

Security Architecture



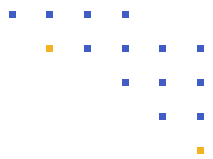


Table of Contents

Introduction..... 3

Kiteworks Architecture 5

Kiteworks Security and Compliance 12

Authentication and Authorization..... 18

Kiteworks Key Management 31

Data Loss Prevention (DLP) 32

Data Sovereignty..... 34

Secure Container and Remote Wipe 35

Compliance and Auditing 38

Network Security 38

Kiteworks Security for Android and iOS Mobile Devices 39

Resources 41

Introduction

The Kiteworks® content firewall helps IT executives lock down the exchange of confidential enterprise information with digital supply chain partners by unifying visibility and security across siloed third-party communication channels, including email, file sharing, mobile, chat, web forms, managed file transfer, and SFTP. Thousands of global CIOs and CISOs trust Kiteworks to give their organizations protection, privacy, and peace of mind. The content firewall integrates with the different point products to gain end-to-end visibility across the different content flows. It uses plugins and other connectivity technology to hook into the various content systems and deliver the content according to the organization's security policy. It keeps a detailed record of the actors, actions, assets, and circumstances of each transaction. And finally, it exports all that data in real time to the organization's security and compliance monitoring systems and personnel.

Kiteworks Security Overview

This section provides an overview of some of the main security features available to Kiteworks customers. The following pages describe these features and the set of security controls in detail.

System Security

The following features highlight Kiteworks system security:

- Hardened appliance with all unnecessary services disabled.
 - Kiteworks is built with only the necessary software for Kiteworks to run.
- Secure communication between servers in a Kiteworks cluster.
- Private cloud comprehensive security:
 - Enterprise Security
 - Encryption
- Limits attack surface to a tier that does not store sensitive information.
 - All requests are funneled through secure channel communication.
- SSH access is shut off by default.

Data Security

Kiteworks secures data in every state. Some of the key data security features are:

- Data in transit is always sent via a secure TLS connection.
- FIPS 140-2 certified.
- Access control lists at the application level to secure access to data.
- Integration with antivirus (F-Secure Atlant) and data loss prevention (ICAP) systems.
- Data sovereignty.
- Encrypts data at rest at 2 levels.
 - Disk level encryption is handled by dm-crypt, using AES-256-XTS.
 - File level encryption is done using AES-256-CTR.

Access Security

The following list highlights Kiteworks access security methods:

- Password policy.
 - Administrators control the password policy, including length of password, upper-case, lower-case, special characters, numeric characters, password expiration, and password history.
- LDAP/Active Directory integration.
- LDAP group integration.
- Integration with external systems for authentication and authorization.
 - SAML SSO.
- Kerberos SSO.
- OAuth integration authentication.
- Two-factor authentication.
- SFTP certificate authentication.
- Seamless connection to enterprise content sources using enterprise content management (ECM) native rights.
- Configurable watermarking of recipient emails.
- Restricted folders provided limited access to users.

SOC 2 Compliance (Service Organization Control) and FedRAMP

Kiteworks is authorized with FedRAMP and enables government employees at all levels to securely access, transfer, and collaborate on sensitive government documents.

Kiteworks offers the following FedRAMP package features:

- Separate customer virtual private cloud (VPC) for all processing.
- Dedicated servers.
- Data isolated from all other customers.
- Encrypted file storage and transfer.
- Remote wipe for all mobile clients.
- Reporting and audit trails.
- Continuous monitoring for intrusions and other threats.

Client Security

The following primary endpoint security features determine what actions administrators can perform to protect data:

- Secure desktop container with embedded productivity suite to prevent data leaks.
- Remote wipe for all mobile devices.
- All software updates (OS, services, application) provided by hardened Kiteworks update servers over SSL.
 - Packaging is digitally signed and validated upon update.
- Mobile security.

Compliance and Audits

Kiteworks stays abreast of compliance and audits requirements, including but not limited to:

- Reporting and audit command language (ACL) required for HIPAA, SOX compliance.
- All end user and administrator activity recorded.
- Syslog integration.
 - SIEM integration via the syslog.
- Periodic penetration tests and regular security audits.
 - These include web, server, Enterprise Connect, mobile and desktop clients.
 - High priority for stringent development processes to fix security issues.
 - Conform to Open Web Application Security Project (OWASP).
 - Multiple bug bounty programs.
- Audit logs.

Kiteworks Architecture

Kiteworks consists of three primary components.

- Application Tier: This is where the metadata from all the information inside the Kiteworks cluster is stored.
- Storage Tier: This is where customer data is stored.
- Web Tier: The Web Role is the endpoint which the DNS points at to serve web requests to end users.

Kiteworks supports a broad device list of operating systems, including Windows, Mac OS X, Android, and iOS.

Unlike other solutions, Kiteworks true multi-tier deployment offers security in depth to protect your files and secure your infrastructure. The flexible deployment model enables you to deploy Kiteworks in a way that fits your security needs.

Kiteworks patented auto-server configuration enables administrators and IT personnel with limited experience in distributed systems to build clusters of Kiteworks servers that communicate securely at different levels.

An important factor to note is that in highly regulated industries and organizations, other solutions in the market present some privacy concerns on stored data. Firstly, in some cases metadata, specifically file and folder names, are visible. Secondly, as the authentication provider, the hosted solution must be trusted to authorize access to files. Kiteworks is the solution to this problem. The web servers are installed in the DMZ and the data is stored in the private network behind the organization's firewall. Additionally, unlike less secure solutions, the Kiteworks database is not open to external access and everything resides on-premise.

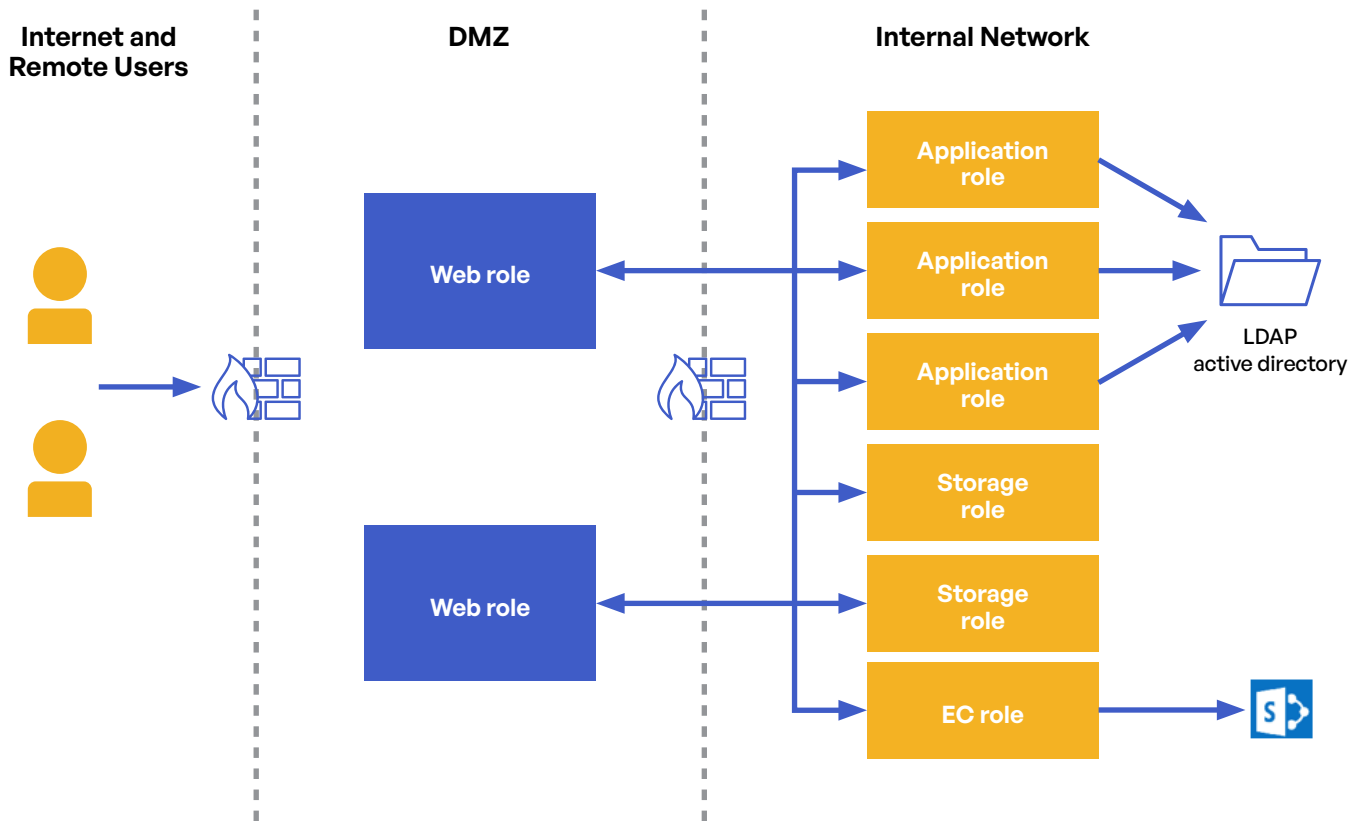


Figure 1. Kiteworks Components Overview

AWS Hosted Service Architecture

When Kiteworks hosts services on the Amazon Web Services (AWS) cloud infrastructure, the compliance responsibilities are shared. AWS computing environments are continuously audited with certifications from accredited bodies across geographies and verticals, including SOC 1/SSAE 16/ISAE 3402 (formerly SAS 70), SOC 2, SOC 3, ISO 9001/ISO 27001, FedRAMP, DoD SRG, and PCI DSS Level 1.i.

AWS continuously undergoes assessments of its underlying infrastructure, including the physical and environmental security of its hardware and data centers. Kiteworks benefits from this compliance and inherits these controls when hosting instances in the AWS infrastructure.

For single-hosted systems, Kiteworks adheres to the standard best practice of using VPC, Subnets, Network ACL, and Security Group to ensure that customer data remains separated on its virtual private cloud.

View the [Kiteworks Hosted Service Level Agreement](#).

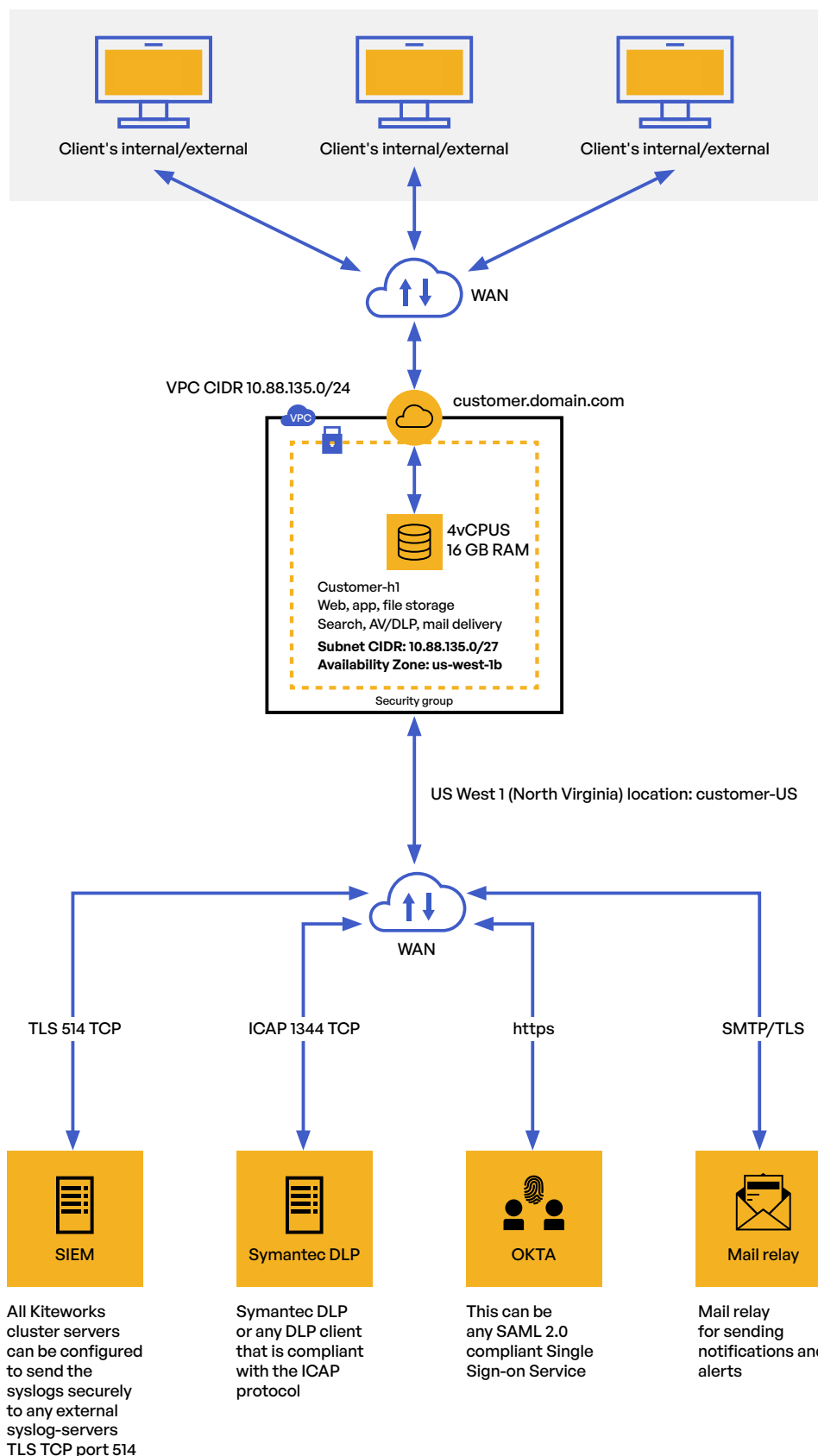
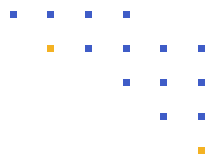


