WHITE PAPER – **NOVEMBER 2016**

DELIVERING SECURITY AND ONE TOUCH SINGLE SIGN ON FOR NATIVE MOBILE APPS ON ANY DEVICE WITH WORKSPACE ONE

vmware®

Table of Contents

Workspace ONE Overview

The digital workspace is the defining model for end-user computing in the mobile cloud era, delivering secure anytime, anywhere access to any application to any device including desktops, smartphones and tablets.

Organizations delivering a digital workspace are enabling users to take full advantage of a portfolio of devices and applications that are increasingly diverse and heterogeneous.

The digital workspace advances the early concepts of the client-server era desktop to give IT a more efficient, simplified way of managing devices and applications, while providing users with seamless access all of their business resources, regardless of device type. It also gives the line of business a secure and powerful platform on which to build and rebuild business processes that make an increasingly mobile workforce more productive in their roles and more competitive in the market.

Taking its inspiration from the latest advances in consumer technologies, the digital workspace enhances the end-user experience and simplifies IT management while preserving all of the reliability and security aspects required for business critical applications and sensitive corporate data. This "consumer simple – enterprise secure" relationship is a central pillar of the digital workspace. With the expansion of mobility, it's become increasingly difficult for organizations to bridge the gap between support for their existing enterprise applications and new cloud and mobile apps, while simplifying access for employees.

The consumer grade self-service capability of VMware Workspace™ ONE™ puts access back into the hands of the user, with policy-controlled options for managed and unmanaged devices. It also offers the most comprehensive coverage of internal and public cloud, mobile and Windows applications and is the industry's first for public apps. One-touch mobile SSO leverages an industry-first feature for single sign-on allowing users to access these apps without passwords or complex PIN challenges.

Single Sign-On from Devices

In situations where app wrapping is practical, providing single sign-on is already possible, as you can see here. However, in cases where the app is obtained *from the public app store*, the challenges are harder, and this is exactly what this paper and Workspace ONE addresses. Workspace ONE leverages operating system level features offered by the dominant mobile operating systems - iOS, Android and Windows (10) - to provide single sign-on capabilities.

In order for the end user to properly use the SSO functionality they must have an AirWatch managed mobile device that is in compliance and has downloaded all of the recommended applications to their device. As you read on, you will see how VMware and take into account differences in the three platforms and provides a simple end user experience that is an industry first.

Apple iOS

WorkSpace ONE uses the native iOS Kerberos capabilities along with technology developed in VMware Identity Manager™ that provides a bridge between SAML and Kerberos for authentication. iOS supports Kerberos via it's MDM profile, com.apple.sso, however for enterprise use, there are a couple of questions around Kerberos which customers may have:

- The Key Distribution Center (KDC) component of Kerberos contains company secrets, and for this reason usually sits deep inside the security zones of a company. How are we planning to give access to the KDC so that mobile devices (potentially out in the internet) can access it?
- Many modern service providers like Salesforce.com or ServiceNow.com, etc. do not support Kerberos, or are not "Kerberized." They do not accept Kerberos Tickets. Even if they did, the Kerberos scheme would require the KDC of all companies to know some secret of the service providers. How will this work?

VMware Identity Manager solves both these problems by standing up a KDC in the cloud (a.k.a. Built-In KDC). This KDC does not need any company or user secrets because it relies on the user certificate which is presented by the user/device (provisioned to the device using VMware AirWatch® enrollment). Further, VMware Identity Manager bridges the "protocol gap" by talking SAML with the service provider and talking Kerberos with the iOS device. This creates a secure mechanism which works nicely for end users, service providers and most importantly, our customers.

Windows 10

The Windows 10 platform offers a well-defined mechanism for vendors to provide single sign-on to apps. There is a way to provision certificates on the device during MDM enrollment, and thereafter.

Android

The Android platform does not lend itself that easily for this type of functionality. It has no out of the box way of provisioning certificates to the device, however does offer all apps the access to a managed key store on the device, which can potentially be leveraged for SSO.

