Whitepaper

Okta FastPass Technical Whitepaper

Contents

What is Okta FastPass

Okta FastPass is a Zero Trust authentication solution, designed for defense in depth. FastPass, when used in conjunction with a FIPS 140-3 Level 2 compliant device, satisfies Revision 4 of NIST Special Publication 800-63 Digital Identity Guidelines for Authentication Assurance Level 3 (AAL3), providing the highest level of authentication assurance.

By securing at the first point of authentication and then along the lifespan of the active single sign-on (SSO) session – FastPass can mitigate the impact of phishing attacks, session theft, and unauthorized local activity. It does so by enabling passwordless, cryptographically secure access – only to trusted applications – with an intuitive user experience consistent across major platforms and devices, managed or unmanaged. With silent context evaluations of browsers and devices at every app login, if enabled, together with signals from your broader security solution ecosystem, FastPass strengthens the Zero Trust security of your organization.

To use FastPass, end users must have the most recent version of the Okta Verify authenticator app available on their desktops, laptops, or mobile devices. Okta Verify with FastPass support is available for iOS, Android, Windows, and macOS platforms for end users to sign in to Okta-protected OIDC, SAML, or Web Services Federation applications.

Okta Verify and FastPass work together to provide end users with one of the safest ways to log in. Okta Verify is the application, while FastPass is one of three authenticators supported by Okta Verify, the other two being push notifications and one-time passwords (OTP), which are only supported on the mobile versions of Okta Verify (i.e., Okta Verify for Android and iOS) and are out of the scope of this whitepaper. Admins can set up FastPass to not only be a phishing-resistant authenticator, but also a solution to collect and evaluate device posture as part of the authentication process. As the application that enables FastPass behavior, customers must purchase a service that incorporates the Okta Verify service, and users must enroll in Okta Verify to use FastPass.

Benefits of Okta FastPass

Benefits to admins

Phishing resistance

FastPass can prevent the most common phishing attacks for managed and unmanaged devices on all supported platforms when required by policy. When used to sign into an application, FastPass validates whether the origin header of the authentication request, which may be coming from a malicious webpage, matches the site where the user is trying to gain access. This ensures that a user isn't tricked into providing credentials to a malicious site posing as legitimate. Administrators can enable phishing-resistant constraints in the authentication rules for applications they want to protect. FastPass, as a phishing-resistant authenticator, aligns with NIST 800-63-3 AAL2 and AAL3; as well as the draft guidance around NIST 800-63-4.

Elevated security posture

FastPass uses public key cryptography to authenticate the user, eliminating the need for passwords (and the risk of common attack vectors leveraging passwords). When an end user enrolls in FastPass, public and private keys are generated on the device. The private keys are stored securely on the device (in a hardware module if available), whereas the public keys are sent to Okta’s cloud service. When challenged by Okta’s cloud service, the private key is used to sign a one-time nonce, which is then returned to Okta. Okta’s cloud service then validates the digital signature before access is granted to the user.

Rich device context

FastPass verifies the device and browser in use during authentication, collecting signals from first-party and third-party sources for more informed authentication and authorization decision-making. Device context is evaluated when a user first logs in, and silently, each time the user opens a new application – providing additional assurance that the device hasn’t changed before allowing access to downstream resources. Should the device-in-use fail a device posture policy condition, users are empowered to correct any failed device or browser security checks with self-remediation instructions.

Signals can come from several sources, including device assurance policies to assess device health and device management solutions to verify management attestation. Device context can also include signals from endpoint security integrations. FastPass can collect information on a device's security posture (e.g., risk score) from Unified Endpoint Management (UEM) and Endpoint Detection and Response (EDR) solutions.

Interoperability

FastPass authenticates inline with any Okta-protected OIDC, SAML, and WS-Federation app. As the identity provider (IdP), Okta handles FastPass authentication from the end user's device without any changes to the service provider (SP) end. FastPass can also integrate with device built-in authenticators such as Windows Hello, Touch ID, or Face ID to support biometric authentication and complete multi-factor authentication (MFA).

FastPass is both an authenticator and a solution for device posture evaluation. It can be combined with another authenticator to meet MFA requirements, or in cases where an admin prefers users to use another authenticator, such as a FIDO2 security key, FastPass can still be deployed in combination with the other authenticator to bring rich device signals to the authentication flow.

Device lifecycle management

The Devices page under Universal Directory provides basic device information once the device has enrolled in Okta Verify. This includes security signals, device management state, and device identifiers. Through the Okta Admin Console, administrators can manage the lifecycle of devices with options to remotely suspend/unsuspend and activate/deactivate devices as they see fit. Such actions are also supported via APIs, allowing for more custom workflows, such as temporarily restricting access from non-compliant devices.

Benefits to end users

Passwordless login experience for increased productivity

End users can enjoy passwordless authentication to all FastPass-protected resources. This significantly improves the employee experience by reducing the friction introduced by passwords (and password resets) and out-of-band factors such as push, time-based one-time passwords (TOTP), and SMS. And with seamless support of platform authenticators for biometric authentication, users can satisfy higher security assurance requirements with little to no additional friction.

Consistent user experience across platforms

FastPass provides a user-friendly and consistent authentication experience regardless if the user is on a managed or unmanaged device, or using an iOS, Android, Windows, or macOS machine. Windows support includes virtual desktop infrastructure (VDI) environments, including Windows 365, Citrix, and AWS WorkSpaces. End users benefit from having the same secure, passwordless experience across all their devices.

Key Concepts

Device registration vs. device management

A device is considered "registered" when it is enrolled in FastPass; this is different from a device that's considered "managed," which requires the device to be controlled or managed by a mobile device management (MDM) solution or equivalent. Through FastPass, admins cannot read or access personal data on the device, remotely wipe the device, or track an exact location, which may be possible for a device managed by an MDM.

Device identity and enrollment

FastPass is one of several authenticators supported by the Okta Verify application, and users must enroll in Okta Verify to use FastPass.

When a user enrolls in the Okta Verify app, a unique device identity is created in Okta's Universal Directory. User and device association is made. The device itself is assigned a Device ID. Universal Directory securely stores additional context about the device, such as the device's display name, OS, model, manufacturer, management status, etc. The device details are updated each time users authenticate successfully with FastPass. For a complete list of device details, please visit the product documentation. An administrator can search for a device in the Okta Admin Console and take action on it: suspend, unsuspend, deactivate, reactivate, and delete. More information about different lifecycle states can be found here.

