Whitepaper

FastPass Technical Whitepaper

FastPass Technical Whitepaper

What is Okta FastPass?

Okta FastPass is a cryptographic multi-factor authenticator that enables end users to sign in to Okta-protected applications without using a password. Okta's End User Dashboard, along with any OIDC, SAML, or Web Services Federation application available on the Okta Integration Network (OIN), can be made available in a passwordless and Zero Trust manner (users are given access on an as-needed basis).

In order to use FastPass, end users need to have the most recent version of the "Okta Verify" authenticator app available on their desktop, laptop, or mobile device. Okta Verify with FastPass support is available for iOS, Android, Windows, and macOS platforms. Thus, passwordless flows can be enabled on these devices.

FastPass uses public key cryptography to authenticate the user.
When an end user enrolls in Okta Verify, public and private keys are generated on the device. The private keys are stored securely on the device, whereas the public keys are sent to Okta's cloud service. When challenged by Okta's cloud service, FastPass signs a one-time nonce and returns it. Okta's cloud service validates the digital signature to authenticate the factor.

FastPass can be deployed in a phishing-resistant way, as it validates the origin header from a malicious website and detects any mismatch from the original website.

FastPass integrates with the platform authenticator on the device (such as Windows Hello, Touch ID, or Face ID) and can support biometric authentication. FastPass can satisfy both "Proof of Possession" and "User Verification / Inherence" factor requirements. If there is a PIN fallback method, such as on Windows Hello, the factor is then "Proof of Knowledge."

FastPass can integrate with Unified Endpoint Management (UEM) and Endpoint Detection and Response (EDR) vendors. When integrated, FastPass can collect additional security signals such as device management attestation and device security postures (example: risk score). Okta Admins can build authentication policies (per app) using these device posture signals to further enhance their Zero Trust implementations.

Benefits of FastPass

Benefits to end users

Passwordless login experience

End users can enjoy passwordless authentication to all the services protected by Okta. This significantly improves the employee experience by reducing the friction introduced by passwords. FastPass also reduces the dependency on out-of-band factors such as Push, Time-based One-Time Passwords (TOTP), and SMS.

Consistent user experience across platforms

FastPass works on iOS, Android, Windows, and macOS. End users can benefit from having the same passwordless experience across all their devices.

Biometric support

FastPass integrates seamlessly with platform authenticators, such as Face ID, Touch ID, and Windows Hello, that may be needed to access applications requiring higher assurance.

Benefits to admins

Phishing resistance

FastPass prevents most common phishing attacks for managed and unmanaged devices on all supported platforms when required by policy. Supported platforms include Windows, Android, MacOS, and iOS devices. Administrators can enable phishing-resistant constraints in the app sign-on rules for applications they want to protect. Both “Proof of Possession” and “User Verification” factors enrolled with FastPass are phishing-resistant factors. Both admins and end users are notified when a phishing attempt is detected.

Elevate security posture

Eliminating passwords reduces the attack surface (example: credential stuffing). Replacing the passwords with device-bound cryptographic authentication factors enhances security and is foundational to Zero Trust.

Increased Productivity

After introducing FastPass, Okta's internal IT team saw a 98% reduction in password resets. When employees are not busy typing and resetting their passwords, they're getting more work done.

Rich Device Context

FastPass verifies the device in use during authentication. Administrators can leverage rich device context to make authentication and authorization decisions. Okta Verify also integrates with endpoint security software to collect additional signals to enhance security checks. Okta Verify agents communicate with Workforce Identity Cloud via a SSL channel.

Manage Device Lifecycle

Administrators can manage the lifecycle of the devices enrolled in FastPass. Administrators can remotely suspend/unsuspend, activate/deactivate devices as they see fit. These actions are also available as APIs. See the API documentation.

Interoperability

FastPass fits seamlessly into an Okta org's OIDC, SAML, and WS-Federation flows. The IdP (Okta) handles collection of the FastPass-based factors from the end user's device without any changes on the Service Provider (SP) end. It is an authenticator that can be combined with any other authenticator to meet high assurance standards for a sign-on policy.

Moreover, irrespective of whether the user begins their journey on an SP site or Identity Provider (Okta) site, the FastPass experience is consistent.

Key concepts

Here are some key concepts that make FastPass work.

Device identity

When a user enrolls in FastPass via the Okta Verify app, a unique device identity is created in Okta's Universal Directory (UD). The device itself is assigned a Device ID that persists across enrollments. Universal Directory stores additional context about the device such as the device's display name, OS, model, manufacturer, management status, etc. For a full list, go here. The device context information is always stored and encrypted in UD.

An org administrator can search for a FastPass-enrolled device in the admin UI and take action on a device: suspend, unsuspend, deactivate, reactivate, and delete. More information about different lifecycle states can be found here.

Device enrollment

When the end user enrolls in the Okta Verify app, the device is registered into Okta's Universal Directory. This step encapsulates the following:

- Key pairs are generated on the device, and public keys are sent to the Okta service.
- User and device association is made.
- User is enrolled in the appropriate factors ("Proof of Possession" factor and optionally "User Verification" factor on the current device).

Phishing resistance (NIST definition)

The National Institute of Standards and Technology (NIST) has published Special Publication 800-63B, which articulates technical requirements for federal agencies implementing digital identity services and helps define phishing-resistant MFA. The key phishing resistance attributes identified in this publication include:

- **Verifier impersonation resistance** using cryptographic binding between authenticator and user identity
- **Replay resistance** via OTP devices, cryptographic authenticators, and look-up secrets
- **Verifier-compromise resistance** by ensuring that any public keys stored by the verifier are associated with the use of approved cryptographic algorithms
- **Authentication intent** that requires the user to explicitly respond to each authentication or re-authentication request

“Proof of Possession” factor
This factor satisfies possession requirements (“something you have”) using cryptographic signature verification.

