Zararlı Görüntü

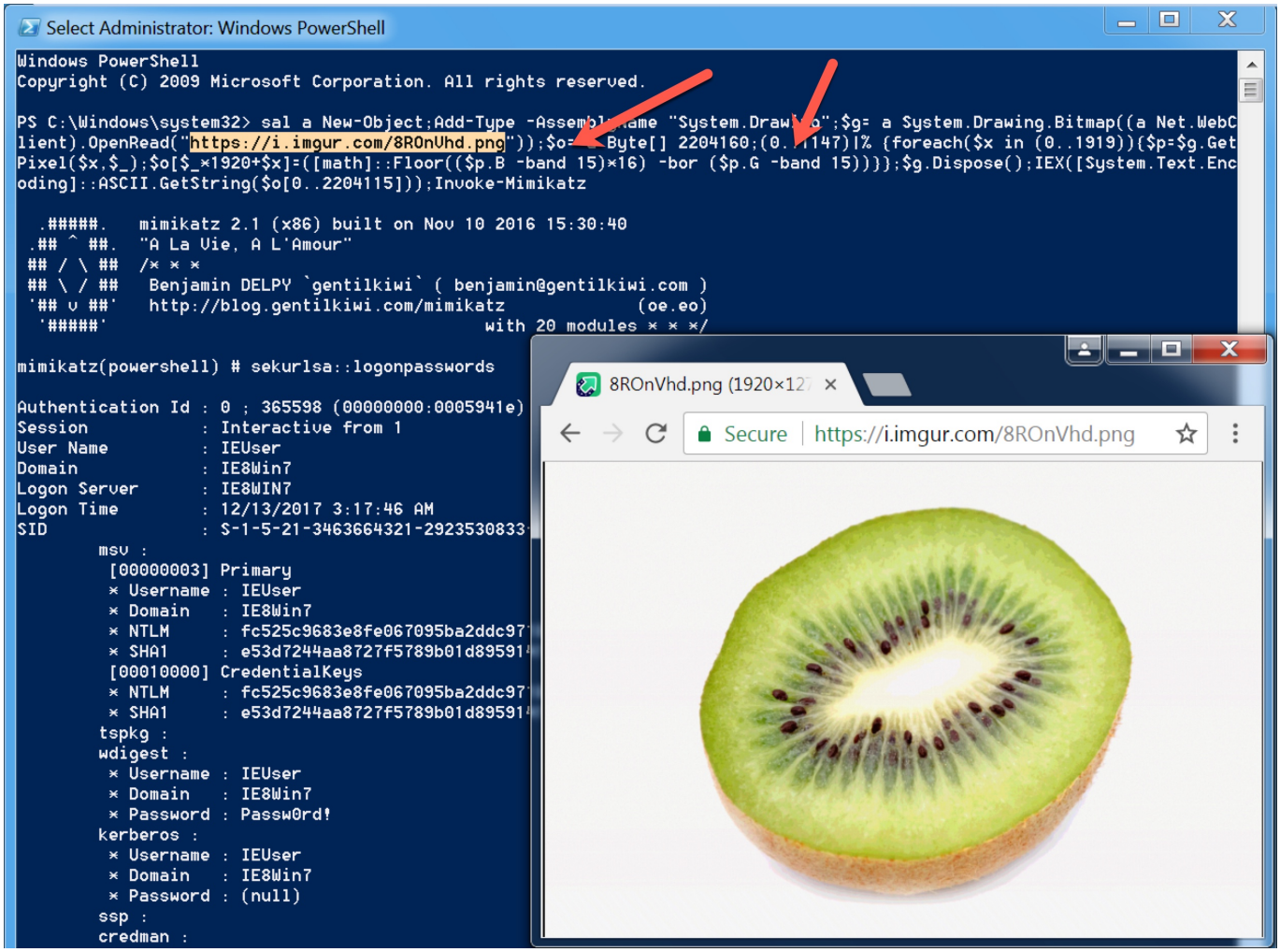
written by Mert SARICA | 1 December 2022
If you are looking for an English version of this article, please visit here.

Türkiye'deki kurumları da hedef alan Muddy Water gibi organize siber saldırı gerçekleştiren APT gruplarının son yıllarda gerçekleştirdikleri operasyonlarına baktığımızda, zaman zaman Steganografi tekniğinden faydalandıklarını görüyoruz. Bu teknik sayesinde siber saldırganlar, sosyal mühendislik saldırısı ile sızmaya çalıştıkları hedef son kullanıcı sistemine, gözle ayırt edilemeyecek zararlı kod parçasının görüntü dosyasının içinde indirilmesini ve çalıştırılmasını sağlamaktadırlar.

