mi·token
Product Overview

1. Introduction

Mi-Token has been delivering authentication solutions to financial, enterprise and government institutions for almost ten years. Mi-Token was designed from the ground-up to provide a seamless experience for administrators and users, while reducing the cost and complexity of Two Factor Authentication (2FA) security.

Mi-Token provides multi-factor authentication in a variety of security scenarios including:
•VPN remote access (including Cisco AnyConnect, Juniper SSL VPN, Citrix NetScaler and NetMotion, etc)
•Access to on premise applications (including Exchange, Remote Desktop, SharePoint and custom web-based portals)
•Single Sign On (SSO) for cloud based applications (such as Office 365, Salesforce or any other SAML-enabled solution)
•Any RADIUS complaint infrastructure (ranging from cloud based Amazon AWS Console to on premise managed Ethernet router).

2. Why Mi-Token ?

2.1 Security - No Custom Authentication Server

Mi-Token makes authentication decisions without a dedicated Authentication Server (AS) commonly responsible for this functionality. The AS usually sits behind a firewall and validates or rejects authentication requests based on user credentials including one time password (OTP). Firewalls are commonplace and bear the brunt of external attacks, yet the number of system breaches has not subsided despite the implementation of firewalls in most environments. This makes robust security of the components behind the firewall paramount.

Mi-Token avoids deploying a dedicated AS ensuring that when you decide to implement 2FA, security is indeed increased rather than is influenced by having an AS which might not have been sufficiently hardened or is being patched on a schedule different from the operating system security updates.

2.2 High Availability - No Single Point of Failure

When Mi-Token installer runs, it creates a local instance of a distributed database which is not unlike Active Directory (AD) database used by Microsoft domain controllers. It's a lightweight version of AD database called AD LDS. After the installer finishes, (which typically takes about a minute), the newly created instance of AD LDS database immediately starts replicating in a bidirectional manner with other database instances used by Mi-Token components installed on other servers. As a result neither the Mi-Token software nor the database underpinning it represents a single point of failure.

2.3 Speed and Reliability - No Remote Database Access

When one logs into a Windows domain their password is verified by a domain controller against an internal AD database. For the users of the same domain this database resides on the domain controller itself. If it was remote then AD would have only a fraction of its speed and reliability. When one logs into a system protected by 2FA, their one-time password (OTP) is verified against a database which then needs to be updated to prevent a replay attack when the same OTP is used twice. In the Mi-Token case this database is always local, no authentication decisions are ever made using a remote database.

3. Functionality

3.1 VPN Access

Mi-Token offers a Plugin for Microsoft Network Policy Server (NPS) which evaluates an OTP supplied during the login process and helps NPS to make authentication decisions. As a result, Mi-Token doesn't employ a custom AS which a firewall or VPN appliance has to be connected to. The AS which makes decisions to accept or reject users needs to be hardened against hackers and this requires an extraordinary amount of resources with rather prohibitive cost.

In case of Mi-Token the NPS already hardened by Microsoft plays the AS role. The NPS consults the Mi-Token Plugin while making the authentication decisions. If NPS decides it has come under attack then it won't consult the Mi-Token Plugin which derives security benefit from running inside NPS.

Another consequence of this approach is the simplicity of Mi-Token integration into an existing VPN solution. Depending on VPN setup users can either concatenate their password and OTP and use the concatenation instead of the password or enter OTP at a separate 'secondary credential' prompt displayed by VPN client. But in either case VPN clients and servers are not aware of of the existence of Mi-Token. VPN servers (and in some special cases VPN clients) only know they communicate with NPS which doesn't let any other system component know it has loaded the Mi-Token Plugin. Therefore Mi-Token integration with VPN servers and clients becomes a moot point – no specific Mi-Token documents to read on this topic, no configuration tweaks and no compatibility or versioning issues as the AS is Microsoft's Network Policy Server, not a standalone AS provided by Mi-Token. This is better for Mi-Token and better for you – the customer.

The simplicity of integration is complimented by the simplicity of installation. In order to enable Mi-Token for VPN, only two installers need to be executed: one for the NPS Plugin and another for Active Directory User Interface (AD UI) used to manage Mi-Token. The first installer usually takes approximately a minute or so to run, doesn't require any data to be entered. It will create a local instance of AD LDS database to hold token data. The second installer takes less than a minute to run, also doesn't require any input.

