Plan for the day

- Crypto Reminder
- Transport Security Layer (TLS)
- Denial of Service
- Firewalls, DMZ,…

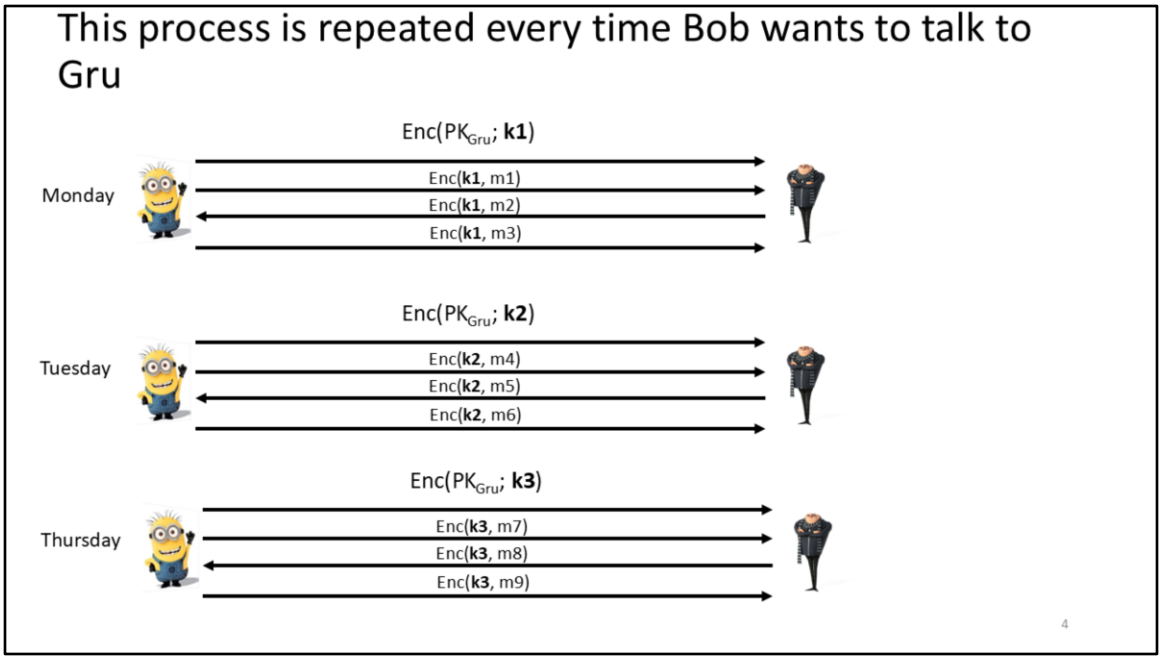
Computer Security (COM-301)
Applied Crypto - A Reminder

Original slides by Carmela Troncoso
Some slides/ideas adapted from: George Danezis

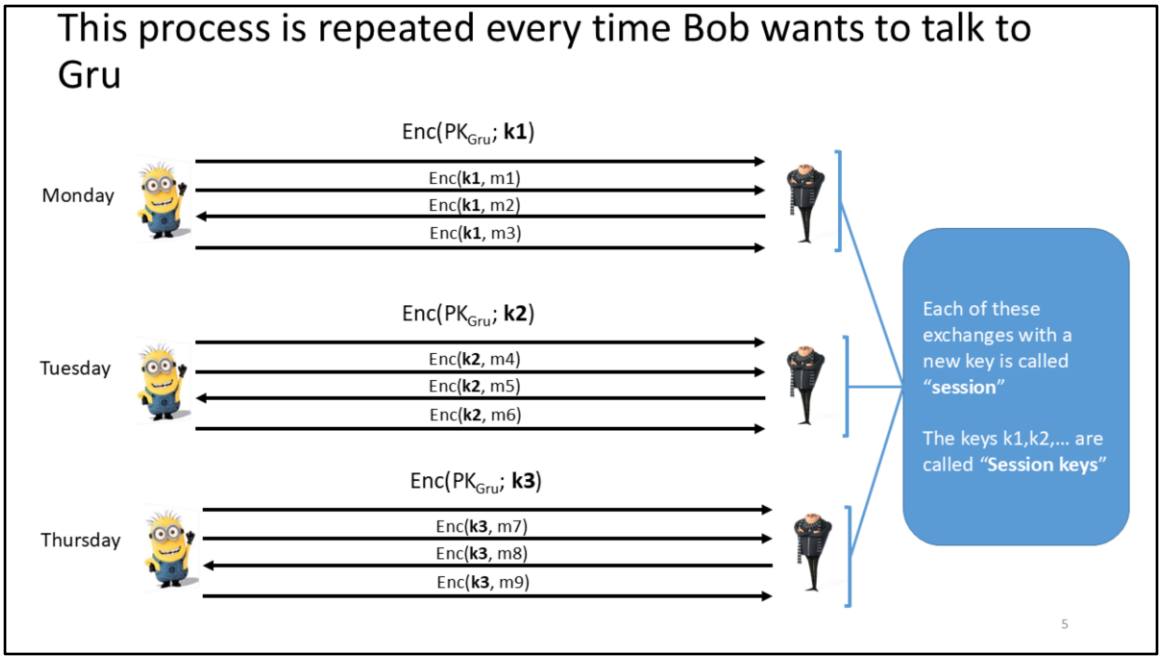
Two variations of public key cryptography:

- DH key exchange (we get a new “random” key for every session, out of nowhere)
- RSA-based (public/private key encryption)