When the user enrolls in the Okta Verify app on a device, Okta Verify generates a key pair and designates it as the “Proof of Possession” key pair. The private key is stored in the hardware keystore of the device if available, whereas the public key is sent to the Okta service. During the authentication flow, the Okta service uses this public key to verify that the signature of the payload was signed by the Okta Verify app with the corresponding private key.

If the signature verification is successful, the user is considered to have provided a Possession factor. If the application sign-on policy for the application requires additional factors, the Okta service challenges the user for another authentication factor.

This factor can be configured to be collected with or without the User Presence check. If the User Presence check is enabled, Okta Verify will ensure it gets a user interaction before proceeding with the verification.

“User Verification” factor
This is the biometric factor (“something you are”) and satisfies User Presence and User Verification requirements.

During enrollment, if the end user chooses to enroll in the “User Verification” factor, an additional key pair is generated. The private key is stored behind the platform’s biometric capabilities (example: Touch ID for macOS) and the public key is sent to the Okta service. The Okta Verify app is able to sign a payload with this key only after the end user provides their biometrics.

If the device has a new biometric added after this key is generated (for example, if a user registers a new fingerprint on their phone after enrolling), the key will be invalidated and the end user will have to authenticate inside of Okta Verify in order to generate a new key to use for challenges that want a User Verification factor.

Device context

During the authentication flow, Okta Verify collects context related to the device. It includes basic device signals such as platform name, OS version, and device display name. Based on the configuration, Okta Verify can also collect signals such as management attestation, device compliance, jailbreak status, etc.

Device probing

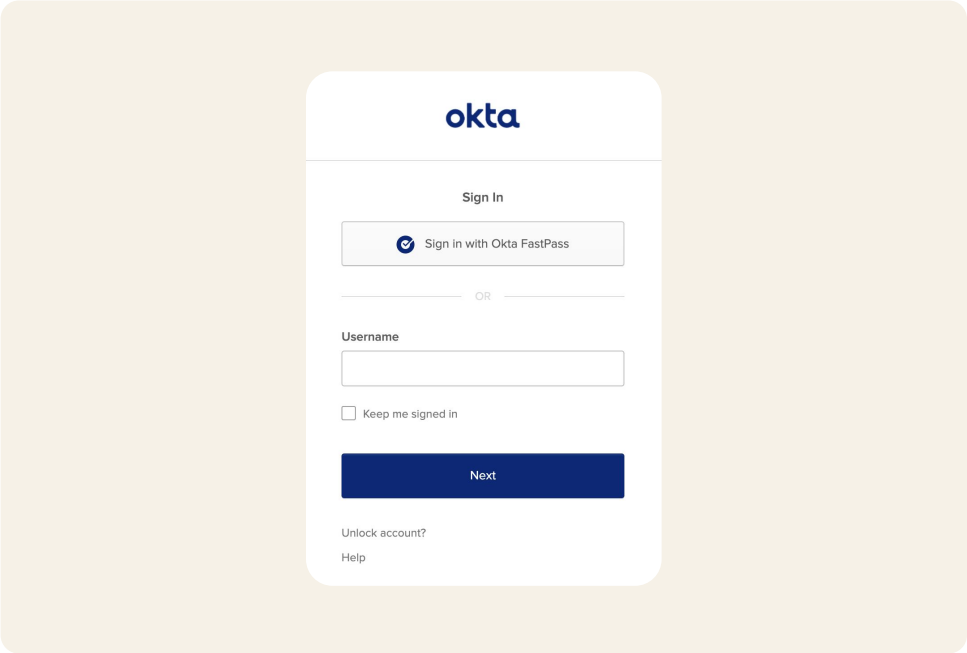
Probing is a mechanism by which Okta's Sign-in Widget (SIW) running in a browser tab communicates with Okta Verify on the device. If Okta Verify is not installed, then the probing mechanism fails and the user is prompted for other authenticators.

Silent vs interactive probing

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

Method Name	Method Type	Supported Platforms
Loopback	Silent	macOS, Windows, Android, iOS
Credential SSO Extension (managed devices only)	Silent	iOS, macOS (Safari browser only)
AppLink	Interactive	Android
Custom URL	Interactive	Windows, macOS
Universal Link	Interactive	iOS

The SIW falls back to an interactive probing method if a silent probing method is not available (for instance if the loopback server fails to start). In these cases, end users need to launch Okta Verify with 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, FastPass attempts to collect:

- Proof of Possession factor only
- Proof of Possession factor and User Verification factor

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

Security model

Key generation and protection

A user's enrollment in FastPass on a device generates two key pairs — one for "Proof of Possession" factor and one for the "User Verification" factor. By default, the private keys are stored in a device's hardware key store if available. Some examples of this include the Trusted Platform Module (TPM) or Secure Enclave (iOS). The private keys never leave the hardware keystore 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 software key store.

