Rehber Hırsızı Hesperbot

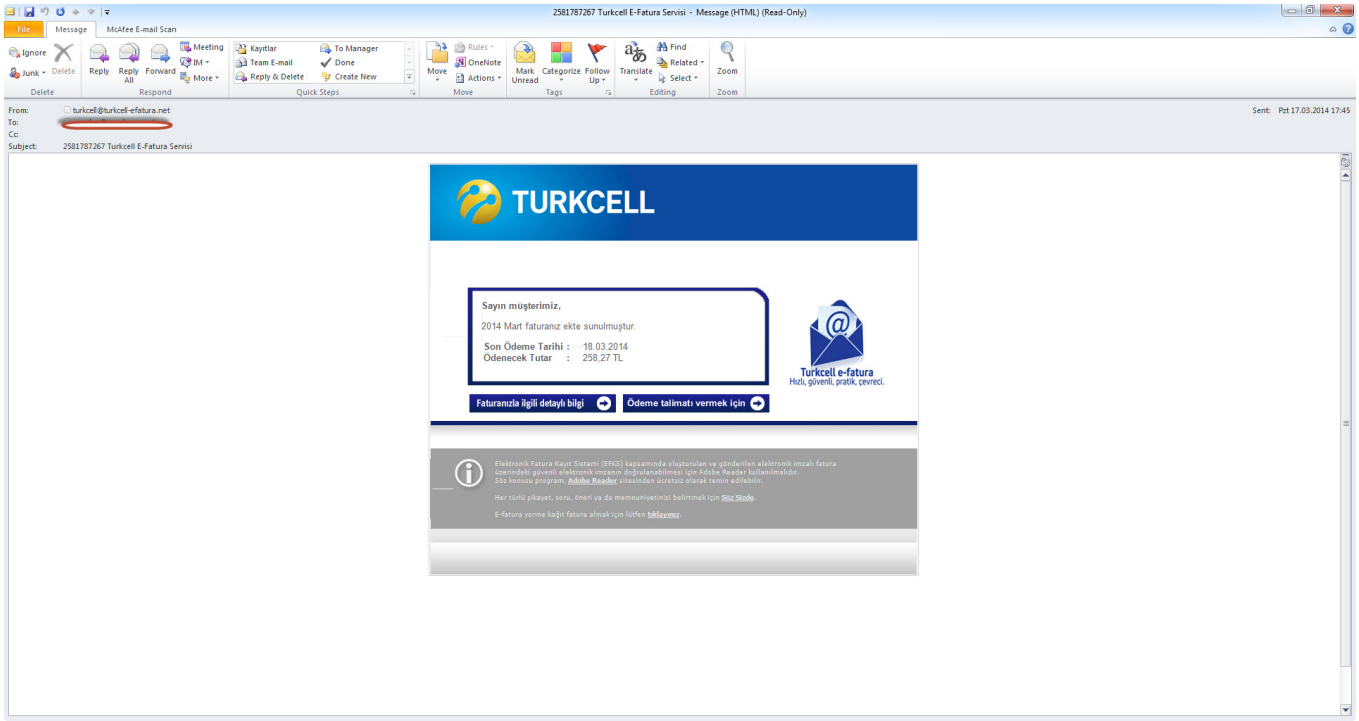
written by Mert SARICA | 2 June 2014

Son 1.5 yıldır hız kesmeden sahte fatura e-postaları ile ağına internet bankacılığı kullanıcılarını düşürmeye çalışan Hesperbot için son aylarda daha fazla mesai saati harcadığımı farkettim.