Single Sign On on iOS

Steps 1 through 6 below describe the process flow, combining elements of SAML and Kerberos.

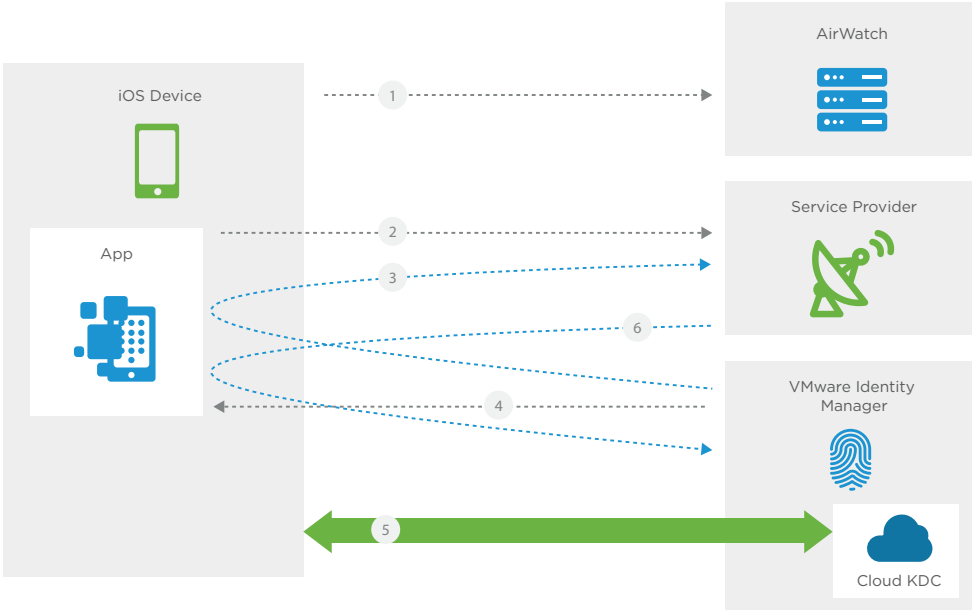


Figure 1: High Level Flow Diagram of the Single Sign On Process

1. After buying the device, the employee initiates the enrollment process with an AirWatch server. Enrollment is 'enabled' by the employee authenticating himself to his employer and results in an enterprise-trusted iOS device. At the end of the enrollment process, the device has:

   a. A private and public key pair for the device, with the public key being stored on the AirWatch server for trust purposes. This allows the AirWatch server to trust the device every time the device "checks-in". iOS does not allow access to the private key. Only IT can access it.

   b. A Kerberos x.509 certificate which is issued to the user. This is a regular Kerberos certificate with the addition of key value pairs as specified by Kerberos version 5.

   c. The com.apple.sso MDM profile which specifies which specific applications on the device will need to participate in the Kerberos process. This is the MDM administrator telling the iOS operating system which applications have been recommended by the company for Kerberos. For all other applications, iOS will not interfere in the authentication process.