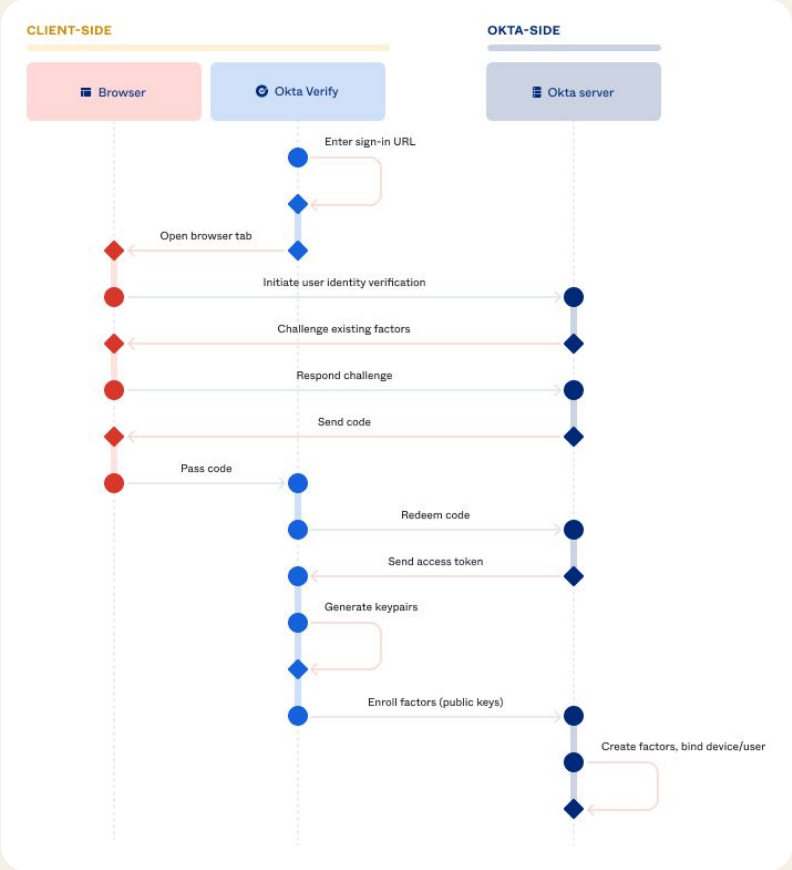
During factor collection, the Okta Verify application uses the hardware keystore to get the digitally signed output for a given input payload. This signed output payload is sent to the Okta service for signature verification. The org administrator can build sign-on policies to require hardware-bound (i.e., TPM stored) keys. See "Configure an authentication policy for Okta FastPass" for more details.

Platform	Digital Signature Algorithm Used
macOS	ES256
Windows	RS256
iOS	ES256
Android	RS256

iOS and Android Okta Verify apps also support TOTP and Push factors. These factors are out of the scope of this whitepaper.

FastPass In Depth

Enrollment



FastPass enrollment uniquely binds a set of keys to the device and to the user. Before enrolling a user, the Okta Verify app challenges the user for other factors. The current policy is to require two-factor authentication if possible; admins can configure enrollment policies. Once the identity is verified, Okta Verify asks if the user wants to enable biometric authentication (Touch ID, Face ID, Windows Hello, etc.). Okta Verify generates a key pair for “Proof of Possession” factor, as well as a key pair for “User Verification” factor if biometrics are enabled.

Admins can require a phishing-resistant authenticator to enroll additional authenticators. And for end users that want to enroll a second device in Okta Verify, they can do so in a phishing-resistant manner by Bluetooth (for Android, iOS, macOS, or Windows) or via Yubikeys.

Probing schemes

The Probing schemes, when successful, bind the device to the session. FastPass supports the following 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 available on Windows, macOS, Android, and iOS.

Credential SSO extension

Users on managed iOS and macOS (Safari browser only on macOS) devices can have a seamless FastPass experience with Credential SSO Extension.

Universal Link

On iOS devices, Universal Link allows 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.

App Link

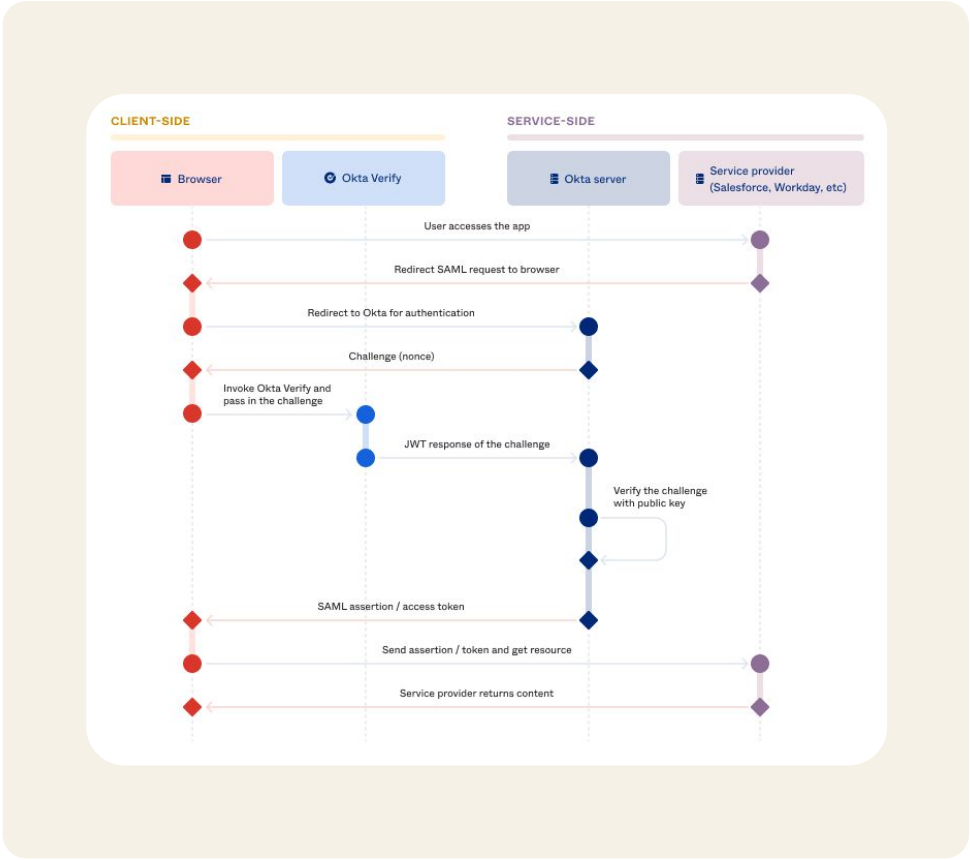
On Android devices, App Link allows the SIW to launch the Okta Verify app with a user click. The SIW appends the challenge request to the App Link so that when the user clicks on "Sign In with FastPass," the challenge request is available to FastPass. App Link will not always work in native application authentication flows based on Okta's testing.

