Notes on privacy and data collection of Matrix.org

DISCLAIMER: This research and investigation work is based on several years of experience within the Matrix ecosystem and validation of facts via public and private communication.
Reverse engineering was used to ensure some of the statements presented as facts regarding implementations are accurate.
Nonetheless it is possible that a mistake has made its way in these notes. If that is the case, please get in touch with the author which will fix any factual mistakes in good faith. **We always encourage people not to trust statements at face value and always double-check for themselves.**

TL;DR: matrix.org and vector.im receive a lot of private, personal and identifiable data on a regular basis, or metadata that can be used to precisely identify and/or track users/server, their social graph, usage pattern and potential location. This is possible both by the default configuration values in synapse/Riot that do not promote privacy, and by specific choices made by their developers to not disclose, inform users or resolve in a timely manner several known behaviours of the software.
Data sent on a potential regular basis based on a common web/desktop+smartphone usage **even with a self-hosted client and Homeserver**:

- The Matrix ID of users, usually including their username.
- Email addresses, phone numbers of the user and their contacts.
- Associations of Email, phone numbers with Matrix IDs.
- Usage patterns of the user.
- IP of the user, which can give more or less precise geographical location information.
- The user's devices and system information.
- The other servers that users talks to.
- Room IDs, potentially identifying the Direct chat ones and the other user/server.

With default settings, they allow unrestricted, non-obfuscated public access to the following potentially personal data/info:

- Every file, image, video, audio that is uploaded to the Homeserver.
- Profile name and avatar of users.

See below for a detailed analysis.
If you have questions, want clarification, have spotted factual errors, or just want to discuss about privacy in Matrix and alternatives, come to our general Matrix room: #kamax-matrix:kamax.io

---

Commented [1]: All projects mentioned below are open source (other than the integration manager, which instead is being openly specified now) and so talk of reverse engineering is irrelevant & alarmist.

Commented [2]: matrix IDs by definition include a username...

Commented [3]: Only if optionally provided by the user.

Commented [4]: Only if the user explicitly opts into sharing their contacts from Riot/Mobile.

Commented [5]: The definite article is disingenuous here - we strip any identifying data from matomo analytics, _and_ they're opt in.

Commented [6]: no, the URLs in the content repository are obfuscated, e.g. mxc://matrix.org/oUxxDyzQOHdVDMxgwFzyCWEe

Commented [7]: Obfuscated and encrypted where encryption is enabled

Foreword

On the 12th of June 2019, after 5 years of hard work, Matrix.org released the v1.0 of the protocol, alongside v1.0.0 of the reference Homeserver implementation, synapse, and Riot v1.0.0 earlier in the year.
Having studied Matrix for more than 2 years, creating various implementations with mxisd as our most notable software, we decided to make a review of the protocol on three values that we believe are fundamental to any open protocol:

- **Privacy:** Data/metadata should strictly only be accessible to user's intended recipient(s).
- **Decentralisation:** Network operations can not depend on a "central" set of server(s).
- **Security:** Access to data/metadata is authenticated and authorised by default, with passive or active safeguards (like End-to-End encryption) being on by default.

After the Matrix.org security breach, when an unauthorised person gained access to personal and private data, we believe this review to be critical and necessary on topics not often discussed.

Purpose and Scope
In this document, we will attempt to answer the following questions:

- Following the Matrix.org client/server recommendations/guides, can you be sure that your privacy is respected and your data secure?
- Using the default/recommended settings in the recommended clients/servers, where is your data/metadata flowing?
- If you were to create several direct message channels with others, will a 3rd party be aware of it and if yes, which?
- Are the default/recommended client/server explicit about where the data is flowing, which 3rd party is it shared with?
- Given the recent security breach of Matrix.org, what kind of information was accessed?

To do so, we will follow the most common setup and the ones recommended by Matrix.org itself for self-hosting:

- Self-hosting/installing your own client.
- Self-hosting your Homeserver.
- **NOT** self-hosting your own Identity server.

Matrix.org only gives self-hosting recommendations for client, Home and Identity server, a full Matrix stack also include several other items. We will also cover those unspoken items in this review.
We believe that privacy and security is only as good as default settings, software and recommendations given to users. This review will therefore be based on the spirit that *Default Settings Matter*, a view shared by core Web actors like Mozilla.

