Aurex
Application Security & Privacy

July
2022

Abstract

This document is intended to answer questions, such as *How does Aurex help me ensure that my data is secure?* Specifically, this document describes Aurex security and privacy processes within the product

How do we protect your information?

We implement a variety of security measures to maintain the safety of your personal information when you place an order or enter, submit, or access your personal information. We offer the use of a secure server. All supplied sensitive/credit information is transmitted via Secure Socket Layer (TLS) technology and then encrypted into our Payment gateway providers database only to be accessible by those authorized with special access rights to such systems, and are required to keep the information confidential. At the same time, every customer who purchases Aurex on an on-premises setup, gets a separate instance of MySQL database that ensures no 2 customer data and clubbed together and are secure with explicit boundaries set. Aurex is built on the best practices of application multi-tenancy.

Introduction

Aurex is an integrated audit risk and compliance platform developed by Beinex, designed to empower organizations and financial institutions to meet their most challenging requirements, leveraging cutting edge technology and the power of AI to remove various Internal Audit and Risk pain points while cutting down the cost and saving valuable time and resources.

Aurex helps organizations to identify and mitigate risks by combining four independent processes in silo, onto one single integrated platform – Continuous audit, audit analytics, audit management and risk management. Aurex is available as a SaaS service on Cloud (or) as a configurable service on-premises.

Aurex Security Layers

Authentication Security

Authentication verifies a user's identity. Everyone who needs to access the application - whether to manage it, browse or administer content—must be represented as a user in the Aurex application. The method of authentication may be performed by Aurex application ("local authentication"), or authentication may be performed by an external process. In the latter case, you must configure Aurex application for external authentication technologies such as SAML, or OpenID. In all cases, whether authentication takes place locally or is external, each user identity must be represented in the Aurex application repository. The repository manages authorization meta data for user identities.

Although all user identities are ultimately represented and stored in the Aurex application repository, you must manage user accounts in an identity store. There are two, mutually exclusive, identity store options: Active Directory and local. Aurex application will be optimized for Active Directory authentication in a future release. Alternatively, if you are not running an LDAP directory, you can use the Aurex application local identity store.

As shown in the following table, the type of identity store you implement, in part, will determine your authentication options.

Identity Store	**Basic**	**SAML (Okta)**	**(OAuth)**
Local	X	X	X
Active Directory	X	X	

Access and management permissions are implemented through RBAC. Roles define which users are administrators, and which users are content consumers and publishers on the application.

Local authentication

If Aurex is configured to use local authentication, then the users are authenticated when the sign-in and enter their credentials

To enable this scenario, you must first create an identity for each user. To create an identity, you specify a username and a password. To access or interact with content on the application, users must also be assigned a role. User identities can be added to Aurex application in the server UI.

You can also create groups in Aurex to help manage and assign roles to large sets of related user groups (e.g., “Marketing”).

When you configure Aurex for local authentication, you can set password policies and account lockout on failed password attempts.

Active Directory authentication

You can also configure Aurex to use LDAP for user authentication. Users are authenticated by submitting their credentials to Aurex, which will then attempt to bind to the LDAP instance using the user credentials. If the bind works then the credentials are valid and Aurex grants the user a session.

“Binding” is the handshake/authentication step that happens when a client tries to access an LDAP server. Aurex application does this for itself when it makes various non-authentication related queries (such as importing users and groups).

You can configure the type of bind you want Aurex to use when verifying user credentials. Aurex supports simple bind, where credentials are passed directly to the LDAP instance. We recommend that you configure SSL to encrypt the bind communication

SAML

You can configure Aurex to use SAML (security assertion markup language) authentication. With SAML, an external identity provider (IdP) authenticates the user's credentials, and then sends a security assertion to Aurex application server that provides information about the user's identity.

2-Factor authentication

The classic authentication approach for web applications requires a user to enter a username and password. However, things like password reuse, poorly encrypted passwords, social hacking, and hacked databases make even a secure password vulnerable. By requiring users to add a second factor to their authentication flow, an account with a compromised password will still be secure.

Mobile phone 2FA has become the industry standard, as most people carry their mobile phones at all times. It is a user-friendly flow, and dynamically generated passcodes are safe to use and users can receive special tokens through SMS or a dedicated app

OAuth (**Available in a future release)

OAuth is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google. After they have successfully signed in to their IdP, they are automatically signed in to Aurex. To use OAuth, the Aurex application must be configured to use the local identity store. Active Directory or LDAP identity stores are not supported with OAuth.

Data access and source authentication

You can configure Aurex to support a number of different authentication protocols to the MySQL backend database of Aurex application. Data connection authentication may be independent of Aurex authentication.

