

Abusing Windows 10 Narrator's 'Feedback-Hub' URI for Fileless Persistence

While investigating Ease of Access options in Windows 10 for new persistence techniques, I have actually found an undocumented one via 'Provide Narrator feedback' functionality.

Behind the scenes the Narrator feedback consists in launching the custom handler via URI scheme 'feedback-hub'. However, in a post exploitation scenario is possible to trivially backdoor this component with fileless payloads hosted in the registry.

Even if there is no security boundary between windows logon screen and the default user desktop (indeed both part of the same window station WinSta0) the possibility of the interaction between the Narrator instance running in the environment of the locked out users and the Windows logon screen opens the chance to trigger the malicious command defined in the registry as soon as the 'Provide Narrator feedback' combination keys are pressed in the latter context.

The novel technique presented in this article has the following advantages in respect to already known Ease of Use abuses (see next paragraph):

- fileless (*Living off the Land approach*)
- no administrative privileges required (*if physical access scenario and victim user is locked out*)

Demo video available here:

<https://www.youtube.com/watch?v=oPKnYO9V8-M>

For the insights, have a look at the documentation for Universal App URI schemes persistence:

<https://github.com/giuliocomi/backoori>

Quick recap of Accessibility Features for Red Teamers

The Windows Accessibility Features, a set of tools available in the Windows logon screen (like Sticky Keys), are designed to be launched via predefined combination of keys to assist the end users. These Windows features are also quite famous because have been abused by [APT groups for backdooring target systems](#) in the past. Having administrative privileges is a requirement in order to replace the genuine Windows binary of the tool ('sethc.exe' or 'narrator.exe', 'magnify.exe', etc.) with an ad-hoc binary.

Moreover, Microsoft has started monitoring this category of issues via Windows Defender in September 2018 (<https://www.microsoft.com/en-us/wdsi/threats/threat-search?query=Trojan:Win32/AccessibilityEscalation.A>).

@giulio_comì – <https://github.com/giuliocomi/backoori>

This novel technique I have discovered overcomes the admin privileges condition (provided that it is physical access scenario and the victim account is locked), is fileless and currently not monitored by security detection tools.

Overview of the Universal Apps URI schemes persistence

The Accessibility feature is a specific case of the more comprehensive URI persistence technique that affects all Universal Apps URI, which is applicable to every URI protocol listed in the Settings under “*Choose default apps by protocol*”. Some of these protocols are very interesting, like ‘https’ because in this case it will be possible to trigger the payload from a crafted web pages (with for example an <a> tag link) and the payload will be “MiTM” for the request by executing itself and transparently forwarding the arguments to the legitimate default browser of the unaware victim (for more details have a look at [backoori](#)).

Tweaking Narrator’s settings of the compromised user

But let’s not digress, the focus of this walk-through is on the Narrator feature abuse.

Every time the ‘feedback-hub’ URI is triggered via:

- shortcut key for Feedback Hub in the desktop environment
- the task manager ‘Send feedback’ option
- ‘explorer.exe feedback-hub:’ command
- Narrator Ease of Use feedback in the windows logon desktop

the defined payload will get executed.

For backdooring the last option, the one that involves the Narrator, it is recommended to apply the configuration displayed in the screenshots below. The reason is that the Narrator does not start automatically, it is very loudly and its cursor catches the yes of the victims. Moreover, as said before the Narrator abuse works out of the box for locked out victims (therefore physical persistence), but for signed out users it is necessary to also enable “Start Narrator before sign-in for everyone” with a compromised administrative account.

Use Narrator

Turn on Narrator



On

[Open Narrator Home](#)

[View the complete guide to Narrator online](#)

Start-up options



Allow the shortcut key to start Narrator

Press the Windows logo key  + Ctrl + Enter to turn Narrator on or off.



Start Narrator after sign-in for me



Start Narrator before sign-in for everyone



Show Narrator Home when Narrator starts



Minimize Narrator Home to the system tray

When this box is unchecked, Narrator Home will minimize to the taskbar.

Use Narrator cursor

The Narrator cursor is where Narrator is focused on your screen.



Show the Narrator cursor



Move my cursor with the Narrator cursor as Narrator reads text



Sync the Narrator cursor and system focus



Read and interact with the screen using the mouse


Select the Narrator cursor navigation mode

Normal 

Normal mode is recommended.