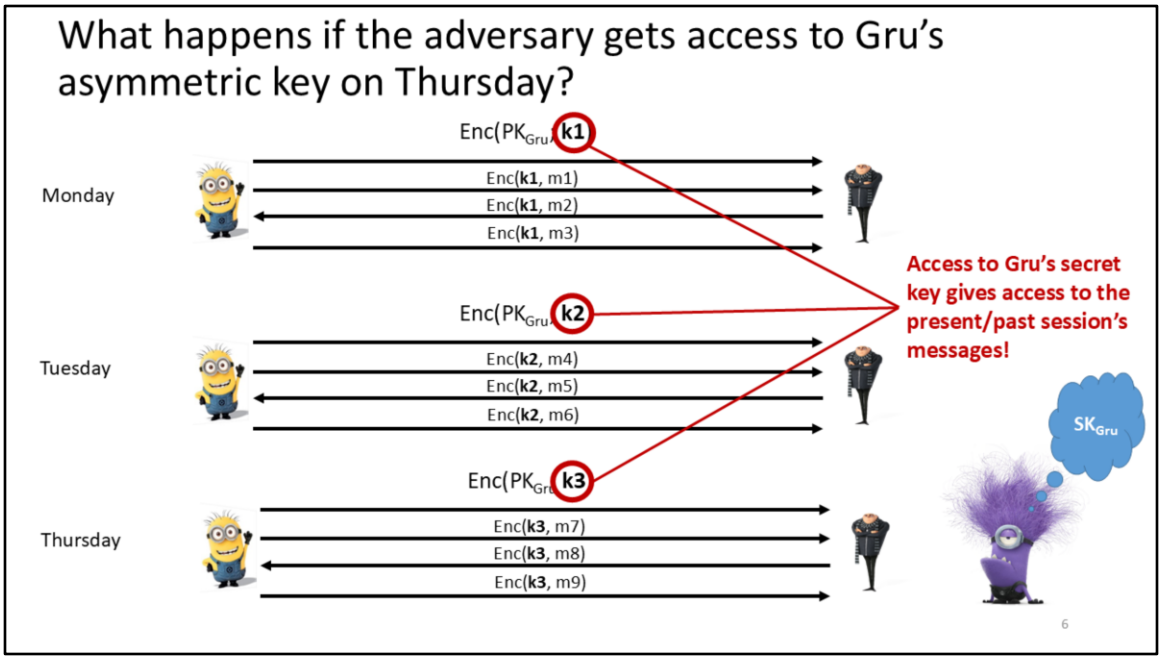
3



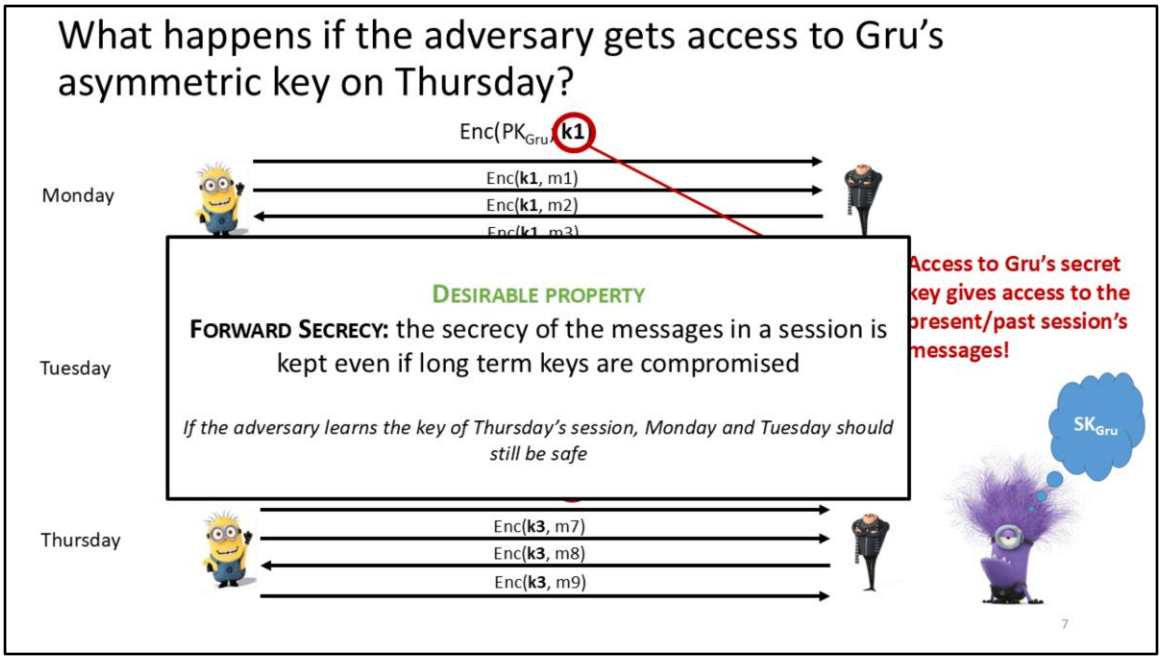
Every time Bob communicates with Gru, he can create a new key. These are called session keys.



Every time Bob communicates with Gru, he can create a new key. These are called session keys.

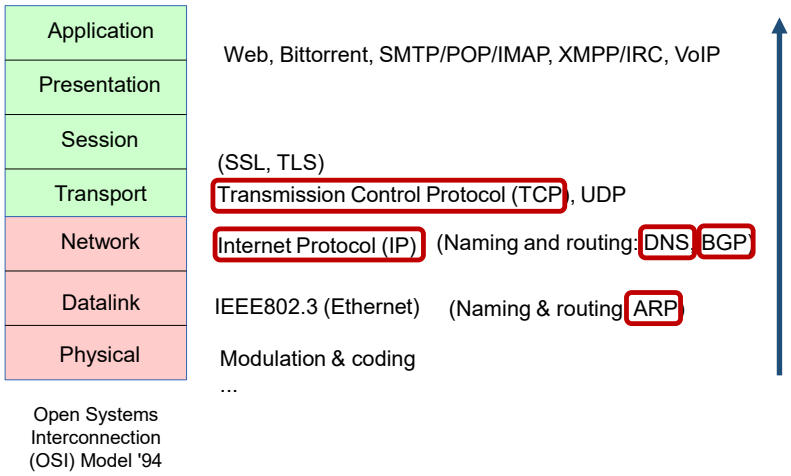


But there is a problem in this scheme: If the adversary gets hold of Gru's secret key , the secrecy of all past sessions is compromised



We want to avoid this. A key compromise at time **t** should not compromise the secrecy of any past conversation.

Where are the problems?



8

Computers communicate with each other at different layers. These are typically modeled by the Open System Interconnection (OSI) Model. This model covers from the physical layer, where pulses that codify bits are sent, to the application layer where programs talk to each other.

In this lecture we will cover protocols at different layers, and see the security problems and potential solutions.

Computer Security (COM-301)

Network security
Transport Layer Security

Original slides by Carmela Troncoso
Some slides/ideas adapted from: George Danezis

Basic steps of TCP hijacking

Who: a man in the middle adversary (MITM)
- can observe communication
- can intercept and inject packets

How can we solve this?