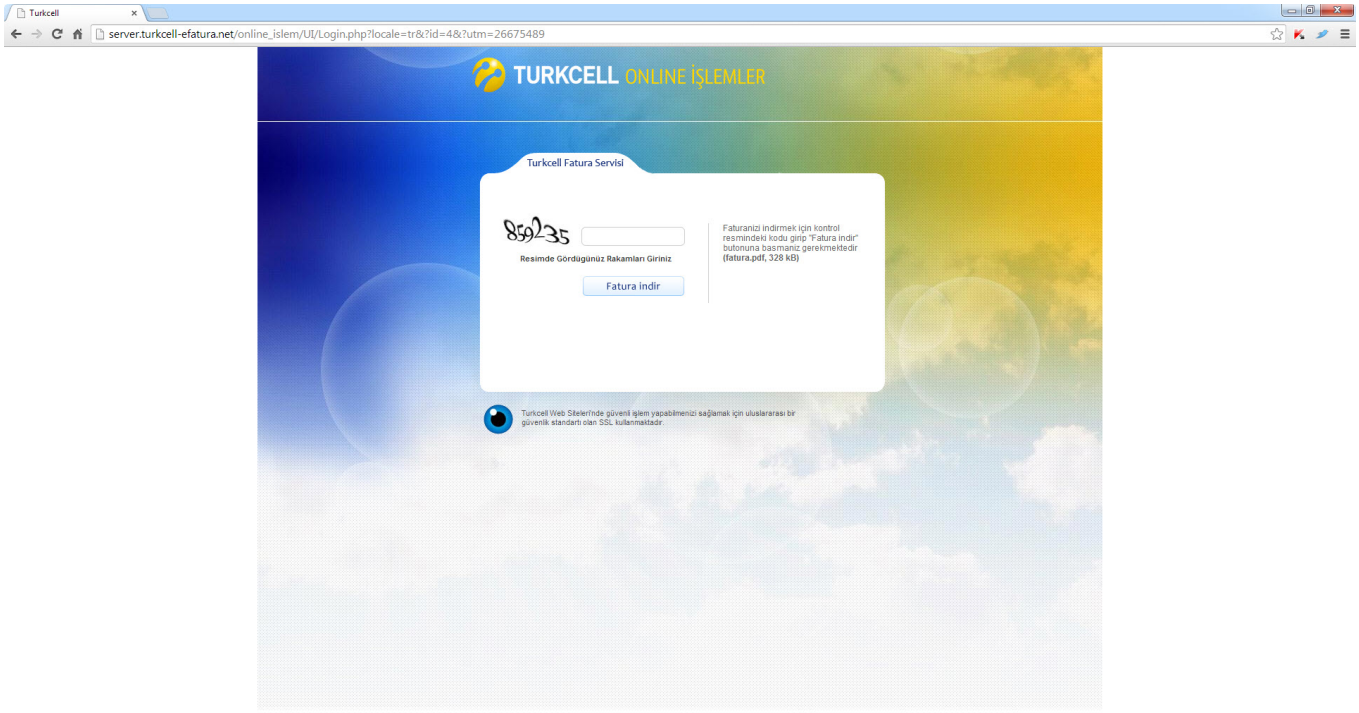
Son salgınların çoğunda, aynı tip sahte Turkcell fatura e-postasının gönderilmesine, art niyetli kişilerin fatura ve Turkcell kelimelerinden oluşan alan adlarını kullanıyor olmasına ve bankaların bununla ilgili uyarı mesajları gönderiyor olmasına rağmen, kullanıcıların hala bu oltaya düşüyor olmaları da, hem kurumlar hem de medya tarafından Hesperbot'a daha fazla dikkat çekilmesi gerektiğini gösteriyordu. (*Heartbleed virüsü gibi trajikomik haberlere imza atan medyamız, Hesperbot ile ilgili daha çok haber yer vermiş olsaydı eminim bu zamana dek daha az vatandaşımız bu dolandırıcıların tuzağına düşmüş olurdu!*)

Hesperbot'un en son salgında kullandığı sahte e-posta mesajını ve zararlı yazılımı yaymak için kullandığı web sitesini aşağıda görebilirsiniz.

