Self-Extracting Archives, Decoy Files and Their Hidden Payloads

By Jai Minton - Falcon OverWatch Team

Archived: 2026-04-05 23:43:44 UTC

Self-extracting (SFX) archive files have long served the legitimate purpose of easily sharing compressed files with someone who lacks the software to decompress and view the contents of a regular archive file. However, SFX archive files can also contain hidden malicious functionality that may not be immediately visible to the file's recipient, and could be missed by technology-based detections alone.

The CrowdStrike® Falcon OverWatch™ team recently observed the use of a seemingly empty SFX archive as part of an interactive intrusion. Deeper investigation uncovered that this file — which could easily be overlooked by defenders — had the potential to provide the adversary with a persistent backdoor to the victim environment when paired with a specific registry key.

Watch this short video to see how Falcon OverWatch proactively hunts for threats in your environment.

SFX Archives in Business Environments

SFX archives are executable files that function by appending the content to be archived to a decompressor stub, which is what is executed upon running the file. This stub seamlessly decompresses and displays the file contents, saving the recipient from requiring specialized software.

PE File Header
Decompressor/Method Stub (the executable code)
Archive File Overlay (the content to be shared)

Figure 1. Standard SFX archive format Because of the ubiquitous nature of unarchiving software, SFX archives are far less common in corporate settings than their standard compressed archive counterparts. Although many software installers may also use an SFX archive for ease of installation, use of these archives is gaining traction among adversaries as a way of bypassing security tools and running malicious code.

Not All SFX Archives Are Equal

A wide variety of SFX decompressor stubs are in use, depending on the product used to create an SFX archive, and not all products have the same functionality. Many of these tools are widely available and are user-friendly to operate. Two examples of software that allow the creation of an SFX archive are 7-Zip and WinRAR — each has a specific version of a decompressor stub, and both have features that can be used or abused.

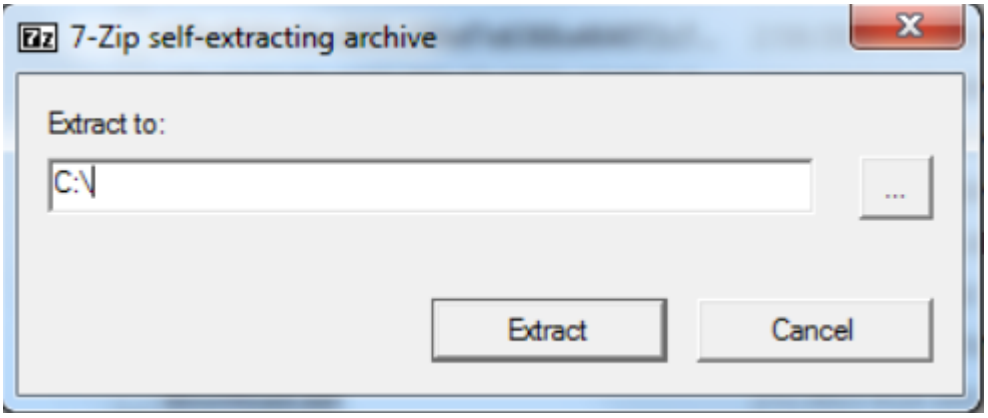

Figure 2. 7-Zip standard SFX archive prompt (click to enlarge)

