ADMan

Jonathon Reinhart

Jun 25, 2022

CONTENTS

ADMan is a tool for performing various *automated tasks* against an Active Directory domain.

CHAPTER

ONE

INSTALLATION

1.1 Requirements

The following Python packages are required:

- setuptools – For installation
- pip – For installation using pip (recommended)
- python-ldap – LDAP client
- dnspython – DNS client
- PyYAML – YAML parser
- pysmbc – SMB client
  - Only required if `userdirs` configuration is present and `user mkdirs` or `allmaint` command is run

Also, your system must have the GSSAPI module for SASL authentication.

Where possible, it is preferable to install Python packages using your Linux distribution's package manager, rather than from PyPI (using pip). This helps avoid package conflicts.

1.1.1 Debian

To install prerequisites on Debian:

```
apt install \
    python3-setuptools \
    python3-pip \
    python3-ldap \
    python3-dnspython \
    python3-smbc \
    python3-yaml \
    libsasl2-modules-gssapi-mit
```

1.1.2 Fedora

To install prerequisites on Fedora:

```
dnf install \
    python3-setuptools \
    python3-pip \
    python3-ldap \
    python3-dns \
    python3-smbc \
    python3-pyyaml \
    cyrus-sasl-gssapi
```

1.2 Installation

Then install Adman, either using pip:

```
pip3 install adman
```

or from source:

```
tar xf adman-*.tar.gz
cd adman-*
python3 setup.py install
```

CHAPTER

TWO

SETUP

2.1 Account setup

ADMan requires a privileged domain account (because it does privileged things in the domain). This account can be named anything, but here we use `domain-janitor`.

2.1.1 Samba

On a Samba 4 Active Directory domain:

Create the `domain-janitor` user and set its password to not expire:

```
samba-tool user create domain-janitor --random-password
samba-tool user setexpiry --noexpiry domain-janitor
```

Add the user to `Domain Admins`:

```
samba-tool group addmembers 'Domain Admins' domain-janitor
```

Export the user's Kerberos keytab:

```
samba-tool domain exportkeytab --principal='domain-janitor' domain-janitor.keytab
```

2.2 Configuration

First, we'll create a minimal *config file* to get up and going.

- Create `adman/config.yml` in the *appropriate path*:

```
domain: ad.example.com

ldap_auth:
  mode: gssapi
  krb_username: domain-janitor
  krb_keytab: domain-janitor.keytab
  krb_cache: /tmp/domain-janitor.cc
```

- Copy the exported keytab to the path specified in `config.yml`. (The above example specifies `domain-janitor.keytab` in the same directory).

Warning: `domain-janitor.keytab` is password-equivalent; ensure it is carefully protected!

2.3 First run

To test LDAP connectivity and authentication, run the *user list* command:

```
adman user list
```

Before `uidNumber`/`gidNumber` values can be assigned, the next-id state (stored in LDAP) must be initialized using the *state init* command:

```
adman state init
```

2.4 Run automatically

To perform all automated maintenance (assign IDs, UPNs) every minute, run `crontab -e` and add this line (changing the path to `adman` if necessary) to run the *allmaint* command:

```
*/1 * * * *         /usr/local/bin/adman allmaint
```

Note: `adman` will likely be installed in a path not normally searched by `cron`, so we use the full path (revealed by `which adman`).

Note: The *allmaint* command does *not* include *findstale*, as that will usually be done on a much longer interval. Add another cronjob (e.g. weekly) for *findstale*.

CHAPTER

THREE

TASKS

3.1 ID number assignment

Its initial purpose, ADMan can assign **RFC 2307** LDAP `uidNumber/gidNumber` attributes for users, computers, and groups.

3.1.1 State

Adman assigns UID/GID numbers sequentially from a user-defined range, and stores the next-highest value in the `msSFU30MaxUidNumber/msSFU30MaxGidNumber` attributes in LDAP. This ensures that even if users/groups are removed, UID/GID values will not be re-used.

These state variables are referred to by ADman as "next uidNumber" and "next gidNumber".

3.1.2 Actions

For all configured **groups**, ADMan will:

- Assign `gidNumber` values
  - The next `gidNumber` to be assigned is stored in `msSFU30MaxGidNumber`.