Phishing resistance

The National Institute of Standards and Technology (NIST) provides technical guidelines to organizations for the implementation of digital identity services. In their Special Publication 800-63B, NIST defines some key phishing resistance attributes, which include:

- **Verifier name binding** to establish an authenticated protected channel with the verifier and generate an authenticator output that is cryptographically bound to a verifier identifier (e.g., origin domain)
- **Replay resistance** via authenticators such as OTP devices, cryptographic authenticators, and look-up secrets, as well as protocols that use nonces or challenges and timeliness data
- **Verifier-compromise resistance** through authentication protocols that do not require persistently stored secrets, such as using a cryptographic authenticator and ensuring that any stored public keys are associated with the use of approved cryptographic algorithms
- **Authentication intent** that requires the user to respond to each authentication or re-authentication request explicitly

Proof of possession factor

This factor type satisfies possession requirements (“something you have”). When the user enrolls in FastPass, Okta generates a key pair and designates it as the FastPass proof of possession key pair. The private key is stored in the hardware key store of the device, if available, or otherwise in a non-exportable software key store supported by the specific operating system. The public key is sent to the Okta server. During the authentication flow, the Okta server uses this public key to verify that the signature of the payload was signed by the corresponding private key.

If the signature verification is successful, the user is considered to have provided a possession factor. If the authentication policy for the application requires additional factors, the Okta server challenges the user for another authentication factor.

FastPass, as a proof of possession factor, can be configured to be collected with or without checking for user presence. If user presence is required, Okta Verify will ensure it gets a user interaction (e.g., “Yes, it's me” prompt) before proceeding with the verification.

User verification factor

This factor type satisfies user verification by providing an inherence factor (e.g., biometrics) or a knowledge factor (e.g., a PIN or passcode) as the second factor on top of the proof of possession that FastPass satisfies.

During Okta Verify enrollment, additional key pairs are generated when the end user provides their biometrics or device passcode for user verification. If available, the private key is stored in the device’s hardware key store; the public key is sent to the Okta server. The private key can only be used to sign a payload after the end user provides their biometrics or passcode to satisfy user verification requirements.

Device context

During the authentication flow, FastPass collects context related to the device in use. This includes essential device signals such as platform name, OS version, device display name, etc. Then, based on what the admin configures as part of the authentication policy, FastPass can also collect signals such as management attestation, jailbreak status, and other security signals needed for device compliance.

Device probing

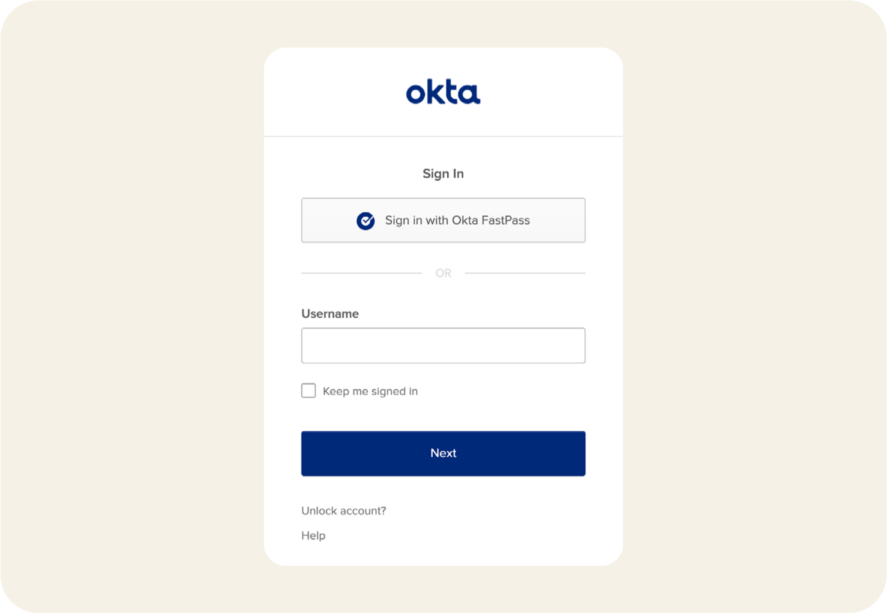
Probing is a mechanism by which Okta's Sign-in Widget (SIW), running in a browser tab or webview, communicates with Okta Verify on the device. If Okta Verify is not installed, the probing mechanism fails, and the user is prompted to use other authenticators.

Silent versus interactive probing

FastPass supports silent probing and interactive probing methods for every browser. Silent probing requires no user interaction and provides the best user experience. As a result, FastPass always attempts to do silent probing first.

Method Name	Method Type	Supported Platforms
Loopback	Silent, Interactive	macOS, Windows, Android, and iOS
Credential SSO Extension	Silent, Interactive	Managed iOS and macOS
Universal Link	Interactive	iOS
App Link	Interactive	Android
Custom URI	Interactive	Windows and macOS

The SIW falls back to an interactive probing method if a silent probing method is unavailable, such as if the loopback server fails to start. In these cases, end users must launch Okta Verify using an interactive method. Users will be prompted to click or tap on a "Sign in with Okta FastPass" button on the SIW.



Irrespective of which probing method is used, the SIW attempts to collect either:

- Proof of possession factor only
- Proof of possession factor and user verification factor

The decision to collect one factor or both factors is dependent on both Okta Verify configurations and the application authentication policy configurations.

When the "Sign-in with Okta FastPass" button is turned on, it provides an easy way for users to authenticate without entering either a username or password. It also helps users to onboard to Okta Verify and FastPass if they still need to enroll, providing setup instructions and prompts.

