A “Bump in the Stack” Encryptor for MS-DOS Systems

David A. Wagner
daw@cs.berkeley.edu
U.C. Berkeley

Steven M. Bellovin
smb@research.att.com
AT&T Bell Laboratories

Roadmap

- Background on ipsec
- Overview of our approach
- Details on the prototype
- Concluding notes

Background: ipsec

- A future IETF standard for IPv4 and IPv6
- Provides IP-layer encryption and authentication
- Establishes security associations: host-to-host, host-to-gateway, ...

Encapsulation:

- Transport mode (host-to-host)

Eth	IP	ipsec	TCP	data

- Tunnel mode (routers, gateways)

Eth	IP	ipsec	IP	TCP	data

Overview

Motivation:

- MS-DOS implementation of ipsec
- "Call-home" security for mobile users

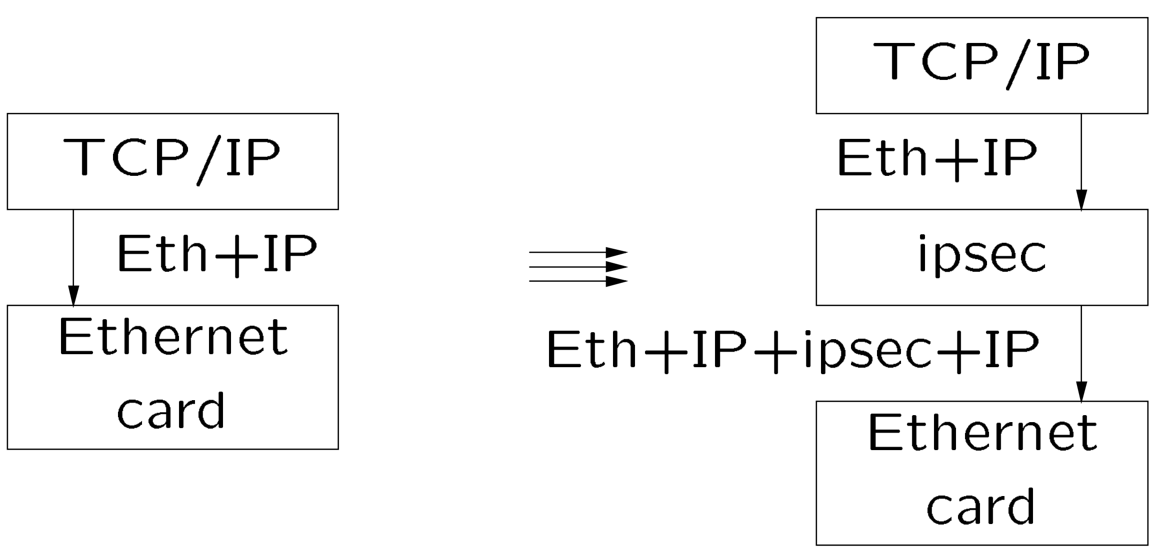
Problems:

- Source to TCP/IP stacks not available
- Many different MS-DOS TCP/IP stacks
- Multiple hardware driver interfaces: packet driver, NDIS, ODI

Overview

Our approach:

- Conceptually, an outboard encryptor (a virtual "bump-in-the-cord" unit)
- The implementation actually resides *below* the TCP/IP stack
- Emulates Ethernet card to TCP/IP stack, and vice versa



Details

Outbound processing:

- Select security association, according to local policy
- Encapsulate via tunnel mode
- Add authentication
- Select route to next hop
- Fragment

Details

Inbound processing:

- Reassemble fragments
- Verify authentication
- Remove encapsulation
- Filter unauthenticated IP packets, according to local policy
- Beware subtle spoofing attack when TCP/IP stack will act as a router

Details

Policy:

- Set manually by local admin
- Specifies transforms used on outgoing packets and required on incoming packets
- IP address granularity

Key management:

- Should, in general, use a key management module
- Currently manually specified as list of security associations

Details

Why fragment and reassemble? Correctness.

- Fragmentation needed on outbound:
  — adding ipsec header might increase packet length above the MTU
- Reassembly needed on inbound:
  — sender or intermediate routers may fragment packets
  — we cannot verify authentication on fragments

Implementation notes:

- IP interface not open; we must reimplement fragmentation/reassembly
- Other solutions not so good
  (path MTU discovery, lie about MTU, munge TCP MSS)

Details

Why tunnel-mode encapsulation? Security.

- Fragmentation/reassembly necessary for correctness (see above)

- Authenticating fragments with transport-mode encapsulation introduces subtle security flaw

Concluding notes

Limitations (areas for future work):

- Only includes authentication
- Uses packet driver interface
- Performance: 188 KB/sec $\rightarrow$ 65 KB/sec
- Restricted to manual keying
- How does ipsec fit into an overall security architecture?

Concluding notes

Lessons for “bump-in-the-cord” encryptors:

- Beware MTU;
  Will need to do fragmentation/reassembly

- Beware fragment attack;
  Will need to use tunnel mode encapsulation

- Beware spoofing attack on routers

General lessons:

- Avoid fixed magic values (e.g. MTU)

- Modularity considered helpful

Concluding notes

Conclusions:

- "Bump in the stack" model enables MS-DOS ipsec implementation
- Source to TCP/IP stack is a luxury, not a necessity