. For example, in IPv4 it was common practice to define an easy-to-remember address for critical infrastructure devices which was found to lead to obvious reconnaissance attacks. The IPv6 addresses are almost impossible to remember and it doesn't make sense to define an easy-to-remember IPv6 address to infrastructure devices so they should be obscured as much as possible. Instead, with IPv6, DNS will play a more important role when dealing with IPv6 addresses and it is expected that naming conventions will become much more important.

A critical IPv6 network design consideration is how hosts connected to an IPv6 network create their interface identifiers since this has several security implications. IPv6 addresses are formed by combining network prefixes with an interface ID. The interface ID is guaranteed to be unique on a given subnet and comprises of the 64 rightmost bits of an 128-bit IPv6 address. Often, if the interface ID is automatically generated, it is based on the EUI-64 format that is constructed from the IEEE 48-bit MAC address, as shown in figure 11.

Figure 11. General IPv6 Address Structure and Interface ID



The interface ID can be manually configured or derived using either stateless or stateful autoconfiguration. While IPv6 was designed with automation in mind, automatic configuration and security are often at opposing ends of the spectrum and the trade-offs must be adequately considered. Where addressing is concerned, the majority of the security concerns lie in mitigating the possibility that an IP address can be spoofed or modified in transit and that traffic can be audited such that any malicious traffic can be traced to its source. The next sections will take a look at some of these tradeoffs.

7.1. *Manually Configured Addresses*

It is rare that manually configured interface ID's will be used. In practice it has mostly been used to create specific point-to-point or other tunnel end-point addresses which do not conform to the /64 subnet boundaries. Note that RFC3627 indicates that the use of a /64 is the best solution for point-to-point links while a /112 can be used if that's not possible. However, in current

deployments where it is felt that using a /64 is wasteful for point-to-point links, many opt to use a /127 or /126 subnet boundary and create manually defined IPv6 addresses for the point-to-point or tunnel endpoints.

An important consideration for manually configured addressing is to make them hard to guess whenever possible. When manually configuring interface ID's, the more common forms of starting at the beginning or end of a subnet boundary (i.e using a 1 or FF for routers) should be avoided. This will make any potential reconnaissance attack attempt much more difficult. Although some common multicast groups are defined for important networked devices and use of commonly repeated addresses make it easy figure out what the name servers, routers or other critical devices are, a non-random manual address scheme also makes it easy for a potential attacker using a "dictionary attack" of commonly used interface IDs to find your critical infrastructure.

Note that appropriate filtering mechanisms and auditing traffic to and from critical devices that are manually configured will help mitigate security risks – this is true for most attacks based on address spoofing/manipulation.

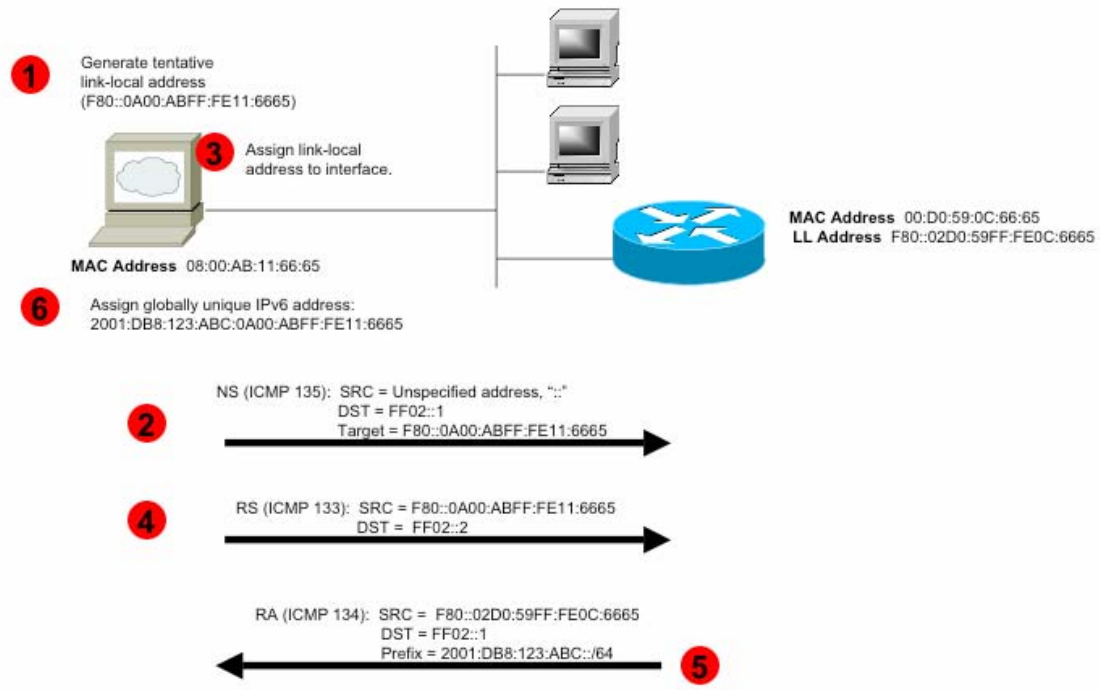
7.2. *Stateless Autoconfiguration*

Stateless autoconfiguration (RFC 2462) enables basic configuration of the IPv6 interfaces in the absence of a DHCPv6 server. This technique allows systems to generate their own local and global IP addresses and checks for address duplication. It utilizes the neighbor discovery (ND) protocol (RFC 2461) which defines the neighbor solicitation (NS) process, the neighbor advertisement (NA) process, the router solicitation (RS) process, the router advertisement (RA) process and the duplicate address detection (DAD) mechanism.

Stateless autoconfiguration relies on the information in the RA messages to configure the interface. The /64 prefix included in the RA is used as the prefix for the interface address. For Ethernet, the remaining 64 bits are obtained from the interface ID in EUI-64 format. Thus, an IPv6 node can autoconfigure itself with a globally unique IPv6 address by appending its link-layer address-based interface ID to the /64 prefix.

Figure 12 illustrates the operation of stateless autoconfiguration.

Figure 12. Stateless Autoconfiguration



The steps are as follows:

Step 1. When the IPv6 host first connects to the network, the host automatically generates a tentative link-local IPv6 address based on the MAC address of the interface.

Step 2. The host then ensures uniqueness by performing DAD. First the host transmits a NS message to the all-nodes multicast address (FF02::1) using its tentative link-local address as the target.

Step 3. Neighboring hosts see those multicast solicitations and if the address is in use an NA is returned and the address cannot be assigned to the interface. In this example there is no response and the link-local address is assumed to be unique and available and is assigned to the interface.

Step 4. The node now sends an RS message to the all-routers multicast group (FF02::2). Note that this step may be omitted.

Step 5. The responding RA provides address prefixes, link configuration parameters, and

information as to whether or not to use a stateless or stateful method for global address assignment, and additional network configuration parameters using the Dynamic Host Configuration Protocol for IPv6 (DHCPv6).

Step 6. Assuming the host is instructed to use the stateless method for address configuration, it can now use the router prefixes announced to form IPv6 addresses from those prefixes by appending the EUI-64 format link-local address to that prefix to create an IPv6 Address. If the host is instructed to use the stateful method for address configuration, then DHCPv6 can be used to configure additional hosts' addresses - this is described in more detail in section 7.3. Note that the DAD process need not be repeated for stateless autoconfiguration since it was already previously determined that the link-local address was unique and available.

While many of the security threats against neighbor discovery have been documented in IPv6 ND Trust Models and Threats [RFC 3756], the majority of the security issues with neighbor discovery and stateless autoconfiguration relate to the following issues/threats:

- Neighbor solicitation (NS) or neighbor advertisement (NA) spoofing to get access to a local link such that a malicious end node can get access to other node on that local link.
- Getting access to the local link and receiving RA messages to generate a spoofed global address and gain unauthorized access to other parts of the network.
- Redirect attacks where a malicious user could generate fake RAs and set up a router connected to a network and be able to capture every outgoing packets from that network and redirect it to another destination.
- Denial-of-Service attacks such as a DAD attack where no legitimate node can connect to the network or some other variant which are primarily caused by NS and NA messages that are spoofed.

IPsec is not always a valid security option because of a bootstrapping problem. In instances where you have no apriori trust relationship, you need to first establish an IPv6 address in order to set up the IPsec security associations. This creates a chicken-and-egg problem when the parameter you are trying to derive in a secure manner is the actual IPv6 address. Note however, that IPsec can be used in environments where an apriori trust relationship exists and there is a pre-defined security model in place which relies on either pre-configured keys (i.e. pre-shared keys) or a PKI Infrastructure.

So how do you authenticate nodes, authorize MAC address to IPv6 address bindings and provide access control for your node addresses?

To avoid end nodes being visible from the outside world, you could announce RAs with Unique Local IPv6 Unicast Addresses [RFC 4193] . There is currently no per-host policy control to limit reachability although there is subnet-grain control whereby stateless autoconfig does not always

mean worldwide reachability. A simple analogy is the "roaming" cell phone. The cell phone is permitted to obtain dialtone anywhere, regardless of whether a particular company likes the idea that someone could enter their premises, and place a phone call from inside their building. IPv6 Stateless Auto-Configuration is comparable to dial tone.

Many of the security issues surrounding spoofed addresses can be resolved by utilizing the Secure Neighbor Discovery (SeND) protocol which is detailed in the next section.

7.2.1. Secure Neighbor Discovery (SeND)

Secure Neighbor Discovery (SeND) [RFC 3971] is a mechanism designed to secure neighbor discovery without using IPsec. Although existing IPv6 standards specify that IPv6 neighbor discovery and address autoconfiguration mechanisms may be protected with IPsec AH, the practical solutions were found to be limited to manually preconfiguring any IPsec security associations which is unacceptable in any practical large-scale deployments. IPv6 Neighbor Discovery Trust Models and Threats [RFC 3756] discusses the requirements for securing neighbor discovery, which resulted in the creation of the SeND protocol. .

How does SeND work? Two new ICMPv6 options are specified: the RSA signature option and the Cryptographically Generated Addresses (CGA) option. The Neighbor Discovery RSA public key signatures are used to protect all messages - it ensures the integrity of the message and provides authentication for the identity of the sender. Authority of the public key is established by an authorization delegation process through the use of digital certificates. The address ownership proof mechanism is performed by using CGA. SEND also provides replay protection through the use of a timestamp (for multicast traffic) and a nonce (for traffic between a communicating pair).

SEND protects against:

- Spoofed Messages To Create False Entries In Neighbor Cache
- Neighbor Unreachability Detection Failure
- Duplicate Address Detection DoS Attack
- Router Solicitation and Advertisement Attacks
- Replay Attacks
- Neighbor Discovery DoS Attacks

SEND does NOT protect against

- Statically configured addresses
- Addresses configured using fixed identifiers (i.e. EUI-64)
- Network snooping – i.e. it does not provide confidentiality
- An unsecured link-layer – there's no guarantee that payload packets came from a node that

used SEND

While no known shipping SEND implementations are known, there are a few vendors who are in the process of implementing this protocol into their products. There are some intellectual property rights concerns since CGA is based on protocols that have Intellectual Property Rights (IPR) claims on them and though they are offered on a "Royalty-Free, Reasonable and Non-Discriminatory License to All Implementers", the fact that a license is required may hinder the widespread adoption by many implementers. [6]

7.2.2. Using IPsec to Secure Neighbor Discovery

IPsec can also be used to protect neighbor discovery if you are in an environment with an apriori trust relationship between communicating peers, i.e. where security credentials such as a pre-shared key or a PKI cert were pre-established between the peers. In this manner, the peers could each use a temporary IPv6 address to initially set up an IPsec SA and then derive the real IPv6 addresses using the secured IPsec communication. Once the legitimate IPv6 addresses were established, 'new' IPsec SA's would be created using these new IPv6 addresses for any further communication. By virtue of having the capability to initially establish an IPsec SA with a peer it is assumed that you are communicating with a trusted entity. This scenario would not prevent malicious nodes from sourcing traffic on the local network or trying to gain a valid IPv6 address but if there wasn't a mechanism for that malicious node to establish an initial IPsec SA, then damage would be limited to just the local network. Note that more work needs to be done to adequately understand how to recover from a compromised apriori key and ascertain what the actual limitations of this scenarion are. However, it does offer an alternative to SeND in some environments.

7.3. *Stateful Autoconfiguration and DHCP Considerations*

Using DHCPv6 [RFC 3315], IPv6 also supports stateful configuration of IP addresses to nodes. Clients can send a Solicit message to the All_DHCP_Relay_Agents_and_Servers address and request IP address assignment and other configuration options from the DHCPv6 server. Called stateless DHCPv6 [RFC 3736], this option is used to deliver information such as Domain Name System (DNS) [RFC 3646] or Session Initiation Protocol (SIP) server addresses. The client-server message exchange can consist of either two or four messages.