Device probing schemes

Probing schemes, when successful, bind the device to the session. FastPass supports the following probing schemes:

- **Loopback:** Okta Verify runs a server on a local host port that can respond to probing requests from the SIW. When reached from the SIW, the loopback server can accept the challenge to digitally sign a nonce. This probing scheme is phishing-resistant and available on Windows, macOS, Android, and iOS.
- **Credential SSO Extension**: Users on managed iOS and macOS devices can have a seamless FastPass experience with an SSO extension deployed to devices by an MDM solution. This probing scheme is phishing-resistant for supported apps and browsers.
- **Universal Link**: On iOS devices, universal links allow the SIW to launch the Okta Verify app with a user click. The SIW appends the challenge request to the universal link so that when the user clicks on “Sign In with FastPass,” the challenge request is available to FastPass. Universal links can be phishing-resistant when combined with loopback, which Okta attempts to do automatically whenever possible.
- **App Link**: On Android devices, app links allow the SIW to launch the Okta Verify app with a user click. The SIW appends the challenge request to the app link so that when the user clicks on “Sign In with FastPass,” the challenge request is available to FastPass. App links will not always work in native application authentication flows based on Okta’s testing. App links can be phishing-resistant when combined with loopback, which Okta attempts to do automatically whenever possible.
- **Custom URI**: On macOS and Windows devices, the custom URI allows the SIW to launch the Okta Verify app with a user click. Similar to the universal link and app link schemes, the Okta Verify app responds to the custom URI scheme when clicked. The SIW appends the challenge request to the custom URI. Custom URI is not phishing-resistant but can be used to start up the application's loopback server.

Security Model

Key generation and protection

A user's enrollment in Okta Verify on a device may generate two key pair types — proof of possession key pair and user verification key pair. By default, the private keys are stored in a device's hardware key store, if available, such as a TPM or a Secure Enclave (for iOS). The private keys never leave the hardware key store and cannot be backed up or exported to other devices. If the device does not have a hardware key store, the private keys are stored in a non-exportable software key store available with the platform or operating system.

During factor collection, FastPass uses the key store to get the digitally signed output for a given input payload. This signed output payload is sent to the Okta server for signature verification via a pinned TLS connection, which intermediate parties can't intercept.

Administrators can build authentication policies to require hardware-bound keys. See "Configure an authentication policy" for more details.

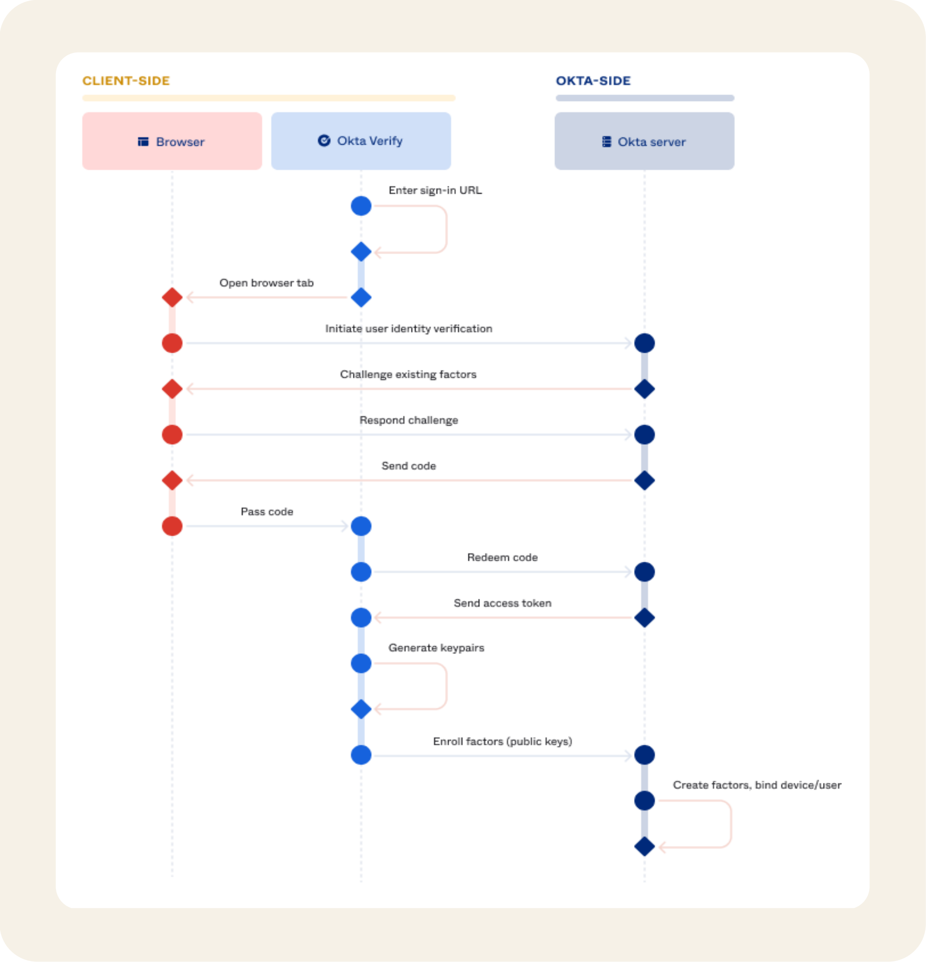
Platform	Digital Signature Algorithm Used	Key Size
macOS	ES256	256 bit
Windows	RS256	2048 bit
iOS	ES256	256 bit
Android	RS256	2048 bit

Tamper protection

The Okta Verify application is hardened with tamper protections to ensure trusted behavior from the app. Tamper protections for Okta Verify will detect if malware or some other attacker is trying to modify the intended behavior of the app so that in the event of an attack, Okta Verify will immediately crash to disable all processes.

Okta FastPass In Depth

Enrollment



When the end user enrolls in FastPass, the process ensures a strong binding between the authenticator, device, and user. Only a single FastPass enrollment is allowed per device and user account.

