EMPFEHLUNG: IT IM UNTERNEHMEN

Sicherer Einsatz von Jitsi Meet

Die heutige Kommunikation verlagert sich immer stärker von der Nutzung der herkömmlichen Telefoninfrastruktur in das Internet. Technologien wie Voice over IP (VoIP) sind allgegenwärtig und bieten durch Verwendung der bestehenden IT-Infrastruktur die Möglichkeit, Telefonanrufe annähernd überall und mit für jeden verfügbaren Mitteln zu führen.

Nicht zuletzt durch die COVID-19-Pandemie ist auch der Bedarf an Videotelefonie sowie Videokonferenzen stark angestiegen und das Angebot an Videokonferenzlösungen ist größer geworden. Diese Angebote unterscheiden sich in Funktionsumfang aber auch Auswirkungen auf die IT-Sicherheit. Ein besonderes Augenmerk liegt dabei auf der Frage, wo die Inhalte von Gesprächen und Videodaten verarbeitet werden und ob dort die erforderlichen Voraussetzungen vorliegen, um deren Vertraulichkeit zu gewährleisten.

Wird das Videokonferenzsystem auf eigenen Systemen betrieben oder individuell bereitgestellt, bleibt die Hoheit und Kontrolle über die anfallenden Daten in vollem Umfang in der Verantwortung der betreibenden Organisation erhalten. Im Gegensatz dazu ist bei Nutzung einer Cloud-basierten Lösung (z.B. Software-as-a-Service) die Verantwortung für Sicherheit und Betrieb zwischen nutzender Organisation und Diensteanbietenden verteilt.

1 Ziel

Dieses Dokument zeigt am Beispiel der freien und quelloffenen Videokonferenzsoftware Jitsi Meet[1] auf, wie ein selbstverwaltetes Videokonferenzsystem konfiguriert sowie betrieben werden kann und welche Aspekte der IT-Sicherheit beachtet werden sollten.

Dabei werden zunächst die Rahmenbedingungen und Überlegungen vor einer Installation von Jitsi Meet dargestellt. Anschließend werden Einstellungen aufgezeigt, die besondere Auswirkungen auf die IT-Sicherheit beim Betrieb von Jitsi Meet entfalten können. Abschließend wird auf Maßnahmen eingegangen, die beim dauerhaften Betrieb zu beachten sind.

Bei sorgfältiger Prüfung und Umsetzung der in diesem Dokument dargestellten Hinweise und Konfigurationsempfehlungen sollte eine sichere Installation der Videokonferenzsoftware Jitsi Meet entstehen, die auf einer geeigneten Plattform die weitgehende Kontrolle und Vertraulichkeit der Gesprächsinhalte gewährleistet.

[1] https://jitsi.org/jitsi-meet

2 Sichere Konfiguration

Im Folgenden werden Hinweise zur Installation und sicheren Konfiguration von Jitsi Meet im Eigenbetrieb gegeben. Die Ausführungen orientieren sich dabei an der zum Zeitpunkt der Erstellung dieser Empfehlung verfügbaren Jitsi Meet Version 2.0.9584. Andere Softwareversionen können sich ggf. nicht so wie beschrieben verhalten oder abweichende Konfigurationsoptionen aufweisen. Für ein grundlegendes Verständnis der Funktionsweise von Jitsi Meet sowie der zugehörigen Komponenten sei auf die entsprechenden Erläuterungen im Handbuch[2] der Entwickelnden verwiesen.

Auf die Konfiguration von Jitsi as a Service (JaaS)[3] und von öffentlich angebotenen Jitsi-Installationen (als Platform as a Service, kurz: PaaS) sowie die IT-Sicherheit von frei zugänglichen Jitsi-Meet-Servern wird im Rahmen dieses Dokumentes nicht eingegangen.

2.1 Ressourcenbedarf

Bevor mit der Installation von Jitsi Meet begonnen wird, ist es erforderlich, die Dimensionierung der benötigten Ressourcen anhand der Anforderungen zu ermitteln und festzulegen. Dabei ist es entscheidend abzuschätzen, welche Anzahl von parallel teilnehmenden Nutzenden für Telefon-/Videokonferenzen zu erwarten sind.

Die besonders begrenzenden Faktoren für ein Videokonferenzsystem sind die dem Serversystem zur Verfügung stehende Bandbreite (z. B. über das Internet) und die verfügbare Rechenleistung.

In einem Lasttest[4] haben die Entwickelnden auf einem nativen System mit Intel Xeon CPU (E5-1620 v2 @ 3,70GHz) dargestellt, dass Jitsi Meet bei

- 90 parallelen Videostreams eine Bandbreite von 47,6 Megabits und 3,1 % CPU-Last,
- 380 parallelen Videostreams eine Bandbreite von 199,4 Megabits und 8,0 % CPU-Last,
- 1056 parallelen Videostreams eine Bandbreite von 550,4 Megabits und 20,3 % CPU-Last

nutzt.