Personalize Narrator's voice

Choose a voice

Microsoft David - English (United States) 

[Add more voices](#)

Change voice speed



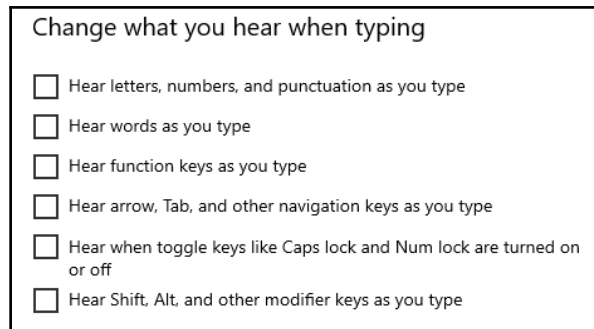
Press the Narrator key + Plus (+) or Narrator + Minus (-) to change voice speed.

Change voice pitch

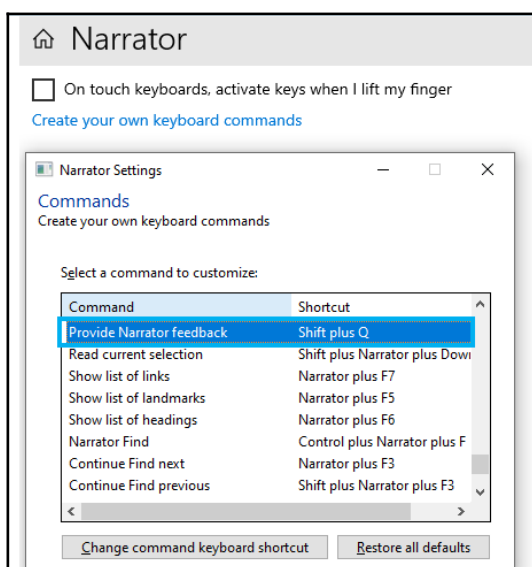


Change voice volume





And choose a shortcut key for the ‘Provide Narrator feedback’ setting.



Backdooring ‘Feedback-Hub’ URI functionality

There are two approaches, the expected way is to develop a Universal App and set it as default handler and the more smoothly one based on the editing of registry keys. Let’s focus on the second one.

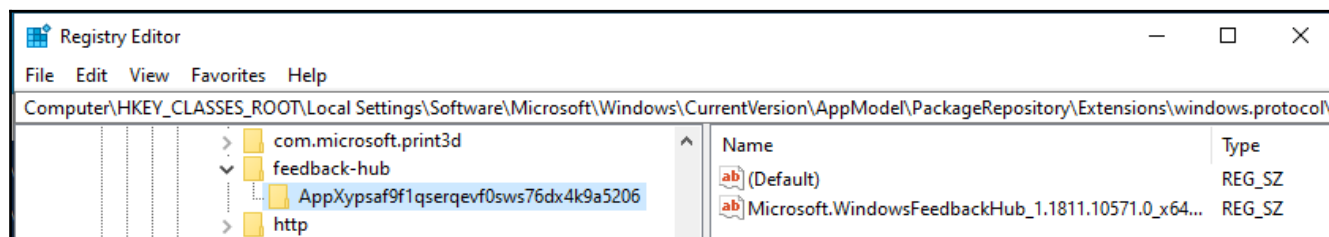
We need to track down the essentials keys to modify in the Registry in order to point the Feedback Hub Microsoft URI scheme to our own payload.

To have a better insights on the few steps involved, have a look at https://github.com/giuliocomi/backoori/blob/master/agent/agent_plate.ps1, the agent template part of the tool created as PoC to automate this persistence technique for arbitrary specified URLs.

By looking up the registry for ‘feedback-hub’ key, we find out one registered Universal App Id:

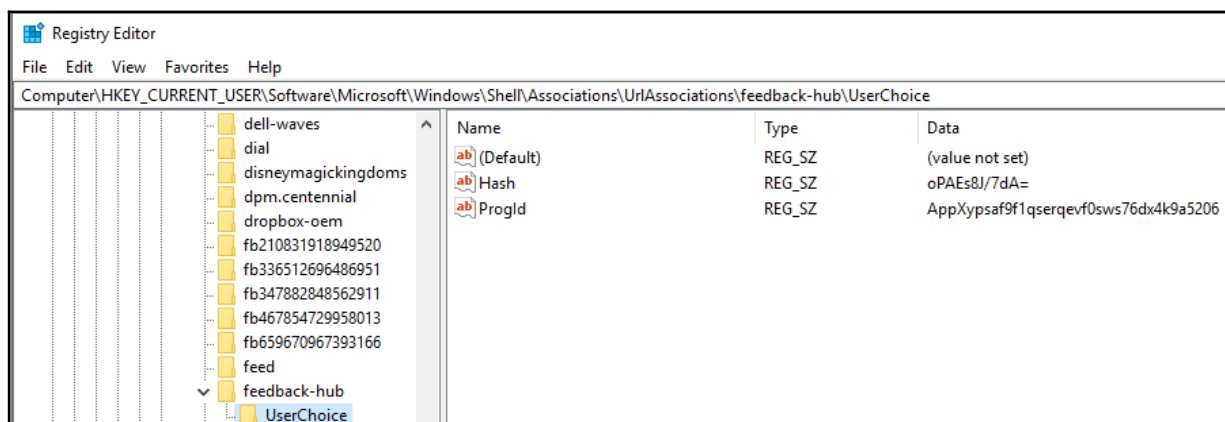
```
HKCR:Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\PackageRepository\Extensions\windows.protocol\feedback-hub
```