[6] CGA IPR Statements to the IETF are available from:
https://datatracker.ietf.org/public/ipr_search.cgi?option=rfc_search&rfc_search=3972

In the case of routers, it is typical to want to automatically configure IPv6 prefixes. In some situations a delegating router may not have knowledge about the topology of the networks to which the requesting router is attached. This is a likely scenario when a service provider assigns a prefix to a Customer Premise Equipment (CPE) device acting as a router between the customer's internal network and the service provider's internal network. Because CPEs might receive a /48 or /64 prefix, a technique called DHCPv6 prefix delegation [RFC 3633] can be used by delegating routers to assign variable-length prefixes to requesting routers or CPEs. Requesting routers use DHCP options to request prefix(es) from the delegating router via a unique identifier.

DHCPv6 exchanges can be hardened against spoofing through integrating the DHCP service with IPSec, secure neighbor discovery, or simple AAA authentication methods. DHCP can also be used to enforce address assignment policy, such as rotating addresses for privacy or to prevent effective network surveillance.

DHCP is currently the only widely deployed method for automatically sending DNS server addresses to host applications . This alone makes DHCPv6 a necessary part of most enterprises IPv6 networks even if the network deploys stateless autoconfiguration methods. It is a necessary autoconfiguration tool to assign network service information to hosts during bootstrapping since information about network servers is not carried by router advertisements.

7.4. *Further DHCP Considerations*

DHCP is currently required to auto-configure DNS information for IPv6 hosts, unless manual DNS configuration is done which is operationally not maintainable for large networks. Unless a DNS router advertisement option is deployed in normal IPv6 router autoconfiguration, the reliance on DNS autoconfiguration in most networks means that most enterprises will continue to require DHCP.

The currently deployed security architectures are another reason to use DHCP. With current reliance on stateful addresses and access control list security, it is far simpler to deploy a stateful DHCP solution for configuring the global addresses and DNS/router information for hosts. DHCP is very well understood and it's very easy to add authentication to the process. Until the industry gets better adoption and understanding of using SEND and CGA and other components that support stateless security, it is advisable to continue to use DHCP to autoconfigure global addressing for hosts.

DHCPv6 also offers a good method to centrally control IP address number assignments to achieve security or QOS policies. As an example, DHCPv6 could be used to assign, rotate, and log assignment of randomly generated addresses to achieve addressing privacy. By having stateful control of random addressing, various security policies and auditing requirements can be enforced.

There is some misunderstanding in the industry that users will not use DHCPv6 for IPv6 LAN configuration. On the contrary, most will begin to use DHCPv6 as the server is used today for

DHCPv4 and migrate to Stateless autoconfiguration. The DHCPv6 PD for routers is only for prefix delegation not full configuration from DHCPv6 server (e.g. DNS Server, Time Stamps, Link Information, et al).

7.5. *Privacy Addresses*

Randomly generating an interface ID, as described in RFC 3041, is part of stateless autoconfiguration and used to address some security concerns. Stateless autoconfiguration relies on the automatically generated EUI-64 node address, which together with the /64 prefix make up the global unique IPv6 address. As was illustrated in figure 12, the EUI-64 address is generated from the MAC address. Since MAC addresses for specific vendor equipment can be know, it may be easy for a potential attacker to perform a more directed intelligent scan to try and ascertain specific vendor device reachability for exploitation. Privacy addressing attempts to mitigate this threat.

As privacy addressing could also be used to hide illegal and unsavory activities, privacy addressing can be assigned, audited, and controlled in managed enterprise networks via DHCPv6.

Some people also feel that stateless addressing means that we may not know addresses operating in our networks ahead of time in order to to build access control lists (ACLs) of authorized users. While privacy addresses are truly generated randomly to protect against user tracking, but assuming that nodes use the EUI-64 format for global addressing, a list of *expected* pre-authorized host addresses can be generated. DHCP can also be used to statefully assign addresses. Of course the MAC used to generate an EUI-64 can be changed/spoofed and with manual configuration any address can be changed/spoofed as can the IPv4 address today. For this reason, using ACLs that rely on knowing the stateful IP transport address (either IPv4 or IPv6) provide a false sense of security since the least sophisticated hackers can easily bypass address screening. Cryptography using either IPsec or cryptographically generated addresses is a far better method to establish trusted access.

7.6. *DNS Considerations*

As mentioned earlier in this paper, DNS operations and security become more critical to network operations as more communications are converged on the IPv6-based Internet. Most every Internet user relies on DNS's keyword-based redirection service to locate the network addresses of Internet resources. The DNS server implementing Berkeley Internet Name Domain (BIND) version 9 or higher or some newer commercial DNS solutions can be used for dual-stack DNS operations. A new IPv6 resource record type called “AAAA” record is designed to carry four 32bit records to make a complete 128bit IPv6 address while maintaining compatible with the existing “A” record used in IPv4 DNS implementations. In the forward zones, IPv6 addresses are represented using AAAA records. In the reverse zones, IPv6 addresses are represented using PTR records in the nibble format under the ip6.arpa tree. Generally the IP transport (v4 or v6) should be independent from the record type so that IPv4 can carry both A and AAAA and DNS queries across IPv6

transport can also carry both types. To prevent Internet fragmentation all IPv6-capable DNS recursive resolvers should be dual-stacked to maintain backwards compatibility with IPv4-only legacy DNS servers. See RFC3596 for more about IPv6 DNS usage, and RFC3363 or RFC3152 for background information.

Most operational issues are discussed in RFC4472, Operational Considerations and Issues with IPv6 DNS. Most operational considerations with DNS are not security issues with the exception that you can think of them as potential methods for accidental or deliberate denial of service attacks. The DNS vulnerabilities to be concerned about, regardless of transport used, include corrupting the data, unauthorized updates, impersonating a primary name server, cache pollution by data spoofing, and cache impersonations. DNS data can be spoofed and corrupted on its way between server and resolver or forwarder because the current DNS protocol does not enable you to check the validity of the DNS data. Many of the issues surrounding securing the DNS transactions are addressed via DNS Security (DNSsec) and TSIG which provide better authentication mechanisms based on cryptographic signatures to validate the integrity and origin of the DNS data. They are mostly transport agnostic and whether you are using IPv4 or IPv6 has little impact.

Some security points to keep in mind for IPv6 DNS are worth mentioning. For one, local addresses should never be published so they should never be included in forward or reverse zone files. Use of a split DNS, the creation of an internal view and an external view, is considered by many to be a good solution. Mobile nodes that do not wish to have their new locations tracked via IP addressing should not use active DNS to post changing network addresses, but rather they should employ Mobile IPv6 with a home agent on their home network to handle address forwarding. Also, reverse chains for 6to4 addresses and Teredo addresses are impractical with dynamic DNS updates. Lastly, it is always good practice to provide some filtering of both TCP and UDP port 53 traffic at the network infrastructure level. However, keep in mind that the maximum size of a UDP DNS response is 512 bytes and if a DNS resource record exceeds this size, it will be truncated and rather than send UDP fragments, the information is sent via TCP. If inbound TCP port 53 is blocked (both source and destination port) to prevent unauthorized zone transfers, you also block any external host from resolving large responses. To avoid this problem, block traffic to destination port 53 only and allow traffic to source port 53 that already has an established connection.

8. Transition Mechanism Security Considerations

Many environments which currently run IPv4 will have a definitive transition strategy in place to migrate to IPv6. Most transition strategies will be a combination of dual-stack environments and tunneling. In a dual-stack approach both IPv4 and IPv6 is running on the device and the application will decide which transport layer to use. In a tunneled situation, the there may exist IPv6 traffic encapsulated in IPv4 packets or IPv4 traffic encapsulated in IPv6 packets. The tunnels can be either manually configured or automatically established. The main security concern for any of these

transition mechanisms is an operational one since there is now added complexity for configuring devices as well as logging and monitoring the traffic. Another concern is whether current security devices such as firewalls, IDS and IPS systems are able to fully support a dual-stacked and/or tunneled environment. It would be desirable to have mechanisms to correlate logs and auditing tools on both IPv4 and IPv6 traffic to discern any potential attacks which may use both transports as a means to obscure the attack.

Even if an organization is delaying its migration to IPv6, IPv6-capable IA infrastructure may be needed to look for IPv6-based tunneling security concerns. Some IPv6 tunneling mechanisms (as well as IPv4-in-IPv4 tunnels) have been exploited to specifically penetrate IPv4 firewalls and NATs by riding over common traffic ports. Any transition mechanism that relies on one of the many tunneling mechanisms available is subject to a number of security issues which include

- Bypassing ingress filtering checks since it is typical to create a hole in the firewall to allow tunneled traffic to pass through (i.e. permit protocol 41).
- Exploiting the tunnel interface. Several IPv6 security mechanisms depend on checking that the hop count equals 255 and/or that link-local addresses are used to ensure that packets originated on-link and can be 'trusted'. Tunnels are more vulnerable to a breach of this assumption than physical links, as an attacker anywhere in the Internet can send an IPv6-in-IPv4 packet to the tunnel decapsulator, causing injection of an encapsulted IPv6 packet to the configured tunnel interface unless the decapsulation checks are able to discard packets injected in such a manner.[7]

As early as 2001, there was an exploit which used IPv6 tunnels to create an IRC channel used for malicious behavior.

8.1. *Manually Configured Tunnels*

Manually configured tunnels have end-points that are statically defined which allows more control of the tunnel set-up. To mitigate the IPv4 address of the encapsulating ("outer") packet from being spoofed, ingress filtering should be deployed. However, in the event that ingress filtering is not ubiquitously deployed, the decapsulating device should accept only encapsulated packets from the explicitly configured source address (i.e., the other end of the tunnel). While this does not provide complete protection it does provide a significant increase in security. Additionally, the IPv6 address of the encapsulated ("inner") packet can be spoofed since it may not be subject to IPv6 ingress filtering. To mitigate this latter issue, the decapsulating device should verify whether the inner IPv6 address is a valid IPv6 address and also use IPv6 ingress filtering before accepting the IPv6 packet.

[7] RFC4213 Basic Transition mechanisms for IPv6 Hosts and Routers

Manually configured tunnels can use added protection by deploying IPsec between the tunnel endpoints. Currently there is ongoing work in the ietf (draft-ietf-v6ops-tunnels-02.txt) to standardize the securing of IPv6-in-IPv4 tunnels.

While manual tunnels can offer some more control over automated tunnels, the administrative overhead to configure the manual tunnels are not always operationally optimal.

8.2. *Automatic Tunnels*

Automated tunnels require much less administrative work than manually configured tunnels but they are also more susceptible to misuse since there is no pre-configured end-point association and at the receiving tunnel end the packets have to be accepted and decapsulated from any source. More care needs to be taken for automated tunnels to monitor traffic and detect abnormal behavior. The problem you may run into with automated tunnels is not being able to trace back a problem to its source unless you have a means to capture the traffic and analyze it. Some of the more specific issues surrounding 3 common automated tunneling mechanisms (6to4, ISATAP, Teredo) are listed below.

8.2.1. 6to4

The 6to4 transition mechanism is documented in rfc3056. This architecture uses automatic IPv6-over-IPv4 tunneling to interconnect IPv6 networks and uses 6to4 routers and relays which must behave as follows:

- All 6to4 routers must accept and decapsulate IPv4 packets from every other 6to4 router and from 6to4 relays
- All 6to4 relay routers must accept traffic from any native IPv6 node

The security issues for 6to4 transition mechanisms have been documented in ‘Security Consideration for 6to4’, RFC 3964. This section will highlight the main concerns from this document but the reader is encouraged to read rfc 3964 for a more detailed security analysis.

Most of the security issues surrounding 6to4 are based on being able to authenticate legitimate endpoints. One critical problem is that 6to4 routers are not able to identify whether any 6to4 relays are legitimate. Another problem is that 6to4 relays can be subject to "administrative abuse" e.g., theft of service or being seen as a source of abuse. Also, the 6to4 architecture can be used to participate in DoS or reflected DoS attacks or made to participate in "packet laundering", i.e.,making another attack harder to trace – this makes the logging and auditing functions of 6to4 traffic extremely critical

It is extremely important that 6to4 router or relay security checks be correctly implemented to gain some trust that at least the basic known security issues can be appropriately mitigated.

6to4 Routers

The 6to4 routers act as the border routers of a 6to4 domain. They do not have a native global IPv6 address except in certain special cases. The main functions of the 6to4 router are as follows:

- Provide IPv6 connectivity to local clients and routers.
- Tunnel packets sent to foreign 6to4 addresses to the destination 6to4 router using IPv4.
- Forward packets sent to locally configured 6to4 addresses to the 6to4 network.
- Tunnel packets sent to non-6to4 addresses to the configured/closest-by-anycast 6to4 relay router.
- Decapsulate directly received IPv4 packets from foreign 6to4 addresses.
- Decapsulate IPv4 packets received via the relay closest to the native IPv6 sources. Note that it is not easily distinguishable whether the packet was received from a 6to4 relay router or from a spoofing third party.

