AuthLite Administrator's Manual
for software revision 2.3

(The following graphics include screen shots from Microsoft® Windows and other properties of Microsoft Corp. and are included here for instructive use. Some images illustrate AuthLite, which is the property of AuthLite LLC.)

Table of Contents

OTP Token Types

YubiKey:

AuthLite uses the [YubiKey](#) from [Yubico](#) Inc. as an inexpensive, robust one-time-code generating device. YubiKeys have no display or battery, no moving parts, and are waterproof and virtually indestructible even in rugged environments. They draw power from the USB port and are treated as an HID keyboard device so they work without special drivers on all platforms.

The YubiKey platform also contains extra cryptographic security features that allows AuthLite to securely authenticate users to offline (disconnected) domain workstations.

AuthLite supports the YubiKey v4, Edge (v4 lite), Nano, and NEO. AuthLite cannot use the blue "U2F Security Key" because it does not support OTP or challenge/response modes.

OATH Token:

AuthLite is also compatible with OATH Time-based One-time passcodes (TOTP) generated by smart-phone soft-token apps such as the free cross-platform Google Authenticator app. This is useful for cases where carrying a hardware token is undesirable or plugging in a USB device is impossible (such as authenticating to a 2-factor system from a smart phone).

Requirements and platforms

AuthLite is licensed on a per-user basis, so you don't have to worry about counting the number of servers or workstations. Here is some high level guidance on where software needs to be installed, and what is supported. For scenarios not covered here, please [contact us](#) for assistance.

Note: 32 and 64 bit platforms are supported.

Workstations:

Unless cached logon is disabled by group policy, you should install AuthLite software on any workstations that will have AuthLite users logging on. Supported platforms:

- Windows Vista
- Windows 7
- Windows 8
- Windows 10

Domain Controllers

AuthLite software **must** be installed on at least one Domain Controller in your organization. AuthLite uses an Application Partition to store and distribute its user data.

You **must** install the software on every DC that could be used to authenticate AuthLite users.[1]

---

1 If you have multiple AD sites and your servers and clients have their site membership defined correctly, then you could use this mechanism to limit AuthLite installs to only one, or a subset of your sites. If you do not have sites defined, then any client could choose any DC for authentication, therefore you need AuthLite on

The data partition should be [replicated](#) as needed. By default each DC where you install AuthLite will host a replica of the data partition.

Supported platforms:

- Windows 2008 R2
- Windows 2012 (R1/R2)
- Windows 2016 (presently "Additional LSA Protection" is not supported)
- Windows 2019 (presently "Additional LSA Protection" is not supported)

Read-only DCs

AuthLite can function on RODCs, but because it uses one-time passcodes, at each successful logon the RODC must update OTP counter attributes on a writeable DC. The necessary configuration and permissions to allow this RODC → WDC access are applied automatically.

"Core" (command-line mode) DCs

AuthLite supports server Core. Please see [this KB article](#).

RADIUS servers

To use RADIUS with AuthLite, you need to install the Microsoft IAS/NPS service on one or more domain servers, install AuthLite, and activate the AuthLite IAS/NPS plug-in. Supported platforms:

- Windows 2008 R2
- Windows 2012 (R1/R2)
- Windows 2016 (presently "Additional LSA Protection" is not supported)
- Windows 2019 (presently "Additional LSA Protection" is not supported)

Terminal servers

If you want AuthLite users to connect to RDG/TSG or standalone terminal servers, the AuthLite software should be installed on each server. This allows the Windows NTLM authentication to handle AuthLite OTP entries that will be sent in the "username" field of the remote desktop software. Supported platforms:

- Windows 2008 R2
- Windows 2012 (R1/R2)
- Windows 2016 (presently "Additional LSA Protection" is not supported)
- Windows 2019 (presently "Additional LSA Protection" is not supported)

Exchange client access servers

If you use Exchange 2013 or later, or 2010 with Outlook Anywhere, your Exchange servers that host IIS, RPC/HTTP, and the Exchange components will need the AuthLite software installed. This allows the Windows NTLM authentication to handle AuthLite OTP entries that

every DC. Also, see security notes in [Requiring Two-factor Authentication](#).

will be sent in the "username" field of Outlook. Exchange 2013 and later works best, presently. The 2010 version can be made to work but requires extra configuration. Supported platforms:

- Windows 2008 R2
- Windows 2012 (R1/R2)
- Windows 2016 (presently "Additional LSA Protection" is not supported)
- Windows 2019 (presently "Additional LSA Protection" is not supported)

Other domain member machines

For other systems, AuthLite software may need to be installed if AuthLite users will be logging on interactively to the system, but generally not otherwise. For example you do *not* need to install software on your back-end Exchange servers, because the actual logon is performed elsewhere.

Supported platforms:

- Windows 2008 R2
- Windows 2012 (R1/R2)
- Windows 2016 (presently "Additional LSA Protection" is not supported)
- Windows 2019 (presently "Additional LSA Protection" is not supported)

Standalone system with local users

At this time, AuthLite version 2 can only be used on Domain member machines. AuthLite v1.2 can be installed and used for standalone servers/workstations up to 2008R2 and Windows 7.

Installation

Prerequisites

- Microsoft .NET framework version 4 or later

- All installs require administrator permissions

- The first Domain Controller install should be run as a user with permission to add/modify an Active Directory Application Partition (i.e. Enterprise Admin) and add properties to the AD Schema (i.e. a Schema Admin). Subsequent DC installs should just require local administrator on the DC.

Version Interoperability

If you intend to use AuthLite on any workstations, ensure that the domain controllers are all running a version at least as "new" as the workstations run. That way, new features that may be needed on the workstation side will be available on the servers. For example 2.3 adds some features that won't work correctly if the servers are 2.3 or older.

Installers

The same installation software is used for both workstations and servers:

- AuthLite_installer_x86.msi for 32-bit platforms

- AuthLite_installer_x64.msi for 64-bit platforms

This installer is designed to be run through the user interface, but may also be installed unattended through group policy.

Post-install

A system reboot is required to load (or update, or unload) AuthLite infrastructure components. After the Finish screen, the installer will remind you a reboot is needed.

Before you can start using AuthLite, you need to enter a license number (either an evaluation key or a purchased license). The License Information screen of the AuthLite Configuration program is used to accomplish this process.

Upgrading

Most version updates can install "over" the old version correctly. If this is not possible then the installer will instruct you how to properly upgrade. After installing the new version, you must reboot the system.

Uninstalling

Uninstalling AuthLite software from all systems in your enterprise will effectively revert everyone to normal one-factor password-only logins. Please note that any offline AuthLite users will need to reconnect their workstations to the Domain LAN at least once in order to re-authenticate, because their cached credentials will be invalid after the AuthLite software is removed.

If you have users whose sessions are being granted dynamic groups by AuthLite, they will no longer get their groups swapped when they log on. For example:

- If you are using the "AuthLite DA → Domain Admins" configuration, users in "AuthLite DA" will no longer be able to manage the domain when they log on.

- If you have enacted group policies that deny logon when the "AuthLite 1F Tag" is detected, any users who are represented in that group will be unable to log on.

So, make sure to account for whatever permissions and groups you have changed when removing AuthLite.

To prevent accidental data loss, the AuthLite uninstaller never removes the AD data partition. If you are certain you no longer need the partition, you can remove it manually using Microsoft's **ntdsutil** command.

The schema elements added for the AuthLite partition cannot be deleted (by design of Microsoft's LDAP service model) but they are only used in the AuthLite partition, so they will have zero impact on your domain, replication, or anything else if you have removed the AuthLite data partition.

Licensing AuthLite
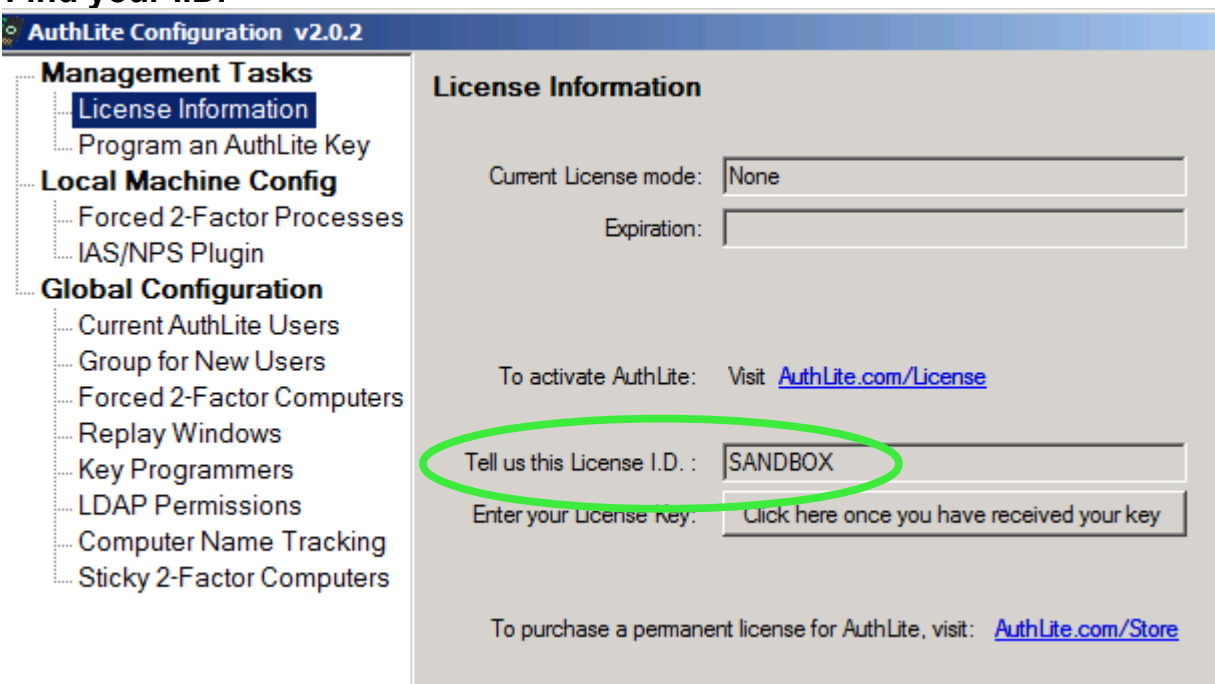
After installing AuthLite, be sure to reboot first before setting up the license.

Log in as a domain administrator to a DC that has AuthLite installed, and follow the license procedure below. **You only need to set up the license once**, and it will be automatically used by all servers and workstations in your domain. Due to AD replication settings the license value may not immediately propagate between all servers.

License procedure

We try to make this process as friendly as possible. Our support staff will respond to your request personally, and we will be available to assist you with any problems.

Find your I.D.

From the Start menu, launch the AuthLite Configuration application. The License I.D. is shown in the dialog. It is the NETBIOS name of your domain. You will need this value for the next step. You must tell us the exact name shown in the dialog, or the key we generate for you won't be recognized by the software.

Request a key

- Go to AuthLite.com/License and enter your I.D. found above.

- If you are **evaluating** you can enter "Eval" in the "Order number" field.

- Enter your contact information so our support staff can send your key.

- After you submit the form you should immediately receive an email confirming the request, and a URL that allows you to check the status of your ticket on the web.

- We will create and send your key as soon as possible.

Enter your key

- You can only enter the license key on a Domain Controller. On other systems this field will display as read-only.
- Type or paste in the license key you receive into the configuration dialog and click "Apply"

- The "License mode" and "expiration" fields should update to reflect the status of your license.

License Expiration

If your evaluation period expires, or the software is used for a higher number of users than it is licensed for, then certain functions will be disabled until a valid license is entered.

Disabled features:

- Importing or associating new tokens

- Reassigning tokens to different users

Features that stay enabled even after the license is invalid/expired:

- Authenticating with existing tokens and users
- All security and logging features

Identifying AuthLite Users

AuthLite User Groups

Quick start

Open "AD Users and Computers" and select View->Advanced Features.

Create three global groups anywhere in your AD domain:

- AuthLite Users
- AuthLite 1F Tag
- AuthLite 2F Tag

In *AuthLite Users* properties:

- This is the group where your users will go to signify they will be treated as AuthLite users.
- In the Security tab, add "Network Service" and select "Write" in the "Allow" column. This will allow the AuthLite service on your domain controllers to write changes to this user group automatically.

In *AuthLite 1F Tag* properties:

- In the description, add "Membership is dynamically assigned; do not add users!"
- Add *AuthLite Users* as a member
- In the Security tab, find the *Domain Admins* row and select the checkbox "Write" in the "Deny" column (*not the allow column!*) This will put a "speed bump" in place against anyone accidentally misunderstanding and trying to add users to this tag group.

In *AuthLite 2F Tag* properties:

- In the description, add "Membership is dynamically assigned; do not add users!"
- Leave membership list empty
- In the Security tab, find the *Domain Admins* row and select the checkbox "Write" in the "Deny" column (*not the allow column!*) This will put a "speed bump" in place against anyone accidentally misunderstanding and trying to add users to this tag group.

In the *AuthLite Configuration* app, go to the *AuthLite User Group Pairs* dialog and add the pair containing *AuthLite 1F Tag* and *AuthLite 2F Tag*.

In the *Group for New Users* dialog, add the *AuthLite Users* group.

Permission checking

AuthLite uses Active Directory group memberships to discern which users should be considered as AuthLite users. These groups are used to decide which users should be required to provide 2-factor authentication when logging on at the computers and processes you have selected to [Require Two-factor Authentication](#).

Users that are not in any AuthLite group will not ever be affected by AuthLite.

The AuthLite User Group Pairs dialog allows you to select the groups that will be used to perform this permission check. For example when you add a group called "AuthLite 1F Tag" to this dialog, then any user who has membership in that group (directly or nested) will be considered an AuthLite user for the purpose of permission checks. In the Quick Start example previously, the "AuthLite Users" group was nested in the "AuthLite 1F Tag" group, so all users in the "AuthLite Users" group will be considered by AuthLite. (This is the basis for naming the groups as chosen in the example, but you can select different names if you prefer).

A user account NOT represented by any 1-Factor Tag group will never have any restrictions placed on it by AuthLite. Thus, it is very important to make sure your users are members of the correct groups. The token provisioning interfaces all provide a way to automate adding users to the "New Users" Group. In the Quick Start above we nested this into a1-Factor Tag group, ergo as soon as you add a user they will be considered an AuthLite user immediately.

Groups chosen **must be Global Groups**, and in general they should not be nested into any other AD group, apart from "Domain Local" and "Builtin" groups.

These limitations are imposed by the technical method AuthLite uses to do group checking. For more information on these limitations, please see this tutorial video series and if necessary contact customer service.

Two-factor Session tagging

This dialog also tells AuthLite how to dynamically switch group membership of AuthLite users' sessions depending on how they authenticate. Consider the following use case:

Normally, every Kerberos ticket-granting ticket is essentially the same. Once a user authenticates, there is no way to tag that session's level of security. Consider a user who is permitted to authenticate with 1-factor on computer A, but you wish to protect resources on server B from 1-factor access by that user. Since the authentication already happened at logon time, and the user has a Kerberos TGT, there is no additional way to check or protect the resources at B. Access will be granted regardless of your wishes. You'd either have to require 2-factor authentication for that user *everywhere*, or accept that it's possible to circumvent the protection of those resources.

To solve this problem, each line of the AuthLite Group dialog refers to a *pair* of AD groups:

1. The first group is called a 1-Factor Tag group. Your AuthLite users will have membership in this group naturally (by virtue of being nested through the "AuthLite Users" group). Thus any time the users log on, their session token will, by default, carry the SID for this group. If they don't log on with 2-factors, this will remain the case. However:

2. The second group is called a 2-Factor Tag group. Its membership in AD should be left **empty**; do not assign any users or nested groups to it. Any time a user logs on with two factors, AuthLite on the DC will edit the user's session token. Each instance of a 1-Factor Tag group SID will be *replaced* with its corresponding 2-Factor Tag group SID.

In other words, the user's effective group membership for this session will depend on how they authenticated. If it was a one-factor password-only logon, then their session token will reflect the default group membership: the 1-Factor Tag. This can be used by group policy and other ACLs to restrict access to AuthLite users who have not done a 2-factor logon.

But if it was a two-factor (OTP and password) logon, then the session token will be changed

to include the 2-Factor Tag group instead. Thus, any group policy or ACL designed to block the 1-Factor tag will not trigger, and the user will be allowed to continue logging on.

You can alternately use the 2-Factor Tag directly in ACLs to grant (allow) access. Keep in mind, however, that a user just needs to have membership to *any one group* in the Allow list of an ACL. Adding the 2-Factor Tag group will just grant more users rights to access the resource, it won't block anyone! So to use this as an access control method, you have to make sure your ACL only allows combinations that you actually desire to have access.