Steganografinin ilk kullanımına M.Ö 400'lü yıllarda "Tarihin babası" olarak anılan Herodot'un Historia olarak bilinen eserinde rastlanmaktadır.Bu eserde Demaratus, Yunanistan'a yaklaşan bir saldırıyı tahta bir tabletin üzerine kazıdıktan sonra üzerini balmumu ile kaplar ve mum eritildikten sonra tablette saldırıya dair uyarı ortaya çıkar.

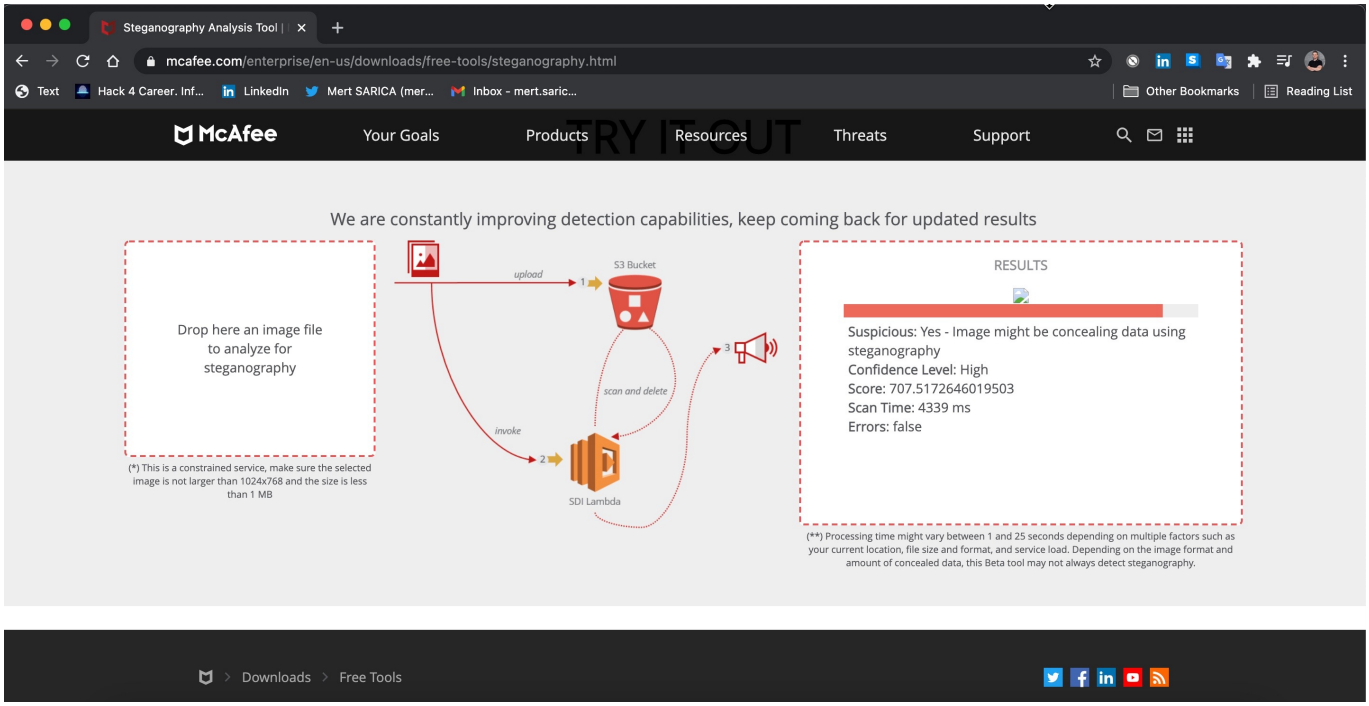
Hem 2018 yılından bu yana haberlere konu olması hem de Mitre'nin T1027 tekniği sayfasında yer alması sebebiyle PNG görüntü dosyasının piksellerine Powershell kodu eklemeye yarayan Invoke-PSImage aracı dikkatimi çekti. Internette ufak bir araştırma yaptığımda bu araç ile oluşturulmuş PNG dosyasındaki Powershell kodunu ortaya çıkaran bir araç bulamadığım için olay müdahale (IR) uzmanlarına fayda sağlayacak bir araç hazırlamaya karar verdim.

İlk olarak Invoke-PSImage aracının nasıl çalıştığını anlamak için kaynak koduna göz attığımda hedef olarak verile bir PNG dosyasının R (RED), G (GREEN), B (BLUE) renk kodlarından (her biri 8 bayt boyutundadır) G ve B'nin en düşük değerlikli bitleri (Least Significant Bit-LSB) ile oynayıp (bu sayede görüntü kalitesi düşse de insan gözü bu farkı kolay kolay anlayamaz) her defasında aynı aritmetik işlemlere ([math]::Floor((`$p.B-band15)*16)-bor(`$p.G -band 15)) sokarak gizlenmiş Powershell kodunu çalıştırdığını öğrendim. Hep aynı aritmetik işleme soktuğu için işlemi tersine çevirerek gizli Powershell koduna ulaşmak pratikte mümkün olduğu için amacıma bir adım daha yaklaşmış oldum.