Custom URL

On macOS and Windows devices, the Custom URL 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 scheme when clicked. The SIW appends the challenge request to the Custom URL.

Authentication flow and probing

The following diagram illustrates a typical FastPass passwordless flow for the SAML-based single sign-on (SSO) process.



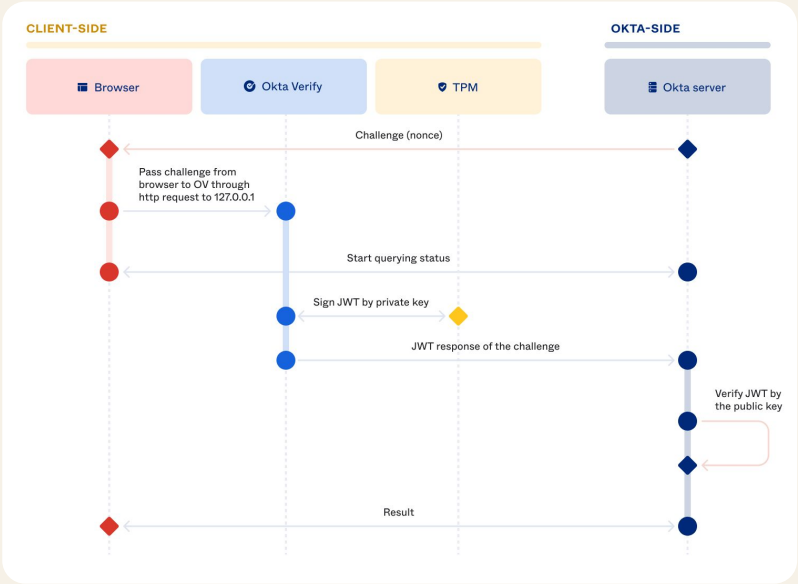
A typical SAML SSO flow, with FastPass

1. End users access services such as Salesforce or Workday from a browser.
2. Service providers generate the SAML request and redirect the request to Okta.
3. Okta backend evaluates the request and generates the device challenge if a device condition is configured in the app sign-on policy. Okta backend sends the device challenge along with the SIW to the browser. The SIW on the browser invokes Okta Verify installed on the device.
4. Okta Verify collects the device signals (such as device platform, device name, device hardware capabilities, etc.) and sends the signals signed by the private key (enrolled by the user on the client side) to the Okta server.

5. The Okta server identifies the user by checking the signature. The device information thus collected can be used during policy evaluations for making authentication decisions.
6. If the policy conditions and device assurance match, Okta backend generates the SAML assertion and redirects the browser back to the service provider. Otherwise, Okta backend will send a challenge for the second factor or block access altogether based on the result of policy evaluation.

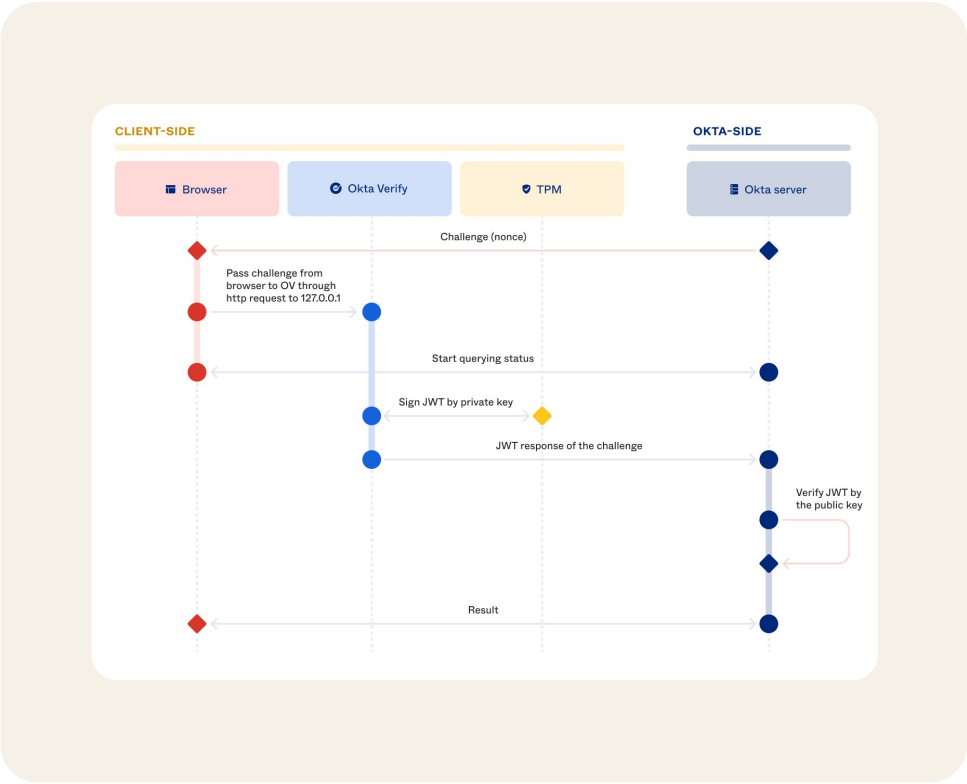
Factor collection when using loopback

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 and device-local communication between the browser session and the local app install. This server allows 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/intercept certain http traffic between the browser and Okta server. When a 401 status is detected, Okta Verify initiates the signed nonce challenge and response flow, as with Loopback.