With AuthLite authentication, the actual content of the session's group SIDs will now be different depending on the level of authentication that was used. You can now assign ACLs and permissions throughout your domain that are sensitive to the level of security the authentication used. Resuming the example above, if only the 2-Factor Tag group is in server B's ACL, then only AuthLite users who have logged on with 2-factor auth will have permission to those resources.

The system AuthLite uses is similar in spirit and purpose to the [Authenticaton Mechanism Assurance](#) implemented by Microsoft in order to discriminate between smart card and password-only logons. Our approach additionally allows even AuthLite-unaware systems to be ACL'd securely against 1-factor access (because all systems can read the AD groups of the session, regardless of whether AuthLite is installed).

You can also specify *more than one pair* of groups because you may not want all users to be lumped together in the same 1-factor group. For example, you may want to allow access to a group containing 2-factor authenticated *domain admins*, but *not* 2-factor authenticated lower-level users. Having multiple group pairs allows you to define your own classes of users and apply different permissions to them, as a natural extension of how Microsoft authorization ACLs work.

AuthLite's use of group pairs to do session tagging is extremely powerful, but a little subtle to consider initially. The bottom line is that the users' effective group membership will switch around deterministically depending on how they authenticated, and you can use this knowledge to restrict access to your AD systems and resources.

Please refer to this [tutorial video series](#) for an illustrated discussion of these important topics.

Group for New Users
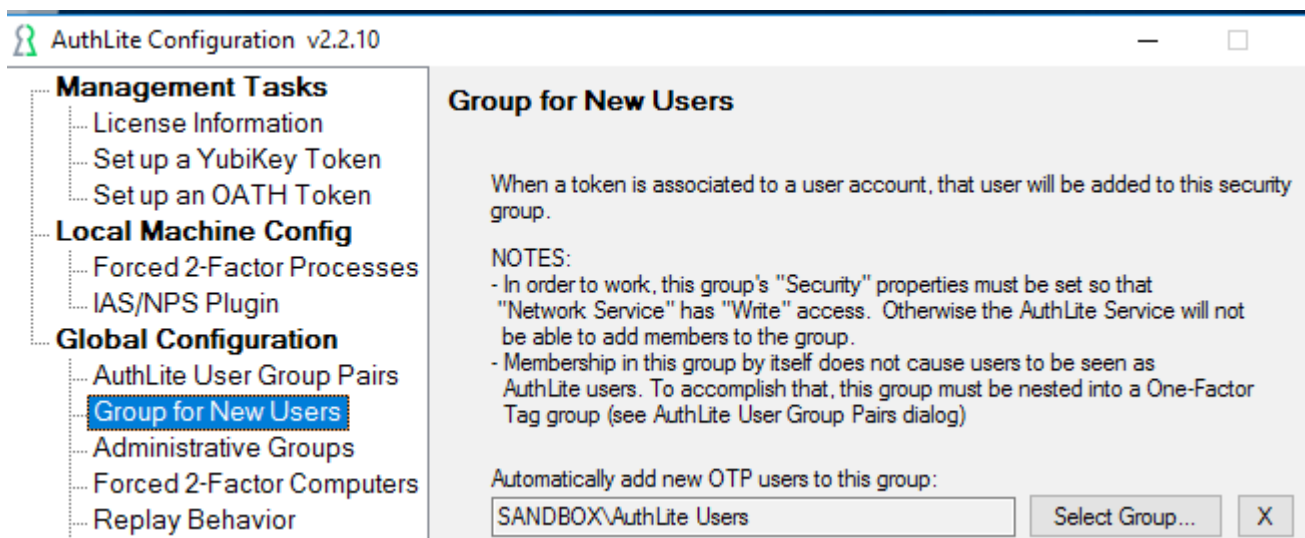
This is a security group that will be used as a simple bucket to drop new users into when they get a token assigned (via an optional checkbox in the UI). By default, this group doesn't grant any special significance to the user. If you are administratively provisioning your users, they may not have possession of the token right away. In this case you would not want to enforce 2-factor restrictions on them immediately. But this group still allows you to collect the set of users who have at least one token assigned. Later on, you could add this to the "AuthLite 1F Tag" group to begin enforcing on the users all at once.

However, if you set up your groups as in the Quick start, then these users will automatically get treated as AuthLite users right away after token assignment (due to this group being nested into a 1F tag group, which is the "real" test of AuthLite-user-ness but somewhat less obvious to grasp).

Provisioning YubiKeys
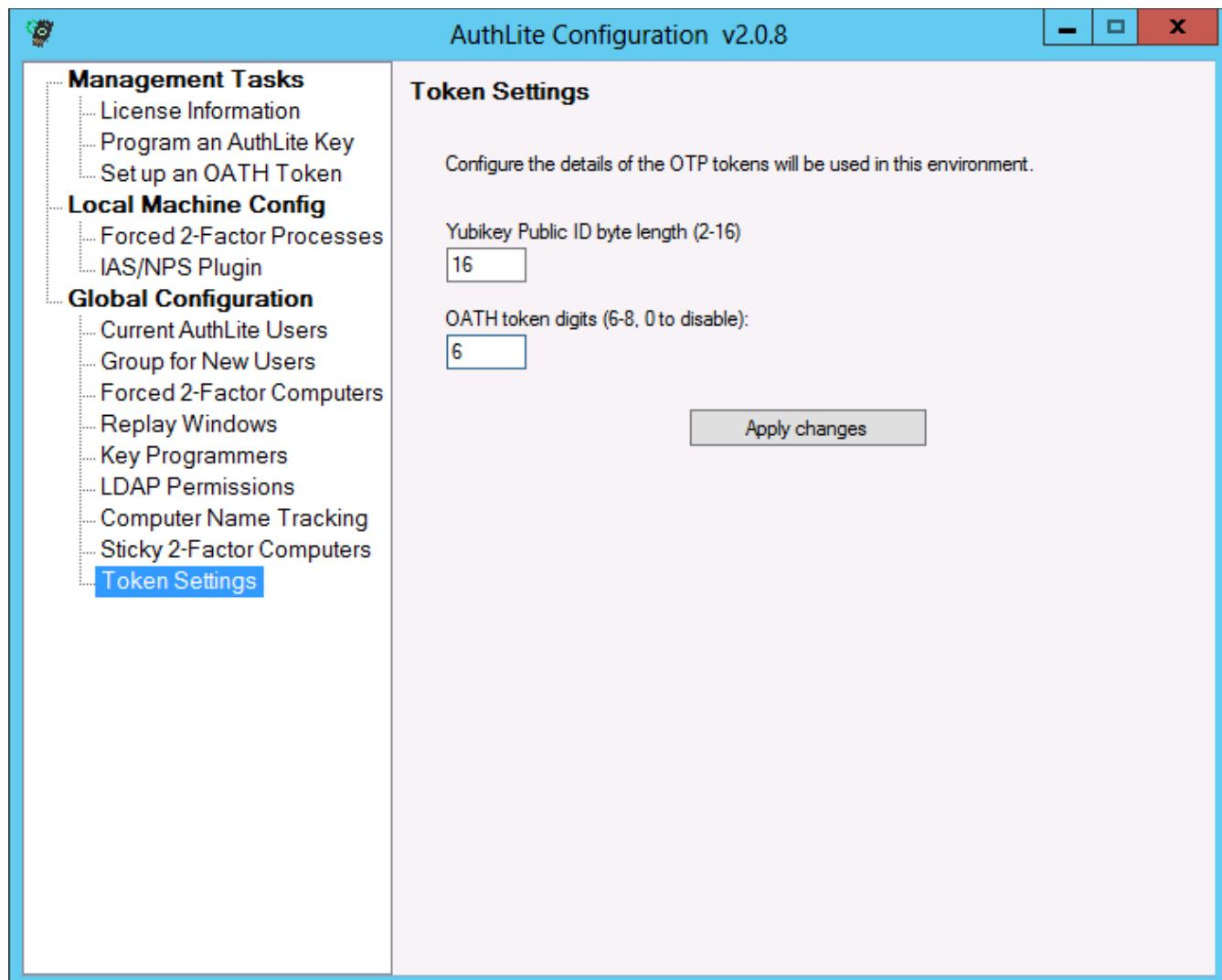
Token Settings

Historically, AuthLite-programmed YubiKeys have used the longest possible public ID (16 bytes), to eliminate the chance of an attacker guessing the ID and also because the key's record in the data store is encrypted by the hash of the public ID for slightly greater protection.

As of AuthLite version 2, it is possible to specify that your domain will use shorter public IDs. Note that the number of possible YubiKeys in your environment will be limited to $2 \wedge (8 \times$ byte length). So for example using a 2-byte public ID limits you to $2 \wedge (8 \times 2) = 65536$ possible keys.

If you use a short Public ID length, then you should program your keys with sequential IDs instead of random ones, to avoid collisions.

If you plan to use OATH Tokens (e.g. Google Authenticator) then set "OATH token digits" to "6". If you are using physical tokens that have an 8-digit readout, set it to "8". This setting applies domain-wide, so you must select OATH tokens that all have compatible length OTPs.

One-off YubiKey provisioning from the console

If you have physical console access (not a remote session or virtual console) to a domain member server or workstation with AuthLite software installed, then you can provision YubiKeys one at a time with this simple process. Note that this method always uses randomly generated Public IDs, which is OK for long IDs but if you have shortened your Token Settings you should program using the external app instead so you can use sequential IDs. See Bulk programming from a standalone workstation.

Prerequisites

- AuthLite software installed on domain controllers and (if you are going to administer from a workstation) on that workstation machine.

- Valid license or evaluation key entered.

- A blank YubiKey, or a formerly provisioned key that you want to **erase** and reprogram. (Note: The blue "U2F Security Key" **cannot be used** because it's not actually a YubiKey, and doesn't support OTP or Challenge/Response modes)

Procedure

- Insert the YubiKey into a USB port on the server/workstation you are using.

- Launch AuthLite Configuration.

- Select the item "Program an AuthLite Key":

- The "Domain Name" box should contain the NETBIOS domain of the user account you wish to associate with this YubiKey. If your user is **not** in this domain, then you must install AuthLite in the domain where the user is homed.

- In the "Username" box, enter the username (SAM account name, NOT UPN) of the

user account you wish to associate with this YubiKey.

- If this user will use this key for logon to domain member workstations that can go offline from the domain (i.e. need to use cached logon credentials) then select the "Support offline workstation logons with this key" checkbox.[2]

- If you are using the "New Users" Group feature, then you can select the corresponding checkbox and the user account will automatically be added to the "New Users" group.

- Click the "Program AuthLite key" button.

- If you are NOT using the "New Users" Group feature, then you must manually add this user into one of your AuthLite User Groups.

- This YubiKey is now associated to the user account you specified.

Bulk programming from a standalone workstation

If you do not have direct physical console access to your domain, or if you want to program a large number of keys efficiently in one sitting, then you can use the following process.

Program YubiKeys

- From AuthLite.com/downloads, install the "YubiKey Programmer" standalone program on your workstation.

- Configure slot #1 for "AuthLite OTP" and slot #2 to "AuthLite Challenge/Response"[3]
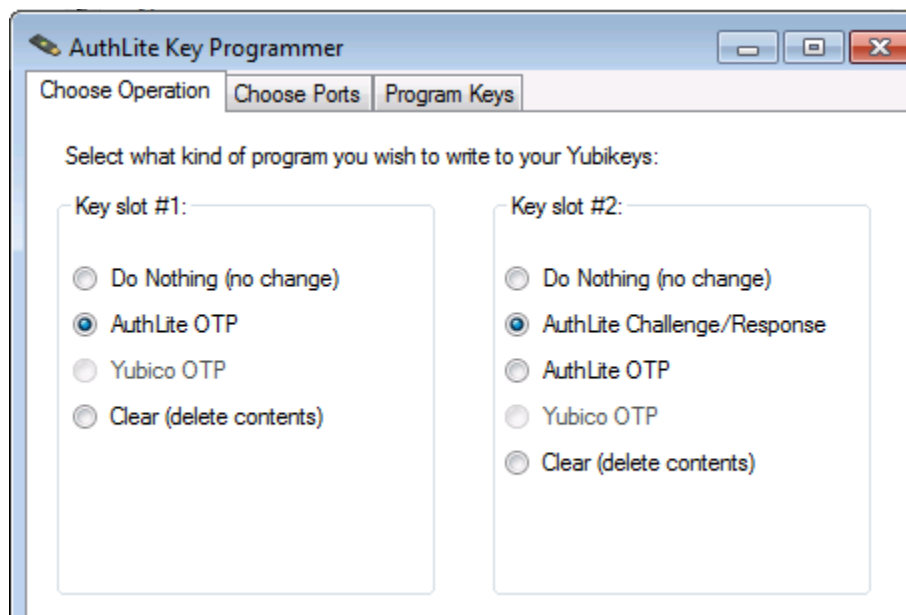
- If you are using shorter public IDs than the default, specify the number of bytes on this page as well.

- The point of a random public ID is to use it as an encryption key to protect the key's record in the data store. If you are using very short IDs, you should NOT randomly-

---

2 This will program the second identity slot of the YubiKey with a challenge/response token that will be used to provide 2-factor authentication when the key is plugged in to offline workstations. If you do not program this identity, then the key will NOT work for logon to offline workstations.
3 This will program the second identity slot of the YubiKey with a challenge/response token that will be used to provide 2-factor authentication when the key is plugged in to offline workstations. If you do not program this identity, then the key will NOT work for logon to offline workstations.

generate them because you will end up with collisions (more than one key programmed with the same ID, which will cause an import error).

- Go to the Choose Ports tab. Plug in one or more YubiKeys to identify which USB ports will be used during this programming session. If you plug them in one-at-a-time and wait for each port to be recognized, you will know which port corresponds to which key. This will in turn make it easier to find which key to unplug/re-plug in the event of a programming error. (Note: Due to Windows slow USB enumeration, there doesn't turn out to be much speed advantage to using many ports at once.)
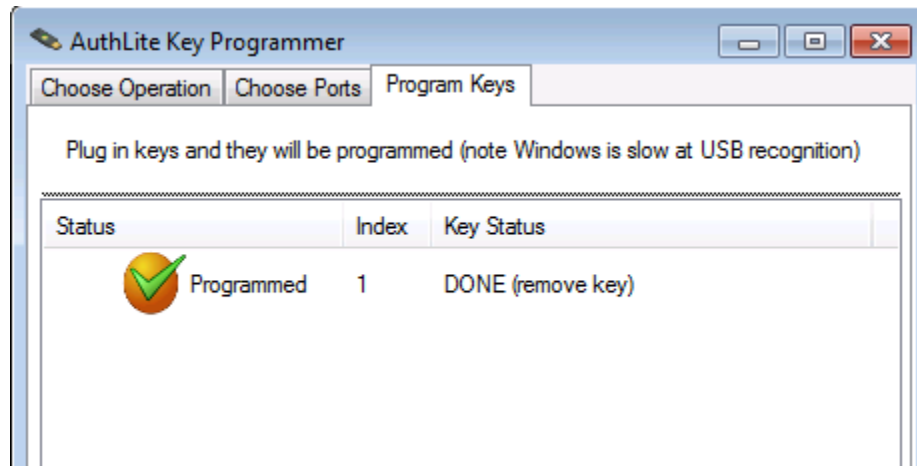
- When ready, click "Start Programming", and each time you plug in a YubiKey to one of the ports you chose, it will be programmed.

- Remove each key when the status reads "DONE (remove key)". Do **not** click "Finish" at this point, if you have more keys to program.

- You can continue to plug in new keys and program them. When finished with all keys, click the "Finish" button.

- Save the resulting XML file. This file should be considered sensitive information, as it contains all the secret values programmed into the keys. Treat this with the same security measures you would use for a password list or other secure document.

