Tyler Jackson

4/22/2015

Data and Network Security

HW4

1. For this problem remember AH has 32 bit SPI, 32 bit Sequence #, and 32 bit ICV(encrypted MAC)

-also remember before comparing hash on outbound processing to zero out all mutable fields

For this problem remember ESP adds an ESP trailer and ESP Auth. For the simplicity of the model I left out the TCP header, ESP trailer, and ESP Auth in my packet descriptions. The TCP header and ESP Trailer are included in the ESPs encryption

a)

A--------------->R1 then R1--->R2---->R3 then R3---------->B

unencrypted encrypted unencrypted

Packet 1

IP HEADER | Data

Dest. IP | Src. IP

192.168.2.66 | 204.171.2.3 | all 1280 bytes of Data

Packet 2

New IP header| ESP | orig. IP header | data

234.167.57.223 | 129.111.2.47 | ESP | Encrypted orig. IP header | encrypted first 1024 bytes

of Data-header - size

of other fields

-ESP would have a hash of the new payload (orig. IP header | data) for integrity

Packet 3

234.167.57.223 | 129.111.2.47 | ESP | Encrypted orig. IP header | encrypted rest of bytes

-packet 2 and 3 would be relayed through R2

Packet 4

-packets 2 and 3 get merged back together at R3 and the new IP header and ESP get dropped

-ESP confirms the data wasn’t tampered with

192.168.2.66 | 204.171.2.3 | all 1280 bytes of Data

1. b)

A-------------->R1-------------------->R2-------------------------->R3---------------------->B

Transport Tunnel Nested Tunnel Tunnel

encrypted encrypted encrypted Encrypted

Packet 1

IP Header | AH | Data

Dest. IP | Src IP | AH | Data

192.168.2.66 | 204.171.2.3 | AH | All 1280 Bytes

- AH would hold a Hash of the Data and the immutable IP header fields (i.e.

dest and src address)

Packet 2

new IP header | new AH | orig. IP header | AH | first 1024 bytes of data - size of

other fields

192.168.2.66 | 192.111.2.47 | AH |IP hdr from packet 1 | AH | first 1024 bytes -

size of other fields

Packet 3

192.168.2.66 | 192.111.2.47 | AH |IP hdr from packet 1 | AH | rest of data bytes

- The new AH would hold a hash of the immutable fields in the new IP header (i.e. new dest. and src. IP addresses) as well as the payload hash which would include the first packets AH

- At R2 these packets would stop and enter another nested tunnel

Packet 4 ←------------------------encrypted ----------------------------------->

new IP header |ESP | 2nd IP hdr|AH | orig. IP hdr |AH |1st 1024 bytes data-

size of other fields

234.167.57.223| 213.71.82.34|ESP |2nd IP hdr |AH |orig. IP hdr | AH | first 1024 bytes

- size of other fields

Packet 5 ←------------------------encrypted ----------------------------------->

new IP header |ESP | 2nd IP hdr|AH | orig. IP hdr |AH |rest of data

234.167.57.223| 213.71.82.34|ESP |2nd IP hdr |AH |orig. IP hdr | AH | rest of data

* for the nested tunnel we add a second level of new IP headers and this time our tunnel uses ESP. That means that we encrypt the data (assuming we are using encrypted ESP), and take a hash of the payload
* At R3 these new headers get stripped off, and our ESP is confirmed to match the data so that we know the payload hasn’t been changed and then packets 2 and 3 continue on to B
* At B packet 2 and 3s new headers get stripped off and we confirm with the hash in our AH from our first tunnel that the payload wasn’t messed with and we are left with packet 1 that has been fragmented into 2 packets. These packets can be merged together and we can compare the AH from our transport protocol to make sure that the packet hasn’t been altered.

2. 16 bit-sliding replay window

current window with slots

0011111101111110

first slot = N

Receive N+4

* We already have that packet - verifies ICV and then rejects packet

Receive N+2

* WE already have that packet - verifies ICV and then rejects packet

Receive N+8

* We don’t have that packet yet. Verify ICV and then accept packet

Receive N+19

* This is past our window range. Verify ICV and then accept packet and slide window up 4 slots.

Receive N+2

* This is before our new window range. Reject packet. Although it doesn’t matter, because we already received that packet.

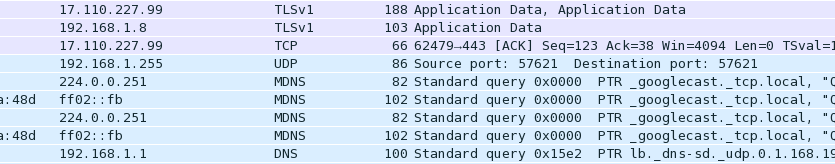
3. SSL VPNs vs. IPSec VPNs

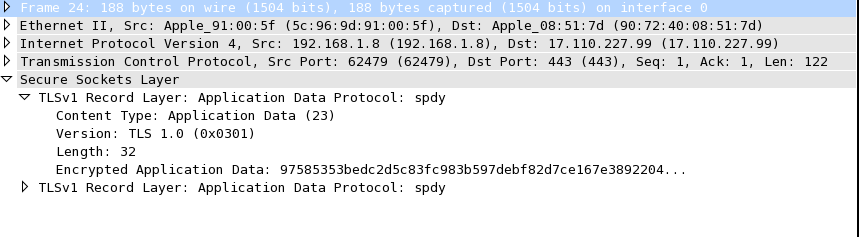
IPSec requires changes to the operating system but no application changes and people are reluctant to use it because of how complex it is (way over-engineered) and because of its lack of interoperability. SSL requires changes to applications but no changes to the O.S. People are reluctant to adjust their applications to be SSL compatible when they are already IPSec compatible. Both have integrity, encryption, and authentication. SSL is a simpler specification.

4.

Here is my connection using SSL







Setting up a VPN with IPSec

I tried a couple different ways, either using my windows/mac vpn guis, or using openswan and setting up an ip.config file, but I wasn’t able to get it to work. That being said, for an IPSec tunnel, the ip header would have been encrypted in the pay load, and a new IPSec header would have been prepended to the packet with an ESP header holding an encrypted hash of the pay load.

