SAVONIA

Cybersecurity in computer networking

Cybersecurity Fundamentals

Markku Kellomäki
Jussi Nivamo

SAVONIA

Computer networking in a nutshell

- Computers need to communicate with each other.
- Communication is done through networking devices and peripherals.
- All networking devices and peripherals utilize protocols to facilitate necessary functions.
- Protocols as communication standards provide common rules for communication between devices and applications.
- *When rules for communication are known, they can be broken or exploited.*

SAVONIA

Cybersecurity in networking

- Planning, designing and implementing secure solutions which are even partially composed of network traffic.
- Recognizing the bottlenecks, known factors, anticipating unknown factors.
- Doing whatever you can to mitigate possible attack vectors.

SAVONIA

Types of malicious interference in networks

- (Distributed) Denial of Service
- Honeypotting
- Man-in-the-middle
- Device impersonation
- Session hijacking
- Weak cryptography abuse

SAVONIA

Mitigation of malicious interference in networks

- Firewalling
  - Software
  - Hardware
- Zero-trust-principle
  - Expect all devices on the whole route from sender to receiver to be compromised at some point.
- Physical security
  - Ensure no physical cabling can be changed.
  - Learn to recognize unauthorized traffic in networks.

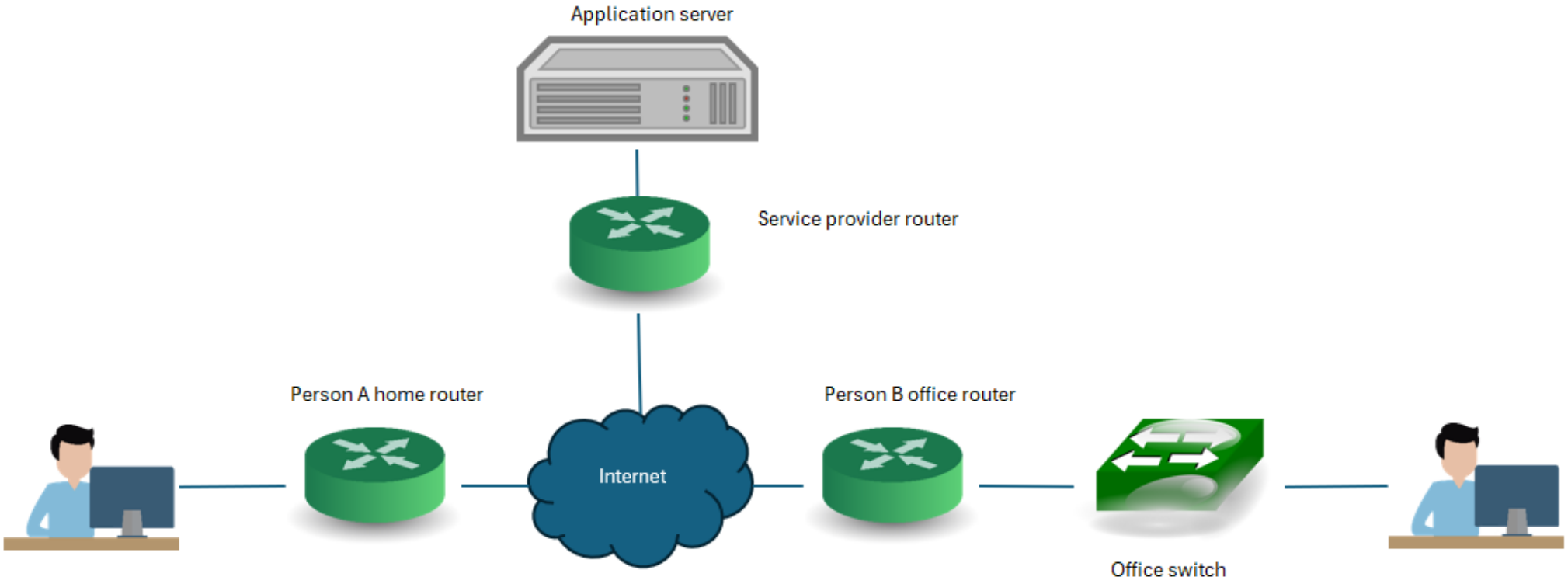
SAVONIA

Exercise 1 :

- Think about all steps what happens when a datagram (packet) travels from an application in a computer device to another application to another computer device through a router.
- Write down in groups what kind of information is transferred in every step.
- Padlet link: Tähän linkki