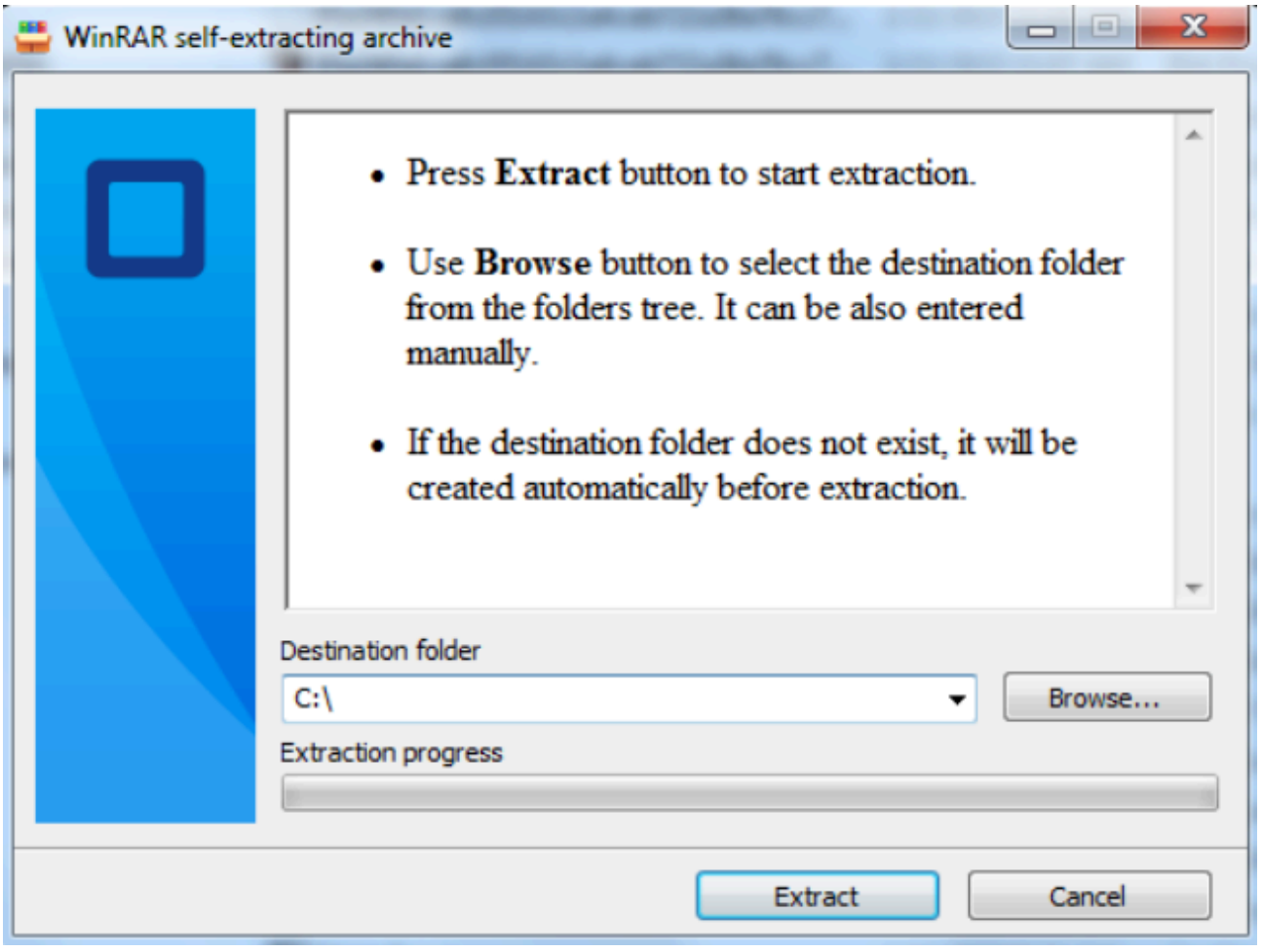

Figure 3. WinRAR standard SFX archive prompt (click to enlarge)

Password-Protected SFX Archives

Password-protected SFX archives are more likely to be seen in business environments where a commercial product is used to protect a file by encrypting it and requiring a password for access. The resulting file is often an SFX archive with an executable extension that can only be accessed if the correct password is given. This same method of protecting files has also been used to facilitate intrusions.

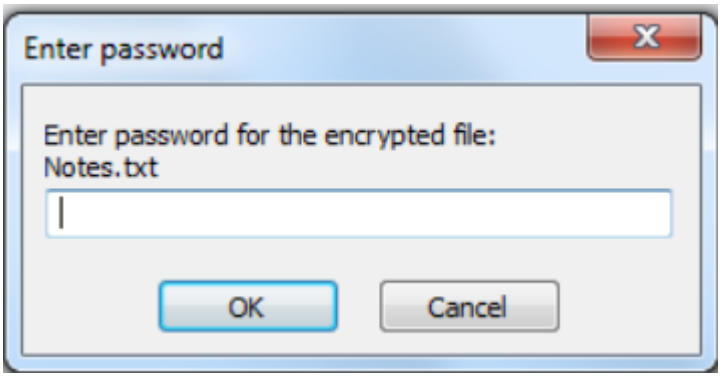

Figure 4. Example of WinRAR encrypted SFX archive prompt (click to enlarge)

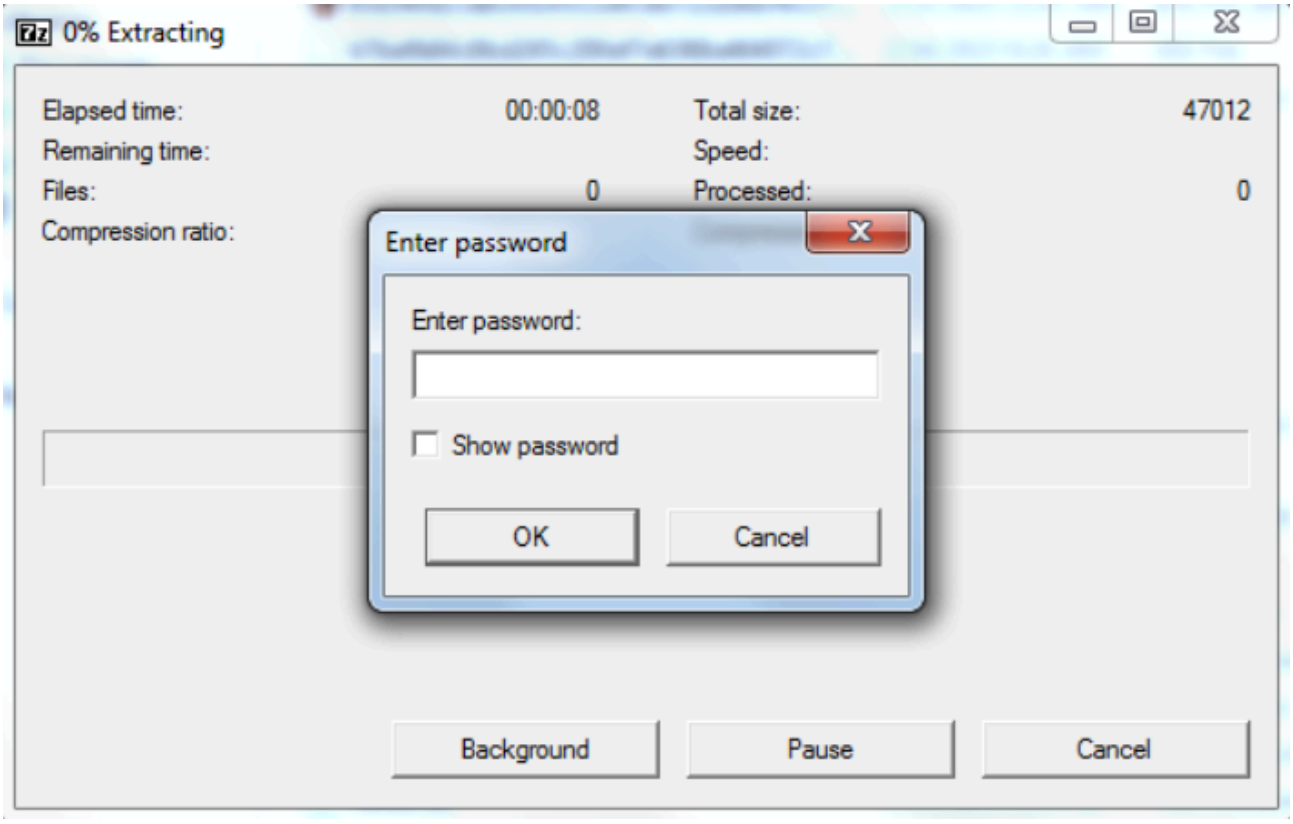

Figure 5. Example of 7-Zip encrypted SFX archive prompt (click to enlarge)

A Trustwave blog post published in October 2022 details how the notorious Emotet botnet was sending out an SFX archive that, once opened by a user, would automatically extract a second password-protected SFX archive, enter the password, and execute its content without further user input. The archive also displayed a decoy file to avoid raising suspicions.

