# Modules

- Network-Role-Play
- Confiscation
- Keepass-As-Group

# Network-Role-Play

> [!toc] Table of Contents

This game aims to illustrate the process, but not the functionality, of network communication using the example of emails without encryption, with transport encryption, and with end-to-end encryption. It is not really a game that is meant to be fun, but rather a means of explaining the topic without having to use a network diagram, thus making it more accessible to less technical people.

# Roles

- 2x servers (systemli.org & gmail.com)
- 2x communication parties (Alice & Bob)
- 1x (or more) police officers (Eve)
- 3x "The Internet" (optional)

# Materials

- 1x sheet of paper for text
- 1x sheet of paper with email metadata
- 3x sheets of paper with IP metadata for the routes between the nodes
- 2x signs with the names of the servers
- 2x signs with the email and IP addresses of the communication parties
- 1x small box that can be locked with a padlock (large enough to hold the sheet of paper with the text)
- 3x large boxes with lids (large enough to hold the other box)
- 2x padlocks
- 3x chairs

Ideally, the sheets should be laminated and written on with whiteboard markers. This way, they can be easily reused.

# Procedure

In preparation, the Internet metadata information is stuck on each of the large boxes.

Then the roles are assigned. The role of the police should preferably be played by someone without much technical knowledge, so that creativity is required for the attacks. The rest of the people watch.

The two servers and two communication parties stand in a square. The servers and communication parties are given the signs with their information.

One chair is placed between each of the four people, on which one person playing the internet sits. They are also given the box with the corresponding internet metadata.

Alice writes a message to Bob on the sheet for the text and fills in the metadata that is not already filled in on the sheet with the metadata.

Now the various scenarios are played out. Each scenario is shown once without MITM and once with MITM (in our case by the police). The role of the police is to come up with their own ideas on how to attack the scenario. The only exceptions are attacks on Alice and Bob, which are not the aim of this game. Furthermore, the legality of the attacks or whether the parties would hand over the data to the authorities is not discussed; all technically possible attacks can be considered. The police can only attack the internet and the servers.

If the police cannot think of any way to attack, the audience can help. If they also have no ideas, the moderator can help.

The audience should then explain what happened, whether the attack worked, and what data the police obtained.

# Unencrypted

Anna gives the sheets with the text and the email metadata to the Internet, which gives them to the first server, which sends them back to the Internet, which sends them to the second server, which sends them back to the Internet, which finally sends them to Bob. At each node, the sheets are placed in the box with the corresponding IP metadata.

## Unencrypted - MITM

Possible targets of attack are:

- The internet
- The servers

All data can be intercepted at both points.

# Transport encrypted

This time, the boxes are “locked” with lids. Although these boxes are not locked in the game, it is pointed out that they should still be considered secure. However, they only provide protection during transport; the nodes must be able to open the corresponding boxes.

Otherwise, it works the same as in the unencrypted scenario. It is important to ensure that both sheets are taken out of the box at each node and then placed in the appropriate other box. This is necessary because the servers need the metadata to know where to forward the mail.

## Transport encrypted - MITM

Possible targets for attack are:

- The servers

All data can be intercepted there.

# End-to-end encryption

First, it is explained that end-to-end encryption involves a public key and a private key. We represent the public key as a padlock and the private key as the key for the lock. It is briefly pointed out that this public key must be exchanged in such a way that it is certain that it belongs to the person. For this scenario, we do this by having Bob go to Alice in person and give her the padlock.

Alice puts the piece of paper with the text in the small box, locks it with the padlock, and puts this box together with the sheet containing the metadata in the large box. She then sends it to the Internet. After that, the process is the same as before: the large box is unpacked and repacked at each node, and the small box is finally opened at Bob's end.

## End-to-end encryption - MITM

Possible targets for attack are:

- The servers

Only the metadata can be accessed there.

# End-to-end encryption with TOFU

This time, the public key is exchanged as usual by email without being verified.

1. Alice writes to Bob, “Send me the key.”
2. Bob sends the key.
3. Alice writes end-to-end encrypted as above.

## End-to-end encryption with TOFU - MITM

Possible targets of attack are:

- The servers

All data can be intercepted there.

The attack proceeds as follows:

1. Alice writes to Bob, “Send me the key.”
2. Bob sends Alice the key.
3. The police intercept the key and replace it with their own.
4. Alice encrypts the message with the police key.
5. The police intercept the message and read it.
6. The police re-encrypt the message with Bob's actual key and send it on.

Neither Alice nor Bob are aware of the attack, but the police can read everything. Because the key was replaced by the police, a second padlock is required here.

# Confiscation

> [!toc] Table of Contents

After seizures, there is often a lot of speculation about what information could potentially be found on the devices. In these moments, we remember all the little “security sins” we have committed over the years: photos, chats, contacts, etc. that were not deleted.

The shock often hits hard and is mixed with concern about what the rest of the group will say when they find out that our mistakes could now cause problems for them too.

