

Videoconferencing Security and Privacy

An Overview of Best Practices

Klaus Möller
WP8-T1

Webinar, 28th of September 2020

Public

www.geant.org

Videoconferencing (VC) Systems

- Not only video and audio
 - Screen sharing (for presentations)
 - Collaboration (editing of documents)
 - Text chat
 - File transfer
 - Integration into task management software, etc.
- Multitude of products
 - Zoom, Webex, Whereby, Pexip, Facebook, Google Meet, Microsoft Teams, ...
- Have become indispensable tools in pandemic lock-downs

Do you hate your Firewall Administrator?



- Prank: Lets play a round of “*annoy-the-admin*”
 - Go to your network administrator when they are a bit stressed
 - Just a little tense, but still willing to answer questions
 - Ask her/him: “I need your help with getting this videoconferencing app to work over our firewall/NAT-gateway”
 - Use a currently unsupported/unknown one, of course
 - **Duck!**
- Just kidding, of course
 - No responsibility taken for lost accounts, etc. :)
 - You owe them something next sysadmin appreciation day (at least)
- But if this sounds familiar, why is it so?



Videoconferencing as seen from the Network

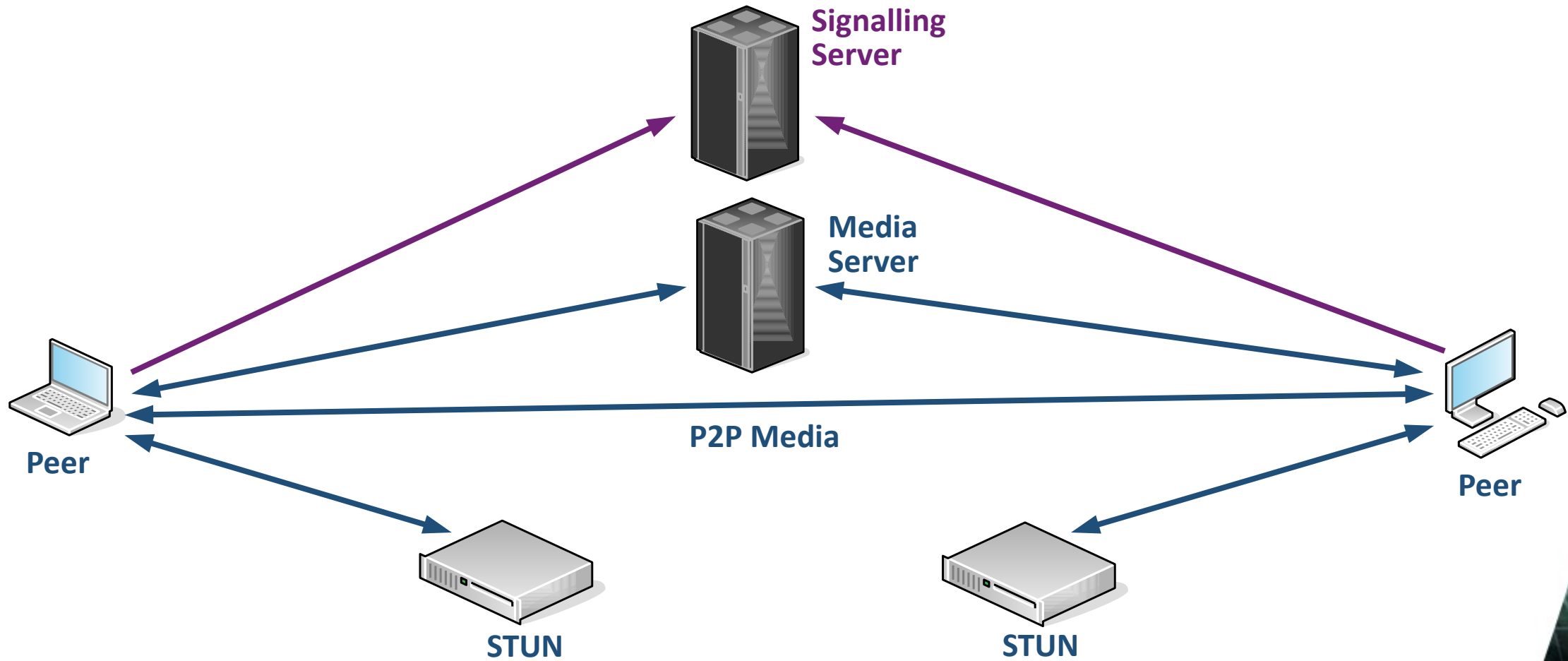
- A multitude of protocols needed
 - Call setup, control: SDP, SIP(S), H.320, H.323, ...
 - Video/Audio/Data transfer: RTP(S), RTCP(S), (S)RTSP, RTMP(S), ...
 - Resource Reservation: RSVP, DCCP, ...
 - NAT/Firewall Traversal: ICE, STUN, TURN, H.460, ...
 - More proprietary protocols
- So far, time-consuming, but doable
 - Multiply by the number of VC systems in use



