UPX Packing

Bu yazıda C dilinde hazırlanan basit uygulamaların yaygın kullanılan paketleme yazılımlarından biri olan UPX paketleyiciyle paketlendiklerinde nasıl görüldükleri incelenmeye çalışılacaktır. Analiz süresince Assembly dili üzerinde çalışmalar yapılacağı için Assembly diline hakim değilseniz bu yazıdan önce "**Assembly**" dizini altındaki yazılarıma göz atmanızı tavsiye ederim.

Hazırlanan kaynak kod derlendikten sonra UPX paketleme aracıyla paketleyebilmek için kullanılan komut satırında "**upx <Command> < Options> <Input File Name> <Output File Name>**" formatında paketlenecek veya paket dışına çıkarılacak dosyanın belirtilmesi gerekiyor (UPX paketleme aracına kendi sitesinden kullandığınız işletim sistemine uygun halini indirebilirsiniz). Bunun için UPX dosyasının bulunduğu dizine giriş yapılarak "**upx -9 -o <Output File Name And Directory> <Output File Name And Directory>**" komutu kullanılarak dosyalar UPX aracıyla paketlendi.

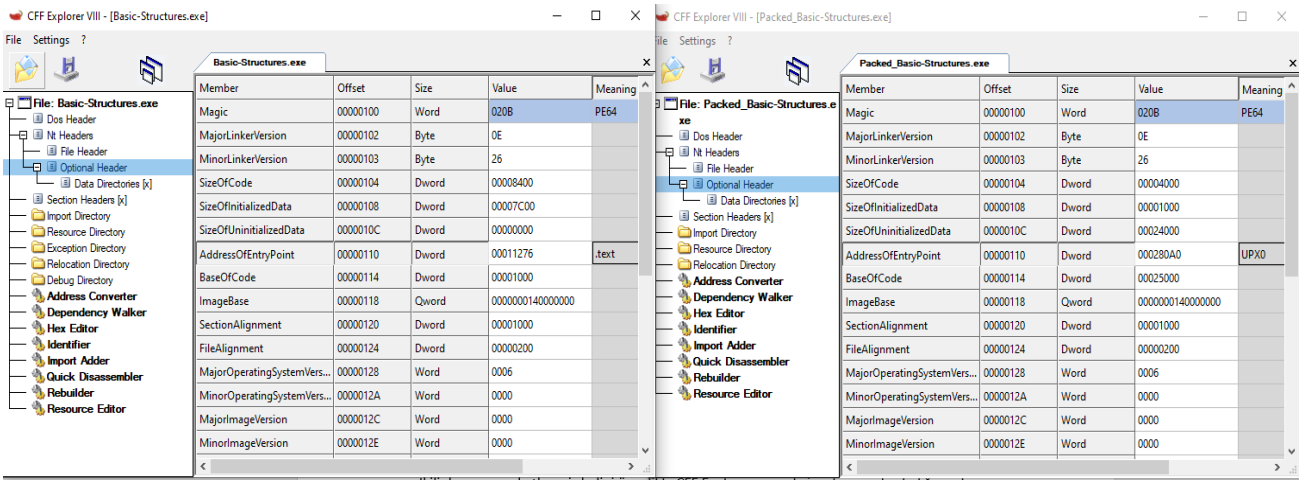
```
C:\Users\VMUser\Downloads\upx-4.2.2-win64>upx.exe -9 -o .\..\..\Desktop\Packed_Basic-Structures.exe ".\..\..\Desktop\Bas
ic-Structures.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2024
UPX 4.2.2       Markus Oberhumer, Laszlo Molnar & John Reiser    Jan 3rd 2024

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     64512 ->     15360   23.81%    win64/pe     Packed_Basic-structures.exe

Packed 1 file.
```

