The Evolution of IPsec
Topics

• What is IPsec?
• How did it evolve? Why is it the way that it is?
  • Origin
  • Technical constraints
  • Organizational, political, and other non-technical issues
• Yes, non-technical issues matter…
What is IPsec?

- Encryption at the IP packet layer
- Protect *all* packets, without changing applications
- Must conform to the IP service model:
  - Stateless—each packet stands by itself
  - Packets may be dropped, duplicated, damaged—correctness is end-to-end, i.e., handled by the transport layer (TCP)
  - But: at the *network* (IP) layer
- Network layer encryption can protect all packets, even those from naive applications
## Quick IP Refresher

- The IP header is used to route the packet to the destination
  - It contains source and destination IP addresses, plus a “next protocol” indicators
- The TCP header contains port numbers, a checksum, and sequence number data
- Packets may be lost, damaged, duplicated, delayed, replicated, delivered out of order, etc.

<table>
<thead>
<tr>
<th>Payload (User Data)</th>
<th>TCP Header</th>
<th>IP Header</th>
</tr>
</thead>
</table>

- The IP header is used to route the packet to the destination
- It contains source and destination IP addresses, plus a “next protocol” indicators
- The TCP header contains port numbers, a checksum, and sequence number data
- Packets may be lost, damaged, duplicated, delayed, replicated, delivered out of order, etc.
Operational Scenarios

- End-system to end-system
- End-system to gateway (firewall)
- Gateway to gateway
Topologies
Topologies
Topologies
Generic Structure

End System to End System

Encryptor to Gateway
SP3: Early Network-Layer Encryption

- Part of DoD’s Secure Data Network System
- Interesting points:
  - SP3 supported OSI and IP (“DoD Internet”) protocols
  - OSI terminology (PDU, NSAP, etc.)
  - Military terminology: “red” and “black” nets
  - Confidentiality and integrity checks are both optional services
  - Variable-length fields
SP3: Packet Format

Confidentiality

SP3

Clear Header

Protected Header

“protected data”

ICV

Integrity

Clear Header

Protected Header

Length

SE (type)

KEY_ID

1

1

var

Protected Header

Length

Flags

Pad

IP Header

1

1

var

var
Integrity and Confidentiality

- The “protected header” is *always* integrity-protected—necessary for access control

- Integrity only—for export control reasons?

- Confidentiality only—especially when this was designed, cryptographic processing was *expensive*: eliminate integrity checks on high-speed, bulk transmissions that could tolerate occasional bit errors (e.g., video in OFB mode)

- *No* details are given for the cryptographic algorithms
Interesting Aspects

- Cryptographic details—algorithm, block size, length of ICV, etc.—are all identified by the KEY_ID. (Why?)
  - KEY_IDs specify permissible source and destination addresses—used for access control
  - Address format linked to KEY_ID
  - Key negotiation is handled externally
- The flag field only indicates the direction of the packet—prevent reflection attacks
  - Both directions of a connection might share the same KEY_ID, though they didn’t have to
- Padding here can be used to align the IP header to a 4-byte boundary
Key Management

- Separate policy from mechanism
- Slower and more complex, but also done much less often
  - Put per-packet encryption in the kernel; do key management at user level
  - Actually, per-packet encryption can be done outboard, in hardware
  - Allow for complex policies, CRLs, etc.
  - Negotiate multiple keys: different directions, integrity versus confidentiality, etc.
  - Forward secrecy
Policies

• Encryption and/or integrity protection

• What should be encrypted on transmission?
  • By destination IP or net address? (By host name?)
  • Port numbers?