For example, you may configure user authentication to Aurex with local authentication, while configuring OAuth or SAML authentication to the MySQL database.

Authorization Security

Authorization refers to how and what users can access on Aurex after authentication has been verified. Authorization includes:

- What users can do with content hosted on Aurex modules that includes risk, audit, analytics, timesheet, budget management, assessments, and dashboards
- What users are allowed to do with the MySQL database that is integrated to the Aurex application
- What tasks users are allowed to perform to administer Aurex, such as configuring server settings and other tasks.

Roles

Roles define who is an administrator. For non-admins, roles indicate the maximum level of access a user can have on Aurex subject to permissions set on content assets.

Permissions

Permissions determine whether a given user is allowed or denied to perform a specific action on a specific content asset.

As an administrator setting up Aurex application, it is important that you understand how permissions are evaluated. Understanding the Aurex permissions process will enable you to set up and configure permissions on the assets that you can control about how content and data is shared, published, viewed, extracted, and imported

Data Security & Privacy

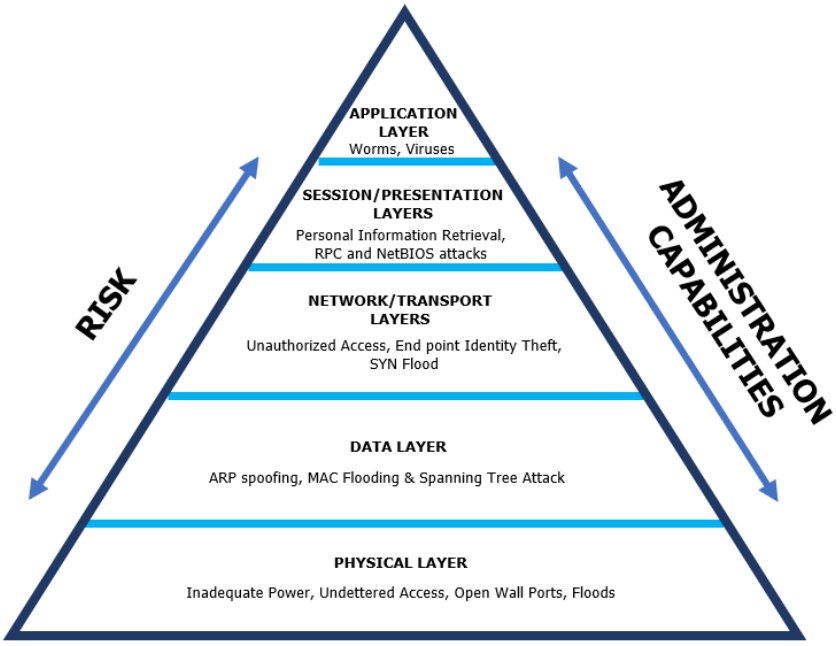
Aurex provides several ways to control which users can see which data. For MySQL backend database, you can also control whether users are prompted to provide database

credentials when they access a specific module in Aurex. The following three options work together to achieve different results:

- Database login account: Aurex makes use of the MySQL database's built-in security mechanism
- Authentication mode: Access to a module in Aurex is based on the permissions defined for the user who logs in
- User filters: You can set filters based on modules, permissions, entity and others in Aurex

Security Over The Network OSI Layers

In the OSI model, security is addressed at each layer of the OSI model, shown below. By comparing in depth, the OSI model with the concept of application security goes to prove that securing enterprise application is more than authentication, encryption, OS hardening, etc. At each level of the OSI model there are security vulnerabilities and therefore, security prevention measures that can be taken to ensure that enterprise applications are protected.



Layer 1 (physical security)

The Physical layer of the OSI model consists of the actual physical connections to and within the network, including wiring, devices used to connect the NIC to the wiring, the signaling involved in transmitting and receiving data, and the ability to detect signaling errors on the network media. It is the responsibility of the customer to take care of the layer 1 security in an on-premises infrastructure. The OSI Physical layer comprises the enterprise's physical and site security concerns, which includes all these aspects:

· Access Control

· Power

· Environment

· Smoke & Fire

· Water

· Backups

Layer 2 (switch security)

The Data Link layer is primarily concerned with physical addressing, line discipline, network topology, error notification, ordered delivery of frames, and flow control. Devices such as switches and bridges work at this level. It is the responsibility of the customer to take care of the layer 2 security in an on-premises infrastructure. Security threats that may occur at this level are the following:

· Gratuitous ARPs or ARP spoof

· MAC flooding

· Spanning tree attack

Layers 3–4 (router/firewall security)