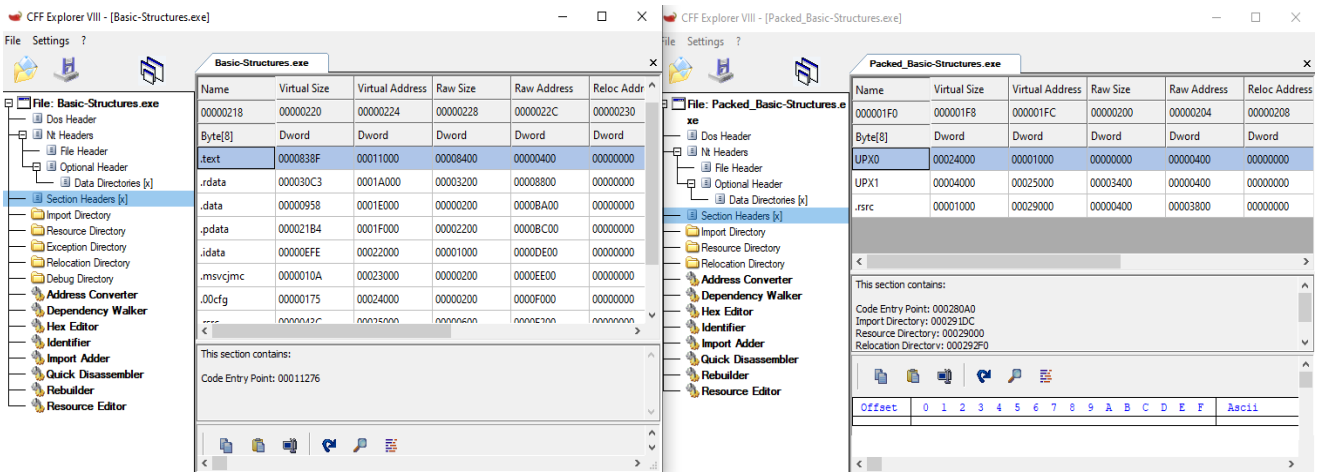
İkili dosyaları öncelikle CFF Explorer aracı üzerinde incelediğimizde;

- İlk göze çarpan kısmı "Optional Header" kısmındaki program kodlarının başlangıç adresini gösteren "AddressofEntryPoint" değeri kodun bulunduğu ".text" sekmesini göstermesi gerekirken farklı bir alandan çalışmaya başlayacağını gösteriyor.

