An Overview of Security in Next-Generation IP (IPv6)

Randall Atkinson <atkinson@itd.nrl.navy.mil>

IPng Design Review, 22 August 1994

Outline

- Design Goals
- Authentication Header
- Encapsulating Security Payload
- Key Management
- Firewalls
- Status
- Application-related Issues
- Other Issues

Security Design Goals

- Provide authentication and integrity protections without confidentiality so that export rules don't interfere.
- Provide a standard encryption mechanism to provide confidentiality.
- Ensure that the encryption mechanism can encrypt either entire IP datagrams or just transport protocol frames.
- Keep all mechanisms algorithm-independent so that both military and commercial algorithms can be used and also so that we can substitute improved algorithms at a later date without changing the protocol.
- Keep key management decoupled from the rest so that key mgmt protocol vulnerabilities discovered later will not require the other mechanisms to be revised.
- Retain the ability to use firewalls (much though we might not like them).

IPv6 Authentication Header

- Intended to provide optional authentication and integrity to all IPv6 users.
- Provides authentication and integrity for all invariant fields, without providing confidentiality.
- Appears to be widely exportable and useable because it does not provide confidentiality.
- Addresses a large subset of current Internet vulnerabilities (e.g. source routing attacks).
- Proposes use of keyed MD5 as the preferred Internet algorithm.
- MD5 performance is an open issue, suggestions on alternative algorithms are desired.
- Uses Security Association Identifier as the only coupling with the key mgmt protocol/ mechanism.
- Receiver-oriented SAIDs support IP multicasting.

Authentication Header Format

- Proposed format below was discussed at the IPng Implementer's Meeting in Toronto.
- Header Format is stable, but might change if problems are discovered.
- IP Security WG is considering adopting a variant of this for use with IPv4.

Next Header	Header Length	(Reserved, must send as zero)
Security Association Identifier (SAID), 32 bits long		
Authentication Data (length varies with authentication algorithm, 128 bits for MD5)		

Authentication Header bit format

Encapsulating Security Payload

- Intended to provide optional confidentiality via encryption.
- Universal implementation and deployment unlikely due to legal issues.
- Many countries regulate export and/or use of encryption by private firms and individuals.
- Can encrypt either an entire IPv6 datagram or only TCP/UDP.
- Proposes use of DES in CBC mode as standard Internet confidentiality algorithm.
- Uses Security Association Identifier as the only coupling to the key mgmt protocol/ mechanism.
- Receiver-oriented SAIDs support IP multicasting.

ESP Format

- This format was discussed in detail at the IPng Implementer's Meeting in Toronto
- This format is being considered for use with both IPv4 and IPv6.
- Most discussions on this design are occurring within the IP Security WG.'
- The design is subject to change

<table>
<tr><td colspan="3">Security Association Identifier or SAID (32 bits)</td></tr>
<tr><td colspan="3">Algorithm-dependent Initialisation Vector, Crypto Sync, etc. (variable length)</td></tr>
<tr><td>Next Header (8 bits)</td><td>Length (8 bits)</td><td>Reserved, sent as zero (16 bits)</td></tr>
<tr><td colspan="2">Protected Data (either entire IPv6 datagram or just a sequence of zero or more IPv6 optional headers and TCP or UDP user data). The Next Header field must always start on a 32-bit boundary.</td><td>Algorithm-dependent Trailer</td></tr>
</table>

Proposed Encapsulating Security Payload bit format

Key Management

- Outside the scope of the IPng effort, but an essential security component.
- Is the difficult and missing piece needed for real Internet security.
- Both Authentication Header and Encapsulating Secuirty Payloaduse common 32-bit SAID as the connection with future key management protocol and other mechanisms (e.g. manual key distribution).
- Jeff Schiller, Security Area Director, has asked Steve Bellovin of AT&T Bell Laboratories to head up a new IETF Key Management WG.
- This new WG would be responsible for specifying a key management protocol suitable for use with not only the IPv6 Authentication Header and IPv6 Encapsulating Security Payload, but also with the IPv4 Security Protocol (once defined), and other IETF protocols (e.g. OSPF authentication).

Firewalls

- While many view them as undesirable, firewalls are unlikely to disappear in the future.
- IPv6 continues to permit application-based packet filters, but identifying the transport protocol and transport protocol port number takes more effort because IPv6 headers are daisy-chained.
- Authentication Header might be useful in increasing assurance in firewalls.
- IPv6 Encapsulating Security Payload will make encrypting gateways and encrypting firewalls more practical.

Status

- Experimental implementations of the SIPP Authentication Header exist.
- IPv6 Authentication Header implementations are planned (differences are slight).
- Interoperability testing for the Authentication Header will be part of general IPv6 interoperability testing, to be held after IPv6 implementations are farther along.
- Implementations of the IPv6 Encapsulating Security Payload are planned.
- Key management is not yet defined, but is actively being worked within the IETF.
- Efforts are underway to converge this IPv6 work with recent proposals for IPv4 arising from the IETF's IPv4 Security Protocol WG. Some convergence appears likely, but most details are currently unresolved.

Application-related Issues

- How should an application request specific security services ?
- Which APIs should be used to access security features (GSS-API, Sockets, XTI/TLI, all of these) ?
- What behaviour should the application see when an inauthentic packet arrives ?
- What behaviour should the application see when authentication was expected but is missing ?
- If authentication is computationally expensive, how can we selectively protect important messages (e.g ICMP redirect, source-routed packets, TCP FIN and TCP FIN ACK) ?
- If a non-authenticated source-routed packet arrives, should the kernel decide whether to invert the source route or should the application decide ?

Other Issues

- Which confidentiality algorithms should be Internet standard(s) -- IDEA, DES, or another ?
- Which authentication algorithms should be Internet standard(s) -- MD5, SHA, or another ?
- Are those algorithms fast enough and strong enough for normal Internet use ?
- Will vendors really implement and ship the Authentication Header in every box ?
- What legal restrictions might affect widespread deployment of a key management protocol ?
- Which, if any, new ICMP messages are needed (e.g. to indicate that authentication or confidentiality is required but was missing) ?
- What should be done about IP security labelling ?
- Should there be an Informational RFC giving guidance on auditing of security-related protocol events (e.g. what should the OS do when a received packet fails authentication ?) ?