• What should have been encrypted on receipt
  • Algorithms, e.g., open or NSA Type 1?
    • Key lifetimes, in seconds or bytes
  • Address of decryptor
Ioannidis and Blaze: swIPe (1993)

• Simplification of SP3
  • Eliminate most options
  • Internet-only—no OSI support
• But: a sequence number is added “to protect against replay”
  • Huh? The IP service model permits packet duplication—is this needed?
  • No further explanation given
• Freely available running code for two popular Unix variants
Enter IPsec

- An IETF Working Group
- Goal: an Internet standard for packet-level encryption
- A descendant of SP3 and swIPe—the designers of IPsec were very familiar with both
  - The designers of swIPe were part of the IPsec process
- An Internet standard has to have more generality, and hence more options, than swIPe
  - Example: must support multicast and MobileIP
Desiderata

• (Availability of) ubiquitous network-layer encryption
  • Network layer, because that would protect all traffic, even that of naive applications
  • “Availability of” because computers were too slow then to encrypt everything—but we knew they’d get faster
  • We wanted to replace address-based authentication
  • Security policy “selectors” would include IP addresses, host names, port numbers, and usernames
    • My traffic could be protected differently than yours
  • Multiple granularities of encryption: network pair, host pair, per-user, per-connection
Constraints

• US export controls on cryptography
  • You needed a license to export confidentiality technology; authentication technology was not restricted

• Limited cryptographic state of the art

• Designers had somewhat limited cryptographic knowledge
RFC 1825 Architecture

- Separate confidentiality (ESP) from authentication (AH)
- Explicit “transport” versus “tunnel” mode, similar to SP3
- Have a separate key management/policy protocol—but it was never defined
- Relied on SPI—security parameter index—that serves the same role as the KEY_ID in SP3
- Unlike swIPe, no sequence numbers
Packet Layouts

- IP
- ESP
- TCP
- user data

- IP
- AH
- ESP
- IP

- TCP
- IP
- ESP
- AH
- IP

- user data
- TCP
- IP
- ESP
- AH
- IP
The IPsec SPI

- Random
- Separate SPI for each direction
  - No need for SP3’s flag
  - In theory, harder to link traffic in opposite directions, since the SPIs don’t match
  - Is it a problem in practice?
- Bound to a source/destination address pair
- The pair `<dest_IP, SPI>` identified the key and parameters
  - Multicast packets shared a multicast destination IP address
  - Also need SPI for rekeying
- Padding was for cipher block size—but could also be increased to (try to) defeat traffic analysis
- (IV shown is 8 bytes, for DES)
- (Shaded portion is encrypted)
- ESP use was *optional*
Integrity: AH

- Similar SPI definition
- The AH header covered not just the remainder of the packet but also part of the preceding IP header, e.g., the source and destination addresses and some IP options
  - Layering violation—and made implementations very messy!
  - N.B.: some parts of the IP header change en route
- Use of only AH was export-friendly and permitted firewalls to inspect packets without knowing a key
- AH was optional—integrity check could be omitted

<table>
<thead>
<tr>
<th>proto</th>
<th>length</th>
<th>reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>digest (variable length)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Other Issues

• Sequence numbers—were they needed?
  • Matt Blaze said “yes”, because of replay attacks
  • I said “no”, because IP had to handle packet duplication anyway
  • I won…

• Some people wanted RC4 for encryption—much faster
  • RC4 is a stream cipher, which must not be used with manual key management (though WEP did it…)
  • But there was a strong desire to support manual keying

• Many people disliked AH because of its layering violations
Key Management for IPsec

- Basic framework proposal: ISAKMP, from the NSA
  - The IETF added the cryptography (called IKE): roughly speaking, an RSA-signed dialog with optional Diffie-Hellman exchange for forward secrecy
  - There was another protocol proposed, Photuris
- ISAKMP/IKE is *horribly* complex
  - Includes session management as well as key negotiation
  - Had many different modes, phases, authentication schemes, etc.
  - No time for a thorough treatment of it—but it was a disaster (and had serious functionality bugs)
**SKIP: The Road Not Taken**

- IP is a stateless datagram protocol
- ESP/AH (and SP3 and swIPe) require key negotiation and setup—and that requires state
  - It’s no longer a pure datagram protocol
  - That’s why ISAKMP has session management: when are keys deleted?
- SKIP was a stateless alternative
• Agree on a Diffie-Hellman modulus $p$ and base $g$

• Each node $i$ has a certificate for its DH half-key: $g^i \bmod p$

• The key $K_{ij}$ for traffic between $<i,j>$ is $g^{ij} \bmod p$

• Use that key to create a traffic key $K_p$ — rekey after some fixed limit

• Integrity, encryption, sequence, compression are all optional (controlled by flag field)
Problems...