2. Employee taps on an app obtained from the app Store. That app now needs to understand how to authenticate this user. For this purpose, the app invokes what is known as "Tenant Discovery". With the help of its backend server (its cloud-based server counterpart) the app challenges the user for his email address, or some other company identifier, which the employee does. If the server can recognize the company as a customer, it also has some settings to decide how to authenticate the user for customer. In this case, that answer should be VMware Identity Manager, and the authentication method is SAML.
3. The server starts the SAML process by sending back a SAML request in a URL which the app HTTP redirect sends to the VMware Identity Manager address (in the cloud or on). The VMware Identity Manager address is also contained in the redirect URL.
4. When VMware Identity Manager gets the SAML request, it does its usual checks to determine whether the single sign-on should be allowed or not. In this case, because this is the first time the user and device are coming to VMware Identity Manager, it sends an HTTP 401 (Unauthorized) back to the app. Per iOS requirement, it includes a "WWW:Authenticate – Negotiate" in the header of this HTTP 401. This is where iOS recognizes that a 401 has come for an app (listed in com.apple.sso) which needs Kerberos authentication. It intercepts that 401.
5. iOS then participates in the Kerberos process with Built-In KDC inside VMware Identity Manager. Kerberos v5 includes support for PKINIT, using whichever method iOS is able to (using the Kerberos x.509 Certificate in it's possession) prove to KDC that the user is who he says he is. There are three transactions in the Kerberos Process.
   a. AS-REQ and AS-REP: iOS and Built-In KDC establish mutual trust and Built-In KDC issues a Ticket Granting Ticket (TGT) to iOS.
   b. TGS-REQ and TGS-REP: iOS presents the TGT to Ticket Granting Server (a component of Built-In KDC) and obtains a Kerberos ticket in return.
   c. AP-REQ and AP-REP: iOS submits the ticket to what it thinks is the service. In fact, iOS submits the ticket to VMware Identity Manager, and VMware Identity Manager tells iOS that access is granted (in a format prescribed by Kerberos).
6. VMware Identity Manager replies back to the HTTP redirect (from 1 above) with an HTTP redirect of its own with a SAML response which contains the SAML assertion. This is sent back to the Service Provider, which will consume that SAML response. If trust is already established between it and VMware Identity Manager, the Service Provider will be able to validate that SAML and grant the user access to its app.

Components on iOS

Applications

Most publicly available and enterprise-built iOS applications will support this single sign-on functionality out of the box. In order for an application to support SSO, it must have enabled SAML 2.0 and use the native iOS webKit.

Built-In KDC (Key Distribution Center)

The Built-In KDC project consists of two main components within VMware Identity Manager: The Key Distribution Center (KDC) itself and a KDC Kerberos authentication adapter along with related administration functions. The KDC is based on the MIT Kerberos project that provides an embeddable Kerberos implementation including a KDC.

The VMware Built-In KDC acts as a Kerberos validation server/adapter. To do so, it stores a root CA for each tenant and the appropriate OSCP validation URL for the certificate. The OCSP URL is contained within the Certificate.

When the device begins the SSO process, it sends its authenticated user's certificate to the Built-In KDC. The Built-In KDC verifies the certificate signature and the validity of the certificate internally. It may optionally validate the certificate status over OCSP as well.

Management Console

Configuration of com.apple.sso within the AirWatch console is fairly straightforward.

Figure 2: Configuring Kerberos for an app in the AirWatch console

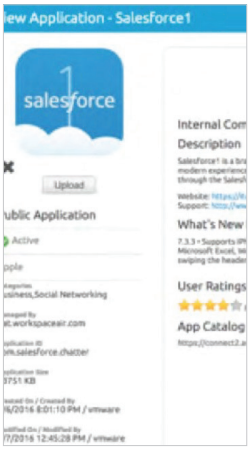


Figure 3: Information about the SalesForce1 app displayed in the AirWatch Console