New evidence indicates that core SFX archive functionality is being abused in different ways.

The Curious Case of an Empty Archived File

Falcon OverWatch threat hunters recently discovered an adversary attempting to establish persistence through the use of an Image File Execution Options debugger after gaining access to a system using compromised credentials.

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\utilman.
```

This command line attempts to configure a debugger in the Windows registry to pass `utilman.exe` as a parameter to the specified debugger executable whenever `utilman.exe` is run. `Utilman.exe` is an accessibility application that can be run before user login. As such, it is commonly abused by adversaries to run a binary of their choosing at the Windows logon screen (commonly `cmd.exe`), bypassing the need to authenticate to a system should they lose access to any compromised credentials they were using. In addition, binaries run through this method are executed under the local system account (NT AUTHORITY\SYSTEM), which allows running commands with higher privileges than that of a standard administrator account.

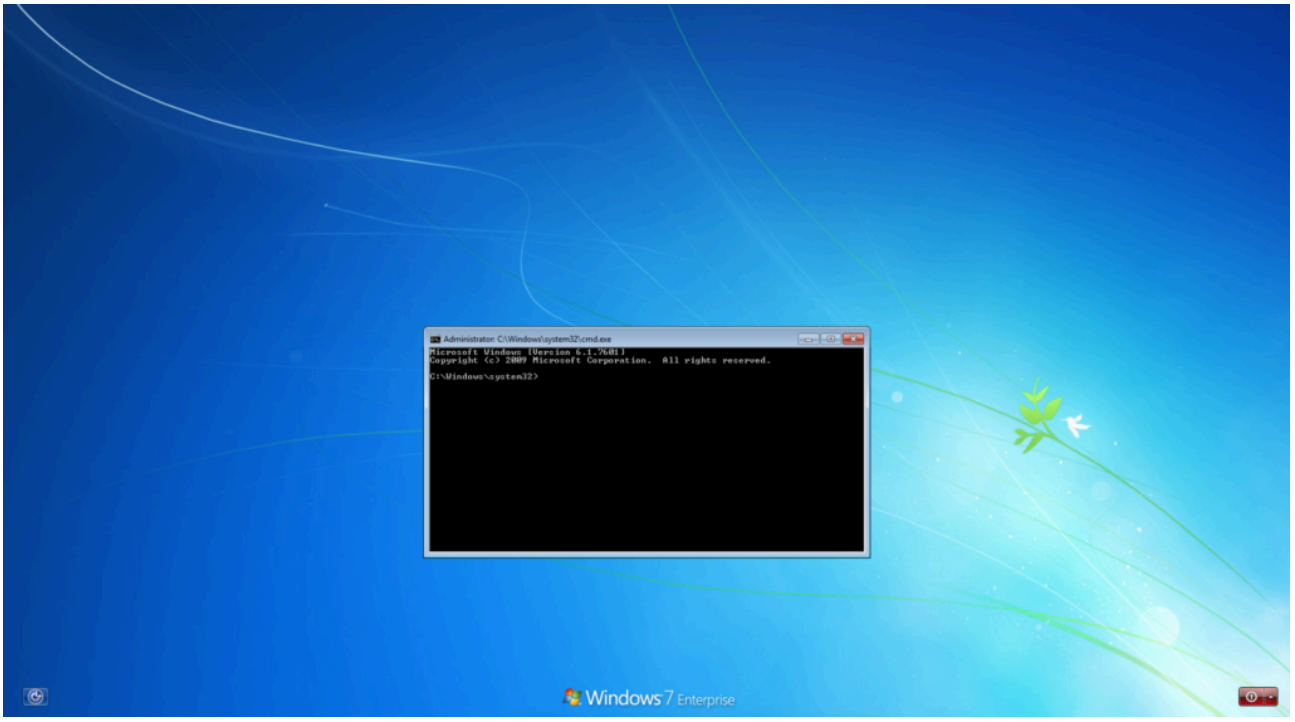

Figure 6. Example of Utilman administrative command prompt at logon screen (click to enlarge)