Subsequently to that the Plugin can be optionally installed on another NPS server for redundancy/fault tolerance, it will create a local AD LDS database as well and the two databases will start replicating to each other as soon as the installer finishes. The Plugin can be installed on any number of NPS servers with all the created databases replicating to each other. The AD UI can additionally be installed on any domain server or workstation. There are no post-install configuration adjustments needed for the Plugin, AD UI and AD LDS database.

3.2 AD LDS database

It is worth mentioning that most security guides consider a firewall to be the first line of defense only. Firewalls are affordable nowadays and used very widely yet the number of successful hacking attempts doesn't dwindle to zero. So the security related software components behind the firewall play a crucial role and this certainly includes the database.

Mi-Token doesn't use a conventional SQL compliant RDBMS for authentication decision making. It uses an AD LDS database which is powered by the same technology as Active Directory databases hosted by every domain controller. The AD/AD LDS databases are substantially more secure, there are no recent reports of a hacker being able to retrieve or change the data in an AD LDS database. This compares well to RDBMS where such events are more frequent with hacking techniques like SQL injection etc. being used to gain unauthorized access to the data.

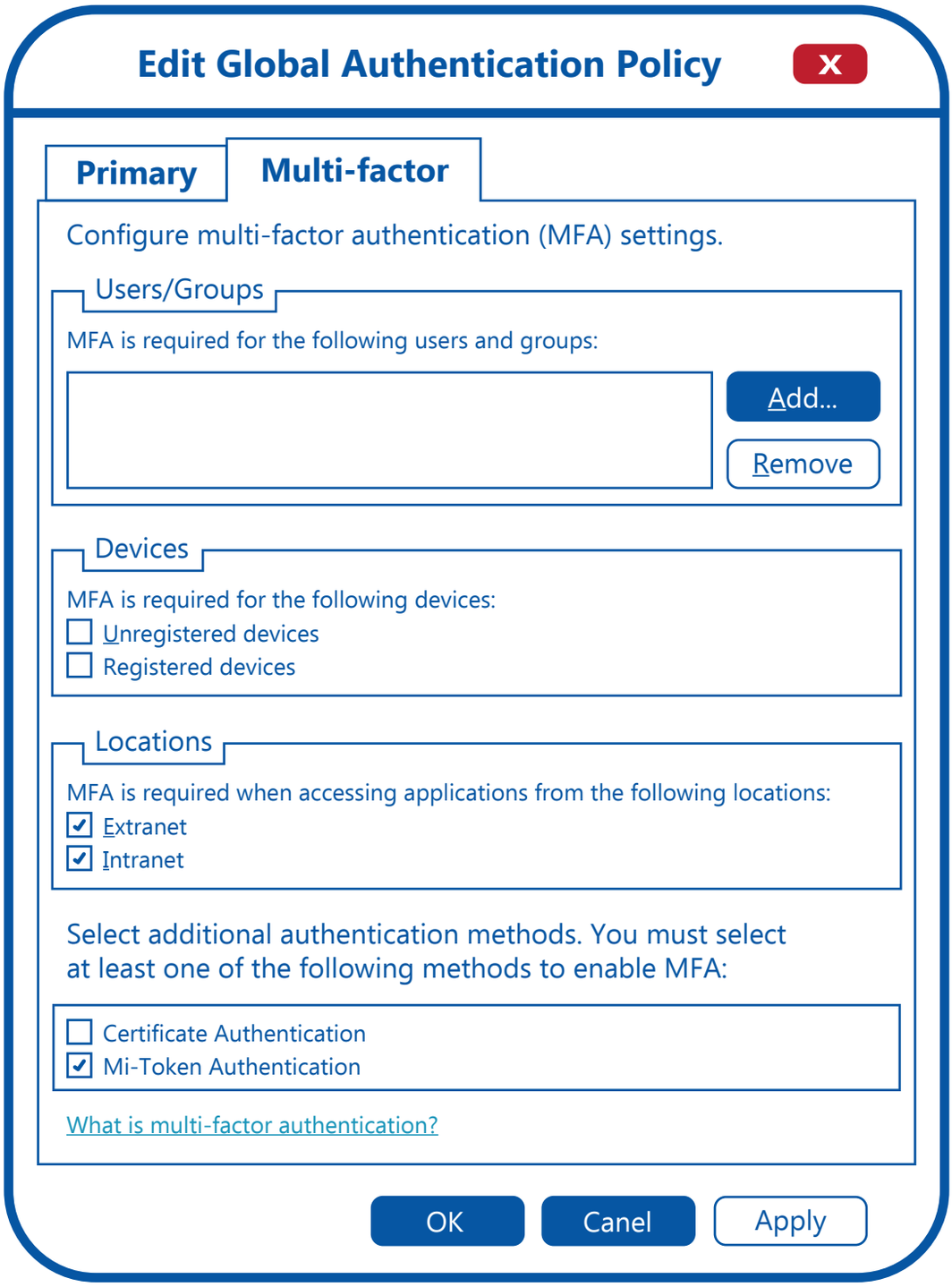
As already noted the built-in ability of AD LDS databases to replicate makes for High Availability. With no installation time configuration adjustments and no post-install tweaks, Mi-Token AD LDS databases compare favourably to its SQL compliant counterparts. For a typical RDBMS implementing replication between two database instances is an advanced task which requires considerable skills and time, this task becomes arduous if the number of database instances grows and they all need to replicate bi-directionally to each other.