- The varying offsets, depending on options and algorithms, make parsing more difficult.
- The algorithm identifiers are sent in the clear—might aid cryptanalysts.
- Policy is less flexible; no provision for forward secrecy.
- *There needed to be universal agreement on algorithms—no chance to negotiate them.*
- *There needed to be universal, permanent agreement on Diffie-Hellman parameters.*
Organizational Politics Time

- Many people preferred Photuris to ISAKMP/IKE
  - There were “personality conflicts” regarding Photuris—only ISAKMP remained
- There was a bitter split, and no consensus, over ESP/AH versus SKIP
  - SKIP was enhanced—and made more complex—to handle optional forward secrecy
- Ultimately, the working group could not decide; the Security Area Director had to call it
  - Crucial issue: *the inability to change the Diffie-Hellman parameters*
- Sun Microsystems (which was behind SKIP) had recently had a security disaster with bad Diffie-Hellman parameters
Final Outcome

- ESP/AH won over SKIP
- Sequence numbers were deleted
- No design concessions were made to the export rules ("the Danvers doctrine")
  - Given the expense of encryption with 1995 hardware, integrity-only (i.e., AH) was a rational alternative
- Many working group members were exhausted by this time
  - ISAKMP was selected as the only choice; no one had the energy to propose an alternative
There Were Problems…

• There were no good integrity algorithms then
  • HMAC, once invented, was a drop-in replacement
• Lack of sequence numbers was a mistake
• Lack of mandatory integrity checks was a mistake
• The suggested IV selection method for DES-CBC—a simple counter—was a mistake
• ISAKMP was a mistake
• Most of these issues had to do with lack of cryptographic expertise in the IETF’s IPsec working group
A Helpful but Misunderstood Rumor

- There was a rumor floating around that the NSA could break CBC encryption
- The claim seemed obviously wrong to me—but I decided to investigate
- That was a good move…
CBC Mode

An IV plus \( n \) blocks of plaintext yields \( n \) blocks of ciphertext. But—cutting and pasting CBC streams is interesting…
The Prefix of a CBC Message is a Valid CBC Message
The Suffix of a CBC Message is a Valid CBC Message
Error Propagation is Limited
Environmental Assumptions

- Host-pair keying: a single key exists between each pair of hosts
- Only encryption is being done; there is no cryptographic authentication header
- The attacker may have a login on one or both of the machines
- The attacker can monitor, delete, modify, or inject messages onto the wire (a standard assumption)
Reading a Message

User sends:

```
|   IP   |  ESP  |   TCP   |  data  |  data  |
```

Enemy sends:

```
|   IP   |  ESP  |   UDP   |   any  |   any  |   any  |
```

Inject:

```
|   IP   |  ESP  |   UDP   |   TCP  |  data  |  data  |
```

Paste together enemy UDP header with target's payload—
and IPsec will do the decryption for the attacker
Use the target's TCP header, plus attack text encrypted for the enemy. Calculating a checksum fix-up isn't hard, and is only $2^{16}$ tries anyway.
More Attacks Like These!

- Generate full-scale packets
- Guess at passwords *without* a login on either machine
- Many more!
What Went Wrong?