Although the abuse of `utilman.exe` is not a novel technique, the binary the Image File Execution Options key was pointing to was an SFX archive, which is unusual. Interestingly, this binary was password-protected, so although it is possible to trigger its execution with the debugger, it is not possible to unarchive without the correct password. The execution path of this attack so far is as follows:

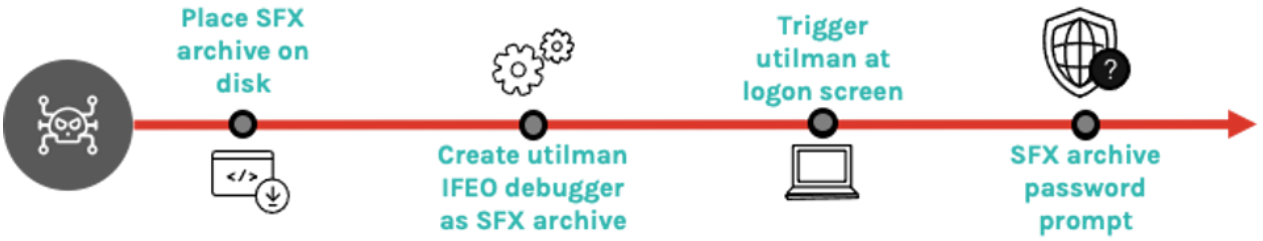

Figure 7. Partial adversarial persistence attack chain (click to enlarge)

One of the advanced features of WinRAR SFX archive files is the ability to include extended SFX commands, which run upon successfully unarchiving a file. Within these commands is a setup option used to specify the executable, which should be run following successful unarchiving. This is commonly abused by adversaries to tell the decompressor stub to run malware contained within an SFX archive once it has decompressed the malware to disk. However, a malicious SFX archive doesn't need to contain malware; instead it could be used to invoke a malicious command using native tooling as part of the decompressor stub functions.

Using this information, Falcon OverWatch uncovered the underlying hidden functionality of the SFX archive file. Because an SFX archive contains a valid archive, the metadata of each file contained within this archive is often not encrypted and password-protected, even if the contents of those files