Figure 2. Single Kiteworks Server Setup in AWS



Application Tier Security

Encryption

To protect customer data in transit, Kiteworks supports TLS 1.2 with up to 256-bit AES encryption and no less than 128-bit encryption with the negotiation to TLS/AES-256 dependent on whether the end user's device or proxy supports TLS/AES-256.

Note: Kiteworks supports TLS 1.2 which cannot be disabled. The administrator has the option to enable TLS 1.0 and TLS 1.1, both of which are disabled by default.

Ciphers Used by Kiteworks

Mandatory Ciphers

- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES128-GCM-SHA256
- ECDHE-ECDSA-AES128-GCM-SHA256

Optional Ciphers

- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-RSA-AES128-SHA256
- AES256-GCM-SHA384
- AES128-GCM-SHA256
- AES256-SHA256
- AES128-SHA256
- ECDHE-RSA-AES128-SHA
- ECDHE-RSA-AES256-SHA
- AES128-SHA
- AES256-SHA
- ECDHE-ECDSA-AES128-SHA
- ECDHE-ECDSA-AES256-SHA
- AES
- CAMELLIA

Metadata Stored by Kiteworks

Metadata is defined as 'data about data' or data that describes other data. Kiteworks stores the following metadata attributes on the application tier's database servers:

User Metadata

- Name
- User Login (Email Address)
- Salted Password Hash (only for Kiteworks users)

File Metadata

- File Name
- File Description
- File Location
- File Size
- File Hash
- File Creation Date
- Client Create Date
- Client Modified Date (only when syncing from the Kiteworks for Desktop sync client)

Other Metadata

- Audit and reporting permissions.

External User Account Activation Process

There is no difference between internal and external users account activation process as all users who register in the system are active by default.

Storage Tier Security

Unlike other solutions, the Kiteworks storage tier is managed completely by the customer for on-premise deployments. All data being transported or stored within Kiteworks is encrypted with the highest levels of encryption available for each phase of its lifecycle, to protect files the moment they leave a client's computer. Data in flight and at rest are encrypted using military grade AES encryption set to its highest 256-bit level.

Securing File Upload and Download Requests

When a user uploads or downloads a file, the request is made to their server to authenticate and upload or download the file. In contrast from other solutions (where messages need to be sent from the main application API servers to the storage tier hosting the file), with Kiteworks no additional steps are required. Kiteworks has folder level permissions, and these permissions are stored in the metadata. Everything related to Kiteworks is stored in a database on-premise.

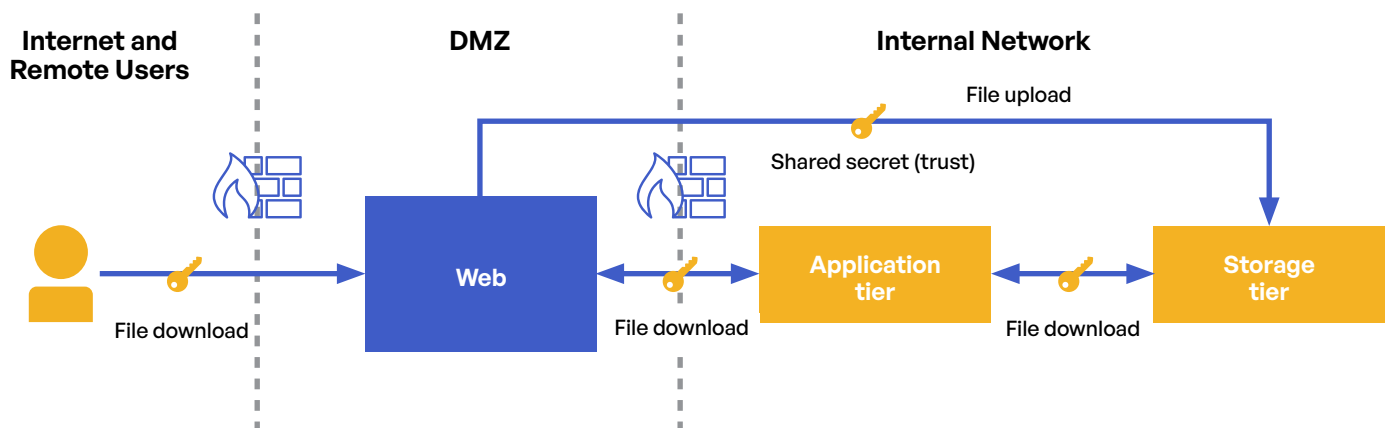


Figure 3. Secure File Upload and Download Requests

Security

Encryption in Transit

Client files are protected in transit between the web, application and storage servers using SSL/TLS and Advanced Encryption Standard (AES) encryption.

Encryption at Rest

There are two layers of encryption for data on the system. All client files are encrypted using FIPS 140-2 approved AES 256-bit algorithm for disk encryption. In addition, each file is individually encrypted with a unique key using FIPS 140-2 approved AES-256-CTR algorithm.

Data Replication

Customer files are stored redundantly within the cluster. Data replication can be configured to replicate data in different regions and also create high availability for the stored data with multiple servers.

Antivirus (AV)/Data Loss Prevention (DLP)

Kiteworks enables you to configure antivirus servers in your cluster that can be set up as a dedicated role to scan client files for malware.

For information on DLP, refer to [Compliance and Auditing](#).

Web Tier Security

The Web tier controls access to content on the backend servers, reducing attack vectors. The Kiteworks multi-tier architecture topology can be designed such that only the Web tier is exposed to the public. This prevents external users from accessing the corporate network. This also enables moving all metadata, file storage, and roles in Kiteworks to a protected area behind the corporate firewall for close monitoring.

Where Does the Metadata Reside?

Metadata always resides in the Application tier, with only local connections to the metadata. The only software that has access to the metadata is the Kiteworks code. There are no cross-server connections from the software to the database that houses the metadata.

LDAP Mapping (Geographical Segregation)

LDAP mapping enables administrators to match an LDAP attribute to a location. Whatever attribute is configured in the LDAP Mapping Value will map to the servers in that location. For example, the following illustration shows the LDAP Mapping Value field is Palo Alto, CA. This means all the files uploaded from any location in the world will be saved to the server in Palo Alto. The administrator controls the location where user files get uploaded.

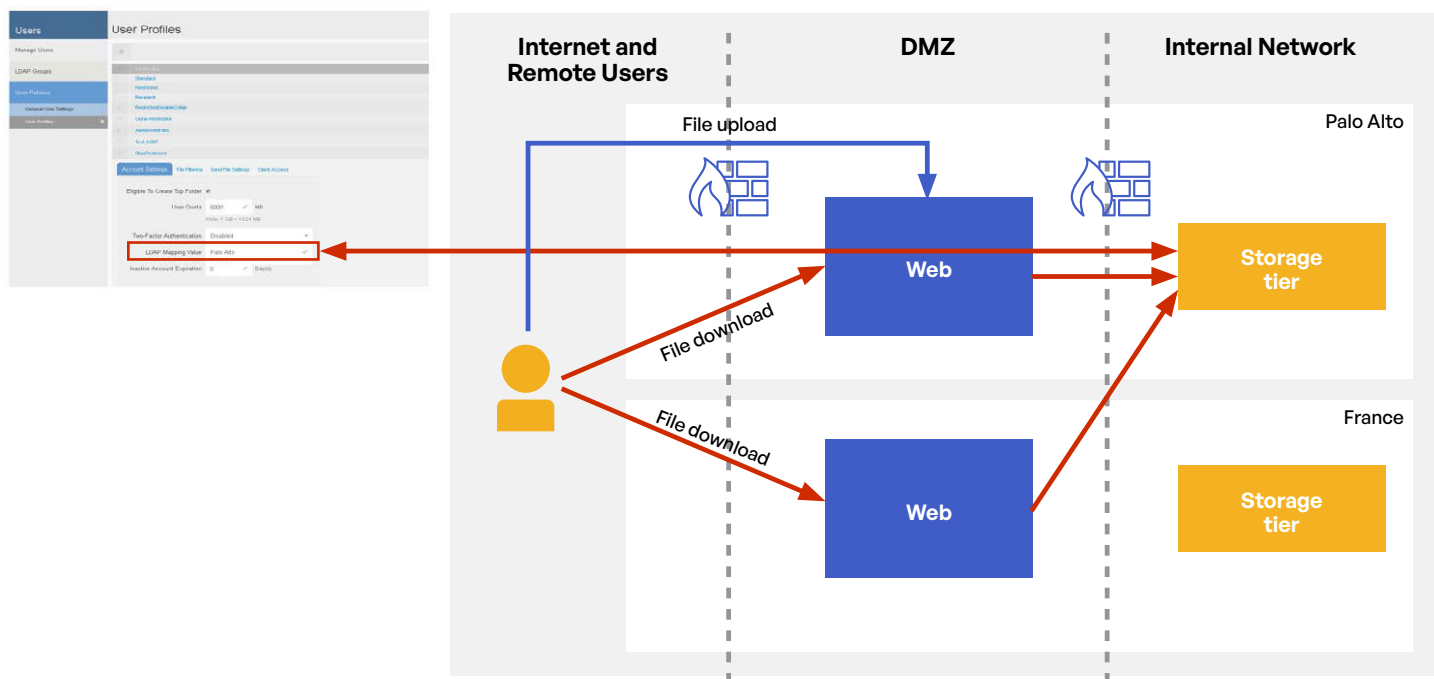


Figure 4. LDAP Mapping (Geographical Segregation)

Data Flow

The following diagram depicts the data flow of the metadata, download, and uploads from the user to the Application and Storage tiers behind the corporate firewall.