This review is based on the principle that no consent is given to any 3rd party service/Privacy policy/Terms of Services of any kind unless specifically prompted, following the expectations of users that their data and metadata is self-hosted and under their control. This is in line with EU GDPR laws.
Many people looking into Matrix are in dire need of privacy and security: activists, journalists, minorities, etc. It is crucial that they are informed about possible privacy leaks that could later be used to identify them. Such identification usually leads to abuse, harassment, assault, threats, blackmail. It is crucial that users are not mislead. It is equally crucial that they are able to evaluate the real value of the ecosystem when used in a daily, real-world setting.

Setup
The following stack will be used as reference, with users connecting via web, desktop and smartphone clients:

- Client: Riot-web v1.2.1, Riot Desktop v1.2.1, Riot Android v0.9.1
- Server: Synapse v1.0.0

Riot has been chosen following the big promotion under "Try now" on Matrix.org, appearing on the dedicated landing page with the hereafter quote and being the first and second recommendation in Clients.
The easiest way to try Matrix is to use the Riot Web client in your browser
Synapse has been chosen because it is the first recommendation on the Matrix.org website, and is the only server feature-complete enough to be used on a day-to-day basis.
This choice of client and server matches our knowledge and experience that it is representative of the overwhelming majority of the Matrix ecosystem.

Riot, the reference client implementation

Overview

Riot is a software written by New Vector Ltd, a UK for-profit created in 2017 to support the people who created Matrix.org after Amdocs, the original founder of Matrix.org, cut them loose. While synapse and nearly all implementations are made under Matrix.org ownership are called "reference implementations". Riot is a "reference implementation" put together by another entity, using SDKs from Matrix.org. Since Matrix 1.0, The Matrix Foundation is officially the owner of anything under Matrix.org, making matrix.org and vector.im legally distinct.
Riot-web and Riot desktop share the same code base and both ship with a default config file that contains several URLs/domains that we will explore in the various sections of this review.
This part of the review explores the default settings and behaviours specific to Riot as a Matrix client. Throughout the whole document, we will assume only the Homeserver URL was changed.

Personal Identifiers

NOTE: This section might contain specifics to Riot Web and Desktop, and overlaps with the Identity Server section.
The first two elements are the Homeserver and the Identity server: matrix.org and vector.im, respectively. If we are to examine the "Register" screen of Riot, we see that only the Homeserver URL is mentioned and selected, while vector.im is not displayed. Only if you click on "Change", you are prompted for both URLs.
On the screen allowing to set custom URLs, one can click on "What does this mean?". The identity server URL explanation reads:
You can also set a custom identity server, but you won't be able to invite users by email address, or be invited by email address yourself.
So far we see that Identity servers are explained to not be important for the day-to-day usage of Matrix in the FAQ, or even reducing usability in Riot. **This is only true if sydent is used, the reference identity server.** Riot devs are well aware of the only other Identity server mxisd which federates and can include data from vector.im. **This is highly misleading and pushes users not to even try to self-host Identity servers, or use another than the default, out of fear.** We will see later the importance of them in terms of privacy.
In a self-hosted scenario, following the recommendation of Matrix.org, only the Homeserver URL would be changed while keeping the Identity Server URL to vector.im.
When registering without an email, we are prompted with a **Warning!** sign and the following text:
If you don't specify an email address, you won't be able to reset your password. Are you sure?
At this point, if we are to cancel and enter an email address in fear to be locked out of our account, we are prompted to validate it using a token/link sent by email. Riot does not give any kind of explanation that the Identity server has been contacted to validate the email, in this case vector.im. The Identity server will therefore have the following information upon successful verification of the email:

- The given email address.
- When submitting the token via HTTP request directly to the Identity server:
    - The IP of the user.
    - Browser/app information via HTTP header User-Agent.
    - Any other information sent by browsers by default.
- The Matrix ID of the user, usually including their username, is also made public without any authentication under the lookup endpoint on https://vector.im.