Daher sollte zunächst auf Grundlage des Bedarfes der maximalen Anzahl von parallelen Videokonferenzen sowie parallel Teilnehmenden entschieden werden, welche Serverleistung eingeplant werden muss und geprüft werden, ob die benötigte Bandbreite zur Verfügung steht. Im Zweifel empfiehlt es sich, den Server und die Bandbreite eher größer als kleiner zu dimensionieren.

Bei der Internetverbindung ist zudem zu beachten, dass die Bandbreite synchron zur Verfügung stehen muss, d.h. Up- und Downloadbandbreite unabhängig voneinander sind und in der genannten Höhe zur Verfügung stehen. Eine in Deutschland übliche Internetanbindung für den Heimgebrauch ist zumeist asymmetrisch und wird sich in der Regel nur sehr eingeschränkt für einen zuverlässigen und stabilen Betrieb eines Jitsi-Meet-Servers eignen.

In der Konfiguration von Jitsi Meet lassen sich darüber hinaus noch weitere Einstellungen vornehmen, mit denen sich eine Verminderung von Bandbreiten- und CPU-Last sowohl auf dem Server als auch den Clients erzielen lassen. Weitere Informationen dazu können dem Abschnitt 2.8.3 Reduktion der Datenübertragungen entnommen werden.

[2] https://jitsi.github.io/handbook/docs/architecture
[3] https://jaas.8x8.vc/
[4] https://jitsi.org/jitsi-videobridge-performance-evaluation/

2.2 Betrieb in Virtuellen Maschinen/Containern

Jitsi Meet lässt sich auf einem Server auf verschiedene Arten betreiben:

- direkt auf dem System installiert (sog. „Bare-Metal“)
- in einer separaten Virtuellen Maschine (VM) installiert
- auf einer vertrauenswürdigen Containerisierungsplattform (z.B. Linux Containers, kurz: LXC)

Die Lösungen bieten verschiedene Vor- und Nachteile und unterscheiden sich hauptsächlich in Flexibilität und Sicherheit. Gängige Praxis ist es jedoch, die Funktionen eines Servers von anderen, nicht damit zusammenhängenden Funktionen und Diensten, zu separieren. Im Zweifel sollte sich daher eher für die Realisierung in einer VM oder in einem Container entschieden werden.

Damit wird die Möglichkeit erlangt, Jitsi Meet und die zugehörigen Programme und Bibliotheken eng beisammen zu halten und bei Bedarf auch einfach und rückstandslos wieder vom System entfernen zu können. Darüber hinaus können VMs und Container separat aktualisiert und so vermieden werden, dass Interferenzen und Versionskonflikte zu anderer Software entstehen. Zusätzlich existieren einfache Möglichkeiten für Datensicherungen, Parallelbetrieb mehrerer Instanzen sowie eine Härtung gegenüber Angreifenden, falls es diesen gelingen sollte, Teile des Systems zu übernehmen.

Für Jitsi Meet existieren bereits vorgefertigte Docker-Container[5] auf Docker-Hub[6], die eine schnelle und einfache Installation ermöglichen, welche allerdings nicht für einen produktiven Einsatz zu empfehlen sind, da die Herstellung der Images nicht überprüft werden kann.

2.3 Vertrauenswürdige Quellen

Die Installation von Software aus Drittquellen ohne weitere Prüfung oder aus nicht vertrauenswürdiger Herkunft birgt Risiken, da dadurch fehlerhafte oder inkompatible Software oder aber auch Schadsoftware installiert werden kann.

Wie bei jeder Software sollte daher auch bei Jitsi Meet darauf geachtet werden, dass sowohl bei der Installation als auch bei Aktualisierungen die Daten aus vertrauenswürdigen Quellen bezogen werden und deren Integrität vor der Nutzung verifiziert wird.

Als Quellen im Kontext Jitsi Meet können dabei vor allem

- die Webseite der Entwickelnden: https://jitsi.org,
- die Daten auf GitHub: https://github.com/jitsi/jitsi-meet,
- die offizielle Jitsi-Meet-App aus den bekannten App Stores[7,8,9] und
- die offizielle Jitsi Meet-Desktopanwendung für Windows, macOS und Linux (AppImage sowie .deb-Paket), basierend auf dem Electron-Framework[10], genutzt werden.

Hinter dem Jitsi-Projekt steht das US-amerikanische Unternehmen 8x8, Inc., die auch die mobilen Apps über die App Stores bereitstellen.

[5] https://www.docker.com
[6] https://www.docker.com/products/docker-hub/
[7] https://play.google.com/store/apps/details?id=org.jitsi.meet&hl=de&gl=DE
[8] https://apps.apple.com/de/app/jitsi-meet/id1165103905
[9] https://f-droid.org/de/packages/org.jitsi.meet/
[10] https://github.com/jitsi/jitsi-meet-electron

Für Debian/Ubuntu werden darüber hinaus separate Paketquellen[11] angeboten, über die sich die Verfügbarkeit aktueller Jitsi-Meet-Pakete über die native Paketverwaltung sicherstellen lässt. Hierzu ist die Aufnahme in die Paketverwaltung des Betriebssystems erforderlich. Der hierfür benötigte Repository-Key[12] sollte nur von der Webseite der Entwickelnden heruntergeladen werden. Daneben können fertige Softwarepakete für Ubuntu und Debian bezogen werden oder der Softwarequellcode selbst heruntergeladen und kompiliert werden.