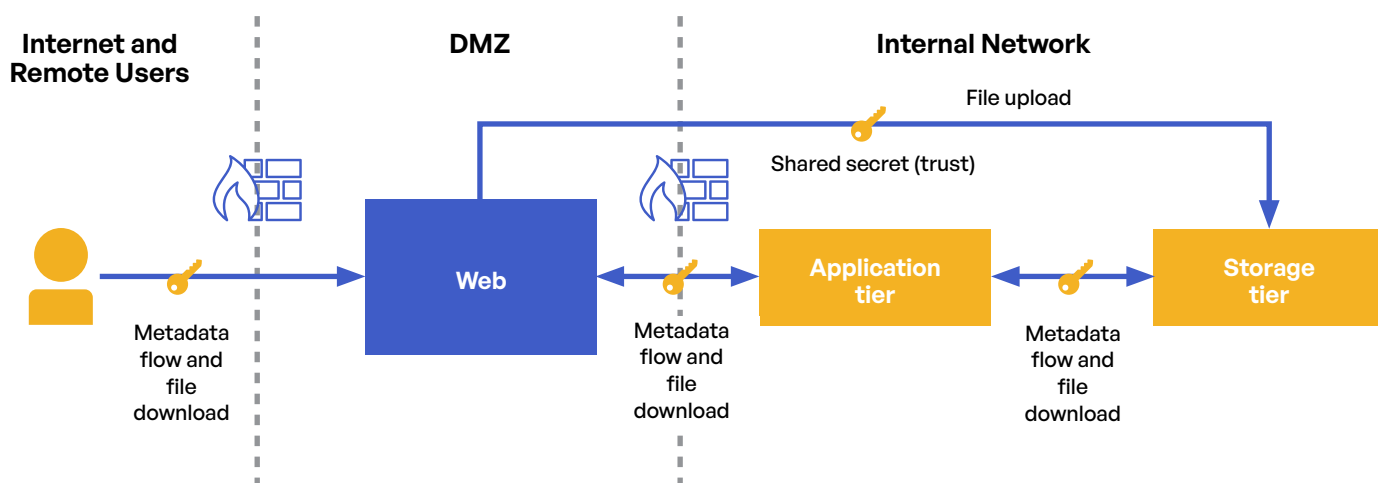


Figure 5. Kiteworks Secure Data Flow

Enterprise Connect (EC) Data Flow

The EC data flow is similar to all other data flows, but with an additional layer of security using an EC connector or conduit in Kiteworks to communicate with the Enterprise Connect data behind the corporate firewall.

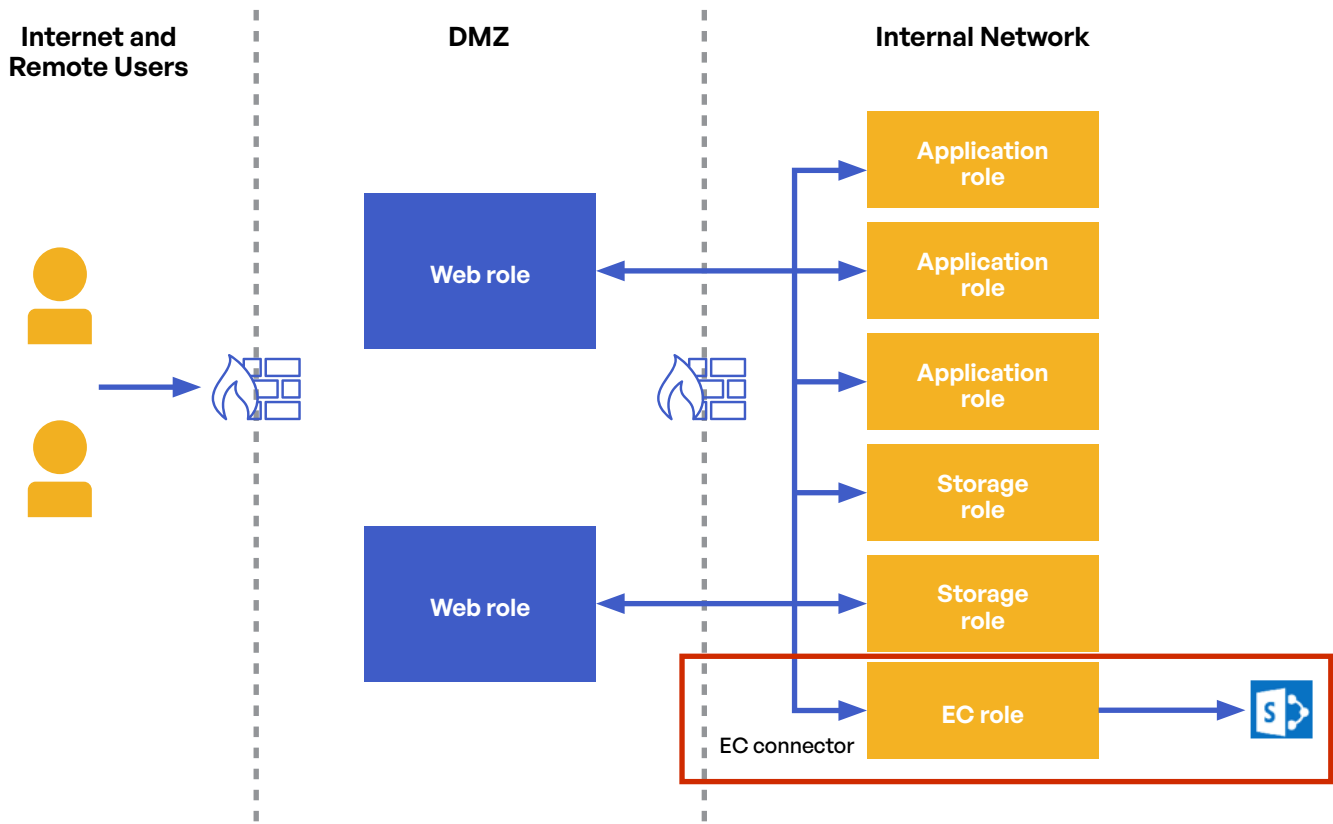


Figure 6. Kiteworks Secure EC Data Flow

Kiteworks Security and Compliance

Kiteworks offers a hardened appliance with all unnecessary services disabled, built with only the necessary software for Kiteworks to run. All communications to the server are over an SSL/TLS encrypted transport layer. All requests are funneled through a secure channel communication limiting the attack-surface to a tier that does not store sensitive information.

A multi-tier architecture provides flexibility in deployment options that are explained in the next section. SSH access is shut off by default and file storage is completely encrypted. Files are directly available only to the system they are mounted on, and services running on other systems access files through a secure internal API not available to the outside world.

Kiteworks integrates with an on-system antivirus scanner (F-Secure Atlant) and with third-party DLP systems that support the ICAP protocol. Periodic penetration tests and security audits are performed. These include the web server, Enterprise Connect, mobile, and desktop Clients. Kiteworks is built using OWASP secure coding practices and is FIPS-140-2 certified. All software updates (OS, services, and application) are provided by Kiteworks hardened update servers over SSL/TLS. Packaging is digitally signed and validated upon update.

Private Cloud: Comprehensive Security

Kiteworks implements comprehensive security according to the following pillars:

- Enterprise Security
- Web Application Firewall (WAF)
- HIPAA/PCI Compliance
- Encryption
- Mobile Security

Enterprise Security

Auditing and Reporting

Kiteworks provides reporting to both end users and administrators.

Administrators view all end user and administrator activity through the Admin Console on the Kiteworks server. They can also export activity logs to an external syslog server.

End users view activity through the Kiteworks web application, such as file uploads, downloads, deletions, comments, and tasks for folders that have been shared with them. Each Kiteworks folder has its own activity log which tracks its file uploads, downloads, and deletions as well as the creation or deletion of subfolders.

Antivirus (AV)

Kiteworks uses F-Secure Atlant antivirus to check incoming and outgoing files.

Restricted Administrator Access to Content

Kiteworks enables system administrators to have restricted access to content.

SSO (SAML/Kerberos)

SSO (SAML and Kerberos) work independently from LDAP, but can be used in conjunction with it. If a user is found in LDAP, user profile filters and mapping policies will be applied.

LDAP/AD Integration

Kiteworks can be integrated with LDAP/AD so users can sign in with their network password. Additionally, Kiteworks allows administrators to add LDAP groups.

FIPS: 140-2 certified

Data Loss Prevention (DLP)

Kiteworks integrates with several market-leading DLP products enabling content-aware sharing restrictions.

File Integrity Monitoring

Kiteworks offers file integrity monitoring with open source host intrusion detection security (Wazuh).

If any file modification is detected, a warning displays with the exact time and when the file modification was detected. The syslog shows the activity, before and after the modification of the file. The email daemon sends an email to the administrator. The Activities report displays the alert and the details of the activity.

Fail2ban Intrusion Prevention

Kiteworks offers the Fail2ban intrusion prevention framework to protect servers from brute-force attacks. Specific IPv4 and IPv6 addresses can be banned. The functionality offers:

- Banning of specific IP addresses.
- Banning of IP address ranges.
- Banning IPv4 or IPv6 addresses.
- Changing the number of failed attempts before banning.
- Changing the time IP addresses are blocked for, after failed login attempts.
- Maintaining a list of banned and allowed IP addresses (and the settings) after Fail2ban is restarted.

By default, Kiteworks blocks an IP address for 10 minutes after five failed SSH or SFTP login attempts. Blocked IP addresses are then listed in a Banned IP list.

Web Application Firewall (WAF)

Kiteworks utilizes the ModSecurity Web Application Firewall (WAF) to actively defend against web-based attacks. The ModSecurity WAF is installed on every Kiteworks web node.

ModSecurity protection is based on a local database (ruleset) of known signatures of attacks (or rules). Each rule in the ruleset is comprised of two parts: a set of characteristics that define a known attack and an assigned risk score. Every web request arriving at a Kiteworks web node is analyzed and compared against the local ruleset. If it matches, the assigned risk score is accumulated to the request's risk score. Once a request has accumulated a "high-enough" risk score, it is blocked by the WAF.

The Kiteworks ModSecurity signatures database is specially curated by internal Kiteworks security researchers and is continuously maintained by them. This allows for maximizing protection against attacks, while keeping false positives to an absolute minimum. Protection against all major web-based attacks, such as XSS, CSRF, SQL injections, and more, is included as they pertain to Kiteworks. The WAF ruleset on Kiteworks instances is updated with each release of the product.

Note: The Kiteworks implementation of ModSecurity is highly tuned to protect against attacks against the Kiteworks platform. It is not meant to be a generic WAF and must not be used as a protection mechanism for any other machine or platform on the network.

HIPAA/PCI Compliance

Kiteworks complies with Health Insurance Portability and Accountability Act (HIPAA) requirements reducing the risks of Protected Health Information (PHI) data breaches. Kiteworks complies with the following HIPAA technical safeguards.

Access Control

Kiteworks data security, access control, and reporting provide customers the tools to comply with HIPAA regulations.

Audit Control

All events are logged and presented in a report that is available to application administrators and system administrators of Kiteworks. These administrators have access to full audit logs of all actions taken on the system.

Integrity

Kiteworks ensures the availability, integrity, and confidentiality of customer information for HIPAA compliance.

Authentication

Kiteworks supports industry standard protocols for exchanging authentication and authorization information between different security domains while complying with HIPAA regulations.

Transmission Security

Kiteworks has outlined the security measures required to protect and secure the transmission of data to comply with HIPAA standards.

The Kiteworks for Outlook plugin can be used to encrypt email and securely send and request files through the email systems already in use. Files containing sensitive information can be shared and viewed while supporting HIPAA compliance.

Kiteworks is FIPS 140-2 Level 1 certified. It provides the necessary security and encryption technologies to meet HIPAA 2.0 compliance requirements. The Kiteworks application satisfies the "secure transmission" requirement by providing all communication (login sessions by users and administrators, uploads, and downloads) over HTTPS/SSL. Kiteworks also provides an audit trail of all transactions. The administrator can view the following information in the audit trail:

- Email addresses of senders and recipients.

- File names.
- Times and dates of file uploads and downloads.
- File sizes.
- Source IP addresses advertised by senders and recipients.

Kiteworks provides Payment Card Industry Data Security Standard (PCI DSS) compliance by providing absolutely no access to user files in the Kiteworks system. Files are stored dereferenced from their file name, encrypted, and are not accessible by anyone including Kiteworks support personnel. In order to access a file, you need to provide decryption information that is bundled into the link sent to the recipient.

Support for integration is available for Identity Providers (IdP's) that use SAML v2.0 such as Microsoft ADFS, PING Identity, etc. Kiteworks also supports authentication, providing a consistent SSO log-in experience across all devices, the desktop client, and the plugins.

Encryption

Kiteworks provides the following encryption features.

Encrypted Data While at Rest and in Transit

Files are encrypted while in motion and at rest using AES 256-bit disk encryption, and files through transit using a secure SSL connection. User files on the system are also encrypted at the file level; each file is encrypted with its own key. AES-256-CTR is used to encrypt each file.