Example: If you were to register with the email dummy@example.org, you can go to the following URL and see the JSON response including your Matrix ID:
https://vector.im/_matrix/identity/api/v1/lookup?medium=email&address=dummy@example.org

Change the address query parameter to your email to see the mapping which never expires.
At this point, 3 personal identifiable pieces of information are shared with vector.im, a 3rd-party for-profit company directly from a matrix.org recommendation without any prompt, explicit information or given informed consent from the user as per GDPR requirements. **Two of them can be queried unrestricted and without any credential.**

Commented [8]: No, all of the current non-centralised identity servers (e.g. mxisd) either restrict you from being contactable (by not publishing your email address to the wider identity db) or from being able to contact people (because their emails are not published on the wider identity db). The fact that mxisd can delegate lookups to the vector.im server doesn't change this, and this is why the warning exists. This is also why we've been holding out for genuinely decentralised rather than federated identity architecture to replace sydent.

Commented [9]: There are at least two other identity serve implementations out there, fwiw - https://github.com/sroycode/identt and https://github.com/Peyk/D1agonal, as well as proprietary implementations.

Commented [10]: The fact that Riot doesn't advise the user in this scenario is indeed an oversight which we need to solve.

Commented [11]: Yup, this is the point of the service - to map email addresses and phone numbers to matrix IDs.

Commented [12]: again, the point of the service is to map email addresses to matrix IDs, so people can discover people to talk to *if they already know their email address*.

Vector.im has a privacy policy which only applies to jobs and related applications, and does not seem to cover the Identity Server usage specifically, while only giving one lawful basis for processing, directly related to recruitment. It is therefore not known how the data submitted to vector.im is processed or shared.
From our experience in the Matrix.org community and various discussion with the Matrix.org people, we came to realise that the Identity Server under vector.im is part of a cluster that at least integrates an Identity Server under matrix.org and replicates all data from one onto the other. Matrix.org also has a privacy policy making New Vector Ltd the Data controller of the service.
NOTE: You may check it for yourself, replacing vector.im in the lookup URL above with matrix.org
Riot therefore uses, by default, a for-profit service that has no related privacy policy, sharing Personal Identifiable data with a 3rd-party without informing its users, while relying on the trust of a Matrix.org recommendation.

Commented [13]: The privacy policy for the services run by New Vector (e.g. the matrix.org homeserver, the matrix.org identity server, and the vector.im identity server) are at https://github.com/vector-im/policies/blob/master/docs/matrix-org/privacy_notice.md, which all users on the matrix.org homeserver have to click through. It's true that the same click-through should be imposed for users of the matrix.org & vector.im homeserver too, but this is an oversight from the rush of implementing GDPR last year.

Welcome Bot

One of the other default settings is a Matrix ID for the Welcome Bot feature. This feature automatically creates a direct chat with an automated program controlling a Matrix user, allowing a user without prior Matrix experience to ask questions and get useful links. The Matrix ID of the user is @riot-bot:matrix.org. Upon inviting the user, a request will be made by the user Homeserver to the matrix.org Homeserver, allowing the collection of the following information:

- The Matrix ID of the user, built from their username, and which Homeserver/domain they are using.
- The date and time at which the account was created.
- The IP/hostname of the server connected to the user, which might allow to identify a user in case of a single-user Homeserver.
- From the Homeserver IP, their potential GeoIP country/city.

Identity Server

NOTE: Some of the described behaviour is specific to Riot Android and possibly iOS
Identity servers are one of the most misunderstood services in Matrix. Contrary to common belief, Identity servers do not deal with accounts or authentication, but with Identifiers labelled 3PID, a technical term used to describe things like Email and Phone numbers. We'll use *Email* in a generic way in this section.
While Matrix.org does not recommend self-hosting Identity servers, they deal with several key behaviours and personal identifiers:

- Adding/Removing an *Email* to one's profile for discovery by other users.

- Adding an *Email* to allow password reset of an account, being the only self-service way to regain control back after forgetting your password.
- Search for other users to connect with by looking up their *Email*.

Control of the vector.im and/or matrix.org server allows several Denial of Services in terms of blocking 3PID associations and finding other people.
More worryingly, a central server has control over the associations between Email/phone numbers and Matrix IDs and may create them arbitrary, hide or remove them as there are no proof or signature that the 3PID owner allowed such association. This can be used to blacklist/abuse people by abusive administrators relying on an expectation of trustworthiness, but also it allows to target people of interest like activists, people from minorities, etc.