Cryptographically authenticate all exchanges! Not only at the start

But TCP cannot* do that…

What:

1- Wait for TCP session to be established between client and server

2- Wait for authentication phase to be over

3- Only then use knowledge of sequence numbers to take over the session and inject malicious traffic.

4- Use malicious traffic to execute commands, …

5- The genuine connection gets cancelled (desynchronized or reset).

https://www.techrepublic.com/article/tcp-hijacking/

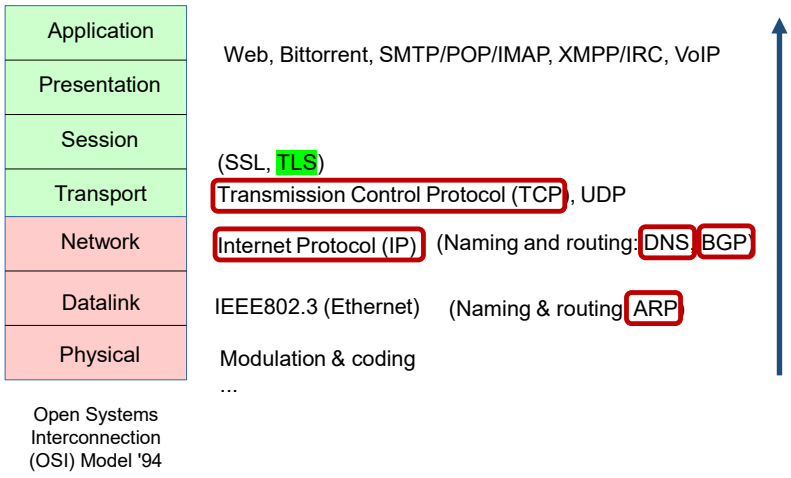
10

In the case of TCP, it is hard to add cryptographic operations to authenticate the exchange, as it was done with previous protocols.

Security checks cannot only happen during the login phase (the establishment of the TCP connection).
To apply the mediation principle, you would need per packet authentication. Every single piece of data sent must be cryptographically signed so the receiver knows it came from the real user, not an imposter injecting data into the stream.

TCP cannot* do that: TCP-AO, TCP-MD5 (weak)…

Where are the problems?



11

Computers communicate with each other at different layers. These are typically modeled by the Open System Interconnection (OSI) Model. This model covers from the physical layer, where pulses that codify bits are sent, to the application layer where programs talk to each other.

In this lecture we will cover protocols at different layers, and see the security problems and potential solutions.

The Transport Layer Security (TLS)

- Cryptographic protocols above TCP -- “middlelayer”

- **Goal**: providing communications security:
  - Confidentiality: symmetric encryption
  - Authentication (One or two-side): public key cryptography
  - Integrity: MAC and signatures

- Provides **forward secrecy**
  - Learning a secret at one point in time does not reveal anything about the past

- State of the art: TLS v3
  - Reality: a zoo in the Internet (it is difficult to upgrade a huge number of computers)
  - Previous protocol: SSL, same principles but many vulnerabilities -- deprecated!

12

To solve this issue, a new “layer” appeared between TCP/IP and application. Here, the **Transport Layer Security Protocol (TLS)** provides the properties that TCP cannot:

- Confidentiality: it enables hosts to agree on a symmetric key to encrypt their packets
  - This key can be agreed in such a way that the protocol provides forward secrecy, i.e., if the adversary learns the key of one TLS session, this key does not enable the adversary to decrypt previous sessions.
- Authentication: it enables hosts (servers and clients) to use public key cryptography to authenticate themselves
- Integrity, through the use of signatures when using public key crypto, and MAC when using symmetric crypto.

Even though the state-of-the-art is TLS 1.3, there are many lower, insecure, versions of TLS (and even its predecessor SSL) are still deployed. This is because updating machines is hard. Many are badly managed, or managed with people with low security knowledge, or simply need to maintain the previous version to keep their operation.

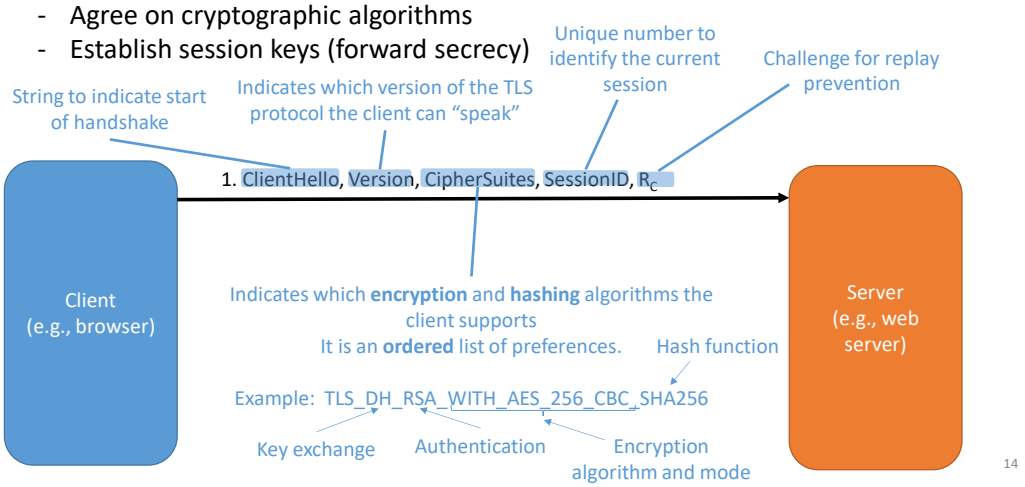
The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)

All the security properties provided by TLS are bootstrapped in the TLS handshake in which two hosts (e.g. Client and server) agree on a key.