SAVONIA

Case study – Application to application communication



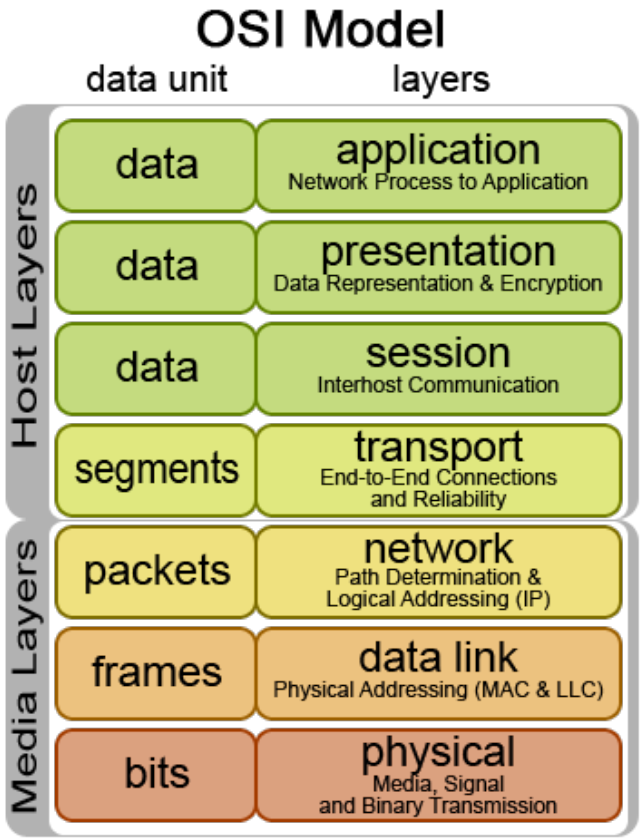
SAVONIA

What to consider in cybersecurity from point of view of networks?

1. Map out the whole journey of the intended packet.
2. Identify physical and logical route to destination.
3. Identify protocols used in communication in all steps.
4. Identify how the utilized protocols work.
5. Map out the bottlenecks where the protocols can be abused.
6. Research how to prevent abuse.
7. Implement necessary features to mitigate risk.

SAVONIA

OSI-Model – Also modeling security



Picture: https://commons.wikimedia.org/wiki/File:Osi-model.png Dino.korah

OSI-Model – Physical layer

- Determines how devices are connected physically.

- Security concerns:

    - Access to cabling – Can someone exchange cable into another?

    - Wiretapping wireless communication – Promiscuous interface configuration can listen to other devices in same medium.

SAVONIA

OSI-Model – Data Link Layer

- Determines the protocols on how devices communicate through physical mediums.

- Security concerns:

  - Device impersonation

  - Traffic interception

  - MAC-table flooding

SAVONIA

OSI-Model – Network layer

- Determines the protocols on how packets are routed between devices and interfaces.
- Security concerns:
  - Sender or receiver address rewriting.
  - Routing table poisoning.
  - Routing protocol information interception and modification.

SAVONIA

OSI-Model – Transport layer

- Determines the protocols on how packet transportation is done after a connection is established.

- Security concerns:

  - Man-in-the-middle.

  - Packet interception and modification.

  - Session hijacking.

SAVONIA

OSI-Model – Application/Session/Presentation layers

- Determines the business logic of the application on how to process, send and receive network datagrams.

- Security concerns:

  - Third party software package-vulnerabilities.

  - Bugs, self made software vulnerabilities.

  - Supply chain injection.

SAVONIA

”Everything is an injection”

- Every time a datagram (packet) is…
  - Sent, received, forwarded or processed

The devices responsible for the aforementioned action is potentially capable of…

- Modifying it’s contents.
- Redirecting it to another destination.
- Blocking it.

SAVONIA

What information in datagram can be modified

- Sender address (IP, MAC, port)
- Receiving address (IP, MAC, port)
- Sequence numbers (i.e. TCP/IP sequence)
- Flags (i.e. VLAN tags, state change bits)
- Payload (data to be processed in application)

