“On the Internet, nobody knows you’re a dog.”
THREATS TO DISTRIBUTED APPLICATIONS 1

“How do I know I am connecting to my bank?”

Maybe an attacker . . .

. . . sends you phishing email (pretending to be from your bank) and you click on a link in it

. . . gets access to your packet stream and changes the IP destination addresses

. . . hijacks the DNS entry of bigbank.com

. . . hijacks routing to the IP address of bigbank.com
THREATS TO DISTRIBUTED APPLICATIONS 2

WHO COULD SPY ON THEM OR TAMPER WITH COMMUNICATION?

- the networks themselves
- people who . . .
  - . . . tap a wire
  - . . . penetrate an infrastructure machine
  - . . . put a wireless receiver near a transmitter
  - . . . connect to a wired broadcast medium such as a cable network

WHAT CAN ATTACKERS DO?

- read data
- absorb packets
- modify packets
- inject packets into the stream
"WHOM AM I TALKING TO?"

the answer (whatever it is) is an identity

most likely, the bank wants as identity the user name “Jane Q. Public,” . . .

. . . and asks for a password for endpoint authentication
IDENTITIES USED IN SESSION PROTOCOLS

- bank server's IP interface (name is IP address)
- network member, session endpoint
- has access to (checked by endpoint authentication)
- identity
  - is the user of this session endpoint
  - domain bigbank.com, which is the name of the Web server
  - owns (certified by certificate authority)
- public/private key pair
  - each key is at least 2,000 bits
PUBLIC-KEY ENCRYPTION

\[ K^+(K^-(\text{data})) = K^-(K^+(\text{data})) \]

BUT knowing one of the pair, it is very difficult to compute the other!

CHALLENGER B

VERIFIABLE ENDPOINT A

Are you A, who has public key \( K^+ \)? Nonce \( n \)

\[ K^-(n) \]

\[ K^+(K^-(\text{data})) = n \]

OK!
THE TLS HANDSHAKE

accomplishes . . .

(1) agreement on cipher suite,
(2) endpoint authentication of server,
(3) key exchange

important to do a good job of this

CLIENT B

- cipher suites I support,
  my nonce NB

SERVER A

- choice of cipher suite,
  my nonce NA, my certificate

- important to use an up-to-date one, cf. TLS 1.3

- using private key K, compute
  $K^{-1}(K^+(\text{pre-master-secret}))$ to get
  pre-master-secret

- from pre-master-secret, NB, and NA, compute 2 encryption
  keys and 2 authentication keys

from pre-master-secret, NB, and NA, compute 2 encryption
keys and 2 authentication keys

lots of keys!
A HIGH-LEVEL IDENTITY IS MOBILE

CLIENT CAN . . .

- log in from another computer
- disconnect identity from session by logging out
- move around while using a mobile device (even if the identity goes with the device)

SERVER CAN . . .

- lend keys and certificate to a trusted representative, e.g., a content-delivery network
- attach a digital signature to data, so its identity can travel anywhere with the data
DATA ENCRYPTION AND MESSAGE AUTHENTICATION 1

TLS PACKET

- IP header
- TCP header
- TLS header
- data
- encrypted
- authenticated
- MAC

ESP PACKET, TUNNEL MODE

- IP header
- ESP header
- IP header
- TCP header
- data
- ESP trailer
- different networks, different addresses
- MAC

ESP PACKET, TRANSPORT MODE

- IP header
- ESP header
- TCP header
- data
- ESP trailer
- MAC

encryption embedded in TCP

TCP embedded in encryption
MESSAGE AUTHENTICATION 2

TLS PACKET

IP header
TCP header
TLS header

data
MAC

MAC = H(data + key)
cryptographic hash function

authentication key, just for this direction, and known by both endpoints

recipient recomputes same formula, checks that it is the same as MAC
MESSAGE AUTHENTICATION 3: THE CATCH

MAC ensures that received packet came from sender without modification.

But attacker could still delete, re-order, or replay packets.

