**أنواع الـ Shellcode :** ( غير مُكتمل )

* Connect shellcode
* Bind shell
* Socket reuse
* Stagged
* Download and execute

**ملاحظات:** ( قابل للتعديل )

* الــ Shellcode يجب أن يكون مُمثّل بالــ Machine code
* يوجد عدة طرق لكتابة الــ Shellcode ومن ثم تحويله الى الــ Machine code
* فيما يلي بعض الطرق ( اليدوية ، بدون الإستعانة بأدوات تحاكي العملية ) :
  + **1 – بإستخدام لغة الأسمبلي Assembly** 
    1. كتابة الــ Shellcode بلغة الــ Assembly
    2. عمل Compile للــ Assembly code ، المخرج > file.obj
    3. عمل Disassemble للــ file.obj ( الهدف من الخطوة (2)و (3) هو الحصول على الــ Machine code المقابل للــ Assembly code )

مثال (هذا المثال فقط لتوضيح جزئية الــ Machine code ، بغض النظر عما يقوم به كود الأسمبلي المقابل له) :



هذه هي الجزئية الخاصة بالــ Machine code

بعد عملية الــ Disassemble

* + 1. إضافة الــ Prefix التالي : **\x** قبل قيم الــ HEX في الــ Shellcode
  + **2 – بإستخدام أحد اللغات البرمجية عالية المستوى (high-level language)**
    1. كتابة الــ Shellcode بأحدى اللغات عالية المستوى ( لغة C مثلًا (
    2. عمل Compile للــ file.c، المخرج > file.exe
    3. عمل Decompile للــ file.exe ( الهدف من الخطوة (2)و (3) هو الحصول على الــ Machine code )

|  |  |
| --- | --- |
| |  | | --- | | https://mail.google.com/mail/u/0/images/cleardot.gif | |

**ملاحظات خاصة بكتابة Shellcode على نظام  Windows :** ( قابل للتعديل )

* البرامج (.exe ) الخاصة بنظام التشغيل Windows تحتوي على مكتبات ( أو ملفات .dll ) مضمنة معها
* ملفات الـ dll. في الأساس ماهي إلا ملف برمجي قد يُكتب بلغة C أو لغات أخرى
* الوظيفة الأساسية لملفات الـ dll. دعم عمل للبرنامج الأساسي (.exe ) والغرض منها يختلف بحسب طبيعة البرنامج والوظيفة التي يؤديها، على سبيل المثال برنامج (FTP ) مدمج معه ملف .dll معين يُتيح للبرنامج الوصول الى بعض موارد النظام
* ملفات الـ dll. قد يتم مشاركتها (أي إمكانية الوصول لها) بين أكثر من برنامج ( أي أكثر من برنامج يستطيع إستخدام نفس ملف الـ dll. في نفس الوقت)
* في مرحلة كتابة الـ Shellcode قد نقوم بكتابة Shellcode الهدف منه هو "فتح جلسة في الـ Command prompt في النظام"، يوجد بعض الأمور التي يجب التأكد منها في حالة أردنا كتابة مثل هذا الـ Shellcode وهي :
* التأكد من وجود ملف .dll يحتوي على دالة ( function ) تقوم بهذا الغرض، حيث أننا سنقوم بإستخدام هذه الدالة الموجودة في ملف الـ .dll ونداءها ( في داخل الـ Shellcode الخاص بنا ) عن طريق جلب عنوان الذاكرة الخاص بها، وهذا العنوان نجده من خلال البحث في ملف الـ .dll ( نقوم بعملية disassemble له )
* في حالة عدم وجود أي من أساليب الحماية مثل ASLR فالعناوين الخاصة بالدوال في ملفات الـ .dll تكون ثابته (على سبيل المثال لو قمنا بعمل disassemble لملف الـ .dll في جهاز آخر سنحصل على نفس العنوان )
* لكن في حالة وجود أحد أساليب الحماية من "إستغلال ثغرات فيضان ذاكرة التخزين المؤقت" فالعناوين الخاصة بالدوال تختلف ( مع كل عملية تشغيل للجهاز يتم إعطاء عناوين جديدة )