256-bit AES

All data is stored either on the Kiteworks server or client. In the Kiteworks mobile apps, all data is secured with 256-bit AES encryption while at rest. As data is moved throughout the Kiteworks system, 128-bit or 256-bit TLS is implemented to protect data in transit. All data transactions are also logged by user/IP address. Significant metadata is captured for each transaction.

Ownership of Encryption keys

Data can be encrypted with data-owner encryption keys and stored in a cloud storage account. Administrators can be provided tools to modify keys for in transit as well as at rest encryption.

Key Rotation

Rotation of the actual key that encrypts all the keys for each of the individual files.

Mobile Security

Kiteworks provides the following mobile security features:

Encrypted File Storage

All files downloaded locally for offline use are stored on mobile device in encrypted states. Encryption keys are stored in a Keychain wrapper.

"Allow List" Helper Applications

Whitelisted app bundle allows the admin to control the "Open in" functionality. Only whitelisted apps are allowed to open files from Kiteworks.

Remote Wipe

Remote wipe can be done to wipe out account specific data/downloads from a device installation as a security measure. This could be used, for example, if a device is stolen, to be decommissioned, or to be re-assigned to another user.

Biometry Authentication

Facial or fingerprint recognition can be used as an additional layer of security to access the Kiteworks mobile application.

View-only Watermarked Images

Administrators can watermark images to control the preview for users assigned the Viewer role. If enabled, viewers can only preview watermarked files in the mobile app. They cannot download or open the files in another app.

Device Agnostic and Independent of EMM Products

All Kiteworks features operate without Enterprise Mobility Management (EMM) configuration.

Secure Offline Access

Ability to access offline on the mobile phone securely using biometric authentication.

Authentication and Authorization

The following features make Kiteworks a secure solution:

- Internal Authentication
- LDAP Authentication
- Kerberos SSO Integration
- SAML SSO Integration
- OAuth Integration
- Two-factor Authentication Integration
- SFTP Certificate Authentication

Internal Authentication

By default, Kiteworks provides secure authentication using username and password credentials, which are securely stored in the Kiteworks database. To enhance the security of this authentication system, Kiteworks provides the following mechanisms:

- All passwords are salted and hashed using a secure hashing algorithm prior to being stored in the database.
- Administrators can enforce policies to strengthen password security, such as:
 - Specify minimum password length and characters combination.
 - Specify password expiration.
 - Limit users from reusing previous passwords.
 - Disallow username/password auto-fill.

This internal authentication system is very useful in many scenarios, for example, for external users. However, customers are encouraged to integrate Kiteworks with enterprise authentication systems using LDAP, Kerberos, or SAML.

LDAP Authentication

LDAP is one of the authentication mechanisms supported by Kiteworks. With established connections to various LDAP sources, users are able to authenticate against an LDAP source instead of managing another set of credentials.

A user is found in LDAP using the user filter specified by the administrator on creation/update of the LDAP source. Kiteworks passes the user credentials via the LDAP protocol for authentication, and uses the response to determine if the user is able to bind. Filters are then applied to determine the type (profile) of the user. The system checks the user's DN in conjunction with the filter to determine if the user matches the filters.

LDAP protocol calls are made via a library that encapsulates the LDAP protocol. Kiteworks binds, queries, and processes the results from that library. The following protocol is used to make the calls: <https://tools.ietf.org/html/rfc4511>.

For enhanced security, Kiteworks also support the LDAPS protocol, where LDAP calls are made over secure SSL connections.

Responses received are typically result sets from the LDAP source of key/value pairs. The information on the wire depends on the queries and operations being called, but follows the LDAP protocol. Typical data would be filters, LDAP attributes, and LDAP attribute values. Furthermore, LDAP attribute values can be used for passing additional user information, such as display name, group membership, user type (profile), and location.

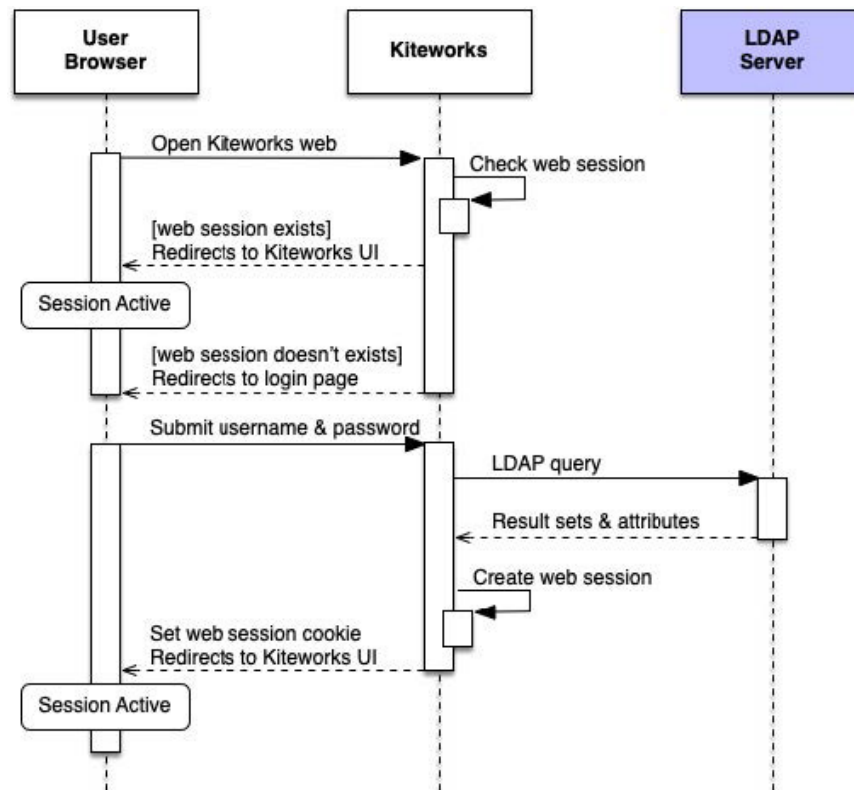


Figure 7. LDAP Authentication Flow

Kerberos SSO Integration

Kerberos is a mature, widely used network authentication protocol which allows nodes in a non-secure network to authenticate to one another using tickets in a secure manner. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. It has the following characteristics:

- It's secure. It never sends a password unless it is encrypted.
- Only a single login is required per session. Credentials defined at login are then passed between resources without the need for additional logins.
- The concept depends on a trusted third party – a Key Distribution Center (KDC). The KDC is aware of all systems in the network and is trusted by all of them.
- It performs mutual authentication, where a client proves its identity to a server and a server proves its identity to the client.

Overview

Kiteworks supports Kerberos Version 5 authentication protocol. A web server module is used for Kerberos authentication.

High Level Diagram

The following diagram identifies the parties involved.

- User browser as the authentication client.
- Kiteworks, which acts as a service server (a “principal” in Kerberos vocabulary).
- Ticket Granting Server (TGS) or Key Distribution Center (KDC), which acts as an Authentication Server controlled by the customer and external to Kiteworks, similar to SSO and LDAP sources.

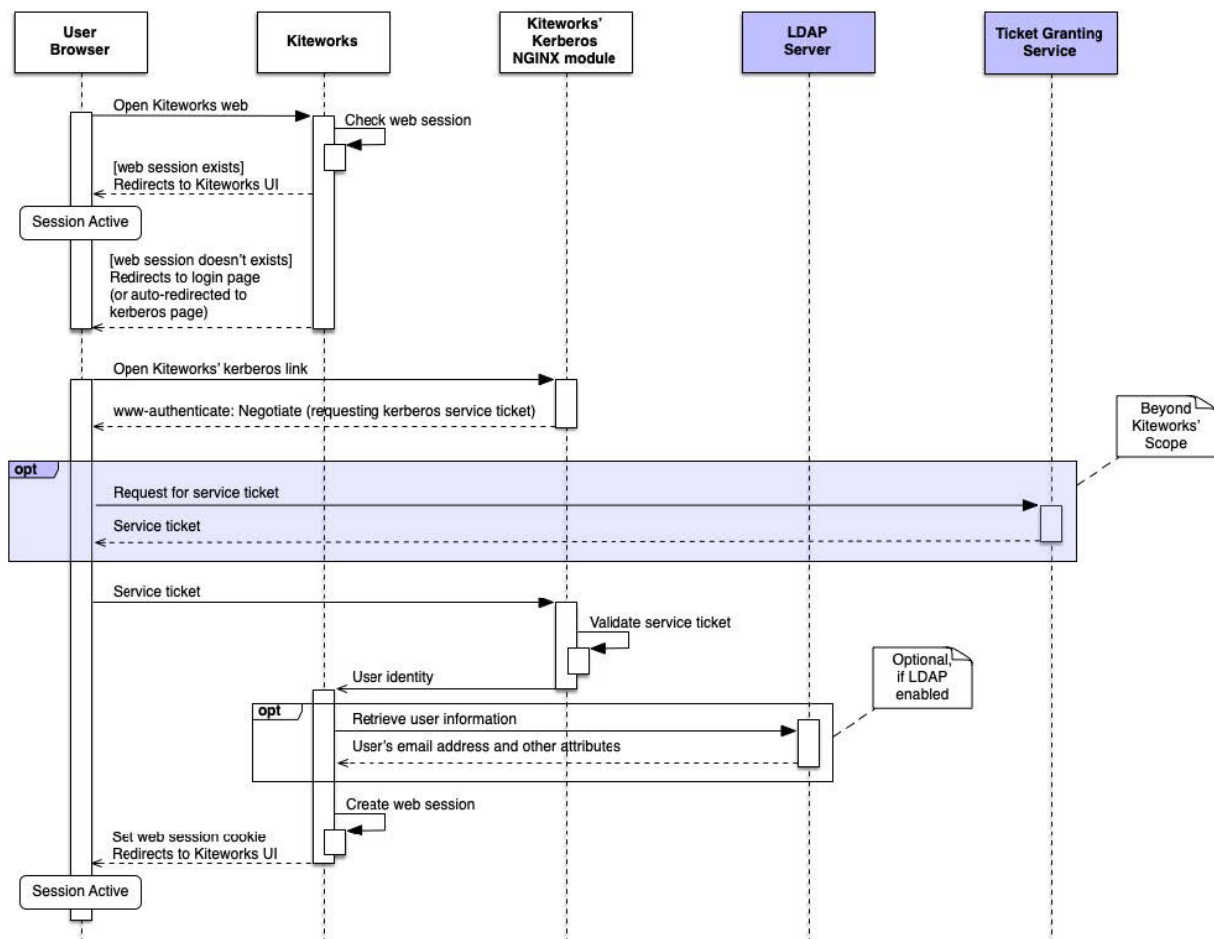


Figure 8. Kerberos Authentication Flow

How Kerberos Authentication Works

1. When a user's web session does not exist, the user clicks the "Login via Kerberos" link on the Kiteworks sign-in page. (Alternatively, the user can be auto-redirected based on the IP address.)
2. Kiteworks sends authentication negotiation (SPNEGO) to the browser, which expects a Kerberos service ticket in return.
3. The browser fetches the Kerberos service ticket for accessing Kiteworks from TGS/KDC. This activity is beyond the scope of Kiteworks.
4. The browser posts the Kerberos service ticket to Kiteworks.
5. Kiteworks validates the Kerberos service ticket.
6. On successful validation, Kiteworks extracts the user identity from the Kerberos ticket.
7. If LDAP/AD is enabled, Kiteworks queries the LDAP/AD for retrieving the user's email address and other user attributes using the identity obtained from the Kerberos service ticket. If LDAP/AD is disabled or if Kiteworks is unable to obtain the email address, it defaults to "identity@REALM" as the email address.
8. Using the email address obtained, Kiteworks creates a web session for the user.
9. Access to Kiteworks is granted and the user is signed in to Kiteworks.

Advantages to Kerberos Authentication

- User passwords are not sent over the wire.
- Leverages an existing enterprise class authentication mechanism.
- Users who are signed in to AD domains can conveniently sign in to Kiteworks without being prompted for a password.
- Eliminates the threat of exposure of the user password.
- Centralized federation provides a single point of Kiteworks application access, control, and auditing, which has security, risk, and compliance benefits.
- Users can obtain a Kerberos ticket for the Kiteworks application once and then reuse that ticket.