 Import programmed keys
- On a domain controller, launch the AuthLite Token Manager and select Import Keys from the File menu:
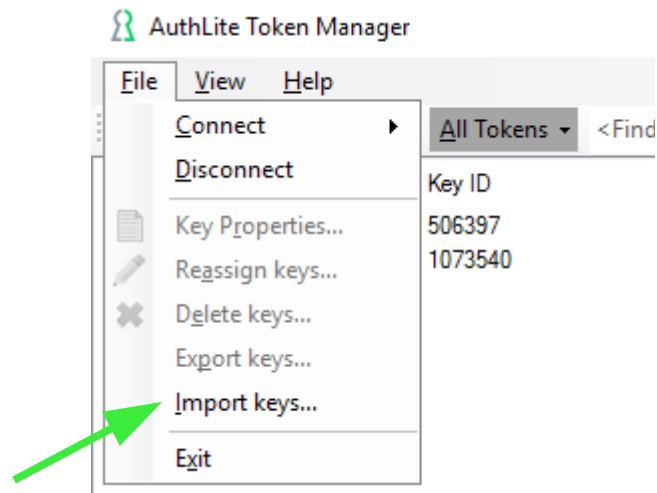
- Select the XML file generated when you programmed your keys, and click "Open".
- The key records will be imported:
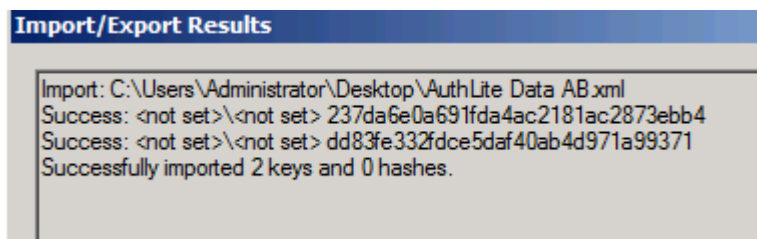
- You should then see new unassigned YubiKey records in the list:
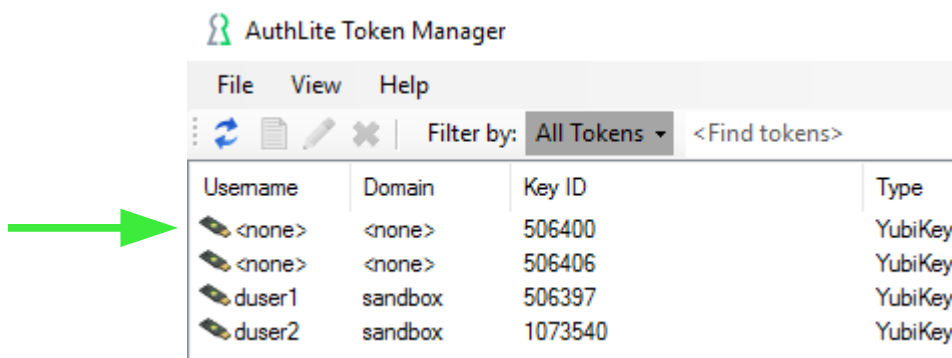
- Now that the file is imported, you should **DELETE the XML file** from your workstation and the DC. It is no longer needed once the keys are imported. This file should be considered sensitive information, as it contains all the secret values programmed into the keys.

  Next, we need to assign the YubiKey tokens to users.

Option 1: Administratively Associate each YubiKey to a User Account
- Right-click an unassigned YubiKey record and select "Reassign":
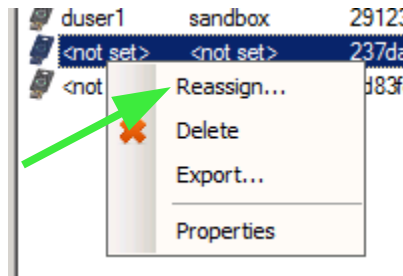
- The "Domain Name" box should contain the NETBIOS domain of the user account you wish to associate with this YubiKey. Since this AuthLite installation will only work in the installed domain, there's no reason you'd ever change this from the default. (To protect users in other domains of the forest, install AuthLite in those domains and configure the users there.)

- In the "Username" box, enter the username (SAM account name, NOT UPN) of the user account you wish to associate with this YubiKey.

- If you are using the "New Users" Group feature, then you can select the corresponding checkbox and the user account will automatically be added to the "New Users" group.

- Click the "Reassign" button.

- If you are NOT using the "New Users" Group feature, then you must manually add this user into one of your AuthLite User Groups.

- This YubiKey is now associated to the user account you specified.

Option 2: Users can associate their own keys

If you have programmed and imported your keys but don't want the administrative overhead of associating each key to a specific user and then handing them out, you can let users associate their own keys. All you need is an IIS server that can run ASP.net 4.5. See Deploying the AuthLite Token Profile Manager intranet application for more information.

- Distribute your programmed (but still unassigned) YubiKeys out to the users you wish to use AuthLite.

- Each user visits the URL of your ASP.net token association web app.
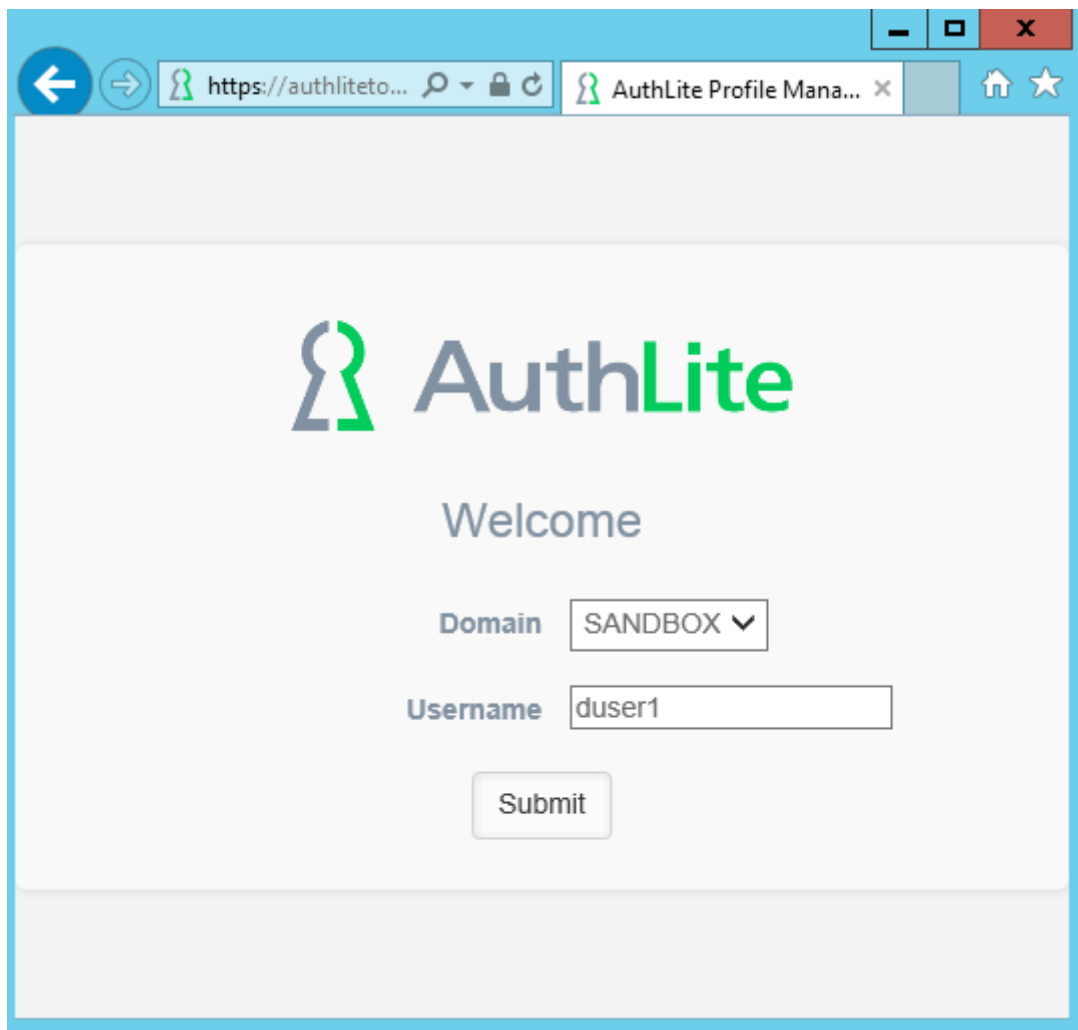
- Enter the username and password for the account.
- Select "Add YubiKey"
- Highlight the YubiKey Passcode field and touch the YubiKey button to enter an OTP from this key.
- Click "Add" and then the YubiKey will automatically be associated with this user account, and the user will be added to the "New Users" Group.
- After enrolling in AuthLite, the user can also log in to this portal again (with 2-factors) and add/manage their tokens.

Overwriting (reprogramming) an existing YubiKey

There are several situations where the existing settings of a YubiKey are no longer useful, and you want to reprogram it:

- The key was purchased from Yubico and is set up to use their online service, but you want to use it for AuthLite instead

- The key's former user is gone, or has changed their own account back to use a password-only logon

You can overwrite an already-programmed YubiKey in any of the normal AuthLite programming operations. You will see an additional dialog warning you that the old information on the key will be destroyed[4].

Provisioning OATH Tokens

Token Settings

By default, OATH token support is turned off. To enable it, set the number of digits in the Token Settings dialog:

---

4   Except in the bulk Key Programmer, which overwrites key data without warning you every time.
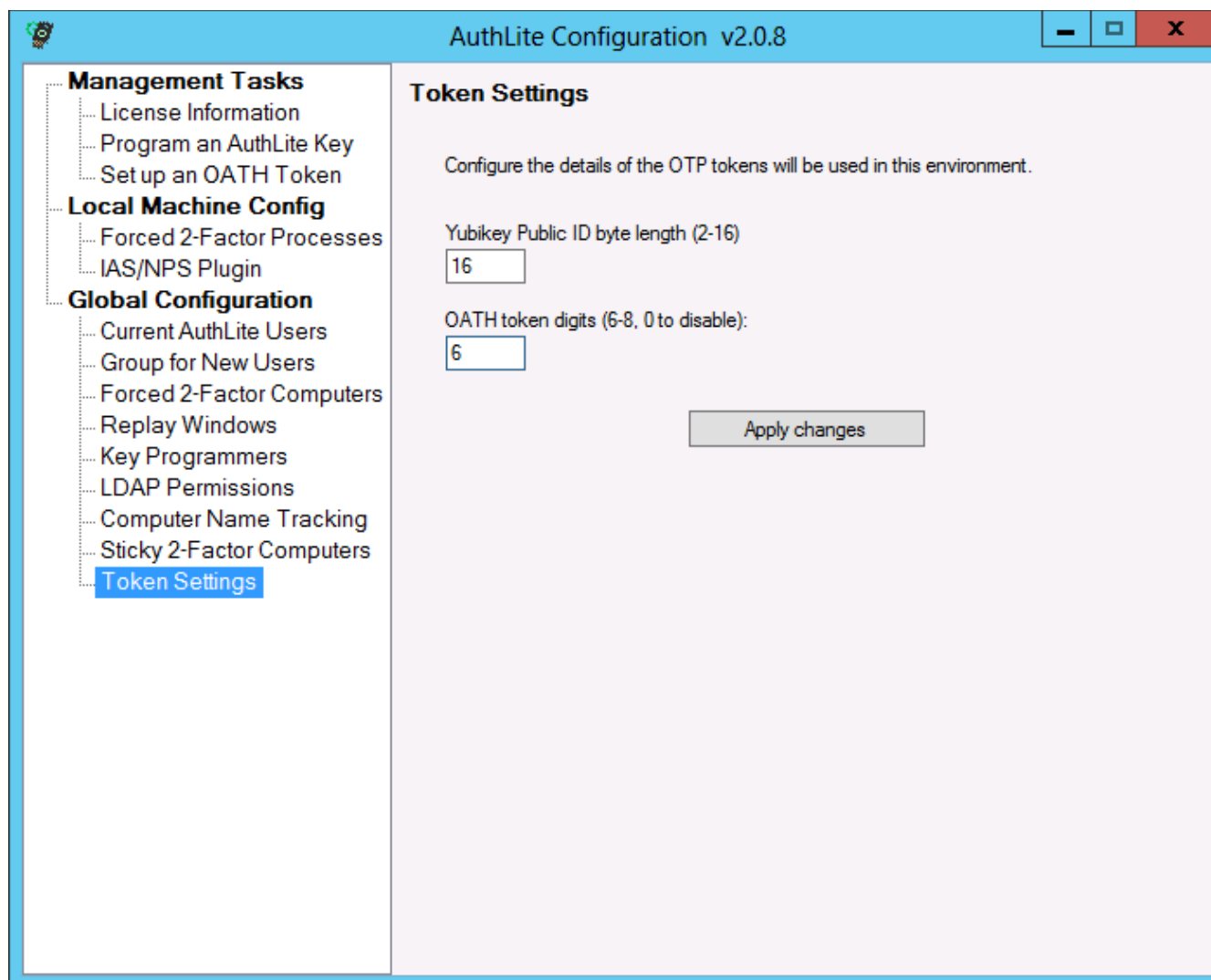
Google Authenticator uses 6 digits. If you are using some other token app/scheme, you may need to use 8 digits instead. NOTE: You cannot force a token app to use a different number of digits than it is programmed to use by setting this value.

One-off key provisioning from the console

AuthLite can easily create a record for an OATH token, associate it to a user, and issue a QR-code to program the Google Authenticator app.

Prerequisites

- AuthLite software installed on domain controllers and (if you are going to administer from a workstation) on that workstation machine.

- Valid license or evaluation key entered.

- Smart phone with Google Authenticator app or other compatible OATH TOTP app that can recognize QR-codes for token programming.

Procedure

- Launch AuthLite Configuration.

- Select the item "Set up an OATH Token":

- The "Domain Name" box should contain the NETBIOS domain of the user account you wish to associate with this token. Since AuthLite only operates on the domain which it is installed, this should always be the current domain you are joined to.

- In the "Username" box, enter the username (SAM account name, NOT UPN) of the user account you wish to associate with this token.

- In Description/Serial, enter a unique ID for this key to distinguish it in data store. It will also be encoded into the QR code for use in the Google Authenticator app. If you leave this field alone, a value will be generated automatically.

- If you are importing a hardware token, you must enter the seed from your OTP token's manufacturer. You can enter it in Hex, Base32, or Base64 format.

- For Google Authenticator, you MUST leave the interval at 30 seconds. Specify an interval of 60 for Feitian/FTSafe tokens (c200). Some hardware tokens only change once per minute, instead of the normal interval of 30 seconds. If you do not set this value correctly, your token will not work. Please note this interval does NOT tell how AuthLite how long a code should be "valid for". If you try to change this setting to make your tokens more lax, you will just break the token record instead of accomplishing what you want. By default a token code should function for about 1 minute in either

direction to account for clock differences. Having said that, once you use the code it cannot be re-used again without a corresponding replay window.

- If this token is going to be used to log on to domain servers or online workstations, choose "Online authentication". If you are defining an OATH token that will be used for offline workstations disconnected from the domain, choose "Offline". A single OATH record cannot do both tasks, because the workstations must know the OATH secret for offline tokens, and this causes them to be less secure than the online tokens.

- If you are using the "New Users" Group feature, then you can select the corresponding checkbox and the user account will automatically be added to the "New Users" group.

- Click the "Set up OATH Token" button.

- If you are NOT using the "New Users" Group feature, then you must manually add this user into one of your AuthLite User Groups.

A QR-code and encoded secret values will now be shown. Now:

- Scan the QR with Google Authenticator on the user's mobile device to set up a token within the app that can be used with AuthLite. Or,

- Send the Base32 code via SMS to your user's phone, and they can enter it manually into Google Authenticator.

- If you are using a hardware token, make sure the proper user gets the token.

NOTE: The QR-code and other code readouts should be considered sensitive information, as they contain the shared secret used between the OATH token and AuthLite! Do not email your codes, because the user's email is probably protected by just their password, and this transmits the new secret over a medium controlled only by the old factor, nullifying its security.

Users can associate their own OATH soft tokens

If you don't want the administrative overhead of setting up OATH soft tokens for your users, you can let users associate their own tokens. All you need is an IIS server that can run ASP.net 4.5. See Deploying the AuthLite Token Profile Manager intranet application for more information.

