ActivePasswords Introduction

ActivePasswords is a compact, lightweight and powerful custom Windows password filter. ActivePasswords makes it possible to manage multiple password complexity policies for a selection of Active Directory security groups and/or organizational units. ActivePasswords also periodically queries your Active Directory for changes and makes those changes available in a practical textual format for processing by various custom scripts.

30-day trial period and Licensing

If no license key is entered ActivePasswords will be fully functional for 30 days after installation. After this period ActivePasswords will stop functioning and your servers will continue to work like they did before installing ActivePasswords. ActivePasswords pricing is based on the number of enabled Active Directory users and is subscription based. A subscription gives you usage rights, updates and e-mail support for 1 year.

Contents of the ActivePasswords.zip archive

ADMX (folder)	This folder contains the custom policy template definitions for ActivePasswords (admx/adml files).
wsap.dll (in MSI)	Password filter DLL that contains special functions called by Windows when a user password is changed.
wsad.exe (in MSI)	AD query tool to retrieve account information and changes.
wssync.exe (in MSI)	Sync tool that is used to run and log any active sync scripts (optional)
pcr.exe (in MSI)	Password Change Request client utility.
Readme.pdf	The file you are reading now.
Agreement.pdf	End user license agreement (EULA)

Prepare your Domain

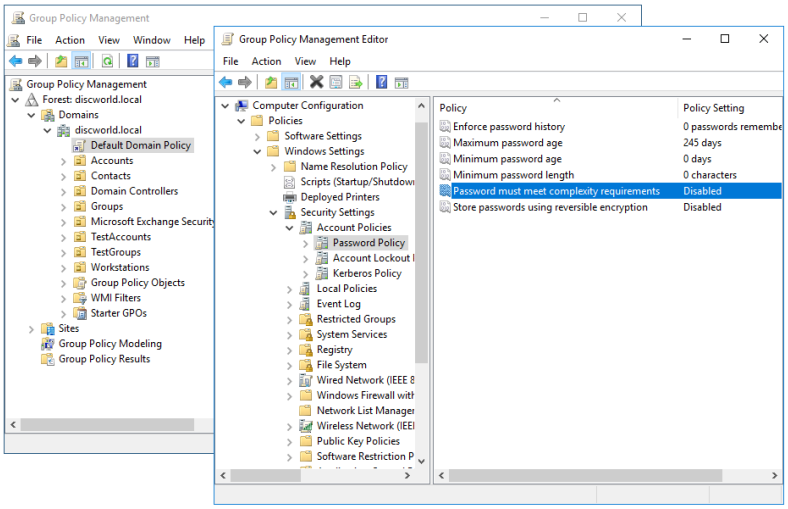
It is a requirement to install ActivePasswords *on every domain controller* (DC) in your Active Directory domain (AD). This is necessary because password changes are sent to one random DC near the domain user that initiated the password change. You achieve this by simply running the silent installer ActivePasswords.msi on all DC's. A manual reboot is necessary to complete the setup. The setup will NOT do this automatically.

The installer performs the following actions:

- Copy wsap.dll, wsad.exe and wssync.exe to %windir%\system32
- Add wsap to the registry value HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Notification Packages
- Copy ActivePasswords.admx and the related .adml file to the central GPO template store
- Create %windir%\wsap and sub folders
- Create ActivePasswordsADQuery task to retrieve AD user changes
- Create ActivePasswordsSync task to run any enabled scripts
- Runs the AD user data query for the first time