Single Sign On - Detailed Flow Diagram on iOS

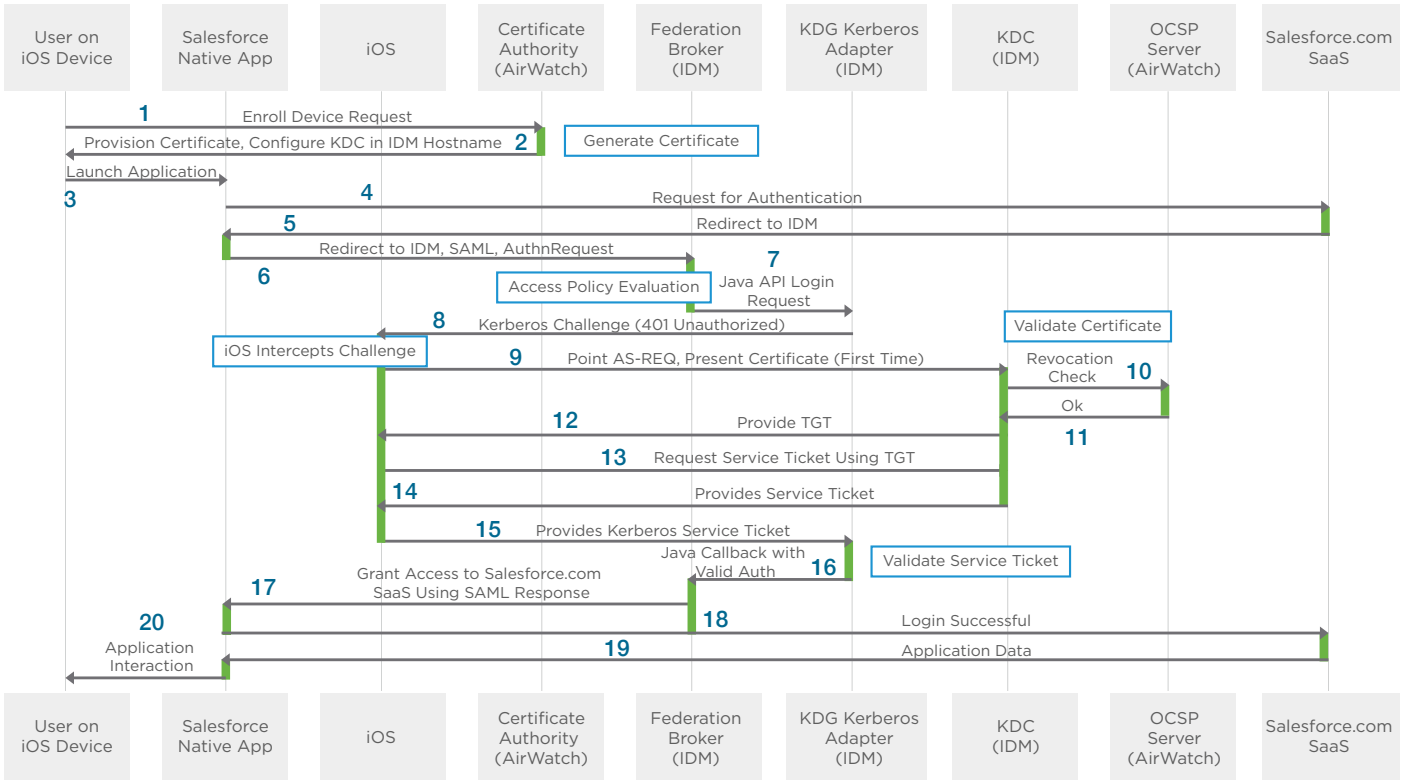
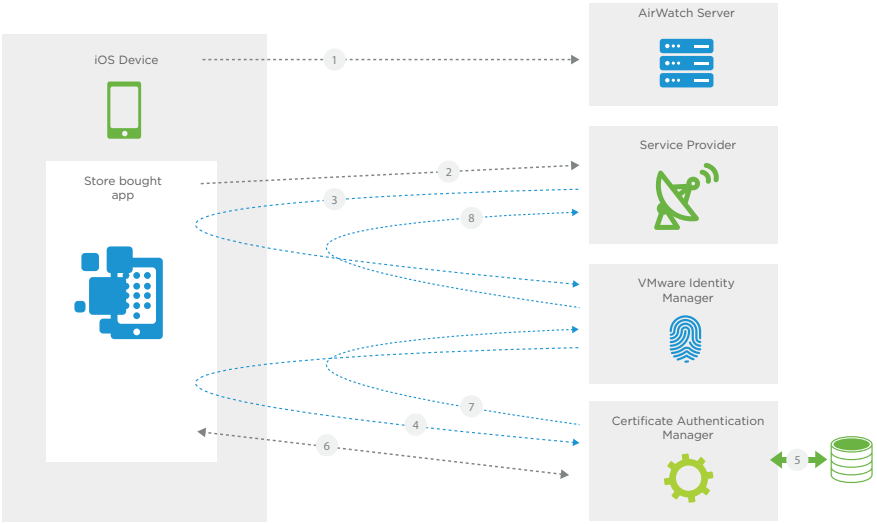