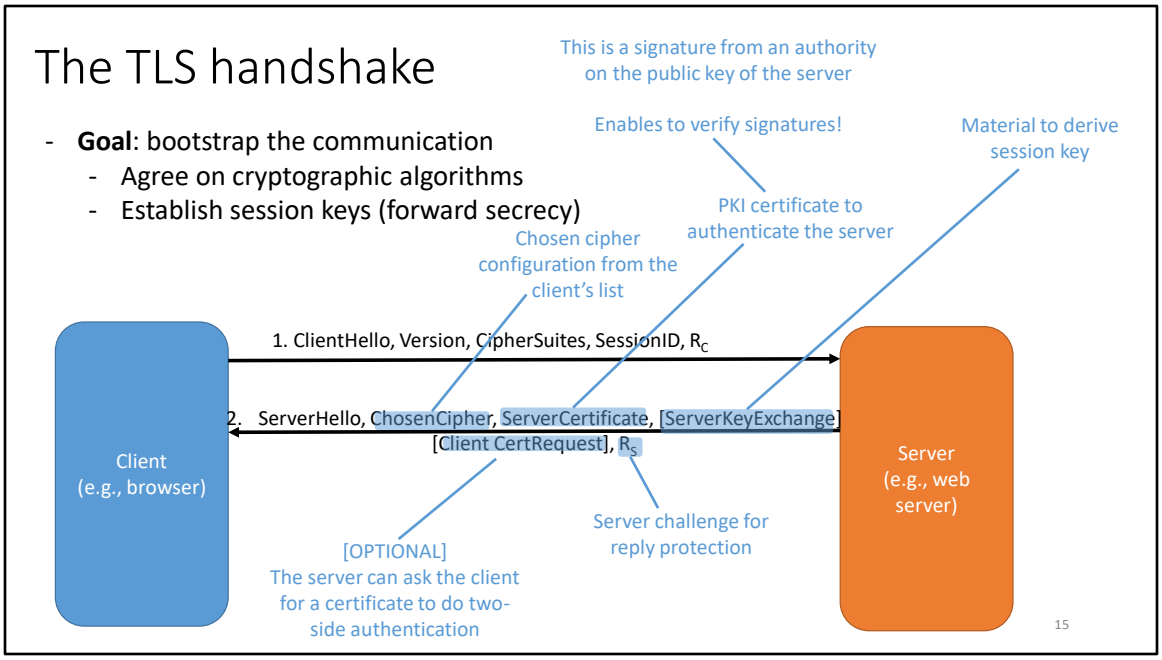
The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)



Step 1: the client declares its willingness to start a secure TLS connection and informs the server of:

- The higher TLS version it is able to speak
- The cipher suites (algorithms for key agreement, signature, encryption, and hashing) it can compute. This is a list of tuples such as the one in the example, ordered by the client preferences – in general from more secure to less secure.
- A session ID to identify the TLS flow
- A challenge to ensure freshness of the communication (i.e., reply prevention)



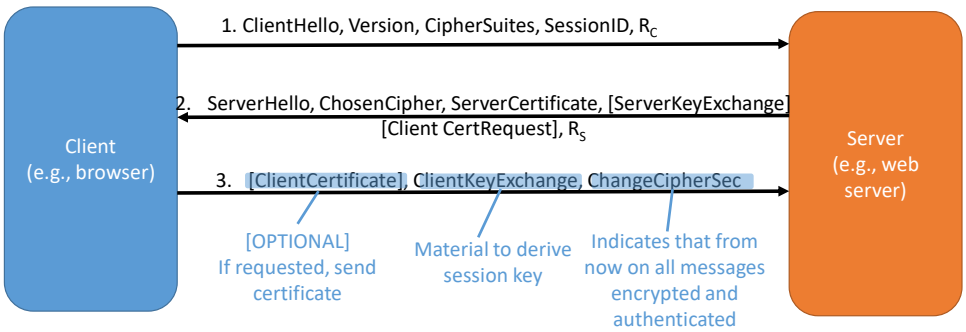
Step 2: the server declares its willingness to continue the secure TLS informs the client of:

- From the cipher suites offered by the client, which one is preferred by the server.
- The server certificate: a signature on the public key of the server by an authority trusted by the client. It enables the client to verify the identity of the server given a signature with the server's private key.
- If the client and server are agreeing on a key, e.g., using Diffie-Hellman, then the server provides material to derive the key (e.g., g^a).
- Optionally, the server can also ask the client to provide a certificate for two-sided authentication.
- A challenge to ensure freshness of the communication (i.e., reply prevention).

The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)

After step 3 Client and server have a shared session key!!!

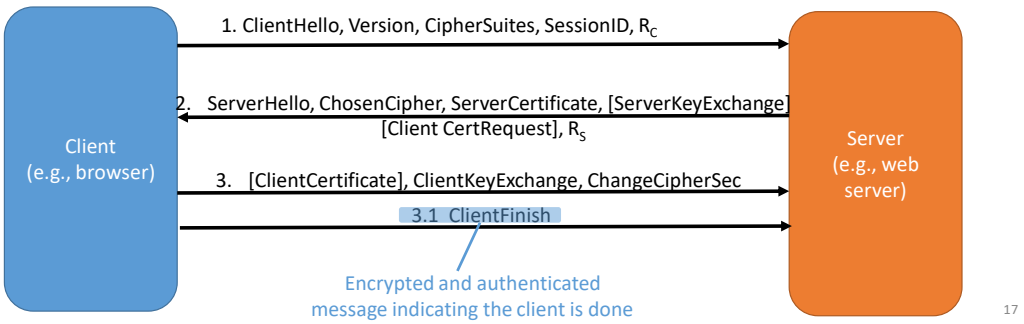


16

Step 3: the client responds with:
- If the server requests it: the client's certificate: a signature on the public key of the client by an authority trusted by the server. It enables the server to verify the identity of the client given a signature with the client's private key.
- If the client and server are agreeing on a key, e.g., using Diffie-Hellman, then the client provides material to derive the key (e.g., g^b).
- Sends a flag indicating that from then on the client will send all messages encrypted and authenticated using the algorithms agreed on Step 2 using the key derived on Step 3

The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)

