cyberDefenders Task

scenario :

- The compromised installer came as a ZIP file [version.zip], which the victim extracted before launching the embedded executable [setup.exe].
- The executable was a legitimate Advance IP scanner, which side-loaded a modified Python DLL specifically designed to execute Nitrogen code.
- This process then dropped a Sliver beacon in an AppData subfolder named “Notepad.” [slv.py & data.aes]
- the attacker initiated hands-on keyboard discovery, utilizing Windows utilities such as net, ipconfig, and nltest [Discover.bat]
- additional Sliver beacons were deployed on the compromised host, with persistence established through scheduled tasks and registry key modifications [UpdateEdge.bat]
- the threat actor deployed additional malware for The discovery phase [Discovery.bat]
- dump domain credentials from LSASS using mimikatz [x]
- the threat actor leveraged Impacket’s wmiexec to move laterally to a server [x]
- they used curl to download a ZIP file containing their tools [Tools.bat]
- they repeated the same persistence techniques observed on the beachhead, creating scheduled tasks and modifying registry keys [up.bat]
- Using Restic, the attacker exfiltrated data from the file shares to a remote server [Tools.bat]
- they ran a batch script on the domain controller , which changed that accounts credentials [up.bat]
- the attacker began distributing the BlackCat ransomware binary [example.exe]
- The final script executed a series of actions on remote hosts, including configuring them to start in Safe Mode with Networking and setting a registry run key to launch the ransomware binary upon reboot [1.bat]

Challenge Files :

Version.zip contained mainly:
- setup.exe (Advanced IP Scanner executable)
- two hidden Python DLLs (python311.dll & python311x.dll)
- service_probes.aes
- important.txt

walkthrough :

- version.zip

- setup.exe
- python311.dll
    - decrypts an AES-encrypted 'service_probes.aes'
- python311x.dll
    - Discover.bat & 1.bat & up.bat & Tools.bat & Discover.bat
- service_probes.aes
    - pycryptodome.bat & & data.aes
    - slv.py
        - decrypts an AES-encrypted 'data.aes' and execute 'pythonw.exe'
    - pythonw.exe
        - worksliv.py & wo14.py & wo12.py
        - example.exe
            - RECOVER-wragz12-FILES.txt & UpdateEdge.bat & example.py

create the challenge :

[01] ---

I used Reverse Shell Generator website to generate python shell [shell2.py] then i used [encode_aes_to_base64.py] to encrypt the shell with AES 128 in base64 format [shell2.txt] then i write the script and obfuscated it using PyFuscate [wo14.py]

- I used AES 128 with key "we3p2v5t85"
- my C2 : (192.92.250.60:443)

I used Reverse Shell Generator website to generate python shell [shell3.py] then i used [encode_aes_to_base64.py] to encrypt the shell with AES 128 in base64 format [shell3.txt] then i obfuscated it using PyFuscate [wo12.py]

- I used AES 128 with key "tiqny2q2je"
- my C2 : (192.92.250.65:443)

I used Reverse Shell Generator website to generate python shell [shell4.py] then i used [encode_aes_to_base64.py] to encrypt the shell with AES 128 in base64 format [shell4.txt] then i obfuscated it using PyFuscate [worksliv.py]

- I used AES 128 with key "tiqny2q2je"
- my C2 : (192.169.175.134:8443)

[02] ---

create company.exe
note : for fake ransomware & clearing logs
it drops [UpdateEdge.bat] and [example.py]

example.py which configured the ransomware, cleared logs (i obfuscated it using PyFuscate) work after 60 sec

[03] --

create pythonw.exe (which will change to [updateJson.exe])
drop in (" c:\Windows\adfs\py\ ") and execute 4 files : (worksliv.py & wo14.py & wo12.py & company.exe)

[04] --

I used Reverse Shell Generator website to generate python shell [shell1.py] and use [encrypt_aes128.py] to create [data.aes]

- I used AES 128 with key "we3p2v5t85"
- my C2 (192.49.94.18:8443)

create [slv.py] (i obfuscated it using PyFuscate) to

- decrypt 'data.aes' file and run it (Fileless Malware technique)
- change name and execute [pythonw.exe] to [updateJson.exe]

[05] --

create [service_probes.exe] :

- to drop files (pythonw.exe, pycryptodome.bat, slv.py, data.aes) in (%AppData%\Notepad)
- then AES-encrypted service_probes.aes using [encrypt_aes128.py]
    - i used key "we3p2v5t85"
    - i used pycryptodome.bat to install pycryptodome and update pip

[06] --

create [python311.c] to create [python311.dll] :

- decrypt AES-128-CBC encrypted file [service_probes.aes] and execute it
  create [test_loader.c] to test [python311.dll]

[python311] ---

to test [python311.dll] :

- python311.dll
- test_loader.exe
- service_probes.aes

You can watch the result **[python311.mp4]**

[06] --

create UpdateEdge.bat : (for scheduled tasks)
used to run : wo12.py

create Discover.bat :
discover the utilizing Windows utilities such as net, ipconfig, and nltest and sent them to attacker server

create Tools.bat :
used curl to download a ZIP file containing their tools and use restic and PsExec64 and sent them to attacker server

create 1.bat :
start in Safe Mode with Networking and setting a registry run key to launch the ransomware binary upon reboot and login in with "blackcat" user

create up.bat :
create new user and Add to Administrators group
same persistence techniques observed on the beachhead, creating scheduled tasks and modifying registry keys

[08] --

create [python311x.c] to create [python311x.dll]
create [loader.c] to test [python311x.dll]

[python311x] --

to test [python311x.dll] :

- python311x.dll
- loader.exe

You can watch the result **[python311x.mp4]**

[09] --

create [setup.exe] to run Advanced IP Scanner and both (python311.dll & python311x.dll)

[Version] --

to test [setup.exe] :

- python311.dll
- python311x.dll
- service_probes.aes

You can watch the result **[setup.mp4]**