Videoconferencing & Security Headaches?

- VC Data transfer protocols (RTP & Co)
 - Have to be reachable on random ports from the internet
 - No problem if participants are directly connected
- Firewalls and NAT gateways get in the way
 - Stateful Packet filters and NAT Gateways will block incoming packets
 - Unless the client has send a packet in the opposite direction **before**
 - But the client doesn't know from where a call will be coming
 - In other environments, firewall blocks all traffic, except to/from proxies
- Solution: NAT (and Firewall) traversal protocols
 - Interactive Connection Establishment (ICE)
 - Basically a combination of STUN and TURN

Videoconferencing Connection Diagram (w/o NAT)



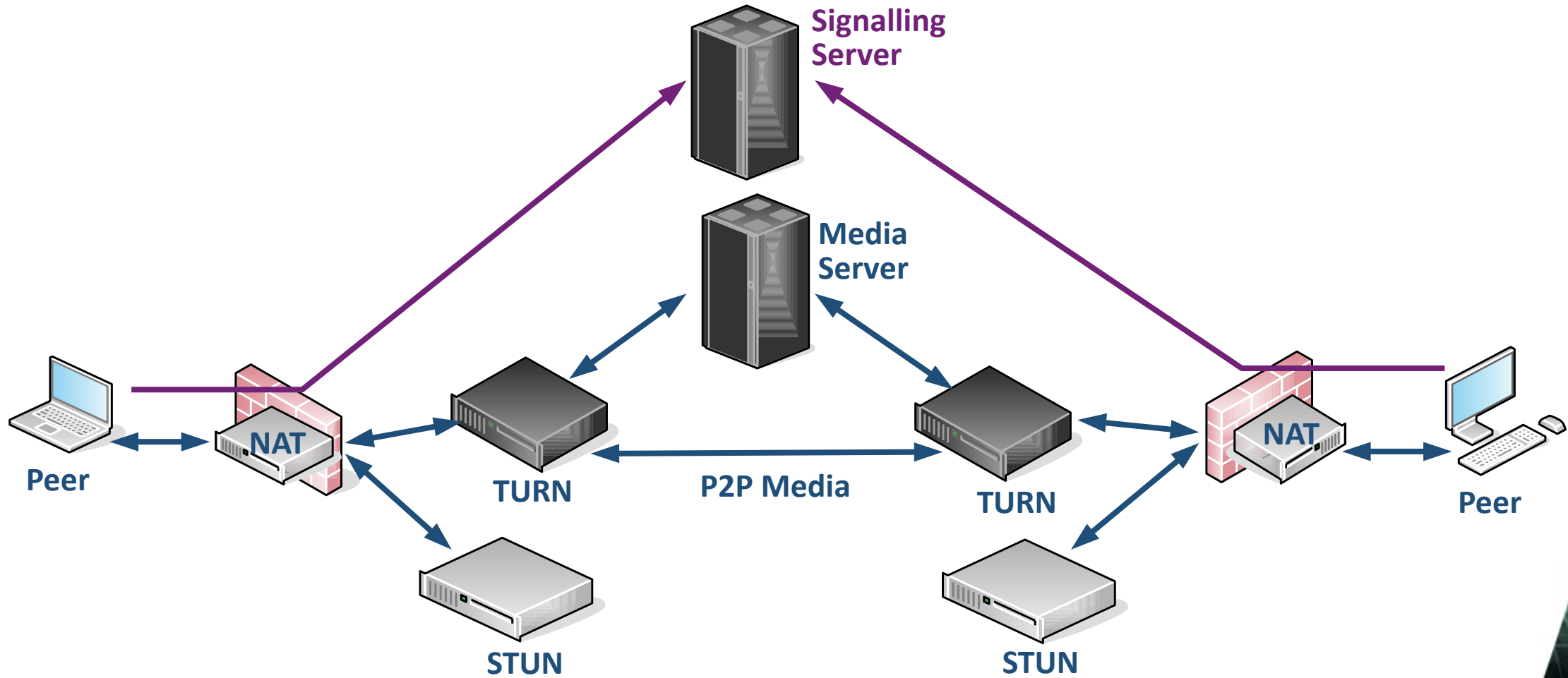
Session Traversal Utilities for NAT (STUN)

