TAK Server

Configuration Guide

Version 4.7

January 2023

1. Table of Contents

1 About TAK Server

TAK Server is a situational awareness server, that provides a dynamic Common Operating Picture to users of the Team Awareness Kit, including ATAK (Android), WinTAK (Windows) and WebTAK. TAK enables sharing of geolocated information in real time for military forces, law enforcement, and emergency responders. It supports both wireless and wired networks, as well as cloud and data center deployment.

The references to https://wiki.tak.gov require an account.

2 Change Log

See https://wiki.tak.gov/display/DEV/TAK+Server+Change+Log

3 System Requirements

Supported Operating Systems:

- Red Hat Enterprise Linux (RHEL) 7 or 8
- CentOS 7 (***not*** CentOS 8 Stream)
- Rocky Linux 8 (Replacement for CentOS 8, which is EOL)
    - Java 11 (installed by default via RPM dependencies, but you if you have Java 8 installed also, you must ensure that TAK Server is using Java 11)

3.1 Server Requirements

- 4 processor cores
- 8 GB RAM
- 40 GB disk storage

NOTE: Insecure ports are a potential security risk and may allow attackers to gain access to the system resulting in the disclosure of personal and sensitive information. Use of unencrypted ports should be avoided to ensure a secure TAK Server deployment.

3.2 AWS / GovCloud Recommended Instance Type

- c5.xlarge
  - 4 vCPU
  - 8 GB RAM
  - Up to 10 Gbps network bandwidth
- For 2-server installation, use this instance type for both servers.

TAK Server is a TLS-enabled networking server. In order to ensure consistent performance, burstable AWS EC2 instance types such as T2 are not recommended. TLS and TCP processing requires consistent, continuous CPU performance. C4 and C5 instances are designed for predictable CPU performance, and are better-suited for TAK Server deployments.

More information about instance types may be found here:

https://aws.amazon.com/ec2/instance-types

Usage of larger instance types or physical servers is supported for scalability, to support more concurrent active users.

4 Installation

4.1 Overview and Installer Files

TAK Server supports multiple deployment configurations:

- **Single server install**: One server running TAK Server core (messaging, API, plugins and database): recommended for fewer than 500 users.

- **Two server install**: One server running TAK Server core (messaging, API, plugins and database) and a second server running PostgreSQL database: recommended for more than 500 users.

- **Containerized docker install**: One container running TAK Server core (messaging, API, plugins and database) and another container running PostgreSQL database (designed for operating systems other than CentOS 7 / RHEL 7)

The following installation files are provided:

`takserver-4.7-RELEASE-x.noarch.rpm` – I**nstaller for single-server install**

`takserver-database-4.7-RELEASE-x.noarch.rpm` – **Database installer for two-server install**

`takserver-core-4.7-RELEASE-x.noarch.rpm` – **Core installer for two-server install**

`takserver-docker-4.7-RELEASE-x.zip` **– Containerized docker install bundle**

`takserver-docker-hardened-4.7-RELEASE-x.zip` **– Containerized hardeneded docker install bundle**

`takserver-fed-hub-4.7-RELEASE-x.noarch.rpm` – **Installer for federation hub (beta)**

Federation hub documentation available here:

https://wiki.tak.gov/display/TPC/Federation+Hub

4.2 New Installation: One Server

Start with a fresh install of a supported OS. For AWS / cloud installation, see recommended instance type on page 4. An OS install with a GUI is recommended, so that a web browser can be run locally to configure TAK Server.

Increase system limit for number of concurrent TCP connections (do once):

> echo -e "* soft nofile 32768\n* hard nofile 32768" | sudo tee --append /etc/security/limits.conf > /dev/null

Install epel on CentOS and Rocky Linux:

> sudo yum install epel-release -y

Install epel on RHEL:

> sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm

Install postgres yum repository on CentOS 7 and RHEL 7 (required in order to install up-to-date Postgresql and PostGIS packages.):

> sudo yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm -y

Install postgres yum repository on RHEL 8 and Rocky Linux 8:

> sudo yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm -y && sudo dnf -qy module disable postgresql

> sudo yum update -y

Install java 11 on RHEL 8 and Rocky Linux 8:

> sudo yum install java-11-openjdk-devel -y

Enable PowerTools on Rocky Linux 8:

> sudo dnf config-manager --set-enabled powertools

Enable Repository Management and repository CodeReady Builder on RHEL 8:

> sudo subscription-manager config --rhsm.manage_repos=1

> sudo subscription-manager repos --enable codeready-builder-for-rhel-8-x86_64-rpms

Note: If you get the error 'This system has no repositories available through subscriptions', you need to subscribe your system with "sudo subscription-manager register --username <your_username> --password <your_password> --auto-attach"

Install Postgis on RHEL 8 and Rocky Linux 8:

> sudo dnf -y install postgis30_10

Install TAK Server

> sudo yum install takserver-4.7-RELEASEx.noarch.rpm -y

Apply SELinux takserver-policy on RHEL 8 and Rocky Linux 8:

> cd /opt/tak
> sudo ./apply-selinux.sh
> sudo semodule -l | grep takserver

Check Java version:

> java -version

This should tell you you have 11.x.y

If the "java -version" command tells you your Java version is not 11.x.y, then you can use the alternatives command to change it:

> sudo alternatives --config java

Using this command to switch to Java 11 will be necessary if this machine was running Java 1.8 prior to this install.

> sudo /opt/tak/db-utils/takserver-setup-db.sh

> sudo systemctl daemon-reload

> sudo systemctl [start|stop] takserver

You can set TAK Server to start at boot by running

sudo systemctl enable takserver

For secure operation, TAK Server requires a keystore and truststore (X.509 certificates).

Next, follow the instructions in Appendix B to create these certificates.

After creating certificates, restart TAK Server so that the newly created certificates can be loaded.

> sudo systemctl restart takserver

While following the instructions in Appendix B, you will have created an **admin** certificate. Import this certificate into your browser, so that you can access the Admin. It will be located here on your TAK Server machine:

/opt/tak/certs/files/**admin**.pem

Import this client certificate into your browser.

If you are using Firefox, go to Settings -> Preferences -> Privacy & Security -> Certificates -> View Certificates

Go to Import. Upload this file:

/opt/tak/certs/files/**admin**.p12

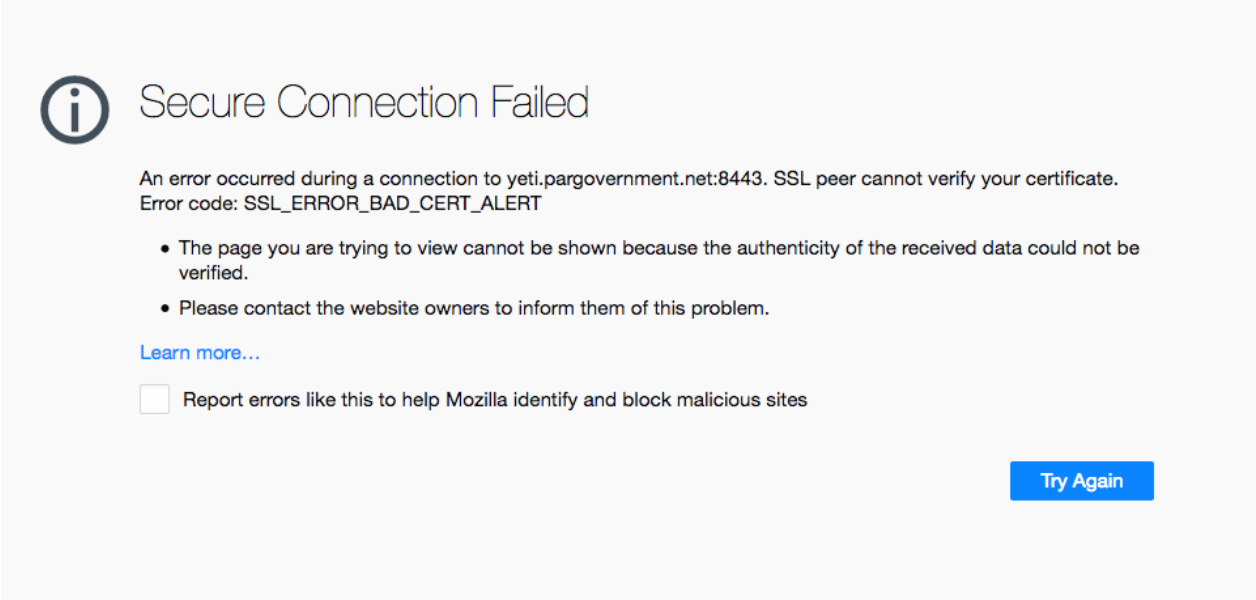
Enter the certificate password. The default password is *atakatak*

Browse to:

https://localhost:8443

Select the **admin** certificate to log in.

An error message similar to this indicates that the correct client certificate has not been imported into the browser:



Optional: Create an admin user and use a browser locally to set up security configuration.

Create login credentials for local adminstrative access to the configuration interface:

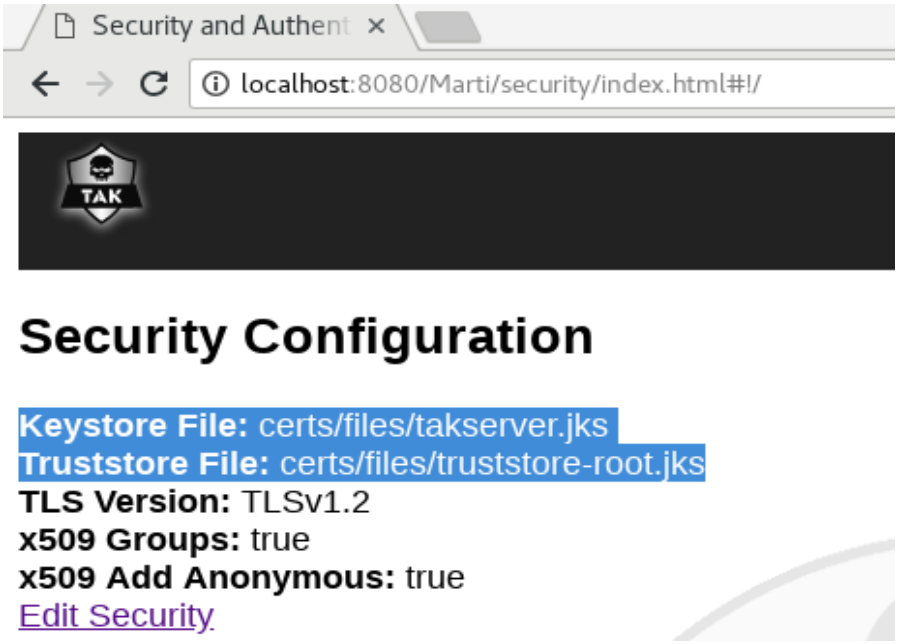
> sudo java -jar /opt/tak/utils/UserManager.jar usermod -A -p <password> <username>

Using Firefox or Chrome on this computer, browse to this address to verify keystore and truststore configuration:

http://localhost:8080

Login with the admin username and password just created above.

This will bring you to the security and authentication configuration page. If you followed the certificate generation instructions in Appendix B verbatim, then the default settings will be correct. Verify the file locations for keystore and truststore:



If you make changes to the keystore or truststore settings, restart takserver:

> sudo systemctl restart takserver

4.3 New Installation: Two Servers

Follow the procedures in the following two sections to install the database server, and the messaging server. For AWS / cloud installation, see recommended instance type on page 4. Use this instance type for both servers.

4.3.1 Server One: Database Server

Start with a fresh install of a supported OS.

> sudo yum install epel-release -y

Install postgres yum repository (required in order to install up-to-date Postgresql and PostGIS packages.)

> sudo yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm -y

> sudo yum update -y

Install TAK Server database. ***Use database RPM***

> sudo yum install takserver-database-4.7-RELEASEx.noarch.rpm -y

> sudo /opt/tak/db-utils/takserver-setup-db.sh

Update firewall rules to allow communication with server two, for TCP port 5432.

Open the file /opt/tak/CoreConfig.example.xml and look for the auto-generated password for the database. This password will be used to configure the Core Server.

<connection url="jdbc:postgresql://127.0.0.1:5432/cot" username="martiuser" password="**Database_password**" />

4.3.2 Server Two: Core Server

Start with a fresh install of a supported OS. An install with a GUI is recommended, so that a web browser can be run locally to configure TAK Server.

Increase system limit for number of concurrent TCP connections (do once)

> echo -e "* soft nofile 32768\n* hard nofile 32768" | sudo tee --append /etc/security/limits.conf > /dev/null

Install TAK Server.

> sudo yum install takserver-core-4.7-RELEASEx.noarch.rpm -y

> java -version

This should tell you you have 11.x.y

If the "java -version" command tells you your Java version is not 11.x.y, then you can use the alternatives command to change it:

> sudo alternatives --config java

Using this command to switch to Java 11 will be necessary if this machine was running Java 1.8 prior to this install.

Configure database connection by updating /opt/tak/CoreConfig.xml:

<repository enable="true" numDbConnections="200" primaryKeyBatchSize="500" insertionBatchSize="500">
    <connection url="jdbc:postgresql://**<Database_server_IP_address>**:5432/cot" username="martiuser" password="**Database_password**"/>
</repository>

> sudo systemctl daemon-reload

sudo systemctl [start|stop] takserver

You can set TAK Server to start at boot by running

sudo systemctl enable takserver

For secure operation, TAK Server requires a keystore and truststore (X.509 certificates).

Next, follow the instructions in Appendix B: Certificate Generation to create these certificates.

After creating certificates, restart TAK Server so that the newly created certificates can be loaded.

> sudo systemctl restart takserver

Import this client certificate into your browser.

If you are using Firefox, go to Settings -> Preferences -> Privacy & Security -> Certificates -> View Certificates

Go to Import. Upload this file:

/opt/tak/certs/files/admin.p12

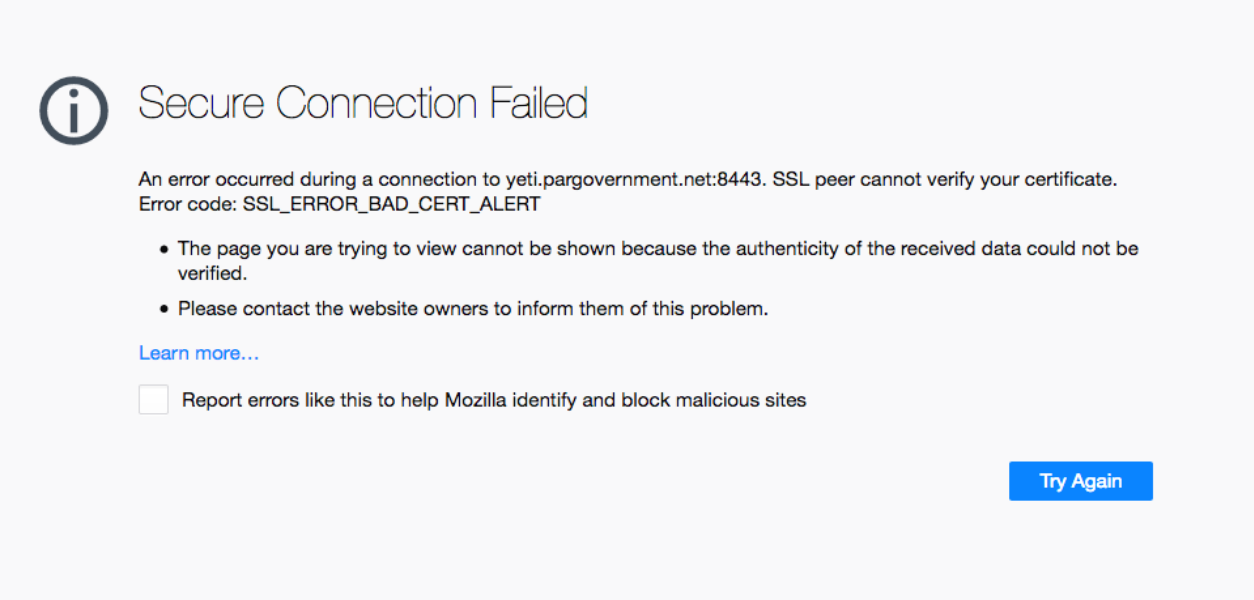
Enter the certificate password. The default password is atakatak

Browse to:

https://localhost:8443

Select the admin certificate to log in.