Sahte Fatura E-postası:

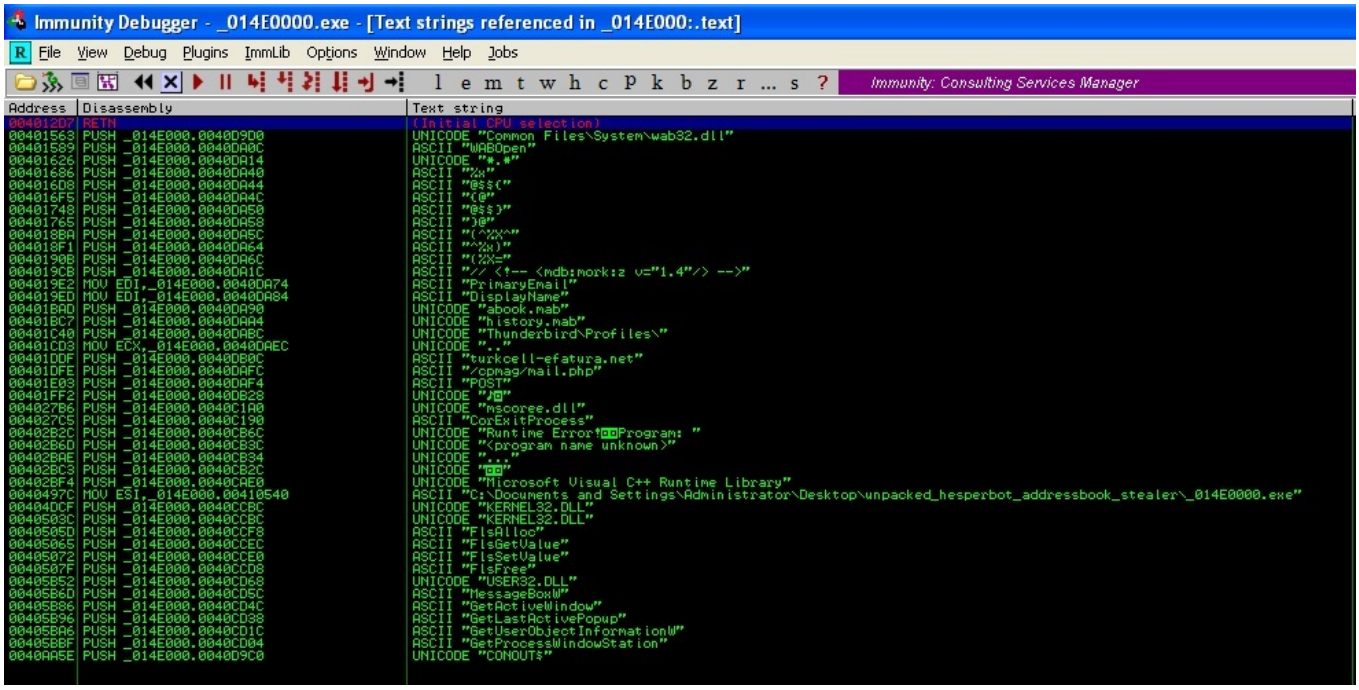


Sahte Fatura Web Sitesi:

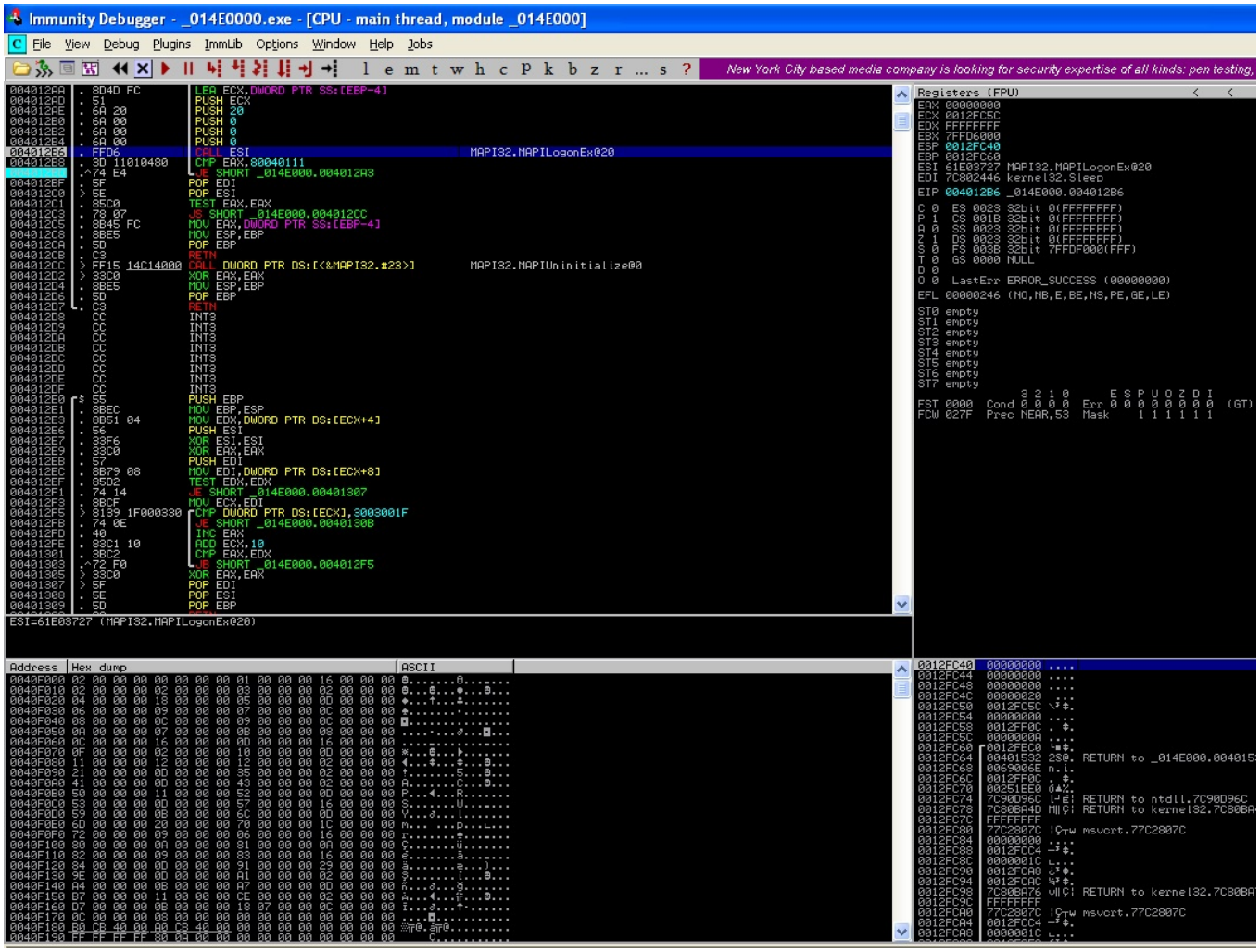


ESET’in Hesperbot raporu incelendiğinde, zararlı yazılımın hedef sistemden e-posta adreslerini çaldığı ve uzaktaki sunucuya bu bilgileri ilettiği belirtiliyordu fakat bununla ilgili teknik detaylara yer verilmemişti. Hesperbot’u detaylı bir şekilde incelerken, bulaştığı sistemdeki e-posta bilgilerini nasıl ele geçirdiğini ve uzaktaki sunucuya nasıl gönderdiğini inceleme fırsatım olduğu için bunu sizlerle de paylaşmak istedim.