For all configured **users and computers**, ADMan will:

- Assign `uidNumber` values
  - The next `uidNumber` to be assigned is stored in `msSFU30MaxUidNumber`.
- Update the `gidNumber` to match that of the user's primary group (`primaryGroupID`)

3.1.3 Configuration

The following configuration options (keys) exist under `id_assign`:

Config Key	Type	Default	Description
`uid_range`	*range*	*(required)*	The range of values to use for assigning `uidNumber` attributes
`gid_range`	*range*	*(required)*	The range of values to use for assigning `gidNumber` attributes
`computers`	*bool*	True	Whether or not to assign `uidNumber` to computer accounts
`only`	*containers*	`'all'`	LDAP containers for which members will be assigned IDs

Example configuration

```
id_assign:
  # Range of values to use for assigning uidNumber attributes
  uid_range:
    min: 100000
    max: 200000

  # Range of values to use for assigning gidNumber attributes
  gid_range:
    min: 100000
    max: 200000

  # Assign uidNumber to computer accounts? (default True)
  computers: True

  # The "only" key, if present, will restrict ID assignment to members of the
  # given containers. Optional scope can be be 'one' or 'subtree' (default).
  # This applies to both users (including computers) and groups.
  only:
    # Recommended to always include these three containers
    CN=Users:
    CN=Computers:
    OU=Domain Controllers:

    # Other custom containers
    OU=ADTest People:
      scope: one
```

3.1.4 Commands

Relevant CLI commands:

- *assignids*
- *computer assign*
- *group assign*
- *user assign*

3.2 UPN suffix consistency

It is recommended that an AD domain be a subdomain of an organization's top-level DNS domain name (e.g., `ad.contoso.com`). It is also recommended that each user's *user principal name* (UPN) match their email address (e.g., `jsmith@contoso.com`).

Together, these recommendations lead to the need to add a secondary UPN suffix: one for the top-level domain. ADMan can ensure that users' UPNs are consistently set.

References:

- Microsoft Docs: User Naming Attributes: userPrincipalName
- Samba Wiki: Active Directory Naming FAQ

3.2.1 Actions

For each configured container, ADMan will enumerate the users and change their `userPrincipalName`, if necessary, to match the desired UPN suffix.

3.2.2 Configuration

`upn_suffixes` is a *mapping* (dictionary) similar to the *containers* type, where the *key* is the the container holding the users to which the UPN suffix will be applied. The *value* is either 1. the UPN suffix to apply, or 2. a mapping with the following keys:

Config Key	Type	Default	Description
`suffix`	*string*	*(required)*	The UPN suffix to apply
`scope`	*string*	`subtree`	Scope of LDAP search in the container: either `one` or `subtree`

Example configuration

```
upn_suffixes:

  # The key is the container which specifies the set of users to which the UPN
  # suffix will be applied. There are two ways to specify the UPN suffix to be
  # applied to a container:

  # 1. The simple format just specifies the suffix:
  CN=Users: example.com

  # 2. The complex format allows the scope to be specified,
  # which can be either 'one' or 'subtree' (the default)
  OU=Special Users,OU=People:
    suffix: special.com
    scope: one
```

3.2.3 Commands

Relevant CLI commands:

- *user setupns*

3.3 User directory creation

ADMan can create per-user directories in any number of base directories on remote SMB servers. It will configure permissions and can even create a default set of subdirectories.

3.3.1 Actions

For each configured *userdirs* entry, ADMan will:

- Create a directory under `basepath` for each user, as limited by the `only` specification
- Set the owner a according to `owner` and `group`
- Add templated access control entries to the ACL according to `acl`
- Create a set of subdirectories under the user directory, and set the owner, group, and ACL

3.3.2 Configuration

`userdirs` is a *list* of entries, each (starting with a hyphen) with the following sub-keys:

Config Key	Type	Default	Description
`basepath`	*string*	*(required)*	UNC path of the directory in which to create each userdir
`only`	*containers*	`'all'`	LDAP containers for which userdir creation will be limited
`owner`	*string*	*None*	The user account name to set as the owner of each userdir. If None, left as-is after creation. Can include template variables: `${user}`
`group`	*string*	*None*	The group account name to set as the group of each userdir If None, left as-is after creation. Can include template variables: `${user}`
`acl`	list<*tmpl_ace*>	A list of access control entries.	*(empty)* | Can include template variables: `${user}`
`subdirs`	list<*subdir*>	*(empty)*	A list of additional subdirectories to create in each user's directory

subdirs – A *mapping* with the following sub-keys:

Config Key	Type	Default	Description
`name`	*string*	*(required)*	Name of the directory to create (in each user directory)
`acl`	list<*tmpl_ace*>	*(empty)*	A list of access control entries. Can include template variables: `${user}`

tmpl_ace – A *string*, representing an Access Control List Entry (ACE), with the following format:

```
sid_or_name:type/flags/mask
```

- `sid_or_name` – Either a SID (e.g. `S-1-5-21-1004336348-1177238915-682003330-512`) or principal name (e.g. `Domain Users`), to which the ACE applies. Can include template variables: `${user}`
- `type` – The type of ACE (see ACE_HEADER)
  - `0` – Access Allowed (typical ACE usage)
  - `1` – Access Denied
- `flags` – A decimal integer of ACE flags which can be ORed together (see ACE_HEADER)
  - `0x01` – `OBJECT_INHERIT`
  - `0x02` – `CONTAINER_INHERIT`
  - `0x04` – `NO_PROPAGATE_INHERIT`
  - `0x08` – `INHERIT_ONLY`

  Example: `11` (`OBJECT_INHERIT | CONTAINER_INHERIT | INHERIT_ONLY`)
- `mask` – A hexadecimal integer of access flags which can be ORed together (see ACCESS_MASK)
  - `0x00000001` – File: Read data / Directory: List
  - `0x00000002` – File: Write data / Directory: Add file
  - `0x00000004` – File: Append data / Directory: Add subdirectory
  - `0x00000008` – File/Directory: Read extended attributes
  - `0x00000010` – File/Directory: Write extended attributes
  - `0x00000020` – File: Execute / Directory: Traverse
  - `0x00000040` – Directory: Delete child
  - `0x00000080` – File/Directory: Read attributes
  - `0x00000100` – File/Directory: Write attributes
  - `0x00010000` – Delete an object
  - `0x00020000` – Read the security descriptor of an object
  - `0x00040000` – Change the access control list of an object
  - `0x00080000` – Change the owner of an object
  - `0x00100000` – Synchronize or wait on the object

Common combinations:

- 0x00120089 – SEC_RIGHTS_FILE_READ / SEC_RIGHTS_DIR_READ
- 0x001200a0 – SEC_RIGHTS_FILE_EXECUTE / SEC_RIGHTS_DIR_EXECUTE
- 0x00120116 – SEC_RIGHTS_FILE_WRITE / SEC_RIGHTS_DIR_WRITE
- 0x001f01ff – SEC_RIGHTS_FILE_ALL / SEC_RIGHTS_DIR_ALL

Note: In the future, we hope to support SDDL for expressing ACLs. See issue 18.

Example configuration

```
userdirs:
    # basepath is the directory in which to create each userdir
  - basepath: '//dc1.ad-test.vx/netlogon/users/'
    # Limit to these users
    only:
      OU=ADTest People:
        scope: one
    # owner is the account name to set as the owner of each userdir
    owner: Fileshare Owner
    group: Storage Admins
    acl:
      - "${user}:0/0/0x001201ff"    # Basically everything but delete
      - "${user}:0/11/0x001f01ff"   # Everything (inherit only)
      - "Domain Users:0/0/0x001200a9"   # Users can... traverse? (Requires access-based␣
↪enumeration)
    # additional subdirectories to create in each user's directory
    # owner and group are inherited from above
    subdirs:
      - name: 'public'
        acl:
          - "${user}:0/0/0x001f01ff"        # Everything
          - "Domain Users:0/0/0x001200a9"   # TODO
```

3.3.3 Commands

Relevant CLI commands:

- *user mkdirs*

3.4 Password expiry notification

ADMan can notify users via email when their password is about to expire in AD. The notification threshold and interval are configurable, along with the templated message to be sent.

3.4.1 Actions

For all users, whose password is not marked as "never expires" (in `userAccountControl`), and whose password has ever been set, ADMan will send an email when their password is about to expire in a given number of days.

Note: This requires the user's `mail` attribute to be set.