- Simplified: An elaborate way of knowing your external IP-address
 - Think of `whatsmyip.com`, but without `http(s)`
- Tells also if you can be reached at a given port number
- Insider attack scenario
 - Spoofing STUN messages to expose ports of other internal systems
 - Using IP-addresses and ports of other internal systems
 - Should not happen if STUN is authenticated properly
- Information disclosure
 - STUN server knows about your internal IP-Addresses
 - Can you trust external operators (VC provider, Google, Cloud*)?

Traversal Using Relays around NAT (TURN)

- Think of it as a sort of reverse proxy for VC clients
 - Client connects/registers with (local) TURN server
 - TURN server has public IP-address, can thus accept incoming calls
 - Often from other TURN servers
 - Knows where to relay the call to
- Usually run by the videoconferencing provider
- If run locally
 - Can be used to simplify/tighten firewall rules
 - Random ports/connections only to the TURN server
 - Internal IP-addresses stay undisclosed
 - TURN server needs to be in an unprotected network or DMZ

Videoconferencing Connection Diagram (w/ NAT)



WebRTC

- Do all the VC client stuff from the browser
- May leak internal IP-addresses through STUN
 - See browserleaks.com WebRTC leak test
- Has the same connectivity problems as a VC appliance
 - Will work on public networks (i.e. your mobile phone/tablet)
 - Or behind simple NAT gateways (i.e. your home router)
- Not needed for many VC systems
 - If so, can turn WebRTC off in the browser
 - *WebRTC Control* Extension (all browsers)
 - `about:config` in Firefox-based browsers (see References)

Videoconferencing Systems that don't use WebRTC

- These are the most common today
 - Zoom, WebEx, Whereby, ...
- Admins like them because
 - Use HTTPs for everything
 - All we need here are proper proxy settings
 - All traffic goes through their servers (no P2P)
- Things just work
- And we can pretend it's secure
 - Because HTTPS
 - And VC provider says so



Example: GÉANT Zoom Session

- As seen from a client

```
# ss -antp
State      Recv-Q  Send-Q   Local Address:Port   Peer Address:Port
ESTAB      0        0        10.0.6.4:59544       185.174.117.137:443
CLOSE-WAIT 32        0        10.0.6.4:47010       52.202.62.244:443
ESTAB      0        0        10.0.6.4:59540       185.174.117.137:443
ESTAB      0        0        10.0.6.4:59542       185.174.117.137:443
ESTAB      0        0        10.0.6.4:59538       185.174.117.137:443
ESTAB      0        0        10.0.6.4:59536       185.174.117.137:443
ESTAB      0        31       10.0.6.4:59534       185.174.117.137:443
ESTAB      0        0        10.0.6.4:59530       185.174.117.137:443

# host 185.174.117.137
137.117.174.185.in-addr.arpa domain name pointer mmr137-117.zoom.nordu.net.
# host 52.202.62.244
244.62.202.52.in-addr.arpa domain name pointer ec2-52-202-62-244.compute-1.amazonaws.com.
```