@giulio_comì – <https://github.com/giuliocomi/backoori>



In case the default handler was already explicitly chosen by the user it will be under key:

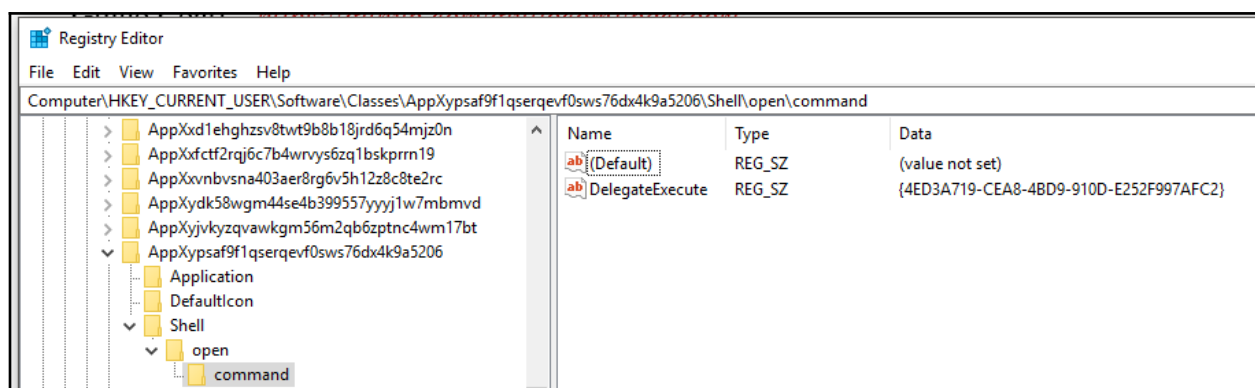
Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\Shell\Associations\UrlAssociations\feedback-hub\UserChoice



And then again, by looking under

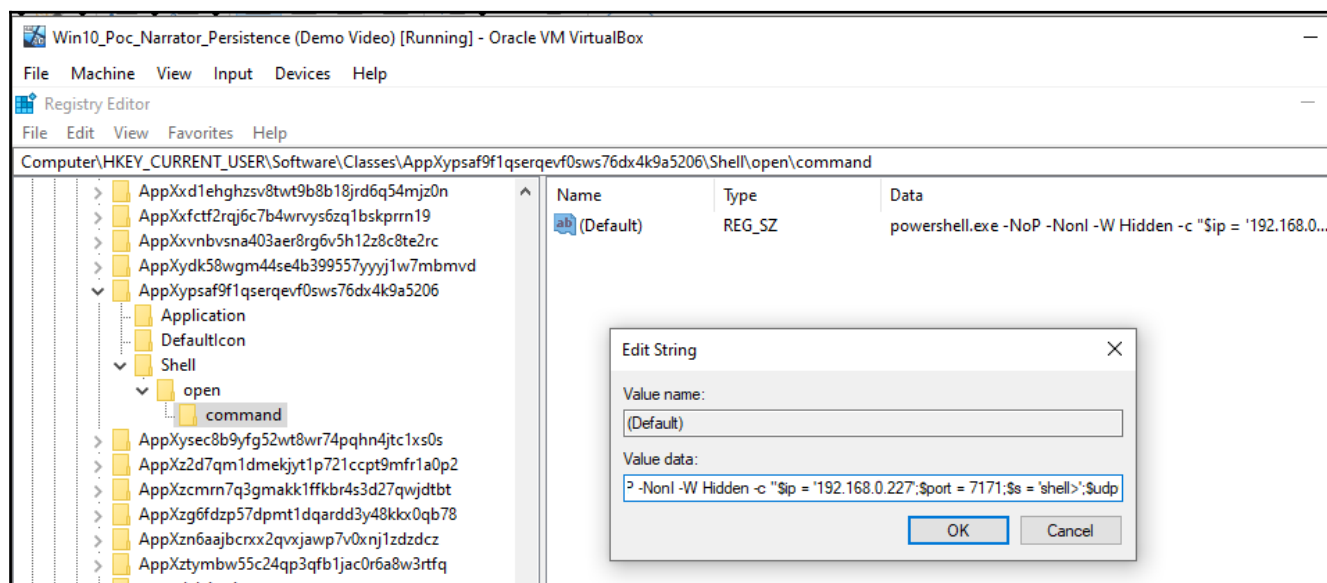
Computer\HKEY_CURRENT_USER\Software\Classes\Appxypsaf9f1qserqevf0sws76dx4k9a5206

we get the following configuration (by the way it is the standard one for all Universal Apps):