- Each user visits the URL of your ASP.net token association web app.
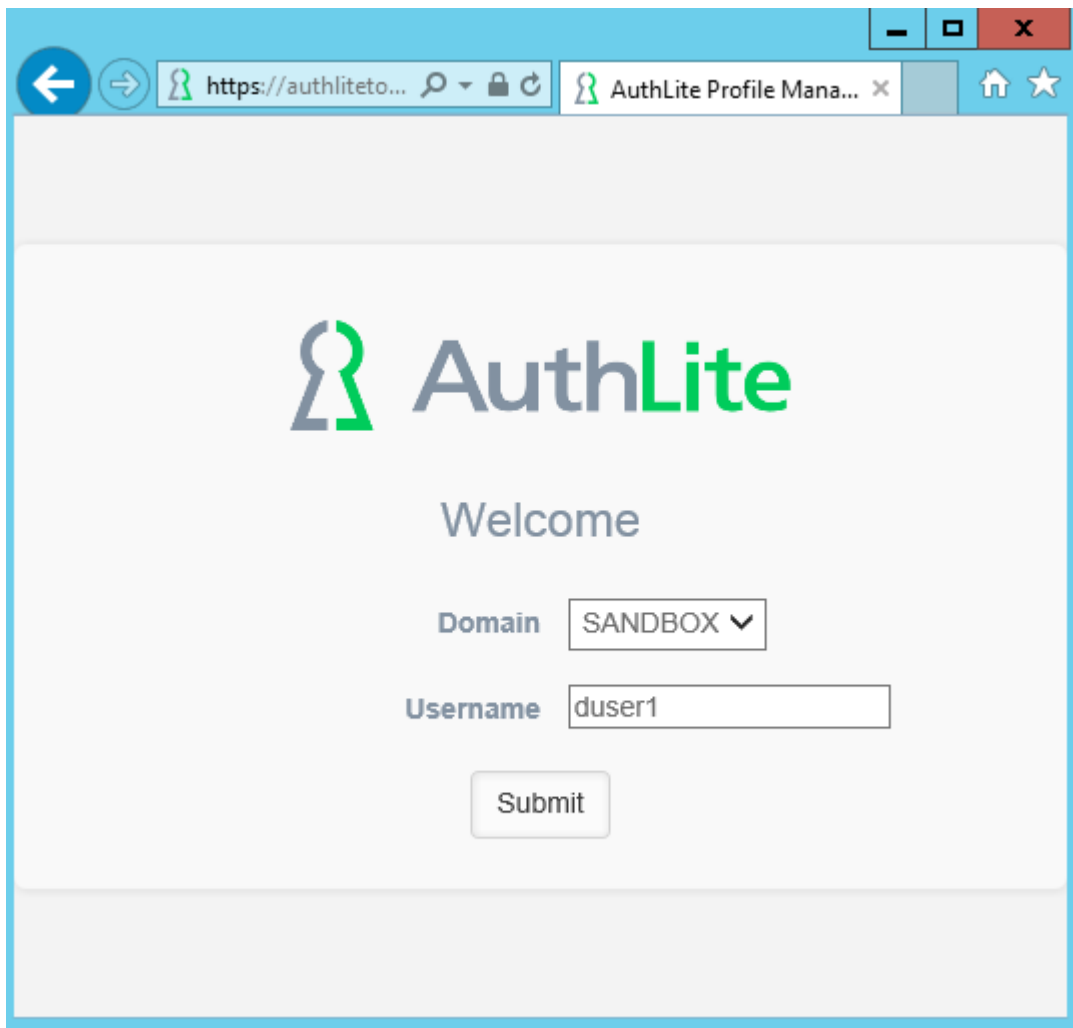
- Enter the username and password for the account.

- Select "Add OATH token"

- Click "Add" and then a QR code will be displayed. The user can snap this with the Google Authenticator app on their phone, and then click "Done". The new OATH token will automatically be associated with this user account, and the user will be added to the "New Users" Group.

- After enrolling in AuthLite, the user can also log in to this portal again (with 2-factors) and add/manage their tokens.

Bulk importing OATH tokens

You can import multiple hardware tokens, if you possess a properly formatted XML file. To construct the proper format:

- Import one token through the UI as described in One-off key provisioning from the console.

- *Before using the token*, export that key from the AuthLite Token Manager.

- Look at the exported XML file to see the proper format and values.

- You can omit Username and Domain if you don't want to import keys already

associated to specific users.
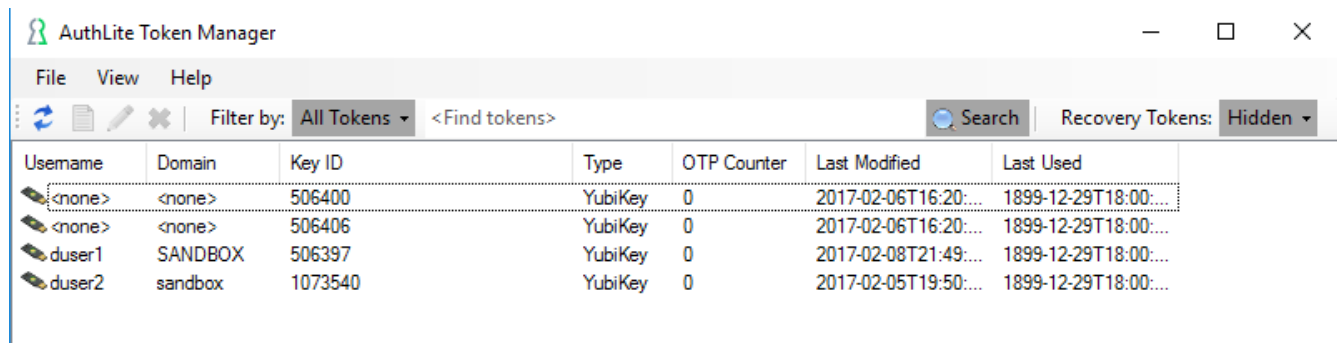
- Public ID can be any 32-character hex string. It must be unique for each key in your domain.

- You may wish to store the token's serial number in the DescriptiveID tag. There is a nil tag called SerialNumber, but it cannot be used for this since it is an integer only suitable for YubiKey-sized values.

- Note carefully that the OathDrift tag is set to 2147483647. This special value indicates that the token is new and has not yet been clocked to track its drift. AuthLite will update this value in the data store once the token has been characterized.

- Once you complete an XML file, you can import it from the Token Manager to bring in your tokens in bulk.

AuthLite Data Management

For maximum data safety, AuthLite stores key data forever by default, and does not run any logic to prune old users or keys out of its database. For example, when a user re-programs an existing key, the key's old data record is no longer used, but still remains in the data store. This does not present any security, performance, or functionality problems[5], but you may wish to remove old records to keep the database tidy.

The AuthLite Token Manager program can be run from any AuthLite-aware domain controller. It can also be installed on other member machines, although this feature is not selected by default.

This tool allows you to display, sort, and search the data records for all the AuthLite keys that are currently provisioned in your domain.

Connecting to Active Directory

When you first start the AuthLite Token Manager, the application will automatically connect to the Active Directory instance on the local machine. The name of the domain controller to which the application is connected is shown in the status bar. To connect to a different domain controller, choose the **Connect to...** menu item from the **File** menu.

"Filter by" dropdown

When you first open the app, all tokens are shown except Recovery Tokens. You can change which tokens are shown by selecting a different term in the dropdown. Some selections require a search term to be entered in the "Find tokens" box.

Find tokens

Click inside the "Find tokens" box in the tool bar, enter your search term, and hit the **Enter** key. Hit **ESC** or press the "Clear Search" button to reset the search. The fields to search can be selected from the drop-down on the left.

Find by OTP

If you have a YubiKey OTP, and need to find the corresponding key entry, enter the OTP into the Find Tokens box and hit the **Enter** key or Search button. The key record corresponding to that OTP will be shown. This only works with the Filter dropdown set to "All Tokens", "Assigned tokens", or "Key ID Matches:".

---

5   Key data is only useful while it matches the AES key of a programmed AuthLite token, and correlates with a user. After a key is re-programmed, the old data no longer matches it and cannot be used either accidentally or by an attacker.

View a Token's Properties

To view the properties of a token, double-click on a row in the list, or right-click on a record and choose **Properties** from the context menu.

Delete a Token

Select one or more rows in the list, choose the **Delete key(s)...** option from the **File** menu, or hit the **Delete** key, or press the **Delete** toolbar button, or right-click and choose the **Delete** option from the context menu. **Be careful** because deleting token records is a permanent operation with no "Undo". If a record is in use, then deleting it will prevent that user from logging on to AuthLite-aware systems/services with that token ever again.

Reassign a Token

From the key properties dialog, click the **Reassign key** link. Or choose **Reassign key...** from the **File** menu. Or select one or more keys, right-click, and choose **Reassign** from the context menu.

In addition to selecting a different user, this dialog allows you to revert a token to its "Unassigned" state.

You can also set more than one token at a time. In the above example image, two tokens have been selected. Whatever action is taken in this dialog, both tokens will be set to that user.

Recovery tokens

Beginning with AuthLite version 2.2, each AuthLite-aware workstation will create a "recovery token" for each AuthLite user that logs in, and publishes it to your AuthLite data store. If a user is offline from the domain and does not have their token, then an administrator can find the correct recovery token for that user and workstation, and reveal the current OTP value for that token. Reading the value (for example) over the phone to the user, will allow them to enter the value and authenticate to the workstation without their normal token.

To use:

Filter the key records to show only the user you want to see
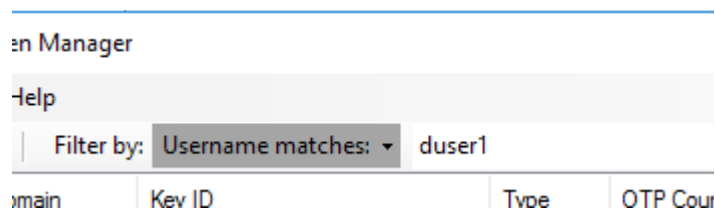
Select "Show Recovery OATH tokens" from the View menu or the toolbar.
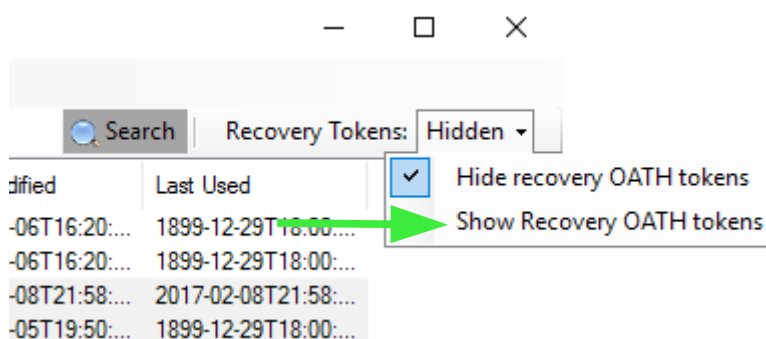
Locate the correct record for the workstation that the user is trying to log into:
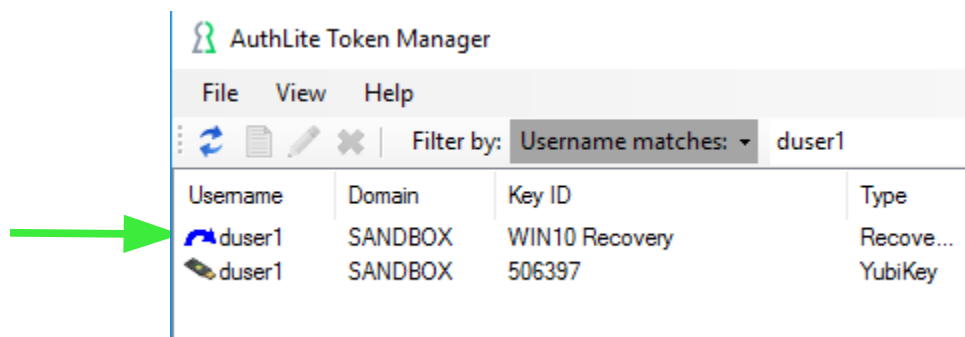
In the Properties dialog for this recovery token, click the "Show Current Code" link.
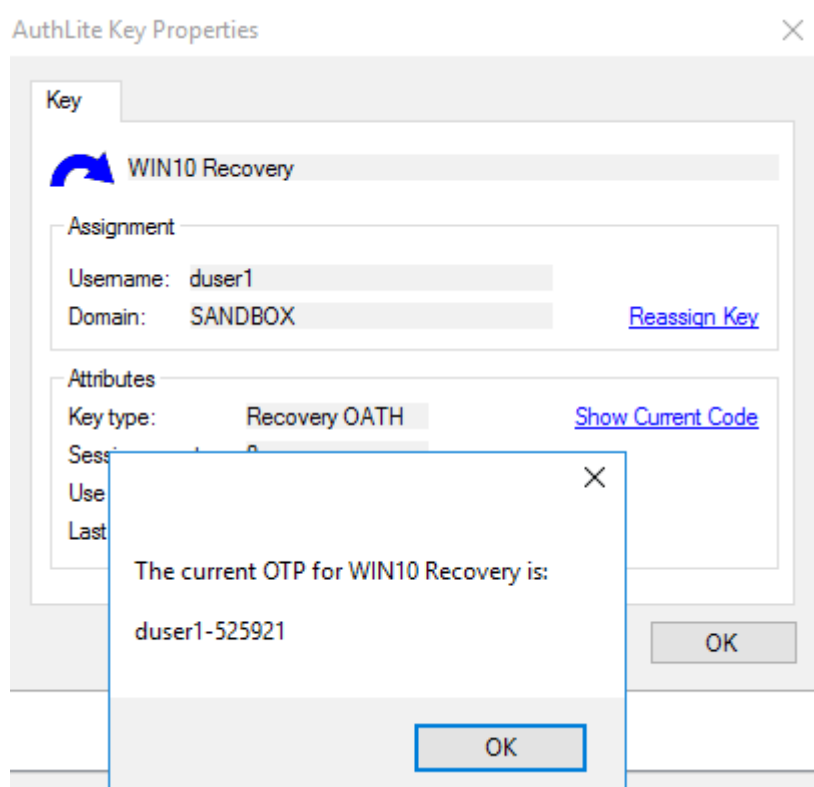
Communicate this OTP to the user in an out-of-band fashion such as via a phone call. Please note that this code will change every 30 seconds. Although there is a grace period around the OATH token's usage, you should wait to view the current OTP code until the user is ready to enter their logon credentials.

The user enters this value in just the same manner as they would use any other normal OATH token. The only difference here is that the code is being viewed by an administrator on the LAN instead of by the user on their phone.

Notes:

- To work, this feature requires that you enable OATH tokens in Token Settings.
- Logon sessions using a recovery token (or an Offline OATH token) cannot acquire 2-factor network logons such as Kerberos or NTLM. It will only authenticate to the workstation's desktop.

Delegating Token Administration

In the "Administrative Groups" dialog, you can select a security group whose users will be allowed to program and import new YubiKeys, Provision OATH Tokens, and assign/reassign tokens to users. This is a very powerful permission since these admins can mint their own 2-factor tokens for any user, but of course less powerful than making them Domain Admins.

You can also select a security group whose users will be allowed to see the OTP of Recovery tokens. This can allow a help desk worker to assist a user to access their offline workstation for example, without giving that worker a more powerful access.

Multiple users with the same OTP Token

At this time it is not possible for more than one AD user account to share a single token. In cases where it would be inconvenient for one human to carry multiple YubiKeys (such as administrators with many accounts) then it may be more sensible to create OATH soft tokens for some or all of the accounts. Since they are displayed in separate rows in the Google Authenticator app, having a sizeable number of tokens is still completely reasonable.

Password Operations

When an AuthLite user changes their password or has their password administratively reset, this will not affect their AuthLite keys. They will log in with their existing OTP token and the new password that has been set.

Reverting a user to normal Windows logon

To revert a user back to one-factor logon and remove them from all consideration of AuthLite permission checking, you must remove the account from all AuthLite User Groups. Also, uninstalling AuthLite everywhere will (trivially) have the effect of restoring all accounts to standard Windows functionality. Note however that if you have used dynamically assigned 2-factor groups anywhere in your domain, your users may lose access to those resources since they will no longer be assigned into the dynamic groups.Multiple OTP Tokens for the Same User

You can assign as many YubiKeys and OATH tokens as you want to each user account, and they will all operate interchangeably[6]. Each key will be programmed with its own distinct randomly-generated secret. So the loss or compromise of one key does not implicate any other keys the user still carries.

---

6   Although OATH tokens cannot be used for offline workstation logons.

Requiring Two-factor Authentication

By default, installing AuthLite will not add any authentication requirements to users or computers in your domain. This is by design. You must tell AuthLite which users, computers, and processes should be required to use two-factor authentication.

- You must tell the domain which accounts are AuthLite users by properly configuring the AuthLite User Group Pairs.

Using the Domain's Own Access Control Lists

The most powerful, flexible, and foolproof way to enforce 2-factor authentication at any point in the domain is to use Windows' built-in security controls. Our User Group Pairs feature means that you can set up ACLs to only allow sessions that have been tagged by AuthLite as 2-factor logons.

Tutorial Video!

AuthLite Groups and Policy

For example you can use Group Policy to restrict which security groups are allowed to authenticate to certain machines.

Forced 2-Factor Processes

Explanation

For some servers it is not appropriate to require two-factor for every authentication by AuthLite users. For example, consider an Exchange server which must support OWA and ActiveSync. In this case, you will (ideally) be using client certificates on each ActiveSync device to increase security, and the user will authenticate with username+password only.