SAVONIA

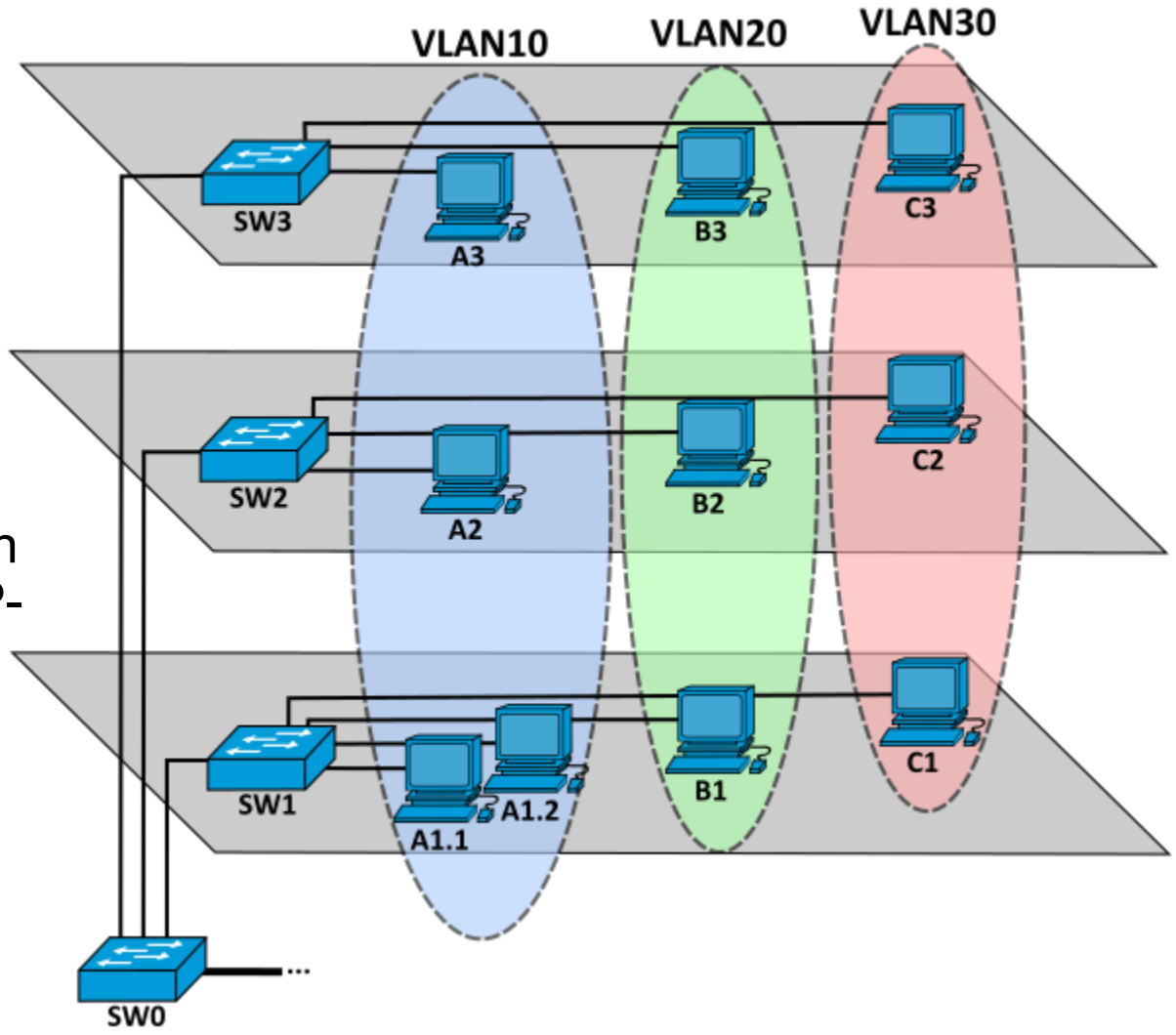
When information in protocol datagram can be modified

- Datagram can be intercepted…
  - **before** processing the received packet.
  - **after** processing the packet.
- Interception can also mean traffic flow duplication to a third party.
  - Captured traffic can be analyzed, modified or replayed from pcap-files.

SAVONIA

Protocol case example – VLAN's

- Tags in Ethernet-frames
  - Determine the physical path datagram takes in switching networks.
  - Because frames are below IP-addresses in OSI model, datagrams can flow to other IP-networks.



Picture: https://commons.wikimedia.org/wiki/File:VLAN_Concept.svg Michel Bakni

SAVONIA

Protocol case example:

Ethernet frame

- Source address
- Destination address
- Sequence Number.
- VLAN-tag.

802.3 Ethernet packet and frame structure

Layer	Preamble	Start frame delimiter (SFD)	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap (IPG)
Length (octets)	7	1	6	6	(4)	2	42–1500[c]	4	12
Layer 2 Ethernet frame	(not part of the frame)		← 64–1522 octets →						(not part of the frame)
Layer 1 Ethernet packet & IPG	← 72–1530 octets →								← 12 octets →

SAVONIA

Protocol case example:

ARP-protocol

- Sender address
  - Who is asking?
- Target address
  - Who are we asking?
- Operation
  - Request
  - Response

Internet Protocol (IPv4) over Ethernet ARP packet

Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	*Hardware Type (1)*																*Protocol Type (0x0800)*															
4	32	*Hardware Length (6)*								*Protocol Length (4)*								*Operation*															
8	64	*Sender Hardware Address*																															
12	96																	*Sender Protocol Address* ↴															
16	128	↪ *Sender Protocol Address (cont.)*																*Target Hardware Address*															
20	160																																
24	192	*Target Protocol Address*																															

SAVONIA

Protocol case example:

IPv4-packet

- Source address
- Destination address
- Time to Live
- Protocol information
- Identification

IPv4 header format

Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version (4)				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time to Live								Protocol								Header Checksum															
12	96	Source address																															
16	128	Destination address																															
20	160	(Options) (if IHL > 5)																															
⋮	⋮																																
56	448																																

SAVONIA

Protocol case example:

TCP/IP header

- Source Port.
- Destination Port.
- Sequence Number.
- Timestamps.
- Different flags
  - Reset-flag
  - Push-flag
  - Synchronize-flag
  - Fin-flag
  - Etc.