The 6to4 router should perform security checks on traffic that it receives from other 6to4 relays, or 6to4 routers, or from within the 6to4 site. These checks include the following:

- Disallow traffic that has private, broadcast or certain specific reserved IPv4 addresses in tunnels, or the matching 6to4 prefixes.
- Disallow traffic from 6to4 routers in which the IPv4 tunnel source address does not match the 6to4 prefix. (Note that the pseudo-interface must pick the IPv4 address corresponding to the prefix when encapsulating, or problems may ensue, e.g., on multi-interface routers.)
- Disallow traffic in which the destination IPv6 address is not a global address; in particular, link-local addresses, mapped addresses, and such should not be used.
- Disallow traffic transmission to other 6to4 domains through 6to4 relay router or via some third party 6to4 router. (To avoid transmission to the relay router, the pseudo-interface prefix length must be configured correctly to /16. Further, to avoid the traffic being discarded, 6to4 source addresses must always correspond to the IPv4 address encapsulating the traffic.)
- Discard traffic received from other 6to4 domains via a 6to4 relay router.
- Discard traffic received for prefixes other than one's own 6to4 prefix(es).

6to4 Relay Routers

The 6to4 relay routers act as a relay between all 6to4 domains and native IPv6 networks. The main functions are as follows:

- Advertises the reachability of the 2002::/16 prefix to native IPv6 routing, thus receiving traffic to all 6to4 addresses from the closest native IPv6 nodes,

- Advertises (if RFC 3068 is implemented) the reachability of IPv4 "6to4 relay anycast prefix" (192.88.99.0/24) to IPv4 routing, thus receiving some tunneled traffic to native IPv6 nodes from 6to4 routers.
- Decapsulates and forwards packets received from 6to4 addresses through tunneling, by using normal IPv6 routing, and
- Tunnels packets received through normal IPv6 routing from native addresses that are destined for 2002::/16 to the corresponding 6to4 router.

The 6to4 relay should also perform security checks on traffic that it receives from 6to4 routers, or from native IPv6 nodes. These checks are as follows:

- Disallow traffic that has private, broadcast, or certain specific reserved IPv4 addresses in tunnels, or in the matching 6to4 prefixes.
- Disallow traffic from 6to4 routers in which the IPv4 tunnel source address does not match the 6to4 prefix. (Note that the pseudo-interface must pick the IPv4 address corresponding to the prefix when encapsulating, or problems may ensue, e.g., on multi-interface routers.)
- Disallow traffic in which the destination IPv6 address is not a global address; in particular, link-local addresses, mapped addresses, and such should not be used.
- Discard traffic received from 6to4 routers with the destination as a 6to4 prefix.

By performing sufficient ingress sanity checks, logging and auditing the tunneled traffic and providing authentication where possible, many of the 6to4 security risks can be effectively mitigated.

8.2.2. ISATAP

ISATAP (Intra-Site Automatic Tunnel Addressing Protocol) is an experimental protocol defined in RFC4214. It connects IPv6 hosts/routers over an IPv4 network. Contrary to 6to4, ISATAP views the IPv4 network as a link layer for IPv6 and views other nodes on the network as potential IPv6 hosts/routers and therefore does not require the underlying IPv4 network infrastructure to support multicast.

ISATAP defines a method for generating a link-local IPv6 address from an IPv4 address. This is done by concatenating FE80:0000:0000:0000:0000:5EFE with the 32 bits of the host's IPv4 address (expressed in hexadecimal form). Neighbor Discovery is performed on top of IPv4. Due to the lack of multicast support, automatic Router Discovery cannot be performed. ISATAP hosts must instead be configured with a potential router list (PRL). Each of these routers are infrequently probed by an ICMPv6 Router Discovery message, to determine which of them are functioning, and to perform unicast-only autoconfiguration. In practice, implementations build their PRL by quering

DNS, e.g. by looking up isatap.example.net if the local domain is example.net. The local domain is usually obtained using DHCP (over IPv4) or statically configured.

Securing ISATAP architectures follows the same similar principles of a layered security approach as most other tunneling mechanisms. The IPv4 virtual link must be delimited carefully at the network edge, so that external IPv4 hosts cannot pretend to be part of the ISATAP link. Site border routers should implement IPv4 ingress filtering and IP protocol 41 filtering.

Additionally, the PRL provides a list of IPv4 addresses representing advertising ISATAP interfaces of routers that hosts use in filtering decisions. Site administrators should ensure that the PRL is kept up to date. Lastly, IPsec should be used to protect the ISATAP traffic – this is accomplished by configuring IPsec for IPv4 policy settings to protect all traffic with the IP protocol set to 41.

8.2.3. Teredo

Teredo is specified in rfc4380 and is meant as a short-term solution to the specific problem of providing IPv6 service to nodes located behind an IPv4 Network Address Translation (NAT) box. It enables nodes located behind one or more NATs to obtain IPv6 connectivity by tunneling packets over UDP. The architecture utilizes Teredo servers to learn a Teredo client's global address and to obtain connectivity and exchange packets with native IPv6 hosts through Teredo relays.. The Teredo servers are stateless, and only have to manage a small fraction of the traffic between Teredo clients; the Teredo relays act as IPv6 routers between the Teredo service and the "native" IPv6 Internet. The relays can also provide interoperability with hosts using other transition mechanisms such as "6to4".

Note that Teredo is defined to be used as a last resort:

> Nodes that want to connect to the IPv6 Internet SHOULD only use the Teredo service as a "last resort" option: they SHOULD prefer using direct IPv6 connectivity if it is locally available, if it is provided by a 6to4 router co-located with the local NAT, or if it is provided by a configured tunnel service; and they SHOULD prefer using the less onerous 6to4 encapsulation if they can use a global IPv4 address.[8]

There are numerous security issues which are identified in the Teredo specification. They are categorized as:

- Security risks of directly connecting a node to the IPv6 Internet by opening a valid outbound connection and therefore bypassing NAT

[8] RFC4380 Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)

- Spoofing of Teredo servers to enable a man-in-the-middle attack
- Potential attacks aimed at denying the Teredo service to a Teredo client
- Denial of service attacks against non-Teredo participating nodes that would be enabled by the Teredo service

Many of these security issues can be defended against by deploying a layered security architecture including ingress filtering for both IPv4 and IPv6, secured relay discovery procedure, monitoring Teredo traffic and by deploying IPsec between the Teredo components. Since Teredo traffic is IPv6 traffic tunneled using an IPv6 header and UDP port 3544, IPsec must be configured to use IPv4 policy settings to protect all traffic with the source or destination UDP port set to 3544. This should be implemented at the host firewall.

8.3. *Tunnel Brokers*

The use of tunnel brokers has become popular to provide an initial IPv6 deployment scenario. There are a variety of tunnel brokers although most commonly the term is used to refer to an IPv6 tunnel broker as defined in RFC3053 where IPv6 tunnels are provided to enduser/endsites using either manual, scripted or automatic configuration.In general tunnel brokers offer tunnels where IPv6 is tunneled directly inside IPv4 using protocol 41. However, since this is problematic in NAT environments, other mechanisms include using AYIYA (http://www.sixxs.net/tools/ayiya/) or Hexago's 6udp4 protocol, both of which send IPv6 inside UDP which is able to traverse most NAT setups and even firewalls.

It is possible for environments to deploy their own tunnel broker solution or to partner with a myriad of 3rd party tunnel broker providers as listed in http://www.sixxs.net/tools/aiccu/brokers/

8.4. *New Tunneling Standards*

A recent development has been the creation of the Softwire working group in the ietf : http://www.ietf.org/html.charters/softwire-charter. This working group is specifying the standardization of discovery, control and encapsulation methods for connecting IPv4 networks across IPv6 networks and IPv6 networks across IPv4 networks in a way that will encourage multiple, inter-operable implementations. It is expected that the outcome of this work will unify some of the existing tunneling mechanisms.

9. Mobility Security Considerations

The reader is expected to be familiar with the basic operation of a mobile environment in IPv6 networks. However, to ensure a basic understanding a brief review is given here and then the security aspects will be detailed.

9.1. *Basic Mobile IPv6 Operations*

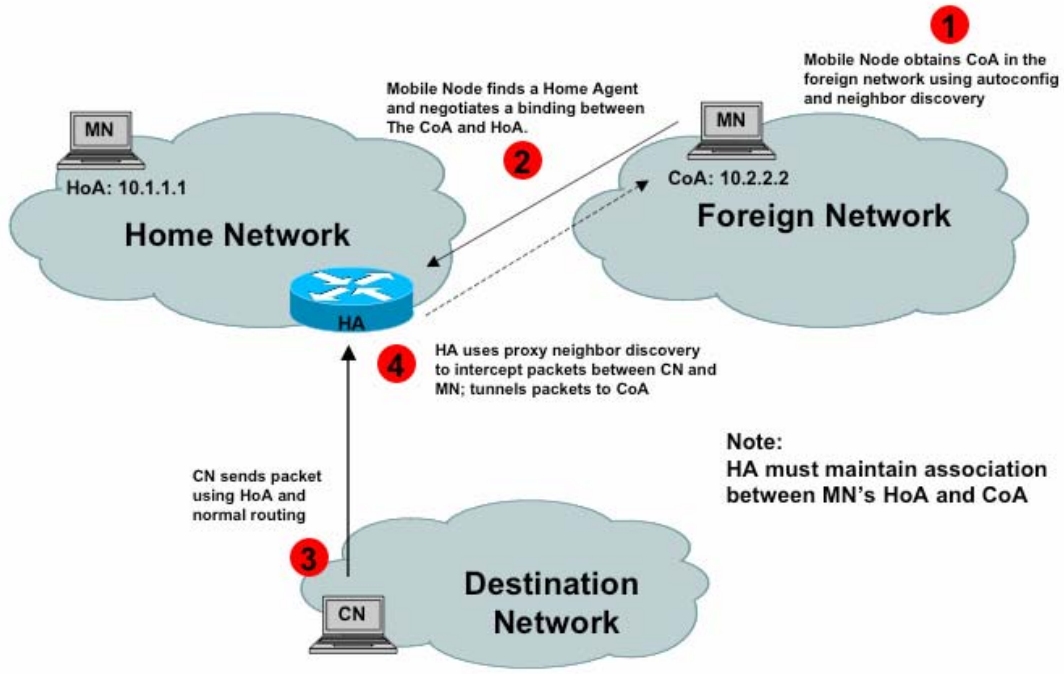
Mobility support in IPv6 is specified in RFC 3775. The architecture defines that each mobile node (MN) is always identified by its home address (HoA), regardless of its current point of attachment to the Internet. While a mobile node is connected to its home network, packets addressed to its home address are routed to the mobile node's home link using conventional Internet routing mechanisms. When a mobile node is away from its home link, the mobile node is also associated with a care-of address (CoA), which provides information about the mobile node's current location. IPv6 packets addressed to a mobile node's home address are transparently routed to its care-of address.

The association between a mobile node's home address and care-of address is known as a "binding" for the mobile node. While away from home, a mobile node registers its primary care-of address with a router on its home link, requesting this router to function as the "home agent" (HA) for the mobile node. The mobile node performs this binding registration by sending a Binding Update (BU) message to the home agent. The home agent replies to the mobile node by returning a Binding Acknowledgement (BA) message. The protocol enables IPv6 nodes to cache the binding of a mobile node's home address with its care-of address, and to then send any packets destined for the mobile node directly to it at this care-of address.

Any node communicating with a mobile node is referred to as a "correspondent node" (CN) of the mobile node, and may itself be either a stationary node or a mobile node.

Figure 13 illustrates the basic Mobile IPv6 operation when bidirectional tunneling is used. This scenario does not require Mobile IPv6 support from the CN and is available even if the MN has not registered its current binding with the CN. Packets from the CN are routed to the HA and then tunneled to the MN. Packets to the CN are tunneled from the MN to the HA (i.e. reverse tunneled) and then routed normally from the home network to the CN. In this mode, the HA uses proxy Neighbor Discovery to intercept any IPv6 packets addressed to the mobile node's home address on the home link. Each intercepted packet is tunneled to the mobile node's primary care-of address using IPv6 encapsulation.

Figure 13. Basic MIPv6 Operation



Route Optimization is a process where the MN informs the CN about its CoA directly, allowing packets from the CN to be routed directly to the CoA of the MN and bypassing the HA. When sending a packet to any IPv6 destination, the CN checks its cached bindings for an entry for the packet's destination address. If a cached binding for this destination address is found, the CN sets the destination address in the IPv6 header to the CoA of the MN, as indicated in the binding, and uses a new type of IPv6 routing header to route the packet to the MN.