Even AuthLite users will authenticate this way, since it is not feasible to use YubiKeys on most mobile devices[7] and in any event it would be impractical to supply an OTP each time ActiveSync is run to synchronize email/contacts/calendar. You cannot use group policy or put the Exchange server in the Forced 2-Factor Computers list, since that would break ActiveSync authentications for AuthLite users. But you may have other services on the same Exchange server such as Outlook Web Access, that you wish to require two-factor authentication for AuthLite users.

Even though the domain controllers are not enforcing two-factor authentication on the Exchange server, the AuthLite code running on the server itself can voluntarily enforce two-factor depending on the name or command-line of the process that is performing the authentication. This is accomplished by configuring the "Forced 2-Factor Processes" list on the server. Each string you place in this list will be matched against the "Logon process ID", the file name, and the command-line of the calling process. If there is a match, then two-factor authentication will be enforced for AuthLite users for that process.

NOTES:

- You must configure this list on each member server separately.

- It is not necessary to set up this list for any machine that is adequately enforced using

---

7   Although you could use an OATH token app. But for the purpose of this example assume not.

Group Policy, or represented in the [Forced 2-Factor Computers](#) list.

- By default, settings you make here will apply equally to all AuthLite user groups. You can select a specific 1-Factor Tag group from the dropdown to enter settings that will apply only to that group. Each group can have its own settings in addition to the settings that apply to every AuthLite user group.

Example Configuration

For a concrete example, we'll consider OWA. On an Exchange server both ActiveSync and OWA run under the same process name (w3wp.exe), but they have different command-lines. You can see what processes have performed authentications so far since the last restart by clicking the "View active logon processes" button:

Note the different command-line options passed to the IIS app pools. Since we want to enforce two-factor authentication for OWA we can add the value "MSExchangeOWAAppPool" to the forced 2-factor processes list, and then OWA logons to this server by AuthLite users will require OTP and password. Other logons will be allowed with password only.

At the time of writing, the following process list causes all non-ActiveSync Exchange processes to require 2-factor for AuthLite users:

- MSExchangeOWAAppPool
- RPCClientAccess
- MSExchangeRPCProxy
- MSExchangeOABAppPool
- MSExchangeServicesAppPool

NOTE: IIS/Exchange caches logons for a period of time. Also, AuthLite only updates its knowledge of groups and settings every 20 minutes. So after you enable two-factor enforcement on OWA, you may still be able to log on with one-factor until these caches expire.

RDP Forcing

To enforce two-factor authentication for the server when Remote Desktop is used, select that checkbox. This appears as a separate option because both RDP and console logons use the same process (so there's no way to add RDP to the above list).

System Forcing

Certain services may perform authentication inside the Windows kernel, thus there is a checkbox to force these processes to require 2-factor for AuthLite users.

Forced 2-Factor Computers

You can tell domain controllers to reject one-factor authentication from AuthLite users if the requests originate from certain machines. This is not as robust or granular as Using the Domain's Access Control Lists, but is still be useful for some circumstances. For example, non-Windows machines (such as MacOS, Linux, etc.) don't honor group policy, so they must be forced by this dialog.

By default, settings will apply equally to all AuthLite user groups. You can select a specific group from the dropdown and create settings that will apply only to that group. Each group can have its own settings applied, in addition to the settings that apply to all groups.

Select by IP Address / Range

- Enter one or more IP ranges into the Forced 2-Factor Computers list.

- Any time an AuthLite user authenticates from a computer represented in this list, the domain controllers will require two-factor authentication before allowing the authentication.

- **Security Note:** If **any** of your domain controllers in the site are missing the AuthLite software, then they will not enforce or understand two-factor authentications.

Selecting by Name/Group

- It is possible but *not recommended* to select machines and groups of machines by name. AuthLite will try to decide what machine is connecting by keeping a map of IP addresses to computer names (see next section). It is not always possible to determine the computer name accurately, so you should use the IP Address selection instead wherever possible.

Computer Name Tracking

- If you are enforcing 2-factor computers based on computer names or groups, it is highly recommended to activate Computer Name Tracking. This will help AuthLite

software on the DC's to associate incoming authentication attempts with the correct domain member machine, and prevent spoofing the identity of client systems for the purpose of bypassing two-factor restrictions:
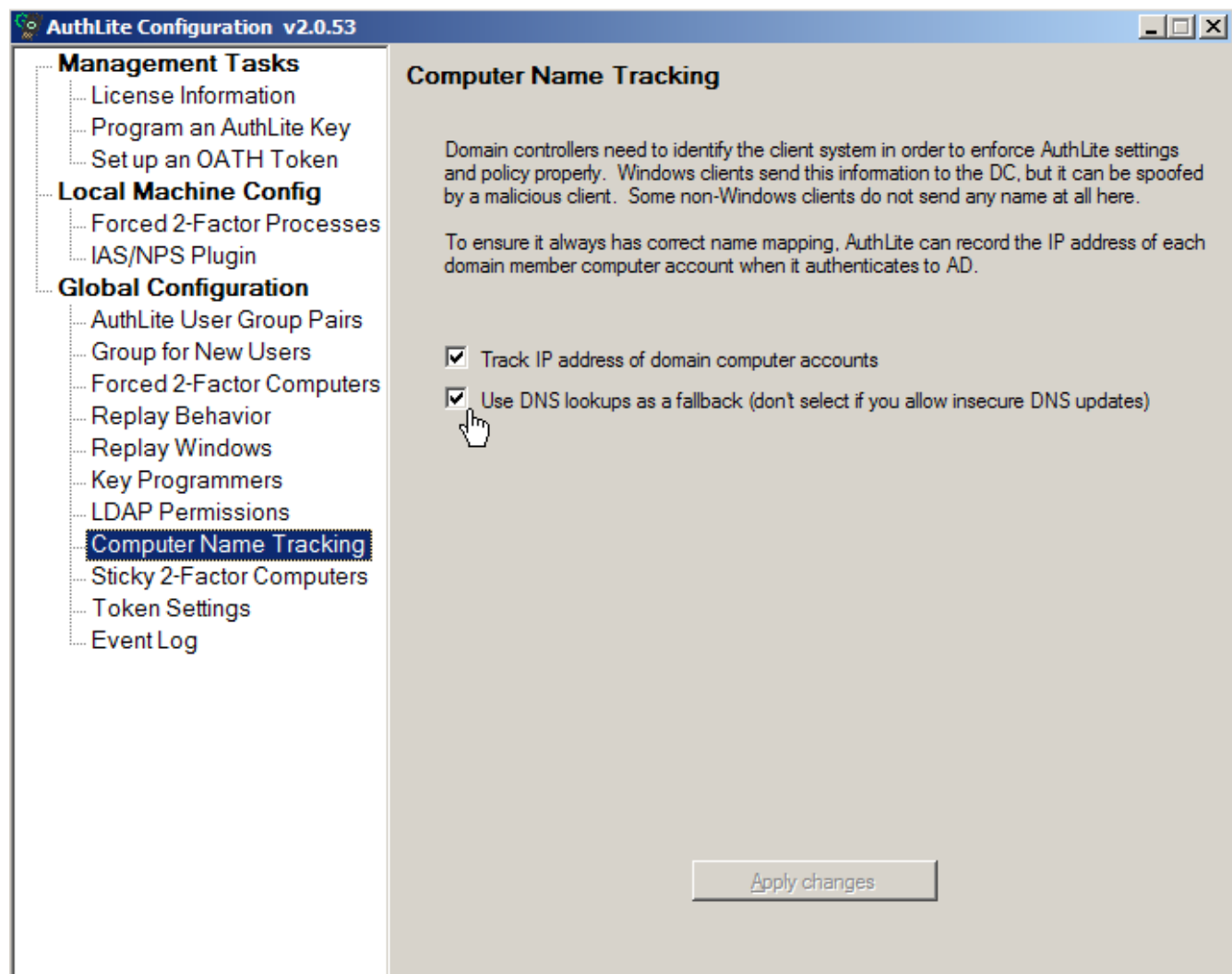
Entering AuthLite Credentials

AuthLite supports several different ways to supply credentials. Which method you choose may be constrained by the application or logon protocol you are using. For example the application may not tolerate an OTP in the username field, or the authentication protocol could hash the password field contents, rendering useless any OTP entered into that field.

For YubiKeys:

- Enter a YubiKey OTP in the username field (instead of the username), then enter password as normal. Supported most broadly, although some third-party applications can't handle seeing something unusual in the username field. Or,

- Type username as normal, then enter (password followed by OTP) into the password field. Only supported if AuthLite is installed on the server requesting authentication and the password field is not hashed). O*r,*

- Type username as normal, then enter (OTP followed by password) into the password field. Same limitations as above, except a bit worse because the YubiKey will auto-"tab" after the OTP is entered, which places you out of the password field.

For OATH OTPs:

- Enter your username followed by a dash "-" followed by the OTP code shown on your oath token. Then enter password as nomal. Supported most broadly, although some third-party applications can't handle seeing something unusual in the username field. Or,

- Type username as normal, then enter (password followed by a dash "-" followed by the OTP code) into the password field. Only supported if AuthLite is installed on the server requesting authentication and the password field is not hashed). Or,

- Type username as normal, then enter (OTP followed by a dash "-" followed by your password) into the password field. Same limitations as above.

Event Logs and Troubleshooting

Because AuthLite uses your existing enterprise software and UIs, it is usually not possible to report meaningful error messages through the user interface when something goes wrong. You will usually just see a generic-looking logon error. In addition, if 2-factor enforcement isn't working the way you want, then there won't be any errors at all since the 1-factor authentication might be allowed when you expected it to be blocked. Because we can't show detailed errors to the user, event logs are used instead.

Event Log

AuthLite records events into the "AuthLite Security" Event Log any time a logon attempt occurs on the system. Also the service records events for the following actions:

- OTP authentication and replay events

- Key provisioning and other management events

- Computer name / IP lookup events

Note: Prior to version 2.0.60, most of these events went into the generic Windows Application log instead of the specialized "AuthLite Security" log.

There will often be *several events* corresponding to what you would consider a "single logon". And events may be spread across the workstation, member server, and one or more domain controllers that are all involved in a "single" logon. These events will only appear when the AuthLite software is installed. (All DCs that may handle authentications from AuthLite users **must** have AuthLite installed to function properly.)

It is possible to turn up/down the level of events that the AuthLite service and infrastructure components emit:
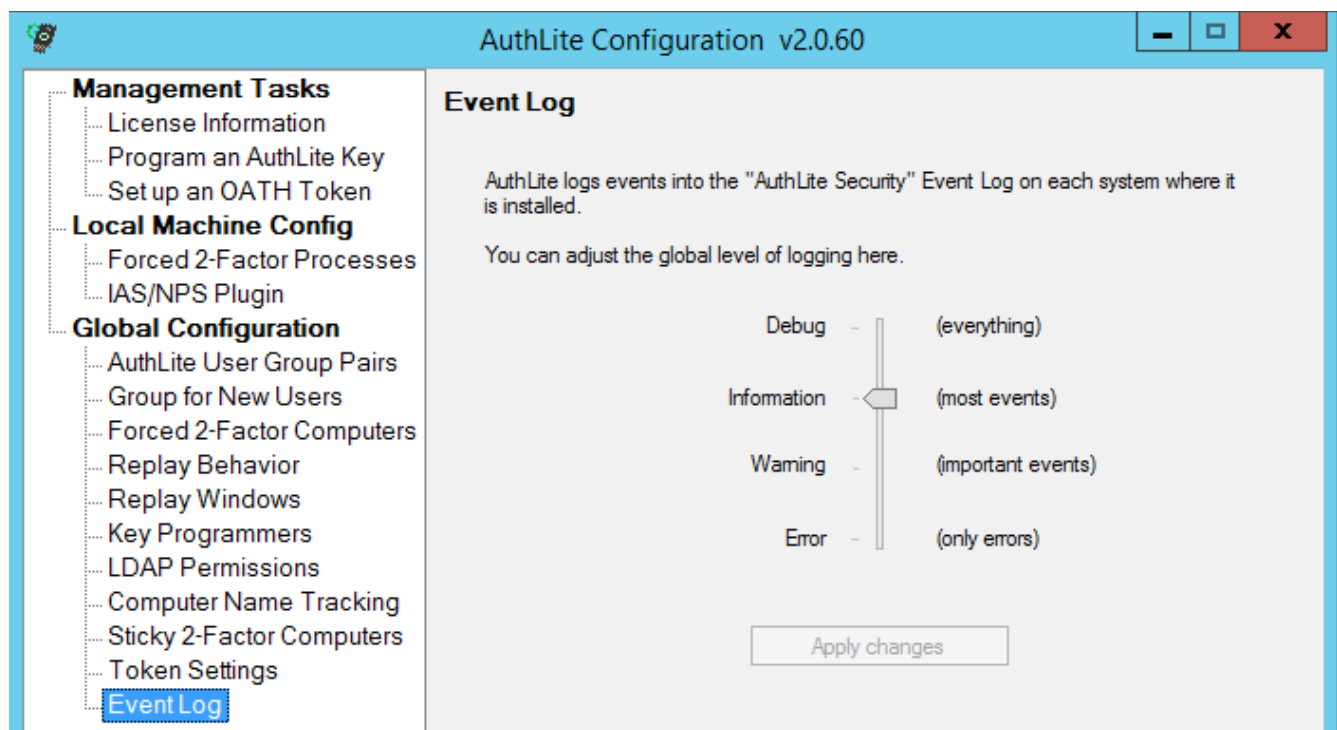
The log level is set to "Information" by default. Please do not turn it up to Debug or Trace unless instructed by customer support. **Warning: If your DCs constantly process more than about 10 authentications per second, then turning on debug logging can slow down your domain services until they cause the system to fail.**

Troubleshooting

When you are trying to determine why authentications are not working as you expect, your first action should be to note down what time you did the failing test, then find all corresponding events from the AuthLite Security event log on every implicated workstation, server, and all DCs in the local site, and read what they say. Events of interest will be clustered together around the time of your test. There may be many other unrelated events too, particularly on busy DCs.

If you open a support ticket, collecting events is among the first tasks we will need you to perform.

Windows "Security" event log

When a user completes a logon to Active Directory, you will still see the normal Windows Security log events as well. AuthLite does not remove or replace any of the default Microsoft authentication technology.

Windows Workstation (Endpoint) Protection

There are several important points to consider regarding workstations and the protection of logons/data on these endpoints.

Threat Modeling

Unlike remote network resources, workstations (especially mobile laptops) present a greater attack surface. In addition to subverting the OS logon itself, an attacker could choose to pull the hard drive and directly attempt to access the stored data. Therefore, solutions that only protect the OS logon itself may not be sufficient protection. Consideration should be given to the protection of stored data as well.

Full Drive Encryption

AuthLite does not provide its own branded/integrated full disk encryption. We recommend using BitLocker with TPM if possible, since this integrates seamlessly into Windows and does not require the entry of any additional credentials.[8]

Without drive encryption, endpoint protection is not effective against an attacker who steals the system or its hard drive. The attacker does not need to authenticate to the account, they can simply read data off the hard drive.

If you choose to use a third-party FDE solution that "synchronizes" with Windows user credentials (meaning you enter the username and password at boot time) then it **will probably not work with AuthLite**. AuthLite OTPs affect the Windows logon on workstations. A*t best* your FDE will still just use your password. *More likely* it will stop being able to synchronize properly with password changes and/or prevent AuthLite logons from working. *At worst* it could prevent you from decrypting your drive. You should contact your FDE vendor and follow whatever solution for strong authentication they support.

Untrusted workstations

With AuthLite version 2 and later, the credential data processed by a workstation cannot be reused in the future to gain another fresh Kerberos or NTLM session. By contrast, a malicious AuthLite v1 workstation could gather enough information to impersonate the user in the future. For more information see [this KB article](#).

Offline Workstation logon with YubiKeys

A randomly-generated challenge/response secret is associated to each YubiKey, and gets programmed into a YubiKey's second identity slot (this option is called "support offline logins" in the admin UIs, and "AuthLite Challenge/Response" in the Key Programmer app.)

The HMAC-SHA1 challenge/response is checked before letting the user log in. For normal online logons, this challenge/response secret is retrieved from the domain controller and synchronized to the workstation. It is not necessary for the YubiKey to be plugged in directly to the workstation (it can operate as a remote keyboard e.g. with RDP or virtual machines).

---

8   This BitLocker+TPM authentication mode is quite secure for most threat models, however it is vulnerable to Cold Boot attacks. To protect against the threat model of an attacker chilling and removing a running workstation's DRAM, another approach should be considered.

