RUTS Security

Steven M. Bellovin

smb@research.att.com

http://www.research.att.com/˜smb

973-360-8656

AT&T Labs Research

Florham Park, NJ 07932

Sequence Numbers

- TCP's sequence numbers (with RFC 1948) are adequate for low-security work if there is no eavesdropping.
- A larger sequence number space is desirable, though still not a solution to eavesdropping.
- Many solutions require something like the 3-way handshake.

Cryptographic Security

IPSEC is (reasonably) cheap, but—how do the parties get the keys?

Pre-shared Doesn't scale.

KDC Requires interaction with online KDC. That requires either (loosely) synchronized clocks or multiple round trips.

IKE IKE or other public-key based systems are expensive and require multiple round trips.

Other public key Expensive; could be insecure if done wrong.

Conclusion: key management is expensive, and has its own set of retry/congestion issues. Keys can be cached—but that requires good locality of reference, and not too many peers.

Stateless Security?

- Packet replay is often a threat.
- If packets have a counter, hosts need to keep that state.
- Challenge/response increases the number of round trips.
- The other choice is synchronized clocks, and one generally needs counter state for the clock skew limit.

⇒ Conclusion: some state is generally needed.

No Transport Security?

- Maybe transport security is too expensive.
- Can *sometimes* use object security (i.e., DNSSEC)
- Better if application-level forwarding is used (email, DNS, etc.)
- But—must still watch for replays, lifetimes, etc.

Denial of Service Issues

- Watch out for messages from bogus source addresses that make a server use up state (i.e., TCP SYN flooding).
- Partial solution: package up state, cryptographically seal, and send back to purported client.
- Client must return packaged state (in its three-way handshake).
- Guards against some IP address spoofing if no eavesdropping used.

My Transport Wish

- Many protocols (i.e., ftp) require secondary channels.
- Each one does it its own way.
- Firewalls need to deal with all this.
- A standard multiplexing mechanism—for UDP as well as TCP—would make firewalls much simpler.