Figure 4: Detailed Flow Diagram of the Single Sign On process

1. End user enrolls device in AirWatch.
2. Device receives user certificate.
3. End user clicks on Salesforce1 (or any Public app) to review application data.
4. Application makes authentication request to Salesforce servers.
5. Response back from Salesforce server contains a SAML request.
6. Response is redirected to federation broker to properly identify user.
7. User's request is authenticated (or not) within the KDC adapter.
8. Initial request for authentication fails (as designed).
9. User's certificate is sent from iOS device to KDC for validation.
10. OCSP check is done to for additional validation.
11. Certificate is successfully checked via OCSP.
12. Ticket Granting Ticket (TGT) is issued by KDC to iOS device.
13. iOS then requests a ticket for the service (in this case, access to the VMware Identity Manager "application").
14. KDC provides a service ticket to the iOS device.
15. iOS device responds with a service ticket to ask for authentication to VMware Identity Manager.
16. KDC Kerberos approves permission for device/user to authenticate against app on iOS device. At this point VMware Identity Manager is able to check if the user is entitled to the app (in this case).
17. Ticket is translated into a valid SAML token and sent to application on iOS device stating that it is ok to authenticate.
18. Application sends authorization SAML token to servers indicating that login was successful.
19. Returns user data to application.
20. User is able to use application.

Single Sign-On on Windows 10

Microsoft offers a way to provision certificates on the device during MDM enrollment, and thereafter offers features in the operating system which allow it to intervene in app traffic and provide authentication or SSO services.