3.4.2 Configuration

The following configuration options (keys) exist under `password_expiry_notification`:

Config Key	Type	Default	Description
`days`	*int* or *list<int>*	*(required)*	A list of the number of days before a user's password expires that they should be notified
`template_file`	*path*	*(required)*	Path to template message to send via email

The template file uses Python template strings to provide expansion of the following variables:

Variable	Description
`${cn}`	User common name (e.g., `jsmith`)
`${upn}`	User Principal Name (e.g., `jsmith@example.com`)
`${expire_days}`	The number of days before the user's password expires (with the word "days")
`${expire_time}`	The date/time when the user's password will expire

Example configuration

```
password_expiry_notification:
  # Users should be notified each time their password expires
  # in this many days
  days: [7, 3, 2, 1, 0]


  # The template to use for sending mail
  template_file: example_pwnotify.tmpl
```

Example template

```
Hi ${cn},

Your Active Directory password for ${upn} will expire in ${expire_days}
at ${expire_time}.

Please change your password before this time:
- Windows: Ctrl+Alt+Delete, "Change a Password"
- Linux: "kpasswd"
- Browser: https://passwd.ad-test.vx

Thank you,
Sysadmin
```

3.4.3 Commands

Relevant CLI commands:

- *user checkexpire*

3.5 Find stale user / computer accounts

ADMan can find stale user and computer accounts in AD, send an email to an admin, and optionally disable the accounts. The definition of "stale" is configurable.

Note: The lastLogonTimestamp LDAP attribute used to determine the staleness of an account is only updated every msDS-LogonTimeSyncInterval days, which defaults to 14. Therefore, the granularity cannot typically be set lower than this.

3.5.1 Actions

For each configured user and computer LDAP container (or all, if none configured), ADMan will:

- Find all "stale" accounts, as configured either domain-wide or for that container
- Disable any stale accounts, if configured either domain-wide or for that container

Then ADMan will:

- Send an email to the admin (if `email_to` is set) with the findings and results, in tabular format.

3.5.2 Configuration

The following configuration options (keys) exist under `stale_accounts`:

Config Key	Type	Default	Description
`email_to`	*string*	*(none)*	Email address to which reports are sent
`older_than`	*duration*	*(none)*	Minimum age of a "stale" account (domain default)
`disable`	*boolean*	False	Whether or not to disable stale accounts (domain default)
`include_disabled`	*boolean*	False	Whether or not to include disabled accounts in report (domain default)
`users`	*stale_accts*	`'all'`	User containers to be searched, along with settings which override domain defaults
`computers`	*stale_accts*	`'all'`	Computer containers to be searched, along with settings which override domain defaults

Note: By default, the entire domain is searched for stale user and computer accounts. That can be overridden for user and computer accounts separately. LDAP containers to be searched can be specified here, along with settings which override those above.

stale_accts – Like *containers* but with additional keys, which override the domain defaults above:

Config Key	Type	Default	Description
`older_than`	*duration*	*(domain default)*	Minimum age of a "stale" account (override)
`disable`	*boolean*	*(domain default)*	Whether or not to disable stale accounts (override)
`include_disabled`	*boolean*	*(domain default)*	Whether or not to include disabled accounts in report (domain default)

Example configuration

```
stale_accounts:
  # Admin email to which reports are sent
  email_to: "System Administrator <sysadmin@example.com>"

  # Domain-wide settings
  # How old an account must be before it is "stale"
  older_than: "120 days"

  # Whether or not to disable stale accounts
  #disable: True

  # Whether or not to report already-disabled accounts (default: False)
  include_disabled: True

  # By default, the entire domain is searched for stale user and computer
  # accounts. That can be overridden for user and computer accounts separately.
```

(continues on next page)

(continued from previous page)

```
  # LDAP containers to be searched can be specified here, along with settings
  # which override those above.
  users:
    OU=Special Users,OU=People:
      # LDAP search scope; can be 'one' or 'subtree' (default).
      scope: one
      older_than: "30 days"
      disable: True
      include_disabled: False

    CN=Users:
      #scope: subtree

  computers:
    CN=Computers:
      disable: True
    OU=Domain Controllers:
```

3.5.3 Commands

Relevant CLI commands:

- *findstale*

ADMan supports the following tasks:

- *ID number assignment*: Assign **RFC 2307** `uidNumber/gidNumber` attributes for users, computers and groups
- *UPN suffix consistency*: Ensure UPN suffixes are consistent within an OU or across the domain (e.g., ensure every user's UPN matches his/her email address)
- *User directory creation*: Create per-user (e.g. "home") directories on multiple file servers
- *Password expiry notification*: Email users about expiring passwords
- *Find stale user / computer accounts*: Find and disable stale accounts, and email a report to an admin

CHAPTER

FOUR

CONFIGURATION

ADMan configuration is done using a YAML file.