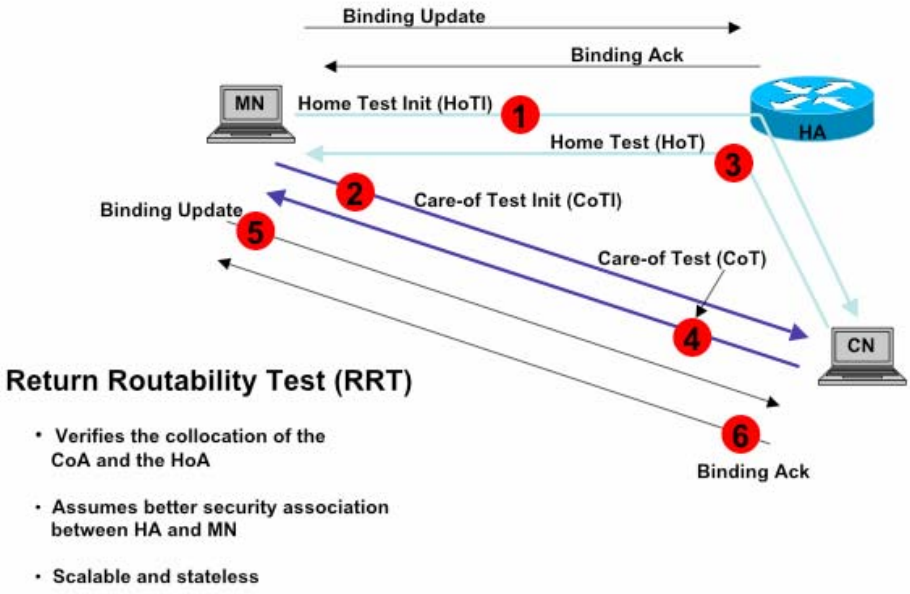
Similarly, the MN sets the Source Address in the packet's IPv6 header to its current CoA. The MN adds a new IPv6 "Home Address" destination option to carry its home address. The inclusion of home addresses in these packets makes the use of the CoA transparent above the network layer (e.g., at the transport layer)

Note that the new routing header , type=2, is only used in mobility environments. It is restricted to carry only one IPv6 address. For sanity checking, all nodes which process it must verify that the address contained in it is the node's own HoA and that it is unicast routable.

Routing packets directly to the mobile node's care-of address allows the shortest communications path to be used and eliminates congestion at the mobile node's home agent and home link. In addition, the impact of any possible failure of the home agent or networks on the path to or from it is reduced.

A security procedure used in route optimization before sending BUs to the CN is called the Return Routability Test (RRT). This is illustrated in figure 14. It is a signaling protocol between the MN and CN, where a mobile node instructs a correspondent node to direct the mobile node's data traffic to its claimed CoA. It verifies that the MN is reachable at its HoA and is able to send/receive packets at its CoA. The RRT requires that minimal state be stored at the CN to prevent DoS type attacks. Overall it provides some security assurance and prevents the misuse of Mobile IPv6 signaling to maliciously redirect the traffic or to launch other attacks.

Figure 14. MIPv6 Route Optimization with RRT



At a basic level, security defined in RFC3775 specifies the following:

- IPsec is used to protect the BUs and BAs between the MN and HA. Both the MN and HA must support and may use the ESP with NULL encryption in transport mode.
- Protecting BUs sent to a CN does not require IPsec but rather , uses the return routability procedure to assure that the right MN is sending the message. It employs a keyed-hash algorithm to provide the integrity and authentication of the BU messages to the CN. While this does not protect against attackers who are on the path between the home network and CN, it limits the potential attackers to those having access to one specific path in the Internet and avoids forged BU from anywhere else.
- No security is required for HA address discovery
- IPsec should be used between the MN and HA to protect Mobile Prefix Discovery (i.e. Mobile Prefix Solicitations and Advertisements).
- Mechanisms related to transporting payload packets - such as the Home Address destination option and type 2 routing header - have been specified in a manner which restricts their use in attacks.
- Traffic tunneled by MN through HA should use ESP with NULL encryption for all traffic amd must use ESP if multicast group membership protocols or stateful address autoconfiguration are tunneled to HA.

9.2. *Mobile IPv6 Security using IPsec*

The base Mobile IPv6 standard specifies that IPsec should be used to protect the signaling between the HA and MN. There are a number of standards and works in progress which enumerate and expand on the use of IPsec in more detail.

9.2.1. Using IPsec to Protect MN and HA Communications

RFC3776, Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents, provides details on how to protect the signaling between the HA and MN. IPsec ESP with NULL encryption in transport mode is used to protect the control traffic which consists of:

- BU and BA messages exchanged between the mobile node and the home agent
- Return routability messages Home Test Init and Home Test that pass through the home agent on their way to a correspondent node
- ICMPv6 messages exchanged between the mobile node and the home agent for the purposes of prefix discovery

The nodes may also optionally protect payload traffic passing through the home agent. If multicast group membership control protocols or stateful address autoconfiguration protocols are supported, payload data protection support is required.

The control traffic between the MN and the HA requires message authentication, integrity, correct ordering and anti-replay protection. The MN and the HA must have an IPsec security association to protect this traffic. Note that while IPsec does not proving correct ordering of messages, this service is ensured by a sequence number in the BU and BA messages. The sequence number in the Binding Updates also provides protection to a certain extent. It fails in some scenarios, for example, if the Home Agent loses the Binding Cache state. Full protection against replay attacks is possible only when IKE is used.

Great care is needed when using IKE to establish security associations to Mobile IPv6 home agents to ensure that the right kind of addresses must be used for transporting IKE. This is necessary to avoid circular dependencies in which the use of a Binding Update triggers the need for an IKE exchange that cannot complete prior to the Binding Update having been completed.

The following mandatory requirements apply to both home agents and mobile nodes:

1) Manual configuration of IPsec security associations must be supported. The configuration of the keys is expected to take place out-of-band, for instance at the time the mobile node is configured to use its home agent. Note that manual configuration is not practical to deploy in large scale operational environments.
2) Automatic key management with IKE may be supported. Only IKEv1 is discussed in RFC3776.
3) ESP encapsulation of Binding Updates and Acknowledgements between the mobile node and home agent must be supported and must be used.
4) ESP encapsulation of the Home Test Init and Home Test messages tunneled between the mobile node and home agent must be supported and should be used.
5) ESP encapsulation of the ICMPv6 messages related to prefix discovery must be supported and should be used.
6) ESP encapsulation of the payload packets tunneled between the mobile node and home agent may be supported and used.
7) If multicast group membership control protocols or stateful address autoconfiguration protocols are supported, payload data protection must be supported for those protocols.

9.2.2. MIPv6 with IKEv2 and Revised IPsec Architecture

Since the IPsec architecture has been revised in RFC4301, there is a new draft which, if approved, will obsolete RFC3776: Mobile IPv6 Operation with IKEv2 and the revised IPsec Architecture

(draft-ietf-mip6-ikev2-ipsec-06.txt). This work takes into account many of the enhancements that the new IPsec architecture provides. For example, the list of selectors has been expanded to include the Mobility Header message type which has an impact on how security policies and security associations are configured for protecting mobility header messages. It becomes easier to differentiate between the various Mobility Header messages based on the type value instead of checking if a particular mobility header message is being sent on a tunnel interface between the MN and the HA. The revised IPsec architecture specification also includes ICMP message type and code as selectors. This makes it possible to protect Mobile Prefix Discovery messages without applying the same security associations to all ICMPv6 messages.

Of the mandatory requirements listed in RFC3776, one was modified and two were added:

- There are no more recommendations regarding support of manual or dynamic IPsec configuration. The use of manually created IPsec security associations and the use of IKEv2 as the automated IPsec key management protocol are just described
- An additional requirement that the home agent and mobile node may support authentication using EAP in IKEv2
- An additional requirement that the home agent and the mobile node MAY support remote configuration of the home address. The home agent can pick a home address from a local database or from a DHCPv6 server on the home link.

9.2.3. MOBIKE

The IKEv2 Mobility and Multihoming Protocol (MOBIKE) is defined in RFC4555. It is an extension to the IKEv2 protocol which allows the IP addresses associated with IKEv2 and tunnel mode IPsec security associations to change. This would be useful in scenarios where a mobile VPN client could keep the connection to the VPN gateway active while in transit or where a multihomed host can seamlessly move the traffic to a different interface should the primary one become unusable.

MOBIKE updates only the outer (tunnel header) addresses of IPsec SAs, while the addresses and other traffic selectors used inside the tunnel stay unchanged. While it allows both communicating parties to move, it is best suited for situations where the address of at least one endpoint is relatively stable and can be discovered using existing mechanisms such as DNS.

The base version of the MOBIKE protocol does not cover all potential future use scenarios, such as transport mode, application to securing SCTP, or optimizations desirable in specific circumstances. Future extensions may be defined later to support additional requirements.

9.2.4. Using IPsec To Protect MN and CN Communications

There is also standards work in progress in securing the signaling between the MN and the CN using IPsec rather than the return routability procedure (draft-ietf-mip6-cn-ipsec-02.txt). This work is still in preliminary stages but may prove to be useful in providing added security afforded by using IPsec.

9.3. *Additional Mobile IPv6 Security Mechanism*

In addition to IPsec security services there exist a number of mechanisms which, if implemented can be used to provide additional or alternative security services.

9.3.1. Alternative Authentication Protocol

The Authentication Protocol for IPv6, defined in RFC 4285 is an informational document that proposes a an alternative solution to IPsec for securing the BU and BA messages between the MN and HA using a mobility message authentication option that is included in these messages. Such a mechanism enables IPv6 mobility in a host without having to establish an IPsec SA with its HA. The authentication mechanism proposed in this document is similar to the authentication mechanism used in Mobile IPv4

The mechanism used to authenticate the MN at the HA or at the Authentication, Authorization, and Accounting (AAA) server in the Home network (AAAH) is based on a shared-key-based mobility security association between the MN and the respective authenticating entity. This shared-key-based mobility security association (shared-key-based mobility SA) may be statically provisioned or dynamically created.

The confidentiality protection of Return Routability messages and authentication/integrity protection of Mobile Prefix Discovery (MPD) is not provided when this option is used for authentication of the MN to the HA. Therefore, unless the network can guarantee such protection (for instance, like in 3GPP2 networks), Route Optimization and Mobile Prefix Discovery should not be used when using the mobility message authentication option.

The scenarios where this authentication option could be considered fall under the following categories:

- Network deployments in which not all Mobile Nodes and Home Agents have IKEv2 implementations and support for the integration of IKEv2 with backend AAA infrastructures.
- Networks that expressly rely on the backend AAA infrastructure as the primary means for identifying and authentication/authorizing a mobile user for MIPv6 service.
- Networks in which the establishment of the security association between the Mobile Node and the authentication server (AAA Home) is established using an out-of-band mechanism

and not by any key exchange protocol. Such networks will also rely on out-of-band mechanisms to renew the security association (between MN and AAA Home) when needed.

- Networks that are bandwidth constrained (such as cellular wireless networks) and for which there exists a strong desire to minimize the number of signaling messages sent over such interfaces. MIPv6 signaling that relies on Internet Key Exchange (IKE) as the primary means for setting up an SA between the MN and HA requires more signaling messages compared with the use of a mobility message authentication option carried in the BU/BA messages.

9.3.2. Securing Mobile IPv6 Route Optimization

To optimize the route optimization process and use fewer exchanged messages, a standard was defined in RFC4449 which describes a mechanism for Securing Mobile IPv6 Route Optimization Using a Static Shared Key. The default mechanism specified in RFC3775 uses a periodic return routability test to verify that the MN has the right to use a specific HoA, as well as validate the claimed CoA. The advantage is that is requires no pre-configuration and no intermediary trusted entities.

While this new route optimization security mechanism offers an alternative lower latency method, it does require the configuration of a shared secret between the MN and its CN. This will limit its use to environments where some pre-configuration is acceptable and where mobile nodes can be trusted not to misbehave since the validity of the claimed CoA is not verified. Replay protection is provided through the use of the sequence number field in the BU.

Note that this mechanism in intended to only be used for BU messages and that a different pre-shared key must be used for each separate MN-to-CN binding. It is also recommended that this mechanism only be used in environments under the applicable nodes are in the same administrative domain.

9.4. *Mobile IPv6 Security Architectures*

When designing a secure mobile IPv6 infrastructure it is important to understand the potential technologies that may be available and which have been discussed in the previous sections. IPsec is the most versatile and usually the most optimal mechanism to provide the appropriate security services. Care must be taken to ensure that IKEv2 or MOBIKE is the key management protocol that is supported in any mobile device that wished to utilize IPsec since this offers the most optimal solution over IKEv1. By standardizing the use of EAP authentication to be used in IKEv2 it makes

it easier to implement authentication solutions that require both device and user-based credentials and can easily tie in with already existing legacy authentication mechanisms (i.e. AAA based solution such as RADIUS). Manually establishing security associations, while a standards requirement to be supported in products, is operationally prohibitive in large scale architectures.

Note that IKEv2 is increasingly being utilized in the IPv6 mobility standards work and therefore it is important to follow what vendors are implementing in their products. As mentioned in section 6.3, IKEv2 is not backwards compatible with IKEv1 and any security architecture utilizing IPsec must be aware of the consequences if some vendors do not support IKEv2.