Adding an Email

When attempting to add an *Email* to the Settings, a request is made to the Homeserver to validate and add it. This request is proxied to the Identity server, hiding the IP and any info in the headers from the Identity server.
The Identity server then sends a validation token either in the form of a browser link, or a code to input. In case of email, a link is provided directly pointing to the Identity server instead of the Homeserver. Upon validation, you go back into Riot and click on "Continue" which triggers the final step of actually linking the Matrix ID and the *Email*.
While Matrix sets publishing the association to the Identity server off by default, Riot explicitly requests it. This makes the association public and queryable **without informing the user or prompting for consent.**
The following information is shared with a 3rd-party:

- The IP of the user.
- Its Matrix ID.

The following information is made queryable without restriction to anyone:

- Association of an *Email* to a Matrix ID.

Removing an Email

Removing an *Email* takes on a different approach: while adding an *Email* requires some kind of validation from the owner, removing it does not. It relies on trusting the user's Homeserver to remove the association in a legitimate manner. The user is never prompted to confirm that such removal is wanted or allowed.

Searching for other users

Searching for users is divided into two main use cases:

- A single, specific search available in all Riot versions.
- A bulk search of contacts to find any match, only available in smartphone versions.

Commented [14]: As of Synapse 1.0.0 (and Matrix 1.0), responsibility for password reset is now that of the homeserver. The identity server should not have control over such a sensitive feature.

Commented [15]: This is considered acceptable UX; if a user is trusting the HS to deliver them messages reliably, it is also reasonable to trust the HS to unbind 3PIDs non-maliciously rather than pester the user with confirmation, given the user already confirmed they wanted the HS to bind the 3PID in the first place.

Those searches use unauthenticated Identity server endpoints that Riot directly connects to, allowing the user IP and its device/Riot version to be visible for each request.
While the single specific search behaviour may or may not be understood by users and system administrators, and that potentially identifiable data is shared with vector.im, it is recognised that such requests are only made in response to explicit requests from the user. The various FAQs are unambiguous that Identity servers are used for this purpose.
What is not really known by users, and tends to only be obvious to people implementing the Identity Server spec, is Riot's behaviour regarding bulk search.
Once connected to a Homeserver and on first usage of Riot Android, users will be shown a prompt when clicking on the "People" button, requesting permission to access their contact list. After granting permission, **every email and every phone number in the user's phone book will be sent to the Identity server without any kind of obfuscation or masking**.
The undocumented behaviour is that **any time the user switches out then back in the People view, the full contact list is sent again.**
This bulk behaviour allows the Identity server to:

- Know the IP, client and system of the user.
- Know the potentially complete social graph of the user.
- Receive personal Identifiers (Email and Phone numbers) sent without obfuscation from users unaware of such sharing.
- Receive requests matching pattern usage for the user, specific to certain devices types (smartphones).

Commented [16]: The prompt says: "Riot needs permission to access your address book contacts to find other Matrix users based on their email and phone numbers. Please allow access on the next pop-up to discover address book users reachable from Riot.".

It is obvious that the act of locating people by email address and phone number will involve sharing them.

Commented [17]: It's true that these could be hashed given we are only comparing them (although this would still be susceptible to rainbow table lookups)

Commented [18]: Needless to say, the data is TLS encrypted, even if the payload isn't obfuscated.

Sharing, Permalinks

Recent versions of Riot have a "Sharing" icon, made of three dots linked together in the shape of a triangle. Riot also has a "Share message" option. Both open a new dialog with a URL starting with https://matrix.to/ and a QR code.
Technically, "sharing" (permalinks) is built around a website https://matrix.to/ instead of a URI scheme. While the website is stateless, a cookie is set on each visit by Cloudflare. This cookie uniquely identifies clients for an unknown purpose. If the link is visited instead of intercepted by the client, the following info is shared with a 3rd party:

- IP address of the client/user.
- Usage patterns of the "Sharing feature".
- Unique ID via cookie _cfduid for the sole purpose to identify a client, on a website that is supposed to protect privacy.

Commented [19]: We weren't aware that cloudflare was setting a cookie actually, but the explanation of its use is: https://support.cloudflare.com/hc/en-us/articles/200170156-What-does-the-Cloudflare-cfduid-cookie-do-.