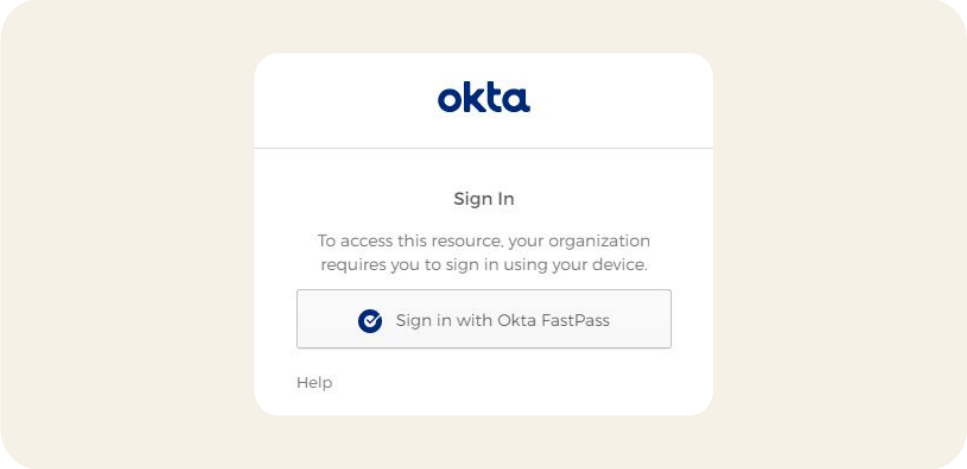
Factor collection when using Custom URL and Universal Link

When Loopback or Credential SSO fails, browsers can also launch and pass challenges to Okta Verify through deep links. For Windows and Mac, we use the Custom URL scheme; for Android, we use App Link; for iOS, we use Universal Link. App Link and Universal Link are more secure techniques since only verified apps can be invoked, but they are not available on all platforms.

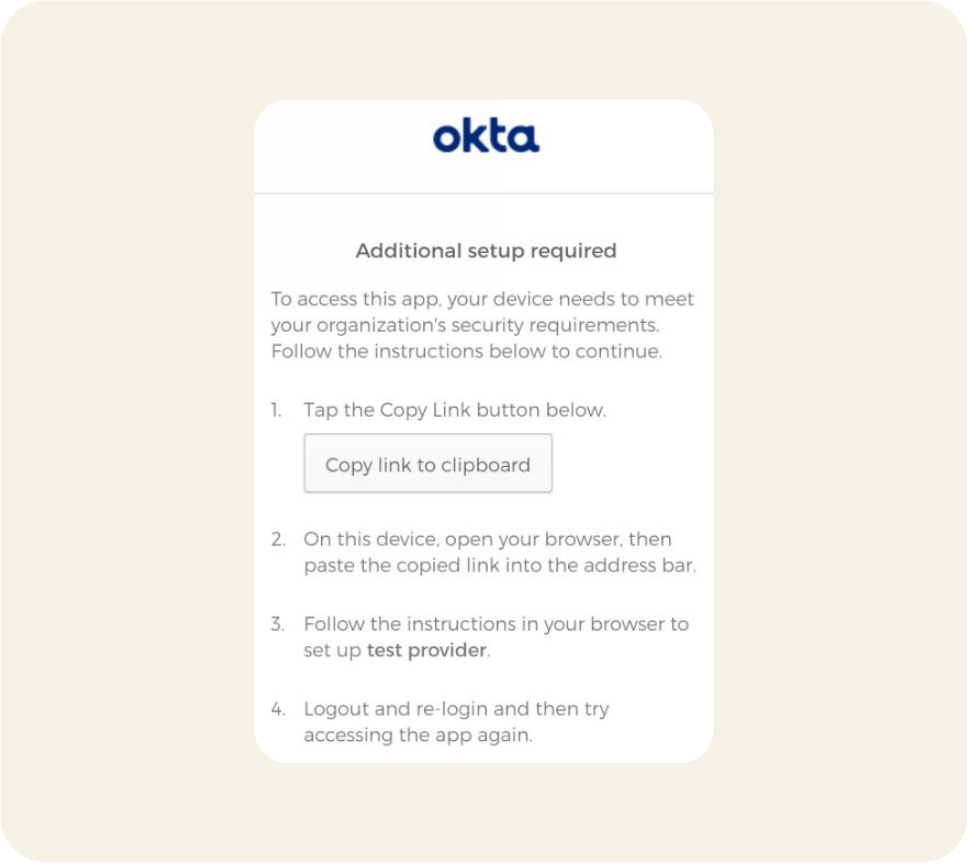
Remediation

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

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



- Example #2: When a managed device is required



Managed devices

FastPass integrates with most of the device management software vendors that support SCEP profiles (distribution of the client certificates for Windows/macOS) and managed app configurations (Okta Verify iOS, Android applications).

During the authentication flow, Okta Verify sends management signals along with other device signals to the server for management attestation verification.

On desktop (Windows and macOS) devices, the client certification issued by Okta CA or your own CA is used to create the management attestation signal.

Okta CA uses SCEP protocol to deploy client certificates to the managed desktop devices by integrating with the device management vendors. Okta CA provides static, dynamic, or delegated modes of SCEP certificate deployments. Okta recommends that the administrators configure the MDM SCEP policy in such a way that private keys are stored in the device hardware key stores and certificates are non-exportable.