As in any environment, care has to be taken to appropriately secure the Mobile IPv6 bootstrapping. For a MN to register with a HA it needs:

1. A home agent
2. A home agent address – this can be learned via manual configuration, anycast discovery mechanisms or DNS lookup
3. A security association with the home agent

There exists a basic trust relationship between the MN and the Mobile Service Provider (MSP) since it is assumed that the MN was provisioned with credentials to authenticate itself to the mobility or access service authorizer and to prove its authorization to obtain service. The configuration information that is exchanged between the MN and the HA needs to be secured using integrity and replay protection. Confidentiality protection should also be provided if necessary. The latter two points are easily done using IPsec.

Additionally there are architecture considerations when it comes to utilizing firewalls. RFC 4487 describes the problems which firewalls may cause in mobile IPv6 networks. In the worst case they may prevent Mobile IPv6 signaling and drop incoming and/or outgoing traffic. If the firewall configuration is modified in order to support the Mobile IPv6 protocol but not properly configured, many attacks are possible including a myriad of denial of service attacks. One example is the misuse of the type=2 routing header which is only applicable in IPv6 mobile environments. In non-mobility environments this header can be misused to redirect malicious traffic and it is good practice to drop packets with a type=2 routing header in environments that do not require mobility.

It is also important to keep in mind that a layered approach to security is always the best risk mitigation technique while keeping in mind to balance the risk versus the cost to provide the protection. You will still be required to harden the mobile hosts themselves in addition to protecting the communication between these hosts. Since the HA is a critical component of the architecture, great care needs to be taken to provide limited, authenticated, and audited access to these devices.

10. IPv6 Management / Security Auditing Tools

In any effective security architecture it is necessary to audit and verify that traffic adheres to required security policies. There are a multitude of reasons why not:

- Configuration mistakes
- Software bugs
- Malicious circumvention

In addition to monitoring the IPv6 traffic patters it is useful to have some sort of alarming system which can alert appropriate personnel of potential intrusions. These typically are a form of intrusion detection systems which have known worm/virus signatures that traffic is inspected against. As mentioned in a previous section, it is likely that host-based IDS systems will be used in parallel with network based IDS systems for IPv6 networks to ensure that even end-to-end encrypted traffic can be scrutinized. This can be important in areas where a supposedly trusted host initiates an end-to-end secure communication with the intent of perhaps causing a denial of service attack. If the host itself is intelligent enough to decrypt a few packets, perform a sanity check and decide that an attack may be in progress, appropriate filtering and circumvention steps can be taken to remove further impact from the source.

Currently the industry is developing a more cohesive network and host based security management solution. Many proprietary solutions exist including:

- Trusted Network Connect (TNC) architecture by the Trusted Computing Group consortium
- Network Admission control (NAC) by Cisco, Trend Micro and other associates
- Network Access Protection (NAP) by Microsoft and other associates

Any IPv6 strategy for secure managed networks should evaluate these security solutions and ensure that these architectures work in an IPv6 network..

10.1. *Management Tools*

Most currently available IPv6 management tools use IPv4 transport. This includes, NTP, SNMP, AAA (Radius and TACACS+) and Syslog. From an operational level, if a node is running in dual-stack mode then there is no reason that the transport should be an issue. It is more of a concern for environments which may to build a native IPv6 infrastructure in which case a v4inv6 tunneling mechanism may have to be created and creates similar tunneling security concerns as explained in the section on transition security considerations.

10.2. *Security Auditing and Network Assessment Tools*

The number of IPv6 enabled security auditing and network assessment tools is increasing as more nodes come on-line and the miscreant underworld discovers an easy target. At the time of this

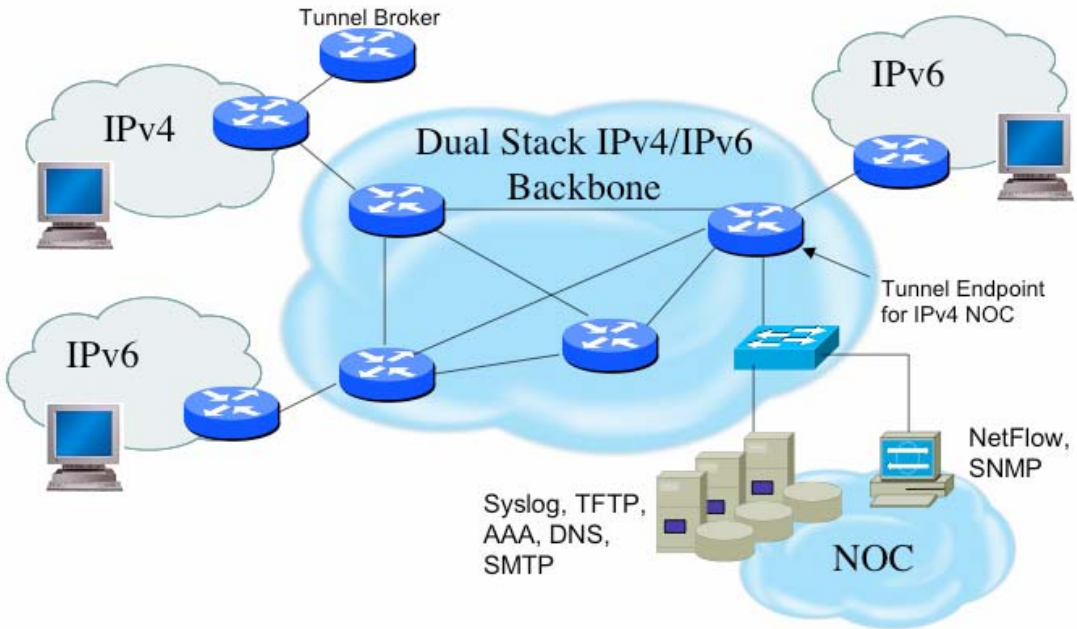
writing, the tools listed in Appendix A are available to help detect and proactively eliminate known vulnerabilities in internal IPv6-enabled infrastructures.

11. IPv6 Security Deployment Best Practice Guidelines

Security policies will dictate which technologies to deploy to ensure the most effective security architecture in a given environment. IPv6 has many capabilities to simplify and automate networked communication but many of these techniques must be considered carefully to understand the impact to a given security policy. This section will enumerate the operational security considerations from a practical deployment viewpoint.

Figure 15 illustrates the components of the sample IPv6 network. The IPv6 security architecture takes into considerations the following concerns: security policy considerations, end-host security and network infrastructure security. In all of these cases you need to be cognizant of preserving the confidentiality, integrity, accountability and availability of the devices and the data.

Figure 15. Sample IPv6 Architecture



11.1. *Security Policy Considerations*

A security policy enumerates the procedures that must be followed to ensure a protected IPv6 communication environment. This policy is the result of a risk assessment analysis and very much tied to the business and operational practices of that corporation. While it is therefore impossible to create a uniform security policy for most environments the main goal is always to preserve the confidentiality, integrity, accountability and availability of the devices and the data

Most environments will already have performed a risk assessment which dictated the security policy implementation for their existing IPv4 network. A similar risk assessment needs to be performed to ascertain how the current policy may need to be modified while migrating to IPv6.

In many situations, the security policy will remain largely the same but will have the following additional considerations:

- Since most devices will have multiple IPv6 addresses per interface, the policy must allow for effective filtering and auditing of these IPv6 addresses.
- IPv6 can offer more secure peer-to-peer communications since IPsec must be implemented in standards compliant implementations. Should any policy now mandate the use of end-to-end IPsec for authentication and integrity for all communication between IPv6 nodes? This would include not only the IPv6 clients and servers but also the dual-stack routers and tunnel broker devices.
- Confidentiality services may need to be revisited. If the filtering and auditing can be performed at the host level instead of within the network infrastructure then end-to-end confidentiality by using IPsec may become a feasible security policy mandate.
- While native IPv6 is the ultimate goal, it will be common to deploy some sort of transition mechanism. Any tunneling solution will create more security concerns due to the ease at which tunnel end points can be spoofed. Where to tunnel and whether to use static or dynamic tunnels will need to be determined.
- Firewall policies will need to be modified to accommodate IPv6 scenarios. More specific details can be found in the section ‘Network Infrastructure Security’ later in this document.
- IPv6 will rely more heavily on DNS and as such it may be more prudent to consider a policy which mandates the use of fully qualified domain names (FQDN) and DNS to locate users. This would require DHCPv6 (server and client) and Dynamic Updates to DNS.

Note that any effective security policy always needs to be technically feasible, operationally deployable and enforceable.

11.2. *End-Host Security*

End-host security in this section pertains to any client and server that is IPv6 capable. The main concerns from an end—host perspective is to ensure that:

- address assignment is performed in a reliable manner and cannot be spoofed
- traffic sourced from or destined to an end-host can be protected from modification, deletion or spoofing
- malicious behavior can be detected and subverted

Obtaining an address that is globally reachable requires policy decisions based on interactions between autoconfiguration, DHCPv6 and DNS. Many of the security concerns were highlighted in section 7 'Addressing Security Considerations'. While IPv6 has the capability to allow end-hosts to automatically configure their IPv6 link-local and global addresses in addition to getting all the information needed to communicate with the rest of the world, this capability will need to be deployed in a conscientious manner.

In some environments it is more important to obscure the IPv6 address and rely on dynamic DNS to provide the information necessary to obtain the IP address when needed. However, in many environments this may present a problem with end-to-end IPsec and it may be more prudent to instead look at effective auditing tools to determine potential malicious behavior instead of relying on address obscurity.

An effective addressing strategy will follow the following guidelines:

- Use EUI-based automatic configuration when the trust domains are such that there is a low probability that spoofing can occur, such as on a subnet where strict ingress/egress filtering is performed and all the end-nodes are effectively hardened.
- Use DHCPv6 to allocate addresses if there is a requirement to have control over the address use. This is typically necessary in larger deployments where end nodes are not always under strict control.
- Use standard but non-obvious static addresses for critical systems – it is best to standardize on short, fixed patterns for interfaces that should not be directly accessed from the outside to allow for a shorter filter list at the border routers. Make it difficult for potential intruder to guess addresses of important infrastructure devices.

At the time of this writing, there are no shipping implementations of SEND. However, once implementations become available, deploying SEND will provide added security and should be considered.

The devices themselves need to be hardened just like in any IPv4 environment today. The exception is that since IPsec should be available, it would be best to use IPsec ESP with NULL encryption to provide authentication and integrity services between all endpoint communications. Additionally, the following guidelines apply:

- Restrict access to the client or server to authenticated and authorized individuals
- Monitor and audit access to the client and server
- Turn off any unused services on the end node
- Use host firewall capabilities to control traffic that gets processed by upper layer protocols. End hosts can accept packets with a routing header extension and can also process routing headers and forward a packet. This can lead to circumventing security policies. Operating systems should not forward packets that include a routing header.
- Use virus scanners to detect malicious programs

11.3. *Network Infrastructure Security*

Network infrastructure security pertains to the components that make up the network infrastructure which includes the routers, switches, network firewalls, network intrusion detection systems as well as the network services such as DNS, AAA, Syslog, DNS, DHCP, SNMP and NTP.

11.3.1. Infrastructure Device Security

All of the network infrastructure devices, irregardless of whether it is a network services server, a tunnel broker, a firewall, an auditing system or a router, the device should be secured by following these guidelines:

- Restrict IPv6 access for telnet and ssh
- Restrict access to the device to authenticated and authorized individuals
- Monitor and regularly audit access to the device
- Turn off unused services on the device
- If applicable, use virus scanners to detect any malicious activity (this will mostly apply to servers that are providing network services such as DNS, DHCP, etc)
- Use IPsec ESP with NULL encryption to provide authentication and integrity services between communicating peers

Note that while it is true that some IPsec implementations are still cumbersome, that is an issue resolved by user demands. If the traffic needs to be inspected in transit then confidentiality can not be configured. However, if end-to-end communication between the infrastructure devices warrants confidentiality and there is no requirement to inspect the traffic in transit, then encryption should be deployed. This is a consideration for critical management traffic such as logging and auditing traffic.

In current IPv4 deployments most management traffic is restricted to an out-of-band (OOB) management infrastructure and rarely is any traffic encrypted. While current IPv6 implementations

still rely on IPv4 management (i.e. most devices have Syslog, NTP, SNMP, etc that use IPv4 transport and do not yet support IPv6), when IPv6 management becomes available it is recommended to look to IPsec to provide enhanced security services.