2.4 Installation

Bei der Installation von Jitsi Meet werden auch weitere Softwareprodukte installiert. Hierbei handelt es sich beispielsweise um den Webserver nginx[13]. Alternativ kann Jitsi Meet aber auch über einen vorab installierten Apache-Webserver[14] betrieben werden. Bei diesen weiteren Softwareprodukten sind ebenfalls die o.g. Grundsätze zu beachten. Als vertrauenswürdige Quellen hierfür können beispielsweise die Distributoren, wie Debian Project oder Red Hat dienen.

Für versionsaktuelle Hinweise zu diesem Thema und Schritt-für-Schritt-Anleitungen empfiehlt es sich, sich an der Dokumentation[15] der Entwickelnden zu orientieren und anhand dieser die Installation von Jitsi Meet vorzunehmen.

2.5 Konfigurationsdateien und -pfade

Nach der Installation sollte die Konfiguration von Jitsi Meet sowie die der weiteren Komponenten bearbeitet und alle Einstellungen den Bedürfnissen und Anforderungen angepasst werden.

Eine Auswahl wichtiger Konfigurationen wird im Folgenden betrachtet. Die dabei zur Verwendung kommenden Dateien befinden sich voreingestellt unter den hier aufgelisteten Dateipfaden:

- Hauptkonfigurationsdatei: `/etc/jitsi/meet/<hostname>-config.js`
- Prosody-Konfigurationsdatei: `/etc/prosody/conf.avail/<hostname>.cfg.lua`
- Prosody-Plugin-Ordner: `/usr/share/jitsi-meet/prosody-plugins`
- Webserver-Konfigurationsdateien:
  - nginx: `/etc/nginx/sites-available/<domain>.conf`
  - apache2: `/etc/apache2/sites-available/<domain>.conf`

Aus Gründen der Nachvollziehbarkeit, zur Fehleranalyse und späterer, möglicher Neuinstallationen sollten alle Konfigurationsänderungen und für Jitsi Meet vorgenommene Einstellungen am Server angemessen dokumentiert werden.

2.6 Zertifikate

Um die Verschlüsselung der Daten auch über ggf. unsichere Netzwerke sicherzustellen sowie darauf vertrauen zu können, dass der angesprochene Server authentisch ist, sollte Jitsi Meet mit gültigen und sicheren TLS-Zertifikaten[16] ausgestattet werden. Während der Installation kann ausgewählt werden, ob

[11] https://jitsi.org/downloads/
[12] https://download.jitsi.org/jitsi-key.gpg.key
[13] https://nginx.org
[14] https://httpd.apache.org/
[15] https://jitsi.github.io/handbook/docs/devops-guide/devops-guide-start
[16] https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2.html

eigene Zertifikate verwendet werden sollen, ein neues selbstsigniertes Zertifikat erstellt oder ein Zertifikat über den Dienst Let's Encrypt[17] erstellt und verwendet werden soll.

Dabei kann es sich beispielsweise um selbstsignierte Zertifikate handeln, denen in der Organisationsstruktur vertraut wird oder um Zertifikate, die von einer vertrauenswürdigen Zertifikatsstelle nach entsprechender Verifikation signiert wurden. Für die Nutzung des von Mozilla bereitgestellten Dienstes Let's Encrypt existiert für Jitsi Meet bereits ein vorgefertigtes Skript[18], das die Erstellung, Signierung und Einbindung von Zertifikaten während der Installation einrichtet und in regelmäßigen Abständen automatisiert aktualisiert.

Falls ein selbstsigniertes Zertifikat oder ein von einer anderen Stelle signiertes Zertifikat verwendet werden soll, muss dieses dem Jitsi-Meet-Server noch bekannt gemacht werden.

Die Zertifikatsdatei sowie der dazugehörige private Schlüssel müssen daher in der entsprechenden nginx-Konfigurationsdatei unter den folgenden Variablen eingetragen werden:

```
ssl_certificate <Pfadangabe zur Zertifikatsdatei>
ssl_certificate_key <Pfadangabe zum privaten Teil des Zertifikats>
```

Auszug 1: /etc/nginx/sites-available/<domain>.conf

Im Falle einer Installation mit einem Apache-Webserver sind die folgenden Variablen zu setzen:

```
SSLCertificateFile <Pfadangabe zur Zertifikatsdatei>
SSLCertificateKeyFile <Pfadangabe zum privaten Teil des Zertifikats>
```

Auszug 2: /etc/apache2/sites-available/<domain>.conf

Nach Übernahme der Änderungen durch Neustart der Dienste kann anschließend mit einem Aufruf der Jitsi-Meet-Domain über den Webbrowser geprüft werden, ob der Server erreicht und das neue Zertifikat korrekt verwendet wird. Insbesondere sollte keine Sicherheitswarnung im Webbrowser auftreten, die auf ein selbstsigniertes oder nicht vertrauenswürdiges Zertifikat hinweist. Ansonsten müssen ggf. noch die zur Validierung benötigten öffentlichen Zertifikate im Webbrowser oder genutztem Client-Betriebssystem hinzugefügt werden.