Configuration

Step 1: Set the Windows default password settings

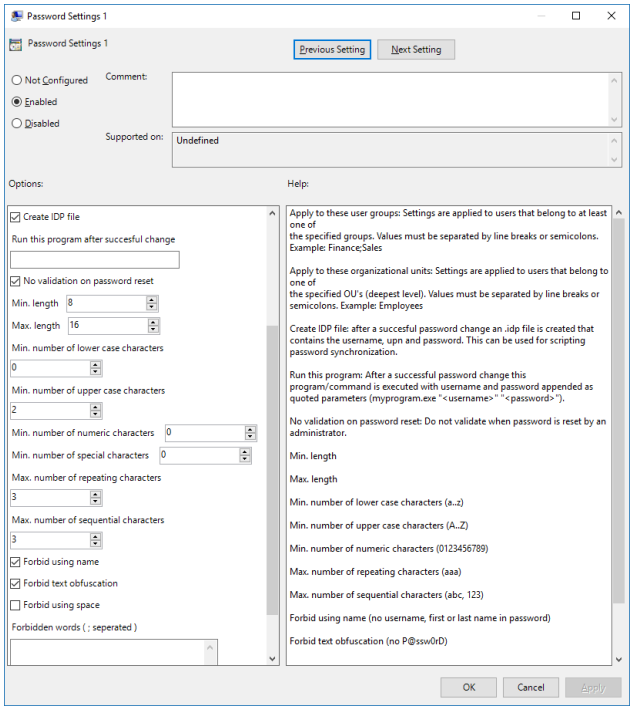


Open the Group Policy Management editor (gpmc.msc) and navigate to Forest/Domains/[DOMAIN]/Group Policy Objects and edit the "Default Domain Policy" object (also linked at the root of your domain). Go to "Computer Configuration/Policies/Windows Settings/Security Settings/Account Policies/Password Policy". Set the history and age options to your liking and *make sure to set "Password must meet complexity requirements" to* ***Disabled*** *and "Minimum password length" to 0*. You will not use the default Windows complexity requirements because ActivePasswords is much more flexible and powerful and replaces this functionality.

If you have configured any 'fine-grained' password settings, make sure to *uncheck* "Enforce minimum length" and "Password must meet complexity requirements". These settings are found in the Active Directory Administrative Center (dsac.exe).

Test these changes by resetting a user password from Active Directory Users and Computers (dsa.msc). A very simple password should be allowed at this moment: the default Windows settings does not interfere anymore with ActivePasswords settings. Of course, we will now make an ActivePasswords policy to force the use of strong passwords.

Step 2: Create ActivePasswords policy



Navigate to the organizational unit (OU) that contains *your* ***domain controllers*** *(not your user accounts!)*. Right click on the OU and choose "Create a GPO in this domain, and Link it here...". Name the policy ActivePasswords. Right click on the newly created GPO and select "Edit". Navigate to "Computer Configuration/Administrative Templates/ActivePasswords".

First, you need to setup some Common Settings: (next page)

Common Settings

Title	Registry Value	Description
License	License	Copy and paste the license you received (the entire line). Leave empty for trial mode. A license has this format: [Company;Domain;Amount;EndDateYyyymmdd;Hash].
Skip these users	SkipUsers	No log entry is created for and no password policy is applied to the specified users.
Skip these groups	SkipGroups	No log entry is created for and no password policy is applied to the specified groups.

You can configure up to 8 Password Settings for different selections of Active Directory groups. To create one, double click “Password Settings 1” and click “Enabled”. A security groups is specified by entering the group name (sAMAccountName attribute). Settings:

Password Settings 1-8

Title	Registry Value	Description
Apply to these user groups	UserGroups	Settings are applied to users that belong to at least one of the specified groups. Values must be separated by line breaks or semicolons. Example: Finance;Sales
Apply to these organizational units	UserOus	Settings are applied to users that belong to one of the specified OU's (deepest level name). Values must be separated by line breaks or semicolons. Example: Employees
Create IDP file	CreateIDP	After a successful password change an .idp file is created that contains the username, UPN and password. This can be used for scripting password synchronization.
Run this program after successful change	NotifyProgram	After a successful password change this program/command is executed with username and password appended as quoted parameters (myprogram.exe "<username>" "<password>").
Base64-encode program parameters	NotifyProgramBase64	The username and password parameters of the specified program will be base64-encoded
No validation on password reset	NoValidateOnReset	Do not validate when password is reset by an administrator.
HaveIBeenPwned check enabled	PwnedEnabled	Use the HaveIBeenPwned.com web service to check if the password is known to be compromised. The password will NOT be checked during password reset events.
Min. length	MinLength	
Max. length	MaxLength	
Min. number of words	MinWords	Password must contain at least the specified number of words (1 or more spaces act as a word separator).
Min. number of lower case characters	LowerCase	a..z
Min. number of upper case characters	UpperCase	A..Z
Min. number of numeric characters	Number	0123456789
Min. number of special characters	Special	~!@#$%^&*()_+{}|:?`-=[]\;',./<> "
Min. number of character categories	CharCategories	Password must contain at least the specified number of character categories (lower case, upper case, numeric, special). This setting has no effect if any of the 'min.