An error message similar to this indicates that the correct client certificate has not been imported into the browser:



Optional: Create an admin user and use a browser locally to set up security configuration.

Create login credentials for local adminstrative access to the configuration interface:

> sudo java -jar /opt/tak/utils/UserManager.jar usermod -A -p <password> <username>

Using Firefox or Chrome on this computer, browse to this address to verify keystore and truststore configuration:

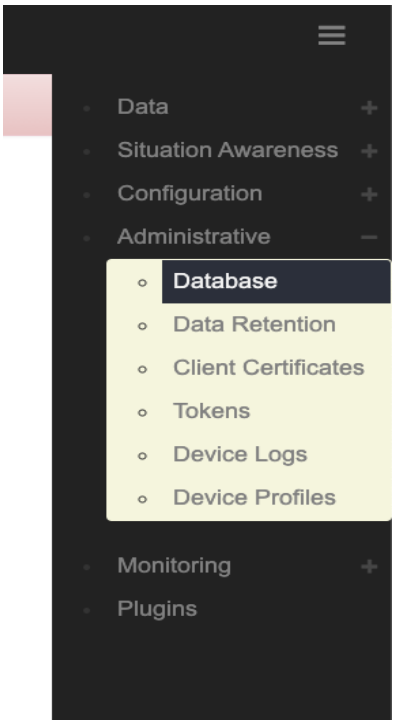
http://localhost:8080

Login with the admin username and password just created above.

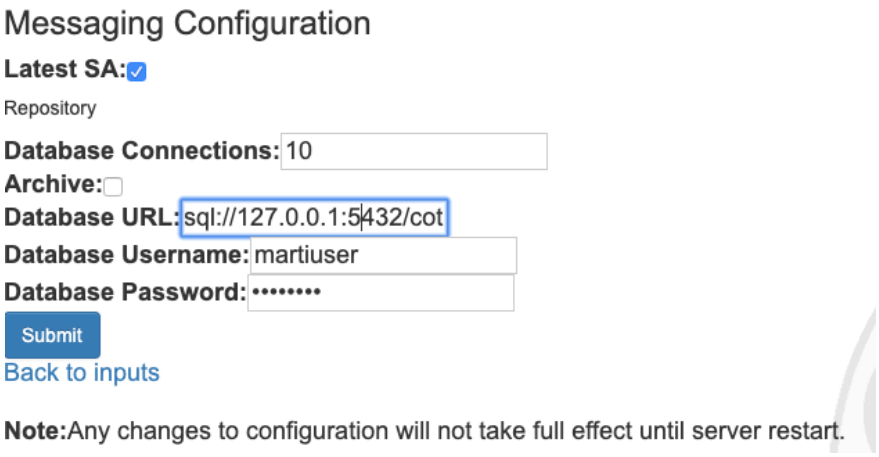
This will bring you to the security and authentication configuration page. If you followed the certificate generation instructions in Appendix B verbatim, then the default settings will be correct. Verify the file locations for keystore and truststore.

Instructions continue on next page

Configure TAK Server to connect to the database. Access the Database configuration settings:



Edit the database connection address, specifying the hostname or IP address of the database server:



Restart TAK Server

> sudo systemctl restart takserver

4.4 Use Setup Wizard to Configure TAK Server

The TAK Server configuration wizard will help you set up common configuration options once you have installed and started TAK Server. The wizard will guide you through the setup process for a secure configuration, using the default ports that ATAK and WinTAK will connect to.

Once you have created your adminstrative login credentials as in the previous section, go to:

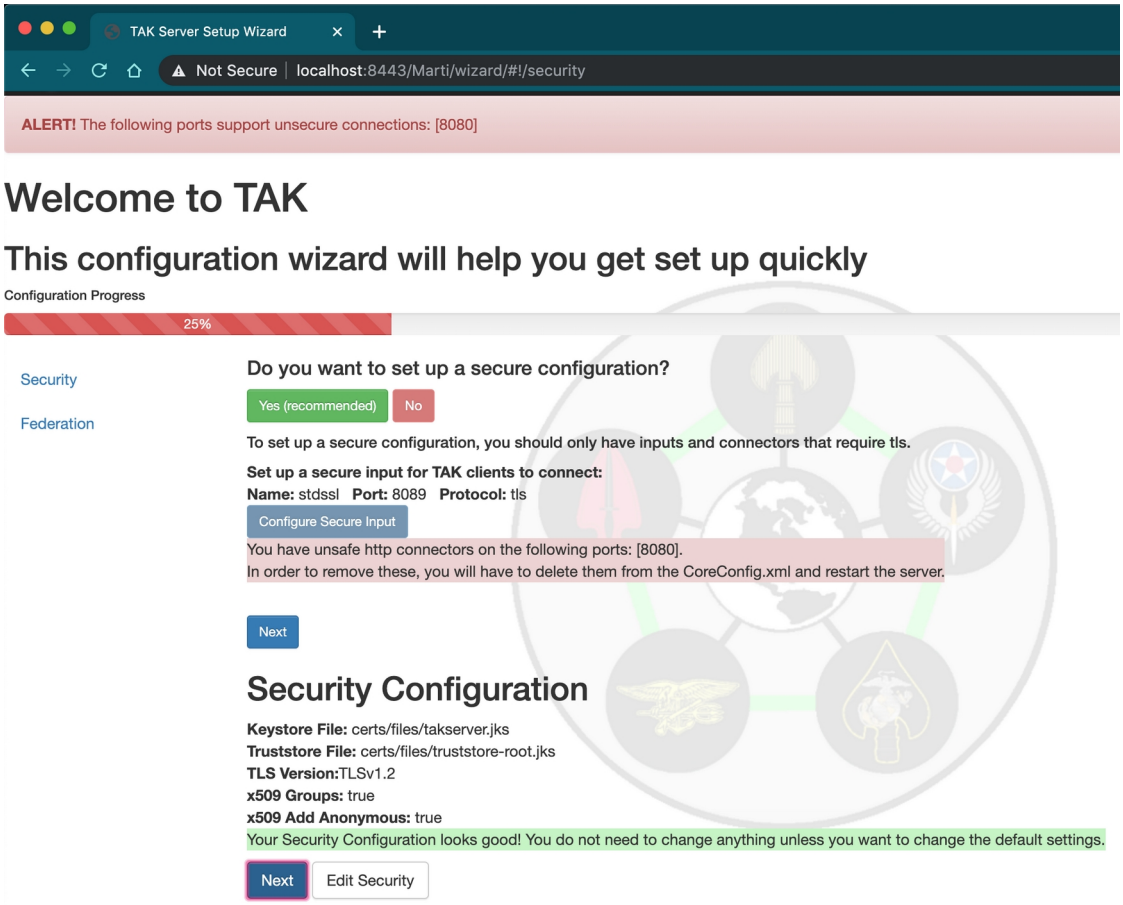
https://localhost:8443/setup/ (Recommended. Uses the more secure client certificate)

or,

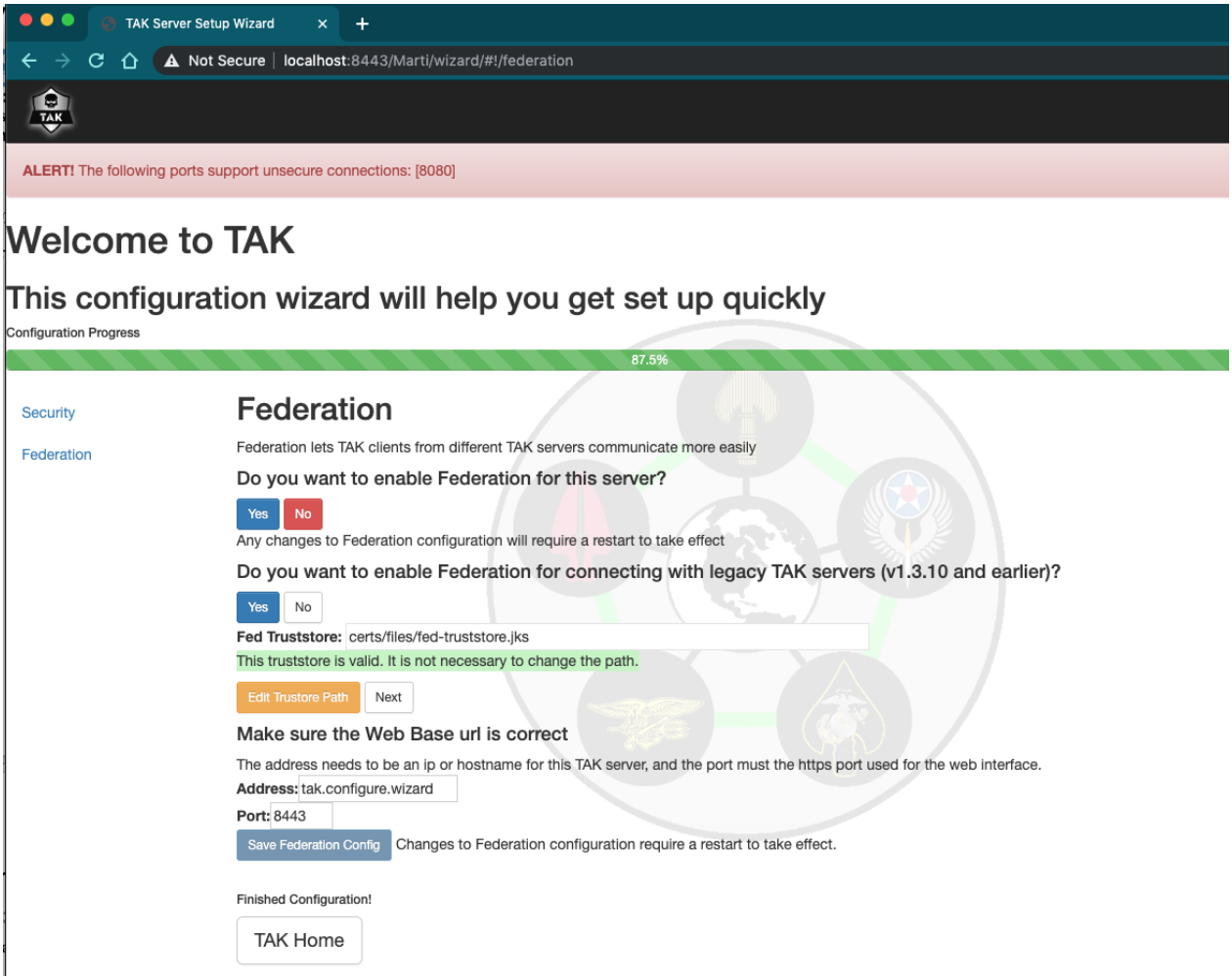
http://localhost:8080/setup/ (if using the username/password authentication method)

Then follow the prompts to begin configuring. The wizard will first walk you through recommended security configuration:

NOTE: Insecure ports are a potential security risk and may allow attackers to gain access to the system resulting in the disclosure of personal and sensitive information. Use of unencrypted ports should be avoided to ensure a secure TAK Server deployment.



Followed by the recommended federation configuration, if you wish to set up your TAK Server to support federation. (For more information on federation, go to section 8):


TAK Server Setup Wizard
Not Secure | localhost:8443/Marti/wizard/#!/federation
ALERT! The following ports support unsecure connections: [8080]
Welcome to TAK
This configuration wizard will help you get set up quickly
Configuration Progress
87.5%
Security
Federation
Federation
Federation lets TAK clients from different TAK servers communicate more easily
Do you want to enable Federation for this server?
Yes
No
Any changes to Federation configuration will require a restart to take effect
Do you want to enable Federation for connecting with legacy TAK servers (v1.3.10 and earlier)?
Yes
No
Fed Truststore: certs/files/fed-truststore.jks
This truststore is valid. It is not necessary to change the path.
Edit Trustore Path
Next
Make sure the Web Base url is correct
The address needs to be an ip or hostname for this TAK server, and the port must the https port used for the web interface.
Address: tak.configure.wizard
Port: 8443
Save Federation Config
Changes to Federation configuration require a restart to take effect.
Finished Configuration!
TAK Home

4.5 Upgrade existing TAK Server installation

4.5.1 Upgrade from 1.3.11-RELEASE21 or higher

Follow this procedure to upgrade a system running TAK Server version 1.3.11-RELEASE21 or higher (such as 4.6). For a **single-server install**, upgrade this package:

> sudo yum install takserver-4.7-RELEASEx.noarch.rpm -y

For a **two-server install**, upgrade these packages on the servers on which they are installed:

> sudo yum install takserver-core-4.7-RELEASEx.noarch.rpm -y

> sudo yum install takserver-database-4.7-RELEASEx.noarch.rpm -y

Update the database schema on the existing TAK Server Postgres database.

> sudo java -jar /opt/tak/db-utils/SchemaManager.jar upgrade

4.5.2 Upgrade from 1.3.11-RELEASE15

Follow this procedure to upgrade **only** a system running TAK Server version 1.3.11-RELEASE15.

Remove Postgis version 2.4

> sudo rpm -e --nodeps postgis24_10 postgis24_10-utils

Upgrade TAK Server package and dependencies.

> sudo yum install takserver-4.7-RELEASEx.noarch.rpm -y

Update the PostGIS extension on the TAK Server database to the new version.

> sudo su - postgres -c "psql -d cot -c 'ALTER EXTENSION postgis UPDATE;'"

Update the database schema on the existing TAK Server Postgres database.

> sudo java -jar /opt/tak/db-utils/SchemaManager.jar upgrade

4.5.3 Upgrade from 1.3.10 or earlier

Follow this procedure to upgrade a system running **version 1.3.10 or earlier** of TAK Server. **These instructions update both the TAK Server software and the Postgresql database.**

Increase system limit for number of concurrent TCP connections (do once)

> echo -e "* soft nofile 32768\n* hard nofile 32768" | sudo tee append /etc/security/limits.conf > /dev/null

Install postgres yum repository

> sudo yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm -y

> sudo yum update -y

Upgrade TAK server package and dependencies

> sudo yum install takserver-4.7-RELEASEx.noarch.rpm -y

Remove Java 1.8

> sudo yum remove java-1.8.0-openjdk java-1.8.0-openjdk-devel java-1.8.0-openjdk-headless -y

Update the database schema on the existing TAK Server Postgres database

> sudo java -jar /opt/tak/db-utils/SchemaManager.jar upgrade

TAK Server version 1.3.11-RELEASE21 uses PostgreSQL 10 and PostGIS 3.0. Use these commands to check your current PostgreSQL and PostGIS versions, before upgrading the database.

> sudo -u postgres psql cot -c "select version();"

> sudo -u postgres psql cot -c "select postgis_full_version();"

Stop TAK server to prevent new records from being written to the database during the update process.

> sudo systemctl stop takserver

The upgrade process consists of dumping the version 9.x database contents into a file, then importing this file into the new version 10 database. The database dump file can be large, so make sure you have plenty of disk space.

Dump the existing database to a file.

> sudo su postgres

> cd ~

> pg_dumpall > *takserver_db_backup_file*

> exit

Stop the currently running version 9 server

> sudo systemctl stop postgresql

Restart the system

> sudo reboot

Inititalize and start the Postgres 10 server, and populate it with the TAK Server database backup. First, become the *postgres* user:

> sudo su postgres

> cd ~

> /usr/pgsql-10/bin/initdb -D /var/lib/pgsql/10/data/

> /usr/pgsql-10/bin/pg_ctl -D /var/lib/pgsql/10/data/ -l logfile start

> /usr/pgsql-10/bin/psql -d postgres -f *takserver_db_backup_file*

> exit

Now set PostgreSQL 10 to run on startup:

> sudo systemctl enable postgresql-10

Remove the Postgres 9 binaries.

> sudo yum remove postgresql postgresql-contrib postgresql-libs postgresql-server -y

4.6 Docker Install

TAK Server can be installed using docker. To begin, you will need the docker release which comes as a zip file called 'takserver-docker-<version>.zip'.