During "offline logons", the AuthLite software communicates directly with the plugged-in YubiKey to do the challenge/response procedure. Any (properly programmed) YubiKey that has previously been used to log in online at this workstation will also be able to log in with offline mode.

Ramifications for YubiKeys

Programming YubiKeys to support AuthLite offline logons will "use up" both identity slots in the YubiKey, and destroy any other (old) information that was previously loaded onto the key.

1. The first identity slot will contain the one-time passcode identity for entering OTPs as keystrokes. This identity is triggered each time the gold contact on the key is pressed. The OTPs generated can be used to log in to any AuthLite-aware system or network.[9]

2. The second identity slot will contain a challenge/response secret that gets automatically used by AuthLite when logging in to workstations that are offline from the domain.

Offline Workstation logon with OATH tokens

Beginning with version 2.2, you can create "offline" OATH tokens that will be synchronized down to your workstations when the user connects online. The offline OATH token can then be used to authenticate to any of the workstations that previously cached its record. "Online" OATH tokens cannot be used in this way, because it's not possible to authenticate them without having a connection to the DC.

Zero-client two-factor workstation logons

It is possible to log in to workstations with 2-factor credentials even if the workstation has no AuthLite software installed. The DC's with AuthLite installed will still enforce Two-factor Authentication as expected.

Please note that without AuthLite software installed, 2-factor offline logins will not be possible. You can push this software with Group Policy and it is recommended at the highest level to install the software if at all possible.

There are several important considerations for zero-client workstations:

- You must disable cached logons for the workstation via Group policy. If you don't do this, then any attacker who knows the username and Windows password can log in simply by forcing the machine to go offline first. This is because without AuthLite software, Windows is unaware of the OTP security and continues to protect cached logons using only the Windows password.

- Windows will hash the password field, so you must enter the OTP in the username field in order for it to reach the DC. On AuthLite-aware systems, no such limitation exists, because the software can intercept and process the authentication request properly regardless of how credentials are entered.

- You should enforce the group policy to prevent the last username from being shown. This setting will ensure you always get a blank, type-able username field during logons and unlocks. If you don't do this, then the unlock screen will only show a password

---

9 Provided the intermediate software supports long enough username/password values. AuthLite OTPs are normally 64 characters. They can be programmed shorter but will in all cases be at least 34 characters long (that shortest length would support a maximum of 255 keys in your whole domain.)

field. To unlock, you'd have to use the "Switch User" functionality and select "other user" so you'd get a new opportunity to enter an OTP into the username field.

- Parts of Windows will display the OTP you entered instead of showing the proper username. Although the correct username is returned in the session token, parts of the default Windows software will use the value that was typed in at logon time. This will result in generally cosmetic issues.

Known workstation limitations

- If a user's session lasts long enough for their Kerberos ticket-granting ticket to expire, the workstation will attempt to acquire a new ticket in the background using the previously entered credentials. For 2-factor users this operation will fail, because (by design) the workstation does not possess the user's full credentials. Therefore, logon sessions that endure past the expiration of the Kerberos TGT will become unable to access network resources. Upon trying to access a resource without a valid TGT, Windows will automatically show a message to the user instructing them to lock and unlock the workstation to provide domain credentials.

Workstation Safe mode operations

AuthLite can be used in *Safe Mode with Networking*. Note that you must use safe mode **with networking**, not the default safe mode or safe mode with command line. The AuthLite service communicates with the Windows infrastructure using a TCP socket, so the networking components must be loaded in order to perform any AuthLite actions.

Note: If your logon fails, it could be that the AuthLite service has not started yet. Wait a few minutes until all services have surely been loaded, and try again.

MacOS Workstation Logon

Please see this video for information on configuring an AD domain-joined MacOS system to authenticate AuthLite users with 2-factor at the logon screen.

Tutorial Video!

[AuthLite for MacOS logon](#)

Please note that unlike Windows workstations, this does not enforce 2-factor authentication for mobile accounts when the workstation is offline, nor for FileVault full disk encryption. At boot time, the FileVault logon will still just be the user's AD password. Also, when the machine is offline, the mobile account will allow logon with just the AD password as well.

When AuthLite is configured properly on the server side, the user will still require a 2-factor logon in order to obtain a Kerberos ticket and network logon credentials.

LDAP logon support/enforcement

Test 2-factor support

In order to check whether your LDAP service can accept AuthLite 2-factor credentials, attempt the following tests. We are not enforcing anything yet, just seeing whether the LDAP service can tolerate a 2-factor credential without erroring.:

1. First, attempt a logon by putting the OTP in the username field:
   ◦ With a YubiKey, tap the OTP into the username field.
   ◦ Or with an OATH token user, type in the username followed by a dash "-" followed by the OTP from your OATH token app.
   ◦ Enter the normal AD password in the password field.
2. Next, attempt a logon by putting the OTP in the password field:
   ◦ Enter the AD username as normal in the username field.
   ◦ Type the AD password into the password field.
   ◦ With a YubiKey, tap the OTP into the password field next, after the password characters.
   ◦ Or with an OATH token user, enter a dash "-" after the password, followed by the OTP from your OATH token app.

If either of these tests gives you a successful logon, then it is highly likely your LDAP service can be integrated as-is with AuthLite to enforce 2-factor logons.

Identify logon method

Find the DC which is being used for your LDAP authentication, and open the "AuthLite Security" Event Log. You can search (using ctrl-F) for your LDAP user's username in the log right at the time you did the above logon tests. There should be an Event ID 20 or 21 (the LDAP service on the DC initiating the authentication). There should also be a nearby Event ID 1, 10 or 11 (the DC's authentication core processing the logon). And there should be a nearby Event ID 0 with the same username and the details "OTP looked up successfully".

The exact combination of these events will tell important details about how the LDAP service is authenticating. Be sure to match the time of the events to the time of your testing. Otherwise you might be looking at log items that refer to some other login somewhere else.

Turn on 2-factor enforcement and compatibility

- If you see event 10 for your LDAP user, then the OTP is being passed to the logon. To enforce 2-factor for AuthLite users, just add the LDAP server to the "Forced 2-factor Computers" list.

- If you also see event 1 for your LDAP user, then the OTP is being passed multiple times (potentially, several logins in fast sequence). Begin with putting the LDAP server in the "Forced 2-factor Computers" list to enforce 2-factor for AuthLite users. Then, add a short Replay Window for the LDAP server so the same OTP can be used several times in a row to let the multiple logins work.

- If you see events 1 and 11 for your LDAP user, then the OTP might be getting looked up and then thrown away before the logon. Begin with putting the LDAP server in the "Forced 2-factor Computers" list to enforce 2-factor for AuthLite users. Then add it to the "LDAP Permissions" list as well so the OTP lookup is allowed to be used during the next authentication from that machine for your username. Then, iIf your authentications are failing, then add the LDAP server to "Sticky 2-Factor Computers" as well. This lets subsequent authentications use the same OTP presented by the

earlier one, even though the subsequent logons aren't passing the OTP.

- If you see just an event ID 20 with "Status 0xc000006d 0xc0000064" or "Status 0xc000006d 0xc000006A", then the LDAP server may be truncating the OTP or otherwise parsing the credentials strangely. Please contact customer support for assistance.

- Also if you see *no events* for your LDAP user matching the time you tried the 2-factor logons, then your service might be totally failing to pass the credentials containing the OTP. Please contact customer support for assistance.

Support for domain-joined Linux systems

If you have Linux systems joined to the domain with sssd, samba4 or similar, you can enforce 2-factor for AuthLite users when they log on to these systems. They will need a PAM module and some configuration to pass the OTP to AuthLite and then allow the password portion of the logon to work. Please see this KB article for more information, and contact customer support for assistance if needed.

VPN/RADIUS Authentication Overview

AuthLite has support for several arrangements of RADIUS authentication. Different situations you may encounter are described below.

- Different RADIUS clients have different expectations about how the authentication will work

- Different VPN tunnel types treat the password field differently

RADIUS for Username and OTP authentication (no password)

Many vendors, such as Citrix and Juniper, allow you to configure 2-factor authentication by setting up two separate authentication mechanisms. The first mechanism (usually Windows native, or LDAP) is used for the "normal" authentication to Active Directory of the username and password. The second mechanism is set to RADIUS, and pointed at an AuthLite-aware RADIUS service.

The RADIUS server will only receive the username and the OTP. In this setup, the bulk of your AD infrastructure need not even be AuthLite-aware, since the only OTP authentication point is the RADIUS service, and the other DC's just get standard username/password requests.

You can set up this configuration with IAS/NPS as your RADIUS service. Select "one factor" in the AuthLite IAS/NPS PAP settings.

RADIUS for authentication of OTP and password together

Some systems such as the Cisco VPN do not split up their authentication into two steps as above. You can use one RADIUS target to authenticate both factors at once, in several different configurations.

How credentials are entered
- MS-CHAPv2 with OTP in the *username* field, and password in the *password* field. MS-CHAPv2 hashes the password field at the client, so the OTP must be sent in the *username* field. Supported by IAS/NPS. *PPTP NOT RECOMMENDED* [10]

- PAP with OTP in the *username* field, and password in the *password* field. Supported by IAS/NPS.

- PAP with the username in the *username* field, and the password and OTP **together** in the *password* field. Supported by IAS/NPS.

Constraints for different authentication scenarios
There are several considerations that will constrain which authentication strategy you can use:

- 802.1x authentication uses PEAP between the workstation and access point, and then RADIUS with MS-CHAPv2 between the access point and the authentication server. MS-CHAPv2 hashes the password field at the client, so the OTP must be sent in the *username* field. Consequently, your access point won't ever see the real user names,

---

10 As of 2012, PPTP tunnel security using the MS-CHAPv2 protocol has been completely broken. Consider immediately changing to some more secure technology.

instead it will see the OTP strings. To support 802.1x authentication you need to use IAS/NPS. Security Note[11]

- If you are using a PPTP VPN tunnel, you must use MS-CHAPv2 authentication. MS-CHAPv2 hashes the password field at the client, so the OTP must be sent in the *username* field. NOT RECOMMENDED[12]

- If the VPN (or front end) server needs to do its own policy checking or logging based on the entered username, then you won't be able to use MS-CHAPv2 because the real username is not ever sent to the VPN (or front end) server. You'll have to choose one of the PAP modes.

---

11 The security of industry standard 802.1x wireless authentication is not affected by the 2012 breaking of the MS-CHAPv2 protocol because the entire tunnel is independently encrypted by PEAP first.
12 As of 2012, PPTP tunnel security using the MS-CHAPv2 protocol has been completely broken. Consider immediately changing to some more secure technology.

Using IAS/NPS for RADIUS with AuthLite

Beginning with AuthLite version 1.2, a plug-in for Microsoft's IAS (also called NPS) RADIUS service is available. Activating this plug-in automatically makes IAS/NPS AuthLite-aware. Simply activate the plug-in and then use the IAS/NPS configuration panel to set up your connection policies.

Procedure

- Open the AuthLite Configuration application on the Domain Member Server you wish to set up as a RADIUS server. (Before version 2.0.62 it was a requirement to use a DC).

- Under Service Configuration, select the "IAS/NPS Plugin" item

- Select the "Enable IAS/NPS support on this server" checkbox

- To allow more flexibility of RADIUS clients, you can select the "Permit requests that don't send the domain name."

- Since Microsoft's IAS/NPS configuration dialogs are not AuthLite-aware, there is one additional setting you must select here. It controls how PAP requests will be processed.

  - One-factor (OTP in password field): In this mode, the server expects the username in the username field, and an OTP in the password field. This is the configuration you want to use if AuthLite is being used to validate **only** the OTP factor, and another process is being used to authenticate the user's name and password. For example, this is how Citrix and Juniper's two-factor authentication works.

  - Two-factor (OTP and Password both included): In this mode, the server expects to see both an OTP and a password included in the request. The OTP can be in the username field, or combined together with the plain text password in the password field[13]. This is the configuration you would use when you want IAS/NPS to authenticate both factors together.

- Apply changes

- **Restart the AuthLite service and also the IAS/NPS service**. Changes are only applied after the services restart.

Notes

- You must set up an appropriate policy in IAS/NPS to allow connections from the RADIUS client of the proper authentication type.

- You do not need to select between PAP and MS-CHAPv2 anywhere in the AuthLite interface, but the policy you configure on IAS/NPS will allow you to select between these settings.

---

13 The reason for this flexibility is that some VPN servers need to see the username in order to enforce their own policy independently of the RADIUS server, or to do their own logging. But if your server does not need to know the username, then your users can enter OTP/password into the VPN client and save effort.

Microsoft VPN Client settings

AuthLite can work with the pre-installed VPN client that ships with Windows, and requires a minimum of configuration on the client side.

Username field retention

The Microsoft VPN client saves the last entered value from the username field by default. If you are using the Username field for OTP entry, then you may wish to change this behavior (so the user doesn't have to remember delete the string each time to enter a new OTP).