SAML SSO Integration

Overview

Security Assertion Markup Language (SAML) is a standard for exchanging authentication and authorization data between security domains. SAML is an XML-based protocol that uses security tokens to pass information about a principal (usually an end user) between a SAML Identity Provider (IdP), and a SAML service provider. SAML enables web-based authentication and authorization scenarios including cross-domain single sign-on (SSO), which helps reduce the administrative overhead of distributing multiple authentication tokens to an end user.

Kiteworks supports single sign-on via SAML 2.0 and can integrate with any IdP that is SAML 2.0 compliant. Kiteworks supports both IdP as well as SP initiated SSO flows.

In Kiteworks, the use of SAML allows a user to access and utilize the capabilities of Kiteworks from various devices: web, mobile, thick client, plugin, third-party product, or automation client. All communications within the control of Kiteworks operate over SSL/TLS.

Identity Provider (IdP)

An Identity Provider (IdP) establishes the identity of the user and authenticates the user with its configured authentication mechanisms. The IdP provides a SAML Response Message to the service provider containing assertions about the authenticated user. The assertions that Kiteworks supports are discussed within the SAML Response Message section. The IdP and the service provider are configured IdP initiated requests and responses.

Any IdP that supports SAML 2.0 can be utilized. Example IdPs are Azure Active Directory (AD), Active Directory Federation Services (AD FS) 2.0, Okta, OneLogin, and OpenIdP. Configuration must occur on both the IdP and Kiteworks service provider to successfully communicate between the external IdP and Kiteworks service provider. This configuration is dependent on the IdP. Kiteworks requires the Issuer ID of the SAML Response Message, the public key of the IdP, the login endpoint for the IdP, and the SP provider ID that the IdP is expecting. Kiteworks also supports SAML metadata to simplify configuration on both sides: SP metadata can be downloaded from the Admin Console on the Kiteworks server to be provided to the IdP, and IdP metadata URL can be specified in the console to automatically configure IdP settings on Kiteworks SP.

The IdP can also enforce authorization to Kiteworks via its processing engine for authentication requests.

Service Provider

Kiteworks is a service provider via the APIs that it exposes. The consumers of the APIs are the devices mentioned in the introduction. The service provider is partially responsible for the authorization of the user. The service provider consumes the SAML assertions provided in the SAML Response Message given by the IdP. Using the information in the SAML assertions, the identity of the user that was authenticated is obtained. The service provider then processes the SAML request based on rules to determine authorization of the user into Kiteworks. Authorization may be restricted due to licensing. The other assertions provided are used to update the user within the Kiteworks system. Details of the assertions are provided in the SAML Response Message section. The Kiteworks service provider provides a web cookie that is consumed by the RelayState value of the request. The web cookie identifies the user within the Kiteworks system. The cookie is then used by the RelayState as needed. For the Kiteworks web application this means that the user is able to utilize Kiteworks via a common web browser. For devices that require an OAuth token, the RelayState will be the OAuth endpoint which consumes the cookie and provides the caller with an OAuth code that can be redeemed for an access token.

SAML Request Message

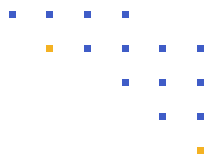
In service provider-initiated SAML SSO, the Kiteworks service provider sends an authentication request to IdP. All SAML requests involve the query string and are done via HTTP-Redirect communication to the IdP.

Request Type	Description	Sample
AuthnRequest	AuthnRequest is a protocol by which a request is made via a SAML AuthnRequest formatted message to the IdP. The message specifies the target application. For this to be initiated, the user is redirected to an endpoint on the Kiteworks service provider which generates the AuthnRequest http://en.wikipedia.org/wiki/SAML_2.0#_Authentication_Request_Protocol . The receiver may or may not support the RelayState argument back to the Kiteworks service provider. Kiteworks uses the state from the AuthnRequest to redirect the consumer back to the RelayState to finish the request flow.	<code>https://kiteworks.local/saml/sp/login.php?RelayState=%2F</code> Redirects to: <code>https://idp.local?SAMLRequest=fVLBUslwFPyVTu5p0za0NAPMoBxkBpUR9ODFCekDMrZJ7EtB%2F95SdMQL1327%2B97uvBHKunJi2vq9eYKPFtAHn3VIUPSDMWkbl6xEjcLIGIB4JVbT%2B4VIQiZcY71VtiIXkusKiQiN19aQYD4bkzeZsaQolAQFRZYkbDPIhopDFm8IByg5DGPYDIk%2BYCR4gQY75Zh0Rp0csYW5QS%2BN7yAWc8oKytJ1kooBFyl%2FJcGsS6ON9L1q77IDEUWV3WkTHrUp7RFDaz7KsmQo43xL4xQSyvNBRguel3TDFJcqL3ks0%2BiULSHB9DfArTXYtCsoDloBc9Pi78V0Oj30O9t7dCa0MmwhEOELqpt2VYQur3r7U5Qb0ulwjPqKKIlwfKn2JvuSm121zvdnEko7tbrJV0%2BrtZkMjr5ir6jZnLtrFF0yRydv%2BGh2zGfLW2liVc0OVP%2B%2F8jkGw%3D%3D&RelayState=%2F</code>

SAML Response Message

Kiteworks supports SAML 2.0 Response Messages sent by IdP via the POST method. This message can be service provider-initiated—a SAML message responding to a prior SAML authentication request sent by Kiteworks service provider or IdP-initiated, which does not have an associated authentication request. The focus of this section is on the assertions supported by Kiteworks. The basics of SAML 2.0 response messages are outside the scope. Kiteworks supports various assertions as each IdP can designate how they issue assertions. Kiteworks supports the following assertion attributes:

Alias of	Attribute Name	Attribute Values	Precedence	Description
N/A	UID	String: email	4	Required (unless using alias). Email of the user that was authenticated.
UID	mail	String: email	1	Support for IdPs that cannot specify UID assertion.
UID	<code>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress</code>	String: email	2	Support for Azure Active Directory (AD) and other IdPs, which uses this attribute name for email.
UID	<code>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name</code>	String: email	3	Support for IdPs where email is not available.
N/A	location	String		Optional. The location value to which location mapping will be applied.



How SAML Authentication Works

For service provider-initiated flows, authentication starts when the user performs SAML login from Kiteworks (as a service provider) as follows:

1. The user clicks the "Login via the External SSO" link on Kiteworks sign-in page. (Alternatively, user can be auto-redirected based on the IP address or email address.)
2. Kiteworks creates a SAML authentication request message.
3. Kiteworks redirects the browser via an HTTPS 302 redirect to the configured IdP's SSO Service URL, along with the SAML request message.
4. The IdP authenticates the user and redirects browser to a SAML response page, on which the SAML response is posted to Kiteworks service provider's Assertion Consumer Service (ACS).
5. Kiteworks service provider validates the SAML response.
6. Optionally, if LDAP is enabled, user's attributes are retrieved from LDAP.
7. If the SAML response validation is successful, Kiteworks creates a web session for the user.
8. Access to Kiteworks is granted and the user is signed in to Kiteworks.

For IdP-initiated flows, the user interacts with IdP, which then triggers the authentication to Kiteworks. For example, the user may click a sign in to Kiteworks link from an IdP/enterprise portal. The interaction between users and IdP in this scenario is beyond the scope of Kiteworks. Below is the flow of the IdP-initiated authentication:

1. Using a browser, the user requests IdP to perform an authentication to Kiteworks, for example by clicking a sign in to Kiteworks link.
2. The IdP authenticates the user and redirects the browser to a SAML response page, on which the SAML response is posted to the Kiteworks service provider's Assertion Consumer Service (ACS).
3. The Kiteworks service provider validates the SAML response. If LDAP is enabled, the user's attributes are retrieved from LDAP.
4. If the SAML response validation is successful, Kiteworks creates a web session for the user.
5. Access to Kiteworks is granted and the user is signed in to Kiteworks.

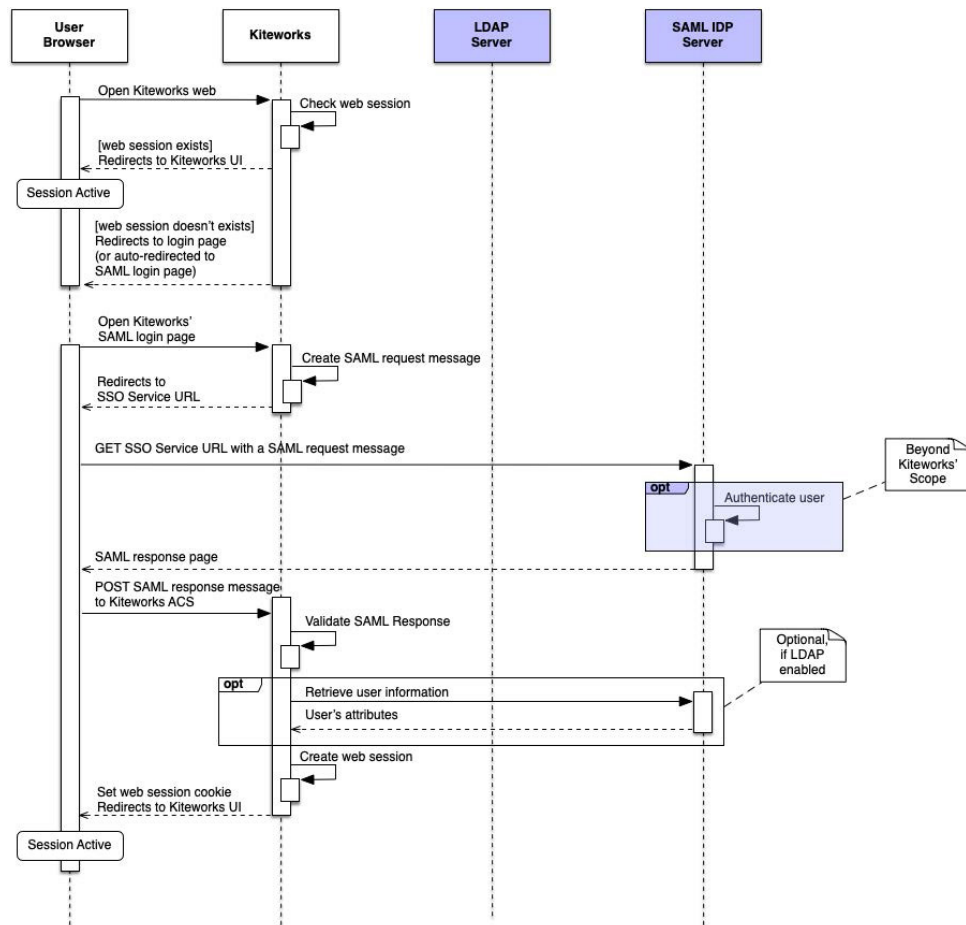


Figure 9. SAML SSO Service Provider-initiated Authentication Flow

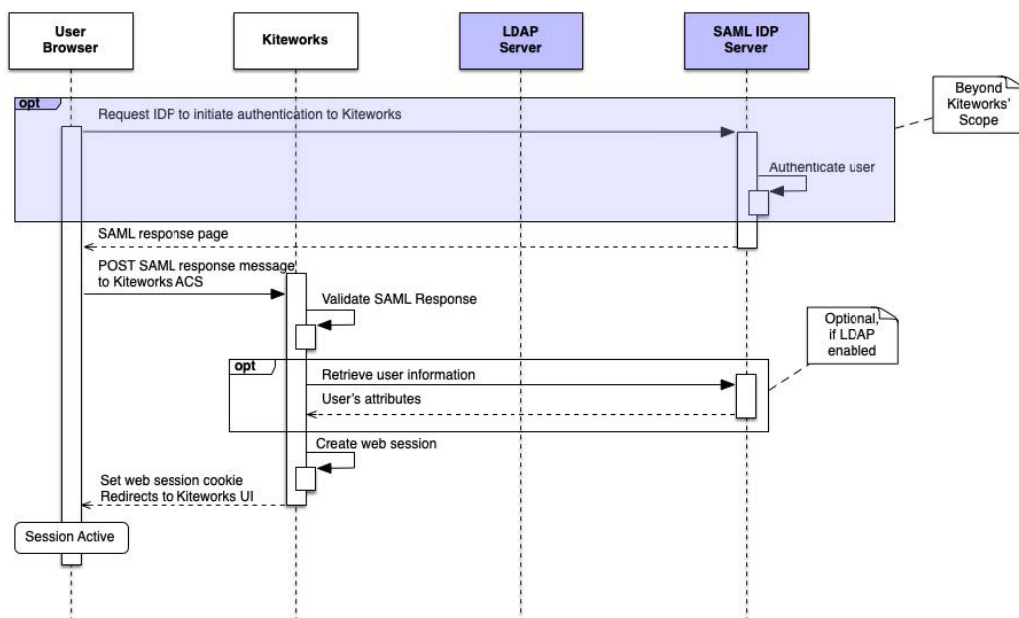


Figure 10. SAML SSO IdP-initiated Authentication Flow

Advantages to SAML Authentication

- User passwords never cross the firewall since user authentication occurs inside of the firewall and multiple web application passwords are no longer required.
- Web applications without passwords are difficult to hack, as the user must authenticate against an enterprise-class IdP first which can include strong authentication mechanisms. Centralized federation provides a single point of Kiteworks application access, control and auditing, which has security, risk, and compliance benefits.
- The ability to offer secure, scalable, standards-based internet SSO to customers, either as a value-added service, a competitive differentiator, or to satisfy customer demands.
- Ability to federate with other service providers, sharing user identity in order to deliver seamless, transparent, value-added services without requiring an additional login.