11.3.2. Routing Control Plane Security

Routing protocol communication is typically protected using a combination of MD5 authentication and filtering. MD5 authentication is used to validate the sending peer and to ensure that the data in transit has not been altered. Most router implementation support MD5 authentication for routing protocols for both IPv4 and IPv6. However, in many instances the implementations do not have a graceful mechanism to change the 'key' (i.e. password) which is an operational consideration. IPsec using IKE should be considered as an alternative option for authentication and integrity validation since IPsec key rollover is much more robust than in MD5. [9]

Ingress/egress traffic filtering at the periphery of inter-connected networks at varying trust boundaries should be deployed to reduce the effectiveness of source address spoofing denial of service attacks. Although BCP38 (RFC2827) and BCP84 (RFC3704) list examples and guidelines for IPv4 networks, the same principles should be applied to an IPv6 infrastructure. These standards recommend filtering ingress packets with obviously spoofed and/or 'reserved' source addresses. These include for IPv4:

- 0.0.0.0/8 (the system has no address assigned yet)
- 10.0.0.0/8 (private)
- 127.0.0.0/8 (loopback)
- 172.16.0.0/12 (private)
- 192.168.0.0/16 (private)

[9] If using MD5, as soon as the shared key is changed at either end, the existing connection is destroyed. When using IPsec, assuming that pre-shared key authentication is used for authenticating IKE peers, if the IKE authentication key is changed at one peer during the life of an IKE SA, that does not affect the SA. This is because the key used to protect the actual traffic (i.e. to provide the authentication and integrity protection) is the result of an ephemeral DH exchange, and is unique per SA. Of course implementations would need to be careful to avoid a pre-shared key change near the time that an IKE SA would timeout. Another advantage in using IPsec is that the SPI in the IPsec header identifies (among other things) the actual key used to protect the packet. During rekeying more than one SPI is accepted by the receiver; thus, both old and new keys can be accepted.

- 169.254.0.0/16 (IANA Assigned DHCP link-local)
- 224.0.0.0/4 (multicast)
- 240.0.0.0/4 (reserved and broadcast)

For IPv6 these would include:

- 0::/16 (compatible, mapped addresses, loopback, unspecified, ...)
- fe80::/10 (link-local)
- fec0::/10 (site-local)
- ff00::/8 (any multicast)
- in the case of 6to4 scenarios, equivalent 2002:v4addr::/48, where v4addr is any of the v4 addresses mentioned in the previous list

In the case of BGP routing, a variety of policies are deployed to limit the propagation of invalid routing information. Some recommended BGP route filtering policies can be found at http://www.space.net/~gert/RIPE/ipv6-filters.html.

Note that validating whether a legitimate peer has the authority to send the contents of the routing update is a difficult problem that needs yet to be resolved in both IPv4 and IPv6 environments.

11.3.3. Firewalls / Filtering

Filtering (i.e. looking at a specified set of parameters in a packet and making a decision to permit or deny the traffic) is an integral part of most network infrastructures. The following filtering rules are currently recommended for any network firewall where IPv6 traffic may be present:

- Filter internal-use IPv6 addresses at organization border routers – These include any site-local addresses and specific multicast addresses such as the all-routers address (FF01::2, FF02::2, FF05::2) or all-nodes address (FF01::1, FF02::1). These filtering rules should be in place as a safeguard for any potential mis-configurations or rogue devices. By logging the exceptions, any potential reconnaissance attack against these addresses can also lead to knowledge of a potential malicious intrusion. Note that it is always prudent to create a rule-set which is easier to maintain so it may be better to define filtering rules to permit what is needed and deny everything else.
- Filter ingress interfaces to deny traffic which contains spoofed traffic with the host portion of the IPv6 address.

- Filter unneeded services – services that are not used should be unreachable at border firewalls to eliminate any additional exploits.
- Allow only authorized tunneling endpoints on outbound firewall filters i.e. IP protocol 41 for 6to4 tunneling and UDP port 3544 for Teredo-based tunneling.

Selectively filter ICMP – IPv6 uses ICMP for neighbor discovery and Path MTU Discovery (PMTUD). It requires ICMPv6 neighbor discovery solicitations (NS) and neighbor discovery advertisements (NA) as well as router solicitation (RS) and router advertisement (RA) messages if autoconfiguration is used and RA messages are sent from the router for prefix lifetime advertisements.

Permit the following through the firewall:

- ICMPv6 type 1 code 0: no route to destination
- ICMPv6 type 2: packet too big (required for PMTUD)
- ICMPv6 type 3: time exceeded
- ICMPv6 type 4: parameter problem (informational when IPv6 node has problem identifying a field in the IPv6 header or in an extension header)
- ICMPv6 type 128: echo request
- ICMPv6 type 129: echo reply

Permit to and from the Firewall:

- ICMPv6 type 2: packet too big – firewall device is not allowed to fragment IPv6 packets going through it and must be able to generate this message for correct PMTUD behavior
- ICMPv6 type 4: parameter problem
- ICMPv6 type 130-132: multicast listener messages – in IPv6 a routing device must accept these messages to participate in multicast routing
- ICMPv6 type 133-134: router solicitation and advertisement – needed for IPv6 autoconfiguration
- ICMPv6 type 135-136: neighbor solicitation and advertisement – used for duplicate address detection and layer2-to-IPv6 address resolution

11.3.4. Logging / Auditing

A critical component to any network infrastructure is the logging and auditing of data traffic which can be used to detect and/or analyze successful security breaches. At this time most logging and auditing of IPv6 traffic is implemented using an IPv4 transport. However, when IPv6 transport becomes available, the following same practices should be used to effectively log and audit your dual-stack network infrastructures.

- Log routing protocol state changes, all device access (regardless of authentication success or failure), all commands issued to a device, all configuration changes and all router events (boot-ups/flaps)
- Logging filtered traffic should be performed on an exception basis (i.e. traffic which is NOT allowed is logged).
- The logged data should contain the source and destination IP addresses, layer 4 port numbers and a timestamp. The timestamp should be derived using NTP
- Use an OOB management network to transfer syslog data from device to syslog server if there is no confidentiality protection
- Use multiple syslog servers for varying infrastructure devices (i.e. one syslog server for backbone routers, one syslog server for customer edge routers, etc.)

11.3.5. IPv6 Security Deployment Summary

For every secure network, the goal is to protect electronic communication from malicious individuals who are determined to spoof, corrupt, alter or destroy the data or render critical services unavailable. Protection is required by every device that is participating in networked communication and all information that is either stored on a device or is in transit between communicating devices. The practices that are used to protect IPv4 networks today are similar to the ones that need to be deployed for IPv6 networks. The biggest difference is that IPsec should be considered more seriously to provide the necessary authentication, integrity and confidentiality services. Additionally, the logging and auditing portions which are now mostly available using IPv4 transport need to be made available using IPv6 transports to provide for better auditing capabilities.

12. Future Considerations

12.1. *Models for More Automated End-to-End Security*

A standardized way to distribute IA policy updates and threat signature would streamline the process of patching systems for known vulnerabilities before those vulnerabilities are turned into attacks. Currently CERT advisories do not include machine-readable attack signatures that could be quickly turned into firewall/IDS updates. Often known vulnerabilities are posted long before the vulnerability becomes part of an attack released “in the wild”. Even once patches/updates are

generated to prevent attacks, many systems are left un-patched and vulnerable so that streamlining the process to distribute firewall/IDS signatures would offer another line of defense to protect vulnerable hosts.

A cross-platform method to distribute IA policy and firewall rules would simplify the design of managed security systems.

12.2. *PKI Requirement and Analysis*

An automated mechanism to establish trust and verify identities of communicating parties would obviate the need of a public key infrastructure (PKI). At this time there still is not a wide deployment of digital certificates. It is expected that tools will become available that will make the deployment of IPsec using digital certificates and a PKI for authentication easier to use and configure. The biggest problem in setting up a PKI has been the initial enrollment process since at the end of the day you have to place trust in some entity to provide proof of identity. In scenarios where deployment of a PKI infrastructure has been successfully accomplished, the ultimate trust was based on an existing trust anchor process such as the one used to obtain company badges or initial access to network resources.

13. A Basic Framework For IPv6 Security

IPv6 provides a flexible framework for deploying new network services and adding new applications at a lower cost. Several advantages to deploying new services over IPv6 include the options of an end-to-end, peer to peer network and advances in imbedded IPsec, local service discovery, multihoming, and multicast addressing. Several enhanced network services such as mobility (MIPv6) and enhanced security (IPsec with existing protcol suites) have already been implemented on IPv6-capable devices and are ready for the initial IPv6 deployment. Additional protocols for enhanced IPsec security, secure neighbor discovery, improved multihoming support, additional multicast services, etc. are being built and deployed to increasingly leverage the basic IPv6 protocol suite. In order to take advantage of the new and emerging network security services based on IPv6, it is necessary to deploy network systems with enough of the basic IPv6 standard to support technology insertion of advanced security features and services.

The basic framework for IPv6 includes for all nodes:

- RFC 1981, Path MTU Discovery for IPv6
- RFC 2460, Internet Protocol v6 (IPv6) Specification
- RFC 2461, Neighbor Discovery for IPv6
- RFC 2462, IPv6 Stateless Address Auto-configuration
- RFC 2463, Internet Control Message Protocol (ICMPv6)

- RFC 4301, Security Architecture for the Internet Protocol (This RFC is the basic framework for a series of related RFCs for IPsec - see the IPsec section below)
- Support for one or more IPv6 link-layer specifications such as Ethernet [RFC 2464], PPP [RFC 2472], ATM [RFC 2492], etc...

The following added basic functionality is recommended for Host-specific IPv6 implementations:

- RFC 3484, Default Address Selection for IPv6
- RFC 3041, Privacy Extensions for Stateless Address Autoconfiguration in IPv6 (Optional)
- RFC 3315, Dynamic Host Configuration Protocol Version 6 for all workstation/PC/PDA hosts that require automatic configuration of Domain Name Service (DNS) and stateful assignment of IPv6 addresses. DHCPv6 provides stateful address assignment and can be adapted to include host authentication and control of advanced IP address policies such as privacy addresses.

The following added basic functionality is recommended for Router-specific IPv6 implementations:

- Routing protocols such as OSPFv3 [RFC 2740]
- RIPNG [RFC 2080]
- BGP Multiprotocol Extensions for IPv6 [RFC 2545]

The IPsec security protocols that are recommended are:

- RFC 4301, Security Architecture for the Internet Protocol: All end nodes and intermediate nodes should deploy now with at least a minimum IPsec suite consisting of the IPsec Encapsulating Security Payload (ESP) with 3DESCBC/AES128CBC/SHA1 transforms as defined in the following RFCs:
- RFC 4301, Security Architecture for the Internet Protocol
- RFC 4303, IP Encapsulating Security Payload (ESP)
- RFC 4305, (ESP and AH) Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)
- RFC 4308, Cryptographic Suites for IPsec
- RFC4309, Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)

For IPsec key management, IPv6 nodes will support manual setup of security associations and keys. Though currently optional, automatic key exchange is necessary to make IPsec scalable and

practical in any large scale enterprise network. Automated key exchange and security association management as currently deployed on most IPv6 implementations is defined in:

- RFC 2407, The Internet IP Security Domain of Interpretation for ISAKMP
- RFC 2408, Internet Security Association and Key Management Protocol
- RFC 2409, The Internet Key Exchange (IKE)
- RFC 4109, Algorithms for Internet Key Exchange Version 1 (IKEv1)

The recommendations for Advanced IPv6 Security include a more robust IPsec suite:

All IPv6 IPsec implementations, VPNs, and IPsec-based network encrypters, should have an upgrade path to IPsec Suite "VPN-B" (defined in RFC 4308 section 2.2) with automated key exchange and security association management as defined in Internet Key Exchange version 2 (IKEv2) or begin migration to this suite which is expected to be the common minimum IPsec security suite required in a few years. Advanced IPv6 network services like Mobile IPv6 (MIPv6) expect to use components of this suite to secure remote communications. This suite provides improved automated security management and key exchange protocols and additional strong encryption and data integrity algorithms as shown below.

IPsec:

Protocol	ESP [RFC 4303]
ESP encryption	AES with 128-bit keys in CBC mode [AES-CBC]
ESP integrity	AES-XCBC-MAC-96 [AES-XCBC-MAC]

IKEv2 Security Management:

Encryption	AES with 128-bit keys in CBC mode [AES-CBC]
Pseudo-random function	AES-XCBC-PRF-128 [AES-XCBC-PRF-128]
Integrity	AES-XCBC-MAC-96 [AES-XCBC-MAC]
Diffie-Hellman group	MODP 2048-bit [RFC3526]

The CREATE_CHILD_SA (for IKEv2) must be supported by both parties in this suite. The initiator of this exchange may include a new Diffie-Hellman key; if it is included, it must be of type 2048-bit MODP. If the initiator of the exchange includes a Diffie-Hellman key, the responder must include a Diffie-Hellman key, and it must be of type 2048-bit MODP.

SEcurity Neighbor Discovery - (SEND)

SEND is an emerging IPv6 service that leverages basic IPv6 features to establish initial trust

relationships between network nodes. Hosts on the same link use Secure Neighbor Discovery (SEND) to securely discover each other's presence and link-layer addresses, to find routers, and to maintain reachability information about the paths to active neighbors. SEND defines mechanisms in addition to IPsec that may be used to discover and authenticate trusted neighbors. SEND for both hosts and routers is defined in:

- RFC 3971 Secure Neighbor Discovery (SEND)
- RFC 3972 Cryptographic Generated Addresses

Many RFCs are specific to the function of various classes of IT equipment. Required vs. optional (must vs should) protocol components depend upon an Enterprise's deployment policy.