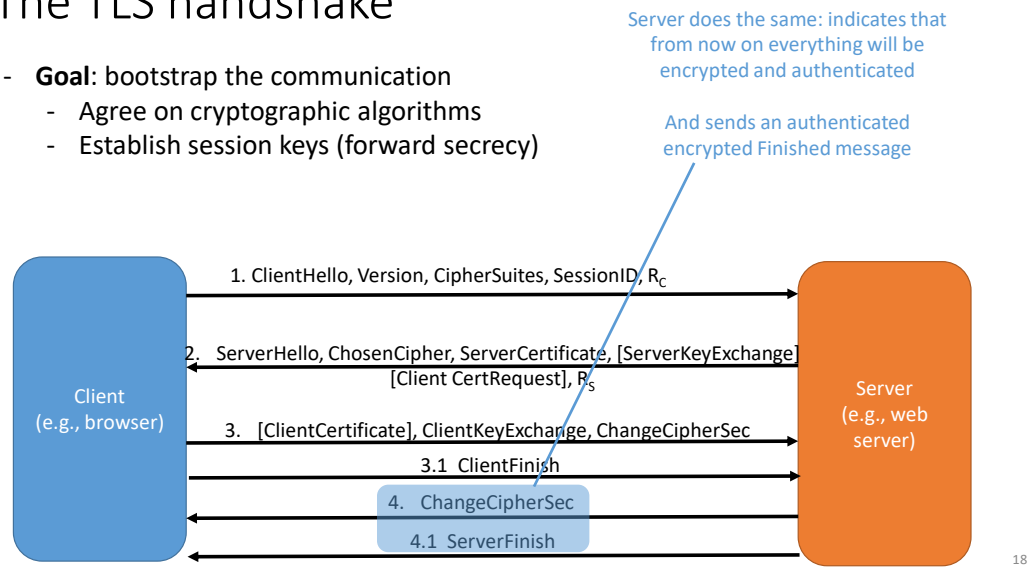


17

Step 3.1: the client informs the server that it has finished its side of the handshake. This message, as it follows the ChangeCipherSec flag, is already encrypted and authenticated.

The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)



Step 4 and 4.1: the server also indicates that it will, from then on, encrypt and authenticate all messages as agreed; and indicates that it has finished its side of the handshake.

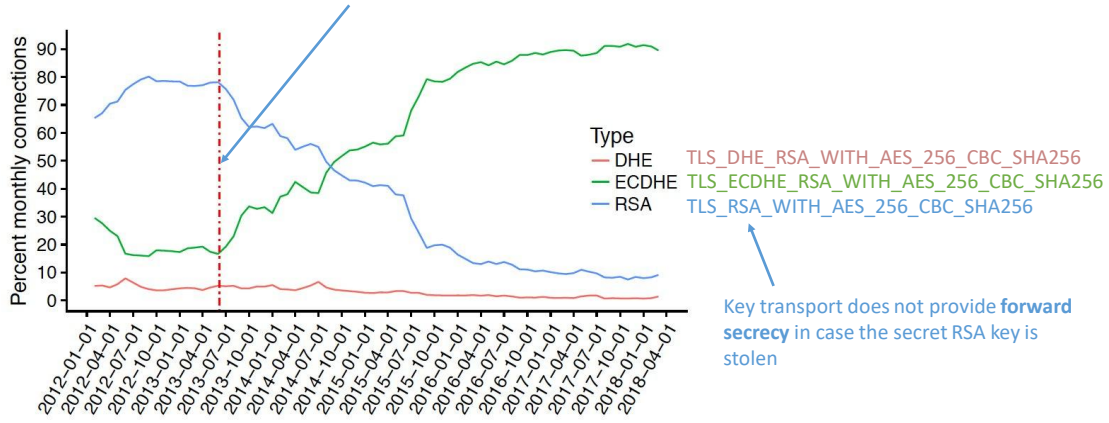


Figure 8: Negotiated RSA and forward secret connections. Dotted line shows the date of first Snowden revelations.

19

TLS offers two modes to obtain a shared key between server and client:

- Key transport: using the public key of the server, the client sends a symmetric key. This mode is not forward secure, as if at any point the key of the server falls in the hand of an adversary, this adversary can decrypt all keys in the past.

- Key agreement: using Diffie Hellman (or its Elliptic curve equivalent) the client and server agree on an ephemeral key for the session. This key is independent in every session.

Attacks on TLS

Advanced
Not for exam

Downgrading attacks (CVE-2014-3511): implementation flaw that enables the adversary to force server and client to use a less secure version of TLS/SSL.

BEAST (CVE-2011-3389): exploits an implementation weakness in TLS 1.0 implementation of Cipher Block Chaining (CBC) which results in predictable initialization vectors. This allowed to decrypt parts of a packet, and specifically to decrypt HTTP cookies when HTTP is run over TLS

Padding Oracle: because of the MAC-then-encrypt design, TLS is vulnerable to padding oracle attacks. These use block padding as an "oracle" to find out whether a decryption is right or wrong

- **Lucky Thirteen** (CVE-2013-0169): timing side-channel attack that allows the attacker to decrypt arbitrary ciphertext

Renegotiation attacks: exploit the "renegotiation" feature of TLS that enable users to have new parameters. The adversary can inject his own packets at the beginning of a connection

Many more… DoS, more crypto problems, more protocol problems… Nowadays provable security in TLS 1.3

https://tools.ietf.org/html/rfc7457

20

SSL is a terrible name – OpenSSL story: https://openssl-library.org/post/2018-12-20-20years/

Computer Security (COM-301)

Network security

Denial of Service

Original slides by Carmela Troncoso
Some slides/ideas adapted from: George Danezis

What Properties?

From Lecture 1

TRADITIONAL PROPERTIES

- **Confidentiality** – prevention of unauthorized disclosure of information
(*e.g. The adversary should not be able to read my bank statement*)

- **Integrity** – prevention of unauthorized modification of information
(e.g. The adversary should not be able to change my bank balance)

- **Availability** – prevention of unauthorized denial of service
(e.g. The adversary should not prevent me accessing my bank account)

Denial of Service

Goal: prevent legitimate users from accessing a service

Option A - Crash victim: exploit software flaws to make it stop

Option B – Exhaust victim's resources

- Network: Bandwidth
- Host
  - Kernel: TCP connection state tables, etc.
  - Application: CPU, memory, etc.

23

The previous attacks focus on Confidentiality, Integrity and Authentication. In the next slides we will study Denial of Service, which focus on attacking Availability. The goal of this attack is to take down a service, or to prevent some clients from accessing this service.