If you using CentOS 7, follow these instructions first to install docker, start the docker daemon and use it as a regular user:

https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-centos-7

Next, unzip the docker zip file. **All further commands should be run from this top level 'takserver-docker-<version>' directory**. If you are familiar with the rpm install, the 'tak' folder within the 'takserver-docker-<version>' directory represents the '/opt/tak' directory installed by the rpm. When the takserver containers are built, the 'tak' directory will be mounted to '/opt/tak' within the containers. Therefore, any references to '/opt/tak' outside this section of the guide will be equivalent to the 'tak' directory you have on the host, or '/opt/tak' if you are working from inside the container.

The 'tak' directory is where the coreconfig, certificates, logs and other TAK configuration/tools will live. This folder is shared between the host, the takserver container and the takserver database container. This means you can tail the logs or manually edit the coreconfig from the host without being inside the container.

Notes for running in Windows Subsystem for Linux (WSL) 2:

When running TAK Server in the WSL2 environment, follow the steps outlined in the Best Practices section here https://docs.docker.com/desktop/windows/wsl/ to maximize TAK Server performance. Specifically, it’s recommended that you copy the 'takserver-docker-<version>.zip' file into your WSL

user’s home directory and execute all docker commands from there (vs accessing your Windows filesystem from /mnt). For example, if your WSL username was ‘tak’ and you’re running Ubuntu-20.04, copy the docker .zip file to \\wsl$\Ubuntu-20.04\home\tak using Windows Explorer. Next, navigate to that directory within WSL. From there, unzip the file and run the docker commands below for TAK Server. It’s important to unzip the file from within WSL to ensure permissions are setup correctly.

TAK Server CoreConfig Setup:

1. Open tak/CoreConfig.xml and set a database password

2. Make any other configuration changes you need

TAK Server Database Container Setup:

1. Build TAK server database image:

```
> docker build -t takserver-db:"$(cat tak/version.txt)" -f docker/Dockerfile.takserver-db .
```

2. Create a new docker network for the current tak version:

```
> docker network create takserver-"$(cat tak/version.txt)"
```

3. The TAK Server database container can be configured to persist data directly to the host or only within the container.

a. To persist to the host, create an empty host directory (unless you have a directory from a previous docker install you want to reuse). For upgrading purposes, we recommend that you keep the takserver database directory outside of the 'takserver-docker-<version>' directory structure.

```
> docker run -d -v <absolute path to takserver database directory>:/var/lib/postgresql/data:z -v
$(pwd)/tak:/opt/tak:z -it -p 5432:5432 --network takserver-"$(cat tak/version.txt)" --network-alias tak-database --name takserver-db-"$(cat tak/version.txt)" takserver-db:"$(cat tak/version.txt)"
```

b. To run TAK server database with container only persistence

```
> docker run -d -v $(pwd)/tak:/opt/tak:z -it -p 5432:5432 --network takserver-"$(cat tak/version.txt)" --network-alias tak-database --name takserver-db-"$(cat tak/version.txt)" takserver-db:"$(cat tak/version.txt)"
```

TAK Server Container Setup:

1. Build TAK Server image:

```
> docker build -t takserver:"$(cat tak/version.txt)" -f docker/Dockerfile.takserver .
```

2. Running TAK Server container: use -p <host port>:<container port> to map any additional ports you have configured. **Adding new inputs or changing ports while the container is running will require the container to be recreated so that the new port mapping can be added**.

```
> docker run -d -v $(pwd)/tak:/opt/tak:z -it -p 8080:8080 -p 8443:8443 -p 8444:8444 -p 8446:8446 -p 8087:8087 -p 8088:8088 -p 9000:9000 -p 9001:9001 --network takserver-"$(cat tak/version.txt)" --name takserver-"$(cat tak/version.txt)" takserver:"$(cat tak/version.txt)"
```

3. Before using the TAK Server, you must setup the certificates for secure operation. If you have already configured certificates you can skip this step. You can also copy existing certificates into 'tak/certs/files' and a UserAuthetication.xml file into 'tak/' to reuse existing certificate authentication settings. **Any change to certificates while the container is running will require either a TAK server restart or container restart.** Additional certificate details can be found in Appendix B.

a. Edit tak/certs/cert-metadata.sh

b. Generate root ca

```
> docker exec -it takserver-"$(cat tak/version.txt)" bash -c "cd /opt/tak/certs && ./makeRootCa.sh"
```

c. Generate server cert

```
> docker exec -it takserver-"$(cat tak/version.txt)" bash -c "cd /opt/tak/certs && ./makeCert.sh server takserver"
```

d. Create client cert(s)

```
> docker exec -it takserver-"$(cat tak/version.txt)" bash -c "cd /opt/tak/certs && ./makeCert.sh client <user>"
```

e. Restart takserver to load new certificates

```
> docker exec -d takserver-"$(cat tak/version.txt)" bash -c "cd /opt/tak/ && ./configureInDocker.sh"
```

f. Tail takserver logs from the host. **Once TAK server has successfully started, proceed to the next step.**

```
> tail -f tak/logs/takserver-messaging.log
> tail -f tak/logs/takserver-api.log
```

4. Accessing takserver

a. Create login credentials for unsecured access on port 8080 (http)

```
> docker exec takserver-"$(cat tak/version.txt)" bash -c  "cd /opt/tak/ && java -jar /opt/tak/utils/UserManager.jar usermod -A -p <password> <username>"
```

b. Create admin client certificate for access on secure port 8443 (https)

```
> docker exec takserver-"$(cat tak/version.txt)" bash -c  "cd /opt/tak/ && java -jar utils/UserManager.jar certmod -A certs/files/<client cert>.pem"
```

Hardened TAK Server Setup:

The hardened TAK Database and Server containers provide additional security by including the use of secure Iron Bank base images, container health checks, and minimizing user privileges within the containers.

The hardened TAK images are available in a zip file, takserver-docker-hardened-<version>.zip. The steps for setting up the hardened containers are similar to the standard docker installation steps given above except for the following:

Certificate Generation:

The certificate generation container is only required to run once for TAK Server initialization. Run all commands in this section from the root of the unzipped hardened docker contents.

1. Build the Certificate Authority Setup Image:

< docker build -t ca-setup-hardened --build-arg ARG_CA_NAME=<CA_NAME> --build-arg ARG_STATE=<ST> --build-arg ARG_CITY=<CITY> --build-arg ARG_ORGANIZATIONAL_UNIT=<UNIT> -f docker/Dockerfile.ca .

2. Run the Certificate Authority Setup Container: If certificates have previously been generated and exist in the tak/cert/files path when building the ca-setup-hardened image then certificate generation will be skipped at runtime.

<docker run --name ca-setup-hardened -it -d ca-setup-hardened

3. Copy the generated certificates for TAK Server:

> docker cp ca-setup-hardened:/tak/certs/files files

> [-d tak/certs/files] || mkdir tak/certs/files \
&& docker cp ca-setup-hardened:/tak/certs/files/takserver.jks tak/certs/files/ \
&& docker cp ca-setup-hardened:/tak/certs/files/truststore-root.jks tak/certs/files/ \
&& docker cp ca-setup-hardened:/tak/certs/files/fed-truststore.jks tak/certs/files/ \
&& docker cp ca-setup-hardened:/tak/certs/files/admin.pem tak/certs/files/ \
&& docker cp ca-setup-hardened:/tak/certs/files/config-takserver.cfg tak/certs/files/

TAK Server Database Hardened Container Setup:

1. Building the hardened docker images requires creating an Iron Bank/Repo1 account to access the approved base images. To create an account, follow the instructions in the IronBank Getting Started page. To download the base images via the CLI, see the instructions in the Registry Access section. After obtaining the necessary credentials, run:

< docker login registry1.dso.mil

2. Follow the instructions in the **TAK Server CoreConfig Setup** section and update the <connection-url> tag with the hardened TAK Database container name. For example:

<connection url="jdbc:postgresql://tak-database-hardened-<version>:5432/cot" username="martiuser" password=<password>/>

3. Create a new docker network for the current tak version:

> docker network create takserver-net-hardened-"$(cat tak/version.txt)"

Ensure in the db-utils/pg_hba.conf file that there is an entry for the subnet of the hardened takserver network. To determine the subnet of the network:

< docker network inspect takserver-net-hardened-"$(cat tak/version.txt)"

Or to specify the subnet on network creation:

< docker network create takserver-net-hardened-"$(cat tak/version.txt)" --subnet=<subnet>

4. Build the hardened TAK Database image:

<docker build -t tak-database-hardened-"$(cat tak/version.txt)" -f docker/Dockerfile.hardened-takserver-db .

5. Run the hardened TAK Database container:

< docker run --name tak-database-hardened-"$(cat tak/version.txt)" --network takserver-net-hardened-"$(cat tak/version.txt)" -d tak-database-hardened-"$(cat tak/version.txt)" -p 5432:5432

TAK Server Hardened Container Setup

1. Build the hardened TAK Server image:

< docker build -t takserver-hardened-"$(cat tak/version.txt)" -f docker/Dockerfile.hardened-takserver .

2. Run the hardened TAK Server container:

< docker run --name takserver-hardened-"$(cat tak/version.txt)" --network takserver-net-hardened-"$(cat tak/version.txt)" -p 8089:8089 -p 8443:8443 -p 8444:8444 -p 8446:8446 -t -d takserver-hardened-"$(cat tak/version.txt)"

Configuring Certificates

1. Get the admin certificate fingerprint

> docker exec -it ca-setup-hardened bash -c "openssl x509 -noout -fingerprint -md5 -inform pem -in files/admin.pem | grep -oP 'MD5 Fingerprint=\K.*'"

2. Add the certificate fingerprint as the admin after the hardened TAK server container has started (about 60 seconds)

> docker exec -it takserver-hardened-"$(cat tak/version.txt)" bash -c 'java -jar /opt/tak/utils/UserManager.jar usermod -A -f <admin cert fingerprint> admin'

Useful Commands

*To run these commands on the hardened containers, add the -hardened suffix to the container names.

- View images:

  > docker images takserver

  > docker images takserver-db

- View containers

  All: > docker ps -a

  Running: > docker ps

  Stopped: > docker ps -a | grep Exit

- Exec into container

  > docker exec -it takserver-"$(cat tak/version.txt)" bash

  > docker exec -it takserver-db-"$(cat tak/version.txt)" bash

- Exec command in container

  > docker exec -it takserver-"$(cat tak/version.txt)" bash -c "<command>"

  > docker exec -it takserver-db-"$(cat tak/version.txt)" bash -c "<command>"

- Tail takserver logs

  > tail -f tak/logs/takserver-messaging.log

  > tail -f tak/logs/takserver-api.log

- Restart TAK server

  > docker exec -d takserver-"$(cat tak/version.txt)" bash -c "cd /opt/tak/ && ./configureInDocker.sh"

- Start/Stop container:

  > docker <start/stop> takserver-"$(cat tak/version.txt)"

  > docker <start/stop> takserver-db-"$(cat tak/version.txt)"

- Remove container:

  > docker rm -f takserver-"$(cat tak/version.txt)"

```
> docker rm -f takserver-db-"$(cat tak/version.txt)"
```

4.7 Configure System Firewall

One of the most common problems people have is the system default firewall blocking their traffic.

The full procedure for configuring the firewall is complex and beyond the scope of this guide, and is an important concern for system configuration. Consult your network administrator and/or the firewalld documentation at https://fedoraproject.org/wiki/FirewallD.

The following tips will get you started for lab/field environments.

To verify whether a firewall is running, use the command:

```
> sudo systemctl status  firewalld.service
```

To see what zones are running:

```
> sudo firewall-cmd --get-active-zones
```

If you are working from a fresh OS install, the only active zone is 'public'.

For each each zone, you'll want to enable TCP (and possibly UDP) ports for the inputs in your CoreConfig.xml file, plus the web server's port. For example,

```
> sudo firewall-cmd --zone=public --add-port 8089/tcp –permanent
```

```
> sudo firewall-cmd --zone=public --add-port 8443/tcp --permanent
```

The ports you'll need to open for the default configuration are 8089 and 8443.

Finally, enable your new firewall rules:

```
> sudo firewall-cmd --reload
```

4.8 Software Installation Location

The RPM installer places the TAK server software and configuration in the directory **/opt/tak**. It creates a user named **tak** who is the owner of the files in that directory tree. **Always use this tak user when editing CoreConfig.xml or generating certificates. You can become the tak user by entering:**

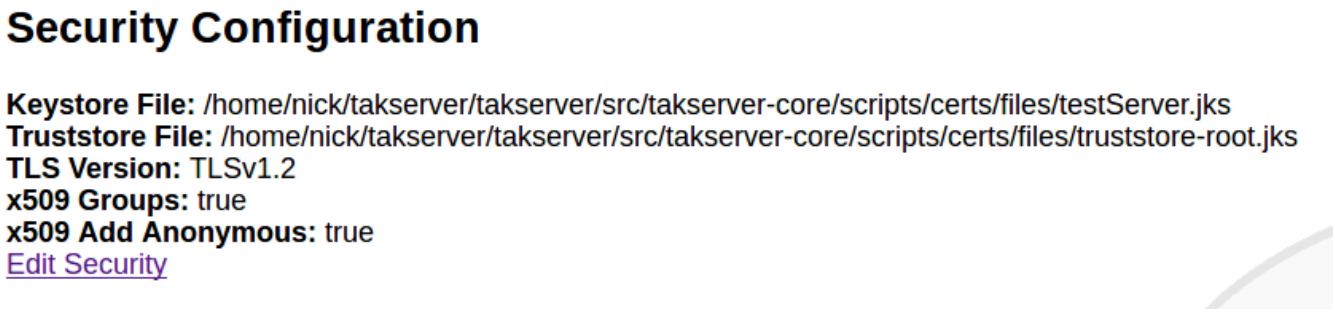
sudo su tak

5 Configuration

Configuration is primarily done through the web interface. Changes made in the interface will be reflected in the /opt/tak/CoreConfig.xml file. If that file does not exist (e.g. on a fresh install), then when TAK Server starts up it will copy /opt/tak/CoreConfig.example.xml. The example has many commented out options. Notable configuration options:

- inputs: In the <network> section there are a series of <input> elements. These define ports the server will listen on. Protocol options are as follows:
  - udp: standard CoT udp protocol; unencrypted
  - mcast: like udp, but has additional configuration option for multicast group
  - tcp: publish-only port; standard CoT tcp protocol; unencrypted
  - stcp: streaming/bi-directional; this is for ATAK to connect to. Unencrypted, for testing only
  - tls: TCP+TLS streaming/bi-directional for encrypted communication with TAK clients
- <auth> : you can use either a flat file or an LDAP backend for group filtering support
- <security>: here you specify the keystore files to use for the secure port(s)

5.1 Configuring Security through Web UI (Certificates/TLS)



Security Configuration Web interface

Security and authentication options for TAK Server can be set up using a web interface. To access this page in the menu bar go to Configuration > Manage Security and Authentication Configuration. This page will contain both Security and Authentication configuration current values. To modify the Security Configuration click "Edit Security". This will allow changes to the server's certificates, the version of TLS used, x509 Groups settings and x509 Add Anonymous settings. Note: Changes made here will only take effect after a server restart.

5.2 Group Filtering

TAK Server has the ability to segment users so they only see a subset of the other users on the system. This is achieved by assigning groups to individual connections. If ATAK-A shares common group membership in at least one group with ATAK-B, they share data with each other. If not otherwise specified, all connections default to being in the special "__ANON__" group (note 2 underscores as prefix and postfix). There are three ways to assign groups to a connection:

- Assigning <filtergroup> elements to <inputs>: this is simple, but provides no access control if you have multiple ports configured on the same server.
- Active Directory / LDAP / Flat file with additional authentication message
- Active Directory / LDAP / Flat file without additional authentication messages (uses certificate-based identification)

Details on the three options:

5.3 Group Assignment by Input

<inputs> can drive group filtering, even without authentication messages. Version 1.3.0 added group filtering based on LDAP groups. This necessitated a new authentication message from ATAK. This worked for the streaming connections, but wouldn't work for the connection-less UDP traffic.

