HATDeX

Volume 1 | Issue 2
17 June 2016

Under the Bonnet

Under the Bonnet is a briefing document series issued by HATDeX to members of the HAT network

Welcome to the second issue of *Under the Bonnet*, the briefing document series issued by HATDeX (HAT Data Exchange) to members of the HAT network. This document series will provide briefings on different aspects of the HAT; its tech, business, markets as well as product updates.

In this issue:

The Security of Your HAT from HATDeX*: We explain the multiple layers of protection employed to protect personal data stored in the HAT, whether at rest, in transit or in use.*

The Security of Your HAT from HATDeX

The HATDeX-serviced HAT is designed with multiple layers of protection, covering Data at Rest, Data in Transit and Data in Use. Methods employed to protect HAT data include encryption, network configuration and application-level controls that are distributed across a scalable, secure infrastructure. HAT users can access their personal data at any time from any certified application.

HATDeX services are developed with multiple layers of redundancy to guard against data loss and to ensure availability. All certified applications connect to the HAT API to provide access to data (and management if you are using the Rumpel hyperdata browser), and to allow data sharing with others. The service can be utilised and accessed via a number of interfaces. Each has its own security settings and features that process and protect the data whilst ensuring ease of access.

Personal data in the HAT ecosystem is secured at all stages:

Data at Rest – The HAT database is isolated at the server level (containerised), encrypted and backed up regularly.

Data in Transit - Communication between any client and any HAT must go through the designed HAT APIs. We strengthen the API-driven communication by requiring SSL-encrypted communication, token-enabled authentication with the HAT and service provider verification to enter the ecosystem from MarketSquare.

Data in Use - Applications must disclose their intention for data use, and are subject to user-managed control of access. In addition, all HAT users collectively monitor service providers (application and hosting), and can report any breach of compliance with the HAT Code of Practice.

HAT Data at Rest

Each HAT stores its data in a HAT Database instance - postgreSQL, using the HAT schema [https://github.com/Hub-of-all-Things/HAT2.0]. The database configuration optimises security and reliability according to the following measures:

- The SuperUser account on PostgreSQL is inaccessible, and is only reachable from inside a secured Virtual Machine (VM), never used or accessed for administrative or other purposes by provisioning systems or administrators.
- Each HAT has its own, separate database, only readable by the single designated HAT account, and differentiated from other HATs running in the same environment.
- Each HAT database runs on one of the large numbers of HAT Database Servers, which also operate as isolated containers across a number of Elastic Compute Cloud EC2 instances for reliability.

Each HAT database instance sits in a Docker container, adding an extra layer of security. This helps in:

- Localising the impact of any security issues.
- Eliminating the possibility of system administrators enabling unauthorised access to data of large numbers of users (typical attacks towards account-based systems).
- Containing any security breaches. Since the HAT ecosystem is not an account-based system, security breaches to one HAT will impact that HAT alone. This therefore de-incentivises security breaches targeting a large number of user accounts.

The database is also encrypted and backed-up:

- HAT data is persisted in encrypted Elastic Block Store (EBS) volumes.
- Database backup uses EBS Snapshots, to make sure individuals' data is not lost even in the case of outages.
- HATDeX-serviced HATs can also offer optional EBS Encryption [http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html].

Infrastructure – Leveraging one of the most secure cloud infrastructures in the industry, HATDeX HATs benefit from Amazon Web Services (AWS)'s built-in security solutions including AWS Infrastructure Security (TLS), Data

Encryption, Inspection, Assessment, Monitoring, Logging, ID and Access Control, Thread Prevention and Penetration Testing.

HAT Data in Transit

All HAT data IO (input/output) must go through designated APIs, and this includes the activities of HAT owner managing their personal data through Rumpel. Such API-only accessibility removes potential security breach from other channels.

Utilising a single sign-on (see below), the HAT user is in full control of any and all data output from a HAT. These are Direct Data Debit transactions, whose activation requires the HAT user's review and approval. In addition, the meta-data of every approved Direct Data Debit transaction is archived to provide ecosystem-level security to prevent, detect and resolve any potential security issue.

SSL-encrypted communication is required between any client (apps, MarketSquare or Rumpel) and any HAT:

- Secure Sockets Layer (SSL)/Transport Layer Security (TLS) for data transit is implemented through the creation of a secure tunnel protected by Advanced Encryption Standard (AES) encryption.
- SSL for HAT Main Domains (second level such as hubofallthings.net). For example, a user's HAT address may be username.hubofallthings.net; a Milliner Service's may be milliner.hubofallthings.net; and the address for an Application Provider "SampleApp" may be sampleapp.hubofallthings.net.