3.3 ADFS Plugin

The integration between ADFS and Mi-Token is achieved by using the ADFS Plugin. It requires ADFS 3.0 which comes with Windows Server 2012 R2. After the ADFS Plugin is installed the following additional "Mi-Token Authentication" entry appears in the ADFS Management Console:

Ticking the "Extranet" and "Intranet" checkboxes in the Global Policy enforces Mi-Token Authentication for all applications configured to use ADFS for authentication. Alternatively these checkboxes can be left unticked in which case the similar checkboxes in the individual application policies (such as Exchange OWA policy or Exchange ECP policy) are not grayed/-disabled so it's possible to apply different settings to different policies. For example enable Mi-Token 2FA only for external (e.g. coming via AFDS proxy) OWA users, disable it for internal OWA users and enforce it for all ECP users.

Mi-Token ADFS Plugin communicates with another Mi-Token component called API Service which is installed on one or more central servers. It can be the same server where the ADFS server role is enabled. The ADFS Plugin needs to be able to communicate with the API Service over HTTPS, port 443. The ADFS Plugin automatically switches to another API Service instance (and its local AD LDS database) in case the currently used instance becomes unavailable. This helps to achieve true fault tolerance with neither the API Service nor its database representing a single point of failure.



In order to use Mi-Token ADFS Plugin only two additional installers are needed for a minimal Mi-Token installation: API Service and AD UI. In more feature-rich Mi-Token installations, other components (like Mi-Token Reporting and the Self-Provisioning Website) can be installed. The licensing cost does not depend on the number of computers where Mi-Token is installed or on the type of Mi-Token components used.

If Mi-Token ADFS Plugin is installed in a domain where Mi-Token was not previously used, then the API Service installer creates a brand new instance of AD LDS database thus making this machine the primary Mi-Token server. Later the same installer can be executed on other server(s) creating replica AD LDS instance(s).

In cases when there is an existing Mi-Token deployment, for example if Mi-Token NPS Plugin is already installed on 2 NPS servers (with one being the primary and the other being a replica), then this configuration can be preserved with the API Service instances installed on both NPS servers. With this arrangement the API Service will share the local AD LDS database with the NPS Plugin so no more AD LDS instances are created. Alternatively the two API Service instances could be installed on other servers as replicas increasing the replica count from one to three.

3.4 Reporting

Mi-Token Reporting is an optional component. It includes Graphical and Text Reporting. The former provides graphs reflecting high-level view of the system e.g. what was the historical up-time of each NPS server and who are the users accessing Mi-Token more frequently than others. The latter provides in-depths view into the system e.g. what property of which token has been modified and when did it happen.

Reporting requires SQL Server 2008-2014 (any edition including Express) and stores very detailed history of Mi-Token usage thus making it available for future analysis, audits, advanced data mining etc. The database is not used by the authentication decision making components which do not even know it exists.