We added an additional configuration option for inputs to allow the connection-less traffic to be routed according to the group filtering. An input definition like this:

```
<input _name="stdudp" protocol="udp" port="8087">
    <filtergroup>TEST1</filtergroup>
</input>
```

would have the effect of making every CoT event that came into the 'stdudp' input be associated with the "TEST1" group *instead of* the anonymous group. If there is no filtergroup specified, the default is the old behavior, which is a special anonymous group. The anonymous group has a name "__ANON__" that can be used to explicitly add it back in if needed. The filtergroup option can be used with the streaming input protocols as well (stcp, tls), the effect of which is that any subscriptions made by connecting to that port inherit the filter group from the input. <filtergroup> cannot be used in conjunction with the "auth" attribute on the same input. You can however use them on separate inputs, for example:

```
<input _name="stdudp" protocol="udp" port="8087">
  <filtergroup>CN=TAK1,DC=...</filtergroup>
</input>
<input _name="sec" protocol="tls" port="8089" auth="ldap" />
```

Note that when trying to interact with LDAP groups, you need to use the fully qualified group name that LDAP/ActiveDirectory reports.

5.3.1 Input Configuration UI

Inputs can be dynamically added, modified and deleted in the TAK Server user interface, under the menu heading **Configuration → Input Definitions**. The UI also shows activity for each input, in terms of number of reads and messages. For the streaming protocols (stcp, tls), the activity is the sum for *all* connections made using that particular input port.

5.4 Group Assignment Using Authentication Messages

If ATAK or another TAK client is configured to send an authentication message after establishing a connection to TAK Server, the username and password credentials contained in that message will be used, in conjunction with an authentication backend, to determine the group membership of a user. TAK Server will then filter messages according to common group membership, in a similar fashion to filtering configured by <filtergroups> for a given <input>.

TAK Server can be configured at the input level to expect authentication messages with each new client connection. If the authentication message is not sent, or is invalid, the client will be disconnected. The "auth" attribute on the input indicates which authentication strategy will be used when processing authentication messages. A value of "file" tells TAK Server to validate authentication credentials using the flat-file backend. A value of "ldap" indicates that an Active Directory or LDAP server should be used to validate the credentials.

For example, this input definition specifies streaming TCP, encrypted with TLS, authenticating the user with a client certificate and also requiring an authentication message, and using the LDAP authentication backend:

```
<input _name="ldapssl" protocol="tls" port="8091"
auth="ldap"/>
```

5.5 Group Assignment Using Client Certificates

TAK Server can be configured to use only the information contained in a client certificate, when looking up group membership for a user. In this case, the TAK client is configured to use TLS and a client certificate when connecting to TAK server, but does not send an authentication message. This eliminates the requirement to cache credentials on the device, or enter credentials prior to establishment of each new connection. TAK Server will then filter messages according to common group membership, in a similar fashion to input-level filtering with filter groups. When analyzing the X.509 client certificate presented by the TAK client, TAK Server will look at the DN attribute in the certificate, extract the CN value from the DN (if present). The CN is regarded as the username, and is used to look up group membership in authentication backends. For example, consider this DN in a client certificate:

```
CN=jkirk, O=TAK, C=US
```

The CN value **jkirk** will be used as the username. The process for deciding which authentication backend to use depends on whether or not an Active Directory (AD) LDAP configuration is present in CoreConfig.xml. Valid service account credentials must be configured in CoreConfig.. If AD authentication is configured, the user account is matched by the **sAMAccountName** LDAP attribute.

At client authentication time, if groups are found in AD for the user, those groups are used by TAK Server. If no groups are found, the flat-file authentication backend is searched for a match on the username. If no groups are found for the user in either repository, the user is assigned to the __ANON__ group.
When configuring the input, a TLS input with an auth type of **x509** directs TAK Server to use the client certificate for both authentication and group assignment. On the input configuration, the on or example, this input definition specifies streaming TCP encrypted with TLS, authenticating the user with a client certificate and also requiring an authentication message, and using the LDAP authentication backend:

```
<input _name="ldapssl" protocol="tls" port="8091"
auth="x509"/>
```

5.6 Authentication Backends

5.6.1 File-Based

There is now a flat-file option available to inputs. Previously the only valid value for the <input> “auth” attribute was “ldap”. “file” is now another valid value. The example configuration file (CoreConfig.example.xml) contains an example of how to configure the File-based backend.

A utility for creating and maintaining that flat file is included in the release. Run

```
sudo java -jar /opt/tak/utils/UserManager.jar
```

and look at the various options for the 'offlineFileAuth'

5.6.2 Active Directory (AD) / LDAP

TAK Server can be configured to use an Active Directory or LDAP server to authenticate users, and assign groups. LDAP configuration for TAK Server varies depending on the configuration of the AD or LDAP server. Here is an example. Note that it contains credentials for a service account. This is required for group membership lookup using a client cert:

```
<auth>

  <ldap url="ldap://a.b.com/ou=MyUserOU,DC=a,DC=b,DC=com"

      userstring="{username}@MYDOMAIN" updateinterval="60000"  style="AD"

      serviceAccountDN="mysearchuser@MYDOMAIN"

      serviceAccountCredential="password"

   />

</auth>
```

5.6.3 Configuring LDAP through Web Interface

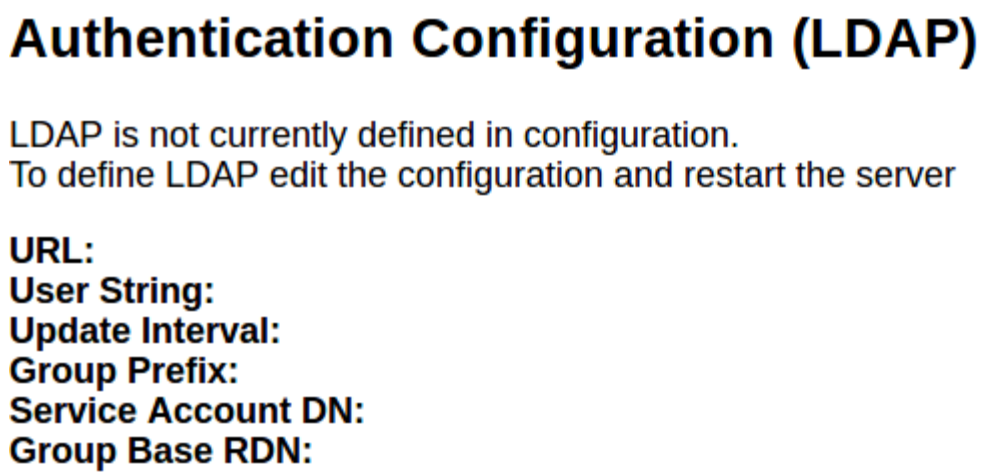
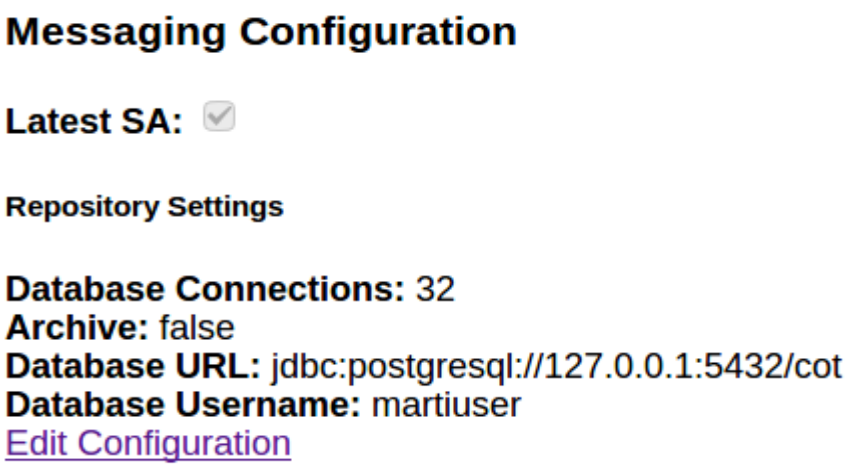


Figure 1: Authentication Configuration Web interface

The LDAP configuration can be changed through an easy to use web page. To access this go to Configuration > Manage Security and Authentication. Under the Authentication heading will be the current LDAP configuration (the values will be empty if LDAP is not configured yet). Click on "Edit Authentication" to be directed to a form to enter desired LDAP settings. Note: Changes made here will only take effect after a server restart.



Messaging Configuration Web interface

Messaging/Repository settings configuration can be done through the input definitions page. To get there go to Configuration > Input Definitions in the menu bar. This page displays the current input definitions at the top and at the bottom the current configuration of Messaging and Repository settings are displayed. To edit these setting click "Edit Configuration". Note: Changes made here will only take effect after a server restart.

5.7 Configuring Messaging and Repository Settings through Web UI

5.8 Optionally Disabling UI and WebTAK on HTTPS Ports

TAK Server can be configured to optionally disable the Admin UI, non-admin UI or WebTAK on any HTTPS connector (port). These options can be used to fine-tune the security profile for each HTTPS connector. For example, the admin web interface can be moved to an alternate port that is protected by a firewall from access on the public Internet.

In the CoreConfig.xml, the *enableAdminUI*, *enableWebtak,* and *enableNonAdminUI* attributes on each <connector> can be used to optionally disable access to any of these three functions for a given HTTPS connector port. The default value for each of these attributes is *true*, so by default these functions are available.

Usage Examples:

Disable webtak on port 8443:

<connector port="8443" _name="https" enableWebtak="false" />

On port 8452, disable admin UI, but enable WebTAK and non-admin UI:

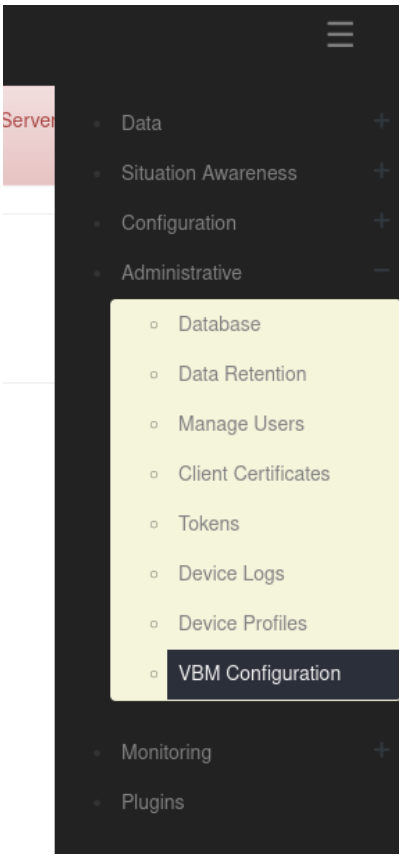
<connector port="8452" _name="https" enableAdminUI="false" />

Disable WebTAK on OAuth port 8446:

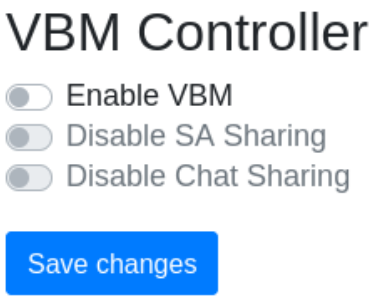
<connector port="8446" clientAuth="false" _name="cert_https" enableWebtak="false"/>

5.9 VBM Admin Configuration

TAK Server can allow for additional filtering of cot messages recieved from inputs (server ports) and data feeds by using the VBM Configuration page in the Admin UI. To navigate there, go to Administative > VBM Configuration as shown below.



You will then see the following options.

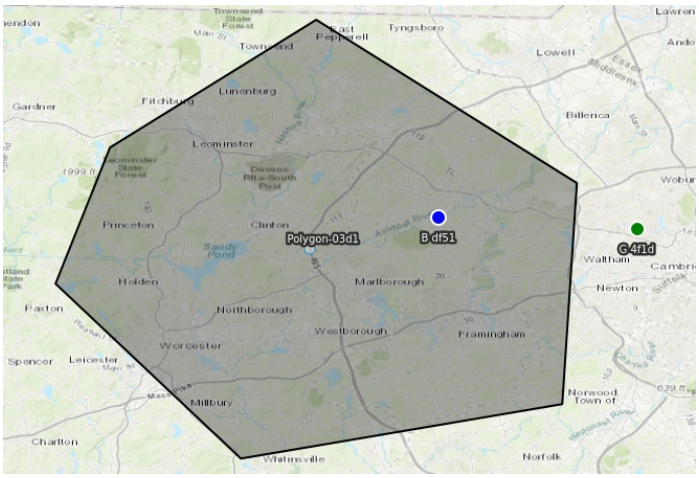


To modify the VBM configurations select the checkbox next to the desired option and when you're finished click "Save changes".

NOTE: "Disable SA Sharing" and "Disable Chat Sharing" will only be used if "Enable VBM" is selected.

The VBM options have the following impact:

If "Enable VBM" is selected, messages recieved on a data feed are only brokered to clients which are subscribed to the mission if the message falls within the bounding box specified by the mission. For example, the message represented by the blue dot would be passed on while the message represented by the green dot would be filtered out for the mission with the displayed bounding box only if "Enable VBM" is turned on.



The second two options are only activated if "Enable VBM" is on and they refer to messages recieved from inputs (server ports). These options filter messages based on whether they are chat messages. Chat messages in this context are cot messages which have a type set to "b-t-f".

If "Disable SA Sharing" is selected, messages recieved from inputs are passed on if the message is a chat message as defined above.

If "Disable Chat Sharing" is selected, messages recieved from inputs are passed on if the messages is **NOT** a chat message as defined above.

These options are not mutually exclusive. Therefore, having both selected will filter out all messages recieved on inputs.

6 WebTAK

The WebTAK front-end application is bundled with TAK Server. The WebTAK back-end WebSockets networking channel and APIs are provided by TAK Server. WebTAK must be accessed using https.

WebTAK can be accessed either with X.509 client certificates (default https port 8443), or by username-password access using OAuth (https port 8446).

In either case, TAK Server must be configured with a server certificate and truststore (see Appendix B).

In the Admin UI menu, use **Situation Awareness -> WebTAK** to access WebTAK.

7 Device Profiles

TAK Server can now assist in provisioning ATAK devices through Device Profiles. The Device Profile Manager (under Administrative Menu, Device Profiles) allows administrators to build profiles that can be applied to clients when enrolling for certificates, and when connecting to TAK server. The Profile consists of a sets of files, which can include settings and data in any file format that is supported by TAK Mission Packages. Profiles can be made public or restricted to individual Groups.

When an ATAK device enrolls for client certificate, or optionally after connecting to TAK server, TAK server will return all profiles that need to be applied to the device. The TAK server administrator can also push a profile to a connected user by clicking the Send link within the Device Profile Manager.

