



redcanaryco / atomic-red-team Public

Notifications Fork 2.8k Star 9.7k

<> Code Issues 6 Pull requests 4 Actions Wiki Security Insights

atomic-red-team / atomics / T1087.001 / T1087.001.md

CircleCI Atomic Red Team doc... Generate docs from job=generate\_and\_commit\_gui... 7091fa8 · 2 years ago

391 lines (189 loc) · 6.83 KB

Preview Code Blame

Raw Copy Download

# T1087.001 - Local Account

## Description from ATT&CK

Adversaries may attempt to get a listing of local system accounts. This information can help adversaries determine which local accounts exist on a system to aid in follow-on behavior. Commands such as `net user` and `net localgroup` of the [Net](#) utility and `id` and `groups` on macOS and Linux can list local users and groups. On Linux, local users can also be enumerated through the use of the `/etc/passwd` file. On macOS the `dscl . list /Users` command can be used to enumerate local accounts.

## Atomic Tests

- [Atomic Test #1 - Enumerate all accounts \(Local\)](#)
- [Atomic Test #2 - View sudoers access](#)
- [Atomic Test #3 - View accounts with UID 0](#)
- [Atomic Test #4 - List opened files by user](#)

Enumerate all accounts by copying `/etc/passwd` to another file

auto\_generated\_guid: f8aab3dd-5990-4bf8-b8ab-2226c951696f

**Inputs:**

Name	Description	Type	Default Value
output_file	Path where captured results will be placed	Path	/tmp/T1087.001.txt

## Attack Commands: Run with `sh`!

```
cat /etc/passwd > #{output_file}
cat #{output_file}
```



## Cleanup Commands:

```
rm -f #{output_file}
```



## Atomic Test #2 - View sudoers access

(requires root)

**Supported Platforms:** Linux, macOS

**auto\_generated\_guid:** fed9be70-0186-4bde-9f8a-20945f9370c2

**Inputs:**

Name	Description	Type	Default Value
output_file	Path where captured results will be placed	Path	/tmp/T1087.001.txt

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

```
sudo cat /etc/sudoers > #{output_file}
cat #{output_file}
```

**Cleanup Commands:**

```
rm -f #{output_file}
```

## Atomic Test #3 - View accounts with UID 0

View accounts with UID 0

**Supported Platforms:** Linux, macOS

**auto\_generated\_guid:** c955a599-3653-4fe5-b631-f11c00eb0397

**Inputs:**

Name	Description	Type	Default Value
------	-------------	------	---------------

output_file	Path where captured results will be placed	Path	/tmp/T1087.001.txt
-------------	--	------	--------------------

Attack Commands: Run with `sh` !

```
grep 'x:0:' /etc/passwd > #{output_file}
cat #{output_file} 2>/dev/null
```

Cleanup Commands:

```
rm -f #{output_file} 2>/dev/null
```

# Atomic Test #4 - List opened files by user

List opened files by user

Supported Platforms: Linux, macOS

auto\_generated\_guid: 7e46c7a5-0142-45be-a858-1a3ecb4fd3cb

Attack Commands: Run with `sh` !

```
username=$(id -u -n) && lsof -u $username
```

Dependencies: Run with `sh` !

Description: check if lsof exists

Check Prereq Commands:

```
which lsof
```

Get Prereq Commands:

```
(which yum && yum -y install lsof) || (which apt-get && DEBIAN_FRONTEND=noninteracti'
```

## Atomic Test #5 - Show if a user account has ever logged in remotely

Show if a user account has ever logged in remotely

**Supported Platforms:** Linux

**auto\_generated\_guid:** 0f0b6a29-08c3-44ad-a30b-47fd996b2110

**Inputs:**

Name	Description	Type	Default Value
output_file	Path where captured results will be placed	Path	/tmp/T1087.001.txt

**Attack Commands:** Run with **sh** !

```
lastlog > #{output_file}
cat #{output_file}
```

**Cleanup Commands:**

```
rm -f #{output_file}
```

**Dependencies:** Run with **sh** !

**Description:** Check if lastlog command exists on the machine

**Check Prereq Commands:**

```
if [ -x "$(command -v lastlog)" ]; then exit 0; else exit 1; fi
```

#### Get Prereq Commands:

```
sudo apt-get install login; exit 1;
```



## Atomic Test #6 - Enumerate users and groups

Utilize groups and id to enumerate users and groups

**Supported Platforms:** Linux, macOS

**auto\_generated\_guid:** e6f36545-dc1e-47f0-9f48-7f730f54a02e

**Attack Commands:** Run with **sh** !

```
groups  
id
```



## Atomic Test #7 - Enumerate users and groups

Utilize local utilities to enumerate users and groups

**Supported Platforms:** macOS

**auto\_generated\_guid:** 319e9f6c-7a9e-432e-8c62-9385c803b6f2

**Attack Commands:** Run with **sh** !

```
dscl . list /Groups  
dscl . list /Users  
dscl . list /Users | grep -v '_'  
dscacheutil -q group  
dscacheutil -q user
```



## Atomic Test #8 - Enumerate all accounts on Windows (Local)

Enumerate all accounts Upon execution, multiple enumeration commands will be run and their output displayed in the PowerShell session

**Supported Platforms:** Windows

**auto\_generated\_guid:** 80887bec-5a9b-4efc-a81d-f83eb2eb32ab

**Attack Commands:** Run with `command_prompt` !

```
net user
dir c:\Users\
cmdkey.exe /list
net localgroup "Users"
net localgroup
```



## Atomic Test #9 - Enumerate all accounts via PowerShell (Local)

Enumerate all accounts via PowerShell. Upon execution, lots of user account and group information will be displayed.

**Supported Platforms:** Windows

**auto\_generated\_guid:** ae4b6361-b5f8-46cb-a3f9-9cf108ccfe7b

**Attack Commands:** Run with `powershell` !

```
net user
get-localuser
get-localgroupmember -group Users
cmdkey.exe /list
ls C:/Users
get-childitem C:\Users\
```



```
dir C:\Users\  
get-localgroup  
net localgroup
```

## Atomic Test #10 - Enumerate logged on users via CMD (Local)

Enumerate logged on users. Upon execution, logged on users will be displayed.

**Supported Platforms:** Windows

**auto\_generated\_guid:** a138085e-bfe5-46ba-a242-74a6fb884af3

**Attack Commands:** Run with `command_prompt` !

```
query user
```