If the app sign-on policy requires the managed device condition, then the Okta service requests the device management attestations during the FastPass protocol. The Okta Verify client on Windows or macOS identifies a correct client certificate deployed on the device and uses it to sign a unique nonce in the request to create management attestation. The Okta service 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 service 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 service associates every client certificate to the device. This prevents misusing client certificates from unauthorized devices to provide management attestation.

On mobile devices, the management signal is a shared secret generated by Okta during the endpoint security configuration, and configured as the managed config attributes on Okta Verify in the MDM system by the administrator. Similar to certificate-based deployments, Okta Verify responds with the shared secret when challenged by the Okta service during the FastPass protocol, and then the Okta service verifies the secret by comparing to a hash that was stored during the initial configuration. Please note that the Okta service does not store the raw secret, so you will need to securely store this yourself.

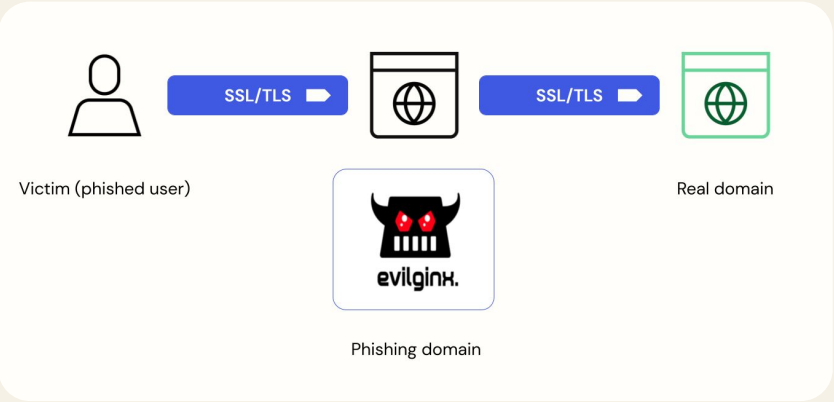
You can read the steps for configuring and deploying managed devices here.

Phishing resistance

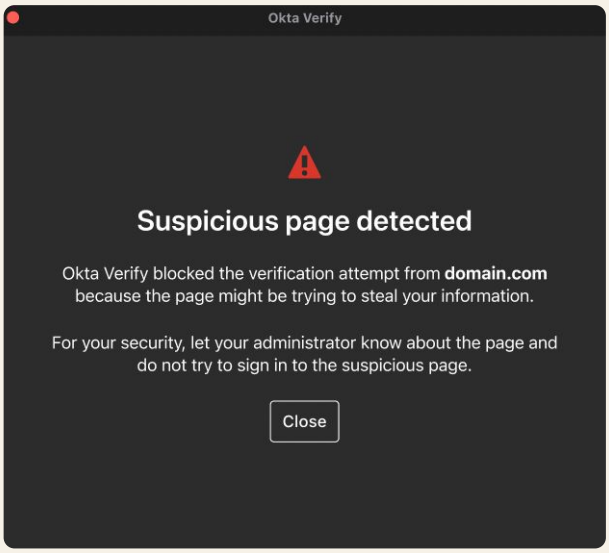
Phishing is a type of social engineering attack often used to steal user data. It typically occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening a link and providing credentials to the attacker.

AiTM Attack Framework with evilginx

Adversary-in-the-Middle Attack Framework used for phishing login credentials along with session cookies



FastPass helps prevent one of the most common types of phishing attacks, Adversary-in-the-Middle (AiTM), with more success than other authenticators. In order to masquerade as a trusted entity, an attacker will need to use a proxy to initiate the Okta SIW. This requires a proxy request from the malicious site to the Okta loopback servers. The loopback server will be able to validate the origin header and detect any mismatches. There is no way for an attacker to programmatically change the origin headers in JavaScript, which makes this feature phishing resistant. When this is detected, authentication fails, the event is logged in SysLog, and the user is shown a suspicious activity page.

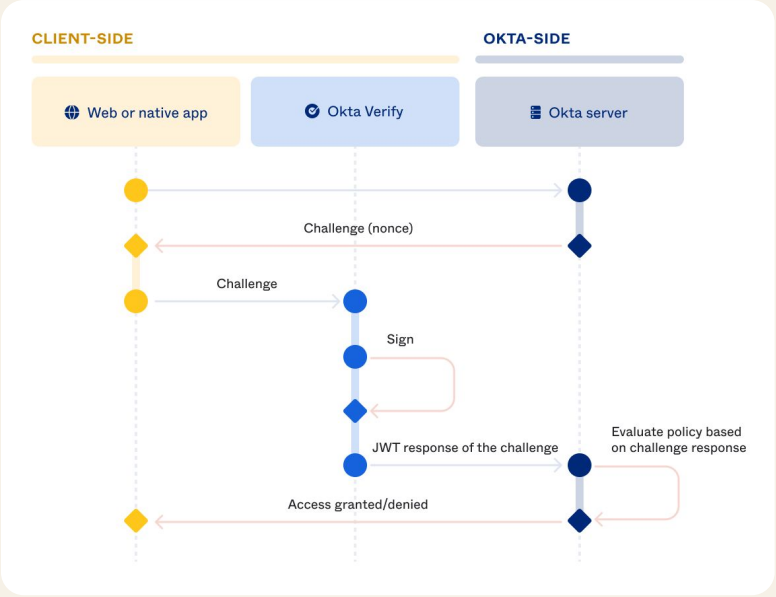


Admins can use Okta Workflows to alert the end user through a back channel such as Slack or email, as well as take other actions such as blocking traffic to and from the phishing site.

Combining this with device assurance policies, Okta can provide strong protection from other forms of endpoint attacks, such as malware or ransomware. Okta's partners, such as Crowdstrike and Windows Security Center, can provide valuable signals on the presence of malware on the device, which Okta leverages to enforce strong authentication policies.