4.1 Default path

By default, Adman looks for its config file at:

- `/etc/adman/config.yml` when run as `root`
- `~/.config/adman/config.yml` when run as a normal user

4.2 Types

ADMan configuration items expect inputs of certain types. The following standard YAML types are used:

- *boolean* – `True` or `False`
- *int* – A decimal integer
- *string* – A string of text; recommended to be elcosed in quotes
- *list<T>* – A YAML list of values of another type

Additionally, the following custom "types" are defined:

path – A YAML string; the path to a file. The path can either be absolute (start with a leading `/`), or relative to the directory containing the `config.yml` file.

range – A YAML mapping with required `min` and `max` integers

duration – A string describing duration of time, expressed as `N UNITS` where `N` is an integer and `UNITS` is a unit of time e.g. "days"

containers – A YAML mapping where:

- The keys identify an LDAP container by its DN, relative to the domain DN. Put another way, they are one or more RDN strings joined by commas. See the example below.
- The values are a mapping with the following keys:

Config Key	Type	Default	Description
`scope`	*string*	"subtree"	Scope of LDAP search in the container: – `one` (this container only) – `subtree` (this container and all child containers)

- If there are no keys (i.e., YAML `null` or `{}`), then no containers are considered.
- Some uses of the *containers* type may accept the string `all` to mean "all containers in the domain", and may default to this if the entire config setting is omitted.

4.3 Common settings

These settings apply to ADMan as a whole, or multiple *Tasks*. Feature-specific configuration are described on each feature's page.

Config Key	Type	Default	Description
`domain`	*string*	*(required)*	DNS name of domain (e.g. `ad.example.com`)
`changelog`	*path*	*(standard error)*	Path to file where changes are written
`ldap_auth`	*ldap_auth settings*	*(required)*	LDAP authentication options
`smtp`	*smtp settings*	(required[1])	Settings for sending email

Note:

4.3.1 `ldap_auth` settings

Config Key	Type	Default	Description
`mode`	*string*	*(required)*	LDAP authentication mode – choices: • `gssapi` – Use Kerberos via GSSAPI
`krb_username`	*string*	(required[2])	Kerberos username
`krb_keytab`	*path*	(required[2])	Kerberos keytab path (see *Setup*)
`krb_cache`	*path*	(required[2])	Kerberos credential cache path

Note:

[1] `smtp` is required if any of the following are used:

- *Password expiry notification*
- *Find stale user / computer accounts* `stale_accounts.email_to` config

[2] `ldap_auth.krb_*` options are required if ADMan is to automatically manage kerberos tickets. These can be left unset if adman is to use the current user's ticket.

4.3.2 `smtp` settings

Config Key	Type	Default	Description
`email_from`	*string*	*(required)*	Email address from which messages should be sent
`host`	*string*	"localhost"	SMTP server hostname/IP to which messages should be sent
`port`	*integer*	25 (or 465 for SSL)	SMTP server port number
`username`	*string*	*(none)*	SMTP username
`password`	*string*	*(none)* *(req'd w/ username)*	SMTP password
`encryption`	*string*	*(none)*	SMTP server encryption node; one of: • (blank) – Not encrypted • `starttls` – Opportunistic TLS (via `STARTTLS`) • `ssl` – Implicit (mandatory) SSL/TLS

4.4 Example config file

```
# Any path entries can be given as either absolute paths
# or as relative paths, relative to the config file directory.

# The DNS name of the domain
domain: ad.example.com

# LDAP authentication
ldap_auth:
  # Mode of authentication; options: gssapi
  mode: gssapi

  # gssapi options
  # These are required if adman is to automatically manage kerberos tickets.
  # These can be left unset if adman is to use the current user's ticket.

  # Kerberos username
  krb_username: domain-janitor

  # Kerberos keytab path
  krb_keytab: domain-janitor.keytab

  # Kerberos credential cache path
  krb_cache: /tmp/domain-janitor.cc


# Path to file to which changes are logged
# Default: write to stderr
```

(continues on next page)