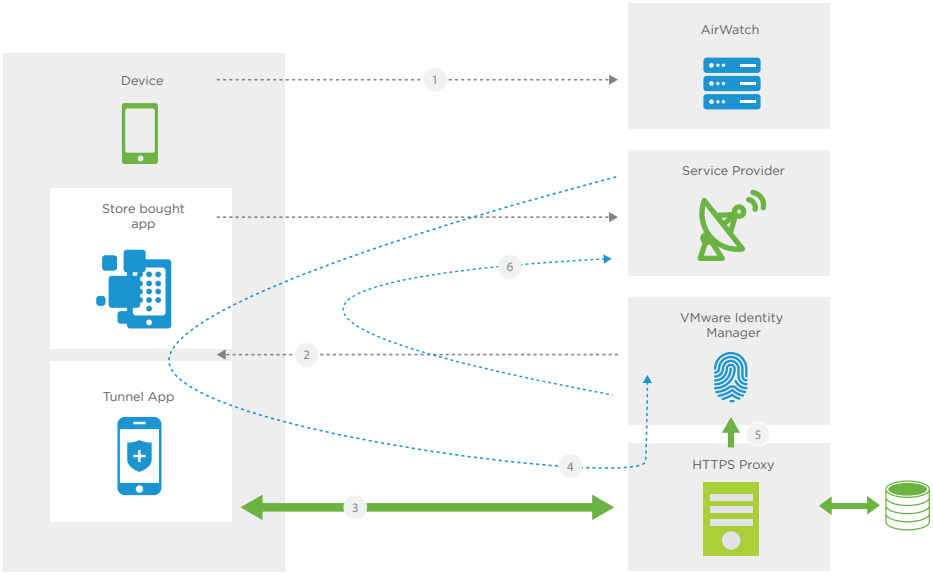


1. After acquiring the Windows 10 device, employee goes through the enrollment process with an server. The enrollment process is 'enabled' by the employee authenticating himself to his employer and in the end, results on an enterprise trusted iOS device. At the end of the enrollment process, the device has:
2. A private and public key pair for the device, with the public key being stored on the server for trust purposes. This allows the server trust the device every time the device "checks-in". iOS does not allow access to the private key. Only it can access it. The public part of this same certificate identifies the user as well although not in the subject name field. So primarily it is a "device" certificate.
3. Now employee taps on an app, which he obtained from the App Store. That app now needs to understand how to authenticate this user. For this purpose, the app invokes what is known as "Tenant Discovery". It (with the help of it's backend server, it's cloud based server counterpart) challenges the user to tell his email address, or in some way identify his company, which the employee does. If the server can recognize the company as a customer, it also has some settings to decide how to authenticate the user for it's customer. In this case, that answer should be Identity Manager, and authentication method is SAML.
4. So the Service Provider server starts the SAML process – it sends back a SAML request in a URL which the app HTTP redirects to the address (maybe in the cloud, maybe on prem).

5. VIDM will examine the request and apply access policies to it. The two parameters it uses are:
   a. What type of end point is the traffic coming from (also known as User-Agent) and,
   b. The network range of the end point. Both these are included in the HTTP Header and cannot be falsified.
6. Access policies determine how the authentication needs to happen. If the SSO is enabled, then VIDM will kick off the SSO process (i.e; it will redirect to the CAS component). If SSO is not enabled then VIDM will challenge the user for credentials. In the current document we will assume that SSO is configured.
7. Step 7 occures in 3 steps:
   a. The CAS component will first establish a one sided SSL with the app – one sided means that the app trusts the CAS but the CAS does not trust the app. CAS will obtain the root CA certificate from it's internal database so that it can validate client certificate presented by the app in the next step.
   b. Now the CAS component tries to establish a new SSL connection with client certificate required, so that the app submits it's certificate and CAS is able to read and verify it. At this stage the CAS is satisfied with the content of the certificate (Expiry, revocation), and constructs a SAML token to represent this user (it obtains user name from the certificate).
   c. It signs the Token and redirects it back to VIDM.
8. VIDM will now more or less transparently passes on the same token to Service Provider. We say "more or less transparently" because it may have to sign it again depending on it's contract with the Service Provider. The Service Provider will consume that SAML response and if trust is already established between it and VIDM, it will be able to validate that SAML and grant access to the user to it's app.

Certificate Authentication Manager

The Certificate Authentication Manager is available in the VMWare Cloud for all customers (including On Prem VIDM customers) to leverage. If some customers would like to install the Certificate Authentication Manager on premises, that option is also available, with the requirement that it should be available directly to the device (which means available on the internet).



Single Sign On on Android

The Android platform unfortunately does not offer a way for MDM vendors to play with a certificate store on the operating system like iOS and Windows does. However, Android does offer the per app VPN facility to an app and this offers VMware an opportunity to provide SSO on Android devices. A key component of this is the AirWatch Tunnel App.

The Tunnel App has two functions –

1. VPN Client: This is the already existing role that the Tunnel App plays. As VPN Client, Tunnel App has the job to provide secure access to resources behind the firewall in partnership with the Tunnel Server. As part of this functionality, which is already shipping in AirWatch 8.3 and before, the Tunnel App has access to a client certificate provisioned to it by AirWatch server during enrollment.
2. SSO Facilitator: This is the new additional role that the Tunnel App takes in AirWatch 8.4 and later - as the SSO Facilitator on Android. In this role, the Tunnel App will intercept any traffic between an app (that needs Tunnel App for VPN) and Identity Manager Server, and helps VIDM maintain a "session" for the device and user combination according to policy.

The process of Android SSO is as follows:

1. As a part of enrollment, AirWatch will let the administrator do a few things:
   a. Define a per app VPN (or a device wide VPN), with user certificate based authentication – for all the apps that need SSO.

   b. Push the Tunnel App to the device. The tunnel app is the VPN Client in this case. AirWatch will make the user certificate available to the Tunnel App.