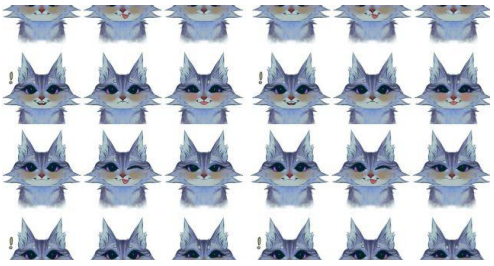
Denial of Service can be achieved by:

- **making the service crash** by exploiting some bug on the software. When the victim crashes, it cannot be accessed

- **exhausting the service's resources** such as bandwidth (so that no-one else can contact the service), or the service's CPU or memory (so that even if the host an be contacted, it cannot be used).

Example 1 – Skype kittens DoS (CVE-2018-8546)

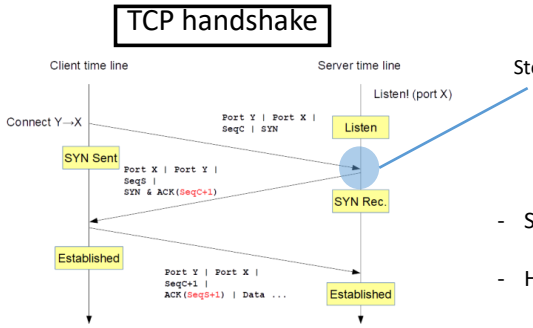
When receiving about 800 emoji of kittens at once, your Skype for Business client will stop responding for a few seconds. If a sender continues sending emojis your Skype for Business client will not be usable until the attack ends.



24

Improper handling of emojis in Microsoft Skype for Business and in Lync enables an adversary to consume all the resources associated to these programs in such a way that they cannot be used for any useful activity (calling, messaging, etc).

Example 2 – TCP SYN flood



At this point, the server is listening, waiting for the next ACK in order to establish the connection

Stores a TCP Control Block (TCB) ~ 280 bytes

How does the attack work?

- Send TCP SYN packets with bogus source address
- Half-open TCB entries exist until timeout
- Kernel limits on # of TCBs!!!

Resources exhausted ⇒ new requests rejected

25

In a SYN flood attack the adversary exploits the fact that after receiving a SYN and sending a SYN/ACK, the server stays waiting for the connection to continue. At that point, the server stores some information about the connection in a so-called TCP Control Block (TCB). The Kernel can host a maximum number of TCBs. An adversary can exhaust the space for TCBs. Then, the host cannot open new connections anymore.

TCP SYN flood Prevention – use “cookies”

Principle: Minimize the state before you are “authenticated” (i.e., before finishing 3-way handshake)

Don't create the full TCP Control Block, instead

- Compress TCP state: Very tiny state representation for half-open conns
- A few bytes per connection => can store 100,000s of half-open connections

Push the state to the client! “SYN cookies”

- Upon receiving a message, derive the state
- Cryptographically protect the state under a fixed key (confidentiality and integrity)
- Send it back to the client
- Require the client to provide it back to complete the protocol

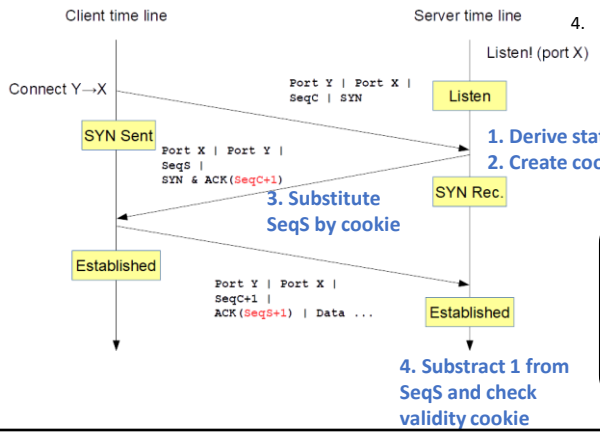
26

One way of preventing TCP SYN flood attacks is to reduce the amount of state kept by the server until the handshake is done. Instead, the server stores a very small state that enables to have open orders of magnitude more connections.

In particular, the idea is to not store the TCB but send it to the client. To ensure its validity, the TCB can be protected with cryptographic means, such that the server can check correctness when the TCB comes back.

TCP SYN flood: a possible prevention

1. Upon receiving a message, derive the state
2. Cryptographically protect the state under a fixed key (confidentiality and integrity)
3. Send it back to the client
4. Require the client to provide it back to complete the protocol



OTHER METHODS: PROOF OF WORK

"economic" measure to deter denial of service attacks.
Require work before anything is done (e.g., compute some hashes). Easy to do once and to verify, expensive to do many and DoS

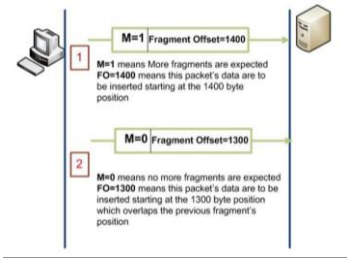
Other methods to deter Denial of Service rely on requiring the client to so work before engaging in the operation. This is called a **proof of work**. It can be for instance compute some hashes which are easy to verify but expensive to compute. This method is not a perfect countermeasure, but increases the cost barrier to deploy DoS at large scale.

Original SYN cookies explanation: http://cr.yp.to/syncookies.html

Example 3 – Teardrop attack

Sometimes packets are too long to be transmitted as 1 unit.

IP includes a process to deal with *fragmentation* to be able to divide packets in transmittable units.



The attack works by sending packets with overlapping off sets.

This would mean the packets would overwrite each other

Some OS process to deal with fragmentation could not deal with this behavior and would either crash or wait for more packets that would never arrive.

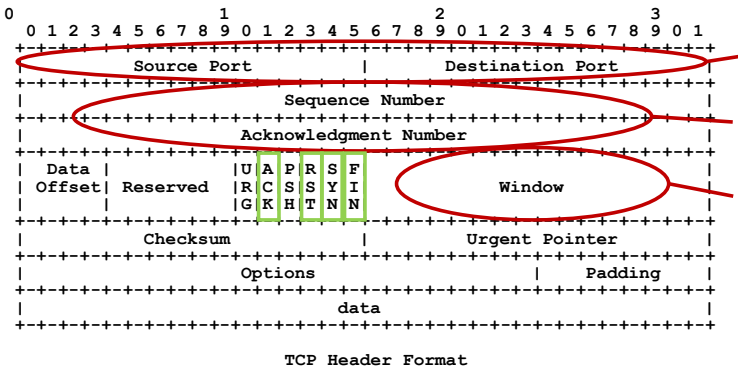
https://www.pluralsight.com/blog/it-ops/ping-of-death-and-dos-attacks 28

Teardrop attack

In a teardrop attack the adversary exploits a bug in TCP/IP that crashed the machine when the adversary sends fragments that cannot be reassembled.