OAuth Integration

Overview

OAuth is an open protocol to allow secure authorization using a simple and standard method from web, mobile, desktop, and third-party client applications.

Kiteworks uses OAuth when users sign in to the server and gives access to the client application. Users are required to be authenticated using any of the authentication methods described previously. When authentication is successful, the client application retrieves an access token from the server that is used in further communications with the server during the session.

The Kiteworks OAuth 2.0 framework allows Kiteworks mobile apps, the Kiteworks for Desktop sync client, the Kiteworks for Outlook Desktop plugin, and third-party client applications to obtain limited access tokens.

How OAuth Authentication Works

User opens a client application and initiates login.

1. The client application opens an authorization endpoint on the Kiteworks server using a web-view component.
2. After verifying the client application's id, Kiteworks checks if the user's web session exists.
3. If the user's web session does not exist, Kiteworks redirects the client's web-view to the sign-in page and performs user authentication using one of the authentication methods (internal authentication, LDAP authentication, Kerberos SSO, or SAML SSO). Optionally (for third-party client applications) asks the user's consent to grant access to the client application.
4. If the OAuth authentication is successful, the Kiteworks server redirects the web view to a designated redirect URL and provides an authorization code for the client application.

5. The client application requests an access token to Kiteworks using the provided authorization code and the client application's credentials.
6. After verifying the authorization code and the client application's credentials, Kiteworks provides an access token to the client application.
7. The client application uses the access token to access user's resources stored in the Kiteworks server.

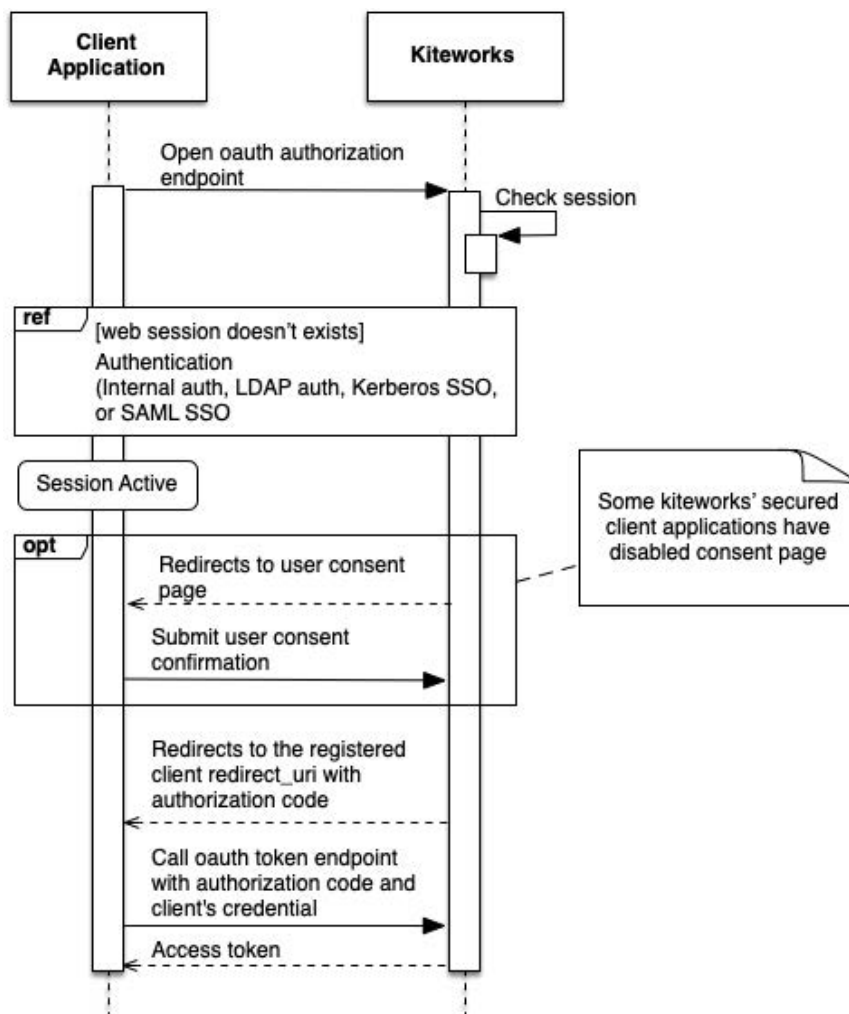


Figure 11. OAuth Flow for Client Applications

Advantages to OAuth Authentication

- Allows access for client applications without providing user passwords to the client applications.
- Access tokens given to the client applications are limited in scope and time.
- For third-party applications, administrators can specify a fine-grained access control by defining the scope on which access tokens are valid.
- Administrators and users can review currently active access token sessions and revoke them when necessary.

Two-factor Authentication Integration

Overview

Two-factor authentication (2FA) is an additional layer of security that requires the user to provide two means of identification from separate categories of credentials—one is typically a physical token, such as a card. The other is usually something memorized, such as a security code. The three most common categories are often described as something you know (the knowledge factor), something you have (the possession factor), and something you are (the inherence factor).

Kiteworks provides various modules to enable 2FA suited to enterprise needs.

RADIUS protocol

For enterprises that use 2FA using a centralized 2FA server, Kiteworks leverages the industry standard RADIUS protocol to integrate with the customer's 2FA solution. Kiteworks supports both simple single access request flow and access-challenge flow. Kiteworks passes the user's credentials, which include a 2FA code or one-time password (OTP), over the RADIUS protocol to the 2FA RADIUS server. The server's response determines if the user is authorized to sign in.

Email-based OTP

In email-based OTP, when user is signing in, Kiteworks generates an OTP and sends it the user's email address. The user supplies the OTP from their email to Kiteworks to complete the sign-in process. The length (the number of decimals) of the OTP, as well as the expiration time, can be specified by the administrator.

SMS-based OTP

SMS-based OTP works similarly to the email-based OTP. For SMS-based OTP, the OTP is sent using an SMS message to the user's mobile number. The first time a user signs in, they are required to provide their mobile number. When needed, the user's mobile number can also be modified by the user or administrator upon successful authentication. Kiteworks sends SMS messages using an SMS service solution procured by the customer, who may incur a cost. Kiteworks virtually supports all SMS service providers who provide web services for sending SMS. Administrators can compose a custom web service request message with rich options in the Kiteworks server Admin Console. Request message templates for several well-known SMS service providers are provided.

Time-based OTP (TOTP)

Kiteworks also supports the industry standard time-based OTP (TOTP) as specified in RFC 6238, <https://datatracker.ietf.org/doc/html/rfc6238>. This TOTP 2FA solution should work with all authenticator apps that support the TOTP algorithm standard defined in the RFC.

The following apps have been tested:

- Microsoft Authenticator
- Google Authenticator
- FreeOTP Authenticator (Red Hat)
- Twilio Authy
- Duo Mobile
- LastPass Authenticator
- OneLogin Protect

On the first login, users are required to provide the TOTP key generated by Kiteworks to their authenticator app, which can be done simply by scanning a TOTP QR code or by entering the key manually in the authenticator app.

Administrators may configure more than one 2FA source using these modules. Each 2FA source can then be assigned to a particular user profile.

Note: 2FA is limited to internal and LDAP authentication. The external SSO server (Kerberos or SAML IdP server) is required to fulfill all aspects of authentication and authorization which includes 2FA.

How Two-factor Authentication Works

1. The user enters their username and password in the Kiteworks application.
2. The Kiteworks application prompts the user for the 2FA code.
3. The user enters the 2FA code. Depending on the source, the code can be:
 - OTP sent via email or SMS.
 - Code generated by user's authenticator app using a TOTP algorithm.
 - Code provided by the enterprise 2FA solution.
4. For a 2FA code obtained via email, SMS, or TOTP authenticator app, Kiteworks internally validates the code.
5. For a 2FA code provided by the enterprise 2FA solution, Kiteworks passes the user's credentials and the 2FA code to the customer's RADIUS server for validation. If the RADIUS server returns an access challenge response, the challenge message is shown to the user and the user is prompted to enter a challenge response.
6. Depending on the result of 2FA code validation or the RADIUS server response, access to the user is granted or rejected.

Advantages to Two-factor Authentication

- Attackers cannot sign in, even if user's password is compromised.
- Leverages the existing enterprise 2FA solution.
- Eliminates the threat of compromise of an account in case a user password is leaked.

SFTP Certificate Authentication

SFTP provides an alternative method for client authentication, referred to as SFTP public key authentication. This method allows users to sign in to an SFTP service without entering a password and is often employed for automated file transfers.

A system that uses digital certificates for authentication to help establish secure communications is called a public-key infrastructure (PKI). A PKI consists of end entities, certification authorities (trusted parties who sign and issue certificates), and registration authorities (parties who handle the identification of end entities).

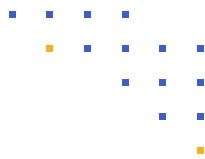
A PKI provides a means for reliable authentication of parties in an online environment by using asymmetric encryption. In addition to authentication, the PKI also enables secure digital communications and transactions.

Public keys are public and can be published on, for example, a web server. Public keys are used for validating signatures and encrypting messages. Before public-key operations can be made, the public key has to be received securely so that no one can substitute the genuine key with a tampered one. Certificates can be used for distributing public keys of end entities.

Certificates are digital documents that are used for secure authentication of communicating parties. Certificates are also used for sending the public keys of the entities to other entities. A certificate binds identity information about an entity to the entity's public key for a certain validity period. Certificates can be thought of as analogous to passports that guarantee the identity of their bearers.

Kiteworks supports certificate-based authentication for SFTP access.

Administrators can allow end users to upload or generate a public/private key pair for the certificate-based authentication. Once SFTP keys are created, the key can be added to the Kiteworks backend to create a list of keys to access the account using SFTP.



Kiteworks Key Management

All files on a Kiteworks system are encrypted in transit and at rest. Data in transit is secured via an SSL/TLS encrypted connection. Kiteworks uses disk level encryption as well as file level encryption to encrypt all the data on the disk.

Kiteworks Disk Encryption

The disk that Kiteworks creates for storing user files is an encrypted volume where the disk is encrypted with a unique key. This key is stored in an index that is associated with all of the encrypted keys on that storage volume. The encryption key for that is stored in the Kiteworks server database and also encrypted with a passphrase. The passphrase is used to encrypt and decrypt the passkey that encrypts all the keys for that entire storage volume. Kiteworks exposes that passphrase to administrators so they can have control over it, and can rotate it on demand per storage volume. Each storage role in the cluster has its own encryption keys, hence distributing the risk around the cluster.

The administrator has the ability to have many storage volumes all with different encryption keys. For example, if one of those disks were lost and if the system was split in such a way that the storage role was separate from the application role, it would be very difficult for anyone to make use of those files. Once the disk is shut off from the operating system and unmounted, you need the keys that are stored in Kiteworks database on a different system to decrypt those files.