Signalling/Call setup

```
users:(("zoom",pid=12161,fd=93))
users:(("zoom",pid=12161,fd=34))
users:(("zoom",pid=12161,fd=91))
users:(("zoom",pid=12161,fd=92))
users:(("zoom",pid=12161,fd=90))
users:(("zoom",pid=12161,fd=70))
users:(("zoom",pid=12161,fd=67))
users:(("zoom",pid=12161,fd=65))
```

Relay/Streaming

Encryption

- Protects the confidentiality and integrity of exchanged data
- Most desirable: End-to-End Encryption (E2EE)
 - Streams get encrypted at the sender and decrypted by the receiver
- Second best: Transport Encryption (most often: TLS)
 - Stream is encrypted on the way to the server, but unencrypted there
- Watch out, however
 - Is encryption used by default?
 - And for what streams?
 - What crypt-algorithms/key lengths are used?
 - Forward Secrecy?
 - Key management?



What Meeting Attendees can do

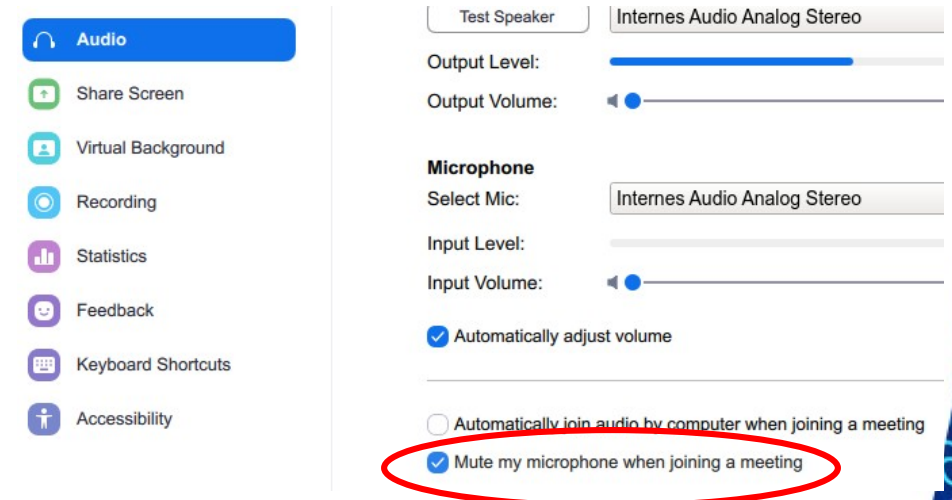
- Keep your Videoconferencing software Up to Date
- Configure your audio/video settings
- Beware of your surroundings
- Don't share Invites
- Don't make recordings

Keep the VC Software Up-to-Date

- On self-administered systems
 - Lookout for Updates regularly (at least once/week)
 - Use features from client SW for auto-updates
 - Like checking automatically when the client is started
 - Use Update channels from the OS Vendor
 - Use Update Manager if both of the above not available
 - I.e. “SUMo”, “Patch My PC”, etc.
 - Alternatively (if possible): Use Web-Client (i.e. web-browser)
 - Pro: Browsers get timely updates (~ 1/month)
 - Con: Browsers huge attack surface
- When told to patch or update: Do it as soon as possible!
- Don't forget to restart the software after updating