Integration server

NOTE: Some of the described behaviours are specific to the Web and Desktop clients.
Riot comes with a proprietary closed-source service (protocol and implementation) called an Integration Server. The Integration server can be used to add services/bots/bridges to a room, like a Jitsi VoIP conference, enhancing the Riot experience. This service is (was?) meant to be the monetisation feature of Riot, remaining closed-source to this date.

Commented [20]: The protocol is being standardised at https://github.com/matrix-org/matrix-doc/blob/travis/msc/integrations/base/proposals/1956-integrations-api-base.md

Riot comes with the default configuration of using scalar.vector.im as its Integration Server. Integration can be triggered using the 4 small squares icon at the top right of a room, connecting to scalar.vector.im and displaying the current configuration of the Room/services already integrated.
To do so, the following handshake is done:

1. Riot requests an OpenID token from the Homeserver. This token can be exchanged for the Matrix ID of the user at the time of writing.

2. Riot connects to the Integration Server to either register a new session with the OpenID token requested earlier, or to validate an existing session.

3. The Integration Server exchanges the OpenID token via the federation API for the user Matrix ID.

4. Riot then calls the Integration Server with the Room ID to get its Integration status.

No information or explanation is given to the user about their Matrix ID, a potential personal identifier, being shared with a 3rd-party service without a privacy policy. No consent is requested either.
What is less known is that step 2 happens **every time a user switches to another room in the UI.** This means that vector.im is receiving the following information without the user's knowledge, some without the user even opening the Integration server window:

- A steady stream of requests directly related to user activity and usage pattern of Riot and Matrix.
- Their Matrix ID and their IP, Riot directly connecting to scalar.vector.im.
- The rooms which the user is part of.

In terms of Integration usage of the scalar.vector.im, several bridges, bots, widgets and sticker packs are provided via the matrix.org Homeserver. It means that by using nearly any of them, matrix.org will be involved directly or indirectly into the room. In case of bridges and bots, a copy of the room history alongside members' display names and avatars will be known/copied to the matrix.org server, further giving a means to directly access data and conversations.
The tight coupling of matrix.org on those servers is never explicitly explained to the users, nor that past chat history could be downloaded in some cases without them being aware, or that any outage to the matrix.org server would also affect those services. Users are also not told that the service is proprietary and closed-source, only allowing alternative implementation by reverse engineering. This does not allow privacy/security reviews of the software stack, while being the element that has direct access to users' data.

Push Server

Commented [21]: While the endpoint happens to be on the federation API namespace, it has nothing to do with federation (see https://github.com/matrix-org/synapse/issues/2843)

Commented [22]: This is a bug, fwiw - https://github.com/vector-im/riot-web/issues/5846

Commented [23]: No, the request that is made looks like this: https://scalar.vector.im/api/account?scalar_token=...&v=1.1.

It does not list the room ID, unless the user is interacting with an integration which actually needs to know which room the user is in (e.g. a widget).

Commented [24]: It's explained right here for users on matrix.org: https://github.com/vector-im/policies/blob/master/docs/matrix-org/privacy_notice.md#bridging Other homeservers are welcome to fork the same privacy policy for their own users.

NOTE: This section is specific to Riot Android/iOS
Matrix uses a concept of Push server to send push notifications to smartphones. The push server is meant to be managed by the application developer. In case of Riot, the push server is configured to matrix.org.
Riot provides two privacy level for notifications:

- Normal (event metadata only)
- Reduced Privacy (full event data)

While Riot gives the high level differences between the two, it does not mention matrix.org involvement or which metadata is shared and visible. Only the Google services are mentioned. The Push server will have access to the following info in each notification:

- The user Matrix ID.
- The room ID.
- The event ID.

And overall:

- Pattern usage/activity of the user and the users they connect to.
- ID of rooms joined by the user, potentially identifying direct message rooms.

Control of the matrix.org push server allows to perform Denial of Service, blocking notifications that people tend to rely on to further participate in conversations when a reply is sent. See below for a real-word impact during the security breach.