- We really needed sequence numbers
  - Benign packet duplication is *not* the same as malicious retransmission
- We really needed integrity checking: CBC’s easy cut-and-paste properties make it crucial
- And the NSA/CBC rumor? Probably, it was the IV algorithm: *predictable* IVs are a serious weakness
- Crypto theory people had tried to tell us these things—but they weren’t as engaged with the working group: the in-person standards process still matters
Fixing IPsec

- The old ESP spec was discarded
- The new one has a sequence number field and an integrity field
  - Yes—I argued to take out sequence numbers, and then argued to put them back in...
  - Integrity can *still* be turned off—again, for high speed, bulk transmissions
New ESP and AH Formats

SPI
sequence
payload
payload
padding
padding
padlen
proto

HMA
Cd
igest
HMA
Cd
igest
HMA
Cd
igest
HMA
Cd
igest
HMA
Cd
igest

next proto
length
reserved

SPI
sequence
HMAC (variable length)
AH Isn’t Needed

- Don’t need to protect IP addresses; they’re bound to the SPI
- Can’t protect other interesting IP header fields, e.g., source routing, since they change en route
- Use a “null cipher” option for authentication-only with ESP
- Many in the IETF would like to move away from it
The IP, TCP, and ESP headers all have sequence numbers—are they redundant?

No—they serve different purposes.

From a security perspective, the ESP sequence numbers are *within the cryptographic module’s trust boundary*. TCP’s are not.

Module boundaries matter—and for security stuff, you want to trust as little as possible from outside.
Key Management

• Because of the complexity and bugs of ISAKMP/IKE, the IETF adopted a newer, (somewhat) simpler version

• Many people were still unhappy
  • Some of us proposed a replacement for IKE, JFK (“Just Fast Keying”)
  • The IETF adopted it—and at the next meeting, changed its mind and went back to the replacement IKE
Other Changes

- Newer cryptographic algorithms and modes of operation have been adopted
  - Elliptic curve, AES, combined confidentiality/integrity cipher modes, longer keys, etc.
- IKE had another, unforeseen bug: new hash algorithms couldn’t be negotiated properly
- The sequence number field was too small: 32 bits
# Feature Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>SP3</th>
<th>swIPe</th>
<th>SKIP</th>
<th>Original IPsec</th>
<th>Final IPsec</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Integrity</strong></td>
<td>Optional; linked to KEY_ID</td>
<td>Optional; flag in header</td>
<td>Optional; flag in header</td>
<td>Optional; requires AH header</td>
<td>Encouraged; linked to SPI</td>
</tr>
<tr>
<td><strong>Algorithms</strong></td>
<td>Linked to KEY_ID</td>
<td>Linked to key identifier</td>
<td>Given in header</td>
<td>Linked to SPI</td>
<td>Linked to SPI</td>
</tr>
<tr>
<td><strong>Sequence Numbers</strong></td>
<td>No</td>
<td>Yes</td>
<td>Optional; flag in header</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Inner Header</strong></td>
<td>Sometimes present; it's complicated</td>
<td>Always</td>
<td>Optional; flag in header</td>
<td>Optional; flag in header</td>
<td>Optional; flag in header</td>
</tr>
</tbody>
</table>
Lessons

- Real-world cryptographic protocols have to be *engineered*—the cryptographic mathematics alone do not suffice
- People matter—we didn’t always have (or heed) the proper expertise
- Process matters
- Requirements vary over time, as speeds increase, threats change, and newer algorithms are developed
Did We Succeed?

• ESP and AH are pretty clean
  • But it was hard for applications to tell if or how a connection was protected, especially since IPsec could be outboard
  • The IETF doesn’t do APIs
• ISAKMP/IKE was (is) a disaster—far too many options made configuration and interoperability very, very difficult
  • The ubiquity of the Web and the spread of SSL (aka TLS) made IPsec less interesting
  • Other technologies, especially NATs and firewalls, got in the way of IPsec
  • Username selectors were a bad idea—wrong layer
• We did not get ubiquitous network-layer crypto, but we did get VPNs
Questions?

Brown creeper, Central Park, April 2, 2019