

# T1036.003 - Masquerading: Rename System Utilities

## Description from ATT&CK

Adversaries may rename legitimate system utilities to try to evade security mechanisms concerning the usage of those utilities. Security monitoring and control mechanisms may be in place for system utilities adversaries are capable of abusing. (Citation: LOLBAS Main Site) It may be possible to bypass those security mechanisms by renaming the utility prior to utilization (ex: rename rundl132.exe ). (Citation: Elastic Masquerade Ball) An alternative case occurs when a legitimate utility is copied or moved to a different directory and renamed to avoid detections based on system utilities executing from non-standard paths. (Citation: F-Secure CozyDuke)

## Atomic Tests

- [Atomic Test #1 - Masquerading as Windows LSASS process](#)
- [Atomic Test #2 - Masquerading as Linux crond process.](#)

- [Atomic Test #3 - Masquerading - cscript.exe running as notepad.exe](#)
- [Atomic Test #4 - Masquerading - wscript.exe running as svchost.exe](#)
- [Atomic Test #5 - Masquerading - powershell.exe running as taskhostw.exe](#)
- [Atomic Test #6 - Masquerading - non-windows exe running as windows exe](#)
- [Atomic Test #7 - Masquerading - windows exe running as different windows exe](#)
- [Atomic Test #8 - Malicious process Masquerading as LSM.exe](#)
- [Atomic Test #9 - File Extension Masquerading](#)

Preview

Code

Blame

Raw



## Atomic Test #1 - Masquerading as Windows LSASS process

Copies cmd.exe, renames it, and launches it to masquerade as an instance of lsass.exe.

Upon execution, cmd will be launched by powershell. If using Invoke-AtomicTest, The test will hang until the 120 second timeout cancels the session

**Supported Platforms:** Windows

**auto\_generated\_guid:** 5ba5a3d1-cf3c-4499-968a-a93155d1f717

**Attack Commands:** Run with **command\_prompt** !

```
copy %SystemRoot%\System32\cmd.exe %SystemRoot%\Temp\lsass.exe  
%SystemRoot%\Temp\lsass.exe /B
```



**Cleanup Commands:**

```
del /Q /F %SystemRoot%\Temp\lsass.exe >nul 2>&1
```



## Atomic Test #2 - Masquerading as Linux crond process.

Copies sh process, renames it as crond, and executes it to masquerade as the cron daemon.

Upon successful execution, sh is renamed to `crond` and executed.

**Supported Platforms:** Linux

**auto\_generated\_guid:** a315bfff-7a98-403b-b442-2ea1b255e556

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

```
cp /bin/sh /tmp/crond;  
echo 'sleep 5' | /tmp/crond
```



**Cleanup Commands:**

```
rm /tmp/crond
```



## Atomic Test #3 - Masquerading - cscript.exe running as notepad.exe

Copies cscript.exe, renames it, and launches it to masquerade as an instance of notepad.exe.

Upon successful execution, cscript.exe is renamed as notepad.exe and executed from non-standard path.

**Supported Platforms:** Windows

**auto\_generated\_guid:** 3a2a578b-0a01-46e4-92e3-62e2859b42f0

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

```
copy %SystemRoot%\System32\cscript.exe %APPDATA%\notepad.exe /Y  
cmd.exe /c %APPDATA%\notepad.exe /B
```



### Cleanup Commands:

```
del /Q /F %APPDATA%\notepad.exe >nul 2>&1
```



## Atomic Test #4 - Masquerading - wscript.exe running as svchost.exe

Copies wscript.exe, renames it, and launches it to masquerade as an instance of svchost.exe.