Layer 3, otherwise known as the Network layer, and Layer 4, otherwise known as the Transport layer, are the most common forms of application/network security. In these layers, firewalls and router Access Control Lists (ACLs) can be found. The Network layer of the OSI model is where routing, layer 3 switching and IP addressing are defined. At this layer traffic is passed between network devices that are not on the same segment. At the Transport layer, data is divided into packets and then reassembled at the destination. Data flow control of and error checking are also provided at this level. TCP and UDP occur at the Transport layer. It is the responsibility of the customer to take care of the layer 3 & 4 security in an on-premises infrastructure. Security threats that occur at these levels include the following:

· Endpoint identification

· Unauthorized Internet access

· SYN flood

· Ping of death

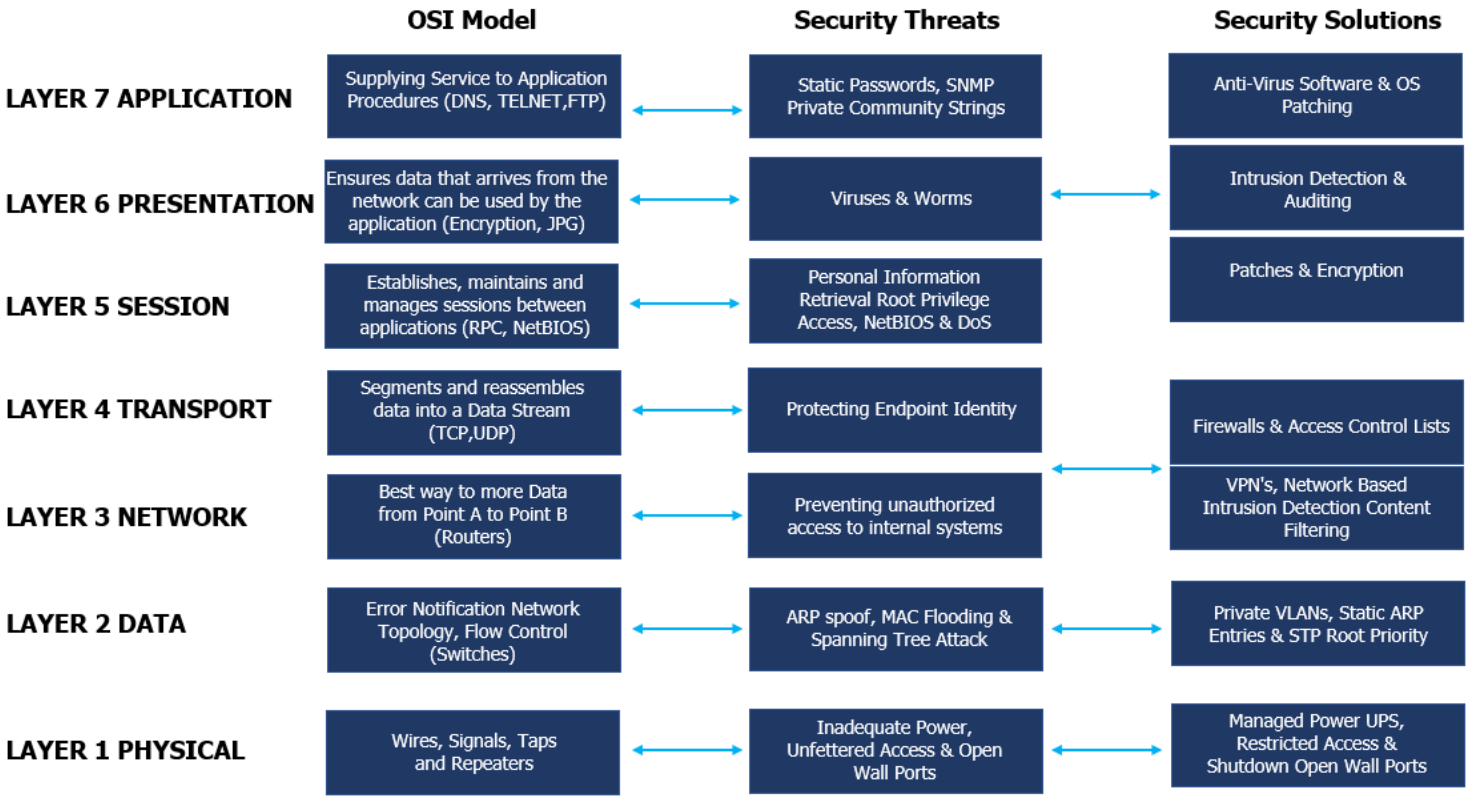
Layer 5–7 (encryption/authentication)

Layers 5–7 are known as the application set. This section will address the Session and Presentation layers as they relate to security. The Session layer (Layer 5) is responsible for creating, managing, and terminating sessions between applications and overseeing data exchange between the Presentation layer (Layer 6) and Transport layer (Layer 4). Login passwords, the exchange of user IDs, and accounting operations can all be handled at the Session layer. The Presentation layer defines how data is formatted, presented, encoded, and converted for use by software at the Application layer. Security threats pertaining to layer 5-7 will be handled by the robustness and effectiveness of the Aurex application build.