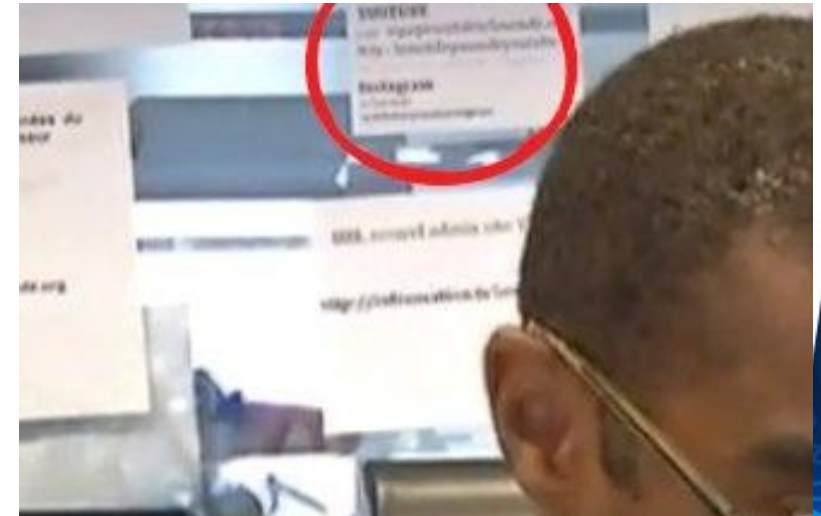
Configure the Audio/Video Settings appropriately

- When joining a video conference
- Client Preferences should be set to
 - No Video
 - Audio Mute
- Enable only when you speak
- You may accidentally leak sensitive/private information
- If you want to be extra secure
 - Cover up camera (tape/shutter)
 - Use a microphone with a true hardware *OFF* switch
 - Use headset instead of speaker



Beware of your Surroundings

- TV5Monde hack anyone?
 - French TV station got hacked in 2015
 - Password visible in background during broadcast (TV, not VC though)
 - Station was taken off the air (temporarily)
- Morale: Make sure that nothing sensitive is visible in the background
 - Empty black-/whiteboard, pinwands, monitors, etc.
 - Plain color works best, esp. with virtual backgrounds
 - Otherwise, parts of the real background may shine through
 - Beware of mirrors and mirroring surfaces behind you!
- This rule applies to Audio too!
 - Close the door/windows
 - Activate background noise reduction for the microphone (if possible)



Beware of your Screen(Background)

- To prevent leaking information during screensharing
- Limit Sharing to one application window
- Share whole desktop only when needed
 - I.e. (full-screen) presentations
 - Multi window demonstrations
- Put all shared programs on one screen/desktop
 - When using multiple screens/virtual desktops
- Close all other programs not needed during the conference
 - You may accidentally expose data on these windows
 - Including window titles
 - Taskbar shows info/preview on mouseover
 - Even for windows on other screens/desktops



Don't share Invites

- When sharing invites publicly, it's like inviting everybody
- OK, then everybody can come
 - Inevitably, people will come that don't belong here
 - C.f. "Zoombombing"
- Ask the organizer if you want to bring along others