Upon execution, no windows will remain open but wscript will have been renamed to svchost and ran out of the temp folder

**Supported Platforms:** Windows

**auto\_generated\_guid:** 24136435-c91a-4ede-9da1-8b284a1c1a23

**Attack Commands:** Run with **command\_prompt** !

```
copy %SystemRoot%\System32\wscript.exe %APPDATA%\svchost.exe /Y  
cmd.exe /c %APPDATA%\svchost.exe /B
```



### Cleanup Commands:

```
del /Q /F %APPDATA%\svchost.exe >nul 2>&1
```



## Atomic Test #5 - Masquerading - powershell.exe running as taskhostw.exe

Copies powershell.exe, renames it, and launches it to masquerade as an instance of taskhostw.exe.

Upon successful execution, powershell.exe is renamed as taskhostw.exe and executed from non-standard path.

Supported Platforms: Windows

auto\_generated\_guid: ac9d0fc3-8aa8-4ab5-b11f-682cd63b40aa

Attack Commands: Run with **command\_prompt** !

```
copy %windir%\System32\windowspowershell\v1.0\powershell.exe %APPDATA%\taskhostw.exe
cmd.exe /K %APPDATA%\taskhostw.exe
```

Cleanup Commands:

```
del /Q /F %APPDATA%\taskhostw.exe >nul 2>&1
```

## Atomic Test #6 - Masquerading - non-windows exe running as windows exe

Copies an exe, renames it as a windows exe, and launches it to masquerade as a real windows exe

Upon successful execution, powershell will execute T1036.003.exe as svchost.exe from on a non-standard path.

Supported Platforms: Windows

auto\_generated\_guid: bc15c13f-d121-4b1f-8c7d-28d95854d086

Inputs:

Name	Description	Type	Default Value
outputfile	path of file to execute	path	(\$env:TEMP + "\svchost.exe")
inputfile	path of file to copy	path	PathToAtomicsFolder\T1036.003\bin\T1036.003.exe

## Attack Commands: Run with powershell !

```
copy #{inputfile} #{outputfile}
$myT1036_003 = (Start-Process -PassThru -FilePath #{outputfile}).Id
Stop-Process -ID $myT1036_003
```



## Cleanup Commands:

```
Remove-Item #{outputfile} -Force -ErrorAction Ignore
```



## Dependencies: Run with powershell !

Description: Exe file to copy must exist on disk at specified location (#{inputfile})

## Check Prereq Commands:

```
if (Test-Path #{inputfile}) {exit 0} else {exit 1}
```



## Get Prereq Commands:

```
New-Item -Type Directory (split-path #{inputfile}) -ErrorAction ignore | Out-Null
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomic"
```



# Atomic Test #7 - Masquerading - windows exe running as different windows exe

Copies a windows exe, renames it as another windows exe, and launches it to masquerade as second windows exe

Supported Platforms: Windows

auto\_generated\_guid: c3d24a39-2bfe-4c6a-b064-90cd73896cb0

Inputs:

Name	Description	Type	Default Value
outputfile	path of file to execute	path	(\$env:TEMP + "\svchost.exe")
inputfile	path of file to copy	path	\$env:ComSpec

Attack Commands: Run with powershell !

```
copy #{inputfile} #{outputfile}
$myT1036_003 = (Start-Process -PassThru -FilePath #{outputfile}).Id
Stop-Process -ID $myT1036_003
```

Cleanup Commands:

```
Remove-Item #{outputfile} -Force -ErrorAction Ignore
```

## Atomic Test #8 - Malicious process Masquerading as LSM.exe

Detect LSM running from an incorrect directory and an incorrect service account This works by copying cmd.exe to a file, naming it lsm.exe, then copying a file to the C:\ folder.

Upon successful execution, cmd.exe will be renamed as lsm.exe and executed from non-standard path.