(continued from previous page)

```
changelog: /var/log/adman-changes.log


# Assign RFC2307 uidNumber/gidNumber attributes to users and groups
id_assign:
  # Range of values to use for assigning uidNumber attributes
  uid_range:
    min: 100000
    max: 200000

  # Range of values to use for assigning gidNumber attributes
  gid_range:
    min: 100000
    max: 200000

  # Assign uidNumber to computer accounts? (default True)
  computers: True

  # The "only" key, if present, will restrict ID assignment to members of the
  # given containers. Optional scope can be be 'one' or 'subtree' (default).
  # This applies to both users (including computers) and groups.
  only:
    # Recommended to always include these three containers
    CN=Users:
    CN=Computers:
    OU=Domain Controllers:

    # Other custom containers
    OU=ADTest People:
      scope: one
# end id_assign

# Automatically create user directories
userdirs:
    # basepath is the directory in which to create each userdir
  - basepath: '//dc1.ad-test.vx/netlogon/users/'
    # Limit to these users
    only:
      OU=ADTest People:
        scope: one
    # owner is the account name to set as the owner of each userdir
    owner: Fileshare Owner
    group: Storage Admins
    acl:
      - "${user}:0/0/0x001201ff"    # Basically everything but delete
      - "${user}:0/11/0x001f01ff"   # Everything (inherit only)
      - "Domain Users:0/0/0x001200a9"   # Users can... traverse? (Requires access-based␣
↪enumeration)
    # additional subdirectories to create in each user's directory
    # owner and group are inherited from above
    subdirs:
      - name: 'public'
```

(continues on next page)

(continued from previous page)

```
        acl:
          - "${user}:0/0/0x001f01ff"        # Everything
          - "Domain Users:0/0/0x001200a9"   # TODO
# end userdirs


# Apply consistent UPN suffixes to all members of a container (OU)
upn_suffixes:

  # The key is the container which specifies the set of users to which the UPN
  # suffix will be applied. There are two ways to specify the UPN suffix to be
  # applied to a container:

  # 1. The simple format just specifies the suffix:
  CN=Users: example.com

  # 2. The complex format allows the scope to be specified,
  # which can be either 'one' or 'subtree' (the default)
  OU=Special Users,OU=People:
    suffix: special.com
    scope: one
# end upn_suffixes


# Notify users when their password is about to expire
# (Useful for LDAP-only users)
password_expiry_notification:
  # Users should be notified each time their password expires
  # in this many days
  days: [7, 3, 2, 1, 0]


  # The template to use for sending mail
  template_file: example_pwnotify.tmpl
# end password_expiry_notification


# Settings used for sending email
smtp:
  # The email address from which messages should be sent (required)
  email_from: "Domain Janitor <domain-janitor@example.com>"

  # Host is optional; defaults to localhost
  host: "smtp.example.com"

  # Port is optional; defaults to 25 (or 465 for SSL)
  port: 25

  # Username/password are optional
  username: "joe"
  password: "password"
```

(continues on next page)

(continued from previous page)

```
  # Encryption is optional and can be "starttls" or "ssl"
  encryption: "starttls"


# Find stale user/computer accounts that haven't recently been logged into
stale_accounts:
  # Admin email to which reports are sent
  email_to: "System Administrator <sysadmin@example.com>"

  # Domain-wide settings
  # How old an account must be before it is "stale"
  older_than: "120 days"

  # Whether or not to disable stale accounts
  #disable: True

  # Whether or not to report already-disabled accounts (default: False)
  include_disabled: True

  # By default, the entire domain is searched for stale user and computer
  # accounts. That can be overridden for user and computer accounts separately.
  # LDAP containers to be searched can be specified here, along with settings
  # which override those above.
  users:
    OU=Special Users,OU=People:
      # LDAP search scope; can be 'one' or 'subtree' (default).
      scope: one
      older_than: "30 days"
      disable: True
      include_disabled: False

    CN=Users:
      #scope: subtree

  computers:
    CN=Computers:
      disable: True
    OU=Domain Controllers:
# end stale_accounts
```

CHAPTER

FIVE

COMMAND-LINE INTERFACE

5.1 Synopsis

```
admin [-h] [-c CONFIG] [-v] [--version]
      [--loglevel LEVEL]
      COMMAND ...
```

Global Options:

-h, --help	Show help message and exit
--version	Show ADMan version and exit
--loglevel LEVEL	Set the logging level (default: WARNING)
	Options: DEBUG,INFO,WARNING,ERROR,CRITICAL
-c CONFIG, --config CONFIG	Alternate path to config file
-v, --verbose	Show verbose output