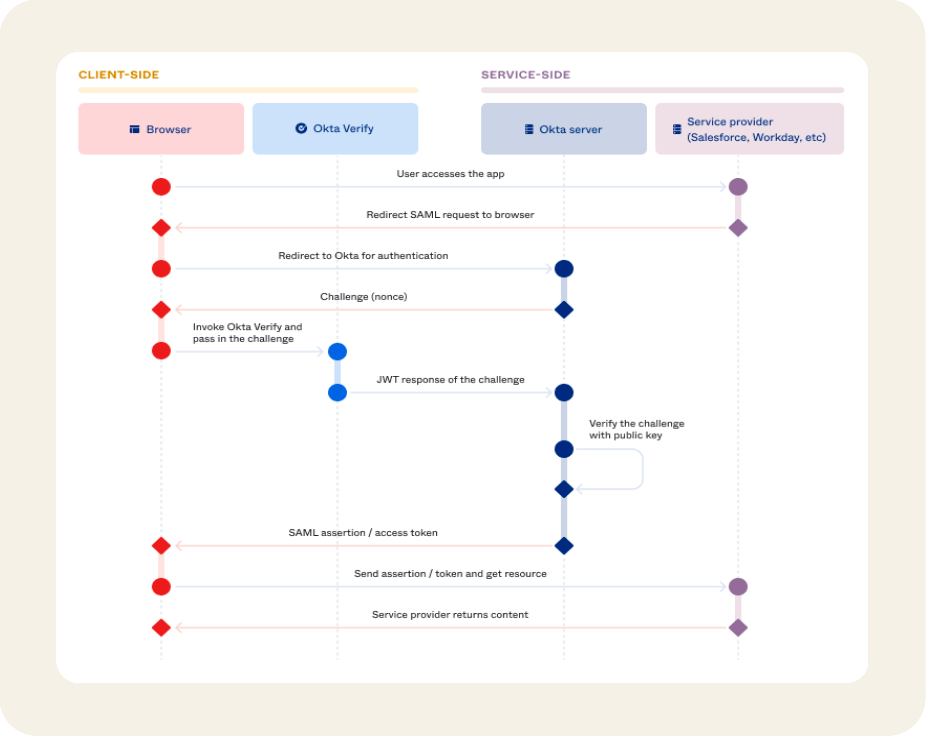
Before enrolling in FastPass, the user must first provide other factors according to the authenticator enrollment policy for identity verification. Once the identity is verified, Okta Verify prompts the user to enable biometric authentication (e.g., Touch ID, Face ID, Windows Hello) or a device passcode. Okta Verify generates a key pair for user verification if enabled.

For a phishing-resistant enrollment process, admins can require proof of a phishing-resistant authenticator as part of the FastPass enrollment flow.

End users who want to enroll a second device in Okta Verify can do so in a phishing-resistant manner by Bluetooth (for Android, iOS, macOS, or Windows) or via YubiKeys.

Authentication flow and probing

The following diagram illustrates a typical FastPass flow for a SAML-based single sign-on (SSO) process, as an example.


CLIENT-SIDE
SERVICE-SIDE
Browser
Okta Verify
Okta server
Service provider (Salesforce, Workday, etc)
User accesses the app
Redirect SAML request to browser
Redirect to Okta for authentication
Challenge (nonce)
Invoke Okta Verify and pass in the challenge
JWT response of the challenge
Verify the challenge with public key
SAML assertion / access token
Send assertion / token and get resource
Service provider returns content

FastPass in a typical SSO flow using a SAML request as an example

1. The end user accesses a service such as Salesforce or Workday from a browser.

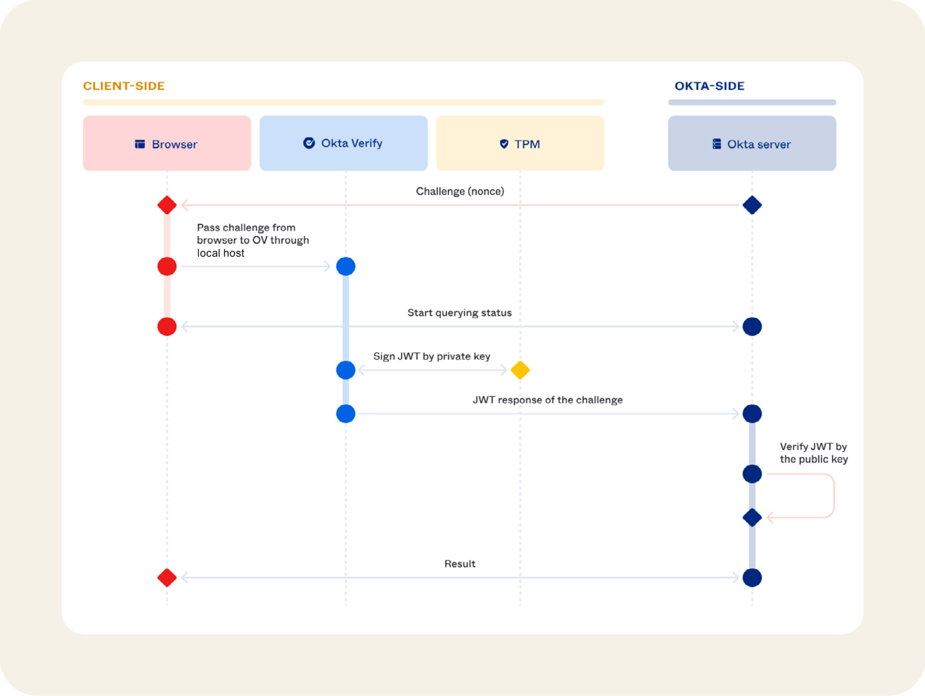
2. The service provider generates the SAML request and redirects the request back to the browser and then to Okta for authentication.

3. The Okta server evaluates the request and generates the FastPass challenge, which may include demands for user verification and specific device conditions as configured in the app authentication policy. The Okta server sends the challenge along with the SIW to the browser. The SIW on the browser invokes FastPass.

4. FastPass generates a response with the appropriate device signals and signs the response with the user verification private key (enrolled by the user on the client side). The user verification private key can attest to both proof of possession and user verification.
5. The Okta server identifies the user by checking the signature with the public key. The device information collected is evaluated against the relevant authentication policy.
6. If the conditions match access requirements, the Okta server generates the SAML assertion and redirects the browser back to the service provider. Otherwise, the Okta server will send a challenge for another factor or block access altogether based on the result of policy evaluation.

Factor collection when using loopback

A method the SIW uses to communicate to the local Okta Verify install is a local server hosted by Okta Verify that is inaccessible by the broader internet. This allows for rich, device-local communication between the browser session and the local app install. This server enables Okta Verify to remain in the background during the authentication flow, only surfacing itself as required by the Okta server to perform actions such as collecting biometrics or getting user consent.



Factor collection when using the credential SSO extension

The credential SSO extension is for managed macOS/iOS only. Okta Verify is configured to monitor and intercept HTTP traffic between the browser and Okta server. When the Okta server initiates a challenge, the challenge is passed from the browser to FastPass through the SSO extension with a 401 response. When a 401 status is detected, FastPass initiates the signed nonce challenge and response flow.