TCP header format[17]

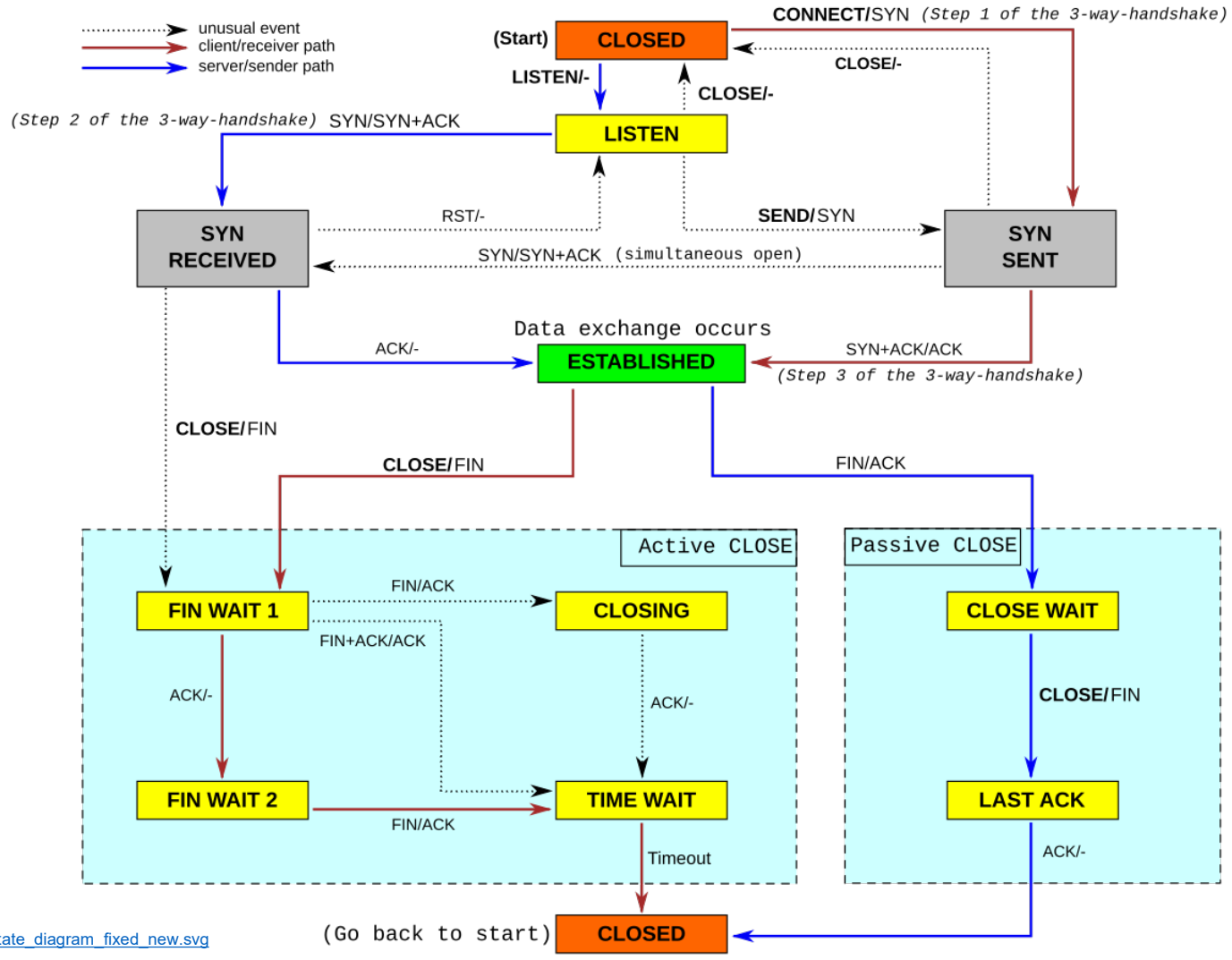
Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source Port																Destination Port															
4	32	Sequence Number																															
8	64	Acknowledgement Number (meaningful when ACK bit set)																															
12	96	Data Offset				Reserved				CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window															
16	128	Checksum																Urgent Pointer (meaningful when URG bit set)[18]															
20	160	(Options) If present, Data Offset will be greater than 5. Padded with zeroes to a multiple of 32 bits, since Data Offset counts words of 4 octets.																															
⋮	⋮																																
56	448																																
60	480	Data																															
64	512																																
⋮	⋮																																

SAVONIA

Understanding protocol states

TCP/IP state diagram

- Protocols with multiple states have state transitions.
- State transition implies message sending, receiving or processing.
- State change can be impersonated, forced or blocked.



Picture: https://en.wikipedia.org/wiki/Transmission_Control_Protocol#/media/File:Tcp_state_diagram_fixed_new.svg
Sergiodc2

SAVONIA

Understanding protocol states

TCP/IP state list

- Protocols with multiple states have state transitions.
- State transition implies message sending, receiving or processing.
- State change can be impersonated, forced or blocked.

TCP socket states

State	Endpoint	Description
LISTEN	Server	Waiting for a connection request from any remote TCP end-point.
SYN-SENT	Client	Waiting for a matching connection request after having sent a connection request.
SYN-RECEIVED	Server	Waiting for a confirming connection request acknowledgment after having both received and sent a connection request.
ESTABLISHED	Server and client	An open connection, data received can be delivered to the user. The normal state for the data transfer phase of the connection.
FIN-WAIT-1	Server and client	Waiting for a connection termination request from the remote TCP, or an acknowledgment of the connection termination request previously sent.
FIN-WAIT-2	Server and client	Waiting for a connection termination request from the remote TCP.
CLOSE-WAIT	Server and client	Waiting for a connection termination request from the local user.
CLOSING	Server and client	Waiting for a connection termination request acknowledgment from the remote TCP.
LAST-ACK	Server and client	Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP (which includes an acknowledgment of its connection termination request).
TIME-WAIT	Server or client	Waiting for enough time to pass to be sure that all remaining packets on the connection have expired.
CLOSED	Server and client	No connection state at all.