Note: Each file is individually encrypted with a unique encryption key. For security purposes, Kiteworks deletes the key and the file from the file system using the standard file deletion protocol.

Kiteworks File Encryption

User files on the system are encrypted at the file level, where each file is encrypted with its own key. AES-256-CTR is used to encrypt each file. Each encrypted file is prepended with its metadata containing the encrypted file key. The CTR mode is used to support an efficient decrypt from any offset of the file. A file digest is kept separately in the database.

Some features of file key management include:

- Each volume (partition) gets its own set of keys.
- Keys are 32 bytes (256 bits) long and are composed of letters and digits.
- A super key is derived from a user provided or randomly-generated passphrase.
- A master key is generated randomly and encrypted with the super key.
- File keys are generated from a master key, using the PBKDF2HMAC function.

- File keys are encrypted/decrypted with the master key.
- The passphrase and master key are stored securely in encrypted form via a super key using AES-256-GCM.
- For generating salt and IV encryption, Kiteworks uses a cryptographically secure method to generate a random number. Salt is 16 bytes (128 bits) and IV is 12 bytes (96 bits).

Key Management Capabilities from HSM Integrations

A hardware security module (HSM) is a dedicated crypto processor designed for the protection of the crypto key lifecycle. Kiteworks offers integrations for HSM capabilities using Gemalto and AWS Key Management Service (AWS KMS). These integrations ensure safe handling and management of cryptographic keys. The local volume passphrase and the file encryption key encryption key (FEKEK) are encrypted once locally and a second time by the HSM and stored on the HSM.

These HSM integrations improve data storage security by introducing an additional layer of encryption because the key is stored in the network-attached HSM.

Ability to Disable/Enable Individual SSL Ciphers

In Kiteworks, an administrator can enable and disable individual Nginx SSL cipher suites. This is done through the Kiteworks server Admin Console.

Data Loss Prevention (DLP)

Overview

Data Loss Prevention (DLP) is designed to detect potential data breaches/data ex-filtration transmissions, and prevent them by monitoring, detecting and blocking sensitive data while in-use (endpoint actions), in-motion (network traffic), and at-rest (data storage). In data leakage incidents, sensitive data is disclosed to unauthorized personnel either by malicious intent or inadvertent mistakes. Such sensitive data can come in the form of private or company information, intellectual property (IP), financial or patient information, credit-card data, and other information depending on the business and the industry.

Kiteworks integrates with several market-leading DLP products, enabling content-aware sharing restrictions. Documents stored on-premise can be examined by any third-party DLP security suite that supports ICAP, a standard network protocol for inline content scanning. Sharing and access privileges can then be adjusted based on the results of the DLP scan and the customer's preferences for strictly controlling access.

Supported DLP Systems

Kiteworks integrates with the ICAP protocol. Because it relies on the ICAP standard for interaction with a customer's DLP server, the Kiteworks DLP integration will work with any ICAP-compliant solution and requires no changes to policies or servers in an existing security suite. ICAP-compliant solutions include:

- Symantec
- Fidelis
- Forcepoint
- Digital Guardian (Code Green Networks acquired by Digital Guardian)

Tying Kiteworks security policies to your existing DLP security suite means you can maintain a single point of policy management for data inspection and security alerts. If you already use one of the Kiteworks supported DLP systems for scanning outgoing email attachments or web traffic, you can point Kiteworks to the same server.

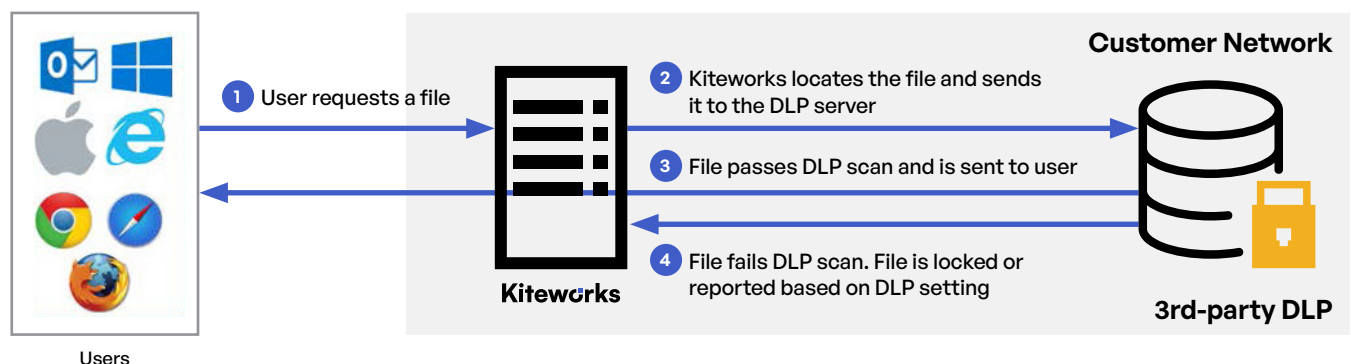


Figure 12. Graphical Overview of DLP Integration

Kiteworks has developed a flexible policy-based system that offers granular access and sharing controls, based on a new classification attribute that is associated with each file. The system uses the DLP scan results to classify every version of every file. There are three data classifications:

- **Scanned: OK.** Files that were scanned by a DLP system and passed OK.
- **Scanned: Blocked.** Files that were scanned by a DLP system and were found to contain sensitive data.
- **Unscanned:** Files that have not yet been scanned (in cases where files exist before DLP is configured, or when the external DLP system is unavailable or slow to respond).

Kiteworks then enforces different access and sharing restrictions for each data classification. For each of the three categories, the administrator chooses which actions to allow:

- Allow internal users to download and share the file.
- Allow third-party users to download and share the file.
- Allow anonymous users to download the file.

These settings constrain the normal permissions and sharing controls available to users as they interact with their data and collaborate with others. For instance, when sending someone a file, users could choose to block anonymous access even if the DLP settings would allow them to share it anonymously. But if they attempt to share a file in a way that the DLP settings prohibit, the system prevents them from doing so.

This flexibility allows you to control the trade-offs between security and usability as best fits your organization. If a document is flagged as sensitive, you could still allow sharing between employees but block sending to anyone outside your organization. Or you could take a stricter approach and block all users (even the owner of the file) from downloading or sharing the file with anyone. If you block downloads, an employee would not be able to access Kiteworks from an unmanaged device, get the file and share it by other means.

For any files that are not yet scanned, you can configure the same sets of constraints. This means Kiteworks could take an "innocent until proven guilty" or "guilty until proven innocent" approach according to your approach for impeding the flow of information.

When the Kiteworks controller sends files to the DLP system for scanning, it includes metadata indicating the owner of the file and the folder path where the file resides in Kiteworks. This allows the DLP server to log incidents and create notifications with enough detail to be actionable.

Data Sovereignty

The Kiteworks data sovereignty (file replication) is based on storage roles in one designated location being set to replicate files to another designated location with storage roles.

When you provision your Kiteworks cluster, it is possible to co-locate servers in different physical locations. The Kiteworks administrative interface similarly allows you to group your Kiteworks servers in specified locations that represent how your cluster is physically distributed.

When a storage role is added to a location, and a file is stored in that location, that storage role can no longer be moved to a different location due to data segregation and mapping rules.

Here's an example explaining file replication. If you have a London location and a Bangkok location, you can create rules to replicate all files in the London storage servers to Bangkok. As files are uploaded to London, the storage role in London replicates those files to the Bangkok location. Similarly, other storage locations will replicate files if the replication rules have been configured. After files have been replicated, the administrator can configure the system so that when a user wants to download a file, the file can be downloaded from the closest location to the user's browser or mobile device.

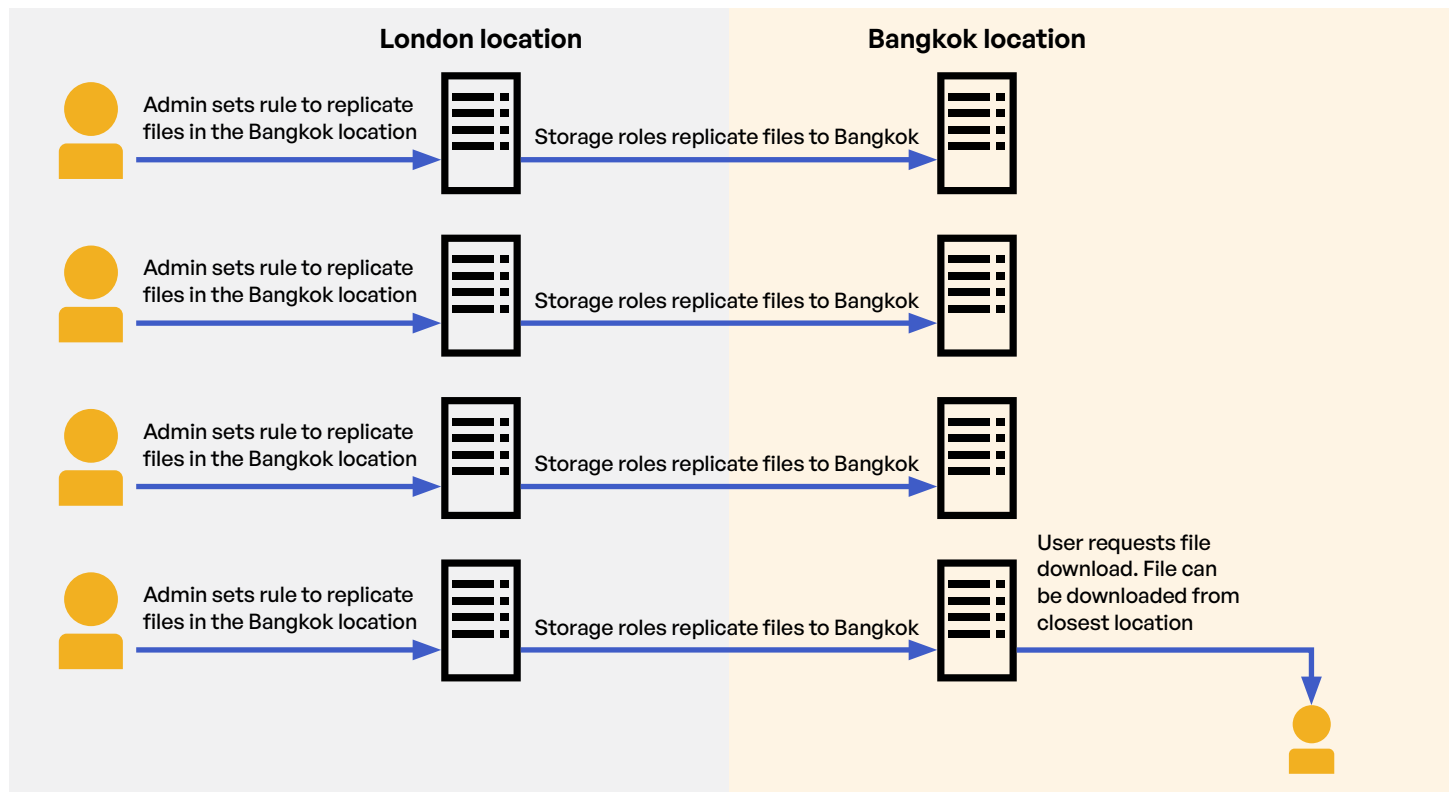


Figure 13. Data Sovereignty (File Replication)

Secure Container and Remote Wipe

Secure Container

Secure container is a separate encrypted volume in a user's local storage for storing everything that is downloaded from the Kiteworks server by the Kiteworks for Desktop sync client.

- On Windows computers, the volume is stored in Virtual Hard Disk format (.vhd file) and is encrypted by Microsoft BitLocker and locked using a password.
- On Mac computers, the volume is stored as a Sparse Bundle Disk Image (.sparsebundle file) and locked using a password.

The Kiteworks for Desktop sync client allows users to easily download corporate data to laptops or home computers. The IT department needs a way to remove access to corporate data in the case of a lost or stolen device, or if an employee leaves the company. Secure container and remote wipe functionality enable the removal of access to corporate data.

A secure container enables protection of intellectual property, allowing access and collaboration of enterprise file synchronization and sharing (EFSS) by supporting remote wipe.