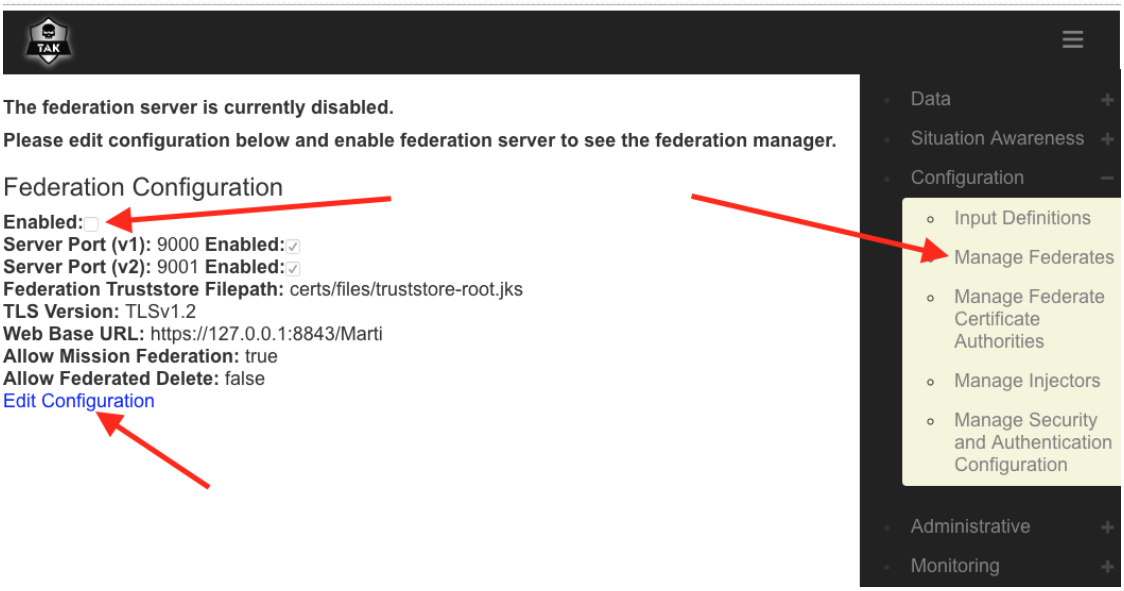
8 Federation

Federation will allow subsets of ATAK users who are connected to different servers to work together, even though each TAK server instance (hereafter refered to as 'federates') may be run by independent organizations / administrative domains. It brings some of the following benefits/restrictions:

1. Each administrative domain does not need to share anything about their internal structure (e.g. LDAP/Active Directory information / users) with the other administrative domain.
2. Each administrative domain has control over what data they share with the other domain, but has no control over what the other administrative domain does with data that is shared.
3. It requires no reconfiguration of ATAKs connected to either TAK Server, and the mechanism for connecting the TAK Servers does not allow direct connections of ATAK devices from the other administrative domain.

8.1 Enable Federation

The first step is to enable federation on your TAK server. To do this, first go the Configuration > Manage Federates page. If federation is not yet enabled, click on the Edit Configuration button. This is also where you can pick the ports for each federation protocol.



Do not forget to restart the server after changing the federation configuration in order for the changes to take effect!

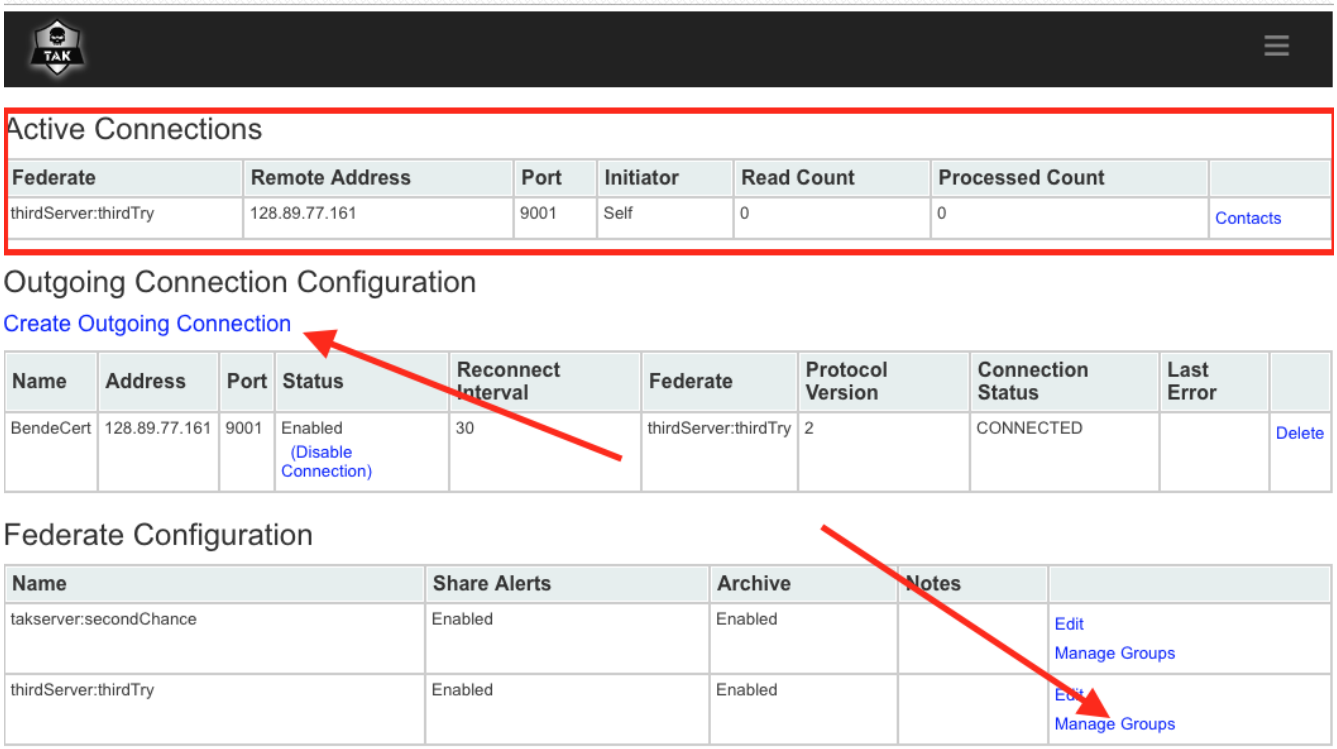
8.2 Upload Federate Certificate

In order for the federate servers to trust each other and their ATAK clients, they must share each others certificate authorities (CAs) in order to create a separate federate trust-store. One of the key components to how TAK Server satisfies all the restrictions is that we use one trust-store for local users, and one for Federates. The trust-store contains all the valid CAs that you will allow client certificates from. By having separate trust-stores, we can have the Federation channels allow connections with certificates from “outside” CAs, while not allowing ATAKs with certs from those “outside” CAs to make direct connections to our server.

Generally, we share the public CA, which you can find at **/opt/tak/cert/files/ca.pem**, via some third channel such as email or a USB drive. Once you have traded CAs, go the the Manage Federate Certifate Authorities page and upload the CA of the federate you want to connect to.

8.3 Make the Connection

Now that we have enabled federation and shared our CA with the other TAK server authority, we are ready to make the connection and start sharing. For this step, only ONE of the servers creates an outgoing connection to the other. If you are starting the connection, go back to the Manage Federates page where you enabled federation from step 1. You will now see three sections, Active Connections, Outgoing Connection Configuration, and Federate Configuration. To create an outgoing connection, click on the corresponding link, and enter in the address and port of the destination server. You can also pick the protocol version (make sure it is the right protocol for the port you are connecting to!), reconnection interval (how long between retries if the connection is lost), and whether or not the connection will be enabled on creation.



Active Connections

Federate	Remote Address	Port	Initiator	Read Count	Processed Count	
thirdServer:thirdTry	128.89.77.161	9001	Self	0	0	Contacts

Outgoing Connection Configuration

Create Outgoing Connection

Name	Address	Port	Status	Reconnect Interval	Federate	Protocol Version	Connection Status	Last Error	
BendeCert	128.89.77.161	9001	Enabled (Disable Connection)	30	thirdServer:thirdTry	2	CONNECTED		Delete

Federate Configuration

Name	Share Alerts	Archive	Notes	
takserver:secondChance	Enabled	Enabled		Edit Manage Groups
thirdServer:thirdTry	Enabled	Enabled		Edit Manage Groups

Now that you have created and started a connection, you will notice that no information is yet flowing between federates. This is because you and your fellow federate must specify which filtering groups you will allow to flow out of and into your server. To manage this, click on the Manage Groups link in the corresponding row of the Federate Configuration section. Here you can specify the groups, including the special __ANON__ group if you want. Once both servers have configured the groups, traffic will start to flow. A server restart is not necessary for these changes to take effect.

Federate Groups

You are configuring groups for federate: **thirdServer:thirdTry**

A federate is another TAK installation with which you wish to share events.

When events arrive from this federate, you may direct them to devices in the local inbound groups you configure below. Similarly, you may send events to this federate from devices in the outbound groups you configure.

Note: Adding and Removing groups affects the runtime environment immediately, and changes are saved to the configuration file. This configuration is associated with connections based on the certificate provided when the TLS connection is initiated. What is stored in the configuration file is the SHA256 fingerprint of the certificate.

You must select at least one inbound or outbound group to effectively activate this federate.

Group

Search LDAP

Direction Both (Inbound/Outbound)

Add Group

Groups configured for this federate:

Group	Direction	
__ANON__	INBOUND	Remove
__ANON__	OUTBOUND	Remove

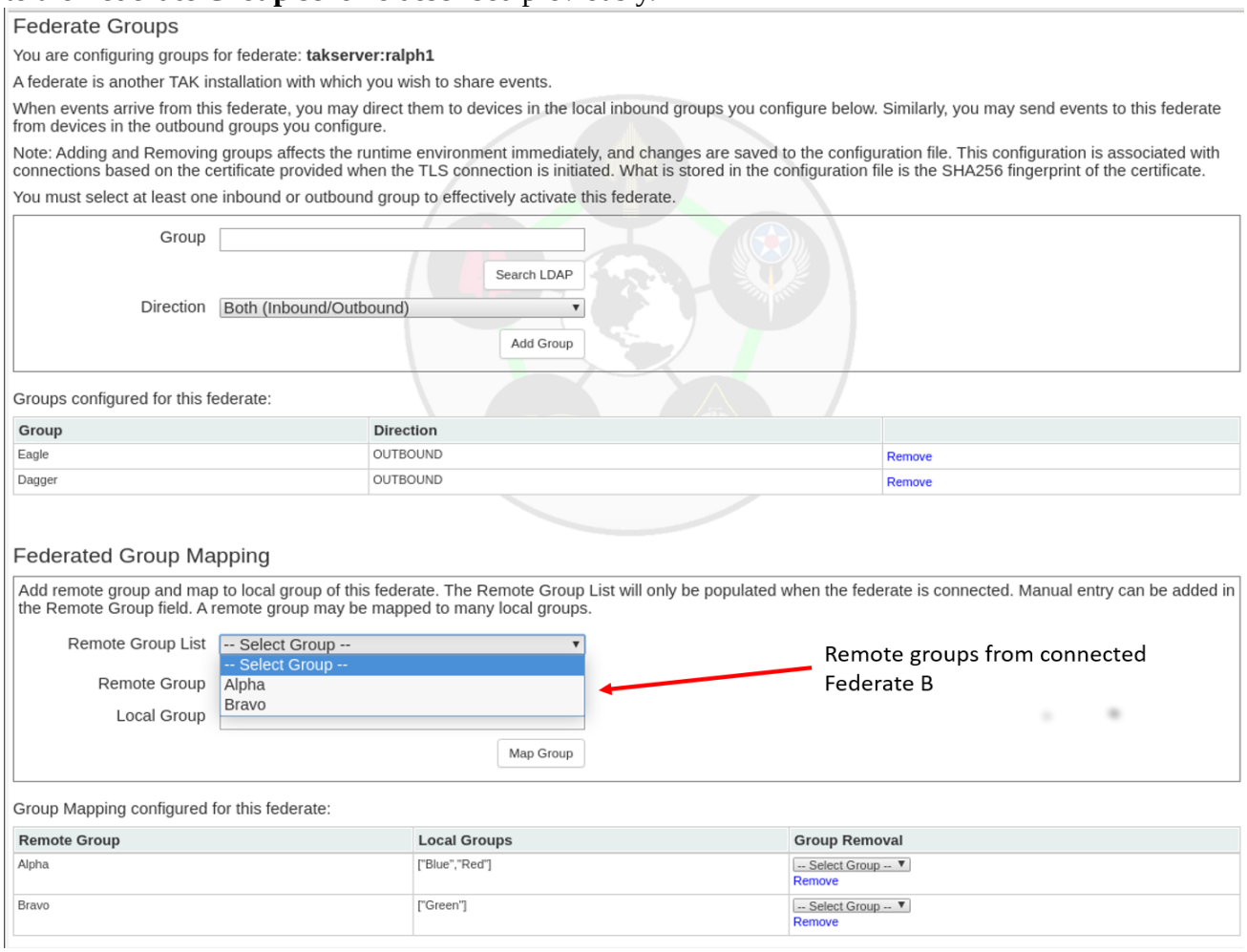
Back To Federates

8.4 Federated Group Mapping

The flow of traffic between Federates may be directed using end-to-end group mapping. The **Federated Group Mapping** section is on the **Federate Groups** page.

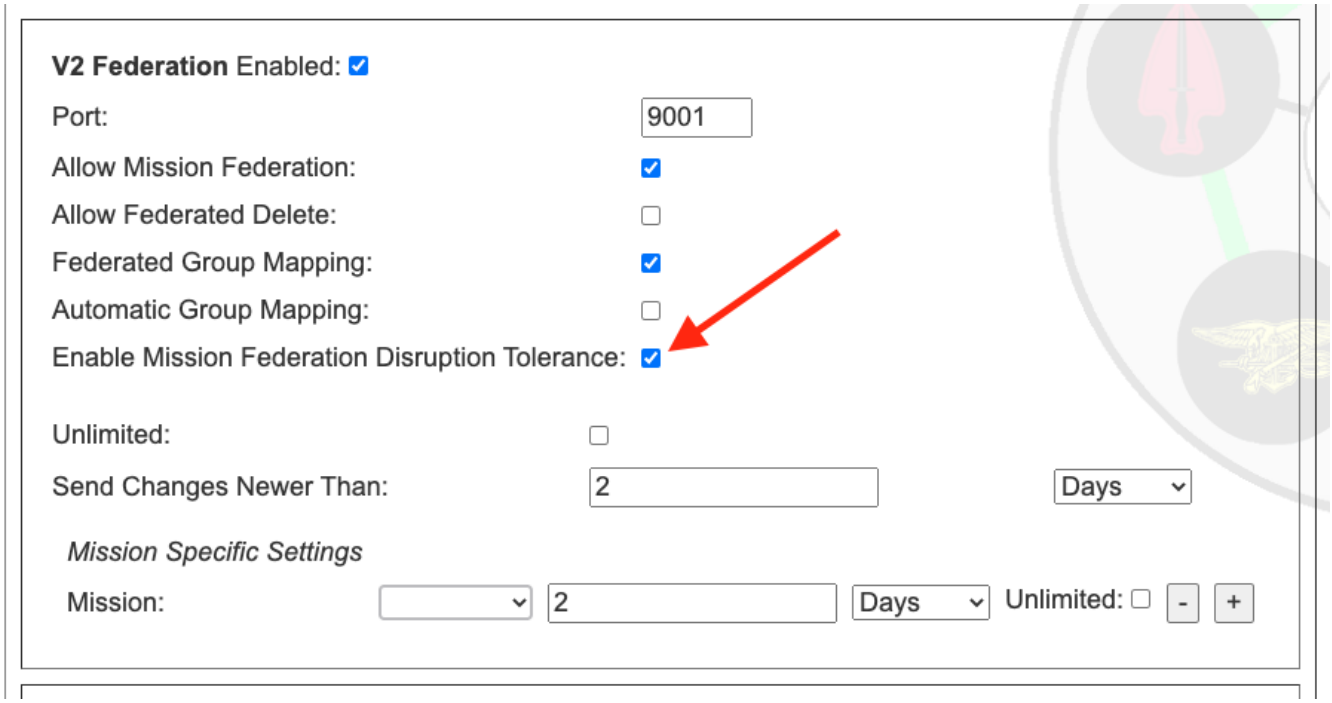
Groups are exchanged during active connections between Federates. The *remote* groups will appear in the 'Remote Group List' drop down in the **Federated Group Mapping** section. Connected Federates must have Federated Group Mapping enabled in order for the Federates to exchange their respective *remote* groups. This parameter is in the **Federation Configuration** section in the Configuration > Manage Federates page.

To configure the end-to-end mapping, select a remote group and map it to a local Federate group. Remote groups may also be entered directly in the 'Remote Group' field. A single remote group can be mapped to many local groups. Additionally, multiple end-to-end group mappings may be defined. With a group mapping configured, traffic from the remote group will only flow to the mapped local group(s). Note: if no incoming traffic matches the remote groups configured, the federation traffic will fall back to the **Federate Group** scheme described previously.