If you manage the workstations, you can set the following registry value:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\Parameters]
"DisableSavePassword"=dword:00000001
```

Although its primary use is to disable the option of saving VPN passwords, it also has the effect of clearing the username field.

Wireless 802.1x Authentication

AuthLite supports 802.1x authentication through its NPS RADIUS plug-in.

Prerequisites

- We do not provide extra client-end software to add 802.1x support to legacy workstations. It is assumed that your client machines are capable of performing wireless authentication, such as is possible with Windows 7. Alternately, you could use a third party solution such as NetMotion Mobility XE.

- It is assumed your Access Point is capable of supporting authentication with RADIUS with PEAP/MS-CHAPv2.[14] We use DD-WRT in our validation testing.

- Microsoft NPS as a RADIUS server. It should be installed on a Domain Controller.

- Windows 7 client machine whose Trusted Root certificate store trusts the certificate your NPS server is using.

- Users you wish to authenticate are Active Directory users. AuthLite will use AD for the password portion of the authentication.

Configuration

- Start **without** AuthLite. Get 802.1x wireless authentication working between your client workstations, the access point, and the NPS server. You should be able to type your username and password into the wireless authentication prompt on your workstation, and be authenticated and connected to the wireless network. Before you can add AuthLite, you need the basic setup to be working.

- Additional Windows 7 client settings to work with AuthLite:

   - In the Security tab of the network, deselect "Remember my credentials"

---

14 The security of industry standard 802.1x wireless authentication is not affected by the 2012 breaking of the MS-CHAPv2 protocol because the entire tunnel is independently encrypted by PEAP first.

(OTPs will only be valid one time!).

- In PEAP settings, select EAP-MSCHAPv2 for the authentication method and configure it NOT to use the Windows credentials automatically (we want to enter OTPs).

- In Security->Advanced, select "user" authentication.

- AuthLite configuration settings:

  - Make sure you have your AuthLite users reflected in one of your [AuthLite User Groups]

  - Enable the IAS/NPS plug-in

  - Restart the AuthLite and NPS services

Testing

After users are in an [AuthLite User Group] and the [NPS plugin] is active, those users will no longer be allowed to authenticate with just username and password. *Instead of entering a username:*

- *For YubiKey users,* tap the OTP into the username field of the wireless authentication pop-up.

- For OATH token users, type in your username followed by a dash "-" followed by the OTP from your OATH token app.

Then enter the password as normal, and you should get authenticated to the network.

DirectAccess and NetMotion Mobility XE

AuthLite is very simple to use with DirectAccess or NetMotion Mobility XE, no special configurations are needed. In a properly functioning DirectAccess/Mobility setup, the workstations operate as though they are always on the LAN.

Procedure

- Get your DirectAccess/Mobility infrastructure set up and working first **without** AuthLite being used at all.

- Install AuthLite on the domain controllers and workstations.

- License AuthLite

- Identify AuthLite Users and Provision their keys.

- For Mobility, be sure to configure machine authentication as well as Windows user authentication.

- For Mobility, AuthLite must be installed on the NetMotion Mobility server.

- At the time the user logs in, the workstation will already have the machine-level DirectAccess/Mobility tunnel up and running. AuthLite communicates with the DC over this channel automatically, and performs OTP validation exactly as if the machine was connected directly to the LAN.

- Be sure to use one of the methods for Requiring Two-factor Authentication. **Otherwise your setup may *support* 2FA but not *enforce* it.**

- For the best sign-on experience, deploy this registry key on your Netmotion workstations:

> Key: HKLM/Software/Policies/Collective Software/AuthLite
> Name: CredprovUseNetmotionOrder
> Value(string): "true"

Notes

- DirectAccess is a complex thing to get set up from scratch, especially if you are trying to do it without the UAG wizard. You should definitely **not attempt to add AuthLite** into the mix until you are finished troubleshooting any DirectAccess issues. Otherwise you'll make it harder to figure out where the problems are.

File Servers

Prerequisites

The instructions below assume you have working shared folders already. Please verify the following points:

- Using a non-AuthLite user who has permissions to do this, verify you can connect and that your credentials give you access to the file server / shared folder.

- If you don't have this working as above, then adding AuthLite will only make things harder to troubleshoot. If you contact us for support the first thing we will do is try a connection with a non-AuthLite user to confirm your end-to-end setup is configured properly.

AuthLite setup for Shared folders

- To require 2-factor for AuthLite users to all shares on the system, simply deploy a group policy Denying network access to the AuthLite 1-Factor Tag(s). Or,

- Set Allow/Deny ACLs on individual shared folders to take advantage of AuthLite Group Pairs.

Using Shared folders with AuthLite

In general if your file servers use Kerberos, then they should pick up the 2-factor logon from your workstation's Kerberos ticket.

If the file server must use NTLM, then adding it to the Forced 2-factor Computers section of the AuthLite configuration will force explorer to re-prompt for credentials. When the authentication prompt appears, tap your AuthLite key *into the Username field.* OATH token users should enter their username followed by a dash "-" followed by the OTP from their token.[15]

---

15 The password field is hashed by the NTLM protocol, so it cannot be used to enter OTPs.

Supporting Remote Desktop logons

This section assumes a direct RDP connection or publishing point. For Remote Desktop Gateway scenarios see the next section.

Prerequisites

The instructions below assume you have a working RDP configuration already. Please verify the following points:

- Using a non-AuthLite user who has permissions to do this, verify you can connect and that your credentials log you in to the Terminal server / remote desktop system seamlessly.

- If you don't have this working as above, then adding AuthLite will only make things harder to troubleshoot. If you contact us for support the first thing we will do is try a logon with a non-AuthLite user to confirm your end-to-end setup is configured properly.

AuthLite setup for RDP usage

- Install AuthLite on the domain controllers.

- Install AuthLite on the Terminal Servers / Remote desktop systems. These systems must be domain members. AuthLite is not needed here for authentication, but will make the password-change dialog work better in cases where the password is expired at logon.

- Make sure all AuthLite users are represented in an AuthLite User Group and have tokens.

- Use one of the methods of Requiring Two-factor Authentication. The best way is to assign Group Policy Allow/Deny permissions based on the AuthLite User Group Pairs feature. Or, add the Terminal Servers / Remote desktop systems (or groups containing them) into the Forced 2-Factor Computers list. Or, select "Remote Desktop Authentications" in the terminal server's Forced 2-Factor Processes section. This will make sure that AuthLite users can only connect if they enter a valid OTP and password.
  NOTE: Non-AuthLite users will **not** be blocked by this setting. To prevent certain non-AuthLite users from connecting, simply use group or local policy to restrict the users and groups allowed to log on.

- An RDP connection using Network Layer Authentication requires two sequential authentication events to establish your session. You need to set a Replay window value greater than zero so one entered OTP can be used for both of the authentications needed to establish your session. Place the RDP servers into the top list of the Replay Window marked "These computers may initiate & share the window".

- Also, if your client is on the LAN, its first authentication will go directly to the DC, so in addition to having the RDP server in the replay window, you should include all possible RDP client IP ranges in the Replay window's bottom list marked "These computers may only initiate the window".

Using RDP with AuthLite

To log in to the remote desktop server:

- Launch the mstsc.exe client and specify the terminal server you are connecting to
- Tap your AuthLite key *into the Username field*. OATH token users should enter their username followed by a dash "-" followed by the OTP from their token.[16]
- Enter your password into the password field
- Connect

---

16 The password field is hashed by the NTLM protocol, so it cannot be used to enter OTPs.

Supporting Remote Desktop Gateway

If you publish a Remote Desktop Gateway to your extranet, you can configure the servers to support AuthLite users.

Prerequisites

The instructions below assume you have a working RDG/TSG configuration already. Please verify the following points:

- A Remote Desktop Gateway is set up and configured with appropriate policies

- One or more Terminal Servers or Remote Desktop services configured

- If possible, set the RDP proxy settings to use the same credentials for the RDG and the terminal server. This will enable you to enter one username and password and connect all the way through to the target system.

- Using a non-AuthLite user who has permissions to do this, verify you can connect from the Internet, through the RDG, and that your credentials log you in to the Terminal server / remote desktop system seamlessly.

- If you don't have this working as above, then adding AuthLite will only make things harder to troubleshoot. If you contact us for support the first thing we will do is try a logon with a non-AuthLite user to confirm your end-to-end setup is configured properly.

AuthLite setup for the RDG scenario (2012R2 and later)

See this video:

Tutorial Video!
[AuthLite with 2012 R2 RDG](#)

AuthLite setup for the RDG scenario (pre-2012R2)

- Make sure all AuthLite users are represented in an [AuthLite User Group](#).

- AuthLite should already be installed on domain controllers.

- Install AuthLite on the RDG system. The RDG server must be a domain member.

- Install AuthLite on the Terminal Servers / Remote desktop systems. These systems must be domain members. AuthLite is not needed here for authentication, but will make the password-change dialog work better in cases where the password is expired at logon.

- Configure some method for [Requiring Two-factor Authentication](#). The best way is to use the [AuthLite User Group Pairs](#) feature. You can set up your gateway to only allow users who have a 2-factor tag group in their session.

- Alternately you could add the Terminal Servers / Remote desktop systems (or groups containing them) into the [Forced 2-Factor Computers](#) configuration list. Or, select

"Remote Desktop Authentications" in the terminal server's [Forced 2-Factor Processes](#) section. This will make sure that AuthLite users can only connect if they enter a valid OTP and password.
NOTE: Non-AuthLite users will **not** be blocked by this setting. To prevent certain non-AuthLite users from connecting, simply use group or local policy to restrict the users and groups allowed to log on.

- If possible, set the RDP proxy settings to use the same credentials for the RDG and the terminal server. This will enable you to enter one OTP and password and connect all the way through to the target system.

- An RDG connection requires several sequential authentication events to establish your session. You need to set a [Replay window](#) value greater than zero so one entered OTP can be used for all the authentications needed to establish your session. We find about 10-20 seconds is usually sufficient in production. Slow/testing environments may require longer. Place the RDG and RDP session host servers into the top list of the Replay Window marked "These computers may initiate & share the window".

- Also, if your client is on the LAN, its first authentication will go directly from client to DC, so in addition to having the RDG and RDP servers in the replay window, you should include all possible RDP client IP ranges in the Replay window's bottom list marked "These computers may only initiate the window".

Using RDG with AuthLite

To log in to the remote desktop server:

- Launch the mstsc.exe client

- In advanced settings, specify the external address of the RDG proxy server

- Select to use the same credentials for the proxy and the terminal server

- Specify the terminal server you are connecting to

- Tap your AuthLite key *into the Username field.* OATH token users should enter their username followed by a dash "-" followed by the OTP from their token.[17]

- Enter your password into the password field

- Connect

---

17 . The password field is hashed by the NTLM protocol, so it cannot be used to enter OTPs.

Exchange/Outlook 2013 Single sign-on

Starting with Exchange 2013/Outlook 2013, it is configured to always use RPC/HTTP instead of the local direct RPC, even when connected to the LAN. Even when logged on to the desktop with Kerberos.

The problem is that by default it's configured to use NTLM even in this case, and Microsoft does not make it easy to change. Since AuthLite uses an OTP, the NTLM authentication will fail if your Exchange server demands 2-factor logon. So your user will get a credential popup when opening Outlook.

To correctly use the desktop's 2-factor Kerberos ticket, you need to get your Exchange environment to use Kerberos when available. This is 100% about configuring first-party Microsoft software, so is technically out of scope for AuthLite. But Microsoft does not make it easy to change. So here is some guidance:

- The services must be told to accept Negotiate auth instead of just NTLM. Something like:

```
Get-OutlookAnywhere -Server $computername -ADPropertiesOnly | Set-OutlookAnywhere
-InternalClientAuthenticationMethod Negotiate -IISAuthenticationMethods Negotiate
```

- This does not take effect immediately. You need to wait for the application event log to record a new event ID 3036, which means the configuration has updated the correct items in IIS

- And, the client outlook needs to have its mode set to Negotiate too. Reg key for that in Outlook 2013:

  HKCU\software\microsoft\office\15.0\outlook\rpc

  Value name:

  proxyauthenticationservice

  Value string:

  Negotiate

Potentially of use:

- http://blogs.technet.com/b/microsoft*exchange*tips/archive/2013/05/28/steps-to-configure-kerberos-authentication-for-the-exchange-2010-cas-array.aspx

- https://support.microsoft.com/en-us/kb/2619234

Supporting Outlook on the Extranet

If you publish Outlook Anywhere to your extranet, you can configure the servers to support AuthLite users.

Prerequisites for Exchange 2007/2010

The instructions below assume you have a working OA configuration already. Please verify the following points:

- A working Exchange organization

- IIS and Exchange front end server

- Outlook Anywhere configured on the server, set to use NTLM authentication (Note: this option also allows Kerberos Constrained Delegation, which is what we'll use for the case of publishing from ISA/TMG server)

- IIS on the Exchange front end server should have the RPC folder's authentication set to Integrated.

- RPC/HTTP configured and working.

- (Optional) OA published through a domain ISA/TMG server, listener pre-authenticating in HTTP/Basic mode, and rule delegating to the IIS server using Kerberos Constrained delegation.

    - If you are not pre-authenticating at an ISA/TMG publishing point, you can still use AuthLite with OA; just ignore the parts of the below setup that deal with ISA/TMG server, and set the Outlook proxy authentication to NTLM instead of Basic.

    - It is not possible to pre-authenticate an OA connection at an ISA/TMG server that is a workgroup machine, it *must* be a domain member.

- Set the Outlook proxy to the public interface of your publishing point, and select Basic authentication (if not using ISA/TMG, select NTLM authentication).

- Using a non-AuthLite user who has permissions to do this, verify you can connect from the Internet, through ISA/TMG server, through RPC/HTTP, and that your credentials log you in to Exchange seamlessly.

- Outlook Anywhere is hard to set up! If you don't have this working as above, then adding AuthLite will only make things harder to troubleshoot. If you contact us for support the first thing we will do is try a logon with a non-AuthLite user to confirm your end-to-end setup is configured properly. Here are some resources that we have found helpful in troubleshooting:

    - How to Install Exchange 2007 SP1 and SP2 Prerequisites on Windows Server 2008

    - How does Outlook Anywhere work (and not work)?

    - How to use the RPC Ping utility

AuthLite setup for the Outlook/Extranet scenario

- Make sure all AuthLite users are represented in an AuthLite User Group.

- Install AuthLite on the DCs, and Exchange front end server containing IIS, RPC/HTTP, and the client access services.

- In the Forced 2-factor Processes list of the front end server, specify strings to match the Outlook Anywhere application pool: "RPCClientAccess" and "MSExchangeRPCProxy". This is the setting that forces AuthLite users to submit 2-factor credentials. **Without this setting, Outlook could tolerate 2FA but it would not enforce it! (**If you enforce the entire system with Group Policy, then this is not required. But if you did that, then you'd be breaking ActiveSync.)

- Set Replay Behavior to Retry, so that the Exchange back-end servers can use stale OTP credentials without balking. This is the default in recent AuthLite build installations.

- Outlook Anywhere 2010 will send your credentials repeatedly over the course of the session. You need to set a very long [Replay window](#) value for your front end server one entered OTP can be used for all the authentications needed to establish your session. After the window duration expires, Outlook will prompt for new logon credentials again.

- Outlook 2013 always uses "outlook anywhere" (rpc/http) but it seems to use the credentials just during session startup, so you should specify a short replay window as you would with RDP, e.g. about 20 seconds.

Using Outlook 2010 on the Extranet with AuthLite

To connect with Outlook Anywhere:

- Go into the advanced connection settings of your Outlook's email account

- Specify the external address of the OA proxy

- Select Basic authentication if using ISA/TMG to pre-authenticate. Select NTLM if authenticating directly to IIS.

- When prompted for credentials, tap your AuthLite key *into the username field*. OATH token users should enter their username followed by a dash "-" followed by the OTP from their token.[18]

- Enter your password into the password field

- Connect

- With Exchange/Outlook 2010, after the [Replay window](#) expires, you will be re-prompted for new credentials. Enter a new OTP and your password to continue.

18 . The password field is hashed by the NTLM protocol, so it cannot be used to enter OTPs.

Replay windows to support re-authenticating protocols

When a user wants to connect to a service, they enter their credentials one time, and expect the software to use these values as many times as needed in order to log on and maintain their session. Some network protocols such as Outlook Anywhere and Terminal Services authenticate at each hop along the way from the client to the final destination server. Outlook opens and closes connections periodically, attempting to use the same credentials that the user entered to log in each time. HTTP "basic" authentication provides the same credentials for each connection.

Since AuthLite is a "one-time passcode" system, normally every attempt to use the same OTP again would result in the request being denied. In order to support multiple-authentication protocols, we provide a way to tell AuthLite to allow a user's latest OTP entry to be "replayed" for a certain amount of time without denying the authentication attempt.

This setting is controlled by the "Replay Windows" item of the AuthLite configuration tool.
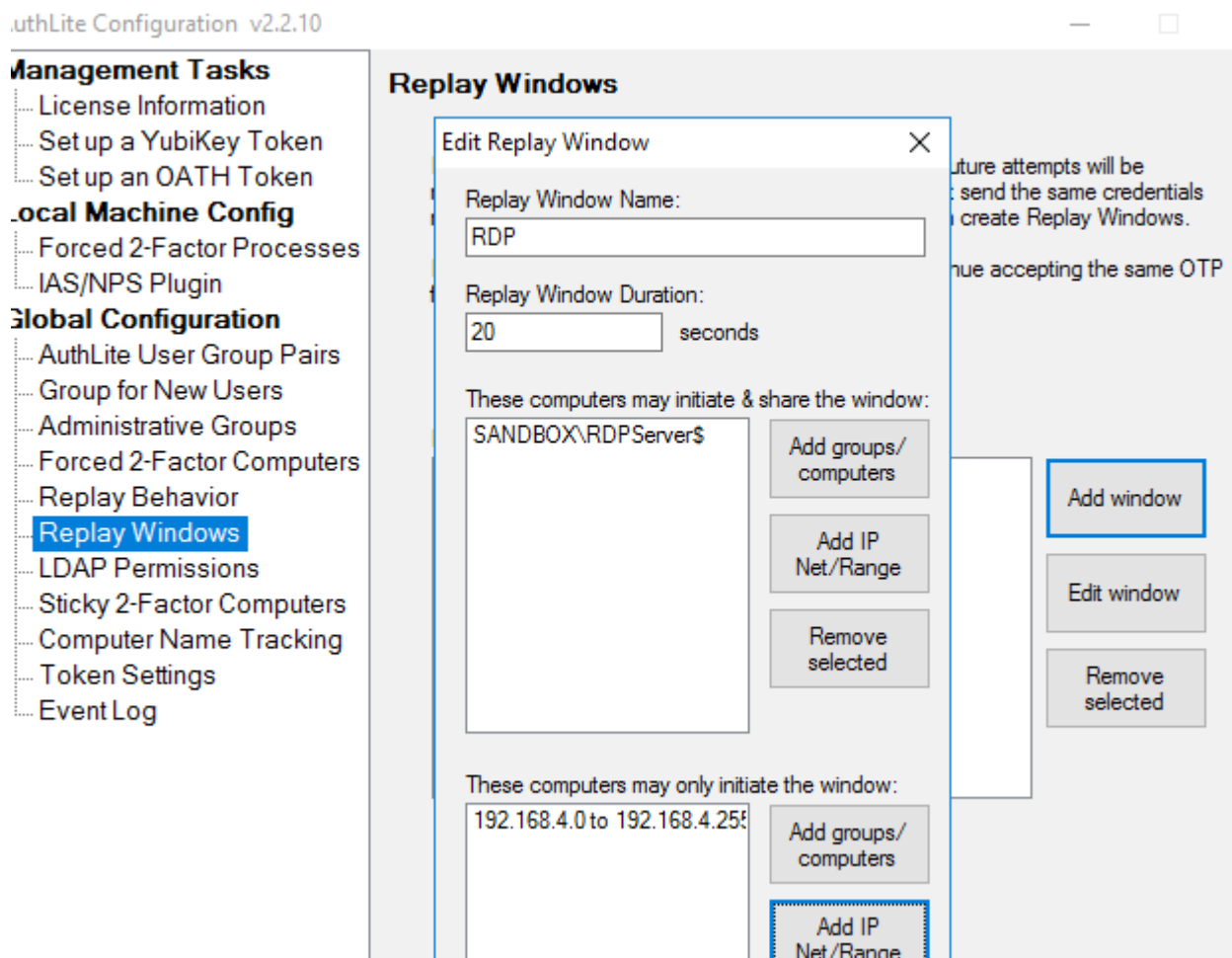
Each Replay Window you define will apply to a set of one or more computers or IP address/ranges on the network, and they are enforced by the domain controllers.

For a concrete example: In an RDG scenario you would specify each server that will participate in the authentications together: the RDG server and all its possible terminal

servers should share *the same* Replay window. Add them to the top list marked "These computers may initiate & share the window". Also if clients are coming from the LAN, then you should also add the IP range of possible clients to the bottom list marked "these computers may only initiate the window". You can use groups or IP ranges rather than placing individual computer names into the list directly. You can specify servers and IPs in more than one replay window if needed.

NOTE: As with most AuthLite settings, Replay window configuration can take 20 minutes to propagate to all systems.

Security considerations

The window mechanism only allows a limited replay on a user's most recently entered OTP. So no matter what the size of your replay window you need not be concerned about previously-entered OTPs being used again maliciously. Only the freshest, most recently entered OTP is allowed to authenticate repeatedly during the window, and as soon as the user enters a new OTP any time remaining on the old window is canceled.

A short replay window such as 10-20 seconds does not notably diminish the security of an OTP system against most types of attacks. However if an adversary can launch immediate parallel sessions from your machine or in some automated, instantaneous fashion, then any replay window at all can allow impersonation. If you are not using any multiple-authentication protocols with AuthLite you can run without any replay windows configured, disabling this behavior completely.

A longer replay window such as is needed for Outlook Anywhere on Exchange 2010, or HTTP Basic authentication, decreases some of the benefits of a one-time passcode system. An attacker who is able to capture the user's OTP and password would have the ability to use these credentials *during the replay window*, and impersonate the user. A long window gives an adversary more leisure to perform an impersonation attack, and requires less sophistication. The dual credentials are still far more secure than a plain one-factor password, and the vulnerability is still time limited, but this is not *as* secure as using a small (or no) replay window.

RDG/RDP settings

For an RDG scenario, a short window value of 20 seconds is probably long enough to allow all the authentications (RDG, RPC, Network Layer Authentication, remote desktop session) to complete. And after those initial connections, no more authentications need to be performed. If you are not able to connect, check your [AuthLite Log](#) on your DCs for recent replay events, and increase the associated replay window if needed.

Also, if your client is on the LAN, its first authentication will go directly to the DC, so in addition to having the RDP server in the replay window, you should include all possible RDP client IP ranges in the bottom list marked "these computers may only initiate the window"

Outlook Anywhere settings

For Outlook Anywhere 2010, several connections will be opened and closed throughout the user's session, and Outlook will keep using whatever credentials the user most recently entered. After the OTP replay window expires, Outlook's new connections will fail, and the

user will see a pop-up dialog requesting new credentials.

So the length of your OTP replay window will effectively determine how long a user can use OA before needing to enter a new OTP and their password into the authentication pop-up. Set the window long enough so as not to be overly annoying, but short enough to mitigate the threat of an attacker recording and re-using the credentials later. For example 30 minutes (set as 1800 seconds) is a reasonable session length, but 8 hours would probably be irresponsibly long.

Outlook 2013 seems to do a better job keeping existing connections alive. Best practice is to only force 2-factor authentication on the front end RPC/HTTP server, and configure Replay Behavior to allow 1-factor retry of stale 2-factor credentials, so the back end servers can re-use the entered credentials.

Replay Behavior setting

One thing we have not yet considered is what behavior should occur when a two-factor authentication attempt is a non-allowed replay. In old installations, AuthLite will force the authentication to fail right at the DC, and return an error message. This behavior was only the default for historical reasons, however it is unlikely to be the way you want to keep it.

There are many situations where credentials are stored and sent repeatedly by well-meaning client software. Each time you authenticate to a new NTLM resource, the credentials you entered at login-time are being sent AGAIN to the server. For one-factor passwords this is a detail you never need to think about, but with one-time credentials it can be a problem. If several stale credential attempts are made in a short interval, the account could get locked out even though the user hasn't done anything wrong.

So here is where we come to the Replay Behavior configuration. You can tell AuthLite that when a two-factor authentication is rejected as a replay, to discard the OTP and try again as if it was a simple one-factor logon instead. This allows services that don't need 2-factor security to tolerate seeing stale 2-factor credentials, and treat them as if they were just normal one-factor password-only logons.

This is the new default for recent installations. It effectively puts off the decision about whether to block the logon until later. If there's a group policy blocking a 1-factor Tag group, or the authentication is coming from a Forced 2-factor Computer, or on a Forced 2-Factor Process, then it will be blocked. But if all those things are false, then you are effectively saying that for this server/process/resource you don't *need* two-factor authentication, and the 1-factor logon will be allowed.

Some customers mistakenly set this setting to "Fail" because they believe it's the only way that 2-factor will ever be enforced. But really it just makes the authentication fail so soon that most of the more nuanced enforcement mechanisms will never be reached. It's a sledgehammer setting that is almost never appropriate for a modern installation.

AuthLite Configuration v2.0.35

Management Tasks
- License Information
- Program an AuthLite Key
- Set up an OATH Token

Local Machine Config
- Forced 2-Factor Processes
- IAS/NPS Plugin

Global Configuration
- AuthLite User Group Pairs
- Group for New Users
- Forced 2-Factor Computers
- Replay Behavior
- Replay Windows
- Key Programmers
- LDAP Permissions
- Computer Name Tracking
- Sticky 2-Factor Computers
- Token Settings

Replay Behavior

How should AuthLite treat authentications that contain rejected OTP replays?

The default is to reject the entire authentication attempt as if the password entered was incorrect. This is the most secure option.

However, in some cases the authenticating computer may not require 2-factor logon. So it may be preferable to allow the "failed" 2-factor auth to be retried as if it was sent as a 1-factor password-only authentication.

Note: If the authenticating computer does not allow 1-factor logons, then a rejected replay OTP will always cause the whole authentication to be rejected.

When a replayed OTP is rejected:
- ○ Make the authentication attempt fail (default)
- ● Retry as if it was a 1-factor authentication attempt

Apply changes

This setting does not decrease security, and all your two-factor policies and security settings will still apply the same way; it is simply a new feature designed to make life easier to authenticate to non-secure services, and prevent accidental lockouts.

Support for Other Configurations

A rich ecosystem of applications all rely on the built-in authentication mechanisms of Active Directory. With AuthLite we have tried to support the most common cases as transparently as possible. If you have difficulty using an application with AuthLite, there are a number of approaches we can take to support your needs, as appropriate:

- Troubleshooting-- If a configuration is supposed to work and does not seem to, the first thing we will do is look at event logs to try to determine what the authentication looks like, and where the failure occurs. For successful authentication, many components must work together seamlessly.

- .NET API-- It is possible to provision and validate AuthLite users through a .NET interface, enabling custom administrative tools to be used. The AuthLite Token Profile Manager intranet application is one such example of an application that uses the API.

- Windows / Application API hooking-- We can help you to determine what method or API your application is using to authenticate to AD, and possibly create a hook to modify this behavior to be AuthLite compatible.

Please open a support request, or contact your representative to get started.

Appendix A: Account Recovery Considerations

Active Directory Accounts

In an Active Directory domain, AuthLite user accounts can be administratively reset through the normal AD Users and Computers console, without data loss. The directory has built-in technology to avoid the loss of EFS/Bitlocker and certificate access that will occur for standalone accounts. Therefore, there are no special precautions you need to take in an AD environment to protect your normal users from account and data loss.

It is definitely advisable to have at least one domain administrator account that is *not* an AuthLite user, in the event that you need to perform logons or operations where AuthLite software is not installed or is functioning improperly.

Special note: If you have restricted access to your domain controllers by policy to domain admins with a Two-Factor tag (via AuthLite User Group Pairs) then you should **also** explicitly allow logon from at least one non-AuthLite domain admin. Consider the case where AuthLite software gets uninstalled and the DC gets rebooted. No accounts will have the Two-Factor group tag, since AuthLite is no longer running. So it is crucial that you not accidentally lock yourself out of being able to log in to your own DC's. An emergency recovery domain administrator account is useful in this instance.

Be skeptical of the security of any logon product that offers a way to log on without using all of the security factors! Any time there is a way to bypass the highest level of security, that means an attacker could use the lower security method to compromise your account more easily. This is a common security precept known as "low hanging fruit" or the "path of least resistance". Offering a lower security recovery option to the user means that the high security is effectively *optional*, and thus it will not provide any strength against a smart attacker.

Appendix B: Active Directory Deployment Notes

Licensing

Enter the license code on one Domain controller where AuthLite is installed, and it will propagate to other DC's via replication. Member servers and workstations will read the license value from the directory as well.

Due to replication delays, some servers may temporarily still believe they are unlicensed.

Software Installation

In order to authenticate AuthLite users, a Domain Controller must have the AuthLite infrastructure software installed. If a workstation (or member server) connects to a DC where AuthLite is *not* installed, all the AuthLite operations will fail on that machine, and AuthLite users will *not* be required to use two-factor authentication, since this enforcement is done at the domain controller.

Domain Controllers with AuthLite software installed will still function normally for all their normal roles, including authentication of non-AuthLite users. AuthLite does not replace or remove built-in Microsoft components or behaviors.

Application Partition (database)

AuthLite takes advantage of the robust, multi-master replication offered by Active Directory by using an Application Partition to store all its user data and domain-wide settings. This is the same method that Microsoft's own DNS server uses to manage its data.

The first installation of AuthLite on a Domain Controller in your enterprise will automatically create this partition, and the schema additions necessary to support it. Thus, the first DC you install AuthLite on will be a replication host for the partition, by default.

AuthLite does not add or change any schema properties on the "user" or other built-in objects in Active Directory. All AuthLite data is stored separately in the AuthLite Application Partition. Adding the AuthLite schema elements will have *no performance impact* on user object replication since we don't touch those objects at all.

Replication hosts

By design, AD Application Partitions do *not* automatically replicate to every DC in your enterprise, because it is assumed that the data they contain may only be needed by a subset of the enterprise. If a DC receives a directory query for a partition that's not stored locally, it will refer the request to a remote DC that stores the partition.

This referred connection can be slow if the remote DC is in another site. Since AuthLite requires access to its partition in order to operate, any AD site where AuthLite is used should have at least one DC that hosts a replica of the AuthLite data partition.

Also, to support redundancy if the first partition host goes offline, you may wish to host the partition on more than one Domain Controller in the same site. (Without access to the AuthLite partition, all AuthLite operations will not function.)

You can easily specify whether a DC should host a replica the AuthLite partition at install time, by enabling or disabling the installer feature "Create a Data Store Replica on this DC".   This

feature is selected by default when installing AuthLite onto a DC. Thus if you do not change the selection, every Domain Controller where AuthLite is installed will also have a replica of its data partition, which is a reasonable default.

This option in the AuthLite installer doesn't perform any proprietary operations when making a replica, it is just a convenience. You can also use the Microsoft command **ntdsutil** to control the replication hosts for Application Partitions. The partition's distinguished name will be

```
DC=AuthLite,[your-domain's-dn]
```

Content

Although not necessary for normal operation, you can browse and change the partition content with the Microsoft MMC plugin **adsiedit**.

To browse the AuthLite key data, you can use the [AuthLite Token Manager](#) application installed on the domain controllers.

Deletion

Uninstalling AuthLite does not remove the data partition or affect which DC's host its replicas. This way, if AuthLite is reinstalled, all the existing data can be used again, and users can continue to authenticate. If removal of the partition is desired, this can be accomplished with the Microsoft **ntdsutil** command.

Appendix C: Kerberos Constrained Delegation Notes

Kerberos Constrained Delegation is a useful Microsoft extension that can allow IIS web servers to trust credentials provided by ISA/TMG server, even in scenarios when the user does not ever provide their password to log on to the Kerberos (AD) domain in the first place. AuthLite makes use of this technology in the following scenarios:

- Mobile devices that must log on to extranet services with only username and password.

- Extranet services that use only the username and OTP for logon. The password is never entered, so the users cannot be logged in to AD.

In these cases, ISA/TMG Server (with AuthLite installed) can verify the identity of the user, but there is not enough information provided to get an impersonation logon token from AD. The best ISA/TMG can do is get a Kerberos ticket via the "Service for User" (s4u) system. This kind of ticket provides a useful token that the local machine can use, but we still need a way to tell the remote IIS machine about the user.

This is where KCD comes in. ISA/TMG can take the s4u ticket and use it to get another ticket that will work on the remote service (for example the IIS server on your front end Exchange server). This will only work for the specific services that ISA/TMG's computer account is allowed to delegate to (this is why it's called "constrained"). The target service (for example an IIS Server hosting OWA) is trusting ISA/TMG to do a good job of authenticating the users.

KCD technology is built in to AD, ISA 2006/TMG and IIS. You can use it instead of other delegation options in ISA/TMG even when full credentials are available. In the above scenarios for AuthLite, however, there is no other choice than to use KCD, because full user credentials are never provided. You are choosing to say, in effect, even though we don't have complete information to do an authentication to Active Directory, the information we *do* have is sufficient to authenticate the user for the purposes of using these Extranet services. It is a matter of the security policy of your organization, the threat models you are addressing, and the usability of the Extranet services.

Several things must be configured correctly to use KCD. These requirements are based on the way Microsoft has implemented the technology, and don't have anything to do with AuthLite, per se. An excellent resource for KCD configuration between ISA and IIS (Exchange) is the article [Kerberos Constrained Delegation in ISA Server 2006](#).

Appendix D: Deploying the AuthLite Token Profile Manager intranet application

Prerequisites

- AuthLite version 2.2 or later on domain controllers

- A domain member IIS web server, running AuthLite version 2.2 or later

- Microsoft .NET framework version 4.5 or newer on the IIS server

Installation and Configuration

- If the .NET framework was installed **before** the IIS role, run: http://msdn.microsoft.com/en-us/library/k6h9cz8h.ASPX , or else your ASP.net will not be registered properly.

- Obtain AuthLiteTokenProfileManager.zip from AuthLite.com/downloads or contact support if it is not listed on the web page.

- On IIS, go to registry key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters , create value named "CacheS4UTickets", DWORD=0 . Without this, the portal will make incorrect decisions about whether users are in the AuthLite Users groups.

- Create and bind an SSL site on IIS to use for this portal. Details of binding certificate and host name are omitted.

- In the physical path of the above site you created, drop the files from the AuthLiteTokenProfileManager zip file.

- In the root of the web site, edit web.config to set proper values for your environment.

- Visit the base URL for your site and you should see a prompt to enter a username and continue

Appendix E: Using Group Policy to deploy software/settings

Administrative Template for settings

In a domain environment, most AuthLite settings are stored on domain controllers, in the data partition. These settings are automatically applied by all AuthLite-aware systems as needed. But certain settings are server-specific and stored in the registry.

In order to deploy a per-machine setting administratively to a group of systems, you can use a group policy Administrative Template. Save the following lines as a Unicode text file with an .adm extension:

```
CLASS MACHINE
CATEGORY "SOFTWARE\Policies\Collective Software\AuthLite\<etc>"
  POLICY <name>
  KEYNAME "SOFTWARE\Policies\Collective Software\AuthLite\<etc>\<name>"
    PART <settingname> EDITTEXT
    VALUENAME "<settingvalue>"
    END PART
  END POLICY
END CATEGORY
```

Where the <values in brackets> depend on what setting you are trying to control. You can then load this adm file into the group policy editor, in the "Administrative Templates" section of the Computer Configuration, and assign a value. This setting will then be applied along with the rest of the machine policy.

Note that the AuthLite Service only reads some settings on start up, so changing its values via policy will not have an immediate effect, even if you run "gpupdate" and apply the policy immediately. For other settings there is a 20 minute cache timeout.

Software deployment

See this KB article for information on deploying AuthLite software unattended, via Group policy.

Appendix H: Remote Data Store mode

There are some situations where you need the AuthLite core and service to behave as if it was installed on a domain controller, even though in reality it is not.

In Remote Data Store scenarios you still have to install AuthLite on at least one read/write Domain Controller, in order to create/replicate the data partition and make the configuration and Token Manager tools available. Install AuthLite normally on one or more DC's.

 On the "remote" machine you must then launch the AuthLite core software installer from the command line, using the extra property switch REMOTEDATASTORE=1. For example:

```
AuthLite_installer_x86.msi REMOTEDATASTORE=1
```

or

```
AuthLite_installer_x64.msi REMOTEDATASTORE=1
```

and then configure extra settings in the Remote Data Store section of the AuthLite configuration tool.