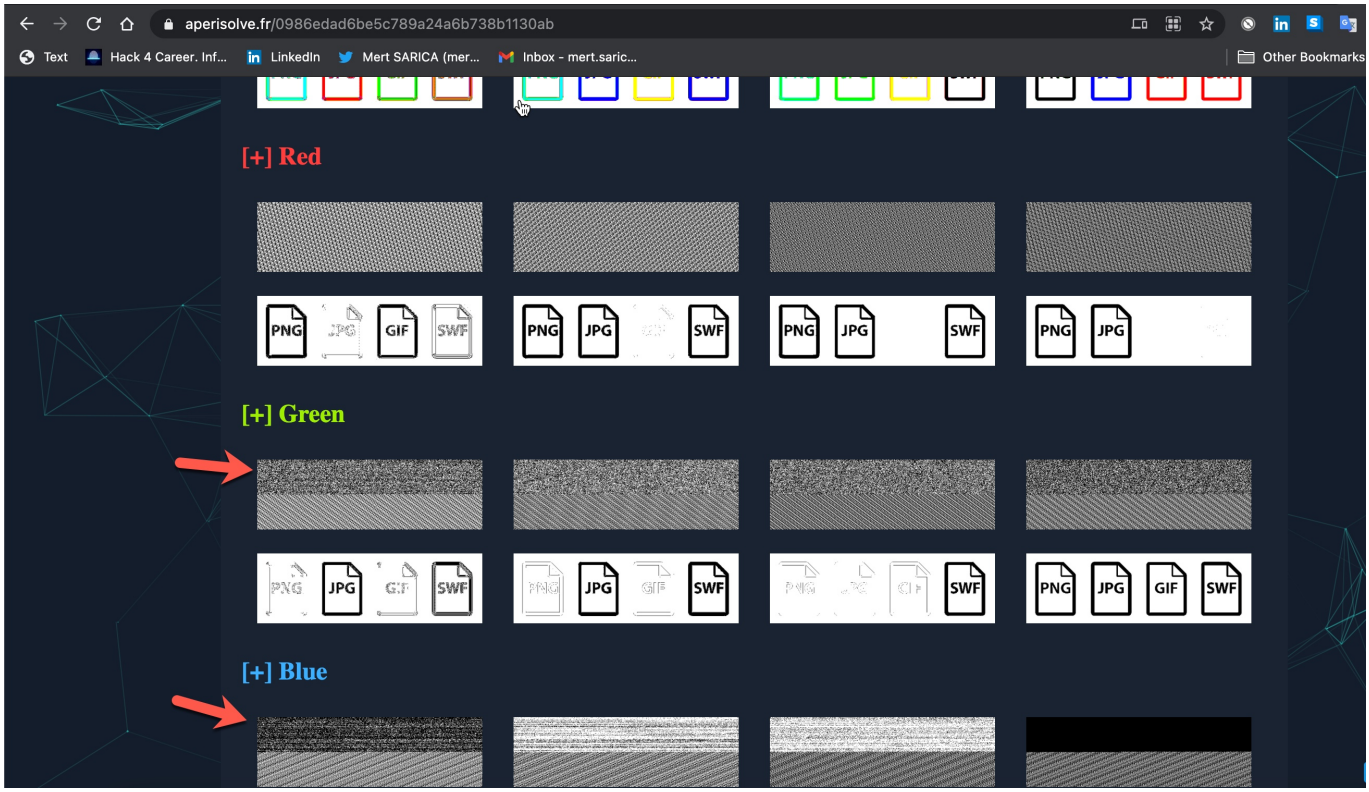


Aracı hazırlamadan önce hedef görüntü dosyasında gizli bir mesaj olup olmadığını anlamanın kolay bir yolu var mı diye araştırırken çok sayıda aracın yanısıra iki tane oldukça faydalı web sitesi ile karşılaştım. İkisinde de Muddy Water APT grubu tarafından bir operasyonda kullanılan EPUWBt3.png dosyasını analiz ettiğimde tatmin edici sonuçlar ile karşılaştım.

Bunlardan ilki olan McAfee'nin web sitesine (FireEye – McAfee birleşmesi sonrasında bu aracı web sitesinden kaldırdılar) EPUWBt3.png dosyasını yüklediğimde dosyanın şüpheli olduğuna dair bir uyarı mesajı ile karşılaştım.