2. Because the Tunnel App is the designated VPN Client for the store bought app, it has the privileged position to inspect and intervene in the traffic going in and out of the app. It's "normal" job is to just proxy all traffic to the Tunnel Server, however anytime the app tries to communicate with VIDM, the Tunnel App wears it's SSO hat. So if the user taps on the app, the app will first go to the Service Provider in the cloud. If the Service Provider (via a browser redirect) asks for a SAML request, then the Tunnel App interrupts the flow momentarily.
3. It establishes a mutually authenticated SSL session with the HTTPS proxy. Normally an SSL session helps prove the identity of the server (not client), but in this case the HTTPS proxy is configured to "require client certificate" because of which the SSL session is mutually authenticated. The HTTPS proxy gains access to the client certificate presented by the Tunnel Client, verifies with it's preloaded Root CA Certificate chain, and allows the SSL session to be established.
4. Now the Tunnel Client will "allow" the traffic continue on, and the app establishes an SSL with VIDM inside the earlier established SSL Session. The App "thinks" it is talking directly to VIDM, however the Tunnel Client is pushing the communication through it's own SSL session (which proves the device and user identity) to HTTPS proxy. Now VIDM will normally want to challenge the user for some type of credentials, but it knows that this traffic is coming from the HTTPS proxy, so it knows that this is an SSO situation and challenges the HTTPS proxy for the certificate.
5. Using a pre-established (and proprietary) contract, upon request, HTTPS proxy will present the certificate to VIDM which VIDM will use as representative of a session.
6. If VIDM is satisfied with the certificate, it will construct a SAML and send it back via the usual means, i.e; browser redirect to the Service Provider. The Service Provider is able to trust the SAML assertion via a pre-established trust between VIDM and Service Provider. If Service Provider is satisfied, it will grant access to the user.

The key to this process is the mutually authenticated SSL Session between Tunnel App and HTTPS proxy. SSO exists as long as this session exists. If in case that session is destroyed (generally in case of network changes) or Time to Live configuration, then the SSL process is repeated (Tunnel App has to again prove it's identity to HTTPS Proxy) and VIDM will again challenge the HTTPS proxy for the certificate.

If the Tunnel App is removed from the device for some reason, then VIDM will challenge the end user for credentials, thus falling back to the non-SSO process.

HTTPS Proxy

The HTTPS proxy is available in the Cloud for all customers (including VIDM customers) to leverage. If some customers would like to install the HTTPS proxy, that option is also available.

Security and Single Sign On

All the features offered by Vmware around single sign-on for Native Apps are based on features already offered by the respective operating systems. However, the important point is that Vmware is able to bring a single pane of glass administration experience because of which the IT department need not know or understand how the operating systems are different from each other. Vmware "compensates" for some missing functionality in some cases (for example Android), bridges some "old" functionality in some cases (for example iOS and Kerberos) and simply provides support in some cases (for example Windows 10).

If a company wants to make use of com.apple.sso profile without the Built-In KDC implementation from AirWatch, they must have their end users establish a VPN tunnel with every time they use a public app. Even if the end users are willing to do that, it is still hard for companies security policies to allow this access to the KDC.

Further, with most cloud-based services for enterprises, it is likely that the server does not support Kerberos; i.e., does not accept Kerberos Tickets. In such cases, the implementation of KDC (Built-In KDC) by Identity Manager is beneficial.

To take advantage of the Built-In KDC implementation of Identity Manager, there is nothing customers have to do in their infrastructure other than upgrade to VMware Identity Manager version 2.6. If they are customers of Identity Manager hosted, they do not have to upgrade as this will be taken care of by the VMware operations teams.

End users who are already enrolled in AirWatch can stay enrolled, and the administrator's definition of com.apple.sso will be provisioned to their device when their device checks in. AirWatch has the support for com.apple.sso, so end users are in for a pleasant surprise when they get upgraded to Identity Manager 2.6.

Lastly, we should not overlook the importance of certificate based authentication. It is the best form of authentication known to man, as it offers seamless experience to the end user (no challenge), is considered a strong form of authentication and at the same time allows the to be assured that they can revoke the certificate at any time. While SSO is the main value of this feature and that value is useful for making mobility projects successful in companies, the nice thing is that this is the best way of securing data in device and in transit.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com