14. Acknowledgements

The author wishes to acknowledge David Green for the overall development of this paper as well as the technical contributions relating specifically to distributed firewalls and the basic framework requirements. Also, thanks to Gene Cronk and Joseph Klein for the list of IPv6 Capable Network Assessment Tools listed in Appendix A and to the following reviewers for their helpful comments: Yannick Pouffary, Joseph Klein and John Spence.

15. NAv6TF Disclaimer

Data and information is provided for informational purposes only, and is not intended for business purposes. Neither IPv6 Forum/NAv6TF or its affiliates nor any of its data or content providers shall be liable for any errors in the content, or for any actions taken in reliance thereon. IPv6 Forum/NAv6TF shall not be liable for any damages or costs of any type arising out of or in any way connected with your use of the content published herein.

16. About NAv6TF

The North American IPv6 Task Force (NAv6TF) www.nav6tf.org is a sub-chapter of the IPv6 Forum www.ipv6forum.org dedicated to the advancement and propagation of IPv6 (Internet Protocol, version 6) in the North American continent. Comprised of individual members, rather than corporate sponsors, the NAv6TF mission is to provide technical leadership and innovative thought for the successful integration of IPv6 into all facets of networking and telecommunications infrastructure, present and future.

Through its continued facilitation of technical and business case whitepapers, IPv6-centric conferences, IPv6 test and interoperability events, IPv6 deployment readiness guides, and collaboration with IPv6 task forces from around the globe, the NAv6TF will strive to be the guiding force for IPv6 adoption and readiness in the U.S. and Canada.

17. About the Author

Merike Kaeo is the Founder of Double Shot Security www.doubleshotsecurity.com a technical strategy and business consulting firm concentrating on education, analysis and design of secure IPv4 and IPv6 network infrastructures. Author of "Designing Network Security," published by Cisco Press, she is a frequent speaker on security issues and solutions at global security-related conferences and ISP forums.

18. Appendix A – IPv6 Capable Network Assessment Tools

This appendix addresses what tools are available for doing network assessment and penetration for IPv6, what tools are used in IPv4 networks to do this job, and ways to make IPv4 only tools attack and assess IPv6 networks. It is laid out in three areas:

The Good: Tools that natively support IPv6.

The Bad: Tools that do not support IPv6 at all, but are used in many IPv4 testing scenarios.

The Ugly: Tools that either can be used on IPv6 networks via a transitioning mechanism, or only have partial support for IPv6.

Many of these tools are either on the Top 75 tools list at http://www.insecure.org, or were recommended by security professionals that work in the field.

The Good:

Name: Argus the All Seeing
Description: A system/network monitoring application. It p resents a nice clean, easy to view web interface that will keep both the managers and techs happy. Can send alerts numerous ways (such as via pager).
License: Perl Artistic License
Platforms: *NIX
Availability: http://argus.tcp4me.com/download.html
Current Version: 3.4

Name: LSOF (LiSt Open Files)
Description: This Unix specific diagnostic and forensics tool lists information about any files that are open by processes currently running on the system.
License: F/OSS
Platforms: *NIX

Name: LSOF (LiSt Open Files)
Availability: ftp://vic.cc.purdue.edu/pub/tools/unix/lsof
Current Version: 4.77

Name: Snoop
Description: Network sniffer for Solaris similar to TCPDump, Snoop listens for all traffic on a specific interface. Available in Solaris since 8.
License: Sun Software License
Platforms: Sun Solaris
Availability: http://www.sun.com/software/solaris
Current Version: n/a

Name: DIG DNS Query Tool
Description: A handy DNS query tool that comes free with BIND. Available in BIND DNS since 8.3.
License: F/OSS
Platforms: Windows, *NIX
Availability: http://www.isc.org
Current Version: n/a

Name: Etherape
Description: A graphical network monitor for Unix modeled after etherman. Featuring link layer, ip and TCP modes, it displays network activity graphically. Hosts and links change in size with traffic. Color coded protocols display.
License: GPL
Platforms: *NIX
Availability: http://etherape.sourceforge.net
Current Version: 0.9.6

Name: Ethereal
Description: Used by network professionals around the world for troubleshooting, analysis, software and protocol development, and education. It has all of the standard features you would expect in a protocol analyzer, and several features not seen in any other product.
License: GPL

Name: Ethereal
Platforms: Windows, *NIX
Availability: http://ethereal.com
Current Version: 0.99.0

Name: Fping
Description: Parallel ICMP scanner. Can ping multiple hosts from command line or text file. Great for scripting.
License: F/OSS
Platforms: *NIX
Availability: http://www.fping.com
Current Version: 2.4b2

Name: LibNet
Description: High level network API. Allows an application programmer to construct and inject network packets.
License: F/OSS
Platforms: *NIX
Availability: http://www.packetfactory.net/libnet
Current Version: 1.1.2.1 (Stable) 1.1.3 (Beta)

Name: NTOP
Description: Web based traffic probe. Users access a web page of an NTOP server to get graphical visualizations of network use and abuse.
License: GPL
Platforms: *NIX
Availability: http://www.ntop.org
Current Version: 3.2

Name: PF
Description: Packet filter originally included with OpenBSD, ported to FreeBSD. Comes with FreeBSD 5.xx and OpenBSD 3.xx
License: BSD

Name: PF
Platforms: BSD Platforms
Availability: http://www.openbsd.org and http://www.freebsd.org
Current Version: n/a

Name: SendIP
Description: Command line tool for sending arbitrary IP packets. Command line options to specify the content of every header of a NTP, BGP, RIP, RIPng, TCP, UDP, ICMP or raw IPv4 and IPv6 packets.
License: GPL
Platforms: *NIX
Availability: http://www.earth.li/projectpurple/progs/sendip.html
Current Version: 2.5

Name: TCPDump/WinDump
Description: Classic tool for network monitoring and data aquisition.
License: BSD
Platforms: Windows, *NIX
Availability: http://www.tcpdump.org or http://www.winpcap.org
Current Version: 3.9.4 (*NIX) 3.1 (Windows)

Name: IP6SIC
Description: IPv6 Stack integrity checker.
License: BSD
Platforms: *NIX
Availability: http://cvs.sourceforge.net/viewcvs.py/ip6sic/ip6sic/
Current Version: 0.1

Name: Ngrep
Description: Network Grep strives to provide most of GNU Greps' features over the network layer. IPv6 support must be compiled into libpcap.
License: F/OSS

Name: Ngrep
Platforms: *NIX
Availability: http://ngrep.sourceforge.net/
Current Version: 1.44

Name: THC Amap
Description: Application written by The Hacker's Choice for application fingerprinting.
License: GPL
Platforms: *NIX
Availability: http://www.thc.org
Current Version: 5.2

Name: THC-IPV6
Description: A complete tool set to attack the inherent protocol weaknesses of IPV6 and ICMP6, and includes an easy to use packet factory library.
License: GPL
Platforms: *NIX
Availability: http://thc.org/thc-ipv6/
Current Version: 0.6

Name: SinFP (Perl module Net::SinFP)
Description: SinFP is a new approach to OS fingerprinting, which bypasses limitations that NMAP has.
License: F/OSS
Platforms: Perl
Availability: http://www.gomor.org/cgi-bin/index.pl?mode=view;page=sinfp#8
Current Version: n/a

Name: STunnel
Description: A general purpose SSL cryptographic wrapper. IPv6 enabled with ./configure –enable-ipv6.
License: GPL

Name: STunnel
Platforms: Windows, *NIX
Availability: http://www.stunnel.org
Current Version: 4.15

Name: 4to6 DDoS
Description: 4to6ddos is a distributed denial of service against ipv6 that works without installing ipv6 support. It shoots ipv6 encapsulated in ipv4 packets directly to the ipv4-to-ipv6 tunnels.
License: n/a
Platforms: *NIX
Availability: http://www.pkcrew.org
Current Version: n/a

Name: v6scan.c
Description: V6scan is an ipv6 port scanner. Checks 14 different tcp ports which are commonly used by attackers.
License: n/a
Platforms: *NIX
Availability: http://packetstormsecurity.org/UNIX/misc/v6scan.c
Current Version: n/a

Name: LandIPv6.c
Description: Microsoft Windows XP/2003 ipv6 remote denial of service exploit.
License: n/a
Platforms: *NIX
Availability: http://packetstormsecurity.org/0505-exploits/LandIpV6.c
Current Version: n/a

Name: cb4n6.c
Description: This is an IPv6 banner grabber by c1zc0 Security.
License: n/a

Name: cb4n6.c
Platforms: *NIX
Availability: http://c1zc0.com
Current Version: n/a

Name: pmacct
Description: A small set of passive network monitoring tools to measure,account and aggregate IPv4 and IPv6 traffic; aggregation revolves around the key concept of primitives (VLAN id, source and destination MAC addresses, hosts, networks, AS numbers, ports, IP protocol and ToS/DSCP field are supported) which may be arbitrarily combined to build custom aggregation methods; support for historical data breakdown, triggers and packet tagging, filtering and sampling. Aggregates can be stored into memory tables, SQL databases (MySQL or PostgreSQL) or simply printed to stdout. Data is collected from the network either using libpcap (and optionally promiscuous mode) or reading NetFlow v1/v5/v7/v8/v9 and sFlow v2/v4/v5 datagrams, both unicast and multicast.
License: n/a
Platforms: *NIX
Availability: http://www.ba.cnr.it/~paolo/pmacct/
Current Version: n/a

Name: ASB
Description: Advanced Socket Bouncer (ASB) is another kind of network tool. It supports ipv6 (detects automatically ipv6 hostnames/addresses), SQUID (connect method and SQUID with SSL support but no SSL proxy), SOCKS4, SOCKS5, and WINGATE.
License: n/a
Platforms: *NIX
Availability: http://wildandi.void.at
Current Version: 0.1

Name: NFR
Description: Now supports the detection of a wide variety of application vulnerable across IPv6 and IPv6 specific vulnerabilities.
License: Commercial
Platforms: Windows, *NIX
Availability: http://www.nfr.com
Current Version: Several tools.

Name: Symantec NetReconT
Description: Scans for eight well defined IPv6 vulnerabilities. It does not scan application vulnerabilities over IPv6.
License: Commercial
Platforms: Windows
Availability: http://www.symantec.com
Current Version: 3.6

Name:puTTY
Description: An excellent GUI SSH client. Can be compiled for many platforms.
License: MIT
Platforms: Windows, *NIX
Availability: http://www.chiark.greenend.org.uk/~sgtatham/putty/
Current Version: 0.58

The Bad:

Name: CheopsNG
Description: Cheops-ng is a Network management tool for mapping and monitoring your network. It has host/network discovery functionality as well as OS detection of hosts. Cheops-ng has the ability to probe hosts to see what services they are running. The GUI does not support IPv6 addresses.
License: GPL
Platforms: *NIX
Availability: http://cheops-ng.sourceforge.net/
Current Version: 0.2.3

Name: EttercapNG
Description: Ettercap is a suite for man in the middle attacks on LAN. It features sniffing of live connections, content filtering on the fly and many other interesting tricks. IPv6 support discussed in forums, but not implemented.
License:
Platforms: Windows, *NIX
Availability: http://ettercap.sourceforge.net
Current Version: 0.7.3

Name: Firewalk
Description: Active reconnaissance network security tool that attempts to determine what layer 4 protocols an IP forwarding device will pass. All libraries IPv6 aware. Last update: 07/2003.
License: BSD
Platforms: *NIX
Availability: http://www.packetfactory.net/projects/firewalk
Current Version: 5.0

Name: DSniff
Description: dsniff is a collection of tools for network auditing and penetration testing. dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.). arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching). sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKI. Last update: 05/2002.
License: F/OSS
Platforms: *NIX
Availability: http://www.monkey.org/~dugsong/dsniff/
Current Version: 2.4b1

Name:TCPReply
Description: A suite of tools written by Aaron Turner which gives you the ability to use previously captured traffic in libpcap format to test a variety of network devices. IPv6 Planned in a post 3.0 release. Last release 09/2004.
License: BSD
Platforms: *NIX
Availability: http://tcpreplay.synfin.net/trac/
Current Version: 3.0 Beta 7

Name: FPort
Description: Foundstone's enhanced netstat. Last release 05/2001
License: Freeware (no sourcecode)
Platforms: Windows
Availability: http://www.foundstone.com

Name: FPort
Current Version: 2.0

Name: Fragroute
Description: Intercepts and rewrites egress traffic, implementing many intrusion detection evasion attacks. Last release 04/2002.
License: BSD
Platforms: *NIX
Availability: http://www.monkey.org/~dugsong/fragroute
Current Version: 1.2

Name: GFI LANGuard N.S.S.
Description: Checks your network for all potential methods that a hacker might use to attack it. By analyzing the operating system and the applications running on your network, GFI LANguard N.S.S. identifies possible security holes.
License: Commercial
Platforms: Windows
Availability: http://www.gfi.com/lannetscan/
Current Version: 7

Name: Hunt
Description: An advanced packet sniffing and connection intrusion tool for Linux. Developed on a Linux 2.2.xx kernel. Last update: 05/2000.
License: GPL
Platforms: *NIX
Availability: http://www.mirrors.wiretapped.net/security/packet-capture/hunt/
Current Version: 1.5