TCP header

Refresher



TCP Header Format

Multiplexing

Reliability
Congestion control

Flow Control

Well known Ports:
20-21 – FTP
22 – SSH
25 – SMTP
53 – DNS
80 – HTTP
110 – POP3
143 – IMAP
443 – HTTPS

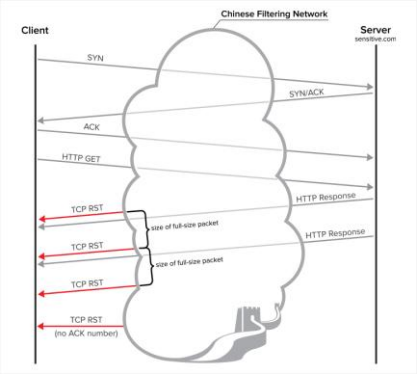
RFC793 (1981) http://www.ietf.org/rfc/rfc793.txt

30

Example 4 – DoS without flooding: TCP RST Injection
(e.g., used by the Great Firewall of China)

When the Great Firewall detects an undesired flow, it injects forged TCP resets (with the RST flag bit set) into the data streams so that the endpoints abandon the connection.

Why not simply dropping the packets from undesired source/destinations?



https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClIyCAC

31

The Great Firewall of China exploits the fact that there is no authentication in TCP to just end connections initiated inside the network. After the TCP 3-way handshake is finished, instead of stopping the messages from the server, the Great Firewall sends forged RST packets to the client. When receiving this packets the client closes the connection, *even* if the server still sends answers. This way, the denial of service happens without the server being attacked (from the point of view of the server, the client just lost its connection).

1. To drop a packet, a firewall must act as a MITM. It must physically receive the packet, hold it, inspect it, and then decide whether to let it go or destroy it.
2. What are the risk of crashing

Computer Security (COM-301)

Network security
Other protection technologies

Slides originally created by Carmela Troncoso
Some slides/ideas adapted from: George Danezis

Other dimensions of networking and security

- **Remember**: cryptography is key for protection.
- Other dimensions to protect the network:
  - Firewalls
  - DMZs
  - Intrustion Detection System

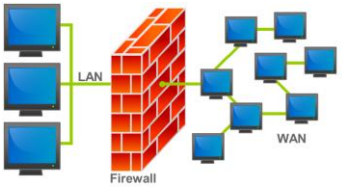
– NATs?

33

For most of the attacks that we have seen in this lecture, the protections come from some cryptographic protection, or from pushing work to the client.

There are other protections that help mitigating attacks, mainly by reducing the chances that an adversary can interact with the victim machine.

Network Firewalls



FIREWALL: network router that connects an internal network to an external (public) network. It *mediates* all traffic, and makes "access control" decisions according to a policy.

- Firewall "access control":
  - Inspects characteristics of the traffic,
  - "Allow" or "deny" traversal across the firewall.
  - Prevent flows that could be dangerous or contravene to a security policy in the internal network.

34

Another common protection is the use of firewalls. These are routers that inspect the communication and implement some access control rules to decide which flows should enter in the local network.

Firewalls - Simple packet filter (1980s)

Inspect each packet in isolation and Reject/Allow depending on certain "rules"

Rules:

- "Equal" or "not equal", or "in range".
- Fields: IP Src, IP Dest, Port numbers, Protocol Type.
- **Example**:
  - Force all email traffic to go to a specific mailserver:
    `(Dest IP = mailserver, Dest Port = 25) → Allow`
  - Only allow mailserver to connect to other mailservers:
    `(Src IP = mailserver, Dest Port = 25) → Allow`
    `(Src IP = *, Dest Port = 25) → Deny`

Advantages: simple to implement in sw, can also be implemented in hw, very fast decisions

Disadvantages: Limited policies can be expressed, limited filtering on content possible

35

The simplest policies are those made by basic rules based on the operators "equal", "not equal", or "in range" applied to the IP source and destination, the port numbers and the protocol.

For example, a rule might allow Web traffic (port 80) onto the network only if packets have the destination IP address of the organization's Web server.

This type of policies are very simple to implement, and are very fast on runtime. On the downside, they cannot express very complex reasoning, and they are very static (i.e., they cannot support dynamic protocols that require opening of ports). Also, it only operates on the headers and does not allow any decision based on content.

Stateful Firewalls (1990s)

Understand TCP/UDP semantics → can Reject/Allow depending on the state

Stateful firewall vs. stateless packet filter – Example

FTP protocol client opens a connection to the server, and then the server connects back to a high port of the client to transfer the file.

- *Simple packet filter (stateless firewall)*: choice between allowing all packets to high ports all the time or none
- *Stateful firewall*: can detect an active FTP session with the server and allow a connection back to a high port from the same server to the same client!

36

A more complex policy, allows to deal with more dynamic protocols in such a way that decisions for allowing or rejecting flows depend on the state of the system.

A simple example is that of FTP, which uses two channels, one for issuing commands and other for transferring data. These channels use two ports, traditionally 21 for the commands, and the data port may vary typically being a very high port. (https://www.techrepublic.com/article/how-ftp-port-requests-challenge-firewall-security/)

Using the simple packet filter policy, a firewall can only decide to have high ports open all the time, or never let an FTP connection use those ports for data transmission.

A stateful firewall, on the contrary, would detect the existence of an FTP session, and then and only then allow connections in high ports.

Application Firewalls (1990s)

DEEP PACKET INSPECTION (DPI): evaluate the content, and allow/ reject based rules
can be stateful or stateless

Examples:

- Transparent redirection of HTTP traffic to a local proxy to save bandwidth
- Transparent blocking of certain websites (social networks from a workplace)
- Scanning downloaded executable resources for viruses
- Blocking peer-to-peer protocols, no matter which port they use
- Monitoring traffic to detect leaks of sensitive documents

And if traffic is encrypted (IPSec, SSL/TLS)?

- Option 1: block all encrypted traffic.
- Option 2: Install client certificates that enable for decryption & inspection at the firewall.