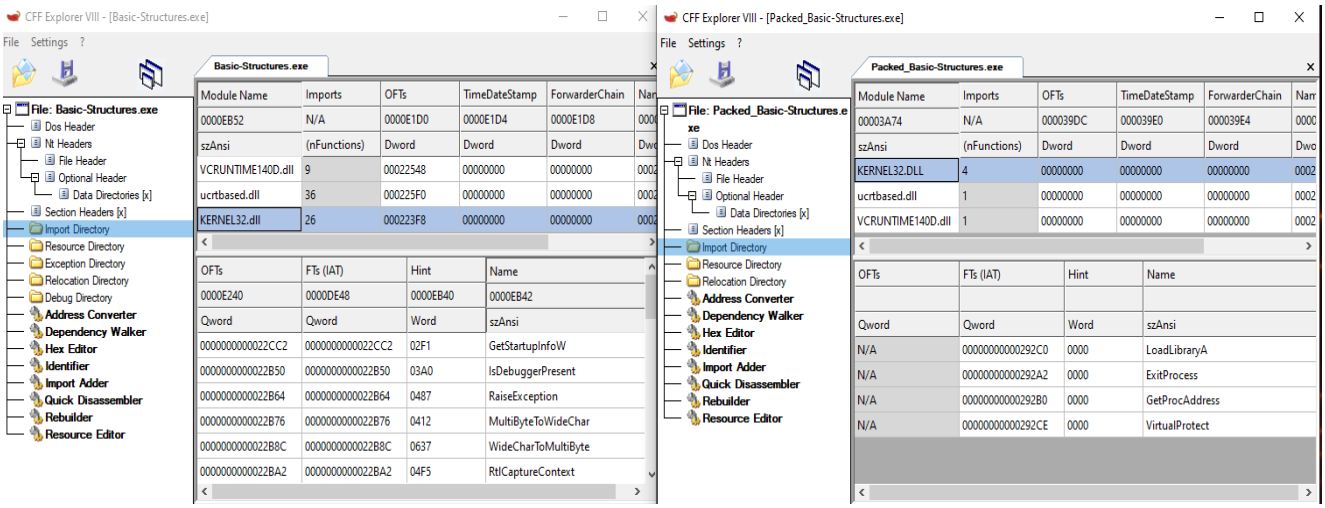


- Göze çarpan ikinci bir kısım ise "Section Header" alanındaki başlık bilgileri oldu. Burada Paketlenmemiş dosyada normal bir ikili dosyada olması gereken ".text", ".data" gibi sekemler bulunurken paketlenmiş ikili dosyada "UPX0", "UPX1" gibi normalin dışında sekmeler görünüyor.

- Burada dikkat edilmesi gereken ikinci bir detay ise **Raw_Size** (Section'ın disk üzerindeki boyutunu gösterir) sütunundaki değerler olmalıdır. Paketlenmemiş ikili dosyada **.text** kısmında bu değer sıfırdan farklıyken paketlenmiş ikili dosyada UPX0 (.test/.code bölümü) kısmı için bu değer sıfırdır. Bu durumdan **Raw_Size** ile **Virtual_Size** arasındaki değer farklının yüksek olması ikili dosyanın paketlenmiş olma ihtimaline bir işaret olabileceğini gösterir.
    - UPX0 kısmı orijinal kodu yüklemek için ayrılmış alandır. UPX1 kısmında ise Stub ve orijinal kodun sıkıştırılmış hali bulunur. UPX0 alanının Raw_Size değerinin sıfır görünme nedeni, program çalıştırıldığında sıkıştırılmış haldeki kod parçasının orijinal hale getirilerek buraya yüklenecek olmasıdır.

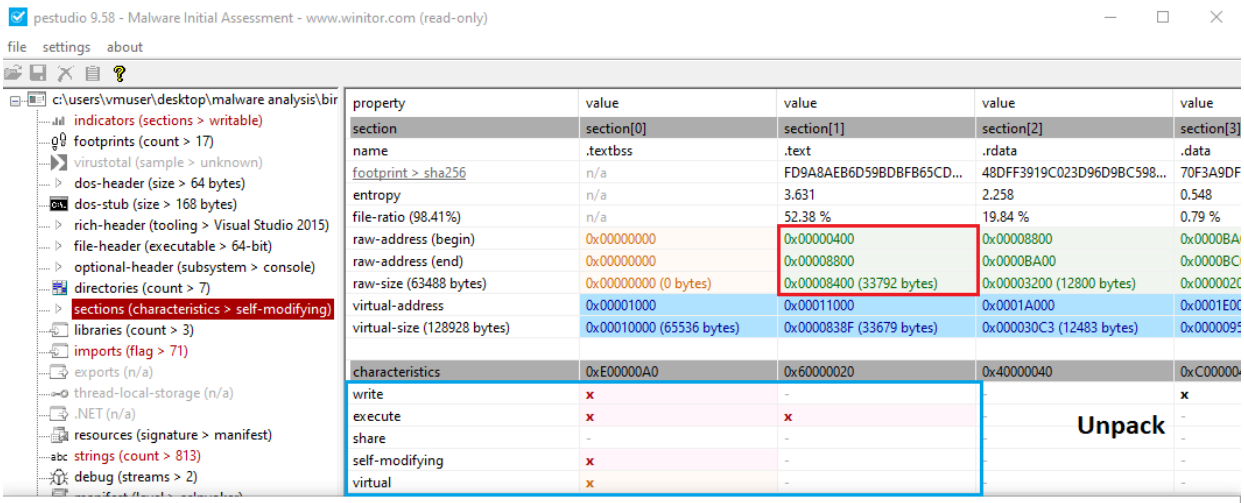


- İkili dosyanın paketlenmiş olabileceğini gösteren ikinci bir işaret ise program çalıştırıldığında içe aktarılan DLL dosyalarının ve bu DLL dosyalarının içerisinde kullanılan fonksiyonların miktarı olacaktır. Paketlenmemiş ikili dosyada bu değerler çok daha yüksek olurken paketlenmiş ikili dosyada çok daha az miktardadır.
    - Burada dikkat edilmesi gereken ikinci bir ipucu ise paketlenmiş ikili dosyada KERNEL32.DLL kütüphanesi içinde "**LoadLibraryA**" ve "**GetProcAddress**" fonksiyonlarının kullanılmasıdır.



