Security Mechanisms & Procedures

An introduction to Orbit security documentation

By Anders Ingemann
Operations Manager at Orbit Online

Welcome to this introduction to Orbits security documentation

These areas include, but are not limited to

Welcome to this introduction to Orbits security documentation 1
Orbit Online maintains a high level of security in orbit 2
User Access To Orbit 3
Manage Permissions In Orbit 4
Integration of External Services 5
Access of Orbit Online Personnel 6
Data Encryption Policy 7
Handling Security Exploits & Incidents 8
Isolation Of Components 10
The Integrity Of The Orbit Codebase 11

Orbit Online maintains a high level of security in orbit

This is achieved by employing a holistic view on the ecosystems that Orbit is developed in, deployed to and used in.

Since a chain is only as strong as its weakest link, Orbit Online iteratively identifies areas that could benefit most from an improvement in procedures, software, or hardware.

Once an area has been identified, it is analysed for potential weaknesses. Thereafter, a plan is formed to migrate to a better security model, which is then rolled out in the organisation in a way that is least likely to disrupt the workflow of Orbit users.

A multitude of security procedures are employed for Orbit Online to ensure that the above-mentioned areas are adequately covered.

User Access To Orbit

Authenticate a user in Orbit

First and foremost, unauthorised access to Orbit must be prevented. Since there is more than one way to authenticate as a user in Orbit, all avenues of ingress are scrutinised for potential exploits.

Username & password

The simplest form of authentication is by using a username & password. Here, Orbit ensures that passwords are saved only as hashes in the database: Each password has a unique salt and goes through 10,000 rounds of SHA512 before it is saved.

Consider the time it takes the server to respond to a failed login attempt, as an example of a non-obvious security measure that Orbit Online has taken with Orbit. This may reveal whether a given username is valid since no hashing takes place when the username does not exist. Resulting in a faster response time. This is mitigated by delaying responses of failed attempts by a random amount of time.

Token authentication

API integrations do not use username & password authentication for access to Orbit. Instead, they possess a single bearer token that Orbit handles in much the same way as passwords (i.e., hashed multiple times).

ADFS

Orbit integrates with Microsoft Active Directory via ADFS. Here, much of the security relies on ADFS being secured properly by the customer. When ADFS directs a user to Orbit, with a signed SAML LoginRequest in hand, Orbit naturally performs all necessary signature verifications. Orbit also complements this with internal permission checks of the user logging in.

Manage Permissions In Orbit

Assign roles to users

Multiple ways of assigning roles to users in Orbit

- Data import from Active Directory or an ERP system
- ADFS - Active Directory Federation Services
- Manually by Orbit Online developers
- Set by customer Admin users directly in Orbit

All of the above-mentioned methods require appropriate verification procedures that are adapted to the circumstances in which they are deployed.

Assign roles by import

Before a data import can assign roles to users, the customer must ensure that the users who can affect a change in the data export end do not have fewer permissions on the data import end. Similarly, the customer must ensure restricted access to any system that may affect what roles an ADFS server instructs Orbit to entrust a given user with.

Manual process

With regards to manually assigning roles, Orbit Online developers never perform any assignments without explicit written instructions to do so by previously agreed upon employees of the customer.

User assign roles to other users

Orbit also provides the ability to enable users to assign privileges to other users through their profile. In these circumstances, Orbit Online makes sure that no combination of the assigned rights allows any user to elevate the permissions to a level beyond the ones currently assigned.

Integration of External Services

Security procedure and permissions

External services that integrate with Orbit are given a minimal set of permissions required to perform their function.

Validation rules

External access is validated through the token authentication mentioned in the article - User access to Orbit. Orbit Online keeps an encrypted backup of those tokens. However, their safekeeping on the customer side is undoubtedly the responsibility of the customer.

Transferring of the access token

The tokens are transferred via a two-channel method. Commonly, Orbit Online encrypts the secret strings with a password, sends the ciphertext via email, and transmits the encryption key via either a phone call or text message. Upon transmission, both the ciphertext and the password are deleted from Orbit Online's records. Reducing the likelihood of an attacker gaining access to Orbit if one, or both, communication channel should suffer a security breach.

Orbit Online recommends that your organisation follows the same procedure.

Access of Orbit Online Personnel

How security sensitive situations are to be handled

Personnel access roles

Access of Orbit Online personnel to servers and services is restricted using a matrix-based system where employees are divided into roles. Where any new assignment is accompanied by a thorough briefing, including updates at a later date. The briefing includes details on how security sensitive situations are to be handled, which precautions must be taken to avoid potential security issues, how to recognise potential weaknesses in security before they are exploited, and how to act in case of a security breach.

Orbit Online constantly assesses employee access to its services and determines whether any rights should be revoked, based on whether a person still requires access to a given service. The customer also needs to be aware of the fact that GDPR prohibits any Orbit Online employee to relay any of the customers' data to the customer without written consent.