2.7 Nutzendenverwaltung und Absicherung der Konferenzräume

In der Vorkonfiguration kann jeder Besuchende, der den Webserver erreichen kann, beliebigen Jitsi-Meet-Konferenzen beitreten und eigene Konferenzen erstellen. Dies kann unter Umständen ein Sicherheitsrisiko darstellen, da Personen hierdurch unbefugt an für sie nicht vorgesehene Informationen gelangen oder die Ressourcen des Servers belasten könnten.

Für diesen Fall bietet Jitsi Meet die folgenden Möglichkeiten, um den Server als ganzen oder die Konferenzräume abzusichern.

2.7.1 Einrichten einer Nutzendenverwaltung

Die Nutzendenverwaltung von Jitsi Meet kann so eingerichtet werden, dass entweder jeder Teilnehmende Zugangsdaten benötigt oder nur die Erstellenden von Konferenzen sich vor Beginn einer Konferenz als sogenannte Moderierende authentisieren müssen. Der letztere Fall stellt einen guten Mittelweg dar, um ggf. auch vorher unbekannten Teilnehmenden einen Zugriff zu ermöglichen und gleichzeitig zu verhindern, dass beliebige Besuchende eigenständige Konferenzen erstellen können. In dieser Konfiguration muss bei Eintritt in die Konferenz organisatorisch die Berechtigung zur Teilnahme geprüft werden.

[17] https://letsencrypt.org
[18] https://github.com/jitsi/jitsi-meet/blob/master/resources/install-letsencrypt-cert.sh

Versionsaktuelle Hinweise und Schritt-für-Schritt-Anleitungen zu beiden Anwendungsfällen lassen sich der Dokumentation[19] der Entwickelnden entnehmen.

2.7.2 Nutzung einer Identity-and-Access-Management-Lösung (IAM)

Jitsi kann auch an eine zentrale IAM-Lösung (beispielsweise einen Keycloak-Server[20]) angebunden werden, um zahlreiche Authentisierungsmethoden nutzbar zu machen. Dafür muss die Token-Authentifizierung[21] aktiviert und eine geeignete Adapter-Komponente (z.B. von Nordeck[22]) integriert werden. Wichtig ist hierbei nur, dass sich auch die Adapter-Komponente gegenüber dem Authentifizierungsserver authentisiert.

Damit lässt sich die Authentifizierung flexibel und feingranular gestalten. Es können dann entweder Accounts mit festen Nutzendennamen eingerichtet oder auf die Anbindung von bestehenden Verzeichnisdiensten gesetzt werden. Auch eine Anmeldung mittels Zwei-Faktor-Authentifizierung oder Passkeys[23] wird dadurch ermöglicht.

Sollen für Gäste keine eigenen Accounts angelegt werden, müssen diese eine Zugangs-URL mit validem Token[24] erhalten. Dabei ist darauf zu achten, dass jede Person mit Zugang zur URL der Konferenz beitreten kann.

Zusätzlich lassen sich über die Token-Authentifizierung weitere Einstellungen (z.B. bezüglich Anzeigename oder Lobbyfunktion) konfigurieren. Wichtig dabei ist jedoch, dass diese Optionen separat konfiguriert werden müssen und nicht allein durch die Installation der Token-Funktionalität verfügbar sind.

2.7.3 Namensgebung erzwingen

In der Vorkonfiguration von Jitsi Meet wird allen Teilnehmenden der Name „Fellow Jitster“ zugewiesen, sofern diese keinen Anzeigenamen für sich festlegen. Für andere Teilnehmende kann es somit schwer festzustellen sein, wer die Gesprächspartner sind oder ob sich ggf. fremde Personen in der Konferenz befinden.

Es kann daher hilfreich sein, die Konfiguration von Jitsi Meet dahingehend anzupassen, dass Teilnehmende vor Beitritt immer einen Anzeigenamen wählen müssen. Dies kann in der Konfigurationsdatei von Jitsi Meet geschehen, indem der Eintrag wie folgt abgeändert wird:

```
requireDisplayName: true,
```

Auszug 3: /etc/jitsi/meet/<hostname>-config.js

Falls eine Authentifizierung mittels Token aktiviert ist, wird der Nutzendenname aus dem Token übernommen. Jedoch kann dieser vor dem Beitritt noch von den Nutzenden selbst überschrieben werden. Um Verwirrung und Täuschung vorzubeugen, kann das manuelle Setzen der Nutzendennamen über folgende Einstellung deaktiviert werden:

```
readOnlyName: true,
```