are. Examination of the file metadata within this archive revealed that the archive contained an empty text file created in September 2022, and although this could be construed as benign, this served only as a decoy at first glance when examined.

Closer inspection of the SFX archive revealed that it functions as a password-protected backdoor by abusing WinRAR setup options rather than containing any malware. The file contained an archive comment that read "The comment below contains SFX script commands" and then included several setup commands to be run:

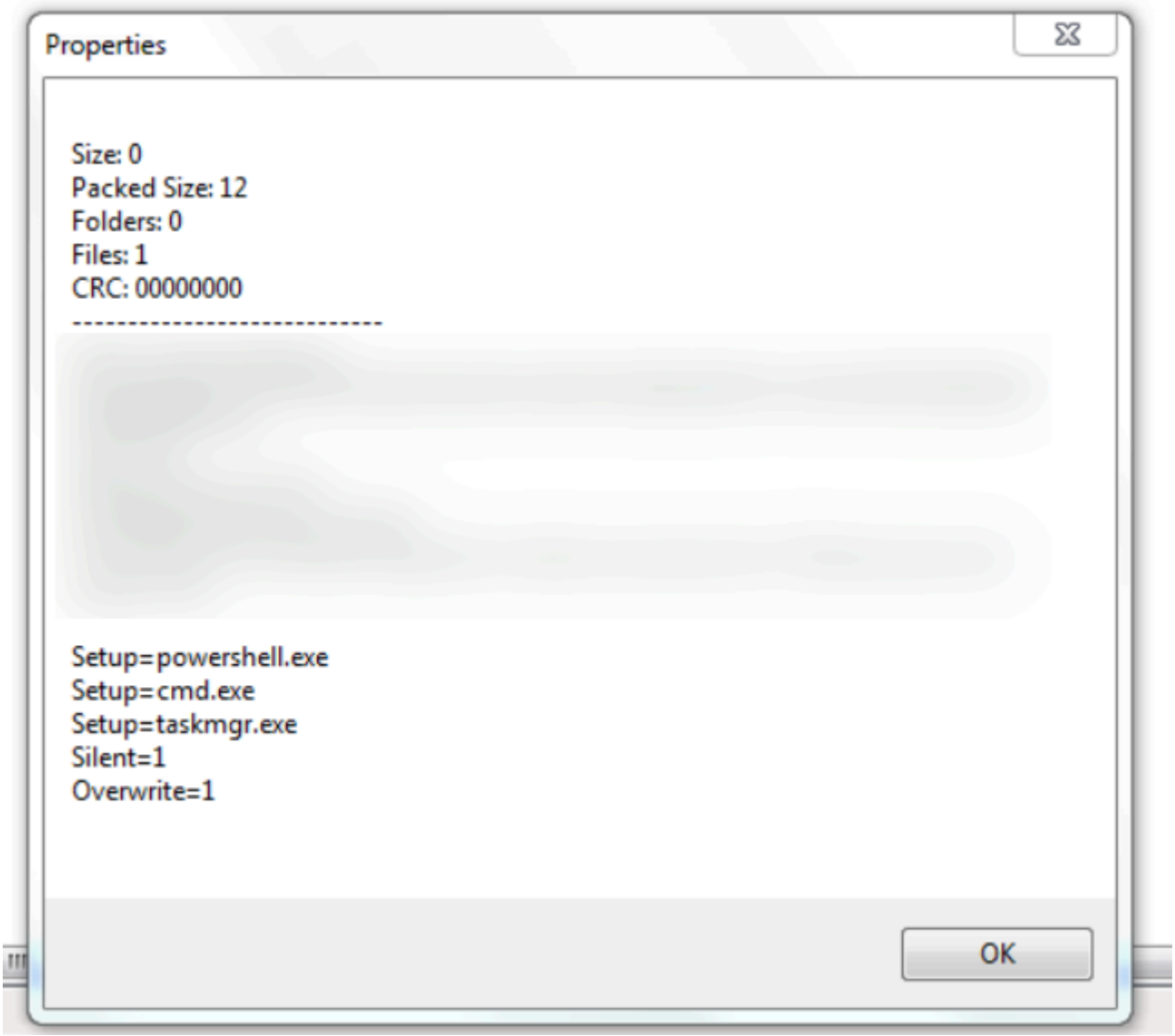

Figure 8. WinRAR setup commands used as SFX archive backdoor (click to enlarge)

These comments are appended to any SFX archive created using advanced options to ensure the decompressor stub knows to automatically overwrite any existing files when extracting the archive, hide any dialogs involved with this process, and upon finishing run `powershell.exe`, `cmd.exe` and `taskmgr.exe`. Creation of such a file through WinRAR shows these settings under "Advanced SFX options."

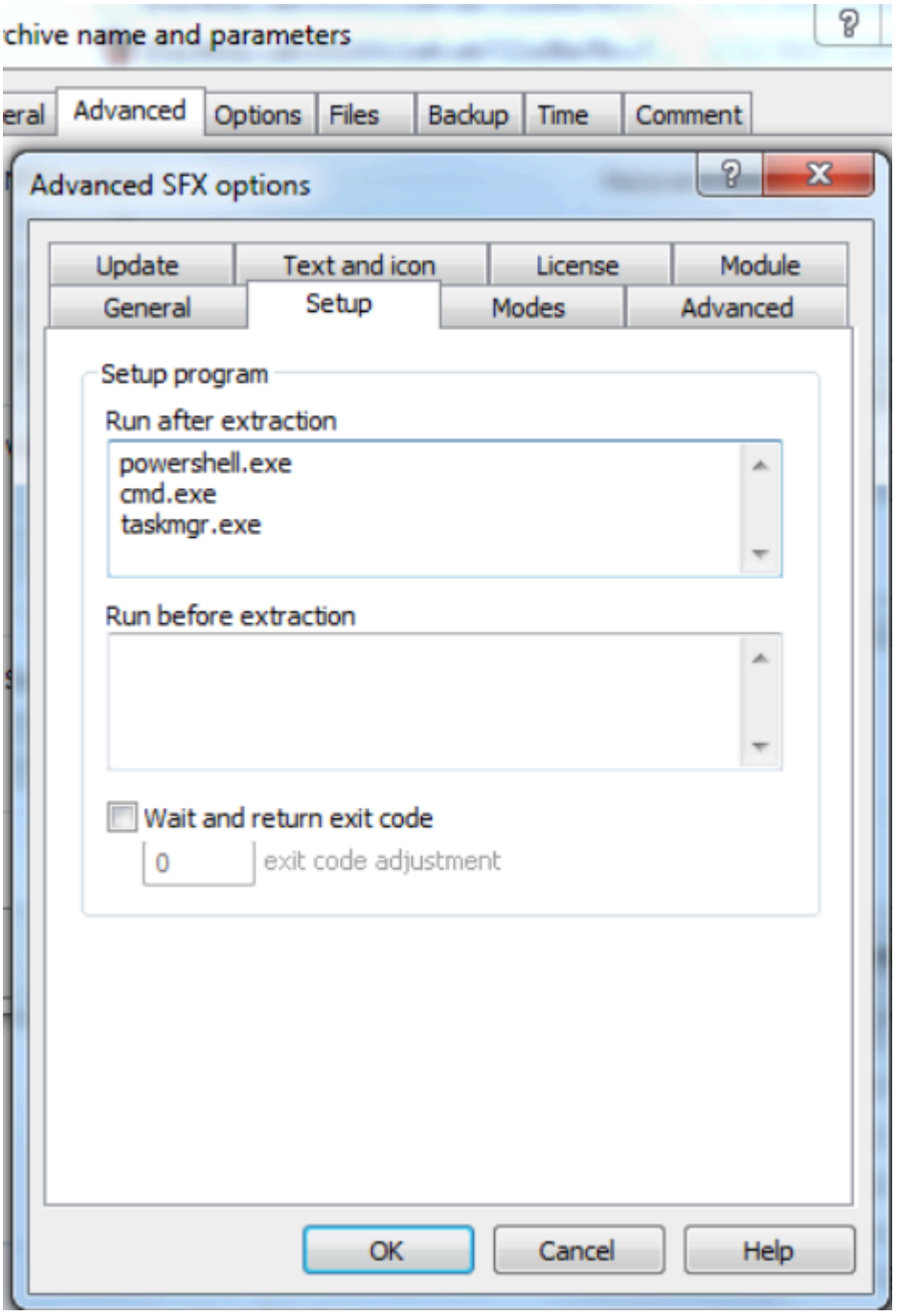

Figure 9. Example of WinRAR advanced SFX setup options (click to enlarge)

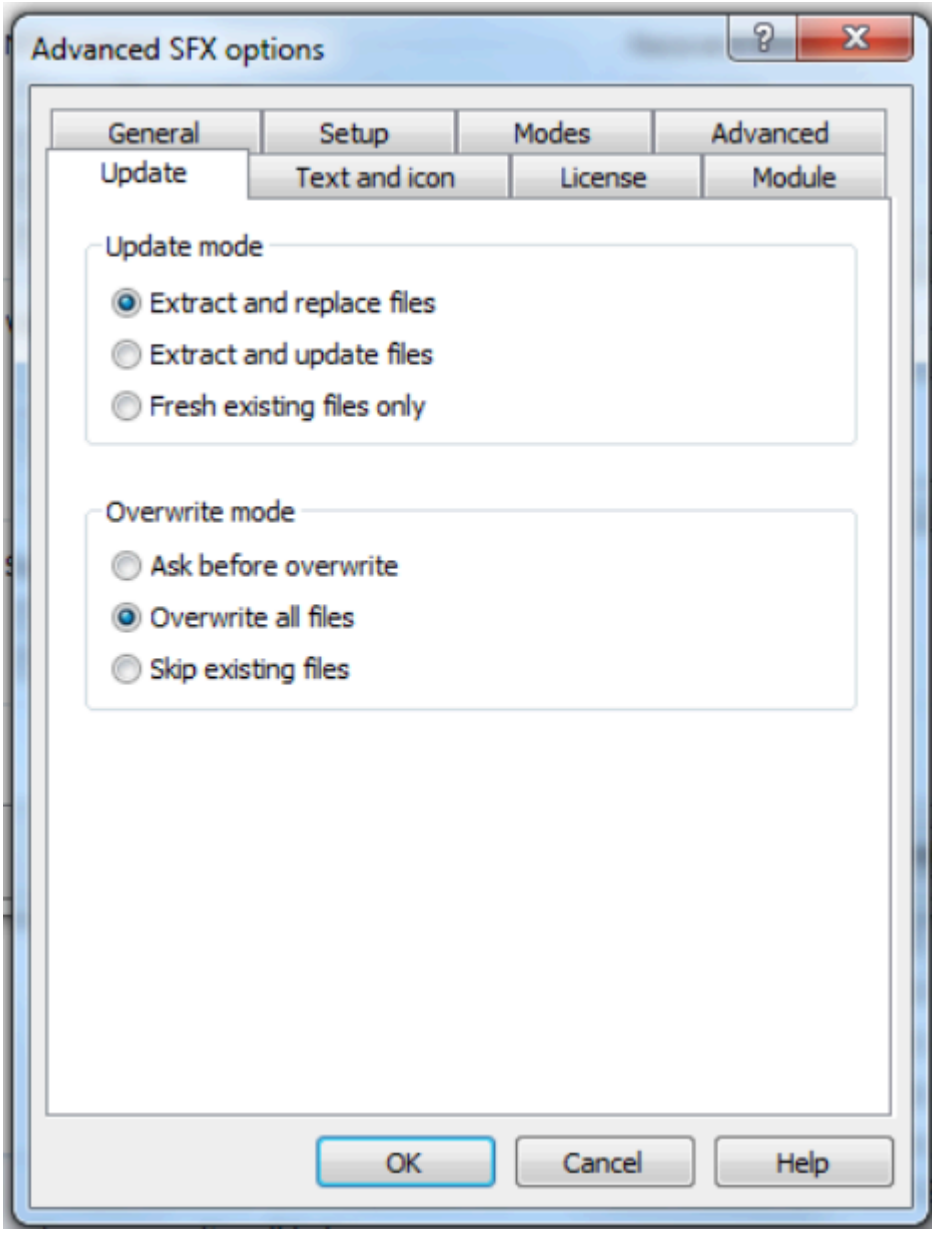

Figure 10. Example of WinRAR advanced SFX overwrite options (click to enlarge)

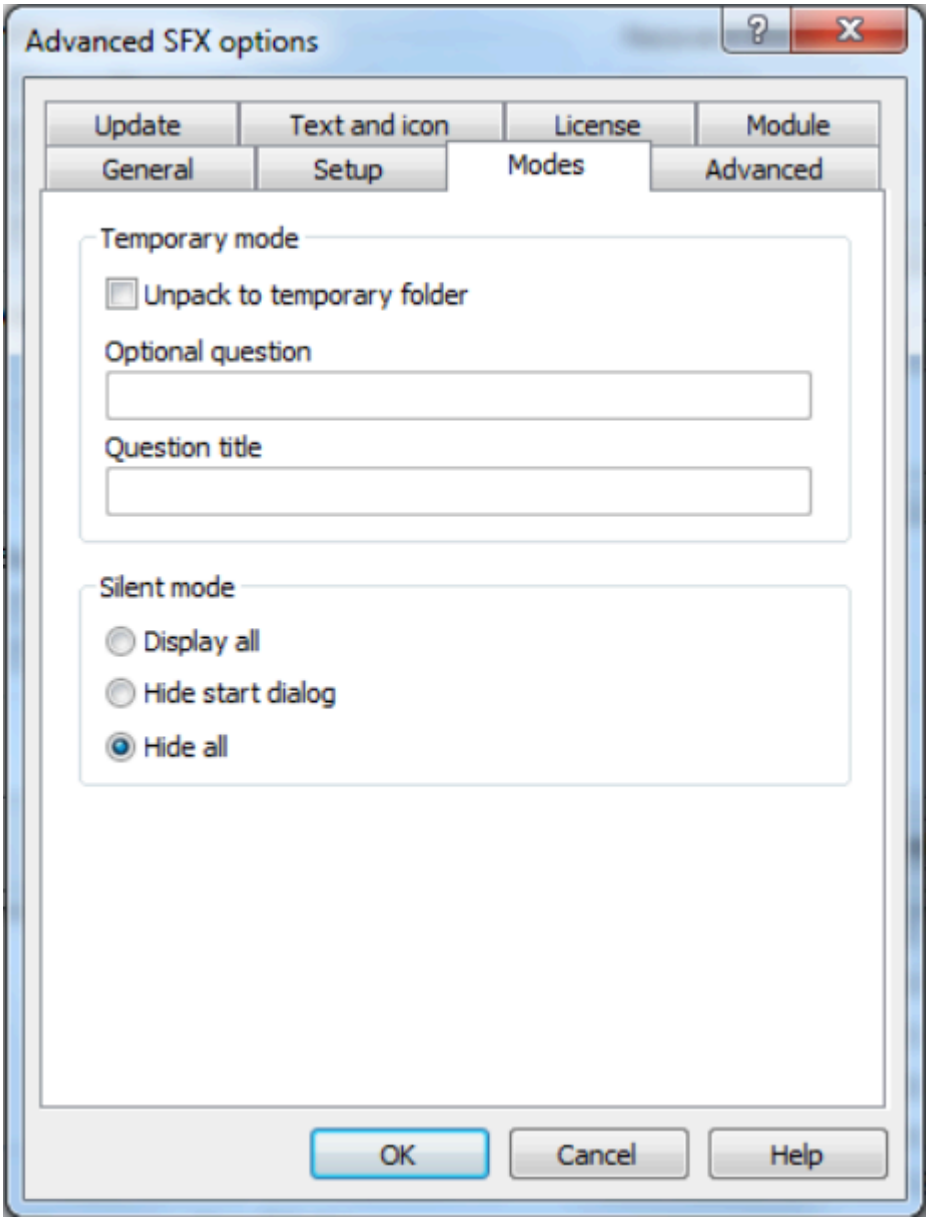