Synapse, the reference server implementation

Basic network calls

When interacting with users from other servers or rooms containing them, synapse uses the S2S protocol, a specific set of endpoints that usually require authentication/authorization.
Up until recent synapse versions, self-signed certificates were accepted as signing keys and certificate fingerprints were checked via a validation approach borrowed from the perspective project. Synapse would check that the keys received are the same if requested by another server, called a Notary server.
Prior to v0.99.2, synapse contained a perspective key in its configuration which was uncommented (enabled) by default pointing to matrix.org. The server would be queried on a regular basis to fetch the keys of every other server synapse was talking to.
From v0.99.3, the configuration was commented out and instead hardcoded into the source code, making it so that **even if the configuration was manually commented out, or removed, synapse would still talk to matrix.org by default and reference all other servers that the Homeserver is in contact with.**
One of the big change for synapse v1.0 and Matrix was the switch from a self-signed, perspective approach to regular CA TLS certificates via MSC 1711. This proposal recognised

Commented [25]: Hardcoding a default config applies to all the configuration options here, so that the defaults are well-defined rather than dependent on the config file, as per https://github.com/matrix-org/synapse/pull/4863.

the centralisation problem and the attack surface of how synapse used matrix.org as a single notary server. The proposal was meant to move away from the perspective model, validate TLS certificates directly and not require notary servers anymore (or so understood), decreasing centralization.
As of synapse v1.0.0, we see that the perspective key has been switched for a new key called trusted_key_servers which is commented out in the default generated configuration. But matrix.org is still hard-coded in the source code.
We have confirmed that synapse v1.0.0 still connects to matrix.org to fetch keys, even if no longer necessary, and does so for every single server your Homeserver talks to. We also confirmed that despite the key lookup endpoint not requiring authentication, synapse does send cryptographically signed requests to matrix.org which ensures the requester can be identified.
matrix.org has therefore the ability to:

- Know which servers exchange data with each other.
- Build a social server graph of many, if not nearly all, federation servers.
- Build usage patterns from the regular re-validation requests.
- Block servers from talking to each other by returning invalid key data.
- Still be a Man-in-the-middle source of attack for anyone who would have access to the matrix.org servers.

Synapse developers on Matrix.org do not give the details and impact information to system administrators about how potentially private information (in case of single-user Homeserver for example) is shared with a 3rd party without consent. They also take on a non-intuitive approach in regards of configuration, relying on hard-coded configuration in case it was commented out / removed from the configuration file.
We have confirmed that removing the hard-coded values from the source code and all possible configuration options does not prevent synapse from exchanging data with other servers in a secure manner to the best of our knowledge. We have been running such a setup on some of our Homeservers for several months without any issue.

Media repository

Users can exchange files, images, video and audio using the Media Repository feature. Upon successful upload, a URI is returned that can be embedded in messages or used to build a classic URL to access the media. The media repository is primarily used to store users' avatar for their profile, but can and will contain sensitive and highly private data, like pictures of one's family, PDFs of private scanned papers, proprietary and closed source project files/documents, etc.
In terms of privacy and access security, it has two major issues:

- It is not possible to delete/remove a file from the repository using the regular Client API.
- Files are directly accessible via a public, unauthenticated endpoint/URL.

Commented [26]: your notary server (by default matrix.org) is still used to fetch signing keys (not TLS certificates), as per https://matrix.org/docs/spec/server_server/r0.1.2#querying-keys-through-another-server. You can configure whichever notary you want.

Commented [27]: this is just an oversight.

Commented [28]: only if you have matrix.org configured as the notary server.

Commented [29]: Notary servers are optional; you can also query the target servers directly; it just uses more traffic and will fail if the target server is temporarily offline - hence using a notary.

Commented [30]: If the files are in an end-to-end encrypted chatroom, they are essentially useless without the encryption keys for the message in question - so the fact that an attacker could access the data becomes much less significant.

Commented [31]: For the record, https://github.com/matrix-org/matrix-doc/issues/701 is the MSC for authenticating these URLs in future.

Each file is given a random ID. While IDs can't be guessed, there is no protection against listing attacks where the attacker simply tries all possible IDs over several days. Each new listing would be easier than the one before using the knowledge of which IDs are already in use. Each listing would decrease in difficulty over time.
Riot does not inform the user of such lack of basic access control and privacy when uploading files to rooms. From our experience, users believe that access to files are controlled in the same way that access to those rooms is, keeping files private and inaccessible to anyone outside of a non-public room.
It is interesting to note that the undocumented server version of this API uses authentication in synapse while the client version does not.