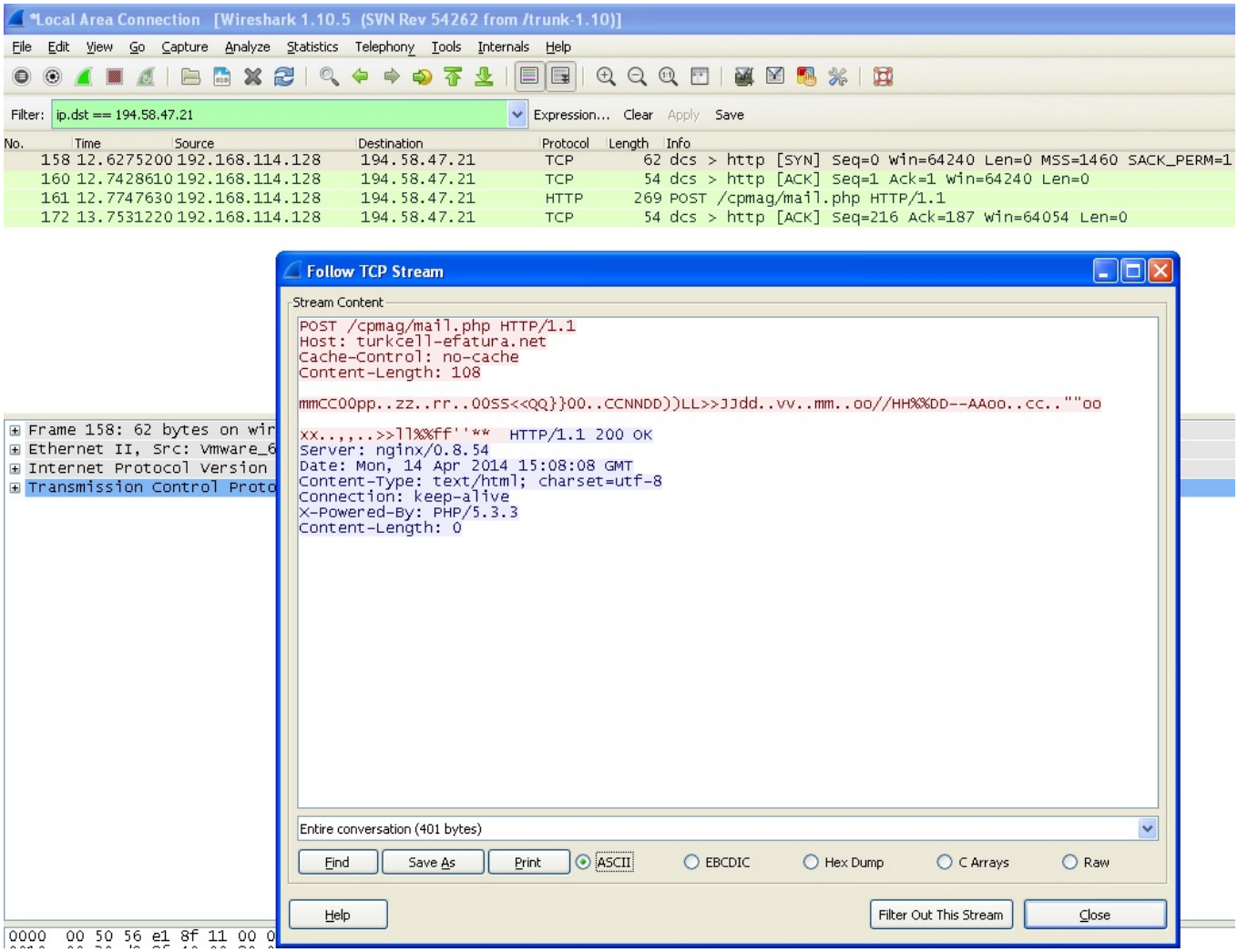
Hesperbot’un hedef sisteme bulaştıktan bir zaman sonra zararlı yazılımı yaymak amacıyla kullanmış olduğu sunucudan ege.xe adında bir yazılım indirdiğini tespit ettim. Paketlenmiş (packed) olan bu yazılımın uzantısını .exe olarak değiştirip her zamanki gibi Immunity Debugger aracı ile incelemeye başladım. Zararlı yazılımı paketinden çıkardıktan sonra ilk iş olarak karakter dizilerini (strings) incelemeye başladım.