Enterprises performing secure EFSS need the ability to protect their intellectual property while allowing access and collaboration. This enables the system administrator to remove sensitive content from the user's device when necessary. By enabling this option in Kiteworks, use of the secure container for syncing is enforced.

BitLocker

Microsoft BitLocker is required for the Kiteworks secure container in the Kiteworks for Desktop sync client installation. However, BitLocker is not available for virtual environments.

Source: <https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-overview>

BitLocker Encryption

Microsoft BitLocker is a full disk encryption feature included with Windows designed to protect data by providing encryption for entire volumes. By default, it uses the AES encryption algorithm in cipher block chaining (CBC) or XTS mode with a 128-bit or 256-bit key.

New BitLocker Features in Windows 10, Version 1511

BitLocker now supports the XTS-AES encryption algorithm. XTS-AES provides additional protection from a class of attacks on encryption that rely on manipulating cipher text to cause predictable changes in plain text. BitLocker supports both 128-bit and 256-bit XTS-AES keys. It provides the following benefits:

- The algorithm is FIPS-compliant.
- It's easy to administer. The BitLocker Wizard can be used to manage-bde, Group Policy, MDM policy, Windows PowerShell, or WMI to manage it on devices in an organization. Source: <https://docs.microsoft.com/en-us/windows/whats-new/whats-new-windows-10-version-1507-and-1511#bitlocker>

Password Generation Mechanisms

Windows

- Password is generated as random GUID using a Microsoft API.
- Password is encrypted using Microsoft's ProtectedData API.
- Encrypted password is encoded in Base64.
- Password is stored in the computer registry.

Location: HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Accellion\<secure_container_regkey>

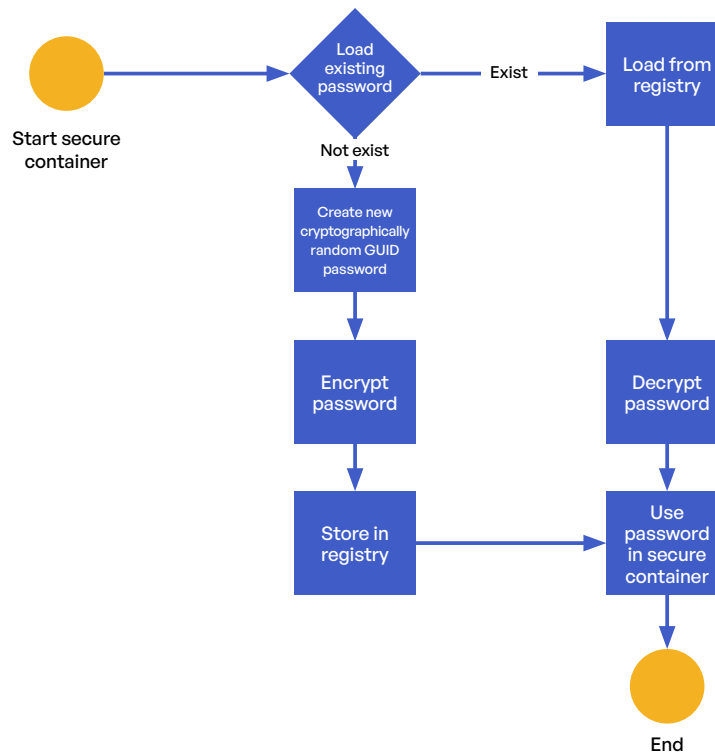


Figure 14. Secure Container Password Generation and Storage in Windows

Mac

- Password is generated using Apple's cryptographically secure SecRandomCopyBytes API.
- Keychain access.
 - Password is stored in Keychain as bytes, not string.
 - The Keychain Access application will not be able to display the value. Even though a user clicks the "Show Password" link, the value will not be displayed. The stored bytes value will only be accessible by the application programmatically.

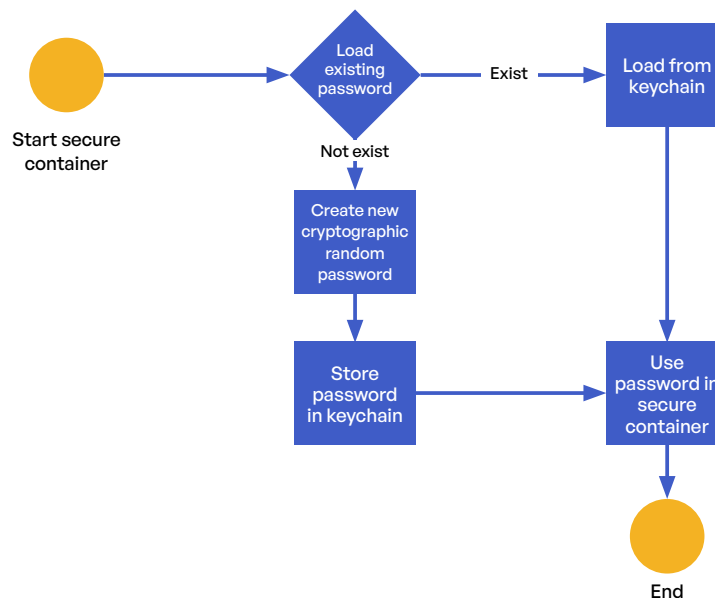


Figure 15. Secure Container Password Generation and Storage in macOS

Remote Wipe

A remote wipe can be used as a security measure to wipe out account specific data or downloads from a device. Reasons for performing remote wipe can be that the device was stolen, is to be decommissioned, is to be reassigned to another user. A remote wipe can be performed once the secure container is configured in Kiteworks.

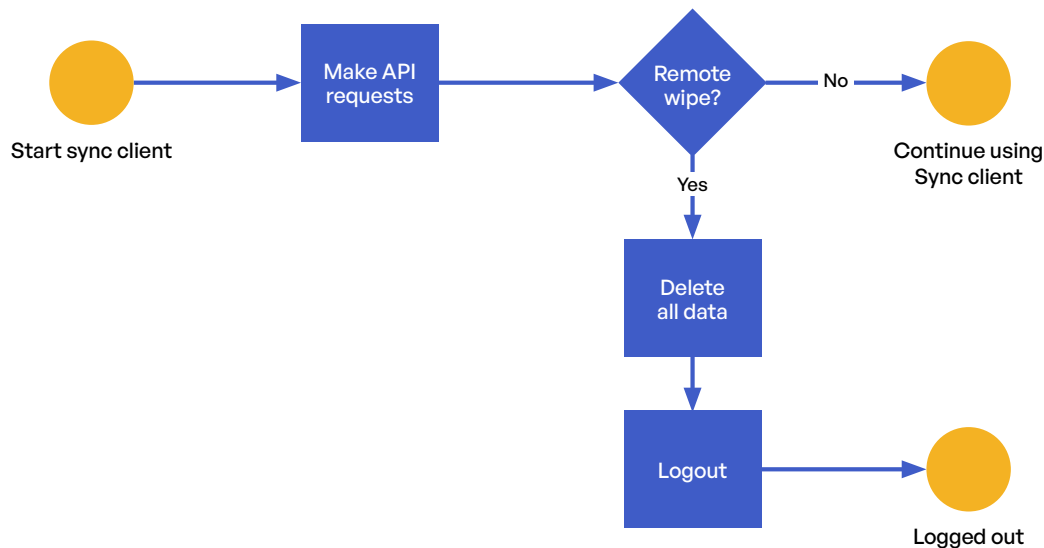


Figure 16. Kiteworks for Desktop - Remote Wipe Flow

Compliance and Auditing

For compliance and auditing purposes, all end user and administrative activities are recorded. Kiteworks provides reporting and analytics required for HIPAA, SOX and other standards and regulations. Kiteworks supports SIEM integration via the syslog.

Network Security

Kiteworks provides features to help ensure and maintain a secure network.

Secure Inter-node Communication (INC)

A Kiteworks cluster offers a secure and hardened way for nodes to interact. All API calls between nodes are authenticated using JSON Web Tokens (JWTs) and private/public keys. All inter-node communication is validated using signed JWT. This provides a defense in depth against replay and man-in-the-middle (MiTM) attacks on the inter-node communication.

NFS Support

NFS support is available to Kiteworks administrators so that they can add storage that is not part of the Kiteworks cluster. This lets them increase data storage easily. Kiteworks uses file level encryption when writing files to an NFS volume. More specifically, files going into NFS via Chunk Upload API (including temporary files) are encrypted by file level encryption. When the files are encrypted, they are stored prepended with the encryption metadata in the file header. Files are encrypted using AES-256 with Counter (CTR) mode of operation (AES-256-CTR). CTR mode provides efficient decryption from any offset of the file.

Network Activity Data Collected with Splunk UF

Universal Forwarders (UF) provide reliable, secure data collection from remote sources, and forward that network data into Splunk for indexing and consolidation. UFs can scale to tens of thousands of remote systems collecting terabytes of data.

Kiteworks offers integration with the Splunk Universal Forwarder, which can be enabled in the Kiteworks server Admin Console. The administrator can also enable SSL to ensure security of logs in transit. As part of this integration, the Kiteworks Splunk CISO Dashboard is provided as a network security management tool for Chief Information Security Officers (CISOs). The dashboard provides a global snapshot of resources, content, activities of content, and performance within an organization using the platform's extensive syslog. You can drill-down to view the lowest level details of the activities. The CISO Dashboard is enabled in the Kiteworks Governance and Compliance platform for all single-tenant enterprise deployments.

Kiteworks Security for Android and iOS Mobile Devices

Introduction

This section provides the technical details of Kiteworks security for Android and iOS mobile devices.

Authentication

The following processes authenticate the user on Android and iOS mobile devices.

Login

OAuth authentication is used to sign in to Kiteworks from the mobile application. The application requires the user to enter a username and password in Web view to log in. After successful authentication, the application retrieves the access token from the server which is used for further communications with the server.

The duration of the validity of the access token lifetime is configurable on the Kiteworks server. If the access token expires, the user is required to sign in to the application again. The access token and the other policies configurable by the administrator are stored in the Android and iOS user preferences.

Offline Biometric Authentication

User are asked to enable biometric (facial or fingerprint recognition) authentication if they want to access the mobile application in offline mode and don't want to enter password every time. If not set up, users will not be able to access the application in offline mode and will need to re-enter their credentials when the app times-out.

Users registered for biometric authentication don't need to sign in again during the lifetime of the access token. Kiteworks mobile application uses native device biometry authentication APIs. Each active session is valid for 3 minutes. If the application moves to the background on the device, the user is asked again to provide biometry verification for subsequent access to the application.

Administrators can configure Kiteworks to require that users sign in with credentials every time after time-out.

Network Communication

The Kiteworks Android and iOS mobile applications use REST APIs to communicate with the Kiteworks server. All communications including downloads and uploads are done over SSL.

Encryption

Encryption Key

All files downloaded to a mobile device are stored in device storage after encryption. The AES-256 encryption algorithm is used to encrypt the files. For iOS, all files are stored in the App Sandbox after encryption.

The data encryption key is sent to the server and is also stored in the local device using the Keychain/Keystore wrapper for the purpose of accessing files offline. This data encryption key is used to encrypt and decrypt files that have been downloaded to all devices. The same key is used to encrypt the database storing file metadata information.

Storage of Data Encryption Key

A user's data encryption key is stored in the Keychain/Keystore wrapper.

Remote Wipe

Remote wipe is a feature that allows the administrator to wipe all the downloaded data on a device. The remote wipe call can be issued by the administrator to a specific user on a specific device. When the application determines that remote wipe has been issued (typically when the user does an action in the application which requires communicating with the server), all the downloaded files cached on the device get deleted from the server and the data encryption key gets deleted from the mobile preferences. The user is required to sign in again. After signing in again, the data encryption key is reset and stored once again on the device.

Sharing Data with External Applications

Files downloaded to the device are editable using third-party Android and iOS applications installed on the device. Files are opened from within the Kiteworks mobile application using the "Open in" option. When a file is being edited in a third-party application, the file is decrypted temporarily and shared with the other application. Once the editing is completed, the file is saved back to Kiteworks and encrypted.

Only applications in the "Allow List" on the Kiteworks server are allowed to be used to view and edit files.

Resources

For support on technical issues, please visit the [Kiteworks Support Portal](#).