Don't take screenshots

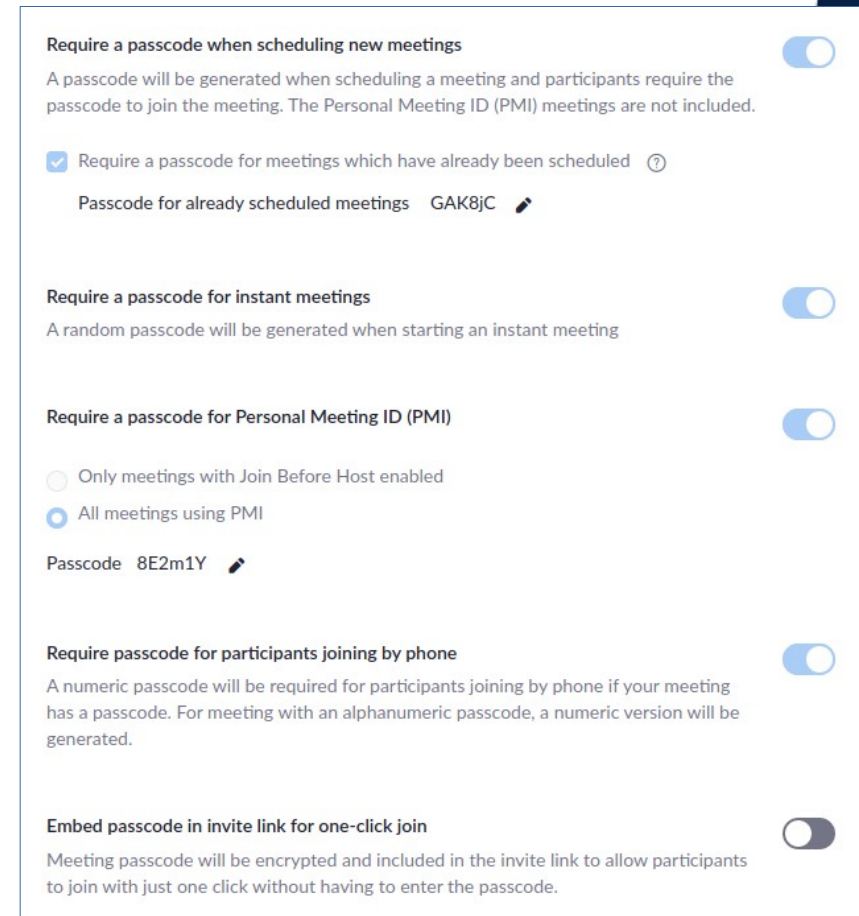
- Or make audio/video recordings, or record text chats
- Likely against the law if meeting is non-public
 - Consequences will depend on the legislation in your country
 - But this is usually not a petty offense
- Other laws and disciplinary action may also apply
 - Violation of privacy laws will probably be your least problem
- Yes it can be done without the others noticing
 - When you use other tools than the VC client
- Ask for permission (opt-in of all attendees)
 - One attendee who does not consent may mean you're not allowed to
- No streaming/recording without permission either

What Meeting Organizers can do

- Everything an Attendee should do
- Require authentication for joining meetings
- Review attendees
- Eliminate disruptors
- Lock the conference
- Inform if recording
- Disable screenshots/recordings

Require Authentication for Conferences

- When setting up a conference, require passwords or other authentication
- Room numbers may not be very random
 - And a name like “*my meeting*” certainly is guessable
 - Zoom meeting room numbers used to be 9 digits
 - Attackers have no trouble trying a couple of billion numbers
 - Scripts & botnets
- If the name/id/number is public, choose a **good** password
- Consider distributing conference link and password through different channels