Auszug 4: /etc/jitsi/meet/<hostname>-config.js

[19] https://jitsi.github.io/handbook/docs/devops-guide/secure-domain
[20] https://www.keycloak.org/
[21] https://github.com/jitsi/lib-jitsi-meet/blob/master/doc/tokens.md
[22] https://github.com/nordeck/jitsi-keycloak-adapter
[23] https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Cyber-Sicherheitsempfehlungen/Accountschutz/Passkeys/passkeys-anmelden-ohne-passwort_node.html
[24] https://github.com/jitsi-contrib/jitok

2.7.4 Nutzung von Passwörtern für Konferenzräume

Jitsi Meet lässt sich so konfigurieren, dass Teilnehmende vor Beitritt zu einer Konferenz ein Passwort eingeben müssen. Dieses Zugangspasswort kann von Moderierenden (ohne Nutzendenverwaltung ist dies die erste Person, die einen Raum betritt) frei vergeben werden.

Auf diese Weise lassen sich Konferenzen auch ohne größere Nutzendenverwaltung absichern. Die Möglichkeit für Besuchende des Webservers, beliebig eigene Konferenzen zu starten, bleibt dabei allerdings erhalten.

2.7.5 Namensgebung für Konferenzräume

Ein wesentlicher Unterschied zu anderen Videokonferenzsystemen besteht darin, dass Konferenzräume und zugehörige Zugangspasswörter in Jitsi Meet voreingestellt nicht im Voraus reserviert werden können. Ihre Namen können frei gewählt werden. Dabei sollte auf die Verwendung von Sonderzeichen verzichtet werden. Es wird kein Verzeichnis darüber geführt, welche Raumnamen bereits vergeben sind. Ein Raumname ist reserviert, wenn der entsprechende Konferenzraum durch Konferenzteilnehmende tatsächlich genutzt wird. Bei Verlassen des letzten Teilnehmenden wird der Raum serverseitig geschlossen. Werden kurze oder einfach zu erratende Raumnamen für Konferenzen gewählt, besteht insbesondere bei nicht passwortgeschützten Konferenzen das Risiko einer Einwahl nicht geladener Teilnehmenden. Auch sind Namenskollisionen möglich, sodass sich unterschiedliche Teilnehmendenkreise in ein und demselben Konferenzraum einfinden.

Externe Reservierungssysteme können über die REST API von Jitsi Meet angebunden werden, damit Räume auch vorab reserviert oder nach Zeitablauf automatisiert geschlossen werden[25]. Diese können dann auch passwortgeschützt werden.

2.7.6 Verwendung der Lobby-Funktion

Zusätzlich bietet Jitsi Meet die Möglichkeit, dass Moderierende die Lobby-Funktion eines Konferenzraumes aktivieren. So können Teilnehmende nicht mehr selbstständig einer Konferenz beitreten, sondern der Moderierende muss Beitrittsanfragen vorher zustimmen. Sollte die Token-Authentifizierung genutzt werden, können bestimmte Accounts auch so konfiguriert werden, dass sie die Lobby umgehen.

Diese Funktion hat den Vorteil, dass Konferenzen weniger gestört werden und vor Beitritt anhand des Anzeigenamens geprüft werden kann, ob Teilnehmende zum Beitritt einer Konferenz berechtigt sind. Allein mit aktivierter Lobbyfunktion bleibt es jedoch weiterhin möglich, dass anonyme Besuchende eigene Konferenzen auf dem Jitsi-Meet-Server erstellen können.

2.8 Externe Serverdienste und Übertragung

Jitsi Meet greift zur Realisierung bestimmter Funktionen teilweise auf externe Dienste zurück. Bei diesen sollte für den entsprechenden Einsatzzweck selbst entschieden werden, ob eine Verwendung akzeptiert werden kann oder, z. B. aus Datenschutzgründen, Änderungen vorzunehmen sind. Neben den Daten, die für die externe Funktion benötigt werden, sollten auch die dabei entstehenden Metadaten nicht vergessen werden. Diese lassen gegebenenfalls Rückschlüsse auf Konferenzzeitpunkte und -teilnehmenden zu.

Solche Dienste müssen zunächst konfiguriert werden, indem in der Jitsi-Konfigurationsdatei eine entsprechende URL und gegebenenfalls Anmeldeinformationen angegeben werden. Das gilt zwar auch für Analyse-Funktionen wie Google Analytics, jedoch können über folgende Konfigurationseinstellung nicht nur alle externen Analytics-Dienste, sondern auch externe Avatar-Importe und GIF-Funktionen blockiert werden:

```
disableThirdPartyRequests: true,
```

Auszug 5: /etc/jitsi/meet/<hostname>-config.js

[25] https://jitsi.github.io/handbook/docs/devops-guide/reservation/

2.8.1 Session Traversal Utilities für NAT (STUN)

Um an Konferenzen im Internet hinter einer Firewall oder einem Router teilnehmen zu können, ohne vor jeder Verbindung spezielle Konfigurationen vornehmen zu müssen, werden häufig STUN-Server verwendet. In der mitgelieferten Konfiguration ist ein externer STUN Server voreingestellt. Grundsätzlich stellen STUN-Server kein Risiko dar, da damit allein die Verbindungseigenschaften festgestellt werden und z. B. die öffentliche IP-Adresse eines Teilnehmenden ermittelt wird. Die Kommunikation zwischen Teilnehmenden und Server kann über STUN-Server nicht eingesehen werden. Jedoch könnten die Betreibenden von STUN-Servern Profile von Teilnehmenden erstellen und darüber z. B. ermitteln, wann regelmäßig Konferenzen stattfinden. Es bietet sich daher an, einen STUN-Server zu wählen, bei dem keine Datenschutzbedenken bestehen.