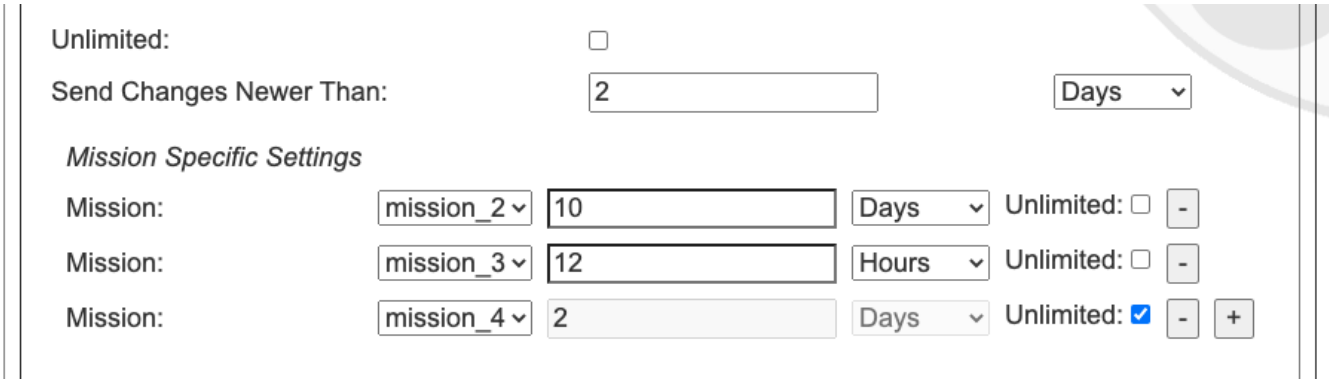
8.5 Mission Federation Disruption Tolerance

Traffic between federated servers may be disrupted, and updates to missions could happen during that disruption. Mission federation disruption tolerance will update each server with changes to federated missions that occured during the disruption. To enable this feature, check the box in the Federation Configuration page:

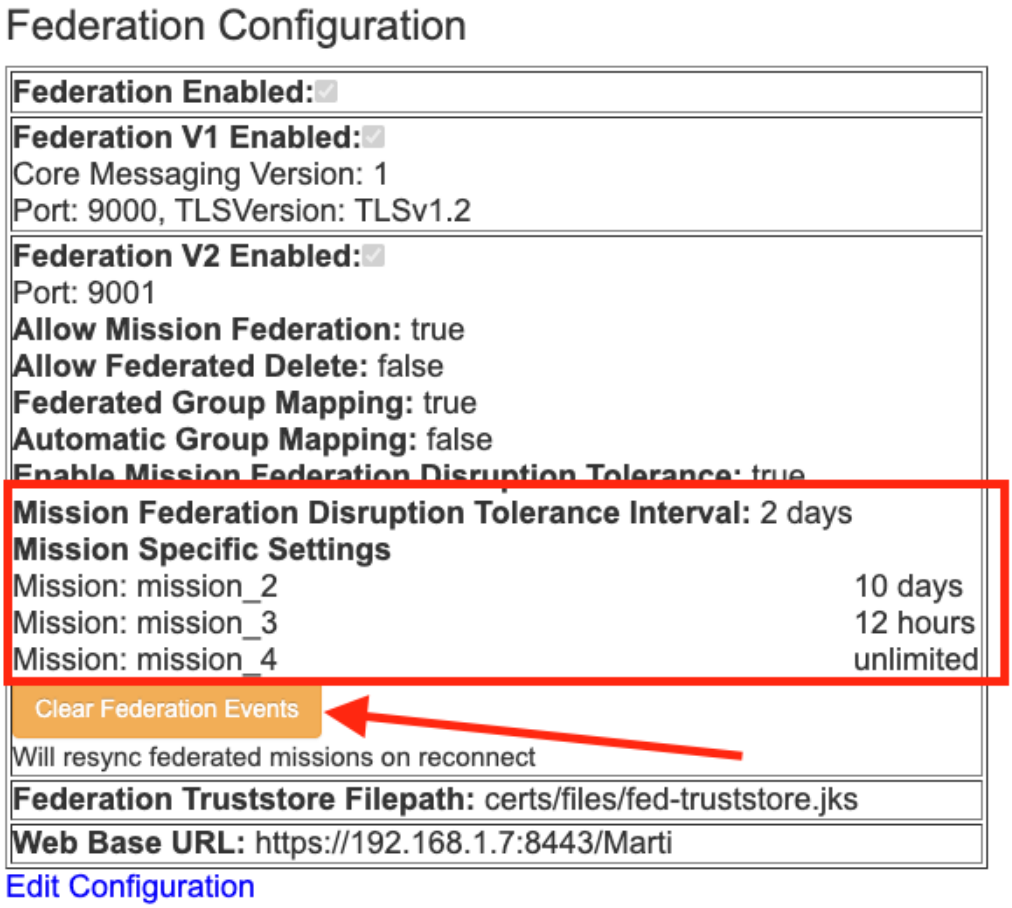


Sending all the changes that occured between disruptions could potentially take a lot of bandwidth, so by default, we limit the changes to those that occured within the last 2 days. For example, if a disruption lasted 3 weeks, we would only send the changes from the previous 2 days. However, if the disruption only lasted a few hours, only the changes since the last disruption would be sent. If the Unlimited checkbox is checked, then all changes since the last disruption would be sent. The 2 day limit can be changed to any length with the Send Changes Newer Than setting, and the period can be selected as days, hours, or seconds.

It is also possible to override the global setting for a particular mission, if so desired.



For example, in the above image, we see that mission_2 will send updates up to over the last 10 days, mission_2 only over the last 12 hours, and mission_4 will send all updates since the last disruption. Any mission that is not listed, and any subsequently added mission, will follow the global setting of 2 days, as set above.



The Clear Federation Events button will reset the disruption history for federation. This means that on the next reconnection, the server will send the max allowed mission changes according to the Mission Federation Disruption Tolerance settings. In the above case, that would be 10 days for mission_2, 12 hours for mission_3, and the entire change history for mission_4. For all other missions this would be 2 days worth of mission changes.

8.6 Data Package and Mission File Blocker

Data packages, mission packages, and missions can be federated between servers and their respective ATAK clients. These packages may contain configuration files such as ATAK .pref files that can result in the distribution of unwanted configuration changes to ATAK devices. A filter can be enabled to block files by file-type. To enable this feature, check the Data Package and Mission File Blocker box in the Federation Configuration page. The default file extension value is 'pref'. This can be changed by entering a new file type, clicking on the 'Add' button to add the entry, and clicking on the 'Save' buton at the bottom of the Federation Configuration page.

Data Package and Mission File Filter

Enable Data Package and Mission File Filter: ☑

File Extension: pref

Add File Extension: Add *The default extension is .pref.*

Enter the extension of the file type to block in federated data packages and missions.

8.7 Federation Example

The figure below shows a connectivity graph of two distinct administrative domains. Each administrative domain has multiple sub-groups (e.g. “CN=Alpha”) utilizing the group-filtering. The color coding indicates the CA that is used to sign the certificate used for connections. Enclave 1's CA signs ATAK client certs and a server certificate. Enclave 2's CA also signs ATAK client certs and a server cert. The trust-store listing the allowed CAs for the “User Port” only contains a single CA (i.e. *Enclave 1 CA* for Enclave 1). To federate the servers, Enclave 1 and Enclave 2 send each other the “public” CA cert. Those certificates are put in a separate trust store that is used only for federation connections. The “Fed. Port” is configured with this separate trust-store.

The server cert from each administrative domain can now be used to connect to the “Fed. Port” of the other domain.

Figure 1: Federation Example

8.8

Alternate Configuration

The first example had each federate using the same CA and server certificate for local and federate connections. If you are very paranoid, or don't want to share anything about the crypto being used for local clients, you can have a wholly separate CA+server certificate chain that is used for federation. Figure 5 shows how this would work.

Figure 2: Alternate Federation Example

This adds some complexity, but can be used if you don't want to expose your 'internal' CA to the organizations that you are federating with.

9 Metrics

The TAK Server Metrics Dashboard is available in the Monitoring menu. The dashboard continuously renders the following information:

Server Start Time and Server Up Time: This tell you when the server was turned on and how long it has been operating.

Clients Connected: This tells you how many connections your client is currently servicing. This corresponds to the number of clients that are displayed in the client dashboard.

Heap Usage: TAK server runs inside one or more Java Virtual Machines (JVM). Heap Commited is how much heap memory in MB is allocated to the API process for TAK Server, and Heap Used is how much of that is currently being used.

Network I/O and Reads/Writes: This tells you how much TCP and UDP traffic the server is currently handling, as well as a brief history.

CPU Usage: How much of the CPU of the machine the server is running on is currently being used.

Figure 3: Metrics Dashboard

10 Logging

TAK Server provides several log files to provide information about relevant events that happen during execution. The log files are located in the `/opt/tak/logs` directory. This table shows the name of the log files, and their function.

Name of Log File	**Purpose**
takserver-messaging.log	Execution-level information about the messaging process, including client connection events, error messages and warnings.
takserver-api.log	Execution-level information about the API process, including error messages and warnings.
takserver-messaging-console.log	Java Virtual Machine (JVM) informational messages and errors, for the messaging process.
takserver-api-console.log	Java Virtual Machine (JVM) informational messages and errors, for the API process.

11 Group Filtering for Multicast Networks

The proxy attribute on the CoreConfig input element (<input … proxy="true" … />) was removed in TAK Server 4.1. The intent of the proxy attribute was to control bridging of multicast networks and federating multicast data. As TAK Server's group filtering capabilities have evolved, having a dedicated proxy attribute is no longer needed. Using filtergroup on the mcast input you can achieve greater control over multicast traffic.

The default behavior in TAK Server 4.1 and higher is to put multicast traffic in the __ANON__ group. You can use a filtergroup on the mcast input to put your mcast traffic into a dedicated multicast group, for example:

<input auth="anonymous" _name=" SAproxy " protocol="mcast" port="6969" group="239.2.3.1">

<filtergroup>__MCAST__</filtergroup>

</input>

Then add the __MCAST__ group as a filtergroup on any other inputs you wanted to share multicast traffic with. For example, to share multicast traffic with the tls/8089, configure your input filtergroups as follows:

<input auth="anonymous" _name="stdssl1" protocol="tls" port="8089" archive="true">

<filtergroup>__ANON__</filtergroup>

<filtergroup>__MCAST__</filtergroup>

</input>

This same approach works for federations. You can __MCAST__ as an outboundGroup on any federates that you wanted to share multicast traffic with. Using the filtergroup approach allows for creation of input specific multicast groups, allowing control of how messages from multicast networks are routed.

12 OAuth2 Authentication

TAK Server provides OAuth2 Authorization and Resource server capabilities using the OAuth2 Password authentication flow. OAuth2 integration works with existing authentication back ends, allowing TAK Server to issue tokens backed by the File or LDAP authenticators. TAK Server issues JSON Web Tokens (JWT) signed by the server certificate, allowing external systems to validate tokens against the server's trust chain. The OAuth2 token endpoint is available at https://<takserver>:8446/oauth/token.

13 User Management UI

Overview

The User Management UI provides an intuitive drag-and-drop mechanisms for managing TAK user accounts. The tool is integrated within TAK Server and can be accessed from the TAK main menu, under Administrative >> Manage Users. Users need to have an admin role to access the tool. Currently the User Management UI supports only file-based users and not LDAP/AD users.

The tool allows TAK administrators to create, manage, inspect and delete TAK user accounts. More specifically, the tool allows TAK administrators to:

- View, filter and search for existing user accounts and groups.

- View a list of users in each group.

- Change password for each user account.

- View and update groups (IN group, OUT group and both) for each user account.

- Delete user accounts.

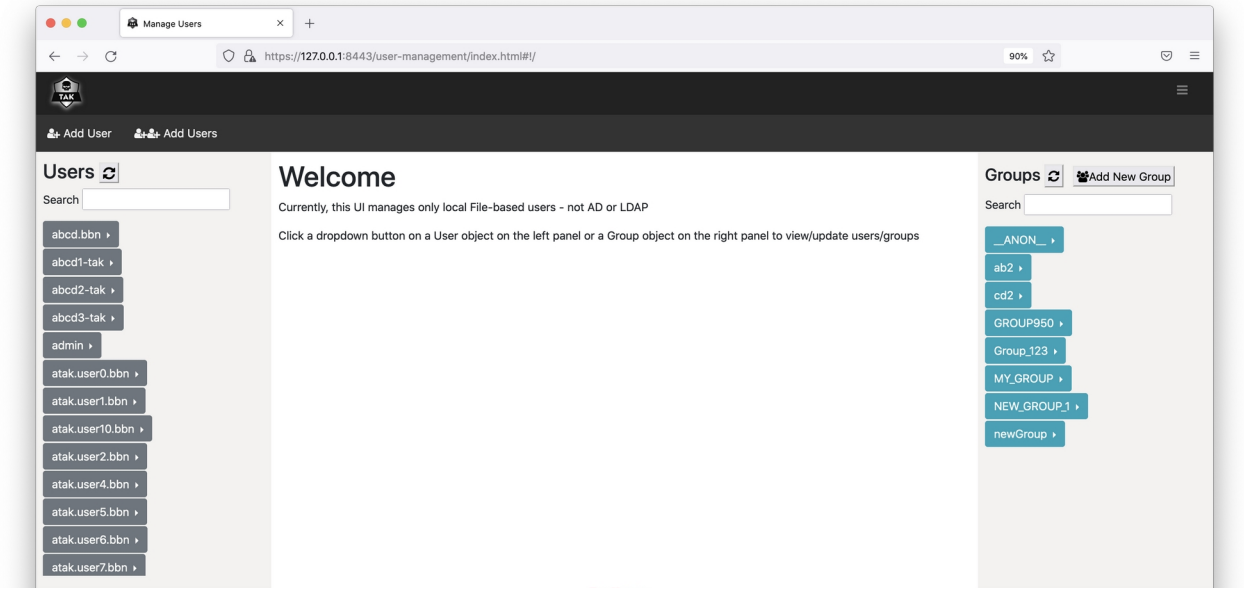
- Create a new user account with a specified password and groups. Password complexity is checked to confirm compliance.

- Create new user accounts in bulk with username following a pattern. System uses password generation mechanism to create passwords that meet TAK password complexity requirements. System produces output file with user/password combos as a one-time downloadable item, after which system forgets the un-hashed passwords.

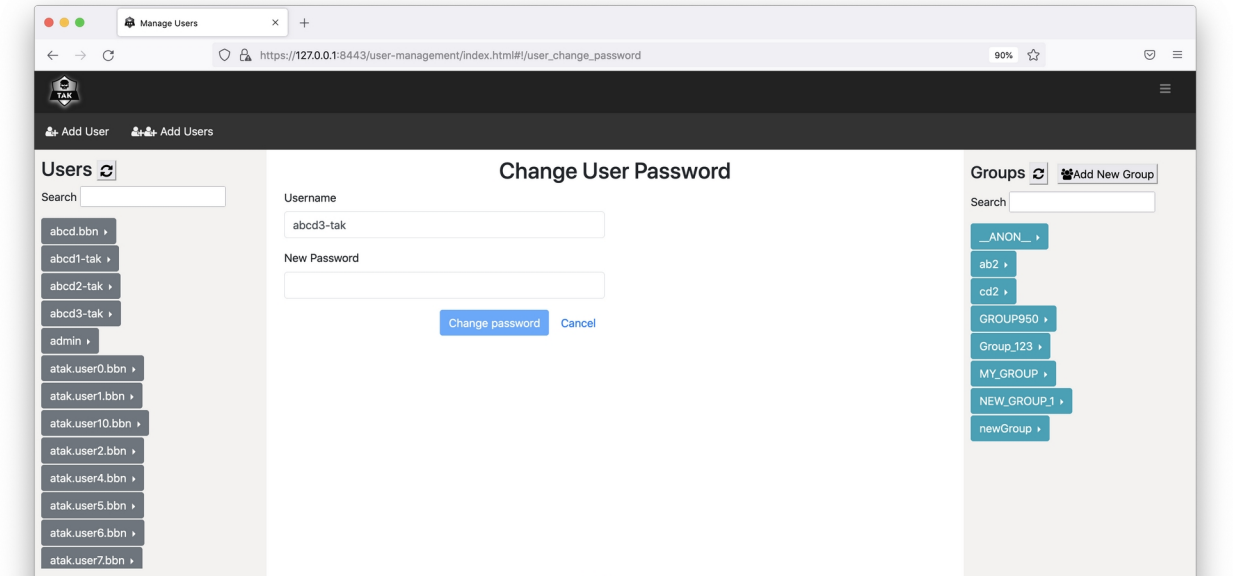
- Create new groups.