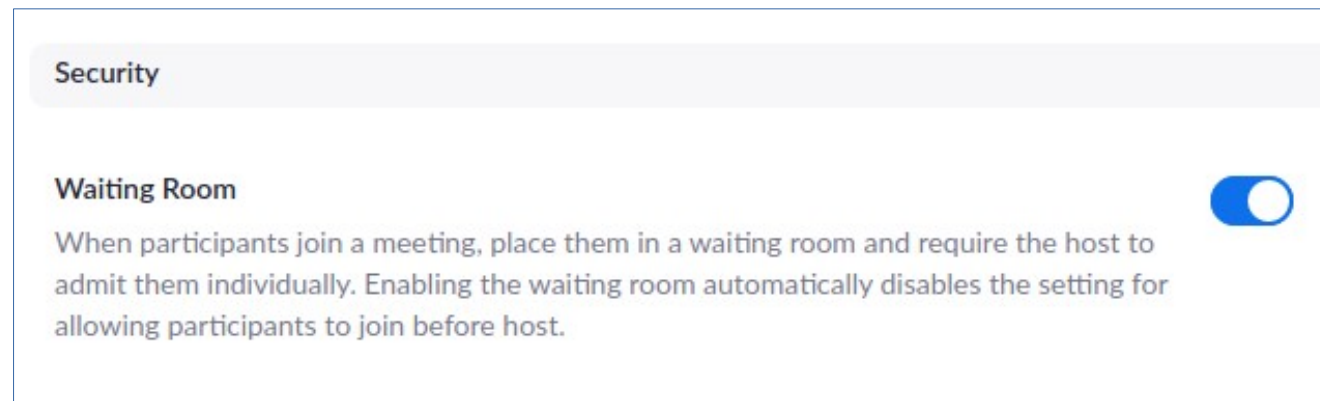


The screenshot shows the Zoom meeting settings for authentication. It includes the following options:

- Require a passcode when scheduling new meetings** (toggle on): A passcode will be generated when scheduling a meeting and participants require the passcode to join the meeting. The Personal Meeting ID (PMI) meetings are not included.
- Require a passcode for meetings which have already been scheduled** (toggle on): Passcode for already scheduled meetings: GAK8JC
- Require a passcode for instant meetings** (toggle on): A random passcode will be generated when starting an instant meeting.
- Require a passcode for Personal Meeting ID (PMI)** (toggle on):
 - Only meetings with Join Before Host enabled
 - All meetings using PMIPasscode: 8E2m1Y
- Require passcode for participants joining by phone** (toggle on): A numeric passcode will be required for participants joining by phone if your meeting has a passcode. For meeting with an alphanumeric passcode, a numeric version will be generated.
- Embed passcode in invite link for one-click join** (toggle off): Meeting passcode will be encrypted and included in the invite link to allow participants to join with just one click without having to enter the passcode.

Review Attendees

- Make sure that no one is on the list that is not invited
- Visually identify known persons (for sensitive matters)
- If there's someone you don't know – ask what she's doing here
- Throw them out, unless they have reason to be here
- Careful though, esp. with shared meeting spaces
 - You had reserved **this** room for **this** time slot, right?





Eliminate Disruptors / Lock the Conference

- If people misbehave in a conference – kick them out
 - Ban them from re-joining (if supported)
 - Or revoke their credentials
- Locking: Nobody can enter the conference anymore
 - Keeps out unwanted attendees
 - Prevents kicked/banned attendees from re-joining
 - However: late-comers can't join
 - Nor can those who had network problems and disconnected
 - You may want to watch your IM or E-mail to let them back in

Inform beforehand if recording/streaming

- Otherwise, see attendees a few slides before
- Get consent!
- Good practice:
 - Tell why you are recording
 - What will be done with the recording
 - How long will the recording be kept
 - Where will it be kept
 - Who will have access to it
 - Etc.
- If the recording contains sensitive data, secure access/storage
 - Authentication, Encryption

Recording disclaimer 

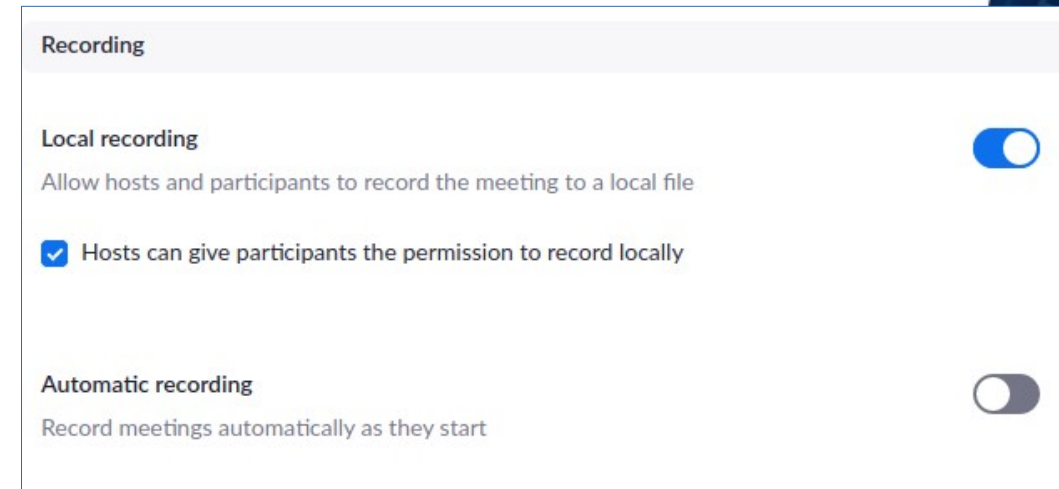
Show a customizable disclaimer to participants before a recording starts 