Commented [32]: Such an ID has (26*2)^24 different combinations - i.e. 1.53e+41 options. If you query each option one by one over HTTP, say 10ms per request, this would take 4.8e31 years. The heat death of the universe is in roughly 1e10 years.

Commented [33]: all S2S APIs use authentication by default. the one in question here is obsolete, hence undocumented.

User profiles

User profiles hold the information about the display name and avatar of a user. Those are set in Riot in the Settings view. Access is unrestricted and unauthenticated using a specific endpoint, This means that the following information, if configured, is directly available publicly, in an unrestricted manner without the user informed and explicit consent:

- Their display name, which can include their real first/last/middle name.
- Their avatar, which may be a picture of themselves.

Commented [34]: This is so that in general user accounts are visible to the wider Matrix network. Server admins can of course restrict the endpoint if they desire, as per https://github.com/matrix-org/synapse/pull/5083. MSC1301 (https://github.com/matrix-org/matrix-doc/issues/1301) tracks a better solution to this.

Identity Servers

By default, synapse only allows two Identity servers to be used for the various 3PID interactions:

- matrix.org
- vector.im

The initial idea and concepts behind Identity servers was to be independent of Home servers and only hold association data. Homeservers would hold Administrative data for their use to interact with the user directly.
In practice, execution of this idea has lead to only trusting central servers and disallowing clients and users from picking Identity servers they trust: they must all be manually set in synapse default configuration. Such a change may or may not be possible, depending on the level of control the user has over the Homeserver configuration, or ability to reach/communicate with the system administrator.
Due to the difficulty of adding new ones, system administrators tend to either leave the default configuration or add new ones **without ever removing the default ones.**

Usage of Matrix.org and Vector.im

All services (hosted under matrix.org, vector.im and scalar.vector.im) are going through Cloudflare, a US-based CDN. TLS termination is done at the Cloudflare level, allowing them to decrypt and see in clear all the traffic coming in and out.

Commented [35]: For the record, the reason we use Cloudflare is to mitigate against DDoS attacks, which previously took Matrix.org entirely offline. Between a choice of our services being unavailable and the marginal risks of Cloudflare being a bad actor, we'd rather be online.

It is important to put this information in perspective of all the data/metadata shared given all the points above, allowing a foreign 3rd-party to directly have access to plain text traffic, private identifier, data and metadata without ever being mentioned anywhere.

Matrix.org security breach on Apr 11 2019

Commented [36]: The full details of this incident are available at https://matrix.org/blog/2019/05/08/post-mortem-and-remediations-for-apr-11-security-incident/ for reference.

Timeline and events

For those unaware, Matrix.org was breached by an attacker for several days which triggered service downtime and a full rebuild of the Matrix.org infrastructure. Many people were amazed to see how this did not impact their services and they could continue to talk to others without interruptions and their data was safe on their own servers. That being true for most people, the reality was not so straight-forward.
On their initial communication, they say:

The security breach is not a Matrix issue.

The hacker exploited a vulnerability in our production infrastructure (specifically a slightly outdated version of Jenkins). **Homeservers other than matrix.org are unaffected.**
While the security breach was not in of the Matrix protocol, other Homeservers **were** affected by it. As per our analysis above, we know that people hosting a typical stack would have the following services not available to them:

Commented [37]: This is taken out of context. The full paragraph was:

"The security breach is not a Matrix issue.

The hacker exploited a vulnerability in our production infrastructure (specifically a slightly outdated version of Jenkins). Homeservers other than matrix.org are unaffected."

The point being that **other homeservers were not compromised by the attacker**. Obviously servers were indirectly affected by the outage.

- No key signature verification via notary, without visible impact to users.
- No push service, with direct impact to users (we were affected) for 24h+ reported to us.
- No bridges/bots/widgets hosted on the matrix.org Homeserver.

The announcement does not mention anything about collected data from Homeservers as part of the natural behaviour of the network, even though "*the attacker did have access to the production database*".