Use cases (aka user journeys)

Scenario #1: Silent authentication



As the name suggests, the silent authentication experience refers to the passwordless 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 service and the Okta Verify app installed on the device. If the user and the device satisfy the assurances required to access the application, the Okta service grants the access.

1. User initiates authentication by visiting an Okta protected resource.
2. Okta server issues a unique challenge for that authentication request.
3. 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. Okta Verify generates a response with the appropriate device signals and signs the response with one of the private keys that was previously enrolled by the user.
5. Okta Verify sends the challenge-response to the server.
6. Okta server validates the signature and that the response corresponds to the unique challenge that was issued originally.
7. Okta Policy is evaluated based on the device context collected, and if satisfactory, the user is logged in.

Scenario #2: User presence or biometric authentication

An Okta administrator can configure authentications to take advantage of user presence or user verification features.

In user presence based authentications, FastPass prompts the user to verify that they intend to login to the specified application by showing a pop-up screen with a confirm button.

In user verification flows, FastPass takes advantage of the biometric features of the hardware device, such as Touch ID and fingerprint, to perform user verification. In this case, Okta Verify uses the user verification private keys that are stored in the secure hardware. Access to such private keys requires user biometric presentation by using platform authenticators like Touch ID, FaceID, fingerprint, Windows Hello, etc.

Here is a typical flow:

1. 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.
4. Okta Verify prompts the user for either a confirmation prompt or biometric prompt, and once the user provides confirmation, Okta Verify generates a response with the appropriate private key that was previously enrolled by the user.
5. Okta Verify 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. Okta Policy is evaluated based on the device context collected, and if satisfactory, the user is logged in.

Scenario #3: Managed device authentication

In the enterprise world, workforce devices are managed using endpoint management software. The administrator uses this software to manage device lifecycle, software installation, device security postures, device compliance, and more to ensure better security in the organization.

To enable more access control, Okta provides administrators the ability to support access to applications from managed devices only. An Okta administrator could also configure an app sign-on policy such that access from a managed device needs lower assurances compared to the unmanaged devices.

Here is a typical device attestation flow from a desktop device:

1. Okta sign-on policy requires management attestation for the resource access.
2. The user initiates the sign-in flow through Okta's SIW from a managed device.
3. Okta responds and challenges the device condition. Okta also requests the management attestation along with the device context to satisfy the policy.
4. The SIW passes the device challenge to Okta Verify.
5. Okta Verify generates a response with device signals and management attestations.
    a. Management attestations are generated by signing the unique nonce with the certificate deployed on the device.
6. Okta Verify then signs the response with private keys that were enrolled by the user.
7. Okta Verify sends the challenge-response to the server.
8. The Okta server validates the signature and that the response corresponds to the unique challenge that was issued.
9. Okta policy is evaluated based on the management status of the device, and if satisfactory, the user is logged in.

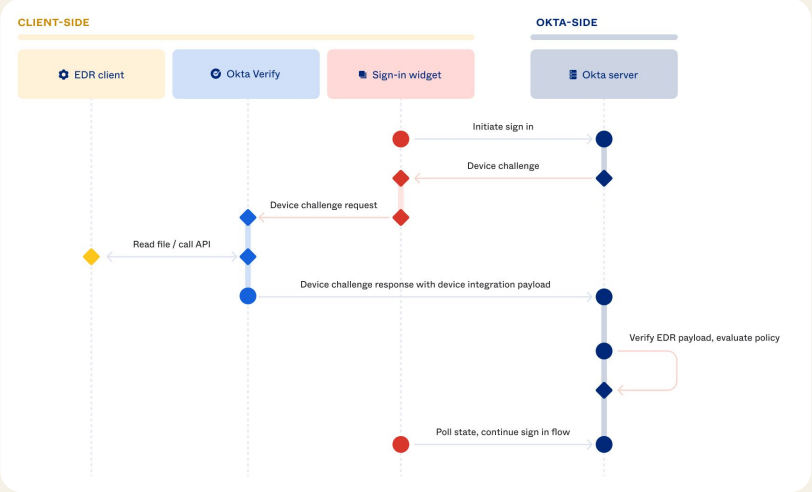
Scenario #4: FastPass integrations

FastPass integrates with Endpoint Detection and Response (EDR) software to collect additional device security posture signals. Okta Verify collects security signals from the partner EDR software installed on the endpoint during the authentication process. These signals could be used in defining application sign-on policies to guard secured resources.

Okta has defined a plugin framework to standardize how endpoint security integrations transport the signals to Okta in a secure way.

Okta currently supports integration with CrowdStrike and Windows Security Center (WSC).