Factor collection when using custom URI and universal link

When loopback or credential SSO fails, browsers can launch and pass challenges to FastPass through deep links. For Windows and macOS, we use the custom URI scheme; for Android, we use app links; for iOS, we use universal links. App links and universal links are secure in that only verified apps can invoke them, but they are not supported on all platforms.

Device assurance policies and context re-evaluation

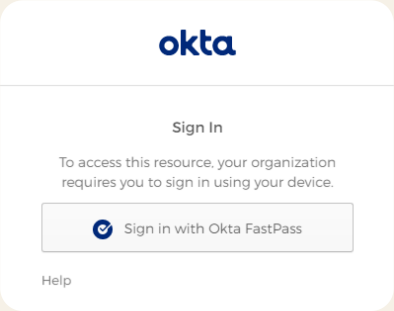
FastPass collects first-party device posture signals through device assurance policies. With device assurance policies, admins can configure sets of security-related device attributes to be checked as part of an authentication policy. For example, a device assurance policy can be used to ensure a specific operating system version or security patch is installed before that device can be used to access Okta-protected resources. With such device checks, minimum requirements are established for devices with access to sensitive systems and applications. If a user is not in compliance with a required device attribute, the Okta SIW provides remediation instructions.

When a user authenticates with FastPass, all device signals, including third-party signals such as those from endpoint security integrations, are pulled and assessed. This is done not only when an SSO session is first established, but the device signals are re-evaluated each time a new application is opened from the Okta Dashboard and when re-authentication is required. These silent context checks can help facilitate the continued security of the devices in use and mitigate the risk of session hijacking by detecting a potential attack and blocking access to downstream apps.

Remediation

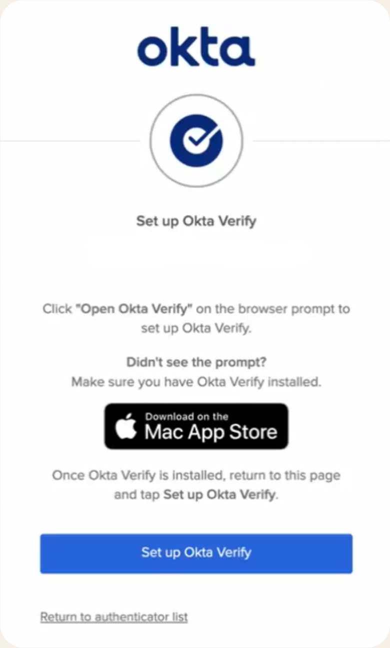
The SIW provides remediation hints to end users when device conditions fail:

- Example #1: When FastPass is required to access an application



To access this resource, your organization requires you to sign in using your device.

- Example #2: When a registered device is required for access, and Okta Verify is not installed

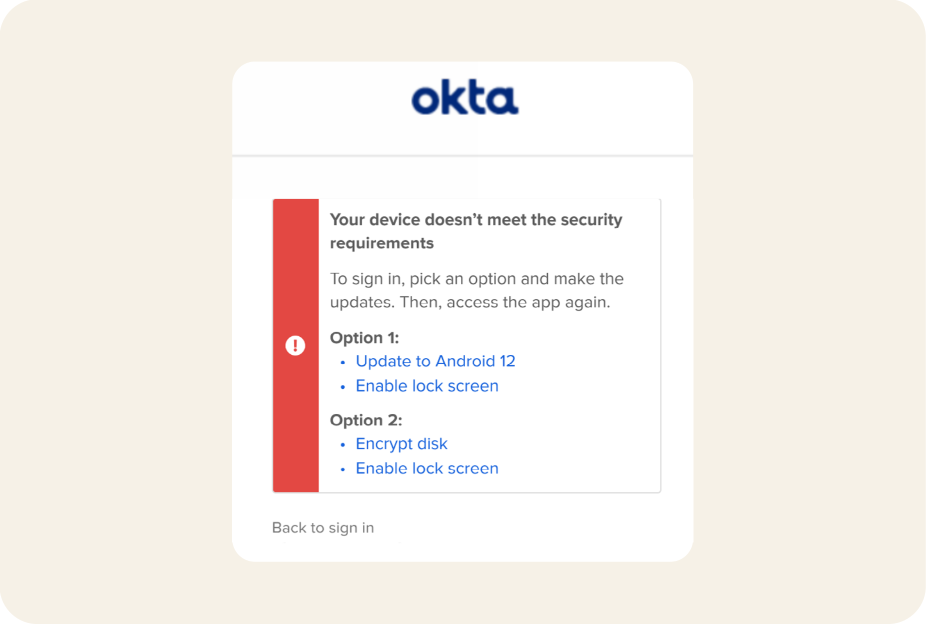


Click **"Open Okta Verify"** on the browser prompt to set up Okta Verify.

Didn't see the prompt?
Make sure you have Okta Verify installed.

Once Okta Verify is installed, return to this page and tap **Set up Okta Verify**.

- Example #3: When a device assurance policy condition fails, such as a requirement for the latest OS version



Your device doesn't meet the security requirements

To sign in, pick an option and make the updates. Then, access the app again.

Option 1:
- Update to Android 12
- Enable lock screen

Option 2:
- Encrypt disk
- Enable lock screen

Managed devices

Okta integrates with device management software providers to ensure that devices are properly managed before end users can access apps from those devices. During the authentication flow, FastPass collects management signals, which are shared with the Okta server for management attestation verification. You can read the steps for configuring and deploying managed devices here.

On desktop (Windows and macOS) devices, the client certificate issued by the Okta Certificate Authority (CA) or your own CA is used to create the management attestation signal. The Okta CA uses the SCEP protocol to deploy client certificates to managed desktop devices. The Okta CA provides static, dynamic, or delegated modes of SCEP certificate deployments. For Windows, Okta recommends that the administrators configure the MDM SCEP policy so that private keys are stored in the device hardware key stores and certificates are non-exportable.

If the app authentication policy requires the managed device condition, the Okta server requests the device management attestation through the FastPass protocol. The Okta Verify client on Windows or macOS identifies the correct client certificate deployed on the device and uses it to sign a unique nonce in the request to create management attestation. The Okta server first validates if the client certificate is issued by the known CA, then validates the management attestation signature using the public key of the client certificate.