Figure 11. Example of WinRAR advanced SFX silent options (click to enlarge)

Because this SFX archive could be run from the logon screen, the adversary effectively had a persistent backdoor that could be accessed to run PowerShell, Windows command prompt and task manager with NT AUTHORITY\SYSTEM privileges, as long as the correct password was provided. This type of attack is likely to remain undetected by traditional antivirus software that is looking for malware inside of an archive (which is often also password-protected) rather than the behavior from an SFX archive decompressor stub.

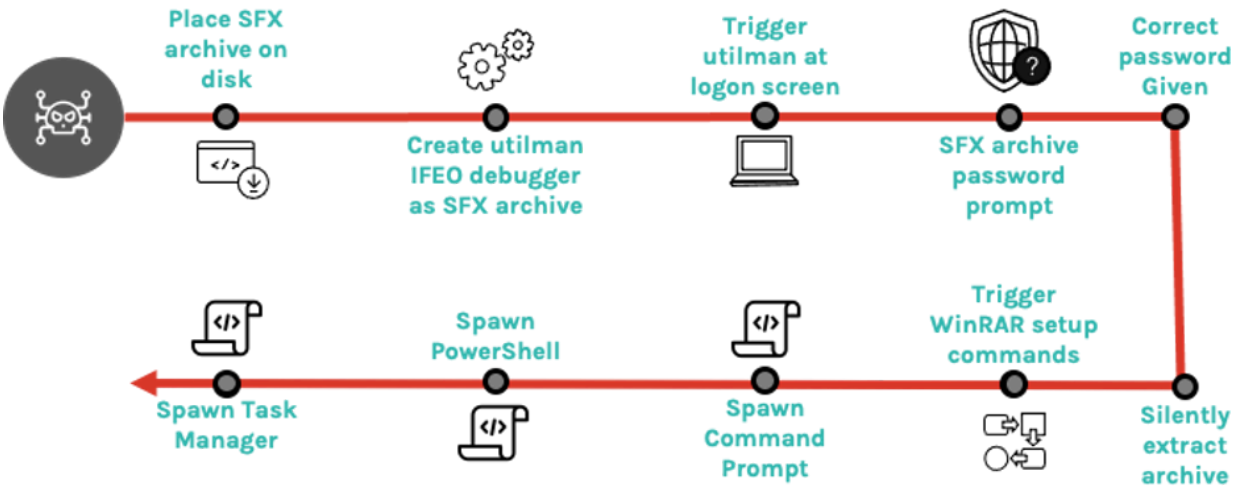

Figure 12. Complete adversarial persistence attack chain (click to enlarge)

Recreating this attack in a lab environment shows the outcome of running the ease-of-access `utilman.exe` binary.

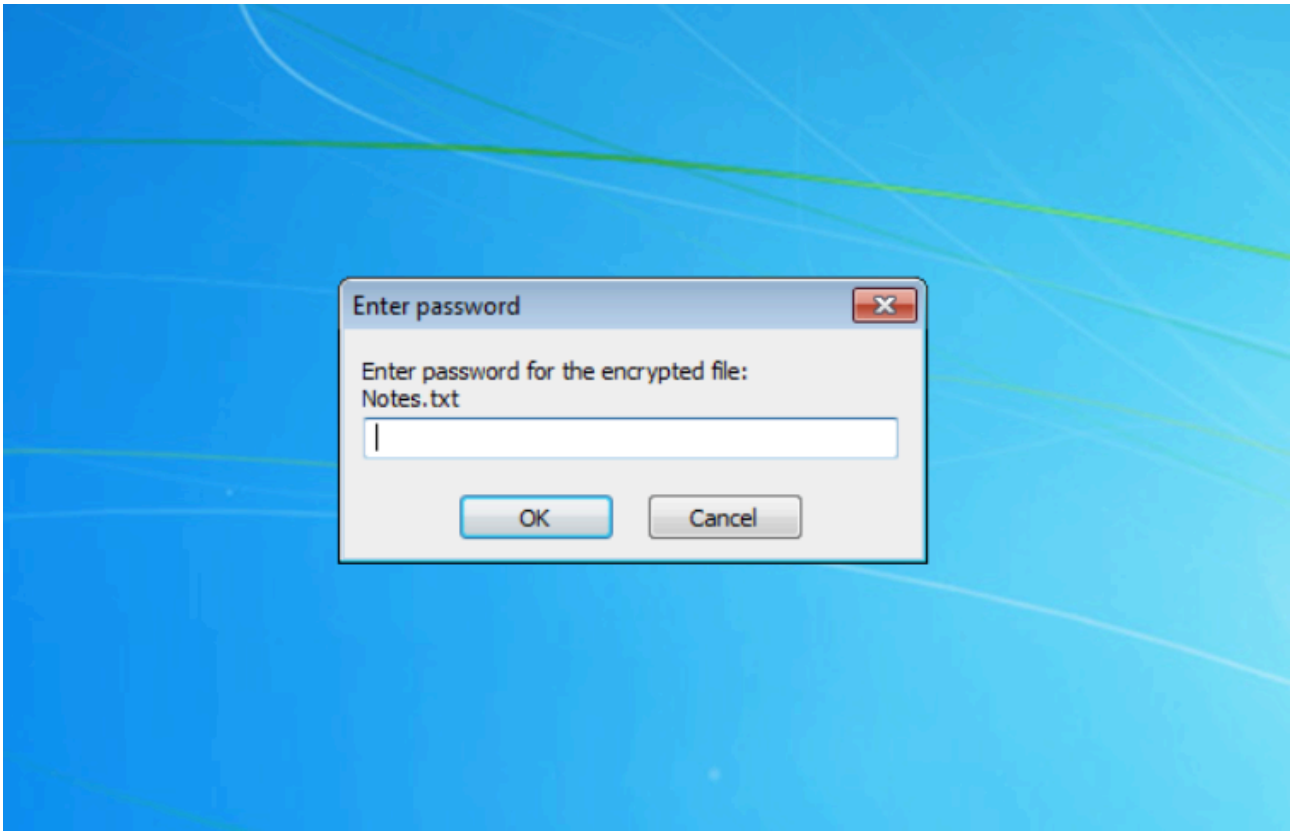

Figure 13. Example of backdoored SFX archive execution at logon screen (click to enlarge)

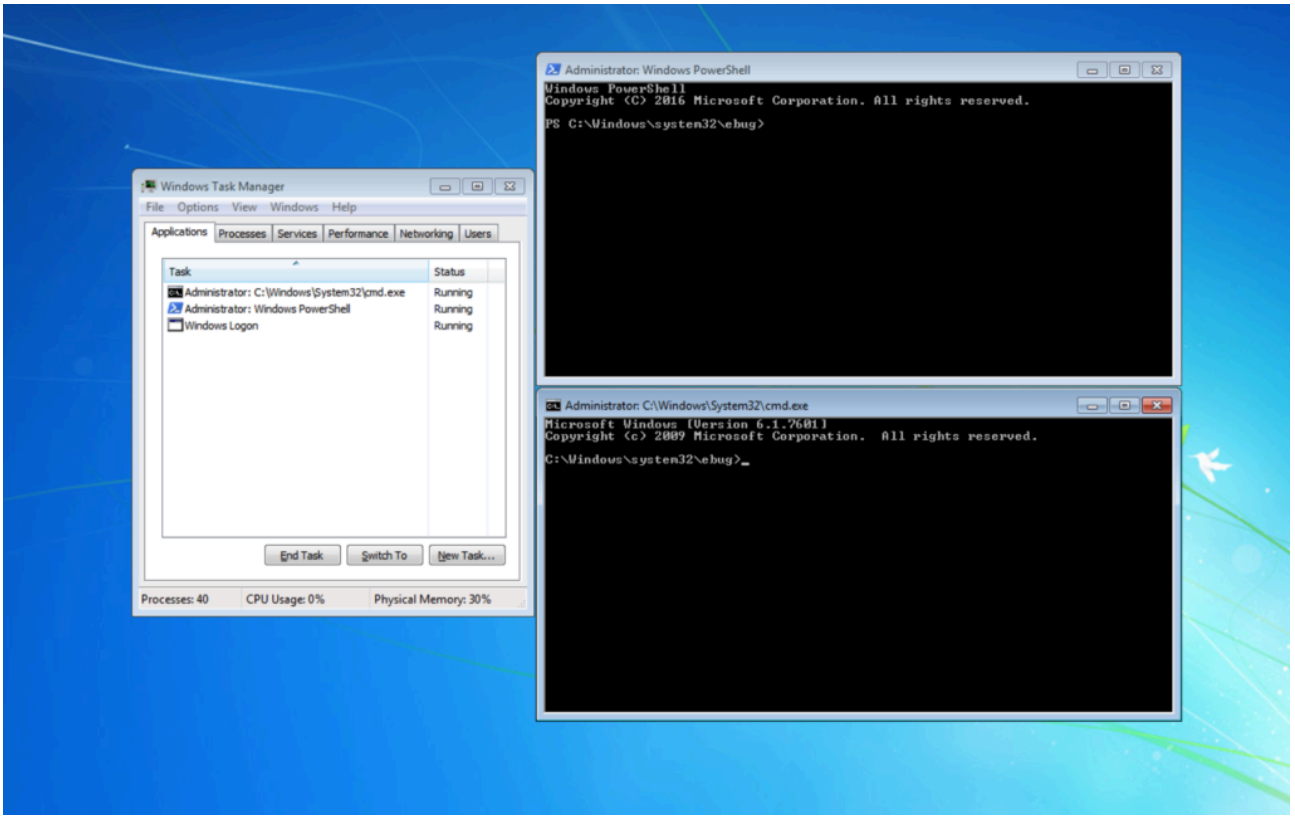

Figure 14. Example of backdoored SFX archive execution with correct password at logon screen (click to enlarge)

Falcon OverWatch Threat Hunters Recommend Proactive Defense in the Face of Widespread SFX Archive Abuse