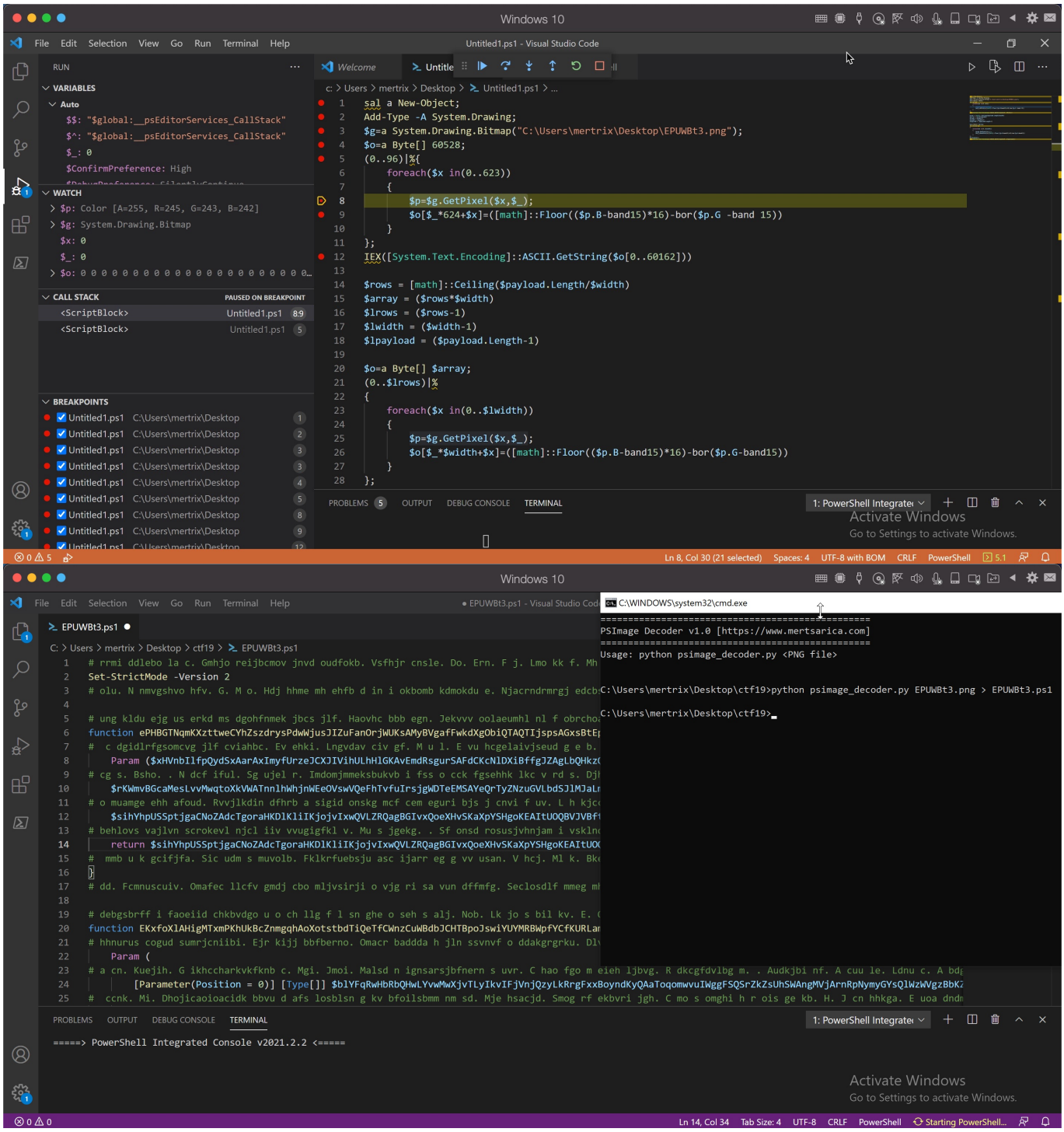


Aperi’Solve isimli diğer web sitesine EPUWBt3.png dosyasını yüklediğimde ise özellikle yeşil ve mavi renk kodlarındaki farklılık bu görüntü dosyasında şüpheli bir durum olduğunu ortaya koyuyordu.



Sıra aracı hazırlamaya geldiğinde EPUWBt3.png görüntü dosyasını çözen Powershell kodunu Visual Studio Code ile hata ayıklama ile adım adım analiz ettikten sonra Invoke-PSImage aracı ile görüntü dosyalarına gizlenen Powershell kodlarını ortaya çıkaran psimage_decoder.py isimli yeni aracım,

siber güvenlik uzmanlarının kullanımı için hayata merhaba demiş oldu.



Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not:

1. Bu yazı ayrıca Pi Hediyem Var #19 oyununun çözüm yolunu da içermektedir.