		number of x characters' policies are configured because they conflict with each other.
Max. number of repeating characters	MaxRepeating	aaa
Max. number of sequential characters	MaxSequential	abc, 123
Forbid using name	NoName	No username, first or last name in password
Forbid text obfuscation	NoObfuscation	No P@ssw0rD
Forbid using space	NoSpace	No ‘ ‘
Forbid using vowel	NoVowel	No vowels are allowed (AEIOU)
Only allow these characters	AllowedChars	Password must contain only a selection of these specific characters (e.g. ABCDEFGHIJKLMNOPQRSTUVWXYZ: only upper case letters are allowed)
Forbid these characters	ForbiddenChars	Password must not contain any of these characters (e.g. _#)
Forbid these characters at start	ForbiddenStartChars	Password must not start with any of these characters (e.g. 0123456789)
Forbidden words	Forbidden	Custom prohibited words like company;department;brand (the password myCompany11 will not be accepted). Words must be separated by line breaks or semicolons.
Forbidden words file	ForbiddenFile	Enter the name of a file that contains a list of forbidden words (must be located at %windir%\wsap\words). Words must be separated by line breaks.
RegEx	RegEx	'abC' will pass '[a-z]b[A-Z]'; 'abc' will not
RulesExplanation	RulesText	This text will be displayed to the user when the password almost expires (only displayed when the optional pwcr.exe utility runs on the users workstation). You can use \n to add extra new lines.

After closing the Group Policy editor, you can run the command *gpupdate* to immediately update the configuration of the domain controller. If you did not reboot yet, now is a good moment.

About Have I Been Pwned?

Please read the website www.haveibeenpwned.com for more information. If the “HaveIBeenPwned” check is enabled, a new password will be checked for exposure with this web service. The password hash will be send in a HTTPS POST request. Because of web service performance issues and rate limits the password will NOT be checked during reset events.

Password expire warning

Optionally you can run the small tool PCR (short for PasswordChangeRequest) on client computers that periodically warns the end user starting 8 days before the password expires. The tool can be found in \\domain.local\netlogon\wsap\pcr.exe . You can start this e.q. from a login script by adding the command [start \\domain.local\netlogon\wsap\pcr.exe] to it. For testing purposes you can use the /test parameter which will cause pcr.exe to display the configured message for the current user (if any).

Troubleshooting

If things don’t work the way they should, these suggestions may help you:

- The ActivePasswords group policy must be applied to the OU that contains your domain controllers
- The ActivePasswords policy settings must include the targeted active directory user group(s) (eg mygroup1;mygroup2)
- You can disable ' No validation on password reset ' to make testing as administrator easier (you can reset user passwords as admin from dsa.msc and see if the password is accepted or not)
- DC's must be rebooted at least once and it doesnt hurt to run gpupdate on each DC after changing policies when testing
- Open the eventvwr and look for events with source ActivePasswords on the DC's and/or open the text file c:\windows\wsap\wsap.log . Does it contain an invalid license message or any other indications of what goes on?
- Located in c:\windows\wsap\users must be a .prop file for every user. Its a plain text file that can be viewed in notepad.
- One can also run wsad.exe /refresh on each DC to force recreating all .prop files
- You can check the registry location HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa . The word wsap must be mentioned in one of the lines.
- On a password change an event is logged that is viewable in eventvwr and there is also a log added to the file c:\windows\wsap\wsap.log .
- Did you adjust the default Windows password filter as mentioned in Configuration? If any of the installed password filters does not accept the new password, the change will not be accepted.

Custom Scripting

Option 1: Scheduled scripts

If “Create IDP file” is enabled an Identification Password file (<username>.idp) is created in the folder %windir%\wsap\idp when a password is changed or set. This file contains information about the password change in the format *datetime<tab>username<tab>user principal name<tab>display name<tab>password* . *Note that we can also supply a version of ActivePasswords with all password related scripting features removed.*

ActivePasswords periodically queries your Active Directory for changes. If there are any, a Property file (<username>.prop) is created in %windir%\wsap\users . This file contains the values of important Active Directory attributes of the user. The file %windir%\wsap\timestamp contains the date and time of the last completed AD query. The file %windir%\wsap\status contains a list of all users, the date and time of the last change and a checksum for each user which is calculated from attributes that have a non-empty value and that do not contain a date/time/sequential number. This checksum changes when attributes like the name or group membership of the user is changed and is useful to detect ‘updates of real interest’.

All files are utf-8 encoded and tab/crlf separated.

If an Active Directory account has a photo set (thumbnailPhoto), it will be exported to the folder %windir%\wsap\users\photos\<username>.jpg .

These files can optionally be used by custom scripts to setup users for external services like Google Apps or Microsoft Office 365. Place your script in %windir%\wsap\scripts . This folder contains 2 example scripts to sync your AD users to Microsoft Office 365 and/or Google Apps. If a script completes successfully it must output a line [[SUCCESS]]. A script will not run/is disabled if it is prefixed with a dash; ‘- ‘. Script output is logged to a file with the name [script filename.datetime.log] in the log sub folder. If a script did not complete, it is prefixed with an exclamation mark; ‘!’

A script enumerates through all .idp and/or .prop files and processes these. It records the datetime and/or checksum of each successfully processed user. Next time it runs, it can compare the datetime and/or checksum of each file and skip the users that are unchanged.