Hunting across public and private malware repositories found a plethora of WinRAR SFX archives designed to function in different ways. These samples included some that act as download cradles designed to retrieve and invoke a remote payload in memory, some that unarchive a script used to launch malware contained within it, and some that launch malware within the archive but also display a decoy document to the user. Malicious samples that were either password-protected or that contained benign files but used WinRAR setup parameters to execute malicious commands had relatively low detection rates, either at submission or in some cases even after being publicly available for multiple years. This indicates that abuse of WinRAR SFX archives will likely continue being an effective means for an adversary to remain undetected, now and in the future.

Because of the widespread abuse of SFX archives, it’s important to understand the extended functionality provided by some SFX archives, and the various ways adversaries are leveraging these in their intrusions. Some of the ways they’re being abused include:

1. Encrypting a malicious script or executable in an SFX archive that extracts and executes once a correct password is provided.
2. Compressing a benign file while attaching malicious commands to be run as an archive comment that are evaluated as part of the SFX archive decompressor stub (in this instance, we note the use of WinRAR setup parameters).

3. Displaying a decoy document from within an archive while silently running a malicious script or executable upon execution.

To combat this, security professionals should:

1. Examine SFX archives through unarchiving software or other tools to view any potential scripts or executables that are set to extract and run upon execution.
2. Look beyond what is contained within an SFX archive, and examine the functionality provided by the SFX archive decompressor stub itself to identify any commands that will be run either during, before or after successful extraction.
3. Develop a process to validate if a password-protected SFX archive contains malicious or suspicious content.
4. Thoroughly examine any SFX archive that contains only a null-byte file for any added functionality.
5. Wherever possible, use installed unarchiving software to extract or view a SFX archive rather than running the SFX archive itself. Because the archive exists as an overlay, it can also be carved out from the executable using a hex editor if required.

Additional Resources

- *Learn more about Falcon OverWatch's proactive managed threat hunting.*
- *Discover the power of tailored threat hunting provided by Falcon OverWatch Elite.*
- *Find out why part-time threat hunting is simply not enough.*
- *Learn more about the CrowdStrike Falcon® platform.*

Source: https://www.crowdstrike.com/blog/self-extracting-archives-decoy-files-and-their-hidden-payloads/