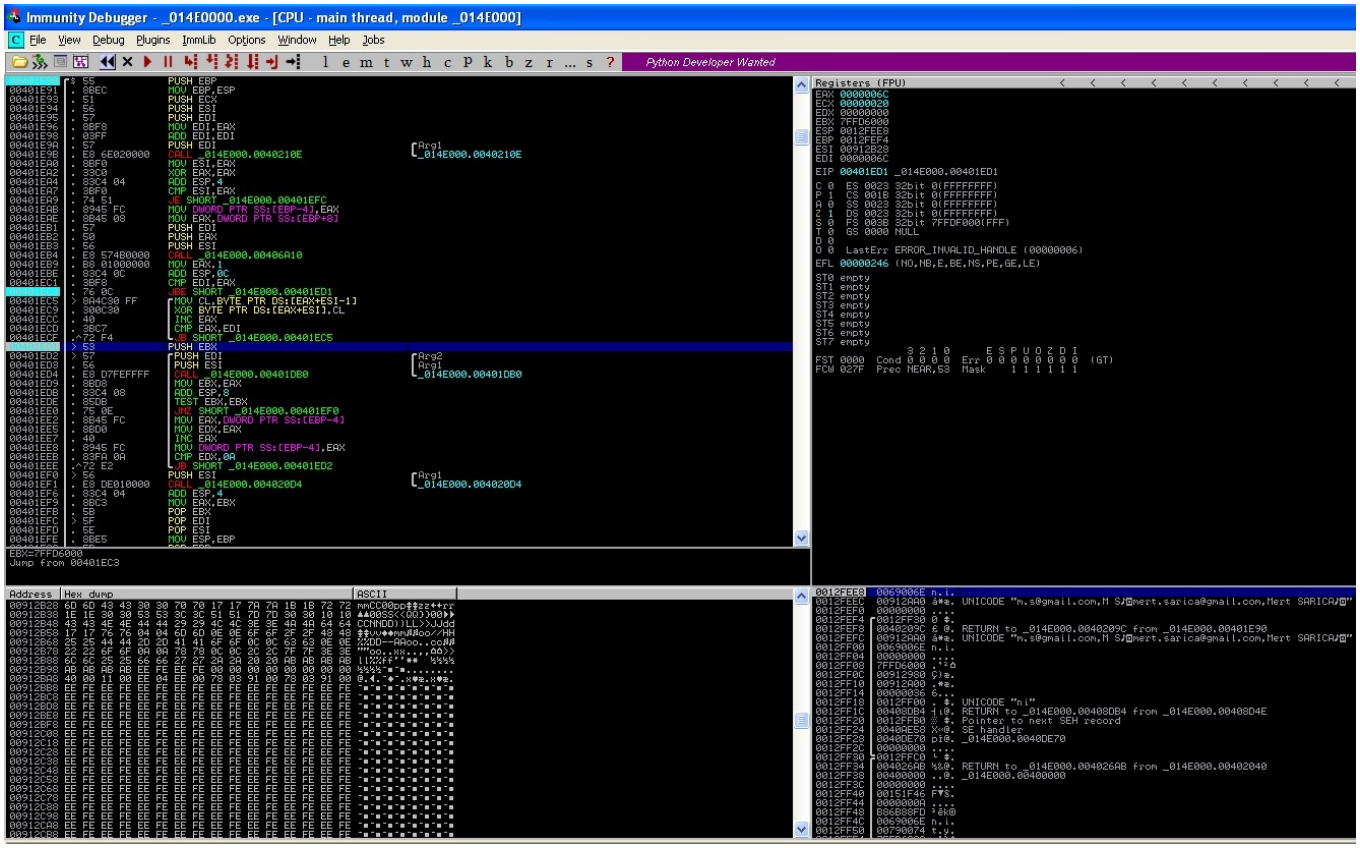
WABOpen fonksiyonundan bunun adres defterinde yer alan e-posta adres bilgilerini çaldığını tahmin etmem pek güç olmadı. Immunity Debugger ile yazılımı çalıştırdığımda herhangi bir HTTP trafiği oluşturmadığında birşeylerin ters gittiğini anladım. Yazılım çalışmasına rağmen herhangi bir web trafiği üretmemesi nedeniyle bir yerlerde kısır döngüye girmiş olabileceğinden şüphe ederek PAUSE butonuna bastım. Ardından kendimi ntdll.dll içinde bulduğum için Debug -> Execute till user code ile yazılımın koduna geçiş yaptım. Sistem üzerinde yüklü olan Outlook üzerinde geçerli bir profil olmadığı için çağırılan MAPILogonEx fonksiyonunun [MAPI_E_LOGON_FAILED(80040111)] hata alması nedeniyle kısır döngüden çıkamadığını gördüm ve akışın POP EDI komutu üzerinden devam etmesini sağladım. Akışın devam edebilmesi adına Outlook'un adres defterine 2 adet kayıt girdim ve programın devam etmesini sağladım.