In third-party CA-based deployments, the Okta server periodically (every 6 hours) checks the third-party certificate revocation list (CRL) and invalidates non-active client certificates. This limits the ability of the revoked, suspended, or on-hold client certificates to satisfy management attestation.

The Okta server associates the client certificate to the device object in Universal Directory. If replayed later from another device, the management attestation fails. This prevents misusing client certificates from unauthorized devices to provide management attestation.

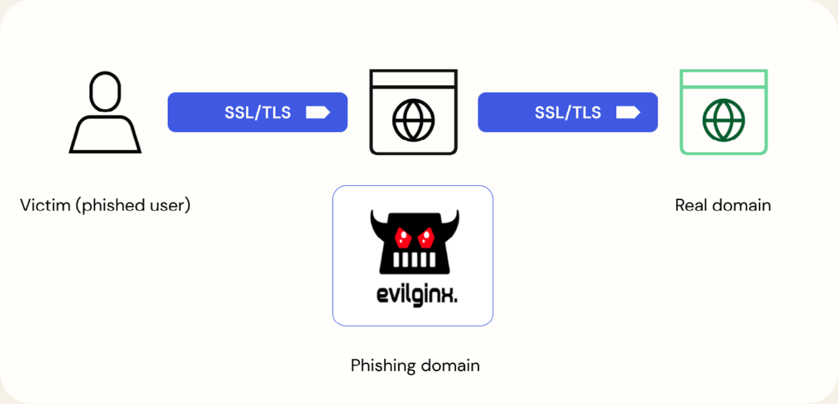
On mobile (iOS and Android) devices, a management hint (shared secret) is deployed to the device through a managed app configuration. Similar to certificate-based deployments, Okta Verify responds with the shared secret when challenged by the Okta server through the FastPass protocol. The Okta server verifies the secret by comparing it to a hash that was stored during the initial configuration. The Okta server does not store the raw secret, so you must securely store this yourself.

Phishing resistance

Phishing is a type of social engineering attack often used to steal user credentials in order to impersonate the user and gain access to data. The attack dupes a victim into opening a link to an illegitimate site masquerading as a trusted entity, and providing credentials to the attacker. As an advanced phishing-resistant authenticator, FastPass can prevent one of the most common types of phishing attacks, Adversary-in-the-Middle (AiTM) attacks.

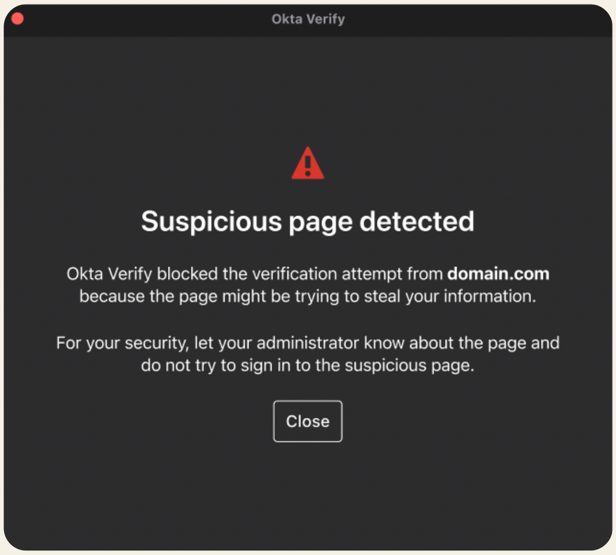
AiTM Attack Framework with Evilginx

Adversary-in-the-Middle attack framework used for phishing login credentials and sessions



In an AiTM attack, the bad actor uses a proxy to initiate the Okta SIW in order to masquerade as a trusted entity. This requires a proxy request from the malicious site to the Okta server. The server can validate the origin header and detect any mismatches. When a domain mismatch is detected, FastPass authentication fails, the event is logged in Okta SysLog, and the user is shown a suspicious activity warning.

In an AiTM attack, the bad actor uses a proxy to initiate the Okta SIW in order to masquerade as a trusted entity. This requires a proxy request from the malicious site to the Okta server. The server can validate the origin header and detect any mismatches. When a domain mismatch is detected, FastPass authentication fails, the event is logged in Okta SysLog, and the user is shown a suspicious activity warning.



Suspicious page detected

Okta Verify blocked the verification attempt from **domain.com** because the page might be trying to steal your information.

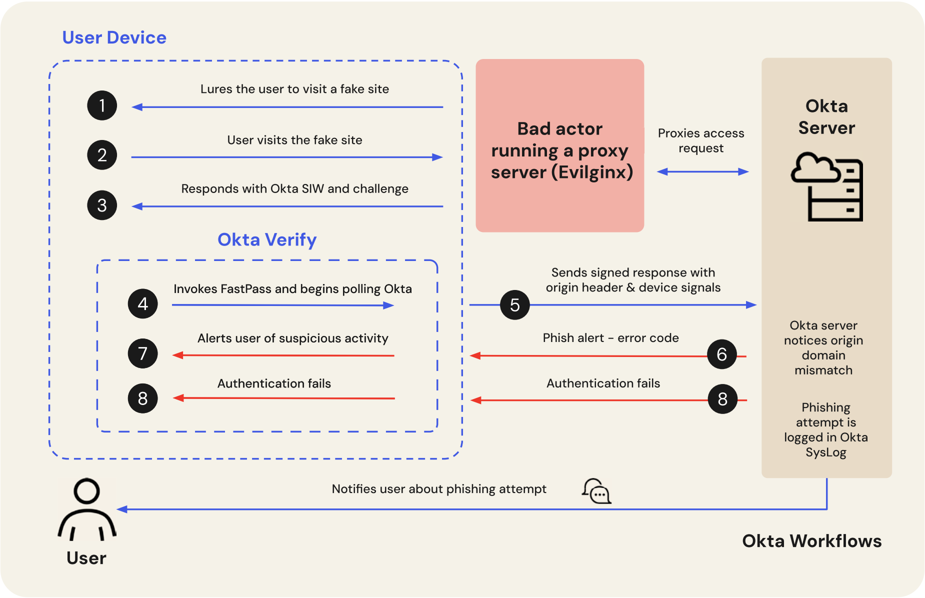
For your security, let your administrator know about the page and do not try to sign in to the suspicious page.