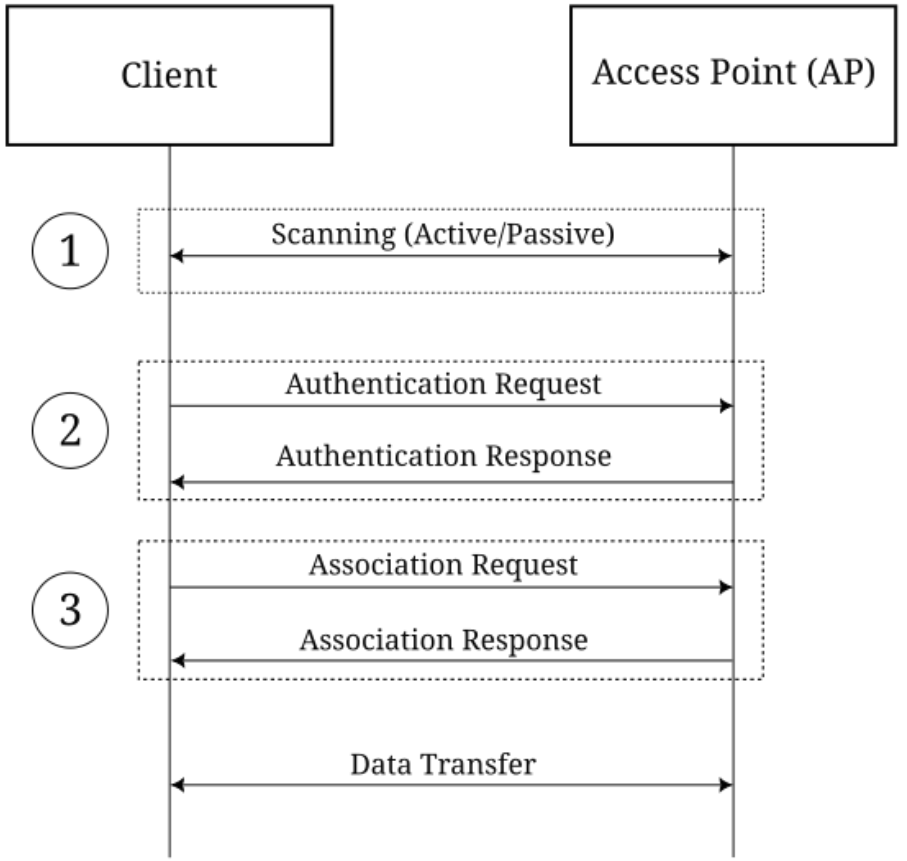
Picture: https://en.wikipedia.org/wiki/Transmission_Control_Protocol#/media/File:Tcp_state_diagram_fixed_new.svg

SAVONIA

Understanding protocol states

802.11 – WiFi connection

- Protocols with multiple states have state transitions.
- State transition implies message sending, receiving or processing.
- State change can be impersonated, forced or blocked.