Supported Platforms: Windows

auto\_generated\_guid: 83810c46-f45e-4485-9ab6-8ed0e9e6ed7f

Attack Commands: Run with command\_prompt ! Elevation Required (e.g. root or admin)

```
copy C:\Windows\System32\cmd.exe C:\lsm.exe
C:\lsm.exe /c echo T1036.003 > C:\T1036.003.txt
```

Cleanup Commands:

```
del C:\T1036.003.txt >nul 2>&1
del C:\lsm.exe >nul 2>&1
```



## Atomic Test #9 - File Extension Masquerading

download and execute a file masquerading as images or Office files. Upon execution 3 calc instances and 3 vbs windows will be launched.

e.g SOME\_LEGIT\_NAME.[doc,docx,xls,xlsx,pdf,rtf,png,jpg,etc].[exe,vbs,js,ps1,etc] (Quartelyreport.docx.exe)

**Supported Platforms:** Windows

**auto\_generated\_guid:** c7fa0c3b-b57f-4cba-9118-863bf4e653fc

**Inputs:**

Name	Description	Type	Default Value
exe_path	path to exe to use when creating masquerading files	path	C:\Windows\System32\calc.exe
vbs_path	path of vbs to use when creating masquerading files	path	PathToAtomicsFolder\T1036.003\src\T1036.003_masqueradir
ps1_path	path of powershell script to use when creating masquerading files	path	PathToAtomicsFolder\T1036.003\src\T1036.003_masqueradir



## Attack Commands: Run with `command_prompt` !

```
copy #{exe_path} %temp%\T1036.003_masquerading.docx.exe /Y
copy #{exe_path} %temp%\T1036.003_masquerading.pdf.exe /Y
copy #{exe_path} %temp%\T1036.003_masquerading.ps1.exe /Y
copy #{vbs_path} %temp%\T1036.003_masquerading.xls.vbs /Y
copy #{vbs_path} %temp%\T1036.003_masquerading.xlsx.vbs /Y
copy #{vbs_path} %temp%\T1036.003_masquerading.png.vbs /Y
copy #{ps1_path} %temp%\T1036.003_masquerading.doc.ps1 /Y
copy #{ps1_path} %temp%\T1036.003_masquerading.pdf.ps1 /Y
copy #{ps1_path} %temp%\T1036.003_masquerading.rtf.ps1 /Y
%temp%\T1036.003_masquerading.docx.exe
%temp%\T1036.003_masquerading.pdf.exe
%temp%\T1036.003_masquerading.ps1.exe
%temp%\T1036.003_masquerading.xls.vbs
%temp%\T1036.003_masquerading.xlsx.vbs
%temp%\T1036.003_masquerading.png.vbs
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -File %temp%\T1036.003_m
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -File %temp%\T1036.003_m
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -File %temp%\T1036.003_m
```

## Cleanup Commands:

```
del /f %temp%\T1036.003_masquerading.docx.exe > nul 2>&1
del /f %temp%\T1036.003_masquerading.pdf.exe > nul 2>&1
del /f %temp%\T1036.003_masquerading.ps1.exe > nul 2>&1
del /f %temp%\T1036.003_masquerading.xls.vbs > nul 2>&1
del /f %temp%\T1036.003_masquerading.xlsx.vbs > nul 2>&1
del /f %temp%\T1036.003_masquerading.png.vbs > nul 2>&1
del /f %temp%\T1036.003_masquerading.doc.ps1 > nul 2>&1
del /f %temp%\T1036.003_masquerading.pdf.ps1 > nul 2>&1
del /f %temp%\T1036.003_masquerading.rtf.ps1 > nul 2>&1
```

## Dependencies: Run with `powershell` !

Description: File to copy must exist on disk at specified location (#{vbs\_path})

## Check Prereq Commands:

```
if (Test-Path #{vbs_path}) {exit 0} else {exit 1}
```

#### Get Prereq Commands:

```
New-Item -Type Directory (split-path #{vbs_path}) -ErrorAction ignore | Out-Null  
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomic"
```



Description: File to copy must exist on disk at specified location (#{ps1\_path})

#### Check Prereq Commands:

```
if (Test-Path #{ps1_path}) {exit 0} else {exit 1}
```



#### Get Prereq Commands:

```
New-Item -Type Directory (split-path #{ps1_path}) -ErrorAction ignore | Out-Null  
Invoke-WebRequest "https://github.com/redcanaryco/atomic-red-team/raw/master/atomic"
```