Deleted: t

In terms of personal identifiers like emails and phone numbers, you can read:
What has not been affected?
Identity server data does not appear to have been compromised
While technically correct, Identity data as most commonly understood is also present in the Homeserver database which **was** accessed by the attacker. They eventually posted a screenshot of various commands ran on a DB extracts: how much the attacker actually accessed is unclear given the Matrix.org communication.
Finally, on the 12th of April, the attacker used collected credentials (before being locked out) to take control of Cloudflare and pointing matrix.org to another website. The communication is not clear if the defacement affected the /_matrix API endpoints and its data coming from others servers.

Privacy and Security Impact

Taking into account all the data and metadata flowing to matrix.org, the security breach is a concerning event as an attacker had means to collect and process those data mostly found in system/application logs, database and reverse proxy logs. Such data could also be actively collected via a traffic sniffing of any sort if the TLS-terminated traffic at Cloudflare also flow unencrypted into the internal infrastructure.
The attacker could also have directly disrupted the federation in a significant manner via Denial of Service and cryptographic poisoning for the Notary and Push services. The attacker had access to hypothetical private room messages in which Integration services are used like bots or bridges.

Closing words

In a world where Privacy and Security are extremely hard to come by, protocols that give the means for decentralised, secure and private communications are highly sought, sometimes to the point where users will turn a blind eye to *minor* issues and *inconveniences* that might be solved down the line. Several of these shortcomings, leaks and issues have been brought up to the Matrix.org team and have witnessed first hand disregard for such reports, and purposeful de-prioritisation of issues while working on mxisd, our Federated Identity server focusing on privacy.
Privacy destruction is never about a single HTTP call, or a specific piece of data being leaked. It's always about putting together data from various sources, the amount and regularity of receiving such data. Privacy protection is a mindset, where one understand the cumulative effect of small, isolated pieces of data when put together.
By releasing v1.0, Matrix.org makes a promise of a secure and self-contained protocol while promoting privacy. But at the same time, has a near-monopol in the whole ecosystem in terms of client and server use: Riot and synapse, also labelled "reference implementations". We believe that reference implementations should reflect the core values of the protocol. They currently fail to do so and instead produce a near-centralised network which fails to protect people's privacy.
Security breaches in Matrix.org are an important reminder that we also are at the mercy of 3rd party entities with which we share our personal information unknowingly. They might leak private data unintentionally/unknowingly but still with a strong impact on the user, like it has happened many times in the past with security breaches across the Internet.
While users on the matrix.org Homeserver have to explicitly agree to the Terms of Use and the Privacy policy, no agreement is ever sought from users on self-hosted servers that also use matrix.org and vector.im. How is their data handled? Are they processed in some way? Which method of lawful processing under GDPR allows for this constant sharing of (meta)data? We hope such questions will be answered to ensure users' privacy is handled appropriately.

We do not claim we have made a full investigation or review, but we hope these notes will be useful for you to better understand:

- How Matrix works.
- The entities behind Matrix.org and how they relate to each other.

Commented [38]: they do not.

Commented [39]: The attacker could also have done a lot of very unpleasant things. https://matrix.org/blog/2019/05/08/post-mortem-and-remediations-for-apr-11-security-incident has the full analysis.

Commented [40]: Yup, it's true that bug reports get prioritised in the context of Matrix as an overall project. For instance, changing an API to stop a malicious homeserver admin depublishing an email address from an identity server is way way way less important than, say, fixing room hijack attacks, or making e2e encryption work reliably.

Commented [41]: given the protocol only exited beta a few days ago, it should not be that surprising that the reference implementations dominate the ecosystem currently.

Commented [42]: This is simply false - as pointed out by the author above, "people were amazed to see how this did not impact their services and they could continue to talk to others without interruptions and their data was safe on their own servers. That being true for most people"

Commented [43]: As per above, the question of how to handle data processing for users of the matrix.org/vector.im identity & integration servers is valid one that we need to tackle.

- What happens when you use Riot and synapse with only changing your Homeserver URL.
- How your private data and metadata are sent to those entities most likely without your knowledge or consent.

That you'll be able to make an educated and informed decision when choosing which Client, Homeserver, Identity server and Integration server you wish to run in the future. That you'll know which questions to ask when looking for the next best thing.
To discuss further, come to our Matrix room: #kamax-matrix:kamax.io.
- *The Kamax.io Team*

Commented [44]: What Max has forgotten to disclose is that he's working on a hostile fork of Matrix.