Name: IPTraf
Description: P network monitoring software based on Ncurses. No support for IPv6, only for raw sockets and IPv4.
License: GPL
Platforms: Linux
Availability: http://iptraf.seul.org/
Current Version: 3.0.0

Name: ISS Internet Scanner
Description: Application level vulnerability assessment scanner. No IPv6 capabilities.
License: Commercial
Platforms: Windows
Availability: http://www.iss.net/products
Current Version: 7.2.4

Name: NBTScan
Description: A program for scanning IP networks for NetBIOS name information. It sends NetBIOS status query to each address in supplied range and lists received information in human readable form. NetBIOS over IPv6 not enabled on any Microsoft Oses (except Vista). Last update: 06/2003.
License: F/OSS
Platforms: Windows, *NIX
Availability: http://www.inetcat.org/software/nbtscan.html
Current Version: 1.5.1

Name: Nessus
Description: The world's most popular vulnerability scanner used in over 75,000 organizations world-wide.
License: Dual (3.0+ Commercial, 2.2 and below GPL)
Platforms: Windows, *NIX
Availability: http://www.nessus.org
Current Version: 3.0.3

Name: Paketto Keiretsu
Description: A tool for stretching TCP/IP networks and protocols beyond what they were intended for. Because of the packet manipulation at a raw level and the header differences of v4 and v6, would take almost an entire rewrite to port to IPv6.
License: GPL
Platforms: *NIX
Availability: http://www.doxpara.com
Current Version: 2.00pre5

Name: Retina
Description: A flexible vulnerability scanner, similar to Nessus and ISS Internet Scanner. No IPv6 Support from Eeye
License: Commercial
Platforms: Windows
Availability: http://www.eeye.com
Current Version: n/a

Name: SARA -- Security Auditor's Research Assistant
Description: A security assessment tool derived from the infamous SATAN scanner.
License: F/OSS
Platforms: Windows, *NIX
Availability: http://www-arc.com/sara
Current Version: 6.0.7e

Name: Shadow Security Scanner
Description: A commercial vulnerability assessment tool.
License: Commercial
Platforms: Windows
Availability: http://www.safety-lab.com/en/products/securityscanner.htm
Current Version: n/a

Name: Solar Winds Toolsets
Description: A plethora of network discovery, monitoring and attack tools. Dozens of special purpose tools targeted at systems administrators. No IPv6 support.
License: Commercial
Platforms: Windows
Availability: http://www.solarwinds.net
Current Version: Several different programs

Name: TCPTraceRoute
Description: A traceroute implementation using TCP packets. No IPv6 support, but IPv6 is supported in the libraries it

Name: TCPTraceRoute
uses.
License: GPL
Platforms: *NIX
Availability: http://michael.toren.net/code/tcptraceroute/
Current Version: 1.5 Beta 7

Name: VisualRoute
Description: Helps determine if a connectivity problem is due to an ISP, the Internet, or the web site you -- or your customers -- are trying to reach, and pinpoints the network where a problem occurs.
License: Commercial
Platforms: *NIX, Windows
Availability: http://www.visualroute.com/index.html
Current Version: 2006

Name: Winfingerprint
Description: Win32 Host/Network Enumeration Scanner. Winfingerprint is capable of performing SMB, TCP, UDP, ICMP, RPC, and SNMP scans. No IPv6 support.
License: GPL
Platforms: Windows
Availability: http://winfingerprint.sourceforge.net
Current Version: 0.6.2

Name: XProbe2
Description: A tool for determining the OS of a remote host. It uses the same techniques of NMAP as well as a few others. Emphasizes ICMP as the fingerprinting approach. No IPv6 support.
License: GPL
Platforms: *NIX
Availability: http://www.sys-security.com/index.php?page=xprobe
Current Version: 0.3

Name: Zone Alarm
Description: Personal firewall software for Windows. Asks to block an IPv6 query, then doesn't.
License: Commercial
Platforms: Windows
Availability: http://www.zonelabs.com
Current Version: 6.5

The Ugly:

Name: THC Hydra
Description: Parallelized network authentication cracker for FTP, POP3, IMAP, NBT, Telnet, HTTP, LDAP, NTP, VNC, ICQ, SOCKS and more. Includes SSL support. IPv6 enabled on Windows, all others could be SSH tunnelled.
License: GPL
Platforms: Windows, *NIX
Availability: http://www.thc.org
Current Version: 5.3

Name: Whisker/LibWhisker/Achilles/Nikto/NStealth/Spike Proxy/Brutus
Description: Web proxy and attack tools. These are all IPv4 only, but could easily be tunnelled via SSH, proxies or other transitioning mechanisms.
License: varies per tool
Platforms: varies per tool
Availability: varies per tool
Current Version: varies per tool

Name: NetCat
Description: A simple utility which reads/writes data across network connections using TCP or UDP. AKA "The Hacker's Swiss Army Knife". Netcat6 is a separate project.
License: GPL
Platforms: *NIX
Availability:http://netcat.sourceforge.net/ (IPv4 only) http://www.deepspace6.net/projects/netcat6.html(IPv4 & IPv6)
Current Version: 0.7.1

Name: SAINT (Security Auditor's Integrated Network Tool)
Description: A tool much like Nessus or eEye Retina designed exclusively for UNIX. IPv6 support in Linux only.
License: Commercial
Platforms: *NIX
Availability: http://www.saintcorporation.com
Current Version: 5.9.6

Name: Netstumbler
Description: A tool for Windows that allows you to detect Wireless Local Area Networks (WLANs) using 802.11a/b/g. Mainly a layer 2 tool, but only detects IPv4 addresses.
License: Freeware
Platforms: Windows
Availability: http://www.netstumbler.com
Current Version: 0.4.0

Name: Kismet
Description: An 802.11 layer 2 wireless network detector, sniffer, and intrusion detection system. Kismet will work with any wireless card which supports raw monitoring mode, and can sniff 802.11 a/b/g traffic. Mainly a layer 2 tool, but only detects IPv4 addresses.
License: GPL
Platforms: *NIX, Windows
Availability: http://www.kismetwireless.net
Current Version: 2006-04-R1

Name:Net Filter
Description: The current Linux packet filter/firewall. Iptables userspace command is used for configuration. Supports packet filtering and NAT. IP6Tables only supports stateless firewalling on 2.6.11 or older kernels.
License: GPL
Platforms: Linux
Availability: http://www.netfilter.org
Current Version: 1.3.5

Name: TCP Wrappers
Description: IP based access control and logging mechanism. Many default installs do not include IPv6 support.
License: F/OSS
Platforms: *NIX
Availability: ftp://ftp.cerias.purdue.edu/pub/tools/unix/netutils/
Current Version: 7.6

Name: Sam Spade
Description: GUI for many handy network tasks including nslookup, dig, whois, ping, traceroute, raw HTTP, DNS zone transfer, website searching and SMTP relay checks. Doesn't support IPv6 natively, but many tools could be used via transitioning mechanisms.
License: Freeware
Platforms: Windows
Availability: http://www.samspade.org
Current Version: 1.14

Name: NMAP (Network MAPper)
Description: An open source utility for network exploration or security auditing. It uses raw IP packets in novel ways to determine what hosts are available on a given network. "6" option enables IPv6 support. Only supports ping scan, TCP scan and TCP connect scan. An alternative (but older) patched version does other scan types. It requires NMAP 2.54Beta36 and patches from http://nmap6.sourceforge.net Does not do network scanning (for obvious reasons).
License: GPL
Platforms: *NIX
Availability: http://www.insecure.org
Current Version: 4.03

Name: Cain & Abel
Description: A free password recovery tool for Windows. Allows easy recovery of passwords by network sniffing, revealing password boxes, uncovering cached passwords and analyzing routing protocols. Local password cracking works fine. No IPv6 support otherwise.
License: Freeware
Platforms: Windows

Name: Cain & Abel
Availability: http://www.oxid.it
Current Version: 2.9

Name: Hping2(3)
Description: Assembles and sends custom ICMP/UDP/TCP packets and displays any replies. Hping 2 and 3 do not support IPv6. There are patches available for a beta version of Hping 2.
License: GPL
Platforms: *NIX
Availability: http://www.hping.org
Current Version: 3

Name: HoneyD
Description: A small daemon that creates virtual hosts on a network, running arbitrary services. TCP signatures can appear to be running different Oses and services. While HoneyD supports IPv6, no F/OSS NIDS for *Nix currently fully supports decoding IPv6 packets.
License: GPL
Platforms: *NIX
Availability: http://www.honeyd.org
Current Version: 1.5a

Name: Snort
Description: De facto standard F/OSS NIDS. Many commercial products are based on Snort. It has partial IPv6 support in the latest release.
License: GPL
Platforms: *NIX, Windows
Availability: http://www.snort.org
Current Version: 2.6.0

Name: GNUPG (GNU Privacy Guard)
Description: A GNU tool for encrypting and decrypting files and communications, based on Phil Zimmerman's PGP standard. Patches available for IPv6 support.
License: GPL

Name: GNUPG (GNU Privacy Guard)
Platforms: Windows, *NIX, others
Availability: http://www.gnupg.org
Current Version: 1.4.3

19. References

[I-D.ietf-mip6-cn-ipsec] Dupont, F. and J. Combes, "Using IPsec between Mobile and Correspondent IPv6 Nodes", draft-ietf-mip6-cn-ipsec-02 (work in progress), December 2005.

[I-D.ietf-mip6-ikev2-ipsec] Dupont, F. and V. Devarapalli, "Mobile IPv6 Operation with IKEv2 and the revised IPsec Architecture", draft-ietf-mip6-ikev2-ipsec-06 (work in progress), April 2006.

[I-D.ietf-v6ops-icmpv6-filtering-recs] Davies, E. and J. Mohacsi, "Recommendatioons for Filtering ICMPv6 Messages in Firewalls", draft-ietf-v6ops-icmpv6-filtering-recs-02 (work in progress), July 2006.

[I-D.ietf-v6ops-ipsec-tunnels] Savola, P., "Using IPsec to Secure IPv6-in-IPv4 Tunnels", draft-ietf-v6ops-ipsec-tunnels-02 (work in progress), March 2006.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC2367] McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2", RFC 2367, July 1998.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[RFC2404] Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.

[RFC2405] Madson, C. and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405, November 1998.

[RFC2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[RFC2408] Maughan, D., Schneider, M., and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC2410] Glenn, R. and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec", RFC 2410, November 1998.

[RFC2411] Thayer, R., Doraswamy, N., and R. Glenn, "IP Security Document Roadmap", RFC 2411, November 1998.

[RFC2412] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.

[RFC2431] Tynan, D., "RTP Payload Format for BT.656 Video Encoding", RFC 2431, October 1998.

[RFC2451] Pereira, R. and R. Adams, "The ESP CBC-Mode Cipher Algorithms", RFC 2451, November 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[RFC2463] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, December 1998.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC3041] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January 2001.

[RFC3053] Durand, A., Fasano, P., Guardini, I., and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.

[RFC3152] Bush, R., "Delegation of IP6.ARPA", BCP 49, RFC 3152, August 2001.

[RFC3177] IAB and IESG, "IAB/IESG Recommendations on IPv6 Address Allocations to Sites", RFC 3177, September 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3363] Bush, R., Durand, A., Fink, B., Gudmundsson, O., and T. Hain, "Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)", RFC 3363, August 2002.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC3542] Stevens, W., Thomas, M., Nordmark, E., and T. Jinmei, "Advanced Sockets Application Program Interface (API) for IPv6", RFC 3542, May 2003.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

[RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.

[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[RFC3776] Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", RFC 3776, June 2004.

[RFC3964] Savola, P. and C. Patel, "Security Considerations for 6to4", RFC 3964, December 2004.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "Secure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC4109] Hoffman, P., "Algorithms for Internet Key Exchange version 1 (IKEv1)", RFC 4109, May 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC4214] Templin, F., Gleeson, T., Talwar, M., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 4214, October 2005.

[RFC4285] Patel, A., Leung, K., Khalil, M., Akhtar, H., and K. Chowdhury, "Authentication Protocol for Mobile IPv6", RFC 4285, January 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

[RFC4449] Perkins, C., "Securing Mobile IPv6 Route Optimization Using a Static Shared Key", RFC 4449, June 2006.

[RFC4472] Durand, A., Ihren, J., and P. Savola, "Operational Considerations and Issues with IPv6 DNS", RFC 4472, April 2006.

[RFC4487] Le, F., Faccin, S., Patil, B., and H. Tschofenig, "Mobile IPv6 and Firewalls: Problem Statement", RFC 4487, May 2006.

[RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, June 2006.

http://www.cisco.com/security_services/ciag/documents/v6-v4-threats.pdf

http://csrc.nist.gov/publications/drafts/DRAFT-SP800-81.pdf

http://www.ipv6style.jp/en/tech/20050704/index.shtml

http://www.ipv6style.jp/en/tech/20050808/index.shtml