Admins can use Okta Workflows to alert the end user through a back channel such as Slack or email and take other actions such as blocking traffic to and from the phishing site.

This verifier name binding through origin domain headers makes FastPass phishing-resistant. In rare instances, such as when native applications and malicious browser plugins are involved, an attacker may be able to change the origin headers in JavaScript programmatically. This gap would apply to most phishing-resistant authenticators available in the market today. On desktop devices, FastPass supports trusted app filters, which ensures that only trusted apps can invoke the FastPass authentication. Admins can create an allowlist of apps and require apps to be signed and verified in order to invoke FastPass. This helps ensure malicious or unverified apps cannot exploit FastPass to gain unauthorized access.

Sample User Journeys

Scenario #1: Phishing attempt


User Device
1 Lures the user to visit a fake site
2 User visits the fake site
3 Responds with Okta SIW and challenge
Okta Verify
4 Invokes FastPass and begins polling Okta
7 Alerts user of suspicious activity
8 Authentication fails
Bad actor running a proxy server (Evilginx)
Proxies access request
Okta Server
5 Sends signed response with origin header & device signals
Okta server notices origin domain mismatch
6 Phish alert – error code
8 Authentication fails
Phishing attempt is logged in Okta SysLog
Notifies user about phishing attempt
User
Okta Workflows

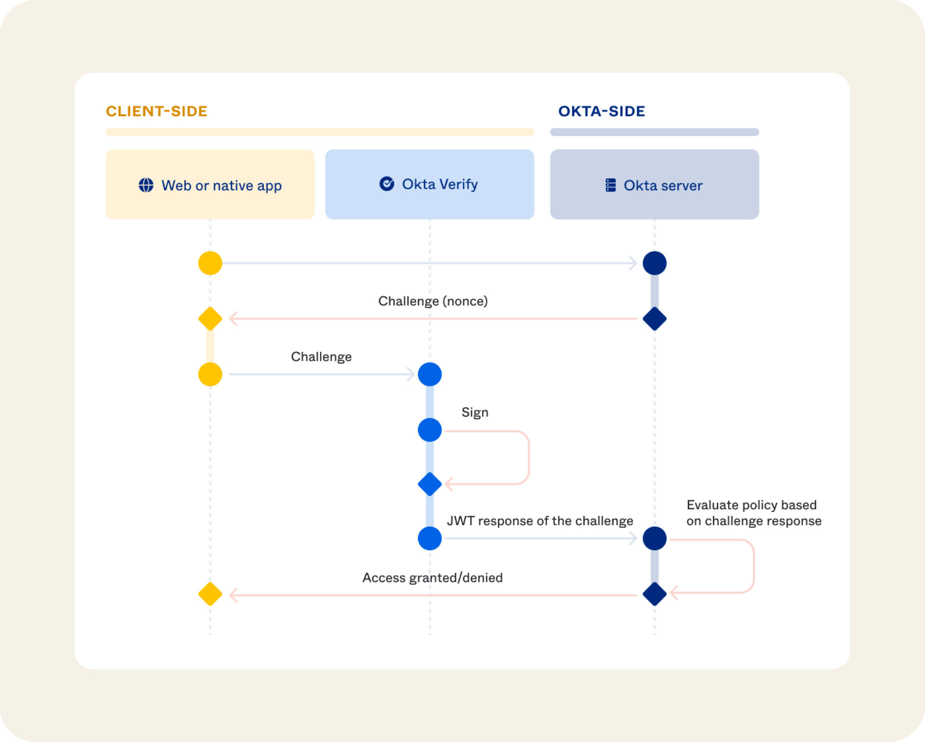
Here is a typical flow of FastPass phishing resistance:

1. The bad actor lures the user to visit a malicious website with a phishing email.
2. The user falls for the phishing attempt and clicks on a link to visit the fake site, which kicks off the authentication process. The malicious site proxies the request for access to the Okta server.
3. The malicious site then proxies the response, which includes the Okta SIW with the challenge nonce, back to the user.
4. The user decides to sign in with FastPass, and the SIW invokes FastPass via the loopback server and begins to poll the result from the Okta server.
5. FastPass signs the nonce, device signals, and the origin header, and posts the response back to the Okta server.
6. The Okta server validates the origin header and detects the mismatch between the expected origin header and the origin provided by FastPass. The Okta server responds back with an error code.
7. Okta displays a suspicious activity page to alert the user.

8. The Okta server logs the phishing attempt to the Okta SysLog, and fails the authentication.

9. The user is notified about the phishing attack through other channels, such as email or messaging if the admins have configured Okta Workflows to do so.

Scenario #2: Silent authentication



As the name suggests, silent authentication refers to the FastPass experience using the silent probing method. In this case, a user accesses the application from a registered device. This triggers a set of validation actions between the Okta server and the Okta Verify app installed on the device. If the user and the device satisfy the assurances required to sign in to the application, the Okta server grants access.

1. The user initiates authentication by visiting an Okta-protected resource.

2. The Okta server issues a unique challenge for that authentication request.

3. The SIW on the browser or native app forwards that challenge to Okta Verify that is installed on the same device by using loopback or credential SSO extension binding.
4. FastPass generates a response with the appropriate device signals and signs the response with the proof of possession private key that the user previously enrolled.
5. FastPass sends the challenge-response to the server.
6. The Okta server validates the signature and that the response corresponds to the unique challenge that was issued originally.
7. The Okta policy is evaluated based on the device context collected, and if satisfactory, the user is logged in.

Scenario #3: User presence and user verification

An Okta administrator can configure authentications to provide proof of user presence and user verification. In user presence-based authentications, FastPass prompts users to verify that they intend to log in to the specified application by showing a pop-up screen with a confirm button. When the response is signed with the proof of possession key, a particular claim is included in the response to indicate that user consent was approved.

In user verification flows, Okta Verify takes advantage of the biometric features of the hardware device, such as Touch ID, to perform user verification. When biometrics is not supported or preferred, a device passcode or PIN can also be used for user verification. In either case, the challenge is signed with the relevant user verification private key stored on the device. Access to such private keys requires user biometric presentation, using platform authenticators like Touch ID, Face ID, Windows Hello, and so on, or the input of the device passcode. As additional validation on top of FastPass, the user verification private key can attest to both proof of possession and user verification.