HAT Community Foundation (HCF) Certification is required for any organisation prior to obtaining an SSL to operate in the HAT ecosystem.

Network security – HATDeX employs industry-standard protection techniques, including firewalls, network security monitoring and intrusion detection systems, to ensure that only eligible traffic is able to reach the infrastructure.

HAT Data in Use

Managerial Usage – Is available only to the HAT owner with the owners' SSL certificate. Only the owner decides what data flows into that HAT and how to organise it, what to share, at what price (or for what other types of incentives such as vouchers), with whom, and for how long, within the HAT ecosystem.

App Usage – Available to App Developers & HAT Application Providers (HAPs)

- App Developers must sign up through MarketSquare, and agree to the HAT Code of Practice.

- HAPs - must disclose their intentions on data usage, and request access to the data bundles.

Ecosystem Exchange Archiving – Archiving meta-data of data exchange within the HAT ecosystem is designed to provide additional layers of protection and defence. It offers additional evidence should disguised data usage be reported (by the users) or detected, and helps to identify the trading party that does not have the required honest data usage disclosure.

HAT Authentication

The following added extra layers of security practice for HAT authentication help to enhance the level of security in the HAT ecosystem.

Cryptographically-signed JWTs (JSON Web Token) are used for authentication with the HAT in the following steps:

- When a HAT user wants to sign in on an application (such as MarketSquare) or Rumpel, they will be redirected to their own secure HAT with SSL-encrypted connection.
- They enter their login credentials (username/password) on the login form.
- The HAT generates a signed JWT and the user is redirected back to the originating service.

Encryption key management infrastructure is designed with operational, technical and procedural security controls with very limited direct access to keys. Encryption key generation, exchange and storage are distributed for decentralised processing.

MarketSquare acts as an additional trust anchor for third-party applications. By default, no third-party application can add data to or request data from a HAT through a data debit, without an approved account on an individual HAT. And in order to create an account (and obtain approval) on a HAT, a third party application must go through MarketSquare. (This is not to create and approve data debits or other data transactions).

MarketSquare enforces the following process to provide such a trust anchor:

- An application developer must retrieve a JWT crypto-signed token from MarketSquare when they register an application (e.g. data plug) and include that in the application’s configuration.
- Once the app is registered, it provides MarketSquare with its hash (one-way encrypted password, using BCrypt algorithm), which will then be given by MarketSquare to the HAT when the app creates an account.
- The app can ask for access to a specific HAT by requesting MarketSquare (via an API call) to create an account on the HAT, providing the token to prove its identity.
- Upon successful account creation, the app can directly log into the HAT to retrieve the HAT’s token (as in the first item above), using the password known only to the app, which corresponds with the encrypted value provided to the HAT via MarketSquare.

In the July 2016 release of the HAT, all external user accounts (such as marketers and application providers) will be verified and approved or rejected by the administrators of MarketSquare, as will all data offers and applications.

Security for HAT Provisioning

HATDeX offers its Milliner Service and its APIs to other organisations to enable them to host HATs (read more about this in Issue 1 of Under the Bonnet). The following additional layers of security solutions have been implemented to ensure secured HAT provisioning:

A Milliner needs to control HAT configuration variables, such as internal user details and passwords, to successfully start, manage and recover HATs. Therefore, when provisioning a HAT database, a Milliner uses a temporary administrator account, which the HAT immediately deletes upon its successful launch, cutting off the Milliner from potential data access.

The Milliner itself treats HATs as generic Docker containers (or Kubernetes Pods), without internal knowledge or exposure to how they operate. When provisioning HATs, the Milliner sets up:

- HAT database on one of the existing servers or a new server when required, removing its own access to the database after its successful launch. EBS storage volumes are set up when creating a new Database Server, at which point they can be configured to be encrypted.
- HAT software stack in a separate container, with the right configuration to access the database as well as HAT-specific security credentials (such as public keys, SSL certificates, etc).
- HAT software stack container that is provisioned on one of the existing VMs, without "pinning" to a specific VM.
- DNS name and load-balancer configuration to make a HAT accessible via a domain name and SSL-secured connections.

Comments? Views?

Please contact inquiry@hatdex.org or go to http://forum.hatdex.org/ to discuss further!