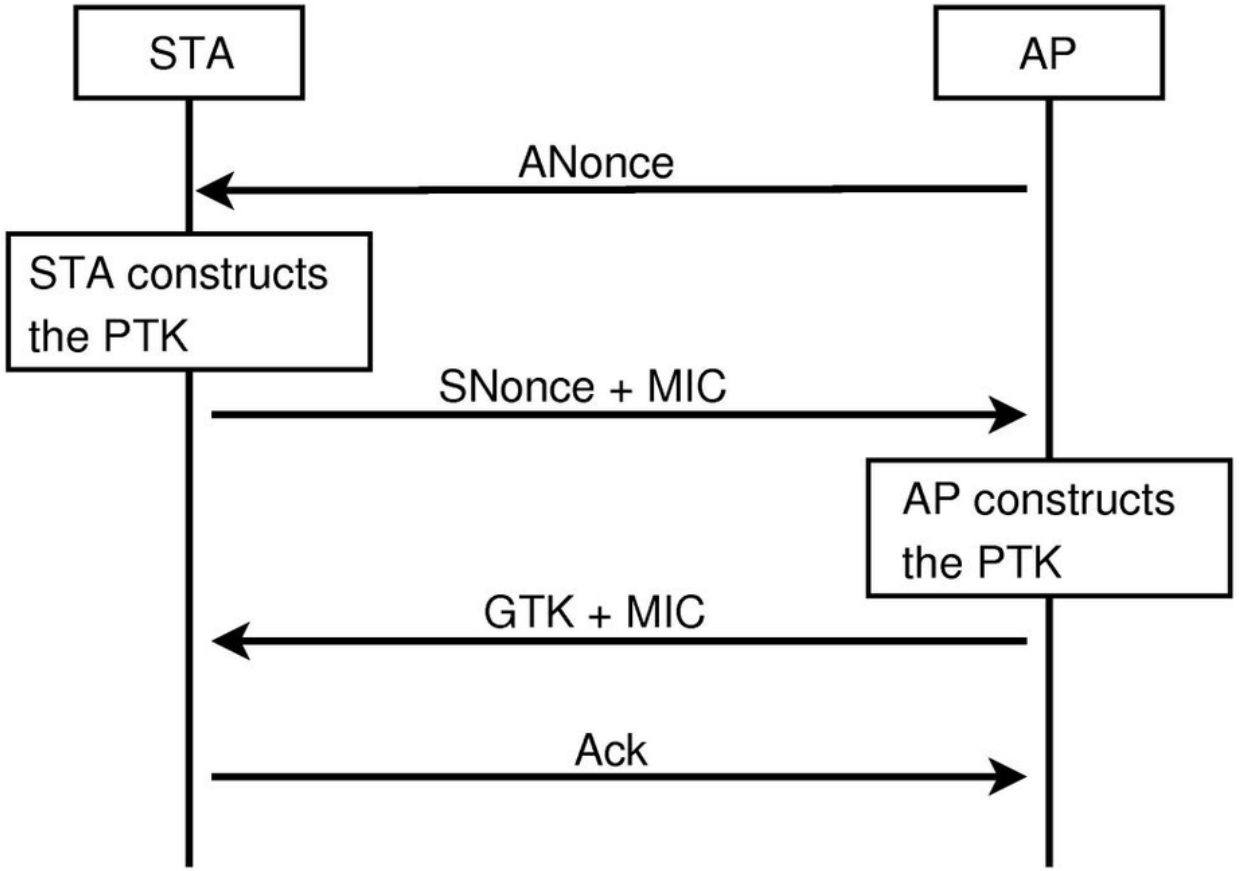
Picture: https://commons.wikimedia.org/wiki/File:802.11_Connection_Setup.svg Superspritz

SAVONIA

Understanding protocol states

802.11 – WiFi - WPA2-encryptic

- Protocols with multiple states have state transitions.
- State transition implies message sending, receiving or processing.
- State change can be impersonated, forced or blocked.