Usage

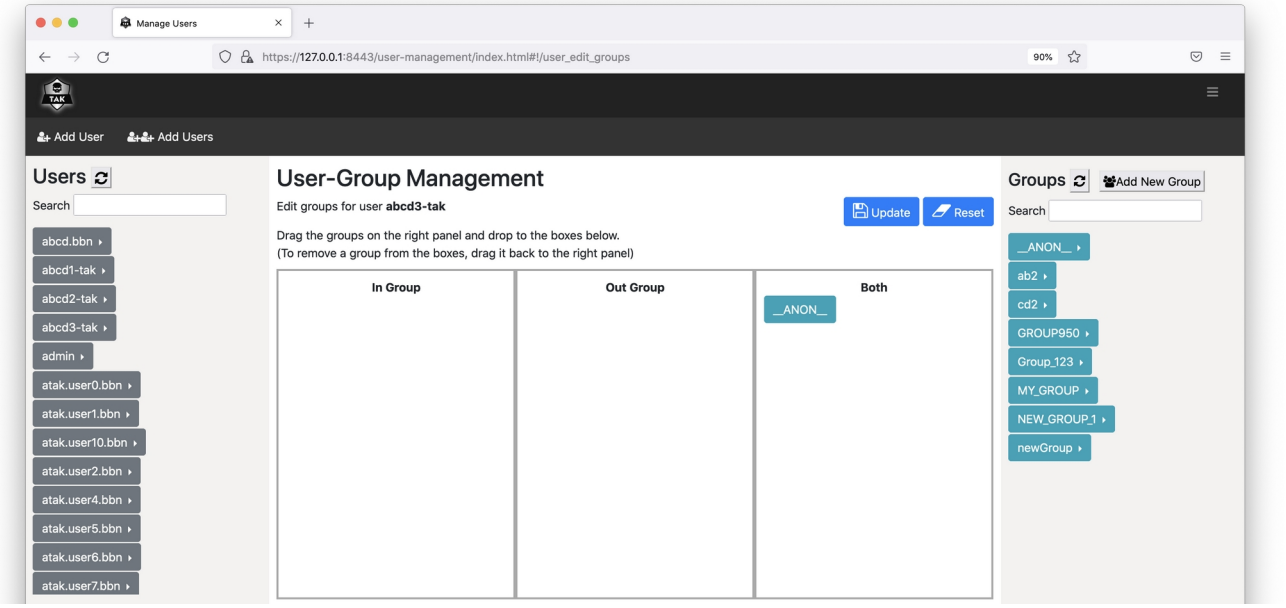
The below figure shows the main page of the User Management UI. The left panel lists all user accounts, which can be filtered using the Search box on the top. The right panel lists all existing groups, which can be filtered using the Search box on the top.



To change user's password, click on the arrow right next to the username and select "Change password".

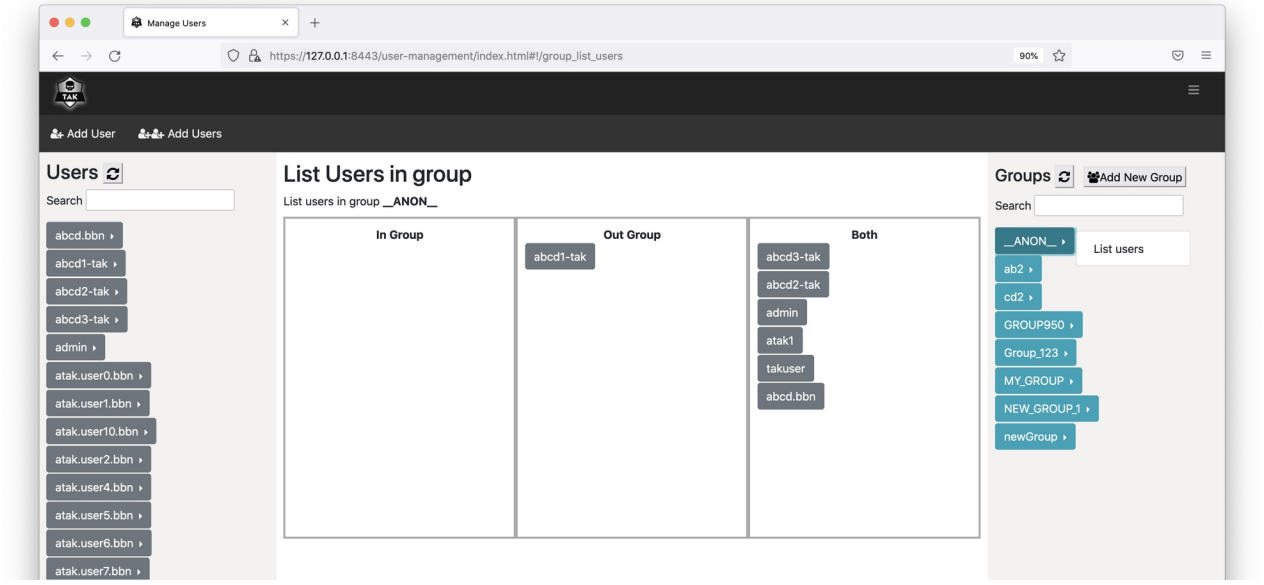


To view/edit groups for a user account, click on the arrow right next to the username and select "View/Edit groups". You can drag the groups from the right panel and drop to one of the three boxes in the middle panel. Click on "Reset" button to bring the UI back to showing the current groups of the user. Click on "Update" button to update the groups of the user.

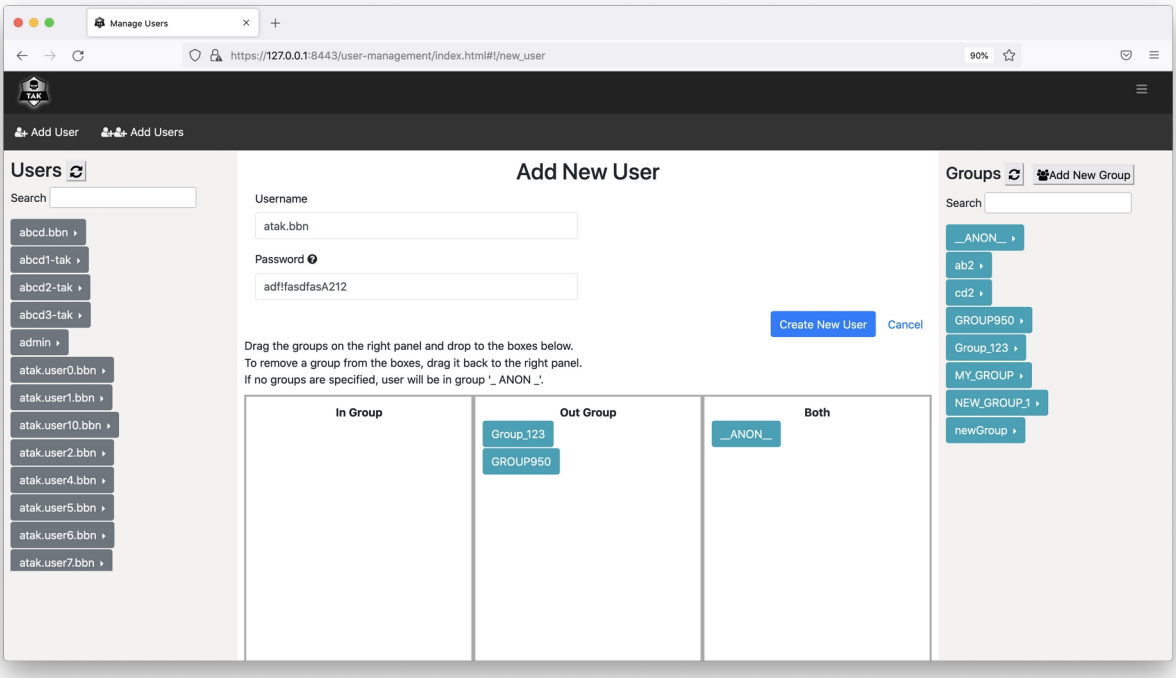


To delete an account, click on the arrow right next to the username and select “Delete User”. You will be prompted to either confirm or cancel the action.

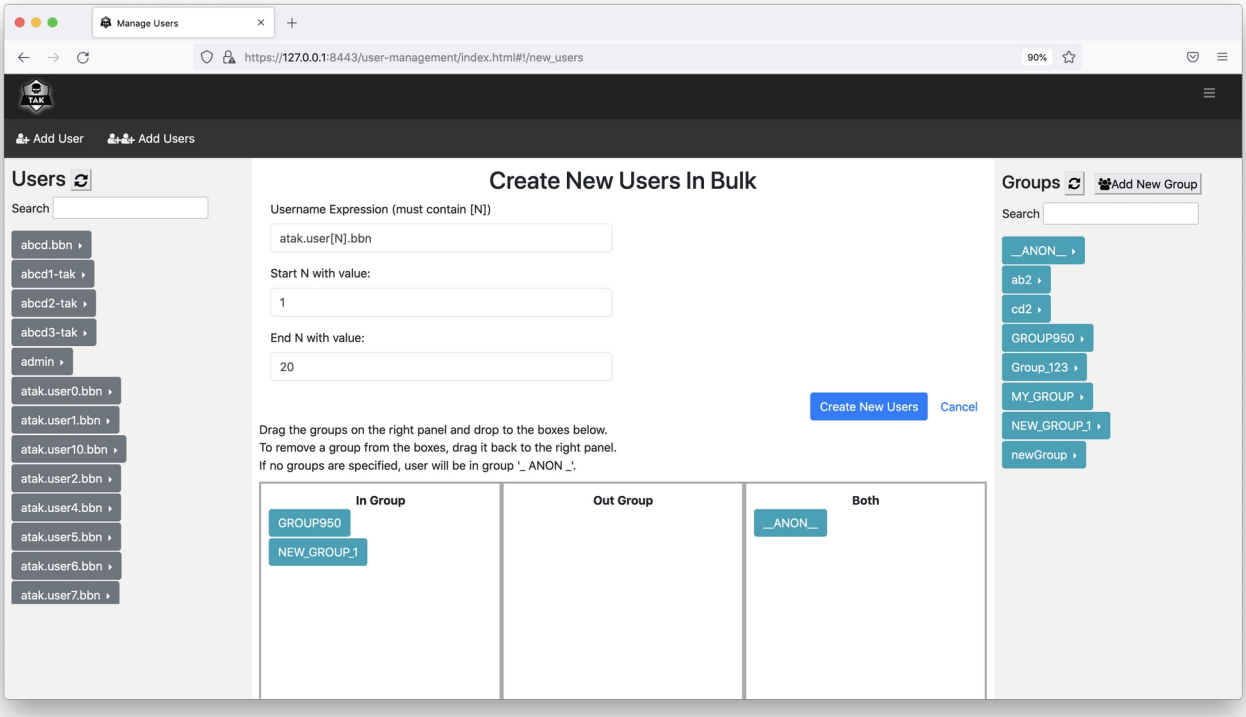
To list all users in a group, click on the arrow right next to the group name and click on “List users”.



To create a new user, click on “Add User” on the menu bar.



To create new users in bulk, click on “Add Users” on the menu bar.



14 Data Retention Tool

Information regarding the use of the Data Retention Tool is available on the tak.gov wiki:

https://wiki.tak.gov/display/TPC/Data+Retention+Tool

Appendix A: Acronyms and Abbreviations

ATAK – Android Team Awareness Kit
CA – Certificate Authority (for digital certificates)
CN – Common Name (of a digital certificate)
CoT – Cursor-on-Target, an XML-based data interchange format
CRL – Certificate Revocation List
DoD – Department of Defense (United States)
DISA – Defense Information Systems Agency
ESAPI – Enterprise Security Application Programming Interface
EPL – Evaluated Products List
HTTP – Hypertext Transfer Protocol
IA – Information Assurance
IP – Internet Protocol
IPv4 – Internet Protocol, version 4. The commonly-used version of IP, in which addresses consist of four integers from zero to 255 (inclusive), separated by periods, such as 192.168.123.4
JCE – Java Cryptography Extensions
JDK – Java Development Kit, a JRE with additional tools and libraries.
JKS – Java Key Store
JRE – Java Runtime Environment
KML – Keyhole Markup Language, the XML-based data format used by Google Earth
OS – Operating System
OWASP – Open Web Application Security Project
NIAP – National Information Assurance Partnership
PKCS12 – Public-Key Cryptography Standard #12
TCP – Transmission Control Protocol
RHEL – Red Hat Enterprise Linux
UDP – User Datagram Protocol
SSL – Secure Sockets Layer
TAK – Team Awareness Kit, a mobile or desktop application that sends and receives real-time information through TAK Server.
TLS – Transport Layer Security, a newer and more-secure protocol derived from SSL. The terms SSL and TLS are often used interchangeably. Technically, TLS provides a superset of SSL's capabilities and should always be preferred.
XML – Extensible Markup Language

Appendix B: Certificate Generation

TAK Server includes scripts for generating a private security enclave, which will create a Certificate Authority (CA) as well as server and client certificates.

First, figure out how many client certificates you are going to need. Ideally you should have a different client cert for each ATAK device on your network.

Become the *tak* user: sudo su tak

Edit the certificate-generation configuration file, at this location:

```
/opt/tak/certs/cert-metadata.sh
```

Set options for country, state, city, organization, and organizational_unit. Delete this line:

```
### delete this line once you have edited the above fields
```

Create a certificate authority (CA):

```
./makeRootCa.sh
```

Follow the prompt to name the CA.

Create a server certificate:

```
./makeCert.sh server takserver
```

this command will result in a server certificate named /opt/tak/certs/files/takserver.jks

Create one or more client certificates. You should use a different client cert for each ATAK device on your network. This username will be provisioned in the certificate as the CN (Common Name). When using certs on devices that are connected to an input that is configured for group filtering without authentication messages, this username will be used by TAK Server to look up group membership information in an authentication repository, such as Active Directory (AD). This command will create a cert for the username **user**:

```
./makeCert.sh client user
```

Generate another cert, named **admin** to access the admin UI:

```
./makeCert.sh client admin
```

The generated CA truststores and certs will be located here:

```
/opt/tak/certs/files
```

Follow the instruction on "Configure TAK Server Certificate" to set up the server to use the generated certs and to authenticate users on a TLS port.

Restart the TAK Server:

```
sudo systemctl restart takserver
```

Authorize the **admin** cert to perform administrative functions using the UI:

```
sudo java -jar /opt/tak/utils/UserManager.jar certmod -A
/opt/tak/certs/files/admin.pem
```

Become a normal user:

```
exit
```

1 Configure TAK Server Certificate

In `/opt/tak`, check the following settings in `CoreConfig.xml`:

1. In the `<tls>` element, the `keystoreFile` attribute should be set to the server keystore that was generated with `makeCerts.sh`, above. If you followed the instructions verbatim, the server keystore is `/opt/tak/certs/files/takserver.jks`.
2. Also in the `<tls>` element, the `trustoreFile` attribute should be set to the the trust store that was generated with `makeCerts.sh`, above. If you used the default arguments, your trust store file is `/opt/tak/certs/files/truststore-root.jks`.
3. In the `<network>` element, add a TLS input, specifying group-based filtering without requiring an authentication message:

   ```
   <input _name="tlsx509" protocol="tls" port="8089" auth="x509"/>
   ```

   You can change the port number if you want.