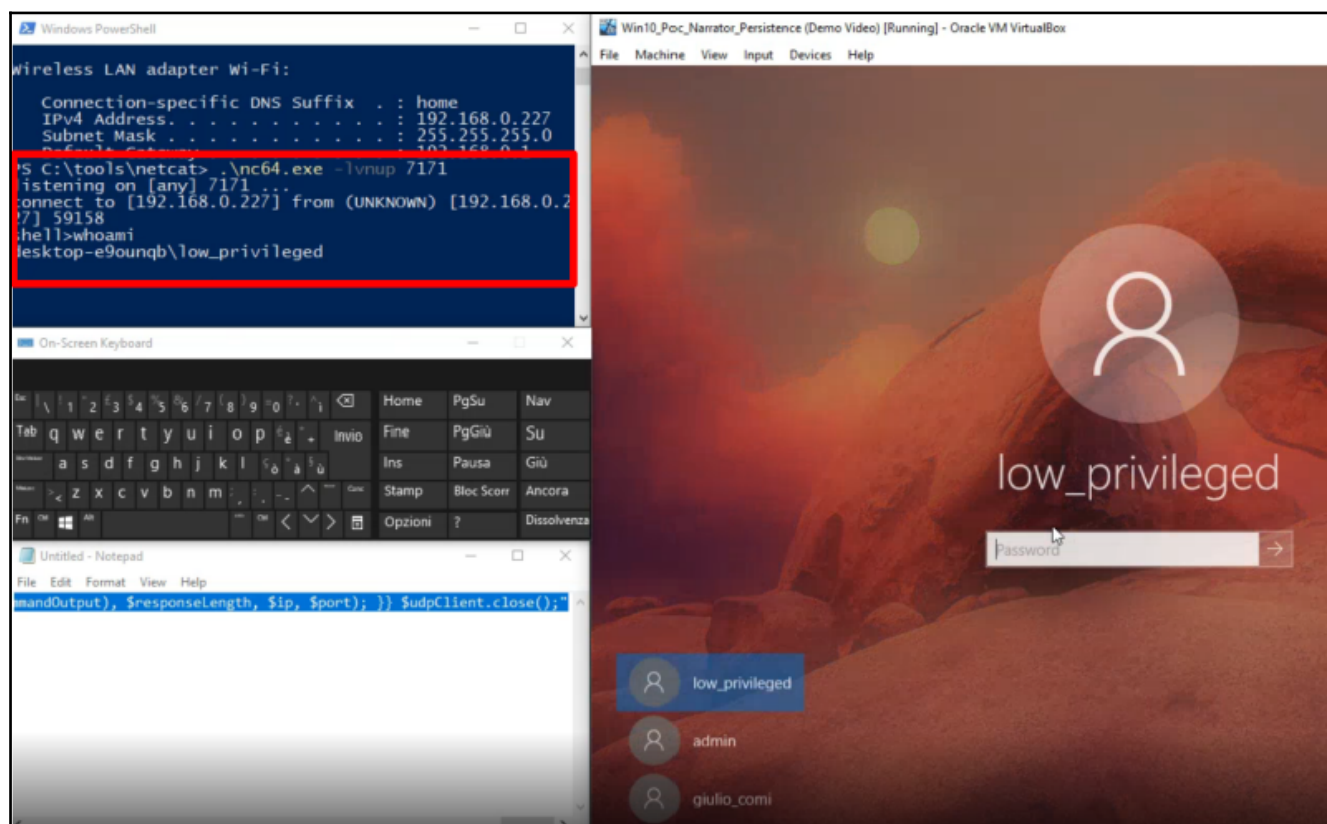


Turned out after a not-so “educated” guess that by getting rid of the *DelegateExecute* entry and then adding a Powershell payload for the ‘Default’ value we will open rooms for this fileless persistence technique:

@giulio_comì – <https://github.com/giuliocomi/backoori>



The payload will be executed by pressing the 'Provide Narrator feedback' shortcut.



Full video here: <https://youtu.be/oPKnYO9V8-M>

@giulio_comì – <https://github.com/giuliocomi/backoori>

Conclusion

The feedback survey feature should not be available in the Windows logon screen, despite not being a trust boundary misconfiguration, because it is a graphic window with a form to send. The possibility to trigger the ‘feedback-hub’ URI scheme in this context exposes Windows 10 users to an additional Accessibility Feature abuse that has the main advantage of being fileless.