However, any installation of antivirus on the virtual machines running Aurex application and MySQL database that are governed by the client's IT security policies are to be catered by the client itself and not the Aurex team. Security threats that occur at these layers involve:

· Unauthorized Login/Password/Personal Data Access

· RPC & NetBIOS Attacks



Network Communication Security

Client to Aurex application server

Communications between Aurex and its clients use standard HTTP requests and responses. We recommend configuring Aurex for HTTPS for all communications. When Aurex application server is configured for SSL, all content and communications between clients are encrypted with Transport Layer Security best practices as the HTTPS protocol is used for requests and responses.

By default, passwords are communicated from browsers to Aurex using 4096-bit public/private key encryption. This level of encryption is not considered robust enough for secure communications. Additionally, this method, where a public key is sent to the recipient in the clear and without network layer authentication is susceptible to man-in-the-middle attacks.

To adequately secure network traffic from clients to Aurex application server, you must configure SSL with a certificate from a trusted certificate authority.

Aurex to MySQL database

Aurex makes dynamic connections to MySQL database to read / write all user interactions happening to the Aurex application. It uses native drivers to connect to databases whenever possible and relies on a generic ODBC adapter when native drivers are unavailable. All communications to the database are routed through these drivers. As such, configuring the driver to communicate on non-standard ports or provide transport encryption is part of the native driver installation. This type of configuration is transparent to Aurex

When a user stores credentials on Aurex, they are stored encrypted in the MySQL database Aurex is integrated with. When a user process uses those credentials to query the application and thereby the database, the process retrieves the encrypted credentials from the database and decrypts them in process.

Client access from the Internet

We recommend a gateway proxy server to enable secure client access from the internet to your on-premises Aurex application server. We do not recommend running Aurex in a DMZ or otherwise outside your protected, internal network.

Configure a reverse proxy server, with SSL enabled, to handle all inbound traffic from the internet. In this scenario, the reverse proxy is the only external IP address (or range of addresses if multiple reverse proxies are load-balancing inbound requests) that Aurex application server will communicate with. Reverse proxies are transparent to requesting clients, thereby obfuscating Aurex network information and simplifying client configuration.

Aurex to a SMTP server

You can configure Aurex to send event notifications to administrators and users as it supports TLS for the SMTP connection.

Communication with the MySQL database

Aurex can be configured to use Secure Sockets Layer (SSL) for encrypted communications on all traffic that is exchange with the MySQL database to and from the application server components.

Clickjack protection

By default, Aurex has clickjack protection enabled. This helps prevent certain types of attacks in which the attacker overlays a transparent version of a page on top of an innocuous-looking page in order to lure a user into clicking links or entering information. With clickjack protection enabled, Aurex imposes strict restrictions on hackers gaining access to any sort of confidential data of the customer.

NetBIOS name server, RPC portmap & sentinel reflection protection

By default, Aurex has NBNS, RPC portmap and sentinel reflection protection enabled. This prevents attacks where the victim’s DDoS response is much larger than the attacker’s query and not allowing to amplify the attacker’s capabilities. The attacker is then prohibited to send hundreds or thousands of queries at high rates to a large list of victims by automated the

process with an attack tool, thus causing them to unleash a flood of unwanted traffic and a denial of service outage at the target.

Encryption protection

Encryption helps protect information from data breaches, whether the data is at rest or in transit. For example, even if a corporate-owned device is misplaced or stolen, the data stored on it will most likely be secure if the hard drive is properly encrypted. Encryption also helps protect data against malicious activities like man-in-the-middle attacks, and lets parties communicate without the fear of data leaks. Aurex makes use of the default SHA-256 AES encryption mechanism, which is built into the Ubuntu operating system that can be configured at the time of installation. This way, any data written on the root volume or the data volumes of the Ubuntu OS is encrypted using a symmetric key mechanism. The symmetric key will be shared with the client to secure it. Data can only be decrypted using the same key, thereby preventing any sort of security breach. The same is the case for the MySQL database that Aurex backends with, wherein the AES encryption mechanism of Ubuntu is used.