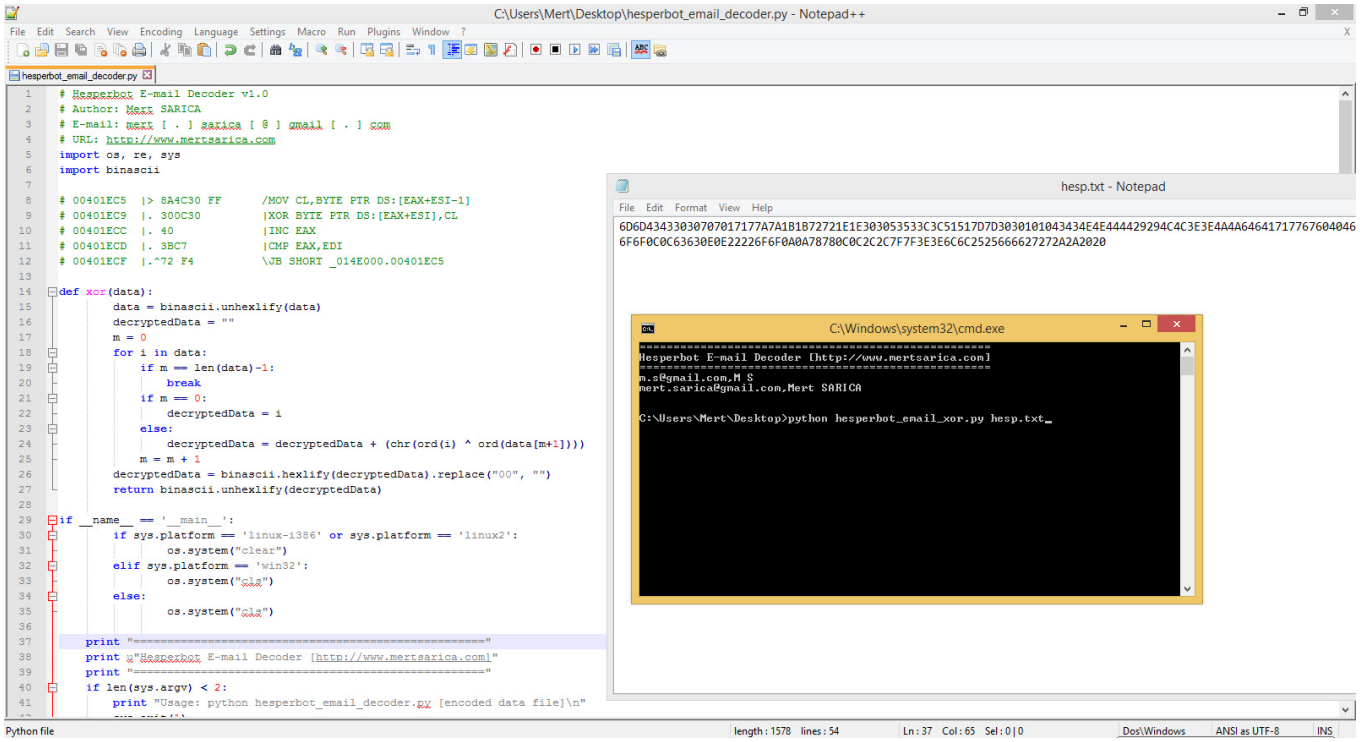
Yazılım son sürat çalıştıktan sonra oluşan trafiği Wireshark ile izlediğimde, uzaktaki sunucuya (http://turkcell-efatura.net/cpmag/mail.php) gönderilen e-posta bilgilerinin okunamaz halde (encoded) gönderildiğini gördüm ve bu fonksiyonu bulmaya karar verdim.



Immunity Debugger ile yazılım üzerinde biraz gezindikten sonra e-posta adreslerini gizleyen XOR işlemini buldum. Bu işlem ile e-posta adresinde yer alan her bir bayt, bir sonraki bayt ile (mert.sarica örneğinde m harfi e ile gibi) XOR işlemine sokuluyor ardından işlem tamamlandıktan sonra sunucuya gönderiliyordu.



XOR işlemi tersine çevrilebilir olduğu için Python ile Hesperbot Email Decoder adında ufak bir araç yazarak ağ trafiğinden elde edilen gizlenmiş e-posta adreslerini okunur hale getirebildim.



Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.