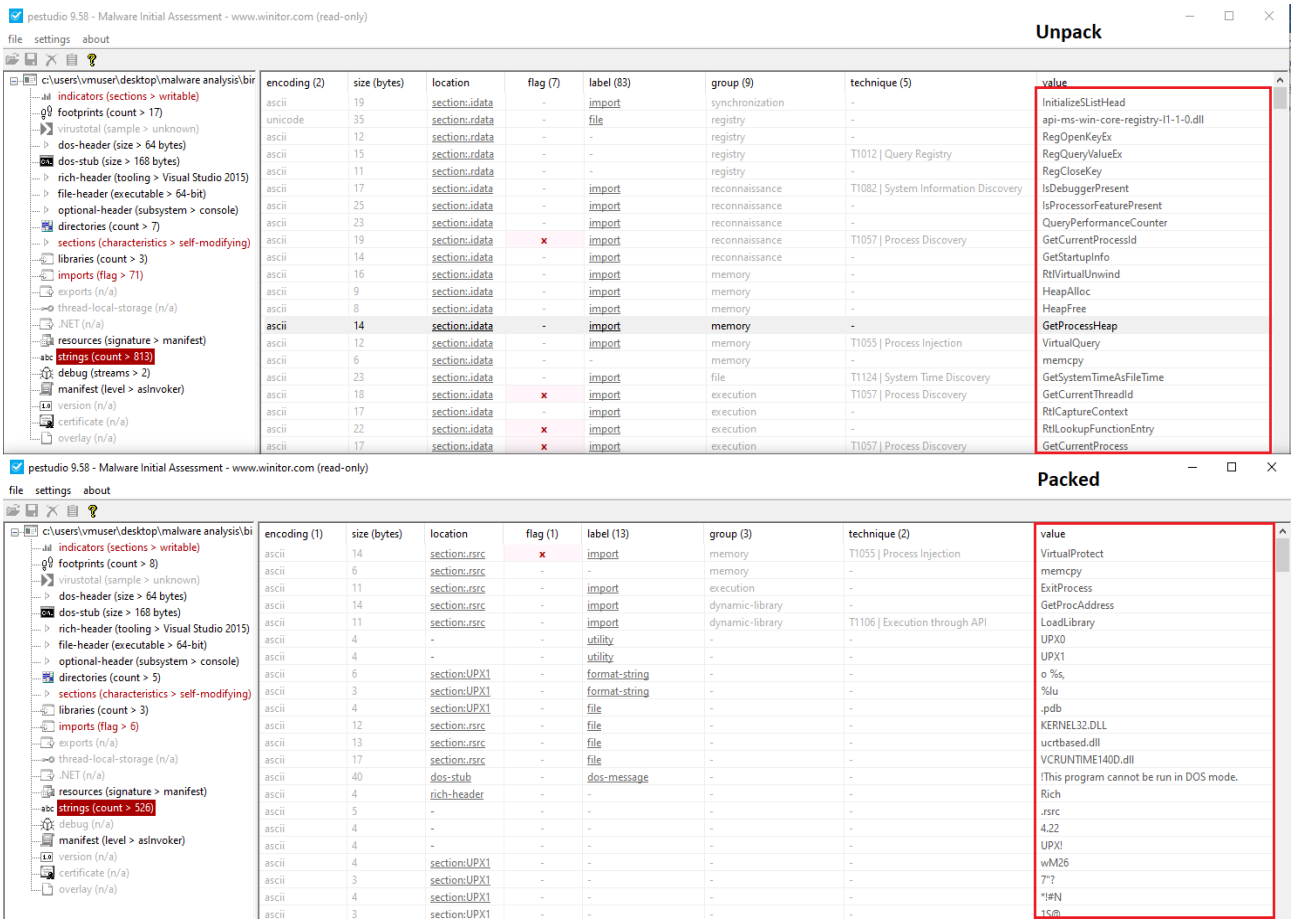
İkili dosyaları PE Studio aracı üzerinde incelediğimizde;