37

The stateful firewall improves over the stateless in that it enables dynamism requires in many protocols. It does not, however, deal with the connection content. It is still focused on the headers.

This can be done using **Deep Packet Inspection (DPI)**, a technique that evaluates the content an applies rules. This enables much more sophisticated filtering, as those mentioned in the slide.

DPI cannot be applied when the traffic is encrypted. At that point there are two options: block all encrypted traffic, or ask clients to use keys (or certificates to agree on the keys) known to the firewall so that the firewall can decrypt and inspect the communication.

Downsides with firewalls

Key problems

- Full mediation is **slow** (read/check/write) – observation is cheaper (read/inject).
- Can a firewall authenticate any principals?
- Can a firewall ensure the correctness of the data on which it makes decisions?

Role of firewall in security engineering

- Only allow "known good traffic"? ← not possible at this level (what is good traffic in an intranet?)
- Therefore: "filter out definitely bad traffic" and "filter all traffic of a certain class".
- Remove the noise of background network attacks
- Hosts will still have to implement robust defences to prevent security policy violations by bad behaviour that has been made to "spoof" good characteristics
- **Key lesson**: a firewall **is not** a full substitute for other host and network security mechanisms!

38

Even though firewalls are powered by hardware enhancements, they are still slow: they need to take the information from the wire, read it, process it, and write it again.

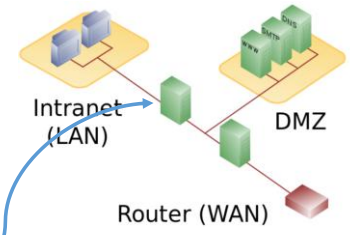
The firewall acts at the network level, thus it cannot make any decision about authentication or authorization at the application layer, as it does not know the principals behind the flow.

Moreover, remember the rest of the lecture: network protocols have very little security. In particular headers (the main target of firewalls) have no confidentiality or authenticity. How can the firewall be sure it acts on authentic information?

In general, firewalls are not strong security mechanisms, but they help reducing the surface of attack.

Defence in depth: the De-Militarized Zone (DMZ)

Split “the world” into 3 zones
- **WAN** – outside
- **DMZ** – with public services
- **LAN** – for internal users only



Relies on a firewall to
- Ensure only traffic to well known services traverses outer firewall.
- Ensure only traffic from “**bastion host**” enters LAN from DMZ. Thus the bastion host can perform access control and filtering (eg. VPN/IPSec, Proxy).
- Result: LAN can access DMZ and WAN; DMZ can access WAN. But flows in the other direction are restricted, monitored and authenticated.
- In case a service is compromised internal resources are safe!

39

Firewalls are useful to establish defense in depth. This is achieved by using two firewalls. The outer one has lax filtering policies, and let’s packets enter to the services that must be connected to the outside (email server, web server, etc….).

The second firewall, also known as **bastion host**, has much stricter rules and can use more expensive policies as it supports much less traffic than the outer firewall.

In this combination, if web services are compromised, they can only

Internet toDMZ: Allowed (for specific services).
DMZ to Internet: Allowed.
LAN to Internet: Allowed.
LAN to DMZ: Allowed (admins need to manage the servers).
Internet to LAN: BLOCKED.
DMZ to LAN: BLOCKED. (This is the most critical rule. If a server in the DMZ is compromised, it cannot open a connection to your internal database).
affect other web services and not the internal services behind the bastion host.

Intrusion Detection Systems

It monitors and alerts on suspicious activity.
Acts as a "Burglar Alarm" not a lock.

Detection Methods:
Signature-Based: Matches known attack fingerprints.
Anomaly-Based: Flags deviations from "normal" behavior.

Deployment Types:
- NIDS: Network-based (Monitors the road).
- HIDS: Host-based (Monitors the house).

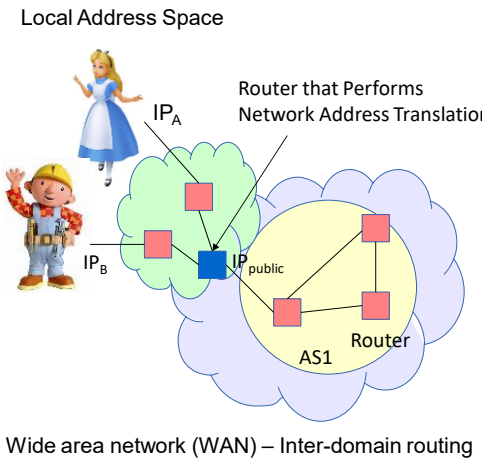
Major Limitation: It is effectively blind to encrypted traffic, without expensive decryption mechanisms (TLS inspection).

40

Network Address Translation (NAT) is the process where a network device, usually a firewall, assigns a public address to a computer (or group of computers) inside a private network. The main use of NAT is to limit the number of public IP addresses an organization or company must use, for both economy and security purposes.

While the motivation to use a NAT is mostly economical, it also brings security benefits. As there is only one public IP, the adversary cannot contact machines in the network unless there is already a mapped port open to the public.

Network Address Translation (NAT)



NAT: router that maintains routing tables of the form: (Internal IP, port) ↔ (External IP, port).

Why? Save IPv4 address space (only 32 bits of it!)

Security implications: an external entity *cannot* route into the NAT unless it knows (or figures out) an already mapped port.

41

Network Address Translation (NAT) is the process where a network device, usually a firewall, assigns a public address to a computer (or group of computers) inside a private network. The main use of NAT is to limit the number of public IP addresses an organization or company must use, for both economy and security purposes.

While the motivation to use a NAT is mostly economical, it also brings security benefits. As there is only one public IP, the adversary cannot contact machines in the network unless there is already a mapped port open to the public.

Summary

- The network is **hostile – insiders can be as evil as outsiders**
- **Cryptography is an important tool** to address network security problems
  - Authenticity, Confidentiality and integrity of traffic content and sessions
  - Authentic binding of names → secure naming (DNS, ARP)
  - Authenticity of routes & routing updates → secure routing (BGP)
  - Strong authentication allows for reliable authorization
- Denial of service **can be defended**
  - Try to **not keep state** and try to **make the adversary work**
- Other techniques are ultimately **weak(er)**:
  - Firewalls, IDS, filtering, … → weak against strong network adversaries that can MITM
  - They may protect against weak adversaries and/or provide some defence in depth.