

# Security 3

---

## Almost-SSL Example

This is almost how SSL works...

Alice contacts Bob

Bob forms pub/priv at random, sends pub to Alice

Alice forms "session" key  $k$ , sends to Bob, encrypted with pub

Now both have session key  $k$ , for traditional encryption for the duration of the session

## Attack: Man In The Middle (MITM)

Carl takes over router between Alice and Bob

Alice thinks she is talking to Bob, but in fact is talking to Carl

Bob thinks he is talking to Alice, but in fact is talking to Carl

For the Almost-SSL example above, Carl can substitute his own pub/priv talking to Alice, and so get  $K$  and observe or alter the communication

## MITM Truism

If two parties do not share any prior info, they are always subject to the MITM attack

Anything Bob can do, Carl can do

If Alice does not have any prior knowledge about Bob, there is nothing Bob can do to distinguish himself from Carl

## Attack: DNS poisoning

A way that Carl might get between Alice and Bob

Feed bad data to the local DNS server, so that when a local user contacts "www.bank.com", the DNS server gives Carl's IP addr.

Carl puts up a page that looks like the bank page

## Attack: Fake ATM

Over long weekend, put up a fake ATM in front of the real ATM, with some "work in progress" signs.

People put in their card, type in their PIN number

Carl records the info, spits out the card with an error message

Later, use the info to pillage the accounts

Conclusion: would like mutual authentication -- each party can verify the identity of the other

## MITM Solution - Certificates

Alice contacts Bob

Bob sends back certificate (cert) giving Bob's pub key  
The Cert is signed by a Certificate Authority (CA), known to Alice.  
The Cert associates Bob's identity/DSN name, with Bob's pub key  
Alice verifies that the cert is valid, since the cert is signed by the CA  
Alice forms session key k, sends out with pub  
Carl can intercept the traffic, but cannot do anything with it. Only the true Bob  
can recover k, since only Bob knows priv.  
The CA acts as a 3rd party, verifying for Alice what the pub key is for the alleged  
other party.

## Own CA

You can be your own CA if you like, you just need to add yourself to the local list  
of trusted CAs.

e.g. foo.com sets up three servers, A, B, C, each with a pub/priv, signed by the  
foo.com CA key. The programs on A, B, C are set to trust the foo.com CA  
When A connects to B, it gets and verifies the cert as usual -- protecting against  
MITM

## Identity Conclusions

Don't trust DNS/IP addr to provide identity of other party  
Solutions...

1. Shared secret key, exchanged earlier
2. Cert identifying the pub key of the other party. Send them a session key under  
that pub key -- only the correct other party will be able to decrypt and use the  
session key

## Technologies

### Special IP Adrs

The IP subnet 192.168.0.0 is set aside as a special "non-routable" network. An  
organization can use 192.168.0.0 addresses internally, and the routers are not  
supposed to let those adrs leak outside the local network. That way, another  
organization can also use 192.168.0.0 addresses, and the two organizations will  
not conflict. 10.0.0.0 is another non-routable network, but 192.168.0.0 is more  
commonly used and so is probably more reliable.

### Network Address Translation -- NAT

Cable/DSL modem has one IP addr and an upstream router as usual  
Network Address Translation NAT

RFC 2663

Allow computers on a network to share a single IP addr on the Internet.  
Translating router

Single transfer point between the internal network and the Internet at large -- it has a "LAN" side and an "Internet" (aka "WAN") side

The router has one IP addr on the Internet side

On the LAN side, the router is 192.168.1.1

The hosts on the LAN have addrs in the range 192.168.1.100, 192.168.1.101, 192.168.1.102, ....

The router will probably use a DHCP server to distribute configuration to the LAN computers

## NAT Steps

The NAT router notices the start of a TCP connection from its LAN side -- noting the IP addr and port number of the originating LAN computer and the IP addr and port number of the destination computer.

Suppose we have a computer on the LAN with IP addr 192.168.1.100

A computer on the LAN operates like any other TCP/IP computer. It forwards its packets to its router in the usual way.

Suppose the router's IP addr on the Internet side is 1.2.3.4

The router takes in traffic from its LAN side, and then translates the packet on its way out -- the packet used to say "From: 192.168.1.100", gets translated to appear to be from the router itself "From:1.2.3.4". The other end of the connection thinks it is interacting with 1.2.3.4 in the usual way. However, the router knows to re-send incoming packets in that conversation on its internal network to 192.168.1.100. The router keeps track of which port numbers are currently part of which conversation.

## NAT Results

Multiple computers on the LAN side can have conversations at the same time.

The router needs to keep these straight.

To the outside world, it appears that all of the traffic is coming from 1.2.3.4 -- just a lot of traffic, since it's actually the sum of all the traffic from the LAN computers.

Internally, the computers think they have IP addrs in the 192.168.1.100 -- which is not the same addr that the computers in the outside world think they are using.

Problem: if a computer on the inside sends what it thinks is its IP addr to a computer outside the LAN, it's not going to work. Mostly, that's not a problem -- regular "call/response" connections work fine.

Incoming calls don't work well -- what LAN computer should it go to? Can program the NAT router to send certain port numbers to certain fixed IP addrs on the LAN side.

Problems with UDP datagrams for IP phone -- the protocol arranges for sound packets to arrive on some port number, e.g. 20561, but the NAT router doesn't know which client machine to send it to.

# Firewall

Similar to NAT

Separates the internal net from the rest of the internet.

Only allow certain port#'s connections to go through in certain directions.

e.g. incoming port 80 requests, only allowed to the web server, not any other internal IP addr.

Also, can provide a false sense of security -- if the HTTP server has a vulnerability, the firewall is no help.

Modern worms, traverse in multiple ways (email), so they get inside the firewall

Firewalls can cause legitimate network services not to work -- creates a tension where the security people don't want to allow anything. In sum, their policy may be costing more in lost productivity than it saves.

# Virtual Private Network

Set up an encryption layer in software between all your computers on the Internet. The VPN traffic is all encrypted/authenticated before being sent inside regular Internet packets on the Internet at large.

IPv6 does this

# Security -- Blame The User

Many computer users are casual, and do not actively maintain their system

Blaming the user for not being up to date is not a realistic way to deal with security

# Server Security vs. Complexity

Set up server with web server, file sharing, ....

The system is complex, so it's hard to get every detail right.

If there are 1000 details, it takes real dedication to get all 1000 right.