5.2 Commands

Command	Description
	Top-level commands
allmaint	Perform all automated maintenance (assign IDs, UPNs)
assignids	Assign all missing `*idNumber` attributes
clearids	Clear all `*idNumber` attributes
exec	Execute a command in Kerberos context
findstale	Find stale accounts and report/disable per config
computer	**Computer sub-commands**
computer assign	Assign missing `uidNumber` attributes
computer list	List computers
group	**Group sub-commands**
group assign	Assign missing `gidNumber` attributes
group list	List groups
state	**State sub-commands**
state list	List state information
state init	Initialize state information
user	**User sub-commands**
user assign	Assign missing `uidNumber` attributes
user checkexpire	Check for expiring/expired passwords
user setupns	Set `userPrincipalName` attributes
user list	List users
user mkdirs	Make user directories

5.2.1 Top-level commands

`allmaint`

Shortcut command which runs all* automated maintenance commands:

- *assignids*
- *user setupns*
- *user checkexpire*
- *user mkdirs*

Note: The *allmaint* command does *not* include *findstale*, as that will usually be done on a much longer interval.

assignids

Shortcut command which runs the following *ID number assignment* commands:

- *group assign*
- *user assign*
- *computer assign* (if configured)

clearids

This command will clear all `*idNumber` attributes for the configured:

- Group `gidNumber`
- User `uidNumber` & `gidNumber`
- Computer `uidNumber` & `gidNumber`

exec

(Added in v0.6.0) This command enables running arbitrary command lines in the ADMan Kerberos context. This is useful for `samba-tool` commands which support Kerberos.

Example:

```
$ adman exec samba-tool domain backup online --server=dc1.example.com --
↪targetdir=domainbakup -k yes
```

findstale

(Added in v0.7.0)

This command will find stale user/computer accounts and disable them as configured. If configured, it will send a report to the admin.

See *Find stale user / computer accounts*.

5.2.2 Computer commands

computer assign

See *user assign*.

computer list

List all computers.

5.2.3 Group commands

group assign

This *ID number assignment* command will:

- Assign `gidNumber` values to all configured groups.
  - The next `gidNumber` to be assigned is stored in `msSFU30MaxGidNumber`.

group list

List all groups.

5.2.4 State commands

These commands interact with the *ADMan-related state* recorded in LDAP.

state list

List the current state:

```
$ adman state list
Next uidNumber: 100011
Next gidNumber: 100008
```

state init

Initialize the ADMan state.

```
adman state init [-h] [--force | --ignore]
```

Options:

-h, --help	Show help message and exit
--force	Force re-initialization; overwrite existing values with `MAX(xidNumber)+1`
--ignore	Ignore partially-initialized state and initialize other values

This command evaluates the `uidNumber`/`gidNumber` values currently assigned to users/groups, and sets the "Next uidNumber" and "Next gidNumber" values accordingly:

- If no `xidNumber` are currently assigned, sets "next" to the beginning of the configured range.
- Otherwise, sets "next" to `MAX(xidNumber)+1`.

If the state is already initialized and is as expected, nothing is done:

```
$ adman state init
Next uidNumber: 100011
Next gidNumber: 100008
```

If the state is already initialized but doesn't match the expected value, an error is printed:

```
$ adman state init
Next uidNumber: 100011
Next gidNumber: 100008

Error: Domain state next_uid already set to 100011, doesn't match expected 100008
Use --force or --ignore
```

5.2.5 User commands

`user assign`

This *ID number assignment* command will:

- Assign `uidNumber` values to all configured users.
  - The next `uidNumber` to be assigned is stored in `msSFU30MaxUidNumber`.
- Update the `gidNumber` to match that of the user's primary group (`primaryGroupID`).

`user checkexpire`

(Added in v0.2.0)

This command will send an email to users whose passwords are expiring in the configured time window.

See *Password expiry notification*.

`user setupns`

This command will update users' `userPrincipalName` attribute if necessary to match the configured UPN suffix.

See *UPN suffix consistency*.

`user list`

List all users.

`user mkdirs`

This command will create per-user directories as configured.

See *User directory creation*.

CHAPTER

SIX

TROUBLESHOOTING

6.1 No worthy mechs found

```
ldap.AUTH_UNKNOWN: {'desc': 'Unknown authentication method', 'errno': 22, 'info': 'SASL(-
↪4): no mechanism available: No worthy mechs found'}
```

You need to install the GSSAPI SASL modules. On Debian:

```
apt install libsasl2-modules-gssapi-mit
```

6.2 Insufficient access

```
ldap.INSUFFICIENT_ACCESS: {'desc': 'Insufficient access', 'info': '00002098: Object␣
↪CN=adtest,CN=ypservers,CN=ypServ30,CN=RpcServices,CN=System,DC=ad-test,DC=vx has no␣
↪write property access\n'}
```

The ADMan user needs to be a member of `Domain Admins`.