Here is a typical flow that asks for user verification in the form of biometrics:

1. The user initiates authentication by visiting an Okta-protected resource.
2. The Okta server issues a unique challenge for that authentication request.
3. The SIW on the browser or native app forwards that challenge to the Okta Verify app that is installed on the same device.
4. FastPass prompts the user for biometrics, and once the user addresses the required prompt, FastPass generates a response with the user verification private key previously enrolled by the user.
5. FastPass sends the challenge-response to the server.
6. The Okta server validates the signature and that the response corresponds to the unique challenge that was issued originally.
7. The Okta policy is evaluated based on the device context collected, and if satisfactory, the user is logged in.

Scenario #4: Managed device authentication

In the enterprise world, workforce devices are managed using an MDM or another endpoint management software. The administrator uses this software tool to manage device lifecycle, software installation, device compliance, and more, to ensure greater security in the organization.

Okta allows administrators to allow access to applications from managed devices only. Admins can also configure an app authentication policy such that access from a managed device needs lower assurances than an unmanaged device.

Here is a typical management attestation flow on a desktop device:

1. The user initiates the FastPass sign-in flow through Okta's SIW from a managed device.
2. The Okta server requests management attestation and device context to satisfy the policy.
3. The SIW passes the challenge to Okta Verify.
4. FastPass generates a response with device signals and management attestation. Management attestations are generated by signing the unique nonce with the certificate deployed on the desktop. The response is also signed with the proof of possession private key that the user enrolled.
5. FastPass sends the challenge-response to the server.
6. The Okta server validates the signatures and that the response corresponds to the unique challenge that was issued.
7. The Okta policy is evaluated based on the management status of the device and the device context collected, and if satisfactory, the user is logged in.

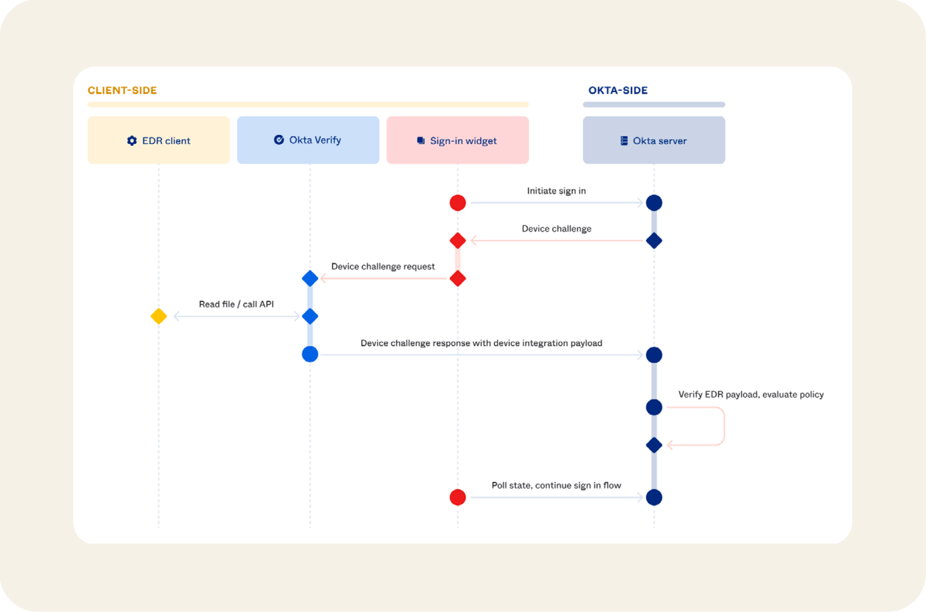
Scenario #5: FastPass integrations

FastPass integrates with third-party Endpoint Detection and Response (EDR) software to collect additional device security posture signals during authentication. These signals could be used in defining application authentication policies to guard access to sensitive resources.

Okta has a plugin framework to standardize how endpoint security integrations securely transport signals to Okta. Okta currently supports integrations with CrowdStrike and Windows Security Center (WSC). Details on how to configure these integrations can be found here.

Okta also supports the Chrome Device Trust connector to secure access to Okta-protected resources on managed ChromeOS and managed Chrome browsers on Windows and macOS. However, this connector does not require Okta Verify or FastPass to be used.

Here is a typical flow where endpoint security signals gate access:



1. The user initiates the FastPass sign-in flow through Okta's SIW.
2. The Okta server responds and challenges for the device condition.
3. The SIW passes the challenge to Okta Verify.
4. FastPass retrieves the required endpoint security signals from an EDR client using predefined integration methods. FastPass provides a response with the signals from the EDR client and the signals collected natively by FastPass, which is then signed with the proof of possession private key previously enrolled by the user, and sends the challenge-response back to the server.
5. The Okta server validates the signature and that the response corresponds to the unique challenge that was issued originally. It also verifies the uniqueness and authenticity of the endpoint security signals.
6. The Okta policy is evaluated based on the endpoint security signals and other device context collected, and if satisfactory, the user is logged in.

Conclusion

Okta FastPass offers users one of the safest ways to log in as a Zero Trust authenticator, designed for defense in depth to enable phishing-resistant authentication that continues to protect long after the initial request for access. By leveraging passwordless, phishing-resistant flows and device posture checks, FastPass can help ensure secure access to corporate resources while minimizing end-user friction. With FastPass, enterprises can seamlessly evaluate device context each time the user opens a protected resource for additional assurance of device posture before allowing access to downstream resources. FastPass also provides a user-friendly, consistent experience across all major platforms and devices, managed or unmanaged.

With its comprehensive features and focus on security, FastPass is the ideal solution for organizations aiming to balance security and user convenience for today's hybrid workforce.

To learn more about Okta FastPass, visit www.okta.com/fastpass

Disclaimer:

These materials and any recommendations within are not legal, privacy, security, compliance, or business advice. These materials are intended for general informational purposes only and may not reflect the most current security, privacy, and legal developments nor all relevant issues. You are responsible for obtaining legal, security, privacy, compliance, or business advice from your own lawyer or other professional advisor and should not rely on the recommendations herein. Okta is not liable to you for any loss or damages that may result from your implementation of any recommendations in these materials. Okta makes no representations, warranties, or other assurances regarding the content of these materials. Information regarding Okta's contractual assurances to its customers can be found at okta.com/agreements.