Voreingestellt wird der Server der Jitsi-Entwickelnden verwendet. Es kann aber auch eine lokal betriebene Variante verwendet werden. Dafür kann beispielsweise der durch Jitsi Meet mitinstallierte Linux-TURN/STUN-Server `coturn` verwendet werden. Nach der Installation muss dazu folgende Konfigurationszeile angepasst werden:

```
stunServers: [
     { urls: 'stun:jitsi-meet.example.com:3478' },
],
```

Auszug 6: /etc/jitsi/meet/<hostname>-config.js

2.8.2 Schutz vor unerwünschten Übertragungen

Bei Jitsi Meet treten Teilnehmende den Konferenzen voreingestellt mit aktiviertem Mikrofon und / oder aktivierter Kamera bei. Dies kann unerwünschte Effekte haben, weil z. B. Informationen übertragen werden, die nicht für eine Übertragung bestimmt waren.

Um dies zu verhindern, sollte Jitsi Meet so konfiguriert werden, dass Teilnehmende immer stumm geschaltet und mit deaktivierter Videoübertragung beitreten. Eine spätere Aktivierung muss dann durch die Teilnehmenden selbst erfolgen, sobald diese etwas zur Konferenz beitragen möchten.

Zur Umsetzung des beschriebenen Verhaltens müssen in der Konfigurationsdatei von Jitsi Meet die Optionen wie folgt abgeändert werden:

```
startWithAudioMuted: true,
[…]
startWithVideoMuted: true,
```

Auszug 7: /etc/jitsi/meet/<hostname>-config.js

Es ist zu beachten, dass sich diese Einstellungen nur lokal auf das verwendete Endgerät auswirkt. Verlassen Teilnehmende eine Konferenz mit geöffnetem Mikrofon oder eingeschalteter Webcam und treten derselben oder einer anderen Konferenz des Jitsi-Meet-Servers zu einem späteren Zeitpunkt wieder bei, so ist das Mikrofon gegebenenfalls nicht stummgeschaltet bzw. die Kamera automatisch eingeschaltet, da die lokale Einstellung nicht erneut durch die Servereinstellung überschrieben wird.

Durch Nutzung einer vorgeschalteten Beitrittsseite (sog. „Prejoin page“), auf der Nutzende ihren Anzeigenamen sowie Einstellungen vor Konferenzbeitritt ändern können, lässt sich dieses Verhalten geringfügig eingrenzen. Ein versehentlich geöffnetes Mikrofon / eingeschaltete Kamera bei Konferenzbeitritt kann hiermit vermieden werden. Die vorgeschaltete Prejoin-Page wird nur bei einer Webbrowsernutzung von Jitsi Meet angezeigt.

Die Prejoin-Page muss in der Konfigurationsdatei von Jitsi Meet wie folgt konfiguriert und aktiviert werden:

```
prejoinConfig: {
[…]
    enabled: true,
[…]
},
```

Auszug 8: /etc/jitsi/meet/<hostname>-config.js

2.8.3 Reduktion der Datenübertragungen

Sollte der Jitsi-Meet-Server durch Video- und Audioübertragungen mehr Bandbreite erfordern, als zur Verfügung steht, kann es hilfreich sein, die Einstellungen anzupassen und somit den Datenverbrauch zu reduzieren.

Jitsi Meet bietet mehrere Möglichkeiten, durch die auf die erforderliche Bandbreite eingewirkt werden kann. Dabei können einzelne oder alle Einstellungen angepasst werden, abhängig davon, welche damit verbundenen Einschränkungen akzeptiert werden.

Die folgenden Einstellungen können in der Konfigurationsdatei von Jitsi Meet vorgenommen werden:

Deaktivieren von Videoübertragungen bei Konferenzbeginn:

```
startAudioOnly: true,
```

Auszug 9: /etc/jitsi/meet/<hostname>-config.js

Reduzierung der Videoauflösung auf 240p:

```
resolution: 240,
```

Auszug 10: /etc/jitsi/meet/<hostname>-config.js

Reduzierung der übertragenen Videodaten auf 240p:

```
constraints: {
     video: {
         height: {
             ideal: 240,
             max: 240,
             min: 240,
         },
     },
},
```

Auszug 11: /etc/jitsi/meet/<hostname>-config.js

Reduzierung aktiver, paralleler Videoübertragungen auf 1:

```
channelLastN: 1,
```

Auszug 12: /etc/jitsi/meet/<hostname>-config.js

Reduzierung der Bildübertragungsrate bei Desktopfreigaben auf 1 Bild pro Sekunde:

```
desktopSharingFrameRate: {
    min: 1,
    max: 1,
},
```

Auszug 13: /etc/jitsi/meet/<hostname>-config.js

2.9 Absicherung mit einer Firewall

Um eine möglichst geringe Angriffsfläche zu bieten, sollte der Jitsi-Meet-Server mit einer Firewall abgesichert werden. Dabei sollten nur die zwingend erforderlichen Ports zugelassen werden. Alle nicht benötigten Ports sollten blockiert werden.