Once this change has been made, you must remove the stale credential cache, e.g.:

```
rm /tmp/domain-janitor.cc
```

6.3 Server not found in Kerberos database

```
SASL: GSSAPI Error: Unspecified GSS failure. Minor code may provide more information␣
↪(Server not found in Kerberos database).
```

Various problems can lead to this error. One common case I've encountered is that a reverse DNS (PTR) record does not exist for the DC(s).

CHAPTER

SEVEN

CHANGE LOG

All notable changes to this project will be documented in this file. This project adheres to Semantic Versioning.

7.1 0.9.0 - 2022-06-25

- Allow explicitly empty "container" config items (!36, !38)
- Add `include_disabled` config to `stale_accounts` (!37)
- Update stale account report header (!39)

7.2 0.8.0 - 2021-12-12

- Only connect to DC holding PDC Emulator FSMO role for single-master safety (!26)
- Switch from unittest to pytest-based tests (!29)
- Improve LDAP object model, adding tests and using LDAP REPLACE to update non-atomic attributes (!28)
- Ensure LDAP attributes used to hold next uid/gid values are updated atomically (!30)

7.3 0.7.3 - 2021-12-05

- Write to changelog when users are disabled (!27)

7.4 0.7.2 - 2021-11-22

- Add HTML format to stale account report (!25)

7.5 0.7.1 - 2021-11-06

- Fix missing dnsdomain attribute (!24)

7.6 0.7.0 - 2021-11-06

- Add `findstale` command and `stale_accounts` config which enables finding and disabling stale user or computer accounts, and sending an email to the administrator (!23)
- Allow running without `krb_*` config parameters being set (!18)

7.7 0.6.1 - 2021-10-23

- Fix error parsing klist output due to new "Ticket server" line in krb5 1.18 (!20)

7.8 0.6.0 - 2021-05-21

- Add `exec` command for running arbitrary commands in ADMan kerberos context

7.9 0.5.1 - 2021-02-02

- Run `klist` using the 'C' locale to avoid locale-induced date parsing errors (!16)
- Make pysmbc an optional dependency, only needed fo `user mkdirs` command (!17)

7.10 0.5.0 - 2020-03-17

- Fix unnecessary Kerberos ticket request on every invocation (!11)
- Make `id_assign` config section optional (!12)
- Add `user mkdirs` option to create configured `userdirs` (!13)`
- Exclude disabled users from all assignments (!14)

7.11 0.4.0 - 2020-02-18

- Add new `only` key to `id_assign` config which enables restricting the containers whose users and groups are assigned `uidNumber`/`gidNumber` attributes (!8)
- Add timestamps to changelog output (!9)
- Move changelog path from command line option to config file (!10)

7.12 0.3.0 - 2020-02-16

- Move `uid_range`/`gid_range` configuration options under a new `id_assign` key
- Assign gidNumber to computer groups (e.g. `Domain Computers`) and assign uidNumber to computer objects. This new behavior is on by default but can be disabled via new `id_assign.computers` key. (!7)

7.13 0.2.3 - 2020-02-11

- Change kerberos code to be Python 3.5 compatible (#14)
- Don't require smtp config if email won't be used (#9)

7.14 0.2.2 - 2020-01-17

- Fix issue where a domain with password expiry disabled would lead to a datetime error (#11)

7.15 0.2.1 - 2020-01-06

- Fix issue where an expiring TGT can lead to a GSSAPI error (#10).
- Add -v option for 'user list' and 'group list'

7.16 0.2.0 - 2019-12-26

- Add pending password expiry notification feature
  - Added 'user checkexpire' command and related configuration
  - Added SMTP configuration
- Set default config and data paths
- Add –version option
- Fix exception when domain has no configured alternate UPN suffixes
- Refactor LdapObject code to simplify handling of known attributes

7.17 0.1.0 - 2019-07-18

- Add support for UPN suffix assignment via `user setupns` command
- Rename `assign`/`clear` to `assignids`/`clearids`
- Add `allmaint` command

7.18 0.0.2 - 2019-07-16

- Rename to “adman”

7.19 0.0.1 - 2019-06-23

- Initial release as “adam”

Adman can run on any Linux system; the host system does not even need to be joined to the domain. Adman typically runs with a *dedicated user* (e.g. `domain-janitor`) and uses a Kerberos keytab, rather than password-based authentication.

INDEX

R