Example:

```
<network multicastTTL="5">
      <!-- <input _name="stdtcp" protocol="tcp" port="8087"/> -->
      <!-- <input _name="stdudp" protocol="udp" port="8087"/> -->
      <input _name="stdssl" protocol="tls" port="8089" auth="x509"/>
      <!-- <input _name="streamtcp" protocol="stcp" port="8088"/> -->
      <!-- <input _name="SAproxy" protocol="mcast" group="239.2.3.1" port="6969"
proxy="true"/> -->
      <!-- <input _name="GeoChatproxy" protocol="mcast" group="224.10.10.1"
port="17012" proxy="true"/> -->
      <!--<announce enable="true" uid="Marti1" group="239.2.3.1" port="6969"
interval="1" ip="192.168.1.137" />-->
</network>
...
<security>
   <tls context="TLSv1"
         keymanager="SunX509"
         keystore="JKS"
      keystoreFile="certs/files/takserver.jks" keystorePass="atakatak"
      truststore="JKS"
```

```
      truststoreFile="certs/files/truststore-root.jks" truststorePass="atakatak">
                  (Uncomment the following if you are using a CRL)
                <!-- <crl _name="Marti CA" crlFile="certs/ca.crl"/>  -->
      </tls>
</security>
```

Then (re)start the TAK Server as normal.

2 Installing Client Certificates

Take the `truststore-root.p12` and `user.p12` files and copy them to your Android device. In ATAK, open `Settings->General Settings->Network Settings`

and set the SSL/TLS Truststore and Client Certificate preferences to point to those .p12 files.

Repeat the proceedure described above for creating a new server connection, but be sure to select **SSL** as the protocol.

These same .p12 files can be installed in a browser, and used to access the Web UI (for admin use) and WebTAK (for normal users or admins). The process to install these files varies by browser and operating system, but can generally be configured by going to the browser preferences, and the security or certificates section.

Appendix C: Certificate Signing

TAK Clients can enroll for new client certificates by submitting a Certificate Signing Request (CSR) to TAK Server. The Certificate Signing endpoint resides on port 8446 and requires HTTP Basic Authentication backed by either File or LDAP authentication. Ensure that the tomcat connector for port 8446 is active within tomcat-home/conf/server.xml.

The CertificateSigning section in CoreConfig.xml specifies how CSRs are processed. TAK Server can be configured to sign certificates directly, or proxy CSRs to a Microsoft CA instance running Certificate Enrollment Services. To configure TAK Server to sign certificates, set the CA attribute to "TAKServer". To configure TAK Server to proxy the CSR to MS CA, set the CA attribute to "MicrosoftCA".

```
<certificateSigning CA="{TAKServer | MicrosoftCA}">
      <certificateConfig>
            <nameEntries>
                  <nameEntry name="O" value="Test Org Name"/>
                  <nameEntry name="OU" value="Test Org Unit Name"/>
            </nameEntries>
      </certificateConfig>
      <TAKServerCAConfig
            keystore="JKS"
            keystoreFile="../certs/files_intCA/intermediate-ca-signing.jks"
            keystorePass="atakatak"
            validityDays="30"
            signatureAlg="SHA256WithRSA" />
      <MicrosoftCAConfig
            username="{MS CA Username}"
            password="{MS CA Password}"
            truststore="/opt/tak/certs/files_MSCA/keystore.jks"
            truststorePass="atakatak"
            svcUrl="https://win-kbtud3n1hjl.tak.net/tak-WIN-KBTUD3N1HJL-
CA_CES_UsernamePassword/service.svc"
            templateName="Copy of User"/>
</certificateSigning>
```

Prior to submitting a CSR, Clients query TAK Server for Relative Distinguished Names (RDNs) that need to go into the CSR. The nameEntries element in CoreConfig.xml specifies the required RDNs, giving the administrator control over generated certificates. The CN value in the CSR will be equal to the HTTP username. TAK Server validates all required fields in the CSR prior to signing.

The extra step of having client query TAK Server for RDNs wouldn't be required if TAK Server were signing certificates exclusively, since TAK Server could just add these names to the certificate.

However, when proxying the CSR to an external CA, this allows additional flexibility in controlling the subject name within the certificate.

The TAKServerCAConfig element specifies the keystore that TAK Server will use for signing certificates. The keystore must hold the CA's private key along with it's full trust chain. The makeCert.sh script will produce a signing keystore when generating an intermediate CA certificate. Certificates signed by TAK Server will be valid for the specified validityDays, and will be signed using the algorithm specified by signatureAlg.

The MicrosoftCAConfig element defines how TAK Server will connect to the Certificate Enrollment Services (CES) endpoint. The CES endpoint is defined by the svcUrl attribute. The CES endpoint must be configured to use Username/Password authentication, and by default will include 'UsernamePassword' in the URL. The username and password attributes refer to an account configured on the MS CA Server to access the the CES endpoint. The truststore and truststorePass attrbitues point to a Java keystore (.jks) file that contains the trust chain for the svcUrl endpoint. Lastly, the templateName defines the certificate template that will be used to sign CSRs sent from TAK Server.

Appendix D: OpenJDK 11 on CentOS 6 systems

TAK Server now requires OpenJDK 11 to run but on CentOS 6 systems. **CentOS 6 is not officially supported – the recommended version of CentOS is version 7**. OpenJDK 11 is unavailible through the normal package installation (in this case CentOS uses yum). This section will offer a workaround to use for OpenJDK 11.

Remove any existing old Java installations

- From a terminal run rpm –qa | grep java
    - This will list all current Java packages the system has installed (on fresh CentOS 6.9 install there were only jdk 1.7.0 and jdk 1.6.0)

Install OpenJDK 11 and setting it up to be used

- Download the latest release of OpenJDK 11 from https://jdk.java.net/11/
    - This will be in a .tar.gz file
- Extract the downloaded file by running: *tar zxvf openjdk-11.0.2_linux-x64_bin.tar.gz*
    - At the time of writing this the latest OpenJDK 11 was version 11.0.2. Your downloaded version may be different which will change the names of these files that are referenced
- Move the folder that was produced from the above tar command to /usr/local/ by running: *mv jdk-11.0.2 /usr/local/*
    - This will require elevated permissions so either use *sudo* or change to the root user before running this command
- Create a script in /etc/profile.d/ that will allow the system to know where to look for the new openJDK 11
    - With preferred text editor create a file called *jdk11.sh*
    - In this file put the following 2 lines:
        - *export JAVA_HOME=/usr/local/jdk-11.0.2*
        - *export PATH=$PATH:$JAVA_HOME/bin*
    - Move this file to /etc/profile.d/ - this will require elevated permissions
    - In any currently open terminals run: *source /etc/profile.d/jdk11.sh* in order to get the java executables on the path (any newly opened terminals will not need this to be run)
- In order to test that OpenJDK 11 is now the version being used run java –version in a terminal. The output should be something like: *openjdk version "11.0.2" 2019-01-15*. The important thing to look for is that the first number is 11.

Since this is a workaround, CentOS will not recognize that Java 11 is installed. This will affect rpm installs of TAK Server because the package manager will not see the Java-11 dependency satisfied when running the rpm command to install a new TAK Server distribution. In order to get around this be sure that all dependencies are satisfied (these include postgresql10, postgresql10-server, postgresql10-contrib, postgis24_10, and postgis24_10-utils) and use the option –nodeps when using rpm to install TAK Server. This will disable dependency checking so if any dependencies are missing TAK Server will not work correctly after installation.

Appendix E: Previous Change Logs

4.5

- Platform
  - Add support for Red Hat Enterprise Linux 8 and CentOS 8
  - Significant performance improvements
  - Fix database container restart connectivity issue in docker. See section 4.6 Docker Install
- Admin UX
  - Add new User Management UI for local file-based users.
  - Added new takserver-esapi.log for intrusion detection messages. These messages were previously in takserver-api.log and are now written to a dedicated log file.
  - Added persistence for CoT injectors.
  - In UserManager.jar, add options -ig and -og to UserManager to dynamically specify In groups and Out groups. Fix username / password user deletion logic in UserManager. Update groups dynamically when changed from UserManager.
  - In UserManager.jar when changing groups for a user, instead of always adding what's specified (group name), require the full list of group to be specified, and add / remove internally to match. This adds support for dynamically removing groups from a user.
  - On upgrade, preserve existing `clear-old-data.sql`script
- Security
  - Updated default CRL validity period to 2 years, matching default validity of server certificates.
  - Update tomcat to 9.0.54 to address CVE-2021-42340 Denial of Service see https://us-cert.cisa.gov/ncas/current-activity/2021/10/15/
  - Dependencies updates, including Apache Ignite
  - Support SELinux enforcing mode
- Situational Awareness
  - Added optional user facing web page for submitting location reports, configurable to broadcast or add locations to a mission
- Authentication
  - Add controls for limiting access to a set of ldap groups
  - Add option to set group name based on DN in CAC/PIV
  - Allow for concurrent WebTAK logins with same username
  - Support revocation of client certifications through enrollment, without requiring a server restart.
- Data Management
  - Add feature to support data sync mission archive and retrieval.
- TAK Server Plugin SDK
  - Add new MessageInterceptor plugin type. Plugins of this type intercept messages after they are received by TAK Server from clients or federates, but before the messages are broadcast out to receiving clients. Plugin code can modify or enrich each message that is intercepted.
  - SDK available here: https://git.tak.gov/sdks/takserver/tak-server-sdk/-/tree/develop/4.5
- Federation
  - Add feature to block files by file-type (such as ATAK .pref files)
  - Block duplicate connections
- WebTAK

- Features
  - added routes tool (receive-only)
  - delete feed in Data Sync
  - re-write of Data Sync tool to improve operations
  - better recognition and notification of lost server connection
  - ability to reconnect with server without a page refresh
- Bugfixes
  - disabled password autofill on text inputs
- Security Patches
  - n/a

4.4

- Messaging and Quality of Service (QoS)
  - Add dynamic reporting rate limiting on client message reads
  - Add dynamic read-rate limiting (Denial of Service mitigation)
  - Docs here: https://wiki.tak.gov/display/TPC/Quality+of+Service
  - Signficant reductions in CPU utilization
  - Improve performance on lower-spec'd servers
  - Optimizations to support running in exploded WAR format on I/O limited devices
  - Update DropTypeFilter to work without time threshold, supports blocking of messages by type regardless of frequency
- Channels
  - Added Channels capability that allows end users to select their active groups
  - Docs here: https://wiki.tak.gov/display/TPC/Channels
- GeoChat
  - Added Store / Forward for GeoChat to let users retrieve messages that were missed while offline
- Security
  - Add support for moving admin UI to an alternate port (can be placed behind a firewall.)
  - For HTTPS ports, add a feature to optionally disable WebTAK, Admin Web UI and non-admin Web UI
  - Dependency updates
  - Add a password complexity requirement for username / password users created using UserManager.jar
- Device Profiles
  - Updated Device Profile Manager to set per-user attributes at enrollment time (callsign, color, role)
- Mission API (Data Sync)
  - Significant performance optimiziations
  - Internal data caching
  - Improve cache performance
- Federation
  - Show remote IP address and port for federation connections
  - Show groups for active federation connections
  - Add link to configure federation in active connections display
  - Improve error messaging when remote federate is not available or untrusted

    - Updates to federation hub supporting hub-spoke federation as part of TAK Server. Docs here: https://wiki.tak.gov/display/TPC/Federation+Hub
- Authentication
    - Increased maximum number of groups from 4,096 to 32,768
- WebTAK 4.4
    - Features
        - added initial support for Cesium 3D integration and views
        - rewrote the KML/KMZ parser
        - updated the toolbar to simplify its interface
        - improved support for CoT b-type messagesrewrote the KML/KMZ parser
    - Bug Fixes
        - updated logic for logout button so that it is always displayed when logged via username/password
        - resolved file import delete button not updating the UI
        - resolved widgets not displaying in certain situations
        - resolved issue with team names with a space causing app crashesperformance improvements

4.3

- Messaging Quality of Service (QoS)
    - Dynamic reporting rate limiting. As number of clients grows, reporting rates are limited automatically in the server to reduce bandwidth load and conserve resources.
    - UI to dynamically enable / disable this feature and monitor current state.
- Mission API (Data Sync)
    - Performance updates
    - Enhanced data caching
- Database
    - Database connection pool auto-sizing
    - UI updates
- TAK Server Plugin SDK
    - Add new SenderReceiver plugin type
    - SDK available here: https://git.tak.gov/sdks/takserver/tak-server-sdk
- Federation
    - Federation (v2) connection stability improvements
    - bug fixes
    - New federation hub supports hub-spoke federation as part of TAK Server. Information here: https://wiki.tak.gov/display/TPC/Federation+Hub
- Authentication
    - Added support for setting Read / Write access for Active Directory users.
- WebTAK 4.3.0
    - performance improvements
    - file storage and display
    - CoT handling
    - added map grid for widget placement
    - added widget drawer
    - made AOI widget responsive

- added go-to tool
- added bottom drawer support
- updated footer and moved TAK Server status
- added ability to import a custom icon palette
- removed models and services deprecated in 4.2.1

4.2

- Mission API (Data Sync)
  - Significant performance enhancements
  - Per-mission fine-grained caching
- Data Retention Tool
  - Add a data retention tool, including admin UI
  - Specify retention periods by data type (messages, GeoChat, missions, mission packages)
  - Documentation: https://wiki.tak.gov/display/TPC/Data+Retention+Tool
- TAK Server Plugin SDK
  - Support starting and stopping plugins interactively through admin UI
  - Add plugin support for TAK Server cluster (kubernetes)
  - SDK available here: https://git.tak.gov/sdks/takserver/tak-server-sdk
- Federation
  - Add end-to-end group mapping support for Mission API and mission packages
- Admin UI
  - Support paging in client dashboard
- Messaging
  - Support designation of a specific network interface for multicast
- WebTAK
  - Added support for GeoTIFFs and GRGs
  - Added generic menu component
  - Various core updates and bug fixes

4.1

- TAK Server Plugin SDK
  - New native plugin architecture
  - Supports sending and receiving messages, including group support and direct messages to individual users
  - No additional configuration required to run plugins
  - Includes Admin UI
  - SDK available at https://git.tak.gov/sdks/takserver/tak-server-sdk
- TAK Server Cluster (Kubernetes)
  - Transforms TAK Server into a horizontally scalable distributed application
  - Uses Kubernetes for cloud-native container orchestration
  - Currently available as a Beta
- Federation
  - Data Sync Federation Disruption Tolerance
    - Federate mission (feed) changes that occur during periods of disconnectivity
    - Can be configured in UI
  - Automatic end-to-end group mapping

        - Fine-grained data grouping within an organizational domain
    - TLS 1.3 support
- Token administration UI
    - Allows users to view and revoke OAuth2 tokens
- Certificate enrollment UI
    - Allows users to configure certificate enrollment without touching CoreConfig.xml
- User registration system
    - Allows users to register for TAK server account using approved email domain or via invitation email
    - Added support for periodic re-authorization of X509 clients
- Mission API Improvements
    - Multiple performance improvements to optimize cache and database access
    - Return list of group names with mission, allow mission editor to update mission groups
    - Update hashtags endpoint to not require separate delete call, eliminating extra notification
    - Limit deepDelete option to Mission Owner role
    - TLS 1.3 support (for all HTTP APIs)
- WebTAK
    - added Data Sync tool
        - added support for creating Data Sync missions for different TAK Server groups
        - added link to Data Sync in Overlays
    - added Video Player tool
    - added Emergency Alerts tool
    - added receipt-only Data Packages tool
    - added notification system
        - added notification counters for chat messages and Data Sync updates
    - updated default toolbar list
    - overhauled preferences architecture and UI
    - added instructions component
    - added built-in instructions for shape creation and moving user's own marker
    - added userAgent logic to better check for device type
    - fixed circle shape bugs
    - added polygon step instructions and undo/end buttons
    - added info box for map items
    - added tooltip component
    - added user disconnect message support
    - fixed All Chat Rooms bug
    - fixed Overlays search bug
    - fixed AOI creations bugs
    - added graceful fail on contacts request error
    - various architecture updates

4.0

- Federation: End-to-End Group Mapping
- Authentication: OAuth2 Support
- Performance: Internal usage of Protocol Buffers
- Initial Setup: LDAP settings validator

- Group-filtering replacement of proxy flag