Jitsi Meet verwendet vorkonfiguriert die folgenden Ports für die angegebenen Aufgaben:

- TCP 443: Darstellung der Web-Oberfläche und Management der Videokonferenzen
- UDP 10000: Übertragung der Audio- und Videodaten
- UDP 3478: Anfragen an den STUN-Server (wird nur benötigt, wenn STUN über den `coturn`-Dienst auf dem Jitsi Meet Server selbst betrieben wird, siehe 2.8.1 Session Traversal Utilities für NAT (STUN))
- TCP 5349: Rückfallkanal für Audio- und Videodaten über TCP, sofern UDP blockiert ist (wird vom `coturn`-Dienst bereitgestellt, der von Jitsi Meet mitinstalliert wird).

Soll das TLS-Zertifikat des Servers über Let's Encrypt bezogen werden, muss folgender Port zumindest im Rahmen des Prozesses zugänglich gemacht werden, damit in regelmäßigen Abständen eine automatische Erneuerung des Zertifikats erfolgen kann:

- TCP 80: Let's Encrypt-Zertifikat

Alternativ ist es möglich, den benötigten Port ausschließlich bei Erneuerung des Let's Encrypt-Zertifikats (alle 90 Tage) zu öffnen.

Wenn Jitsi Meet in einer Virtuellen Maschine oder einem Container betrieben wird, muss darüber hinaus sichergestellt sein, dass die erforderlichen Ports an die Virtuelle Maschine / den Container weitergeleitet werden.

2.10 Mobile App

Für Jitsi Meet steht eine mobile App in den gängigen App-Stores zur Verfügung (siehe 2.3 Vertrauenswürdige Quellen). Dabei sollte berücksichtigt werden, dass die Apps für Android und iOS teilweise Drittanbieter-Tools, wie Google CrashLytics, Google Firebase Analytics und Amplitude beinhalten.

Die folgenden Konfigurationen sollten in den jeweils genutzten mobilen Apps beachtet werden:

Server-URL:	Sichere (https) Web-Adresse des eigenen Jitsi Meet-Servers; andernfalls wird der offizielle Jitsi Meet-Server der Entwickelnden verwendet und Gesprächsinhalte können ggf. Dritten offenbart werden.
Stumm beitreten:	aktiviert
Ohne Video beitreten:	aktiviert
Absturzberichte deaktivieren:	aktiviert

3 Regelmäßiger Betrieb

3.1 Aktualisierungen

Vor Aktualisierungen von Jitsi Meet sollte in jedem Fall sichergestellt werden, dass alle Änderungen an der Konfiguration dokumentiert sind und Sicherheitskopien der Konfigurationsdateien und der Daten vorliegen. Das Vorgehen bei der Aktualisierung selbst ist von der gewählten Installationsmethode abhängig (siehe 2.4 Installation).

Im Rahmen der Weiterentwicklung von Jitsi Meet kann es vorkommen, dass die Entwickelnden neue Konfigurationsoptionen einführen, das Verhalten bestehender Konfigurationen ändern oder um zusätzliche Parameter ergänzen. Bei der Weiterverwendung von alten Konfigurationsdateien kann dabei nicht sichergestellt werden, dass diese Änderungen umgesetzt werden, was ein Risiko für das System darstellen kann.

Daher empfiehlt es sich, bei Aktualisierungen immer die aktuellsten Konfigurationsdateien der Software zu nutzen, vorherige Anpassungen erneut vorzunehmen und zu prüfen, ob neue Konfigurationsoptionen vorhanden sind, die Auswirkungen auf die Sicherheit haben und ggf. angepasst werden müssen.

3.2 Pflegen von Nutzendenkonten

Im Falle der Verwendung von Nutzendenauthentifizierung (siehe 2.7 Nutzendenverwaltung und Absicherung der Konferenzräume) sollte während des Betriebes regelmäßig geprüft werden, ob alle existierenden Nutzendenkonten weiterhin berechtigt sind, den Jitsi-Meet-Server zu nutzen oder ob ggf. Nutzendenkonten existieren, die entfernt werden sollten.

Die zur Nutzendenverwaltung von Jitsi Meet verwendete Software Prosody bietet keine Möglichkeit, die existierenden Konten übersichtlich aufzulisten. Um an diese Informationen zu gelangen, müssen diese daher über das Datenverzeichnis von Prosody ausgelesen werden.

```
ls /var/lib/prosody/*/accounts
```

Kommandozeilenbefehl 1: Auflistung der Nutzendenkontendateien (<Kontoname>.dat)

Zu jedem vorhandenen Konto existiert eine .dat-Datei. Bei den Konten „jvb“ und „focus“ handelt es sich um Systemkonten des Jitsi Meet-Servers, die nicht entfernt werden sollten, um die Funktionsfähigkeit nicht zu gefährden.