**TLS PACKET**

TCP sequence numbers won’t help because attacker can change those to make sequence look right.

so TLS packets have implicit sequence numbers (can make them implicit because of reliance on TCP)

MAC = H(data + key + number)

**ESP PACKET, TRANSPORT MODE**

can’t rely on TCP, so ESP headers have explicit sequence numbers . . .

. . . which may not arrive in order

so receiver deletes packets with previously received numbers (replay attacks)
ENCRYPTION AND AUTHENTICATION 4: SCOPES

WHY ARE THE SCOPES OF ENCRYPTION AND AUTHENTICATION DIFFERENT?

TLS PACKET
- IP header
- TCP header
- TLS header
- data
- MAC
- authentication is extra-secure?
- encrypted
- authenticated

ESP PACKET, TUNNEL MODE
- IP header
- ESP header
- IP header
- TCP header
- data
- ESP trailer
- MAC
- a middlebox can authenticate even if it doesn’t decrypt?

why not authenticate the IP header, too?

IPsec AH does this, and packets can’t even pass through a NAT
Why? A NAT cannot make a compound ESP session, because the session identifier is not standard.

Ugly hack: pretend UDP has persistent sessions, use with well-known port 4500, this signals endpoints that ESP is embedded inside UDP.
IF THE MIDDLEBOXES HAVE THE SAME INTERESTS AS ADJACENT ENDPOINTS, THEY CAN BECOME PART OF THE TLS SCHEME.

**Endpoint Authentication and Key Exchange for this Hop**

**this handshake does endpoint authentication for the real endpoints, key exchange for the middle hop**

**Endpoint Authentication and Key Exchange for this Hop**

**middleboxes could be doing security, performance optimization**
TRUSTED MIDDLEBOXES ARE PART OF THE APPLICATION

—ABOVE THE ENCRYPTION!
ENCRYPTION AND MIDDLEBOXES 3

paths through the enterprise network where packets are not encrypted

SIDESTEP THE ENCRYPTION!
ENCRIPTION AND MIDDLEBOXES 4

MIDDLEBOXES ARE BELOW THE ENCRYPTION!

<table>
<thead>
<tr>
<th>N1 header:</th>
<th>N2 headers:</th>
<th>N3 headers:</th>
<th>N4 headers:</th>
<th>N4 payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet</td>
<td>IP, UDP, GTP</td>
<td>IP, UDP, ESP</td>
<td>IP, TCP</td>
<td></td>
</tr>
</tbody>
</table>

at Level 4, everything in the packets is encrypted

at Level 3, data is encrypted, headers are not

Level 2 is a cellular network, which has several middleboxes that may care about Level 3 headers

IP + UDP + ESP

IP + UDP + GTP
SECURITY FOR CONTROL PROTOCOLS

BECAUSE THEY CHANGE THE STATE OF THE NETWORK!

CANNOT ALWAYS USE TLS OR ESP

- in session-location mobility, an identity must update its own location, but may not have a certificate or past history with the server

- control protocols can be very high-volume (DNS, routers exchanging filtering information)

- protocols may be too old

*and attackers can afford a lot of tries, guessing how to get in, because there is little risk*
HELP FOR CONTROL PROTOCOLS

DON’T ACCEPT UNSOLICITED REPLIES

e.g., ARP accepts unsolicited replies to requests . . .

. . . which are broadcast to every member of network . . .

. . . so any member of network can reply “I have requested IP address”

USE NONCES OR RANDOMIZATION

attacker queries local server for cnn.com

so local server may need to query another server, if no cache or old cache

CHECK REPLIES FOR CREDIBILITY

in the U.S., the closest cnn.com server is not in Brazil

woe to those late-blooming U.S. services whose IP addresses are in Brazil

where is cnn.com?

cnn.com is at my address

at least, when updates are requested

to prevent this, server can put a nonce (random number) in a new or unused field of the request . . .

. . . and expect the reply to carry the same information