- CFF Explorer aracından farklı olarak Section kısımlarının yetkileri de dikkat çekiyor. Paketlenmemiş ikili dosyada olması gerektiği gibi sadece **.text** alanının çalıştırma yetkisi, **.data** alanının yazma yetkisi bulunurken, paketlenmiş ikili dosyada UPX0 alanının da UPX1 alanının da yama ve çalıştırma yetkileri bulunuyor.
    - CFF Explorer aracından görüldüğü gibi UPX0 alanının bellekteki boyutu 0 bayt görünüyor.
    - Alanların içindeki karakterler arası ilişki/anlamlılık yüzdesi olarak ifade edilebilecek Entropy değerlerindeki fark da göze çarpıyor. Paketlenmiş ikili dosyada paketlenmemiş ikili dosyaya kıyasla içerisinde bulundurduğu karakterler arası daha az ilişkiye/anlamlı yapılara sahip olduğu görülebilir. Bunun nedeni paketlenmiş ikili dosyada orijinal kod parçasının sıkıştırılmış durumda saklanmasıdır.



- String.exe gibi ikili dosya içerisindeki değerleri ASCII veya Unicode karakterlere çeviren araçlar kullanılarak dosya içeriğinde kullanılan kütüphane, fonksiyon isimleri görüntülenebiliyor. Bu nitelik de paketlenmemiş bir dosyada paketlenmiş bir ikili dosyaya kıyasla çok daha fazla sayıda anlamlı metinleri görüntülenebildiği görülüyor.

- Buradaki çıktılar doğrultusunda ikili dosyanın paketlenme süreci hakkında detaylı bilgi edinilebiliyor. Örnek olarak aşağıdaki çıktıda **GetProcAddress, LoadLibrary, UPX0, UPX1** gibi kısımlar verilebilir.



Manual Unpacking

İkili dosyanın paketlendiği anlaşıldıktan sonra sıkıştırılmış orijinal kod parçasını dışarı çıkarabilmek gerekiyor. Bunun için kodun bir disassembler üzerinde açıp analiz edilmesi gerekiyor. Analiz sürecinde aranması gereken ipuçları listelendiğinde;

-

- UPX (Ultimate Packer for Executables),
    - https://upx.github.io/

Kaynaklar

- https[:]//dlnhxyz.medium.com/manually-unpacking-a-upx-packed-binary-with-radare2-part-1-7039317c2ed8
- https[:]//dlnhxyz.medium.com/manually-unpacking-a-upx-packed-binary-with-radare2-part-2-be00860b5eac
- https[:]//www.youtube.com/watch?v=Npm5tuy1Pp4
- https[:]//tech-zealots.com/reverse-engineering/dissecting-manual-unpacking-of-a-upx-packed-file/
- https[:]//www.travismathison.com/posts/Manually-unpacking-malware/
- https[:]//medium.com/@0x8080/how-to-manually-unpack-a-binary-e56e18732b3b
- https[:]//www.manrajbansal.com/post/manually-unpacking-a-upx-packed-binary
- https[:]//www2.cs.arizona.edu/people/debray/Publications/static-unpacking.pdf
- https[:]//www.youtube.com/watch?v=J9I-Sewaihs
- https[:]//www.ired.team/offensive-security/defense-evasion/t1045-software-packing-upx
- https[:]//tech-zealots.com/reverse-engineering/dissecting-manual-unpacking-of-a-upx-packed-file/
- https[:]//www.ired.team/offensive-security/defense-evasion/t1045-software-packing-upx
- https[:]//sam0x90.blog/2020/06/06/unpacking-binary-101/