3.3 Prüfen der Systemressourcen

Jitsi Meet kann, abhängig von Anzahl der Konferenzen, Anzahl der Teilnehmenden und verwendeten Multimediafunktionen, eine hohe Systemauslastung verursachen. Um Verzögerungen bei der Übertragung von Konferenzdaten oder Systemausfälle zu vermeiden, sollte daher regelmäßig geprüft werden, wie hoch die Systemauslastung ausfällt. Die wichtigsten Parameter, die überprüft werden sollten, sind dabei die Prozessorauslastung, die Speicherauslastung und die Netzwerkauslastung.

Unter Linux steht dafür eine große Auswahl von Werkzeugen zur Verfügung. Zusätzlich werden häufig separate Werkzeuge zur Überwachung von Virtuellen Maschinen oder Containern mit ausgeliefert.

Ein allgemeiner Überblick kann beispielsweise mit dem Programm `sar` aus der `sysstat`-Werkzeugsammlung erhalten werden. Über den folgenden Aufruf kann sich so ein Eindruck über die Systemauslastung verschafft und im Beispiel alle drei Sekunden aktualisiert werden:

```
sar -h -u -P ALL -r -n DEV 3
```

Kommandozeilenbefehl 2: Prüfung der Systemauslastung mit sar

Im Wesentlichen sollte hier darauf geachtet werden, dass die Inaktivitätszeit (idle) der CPU-Kerne nicht 0,00% erreicht, die Arbeitsspeichernutzung (`memused`) nicht 100% erreicht und die Netzwerkaktivität (rxKb/s, txKb/s) nicht die zur Verfügung stehende Bandbreite übersteigt.

3.4 Protokollierung und Fehlersuche

Für die Fehlersuche und Protokollierung legt Jitsi Meet voreingestellt alle Vorgänge in einem separaten Verzeichnis unter dem systemweiten Protokollpfad ab. Die wichtigsten Informationen zu den Audio-/Videokonferenzen selbst können dabei aus der Datei `jicofo.log` und `jvb.log` bezogen werden:

- `/var/log/jitsi/jicofo.log`
- `/var/log/jitsi/jvb.log`

Protokolle zu Authentifizierungen können aus der Protokolldatei von Prosody entnommen werden, die als `prosody.log` benannt ist und unter einem separaten Verzeichnis im systemweiten Protokollpfad zu finden ist:

- `/var/log/prosody/prosody.log`

Sollten konkrete Verbindungsinformationen mit IP-Adressen benötigt werden, können diese aus den Protokolldateien zur Videobridge oder des genutzten Webservers (z.B. nginx oder Apache) entnommen werden:

- `/var/log/nginx/access.log`
- `/var/log/apache2/access.log`

Der Umfang der zu protokollierenden Informationen von Jicofo, Prosody und der Videobridge lässt sich in den Konfigurationsdateien der zugehörigen Verzeichnisse konfigurieren:

- `/etc/jitsi/jicofo/logging.properties`
- `/etc/prosody/prosody.cfg.lua`
- `/etc/jitsi/videobridge/logging.properties`

Darüber hinaus kann es im Rahmen der Fehleranalyse hilfreich sein, auch die Seite der Teilnehmenden zu betrachten. Besonders bei der Teilnahme über Webbrowser werden hilfreiche Informationen in der Browser-Konsole angezeigt, die zumeist über die Taste F12 oder das Menü des Webbrowsers aufgerufen werden können.

4 Weiterführende Informationen

Für weiterführende Informationen sollte in erster Instanz auf die folgenden, offiziellen Quellen der Entwickelnden zurückgegriffen werden:

- Die Webseite der Entwickelnden: https://jitsi.org/jitsi-meet/
- Der Quellcode von Jitsi inkl. Dokumentationen: https://github.com/jitsi
- Das Jitsi-Forum für Entwickelnden und Nutzenden: https://community.jitsi.org

Darüber hinaus bietet das BSI unter

https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Kompendium-Videokonferenzsysteme.pdf

ein Kompendium für Videokonferenzsysteme an, dass sich an Personen richtet, die für Entscheidung, Planung, Beschaffung, Betrieb, Administration und Audits zuständig sind. Darüber hinaus sind auch

hilfreiche Informationen für Endnutzende enthalten, die über Videokonferenzen Inhalte beziehungsweise Informationen mit normalem und erhöhtem Schutzbedarf austauschen.

Mit den BSI-Veröffentlichungen publiziert das Bundesamt für Sicherheit in der Informationstechnik (BSI) Dokumente zu aktuellen Themen der Cyber-Sicherheit. Kommentare und Hinweise können von Lesern an info@cyber-allianz.de gesendet werden.