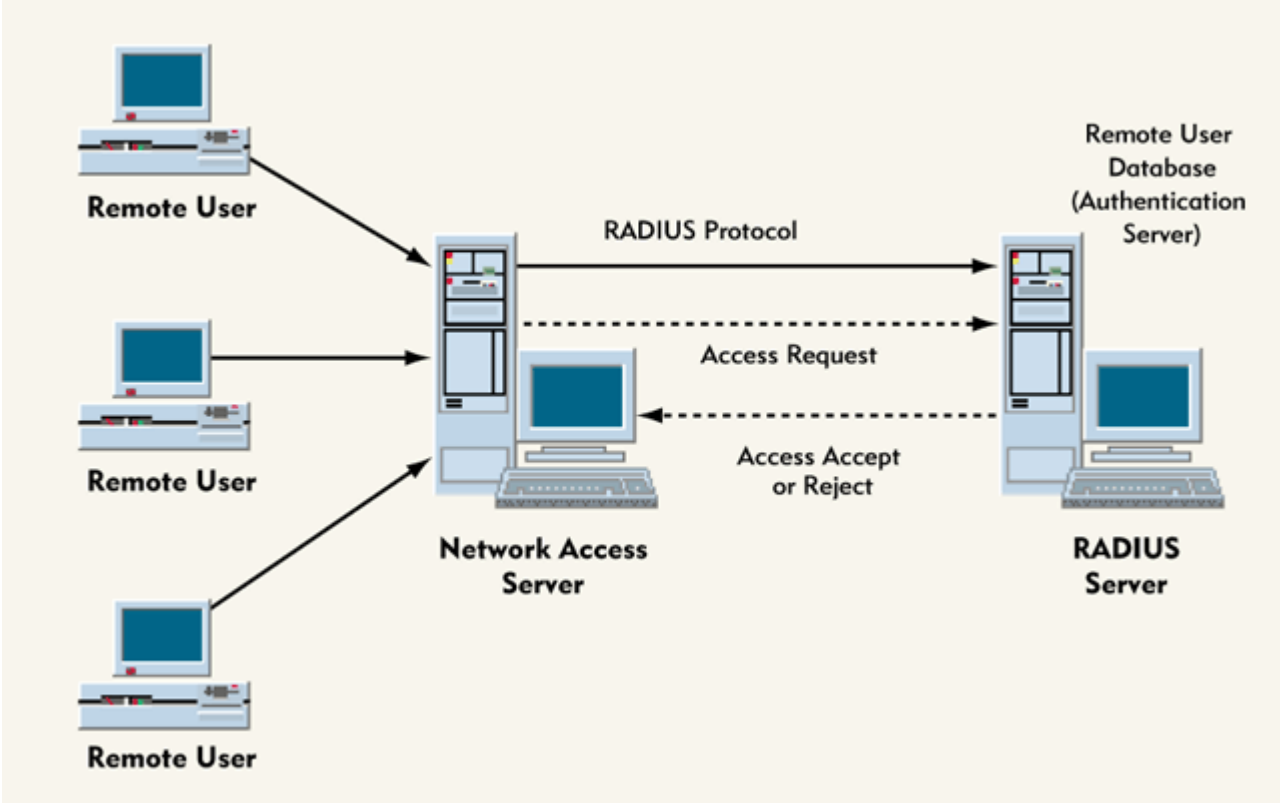


SAVONIA

Understanding protocol states

Radius authentication

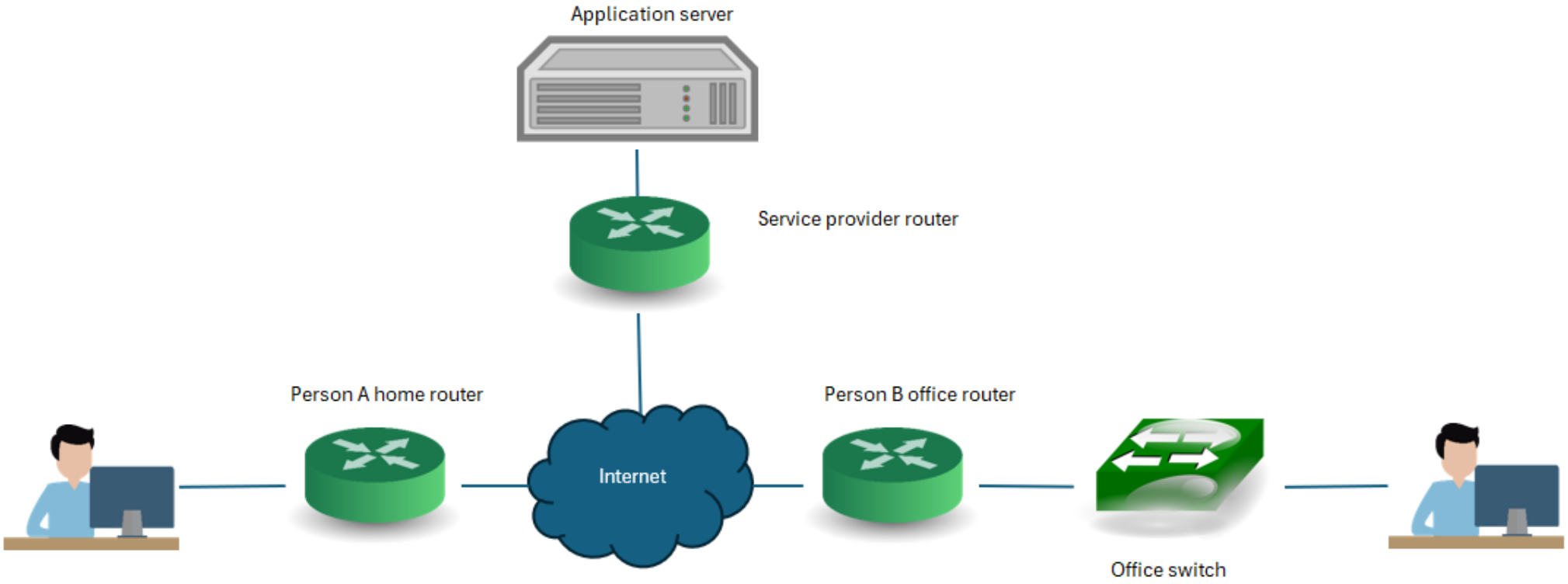
- Protocols with multiple states have state transitions.
- State transition implies message sending, receiving or processing.
- State change can be impersonated, forced or blocked.



