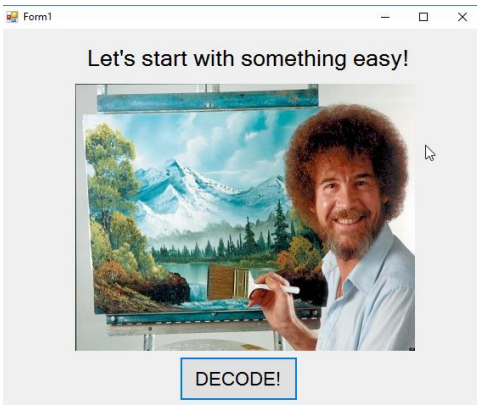
Environment setup

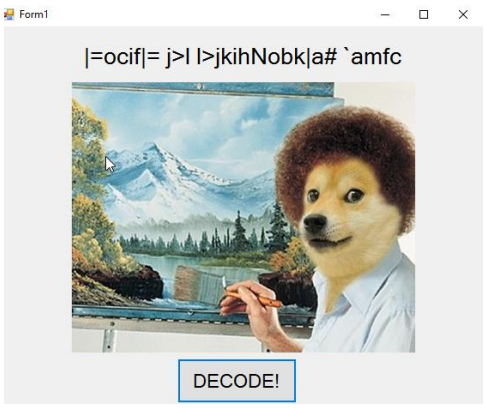
Environment setup is crucial when conducting such an experiment to ensure the vulnerability does not affect the personal infrastructure, including but not limited to the host machine, local network, etc. It is always recommended to create a virtual environment to perform such experiments. One such environment was also made while performing this experiment. We need to work in the Windows VM with protection disabled. We also need CFF explorer, strings.exe, ILSpy, and an editor like Visual Studio.

Description of techniques and exploits:

After setting up the environment, we run our malware 'PMA132.exe,' and screenshot 1 and 2 shows the result.

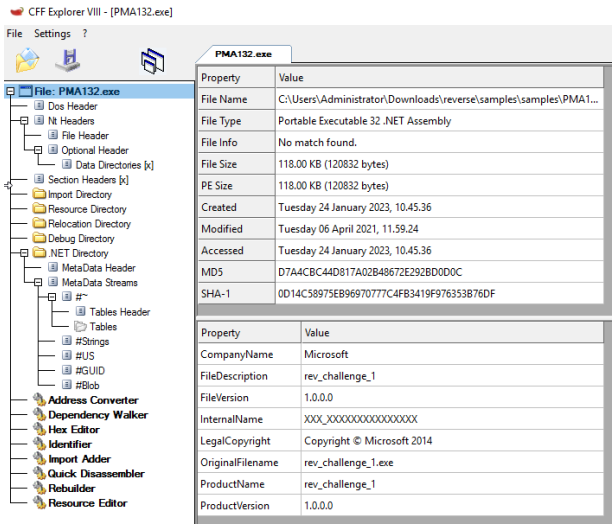


Screenshot 1

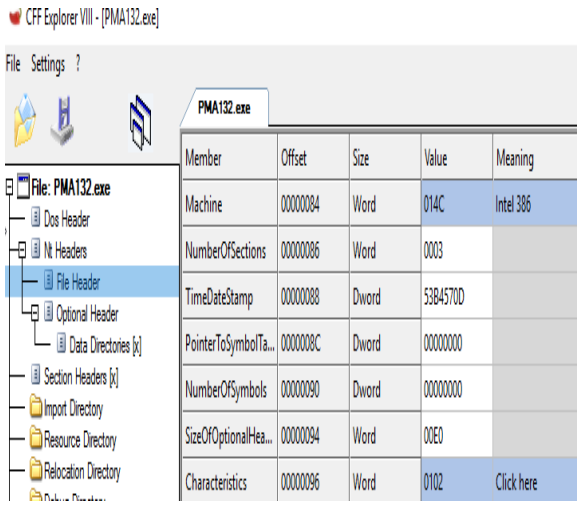


Screenshot 2

So, to identify file details, we open the malware with CFF explorer. We can see from screenshots 3 and 4 about the file type and machine type that this malware is supposed to run.



Screenshot 3



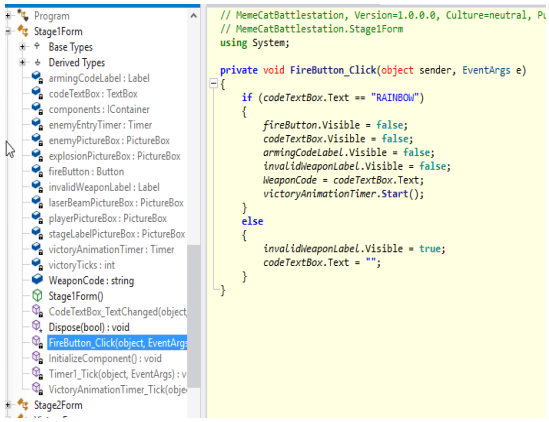
Screenshot 4

After that, we use ILSpy to decompile the malware. In the Form1 section, we see 'btnDecode_Click', which contains data_secret, which can be used to decode the encoded part. So to decode, we add C#

code which will contain the secret decoding file in the command line, and when we compile and run, we get the flag as shown in screenshot 5.

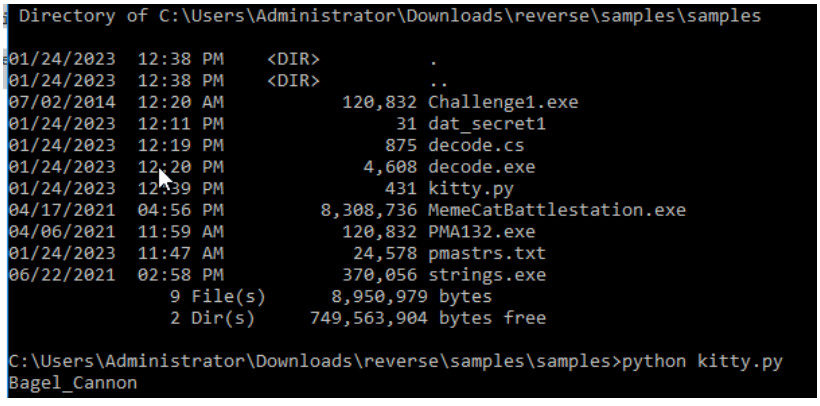
Screenshot 5

After this, there is one more malware that I analyzed and decompiled named 'Battlestation.exe'. As we can see from screenshot 6, we need some text to attack the weapon we need to find. So we decompile the malware using ILSpy, and as shown in screenshot 7, there are two stages form. For the stage, I saw that 'FireButton_Click' had a rainbow as text that can be used to attack stage 1 and works.



Screenshot 6 Screenshot 7

For stage 2, under 'isValidWeapon', it shows an array that is true and the same as shown in the section. So to decode that array, we just use the decrypt function and add that cipher/array in it, and we get the flag as shown in screenshot 8.



Screenshot 8