Ask participants for consent when a recording starts

Ask host to confirm before starting a recording

Disable Screenshots/Recordings

- Organizer can disable screenshots/recordings for attendees
- Affects only the client software,
- Attendees may record with other tools
- However, it makes the policy clear
- Also: wipe shared whiteboards, etc.
 - After the meeting ends
 - If on providers VC cloud space
- If third-party tools are required, inform beforehand about it
 - So needed programs/browser-extensions can be tested/reviewed



What have you learned?

- Why configuring networks for videoconferencing may be hard
- What to look for when selecting a good VC software/provider
- What you have to do to keep video conferences secure
 - As an organizer
 - As an attendee

Thank you

Any questions?

Next module: Office Security & Privacy

30th of September 2020

www.geant.org



References

- “A Norwegian school quit using video calls after a naked man ‘guessed’ the meeting link” <https://techcrunch.com/2020/03/26/norwegian-school-whereby>
- Tips for Secure Video Conferencing (Zip file to download) <https://www.sans.org/sites/default/files/2020-04/Video%20Conference%20Tips%20Rev.zip>
- SuMO Update Manager: <http://kcsoftwares.com/index.php?sumo>
- Patch My PC Update Manager: <https://patchmypc.com/>
- WebRTC settings for Firefox: <https://wiki.mozilla.org/Media/WebRTC/Privacy>
- WebRTC Control Extension: <https://mybrowseraddon.com/webrtc-control.html>
- Real-Time Messaging Protocol (RTMP)/ Secure RTMP (RTMPS) Specification, V1.0, <https://www.adobe.com/devnet/rtmp.html>

RFCs

- Session Initiation Protocol (SIP), Secure SIP(SIPS), RFC 3261: <https://tools.ietf.org/html/rfc3261>
- Session Description Protocol (SDP), RFC 4566: <https://tools.ietf.org/html/rfc4566>
- Real-time Transport Protocol (RTP), RFC 3550: <https://tools.ietf.org/html/rfc3550>
- Secure RTP (RTPS), RFC 3711: <https://tools.ietf.org/html/rfc3711>
- RTP Control Protocol (RTCP), RFC 3550: <https://tools.ietf.org/html/rfc3550>
- Secure RTCP (RTCPS), RFC 3711: <https://tools.ietf.org/html/rfc3711>
- Real Time Streaming Protocol (RTSP), RFC 7826: <https://tools.ietf.org/html/rfc7826>
- RTSP over TLS (RTSPS), RFC 7826: <https://tools.ietf.org/html/rfc7826>
- Resource ReSerVation Protocol (RSVP), RFC 2205: <https://tools.ietf.org/html/rfc2205>
- Datagram Congestion Control Protocol (DCCP), RFC 4340: <https://tools.ietf.org/html/rfc4340>
- Session Traversal Utilities for NAT (STUN), RFC 5389: <https://tools.ietf.org/html/rfc5389>
- Traversal Using Relays around NAT (TURN), RFC 5766: <https://tools.ietf.org/html/rfc5766>
- Interactive Connectivity Establishment (ICE), RFC 8445: <https://tools.ietf.org/html/rfc8445>

Standard Port Numbers for SIP, STUN & TURN

- SIP: 5060 UDP & TCP
- SIPS: 5061 UDP & TCP
- STUN, TURN: 3478 UDP & TCP
- STUN, TURN over D(TLS): 5349 UDP & TCP