Picture: https://commons.wikimedia.org/wiki/File:RADIUS-Server.gif Buyanzayad

SAVONIA

Case study – Application to application communication



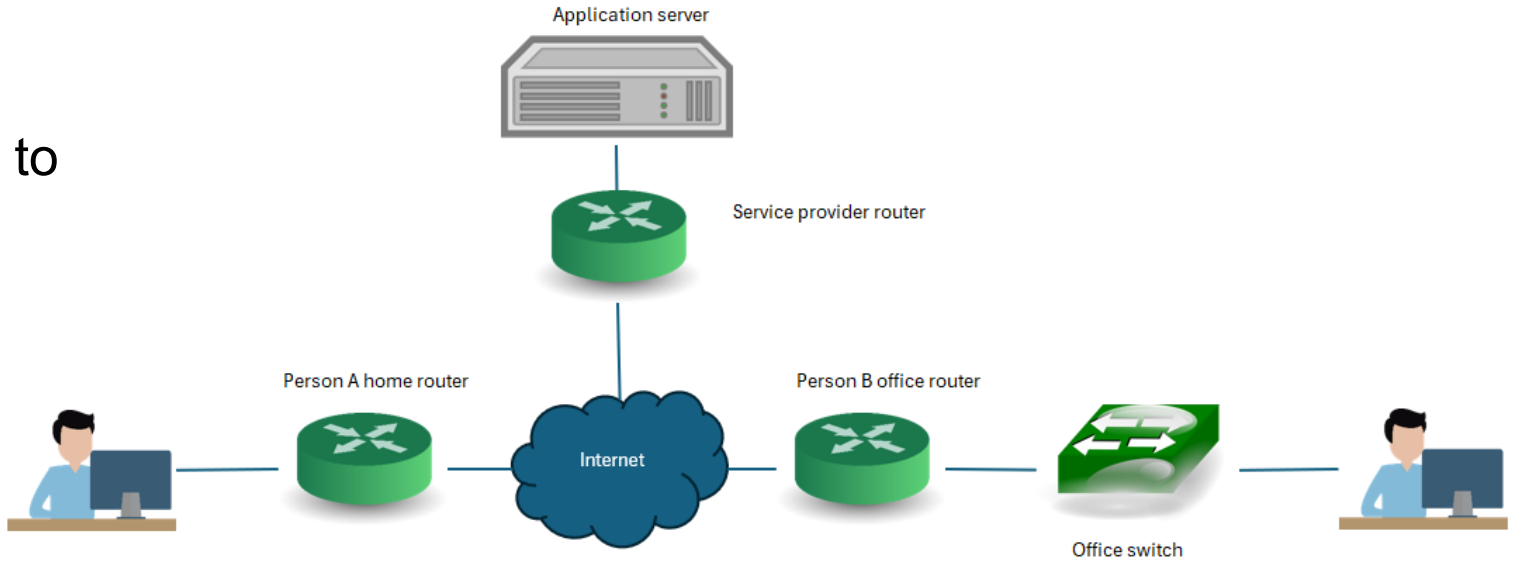
SAVONIA

Case study – Person A sending a message to Person B with messaging software

Person A sends a message to person B with messaging application.

In every step message is and/or processed so that either…

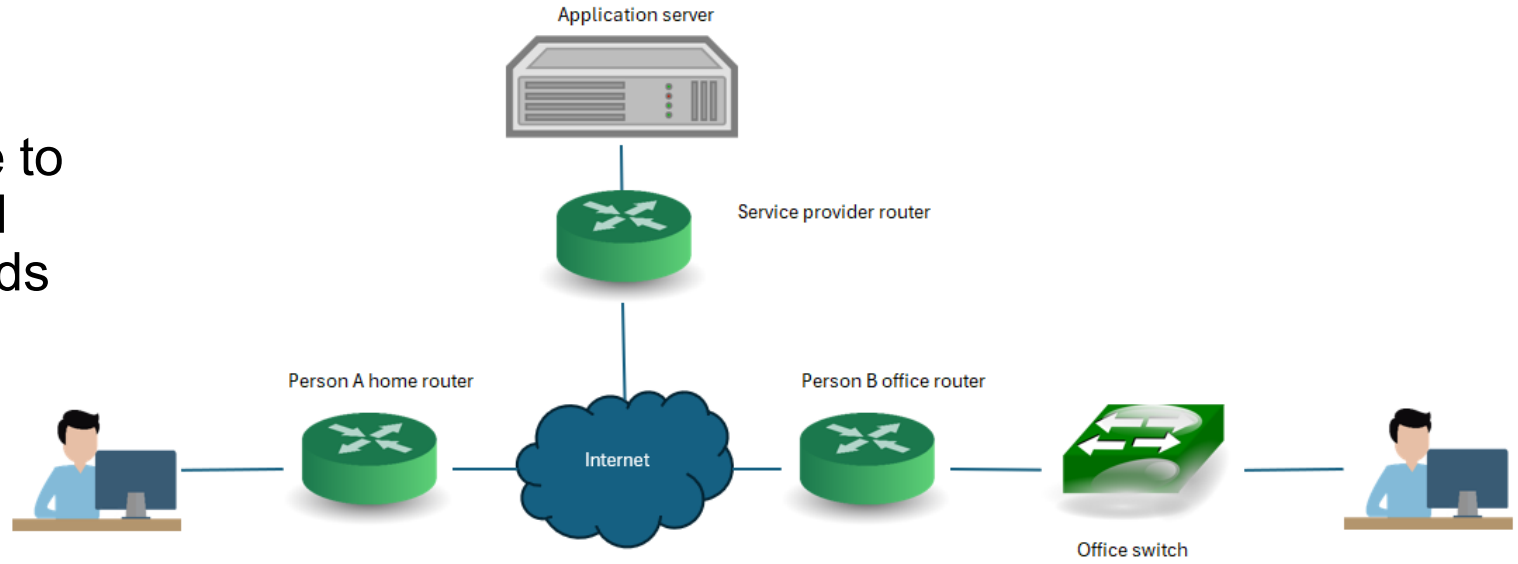
- Something is read from the packet data, header or metadata.
- Something is written to the packet data, header or metadata.



SAVONIA

Case study – Person A sending a message to Person B with messaging software

Person A sends a message to chat application server, and chat application server sends the message to Person B.



What protocols are used in every step of this diagram from left to right?

SAVONIA

In upcoming labotarory classes

- Using (Kali) linux to monitor network traffic.
- Finding network capable devices in local network.
- Searching for vulnerabilities.
- Probing services/devices found in local networks.

SAVONIA

Thank you!