Two-factor authentication

Orbit Online secures all services it uses with two-factor authentication using varying mechanisms. Where possible, YubiKeys are associated with the logins, though many services only support TOTP, which is then used in its stead. Services that support SAML authentication are linked with the G-Suite accounts used by Orbit Online, allowing them to piggyback on Google's well-tested and constantly monitored login service that supports 2FA.

Employee access security to Orbit solutions is currently being hardened with 2FA as well.

We have integrated YubiKeys into the terminal login procedure to Orbit servers. This means that employees must have a physical token to access any server. Currently, some employees still use file-based, passphrase-protected, authentication certificates.

Handling of customer data

At Orbit Online, all workstations containing sensitive data use encrypted disks, nullifying the possibility of data leaks in the event of physical theft.

Data Encryption Policy

On both backup and transferred data

Transferring data

Orbit Online enforces a strict policy dictating that all data must be transferred encrypted.

This policy spans through:

- Customer data-transfer to and from Orbit
- Machine-to-machine transfers between workstations on Orbit Online's internal network
- All internal employee communication channels
- Any communication with services used by Orbit Online employees or Orbit itself

Data backups

Before being sent to a remote location, customer data backups are encrypted with a key known only by a limited set of Orbit Online employees. Neither the backup service supplier nor most of the Orbit Online support personnel, are therefore able to access these backups.

Handling Security Exploits & Incidents

How critical issues are identified and fixed

Mitigating security exploits

The majority of the code in Orbit undergoes reviews and is scrutinised for security exploits multiple times a year, where critical issues are fixed and promptly deployed to vulnerable installations.

The fixes are also backported to solutions that may run older versions of Orbit. Orbit Online does however try to keep all customer solutions up to date with the latest stable version of Orbit at all times.

Orbit relies on a considerable number of supporting services to perform its functions. Similar to how Orbit itself is monitored for issues. Orbit Online employees monitor the supporting services for any vulnerabilities that have been published by their maintainers or security researchers.

To facilitate a short reaction time when exploits are published, all Orbit servers support being updated through a provisioning tool. This tool can disable services, update existing ones, or reconfigure them to mitigate any issues.

Handling of Security Incidents in Orbit

At Orbit Online the guiding metric for evaluating, mitigating, and finally fixing the root cause of security incidents is severity. During the processing of a security incident, the discovery of new information may change the severity and cause the process to accelerate or decelerate.

If the incident is reported by a user of the system, both the authenticity of the user and the veracity of the claim are confirmed first. This is done to prevent the process itself from being the cause for a security incident (e.g., causing a denial of service by claiming there is a critical, but hard to reproduce security vulnerability. And, requesting a shutdown of the system while it is being resolved).

Once the reported symptom or symptoms have been verified, a severity (high, medium, low) is assigned to the incident. This is done to make better-informed decisions later in the process. When determining the severity, the requests of the customer and their unique circumstances are, of course, taken into account.

Using the initial assessment of severity, a mitigation strategy is formulated. Here, multiple factors are taken into account:

- Is mitigation without first finding the root cause possible or at all effective?
- Will the mitigation cause interruption of service?
- Could mistakes made during the implementation of the mitigation cause further issues?
- Does this issue affect multiple customers?

Searching for the root cause

While the mitigation strategy is formulated, a search for the root cause of the issue is put into motion. The customer is informed of either the mitigation strategy that will be implemented, depending on the estimated completion of this search. Or said strategy is discarded in favour of fixing the root cause instead.

Once the root cause has been identified, other potential symptoms are extrapolated and evaluated as well. This may result in an escalation of the incident severity. A plan for fixing the root cause is then developed. Based on the estimate of the implementation and severity of the incident, an alternative mitigation strategy may be applied while the root cause is being fixed.

Using the root cause analysis, the actual impact of the issue is determined by way of audit logs and other forensic tools. Future potential impact assessment is performed as well. When all of the above information has been gathered, the results of the analysis are sent to the customer.

Depending on the severity of the issue and instructions from the customer, the root cause will be either hotfixed or added to the bug list of the next patch release of Orbit.

Isolation Of Components

Protecting data against malicious users

Regardless of how well a system is maintained, 0-day exploits do exist and require additional security measures to protect against security breaches.

Surface protection

For that reason, Orbit Online reduces Orbit's attack surfaces by requiring a user to be logged in before being able to send requests to any supporting services.

Public-facing services protection

To protect against malicious users, and attacks on the public-facing services, Orbit Online compartmentalised most of the code in Orbit. This ensures that a breach in one part of the system does not allow the attacker to fan out into the rest of the system unopposed.

The Integrity Of The Orbit Codebase

Using a code-signing scheme

Discovering malicious changes

The integrity of the Orbit codebase is first and foremost verified through git, which disallows any rewriting of versioning history without explicit consent. This means that any malicious changes to the codebase must be applied on top of the current history, making the likelihood of a reviewer discovering the change almost a certainty.

The signing procedure

Orbit Online is using a code-signing scheme where every change is cryptographically signed with a key that resides on a physical YubiKey. This ensures a form of 2FA since said key is impossible to read from the YubiKey. Instead, the YubiKey itself performs the signing procedure.