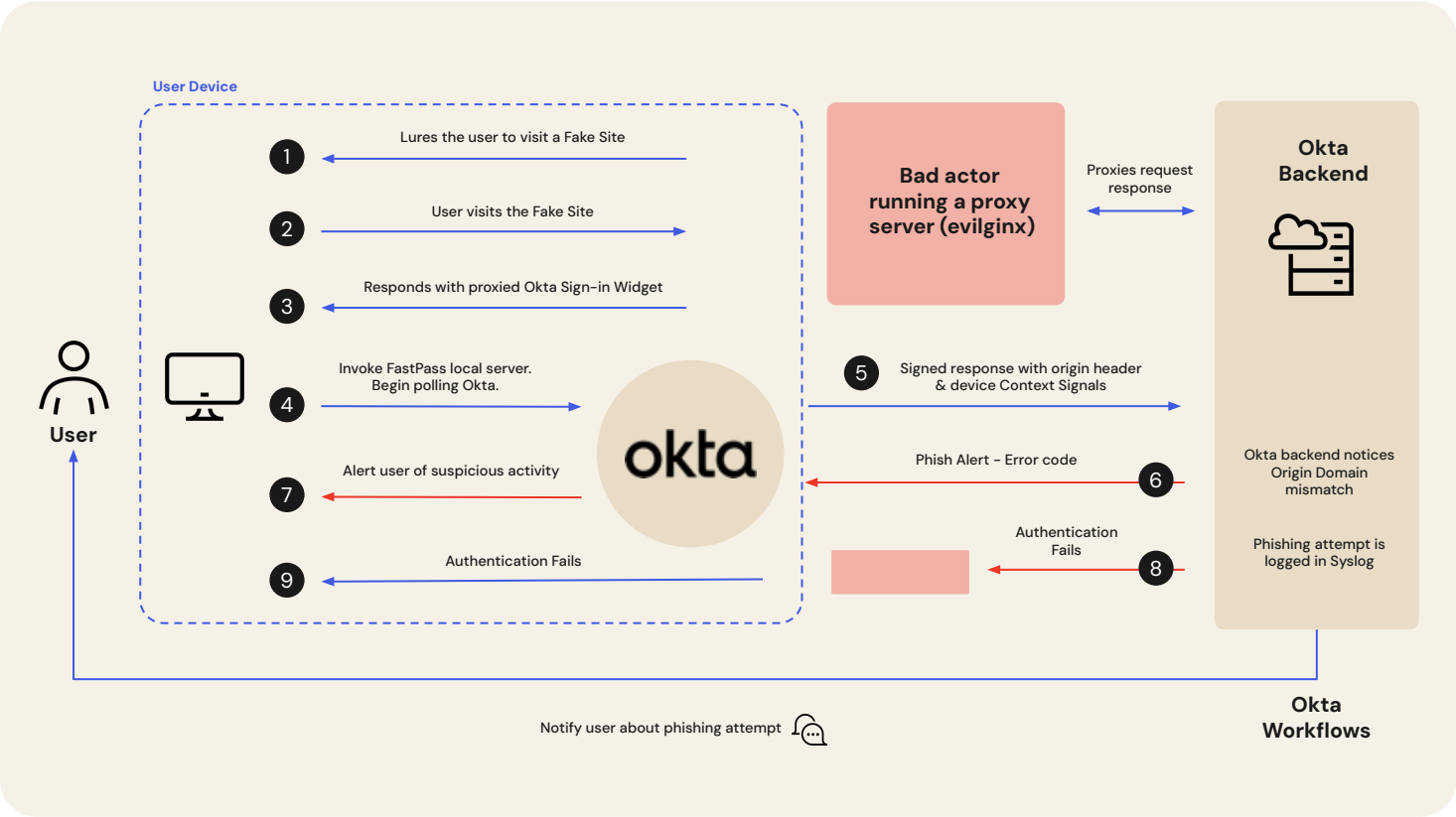
Details on how to configure the integrations can be found here.



Here is a typical flow:

1. Application access is gated by the endpoint security signals.
2. The client initiates the sign in flow through Okta's SIW.
3. Okta responds and challenges the device condition.
4. The SIW passes the device challenge to Okta Verify.
5. Okta Verify retrieves endpoint security signals from an EDR client using predefined integration methods. Then, it generates a response with the appropriate device signals, including the signals from the EDR client and the signals collected natively by Okta Verify. Okta Verify signs the response with the private keys that were previously enrolled by the user and 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. It also verifies uniqueness and authenticity of the endpoint security signals.
7. Okta Policy is evaluated based on the endpoint security signals, and if satisfactory, the user is logged in.

Scenario #5: Phishing attempt



Here is a typical flow:

1. The bad actor lures the user to visit a malicious website by sending the malicious link via text message or email.
2. The user clicks the malicious link to visit the malicious site.
3. The malicious site proxies the request to Okta backend and responds back to the user with the Okta SIW.
4. The SIW invokes FastPass via loopback server and begins to poll the result from the server.
5. Okta Verify signs the nonce, device context and the origin header, and posts the response to Okta backend.
6. Okta backend validates the origin header from Okta Verify and detects the mismatch between the expected origin header and the origin from Okta Verify. Okta backend responds back with an error code.
7. Okta Verify displays a suspicious activity page to alert the user.
8. Okta backend logs the phishing attempt to the SysLog, and fails the authentication.
9. The attacker fails to authenticate.
10. The end user is notified about the phishing attack through other channels such as email or instant messaging if the admins have configured Okta Workflows to achieve that.

Conclusion

Okta FastPass is a powerful cryptographic multi-factor authenticator that provides secure passwordless authentication to SAML, OIDC, and WS-Fed applications. By leveraging phishing-resistant flows and adaptive policy checks, FastPass ensures secure access to corporate resources while minimizing end user friction. With FastPass, enterprises can protect against real-time credential phishing attempts on any device across all major operating systems. FastPass integrates with any device management tool to enforce phishing-resistant flows and amplifies the security of your devices with device assurance policies. With support for device-level biometrics, FastPass helps eliminate additional login friction when accessing Okta-managed apps. With its comprehensive features and focus on security, Okta 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 https://www.okta.com/fastpass/

About Okta

Okta is the World's Identity Company. As the leading independent Identity partner, we free everyone to safely use any technology — anywhere, on any device or app. The most trusted brands trust Okta to enable secure access, authentication, and automation. With flexibility and neutrality at the core of our Okta Workforce Identity and Customer Identity Clouds, business leaders and developers can focus on innovation and accelerate digital transformation, thanks to customizable solutions and more than 7,000 pre-built integrations. We're building a world where Identity belongs to you.

Learn more at **okta.com**.