It is precisely at these moments that we ask ourselves:

> [!quote] Why didn't we prepare better for this?! {static}

That's exactly what this module is about:

> [!success] Goals {static}
>
> - Here, we want to walk through the preparation and follow-up of a seizure of technical devices.
> - In doing so, we want to help you take possible precautions and take the necessary measures to limit damage afterwards.

# Prevention

We should think carefully about the following things in advance, as they can save us a lot of stress afterwards.

## Encryption

- Are the devices encrypted?

- Are they encrypted with strong passwords?
- Are all storage media such as USB sticks, hard drives, and SD cards encrypted with **strong passwords**?

# Passwords

- Are all passwords stored securely in a **password manager**?
- Is there a **current backup of the password database** in a secure location, so that you can recover easily, when your daily password database get's confiscated?
- Are there any passwords written down on paper lying around somewhere? If so, **destroy them**.
- Have you set up two-factor authentication on *at least all important* accounts? Especially email accounts, because they can often be used to reset passwords from other services, that you used this email for.

# Data hygiene

The less data you accumulate, the less data can be seized from you: Read the article on data hygiene:

> [!tip] {static}
>
> - When data is collected, ask yourself: “Do we really still need this data?”
> - It is not always possible to encrypt unencrypted media drives afterwards without leaving traces. Encrypt your devices from the beginning on.
>   - Data that was unencrypted may still be recovered even after deletion
>   - Deleting encrypted data is not a problem

# Signal

In Signal, you should definitely:

- Set disappearing messages be set *(e.g. 1 week)* so that as few chats as possible are stored on the device at any given time.
- Deactivate the option finding by phone number.
- Set the registration PIN.

# Backups

Confiscation means: devices and data are gone. Can you “recover” from this loss as quickly as possible, i.e., restore your data to other devices?

Making backups is annoying, but without them, you and others could suffer significant damage. Sometimes, years of work are lost because essential data/results were confiscated and there was no backup strategy.

> [!tip] {static} That's why you should make backups!

# Turn off devices

Devices are only properly encrypted when they are turned off, because after they are unlocked for the first time (immediately after booting up), the encryption key is stored in the device's RAM.

Therefore, try to turn off your devices, before they get confiscated. House searches often happen at night, or very early in the morning. Setting up auto reboot for all you devices ensures that they will be secure at every morning!

> [!tip] {static}
>
> - Switch off devices before seizure!
> - Set up Auto Reboot

# Follow-up

Now the devices have been seized and are out of reach. Have all of the above points been taken into account? If yes - Good job! But what if not?

In any case, you should contact a lawyer as soon as possible and tell them about what happened. We are not lawyers and therefore cannot give legal advice. You should also discuss the following points with them if possible.

# Evaluation

>

> [!failure] What information could have been compromised by this seizure? {static}

- Who should you report this to?
- Has your account been removed from all chat groups by your comrades so that the authorities cannot read your messages.
- Change problematic group names in Signal as quickly as possible. Only **that** the name has been changed will be visible, but not what the group was called before. This may be useless for other messengers.
- *The above last two points will only work as long as the device still has a network connection, but it doesn't hurt to try.*

> [!failure] Have passwords/accounts been compromised? {static}

- Change the relevant passwords - If you haven't already done so, set up two-factor authentication to prevent the authorities from accessing your accounts with your password.

# Restore backups

Now you will want to get your data back, which will be no problem, if you've made your backups regularly.

# Keepass-As-Group

> [!toc] Table of Contents

KeePassXC is not actually cloud software, which is one of the reasons it is so popular - none of your passwords ever leave your own computer. However, this makes it impractical for managing different logins as a group. Every time a change is made to the database, *(new password added, password changed, etc.)* everyone has to be notified of the changes so that they can apply them locally.

However, similar to cloud-based password managers such as Bitwarden, we can also keep a KeePassXC database automatically synchronized across different computers, while still opening and editing the database as usual with KeePassXC.

This requires a working cloud environment for your group to be set up. We would recommend Nextcloud, as it's not only open source, but is also hosted by various very nice tech collectives, meaning you don't have to set NextCloud up yourself. See our instructions on Nextcloud groups and device synchronization for more information on using NextCloud.

After successful installation and setup, each member of your group should have direct access to all of your cloud files locally, i.e. directly on the computer.

# Sharing the password database (and other files)

Now, the group password database, *hereinafter referred to as “the database”*, is uploaded from the **group account** to your own files:

database upload

## Share file

- Click on the `Share` icon *(or on `Details` under the three dots)*:

database upload

- Enter the name of the team in the search field and select the team *(the name may need to be written out in full)*:

database upload

- Give the group editing rights *(otherwise the team members will not be able to edit the file, which is necessary, if they should be able to add, or change passwords e.g.)*:

allow editing button

Here you can now see who has access to this resource:

sharing info of file

All team members should now have access to this database in the files of their own accounts. (See this graphic, that explains how shared files are accessed from different accounts)

# Open shared database

- In KeePassXC, select `Open database` in the bar at the top left.

open database in KeePassXC

- In the window that opens, select the shared database from the Cloud folder.

select database from file manager