Option 2: Run program after a password is accepted

Another option is to enable the setting “Run this program after successful change”. You must enter the full path to either an executable or a script file (.exe/.cmd/.bat/.vbs/.js). A PowerShell script (.ps1) is the notable exception because it does not have a default file hander associated. In that case or in general, you can also use a ‘boot batch file’ with a command like `powershell.exe -ExecutionPolicy ByPass -NoLogo -NonInteractive -NoProfile -WindowStyle Hidden -File "myscript.ps1" %1 %2` to start the PowerShell script. The specified program will be called with the username and password as double quoted parameters after a password is accepted.

Example setting: c:\scripts\mysync.cmd . In this case the mysync.cmd will be called with username and password as parameters. The working directory will be the folder where the script is located (c:\scrips). The contents of a simple batch file for testing purposes can be:

```
echo %date% %time% >> output.txt
echo Script full file path: %~f0 >> output.txt
echo Script parameters AS-IS: %* >> output.txt
echo Script directory: %~dp0 >> output.txt
echo Working directory: %cd%\ >> output.txt
echo Parameter 1: %1 Parameter 2: %2 >> output.txt
```

Import into Active Directory

ActivePasswords can also import new and changed users into Active Directory from text files (UTF-8 encoded). The expected formats of these files are the same as the files that ActivePasswords creates. To add a new user or set a password create an IDP file in %windir%\wsap\import\idp . To set other user properties create a property file in %windir%\wsap\import\users . If you add a file <username>.jpg in %windir%\wsap\import\users\photos, it will be uploaded into the user thumbnailPhoto AD attribute.

Logging

ActivePasswords keeps a log file in *%windir%\wsap\wsap.log* . This log file contains information about password change events in the format *datetime<tab>username<tab>user principal name<tab>display name* .

ActivePasswords also creates events in the Windows Application event log. The following events are logged with source "ActivePasswords":

Event ID	Level	Text
1	Informational	Password changed for user <username>
2	Informational	The requested password change for user <username> was not accepted because it did not meet these requirements: <details>
11	Error	Invalid or expired ActivePasswords license on server <computer>

Uninstalling ActivePasswords

Simple run *appwiz.cpl* ("Uninstall or change a program"), select ActivePasswords and click on Uninstall. A reboot is necessary.

Format User Property File

The user property file (.prop) contains important user properties/attributes that can be also used for your own scripting purposes. It consists of tab delimited name value pairs. Attributes prepended with an underscore (_) are calculated/virtual attributes. Date/time notation is in international date format (yyyy-mm-dd hh:mm:ss).

whenChanged	2017-01-09 11:04:31
sAMAccountName	mustrum
userPrincipalName	mridcully@unseenuniversity.edu.am
userAccountControl	512
distinguishedName	CN=Mustrum Ridcully,OU=Users,OU=Accounts,DC=ntdom,DC=local
objectGUID	54D049D8-EC49-4A2A-BD6E-8F190DDC087D
objectSID	S-1-5-21-1314291356-3349140741-1111827591-1346
adsPath	LDAP://ntdom.local/CN=Mustrum Ridcully,OU=Users,OU=Accounts,DC=ntdom,DC=local
whenCreated	2007-04-02 10:30:01
accountExpires	
lastLogonTimestamp	2017-01-09 11:04:32
msDS-ResultantPSO	
pwdLastSet	2016-10-20 08:36:09
cn	Mustrum Ridcully
mail	mridcully@unseenuniversity.edu.am
displayName	Mustrum Ridcully
sn	Ridcully
givenName	Mustrum
title	Archchancellor

description	
info	info 1;info 2
company	Unseen University
department	Top Position
mobile	
telephoneNumber	
homePhone	888
c	NL
co	Nederigland
l	Ankh-Meurbork
st	NB
postalCode	1234AB
postOfficeBox	
physicalDeliveryOfficeName	
facsimileTelephoneNumber	
initials	
streetAddress	Laan 13
wWWHomePage	https://wiki.lspace.org/mediawiki/Unseen_University
manager	
ipPhone	
pager	
homeDirectory	
homeDrive	
profilePath	
scriptPath	start.cmd
middleName	
employeeID	
url	
otherMailbox	
comment	
usnChanged	7791953
_passwordExpired	False
_passwordNeverExpires	False
_accountExpired	False
_accountDisabled	False
_maxPasswordAgeDays	200
_passwordAgeDays	88
_hasPhoto	True
_ou	OU=Users,OU=Accounts,DC=ntdom,DC=local
_groups	Gebruikers;RG;UG;Managers;VPN;Domeingebruikers
_checksum	187F4631